SYSTEM SETUP GUIDE

STEP Trailblazer

Table of Contents

Introduction

About This Guide

The System Setup Guide describes how you setup the fundamental elements of the STEP system. The general functionality of the STEP system is explained in the STEP User Guide.

The guide assumes the following:

- Users are familiar with the STEP User Guide
- Users have attended STEP standard training

Accessing Short-cut Menus on PC and Mac

You can access shortcut menus by right-clicking on both PC and Mac if you are using a two button mouse. If you are working on a Mac with a one-button mouse, use CTRL-click to access the shortcut menu. This guide uses the term right-click.

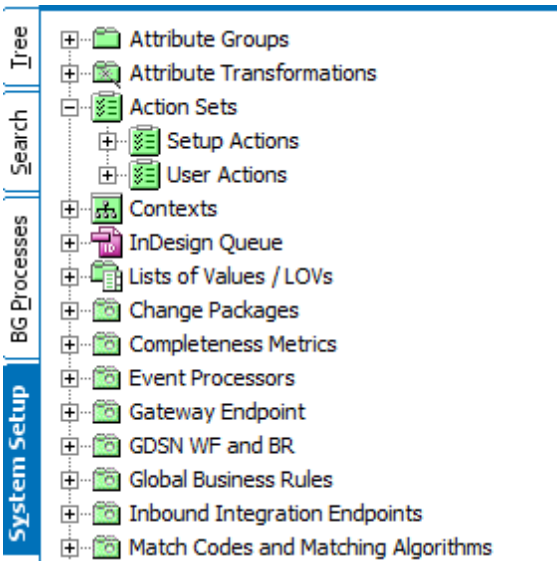
System Setup

The System Setup documentation describes the primary configuration elements of the STEP system. Topics in this section primarily deal with functions used by STEP system administrators.

Action Sets

Action Sets are used to define the actions or privileges a user or group of users are permitted to perform in STEP. Actions are grouped into Action Sets and made available for User Groups under the **Users & Groups** setup area. (For more information about Users & Groups, see the **About Users and Groups** section of **Users and Groups**.)

Action Sets are maintained in System Setup > **Action Sets**.



Whereas an **Action** represents a single operation (such as create product, move product, delete product, or view product), an **Action Set** is a *group* of actions that contains the set of operations that are necessary to perform one or more tasks. An example Action Set could be:

Action Set	Operations
Maintain Products	• Create Products • Move Products • Delete Products • View Products

There are two types of Action Sets:

1. **Setup Actions:** Actions related to system configuration or system setup. Setup Actions generally define actions that can be performed in the System Setup tab. Many of these actions are considered administrator functions, but some of them may be required for non-administrator users, depending upon what functionality is required for their user roles. Create a Setup Action Set for each role and add the minimum actions that they need for that role.
2. **User Actions:** Actions related to data maintenance. User Actions generally define actions for end users working in the STEP Tree hierarchy and/or Web UI. Nearly all User Actions will fall into a 'general function' category:
   - View
   - Create
   - Modify
   - Approve
   - Delete
   - Classify

Depending upon the user role you are trying to create, it may be necessary to create a combination of Action Set types to facilitate the functions of that role. That is, you may need to create both a **User Actions** action set and a **Setup Actions** action set so the user has access to all of the functionality required to perform their specified tasks.

Recommendations for Action Sets

- Design for change. Requirements will change over time.
- Define Action Sets based on roles and general functions.
- Each role should have its own Action Set. This makes it easy to add or remove an action for a role without affecting other roles.

Setup Actions

Setup actions grant users the privileges to view and maintain (create, edit, delete) the object or task in question.

Important: At a minimum, all users must be granted the '**View context**' privilege in order to log on to STEP.

Action Sets

Action	Privileges Granted
Maintain action sets (create, delete, add to, remove from)	• Create action sets • Delete action sets • Add actions to action sets • Remove actions from action sets

Administration

Action	Privileges Granted
View Administration	Ability to access the STEP System Administration pages from the Web Start page
View System Setup Logs	Ability to view the System Setup Log and Workspace Log (available from View > Logs in the workbench) and the Log tab (available in the editor on most System Setup objects). Without this privilege, the System Setup Log option in the View > Logs menu is grayed out, and the Log tab is not visible on System Setup objects.

Assets

Action	Privileges Granted
Modify dimension dependency of all assets	Set dimension dependency of asset object types: • Add dimension to asset • Remove dimension from asset
Push Asset	Ability to push assets from within the Tree

Attributes

Note that the actions concerning domains are also needed for LOV maintenance.

Action	Privileges Granted
Add attribute to attribute group	Ability to add to and create attributes in attribute groups **Note:** The 'Create product attribute' action must also be added before users can add attributes to attribute groups.
Create attribute group	Ability to create attribute groups **Note:** The action 'Modify name/description of attribute group (also translate)' must also be added before users can create attribute groups.
Create product attribute	Ability to create attributes • To create attributes, the actions 'Add attribute to attribute group' and 'Modify definition of product attribute (domain and default unit)' must also be added. • To create LOV attributes, the actions 'Create domain' and 'Modify definition of domain (validator, etc.)' must also be added.
Delete attribute group	Ability to delete attribute groups
Delete product attribute	Ability to delete attributes • To delete attributes, the action 'Remove attribute from attribute group' must also be added. • To delete LOV attributes, the action 'Delete domain' must also be added.
Maintain validation templates	Create a validation template (validation base type) for attributes and LOVs
Remove attribute from attribute group	Ability to remove (unlink) attributes that are linked to other attribute groups as well as add attributes to a group.
Merge product attributes	Ability to merge attributes
Modify definition of product attribute	• Set and edit the validation base type of attributes • Edit the type of attributes (specification and description)

Action	Privileges Granted
(domain and default unit)	• Modify filters on LOV attributes. **Note:** The 'Create product attribute' action must also be added in order to set validation of attributes.
Modify dimension dependency of product attribute	Edit the dimension dependencies of attributes
Modify metadata for product attribute (property value) (also translate)	• Edit values of description attributes linked to attributes • Translate description attribute values linked to attributes
Modify name/description of attribute group (also translate)	• Edit the name of attribute groups • Translate the name and values in attribute groups
Modify name/description of product attribute (also translate)	• Edit names of attributes • Translate name and values in attributes **Note:** The 'Modify name/description of domain (also translate)' action must also be added in order to create LOV attributes.
Modify valid (node) types for product attribute	Set which object types the attributes should be valid for.
View attribute	Ability to view attributes
View attribute group	Ability to view attribute groups The 'View attribute group' action must be added for users to maintain attributes.
View metadata for attribute (property value)	View values of description attributes linked to an attribute

Background Processes

Action	Privileges Granted
Delete background-processes	Delete queued, started, and ended background processes
View Background Processes of Other Users	View and download files from background processes started by other users. For users without this privilege: • the 'Show processes of everyone' button on the workbench BG Processes tab is disabled • any processes started by other users are not visible in the workbench or Web UI • all inbound / outbound integration endpoints and event queues are not visible. (Gateway integration endpoints are still visible as they do not have associated background processes.)

Bulk Updates

Action	Privileges Granted
Perform Bulk Update	Ability to run or schedule a Bulk Update from the File menu.

Business Rules

Action	Privileges Granted
Maintain business-rule	Create, maintain, and delete business rules

Change Packages

Action	Privileges Granted
Maintain change package	Grants all privileges for change packages including creation of a change package, deletion, Start Impact Report, Updates, and Seal Package
View change package	Grants the privilege to view change packages from the System Setup tab.

Contexts

Action	Privileges Granted
Maintain dimensions and contexts	Create and delete dimensions, dimension points, and contexts

Cryptographic Keys

Action	Privileges Granted
Maintain cryptographic keys	Create or delete cryptographic keys, and change their passwords.

Data Quality

Action	Privileges Granted
Maintain unique keys	• Create new keys • Edit existing keys • Activate, deactivate, check, and delete keys
Modify unique key value	Update locked attribute values used in unique keys

Deduplication

Action	Privileges Granted
Maintain Deduplication Configurations	Create, maintain, and delete 'Match Codes and Matching Algorithms' and maintain deduplication configuration options
Maintain Possible Duplicates	Maintain, merge, and delete possible duplicates

Events

Action	Privileges Granted
Maintain Event Processor	• Set up or delete an Event Processor • Enable an Event Processor • Disable an Event Processor • Purge events • Publish events
Maintain event queues	Add, modify or to delete event queues. Removal of this action hides the Event Queues root in System Setup.
View Event Processor	View Event Processors

GDSN Data Pools

Having setup actions for GDSN data pools allow users to view and maintain (create, edit, delete) GDSN data pools and GDSN subscriptions.

Action	Privileges Granted
Create GDSN Data Pool Publisher	Create a GDSN Data Pool Publisher for publishing products to the GDSN. This is the action required to use the Easy setup of GDSN Component Model.
Create GDSN Data Pool Receiver	Create a GDSN Data Pool Receiver for receiving products from the GDSN
Create GDSN Recipient	Create GDSN recipient(s)
Create GDSN Subscription	Create GDSN subscriptions
Delete GDSN Data Pool Publisher	Delete a GDSN Data Pool Publisher
Delete GDSN Data Pool Receiver	Delete a GDSN Data Pool Receiver
Delete GDSN Recipient	Delete GDSN recipient(s)

Action	Privileges Granted
Delete GDSN Subscription	Delete GDSN subscriptions
Maintain GDSN subscription	Maintain GDSN subscriptions
Modify GDSN Data Pool Publisher metadata	Modification of the GDSN Data Pool Publisher metadata
Modify GDSN Data Pool Publisher name	Modification of the GDSN Data Pool Publisher name
Modify GDSN Data Pool Receiver metadata	Modification of the GDSN Data Pool Receiver metadata
Modify GDSN Data Pool Receiver name	Modification of the GDSN Data Pool Receiver name
Modify GDSN Recipient meta-data	Modification of the GDSN Data Pool Recipient metadata
Modify GDSN Recipient name	Modification of the GDSN Data Pool Recipient name
Modify GDSN Subscription meta-data	Modification of GDSN Subscription metadata
Modify GDSN Subscription name	Modification of the GDSN Subscription name
View GDSN Data Pool Publisher	View the GDSN Data Pool Publisher data
View GDSN Data Pool Publisher metadata	View the GDSN Data Pool Publisher metadata
View GDSN Data Pool Receiver	View the GDSN Data Pool Receiver data
View GDSN Data Pool Receiver metadata	View the GDSN Data Pool Receiver metadata
View GDSN Recipient	View the GDSN Recipient data
View GDSN Recipient meta-data	View the GDSN Recipient metadata

Action	Privileges Granted
View GDSN Subscription	View the GDSN Subscription data
View GDSN Subscription meta-data	View the GDSN Subscription metadata

Integration Endpoints

Action	Privileges Granted
Maintain Derived Event Types	Configure and maintain derived events. If Maintain Derived Event Types is disabled, there is no option to add one.
Maintain Integration Endpoint	Create, delete, enable, disable, and invoke integration endpoints. Users must also have the privilege 'View Integration Endpoint' in order to maintain an integration endpoint.
View Integration Endpoint	Ability to view integration endpoints

Link Types

Action	Privileges Granted

Maintain Link Type	• Create and delete reference types and product to classification link types
	• Modify reference types and product to classification link types
	• Apply description attributes to link types
	• Set dimension dependencies of link types
	Note: In order to apply description attributes to link types, these actions are also needed:
	  • View attribute
	  • View attribute group
	  • Modify valid (node) types for product attribute

List of Values (LOVs)

Action	Privileges Granted
Create domain	Ability to create LOVs **Note:** The actions 'Modify name/description of domain (also translate)' and 'Modify definition of domain (validator, etc.)' must also be included in order to create LOVs.
Create value in hard domain	Create values in LOVs where users are not allowed to add values ('hard' LOVs)
Create value in medium domain	Create values in LOVs where users are allowed to add values ('medium' LOVs)
Delete domain	Ability to delete LOVs
Delete value in hard/medium domain	Delete values in both types of LOVs: • Where users are not allowed to add values ('hard' LOVs) • Where users are allowed to add values ('medium' LOVs)
Maintain domain groups	Create and delete LOV groups
Merge entire domain	Ability to merge LOVs
Merge value in hard/medium domain	Merge values in both types of LOVs: • Where users are not allowed to add values ('hard' LOVs) • Where users are allowed to add values ('medium' LOVs)
Modify definition of domain (validator, etc.)	Set and edit the validation base type of LOVs **Note:** The 'Create domain' action must also be added in order to set validation of LOVs.
Modify dimension dependency of domain	Edit the dimension dependencies of LOVs
Modify name/description of domain (also translate)	• Edit name of LOVs • Translate names and values in LOVs

Action	Privileges Granted
Modify value in hard/medium domain (also translate)	Edit values in in both types of LOVs (including the privilege to translate): • Where users are not allowed to add values ('hard' LOVs) • Where users are allowed to add values ('medium' LOVs)
View domain	Ability to view LOVs **Note:** The 'View domain' action must be added in order to maintain LOVs.
View metadata for domain (property value)	View values of description attributes linked to an LOV

Object Types

Action	Privileges Granted
Maintain Setup Entity	Create and Entity Root and object type. Example: Completeness Metric
Maintain setup group	Create a Setup Group Root from Maintain > Insert > Setup Group Root...
Maintain system setups	Ability to maintain options under the System Setting tab located on the 'Users & Groups' node in System Setup.
Maintain type hierarchy (node types)	• Create object types • Delete object types In order to link attributes to object types, these actions are also needed: • View attribute • View attribute group • Modify valid (node) types for product attribute
View Setup Entity	View an Entity Root and object type. Example: Completeness Metric

Print Publication

Action	Privileges Granted
Maintain promotional pricing	Deprecated
Modify frozen publication	Modify a frozen publication. This action is only used in conjunction with the 'Publishing - Freeze' component model.

Purging

Action	Privileges Granted
Empty recycle bin	Ability to delete contents of the Recycle Bin. In case the Recycle Bin contains objects that exist in more than one workspace, users must also have the 'Force Delete and Purge' action enabled in order to remove those objects.
Force Delete and Purge	Ability to force purge objects that exist in more than one workspace and objects with events triggered. The action enables the Purge button in the empty recycle bin background process and the 'Force Delete and Purge' action in the Object Menu.
Purge revisions	Ability to delete obsolete revisions: • Single revisions • Specified range of revisions, global purging

Reports

Action	Privileges Granted
Run Reports	Allows users to run reports (including starting a reports background process) from File > Reports in the workbench. Without this privilege, the Reports option in the workbench File menu is grayed out.

SDK / API Documentation

Action	Privileges Granted
View SDK Documentation	Ability to view the SDK / API documentation from server/sdk

Tables

Action	Privileges Granted
Modify table settings	Ability to maintain table settings on table types, row types, column types, rules / line styles, and colors

Tags

Action	Privileges Granted
Maintain Tags	• Maintain style tags, special characters, character tags, footnotes, and hyperlinks • Maintain tag groups
Use tag	Use tags in the rich text editor, table editor, and in table transformations. To restrict certain users to using certain tags, you must create a tag group that contains these tags and link this tag group into an attribute group. In the User Group editor, the attribute group is applied together with the action set that contains the 'Use tag' action. Users belonging to the user group will only be allowed to use the tags in the tag group linked into the attribute group.

Units

Action	Privileges Granted
Maintain units	• Create unit groups, units, and unit conversion rules • Edit unit groups, units, and unit conversion rules • Delete unit groups, units, and unit conversion rules

Action	Privileges Granted
View unit	Ability to view units. This is used when linking units to attributes. **Note:** The 'View unit' action must be included in order to use the 'Maintain units' action

Users & Groups and Privilege Rules

Action	Privileges Granted
Maintain privilege rules (create, delete, modify)	• Create privilege rules for user groups • Edit privilege rules for user groups • Delete privilege rules for user groups **Note:** In order to maintain privilege rules, the action 'Maintain action sets (create, delete, add to, remove from)' is also needed.
Maintain users and groups	• Create users and user groups • Duplicate and copy users and user groups • Delete users and user groups
Maintain user password	Ability to change and reset user passwords
Share user settings	Limits whether or not a user can share their user configured views in Web UI.

Value Search

Action	Privileges Granted
Enable Value Search/Values in typeahead	Ability to search for attribute values within attributes

Web UI

Action	Privileges Granted
Maintain Impersonation	Ability to impersonate another user for Web UI support purposes
Update Web UI configuration	Access to modify and configure any STEP Web UI
View context	In addition to granting access to the STEP workbench, 'View context' also grants the ability to access the Web UI.
Web UI Administration	Grants the ability to: • "Revert" / copy to front • Delete action context menu • Duplicate action context menu • Status tab • Edit Web UI through designer • Edit Web UI through Workbench • Change Web UI ID in the XML

Workflows

Action	Privileges Granted
Disable STEP Workflow Auto-initiation in Imports	Ability to disable auto-initiation when importing data
Initiate Item in STEP Workflow	Ability to start workflows via the Object context menu and via bulk update. It does not have any impact on auto-initiation. The action can be applied to a setup group and will then only take effect for workflows in that setup group.
Maintain STEP Workflow	Ability to create, delete, cut, copy, duplicate, export, and edit workflows. The action can be applied to a setup group and will then only take effect for workflows in that setup group. Users must also have the 'View setup group' privilege in order to view and maintain workflows

Action	Privileges Granted
Remove Item from STEP Workflow	Ability to remove objects from workflows via the Object context menu or the workflow context menu. The action can be applied to a setup group and will then only take effect for workflows in that setup group.
STEP Workflow Administrator	• View other users' tasks • Complete / submit tasks not assigned to the current user • Reassign tasks • Edit task deadlines The action can be applied to a setup group and will then only take effect for workflows in that setup group.
View and Maintain STEP Workflow Profile	View, edit, create, and delete workflow profiles. If a user does not have this action, the STEP Workflow navigator tab sub tab 'Profile' will not be displayed. The action can be applied to a setup group and will then only take effect for workflow profiles in that setup group.
View and use STEP Workflow	View and use workflows. The action can be applied to a setup group and will then only take effect for Workflows in that setup group. It is required to also have the 'View setup group' privilege in order to view workflows.
View setup group	Necessary for users to view workflow tasks
View STEP Workflow Advanced tab	This action grants the privilege to view the 'Advanced' STEP Workflow navigator tab sub tab. It is required to also have the 'View setup group' privilege in order to view workflows.

Workspaces

Action	Privileges Granted
Maintain workspaces	Create and delete workspaces

User Actions

Assets

Maintain actions for assets allow users to maintain (create, edit, delete) images and documents, image and document references, and description attributes that appear on these objects. Note that the relevant **View** actions must be included in order to maintain these objects.

Action	Privileges Granted
Approve Create	Approve the creation of assets
Approve Modify	Approve the modification of assets. Examples of modifications: moving an object in the hierarchy; changes to products, assets, classifications, tables names, asset content, attribute links, product override relations, deletions, and DTP template linking.
Approve References	Approve reference links from products, i.e. links from products to products, links from products to assets, and links from products to classifications.
Approve Values	Approve the modification of specification and description attribute values of products and assets This privilege is also subject to the attribute group setting of Privilege Rules on user groups. For example, if a user only has the privilege to approve attributes in a determined attribute group, this user can only approve certain attributes of a product or asset. Also, approving table values depends on the attribute group setting of the Privilege Rules on user groups.
Classify asset (link asset to classification)	Link images and documents to classifications
Create asset	• Create image or document name (placeholder) • Change object type of images and documents • Duplicate images and documents **Note:** The action 'Modify name/description of asset (also translate)' must be included in order to create images and documents.

Action	Privileges Granted
Create asset reference	Create product and classification to image / document references and image / document to classification references
Declassify asset (unlink asset from classification)	Delete links from assets to classifications
Delete asset	Delete assets of specified object types
Delete asset reference	Remove product and classification to image / document references and image / document to classification references
Download asset	Export assets to hard disk
Modify metadata for asset (property value) (also translate)	Edit and translate values of description attributes linked to assets
Modify metadata for asset reference (link attribute value) (also translate)	Edit and translate description attribute values on asset references
Modify name/description of asset (also translate)	• Edit names of assets • Translate name and metadata attribute values of images and documents
Upload to asset	• Import assets to classifications • Replace contents of assets In order to import and replace images and documents, these setup actions must be added to the privileges: • Create value in medium domain • Modify value in hard/medium domain
View asset	View assets **Note:** The 'View asset' action must be added in order to work with images and documents. To make any of the view actions listed below have an effect, the 'View

Action	Privileges Granted
	asset' privilege must be a part of the action set. For instance, if the action 'View asset reference' is added and not 'View asset,' the user will not be able to see the image or document and therefore not the references either.
View Asset reference	View the product to image and document references
View metadata for asset (property value)	View the values of description attributes linked to assets
View metadata for asset reference (link attribute value)	View description attribute values on asset references These attributes will appear on all product to image or document references on the specified valid asset object types.

Collections

Action	Privileges Granted
Create Collection	Create collections
Create Collection Group	Create collection groups
Delete Collection	Delete collections
Delete Collection Group	Delete collection groups
View Collection	View collections
View Collection Group	View collection groups

Commercial Lists

User actions for commercial lists allow users to view, delete, create, and maintain terms lists / prices. The user actions for commercial lists can be applied to publications or eCatalogs.

Action	Privileges Granted
Create Commercial List	Create commercial terms lists of any type
Delete Commercial List	Delete commercial terms lists of any type
Freeze/thaw commercial list	Enable and disable 'Immutable' status of commercial terms lists
Modify Commercial Data	Modify the individual terms in commercial terms lists of any type **Note:** This action does not grant access to import data. In order to import commercial terms values, the 'Create Commercial List' action is needed.
Modify Commercial List Name	Provide names for commercial terms lists
Modify Metadata Value Commercial List	Modify metadata attribute values on commercial terms lists
View Commercial Data	View commercial terms lists of any type
View Metadata Value for Commercial List	View metadata attribute values on commercial terms lists

Data Quality Functionality

Action	Privileges Granted
Generate/Update Category Profiles	Generate and update Category Profiles
Merge Products	Merge confirmed duplicates
Modify Profile Configuration	Modify the information shown in the Category Profile tab
View Category Profile	See the Category Profile tab on products

eCatalog

Action	Privileges Granted
Modify eCatalog Configuration	Modify eCatalog configurations
Modify eCatalog Metadata	Modify the metadata attributes on eCatalogs (Currency, Default Start Date, Default End Date, etc.) Must have 'View eCatalog Metadata' before 'Modify eCatalog Metadata' is enabled
View eCatalog Metadata	View the metadata attributes on eCatalogs

GDSN Data Pools

Having user actions for GDSN data pools allows users to view and maintain (remove and add items within) the packaging hierarchy, and to send CICs both manually and automatically.

Action	Privileges Granted
Create context reference	Create a reference type that relates a target market to a context
Delete context reference	Delete a reference type that relates a target market to a context
Maintain GDSN publications	Maintain GDSN publications
Maintain GDSN registrations	Maintain GDSN registrations
Maintain packaging hierarchy	Remove and add items in the packaging hierarchy
Modify metadata for context reference (link attribute value) (also translate)	Modify a reference type that relates a target market to a context
Send CIC	Send a CIC manually or set up the system to send a CIC automatically
View packaging hierarchy	View the packaging hierarchy

Products, Classifications, and Entities

Maintain actions for **products** allow users to maintain (create, edit, delete) products, product references, description attributes, and specification attributes in the Tree and the Editor.

Maintain actions for **classifications** allow users to maintain (create, edit, delete) classifications, classification references, description attributes, and specification attributes in the Tree and the Editor.

Note that the relevant **View** actions must be included in order to maintain these objects.

Action	Privileges Granted
Approve Classification Product Links	Approve product classification links
Approve Create	Approve the creation of classifications, products, and assets
Approve Modify	Approve the modification of classifications, products, and assets Examples of modifications: moving an object in the hierarchy; changes to products, images and documents, classifications, tables names, asset content, attribute links, product override relations, deletions, and DTP template linking.
Approve References	Approve reference links from products, i.e. links from products to products, links from products to assets, and links from products to classifications.
Approve Values	Approve the modification of specification and description attribute values of products and assets: This privilege is also subject to the attribute group setting of Privilege Rules on user groups. For example, if a user only has the privilege to approve attributes in a determined attribute group, this user can only approve certain attributes of a product or asset. Also, approving table values depends on the attribute group setting of the Privileges Rules on user groups.
Classify product (link product to classification)	Link products to classifications
Create classification	Create and duplicate classifications of specified object types Note: The action 'Modify name/description of classification (also translate)' must be included in order to create classifications.
Create classification	Create classification references

Action	Privileges Granted
reference	
Create entity	Create, change, and duplicate entities **Note:** The action 'Modify name/description of entity (also translate)' must be included to create entities.
Create entity reference	Create entity references
Create product	• Create products of specified product object types • Change object types of specified products • Duplicate products The action 'Modify name/description of product (also translate)' must be included in order to create products.
Create product reference	Create product to product references
Declassify product (unlink product from classification)	Unlink products from classifications
Delete classification	Delete classifications of specified object types
Delete classification reference	Delete classification references
Delete entity	Ability to delete entities
Delete entity reference	Delete entity references
Delete product	Delete products of specified object types
Delete product reference	Delete product to product references
Edit tables	Create and maintain tables in classification and product editors.
Legalise/unlegalise table types	Ability to legalize tables on classifications and products. 'Legalise/Unlegalise Table Types' does not grant a privilege to also create and

Action	Privileges Granted
	maintain tables. This privilege is covered in the action 'Edit tables'.
Link (product) attribute into classification	Link specification attributes to classifications and products
Modify metadata for attribute to classification link (also translate)	Edit and translate description attribute values on specification attributes linked to classifications or products
Modify metadata for product (property value) (also translate)	Edit and translate the values of description attributes linked to a product
Modify metadata for classification (property value) (also translate)	Modify the metadata for classifications
Modify metadata for classification reference (link value) (also translate)	Modify the metadata for classification references
Modify metadata for entity (property value) (also translate)	Modify the metadata for entities
Modify metadata for entity reference (link attribute value) (also translate)	Modify the metadata for entity references
Modify metadata for product classification (link value) (also translate)	Edit and translate description attribute values on attributes linked to products. These attributes will appear on all specification attributes that are linked to specified valid product object types.
Modify metadata for product reference (link attribute value) (also translate)	Edit description attribute values on product to product references (including privilege to translate). These attributes will appear on all product to product references on the specified valid product object types.

Action	Privileges Granted
Modify name/description of classification (also translate)	• Edit names of classifications • Translate names and values in classifications
Modify name/description of product (also translate)	• Edit names of products • Translate names and values in products
Modify product attribute value (also translate)	Edit and translate values of specification attributes linked to products
Modify name/description of entity (also translate)	• Edit the name of an entity • Translate name and values in entities
Modify sub-product list of product-override	Modify, add, and delete sub-products of a product override.
Move classification from	Move classification nodes by copying / pasting or dragging / dropping classifications from another classification node within the same object type hierarchy. The 'Move classification to' Action must also be applied.
Move classification to	Move classification nodes by copying and pasting classifications to another product node. The 'Move classification from' Action must also be applied.
Move entity from	Copy and paste entities from another entity node within the same object type hierarchy. The 'Move entity to' action must also be applied.
Move entity to	Copy and paste entities to another entity node. The 'Move entity from' action must also be applied.
Move product from	Copy / paste and drag / drop products from another product node within the same object type hierarchy.

Action	Privileges Granted
	The 'Move product to' action must also be applied.
Move product to	Copy / paste and drag / drop products to another product node. The 'Move product from' action must also be applied.
Unlink (product) attribute from classification	Remove the link to specification attributes from classifications and products
View attribute to classification link	View attributes linked to classifications and products
View classification	Ability to view the classification nodes in the classification hierarchy. **Note:** The 'View classification' action must be added to work with classifications. To make any view action have an effect, the 'View classification' privilege must be a part of the action set. For example: If the action 'View classification reference' is added and not the 'View classification' action, the user will not be able to see the classification and therefore not the references either.
View classification reference	View classification references
View entity	Ability to view entities The 'View entity' action must be added to be able to work with entities. To make any of the view actions listed below have an effect, the 'View entity' privilege must be a part of the action set.
View entity reference	View entity references
View metadata for attribute to classification link (link attribute value)	View description attribute values that have been applied to the reference between the attribute and the classification. These attributes will appear on all specification attributes that are linked to specified valid classification object types.
View metadata for classification (property value)	View values of description attributes linked to a classification

Action	Privileges Granted
View metadata for classification reference (link attribute value)	View description attribute values that have been applied to the reference link from the classification. These attributes will appear on all classification to classification references on the specified valid classification object types.
View metadata for entity (property value)	View values of description attributes linked to an entity
View metadata for product classification (link attribute value)	View link attribute values of attributes that have been applied to the link type product attribute validation. These attributes will appear on all specification attributes that are linked to specified valid classification object type.
View metadata for product reference (link attribute value)	View link attribute values of attributes that have been applied to the link type marker edge product. These attributes will appear on all specification attribute to product references on the specified valid product object type.
View metadata for product (property value)	View values of description attributes linked to a product
View product	View the product nodes The 'View product' action must be added in order to work with products. To make any of the view actions listed below have an effect, the 'View product' privilege must be a part of the action set. For example, if the action 'View product attribute value' is added and not the 'View product' action, the user will not be able to see the product and therefore not the attribute values either.
View product attribute value	View the values of specification attributes linked to products
View product reference	View product to product references

Publications

Having user actions for publications allows users to view, delete, create, and maintain publications as well as additional object (node) types used for publications, including publication groups, sections, versions, and planned pages.

Action	Privileges Granted
Create DTP Template Document	Create product templates, page templates, and publication templates (master documents)
Create Generic Page Template	Create page templates
Create Planned Page	Add planned pages into sections
Create Publication	Create publications
Create Publication Document	Save mounted (actual) InDesign pages back to STEP
Create Publication Group	Create publication groups
Create Publication Section	Create publication sections
Create Version	Create publication versions
Delete DTP Template Document	Delete product templates, page templates, and publication templates (master documents)
Delete Generic Page Template	Delete page templates
Delete Planned Page	Delete planned pages
Delete Publication	Delete publications
Delete Publication Document	Delete actual pages
Delete Publication Group	Delete publication groups
Delete Publication Section	Delete publication sections
Delete Version	Delete publication versions
Download DTP Template Document	Download product templates, page templates, and publication templates (master documents)

Action	Privileges Granted
Download Publication Document	Download actual pages
Import Planned Page	Import planned pages
Maintain Flatplan Content	Change the object assigned to an existing frame
Maintain Flatplan Frame	Add, move, delete, and resize frames
Maintain public basket view	Maintain public Flatlanner basket views
Maintain public financial view	Maintain public Financial Summary views (Flatplanner)
Maintain public stickerbook view	Maintain public Sticker Book views (Flatplanner)
Manage autopage objects	Manage AutoPage objects
Manage layout in page inspector	Manage layouts in page inspector (AutoPage)
Manage pagination rules	Manage pagination rules
Modify Basket for Planned Page	Modify Flatplanner baskets at the planned page level
Modify Basket for Publication	Modify baskets at the publication level
Modify Basket for Publication Section	Modify baskets at the section level
Modify DTP Template Document	Modify existing product templates, page templates, and publication templates (master documents)
Modify Generic Page Template	Modify page templates
Modify Planned Page	Add products or images to planned pages. The following menu actions are also controlled by this privilege: • Set page templates on planned pages • Reuse saved layouts

Action	Privileges Granted
	• Stop workflows • Split spreads • Add notes • Remove notes
Modify Publication	Link products and images into publications
Modify Publication Document	Modify actual pages and insert notes on actual pages
Modify Publication Group	Modify the name and object type of publication groups
Modify Publication Section	Add objects to the Flatplanner basket on section nodes
Modify Version	• Set version as master • Set context and workspace for a version • Set commercial list for a version
Perform autopage export actions	Perform AutoPage export actions
Perform autopage server write actions	Perform AutoPage server write actions.
Reorder Planned Pages	Reorder planned pages
View DTP Template Document	View publication templates (master documents), product templates, and page templates
View Generic Page Template	View page templates
View Planned Page	View planned pages
View Publication	View publications
View Publication Document	View actual pages **Note:** An actual page is the same thing as a mounted page—a page saved back to STEP from InDesign.

Action	Privileges Granted
View Publication Group	View publication groups
View Publication Section	View publication sections.
View queued events	View queued events, (Asset push event queues)
View Version	View publication versions

Translations

Action	Privileges Granted
Change Translation	Use the Change Translation Setup and Change Translation Status functions

Web UI

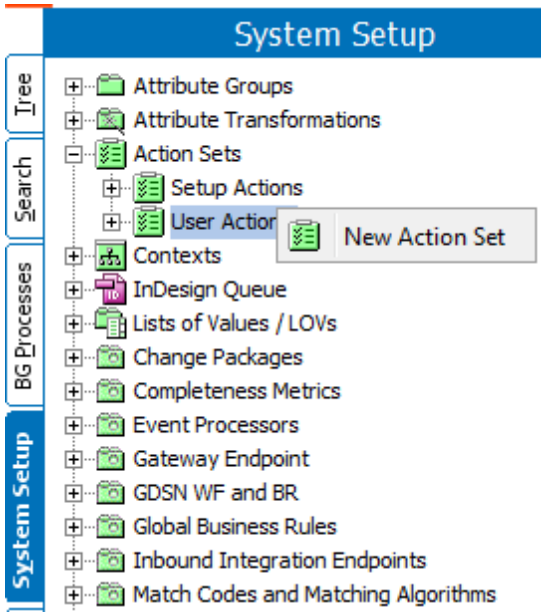
Action	Privileges Granted
Impersonate User	Impersonate a user in a targeted user group in order to troubleshoot issues
Share search with group	Share saved searches with specific user groups
(Supplier Web UI functions)	Use the Supplier Web UI functions in Supplier Web UI

Maintaining Action Sets

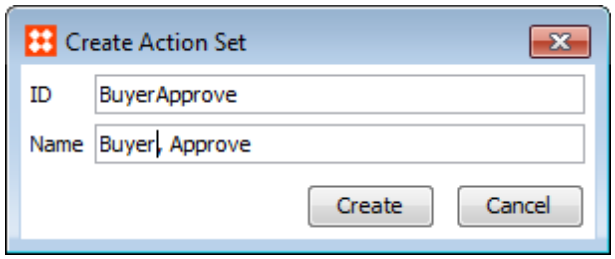
Creating an Action Set

Action Sets are created below either the Action Sets > Setup Actions or the Action Sets > User Actions node.

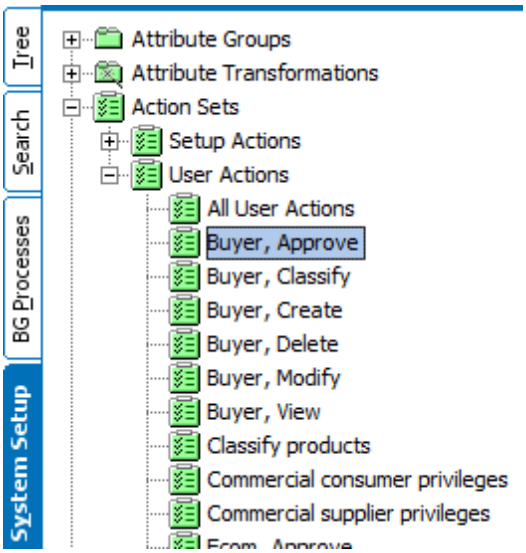
1. In [System Setup], unfold the **Action Set** group and select the relevant main Action Sets folder.
2. Right-click, and then click **New Action Set** (or in the **Object** menu, point to **Create** and then click **Action Set**).



A **Create Action Set** dialog box appears.



3. Type the ID and Name of the Action Set. In this example we will create a 'Buyer Approve' Action Set.
4. Click **Create**.

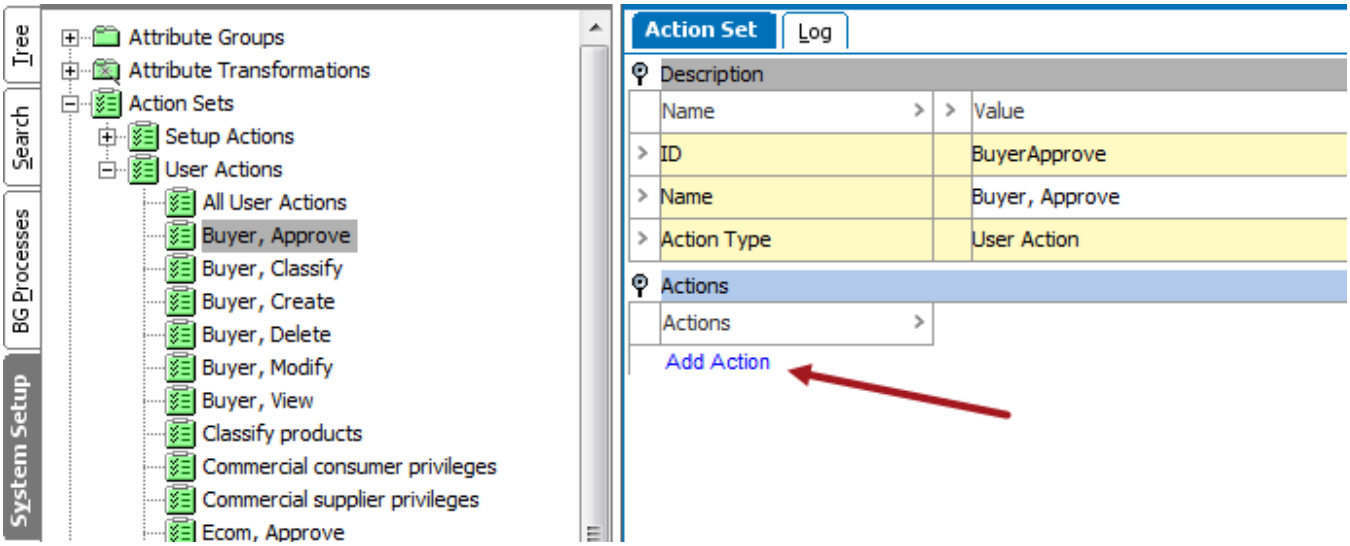


A new Action Set has now been created, and the next step will be to add the specific actions.

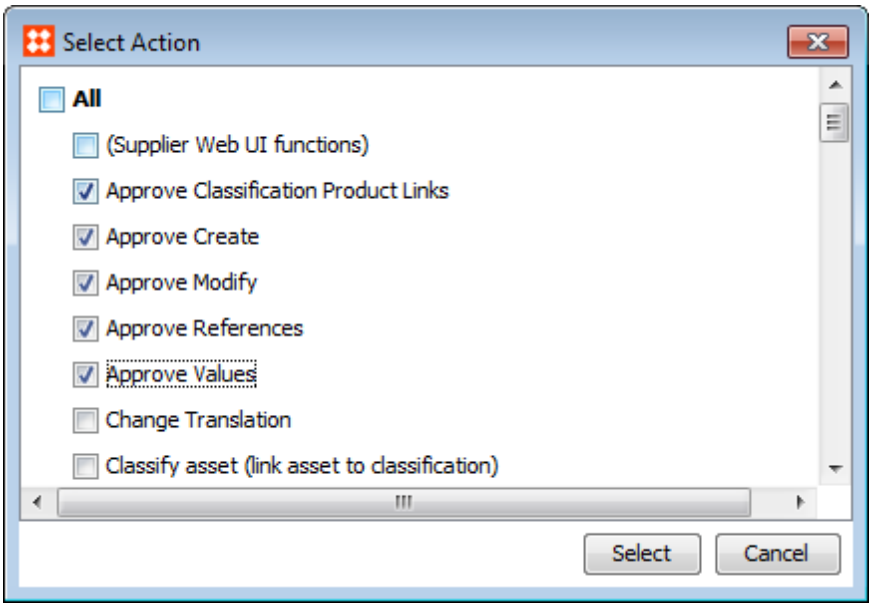
Adding Actions to an Action Set

Actions are added to the Set via the editor Add Action link.

1. In [System Setup], unfold **Action Set**, and then unfold the relevant main Action Sets folder.
2. Select the relevant Action Set and the **Action Set** Editor appears.

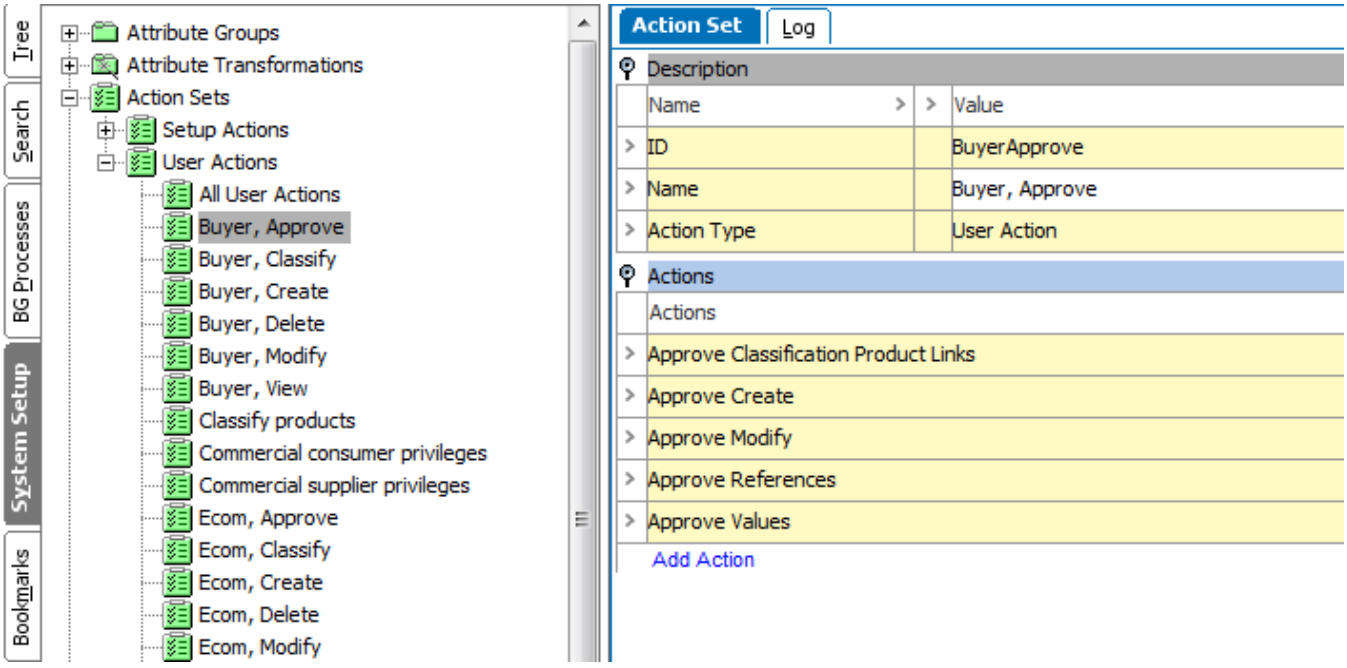


3. Under **Actions**, click **Add Action** and the **Select Action** dialog box appears.



4. Select all, or one or more actions by clicking the relevant check boxes.
5. Click **Select**.

   The selected action(s) appears under **Actions**.

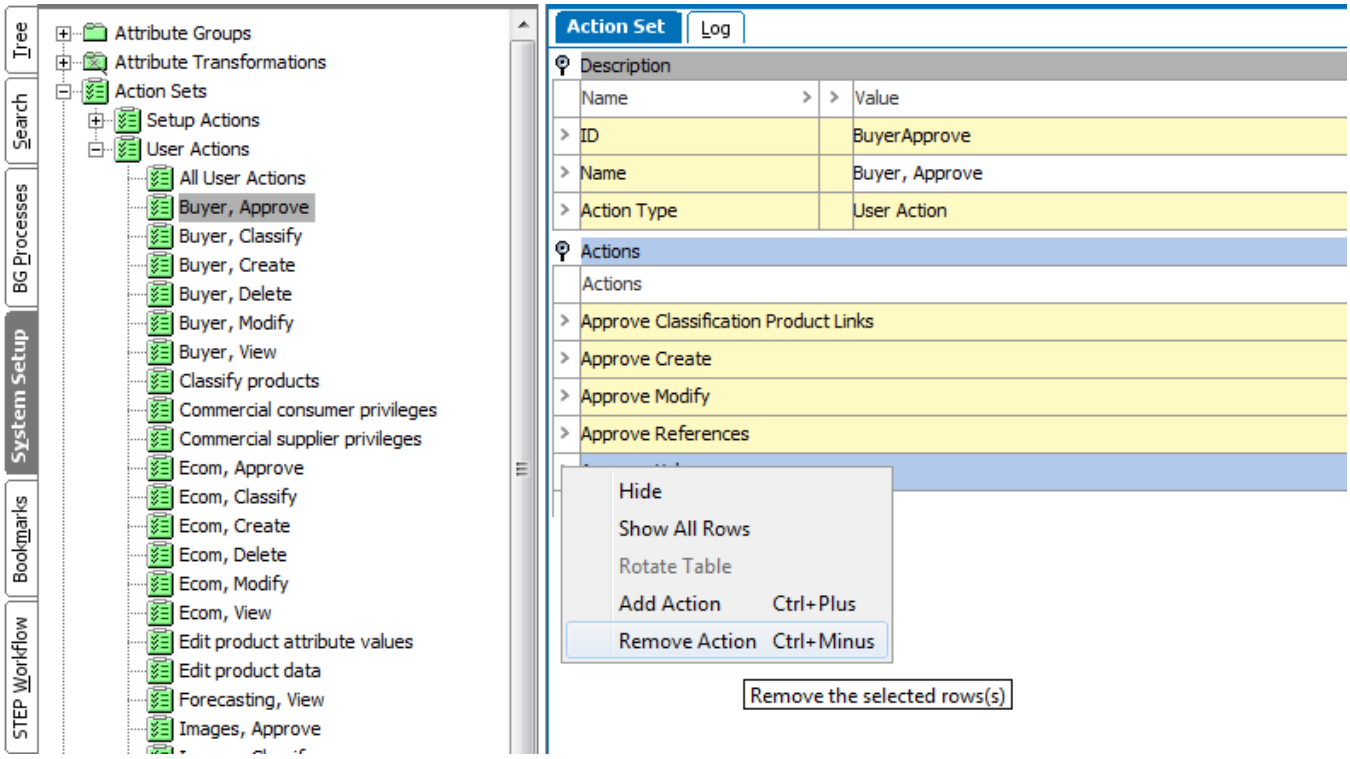


The selected Action Set will now include the added action(s), and when an action set is assigned to a user group, the members will be able to perform all actions in the actions list.

Removing an Action from Action Set

Action Sets are created in [System Setup]: **Action Sets**.

1. In [System Setup], unfold **Action Set**, and then unfold the relevant main Action Sets folder.
2. Select the relevant Action Set and the **Action Set** Editor appears.
3. Under **Actions**, select the action to be removed from the Action Set.

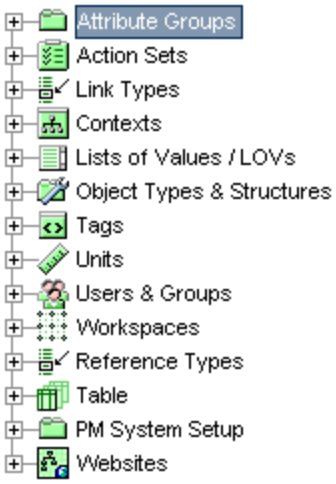


4. Press **Ctrl + < ->**, or right-click the row header arrow, and click **Remove Action**.

An action has now been deleted from an Action Set. When a user group is assigned to this Action Set, the members will no longer be able to perform this specific action.

About Attributes

Attributes are maintained in System Setup under Attribute Groups.



An attribute is a characteristic or a detailed piece of information related to a particular object.

Objects can be:

- Products
- Image & documents
- Classifications
- Link types

Two types of attributes are available:

- Description attributes
- Specification attributes

Attribute Groups

Using attribute groups, you can group attributes, reference types and link types according to their characteristics. An attribute group can also consist of a number of sub attribute groups. This scheme provides a better overview on a larger number of attributes, reference types and link types.

Note: It is recommended to limit the number of main attribute groups, and create sub attribute groups instead.

Examples of main attribute groups could be:

- Price group
- Technical group

The following are examples of sub attribute groups:

Attribute Main Group	Included Sub Groups	Included Attributes
Technical	Dimensions	Length, Depth, Height, Width
	FrameColor	FrameColor, SeatColor
	Temperature	Minimum, Maximum, Outdoor, Indoor, Operating
PriceGroup	Stibo Main	Price Break 1, Price Break 2, Discontinued
	Stibo UK	Price Break 1, Price Break 2, Discontinued
	Stibo US	Price Break 1, Price Break 2, Discontinued

Creating sub groups makes it easier to organize and view related attributes. Typically, you will need to link all (or most) of the Attributes in a sub group, which is done by multi-selecting and linking the selected Attributes.

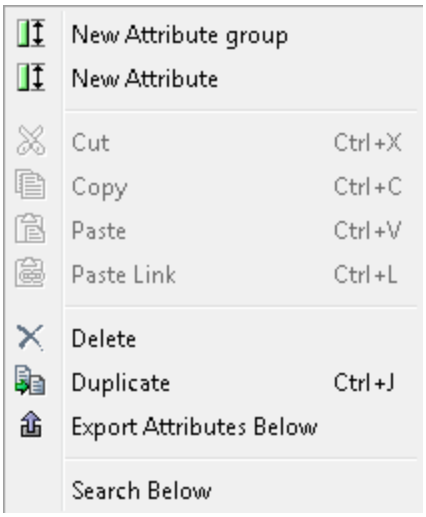
Note: Attribute groups can be specified in privilege rules. For example, specified users can change the values of attributes within the 'PriceGroup' attribute group.

Creating Attribute Groups

1. In **System Setup**, expand **Attribute Groups**, and then select the relevant attribute main group folder.

   Note: It is recommended to limit the number of main groups. Create sub-attribute groups instead.

2. Right-click and choose **New Attribute Group**. The **Create Attribute Group** dialog appears.



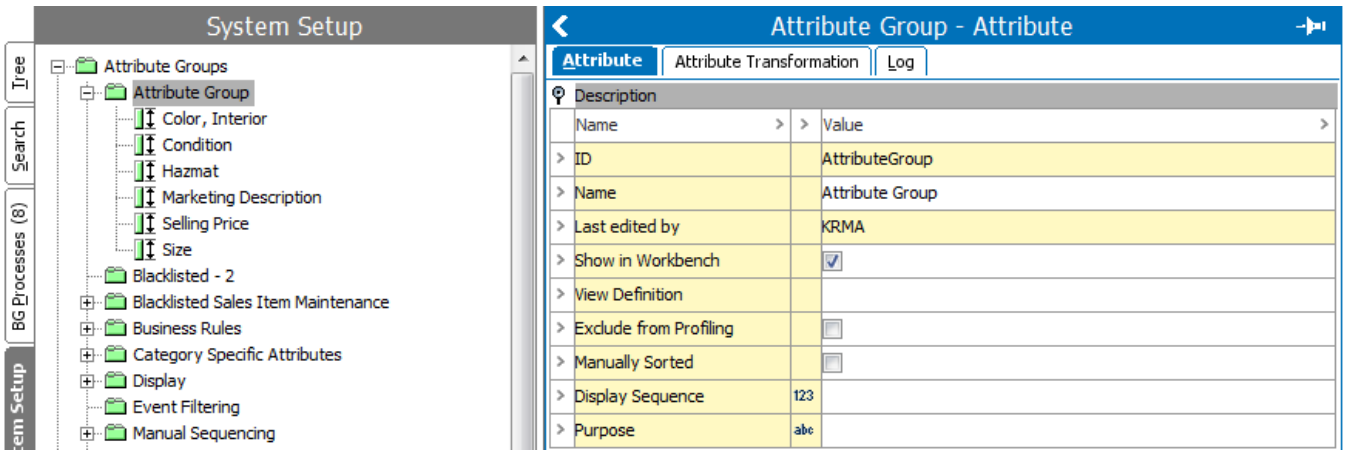
3. Enter an ID and a name for the attribute group.
4. Click **Create**. The new attribute group will appear in **System Setup**, below the attribute group where you created it.

You can now add attributes, reference types and product to classification link types to the attribute group.

Creating a Customized View

Attribute groups that contain reference types and product to classification link types can have their views defined. The customized view defines general aspects, meta data attributes and attributes on referenced objects. These referenced objects appear in the references editor on products, classifications and assets.

1. In **System Setup**, expand **Attribute Groups**, and then select the relevant attribute group. The **Attribute Group Editor** appears.

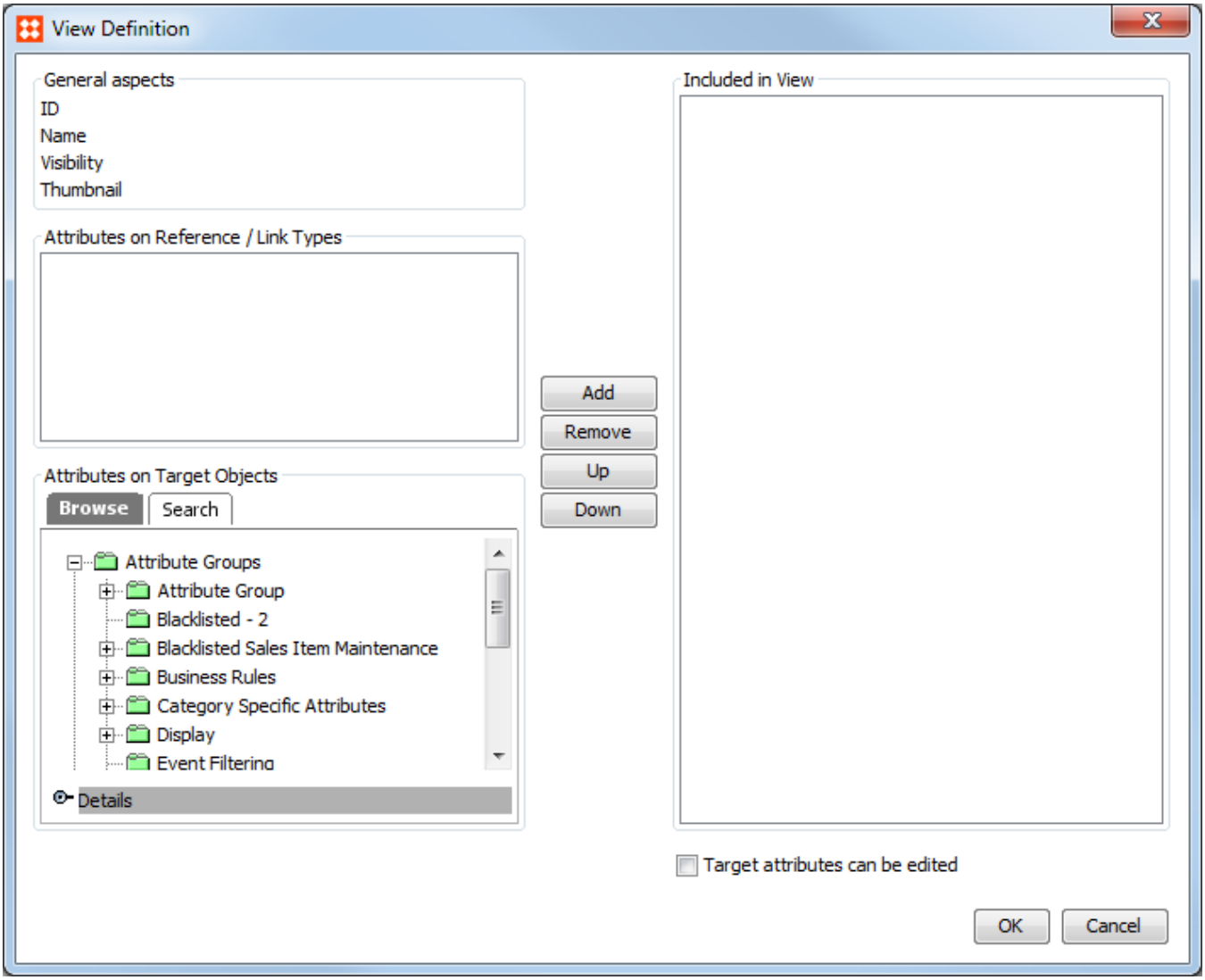


2. Check **Show in Workbench** if all reference types, product to classification link types and attributes linked into the Attribute Group should appear on products / classifications or Assets. **Show in Workbench**is

typically unchecked if a privilege needs to be applied to the attribute group to prevent the objects from appearing on products, classifications and assets.

3. In **View Definition**, click the ellipses **...** to specify the appearance of reference types and product to classification link types linked into the attribute group. The **View Definition Dialog** appears.

Note: A saved view definition does not influence attributes linked into the attribute group.



In the **View Definition** dialog, a number of choices are available to be included in the view.

General Aspects	Definition
ID	ID shows the ID of the referenced / linked object.
Name	Name shows the name of the referenced / linked object
Visibility	Visibility shows in which contexts a reference or product to classification link is visible. Reference types or product to classification link types that are dimension dependent are editable by users with the required privileges.
Thumbnail	For product to product References and product to image references, a thumbnail of the primary image of the target object is shown.

Attributes on References / Link Types	Definition
Metadata Attributes	This field lists the metadata attributes that have been applied the reference types and product to classification link types linked into the attribute group.

Attributes on Target Objects	Definition
Description Attributes or Specification Attributes	In Attributes on Target Objects you can search or browse for attributes. When you add an attribute to the view from this field, it displays the attribute value in the reference editor. **Note:** Each reference type establishes the hierarchical position among the referenced objects. The direction will identify one object as the master and the referenced object as the subordinate. A master object is the object that the reference is made from. A subordinate object is the object that the reference is made to. Attributes applied from this field are values from the subordinate object. **Important:** Checking 'Target Attributes can be edited' enables maintenance of the values of the selected attributes from the reference editor. Editing is allowed in the reference editor if the user privileges allow maintenance of the attribute values.

4. Click **OK** to save the view

In the reference editor on products, classifications and assets, the view definition is used where the reference types and product to classification link types are valid.

Manually Sorting in Attribute Groups

It is possible to manually define the order of attributes within an attribute group. To enable manual sorting, select the attribute group in **System Setup** below **Attribute Groups**. In the editor for the attribute group, select Manually Sorted. Note that Attribute groups can only be Manually Sorted if they only have attributes as children—attribute groups, reference types etc. are not allowed.

After the attribute group has been configured as Manually Sorted, drag and drop can be used for setting the order of attributes within the attribute group. This order will override any other attribute ordering used in various places where that particular attribute group is used, e.g. in the STEP Workbench, when viewing the values of a product below the attribute group flipper for this attribute group. If the attribute is used in another attribute group as well, the ordering inside that attribute group will be independent of the ordering within this group.

Attribute Transformations

This section describes the different attribute transformations.

About Attribute Transformations

Attribute transformations are maintained in System Setup > **Attribute Transformations**.

Example attribute transformations could be:

- Convert numbers to only have two decimals.
- Replace comma with a dot.
- Insert a suffix on a product name. A suffix could be the text 'NEW' and used on new products.
- Transform dates by pattern or locale during import and export.

The examples illustrate that the attribute transformations can be used to perform very different and advanced tasks within the STEP system.

Note: Attribute transformations are set up in STEP PIM and can be used to transform attribute values, product names, and terms lists.

Available Attribute Transformation Rules

A number of Attribute Transformation rules are available:

Transformation Rules	Description
Append text	This transformation will append a specified text string to the value. The text will be inserted as a suffix to the value.
Change case	This transformation can change the case of a value. The different options are: • Upper Case - Applying this transformation will capitalize all characters of a value. • Lower Case - Applying this transformation will change all characters of a value to small capitals. • Mixed Case - Applying this transformation will capitalize the first character of a value and change the rest of the value to small capitals.
Convert fractions to parsabletags	This transformation scans a text string for fractions. If a fraction is found, then the fraction will be converted into parsable tags.

Transformation Rules	Description
	Example: A text string that contains the fraction 1/2 cm will be transformed to <StartNumerator/><bfldin/>1 <FractionBar/> <StartDenominator/>2 <EndFraction/> cm Note: It is needed to setup an output formatting to InDesign for <StartNumerator/>, <bfldin/>, <FractionBar/>, <StartDenominator/> and <EndFraction/> in order to mount the fractions.
Extract characters after a given text in the value	This transformation extracts all characters after a specified text string.
Extract characters at position	This transformation extracts a specified number of Characters after a specified position in a text string.
Extract characters before a given text in the value	This transformation extracts all characters before a specified text string.
Extract characters between two pieces of text in the value	This transformation extracts all characters between two pieces of texts.
Extract prefix, suffix and value from an embedded number	This transformation extracts a prefix, suffix and value from an embedded number.

Transformation Rules	Description
Extract unit from an embedded value	This transformation extracts only the unit from an embedded value. The prefix, suffix and value are skipped and will not be extracted.
Extract unit part of value	This transformation extracts only a unit used in a value of a Numeric Text, Integer, Fraction or Number Attribute.
Extracts the last N characters of the value	This transformation extracts a specified number of Characters of a Value. The number of Characters that will be extracted are the last Characters of the Value.
Format Number	This transformation makes it possible to format a number to only have a specified number of decimals.
Insert text at a given position	This transformation inserts a specified text in a specified position of a value.
Insert text before	This transformation inserts a specified text string as a prefix of a value.
Math operation (+,-,*,/)	This transformation enables mathematical rules to be used on integer and number attributes.
Remove Unit from Value	This transformation will extract the value of an attribute. If the values also includes a Unit, then the Unit will be ignored.
Replace Substrings of the value	This transformation will replace specified Characters of a Value. Example: All instances of 'Personal Computer' will be replaced with 'PC'.
Replace Substrings of the value using regular expressions	This Transformation makes it possible to use regular expressions to replace substrings of a value.
Replace the whole value	Replace the whole value with another value.

Transformation Rules	Description
Split and extract	Split the value where a given string is found and returns one of the created fields.
Transform date by locale	On Import: Transforms locale-formatted Date, ISO Date, or ISO Date and Time attributes. The locale format must match the context being used in Workbench at the time of import. If not, the 'Transform date by pattern' transformation should be used. On Export: Transforms Date, ISO Date, or ISO Date and Time attributes to localized formats. Locale is determined by the context being used in Workbench at the time of export. Localized dates follow the DateFormat.SHORT JavaScript formatting style. Times will be shown in the DateFormat.MEDIUM formatting style. Step-by-step instructions can be found in the Attributes section of the System Setup / STEP Super User documentation.
Transform date by pattern	On Import: Transforms patterned dates (e.g., yyyy.MM.dd G 'at' HH:mm:ss z, EE, MMM d, ''yy) to Date, ISO Date, or ISO Date and Time. Validation Base Type for the mapped attribute must match the 'Transform to' dropdown choice. On Export: Transforms Date, ISO Date, or ISO Date and Time attributes to a patterned format (e.g., yyyy.MM.dd G 'at' HH:mm:ss z, EE, MMM d, ''yy). Users can use the pre-filled pattern provided or manually enter (or cut and paste) a different pattern format. Validation Base Type for the attribute being exported must match the 'Transform from' dropdown choice. If exporting data to a pattern that includes a time zone, the time zone is determined by the STEP server location. Step-by-step instructions can be found in the Attributes section of the System Setup / STEP Super User documentation.
[Transform using attribute transformation]	Extracts data based on the attribute transformation that is already defined in System Setup under Attribute Transformations.

Multivalue Separator	Description
Separator	In this field it is possible to key in a character or character tag to be used to specify how values

Multivalue Separator	Description
	in a multivalued attribute should be mounted / extracted. If a character is keyed in, then the character will be inserted between each value in the multivalued Attribute. Example: if, is used as a separator then a comma will be inserted between each value. Red, Green, Blue. If a character tag is keyed in, then the output formatting of the specific character tag is used to separate the values. Example: If </return> is used as a separator, then each value will be extracted with a return. Red Green Blue

Note: You can apply multiple transformations, but you do have to be careful about the order in which the transformations are applied.

Note: Transformations are not intended to be a substitute for a programming language. The intent is to give a non-scientific user the opportunity to modify data, using series of standard modification operations.

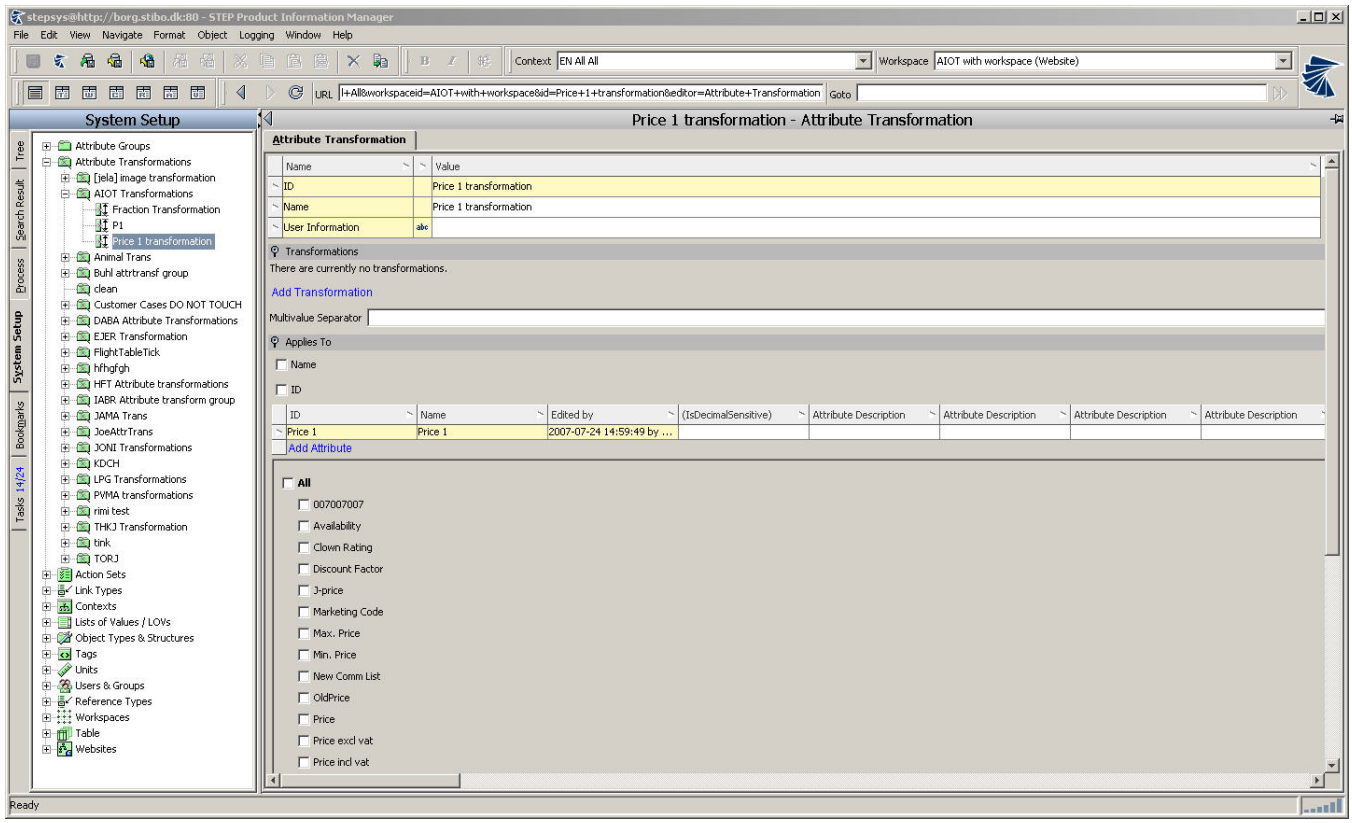
Maintaining Attribute Transformations

Creating a Transformation

Attribute Transformations are maintained in [System Setup]: **Attribute Transformations**.

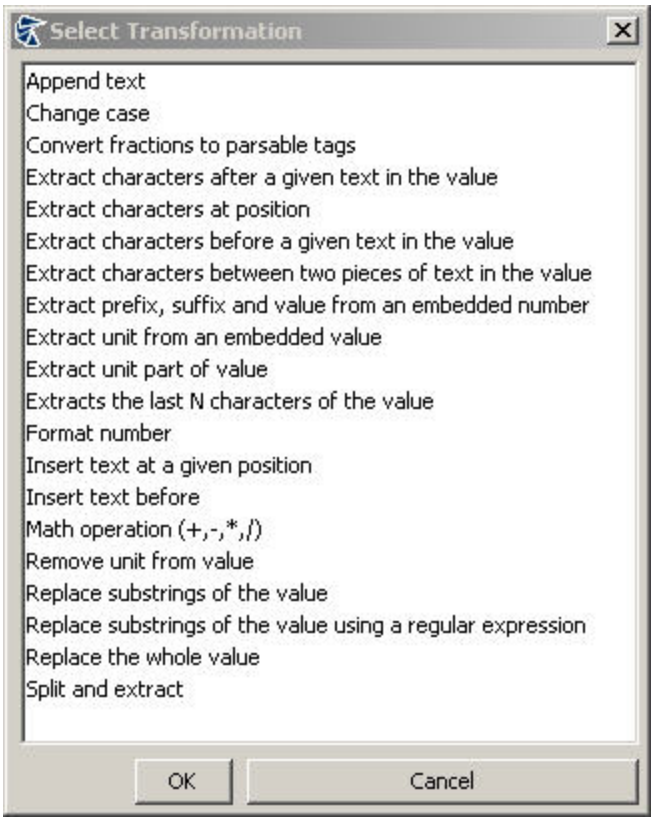
1. In [System Setup], unfold **Attribute Transformation**, and then select an Attribute Transformation.

   An **Attribute Transformation** Editor appears.



2. Under **Transformations**, click **Add Transformation**.

   A **Select Transformation** dialog box appears with a list of rules to be used.



3. Select the relevant Transformations, and then click **OK**.

   For a description of the Translation Rules, see Available Attribute Transformation Rules.

   Note: It is possible to create multiple sequential transformations as a rule, but you do have to be careful about the order in which the transformations are applied.

4. Under **Transformations**, in the **Multivalue Separator** field, type a character or character tag to be used to specify how values in a Multivalued Attribute should be mounted / extracted.

   For more information, see Available Attribute Transformation Rules.

5. Under **Applies To**, check the **Name** check box if it should be legal to use the Transformation on Object Names. An object name could be a product name.

   The transformation can be used in:

   - Tables on columns or rows where an Object Name is extracted. Typically it would be used on a column or a row that lists product names. For more information about applying Attribute Transformations, see Online Help - Tables and STEP Table Component User Guide.
   - InDesign on product templates. The product template can be setup to include the transformation, when an Object Name is mounted. See STEP'n'design User Guide for further information.

6. Under **Applies To**, check the **ID** check box if it should be legal to use the Transformation on object IDs. An object ID could be a product ID.

The transformation can be used in

- Tables on columns or rows where an Object ID is extracted. Typically it would be used on a column or a row that lists product IDs. For more information about applying attribute transformations, see Online Help - Tables and STEP Table Component User Guide.
- InDesign on product templates. The product template can be setup to include the transformation, when an Object ID is mounted. See STEP'n'design User Guide for further information.

7. Under **Applies To**, click **Add Attribute**, and select an attribute group or an attribute to legalize the transformation on an attribute. If the transformation is made legal for an attribute group, then the transformation will be made legal for all attributes in the group.

   The transformation can be used in

   - Tables on columns or rows where attributes are extracted. The transformation can be applied columns or rows that list attribute values. For more information about applying attribute transformations, see STEP Table Component User Guide.
   - InDesign on product templates. The product template can be setup to include the transformation, when an attribute is mounted. See STEP'n'design User Guide for further information.

8. Under **Applies To**, select a Terms List to legalize the transformation on a terms list.

   The transformation can be used in:

   - Tables on columns or rows where contents in the specific terms list is extracted. For more information about applying attribute transformations, see STEP Table Component User Guide.
   - InDesign on product templates. The product template can be set up to include the transformation, when contents in the specific terms list is mounted. See the STEPn´Design User Guide for further information.

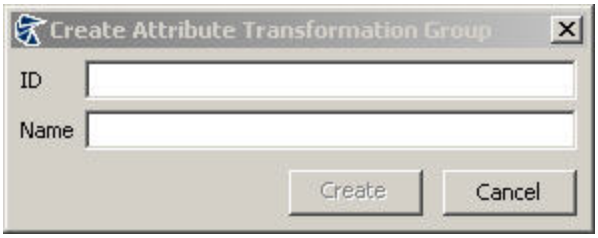
Important: If nothing is selected in the **Applies To** field, then the transformation will be considered as a Global Transformation. It means that it will be legalized on all types of Objects—Name, ID, Attributes and Terms Lists.

Creating an Attribute Transformation Group

Attribute transformations are maintained in [System Setup]: **Attribute Transformations**.

1. In [System Setup], unfold **Attribute Transformation**, right-click, and then click **Create Attribute Transformation Group**.

   A **Create Attribute Transformation Group** dialog box appears.



2. Type an ID and Name of the Attribute Transformation Group, and then click **Create**.

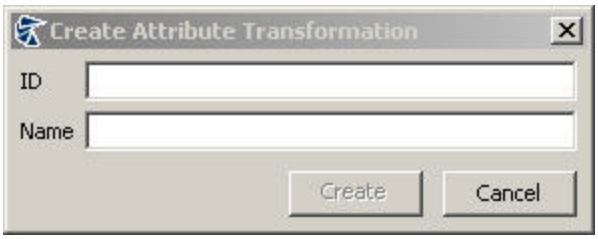
   The Attribute Transformation Group is created, and you can now define an Attribute Transformation for it.

Applying an Attribute Transformation

Attribute Transformations are maintained in System Setup > **Attribute Transformations**.

1. In [System Setup], unfold **Attribute Transformation**, click an **Attribute Transformation Group**, right-click, and then click **Create Attribute Transformation**.

   A **Create Attribute Transformation** dialog box appears.

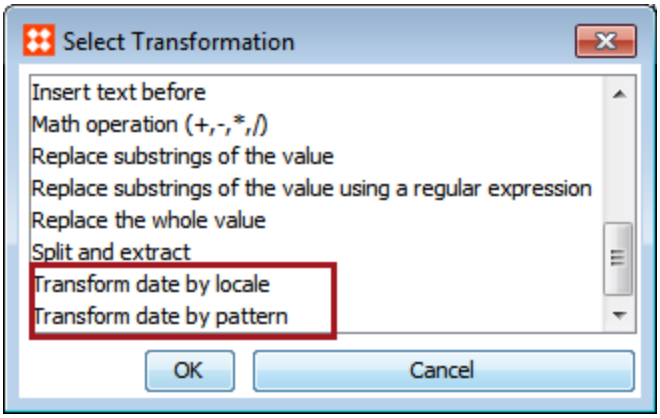


2. Type an ID and Name of the Attribute Transformation, and then click **Create**.

   The Attribute Transformation has been created. The next step will typically be to set up a Transformation. For more information, see Maintaining Attribute Transformations.

Date Attribute Transformations During Import and Export

Attributes with a Validation Base Type of Date, ISO Date, and ISO Date and Time can be transformed by locale or by pattern during the STEP Workbench import and export processes.



While it is recommended that users do STEP search, compare, and order operations with ISO date attributes, users may want to transform date attributes by locale or another recognizable format to meet a business requirement. Using the locale and pattern transformations does not impact the attribute Validation Base Type.

Transform date by locale

Dates converted to locale will follow the DateFormat.SHORT Java formatting style. Times converted to locale will be the DateFormat.MEDIUM formatting style. This locale information is based on the context being used in Workbench at the time of export or import.

Date and Time (DateFormat) Samples	English (United States)	Danish (Denmark)	French (France)
Date (SHORT)	8/14/15	14-08-15	14/08/15
Time (MEDIUM)	11:25:40 AM	11:25:40	11:25:40

Transform date by pattern

A few examples are shown in the following table. If time zone is included in a pattern used to export data, the time zone abbreviation is determined by the location of the STEP server.

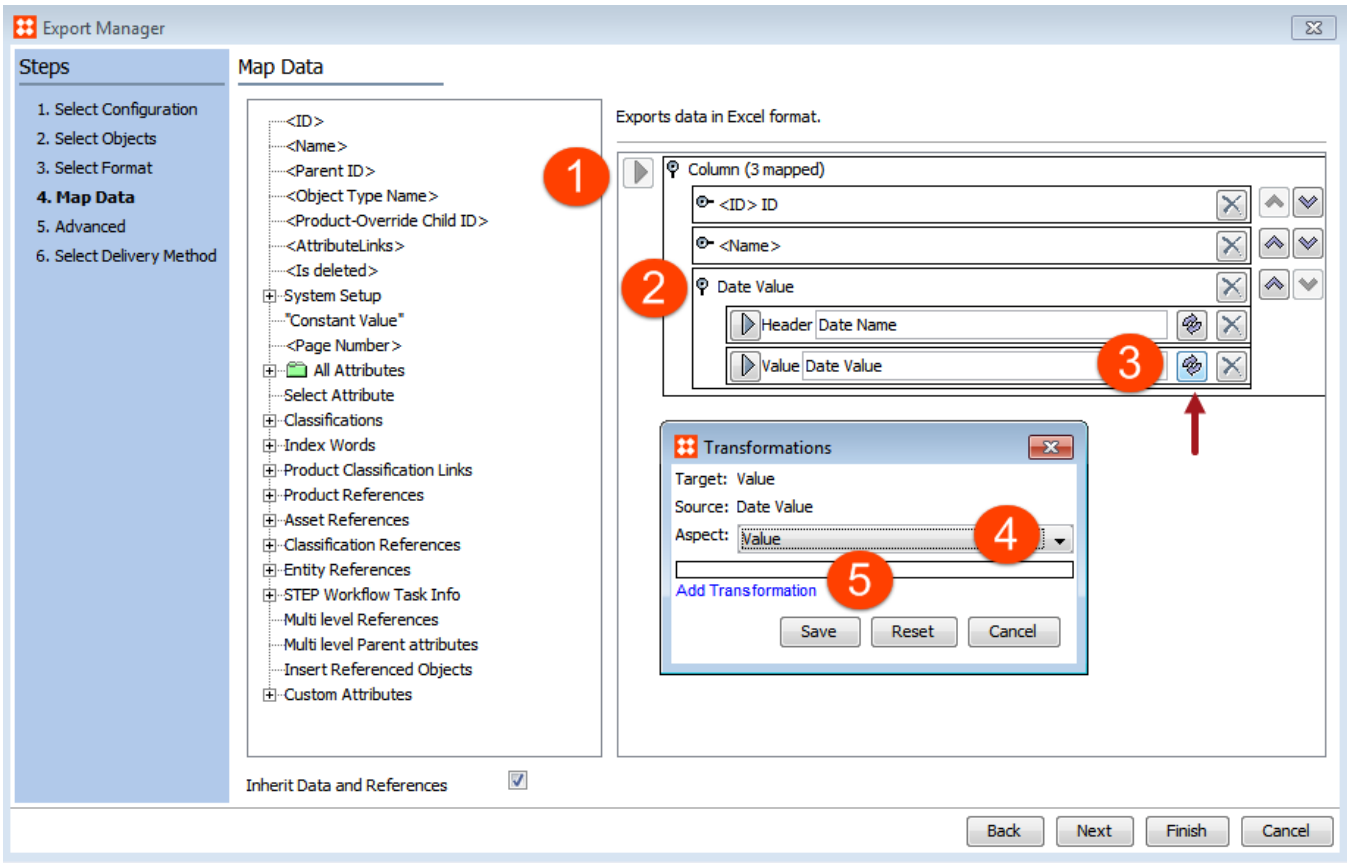
Date (and Time) Pattern Examples	United States Locale Example
yyyy.MM.dd G 'at' HH:mm:ss z	2015.08.14 AD at 11:25:40 EDT
EEE, MMM d, "yy	Fri, Aug 14, '15
dd.MM.yy	14.08.15

Export Transformations

The steps for exporting data using date transformations should be followed very carefully. If not, the export may fail and the entire process will need to be restarted.

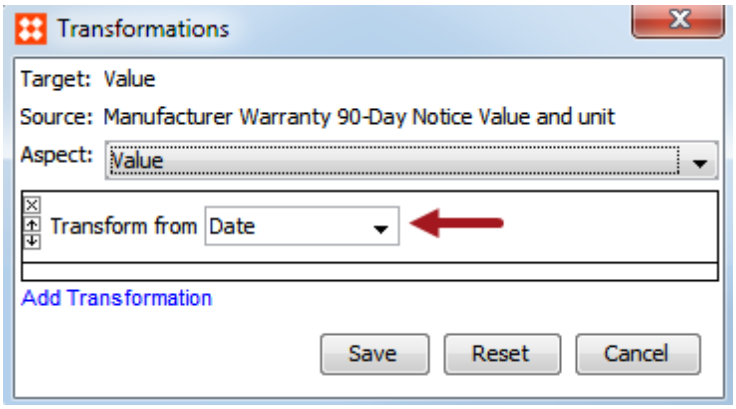
- Verify that the STEP Workbench Context is set correctly. This will impact locale transformations.
- Begin the export process by selecting all the applicable objects on the STEP Workbench Tree.
- Right-click while on a selected object, choose Export Data Below and then follow the export wizard steps. (Export can also be done by going to File > Export > Data and following the steps, including manually selecting the objects to export in Step 2 (Select Objects).)
- Next, make a format selection in Step 3 (Select Format) of the Export Manager.
- Transformations are set up in Step 4 (Map Data).

1. Begin by mapping the attributes to be exported. Pick the attributes from the tree and click the mapping icon.
2. Click the flipper to expand the options for the attribute you wish to transform. For date values, you will see a Header row and a Value row.
3. Click the transformations button to the right of the Value row.
4. A Transformations dialog box will appear. Change the Aspect to Value using the dropdown.
5. Click the Add Transformation link and make a 'Transform date' selection: Transform date by locale / Transform date by pattern. Follow the directions for the applicable selection and then continue on to the next step in these directions.



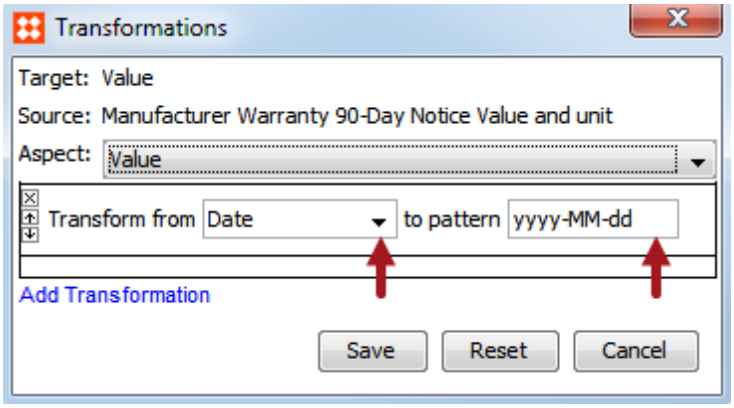
Transform date by locale

- Click on **Transform date by locale** and then click OK.
- From the Transformations screen, use the 'Transform from' dropdown to choose the validation base type of the date attribute being exporting.



Transform date by pattern

- Click on **Transform date by pattern** and then click OK.
- From the Transformations screen, choose the validation base type of the date attribute you are exporting (Date, ISODate, ISODateTIme). And, enter the desired pattern manually or copy and paste it from an external source. Click Save.



Once you have made a selection and picked the source date validation base type and pattern (if applicable), continue on with the rest of the steps.

6. Select Next to move to Export Manager Step 5 (Advanced). Make any additional selections, if desired. However, no additional configurations are required to complete the process. Click Next to move to Step 6 or Finish to skip ahead to the export. *(Once an end user becomes familiar with the transformation process, they may choose Finish after Step 4 of the Export Manager process and export data without any additional setting changes.)*
7. Optional: Make any Step 6 (Select Delivery Method) selections and click Finish.
8. Make selections on the Save Export Configuration screen, if desired. Click OK.
9. The 'Starting process Exporting' dialog will appear. Choose 'Go to process' to access the file.
10. On the Background Process screen, determine if the export was successful or not.
    - If the process failed, go back and try the steps again (making sure the correct attribute validation base type was selected on the Transformations screen).
    - If the process was successful and the delivery method was set to file, an export file (expand the Result flipper, if needed) will be available at the bottom of the Background Process page. Double-click the file name to open the file, or click the Save icon to the right of the file name to save the file without immediately opening it.
11. Verify the exported data is correct.

If transforming by locale, the correct locale format should be showing. The locale is determined by the context set in the workbench at the time the export process begins.

If doing a pattern transformation, verify the date format matches the pattern entered. If the date pattern includes a time zone, the time zone is determined by the location of your server. It is not impacted by context set in the workbench.

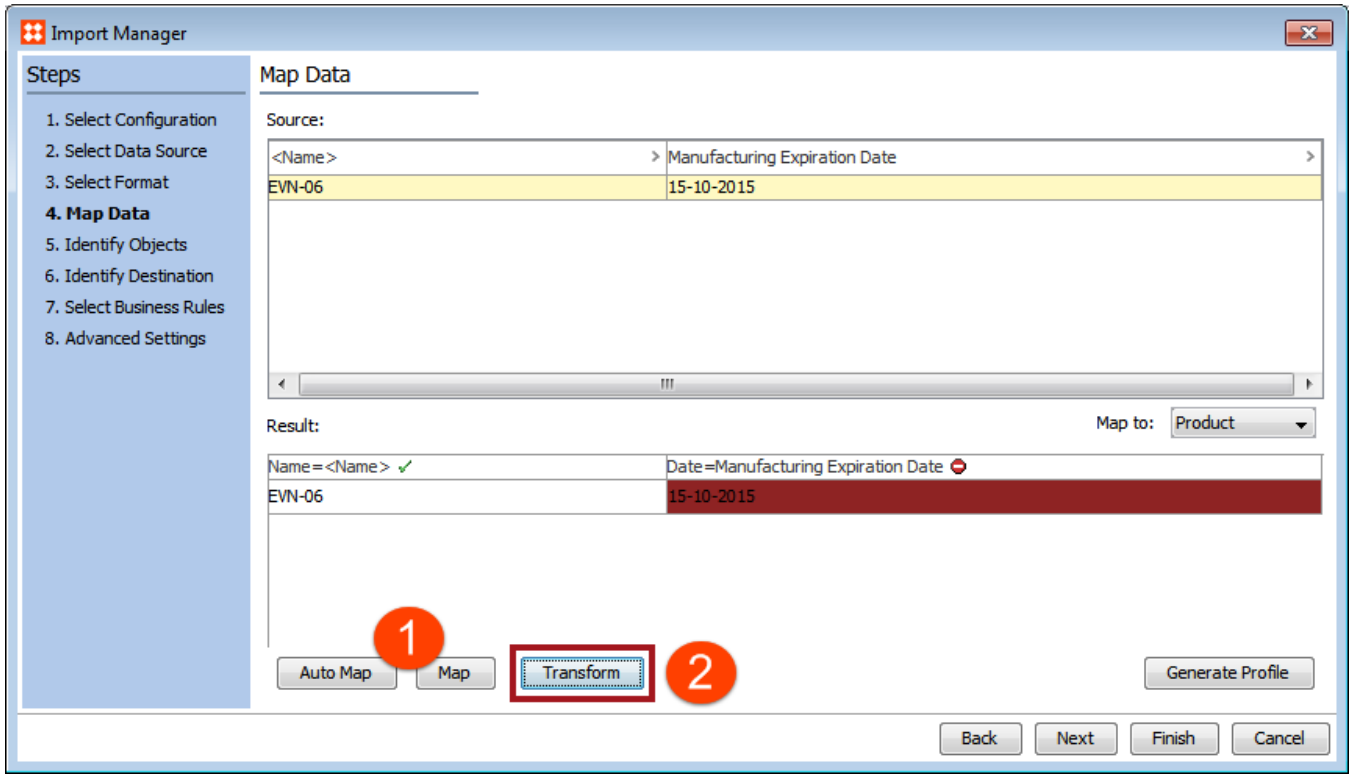
Import Transformations

The steps for importing data using date transformations should be followed very carefully. If not, the import may fail and the entire process will need to be restarted.

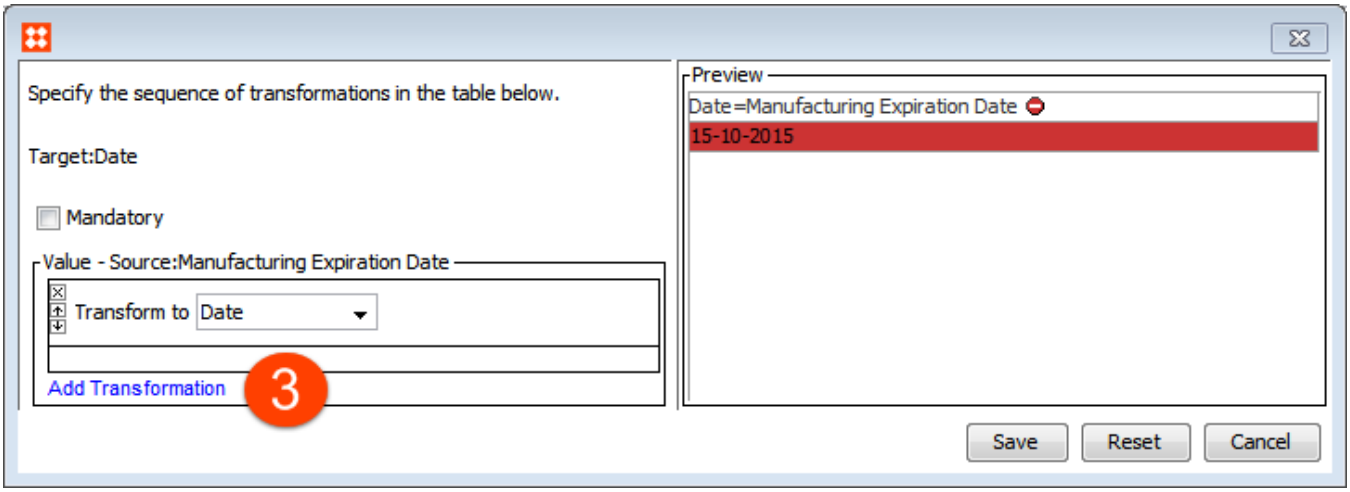
Important: Begin by making sure the date fields in the input file do not have any custom formatting. The best results will occur when importing dates that are in text format. Users should convert the date fields to text format prior to starting the import process.

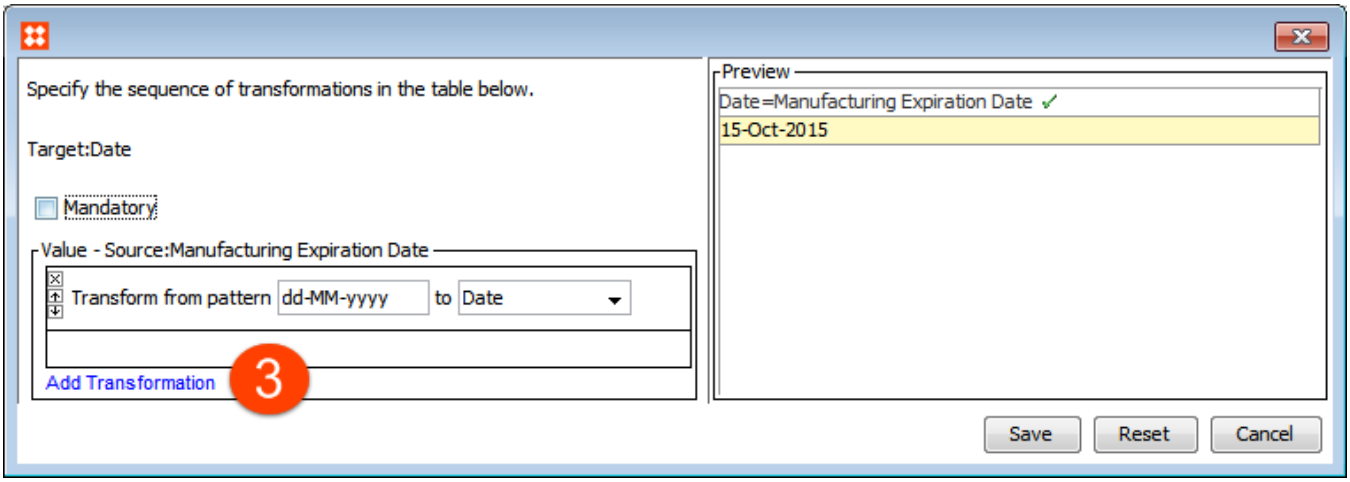
- Go to File > Import > Data and following the instructions for Step 1 (Select Configuration) and clicking Next.
- In Step 2 (Select Data Source), link the file you are importing by searching / selecting or manually typing in (or pasting) the file location.
- Next, make a format selection in Step 3 (Select Format).
- Just like the export process, transformations are set up in Step 4 (Map Data) of the Import Manager.

1. Begin data mapping by selecting Auto Map. If the date column is unmapped, then use Map to manually tie the attribute values to be imported to an existing attribute.
2. Under Result, select the column you want to transform during the import process. While the column is selected, click on the Transform button.



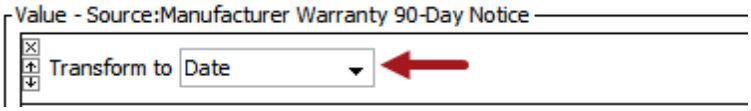
3. Click the blue Add Transformation link in the bottom left of the transformations screen. Make a selection: Transform date by locale / Transform date by pattern. Follow the directions for the applicable selection and then continue on with the remaining steps.





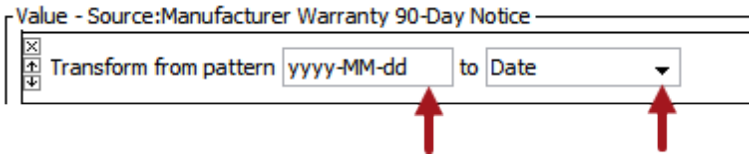
Transform date by locale

- Click on **Transform date by locale** and then click OK.
- Choose the validation base type of the target date attribute (Date, ISODate, ISODateTIme). Once you have picked the correct option, the Preview data will turn yellow. And, click Save.
- If you try all three selections and the Preview remains red, then you will need to use the Transform date by pattern option. To use the locale import option, the date values must match the locale format for the context that the workbench is currently set to.



Transform date by pattern

- Click on **Transform date by pattern** and then click OK.
- From the Transformations screen, enter the pattern of the date values you are importing. Then, tab over to or click on the dropdown for the validation base type of the target attribute. Make a selection from the dropdown or leave as-is. Click Save.
- The Preview section of the screen will turn yellow when the validation base type is correct. Before continuing make sure the Preview date is showing correctly and verify that the correct pattern is entered.



Once the transformation is set up and screen information is validated, continue on with the rest of the steps.

4. Select Next to move to Step 5 (Identifying Objects). If you are importing an ID field with the objects, the data will be automatically matched up with existing data. If you are importing data without an ID field, click a Source column and then press the Match button to match the source data up with existing STEP data. If you skip this step, you will create new objects in the workbench instead of importing into existing ones. Click Next.
5. Optional: Make any Step 6 (Identifying Destination), Step 7 (Select Business Rules) and Step 8 (Advanced Settings) selections. While selections are not required, it is recommended that the end user review each screen. *(Once an end user becomes familiar with the transformation process, they may choose Finish after Step 5 of the Import Manager process and import data without any additional setting changes.)*
6. Click Finish when ready for the import process to begin.
7. Choose Start Import Process on the Save Import Configuration dialog box.
8. The 'Starting process Importing' dialog will appear. Choose 'Go to process.'
9. On the Background Process screen, users can determine if the import was successful or not.
    - If the process failed, go back and try the steps again (making sure the correct attribute validation base type was selected on the Transformations screen and the identification was done properly in Step 5 of the Import Manager).
    - If the process was successful, the user will see an 'import completed' message at the bottom of the process log.
10. Verify the imported data is correct.

Considerations and Limitations

- Date localization and transformation works on attributes with these validation base types: Date (DD-MON-YYYY), ISO Date (YYYY-MM-DD), and ISO Date and Time (YYYY-MM-DD HH24:MI:SS).
- The steps for import and export must be followed carefully or the process may fail.

Details about available Attribute Transformations can also be found in the System Setup / STEP Super User documentation.

For additional help importing data, see the Importing Data section of the System Setup / STEP Import Manager User Guide documentation. And, for additional help exporting data, see the Exporting Data section of the System Setup / STEP Export Manager User Guide documentation.

For information regarding converting attributes from Date type to ISO Date type, see the Maintaining Attributes section of the System Setup / STEP Super User documentation.

Calculated Attributes

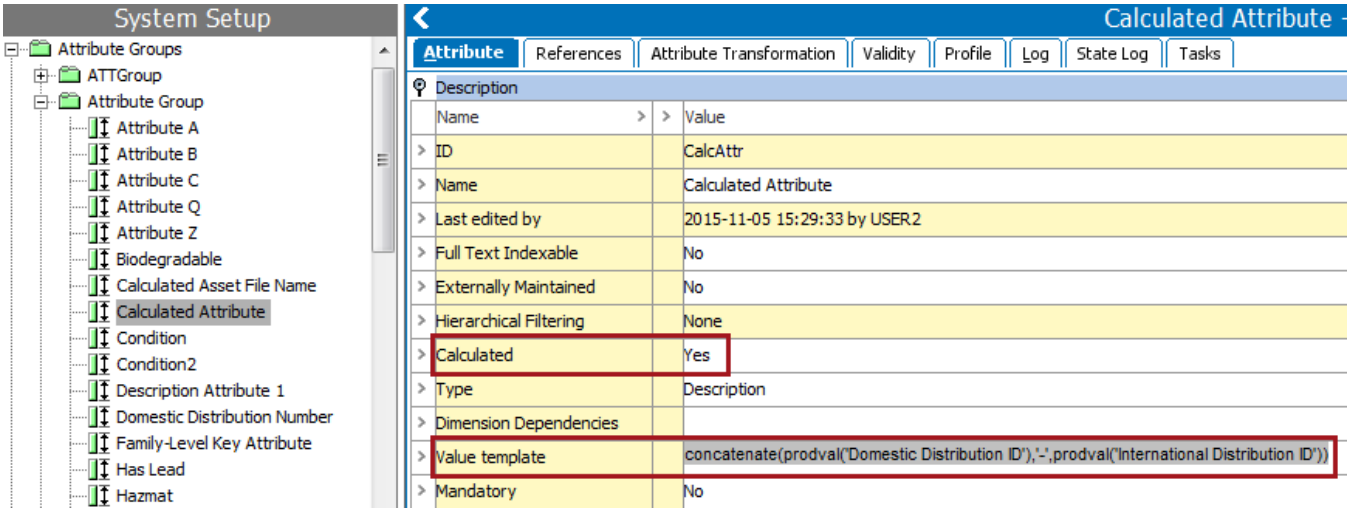
A calculated attribute is an attribute whose value is determined by a value template that includes a formula of functions and arguments and/or static text. A value template can also give access to information from references, compare data between objects, round numbers, modify text, and many other options, which are defined in the **Calculated Attribute Functions Overview** section of the **System Setup / STEP Super User Guide**.

Calculated attributes are created in the workbench System Setup attribute editor using the Calculated field on the Attribute tab and the Web UI Attribute Management screen. Both description attributes and specification attributes can be set as calculated attributes.

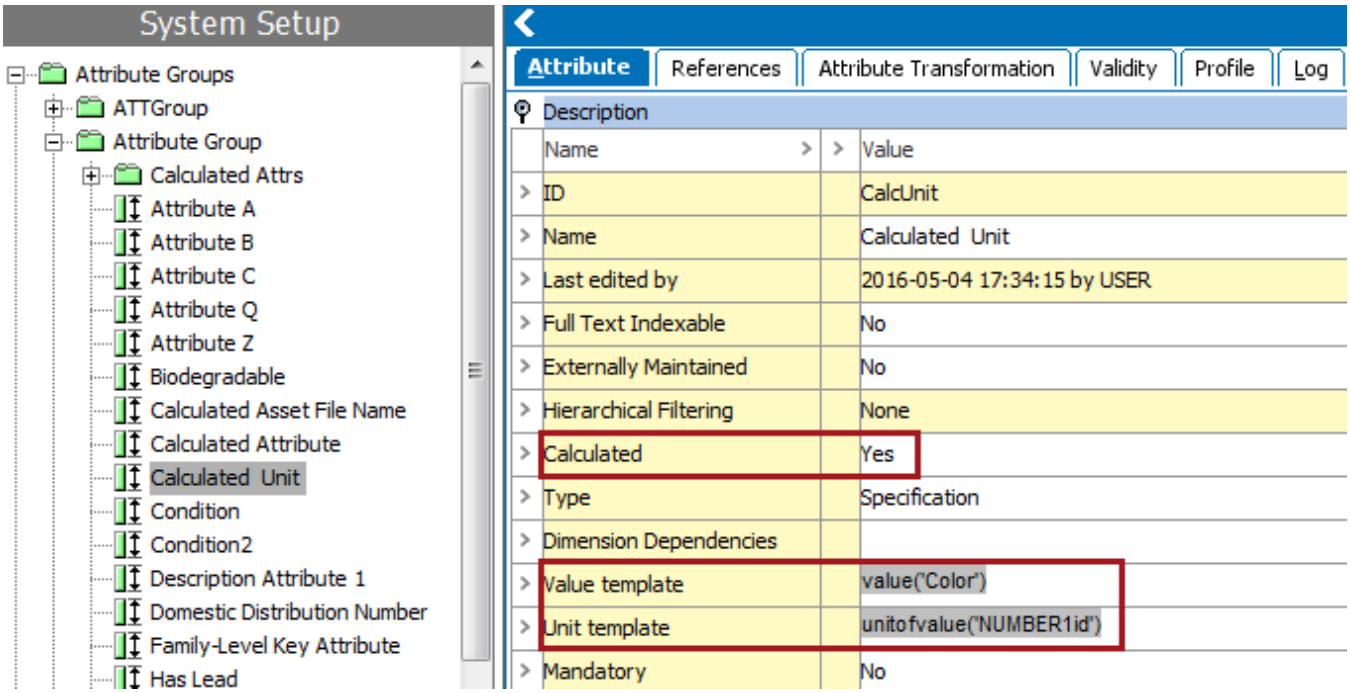
When an attribute is set as calculated, a Value template field displays.

Functions perform predefined actions on the argument(s) that are supplied. Arguments are placeholders for data (like an attribute ID or text that should be appended to a value). The following functions and arguments are included in the workbench example below:

- The 'concatenate' function takes a comma-delimited list of arguments. In this example, the three arguments are: prodval('Domestic Distribution ID'), "-", and prodval('International Distribution ID').
- The 'prodval' function takes an argument of an attribute ID and retrieves the value of the attribute identified. In this example, the two attribute IDs are: 'Domestic Distribution ID' and 'International Distribution ID'.
- Assuming that the 'Domestic Distribution ID' value is 1234 and the 'International Distribution ID' value is 5678, the result of this Value template would be 1234-5678.



When a calculated attribute uses the validation base type of Embedded Number, Fraction, Integer, Number, or Numeric Text, a **Unit template** field displays. This field can hold a static Unit ID and/or an expression that evaluates to a Unit ID.



For more information, see the **Calculated Attributes in Web UI** section of the **Web User Interfaces** documentation.

Differences Between Calculated Attributes and Standard Attributes

- Multi-valued attributes and attributes with LOV validation cannot be set as calculated.
- An attribute with 'Calculated = Yes' should also have 'Mandatory = No' since the actual value is not maintained in the database.
- Values of calculated attributes are not available using Search since the value does not exist until the user views the object or the object is exported.
- The sharedconfig.properties file setting 'Calculated.UseLegacyLocalization=true' enables calculated attributes to honor Context Locale settings. For more information about contexts and locales, see the **Contexts** section of the **System Setup / STEP Super User Guide** documentation.

Value Template Expression Examples

This section provides a way to see the relationship between the values used by an expression, the value template expression itself, and the resulting calculated attribute values.

Attribute Values

The expression examples in this section use the following set of attribute values:

Attribute	Height	Width	Length	Color Front	Color Back
Value	22 cm	39 cm	54 cm	Black	White

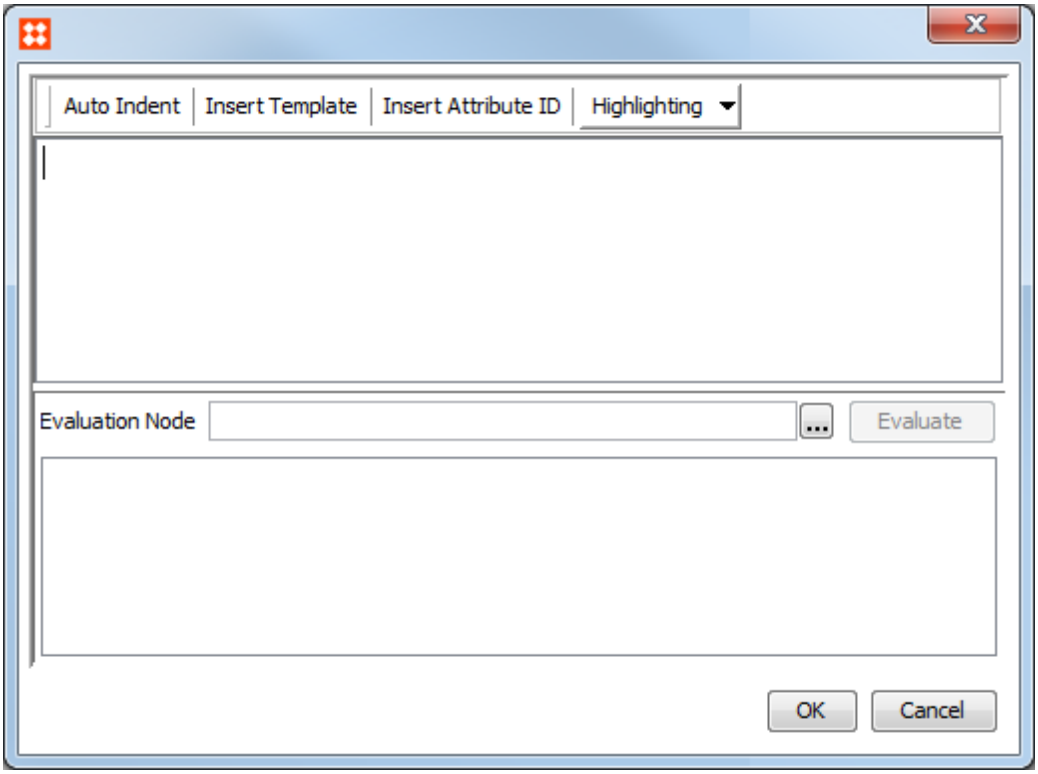
Expression Examples

Attribute	Value Template - Expression	Description	Resulting Calculated Attribute Value
Dimensions	CONCATENATE (PRODVAL (“height”),” “,PRODUNIT (“height”),” x “,PRODVAL (“width”),” “,PRODUNIT (“width”), “ x “, PRODVAL (“length”),” “, PRODUNIT (“length”))	Displays the dimension of the current product in the form 'Height x Width x Length'	22 cm x 39 cm x 54 cm
Volume	PRODVAL (“height”,”cm”) * PRODVAL (“width”,”cm”) * PRODVAL (“length”,”cm”)	Calculates and displays the volume of the current product.	46332 cm^3
Volume (Liter)	TEXT (PRODVAL (height”,”cm”)*PRODVAL (“dth”,”cm”)*PRODVAL gth”,”cm”)/1000,”#.0”)	Calculates the volume in Liter and displays with one decimal.	46.3
Form	IF (PRODVAL (“length”, “cm”) = PRODVAL (“width”,”cm”),”Square”, “Not Square”)	Displays if the bottom of the box is square or not.	Not Square
Color scheme	IF (EXACT (PRODVAL (“Front color”),PRODVAL (“Back color”)),”Same color front and back”,”Different colors front and back”)	Compares the front and back colors of the current product and displays the result.	Different colors front and back
Size for standard package	IF (AND (PRODVAL (height”, “cm”) <= 40, PRODVAL (“width”,”cm”) <= 40), PRODVAL (“length”,”cm”) <= 40),”Size OK for standard package”, “Too big for standard package”)	Since a delivery service charges extra for oversized packages, this could be used to determine if a dimension exceeds 40 cm.	Too big for standard package
Size for	IF (AND (OR (AND (PRODVAL (“height”, “cm”) <= 23, PRODVAL	An air carrier has a limit on the size of the	Size OK for

Attribute	Value Template - Expression	Description	Resulting Calculated Attribute Value
cabin baggage	(“length”, “cm”) <= 40, PRODVAL (“width”, “cm”) <= 55), AND (PRODVAL (“height”, “cm”) <= 23, PRODVAL (“length”, “cm”) <= 55, PRODVAL (“width”, “cm”) <= 40), AND (PRODVAL (“height”, “cm”) <= 40, PRODVAL (“length”, “cm”) <= 55, PRODVAL (“width”, “cm”) <= 23), AND (PRODVAL (“height”, “cm”) <= 40, PRODVAL (“length”, “cm”) <= 23, PRODVAL (“width”, “cm”) <= 55), AND (PRODVAL (“height”, “cm”) <= 55, PRODVAL (“length”, “cm”) <= 23, PRODVAL (“width”, “cm”) <= 40), AND (PRODVAL (“height”, “cm”) <= 55, PRODVAL (“length”, “cm”) <= 40, PRODVAL (“width”, “cm”) <= 23)), PRODVAL (“height”, “cm”) + PRODVAL (“length”, “cm”) + PRODVAL (“width”, “cm”) <= 115), “Size OK for cabin baggage”, “Too Big for cabin baggage”)	cabin baggage. In this example, 1) max dimension allowed is 23 x 40 x 55 cm, and 2) the sum of these must be max 115 cm. This calculated attribute delivers the result of the calculation of size with respect to both rules.	cabin baggage
Size for special storage	IF (AND (OR (PRODVAL (“length”) > 40, PRODVAL (“width”) > 40), NOT (AND (PRODVAL (“length”) > 40, PRODVAL (“width”) > 40))), “OK - One and only one dimension > 40”,”Not OK - none or two dimensions > 40”)	A special storage requires that the product has either width or length greater than 40 cm, but not both.	OK - One and only one dimension > 40

Function Editor Overview

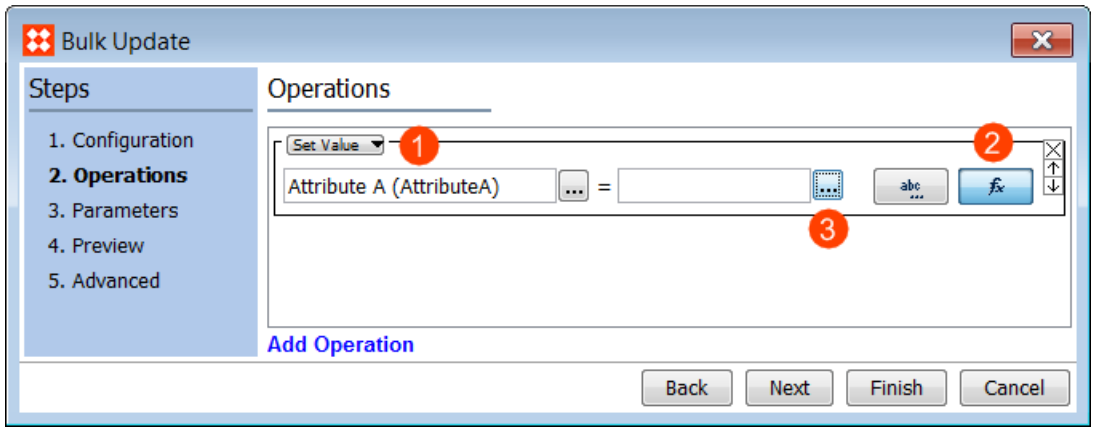
With the Function Editor, in addition to writing functions manually, you can create complex functions by using one or more predefined templates. In addition, you can choose to automatically highlight function names, automatically indent lines in the function, and evaluate part of a function or an entire function against objects in the Tree.



Accessing the Function Editor

The Function Editor is available in the following locations:

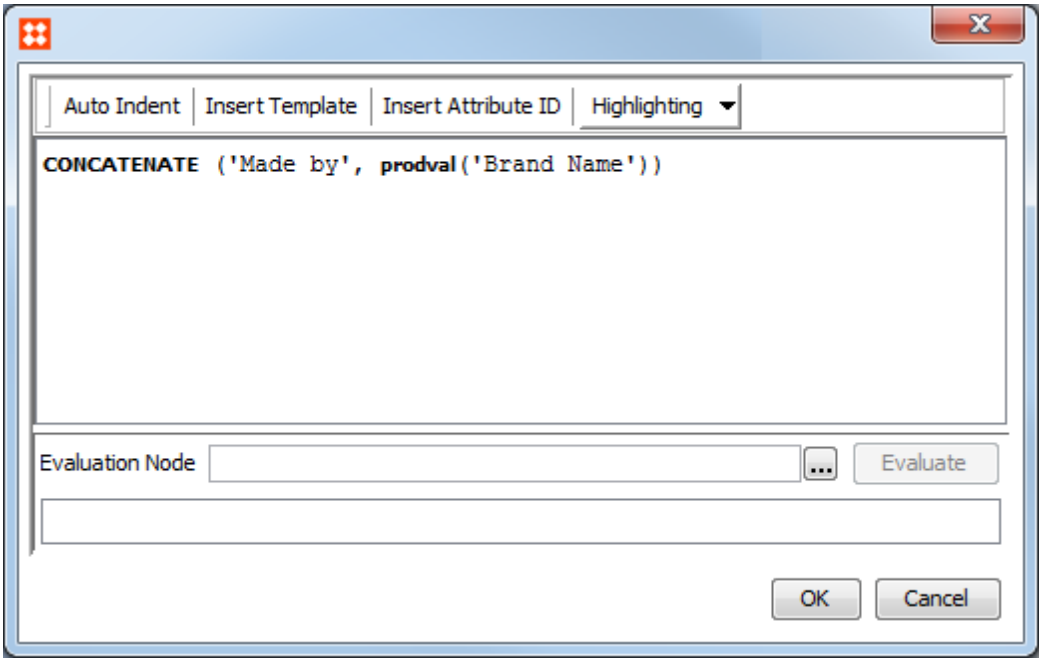
1. **Attribute editor**
   - In System Setup, navigate to an attribute that has been configured as calculated.
   - On the **Attribute** tab under the Description flipper, click the 'Value template' field for editing.
   - Right-click the 'Value template' field and choose **Insert Function**.
2. **Bulk Update wizard**
   - On step 2 **Operations**, select the 'Set Value' operation, click the Function button (), and then click the ellipsis button **(...)**.



3. **Flatplanner**
   - When working with Flatpanner publications, you can add a calculation option to the Basket view, Financial Summary view, and/or the Sticker Book view. For more information, see the **Customized Views** section of the **Flatplanner** documentation.

Using the Function Editor

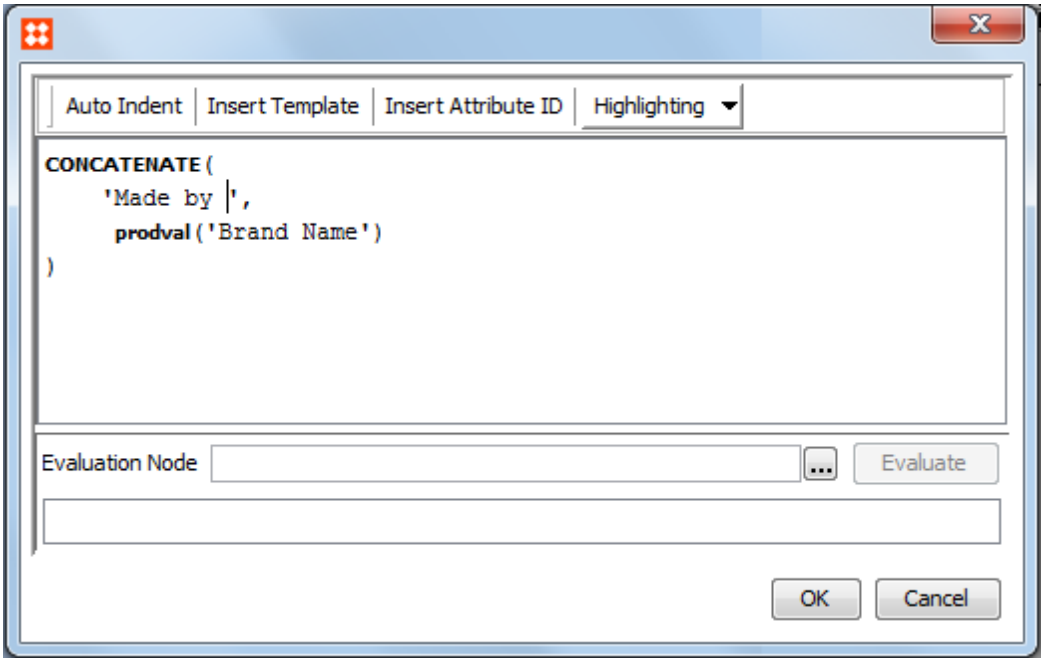
If you know the name and required arguments of the needed function, type directly into the upper text field. For more information on the available functions and their syntax, see the **Calculated Attributes** documentation for **Calculated Attributes Functions Overview** or for **Inserting Templates**.



Function Editor Options

The following options can be accessed in the Function Editor either by clicking on the menu tabs across the top or by right-clicking inside the upper text box and choosing from the popup menu.

- **Auto Indent:** Displays the function in a typical coding layout. Indents and hard returns are added for an easier-to-read format, as shown below.



- **Insert Template:** Allows a preconfigured calculated attribute function template to be inserted at the current location of the cursor. For more information, see the **Calculated Attributes** documentation for **Inserting Templates**.
- **Insert Attribute ID:** Displays an attribute browse / search dialog, and inserts the attribute ID of the selected attribute at the current location of the cursor. For more information, see the **Calculated Attributes** documentation for **Inserting Attribute IDs in Functions**.
- **Highlighting:** Displays or removes highlighting for parentheses and/or bold for function names.
- **Evaluate:** Tests part of a function or an entire function against objects in the Tree. For more information, see the **Calculated Attributes** documentation for **Evaluating Functions**.

Creating Function Templates

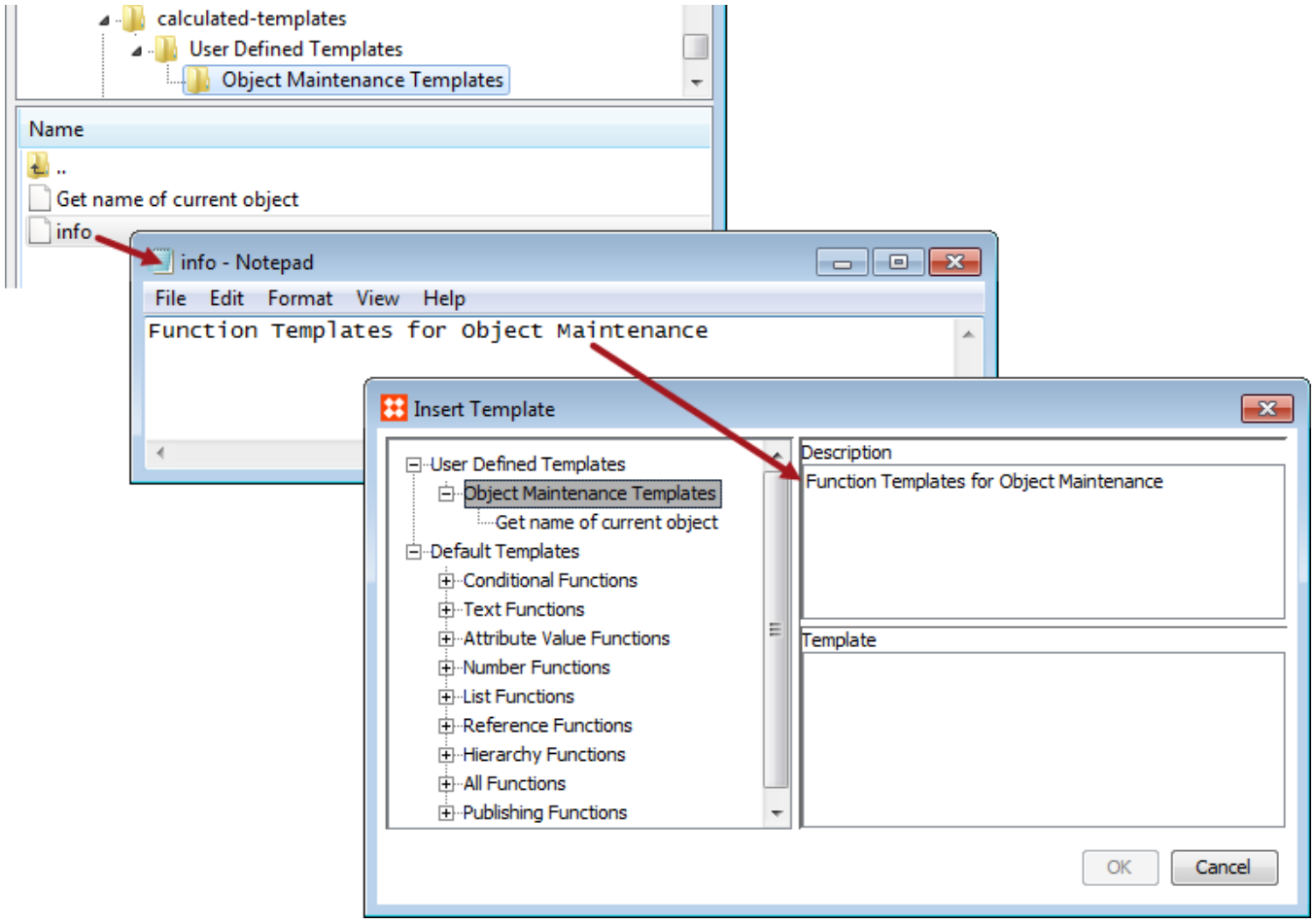
STEP includes default function templates, but also allows you to create your own.

User-defined function templates are stored in the directory identified by the 'Calculated.TemplateRoot' system property in the sharedconfig.properties file. The directory is on the application server, or when using a clustered setup, on a shared drive. For example, on a Linux system, the property could look like this:

```
Calculated.TemplateRoot=/shared/calculated-templates
```

The following images show the relationship between the folder / file levels and the information displayed in the **Insert Template** dialog. Within the function template directory, create as many levels with corresponding descriptions as needed.

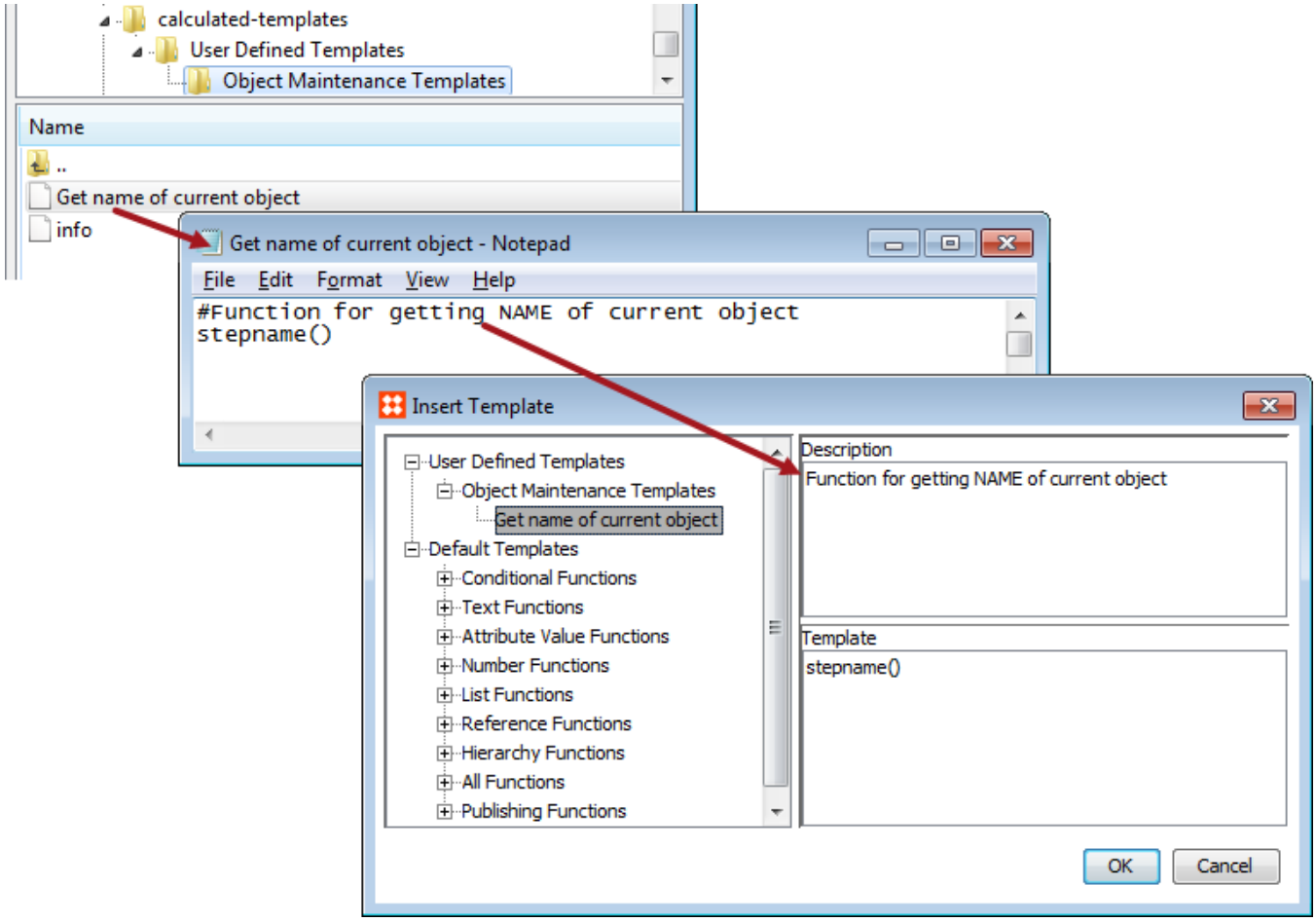
- To add a new level in the Insert Template dialog tree, create a new directory below the root, such as 'Object Maintenance Templates.'
- To display a **Description** for a *directory* in the Insert Template dialog, create a file named 'info' in the directory, and include the description text in it, such as 'Function Templates for Object Maintenance'.



The following images show the relationship between the template file and the information displayed in the **Insert Template** dialog. Within the function template directory, create a template file as follows:

- The name of the template file appears in the Insert Template dialog tree. For example, 'Get name of current object'.
- To display a **Description** for the *template file*, start a line with a number sign (#) and add description text. For example, 'Function for getting NAME of current object'.
- To add the function within the template, start on a new line. The function displays in the **Template** area of the Insert Template dialog. For example, 'stepname()'.

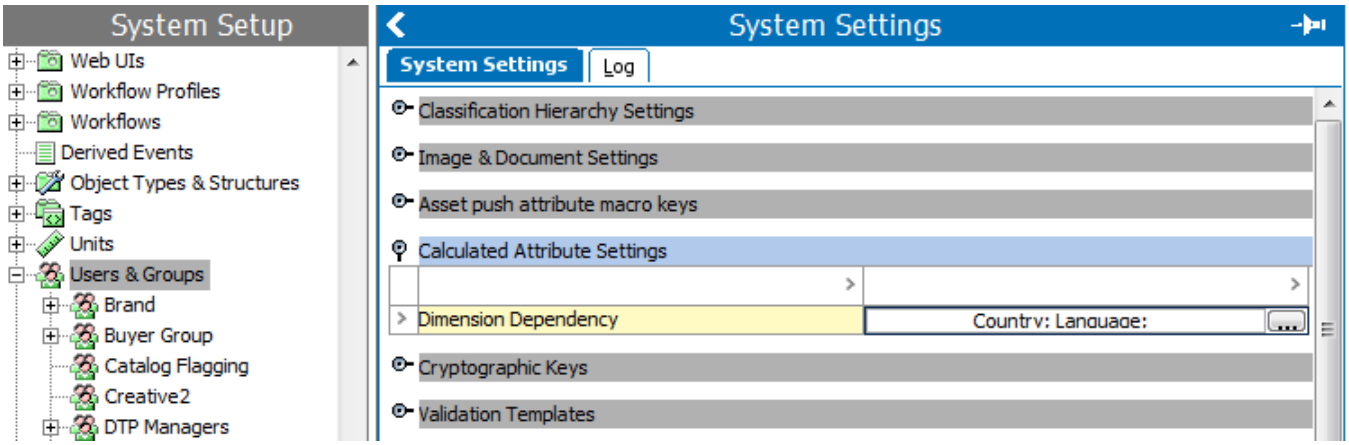
The following image shows the template setup and details in the Insert Template dialog.



Dimension Dependencies

System Setup for Users & Groups includes a global setting to make all calculated attributes dimension dependent. This allows value templates to differ across contexts by using a different formula template for the same attribute, based on the selected dimension.

Note: If translation of a value template is required, ensure that the expression syntax is valid.



For example, when calculating 'width x depth x height' using the language dimension dependency,

- The following value template for English could return uppercase letters for width, depth, and height as '12W x 16D x 24H':

```
Concatenate(prodval('Width'),'W x ', prodval('Depth'), 'D x ', prodval('Height'), 'H')
```

- The following value template for French could return the prefered lowercase letters for largeur, profondeur, and hauteur as '12l x 16p x 24h':

```
Concatenate(prodval('Width'),'l x ', prodval('Depth'), 'p x ', prodval('Height'), 'h')
```

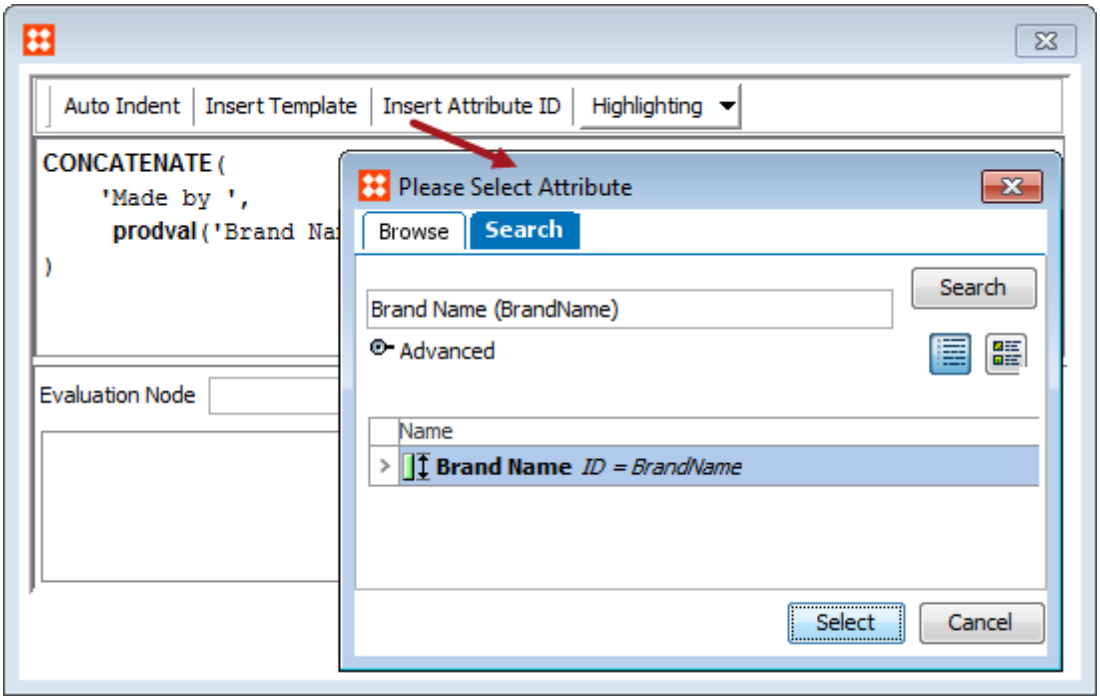
For more information about setting a calculated attribute to be dimension dependent, see the 'Calculated Attribute Settings' within the **Users and Groups About System Settings** section of the **System Setup / STEP Super User Guide** documentation.

Inserting Attribute IDs in Functions

Many functions accept an Attribute ID as an argument. The **Insert Attribute ID** button allows you to locate the relevant attribute through a browse / search dialog instead of manually entering the ID. This helps to ensure that the correct item is being used in the function by limiting user error.

When referencing attributes and/or units in the Value Template or the Function Editor, always use the ID (not the Name) to precisely locate the appropriate attribute or unit.

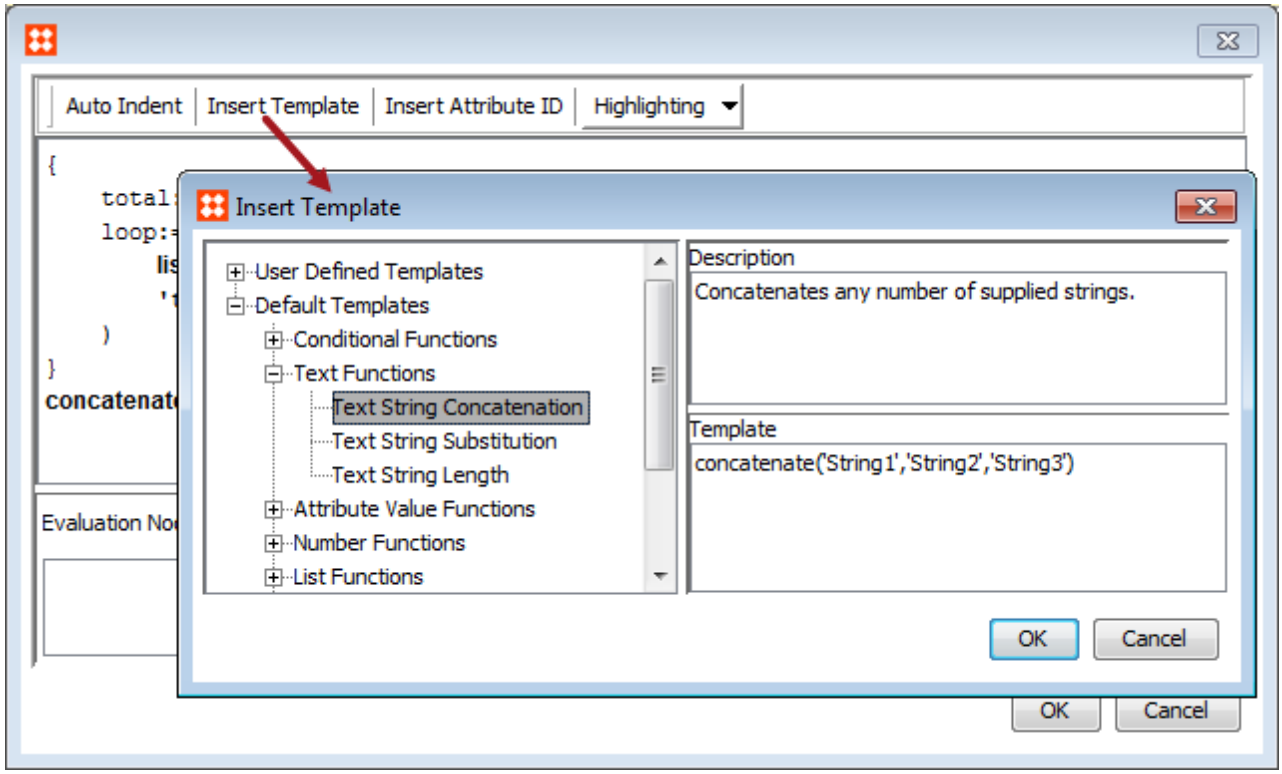
To display the attribute selector dialog, click the Insert Attribute ID button or right-click and select the option from the menu. Use browse or search to select the attribute. The Attribute ID is inserted at the cursor position.



Inserting Templates

While on the Function Editor, for assistance with available functions, or to use a predefined Function Template, click the **Insert Template** button or right-click and select the option from the menu.

The **Insert Template** dialog is divided into three areas as shown below:



- The left panel displays a tree structure that contains the function templates provided with STEP and user-created templates.
- The Description area displays the provided description of the template selected from the tree area.
- The Template area displays the actual template that will be inserted at the position of the cursor.

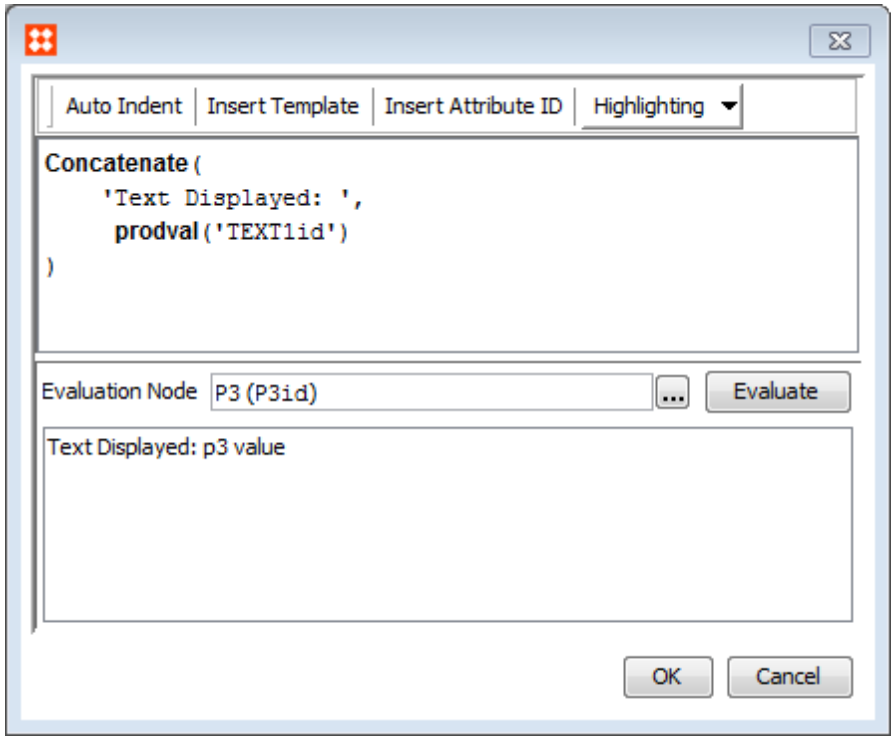
For information on creating user defined templates, see **Creating Function Templates** in the **System Setup / STEP Super User Guide**.

Evaluating Functions

Evaluating allows you to test that the syntax and structure of a function is correct.

1. On the Function Editor, select the object to evaluate against using one of the following options:
    - type in the **Evaluation Node** text field and select from the type ahead suggestion list displayed
    - click the ellipsis button **(…)** and search or browse to select the object using the **Evaluation Node** dialog.
2. Click the **Evaluate** button. The result appears in the bottom text box. In the following example, the result is 'Text Displayed: p3 value'

Note: To evaluate only a specific part of the function, highlight it and click the Evaluate button.



The Function Editor is available in several locations within STEP. For more information, see the **Function Editor Overview** section of the **System Setup / STEP Super User Guide** documentation.

Calculated Attribute Functions Overview

- A calculated description attribute is available to objects based on the object type, just as any other non-calculated description attribute. A calculated description attribute can also be added to a classification (or asset), however any products linked to the classification will not inherit the formula, just as a product would not inherit a non-calculated description attribute on a classification.
- A calculated specification attribute can be available on any object, meaning that the calculated attribute formula will inherit down to the subordinate objects. The value will be displayed based on the attribute(s) at the level where the calculation is displayed, considering any overridden values.

Calculated Value Override

When a calculated attribute displays on an object, the value is read only. Occasionally, the calculated value for a specific product may not be correct. To manually override the value on specific product (and child products for specification attributes, if the value is inherited), right-click the calculated value and update it.

To restore the calculated value, simply delete the manually entered value.

Available Function Types

Additional information for each of the calculated value function types is included in the following sections:

- Attribute Value Functions
- Comments
- Commercial Value Functions
- Conditional Functions
- Date Functions
- List Functions
- Number Functions
- Other Functions
- Text Functions
- Variables

Attribute Value Functions

Calculated Attribute expressions can be written using the following available functions. Examples follow the Available Functions table.

Note: Text used for 'attribute-id' is case-sensitive.

Available Functions

Function	Parameters	Description
PRODVAL or VALUE	('attribute-id')	Return the value of the supplied attribute (attribute-id) for the current object. This also works for non-products, for example, by using the iterate function to get another current object.
PRODVAL or VALUE	('attribute-id', 'unit-id')	Return the value of the supplied attribute (attribute-id) converted into the supplied unit (unit-id) for the current object. This also works for non-products, for example, by using the iterate function to get another current object. In order to convert units, relevant units must have a Base Conversion defined in the Units area of System Setup.
PRODVALSIMPLE or SIMPLEVALUE	('attribute-id')	Return the value of the supplied attribute (attribute-id) for the current object, concatenated with the name of the unit (if any). For embedded numbers, this function returns a string like: suffix value unit-name prefix.
PRODVALSIMPLE or SIMPLEVALUE	('attribute-id', 'unit-id')	Return the value of the supplied attribute (attribute-id), converted into the supplied unit (unit-id) for the current object, concatenated with the name of the resulting unit. For embedded numbers, this function returns a string like: suffix value unit-name prefix. In order to convert units, relevant units must have a Base Conversion defined in the Units area of System Setup.
PRODUNIT or UNITOFVALUE	('attribute-id')	Return the name of the unit of the value of the supplied attribute (attribute-id) for the current object. The expression cannot evaluate the Unit from other Calculated Attributes.
PRODUNITID or UNITIDOFVALUE	('attribute-id')	Return the ID of the unit of the value of the supplied attribute (attribute-id) for the current object. The expression cannot evaluate the Unit from other Calculated Attributes.
ATTRNAME	('attribute-id')	Return the name of the attribute (attribute-id).
GETEMBEDDEDPREFIX	(PRODVAL ('attribute-	Return the prefix part of an embedded number value. If (attribute-id) is not an embedded number, an empty text string is returned (rather than 'NA' which is returned by some functions).

Function	Parameters	Description
	id'))	For more information about embedded numbers, see the **Validation Rules** of the **System Setup / STEP Super User Guide** documentation.
GETEMBEDDEDSUFFIX	(PRODVAL ('attribute-id'))	Return the suffix part of an embedded number value. If (attribute-id) is not an embedded number, an empty text is returned. For more information about embedded numbers, see the **Validation Rules** of the **System Setup / STEP Super User Guide** documentation.
GETEMBEDDEDVALUE	(PRODVAL ('attribute-id'))	Return the value part of an embedded number value. If (attribute-id) is not an embedded number, an empty text is returned. For more information about embedded numbers, see the **Validation Rules** of the **System Setup / STEP Super User Guide** documentation.
UNITCONV	(value, unit-id, to-unit-id)	Return number of 'value' in 'unit-id' converted into 'to-unit-id'.
VALUELOVID	(attribute-id)	Return the LOV ID of the LOV value of the supplied attribute (attribute-id) for the current object, if Value IDs are used on the LOV. If 'Use Ids on values' is set to 'No', N/A is returned.

Examples

The examples below assume a setup including these **attributes** and **products**:

ATTRIBUTE Type	ID	Name	AttrDesc
Text-validated product attribute	TEXT1id	TEXT1	TEXT1 description
Number-validated product attribute with length units	NUMBER1id	NUMBER1	NUMBER1 description
Embedded number-validated product attribute with length units	EMBEDDEDNUMBER1id	EMBEDDEDNUMBER1	EMBEDDEDNUMBER1 description
Text-validated attribute property	AttrDescid	AttrDesc1	AttrDesc description
LOV-validated product attribute	LOVAttr1id	LOVAttr1	LOV description
LOV with 'Use IDs on Values' = Yes; and Values =Yes and No	LOV1id	LOV1	

PRODUCTS ID	Name	Parent	TEXT1id	NUMBER1id	EMBEDDEDNUMBER1id	LOVAttr1id
P-family-id	P-family	Product hierarchy root	family value	2 m		
P1id	P1	P-family-id	p1 value	4 m		
P2id	P2	P-family-id	p2 value	5 m		No
P3id	P3	P-family-id	p3 value	6 m	prefix 3 m suffix	Yes

Copy and paste any of the following examples into the Function Editor and then evaluate against the specified object to get the defined result.

```
prodval('TEXT1id') // = "family value" when evaluated on P-family-id
prodval('NUMBER1id') //  = "2" evaluated on P-family-id
produnit('NUMBER1id') // = "m" evaluated on P-family-id
prodval('NUMBER1id', 'unece.unit.MMT') // = "2000" evaluated on P-family-id
prodvalsimple('NUMBER1id') // = "2 m" evaluated on P-family-id
prodvalsimple('NUMBER1id', 'unece.unit.MMT') // = "2000 mm" evaluated on P-family-id
attrname('NUMBER1id') // = "NUMBER1" evaluated on P-family-id
getembeddedprefix(prodval('EMBEDDEDNUMBER1id')) // = "prefix" evaluated on P3id
getembeddedvalue(prodval('EMBEDDEDNUMBER1id')) // = "3" evaluated on P3id
getembeddedsuffix(prodval('EMBEDDEDNUMBER1id')) // = "suffix" evaluated on P3id
prodvalsimple('EMBEDDEDNUMBER1id') // = "prefix 3 m suffix" evaluated on P3id
prodvalsimple('EMBEDDEDNUMBER1id', 'unece.unit.MMT') // = "prefix 3000 mm suffix" evaluated
on P3id
produnit('EMBEDDEDNUMBER1id') // = "m" evaluated on P3id
unitconv('2','unece.unit.MTR','unece.unit.MMT') // = "2000" evaluated on P3id
valuelovid('LOV1Attrid') // = "LOV1id" evaluated on P3id
```

Comments

Calculated Attribute expressions can be written using the following available functions. Examples follow the Available Functions table.

Available Functions

Type	Description
multi line	Commented text that spans multiple lines. /* is used to start the comment text */ is used to conclude the comment text
inline	Commented text on the current line, does not continue on the new line. // is used to start comment text for the current line

Examples

Copy and paste any of the following examples into the Function Editor and then evaluate against the specified object to get the defined result.

```
prodval('TEXT1id') // This text is a comment
prodval('NUMBER1id') /* This text is a comment and
commented text continues on each line until the concluding
comment indicator is reached.*/
```

Commercial Value Functions

Calculated Attribute expressions can be written using the following available functions. Examples follow the Available Functions table.

Available Functions

Note: The [effective date] parameter uses ISO date time format (YYYY-MM-DD 24:MI:SS). If not supplied, the current date / time are assumed.

Function	Parameters	Description
TERMVAL	('termslisttype-id', 'break-no'[, effectivedate])	Returns the term value of the supplied 'termslist type id, ' for the current publication version in the supplied 'break no' for the given effective date. [effective date] is optional.
TERMUNIT	('termslisttype-id', 'break-no'[, effectivedate])	Returns the name of the unit of the term of the supplied 'termslist type id' for the current publication version in the supplied break-no for the given effective date. [effective date] is optional.
TERMUNITID	('termslisttype-id', 'break-no'[, effectivedate])	Returns the ID of the unit of the term of the supplied 'termslist type id' for the current publication version in the supplied break-no for the given effective date. [effective date] is optional.
TERMQTYMAX	('termslisttype-id', 'break-no'[, effectivedate])	Return the maximum quantity of the supplied 'termslist type id' for the current publication version in the supplied break-no for the given effective date. [effective date] is optional.
TERMQTYMIN	('termslisttype-id', 'break-no'[, effectivedate])	Return the minimum quantity of the supplied 'termslist type id' for the current publication version in the supplied break-no for the given effective date. [effective date] is optional.
TERMVALID	('termslist-id', 'break-no'[, effectivedate])	Return the term value of the supplied 'termslist id' for the current publication version in the supplied break-no for the given effective date. [effective date] is optional.

Examples

Term List ID = TermListID			Term List ID = TermListID			
Product Name	Value	Unit - name(id)	Min Qty	Max Qty	Start	End
P&G Party Hat 1	$7.79	$(iso4217.unit.USD)	1	11	Jan-1-15	Dec-31-15
P&G Party Hat 2	$2.79	$(iso4217.unit.USD)	12	24	Jan-1-15	Dec-31-16
B&Y Party Hat 1	£6.79	£ (iso4217.unit.GBP)	1	11	Jan-1-15	Dec-31-17
B&Y Party Hat 2	£3.79	£ (iso4217.unit.GBP)	12	24	Jan-1-15	Dec-31-18

Copy and paste any of the following examples into the Function Editor and then evaluate against the specified object to get the defined result.

```
termval('PriceID', '1', '2015-12-31 01:01:01') // returns $7.79 for P&G Party Hat 1
termval('PriceID', '2', '2015-12-31 01:01:01') // returns $2.79 for P&G Party Hat 2
termunit('PriceID', '1') //returns $ for P&G Party Hat 1; returns £ for B&Y Party Hat 1
termunitid('PriceID', '1') //returns 'iso4217.unit.USD' for P&G Party Hat 2; returns 'iso42
17.unit.GBP' for B&Y Party Hat 2
termqtymax('PriceID', '1') // returns 12 for B&Y Party Hat 2
termqtymin('PriceID', '1') // returns 24 for B&Y Party Hat 2
termvalid('TermListID', '1') // returns 24 for B&Y Party Hat 2
```

Conditional Functions

Calculated Attribute expressions can be written using the following available functions. Examples follow the Available Functions table.

Available Functions

Conditional Function	Parameters	Description
IF	(condition, 'value_if_true', 'value_if_ false')	If 'condition' is true, then 'value_if_TRUE' is returned, otherwise 'value_if_FALSE' is returned.

Examples

Copy and paste any of the following examples into the Function Editor and then evaluate to get the defined result.

```
if (1=1, 'true', 'false') // = "true"
if (not(1=1), 'true', 'false') // = "false"
if (exact('hello','you'), 'true', 'false') // = "false"
if (or(1=1,exact('hello','you')), 'true', 'false') // = "true"
if (and(1=1,exact('hello','you')), 'true', 'false') // = "false"
```

Date Functions

Calculated Attribute expressions can be written using the following available functions. Examples follow the Available Functions table.

Available Functions

Date Function	Parameters	Description
FORMATDATETIME	(date, format-string[, locale])	Determine the output by formatting a date according to the format-string, see http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html [locale] is optional
LOCALIZEDATE	(date[, locale])	Localize the date to a string using the locale in the current / extracting / resolving context, or using the supplied locale. [locale] is optional
NOW	()	Returns current time on server in yyyy-MM-dd HH:mm:ss format.

Examples

AttrDateID on Product	Value
P1id	2010-10-15
P2id	1997-12-25
P3id	2016-05-03

Copy and paste any of the following examples into the Function Editor and then evaluate against the specified object to get the defined result.

```
formatdatetime(prodval('AttrDateid'), "EEE, MMM d, ''yy") // = "Fri, Oct 15, '10" for P1id
formatdatetime(prodval('AttrDateid'), "yyyyy.MMMMM.dd" ) // = "01997.December.25" for P2id
formatdatetime(prodval('AttrDateid'), "MMM d, yy" ) // = "May 3, 16" for P3id
localizedate(prodval('AttrDateid')) // = "03-May-2016" for P3id when in Danish context
now() // = "2016-05-04 12:06:25" when executed on 4 MAY 2016 just after midday
```

List Functions

Calculated Attribute expressions can be written using the following available functions. Examples follow the Available Functions table.

Available Functions

Function	Parameters	Description
LISTCONCATENATE	(list1, list2, ...)	Returns a list with the elements of all the supplied lists. The order of the elements in the resulting list is: the elements of list1, elements of list2, etc.
LIST	(list, separator[, last-separator])	Returns text with all the elements of 'list'. The elements are separated by the text 'separator'. [last-separator] is optional and controls the last separator.
ITERATE	(list, expression)	Returns a list where the elements of 'list' are applied to 'expression'. Iterate allows data to be pulled from an object other than the current object, even if the list created is only a single object. For more information, see **Use Case: Calculated Attribute for References** of the **Calculated Attributes** documentation. The 'index', 'item', and 'size' variables are available. For more information, see the **Other Functions** section of the **Calculated Attributes** documentation.
FILTER	(list, expression)	Returns a list with the elements of 'list' where 'expression' evaluates to true. The 'index', 'item', and 'size' variables are available. For more information, see the **Other Functions** section of the **Calculated Attributes** documentation.
REVERSE	(list)	Returns a list with the elements of 'list' in reverse order.
SORT	(list)	Returns 'list' sorted. **Note:** No locale sorting is supported. All sorting is done in LATIN1 character set, which means that Unicode characters are not sorted correctly.
SORTNUMERIC	(list)	Returns 'list' sorted numerically.
UNIQUE	(list)	Returns list where duplicates in 'list' are removed.
RANGE	(list, separator [, separator-between-ranges[, sorted-valued-list]])	Returns text representing the ranges in 'list'. [separator-between-ranges] and [sorted-valued-list] are optional.

Function	Parameters	Description
LIST2MULTIVALUE	(list)	Returns multi-valued text, where the values equal the elements in the 'list'.
MULTIVALUE2LIST	(text)	Returns a list where the elements equal the values in the multi-valued 'text'.
LISTLEN	(list)	Returns the number of elements in 'list'.
LISTITEM	(list, indexno)	Returns the item identified by 'indexno' in 'list'. The index number of the first element is 1.
LISTCONTAINS	(list, text)	Returns TRUE (=1) if 'list' contains the element 'text'; otherwise returns FALSE (=0).

Examples

The examples below assume a setup including these **attributes** and **products**:

ATTRIBUTE Type	ID	Name	AttrDesc
Text-validated product attribute	TEXT1id	TEXT1	TEXT1 description
Number-validated product attribute with length units	NUMBER1id	NUMBER1	NUMBER1 description
Embedded number-validated product attribute with length units	EMBEDDEDNUMBER1id	EMBEDDEDNUMBER1	EMBEDDEDNUMBER1 description
Text-validated attribute property	AttrDescid	AttrDesc1	AttrDesc description
LOV-validated product attribute	LOVAttr1id	LOVAttr1	LOV description
LOV with 'Use IDs on Values' = Yes; and Values =Yes and No	LOV1id	LOV1	

PRODUCTS ID	Name	Parent	TEXT1id	NUMBER1id	EMBEDDEDNUMBER1id	LOVAttr1id
P-family-id	P-family	Product hierarchy root	family value	2 m		
P1id	P1	P-family-id	p1 value	4 m		
P2id	P2	P-family-id	p2 value	5 m		No
P3id	P3	P-family-id	p3 value	6 m	prefix 3 m suffix	Yes

Copy and paste any of the following examples into the Function Editor and then evaluate against the specified object to get the defined result.

```
listconcatenate(iterate(subproducts(), 'prodval("TEXT1id")'), iterate(subproducts(), 'prodval("NUMBER1id")')) // = "p1 valuep2 valuep3 value456" evaluated on P-family-id

list(iterate(subproducts(), 'prodval("TEXT1id")'), ', ') // = "p1 value, p2 value, p3 value" evaluated on P-family-id

list(iterate(subproducts(), 'prodval("TEXT1id")'), ', ', ' & ') // = "p1 value, p2 value & p3 value" evaluated on P-family-id

iterate(filter(subproducts(),'prodval("number1id")>3'),'concatenate("Name = ",stepname(),"; ID = ", stepid(), ". ")') // = extracts id of all child products where number1id is greater than 3: Name = P1; ID = P1id. Name = P2; ID = P2id. Name = P3; ID = P3id.

reverse(iterate(subproducts(), 'prodval("TEXT1id")')) // = "p3 valuep2 valuep1 value" evaluated on P-family-id

sort(iterate(subproducts(), 'prodval("NUMBER1id")')) // = "456" evaluated on P-family-id

list(sortnumeric(listconcatenate('2', '56', '11', '0', '21', '13')), ',') // = "0,2,11,13,21,56"

unique(listconcatenate('a','b','a','c','b')) // = "a, b, c"

range(listconcatenate('1','5','8','6','7'), '-') // = "1-8"

range(listconcatenate('1','5','8','6','7'), '-', ', ') // = "1, 5-8"

list(list2multivalue(value('AttributeA'))) // = "value a<multisep/>value b<multisep/>value c" evaluated on multivalue attribute with the values: "value a", "value b", and "value c"

list(multivalue2list(value('AttributeA')),', ') // = "value a, value b, value c" evaluated on multivalue attribute with the values: "value a", "value b", and "value c"

listlen(iterate(subproducts(), 'prodval("TEXT1id")')) // = "3" evaluated on P-family-id

listitem(iterate(subproducts(), 'prodval("TEXT1id")'),2) // = "p2 value" evaluated on P-family-id

{x := iterate(subproducts(), 'prodval("TEXT1id")')} if(listcontains(x,"p1 value"), "p1 is present","p1 is NOT present") // = "p1 is present" evaluated on P-family-id; this is the same formula as shown following except the variable x is used to separate the list formula

if(listcontains(iterate(subproducts(), 'prodval("TEXT1id")'),"p1 value"), "p1 is present","p1 is NOT present") // = "p1 is NOT present" evaluated on a family other than P-family-id; this is the same formula as shown previously except the list formula is written in full within the LISTCONTAINS function
```

Number Functions

Calculated Attribute expressions can be written using the following available functions. Examples follow the Available Functions table.

Available Functions

Number Function	Parameters	Description
TEXT	(number, format string)	Returns a textual representation of the supplied number, formatted by the supplied formatting string.
TRUNC	(number)	Returns the integer part of 'number' by removing the decimal and any numbers following it. No rounding is performed.
INT	(number)	Returns the integer part of 'number' by rounding down to the nearest integer. Negative numbers are rounded down.
ROUND	(number, precision)	Returns 'number', rounded to a specified number of digits. • If 'precision' is positive (1), then number is rounded to the specified number of decimal places. • If 'precision' is zero (0), then number is rounded to nearest integer. • If 'precision' is negative (-1), then number is rounded to the left of the decimal point.
LOCALIZENUMBER	(number[, locale])	Localizes the number to a string, using the locale in the current / extracting / resolving context, or using the supplied locale. [locale] is optional
Binary number functions	+, -, *, /	Returns the computed number of the operation. If possible, text is converted to a number. If number conversion fails, 'NaN' (meaning 'Not a Number') is returned. **Note:** Rounding errors can occur when using the binary number functions in number calculations. To correct rounding in these cases, use the ROUND or TEXT function.
Binary logical functions	<, >, <=, >=, =,!=	Returns TRUE (=1) or FALSE (=0). **Note:** These functions do NOT work on text strings and always attempt to convert to a number.

