Oracle® Product Lifecycle Management
Implementation Guide
Release 12
Part No. B31989-01

December 2006
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send Us Your Comments</td>
</tr>
<tr>
<td>Preface</td>
</tr>
<tr>
<td>1 Implementation Overview</td>
</tr>
<tr>
<td>Set Up Oracle Applications Technology</td>
</tr>
<tr>
<td>Setup Prerequisites</td>
</tr>
<tr>
<td>Using Oracle iSetup</td>
</tr>
<tr>
<td>Setup Steps</td>
</tr>
<tr>
<td>Profile Options</td>
</tr>
<tr>
<td>2 Defining Responsibilities and Users</td>
</tr>
<tr>
<td>Defining Responsibilities and Users</td>
</tr>
<tr>
<td>Creating Responsibilities</td>
</tr>
<tr>
<td>Creating Internal Users</td>
</tr>
<tr>
<td>Setting Up Supplier Users</td>
</tr>
<tr>
<td>Setting Up Customer Users</td>
</tr>
<tr>
<td>3 Defining the Item Catalog</td>
</tr>
<tr>
<td>Overview of Item Catalog</td>
</tr>
<tr>
<td>Build the Item Catalog Text Index</td>
</tr>
<tr>
<td>Defining Item Catalog Categories</td>
</tr>
<tr>
<td>Defining Value Sets for User-Defined Attributes</td>
</tr>
<tr>
<td>Defining Item Attributes and Attribute Groups</td>
</tr>
<tr>
<td>Display Attachments on Item Attribute Pages</td>
</tr>
<tr>
<td>Setting Up Item Attribute Group Security</td>
</tr>
</tbody>
</table>
Implementing User-Defined Functions for Item Attributes .................................................. 3-17
Associating Item Catalog Attributes................................................................................... 3-21
Defining Item Attachment Categories............................................................................... 3-23
Associating Item Attachment Categories......................................................................... 3-24
Setting Up Item Templates................................................................................................. 3-24
Defining Lifecycle Phases................................................................................................. 3-26
Defining Lifecycles............................................................................................................ 3-28
Associating Lifecycles to Item Catalog Categories......................................................... 3-29
Defining Change Policies................................................................................................. 3-30
Defining New Item Request Types.................................................................................... 3-31
Defining New Item Request Workflows............................................................................ 3-33
Setting Up New Item Request for an Item Catalog Category........................................... 3-33
Defining Item Number and Description Generation Rules............................................... 3-35
Defining Item Catalog Search Criteria............................................................................. 3-37
Defining Item Catalog Display Formats........................................................................... 3-38
Defining Item Catalog Import Formats............................................................................ 3-38
Defining Report Templates Using XML Publisher............................................................ 3-39
Defining Cross Reference Types....................................................................................... 3-41

4 Defining Catalogs

Overview of Catalogs........................................................................................................ 4-1
Defining Catalog Categories.............................................................................................. 4-3
Defining Catalogs..................................................................................................................4-4
Implementing Catalog Security.......................................................................................... 4-4

5 Defining Change Categories and Types

Overview of Change Categories and Types...................................................................... 5-2
Build the Change Management Text Index........................................................................ 5-3
Defining Change Categories.............................................................................................. 5-3
Creating Priority Codes..................................................................................................... 5-8
Creating Reason Codes....................................................................................................... 5-9
Creating Classification Codes............................................................................................ 5-9
Creating Statuses................................................................................................................ 5-9
Defining Workflow Templates............................................................................................ 5-10
Attaching Workflow Templates......................................................................................... 5-12
Customizing Change Management Workflows.................................................................. 5-13
Creating Task Templates................................................................................................... 5-14
Defining Header/Line Type Attributes and Attribute Groups.......................................... 5-15
Defining Header Types....................................................................................................... 5-18
Defining Line Types........................................................................................................... 5-21
Associating Change Type Attributes...................................................................................... 5-22
Associating Change Line Type Attributes.............................................................................. 5-24
Setting Up Change Attribute Group Security........................................................................ 5-24
Implementing User Defined Functions for Change Type Attribute..................................... 5-25
Defining Change Category Search Criteria............................................................................ 5-28
Defining Change Category Display Formats......................................................................... 5-28
Defining Change Category Reports...................................................................................... 5-29

6 Defining Item Structures
Overview of Item Structures..................................................................................................... 6-1
Defining Structures................................................................................................................... 6-2
Display Formats and Search Criteria for Structures................................................................ 6-4
User Defined Attributes for Structures and Components....................................................... 6-5
Valid Component Type Rules.................................................................................................. 6-6

7 Setting Up E-Business Suite Attachments
Overview of E-Business Suite Attachments.......................................................................... 7-1
Managing Workspaces or Shared Folders.............................................................................. 7-1
Structuring Folders/Workspaces............................................................................................. 7-2
Attachment Review and Approval............................................................................................ 7-4
Change Management for Attachments.................................................................................. 7-5
Setting Up a Repository for Attachments.............................................................................. 7-6

8 Setting Up Document Management
Using Document Management................................................................................................. 8-1
Setting Up the Repository ....................................................................................................... 8-1

9 Document Change Management and Distribution
Document Change Management and Distribution Process Overview.................................... 9-1
Managing Document Changes............................................................................................... 9-13

10 Defining Roles
Overview of Roles.................................................................................................................. 10-1
Defining Roles....................................................................................................................... 10-2

11 Implementing Role Based Security
Overview Role Based Security............................................................................................... 11-1
Defining Role Based Security................................................................................................. 11-1
12 Understanding Product Data Synchronization

Overview of Product Data Synchronization................................. 12-1
Overview of Global Data Synchronization Network and 1SYNC......................... 12-2
Global Data Synchronization Network........................................ 12-3
Managing Global Trade Item Number Information................................. 12-11
Oracle 1SYNC Registry Messages.................................................. 12-12
Support 1SYNC Supplier Synchronization Suite................................. 12-14
Notification Processing............................................................... 12-16
Publish the Registered Information to Trading Partners............................ 12-18

A Workflows for Change and Document Management

Workflows for Change and Document Management............................. A-1

B Item Business Events

Business Events........................................................................ B-1

C Reading Data for User-Defined Attributes

Overview................................................................................ C-1

D Synchronizing Item User-Defined Attributes with Item Descriptive Elements

Overview................................................................................ D-1

E Change Management Business Events

Description of Change Management Business Events............................ E-1
Send Us Your Comments

Oracle Product Lifecycle Management Implementation Guide, Release 12
Part No. B31989-01

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on Oracle MetaLink and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.
Preface

Intended Audience

Welcome to Release 12 of the Oracle Product Lifecycle Management Implementation Guide.

See Related Information Sources on page x for more Oracle Applications product information.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.
Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

1 Implementation Overview
2 Defining Responsibilities and Users
3 Defining the Item Catalog
4 Defining Catalogs
5 Defining Change Categories and Types
6 Defining Item Structures
7 Setting Up E-Business Suite Attachments
8 Setting Up Document Management
9 Document Change Management and Distribution
10 Defining Roles
11 Implementing Role Based Security
12 Understanding Product Data Synchronization
A Workflows for Change and Document Management
B Item Business Events
C Reading Data for User-Defined Attributes
D Synchronizing Item User-Defined Attributes with Item Descriptive Elements
E Change Management Business Events

Related Information Sources

Integration Repository

The Oracle Integration Repository is a compilation of information about the service endpoints exposed by the Oracle E-Business Suite of applications. It provides a complete catalog of Oracle E-Business Suite's business service interfaces. The tool lets users easily discover and deploy the appropriate business service interface for integration with any system, application, or business partner.

The Oracle Integration Repository is shipped as part of the E-Business Suite. As your instance is patched, the repository is automatically updated with content appropriate for the precise revisions of interfaces in your environment.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.
Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using an Oracle Applications form can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.
Implementation Overview

This chapter covers the following topics:

- Set Up Oracle Applications Technology
- Setup Prerequisites
- Using Oracle iSetup
- Setup Steps
- Profile Options

Set Up Oracle Applications Technology

This overview explains how to implement the parts of Oracle Applications. You need to complete several setup steps including:

- Perform system wide setup tasks such as configuring concurrent managers and printers

- Manage function and data security, which includes setting up responsibilities to allow access to a specific set of business data and complete a specific set of transactions, and assigning individual users to one or more of these responsibilities.

Also, if your product uses Oracle Workflow to manage the approval of change requests (for example) or to send change notifications, you need to set up Oracle Workflow.

Related Topics

Oracle Workflow Guide

Setup Prerequisites

Before setting up this application, you must complete the required setup steps for the following:
• In Oracle Inventory, see "Setting Up" in the Oracle Inventory User’s Guide

• In Oracle Bills of Material, see "Setting Up" in the Oracle Bills of Material User’s Guide

• In Oracle Engineering, see "Setting Up" in the Oracle Engineering User’s Guide

• If you are using lifecycle tracking projects, refer to Setup and Implementation in Oracle Projects.

• In Oracle Sourcing, if you intend to collaborate with suppliers complete the setup steps for defining suppliers see the Oracle Sourcing Implementation and Administration Guide

• In Oracle Receivables, if you intend to collaborate with customers complete the setup steps for defining customers see "Setting Up Receivables" in the Oracle Receivables User’s Guide

Using Oracle iSetup

If you are implementing a new instance of the Oracle e-Business Suite and have relatively standard implementation requirements, you can use Oracle iSetup to rapidly configure the following applications:

• Oracle Purchasing

• Oracle Inventory

• Oracle Bills of Material

• Oracle Order Management

• Oracle Shipping

• Oracle Costing

• Oracle Planning

• Oracle Engineering

• Oracle Work in Process

• Oracle General Ledger

• Oracle Payables

• Oracle Receivables
• Oracle Cash Management

• Oracle Assets

Oracle iSetup simplifies the setup process by providing templates and predefined setup steps that may or may not apply to your enterprise.

Setup Steps

Following is a summary of the setup steps:
• Defining Responsibilities and Users
• Defining the Item Catalog
• Defining Catalogs
• Defining Change Categories and Types
• Defining Item Structures
• Setting Up Document Management
• Defining Roles
• Implementing Role-Based Security

Profile Options

Profile options enable or change the behavior of certain features in the application. Following is a description of each of the system profiles used in this application, and their default values.

<table>
<thead>
<tr>
<th>Profile</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGO: Enable exporting to Excel</td>
<td>Governs the action of exporting search results. Changing the value to Yes enables users to export certain search results data.</td>
<td>No</td>
</tr>
<tr>
<td>EGO: Internal User Default Role for Items</td>
<td>Governs the default role that is set up for internal users who access items. This role is assigned to internal users for all items in the organizations to which the users have access. Design Reviewer is seeded for Forms interoperability. <strong>Note:</strong> Forms does not have data security, so users must be given additional privileges on specific items (either directly or via inheritance) to perform actions on items. Users with the Design Reviewer role can view/access all item information. If you want to restrict access further, define a custom role and then reset the default role for all enterprise internal users. It is recommended that you always specify a role that minimally has the View Item privilege.</td>
<td></td>
</tr>
<tr>
<td>EGO: Maximum number of rows for sorting</td>
<td>Specifies the maximum number of rows to sort during an item search. If a search retrieves more than the maximum number of rows, the search results are displayed without sorting (a warning is displayed). <strong>Note:</strong> Specifying a value greater than 2000 may have a negative impact on search performance.</td>
<td>2000</td>
</tr>
<tr>
<td>EGO: Number of recent links to display</td>
<td>Specifies the number of links to display in the Recently Visited list</td>
<td>20</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------------</td>
<td>----</td>
</tr>
<tr>
<td>EGO: Restrict actions in Item form</td>
<td>Governs the Forms actions for items. Set to Yes to ensure that data level role-based security is honored in Forms. When set to Yes users will be unable to access the following functionality in Forms:</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>• Categories</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Catalog</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Revisions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Organization Assignments</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Attachments (disables attachments)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Keep the default value No if you wish to make all actions available in Forms.</td>
<td></td>
</tr>
</tbody>
</table>
ENG: Internal User Default Role for Changes

Reviewer

Governs the default role that is set up for internal users who access change objects. This role is assigned to internal users for all change objects in the organizations to which the users have access.

Reviewer is seeded for Forms interoperability.

**Note:** Forms does not have data security, so users must be given additional privileges on specific change objects (either directly or via inheritance) to perform actions on change objects.

Users with the Reviewer role can view/access all information for a change object. If you want to restrict access further, define a custom role and then set the default role for all enterprise internal users.

It is recommended that you always specify a role that minimally has the View Item privilege.
EGO GDSN ENABLED | Governs the GDSN Synchronization functionality.
---|---
No |

Changing the value to Yes enables GDSN functionality.

Once you turn the flag to yes, then

- GDSN default functional area appears
- GDSN attributes appear in setup list page
- GDSN attributes get associated to Item Catalog Category
- GDSN Syndicated functional area flag on the Create Item and Functional Classification page gets enabled.
Defining Responsibilities and Users

This chapter covers the following topics:

- Defining Responsibilities and Users
- Creating Responsibilities
- Creating Internal Users
- Setting Up Supplier Users
- Setting Up Customer Users

Defining Responsibilities and Users

You can securely collaborate with internal users, suppliers, and customers. You can set up users and grant them responsibilities to enable specific business functions. You can further restrict access to specific items, catalogs, or change management objects with role-based security.

<table>
<thead>
<tr>
<th>Task</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating Responsibilities</td>
<td></td>
</tr>
<tr>
<td>Creating Internal Users</td>
<td>Yes</td>
</tr>
<tr>
<td>Setting Up Supplier Users</td>
<td></td>
</tr>
<tr>
<td>Setting Up Customer Users</td>
<td></td>
</tr>
</tbody>
</table>

Creating Responsibilities

The top-level menu for the seeded responsibilities (Development Manager,
Development Engineer, and Supplier Engineer) is the EGO Manager Menu (EGO_MANAGER_MENU); you should view the seeded menus and functions before creating ones of your own. To view the seeded menus and functions, open the EGO Manager Menu in the Application Developer’s Menu form and click the View Tree button. In the Menu Viewer you can expand each menu as you decide which menus to include in your custom responsibilities. If a node appears with no prompt under a menu, then you should include that function to ensure that all the menu items work correctly. You can use menu exclusion rules when defining your responsibilities if you need to exclude any specific menus or functions for a user.

**Top level menu with seeded responsibilities**

![Menu Viewer Image]

<table>
<thead>
<tr>
<th>Seq</th>
<th>Prompt</th>
<th>Submenu</th>
<th>Function</th>
<th>Description</th>
<th>Grant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Items</td>
<td>EGO Items</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Structures</td>
<td>EGO Structures</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Change Management</td>
<td>EGO Change Management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Project</td>
<td>EGO Projects</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Manufacturing</td>
<td>EGO Routings</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Inventory Management</td>
<td>EGO Inventory Manager</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Cost Management</td>
<td>EGO Cost Management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Quality Management</td>
<td>EGO Quality Management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CAD/View3D</td>
<td>CAD/View3D User All M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>EGO Product Intelligence</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Menu Viewer

Menu Viewer

Entry Information
- Sequence
- Prompt
- Description

Menu
- ID: 1910778
- Name: EGO_MANAGER_MENU
- Display Name: EGO Manager Menu

Defining Responsibilities and Users   2-3
Creating Internal Users

You can define E-Business Suite users and assign them responsibilities that grant them access to specific E-Business Suite applications. After creating an employee, customer or supplier, create a user account to which you then link the employee, customer, or supplier. Each internal user should have an employee ID with an e-mail address specified so he or she can receive notifications. You can assign internal users a custom responsibility or one of the following seeded responsibilities:

- Development Engineer
- Development Manager
- Manufacturing Engineer
- Product Manager
- Program Manager
- Project Engineer
- Supplier Engineer
Setting Up Supplier Users

You can securely collaborate with suppliers. You can set up E-Business Suite users for each of the suppliers with whom you collaborate on product and component designs.

Assuming the supplier company has already been defined in Oracle Purchasing or Oracle Sourcing, you can register a supplier user in Oracle Sourcing using the Sourcing Super User responsibility. You need to specify the supplier’s e-mail address as the user ID so that the system can e-mail the supplier his or her password. After the supplier user is registered via Oracle Sourcing, you can assign him or her a responsibility such as Supplier Engineer, which specifies the functions available to that supplier user. For example, the seeded responsibility Supplier Engineer provides a restricted set of functions that enable the supplier to view specific items (secured with item roles), create issues and change requests for an item, but not create or view change orders.

For more details about registering and setting up supplier users, see the Oracle Sourcing Implementation and Administration Guide and Oracle Applications System Administrator’s Guide.

Setting Up Customer Users

You can securely collaborate with your customers. You can set up E-Business Suite users for each of the customers with whom you collaborate on product and component information.

Assuming that you have already defined the customer company in the Oracle E-Business Suite, create customer users by selecting the Trading Community Manager responsibility. In the Customers – Standard form create a new Customer Type of Person. You must specify a valid Address.

Then navigate to the Party Relations window and assign the person to the customer company as follows:

**Relationship Type**

EMPLOYMENT

**Relation**

Employee Of

**Object**

Company name

**Start Date**

Specify a start date. After assigning a person to the customer company, select the System Administrator responsibility to define the customer user. Then, in the Users form, enter the required information and specify the customer person name in the Customer field.
For more details about setting up customer users, see the Oracle Receivables Users Guide and Oracle Applications System Administrator’s Guide.

Creating a new customer person

[Image of customer person creation interface]
Setting up party relations for a new customer person

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>Relation</th>
<th>Object</th>
<th>Object Type</th>
<th>Start Date</th>
<th>End Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYMENT</td>
<td>Employee Of</td>
<td>A. C. Networks</td>
<td>ORGANI</td>
<td>03-SEP-2003</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Defining a customer user

User Name: JWILLIAM
Description: 
Password: 
Password Expiration: 
- Days: 
- Accesses: 
- None: 

Person: Jane Williams
Customer: 
Supplier: 
E-mail: 
Fax: 

Effective Dates
From: 03-SEP-2003
To: 

Responsibilities

<table>
<thead>
<tr>
<th>Responsibility</th>
<th>Application</th>
<th>Description</th>
<th>Security Group</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development Engineer</td>
<td>Oracle Engineering</td>
<td></td>
<td>Standard</td>
<td>03-SEP-2003</td>
<td></td>
</tr>
</tbody>
</table>
Defining the Item Catalog

This chapter covers the following topics:

- Overview of Item Catalogs
- Build the Item Catalog Text Index
- Defining Item Catalog Categories
- Defining Value Sets for User-Defined Attributes
- Defining Item Attributes and Attribute Groups
- Display Attachments on Item Attribute Pages
- Setting Up Item Attribute Group Security
- Implementing User-Defined Functions for Item Attributes
- Associating Item Catalog Attributes
- Defining Item Attachment Categories
- Associating Item Attachment Categories
- Setting Up Item Templates
- Defining Lifecycle Phases
- Defining Lifecycles
- Associating Lifecycles to Item Catalog Categories
- Defining Change Policies
- Defining New Item Request Types
- Defining New Item Request Workflows
- Setting Up New Item Request for an Item Catalog Category
- Defining Item Number and Description Generation Rules
- Defining Item Catalog Search Criteria
- Defining Item Catalog Display Formats
Overview of Item Catalogs

Oracle enables you to define hierarchical taxonomies of products, components, and items. The Item Catalog contains all the items in the system; other catalogs are a sub-set of the item catalog. For example the Product Catalog contains all products that are sold. Customers can also define additional catalogs for browsing and reporting purposes. Each catalog has a hierarchy of categories containing assigned items. The item catalog and product catalog are pre-defined catalogs.

<table>
<thead>
<tr>
<th>Task</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build the Item Catalog Text Index</td>
<td>Yes</td>
</tr>
<tr>
<td>Defining Item Catalog Categories</td>
<td>Yes</td>
</tr>
<tr>
<td>Defining Value Sets for User-Defined Attributes</td>
<td></td>
</tr>
<tr>
<td>Defining Item Attributes and Attribute Groups</td>
<td></td>
</tr>
<tr>
<td>Setting Up Item Attribute Group Security</td>
<td></td>
</tr>
<tr>
<td>Implementing User-Defined Functions for Item Attributes</td>
<td></td>
</tr>
<tr>
<td>Associating Item Catalog Attributes</td>
<td></td>
</tr>
<tr>
<td>Defining Item Attachment Categories</td>
<td></td>
</tr>
<tr>
<td>Associating Item Attachment Categories</td>
<td></td>
</tr>
<tr>
<td>Defining Lifecycle Phases</td>
<td></td>
</tr>
<tr>
<td>Defining Lifestyles</td>
<td></td>
</tr>
<tr>
<td>Associating Lifestyles to Item Catalog Categories</td>
<td></td>
</tr>
</tbody>
</table>
Build the Item Catalog Text Index

To take advantage of the Oracle Text search features available such as keyword search, stemming search, and fuzzy search—run the concurrent program Item Catalog Text Index Build. Any user with the Development Manager responsibility can submit this concurrent program from the Submit Request window. Select the concurrent request parameter Action=CREATE when submitting the request.

Related Topics

Optimizing the Item Catalog Text Index, Oracle Product Lifecycle Management User’s Guide

Defining Item Catalog Categories

The item catalog contains all items defined for your company or organization. Within the item catalog, item catalog categories provide a way to logically classify items that have similar characteristics. So, you can classify a computer monitor in an item catalog category named Computer Parts and Components. Item catalogs are hierarchical, and relationships between the various item catalog categories (such as an item’s catalog category attributes) are inherited. You can define a rich hierarchical taxonomy of item catalog categories that enable you to set up item attribution, lifecycles and their change policies, security, attachment categories, results formats, criteria templates, and import formats.

When defining item catalog categories, do the following:

• Create a list of all your items.

• Classify the items in unique item catalog categories and sub-categories that suit your business needs.

• Consider which category is a sub-classification within a category (for example, SRAM, DRAM could be subcategories of the item catalog category Memory).

• List the required and optional item attributes for each item catalog category

Following are some of the basic item catalog category terms:
• **Item Catalog Categories**
  Item catalog categories inherit attributes from the parent category; so consider which attributes to associate at higher levels in the item catalog category. Inheritance helps to simplify maintenance.

• **Item Lifecycles**
  Item lifecycles enable you to track and control the lifecycle phases of items. You can specify an item lifecycle for each item catalog category. Item lifecycles are inherited from the parent item catalog category.

• **Change Policy**
  Item behavior is described by defining a change policy for each lifecycle phase in a given lifecycle.

• **Security**
  You can control item creation, viewing and update access by assigning a role on the item catalog category to a user or group of users.

• **Attachment Categories**
  You can classify a file that is attached to an item using an attachment category. Attachment categories are inherited from the parent catalog category, and those most commonly used can be defined at a higher node. You can then add more specific attachment categories for the child item catalog categories.

  **Note:** Catalog Groups defined in Oracle Inventory will appear as Item Catalog Categories.

For details about seeded base attributes, see the *Oracle Inventory User's Guide*.

### Defining Value Sets for User-Defined Attributes

User-defined attributes capture all the detailed information (such as cost information) about an object (items, change requests or change orders). The item catalog enables you to create user-defined attributes with validation logic to support the needs of your organization. To do so, create value sets and associate the value sets with user-defined attributes. Attributes can have a static or dynamic list of valid values, or a range of values.

Prior to creating value sets for user-defined attributes establish the validation rules for each attribute as follows:

• Choose the data type for the values.

• Know the range of the values.
• Know which set of actual values will satisfy the required condition for the attribute. For static values, specify a fixed set of values. Dynamic values are retrieved from a database table or view.

• Select the manner in which the values will be displayed: list of values (LOV), pop list (dropdown) or radio group.

The following table shows the validation rules for the attribute group Benchmark Rating.

<table>
<thead>
<tr>
<th>Attribute Group</th>
<th>Attributes</th>
<th>Data Type</th>
<th>Value Range</th>
<th>Values</th>
<th>Display</th>
<th>Create Value Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark Rating</td>
<td>Risk Rating</td>
<td>Number</td>
<td>1 to 5</td>
<td>Static: 1,2,3,4,5</td>
<td>Pop list Rating</td>
<td></td>
</tr>
<tr>
<td>Benchmark Rating</td>
<td>Quality Rating</td>
<td>Number</td>
<td>1 to 5</td>
<td>Static: 1,2,3,4,5</td>
<td>Pop list Rating</td>
<td></td>
</tr>
<tr>
<td>Benchmark Rating</td>
<td>I/O Subsystem Rating</td>
<td>Number</td>
<td>1 to 5</td>
<td>Static: 1,2,3,4,5</td>
<td>Pop list Rating</td>
<td></td>
</tr>
<tr>
<td>Benchmark Rating</td>
<td>Video System</td>
<td>Number</td>
<td>1 to 5</td>
<td>Static: 1,2,3,4,5</td>
<td>Pop list Rating</td>
<td></td>
</tr>
<tr>
<td>Benchmark Rating</td>
<td>Overall Rating</td>
<td>Number</td>
<td>1 to 5</td>
<td>Static: 1,2,3,4,5</td>
<td>Pop list Rating</td>
<td></td>
</tr>
<tr>
<td>Benchmark Rating</td>
<td>Failure Rate</td>
<td>Number</td>
<td>Min 0 Max100</td>
<td>Within the range</td>
<td>Text Field</td>
<td>Percent Range</td>
</tr>
<tr>
<td>Benchmark Rating</td>
<td>Supplier</td>
<td>Char</td>
<td>N/A</td>
<td>Dynamic: Queried from tables in the database</td>
<td>LOV</td>
<td>Primary Supplier</td>
</tr>
</tbody>
</table>

After analyzing the required validation rules, create the value sets as follows:
1. Specify the Name and Description.

**Important:** Once specified, you cannot edit the value set name.
2. Select the Data Type. The data type determines the values that are available in the value set. An attribute's data type must match the data type specified for that attribute's value set. In the preceding table, the data type Number has been specified for the value set Rating because it contains all numeric values. In comparison, the value set Primary Supplier uses the data type Char because it only contains text values.

   **Important:** You cannot edit the data type once the value set has been created.

3. Choose the value for Maximum Size if you wish to limit the user's input in the attribute text field. For example, in some cases you may wish to limit the number of characters in the attribute Date to 10 characters, or the number of characters in Name to 50. Keep the default value of 0 if you wish to omit this particular validation.

4. Select the Validation Type:
   - **None:** Select None to indicate that there is no explicit set of values against which the user's input is validated.
   - **Independent:** You define the explicit values against which the user's input is validated here. To create explicit values, select Independent and click Add Values. Enter each value by specifying its internal/display name, sequence and the start/end date.

For example, the value set Rating can have certain values disabled by setting the end date.

You can specify the way the values are displayed as follows:
   - **List of values:** Users select values by clicking the list of values (LOV).
   - **Pop list:** Users select values from a dropdown.

   **Note:** Display as Radio Group can be set while creating or editing attributes. For more details see Define Item Attributes and Attribute Groups. Oracle recommends that when you use the value set type Independent, you save it as a pop list.

   - **Table:** The explicit values against which the user's input is validated comes from a database table. To create explicit values, select Independent and click Add Values, or click Table and then click Edit Table Information. This information can be dynamically changed.
Note: Value sets using the Table validation type are always displayed as a list of values (LOV).

Example of the value set Primary Supplier

The Value Set Details page is displayed upon completion. You can edit certain settings after creation by clicking Update. You can find all existing value sets on the Value Sets page. You can search for value sets using the criteria Name, Description, Data type, or Validation type.
You can use one of the seeded value sets (Yes or No) that are automatically displayed when you select the Display Option for the attribute. You can see some of the seeded value sets on the Value Set page; these are used to support images and attachments in the long description on the Item Summary pages.

**Defining Item Attributes and Attribute Groups**

Every catalog item has a set of operational attributes that determines the behavior of features in other E-Business Suite applications. You can define attributes that more specifically identify items, their characteristics and specifications, and capture business process information. The item catalog enables you to create user-defined attributes with certain validation logic (for example, value sets) and indexes.

Attributes are defined by their names and values, and are saved within attribute groups. You can associate attribute groups with an item catalog category; consequently, items belonging to the same catalog category have the same attributes. You can reuse the same attribute group across different item catalog categories. You can also set up as many attributes as necessary to define an item. Later, users can enter the values for the attributes on the item pages. Although item categories inherit attributes from the parent category, each child category can also have its own unique attributes. You can also configure how attributes are displayed to improve usability. You can specify attributes at the item or item revision level.

**See Also:** For details about associating attributes, see Associating Item Catalog Attributes.

Prior to creating user-defined attributes you should:

- Group related attributes within the same attribute group. The following table shows some examples of attribute groups.

<table>
<thead>
<tr>
<th>Attribute Group</th>
<th>Attributes</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Range</td>
<td>Value Set for Percentage Range Validation</td>
<td>Number</td>
</tr>
</tbody>
</table>
Capacitor Specifications

<table>
<thead>
<tr>
<th>Capacitor Specifications</th>
<th>Dielectric</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Tolerance</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Minimum Temperature</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Maximum Temperature</td>
<td>Number</td>
<td></td>
</tr>
</tbody>
</table>

Capacitor Packaging Specs

<table>
<thead>
<tr>
<th>Capacitor Packaging Specs</th>
<th>Package Type</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Case Size</td>
<td>Number</td>
<td></td>
</tr>
</tbody>
</table>

- Create an object role (item) with custom privileges to define attribute group security (if necessary). You can secure the attribute group by setting Edit/View privileges. Later on, only users with certain roles can view or edit those attributes. See Also: For more details, see "Implementing Item Attribute Group Security."

- Determine the data type of the attributes (for example, number, char, date).
  
  **Important:** You cannot edit the data type after you save the attribute.

- Set up the Unit of Measure Classes (for example, Currency) and Units of Measure (for example, US Dollars).

- Establish the validation rules for each attribute, if necessary. For more details, see Defining Value Sets for User Defined Attributes.

- Decide whether you want the attribute to be searchable. If you want the attribute to be searchable, then (when creating the attribute) select an indexed column in the database. Ensure that the Indexed checkbox is selected before you complete the attribute definition. If no indexed columns are available in the database, and you still want a searchable attribute, select a non-indexed column, and ensure the Indexed checkbox is selected before completing the attribute definition; then the database column is automatically indexed. If you do not want the attribute to be searchable, and the only columns available in the database are indexed, then ensure that the Indexed checkbox is not selected before completing the attribute definition.

- Choose the display format of the attribute group:
• **Single row** Displays a Text field with a value.

• **Multi-row** Displays multiple values (rows) for the same attributes (column) in a table.

  **Important**: Define which attributes or combination of attributes will maintain uniqueness of records in cases where the attribute group is displayed as multi-row.

The figure on "Defining item attributes and attribute groups" shows how the display options are set and used when specifying the value for attributes. A few attributes (such as Risk Rating, and Quality Rating) are set to display as a text field. However, the value set Rating is saved as a pop list; therefore, the attribute values are selected from the dropdown. The attribute Supplier uses the Primary Supplier value set, which is a list of values. The attribute Mean Time Between Failure uses the data type Number. Failure Rate also uses the Number data type and the Unit of Measure (UOM) is Percent.

**Defining item attributes and attribute groups**

**Attribute Group Details**

Data Security

Attributes

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Internal Name</th>
<th>Display Name</th>
<th>Data Type</th>
<th>Display As Value</th>
<th>Set Name</th>
<th>Enabled</th>
<th>Required</th>
<th>Unique</th>
<th>Columns</th>
</tr>
</thead>
</table>
| 1        | Risk_Rating  | Risk Rating  | Text Field| Rating           | Yes      | No      | No       | Yes    | N_D_E_ATF1
| 2        | Quality_Rating| Quality Rating| Text Field| Rating           | Yes      | No      | No       | Yes    | N_D_E_ATF2
| 3        | CPU_Rating   | Atomic CPU Rating| Text Field| Rating           | Yes      | No      | No       | Yes    | N_D_E_ATF3
| 4        | Video_Subsystem| Video Subsystem| Text Field| Rating           | Yes      | No      | No       | Yes    | N_D_E_ATF4
| 5        | Subsystem_Rating| V0 Subsystem Rating| Text Field| Rating           | Yes      | No      | No       | Yes    | N_D_E_ATF5
| 6        | OSE_Rating   | OSE Rating    | Text Field| Rating           | Yes      | No      | No       | Yes    | N_D_E_ATF6
| 7        | Supplier     | Supplier      | Text Field| Primary Supplier | Yes      | No      | No       | Yes    | N_D_E_ATF7
| 8        | MTBF         | Mean Time Between Failure (Char) | Text Field| MTBF             | Yes      | No      | No       | Yes    | N_D_E_ATF8
| 9        | MTBF         | Mean Time Between Failure (Time) | Text Field| MTBF             | Yes      | No      | No       | Yes    | N_D_E_ATF9
| 10       | Failure_Rate | Failure Rate (Char) | Text Field| Failure Rate     | Yes      | No      | No       | Yes    | N_D_E_ATF10

Also notice in the figure above that certain attributes are set with the Enabled parameter set to No, indicating that such attributes are not displayed. You can always disable certain attributes. However, you cannot delete attributes after an attribute group has already been associated with an item catalog category.
Defining item attributes and attribute groups (setting attributes values)

Benchmark Ratings

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor (CPU)</td>
<td>2.8</td>
</tr>
<tr>
<td>Video Subsystem</td>
<td>3.5</td>
</tr>
<tr>
<td>Supplier</td>
<td>3.2</td>
</tr>
<tr>
<td>Mean Time Between Failure</td>
<td>3.0</td>
</tr>
<tr>
<td>Failure Rate</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Defining item attributes and attribute groups

The following figure is an example of the attribute group Marketing. Notice that the attribute Name is maintained as a Part of Unique Key and is set as required.

You can add/edit the unique key as long as doing so does not destroy the uniqueness (creating duplicates) of existing records.

Display Attachments on Item Attribute Pages

You can display items or change attachments as attributes on an item attribute page. Select a specific file from the existing item/change attachment list to display the file as a link within an attribute group on an item or change object page. You can then set up a Results Format that includes the attachment attribute as a column. Consequently, the link to the attachment is displayed in the search results.
To display an attachment on an item attribute page:

1. Create a value set that queries the values for the object primary key and retrieves the information for the object attachment data.

2. Create an attribute group with an attribute where Display As is set to Attachment.

3. Associate the value set to the attribute.

4. Associate the attribute group to an item catalog category.

5. Create an item page to display the attribute group.

6. Add attachments to the item attachment list.

See Also: For more details, see:

- Defining Item Attributes and Attribute Group
- Define Value Set for Attribute
- Associating Item Catalog Attributes

**Note:** To see Attachments in the Display As dropdown, your attribute should use the Number data type.

When creating a value set to display item attachments, select the Table validation type and the Number data type (as shown in the following table).

<table>
<thead>
<tr>
<th>Attribute Group:</th>
<th>Attribute Group:</th>
<th>Attribute Group:</th>
<th>Attribute Group:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>Computer</td>
<td>Computer</td>
<td>Computer</td>
</tr>
<tr>
<td>Specifications</td>
<td>Specifications</td>
<td>Specifications</td>
<td>Specifications</td>
</tr>
<tr>
<td>Attribute:</td>
<td>Attribute:</td>
<td>Value Set:</td>
<td>Value Set:</td>
</tr>
<tr>
<td>Release Notes</td>
<td>Release Notes</td>
<td>Item Attachment</td>
<td>Item Attachment</td>
</tr>
<tr>
<td>Data Type</td>
<td>Display as</td>
<td>Data Type</td>
<td>Validation Type</td>
</tr>
<tr>
<td>Number</td>
<td>Attachment</td>
<td>Number</td>
<td>Table</td>
</tr>
</tbody>
</table>
Create a value set for an attribute to display an item attachment

**Value Set Details**

<table>
<thead>
<tr>
<th>Value Set Name</th>
<th>Item Attachments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Item Attachments</td>
</tr>
<tr>
<td>Data Type</td>
<td>Number</td>
</tr>
<tr>
<td>Maximum Size</td>
<td>0</td>
</tr>
<tr>
<td>Validation Type</td>
<td>Table</td>
</tr>
</tbody>
</table>

**Validation Table Information**

<table>
<thead>
<tr>
<th>Table Application</th>
<th>Oracle Engineering Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name</td>
<td>FND_ATTACHED_DOCUMENTS, FND_ATTACHED_DOCUMENTS_TL, FND_ATTACHED_DOCUMENTS_D</td>
</tr>
</tbody>
</table>

**Value Column**

- **Name**: ILJ_DESCRIPTION
  - **Type**: C
  - **Size**: 0

**ID Column**

- **Name**: ATTACHED_DOCUMENT_ID
  - **Type**: N
  - **Size**: 0

**Meaning Column**

- **Name**: N/A
  - **Type**: N
  - **Size**: N/A

**Where Clause**

ILJANGUAGE = 'en' AND ILJ_ENTITY_NAME = 'ITEM_SYSTEM INDEX' AND ADOCICY_ID = ILJ_DOCUMENT_ID AND ILJ_DOCUMENT_ID = ADOCICY_ID AND ADOCICY_ID = 'OBJECT', ORGANIZATION_ID AND ADOCICY_ID = 'OBJECT', ORGANIZATION_ID

**Associate value set to the attribute**

**Attribute Group Details**

- **Internal Name**: Computer_Specs
- **Display Name**: Computer Specifications
- **Description**: N/A
- **Multi-Row**: No

**Data Security**

- **View Privilege**: N/A
- **Edit Privilege**: N/A

**Attributes**

<table>
<thead>
<tr>
<th>Select Item(s) and ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select All</td>
</tr>
<tr>
<td>Select Item(s)</td>
</tr>
<tr>
<td>Select Attribute</td>
</tr>
<tr>
<td>Select Classification</td>
</tr>
<tr>
<td>Select Data Level</td>
</tr>
<tr>
<td>Select Enabled</td>
</tr>
<tr>
<td>Select Display Pages</td>
</tr>
</tbody>
</table>

**Associations**

<table>
<thead>
<tr>
<th>Classification</th>
<th>Data Level</th>
<th>Enabled</th>
<th>Display Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Systems</td>
<td>n/a</td>
<td>Yes</td>
<td>Technical Specifications</td>
</tr>
</tbody>
</table>
Setting Up Item Attribute Group Security

When implementing role-based item security you can set up privileges to control the view and edit permissions for specific item attribute groups. You can control which users can view and/or edit certain attribute groups for an item by assigning a role granting those specific privileges. By default, an item role's View Item and Edit Item privileges control whether or not you can view or edit item attributes that are not controlled specifically at the item attribute group level. In other words, when implementing item security you do not have to specify a view or edit privilege for each item attribute group.

Example: Attribute Group Security

Suppose your company is designing, along with your supplier, a new motherboard for its next generation of desktop computers. To improve design collaboration you would like to securely share item information about the motherboard—both internally between departments and externally with your suppliers and contract manufacturers. The Supplier Engineer should only be able to view specific item attribute groups such as the Technical Specifications. The Supplier Engineer should not be able to view the Market Research attributes. The Engineering Manager and Marketing Manager should be able to view and edit the Market Research attributes, while a Design Engineer should
only view the Market Research attributes. There are three sets of Market Research attributes (attribute groups): Key Metrics, Target Markets, and Competitors.

**To implement attribute group security for this scenario:**

1. Select the Application Developer responsibility, navigate to the Form Functions form, and create Form Functions for each privilege that controls view and edit permissions for the Market Research attribute groups.

2. Select the Development Manager responsibility and navigate to the Setup Workbench. In the **Attribute Group Details** page for each Market Research-related attribute group (for example, Target Markets) specify the View Privilege (for example, View Target Markets) and Edit Privilege (for example, Edit Target Markets) in the Data Security section.

3. On the **Item Role Detail** page for the Marketing Manager and Engineering Manager roles, grant the following privileges:
   - View Target Markets
   - View Key Metrics
   - View Key Metrics
   - View Competitors
   - Edit Target Markets
   - Edit Key Metrics
   - Edit Competitors
   
   For the Design Engineer role, grant the following privileges:
   - View Target Markets
   - View Key Metrics
   - View Competitors

Do not grant any of the Market Research privileges to the Supplier Engineer role.
Defining Form Functions to create user-defined item role privileges

Defining Item Attribute group with view and edit privileges
Defining the Marketing Manager role with specific item attribute group privileges

**Item Role Detail**

Name: Marketing Manager  
Description: Marketing Manager  
Object Name: Item

### Item Privileges

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Target Markets</td>
<td>View Target Markets</td>
</tr>
<tr>
<td>Add/Delete Item Attachment</td>
<td>Add/Delete Item Attachment</td>
</tr>
<tr>
<td>Edit Competitors</td>
<td>Edit Competitors</td>
</tr>
<tr>
<td>Edit Item</td>
<td>Edit Item</td>
</tr>
<tr>
<td>Edit Key Metrics</td>
<td>Edit Key Metrics</td>
</tr>
<tr>
<td><strong>Edit Target Markets</strong></td>
<td><strong>Edit Target Markets</strong></td>
</tr>
<tr>
<td>Grant/Revoke Role on the Item</td>
<td>Grant/Revoke Role on the Item</td>
</tr>
</tbody>
</table>

Defining the Supplier Engineer role with no specific item attribute group privileges

**Item Role Detail**

Name: Supplier Engineer  
Description: Supplier Engineer  
Object Name: Item

### Item Privileges

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Item Revision List</td>
<td>View Item Revision List</td>
</tr>
<tr>
<td>View Item</td>
<td>View Item</td>
</tr>
</tbody>
</table>

### Role Mappings

<table>
<thead>
<tr>
<th>Type</th>
<th>Role Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issue</td>
<td>Change Supplier Engineer</td>
</tr>
</tbody>
</table>

Implementing User-Defined Functions for Item Attributes

You can define your own custom logic by adding user-defined functions and actions to existing item pages. Doing so saves you from having to customize an entire page.
Using different algorithms, you can calculate values by passing attribute values to functions. Actions—which can be displayed as buttons or links—become the trigger points for functions. You can determine the conditional visibility of the button or links, and also prompt users based on their input. You can also trigger a function when you save changes to attributes (for details, see "Appendix D: Synchronizing Item User-Defined Attributes with Item Descriptive Elements").

User-defined functions can be Java, URL, or PL/SQL functions. Prior to setting up user-defined functions and actions you should:

To set up user defined attributes:

1. Set up user-defined item attributes. Create the necessary item pages associated with the item catalog categories.

2. Determine which attributes are functions of other attributes. Create user-defined functions to perform the necessary calculations (for example, summation of costs, efforts, ratings).

3. Create custom privileges and roles if you need to secure access and control who can/cannot execute the function/action.

4. Determine the conditions for displaying or changing the prompt of the button/link that executes the function. For example, the button may not be displayed until certain required attributes have values entered. The name of the button may change depending on the values of certain attributes.

   Java Function: Place the Java class files in any directory and append this directory to the Apache servlet classpath.

   PL/SQL Function: Run the PL/SQL package in your custom schema and then create a synonym for this package in the APPS schema.

   URL Function: Specify an absolute or relative URL.

When setting up parameters for functions:

1. Select the parameter type for each parameter based on whether the corresponding attribute is providing an input parameter to the function or expecting a return value from the function (for example, input, output, input/output).

2. Specify the order sequence of the function parameters; this is the order in which the parameters are passed to a function or procedure. For example, you need to calculate time duration in days by using the attribute group "Duration in days" where Duration = End Date – Start Date.
## Defining the Item Catalog

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Attribute Group: Duration in Days</th>
<th>Attribute Group: Duration in Days</th>
<th>Mapping Attribute &amp; Parameters</th>
<th>Java Function: Duration</th>
<th>Java Function: Duration</th>
<th>Java Function: Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start Date</td>
<td>Date</td>
<td>Date 1 Date Input</td>
<td>Duration</td>
<td>Duration</td>
<td>Duration</td>
</tr>
<tr>
<td>2</td>
<td>End Date</td>
<td>Date</td>
<td>Date 2 Date Input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Duration</td>
<td>Number</td>
<td>Result Integer Return Value</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Once the function and parameters for Duration (based on the Java method that resides in the Java class) are set up, create an action to trigger the function on the item pages. Navigate to the attribute groups list for the corresponding item catalog category (such as Computer Systems) and select the Update action.

**When setting up an action:**

1. Add the Dynamic Visibility function (written in Java only) to the action if there is a certain condition that must be satisfied to view the action button or link. For example, if one or all of the fields are empty, then the user won’t see the button/link to execute the action.

2. Add the Dynamic Prompt function (written in Java only) to the action if the label for the button or link must be changed depending on a certain condition. For example, if the Total Implement Cost field is empty, then the button label is Apply; otherwise the button label is Apply Changes.

3. After creating the action, map the function parameters to the corresponding attributes.

The mapping section on the **Action Detail** page contains mapping information for function parameters. You can also map function parameters to the object's primary key value (for example, ORGANIZATION_ID, INVENTORY_ITEM_ID for item objects).

**Important:** Dynamic Prompt and Dynamic Visibility function parameters must be mapped to the corresponding attributes as...
well.

6. The following figures show the setup details for the Calculate Grade function. The function was created to calculate the overall rating for items belonging to the Motherboard item catalog category. Benchmark Rating attribute values were used as parameters.

*See Also:* For details about setting up item attributes see Defining Item Attributes and Attribute Groups and Associating Item Catalog Attributes.

### Defining a function

**Function Details**

<table>
<thead>
<tr>
<th>Internal Name</th>
<th>Display Name</th>
<th>Data Type</th>
<th>Parameter Type</th>
<th>Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td>calculate_grade</td>
<td>Calculate Grade</td>
<td>PL/SQL</td>
<td>Function</td>
<td></td>
</tr>
<tr>
<td>Motherboard_Rating</td>
<td>Benchmark Rating</td>
<td>Number</td>
<td>Input</td>
<td></td>
</tr>
</tbody>
</table>

**Function Parameters**

<table>
<thead>
<tr>
<th>Action Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate Grade</td>
</tr>
</tbody>
</table>
**Defining the Item Catalog**

**Action associated to attribute group to trigger the function**

**Action Details for Calculate_Rating**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Mapped To</th>
<th>Mapped Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component/Rec</td>
<td>Attribute Group</td>
<td>Boss Rating</td>
</tr>
<tr>
<td>Last Time</td>
<td>Attribute Group</td>
<td>Quality Rating</td>
</tr>
<tr>
<td>Cost</td>
<td>Attribute Group</td>
<td>Atomic CPU Rating</td>
</tr>
<tr>
<td>Supplier/Rec</td>
<td>Attribute Group</td>
<td>I/O Subsystem Rating</td>
</tr>
</tbody>
</table>

**Item attribute page displays action button**

**Benchmarks Ratings**

Note: You can re-use the same attribute group throughout the item catalog for different item catalog categories. Children categories inherit attribute groups from the parent category. You cannot edit inherited data.

Consider which item catalog categories in the item catalog hierarchy need to display the attribute groups.
Associate an attribute group to a parent category only if you need to view those attributes in all items within that hierarchy. For example, the Benchmark Rating attribute group is associated with the Computer System item catalog category. The Computer System category is the parent category for Desktop and Laptop. All items belonging to the parent category Computer System, as well as to its children Desktop and Laptop, will have Benchmark Rating attributes displayed on the items pages.

To associate an attribute group to an item catalog category:

1. Add the attribute group to the Attribute Groups list for that item catalog category. You can associate attribute groups to an item or item revision.

   **Note:** Item level attributes are specific only to the Master Organization. However, the item revision level attributes are always organization-specific. For example, the Benchmark Rating attribute group is an item level attribute group and Desktop Specifications is associated at the item revision level. Item VI1004 is assigned to the master organization Vision Operations and to Seattle Manufacturing.

2. Define the item pages for the attribute groups. You can specify one or more attribute groups on a page or create a separate page for each attribute group. Item pages are also inherited from the parent item catalog categories.

   **Associate item catalog attribute and create item page**

   In the figure below you can see that the Desktop catalog category has five item pages, one of which is the Benchmark Rating page containing the Benchmark Ratings attribute group.
Defining Item Attachment Categories

Whenever users add attachments to an item, they specify an attachment category. Item attachment categories provide a way to classify attachments—this classification identifies the purpose of the attachment to the business object, and can also be used as one of the search criteria within an attachment list. If users do not specify an attachment category for an item catalog category, and none are inherited from its parent catalog category, they always have the option of using the attachment category Miscellaneous.

Note: Attachment categories inherited from a parent catalog category cannot be deleted at the child catalog category level.

Attachment categories are defined by the system administrator and are available across the various applications of the E-Business Suite. Attachment categories that you define can be made available to any E-Business Suite application. Thus, attachment categories are shared across the E-Business Suite applications and this should be kept in mind when defining any new attachment categories. Rules defining the usage of attachment categories are defined independently by each application.

You can define attachment categories in the Document Categories form. Before defining a new attachment category, verify that it does not already exist. If it already exists, then you can enable it for item catalog categories.
Associating Item Attachment Categories

Once you define an attachment category (also known as a "document category"), you can enable it for an item by associating it with an item catalog category.

All item catalog categories inherit attachment categories (in the same way they inherit item attributes) from their parent catalog category. So, before associating attachment categories with item catalog categories create a list of all attachment categories to be used in your item catalog. Consider which attachment categories are common among item catalog categories within each branch of the item catalog hierarchy. If an attachment category is applicable to all the child catalog categories, then the attachment category should be associated with the parent catalog category.

When associating Attachment categories with item catalog categories, you can specify the effectivity of this association by providing a Start Date and an End Date. If an attachment category association has an end date, that type of attachment cannot be used for the items belonging to the category beyond the specified date.

### Associating an attachment category with an item catalog category

<table>
<thead>
<tr>
<th>Attachment Categories</th>
<th>Item Catalog Category</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Reference</td>
<td>PLM High Tech</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specification</td>
<td>Motherboards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document</td>
<td>Motherboards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datasheet</td>
<td>Motherboards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td>Motherboards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collateral</td>
<td>Motherboards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To Buyer</td>
<td>Motherboards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To Mobile Receiver</td>
<td>Motherboards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To Mobile Pull Away</td>
<td>Motherboards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To Mobile Fisher</td>
<td>Motherboards</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Setting Up Item Templates

You can define Item Templates for the Item Catalog to include operational attributes. You can also include user-defined attributes in an Item Template once it is associated with an Item Catalog Category. Item Templates may be inherited for use down through the item catalog hierarchy. You can specify a default Item Template for an Item Catalog Category. A reset is available to restore the default attributes and values inherited down through the item catalog hierarchy.

**Benefits:**
• Default operational and user-defined attributes to speed up the process of specifying item attribute values

• Enforce consistency of item attribute values for items belonging to different item catalog categories or item types. For example, you may have an item template for all purchased computer hard-drive assemblies.

How to Set Up Item Templates:
1. Create Item Template in Setup Workbench to specify operational attributes

2. Associate item templates with any item catalog category. Item templates are inherited down through the item catalog hierarchy

3. Specify which item template is the default for the item catalog category

4. Specify values for any associated user-defined attribute groups to the item catalog category. When you define a template at a parent category, the same template is inherited by all child categories. You can modify attribute values in the inherited template and if needed revert to inherited values at a later time.

*Updating item template for user-defined item category attributes*

<table>
<thead>
<tr>
<th>Attribute Group</th>
<th>Operating Conditions</th>
<th>Purchased Item</th>
<th>Item Catalog Category</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal Temperature</td>
<td>70</td>
<td>Celsius</td>
<td>Motherboards</td>
<td>✔</td>
</tr>
<tr>
<td>Humidity</td>
<td>60</td>
<td>%</td>
<td>Motherboards</td>
<td>✔</td>
</tr>
<tr>
<td>Maximum Wet Bulb Temperature</td>
<td>20</td>
<td>Celsius</td>
<td>Motherboards</td>
<td>✔</td>
</tr>
<tr>
<td>Minimum Dew Point</td>
<td>65</td>
<td>Celsius</td>
<td>Motherboards</td>
<td>✔</td>
</tr>
<tr>
<td>Chassis Fan Speed</td>
<td></td>
<td></td>
<td>Motherboards</td>
<td>✔</td>
</tr>
</tbody>
</table>
### Creating/updating item template for item master operational attributes

**Template Details**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Controlled At</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing</td>
<td>Organization Item</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Processing</td>
<td>Organization Item</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Postprocessing</td>
<td>Organization Item</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Final</td>
<td>Organization Item</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Variables</td>
<td>Organization Item</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Cumulative Mfg</td>
<td>Organization Item</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Cumulative Total</td>
<td>Organization Item</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Lead Time Limit</td>
<td>Organization Item</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

**Important:** The ability to associate a lifecycle tracking project to an Item and/or Item revision is only available to customers who have licensed Product Lifecycle Management, and it is not available to licensees of Product Information Management Data Librarian.

An item lifecycle is a sequence of project phases. Each phase represents a set of tasks and deliverables that are required before promoting the item to the next phase. For example, the lifecycle phases for a computer component lifecycle might be: Concept, Design, Prototype, Pre-Production, Production, and Retirement. If you are using Oracle Projects you can associate the item lifecycle with a project based on a project template or an existing project. The project workplan's top tasks can be mapped to the lifecycle phases.

**Important:** If you are using a lifecycle tracking project, then set the current phase within the project workplan. This action will automatically update the item phase.

Define lifecycle phases in Oracle Projects. For more details about defining lifecycle phases, see the *Oracle Projects User’s Guide*.

---

**How to Use Item Templates**

1. During the new item creation process you can specify an item template or use the default if one was setup at the item catalog category level.

2. The operational attribute values specified in an item template are not set for ‘unapproved’ items (i.e. items requiring a new item request). The item template will need to be applied manually once the item is approved.

---

**Defining Lifecycle Phases**

**Important:** The ability to associate a lifecycle tracking project to an Item and/or Item revision is only available to customers who have licensed Product Lifecycle Management, and it is not available to licensees of Product Information Management Data Librarian.
Defining lifecycle phases

Specify the valid item statuses for each lifecycle phase. You can select a default item status for each phase. When you promote an item to the next phase the default item status is automatically applied to the item. These item statuses set operational characteristics of the item (such as whether the item is Purchasable, Transactable, Stockable, or whether BOMs are allowed). For example, in the Concept and Design phase you may have a default status of Inactive, so you cannot transact the engineering item in inventory or purchasing. However, in the Pre-Production and Production phases you may specify that the default item status is Active so that it is fully transactable in the E-Business Suite.

For more details about associating lifecycles to item catalog categories, see the following topics:

- Associating Lifecycles with an Item Catalog Category
- Managing Lifecycle Change Policies
- Managing Lifecycle Phases
Valid item statuses for lifecycle phase

Phase Details

Display Name: Pre-Production
Description: Production trial runs, validation testing, release to manufacturing
Effective Start Date: 01-Jun-2003
Effective End Date: 

Valid Item Statuses

<table>
<thead>
<tr>
<th>Item Status</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>All functions enabled</td>
<td>✔️</td>
</tr>
<tr>
<td>Engineer</td>
<td>Engineering item</td>
<td>❌</td>
</tr>
</tbody>
</table>

Item Statuses

<table>
<thead>
<tr>
<th>Status Attributes</th>
<th>Value</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOA Allowed</td>
<td>Yes</td>
<td>Default</td>
</tr>
<tr>
<td>Build in WIP</td>
<td>Yes</td>
<td>Default</td>
</tr>
<tr>
<td>Customer Orders Enabled</td>
<td>Yes</td>
<td>Default</td>
</tr>
<tr>
<td>Invoice Orders Enabled</td>
<td>Yes</td>
<td>Default</td>
</tr>
<tr>
<td>Purchaseable</td>
<td>Yes</td>
<td>Default</td>
</tr>
<tr>
<td>Draftable</td>
<td>Yes</td>
<td>Default</td>
</tr>
<tr>
<td>Transactable</td>
<td>Yes</td>
<td>Default</td>
</tr>
</tbody>
</table>

Defining Lifecycles

Important: The ability to associate a lifecycle tracking project to an Item and/or Item revision is only available to customers who have licensed Product Lifecycle Management, and it is not available to licensees of Product Information Management Data Librarian.

You can create lifecycles to manage the lifecycle phases of your items and item revisions. For example, you may have one lifecycle at the item level to manage new part introductions while having another lifecycle to manage the part revision process. You can use different lifecycles to manage manufactured assemblies and outsourced components.

Once the lifecycle phases have been defined, you create a lifecycle in Oracle Projects. To
associate a lifecycle with an item or item revision you must check Enable for Items on the Lifecycle Details page. You must also check Enable for Projects if you use Oracle Projects.

If you intend to associate lifecycle tracking projects with your item lifecycles, then when setting up your project or project template do the following.

- Assign the lifecycle to a work plan for the project or project template
- Assign a lifecycle phase to each top task.

See Also: For more details about defining lifecycles, see the Oracle Projects Users Guide.

**Defining Lifecycles**

**Lifecycle Details**

- Short Name: Computer Component Lifecycle
- Description: Computer component item lifecycle

**See Also:** For more details about defining lifecycles, see the Oracle Projects Users Guide.

**Lifecycles**

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Concept</td>
<td>Evaluate market, define requirements</td>
</tr>
<tr>
<td>2</td>
<td>Design</td>
<td>Design</td>
</tr>
<tr>
<td>3</td>
<td>Prototype</td>
<td>Build and test prototypes</td>
</tr>
<tr>
<td>4</td>
<td>Pre-Production</td>
<td>Production, validation testing, release to manufacturing</td>
</tr>
<tr>
<td>5</td>
<td>Production</td>
<td>Begins immediately with the production output, marking the last phase of the implementation and the beginning of the system support cycle</td>
</tr>
</tbody>
</table>

**Associating Lifecycles to Item Catalog Categories**

**Important:** The ability to associate a lifecycle tracking project to an Item and/or Item revision is only available to customers who have licensed Product Lifecycle Management, and it is not available to licensees of Product Information Management Data Librarian.

Associating a lifecycle with an item catalog category makes the lifecycle available for all items in the item catalog category. Lifecycles are inherited throughout the item catalog category hierarchy—child categories inherit the lifecycle from the parent category. The following figure illustrates the valid lifecycles for the Motherboard item.
Associating lifecycles with item catalog categories

Lifecycles
Item Catalog Category: Motherboards

Select Lifecycle Associations and...

<table>
<thead>
<tr>
<th>Select Name</th>
<th>Description</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑ Computer Component Lifecycle</td>
<td>Computer component item lifecycle</td>
<td>Motherboards</td>
</tr>
<tr>
<td>☑ Computer Component Revision Lifecycle</td>
<td>Computer component item revision lifecycle</td>
<td>Motherboards</td>
</tr>
</tbody>
</table>

Defining Change Policies

Administrators can define and update item change policies for item catalog categories. These item change policies determine the rules for how and when an item's attributes, attachments, and associations are changed. For example, a company manufacturing an engine that has hundreds of specifications may wish to define item change policies for the different phases of development. So when the engine is in the concept or design phase, many of the attributes are allowed to change without formal approval, and the lifecycle policy is not very restrictive. When the engine progresses to the prototype phase, the company might wish to place all attributes, attachments, and associations under stricter change control, perhaps requiring a change order for all modifications. Later, when the engine is in the production phase, the company will likely want every facet of the engine's production under tight control.

Defining item change policies enables a company to:

- Specify whether or not item changes are allowed in a particular lifecycle phase of an item lifecycle
- Specify what types of changes (attributes, attachments, associations) are allowed for an item in each lifecycle phase

For each lifecycle associated with an item catalog category, a Change Policy can be defined as detailed under Defining Change Policy. You can specify for each lifecycle phase of the item whether a change is allowed or not allowed or is possible only through a change order.
• Specify whether or not a change order is required to make particular changes to an item in a specific lifecycle phase

**Note:** Change policies inherited from a parent catalog category cannot be edited at the child catalog category level. If you do not define a change policy for an item, then by default changes are allowed in all lifecycle phases. Item change policies only take effect after the item has been approved.

On the Update Change Policy for Attributes (Associations or Attachments) page, select the appropriate attribute group (association or attachment category) and the change policy you wish to apply for the particular phase. The valid change policies are:

• Allowed

• Not Allowed

• Change Order Required

### Setting lifecycle change policy

**Update Change Policy for Attributes**

Item Catalog Category: Motherboards  Lifecycle: Computer Component Lifecycle

<table>
<thead>
<tr>
<th>Concept</th>
<th>Design</th>
<th>Prototype</th>
<th>Pre-Production</th>
<th>Production</th>
<th>Retirement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prototype</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select an attribute and...</td>
<td>Remove</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Select All** | **Select None**

<table>
<thead>
<tr>
<th>Select Attribute Group</th>
<th>Change Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assembly Clearances</td>
<td>Allowed</td>
</tr>
<tr>
<td>Benchmark Ratings</td>
<td>Allowed</td>
</tr>
<tr>
<td>Operating Conditions</td>
<td>Change Order Required</td>
</tr>
<tr>
<td>Motherboard Specification</td>
<td>Change Order Required</td>
</tr>
<tr>
<td>Memory Configurations</td>
<td>Change Order Required</td>
</tr>
<tr>
<td>Add another row</td>
<td>Allowed</td>
</tr>
</tbody>
</table>

**Note:** Before establishing a change policy for attributes or attachments, you must first associate the attribute group or attachment category with the item catalog category.

### Defining New Item Request Types

New Item request is a workflow process, which allows you to route the definition and approval of a new item. When creating a new item, various aspects of an item like base
attributes, user-defined attributes, AML etc. could be defined by various people in the organization using a workflow process. The definition routing and approval routing could be combined in a new item request workflow.

New Item Request (NIR) is a Change category; there could be many types of New Item Request that suits a particular business process. Each type of NIR allows you to specify NIR number generation function, default assignee type and assignee. You can create a NIR type for a outsourced components which will have a specific numbering prefix and assigned to person with a role at the item category level.

**Defining New Item Request Type**

Once a specific NIR type is created, the administrator can assign additional information like the NIR related attributes to be associated. Examples of such attributes are Cost Impact, Implementation costs etc. These attribute groups then could be organized in different pages similar to the Associating Item Catalog Attributes.

**Defining User defined Attributes for New Item Request Type**

<table>
<thead>
<tr>
<th>Attribute Groups</th>
<th>New Item Request</th>
<th>Standard New Item Request Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Category</td>
<td>Description</td>
<td>Cost Impact</td>
</tr>
<tr>
<td></td>
<td>Date Level</td>
<td>Change</td>
</tr>
<tr>
<td></td>
<td>Pages</td>
<td>Change</td>
</tr>
</tbody>
</table>

Oracle Product Lifecycle Management Implementation Guide
Defining New Item Request Workflows

One of the most important functionality in NIR type, is to be able to associate workflow. The administrator can add workflow steps and specify the status of the NIR in which the workflow will be executed and assign the workflow, by choosing it from a list of workflow template. The administrator should specify the step number, status of the NIR process in which the workflow will be initiated and finally associate the workflow by choosing it from the workflow templates.

Defining a workflow template that could be assigned to a New Item Request is detailed in Defining Workflow Templates section later in the document. NIR workflow are typically Definition and Approval Workflow, an example of NIR workflow is outlined in the following figure.

NIR Workflow example

Once you associate the workflow template with the NIR type, you need to specify the category association for the NIR.

Setting Up New Item Request for an Item Catalog Category

To associate a specific New Item request type with a category, you need to navigate to category details and specify the New Item request type in the New Item Request page.

In the Workflow/Approval section, you can specify which people are required to enter attributes during the workflow process for each lifecycle phase of the new item request.
For example, when the new item request is in the Open phase, one person in the workflow may be responsible for some technical specifications associated with the new item being created; another might need to provide details about the new item's safety standards—in this case you would be sure to associate the attribute groups related to technical specifications and safety standards so that the appropriate people can add the information necessary to define all aspects of the new item.

Since NIR is change, it has a status type. Only the certain workflow type is allowed for a given NIR status type as tabulated below.