Examples

Copy and paste any of the following examples into the Function Editor and then evaluate against the specified object to get the defined result.

```
text(2.45, '000.0') // = "002.5"
```

```
text(2.45, '.0') // = "2.5"
text(2000.45, ',.0') // = "2,000.5"
trunc(5.64) // = 5
int(5.64) // = 5
int(-5.64) // = -6
trunc(-5.64) // = -5
round(5.64, 0) // = 6
round(5.64, 1) // = 5.6
round(5.64, -1) // = 10
1.25*11 // = 13.75
```

Other Functions

Calculated Attribute expressions can be written using the following available functions. Examples follow the Available Functions table.

Note: Adding a new line by pressing the Enter key or adding "\n" within the Function Editor template allows the resulting values to display in multiple lines.

For more extended scenarios and function templates, see the available **Calculated Attribute Use Case Examples** in the **Calculated Attributes** section of the **System Setup / STEP Super User Guide** documentation.

Available Functions

Other Function	Parameters	Description
OR	(condition1, condition2, ...)	Returns TRUE (=1) if one of the supplied conditions is true; otherwise returns FALSE (=0).
AND	(condition1, condition2, ...)	Returns TRUE (=1), if all the supplied conditions are true; otherwise returns FALSE (=0).
NOT	(condition)	Returns TRUE (=1) if condition is false; otherwise returns FALSE (=0).
STEPID	()	Returns the ID of the current object.
STEPNAME	()	Returns the name of the current object.
STEPNAME	('url')	Returns the name of the object pointed to by 'url'.
STEPURL	()	Returns the STEP URL of the current object.
STEPURL	('object-type', 'object-id')	Returns the STEP URL of the object with ID 'object-id' and of type 'object-type'. The object type could be product, classification, entity, asset, attribute, etc.
STEPURL2OBJID	('url')	Returns the internal object representation of the object pointed to by 'url'. **Note:** This function must be used in combination with a LIST or ITERATE function.
STEPOBJECTTYPE	()	Returns the STEP name of the object type of the current object.
STEPOBJECTTYPE	('url')	Returns the STEP name of the object type of the object pointed to by 'url'.

Other Function	Parameters	Description
CHILDREN	()	Returns true children (not links) of an asset, a classification, an entity, a product, or a product-override. Similar to SUBPRODUCTS, but this function returns more than products. **Note:** This function must be used in combination with a LIST or ITERATE function.
SUBPRODUCTS	()	Returns sub products of a product, product-override, or classification. Similar to CHILDREN, but this function returns only products. **Note:** This function must be used in combination with a LIST or ITERATE function.
CLASSIFICATIONS	([link-type])	Returns the names and/or IDs of the classifications of an asset, a product, or a product-override (classifications of product-overrides + classifications of overridden product) of the specified 'link-type'. [link-type] is optional.
CLASSIFICATIONPRODUCTLINKS	([link-type])	Returns the metadata values placed on the links between an object and classification of the specified 'link-type'. [link-type] is optional, all are returned if not specified.
CPLCLASSIFICATION	()	Returns the classification object from a classification-product link.
CPLPRODUCT	()	Returns the product object from a classification-product link.
REFERENCES	([object-type, [reference-type]])	Returns the references to objects of type 'object-type' and with the reference type 'reference-type'. Object-type could be product, classification, entity, or asset. [object-type] and [reference-type] are optional, all are returned if not specified or blank.
REFERENCESTO	([object-type[, reference-type]])	Returns the references pointing to this object from objects of type 'object-type' and with specified reference type. Object-type could be product, classification, entity, or asset. [object-type] and [reference-type] are optional, all are returned if not specified or blank.
REFERENCETARGET	()	Returns the target (destination) object of reference.
REFERENCEOWNER	()	Returns the owner of the reference.
REFERENCETYPEID	()	Returns the ID of the reference type.
PARENTID	()	Returns the parent ID if node is a tree node.
PATH	()	Returns the parent path of a product, entity, or a classification. The path shows the

Other Function	Parameters	Description
		top node in the tree as the first element and the current object is the last element.
ISBELOW	(url)	Returns TRUE (=1) if the object pointed to by 'url' is in the parent path of the current object.
INDEX	()	This variable holds the current index number of the iteration and is only available within the FILTER or ITERATE functions.
SIZE	()	This variable holds the size of the iterated list and is only available within the FILTER or ITERATE functions. **Note:** The LISTLEN function provides the same results and is easier to use.
ITEM	()	This variable holds the current item in the iteration and is only available within FILTER or ITERATE functions.
REVISIONS	()	Returns the list of all revision objects for the current object. The revisions are displayed with the first revision object first and the most recent revision object last. **Note:** This function must be used in combination with a LIST or ITERATE function.
REVISIONNAME	()	Returns the name of the current revision (or most recent revision of current object). The revision number is shown as '0.1', '1.0', etc.
REVISIONEDITEDBY	()	Returns the name of the user responsible for the change in the current revision (or most recent revision of current object).
REVISIONCOMMENT	()	Returns the comment of the current revision (or the most recent revision of current object).
REVISIONISMAJOR	()	Returns the Boolean TRUE (=1) or FALSE (=0) to indicate if the current revision (or most recent revision of current object) is a major revision or not.
REVISIONCREATEDATE	()	Returns the creation date of the current revision (or most recent revision of current object). The date is returned in ISO format "YYYY-MM-DD HH:MM:SS".
REVISIONEDITDATE	()	Returns the last edit date of the current revision (or most recent revision of current object). The date is returned in ISO format "YYYY-MM-DD HH:MM:SS".
FIRSTREVISIONCREATEDATE	()	Returns the creation date of the first revision (not the initial revision which could have been purged). The date is returned in ISO format "YYYY-MM-DD HH:MM:SS".
OVERRIDDENPRODUCT	(product-override-id)	Returns the ID of the overridden product for the product override with supplied ID. Returns N/A if the object is not an overridden product.

Other Function	Parameters	Description
OVERRIDDENPRODUCTOBJECT	()	Returns the product being overridden by the current product override. Returns an empty value if the object has no overridden product.
OVERRIDDENBY	()	Returns the product overrides for the current product. Returns an empty list if the object is not overridden.
PRODUCTOVERRIDEPARENTS	()	Returns the product overrides having this product as child. Returns an empty list if the object is not linked into any product overrides.
REPLACEVALUEBYLOOKUP	(asset-id, value)	Returns the 'value' substituted by using the transformation lookup table referred to in 'asset-id'. For more information, see the **Transformation Lookup Tables** section of the **Data Quality** documentation.
REPLACEWORDBYLOOKUP	(asset-id, words)	Returns what each word in 'words' is substituted by, using the transformation lookup table referred to in 'asset-id'. For more information, see the **Transformation Lookup Tables** section of the **Data Quality** documentation.
COMPLETENESS	([completeness-metric-id])	Returns the completeness score of the current object as a percentage (decimal number between 0 and 100). If completeness does not make sense for the current object, an empty string is returned. [completeness-metric-id] is optional. The default completeness metric is used if none is supplied. For more information, see **Setting Up Completeness Metrics** in the **Data Quality** documentation.
PACKAGINGHIERARCHYCHILDLINKS	()	Returns the packaging hierarchy child links of the current product.
PACKAGINGHIERARCHYPARENTLINKS	()	Returns the packaging hierarchy parent links of the current product.
PHLCHILD	()	Returns the child product of the current packaging hierarchy link.
PHLPARENT	()	Returns the parent product of the current packaging hierarchy link.
PHLQUANTITY	()	Returns the quantity of the current packaging hierarchy link.
WORKFLOWIDS	()	Return the list of workflow IDs for the STEP workflows that include the object.
CURRENTWORKFLOWSTATES	(workflow-id)	Returns the list of IDs for the STEP workflow states that include the current object for the supplied 'workflow-id'.

Examples

The examples below assume a setup including these **attributes** and **products**:

ATTRIBUTE Type	ID	Name	AttrDesc
Text-validated product attribute	TEXT1id	TEXT1	TEXT1 description
Number-validated product attribute with length units	NUMBER1id	NUMBER1	NUMBER1 description
Embedded number-validated product attribute with length units	EMBEDDEDNUMBER1id	EMBEDDEDNUMBER1	EMBEDDEDNUMBER1 description
Text-validated attribute property	AttrDescid	AttrDesc1	AttrDesc description
LOV-validated product attribute	LOVAttr1id	LOVAttr1	LOV description
LOV with 'Use IDs on Values' = Yes; and Values =Yes and No	LOV1id	LOV1	

PRODUCTS ID	Name	Parent	TEXT1id	NUMBER1id	EMBEDDEDNUMBER1id	LOVAttr1id
P-family-id	P-family	Product hierarchy root	family value	2 m		
P1id	P1	P-family-id	p1 value	4 m		
P2id	P2	P-family-id	p2 value	5 m		No
P3id	P3	P-family-id	p3 value	6 m	prefix 3 m suffix	Yes

Copy and paste any of the following examples into the Function Editor and then evaluate against the specified object to get the defined result.

```
or((value('TEXT1id')= 'p1 value'), (value('NUMBER1id')= '4')) // = 1 (TRUE) evaluated P1id
or((value('TEXT1id')= 'p0 value'), (value('NUMBER1id')= '3')) // = 0 (FALSE) evaluated P1id
and((value('TEXT1id')= 'p1 value'), (value('NUMBER1id')= '4')) // = 1 (TRUE) evaluated P1id
and((value('TEXT1id')= 'p0 value'), (value('NUMBER1id')= '3')) // = 0 (FALSE) evaluated P1i
d
not(value('TEXT1id')= 'p1 value')) // = 0 (FALSE) evaluated P1id
not(value('TEXT1id')= 'p0 value')) // = 1 (TRUE) evaluated P1id
stepid() // = 'P1id' evaluated P1id
stepname() // = 'P1' evaluated P1id
stepname(step://product?id=P1id) // = 'P1' evaluated P1id
```

```
stepurl() // = 'step://product?id=P1id' evaluated P1id

stepurl('Item','P1id') // = 'step://product?id=P1id' evaluated P1id

stepobjecttype() // = 'Item' evaluated P1id

stepobjecttype('step://product?id=P1id') // = 'Item' evaluated P1id

list(iterate(children(),'concatenate(stepname())'),', ') // = 'P1, P2, P3' evaluated P-Fami
ly since these are products

list(iterate(subproducts(),'concatenate(stepname())'),', ') // = 'P1, P2, P3' evaluated P-F
amily since these are products

list(iterate(children(),'concatenate(stepname())'),', ') // = 'E1, E2, E3' evaluated entity
with three children, none of which are products

list(iterate(subproducts(),'concatenate(stepname())'),', ') // = '' evaluated entity with t
hree children, none of which are products

list(iterate(iterate(references('product', 'BillOfMaterials'),'referencetarget()'),'stepnam
e()'),', ') // = 'P3' evaluated against P1 when P1 has a BillOfMaterials reference to P3

parentid() // = 'P-family' evaluated P1id

list(iterate(path(),'concatenate(stepname())'),', ') // = 'Primary Product Hierarchy, Produ
cts, P-family' evaluated P-Family

isbelow('step://product?id=P1id') // = 0 (FALSE) evaluated P-Family

isbelow('step://product?id=P-Family') // = 1 (TRUE) evaluated P1

list(iterate(listconcatenate('a', 'b', 'c'), 'concatenate("INDEX: ", index, " ITEM: ",
item)'),', ') // = 'INDEX: 1 ITEM: a, INDEX: 2 ITEM: b, INDEX: 3 ITEM: c' evaluated P-Famil
y

list(iterate(revisions(), 'revisioncreatedate()'), ', ') // = '2016-04-20 17:18:42, 2016-0
4-22 09:44:36, 2016-04-25 09:34:36, 2016-05-06 11:04:09' evaluated P-Family

revisionname() // = '0.3' evaluated P-Family

revisioneditedby() // = 'USER' evaluated P-Family

revisioncomment() // = 'Auto Generated' evaluated P-Family

revisionismajor() // = '0' (FALSE) evaluated P-Family

revisioncreatedate() // = '2016-04-25 09:34:36' evaluated P-Family

revisioneditdate() // = '2016-04-25 16:16:30' evaluated P1

firstrevisioncreatedate() // = '2016-04-20 17:20:41' evaluated P1

completeness() // = '51.28205128205128460194828221574 4' evaluated P1

round(completeness(),0) // = '51' evaluated P1

workflowids() // = 'ItemCreationOrderProcessWorkflow' evaluated P1 (2 workflows returned: I
tem Creation and Order Process Workflow)

currentworkflowstates('OrderProcessWorkflow') // = 'State-2'  evaluated P1

list(iterate(classifications(), "stepid()"), "/ ") // = Classification1/ Classification2 ev
aluated product that exists in two classifications
```

- The following template returns the value of the metadata on the attribute 'TEXT1id' as well as the value of the attribute on the evaluated object

```
{meta := iterate(stepurl2objid(stepurl('attribute','TEXT1id')), 'value("Attribute Descripti
on")'),val := value('TEXT1id')}
```

```
concatenate("\'", meta,"\'", " is the metadata on the \'TEXT1id\' attribute which contains
the value: ", val)

/* "'TEXT1 description' is the metadata on the 'TEXT1id' attribute which  contains the valu
e: family value" evaluated P-family*/
```

- The following templates use the product folder 'Batteries' (child of 'Battery Items') which is overridden by 'Batteries Override 1', 'Batteries Override 2', and 'Batteries Override 3'.

Template	Results
overriddenproduct('BatteriesOverride1ID')	**Evaluated on Batteries Override 1:** Batteries
overriddenproduct('Batteries')	**Evaluated on Batteries Override 1:** N/A

{
x := listlen(**overriddenby**()),
z := iterate(overriddenby(), 'stepname()'),
z := list2multivalue(z),
z := substitute(z, "<multisep/>", "
")
}
concatenate("This product is overridden by ", x,
", product override(s):
", z)
	Evaluated on Batteries: This product is overridden by 3 product override(s):
	Batteries Override 2
	Batteries Override 1
	Batteries Override 3

{
x := iterate(**overriddenproductobject**(), 'stepname
()'),
z := listitem(x,1)
}
concatenate('This PO overrides the product "', z,
'"')
	Evaluated on Batteries Override 1: This PO overrides the product "Batteries"

list(iterate(**productoverrideparents**(), "stepid()"), "/ ")	**Evaluated on Battery Items:** BatteriesOverride2ID/BatteriesOverride1ID/BatteriesOverride3ID

- The following templates use this lookup table data.

Asset: UOMLookupTable			
From	EACH	DOZEN	CASE
To	Ea.	Dz.	Case

```
REPLACEVALUEBYLOOKUP('UOMLookupTable','EACH') // = 'Ea.' evaluated P1

REPLACEWORDBYLOOKUP('UOMLookupTable','EACH, DOZEN, CASE') // ='Ea., Dz., Case' evaluated P1
```

- The following template returns name and ID of parent product. Use -2 (instead of -1) to get name / ID of grandparent, etc.

```
{i:=path(), parent:=listitem(i,listlen(i)-1)} iterate(parent, 'concatenate("name=",stepname
(),"; id=", stepid())') // = 'name=P-family; id=P-family-id' evaluated P1
```

- The following template returns the value for attribute ID 'BrandName' that occurs the most times in its children.

```
{
all:=iterate(subproducts(), "prodval('BrandName')"),
unique:=unique(all),
oftenIndex:=1,
oftenCount:=0,
dummy:=iterate(unique, '
{
curval:=item,
count:=listlen(filter(all,"exact(curval,item)"))
}
if (count>oftenCount, concatenate(
oftenCount:=count,
oftenIndex:=index),
0)')
}
if (oftenCount>0, listitem(unique,oftenIndex), "")
```

- The following template extracts values for attribute 'grandchildattrid' of all grandchildren into a list.

Note: Use cautiously. This operation can drain system CPU since all grandchildren are read for each product.

```
{attrid:='grandchildattrid'}listconcatenate(iterate(subproducts(), 'iterate(subproducts(),"
prodval(attrid)")'))
```

- The following template compares the value of 'RemovalDate' attribute on the child objects and returns the one that is closest to the current date while still less than the current date. Returns 'None Newer' if no such child exists.

```
{
msg := "Closest Date That Is Less Than The Current Date Is: ",
datenow := left(substitute(now(), "-", ""), 8),
datelist := filter(iterate(children(), 'value("ISODate")'), 'substitute(item, "-", "") <l
t/> datenow'),
neardate := listitem(reverse(sortnumeric(datelist)), 1)
}
if(exact(neardate, "N/A"), "None Newer", concatenate(msg, neardate))
```

Text Functions

Calculated Attribute expressions can be written using the following available functions. Examples follow the Available Functions table.

- Use '\n' for producing a new line in a string constant.
- All constant text is escaped using the escape character '\'. When writing regular expressions, to get a single '\' from a string constant, use '\\'. This is only true for text constants written in a formula—not when taken from an attribute value.
- Constant strings within each other (for example, iterate(list, "prodval('single-back-slash: \\\\')")) require extra levels of escaping—one level of escaping per constant text.

Available Functions

Text Function	Parameters	Description
EXACT	(text1, text2)	Returns TRUE (=1) if 'text1' equals 'text2', otherwise return FALSE (=0).
CONCATENATE	(text1, text2, ...)	Returns the concatenated text of all the supplied texts in the parameter string.
LEFT	(text [, len]):	Returns the first 'len' number of characters of the text. [len] is optional, the default is 1.
RIGHT	(text[, len])	Returns the last 'len' number of characters of the text. [len] is optional, default is 1.
MID	(text, pos, len)	Returns the substring of 'text' starting at character position 'pos' with length 'len'.
REPLACE	(text, pos, len, newtext)	Returns 'text' where the substring starting at character position 'pos' with length 'len' has been replaced with the text 'newtext'.
SUBSTITUTE	(text, from, to [, indexno])	Returns 'text' where all instances of text 'from' are replaced with text 'to'. [indexno] is optional. If supplied, only that one instance is replaced.
FIND	(searchtext, text [, pos])	Returns first character position in 'text' where 'searchtext' is found—search will begin at character position 'pos' (default is 1). Search is case sensitive.

Text Function	Parameters	Description
		[pos] is optional. If not found, 'N/A' is returned.
REGEXPSEARCH	(searchtext, text [, pos])	Returns first character position in 'text', matching the regular expression 'searchtext'. Search will begin at character position 'pos'. Search is case sensitive. [pos] is optional, the default is 1. If not found, 'N/A' is returned. **Note**: \ is used to escape characters in a text constant (see bullets above). For example, to get a regexp string like '\s', use double escaping of the backslash character, like '\\s'.
REGEXPSEARCHI	(searchtext, text [, pos])	Same as function REGEXPSEARCH, but search is case insensitive. [pos] is optional, if not found 'N/A' is returned. **Note:** \ is used to escape characters in a text constant (see bullets above). For example, to get a regexp string like '\s', use double escaping of backslash character like '\\s'.
REGEXPSUBSTITUTE	(text, from, to)	Return 'text' where all instances of text 'from' is replaced with text 'to'. Regular expressions are allowed as defined at http://oracle.com/javase/. Search for 'Pattern (Java Platform SE 7)'. **Note:** \ is used to escape characters in a text constant (see bullets above). For example, to get a regexp string like '\s', use double escaping of backslash character like '\\s'.
LEN	(text)	Return number of characters in 'text'.
TRIM	(text)	Return 'text' where leading and trailing spaces are removed.
UPPER	(text)	Return 'text' where all characters are in upper case.
LOWER	(text)	Return 'text' where all characters are in lower case.
PROPER	(text)	Return 'text' where the first letter of all words in the text string (and the first letter that follows any character other than a letter) is capitalized. Converts all other letters to lowercase letters.
FIRSTNAMEFIRST	(text)	Function for reorganizing names formatted with family name first followed by a comma and then first name.
SOUNDEX	(text)	Return the four-character Soundex encoding of 'text'. The Soundex code for a name consists of a letter followed by three numerical digits: the letter is the first letter of the name, and the digits encode the remaining consonants. Similar-sounding consonants share the same digit. For example, the consonants B, F, P, and V are each encoded as the number 1. Vowels can affect the coding, but are not coded themselves except as the first letter. However, if 'h' or 'w' separate two consonants that have the same Soundex code, the consonant to the right of the vowel is not coded. The correct value can be found as follows: The first letter of the name is the letter of the Soundex code, and is not coded to a number.

Text Function	Parameters	Description
		Replace consonants with digits as follows (after the first letter): b, f, p, v > 1 c, g, j, k, q, s, x, z > 2 d, t > 3 l > 4 m, n > 5 r > 6 h, w are not coded Two adjacent letters with the same number are coded as a single number. Letters with the same number, separated by an 'h' or 'w', are also coded as a single number. Continue until you have one letter and three numbers. If you run out of letters, fill in 0s until there are three numbers.
METAPHONE3	(text)	The function returning the primary Metaphone 3 code. Metaphone 3 has two 'sounds like' strings for a given input string—a 'primary' and an 'alternate'. In most cases they are the same, but for non-English names they can be a bit different, depending on pronunciation. Metaphone 3, like Soundex, is based on the idea of constructing a representative code for an input string. Metaphone 3 is a good choice for person names. Metaphone 3 tries to give name variations or names that sounds like the same code.
METAPHONE3ALTERNATE	(text)	The function returning the alternate Metaphone 3 code.
ESCAPEMERGE	(split-character, escape-character, text1, ...)	Used in text manipulation. Merges text1...textN into a single string. The 'split-character' is used as a separator between the elements. If the 'split-character' is found in the text elements, then the 'escape-character' precedes it. Opposite function of ESACPESPLIT.
ESCAPESPLIT	(split-character, escape-character, text)	Used in text manipulation. Takes a single string and splits it into a list of elements, using the 'split-character' in the string, and removing the 'escape-character' after splitting the elements. To remove duplicates in a string, use ESCAPESPLIT to split a single text to multiple elements. Then use the UNIQUE function to remove the duplicates. Afterward, use LISTCONTATENATE to return the list elements to single string. Opposite function of ESCAPEMERGE.

Examples

Copy and paste any of the following examples into the Function Editor and then evaluate to get the defined result.

```
exact("hello", "hallo") //= 0
exact("hello", "hello") //= 1
```

```
concatenate("hello", "hallo") //= hellohallo
left('hello') //= "h"
left('hello',3) //= "hel"
right('hello') //= "o"
right('hello', 3) //= "llo"
mid('hello',2,3) //= "ell"
replace('hello',2,3,'el') //= "helo"
substitute('hellohellohello', 'ell', 'al') //= "halohalohalo"
substitute('hellohellohello', 'ell', 'al', 2) //= "hellohalohello"
find('llo','hellohello') //= 3
find('llo','hellohello', 4) //= 8
find('lllo','hellohello') //= "N/A"
regexpsearch('e.*he', 'hellohello') //= 2
regexpsearch('[ABC]\\d', 'HELLOHA2MSTER') //= 7
regexpsearchi('[efg]|[ojk]', 'HELLOHELLO') //= 2
regexpsearchi('[abc]|[ojk]', 'HELLOHELLO', 3) //= 5
regexpsubstitute("kat;abe", "(.*);(.*)", "$2$1") //= "abekat"
regexpsubstitute("kat  abe", "(\\w*)\\s*(\\w*)", "$2$1") //=  "abekat"
len('hello') //= 5
trim('   hello   ') //= 'hello'
upper('hello') //= 'HELLO'
lower('HELLO') //= 'hello'
proper("hello, are you john's son?") //= "Hello, Are You John'S Son?"
firstnamefirst("Johnson, John S.") //= "John S. Johnson"
firstnamefirst("Joe Dalton") //= "Joel Dalton"
soundex("hello") //= H400
metaphone3("Joel Dalton") //= JLTLTN
metaphone3alternate("Joel Dalton") // = ALTLTN
escapemerge(';', '/', "Red", "Orange; Blue", "Violet") //= Red;Orange/; Blue;Violet
escapesplit(';', '/', 'Red;Orange/; Blue;Violet') // = RedOrange; BlueViolet
```

Variables

A variable declaration can be added before a function call as a comma-separated list in curly brackets {}. The scope of the variables declared are only available inside the function parameters that follow.

By default, the three variables 'index', 'item', and 'size' are available within all ITERATE and FILTER function expressions. For more information, see the **List Functions** section of the **Calculated Attribute** documentation.

Variables can be reassigned to another value by normal expression 'var:=expr'. This expression evaluates to the value of 'expr' by assigning the value to variable 'var'.

Examples

The examples below assume a setup including these **attributes** and **products**:

ATTRIBUTE Type	ID	Name	AttrDesc
Text-validated product attribute	TEXT1id	TEXT1	TEXT1 description
Number-validated product attribute with length units	NUMBER1id	NUMBER1	NUMBER1 description
Embedded number-validated product attribute with length units	EMBEDDEDNUMBER1id	EMBEDDEDNUMBER1	EMBEDDEDNUMBER1 description
Text-validated attribute property	AttrDescid	AttrDesc1	AttrDesc description
LOV-validated product attribute	LOVAttr1id	LOVAttr1	LOV description
LOV with 'Use IDs on Values' = Yes; and Values =Yes and No	LOV1id	LOV1	

PRODUCTS ID	Name	Parent	TEXT1id	NUMBER1id	EMBEDDEDNUMBER1id	LOVAttr1id
P-family-id	P-family	Product hierarchy root	family value	2 m		
P1id	P1	P-family-id	p1 value	4 m		
P2id	P2	P-family-id	p2 value	5 m		No
P3id	P3	P-family-id	p3 value	6 m	prefix 3 m suffix	Yes

Copy and paste any of the following examples into the Function Editor and then evaluate against the specified object to get the defined result.

```
{txt:=prodval('TEXT1id'), pos:=find('p3', txt)} if(pos='N/A', 'no P3', replace(txt, pos, 2,
'??')) // = "?? value" evaluated P3; "no P3" evaluated P2
```

```
{total:=0, loop:=iterate(listconcatenate(1,4,8,13), 'total:=total+item')} concatenate
(total) // = 26
```

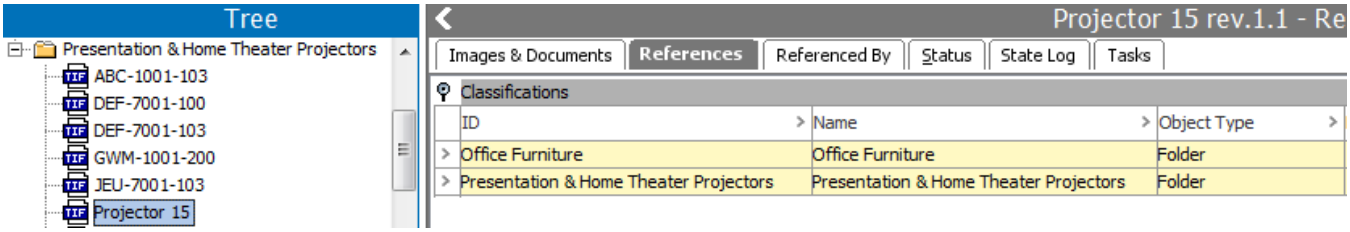
Calculated Attribute Use Case Examples

Scenarios and formula templates follow for these use cases:

- Calculated Attributes for Classifications
- Calculated Attribute for Overrides
- Calculated Attributes for References

Use Case: Calculated Attribute for Classifications

An asset is linked to two classifications.

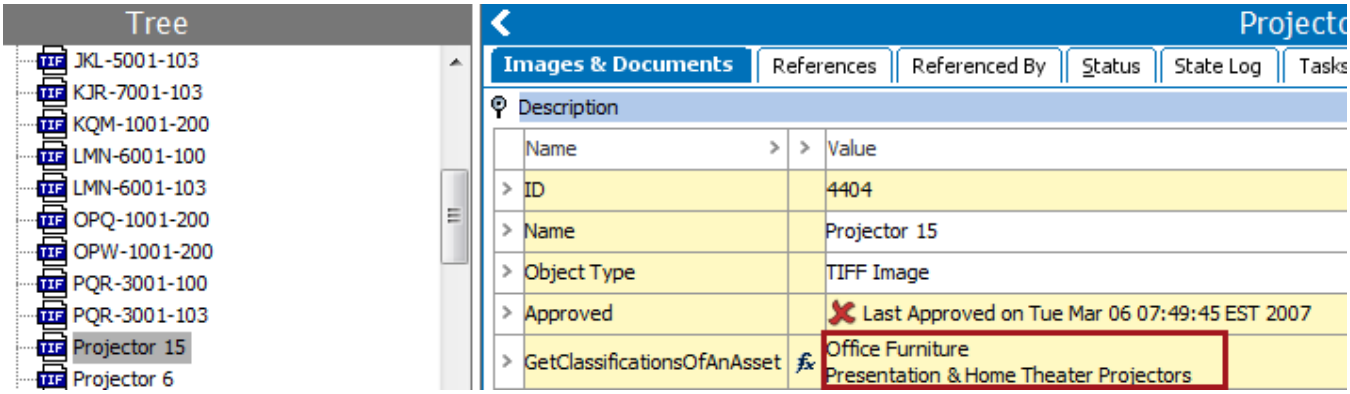


Get All Classifications

A metadata calculated attribute on the asset will display the name and/or ID for all classifications. The calculated attribute formula could be written as:

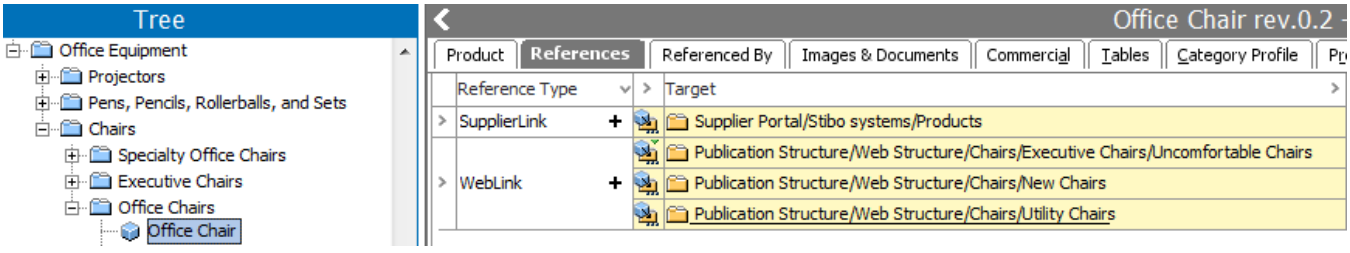
```
list(iterate(classifications(), 'stepname()'), "
")
```

The new line (added by pressing Enter or using "\n") within the formula allows the attribute value to display on multiple lines, as shown in the following image.



Get Classifications Based on Type

Multiple types of classifications exist.

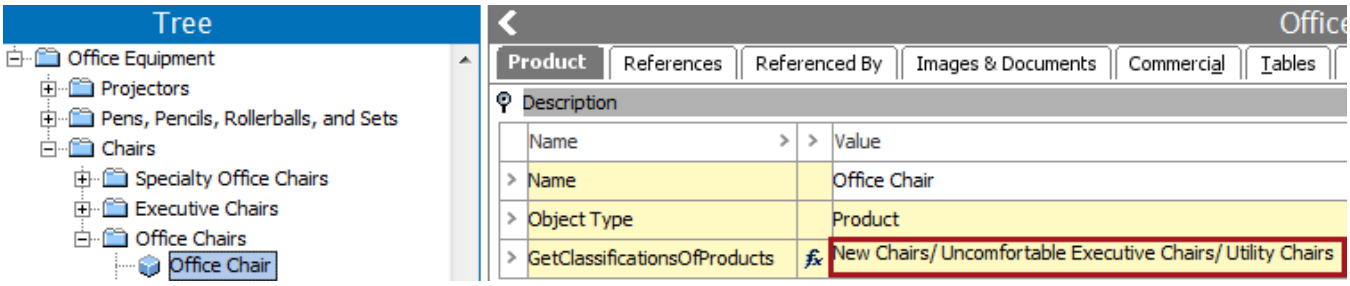


A metadata calculated attribute on the asset could display a limited number of classifications. For example, only include the 'WebLink' references type, which have an ID of 'WebInclude'.

The calculated attribute formula could be written as:

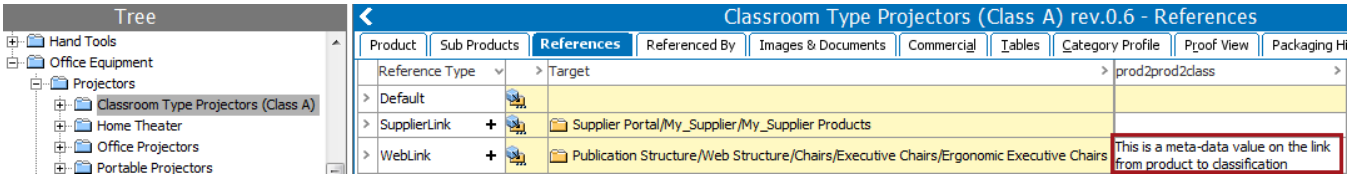
```
list(iterate(classifications("WebInclude"), "stepid()"), "/ ")
```

The result displays the references separated by the text string "/ ".



Get Metadata Value Located on the Link From Classification-to-Product

The product is linked to a classification folder with certain reference type, and includes metadata on that link. For example, Classroom Type Projectors has a WebLink reference (ID = 'WebInclude'), that has the attribute prod2prod2class.

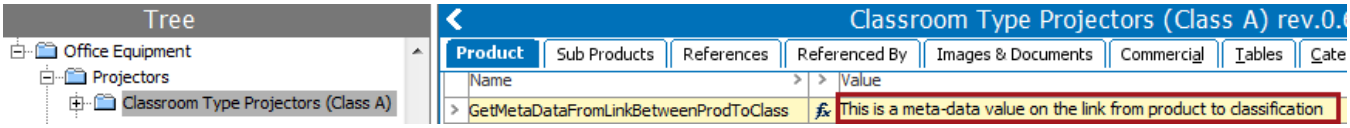


The calculated attribute formula could be written as:

```
{
j := iterate(classificationproductlinks("WebInclude"),'value("prod2prod2class")'),
z := listitem(j,1)
}
concatenate(z)
```

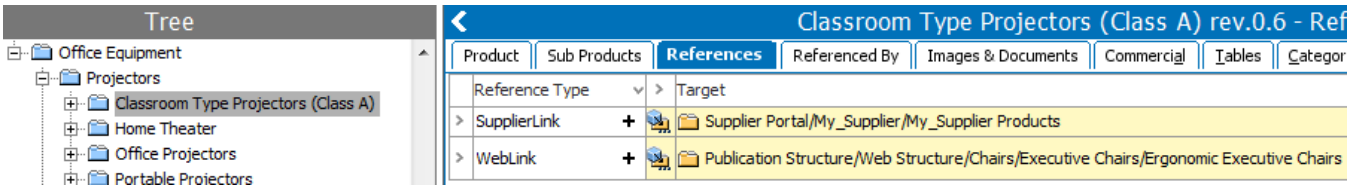
Since only one link is assumed, the formula uses LISTITEM(). If more links use the same reference type to other classifications, additional filtering would be required.

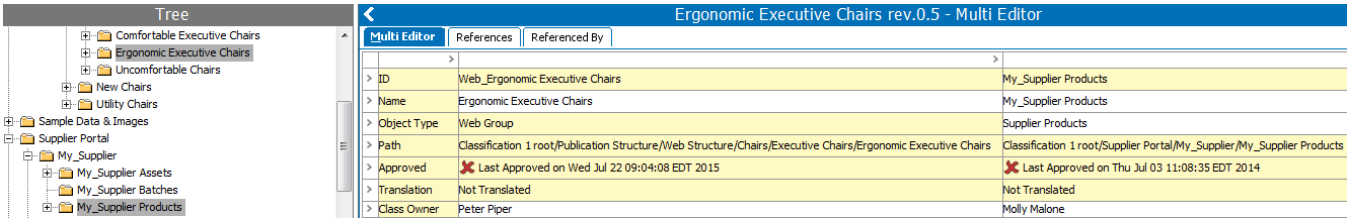
The result displays the value of the linked metadata on the original product.



Get CPL Classifications Using Classification Product Links

A product is linked into one or more classifications and metadata from each classification is needed. For example, for the Classroom Type Projectors product, linked using SupplierLink and WebLink, the calculated attribute will display the Class Owner attribute (ID = 'ClassOwner') from the classification.



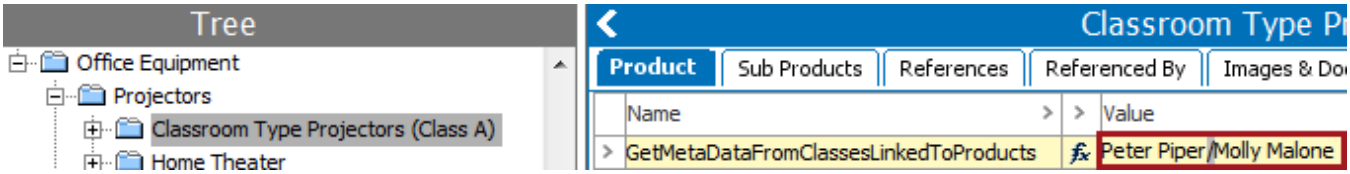


The calculated attribute formula could be written as:

```
{
meta := iterate(iterate(classificationproductlinks(), 'CPLClassification()'), 'value("Class
Owner")'),
listmeta := list2multivalue(meta)
}
concatenate(listmeta)
```

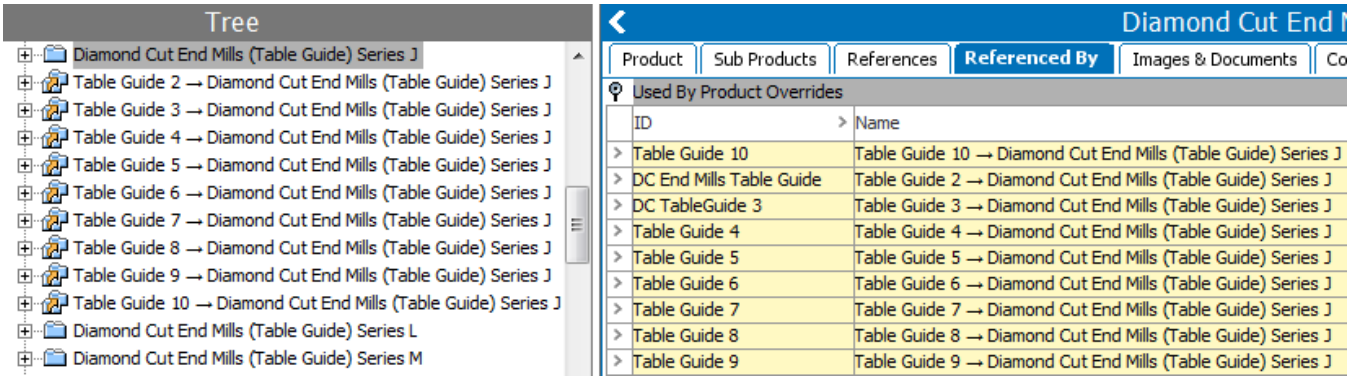
The ITERATE function shown in bold above returns a list of the internal 'labels' for the classifications. The surrounding ITERATE function scans through the list and gets the ClassOwner attribute value from the classifications.

The result is a list of the owners, separated by a slash (/).



Use Case: Calculated Attribute for Overrides

A product has been overridden several times.



Get Overridden By Products

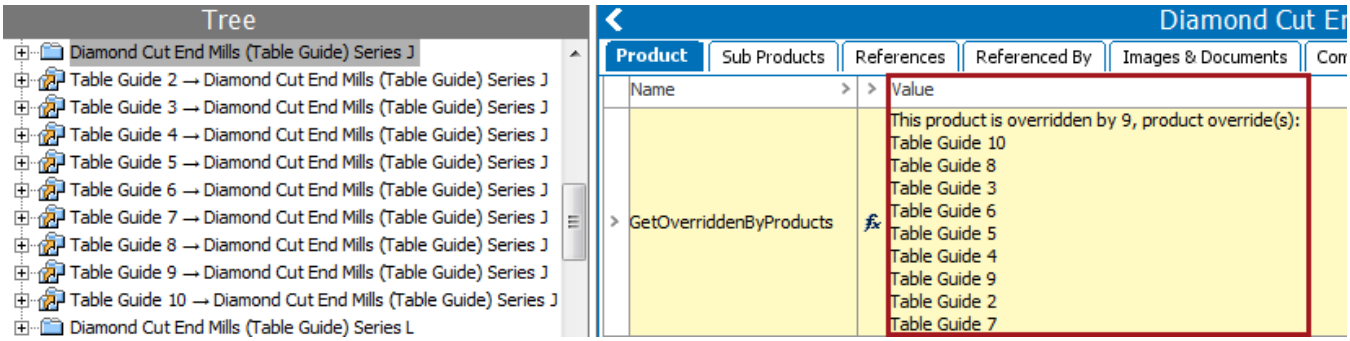
A calculated attribute on the originating product will list the names of the products that are overriding it. In this example, Diamond Cut End Mills (Table Guide) Series J has multiple overrides.

The formula for this calculated attribute could be written:

```
{
x := listlen(overriddenby()),
z := iterate(overriddenby(), 'stepname()'),
z := list2multivalue(z),
z := substitute(z, "<multisep/>", "
")
}
concatenate("This product is overridden by ", x, ", product override(s):
", z)
```

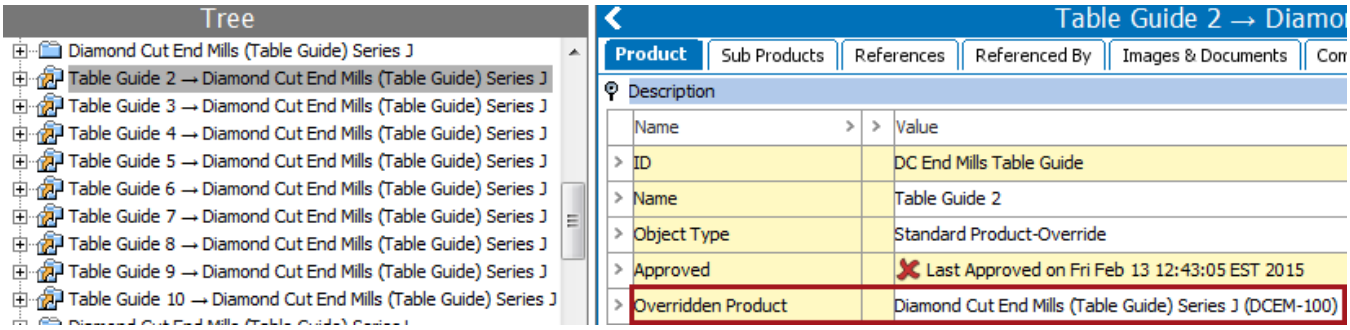
This formula uses the following elements:

- Adding a new line using the Enter key or adding "\n" allows the results to be displayed on individual lines.
- The LISTLEN() function gets the number of override items.
- The LISTLEN() function, used with STEPNAME(), gets the names of the override items.
- The list is converted to a multivalued text string.
- The final output is assembled using a text string, the list count, and the STEPNAME of the overrides with new lines for each name.



Get Overridden Product Object

A calculated attribute on the product override shows the name of the product being overridden. In this example, Table Guide 2 is overriding the product Diamond Cut End Mills (Table Guide) Series J with ID DCEM-100.

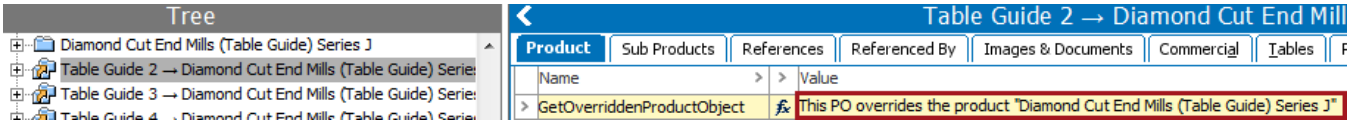


A calculated attribute on the product override could be written with the following formula:

```
{
x := iterate(overriddenproductobject(), 'stepname()'),
z := listitem(x,1)
}
concatenate('This PO overrides the product "', z, '"')
```

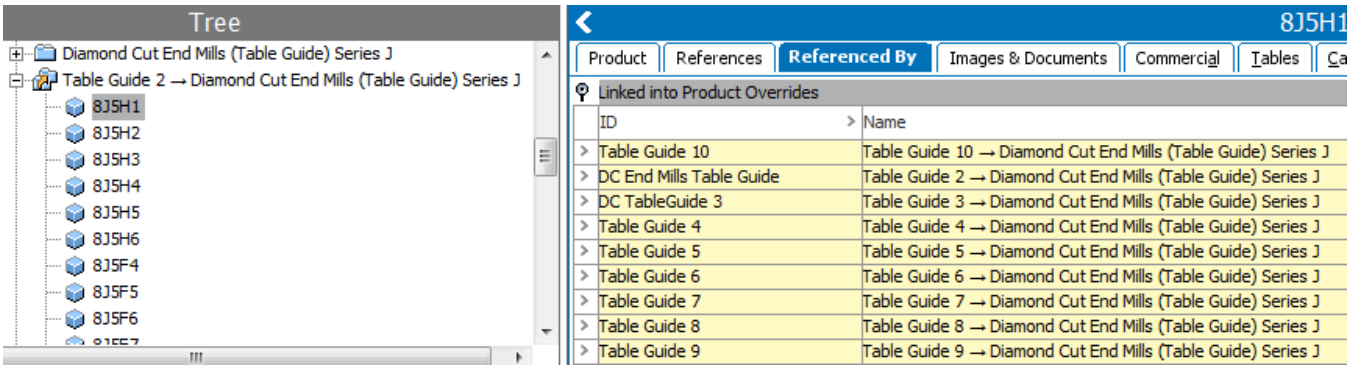
This formula uses the following elements:

- The ITERATE() function gets information from an object other than the currently selected object.
- Since the maximum list length is 1, use the LISTLEN() function to get the name of the override product.
- The final output is assembled using a text string and the name of the product that is overridden.



Get Product Override Parents

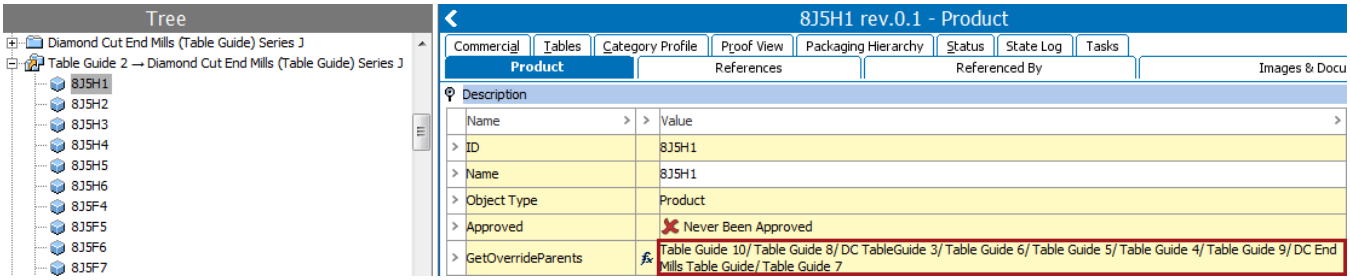
A product is used in one or more product overrides.



A simple metadata calculated attribute on the product that displays the IDs of the product overrides could be written with the following formula:

```
list(iterate(productoverrideparents(), "stepid()"), "/ ")
```

The result of this formula is:

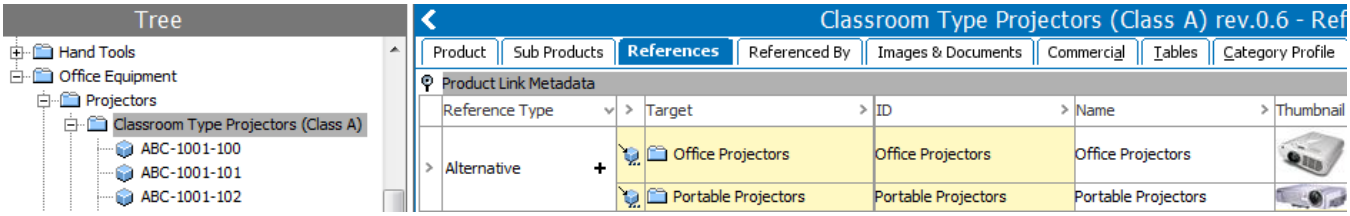


Use Case: Calculated Attribute for References

A product exists and has references to multiple other products.

Get List of References

For example, the Classroom Type Projectors product is linked to two other products: Office Projectors and Portable Projectors. A calculated attribute on Classroom Type Projectors will list the names of the referenced products.



The formula for this calculated attribute requires two stages or iterations:

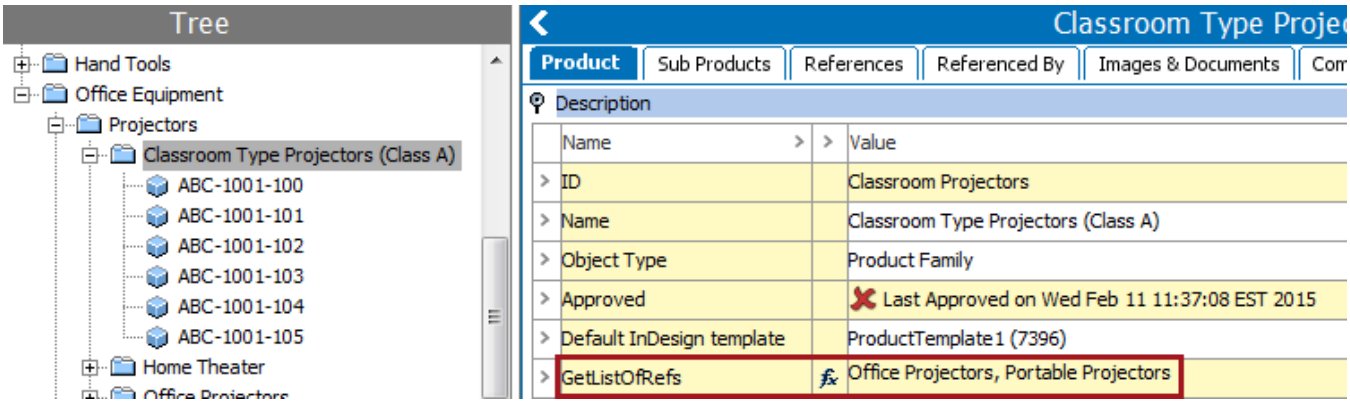
```
{
i := iterate(iterate(references("product", "Alternative"), 'referencetarget()'), 'stepname(
)'),
x := list(i, ", ")
}
concatenate(x)
```

The ITERATE() function is used to pull information that is not on the current product. The second ITERATE() function (shown as bold above) returns a list of 'labels' for all of the references that Classroom Type Projectors has made to other products (i.e., reference targets) with the reference Type ID of 'Alternative'.

This list of labels is not needed. Instead, another ITERATE() function is added, surrounding the previous ITERATE() function, and it gets the names of the referenced products from the labels:

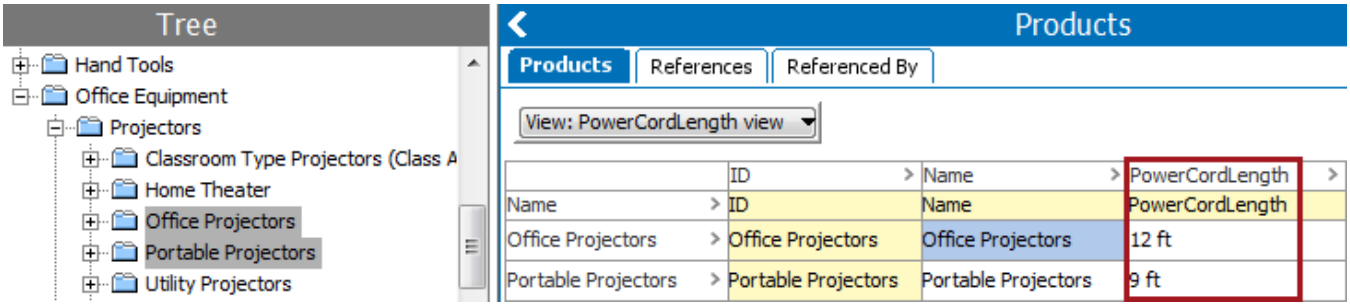
```
iterate(iterate(references("product", "Alternative"), 'referencetarget()'), 'stepname()')
```

The result of the calculated attribute displays the name of the references.



Get Value From a Referenced Product

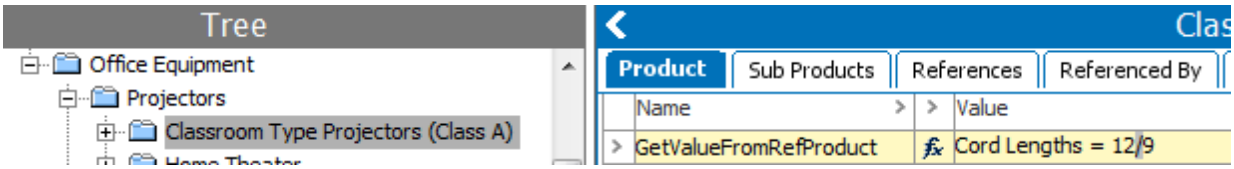
To expand on this, get an attribute *value from* the target products, instead of the *name of* the target product. For example, the target products have a 'PowerCordLength' attribute:



A calculated attribute with the following formula would allow the product Classroom Type Projectors to display the lengths of the power cords of the referenced products.

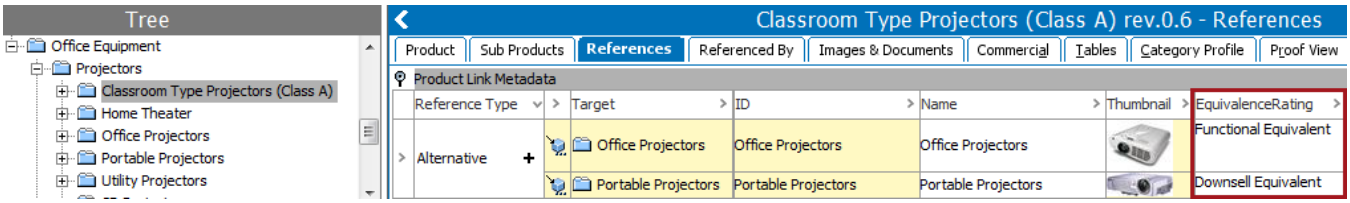
```
{
i := iterate(iterate(references("product", "Alternative"), 'referencetarget()'), 'value("PowerCordLength")')
}
concatenate("Cord Lengths = ",list2multivalue(i))
```

This is much like the prior example, but the LIST2MULTIVALUE() function is used since it automatically places a STEP tag <multisep/> between values. Based on the configuration of the tag, it is rendered as a slash in the value.



Get Metadata Value Located on the Link From Product-to-Product

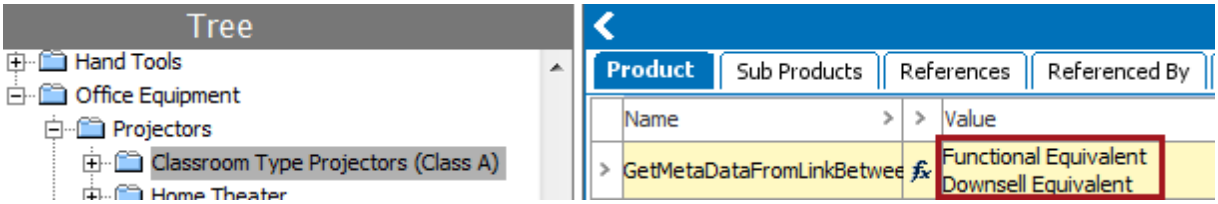
Additionally, metadata value on references can also be returned. For example, the product-to-product 'Alternative' reference has an 'EquivalenceRating' attribute:



A calculated attribute with the following formula will display the metadata on the original product.

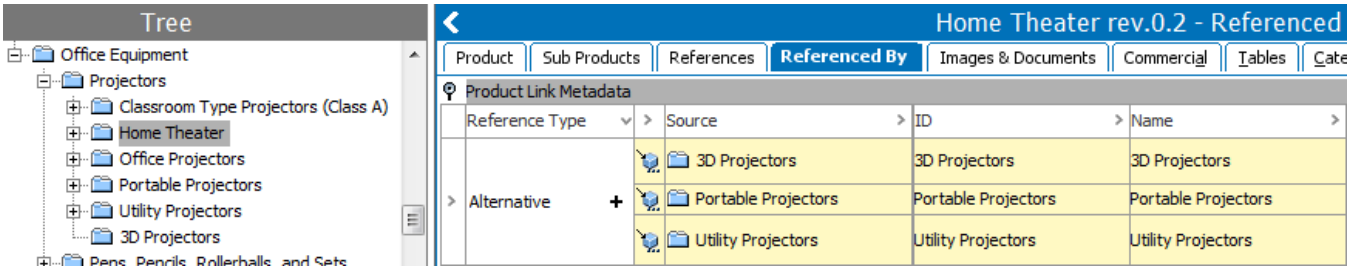
```
{
j := iterate(references("product", "Alternative"),'value("EquivalenceRating")'),
z := list2multivalue(j),
z := substitute(z, "<multisep/>", "
")
}
concatenate(z)
```

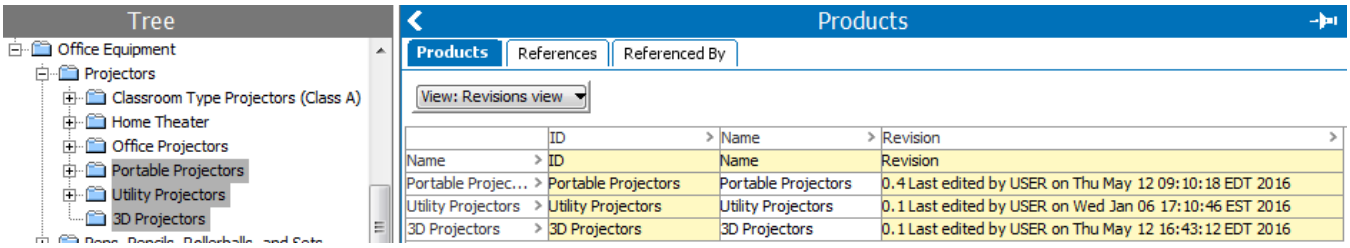
The default separator used by the LIST2MULTIVALUE() function is converted to a carriage return, so that the workbench displays a new line for each entry within the same value field.



Get Referenced By Using Reference Owner

A product is being referenced by two other products. In addition to the name of the products being referenced, data from those products is also needed. For example, the last time the 'Alternative' referenced products were edited.





The following calculated attribute formula could be written:

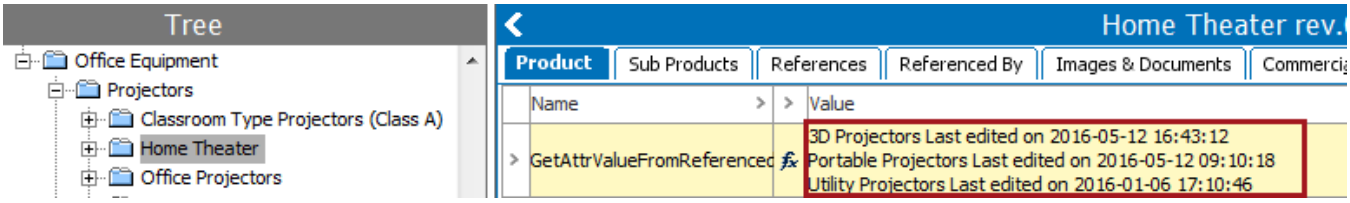
```
{
refnames := iterate(iterate(referencesto("product", "Alternative"), 'referenceowner()'), 's
tepname()'),
editdates := iterate(iterate(referencesto("product", "Alternative"), 'referenceowner()'), '
revisioneditdate()'),
combo := iterate(refnames, 'concatenate(item, " Last edited on ", listitem(editdates,inde
x))'),
fin := substitute(list2multivalue(combo), "<lt/>multisep/<gt/>", "
")
}
concatenate(fin)
```

This formula uses the following elements:

- List 1 'refnames' uses the REFERENCESTO() function in conjunction with the REFERENCEOWNER() function to get data from the referencing product, in this example it gets the STEPNAME().
- List 2 'editdates' gets a list of the last edited metadata on the referencing products.
- List 3 'combo' iterates through List 1 'refnames' and then concatenates the 'current' value indicated by the field 'item' (i.e., the referencing product's name), with the corresponding list item number in the List 2 'editdates'.
- The function LISTITEM(editdates,index) uses List 2 'editdates' and gets the entry indicated by the number in the index field. So if you are on item 1 in List 1 'refnames', it gets item 1 value in List 2 'editdates'. If you are on item 2, it gets item 2 value in List 2 'editdates', and so on.

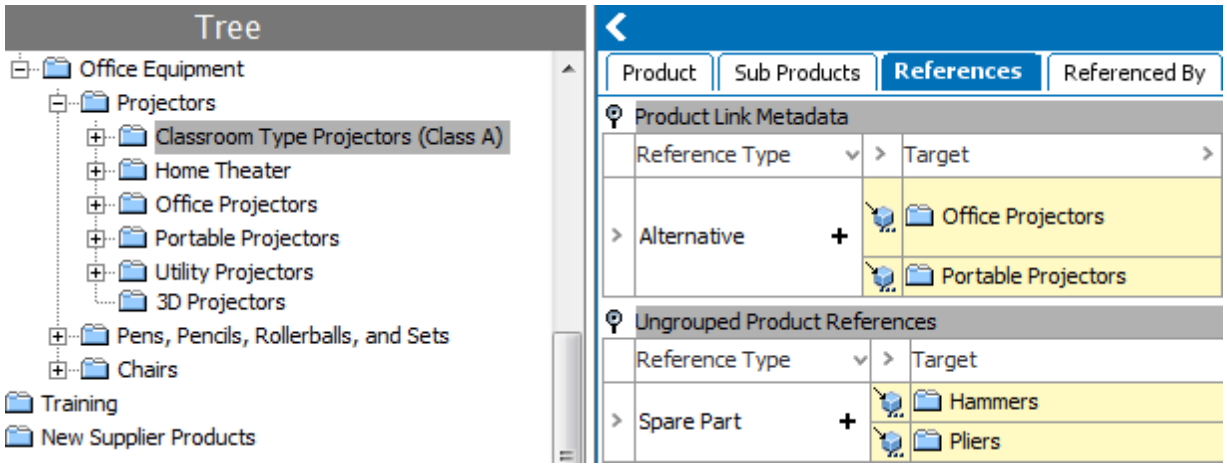
Important: When using a calculated attribute to 'marry' information from one list to the corresponding information in another list, the number of entries in each list MUST be identical.

The results display the referenced product and the corresponding last edit date.



Get Reference Type ID

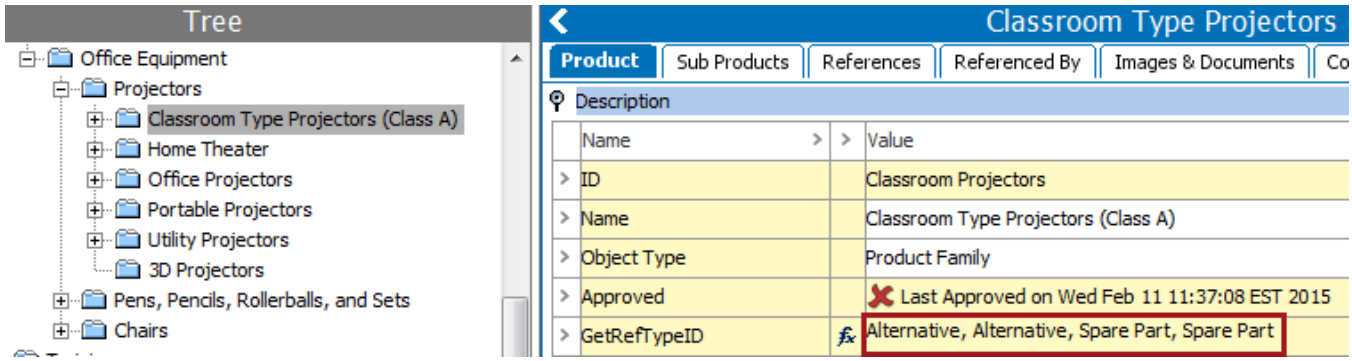
A product is being referenced by other products. A calculated attribute can display the reference types being used. For example, the product Classroom Type Projectors has four references to other products: two with the reference type of Alternative and two with the reference type of Spare Part.



The following calculated attribute formula could be written:

```
{
refids := iterate(references("product"), 'referencetypeid()'),
refids := list(refids, ", ")
}
concatenate(refids)
```

The results display each of the reference type IDs.



Although accurate, the display could be more informative with some modifications. You could filter the list so that only unique values are returned (resulting in only two reference types) or you could list each of the names and/or IDs of the referenced products together with the reference type.

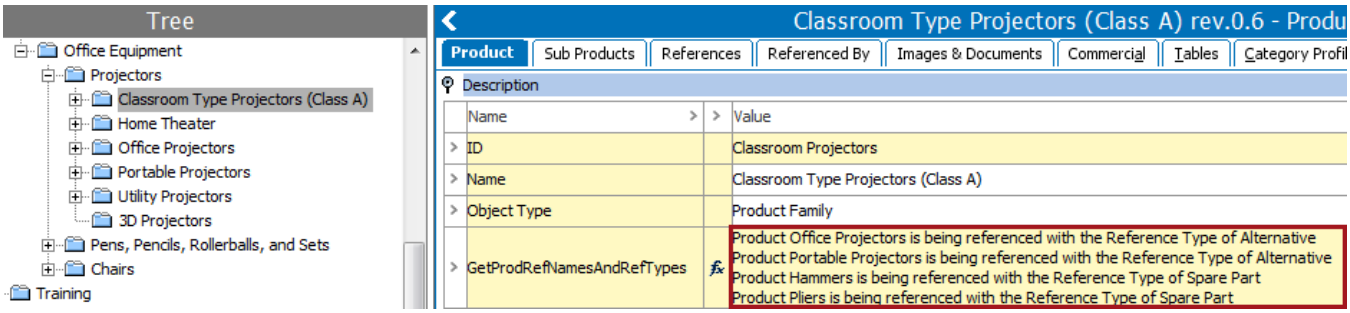
Similar to the above example of referencing product names and their edit dates, the formula could be written:

```
{
```

```
refnames := iterate(iterate(references("product"), 'referencetarget()'), 'stepname()'),
reftypes := iterate(references("product", ""), 'referencetypeid()'),
combo := iterate(refnames, 'concatenate("Product ", item, " is being referenced with the Re
ference Type of ", listitem(reftypes,index))'),
fin := substitute(list2multivalue(combo), "<multisep/>", "
")
}
concatenate(fin)
```

Again, three lists are made: a list of the names of the referenced products, a corresponding list of reference types, and a list that combines the corresponding values in list one and two. Finally, added text strings clarify the results.

The results display each product being referenced, as well as the reference type ID.



Conditional Attribute Display

As is often the case, an attribute may not be relevant unless another attribute holds a certain value. For example, attributes specific to hazardous materials may become valid if the attribute 'Has Lead' has a value of 'Yes'. To facilitate this, STEP allows for attributes to be conditionally displayed in Web UI and Smartsheets.

A condition can be configured in two ways: as metadata on a Description Attribute or as a Specification Attribute link. When configured, it indicates the circumstances under which the relevant attribute is valid.

This enables business users to steer the Web UI and Smartsheets behavior for end users without requiring use of the STEP Web UI Designer.

Note: Conditional attribute support does not apply to STEP Workbench as all attribute visibility in the workbench is determined solely by user privileges.

Terminology

For additional context, see relevant definitions below:

- **Conditional Attribute** - This attribute appears as a metadata field on Description attributes and on Specification attribute links in the Product Hierarchy. Conditional expressions are stored on this attribute.
- **Conditional Expression** - Validity rules for the Conditionally Valid Attribute. Expressions are stored in the Conditional Attribute field for the relevant attribute or attribute link.
- **Conditionally Valid Attribute** - These attributes are only valid under the conditions specified by a conditional expression. In Web UI, Conditionally Valid Attributes are only displayed to the user when the Conditionally Driving Attribute is populated according to the Conditional Expression.
- **Conditionally Driving Attributes** - Attributes that are specified in a conditional expression. Conditionally Valid Attributes rely on populating this attribute with the correct value. In Web UI, Conditionally Driving Attributes are always displayed.

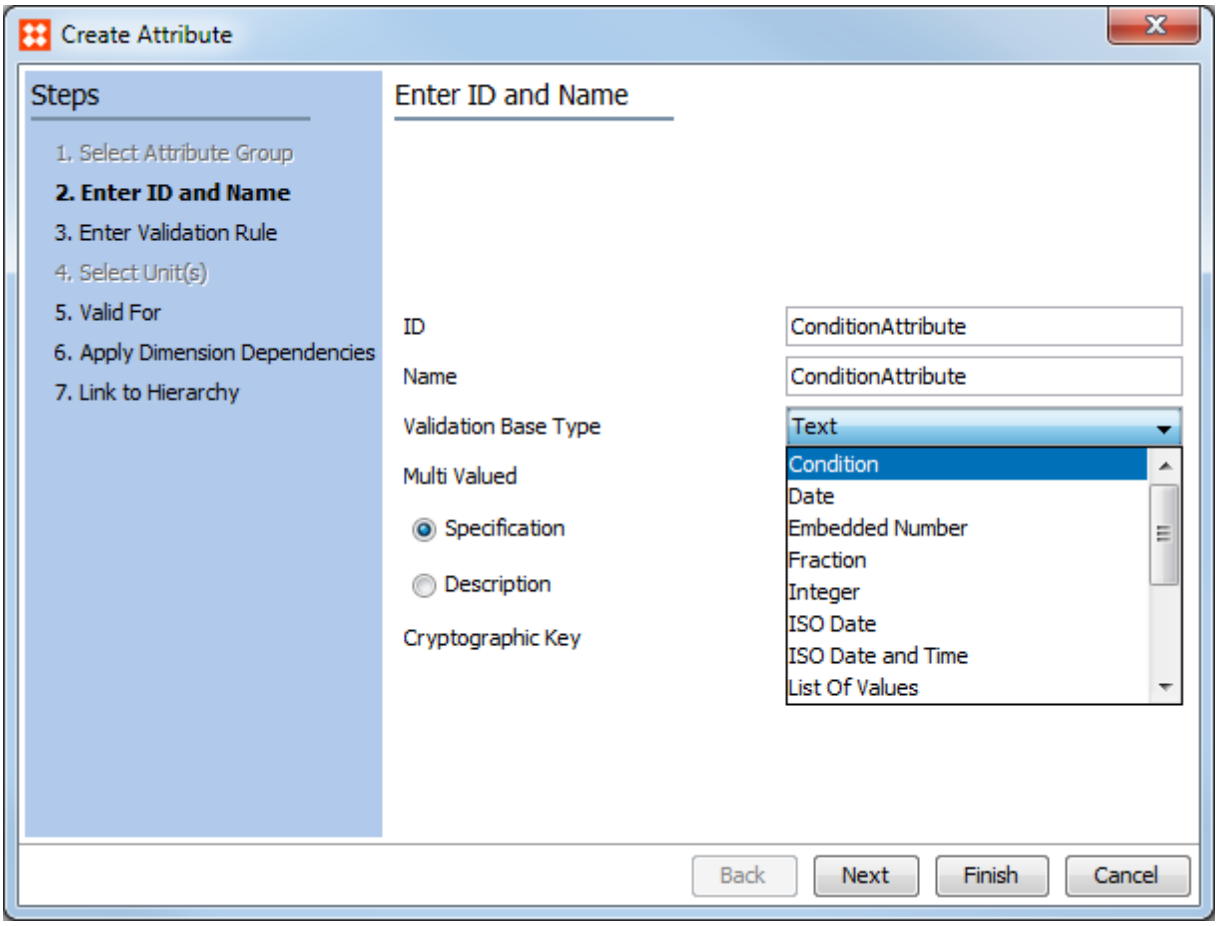
Creating the Conditional Attribute

Before conditions can be set on attributes, the Conditional Attribute must be created.

Note: Only one Conditional Attribute can be active on the system at any given time. To be clear, the Conditional Attribute is simply the attribute that holds the logic for the Conditionally Valid Attributes. Therefore, any number of Conditionally Valid Attributes and their associated expressions can be configured using a single Conditional Attribute.

---

1. In a relevant **Attribute Group** in **System Setup**, create a **Description Attribute** with the **Base Validation Type** called **Condition**.

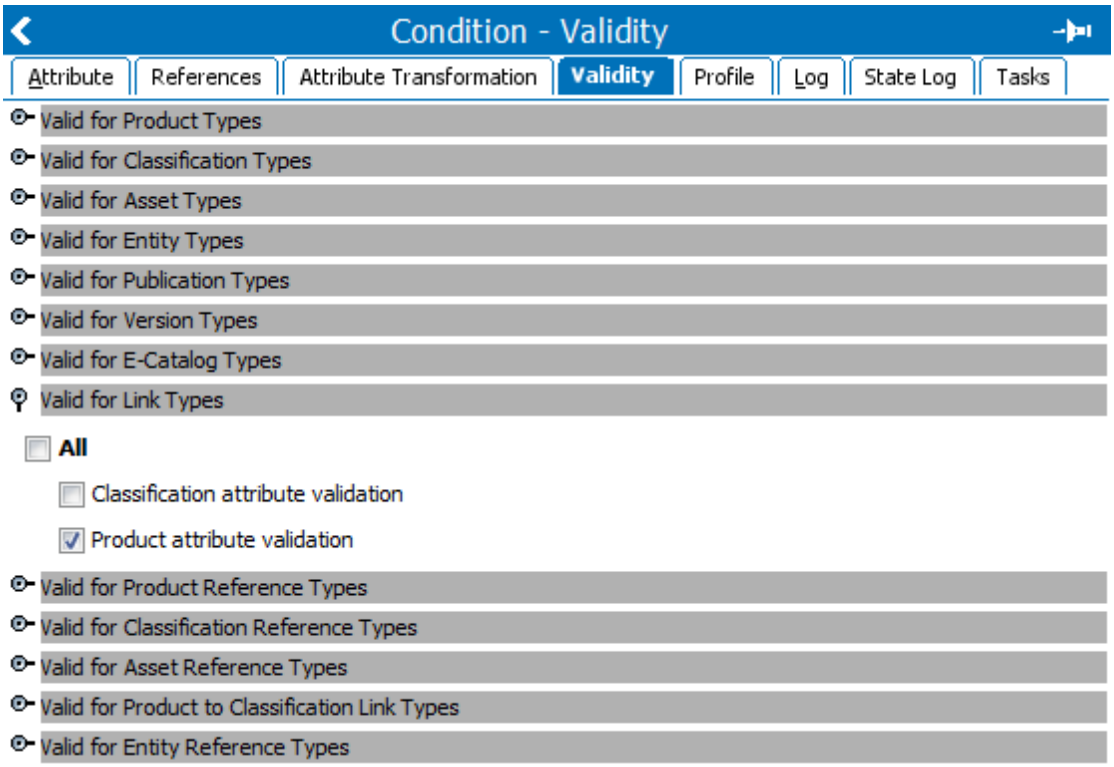


2. Fill out the rest of the relevant information and **Finish** the creation of the attribute.

Important: It is recommended that attributes used for conditional validity (as the Conditional Attribute or as a Conditionally Driving Attribute) do not have dimension dependencies.

---

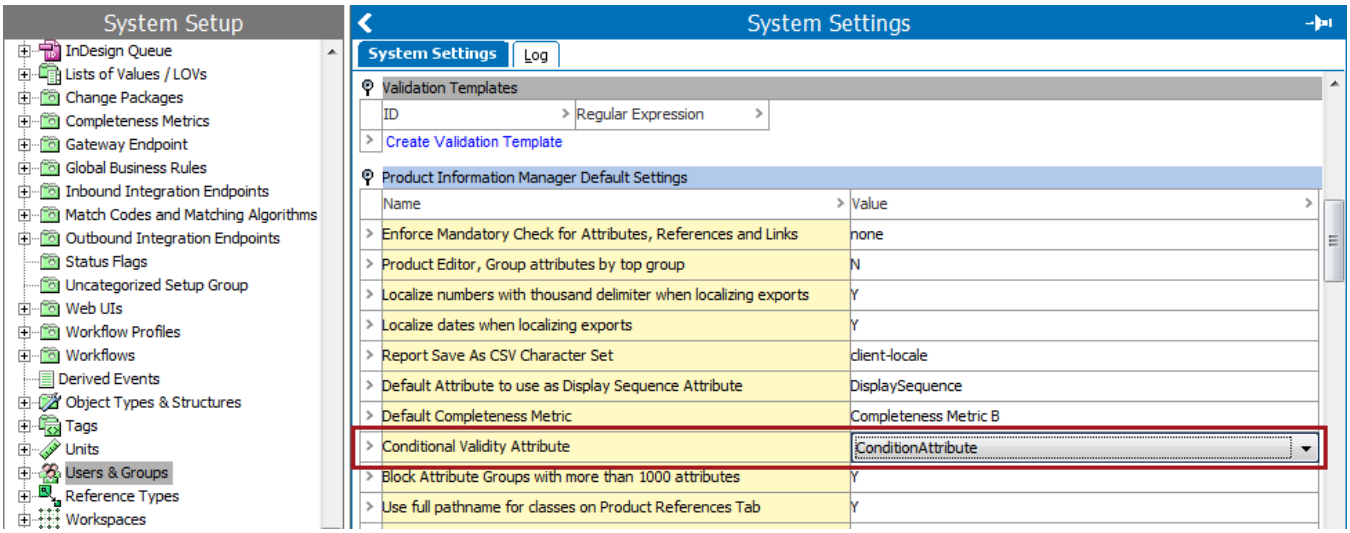
3. Set the validity of the attribute for the **Link Type** called **Product attribute validation**.



Activating a Conditional Attribute

Once created, the Conditional Attribute must be activated.

1. Navigate to **System Settings** and select **Users & Groups**.
2. Locate the **Product Information Manager Default Settings** section.
3. In the **Conditional Validity Attribute** field, select the Conditional Attribute configured in the previous section.



Configuring Conditionally Valid Attributes

Where and how you configure a condition for an attribute depends on whether it is a Description or Specification attribute.

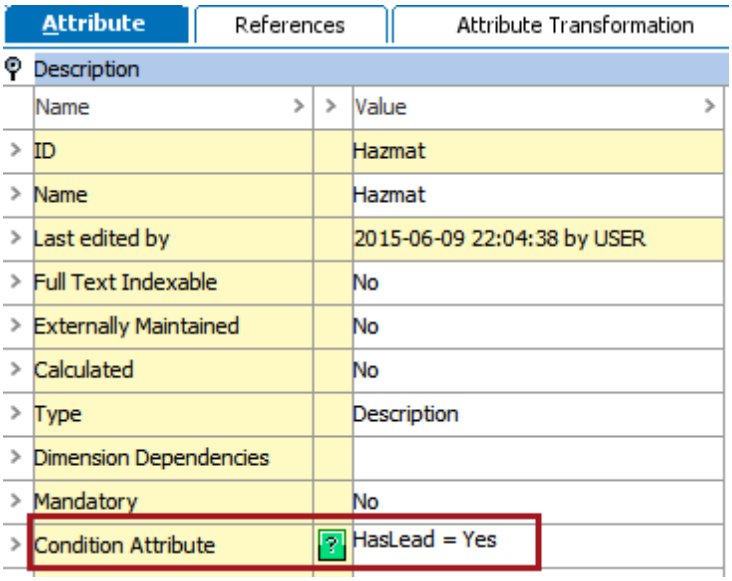
Description Attributes

The condition for a Description Attribute is configured as metadata on the attribute itself. Before you can add it to an attribute, however, the Conditional Attribute must be made valid for the Attribute object type.

1. In **System Setup**, navigate to **Objects & Structures** and expand **Basic Object Types**, then **Attribute Groups**, and then click on **Attribute**.
2. In the **Valid Attributes** section of the **Object Type** tab, click **Add Attribute** and select the Conditional Attribute from the list that appears.

   For more information, see the **Description Attribute** section of the **System Setup / Super User Guide** documentation.

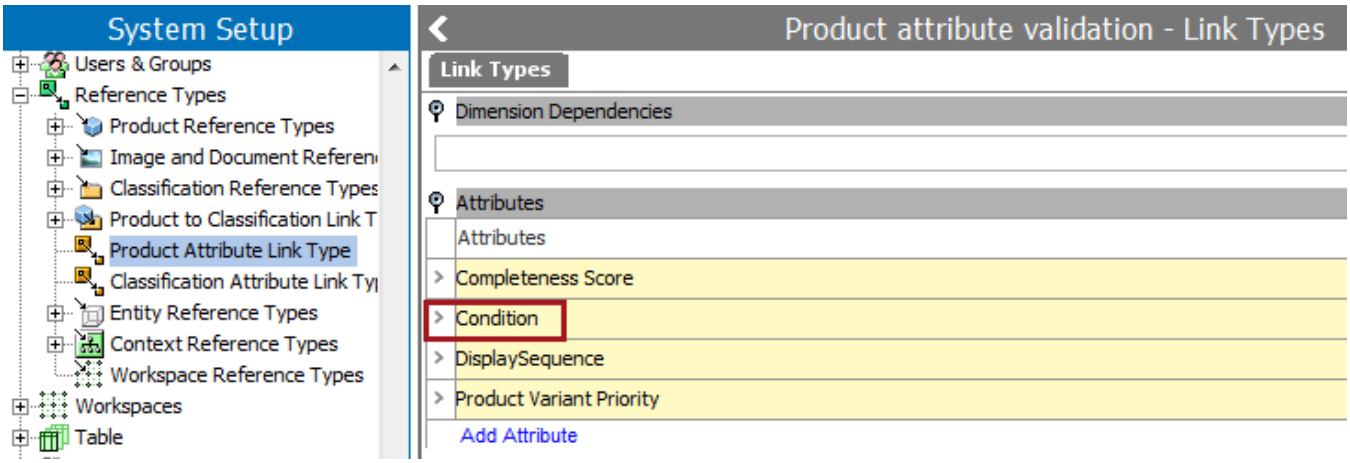
3. Next, navigate to **System Setup** and select the desired Description Attribute.
4. In the Conditional Attribute field, enter the desired condition rules for the Conditionally Valid Attribute to follow.



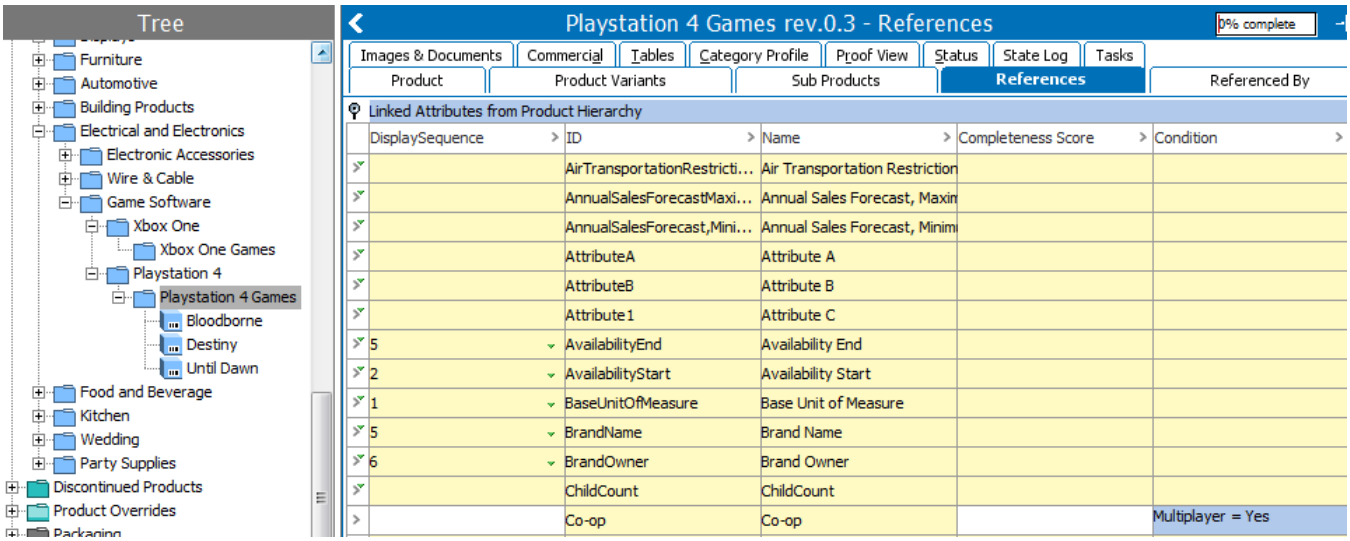
Specification Attributes

The condition for the Specification Attribute is configured by linking it to the relevant products on the References tab. Before you can add it to a product, however, the Conditional Attribute must be made valid for the Product Attribute Link Type object type.

1. In **System Setup**, expand **Reference Types** and open the **Product Attribute Link Type** node.
2. Click **Add Attribute** and select the Conditional Attribute.



3. Next, navigate to the **Tree** and open the relevant node in the product hierarchy.
4. On the list of linked attributes, locate the desired Specification Attribute. In Conditional Attribute column, enter the conditional expression.



Writing Conditional Expressions

Conditional expressions are written directly in the STEP Workbench text editor.

The expressions are written in a simple syntax in the format [AttributeID] = [Value1];[Value2], where multiple values act as an 'OR' operator. For example, a Condition Attribute called 'Condition Attribute' specifies that the 'Hazmat' attribute will only appear if the 'HasLead' attribute has a value of true, using the syntax [AttributeID] = [Value1]:

Syntax forms:

- Condition: <language id><attribute id> = <value list>
- Language ID: .*:|<nothing>

- Value list: <value>;<value list>
- Value: .*

Rules that must be observed for the conditions:

- The used attribute must not have spaces in its ID.
- The used attribute must not have ':' in its ID.
- If a value contains ';' it must be escaped with \'
- If a value contains '\' it must be escaped with \'

Note: Conditional expressions using attributes with LOVs that utilize value IDs will evaluate the value ID. For example, assume a Yes / No LOV exists that uses value IDs , where 1=Yes and 2=No. The conditional expression for this must use the value ID, e.g. 'AttributeA = 1' if the attribute should only be available when the value of AttributeA is 'Yes'.

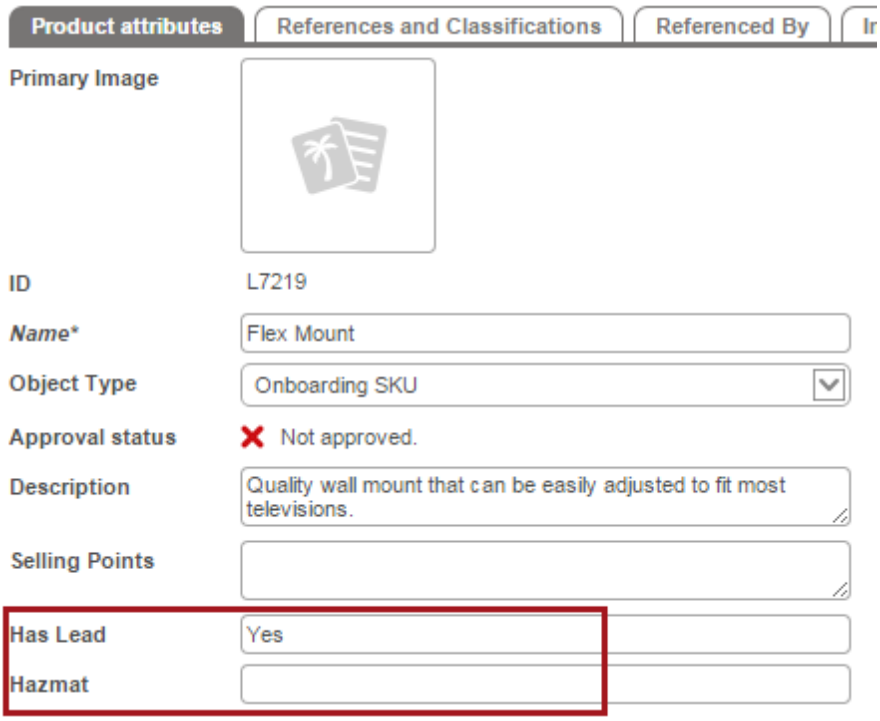
For more information on how this impacts Web UI, see **Conditional Attributes** in the **Web UI Getting Started** documentation, and the **Conditional Attributes** section of the **Business Rules** documentation.

Note: Conditional attributes can also be exported using SmartSheets. When exporting products with conditional attributes any cells containing invalid conditionally displayed values will be automatically locked for editing. In order to unlock the cell and make changes, the user must first fulfill the proper conditions for the attribute and then validate the sheet. For more information of exporting SmartSheets see the **Creating SmartSheet Data and Template Configurations in the Export Manager** topic within the **SmartSheet for Excel** documentation.

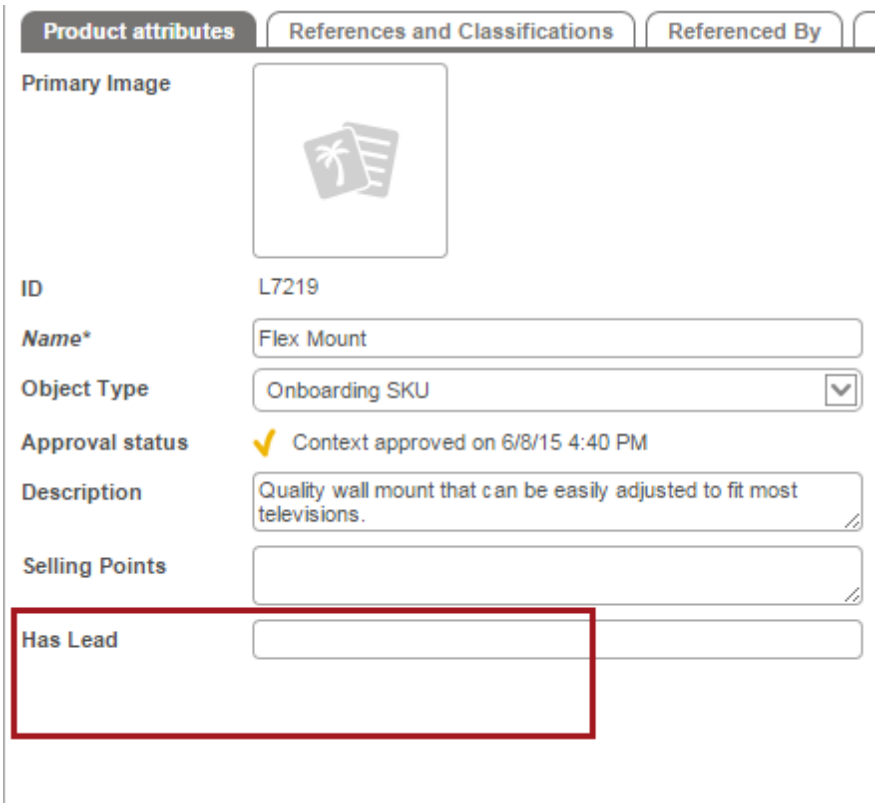
Conditional Attribute Example

In the below example, notice that the 'Hazmat' attribute is available when 'Has Lead' is set to 'Yes', but is unavailable when 'Has Lead' is left unpopulated (or set to 'No').

Set to 'Yes':


Product attributes
References and Classifications
Referenced By
Primary Image
ID
L7219
Name*
Flex Mount
Object Type
Onboarding SKU
Approval status
Not approved.
Description
Quality wall mount that can be easily adjusted to fit most televisions.
Selling Points
Has Lead
Yes
Hazmat

No Value:


Product attributes
References and Classifications
Referenced By
Primary Image
ID
L7219
Name*
Flex Mount
Object Type
Onboarding SKU
Approval status
Context approved on 6/8/15 4:40 PM
Description
Quality wall mount that can be easily adjusted to fit most televisions.
Selling Points
Has Lead

Description Attributes

These attributes hold characteristics, typically used for internal or search purposes. The contents of description attributes are not normally intended for direct publishing.

Description attributes are directly related to a specific object type such as product, classification, image, document, publication, and are automatically displayed in the **Description** field.

Description attributes can also be applied to references, such as product to product references and attribute to product references.

The following are description attribute examples:

- Name (name of the specific object)
- Keywords

Description attributes are typically made globally valid for specific object types.

A description attribute value is the actual content associated with a description attribute.

Description Attribute	Value
Loading date	05-jul-2008
Image quality	• High • Medium • Low

Meta Attributes

A meta attribute is a description attribute that can be applied to the link between two objects. A link is, for example, a product to product reference or an attribute to product reference.

Meta attributes hold information specific to a link between two nodes.

The following are meta attribute examples:

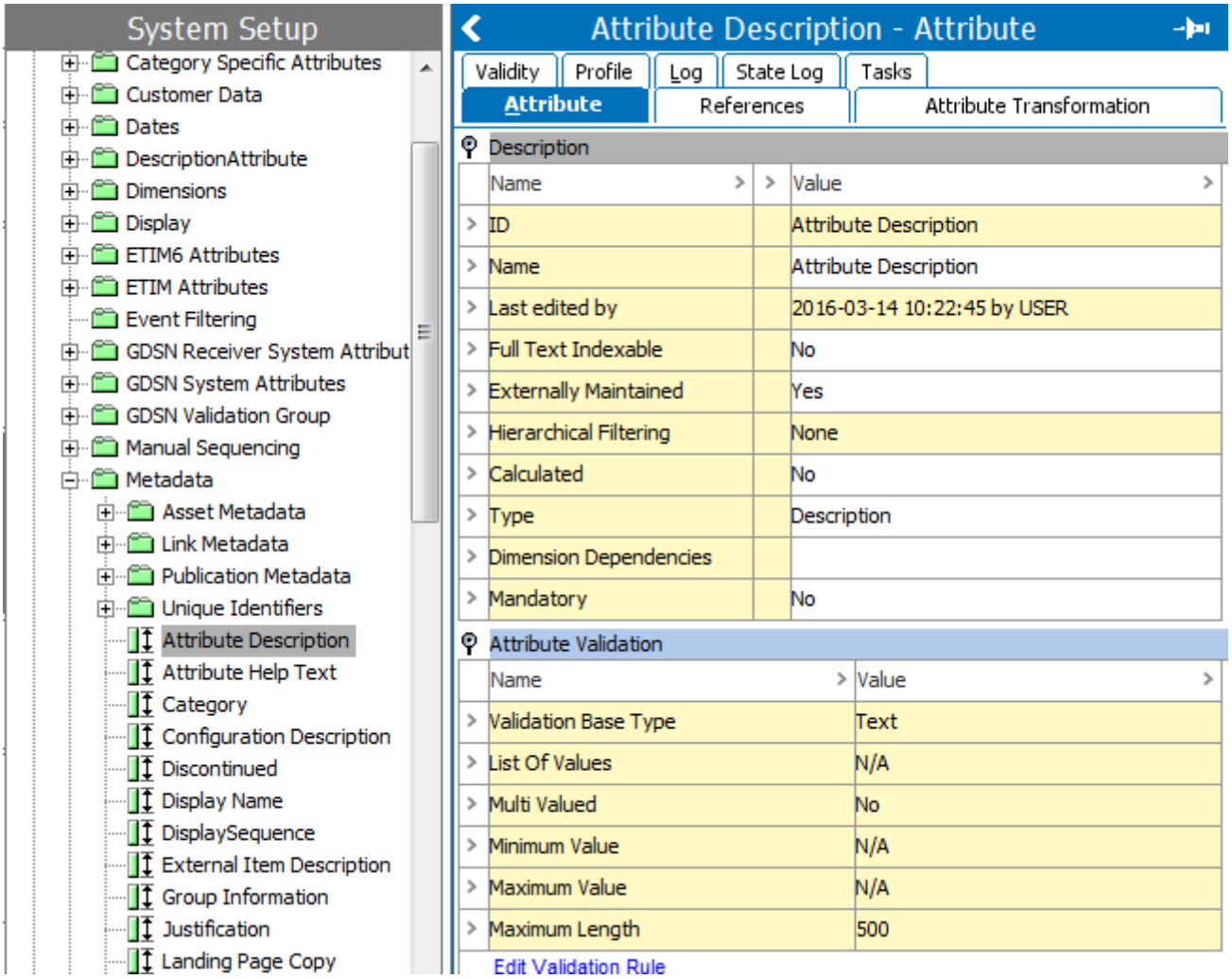
- Display sequence, as applied to the link between a product node and an attribute node. It indicates a display sequence for the attributes relevant to the specified product folder or product.
- Mandatory, as applied to the link between a product node and an attribute node. It indicates that the product and all its sub products must have a value for the specified attribute.

Attribute Metadata on Attributes

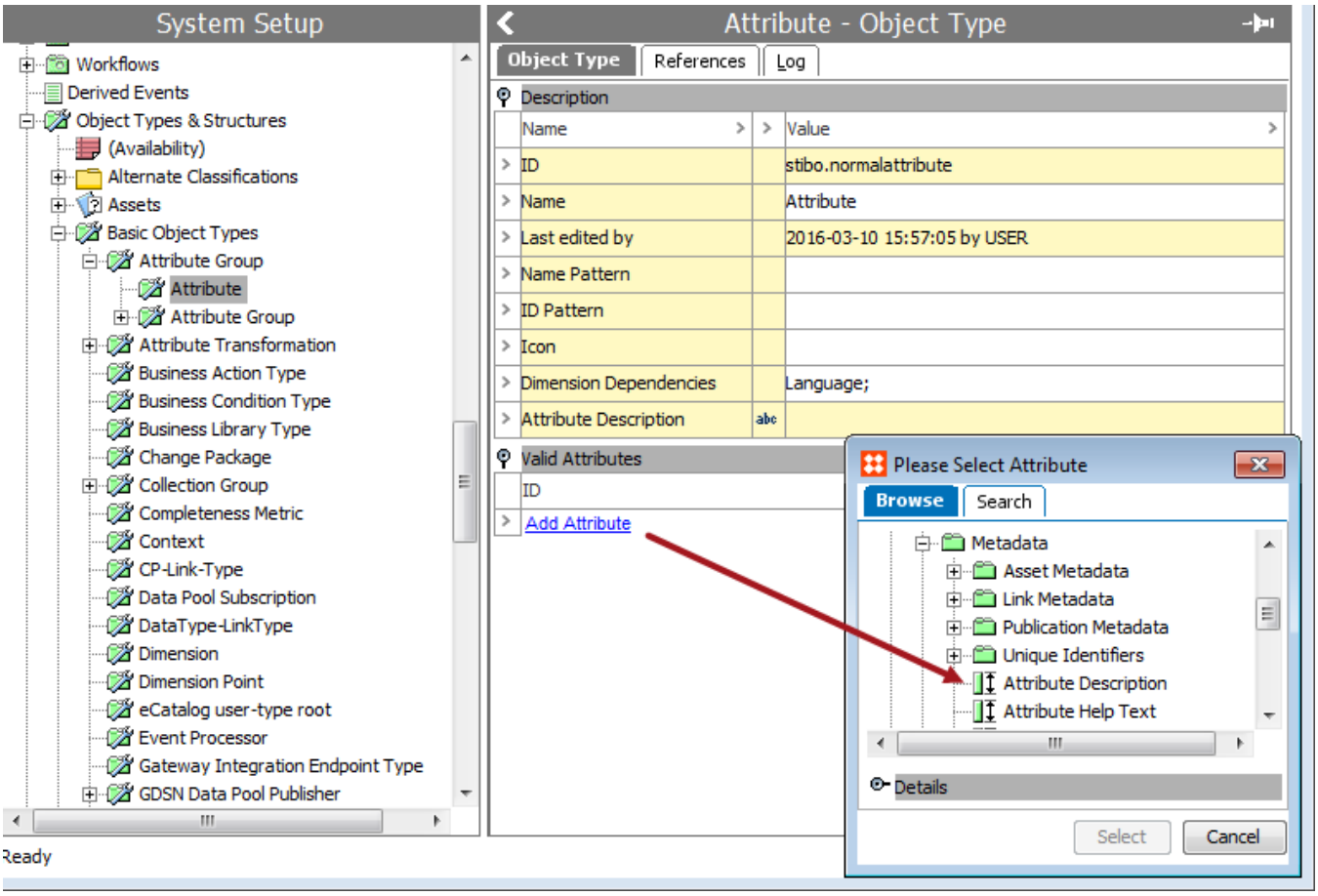
There are occasions when it will be necessary to add a metadata (Description) attribute to an attribute. For instance, you might need an attribute to describe what an attribute is used for, or to provide information about an attribute in the help text that appears in the Web UI. Creating and maintaining metadata attributes on attributes is similar to any other metadata attribute in every way *except* in defining the validity of the attribute. Follow these steps to make an attribute valid for another attribute.

Steps to Creating a Metadata Attribute on an Attribute

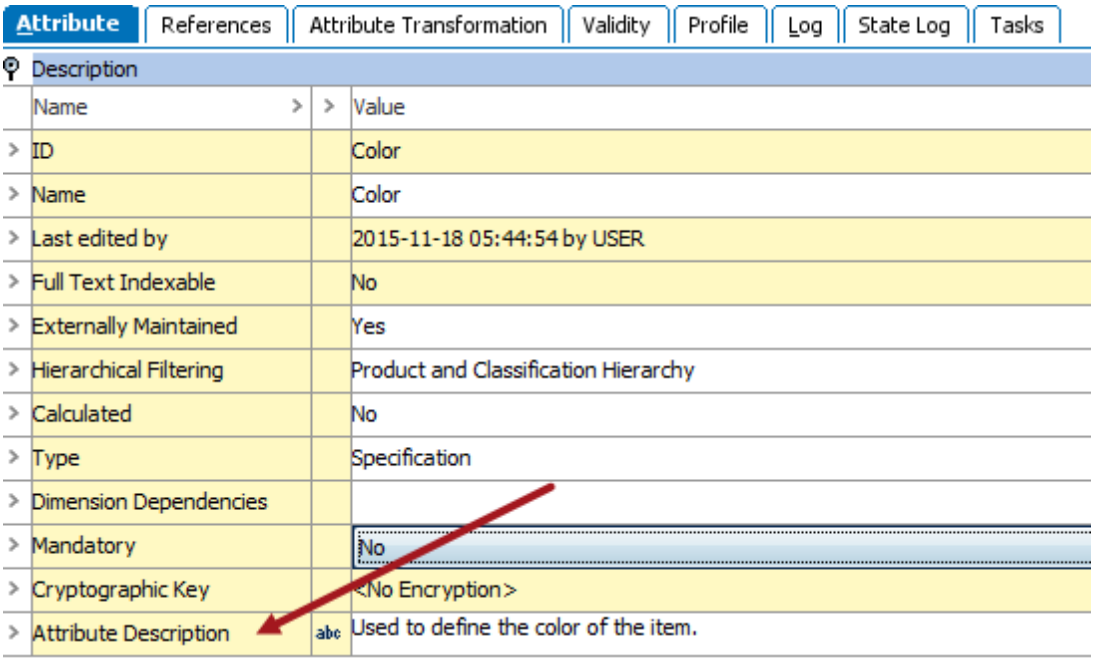
1. Create the metadata (Description) attribute following the same procedure as you would to create any Description attribute. See **Creating Attributes** in the **System Setup** / **STEP Super User** documentation for more information.



2. After creating the metadata attribute, the attribute must be made valid for the **Attribute** object type. Navigate to the Attribute object type under System Setup > Basic Object Types > Attribute Group. Select the Attribute object type, then click **Add Attribute**. The **Please Select Attribute** dialog displays.
3. In the **Please Select Attribute** dialog, browse to or search for the Description attribute that was just created, then click **Select**.



4. The metadata attribute is now ready for use.


Attribute
References
Attribute Transformation
Validity
Profile
Log
State Log
Tasks
Description
Name
Value
ID
Color
Name
Color
Last edited by
2015-11-18 05:44:54 by USER
Full Text Indexable
No
Externally Maintained
Yes
Hierarchical Filtering
Product and Classification Hierarchy
Calculated
No
Type
Specification
Dimension Dependencies
Mandatory
No
Cryptographic Key
<No Encryption>
Attribute Description
abc
Used to define the color of the item.

Description Attributes and Object Types

Description Attributes can be linked to all Object Types. A Description Attribute that is linked to an Object Type will appear automatically in the appropriate editor for all Objects of this Object Type.

Note: Description Attributes can also be linked to Product, Classification and Asset Object Types in the Attribute Editor: **Attribute** tab under Validity.

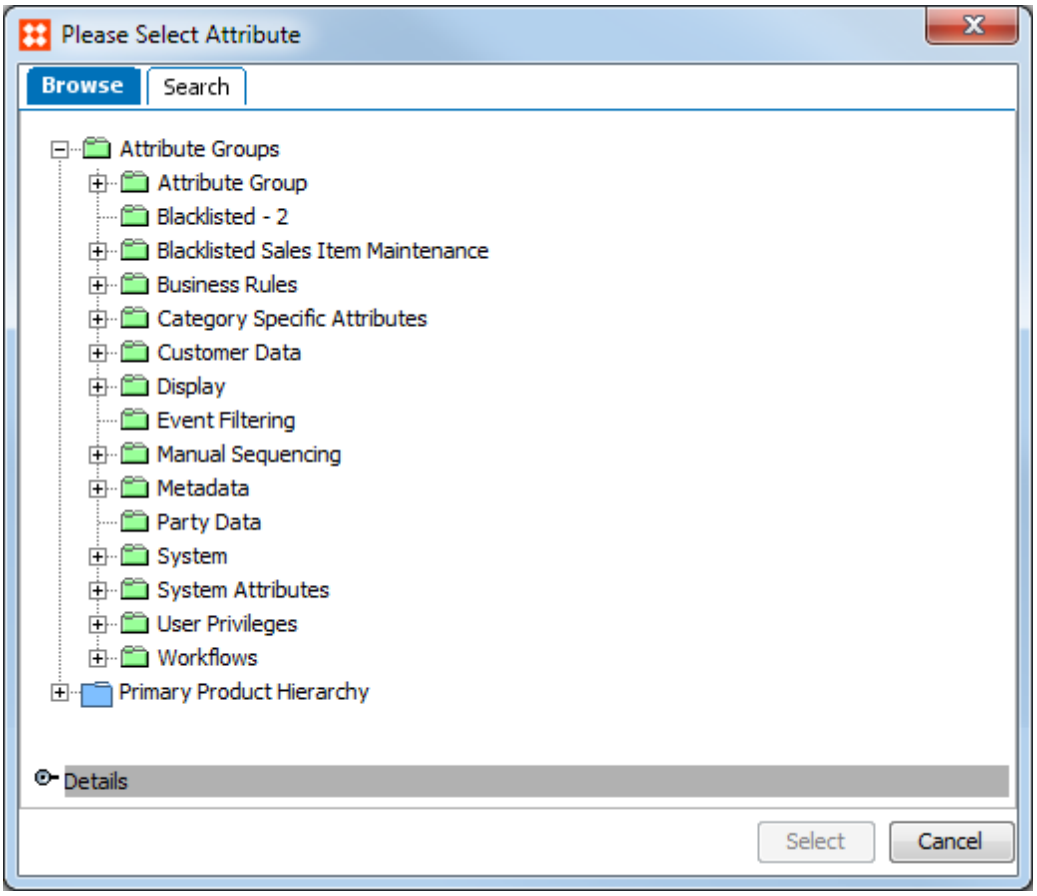
Some of the Object Types and linked Attributes are:

Object Type	Attribute
Product Object Types	The Attribute will appear automatically for all Objects of the specified set of Product Object Types. By default, the Attribute will be valid for all Product Object Types.
Classification Object Types	The Attribute will appear automatically for all Objects of the specified set of Classification Object Types.
Asset Object Types	The Attribute will appear automatically for all Objects of the specified set of Asset Object Types (Images & Documents).

For a description of all Object Types, see the Basic Object types section and Object Types for Classification, Product, Assets and Entities.

Linking Description Attributes to an Object Type

1. In System Setup, open **Object Types & Structures** and select the relevant Object Type to display the **Object Type** Editor.
2. Open the **Valid attributes** flipper and click the **Add Attribute** link to display the **Select Attributes** dialog.



3. Search or browse to the relevant **Description** Attribute, select it and click the **Select** button.

Note: The dialog shows both Description and Specification Attributes. A message is displayed when a Specification Attribute is selected.

4. The Description Attribute displays as a field in the **Description** flipper in the appropriate editor for nodes of the selected Object Type.

About Dimension Dependent Attributes

Dimension dependencies on Attributes are set when creating the Attribute, or in the **Attribute** Editor, accessed through the **Attribute** tab: **Dimension Dependencies** field.

A Dimension dependent Attribute can have different values for different Dimension Points. The actual Value is thus determined by the Context in which the Attribute is presented.

- Example (1): The Value depends upon the **Language Dimension**. This will typically be relevant for Attributes with validation base type “text”. The Attribute can have a specific Value for each Dimension Point (e.g. English, German, French) in the **Language Dimension**.
- Example (2): The Value depends upon the **Price Dimension**. This will typically be relevant for Attributes that contain pricing information. These Attributes can have specific Values for each Dimension Point (e.g. GBP, EUR, USD) in the **Price Dimension**.

Setting Dimension Dependencies thus allows different values for the Attribute to be displayed or published, according to the specified Context.

The following cases will illustrate how the Dimension Dependency settings on attributes work, and the influence it has on the Values.

Case 1

For a Product, the Value of “Warning label required” will always be true or false, regardless of the Context. The Attribute will only be used to indicate a “Warning” symbol (flag) in connection with a specific publication. Translation is thus not relevant.

Attribute	Value	Dimension Dependency	Description
Warning label required	true/ false	No dependency	A Product will have the same Value, true or false, for this Attribute in all Contexts.

Case 2

For a Product, the Value of “Warning label required” will always be true or false, as in case 1, but because the Attribute value is presented as “text” in the catalog, different Values are required for different languages.

Attribute	Value	Dimension Dependency	Description
Warning label required	true/ false	Language	Values can be translated and then be presented as e.g. Yes (Eng) = Ja (Ger) = Oui (Fr), depending on the actual Context.

Case 3

For a Product, the Value of "Warning label required" may be either true or false, depending on which market the Product is sold in. The Attribute will only be used to indicate a "Warning" symbol (Flag) in connection with a specific publication. Translation is thus not relevant.

Attribute	Value	Dimension Dependency	Description
Warning label required	true/ false	Market	A "Warning" label is not required in all markets, so the Flag will only appear in the relevant publications. Depending on the context, the values may be true or false.

Case 4

For a Product, the Value of "Warning label required" may be either true or false, depending on which market the Product is sold in. But because the Attribute value is presented as "text" in the catalog, different Values are required for different languages.

Attribute	Value	Dimension Dependency	Description
Warning label required	true/ false	Language Market	A "Warning" label is not required in all markets, so the "text" will only appear in the relevant publications The "text" will then be presented as e.g. Yes (Eng) = Ja (Ger) = Oui (Fr), depending on the actual Context. The actual Context must then include the relevant dimension points for both the Market and the Language dimension.

Attribute Display Sequence

Per default, attributes shown in the Product Editor of a product are shown in alphabetical order. If an Attribute Display Sequence Attribute has been set on your STEP system, the attributes shown in the Product Editor are sorted according to the order set in the Display Sequence attribute.

Important: The Display Sequence attribute has only an impact on the attributes shown in the Product Editor. It has no impact on the order in which attributes are exported.

For more information about how to setup an Attribute Display Sequence, see the STEP PIM Super User Guide. For more information about the Display Sequence attribute, see The References Tab in the STEP User Guide.

Sorting Attributes

The attributes shown in the Product Editor can be sorted according to the order set in the Display Sequence attribute. This can be done in two ways:

- by setting a number in the **Display Sequence** field of the attribute itself.
- by setting a number in the **Display Sequence** column under **Linked Attributes**.

Setting a Display Sequence number on the attribute

1. In [System Setup], unfold **Attribute Groups**, then click the relevant attribute.

   An **Attribute** Editor appears.

2. In the Display Sequence field, type a number.
3. Type the appropriate sequence number for each attribute in an attribute group. You can choose to set a sequence by using eg. 10, 20, 30, etc., or by using 1, 2, 3, etc. If you want to add attributes in between, you can then use for instance 15 in the first case, or 1.5 in the latter case.

   Note: If you do not know which attribute on your system is set up as the Display Sequence attribute, you can go [System Setup], click **Users & Groups**, and verify the name of the Display Sequence attribute under **Default Attribute to use as Attribute Display Sequence Attribute**. Alternatively, click a product, click the **References** tab. Under **Linked Attributes**, the first column shows the Display Sequence attribute on your system.

Setting a Display Sequence number under Linked Attributes

1. In [Tree], unfold the Product Hierarchy, click the relevant product folder or product.
2. Click the **References** tab.
3. Under Linked Attributes, type a number in the first column (the Display Sequence attribute column) in front of the relevant attribute.

Linked Attributes

StiboDisplaySequence	ID	Name	Attribute Groups	Mandatory	Inherited from	Visibility
40	StiboColour1	Colours	StiboTestAttributes	☐	Chairs	English
15	StiboDescriptionLong	LongDescription	StiboTestAttributes	☐		English
	NumberAttribute	NumberAttribute	Global Product Attributes	☐	Stibo Area	All Languages
50	StiboPrice1	Price	StiboTestAttributes	☐	Chairs	English
20	StiboDescriptionShort	ShortDescription	StiboTestAttributes	☐	Chairs	English
10	Title	Title	StiboTestAttributes	☐	Chairs	English

Link to Attribute

Note: You are only able to edit the Display Sequence attribute column if the attribute is linked in at the Product Hierarchy level that you have selected. The **Inherited from** column shows where the attribute is linked in. An attribute can be linked in to one or more Attribute Groups. In the **Attribute Groups** column, you can see to which Attribute Group(s) the attribute belongs.

4. Optionally, if you want to create another Display Sequence at the level that you have selected, right-click the relevant row, click **Create Local link**. You are now able to type a local value in the Display Sequence column.

   Note: Attributes that have no sequence number, and therefore are alphabetically sorted, will be placed last in the list.

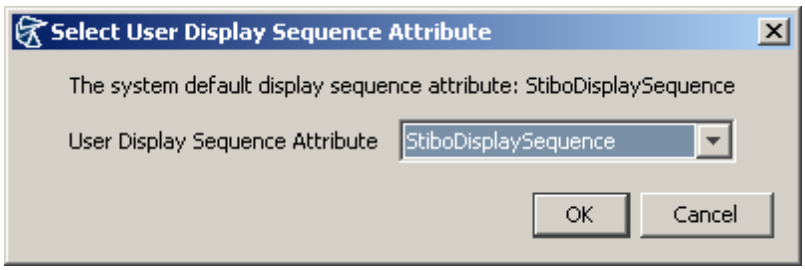
Important: Information about the Display Sequence of attributes is exported in an XML export. If the Display Sequence number is set on the attribute itself, the XML Export will contain the Attributes with Display Sequence attribute value. If the Display Sequence number is set on the Attribute Link, the XML Export will contain the Display Sequence information on the Attribute Links. A combination of the two is possible.

Selecting a Personal Attribute Display Sequence Attribute

If you are interested in sorting attributes in the Product Editor on another Display Sequence attribute then the one set in [System Setup], **Users & Groups**, under **Default Attribute to use as Attribute Display Sequence Attribute**, do the following:

1. On the **View** menu, click **Choose Attribute Display Sequence Attribute**.

   A **Select User Display Sequence Attribute** dialog box appears.



2. If you now select a product folder /product, then select the **References** tab, the first column under **Linked Attributes** will now show your personal Display Sequence attribute.

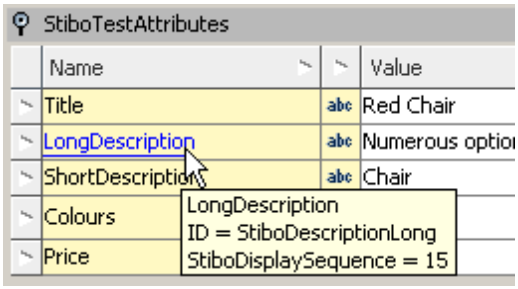
Display Sequence Attribute and the Product Editor Display

In the Product Editor, the attributes are sorted on the Display Sequence attribute, within an Attribute Group. If no Display Sequence attribute is set, the attributes are shown in alphabetical order.

Tooltip

1. Point the mouse to the Attribute name.

   A Tool tip will appear.



2. Information about the Attribute **Name**, **ID** and the Display Sequence attribute is shown.

Note: If you select two or more products, the **Products** Editor shows the attributes always in alphabetical order. The reason for this is that you can select two or more products, that have the same attribute, but with different values for the Display Sequence attribute.

Display Sequence Attribute & Classifications

Your system can be set up to display the Display Sequence attribute at Classification level.

1. In [Classification], select a folder, then click the **References** tab.

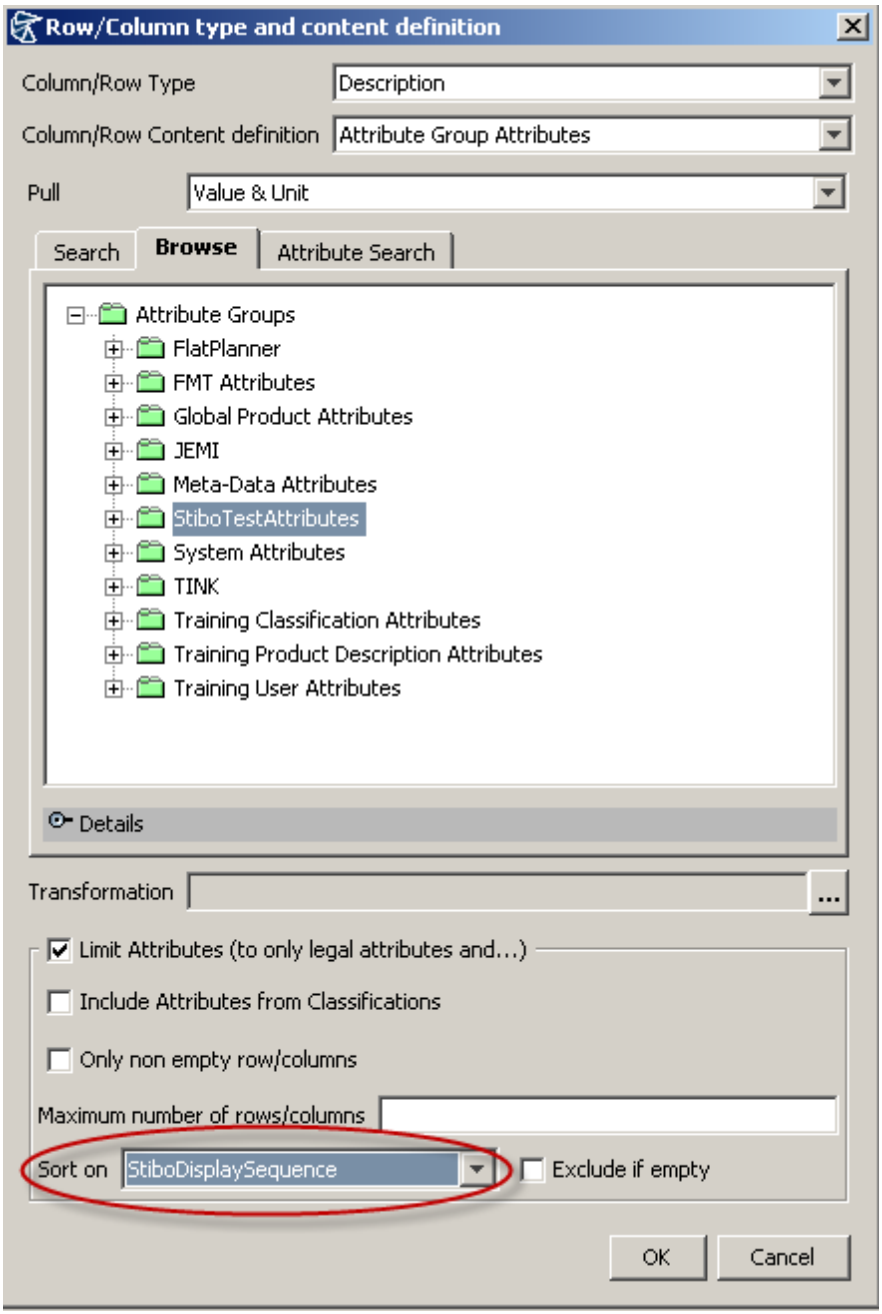
   Under **Attributes** a Display Sequence attribute column is shown.

Important: It is of no use though to maintain the Attribute Display sequence here, as it will have no effect anywhere, not in the Classification editor, nor in an export.

Display Sequence Attribute & Tables

You can use the Display Sequence attribute to sort attributes within Attribute Groups in Tables. For more information, see the STEP PIM Table Component User Guide.

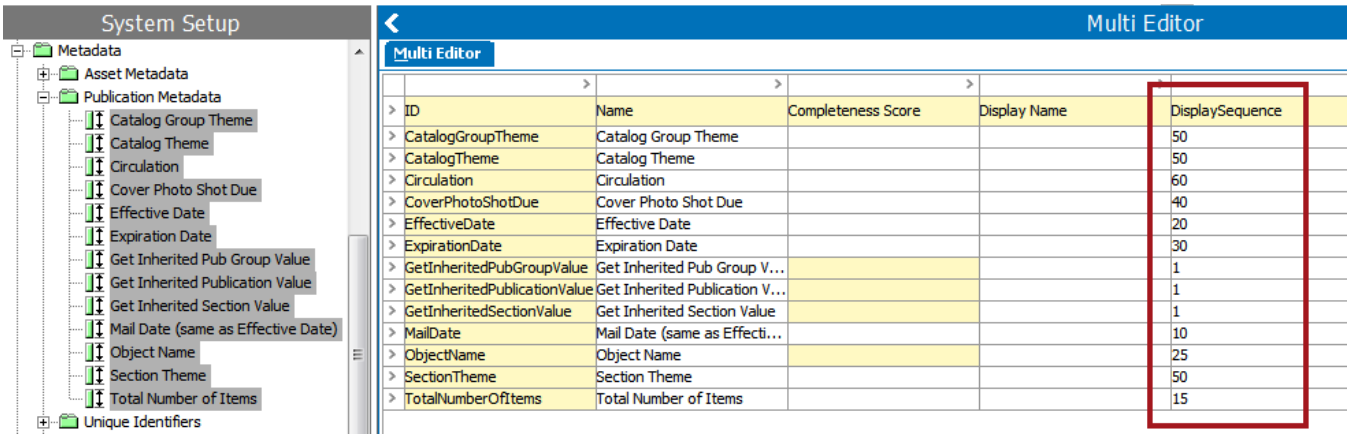
Below you find a screen shot, showing where you select the Display Sequence attribute that will determine the order in which the attributes within an Attribute Group will show in a table.



Setting the Attribute Display Sequence on Publication (Green) Hierarchy Objects

Metadata (Description) attributes on publication hierarchy objects (publication groups, publications, sections, and planned pages) may also be sequenced using a display sequence attribute.

Because publication hierarchy objects do not use Reference Types, attribute sequencing cannot be set from a 'Linked Attributes' flipper on the References tab, which is unavailable on publication hierarchy objects. Instead, metadata attribute sequencing must be set up at the attribute level. However, attributes may be multiselected to allow sequencing from a single location.



Best Practice: Attribute Display Sequence

Per default, attributes shown in the Product Editor of a product are shown in alphabetical order. This section describes how you can set up the STEP Workbench, so that the attributes shown in the Product Editor are displayed according to the order set in the Display Sequence attribute.

Setup Display Sequence Attribute

A Display Sequence Attribute has to meet the following conditions:

- The attribute has to be a **Description** attribute
- The attribute has to be attribute with Validation Base Type: **number** or **integer**
- The attribute has to be valid for the Link Type**Product attribute validation** and/or valid for the Object Type **Attribute**
- It has to be selected in System Setup, under Users & Groups, in the System Settings tab, under Product Information Manager Default Settings, **Default Attribute to use as Attribute Display Sequence Attribute**

View Menu: Choose Attribute Display Sequence Attribute

By default, users can select their own Display Sequence attribute.

If you want to disable this possibility, do the following:

1. In System Setup, unfold **Users & Groups**, then select the relevant User Group.
2. Select the **GUI Set-Up** tab.

Per default, **the Choose Attribute Display Sequence Attribute** radio button is set to ON, turn it OFF.

For more information on how to use the Display Sequence Attribute, see the STEP User Guide.

Handling Duplicated Attributes in Web UI

It is common for attributes to be linked to multiple attribute groups within STEP. This can lead to redundancy and confusion as attributes may appear multiple times on a Web UI screen when configured to display attribute groups that contain overlapping attributes. To manage this, it is possible to indicate groups of attributes that should not be displayed within the Web UI.

Even if duplication is not an issue, a single attribute group or multiple groups may contain sensitive information that should be hidden from view on the Web UI screen. A blacklisted attribute group can be used to hide this sensitive information.

To illustrate this functionality, consider attribute groups X, Y, and Blacklist. The 3 groups contain a variety of attributes, but each includes Attribute A.

If adding the Group Blacklist to the BlacklistedAttributeGroup for Group Y but not for Group X, then Attribute A will display with Group X attributes on the screen, while being omitted from Group Y attributes. This will eliminate duplicates being shown when displaying both groups on the screen.

When Group Blacklist is added as the BlacklistedAttributeGroup for both Group X and Group Y, then neither group will show Attribute A. Alternatively, if nothing is chosen as a BlacklistedAttributeGroup for either Group X or Group Y, then both groups will show Attribute A.

Create Attribute Group(s) to Hold Attributes Known to Exist in Multiples Groups

In the STEP Workbench, one or more attribute group(s) should be created to specifically hold attributes that should not be displayed within the Web UI.

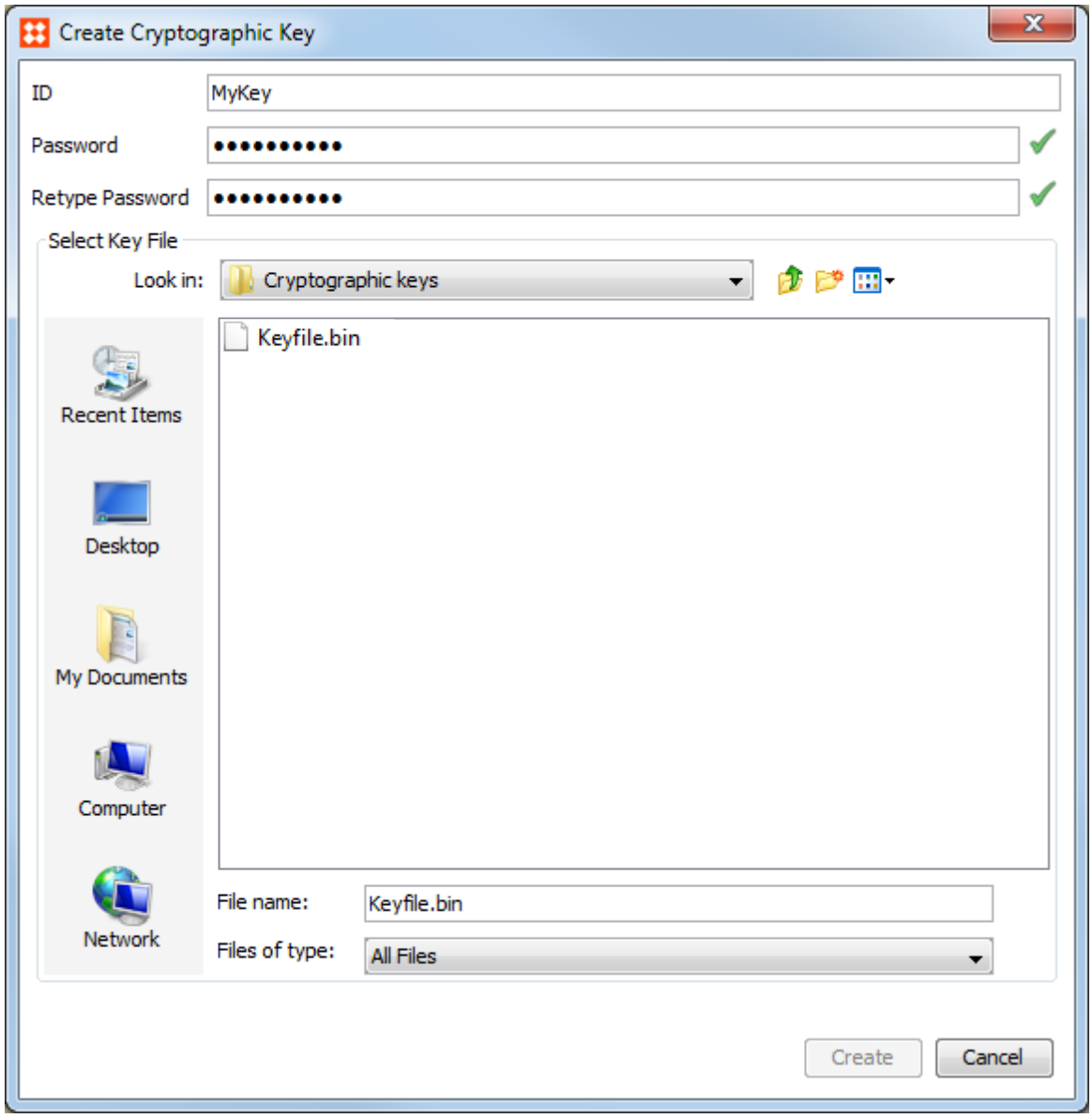
More information about attributes and creating attribute groups is found under System Setup in STEP Help.

Encrypted Attributes

Encrypted attributes require a password to view or edit the values of the attribute. Before creating an encrypted attribute, create a cryptographic key.

Cryptographic Key

Create a cryptographic key in **System Setup** on the **Users and Groups** node. When creating a new cryptographic key in STEP, a 32 byte file with the actual key is required. Keep this key file in a safe place, possibly encrypted as well, where it cannot be accessed by intruders. In addition to the key file, an ID and a password must be supplied.



The ID is shown to users when prompting for the password. Therefore, the ID should be human-readable. The password is required each time attributes using this key are encrypted or decrypted.

The password can be changed when a cryptographic key has been created. After changing the password, all attributes using the cryptographic key use the new password, but the data values are not modified.

Password requirements

The minimum requirements for passwords are 8 characters containing at least three of these four groups:

- Lower case characters a-z
- Upper case characters A-Z

- Digits 0-9
- Symbols

The setup action 'Maintain cryptographic keys' is required to create or delete cryptographic keys, and to change their passwords.

Creating Encrypted Attributes

When creating a new attribute, it is possible to select an encryption key.

Note: The cryptographic key cannot be changed or set after creation.

Properties of attributes that cannot be used with encryption

- LOV validation - since an LOV can be shared between multiple attributes, it cannot be encrypted.
- Full Text Indexable - because the value is stored encrypted, this would represent a security risk.
- Calculated - a calculated value is usually not stored as a value, but derived from other data.
- The validation rules cannot be changed after creation, except multi-valued if the data allows it.

Accessing encrypted data

An encrypted attribute will not initially show any value in STEP Workbench or Web UI. Instead, a **Decrypt** button displays in its place. Clicking this button prompts the user for the cryptographic key file password for the key file used on the attribute.

When the correct password is entered, the attribute looks like any other attribute, showing its value. In this state, it is also possible to edit the value. The client remembers the password for the key file for a limited time, which allows editing multiple attributes without having to re-enter the password.

Each time a user decrypts an encrypted attribute, it is logged in the system log.

Restrictions of Encrypted Data

- Encrypted data cannot be recovered if the key or password is lost. Keep a backup of the original key file used when creating the key in STEP.
- Keys cannot be exported from or imported into STEP.
- Encrypted attributes cannot be exported.
- Encrypted attributes cannot be profiled in data quality.
- It is not possible to search for values of encrypted attributes.
- It is not possible to add / remove encryption to an attribute after it has been created.
- Background processes cannot access encrypted values.

Important: When using encryption in STEP, it is very important that the system is set up to use https, otherwise passwords, values to be encrypted, and decrypted values are transferred in clear text and security is compromised.

About Attribute Links

Attributes are linked to Product and Classification nodes in the **Product** tab and **Classification** tab of the Editors.

Attribute linking makes a selection of Attributes available to objects that are linked to (or below) the specified node in the Classification or Product hierarchy.

Before linking Attributes, the following issues must be considered:

- Where to link an Attribute:

  Before linking Attributes, it is very important to consider the scope of these Attributes, i.e. for which objects are they supposed to be relevant.

  A best practice is to organize the Product hierarchy or the Classification hierarchy so similar Products, that share the same characteristics, are grouped together. These Products will then be able to share a set of Attributes by inheritance from the hierarchy.

- Which hierarchy levels to link to:

  If you have a hierarchy where an upper level node has more sub nodes that all share the same characteristics, the Attributes should be linked at that upper node level. The Attributes will then be inherited to the sub nodes.

  If, however, the Attributes only are relevant to a sub node, they should be linked there. Read about Attribute inheritance below.

- Which Product object types to link to:

  It may also be important to specify which type of Product the attribute is describing.

  For example, Attributes such as “Price” and “Order Code” will only be relevant to Products with object type “Article”. This means that e.g. Pro-ducts with object type “Product Family” cannot have a “Price” or an “Order Code”.

  On the other hand, an Attribute such as “Long Description” may be relevant to a “Product Family”, but not to an “Article”.

- Attribute inheritance:

  An Attribute that is linked to a node in the hierarchy is inherited to the nodes below. This way, Attributes are shared among nodes.

  An example, a Specification Attribute called “BladeWidth” might be linked at the “Saws” node (below the” Power Tools” node). Therefore, the Attribute “BladeWidth” is available to both the sub nodes “Circular Saws” and “Table Saws”, by inheritance.

  This inheritance feature is useful when there are globally used Attributes – these are Attributes that are ”common” to all Products. For example, “Warranty”, “Price”, and “CountryofManufacturer” would most likely be global Attributes, since each and every Product would have a value for these characteristics. Therefore, these Attributes would be linked at the top node in the hierarchy, and in doing so, be inherited to all of the nodes below. See the figure in Online Help - About Linking Attributes.

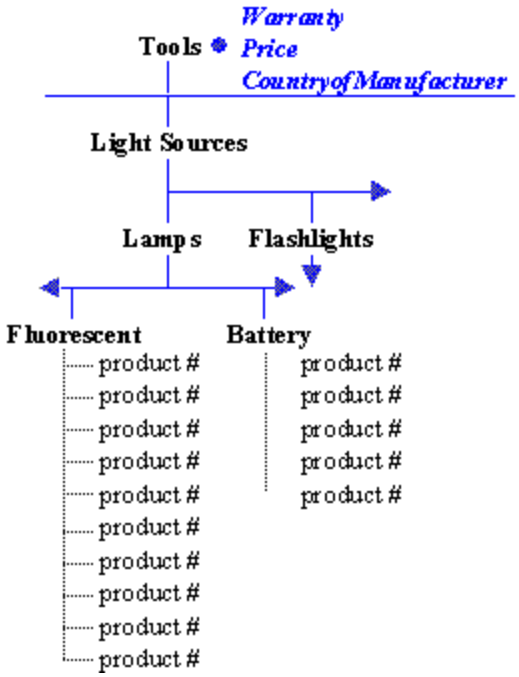


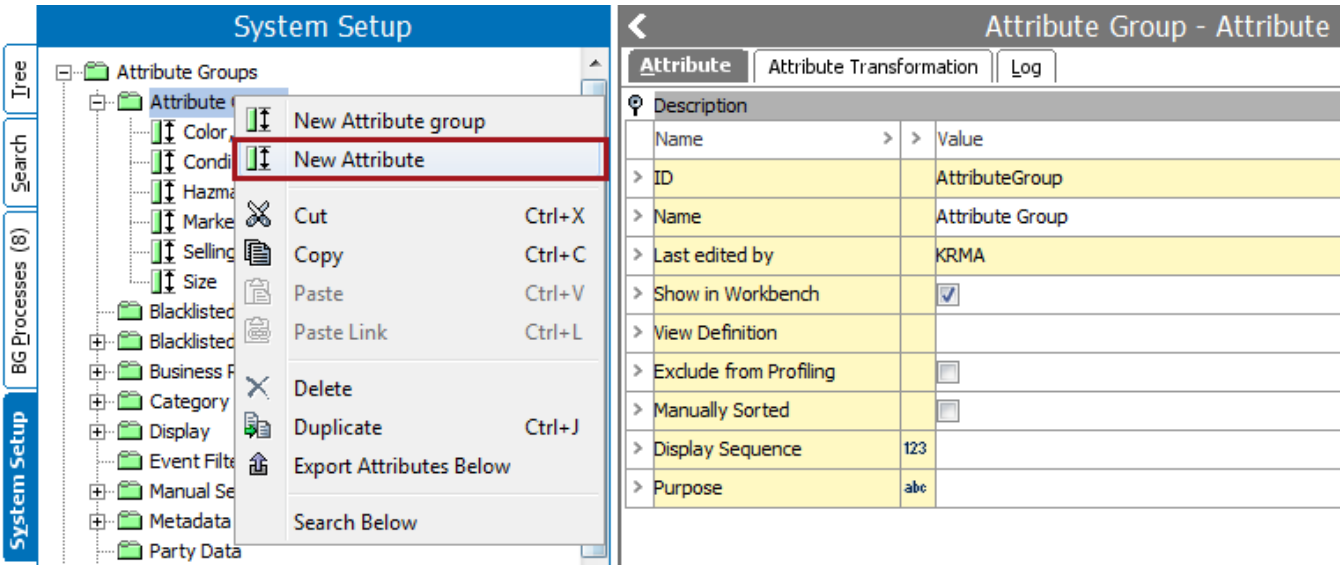
Figure 1 above: A part of a product hierarchy showing the principle of the inheritance of product attributes.

Creating Attributes

Attributes are maintained in [System Setup]: **Attributes Groups**.

1. In [System Setup], expand **Attribute Groups**, and then select a Group for the Attribute.

The new Attribute belongs under this Group.



2. In the **Object** menu, navigate to **Create** and then click **Attribute**, or right-click and click **New Attribute**.

A **Create Attribute** wizard appears.

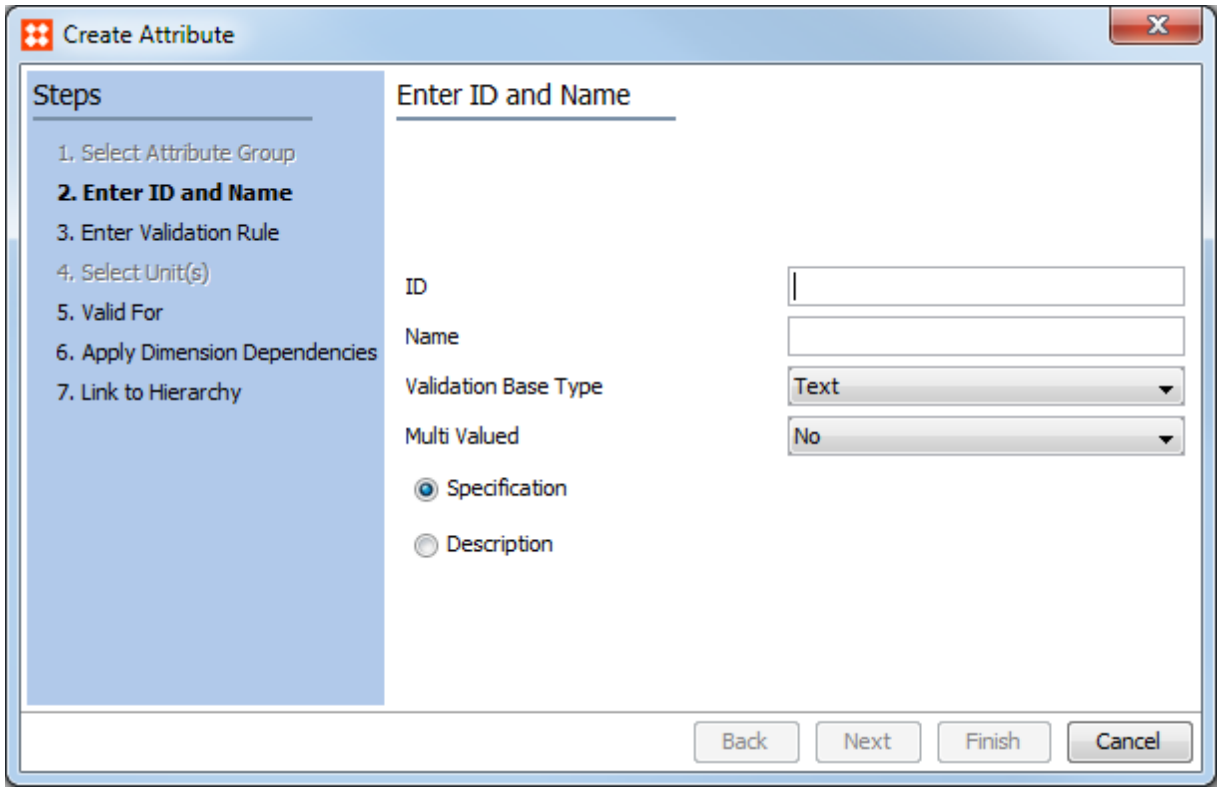
3. Type an ID and Name.

Select the sort of validation that the Attribute should be based on from the list.

Select if the Attribute should hold multiple values on each product or only one value per product.

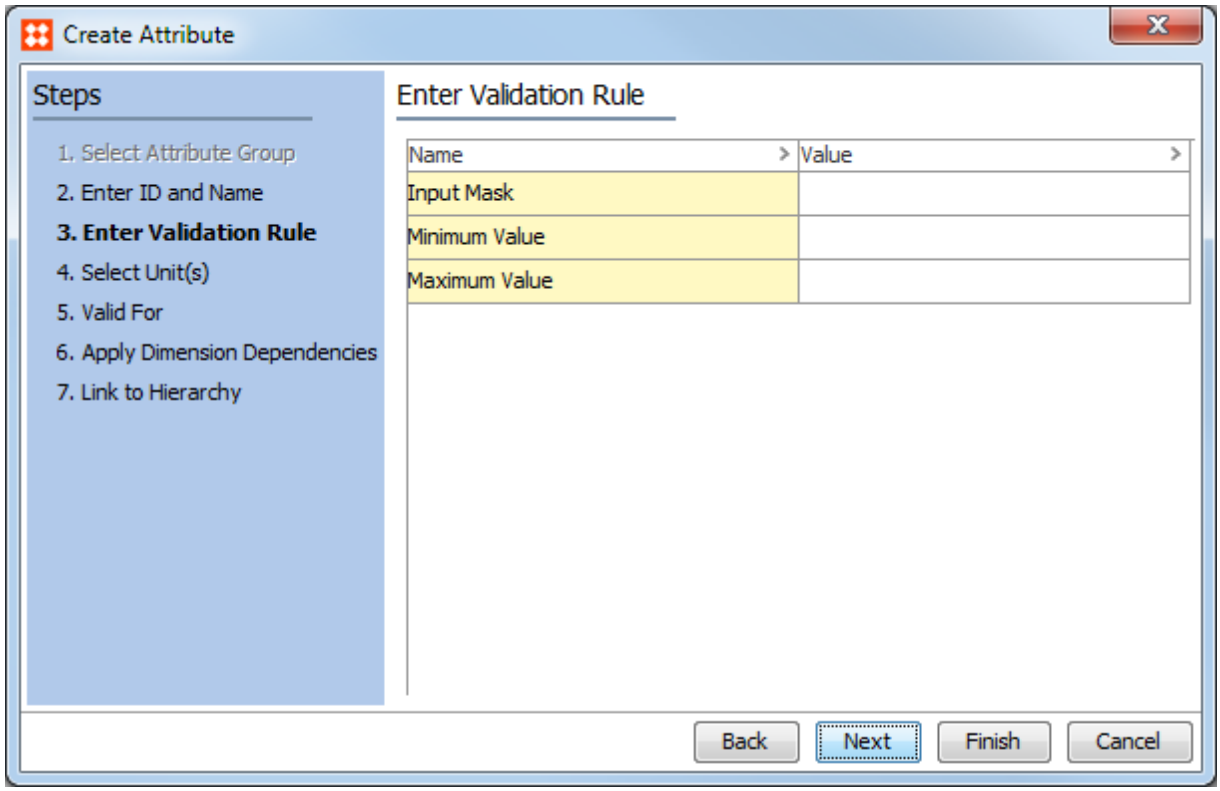
Determine whether it is a Specification or Description Attribute.

Click **Next**.



4. Type an Input Mask, Minimum Value and/or Maximum Value.

   Click **Next**.

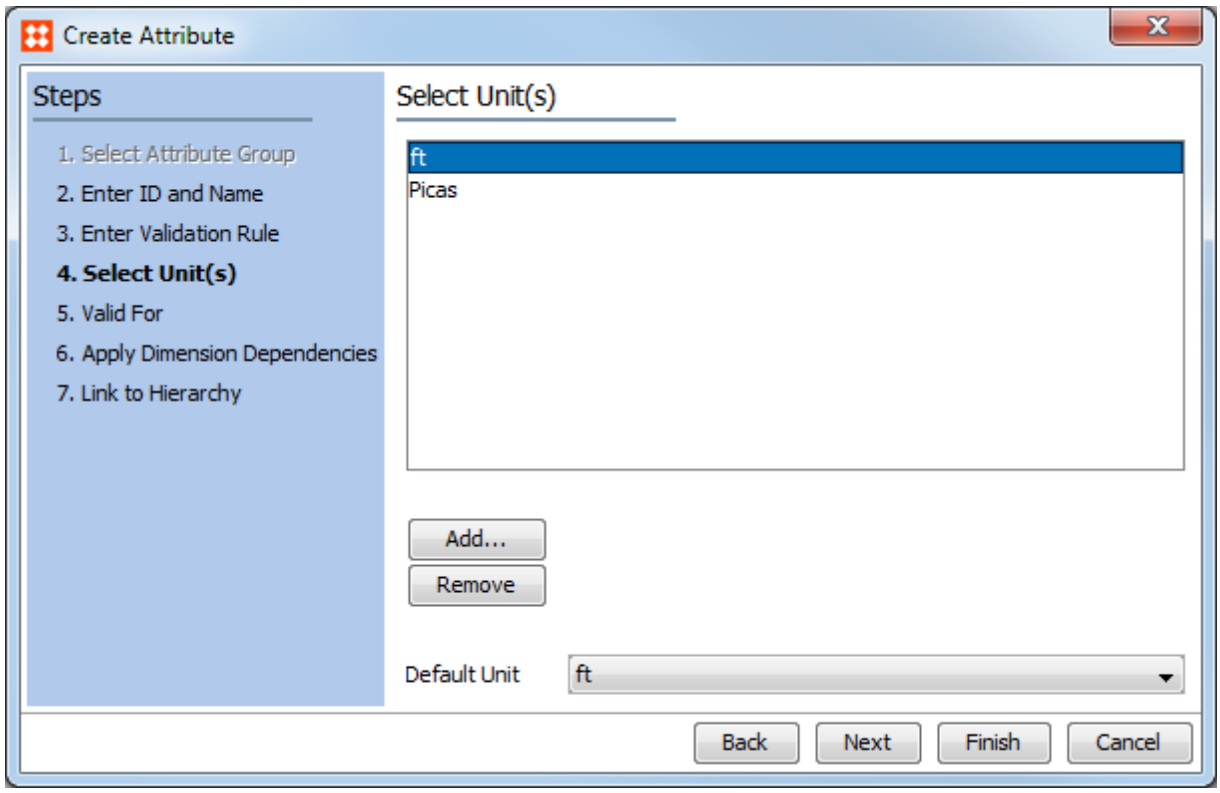


5. The **Select Units** step will only be visible if a numeric validation base type has been selected.

Search or browse to add one or more relevant Units or Unit Groups.

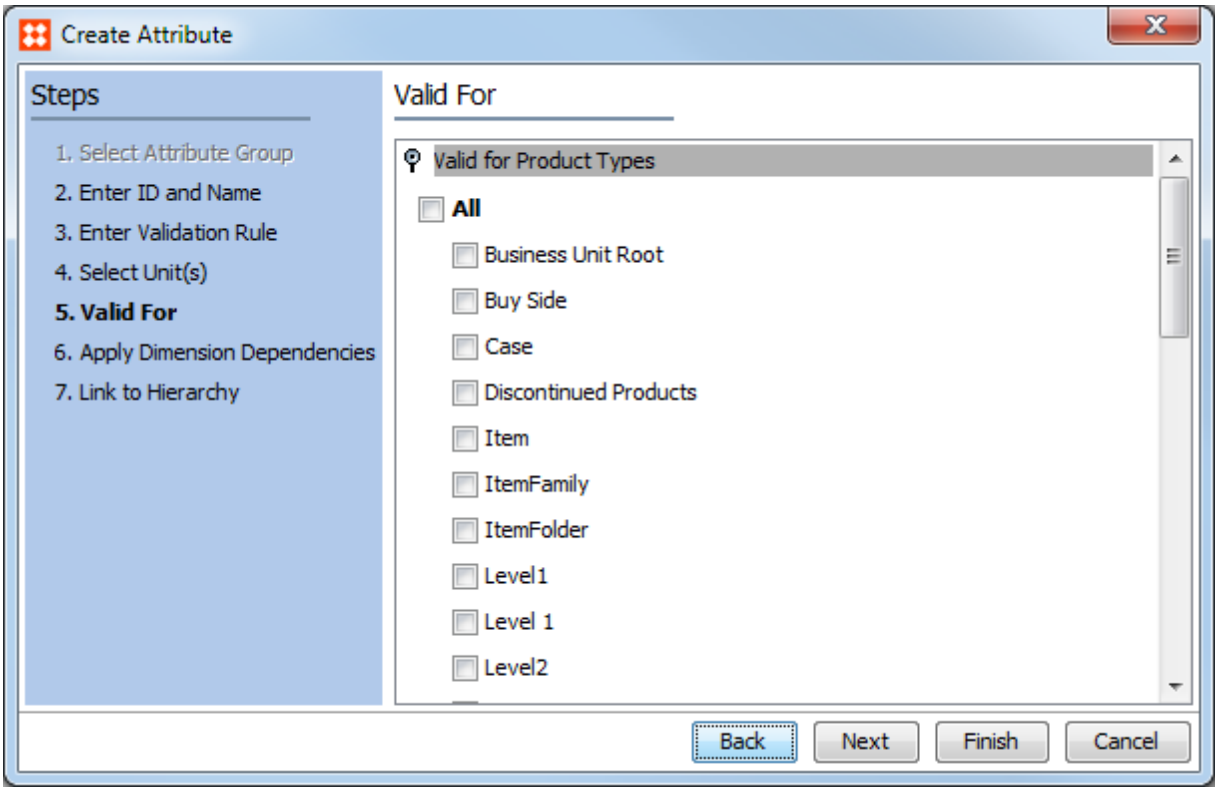
Select a unit to be selected as default unit.

Click **Next**.



6. Select the product types the attribute is associated with.

Click **Next**.

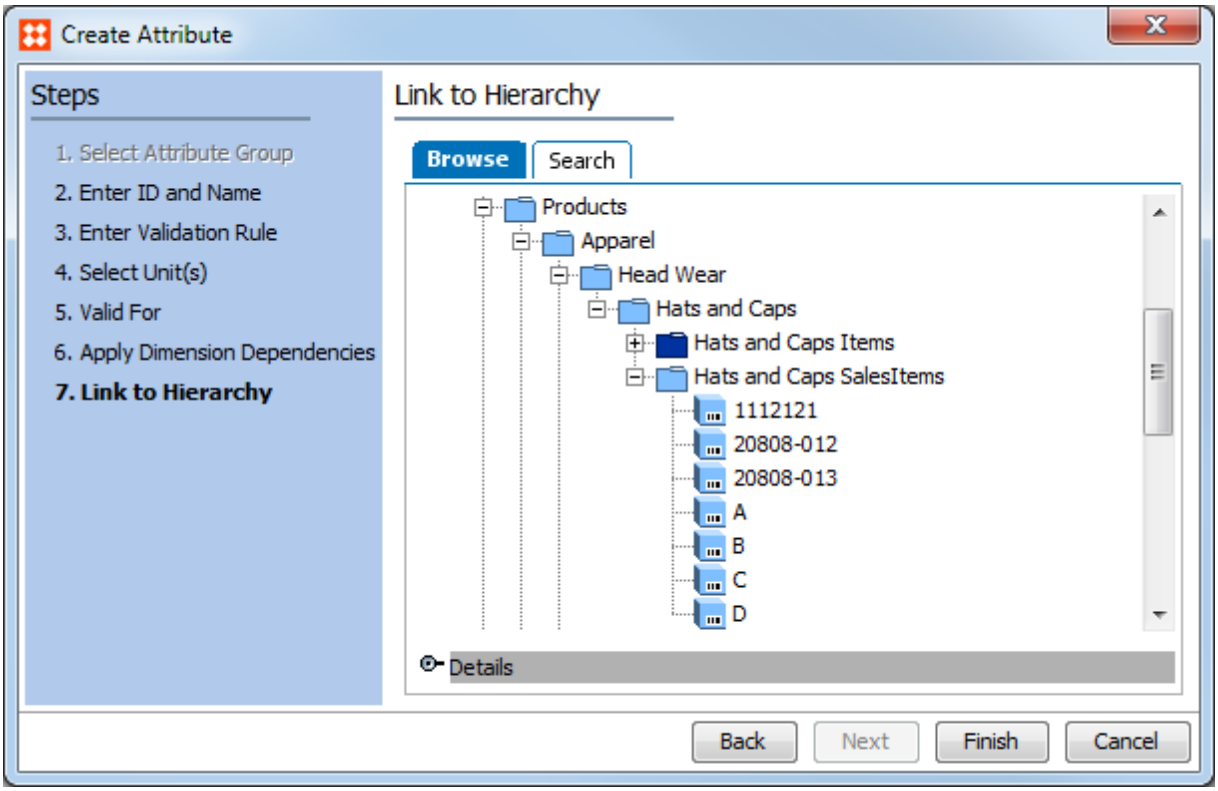


7. Apply Dimension Dependencies.

Click **Next**.

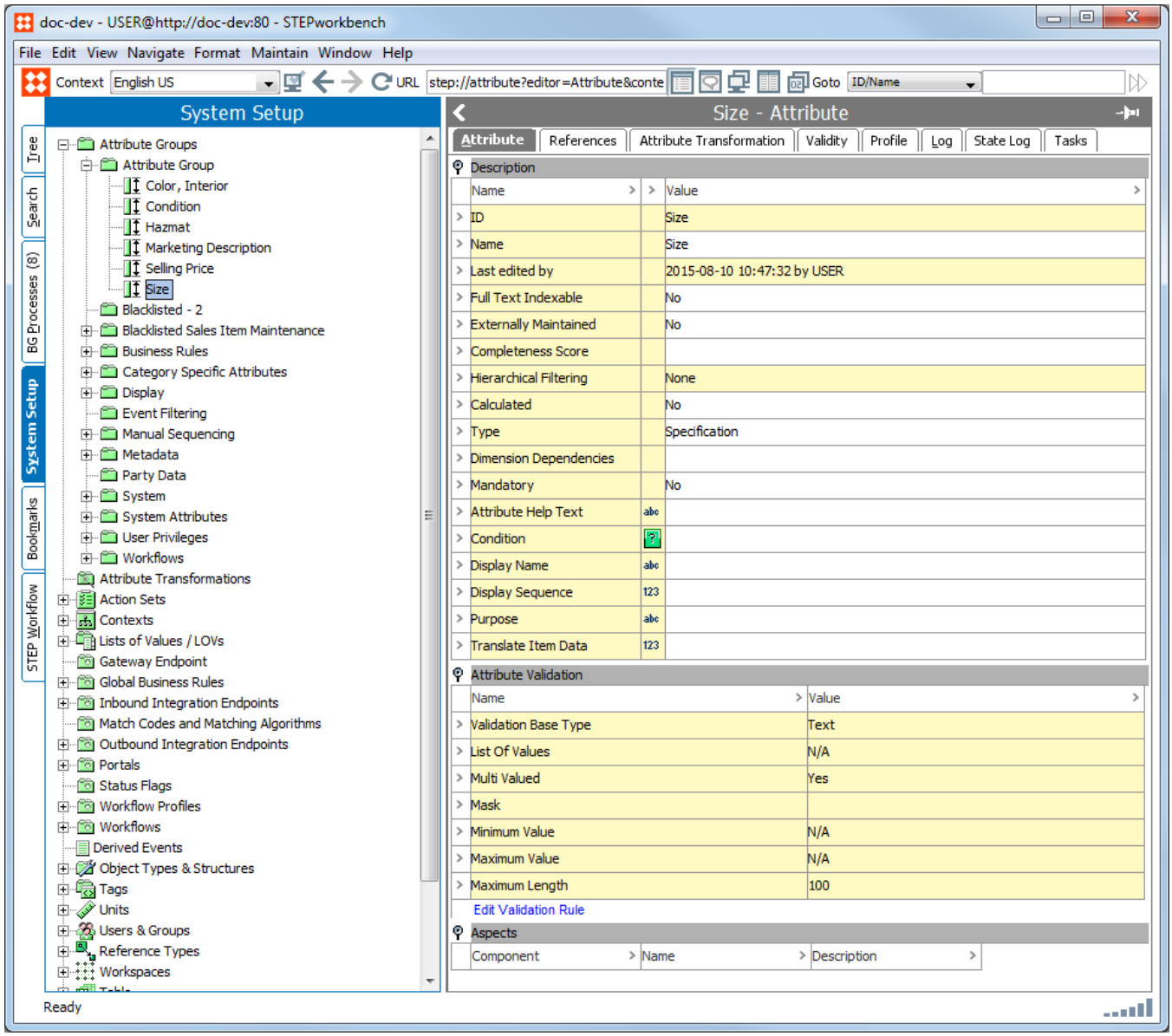
8. Select a tree node where the attribute should be linked.