<table>
<thead>
<tr>
<th>NIR Status Type</th>
<th>Valid Workflow Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Definition</td>
</tr>
<tr>
<td>Approval/Review</td>
<td>Definition and Approval, or Approval</td>
</tr>
<tr>
<td>Others</td>
<td>Generic</td>
</tr>
</tbody>
</table>

**Associating a New Item Request with an Item Catalog Category**

<table>
<thead>
<tr>
<th>Item Request</th>
<th>Workflow/Approval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item Request Required</td>
<td>Yes</td>
</tr>
<tr>
<td>Item Request Type</td>
<td>New Part Request</td>
</tr>
</tbody>
</table>

**Workflow/Approval**

- Status: Approved
- Name: New Part Definition and Approval
- Description: New Part Definition and Approval
- Start Date: 15 Sep 2014
- End Date: 15 Sep 2014
- Type: Definition and Approval

<table>
<thead>
<tr>
<th>Catalog Category</th>
<th>Attribute Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacities</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workflow Process</th>
<th>Responsibility Required</th>
<th>Type</th>
<th>Name</th>
<th>Days to Respond</th>
<th>Attribute Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition and Approval</td>
<td>One Assignee</td>
<td>Item Role</td>
<td>Commodity Fat Manager</td>
<td>3</td>
<td>Capacitor Specifications, Capacitor Packaging Spec, Material/Specs, Lead (Pb)-Free Profile, Recycle and Reuse, Compliance and Certification</td>
</tr>
<tr>
<td>Definition and Approval</td>
<td>One Assignee</td>
<td>Item Role</td>
<td>CAD Librarian</td>
<td>3</td>
<td>Technology, Layout Specifications, Symbol, Footprint</td>
</tr>
<tr>
<td>Definition and Approval</td>
<td>One Assignee</td>
<td>Item Role</td>
<td>Buyer</td>
<td>3</td>
<td>Ratings, Reliability</td>
</tr>
<tr>
<td>Definition and Approval</td>
<td>One Assignee</td>
<td>Item Role</td>
<td>Commodity Fat Coordinator</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>P1</td>
<td>None</td>
<td>Item Role</td>
<td>Commodity Fat Manager</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Associating Item Attribute Groups with Definition Steps

Defining Item Number and Description Generation Rules

For a given category, item number and description could be automatically generated when an user wants to create an item. Various item number and description generation methods are supported such as:

**Sequence Generated**

If the item request numbering method is sequence generated, then all item number requests will be automatically generated based on the sequence generation rules specified here. You can specify a prefix, starting number, increment, and suffix.

**Note:** Sequence generation does not apply to Item Description generation—it is for item number generation only.

**Function Generated**

If the item request numbering/description method is function generated, then all number/descriptions for new items in the item catalog category will be automatically generated based on a custom function. When specifying that an item number/description is function generated, select the function that you want to use, and then map the function parameters to the user-defined attributes that you want included in the item number/description. When you create a new item, users will be prompted to enter the user-defined attributes used in generating the item number/description.

**Note:** You must have already defined a custom function before using this item number/description generation method.

**Inherited from Parent**

If the item request numbering/description generation method is inherited from parent, then the item number/descriptions for the items will be generated using the same method as the parent item catalog category.
Note: All item catalog category subcategories have, by default, a number/description scheme that is inherited from parent, which could be overridden at the child level.

User entered

If the item request numbering/description method is user entered, then the user is required to manually enter a number or a description for new items.

**Item Number Generation**

**Auto Generation**

Catalog Category: Capacitors

**Number Generation**

<table>
<thead>
<tr>
<th>Item Numbering</th>
<th>Sequence Generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix</td>
<td>VI</td>
</tr>
<tr>
<td>Starting Number</td>
<td>65000</td>
</tr>
<tr>
<td>Increment</td>
<td>1</td>
</tr>
<tr>
<td>Suffix</td>
<td>CAP</td>
</tr>
</tbody>
</table>

**Setting up Item Number Generation**

Catalog Category: Capacitors

To set up automatic change number generation using user-defined functions, follow these steps:

1. Update the header type.
2. Select **Function Generated** from the Number Creation drop-down list.
3. Associate the function.
4. Associate the function parameters to the attributes.
Defining Item Catalog Search Criteria

Criteria templates offer you a way to save frequently used search criteria—essentially, acting as a saved list of specific attributes and attribute values. Criteria templates that you—the administrator—create are available to all users, and serve to expedite searches for an item, thereby saving users the time it would take to specify search criteria and execute frequent item searches.

Of course, users also have the option of creating criteria templates for their own private use. In fact, when both a user and administrator have defined a default criteria template for a particular item catalog category, the user-defined criteria template takes precedence.

Because criteria templates contain user-defined attributes, you must always define them in the context of an item catalog category. You can define as many criteria templates as you wish, and you can also denote the most commonly used as the default criteria template.

The following figure shows the Advanced Search page, where you would define a criteria template for the item catalog category Capacitors. Begin by listing the common attributes a typical user might search for when using the item catalog category Capacitors. Here, attributes like Capacitor Voltage and Capacitor Value, among others, are specified. After adding attributes, you can specify operators and values to further restrict the scope of your search.
Defining Item Catalog Search Criteria

Defining Item Catalog Display Formats

Results Formats enable you to predefine search results views for each item catalog category. You can use these views to look at different sets of attributes of the items that are returned by a search. Both administrators and users can create results formats. Administrator-created results formats are available to all users. User-created results formats are available only to the users who created them. You can define as many results formats as you wish, and you can also denote the most commonly used as the default results format.

You can define a results format by listing the common information a typical user might wish to view for an item. For example, for the item catalog category Capacitor, you might select Item Name, Description, Item Status, or other important item attributes such as Capacitance Value, or Tolerance.

Defining Item Catalog Import Formats

An import format identifies the base and user-defined attributes (but not the operational attributes) in an item catalog category that is imported into the application using a spreadsheet. You cannot import operational attributes; however, you can specify an item template and item status in your import format. Both item templates and item statuses can be used to set operational attributes when importing items.

The base item attributes are:
• Item Name
• Item Catalog Category
• Description
• Primary Unit of Measure
• Lifecycle
• Lifecycle Phase
• Item Status
• Long Description

You can define import formats for both items and item revisions. An import format can also import values for a multi-row attribute group.

**Defining Report Templates Using XML Publisher**

Creating a template file consists of two basic steps:

1. Design your template layout - Use the formatting features of a word processing application and save the file as RTF.

2. Mark up your template layout - Insert the XML Publisher simplified tags.

To create a template, you need to create or use any of the existing seeded data definitions and associate or assign the change template file with the corresponding data definition file. For example, if you have a Change Order template file, associate it to a Change Order data definition file.

When you create a template, you need to assign it a data definition and upload the RTF. There are seven seeded “Summary” templates available in the system. These cannot be updated. Initially, upload one template file for a specific language and territory combination. This file will become the Default Template File. To upload additional template files or to change the Default Template File, use the View Template page.

**To create a template:**

1. Using the XML Publisher Administrator responsibility, navigate to the Templates page.

2. Select the Create Template button

3. Provide the following information:
   
   Name
Enter a user-friendly name for your template.

**Code**

Assign a template code using the product short name and a descriptive ending.

**Application**

While creating templates, select the correct application from the LOV. This application should be the same as the application of the Data definition. For Change Templates, the supported application is "Engineering".

**Data Definition**

Select the product's data definition from the LOV. Oracle Product Lifecycle Management. comes seeded with seven data definitions (Issue, Idea, Change Request, Change Order, New Item Request, Attachment Approval, Attachment Review). For example for an Issue Report Template, select the seeded Data Definition for Issues. This will result in the Template being available for Issue Reports. Type Select the input template type (RTF, PDF, or XSL-FO) from the LOV.

**Start Date**

Enter the date from which the template will be active.

**End Date**

To make the template inactive, enter an end date.

**Subtemplate**

If this is a subtemplate, select the check box.

*Note:* A subtemplate is referenced by other templates, but cannot run on its own

**File**

Use the Browse button to upload your RTF or PDF template layout file.

**Language**

Select the template language from the LOV. Add more language template files to your template definition from the View Template page.

**Territory**

Select the language territory from the LOV.

*Important:* After the template definition is created, the following fields cannot be updated: Application, Code, and Type. You can update the template from the View Template page.

In the template file, you can include simplified XML tags to refer the Item
information you want XML Publisher to print while generating the report.

Here is an example of the XML Input file that will generate report for Motherboard Items using the RTF template.

For further information, see:

- Oracle XML Publisher User’s Guide
- XML Publisher Feature Listing

**Associating Report Templates to Item Catalog Categories:**

Associating a Report Template with an item catalog category makes these templates available for all items in the item catalog category. Report Templates are inherited throughout the item catalog category hierarchy—child categories inherit the Report Templates from the parent category. The following figure illustrates the valid Report Templates for the Motherboard item.

**Note:** For details about how to associate Report Templates, see "Associating Report Templates to an Item Catalog Category" in Oracle Product Lifecycle Management User’s Guide.

**Defining Cross Reference Types**

Item can have many cross-references. Example of cross-references are, customer part number, superseding/preceding item number etc. An administrator can define their own cross-reference types based on their business need. User-defined flex-fields and value sets can be associated with each cross-reference types. Cross-reference types have effectivity dates, which controls them being active or inactive.

### Defining Cross Reference Types

- **Search**: Cross Reference Types

    **Search**
    
    To find your cross-reference type, select a filter in the popup, and enter a word in the text field, then click the "Go" button. To see a list of all the cross-reference types, click the "All" button, and click the "Go" button.
    
    * Search by [Type] [Go]

    **Results**

    | Cross Reference Type | Description             | Inactive Date | Update | Additional Attributes | Delete |
    |----------------------|-------------------------|---------------|--------|-----------------------|--------|
    | Charge Master        | Charge Description Master|               |        |                       |        |
    | Customer             | Customer part number    |               |        |                       |        |
    | Substitute           | Substitute part number  |               |        |                       |        |
    | Vendor               | Vendor part number      |               |        |                       |        |
Defining Catalogs

This chapter covers the following topics:

- Overview of Catalogs
- Defining Catalog Categories
- Defining Catalogs
- Implementing Catalog Security

Overview of Catalogs

Defining catalogs includes the following tasks.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining Catalog Categories</td>
<td>Yes</td>
</tr>
<tr>
<td>Defining Catalogs</td>
<td>Yes</td>
</tr>
<tr>
<td>Implementing Catalog Security</td>
<td>Yes</td>
</tr>
</tbody>
</table>

A catalog is a collection of items classified within a hierarchical taxonomy. A catalog consists of a hierarchy of catalog categories, each containing items assigned to the catalog category. Catalogs provide a convenient way to organize and present your items under various taxonomies. For example, you can set up a Product Catalog of sales items that the Sales and Marketing organizations, as well as your customers, can browse to find products. You can set up a Purchasing Catalog, which contains all the purchasable items that the Engineering, Manufacturing and Procurement organizations can browse to locate parts. You can also set up a Service Catalog that contains all serviceable items and service products for your field repair and customer service departments. The following figure shows the catalog hierarchies for a Product catalog,
Purchasing catalog and Service catalog. You can define any number of catalogs and assign an item to more than one catalog. Depending on your setup, you can also assign an item to more than one catalog category within a catalog.

**Note:** The *category sets* defined in the E-Business Suite appear as *catalogs*. The categories within each of these *category sets* appear as catalog categories in the respective catalog.
**Example of catalog hierarchies**

<table>
<thead>
<tr>
<th>Product Catalog</th>
<th>Purchasing Catalog</th>
<th>Service Catalog</th>
</tr>
</thead>
<tbody>
<tr>
<td>▼ Computer Systems</td>
<td>▼ Electromechanical Components</td>
<td>▼ Computer Parts and Components</td>
</tr>
<tr>
<td>▼ Desktop Products</td>
<td>▼ Audio Devices</td>
<td>▼ Hard Drives</td>
</tr>
<tr>
<td>▼ Laptop Products</td>
<td>▼ Circuit Breakers</td>
<td>▼ Monitors</td>
</tr>
<tr>
<td>▼ Computer Parts and Components</td>
<td>▼ Fans</td>
<td>▼ Holders</td>
</tr>
<tr>
<td>▼ Motherboards</td>
<td>▼ Industrial Control &amp; Sensors</td>
<td>▼ Memory</td>
</tr>
<tr>
<td>▼ PCMCIA</td>
<td>▼ Knobs &amp; Dials</td>
<td>▼ Motherboards</td>
</tr>
<tr>
<td>▼ Keyboards</td>
<td>▼ Power</td>
<td>▼ PCMCIA</td>
</tr>
<tr>
<td>▼ Mouse</td>
<td>▼ Relays &amp; I/O Modules</td>
<td>▼ Keyboards</td>
</tr>
<tr>
<td>▼ Memory</td>
<td>▼ Switches</td>
<td>▼ Mouse</td>
</tr>
<tr>
<td>▼ Hard Drives</td>
<td>▼ Electronic Components</td>
<td></td>
</tr>
<tr>
<td>▼ Monitors</td>
<td>▼ Discretes</td>
<td>▼ Electromechanical Components</td>
</tr>
<tr>
<td>▼ Holders</td>
<td>▼ Diodes</td>
<td>▼ Mechanical Components</td>
</tr>
<tr>
<td>▼ Mounts</td>
<td>▼ Resistors</td>
<td>▼ Networking Components</td>
</tr>
<tr>
<td></td>
<td>▼ Inductors</td>
<td>▼ Ethernet Controllers</td>
</tr>
<tr>
<td></td>
<td>▼ LC Networks</td>
<td>▼ Ethernet Cards</td>
</tr>
<tr>
<td></td>
<td>▼ Resistors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ Signal Transformers</td>
<td>▼ Consulting Services</td>
</tr>
<tr>
<td></td>
<td>▼ Interconnect components</td>
<td>▼ Enterprise Solutions</td>
</tr>
<tr>
<td></td>
<td>▼ Mechanical Components</td>
<td>▼ Small Business Solutions</td>
</tr>
<tr>
<td></td>
<td>▼ Enclosures</td>
<td>▼ Helpdesk Outsourcing</td>
</tr>
<tr>
<td></td>
<td>▼ Fasteners</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▼ Heat Sinks</td>
<td>▼ Support</td>
</tr>
<tr>
<td></td>
<td>▼ Networking Components</td>
<td>▼ Business Support</td>
</tr>
<tr>
<td></td>
<td>▼ Ethernet Controllers</td>
<td>▼ Gold Supported</td>
</tr>
<tr>
<td></td>
<td>▼ Ethernet Cards</td>
<td>▼ Silver Support</td>
</tr>
<tr>
<td></td>
<td>▼ Test, Tools &amp; Supplies</td>
<td>▼ Bronze Support</td>
</tr>
</tbody>
</table>

**Defining Catalog Categories**

You use a catalog category to classify items within a catalog. You can define unlimited catalog categories and reuse catalog categories in more than one catalog. Before creating
catalog categories, consider all the catalogs you need to define throughout your enterprise (for example, Product catalog, Purchasing catalog, Service catalog). Then create a list of all the catalog categories and sub-categories you will need to correctly classify each item within your catalogs. Can you reuse the same catalog category in several catalogs? For example, you may have a catalog category for Hard Drives in each of your catalogs since you sell hard drives to customers, purchase hard drives from your supplier, and service hard drives. You should use the same flex structure for catalog categories so you can reuse them in different catalogs.

**Defining Catalogs**

A *catalog* is a hierarchy of catalog categories used to classify items defined in your item catalog. Once you have defined all of your catalog categories you can create each of your catalogs. When you construct your catalog category hierarchy keep in mind that only the leaf nodes of the catalog can contain items.

You can set up catalogs in a way that the items may be classified the same across all organizations or differently between organizations. You can enable item assignments to more than one catalog category within your catalog.

**Related Topics**


Assigning a Category to a Catalog, *Oracle Product Lifecycle Management User’s Guide* or *Oracle Product Information Management Distribution Librarian User’s Guide*

Creating a Catalog Category within a Catalog, *Oracle Product Lifecycle Management User’s Guide* or *Oracle Product Information Management Distribution Librarian User’s Guide*

**Implementing Catalog Security**

You can use role-based security to control which users, groups, companies, or organizations can view and manage catalog items. For example, you may allow customers and the Sales organization to view your Product catalog, while only Engineering, Procurement, and Manufacturing can view the Purchasing catalog.

There are two privileges that you can grant a user with a role on the catalog: View Catalog and Manage Catalog. The two seeded roles are Catalog Viewer and Catalog Manager. Users who have been granted the View Catalog privilege can browse and view all items in the catalog. A user granted the Manage Catalog privilege can assign items to a catalog category within the catalog.

<table>
<thead>
<tr>
<th>Role</th>
<th>View Catalog Privilege</th>
<th>Manage Catalog Privilege</th>
</tr>
</thead>
</table>

4-4 Oracle Product Lifecycle Management Implementation Guide
### Related Topics

Assigning People to a Catalog, *Oracle Product Lifecycle Management User’s Guide* or *Oracle Product Information Management Distribution Librarian User’s Guide*

Assigning People to a Valid Category, *Oracle Product Lifecycle Management User’s Guide* or *Oracle Product Information Management Distribution Librarian User’s Guide*
This chapter covers the following topics:

- Overview of Change Categories and Types
- Build the Change Management Text Index
- Defining Change Categories
- Creating Priority Codes
- Creating Reason Codes
- Creating Classification Codes
- Creating Statuses
- Defining Workflow Templates
- Attaching Workflow Templates
- Customizing Change Management Workflows
- Creating Task Templates
- Defining Header/Line Type Attributes and Attribute Groups
- Defining Header Types
- Defining Line Types
- Associating Change Type Attributes
- Associating Change Line Type Attributes
- Setting Up Change Attribute Group Security
- Implementing User Defined Functions for Change Type Attribute
- Defining Change Category Search Criteria
- Defining Change Category Display Formats
- Defining Change Category Reports
Overview of Change Categories and Types

**Important:** The seeded Change Management categories of Idea and Issue are available only to customers who have licensed Product Lifecycle Management, and they are not available to licensees of Product Information Management Data Librarian.

You must perform the following tasks to define Change categories and their associated types.

<table>
<thead>
<tr>
<th>Task</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining Change Categories</td>
<td></td>
</tr>
<tr>
<td>Defining Change Types</td>
<td>Yes</td>
</tr>
<tr>
<td>Defining Line Types</td>
<td></td>
</tr>
<tr>
<td>Defining Change Type Attribute Groups</td>
<td></td>
</tr>
<tr>
<td>Defining Change Line Attribute Groups</td>
<td></td>
</tr>
<tr>
<td>Setting Up Change Attribute Group Security</td>
<td></td>
</tr>
<tr>
<td>Implementing User-Defined Functions for Change Type Attributes</td>
<td></td>
</tr>
<tr>
<td>Associating Change Type Attributes</td>
<td></td>
</tr>
<tr>
<td>Customizing Change Management Workflows</td>
<td></td>
</tr>
<tr>
<td>Defining Approval Routing Templates</td>
<td>Yes</td>
</tr>
<tr>
<td>Creating User-Defined Priority Codes</td>
<td>Yes</td>
</tr>
<tr>
<td>Creating User-Defined Reason Codes</td>
<td></td>
</tr>
<tr>
<td>Creating User-Defined Statuses</td>
<td></td>
</tr>
<tr>
<td>Defining Change Category Criteria Templates</td>
<td></td>
</tr>
</tbody>
</table>
Build the Change Management Text Index

To take advantage of the Oracle Text search features such as keyword search, stemming search, and fuzzy search—run the concurrent program Item Catalog Text Index Build. Any user with the Development Manager responsibility can submit this concurrent program from the Submit Request window. Select the concurrent request parameter Action=CREATE when submitting the request.

Related Topics

Optimizing the Change Management Text Index, Oracle Product Lifecycle Management User’s Guide

Defining Change Categories

You can use change categories to define and manage changes required by your enterprise. In addition to the seeded change categories (issues, change requests, change orders) you can create change categories specific to your business needs. For example, you can create the change category Enhancement Requests to track your customers’ product enhancement requests. You cannot delete seeded change categories; however, you can disable them.

You can configure each change category to have revised items or request lines based on the business purpose of the category. For example, a Change Order can have revised items as they allow a change order to implement item related changes. Request lines enable you to request changes or specify tasks related to an item and assign it to a person or group. You can associate criteria templates and result formats to a change category for frequently executed search criteria.

The following change categories are seeded:

**Issues** To track, manage and resolve various product/process related issues

**Change Requests** To request changes and get approval for requested changes

**Change Orders** To implement requested changes and revise items

**Ideas** To capture suggestions, innovations, improvements etc. from customers and internal users

**New Item Request** To provide a formal process of definition and approval of a new item
**File Review** To allow for an informal process of review and feedback on attachments

**File Approval** To allow for a formal process of review and approval of attachments

You can specify a form function to control who can view and create change category objects. The form function must be added to the change management security menu (EGO_CHGMGMT_USER_SECURITY), which in turn is referenced by a user responsibility (such as Development Manager or Development Engineer). You can also enable (in the Item Catalog workbench) a tab to show all instances of a change category for the item in context.

---

**New change category Enhancement Requests enabled in the Item Catalog**

<table>
<thead>
<tr>
<th>Item Enhance Request Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>Linux OS should be pre-installed on the computer</td>
</tr>
<tr>
<td>Feature</td>
<td>The cost of the standard desktop configuration can be reduced if we pre-install Linux</td>
</tr>
<tr>
<td>Feature</td>
<td>Desktop with Intel’s AGP chipset</td>
</tr>
<tr>
<td>Feature</td>
<td>Increase data transfer speed to support data parallel computing</td>
</tr>
</tbody>
</table>

**Enabling Change Category Function Security and the Item Change Category Tab**

To enable the change category function and the item change category tab:

1. Create the Form Function for the Item Enhancement Request tab.

   **Description Tab**

   **Function**

   EGO_ITEM_ENH_REQ

   **User Function Name**

   EGO User Item Enh Request tab

   **Description**

   EGO User Item Enh Request tab

   **Properties Tab**

   **Type**

   SSWA JSP FUNCTION

   **Maintenance Mode Support**
None

**Context Dependence**

Responsibility

**Web HTML Tab**

**HTML Call**

OA.jsp?page=/oracle/apps/ego/item/eu/webui/EGOTEMCHANGEMGMTLISTPGL=431=RP=EGO_USER_WORKBENCH_HOMEPAGE=EGO_ITEM_ENH_REQ=ENHANCEMENT_REQUEST

**Important:** Boldfaced characters represent the form function and internal name of the change category, respectively.

2. Create the Form Function for Enhancement Request security. The function name must be 'ENG_CREATE_' (for example, ENG_CREATE_ENHANCEMENT_REQUEST):

**Description Tab**

**Function**

ENG_CREATE_ENHANCEMENT_REQUEST

**User Function Name**

Engineering Create Enhancement Request

**Properties Tab**

**Type**

SSWA JSP FUNCTION

**Maintenance Mode Support**

None

**Context Dependence**

Responsibility

**Web HTML Tab**

**HTML Call**

OA.jsp?page=/oracle/apps/eng/changemgmt/webui/SelectChgTypePG=ENHANCEMENT_REQUEST

**Note:** Boldfaced characters represent the internal name of the change category.
3. Add the Create Enhancement Request function to the change management navigation menu (see the menu EGO_CHANGE_MGMT_MENU).

   **Prompt**
   Create Enhancement Request

   **Function**
   Engineering Create Enhancement Request

4. Add the Enhancement Request security function to the change management security menu (see the menu EGO_CHGMGMT_USER_SECURITY).

   **Function**
   Engineering Create Enhancement Request
5. Add the Item Enhancement Request function to the item change management tab menu (query the menu EGO_USER_CHANGE_MANAGEMENT_TAB)

**Prompt**
Enhancement Requests

**Function**
EGO User Item Enh Request tab

*Important:* You must enter a prompt when adding a function to a change management menu such as EGO_USER_CHANGE_MANAGEMENT_TAB.

6. Add the new change category for Enhancement Requests by duplicating an existing Change Category. In example shown in the new Enhancement Request Change Category is being defined by duplicating the Issue category:

**Internal Name**
ENHANCEMENT_REQUEST

**Name**
Enhancement Request

**Description**
Enhancement Request

**Plural Name**
Enhancement Requests

**Sort Seq**
9
Start Date
defaults to sysdate

Number Generation
Sequence Generated
Prefix
ER-
Next Available Number
0001

Add new change category by duplicating an existing change category

Note: Oracle recommends that you stop and then start the Jserv and Apache listener middle tier ports after completing these setup tasks.

Related Topics

Creating Priority Codes
Priority codes enable you to track the urgency of an issue/change. You can create priority codes to capture different priorities (such as High, Medium or Low).
Priority codes are applicable to all change categories and their types.
You cannot delete seeded priority codes; however, you can disable the seeded priority codes and define new ones specific to your use. You can disable a priority on a given date by specifying a date in the Inactive On field.

Creating Reason Codes

Reason codes enable you to track the reason for which the issue/change has been created. Create reason codes to capture reasons for the issue/change (such as Quality Improvement, Design Improvement, Cost Reduction, Test Failure and Non Conformance).

Reason codes are applicable to all change categories and their types.

You cannot delete seeded reason codes; however, you can disable the seeded reason codes and define new ones specific to your use. You can disable a reason on a given date by specifying a date in the Inactive On field.

Creating Classification Codes

Classifications provide a mechanism for companies to automate the categorization of change orders, and to also indicate to users exactly how the change order will impact their production. Oracle provides two types of classifications:

Derived

Derived classification codes are derived from a user-defined function. For example, a division of a company, Vision Operations, needs to automate the process whereby a change order is assigned a particular classification code. To create an automated classification process, Vision has created a set of attributes that, when filled in by users, classifies change orders into a particular classification or workflow routing. The user-supplied attributes are mapped to user-defined functions. The function takes the data supplied in the attributes and derives a valid classification code. Derived classification codes appear to users as read-only data. For details about user-defined attributes and functions see Defining Header/Line Type Attributes and Attribute Groups, and Defining User-Defined Functions. For details about setting up user-defined functions for change type attributes, see the example: Implement User-Defined Functions for Change Type Attributes.

Valid

Valid classification codes are selected by the user from a list of values. Valid values are specified in the change header type. Note: Classifications are available only to change categories whose base category is Change Order.

Creating Statuses

Statuses enable you to manage an issue/change through its lifecycle. You define statuses to indicate various states of an issue/change (for example, Open, On Hold, Complete
and Cancelled).

Statuses are applicable to all change categories and their types

You cannot delete or disable seeded statuses; however, you can define new statuses specific to your business processes. You can disable user-defined statuses on a given date by specifying a date in the Inactive On field.

**Defining Workflow Templates**

Workflow templates enable you to predefine a business approval process. Change categories such as issues, change requests and change orders can only be approved via the successful completion of an approval routing. You can create and maintain workflow templates for each change type using workflow templates listed in the Setup Workbench under the Change Management Workflow tab.

**Creation of a new Workflow Template**

![Image of workflow template creation interface]

While creating a workflow template its Type needs to be specified.

**Important:** Once a workflow template is created of a particular type, its type cannot be changed.

Currently the following workflow template types are supported:

**Approval**

The Approval workflow template type is valid only for workflows with status type Approval.

**Definition**

The Definition workflow template type is used primarily for workflows in New Item Requests with status type Open. Definition and Approval workflow template type is used primarily for New Item Requests of status type
Defining Change Categories and Types

Approval.

Generic

The Generic workflow template type is used for all other status types.

Certain workflow types can be associated with specific statuses. Following are the associations between status type and workflow types:

### Valid Workflow Types for Change Categories (except New Item Request)

<table>
<thead>
<tr>
<th>Status Type</th>
<th>Valid Workflow Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval/Review</td>
<td>Approval</td>
</tr>
<tr>
<td>Others</td>
<td>Generic</td>
</tr>
</tbody>
</table>

### Valid Workflow Types for New Item Requests

<table>
<thead>
<tr>
<th>Status Type</th>
<th>Valid Workflow Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Definition</td>
</tr>
<tr>
<td>Approval/Review, Definition and Approval, or Approval</td>
<td>Approval</td>
</tr>
<tr>
<td>Others</td>
<td>Generic</td>
</tr>
</tbody>
</table>

Workflow templates are basically made up of approval steps; each approval step describes a workflow process and specifies the assignees. For example, you can create steps to request approval, request comment, or send an FYI notification.

You can use any of the following seeded workflow processes in a workflow template:

**Request Approval**

The Request Approval workflow enables you to request approvals from a person or group.

**FYI**

The FYI workflow enables you to send an FYI notification to a person or group.

**Request Comment**

The Request Comment workflow enables you to request comments from a person or group.
Definition
The Definition workflow is primarily used in the New Item Request Process and allows for the association of item attribute groups in the New Item Request process so that they can be defined by the step assignee.

Definition and Approval
The Definition and Approval workflow is similar to the Definition workflow but also requires an approval by the step assignee. For more details please refer to Defining New Item Request section.

You should plan and document the approval processes needed for specific types of changes in your enterprise. Pre-planning of these processes enable you to define workflow templates that adhere to your business processes.

You can manage parallel and serial approvals by assigning multiple people or groups to a routing step, or by assigning individual people or a group to each routing step. You can assign a specific role, person, or group to a routing step. Item roles need to be mapped to the change roles assigned on an approval routing step to ensure a person or group is assigned when the change is created (see "Defining Roles" for details on mapping item roles to change roles).

Attaching Workflow Templates
In the Setup Workbench you can associate multiple workflow templates with a given change type. You associate multiple workflow templates with a workflow status at the header level of a issue, change request, or change order. You can also add ad-hoc step assignees to the workflow based on role, group, and person filters. This enables a user to abort the workflow within a status, and to choose a different template for that status.

Once these associations are made, when creating an individual change object within that type, you can select various workflow options. This enables you to select a specialized workflow to reach those people who need to approve specific changes different from those who would approve changes.

You can set up workflow templates to use when requesting approval or review of an issue, change request, or change order. This enables reviewers and approvers to select the template with the steps that best meets their needs and specific people, based on role, group, and person filters. You can have multiple templates at any status.

To change templates before moving to Approval status:
1. Navigate to the Change Simple Search page.
2. Search for a Change Request record using the query ECR%.
3. Select Number for a specific change request.
4. In the Change Request Summary page, select Change Request Number.
5. Select the Workflow tabbed region, and select the Approval status.

6. In the Status Approval region, select Switch Template

7. In the Workflow field, select a different template using the pulldown list. The workflow steps for the selected template are displayed.

8. Select Apply to save your work.

   The selected workflow template is ready for use on the Approval Status for this particular Change Object. Restart the Workflow and the Change Object flows through the steps for this template for the Approval Status.

You have the option to designate new assignees based on the new workflow template.

**To change templates when the Approval status is in process:**

1. Navigate to the Abort Workflow page.

2. Select Reset Workflow.

3. Select Switch Template.

4. In the Workflow field, select a different template using the pulldown list. The workflow steps for the selected template are shown below.

5. Select Apply.


7. Optionally, select Refresh Assignees to designate new assignees before starting the new workflow.

**Customizing Change Management Workflows**

Oracle supports the extension of workflows. Extensions include using existing seeded processes to build new workflow processes and modifying parameters of an activity without changing process logic (for example, adding custom logic in an Abort Approval Routing subprocess).

If the Change Management seeded workflows do not meet your business processing needs, you can customize or extend the seeded Change Management Workflow processes by creating your own processes, modifying the seeded ones, adding a new activity to a seeded process, and/or modifying custom hook PL/SQL procedures.
Suggestions for Customizing Workflows

- If you decide to create a new workflow process for Priority Change rather than modifying the seeded workflow, you must write your own custom logic to start your custom workflow in the custom hook called in Start Workflow.

- If you add a new workflow process in the Change Approval Routing Step (ENGCSTEP), then the process will be enabled as an Approval Routing Step Workflow in the Approval Routing pages. The workflow process is automatically called from the Approval Routing workflow.