Note: If the Attribute should be a Description Attribute or a Link Attribute, it is not necessary to link the Attribute to a Product or a Classification.



Click **Finish**.

The Attribute has now been created.



Note: Attributes are created as Specification Attributes by default. Specification Attributes are only valid for Product Object Types. If an attribute should be valid for e.g. Classification Object Types, it must be changed into a Description Attribute.

The next steps in Attribute creation could be to choose Description type for the attribute, eventually specify Externally Maintained, Calculated, and/or specify the Validity.

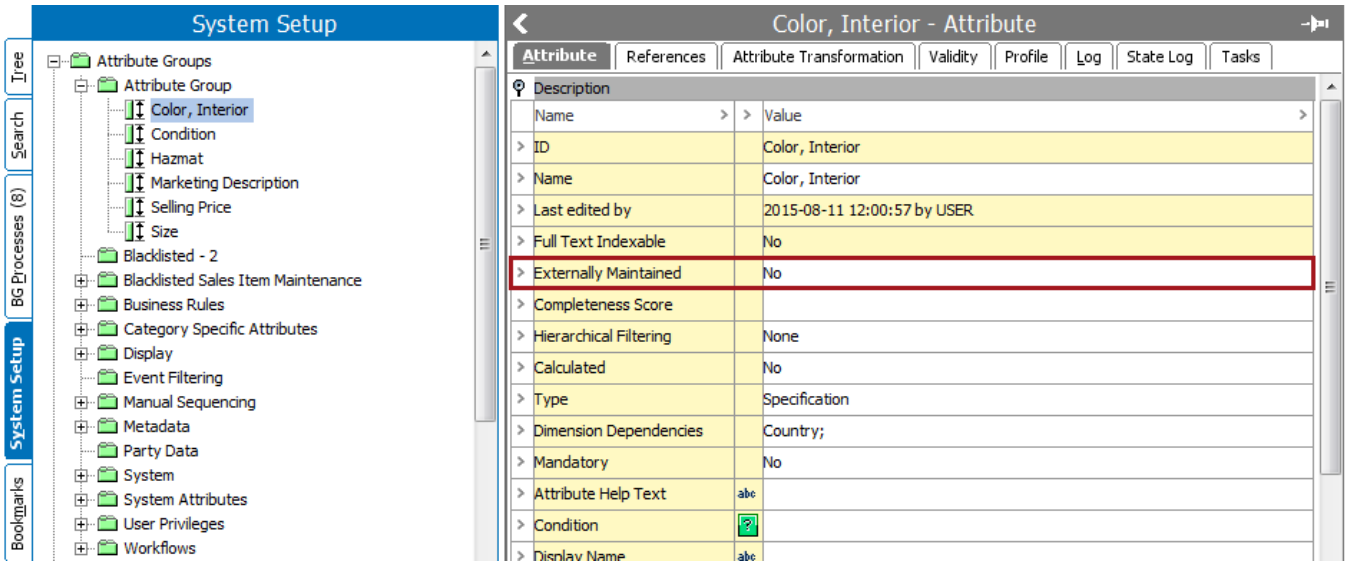
Externally Maintained Attributes

Setting an Attribute as externally maintained is done in the **Attribute** tab of **Attribute Groups** in System Setup.

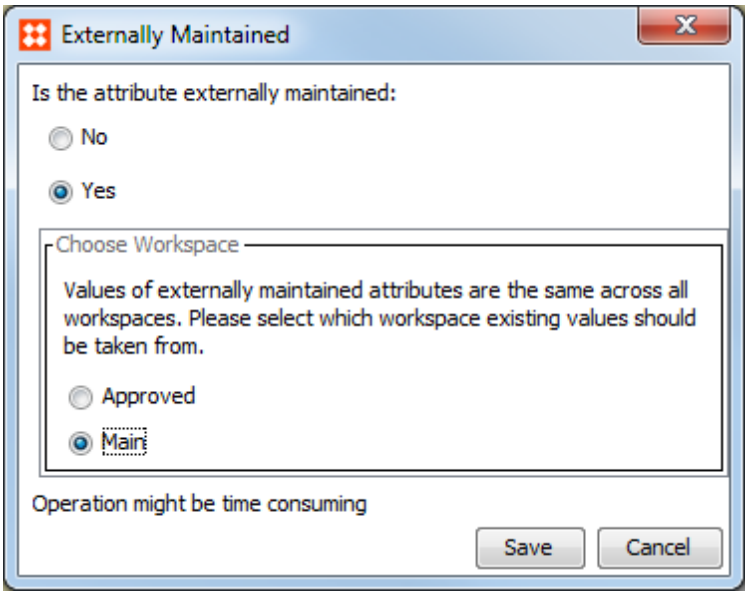
An externally maintained Attribute 'lives' outside of STEP. The Values are external systems. The typical example are the prices that are maintained in the ERP environment and transferred to STEP for publishing and distribution.

Item	Description
Revision control	Externally maintained Attributes are not included in the Revision control. This means that when a value is imported to one of the workspaces, the new value will automatically be applied to all other workspaces. For example, if a Supplier enters a value in the 'Staging' workspace for an externally maintained Attribute, this value will automatically be transferred to the other workspaces, e.g., 'Main' and 'Approved'. If an already created Attribute, that has values, should be set as externally maintained, you must decide which workspace these values should be taken from. This should typically be from the workspace that contains the most recent updates, e.g., 'Main'.
Sharing LOVs	Externally maintained Attributes cannot share LOVs with internally maintained Attributes, as values cannot be maintained both internally and externally. Setting as externally maintained: If an LOV that is used by one or more internally maintained Attributes is selected, and the Attribute is then set as externally maintained, all Attributes will become externally maintained. Setting as internally maintained: This works the other way around as well. If an LOV that is used by one or more externally maintained Attributes is selected, and the Attribute is then set as internally maintained, all Attributes will become internally maintained.

1. In System Setup, expand **Attribute Groups**, expand the relevant Group, and then select the Attribute to be set as externally maintained. The **Attribute** Editor appears.



2. Click the **Attribute** tab.
3. Under the **Description** flipper, double-click the **Externally maintained** field. The **Externally Maintained** dialog appears.



4. Click **Yes**. The **Choose Workspace** field will be enabled.
5. Select the workspace that contains the existing values that will be used. An externally maintained attribute will hold the same values across all workspaces.
6. Click **Save**.

Note: If an Attribute has values this may be a very time consuming operation.

Setting Hierarchical Filters

Hierarchical filtering is a way of setting a filter on a LOV that is used on a specific product, product family, or classification and defines which values are valid for a certain product. For more information about LOVs, see the **List of Values (LOVs) Overview** section.

Hierarchical filters enable the use of only one LOV with an assortment of values, while differentiating the values available for different Products.

For example, an LOV contains the following values:

- Green
- Red
- Yellow
- Oak
- Mahogany

The Products (or Product Families or Classifications) are of the type:

- Plastic
- Wood

Adding a filter on the LOV for the Plastic Products limits the available LOV values to:

- Green
- Red
- Yellow

Adding a filter on the LOV for the Wood Products limits the available LOV values to:

- Oak
- Mahogany

Relation between LOV filters

An attribute may have multiple LOV filters, both on the attribute itself and on Attribute Links, in both the Product Hierarchy and Classification Hierarchies. To avoid confusion, it is recommended that no LOV filter can allow values that have been removed by LOV filters above it in the hierarchy. This is also supported when editing LOV filters, which simply do not allow editing restricted values already in the filter.

This also makes it possible to define different LOV filters in the Product Hierarchy and Classification Hierarchies, which are valid for the same products. The effective filtering for values on a product is the intersection of these LOV filters. It is, however, not recommended to combine LOV filters from different hierarchies, as it easily becomes difficult to manage.

Viewing and editing LOV Filters

The LOV Filter Dialog is accessed by clicking the filtering icon on attribute links. It displays the current local LOV Filter on the attribute link, or if no local LOV Filter is set, it shows the effective LOV Filter inherited from the 'parents' (e.g., attribute links in classifications or products, the attribute, or the LOV itself). If the user is privileged, it also allows editing, removing, or adding LOV Filters.

Options affecting Performance

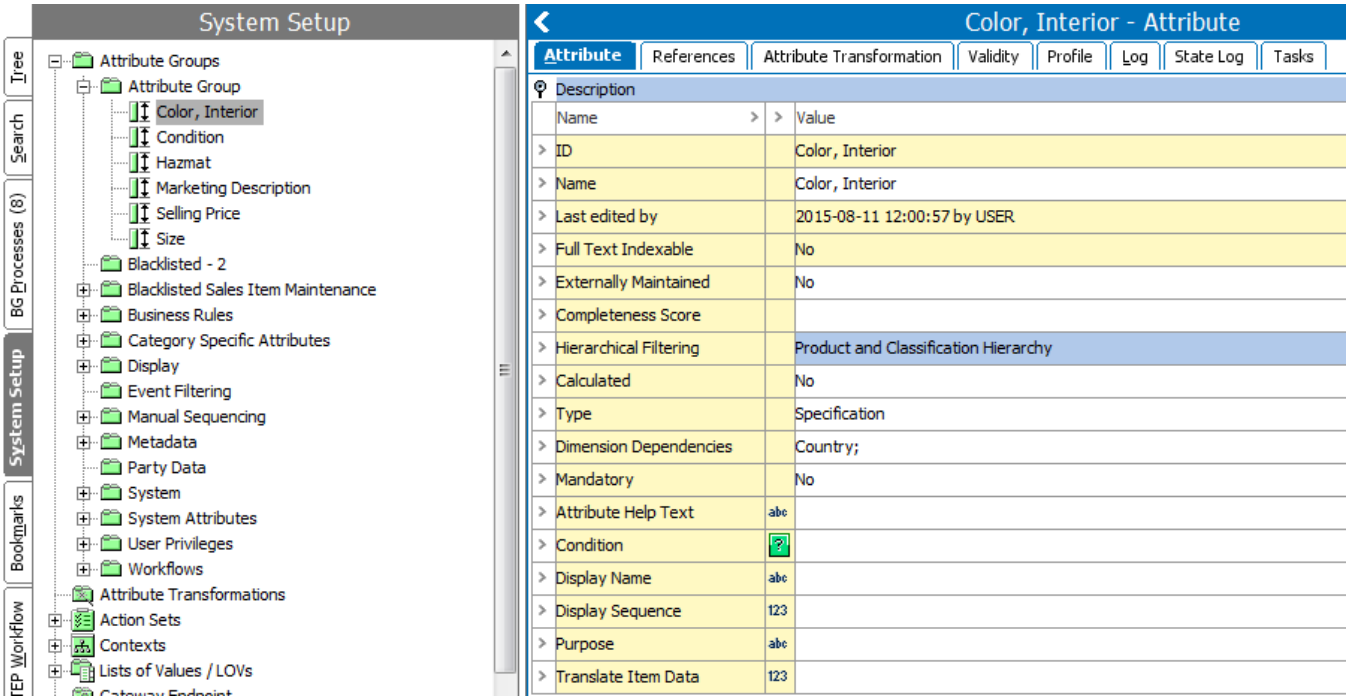
You can define whether an attribute is allowed to use Hierarchical Filtering in both Hierarchies, just the Product Hierarchy, or not at all. As this may affect performance, it is recommended to allow as little as possible.

Additionally, it is also possible to restrict which parts of the Classification Hierarchies are used for LOV filter inheritance, by setting the 'Ignore LOV filters' option on the relevant Product to Classification Links. However, unless absolutely necessary, this is not recommended, and may become harder to manage.

Note: The Smartsheet exporter outputs an Excel file with limited support for LOV filtering, but LOV filters inherited from Classification or Product-Overrides are not supported.

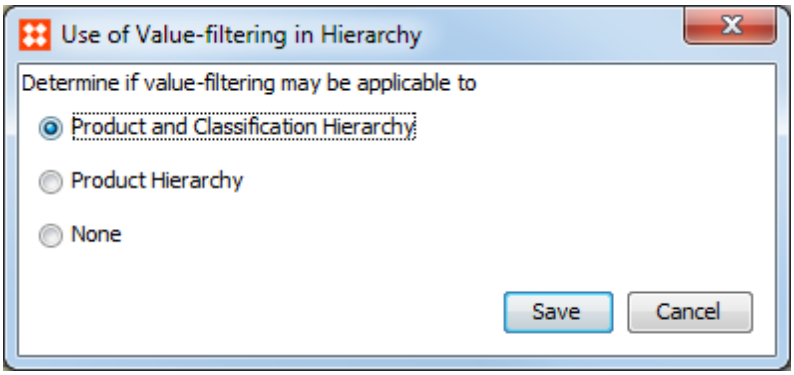
To setup a hierarchical filter:

1. In [System Setup], expand **Attribute Groups**, expand the relevant Group, and then select the Attribute to be set to hierarchical filtering.

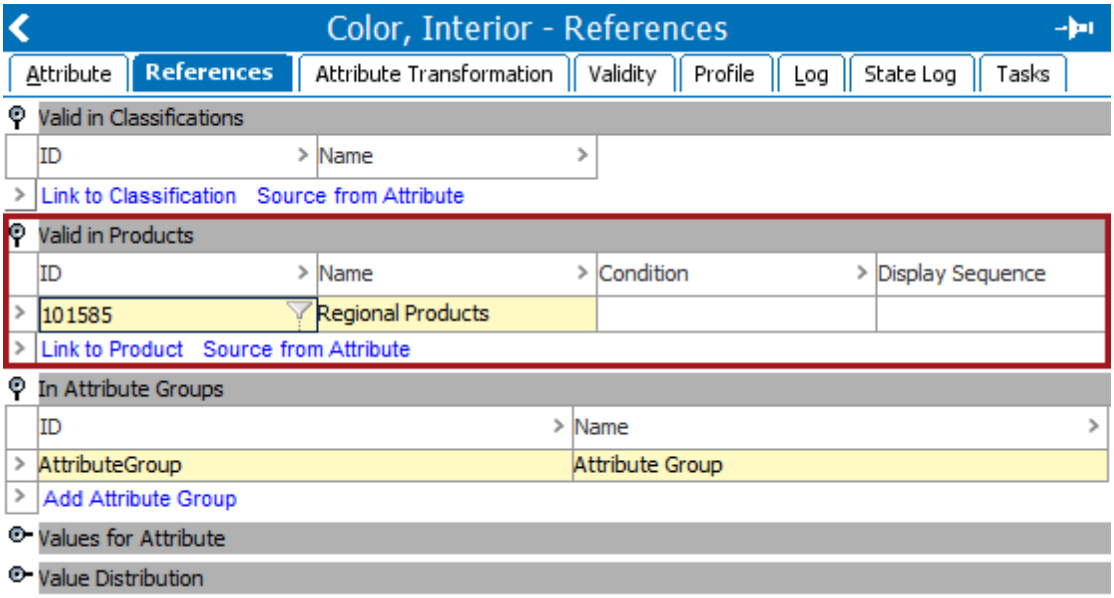


2. Click the **Attribute** tab.
3. Under **Description**, double-click the **Hierarchical Filtering** field.

   The **Use of Value-filtering in Hierarchy** dialog box appears.



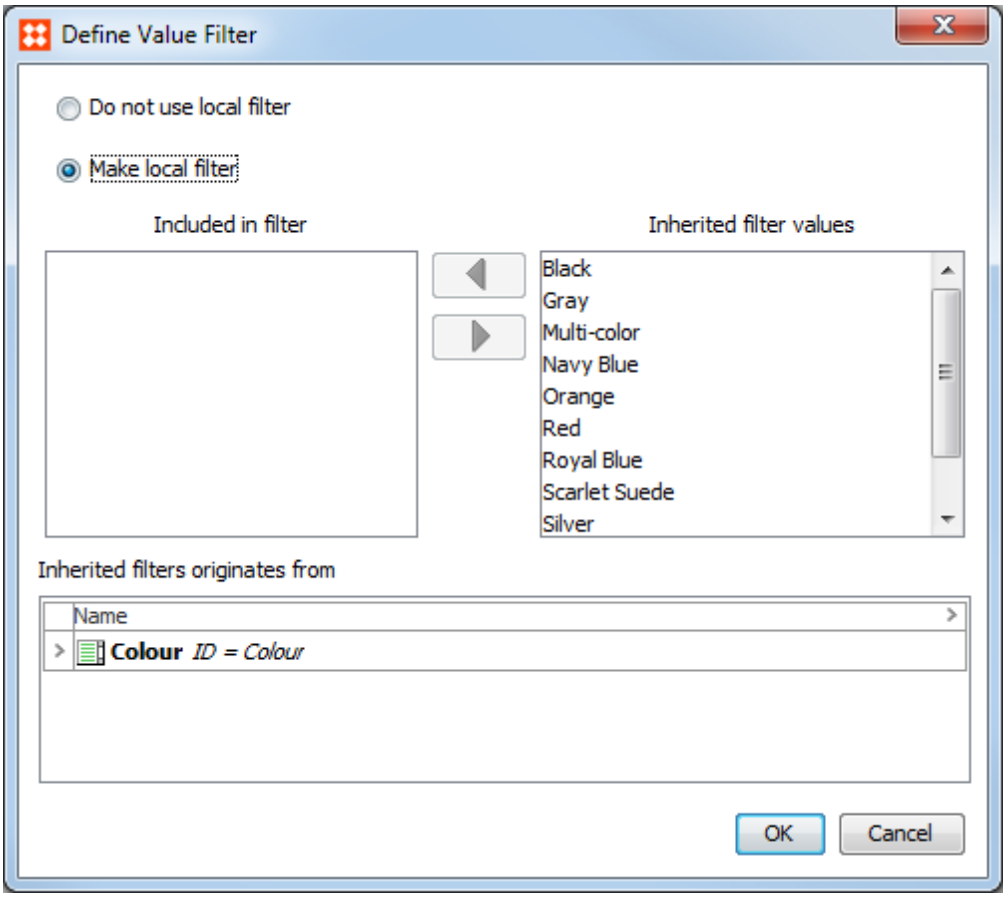
4. Click **Product and Classification Hierarchy** or **Product Hierarchy**, and then click **Save**.
5. Click the **References** tab.



6. Under **Valid in Products**, click the filtering icon in field containing the relevant Product.

Note: To modify filters you must have privileges to maintain Attributes. For more information see About Attribute Links.

A **Define Value Filter** dialog box appears.

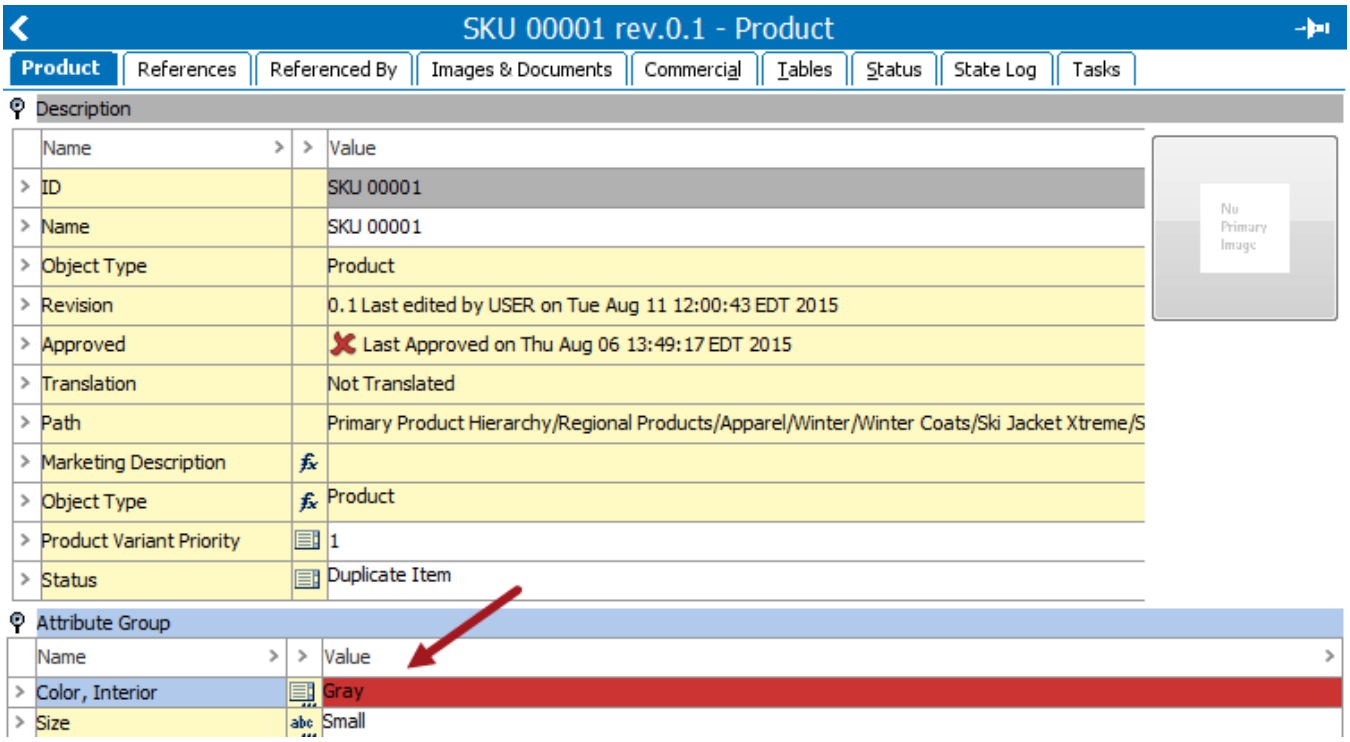


7. Click **Make local filter**, select the values (e.g. by double-clicking on the value) to be included in the filter, and then click **OK**.

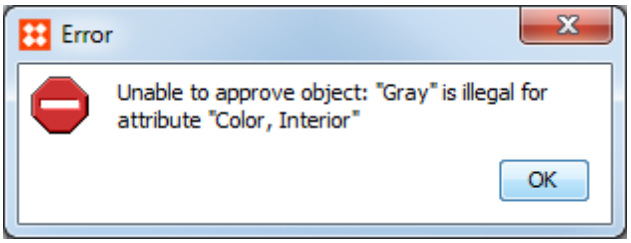
Note: To exclude values from a LOV, double-click on the value under **Included in filter**.

The defined values can now be selected on the Product in the **Product** Editor, under a Specification Attribute or Description Attribute. For more information see the **About Attributes** section.

In the **Product** Editor it is possible to see if the chosen value is legal according to the filter setup. If not legal, the field containing the value is red.



When trying to approve an illegal value, a warning message will appear:



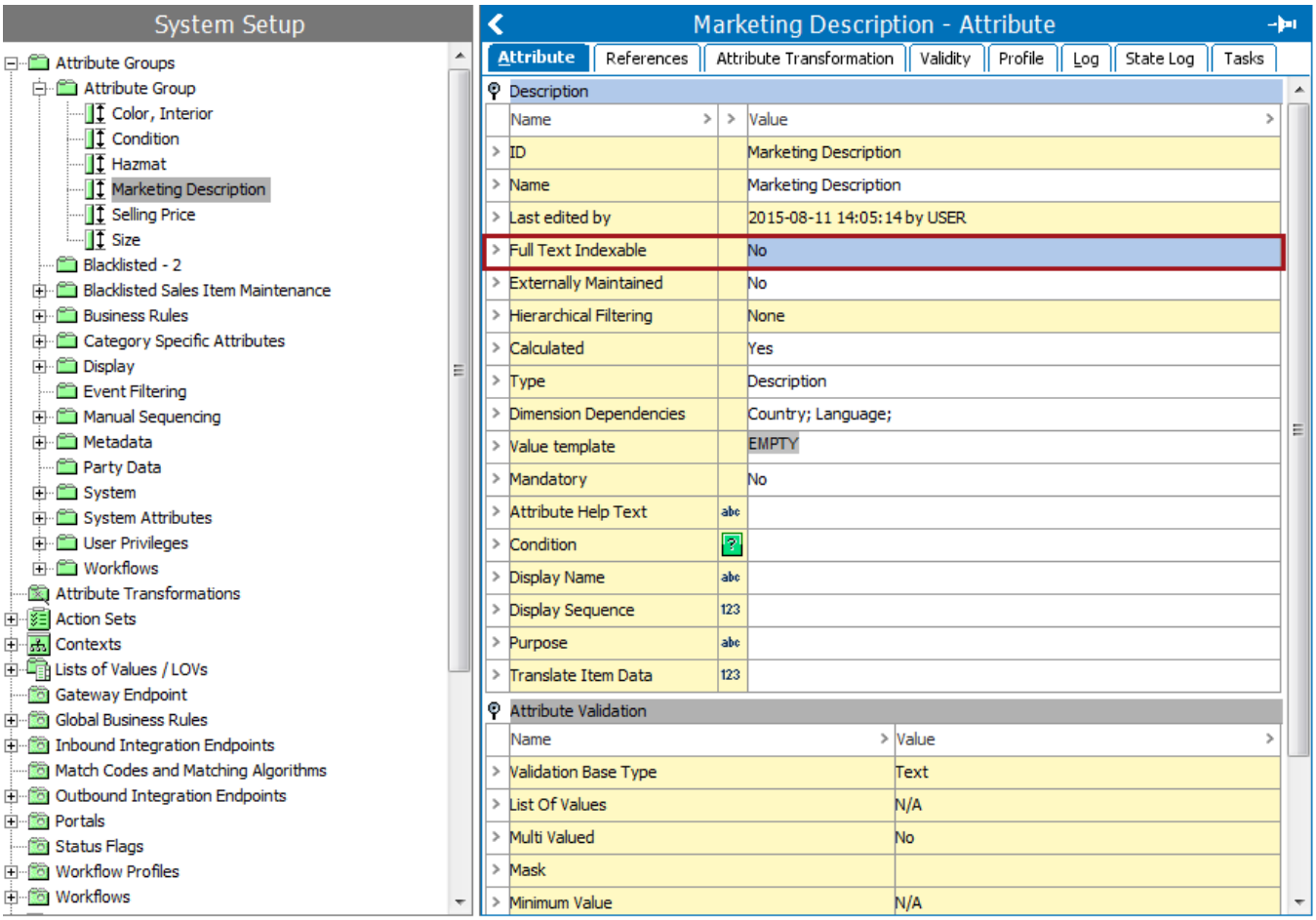
Click **OK**, and then select a legal value.

Setting Attributes as Full Text Indexable

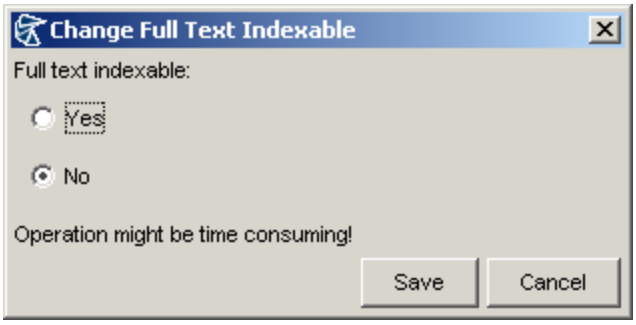
To set an Attribute as full text indexable, navigate to [System Setup], click on **Attribute Groups**, and then the **Attribute** tab.

Full text indexable is a way of indexing the values of an Attribute with validation base type “text” in words. This will enable you to search for words within “text” values.

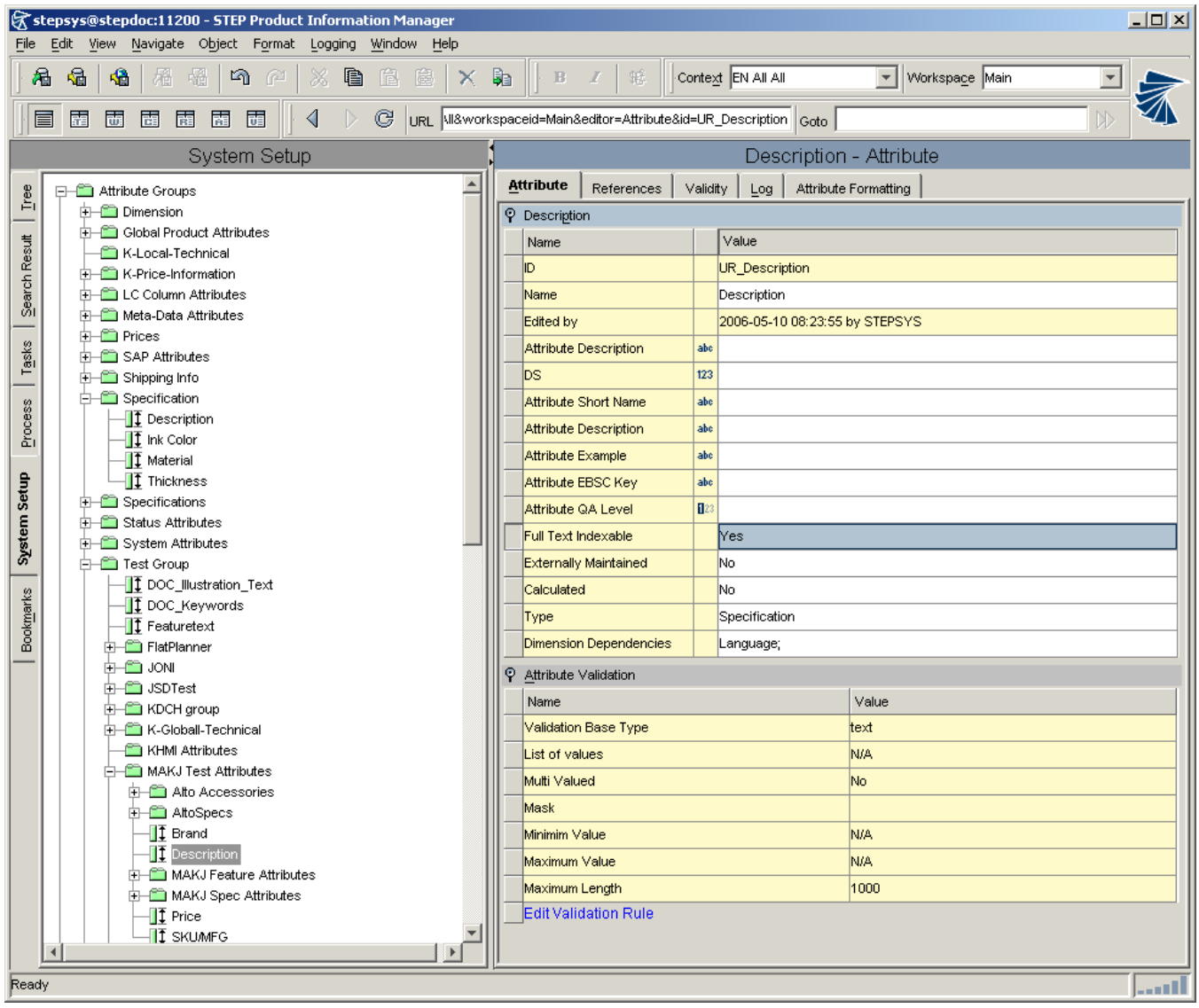
1. In [System Setup], expand **Attribute Groups**, expand the relevant Group, and then select the Attribute to be set as full text indexable.



2. Click the **Attribute** tab.
3. Under **Description**, double-click the **Full text indexable** field. The **Change Full Text Indexable** dialog box appears.



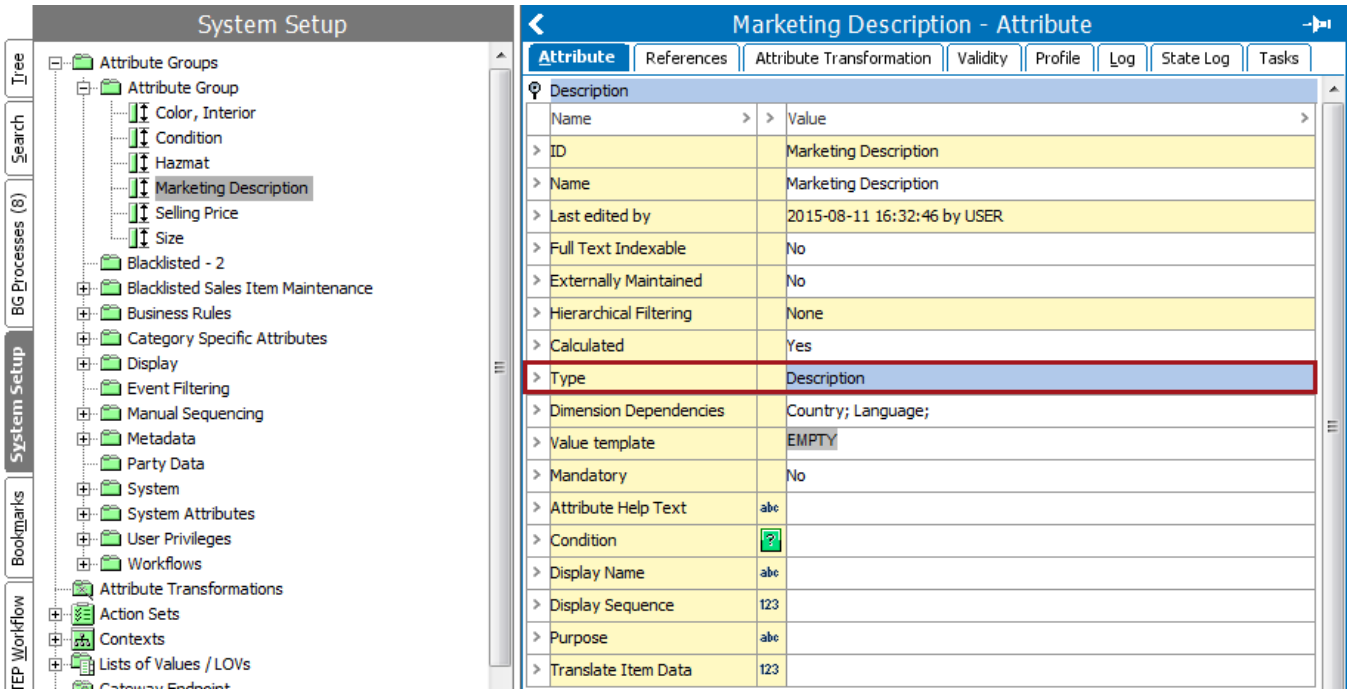
4. Click **Yes**, and then click **Save**.



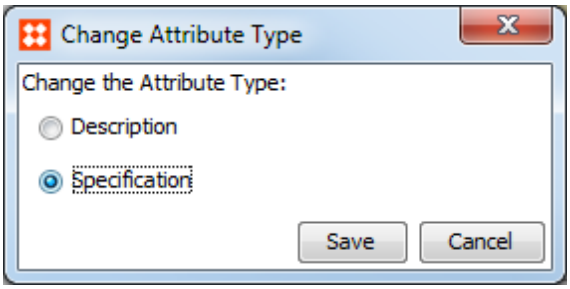
Setting Attribute Types

To set the Attribute Type, navigate to [System Setup], click on **Attribute Groups**, and then the **Attribute** tab.

1. In [System Setup], expand **Attribute Groups**, expand the relevant Group, and then select the relevant Attribute to set the type. The **Attribute** Editor appears.



2. Click the **Attribute** tab.
3. Under **Description**, double-click the **Type** field. The **Change Attribute Type** dialog box appears.



4. Click the relevant type.
5. Click **Save**.

Setting Validity of Specification Attributes

To set the Validity for Specification Attributes, navigate to [System Setup], click on **Attribute Groups**, and then the **Attribute** tab.

Specification Attributes can be made valid for:

- Product Object Types

Setting validity for a Specification Attribute means specifying the Product Object Types that can use the Attribute.

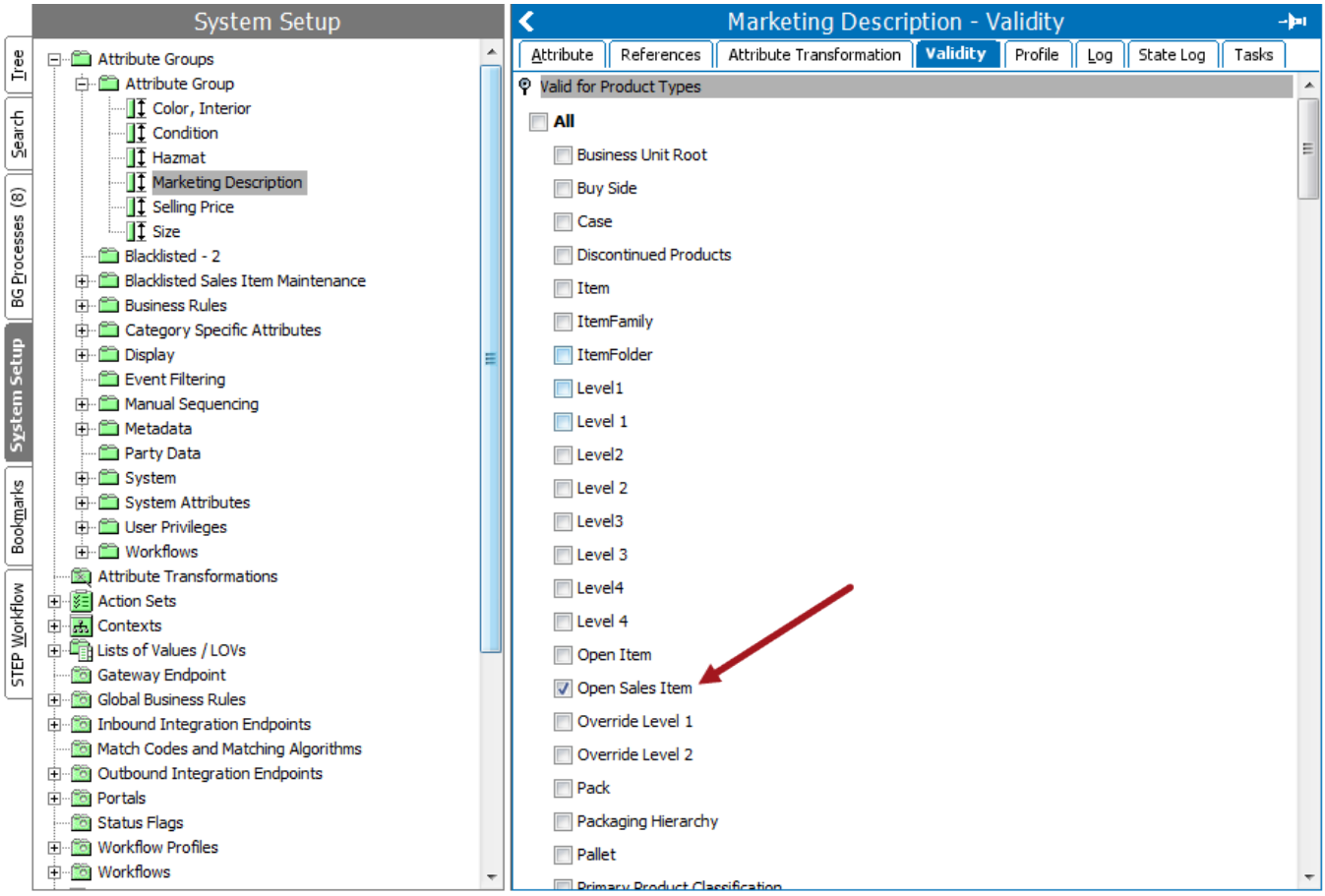
1. In [System Setup], expand **Attribute Groups**, expand the relevant Group, and then select the relevant Attribute.

The **Attribute** Editor appears.

2. Click the **Validity** tab.

The available Product object types appear.

Note: The number of available Object types depends on whether the Attribute is a Specification or Description Attribute.



3. Specify which Product Object types the Attribute should be valid for by clicking the check boxes.

Note: The Attribute will now be available for the selected Product object type(s).

Note: Even if an Attribute is not valid for a certain Object type, it is possible to link it.

Setting Validity of Description Attributes

To set the Validity for Description Attributes, navigate to [System Setup], click on **Attribute Groups**, and then the **Attribute** tab.

Setting validity for a Description Attribute means that it will appear automatically in the **Description** field of all objects of this object type.

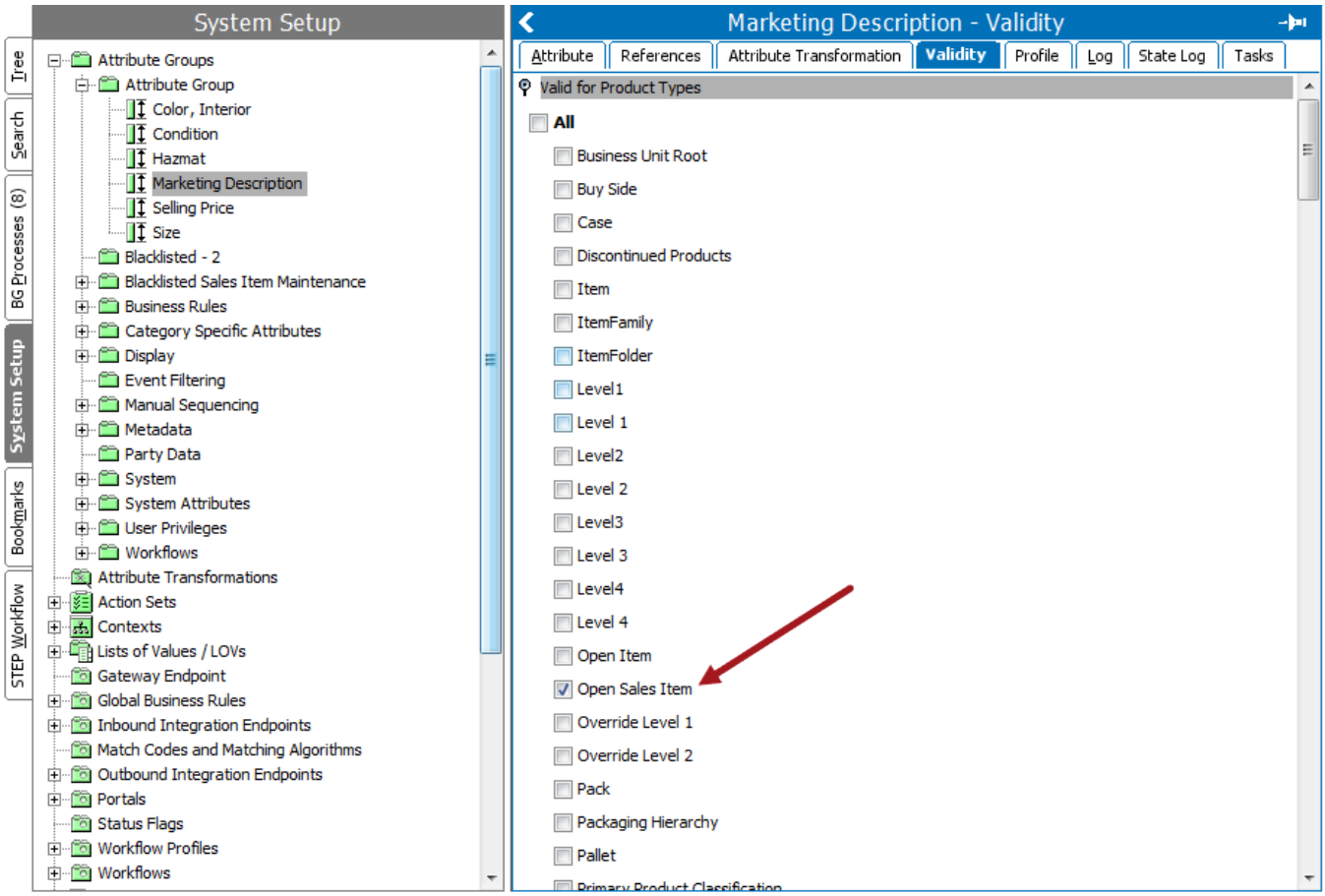
Object type	Description
Product object types	The Attribute will be valid for the selected Product object types, and will appear automatically in the [Description] field. Note: As default, the Attribute is set to be valid for all Product object types.
Classification object types	The Attribute will be valid for the selected Classification object types, and will appear automatically in the [Description] field.
Asset object types	The Attribute will be valid for the selected Asset object types (Images & Documents), and will appear automatically in the [Description] field.
Link types	The attribute will be valid for the selected Link types, and will appear automatically in the Reference Editors in Product, Classification and/or Asset Editors.

1. In [System Setup], expand **Attribute Groups**, expand the relevant Group, and then select the relevant Description Attribute.

The **Attribute** Editor appears.

2. Click the **Validity** tab.

The available Object types appears.



3. Select which Object types the Attribute should be valid for, by clicking the check boxes on/off.

Note: It is possible to select Product, Classification and Asset Object Types. Selecting an Object Type will automatically let the Attribute appear in the **Description** field of all selected Object types.

Note: It is possible to select a Link Type. Selecting a Link Type will automatically let the Attribute appear on Product, Classification and/or Asset References.

Adding Inline References to Attribute Values

Inline references can be embedded in attribute values for attributes that have text and numeric text validation base types. An inline reference use case follows:

You are working with product objects whose model number is the same as their STEP ID (for example, '12345'). On these objects there is an attribute called 'Product Description' that needs to dynamically reference the object's model number within the text of the attribute. For a Product Description such as 'Acme lawn mower, model number 12345,' the inline reference to the STEP ID would populate '12345.' Using an inline reference in this instance helps to avoid duplication of effort—since the product's model number already exists in STEP as its ID, there is no need for users to manually enter it a second time into the system.

Also, in many cases inline references are easier to use than calculated attributes and easier on system resources. (For example, it may be easier for users to use an inline reference to pull in a STEP ID than to create a calculated attribute using the formula 'StepID()'.)

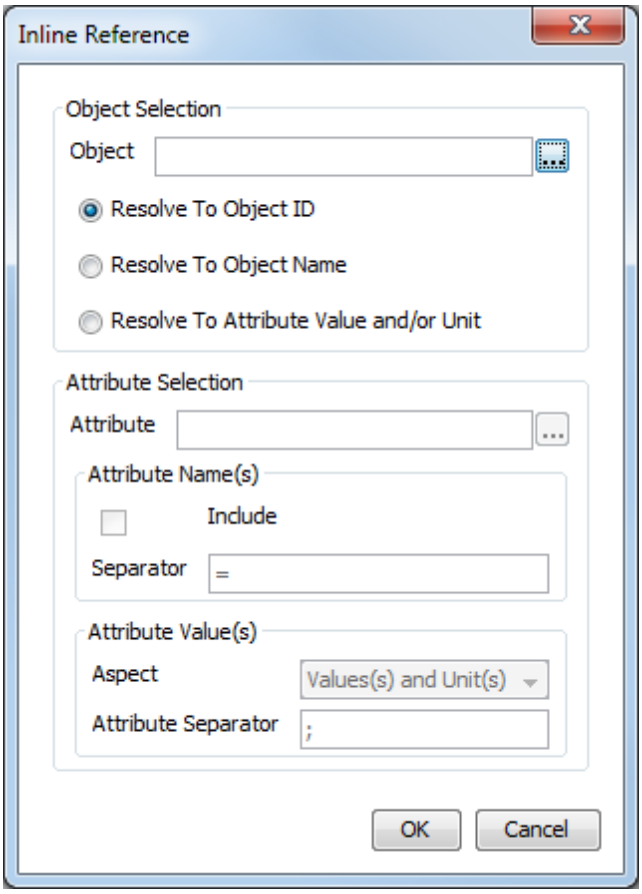
One attribute value can include several inline references. An inline reference can reference either a single attribute or an attribute group.

Inserting an Inline Reference using Workbench

1. Select the relevant node. The corresponding editor appears.
2. Click on the leftmost tab within the editor, which is different depending on the object type (for example, 'Product', 'Classification', 'Images & Documents', 'Publication,' and so forth).
3. Activate the **Value** field of the relevant attribute where you want to create or edit the inline reference.

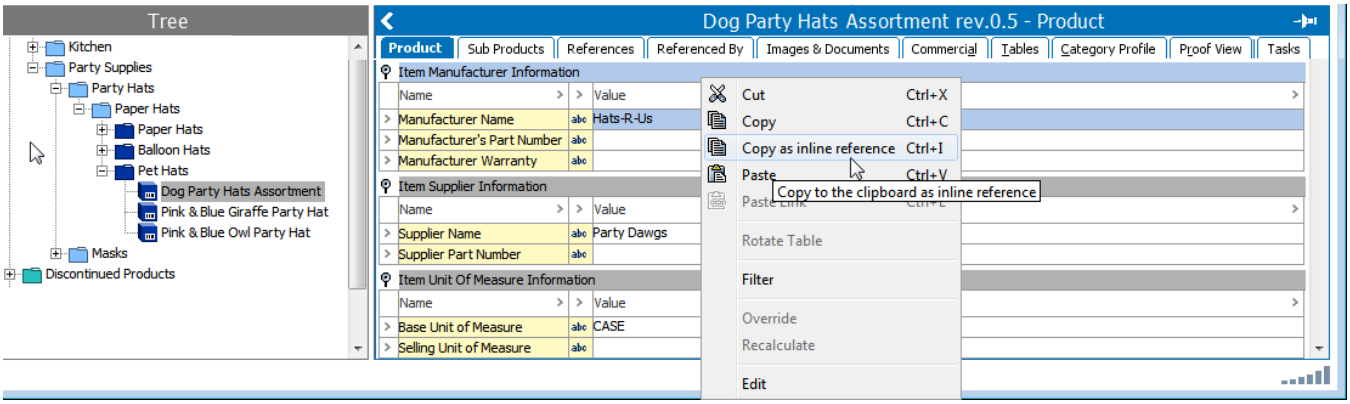
   Note: Press **Alt + F2** to open a value editor.

4. Place the cursor at the position in the **Value** field where you want to insert the inline reference, or place the cursor on an existing inline reference to be edited.
5. Right-click, and then click either 'Insert Inline Reference' or 'Edit Inline Reference' to display the **Inline Reference** dialog.



6. For the **Object** field, click the ellipsis button **(…)** to open the 'Select Object' dialog. Search for or browse to the object that holds the relevant attribute value, then click **Select**.
7. Choose 'Resolve to Object ID,' 'Resolve to Object Name,' or 'Resolve to Attribute Value and/or Unit' by clicking the corresponding radio button.
8. For the **Attribute** field, click the ellipsis button **(…)** to open the 'Select Attribute Group' dialog. Search for or browse to the relevant attribute(s) for the inline reference, and then click **Select**. You can choose a single attribute or an attribute group.
9. For **Attribute Name(s)**, check the **Include** box if the name of the attribute(s) should be included in the inline reference.
10. In the **Separator** field, type a separator to be used between the name and value of the attribute. The default is '=.' Note that the Separator field is not activated unless the **Include** box has been checked.
11. Under **Attribute Value(s)**, select the relevant **Aspect** from the dropdown list. Available options are 'Value(s) and Unit(s),' 'Unit(s),' and 'Value(s).'
12. In the **Attribute Separator** field, type a separator to specify how values should be separated. The default is ';.' The Attribute Separator field is only available if a multi-valued attribute is selected.
13. Click **OK**.

Alternatively, attribute values (if valid to use for inline references) can be right-clicked within workbench, and then copied and pasted into another attribute value as an inline reference. Working with that inline reference can then be done through standard editing (Alt + F2 to open the editor or by right-clicking the 'destination' attribute value and choosing to 'Edit Inline Reference').



Inserting an Inline Reference using Web UI

Rich text editing allows users to enter and edit text within a web browser. Inline references can be inserted into an attribute value using a Rich Text Editor (RTE) within a Node Editor.

Adding an inline reference in Web UI is similar to adding one in the workbench. For instructions, see the **Rich Text Editor** section of the **Web User Interfaces** documentation.

About Attribute Links

Attributes are linked to Product and Classification nodes in the **Product** tab and **Classification** tab of the Editors.

Attribute linking makes a selection of Attributes available to objects that are linked to (or below) the specified node in the Classification or Product hierarchy.

Before linking Attributes, the following issues must be considered:

- Where to link an Attribute:

  Before linking Attributes, it is very important to consider the scope of these Attributes, i.e. for which objects are they supposed to be relevant.

  A best practice is to organize the Product hierarchy or the Classification hierarchy so similar Products, that share the same characteristics, are grouped together. These Products will then be able to share a set of Attributes by inheritance from the hierarchy.

- Which hierarchy levels to link to:

  If you have a hierarchy where an upper level node has more sub nodes that all share the same characteristics, the Attributes should be linked at that upper node level. The Attributes will then be inherited to the sub nodes.

If, however, the Attributes only are relevant to a sub node, they should be linked there. Read about Attribute inheritance below.

- Which Product object types to link to:

It may also be important to specify which type of Product the attribute is describing.

For example, Attributes such as “Price” and “Order Code” will only be relevant to Products with object type “Article”. This means that e.g. Pro-ducts with object type “Product Family” cannot have a “Price” or an “Order Code”.

On the other hand, an Attribute such as “Long Description” may be relevant to a “Product Family”, but not to an “Article”.

- Attribute inheritance:

An Attribute that is linked to a node in the hierarchy is inherited to the nodes below. This way, Attributes are shared among nodes.

An example, a Specification Attribute called “BladeWidth” might be linked at the “Saws” node (below the” Power Tools” node). Therefore, the Attribute “BladeWidth” is available to both the sub nodes “Circular Saws” and “Table Saws”, by inheritance.

This inheritance feature is useful when there are globally used Attributes – these are Attributes that are ”common” to all Products. For example, “Warranty”, “Price”, and “CountryofManufacturer” would most likely be global Attributes, since each and every Product would have a value for these characteristics. Therefore, these Attributes would be linked at the top node in the hierarchy, and in doing so, be inherited to all of the nodes below. See the figure in Online Help - About Linking Attributes.

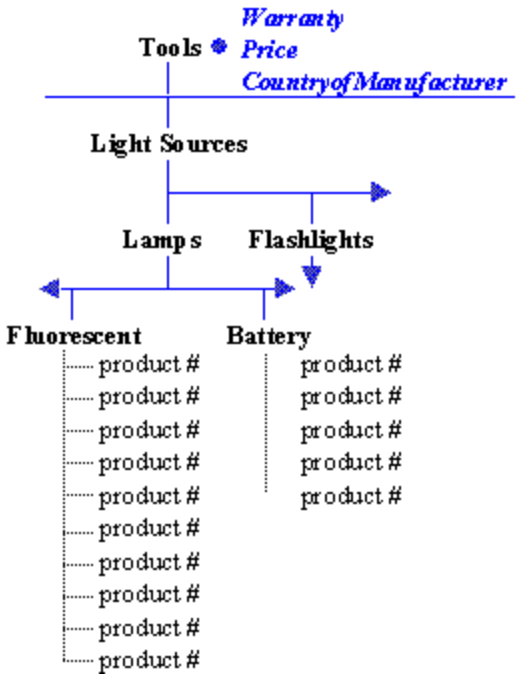


Figure 1 above: A part of a product hierarchy showing the principle of the inheritance of product attributes.

Linking Specification Attributes

Specification Attributes are linked to Products or Classifications either from:

- Attribute Editor **References** tab
- Product Editor **References** tab: **Linked Attributes** field
- Classification Editor **References** tab: **Attributes** field

Linking Specification Attributes is available for:

- Classifications
- Products

Note: Description Attributes are linked globally to an Object Type. They will then be automatically displayed in the **Description** fields.

Note: If an Attribute is not in the list of selectable Attributes, it may not be available in the current Context.

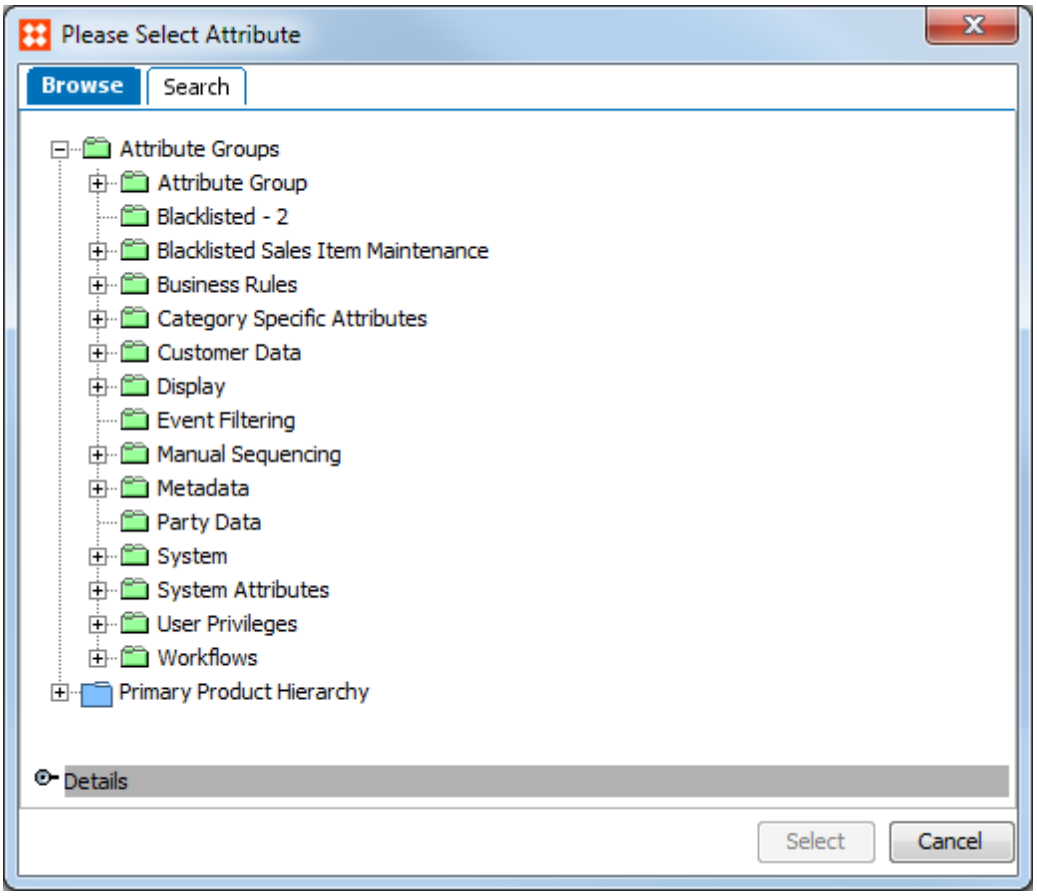
---

1. In [Tree], select the relevant node.

   The corresponding Editor appears.

2. Click the **References** tab.
3. Under **Linked Attributes**, click **Link to Attribute**.

   The **Select Attribute** dialog box appears.



4. Search or browse for the Attribute Group you want to link to, select one or more Specification Attributes from the group, and then click **Select**.

The selected Attribute(s) will appear in the list of linked Attributes.

One or more Specification Attributes have now been linked to a Product or a Classification node, and the Attribute(s) will be available to the linked Products and sub products.

Linking Orphan Attributes

Orphan attributes are linked in the **Attribute** Editor **References** tab. It is also possible to link Attributes from the **Product** Editor **Product** tab and the **Classification** Editor **Classification** tab.

An Orphan attribute is a Specification Attribute with a value that has not been linked to a Classification or a Product through the Hierarchy.

Linking an Orphan attribute means “picking it up” by linking it to a Product or a Classification hierarchy. The difference between linking an ordinary Attribute and an Orphan is that the Orphan already has a value for specific Products.

1. In [Tree], select the Product with the Orphan attribute.

The **Product** Editor appears.

Note: Alternately you can search and select the Attribute in the [Search Result] tab.

2. In the Product Editor, click the **Product** tab.

Note: An orphan attribute is indicated by *italic*.

3. Click the orphan attribute to be linked.

   The **Attribute** Editor appears.

4. In the Attribute Editor, click the **References** tab.
5. Click **Link to Classification** or **Link to Product**.

   The **Select Classification** dialog box or **Select Product** dialog box appears.

6. Search for or browse to the Classification or Product hierarchy node to link to.
7. Select the relevant Classification or Product node.
8. Click **Select**.

   The Attribute will appear under **Valid in Classification** or under **Valid in Products**.

   An Orphan attribute has now been made valid for the selected Classification or Product node (and below), and the *italic* indication will disappear from the Product Editor.

Sorting Linked Attributes

Sorting Attributes linked to Product, Classification, or Publication objects is done in the **Display Sequence** field of the **Attribute** Editor, if such a field is set up on your system. Both Description (metadata) and Specification attributes may be sequenced.

Attributes can be sorted for:

- Products

Sorting Attributes is setting the sequence (1, 2, 3, or 10, 20, 30 etc.) in which the linked Attributes will be shown in the Product Editor.

The attributes shown in the Product Editor can be sorted according to the order set in the Display Sequence attribute. This can be done in two ways:

- By setting a number in the **Display Sequence** field of the attribute itself.
- By setting a number in the **Display Sequence** column under **Linked Attributes**.

Setting a Display Sequence Number on the Attribute

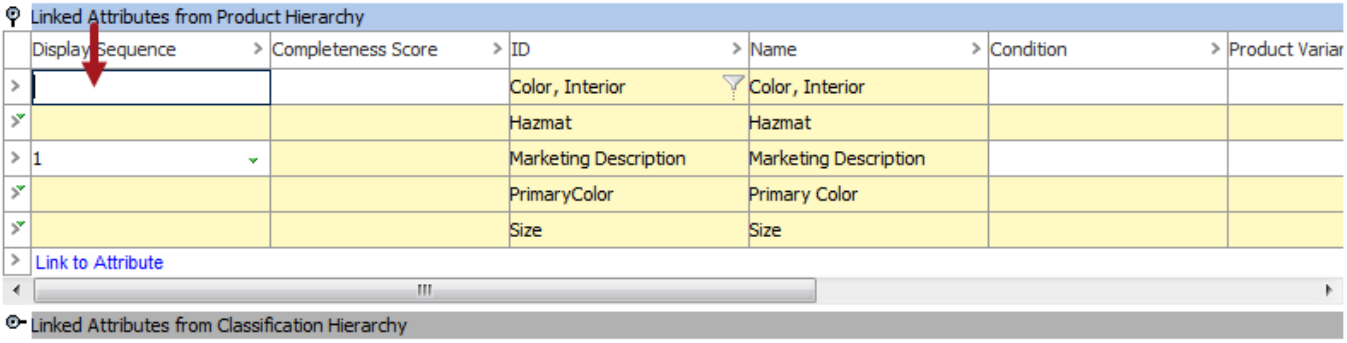
1. In [System Setup], unfold **Attribute Groups**, then click the relevant attribute. The **Attribute** editor appears.
2. In the Display Sequence field, type a number.
3. Type the appropriate sequence number for each attribute in an attribute group.

Note: You can choose to set a sequence by using e.g. 10, 20, 30, etc., or by using 1, 2, 3, etc. If you want to add attributes in between, you can then use for instance 15 in the first case, or 1.5 in the latter case.

Note: If you do not know which attribute on your system is set up as the Display Sequence attribute, you can go [System Setup], click **Users & Groups**, and verify the name of the Display Sequence attribute under **Default Attribute to use as Attribute Display Sequence Attribute**. Alternatively, click a product, click the **References** tab. Under **Linked Attributes**, the first column shows the Display Sequence attribute on your system.

Setting a Display Sequence Number under Linked Attributes

1. In [Tree], unfold the Product Hierarchy, click the relevant product folder or product.
2. Click the **References** tab.
3. Under Linked Attributes, type a number in the first column (the Display Sequence attribute column) in front of the relevant attribute.



Note: You are only able to edit the Display Sequence attribute column if the attribute is linked in at the Product Hierarchy level that you have selected. The **Inherited from** column shows where the attribute is linked in.

Note: An attribute can be linked in to one or more Attribute Groups. In the **Attribute Groups** column, you can see to which Attribute Group(s) the attribute belongs.

4. Optionally, if you want to create another Display Sequence at the level that you have selected, right-click the relevant row, click **Create Local link**. You are now able to type a local value in the Display Sequence column.

Note: Attributes that have no sequence number, and therefore are alphabetically sorted, will be placed last in the list.

Important: Information about the Display Sequence of attributes is exported in an XML export. If the Display Sequence number is set on the attribute itself, the XML Export will contain the Attributes with Display Sequence attribute value. If the Display Sequence number is set on the Attribute Link, the XML Export will contain the Display Sequence information on the Attribute Links. A combination of the two is possible.

For more information, see the STEP PIM User Guide, "Sorting Attributes" on page 144.

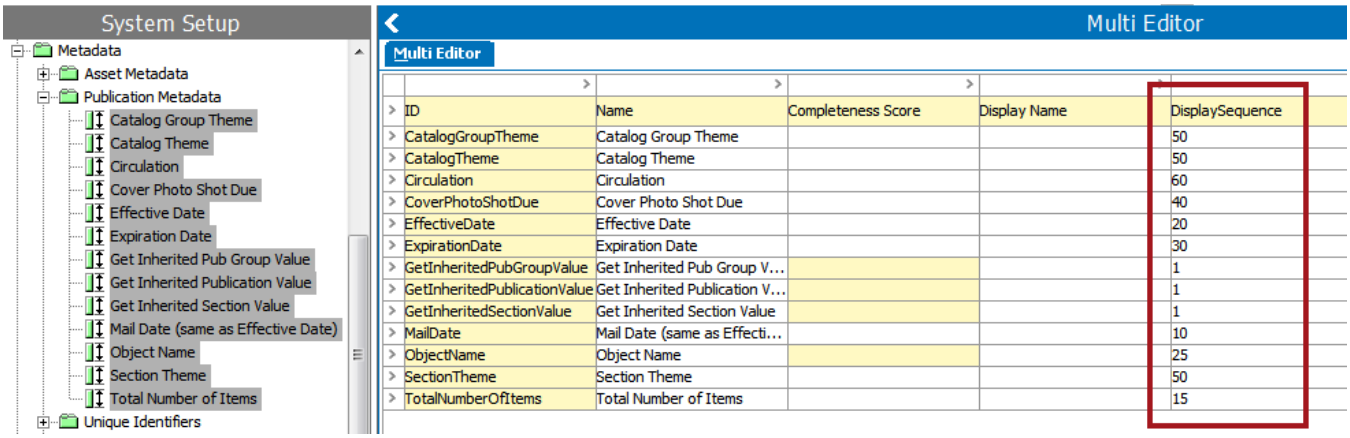
Setting the Priority of a Product Variant Attribute

1. In [Tree] select the "super product" (the parent) containing the product variants.
2. Click the **References** tab.
3. Under Linked Attributes, type a number in the column for the "Attribute indicating the Priority of Product Variant Attributes" attribute specified in User & Groups under "Product Information Manager Default Settings".

Setting the Attribute Display Sequence on Publication (Green) Hierarchy Objects

Metadata (Description) attributes on publication hierarchy objects (publication groups, publications, sections, and planned pages) may also be sequenced using a display sequence attribute.

Because publication hierarchy objects do not use Reference Types, attribute sequencing cannot be set from a 'Linked Attributes' flipper on the References tab, which is unavailable on publication hierarchy objects. Instead, metadata attribute sequencing must be set up at the attribute level. However, attributes may be multi-selected to allow sequencing from a single location.



Viewing Attribute Values

Attribute values can be viewed in the Attribute Editor: [References] tab.

Viewing the values of an Attribute means getting a list of the objects in which the current Attribute has values, and what the value is.

The list is very useful to get an overview of where an Attribute is used, e.g. before cleaning up, deleting, or merging Attributes.

Note: You cannot edit in the list.

---

1. In the Attribute Editor, click the **References** tab.
2. Unfold **Values for Attribute**.

A list will appear of Objects with the corresponding values of the Attribute.

3. Click an object in the list to open the corresponding Editor, e.g. to edit the value.

Merging Attributes Overview

Merging consolidates a Master Object with other nearly identical Objects in order to eliminate redundancies. Merging improves data quality by establishing a distinct set of Attributes and Attribute Values without unnecessary repetition.

For example, the following Attributes exist, perhaps because users did not follow the naming convention rules. Although these attributes have different names and IDs, they are logically the same and are valid for the same set of products.

- Width
- width
- Width (mm)

Merging eliminates two of the attributes, leaving a single attribute to track 'width' on all products.

After reviewing the following merge requirements, choose Option 1 to use Merge Attributes or Option 2 to use Bulk Update, based on the state of the data to be merged.

Attribute Merge Requirements	Option 1	Option 2
Single-Update Mode[1]	Yes	No
Identical Dimension Dependencies	Yes	No
Identical Validation Base Types	Yes	No

Attribute Value Merge Allowed On	Option 1	Option 2
All products where merged attributes are linked	Yes	No
Subset of products	No	Yes

Merging Allowed When	Option 1	Option 2
Source **is not** Dim Dependent / Target **is** Dim Dependent	No	Yes

Merging Allowed When	Option 1	Option 2
Source **is** Dim Dependent / Target **is not** Dim Dependent	No	No
Source is **Number** Validated / Target is **Text** Validated	No	Yes
Source is **Text** Validated / Target is **Number** Validated	No	Yes [2]

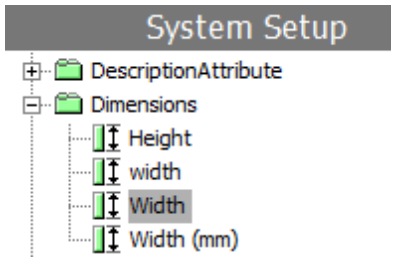
[1] other users are not allowed to update any values in STEP during the merge operation

[2] only for text values containing a number

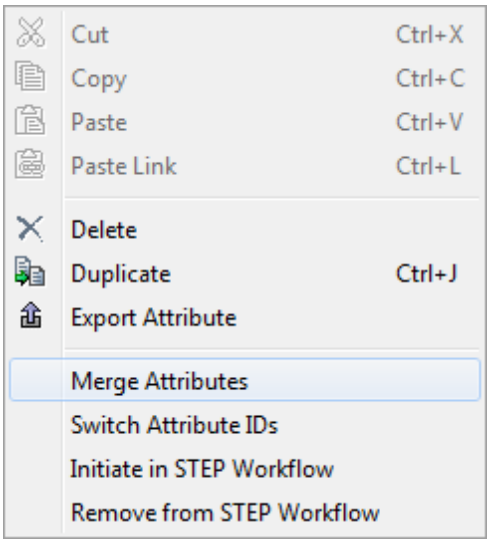
Merging Attributes - Option 1

Using the Merge Attributes option, Values from a specified Attribute (selected manually or automatically based on priority) are written to the 'master' Attribute. Redundant attributes are deleted as part of the merge process.

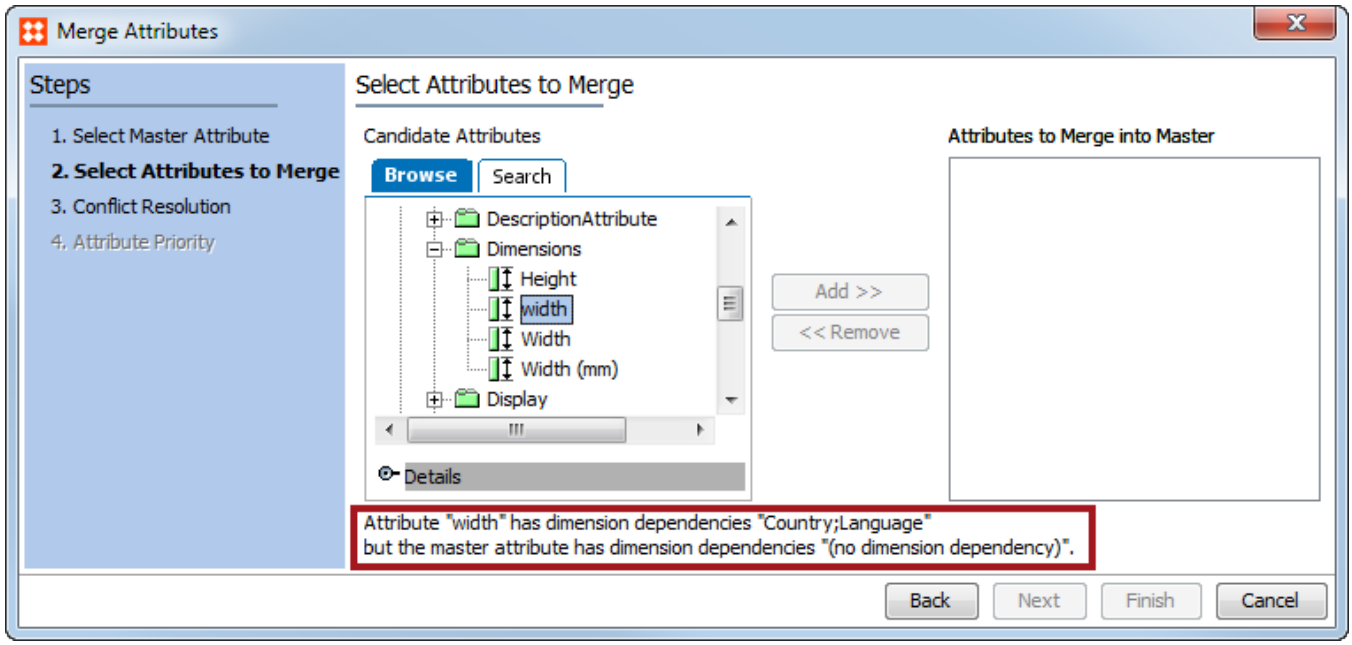
1. On the System Setup tab, select the **Master** Attribute. Other Attribute values will be merged into the Master.



2. Right-click and select **Merge Attributes**.



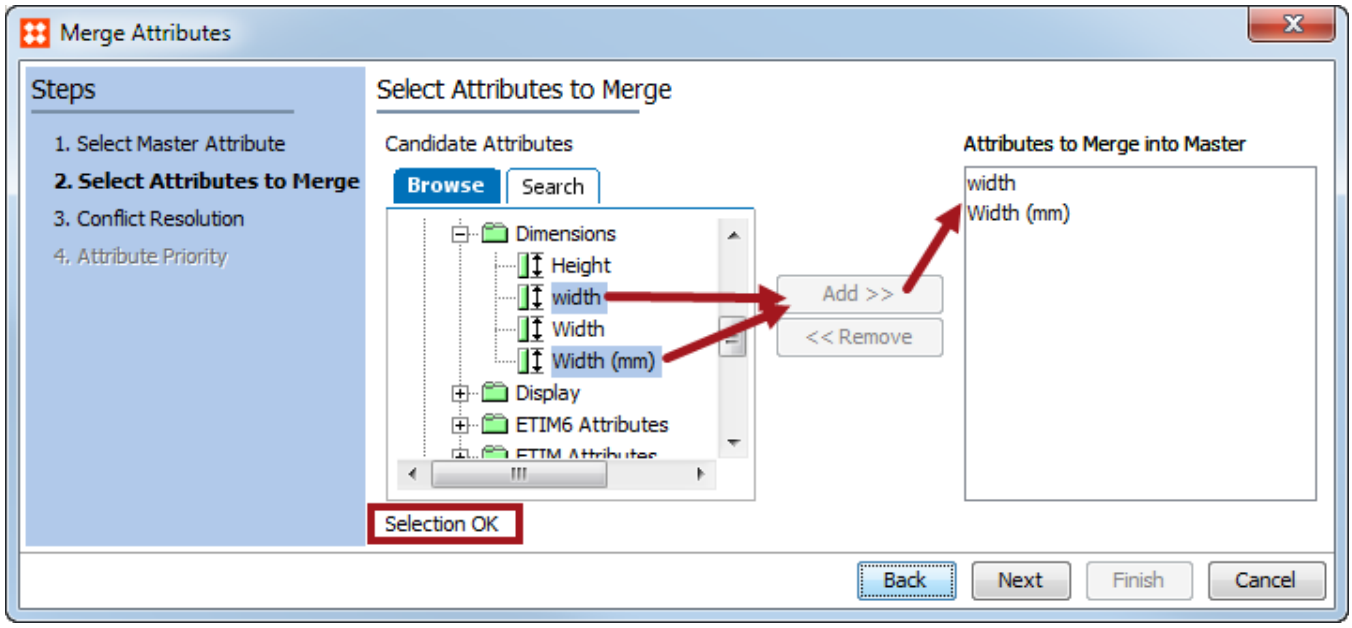
3. Select each redundant attribute individually to display any processing errors.



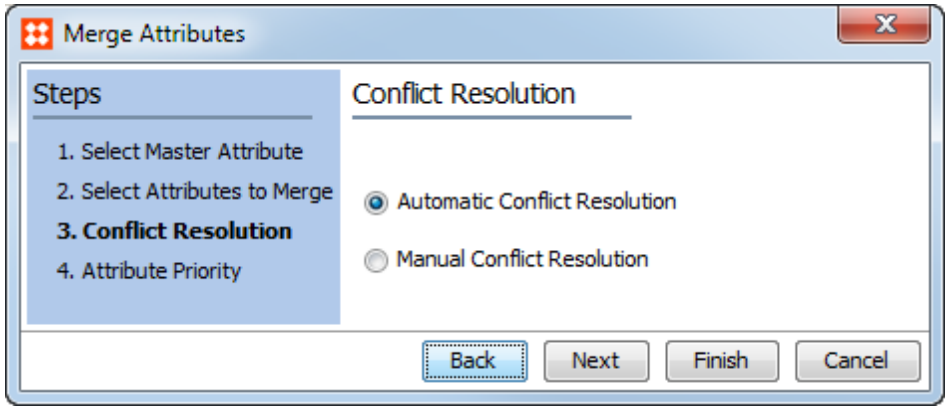
The following Errors must be resolved before the merge can proceed.

Error	Resolution
Same attribute	Selected Attribute and Master Attribute must be different.
Dimension dependencies	Selected Attribute and Master Attribute must have the same Dimension dependencies.
Validation base type	Selected Attribute and Master Attribute must have the same Validation Base Type.
Attribute Type	Selected Attribute and Master Attribute must have the same type (specification / description).
Different LOVs	Selected Attribute and Master Attribute must have the same LOV. Merge LOVs before merging Attributes.
Full text index	Selected Attribute and Master Attribute must have the same full text index setting.

4. When the Selection indicator displays OK for each Attribute that will be merged, multi-select all attributes to merge, click the **Add** button to display them on the Attributes to Merge into Master list.



5. Click **Next** and select a Conflict Resolution option.



- **Automatic Conflict Resolution** allows you to specify the priority of Attributes to be used for merging when two or more of the selected Attributes exist on same Object. Starting with the first attribute in the Attribute Priority list, the first value found will be written to the Master Attribute for all Objects merged.
- **Manual Conflict Resolution** allows you to decide individually, for each Object merged, which value to write to the Master Attribute.

Follow the steps below to complete the merge, based on the Conflict Resolution option selected.

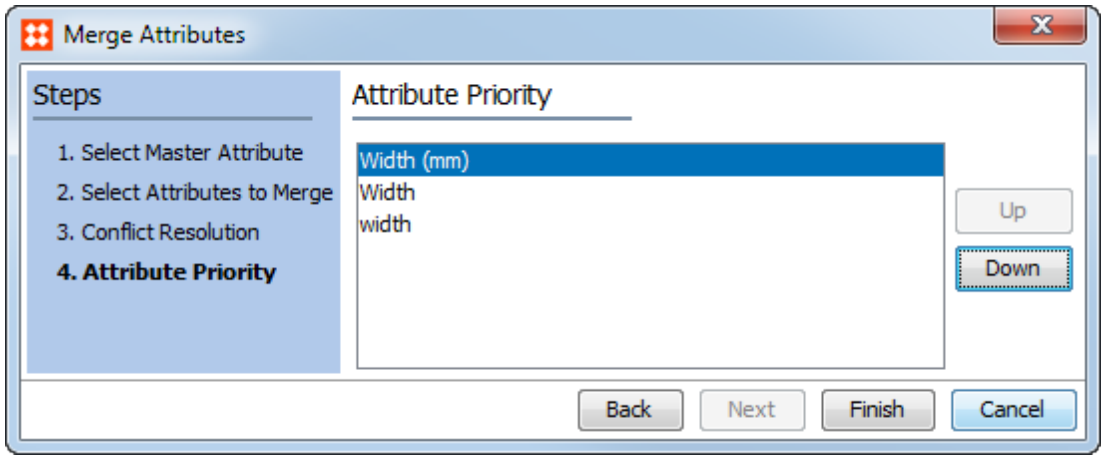
Using Automatic Conflict Resolution

An object has the following redundant attributes:

Dimensions

Name	Value
width	12 in
Width	13 in
Width (mm)	14 in

1. Click **Next** and use the Up / Down buttons to arrange the Attributes in order of priority.



2. Click **Finish** to perform the merge.

   The products now have no redundant attributes and the value of the top priority attribute is assigned to the Master Attribute.

Dimensions

Name	Value
Width	14 in

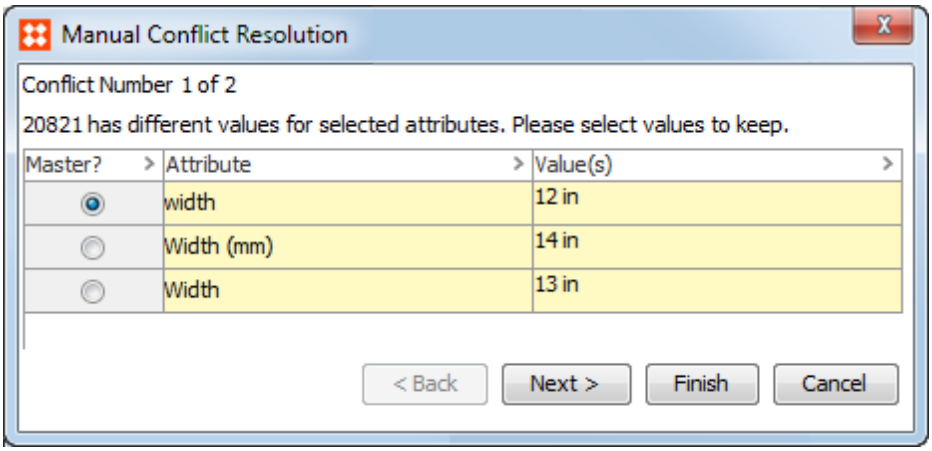
Using Manual Conflict Resolution

An object has the following redundant attributes:

Dimensions

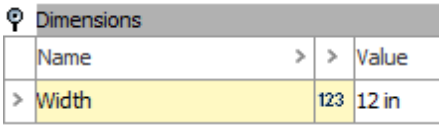
Name	Value
width	12 in
Width	13 in
Width (mm)	14 in

1. Click **Finish** and manually select the value to assign to the Master Attribute for each conflict presented.



2. Click **Next** until all conflicts are resolved.
3. Click **Finish** to perform the merge.

   The object now has no redundant attributes, and the value of the selected attribute is assigned to the Master Attribute.



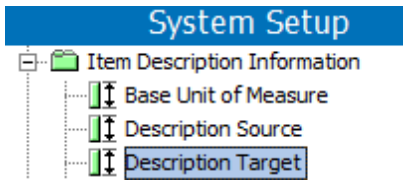
Merging Attributes - Option 2

Using Bulk Update, values in the Target Attribute are overwritten with values from the Source Attribute. Redundant attributes must be deleted manually.

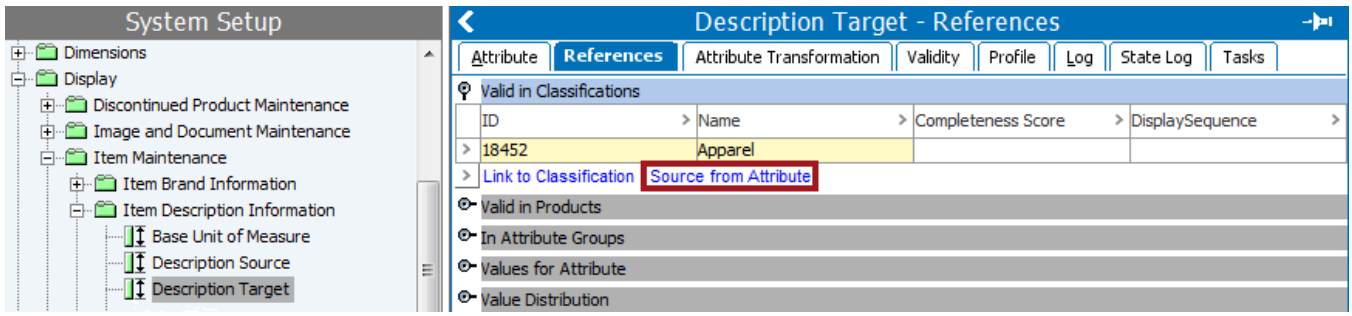
For a successful merge, the steps below will verify the following for the Target Attribute and Source Attribute:

- Both attributes have the same Product and Classification links. This prevents orphaned Target Attributes after the merge.
- Both attributes have the same Object Types. This ensures all Source Attribute values can be written to the Target Attribute.

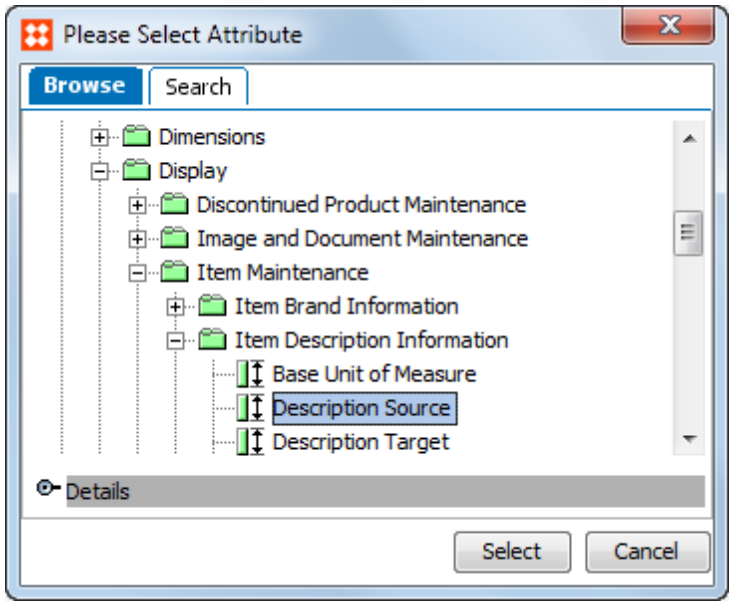
1. In **System Setup**,select the **Target Attribute**.



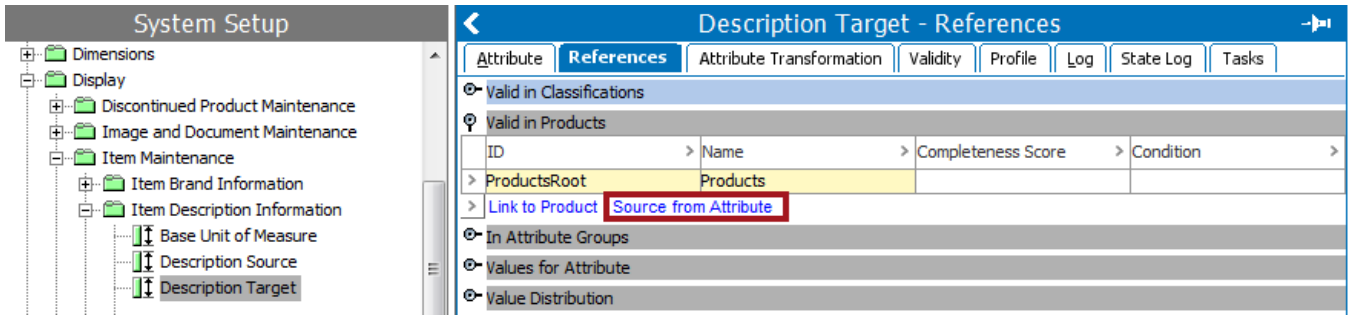
2. On the References tab, open the **Valid in Classifications** flipper, and click the **Source from Attribute** link.



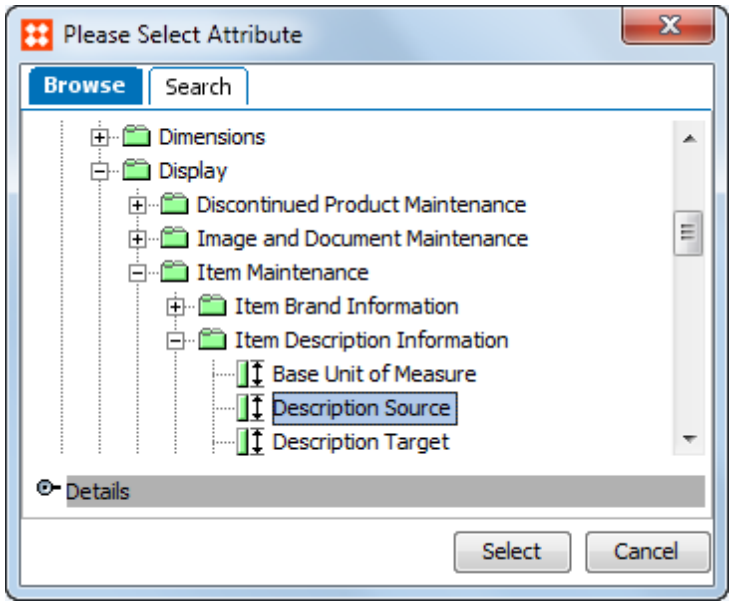
3. Use Browse or Search to find the **Source Attribute** and click **Select**. The Classification links of the Source Attribute are copied to the Target Attribute.



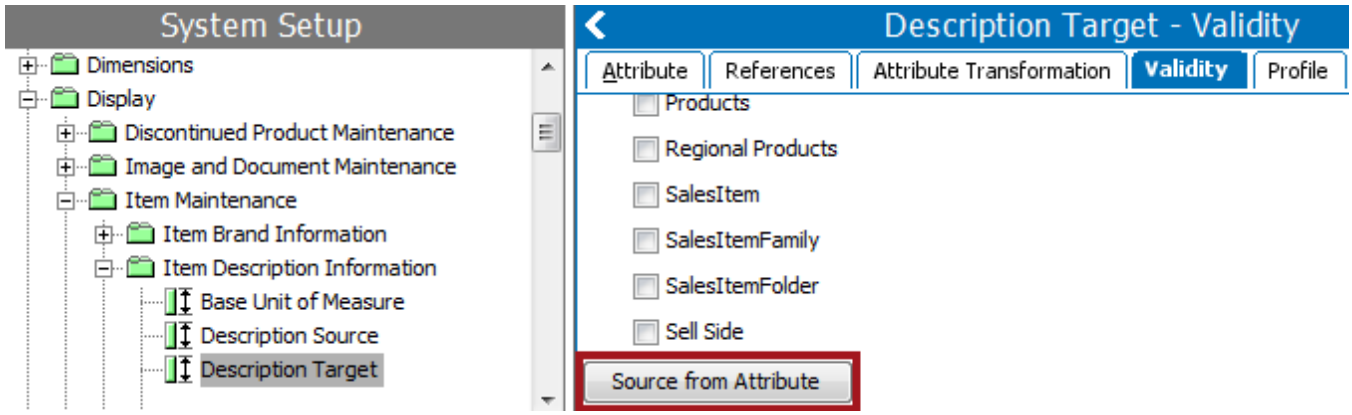
4. On the References tab, open the **Valid in Products** flipper, and click the **Source from Attribute** link.



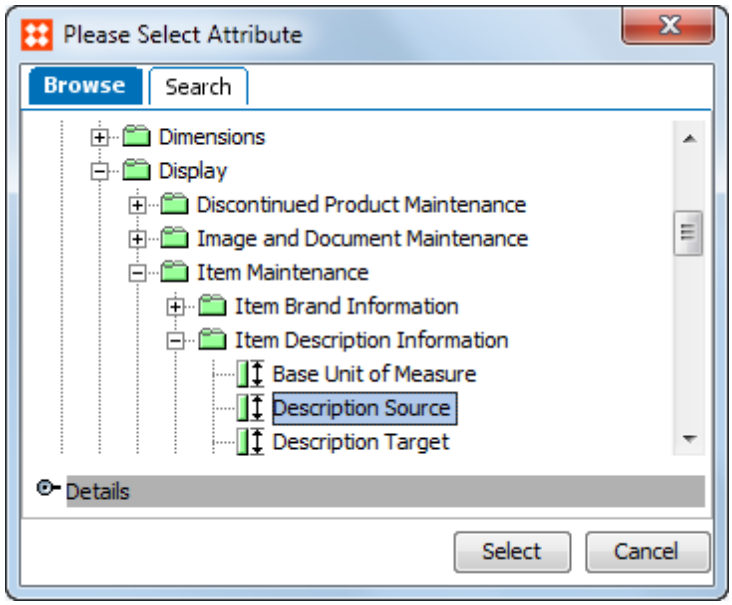
5. Use Browse or Search to find the **Source Attribute** and click **Select**. The Product links of the Source Attribute are copied to the Target Attribute.



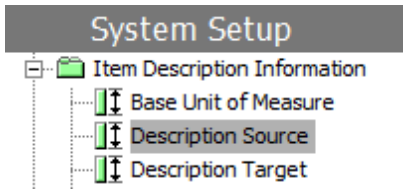
6. On the Validity Tab click the **Source from Attribute** button.



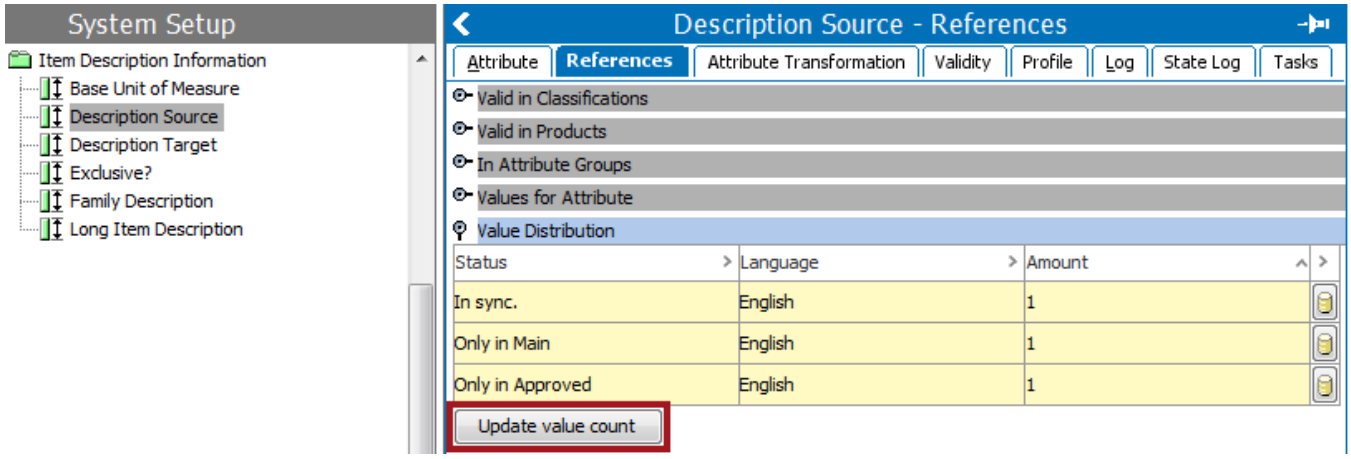
7. Use Browse or Search to find the **Source Attribute** and click **Select**. The Validity of the Source Attribute is copied to the Target Attribute.



8. In **System Setup**,select the **Source Attribute**.



9. On the References tab, open the **Value Distribution** flipper, and click the **Update Value Count** button.



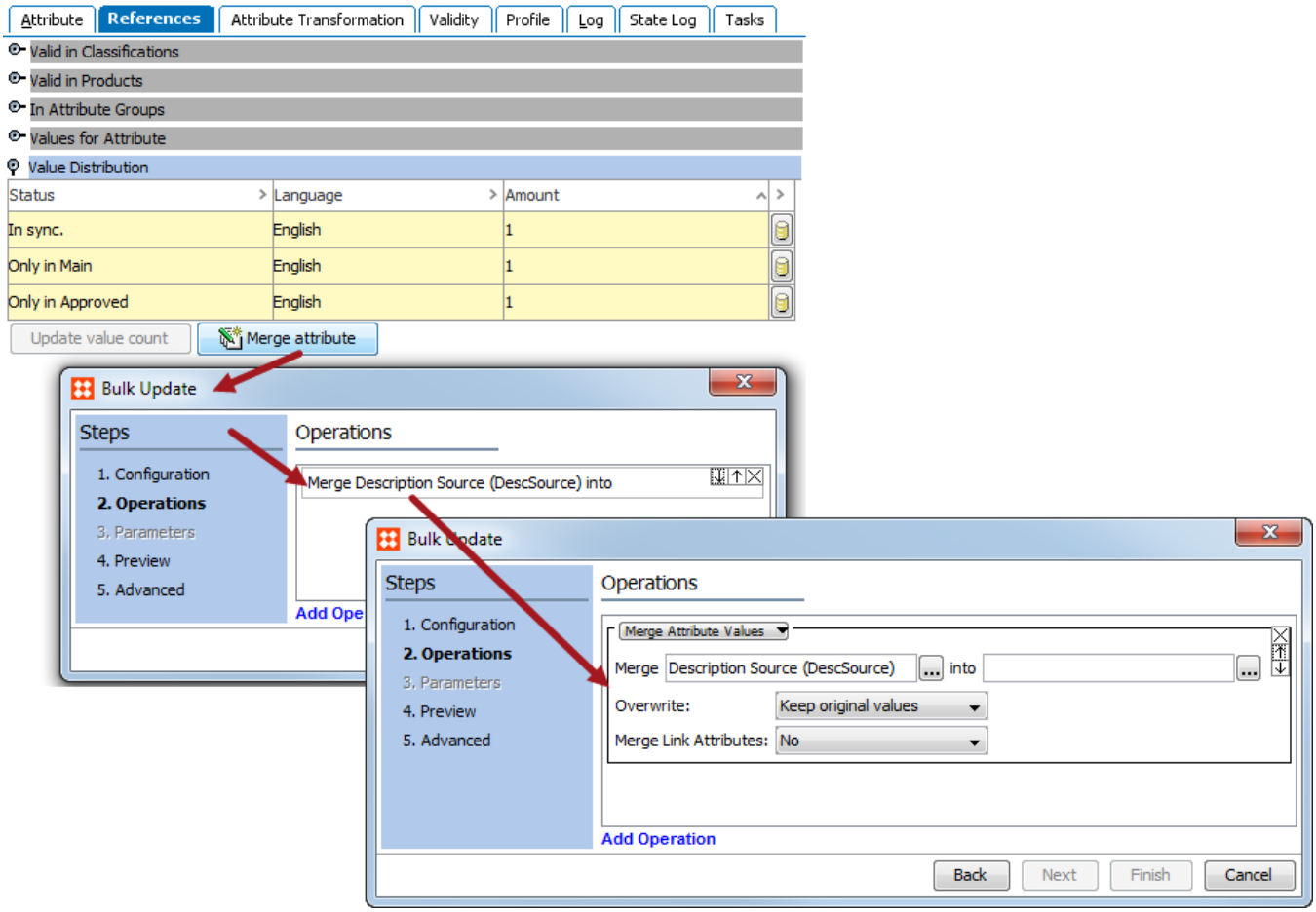
The Value Distribution table is updated with the number of products found (Amount column) based on the Context (Language column) and Workspace (Status column).

The Status column can include the following entries:

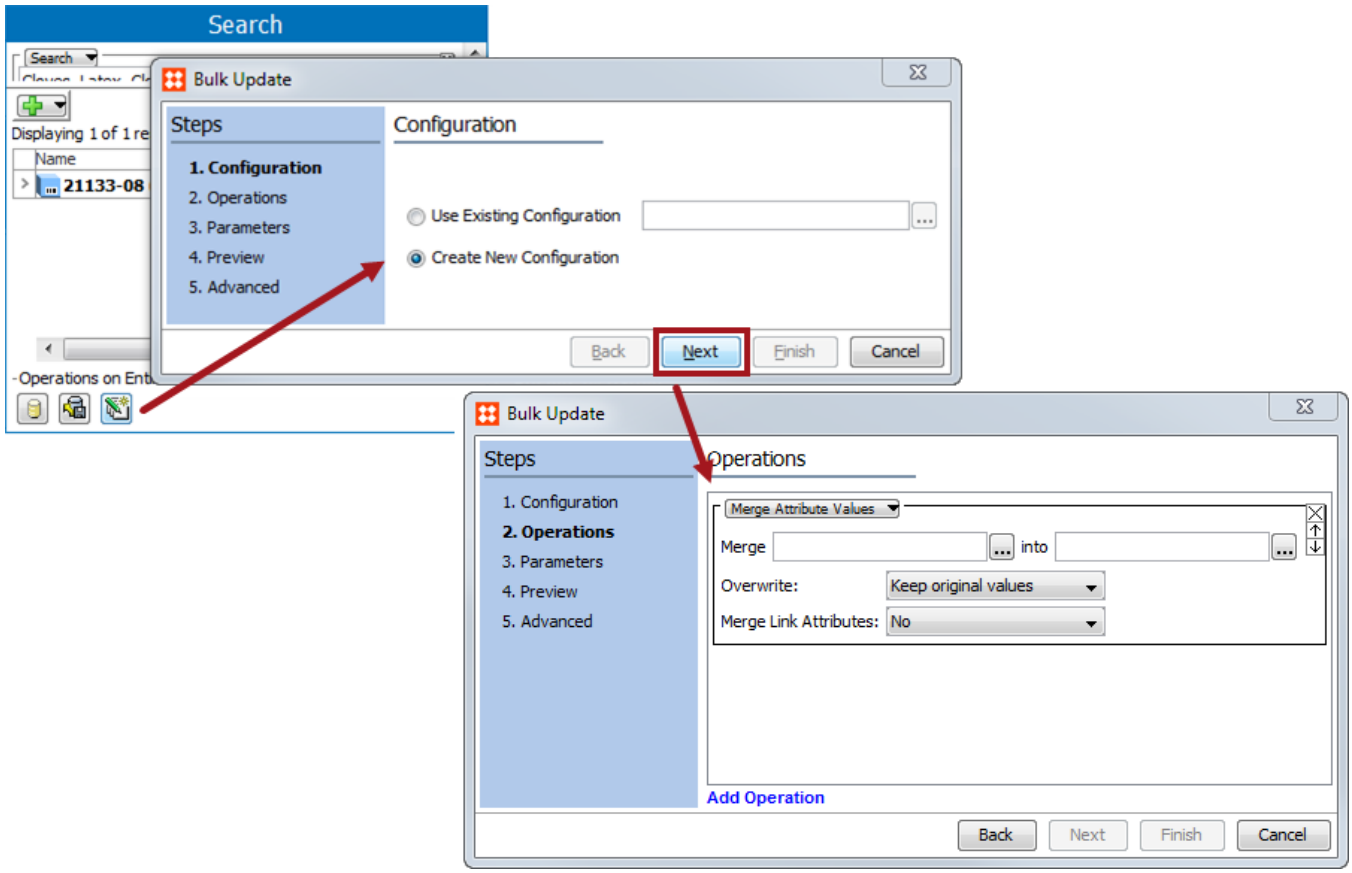
- **In Sync**. indicates the number of Products where Attribute values are the same in both Maintenance Workspace and Approved Workspace.
- **Only in Main** indicates the number of Products where Attribute values are available only in the Maintenance Workspace.
- **Only in Approved** indicates the number of Products where Attribute values are only in the Approved Workspace (the value has been removed from the Maintenance Workspace).

10. Determine which products need to be merged:

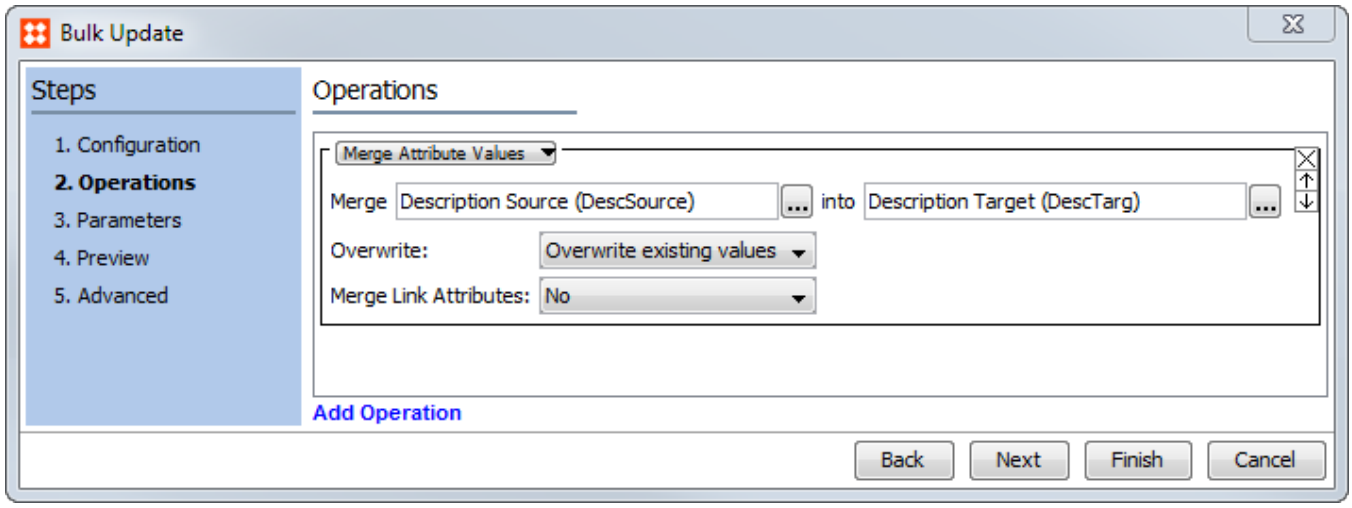
- For **all** products with values for the attribute (including those with values in Maintenance Workspace and Approved Workspace), click the **Merge attribute** button, and click the **Merge {Source Attribute} into** entry to display the available Merge Attribute Values fields.



- For **only a subset** of products with values (for example, only those with values in the Maintenance Workspace), go to the **Search** tab, create Search Results of the Products to merge, click the **Bulk Update** button and click **Next** to display the available Merge Attribute Values fields.

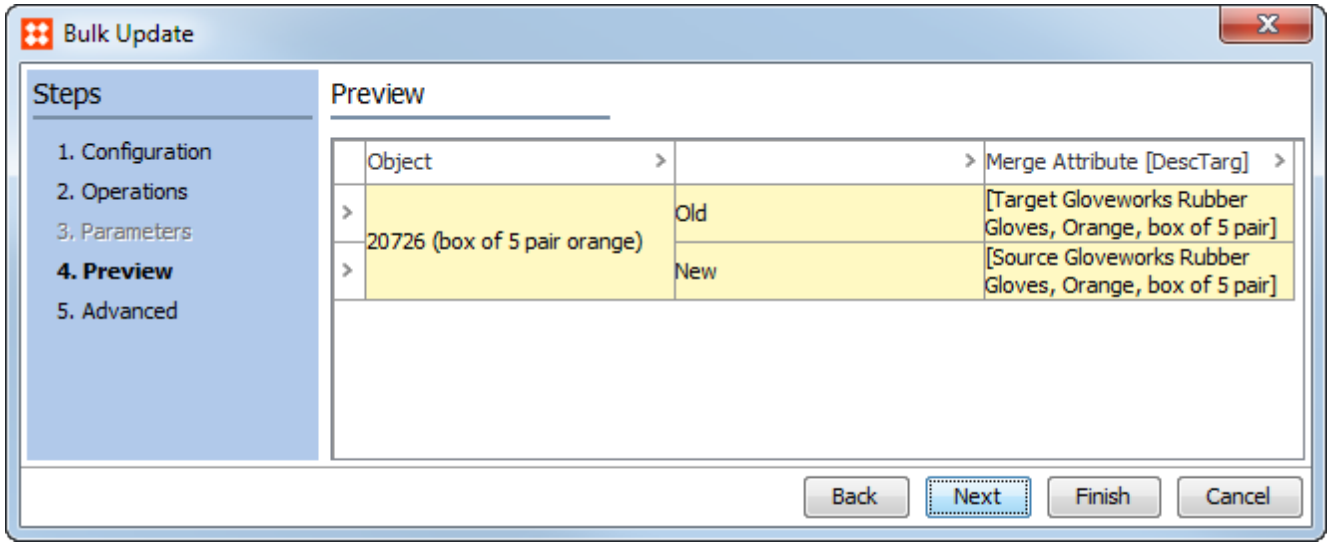


11. Complete the Operations step of the Bulk update by defining the Merge Attribute Values:
    - For the **Merge** and **into** fields, click the ellipsis button **(…)** and select the **Source Attribute** and the **Target Attribute**.
    - For the **Overwrite** field, choose **Overwrite existing values** (the Target value is updated with the Source value) or **Keep original values** (the Target value remains unchanged). If the Target Attribute is blank, the Source Attribute value is written to the Target Attribute.
    - For **Merge Link Attributes** field, choose Yes or No, to determine if link attributes are merged.
    - Click **Next**.

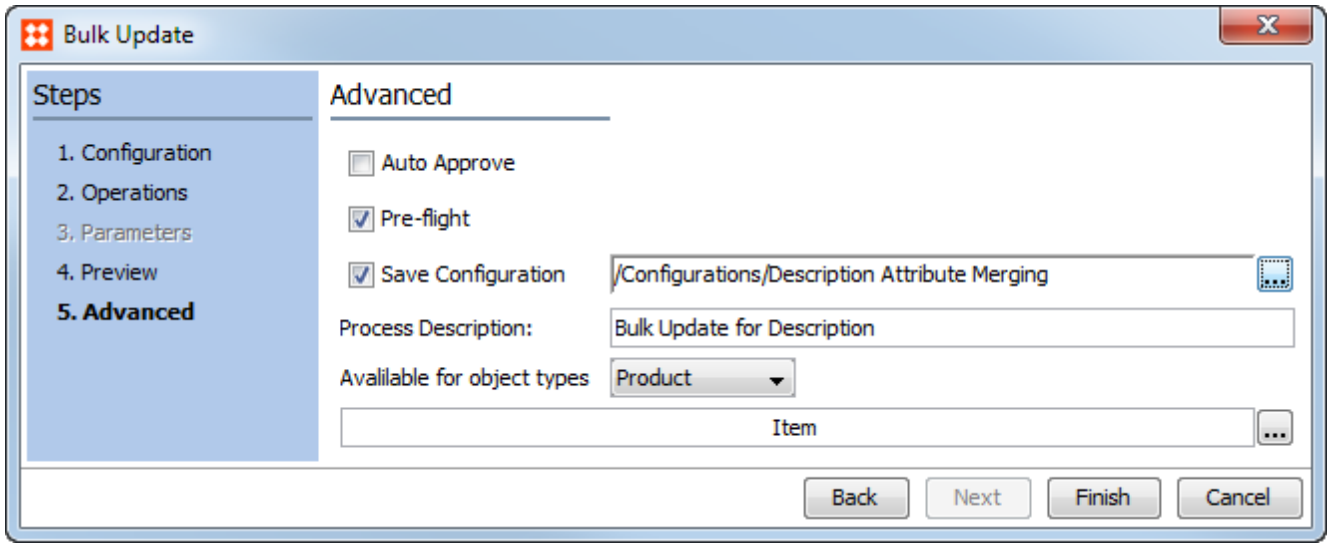


12. Verify the Preview page displays the desired value updates and click **Next**.

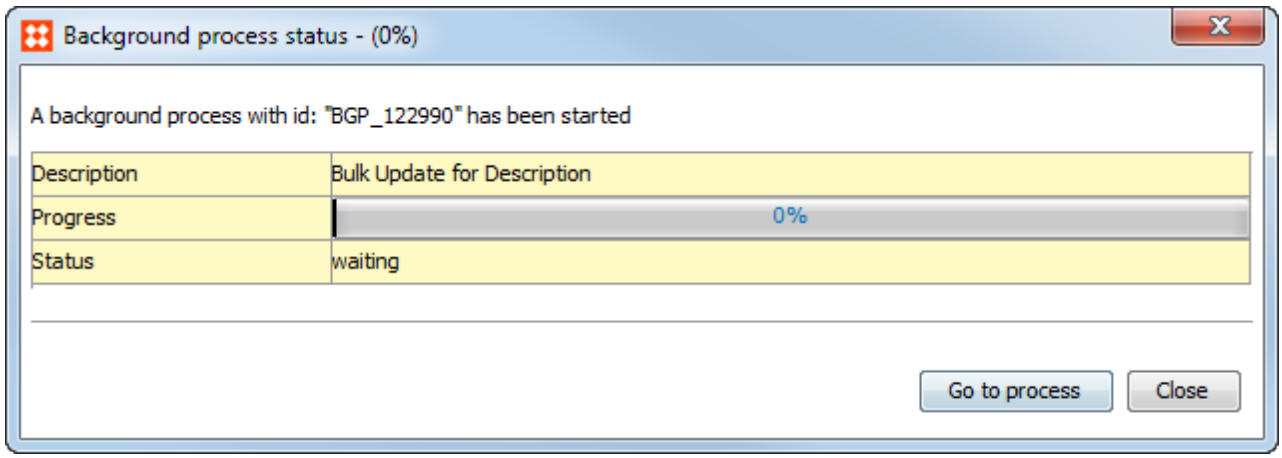
Note: If modifications are needed, use the **Back** button to change a Bulk update setting or use the **Cancel** button to close the Bulk Update dialog.

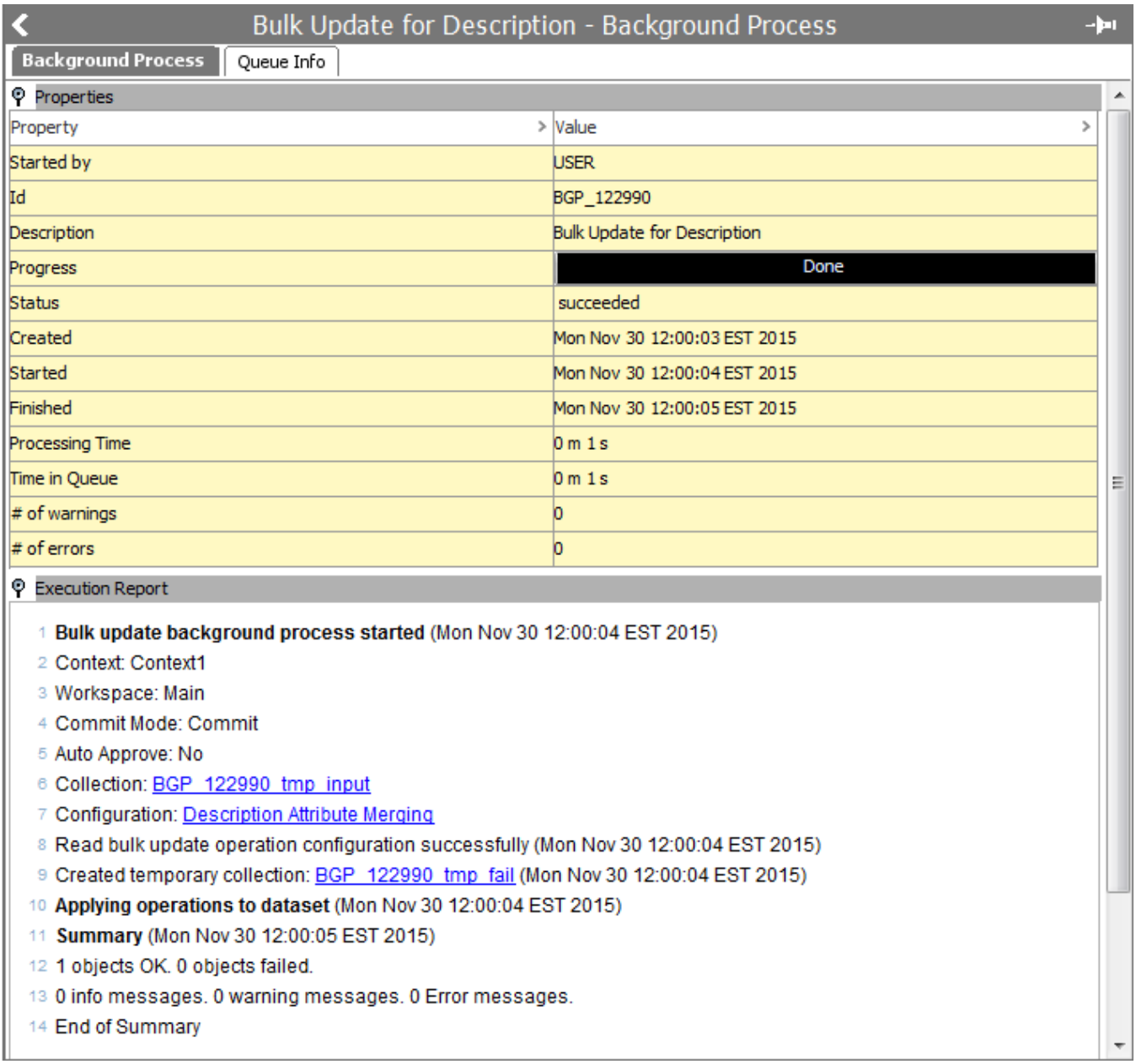


13. Select one or more Advanced options:
    - **Auto Approve** automatically approves the Products once the Attribute values have been merged.
    - **Pre-flight** tests the merge functionality without actually changing any data.
    - **Save Configuration** writes the selected Bulk Update options to a Configuration on Tree. This enables you to easily run this Bulk Update process again.
    - **Process Description** determines the text displayed in the Background Process
    - **Available for object types** allows you to limit the object types that are affected by the merge.
    - Click **Finish** to complete the Bulk Update configuration and start the Merge operation.



14. The Background process status dialog tracks the progress of the operation. Click the **Go to process** button to monitor each step of the merge.





Note: The merge operation does not delete the Source Attribute or its value. If desired, delete the Source Attribute manually.

Converting Attributes from Date to ISO Date

It is recommended that users store date information in STEP using ISO attributes so that they can fully take advantage of search and compare features in STEP, as well as to maintain consistency in information across multiple regions. There are many benefits to using an ISO Date attribute in STEP. The primary reason is that the lexicographical order of the representation of ISO Dates corresponds to chronological order. Simply stated, this means that STEP handles ISO Dates better than Date attributes when doing search, compare, and order operations.

For this reason, STEP users who use attributes with a Date Validation Base Type may want to merge or transform the Date attribute into an ISO Date attribute. Using STEP, users can also switch the ID of the attributes making it possible to easily replace an attribute without having to do a lot of changes to the configurations where the attribute is used.

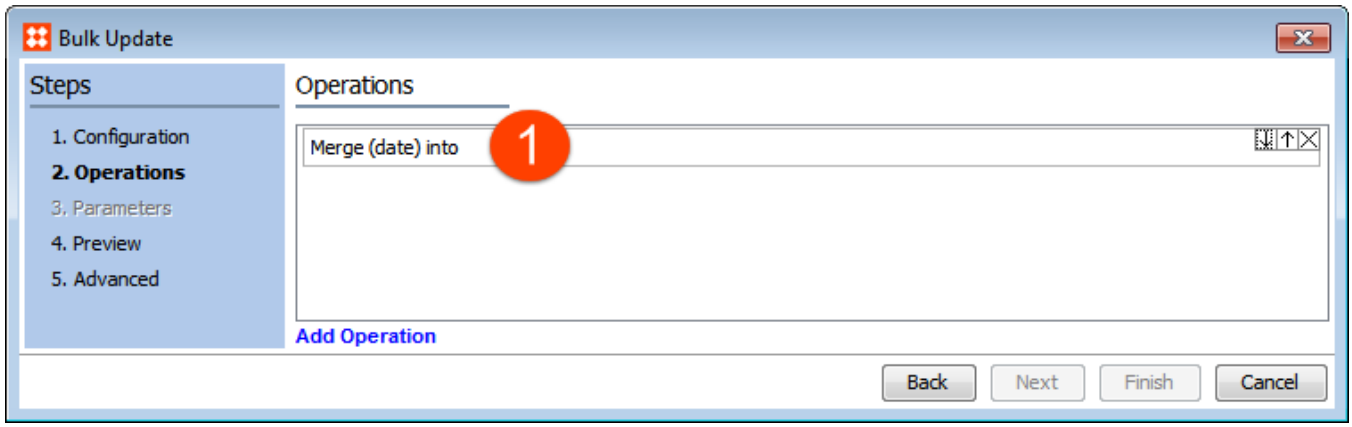
Merging Attributes

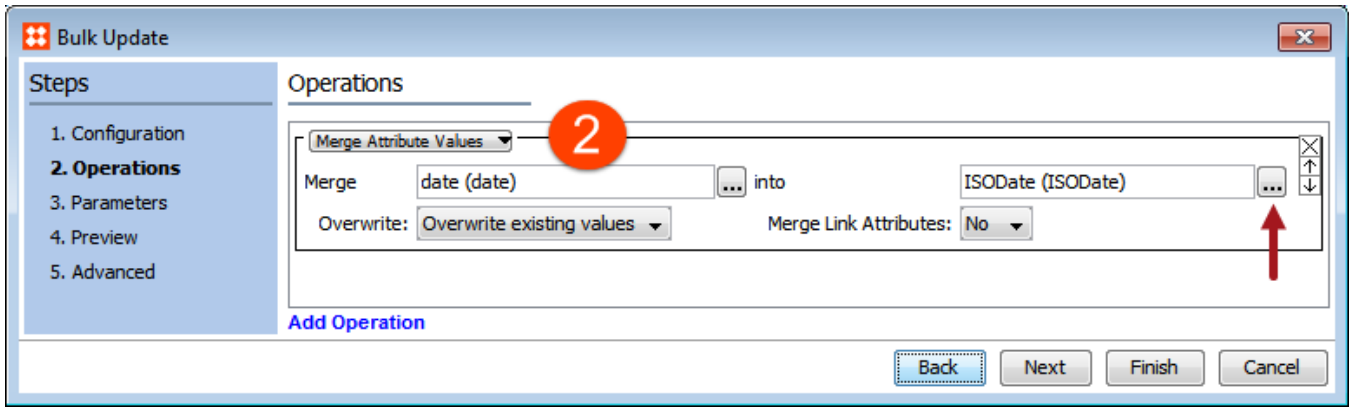
Merging attributes can be done from the References tab of the attribute editor. The References tab of the editor has a section called 'Value distribution.' Here it is possible to start a Bulk Update operation called 'Merge attribute.' The operation will merge data from a source attribute to a target attribute. Date types will be converted to honor the target validation base type while merging. Merge can be performed multiple times to make sure data in the target attribute is up to date. Below are the steps for merging attributes:

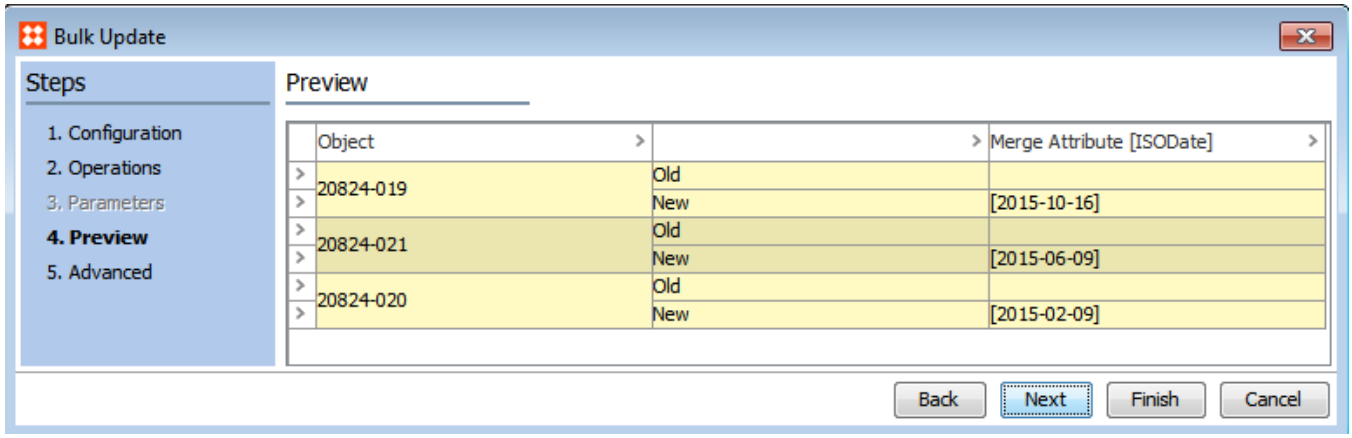
1. Log in to STEP Workbench.
2. On the System Setup tab, navigate to the *source* attribute you want to merge to a target attribute and click on it.
3. Go to the References tab and click the Value Distribution flipper to see the Merge attribute option.
4. Click the **Update value count** button to update the Status / Amount.
5. Next, click the **Merge attribute** button.

The Bulk Update wizard will display. To finish the merge, follow each step shown:

1. Click within the 'Merge (source attribute name) into' box to expand the Merge Attribute Values menu.
2. The source attribute will be pre-populated on the screen and a target attribute must be selected. You may also choose to overwrite the existing target attribute values and merge link attributes, if desired.
3. Click Next to preview the data. Verify the 'New' values shown on the screen are correct before continuing. If overwriting the existing target attribute values, you will see the original (pre-merge) target attribute value in the 'Old' row.
4. From the Preview screen, either continue on to the next step (Advanced) to finish making selections or click Finish to start the merge process.
5. Confirm that the background process completed successfully.







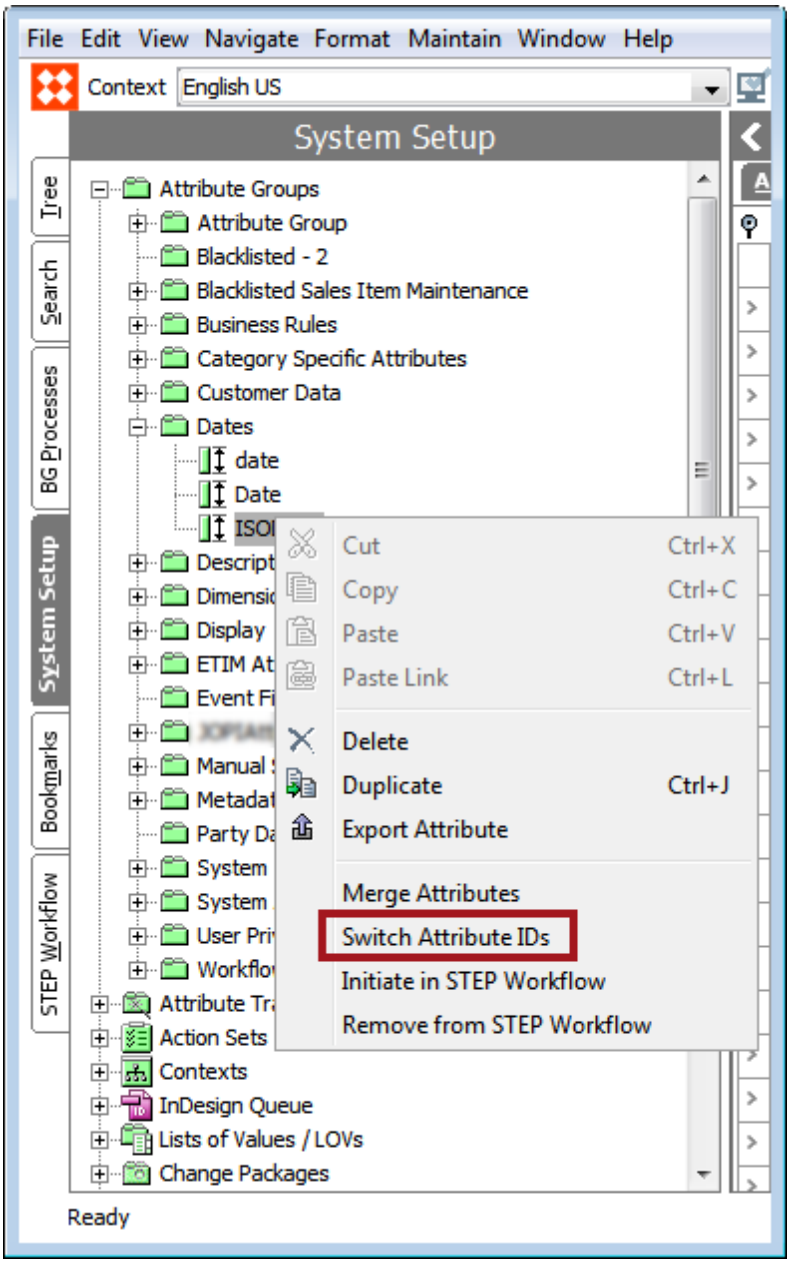
For additional information about merging attributes, see the 'Maintaining Attributes' section of the System Setup / STEP Super User documentation.

Switching Attribute IDs

After a merge is complete, a user may want to start using the target attribute values in place of the values from the source attribute. This can easily be achieved by switching the ID of the source and target attributes. In any STEP configuration where the source attribute is configured for use, the values from the target attribute will be used after the switch.

This also means that if the target attribute ID is configured for use in any configuration, the source attribute values will be used. Users should verify the impact of the ID switch before starting the process and update any configurations as needed.

1. To begin this process, start by navigating to the *target* attribute name in the System Setup tab. Right-click on the attribute name and select **Switch Attribute IDs**.
2. Next, choose the source attribute from the tree shown on the 'Select Attribute for ID-Exchange' screen or use the Search tab to find / select the source attribute.
3. **Click on Select only after you visually confirm that your source attribute selection is correct.**
4. Confirm that the IDs have been switched by looking at each attribute description.


File Edit View Navigate Format Maintain Window Help
Context English US
System Setup
Tree
Search
BG Processes
System Setup
Bookmarks
STEP Workflow
Attribute Groups
Attribute Group
Blacklisted - 2
Blacklisted Sales Item Maintenance
Business Rules
Category Specific Attributes
Customer Data
Dates
date
Date
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste Link Ctrl+L
Delete
Duplicate Ctrl+J
Export Attribute
Merge Attributes
Switch Attribute IDs
Initiate in STEP Workflow
Remove from STEP Workflow
Action Sets
Contexts
InDesign Queue
Lists of Values / LOVs
Change Packages
Ready

Mandatory Attributes

It is often the case that certain attributes are considered critical to all processes, or to a particular process, and therefore there is a desire to make them mandatory (e.g. to enforce that they are populated). There are a variety of methods for doing this, ranging from simple visual cues prompting users to populate value, or going so far as to prevent approval and/or processing of an object until the required values are populated.

This topic describes:

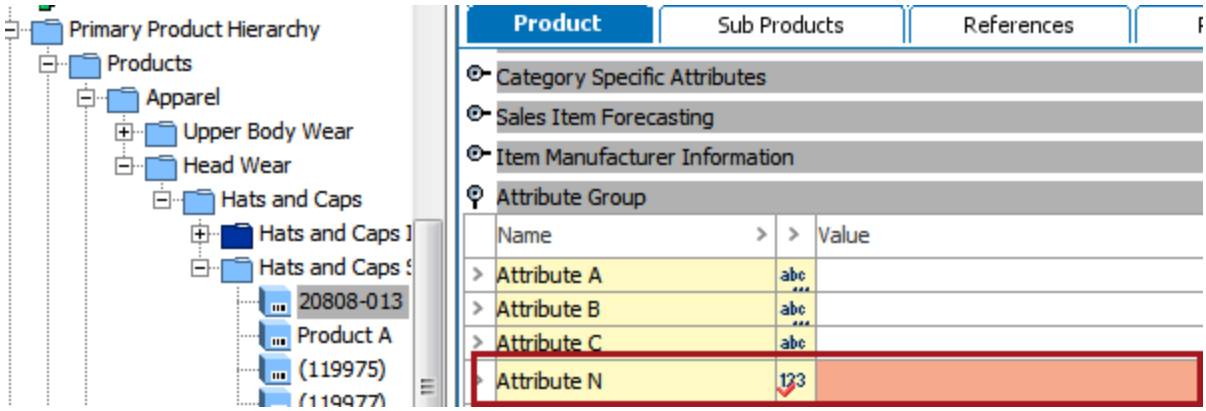
- The options for making attributes mandatory
- Detailed configuration and end user information for the various settings, or links to other topics that provide this information
- Considerations for how the different mandatory options behave across the varying interfaces (workbench, Web UI, and Smartsheets)
- Recommendations for how and when to use the various settings

Approval Mandatory Attributes (Global) / Visual Indicator in Workbench

An attribute can be set to be mandatory directly on the attribute, which enables a visual indicator in the workbench to prompt users to populate the value. This can optionally be used in conjunction with a system setting to enforce that attributes marked as mandatory are populated prior to approval of any object on which the attribute is valid.

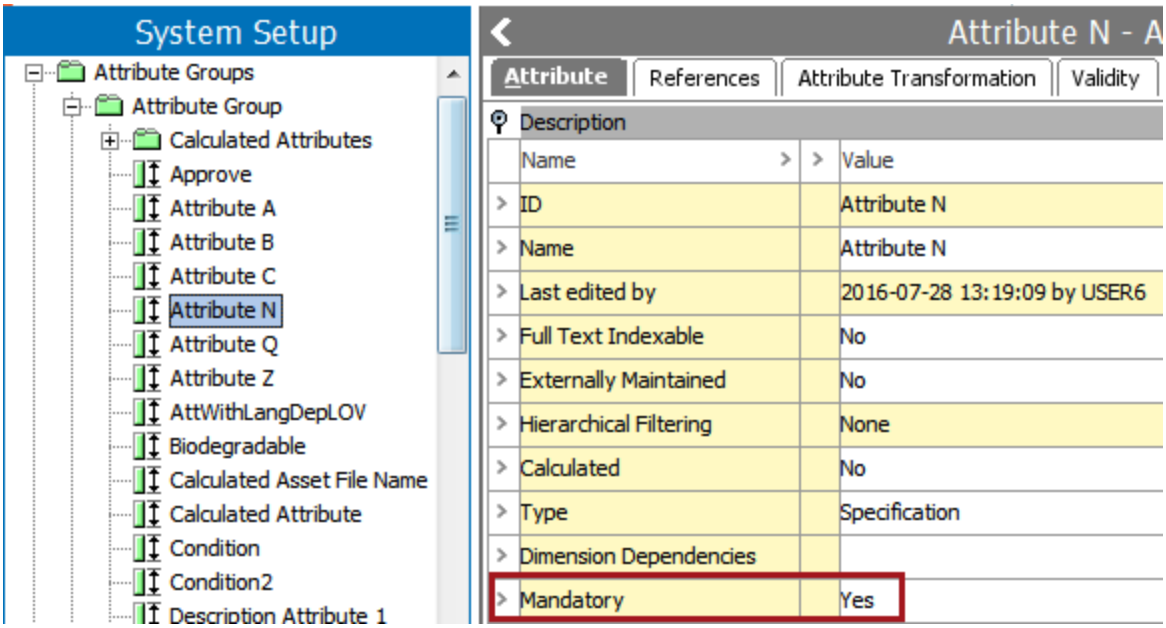
Workbench Visual Indicators

Attributes can be configured to provide a visual indicator within the workbench to prompt users to enter a value.

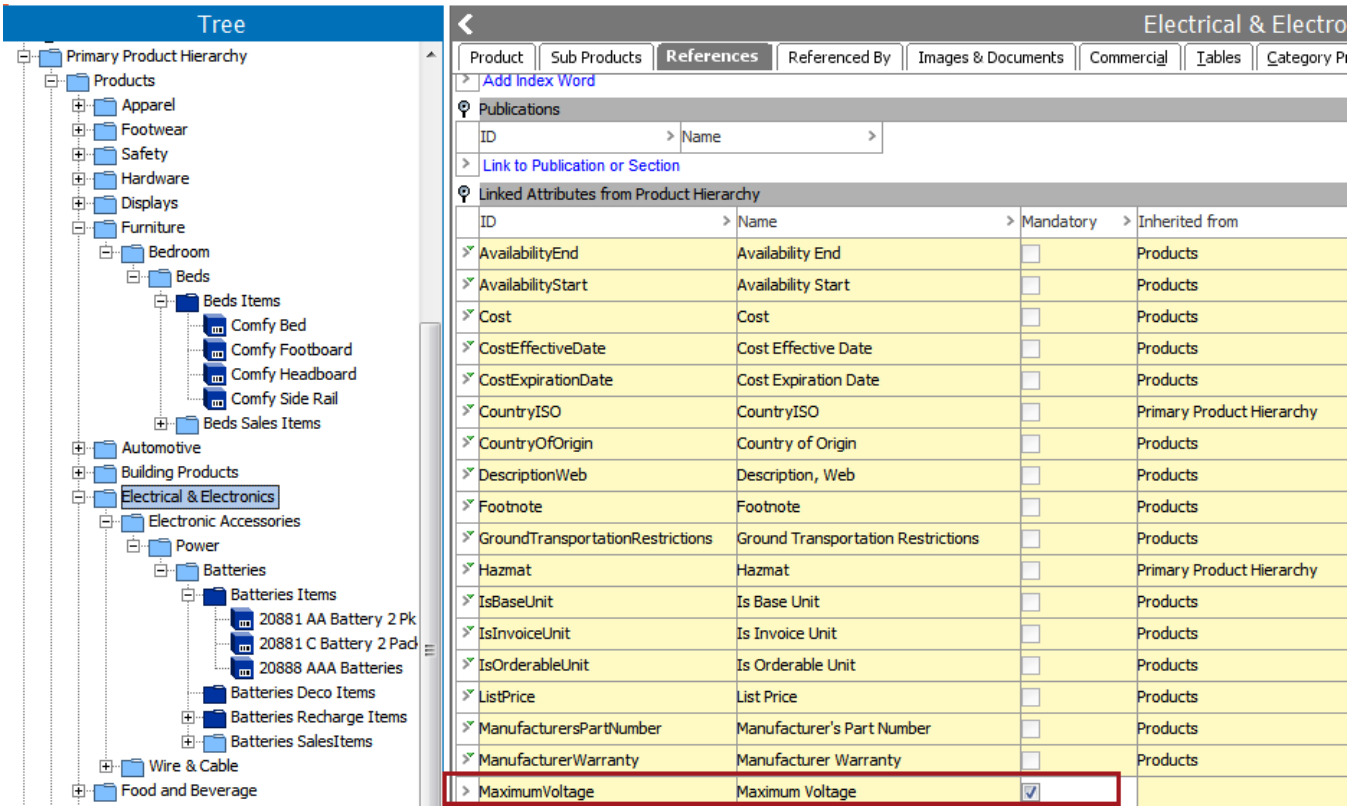


The attribute indicator has a red checkmark and the value field is highlighted red for attributes marked as mandatory. Note that the red highlighting is present regardless of the workbench interface (e.g. the standard object editor in Tree navigator, the Tasks tab on an object in Tree, or the STEP Workflow Items tab when using the STEP Workflow navigator). This is effective in calling attention to key attributes that users generally populate within the STEP Workbench.

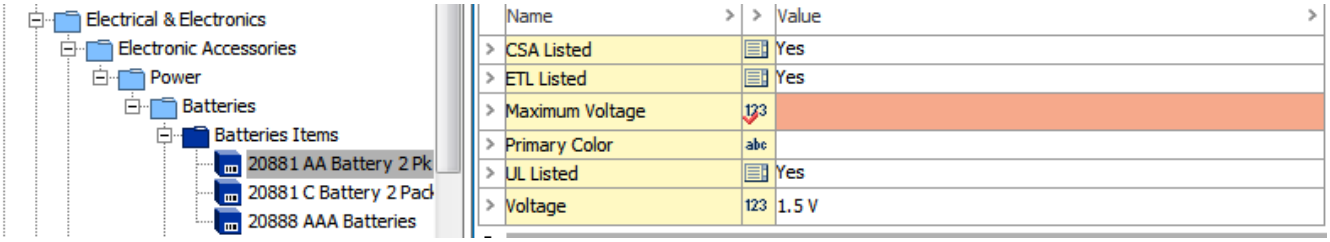
To configure this, set the **Mandatory** setting on the attribute to **Yes**.

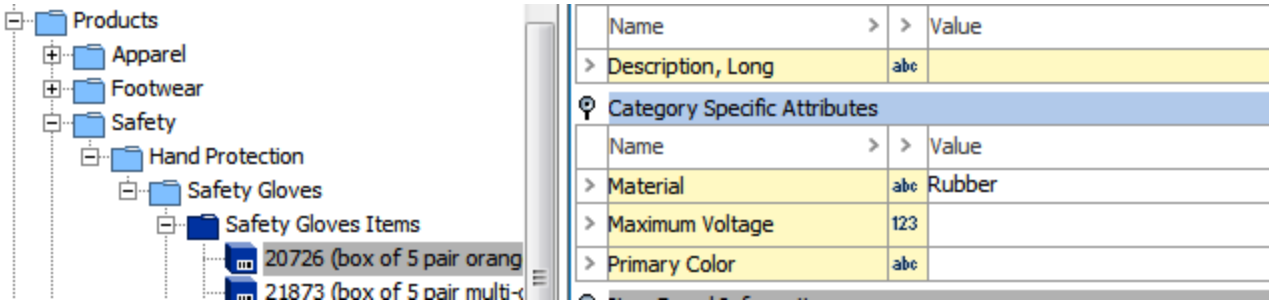


Alternatively, the attribute can be set as mandatory on an attribute link (available on product and classification super types). Attributes set as mandatory in this way behave in the same manner as described above, though at a more granular level as the indicators are only present on objects or their children where the attribute is linked. For example, a Maximum Voltage attribute is linked to an Electrical & Electronics category, and marked as mandatory on that link.



Child objects of the Electrical & Electronics category have the attribute shown as mandatory, while objects under other categories that also have the attribute linked but do not have it marked mandatory do not display any visual indicator.



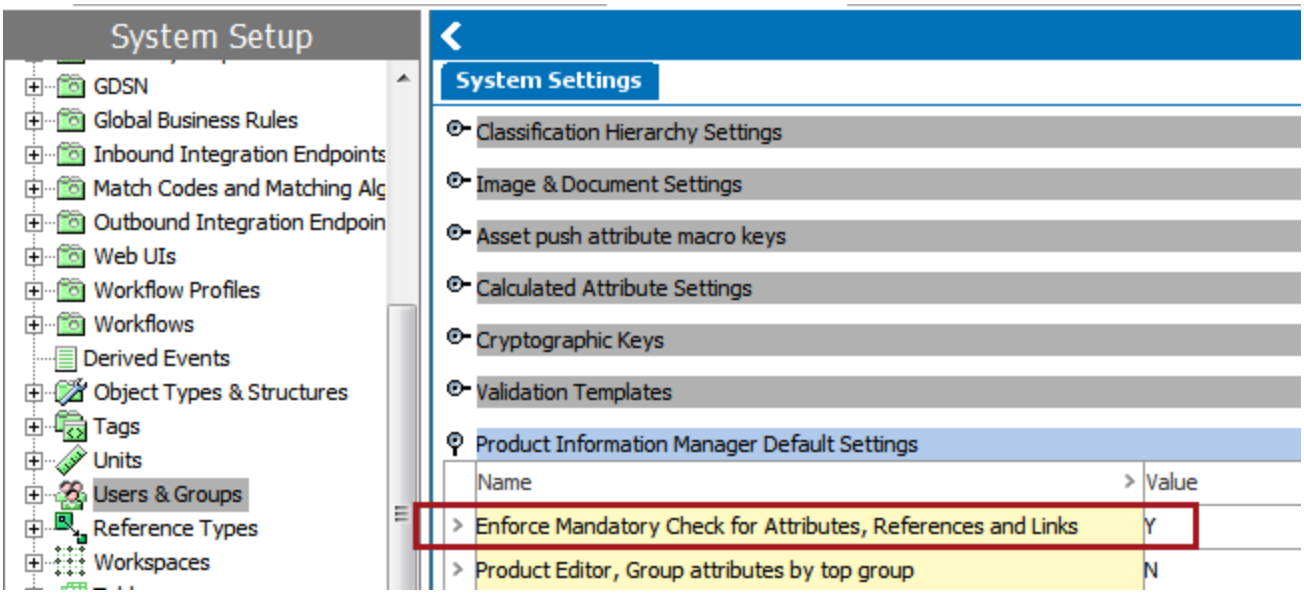


Approval Mandatory Attributes

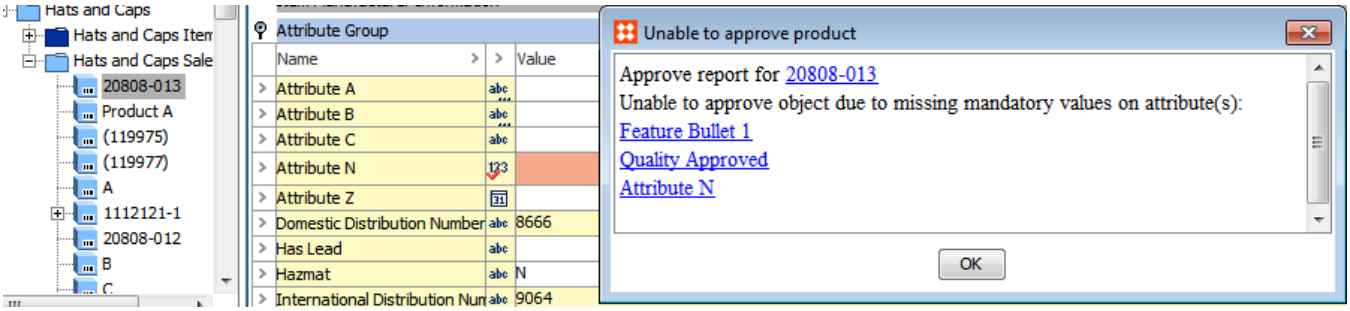
In addition to visual indicators, the system can be configured to *enforce* that the attribute is populated prior to the attribute being approved. This option is most often used only as a final "safety check" and is applied only to attributes that are required for ALL objects of a particular type or category, with no exceptions. As approval is usually a final step in a larger process, it is more common to enforce that values be present at the time that they are actively populated (e.g. within workflows or imports). Additionally, this setting is global and oftentimes mandatory requirements are more granular than that.

For example, if all 'Item' object types require a UPC or GTIN, attributes for those could be made mandatory. Note that validity is taken into account, so the mandatory requirement is only enforced for the object types on which the attribute is valid. Alternatively, consider an 'Address' object type. It might make sense to make an attribute mandatory that holds the street name, but not zip code as not all countries use zip codes.

To configure this, the visual indicators must be configured as described above (e.g. relevant attributes must be set to Mandatory=Yes). In addition, a system setting must be configured to enforce the mandatory checks prior to object approval. To do this, navigate to System Settings > Users & Groups > Product Information Manager Default Settings > **Enforce Mandatory Check for Attributes, References and Links** and set the value to **Y**.



With the above setting in place, the system will display an error to any user attempting to approve an object with one or more unpopulated mandatory attributes. The below shows an example of the error in workbench, as encountered when a user attempts to approve an object. A similar error is shown in Web UI upon attempted approval, though note that visual indicators are not present unless configured on the screen the user is approving from. See the Web UI Screen Mandatory Attributes section below for more information on configuring visual indicators for mandatory attributes in Web UI.



The above settings have no impact on mandatory attributes in Smartsheets as Smartsheets do not contain an approval action. In addition, attributes marked as mandatory for approval do not have any visual indicators in Smartsheets. See the Smartsheet Mandatory Attributes section below for more information on this.

Business Rules for Mandatory Attributes

Business rules can be used to enforce complex and/or granular use cases for specifying mandatory attributes. Whenever possible, standard process-based mandatory settings should be used (e.g. workflow mandatory or Web UI screen mandatory, described below), but when those options are insufficient for the requirements, business rules can be used to close the gap. For example, a list price could be required only if the object has a reference to a web classification. Note that workflow mandatory attribute functionality optionally includes the use of business rules so that attributes are only made mandatory under specific conditions. However, the mandatory enforcement is only applied as part of a workflow, so if it is needed elsewhere (e.g. upon approval, import, or save in Web UI), standard business rule functionality may be necessary.

For more on business rules, see the **Business Rules** documentation.

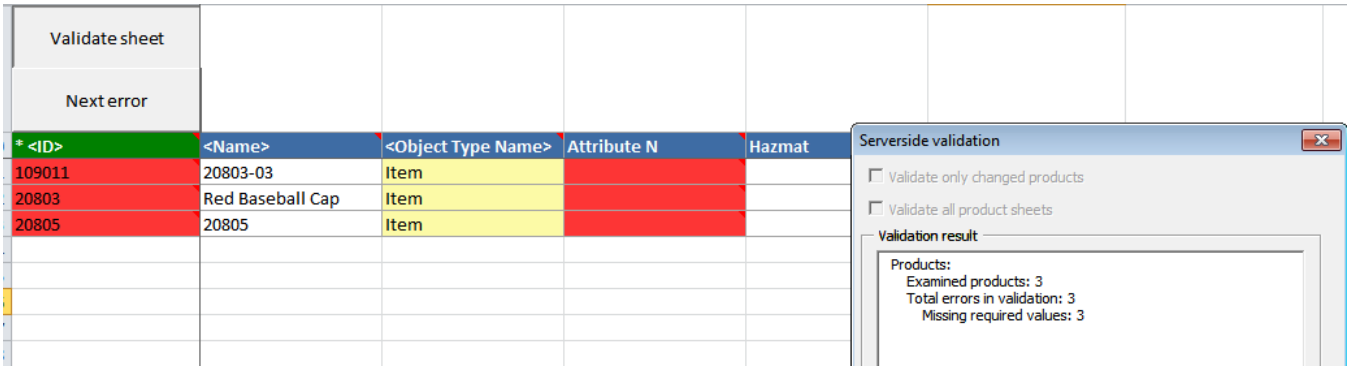
Smartsheet Mandatory Attributes

Attributes can be made mandatory within a Smartsheet, providing both a visual indicator and error reporting on missing values. This is considered a process-based mandatory setting, preferred over general business rules or approval settings as it allows for specific attributes to be indicated as required at the time that they should be provided (rather than at a later approval stage).

Attributes marked as mandatory for Smartsheets have their value cells highlighted blue to prompt the user to populate the field.

* <ID>	<Name>	<Object Type Name>	Attribute N	Hazmat
109011	20803-03	Item		
20803	Red Baseball Cap	Item		
20805	20805	Item		

If the user validates the sheet and has not populated the mandatory values, an error is reported. In this case, the row indicator, as well as the missing field, are highlighted red.

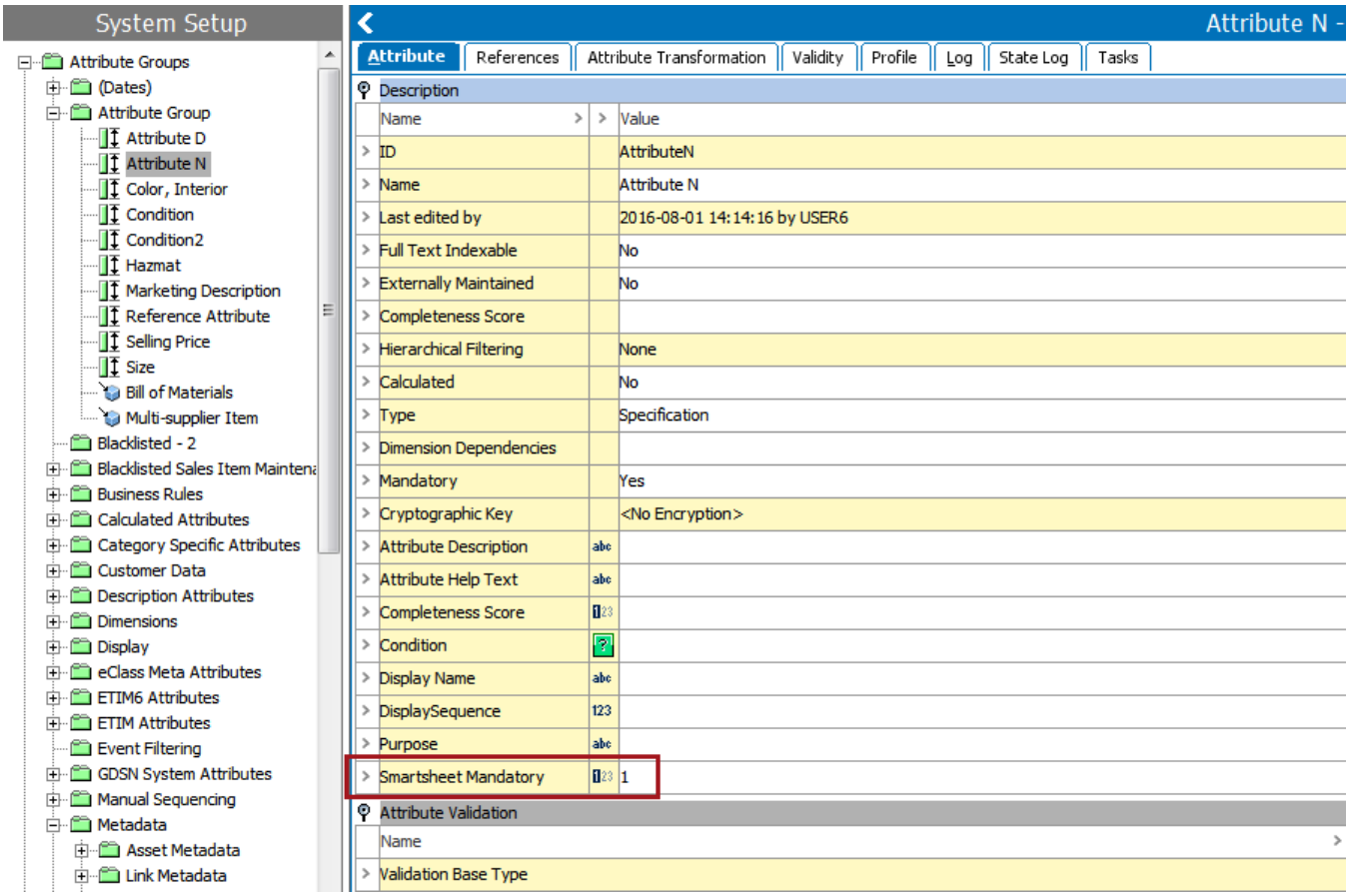


Users may still choose to import the sheet without populating the mandatory values, though this method is generally effective at enforcing population of data when users are instructed not to import until validation of the Smartsheet completes without errors.

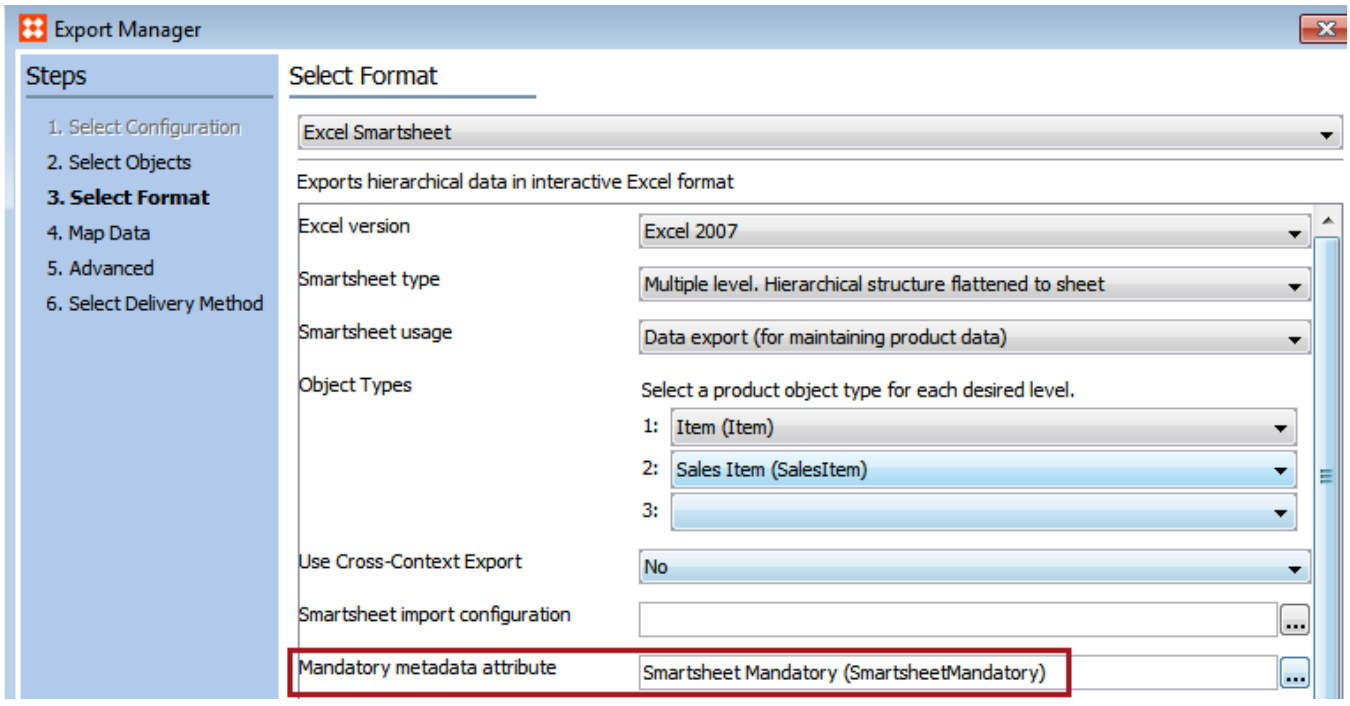
To configure this functionality:

1. Create a Description attribute with a validation base type of Integer and make it valid on the attribute basic object type.

   Details on how to make an attribute valid as metadata on an attribute are available in the Attribute Metadata on Attributes topic. The end result should resemble the below, where there is a metadata attribute called Smartsheet Mandatory that is available on Attribute N.



2. Select the metadata attribute in the Smartsheet export configuration.



More information on configuring Smartsheet exports is available in the **Creating Smartsheet Data and Template Configurations in the Export Manager** topic in the **Smartsheets** documentation.