- The activity will be shared by more than one object’s process, so create the activity in the ENG: Standard (ENGWFSTD) item type. Otherwise, create the activity in the appropriate object’s item type.

- Instead of modifying workflow notifications, you should create new messages. Oracle Workflow Builder does not maintain version information for objects such as item types, item type attributes, messages, and lookup types. For these objects, the latest definition always applies, so you must consider whether a change to any of these objects is backward compatible. If the modification affects existing processes, you should create a new object rather than edit the existing object.

  **Important:** Oracle provides support only for its seeded activities, processes, and the types of extensions described in this guide. Oracle does not provide support for your custom activities and processes.

Creating Task Templates

Task templates define the various tasks that must be completed for a change order. You can create task templates only for categories whose base change category is Change Order.
Creation of a Task Template

The task templates are created for specific Organizations. Hence multiple task templates could be associated with a change order type. Depending on the organization in which the change order is created, the task templates defined for that organization associated with the change order become applicable.

If a task is specified as Mandatory it means that the task must be completed before the change order can proceed to the next status. After creating task templates, you can use them to define the organization policies for change order header types. This has been detailed in the section Defining Header Types.

Defining Header/Line Type Attributes and Attribute Groups

Defining attributes and attributes groups enables you to capture additional information related to issues, change requests, and change orders. You can create user-defined attributes with validation logic and associate them to change types as a collection of attributes within an attribute group.

Attributes are defined by their names and values, and are saved within attribute groups. You can associate attribute groups to a change header type or change line type. You must define separate attribute groups for header types and line types. You can reuse the same attribute group across different change categories and their change types. Users enter the values for the attributes on the pages that you create for each change type. You can also define how attributes are displayed for your change objects to improve usability.

You can index attributes to speed up search performance. For numeric or date data types, a B-tree index enables users to search on a range of values or use relational operators such as "less than" and "greater than," among others. For text attributes, Oracle Text index allows flexible key word searching.
Prior to creating user-defined attributes, do the following:

- Group related attributes within the same attribute group. The following table shows some examples of attribute groupings:

<table>
<thead>
<tr>
<th>Attribute Group</th>
<th>Attributes</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Priorities</td>
<td>Customer</td>
<td>Char</td>
</tr>
<tr>
<td></td>
<td>Priority</td>
<td>Char</td>
</tr>
<tr>
<td></td>
<td>Date</td>
<td>Standard Date</td>
</tr>
<tr>
<td></td>
<td>Review</td>
<td>Char</td>
</tr>
<tr>
<td>Implementation Cost</td>
<td>Manufacturing</td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>Engineering</td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>Supplier Charges etc</td>
<td>Number</td>
</tr>
</tbody>
</table>

- Create an object role (change) with custom privileges to define attribute group security (if necessary). You can secure the attribute group by setting Edit/View privileges. Later on, only users with certain roles can view or edit those attributes.

- Determine the data type of the attributes (for example, number, char, date).
  
  **Important:** You cannot edit the data type after you save the attribute.

- Set up the Unit of Measure Classes (for example, Currency) and Units of Measure (for example, US Dollars).

- Establish the validation rules for each attribute and create the corresponding value sets. For more details, see Defining Value Sets for User-Defined Attributes.

- Decide how you want the attribute group to be displayed:
  - Single row Displays a Text field with a value.
  - Multi-row Displays multiple values (rows) for the same attributes (column) in a table.
Important: Define which attributes or combination of attributes will maintain uniqueness of records in cases where the attribute group is displayed as multi-row.

The figure below shows how the display options are set for the different attributes. The figure on Defining Change Attribute Types shows the results of these display option settings. For example, you can see that the attribute Customer is set to display as a text field, and indeed it does. Note that the attribute Customer uses the value set Customer, which is really a table that is displayed as a list of values (LOV) text field. Also notice that the attribute Customer is maintained as part of a unique key. You can add to/edit the unique key as long as doing so does not destroy the uniqueness (creating duplicates) of existing records.

Also note in the figure below that Enabled parameter is set to No for the Company Name attribute, indicating that this attribute will not be displayed on the change page. You can always disable attributes; however, you cannot delete attributes if the attribute group has already been associated with a change type.

Defining change attributes and attribute groups

<table>
<thead>
<tr>
<th>Attribute Group Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Name</td>
</tr>
<tr>
<td>Display Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Data Security</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Maintain unique key</th>
<th>Add Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Object</td>
<td>Delete</td>
<td></td>
</tr>
<tr>
<td>Select All</td>
<td>Select None</td>
<td></td>
</tr>
<tr>
<td>Select Sequence</td>
<td>Internal Name</td>
<td>Display Name</td>
</tr>
<tr>
<td></td>
<td>Customer</td>
<td>cust</td>
</tr>
<tr>
<td></td>
<td>Plenty</td>
<td>plenty</td>
</tr>
</tbody>
</table>

| Where Used | |
| Name | Data Level | Enabled | Display Pages |
| Feature ER | Change | Yes | Prints |

Defining change type attributes (setting attribute values)

Customer Priorities

- Indicates required field
- Priority: High, Medium, Low
- Date: (MM/DD/YYYY)
Related Topics
Defining Item Attributes and Attribute Group, Oracle Product Lifecycle Management User’s Guide or Oracle Product Information Management Distribution Librarian User’s Guide

Defining Header Types

Change types provide enterprises a way to accommodate business processes and classify types of changes within a change category. For example, you can define different issue types such as Quality Issue, Product Issue, and Performance Issue to capture the various types of issues in your enterprise.

You can configure a change type to have the following:

- **Auto Tracking Numbers**
  You can specify a prefix and a next available number of a sequence. For example, an Engineering Design Change type can be configured to have the prefix DSGN.

- **Default Assignment Role**
  You can specify an item role (such as Design Engineer) as the default assignment role for a specific person or a group. For example, if Design Engineer is selected, then the person or group who has the Design Engineer role on the subject item for which the change is created is assigned to the change. Note that the change is only assigned to one person or group, thereby ensuring that it is assigned only to the person or group for which it was intended. If more than one person must have the specified role, it is recommended that you create a group and place all users with the same role into the group, and then give that group the default assignment. If more than one person has the Design Engineer role, then the change would be randomly assigned to one of those users.

- **Attribute Groups/Pages**
  You can associate attribute groups that have been defined for headers to a change type. These attribute groups enable you to capture additional information regarding the type of change being created, or business process-specific attributes required to process the change. You can create pages for a change type and then organize the attribute groups associated with the change type in a logical manner.

- **Setting up valid codes**
  You can specify the valid codes for Priority, Reason and Classifications for the Change type. (Classification codes are only available change types based on the Change Order category.) This allows you to limit the available values for a user to only the selected values for each of these codes. The creation of these values for the different codes was detailed earlier in this chapter.
• **Configuring Primary Attribute groups and sections**

Depending on your business process, you may elect to enable only certain primary attributes and sections for a change type. For example a Marketing Issue, you can elect not to enable the Project Name and Project Task Name attributes, as these would most likely be not used. Similarly for a Supplier Requested Issue, you can disable the People section, as you may not want the Supplier to have a visibility on who has what role in your organization.

• **Workflow**

You can define the various statuses for the change type under the workflow tab. Associated with each status you can specify the valid statues for promotion and demotion as well associate a workflow with each status. For an Approval status only an Approval type workflow can be selected. Also if desired you can enable Digital Signatures for the Approval. If a workflow is associated with a status, you also have the option to enable Auto-promote and Auto-demote. This will allow the change to be automatically promoted to the selected status on the successful completion of the workflow or the automatic demotion to the selected status on the rejection of a workflow.

• **Organizational Policies and Task Templates for Change Orders**

For change orders only, you can specify task templates and propagation rules by organization. You can group a set of change tasks (defined in a task template) to be performed during or between certain statuses of the change type’s workflow. You can also specify whether or not all or some of these tasks are mandatory.
### Associating Task Templates per Organization

#### Organization Policies

<table>
<thead>
<tr>
<th>Change Category</th>
<th>Change Order</th>
<th>Type</th>
<th>ECO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Organization

- Indicates required field

<table>
<thead>
<tr>
<th>Select Organization</th>
<th>Organization Name</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VI</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Seattle Manufacturing</td>
<td></td>
</tr>
</tbody>
</table>

Add Another Row

#### Propagation Rules

<table>
<thead>
<tr>
<th>Task Template</th>
<th>Start After Status</th>
<th>Complete Before Status</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO Analysis</td>
<td></td>
<td>Approval</td>
<td></td>
</tr>
<tr>
<td>CO Release</td>
<td>Approval</td>
<td>Released</td>
<td></td>
</tr>
<tr>
<td>CO Schedule Global</td>
<td>Approval</td>
<td>Scheduled</td>
<td></td>
</tr>
<tr>
<td>CO Implementation Global</td>
<td>Released</td>
<td>Implemented</td>
<td></td>
</tr>
</tbody>
</table>

Add Another Row

- **Propagation Rules for Change Orders**

Besides associating Task templates for each organization, you can specify the Propagation rules for propagating change orders of this type to other organizations. A propagation rule enables you to specify an organizational hierarchy to propagate the change order, including all associated revised items and tasks. The propagation rule can be executed automatically when the change order enters a specific status.
Defining Line Types

Change line types enable you to capture specific changes to an item or tasks related to a change. For example, you can define different line types to capture item related changes (such as Item Attribute changes, Component redesign changes, Attachment changes and Part Obsolescence).
Creation of a Line type is similar to that of the Change Header type. Each Line type could have a default assignee as well as an associated Subject.

### Defining a Change Line Type

**Create Change Request Line Type**

- **Name**
- **Description**
- **Start Date**
- **End Date**

**Subject Selection Criteria**

- **Subject**
- **Attribute**
- **Value**
- **Allowed**

**Default Assigned To**

- **Assignee**

### Associating Change Type Attributes

After creating the attributes, values, value sets, and attribute groups, associate the attribute group to a change type or a line type. You can define pages to display the associated attribute groups.

**Note:** You can reuse the same attribute group across change types belonging to
To associate attribute groups with a change type:

1. Add the attribute group to the Attribute Groups list for that change type (change order or issue).

For example, the change category Change Order has the change type ECO. The following figure shows that the attribute groups Implementation Cost and Unit Cost are associated with the header type ECO.

2. Define change pages for the associated attribute groups. You can locate one or more attribute groups on a page, or create separate pages for each attribute group. The following figure shows the change order type ECO has two pages: Classification and Impact Analysis. The figure "Create pages for attributes" shows an example of how two attribute groups are rendered on a single page for a change order.
Associating Change Line Type Attributes

You can configure a line type to have Attribute Groups/Pages. You can associate attribute groups that have been defined for lines to a change type. These attribute groups enable you to capture additional information regarding the type of change being created, or business process-specific attributes that are required to process the change. You can create pages for a line type, and then organize the attribute groups associated with the line type in a logical manner.

Attribute Groups for a Change Line Type

Setting Up Change Attribute Group Security

When implementing role-based change management security you can set up privileges to control the view and edit permissions for specific change management attribute groups. You can control which users can view and/or edit certain attribute groups for a change object by assigning a role granting those specific privileges. By default, a change role's View Basic Change Information and Edit/Delete Change privileges control whether you can view or edit attributes that are not controlled specifically at the attribute group level. In other words, when implementing change management security you do not have to specify a view or edit privilege for each attribute group.

Suppose your company is co-designing with your supplier a new motherboard for its next generation of desktop computers. To improve communication with your supplier...
on design changes to the motherboard you would like to securely share change request information externally with your suppliers and contract manufacturers. The Supplier Engineer should only be able to view specific attribute groups.

**To implement attribute group security for change management:**

1. Select the Application Developer responsibility, navigate to the Form Functions form, and create Form Functions for each privilege that controls view and edit permissions for the attribute groups. Specify "Change" or "Change Line" in the Object field for each Form Function.

2. Select the Development Manager responsibility and navigate to the Setup Workbench. On the **Attribute Group Details** page for each attribute group, specify the View Privilege and Edit Privilege in the Data Security section.

3. On the **Change Role Detail** page, grant the privileges that you defined as Form Functions in the previous steps.

For more details about setting up attribute group security for Change Management, see and the following topics in the Oracle Product Lifecycle Management User's Guide or Oracle Product Information Management Distribution Librarian User's Guide:

- Setting Up Item Attribute Group Security
- Creating Custom Privileges
- User Defined Attributes
- Managing Roles

**Implementing User Defined Functions for Change Type Attribute**

You can define your own custom logic by adding user-defined functions and actions to existing change pages. In such cases you do not need to customize the entire page.

You can register user-defined functions for executing customer-specific business rules and calculations. These functions can be written in Java or PL/SQL. URL functions can also pass the values of certain parameters to the URL string and redirect users to a secure page. For each function, you need to register the list of parameters, their data types, and how they get their values by specifying parameter types.

Actions are trigger points for functions and can be displayed as buttons or links. You can determine the conditional visibility of the button/links and also prompt the user based on his or her input.

Example: Implement User-Defined Functions for Change Type Attributes

The following example shows how you would calculate Total Cost by using the
attribute group Implementation Cost.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Implement Cost</td>
<td>Implement Cost</td>
<td>Calculate Cost</td>
<td>Calculate Cost</td>
<td>Calculate Cost</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Attribute Name</th>
<th>Data Type Mapping</th>
<th>Attribute &amp; Parameters</th>
<th>Parameter Name</th>
<th>Data Type Parameter Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Suppliers Charges</td>
<td>Number</td>
<td>---à ---à ---à</td>
<td>Cost 1</td>
<td>Number</td>
</tr>
<tr>
<td>2</td>
<td>Manufacturing</td>
<td>Number</td>
<td>---à ---à ---à</td>
<td>Cost 2</td>
<td>Number</td>
</tr>
<tr>
<td>3</td>
<td>Engineering</td>
<td>Number</td>
<td>---à ---à ---à</td>
<td>Cost 3</td>
<td>Number</td>
</tr>
<tr>
<td>4</td>
<td>Field/Repair</td>
<td>Number</td>
<td>---à ---à ---à</td>
<td>Cost 4</td>
<td>Number</td>
</tr>
<tr>
<td>5</td>
<td>Total Implement Cost</td>
<td>Number</td>
<td>---à ---à ---à</td>
<td>Cost Result</td>
<td>Number</td>
</tr>
</tbody>
</table>

First, register the PL/SQL function "Calculate Cost" (based on the PL/SQL procedure that resides in the PL/SQL package) with all required parameters.

Then navigate to the "ECO" change order type and add the action to the attribute group "Implement Cost" by clicking Update Action. The action executes the function on the change pages. After creating the action, map the function parameters to the corresponding attributes.

The mapping section on the Action Detail page provides the mapping information for function parameters. You can also map parameters of functions to the object’s primary key value (for example, CHANGE_ID for change objects).

**When setting up an action:**

1. Specify a role-based privilege to secure the action.

2. Add the Dynamic Visibility Function (written in Java only) to your action if there is a certain condition that needs to be satisfied before users can view the action button or link. For example, if one or all of the fields are empty, then the user cannot see
3. Add the Dynamic Prompt Function (written in Java only) to your action if the label for the button/link needs to be changed depending on a certain condition. For example, if the Total Implement Cost attribute is empty, then the button label is Apply, otherwise the button label is Apply Changes.

**Important:** Dynamic Prompt and Dynamic Visibility function parameters must be mapped to the corresponding attributes as well.

---

**Associate action with attribute group to execute a function**

**Action Details for Calculate Cost**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Mapped To</th>
<th>Mapped Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost 1</td>
<td>Attribute Group</td>
<td>Suppliers Charges</td>
</tr>
<tr>
<td>Cost 2</td>
<td>Attribute Group</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>Cost 3</td>
<td>Attribute Group</td>
<td>Engineering</td>
</tr>
<tr>
<td>Cost 4</td>
<td>Attribute Group</td>
<td>Field/Repair</td>
</tr>
<tr>
<td>Cost Field</td>
<td>Attribute Group</td>
<td>Total Implement Cost</td>
</tr>
</tbody>
</table>

**Change type attribute page displays action button**

**Implementation Costs**

- **Indicates required:**
  - **Material:**
    - Cost relates to material or repair work
  - **Manufacturing:**
    - Cost relates to manufacturing labor, tooling, equipment, facilities
  - **Field/Repair:**
    - Cost relates to field service and crew, labor, and support
  - **Total Implement Cost:**
    - Cost relates to supplier revision, engineering, manufacturing, and support

**Related Topics**

Defining Change Category Search Criteria

Criteria templates offer you a way to save frequently used search criteria—essentially, acting as a saved list of specific attributes and attribute values. Criteria templates that you—the administrator—create are available to all users, and serve to expedite searches for issues, change requests and change orders, thereby saving users the time it would take to specify search criteria and execute frequent change management searches.

Of course, users also have the option of creating criteria templates for their own private use. In fact, when both a user and administrator define a default criteria template for a particular change category, the user-defined criteria template takes precedence.

Because criteria templates contain user-defined attributes, you must always define them in the context of a change category. You can define as many criteria templates as you wish, and you can also denote the most commonly used as the default criteria template. For example, you may define criteria templates to quickly find Open Change Requests, My Issues, or Scheduled Change Orders.

Defining Change Category Display Formats

Results formats enable you to predefine search results views for each change category. You can use these views to look at different sets of attributes of the change objects (for example, issues, change requests, change orders) that are returned by a search. Both administrators and users can create results formats. Administrator-created results formats are available to all users. User-created result formats are available only to the users who created them. You can define as many results formats as you wish, and you can also denote the most commonly used as the default results format.

You can include any base or user-defined attributes in your results format definitions. You can include display sections in your results formats that provide links directly to the section from your search results. This enables you to display links in your search results to any user-defined change type page (such as Cost Information) or any of the standard display sections: Lines, Attachments, Action Log, Approval, Approval History, Dependencies, Revisions, and People.
Defining Change Category Reports

Change management reports are basically search criteria and display formats that you can save, browse, email, or print. You can create reports for any change management category. You can also browse a report sequentially or using a summary view.

Browsing a report sequentially enables you to step through each change object in a report. A summary view displays the report in tabular column format. You can send a report to other users including registered suppliers and customers.

An Admin can create a report for the most common or frequent searches that users of the system would be performing. This would be based on the business process being implemented. These Admin defined reports would save an ordinary user the time of selecting a change category, search criteria and display format, and repeating the same search over and over again. You can give reports meaningful names as well. For example, a search for all open high priority issues may yield dozens of issues, each identified by issue name and number. You can name the report for these search results “High Priority Open Issues Report” as shown in the figure below.

Report security is consistent with search security: you can only access change object on which you have the required role. Also, administrator-created reports are available to all users, although only the administrator can edit these reports.

You can associate workflows at the line level. Generic and Notification workflow types are supported for the lines—approval workflows are not supported for the lines. Multiple workflows can be associated to the line types, with one of the workflows set as the default.
### High Priority Issue Report

#### Create Report

For any duplicated attribute the search will be conducted on both attribute values.
* Indicates required field.

<table>
<thead>
<tr>
<th>Change Category</th>
<th>Issue Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>High Priority Open Issues Report</td>
</tr>
<tr>
<td></td>
<td>A report of current high priority Open Issues.</td>
</tr>
</tbody>
</table>

#### Search Criteria

**Select Criteria**: Close | Delete | Duplicate | Add Criteria

<table>
<thead>
<tr>
<th>Select Attribute Group</th>
<th>Attribute</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>^ Number</td>
<td>IS</td>
<td></td>
</tr>
<tr>
<td>Primary</td>
<td>^ Name</td>
<td>IS</td>
<td></td>
</tr>
<tr>
<td>Primary</td>
<td>^ Assigned To</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primary</td>
<td>^ Status</td>
<td>IS</td>
<td>Open</td>
</tr>
<tr>
<td>Primary</td>
<td>^ Approval Status</td>
<td>IS</td>
<td></td>
</tr>
<tr>
<td>Primary</td>
<td>^ Priority</td>
<td>IS</td>
<td>High</td>
</tr>
<tr>
<td>Primary</td>
<td>Reason</td>
<td>IS</td>
<td></td>
</tr>
</tbody>
</table>

**TIP**: Only one attribute is indexed. You must enter one value for at least one indexed attribute to perform search.

#### Display Format

**Display Format**: Assignment Information | Go

<table>
<thead>
<tr>
<th>Type</th>
<th>Number</th>
<th>Name</th>
<th>Status</th>
<th>Approval Status</th>
<th>Priority</th>
<th>Assigned To</th>
<th>Need by Date</th>
<th>Action Log</th>
</tr>
</thead>
</table>
Defining Item Structures

This chapter covers the following topics:

• Overview of Item Structures
• Defining Structures
• Display Formats and Search Criteria for Structures
• User Defined Attributes for Structures and Components
• Valid Component Type Rules

Overview of Item Structures

Important: The seeded structure type of Packaging Hierarchy is only available to customers who have licensed Product Information Management Data Librarian, and it is not available to licensees of Product Lifecycle Management except those that have licensed both Product Lifecycle Management and its option, Product Data Synchronization for Global Data Synchronization Network and 1SYNC Services.

Use the following tasks to define item structures.

<table>
<thead>
<tr>
<th>Task</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining Structure Types</td>
<td>Yes</td>
</tr>
<tr>
<td>Defining Structure Names</td>
<td>Yes</td>
</tr>
<tr>
<td>Defining Item Attachment Categories for a Structure Type</td>
<td></td>
</tr>
</tbody>
</table>
Defining Structures

Defining Structure Types

A structure type enables you to classify and characterize item structures throughout the lifecycle of an item. For example, you can have structure types for Engineering, Manufacturing, and Service. You can define an unlimited number of structure names for a structure type, enabling you to create several different structures of the same type that you can use for comparison. For example, you can create several different engineering BOMs (EBOM1, EBOM2, EBOM3) using the structure type Engineering BOM with different component costs and BOM grading attributes. Comparing these BOMs enables you to make better decisions earlier in the lifecycle of your products.

You can associate item attachment categories, criteria templates and results formats with a structure type. Structure types are hierarchical, with structure names, item attachment categories, criteria templates and results formats inherited throughout the hierarchy.

Defining Structure Names

You can define multiple structure names for each structure type and enable them in different organizations. For example, you can have three engineering BOMs (EBOM1, EBOM2, EBOM3) that use the structure type Engineering BOM; you can use these to compare designs. Manufacturing BOMs may be slightly different in each global region because similar parts are procured from different suppliers. You can have a structure type Manufacturing BOM with different structure names (MBOM_US, MBOM_EMEA, MBOM_APAC) for each of your manufacturing regions around the globe. Structures within a structure type share the same common characteristics.

Note: When creating a packaging hierarchy structure:

- Create the structure in the master organization. Reference the structure from other organizations. See: Referencing Common Bills and Routings, Oracle Bill of Material User’s Guide.
- Assign units of measure that belong to the same unit of measure class to each component within the hierarchy. See: Defining Unit of
Defining Item Attachment Categories for a Structure Type

Associating attachment categories with a structure type enables you to view item attachments to structure components within a BOM. Suppose you have a CAD drawing attached to each of the major subassemblies in each of your engineering BOMs. You can associate the attachment category CAD Drawing to the structure type Engineering BOM. The CAD drawings for each of the subassemblies will then appear in the EBOM structures.

Defining a Criteria Template for a Structure Type

Criteria templates enable you to save frequently performed search criteria based on a list of attributes. Criteria templates for structures enable you to search components within a structure. You can associate with a structure type an existing criteria template for a given item catalog category, or create a new criteria template to search items within a structure.

Administrator-created criteria templates are available to all users. This helps users quickly find components within a structure without having to create their own criteria templates. You can specify a default criteria template for each structure type.

Defining Results Format for a Structure Type

Results formats enable you to predefine search results views for a structure type. You can use these views to look at different sets of attributes of component items within a structure. Both administrators and users can create results formats. Administrator-created results formats are available to all users. User-created result formats are available only to the users who created them. You can define as many results formats as you wish, and you can also denote the most commonly used as the default results format.

You can define a results format by listing the common information a typical user might wish to view for components in a structure. For example, you might select the attributes Item Name, Description, Component Quantity, Capacitance Value, or Tolerance. Results formats for structures enable you to include navigation links for actions such as item Where Used, First-Level components, and View Components in Hierarchy.
Display Formats and Search Criteria for Structures

Search Criteria

Administrators can define search criteria for a given structure type. All search criteria defined by administrators are available to all users viewing a structure of the given structure type to which the search criteria is associated. On selecting a search criteria, only those components that satisfy the criteria will be listed in the hierarchy and the components that do not match the criteria will be filtered out. Intermediate nodes will be listed even though they might not satisfy the criteria to maintain the context for a lower level component. Primarily search criteria are useful to filter only relevant components in a bill having a large number of components. Search criteria templates can be built using item and component base and user defined attributes.

Display Formats

Administrators can define display formats for a given structure type. All display formats defined by administrators are available to all users viewing a structure of the given structure type to which the display format is associated. Display formats allow user to view attribute information across an entire structure. The display formats can include item and component base attributes and user defined attributes.

Oracle Product Lifecycle Management provides the following system defined display formats:

<table>
<thead>
<tr>
<th>Display Format Name</th>
<th>Fields Included in the Display Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component Information</td>
<td>Component Name, Description, Revision, Category</td>
</tr>
<tr>
<td>Component Details</td>
<td>Component Name, Description, Revision, Item Sequence, Operation Sequence, UOM, Quantity, Planning%, Yield, Item type</td>
</tr>
<tr>
<td>Change Order</td>
<td>Component Name, Description, Revision, Implemented, Change Order</td>
</tr>
<tr>
<td>Comments</td>
<td>Component Name, Description, Revision, Comments</td>
</tr>
<tr>
<td>Effectivity</td>
<td>Component Name, Description, Revision, Effectivity Control, Effective From, Effective To, Change Order</td>
</tr>
</tbody>
</table>
User Defined Attributes for Structures and Components

User defined attributes for structures and/or components allow the user to capture specific details about the structure or component. The attributes whose value(s) can be captured for a Structure, are defined by the administrator, while defining the Structure Types. User defined attributes has an administrative workbench which helps the setup of the attribute groups, attributes and the value set tied to an attribute, if any. Different types of structures and its components will have different attribution requirements, For example, when performing analysis of various design scenarios, every structure may need to capture the weight and cost attributes. Similarly, components may need to capture attributes like "Mean Time between failure" which are unique to a component usage in a specific structure.

When setting up the Structure Types, administrators can setup the attribute groups associated with that Structure Type.

Administrators can add attribute groups and define the association level of the attribute group, which can be either the Structure or the component.

End User View
The end user view of extensible attributes would be based on the administrative definition of the attribute groups/attributes and the value set tied to them. The pages would be generated dynamically.

**Valid Component Type Rules**

The system enables you to add valid component user items to a structure. It also enables you to copy a structure to a new structure when the item list of values includes only valid items that have defined component types for the parent item type.

**Adding Components to a Structure**

When adding components to a structure using the 'Add' right click action, the Item Search and Item Advanced Search restricts the list of component items that can be added as components to an item. This list will include only the valid component user items configured for the item type of the parent/subassembly.

When adding components to a structure by dragging and dropping components from Favorites, Item Search Results, the system restricts the list of component items that can be added as components to an item. The component items, which can be dragged and dropped, include only the valid component user items configured for the item type of the parent/subassembly.

**Structure Copy**

When copying a structure to a new structure, the Item list of values (LOV) for the target structure includes only the valid items that have defined component types for the parent item type. Therefore, when you are defining the valid component item types for parent item types, if the parent item type is also a valid component item type, define it as a component item type, as well. Complete the same validation when copying a structure for a subassembly, which is added by dragging from another structure window.

**Structure Bulkload and Open Interface**

When importing a structure through bulkload or the open interface, the system automatically considers item type validations.

**Functional Rules**

- Define User Item Types in the Item Master
- Define Valid Component Item Types for a Parent User Item Type
- Validate a parent item type with multiple component types in an OR condition
- Item LOV’s in Add Component and Item Advanced Search will validate the valid component item types for a parent item type to which the component is being added
- Target Structure Item LOV will validate the valid component item types for a
parent item type

- Structures created through bulkload or Open interface need to validate the valid component item types

- Once structures have been defined with existing valid component types on deletion of a component user item type for a parent item type, all new structures will honor the new component user item types definition.
Setting Up E-Business Suite Attachments

Overview of E-Business Suite Attachments

Oracle Product Lifecycle Management and Oracle Product Information Management Data Librarian provide integration to Oracle Files, Oracle Content Services, and Oracle Content DB (a module within Oracle Database). These content repositories provide users with a wide range of file management capabilities. Content DB, however, provides the most flexibility for organizing unstructured data and Oracle recommends using Content DB over other repositories. The content of this chapter explains how to set up E-Business Suite attachments with Oracle Content DB. Some of the set up steps, features, and processes do not apply to other repositories. For additional information about the file management features available with Oracle Content DB, see the Oracle Content DB Web client online help.

Managing Workspaces or Shared Folders

Oracle Content DB uses the concept of workspaces or shared folders. Workspaces are folders that are accessible by a defined set of users who are assigned a set of privileges. The following are examples of privileges:

Administrator

The role of the workspace Administrator is to manage the workspace. The management role includes such functions as adding and removing workspace members, modifying member access, requesting additional workspace quota, and specifying the visibility and workspace type. The workspace Administrator has full access to the files and folders in the workspace, and can unlock files that have been locked by other workspace members.

Participant

A workspace Participant can edit the contents and properties of files and folders in a workspace. A workspace Participant can copy and move items to and from the workspace, and can delete files and folders from the workspace and the workspace's
Trash folder.

**Viewer**

A workspace Viewer is restricted to viewing workspace files and folders.

For more information about the file management features available with Oracle Content DB, see the Oracle Content DB Web client online help.

---

**Structuring Folders/Workspaces**

Oracle Content DB provides a great deal of flexibility in the structuring of data. The following security considerations should factor in to implementation decisions about the organization of unstructured data (documents/files/folders):

- Examine what content is internal to the enterprise and what may be shared externally as part of the collaboration process or on a website.
- Access privileges by organizations modeled as Organizations in EBS.
- Access controlled by item catalog categories or types of items.
- Access controlled by business objects in EBS such as Items, Change Management, Projects, and Purchase Orders.
- Access controlled by the lifecycle state of the item.

Following are some examples for structuring folders/workspaces.