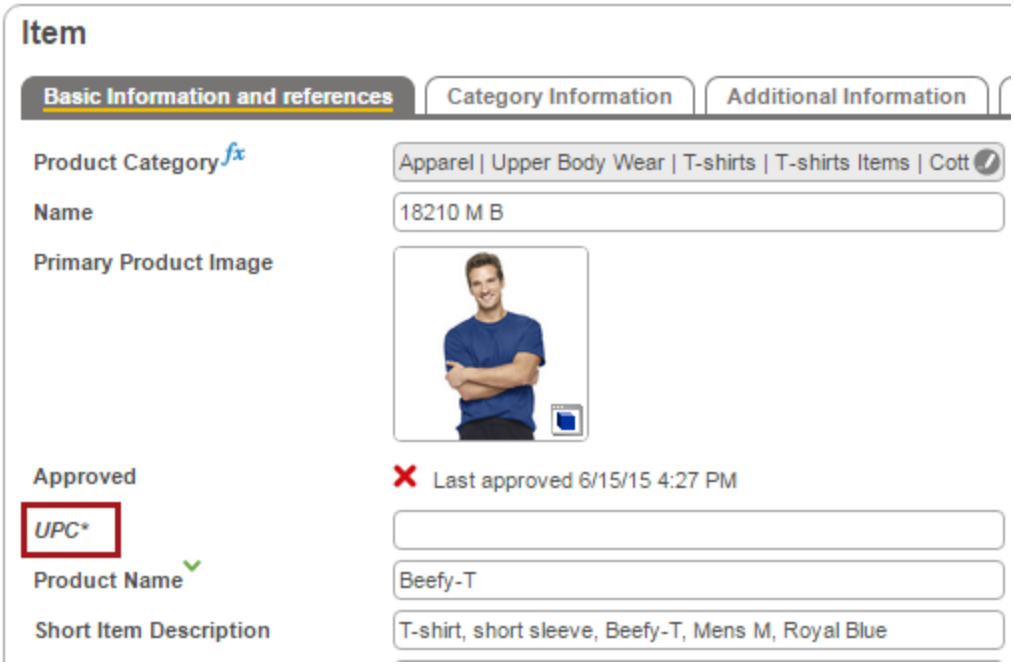
3. For any attribute that should be made mandatory in the Smartsheet, a non-zero integer must be populated in the metadata attribute. For example, as shown in step 1, a metadata value of '1' will cause the attribute to be made mandatory in the Smartsheet. Note that the value itself is not used for anything so entries of 1, 10, and 100 are equivalent as the system is concerned only with the presence / absence of a non-zero value.

Web UI Screen Mandatory Attributes / Visual Indicator in Web UI

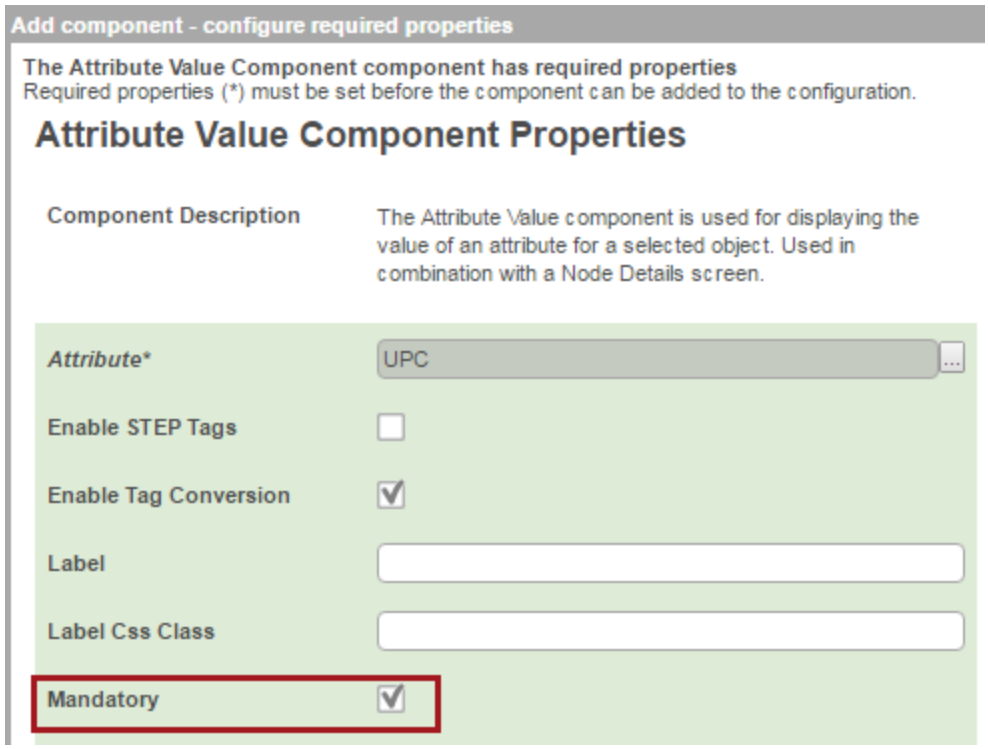
Attributes can be made mandatory in Web UI on a screen by screen basis, providing a visual indicator that data should be populated, as well as an optional enforcement of the setting that prevents users from saving or submitting data until mandatory attributes are populated. This provides a more granular level of enforcement than global approval-mandatory attributes, and when configured for single-use screens, is considered a process-based mandatory setting. However, when screens are shared, this option has larger scale implications that should be considered. It is also important to be aware that an attribute may be marked as mandatory on one screen, but not on another. Therefore, even when mandatory attribute population is enforced on a particular screen, that does not equate to a global mandatory setting.

Web UI Visual Indicators

Mandatory attributes in Web UI appear in italics and are followed by an asterisk.



To configure attributes to be mandatory, simply check the **Mandatory** parameter in any Attribute Value or Attribute Value Group component.

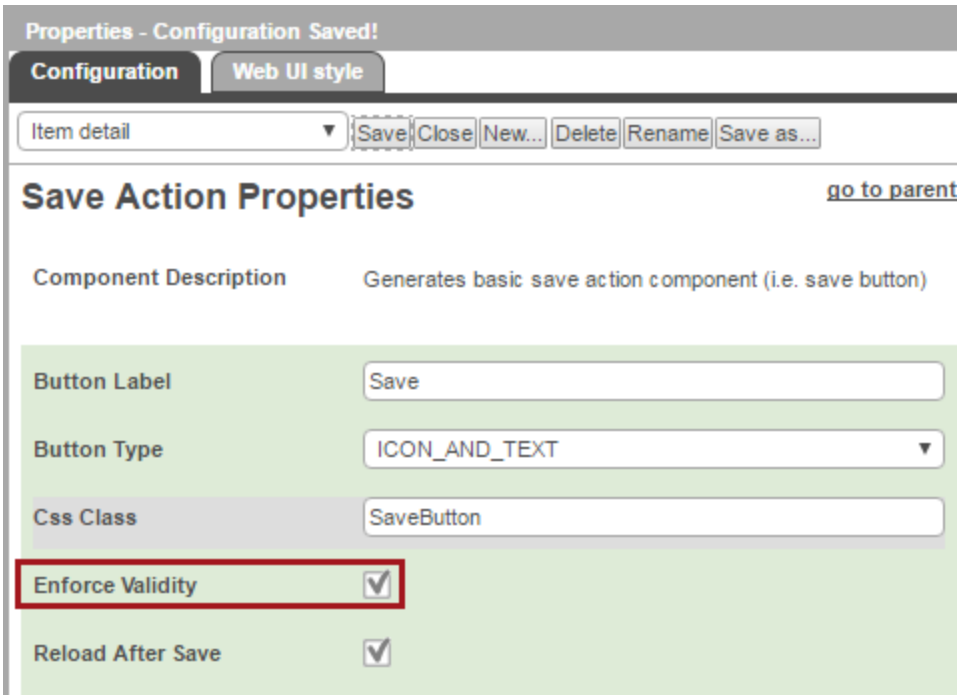


This setting alone does not enforce that the attribute must be populated - it simply provides a visual indicator to prompt the user to provide a value. To enforce population of the value, additional configuration is required, which is described below.

More information on the Attribute Value and Attribute Value Group components is available in the **Using a Web UI** section of the **Web User Interfaces** documentation.

Enforcement of Mandatory Attributes

In order to enforce that mandatory attributes are populated on the selected screen, the attributes must first be set as mandatory (as described above). In addition, any actions available on the screen that should enforce that mandatory attributes are populated must have the **Enforce Validity** parameter selected. Typically this is applied to Save, Submit, or Approve actions.



When Enforce Validity is selected, the relevant action button will not be enabled until all mandatory attributes are populated. It is important to consider whether or not this is truly desired before making this selection. For example, when Enforce Validity is applied to the Save action, users cannot save *any* data until all mandatory attributes are populated. In addition, the implications for shared screens should be considered. For example, if the same screen is reused for both onboarding and maintenance of objects, different data may be required at different times. However, both the mandatory and enforcement settings are applied to the screen itself, not to particular instances where the screen is used, and therefore it cannot be differentiated to make an attribute mandatory in only one of the cases. This situation can be managed by creating separate Web UI screens for the processes, or by having the screens tied to workflows and using the workflow mandatory settings (described below).

Workflow Mandatory Attributes

It is often necessary that certain attributes or attribute groups be set as mandatory for a state or transition in a STEP Workflow. This means that the object can pass further in the workflow only if the indicated attributes have values. To facilitate this, the STEP Workflow Designer includes an option to specify mandatory attributes or groups on both states and transitions, requiring that the indicated attributes are populated prior to the task exiting the state, or taking a particular transition. In addition, attributes and groups can be set to be conditionally mandatory, meaning they are only required if specific conditions are met.

Workflow mandatory attributes are one of the most common ways to setup mandatory attributes. The primary benefits to using this functionality are:

- Designers are given the freedom to make the right data mandatory at the right time (as opposed to using global or approval-based settings)
- Mandatory settings are configured a single time in the STEP Workflow Designer and take effect across all interfaces (workbench, Web UI, and Smartsheets), without requiring any additional configuration

- Functionality includes both visual indicators and user-facing messages at the time of enforcement (e.g. when a task is submitted from a state)

Essentially, workflow mandatory attributes provide a combination of the functionalities described above as they provide both visual indicators and enforcement, and do so across workbench, Web UI, and Smartsheets, as well as allowing for attributes to be conditionally mandatory in all of these interfaces. Because of this, workflow mandatory is the preferred method for handling mandatory attributes. However, this method is of course only available when data is being populated within a workflow so when this is not the case, the other mandatory attribute settings described in this topic can be used.

Detailed information on the configuration and use of workflow mandatory attributes is available in the **Mandatory Attributes in Workflows** section of the **Workflows** documentation.

Single and Multi-valued Attributes

To enter a value for an attribute on an object, navigate to the desired object in **Tree** and select the value field of the relevant attribute.

Note: A value applied to a higher level product, such as a product family, will be shared by its sub products via inheritance.

A Multi-valued attribute is an attribute that can contain more than one value for a specific object.

The following describes the single and multi-valued attributes:

Value Type	Description	Example
Single-valued attributes	Single-valued Attributes can hold one value for a specific object.	A specific Product, such as a “Battery” with order code “ABC-123”, can only have one value for the Attribute “Casing Style”, e.g. “AAA”.
Multi-valued attributes	Multi-valued Attributes can hold more than one value for a specific object.	A specific Product, such as an “Anti Virus” software package with order code “FGH-456” can have one or more values for the Attribute “Platform”, e.g. “Windows NT”, “Windows 2000” and “Windows XP”. Another example is the Description Attribute: Keyword. There may be several different keywords to use for a look-up for a Classification node. For example, some Keyword values for the node: “Table Saws” might be ”Table”, ”Saw”, ”Tablesaw”, ”Bench”, “Benchsaw”, ”Crosscut”, ”Swingarm”, etc.

Entering and Overwriting Values

To enter or edit the value of an attribute on an object, navigate to the desired object in **Tree** and select the value field of the relevant attribute.

Note: A value applied to a higher level product, such as a product family, will be shared by its sub products via inheritance.

An attribute value is the actual content of the attribute, and is always assigned to a specific object.

Examples

1. Specification Attribute: 'SeatColor' could have any of these values:
   - Red
   - Green
   - Blue
2. Description Attribute: 'Keyword' could have one or more of these values:
   - Furniture
   - Office products
   - Chairs
   - Swivel chairs
3. Specification Attribute: 'Short Description' could have any text string that meets the validation rules for the Attribute as these values:
   - High performance notebook computer
   - Wireless mouse
   - Loudspeakers for best music quality

Entering Values

There are different ways of entering values according to the validation base type of the Attribute:

- Values must be in a list of values (LOV). Options include:
  - Selecting the value from the list
  - Typing in a value

  The user can only type a value that is in the list. If the user tries to type a value not on the list, it will be rejected, and the list will appear to select from. Otherwise, the list will appear when double-clicking the field.
- Values can be in a list of values. Options include:

- Selecting the value from the list
- Creating a new value

  In this case, the user is allowed to create a value that is not in the list. The new value must abide by the validation rules of the specific LOV. To add the value to the list, the user must select the empty (blank) field in the list and then key in the new value.

- Values are 'free'. Options include:
  - Typing in a value

    Here, the user can type any relevant value in accordance with the validation rules of the specific attribute.

    See the **Validation Rules** section in the **Attributes** documentation for more information.

Note: The Attribute name can be double-clicked to view all settings in the Attribute Editor.

All value types can be single or multi-valued. See the **Single and Multi-valued Attributes** section in the **Attributes** documentation for more information.

Overwriting Values

Attribute values that are inherited from a higher level are indicated by a symbol in the icon column. When a value is overwritten, it is no longer inherited, but becomes local. This means that if the original value is edited on a higher level, the local value will not be changed. Also, a global translation will not affect this value.

Example:

The Attribute 'Seat color' has the value 'Blue' for the Product family 'XYZ123'. A sub product 'XYZ123b' has the value overwritten with 'Pale blue'. If the value 'Blue' is changed to 'Light blue' in the corresponding LOV, all Products using this value will have the new value, except the Products (e.g. 'XYZ123b') where the value has been overwritten locally.

See the **Validation Rules** in the **Attributes** documentation for more information.

Editing Attribute Values

To edit the value of an attribute on an object, navigate to the desired object in **Tree** and select the value field of the relevant attribute.

Attribute values can be edited either by selecting from a list, typing a value, or overwriting an existing value.

1. In **Tree**, select the relevant node to open the editor.
2. Depending on the object type, click the **Product** tab, **Classification tab** or **Images & Documents** tab.
3. Double click the value field of the relevant attribute and enter the new value.

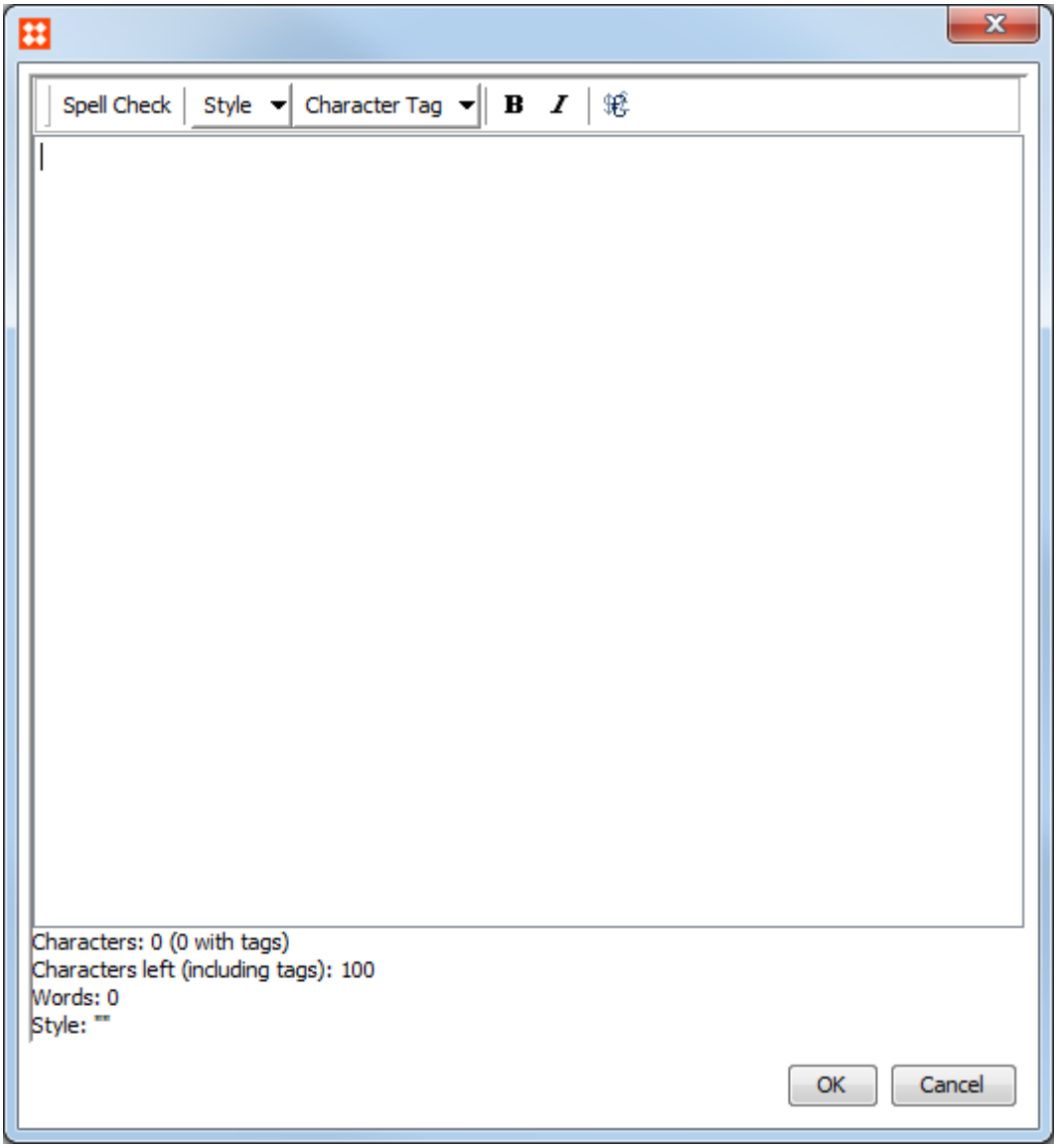
Note: Only white fields are editable.

---

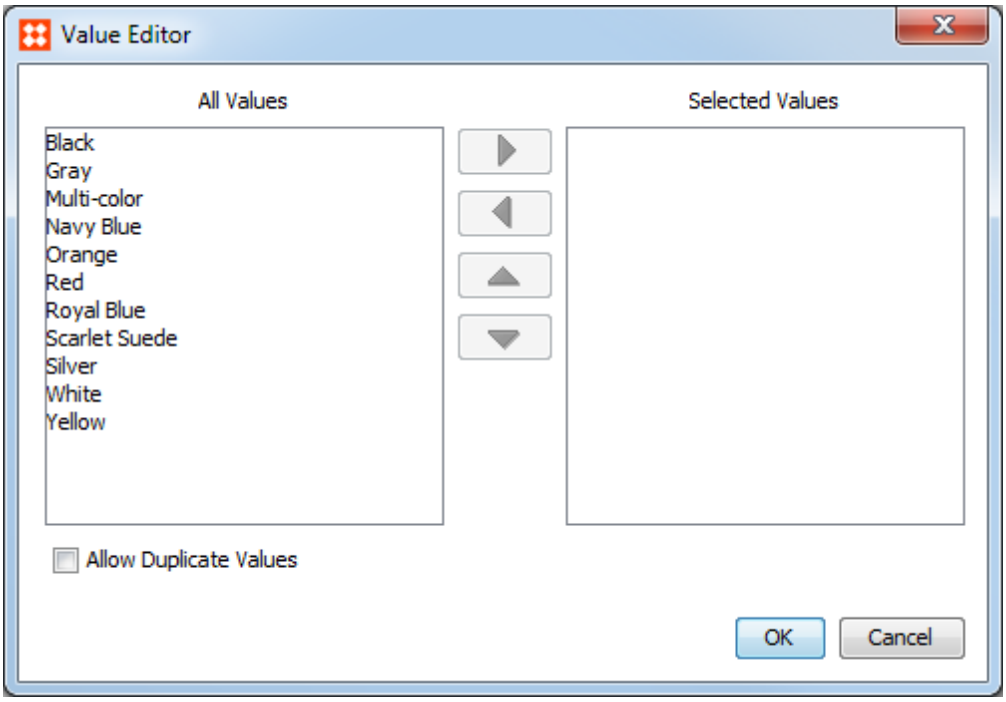
4. For additional options when editing a value field, press **Alt + F2** on the keyboard.

This opens a value editor related to the validation rules. The value editor will assist you in typing a valid value (for example, date, isodatetime or numbers with units, and so on).

Text attribute editor:



Multi-valued attribute and LOV editor:



5. Depending on the attribute, select from a List of Values (LOV) or type the value.
6. Press **F2** on the keyboard to validate the value.

The attribute value has now been edited. If the node is a product containing sub products, the new value will be shared via inheritance.

Inherited and Local Values

Values can be:

- Shared by inheritance from a higher product level (e.g. a product family)
- Local and thus added directly on the current level (e.g. a sub product)

Note: When a value is typed, the applied attribute validation rule is enforced. If a typed value does not match the validation rule, an error message will inform you of the problem.

View Filter Options

You can set various viewing options such as hiding attributes, rotating tables, and hiding attributes with empty or non-mandatory values.

Attributes with Units

Below is a description of Attributes with Units:

Attribute	Description
Attribute with default Unit	If the Attribute has a default Unit, it will appear automatically in the field after pressing <F2>.
Attribute with Units	If the Attribute has Units applied, you can either key in the relevant Unit, or press <Alt> + <F2> to select a Unit from the list.
Attribute is Numeric Text validated	If the Attribute has 'Numeric Text' as Validation Base Type, line breaks are legal. However, these should be avoided if the Attribute has Units, because the Unit will only appear on the last line.

Editing Multi-valued Attributes

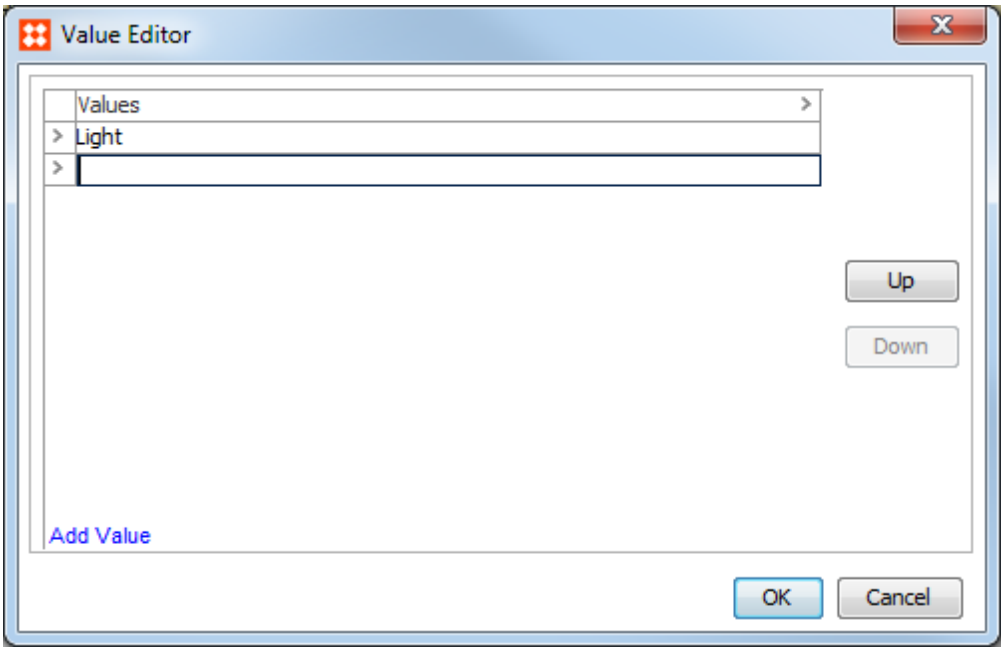
To edit the value of an attribute on an object, navigate to the desired object in **Tree** and select the value field of the relevant attribute.

A Multi-valued attribute is an attribute that can contain more than one value for a specific object.

Note: Multi-valued attributes are available for specification and description attributes.

1. In **Tree**, select the relevant node to open the editor.
2. Depending on the editor, click the **Product** tab, **Classification tab** or **Images & Documents** tab.
3. Double-click the value field of the relevant attribute.

   A **Value Editor** dialog box appears.



4. Click **Add Value** to add a new line.

   Note: To delete a value, highlight it and press **Delete** on the keyboard.

5. Type a new value.

   Note: It is possible to select in which order the values of the multi-valued attribute should appear. The order can be changed by using the buttons **Up** and **Down**.

6. Click **OK**.

   The values of a multi-valued Attribute have now been edited. If the node is a product containing sub products, the new values will be inherited to them. If it is a sub product, the values will be local and only apply to this Product.

Deleting Attributes from Objects

If an attribute is linked to the wrong object or is no longer relevant to the object, the link can be removed.

Note: Globally applied Attributes displayed in the **Description** fields cannot be deleted from an Object.

If an attribute with a value is deleted from a product node, the attribute becomes an 'Orphan Attribute', and will keep values for the attribute. If the attribute is linked to the product again, the value will also appear again.

Attributes can be deleted from objects from two different locations:

- In **System Setup**, open the relevant attribute. Navigate to the **References** tab and locate the desired object under the **Valid in Classification** field or **Valid in Products** field.
- In **Tree**, open the relevant object. Navigate to the **References** tab and locate the desired attribute from the **Linked Attributes** field.

Delete Object from Attribute References Tab

1. In **System Setup**, select the relevant attribute to open the editor.
2. Open the **References** tab where the linked objects are listed.
3. Click the **[->]** next to the desired object, and then click **Remove**.

Delete Attribute from Object References Tab

1. In **Tree**, select the relevant object to open the editor.
2. Open the **References** tab where the linked attributes are listed.
3. Click the **[->]** next to the desired attribute, and then click **Remove**.

The Attribute has now been deleted from an object. If the object has any linked products, the attribute will also disappear from those. If the deleted attribute has a value for the product, the attribute will still be visible as an orphan attribute, which is indicated with the attribute name in italics.

Important: Before deleting an Attribute from an Object, the user must make sure that the Attribute no longer needed. When the link to the Attribute is removed from a node in the Product or Classification hierarchy, it will also disappear from the nodes below that have inherited the link to the Attribute.

Specification Attributes

A specification attribute is a specific characteristic about a product. This could be a physical specification or an immaterial feature:

The total number of specification attributes within the enterprise can be rather large, and not all specification attributes are relevant to all products.

This issue is accommodated by 'linking' the specification attributes to the relevant nodes in the classification or product hierarchies. The specification attributes are then available only to products linked to (or below) these nodes in the hierarchy.

Specification	Example
Physical specification	• Color • Length • Weight • Type of material • Capacity
Feature	• Price • Country of manufacturer

Each of these examples defines a specific characteristic for a relevant set of products.

Specification attributes are only available to products (compared to description attributes, which also are available to classifications, images, documents and link types).

Inherited Attributes

Any specification attributes linked to a node in the product or classification hierarchy are also available by inheritance to the nodes below.

Before linking a specification attribute, the user should consider the level they want to link it to.

- Is the specification attribute relevant only to a single product or product family?
- Is the specification attribute relevant for a whole range of products linked to a classification node?

The availability of specification attributes and the corresponding attribute values are then subject to inheritance according to the following scheme:

Object	Value Inheritance
Product hierarchy	Attributes linked to a product node are available for all sub products linked to (or below) that node. Values applied to these attributes for a specific product will be inherited by its sub products.
Classification hierarchies	Attributes linked to a classification node are available for all products linked to (or below) that node.

Orphan Attributes

An orphan attribute is a specification attribute that contains a value for a specific product, but no link to that product through the hierarchy. Orphan attributes are maintained in the **Tree** from the list of attributes in the Product Editor on the Product tab.

An attribute can become orphaned in several ways:

Cause	Description
Product is moved	A product may have been moved to another node in the product hierarchy, thereby losing the link to its attributes
Product is imported	A product may have been imported with its attributes and values from another system, and the product is not yet classified.
Attribute link is deleted on higher level	A product has an inherited attribute with a specific value, and the inherited link to the attribute has been deleted. As the attribute has a value, it is not removed from the product, but it has become an orphan attribute.

Note: In the Product Editor, orphan attribute names are displayed in italics. Orphan attributes can link the attribute to a relevant node in the product or classification hierarchy.

Validation Rules

Validation rules are set in [System Setup]: **Attribute Groups** and **List of Values**.

A validation rule defines the allowed content of a specific Attribute or List of Values.

A validation rule can include the following rule options:

Validation Rule	Description
Validation Base Type	The basic data type, such as text, integer, number, date or List of Values (LOV). More details in the table below.
List of Values	If Validation Base Type is “LOV”, a reference to the actual LOV is made here.
Multi Valued	Multiple attribute values on a single attribute are allowed / not allowed.
Mask	An input mask defines the pattern of characters allowed when entering a value.
Min Value	E.g.: “5”: means that the minimum numeric value is 5.
Max Value	E.g.: “7”: means that the maximum numeric value is 7. Using these Min / Max examples, the value must be between 5 and 7.
Maximum length	Only for textual data. E.g. “20”: means that a maximum of 20 characters is allowed. Note: The maximum length is validated including use of tags and inline reference tags. E.g.: If you style a text bold, then the characters in the style tags <bold> and </bold> are also counted toward the maximum length.

Note: The validation rule options will change according to the selected Validation Base Type.

In the editors the validation base types are indicated by icons as shown below.

Validation base type	Icons	Description	Example
Date (See also		Allows a date to be entered according to a defined input mask, indicating the valid pattern of the value.	The mask is:

Validation base type	Icons	Description	Example
isodatetime)			DD-MON-YYYY. A valid date is e.g.: 15-JAN-2008
isodatetime		Allows a date to be entered according to a defined input mask, indicating the valid pattern of the value.	The mask is: YYYY-MM-DD HH24:MI:SS A valid date is e.g.: 2008-01-15 12:15:40
isodate		Allows a date to be entered according to a defined input mask, indicating the valid pattern of the value.	The mask is: YYYY-MM-DD A valid date is e.g.: 2008-01-15
embedded_ number		Allows a number to be entered in between a prefix and a suffix. It is allowed, that the number has a unit. It is not allowed, that prefix and suffix start with a number.	The mask is: Prefix Number (Unit) Suffix. A valid embedded_number for a Receiver could be: Converter 20 bit Lambda
Fraction		Allows numbers including dash “-” slash “/”.	Valid values: 1/2, 1-1/2 Invalid value: 1 1/2
Integer		Allows only positive numbers.	Valid values: 5, 26, and 199. Invalid values: .250, 67.34, and 50 1/2
Number		Allows any numeric characters.	Valid values: 78, 1, .34, and 384.5. Invalid values: Fractions such as 50 1/2
LOV		LOV must be selected containing the valid values.	An LOV (List Of Values) will contain the valid values of an attribute, e.g.: • Red • Green • Blue
Numeric text		All characters (numeric and alpha) are valid. It is also possible to link a Unit.	Valid values:

Validation base type	Icons	Description	Example
			Converter 20 bit, Cable 2.2 mm
Text	abc	All characters (numeric and alpha) are valid.	
URL	URL	Allows applying a URL.	
Condition		Allows applying conditions	[AttributeID] = [Value1];[Value2]

Note: Units can be applied to Attributes of all validation base types except: Text, Date, Isodatetime, and LOV.

In the Editors, icons are used to indicate the validation rule together with the optionality and the link status for the Attribute. An Attribute with the validation base type number is used as an example below:

Icons used for single valued attributes	Description
	A single valued integer can be entered.
	A value is inherited from a higher level node. The inherited value can be overwritten.
	Attribute is set as mandatory. An integer must be entered.
	Attribute is set as mandatory. A value is inherited from a higher level in tree. The inherited integer can be overwritten.

Icons used for multi valued attributes	Description
	Multiple values can be entered.
	Values are inherited from a higher level node. The inherited values can be overwritten, and new values can be added.
	Attribute is set as mandatory. Integer(s) must be entered.
	Values are inherited from a higher level in tree. The inherited values can be overwritten, and new values can be added. Attribute is set as mandatory.

Input Masks

Input masks are a part of the Validation rules that are set in **Attribute Group** and **List of Values** in [System Setup].

An input mask sets a valid pattern for a value.

Example masks:

Example	Description
An attribute contains values for the 5+4 U.S. zip codes.	A pattern could be set that would require entering: • Five numerics • A dash (-) • Four more numerics
An attribute contains values for a unique color code.	Examples: BK (01), RD (02), Y (03), and GR (04) Here the rules may be: • The first one or two characters must be alphabetic characters which are then followed by a space. • The next character must be specifically an open parenthesis “(”. • The next two slots must be numerics only. • Then the entire value must end with a close parenthesis “)”.

The following lists the different restrictions that may be imposed on a value through the use of an input mask:

Mask	Description	Example
9	The corresponding character may only be one of the following set {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} or it can be omitted.	Valid values: 581, 68, 3 Invalid values: 2845, A45
#	The corresponding character must be one of the following set: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, <space>, -, +}	Valid values: +89 3 -6, -6, 8 44, +99 Invalid values: +344266799, AA 77
?	The corresponding character must be an alphabetic letter or may be omitted.	Valid values: APO, fb, R Invalid values: Kite, BT7, +GT
a	The corresponding character may be either an alphabetic letter, a numeric or it may be	Valid values: no, B5, r6, P

Mask	Description	Example
	omitted.	Invalid values: Kite, BT7, +GT
C	This will match any character in the corresponding position or it may be omitted.	Valid values: AAA, 5T7, 1, +45, 55% Invalid values: P3T5
&	It has the same rules as “C” but the character must exist once and cannot be omitted.	Valid values: AAA, 5T7, +45, 55% Invalid values: 1, P3T5
L	It has the same rules as “?” but the character must exist once and cannot be omitted.	Valid values: APO Invalid values: fb, R,Kite, BT7, +GT
A	It has the same rules as “a” but the character must exist once and cannot be omitted.	Valid values: no, B5, r6 Invalid values: P, Kite, BT7, +GT
0	It has the same rules as “9” but the character must exist once and cannot be omitted.	Valid values: 581, 374 Invalid values: 68, 3, 2845, A45

Static Mask Characters

If other characters are typed into the mask besides the ones mentioned above, then the value must match that character exactly and it cannot be omitted.

For example, in reference to the mask “T-99”, the value “T-1” is valid but the values “R-33” or “T=33” are not. If it is needed to use one of the above mentioned characters as a static mask character, then it is possible to release it or “escape” it from the mask rules by typing a “\” in front of the character. For example, regarding the mask “\C-9”, the value “C-3” is valid but “H-3” is not.

Changing Validation Rules

Validation rules are changed in [System Setup]: **Attribute Group**: **Attribute** tab.

It is possible to change a validation rule for an Attribute or a List of Value, if needed (see limitations below). Typically a change of a validation rule is needed, when two or more Attributes/List of Values are merged into one. Before the merging can take place, it is necessary that the Attributes and LOVs share the same validation rule.

Important: When changing a validation rule of an Attribute or LOV, it has influence on existing values (see Limitations below).

Limitations to changing a validation rule:

Object	Description
Attribute with values	When an Attribute already has values, it is only possible to change into a matching validation rule. This means that it is possible to change validation from “date” to “isodatetime”, but not “text” to “date”.
LOV with values	It is possible to change an LOV with values into a non-matching validation rule. This means that you must be very careful if changing the validation rule of an LOV.

Revision control also plays a part when changing validation rules.

Example	Description
Text to LOV	Changing an internally maintained text validated attribute to an LOV is not possible. The reason is, that LOVs are not under workspace control, and therefore you cannot specify the workspace that existing values should be taken from.
LOV to Text	Changing an internally maintained LOV validated attribute to a text is not possible. The reason for this is the same as described above.

Setting LOV Filters

Filters for LOVs are set in [System Setup]: **Attribute Group**: **Attribute** tab.

It is possible to setup filters for LOVs on:

- Attribute
- Product Hierarchy
- Classification Hierarchy

For more information about setting filters on the hierarchies, see Setting Hierarchical Filters. For more information about LOVs, see List of Values (LOVs).

When setting an LOV filter on the Attribute, it is possible to define which values are valid for a certain Attribute.

This is useful in case you want to gather the information in one place and use only one LOV containing all the values, and at the same time be able to differentiate the values available for different Attributes.

Example:

Your LOV contains the following values:

- Green
- Red
- Yellow
- Blue
- White

Your Attributes are the following:

- Attribute 1
- Attribute 2

You now want to add a filter on the LOV for Attribute 1, so that the LOV values available are:

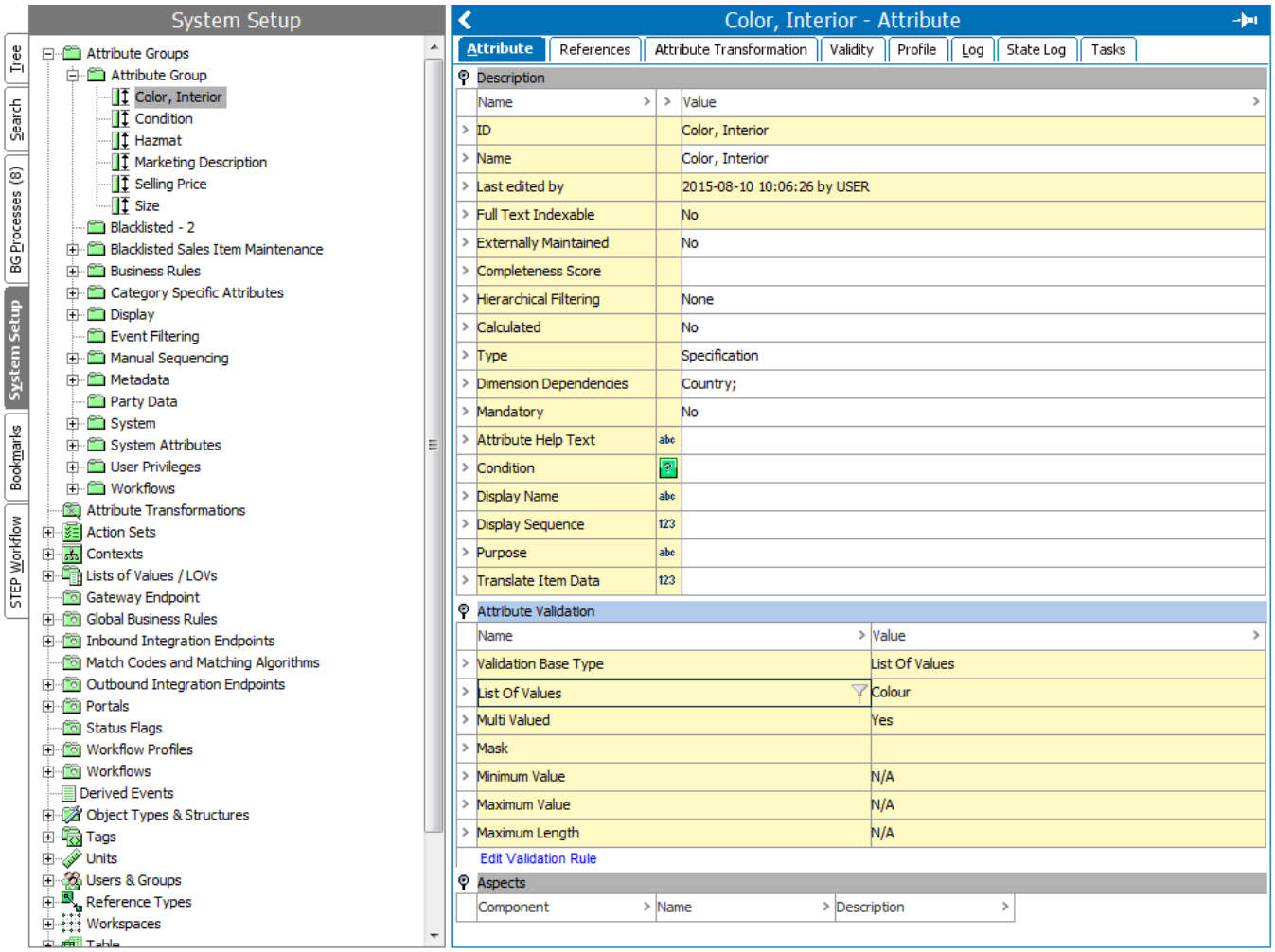
- Green
- Yellow
- White

For Attribute 2, you add a filter on the LOV, so that the LOV values available are:

- Green
- Red
- Blue

Setting LOV Filter

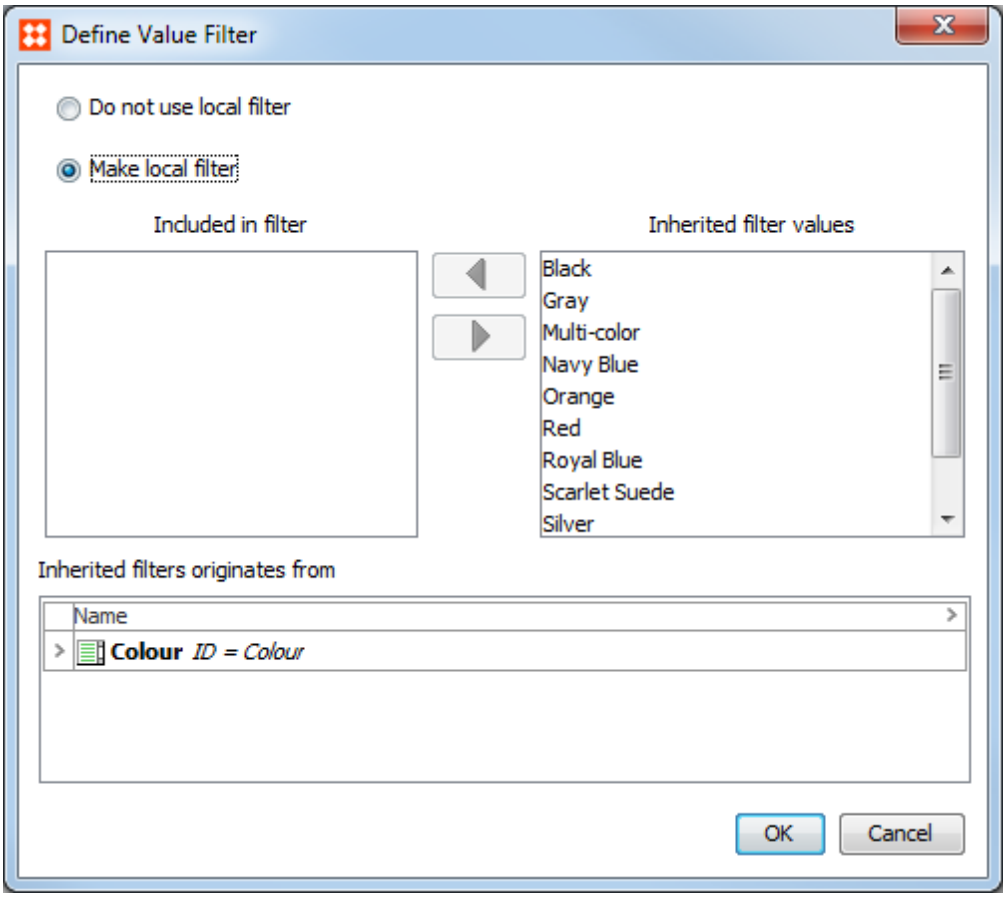
1. In **System Setup**, expand **Attribute Groups**, expand the relevant attribute group, and then select the relevant attribute.



2. Click the **Attribute** tab.
3. Under **Attribute Validation**, click the filtering icon in the **List of Values** field.

Note: To modify filters you must have privileges to maintain attributes. For more information about maintaining Attributes, see About Attribute Links.

A **Define Value Filter** dialog box appears.

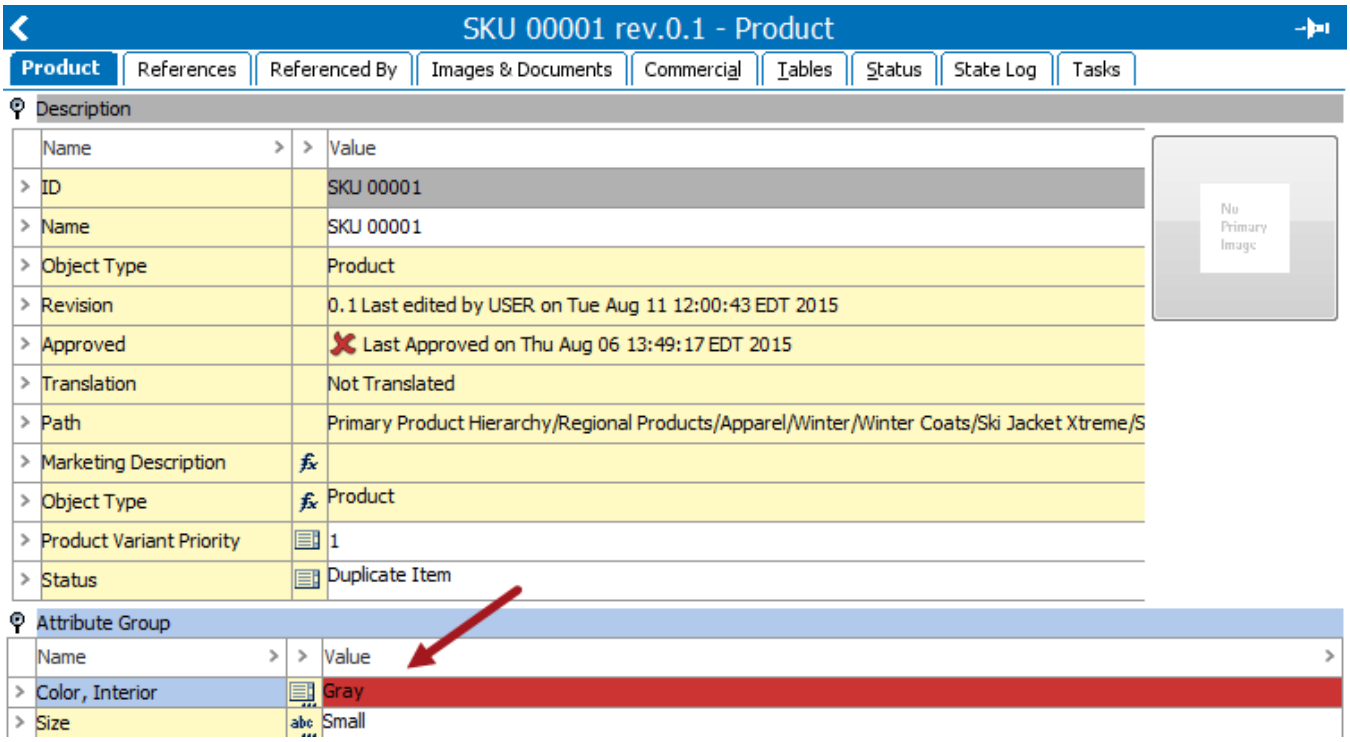


4. Click **Make local filter**, select the values (e.g. by double-clicking on the value) to be included in the filter, and then click **OK**.

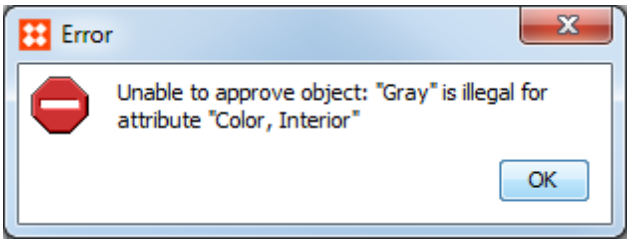
Note: To exclude values from an LOV, double-click on the value under **Included in filter**.

The defined values can now be selected on the Product / Classification / Asset in the relevant Editor, under either a Specification Attribute or Description Attribute. For more information about these Attributes, see About Attributes.

In the Editor it is possible to see if the chosen value is legal according to the filter setup. If not legal, the field containing the value is red.



When trying to approve an illegal value, a warning message will appear:



Click **OK** in the warning dialog box, then select a legal value.

Background Process Queues

This section describes how to create new queues for background processes such as import and export processes, how to assign specific background processes to specific queues, and how to determine the number of processes that can be executed simultaneously on one queue.

Default Configuration

In the default configuration, import processes have their own queue called **IN**. Export and web publisher processes share a queue called **OUT**. Scheduled processes have their own queue called **SCHEDULE**. All other background processes share a queue called **MISC**.

The **IN**, **OUT** and **SCHEDULE** queues are set up to execute one process at a time. The **MISC** queue is set up to execute two processes simultaneously.

Below is an example of the default configuration. The default configuration is a integrated part of the STEP code, therefore, it cannot be directly found in a separate file. Most entries can be verified though in http://<server>/admin/config.html. Go to **STEP WebStart**, under **Administration**, click **STEP System Administration**. Click **View Configuration**, then go to **BackgroundProcess**

BackgroundProcess.Queue.MISC.Size=2

BackgroundProcess.Queue.IN.Size=1

BackgroundProcess.Queue.OUT.Size=1

BackgroundProcess.Queue.SCHEDULE.Size=1

BackgroundProcess.ProcessType.Importer.Queue=IN

BackgroundProcess.ProcessType.Exporter.Queue=OUT

BackgroundProcess.ProcessType.WebPublisher.Queue=OUT

BackgroundProcess.ProcessType.Schedule.Queue=SCHEDULE

Changing the default configuration

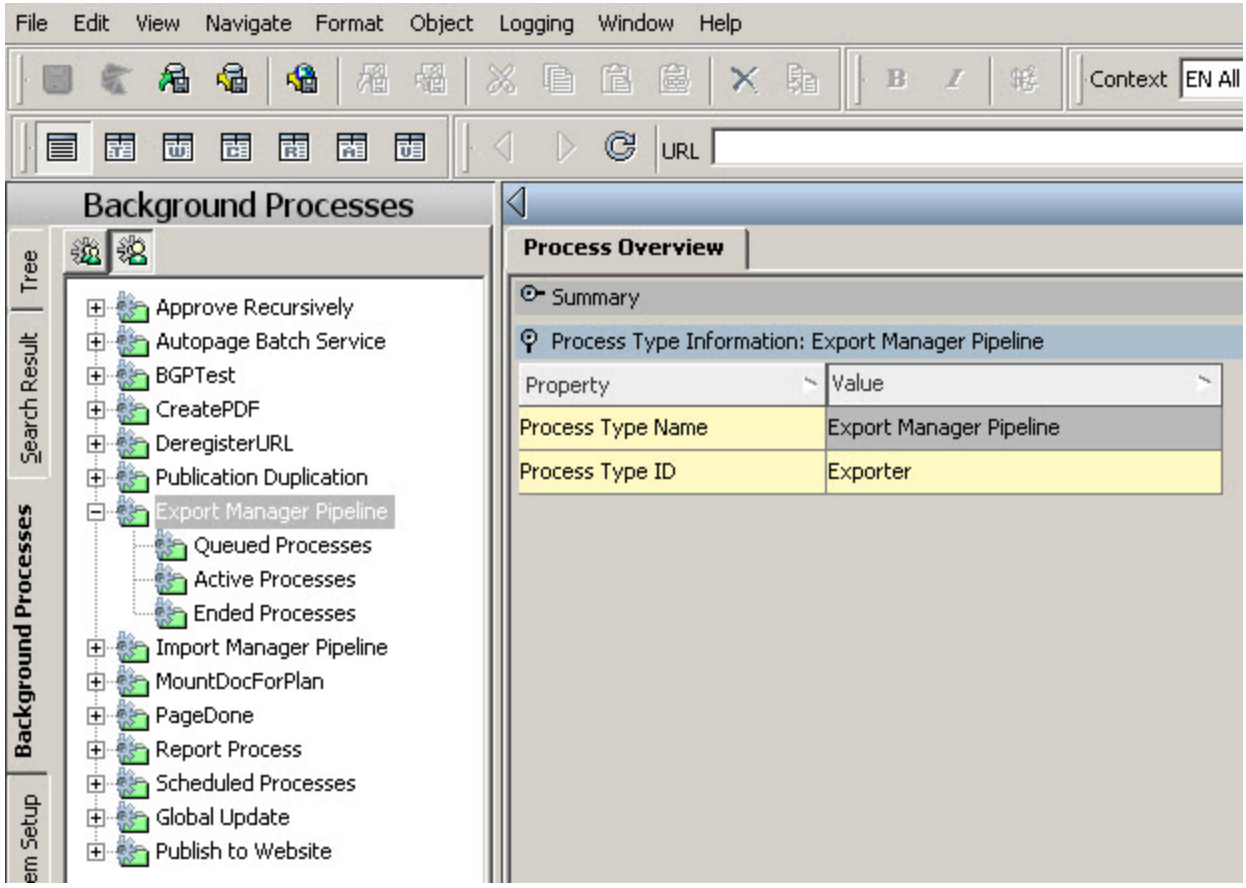
It is possible to change the default configuration.

Example: Below you can find a description of how to change the default configuration, so that export and web publisher processes can be run on separate queues. Furthermore, each queue will be set up to run two processes simultaneously.

Finding the Process Type ID

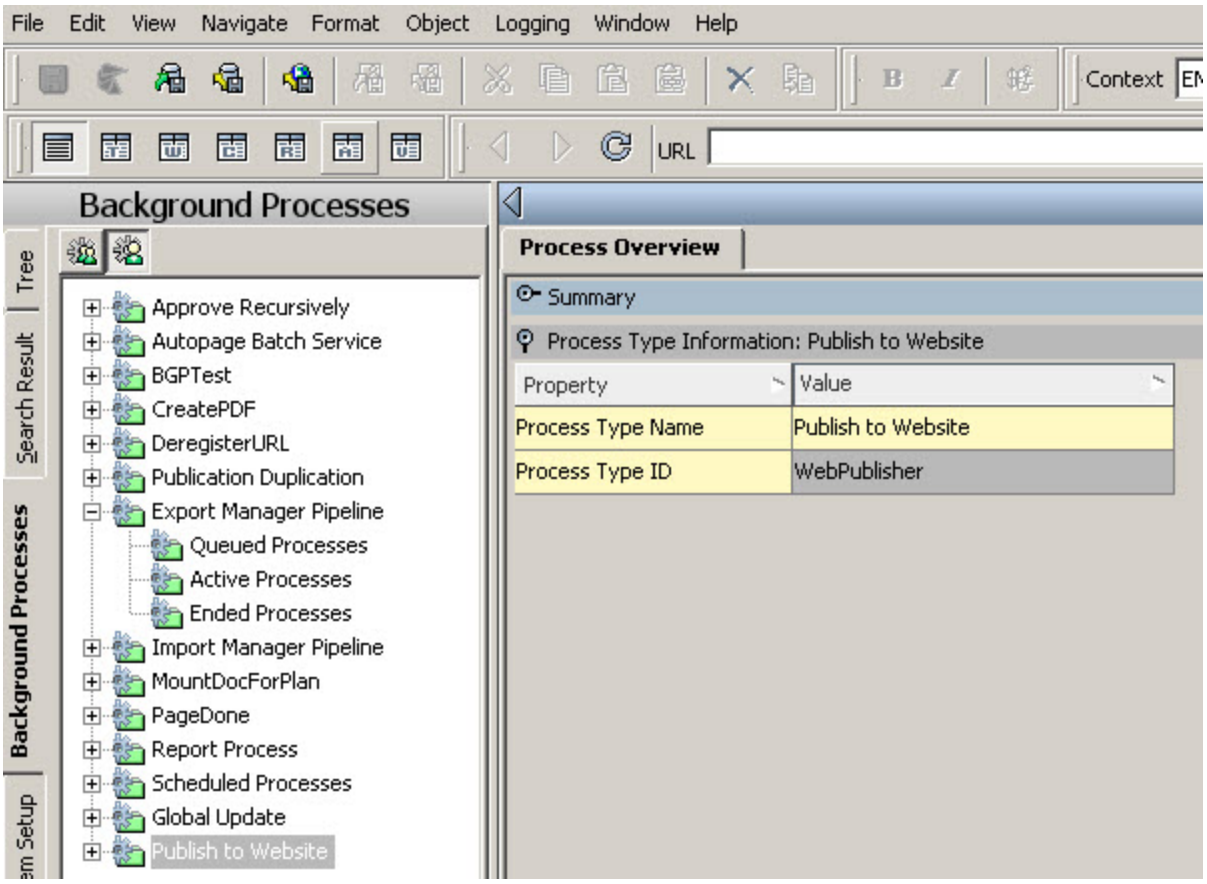
1. Click the [Background Processes] tab.
2. Click **Export Manager Pipeline**.

   A Process Overview window appears.



3. Unfold **Process Type Information: Export Manager Pipeline**
4. In the **Value** column, copy the value of the **Process Type ID 'Exporter**.'
5. Paste the value in a temporary text file.
6. Click [Background Processes], then click **Publish to Website**.

   A Process Overview window appears.



7. Unfold **Process Type Information: Publish to Website**
8. In the **Value** column, copy the value of the **Process Type ID 'WebPublisher**.'
9. Add the value to the temporary text file.

Editing the configuration file

1. Open the configuration file on the application server. The name of the file is **config.properties**.

   Note: If you run a clustered setup, you will typically have a configuration file that is shared between the application servers. Queue configurations for the entire cluster can be made in this shared properties file. Alternatively, the **config.properties** file on each individual application server can be changed.

2. Create two new queues and set the size of these queues, following this pattern: BackgroundProcess.Queue.[Name].Size=[Size]

   BackgroundProcess.Queue.EXPO.Size=2

   BackgroundProcess.Queue.WEBP.Size=2

   In this example, we have named the new Export process queue **EXPO**, and the new Web Publisher queue **WEBP**. Each queue can run two parallel processes.

3. In order to make the export and web publisher processes use these new queues, link the new queues to the relevant process types by adding two additional lines (shown in **bold**) following this pattern:

   BackgroundProcess.ProcessType.[Process Type ID].Queue=[Queue]

   BackgroundProcess.ProcessType.Exporter.Queue=EXPO

   BackgroundProcess.ProcessType.WebPublisher.Queue=WEBP

   The [Process Type ID] has been replaced by the values described in Background Process Queues

4. Close the **config.properties** file. The changes will now take immediate effect.

Note: If you run a clustered setup, and you have made the amendments in the shared file, the settings regarding the size will be applicable for each application server. Hence, if you have edited the shared configuration file as described in the example above on a cluster with two application servers, you would be able to run four exports and four web publisher processes simultaneously on the cluster, i.e. two exports and two web publisher processes on each application server.

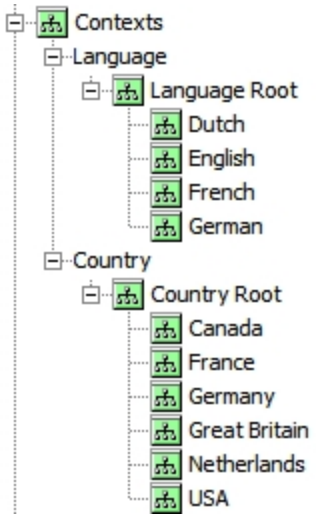
Contexts

This section describes how to set up Contexts.

About Contexts

Contexts are maintained in [System Setup]: **Contexts**.

A Context groups a set of Dimension Points. One and only one Dimension Point from each Dimension can be associated with a specific Context.



Note: When selecting a Context, the view will change in [Tree]. Only data that is valid within the selected Context will be displayed. In [System Setup], this filter is not used, and all Objects are visible. If an Object, e.g. an Attribute, has no name in a Context, the ID will be shown and indicated by '(ID)'.

Dimension Dependent Data

Dimensions are maintained in [System Setup]: **Contexts**.

Dimension Dependent data can have different variations according to the Context in which they are presented. The Context will specify the language, country, etc. for which the data is valid. For more information, see "About Dimensions" on the next page.

Some examples:

- Values that consist of alpha characters may change due to language.
- A product's warranty period may depend on the country in which it is sold.

The typical usage of Dimension Dependent data is within translation, when an Attribute can have different values within different languages. The Attribute will then have values that depend on the **Language Dimension**.

An example of an Attribute where the values are Language dependent is 'Primary Color:

- For English, the values may be 'Red', 'Blue', and 'Yellow'
- For Spanish, the values would be 'Rojo', 'Azul', and 'Amarillo'
- For French, the values would be 'Rouge', 'Bleu', and 'Jaune'

Dimension Dependent Entities

The following database entities may be Dimension Dependent:

- Names (defined per object type)

  It is possible to make the Object Name fields Dimension Dependent at the Object Type level.

  For more information about Object Types, see "" on page 307.

- Values

  The Dimension dependency for values is specified when creating the Attribute.

- Links

  Objects, as such, cannot have Dimension Dependency. However, they can have it indirectly through dimension-specific links, e.g. a product-to-product reference that is only valid in a **German Dimension**.

  For more information about Link Types, see Online Help > Link Types.

About Dimensions

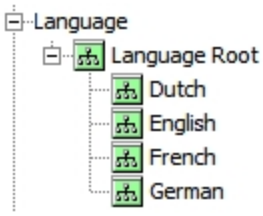
A Dimension determines the structure of Dimension Points. Common examples of Dimensions are Language Dimensions and Country Dimensions.

A Dimension Point is a named locale within a specific Dimension.

Dimension	Dimension Points
Language	• French • English • Dutch • German

Dimension Structure

Each Dimension is set up as a hierarchy of Dimension Points.

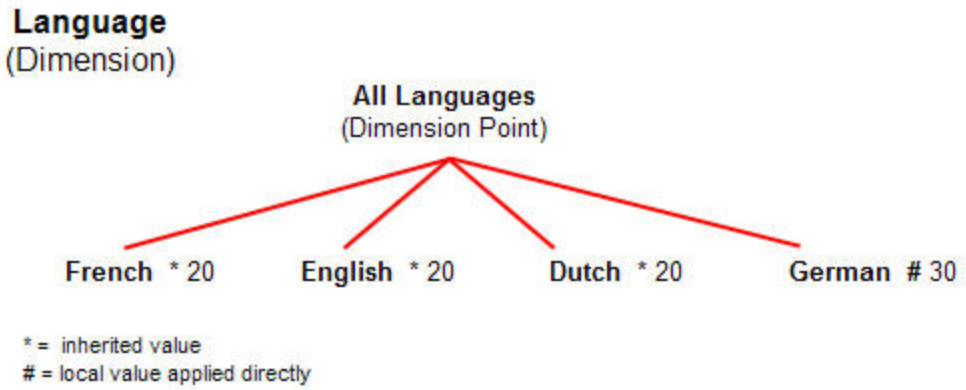


In the figure above, there are five (5) Dimension Points (**Language Root**, **Dutch, English**, **French**, and **German**) within the **Language Dimension**.

Values within a Dimension

Each Dimension is set up as a hierarchy of Dimension Points.

Below is an example of value inheritance when applying the value '20' to a 'Length' Attribute that is not language dimension dependent.



The Dimension in this example is **Language**. If values are applied to the 'Length' Attribute from a Context that includes the **All Languages Dimension Point**, these values for 'Length' are valid by inheritance for the 'child' Dimension Points, which in this case are French, English, Dutch, and German. Therefore, both French, English, Dutch and German inherit the value '20'. However, it is possible to override any value locally as shown in the figure above. The value '30' overrides '20' at the **German Dimension Point**.

Below you will find another example where the value is not inherited, but will be populated in other Dimension Points during a translation process. In this example, the value 'Blue' is applied to a 'Colour' attribute (which is language dependent) in the English context.

Before translation:

After translation:

Changing Dimension Dependency

Dimension Dependencies are maintained in [System Setup]: **Contexts**.

Changing Dimension Dependency means either adding or removing a Dimension from the list of Dimension dependencies after the Object has been created.

Note: The Dimension Dependency should normally not be changed. Be aware of possible conflicts if a Dimension Dependency is removed.

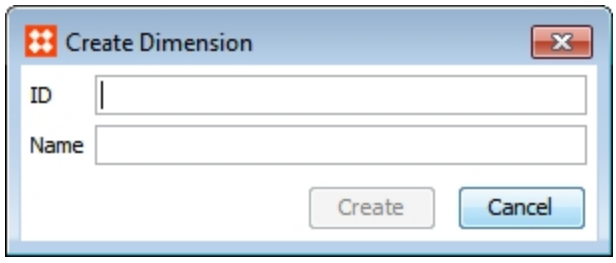
Creating a Dimension

Dimensions are maintained in [System Setup]: **Contexts**.

For more information about Dimensions, see 'About Dimensions' in the online help.

1. In [System Setup], click **Contexts**, right-click, and then click **New Dimension**.

   A **Create Dimension** dialog box appears.



2. Type the ID and Name of the new Dimension.

   Note: If ID and Name should be identical, just press **Enter** on the keyboard to repeat the ID in the **Name** field.

3. Click **Create**.

   The **Dimension** Editor appears.

   A new Dimension has now been created, and when creating a Context, it will be displayed in a column in the **Context** Editor.

Example of Creating a Dimension

A simple PIM database has been established with attribute values in one language only. Later, translation of some or all of the Attribute values is required.

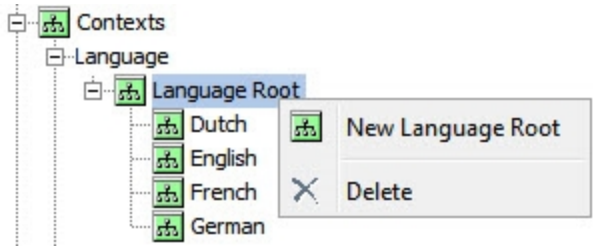
The Attributes for which translation of values make sense (textual data, but not numeric data for instance) are identified. A Dimension called **Language** is created. Within this Dimension the relevant Dimension Points are created, e.g. **English** and **German**. The 'translatable' Attributes are then changed to depend on the **Language Dimension**, and the existing Attribute values are assigned to the Dimension Point **English**.

Creating a Dimension Point

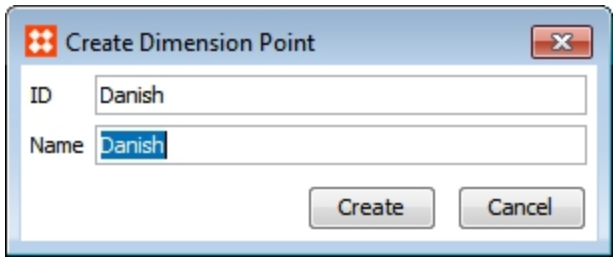
Dimension Points are maintained in [System Setup]: **Contexts**.

A Dimension Point is a named locale within a specific Dimension.

1. In [System Setup], unfold **Contexts**, click the relevant Dimension Point, right-click, and then click **New** (e.g. New Language Root).



   A **Create Dimension Point** dialog box appears.



2. Type the ID and Name of the new Dimension Point.

Note: If ID and Name should be identical, just press **Enter** on the keyboard to repeat the ID in the **Name** field.

3. Click **Create**.

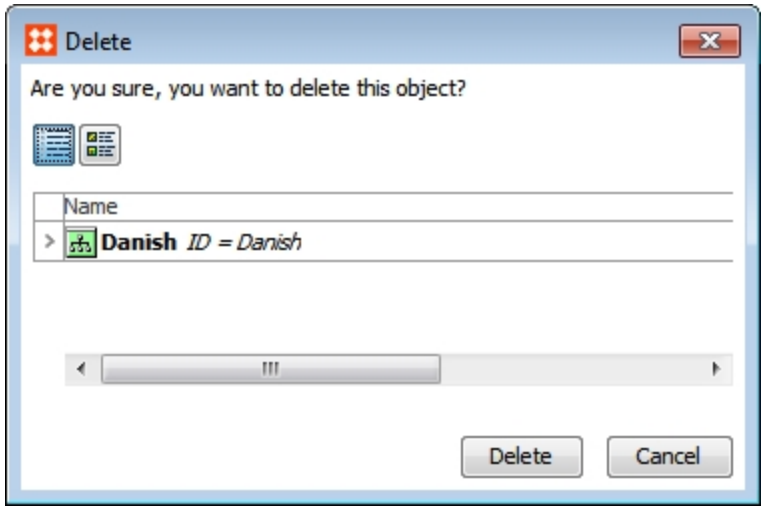
   A new Dimension Point has now been created, and when creating a Context, it will be in the list of available Dimension Points.

Deleting Dimensions and Dimension Points

Dimensions and Dimension Points are maintained in [System Setup]: **Contexts**.

1. In [System Setup], unfold **Contexts**.
2. Click the Dimension or Dimension Point to be deleted.
3. In the **Object** menu, click **Delete**, or right-click and click **Delete**.

   A **Delete** dialog box appears.



4. Click **Delete**.

   The Dimension or Dimension Point will be deleted.

Note: When deleted, the Dimension or Dimension Point is no longer available to choose for Contexts.

Important: When deleting a Dimension or Dimension Point, all data dependent on this Dimension or Dimension Point will be permanently removed from the system. The values will not reappear if the Dimension or Dimension Point is recreated.

Creating a Context

Contexts are maintained in [System Setup]: **Contexts**.

A Context groups a set of Dimension Points. One and only one Dimension Point from each Dimension can be associated with a specific Context.

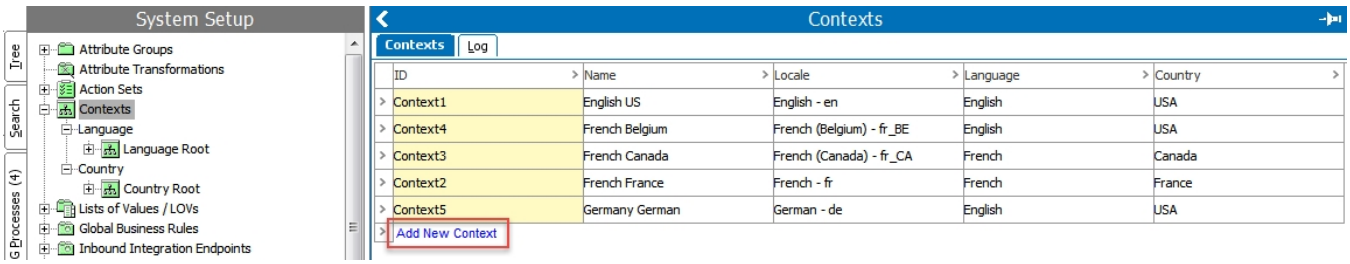
Each Context is also associated with a Locale. A Locale ensures that sorting is the same across all applications of STEP (sorting in STEP Table Component [when using Table Transformation], sorting in InDesign).

The Locale also controls decimal separator (e.g. whether a number is 1234.56 or 1234,56) when working with output to InDesign and working with STEP PIM tables. This means that numbers (attributes with a Validation Base Type of Number, Embedded Number, and Fraction) will be localized.

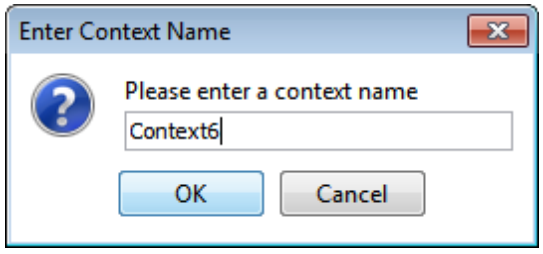
Note: Before creating a Context, the relevant Dimension and Dimension Point must be created. For more information, see "Creating a Dimension" on page 240."Creating a Dimension Point" on page 241.

1. In [System Setup], click **Contexts**.

   The **Contexts** Editor appears listing all previously created Contexts.



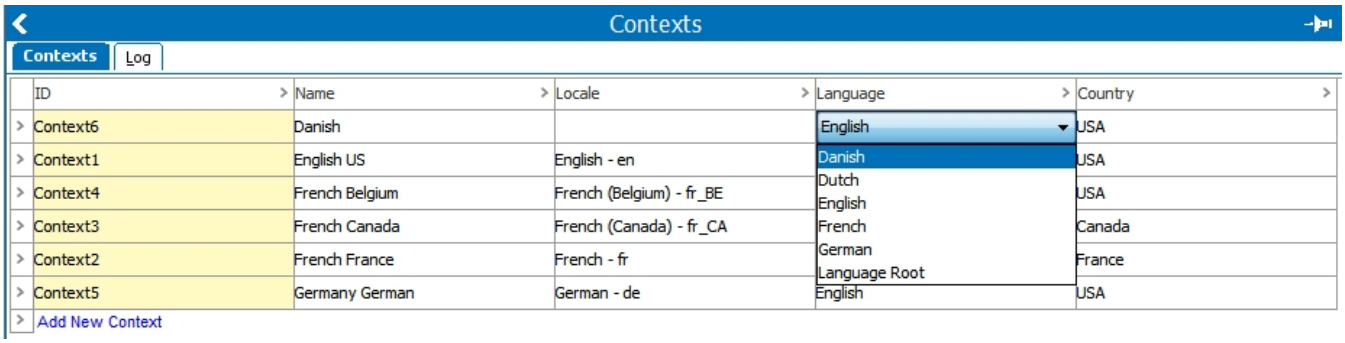
2. Under **Contexts**, click **Add New Context**.
3. An **Enter Context Name** dialog box appears.



4. Type a name for the Context, and then click **OK**. By default, the Name entered in this field also becomes the Context's ID. The Name (but not the ID) can be edited after creation.

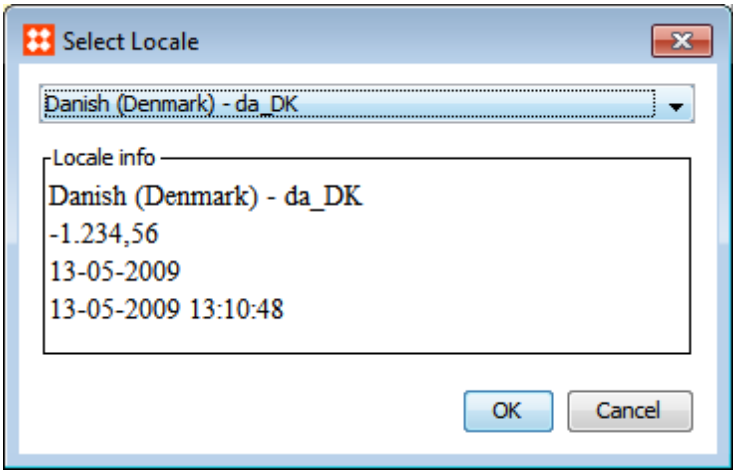
   A new default Context appears with a blank Locale, English as Language, and USA as Country.

5. Change the Name, if needed, then select the relevant Dimension Point from the dropdown list for each Dimension (in this example, Language and Country).



6. Click the **Locale** column for the relevant Context, and then click **[...]** to select a Locale to be used for Data in the specific Context.

   A **Select Locale** dialog box appears.



7. Click the relevant Locale from the list containing all supported Locales, and then click **OK**.

   Now a Context consisting of the relevant Dimension Points has been created. The Context will be in the list of available Contexts that can be selected for Objects.

For information about locales and calculated attributes, see the **Calculated Attributes** section of the **System Setup / STEP Super User Guide** documentation.

Editing a Context

Contexts are maintained in [System Setup]: **Contexts**.

A Context groups a set of Dimension Points. One and only one Dimension Point from each Dimension can be associated with a specific Context. For more information, see "Deleting Dimensions and Dimension Points" on page 242.

Note: Before editing a Context, all the relevant Dimensions and Dimension Points must be created. For more information, see "Creating a Dimension" on page 240.and "Creating a Dimension Point" on page 241.

1. In [System Setup], click the **Contexts**.

   The **Context** Editor appears listing all previous created Contexts.

2. Click the Context to be edited.
3. In the relevant **Dimension** column, click another Dimension Point.

   Now a Dimension Point in a Context has been changed or deselected.

Deleting a Context

Contexts are maintained in [System Setup]: **Contexts**.

1. In [System Setup], click **Contexts**.

   The **Contexts** Editor appears.

2. Click the Context to be deleted, right-click and then click **Remove Context**, or just press **Ctrl + <->** on keyboard.

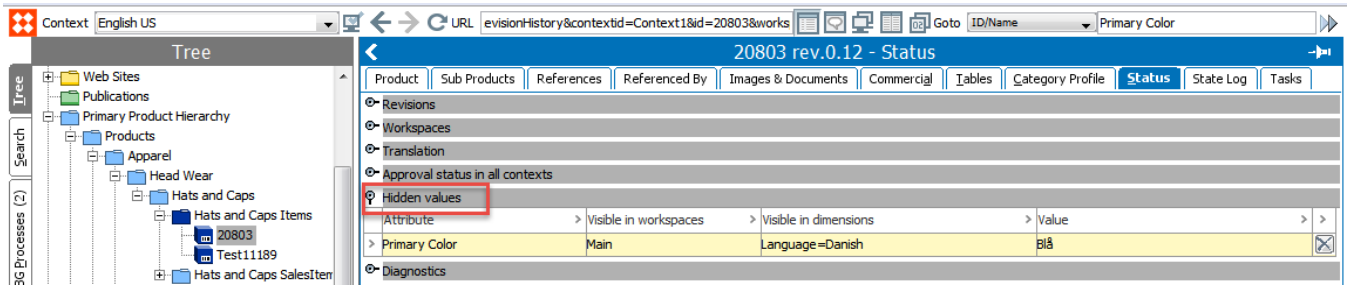
   The Context will be deleted.

   Note: When deleted, it will no longer be in the list of available Contexts.

Hidden Values

If a Context in which Attribute values exist is deleted—and the Attribute(s) are dependent on one or both of the Dimension Points contained within the Context (e.g. Language or Country)—these values will be hidden if no other Contexts use one of these Dimension Points. Hidden values can be viewed on the **Status** tab in the Workbench, beneath the 'Hidden values' flipper.

The values will reappear if the Context is added again, or if another Context is created that uses the same Dimension Point.



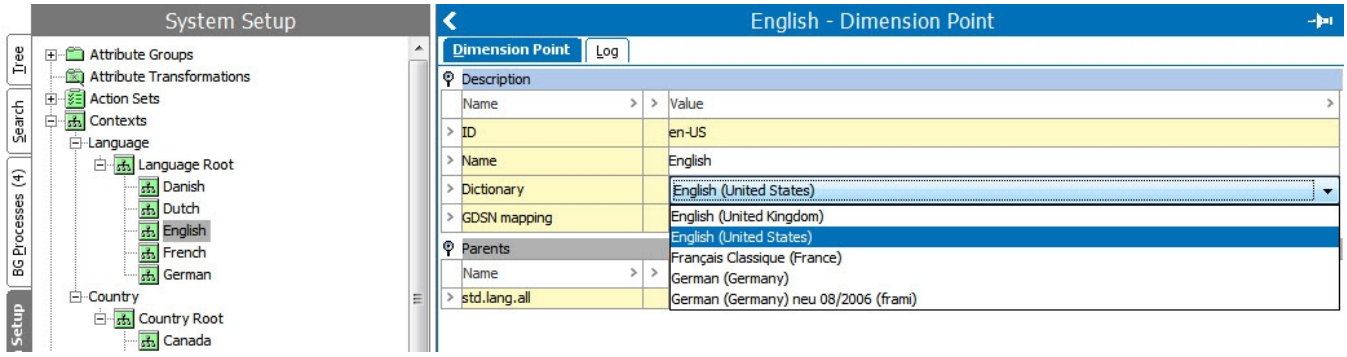
Selecting Dictionary for a Language Dimension Point

For each Dimension Point in the **Language Dimension**, a dictionary can be selected. The dictionary is used to check the spelling when values are typed.

1. In [System Setup], unfold **Contexts**, and then unfold **Language**.
2. Unfold **All Languages**, and then click the **Language Dimension Point** where a dictionary should be added.

   A **Dimension Point** Editor appears.

3. Click the **Dictionary** tab, and then click the relevant dictionary from the list.



Important: The list of available dictionaries on each STEP system depends on a license. You may find a list of dictionaries that does not correspond to the list in this example.

Note: It is possible to extend the list of available dictionaries. Furthermore, it is also possible to add new words to the dictionary.

Setting Dimension Dependency for Objects

Dimension Dependency for Objects is set in various Editors in the [System Setup].

Setting Dimension Dependency is available for a lot of Objects in the STEP PIM. The most common Objects are:

- Attributes
- Link types
- Lists of Values (LOVs)
- Tables
- Object types (Product, Classification, Asset)

For more information about Dimension Dependent Data, see "Changing Dimension Dependency" on page 240.

Setting the Dimension Dependency means that data can have different variations according to the Context in which they are presented.

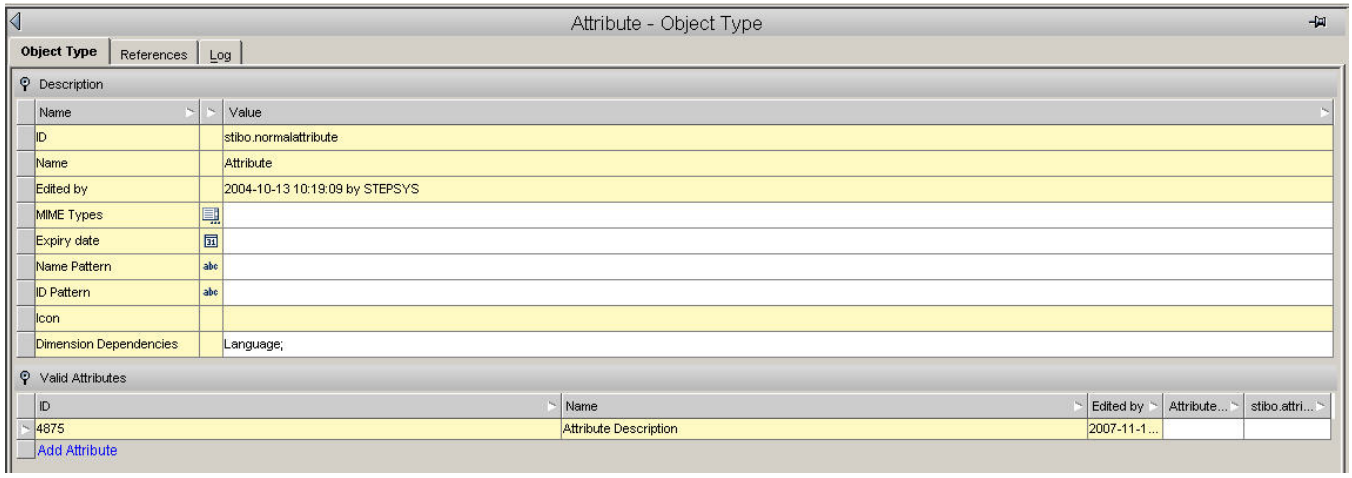
When Dimension Dependency is set for an Object, it is possible to specify the availability of the Object, e.g. to make it available only for specific Contexts. This is done by selecting the relevant Dimension in the **Dimension Dependencies** field.

The following explains the implications for each relevant Object type when setting Dimension Dependency:

Object	Description
Attributes	Attribute values become Dimension Dependent and can, for example, be translated to other languages.
Link types	When Dimension Dependency is set globally on a Link type, a [Dimension dependencies] field will appear in the Object editors in question. Dimension Points can then be selected for the Objects.
Lists of Values (LOVs)	LOV Names and Values become Dimension Dependent and can, for example, be translated to other languages.
Object types	Object Names become Dimension Dependent and can. for example, be translated to other languages. Or the Object can, for instance, have different Names in different markets. **Important:** Dimension Dependency for contents of Images and Documents is set in the [System Setup]: Users and Groups Editor. This setting is required to be able to create local versions of Images and Documents. For more information about creating local versions, see Online Help - Images & Documents.

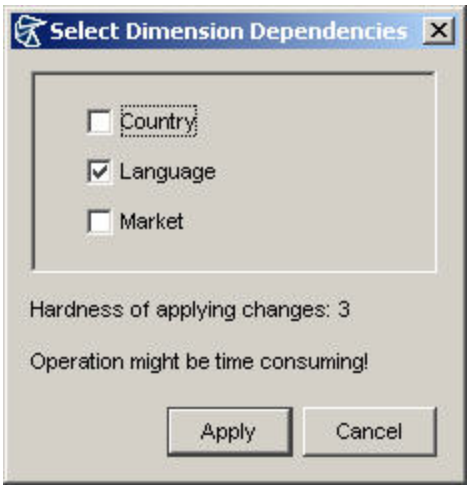
1. In [System Setup], select the Object or Object type (e.g. Attribute).

   The corresponding Editor appears.



2. Double-click in the **Dimension Dependencies** field, then click on the appearing **[...]** button.

   A **Select Dimension Dependencies** dialog box appears.



3. Check the relevant Dimension(s), and then click **Apply**.

   An Object has now been made Dimension dependent.

Removing Dimension Dependency of an Object

Dimension Dependency for Objects is set in Various Editors in [System Setup].

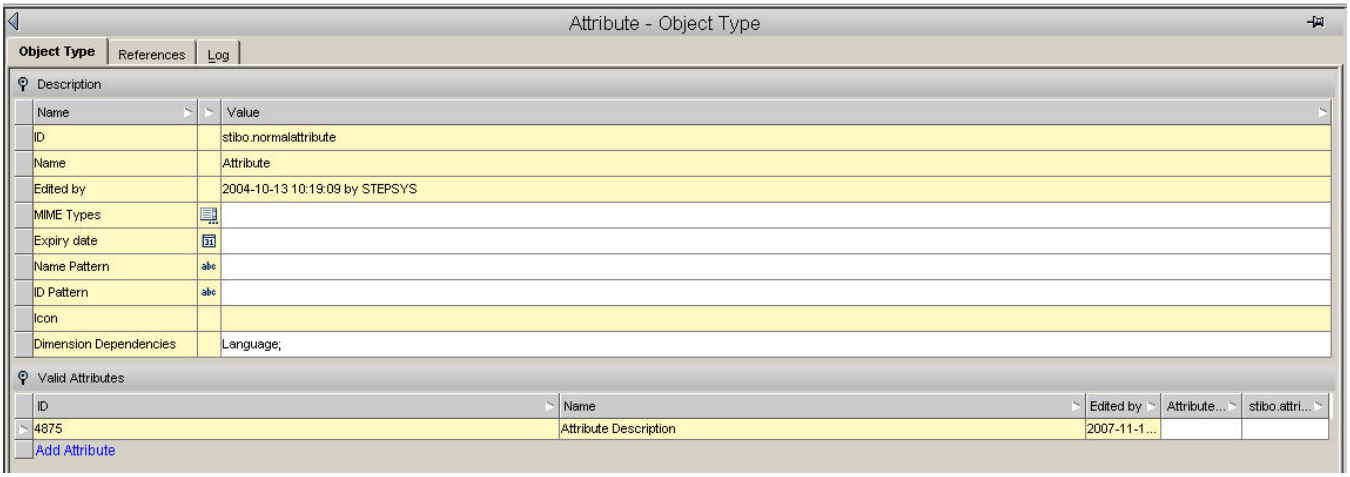
Setting Dimension Dependency is possible for a lot of Objects in the STEP PIM. The most common Objects are:

- Attributes
- Link types
- Lists of Values (LOVs)
- Tables
- Product Object Types

For more information about Dimension Dependent Data, see "Changing Dimension Dependency" on page 240.

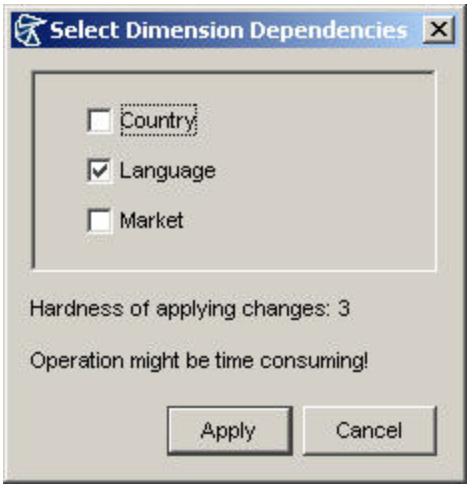
1. In [System Setup], select the relevant Object (e.g. Attribute or LOV).

   The corresponding Editor appears.



2. Double-click in the **Dimension Dependencies** field, and click on the appearing [...] button.

   A **Select Dimension Dependencies** dialog box appears.



3. De-select the relevant Dimension(s), and then click **Apply**.

Note: The dialog box will indicate the complexity of removing the Dimensions. If, for example, values exist in multiple Dimensions and Dimension Points, then it can be time consuming to remove the Dimension Dependency. Hardness 0-5 indicates the complexity of the operation, where hardness 5 is a very complex operation.

4. If there is a conflict when removing Dimension Dependency, then a dialog box appears where you must:

   - Specify the sequence of the Dimension Points for the System to use to remove the Dimension Dependency.
   - Use the **Up** button and **Down** button to specify the priority.

Note: The priority is used to specify from which Dimension Point values should be kept (when removing the Dimension Dependency). This means that if a value exists in the Dimension Point with highest priority, then this value will be kept, else the system will look at the Dimension Points with lower priority to find a value to keep.

- Click **OK**.

The Dimension Dependency will be removed from the Object.

Selecting Visibility for Objects

Selecting Visibility for Objects is done in [Tree] in the **Classification**, **Product**, and **Images & Document** Editors: **Visibility** field.

Note: The **Visibility** field is available when Dimension Dependency has been globally set for the relevant Link type. For more information, see "Setting Dimension Dependency for Objects" on page 246.

For example, making the Link type **Product hierarchy** dependent on the **Market Dimension** will make the **Dimension Dependency** field available to Products.

Selecting Dimension Point is available for:

- Classifications
- Products
- Images & Documents

For more information about Dimensions, see "About Dimensions" on page 238

When a Dimension Point is selected for an Object, it will be valid in, e.g. this specific Market.

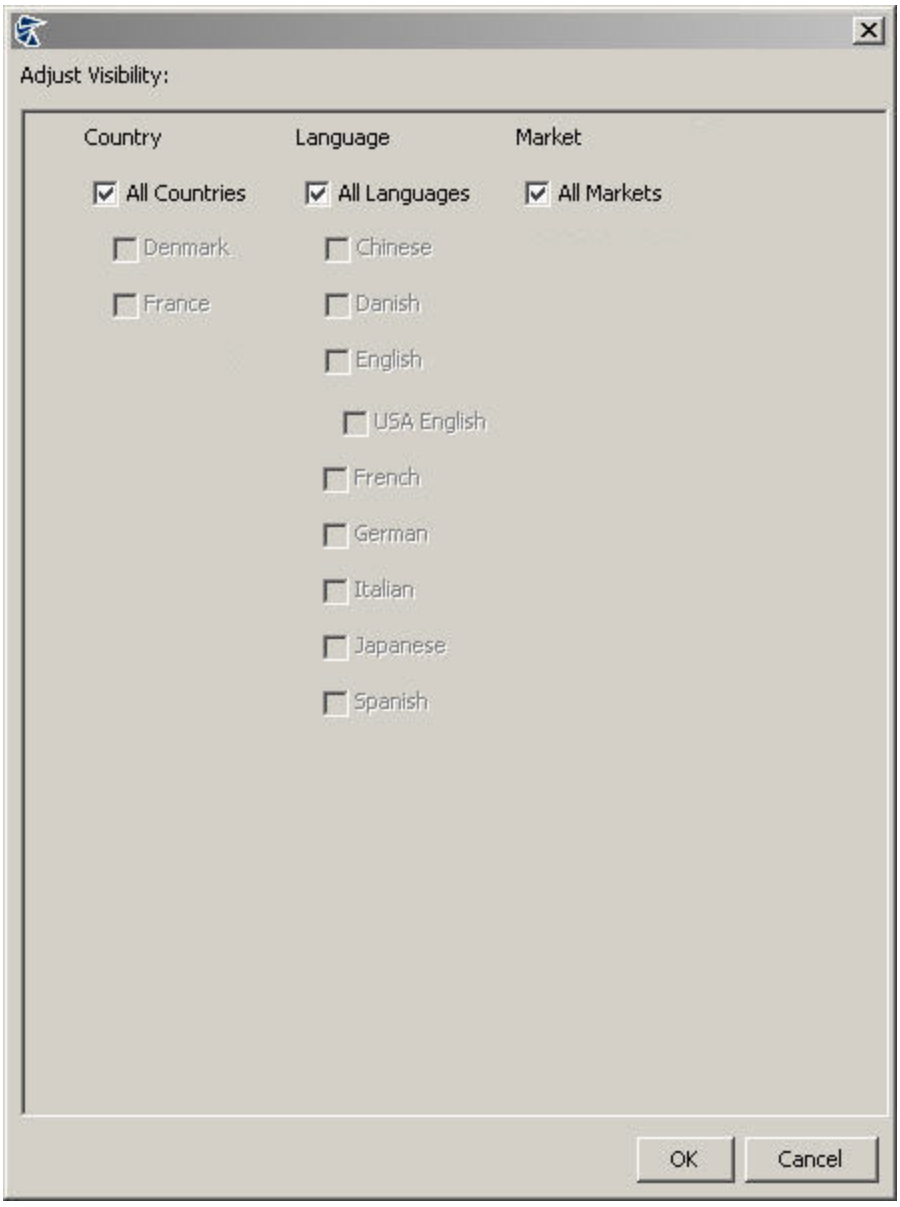
For example, a Product may only be valid for the **German Dimension Point** in the **Market Dimension**, because it is not sold in other markets.

1. In [Tree], select the relevant Object.

   The corresponding editor appears.

2. Click the **Visibility** field, and then click on the appearing **[...]** button.

   A dialog box will appear listing the valid Dimensions.



3. Check the relevant Dimension Points, and then click **OK**.

   The Object will only be visible in the selected Dimension Points.

Showing Source Context (Language)

Showing source language is done in the **View** menu -> **Show Source Context**.

Showing source language is available for:

- Products

Using the **Show Source Context** command will display the **Language Dimension** Dependency on a Product.

This means that the source Language of inherited values on a product object (attribute values and product names) is indicated. In this way you always know which Language the values originate from. This can be useful for translation purposes.

The inheritance of values is defined in the **Language** Dimension hierarchy set up in the [System Setup].

An example:

We have a Dimension hierarchy with these Languages:

Below is a description of the example:

Context	Description
French	The Product has values that are applied in the “French” language, and when the “French” language is selected, a source column will display “French” next to these values. Values applied in the “French” language will be inherited to the “Belgium French” and “Swiss French” languages.
Belgium French	The values that are applied in the “French” language are inherited to the “Belgium French” language. When the “Belgium French” language is selected, the source column will display “French” next to these values. If the values have been applied in this language, the source column will display “Belgium French” next to these values. Values applied in the “Belgium French” language will not be inherited to the “Swiss French” language.
Swiss French	The values that are applied in the “French” language are inherited to the “Swiss French” language. When the “Swiss French” language is selected, the source column will display “French” next to these values. If the values have been applied in this language, the source column will display “Swiss French” next to these values. Values applied in the “Swiss French” language will not be inherited to the “Belgium French” language.

1. From the **Context** list in the toolbar, click the relevant Context.
2. In [Tree], select the relevant Product.

   The Product Editor appears.
3. On the **View** menu, click **Show Source Context**.

   The Product Editor appears with a Source column indicating the source Dimension Point of the values.
4. You can now switch between the relevant Contexts to view and possibly edit names and values.

Specifying a Default Context

The Context determines, for example, the language or country of the data that you enter should apply to. To ensure that STEP always starts up in your preferred Context, you can specify a Default Context as follows:

1. From the **View** menu, select **Default Context**.
2. Select the preferred **Context**.

You can still work in other Contexts, however, if you select a Context that is not the default Context, the toolbar turns pale pink to make you aware that you are not working in the default Context.

Event Processors

An event processor includes a queue of events and a processor engine for processing the events. An event processor has no delivery method, but is otherwise similar to event-based outbound integration endpoints.

To create an event processor, complete the following steps:

1. Perform one-time setup tasks to allow creation of event processors. See **Event Processor Setup**.
2. Create event processors. See **Creating Event Processors**.
3. Enable the event processors. See **Enabling Event Processors**.

If an event process is being used to move data outside of STEP, an OIEP will be required as a delivery method.

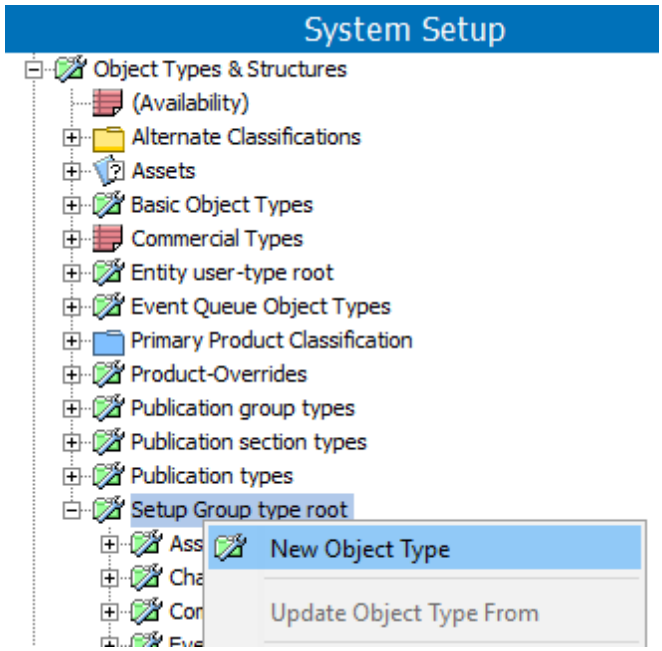
Setting Up for Event Processors

Before you can create an event processor, the following one-time setup steps must be completed:

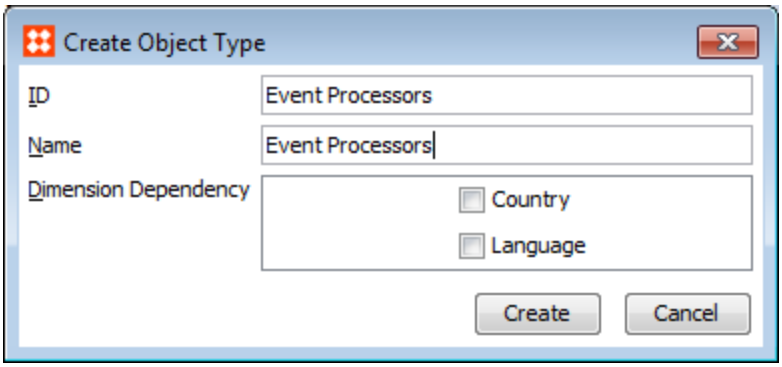
- Create an event processor setup group type.
- Link event processor object types to event processor setup group types.
- Create an event processor setup group.
- Set up action sets and privileges.

Create Setup Group Types for Event Processors

1. In System Setup > Object Types & Structures > right-click Setup Group type root and then click **New Object Type**.



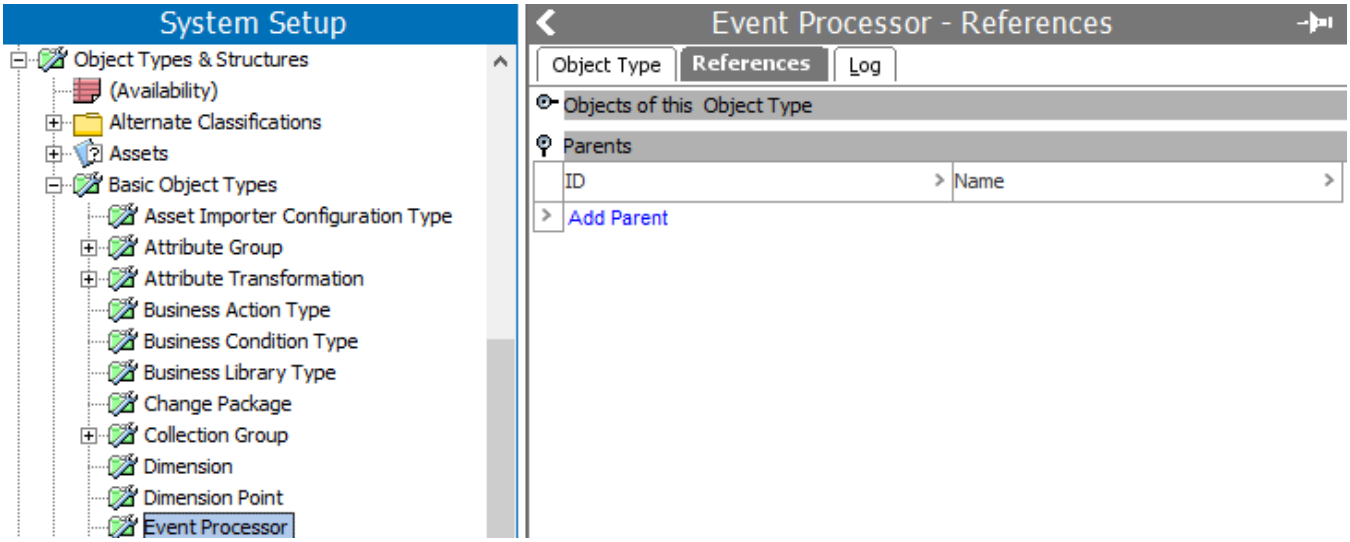
2. Enter an ID and a Name, and then click the **Create** button.



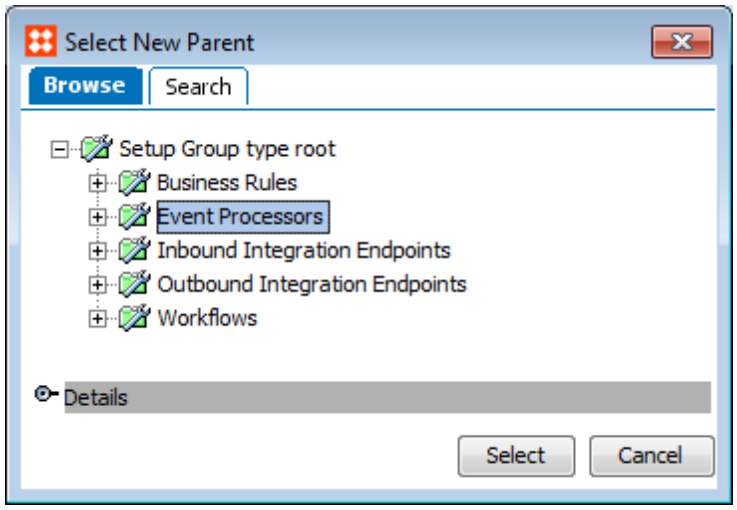
Link Event Processor Object Types to Setup Group Types

Linking determines the object types that can be displayed at each level of a hierarchy.

1. In System Setup> Object Types & Structures > **Basic Object Types**, select your event processor object type to display the editor.

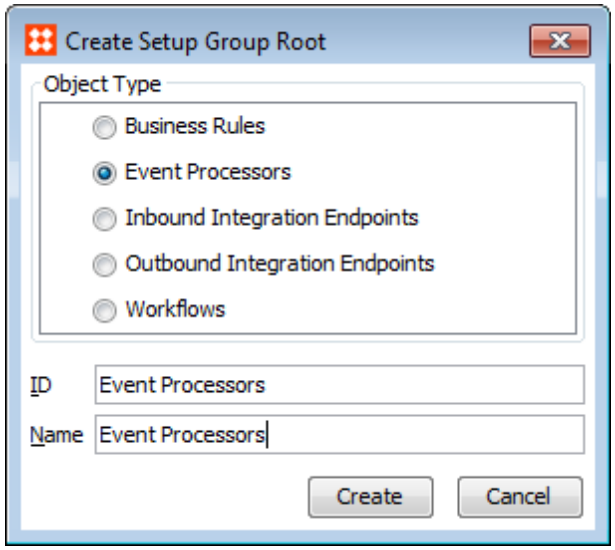


2. On References tab, open the Parents flipper, and click the **Add Parent** link.
3. In the Select New Parent dialog, browse or search for the relevant setup group type, and then click **Select**.

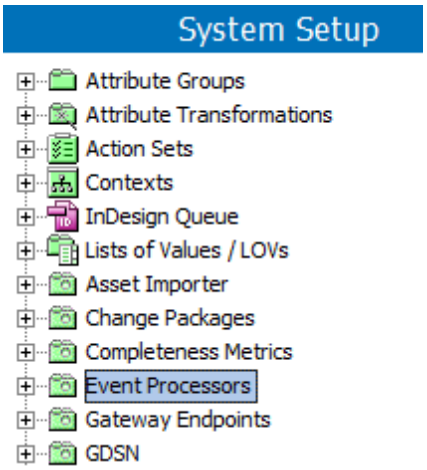


Create Event Processor Setup Groups

1. In System Setup, select any object in the hierarchy.
2. On the Maintain menu, point to Insert, and then click **Setup Group Root** to display the Create Setup Group Root dialog



3. Select your event processors object type.
4. Enter an ID and a Name, and then click **Create**. The setup group is created and appears as a node in the System Setup hierarchy.



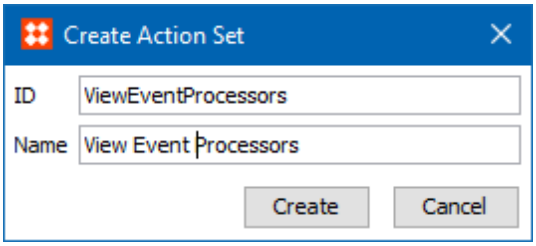
Set Up Event Processor Action Sets and Privileges

When you set up privileges, you first specify one or more action sets that apply to event processors, and then you associate the action sets with one or more users or user groups. The action set defines the actions the members of the user group are allowed to perform on the event processors.

In some situations, it can be useful to create more event processor specific action set. For example, if you have users that are allowed to maintain setup groups and users who are only allowed to view event processors.

Create Event Processor Action Sets

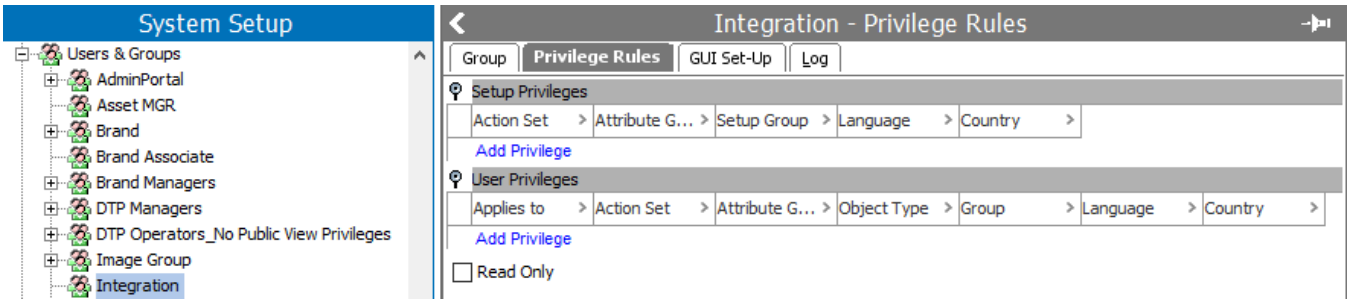
1. In System Setup, expand Action Sets, and then select **Setup Actions**.
2. Right-click Setup Actions, and then select **New Action Set** to display the Create Action Set dialog.
3. Enter an ID and a Name, and then click **Create**.



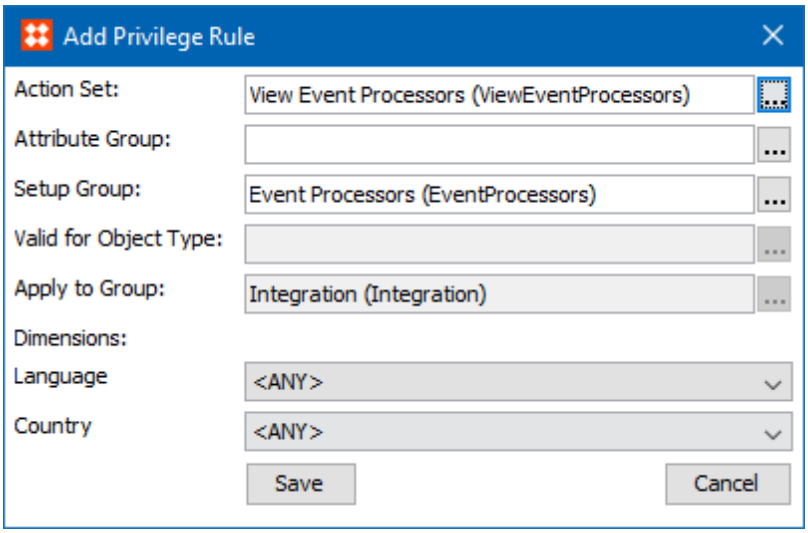
4. Select the new Action Set, and then on the Action Set tab, click **Add Action**.
5. In Select Action dialog, select one or more of the following actions that are relevant for event processors that you want to be part of this action set, and then click **Select**.
   - Maintain Event Processor - Users can create, delete, and configure event processors. Users can also use the cut, copy and paste actions.
   - Maintain Setup Group - Users can create and delete setup groups. User can also use the cut, copy and paste actions.
   - View Event Processor - Users view event processors defined in STEP.
   - View Setup Group - Users can view setup groups.

Set Up Event Processor Privileges

1. In System Setup, expand **Users & Groups**, and then select the relevant user group that will have the new event processor action(s).
2. Click the **Privilege Rules** tab.



3. Under the Setup Privileges flipper, click the **Add Privilege** link to display the Add Privilege Rule dialog.



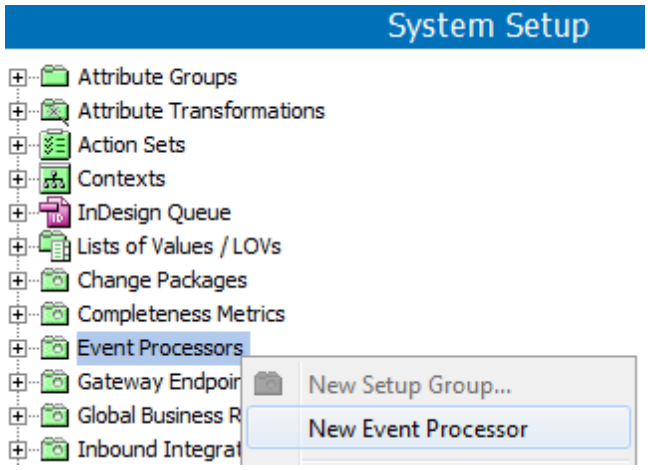
4. In Action Set, click the ellipsis button **(…)**, and then select the relevant event processor action set.
5. In Setup Group, select the relevant event processor setup group, and then click **Save**.

   For more information on privileges, see Adding User Privileges for a Group and for action sets see About Action Sets.

Creating Event Processors

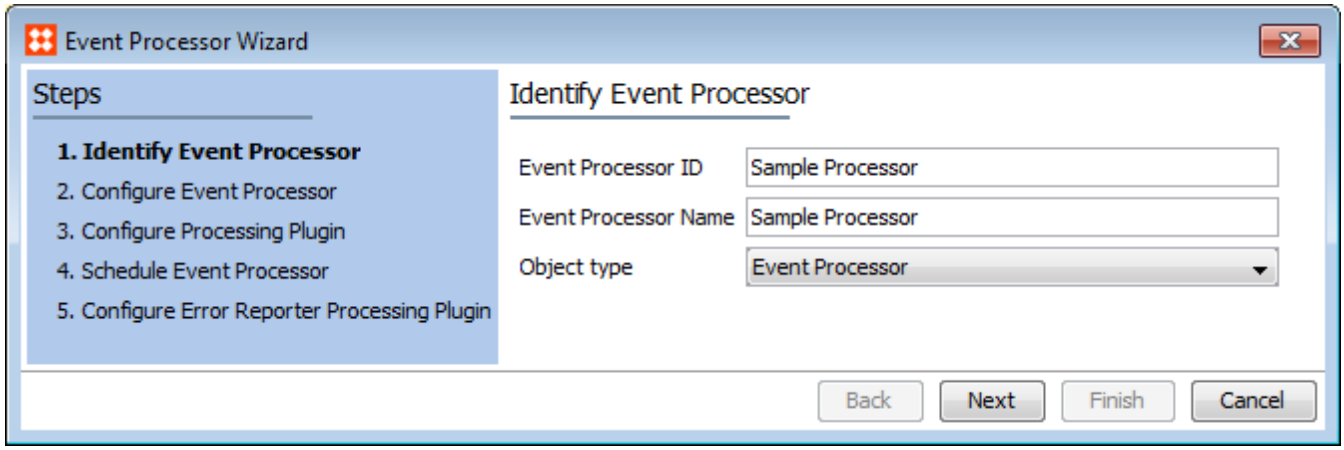
When creating an event processor, consider the following common setup:

1. In **System Setup**, right-click the **Event Processors** setup group, and then select **New Event Processor**.



2. On the **Event Processor Wizard** Step 1, identify the new event processor by adding:
   - an **ID**
   - a **Name**
   - **Event Processor** as the **Object type**

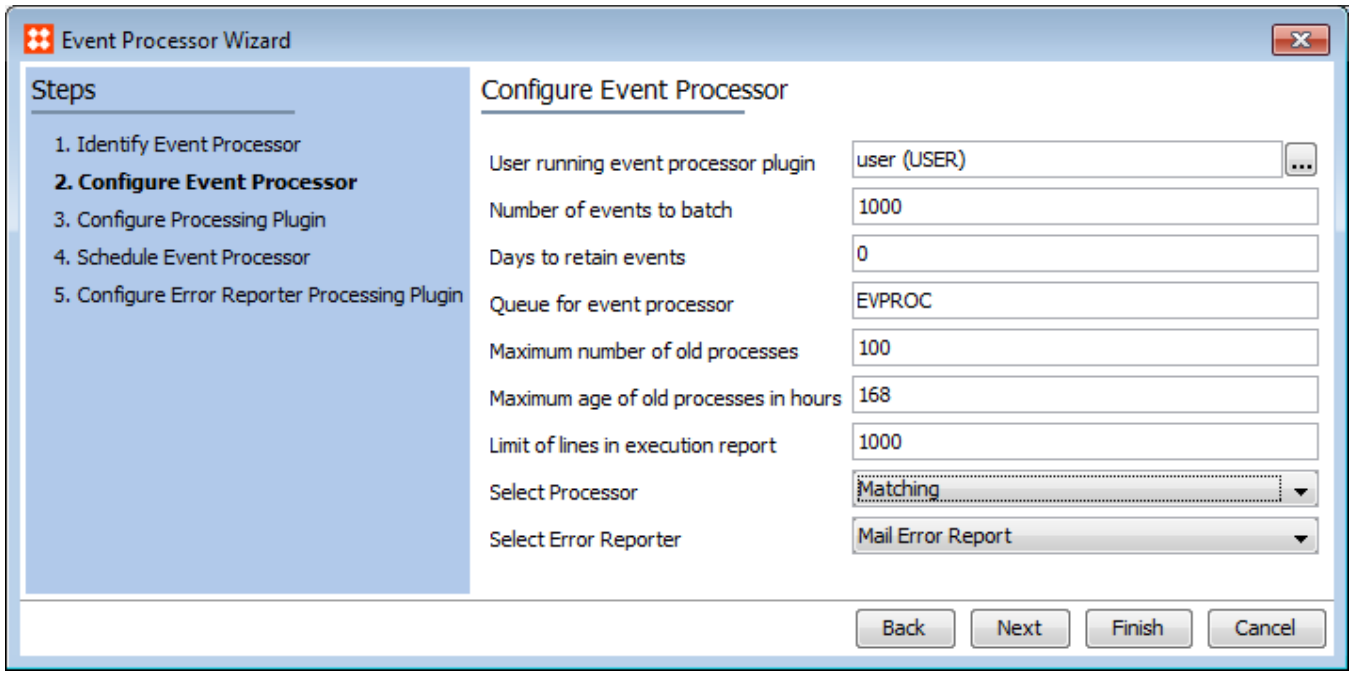
   and click **Next**.



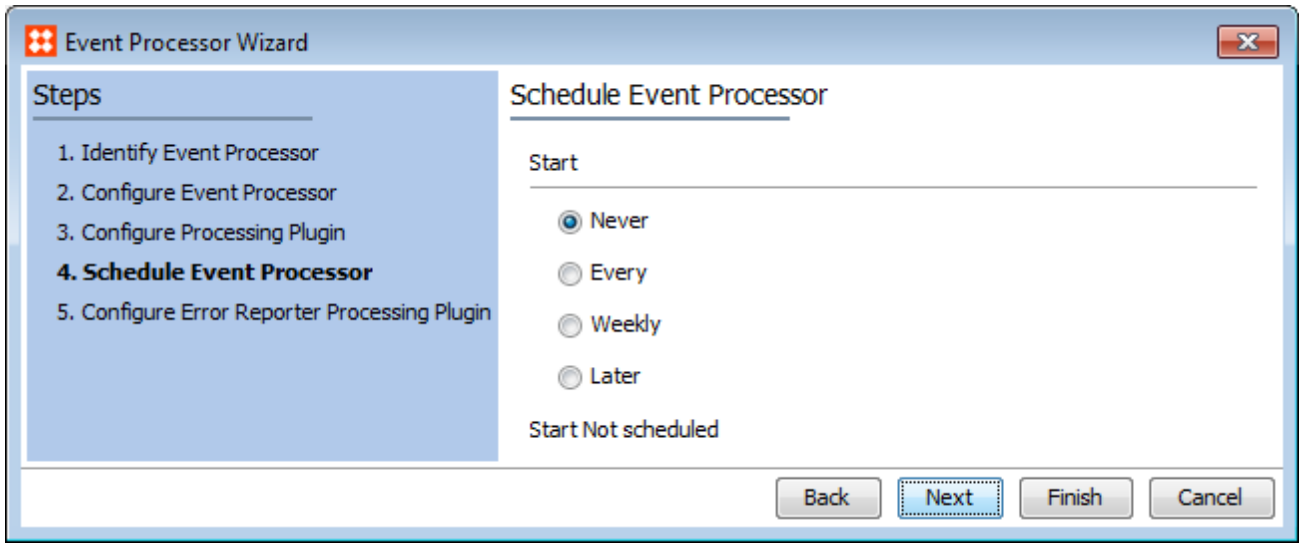
3. On Step 2, configure the new event processor by entering the information detailed below . Common setup is to use the default settings unless there is a valid reason to change them. Once any required changes are made,

click **Next**.

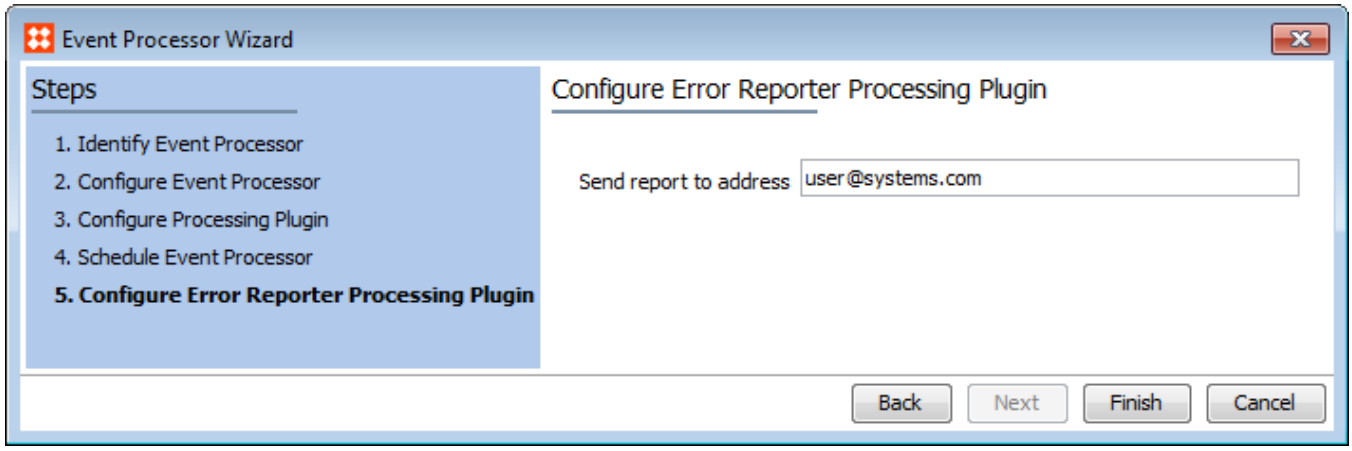
- **User running event processor plugin**: To search or browse for a user, click the ellipsis button **(…)**. The user selected determines the privileges the event processor applies when it processes data, so access must be granted to the objects being processed. Creating a user specifically for an event processor allows you to identify actions taken by the process.
- **Number of events to batch**: Specify the batch size. This allows multiple events to be transmitted in a single batch. Setting this number too low can result in slower processing as more background processes will be required.
- **Days to retain events**: Specify the number of days to keep events once processed. Common setup is to leave this at 0 since this option is not valid for an event processor; it is available for reprocessing events via an OIEP.
- **Queue for event processor**: Enter the name of a queue to process data from the event processor. The queue is automatically created if it does not already exist.
- **Maximum number of old processes**: Specify the number of ended processes the system will keep. Succeeded and ended processes are deleted when the number exceeds the specified limit. The oldest processes are deleted first. Setting this number too high may eventually degrade performance.
- **Maximum age of old processes in hours**: Specify the maximum age of ended processes that the system will keep. Ended processes are is deleted when the maximum age is exceeded. Setting this number too high may eventually degrade performance.
- **Limit of lines in execution report**: Specify the maximum number of lines to store from the execution report in the log. This setting impacts storage used.
- **Select Processor**: Choose from the available options, including the following standard core processors:
  - **Expand LOV Value Changes** allows the event processor to react on all LOV value update events (only for LOVs that use Value IDs) including those made manually from workbench or Web UI, via LOV value merge, via import, etc. Note that the processor handles LOV value updates only, not additions or deletions. An OIEP can be configured to listen for changes to LOV values and generate events for objects using the modified values.
  - **Matching** allows for the generation / updating of match code values, as well as the option to run matching algorithms, whenever an event is processed on the objects acted on by the algorithm. Note that the configuration requires selection of a matching algorithm, and that match codes are only updated for objects acted on by the algorithm and only when events are passed for those objects.
  - **Revision Management** allows automatic purging of object revisions in order to limit the total number of revisions retained. Multiple revision management event processors can be configured, which enables the number of revisions to differ based on the object type.
- **Select Error Reporter**: Choose either Do nothing or Mail Error Report.



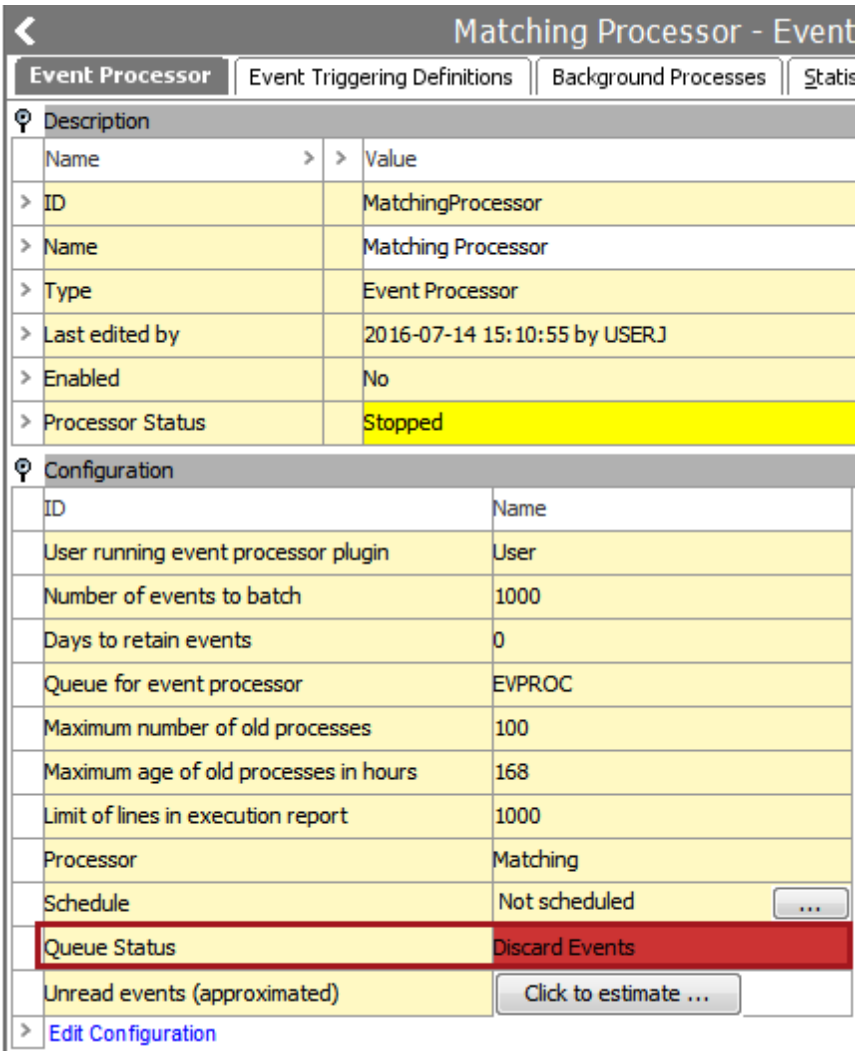
4. On Step 3, depending on the processor selected in Step 2, configure the processing plugin if required, and click **Next**. The following core processors are standard:
   - For the **Expand LOV Value Changes** processor, no further configuration is required. For additional information, see the **Expand LOV Value Changes Processor Parameters and Triggers** section.
   - For the **Matching** processor, see information in the **Matching Processor Parameters and Triggers** section.
   - For the **Revision Management** processor, see information in the **Revision Management Processor Parameters and Triggers** section.
5. On Step 4, schedule the event processor by selecting a start option, and then click **Next**. Options include:
   - **Never:** only start the event processor manually, no schedule is applied.
   - **Every:** interval in number of minutes that the event processor will run. One (1) minute is the shortest interval allowed, and is closest to real time.
   - **Weekly:** specific weekday and time that the event processor will run, also the start and end dates for the schedule.
   - **Later:** a specific time and date that the process will run only once.



6. If the Select Error Reporter option was set to Mail Error Report, Step 5 allows you to add the email address.



7. Click **Finish** to complete the wizard.
8. Next, use the event processor editor to update the **Queue Status**, if necessary. By default, events are discarded. This is useful when testing a processor without modifying data. Click the Queue Status field and select **Read Events** if you are ready for the processor to apply the actions based on the events processed.

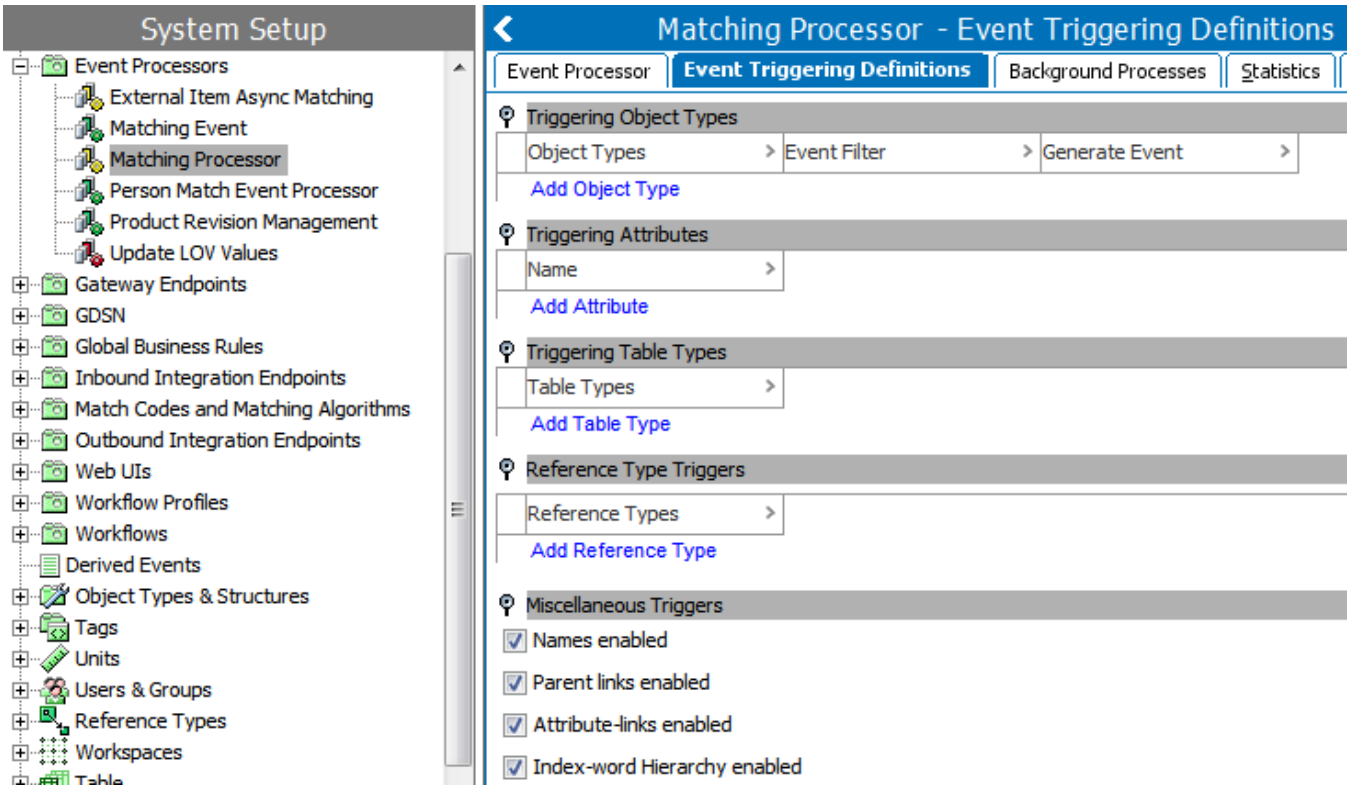


9. Finally, use the event processor editor to configure **Event Triggering Definitions**. All event processors require trigger definitions in order to identify objects, attributes, table types, reference types, and/or miscellaneous elements that will generate an event.

   The options available on the Event Triggering Definitions tab of the Event Processor are the same as those in an OIEP. For details about the options, see the **Outbound Integration Endpoint Event Triggering Definitions** section of the **Integration Endpoints** documentation.

   For information on common trigger setup for the core event processors, see the following sections:

   - **Expand LOV Value Changes Processor Parameters and Triggers**
   - **Matching Processor Parameters and Triggers**
   - **Revision Management Processor Parameters and Triggers**


System Setup
Event Processors
External Item Async Matching
Matching Event
Matching Processor
Person Match Event Processor
Product Revision Management
Update LOV Values
Gateway Endpoints
GDSN
Global Business Rules
Inbound Integration Endpoints
Match Codes and Matching Algorithms
Outbound Integration Endpoints
Web UIs
Workflow Profiles
Workflows
Derived Events
Object Types & Structures
Tags
Units
Users & Groups
Reference Types
Workspaces
Matching Processor - Event Triggering Definitions
Event Processor
Event Triggering Definitions
Background Processes
Statistics
Triggering Object Types
Object Types
Event Filter
Generate Event
Add Object Type
Triggering Attributes
Name
Add Attribute
Triggering Table Types
Table Types
Add Table Type
Reference Type Triggers
Reference Types
Add Reference Type
Miscellaneous Triggers
Names enabled
Parent links enabled
Attribute-links enabled
Index-word Hierarchy enabled

Expand LOV Value Changes Processor Parameters and Triggers

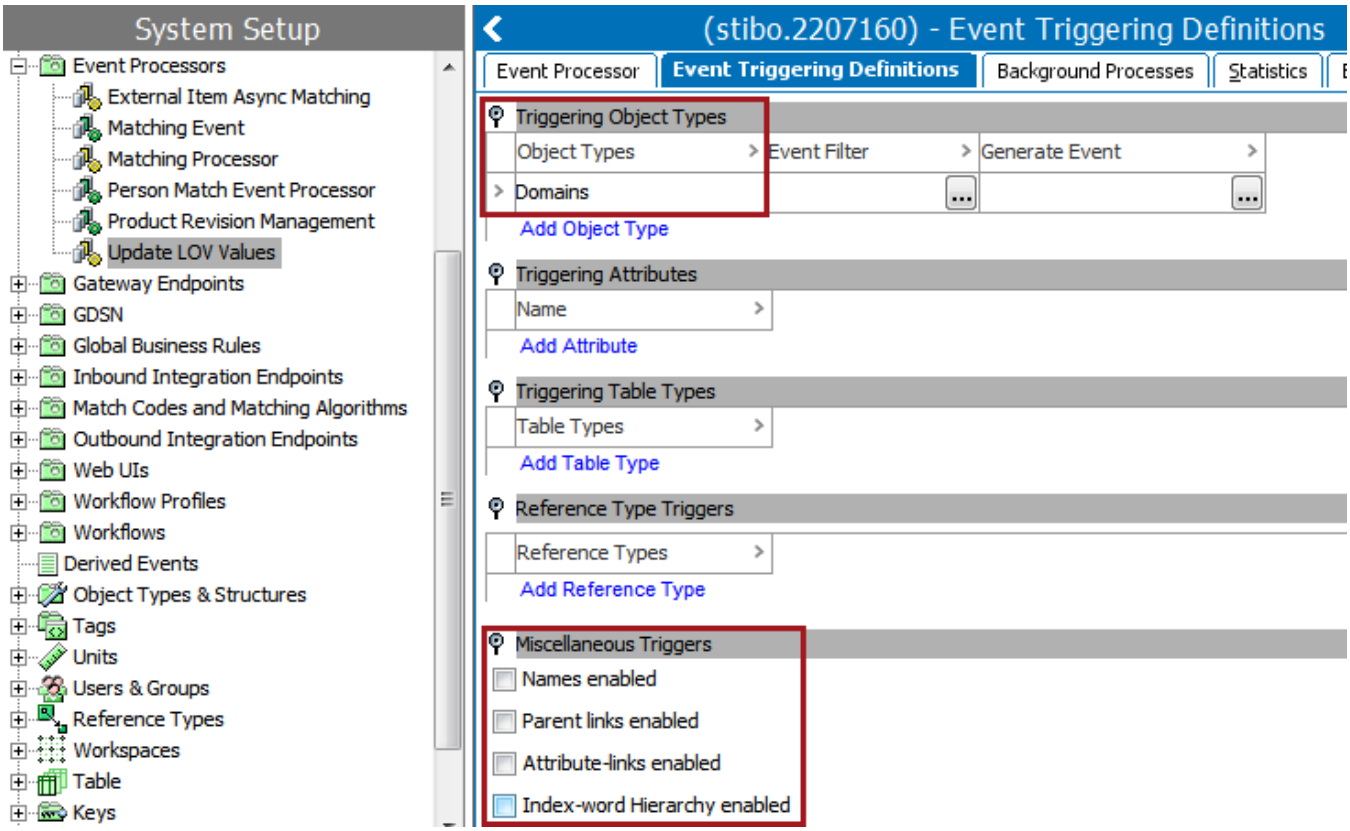
All of the Event Processor Wizard steps are detailed in the **Creating Event Processors** section. The following settings should be considered when creating an event processor for expanding LOV value changes.

Parameters

On the Event Processor Wizard Step 3, no further configuration is required for the Expand LOV Value Changes processor.

Event Triggers

For the event processor to take effect, it must be configured to only listen for changes on LOVs (Domains), and all other default triggering definitions should be unselected.



In order to pass events on objects that have a changed LOV value, set up an outbound integration endpoint to listen for modify events on the relevant objects. Note that if change flags are used in the format for the endpoint receiving the event, the relevant objects will have change flags on all attributes set to true (instead of only the

changed attribute), because the endpoint interprets the event generated by the processor as a change to the object rather than to the individual attribute value.

Note: When an integration endpoint listens for changes on LOV values, it is possible that information about the modified LOV will be published before objects using affected LOV values are published. This means that until both the LOV and the objects are published, an object's value (for example, a product's attribute value) may not correspond with values in the LOV.

For more information, see the **Event-Based Outbound Integration Endpoint Wizard** section of the **System Setup / STEP Super User Guide**.

Matching Processor Parameters and Triggers

All of the Event Processor Wizard steps are detailed in the **Creating Event Processors** section. The following settings should be considered when creating an event processor for matching.

Note: The 'Run Matching Algorithm' functionality is controlled by System Setup > Action Sets > Setup Actions > Maintain Deduplication Configurations.

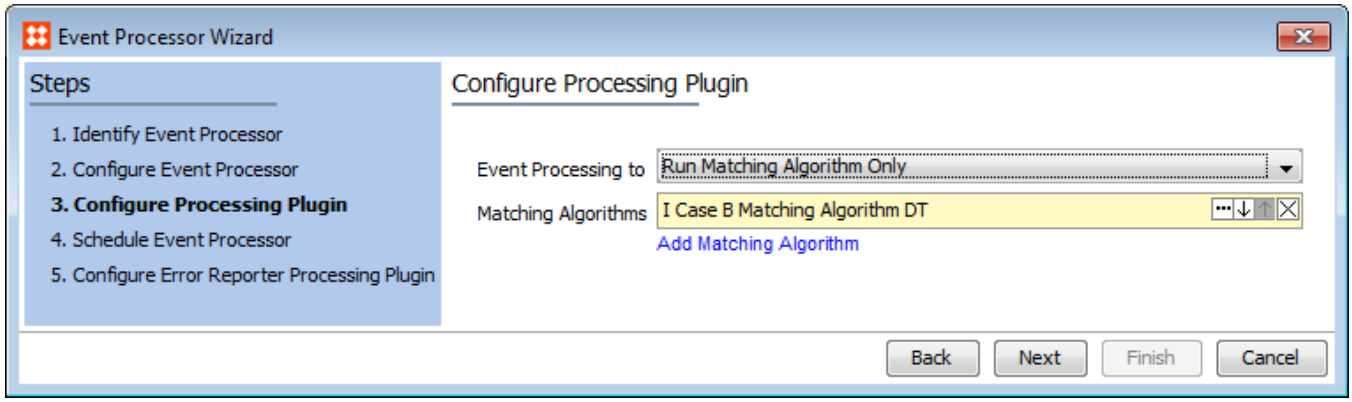
For details about match codes and matching algorithms, see **Matching and Linking Components** in the **Matching and Linking** documentation.

Parameters

On the Event Processor Wizard Step 3, configure the processing plugin.

1. For **Event Processing to**, select an option:
   - **Run Matching Algorithm Only** runs the selected algorithm(s) using the existing match codes.
   - **Generate / Update Match Code Values Only** creates missing match codes, deletes invalid match codes, but does not run the algorithm(s).
   - **Generate / Update Match Code Values and Run Matching Algorithm** does both of the first two options. If configured to generate / update match code values, only objects included in the matching algorithms specified on this step are affected.
2. For **Matching Algorithms**, to specify which matching algorithm(s) to run, click the **Add Matching Algorithm** link and use search or browse. When multiple algorithms are selected, they are processed in order, from top to bottom.

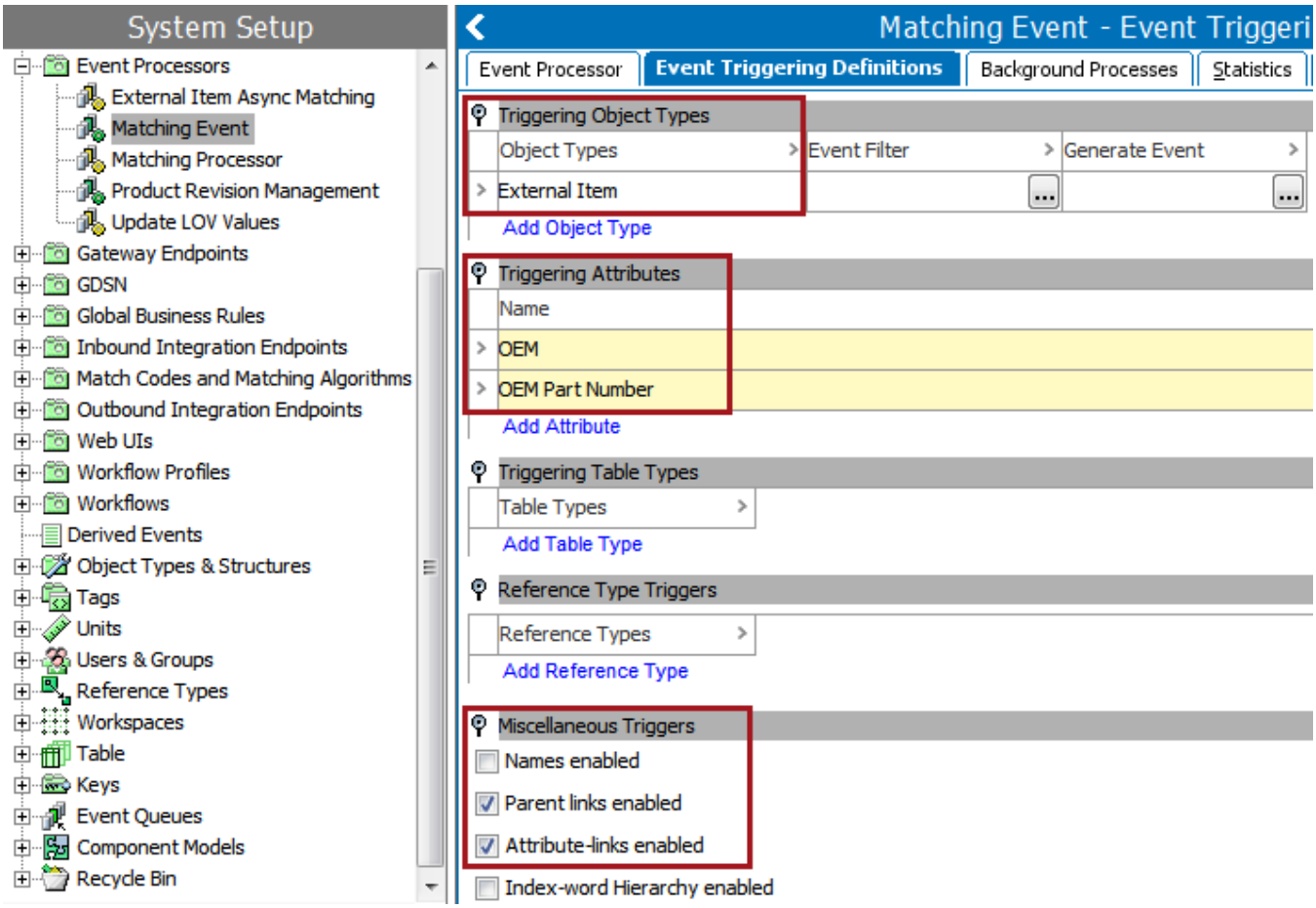
   When working in a golden record scenario, consider the order of algorithms when there are references between multiple golden records within multiple matching algorithms. Common setup is to match 'reference targets' before 'reference sources' to ensure optimal matching performance, and as complete an initial construction of golden records as possible.



Event Triggers

On the Event Processor's Event Triggering Definitions tab, use the Triggering Object Types, Triggering Attributes, and Miscellaneous Triggers flippers to configure the event processor to listen for changes on anything that the match code or matching algorithm needs to perform.

- If the event processor will update match code values, select the attributes used by the Match Code definition.
- If the event processor will run a matching algorithm, select all object types, attributes, and links that are specified by the Matching Algorithm within the Global Binds or the Match Criteria flippers. These can include attribute values, IDs, names, attribute references, and parent links.



Revision Management Processor Parameters and Triggers

All of the Event Processor Wizard steps are detailed in the **Creating Event Processors** section. The following settings should be considered when creating an event processor for managing revisions.

Parameters

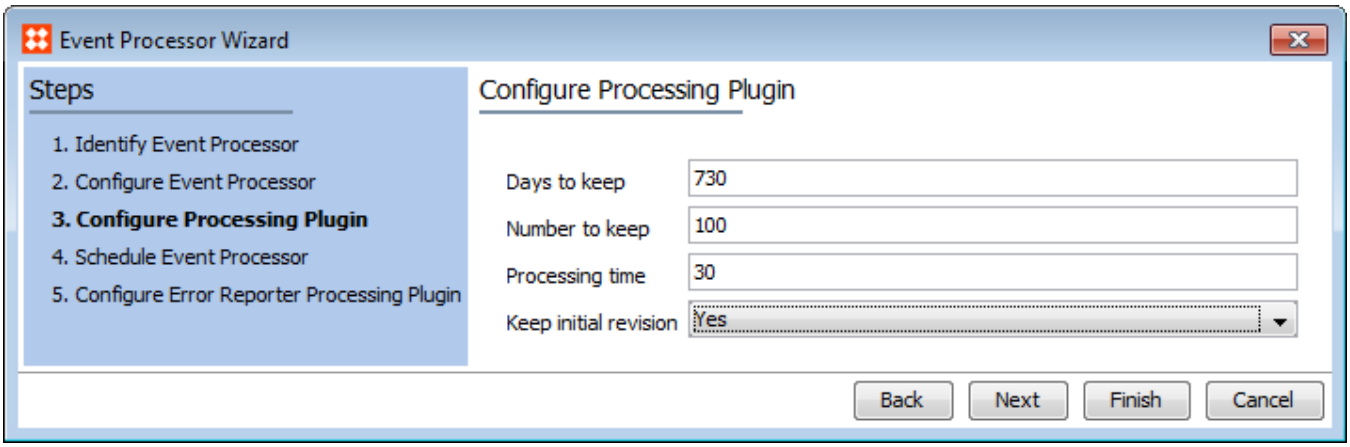
On the Event Processor Wizard Step 3, configure the Revision Management processor.

Note: A revision must fall outside *all* conditions set in the Days to keep, Number to keep, and Processing time parameters in order to be deleted.

- **Days to keep:** uses calendar days to determine the age of a revision, and keeps only those that are less than the number of days specified.
- **Number to keep:** maximum number of revisions remaining after processing.
- **Processing time:** determines the maximum number of seconds that the processor will run on each object.
- **Keep initial revision:** when set to Yes, the first revision is kept, even if it falls outside all other conditions. Common setup is to keep the initial revision if you want to know when an object was initially created.

For example, considering the values in the image below, a revision is deleted if :

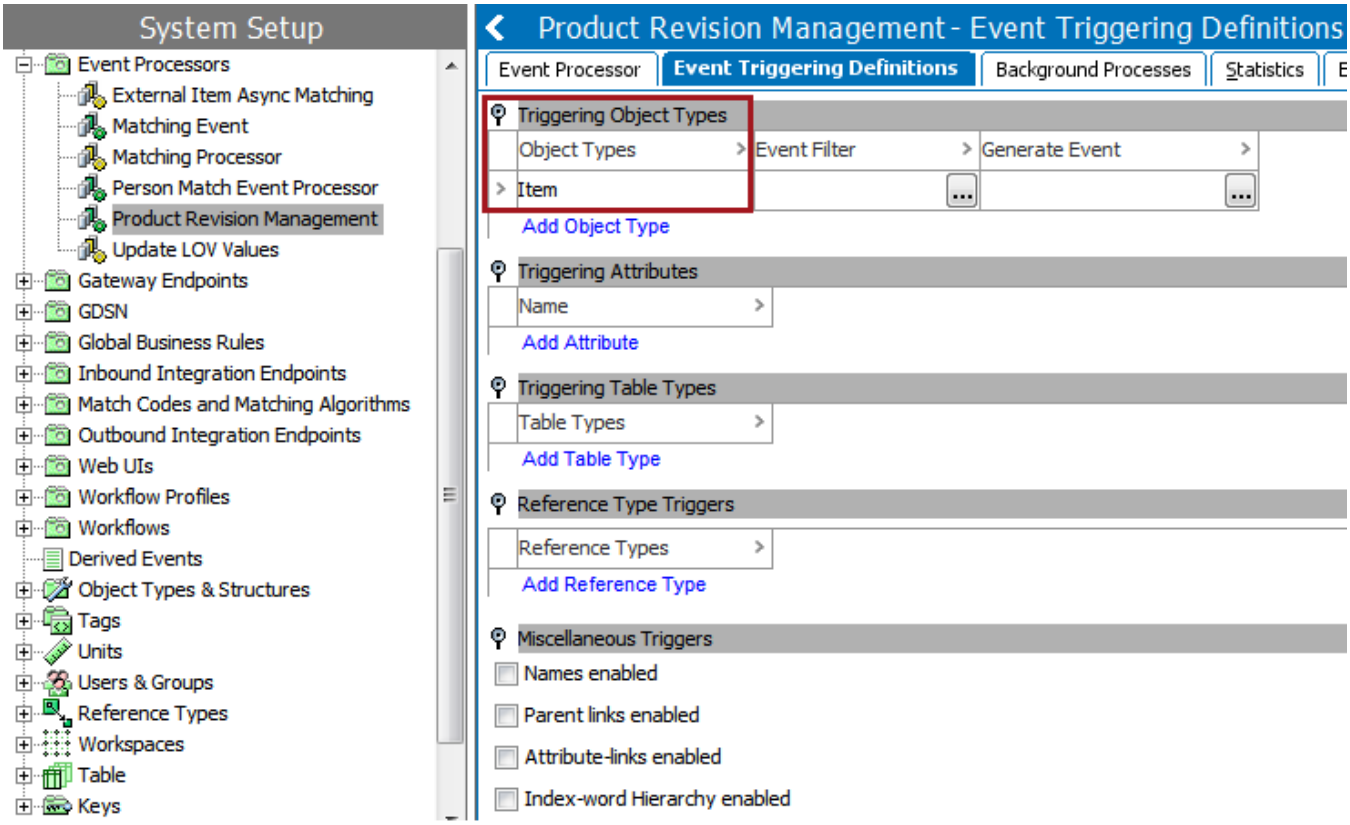
1. The revision being processed is at least 731 days old, **and**
2. More than 100 revisions will remain after deletion of the revision being processed, **and**
3. The processor has not run for 30 seconds on the revision being processed, **and**
4. The revision being processed is NOT the initial revision.



Event Triggers

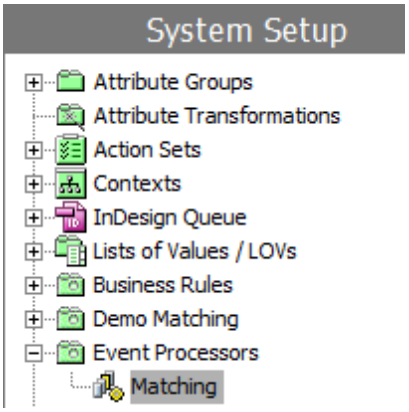
Select the object type(s) that should have revisions deleted according to the configured limits.

If a different number of revisions is needed based on the object type, create an additional event processor with the necessary configuration limits.

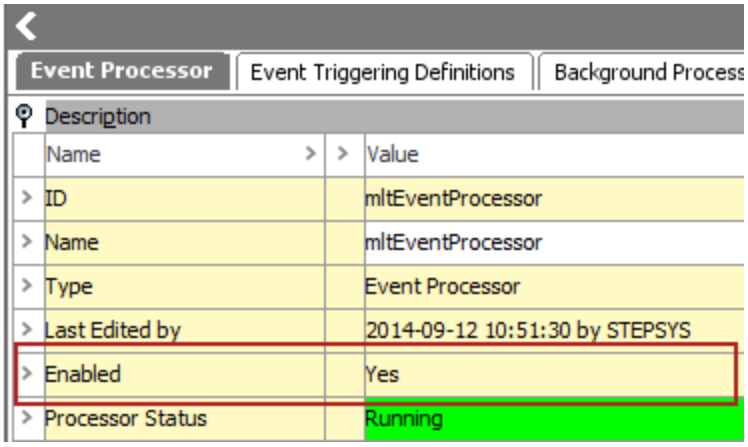


Enabling Event Processors

1. In System Setup, expand the Setup Group, and select the **Event Processor** you want to enable.



2. Right-click the event processor, and click **Enable Event Processor**. On the **Event Processor** tab, the value of the **Enabled** field changes to Yes.



Event Processor States

The state of an event processor is identified by the color of the icon. State information is also displayed in text on the Event Processor tab of the editor, as shown above.

Icons	Description
	Active: the event processor is connected and running.

Icons	Description
	Disabled: the event processor has been disabled by user request.
	Stopped: the event processor has stopped because of a failure.

Editing an Event Processor

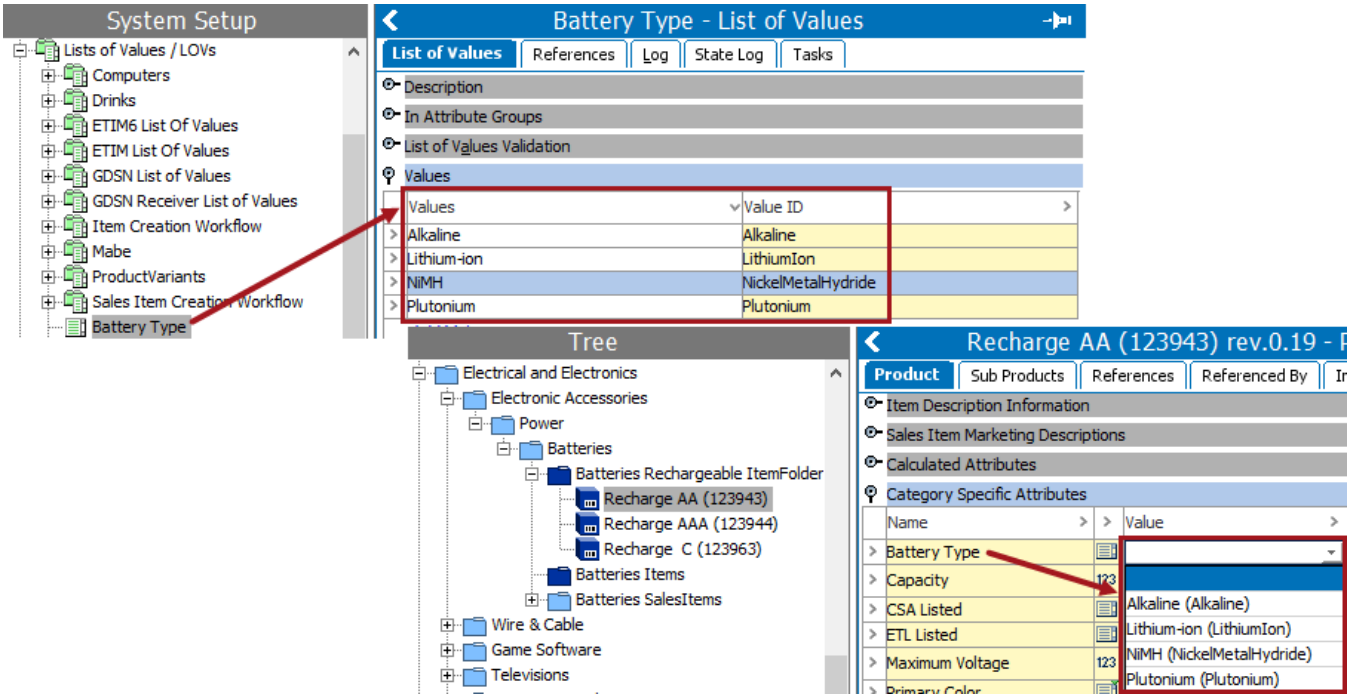
1. On the System Setup tab, select the event processor to be edited.
2. Choose a method to edit the wizard settings displayed in the Event Processor editor > Event Processor tab > Configuration flipper:
   - Right-click the event processor and select 'Edit Event Processor'.
   - Expand the Configuration flipper and click the 'Edit Configuration' link.
3. Change Event Triggering Definitions on the editor's Event Triggering Definitions tab.

Deleting an Event Processor

1. On the System Setup tab, select the event processor, right-click it, and select 'Disable Event Processor'.
2. Select the event processor again, right-click it, and select 'Delete'.
3. Verify that you are deleting the correct event processor by clicking the Delete button.

List of Values (LOVs)

A List of Values (LOV) is a set of values that can be selected for an attribute with the validation base type of LOV. LOVs are maintained on the System Setup tab under List of Values/ LOVs. The values that exist on an LOV are available for selection on an attribute via a dropdown.



Ideally, each LOV value is unique. For example:

- Battery Size—AA, AAA, D, C, 9V
- Material—Brick, Steal, Wood, Plastic, Granite
- Country Code—001, 007, 045, 086, 506, 044

Values by Reference

Values in the LOV are assigned to an attribute 'by reference'. This means that if the values in the LOV are edited, then the changes are immediately reflected in the attributes associated with the LOV.

Revision Control

Revision control of the LOV is determined by the Externally Maintained status of the attribute using the LOV.

- **Externally Maintained = No** on an attribute using the LOV means that the attribute and the LOV are under revision control.

- **Externally Maintained = Yes** on an attribute using the LOV means that the attribute and the LOV are not under revision control.

Validation Base Type

The validation base type for an existing LOV can be changed. When merging multiple LOVs into one LOV, all LOVs being merged must first share the same validation base type.

Working with Large LOVs

When an LOV has over 5000 values, the full list of values cannot be instantly viewed as it would be for a smaller list of values. This is true in both the List of Values editor tab and when selecting from a large list of values on an attribute.

To find a specific LOV value, type a few characters to search for a partial or complete match.

To see all LOV values, type an asterisk (*) to initially display most of the values, and after fully loading, display the complete, scrollable list of values.

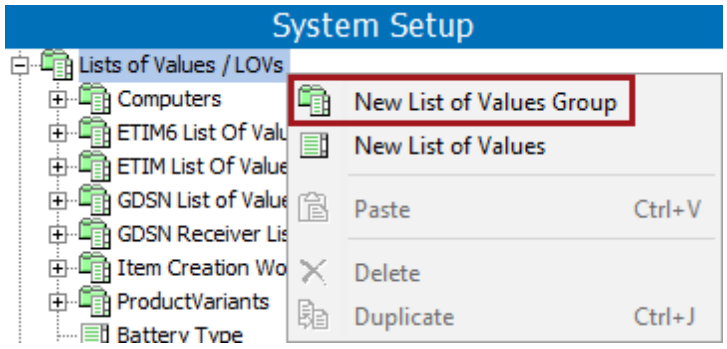
Creating an LOV Group

Individual, but related, Lists of Values can be kept together using a List of Values Group. The group can be linked into an attribute and used in combination with privileges to restrict users in maintaining certain List of Values.

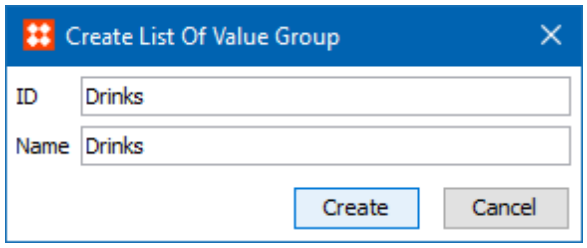
Note: If the LOV Group will be used in combination with privileges to restrict users in maintaining certain LOVs, the group must be linked into an Attribute Group. The Attribute Group can be used in combination with setup privileges to restrict maintenance on certain LOVs.

---

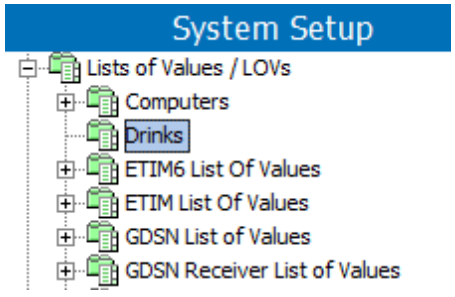
1. In **System Setup**, right-click List of Values (LOVs) and select the **New List of Value Group** option.



2. Key in **ID** and **Name**.



3. Click **Create**.



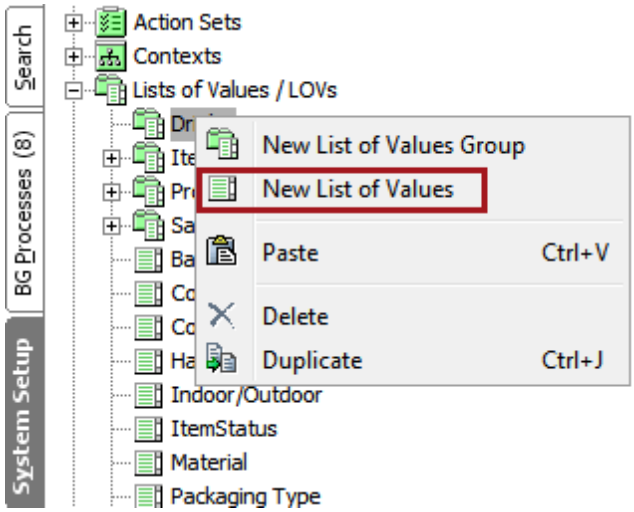
4. Add an LOV to the LOV Group by following the steps in **Creating a List of Values**.

Creating an LOV

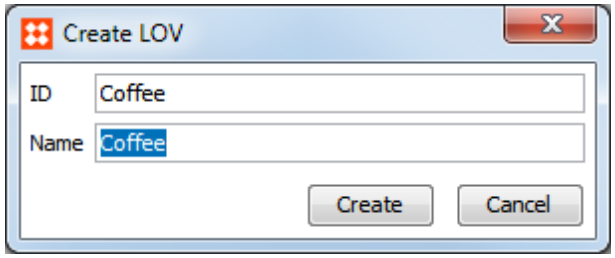
Creating an individual list of values (LOV) involves:

- Setting validation rules
- Selecting valid units
- Selecting dimension dependencies
- Adding values

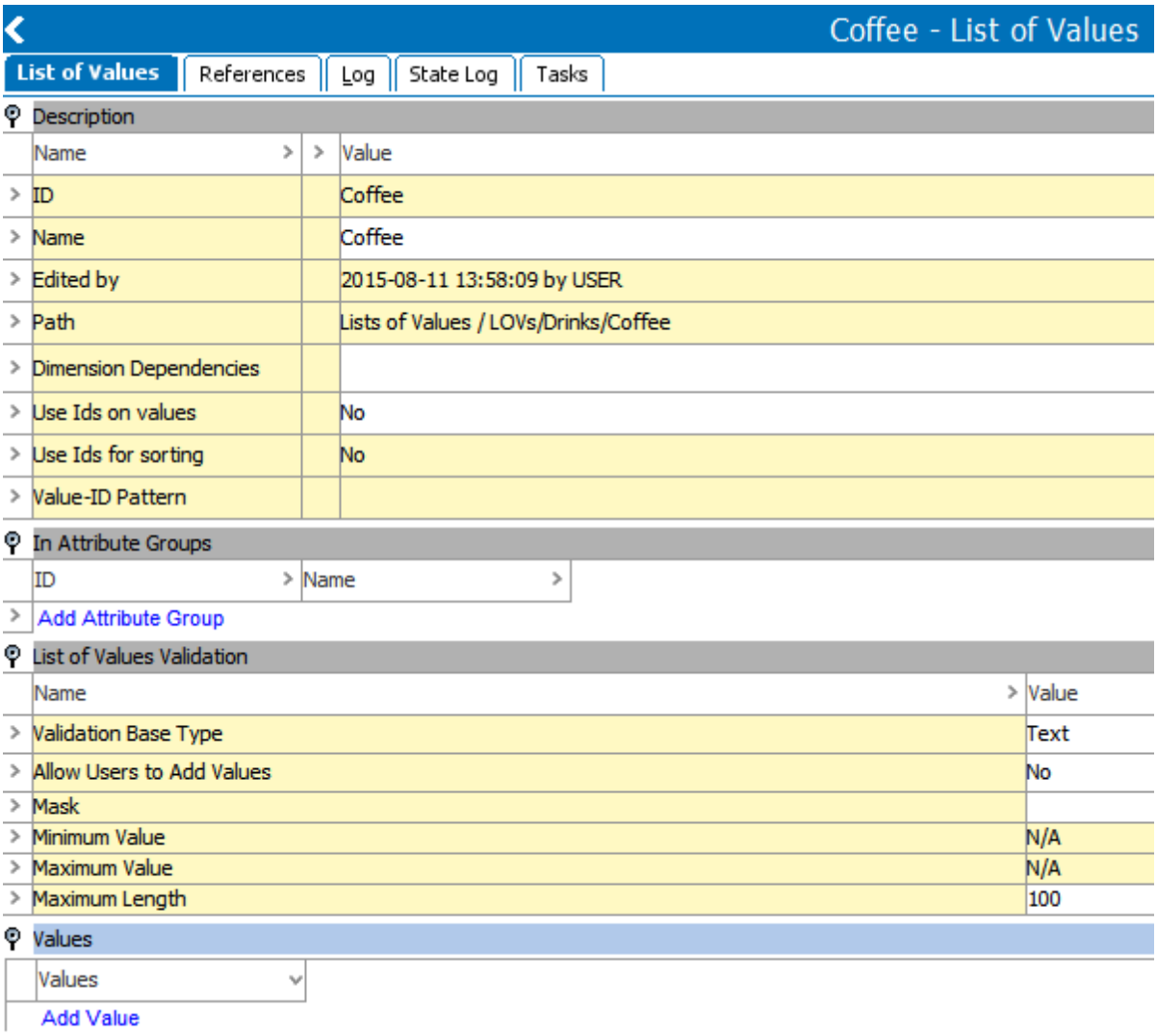
1. In **System Setup**, right-click the main List of Values folder, or the new LOVs folder you just created, and select **New List of Value.**



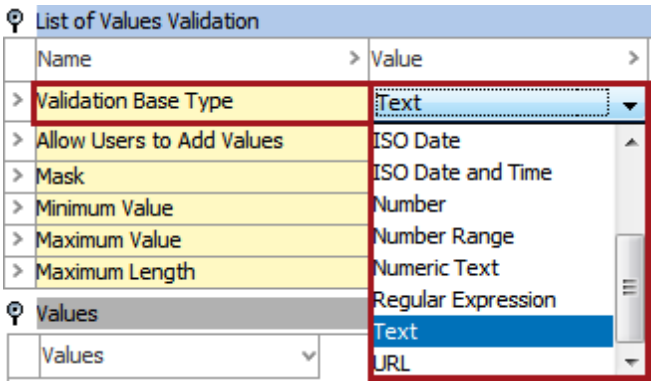
2. Add an **ID** and a **Name.**



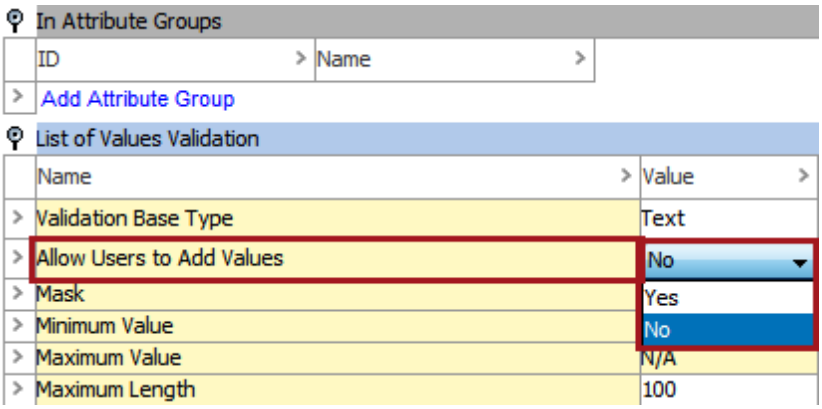
3. Click **Create** and a List of Values editor displays.



4. Under **List of Values Validation**, select validation base type.



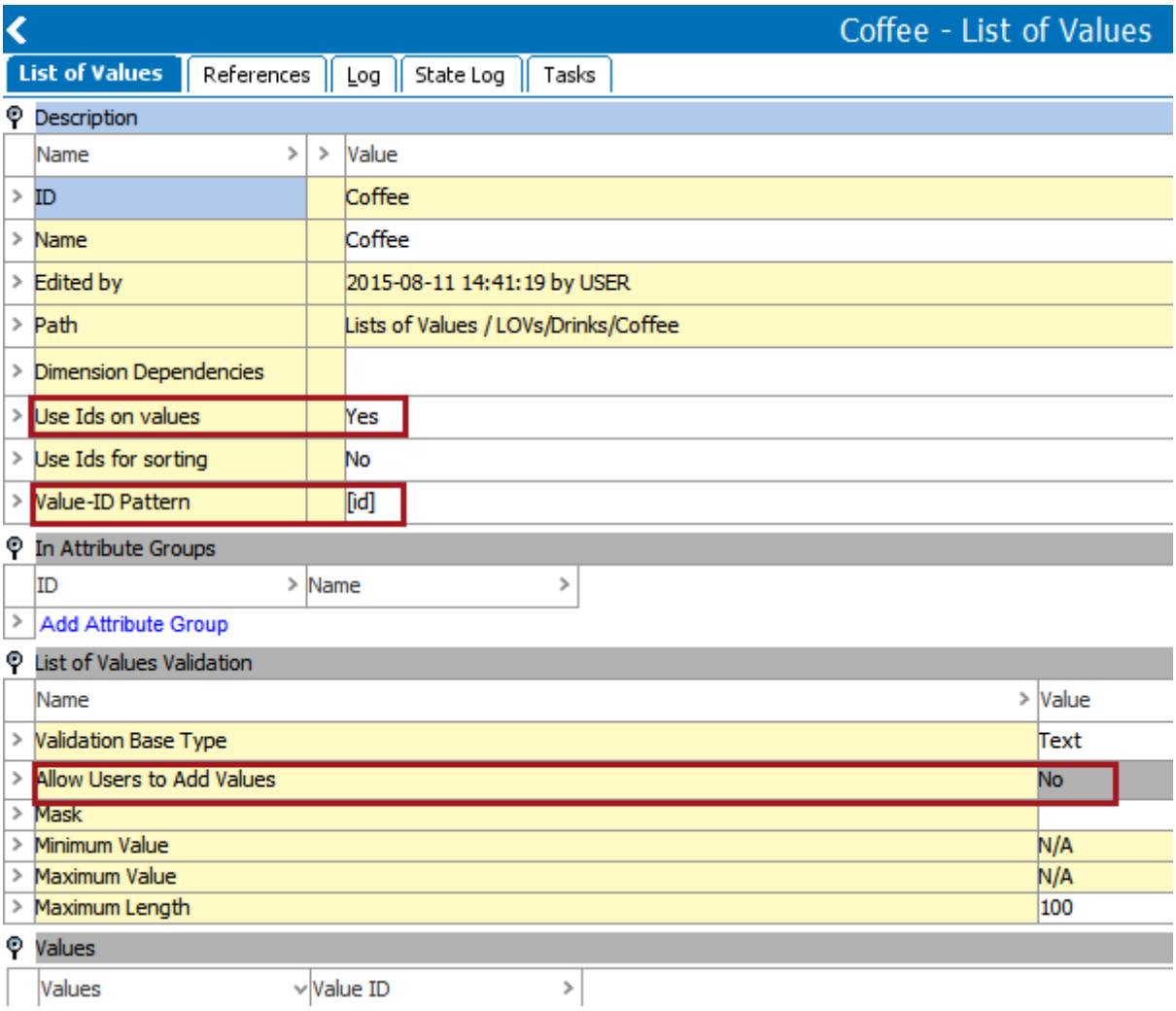
5. Under **List of Values Validation**, determine when users can add values.



- **Yes** means that the user can add values to the existing list from the object editor.
- **No** requires that the user select a value that already exists in the list. New values can only be added from System Setup.

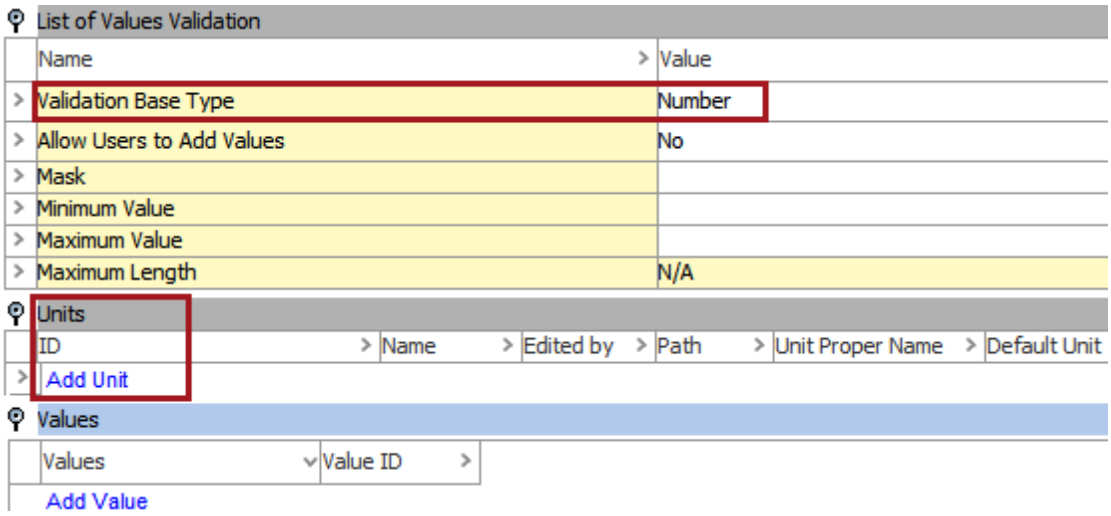
6. **Optional**: Use IDs on Values allows you to apply an ID on each value in the LOV. A Value ID is useful when importing product data and applying a certain value with a specific ID. If **Allow Users to add Values** is set to **No**, set **Use IDs on Values** to **Yes**.

   Setting **Use IDs on Values** to **Yes** enables the **Value ID Pattern** field, where a pattern can be keyed to autogenerate IDs. A simple pattern could be [id], which will autogenerate a unique id for each value created in the LOV. The pattern [id] can be combined with a prefix or suffix if needed.

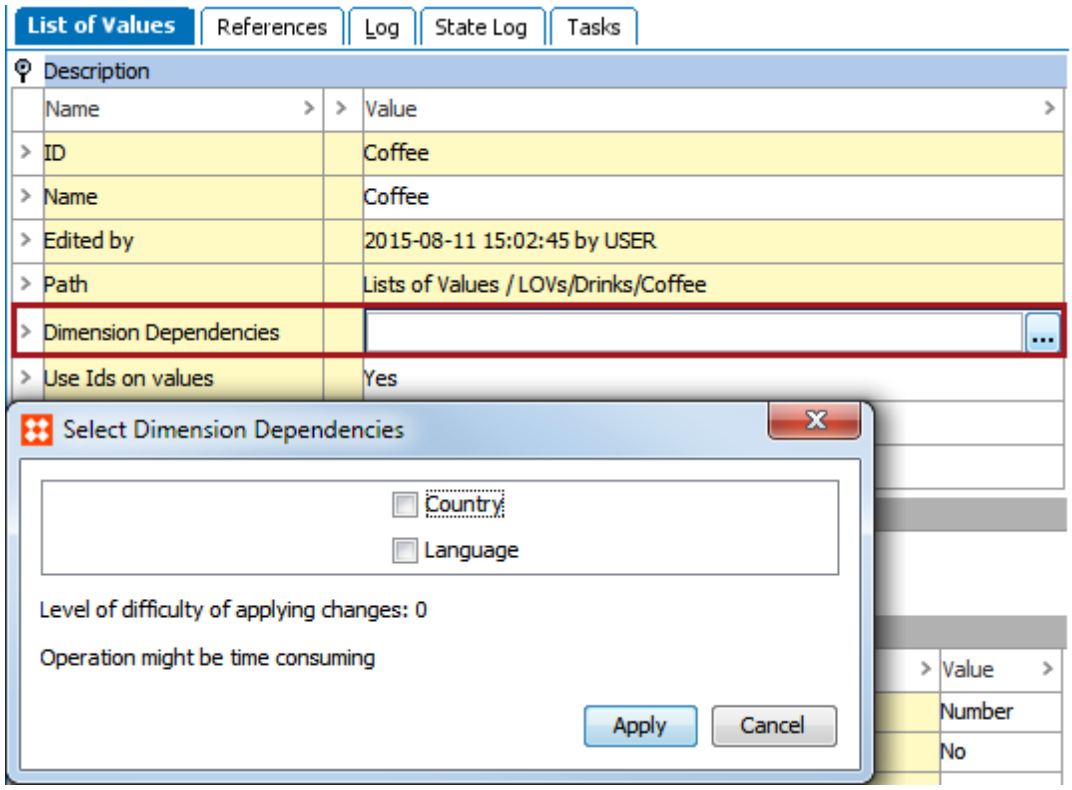


Note: If you set **Use Ids on Values** to **Yes** on an LOV that already contains values, a dialog displays with options to generate IDs on existing values.

7. **Optional**: is necessary, in the **Units** field, select one or more Units. The Unit field is only shown in LOV editor if a numeric validation base type has been selected. For more information, see the **Units** section of the **System Setup / Super User Guide** documentation.



8. **Optional**: in the **Dimension Dependencies** field, click the ellipsis button **(…)**, and select one or more dimensions. For more information, see the **Changing Dimension Dependency** section of the **System Setup / Super User Guide** documentation.



9. Add all required values to the LOV by following the steps in **Adding Values to an LOV**.
10. Assign the LOV to the relevant attributes to allow selection of the values.

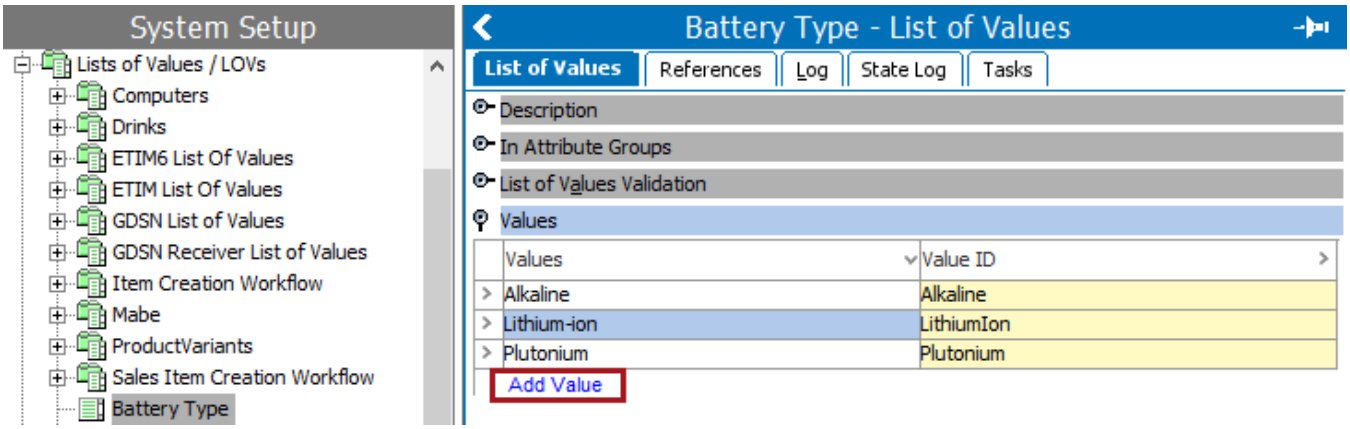
Adding Values to an LOV

Adding new values to an LOV can be done one-at-a-time or in bulk. However, with the necessary privileges, values can always be added to the LOV from the System Setup tab.

Adding a Single LOV Value

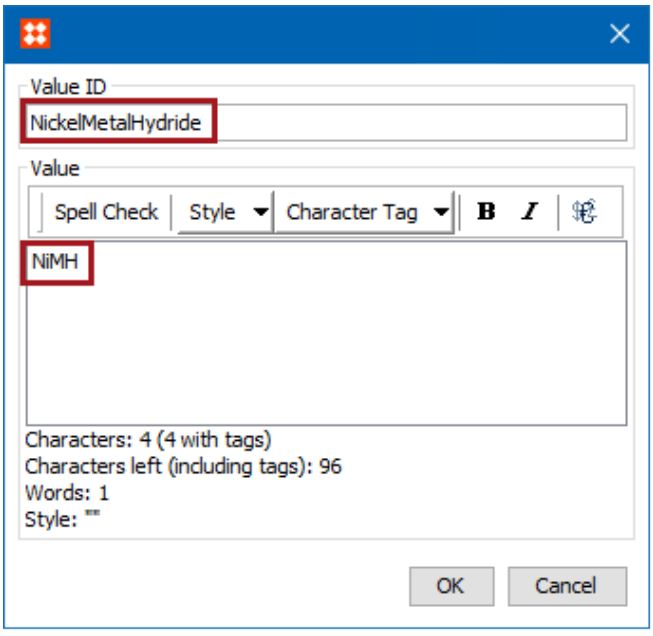
When only a few new values are required, it is simple to add them individually from the System Setup tab.

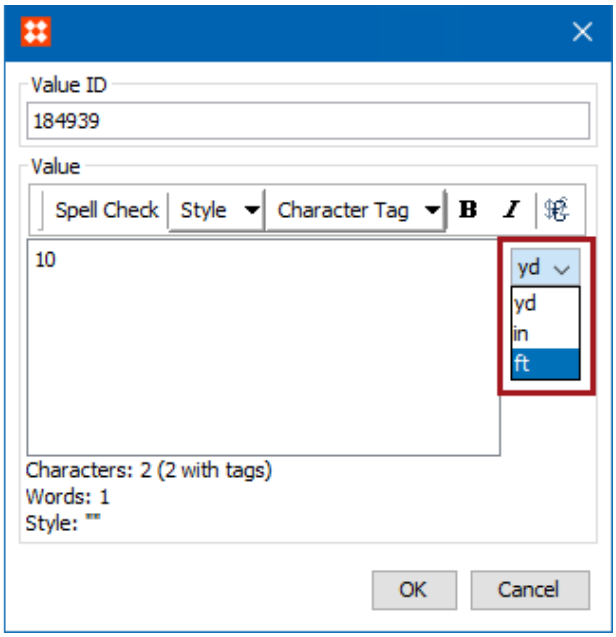
1. In **System Setup**, select the relevant LOV, open the Values flipper and click the **Add Value** link.



2. Enter a **Value ID** (if required by the LOV's Use ValueID parameter) and a **Value**.

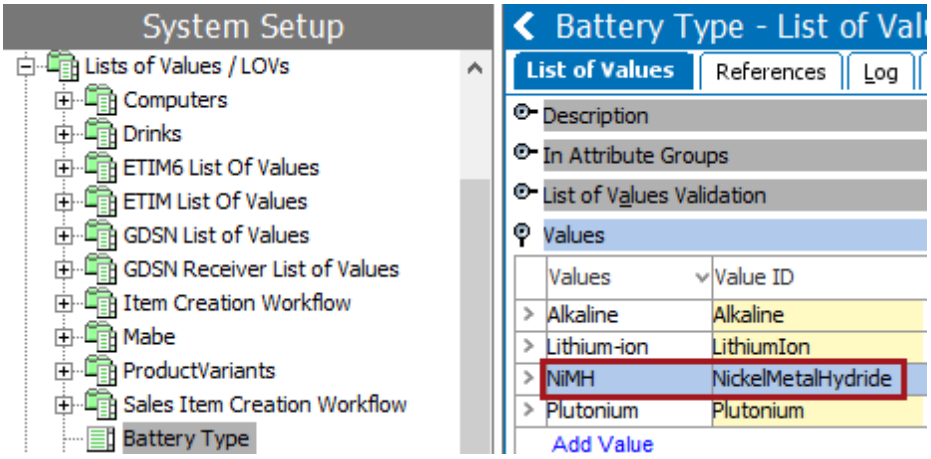
   If units are defined for the LOV, select the appropriate unit for the value from the available dropdown.





and click the **OK** button to add it to the LOV

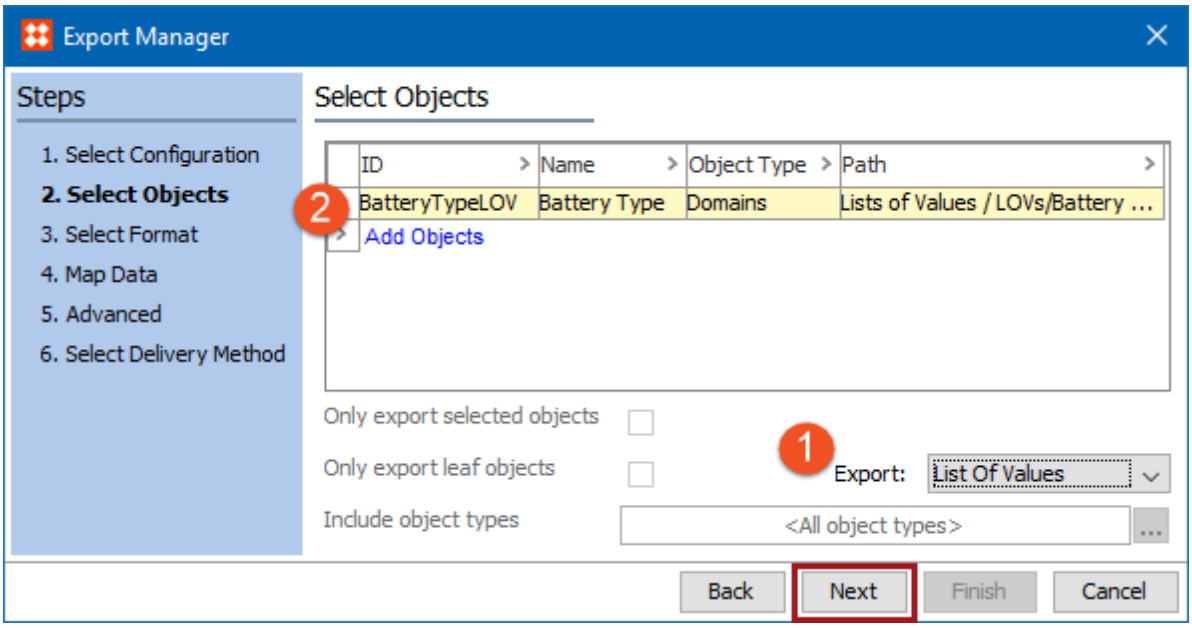
3. Verify the new value is displayed for the LOV.



Adding a Group of LOV Values

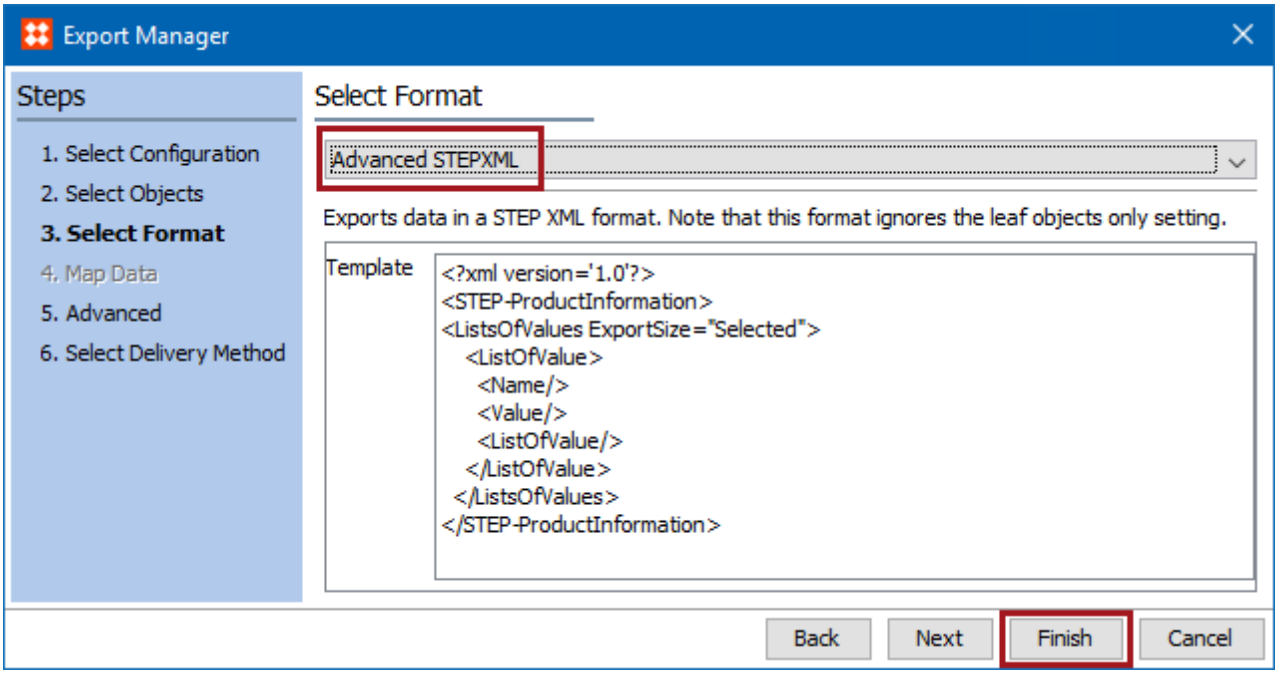
When many new values are required, it may be easier to import the values using STEPXML. This process involves exporting the existing LOV to create a STEPXML file, modifying the file to include the new values, and then importing those new values back into STEP.

1. File > Export > Data opens Export Manager. On step 2 Select Objects:
   - In the Export field, select **List of Values**.
   - In the objects list, use the **Add Objects** link to select the LOV (object type of Domain) that will get the new values.
   - Click the **Next** button.

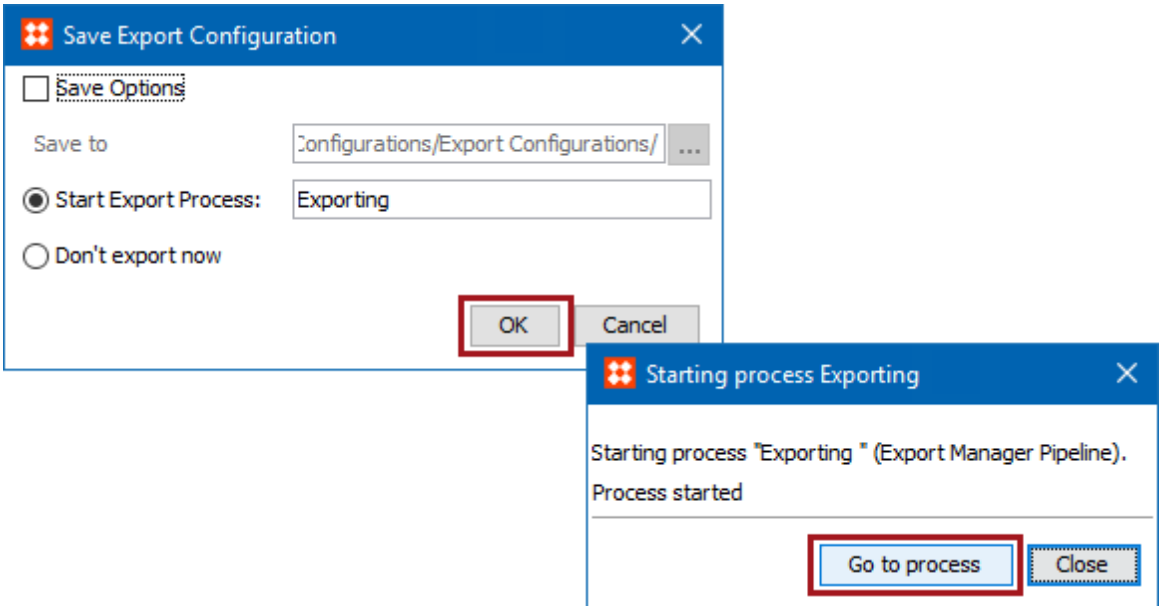


For details about each step and parameter of the Export Manager wizard, see the **Data Export Manager Wizard** section of the **Exporting Data** documentation.

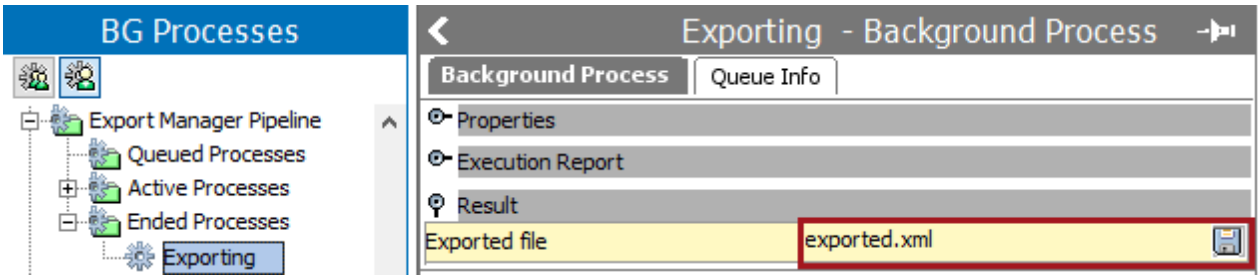
2. On step 3 Select Format, select **Advanced STEPXML** and click the **Finish** button.



3. On the Save Export Configuration dialog, click **OK** to start the export, and on the Starting process dialog, click **Go to process.**



4. On the displayed background process editor, open the output file.



5. In the output file, notice the **UseValueID** setting and the **Value** tags.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Configuration:
<?xml version='1.0'?>
<STEP-ProductInformation ResolveInlineRefs="true">
<ListsOfValues ExportSize="Selected">
    <ListOfValue>
      <Name/>
      <Value/>
      <ListOfValue/>
    </ListOfValue>
  </ListsOfValues>
</STEP-ProductInformation>
Export from Primary Product Hierarchy
Classifications All
Products All
Assets All
-->
<STEP-ProductInformation ExportTime="2016-09-08 16:27:37" ExportContext="Context1" ContextID="Context1" WorkspaceID="Main" UseContextLocale="false">

  <ListsOfValues>
    <ListOfValue ID="BatteryTypeLOV" ParentID="List Of Values group root" AllowUserValueAddition="false" UseValueID="true" Selected="true" Referenced="true">
      <Name>Battery Type</Name>
      <Validation BaseType="text" MinValue="" MaxValue="" MaxLength="100" InputMask=""/>
      <Value ID="Plutonium">Plutonium</Value>
      <Value ID="LithiumIon">Lithium-ion</Value>
      <Value ID="NickelMetalHydride">NiMH</Value>
      <Value ID="Alkaline">Alkaline</Value>
    </ListOfValue>
  </ListsOfValues>
</STEP-ProductInformation>
```

Additionally, if **Units** are being used by the LOV, notice the valid options.

```
<STEP-ProductInformation ExportTime="2016-09-12 16:47:34" ExportContext="Context1" ContextID="Context1" WorkspaceID="Main" UseContextLocale="false">

  <ListsOfValues>
    <ListOfValue ID="Variants" ParentID="List Of Values group root" AllowUserValueAddition="false" UseValueID="true" IDPattern="[id]" Selected="true" Referenced="true">
      <Name>Variants</Name>
      <Validation BaseType="number" MinValue="" MaxValue="" MaxLength="" InputMask="">
        <UnitLink UnitID="unece.unit.INH"/>
        <UnitLink UnitID="unece.unit.YRD"/>
        <UnitLink UnitID="unece.unit.FOT"/>
      </Validation>
      <Value ID="184941" UnitID="unece.unit.FOT">15</Value>
      <Value ID="184940" UnitID="unece.unit.YRD">10</Value>
      <Value ID="184942" UnitID="unece.unit.INH">27</Value>
    </ListOfValue>
  </ListsOfValues>
</STEP-ProductInformation>
```

6. Copy the output text file and save it with a **new name** to create an **import** file.
7. Create new value data in Excel, using a column to hold each individual element of the required STEPXML tags and one additional column to hold the concatenated string.
   - If the LOV parameter **UseValueID="true"**, the Value tag must include a ValueID as shown in column B below, in addition to the actual value text, shown in column D. The result of the concatenate function in column F combines all data for the XML file in the expected format.

	A	B	C	D	E	F
1	<Value ID="	BatType1	">	Battery Type 1	</Value>	<Value ID="BatType1">Battery Type 1</Value>
2	<Value ID="	BatType2	">	Battery Type 2	</Value>	<Value ID="BatType2">Battery Type 2</Value>

- If the LOV parameter **UseValueID="false"**, include only the new value text, as shown in column B below. The result of the concatenate function in column D shows all data for the XML file in the expected format.

	A	B	C	D
1	<Value ID>	Battery Type 1	</Value>	<Value ID>Battery Type 1</Value>
2	<Value ID>	Battery Type 2	</Value>	<Value ID>Battery Type 2</Value>

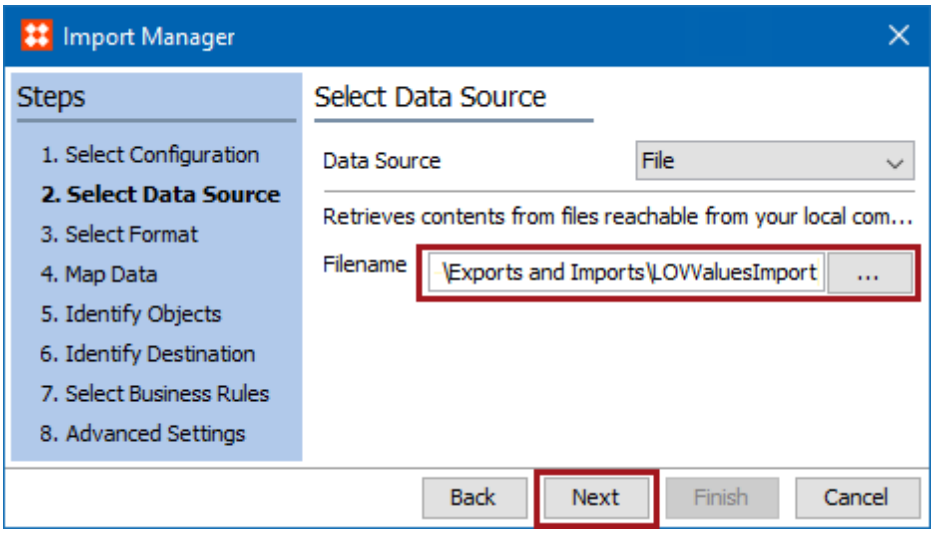
- If the LOV uses **Units**, include the ValueID if necessary as shown in column B below, the UnitID as shown in column E, and the new value text, as shown in column G. The result of the concatenate function in column I shows all data for the XML file in the expected format.

	A	B	C	D	E	F	G	H	I
1	<Value ID="	Var1	"	UnitID="	unece.unit.INH	">	32	</Value>	<Value ID="Var1" UnitID="unece.unit.INH">32</Value>
2	<Value ID="	Var2	"	UnitID="	unece.unit.FOT	">	45	</Value>	<Value ID="Var2" UnitID="unece.unit.FOT">45</Value>
3	<Value ID="	Var3	"	UnitID="	unece.unit.INH	">	58	</Value>	<Value ID="Var3" UnitID="unece.unit.INH">58</Value>
4	<Value ID="	Var4	"	UnitID="	unece.unit.YRD	">	72	</Value>	<Value ID="Var4" UnitID="unece.unit.YRD">72</Value>

8. Copy and paste the new value data into your import file and save it.

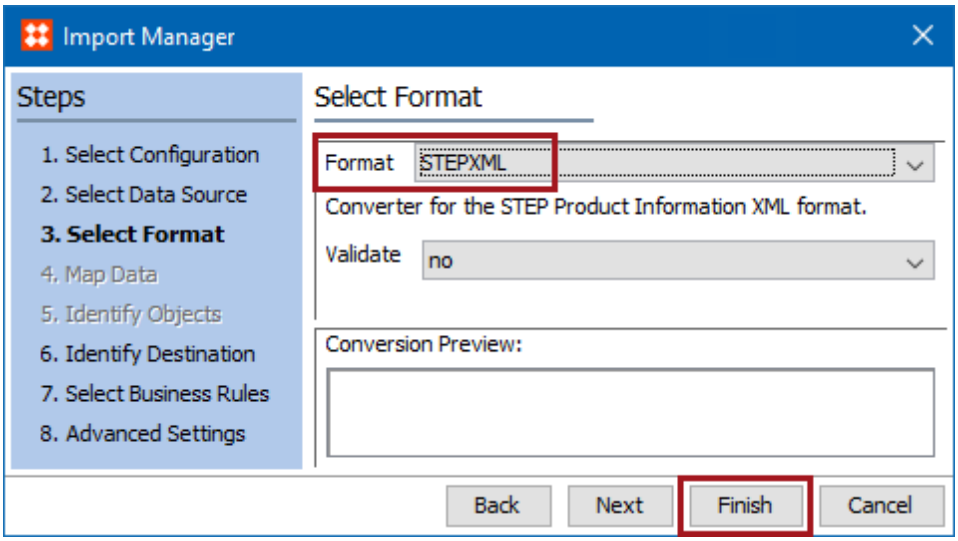
```
<STEP-ProductInformation ExportTime="2016-09-08 16:27:37" Export

  <ListsOfValues>
    <ListOfValue ID="BatteryTypeLOV" ParentID="List Of Values gr
      <Name>Battery Type</Name>
      <Validation BaseType="text" MinValue="" MaxValue="" MaxLer
<Value ID="BatType1">Battery Type 1</Value>
<Value ID="BatType2">Battery Type 2</Value>
<Value ID="BatType3">Battery Type 3</Value>
<Value ID="BatType4">Battery Type 4</Value>
<Value ID="BatType5">Battery Type 5</Value>
<Value ID="BatType6">Battery Type 6</Value>
<Value ID="BatType7">Battery Type 7</Value>
<Value ID="BatType8">Battery Type 8</Value>
<Value ID="BatType9">Battery Type 9</Value>
<Value ID="BatType10">Battery Type 10</Value>
    </ListOfValue>
  </ListsOfValues>
</STEP-ProductInformation>
```

9. File > Import > Data opens Import Manager, on step 2 Select Data Source:
   - In the Filename field, select the **import file name**.
   - Click the **Next** button.

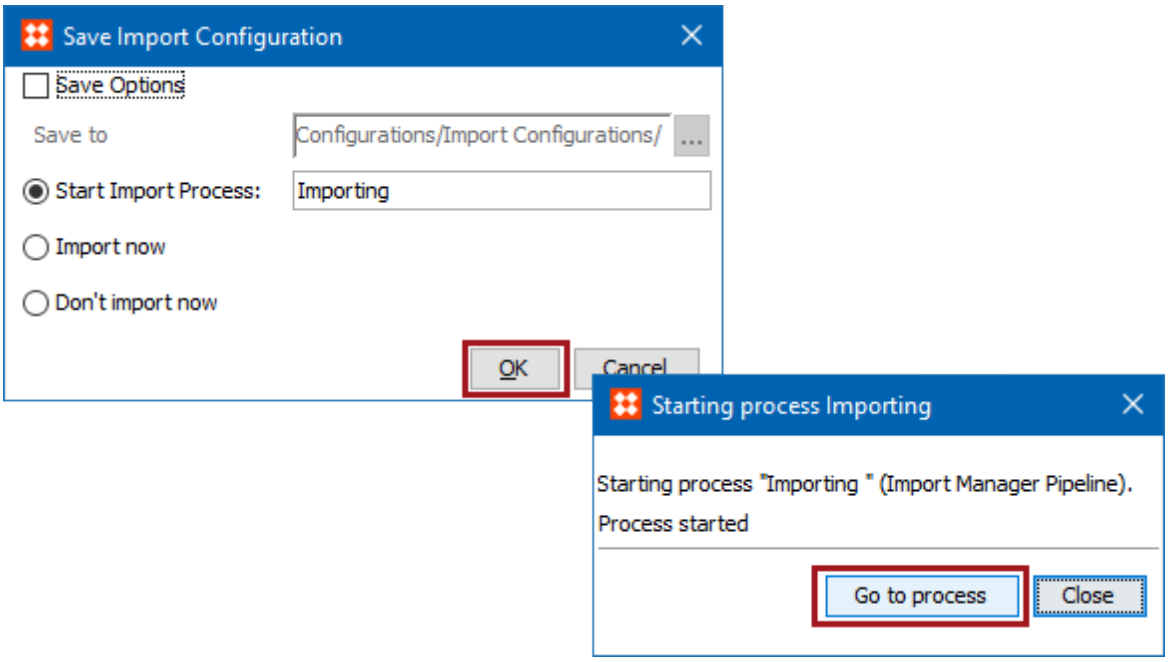


For details about each step and parameter of the Import Manager wizard, see the **Data Import Manager Wizard** section of the **Importing Data** documentation.

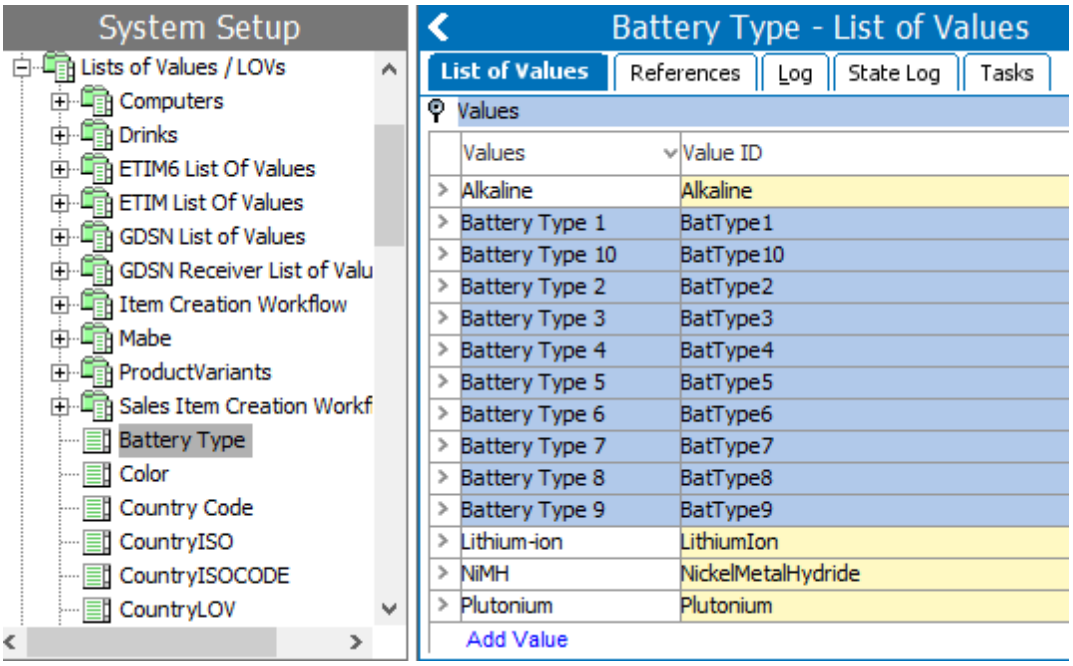
10. On step 3 Select Format, select **STEPXML** and click the **Finish** button.



11. On the Save Import Configuration dialog, click **OK** to start the import, and on the Starting process dialog, click **Go to process.**



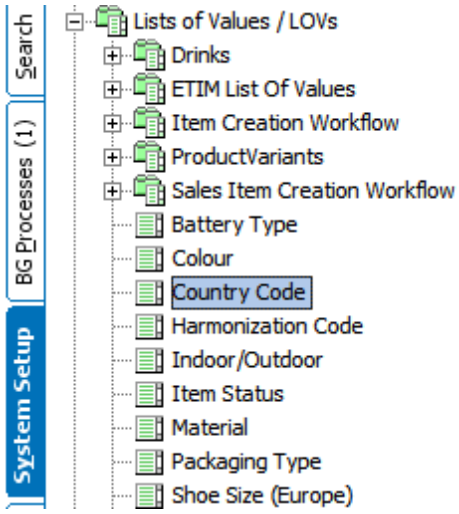
12. On the displayed background process editor, verify the import succeeded.
13. In System Setup, verify the new values are displayed for the LOV.



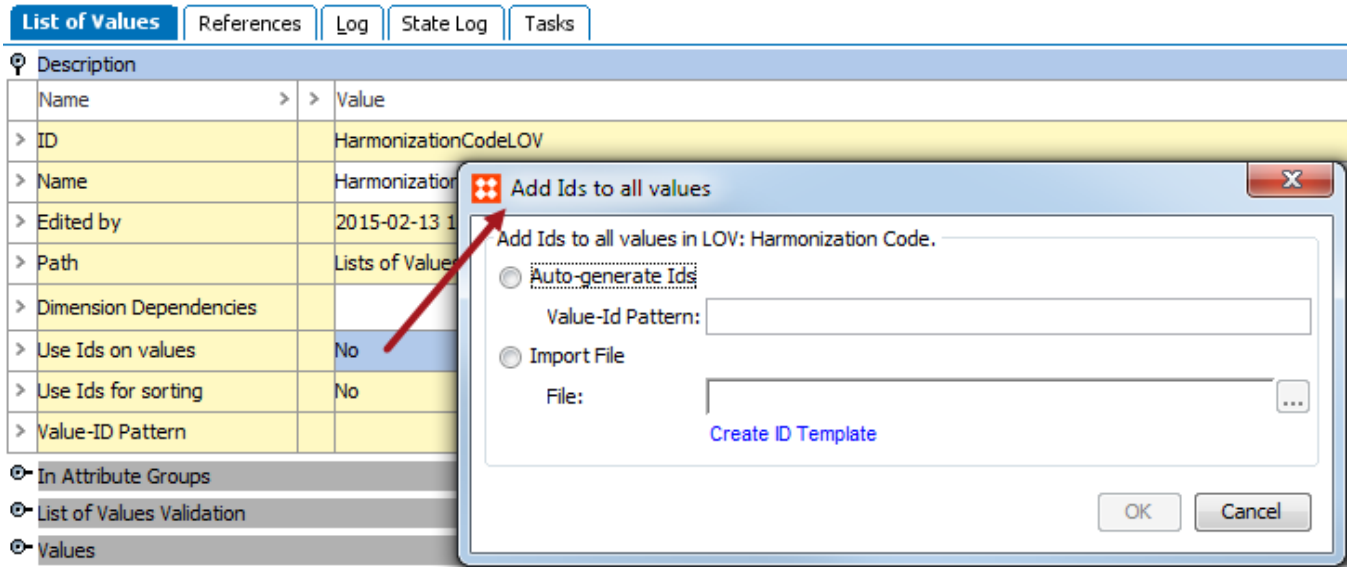
Adding IDs to Values in LOV

For LOVs that already contain values, Value ID generation adds an ID to all values in the LOV.

1. In **System Setup**, select the relevant LOV.



2. On the List of Values tab, double-click the value column of the **Use Ids on values** field.

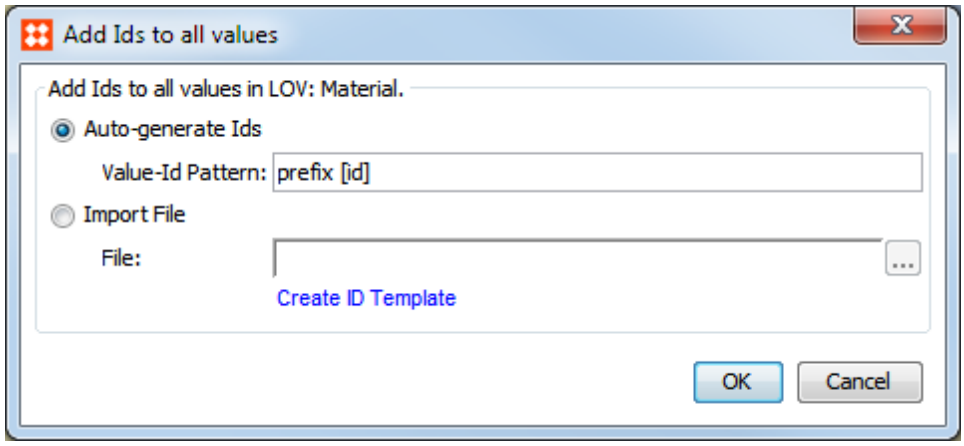


3. On the **Add Ids to all value** dialog, select the radio button to determine how IDs are created.

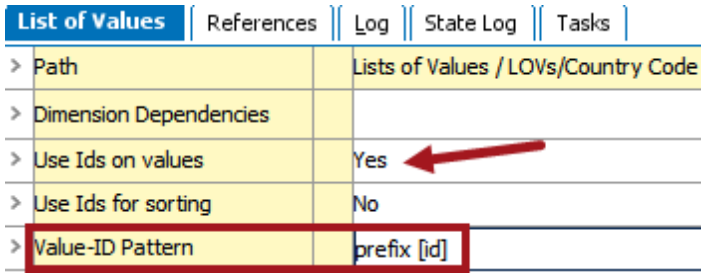
Auto-generate Ids option

This option allows customization of the ID by using a specified value with an auto-generated ID.

1. Select the Auto-generate Ids radio button.
2. In the Value-ID Pattern field, add the variable [id]. [id] is case sensitive. Additional text can precede or follow the [id], if needed.



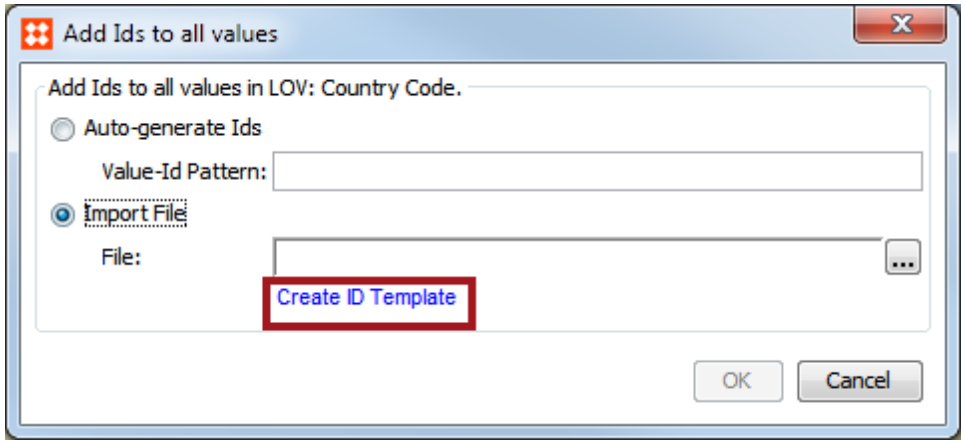
3. Click **OK** to apply the Value-ID Pattern on existing values.



Import File option

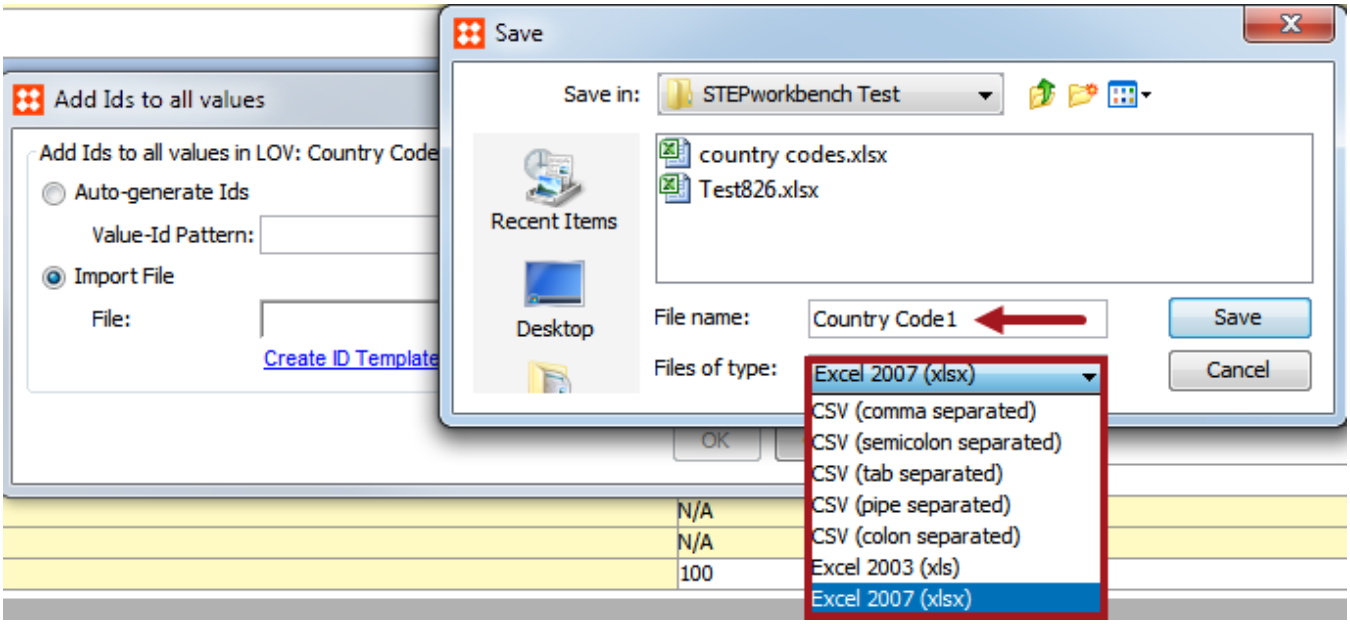
This option uses the selected import file to determine the value for each ID.

1. Select the Import File radio button.
2. Click the **Create ID Template** link.

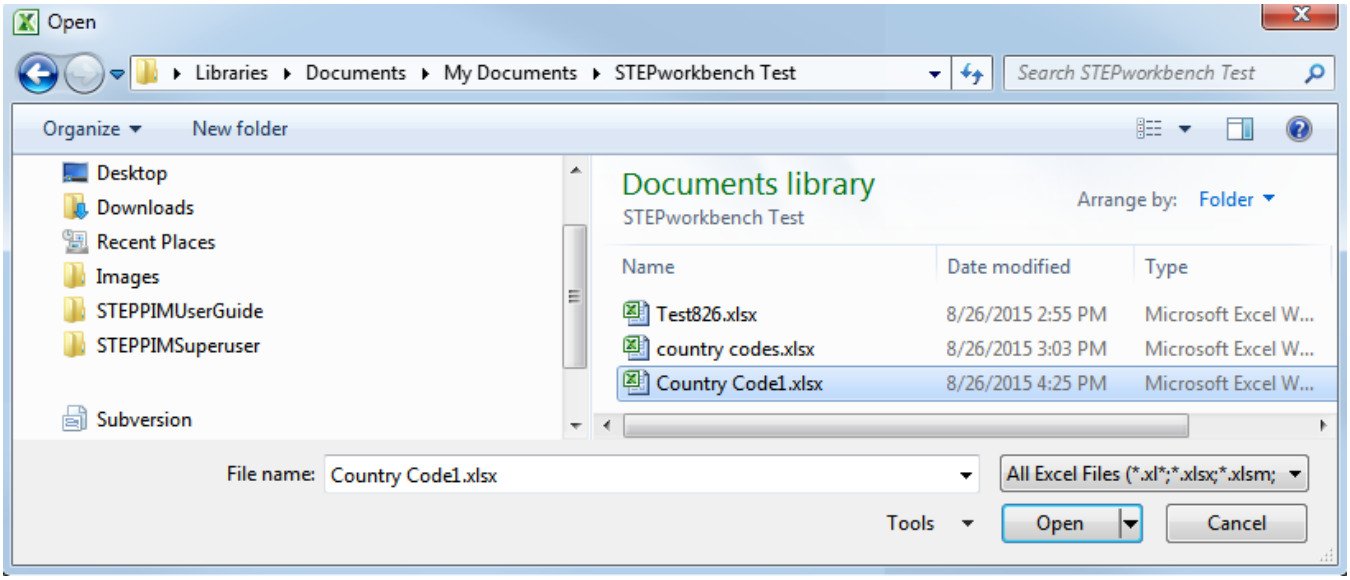


3. Choose destination of the saved template and name the template.

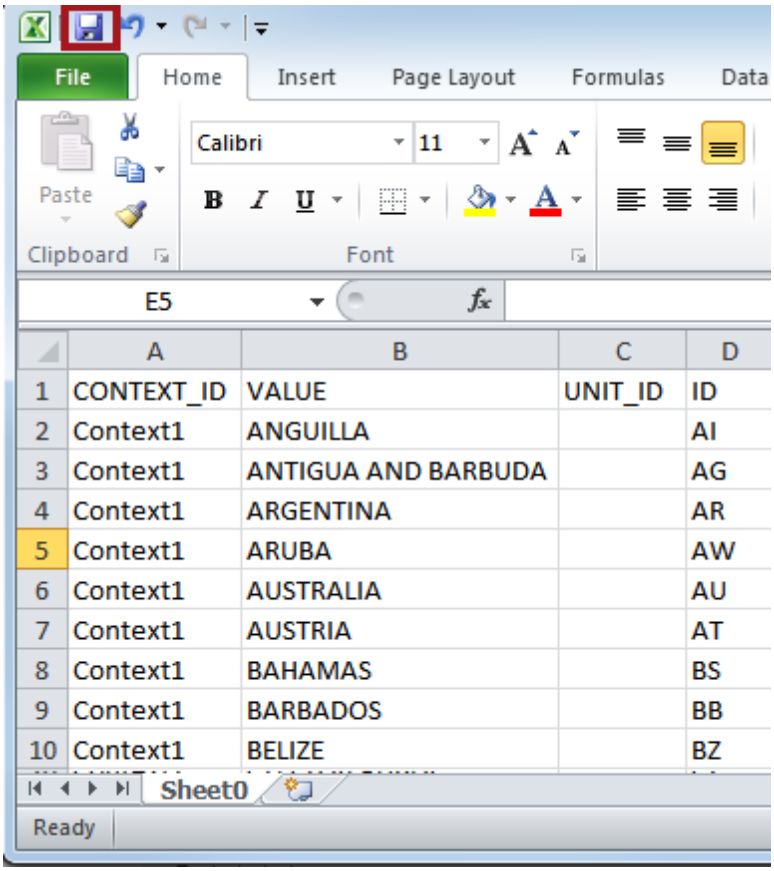
Note: An ID template can be saved in CSV (comma, semicolon, tab, pipe or color separated) or Excel (2003 or 2007) format.



4. Click **Save**, leaving the **Add Ids to all values** dialog open.
5. Open your template using to the program you used to create it.



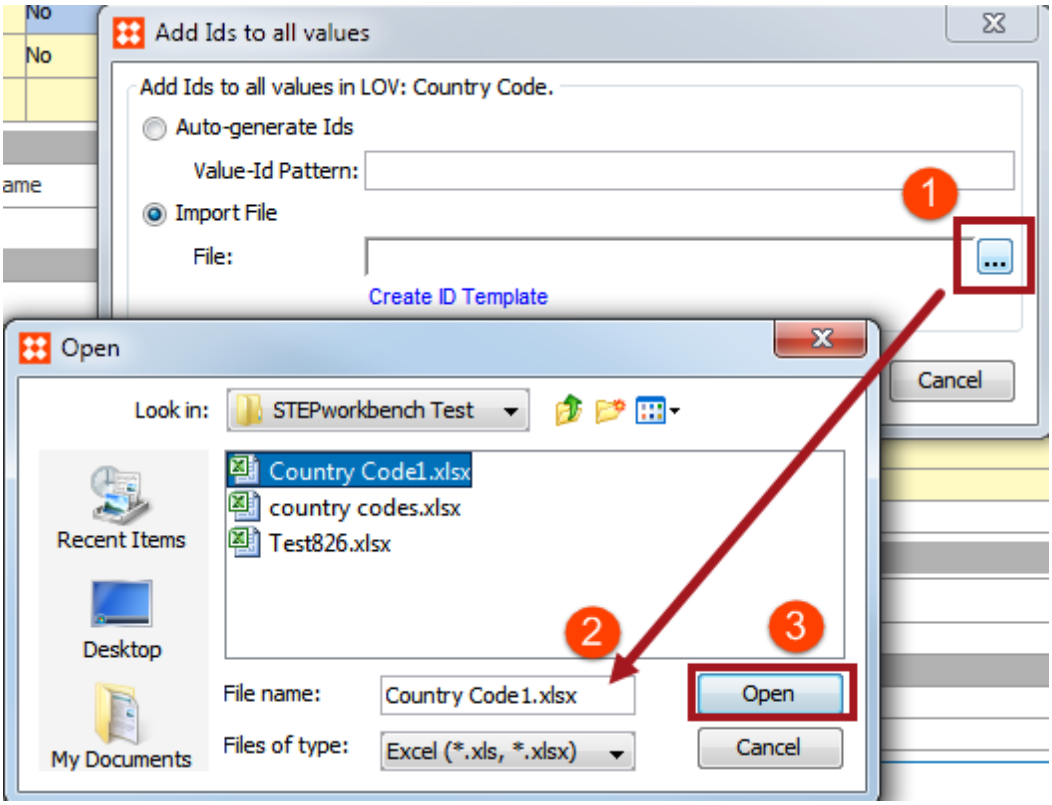
6. Once the template is open, fill in and name your ID column accordingly, and save your file.



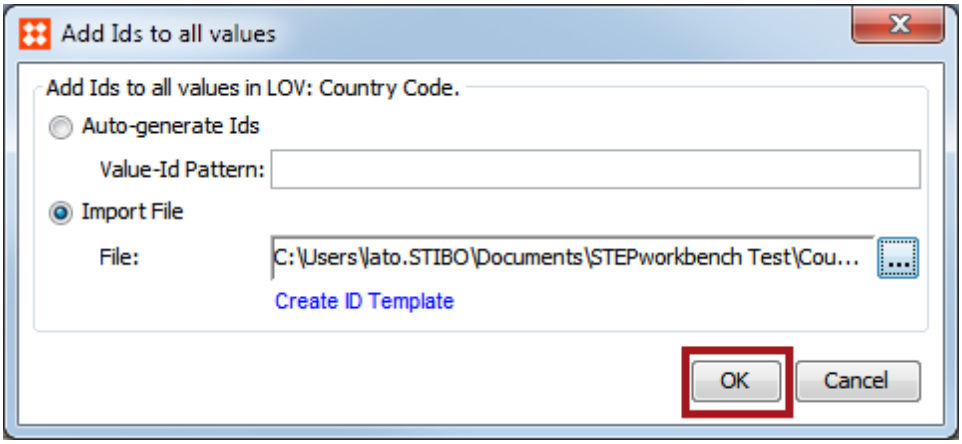
	A	B	C	D
1	CONTEXT_ID	VALUE	UNIT_ID	ID
2	Context1	ANGUILLA		AI
3	Context1	ANTIGUA AND BARBUDA		AG
4	Context1	ARGENTINA		AR
5	Context1	ARUBA		AW
6	Context1	AUSTRALIA		AU
7	Context1	AUSTRIA		AT
8	Context1	BAHAMAS		BS
9	Context1	BARBADOS		BB
10	Context1	BELIZE		BZ

Note: Unit ID column is only filled if there are units, such as inches, pounds, centimeters, etc.

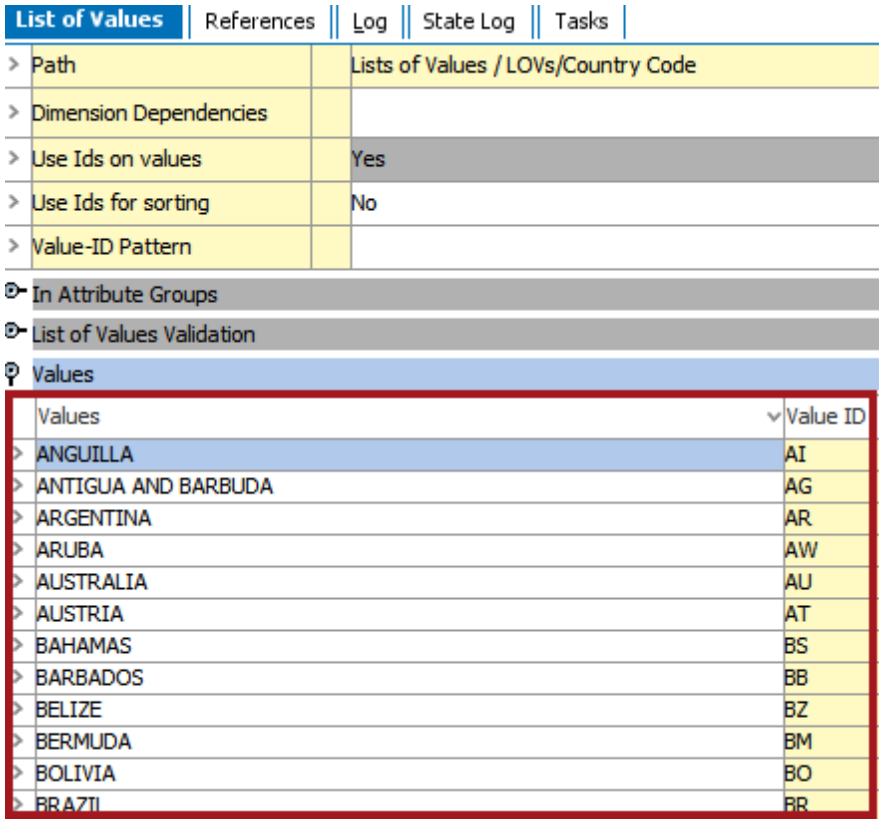
7. On the **Add IDs to all values** dialog, click the ellipsis button **(…)**and select your file. Click the **Open** button.



8. On the **Add IDs to all values** dialog click **OK**.



9. The imported data displays in the Values flipper with the new IDs.


List of Values
References
Log
State Log
Tasks
Path
Lists of Values / LOVs/Country Code
Dimension Dependencies
Use Ids on values
Yes
Use Ids for sorting
No
Value-ID Pattern
In Attribute Groups
List of Values Validation
Values
Values
Value ID
ANGUILLA
AI
ANTIGUA AND BARBUDA
AG
ARGENTINA
AR
ARUBA
AW
AUSTRALIA
AU
AUSTRIA
AT
BAHAMAS
BS
BARBADOS
BB
BELIZE
BZ
BERMUDA
BM
BOLIVIA
BO
BRAZIL
BR

Merge LOVs

Merging List of Values (LOVs) can involve the consolidation of two LOV lists or the consolidation of multiple LOV values. Ideally, LOVs and their values are unique selections. Merging is needed when duplicate lists or values exist due to different spellings, capitalization, or other insignificant discrepancies.

Merging LOVs

The Merge LOVs option can be used to combine two LOVs into a single selected master list.

For example, when two individual LOVs exist for the same set of values:

- Screwdriver Color with the values Black, Gray, and Blue
- Tool Color with the values Red, Black, Gray, and White

After the merge, a single LOV exists with the values: Red, Black, Gray, White, and Blue.

For more information, see **Merging LOVs**.

Merge Values in LOV

The Merge Values in LOV option can be used to combine all three into a single selected master value.

For example, when a single LOV has duplicate existing values:

- yes
- Yes
- YES

After the merge, a single LOV value exists for yes.

For more information, see **Merging Values in an LOV**.

Merge LOVs

Merging List of Values (LOVs) can involve the consolidation of two LOV lists or the consolidation of multiple LOV values. Ideally, LOVs and their values are unique selections. Merging is needed when duplicate lists or values exist due to different spellings, capitalization, or other insignificant discrepancies.

Merging LOVs

The Merge LOVs option can be used to combine two LOVs into a single selected master list.

For example, when two individual LOVs exist for the same set of values:

- Screwdriver Color with the values Black, Gray, and Blue
- Tool Color with the values Red, Black, Gray, and White

After the merge, a single LOV exists with the values: Red, Black, Gray, White, and Blue.

For more information, see **Merging LOVs**.

Merge Values in LOV

The Merge Values in LOV option can be used to combine all three into a single selected master value.

For example, when a single LOV has duplicate existing values:

- yes
- Yes
- YES

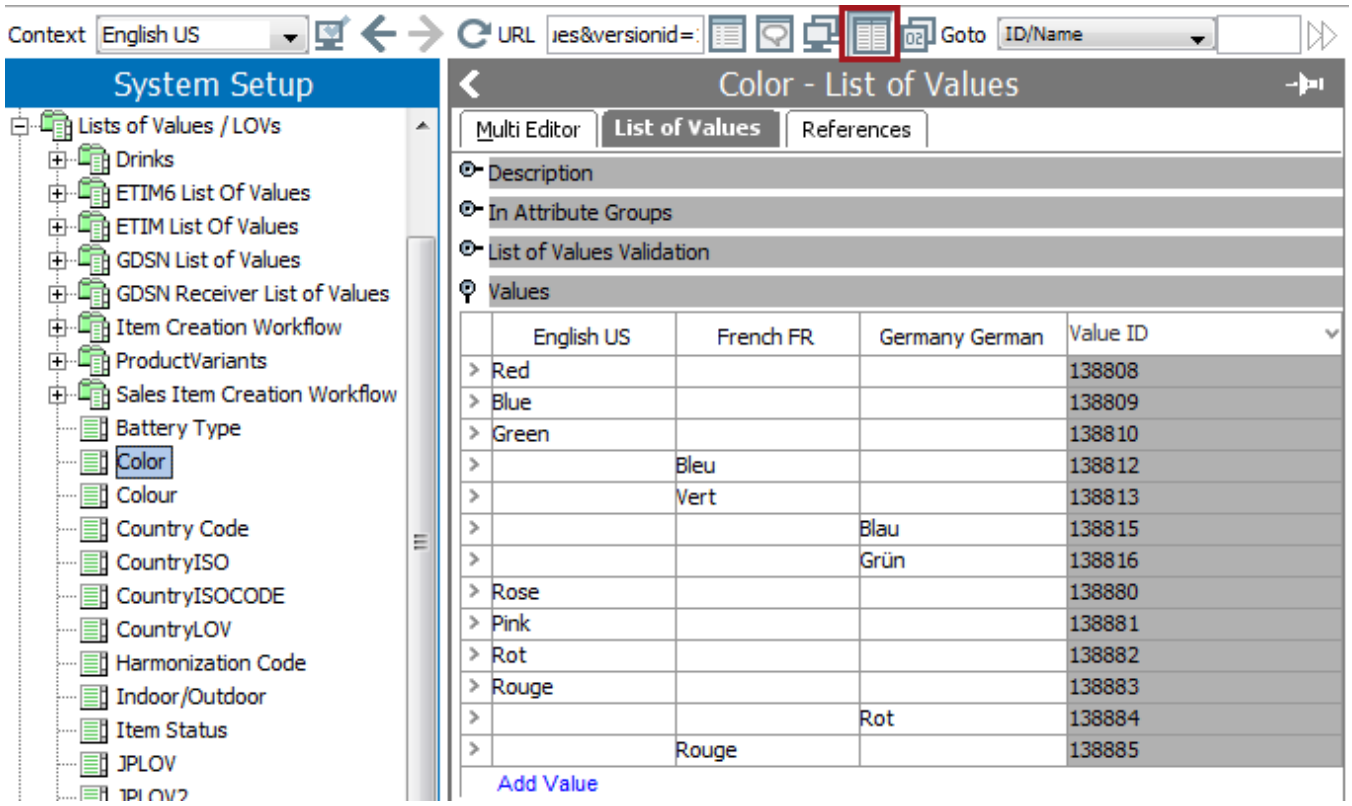
After the merge, a single LOV value exists for yes.

For more information, see **Merging Values in an LOV**.

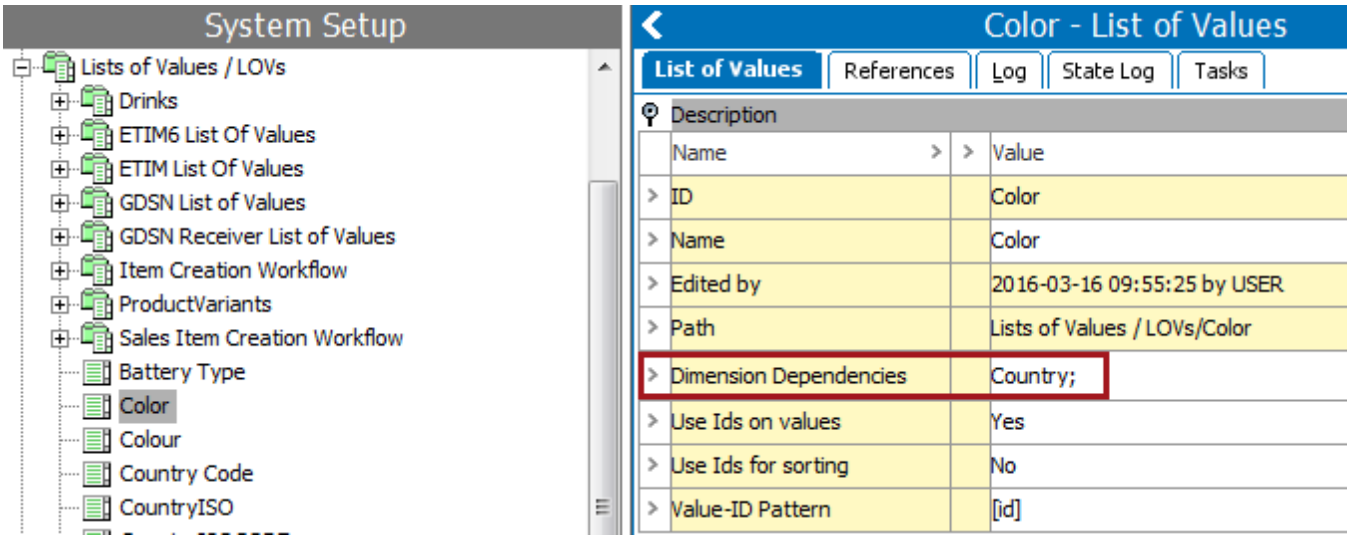
Merging Values in an LOV

If multiple sets of duplicate values exist within a single LOV, run the merge operation multiple times to eliminate each of the duplicate sets.

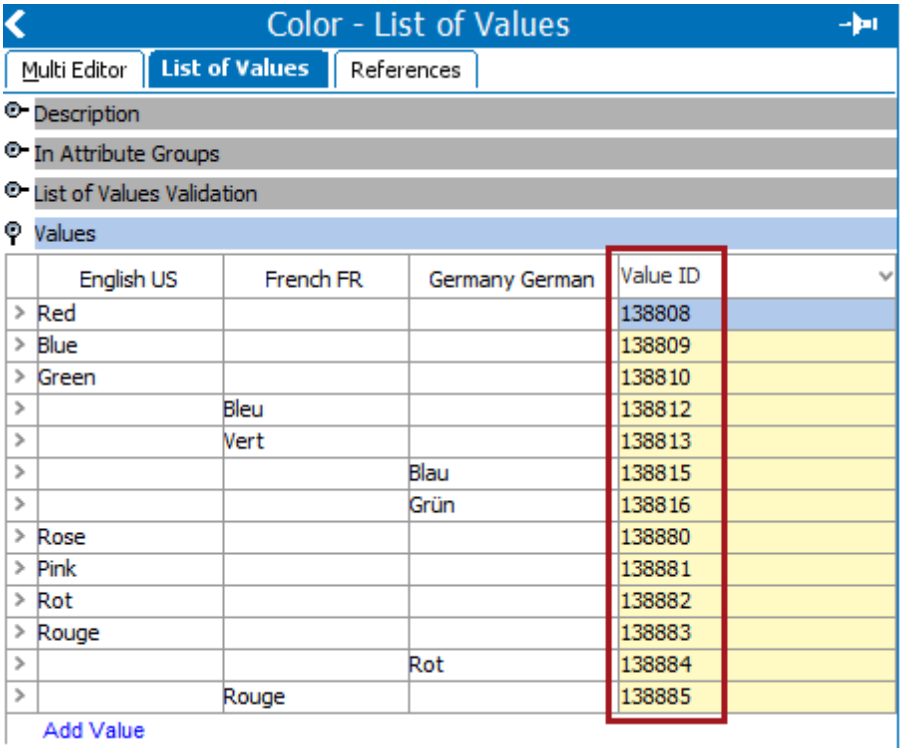
1. In workbench, use the Context option to view all available values for an LOV.



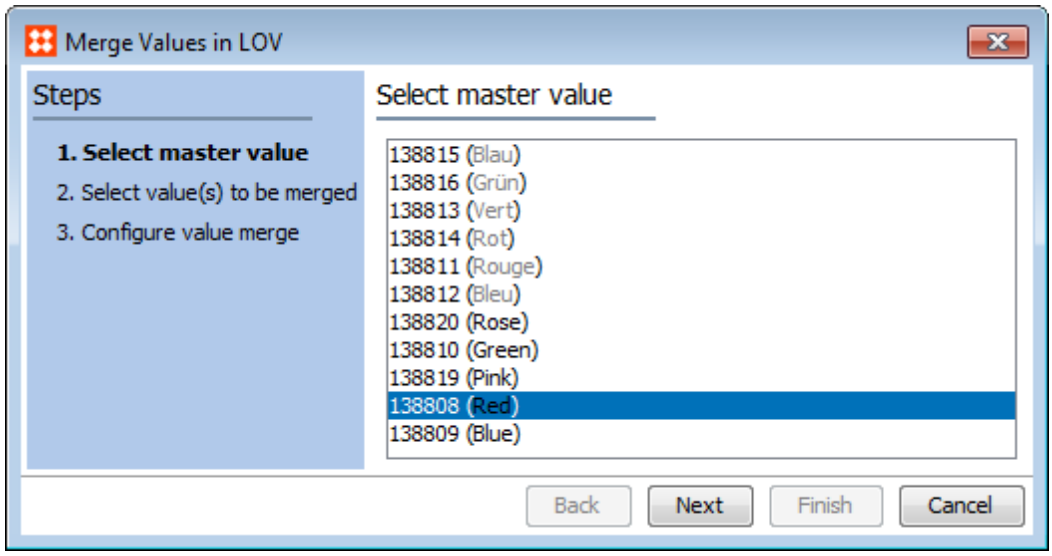
2. In **System Setup**, under List of Values / LOVs, verify that the LOV containing values to be merged has no more than one Dimension Dependency. Merge Values in LOV is not allowed for LOVs with two or more Dimension Dependencies.



3. Verify that the LOV uses Value IDs. If not, double-click the **Use Ids on values** option in the Description flipper to set Value IDs. For more information, see Adding IDs to Values in LOV.

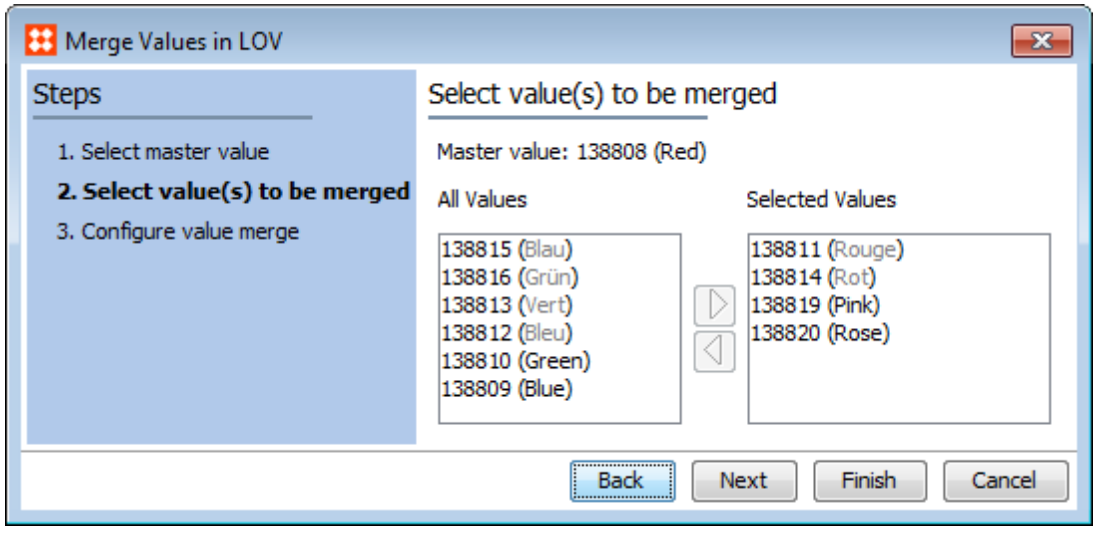


4. Select the LOV, right-click and select **Merge Values in LOV**.
5. Select the value to be the master and click the **Next** button. This is the value that will remain after the merge.

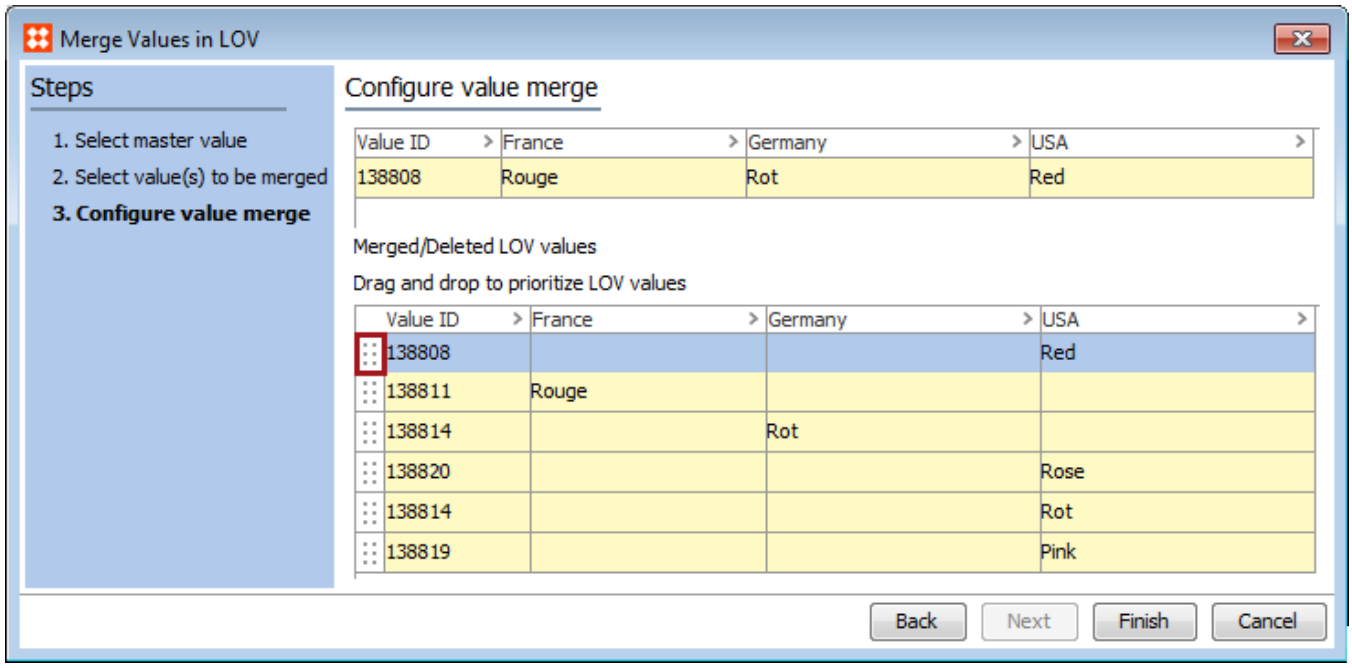


6. Identify the values to be deleted by highlighting a value in the **All Values** column, clicking the right arrow button and moving it to the **Selected Values** column. All Selected Values will be deleted.

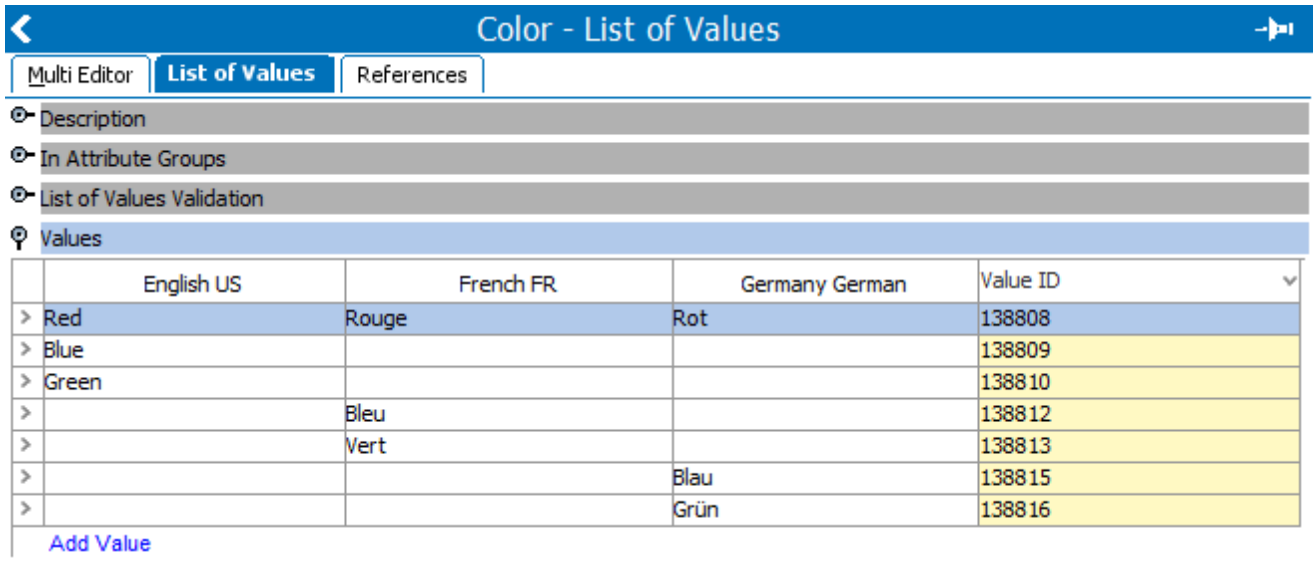
   Repeat this process for all values to be merged into the master value and click the **Next** button.



7. In the 'Merged/Deleted LOV values' table, use the handles to drag LOV values and reorder as necessary.

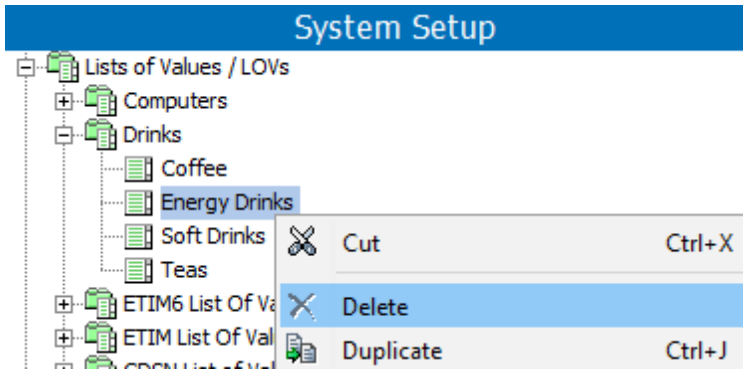


8. Click the **Finish** button to complete the merge process.

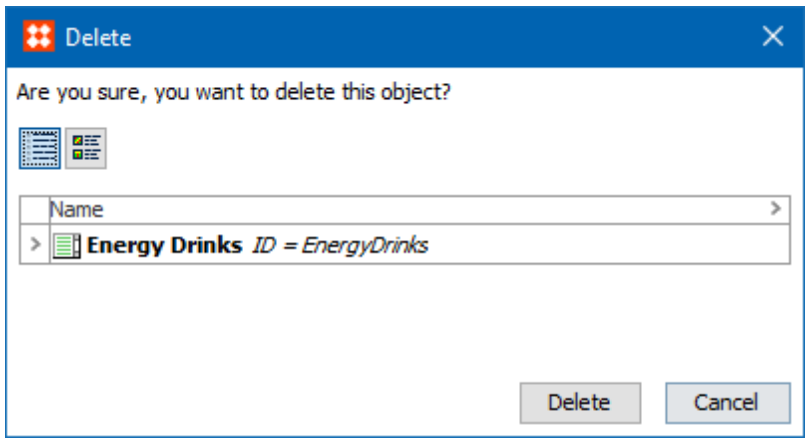


Deleting LOVs

1. In the System Setup tab, expand **List of Values / LOVs** to display the available LOVs.
2. Select the LOV to be deleted and right-click to display the menu.



3. Click **Delete** and a Delete warning displays.



4. Click the **Delete** button to delete the List of Values. Click the **Cancel** button to return to System Setup without deleting.

Managing Revisions in STEP

STEP automatically preserves all versions of a revisable object. Each time a change is made by a different user, a numbered revision is created and logged. Revisions allow for users to revert back to a previous version of a revisable object, identify who has made changes to an object (including what changed and when), and support overall auditing functionality within any system.

The object types STEP maintains revisions for are business rules, workflows, integration endpoints, products, entities, classifications, assets, standard stand-alone event queues, and Web UI configurations. While each of these objects have their own unique features, the overall functionality of creating and maintaining revisions is universal across objects.

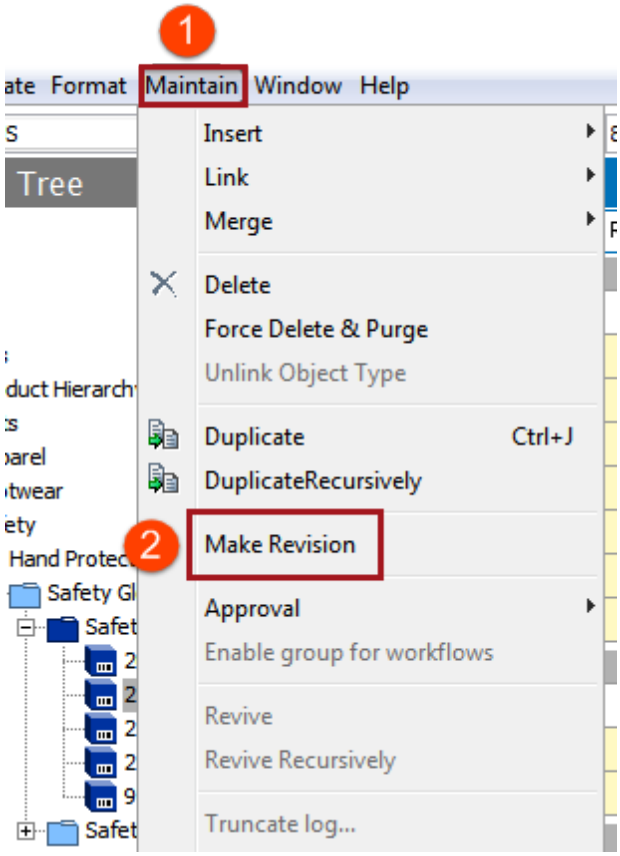
It is important to note the difference between *revisions* and *approvals*. The two concepts are related, but should not be confused. Objects that are *workspace revisable* are subject to approvals, and approving an object automatically generates a revision of it. However, objects that are *globally revisable*, such as System Setup objects like workflows or business rules, are *not* subject to approval. This means that they can have revisions made, but only via the processes described below. For more information on approving objects, see the Object Approval section of the Getting Started / STEP User Guide.

Important: It is critical that revision handling be considered for each system and that appropriate processes be put in place to clean up old revisions and ensure that the number of revisions does not exceed the expected capacity of the system and introduce performance issues.

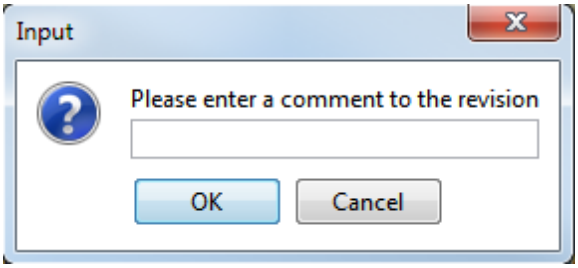
Generating Revisions

A revision can be created in one of four ways:

1. Automatically, each time a change is made by a different user.
2. Manually, by going up to **Maintain** and selecting **Make a Revision** from the drop down menu.



Manually created revisions allow for the user to enter a comment.



3. Automatically, if a user makes a change and the current revision is older than a configurable threshold, even if the same user also made the previous change (see below for more information).
4. Upon approval of an object (applicable only to workspace revisable objects).

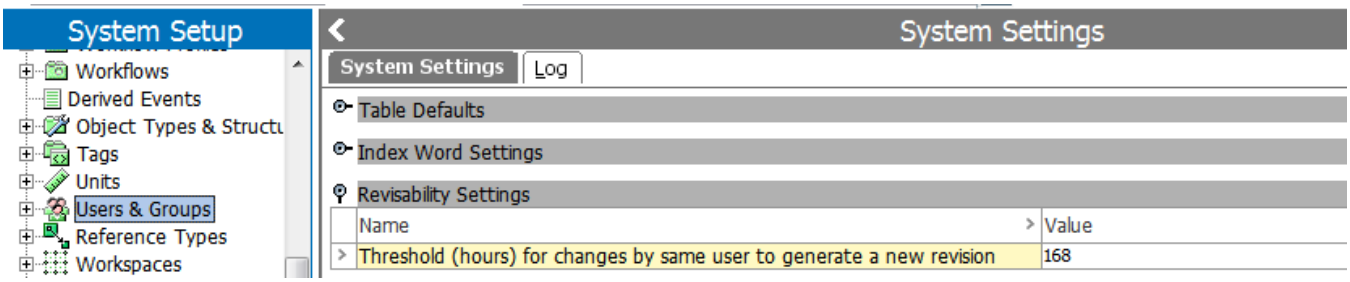
Setting the Timing Threshold for Automatic Revisions

By default, the system creates revisions on objects if a user makes a change to the object and the timing threshold (number of hours) is exceeded from when the object was first touched after the last revision was made. This is

particularly useful in cases where an object is primarily maintained by a single user and would not otherwise have revisions made without that user choosing to do so manually. The threshold functionality ensures that changes are recorded, without creating an excessive number of revisions.

The global default number of hours set for all revisable objects is 168, or one week, before a revision is created. If any user acts on the object during the timeframe specified by the threshold parameter, a revision is made when the time is exceeded. If no one acts on the object during that time, no revision is made.

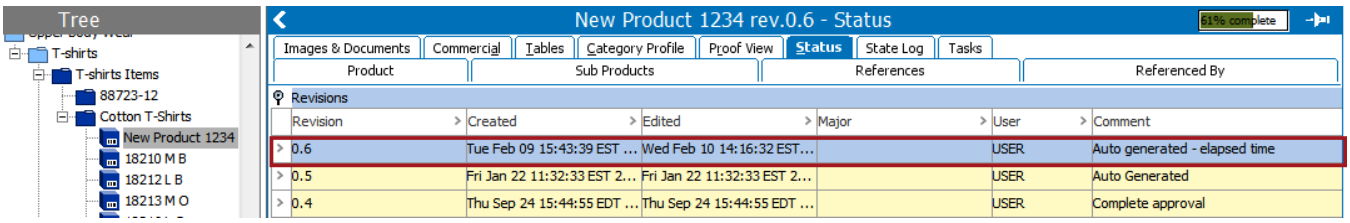
This number can be changed by first selecting the **System Setup** tab, navigating to the **Users and Groups** node, and finding the **Revisability Settings** flipper. The interval can be changed by editing the value in the Threshold parameter.



Important: The minimum allowable number of hours is 24, though there is no maximum. The parameter defaults to 168 (one week) and it is recommended to set the parameter *at or above* this number to prevent an excessive number of revisions from being created.

If the threshold parameter is left blank, the functionality for creating auto-revisions is considered disabled, and all revisions must be made manually or due to different users touching the revisable object.

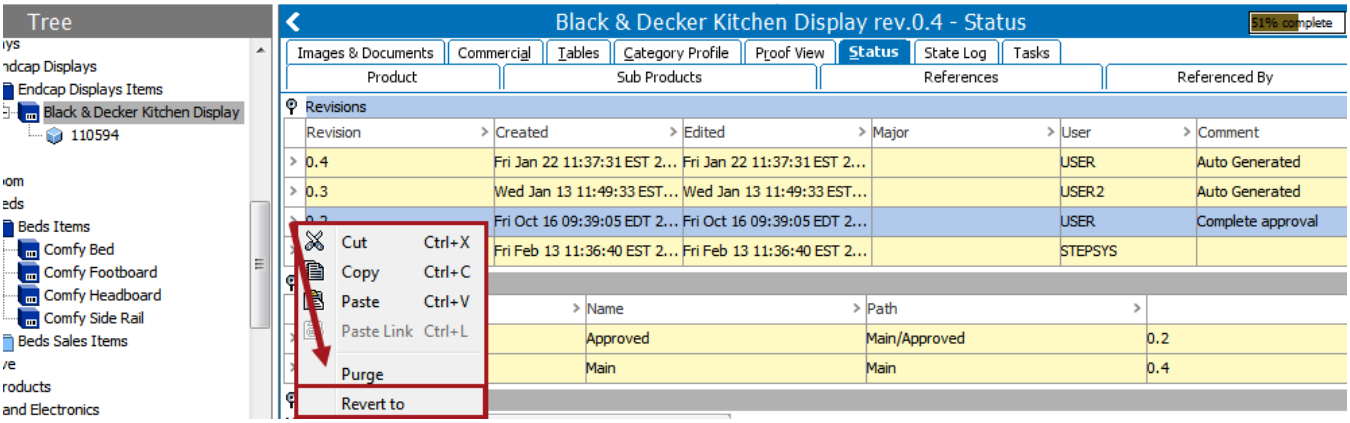
When an object does have a revision made to it due to reaching the timing specified in the threshold parameter, it is noted in the Comment column on the Status tab for that object.



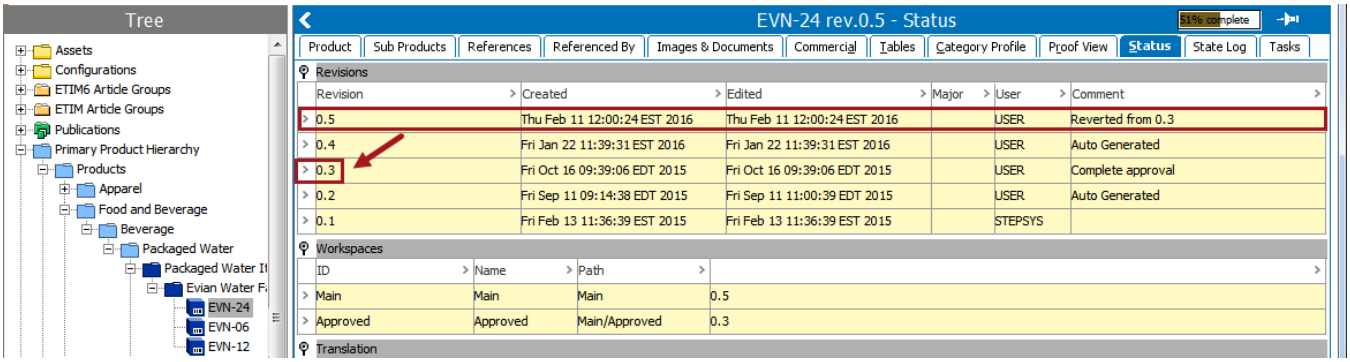
Should a user want to create a revision before the set number of hours expires, they can still do so manually via the **Maintain** menu, as described earlier in this topic.

Reverting to Past Revisions

If a user wishes to revert to a past revision, they can do so by clicking on the arrow next to the desired revision and selecting **Revert to**. Reverting to a revision actually generates a *new* revision that matches the one reverted to. It does not discard any intermediate revisions, nor is any audit trail lost.



When a version is reverted to, an automatic comment stating which revision that object has been reverted to will generate. This is intended to aid the user in tracking the history of a given object.

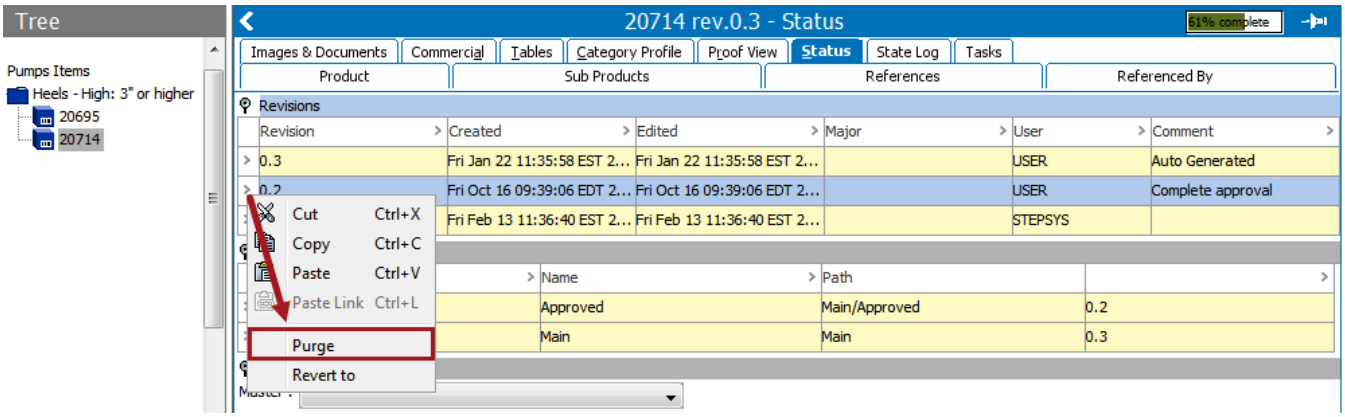


Important: In order to revert back to an old revision successfully, all configurations applied to the object must still be in place. For example, when reverting to a previous version of an integration endpoint, all related objects (e.g. attributes, references, business rules) must still be in place, else the endpoint may not work as intended when revived.

Purging Revisions

Any revision can be purged, but once a revision is purged it can not be recovered.

To purge a revision, click on the arrow next to the desired revision and select **Purge**.

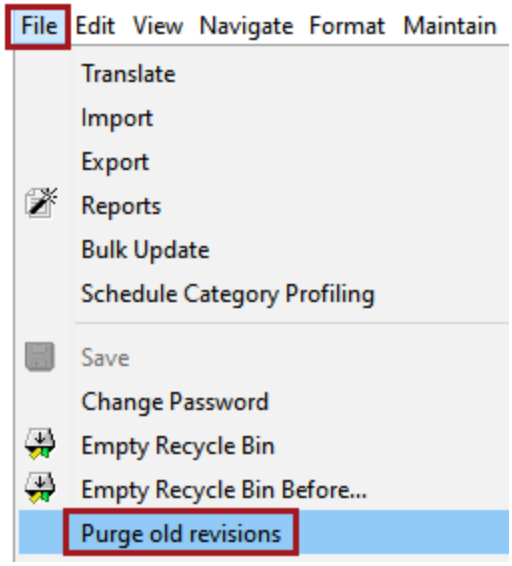


Revision Maintenance and Cleanup

Planning for maintenance and cleanup of revisions is a critical step for all implementations and is best to be carried out early on in the setup of any system.

The Revision Management event processor can be used to aid in cleanup and it is recommended that this be set up on all systems. More information on this event processor can be found in the **Event Processors** section of the **System Setup / Super User** documentation.

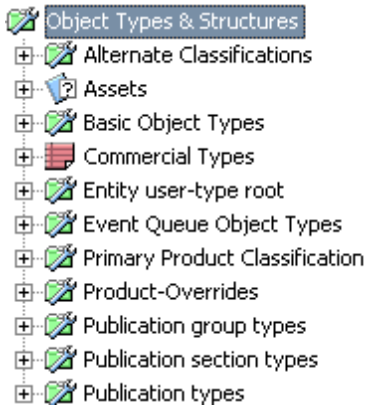
A manual revision cleanup option is also available via the File menu. It is strongly recommended that this be used for ad-hoc cleanup only and that additional automatic revision cleanup processes be in place.



Object Types and Structures Overview

Object Types allow you to define the type of system or product data that can be stored in a node. Grouping items with similar characteristics in a node enables you to identify a required items more easily.

Object Types are maintained on the **System Setup** tab under the **Object Types & Structures** node.



There are two kinds of Object types:

Object Type kind	Description
Basic Object Type	A Basic Object Type is applied to nodes that are used for structuring the *system data*. There are many Basic Object Types including: • Attribute Group • Context • Dimension • User When an Object of any Basic Object Type is created, the relevant Object Type is automatically applied. New Basic Object Types cannot be created. For the complete list of Basic Object Types and more information, see the Basic Object Types section.
Non-Basic	A non-basic Object Type applies to nodes that are used for structuring the product data

Object Type kind	Description
Object Type	hierarchies including: • Alternate Classifications • Primary Product Classification • Assets (images and documents) • Entities • Product-Overrides • Publication types, publication section types, publication group types When creating an object of these types you must manually apply the relevant Object Type. For more information, see the Object Types for Classification, Product, Assets and Entities section.

Attributes Valid for Object Types

Attributes can be made globally-valid for all Objects of a specific Object Type. Description Attributes can be applied to all Object Types.

A Description Attribute made valid for an Object Type will appear as a field in **Description**.

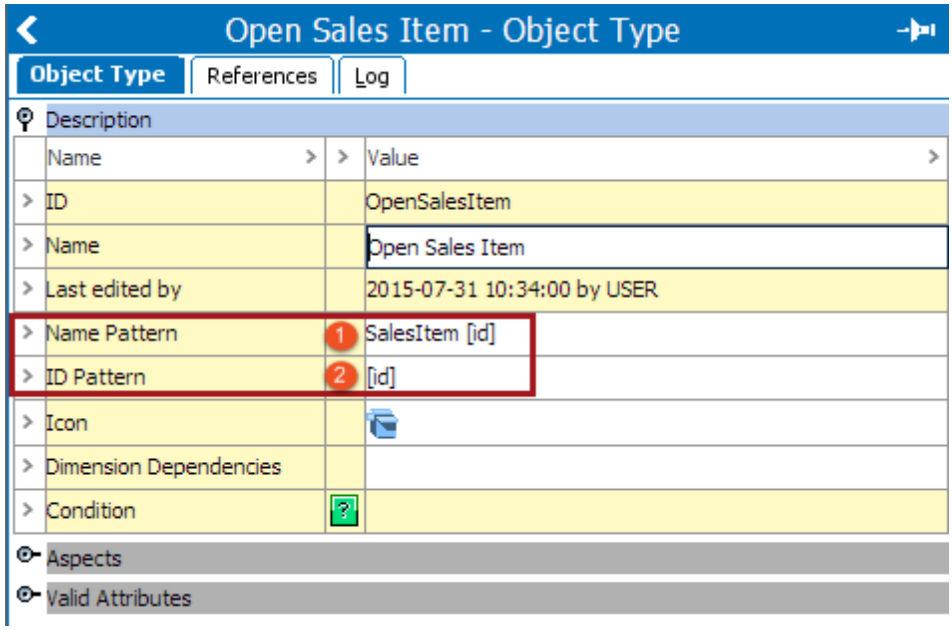
For more information, see the Linking Description Attributes to an Object Type section.

Autogenerate using Name Pattern and ID Pattern

The Name text and ID text of a node can be autogenerated by supplying a pre-defined suggestion. An Object Type with a Name Pattern and/or ID Pattern will suggest the pattern text when a new node of that Object Type is created.

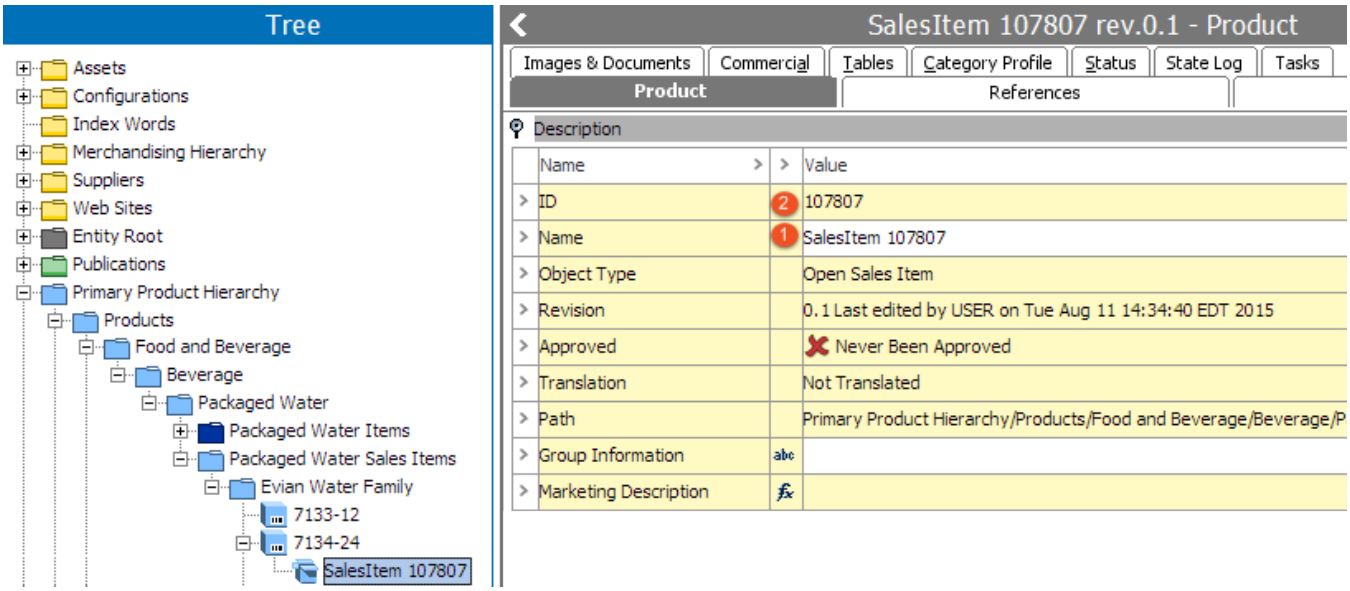
- Static text added in the Object Type displays in the Name or ID field when a new node is created.
- Unique incremental counter automatically displays the next available number in a sequence when the tag '[id]' is used. This counter is extremely advantageous when used for an ID, because it guarantees a unique ID every time a new node of that Object Type is created.

Data that is autogenerated can include an incremental counter and/or static text. As illustrated below, a new node of the Object Type 'Product' has a pattern of 'SalesItem [id]' for the Name field and [id] for the ID field.



When a new node is created on the Tree for the Open Sales Item Object Type:

- The Name is 'SalesItem 107807' because 107807 is the next unique number available to STEP Workbench.
- The ID number is the same as in the Name field since the same tag was used in both Pattern fields on the Object Type.



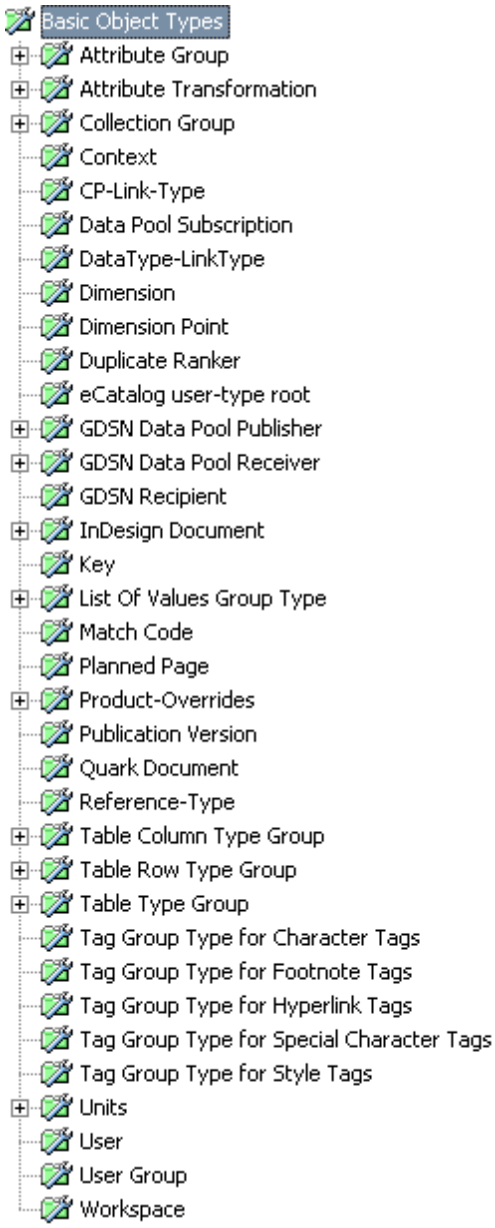
Note: For the Name field, the autogenerated text is only a suggestion and may be overwritten. The ID field autogenerated text cannot be changed once the new node is saved.

Basic Object Types

A Basic Object Type is a node used for structuring the system data, for example, Attributes, Contexts, Users, Tables.

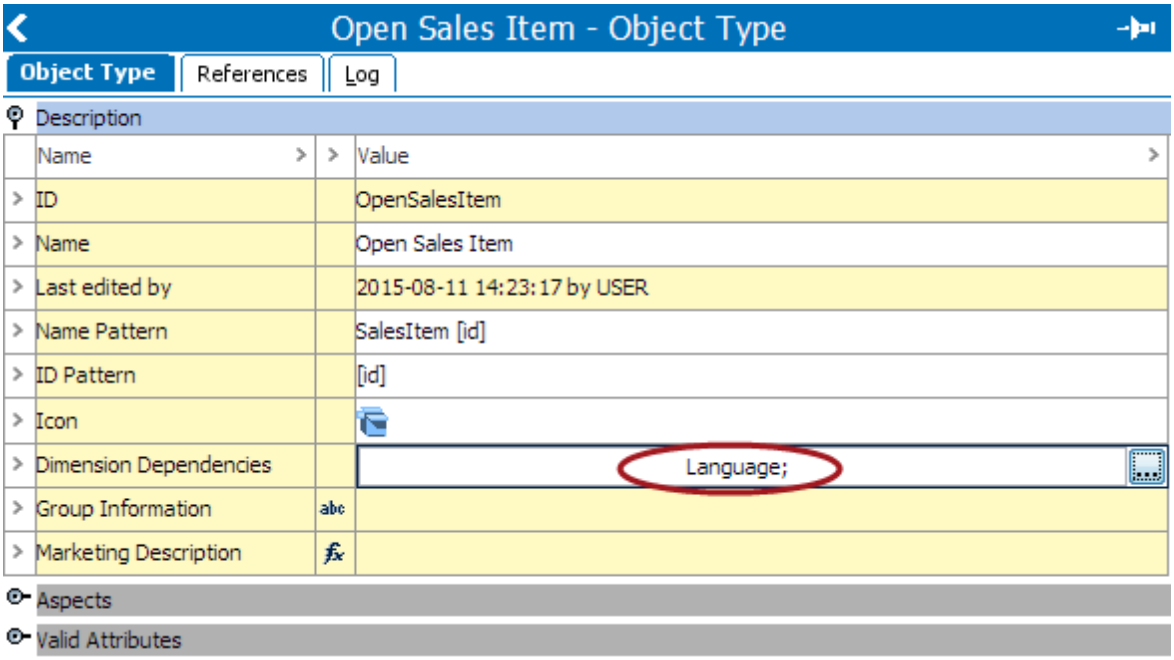
Whenever an Object of a Basic Object Type is created, the relevant Object Type is automatically applied.

New Basic Object Types cannot be created.



Object Type Dimension Dependencies

Setting an Object Type as Dimension Dependent means that the node of that particular Object Type can have a different Name depending upon the Context.



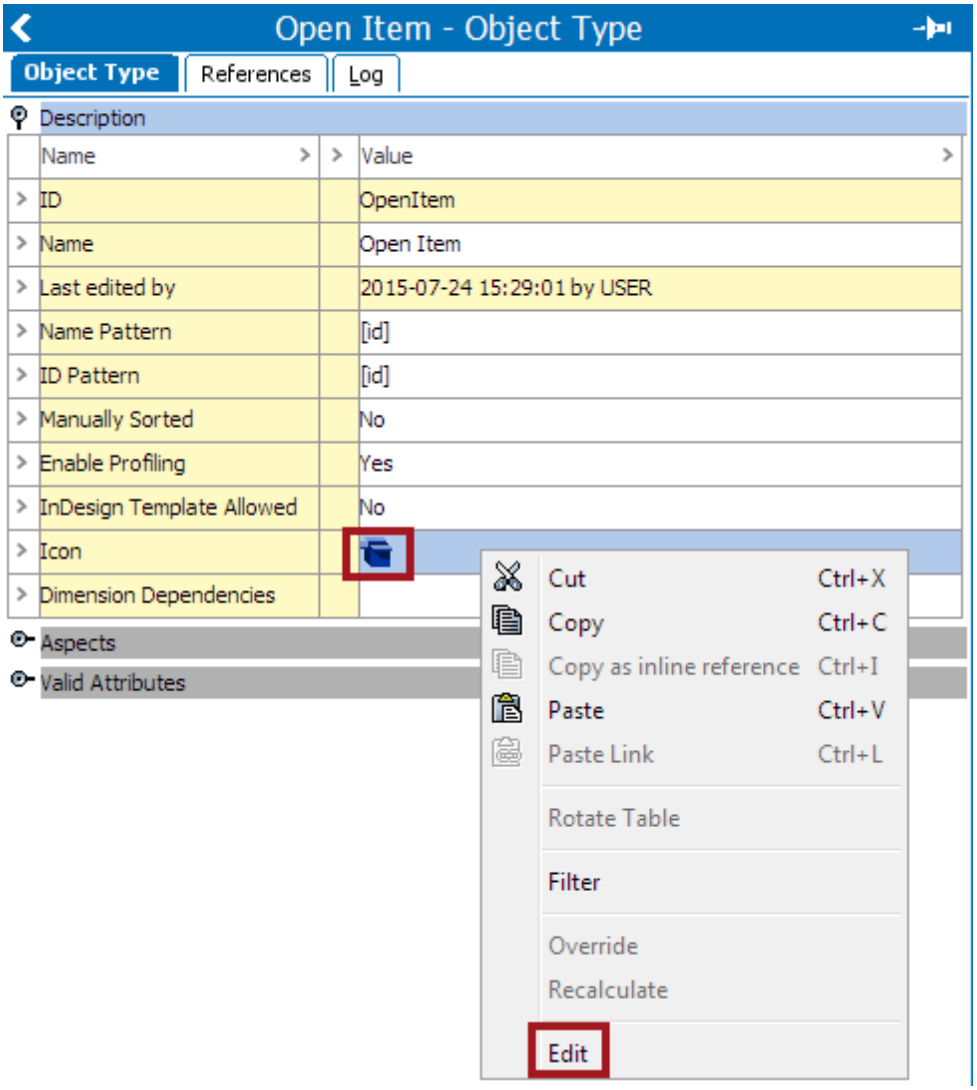
For example, if the Object Type 'Open Sales Item' depends upon the **Language Dimension**, then the Name field of a node of that Object Type may be translated into different languages. This means translation of the Names will support a local language view of the database.

Object Type Icons

An icon is an easy way to identify different objects on the Tree tab. Object Type icons can be set to one of the many Standard Icons or a Custom Icon.

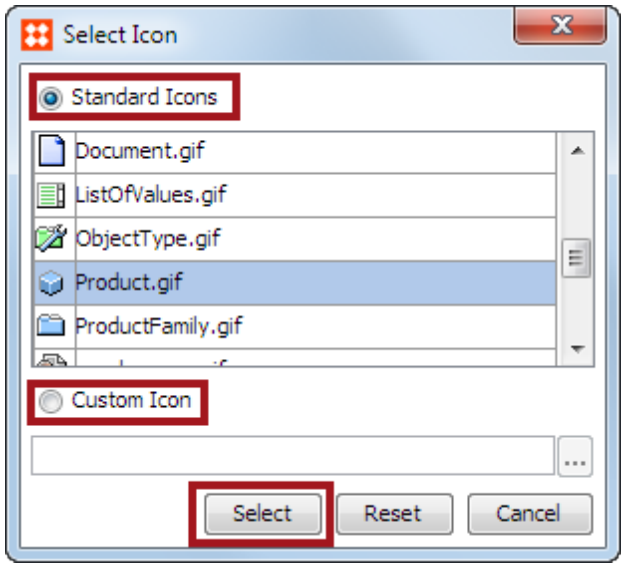
Editing an Object Type Icon

1. On the System Setup tab, select the Icon field of an Object, right-click and select the **Edit** option.

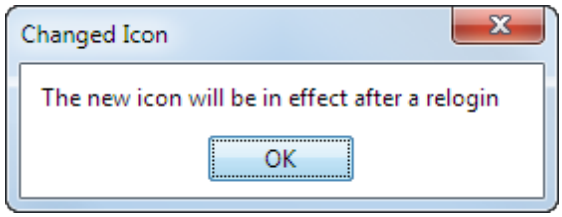


2. Choose a standard or custom icon and click **Select**.

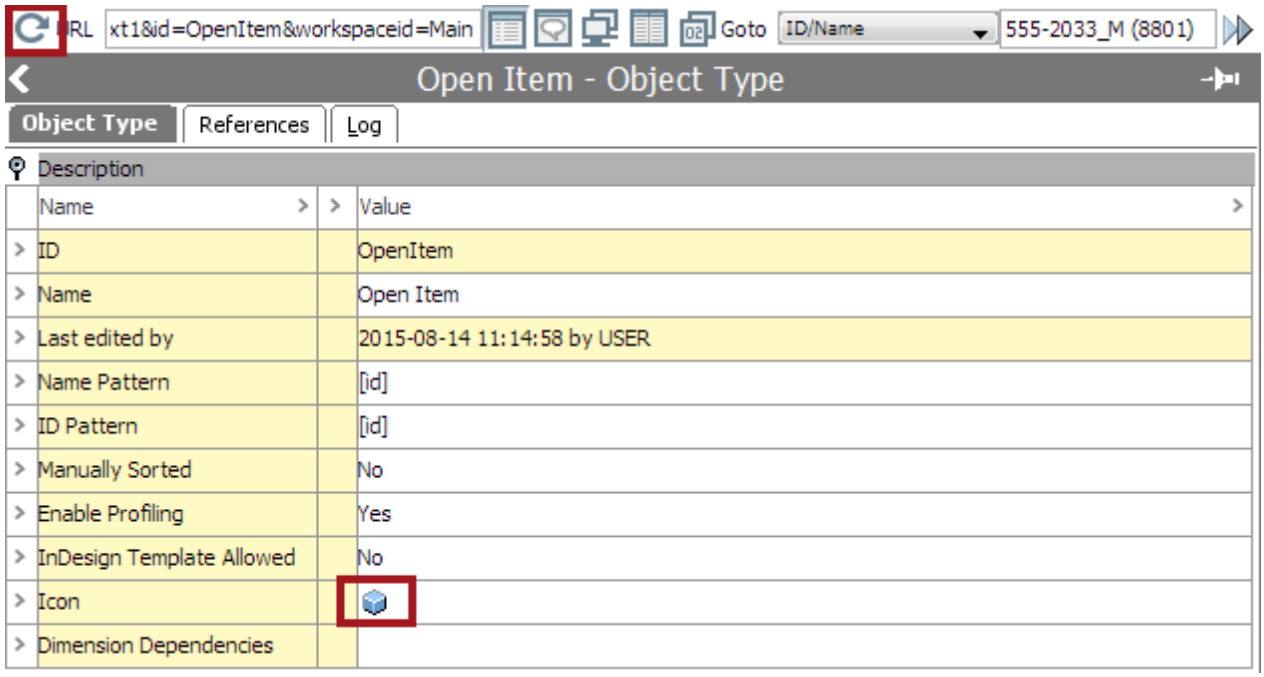
Note: The maximum icon size is 16 x 16 pixels.



3. Click **OK** on the Changed Icon confirmation dialog.



4. On the Object Type, click the **Refresh** button to see the new icon. Logging out is not necessary.



Maintaining Object Types

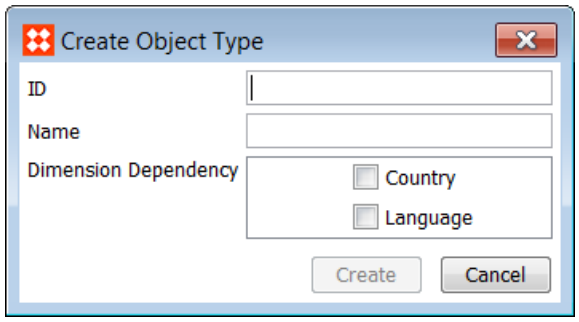
Object Types are maintained in System Setup under **Object Types & Structures**.

An Object Type specifies the characteristics of a group of nodes in the database structure. All nodes are of a specific Object type.

Object Types can be created, deleted, linked and/or unlinked and edited.

Creating an Object Type

1. In System Setup, open **Object Types & Structures** and select the relevant standard Object Type.
2. Right-click and select **New Object Type** to display the **Create Object Type** dialog.



3. Type the **ID** of the new Object Type.

Note: If ID and Name should be identical, press **Enter** to repeat the ID in the **Name** field. Otherwise type a Name.

---

4. If necessary, select one or more dimensions in the **Dimension Dependencies** field. For more information, see the About Dimensions section.
5. Click **Create** to display the **Object Type** Editor with the new Object Type. The new Object type will also be in the list of options when creating a new Object of this type.

Deleting an Object Type

An Object Type that is used or has 'child' Object Types cannot be deleted.

1. In System Setup, open **Object Types & Structures** and select the Object Type to delete.
2. Right-click and select **Delete** from the menu.
3. Click **OK** on the Delete Warning dialog to delete the Object Type. The Object Type is removed from the list of options displayed when creating a new Object of that type.

Unlinking Object Types

1. In System Setup, open **Object Types & Structures** and select the relevant Object Type to display the **Object Type** Editor.
2. Click the **References** tab.
3. In the **Parents** flipper, click the relevant row header arrow and click **Remove Parent**.

Note: At least one parent must exist in the Object Type hierarchy. You cannot remove the last parent.

Updating Object Types from Another Object Type

An Object Type can be updated with the same Attribute Validity, Reference Types Validity, Object Type Parent Links and Object Type Child Links based on another Object Type.

When the new Object Type has been updated in this way, it gets the same Validity settings on Attributes and Reference Types and Object Type Structure as the original Object Type.

1. In System Setup, open **Object Types and Structures** and select the relevant Object Type.
2. Right-Click and select **Update Object Type from** to display the Select Object Type dialog.
3. Select the Object Type to Update from and click **Search**.

   The Object Type Updated status dialog indicates how the selected Object Type was updated:

- number of Valid Attributes added
- number of Parent links added
- number of Child links added
- number of Valid classification-product links added
- number of Reference Types added as Source
- number of Reference Types added as Target

MIME Types

A MIME Type (Multipurpose Internet Mail Extension) is a standard identifier used to indicate the type of data that a file contains. In STEP, they can only be assigned to asset object types.

Specifying a MIME Type allows internet applications to recognize and read the content of an asset object type, and allows imported asset object types to be automatically assigned as an image, application, or a document.

Additional MIME types exist and can be found by searching the internet. Those most commonly used are listed in the table below.

Note: STEP determines that a file is an 'image' if it has a MIME type of **image/***.

MIME Type	Definition
application/*	All
application/pdf	Acrobat PDF
application/illustrator	Adobe Illustrator
application/vnd.ms-excel	Microsoft Excel
application/vnd.ms-powerpoint	Microsoft PowerPoint
application/msword	Microsoft Word
application/postscript	PostScript (including .eps files)
application/x-indesign	InDesign file
application/xml	XML
application/zip	ZIP
image/*	Images of unspecified type
image/bmp	Bitmap image

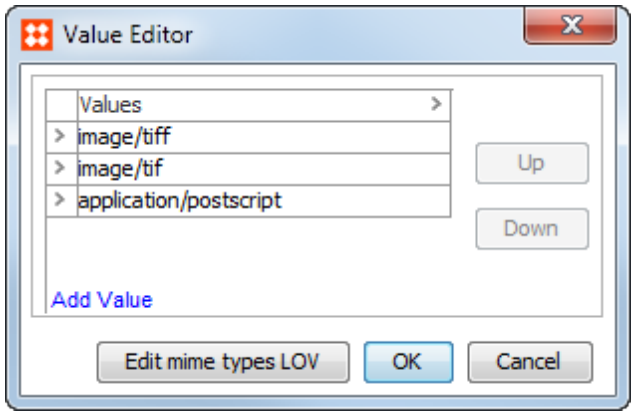
MIME Type	Definition
image/gif	GIF Image
image/jpeg	JPEG JFIF image
image/jpg	JPEG JFIF image
image/png	Portable Networks Graphic
image/tif	TIF Image
image/tiff	TIF Image
text/*	All Text Documents
text/html	Textual data in HTML format
text/rtf	Textual data in rich text format
Text/ plain;charset=ISO-8859-1	Text (Western European character set)
Text/plain;charset=us-ascii	Text (US ASCII character set)
Text/plain; charset=UTF-8	Textual data using the UTF-8 format
video/avi	Covers most Windows-compatible formats including .avi and .divx
video/mp4	MP4 video
video/mpeg	MPEG-1 video with multiplexed audio

Setting MIME Types for Object Types

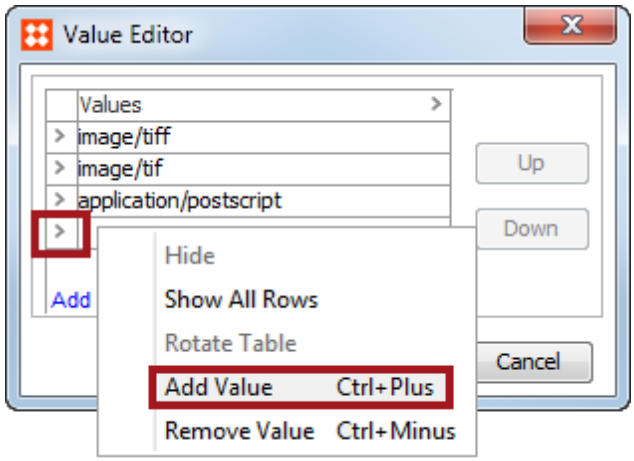
1. In System Setup, select **Object Types & Structures**, then **Asset Type Root**, and select an Image or Document Object Type. Mime Types are not valid for any other object types.

   The corresponding editor appears.

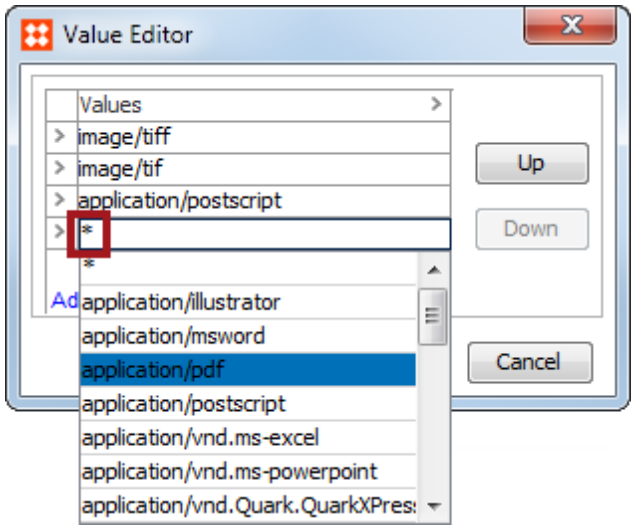
2. Double-click in the **MIME Types** field to display the **Value Editor** dialog.



3. Click the **>** button and select **Add Value** from the list.



4. Click the blank row and type * to display the list of available MIME Types, select a value and press **Enter**.



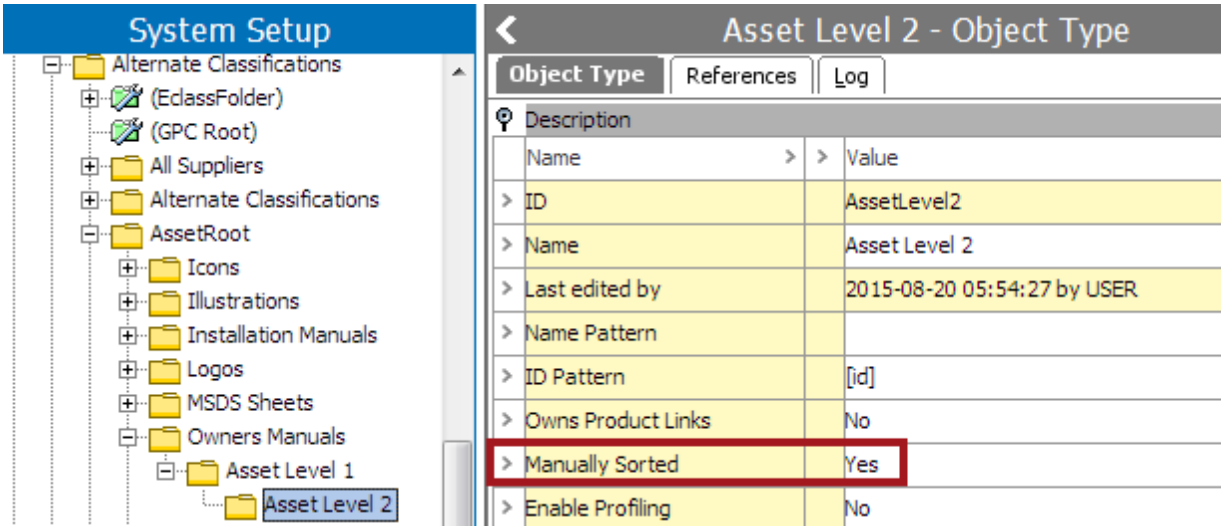
5. Repeat steps 3 and 4 to select more MIME Types.
6. Click **OK** to display the Object Type Editor assigned MIME Type(s).

   When importing an image or a document, it is automatically assigned the Object Type with the specific MIME Type.

Modifying Manually Sorted

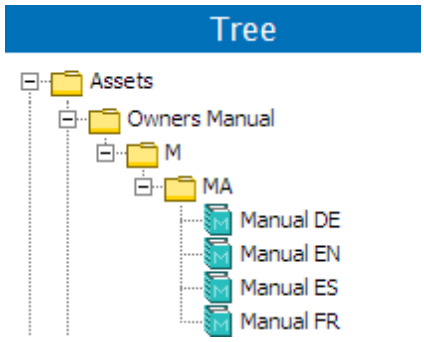
In the Tree, objects in the **Alternate Classifications** hierarchy, **Primary Product Classification** hierarchy, and **Publication** hierarchy sort alphabetically by default. This setting can be modified.

1. In System Setup, view the Object Type tab on the editor of an **Alternate Classifications**, **Primary Product Classification**, or **Publication** object type.

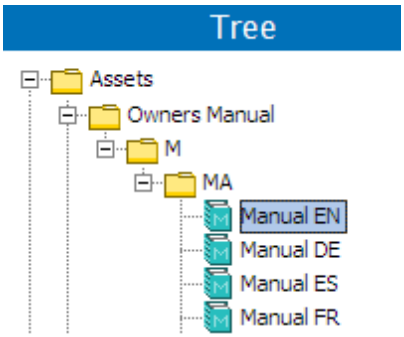


2. Set the **Manually Sorted** field:

   - **No** means children of the object are ordered alphabetically in the Tree. Their sequence cannot be manually changed by a user.



   - **Yes** means a user can manually change the sequence of child objects in the tree hierarchy by dragging an object to another position.


Tree
Assets
Owners Manual
M
MA
Manual EN
Manual DE
Manual ES
Manual FR

Object Types for Classification, Product, Assets and Entities

An Object Type specifies the characteristics of a group of nodes in the database structure. All nodes are of a specific Object Type. Whenever a Classification, Product or Images & Documents Object is created, you must manually apply the relevant Object Type.

Note: Description Attributes applied on an Object Type will appear on all Objects of this Object Type.

Object Type	Definition	Example
Classification	Defines the structure of the Classification hierarchy	• Classification 1 user type root 'Application' • Classification level 1 'Furniture' • Classification level 2 'Office'
Product	Defines the structure of the Product hierarchy.	• Product Group 'Chairs' • Product Family 'Office chairs' • Product 'Acme chair 907098'
Asset	Defines the available types of Assets.	• Image • Document • Default document • Video • Audio • Import Manager configuration • Export Manager configuration
Entities	Defines the structure of the Entity hierarchy.	• Customers

Object Type	Definition	Example
		• Addresses
Publications	Defines the structure of the Publication hierarchy. Includes Publication Object Types, Publication Group Types, and Publication Sections.	• Printed Catalogs • Printed Sale Fliers

Object Type Hierarchy Overview

The Object Type hierarchy defines the relationship (parent - child) between the Object Types. The Object Type hierarchy will also specify the possible depth of a Product or a Classification hierarchy, for example, the number of possible 'levels' before you reach an Object Type with no 'child' Object Type.

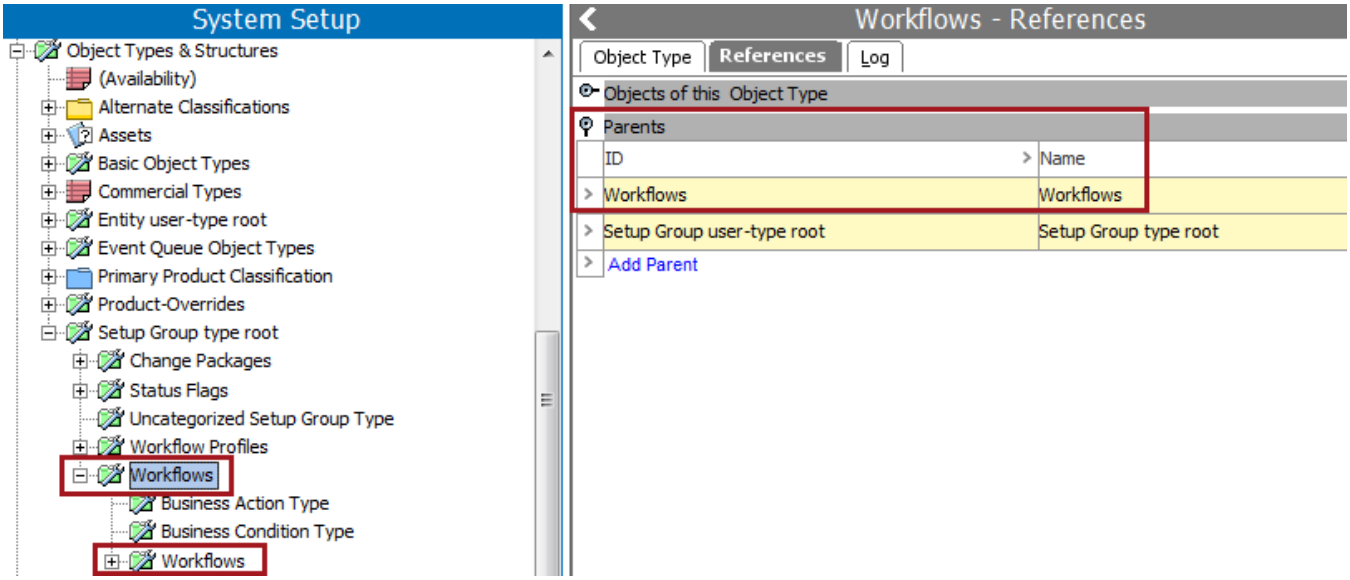
Child nodes of such Object Types are also referred to as 'leaf' nodes in the hierarchies.

The unique structure of each Product or Classification hierarchy is defined by an Object Type hierarchy.

Note: It is important to understand that the construction of the Object Type hierarchies is primarily a one-time setup operation. Once the Object Type hierarchies are established, any changes must be thoughtfully considered as such changes may have serious effects on your global database structure.

The 'labeling' of each node in a Product or Classification hierarchy with a specific Object Type is also used to group a set of nodes that share similar characteristics.

The Object Type hierarchies can be recursive, meaning that a 'child' can be its own 'parent'. This allows further division of the same types of objects within a single folder. In this case, there is no limit to the depth of a Product or Classification hierarchy that includes nodes from such a recursive Object Type hierarchy. To make an Object Type recursive, on the References tab, add the Object Type as its own parent.

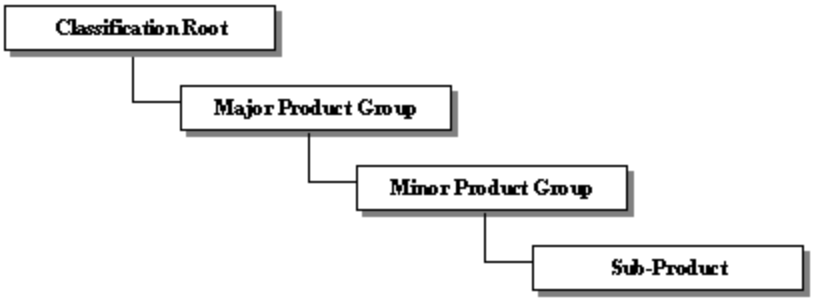


Classification and Product Hierarchy Structures

The two hierarchy structures that may be defined are:

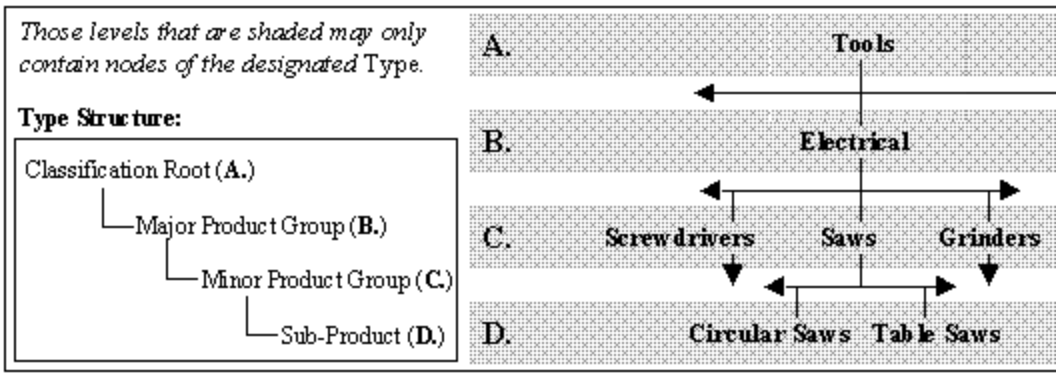
- Classification hierarchy
- Product hierarchy

The following examples of Object Type structures define the hierarchy structures of two different classifications 1) where the Object Type designated the level and 2) where nodes of different Object Types reside in the same level.

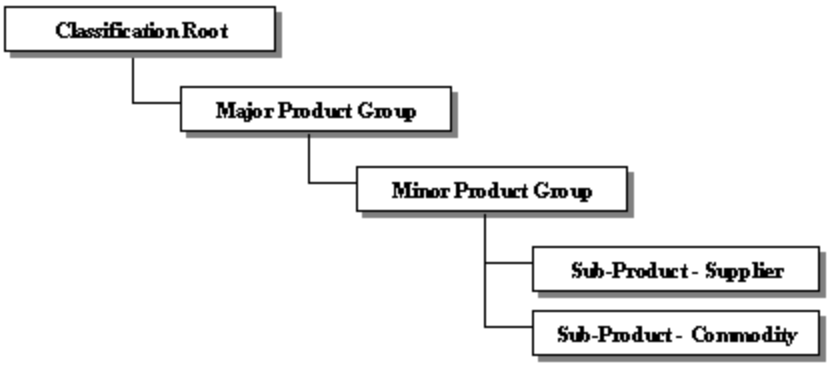


In this example, only a **Major Product Group** may be a child of a **Classification Root**, a **Minor Product Group** may only be a child of a **Major Product Group**, etc. As defined by this structure, it is not possible to create a **Sub-Product** under the **Classification Root**. Any nodes of the Object Type **Sub-Product** automatically become leaf nodes, because the Object Type structure does not allow any nodes of any Object Type to be created under **Sub-Product**.

Now that the Object Type structure is defined, the following hierarchy is possible:



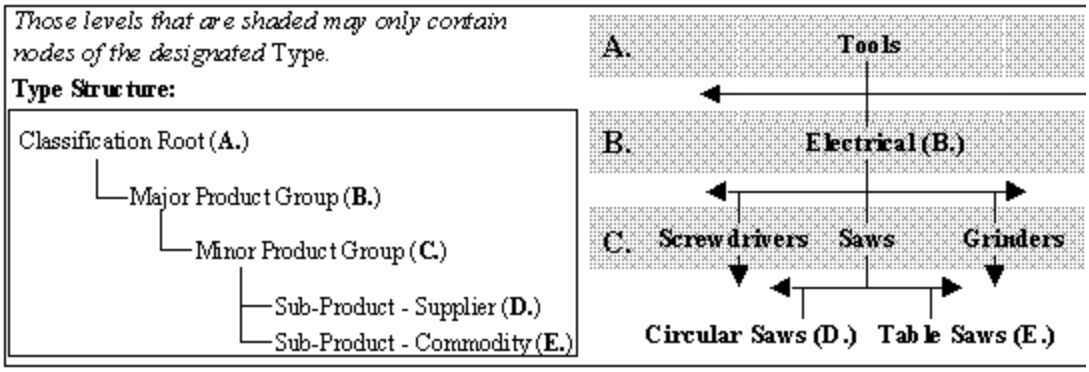
The first example illustrates a simple structure where each Object Type defines a level in the hierarchy. To build on this illustration, the second example shows it is also possible to create an Object Type structure in which a parent Object Type may have multiple children.



In this example, the Object Type structure allows a child node of the Object Type **Minor Product Group** to be created under a **Major Product Group** node. However, two (2) different Object Types may be created under a **Minor Product Group** node: **Sub Product - Supplier** and **Sub-Product - Commodity**.

This shows how nodes within the same level may be of multiple Object Types. Therefore, Object Types do not necessarily represent 'levels' within a hierarchy. Also, both of these Object Types are leaf nodes, as neither allows any children to be created beneath them.

Using this structure, the following a hierarchy is possible:



The example above illustrates that there can be nodes of different Object Types within the same level in a hierarchy:

- Circular Saws is of the Object Type **Sub-Product - Supplier**
- Table Saws is of the Object Type **Sub-Product - Commodity**

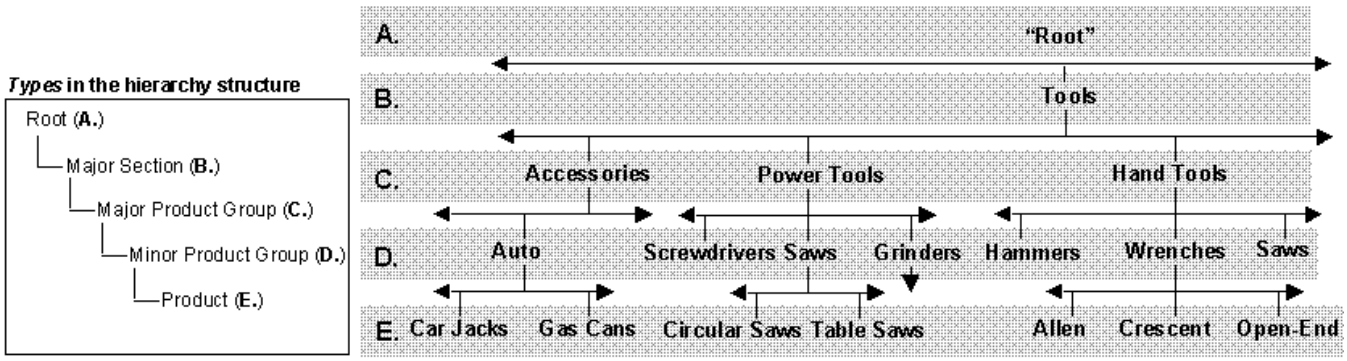
Fixed Hierarchy Structure

Each hierarchy created in the database is defined with a specific structure in mind. Each hierarchy node within each level is assigned an Object Type. The manner in which the hierarchy is structured determines what Object Types that can be assigned to the nodes in each level.

For example, the diagram below shows a fixed (rigid) hierarchy, where all Products have been categorized into four levels (not counting the top or root level). This structure means that all Products, regardless of what type of Product, must fall into this four-tier categorization.

In this example, all the nodes in each level of the hierarchy will have the same Object Type. The overall effect of this type of setup is that each level has been assigned a different Object Type. So, any hierarchy node residing in the specified gray area is of the Object Type that has been assigned to that level.

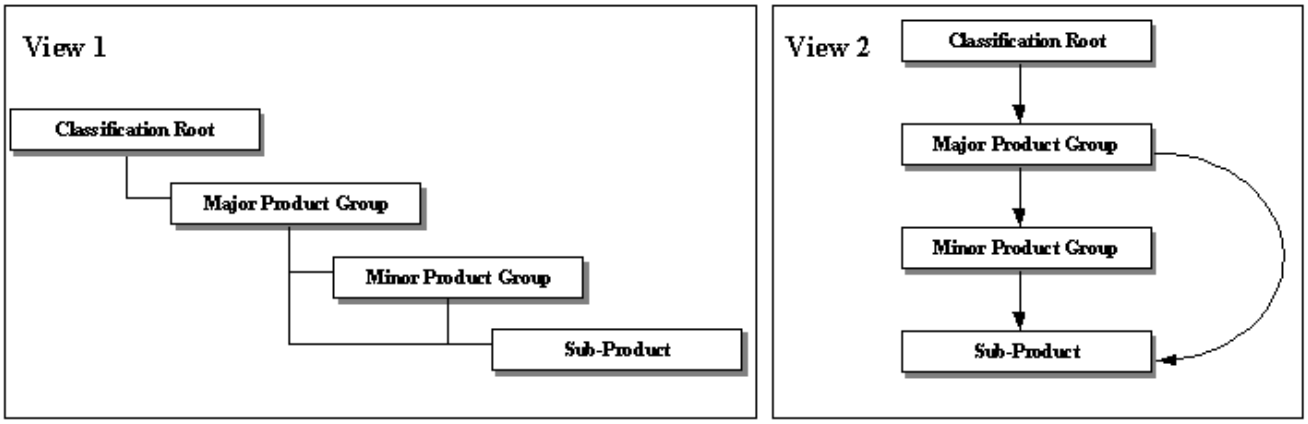
In this example, the 'Screwdrivers' node is of the Object Type called 'Minor Product Group' and the 'Table Saws' node has been given 'Product' as its Object Type. In the hierarchy structure, then, the setup is such that there is a fixed number of levels, and that at each level there is only one 'child' Object Type. For example, the only child of 'Major Product Group' is the one called 'Minor Product Group'.



For information, see the Flexible Hierarchy Structure section.

Variation in Hierarchy Depths

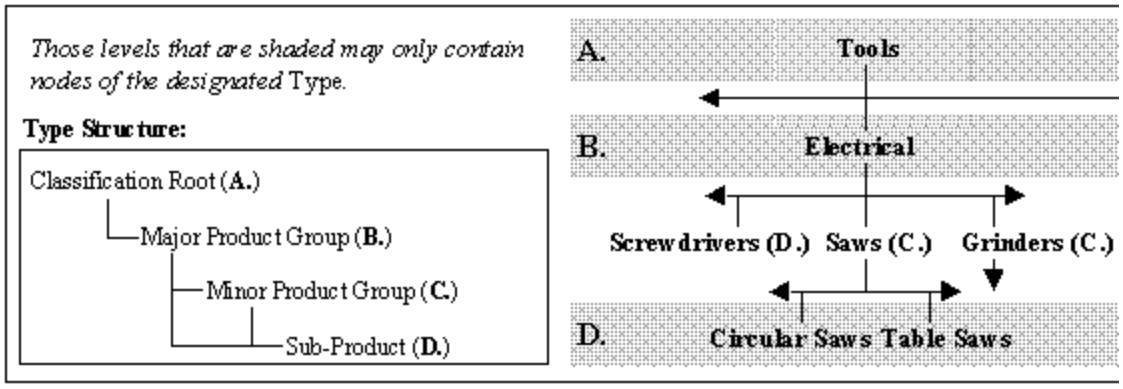
In some instances, the depth of the hierarchy may vary. For example, the leaf nodes are of the same Object Type, but may be created in different levels. This is accomplished by allowing an Object Type to have multiple parents and displayed in the figure below.



The two views represent the same Object Type structure. In this example, when creating a node below a node of the Object Type **Major Product Group**, it is possible to create a node of the Object Type **Minor Product Group** or of the Object Type **Sub-Product**.

If a **Minor Product Group** is created, it is then possible to create a child node of the Object Type **Sub-Product** also. No matter where in the hierarchy the node of the Object Type **Sub-Product** is created (below **Major Product Group** or **Minor Product Group**), it is a leaf node because it is not possible to create any nodes below a **Sub-Product**.

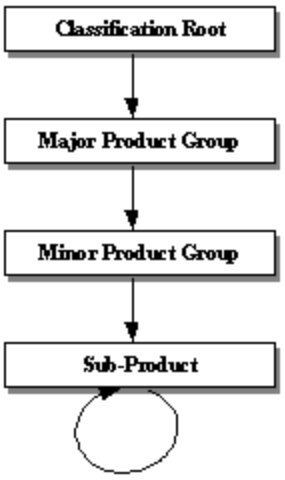
The following figure represents a hierarchy based on a varying Object Type structure.



Multiple Parents

A node of the Object Type **Major Product Group** can have children of either the Object Type **Minor Product Group** or **Sub-Product**. For example, the node Screwdrivers is of the Object Type **Sub-Product** and Saws is a **Minor Product Group**. A node of the Object Type **Minor Product Group** can only have children of the Object Type **Sub-Product**.

The Object Type hierarchy can be structured so it allows an Object Type to become a child of itself. This causes a situation in which a leaf node can never be created in the corresponding Product or Classification hierarchy, because when a node of the recursive Object Type is created, a child node of the same Object Type can always be created as displayed in the figure below.

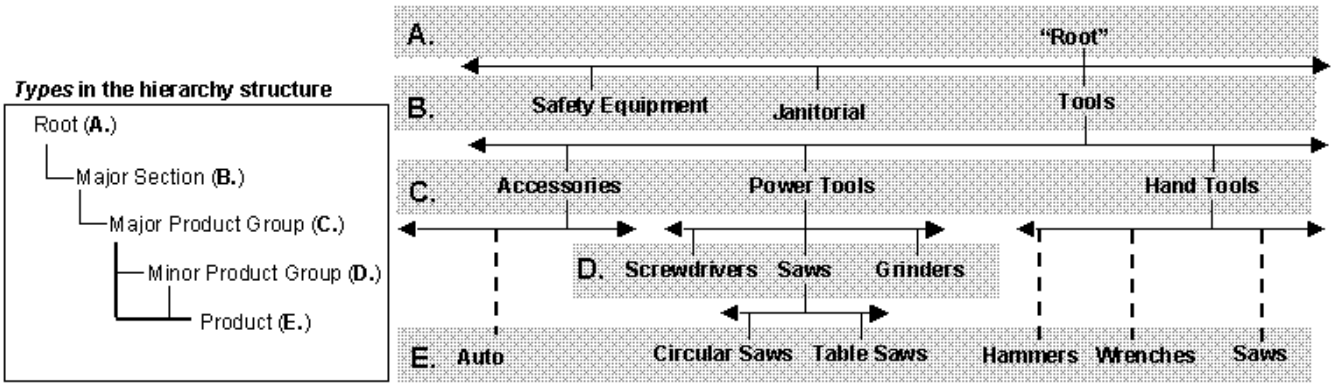


Flexible Hierarchy Structure

Not all hierarchies are suited to a formal type of structure discussed in the Fixed Hierarchy Structure section. Some groups of Products do not require four levels. There may be so few Products in certain groupings that three levels of hierarchy are preferred. Therefore, the structure of the hierarchy needs to be flexible enough to accommodate this.

In the figure below, the Fixed Hierarchy Structure has been modified so that the Object Type called 'Product' can be either a child of the Object Type called 'Minor Product Group', or it can be a child directly of Object Type called 'Major Product Group'. In other words, the Object Type called 'Major Product Group' now has two children instead of only one: the Object Types are 'Minor Product Group' and 'Product'.

In this example, there is only one level below 'Accessories' and 'Hand Tools', but there are two levels of hierarchy below 'Power Tools'.

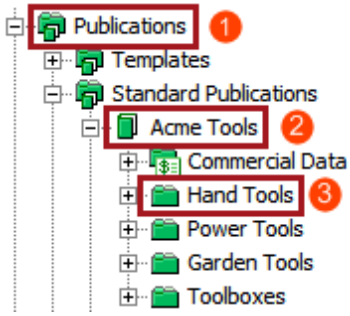


Creating a Standard Publication Object Type Hierarchy

Publication Object Types are created and maintained in System Setup under **Object Types & Structures**. The topics in this section describe how to create the object types needed to build a standard **Publication Hierarchy** in the Tree.

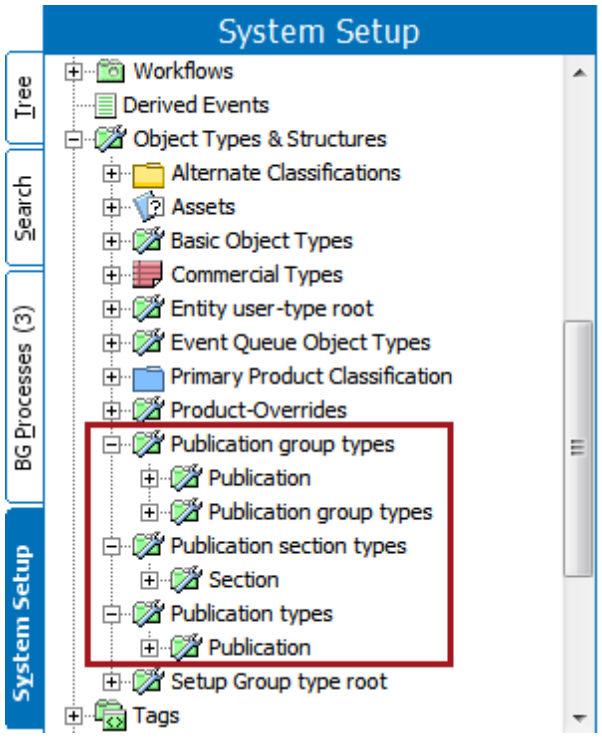
The following screenshot shows what a basic publication hierarchy looks like in the Tree. The publication object types addressed in this documentation are used to build the following objects:

1. Publication Group
2. Publication
3. Section



Typically, STEP systems are delivered preloaded with the following standard object types used in creating a Publication Hierarchy:

- **Publication group types** (ID = Publication group root): This object type serves a dual function as both the root node of the Publication Hierarchy as well as the standard object type used to create **Publication Group** objects in the Tree.
- **Publication types** (ID = Publication root): Root node for **publication** object types in System Setup; not used to create publication objects themselves in the Tree
- **Publication** (ID = Default publication type): Used to create **publication** objects in the Tree
- **Publication section types** (ID = Section root): Root node for **section** object types in System Setup; not used to create section objects themselves in the Tree
- **Section** (ID = Section): Used to create **section** objects in the Tree



Though these basic object types can be used to create an entire publication hierarchy 'out of the box,' you may need to create other publication hierarchy object types for scenarios such as:

- The need to differentiate different collateral types by object type (e.g., catalogs, flyers, brochures)
- The need to assign different User Groups to specific areas of the publication hierarchy, which can be accomplished by configuring User Group privileges in conjunction with the different publication hierarchy object types
- The need to place different types of metadata (by use of Description attributes) on different types of publications, publication groups, and sections

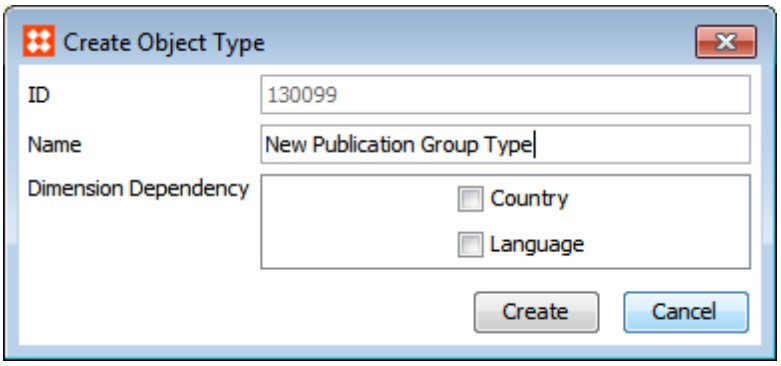
The following sections of this documentation will guide you through the process of creating new publication hierarchy object types as well as configuring valid parent / child relationships between these object types.

If your system is missing any of the three publication hierarchy *root node* object types (Publication group root, Publication root, or Section root), contact Stibo Systems for assistance, as publication hierarchy object types cannot be created unless their corresponding root node object types preexist in System Setup.

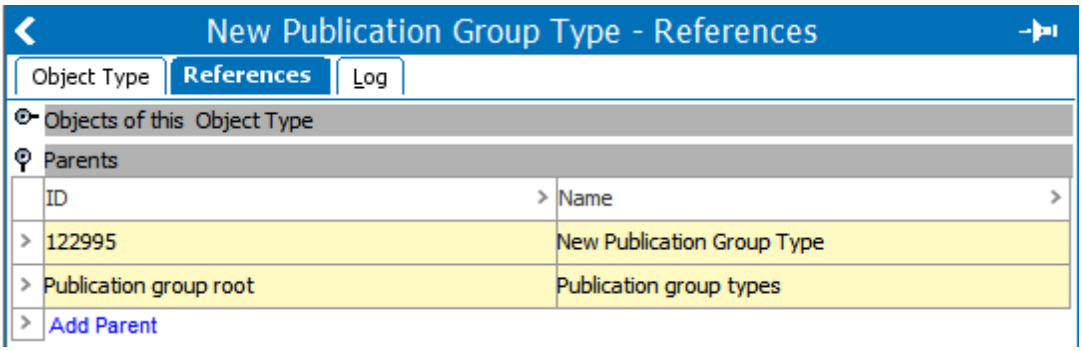
For more information on the creation and configuration of publication hierarchy objects in the Tree, see the **Publication Maintenance** section of the **Getting Started** / **STEP User Guide** documentation.

Creating a Publication Group Object Type

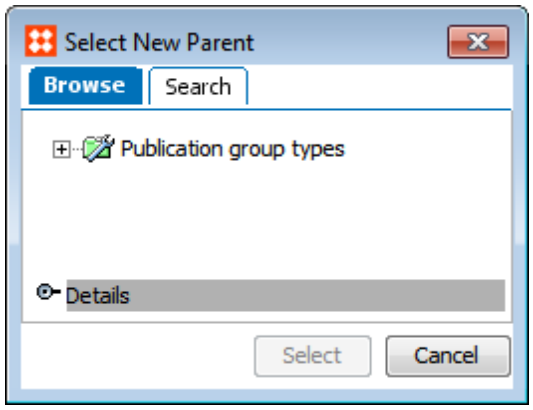
1. In System Setup, open the **Object Types & Structures** hierarchy.
2. Select **Publication group types**, right-click, and then select **New Object Type**. A **Create Object Type** dialog box appears.



3. If the STEP ID for the **Publication group root** object has not been set to autogenerate with an ID Pattern, type an ID in the **ID** field.
4. In the **Name** field, type a name.
5. Check the relevant **Dimension Dependency** check boxes, if applicable. (Dimension dependencies are optional, and are not typically used with publication group object types.)
6. Click **Create**. An Editor displays for the newly created publication group object type.
7. Click the **References** tab.



8. Click **Add Parent** to define which parent publication group object(s) should be valid for the new publication group object. The **Select New Parent** dialog box appears.

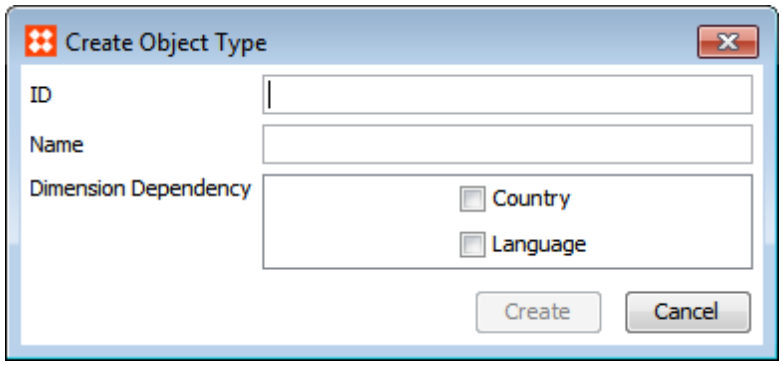


9. Search or browse for the relevant publication group type, then click **Select**.

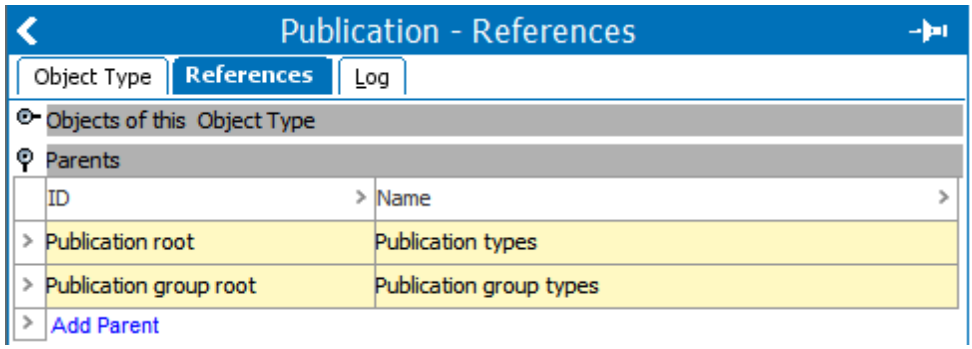
Note: Publication group objects may only be children of other publication group objects. They may not be children of publication objects or section objects.

Creating a Publication Object Type

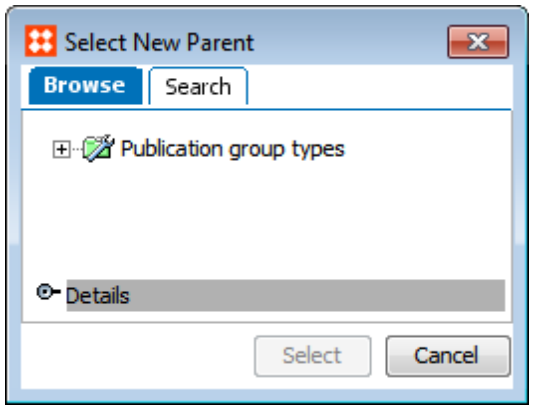
1. In System Setup, open the **Object Types & Structure** hierarchy.
2. Click **Publication types**, right-click, and then click **New Object Type**. A **Create Object Type** dialog box appears.



3. If the STEP ID for the 'Publication root' object has not been set to autogenerate with an ID Pattern, type an ID in the **ID** field.
4. In the **Name** field, type a name.
5. Check the relevant **Dimension Dependency** check boxes, if applicable. (Dimension dependencies are optional, and are not typically used with publication object types.)
6. Click **Create**. An editor displays for the newly created publication object type.
7. Click the **References** tab.



8. Click **Add Parent** to define which parent publication group object(s) should be valid for the publication object. The **Select New Parent** dialog box appears.

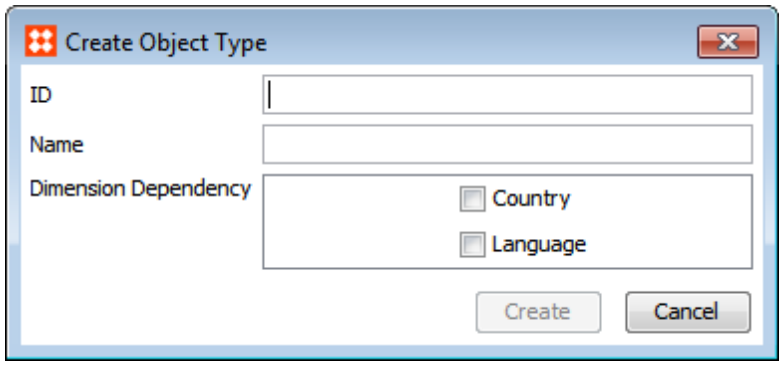


9. Search or browse for the relevant publication group type, then click **Select**.

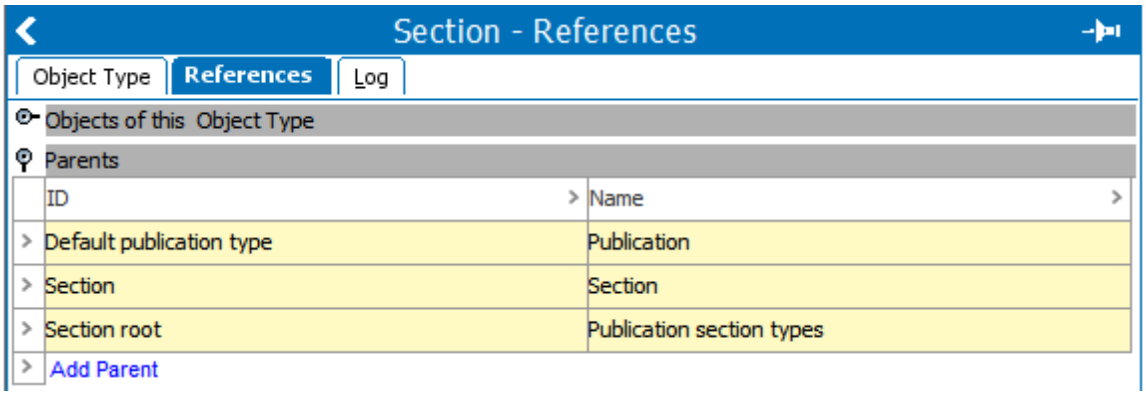
Note: Publication objects may only be children of publication group objects. They may not be children of other publication objects or section objects.

Creating a Publication Section Object Type

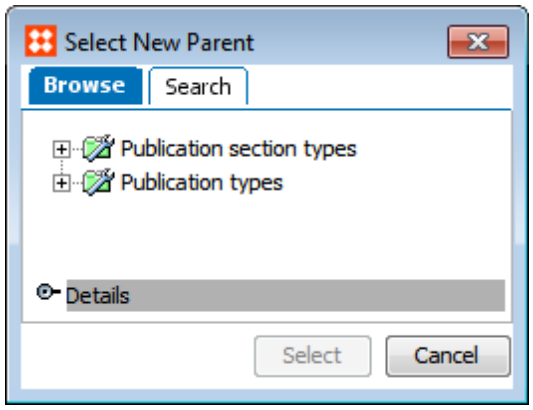
1. In System Setup, open **Object Types & Structures**.
2. Click **Publication section types**, right-click, and then select **New Object Type.** The **Create Object Type** dialog appears.



3. If the STEP ID for the 'Section root' object has not been set to autogenerate with an ID Pattern, type an ID in the **ID** field.
4. In the **Name** field, type a name.
5. Check the relevant **Dimension Dependency** check boxes, if applicable. (Dimension dependencies are optional, and are not typically used with publication section object types.)
6. Click **Create**. An Editor displays for the newly created Section object type.
7. Click the **References** tab.



8. Click **Add Parent** to define which parent publication and/or section object(s) should be valid for the section object. A **Select New Parent** dialog box appears.



9. Search or browse for the relevant publication and/or section type, and then click **Select**.

Note: Publication section objects may be children of publication objects and other section objects but they cannot be children of publication group objects.

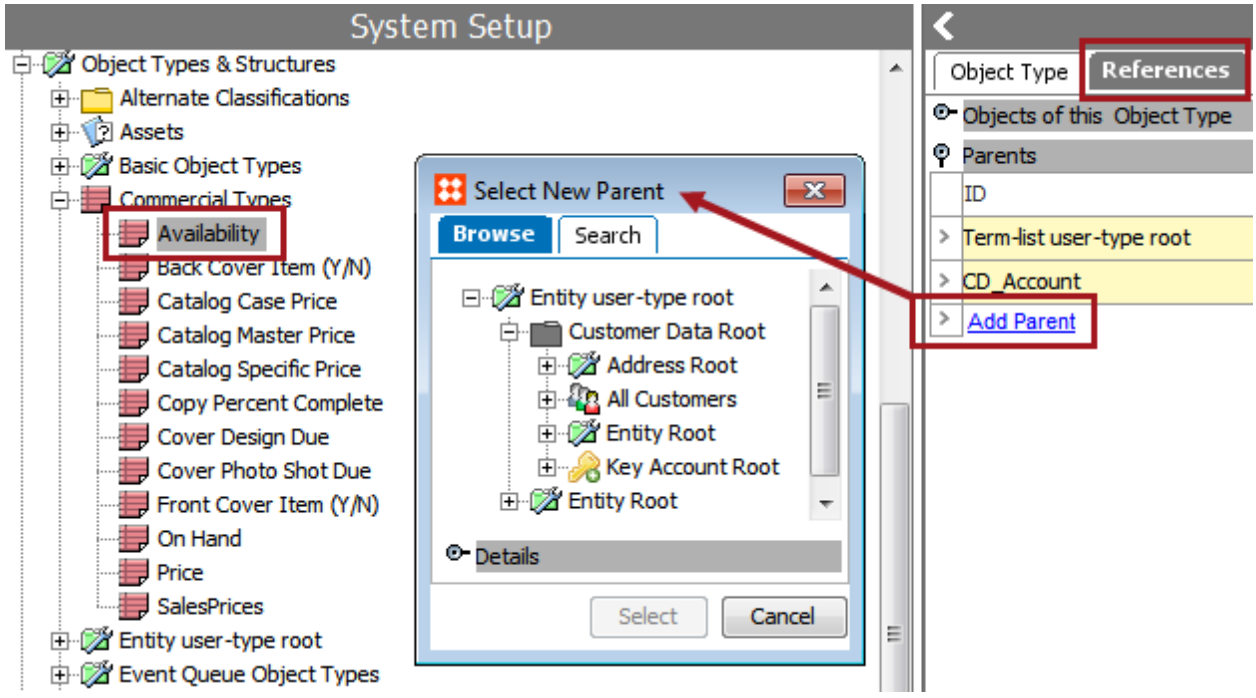
Product-Override Object Types and Commercial Object Types

Product-Override object types and Commercial object types appear in their own hierarchy under Object Types & Structures.

Before using Product-Overrides in the Product hierarchy or commercial data in the Entity hierarchy, the levels in which those objects can be legally created / maintained must be specified.

- Product-Override object types must be linked into the Product Object Type hierarchy.
- Commercial data types must be linked into Entity object types if they are needed within an Entity hierarchy. (Commercial Types can be used directly in publications and eCatalogs without additional linking.)

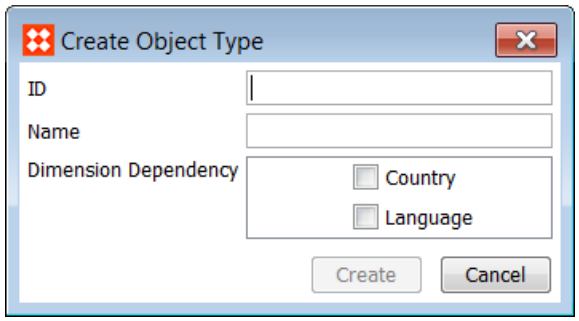
Link the Object Types from the References tab on either a Commercial Type or the References tab on a Product-Override by clicking the **Add Parent** link. A node selector displays a list of legal Object Types for linking.



Note: When Commercial Types must be used in Entity Hierarchies, manually link the Commercial Type into the Entity Hierarchy.

Creating a Commercial List Type

1. In System Setup, open **Object Types & Structures**.
2. Click **Term-list user-type root**, right-click, and select **New Object Type** to display the **Create Object Type** dialog.

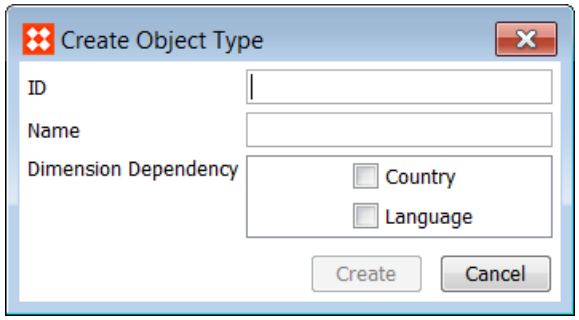


3. Type an **ID** and **Name**. Dimension Dependencies are optional, and are not typically used with Commercial List Types.
4. Click **Create** to display the created Commercial List Type dialog.
5. In the **Object Type** tab, double-click in the **Value** column of the **visible term** fields to display the **Value Editor** dialog.
6. Click **Add Value**, and then click the row under the **Value** column.
7. Click **Dates**, **Quantities** or **Unit** from the list.
8. To add more values, repeat steps 5 and 6.
9. Click **OK** when finished.

Creating a Product-Override Object Type

One standard Product-Override Object Type is supplied with the system. Additional Product-Override Object Types can be created with specific settings for each, such as manual attribute sequencing and rules for template changes.

1. In System Setup, open **Object Types & Structures**
2. Select **Product-Overrides**, right-click, and click **New Object Type** to display the **Create Object Type** dialog.



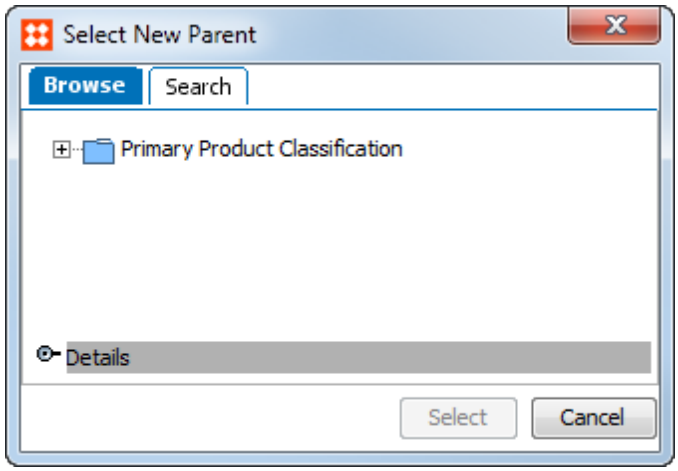
3. Type an **ID** and **Name** and optionally, check Dimension Dependencies.
4. Click **Create** and a new Product-Override Object Type is created.

Note: To enable a Product-Override, additional linking is required. For more information, see the Enabling a Product Override section.

Enabling a Product-Override

A Product-Override Object Type must be linked to the folders in the Primary Product Hierarchy on which a Product-Override should be possible.

1. In System Setup, open **Object Types & Structures**.
2. Open **Product-Overrides**, and select the relevant Object Type to display an editor.
3. On the **References** tab click the **Add Parent** link to display the **Select New Parent** dialog.

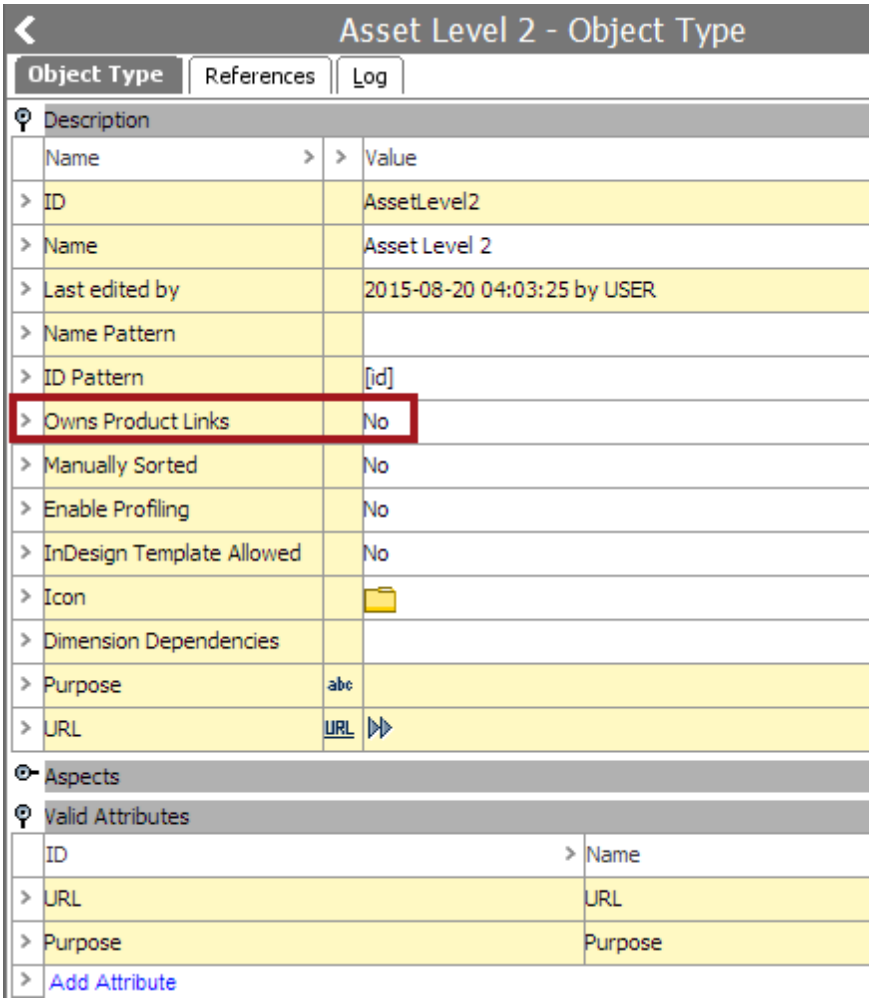


4. Use Search or Browse to select an Object Type under **Product user-type root**.
5. Click **Select** to apply a Product-Override to a Product Object Type.

Note: It may be necessary to define specific settings for each Product-Override Type such as manual attribute sequencing and rules for template changes.

Owns Product Links on Alternate Classifications Object Type

Owns Product Links specifies the ownership of a link from a Product to a Classification. The Owns Product Links field determines if the link should be unapproved when Products are linked to the Classification.



The available options for Owns Product Links are:

- **No** means the Product 'owns' the link to the Classification. When an additional product is linked to the Classification the approval status of the Classification is not changed. It is not necessary to Approve the Classification in order to see linked Products in the Classification in the Approved Workspace.
- **Yes** means the Classification reference 'owns' the link to the Product. When an additional product is linked, the Classification will become unapproved. It is necessary to Approve the Classification in order to see linked products in the Classification in the Approved Workspace.

Owns Product Links Affects STEPXML

When Owns Product Links is Yes, STEPXML exports these references as both 'ProductReferences' under the Classification and as 'ClassificationReferences' under the Product. Note the red text below:

```
<STEP-ProductInformation>
<Classifications>
<Classification ID="Game Consoles" UserTypeID="Classification Folder" ParentID="Electronics" Referenced="true">
<ProductReference ProductID="XBox 360" Type="Classification Owned Product to Classification Link"/>
</Classification>
</Classifications>
<Products>
<Product ID="Xbox 360" UserTypeID="Sellable unit" ParentID="Microsoft Products">
<ClassificationReference ClassificationID="Game Consoles" Type="Classification Owned Product to Classification Link"/>
</Product>
</Products>
</STEP-ProductInformation>
```

To stop the classification-owned links from being exported under the products, set the recorder option 'IncludeClassificationOwned' on the Products element to false as shown in the red text below:

```
<STEP-ProductInformation>
<Products IncludeClassificationOwned="false">
<Product>
</Product>
</Products>
</STEP-ProductInformation>
```

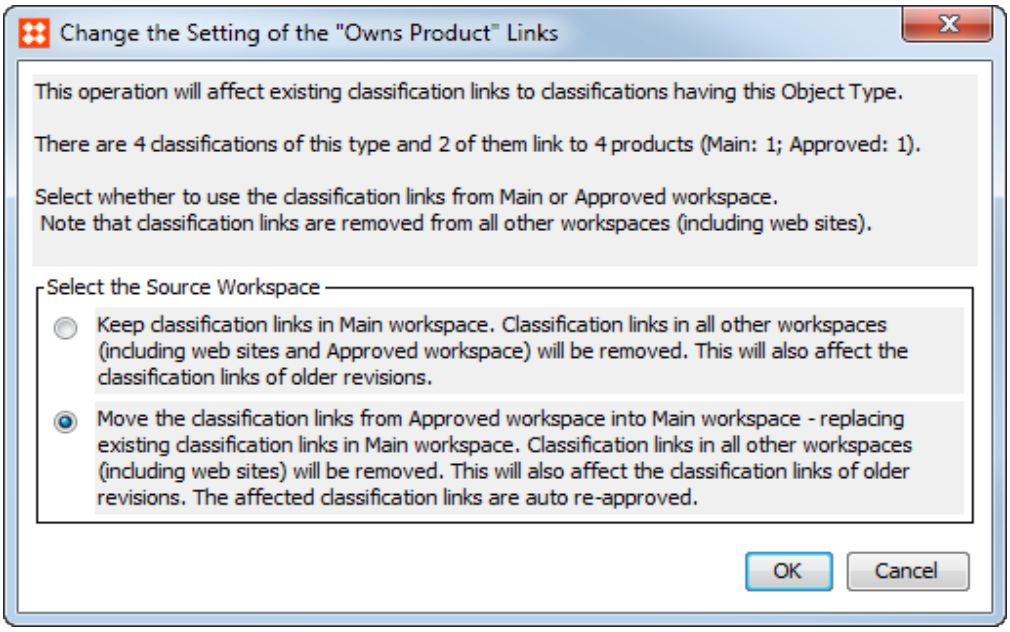
Changing Owns Product Links

Owns Product Links specifies the link ownership when a Product is linked to a Classification. For more information about Owns Product Links, see the Owns Product Links on Alternate Classifications Object Type section.

Note: Changing the setting of Owns Product Links can be a time-consuming operation.

---

1. In System Setup, open **Object Types & Structures**, select the **Alternate Classification Object Type** that needs to have the **Owns Product Links** setting changed.
2. Click the **Object Type** tab.
3. In the **Own Product Links** field, select **Yes** or **No** to display the **Change Owns Product Links Setting** dialog.



4. Select an option to handle the existing links across the Workspace:

   Note: Both options affect the Classification References in older revisions.

   - **Keep classification links in Main Workspace:** the Classification References in the Main Workspace remain untouched. Classification References in all other Workspaces are removed. Since the references in the Approved Workspace are replaced with the references from Main Workspace, the Approved Workspace might not have the same references after the change.
   - **Move the classification links from Approved Workspace into Main Workspace:** *This is the recommended option.* The Classification References in the Approved Workspace remain untouched. Classification References in Main Workspace are replaced with the Classification References from the Approved Workspace. The affected Classification References are auto re-approved. Since the references in the Main Workspace are replaced with the references from Approved Workspace, the Main Workspace might not have the same references after the change.

5. Click **OK** to start the process and display a **Status** dialog.

Revisability on Entity Object Type

The Entity Object Type editor can include a Revisability field. An Entity Object can be Globally Revisable or Workspace Revisable.

- Globally Revisable entities display in all Workspaces and are not under revision control.
- Workspace Revisable entities display in the Workspace where created and are under revision control.

When building an Entity Object Structure, a top Entity Object Type can be Globally Revisable while child Entity Object Types are Workspace Revisable. However, Workspace Revisable top Entity Object Type cannot have child Entity Object Types that are Globally Revisable.

Product Variant Families

A product variant family is a group of products where the members are considered to be the same product except for variations in the predefined attributes. Variations are, for example, different sizes or colors.

Say you have a product folder named "Shoes" that contains two products: "Shoe1" and "Shoe2". It has two variant attributes: Color and Size.

To specify which attributes are considered variant attributes, you use a special meta attribute. The meta attribute must be valid on the link from the parent folder to the variant attributes. This way, each linked attribute can be attached with a number indicating that the attribute is a variant attribute and specifying where in the related views the attribute is shown.

In this example, the variant priority attribute on the link from Shoes to Color is 1 and from Shoes to Size it is 2. This indicates that Color and Size are variant attributes and that Color is shown before Size.

In addition to numbers, the variant priority attribute on a linked attribute can also be given a number prefixed with an "n" (e.g. "n1" or "n12"). This notation is used to specify that the linked attribute is visible in the views related to variant attributes. However, in relation to, for example, validation, deduplication and variant creation it is not considered a variant attribute. Attributes where the variant priority attribute has a value starting with "n" are called "non-variant attributes".

In the "Shoes" example, there is an attribute "Price" linked from the Shoes folder with variant priority attribute value "n3". This indicates that the price of the shoes is shown in the views, but it is not used to distinguish variants.

Product Variant Family Attributes

Attributes linked from a product variant family is categorized into three groups:

Variant attributes: The variant priority attribute is a number: 1,2,3,...

Non-variant attributes: The variant priority attribute is a number prefixed with "n": n1,n2,n3,...

Regular attributes: The variant priority attribute is empty.

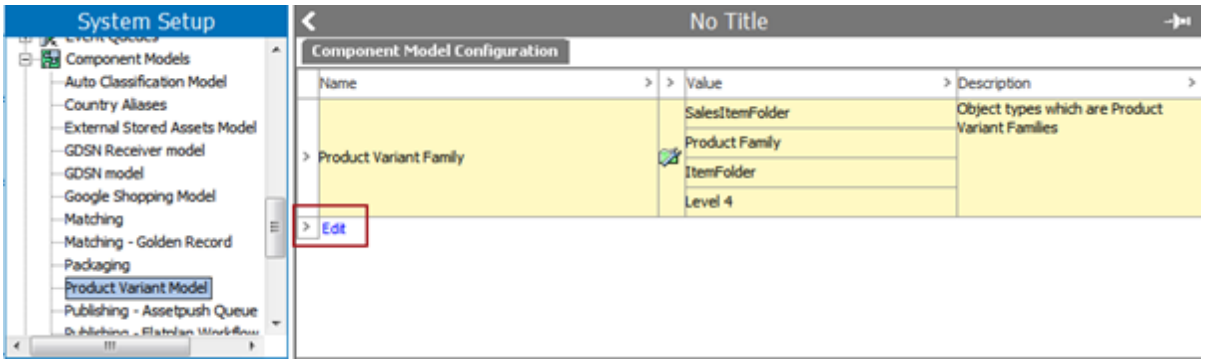
Setting Up Product Variant Families

To make a parent folder in STEP work as a product variant family, go through the following major steps.

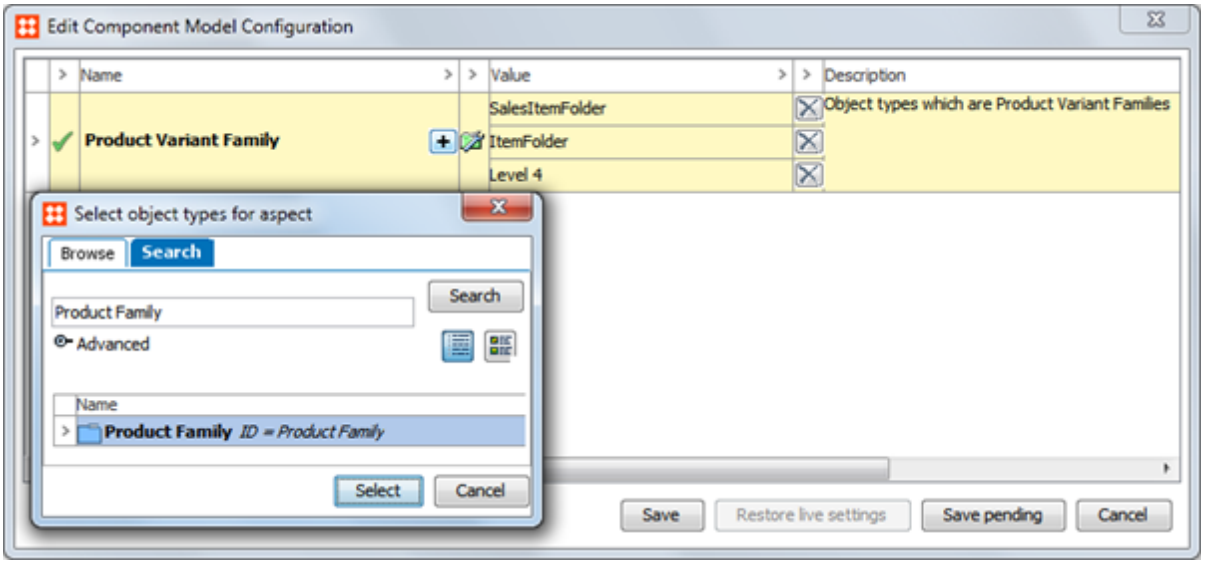
1. Add the object type of the parent folder to the component model **Product Variant Model** to enable the product variant capabilities.
2. Create a meta-data attribute and make it valid on the links from the product folder to the related attributes. This enables you to specify which attributes are variant attributes.
3. Add values to the meta-data attribute on the links from the product folder to the intended variant attributes and non-variant attributes. This allows them to work as variant and non-variant attributes.

Add the object type to the component model

1. In **System Setup**, expand **Component Models**, and then select **Product Variant Model**.
2. On the **Component Model Configuration** tab, click **Edit**.

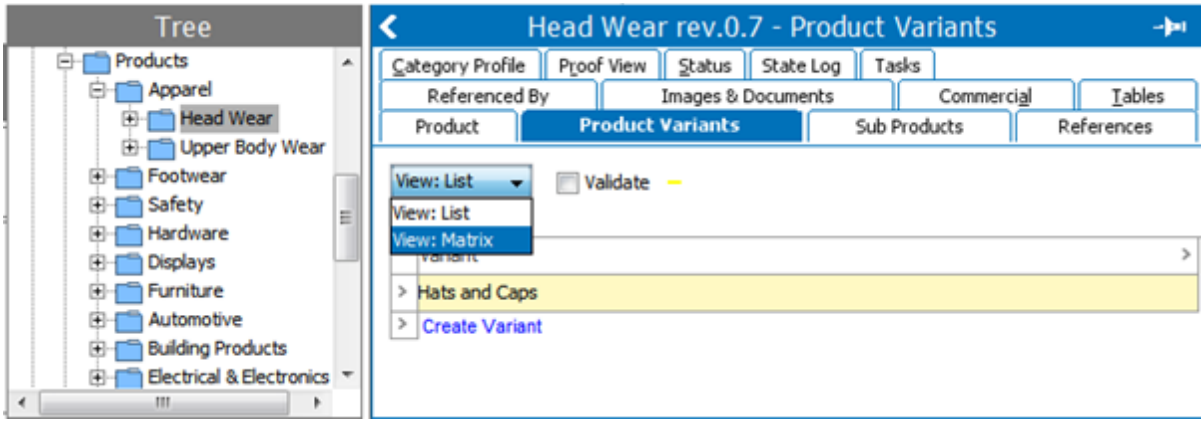


3. In the **Edit Component Model Configuration** dialog, click the plus sign, choose the object type of the parent folder, and then click **Save**.



After having completed this step, a **Product Variants** tab is available for all products of the selected object types.

4. To view the **Product Variants** tab, select an object in the **Tree** in the primary product hierarchy. The tab has two views for managing product variants: **List** view and **Matrix** view.

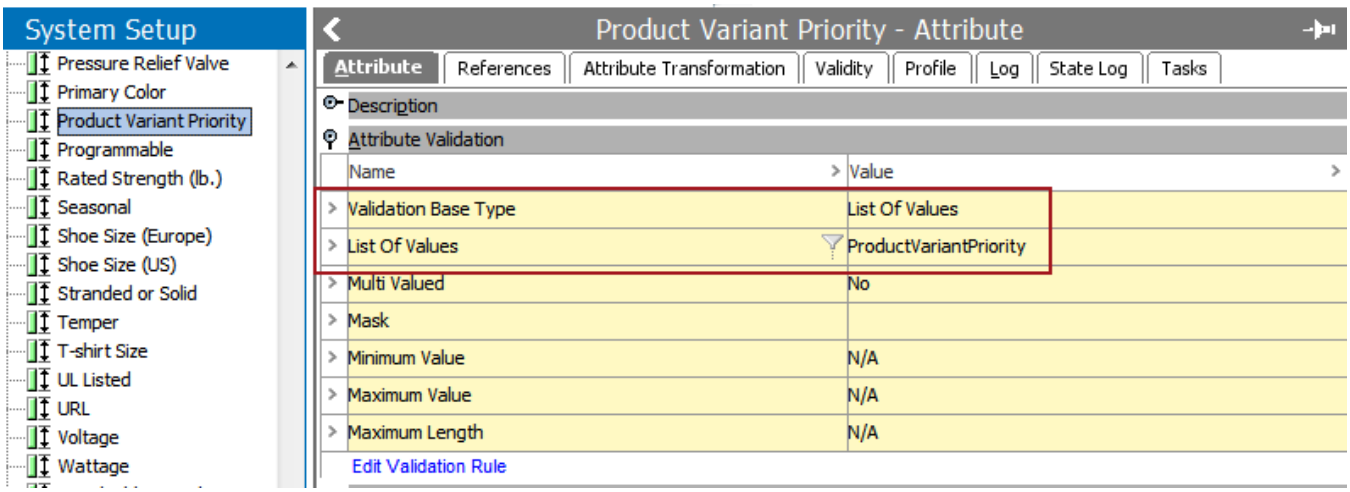


Set up a meta attribute

Use a meta attribute to specify which attributes should be variant attributes. This meta attribute must be valid on links from the parent folder to the attributes.

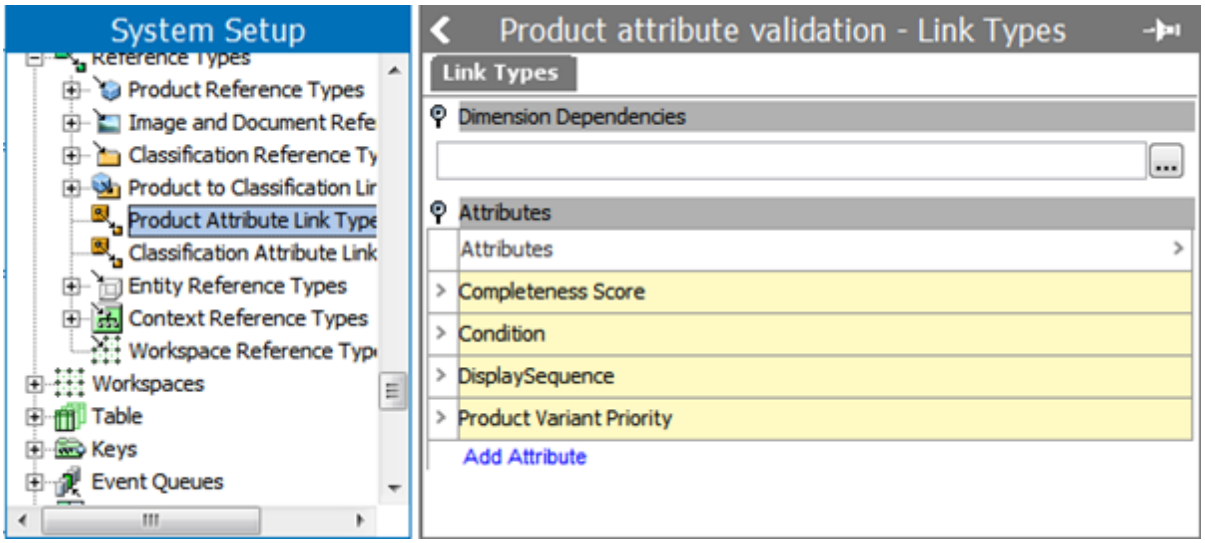
It is recommended that you use a List Of Values attribute as variant priority attribute.

1. Create a list of values with numbers and possible numbers with the prefix "n" prefix: 1,2,3,4,5,6,7,n1,n2,n3,n4.
2. Create a description attribute with the created LOV as validation type. In the following screenshot, the attribute is named variantpriority, and it is validated using the List Of Values variantlov.



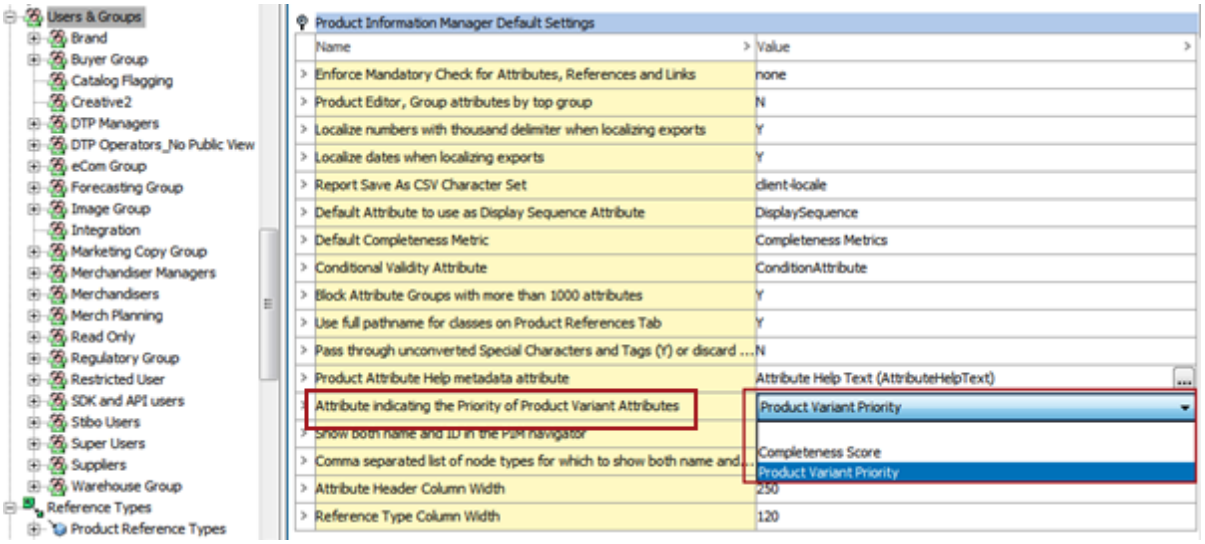
Make the attribute valid on product to attribute links:

3. In **System Setup**, expand **Reference Types**, select **Product Attribute Link Type**, and then add the attribute you just created.



Next, specify that the created attribute is to be used as variant priority attribute.

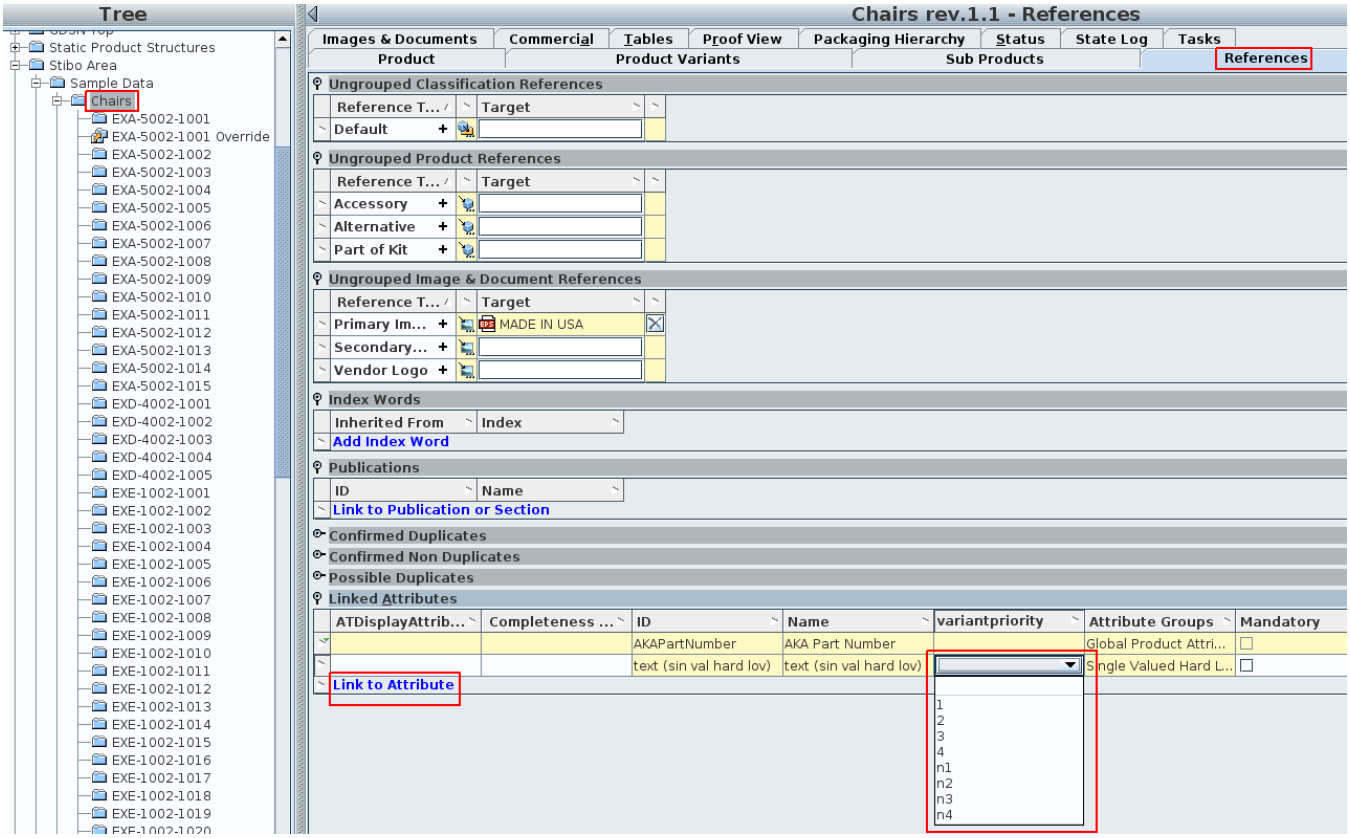
4. In **System Setup** , select **Users & Groups**.
5. On the **System Settings** tab, in the **Product Information Manager Default Settings** area, select the attribute from the **Value** list next to the**Attribute indicating the Priority of Product Variant Attributes**.



Specify variant and non-variant attributes

1. In the **Tree**, expand the **Primary Product Hierarchy**, and select the relevant product folder.
2. Click the **References** tab.
3. In the **Linked Attributes** area, specify the display priority of the linked attributes.

In the following screenshot, "Chairs" is the parent folder and the linked attribute is "text (sin val hard lov)".



Using Variants List View

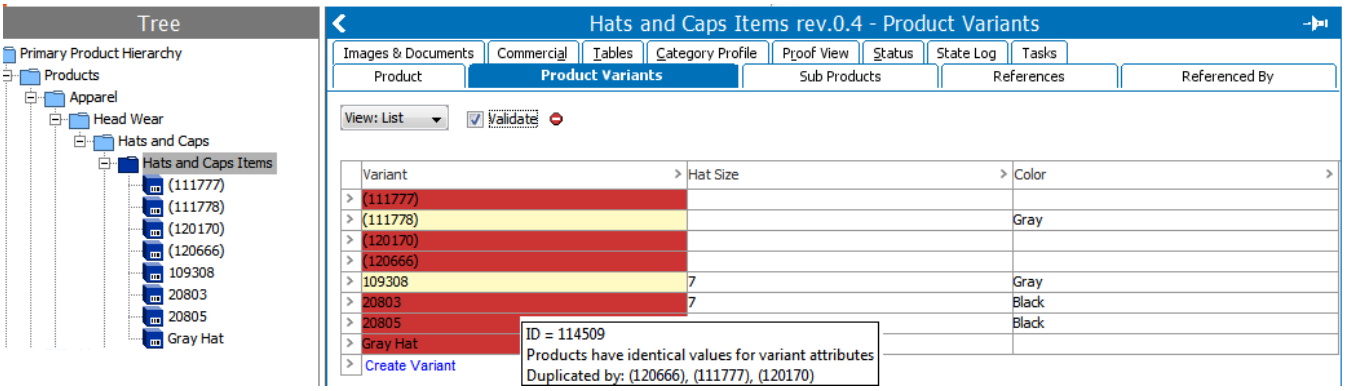
The Variants List View provides an overview of all variants and their variant attribute values. You can also create and delete variants and modify variant values.

Variant products are listed in rows and variant attributes are listed in columns.

You can use the List View to locate possible duplicate values so that you can review the duplicates and decide what to do with them.

To view product variants in list view

1. In the **Tree**, expand the **Primary Product Hierarchy**, and then select the relevant product.
2. Click the **Product Variants** tab, and then from the **View** list select **List**.
3. Check **Validate** if you want to view product variants with possible duplicates. Variants are possible duplicates if the variant attributes are identical. The possible duplicate variants are highlighted in red.



When you hover the mouse over a product variant, information is displayed about other variants that may be possible duplicates. The values of other attributes are not considered.

Note: When **Validate** is selected, validation runs in the background, and the results are updated whenever changes are made. If you have made a validation and then clear Validate afterwards, you can still view the results, but they are not being updated.

To delete a variant

- To delete a variant, click the arrow to the left of the row, and then choose **Remove variant**.

Product variants list in the portal

In the portal, the **VariantsListTabPage** corresponds to the Product Variants list. You can insert it into a **TabControl** on a **NodeDetails** screen.

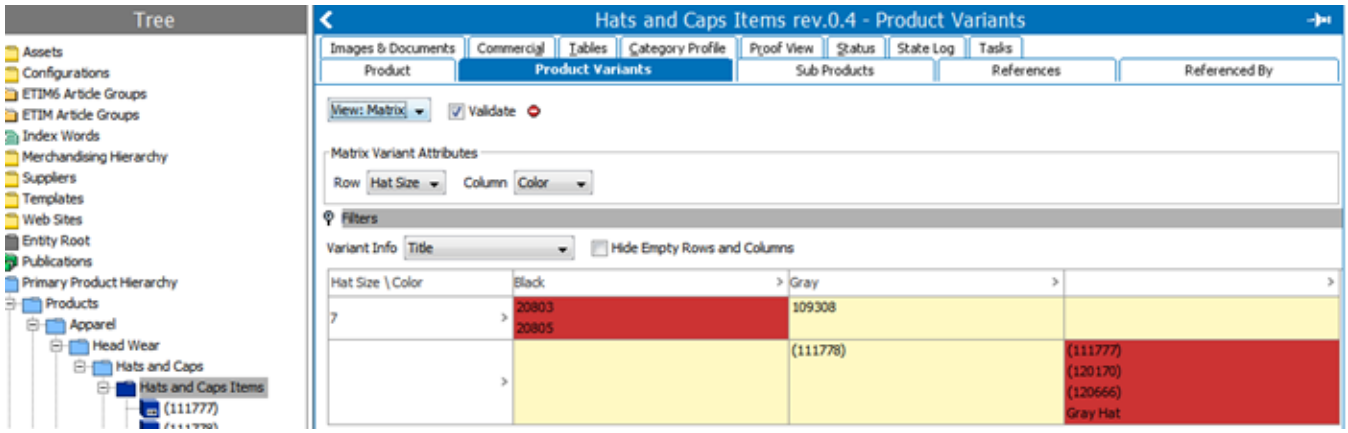
Using the Variants Matrix View

The Matrix view is organized into columns and rows with a variant attribute as row and another variant attribute as column.

Each product variant is placed in a cell with attributes that match the selected row and column attributes. Because a cell contains all product variants that match the selected attributes, it is easier to spot possible duplicates.

To view product variants in matrix view

1. In the **Tree**, expand the **Primary Product Hierarchy**, and then select the relevant product.
2. Click the **Product Variants** tab, and then from the **View** list select **Matrix**.



3. From the **Row** and **Column** lists, select the attributes you want to use as row and column attributes.
4. Check **Validate** if you want to view product variants with possible duplicates. Variants are possible duplicates if the variant attributes are identical. The possible duplicate variants are highlighted in red.
5. In the **Filters** area, you can filter the view on the product variant attributes that are not used as row or column

When you hover the mouse over a product variant, information is displayed about its variant and non-variant values and possible duplicates.

It is also possible to display only the products where the attribute value is empty.

To delete a variant

- To delete a variant, right-click the relevant cell, and then choose **Remove variant**

To create a new variant

- To create a variant, right-click the relevant cell, and then choose **Create variant**. A variant is created with attribute values that correspond to those of the selected cell.

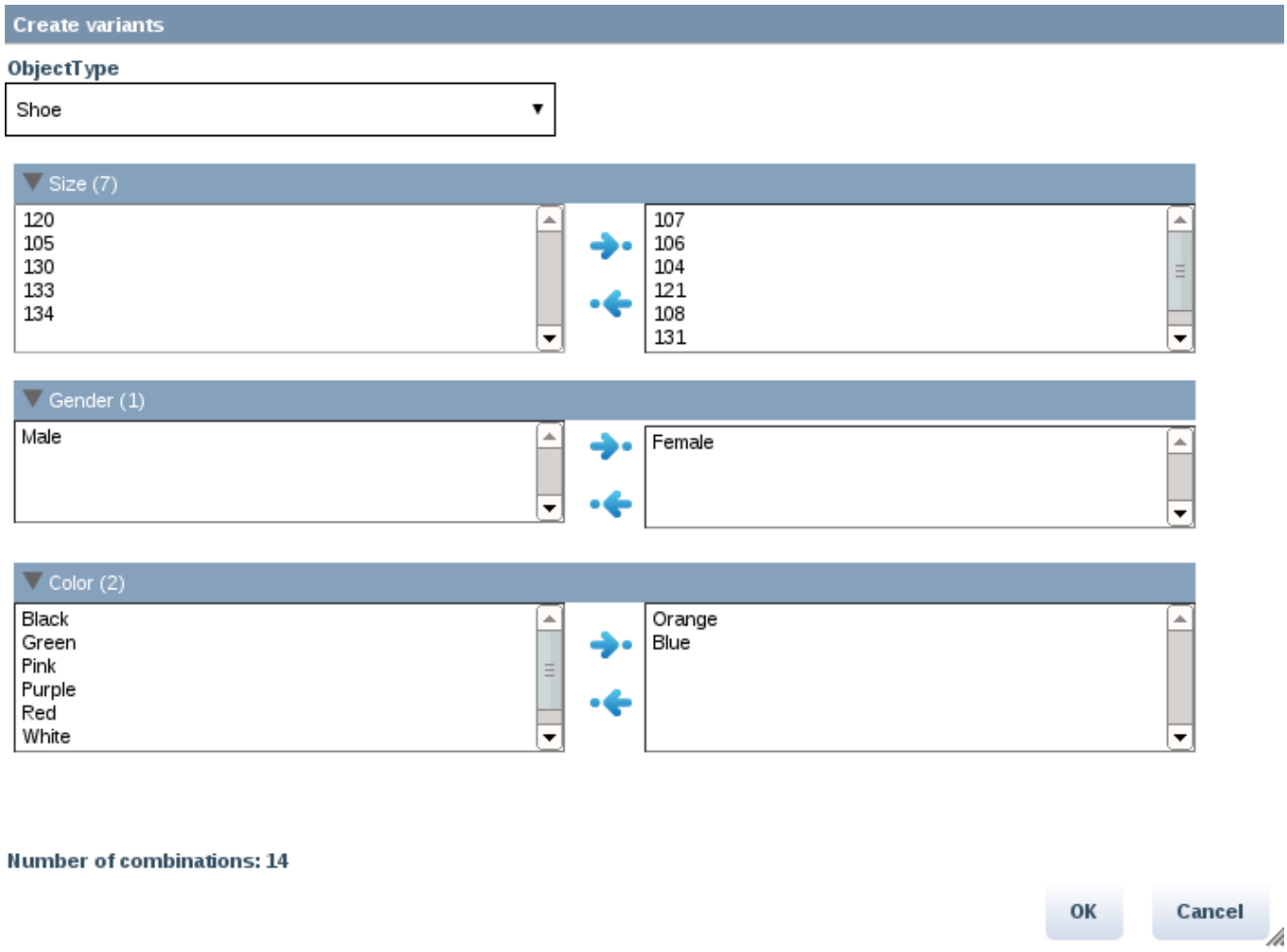
In the portal, the **VariantsMatrixTabPage** corresponds to the Product Variants matrix. You can insert it into a **TabControl** on a **NodeDetails** screen.

Creating Variants Through Web UI

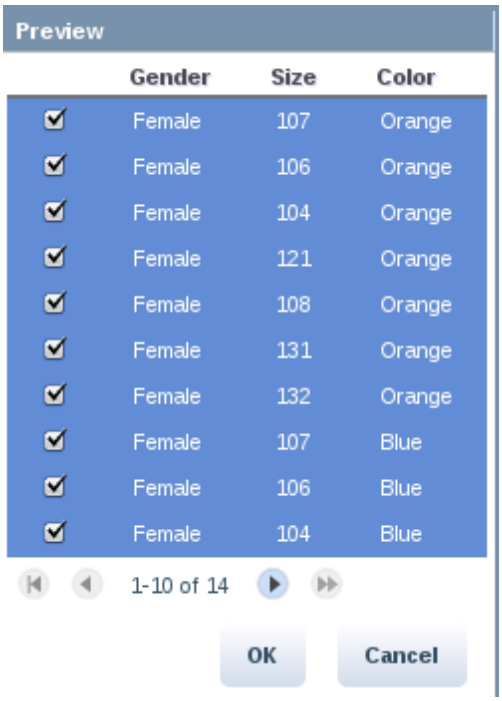
In the Web UI, you can insert a **CreateVariantsAction** into a **VariantsMatrixTabPage**. This allows you to generate a variety of combinations of variants over the variant attributes.

For detailed information about variants in the Web UI, see the Web UI documentation.

In the **Create Variants** dialog, you choose the variant attributes you want to combine.



If the parameter **PreviewVariantsCombinations** has been checked on the **CreateVariantsAction** in design mode, a preview dialog is displayed before the chosen combinations are created. This prevents specific combinations from being created.



On the CreateVariantsAction in design mode, there is also a **PreventDuplicateCreation** check box. If the parameter is checked, the combinations that match an existing variant will not be created when the action is applied.

Business Condition: Validating Product Variant

It is possible to use a business condition to perform validation for product variant duplicates.

For more information about business conditions, see Overview of Business Rule Conditions in the Business Rules documentation.

To use a business condition to validate variants, first create a new business condition.

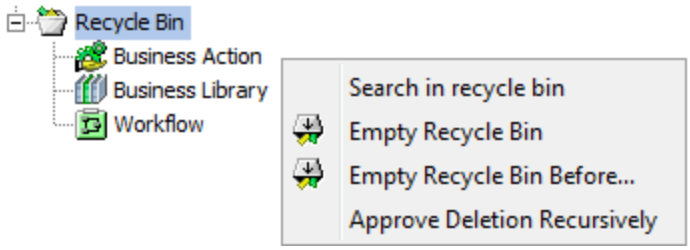
1. In **System Setup**, expand **Business Rules**, and select the relevant business condition.
2. On the **Business Rule** tab, in the **Valid Object Type** field, click the ellipsis button **(…)**, and then select the object types that you want to apply the action to.
3. In the lower left corner, click **Add New Business Condition**.
4. Click the **Edit Operation** icon.
5. In the **Edit Operation** dialog, from the drop-down list, select **Validate Product Variant**.

The Validate Product Variant condition uses the configuration your created when you set up Product Variant Families, so there is no configuration to do. The action works on the Product Variant level, and returns false if the product variant is a duplicate.

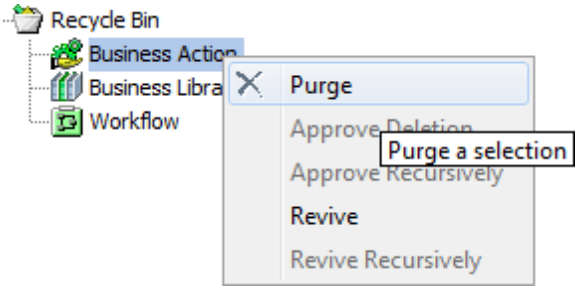
Note: Running the Validate Product Variant business rule may affect performance depending on how many siblings and variant attributes that have to be checked.

Recycle Bin

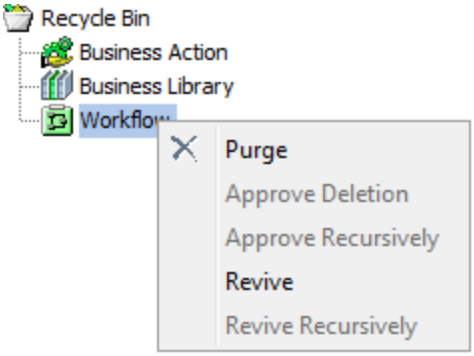
System Setup contains a recycle bin.



Deleted business rules can be seen in the Recycle Bin and can be revived from that location. Similarly they can be purged.

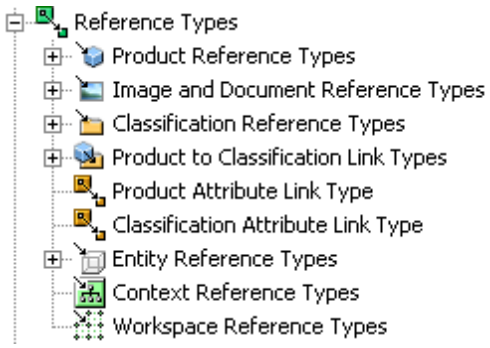


Deleted workflows can be seen in the System Setup Recycle Bin and can be revived from this location. Similarly they can be purged from the bin.



Reference and Link Types

This section describes how to set up Reference Types and Link Types which are managed in **System Setup**: **Reference Types**.



About References Types

A Reference Type is a set of rules that defines how Objects can be referenced. The following types of References can be created:

- Product to Product references
- Product to Image references
- Product to Entity references
- Classification to Product references
- Classification to Image references
- Classification to Classification references
- Classification to Entity references
- Entity to Entity references
- Entity to Product references
- Entity to Classification references
- Entity to Asset references
- Entity to Context references
- Entity to Workspace references

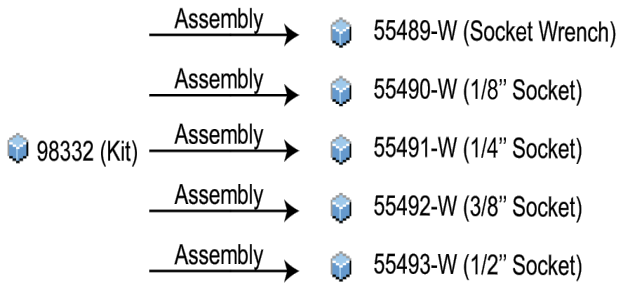
Direction of a Reference

Each Reference Type establishes the position among the referenced objects. The direction will identify one object as the 'source' and the referenced object as the 'target'.

Source: The Object that the reference is made from

Target: The Object that the reference is made to

For example, a 'kit' is a set of objects that are sold as a single product. The sellable product has its own order code. Each of the set of products may also have a order code as they also could be sold as individual products.



In this example, a socket wrench has order code **55489-W**. A buyer could purchase only this socket wrench, only individual sockets, or a kit that includes the socket wrench and five individual sockets. The kit itself has order code **98332**. A product reference called Assembly is created to establish the references between the products. The kit (order code 98332) can then reference order code 55489-W as well as the order codes of the other sockets.

98332 represents the source object, and the component parts order codes (e.g., **55489-W**, **55490-W**, **55491-W**...) represent the target objects.

Important: When creating a reference, verify that the 'from' and 'to' are assigned correctly. 55489-W is an 'Assembly' to 98332, but 98332 is not an 'Assembly' to 55489-W.

Building complex assemblies

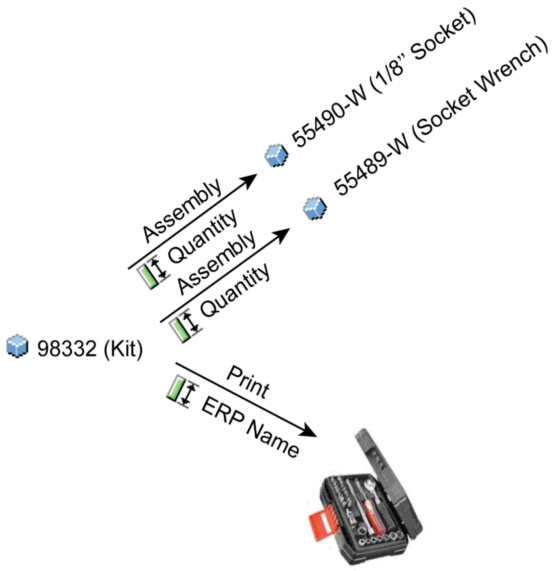
The direction scheme makes it possible to build large kits or assemblies, the components parts of which are themselves assemblies, which themselves comprise of smaller assemblies, and so on. By always having a direction of a references, i.e., a source / target scheme, the system can support rather complex assemblies.

Icons used for References	Description
	Entity, Product, Image, Classification Reference Types
	Inherited Reference Type
	Mandatory Reference Type
	Inherited and Mandatory Reference Type

Metadata Attributes

Description Attributes can be applied a **Reference Type**. It is possible to apply different **Attributes** for different types of Product, Classification and Asset References.

For example, product 98332 (KIT) includes two sockets and is referencing an image to be used for Print. Different Metadata Attributes are needed for references applied Images and references applied Products.



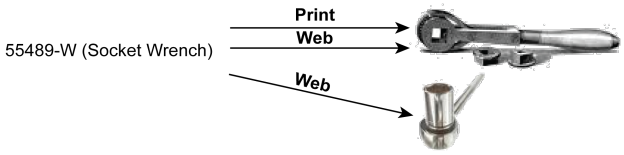
- A **Description Attribute** named **Quantity** has been applied the Reference Type **Assembly**. This Attribute holds number of Sockets to be included in the KIT.
- A **Description Attribute** named **ERP Name** has been applied the Reference Type **Print**. This Attribute holds the name of the Reference used in an ERP System.

Metadata Attributes are applied a Reference Type, from the Validity tab in the Attribute Editor.

Multiple references of different types between same two objects

It is possible to create more references between same two objects but with different Reference Type. Typically such references are created as they have to used for different purposes. The Reference Types might also have different restrictions applied.

For example, a product with order code 55489-W (Socket Wrench) should be represented with one Image for Print and more Images for Web. To establish such restrictions, then two Reference Types are created **Print** and **Web**.



- The same Image is referenced with Reference Type Print and Web from Product 55489-W (Socket Wrench).
- The Reference Type Web allows references to more Images.
- The Reference Type Print only allows one reference to an Image.

For more information, see Changing Reference Types Settings.

Note: The same target (subordinate) can always be referenced from same source (master). The only restriction is that different Reference Types have to be used to reference the Target.

Mandatory References

A Reference Type can be set to be Mandatory. Setting a Reference Type to Mandatory is an indication that the reference is needed to be applied from a Product, Classification or Asset. Approving the Master Object will be prevented, if mandatory references have not been created.

For example, a Product Reference Type named **Part of Kit** is set to be **Mandatory**. It is set up to be valid for Source Object Types **Product Family** and Target Object Types **Articles**.

It is now mandatory that a reference of type **Part of Kit** is established from Products with Object Type **Product Family** to Products with Object Type **Articles**.

Approving a Product with Object Type **Product Family** will be prevented, unless a Reference of type **Part of Kit** has been established.

To set a Reference Type as Mandatory, see Changing Reference Types Settings.

Externally Maintained References

A Reference Type can be set to be **Externally Maintained**. When using an Externally Maintained Reference between two Objects, the Reference will automatically be included in all Workspaces and the Master Object will remain in its current Approval Status.

Important: The Master Object will get an unapproved status if Metadata Attribute values are changed on the reference, and the Metadata Attributes are not Externally Maintained.

For example, References to Vendor Logos have to be imported and not be under Revision Control as data comes from an ERP system.

In this case an **Image Reference Type** named **Vendor Logo** is set to be **Externally Maintained**.

- Importing references from Products to Vendor Logos using the Reference Type Vendor Logo, will automatically make the Reference available in all Workspaces.
- The Approval Status of all Products will remain untouched.

Important: Changing Metadata Attribute Values on Attributes applied the Reference Type Vendor Logo will trigger an Unapproved status of the Products where the Vendor Logos are referenced from in case the Metadata Attributes are not set to be Externally Maintained.

For more information, see Changing Reference Types Settings.

Dimension Dependencies for Reference Types

A Reference Type can be setup to be dependent of a Dimension. When a Reference Type is setup to be dependent of a Dimension, it is typically made dependent of one Dimension only, for example, the language Dimension. A Dimension dependent Reference Type is used to specify in which Contexts a Reference should

appear.

Important: It is possible to have multiple references of the same type from same Master Object but with different Dimension Points. However, it is not possible to make one reference valid for multiple Dimension Points. For more information, see Changing Reference Types Settings.

Suppression of a Reference

A Dimension dependent Reference Type / Product to Classification Link Type enables the option to suppress References /Product to Classification Link Types in specific Dimension Points.

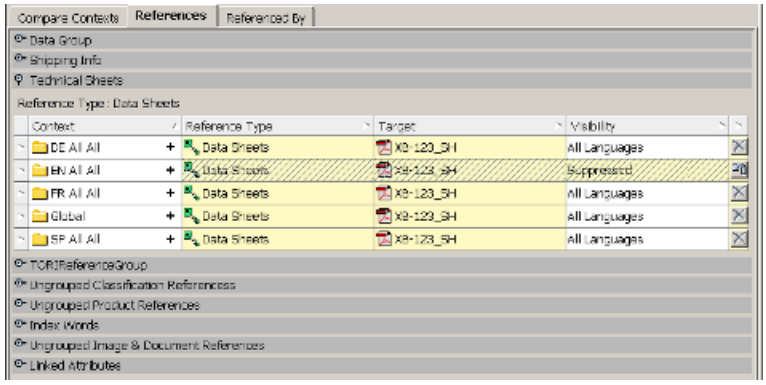
For example, a Product needs a **Data Sheet** reference. The Data Sheet should be linked to the Product in all language versions of the Product except the English language version of the Product.

- A Reference Type is created. It is setup to be language dependent. In this example, the **Reference Type** is named Data Sheet.
- In the Context pointing out **All Languages** a Data Sheet Reference is created. The Reference is inherited to all languages.
- In the Context using the English language the Reference is suppressed by pressing the Delete button. The Reference will appear as shaded.

Using Context Viewing Mode

In the Toolbar you can select different Viewing Modes. Selecting the **Context Viewing Mode** will show a comparison view of Attribute Values and References in different Contexts.

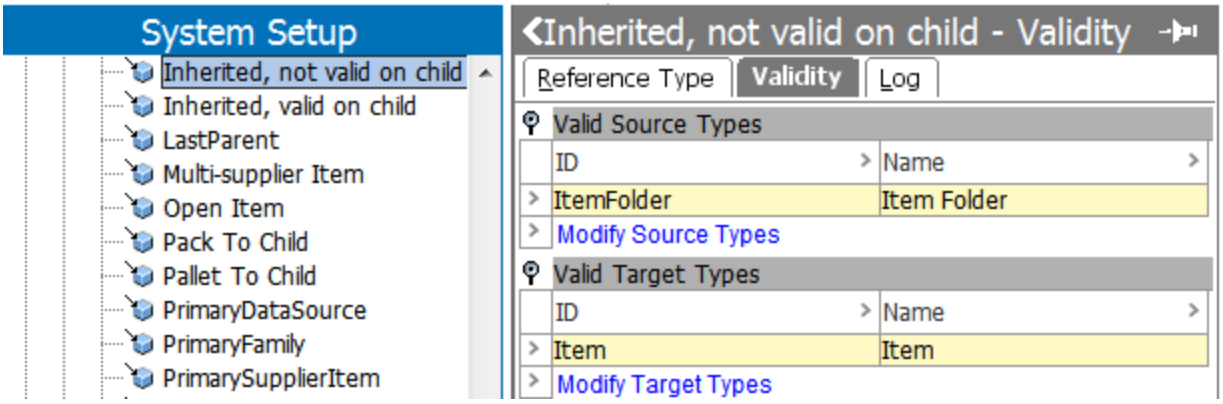
1. In the toolbar, enable the Context Viewing mode, by clicking .
2. In View menu, select Target, and choose the Contexts for the comparison View to include.
3. In the References tab / Referenced By tab, the comparison view enables maintaining References in each shown Context. In the example below, the Delete icon has been pressed on the **Data Sheets** Reference in the EN All All Context. Pressing the Delete button will suppress the Reference in EN All All as it is inherited from All Languages.



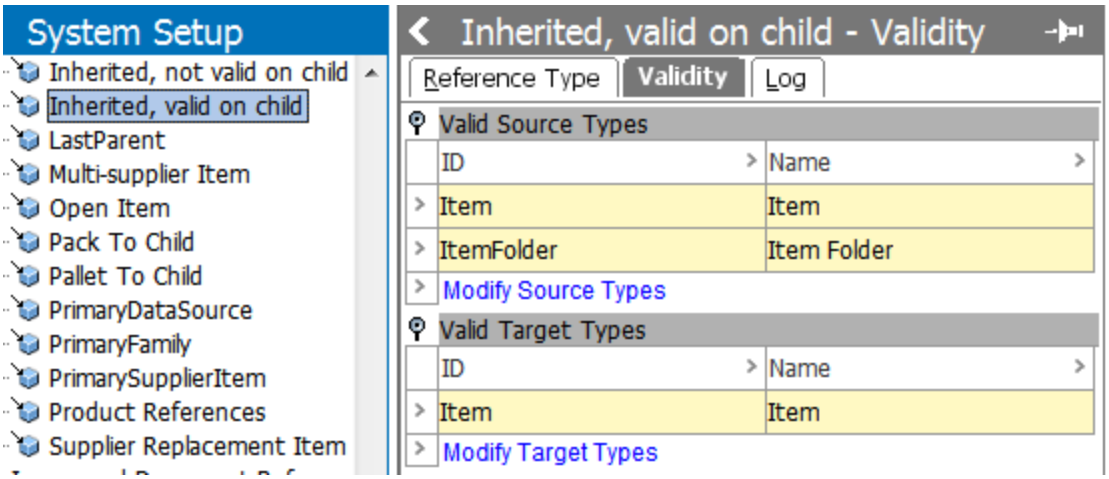
Inheritance of References

Before defining types of inheritance on references, it is important to keep in mind the valid sources and targets for a reference. If a reference is valid on an object, it can also be edited there. However, a reference that is not valid on an object can still be visible on the object, due to the reference being inherited from a parent.

For example, consider a case where an Item Folder object type has valid child object types called Item. A reference is created called 'Inherited, not valid on child' that has the following validity:

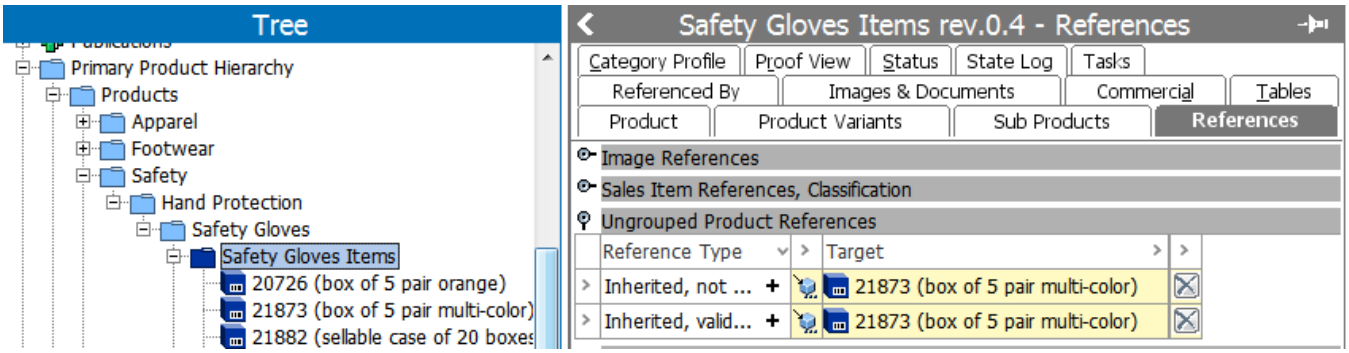


The child object type is a valid target of the reference, but not a valid source object. In contrast, a second reference is created called 'Inherited, valid on child' that has the child object valid as both a source and a target of the reference:

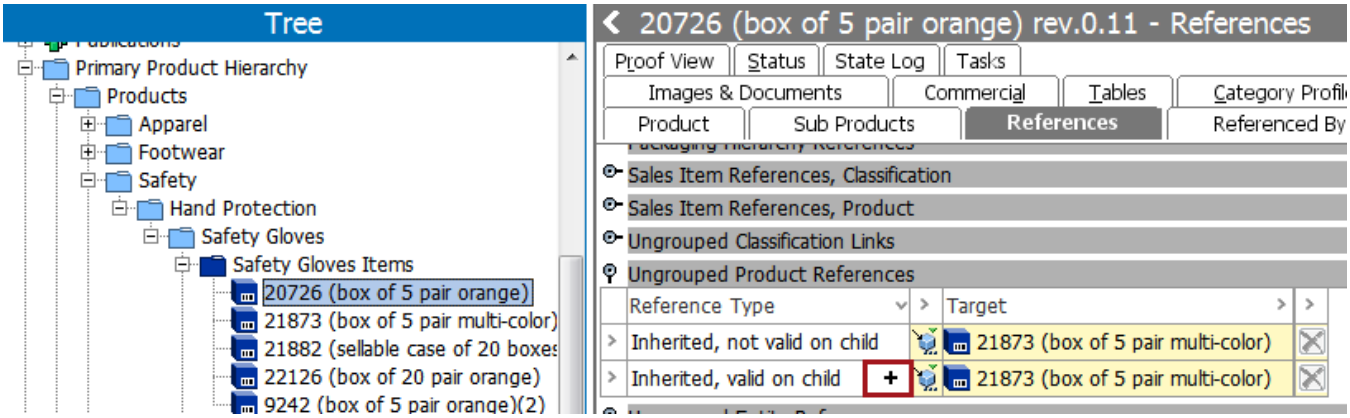


When both references are set with some type of inheritance, any reference added to the parent (Item Folder object) will also be present on the child (Item object).

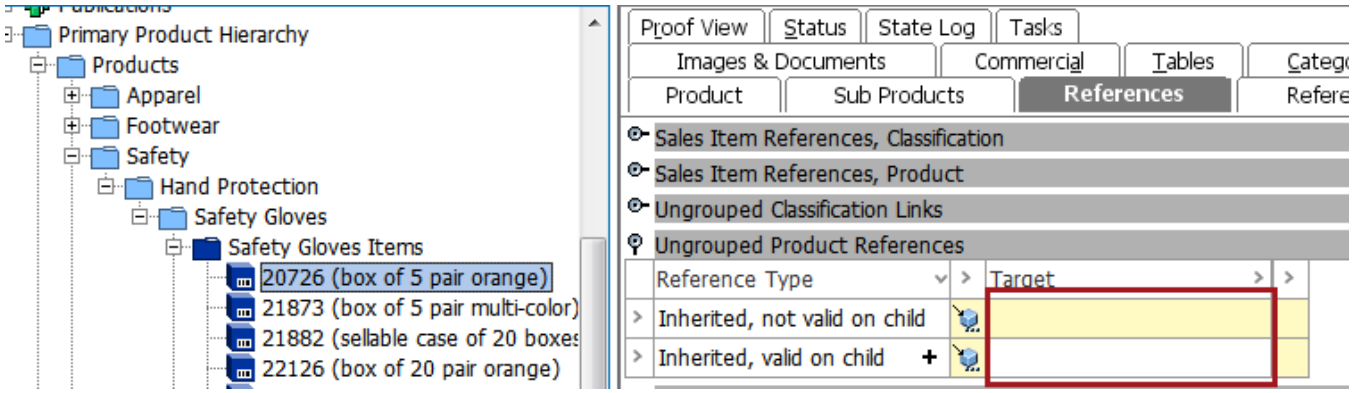
Parent object:



Child object:



Note that only the reference that is also valid on the child has a [+] available, indicating it is editable. This is perhaps better illustrated when the references are removed:



It can easily be seen that the inherited references are displayed / not displayed on the child object based on presence / absence of a referenced target on the parent. However, only references that also have the child object as a valid source of the reference are also editable on the child object.

Keeping validity in mind, you can specify three different settings for the inheritance of reference types:

- **Inherited:** With the Inherited setting, a reference is inherited to all child objects descending from the source (parent) object, and the inherited reference points to the same target(s). If you create a reference of the

same reference type on an object below the source object (assuming the child object is also a valid source of the reference), it overwrites the inherited reference.

A sample use case for the Inherited setting would be for a product family / product category where the majority of the child objects use the same references. For example, a product category of remote controls named 'Remotes' contains child remotes that mostly use AA batteries. A ‘Battery’ reference would be set on the Remotes category to a AA battery product. But, a few remotes in the category require AAA batteries. So, the reference on those remotes would be changed to reference a AAA product instead. This setup helps to minimize work by not requiring an individual Battery reference to be set on every single product. If 60 percent+ use something common, this would be set on the parent and the references on the other 40 percent would be changed individually.

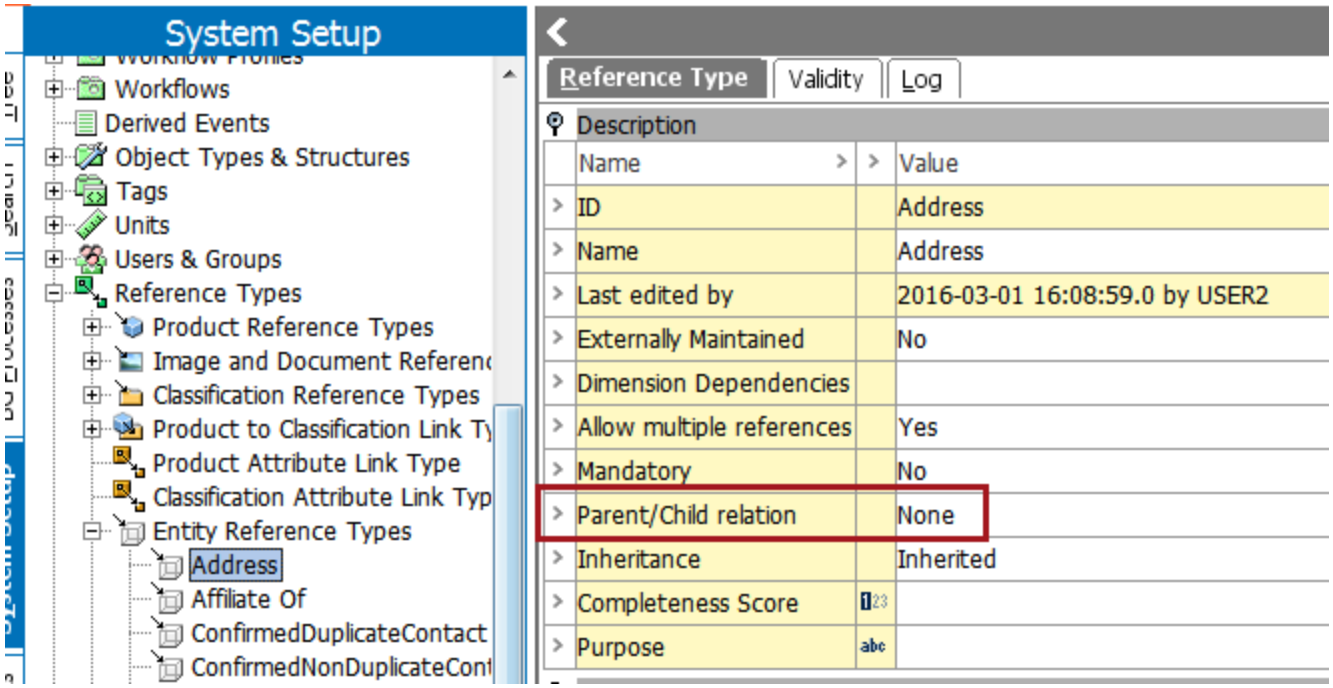
- **Accumulative:** With the Accumulative setting, child objects descending from the source (parent) object inherit the reference from the source object, and the inherited reference points to the same target object(s). However, you can also create a new reference of the same reference type on a child object (assuming that the child object is a valid source of the reference). This reference is then listed along with the inherited reference, and it is also inherited by its descendant objects (along with the reference from the source object). (**Note:** The Accumulative inheritance setting cannot be used on single-valued reference types; that is, link types where 'Allow multiple links' is set to 'No.')

  An sample use case for the Accumulative setting would be for a product family / product category where the child products use the same overall image, but the individual child products need additional linked images. This would not be valid for an image reference such as Primary Image, since only one primary image can be used per product. A relevant example would be for an asset reference type such as 'Lifestyle' images. For example, a product category called 'Picnic Supplies' would have a 'Lifestyle' product image (linked at the parent level) of a family having a picnic in a park. The individual child products should have the same overall image, but each product should also have additional product-specific images. So, for a child product such as a plate, another 'Lifestyle' image would be linked that shows someone putting a piece of pie onto the plate. So, the plate product now has two linked 'Lifestyle' images: the family image that is inherited from the parent product, and the pie on the plate from the local reference.

- **None:** When you set inheritance to None, a reference is not inherited by child objects descending from the source (parent) object. Likewise, a reference on an object below the source object is not inherited by the descendants of the lower level object.

Using the Parent / Child Relationship with Entity to Entity References

The 'Parent/Child relation' field allows an aggregated display of all Entity to Entity reference types in the tree hierarchy navigator.



Note: The Parent/Child relation parameter is only available on references that have *only* entity object types as sources and targets. If any object type is present on the Validity tab of another supertype (e.g., products or classifications), the parameter will not be present.

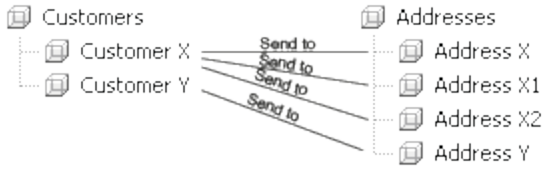
Both the parent and the child objects must be of the Entity object type. On the Entity Reference Types Editor, the Parent/Child Relation field includes these options:

- **None** - Entity to Entity references are not aggregated.
- **Source as parent, Target as Child** - Entity to Entity references are aggregated below the Source Entity (where references originate).
- **Target as parent, Source as Child** - Entity to Entity references are aggregated below the Target Entity (where references end).

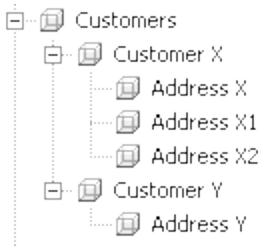
The following example includes:

- a **Customers** Entity hierarchy
- an **Addresses** Entity hierarchy
- a **Send to** Reference Type which is used to link each Customer to its relevant Address(es).

The image below shows that Customer X has three Addresses and Customer Y has one Address. Each of the Addresses are referenced from Customer using the **Send to** Reference Type.



On the **Reference Types** editor, the Reference Type **Send to** has the **Parent/Child Relation** field set to: **Source as parent, target as child**. Customer is the Source and Address is the Target since a Reference is created from a Customer to an Address. On the Tree navigator, expanding the Customer Entity hierarchy displays all referenced Address Entities, as shown in the image below:



When setting the Parent/Child Relation field, be aware of these items:

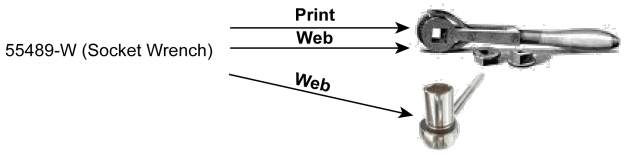
- Different icons for each Entity make the hierarchy overview easier to understand. See the **Object Type Icons** section of the **System Setup / Super User Guide**.
- Aggregated references enables **Copy**, **Paste**, **Drag** and **Drop** between Entity hierarchies. STEP will automatically apply the reference at the new position.
- Entity Objects Types can be set up to be globally revisable or internally revisable. For more information about this functionality, see the **Object Types and Structures** documentation.
- STEP prevents you from setting the Parent/Child Relation field if the objects that are parent and child of the reference already have a true parent / child relationship via the data model. If either object type appears under the Parents flipper on the References tab of the object type, a parent/child relation will not be allowed (the system will error when a user attempts to set the parameter).

Reference Types in Attribute Groups

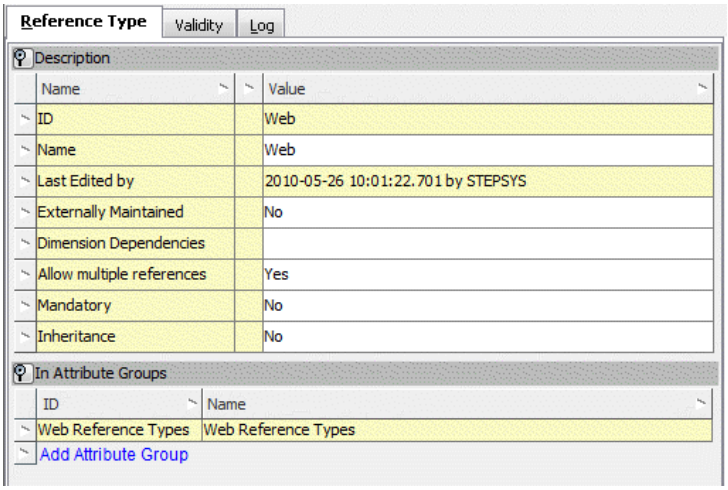
Reference Types can be applied one or more **Attribute Groups**. Grouping **Reference Types** into **Attribute Groups** opens up the ability to setup **User Privileges** where **Approval of References** is only enabled for specific **Reference Types**. Users will only be allowed to Approve **References** of the **Reference Types** linked into the **Attribute Group**. Grouping **Reference Types** into **Attribute Groups** also opens up the ability to setup a customized view of all **Reference Types** linked into the Attribute Group. The customized view is defined on the Attribute Group and will have effect on all **Reference Types** linked into the **Attribute Group**. It defines General Aspects, Metadata Attributes, Attributes on referenced Objects to be shown in the Reference Editor on Products, Classifications and Assets. For more information, see the Creating a Customized View section in the Attribute Groups / System Setup documentation.

Important: Approving a Reference on a Product, Classification or Asset having a customized view defined will approve all General Aspects and Metadata Attributes defined on the Reference. Approving the Reference will not only Approve the General Aspects and Metadata Attributes shown by the Customized View, but Approve all General Aspects and Metadata Attributes defined on the Reference Type.

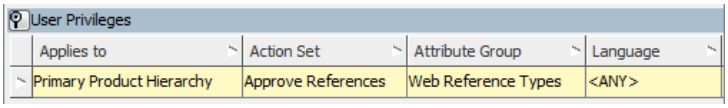
For example, a User should only be allowed to Approve References used for **Web**. Approving Product 55489-W (Socket Wrench) should in this example allow approving the Images used for Web.



The Reference Type named **Web** is added to an Attribute Group named **Web Reference Types**.



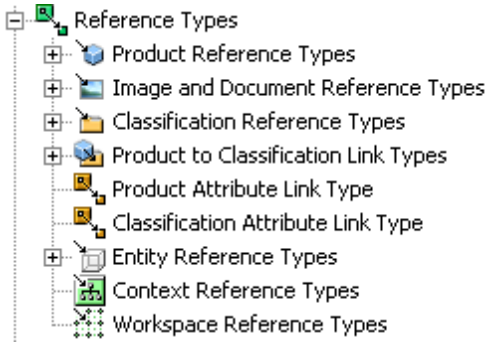
A User Privilege is setup to include the **Approve Reference** Privilege for Reference Types in the Attribute Group **Web Reference Types**. For more information about setting up privileges, see **About Action Sets** in the **System Setup / Super User Guide**.



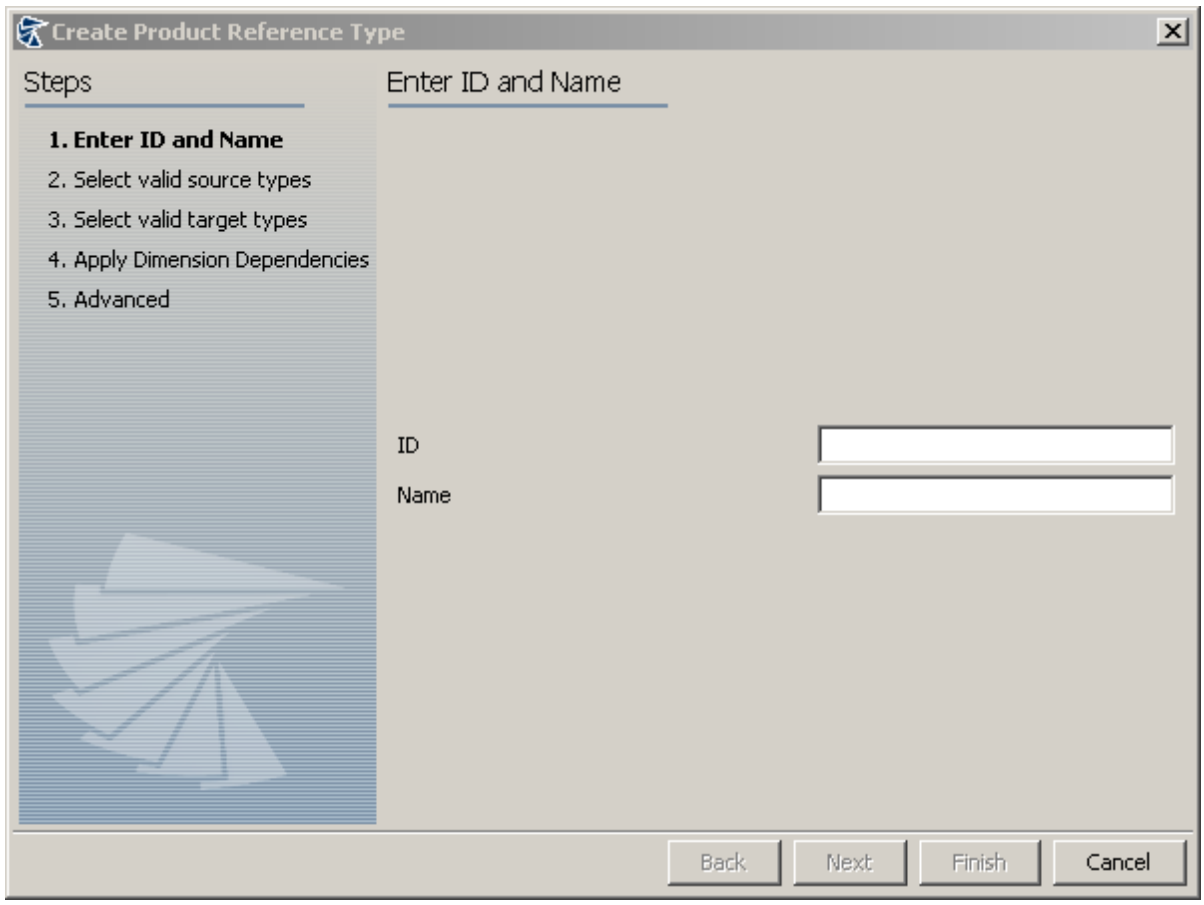
A User with this privilege setup will only be allowed to Approve References of Reference Types Web.

Creating Reference Types

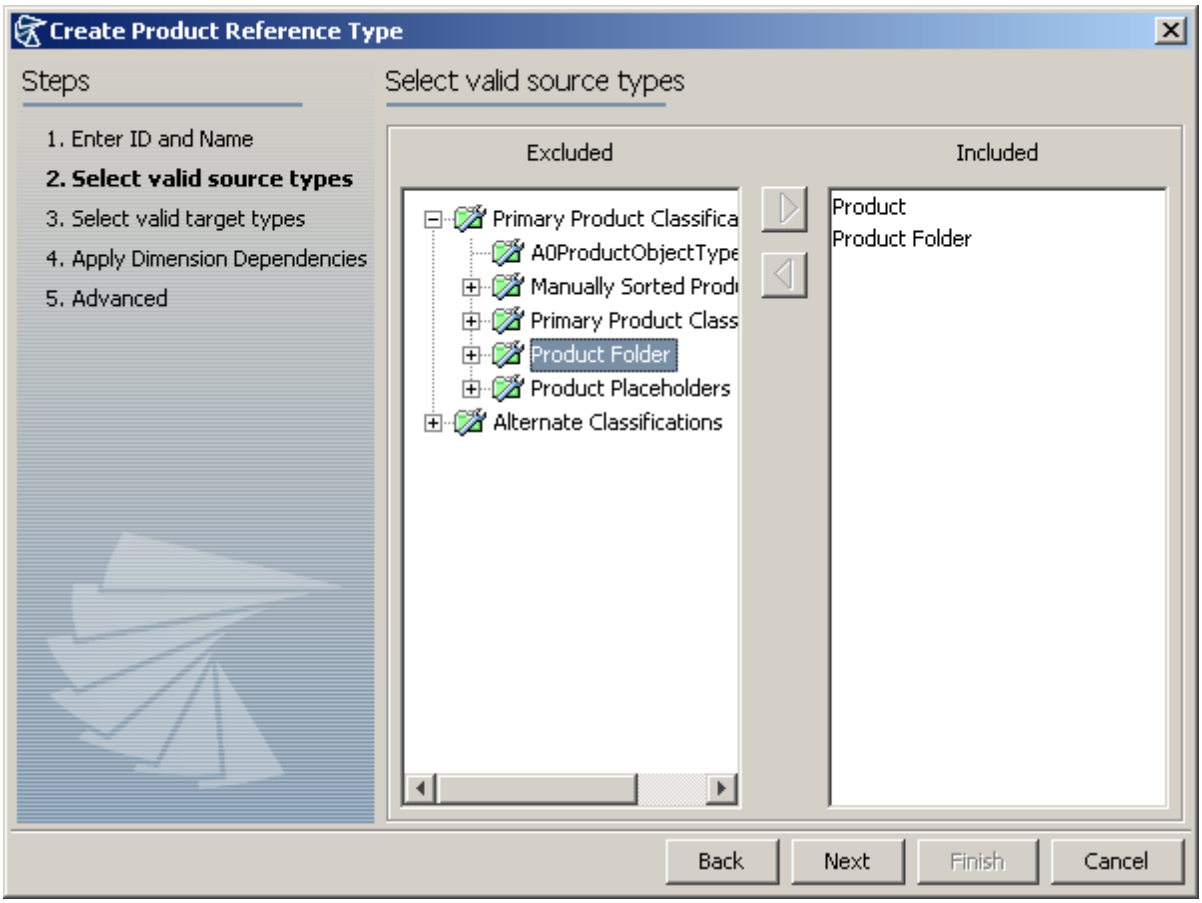
1. In **System Setup**, click the **Reference Types** Hierarchy.



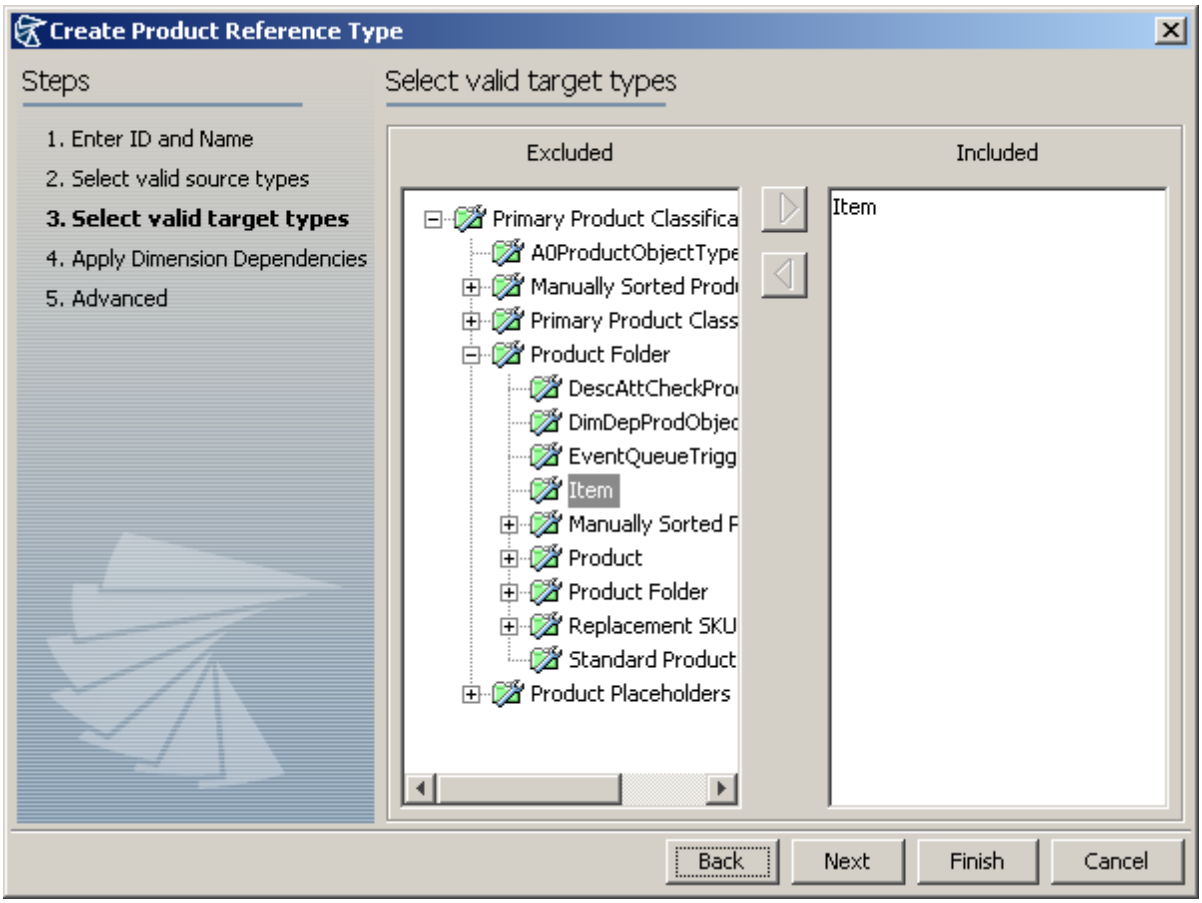
2. Right-Click relevant **Reference Type** to be created, click **New Reference Type**.



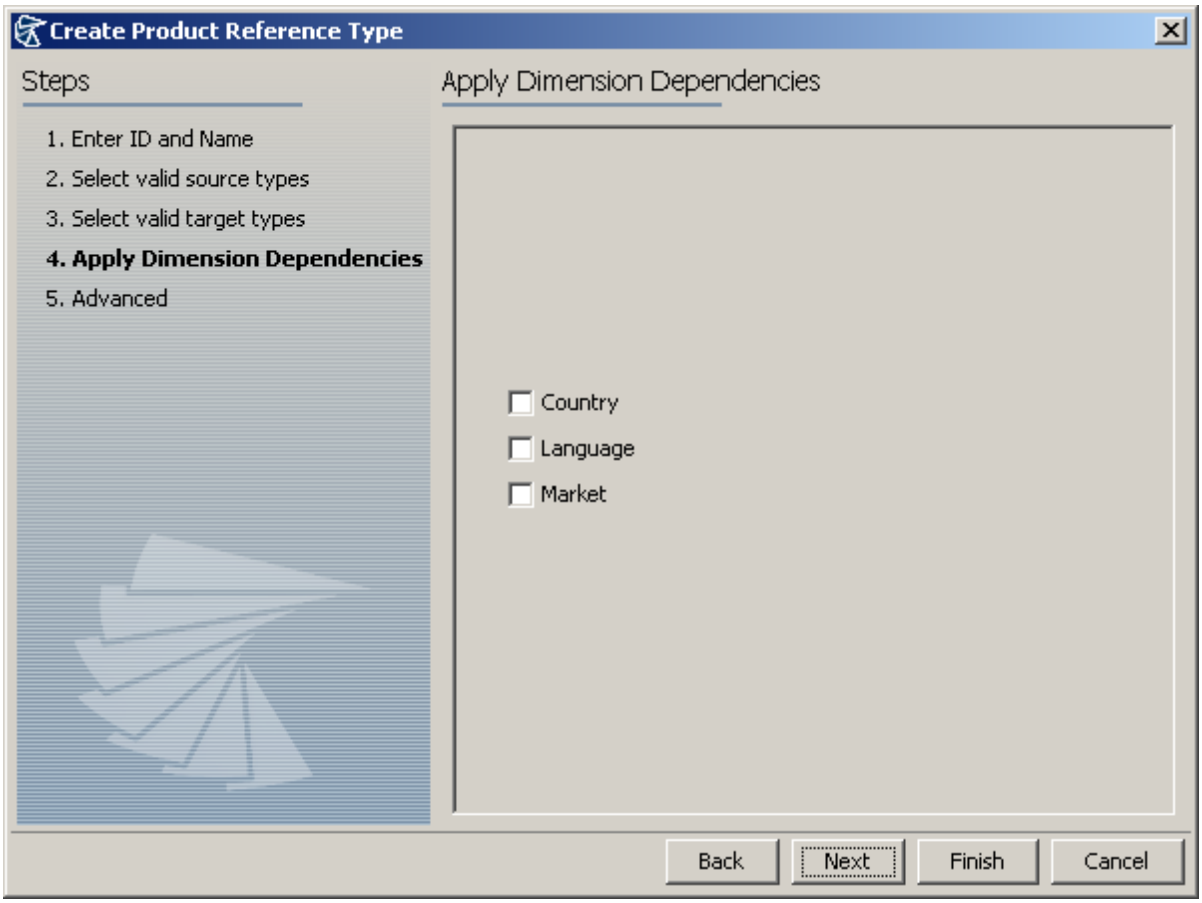
3. Key in **ID** and **Name**, and click **Next**.
4. Select **Valid Source Object Types** for the Reference Type to be created, and click **Next**.



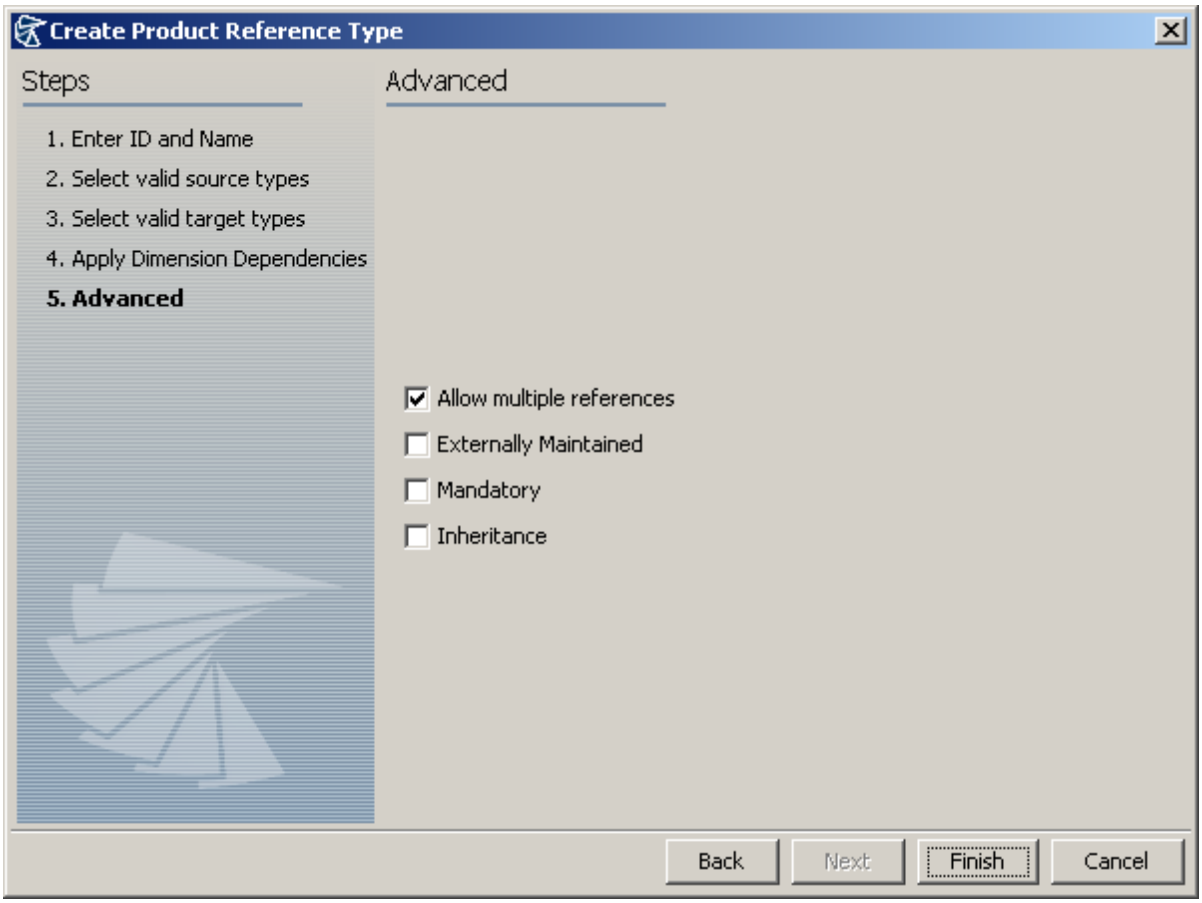
5. Select **Valid Target Object Types** for the Reference type to be created and click **Next**.



6. **Optional**: Click a Dimension, in case the Reference Type should be **Dimension Dependent**, and click **Next**. It is recommended that only one Dimension is applied for one Reference Type.



7. Apply **Advanced Settings** for the Reference Type.



- **Allow Multiple References**:

If checked, the Reference Type can reference more targets from same source.

If not checked, the Reference Type can only reference one target from same source.

- **Externally Maintained**:

If checked, the Reference Type will be **Externally Maintained**. Adding a Reference from a Source Object to a Target Object will not trigger an Unapproved Status.

Note: Changing Values on Attributes applied the Reference Type will trigger an Unapproved Status on the Master Object, if the Metadata Attributes and not Externally Maintained as well.

If Not Checked, the Reference Type will be controlled by the Revision Control. Adding a Reference from a Source Object and/or changing values on Attributes applied the Reference will trigger an Unapproved Status on the Source Object.

- **Mandatory**:

If checked: , the Reference Type is Mandatory to be used from a Source Object to a Target Object. Approval of the Master Object will be prevented, in case the Reference Type is not used.

If not checked:, It is optional to use the Reference Type to reference a Source Object to a Target Object.

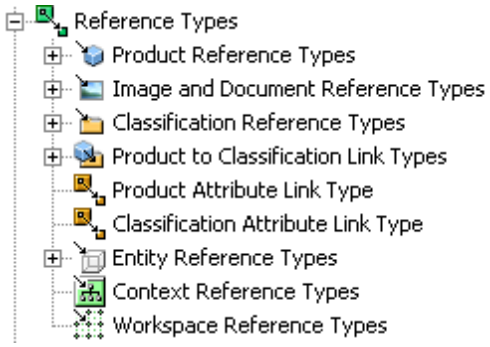
- **Inheritance**

  If checked:, the Reference Type will be inheritable. When creating a Reference from a Source (Master Object) to a Target (Subordinate), the Reference will be inherited on all valid Objects below the Source Object. Each inherited Reference will reference the same Target (Subordinate).

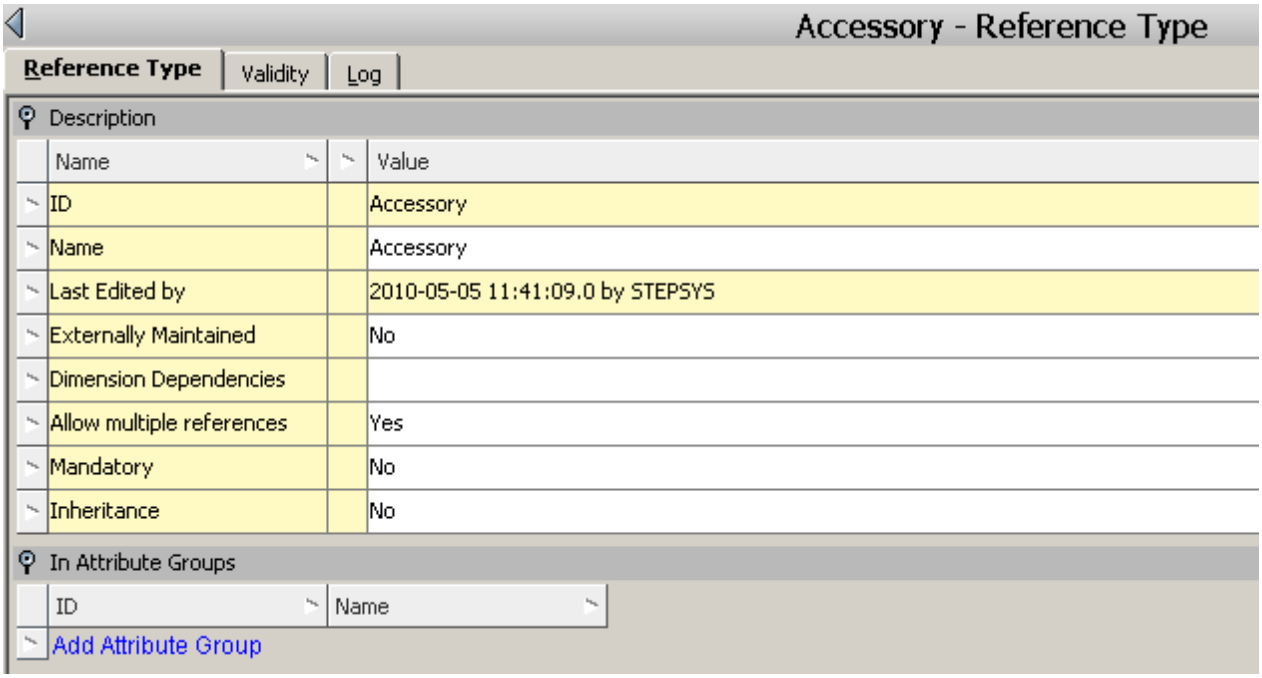
  If not checked:, the Reference Type will not be inheritable. When creating a Reference from a Source (Master Object) to a Target (Subordinate), the Reference will only appear on those two referenced Objects.

Changing Reference Types Settings

1. In **System Setup**, click the **Reference Types** Hierarchy.



2. Select the relevant **Reference Type**



3. Optionally, double-click the **Externally Maintained** field, to specify if **References** of this type should be **Externally Maintained**.

Important: If a Reference Type is changed to be Externally Maintained and the Reference Type has been used to reference Objects, you must select a Workspace. All existing References will be synchronized across all Workspaces based on the selected Workspace.

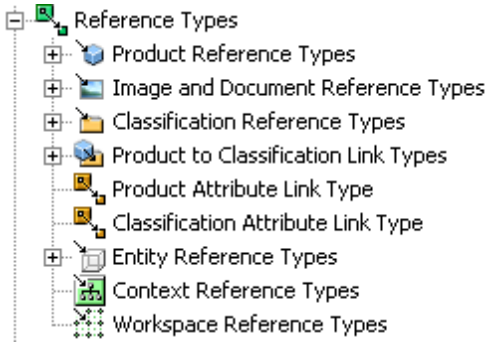
4. Optionally, set the **Dimension Dependency** field to a **Dimension**, if the specific Reference Type should be dimension Dependent. It is recommended to select only one Dimension.
5. Optionally, set the **Allow multiple references** field to **Yes**, if the specific Reference Type should be used to reference more Target Objects from same Source. If **Allow multiple references** is set to **No**, only one Target of this Reference Type can be referenced from same Source.
6. Optionally, set the **Mandatory** field to **Yes**, if the specific Reference Type should be mandatory to be referenced from valid Source Objects. If **Mandatory** field is set to **No**, the Reference Type is optional to be used from a valid Source Object.

Note: If set up, then Approval of Source Objects can be prevented, if Mandatory Reference Types have not been used on Source Objects.

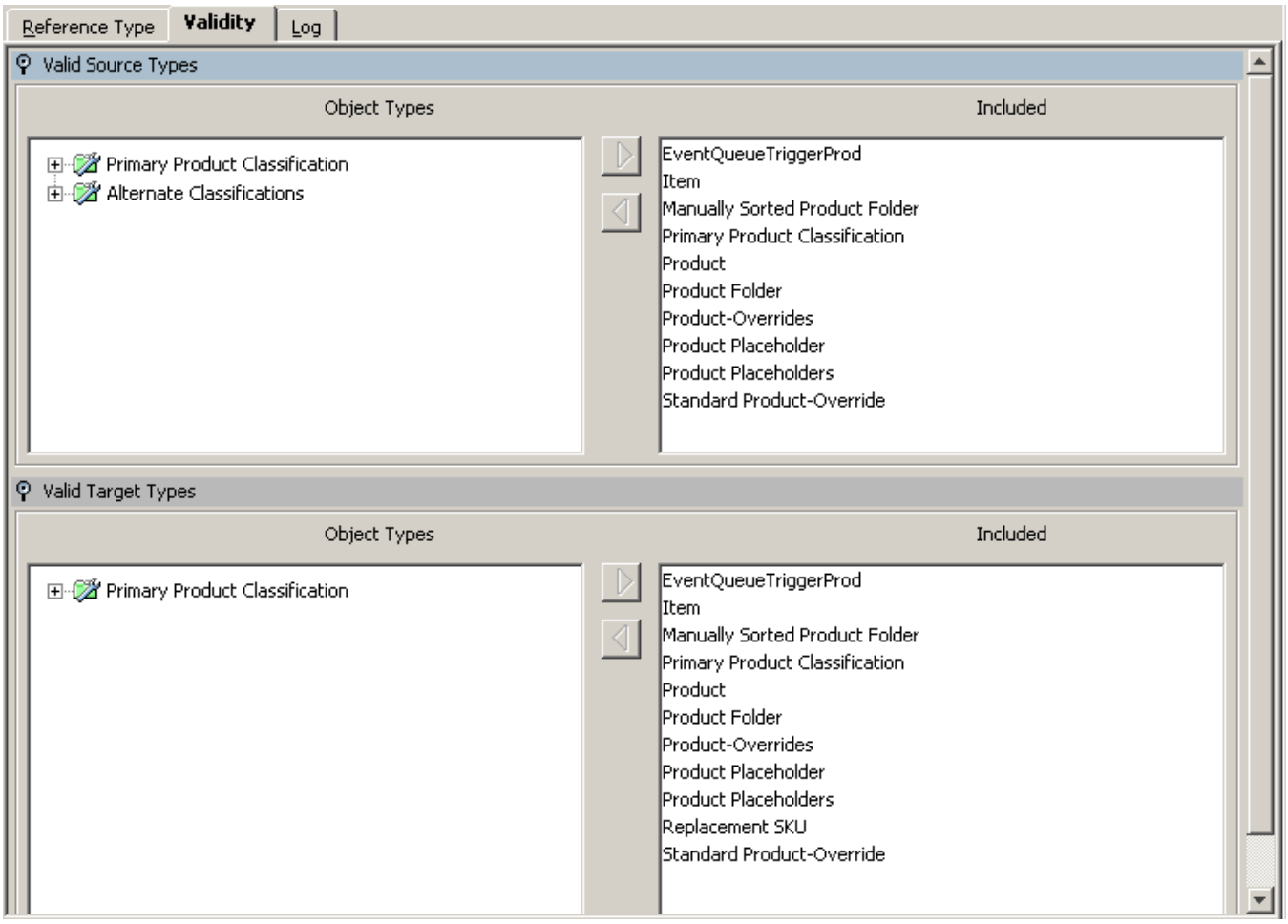
7. Optionally, set the **Inheritance** field to **Yes**, if the specific Reference Type should be inheritable. When creating a Reference from a Source (Master Object) to a Target (Subordinate), the Reference will be inherited on all Objects below the Source Object. Each inherited Reference will reference the same Target. If field is set to **No**, the Reference Type will not be inheritable. When creating a Reference from a Source (Master Object) to a Target (Subordinate), the Reference will only appear on those two referenced Objects.

Changing Valid Source and Target Object Types for a Reference Type

1. In **System Setup**, click the **Reference Types** Hierarchy.



2. Select the relevant **Reference Type**
3. Select the **Validity** tab



4. In the **Valid Source Types** field, use the Add and Remove buttons to specify valid **Source Object Types**.

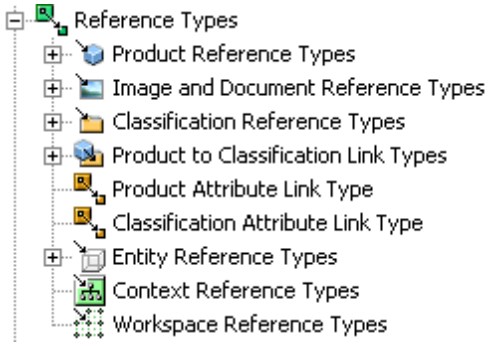
   Important: If an Object Type is removed from the Included Object Types, and References already exist in the database, References will be deleted.

5. In the **Valid Target Types** field use the Add and Remove buttons to specify valid **Target Object Types.**

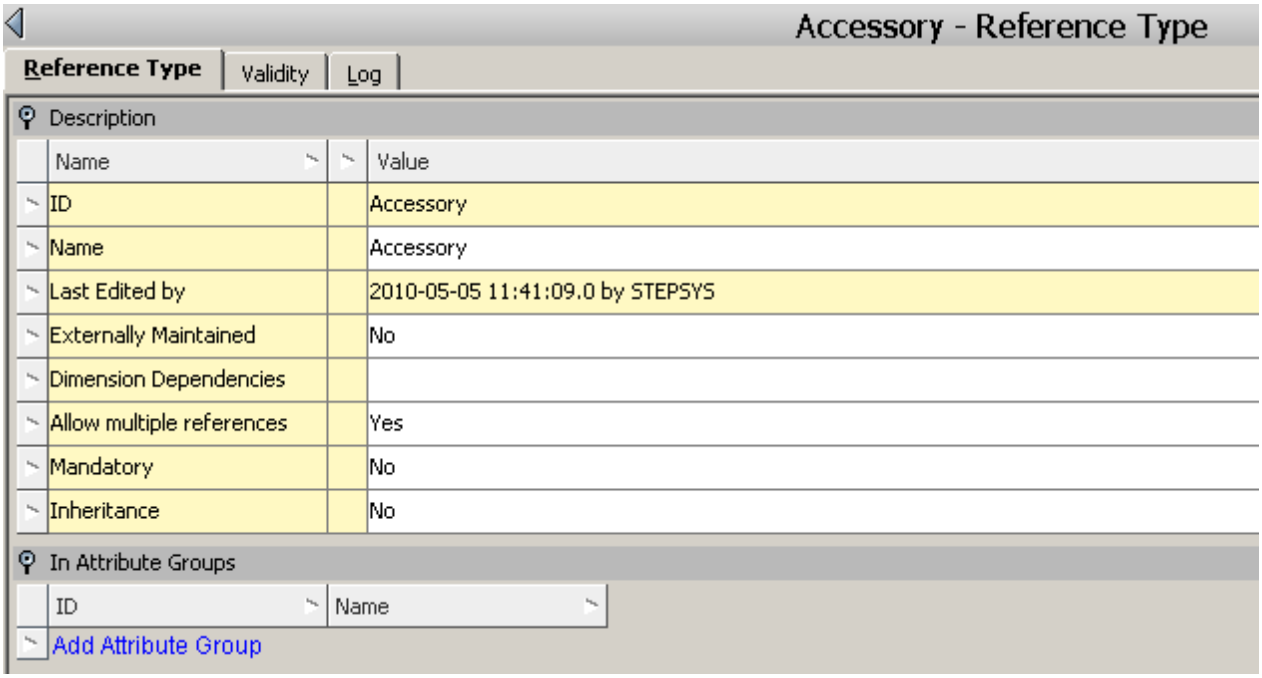
   Important: If an Object Type is removed from Included Object Types, and References to the specific Target already exist with the specific Reference Type, References will be deleted.

Add Reference Types to Attribute Groups

1. In **System Setup**, click the **Reference Types** Hierarchy.



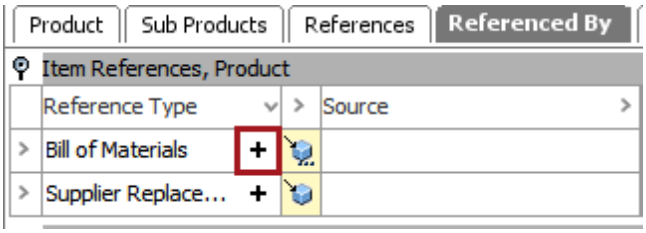
2. Select the relevant **Reference Type**.
3. Click **Add Attribute Group**.



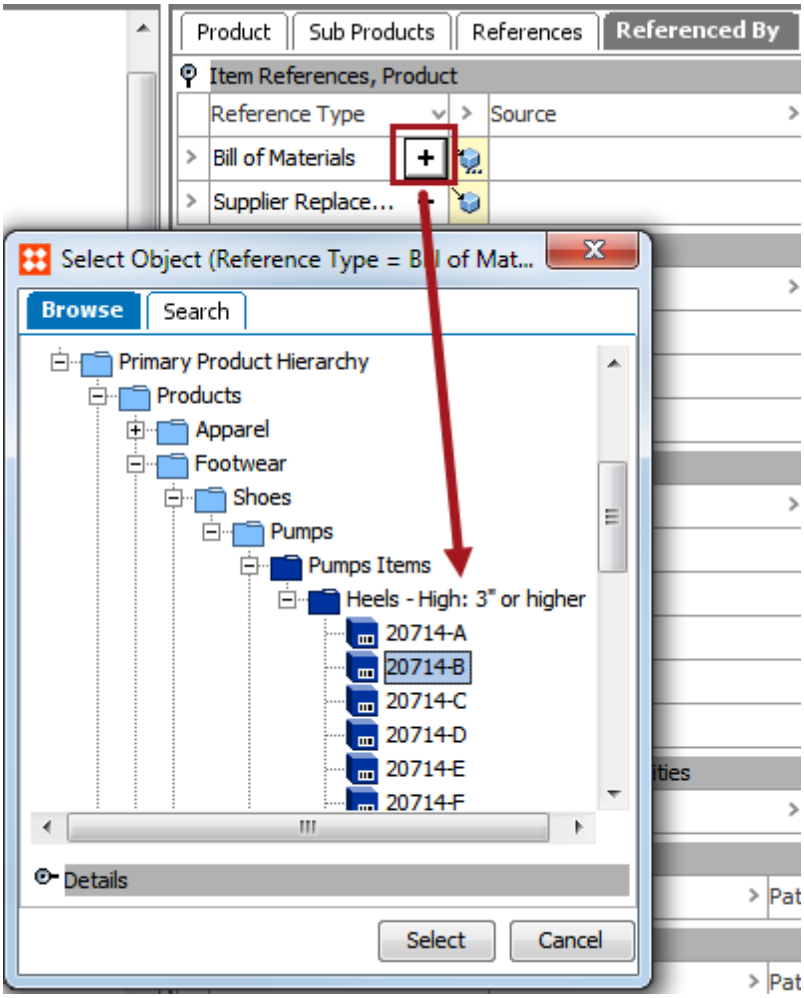
4. Select an Attribute Group or Search for an Attribute Group.
5. Click **Select**. The Reference Type is now added to the Attribute Group.

Adding References on the References and Referenced By Tabs

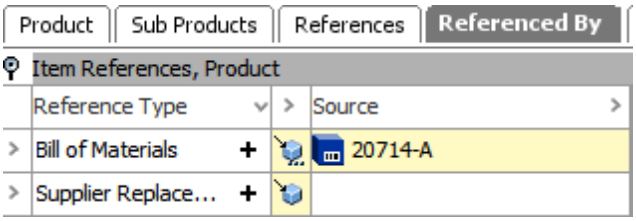
1. To add references to either the **References** or **Referenced By** tabs, click on the plus sign in the desired reference field.



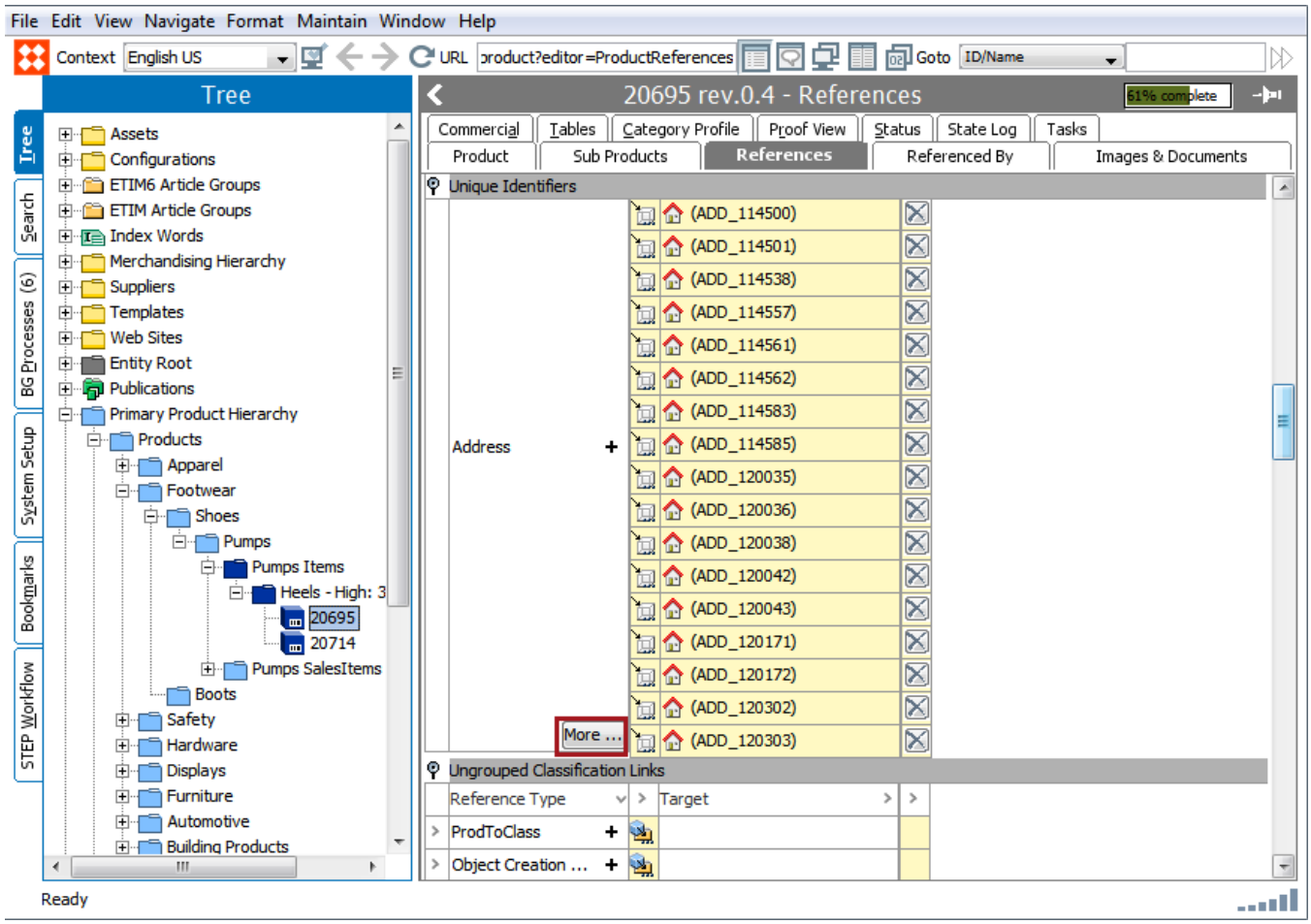
2. Find the reference that is to be added and click Select.



3. Once selected, the object will appear and be added as a reference.



4. If more than 50 references are added, a button entitled 'More...', More ..., displays, allowing users to expand the list and view 50 additional references.



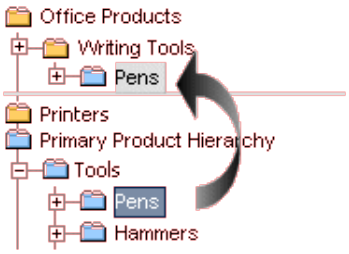
5. Continually clicking the button doubles the number of references shown each time, until the full set is displayed. Note that if new references are added, they display on the list immediately indicating that they were added properly, though normal functionality will resume if the user needs to press the 'More...' button again. This means that they may become hidden until the list is fully expanded.
6. After the list has been fully expanded for the given section, the list will stay expanded until the user refreshes the page or clicks away to another object in workbench.

About Product to Classification Link Types

A Product to Classification Link Type defines a set of rules of how Products can be linked into Classifications.

Icons used for Link Types	Description
	Product to Classification Link Type
	Link is inherited from a higher level in the Tree.
	Link Type is set to be Mandatory. A Link from Product to Classification will have to exist in order to Approve Product or Classification. If a Mandatory Link has not been established, Approval will be prevented. **Important:** The Approval check for Mandatory Product to Classification Links is determined by the ownership of the Product to Classification Link. The Ownership of a Link is setup on the Classification Object Types. For more information, see Object Types and Structures Overview.

For example, a Classification hierarchy could be as illustrated below, where Products like Pens and Pencils should be classified into Writing Tools.



In order to control which Products that are valid to be linked into Writing Tools a Product to Classification Link Type is created.

It defines valid Product Object Types to be linked into the Object Type of Writing Tools.

The Product to Classification Link Type can prevent Products like Hammers to be linked into Writing Tools, but allow Pens to be linked into Writing Tools.

Metadata Attributes

Metadata Attributes can be applied a **Product to Classification Link Type**. It is possible to apply different **Attributes** for different types of Product to Classification Link Types. Metadata Attributes can be maintained from **References** tab on Products.

Metadata Attributes are applied with a Link Type from the Attribute Editor: **Validity Tab**. For more information, see Setting Validity of Description Attributes section of System Setup / Super User Guide documentation.

Multiple Product to Classification Links

Each Link Type has a **Multiple Links** setting available. A Link Type can be setup to allow multiple Links. If Multiple Links Allowed is set to yes, a Product with this link type can be linked into more Classifications. If Multiple Links Allowed is set to no, a Product can only be linked into one Classification. For more information, see Changing Product to Classification Link Types Settings.

Restrict Source and Target Object Types

Each Link Type is setup to allow one or more valid **Source Object Types** and **Target Object Types**. Linking a Product into a Classification will only be allowed for Products and Classifications with valid Object Types. Each Target Object Type can only be used by one Product to Classification Link Type. This means that two or more Product to Classification Link Types cannot include same Target Object Types. For information. see Changing Product to Classification Link Types Settings.

Mandatory Links

A Product to Classification Link Type can be set to be Mandatory. Setting a Link Type to Mandatory is an indication that a link of this type is needed in order to approve a Product or Classification.

Important: The ownership of a Product to Classification Link enforces the check for a Mandatory Product to Classification Link when approving an Object. See Object Types and Structures Overview.

To set a Link Type to be Mandatory, see Changing Product to Classification Link Types Settings.

Externally Maintained Links

A Link Type can be set to be **Externally Maintained**. When using an Externally Maintained link from a Product to a Classification, the link will automatically be included in all Workspaces and the linked objects will remain in their current Approval Status.

Important: The Product or Classification will get an unapproved status if Metadata Attribute values are changed on the Link, and the Metadata Attributes are not Externally Maintained. If the Classification is set to Own Product Links, the Classification will get an unapproved status, otherwise the Product will get an unapproved status.

For more information, see Changing Product to Classification Link Types Settings.

Dimension Dependencies

A Link Type can be setup to be Dependent of a Dimension. When a Link Type is setup to be dependent of a Dimension, then it is typically made dependent of one Dimension only - e.g. the language Dimension. A dimension dependent Link Type is used to specify in which Contexts a Product to Classification link should appear.

Important: It is not possible to make one link valid for multiple Dimension Points. A link will have to be created for each Dimension Point. If the Link Type contains Metadata Attributes, values will have to be keyed in for each Link. Metadata values cannot be shared between two Dimension Points.

For more information, see Changing Product to Classification Link Types Settings.

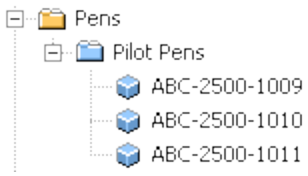
Inheritance of Links, Specification Attributes and Tables

Product to Classification Link Types have three types of inheritance options. A Link Type can be setup to either:

- Not Inherit links, Specification Attributes and Tables from Classification to linked Products
- Inherit links from Classification to Leaf Products of a linked Product Family and not inherit Specification Attributes and Tables from a Classification to any of the linked Products.
- Inherit links, Specification Attributes and Tables from Classification to all linked Products.

For example, to link basic Products into Classifications, a Link Type is created in STEP and named **Basic Products**.

A Product Family named **Pilot Pens** is linked into the Classification **Pens** with **Link Type** Basic Products.



Three inheritance options can be set on the Link Type.

- **No Inheritance**: Links of type **Basic Products** will not be inherited to any of the Products **ABC-2500-1009**, **ABC-2500-1010** and **ABC-2500-1011**.
- **No Inheritance of Tables and Specification Attributes**: Links of Type **Basic Products** will be inherited to all Products below Pilot Pens. Tables and Attributes applied the Pens Classification will not be inherited to any Products.
- **Inheritance of Links, Tables and Specification Attributes**: Links, Specification Attributes and Tables will be inherited from the Classification Pens to all Products below Pilot Pens. In this example:**ABC-2500-1009**, **ABC-2500-1010** and **ABC-2500-1011**.

For more information, see Changing Product to Classification Link Types Settings.

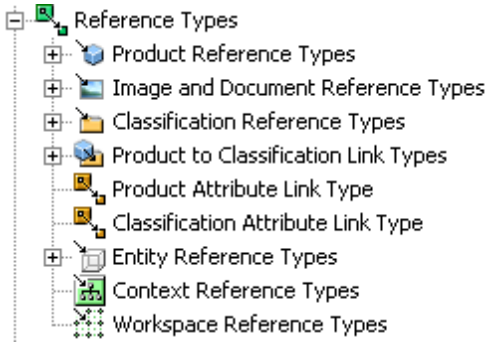
Product Classification Link Types in Attribute Groups

Product to Classification Link Types can be applied **Attribute Groups**.Grouping **Product Classification Link Types** into **Attribute Groups** opens up the ability to setup **User Privileges** where **Approval**of Product to Classification Links is only enabled for specific **Link Types**. Users will only be allowed to Approve **Product to Classification Links** of the **Link Type** linked into the **Attribute Group**. Grouping **Product to Classification Link Types**into **Attribute Groups** also opens up the ability to setup a customized view of all **Link Types** linked into the Attribute Group. The customized view is defined on the Attribute Group and will have effect on all **Link Types** linked into the **Attribute Group**. It defines General Aspects, Metadata Attributes, Attributes on referenced Objects to be shown in the Reference Editor on Products, Classifications and Assets. For more information, see the **Creating a Customized View** section in the **Attribute Groups / System Setup** documentation.

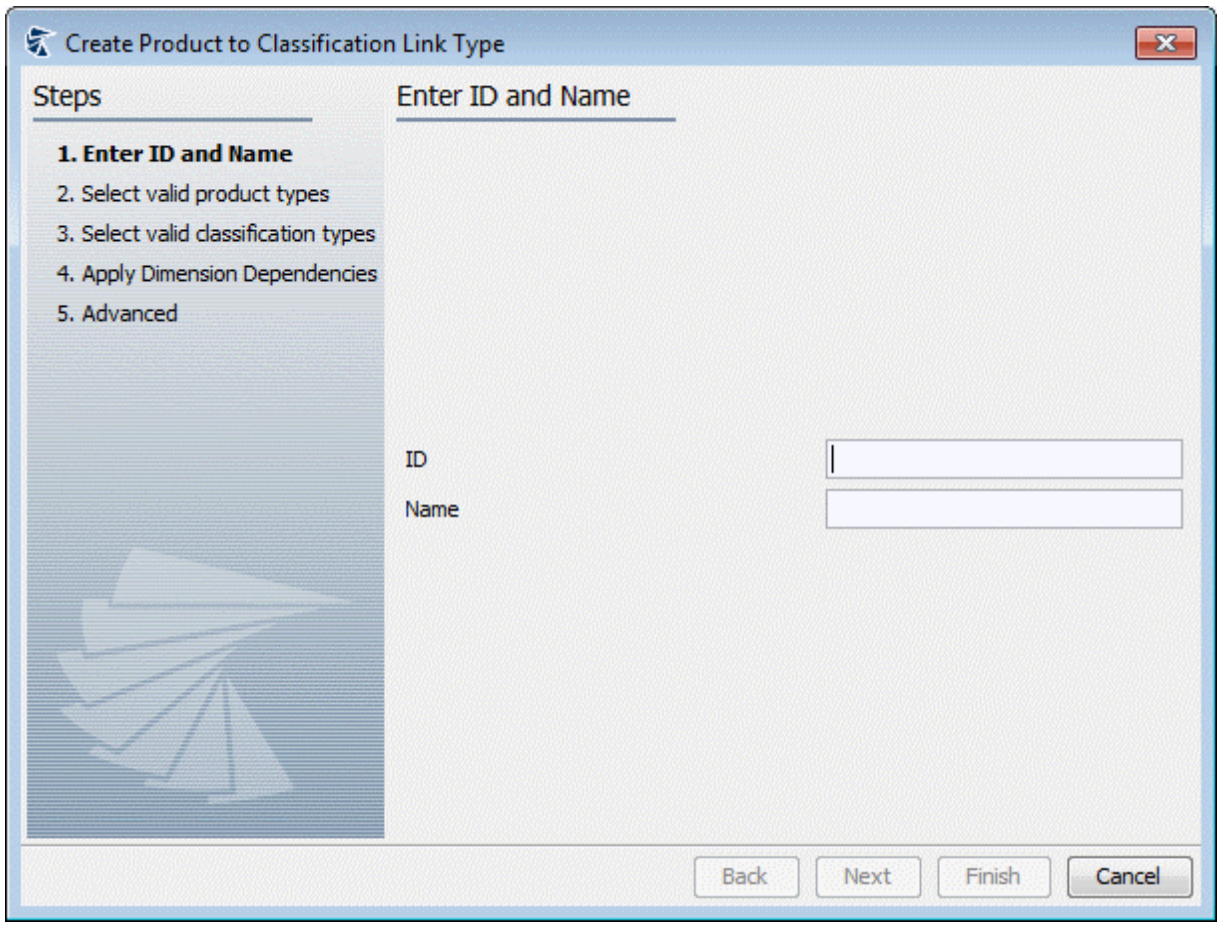
Important: Approving a Product to Classification Link will not only Approve the General Aspects and Metadata Attributes shown by the Customized View, but Approve all General Aspects and Metadata Attributes defined on the Product to Classification Link Type.

Creating Product to Classification Link Types

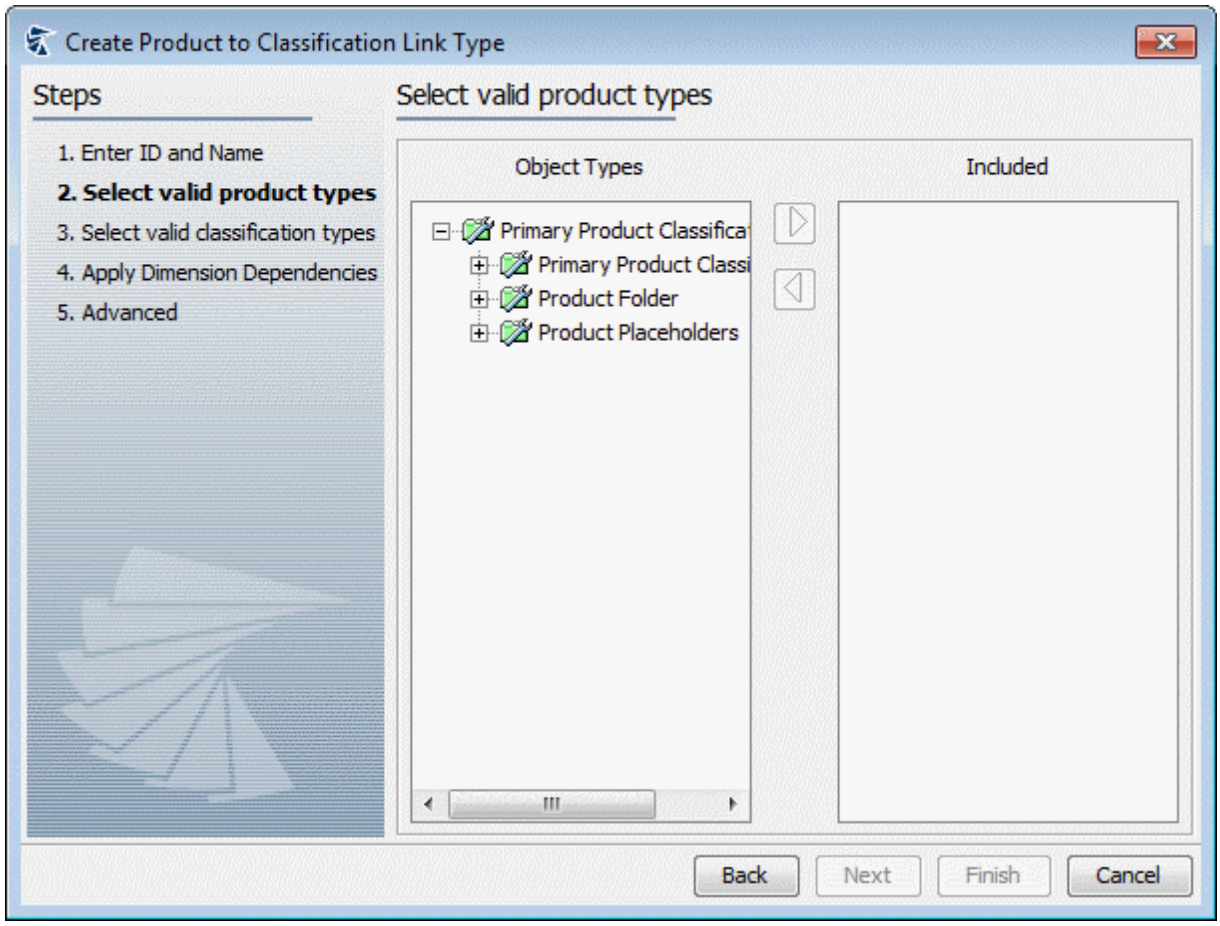
1. In **System Setup**, click the **Reference Types** Hierarchy.



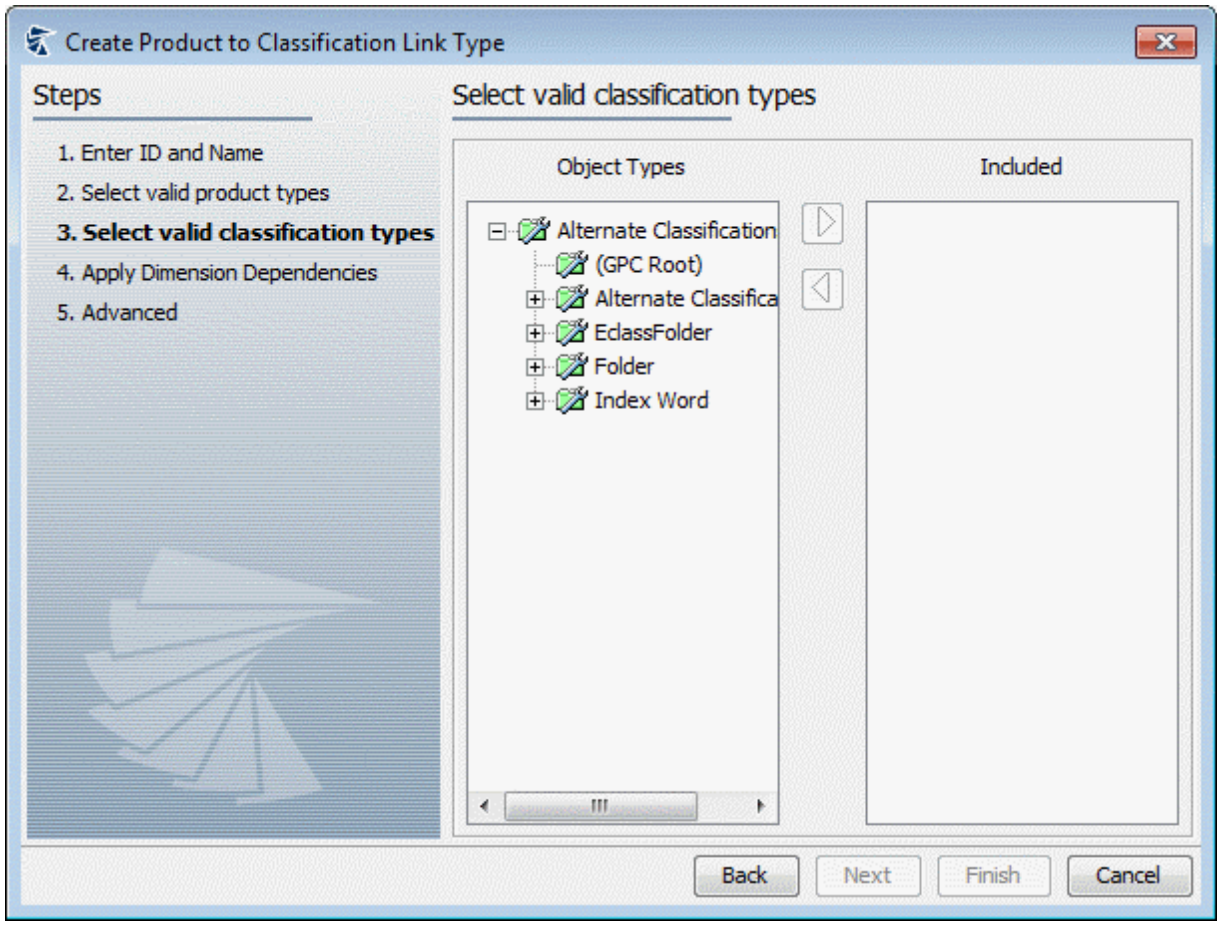
2. Right-Click **Product to Classification Link Types** and click **New Product to Classification Link Type**.



3. Key in **ID** and **Name**, and click **Next**.
4. Select **Valid Source Object Types** for the Link Type to be created, and click **Next**.

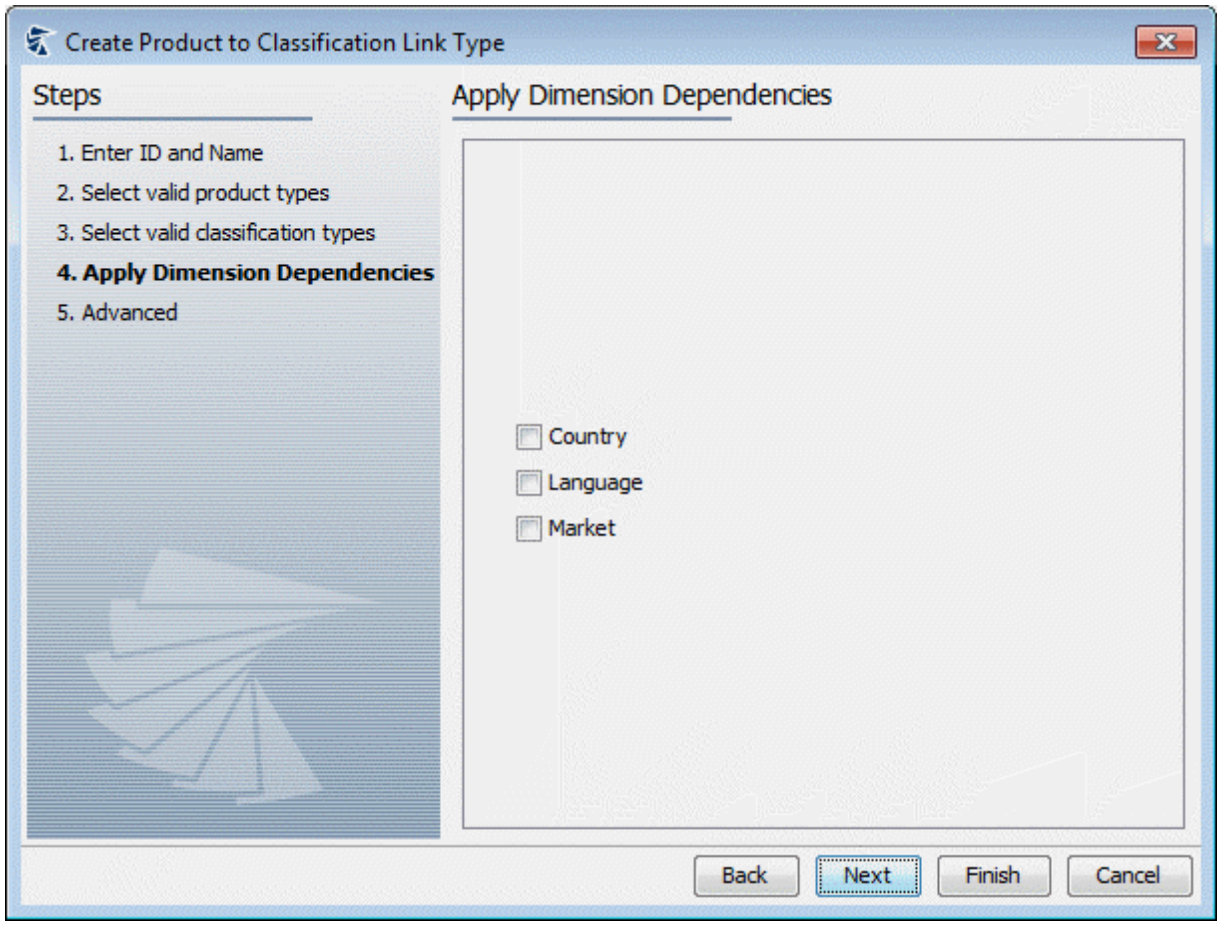


5. Select **Valid Target Object Type** for the Link Type, and click **Next**.



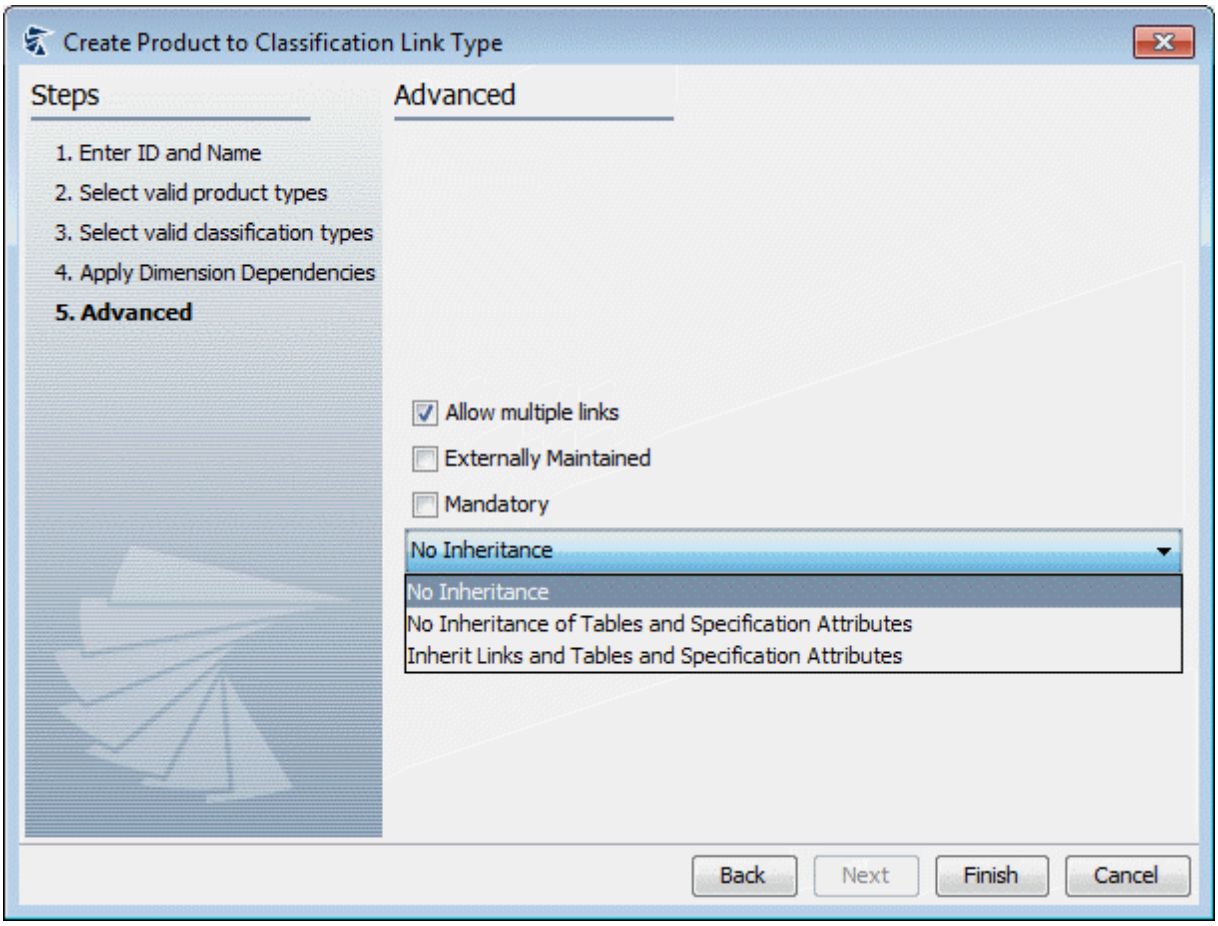
Note: Each Product to Classification Link Type can only include one Classification Object Type.

6. **Optional**: Click a Dimension, in case the Link Type should be **Dimension Dependent**, and click **Next**.



Note: It is recommended that only one Dimension is applied for a Link Type.

7. Apply **Advanced Settings** for the Link Type.



- **Allow Multiple Links**:

If checked, the Link Type can be used to link a Product to more Classifications

If not checked, the Link Type can be used to link a Product to one Classification only. Same Product cannot be linked to more Classifications.

- **Externally Maintained**:

If checked, the Link Type will be **Externally Maintained**. Linking a Product to a Classification will not trigger an Unapproved Status. The Unapproved Status is determined by the ownership of the Product to Classification Link. The Ownership of a Link is setup on the Classification Object Types.

Important: If the Link Type contains Internally Maintained Metadata Attributes, and values are changed on those Attributes, then the Object that owns the Product to Classification Link will trigger an Unapproved Status.

If Not Checked, the Link Type will be controlled by the Revision Control. Linking a Product to a Classification will trigger an Unapproved Status on the Object that owns the Product to Classification Link.

- **Mandatory**:

If checked:A Link with this Type will have to exist in order to Approve the Product or Classification. When Approving a Product or Classification, a warning will appear, if a Link does not exist.

Important: The Approval check for Mandatory Product to Classification Links is determined by the ownership of the Product to Classification Link. The Ownership of a Link is setup on the Classification Object Types. See Object Types and Structures Overview.

If not checked: It is optional to use the Link Type to Link a Product to Classification. Approving a Product or Classification will not trigger a warning, in case the Link Type is not used.

- **SubProducts Inheritance Settings**

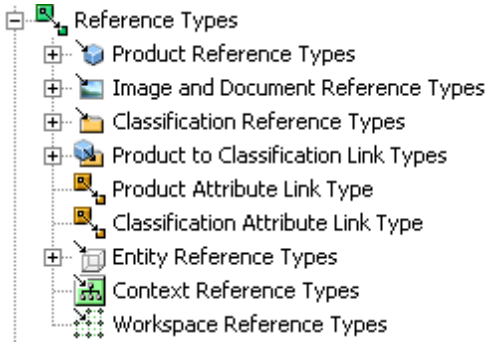
No Inheritance: Link from Product Family to Classification will not be inherited to Leaf Nodes below linked Product Family. Specification Attributes and Table Definitions applied on Classification will not be inherited to any Product below the Classification.

No Inheritance of Tables and Specification Attributes: Link from Product Family to Classification will be inherited to all Leaf Nodes below the Product Family. Specification Attributes and Table Definitions applied the Classification will not be inherited to any Products below the Classification.

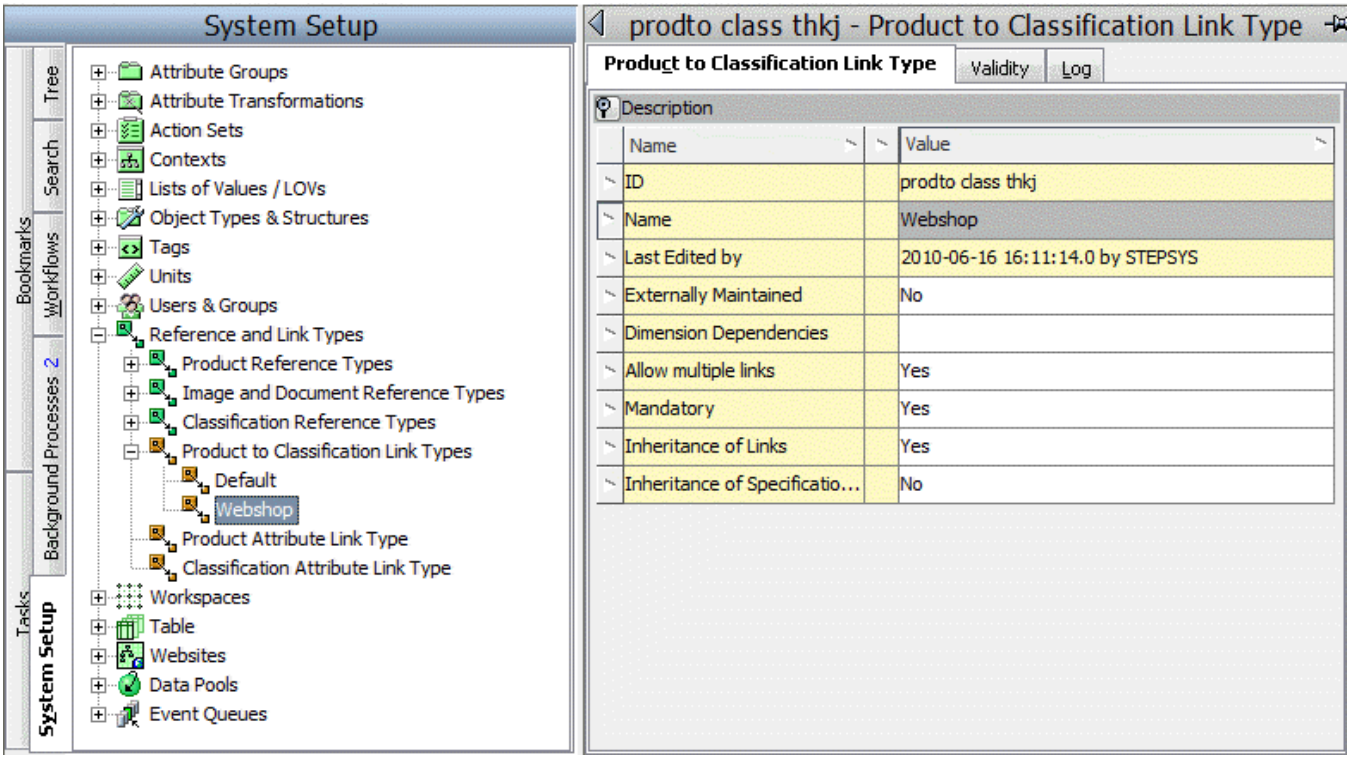
Inherit Links, Tables and Specification Attributes: Link from Product Family to Classification will be inherited to all Products below the Product Family. Specification Attributes and Tables applied the Classification will be inherited to all Products below the Classification.

Changing Product to Classification Link Types Settings

1. In **System Setup**, click the **Reference Types** Hierarchy.



2. Select relevant **Product to Classification Link Type**



3. Optionally, double-click the **Externally Maintained** field, to specify if **Product to Classification Links** of this type should be **Externally Maintained**.

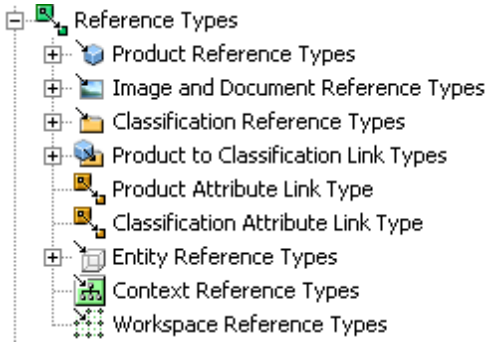
Important: If a Link Type is changed to be Externally Maintained and the Link Type has been used to link Products to Classifications, a select Workspace dialog will appear. All existing Product to Classification Links will be synchronized across all Workspaces based on the selected Workspace.

---

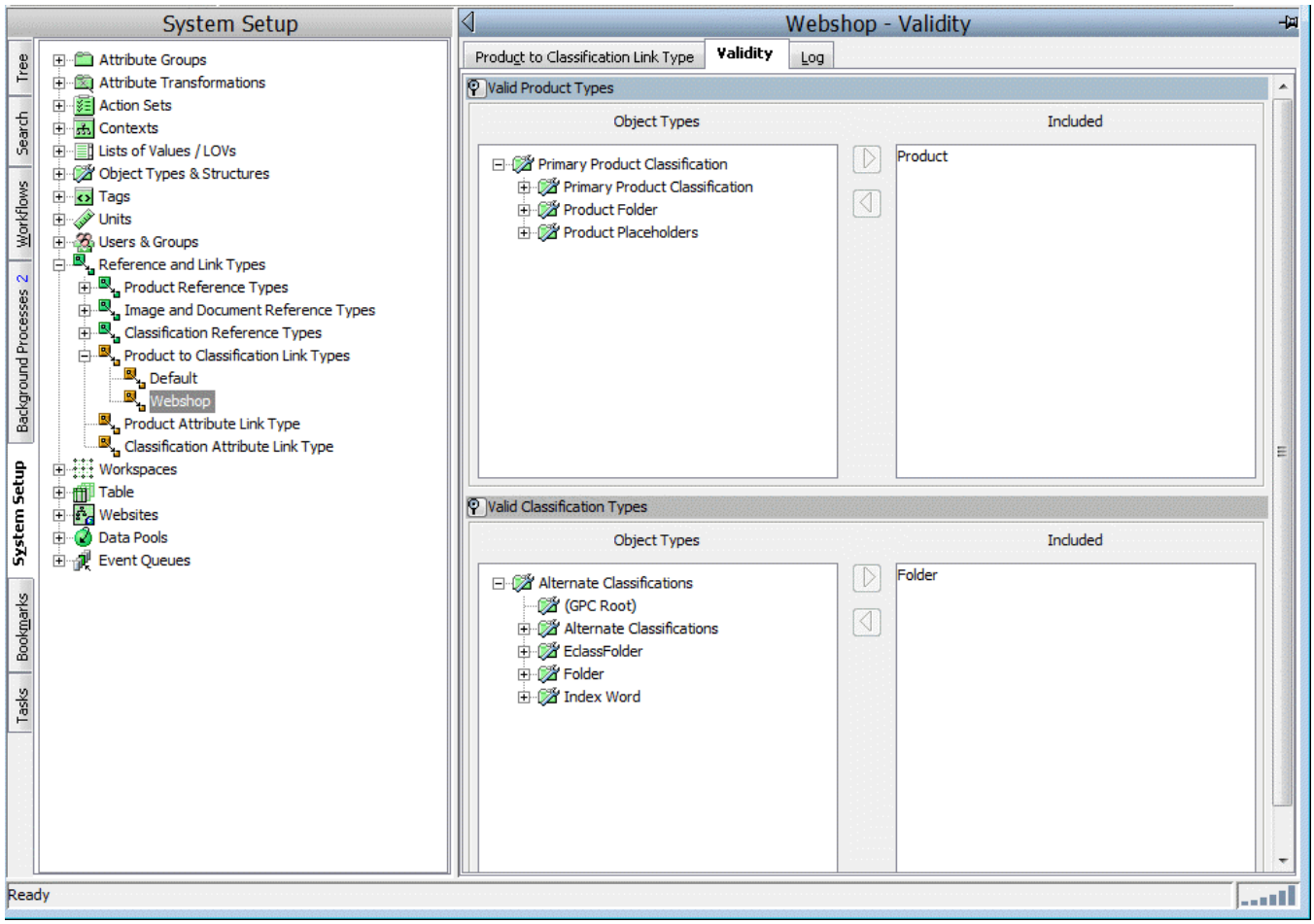
4. Optionally, set the **Dimension Dependency** field to a **Dimension**, if the Link Type should be Dimension Dependent. It is to be recommended to only select one Dimension.
5. Optionally, set the **Allow Multiple links** field to **Yes**, if the Link Type should be used to reference one Product to more Classifications. If **Allow multiple links** is set to **No**, a Product can be linked to one Classification only.
6. Optionally, set the **Mandatory** field to **Yes**, if a Product to Classification link of this type is required in order to Approve the Product or Classification. Approving an Object without a Mandatory link will be prevented. If **Mandatory** field is set to **No**, the Link Type is optional to be used.