**Example 1: Model Workspaces as Item Catalog Categories**

<table>
<thead>
<tr>
<th>/Engines</th>
<th>Workspace</th>
<th>Item Catalog Category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/M10000</td>
<td>Folder</td>
</tr>
<tr>
<td></td>
<td>/M10000 Rev 1.0</td>
<td>Sub Folder</td>
</tr>
<tr>
<td></td>
<td>Functional</td>
<td>Sub Folder</td>
</tr>
<tr>
<td></td>
<td>Specification</td>
<td>Sub Folder</td>
</tr>
<tr>
<td></td>
<td>/CAD View</td>
<td>Sub Folder</td>
</tr>
<tr>
<td></td>
<td>/M10000 Rev 2.0</td>
<td>Sub Folder</td>
</tr>
<tr>
<td></td>
<td>/M10000 CO</td>
<td>Folder</td>
</tr>
<tr>
<td></td>
<td>/Manufacturer</td>
<td>Item Catalog Category</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Item</td>
</tr>
<tr>
<td></td>
<td>/Manufacturer</td>
<td>Item Revision</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Doc. Type</td>
</tr>
<tr>
<td></td>
<td>/Change</td>
<td>Doc. Type</td>
</tr>
<tr>
<td></td>
<td>Management</td>
<td>Item Revision</td>
</tr>
<tr>
<td></td>
<td>/Intranet</td>
<td>Item Specific CR/CO</td>
</tr>
<tr>
<td></td>
<td>/Internet</td>
<td>Workspace</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Manufacturer Name1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workspace</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Manufacturer Name2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workspace</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic CR/CO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workspace</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Public to Enterprise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Workspace</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Public to www</td>
</tr>
</tbody>
</table>
### Example 2: Model Workspaces as Organizations or Divisions in a Company

<table>
<thead>
<tr>
<th>Org 1</th>
<th>Workspace</th>
<th>Organization 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>/M10000</td>
<td>Folder</td>
<td>Item</td>
</tr>
<tr>
<td>/M10000 Rev 1.0</td>
<td>Sub Folder</td>
<td>Item Revision</td>
</tr>
<tr>
<td>/Functional Specification</td>
<td>Sub Folder</td>
<td>Doc Type</td>
</tr>
<tr>
<td>/CAD View</td>
<td>Sub Folder</td>
<td>Doc Type</td>
</tr>
<tr>
<td>/M10000 Rev 2.0</td>
<td>Sub Folder</td>
<td>Item Revision</td>
</tr>
<tr>
<td>/M10000 CD</td>
<td>Folder</td>
<td>Item specific CR/CO</td>
</tr>
<tr>
<td>/Org 2</td>
<td>Workspace</td>
<td>Organization 2</td>
</tr>
<tr>
<td>/Manufacturer 1</td>
<td>Workspace</td>
<td>Manufacturer Name 1</td>
</tr>
<tr>
<td>/Manufacturer 2</td>
<td>Workspace</td>
<td>Manufacturer Name 2</td>
</tr>
<tr>
<td>/Change Management</td>
<td>Workspace</td>
<td>Generic CR/CO</td>
</tr>
<tr>
<td>/Intranet</td>
<td>Workspace</td>
<td>Public to Enterprise</td>
</tr>
<tr>
<td>/Internet</td>
<td>Workspace</td>
<td>Public to www</td>
</tr>
</tbody>
</table>

### Example 3: Model Workspaces as Business Objects in EBS

<table>
<thead>
<tr>
<th>Items</th>
<th>Workspace</th>
<th>Business Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Engine</td>
<td>Folder</td>
<td>Item CatalogCategory</td>
</tr>
<tr>
<td>/M10000</td>
<td>Sub Folder</td>
<td>Item</td>
</tr>
<tr>
<td>/M10000 Rev 1.0</td>
<td>Sub Folder</td>
<td>Item Revision</td>
</tr>
<tr>
<td>/Functional Specification</td>
<td>Sub Folder</td>
<td>Doc Type</td>
</tr>
<tr>
<td>/CAD View</td>
<td>Sub Folder</td>
<td>Doc Type</td>
</tr>
<tr>
<td>/M10000 Rev 2.0</td>
<td>Sub Folder</td>
<td>Item Revision</td>
</tr>
<tr>
<td>/M10000 CD</td>
<td>Sub Folder</td>
<td>Item specific CR/CO</td>
</tr>
<tr>
<td>/Change Management</td>
<td>Workspace</td>
<td>Business Object</td>
</tr>
<tr>
<td>/Project Management</td>
<td>Workspace</td>
<td>Manufacturer Name 1</td>
</tr>
<tr>
<td>/Manufacturer 1</td>
<td>Workspace</td>
<td>Manufacturer Name 2</td>
</tr>
<tr>
<td>/Manufacturer 2</td>
<td>Workspace</td>
<td>Public to Enterprise</td>
</tr>
<tr>
<td>/Intranet</td>
<td>Workspace</td>
<td>Public to www</td>
</tr>
<tr>
<td>/Internet</td>
<td>Workspace</td>
<td>Public to www</td>
</tr>
</tbody>
</table>
Attachment Review and Approval

Attachment Review or Approval provides:

- Flexible and scalable way of defining different types of reviews/approvals;
- Capture of user-defined attributes for them;
- Secure collaboration
- Flexible and advanced approval routing definition.

Complex parallel and serial approval routes can be created as predefined templates to enforce strict approval processes, or ad hoc routings to obtain additional approvals within the enterprise can be specified.

Attachments could be selected and submitted for Review or Approval. The Review and Approval workflows may be independent of each other. The Approval and Review process will use predefined approval templates with support for ad-hoc addition of new steps. You can add steps to the process only if a workflow routing is associated with the particular status. You cannot delete predefined (in the item catalog category) steps. You can add additional assignees to a step, but you cannot delete predefined assignees. Based on the Approval or Review type chosen, the workflow/approval routing is determined. Appropriate notifications are sent at the various stages of the approval process including a final notification of approval. Approvers could be derived from user's role on the business object.
The system indicates the current status of the attachment. An attachment could have one of the following statuses of Approved/Reviewed/Submitted for Approval/Submitted for Review or Rejected. If the Document status is not one of these, then it means that the attachment has not been submitted through a formal review and approval process and is in a status of Draft or Unapproved. As the attachment progresses through the Approval process the attachment status should be automatically updated and the approval or review itself should be appropriately routed to different users for their review or approval.

The Attachment Review and Approval process is built using the Change Management Framework with Approval and Review being modeled as Change Categories. Thus, creating the different Approval and Review types is similar to Defining the Header Types for a Change Category, with a few minor differences. For example Lines and associated Line types cannot be defined for the Approval and Review types. Similarly the References, Dependencies and Attachments sections are not available.

Once an Attachment Approval or Review type has been defined, then with each status a workflow could be associated. An Approval type must have an Approval status with an approval workflow associated with it. For more details about creating workflows, see Defining Workflow Templates, page 5-10.

**Change Management for Attachments**

Change Management for attachments provides change control functionality for all types of attachments, a crucial and important piece of non-structured information about the product.

Types of attachment changes supported using a change order include:

- adding attachments
- deleting attachments

Change control is achieved by providing a mechanism to author, track and manage changes to any of the entities above through a change order approval and implementation process. You can:

- Specify items and attachments that will be revised or effected in a change order.
- Author and specify the exact changes that are desired or required to resolve an issue or change request.
- Route the change order for approval to the appropriate people responsible for the revised item based on item roles, change desired and lifecycle of item.

Once the change order is approved, implement the changes manually or automatically on the specified implementation date.

Change Management for attachments is supported through the Change Management
functionality. User can create a Change Order and specify changes to the attachments of a reviewed item. There are no additional steps required in the setting up of a change order to be able to do attachment changes for a revised item other than those mentioned in Defining Change Categories and Types, page 5-2.

**Setting Up a Repository for Attachments**

Oracle recommends using the Oracle Content Database (DB) as the repository to store files. Oracle Content DB can integrate with other repositories using one of the following services:

- WebDAV
- WebServices

If you use WebServices, Oracle Content DB retrieves your single sign on user ID and password from the Oracle Internet Directory (OID). So, if you are signed on to the Oracle Enterprise Business Suite (EBS), you do not need to sign on to Oracle Content DB separately. If you use WebDAV, you must sign on to Oracle Content DB and Oracle EBS separately. Oracle recommends using the WebServices protocol.

To enable single sign on with WebServices, you must:

- install OID on a standalone server.
- register EBS and Content DB in the same OID.
- export existing EBS users into OID.

**Tip:** If you use Content DB as your repository, your system administrator can convert existing WebDAV files to WebServices files by running the concurrent program WebDAV to WebService Repository Migration. The user name specified in the concurrent program needs sufficient privileges to version all of the files involved. Run the concurrent program in simulation mode first to verify that the user name specified has sufficient privileges.

When using EBS attachments, you can set up an Oracle Content DB repository as described above or use:

- an older Oracle repository such as Oracle Content Services
- third-party repositories that support WebServices or Web-based Distributed Authoring and Versioning (WebDAV) protocol
- the native E-Business Suite repository (Oracle Files)

However, if you choose to use a repository other than Oracle Content DB, you will not
have access to the latest file management features. For more information about setting up repositories, go to http://metalink.oracle.com and search for Metalink Note #305373.1.

To set up the repository:
1. Navigate to the Repository Setup page.
2. Select an existing repository and click Update.
   If no repositories exist, then click Add a Row.
3. In the Update Repositories page, click Add Another Row.
4. Enter data in the following fields:
   • Short Name - the short name of the repository
   • Name - the descriptive name of the repository
   • Description - optional. The description of the repository
   • Integration Mode - select either WebDAV or WebService
   • Service URL - the location (URL) of the integration mode selected.
     Use this format: http(s)://<ocs hostname>:\<port>\content/ws
   • WebDAV Connection URL - the location (URL) of the integration mode selected
     Use this format: http(s)://<ocs hostname>:\<port>\content/dav
   • Sequence - the sequence in which you want the repositories to appear during user selection.
   • Certificate path
5. Click Apply.

Related Topics
Overview of E-Business Suite Attachments, page 7-1
This chapter covers the following topics:

- Using Document Management
- Setting Up the Repository

**Using Document Management**

Document Management and Collaboration (DOM) enables the storing, organizing, and sharing of product information, including unstructured data such as files, internet links, and images. DOM requires a repository to store files and it uses Change Management to control changes to documents and files. DOM uses Oracle Content DB, a module within Oracle Database, as the repository for its documents and files. Using Oracle Content DB provides the most file management features and flexibility, but you can also use other types of repositories with less functionality. For more information about the file management features available with Oracle Content DB, see the Oracle Content DB Web client online help.

**Related Topics**


Setting Up the Repository, page 8-1

**Setting Up the Repository**

Oracle Document Management and Collaboration uses the Oracle Content Database (DB) as the repository to store files attached to documents. Oracle Content DB can integrate with other repositories using one of the following services:

- WebDAV
• WebServices

If you use WebServices, Oracle Content DB retrieves your single sign on user ID and password from the Oracle Internet Directory (OID) so, if you are signed on to the Oracle Enterprise Business Suite (EBS), you do not need to sign on to Oracle Content DB separately. If you use WebDAV, you must sign on to Oracle Content DB and Oracle EBS separately. Oracle recommends using the WebServices protocol.

To enable single sign on with WebServices, you must:
• install OID on a standalone server.
• register EBS and Content DB in the same OID.
• export existing EBS users into OID.

**Tip:** If you use Content DB as your repository, your system administrator can convert existing WebDAV files to WebServices files by running the concurrent program WebDAV to WebService Repository Migration. The user name specified in the concurrent program needs sufficient privileges to version all of the files involved. Run the concurrent program in simulation mode first to verify that the user name specified has sufficient privileges.

You can also define other repositories, such as:
• an older Oracle repository such as Oracle Content Services
• third-party repositories that support WebServices or Web-based Distributed Authoring and Versioning (WebDAV) protocol
• the native E-Business Suite repository

However, if you choose to use a repository other than Oracle Content DB, you will not have access to the latest file management features. For more information about setting up other repositories, go to http://metalink.oracle.com and search for Metalink Note #305373.1.

**To set up the repository:**
1. Navigate to the Repository Setup page.
2. Select an existing repository and click Update.
   If no repositories exist, then click Add a Row.
3. In the Update Repositories page, click Add Another Row.
4. Enter data in the following fields:
- Short Name - the short name of the repository
- Name - the long name of the repository
- Description - optional. The description of the repository
- Integration Mode - select either WebDAV or WebService
- Service URL - the location (URL) of the integration mode selected.
  Use this format: http(s)//<ocs hostname>:<port>/content/ws
- WebDAV Connection URL - the location (URL) of the integration mode selected
  Use this format: http(s)//<ocs hostname>:<port>/content/dav
- Sequence - the sequence in which you want the repositories to appear during user selection.
- Certificate path

5. Click Apply.

Related Topics

Document Change Management and Distribution

This chapter covers the following topics:

- Document Change Management and Distribution Process Overview
- Setting Up Document Change Management and Distribution
- Managing Document Changes

Document Change Management and Distribution Process Overview

Change management enables you to control changes to the document-item relationship at the item level, as well as changes to the document itself at the document level. When changing the relationship of a document attached to an item, the document changes are managed using the item’s change policy. When changing a document directly, the changes must follow the document’s change policy. You can distribute documents and files by attaching them to change notification lines.

Change Management for Item-related Documents

You can associate a document to an item in the Item Catalog, choosing either to associate an item revision to a specific document revision or to dynamically associate an item (disregarding revision) to the latest document revision (a floating relationship). The types of change control provided for item-related documents include:

- adding or deleting a document revision associated to an item or item revision.
- changing the revision of the document associated to an item or item revision.

Define change policy for the item-document association based on the item’s lifecycle phase. The system follows the item’s change policy when changing the item’s related document associations from within the Item Catalog.
### Change Management for Documents

Change management for documents works similarly to change management for items, enabling you to create issues, change requests and change notifications for documents. The only exception is that you cannot create change orders for documents.

### Document and File Distribution

To distribute documents and files, create notifications, add documents to the notification lines, and distribute them. You can search for documents, then, from the search results, select multiple documents to add to the notification as separate lines. You can also add one or more files to a notification line. Each line within a notification can be distributed to a different set of recipients.

### Related Topics

- Associating Documents with Items and Other Documents, Oracle Product Lifecycle Management User’s Guide
- Managing Document Changes, page 9-13

### Setting Up Document Change Management and Distribution

You can define change header and line types for documents just as you do for items within the following seeded change categories:

- Issue
• Change Request

• Change Notification

You can also define change header and line types for documents in any custom change categories that use the above categories as base categories.

**Caution:** You cannot use the Change Order category for documents.

Oracle provides the following seeded change header type, line types, and workflow process for the Change Notification category:

• Seeded change header type
  • Distribution Notification - used for document and attachment approvals.

• Seeded change line types
  • Document Distribution - used to distribute documents for notification and review. Enter the document’s name as the line type subject and add one line type for each document.
  
  • Miscellaneous Distribution - used to distribute files independent of documents. Associate the files at the line level as attachments. You can attach multiple files to one line.

• Seeded workflow process
  • Request Response - used to request a response about a particular line in the change notification. You cannot request approval at the line level.

**To define the change policy for documents associated with items:**

Change policies for documents associated with items apply to a particular item catalog category lifecycle phase. You can define whether or not changes are allowed, require a change order, or are not allowed during a particular item catalog category lifecycle phase. For additional information, see: Defining Change Policies, page 3-30.

1. Navigate to the Setup Workbench (Item Catalog Administration). Search for and select an item catalog category.

2. From the **Basic Information** page, select Lifecycles.

3. From the **Lifecycles** page, select the lifecycle for which you want to define a change policy.

4. In the Change Policy region of the **Lifecycles** page, in the View field, select to view the change policy for associations. Click Go.
5. In the Update field, select the lifecycle phase for which you want to define a change policy. Click Go.

6. In the Update Change Policy for: Associations page, you can either update an existing change policy or add a new change policy.

To add a new change policy, click Add Another Row.
7. When adding a new change policy, in the Associations field, select Related Documents.

8. For both updating and adding a new change policy, in the Change Policy field, select one of the following three options:
   - Allowed - you can make changes to related documents during this lifecycle phase of the item without following the change management process.
   - Change Order Required - you must submit a change order and receive approval before making changes to related documents during this lifecycle phase of the item.
   - Not Allowed - you cannot make any changes to related documents during this lifecycle phase of the item.

9. Click Apply.

To define change header and line types for documents:
1. Navigate to the Categories page in the Setup Workbench, Change Management tab.

   Begin defining a change header type as described in Managing Change Header Types, Oracle Product Lifecycle Management User’s Guide or a change line type as described in Managing Change Line Types, Oracle Product Lifecycle Management User’s Guide. The exceptions you must follow for documents are described in the following steps.

2. In the Subject field, select Document Revision.
3. Under the Subject Selection Criteria heading, click Add Another Row to assign attributes. Choose from the following document attributes:

- Document Catalog Category
- Document Status
- Document Lifecycle Phase

4. For each document attribute selected, assign a value to determine when changes are allowed.

For example, you only want to enable the creation of a change issue when a document is in the lifecycle phase Create. After defining a new issue type for document revisions, select the attribute Document Lifecycle Phase with a value of Create.
To define the distribution of documents, files, and folders:

The document change management distribution process at the change header type level operates similarly to the change management approval process for an item. Oracle recommends a workflow such as this:

**Recommended Workflow for Change Category = Change Notification, Change Header Type = Distribution Notification**

<table>
<thead>
<tr>
<th>Workflow Step Number</th>
<th>Status</th>
<th>Status Type</th>
<th>Default Workflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Open</td>
<td>Open</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>Approval</td>
<td>Approval</td>
<td>Document Approval</td>
</tr>
<tr>
<td>Workflow Step Number</td>
<td>Status</td>
<td>Status Type</td>
<td>Default Workflow</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
<tr>
<td>30</td>
<td>Distribution</td>
<td>Review</td>
<td>-</td>
</tr>
<tr>
<td>40</td>
<td>Completed</td>
<td>Implemented</td>
<td>-</td>
</tr>
</tbody>
</table>

Using the above workflow at the change header type level enables you to:

- gain approval to distribute the documents and files attached to the change notification at the line level.
- automatically start the distribution process at the line level once the header workflow status reaches Approval.
- automatically move the header workflow status to Distribution once all documents and files are distributed at the line level.

When distributing documents and files for review, you can only request a response, not an approval. To distribute documents and files for review, use the Request Response workflow process.

**Important:** Change Notification line types only support the workflow type Notification. They do not support the workflow type Approval.

For more information about defining line types, workflow statuses, and managing workflows, see:

Defining Line Types, page 5-21
Creating Statuses, *Oracle Product Lifecycle Management User’s Guide*
Managing Workflows, *Oracle Product Lifecycle Management User’s Guide*

1. Navigate to the Setup Workbench. Select the Change Management tab, then select the Change Notification category.
2. Within the Change Notification category, select the Types tab. Click on the type name Distribution Notification, then click the Workflow link.

3. In the Workflow page, define a workflow similar to the Recommended Workflow for Change Category = Change Notification, Change Header Type = Distribution Notification table shown previously.

   **Note:** Oracle recommends defining a workflow similar to the above table to automate the distribution process.
4. Navigate back to the **Category** page. Within the Change Notification category, select the Line Types tab. Click on the line type name Document Distribution.

5. In the **Update Change Notification Line Type** page, update the subject selection criteria and assignees as needed.
6. Select a workflow template from the list or click Add Another Row to specify a new workflow template.

**Tip:** The workflow template selected must be of workflow type Notification. This workflow type drives the line distribution process.
To assign workflow request response processes to a person or group

You can define and manage change management workflows for documents similarly to items. For more information about workflows, see: Managing Workflows, Oracle Product Lifecycle Management User’s Guide.

This group of steps explains how to assign a workflow request response process to a person or group.

7. Navigate to the Setup Workbench. Click the Change Management tab, then the Workflow sub-tab. Click the Line Templates link.

8. In the Line Templates page, find and select the workflow template name you want to update. Click Update.
9. From the Update Workflow Template page, you can add, update, or delete workflow processes.

Related Topics

Document Change Management and Distribution Process Overview, page 9-1
Managing Document Changes, page 9-13

Managing Document Changes

You can manage the following changes to documents:

- The relationship between an item and a document:
  - Add or delete a document association to an item or item revision
  - Change the revision of the document associated to an item or item revision

- Issues and change requests for a document

Before approving an issue or change request, send notifications of document and file changes, requesting a response.
To change the association between an item and a document:
You can define the level of control you want over changes to the association between an item and a document. Define this level of control at the item catalog category level. Choose from the following levels of control:

- Changes allowed
- Change order required
- No changes allowed

See: To define the change policy for documents associated with items, page 9-3.

If a change order is required, then use the following steps to change the association between an item and a document.

1. In the Item Catalog, find the item with the associated document that you want to change. See: Searching for Items, Oracle Product Lifecycle Management User’s Guide.

2. From the Search Results page, click the item link in the Item column to navigate to the item’s Overview page.

3. Click the Associations sub-tab, then the Related Documents link.

4. To change the association between the item and a document, click Create Change Order.

**Tip:** The Create Change Order button only appears when you have defined the change policy as Change Order Required.
Create a change order as described in Creating Change Orders, Oracle Product Lifecycle Management User’s Guide.

5. Once you have created the change order, in the **Revised Items** page, select the Related Documents tab. Click Update.

6. In the **Related Documents: (item number)** page, make changes to the item-document relationships. Possible changes include:
   - changing the revision of the associated document to another specific revision or to the latest revision.
   - deleting a document
   - adding a document

   The Action column displays the change made for each related document.

7. Navigate to the item workbench, **Related Documents** page. In the Document Changes field, select the change order you created and click Go.

   You can view the pending changes entered.
To create an issue or change request for a document:

You can enter an issue or change request for a document using change header and line types for document revisions (see: To define change header and line types for documents, page 9-5).

1. Find the document for which you want to enter an issue or change request.

2. In the Document Detail page, select Create Change Request or Create Issue from the Shortcuts field. Click Go.

3. Create a change request or issue as described in Creating Change Requests or Creating Issues, Oracle Product Lifecycle Management User’s Guide, but with the following exception - select a document change header type.
4. In the Create Issue or Create Change Request page, specify a document as the subject of issue or change request.

To create a change notification for a document:
You can create a change notification to distribute documents and files. To distribute a document, associate it to a Document Distribution line type. To distribute files, associate one or more to a Miscellaneous Distribution line type. Before creating a change notification, you must define the selection criteria, the notification assignees,
and modify the workflow template, if necessary. See: To define the distribution of documents, files, and folders, page 9-7.


2. In the Document Detail page, select the Collaboration tab.

   Any distribution notifications involving a specific document are available under this tab. You can view completed distributions, as well as ones that are in various stages of distribution. You can also initiate a new distribution process from this page.

3. In the Create field, select Distribution Notification. Click Go.

4. In the Create Change Notification page, enter the following header information:
   - Change Notification Name (required)
   - Description (optional)
   - Source Type (optional)
   - Priority (optional)
   - Reason (optional)
   - Source Name (optional)
   - Need By Date (optional)

5. In the Subject region, the document defaults to the name of the currently selected document and revision. Optionally, assign the change notification to a project and task. The Assigned to field defaults from the change notification change header type.

   The subject referenced at the header level is mainly for reference purposes, since
documents associated at the header level as the subject are not distributed. To distribute documents or files, you need to associate them at the notification line level.

6. In the Lines tab, click Add Another Row. Specify the documents and files that you want to distribute.

Each document requires its own line type. You can attach more than one file or folder to one Miscellaneous Distribution line type. You can attach multiple line types to a distribution notification.
7. In the Dependencies tab, specify any dependencies on other change notifications.

8. In the References tab, specify any references to an item or project.

9. In the Workflow Approval tab, review the workflow processes the notification generates.

10. In the Attachments tab, add attachments at the change notification header level.
     
     **Note:** Attachments at the change notification header level are not distributed. They are viewable by the assignees to the change header workflow.

11. Click Submit.
    
    The **Change Notification Summary: (Change Notification Number)** page appears.

12. If you have any miscellaneous distribution line types to which you need to attach files, then in the **Change Notification Summary: (Change Notification Number)** page, click on a miscellaneous distribution line type name. Add one or more attachments.

    **To start a line workflow automatically once the notification header reaches a certain status**

You can use the Start After Status field at the change notification line level to start the line workflow automatically. For example, enter Approval in this field to start the line workflow once the change notification header workflow reaches the Approval status. If you leave this field blank, you must start the line workflow manually. Once the line workflow is started, workflow assignees who do not have explicit access to the
document or files already can view them.

13. In the Change Notification Summary: (Change Notification Number) page, select a line name link.

14. In the Actions field, select Update. Click Go.

15. In the Start From Status field, select a status.

16. In the Complete Before Status field, select a status.

   Once the change notification header workflow is at this status, the assignee's view privileges for the line document or files are revoked for assignees who do not have explicit access to the document or files.

**To begin the distribution process:**

To begin the distribution process, the workflow attached at the change notification header level must be at the Approved status. Once a notification is approved, the line workflows are initiated either automatically or manually, which starts the distribution process for the documents and files associated at the line level. (See: To start a line workflow automatically once the notification header reaches a certain status, page 9-20)

Workflow assignees at the change notification header level can view the header subject document and any files attached to the header as long as the files reside in an Oracle Content DB repository, although documents and files attached at the header level are not considered for distribution.

Once all lines are distributed by the line type workflows of workflow type Notification, the change header workflow automatically moves to the Distribution status (assuming this status has a status type of Review). The line workflow status is Distribution in Progress. Once all assignees either receive or decline the distribution the line workflow status changes to Distributed. You must manually move the change header status to Completed once all line distributions reach Distributed status.

**Related Topics**

Managing Workflows, Oracle Product Lifecycle Management User's Guide

Document Change Management and Distribution Process Overview, page 9-1

This chapter covers the following topics:

- Overview of Roles
- Defining Roles

**Overview of Roles**

Oracle provides role-based security to specify the privileges assigned to each user on an item, catalog, catalog category, and change category. A role consists of a set of privileges granted to a user on a specific object, such as an item. For example, you may assign the Design Engineer role to some of the engineers within your organization for a set of items. The Design Engineer role grants them the privilege to view and edit the item. To simplify the maintenance of role-based security, you can define groups of users. You can assign to an item or set of items (for example, all items in an item catalog category) a role for a person, group, company, or all users.

<table>
<thead>
<tr>
<th>Task</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining Change Roles</td>
<td>Yes</td>
</tr>
<tr>
<td>Defining Item Roles</td>
<td>Yes</td>
</tr>
<tr>
<td>Defining Item Catalog Category Roles</td>
<td></td>
</tr>
<tr>
<td>Defining Catalog Roles</td>
<td></td>
</tr>
</tbody>
</table>
Defining Roles

Defining Change Roles

The change role assigned to a user for a change object (for example, issue, change request, change order) determines which actions that user can perform on the change object. For example, a user with an Approver role on a change request is granted the View Basic Change Information and Edit/Delete Change privileges. You can also specify which user-defined attribute groups a user can view and/or edit when granted a change role (see "Setting Up Change Attribute Group Security").

You can assign change roles to a person, group, company, or all users. To simplify maintaining change management security, you can assign change roles directly to the change object or inherit them through item role mapping from the subject item of the change header or change lines. For example, you can map the Design Engineer item role to the Change Design Engineer role for issues, change requests, and change orders. So users with the Design Engineer role on the subject item of the change request header or lines will inherit the Change Design Engineer role as well. You can also assign a default role to all internal users with the site level profile ENG: Internal User Default Role for Changes. A role that is explicitly granted to a user for a change object is a direct role assignment. Roles inherited from an item are inherited role assignments.

Following are the seeded change roles:

- Approver
- Assignee
- Creator
- Requestor
- Reviewer

For more details on creating change roles, see People and Roles Administration, Oracle Product Lifecycle Management User’s Guide or Oracle Product Information Management Distribution Librarian User’s Guide
Defining Item Roles

The item role assigned to a user for an item determines which actions that user can perform on the item. For example, a user with the Design Engineer role is granted the View Item and Edit Item privileges, among other privileges.

You can assign item roles to a person, group, company, or all users. To simplify maintaining item security, you can assign item roles at the item, item catalog category, or organization level. For example, you can assign a user the Design Engineer role for the Motherboard item catalog category; this in turn assigns the Design Engineer role to the user for all Motherboard items. A role that is explicitly granted to a user for an item is a direct role assignment. Roles assigned at the organization or item catalog category level are inherited role assignments. You can map an item role to a change role for each change category. For example, you can map the item Design Engineer role to the Change Design Engineer role for issues, change requests, and change orders. So users with the Design Engineer role on the subject of the change request header or lines will inherit the Change Design Engineer role as well.

Following are the seeded item roles:

- Item Author
• Design Engineer
• Design Reviewer
• Supplier Engineer

For more details on creating item roles, see People and Roles Administration, Oracle Product Lifecycle Management User’s Guide or Oracle Product Information Management Distribution Librarian User’s Guide.
Defining Item Catalog Category Roles

The item catalog category role assigned to a user determines who can create items in a specific item catalog category. For example, a user with the Catalog Category User role on the Motherboard item catalog category is granted the "Create Item of this Catalog Category" privilege, which enables him or her to create Motherboard items.

For more details on creating item catalog category roles, see People and Roles Administration, Oracle Product Lifecycle Management User’s Guide or Oracle Product Information Management Distribution Librarian User’s Guide.
Defining Catalog Roles

The catalog role assigned to a user determines who can view and assign items to a catalog. For example, a user with the Catalog Manager role on the Purchasing Catalog is granted the View Catalog and Manage Catalog privileges. The user can view the items in the Purchasing Catalog and add items to the Purchasing Catalog.

Following are the seeded catalog roles:

- Catalog Manager
- Catalog Viewer
Implementing Role Based Security

This chapter covers the following topics:

- Overview Role Based Security
- Defining Role Based Security

Overview Role Based Security

<table>
<thead>
<tr>
<th>Task</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining Groups</td>
<td></td>
</tr>
<tr>
<td>Implementing Item Catalog Role-Based Security</td>
<td>Yes</td>
</tr>
<tr>
<td>Implementing Catalog Role-based Security</td>
<td>Yes</td>
</tr>
<tr>
<td>Implementing Change Management Role-based Security</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Defining Role Based Security

Defining Groups

You can create groups of people that include internal users, suppliers and customers. Groups are extremely useful for managing role assignments for items, catalogs, and change management. Groups are also useful in change management approval routings, where one or all people in a group must approve a change, or all people in a group should be notified of a change. You can create groups of people for different teams or
departments that serve similar business functions. You can also create groups for different suppliers or customers.

A group consists of one or more members. The person that creates the group is by default a member and the Group Administrator. Each member in a group is assigned a group role that determines which privileges are granted to the member of that group. There three privileges a group role may grant a member: View Group Header, View Group Members, and Manage Group. For example, you can create a Supplier group made up of all your suppliers; however, you do not want any of the members to see who else is in the group. In this case, each of the members would be assigned the Group Reviewer role, which only allows them to see the group header information, but not the other members. In another scenario, you may wish to create an Engineering group where you want members to be aware of who else is in the group. In this case, you can grant each member the View Group Members role. Only a person or group with the Group Administrator role can also manage the group. The Manage Group privilege enables a user to add/delete members and change their roles.

### Group details

Implementing Item Catalog Role Based Security

Role-based data security is provided, enabling users to secure individual data elements. Every item is secured using role-based data security. Your ability to view, edit and perform certain actions on an item is determined by your role on it. At the item level, roles are assigned to users by anyone who has the privilege to add people. You can assign roles at the organization, item catalog category, or item level.
Privileges

A privilege defines a user’s access to an item. For example, a user’s ability to view an item, edit it, and promote its lifecycle phase is determined by his or her privileges on that item. You can define additional privileges to secure user-defined attribute groups.

Roles

A role is a collection of privileges. Roles are a convenient way to group privileges into a bundle that can later on be assigned to users, groups, or companies. For example, a user with the Design Engineer role on an item is granted the privilege to View Item, Edit Item, Add/Delete Item Attachments and more. However, the Design Engineer is not granted the privilege to Promote Item/Item Revision Lifecycle Phase.

You can assign item roles to a person, group, company, or all users. To simplify maintaining item security, you can assign item roles at the organization, item catalog category, or item level. You can specify default roles for all items in an organization. For example, you can assign the Item Reviewer role to the Engineering group so that all engineers can search for and view all items in the item catalog. For each item catalog category you can specify which people can create items by assigning them the Catalog Category User role. For example, you can assign just the NPI engineers the Catalog Category User on the item catalog categories for which they are responsible for maintaining. Next, you can assign default role assignments to people by organization for all items in the item catalog category. For example, you can assign your Motherboard engineers a Design Engineer role for the Motherboard item catalog category. You can assign the buyer for Motherboards a Buyer role. At the item level you can assign direct roles to enable access to specific items. For example, you may assign the role of Supplier Engineer to suppliers with whom you are collaborating on specific items.

Organization level item role assignments
Implementing Catalog Role Based Security

For each catalog you create you can specify which people can view and manage the catalog. Suppose you've set up a Product catalog, Purchasing catalog, and Service catalog. You can assign the Catalog Manager role to your marketing managers to
maintain the Product catalog, while assigning the Catalog Viewer role to All Users so that everyone can view the Product catalog. You can assign your buyers the Catalog Manager role to maintain your Purchasing catalog.

**Assigning Catalog Roles**

![Catalog People](image)

**Implementing Change Management Role Based Security**

You can set up change management security so that most if not all the role assignments are inherited from the subject item of the change object header and lines, the approval routing, and default assignments. You can also assign a default role to all internal users with the site level profile ENG: Internal User Default Role for Changes.