7. Optionally, set the **Inheritance of Links** field to **Yes**, if the Links of this type should be inherited to all Leaf Nodes below a Linked Product Family. If field is set to **No**, the Links of this Type will not be inherited to any Products below a linked Product Family.
8. Optionally, set the **Inheritance of Specification Attributes and Tables** field to **Yes**, if Specification Attributes and Tables applied the Classification should be inherited to all leaf nodes below a Linked Product Family. If field is set to **No**, Specification Attributes and Tables applied a Classification will not be inherited to any Products below the Classification.

Changing Valid Source and Target Object Type for a Product to Classification Link Type

1. In **System Setup**, click the **Reference Types** Hierarchy.



2. Select the relevant **Product to Classification Link Type**
3. Select the **Validity** tab



4. In the **Valid Source Types** field

Use the Add and Remove buttons to specify valid **Source Object Types**

Important: If an Object Type is removed from the Included Object Types, and Product to Classification Links with this type already exist, Links will be deleted.

---

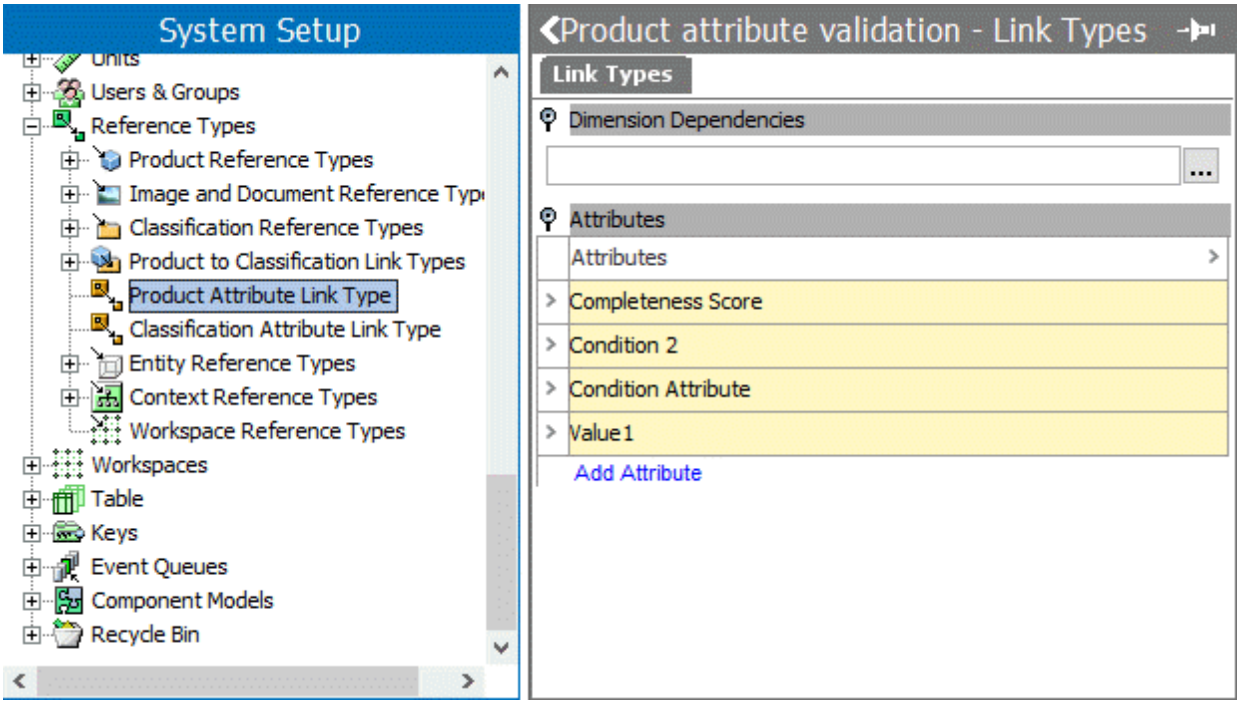
5. In the **Valid Target Type** field

   Use the Add and Remove buttons to specify valid **Target Object Type**. Each Product to Classification Link Type can only include one Target Object Type.

About Product to Attribute Links and Classification to Attribute Links

- The **Product to Attribute Link Type** defines the link between a Product and a Specification Attribute. To setup Product to Attribute Links, click **Product Attribute Links** in the **Reference Types**hierarchy.
- The **Classification to Attribute Link Type** defines the link between a Classification and a Specification Attribute. To setup Classification to Attribute Links, click **Classification Attribute Links** in the Reference Types hierarchy.

Only one **Product to Attribute Link Type** and one **Classification to Attribute Link Type** is available in STEP. Each Link Type is used on all Product to Attribute Links in the database and Classification to Attribute Links in the database. Users cannot create their own Product to Attribute Link Type or Classification to Attribute Link Type.



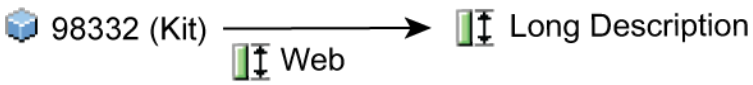
Dimension Dependency for Product to Attribute Links or Classification to Attribute Links

The **Product to Attribute Link Type** and **Classification to Attribute Link Type** can be made dependent of a Dimension. Setting e.g. the Product to Attribute Link Type to be dependent of a Dimension will allow variations of Product to Attribute Links across different Dimension Points or Contexts.

Metadata Attributes

Description Attributes can be applied the **Product to Attribute Link**.

For example, Specification Attributes are linked to Product 98332 (KIT). On each Product to Attribute Link a Metadata Attribute should be applied to specify if the Specification Attribute should be used on Web.



- A **Description Attribute** named **Web**has been applied the Product to Attribute Link Type. This Attribute holds the values yes / no to specify if the Attribute should be displayed on the Web. The Attribute Web will be available on all Product to Attribute Links in STEP.

Setting Dimension Dependency on Product to Attribute Link Type or Classification to Attribute Link Type

1. In **System Setup**, click **Reference Types Hierarchy**
2. Click **Product Attribute Link Type** or **Classification Attribute Link Type**
3. In Dimension Dependency field, click the ellipsis button **(…)**.
4. Click relevant **Dimension**
5. Click **Apply**

Applying Metadata Attributes to Product to Attribute Links or Classification to Attribute Links

1. In **System Setup**, click the **Reference Types Hierarchy**
2. Click **Product Attribute Link Type** or **Classification Attribute Link Type**
3. In Attribute field, Click **[Add Attribute]**
4. Select relevant Attribute by searching or browsing
5. Click **Select**

About Reference Visibility

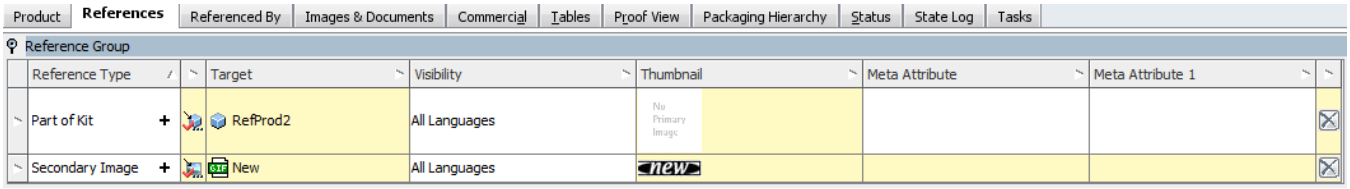
On Dimension dependent references, you can specify the visibility of a Reference in a specific Context.

- You can suppress the visibility of a Reference if you do not want it to be used in a specific Context.
- You can specify whether a Reference is visible in a local or a global Context. If the visibility of a Reference is local, the Reference does not inherit its settings from the Global context. You can therefore apply local attributes, which is useful, for example, if you want to define attributes on a language-by-language basis.

Editing Meta-attributes on References

References can have a number of Meta-attributes applied to them, and you can edit the value of the Meta-attributes.

1. On the **References** tab, double-click the relevant field. You can only edit white fields.Depending on the type of Meta-attribute, you can then edit the value directly in the field, or in a **Value Editor**.
2. If a **Value Editor** opens, click **Add Value**.
3. Enter a value, and then click **OK**. If you want to add more than one value, click **Add Value** again, and add another value. You can change the order in which the values appear by using the **Up** and **Down** buttons.



Meta-attribute values are valid globally. However, when a Reference is only visible in a local Context, it is possible to have a Description value that is only visible in that Context as well.

- You can view all local and global Description values of a Reference on the **View All** tab of the **Adjust Visibility** window.

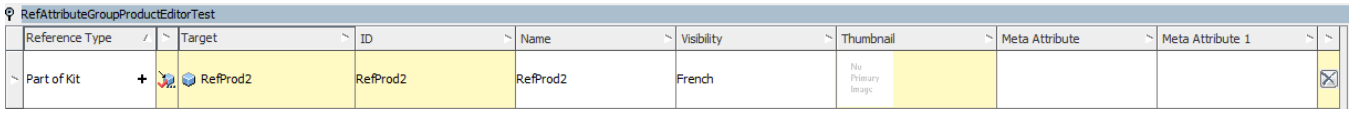
The number of available Meta-attributes on a Reference vary depending on how the View of the Reference Type has been set up. For more information on Views, see Defining a Customized View of Reference Types and Product to Classification Links.

To Specify Global Visibility for a Reference

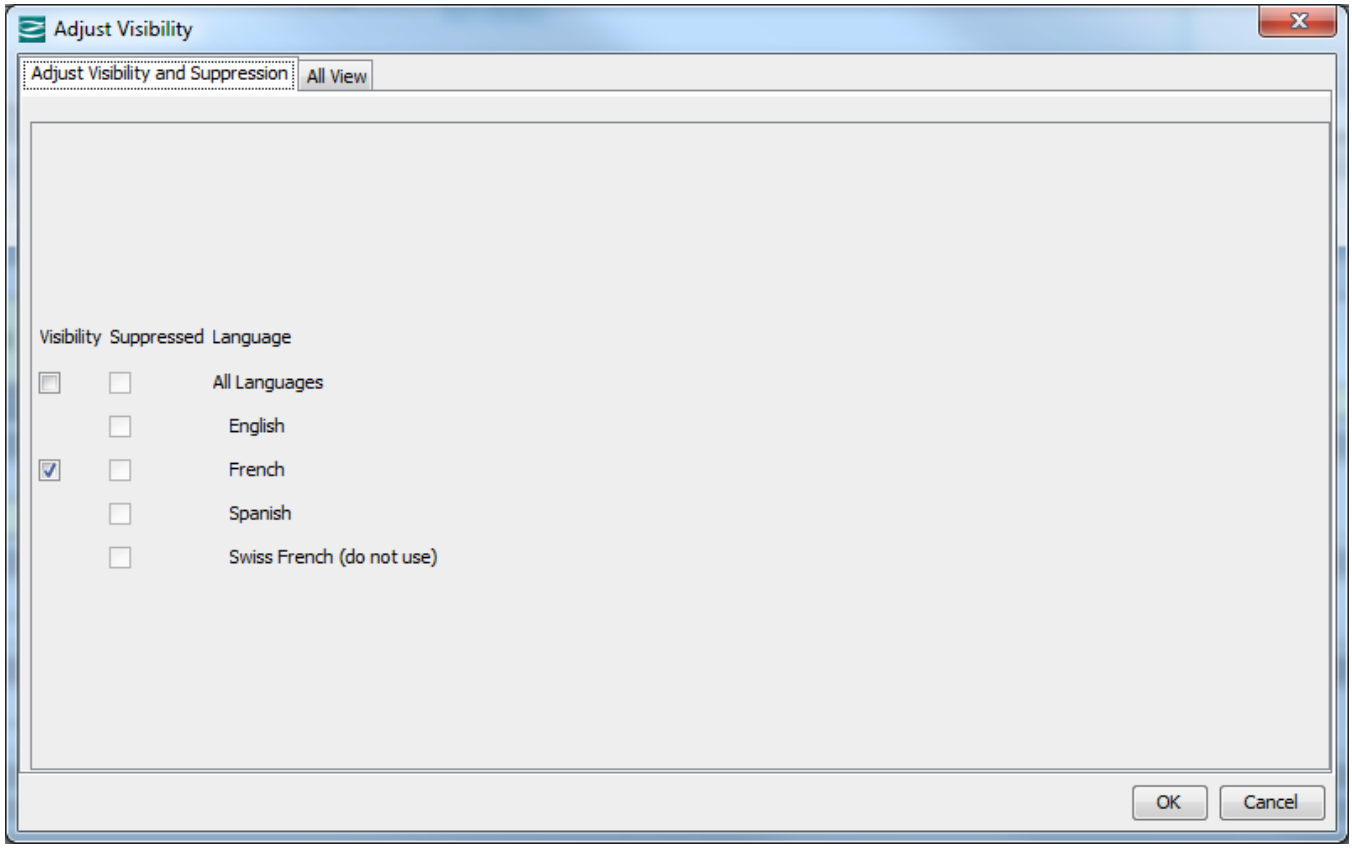
If a Reference is created in a local Context, it is visible in the local Context and no other Contexts.

However, you can specify that you want a Reference to be visible in a Global Context. In other Contexts, the Reference inherits its settings from the Global context settings.

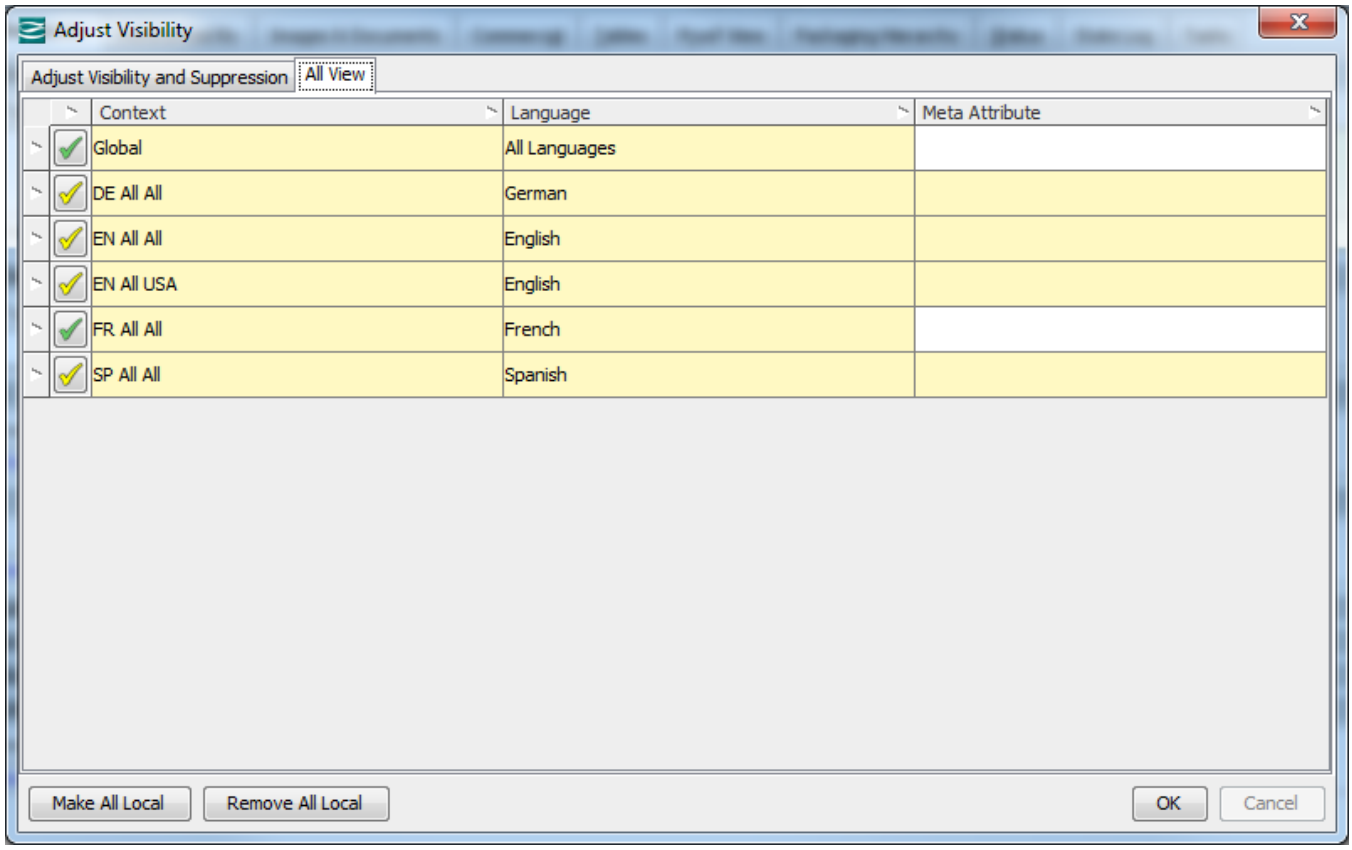
1. In the **Product Hierarchy**, select the relevant product, and then click the **References** tab.



2. Click the **Visibility** field of the Reference whose visibility you want to change, and then click the ellipsis button **(…)**. The **Adjust Visibility** dialog appears.



3. Click the **All View** tab. A green check mark is displayed to the left of the Global context field and yellow checkmarks are displayed to the left of Contexts that inherit the settings of the Reference Type from the Global settings.



4. To make the visibility of a Reference global, click the green **check mark** next to the relevant Context. The check mark turns yellow.
5. To make the visibility of the Reference global for all Contexts, click the **Remove All Local** button. A message is displayed that asks if you want to make references global. Click the **Remove All Local**button. All check marks turn yellow with the exception of the icon next to the Global context.

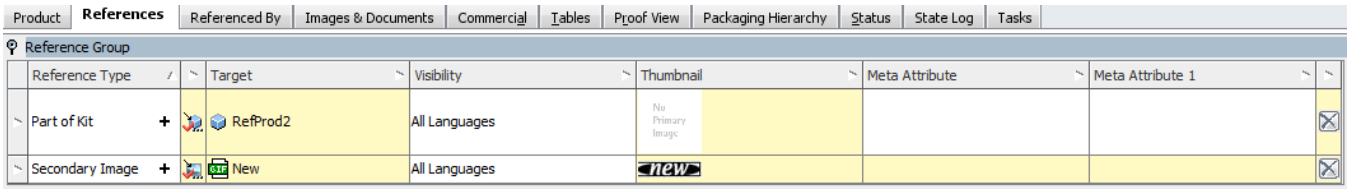
When the visibility of a Reference is set to global it inherits its settings from the Global Context.

To Specify Local Visibility for a Reference

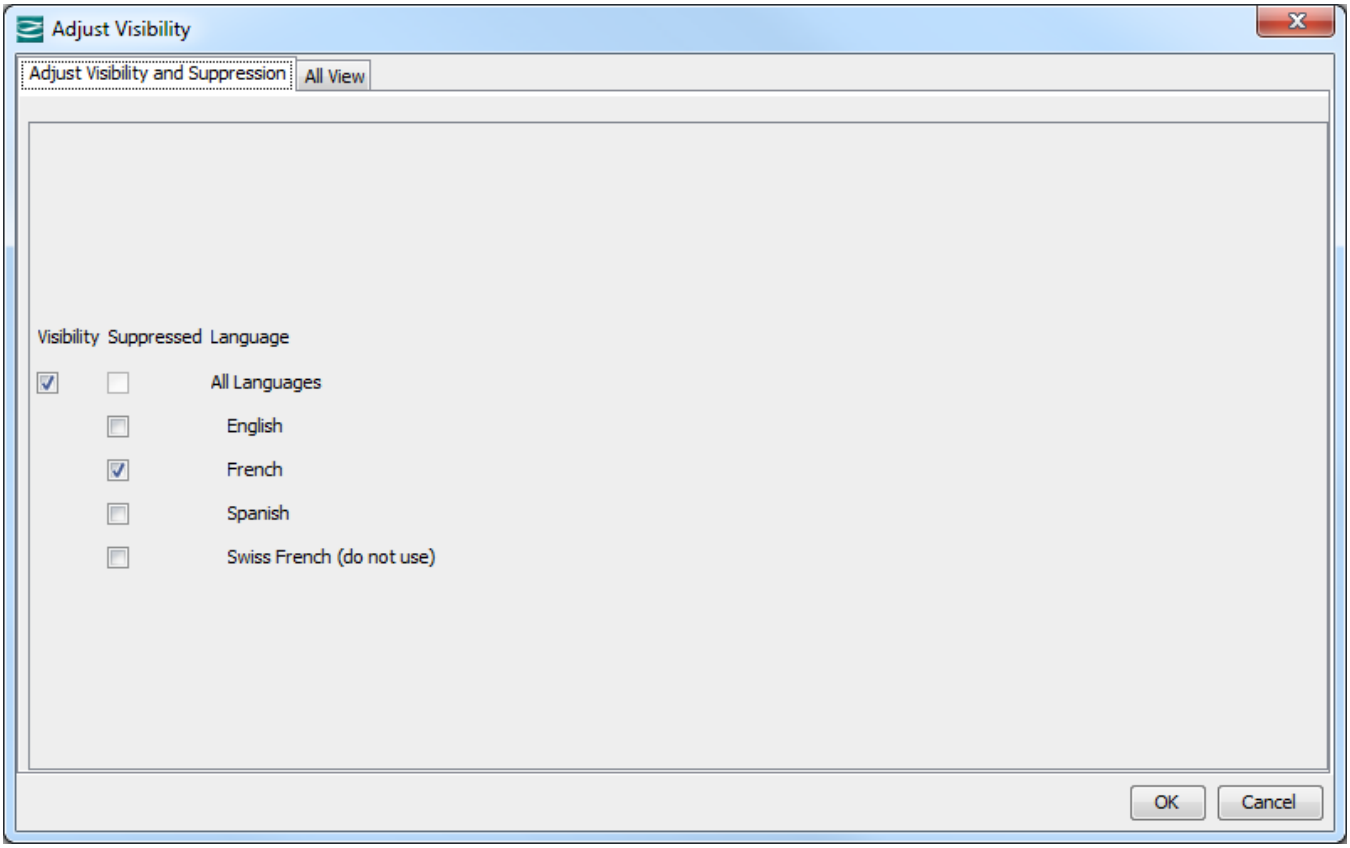
If a Reference is created in a Global Context, it is visible in a Global Context and in all other Contexts. In other Contexts, the Reference inherits its settings from the Global context settings.

However, you can specify that you only want a Reference to be visible in a local Context.

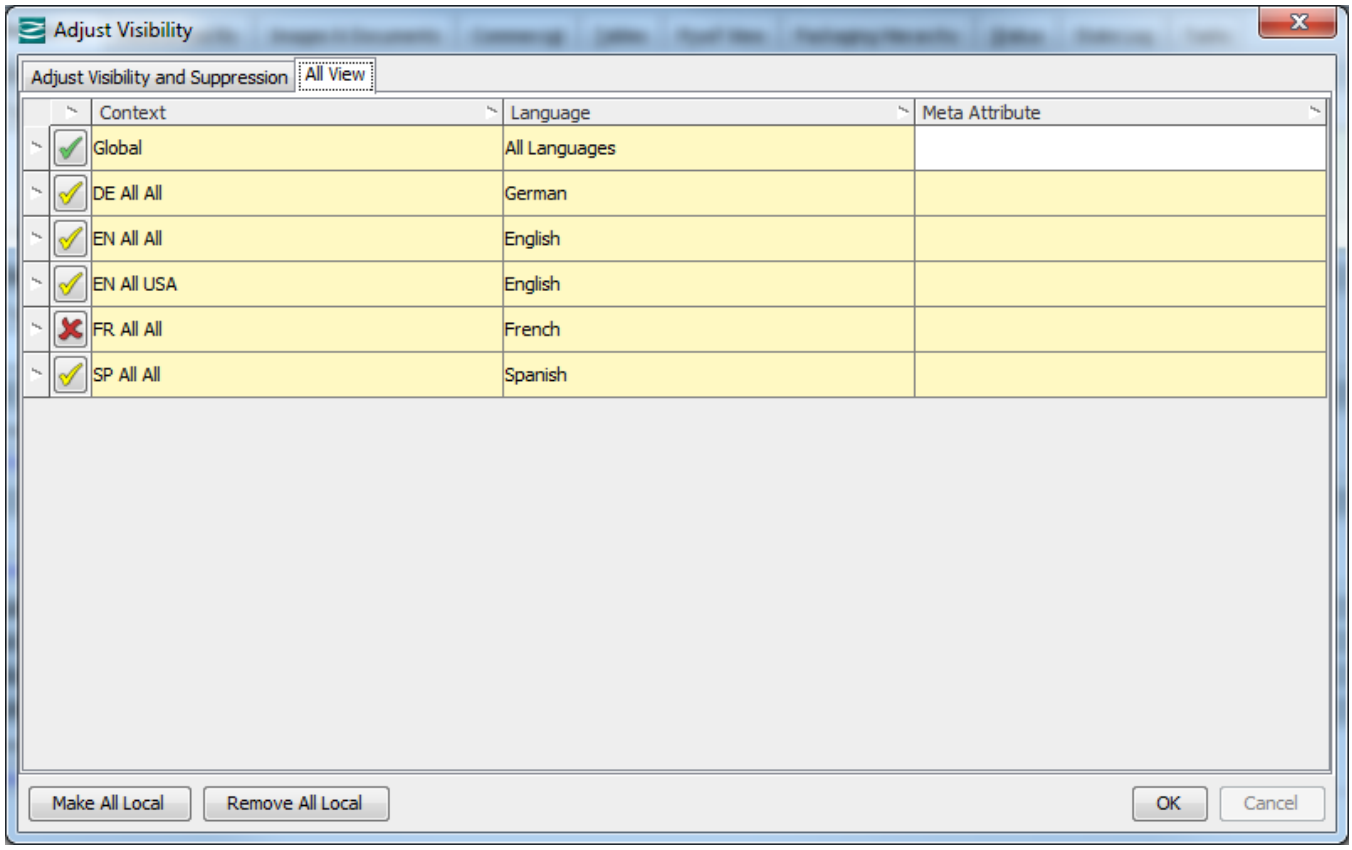
1. In the **Product Hierarchy**, select the relevant product, and then click the **References** tab.



2. Click the **Visibility** field of the Reference whose visibility you want to change, and then click the ellipsis button **(…)**. The **Adjust Visibility** dialog appears.



3. Click the **All View** tab. A green check mark is displayed to the left of the Global context field and yellow checkmarks are displayed to the left of Contexts that inherit the settings of the Reference Type from the Global settings.



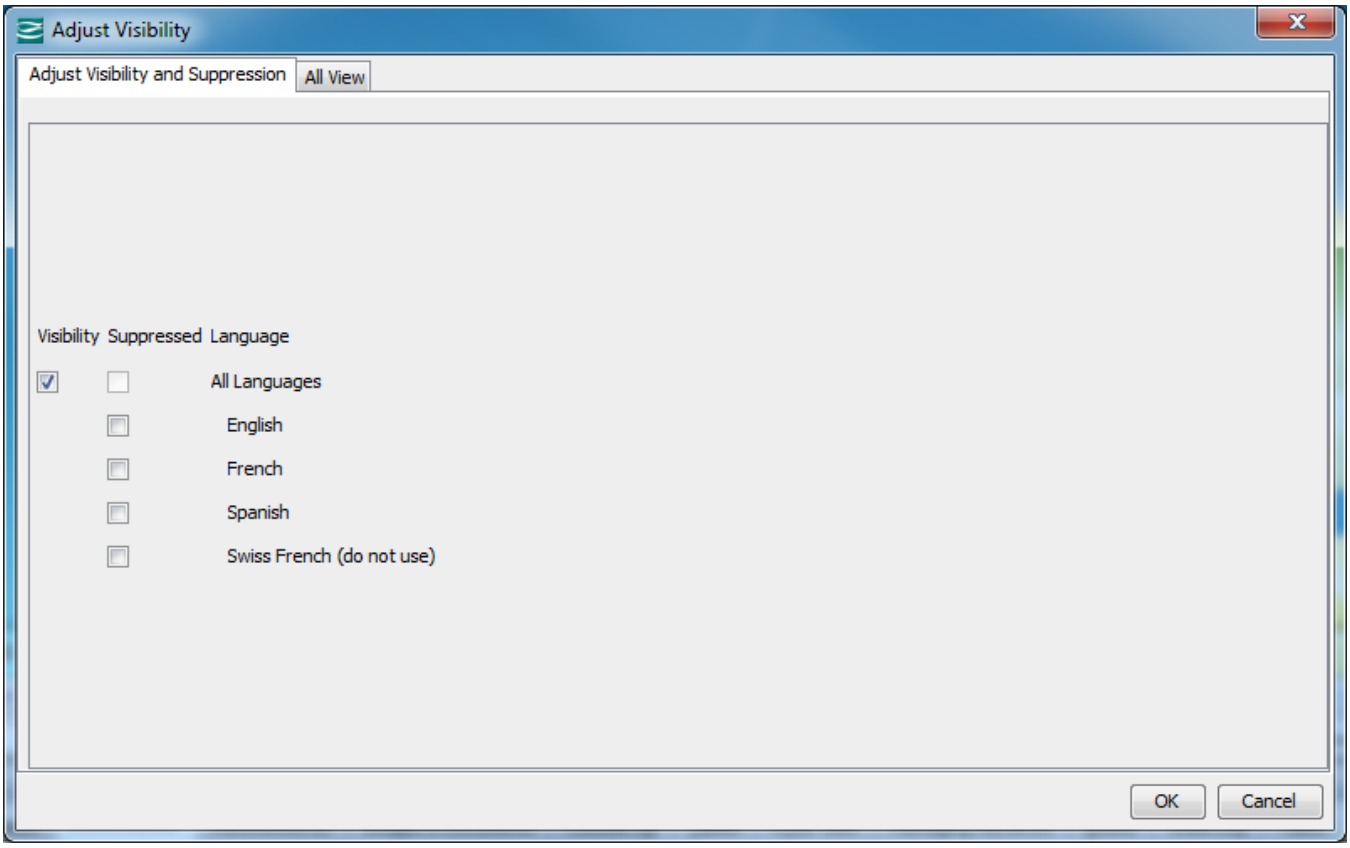
4. To make the visibility of a Reference local, click the yellow **check mark** next to the relevant Context. The check mark turns green.
5. To make the visibility of the Reference local for all Contexts, click the **Make All Local** button. A message is displayed that asks if you want to make references local. Click the **Make Local** button. All check marks turn green with the exception of the icon next to the Global context, which is replaced by a red X indicating that the Reference no longer appears in a Global context.

Note: A red cross indicates that a reference is suppressed or does not exist in the Context. In the sample above the reference is suppressed in FR All All.

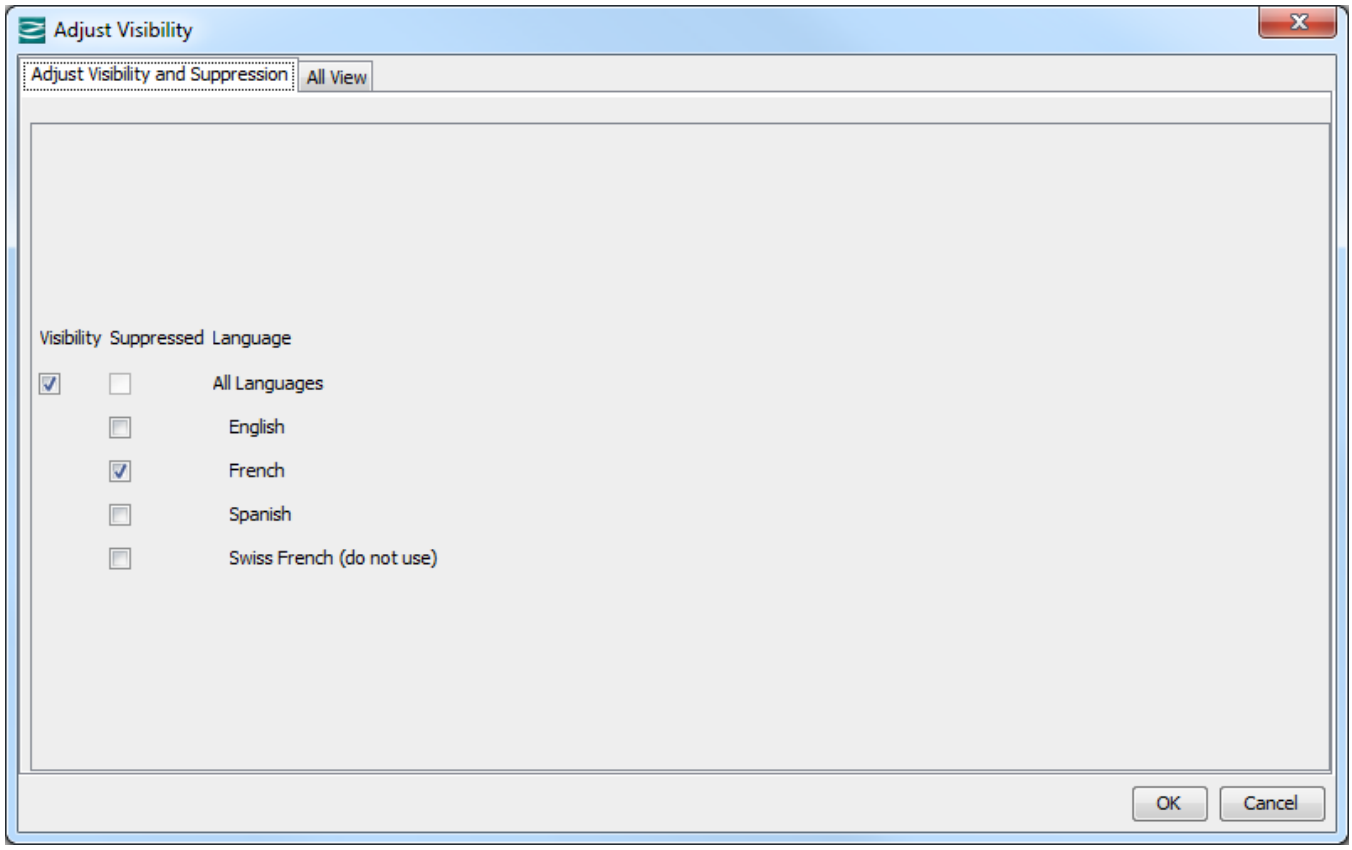
When the visibility of a Reference is set to local it no longer inherits is settings from the Global settings, and you can specify, for example, attribute values that only apply in the local Context.

To Suppress a Reference

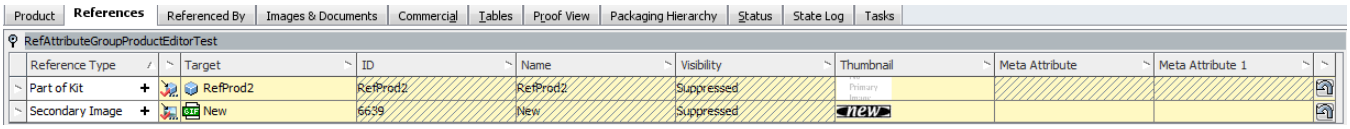
1. In the **Product Hierarchy**, select the relevant product, and then click the **References** tab.
2. Click the **Visibility** field of the Reference whose visibility you want to change, and then click the ellipsis button **(…)**. The **Adjust Visibility** dialog appears.



3. On the **Adjust Visibility and Suppression** tab, specify if you want to suppress the visibility for one or more of the available options. The available options vary depending on the Dimension that is applied. To suppress the visibility of one or more of the options, select the relevant options in the **Suppressed** column.



When visibility has been suppressed for a specific Context, the Description and the Visibility fields are grayed out, and the Visibility field displays the value Suppressed.



- If you want to change the visibility of a Reference from Suppressed to Visible, click the Unsuppress Link icon to the right of the Visibility field.

In all Contexts where the Reference has not been suppressed, the Visibility column will contain a default value that varies depending on the Dimension Type. In this case, if you want to know whether the Reference has been suppressed in other Contexts, you have to click the (…) ellipses point.

Setup Groups

Setup Groups are special folders, which need to be configured to allow holding Integration Endpoints, Business Rules, Portal Configurations and/or Workflow Configurations.

Note: When setting up privileges on a User Group, it is possible to apply an Action Set to only have effect on a specific Setup Group. The general setup of privileges, and a description of all actions, can be found in the **Action Sets** section of the **System Setup / Super User Guide**. A sample of how Setup Groups can be used together with privileges will be illustrated later in this section.

The following Setup Actions are Setup Group specific and can be used on User Groups, to define in which Setup Group a user is allowed to maintain or view Integration Endpoints.

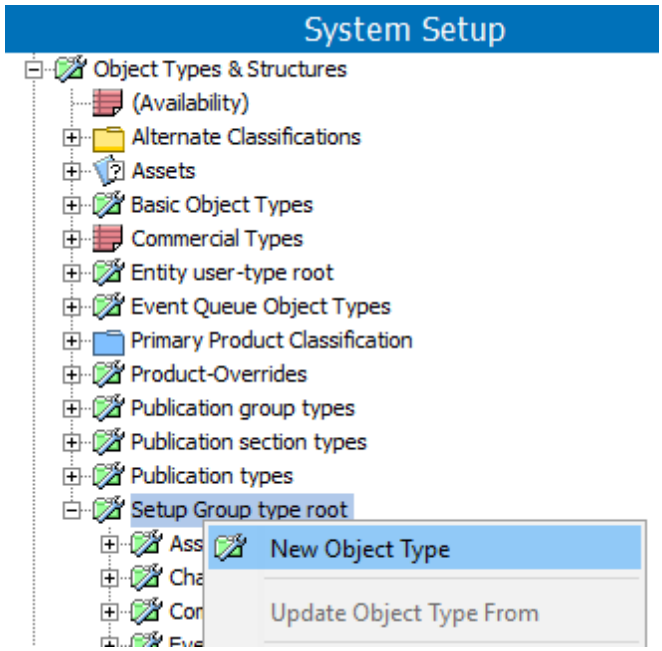
Setup Actions	Description
View Setup Groups	Users with this action applied will be able to View Setup Groups.
Maintain Setup Groups	Users with this action applied will be able to Create and Delete Setup Groups. They are also able to use the cut, copy and paste actions.

Creating Setup Group Object Types

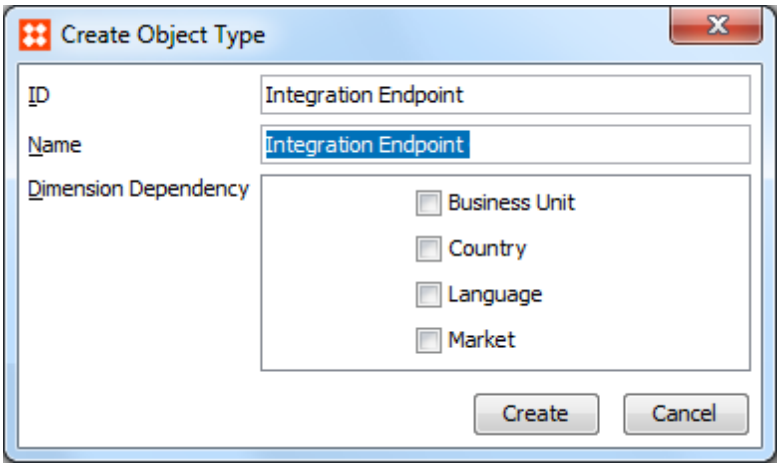
Before you can create any integration endpoints, you need to create one or more **Setup Groups** and define in which Setup Group **Integration Endpoints**, **Business Rules**, **Portal Configurations** or **Workflow Configurations** are allowed to be created.

In the sample below a Setup Group Object Type is created to be used to hold Integration Endpoints.

1. In **Object Types & Structures** in **System Setup**, right-click **Setup Group type root** and click **New Object Type**.

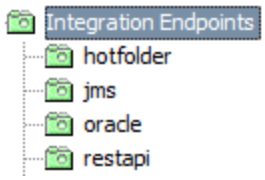


2. Key in **ID**, **Name** and Click **Create**.



The **Setup Group**will appear as a child in the **Setup Group type root**. Repeat from step 1 to create additional object types.

The sample below illustrates a hierarchy of setup group object types. A **Setup Group** named **Integration Endpoints** has been created, and contains four setup group object types named hotfolder, jms, oracle and restapi.



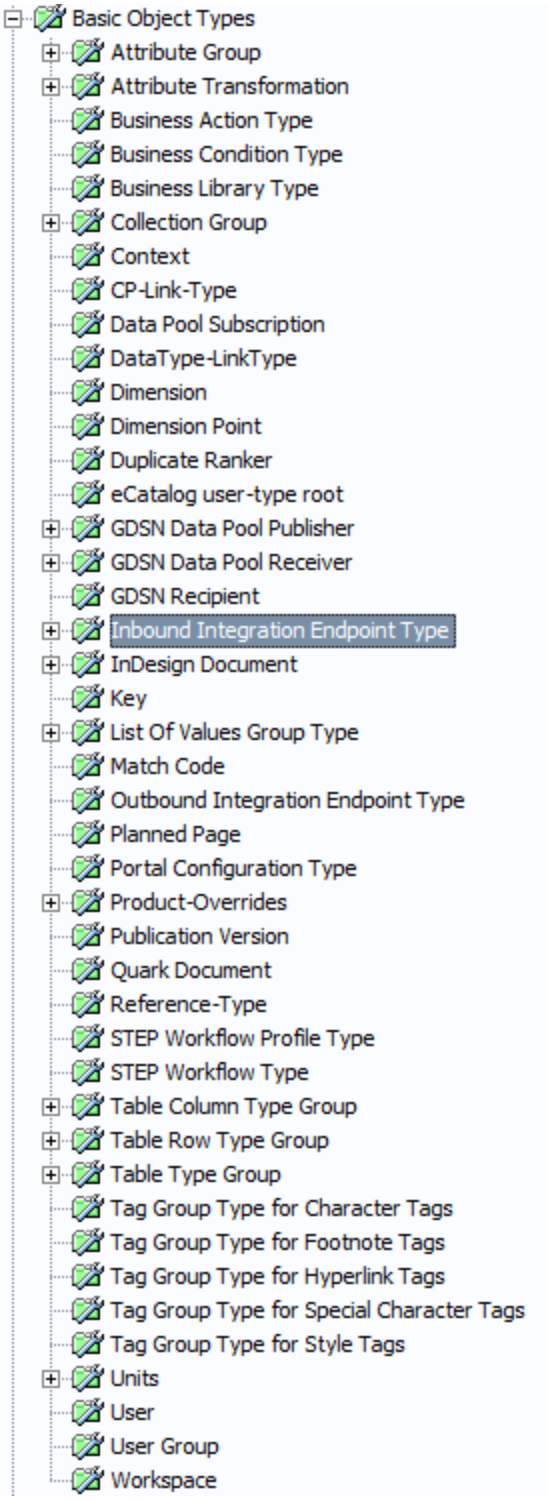
Defining Type of Objects to be Stored in Setup Groups

Before a Setup Group can hold Business Actions, Business Conditions, Business Libraries, Inbound Integration Endpoints, Outbound Integration Endpoints, Workflow Configurations and/or Portal Configurations. Depending on your case, you need to select one of the following Objects in the list of Basic Object Types:

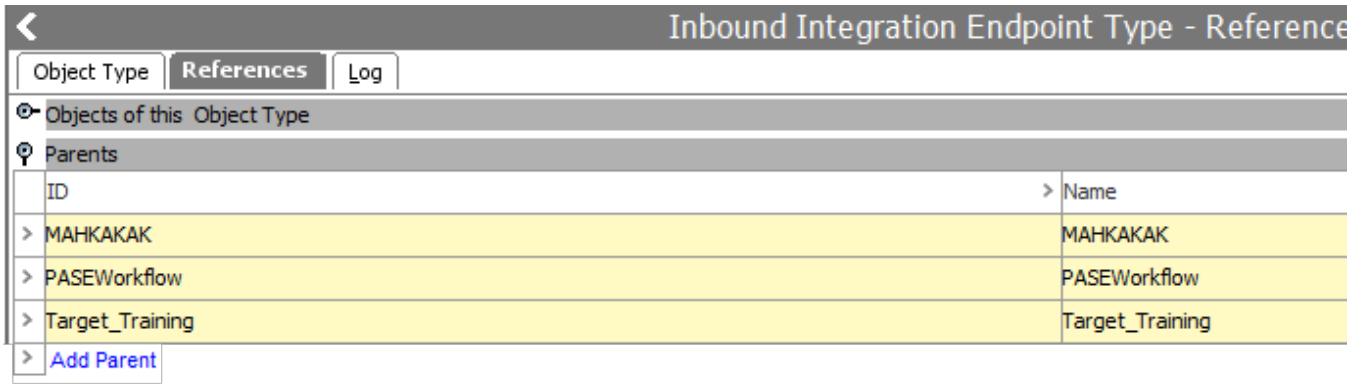
- Business Action Type
- Business Condition Type
- Business Library Type
- Inbound Integration Endpoint Type
- Outbound Integration Endpoint Type
- Portal Configuration Type
- STEP Workflow Type

In the sample below it is illustrated how you are setting up a Setup Group to hold Inbound Integration Endpoints. The same tutorial should be used for the other type of Objects listed above.

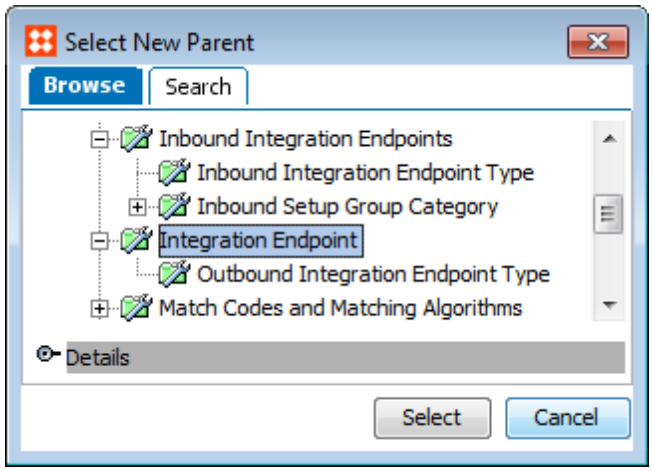
1. In **Object Types & Structures**, in **Basic Object Types**, click **Inbound Integration Endpoints**.


Basic Object Types
Attribute Group
Attribute Transformation
Business Action Type
Business Condition Type
Business Library Type
Collection Group
Context
CP-Link-Type
Data Pool Subscription
DataType-LinkType
Dimension
Dimension Point
Duplicate Ranker
eCatalog user-type root
GDSN Data Pool Publisher
GDSN Data Pool Receiver
GDSN Recipient
Inbound Integration Endpoint Type
InDesign Document
Key
List Of Values Group Type
Match Code
Outbound Integration Endpoint Type
Planned Page
Portal Configuration Type
Product-Overrides
Publication Version
Quark Document
Reference-Type
STEP Workflow Profile Type
STEP Workflow Type
Table Column Type Group
Table Row Type Group
Table Type Group
Tag Group Type for Character Tags
Tag Group Type for Footnote Tags
Tag Group Type for Hyperlink Tags
Tag Group Type for Special Character Tags
Tag Group Type for Style Tags
Units
User
User Group
Workspace

2. In **Inbound Integration Endpoint Type** editor, click **References** tab and **Add Parent**.



A dialog appears where you can search or browse for Setup Groups to link the Object Type into.



3. In **Select New Parent** dialog, browse or search for relevant **Setup Group** and click **Select**.

Creating Setup Groups in System Setup

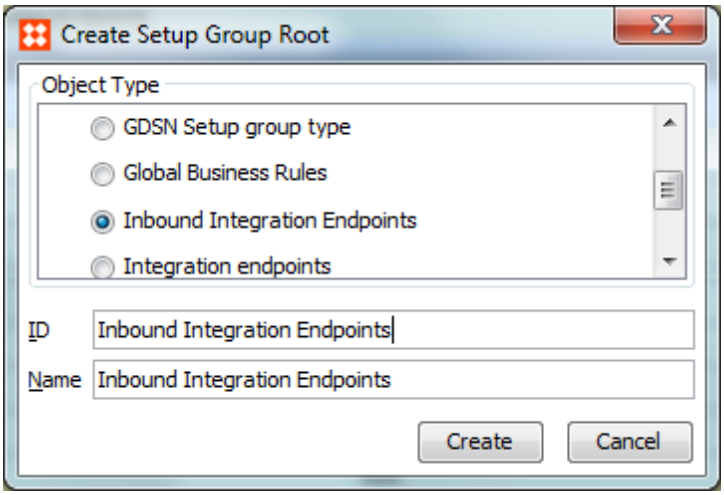
Setup Groups can be used to hold Integration Endpoints, Business Rules and/or Portal Configurations. Before you can create any of those objects, you need to create a Setup Group in System Setup.

Note: It is required that you have created a Setup Group Object Type and specified which type of Object the Setup Group should hold.

---

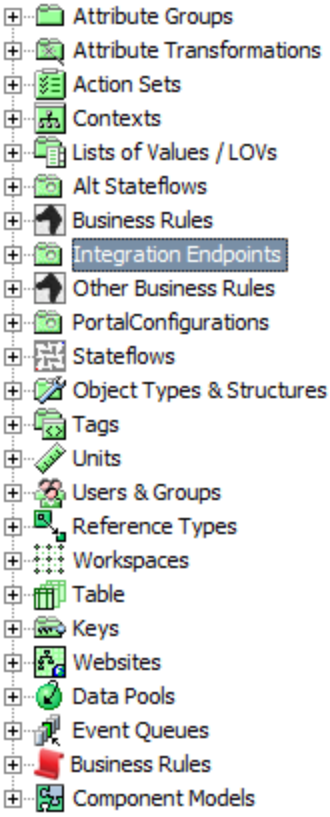
1. In Workbench, click **System Setup** tab, and select any object in the System Setup hierarchy.
2. In main menu, click **Maintain**, and **Insert** -> **Setup Group Root**.

   A **Create Setup Group Root** dialog appears.



3. Key in **ID**, **Name** and click **Create**.

   A Setup Group is created and will appear as a node in the System Setup hierarchy. The Setup Group is ready to be used for Integration Endpoints.



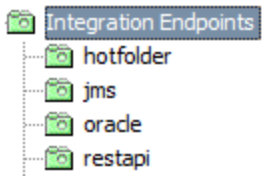
Setting Up Privileges Sample

In this example, two User Groups will be created.

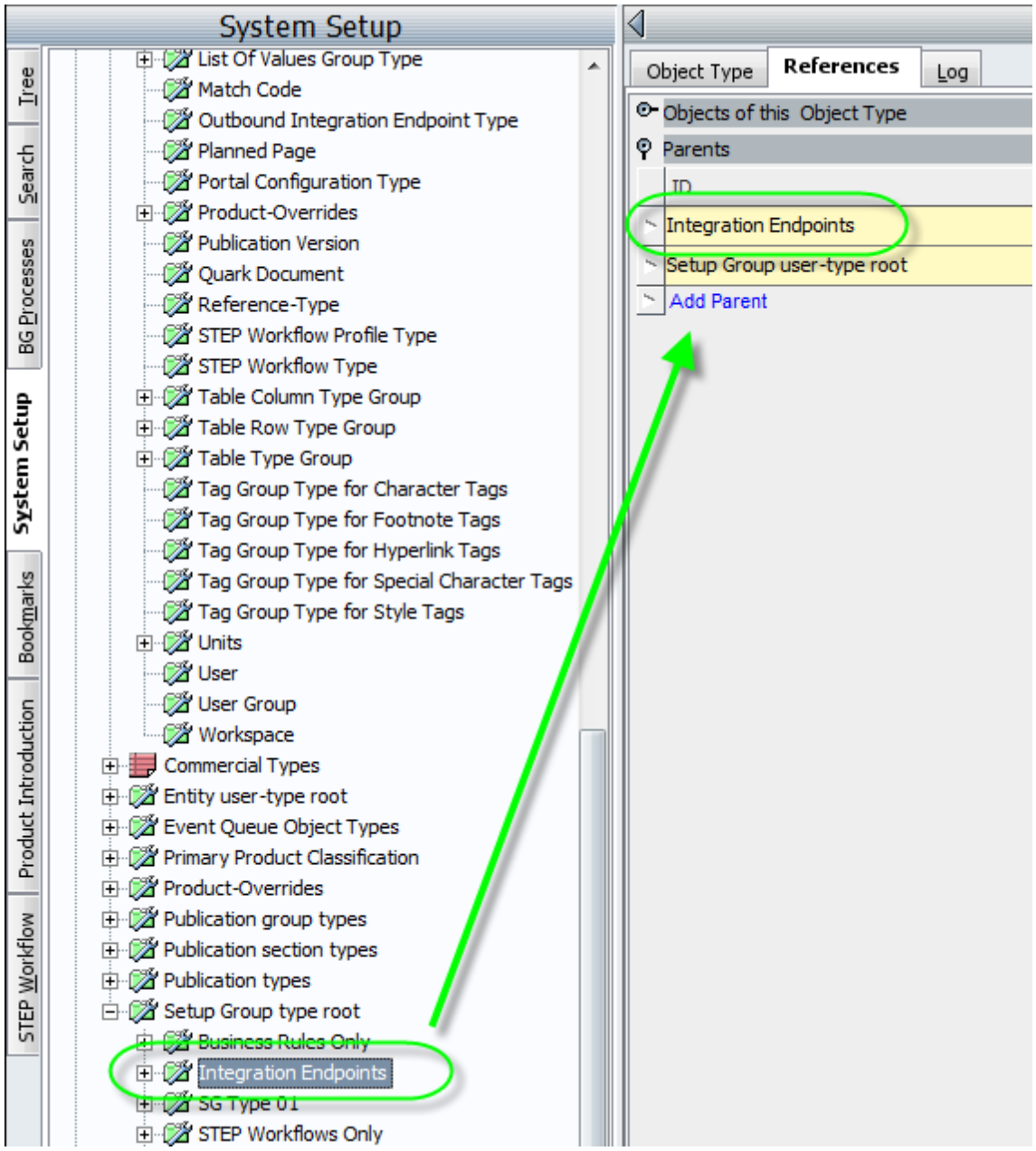
- A User Group named **View Setup Group and Integration Endpoints** will be created. Users linked into this group will be granted the privilege to view Setup Groups and Integration Endpoints.
- A User Group named **Maintain Setup Group and Integration Endpoints** will be created. It will inherit the privileges applied to the **View Setup Group and Integration Endpoint** User Group. Users linked into this group will be granted the privilege to maintain Setup Groups and Integration Endpoints in the Setup Group named hotfolders.

Note: This sample of privilege setup can only be used to link existing Users into one of the two User Groups. Users will have to exist in other User Groups, where they grant the privilege to log on the Workbench. It is beyond the scope of this guide to describe this functionality. For more information about Action Sets please see Action Sets in the Super User / System Setup.

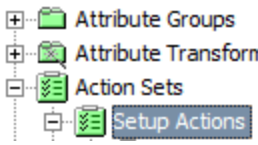
A structure of the Setup Groups could be as illustrated below.



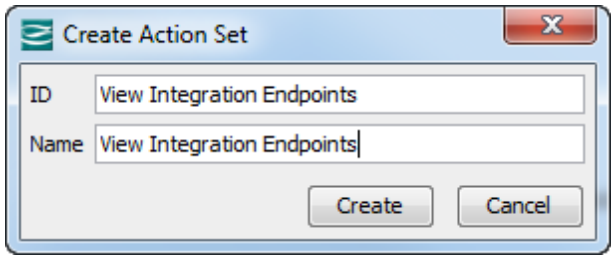
Note: To be able to create additional **Setup Groups**, able to hold Integration Endpoints, below the **Integration Endpoint** Setup Group, as shown in the example above, the **Setup Group type root** "Integration Endpoints" must have itself added as a parent.



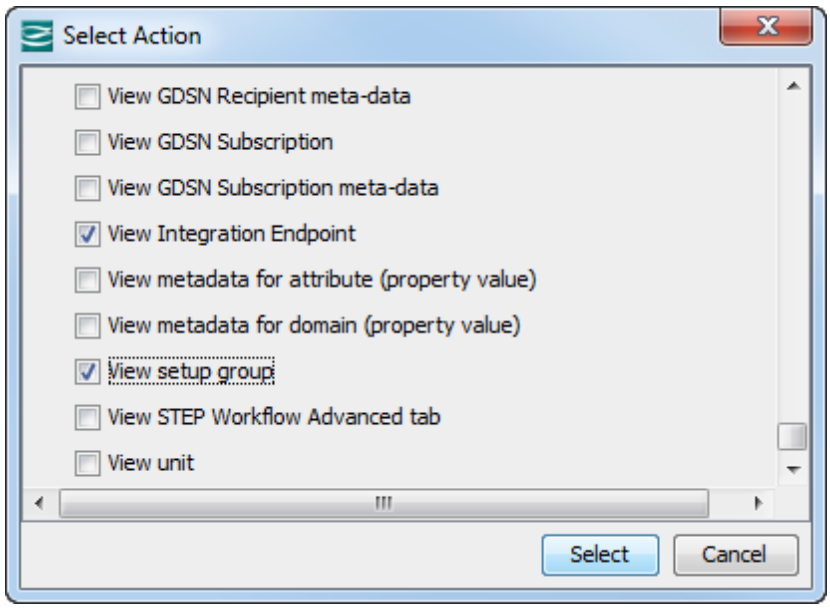
1. In **System Setup**, Select **Setup Actions** in Action Sets.

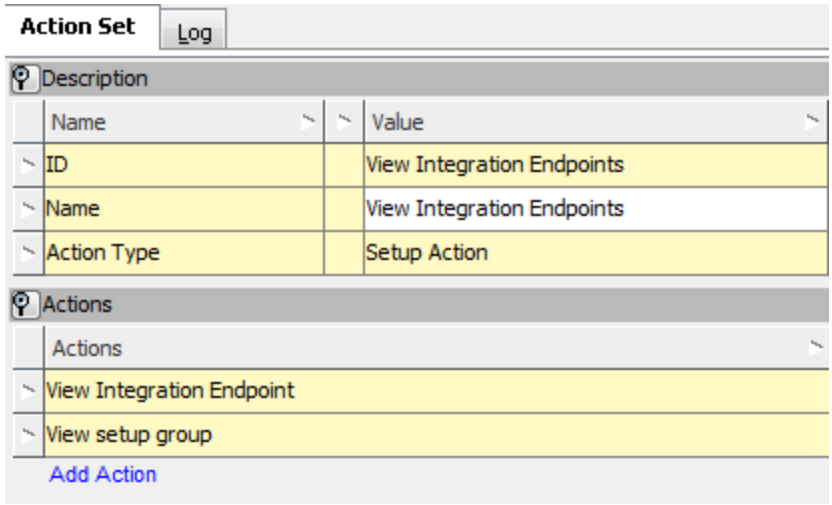


2. Right-Click and select **New Action Set**.
3. Key in **ID**, **Name** and click **Create**.

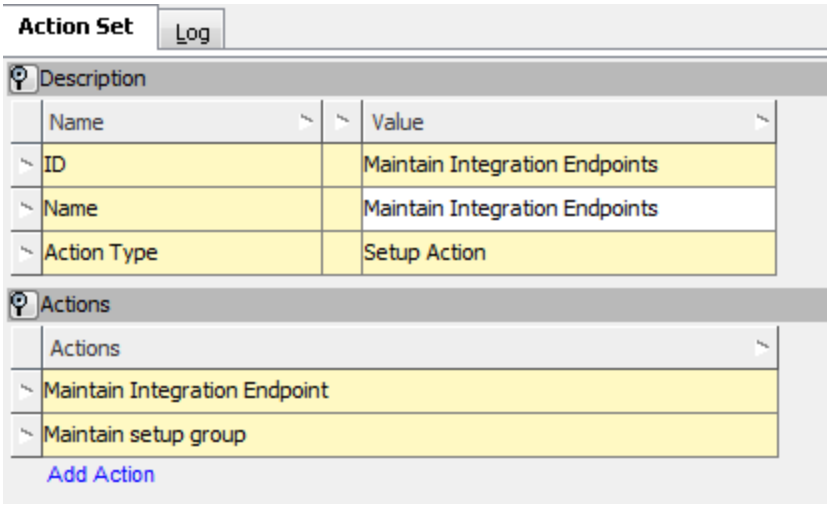


4. In **Action Set** editor, click **Add Action**.
5. In **Select Action dialog**, check **View Integration Endpoint** and **View Setup Group** and click **Select**.

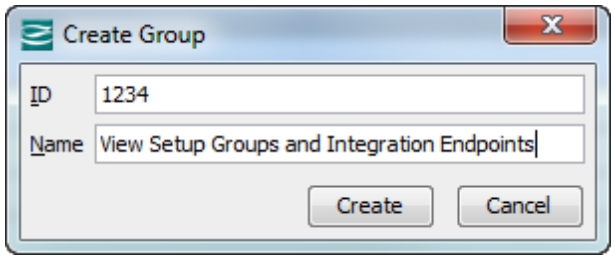




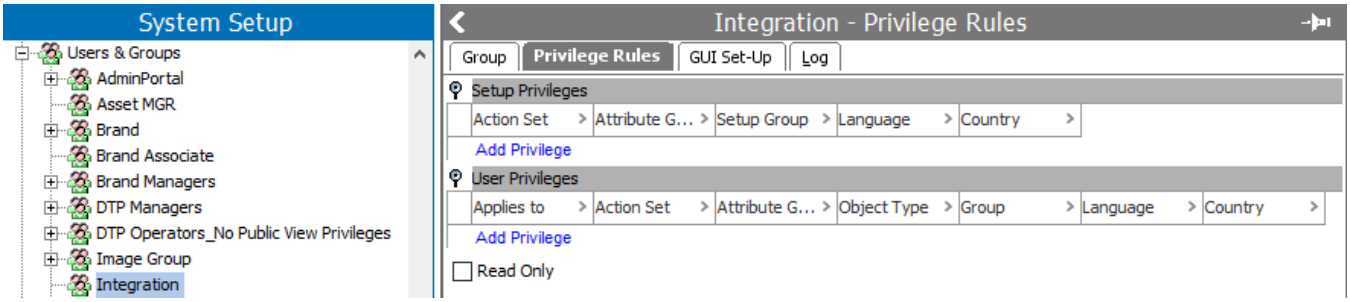
6. Repeat step 1-5 and create an Action Set containing the actions **maintain setup group** and **maintain integration endpoint**.



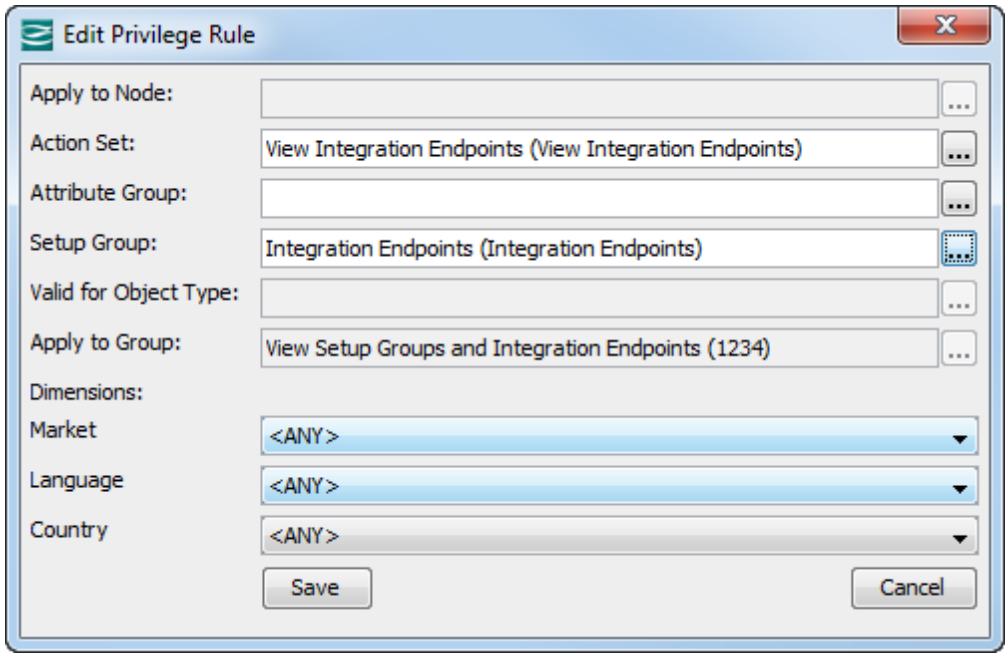
7. In **Users & Groups**, right-click and select **New Group**.



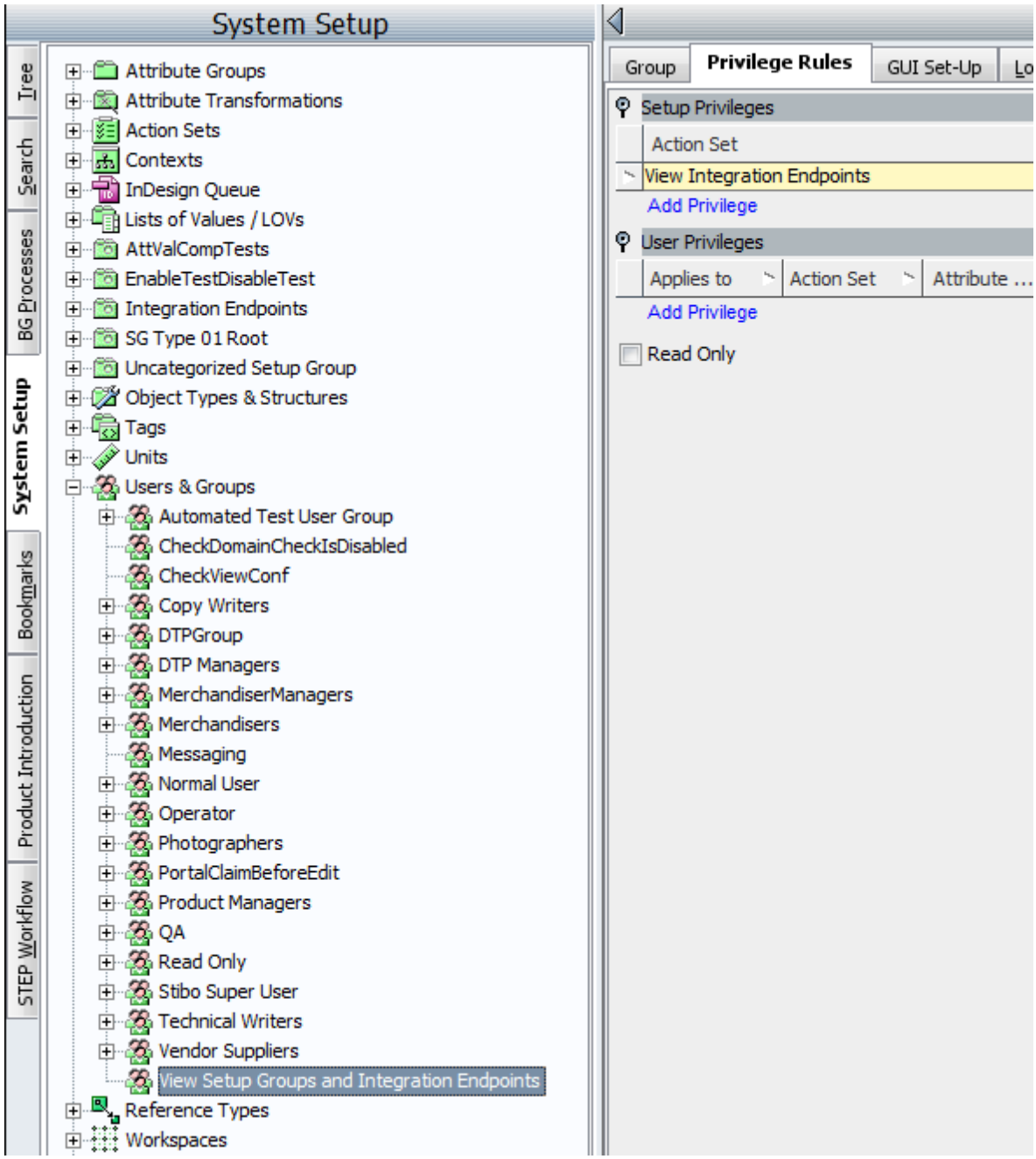
8. Key in **ID** and **Name** and click **Create**.
9. In **User Group editor**, click **Privilege Rules** tab.

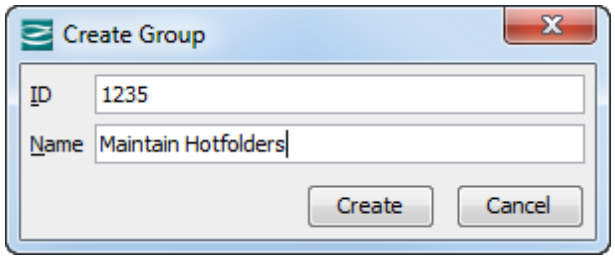


10. In **Setup Privileges**, click **Add Privilege**. The Edit Privilege Rule dialog will appear:



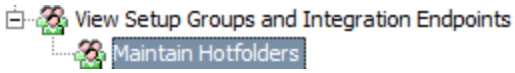
11. In **Action Set**, apply the **View Integration Endpoint** Action Set.
12. In **Setup Group**, apply the Setup Group named **Integration Endpoints**.
13. In **Users & Groups**, right-click View Setup Groups and Integration Endpoints, and select **New Group**.



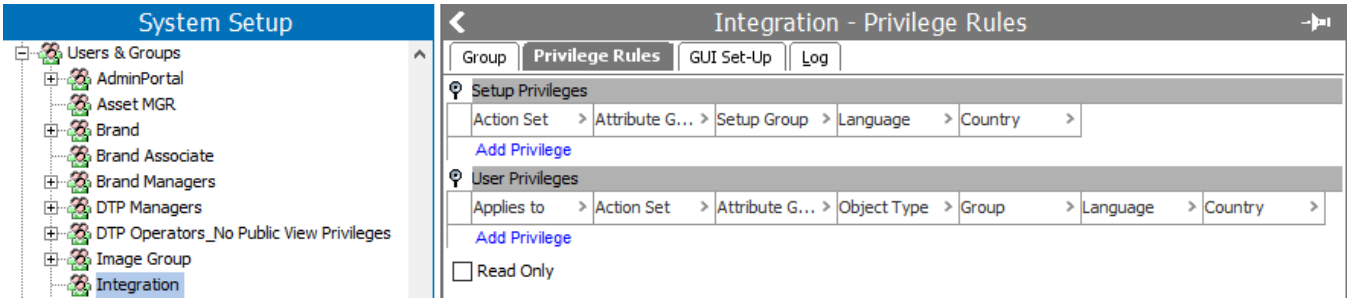


14. Key in **ID**, **Name** and click **Create**.

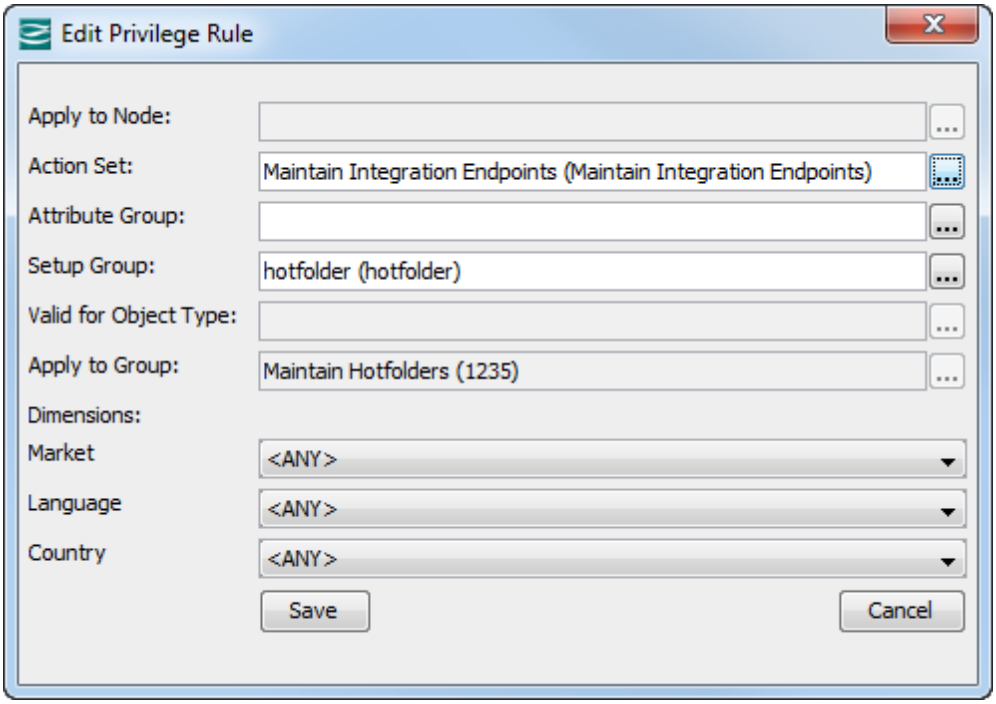
The **User Group** will appear below the User Group **View Setup Groups and Integration Endpoints**.



15. In **User Group editor**, click **Privilege Rules** tab.



16. In **Setup Privileges**, click **Add Privilege**.



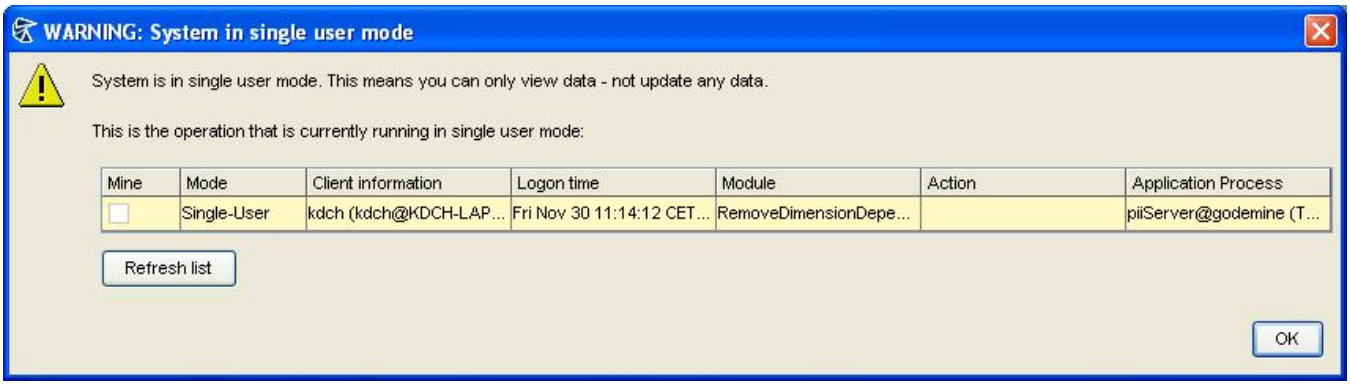
17. In **Action Set**, apply the Action Set **Maintain Integration Endpoints**.
18. In Setup Group, apply the setup group **hotfolder**.

    Users linked into the User Group **View Setup Group and Integration Endpoint** will only be granted the privilege to view Setup Groups and Integration Endpoints. Users linked into **Maintain Integration Endpoints**, will be granted the privilege to maintain Setup Groups and Integration endpoints in the setup group "hotfolder".

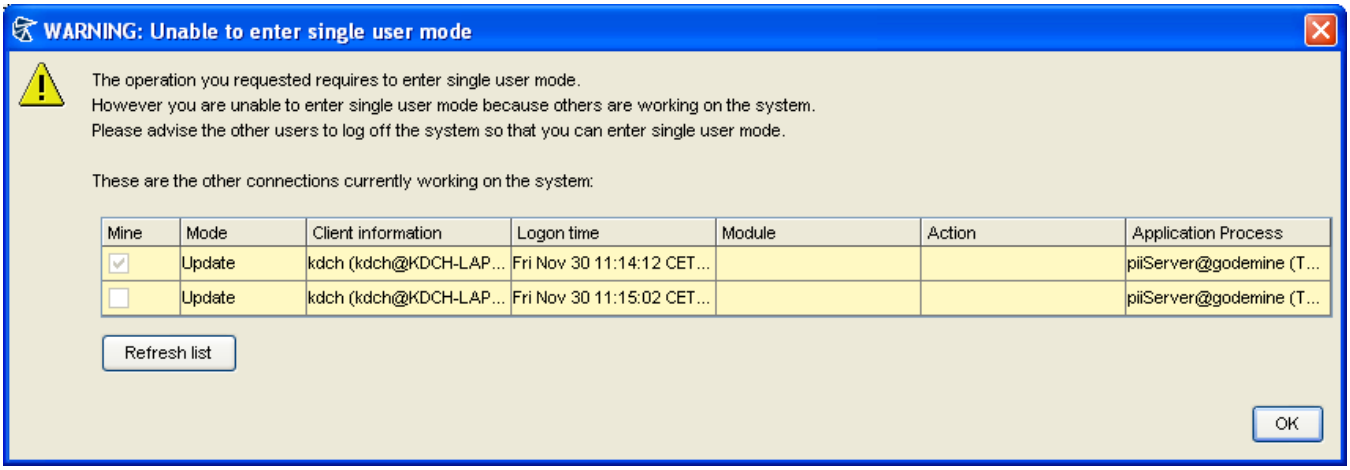
Single User Mode

The system enters Single Update Mode when you perform operations that can affect data elsewhere in the system. When the system enters Single Update Mode, only the connection that is in Single Update Mode can update the database.

A connection can only enter Single Update Mode if no other connection in the system is in update or Single Update Mode. If it is not possible for the system to enter Single Update Mode, a warning appears.

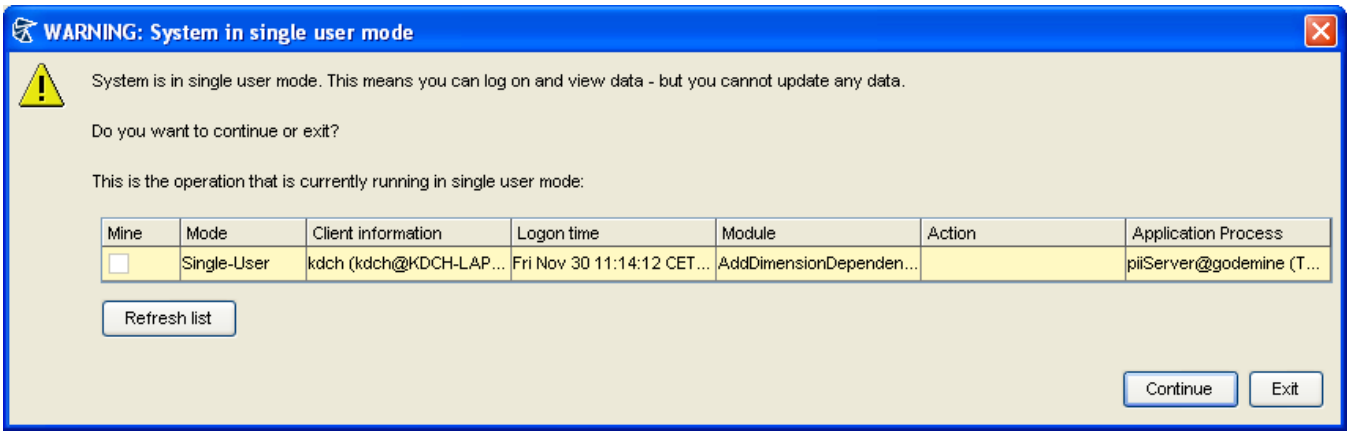


You can still view data when the system is in Single Update Mode. However, if you make changes and try to update when another connection is in Single Update Mode, a warning appears.



The warning also displays information about which connection is currently running in Single Update Mode.

If you log on to the system while it is in Single Update Mode, a warning appears.



Operations that Cause Single Update Mode

The following operations will cause the system to enter Single Update Mode.

- Deleting a Workspace
- Adding or deleting Dimension dependencies
- Merging Attributes (with the exception of Bulk Updates where values are copied to new attributes)
- Changing Attributes to or from being externally maintained
- Changing Attributes to use or not to use LOV
- Changing Attributes to or from being free text searchable
- Changing Attributes to or from being multi valued
- Activating or deactivating unique keys.
- Changing reference types to or from being externally maintained
- Changing reference types to or from being multi valued
- Changing Product to Classification Link Types to or from being externally maintained
- Changing or moving Product to Classification Link Types (for example, when you move one type to another type)
- Changing Product to Classification Link Types to or from being multi valued
- Merging LOVs (unless neither LOV is in use by an attribute)
- Removing Object Types
- Changing the Owns Product Links setting on classification object type
- Changing the Revisability setting of an entity object type

To avoid entering Single User mode when changing attribute definitions, the following procedure can be used:

1. Create a new attribute (Attribute B) with the same desired attribute definition (including the changes from the origin attribute, Attribute A).
2. Use Bulk Update in the STEP Workbench to copy all attribute values from Attribute A to Attribute B. For more information on using bulk update functionality, see the 'Bulk Updates' section of the STEP User / Getting Started documentation.
3. Switch the IDs of Attribute A and Attribute B. To do this, navigate to either attribute on the System Setup tab of the workbench. Right-click on the attribute and select 'Switch Attribute IDs'. In the resulting dialog, select the attribute that the selected attribute should swap IDs with.

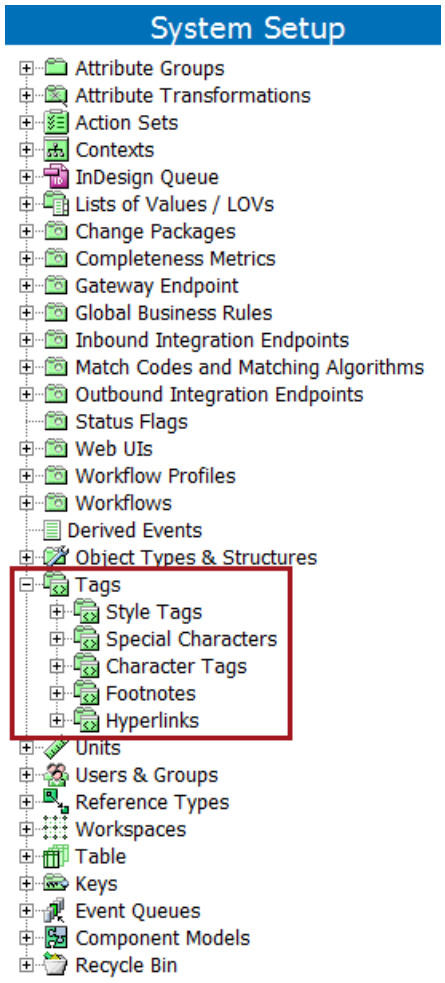
Tags

This section describes how to set up special characters, character tags, and style tags.

About Tags

A **tag** is a markup notation used to indicate specific formatting or special characters. A tag is identified by a unique Tag Name. Tags are available from Rich Text Editors and the STEP Table Component.

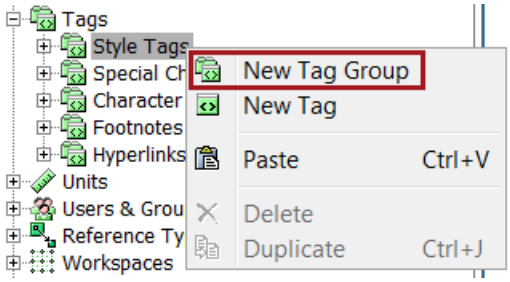
Tags are maintained in **System Setup** in the **Tags** folder.



Creating a Tag Group

A Tag Group holds a number of Tags. The Group can be linked into an Attribute Group and used in combination with privileges to restrict users in using certain Tags.

1. In **System Setup**, right-click Special Characters, Style Tags, Character Tags, Footnotes, or Hyperlinks.
2. In the menu, select **New Tag Group**.



3. Key in **ID** and **Name**, then click **Create**.

Note: If the Tag Group is used in combination with privileges to restrict users in maintaining and/or using certain Tags, you must link the Tag Group into an Attribute Group. The Attribute Group can be used in combination with setup privileges to restrict certain Tags to be maintained or used in Rich Text Editor or Table Editor.

Special Characters

Special characters are characters that are pasted into a text element or a data cell, but where the character as it displays on the screen is not available in a normal font on the print output device.

Examples are the Greek characters α β γ. These characters are valid Unicode characters and can be entered into any text element or into product data. To make these characters output to a printer or to output properly in, for example, InDesign, a formatting has to be set up.

Four standard special characters are predefined:

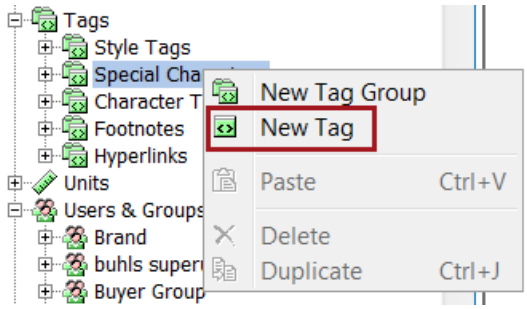
Standard Special Characters	Description
No-break Space (NBSP)	'No-break Space' character is used to prevent splitting a line at a space
Carriage Return (CR)	Carriage Return (CR) is a return
Line Feed (LF)	Line Feed (LF) is a soft return
Horizontal Tabulation (HT)	Tabulator

These four standard special characters represent characters that are not visible in the Rich Text Editor in the STEP Workbench. For example, when the ENTER key is pressed in the Rich Text Editor, the special character Line Feed (LF) is automatically inserted, though it is invisible in the editor.

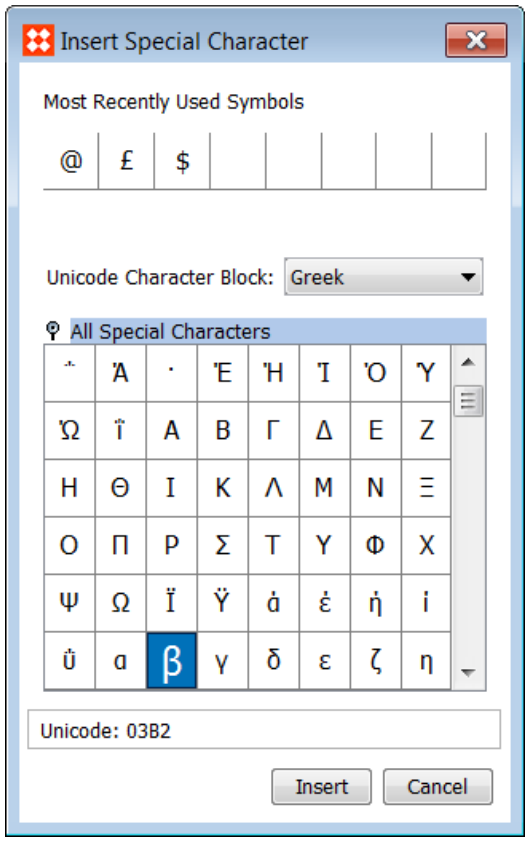
Note: Media-specific formatting (e.g., HTML or InDesign) must be set up for these special characters in STEP in order for the output systems to interpret the characters correctly.

Creating a Special Character

1. In **System Setup**, expand **Tags**.
2. Click **Special Characters**, right-click, and then select **New Tag**.



3. In the **Insert Special Character** dialog, select a character.



4. Click **Insert**. The Special Character is created.

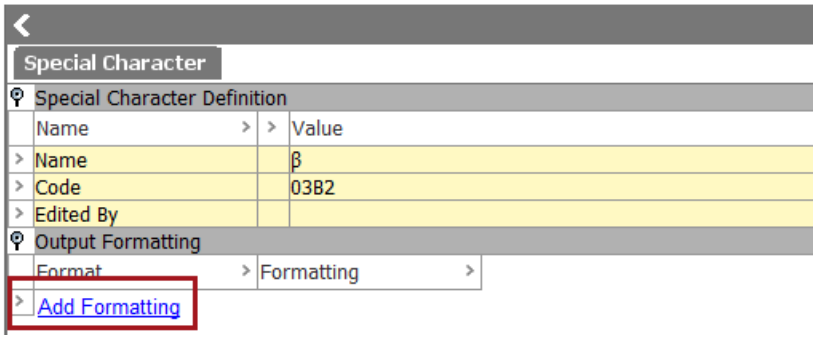
Setting Up Output Formatting for Special Characters

Output formatting is a media-specific interpretation and conversion of STEP tags. Output formatting must be set up for Special Character tags to ensure that the output is displayed properly in applications such as InDesign.

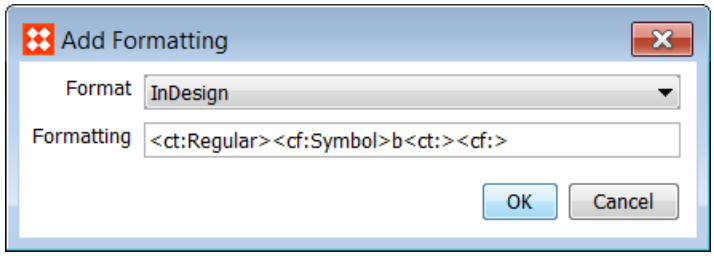
For example, if the special character Beta (β) is selected in the Special Character hierarchy, a conversion of this tag must be set up in order for InDesign to interpret and convert the tag correctly.

In this example, InDesign has been set up to output the β character using the Symbol font. The tag will now be converted into a format that InDesign can interpret. Each time a value containing β is mounted in InDesign, a Beta symbol will appear.

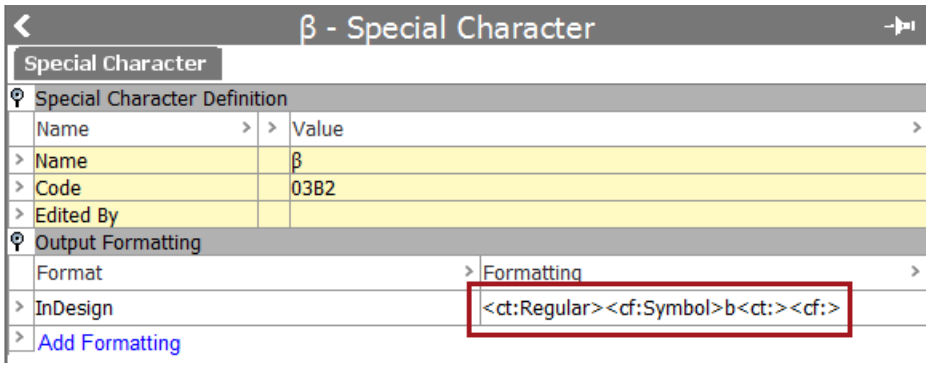
1. In **System Setup**, navigate to the relevant Special Character beneath Tags > **Special Characters**.
2. Under the **Output Formatting** flipper, click **Add Formatting**.



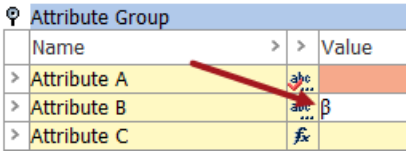
3. In the **Add Formatting** dialog, choose the relevant output format from the **Format** dropdown list. In this example, the InDesign output format is selected.



4. Type a string in the **Formatting** field. Example: `<ct:Regular><cf:Symbol>b<ct:><cf:>`
5. Click **OK**.



Note: Rendering does not have to be defined for special characters, as they are visible within the text editors in STEP.



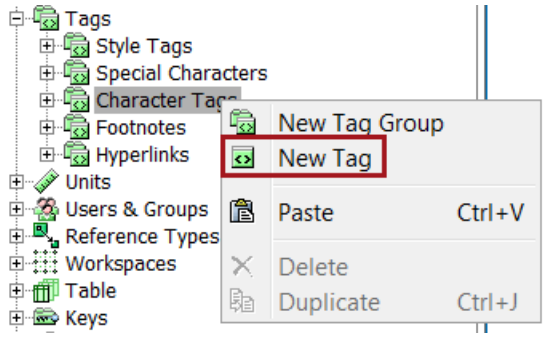
Character Tags

Character tags handle very special characters, sometimes known as Dingbats. These characters are not always available in the Unicode set, or they are difficult to find in the Unicode set. It is therefore easier to use a tag to access the character. This might be the case with characters such as a telephone ☎ or a special arrow ➔.

Because there are several thousand special characters available on many different types of fonts, it may be necessary to invent a tag name (such as 'telephone'), then access that specific character in that font in the character conversion table.

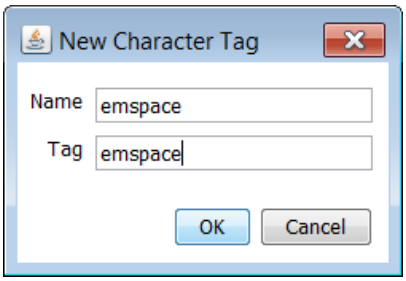
Creating a Character Tag

1. In System Setup, expand **Tags**.
2. Click **Character Tags**, right-click, and then choose **New Tag**.



3. In the **New Character Tag** dialog box, enter a **Name** (for example, emspace). The Name is what shows up in the list of Character Tags on the left of the screen.
4. Next, enter a **Tag** (for example, emspace). The actual Character Tag entry is what STEP expects to see. It is common to use the same word for Name and Tag to avoid confusion.

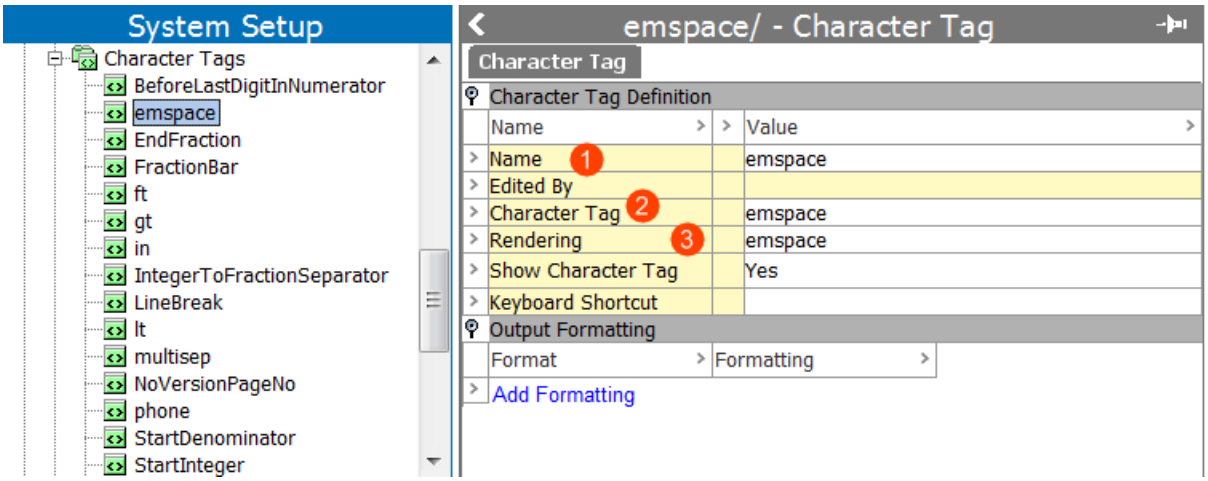
Note: All STEP tags are case-sensitive.



5. Click **OK** to create the Character Tag.

Configuring the Character Tag

When a Character Tag is selected, the **Character Tag** editor is displayed.



The following fields are available in the editor:

1. **Name:** A display name for the Character Tag. A Character Tag name cannot exceed 80 characters. Spaces between words is allowed.
2. **Character Tag:** May only be one word. Spaces and the characters < / > are not allowed. A character tag begins with < and ends with / >. The system will automatically insert the characters < and /> to surround the character tag.
3. **Rendering:** A rendering is used to define how a character tag is displayed in the Rich Text Editors. In this example, the rendering is the word **emspace**.

Setting Up Rendering of Character Tags

Rendering is used to define how a character tag is displayed in the Rich Text Editors in STEP. The rendering entry may be one or more characters. In STEP, these characters will always show up with a gray background.

Note: If a character tag does not have a rendering specified, the tag is rendered [Undefined] in Rich Text Editors. This means that each time the tag is used in Rich Text Editors, then the rendering [Undefined] is shown.

---

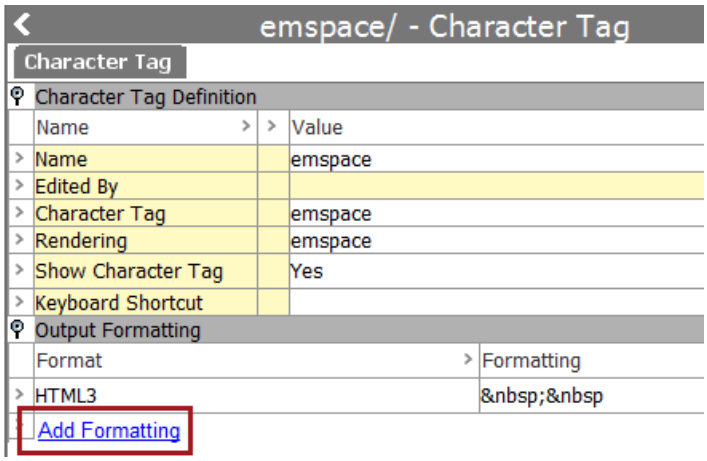
1. On the System Setup folder, expand **Tags**, and then expand **Character Tags**.
2. Select the preferred Character tag.
3. On the **Character Tag** tab, in the **Character Tag Definition** table, double-click the **Rendering** field.
4. Enter a rendering for the character tag.

Setting Up Output Formatting for Character Tags

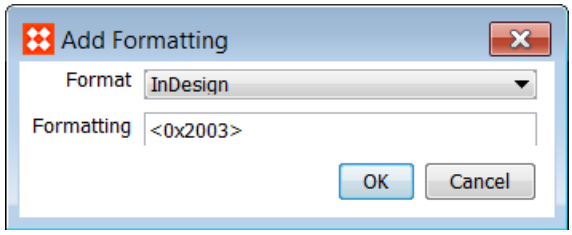
Output formatting is a media-specific interpretation and conversion of STEP tags. Output formatting must be set up for Character Tags to ensure that the output is displayed properly in HTML and applications such as InDesign.

Character Tags are 'self-contained' in that they do not require a opening and closing tag. But when importing these tags into STEP, the syntax must be in accordance with STEP protocol. For the example in the below steps, this would be <emspace/>.

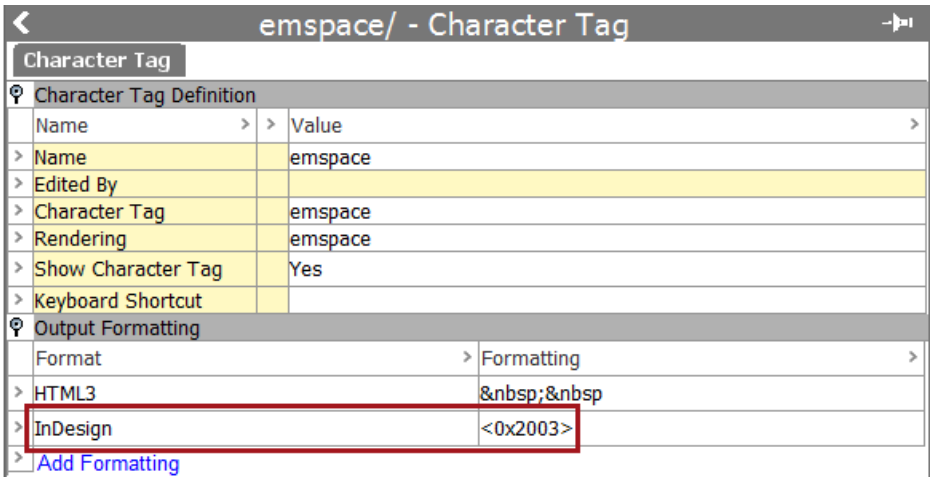
1. In **System Setup**, navigate to the relevant Special Character beneath Tags > **Special Characters**.
2. Under the **Output Formatting** flipper, click **Add Formatting**.



3. In the **Add Formatting** dialog, choose the relevant output format from the **Format** dropdown list. In this example, the InDesign output format is selected.



4. Type a string in the **Formatting** field. Example: `<0x2003>`
5. Click **OK**.


emspace/ - Character Tag
Character Tag
Character Tag Definition
Name
Value
Name
emspace
Edited By
Character Tag
emspace
Rendering
emspace
Show Character Tag
Yes
Keyboard Shortcut
Output Formatting
Format
Formatting
HTML3

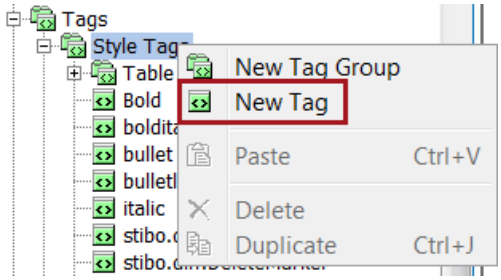
InDesign
<0x2003>
Add Formatting

Style Tags

Style Tags control the font and the appearance of text such as bold, underline, color, and so forth. Style Tags must have an opening and closing tag, for example, <bolditalic> </bolditalic>. The system automatically inserts the symbols < / >. Style Tags are frequently used with STEP Tables in order to apply styling to the contents of table cells.

Creating a Style Tag

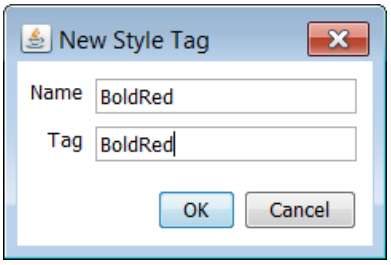
1. In **System Setup**, expand the **Tags** hierarchy.
2. Click **Style Tags**, right-click, and then choose **New Tag**.



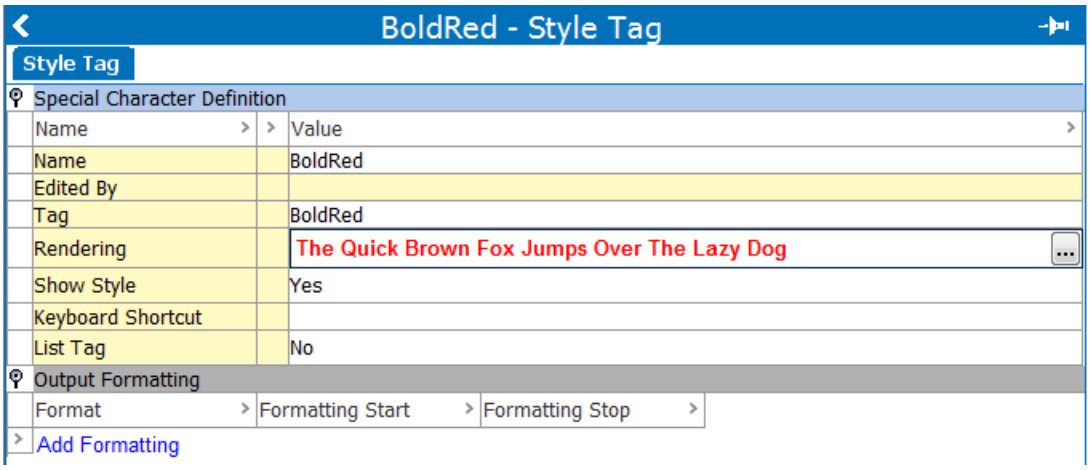
3. In the **New Style Tag** dialog, enter a **Name** and a **Tag** for the new style tag.

   The entry for the **Name** of the tag is what shows up in the list of tags on the left panel. The **Tag** entry is what is actually used. It is common to use the same word for Name and Tag to avoid confusion.

   Note: All STEP tags are case-sensitive.



4. Click **OK**. The tag is created, and is selected in the **Style Tags** hierarchy.
5. On the **Style Tag** tab, modify the relevant fields.



6. The **Name** field displays the name of the style tag. The name can have a maximum of 80 characters. Spaces between words is allowed.
7. In the **Tag** field you can enter only one word. Spaces and the characters < / > are not allowed.
8. In the **Rendering** field, click the ellipses points **[…]** to open the **Rendering** dialog. The settings in this dialog define how a style tag is displayed in the Rich Text Editors in STEP. For more information, see the next section of this documentation, **Setting Up Rendering of Style Tags**.

   Note: If a style tag does not have a rendering, the tag cannot be used in text editors in STEP. It cannot be selected in Rich Text Editors at all.

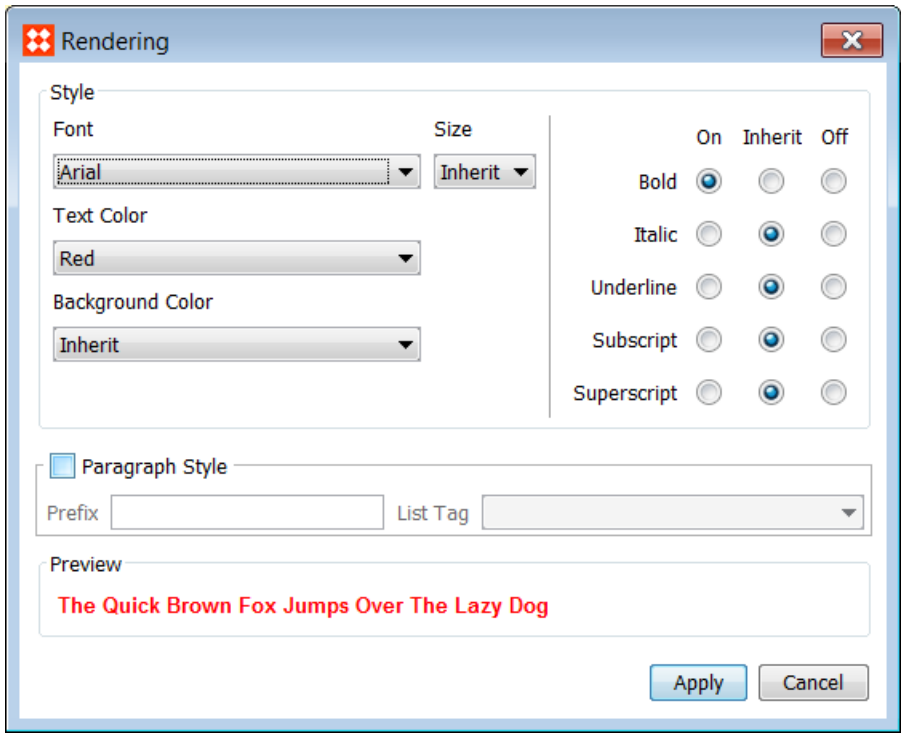
9. In the **Show Style** field, **Yes** is typically selected.

   Note: Selecting **No** will make the tag unavailable for selection in all dropdown lists available in STEP, both in the rich text editors and in the table component. **No** is only used for 'hidden' styles such as stibo.STIBOSPELL, which is used by the system to highlight spelling errors in STEP's text editors.

10. In the **List Tag** field, select **Yes** if the style tag should be used to define bullet lists for HTML output. (E.g., if the style will be used with the HTML tags <ul> and .)

Setting Up Rendering for Style Tags

1. On the System Setup tab, expand **Tags**, and then expand **Style Tags**.
2. Select the preferred style tag.
3. On the **Style Tag** tab, in the **Special Character Definition** table, click the **Rendering** field, and then click the **[…]** ellipses button. The **Rendering** dialog box appears.

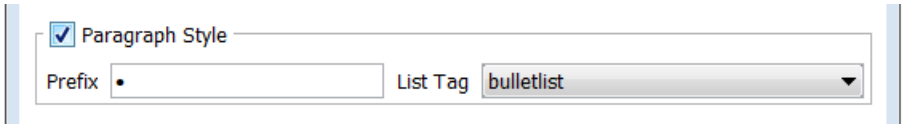


4. In the **Style** area, define the preferred style settings. If you select **Inherit**, no specific style is applied to the style tag. Instead. it inherits a style from another style tag when style tags are combined in the rich text editors.

Style	Description
Font	Displays a list of the fonts installed on the computer. You can select a specific font to be used in the rich text editors.
Size	Displays a list of available font sizes. You can select a specific font size to be used in the rich text editors.

Style	Description
Text Color	Displays a list of standard text colors
Background Color	Displays a list of standard background colors
Bold	Select this to render the style tag 'bold'
Italic	Select this to render the style tag 'italic'
Underline	Select this to render the style tag 'underlined'
Subscript	Select this to render the style tag 'subscript'
Superscript	Select this to render the style tag 'superscript'

5. If you have defined a style tag that will be used to define bullet lists for HTML output, check the **Paragraph Style** check box. The **List Tag** selected from the List Tag dropdown list will be another style tag that was created to start and end an entire bullet list (one that will use <ul> and , for example). The style tag for which you are currently configuring the rendering will be used to define the individual items within the bullet list (<li> and).
   - In the **Prefix** field, enter a character or string of text to be used as prefix for the bullet list, e.g. •
   - From the **List Tag** list, select the list tag to be used for the paragraph (the entire bullet list).



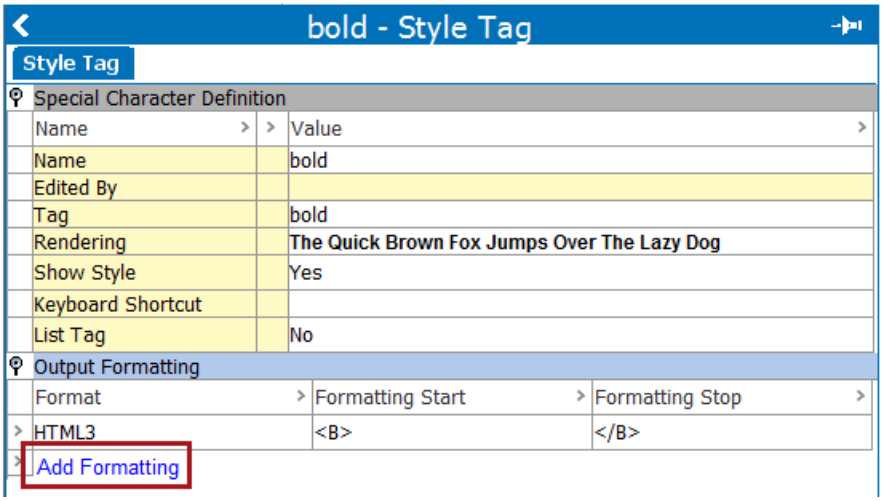
Note: Only check **Paragraph Style** on normal style tags and not on tags defined as list tags. (A style tag is defined as a List Tag if 'Yes' is selected for List Tag on the **Style Tag** tab.)

6. Click **Apply**.

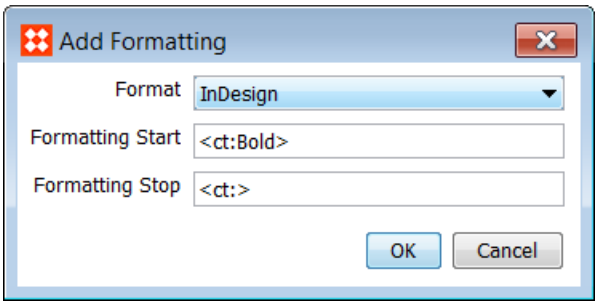
Setting Up Output Formatting for Style Tags

Style Tags must have an opening and closing tag. For example, `<B> ... </B>` for HTML or `<ct:Bold> ... <ct:>` for InDesign.

1. In **System Setup**, navigate to the relevant Style Tag beneath Tags > **Style Tags**.
2. Under the **Output Formatting** flipper, click **Add Formatting**.



3. In the Add Formatting dialog, choose the relevant output format from the **Format** dropdown list. In this example, the InDesign output format is selected.



4. Type a string in the **Formatting Start** field. Example: `<ct:Bold>`
5. Type a string in the **Formatting Stop** field. Example: `<ct:>`
6. Click **OK**.

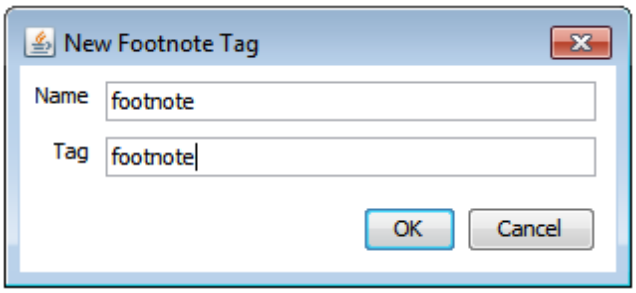

bold - Style Tag
Style Tag
Special Character Definition
Name
Value
Name
bold
Edited By
Tag
bold
Rendering
The Quick Brown Fox Jumps Over The Lazy Dog
Show Style
Yes
Keyboard Shortcut
List Tag
No
Output Formatting
Format
Formatting Start
Formatting Stop
InDesign
<ct:Bold>
<ct:>
HTML3
<B>
</B>
Add Formatting

Footnotes

Footnote Tags are used to define footnotes when they are used in Rich Text Editors for attribute values. They are also required in order to use the Footnote Transformation within STEP tables.

To Create a Footnote Tag

1. On the **System Setup** tab, expand **Tags**.
2. Right-click **Footnotes**, then select **New Tag**.
3. In the **New Footnote Tag** dialog box, enter a display **Name** for the tag. The name can be up to 80 characters long, and you can add space between words.



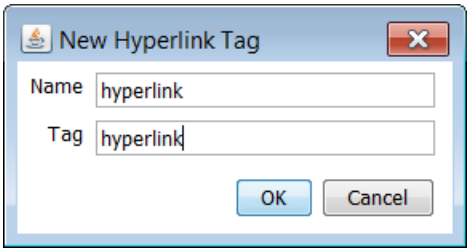
4. In the **Tag** field, enter a tag. You can not use spaces or the special characters <, /, or > .
5. Click **OK**. The tag is created, and is selectable in the **Footnotes** hierarchy.

Hyperlinks

Hyperlink Tags are used to define the formatting of hyperlinks when they are used in Rich Text Editors.

To Create a Hyperlink Tag

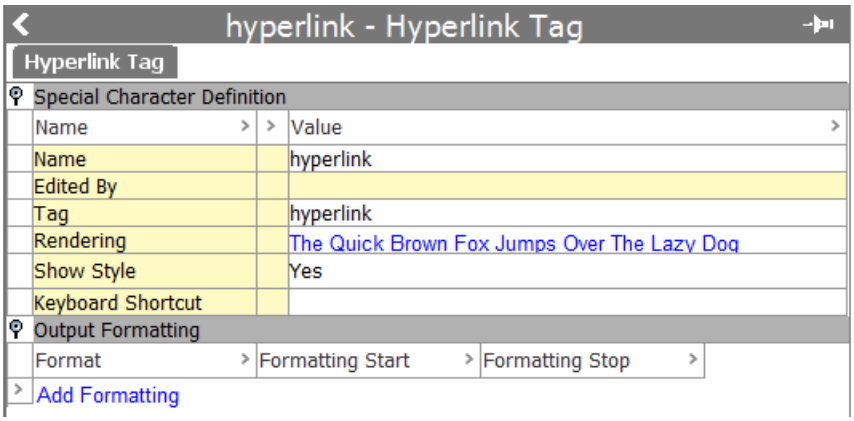
1. On the **System Setup** tab, expand **Tags**.
2. Right-click **Hyperlinks**, then select **New Tag**.
3. In the **New Hyperlink Tag** dialog box, enter a display **Name** for the tag. The name can be up to 80 characters long, and you can add space between words.



4. In the **Tag** field, enter a tag. You can not use spaces or the special characters < / > .
5. Click **OK**. The tag is created, and is selectable in the **Hyperlinks** hierarchy.

Configuring a Hyperlink Tag

1. On the **Hyperlink Tag** tab, modify the relevant fields.

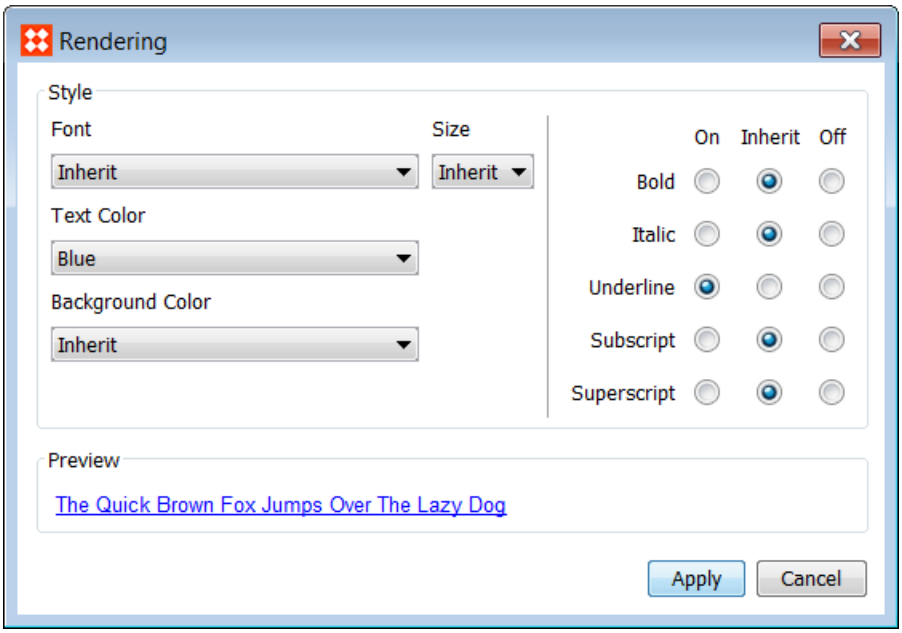


2. The **Name** field displays the name of the hyperlink tag. The name can have a maximum of 80 characters. Spaces between words are allowed.

3. In the **Tag** field you can enter only one word. Spaces and the characters < / > are not allowed.
4. In the **Rendering** field, click the ellipses points **[…]** to open the **Rendering** dialog. The settings in this dialog define how a style tag is displayed in the Rich Text Editors in STEP. For more information, see the next section of this documentation, **Setting Up Rendering of Hyperlink Tags**.
5. In the **Show Style** field, specify whether or not the Hyperlink tag is enabled in the selection list in Rich Text Editors. Select **Yes** to make the Hyperlink tag available and **No** when you do not want the hyperlink tag to be selected from Rich Text Editors.
6. In the **Keyboard Shortcut** field, specify a keyboard shortcut that applies the Hyperlink tag.

Setting Up Rendering of Hyperlink Tags

1. On the **System Setup** tab, expand **Tags**, and then expand **Hyperlinks**.
2. Select the preferred **Hyperlink Tag**.
3. On the **Hyperlink** tab, in the **Special Character Definition** table, click the **Rendering** field, and then click the **[…]** ellipses button. The **Rendering** dialog box displays.



4. In the **Style** area, define the preferred style settings. The following style settings are available:

Style	Description
Font	Displays a list of the fonts installed on the computer. You can select a specific font to be used in the rich text editors.
Size	Displays a list of available font sizes. You can select a specific font size to be used in the rich text editors.
Text Color	Displays a list of standard text colors
Background Color	Displays a list of standard background colors

Style	Description
Bold	Select this to render the style tag 'bold'
Italic	Select this to render the style tag 'italic'
Underline	Select this to render the style tag 'underlined'
Subscript	Select this to render the style tag 'subscript'
Superscript	Select this to render the style tag 'superscript'

Note: If you select **Inherit**, no specific style is applied to the Hyperlink tag. Instead it inherits a style from another style tag when style tags are combined in the Rich Text Editor.

5. Click **Apply**. The Rendering has been set up.

About Unique Keys

A key is a unique database entity composed of (transformed) product data, and represents a specific product in the STEP system. Because no two products can have the same key, they provide an alternative method for identifying products. If STEP ID isn't available (such as with imported data), a key can be used in its place.

Unique Keys are especially useful when STEPXML is delivered to websites or ERP systems that do not know the STEP ID of the exported products and need a specific, unique ID. For more on exporting STEPXML, see the **STEPXML Format** section of the **Exporting Data** documentation.

Each key has a formula that defines what product data is used and how it is combined and transformed. See below for examples of key formulas and their resulting values.

In this example, two products have the following attributes:

Product A:

- Manufacturer Part Number (MPN): 100-440-0.750-3434-A
- Universal Product Code (UPC): 155488876964
- Brand: Acme Ltd.

Product B:

- Manufacturer Part Number (MPN): TSR-1002
- Universal Product Code (UPC): 887554522477
- Brand: Ajax

Key Formula Example:

```
concatenate(prodval('Universal Product Code (UPC)'), lower(left
(prodval('Brand'), 4)), substitute(prodval('Manufacturer Part Number
(MPN)'),'-',''))
```

In this example, the values for the Manufacturer Part Number, Universal Product Code, and Brand of the product are concatenated to form a key. The generated keys would be as follows:

155488876964acme10044007503434A (Product A)

887554522477ajaxTSR1002 (Product B)

Key Formula Example

```
concatenate(upper(left(prodval('Brand'),3)),'-', prodval('Universal
Product Code (UPC)'), '-', upper(substitute(prodval('Manufacturer Part
Number (MPN)'), '-','')))
```

Again, the values for the Manufacturer Part Number, Universal Product Code, and Brand of the product are concatenate to form a key. This time, however, the brand names are abbreviated and appear at the start of the formula, and the other two values are separated by a hyphen. The generated keys would be as follows:

ACM-155488876964-10044007503434A (Product A)

AJA-887554522477-TSR1002 (Product B)

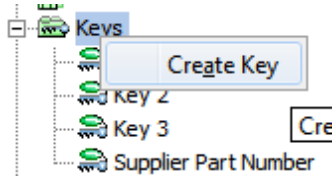
Note: If values for attributes used in the key definition are missing on products, the products are ignored and no key is generated for them.

Creating and Deleting Keys

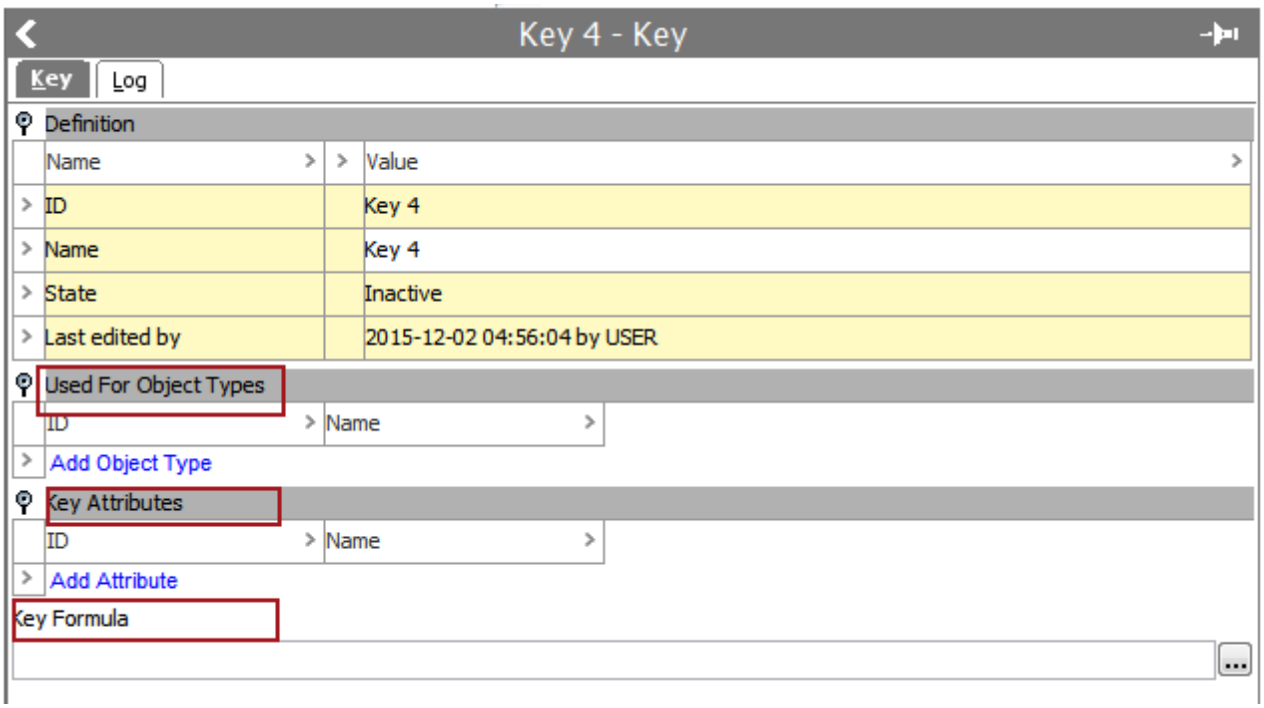
When creating a key, the user defines what product data is used and how it is combined and transformed. Note that multiple keys can be active in the system and valid for the same object type at the same time.

To Create a Key

1. In **System Setup**, right-click **Keys**, and then select **Create Key**.



2. In the **Create Key** dialog enter an **ID** and a **Name** for the key, and then click **OK**.
3. Navigate to the **Key** tab of the newly created key.



4. Under the Used For Object Type flipper, click 'Add Object Type'. Browse or search for the product object types that the key should be valid for.

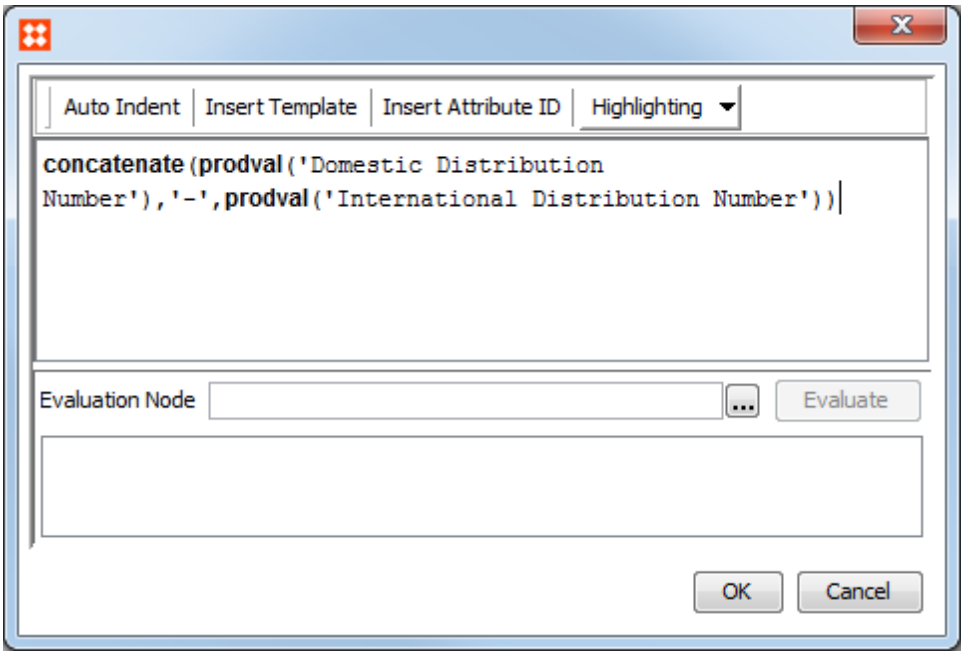
Note: The key functionality does not work with product overrides.

5. Under the Key Attributes flipper, click 'Add Attribute'. Browse or search for the attributes to be used in the key formula. The attributes must be valid for the selected object types. Because a key must be the same in all contexts and workspaces, the selected attributes must comply with the following:
   - The attributes must be externally maintained
   - The attributes can not be dimension dependent
   - The attributes must be single valued
   - The attributes can not be calculated
   - The length of the attributes must not exceed 1,000 characters
   - If the attribute has the validation base type List of Values (LOV), the LOV must not be dimension dependent and the length must not exceed 1,000 characters

Once an attribute is used in a key definition, the properties will be locked and can not be changed as long as the key definition exists. To unlock the attribute the key must be deactivated.

Form more information on deactivating keys, see the **Activating and Deactivating Keys** section of the **System Setup / STEP Superuser Guide** documentation.

6. In the 'Key Formula' field, click the ellipsis button **(…)** and define the key formula in the **Function editor** that appears. This formula defines how the product data is combined and transformed.



The following are valid functions:

- `stepid()`
- `prodval()`

- All text functions: `concatenate`, `replace`, `upper`, `lower`, `left`, and more.
- All number functions: `- +, -, *. /, round, trunc`, and more.

For more information about the Function editor, see the **Function Editor Overview** section of the **System Setup / STEP Super User Guide** documentation.

To Delete a Key

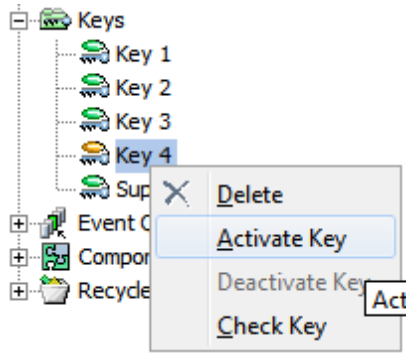
In **System Setup**, expand **Keys**, right-click the relevant key, and then choose **Delete Key**.

Note: A key must be deactivated before it can be deleted.

Activating and Deactivating Keys

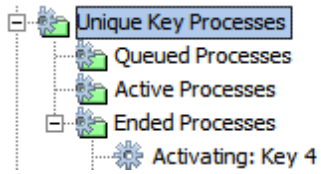
Activate a Key

Before a key can be used it must be activated. To activate a key, navigate to **System Setup**, expand **Keys**, right-click the relevant key, and then choose **Activate Key**.



A background process is started that tries to generate keys for all objects of the specified product object types. If identical keys are generated for two or more products, this is reported as errors in the Execution Report. The generated key and the conflicted products are listed in the report.

To view the Execution Report, in **Background Processes**, expand **Unique Key Processes**, and then select the relevant completed process.

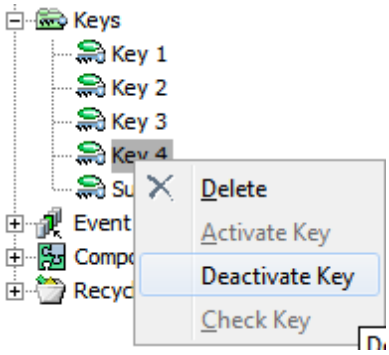


Keys are not generated for objects with missing key attribute values. Once values are supplied for all key attributes on a product, keys are generated on the fly.

When a key has been activated, it is no longer possible to edit the values on the key attributes, unless you have special set up user privileges. However, you have to deactivate the key first.

Deactivate a Key

1. In **System Setup**, expand **Keys**, right-click the relevant key, and then choose **Deactivate Key**.

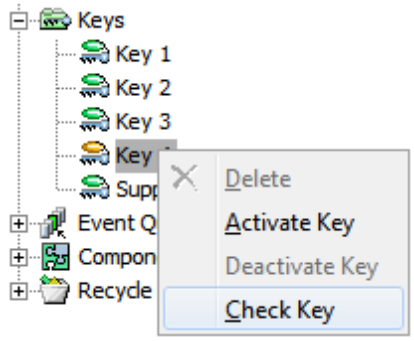


2. Make the preferred changes, and then activate the key again.

Checking Keys

You can check deactivated keys for errors.

- In **System Setup**, expand **Keys**, right-click the relevant key, and then choose **Check Key**.



A background process is started that tries to generate keys for a maximum of 5.000 objects of the specified product object types.

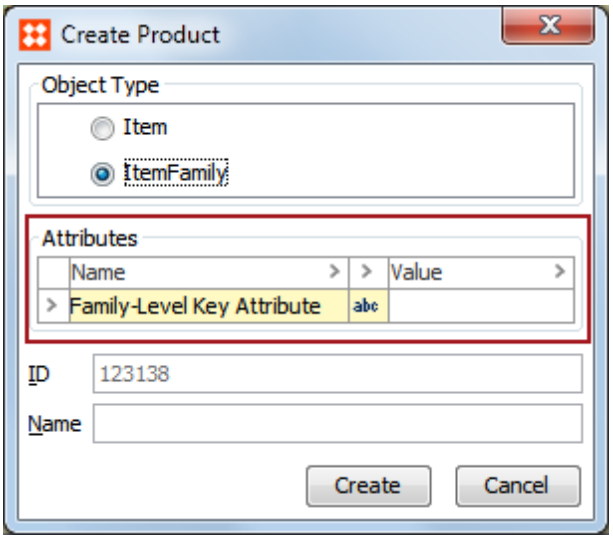
If identical keys are generated for two or more products, this is reported as errors in the Execution report. The generated key and the conflicted products are listed in the report.

- To view the Execution report, in **Background Processes**, expand **Unique Key Processes**, and then locate the relevant completed process.

If there are any conflicts, redefine the key or modify the product data.

Creating Products with Active Keys

When you create new products of an object type for which active keys exist, the **Create Product** dialog includes editors for supplying values for key attributes.



When you have entered all values, STEP tries to generate a key for the product. If this fails because the key is already being used by another product, a dialog is displayed with information about the problem.

When a product of an object type for which an active keys exists is imported, a key is generated provided that all key attributes are populated. Any key conflicts are reported in the Execution report , and the new product will not be created.

Matching Unique Keys During Import

When you import updates to products of an object type with active keys and a key, you do not have to include the ID of Excel, CSV, and XML files, as long as the object type and values for all key attributes are included.

Important: Currently, the **Identify Products** step of the Import Manager wizard does not support this functionality and reports the matched products as "New".

Sample:

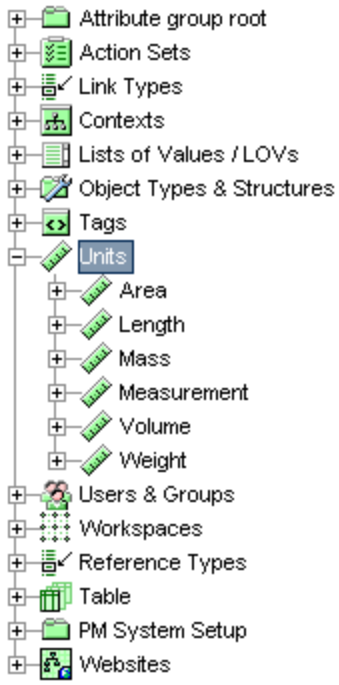
```
<STEP-ProductInformation ExportTime="2013-09-15 16:16:26" ExportContext="GL" ContextID="GL"
WorkspaceID="Main">

  <Products>
    <Product UserTypeID="KeyBProdType" ParentID="UniqueKeyImport">
      <KeyValue KeyID="KeyB">ValAvalBvalC</KeyValue>
      <Name>keyprodXMLImpChanged</Name>
      <Values>
        <Value AttributeID="UniqueKeyImportAttC">valC</Value>
        <Value AttributeID="ImportAtt">valA</Value>
        <Value AttributeID="UniqueKeyImportAttB">valB</Value>
        <Value AttributeID="UniqueKeyImportAttA">ValA</Value>
      </Values>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Units

This section describes how to setup Units.

About Units

Units are maintained in [System Setup]: **Units**.



A Unit is a measurement. A Unit is an Object that can be associated with Attributes or LOVs (List of Values). A measurement stored as an Attribute will typically consist of Value part and a Unit (or Unit of Measurement).

Example Units:

- Cm
- Meters
- Inches
- Feet
- Yards
- Celsius
- Fahrenheit

The separation of the Unit and the Value part of a measurement means following:

- The Value part can be validated according to numeric criteria
- The Value part can be searched and compared according to operators such as “>” and “<”.
- It is possible to establish automated conversion between Values of different Units of measurements.
- More presentation options, e.g. in “specifications tables”.
- Consistency can be enforced in the use of Units.

Note: Units can be applied to Attributes of all validation base types except: Text, Date, Isodatetime, and LOV.

Unit Groups

A Unit Group is a grouping of Units that are compatible. It contains Units that apply to the same measurement. It is thus possible to establish rules for conversion between Units within the group.

Unit Groups are available for:

- Attributes
- List of Values (LOV)

Examples of Unit Groups and Units:

Unit Group	Units
Temperature	• Celsius • Fahrenheit • Kelvin
Dimension	• mm • cm • meters • inches • feet • yards
Volume	• cm3 • m3 • inch3

Example: If Unit Group **Dimension** has been linked to the Attribute **RopeLength**, then the Units **mm**, **cm**, **meters**, **inches**, **feet**, and **yards** can be applied to the Value of **RopeLength**.

Unit Conversions

It is possible to define a conversion rule between two compatible Units. Unit conversions are used when e.g. exporting data, and when searching for Attribute Values.

Examples of Unit conversions:

Description	ID	Name	Conversion
Time (s):			
Second	unece.unit.SEC	s	
Minute	unece.unit.MIN	min	(1 min = 60 s)
Hour	unece.unit.HUR	h	(1 h = 3600 s)
Day	unece.unit.DAY	d	(1 d = 86400 s)

Note: The system will normally include a pre-defined set of Units according to UN/ECE standards. New Units may be added if this standard does not provide a suitable Unit. However, please inspect the available Units carefully before doing so. The pre-defined Units are also provided with standard Unit conversions

For more information, see "Units" on page 451.

Example of Conversion Factor and Conversion Offset

Unit Conversion: A Unit conversion specifies the rules for conversion from one Unit to another. One of the Units has to be a base Unit.

The conversion is specified in terms of a conversion factor and conversion offset.

Example:

Unit	Unit Conversion factor	Conversion offset
Fahrenheit (base Unit)		
Celsius	1.8	32.0

Conversion rule for temperature Celsius into degrees Fahrenheit:

- Temperature Fahrenheit = (1.8 * Temperature Celsius) + 32

To convert a Celsius temperature of 100 degrees into degrees Fahrenheit:

- First multiply the Celsius temperature reading by 1.8 to get 180.
- Then add 32 to 180 and get 212 degrees Fahrenheit.

Note: A conversion factor and conversion offset must be typed with minimum one decimal.

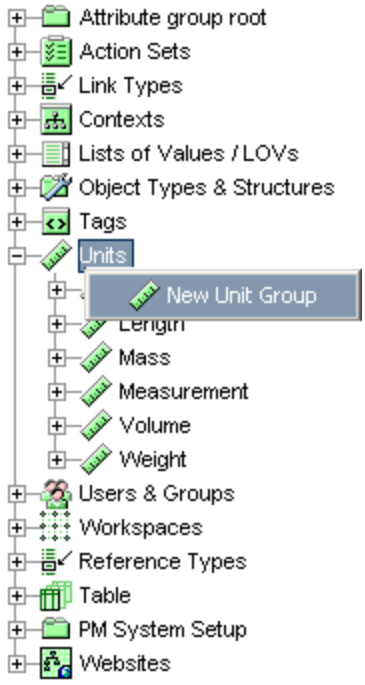
Note: The conversion factor and conversion offset should never be empty. In case one of them is “0” (zero), “0.0” has to be typed.

Maintaining Unit Groups

Creating a Unit Group

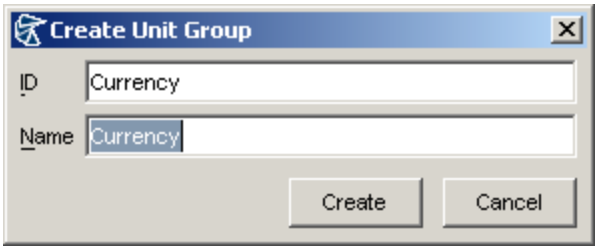
Unit Groups are created in [System Setup]: **Units**.

1. In [System Setup], click **Units**.



2. In the **Object** menu, point to **Create** and then click **Unit Group**, or right-click and click **New Unit Group**.

   A **Create Unit Group** dialog box appears.



3. Type the ID and Name of the new Unit Group.
4. Click **Create**.

A new Unit Group has now been created.

Creating a Unit

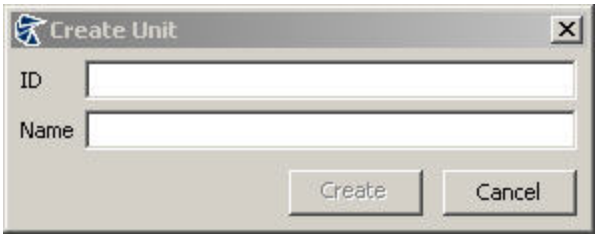
Units are created in [System Setup]: **Units**.

1. In [System Setup], unfold **Units**, and then click the relevant Unit Group.

   Note: You may have to create a new Unit Group for a new unit.

2. In the **Object** menu, point to **Create** and then click **Unit**, or right-click and click **New Unit**.

   A **Create Unit** dialog box appears.



3. Type the ID and Name of the new unit, and then click **Create**.

   The ID appears as a default name.

   The Unit Editor will display the new unit that has now been created in a group.

Note: The Unit can be selected for attributes. For more information about Attributes.

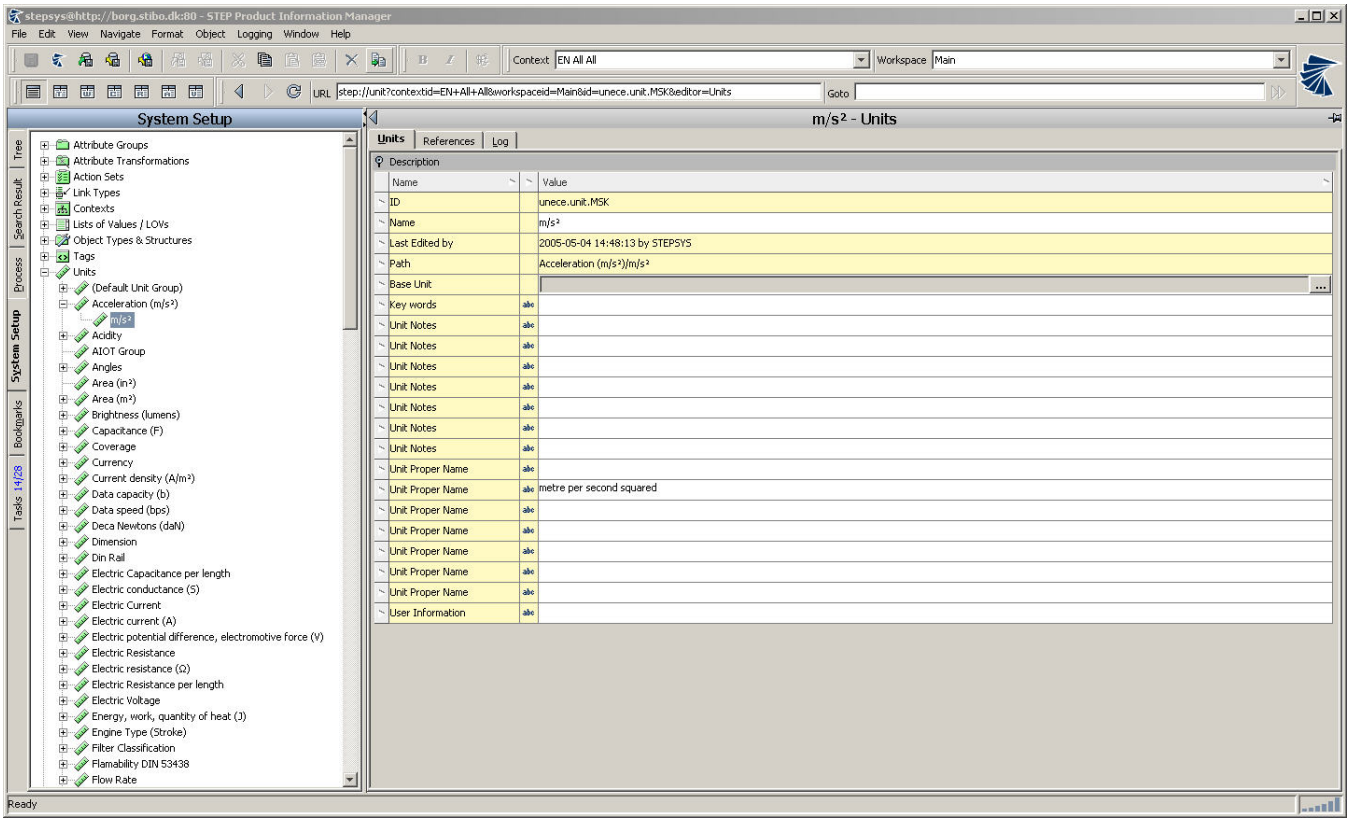
Setting up a Unit Conversion

Unit conversions are maintained in [System Setup]: **Units**.