When defining an item you specify a role mapping for each change category: issue, change request, and change order. For example, suppose a supplier creates a change request for a Motherboard item. Then all people with a Design Engineer role on the Motherboard item will get the Change Design Engineer role on the change request. This is the most efficient way to set up change management security.

When you defined your change types in "Step 4: Defining Change Categories and Types," you specified one or more valid approval routings. Each person in the approval routing receives a role on the change object based on the approval routing activity. All Request Approval assignees get an Approver role. Request Comment and FYI assignees get a Reviewer role on the change object. You can also specify a Default Assigned To role for a person/group when you define a change type. The Assigned To person for every change object gets an Assignee role.
**Item role to change role mapping**

**Item Role Detail**

Name: Design Engineer  
Description: PLM Design Engineer  
Object Name: Item

**Item Privileges**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Target Markets</td>
<td>View Target Markets</td>
</tr>
<tr>
<td>Add/Delete Item Attachment</td>
<td>Add/Delete Item Attachment</td>
</tr>
<tr>
<td>Add/Delete Item Project/Task Association</td>
<td>Add/Delete Item Project/Task Association</td>
</tr>
</tbody>
</table>

**Role Mappings**

<table>
<thead>
<tr>
<th>Type</th>
<th>Role Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issue</td>
<td>Change Design Engineer</td>
</tr>
<tr>
<td>Change Order</td>
<td>Change Design Engineer</td>
</tr>
<tr>
<td>Change Request</td>
<td>Change Design Engineer</td>
</tr>
</tbody>
</table>
Understanding Product Data Synchronization

This chapter covers the following topics:

- Overview of Product Data Synchronization
- Overview of Global Data Synchronization Network and 1SYNC
- Global Data Synchronization Network
- Managing Global Trade Item Number Information
- Oracle 1SYNC Registry Messages
- Support 1SYNC Supplier Synchronization Suite
- Notification Processing
- Publish the Registered Information to Trading Partners

Overview of Product Data Synchronization

Important: The seeded structure type of Packaging Hierarchy is only available to customers who have licensed Product Information Management Data Librarian, and it is not available to licensees of Product Lifecycle Management excepting those that have licensed both Product Lifecycle Management and its option, Product Data Synchronization for Global Data Synchronization Network and 1SYNC Services.

Oracle Product Data Synchronization for Global Data Synchronization Network (GDSN) and 1SYNC Services enables companies to securely deliver product information to 1SYNC and via the Global Data Synchronization Network (GDSN), and then to syndicate that data to trading partners.

Product Data Synchronization (PDS) comes with pre-built item attributes that map to
the attribution defined for product data synchronization by GDSN and 1SYNC. PDS also provides extensive validation checks against the approved formats for these attribute definitions to help assure that only clean data is sent. Besides, PDS provides a complete messaging solution that is built to the format mandated by GDSN and 1SYNC.

PDS is designed to help make sure that companies avoid both the direct and indirect costs of sending incorrect product data to 1SYNC and their trading partners. Direct costs include the fines imposed by 1SYNC for sending ‘dirty data’, and indirect costs include delayed or lost sales, etc. PDS’ item attributes come with extensive data validation checks to make sure that input data is clean. The messaging system provides a complete transaction history of all messages to provide a full audit trail of all communications with data pools and trading partners.

Registration and Syndication

Overview of Global Data Synchronization Network and 1SYNC

The Global Data Synchronization Network and its Global Registry provide a foundation for electronic commerce by providing Internet-based product information registration and synchronization services. These services include compliance verification, synchronization of product information, and registry and life cycle management of synchronized products, user locations, and user trade capabilities. These services allow companies to synchronize item information across their entire supply chain. Manufacturers can register and update their product information with GDSN through standard processes, and any of the manufacturer’s trading partners – distributors, sellers, retailers, etc. can access that information.
Oracle PDS allows companies to efficiently synchronize accurate product data with customers and suppliers. PDS provides companies with a complete solution for managing their product information and then for delivering their relevant product data to their trading partners by enabling them to register Global Trade Item Numbers (GTINs) associated with one or many customer sites (Global Location Numbers) to 1SYNC.

The key features of PDS are:

- Author and manage items/GTINs, attributes and associated packaging hierarchies
- Manage registration process of items/GTINs to the Global Registry (GDSN/1SYNC)
- Manage trading partner subscriptions
- Manage publication/syndication of items/GTINs to trading partners
- Manage ongoing changes to product data and appropriate re-syndications to trading partners

**Global Data Synchronization Network**

**Profile Options**

Oracle provides several system profile options to enable or change the behavior of certain features in the application.

Following is a description of the GDSN profile option:

<table>
<thead>
<tr>
<th>Profile</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
</table>
EGO GDSN ENABLED | Governs the GDSN Synchronization functionality. Changing the value to Yes enables GDSN functionality.

Once you turn the flag to yes, then:

- GDSN default functional area appears
- GDSN attributes appear in setup list page
- GDSN attributes get associated to Item Catalog Category
- GDSN Syndicated functional area flag on the Create Item and Functional Classification page gets enabled.

---

**Lookups**

System level lookups store values for the entire site.

<table>
<thead>
<tr>
<th>Look up</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGO_UCCNET_DATA_SOURC_E_GLN</td>
<td>This lookup allows you to enter the Information provider GLN number and name.</td>
<td>Null</td>
</tr>
<tr>
<td>• GLN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Name | **Important:** To specify the GLN and the name of the Information Provider, you need to update a lookup: 

```
LOOKUP_TYPE = EGO_UCCNET_DATA_SOURC_E_GLN
LOOKUP_CODE = GLN
Meaning = Actual GLN
```

```
LOOKUP_CODE = NAME
Meaning = Actual Name
``` |
Setting up GDSN Catalog

GDSN Catalog is similar to other functional area default catalogs that exists in the Oracle E-Business for example Inventory and purchasing catalog. When you associate an Item/GTIN to GDSN Syndicated functional area, Item/GTIN gets assigned to default Catalog associated to the GDSN Syndicated.

You can associate an existing catalog to this functional area or you can create a new Catalog and associate it.

1SYNC data pool requires use of UDEX Catalog as its Item/GTIN Classification. For example, to communicate with 1SYNC data pool, you can create the GDSN catalog using the UDEX catalog category taxonomy. UDEX Category follows a 3 segment flex structure.

The entire UDEX catalog can be set-up in the GDSN catalog or just the categories that are needed as per your business requirements. For example, ‘Safety/Security & Surveillance/Survival’ has two sub categories: ‘Security Other’, and ‘Fire / Chemical Safety products’. If the company makes Lock and does not make any fire or chemical safety products then the catalog Administrator can setup only one child category as ‘Security Other’.

For details on creating catalogs, see the Defining Catalogs section.

You can then use the Import from excel Item Catalog Category assignments functionality to mass assign Item/GTIN to this GDSN Catalog.

GDSN Syndication Functional Area

<table>
<thead>
<tr>
<th>Functional Area</th>
<th>Catalog</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory</td>
<td>Im Items</td>
<td>Inventory Category Set</td>
</tr>
<tr>
<td>Purchasing</td>
<td>Purchasing</td>
<td>Purchasing Category Set</td>
</tr>
<tr>
<td>Planning</td>
<td>Im Items</td>
<td>Inventory Category Set</td>
</tr>
<tr>
<td>Service</td>
<td>Im Items</td>
<td>Inventory Category Set</td>
</tr>
<tr>
<td>Costs</td>
<td>Im Items</td>
<td>Inventory Category Set</td>
</tr>
<tr>
<td>Engineering</td>
<td>Im Items</td>
<td>Inventory Category Set</td>
</tr>
<tr>
<td>Order Management</td>
<td>Im Items</td>
<td>Inventory Category Set</td>
</tr>
<tr>
<td>Product Line Accounting</td>
<td>Inventory</td>
<td>Inventory Category Set</td>
</tr>
<tr>
<td>Asset Management</td>
<td>Enterprise Asset Management</td>
<td>Enterprise Asset Management Category Set</td>
</tr>
<tr>
<td>Contracts</td>
<td>Contracts</td>
<td>Category Set for Contract Item Categories</td>
</tr>
<tr>
<td>Product Reporting</td>
<td>Product</td>
<td>Product Category</td>
</tr>
<tr>
<td>GDSN Syndication</td>
<td>GDSN Catalog</td>
<td>GDSN Catalog</td>
</tr>
</tbody>
</table>

Setting up GDSN Attributes

Oracle Product Data Synchronization seeds 150+ GDSN attributes in various GDSN Attribute Groups. GDSN attribute groups are similar to the existing User-defined attribute group functionality.


List of seeded GDSN Attributes

<table>
<thead>
<tr>
<th>Select</th>
<th>Display Name</th>
<th>Internal Name</th>
<th>Description</th>
<th>Attribute Group</th>
<th>Multi-Row</th>
<th>Database</th>
<th>View</th>
<th>Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>Bar Code</td>
<td>Bar_Code</td>
<td>UCCost: Bar Code</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Country Of Origin</td>
<td>Country_Of_Origin</td>
<td>UCCost: Country Of Origin</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Data Information</td>
<td>Data Information</td>
<td>UCCost: Data Information</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Delivery Method Indicator</td>
<td>Delivery_Method_Indicator</td>
<td>UCCost: Delivery Method Indicator</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>FMCG Identification</td>
<td>FMCG_Identification</td>
<td>UCCost: FMCG Identification</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>FMCG Marking</td>
<td>FMCG_Marking</td>
<td>UCCost: FMCG_Marking</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>FMCG Measurements</td>
<td>FMCG_Measurements</td>
<td>UCCost: FMCG_Measurements</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>GTIN Color Description</td>
<td>GTIN_Color_Description</td>
<td>UCCost: GTIN Color Description</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Handling Information</td>
<td>Handling Information</td>
<td>UCCost: Handling Information</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Handling Instructions</td>
<td>Handling Instructions</td>
<td>UCCost: Handling Instructions</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Hazardous Information</td>
<td>Hazardous Information</td>
<td>UCCost: Hazardous Information</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Manufacturing Information</td>
<td>Manufacturing_Info</td>
<td>UCCost: Manufacturing Information</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Material Safety Data</td>
<td>Material_Safety_Data</td>
<td>UCCost: Material Safety Data</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Order Information</td>
<td>Order Information</td>
<td>UCCost: Order Information</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Packaging Method</td>
<td>Packaging_Making</td>
<td>UCCost: Packaging Method</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Price Data Information</td>
<td>Price_Data_Information</td>
<td>UCCost: Price Data Information</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Price On Trade Item</td>
<td>Price_On_Trade_Item</td>
<td>UCCost: Price On Trade Item</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Size Description</td>
<td>Size_Description</td>
<td>UCCost: Size Description</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Source Data</td>
<td>Source_Description</td>
<td>UCCost: Source Data</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Temperature Information</td>
<td>Temperature Information</td>
<td>UCCost: Temperature Information</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Trade Item Description</td>
<td>Trade_Item_Description</td>
<td>UCCost: Trade Item Description</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Trade Item Hierarchy</td>
<td>Trade_Item_Hierarchy</td>
<td>UCCost: Trade Item Hierarchy</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Trade Item Measurements</td>
<td>Trade_Item_Measurements</td>
<td>UCCost: Trade Item Measurements</td>
<td>UCCost</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You can customize these GDSN Attributes within the GDSN attribute groups. For example –

- You can create value sets and associate it to the attribute as appropriate. For example if you have few selected brand names, you can create a value set for the brand names and associated to the Brand Name attribute.

- Certain Value sets are seeded. For example EANUCC type. If you only use certain types you can customize the value set by disabling the not required ones.

- You can specify default values for any attribute.

- You can disable these GDSN attributes. For example, Drained Weight is an attribute in the Trade Item Measurements attribute group. Drained weight is not a mandatory GDSN attribute and may not be needed for a company in the Hardlines.
Customer GDSN Attribute

Following are some of the GDSN attribute setups you might want to evaluate per your business needs. You need to set up UOMS for these attributes before you can start entering the values. If you have already entered a value for these attributes, you may not be able to set a UOM. In such cases, clear the values for those attributes (for all the items), and then set up the UOM.

<table>
<thead>
<tr>
<th>Attribute Group</th>
<th>Attribute Name</th>
<th>Unit of Measure Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trade Item Measurements</td>
<td>Diameter</td>
<td>Length</td>
</tr>
<tr>
<td>Trade Item Measurements</td>
<td>Gross Weight</td>
<td>Weight</td>
</tr>
<tr>
<td>Trade Item Measurements</td>
<td>Drained Weight</td>
<td>Weight</td>
</tr>
<tr>
<td>Trade Item Measurements</td>
<td>Peg Horizontal</td>
<td>Length</td>
</tr>
<tr>
<td>Trade Item Measurements</td>
<td>Peg Vertical</td>
<td>Length</td>
</tr>
<tr>
<td>Order Information</td>
<td>Order Lead Time</td>
<td>Time</td>
</tr>
<tr>
<td>Handling Information</td>
<td>Maximum Stacking Weight</td>
<td>Weight</td>
</tr>
<tr>
<td>Hardlines Industry</td>
<td>Pieces Per Trade Item</td>
<td>CNT</td>
</tr>
</tbody>
</table>
Setup of Value sets for the following attributes:

<table>
<thead>
<tr>
<th>Attribute Group</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trade Item Description</td>
<td>Coupon Family Code</td>
</tr>
<tr>
<td>Trade Item Measurements</td>
<td>Net Content Value – Net content Unit of Measure</td>
</tr>
<tr>
<td>GTIN Color Description</td>
<td>Color Code Description</td>
</tr>
<tr>
<td>GTIN Color Description</td>
<td>Color Code List Agency</td>
</tr>
<tr>
<td>Size Description</td>
<td>Size Code List Agency</td>
</tr>
<tr>
<td>Size Description</td>
<td>Size Code Value</td>
</tr>
<tr>
<td>Handling Information</td>
<td>Handling Instructions Code</td>
</tr>
<tr>
<td>Hazardous Information</td>
<td>Class Of Dangerous Code</td>
</tr>
<tr>
<td>Hazardous Information</td>
<td>Dangerous Goods Hazardous Code</td>
</tr>
<tr>
<td>Hazardous Information</td>
<td>Dangerous Goods Regulation Code</td>
</tr>
<tr>
<td>Hazardous Information</td>
<td>Dangerous Goods Packaging Group</td>
</tr>
<tr>
<td>Trade Item Harmonized System Identification</td>
<td>Harmonized Tariff System Identification Code</td>
</tr>
</tbody>
</table>

PDS supports hardlines and FMCG industry extensions. These industry extension attributes are not associated to the Item Catalog Category when the EGO GDSN ENABLED profile option is changed to yes. You need to associate these attributes to the Item Catalog category and then create or update the existing page to display the hardlines or the FMCG attributes.

**Setting Up Packaging Structure**

Oracle seeds Packaging BOM as the structure type. Packaging configuration that is created for this structure type is being sent to GDSN data pool as part of the publication message. You can create Structure name for the Packaging Hierarchy for this structure.
type.
User can use this structure type and structure name to create their packaging
configurations.
For more information about structure type and structure names, see the Defining Item
Structures section.

Setting up GTIN Number Generation
The Global Trade Item Number (GTIN) encompasses existing UPC and EAN identifiers,
uniquely identifying all levels of the product hierarchy and delivering data in a
consistent format and structure. The GTIN is a 14-digit number that contains
information about your company and item, as well as how it is packaged.
Oracle Product Data Synchronization provides the ability to automatically generate and
assign GTIN to your product depending on the user-defined function or allow users to
enter by providing free form number field.
Depending upon your User-defined functions, the product classification or attributes
like UPC or EAN codes can be set as function input parameters.
Oracle validates the GTIN check digit, during GTIN creation process.

Setting up GLN for Customers
GLN is the Global Location Number that is assigned to a company for any of its
physical locations. GLN is 13-digit number with last digit being a check digit. Oracle
allows you to assign GLN numbers to each customer addresses using Customer Online
screens.
Global Location Number is not displayed by default, as it is primarily used for the purpose of Product Data Synchronization. Using personalization framework you can display the field.

**Note:** If the Global Location Number field doesn’t appear on the update customer address screen, then please use OA personalization option to display it.

To display Global Location Number:

1. Using system administrator responsibility set the profile option "Personalize Self-Service Defn" to Yes.

2. Login using your user id and navigate to profile option and set the "FND: Personalization Region Link Enabled" to Yes.

3. Navigate to the Customer Update Address screen page for any customer using Customer Online Super User responsibility.

4. Click the personalize region link next to the country list of values attribute.

5. On the Choose Personalization Context page accept the default values for the fields and click apply.

6. In the Personalization Structure section, click the personalize icon for "RowLayout: (glnRowLayout)"

7. In the Personalize properties section, for Rendered row, change the site setting to true. (Notice: Original Definition is set to False).

8. Click Apply to save the changes
Managing Global Trade Item Number Information

**Important:** To use these features except GTIN Cross References, you must have licensed Oracle Product Data Synchronization for GDSN and 1SYNC Services. You must also set the "EGO GDSN Enabled" profile option to "Yes."

Any item in your item catalog can be enabled for GDSN syndication by assigning it to the catalog associated with the GDSN Syndication Functional Area. Once an Item is classified as GDSN Syndicated, GDSN attributes are displayed and you can begin defining GDSN attributes.

**To create an item for GDSN syndication:**
1. On the create Item page, select Item Catalog Category or existing Item.
2. Select GDSN syndicated
3. On the Create Item: Enter Item/Description page, select Trade Item Unit Descriptor.
4. Add or copy GDSN attributes in the copy step.
5. In the Review Item information, view and update GDSN attributes.
6. In the Categories sub tab, view default GDSN Catalog and category assignments. You can update the category assignment, but cannot delete the GDSN Catalog assignment.

   **Note:** The GDSN Syndicated flag applies only when the Item you are creating is an approved Item and doesn’t require a New Item Request approval.

   The GDSN Syndicated flag does not apply when creating multiple Items.

**To enable an existing Item for GDSN syndication:**
1. Find the item that you want to enable for GDSN syndication.
2. On the Item’s overview page, click the Classification and Categories tab.
3. Check whether the item has been already enabled.

4. Click Update

5. On selecting the GDSN Syndicated check box, the categories assignments region gets refreshed with GDSN Catalog and default category assignment.

6. You can update the category assignment, however you cannot delete the GDSN Catalog assignment.

7. Adding a GDSN Catalog Category assignment automatically selects the GDSN Syndicated checkbox.

8. Click Apply to save your changes.

**Note:** Once you enable an Item for GDSN Syndication, you cannot remove it. You cannot clear the GDSN Syndication, nor delete the GDSN Catalog Category assignment. You can update the GDSN Catalog category assignments to a new GDSN category.

Import of Item Catalog Category assignments to the GDSN Catalog using Microsoft Excel automatically enables Items for GDSN Syndication.

---

**Oracle 1SYNC Registry Messages**

Global Registry is a repository for registry messages. Oracle Messages for 1SYNC Registry offers a bundle of pre-configured, certified 1SYNC XML messages to support Supplier and Retailer data synchronization, along with message management tools to give customers a 360 degree overview of the electronic exchange. These messages allow subscribers to register item attribute information as well as send and receive requests and notifications. XML Gateway, in conjunction with the Workflow business event system (WF BES) is used to generate the messages based on the user initiated actions. Specific XML Gateway maps are built for each of the 1SYNC specified XML messages which are then used to communicate with 1SYNC registry. 1SYNC also provides XML schema definitions to allow validation of the XML messages, prior to their propagation to 1SYNC for the necessary action.

Oracle Messages For 1SYNC Registry is tightly integrated with Oracle Advance Product Catalog. Advance Product Catalog is the base application that stores item information by GTIN, and GTIN hierarchy that mimics 1SYNC data structure, for manufacturing and retail companies. Advance Product Catalog is the initiation point for the Registry messages if the customers are Suppliers.

**Important:** For details on implementing Oracle 1SYNC Registry
The following pre-built XML messages are used in the supply-side functionality in 1SYNC:

- Register Catalog Item Registry (RCIR)
- Catalog Item Notification (CIN)
- Query Worklist Messages

**Register Catalog Item Registry**

Register Catalog Item Registry (RCIR) messages add products to the 1SYNC data pool. It updates the brand owner registry if it is the first occurrence. Possible actions are add, change, correct, and delete. Using RCIR, items can be registered in synchronous or asynchronous modes. When a single message is sent to the registry, the 1SYNC registry response is in the synchronous mode. However, when multiple messages (batch mode) are sent, only an acknowledgement is received from the 1SYNC registry synchronously. To obtain processing details of the batch, the query message transaction is initiated.

**Catalog Item Notification**

Catalog Item Notification (CIN) messages are delivered to the recipient. Since each CIN is targeted to a single recipient, CIN allows individual values of the product. It includes extensions as well as industry specific attributes like those for hardlines, FMCG etc. Possible actions for CIN are add, change, correct, delete, and withdraw.

**Query Worklist Messages**

Query messages are used to retrieve information from the GLOBALRegistry or notifications received from trading partners/recipient. XML schema messages are used to transmit the query request. The different query messages used are RCIR in batch mode, Request For Catalog Item Notification (RFCIN), and CIC. Besides, GTIN query is also used.

RCIR in batch mode is used to collect the processing status for asynchronous messages back from registry.

RFCIN is an XML message request for information pertaining to an item. A 1SYNC recipient trading partner (commonly a retailer) who wants to examine the attributes of an item that has been published in the GDSN (Global Data Synchronization Network), but does not yet want to set up a permanent subscription, sends an RFCIN to request a one-time publication of item information from a source trading partner (commonly a supplier) within the GDSN. The GTINs (Global Trade Item Numbers) included in the RFCIN would have been registered in the 1SYNC Data Pool (called ‘GS1 USA’) and the
The recipient trading partner sends an RFCIN to the 1SYNC Data Pool. Item data can be requested for one or more of the following criteria:

- GTIN
- Category
- Target Market
- Global Location Number (GLN)

The 1SYNC Data Pool receives the RFCIN information, and forwards a copy to the source trading partner. If the supply trading partner uses a different data pool, 1SYNC Data Pool submits the RFCIN message to the GS1 Global Registry. In response to an RFCIN, the source trading partner submits a Catalogue Item Notification (CIN) to 1SYNC. CIN contains Item attribute details, including those of industry extensions. 1SYNC receives the CIN document and sends it to the recipient trading partner’s Worklist. For the first thirty-day period after submission of the RFCIN message, the recipient trading partner will receive item information including changes and/or corrections to the items. If the recipient submits a “Reject” confirmation action prior to the expiration of the thirty-day period, changes to the item data will no longer be received. If no action is taken within the thirty-day period, the recipient trading partner will continue to receive the requested Catalogue Item data, even after the thirty-day period has ended, unless a Catalogue Item Confirmation (CIC) message with an action of “Reject” is submitted in response to the CIN. It is the recommended practice for recipient trading partners to submit a CIC message as soon as they decide on the confirmation action they want to take viz. Accepted, Synchronized, Rejected, or Review. The GS1 Global Registry validates the message, determines the Source Data Pool to route the message to, and delivers the message appropriately.

GTIN Query retrieves the elements defining the GTIN from the 1SYNC registry.

**Support 1SYNC Supplier Synchronization Suite**

The initial release of Oracle Messages for 1SYNC Registry offers the entire Supplier Synchronization Suite. Oracle customers who are suppliers or manufacturers use the entire Synchronization Suite to send item information to 1SYNC or to retrieve synchronization feedback from their trading partners. Oracle Messages for 1SYNC Registry is certified on 3.0 version of Supplier Synchronization Suite.

The following figure shows the components for 1SYNC integration for suppliers. Oracle Messages for 1SYNC Registry and Supply Chain Trading Connector are the messaging components of the integration. 10g PartnerConnect provides the AS2 protocol to transport Registry messages to 1SYNC and to receive Registry messages from 1SYNC.

The following provides further details on the 1SYNC Supplier Synchronization Suite:

- Register Catalogue Item Registry (RCIR) message is the message for registering an
item in Registry or updating core attributes of a registered item. Add, Change and Modify features are supported for this message.

- Catalogue Information Notification (CIN) message is a notification from manufacturers/suppliers to their Trading Partners to notify them of new item registration (GTIN) or item information change.

- Catalogue Item Confirmation (CIC) message is the message returned by the Trading Partner in response to a CIN.

- Request for Catalogue Information Notification (RFCIN) is the message sent by a trading partner to a Supplier to request a CIN for specific GTINs.

- Direct Store Delivery

- Market Group Linking

- Hardline Industry Extension support, other industry extensions are in consideration for future release.

**Supply Chain Trading Connector**

In addition to the registry messages, Oracle Messages for ISYNC Registry uses a monitoring/auditing tool called Supply Chain Trading Connector. This tool sits outside of XML Gateway, monitors and audits all incoming and outgoing ISYNC messages for Oracle E-Business Suite. Two features of the tool Collaboration History and Notification Processing are used by the Oracle Messages for Registry product. Each message exchange is defined as a collaboration in this tool, and is modeled by pre-seeded events and shown as detailed steps in the Collaboration History. When an exception occurs due to message failure or data validation, exception reports are initiated by Notification Processing to notify the Business Users of errors.

The Supply Chain Trading Connector is integrated to provide a comprehensive view of the information and the exchange of the messages. From the Item page, you can view the Collaboration status or jump to the Collaboration History to view the detailed information of that Collaboration. From the Collaboration History screen, a Business User can view the item history after viewing the detailed Collaboration information by displaying the item information page.

**Note:** For details on Supply Chain Trading Connector, see the Oracle Supply Chain Trading Connector Implementation Guide.

**Collaboration History**

Due to the nature of B2B standards, it is often difficult for business end users to understand the status of their trading activity. They often approve something that they
know is sent to a trading partner but do not know the status. It is a complicated issue due to the number of messages and acknowledgements sometimes exchanged, as well as, the number of software components that processed the messages for a specific collaboration.

The Collaboration History addresses this challenge by tracking all the events and messages exchanged during a collaboration and making it available to the business end user for inquiry. Using Collaboration History you can track all the messages exchanged, any detailed information including exceptions, and the XML Payloads where available.

**Major Features of Collaboration History**

- Records all key processing events of a unique collaboration or PIP from the initial event or message, receipt or send, to the final message including any exceptions issued by the trading partner or the B2B Gateway in use.
- Enables users to search history by business document number, date, status, document owner, or trading partner
- Permits you to re initiate a collaboration or PIP if allowed by the collaboration
- Enables users to view the XML Payloads associated with a particular collaboration
- Provides Workflow events to create, update, and add details collaboration history for use when implementing a collaboration

**Note:** For further details on Collaboration History, see *Oracle Supply Chain Trading Connector Implementation Guide*

---

**Notification Processing**

The major features of Notification Processing are:

- To capture notification messages from sources outside the applications such as a B2B Gateway or Trading partner.
- To capture Exceptions generated by the XML Gateways
- To record notifications in Collaboration History
- To perform actions based on the Notification settings. Valid actions are:
  - Call API
  - Notify Document Owner
  - Notify Trading Partner
• Notify Administrator
• Raise Business Event
• Start Workflow

• To provide predefined Notification Actions for all supported Collaborations.
• To permit configuration of the predefined Notifications to fit your business requirements.
• The ability to establish Notification Actions for new Collaborations defined in Collaboration History.

Notification Processing is organized into Notification Types, Notifications and Notification Actions. Notifications are raised to represent the success, exception or error at predefined points within a Collaboration. A Notification can be classified as one of 3 types based upon the Notification Source that raised it: Application, XML Gateway or B2B Gateway. A Notification Action defines the action to be taken at a predefined point within a Collaboration based upon the Notification raised.

Notification Actions
A Notification Action has the following characteristics:
• A Collaboration is an occasion where messages are communicated between an enterprise using the Oracle E-Business Suite and a trading partner usually using the internet, or, between an E-Business Suite application and another application. Collaborations represent business processes such as an Order, Change Order, Forecast, or Work in Process action.

• An Application that owns the collaboration. For example Purchasing or Order Management.

• A Notification Source, which describes the software component generating the Notification.

• A Notification to describe the success, error, or warning.

Important: All Notification Actions required for Supply Chain Trading Connector supported Collaborations have been pre-configured. All pre-configured Notification Actions that handle error conditions are pre-configured with the default action of notifying the Administrator. It is expected that additional business rules are specified, using the Notification Processing system, such as initiating a workflow to place an inbound Purchase Order on hold when, for example, a Trading Partner Identification Error is encountered in a given Collaboration.
To create new notifications:
After first determining the messages you plan to receive in the new collaboration, you need to define any new Notifications that are required and finally you define the Notification Actions to be performed when these Notifications are raised for the new Collaboration.

1. Navigate to the Notification Definition window.
2. Click Create Definition.
3. Choose a Notification Source from the drop-down list.
4. Enter the Notification. This is programmatic value or code that your system passes to representing the notification.
5. Enter Description. This is a textual description of what the Notification means.
6. Click Apply.

Note: For further details on Notification Processing, see Oracle Supply Chain Trading Connector Implementation Guide

Publish the Registered Information to Trading Partners

Users can register items in 1SYNC and publish the item to trading partners. Using Product Data Synchronization for GDSN and 1SYNC Services, you can publish Item-GTIN information to customers to inform them of:

• New Item-GTINs being offered
• Changes to existing Item-GTINs
• Withdrawal of Item-GTINs from a specific customer, or delisting of Item-GTINs from the market.

Product Data Synchronization for GDSN and 1SYNC Services is an efficient platform for all Item-GTIN publication processing and messaging, allowing you to:

• Find all new or updated Item-GTINs that need to be published to customers
• Track publication event history, with transaction disposition, for each Item-GTIN customer combination

Product Data Synchronization for GDSN and 1SYNC Services allows you to publish a single Item-GTIN, or multiple Item-GTINs in a batch. Upon 'Publication', a CIN message is sent to your customer for each Item-GTIN via a GDSN data pool. Your
customer can review each message, and decide whether to 'Accept', 'Reject', 'Synchronize' or 'Review' the Item-GTIN. The customer's response is returned via a confirming CIC message, and is stored in your work list on the GDSN data pool. These global registry messages are transferred from the data pool to Product Data Synchronization for GDSN and 1SYNC Services, where the message content and disposition status can be viewed for each Item-GTIN customer combination.

Disposition statuses are:

- Processed
- Failed
- Customer-Accepted
- Customer-Rejected
- Customer-Review
- Customer-Synchronized

**Note:** You will need Publication Management function security, View Item privilege, and View Customer Item cross-reference privilege in the Master Organization to perform the following actions on an Item-GTIN for a customer:

- Initial load
- Publish
- Withdraw

You must set up customer subscriptions for each Item-GTIN before you begin the publication process. The publication status for a customer indicates whether the Item-GTIN requires publication. The publication validation status on the GDSN Attribute pages and syndication page indicates whether the Item-GTIN packaging hierarchy is compliant for publication.

**Note:** Always publish from the top Item-GTIN of a valid packaging hierarchy to a customer, as the entire packaging hierarchy must be included in the CIN message.

A valid packaging hierarchy has these attributes:

- Contains at least one consumable and one orderable GTIN
- Is not a component in another packaging hierarchy
To publish an Item-GTIN to customers:

1. From the top Item-GTIN page, navigate to the Customer Syndication page. This page displays the list of customers to whom you have already published or plan to publish the Item-GTIN packaging hierarchy.

   **Note**: For details about creating and maintaining customer syndication, see "Managing Customer Syndication".

2. For each customer, there is a publication status and a corresponding disposition status. From the Publication status, you can determine which customers the Item-GTIN has not been published to, or requires re-publication. Publication statuses are:
   - Published
   - Not Published
   - Re-publication Needed
   - Publication In Progress
   - Withdrawn
   - Delisted
   - Rejected

   **Note**: You cannot publish information to customers to whom you have previously published that information if no new GDSN attribute changes have occurred since your last publication.

3. Click the Publish action icon to publish the Item-GTIN to a customer. On the Publish Items page, you can view the entire packaging hierarchy and the list of customers for each packaging hierarchy.

4. You can review the name and GLN of the data pool to which the Item is being published, as well as the customer's name, address and GLN.

5. After you submit the Item-GTIN for publication, a CIN message is sent for each customer that the Item-GTIN is being published to. A message group id is
generated to track all Item-GTIN publication messages sent in this transaction.

6. The Disposition status for each message is updated with the response that is received from the customer via the data pool.

   **Note:** If a customer rejects an Item-GTIN publication, Product Data Synchronization for GDSN and 1SYNC Services will not allow you to re-publish the Item-GTIN to the same customer.

   **Note:** A first time publication message to a customer is sent with a 'New Item' or an 'Initial Load' action. Subsequent re-publication messages are sent with a 'Data Change' action. Product Data Synchronization for GDSN and 1SYNC Services automatically derives these actions.

   **Note:** When you update the Item-GDSN attributes for a published Item, its publication status is automatically changed to 'Re-publication Needed' for all customers. However, if you update a customer-specific Item-GDSN attribute, then the publication status is automatically changed to 'Re-publication Needed' for that particular customer only.

**To withdraw an Item-GTIN from a customer:**

The withdraw action allows you to temporarily discontinue an Item-GTIN synchronization process with one or more customers. Once you have successfully withdrawn the top Item-GTIN from a customer, the Item-GTIN is not published to the customer until you explicitly re-publish the Item-GTIN to this customer.

   **Note:** You can only withdraw Item-GTINs whose publication status is 'Published' and disposition status is 'Not Rejected'.

1. From the top Item-GTIN page, navigate to the Customer Syndication page. This page displays the list of customers to whom you have already published or plan to publish the Item-GTIN packaging hierarchy.

2. For the customers that you have already published to, select those from you which you want to withdraw the Item-GTIN, and click Withdraw.

3. The Withdraw from Customers page allows you to view the entire Item-GTIN packaging hierarchy and the customers from whom you are withdrawing the Item-GTIN.
4. A CIN message is sent to each customer from whom the Item-GTIN is being withdrawn. A message group id is generated to track all messages sent in this transaction.

**To delist an Item-GTIN from customers:**
The delist action permanently discontinues the synchronization process with all customers for this Item-GTIN. Once you successfully delist the top Item-GTIN, Product Data Synchronization for GDSN and 1SYNC Services does not allow you to publish this Item-GTIN to any customer.

*Note:* You can only delist the Item-GTINS whose publication status is published and disposition status is not rejected.

1. From the top Item-GTIN page, navigate to the Customer Syndication page. This page displays the list of customers to whom you have already published or plan to publish the Item-GTIN packaging hierarchy.

2. Click Delist from the customer syndication page.

3. All customers to whom the Item-GTIN was previously published will be delisted.

4. A CIN message is sent to each customer from whom the Item-GTIN is being delisted. A message group id is generated to track all messages sent in this transaction.

**To publish multiple Item-GTINS to customers:**
The publication workbench allows you to view multiple Item-GTINS for a customer. Similar to the registration workbench, you can search for Item-GTINS that are associated with the customer.

*Note:* You can only search for top Item-GTINS in the publication workbench.

You can Publish, Withdraw, and Delist Item-GTINS from this workbench. These actions are described in the section To publish an Item-GTIN to customers, page 12-20.

1. On the Publication Management page, select the Item-GTINS you want to publish and click Publish.

2. On the Publish to Customers page, a list of all the top Item-GTINS you have selected is displayed with its entire packaging hierarchy.

3. Review the list of customers per each Item-GTIN packaging hierarchy.
   You can also select Item-GTINS for Withdraw and Delist.
On the publication workbench, you can select top Item-GTINs and click Check Data Validation to see whether the Item-GTIN packaging hierarchy you plan to publish is compliant.

**To publish an Item-GTIN to customers with Initial load action:**

If a supplier who has recently joined a data pool, such as 1SYNC, wants to establish item synchronization with an existing Demand Side customer through the data pool, they can use the Initial Item load publication to synchronize data for items currently being traded between the two organizations.

1. On the Publication Management page, find the Items you want to publish.

2. Select the Items you want to publish with the Initial load action, click Initial load.

3. On the Publish (Initial Load) to Customers page, a list of all the top Item-GTINS you selected is displayed with their packaging hierarchy.

4. Review the list of customers for each Item-GTIN packaging hierarchy.

   **Note:** Initial load action is only performed for the first time publication with the customer.

**Related Topics**

Managing the Packaging Hierarchy, *Oracle Product Lifecycle Management User’s Guide*
This appendix covers the following topics:

- Workflows for Change and Document Management

### Workflows for Change and Document Management

The system uses Oracle Workflow technology to automate business processes in Change Management, Document Management and Collaboration (DOM), and Group Member registration. You must set up Oracle Workflow to implement Oracle Product Lifecycle Management or Oracle Product Information Management Data Librarian.

**Important:** Oracle Product Lifecycle Management (PLM) and Oracle Product Information Management Data Librarian (PIMDL) share a common technological foundation, but address different business needs through a different mix of core functions, plus specific functions that are unique to each product. Document Management and Collaboration (DOM) is available only to customers who have licensed Product Lifecycle Management, and it is not available to licensees of Product Information Management Data Librarian. Change Management is a core function, available to licensees of both products.

**Note:** Run wfver.sql under $FND_TOP/sql for detailed information about workflow versions.

### Workflow Setup

Refer to the Oracle Workflow Implementation and User Documentation required for setup.
Overview of Change and Document Management Workflows

Change and Document Management use Oracle Workflow technology to execute workflow routing for a change and document objects and events occurring in a change or document object (such as Submit, Reassign, Priority Change, and Status Change actions). Change and Document Management enable you to model your business processes in terms of generic order processes. So you can model your business processes by customizing or extending seeded workflow processes and/or modifying custom hook PL/SQL procedures.

Viewing Processes in Oracle Workflow Builder

From within the Oracle Workflow Builder you can view the different workflow processes and their associated functions, messages, subprocesses, notifications, and properties.

To view processes in Oracle Workflow Builder:

1. Within Oracle Workflow Builder, select Open from the File menu and connect to the database. Alternatively, you can connect to the workflow definitions file engchgmt.wft located in the product directory tree of your Oracle Applications server.

2. Expand the data source, and then select the item type branch within that data source.

3. Expand the processes branch within your item type and select a process activity to display the diagram of the process in a Process window.

<table>
<thead>
<tr>
<th>Display Name (Internal Name)</th>
<th>Purpose</th>
<th>Location of Flat File Store on Server (Location of other Flat File Stores with Item Type embedded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENG: Standard(ENGWFSTD)</td>
<td>Contains a collection of commonly used (by other Eng Workflow Item Types) Function Activities and Lookup Types. If the activity will be shared by more than one object's process, create the activity in the ENG: Standard (ENGWFSTD) item type. Otherwise, create the activity in the appropriate object's item type</td>
<td>$ENG_TOP/patch/115/import //engchmgt.wft</td>
</tr>
<tr>
<td>Process Internal Name</td>
<td>Process Display Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INITIATE_CHANGE</td>
<td>Initiate Change</td>
<td>Sends an assignment notification to assignee and starts &quot;Initiate Change Line&quot; workflow if Change has a line. This workflow is started when users submit a Change Object.</td>
</tr>
<tr>
<td>PRIORITY_CHANGE</td>
<td>Priority Change</td>
<td>Sends a priority change notification to Requestor, Owner, and Assignee for changes. This workflow is started when users change the priority for a Change Object.</td>
</tr>
<tr>
<td>Process Internal Name</td>
<td>Process Display Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>REASSIGN_CHANGE</td>
<td>Reassign Change</td>
<td>Sends an assignment notification to assignee. This workflow is started when users reassign a Change Object.</td>
</tr>
<tr>
<td>REQUEST_COMMENT</td>
<td>Request Comment</td>
<td>Sends a notification to request comment to specified people on the Request Comment Action page. When the recipient responds to the request notification, Workflow sends a response notification to the requestor. This workflow is started when users request comment for a Change Object.</td>
</tr>
<tr>
<td>RESPONSE_FYI</td>
<td>Send Response FYI</td>
<td>This process is called when the recipient responds to the request notification in Request Comment workflow process.</td>
</tr>
<tr>
<td>STATUS_CHANGE</td>
<td>Status Change</td>
<td>Sends a status change notification to Requestor, Owner, and Assignee for a change. This workflow is started when users change the status for a Change Object.</td>
</tr>
</tbody>
</table>

Change Line (ENGCLACT)

<table>
<thead>
<tr>
<th>Process Internal Name</th>
<th>Process Display Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITIATE_CHANGE</td>
<td>Initiate Change</td>
<td>Sends an assignment notification to assignee. This workflow is started from the Change Header’s Initiate Change Workflow or when a user creates a Change Line if Change Header is Open.</td>
</tr>
</tbody>
</table>
REASSIGN_CHANGE  Reassign Change  Sends an assignment notification to the assignee. This workflow is started when a user reassigns a Change Line.

STATUS_CHANGE  Status Change  Sends a status change notification to Header Requestor, Header Owner, Header Assignee, Line Owner and Line Assignee. This workflow is started when a user changes status for a Change Line.

### Change Workflow Routing (ENGCRT)

<table>
<thead>
<tr>
<th>Process Internal Name</th>
<th>Process Display Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUTE_AGENT</td>
<td>Workflow Routing Agent</td>
<td>Controls Workflow Routing and sends Approval Status change notification to Requestor, Owner, and Assignee for a Change. This workflow is started when a user submits an Workflow Routing for a Change Object. Also this workflow is aborted when user aborts an Workflow Routing.</td>
</tr>
</tbody>
</table>

### Change Workflow Routing Step (ENGCSTEP)

<table>
<thead>
<tr>
<th>Process Internal Name</th>
<th>Process Display Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUEST_DEFINITION</td>
<td>Definition</td>
<td>Sends a definition request notification to specified assignees on Workflow Routing page and controls definition request for New Item Request. If the assignees do not respond by the specified period, workflow will also send a reminder notification. Once the step is completed, workflow will return control parent Workflow Routing Agent workflow. This workflow is started from Workflow Routing Agent based on the definition of Workflow Routing.</td>
</tr>
<tr>
<td>REQUEST_DEFINITION_AND_APPR</td>
<td>Definition and Approval</td>
<td>Sends a definition and approval request notification to specified assignees on Workflow Routing page and controls definition and approval request for New Item Request. If the assignees do not respond by the specified period, workflow will also send a reminder notification. Once the step is completed, workflow will return control parent Workflow Routing Agent workflow. This workflow is started from Workflow Routing Agent based on the definition of Workflow Routing.</td>
</tr>
<tr>
<td>WORKFLOW TYPE</td>
<td>WORKFLOW DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>REQUEST_APPROVAL</td>
<td>Sends an approval request notification to specified assignees on Workflow Routing page and controls approval request. If the assignees do not respond by the specified period, workflow will also send a reminder notification. Once the step is approved, workflow will return control parent Workflow Routing Agent workflow. This workflow is started from Workflow Routing Agent based on the definition of Workflow Routing.</td>
<td></td>
</tr>
<tr>
<td>REQUEST_COMMENT</td>
<td>Sends a comment request notification to specified assignees on Workflow Routing page and controls comment requests. If the assignees do not respond by the specified period, workflow will also send a reminder notification. Once the step is completed, workflow will return control parent Workflow Routing Agent workflow. This workflow is started from Workflow Routing Agent based on the definition of Workflow Routing.</td>
<td></td>
</tr>
<tr>
<td>SEND_FYI</td>
<td>Sends an FYI notification to specified assignees on Workflow Routing page. Once the step is completed, workflow will return control parent Workflow Routing Agent workflow. This workflow is started from Workflow Routing Agent based on the definition of Workflow Routing.</td>
<td></td>
</tr>
</tbody>
</table>
Following are two Process Attributes for the Workflow Routing Step workflow process:

1. **Default Change Role Instance Set (DEFAULT_CHANGE_ROLE)**
   Assigned on current or last Workflow Routing. Implicit grants securely based on the steps workflow process.
   
   ENG_CHANGE_WF_APPROVERS -> Granted Role: Approver
   (ENG_CHANGE_APPROVER)
   
   ENG_CHANGE_WF_REVIEWERS -> Granted Role: Reviewer
   (ENG_CHANGE_REVIEWER).
   
   Implicit Role (Instance Set) to be granted is defined in the Step workflow process attributes
   
   DEFAULT_CHANGE_ROLE
   
   In case of seeded Step Workflow Process
   
   Request Approval (DEFAULT_CHANGE_ROLE: "ENG_CHANGE_WF_APPROVERS")
   
   The step assignees get "Approver" implicit role from instance set
   
   ENG_CHANGE_WF_APPROVERS.
   
   Request Comment & FYU (DEFAULT_CHANGE_ROLE: "ENG_CHANGE_WF_REVIEWERS")
   
   The step assignees get "Reviewer" implicit role from instance set
   
   ENG_CHANGE_WF_REVIEWERS. Oracle Workflow does not support other roles by default.

2. **Activity Condition Code (ACTIVITY_CONDITION_CODE)**
   Attribute will indicate whether or not the Condition radio box is displayed on the Workflow Routing Create/Update step page. Acceptable values are Y or N.

3. **Enable Flag in Type: Definition (DEFINITION)**
   Attribute will indicate whether or not the this process is displayed in the Workflow Process pull-down list based on the Definition Type of Workflow Routing (Template) on the Workflow Routing Create/Update step page. Acceptable values are Y or N.

4. **Enable Flag in Type: Approval (APPROVAL)**
   Attribute will indicate whether or not the this process is displayed in the Workflow Process pull-down list based on the Approval Type of Workflow Routing (Template) on the Workflow Routing Create/Update step page. Acceptable values are Y or N.

5. **Enable Flag in Type: Definition and Approval (DEFINITION_APPROVAL)**
   Attribute will indicate whether or not the this process is displayed in the Workflow
Process pull-down list based on the Definition and Approval Type of Workflow Routing (Template) on the Workflow Routing Create/Update step page. Acceptable values are Y or N.

6. **Enable Flag in Type: Generic (GENERIC)**
   
   Attribute will indicate whether or not the this process is displayed in the Workflow Process pull-down list based on the Generic Type of Workflow Routing (Template) on the Workflow Routing Create/Update step page. Acceptable values are Y or N.

7. **Association Enabled Flag (ASSOC_ENABLED)**
   
   Attribute will indicate whether or not the the association for the assignee is available in the Workflow Routing Step for New Item Request on the Create/Update step page. Acceptable values are Y or N. This attribute is only applicable for "Definition" and "Definition and Approval" Workflow.

### Seeded Workflow Item Attributes

The following table describes the common item attributes for Change and Document Management Workflow. Each Workflow Item Type should have these attributes.

**All Item Type**

<table>
<thead>
<tr>
<th>Internal Name</th>
<th>Display Name</th>
<th>Description</th>
<th>Type</th>
<th>Length</th>
<th>Misc info (used for etc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE_ID</td>
<td>Change Id</td>
<td>Engineering Change</td>
<td>Number</td>
<td></td>
<td>Primary Key Values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change Unique Identifier</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHANGE_NOTICE</td>
<td>Change Number</td>
<td>Engineering Change</td>
<td>Text</td>
<td>30</td>
<td>Old Primary Key Values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHANGE_NAME</td>
<td>Change Name</td>
<td>Engineering Change</td>
<td>Text</td>
<td>240</td>
<td>Token replacement for messages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORGANIZATION_ID</td>
<td>Organization Id</td>
<td>Organization Id</td>
<td>Number</td>
<td></td>
<td>Old Primary Key Values</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
<td>Data Type</td>
<td>Length</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------</td>
<td>-----------</td>
<td>--------</td>
<td>--------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>CHANGE_MANAGEMENT_TYPE</td>
<td>Change Management Type</td>
<td>Text</td>
<td>40</td>
<td>Token replacement for messages.</td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>ECO Description</td>
<td>Text</td>
<td>2000</td>
<td>Token replacement for messages.</td>
<td></td>
</tr>
<tr>
<td>STATUS</td>
<td>Status</td>
<td>Text</td>
<td>80</td>
<td>Token replacement for messages.</td>
<td></td>
</tr>
<tr>
<td>APPROVAL_STATUS</td>
<td>Approval Status</td>
<td>Text</td>
<td>80</td>
<td>Token replacement for messages.</td>
<td></td>
</tr>
<tr>
<td>PRIORITY</td>
<td>Priority</td>
<td>Text</td>
<td>50</td>
<td>Token replacement for messages.</td>
<td></td>
</tr>
<tr>
<td>REASON</td>
<td>Reason</td>
<td>Text</td>
<td>50</td>
<td>Token replacement for messages.</td>
<td></td>
</tr>
<tr>
<td>ASSIGNEE_NAME</td>
<td>Assignee Name</td>
<td>Text</td>
<td>360</td>
<td>Token replacement for messages.</td>
<td></td>
</tr>
<tr>
<td>ASSIGNEE_COMPANY</td>
<td>Assignee Company</td>
<td>Text</td>
<td>360</td>
<td>Token replacement for messages.</td>
<td></td>
</tr>
<tr>
<td>WF_USER_ROLE</td>
<td>WF Owner Role</td>
<td>Role</td>
<td></td>
<td>Runtime information for the work item</td>
<td></td>
</tr>
<tr>
<td>WF_USER_ID</td>
<td>WF Owner User Id</td>
<td>NUMBER</td>
<td></td>
<td>Runtime information for the work item</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td>Value</td>
<td>Note</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------</td>
<td>---------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOST_URL</td>
<td>Host URL and port</td>
<td></td>
<td>Runtime information for the work item</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFAULT_STYLE_SHEET</td>
<td>Default Style Sheet</td>
<td>Text</td>
<td>/OA_HTML/cabo/styles/bla\n</td>
<td>Temporary placeholders for HTML notifications</td>
<td></td>
</tr>
<tr>
<td>CHANGE_DETAIL_PAGE_URL</td>
<td>Change Detail Page URL</td>
<td>URL</td>
<td>Temporary placeholders for notifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MESSAGE_TEXT_BODY</td>
<td>Message Text Body</td>
<td>Document</td>
<td>Temporary placeholders for notifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MESSAGE_HTML_BODY</td>
<td>Message HTML Body</td>
<td>Document</td>
<td>Temporary placeholders for notifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REVIEWERS_ROLE</td>
<td>Change Standard Reviewer Role</td>
<td>Role</td>
<td>Temporary placeholders for notifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASSIGNEE_ROLE</td>
<td>Assignee Role</td>
<td>Role</td>
<td>Temporary placeholders for notifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTION_ID</td>
<td>Change Action Id</td>
<td>Change Action Id</td>
<td>Number</td>
<td>Primary Key for Change Actions (ENG_CHANGE_ACTION_S.ACTION_ID)</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------</td>
<td>------------------</td>
<td>--------</td>
<td>---------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>ADHOC_Party List</td>
<td>Adhoc Party List</td>
<td>Adhoc Party List</td>
<td>Text</td>
<td>Runtime information for the work item</td>
<td></td>
</tr>
<tr>
<td>ADHOC_Party Role</td>
<td>Adhoc Party Role</td>
<td>Adhoc Party Role</td>
<td>Role</td>
<td>Temporary placeholders for notifications</td>
<td></td>
</tr>
<tr>
<td>RESPONSE_TIMEOUT</td>
<td>Response Timeout</td>
<td>Response Timeout</td>
<td>Number</td>
<td>Timeout in Minutes for Action calculated from Response_By_Date</td>
<td></td>
</tr>
<tr>
<td>RESPONSE_COMMENT</td>
<td>Response Comment</td>
<td>Response Comment</td>
<td>Text</td>
<td>Temporary placeholders for notifications</td>
<td></td>
</tr>
</tbody>
</table>
| ATTACHMENTS     | Attachments      | Attachments      | Document | Temporary placeholders for notifications: Notification Detail Page Attachment LinkFND:entity==
<table>
<thead>
<tr>
<th>Workflow ID</th>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WF_USER_IDD</td>
<td>WF Owner</td>
<td>User Id of Person Who Started/Aborted the Workflow</td>
</tr>
<tr>
<td>WF_USER_ROLE</td>
<td>WF Owner</td>
<td>Role of Person Who Started/Aborted the Workflow</td>
</tr>
<tr>
<td>WF_ADMINISTRATOR</td>
<td>WF Administrator</td>
<td>Role</td>
</tr>
<tr>
<td>WF_SIG_POLICY</td>
<td>Signature Policy</td>
<td>Signature Policy Text To use Signature Policy defaulting in future</td>
</tr>
<tr>
<td>ERROR_TIMEOUT</td>
<td>WF Error Timeout Value</td>
<td>Timeout Value in Days Number Constant:</td>
</tr>
<tr>
<td>RCS</td>
<td>RCS version</td>
<td>RCS version Text 240 RCS version</td>
</tr>
<tr>
<td>FROM_ROLE</td>
<td>From Role</td>
<td>From Role Role Item Attribute which holds the From Role. Message #FROM_ROLE</td>
</tr>
</tbody>
</table>
ENGCLACT Specific Item Attributes (required attributes when you pass the change line ID to StartWorkflow API)

<table>
<thead>
<tr>
<th>Internal Name</th>
<th>Display Name</th>
<th>Description</th>
<th>Type</th>
<th>Length</th>
<th>Misc info (used for etc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE_LINE_ID</td>
<td>Change Line Id</td>
<td>Engineering Change Line Unique Identifier</td>
<td>Number</td>
<td></td>
<td>Primary Key Values</td>
</tr>
<tr>
<td>LINE_NAME</td>
<td>Change Line Name</td>
<td>Engineering Change Line Name</td>
<td>Text</td>
<td>240</td>
<td>Token replacement for messages.</td>
</tr>
<tr>
<td>LINE_DESCRIPTION</td>
<td>Change Line Description</td>
<td>Change Line Description</td>
<td>Text</td>
<td>4000</td>
<td>Token replacement for messages.</td>
</tr>
<tr>
<td>LINE_SEQUENCE_NUMBER</td>
<td>Change Line Sequence Number</td>
<td>Change Line Sequence Number</td>
<td>Number</td>
<td></td>
<td>Token replacement for messages.</td>
</tr>
<tr>
<td>LINE_STATUS</td>
<td>Change Line Status</td>
<td>Change Line Status</td>
<td>Text</td>
<td>80</td>
<td>Token replacement for messages.</td>
</tr>
<tr>
<td>LINE_ASSIGNEE_NAME</td>
<td>Change Line Assignee Name</td>
<td>Change Line Assignee Name</td>
<td>Text</td>
<td>360</td>
<td>Token replacement for messages.</td>
</tr>
<tr>
<td>LINE_ASSIGNEE_COMPANY</td>
<td>Change Line Assignee Company</td>
<td>Change Line Assignee Company</td>
<td>Text</td>
<td>360</td>
<td>Token replacement for messages.</td>
</tr>
<tr>
<td>LINE_REVIEWERS_ROLE</td>
<td>Change Line Reviewer Role</td>
<td>Change Line Reviewer Role</td>
<td>Role</td>
<td></td>
<td>Temporary placeholders for notifications</td>
</tr>
<tr>
<td>LINE_ASSIGNEE_ROLE</td>
<td>Change Line Assignee Role</td>
<td>Change Line Assignee Role</td>
<td>Role</td>
<td></td>
<td>Temporary placeholders for notifications</td>
</tr>
<tr>
<td>Internal Name</td>
<td>Display Name</td>
<td>Description</td>
<td>Type</td>
<td>Length</td>
<td>Misc info (used for etc)</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>-------------</td>
<td>----------</td>
<td>--------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>ROUTE_ID</td>
<td>Route Id</td>
<td>Route Id</td>
<td>Number</td>
<td></td>
<td>Primary Key Values for Change Route</td>
</tr>
<tr>
<td></td>
<td>(Workflow Routing Id)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STEP_ID</td>
<td>Current Step Id</td>
<td>Current Step Id</td>
<td>Number</td>
<td></td>
<td>Primary Key Values for Current Step Id</td>
</tr>
<tr>
<td></td>
<td>(Workflow Routing Step Id)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROUTE_PEOPEOPLE_ROLE</td>
<td>Route People Role</td>
<td>All people in the Workflow Routing workflow that have already been notified (i.e. completed or in process steps)</td>
<td>Role</td>
<td></td>
<td>Temporary placeholders for notifications</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Internal Name</th>
<th>Display Name</th>
<th>Description</th>
<th>Type</th>
<th>Length</th>
<th>Misc info (used for etc)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ENGCRT Specific Item Attributes

ENGCSTEP Specific Item Attributes
<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUTE_ID</td>
<td>Workflow Routing Id</td>
</tr>
<tr>
<td></td>
<td>Workflow Routing Id</td>
</tr>
<tr>
<td></td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>Primary Key Values for Change Route</td>
</tr>
<tr>
<td>STEP_ID</td>
<td>Workflow Routing Step Id</td>
</tr>
<tr>
<td></td>
<td>Current Step Id</td>
</tr>
<tr>
<td></td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>Primary Key Values for Step Id</td>
</tr>
<tr>
<td>ROUTE_PEO_PLE_ROLE</td>
<td>Workflow Routing People Role</td>
</tr>
<tr>
<td></td>
<td>All people in the Workflow Routing workflow that have already been notified (i.e. completed or in process steps)</td>
</tr>
<tr>
<td></td>
<td>Role</td>
</tr>
<tr>
<td></td>
<td>Temporary placeholders for notifications</td>
</tr>
<tr>
<td>STEP_SEQ_N UMBER</td>
<td>Workflow Routing Step Sequence Number</td>
</tr>
<tr>
<td></td>
<td>Workflow Routing Step Sequence Number</td>
</tr>
<tr>
<td></td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>Temporary placeholders for notifications</td>
</tr>
<tr>
<td>STEP_PEO_PLE_ROLE</td>
<td>Workflow Routing Step People Role</td>
</tr>
<tr>
<td></td>
<td>Workflow Routing Step People Role</td>
</tr>
<tr>
<td></td>
<td>Role</td>
</tr>
<tr>
<td></td>
<td>Temporary placeholders for notifications</td>
</tr>
<tr>
<td>STEP_CONDITION</td>
<td>Workflow Routing Step Activity Condition</td>
</tr>
<tr>
<td></td>
<td>Workflow Routing Step Activity Condition</td>
</tr>
<tr>
<td></td>
<td>Text 30</td>
</tr>
<tr>
<td></td>
<td>Temporary placeholders for notifications</td>
</tr>
<tr>
<td>STEP_VOTING_OPTION</td>
<td>Workflow Routing Step Voting Option for Activity Condition</td>
</tr>
<tr>
<td></td>
<td>Workflow Routing Step Voting Option for Activity Condition</td>
</tr>
<tr>
<td></td>
<td>Lookup</td>
</tr>
<tr>
<td></td>
<td>Standard Voting Option</td>
</tr>
<tr>
<td>YES_RESPONSE_PERCENT</td>
<td>Yes Response Percentage of Votes</td>
</tr>
<tr>
<td></td>
<td>Yes Response Percentage of Votes</td>
</tr>
<tr>
<td></td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>Temporary placeholders for notifications</td>
</tr>
</tbody>
</table>
## Change Management Workflow PL/SQL Packages

Change Management Workflow Private APIs will contain the following list of packages. These are private APIs for reference purposes only for your customization.

<table>
<thead>
<tr>
<th>Package Name</th>
<th>File Name</th>
<th>Description</th>
<th>Location of Flat File Store on Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENG_WORKFLOW_PUB</td>
<td>ENGBWKFS.pls</td>
<td>Private APIs: for developers or consultants at Oracle: All PL/SQL Function for Workflow Function Activity.</td>
<td>$ENG_TOP/patch/115/sql</td>
</tr>
<tr>
<td></td>
<td>ENGBWKFB.pls</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENG_WORKFLOWUTIL</td>
<td>ENGUKFS.pls</td>
<td>Private APIs: This package contains utility functions and procedures for Eng Workflows</td>
<td>$ENG_TOP/patch/115/sql</td>
</tr>
<tr>
<td></td>
<td>ENGUKFS.pls</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENG_WORKFLOWNTFUTIL</td>
<td>ENGUNTFS.pls</td>
<td>This package contains utility functions and procedures for Eng Workflow Notifications</td>
<td>$ENG_TOP/patch/115/sql</td>
</tr>
<tr>
<td></td>
<td>ENGUNTFS.pls</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Package</td>
<td>File Name</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>ENG_WORKFLOW_EXT</td>
<td>ENGXWKFS.pls</td>
<td>This package contains procedures that serve as &quot;hooks&quot; into the Eng Workflows for customization of the workflows. The &quot;StartCustomWorkflow&quot; and the &quot;AbortCustomWorkflow&quot; are called from ENG_WORKFLOW_UTIL package when a workflow process is started and aborted, respectively. Users with customized workflows can add their custom logic in these two procedures to be executed at the appropriate times. Also GetCustomMessageBody is called from the seeded PL/SQL Document APIs when system generates notification. Users can customize the PL/SQL procedure to modify notification contents. Additionally, users can create new procedures in this package for custom workflow activities.</td>
<td></td>
</tr>
<tr>
<td>ENG_WORKFLOW_EXT</td>
<td>ENGXWKFB.pls</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$ENG_TOP/patch/115/sql
This appendix covers the following topics:

- Business Events

### Business Events

Using Oracle Workflow, you can automatically trigger actions based on the following item business events. Examples of actions you can trigger include issuing notifications and launching a workflow.

The following table lists item and item revision events, as well as a description and the event parameters of each event.

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Event Description</th>
<th>Event Parameter Names and Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>oracle.apps.ego.item.postAML.Change</td>
<td>Item AML is changed (such as a manufacturer part number is added, updated, or deleted)</td>
<td>DML_TYPE: CREATE, UPDATE, DELETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INVENTORY_ITEM_ID: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_ID: Organization ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MANUFACTURER_ID: Manufacturer ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MFG_PART_NUM: Manufacturer Part Number</td>
</tr>
<tr>
<td>Event Name</td>
<td>Event Description</td>
<td>Event Parameter Names and Descriptions</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>oracle.apps.ego.item.postCatalogCategoryChange</td>
<td>Catalog category is changed such as when categories are created, updated or deleted</td>
<td>DML_TYPE: CREATE, UPDATE, DELETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CATEGORY_NAME: Category Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CATEGORY_ID: Category ID</td>
</tr>
<tr>
<td>oracle.apps.ego.item.postValidCategoryChange</td>
<td>Category is added to or is removed from a catalog. This event is also raised when the attributes of catalog are changed.</td>
<td>DML_TYPE: CREATE, UPDATE, DELETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CATEGORY_SET_ID: Category Set ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CATEGORY_ID: Category ID</td>
</tr>
<tr>
<td>oracle.apps.ego.item.postItemCreate</td>
<td>Item Creation</td>
<td>INVENTORY_ITEM_ID: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ITEM_NUMBER: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_ID: Organization ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_CODE: Organization Code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ITEM_DESCRIPTION: Item Description</td>
</tr>
<tr>
<td>oracle.apps.ego.item.postItemUpdate</td>
<td>Item Update</td>
<td>INVENTORY_ITEM_ID: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ITEM_NUMBER: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_ID: Organization ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_CODE: Organization Code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ITEM_DESCRIPTION: Item Description</td>
</tr>
<tr>
<td>Event Name</td>
<td>Event Description</td>
<td>Event Parameter Names and Descriptions</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>----------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>oracle.apps.ego.item.postGTI NAttributeChange</td>
<td>GTIN Attribute group row is created or updated.</td>
<td>DML_TYPE: CREATE, UPDATE, DELETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATTR_GROUP_NAME: Attribute Group Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXTENSION_ID: Extension Id that uniquely identifies the record in the UDA extension table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INVENTORY_ITEM_ID: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_ID: Organization ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REVISION_ID: Revision ID</td>
</tr>
<tr>
<td>oracle.apps.ego.item.postItem Approved</td>
<td>On Item Approval</td>
<td>INVENTORY_ITEM_ID: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_ID: Organization ID</td>
</tr>
<tr>
<td>oracle.apps.ego.item.postItem Bulkload</td>
<td>Bulk Item processing (creation, update)</td>
<td>REQUEST_ID: Concurrent Request ID that processed the bulk item import</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>oracle.apps.ego.item.postXref Change</td>
<td>Item cross-reference creation, update, or deletion</td>
<td>DML_TYPE: CREATE, UPDATE, DELETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INVENTORY_ITEM_ID: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_ID: Organization ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CROSSREFERENCE_TYPE: Cross-reference Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CROSS_REFERENCE: Cross-reference</td>
</tr>
<tr>
<td>Event Name</td>
<td>Event Description</td>
<td>Event Parameter Names and Descriptions</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>oracle.apps.ego.item.postRevisionChange</td>
<td>Item revision is created or updated</td>
<td>DML_TYPE: CREATE, UPDATE, DELETE</td>
</tr>
<tr>
<td></td>
<td>Applicable: SSWA, FORMS, IOI, XL import, API</td>
<td>INVENTORY_ITEM_ID: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_ID: Organization ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REVISION_ID: Revision ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQUEST_ID: Concurrent Request ID</td>
</tr>
<tr>
<td>oracle.apps.ego.item.postItemRoleChange</td>
<td>An item role grant is changed (added, modified, or removed)</td>
<td>DML_TYPE: CREATE, UPDATE, DELETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INVENTORY_ITEM_ID: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_ID: Organization ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROLE_ID: Role ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PARTY_TYPE: Party Type (Person, Group or Company)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PARTY_ID: Party ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>START_DATE: Start Date</td>
</tr>
<tr>
<td>oracle.apps.ego.item.postCatalogAssignmentChange</td>
<td>Item Catalog assignment or removal</td>
<td>DML_TYPE: CREATE, UPDATE, DELETE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INVENTORY_ITEM_ID: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_ID: Organization ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CATALOG_ID: Catalog ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CATEGORY_ID: Category ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQUEST_ID: Concurrent Request ID</td>
</tr>
<tr>
<td>Event Name</td>
<td>Event Description</td>
<td>Event Parameter Names and Descriptions</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>oracle.apps.ego.item.preAttri buteChange</td>
<td>Item Attribute group row is created, updated, or deleted.</td>
<td>DML_TYPE: CREATE, UPDATE, DELETE</td>
</tr>
<tr>
<td></td>
<td>This event is raised before the change is committed.</td>
<td>ATTR_GRP_NAME:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Attribute Group Name</td>
</tr>
<tr>
<td></td>
<td>Applicable: SSWA</td>
<td>EXTENSION_ID:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extension ID that uniquely identifies the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>record in the UDA extension table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INVENTORY_ITEM_ID: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_ID: Organization ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REVISION_ID: Revision ID</td>
</tr>
<tr>
<td>oracle.apps.ego.item.postAttri buteChange</td>
<td>Item Attribute group row is created, updated, or deleted.</td>
<td>DML_TYPE: CREATE, UPDATE, DELETE</td>
</tr>
<tr>
<td></td>
<td>This event is raised after the change is committed.</td>
<td>ATTR_GRP_NAME:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Attribute Group Name</td>
</tr>
<tr>
<td></td>
<td>Applicable: SSWA</td>
<td>EXTENSION_ID:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extension ID that uniquely identifies the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>record in the UDA extension table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INVENTORY_ITEM_ID: Item ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ORGANIZATION_ID: Organization ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REVISION_ID: Revision ID</td>
</tr>
</tbody>
</table>

**Common Event Parameters**

<table>
<thead>
<tr>
<th>Event Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event_Type</td>
<td>This parameter identifies the type of transaction that raised the event. The values are Single or Bulk.</td>
</tr>
</tbody>
</table>
## Event Parameter Name

<table>
<thead>
<tr>
<th>Event Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last_Updated_By</td>
<td>Identifier of the user.</td>
</tr>
<tr>
<td>Last_Update_Date</td>
<td>Identifies the date the transaction occurred.</td>
</tr>
<tr>
<td>Created_By</td>
<td>Identifier of the user.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> In bulk events that include both creation and update transactions, this parameter is does not apply and is not available.</td>
</tr>
<tr>
<td>Creation_Date</td>
<td>Identifies the date the creation transaction occurred.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> In bulk events that include both creation and update transactions, this parameter is does not apply and is not available.</td>
</tr>
<tr>
<td>Request_Identifier</td>
<td>When Event_Type = Bulk, this identifies the concurrent request that processed the bulk data set.</td>
</tr>
</tbody>
</table>

## Related Topics


Reading Data for User-Defined Attributes

This appendix covers the following topics:

• Overview

Overview

You can define an unlimited number of attributes for items without programming them. You can define attributes using the setup user interface; definitions are stored in metadata tables. However, you may also need to read attribute data to interface with third party or custom systems.

Generate database views to read User-defined attributes data

Using the attribute group setup screens, you can generate database views for reading user-defined attributes data. These views hide the complexity of dealing with the attribute meta-data.

Using PL/SQL to Access Data for Attributes

Another way to access user-defined attribute data for a particular item is to use PL/SQL. The EGO_USER_ATTRS_DATA_PUB procedure follows:

PROCEDURE Get_User_Attrs_Data

<p>| p_api_version            | IN NUMBER                        |
| p_object_name            | IN VARCHAR2                      |
| p_pk_column_name_value_pairs | IN EGO_COL_NAME_VALUE_PAIR_ARRAY |
| p_attr_group_request_table | IN EGO_ATTR_GROUP_REQUEST_TABLE   |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>p_user_privileges_on_object</code></td>
<td>IN EGO_VARCHAR_TBL_TYPE DEFAULT NULL</td>
</tr>
<tr>
<td><code>p_entity_id</code></td>
<td>IN VARCHAR2 DEFAULT NULL</td>
</tr>
<tr>
<td><code>p_entity_index</code></td>
<td>IN NUMBER DEFAULT NULL</td>
</tr>
<tr>
<td><code>p_entity_code</code></td>
<td>IN VARCHAR2 DEFAULT NULL</td>
</tr>
<tr>
<td><code>p_debug_level</code></td>
<td>IN NUMBER DEFAULT 0</td>
</tr>
<tr>
<td><code>p_init_error_handler</code></td>
<td>IN VARCHAR2 DEFAULT FND_API.G_FALSE</td>
</tr>
<tr>
<td><code>p_init_fnd_msg_list</code></td>
<td>IN VARCHAR2 DEFAULT FND_API.G_FALSE</td>
</tr>
<tr>
<td><code>p_add_errors_to_fnd_stack</code></td>
<td>IN VARCHAR2 DEFAULT FND_API.G_FALSE</td>
</tr>
<tr>
<td><code>p_commit</code></td>
<td>IN VARCHAR2 DEFAULT FND_API.G_FALSE</td>
</tr>
<tr>
<td><code>x_attributes_row_table</code></td>
<td>OUT NOCOPY EGO_USER_ATTR_ROW_TABLE</td>
</tr>
<tr>
<td><code>x_attributes_data_table</code></td>
<td>OUT NOCOPY EGO_USER_ATTR_DATA_TABLE</td>
</tr>
<tr>
<td><code>x_return_status</code></td>
<td>OUT NOCOPY VARCHAR2</td>
</tr>
<tr>
<td><code>x_errorcode</code></td>
<td>OUT NOCOPY NUMBER</td>
</tr>
<tr>
<td><code>x_msg_count</code></td>
<td>OUT NOCOPY NUMBER</td>
</tr>
<tr>
<td><code>x_msg_data</code></td>
<td>OUT NOCOPY VARCHAR2</td>
</tr>
</tbody>
</table>

**p_api_version**
Callers should pass a constant value of 1.0.

**p_object_name**
Callers should pass a constant value of 'EGO_ITEM'

**p_pk_column_name_value_pairs**
This is a table of EGO_COL_NAME_VALUE_PAIR_OBJ objects specifying the Inventory Item ID and Organization ID for which the caller wants attribute data. Callers should create the table with code like the following:

```sql
l_pk_column_values EGO_COL_NAME_VALUE_PAIR_ARRAY;
l_pk_column_values := EGO_COL_NAME_VALUE_PAIR_ARRAY(
    EGO_COL_NAME_VALUE_PAIR_OBJ('INVENTORY_ITEM_ID', <your inventory item ID>),
    EGO_COL_NAME_VALUE_PAIR_OBJ('ORGANIZATION_ID', <your organization ID>));
```

### p_attr_group_request_table

This is a table of EGO_ATTR_GROUP_REQUEST_OBJ objects, one for each attribute group whose values the caller wants. The objects also support requests for only some of the attributes in the attribute group. Each EGO_ATTR_GROUP_REQUEST_OBJ object in the table will look like the following:

```sql
EGO_ATTR_GROUP_REQUEST_OBJ(
    NULL,
    431,
    'EGO.ITEMMGMT_GROUP',
    <your Attribute Group internal name>,
    <your Revision ID if the Attribute Group is associated at the Revision level; otherwise NULL>,
    NULL,
    NULL
)
```

### p_user_privileges_on_object

This is an optional table of VARCHAR2s, one for each privilege the caller wants validated against the View privileges of the attribute groups being requested. For example, if a user invokes a procedure that calls Get_User_Attrs_Data, and the calling procedure wants to enforce attribute group data security, the calling procedure could pass a list of the user’s privileges based on his/her role on the current item. If this parameter is non-null, Get_User_Attrs_Data will enforce data security for those attribute groups that have a View privilege defined.

### p_entity_id, p_entity_index, p_entity_code

Parameters for use with ERROR_HANDLER package. Can safely be defaulted unless caller wants to distinguish among errors for multiple items; if so, please investigate the ERROR_HANDLER package for more details.

### p_debug_level
Parameter to control debug logging; can safely be defaulted.

**p_init_error_handler, p_init_fnd_msg_list, p_add_errors_to_fnd_stack**

More parameters for use with ERROR_HANDLER package; can safely be defaulted.

**p_commit**

Controls whether or not Get_User_Attrs_Data issues a COMMIT statement at the end of processing; present to comply with API standards, but since Get_User_Attrs_Data does not perform any DML, can safely be defaulted.

**x_attributes_row_table**

This is a table of EGO_USER_ATTR_ROW_OBJ objects, each one representing a row of user-defined attributes data (for example, one database table row of data). It contains the Extension ID value from EGO_MTL_SY_ITEMS_EXT_VL (stored as ROW_IDENTIFIER) and serves mainly as a way to organize the objects in x_attributes_data_table.

**x_attributes_data_table**

This is a table of EGO_USER_ATTR_DATA_OBJ objects, each one representing a single attribute value. All of the attribute values for a given row of data will share the same ROW_IDENTIFIER value; as with x_attributes_row_table, the ROW_IDENTIFIER value is the Extension ID from EGO_MTL_SY_ITEMS_EXT_VL. Each object also contains the attribute internal name (stored as ATTR_NAME) and its value (stored as ATTR_DISP_VALUE).

**x_return_status, x_errorcode, x_msg_count, x_msg_data**

Four standard OUT parameters indicating what happened in processing. The first, x_return_status, will equal FND_API.G_RET_STS_SUCCESS if processing succeeded, FND_API.G_RET_STS_ERROR if an expected error case was encountered, or FND_API.G_RET_STS_UNEXP_ERROR if something went wrong during processing. X_errorcode is not used. X_msg_count indicates how many error messages were logged with ERROR_HANDLER; if x_msg_count is 1, then x_msg_data contains that message (to save the caller the inconvenience of interacting with ERROR_HANDLER for only one message).
Synchronizing Item User-Defined Attributes with Item Descriptive Elements

This appendix covers the following topics:

- Overview

Overview

You can define an unlimited number of user-defined attributes for items without programming. These attributes can have different data types, value sets, and can be parametrically searched. This appendix describes how to synchronize user-defined attribute values with descriptive element values. Oracle provides a public API to update Item Descriptive Element values. It also raises a business event when any pre-selected user-defined attribute values change. You can subscribe to this event and then call the Item Descriptive Element API to synchronize the descriptive elements with the attribute values. A business event is raised when any pre-selected item user-defined attribute values change.


User-Defined Attributes Business Event

You can configure user-defined attribute groups to raise a Business Event every time a row is added, altered, or deleted. The name of the Business Event raised is oracle.apps.ego.item.attributesChanged

The parameters passed by this Business Event are

- INVENTORY_ITEM_ID
- ORGANIZATION_ID
- EXTENSION_ID
- ATTR_GROUP_NAME
The Event is disabled by default and must be explicitly enabled for synchronization to work.

**Item Descriptive Elements API**

Oracle provides the following API (with accompanying data type) to maintain Descriptive Elements for a given item.

The data type used by the API follows:

ITEM_DESC_ELEMENT_TABLE: this is an associative array (also known as a PL/SQL table or an index-by table) of ITEM_DESC_ELEMENT records:

```sql
ITEM_DESC_ELEMENT IS RECORD
  (ELEMENT_NAME VARCHAR2(30),
   ELEMENT_VALUE VARCHAR2(30),
   DESCRIPTION_DEFAULT VARCHAR2(1));
```

ELEMENT_NAME: Column ELEMENT_NAME from MTL_DESCR_ELEMENT_VALUES

ELEMENT_VALUE: Column ELEMENT_VALUE from MTL_DESCR_ELEMENT_VALUES

DESCRIPTION_DEFAULT: Column DEFAULT_ELEMENT_FLAG from MTL_DESCR_ELEMENT_VALUES (indicates whether this Descriptive Element will be used in the description generation)

The API signature is:

```sql
INV_ITEM_CATALOG_ELEM_PUB.Process_Item_Descr_Elements
  (
    p_api_version IN NUMBER,
    p_init_msg_list IN VARCHAR2 DEFAULT fnd_api.g_FALSE,
    p_commit_flag IN VARCHAR2 DEFAULT fnd_api.g_FALSE,
    p_validation_level IN NUMBER DEFAULT INV_ITEM_CATALOG_ELEM_PUB.g_VALIDATE_ALL,
    p_inventory_item_id IN NUMBER DEFAULT -999,
    p_item_number IN VARCHAR2 DEFAULT NULL,
    p_item_desc_element_table IN ITEM_DESC_ELEMENT_TABLE,
    x_generated_descr OUT VARCHAR2,
    x_return_status OUT VARCHAR2,
    x_msg_count OUT NUMBER,
    x_msg_data OUT VARCHAR2
  );
```

**p_api_version**

Pass the value 1.0 for this parameter.

**p_init_msg_list**

This parameter can safely be defaulted.

**p_commit_flag**

Indicates whether or not the API commits the transaction.

**p_validation_level**

Determines how much validation occurs; can safely be defaulted.
**p_inventory_item_id**
Pass the value 1.0 for this parameter.

**p_item_number**
This parameter can safely be defaulted.

**p_item_desc_element_table**
Data type defined above.

**x_generated_descr**
Returns the Item Description generated from the updated Descriptive Elements. If you want to update the Item Description with this value, use Item Open Interface.

**x_return_status**
Returns one of the following:
- `FND_API.G_RET_STS_SUCCESS`
- `FND_API.G_RET_STS_ERROR`
- `FND_API.G_RET_STS_UNEXP_ERROR`

**x_msg_count**
Number of messages logged by the API.

**x_msg_data**
The message logged if exactly one message was logged.

---

**Subscribing to the Event and Calling the API**

As briefly mentioned earlier, the Business Event `oracle.apps.ego.item.attributesChanged` must be explicitly enabled through the graphical user interface in order for synchronization to work. After this step is complete, you must subscribe to the User-Defined Attributes Business Event. See: Event Manager, *Oracle Workflow Developer’s Guide*.

As part of Event Subscription, define an Event Subscription Rule Function that will be called when the Event is raised. See Standard API for an Event Rule Function, *Oracle Workflow Developer’s Guide*.

This Rule Function should:

- Identify the attribute group that raised the Event and determine the correct Descriptive Elements for that attribute group (perhaps through customer-supplied mapping metadata).

- Fetch the data altered by the Event (see the following example Rule Function and also the White Paper "Reading Data for Item User-Defined Attributes" for more details) and process it as necessary.
• Call the Item Descriptive Elements API that is passing the processed data.
**Sample Rule Function**

```sql
FUNCTION Synch_Attrs_With_Desc_Elems (  
    p_subscription_guid IN RAW  
    ,p_event              IN OUT NOCOPY WF_EVENT_T  
)  
RETURN VARCHAR2  
IS  

    TYPE ATTR_TO_DESC_ELEM_MAPPING_REC IS RECORD  
    {  
        ATTR_NAME VARCHAR2(30)  
        ,DESC_ELEM_NAME VARCHAR2(30)  
        ,DESCRIPTION_DEFAULT VARCHAR2(1)  
    }  
    TYPE ATTR_TO_DESC_ELEM_MAPPING_TBL IS TABLE OF  
    ATTR_TO_DESC_ELEM_MAPPING_REC  
    INDEX BY BINARY_INTEGER;  

    l_parameter_list WF_PARAMETER_LIST_T;  
    l_next_parameter WF_PARAMETER_T;  
    l_dml_type_param WF_PARAMETER_T;  
    l_attr_group_name_param WF_PARAMETER_T;  
    l_inventory_item_id_param WF_PARAMETER_T;  
    l_organization_id_param WF_PARAMETER_T;  
    l_revision_id_param WF_PARAMETER_T;  
    l_pk_column_values EGO_COL_NAME_VALUE_PAIR_ARRAY;  
    l_request_table EGO_ATTR_GROUP_REQUEST_TABLE;  
    x_attributes_row_table EGO_USER_ATTR_ROW_TABLE;  
    x_attributes_data_table EGO_USER_ATTR_DATA_TABLE;  
    x_return_status VARCHAR2(1);  
    x_errorcode NUMBER;  
    x_msg_count NUMBER;  
    x_msg_data VARCHAR2(1000);  
    l_current_mapping_rec ATTR_TO_DESC_ELEM_MAPPING_REC;  
    l_mapping_rec_table ATTR_TO_DESC_ELEM_MAPPING_TBL;  
    l_attributes_data_index NUMBER;  
    l_current_row_obj EGO_USER_ATTR_ROW_OBJ;  
    l_current_data_obj EGO_USER_ATTR_DATA_OBJ;  
    l_current_item_desc_elem INV_ITEM_CATALOG_ELEM_PUB.ITEM_DESC_ELEMENT;  
    l_item_desc_elem_table  
    INV_ITEM_CATALOG_ELEM_PUB.ITEM_DESC_ELEMENT_TABLE;  
    x_generated_descr VARCHAR2(240);  
    l_return_status VARCHAR2(30) := 'SUCCESS';  

BEGIN  

    l_parameter_list := p_event.getParameterList();  
    --------------------------------------------------------  
    -- Sort the parameters that the Business Event passed  
    --------------------------------------------------------  
    FOR i IN l_parameter_list.FIRST .. l_parameter_list.LAST  
    LOOP
```

Synchronizing Item User-Defined Attributes with Item Descriptive Elements   D-5
l_next_parameter := l_parameter_list(i);
   IF (l_next_parameter.getName() = 'DML_TYPE') THEN
      l_dml_type_param := l_next_parameter;
   ELSIF (l_next_parameter.getName() = 'ATTR_GROUP_NAME') THEN
      l_attr_group_name_param := l_next_parameter;
   ELSIF (l_next_parameter.getName() = 'INVENTORY_ITEM_ID') THEN
      l_inventory_item_id_param := l_next_parameter;
   ELSIF (l_next_parameter.getName() = 'ORGANIZATION_ID') THEN
      l_organization_id_param := l_next_parameter;
   ELSIF (l_next_parameter.getName() = 'REVISION_ID') THEN
      l_revision_id_param := l_next_parameter;
   END IF;
END LOOP;

---------------------------------------------------------
Copy the PK values into our EGO_COL_NAME_VALUE_PAIR_ARRAY object
---------------------------------------------------------

l_pk_column_values :=
   EGO_COL_NAME_VALUE_PAIR_ARRAY(
      EGO_COL_NAME_VALUE_PAIR_OBJ('INVENTORY_ITEM_ID',
         l_inventory_item_id_param.getValue()),
      EGO_COL_NAME_VALUE_PAIR_OBJ('ORGANIZATION_ID',
         l_organization_id_param.getValue())
   );

---------------------------------------------------------------
Prepare an EGO_ATTR_GROUP_REQUEST_TABLE object with one element, an EGO_ATTR_GROUP_REQUEST_OBJ for the Attr Group that raised the Event
----------------------------------------------------------------

l_request_table := EGO_ATTR_GROUP_REQUEST_TABLE();
   l_request_table.EXTEND();
   l_request_table(l_request_table.LAST) :=
      EGO_ATTR_GROUP_REQUEST_OBJ(
         NULL --ATTR_GROUP_ID
         431,
         'EGO_ITEMMGT_GROUP',
         l_attr_group_name_param.getValue(),
         l_revision_id_param.getValue(),
         NULL --DATA_LEVEL_2
         NULL --DATA_LEVEL_3
         NULL --ATTR_NAME_LIST
      );

--------------------------------------------------
Get the data for this Event, if there is any
--------------------------------------------------
IF (l_dml_type_param.getValue() <> 'DELETE') THEN
   EGO_USER_ATTRS_DATA_PUB.Get_User_Attrs_Data(
      p_api_version => 1.0,
      p_object_name => 'EGO_ITEM',
      p_pk_column_name_value_pairs => l_pk_column_values,
      p[attr_group_request_table] => l_request_table,
      p_user_privileges_on_object => NULL,
      p_entity_id => NULL,
      p_entity_index => NULL,
      p_entity_code => NULL,
      p_debug_level => 0)
END IF;
Synchronizing Item User-Defined Attributes with Item Descriptive Elements

D-7

If the call failed, we report that an error occurred (in a production situation, we would retrieve and pass on all errors on the ERROR_HANDLER message stack)

```
IF (x_return_status IS NULL OR 
    x_return_status <> FND_API.G_RET_STS_SUCCESS) THEN 
    IF (x_return_status = FND_API.G_RET_STS_ERROR) THEN 
        RAISE FND_API.G_EXC_ERROR; 
    ELSE 
        RAISE FND_API.G_EXC_UNEXPECTED_ERROR; 
    END IF; 
END IF; 
END IF; 
END IF; 
```

Process the Descriptive Elements according to the Attr Group that raised the Event and the DML type

```
IF (l_attr_group_name_param.getValue() = 'Pella_Windows') THEN 
```

Find mappings from Attr Group Atts to Desc Elems; in this example, hard code the mapping from Attr Group Atts to Desc Elems, but in a production situation customers may have some stored mapping between the Atts in an Attr Group associated to a Catalog Category and the Descriptive Elements for that Catalog Category (similarly, in this example we always set DESCRIPTION_DEFAULT as 'Y' for simplicity)

```
l_current_mapping_rec.ATTR_NAME := 'Style'; 
l_current_mapping_rec.DESC_ELEM_NAME := 'Style'; 
l_current_mapping_rec.DESCRIPTION_DEFAULT := 'Y'; 
l_mapping_rec_table(l_mapping_rec_table.COUNT + 1) := l_current_mapping_rec; 
```

```
l_current_mapping_rec.ATTR_NAME := 'GB_Options'; 
l_current_mapping_rec.DESC_ELEM_NAME := 'Grille Bar Options'; 
l_current_mapping_rec.DESCRIPTION_DEFAULT := 'Y'; 
l_mapping_rec_table(l_mapping_rec_table.COUNT + 1) := l_current_mapping_rec; 
```

```
l_current_mapping_rec.ATTR_NAME := 'Wood_Type'; 
l_current_mapping_rec.DESC_ELEM_NAME := 'Wood Type'; 
l_current_mapping_rec.DESCRIPTION_DEFAULT := 'Y'; 
l_mapping_rec_table(l_mapping_rec_table.COUNT + 1) := l_current_mapping_rec; 
```

```
l_current_mapping_rec.ATTR_NAME := 'Frame_Area'; 
l_current_mapping_rec.DESC_ELEM_NAME := 'Frame Area'; 
```
l_current_mapping_rec.DESCRIPTION_DEFAULT := 'Y';
l_mapping_rec_table(l_mapping_rec_table.COUNT + 1) := l_current_mapping_rec;

l_current_mapping_rec.ATTR_NAME := 'GRADE';
l_current_mapping_rec.DESC_ELEM_NAME := 'Grade';
l_current_mapping_rec.DESCRIPTION_DEFAULT := 'Y';
l_mapping_rec_table(l_mapping_rec_table.COUNT + 1) := l_current_mapping_rec;

l_current_mapping_rec.ATTR_NAME := 'FW_Extterior';
l_current_mapping_rec.DESC_ELEM_NAME := 'Primed Wood Exterior';
l_current_mapping_rec.DESCRIPTION_DEFAULT := 'Y';
l_mapping_rec_table(l_mapping_rec_table.COUNT + 1) := l_current_mapping_rec;

l_current_mapping_rec.ATTR_NAME := 'DATE';
l_current_mapping_rec.DESC_ELEM_NAME := 'Date';
l_current_mapping_rec.DESCRIPTION_DEFAULT := 'Y';
l_mapping_rec_table(l_mapping_rec_table.COUNT + 1) := l_current_mapping_rec;

ELSIF (l_attr_group_name_param.getValue() = 'WD_Attributes') THEN

  l_current_mapping_rec.ATTR_NAME := 'STYLE';
l_current_mapping_rec.DESC_ELEM_NAME := 'WD Style';
l_current_mapping_rec.DESCRIPTION_DEFAULT := 'Y';
l_mapping_rec_table(l_mapping_rec_table.COUNT + 1) := l_current_mapping_rec;

  l_current_mapping_rec.ATTR_NAME := 'WD_Grade';
l_current_mapping_rec.DESC_ELEM_NAME := 'WD Grade';
l_current_mapping_rec.DESCRIPTION_DEFAULT := 'Y';
l_mapping_rec_table(l_mapping_rec_table.COUNT + 1) := l_current_mapping_rec;

  l_current_mapping_rec.ATTR_NAME := 'D ATETIME';
l_current_mapping_rec.DESC_ELEM_NAME := 'Date';
l_current_mapping_rec.DESCRIPTION_DEFAULT := 'Y';
l_mapping_rec_table(l_mapping_rec_table.COUNT + 1) := l_current_mapping_rec;

  l_current_mapping_rec.ATTR_NAME := 'URL_link';
l_current_mapping_rec.DESC_ELEM_NAME := 'URL';
l_current_mapping_rec.DESCRIPTION_DEFAULT := 'Y';
l_mapping_rec_table(l_mapping_rec_table.COUNT + 1) := l_current_mapping_rec;

END IF;

-----------------------------------------------------------------
For every Desc Elem, get its value (a NULL value, or a DML_TYPE of
DELETE, will result in a NULL Elem value), set its DESCRIPTION_DEFAULT flag,
and add it to the ITEM_DESC_ELEMENT_TABLE
-----------------------------------------------------------------

FOR i IN l_mapping_rec_table.FIRST .. l_mapping_rec_table.LAST LOOP
  l_current_mapping_rec := l_mapping_rec_table(i);

Synchronizing Item User-Defined Attributes with Item Descriptive Elements

```sql
l_current_item_desc_elem.ELEMENT_NAME :=
l_current_mapping_rec.DESC_ELEM_NAME;
l_current_item_desc_elem.DESCRIPTION_DEFAULT :=
l_current_mapping_rec.DESCRIPTION_DEFAULT;
-------------------------------------------------------------------
This example uses single-row Attr Groups, so we don't need to make use
of the EGO_USER_ATTR_ROW_TABLE object; but for a multi-row Attr Group,
we would use the ROW_IDENTIFIER field of each EGO_USER_ATTR_ROW_OBJ to
find all the EGO_USER_ATTR_DATA_OBJ objects for that particular row
------------------------------------------------------------------
IF (x_attributes_row_table IS NOT NULL AND
    x_attributes_row_table.COUNT > 0 AND
    x_attributes_data_table IS NOT NULL AND
    x_attributes_data_table.COUNT > 0) THEN

    l_attributes_data_index := x_attributes_data_table.FIRST;
    WHILE l_attributes_data_index <= x_attributes_data_table.LAST
    LOOP
        EXIT WHEN l_current_item_desc_elem.ELEMENT_VALUE IS NOT NULL;
        l_current_data_obj :=
            x_attributes_data_table(l_attributes_data_index);
        IF (l_current_data_obj.ATTR_NAME =
            l_current_mapping_rec.ATTR_NAME) THEN
            l_current_item_desc_elem.ELEMENT_VALUE :=
                SUBSTRB(l_current_data_obj.ATTR_DISP_VALUE, 1, 30);
        END IF;
        l_attributes_data_index :=
            x_attributes_data_table.NEXT(l_attributes_data_index);
    END LOOP;
END IF;
--------------------------------------------------------
-- Now we add the ITEM_DESC_ELEMENT record into our table
-------------------------------------------------------
l_item_desc_elem_table(l_item_desc_elem_table.COUNT + 1) :=
l_current_item_desc_elem;
END LOOP;
----------------------------------------------------
Now that we have all the necessary data and metadata, we
call the API to update the Desc Elems for this Attr Group
(note that the API returns the Item Description generated
from the updated Descriptive Elements; if required, you
can update the Description through Item Open Interface)
---------------------------------------------------
INV_ITEM_CATALOG_ELEM_PUB.Process_Item_Descr_Elements(
    p_api_version => 1.0,
    p_inventory_item_id => l_inventory_item_id_param.getValue() -> l_inventory_item_id_param.get_value(),
    p_item_desc_element_table => l_item_desc_elem_table,
    x_generated_descr => x_generated_descr,
    x_return_status => x_return_status,
    x_msg_count => x_msg_count,
    x_msg_data => x_msg_data
);
RETURN l_return_status;
EXCEPTION
WHEN OTHERS THEN
    WF_CORE.CONTEXT('EGO_EF_QA_PUB', 'Synch_Attrs_With_Desc_Elems',
        p_event.getEventName(), p_subscription_guid);
```

Synchronizing Item User-Defined Attributes with Item Descriptive Elements   D-9
WF_EVENT.setErrorInfo(p_event, 'ERROR');
    RETURN 'ERROR';
END Synch_Attrs_With_Desc_Elems;
Change Management Business Events

This appendix covers the following topics:

- Description of Change Management Business Events

Description of Change Management Business Events

Change Management business events can be triggered for the following actions:

- Approval status change
- Priority changes
- Status changes
- Workflow status changes
- Creation of a change object

**Note:** To subscribe to any of the following business events, see Subscribing to a Change Management Business Event in the Oracle Product Lifecycle Management User's Guide, or the Oracle Product Information Management Data Librarian User's Guide.

<table>
<thead>
<tr>
<th>Business Event Name</th>
<th>Triggering Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>oracle.apps.eng.cm.changeObject.changeApprovalStatus</td>
<td>Approval status changes</td>
</tr>
<tr>
<td>oracle.apps.eng.cm.changeObject.changePriority</td>
<td>Priority changes</td>
</tr>
</tbody>
</table>
oracle.apps.eng.cm.changeObject.changeStatus  Status changes
oracle.apps.eng.cm.changeObject.changeWorkflowStatus  Workflow status changes
oracle.apps.eng.cm.changeObject.create  Creation of a change object
oracle.apps.eng.cm.changeObject.postComment  Posted comment on a change object
oracle.apps.eng.cm.changeObject.reassign  Reassignment of a change object
oracle.apps.eng.cm.changeObject.requestComment  Common request of a change object
oracle.apps.eng.cm.changeObject.submit  Submission of a change object
oracle.apps.eng.cm.changeObject.update  Update of a change object
oracle.apps.eng.cm.changeOrder.changeScheduleDate  Schedule date change
oracle.apps.eng.cm.import.complete  Import is completed
oracle.apps.eng.cm.revisedItem.changeScheduleDate  Revised item schedule date change
oracle.apps.eng.cm.revisedItem.changeStatus  Revised item status changes

**Change Management Dependencies and Interactions**

Change Management business events, depend on the following systems and applications:

- Workflow
- Business events
• XML Publisher

• Oracle TCA

Change Management business events can interact with the following:

• Items

• Multiple modules in Oracle Discrete Manufacturing, including: Engineering, Bills of Materials, Work In Process, and Planning

• CADView-3D

• Projects