A Unit conversion is a conversion script ('translation') from one unit to another. Standard units are defined with standard conversions.

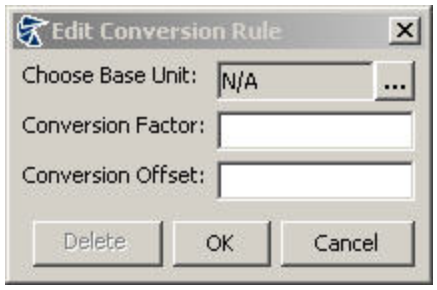
1. In [System Setup], unfold **Units**, unfold the relevant Unit Group, and then click a Unit.

   The **Unit** Editor appears.



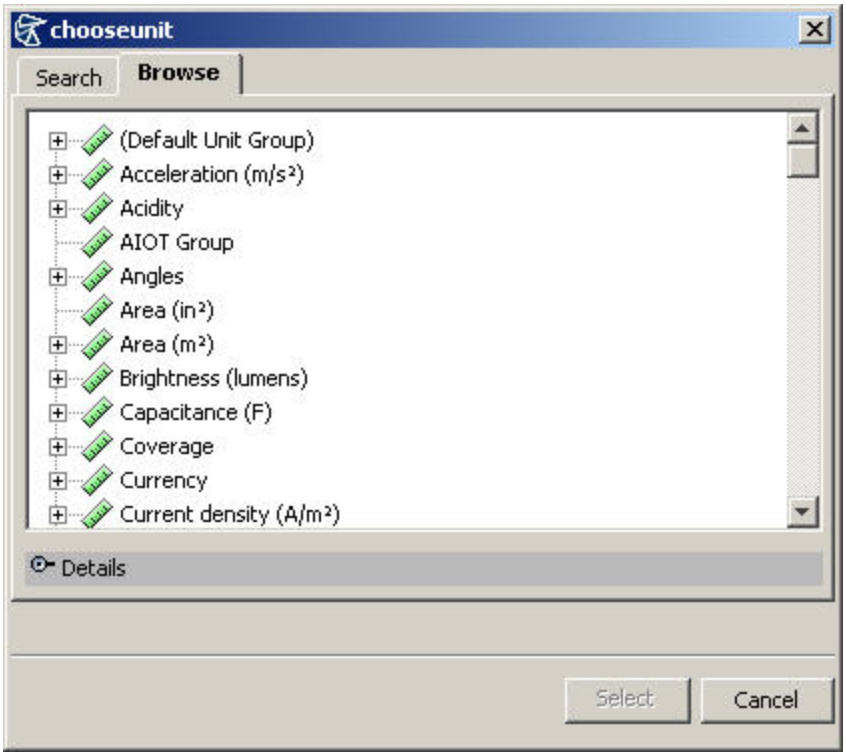
2. Click the **Units** tab.
3. In the **Base Unit** field, click the **[...]** button.

   An **Edit Conversion Rule** dialog box appears.



4. In the **Choose Base Unit**, click the **[...]** button.

   A **Choose Unit** dialog box appears.



5. Search for or browse to the relevant Unit, and then click **Select**.
6. In the **Conversion Factor** field, type the factor. (, see "Units" on page 451
7. In the **Conversion Offset** field, type the offset. (, see "Units" on page 451
8. Click **OK**.

   The conversion rule now appears in the **Base Unit** field.

   A conversion rule has now been created between two Units.

Note: There can only be one Base Unit for each group. If one Unit uses another Unit as Base, it cannot be used as Base itself.

Removing a Unit Conversion

Unit conversions are removed in [System Setup]: **Units**.

1. In [System Setup], unfold **Units**, unfold the relevant Unit Group, and then click the relevant Base Unit where an assigned conversion rule has to be removed.

   A **Unit** Editor appears, displaying a **Unit Conversions From Base** group.

2. Under **Unit Conversions From Base**, right-click on the row header of the conversion rule to be removed from Base Unit, and then click **Remove Conversion**.

   The conversion rule will now be removed from the base unit.

Removing a Unit or Unit Group

Units are deleted in [System Setup]: **Units**.

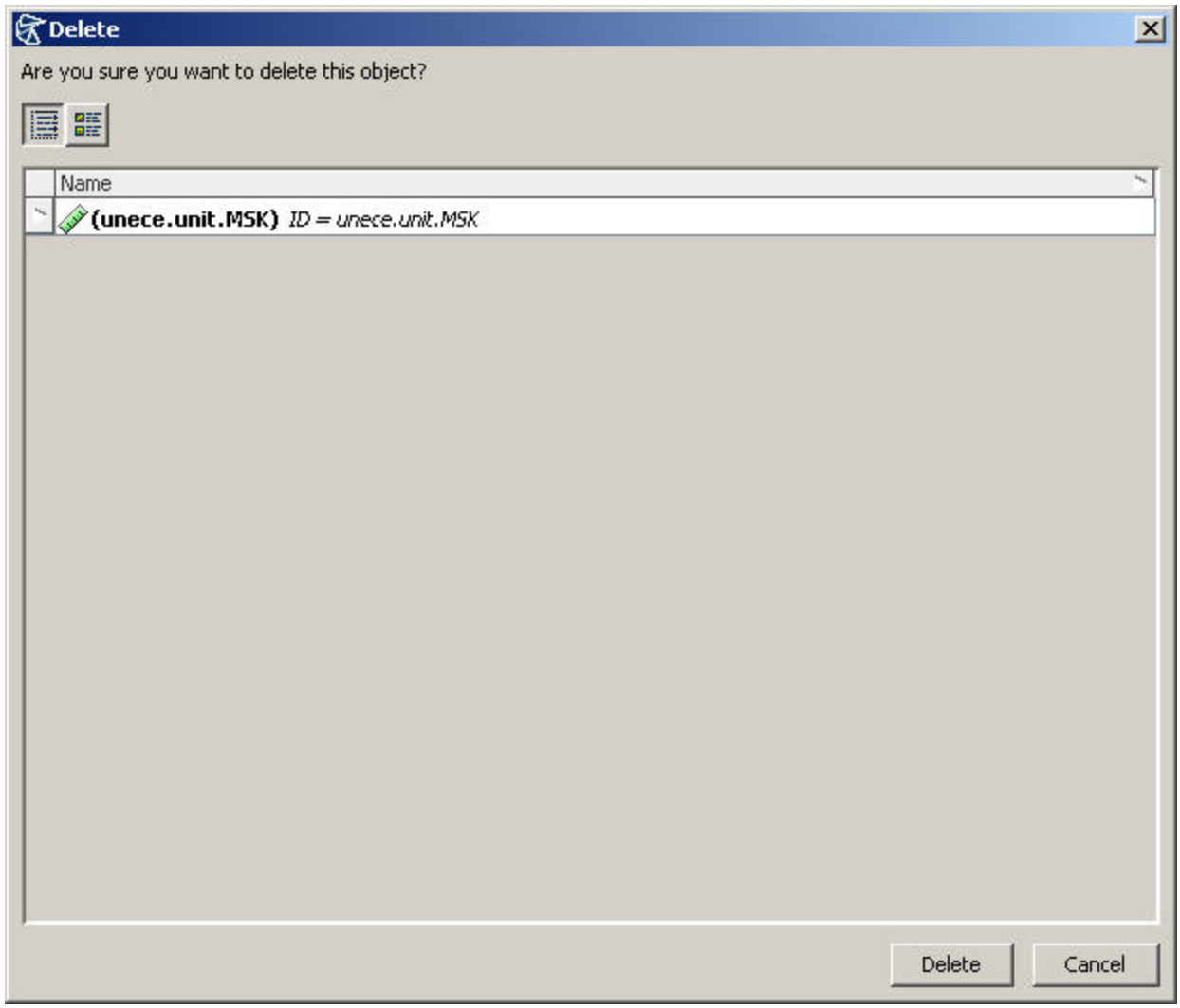
Note: A Unit Group containing a Unit cannot be deleted. All Units must be removed from the group before the Unit Group can be deleted.

Note: A Unit that is used in an attribute cannot be deleted. The Unit must be removed from the attribute before the Unit can be deleted.

---

1. In [System Setup], unfold **Units**, and then click the relevant Unit or Unit Group to be deleted.
2. In the **Object** menu, click **Delete**.

   A Delete warning appears.

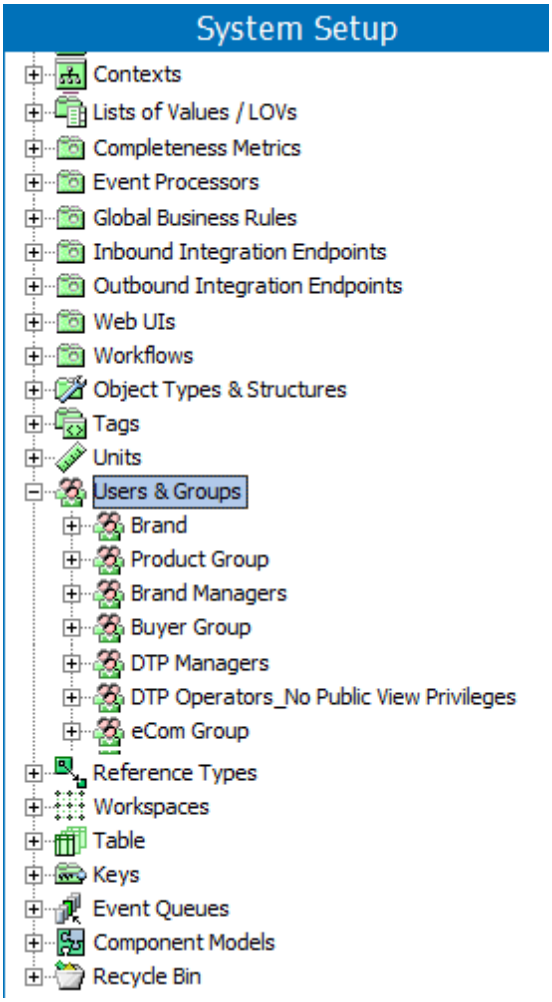


3. Click **OK** to delete.

   The Unit or Unit Group will be deleted.

Users and Groups

Users and Groups are maintained in [System Setup]: **Users & Groups**.



In the Users & Groups hierarchy, you control user permissions and general system settings. A User is an operator of the Workbench, Web UI or DTP, (Desktop Publishing), Client and is identified by unique user name. A Group is a set of one or more Users. All Users that are members of a specific Group share the Privilege rules that are applied to that Group.

A User may be a member of one or more Groups. There is no limitation to the number of Group memberships.

Note: Every User must be a member of a Group, because all permissions are obtained via the Groups.

A Group may contain an unlimited number of Users who may belong to other Groups as well.

An operator who needs to access STEP must first be created in the system as a User. The User must be assigned a user Name and a Password:

User	Description
User Name	The User Name identifies the User working in the system and allows the system to record activity performed by the User. Each user name must be unique. i.e. There cannot be any duplicate user names. User names are case insensitive and special characters are not allowed, e.g. asterisk (*) and percentage (%), etc.
Password	A user's password should be known only to the user. This helps prevent others from logging onto the system using a user name other than their own. Special characters are not allowed, such as an asterisk (*) and percentage (%), etc. Passwords are case sensitive. This password is not visible to any other user in the system. Only an Administrator User has the permissions to set a User's password. For security reasons, the password is stored encrypted in the system so that it cannot be viewed, even by an administrator. For additional information, see the **Security Policies** documentation.

Group Membership

The members of a Group ideally perform the same or very similar roles within the database environment. So the organization of Users and Groups should then reflect the responsibilities that are associated with the corresponding roles within the business.

An example of three Groups is:

- Normal User

  An operator belonging to the **Normal User** Group may typically be working with data entry and does not have the complete knowledge of the system. It thus makes sense to restrict the rights of this type of operator.

- Super User

  A member belonging to the **Super User** Group is typically an administrator who in addition to data entry also is working with system setup and higher level maintenance.

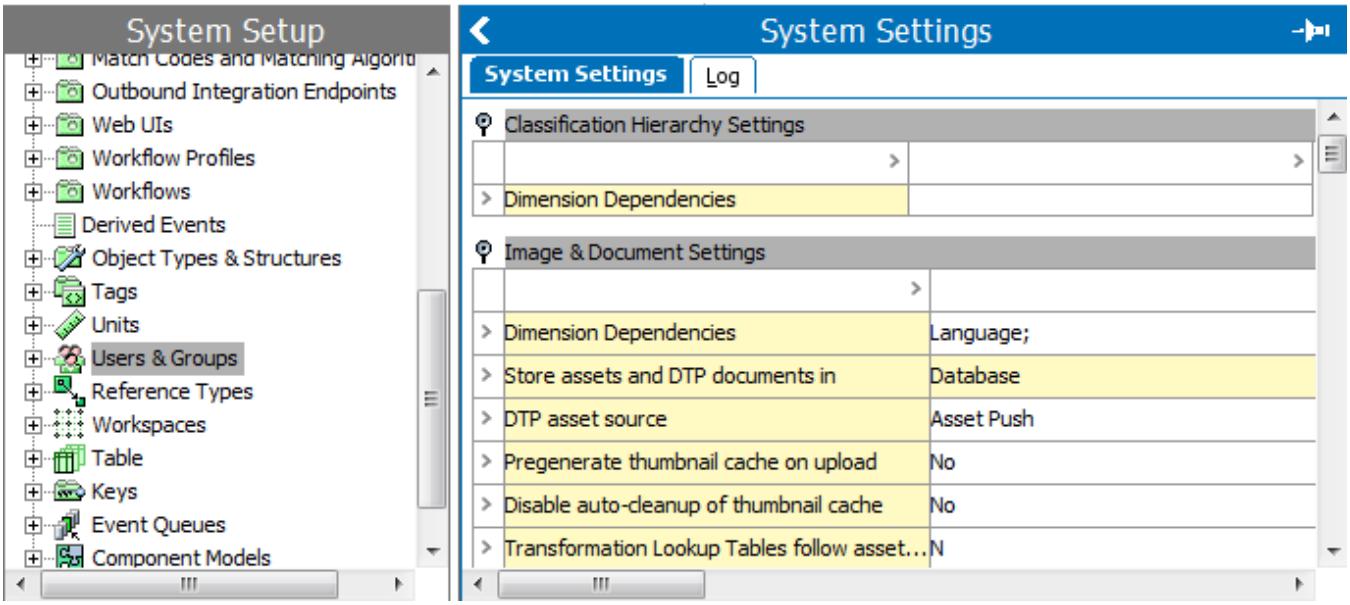
- Read-Only

  The **Read-Only** Group would typically include operators that have an interest in viewing information, but have no permissions to perform any editing operations.

About System Settings

Global system settings are maintained in the **System Setup** tab, under the **Users & Groups** root node.

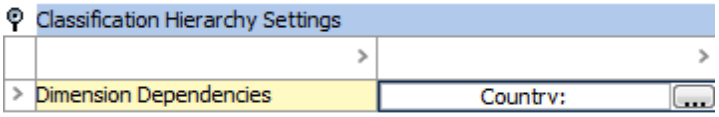
In System Setup, click **Users & Groups**. A **System Settings** editor appears. It lists a number of general and global settings on the system. Each of these settings will be described throughout this section.



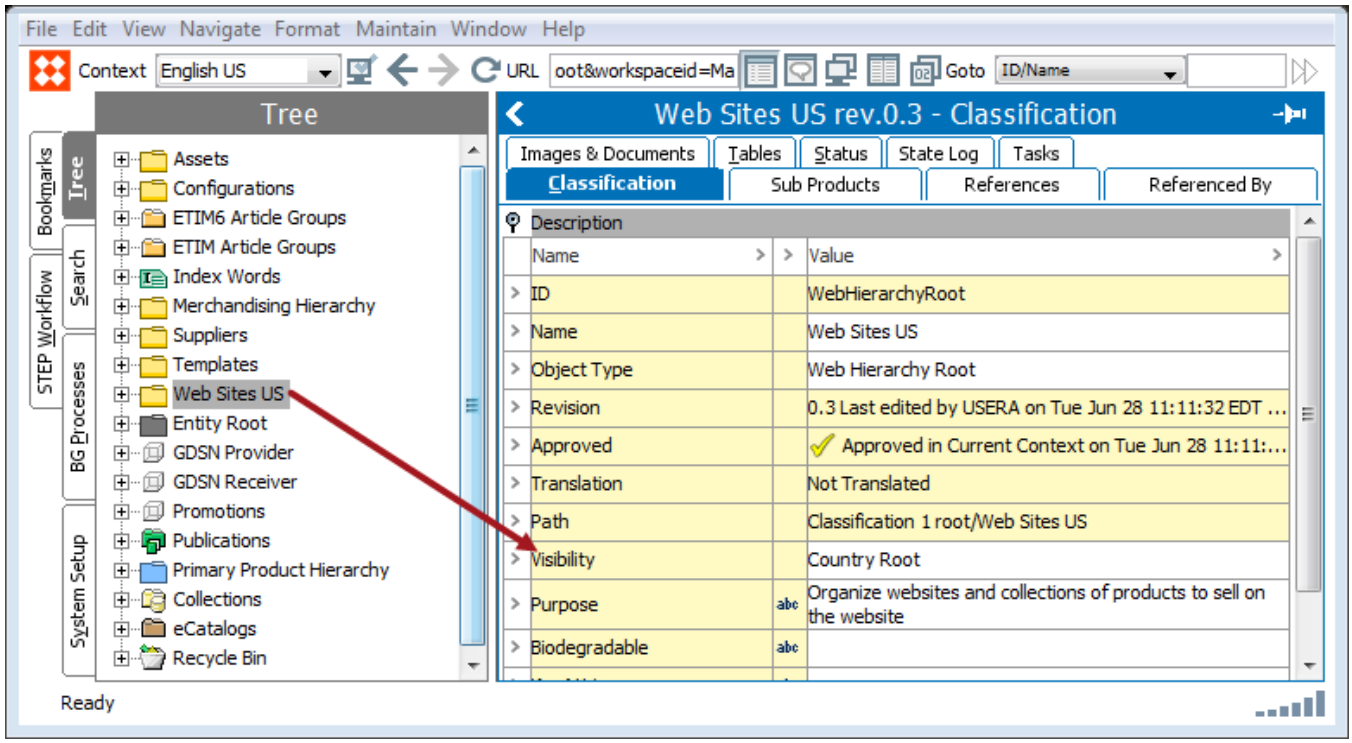
Classification Hierarchy Settings

Dimension Dependencies

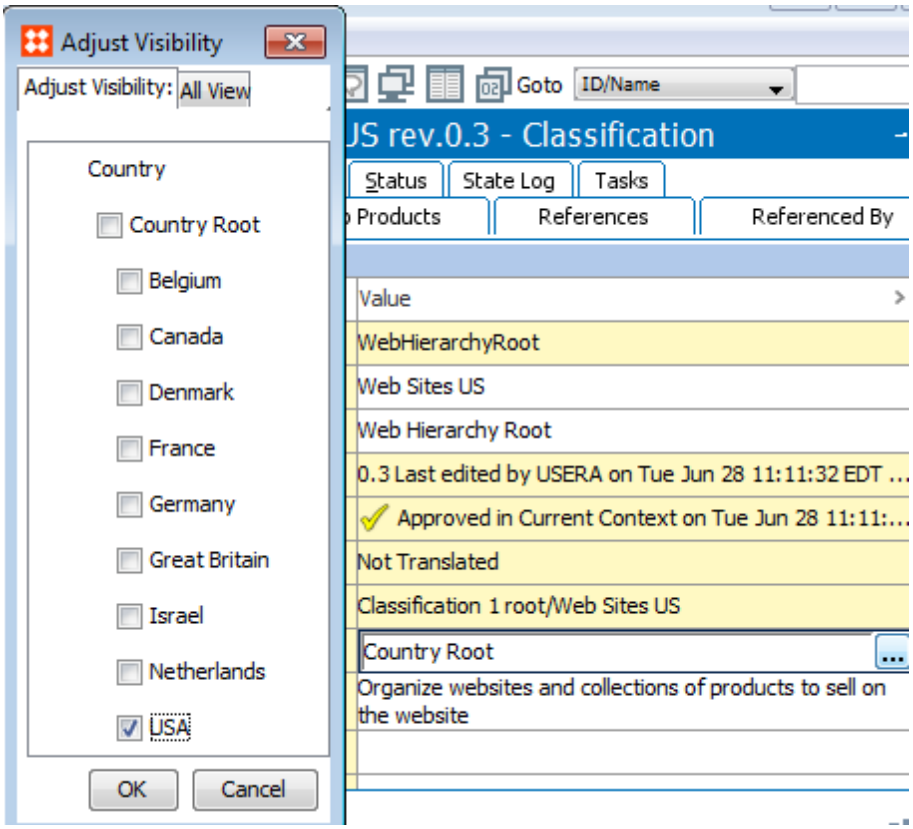
This option allows for **folders** in Classification Structures to be dimension dependent. This is different than the Dimension Dependencies that are set on the Object Type. Selecting Dimension Dependencies here will determine whether a Classification Structure(s) will or will not be shown in specific Languages or Countries. In the example below, the Dimension Dependency has been set for Country by clicking on the ellipses **[...]** button and selecting Country.



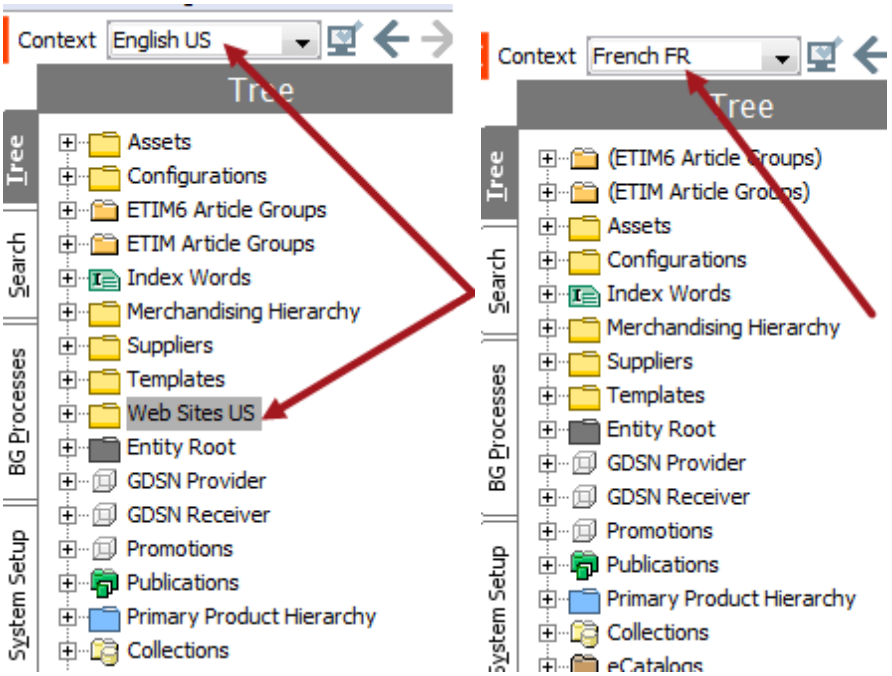
Selecting this Dimension Dependency added the Visibility metadata attribute to the Classification Structure.



Click within the Visibility row, and then click on the ellipses **[...]** button to the right of the words 'Country Root'. From the 'Adjust Visibility' dialog, select which Country or Countries your folder structure will show in. In this example, we have a Classification Structure for Web Sites US.



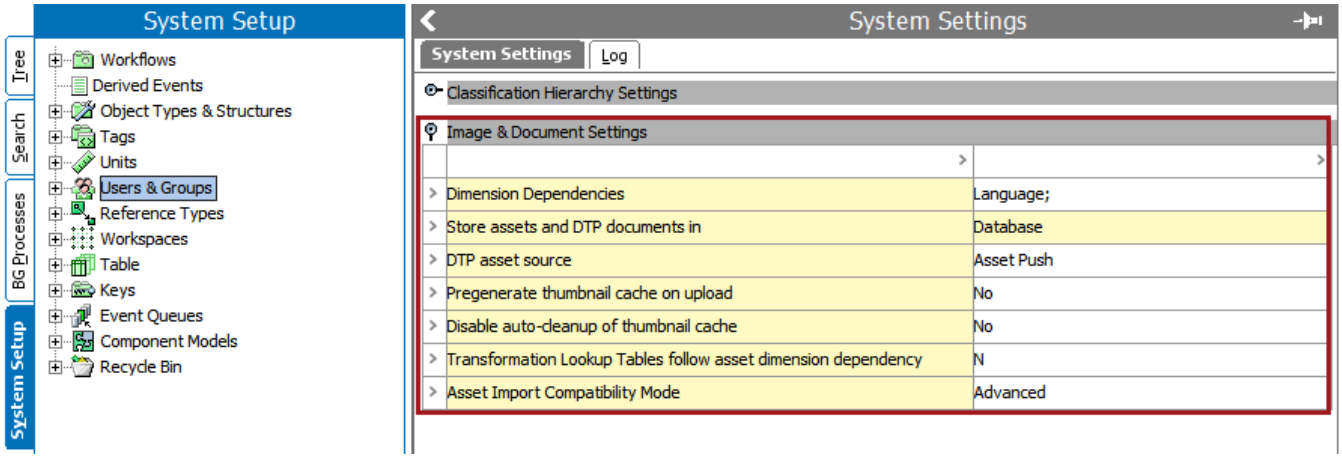
The example below shows an application where classifications have been made Country dependent, and an entire web structure has been created that is visible only in a Context where English US is the selected Country. The Web Sites Classification folder is not visible in the French FR context.



Note that it is also possible to have selected folders within an existing classification structure, not just a new one, that are also dimension dependent. It depends entirely on which Country (in this example) is populated in the currently selected Context. Therefore, it is extremely important that when a classification folder is intended to be visible in all countries that the currently selected Context specifies 'All Countries' as the Country dimension point. Otherwise it will not be visible in all Contexts.

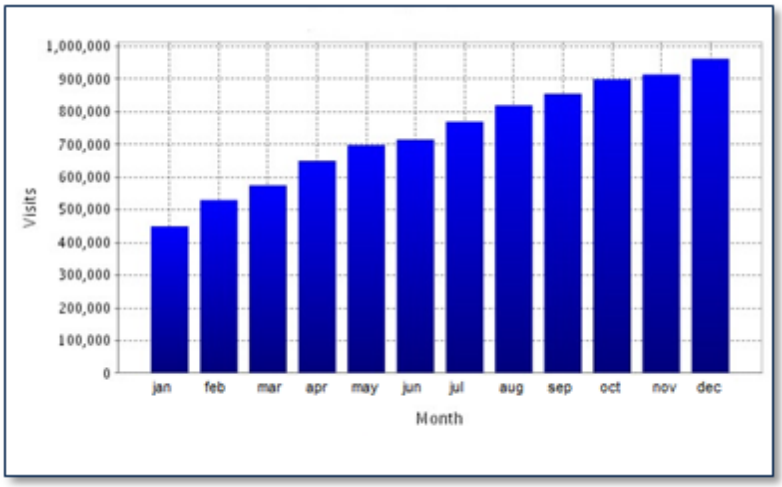
Image and Document Settings

This section is used to describe the various options available in the **System Setup** > **Users & Groups > Image & Document Settings**.



Dimension Dependencies

The Dimension Dependencies option here is used if there is a requirement to store images or documents in different contexts. For example, you may have a requirement store an image that has English labels in the image and the same image with French labels, as seen in the examples below.



To set up, click in the Dimension Dependencies value row, then click on the ellipses **[...]** button. Select the Language option. Now the image in English can be stored in the English context and the image in French can be stored in the French context.

Note: In the image examples used above, both must have the same image ID when loading into STEP and the appropriate Dimension Point must be selected for each image.

Store assets and DTP documents in

This option is used to determine where Assets and DTP (Desktop Publishing) documents are stored. There are only two options used here: Database and File System. Database is the default and stores the Assets and DTP documents to the Oracle DB, which is recommended. However, if there is a business requirement that necessitates the need to store Assets and DTP documents in a separate File System (such as the number and size of the Assets requiring an inordinate amount of time to back up), the option is available but will require further configurations and setups that are outside of STEP.

DTP asset source

This option works in tandem with the **Store assets and DTP documents in** option and the two values available are Asset Push and External Storage. Asset Push is the default option for storing Assets in the database, and External Storage is used only if it is determined that a hidden File System is required.

Pregenerate thumbnail cache on upload

Under normal circumstances, a thumbnail of an image is generated the first time that image is accessed or the first time a product is selected that contains the image as a primary image. Setting this option to 'Yes' generates the thumbnail when the image is uploaded to the system.

Disable auto-cleanup of thumbnail cache

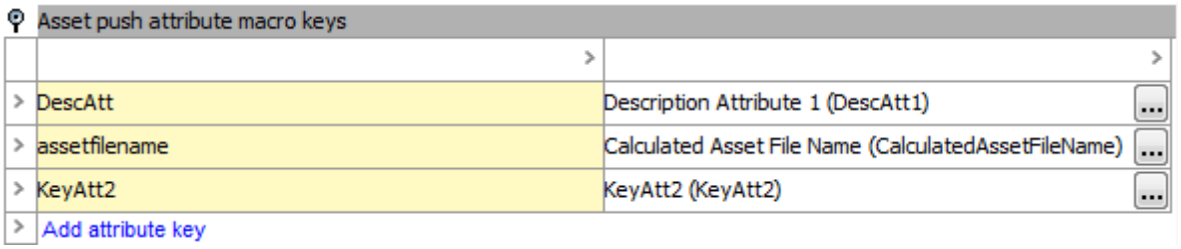
By default, the thumbnail cache in STEP is set to 100MB so that when this limit has been reached, the cache clears. By setting the option to 'Yes,' the cache will grow as large as it needs to maintain the thumbnail images.

Transformation Lookup Tables follow asset dimension dependency

A Transformation that uses a Lookup Table can now utilize a table that is dimension dependent. If set to 'Y', you could have a table that converts the words week, month, and year to wk., mo., and yr. when in the dimension of the English language, and converts the words semaine, mois, and année to the corresponding abbreviations in the French language dimension.

Note: This works only if the Dimension Dependency of the 'Lookup Tables' and the Dimension Dependency of 'Images and Documents' are set the same.

Asset push attribute macro keys



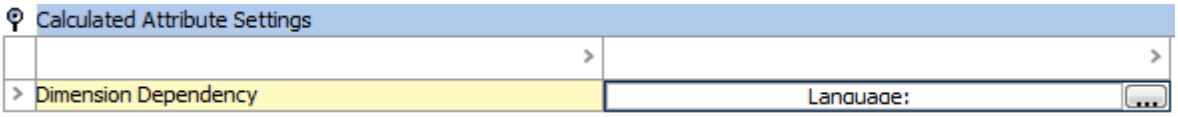
Calculated Attribute Settings

The **Calculated Attribute Settings** located in [System Setup] > **Users & Groups > Calculated Attribute Settings** allows a user to add a dimension dependency to the formula of a calculated attribute. This is useful when text elements are used within a formula. A simple example might be as follows:

English version: Concatenate(prodval('Width'), 'W x ', prodval('Depth'), 'D x ', prodval('Height'), 'H')

French version: Concatenate(prodval('Width'), 'l x ' , prodval('Depth'), 'p x ' , prodval('Height'), 'h')

In this example, the French version would show lower case letters for the words largeur (width), profondeur (depth), and hauteur (height).

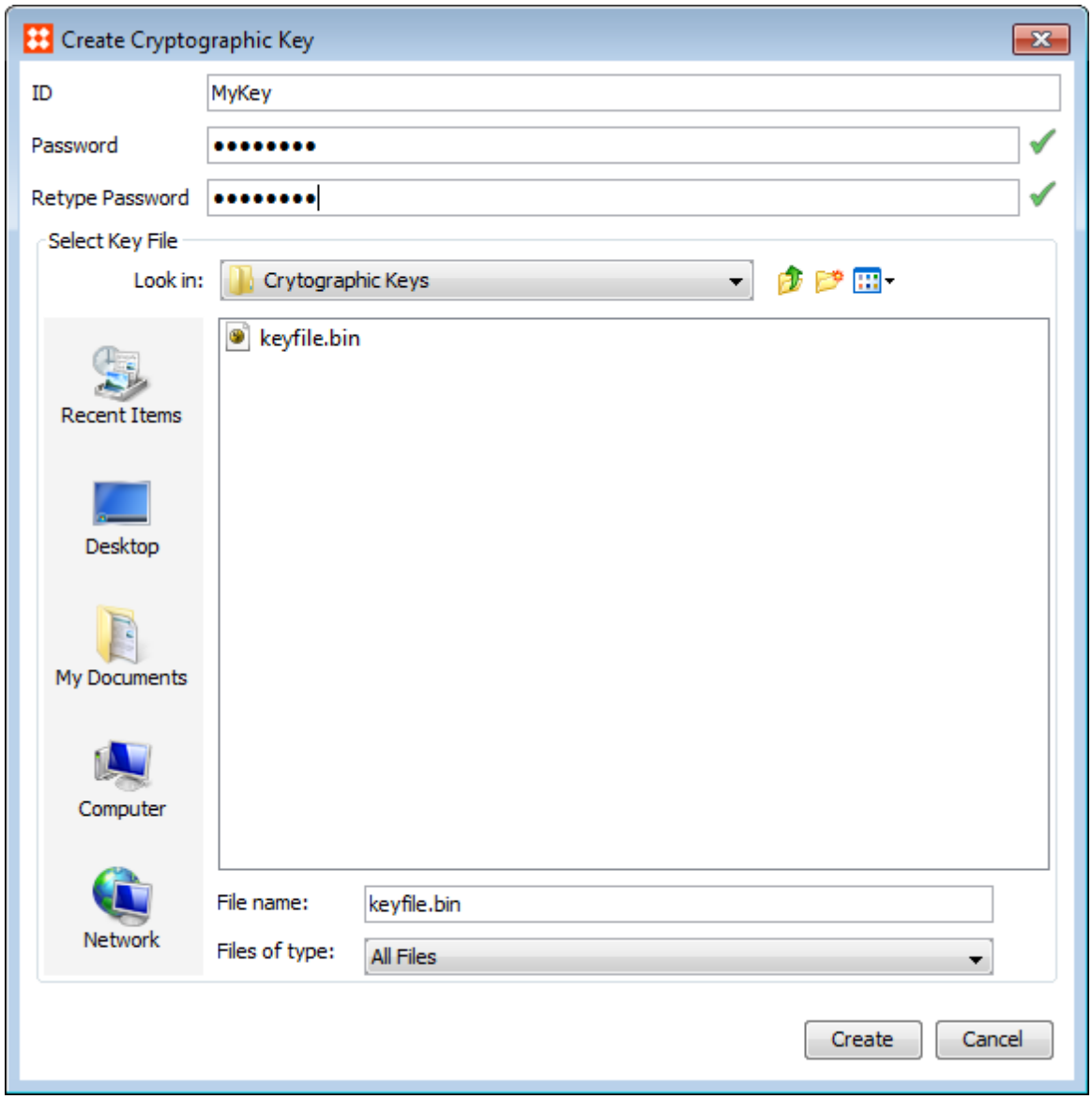


Cryptographic Keys

There may be attributes that contain data that is only to be maintained or viewed by a very limited number of users such as medical information that is HIPPA protected. It is possible to encrypt an attribute so that a password has to be entered in order to view or edit values of the attribute. However, a cryptographic key must be created before creating an encrypted attribute.

Creating cryptographic keys

Cryptographic keys are created in the System Setup on the Users & Groups node. When creating a new cryptographic key in STEP a 32 byte file with the actual key will be needed. This key file must be kept in a safe place afterward, (possibly encrypted as well), where it cannot be accessed by intruders. In addition to the key file, an ID and a password have to be supplied.



The ID will be shown to users when prompting for the password, so it would make sense to make the ID human readable. The password is needed each time attributes using this key are encrypted or decrypted. The password can be changed after a cryptographic key has been created. After changing the password, all attributes using the cryptographic key use the new password but the data / values are not modified by changing the password.

Password Requirements:

Currently the minimum requirements for passwords are 8 characters containing at least 3 of the 4 groups:

- Lower case characters a-z
- Upper case characters A-Z
- Digits 0-9
- Symbols

Note: The Setup Action **Maintain cryptographic keys** is required in order to create / delete cryptographic keys as well as changing passwords for them.

Validation Templates

The standard validations that are available when creating a new attribute (Text, Number, Integer, etc.) may be augmented by adding new validations based on Regular Expressions. These validations may be created attribute-by-attribute or a template might be created that may then be selected and re-used on many attributes that use the same regular expression pattern (e.g. for US telephone number format and re-used for Work, Home, and Mobile phone number attributes).

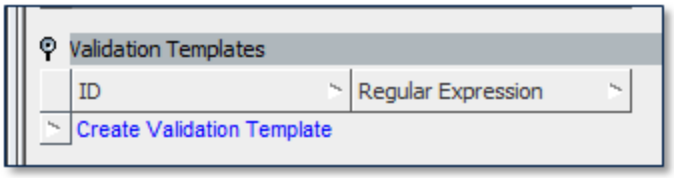
In STEP it is possible to make customized validation base types to be used on Attributes and LOVs. In this field you can create validation templates to be used when creating Attributes or LOVs.

The validation templates require a regular expression. A regular expression is a specific pattern that provides concise and flexible means to 'match', (specify and recognize), strings of text such as particular characters, words, or patterns of characters.

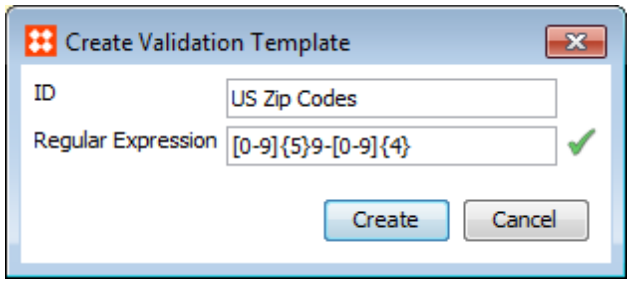
The example below shows how a new Validation Template may be created and how that new template subsequently displays in the drop-down list of validation base types when a new attribute is created (or when changing an existing attribute's validation).

To create a Validation Template:

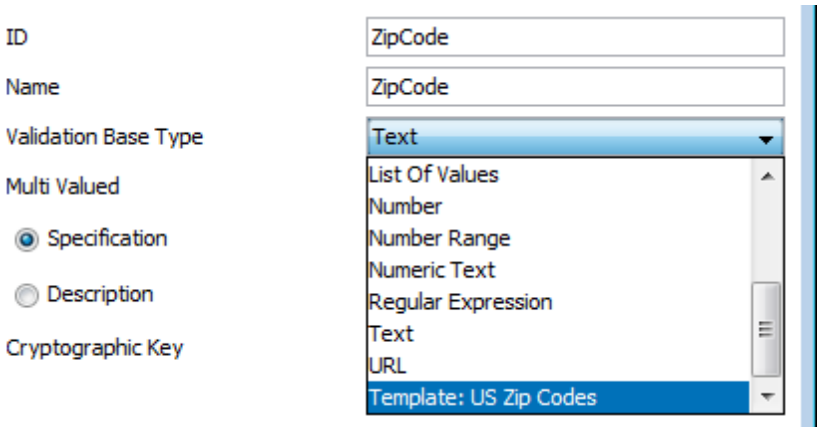
1. Click **Create Validation Template**.



2. In the **Create Validation Template** dialog, type in ID and regular expression and click **Create**.



When a Validation Template has been created, the template can be used when creating new Attributes and LOVs. The template is available among the standard validation base types in the Attribute or LOV editors.



Note: Once you have assigned an attribute to use a Validation Base Type that is a Regular Expression Template, then that template is 'copied' to that particular attribute and the link to the original template is broken. If a user decides that they wish to edit the regular expression for that attribute, they must do it at the attribute level. Changing the Regular Expression Template in System settings will have no effect.

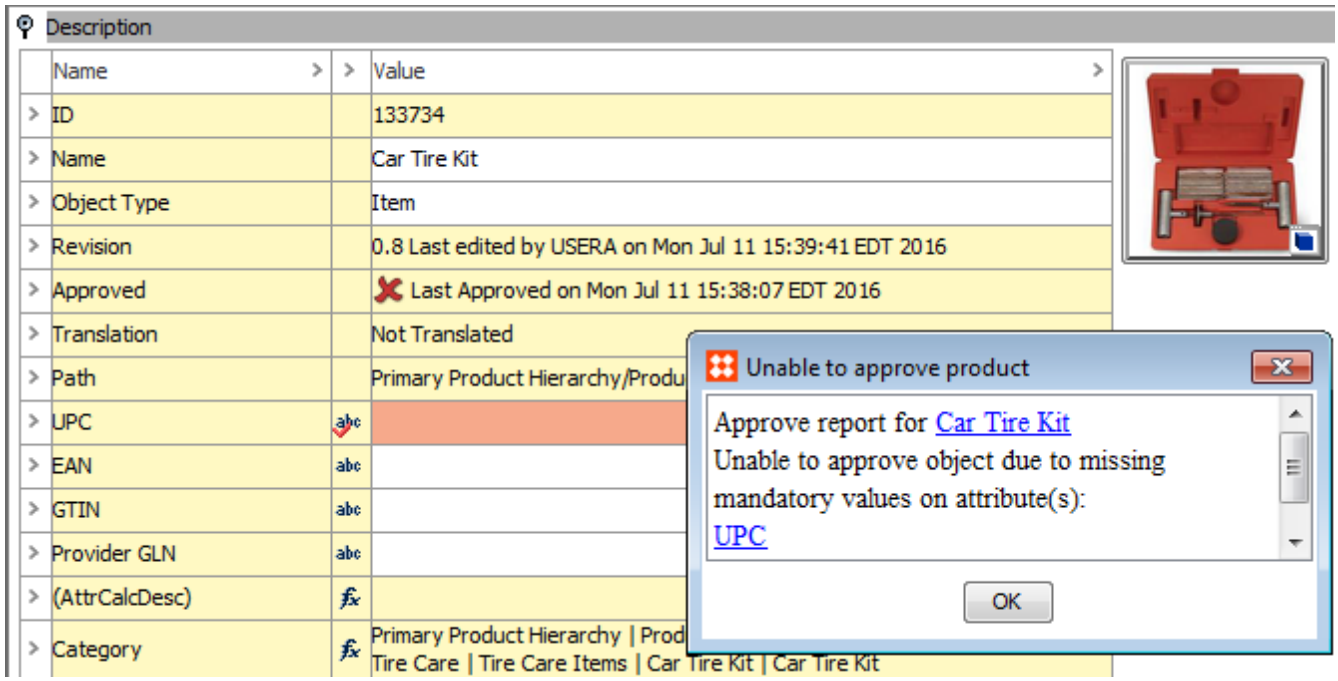
Product Information Manager Default Settings

This section is used to describe the various options available in the In [System Setup] > **Users & Groups > Product Information Manager Default Settings.**

Product Information Manager Default Settings	
Name	Value
Enforce Mandatory Check for Attributes, References and Links	none
Product Editor, Group attributes by top group	N
Localize numbers with thousand delimiter when localizing exports	Y
Localize dates when localizing exports	Y
Report Save As CSV Character Set	client-locale
Default Attribute to use as Display Sequence Attribute	DisplaySequence
Default Completeness Metric	Test
Conditional Validity Attribute	ConditionAttribute
Block Attribute Groups with more than 1000 attributes	Y
Use full pathname for classes on Product References Tab	N
Pass through unconverted Special Characters and Tags (Y) or discard them on output (N)	N
Product Attribute Help metadata attribute	Attribute Description (Attribute Description)
Attribute indicating the Priority of Product Variant Attributes	Product Variant Priority
Show both name and ID in the PIM navigator	Y
Comma separated list of node types for which to show both name and ID	
Attribute Header Column Width	250
Reference Type Column Width	120

Enforce Mandatory Check for Attributes, References and Links

This is the setting that controls whether or not objects can be 'Approved' if they are missing information that has been declared as mandatory. In the image below, the attribute UPC has been set up as mandatory. If the value is set to 'none', the object can still be approved, but the missing attribute value will give a visual indication of the missing data and shows in a background of red / brown. If the value for this setting is set to 'Y', the object cannot be approved.

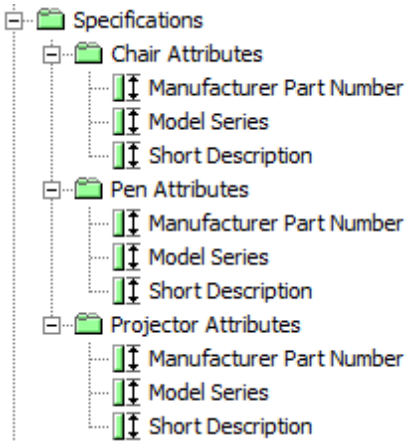


Note: This is a System-wide setting, thus it cannot be turned on and off at different places in the Tree hierarchies.

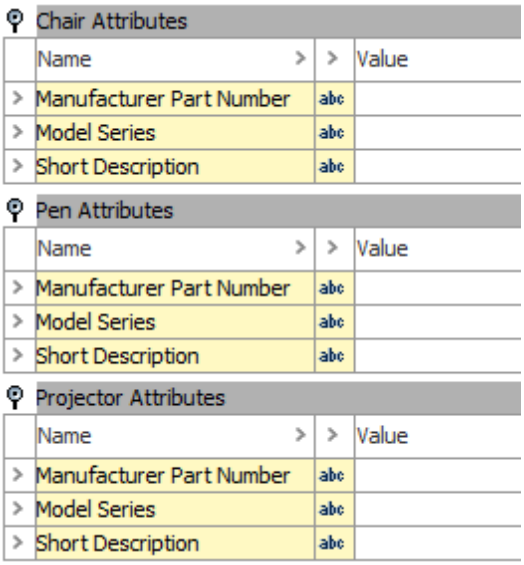
Product Editor, Group attributes by top group

There are cases where a user may create an attribute group that has multiple attribute sub-groups containing the same attributes. This can be the case when setting up User Rights and Privileges where a user can change an attribute value in one sub-group, but does not have the privilege to change the attribute value for the same attribute in the other sub-groups. This setting will control how these attributes are viewed in the workbench.

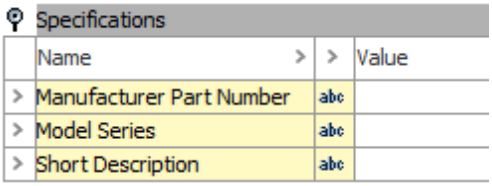
Fore example, you might have an attribute group called 'Specifications' as in the example below. Within that group, you may have several sub-groups containing the same attributes. Again, the use case here would be that the user would be responsible for the 'Chair Attributes', but would not be granted privileges to change the same attributes within the 'Pen Attributes' sub-group or the 'Projector Attributes' sub-group.



When the value for this setting is set to 'N', the attribute sub-groups would show up in workbench as in the following image. In the use case example, the user would only have privileges to change "Chair Attributes' but 'Pen Attributes' and 'Projector Attributes' would still appear.

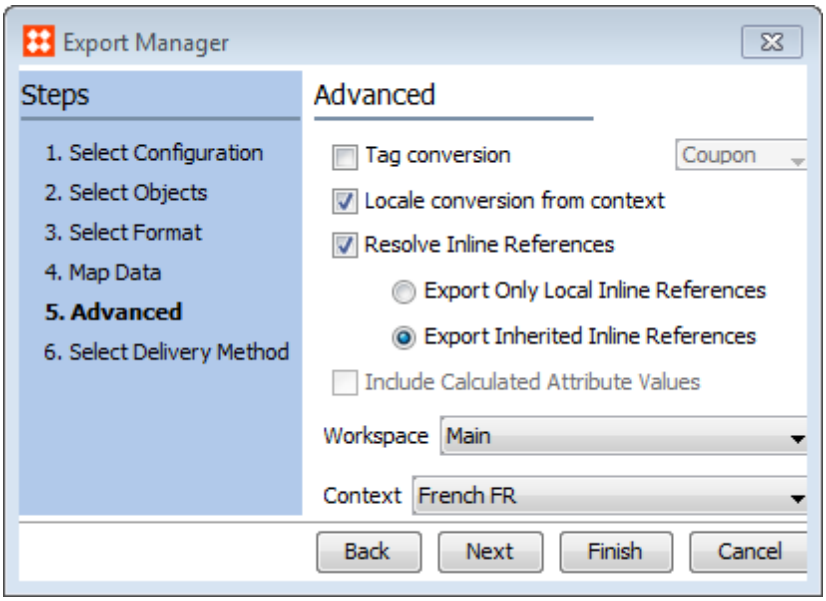


When the value for this setting is set to 'Y', the attributes would roll up into the top group. The user would still have privileges to update 'Chair Attributes' but 'Pen Attributes' and 'Projector Attributes' do not appear.



Localize numbers with thousand delimiter when localizing exports

This setting affects the way numeric values are shown for a selected context. For data exports, this setting causes the system to apply the appropriate thousand delimiter to attributes that have the validation of number, integer, fraction, or embedded number. If set to 'Y', when exporting data the currently selected context is always used. If a locale has been assigned to that context, then that locale will be suggested in the Export Manager's Advanced screen as in the image below.



This setting will also affect Merchandiser, Flatplanner, and the function localizenumber() in calculated attributes.

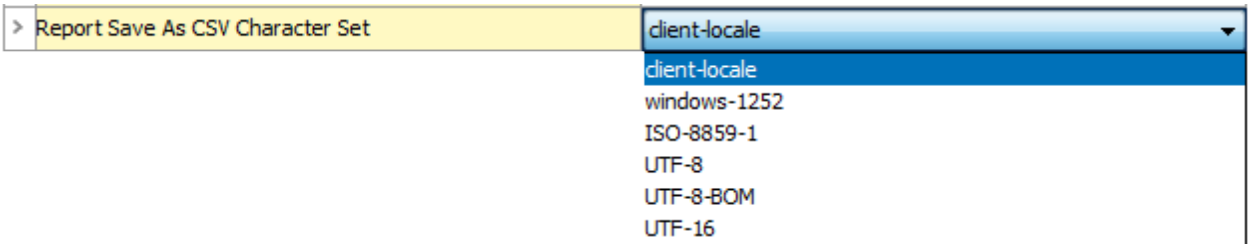
Note: Attributes with the validation of 'Numeric Texts' will not be localized with this setting as they are actually text attributes.

Localize dates when localizing exports

Set the **Localize dates when localizing exports** to Y, if you want number validated values (with base type **Date**, **Iso Date** and **Iso Date and Time**) to be converted into locale values (e.g. YY-MM-DD replaced by DD-MM-YYYY). Values are localized in STEP Flatplanner and optionally in the STEP Export Manager Advanced screen as shown in the 'Localize numbers with thousand delimiter when localizing exports' section.

Report Save As CSV Character Set

Reports can be saved in various character formats. In this drop-down you can choose the character format to be used when saving reports as CSV. The available choices are: Client-locale, Windows-1252, ISO-8859-1, UTF-8, UTF-8-BOM, UTF-16.



Default Attribute to use as Display Sequence Attribute

By default, attributes in attribute groups are shown in alphabetical order when viewed in the workbench or Web UI as shown in the example below. This setting allows the user to select the meta-data attribute to use when displaying the attributes in the Product Editor to sequence the order of the attributes.

Specifications

Name	Value
Application	
Density	
Gross Weight	
HazMat	
Manufacturer Part Number	
Model Series	
Net Weight	
Plug Included	
Short Description	

To use this feature, a meta-data attribute must be created to select as the default attribute for sequencing attributes. In this example, a meta-data attribute with the ID 'Display Sequence' was created and selected as the default attribute.

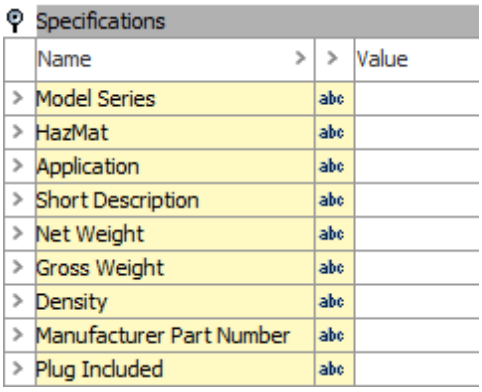
Default Attribute to use as Display Sequence Attribute	DisplaySequence

This option allows for the user to re-order how any or all of the attributes are shown within that attribute group. This can be done either in a single attribute or in the Multi-Editor as shown in the image below.

Multi Editor

ID	Name	DisplaySequence
Application	Application	30
Density	Density	70
Gross Weight	Gross Weight	60
HazMat	HazMat	20
Manufacturer Part Number	Manufacturer Part Number	80
Model Series	Model Series	10
Net Weight	Net Weight	50
Plug Included	Plug Included	90
Short Description	Short Description	40

As a result, the attributes show in the order in which they were sequenced.



Note: If some attributes have a Display Sequence and others do not, the system displays the ones that have a sequence number first, and then reverts to displaying the remainder in alphabetic order. For more information about **Attribute Display Sequence**, see the **Attribute Display Sequence** section of the **Attributes** documentation.

Default Completeness Metric

Conditional Validity Attribute

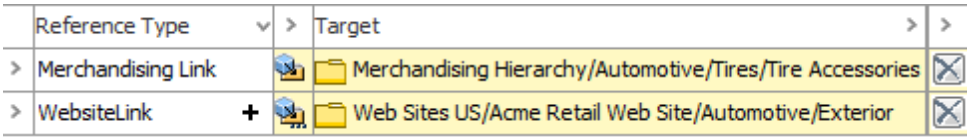
This setting is used to activate a conditional attribute. For more information about conditional attributes please see the **'Conditional Attribute Handling'** section of the **'Attributes Documentation'**.

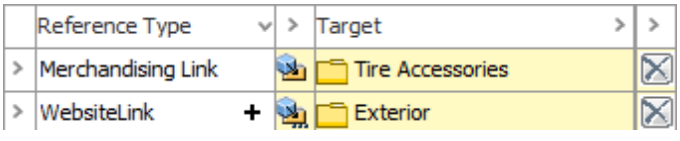
Block Attribute Groups with more than 1000 attributes

For performance reasons, it may be necessary to prevent the opening of attribute groups with more than a 1,000 entries for that group. The attributes are still accessible using the Goto function in workbench or hyper-linking to them from the Edit Panel in single-product mode.

Use full pathname for classes on Product References Tab

This setting affects how the system displays the information concerning a Product-to-Classification link. A setting of 'Y' gives the full path name of the Product-to Classification link as seen in the first image below and a setting of 'N' just gives the Classification folder name as seen in the second image.



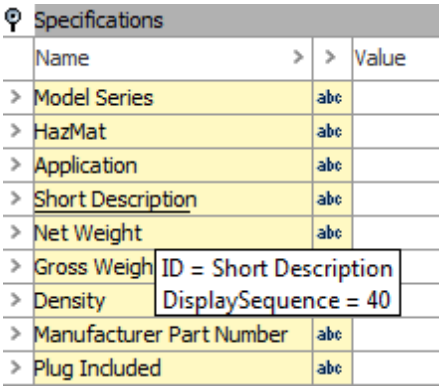


Pass through unconverted Special Characters and Tags (Y) or discard them on output (N)

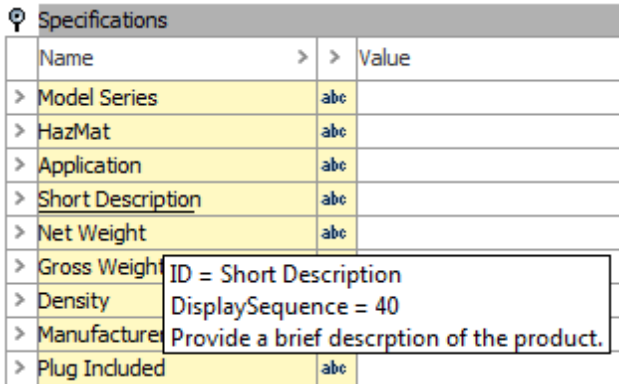
This setting is used to determine whether unconverted Tags and Style Tags should be outputted when there's not an actual conversion for that particular Tag. For example: if this option is set to 'Y', any tag that does not have a conversion in a certain media the actual 'internal' STEP tag will be output as-is. This would normally result in a web application ignoring the tag because it is not HTML, and in InDesign the incorrect command will cause an error when the information is mounted to a page. If set to 'N', when there is no output conversion for a Tag or Style Tag, the internal STEP tag should be suppressed on output.

Product Attribute Help metadata attribute

In workbench, it might be useful to display some informational text about an attribute. A use case for this might be to inform the user of the type of information that is expected for the attribute value. This option is used to select the meta-data attribute created to provide this information. Once selected, attributes containing a value for the metadata attribute will display that information when the cursor is hovering over an attribute. The following image depicts what a user would typically see without the help meta-data attribute.



However, once a meta-data attribute is created and selected in this option, the user will have additional information about the attribute. In this example, when the user hovers over the attribute 'Short Description', the user is being asked to provide a brief descrption of the product.

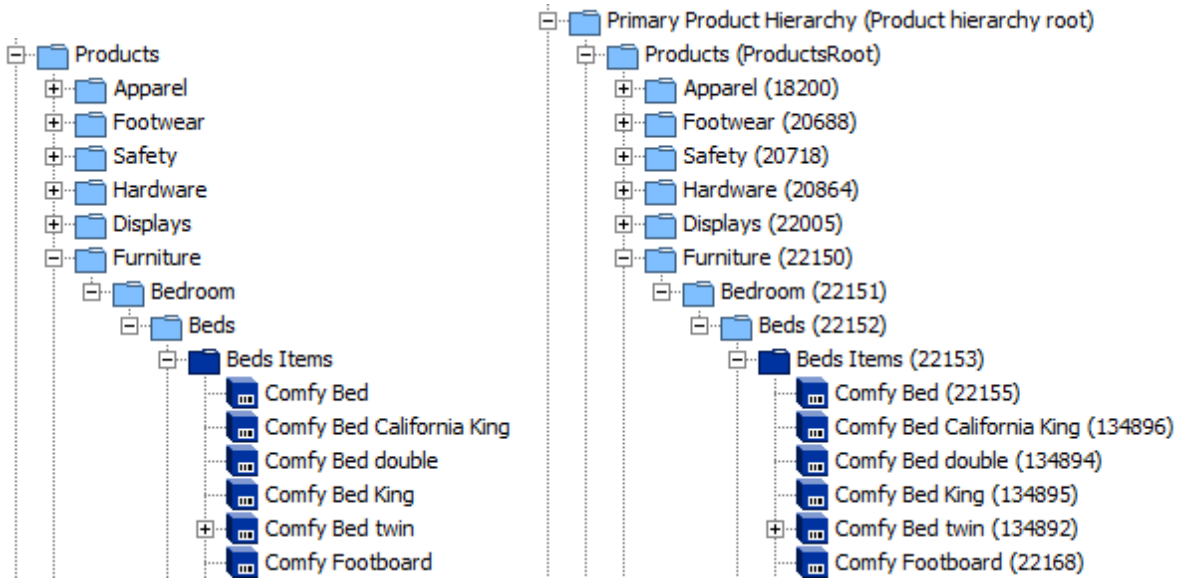


Attribute indicating the Priority of Product Variant Attributes

This setting is used to select the attribute that is to be used as a variant priority attribute. For more information regarding prioritizing product variant attributes, please see the **'Setting Up Product Variant Families'** section of **'Product Variant Families'**.

Show both name and ID in the PIM navigator

This setting works in conjunction with the **'Comma separated list of node types for which to show both name and ID'** and it determines whether just the name appears in the STEP workbench navigator panel as in the image below on the left or both the name and ID are present as in the image below on the right. The default setting is set to 'N'. However, setting the value to 'Y' will have no affect without the values that need to be entered in the **'Comma separated list of node types for which to show both name and ID'** setting.



Comma separated list of node types for which to show both name and ID

This setting determines which node types are valid for showing both Name and ID in the Navigator Panel. Node types are not object types in that they are used to identify the various elements within the STEP database. Example: p=Product, a=Attribute, 1=Classifications, m=Assets etc. The following table lists the values and corresponding node types that are valid for this setting.

Note: When selecting multiple node types, a comma must be used to separate the values.

Note: Entering values in this setting will not be affective unless a value of 'Y' is entered in the **'Show both name and ID in the PIM navigator'** setting.

Key	Node Type
D	Collections
x	Reference Types
t	Object Types
1	Classifications
p	Products
d	LOVs
a	Attributes
u	Units
m	Assets

Attribute Header Column Width

This setting affects the width of the Header Column when viewing products in the multi-products mode regardless of the rotation of the table. In the images below, the Header Column is the left most column in white. The first image has a column width of 250 and the second image has a column of 450.

	ID	Name
Name	ID	Name
20004	133737	20004
20005	133738	20005
20006	133739	20006
2007	138845	2007
20-68204	20682	20-68204
555-22346	6806	555-22346

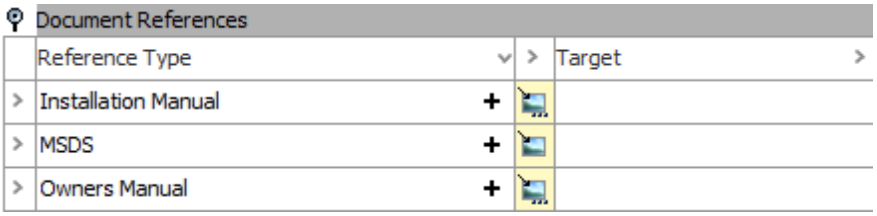
	ID	Name
Name	ID	Name
20004	133737	20004
20005	133738	20005
20006	133739	20006
2007	138845	2007
20-68204	20682	20-68204
555-22346	6806	555-22346

Reference Type Column Width

This setting affects width of the Header Column when viewing references or links. In the images below, the left most column is in white. The first image has a column width of 120 and the second image has a column of 240.

Document References

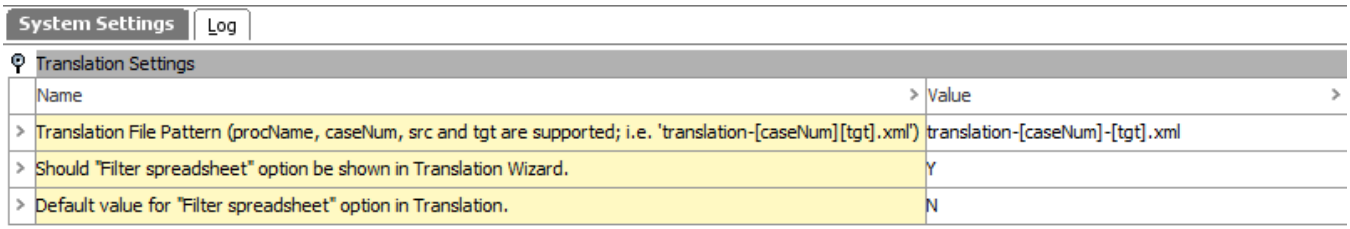
Reference Type	Target
Installation Manual	
MSDS	
Owners Manual	



Translation Settings

This section describes the options available under System Settings related to the **Request Translation** functionality (which is explained in more detail in the **Starting a Translation Workflow** section of the **Translation** documentation).

These options are available under System Setup > Users & Groups > **Translation Settings**.



Translation File Pattern

The default file name for an XML translation file exported from STEP is 'translation.xml'. The 'Translation File Pattern' setting allows users to change the file name pattern to something a little more meaningful. There are some predefined 'macros' in STEP that enable this tailoring. These macros are: procName, caseNum, src, and tgt. The src and tgt macros pull the source and target language. procName pulls the process name for the action (which is always 'Translation'), and caseNum pulls the number of the corresponding background process.

Therefore an entry such as:

[procName]-[caseNum]-[src]-[tgt].xml

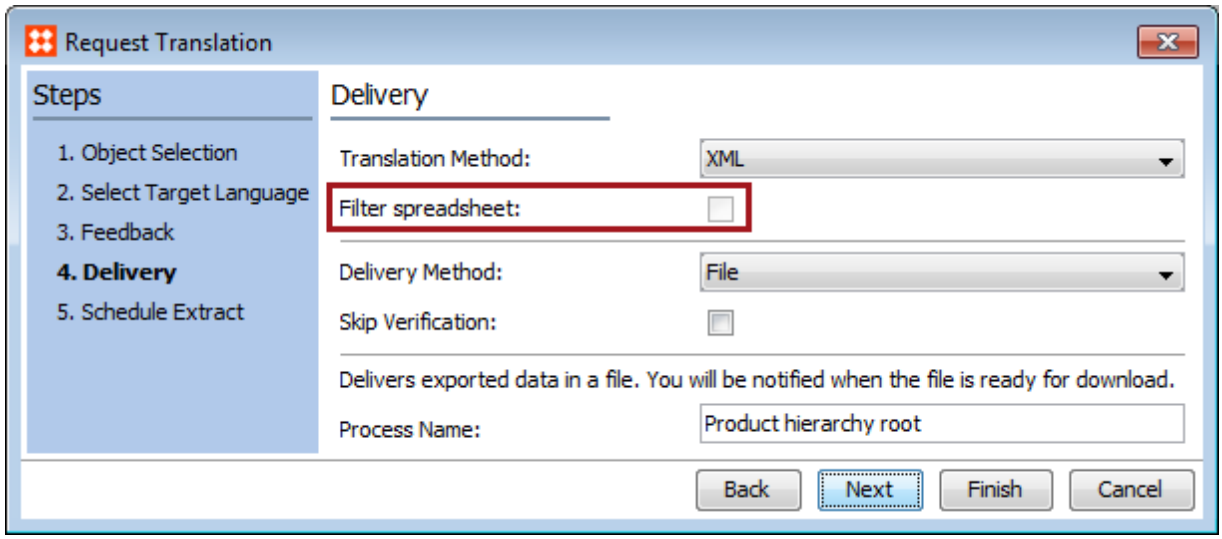
Would create a file with a name such as:

Translation-BGP_24304-US Eng-French Canadian.xml

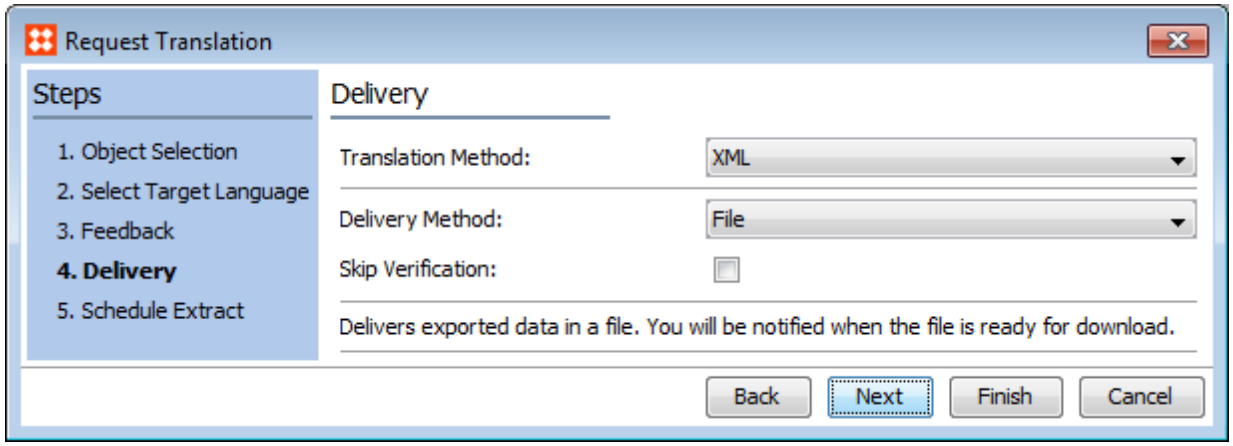
Should "Filter spreadsheet" option be shown in Translation Wizard.

The 'Should "Filter spreadsheet" option be shown in Translation Wizard.' setting provides the ability to control whether or not an XLS translation file should contain both text attribute values that need to be translated as well as text attribute values that do not need to be translated. This is accomplished by giving users the ability to filter out all text entries from the XLS translation file that do not require translation by checking a 'Filter spreadsheet:' option.

If set to **Y**, the 'Filter spreadsheet:' option will display on the 'Delivery' screen (step 4) of the 'Request Translation' wizard, which is used when starting a translation workflow. If checked, the exported XLS file will only include text entries that require translations.



If set to **N**, the 'Filter spreadsheet:' option will not be available on this screen.



This option must be set to **Y** before the 'Default value for "Filter spreadsheet" option in Translation.' setting (described next) can be enabled.

Default value for "Filter spreadsheet" option in Translation.

If set to **Y**, the 'Filter spreadsheet:' option will automatically be checked on the Delivery screen (step 4) of the 'Request Translation' wizard. However, this will only work if the 'Should "Filter spreadsheet" option be shown in Translation Wizard' setting is set to **Y**.

Terms List Settings

Add the products from the selection list import file directly instead of looking up leaf products

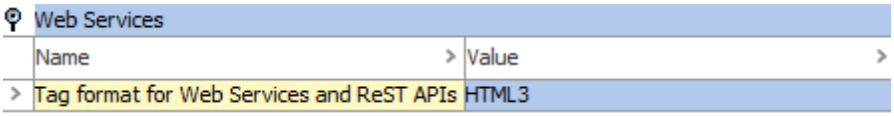
If set to 'Y' then commercial data can be stored against parent nodes. If set to 'N,' commercial data can only be stored against leaf products.

Note: This setting is only valid for eCatalogs and is now deprecated. It has no relationship to commercial terms lists used for print publishing.

Web Services

Tag format for Web Services and REST APIs

This option allows a user to state which output conversion format to use when accessing STEP via the API. The conversion format is not a dropdown list and the value must be entered manually, (e.g., HTML3), as shown in the image below.



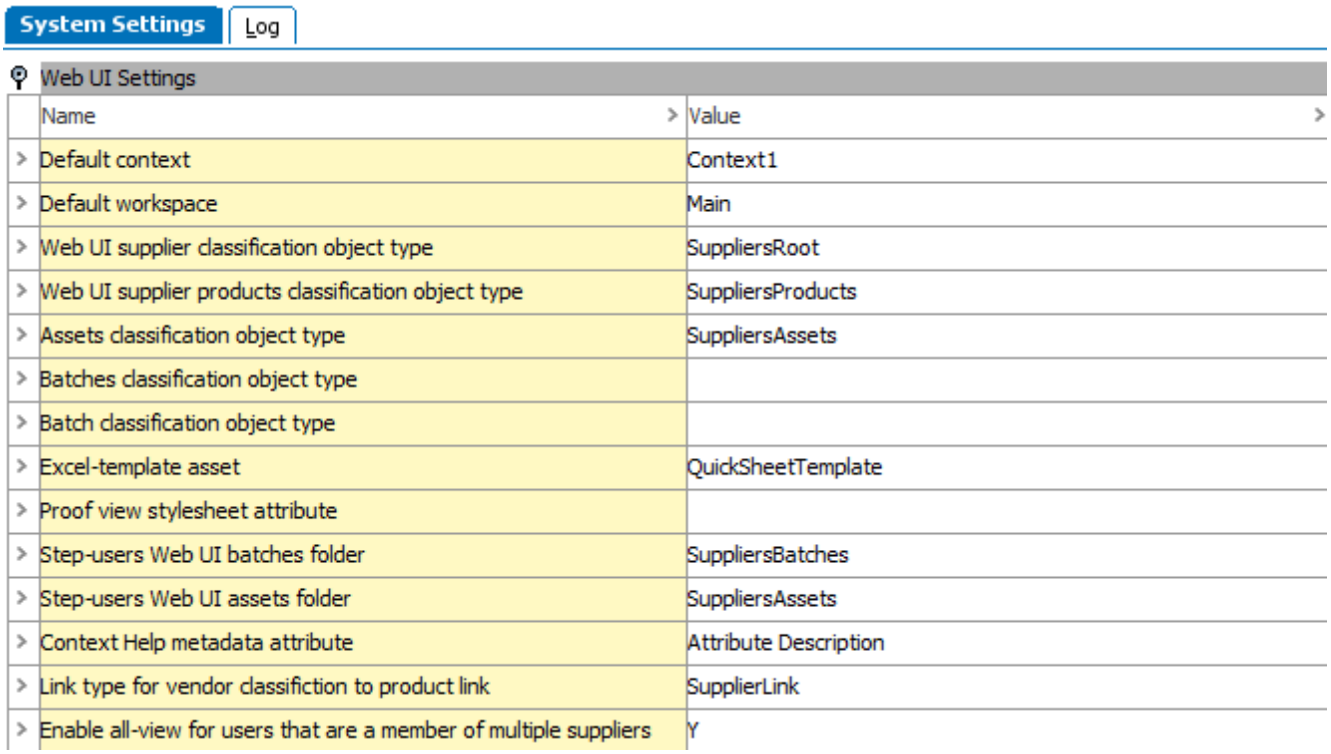
This option is only relevant for STEP systems licensed for the SOAP and REST APIs.

Note: If the purpose of using the API is to be able to edit and resubmit data to STEP, it is recommended that you do not use a conversion format, because the tag will not go through a reverse lookup on re-entry.

Web UI Settings

This section is used to describe the various options available in the **System Setup** > **Users & Groups > System Settings > Web UI Settings**.

Important: Enter the Object Type ID for the values entered in Web UI Settings and not the Name.



Default context

This setting determines the default context that will be displayed when signing into the Web UI by a user for the first time. It does not control how many contexts are available to the user once they are signed into the Web UI. This means that if the setting is set to an English context and a user wants to view an object in French, then the user can change the context to French. It will stay in the French context every time the user signs into Web UI until it is changed by the user.

Default workspace

This setting determines the default workspace that will be displayed when signing into the Web UI by a user for the first time. However, after a user signs into the Web UI, they can change the default workspace if desired.

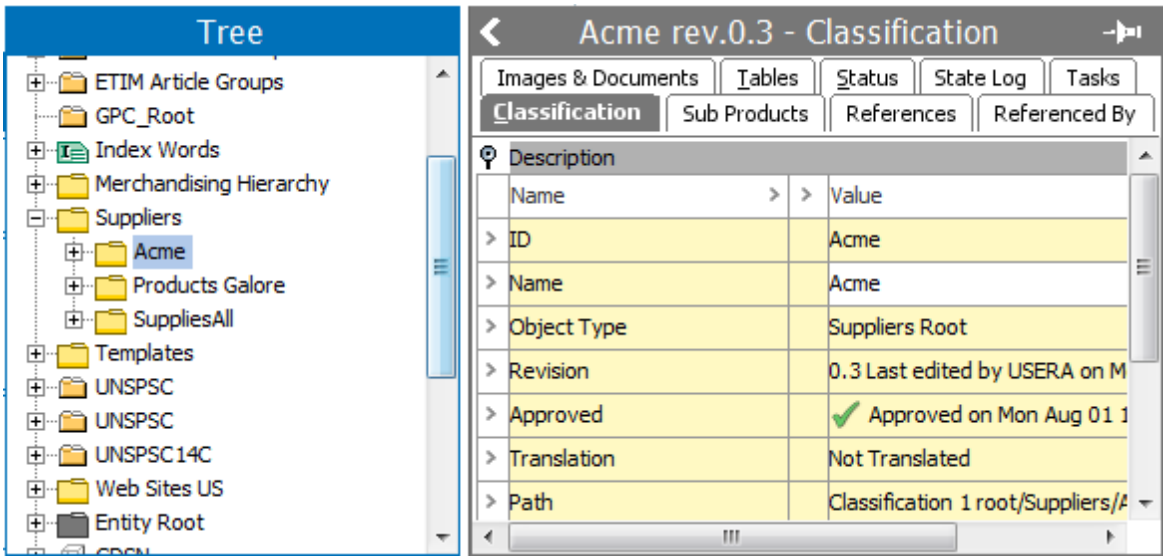
Web UI supplier classification object type

This option determines the Object Type that is 'assigned' to a Web UI user via the Supplier User Group and its associated Classification(s). Filling in this information is only required if you are setting up a Web UI that will include a supplier hierarchy. In the images below, an object type of Suppliers Root has been created. This object type will be the 'root node' for each of the suppliers that are created.

For more information regarding the setup STEP Supplier Web UI's, please see 'Setup of STEP Supplier Web UI.

Note: Enter the ID here and not the Name of the Object Type.

Web UI supplier classification object type	SuppliersRoot



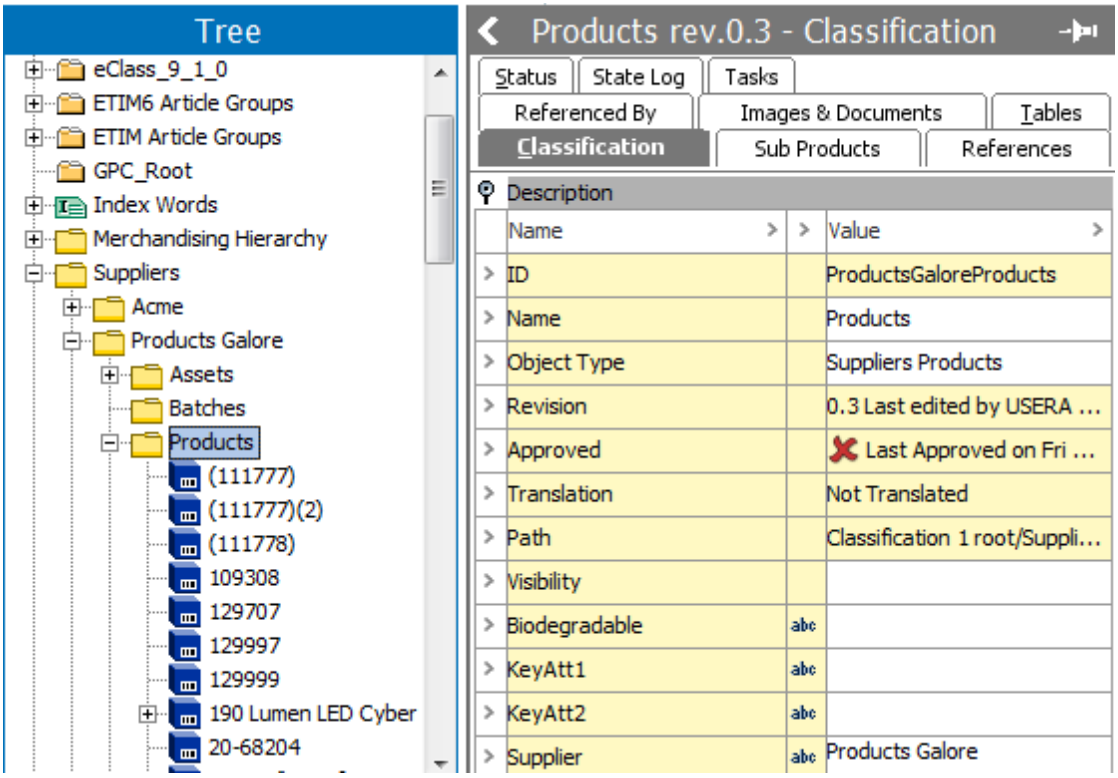
Web UI supplier products classification object type

Working in conjunction with the Web UI supplier classification object type, the Web UI supplier products classification type is the root node where the supplier's products reside as shown in the images below.

For more information regarding the setup STEP Supplier Web UI's, please see '**Setup of STEP Supplier Web UI**'.

Note: Enter the ID here and not the Name of the Object Type.

Web UI supplier products classification object type	SuppliersProducts



Assets classification object type

This is where the object type that was created for storing the Supplier's assets is configured.

For more information regarding the setup STEP Supplier Web UI's, please see 'Setup of STEP Supplier Web UI'.

Note: Enter the ID here and not the Name of the Object Type.

Batches classification object type

The Batches classification object type is the root node where the Batch classification objects reside.

For more information regarding the setup STEP Supplier Web UI's, please see 'Setup of STEP Supplier Web UI'.

Note: Enter the ID here and not the Name of the Object Type.

Batch classification object type

The Batch classification object type is the object type for the batch folders that reside under the Batches classification object type.

For more information regarding the setup STEP Supplier Web UI's, please see 'Setup of STEP Supplier Web UI'.

Note: Enter the ID here and not the Name of the Object Type.

Excel-template asset

When a Web UI user requests an 'Excel Quicksheet' export (for the purposes of bulk product initiation and/or editing), this STEP asset is accessed and downloaded to that user's system. The value in the system setting must point to an existing STEP asset ID containing an XLS template.

Proof view stylesheet attribute

The REST API proof view uses style sheets that are defined in an XML file. This XML must exist as an Asset in the STEP database.

For a standard product proof view, a default set of style sheets is utilized. If for certain products, there is a need to use a different set of style sheets, loaded as an asset to STEP, the metadata attribute specified in this option is used to point to that asset. Thus, different products may display in different styles depending on the style setups in the different XML files. The population of the metadata must have the ID of the asset and it must exist and must be a valid XML file. Otherwise the 'default' styles are used.

Note: This option refers to the REST API proof view, not the InDesign proof view.

Step-users Web UI batches folder

The value entered here is the ID of the classification folder into which an operator on the User Web UI may create batch folders and link products into them as a sort of private 'collection' of products.

Step-users Web UI assets folder

The value entered here is the ID of the classification folder into which an operator in the User Web UI may upload into or download from. It is expected that the Assets folder that is used for the uploads is really only a temporary holding area for the assets and that they will subsequently be moved into their 'proper' folder within the assets hierarchy in the STEP system.

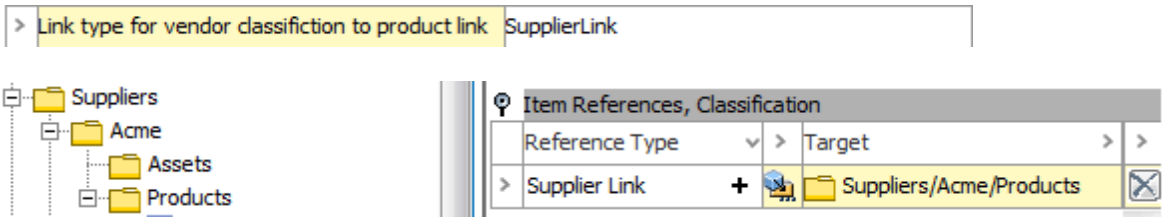
Note: The 'Step-user' referred to in this option is one that has logged in to the User Web UI.

Context Help metadata attribute

The **Context Help metadata attribute** functionality allows a user to select the description attribute that contains the help text for use in Web UI. An example of what the help text does is shown in the image below. After the attribute has been created, the ID of the attribute should be entered here. For more information about using the Context Help metadata attribute, please see **Attribute Help Text in Web UI** in the **Web User Interfaces** documentation.

Link type for vendor classification to product link

This setting is where the Reference Type that has been defined to use for products associated to a Web UI users' Supplier Products Folder is added. As shown in the image below, the Reference Type ID is entered here and will be made available for use in the Web UI.

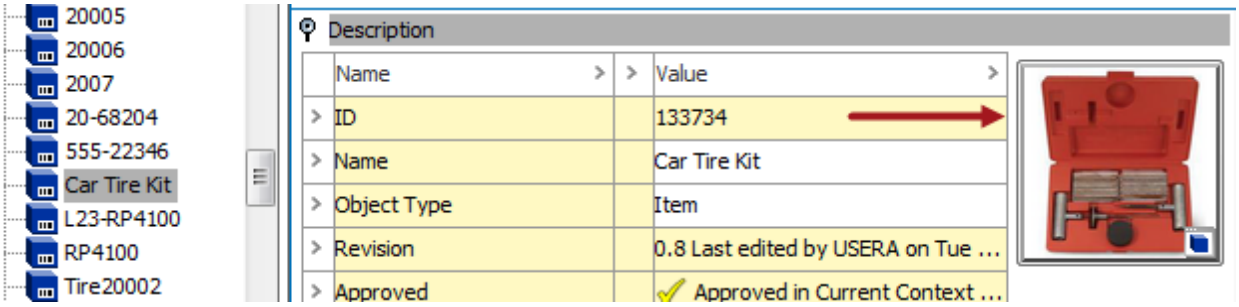


Enable all-view for users that are a member of multiple suppliers

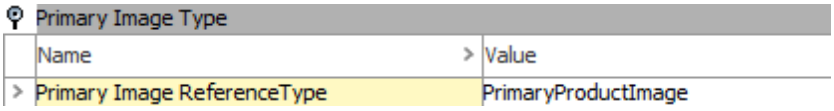
If set to 'Y,' this setting enables a supplier user that is a member of multiple supplier groups to have views across all of the groups when looking at tasks. For more information about the multiple views for suppliers, please see the '**All-View Across Multiple Suppliers**' section of **Setup of STEP Supplier Web UI**.

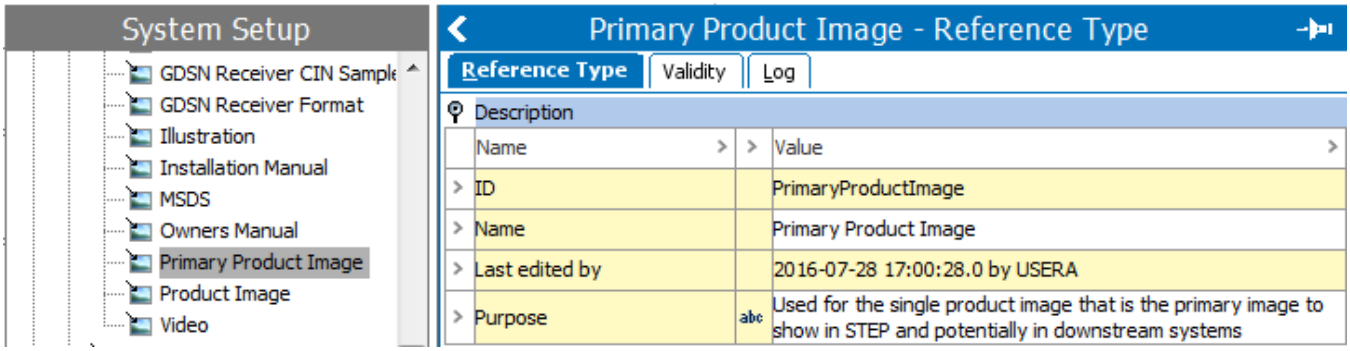
Primary Image Reference Type

The **Primary Image Reference Type** setting is used to display a thumbnail of the object image in the workbench or in the Web UI. An example of how this image looks in the workbench is shown in the image below.



By default, the **Primary Image Reference Type** value is the ID of the Reference Type, 'PrimaryProductImage.' However, if a user has created another reference type to be used to associate an image thumbnail to an object, the ID of that image type can be entered in this field and the new reference type will associate the images to the objects.





Flatplanner Default Settings

This section describes the available options under System Setup > Users & Groups > **Flatplanner Default Settings**.

Flatplanner Default Settings

Name	Value
Enable keep layouts for publications	Y
Enable calculation of border rectangle space usage ...	N
Show frame information on flatplans with layout pic...	Y

Enable keep layouts for publications

When a Flatplanner page is mounted in InDesign (using either the 'Create Document from Plan,' 'Mount planned page,' or 'Mount on current page' function), the page is created using layouts from the product templates that were assigned to objects on the page (typically product objects), either by assignment to individual Flatplanner frames or by the specification of default product templates on the objects themselves.

After pages are mounted and saved back to STEP, if 'Enable keep layouts for publications' is set to **Y**, Flatplanner will save ('keep') the layout of the mounted page and override the layouts from the original product templates. A 'kept' layout is signified on Flatplanner page frames by the text 'Keep Layout,' which appears on the frame where the name of the product template normally displays. (**Note:** View > **Show Product Templates** must be selected from the plan before the 'Keep Layout' label will appear.)

If a page or frame that is designated as 'Keep Layout' is remounted in InDesign, the page will mount with the 'kept' layout instead of the layout of the original product templates. This feature is handy because it is common for users to make manual changes to mounted pages before saving them back into STEP. The ability to 'keep' manually changed page layouts enables the reuse of these layouts, either when the page is remounted in InDesign or 'picked up' for a different publication.

If 'Enable keep layouts for publications' is set to **N**, users are free to save manually adjusted pages back to STEP, but the next time a page is mounted from the corresponding Flatplan, it will mount with the layout from the original product template(s).

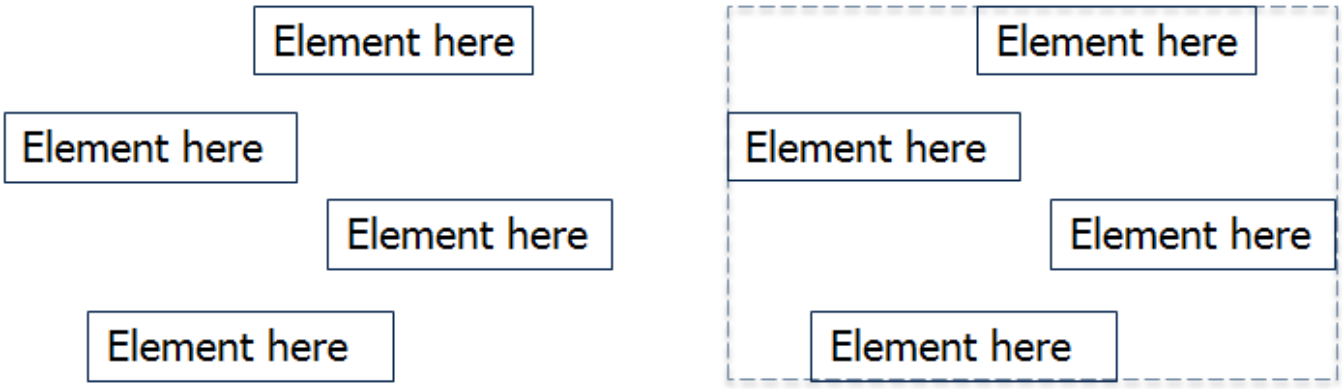
(Note that a 'kept' layout will be removed from the Flatplan if the corresponding mounted [actual] pages are deleted from STEP.)

For more information, see **Reusing Saved Layouts** in the **Flatplanner** documentation.

Enable calculation of border rectangle space usage for actual pages

When viewing actual pages in STEP, the space usage for mounted product and classification objects can be viewed on the **Page data** tab. The system setting 'Enable calculation of border rectangle space usage for actual pages' tells the system whether to calculate the space that is taken up by each individual mounted element (product, classification, or image object) on the page, or if the white space surrounding these elements should also be calculated. Though this setting is an option beneath the Flatplanner settings, it is valid for any STEP publishing license (that is, it will also work with STEP'n'design and AutoPage).

When 'Enable calculation of border rectangle space usage for actual pages' is set to **Y**, the system draws an imaginary bounding box (border rectangle) around all of the elements and takes the total area from that bounding box's width and height, including white space.



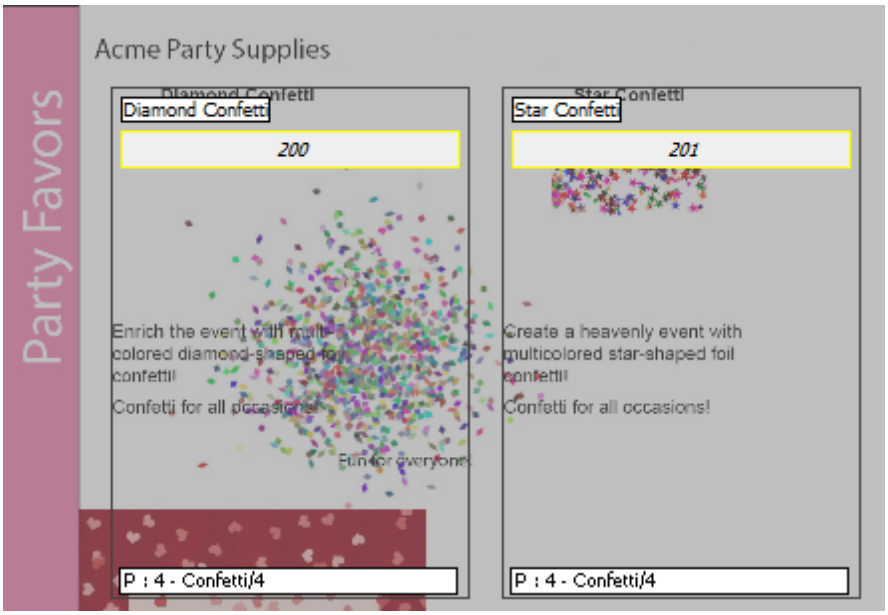
The above example shows the difference between the two methods of measuring space usage. On the left, each of the four elements' individual areas will be used to create the total space usage. On the right, the bounding box is the smallest box that could be drawn to encompass all the individual elements. It is the area of this imaginary box that shows as the space usage.

If 'Enable calculation of border rectangle space usage for actual pages' is set to **N**, the space calculation includes the total area of the individual elements only and does not include white space.

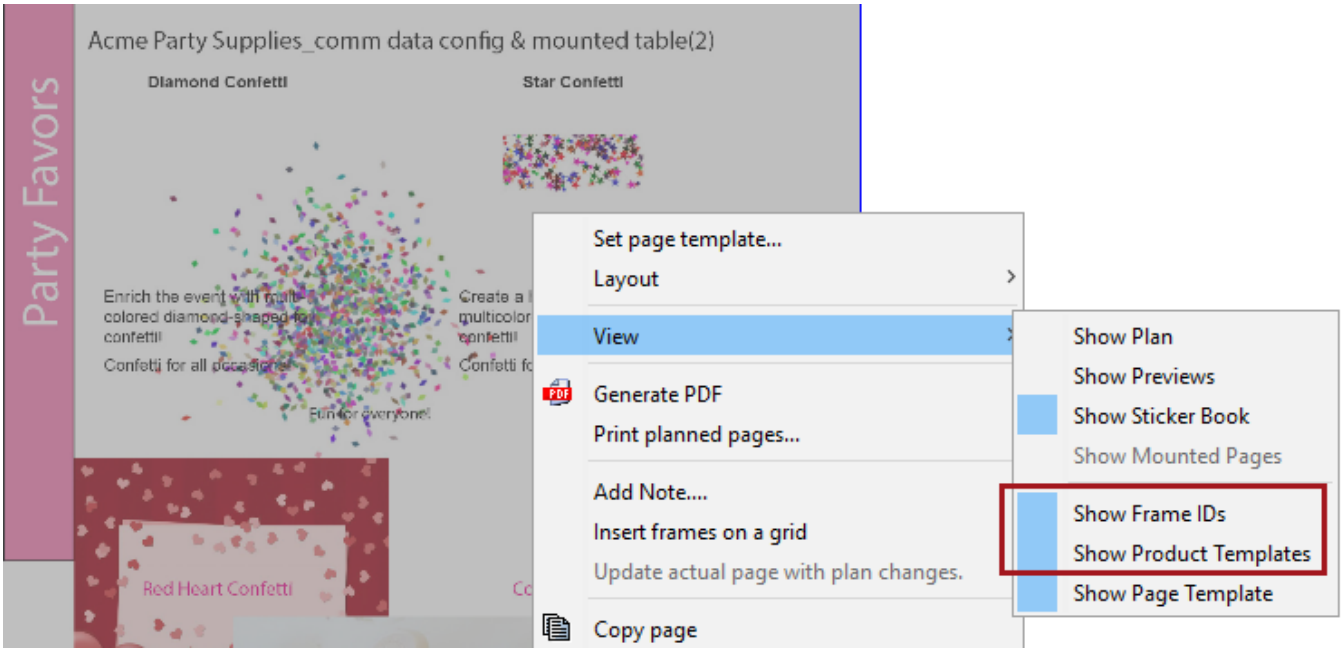
Note that this setting is global for all publications and the decision to set it to Y or N should be made before building a publication. Changing the setting during production could cause space usage to be calculated differently on different pages.

Show frame information on flatplans with layout pickup

When 'Show frame information on flatplans with layout pickup' is set to **Y**, a Flatplan with a page pickup will display the same information as any other Flatplan page—product object, layout indicator (which, in the case of picked-up pages, is the letter 'P' followed by the page number and name of the picked-up page), and frame ID. In addition, the Sticker Book view is accessible, as with any other Flatplan page.

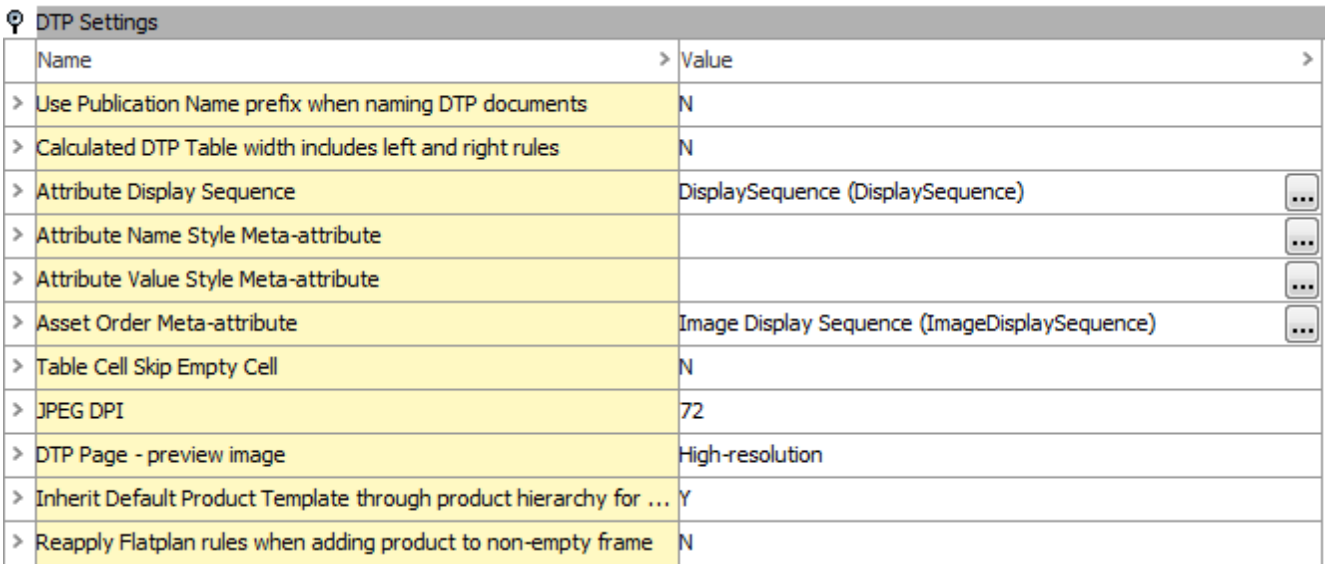


When set to **N**, a Flatplan with page pickup only shows the preview of the picked-up page. No additional frame information is available, even if 'Show Frame IDs' and 'Show Product Templates' are selected under the 'View' menu. The Sticker Book view is also disabled.



DTP Settings

This section describes the various options available under System Setup > Users & Groups > **DTP Settings**.



DTP Settings

Name	Value
Use Publication Name prefix when naming DTP documents	N
Calculated DTP Table width includes left and right rules	N
Attribute Display Sequence	DisplaySequence (DisplaySequence)
Attribute Name Style Meta-attribute	
Attribute Value Style Meta-attribute	
Asset Order Meta-attribute	Image Display Sequence (ImageDisplaySequence)
Table Cell Skip Empty Cell	N
JPEG DPI	72
DTP Page - preview image	High-resolution
Inherit Default Product Template through product hierarchy for ...	Y
Reapply Flatplan rules when adding product to non-empty frame	N

Use Publication Name prefix when naming DTP documents

When a user creates an InDesign page (DTP document) in InDesign from a Flatplanner planned page spread (by right-clicking on the planned page in the STEP Publication view palette and choosing 'Create document from plan'), the default file name of the mounted document is simply the page number (for example, 1.indd). By setting 'Use Publication Name prefix when naming DTP documents' to 'Y', the publication name will be added as a prefix to that document file name. For example, if the publication name in STEP is 'Bath Catalog Fall Collection,' the mounted document will be called ''Bath Catalog Fall Collection_1.indd'

Note: If there are spaces or any invalid or 'questionable' characters in the publication name (for example, ™), they will be removed.

Calculated DTP Table width includes left and right rules

This option tells STEP to make an adjustment to correct an anomaly in InDesign tables when vertical rules are placed on the left and right side of a table. InDesign does not normally include the width of a left or right vertical rule in the overall width of a table. Instead, it effectively oversets the table's width. Setting 'Calculated DTP Table width includes left and right rules' to 'Y' will readjust the left and right vertical rules so they are within the overall table width.

To further explain, when 'Fit calculated Table Width to Frame Width' is checked in the STEP Preferences menu in InDesign, STEP'n'design will fit the mounted table to the width of the frame on the product template that contains

the table tag. If 'Calculated DTP Table width includes left and right rules' is set to 'N,' the table will be slightly wider than the frame because the rule lines are not 'fit' inside the boundaries of the frame. When set to 'N', the table width will exactly match the frame width, as the side rules will be fit inside the frame.

Attribute Display Sequence

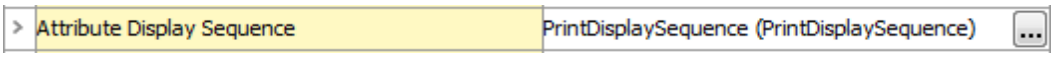
The 'Attribute Display Sequence' setting under DTP Settings is different than the 'Default Attribute to use as Display Sequence Attribute' setting under Product Information Manager Default settings. Whereas 'Default Attribute to use as Display Sequence' is used to sort the order of attributes as they display in the STEP Workbench (for example, on the 'Product' tab of a product object), the 'Attribute Display Sequence' setting controls the order of mounted attribute values from an attribute group on mounted InDesign pages.

On product templates, a tag to mount values from an entire attribute group can be placed into a frame (as opposed to the more common practice of placing single attributes at a time into frames), as pictured below.

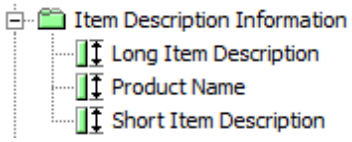
```
<STEPGROUP AG="ItemDescriptionInformation" RS="\n"></STEPGROUP>
```

When an object is mounted onto an InDesign page using a product template containing this type of tag, the values from all attributes within this group are mounted. The 'Attribute Display Sequence' attribute will determine what order the attributes mount on the page. By default, these values would display in the order in which they reside under the attribute group (which is typically in alphabetical order).

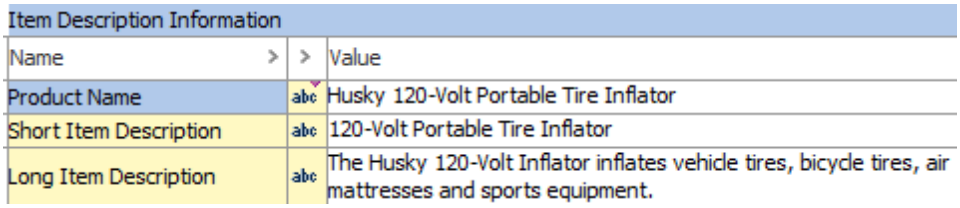
To use the 'Attribute Display Sequence' option, a description attribute is selected that is used to control the order of the mounted attributes on the page. This attribute is a description attribute valid on the 'Attribute' object type (System Setup > Object Types & Structures > Basic Object Types > Attribute Group > **Attribute**). (For information regarding how to create this attribute, see the **Attribute Metadata on Attributes** section of the **Attributes** documentation.) In the example below, the attribute 'PrintDisplaySequence' was created and selected using the ellipsis button.



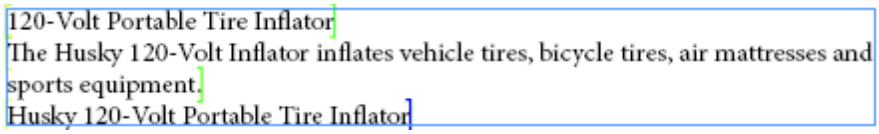
To further illustrate—as shown in the images below, the attributes are arranged alphabetically in their group in System Setup.



The attributes display on the 'Product' tab in a different order, meaning that the 'Default Attribute to use as Display Sequence Attribute' option is being used, using a different metadata attribute to order these attributes in the workbench.



Using the 'Attribute Display Sequence' option and entering values for the order in which the attributes should mount on the page, we can see that the attributes have mounted in a different order than they appear in System Setup *and* in the product editor in the workbench—Short Item Description, Long Item Description, and Product Name.



Attribute Name Style Meta-attribute and Attribute Value Style Meta-attribute

The description attributes used for the 'Attribute Name Style Meta-attribute' and 'Attribute Value Style Meta-attribute' settings are used to call out a paragraph style (not a character style) in InDesign that is applied to the name of a mounted attribute and the value of a mounted attribute, respectively, when mounted as part of an attribute group.

The attributes set for 'Attribute Name Style Meta-attribute' and 'Attribute Value Style Meta-attribute' will be created as description attributes valid on the Attribute object type. The values of these attributes should be the name of a paragraph style that is available in the InDesign publication document (mounting page). When used in conjunction with the mounting of an attribute group, the attributes that have a value for 'Attribute Name Style Meta-attribute' and 'Attribute Value Style Meta-attribute' will be mounted on the page with the style you have applied.

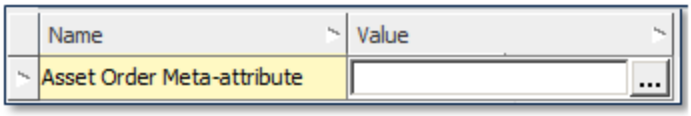
Note that these attributes must use an LOV. This LOV typically holds all the required style names used both for the attribute names and the attribute values.

These attributes need only be used when it is known that the style of the attribute *name* differs from attribute to attribute, or when the style of the *value* changes from attribute to attribute within the attribute group. For those instances where the attribute's *name* style or attribute's *value* style does not change, the product template can normally be used to generate the desired effect.

Asset Order Meta-attribute

This feature enables a sequence order to be set on assets (usually images) when products are mounted to InDesign pages. This sequence is set by the use of a metadata attribute that is valid on the link between products and image reference types. If there are multiple references from a product to various images using the same reference type (for example, 'Product Images'), and the product template calls out the Product Images, then all these images mount to the page. However, without this feature, the order that they mount would be in the same order as they are seen in the workbench—by their alphabetically sorted name. By creating a metadata attribute and placing it on the link between the product and asset (reference type = Product Images in this case), then the user may establish a numeric order to the images. If this system setting specifies this metadata attribute, then the images will mount to the InDesign page in the order specified by this 'sequence' attribute.

For example, here is how the references would mount without an attribute specified—in alphabetical order.



Reference Type	Target	Image Display Sequence
Product Images +	EXA-5002-1001	
	OMC-1005	
	UEC-100AA	
	UETC-1002	
	UGC-1001	

If there *is* a display sequence attribute applied and sequence numbers have been applied, the images will be mounted in the specified order.

Name	Value
Asset Order Meta-attribute	Image Display Sequence ...

Reference Type	Target	Image Display Sequence
Product Images +	EXA-5002-1001	5
	OMC-1005	3
	UEC-100AA	1
	UETC-1002	2
	UGC-1001	4

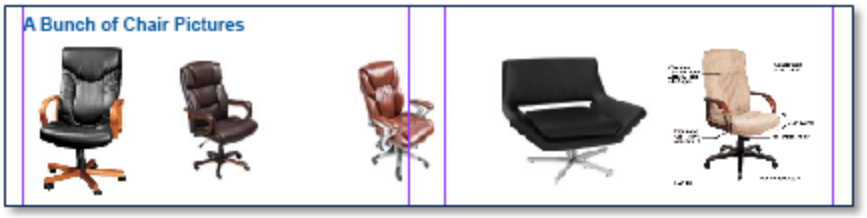


Table Cell Skip Empty Cell

Setting 'Table Cell Skip Empty Cell' to 'Y' solves an issue for customers using the 'Repeated Rule' transformation in tables in such a way that it conflicts with the way in which the system treats rules in tables. The issue concerns how the system decides to choose a rule width if one cell does not have a horizontal rule on the bottom, in effect, and the adjacent cell below it has a horizontal rule, in effect, on the top. In almost all circumstances, this setting should be set to 'N.'

JPEG DPI

DTP Settings	
Name	Value
JPEG DPI	72

This setting affects the regular product proof view (as accessed from the Proof View tab on product objects in the workbench) and the Flatplanner when a plan preview is requested. It also affects the resolution of the thumbnail image that STEP creates when you upload a document to a publication. Any change to this value is not put into effect until the InDesign sidecar (InDesign server) is restarted.

For performance purposes, care should be taken not to enter a value that is too high, e.g. 300 dpi. A proof view of 150 dpi maximum should be sufficient for most purposes.

When mounted documents are uploaded to the publication structure in STEP, a new document thumbnail is created and saved within the document based on the new resolution specified. All existing thumbnails on 'old' documents will still show the 'preview' in the former resolution. Note that in the regular product proof view, the zoom factors will no longer match. For example, 100% will really show as larger than true 100% on-screen if a resolution higher than 72 dpi has been entered.

DTP Page - preview image

DTP Settings	
Name	Value
DTP Page - preview image	Low-resolution

This setting affects how the resolutions of the page previews in STEP'n'design. The options are:

- None: No preview of DTP page. Not recommended.
- Low-resolution: Tells STEP'n'design not to get the high resolution image when displaying page previews. This is recommended.
- High-resolution: STEP'n'design gets the high resolution image when displaying page previews. This is not recommended as it can affect system performance.

Inherit Default Product Template through product hierarchy for Product Overrides.

This setting is to maintain backwards compatibility with prior versions of STEP.

In releases prior to 6.0, any product that was placed into a product override would inherit the default product template placed on that override. The default in 6.0 and up is not to inherit from the override but from the originating (or ;'natural') parent.

This is the only type of inheritance that is done from a product node 'downwards' to the product overrides below it.

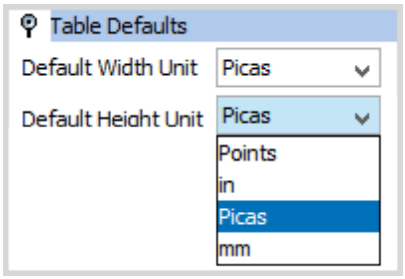
Reapply Flatplan rules when adding product to non-empty frame

This option is only relevant for Galley Flatplans, which allow multiple products to be placed in the same frame.

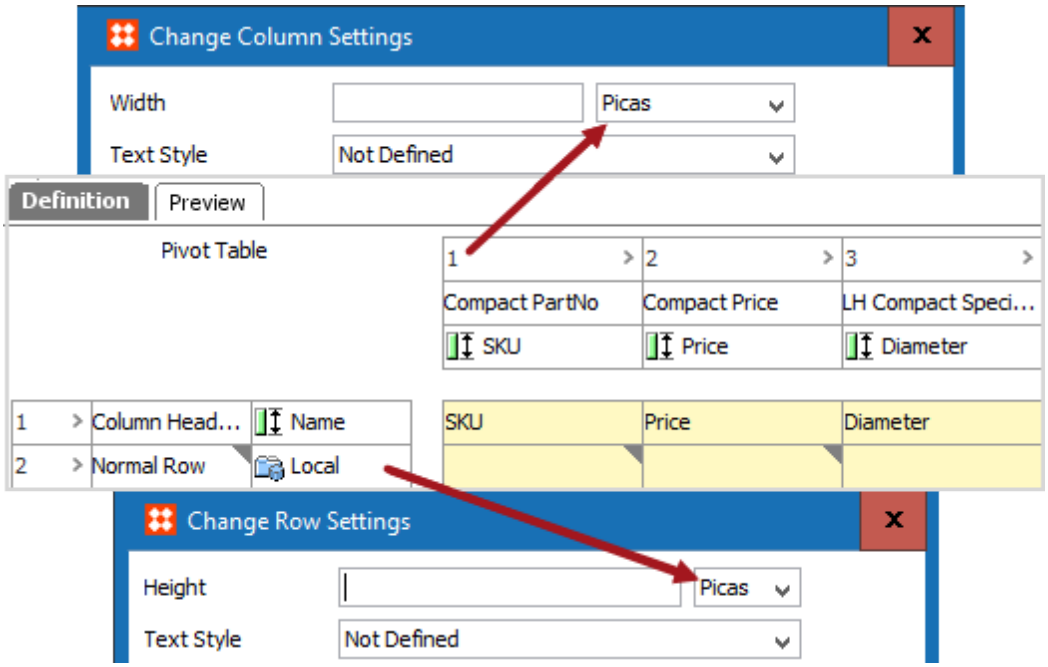
Normally, Flatplanner does not check Flatplan publication rules if a template is already assigned to a frame. Setting this option to 'Y'forces it to do so.

Table Defaults

The settings under Table Defaults establish the default unit that will be applied to the height or width of columns, rows, and entire tables in the workbench table editor for STEP tables. For example, the default unit can be set to Picas from the dropdown menu as shown below.



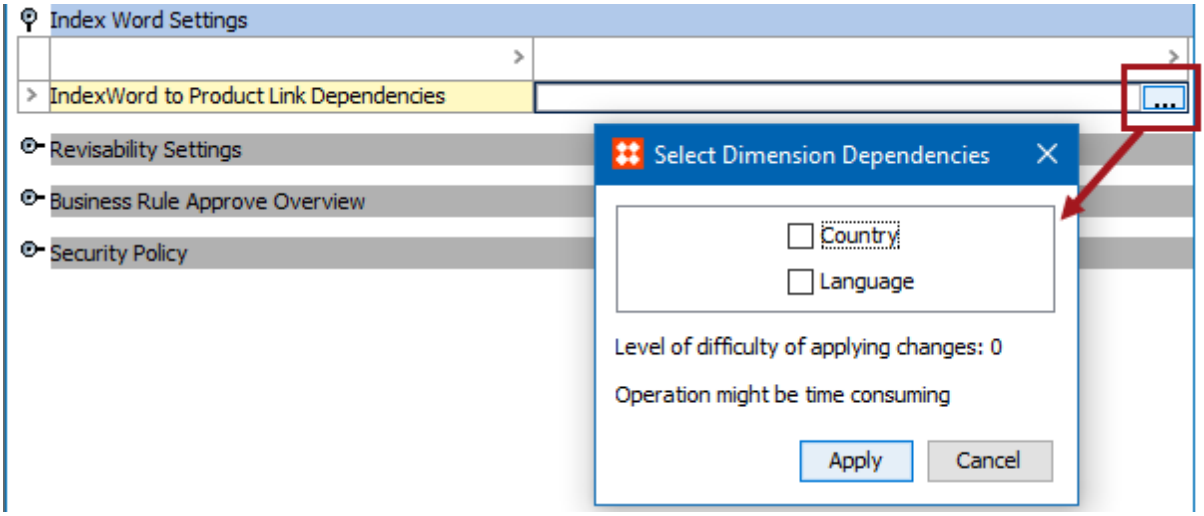
Once the default units have been set, the units will be the default units in the table editor as shown in the image below. If no default unit is selected, the default units will be blank and must be manually selected every time.



For more information on STEP Tables, see **About STEP Tables** in the **Tables** documentation.

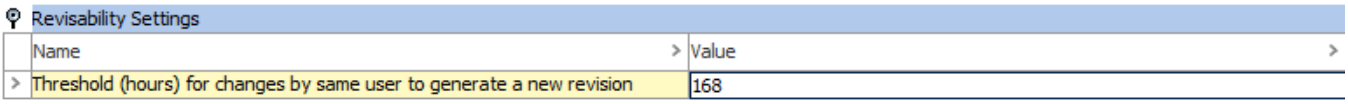
Index Word Settings

By default, the links between products and index words are not dimension independent. By selecting a dimension under 'IndexWord to Product Link Dependencies,' you can make these links dimension dependent. By doing this, the index words hierarchy can be translated and shared for multi-markets.



Revisability Settings

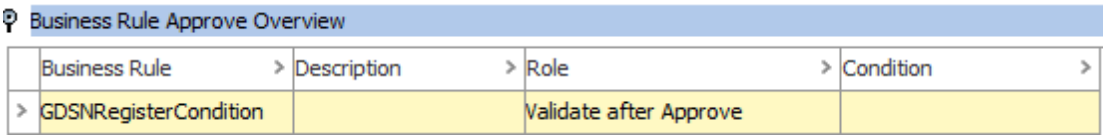
Threshold (hours) for changes by same user to generate a new revision



STEP automatically preserves all versions of a revisable object. Each time a change is made by a different user, a numbered revision is created and logged. This setting sets the threshold for the number of hours to occur before a new revision by the same user is reflected in STEP. The default setting is 168 hours or one week. 24 hours is the minimum number of hours that can be entered here.

For more information about managing revisions in STEP, please see **Managing Revisions in STEP**.

Business Rule Approve Overview



Business Rule Approve Overview

	Business Rule	Description	Role	Condition
>	GDSNRegisterCondition		Validate after Approve	

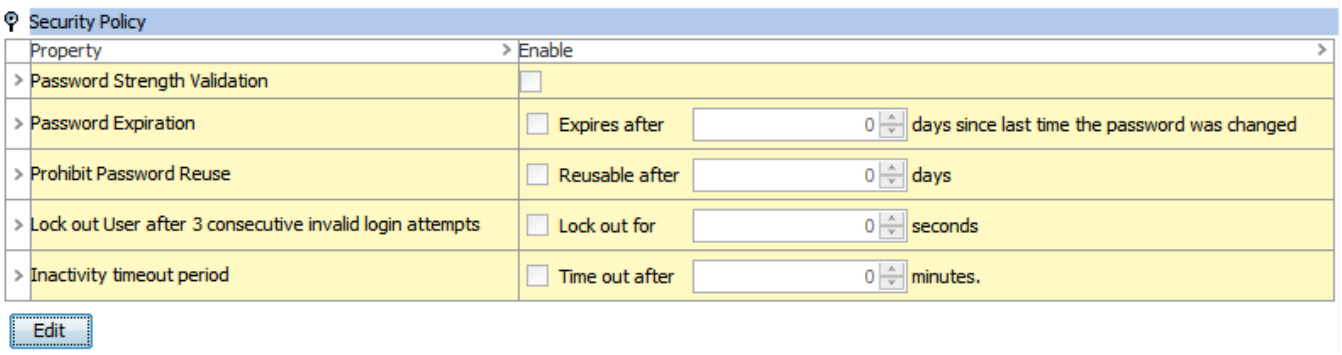
Some STEP implementations have a large number of business rules in place, some of which are executed as part of an approval process (determined by the On Approve setting on the rule itself). The Business Rule Approve Overview section of the System Settings editor serves to consolidate all rules executed with approvals so that they can be easily identified. The name listed under the business rule column is a hyperlink to the business rule so that a user can quickly access and edit the business rule.

For more information about approval-based business rules, please see the **Using Business Rules During the Approval Process** section of the **Business Rules** documentation.

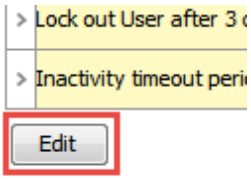
Security Policy

To ensure STEP users adhere to the password security level set by the customer, STEP enables admin users to configure security settings related to passwords.

To access these password settings, first select the **System Setup** tab and then the **Users & Groups** node. Scroll down to the very bottom of the page to find a section called **Security Policy**. This section is comprised of five settings, shown below:



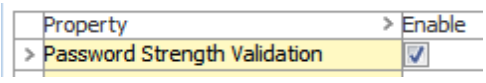
To make changes to these **Security Policy** settings, click the **Edit** button found beneath the section in the lower left-hand corner of the screen:



Important: Any changes made to these five settings will be applied to **all** user groups located under the **Users & Groups** node. These settings cannot be customized for single users or single user groups.

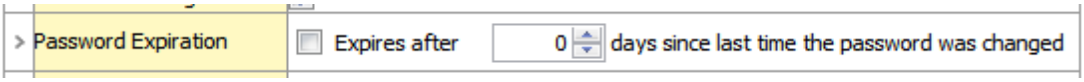
Clicking the **Edit** button brings up the **Security Policy** settings in a new window and makes them accessible for editing:

1. **Password Strength Validation** – If this box is checked then any new password must conform to a series of requirements intended to ensure a strong password.

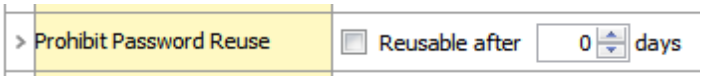


If the new password does not meet a minimum of three of the four password strength requirements – at least one lower-case letter, one upper-case letter, one numeral, and one special character – the user will see this dialog window appear:

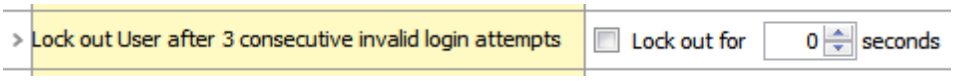
2. **Password Expiration** – With this box checked, all new passwords will expire after a configurable number of days have passed. Once a password has expired, the user will be notified upon their next log-in attempt and prompted to change their password.



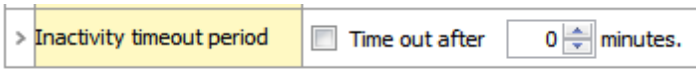
3. **Prohibit Password Reuse** – With this box checked, the system can prohibit a user from reusing old passwords. The number of days old passwords are unavailable for reuse is a configurable value.



4. **Lock out User after 3 consecutive invalid login attempts** – With this box checked, three failed attempts to log-in by entering a wrong or mistyped password will prohibit any new log-in attempts for a variable number of seconds.



5. **Inactivity timeout period** – With this box checked, the system can be configured to automatically log out any logged-in user if they've been inactive for a configurable number of minutes. By signing out inactive STEP users, the speed of STEP is preserved for all active users. This setting is most appropriate to activate for STEP systems with a large number of active users.



GUI Localization

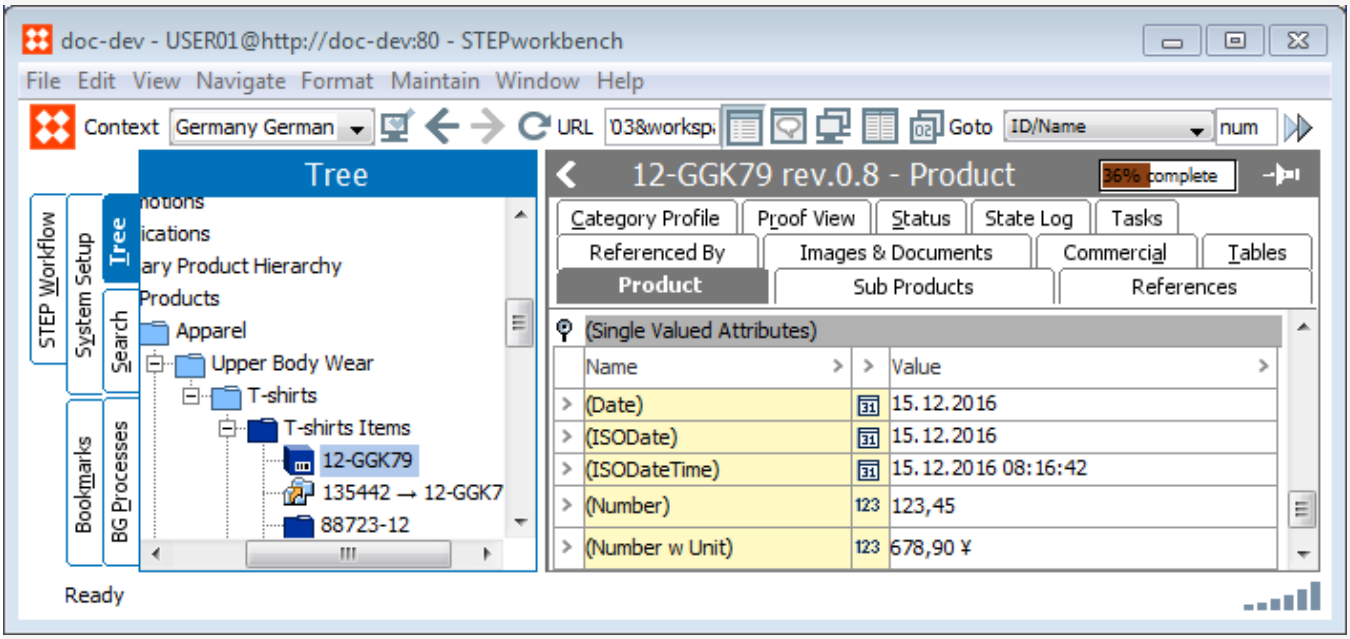
GUI Localization is maintained in [System Setup]: **Users & Groups** -> **Users** in the **System Settings** tab.

In the Workbench when viewing and maintaining **Numbers** and **Dates**, they will as default be maintained and viewed using the English - US locale. A number like 1234.56 will appear with a dot to indicate the decimal. When editing and saving numbers and dates in the database,only an English-US locale is allowed to be used.

In the User editor in the System Settings tab, you have an option to select a specific locale to be used for a User. This locale will change the behavior in the Workbench to match a selected locale for the specific user. This option is useful for users working in other countries and where they want to view and maintain values in the Workbench with a different locale than the English - US.

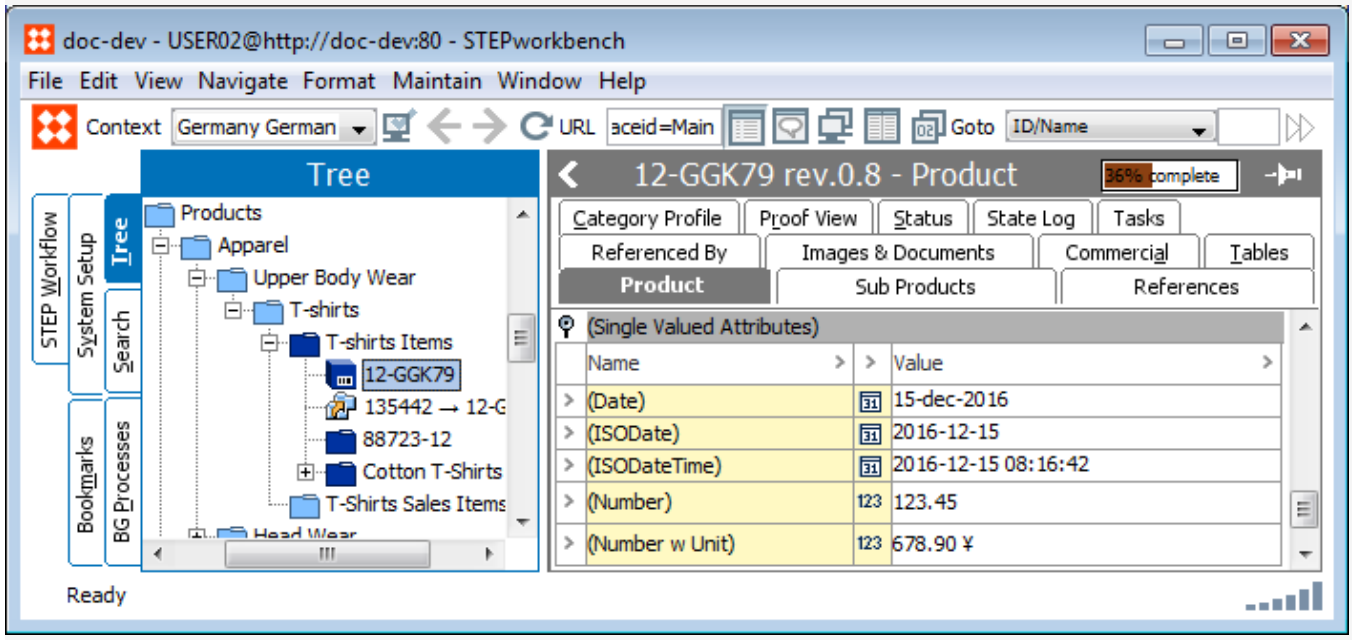
Important: A locale applied a User will only change the view of numbers and dates according to the locale in the Workbench and it will change dialogs to allow saving numbers and dates based on the locale. A number can e.g. be saved with a comma as a decimal separator using a German - de locale. Behind the scenes it is always the English - US locale that is used to save numbers and dates into the database.

Example User 01 using German - de locale



In this example User01 is logged on the Workbench. The User is setup to use a German - de locale for numbers. A comma will appear in single and multi valued attributes / LOVs. User 01 will only be allowed to use a comma as a decimal separator for numbers. When User 01 saves a number into the database, the English - US locale will still be used to save the number. So if User01 enters a number: 2343,76 into a value field, the value will be saved in the database as 2346.76.

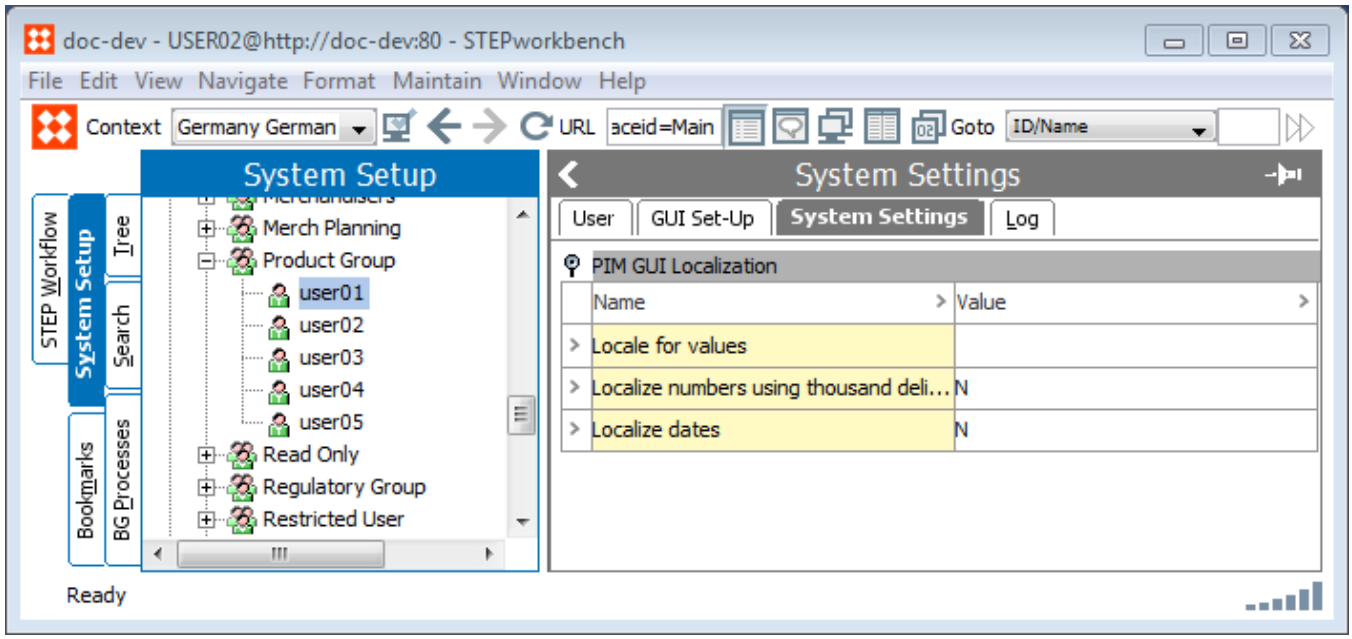
Example User 02 using English - us locale



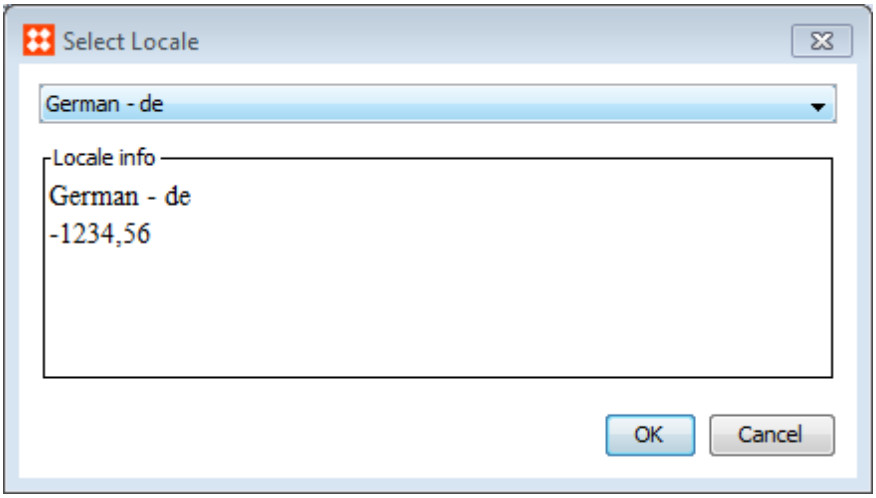
In this example User02 is logged on the Workbench. The User is setup to use a English-us locale for numbers. A dot will appear in single and multi valued attributes / LOVs. User 02 will only be allowed to use a dot as a decimal separator for numbers.

Setting up a GUI Localization for a User

GUI Localization is maintained in [System Setup]: **Users & Groups** -> **Users** in the **System Settings** tab.



1. In **Locale for values**, click field to open a locale selector dialog. In this example a German - de locale is selected for User 01.



2. Click **OK** to apply the German locale for User 01.

Note: Users without any locale applied will as default use the English - US locale to enter values or viewing values in the Workbench editors.

---

3. In **Localize numbers using thousand delimiter**, select Y to localize numbers using a thousand delimiter. In this example a German - de locale is selected for User 01, which will enable a dot to be shown as a thousand delimiter for Number and Integer validated values. For User 01, values like 1000 or 1000.10 will be shown as 1.000 and 1.000,10.

Note: The thousand delimiter is only a display setting for values shown in the Workbench. Saving numbers and integers will not be saved in the database with a dot as a thousand delimiter.

---

4. In **Localize dates**, click Y to localize dates in the Workbench. In this example a German -de locale is selected for User 01. For User 01, isodates will be shown as 13.05.2009 and isodatetime will be shown as 13.05.2009 19:10:48.

About Privilege Rules

Each Group must be assigned with Privilege rules, defining which Actions the members of the Group are allowed to perform, and which data the members of the Group is allowed to work with.

Privilege rules are maintained in [System Setup]: **Users & Groups** -> **Groups**

Note: Defining the Privilege rules forms one part of the Privilege control system settings. The second part, defining Action Sets is performed in [System Setup] under **Action Sets**. For more information about Action Sets, see System Setup/About Action Sets.

A Privilege rule specifies a set of permitted Actions that members of a specific Group can perform on a specified set of data. There is no limit to the number of Privilege rules that can be applied to the same Group.

Each Privilege rule is defined by specifying:

- A Group (of Users)
- An Action Set
- A Product, Classification, Publication, Entity or eCatalog hierarchy node, a node collection or a STEP Workflow and a state.
- An Attribute Group (or one Attribute).
- An Object Type

Note: Setting Attributes is optional.

Note: Setting Dimension Points is optional.

A Privilege rule grants all the Users (members) in a Group the permission to perform any of the Actions in the selected Action Set. This permission can be restricted to specific sets of data.

Important: Privilege Rules applied a User Group will be inherited to sub-User Groups and be available on Sub User Groups. This means it is possible to apply general Privilege Rules on a top level and only specify local Privilege Rules on sub User Groups. GUI setup applied a top User Group will not be inherited to any sub User Group.

Important: It is to be recommended to apply all general Privilege Rules as high as possible in the User Groups Hierarchy to simplify

---

Privilege Rule Types

There are two types of Privilege rules:

Type	Description
Setup privileges	Covering the Actions that are typically performed by the System administrator.
User privileges	Covering the Actions that are typically performed by the Normal user.

Restricting Actions

Actions can be restricted to:

- Objects that are linked to a specific node (or a node below) in the Classification, Product, Publication, Entity or eCatalog hierarchy, a in specific node collection or in a STEP Workflow in a specific state.
- Attributes within a specified Attribute Group (e.g. changing an Attribute Value).

  If no Attribute Group is specified, the Action is permitted for all Attributes.

- Workflows, Integration Endpoints, Business Rules within a specified Setup Group
- Specified Dimension Points.

  If no Dimension Point is specified for a Dimension, the Action is permitted for all Dimension Points within that Dimension.

- Specified Object type.

  If no Object type is specified, the Action is permitted for all Object types.

Unlimited Number of Privilege Rules

Any number of rules may be set up for the same Group, e.g. granting permissions for specific Actions sets on a number of hierarchy nodes and Dimension Points.

Note: A User may be a member of more Groups. The User will then accumulate the Privilege rules from all these Groups, so e.g. a restriction from one Group is ignored if the corresponding Action is permitted via another Group membership.

Editing linked Objects

If e.g. a Privilege rule permits the editing of a specific Attribute Value for Products linked to a specific Classification node, this Attribute Value can also be edited through other Classification nodes that these Products may be linked to.

Access to Classifications and Products

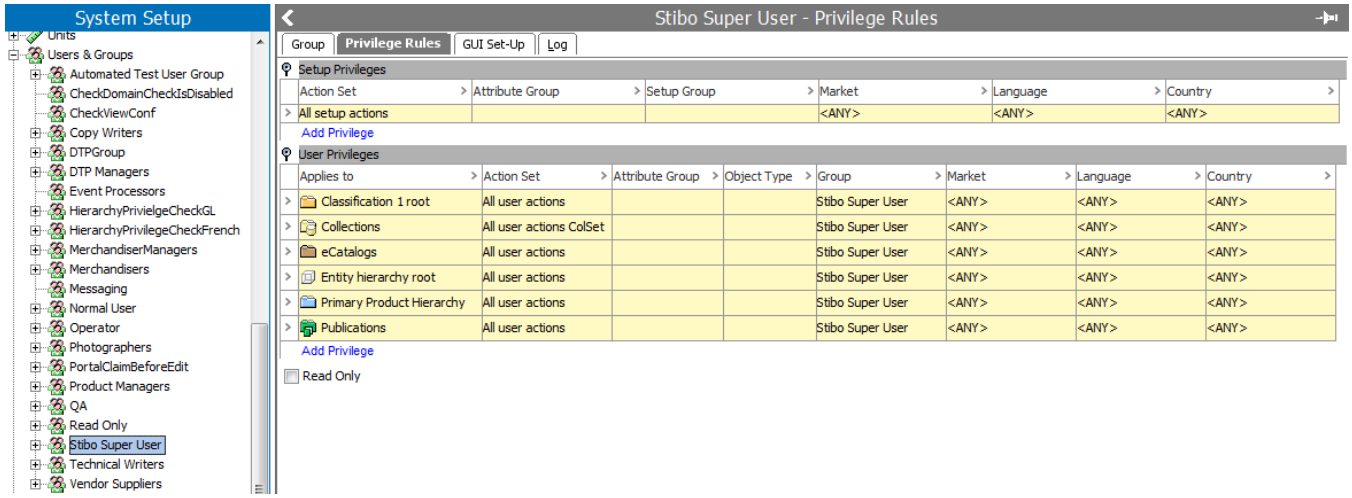
If a Group has permission to work with a specific Classification node, Attribute Values can be edited for all Products linked to (or below) that node. However, access to the Classification node does not grant the privilege to link new Products in the Product hierarchy. To do that, the Group must have permission to work with the specific Product node as well.

Examples of Privilege Rules

Who?	What?	Where?
Group A	is allowed to link/unlink Products	in Classification node X for Dimension Points a, b, c
Group B	is allowed to edit the Attributes	in Attribute Group technical for all Products linked to Classification node Y for Dimension Points d, e, f
Group C	is allowed to create Products	within Product family G
Group D	is allowed to edit the Context specific Name	for Products
Group E	is allowed to delete Products	from Product family Z
Group F	is allowed to link/unlink Attributes	in Classification node W for Dimension Points d, e, f
Group G	is allowed to maintain Workflows	for Workflows in Setup Group X

Editing Privilege Rules

Privileges are maintained in [System Setup]: **Privilege Rule** tab in **Group** Editor.



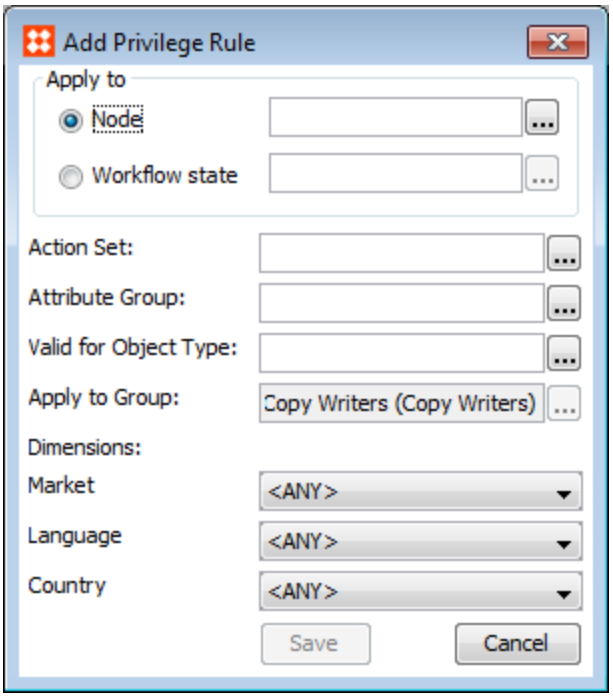
For more information about Privileges Rules and the Privileges Rule Types, see Privilege Rules in Action Sets.

1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group.

   The Group appears in the **Group** Editor.

2. Click the **Privilege Rules** tab.
3. Click the relevant Privilege to be changed, right-click, and then click **Edit Privilege Rule**.

   An **Edit Privilege Rule** dialog box appears.



4. Make the relevant changes, and then click **Save**.

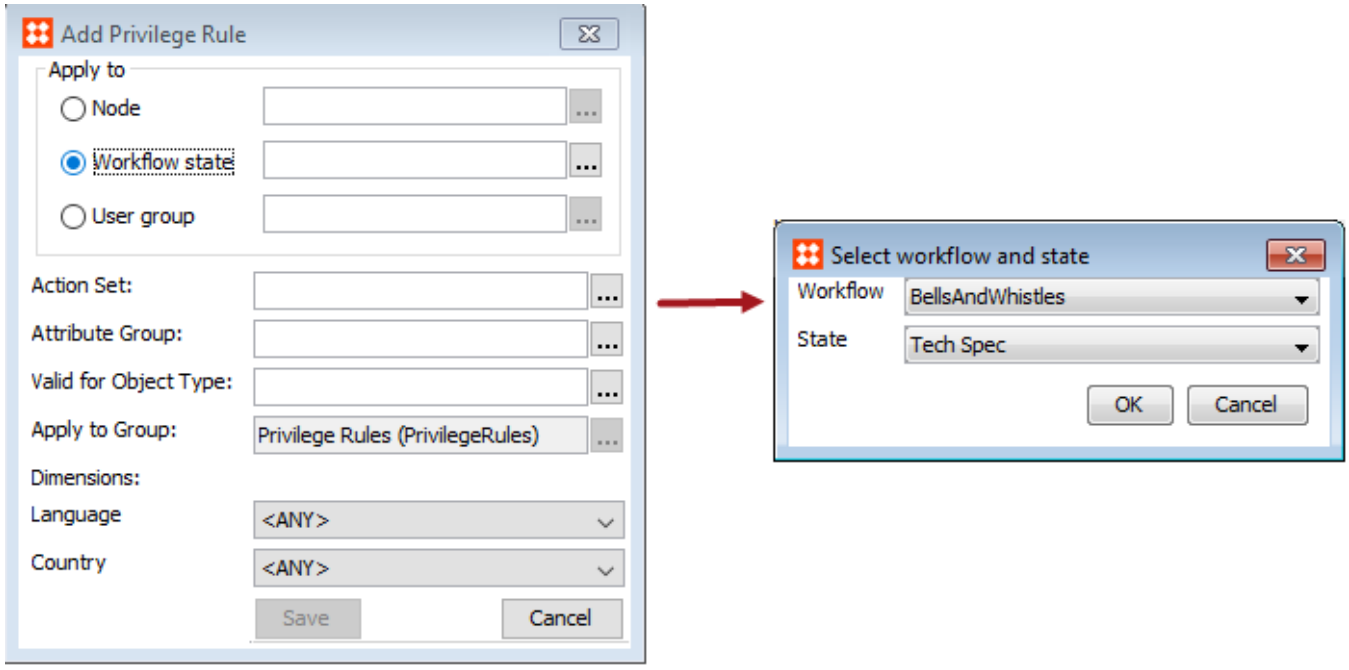
Setting	Description
Apply to	select weather the privilege should apply to a hierarchy below a node or if it should apply to a particular state of a workflow. See more details below.
Apply to node	Select the allowed classification, product, entity, eCatalog, publication or collection node for the group.
Apply to workflow state	Select the STEP Workflow and the state within the workflow, that the privilege should apply to. See more details below.
Action Set	Select the permitted Action Sets for the Group. For more information about Action Sets, see About Actions Sets in System Setup.
Attribute Group	Optional. Select the permitted Attribute Group. Select if the Group should only be allowed to work with attributes in a specific Attribute Group.
Valid for	Optional. Select the permitted Object type.

Setting	Description
Object type	For more information about Object Types, see Object Types and Structures in System Setup.
Apply to Group	Select a Group. Note: If a Group is already selected, it will appear as default.
Market Language Country	Optional. Select the allowed Language, Country and/or Market dimension.

For more on setting up workflow privileges see the Configuring Workflow Privilege topic in the System Setup / STEP Super User Guide documentation.

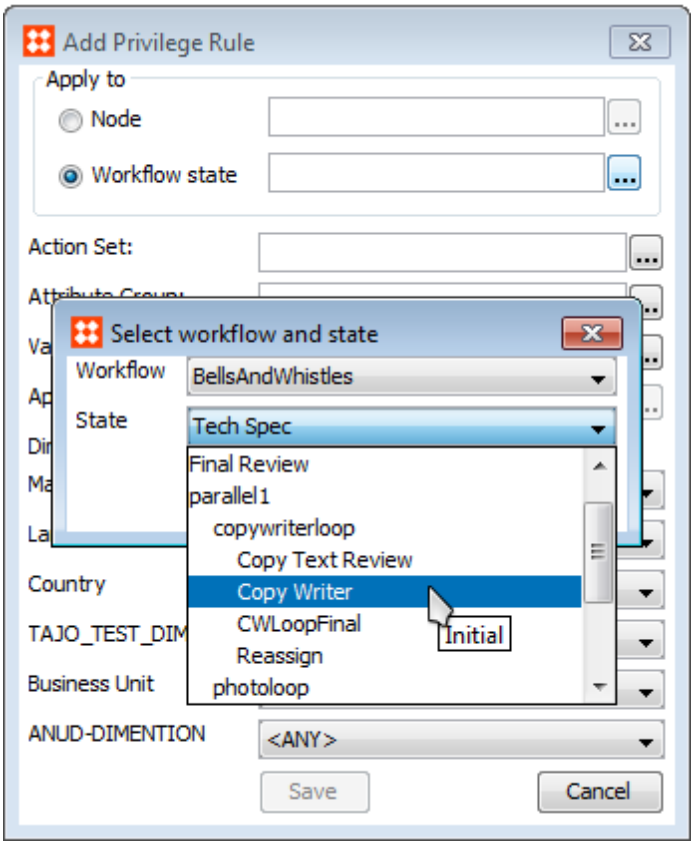
Configuring Workflow Privilege

User privileges can be applied to a STEP Workflow state. When you create a user privilege rule for a node, the group will have that privilege for that particular node and all nodes below it in the hierarchy. When you create a user privilege rule for a STEP Workflow state, the group will have that privilege for nodes in that particular state in the STEP Workflow. A possible use of this is to set up a group of data entry users, that are only allowed to view a part of the product hierarchy. Once the products enters the data entry state, the users can be set up to be able to modify values in a certain attribute group and submit the products once done. If you click the 'Workflow state' radio button in the dialog below, and hit the ellipsis button, you will get the below STEP Workflow state selector:



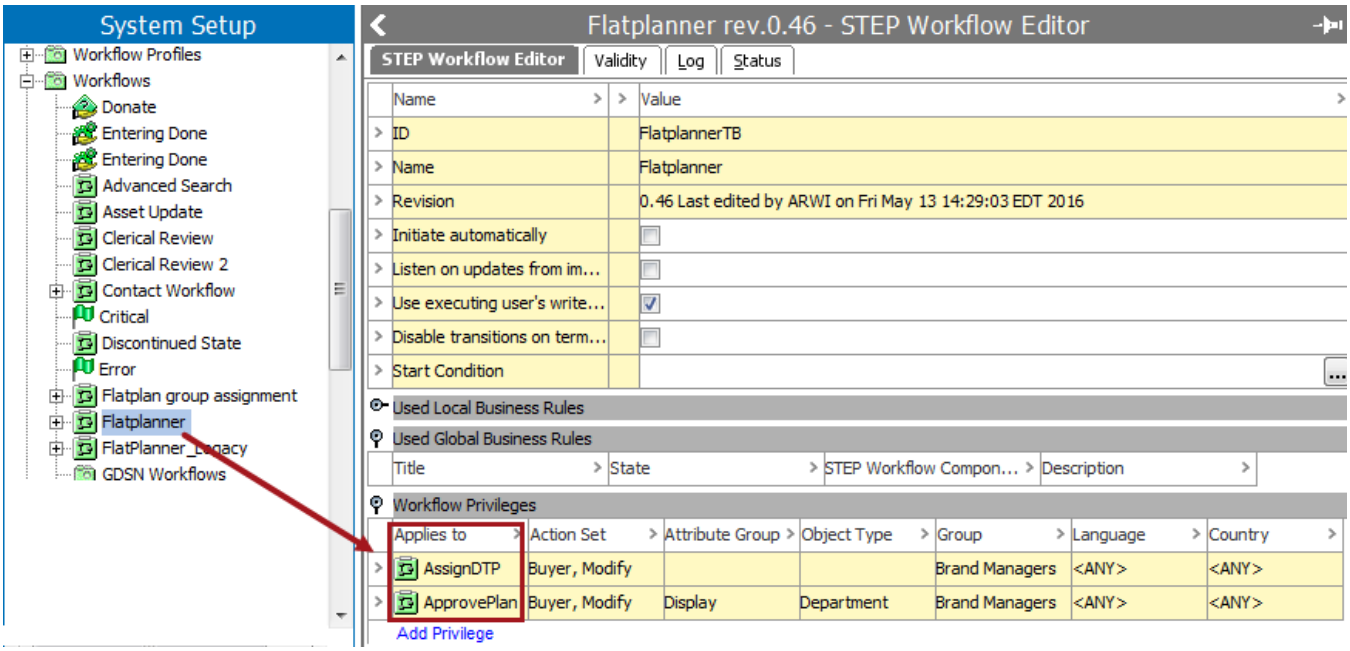
Workflow state selector

The dialog consist of two combo boxes. The top combo box allows you to select the STEP Workflow. Once the STEP Workflow has been selected the button combo box will be populated with all the states within the selected STEP Workflow. The states are arranged such that cluster and parallel states will have their child states shown below them indented. If you hover the arrow on a state, the hover text will give information about the type of the state; initial, cluster, parallel, or final.

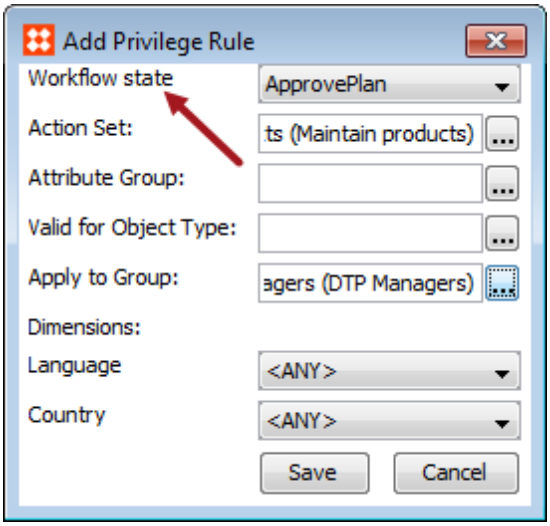


Editing Privilege Rules with focus on STEP Workflows

The above describes how to edit privilege rules with focus on a User Group. However, it is also possible to edit privileges with focus on either a product or classification hierarchy. Thus you can also find the privilege table on the "Referenced By" tab when having selected a product - or on the "References" tab when having select a classification in the Workbench. Additionally, you can set up privileges while focused on STEP Workflows by using the "Workflow Privileges" table on the STEP Workflow Editor. This shows all the privileges rules referring the selected Workflow. The first column in the table is slightly different from the privilege rules table when shown for a group, where it shows the name of the STEP Workflow followed by the name of the state the rule applies to. On the STEP Workflow Editor the "Applies to" column only displays the state - as the STEP Workflow is given from the selection.



Likewise, when editing a privilege rule or adding a new privilege rule via the STEP Workflow Editor, the dialog is slightly different. Instead of radio buttons to select either a node or a Workflow and State, a combo-box allows you to select a state of the given Workflow.



It is possible to see which states have privilege rules in the Workflow Editor. The states will be marked with an encircled "P".

Limitations on Workflow privileges

Not all actions can be granted as Workflow / State privileges. It is valid for a Workflow / State specific privilege rule to add an action set containing some of the actions that does not apply to Workflow / State specific rules. These actions will be ignored. A complete list of the valid actions for Workflow / State specific rules are:

- Approve Classification Product Links
- Approve Create
- Approve Modify
- Approve References
- Approve Values
- Classify asset
- Classify product
- Create asset reference
- Create classification reference
- Create entity reference
- Create product reference
- Declassify asset
- Declassify product
- Delete asset reference
- Delete classification reference
- Delete entity reference
- Delete product reference
- Download asset
- Link (product) attribute into classification
- Modify metadata for asset
- Modify metadata for asset reference
- Modify metadata for attribute to classification link (also translate)
- Modify metadata for classification
- Modify metadata for classification reference (link value) (also translate)
- Modify metadata for entity
- Modify metadata for entity reference
- Modify metadata for product

- Modify metadata for product classification (link value) (also translate)
- Modify metadata for product reference
- Modify name/description of asset
- Modify name/description of classification (also translate)
- Modify name/description of entity
- Modify name/description of product
- Modify product attribute value
- View asset reference
- View attribute to classification link
- View classification reference
- View entity reference
- View metadata for attribute to classification link (link attribute value)
- View metadata for asset (property value)
- View metadata for asset reference
- View metadata for classification
- View metadata for classification reference (link attribute value)
- View metadata for entity
- View metadata for entity reference
- View metadata for product
- View metadata for product classification
- View metadata for product reference
- View product attribute value
- View product
- Unlink (product) attribute from classification
- Upload to asset

Adding User Privileges for a Group

User Privileges are maintained in [System Setup]: **Users & Groups**: **Privilege Rules** tab.

A Privilege rule specifies a set of permitted Actions that members of a specific Group can perform on a specified set of data. There is no limit to the number of Privilege rules that can by applied to the same Group.

1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group.

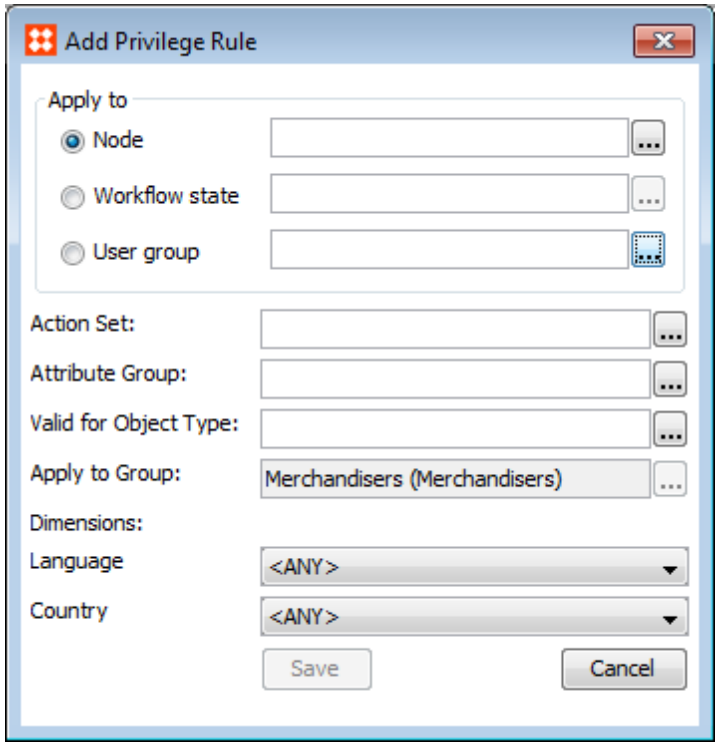
   The Group appears in the **Group** Editor.

2. Click the **Privilege Rules** tab.
3. In the **User Privileges** field, click **Add Privileges**.

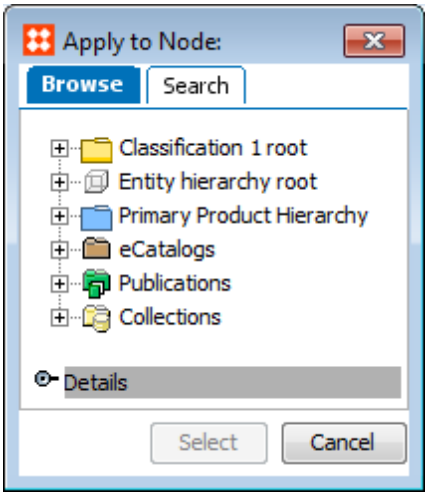
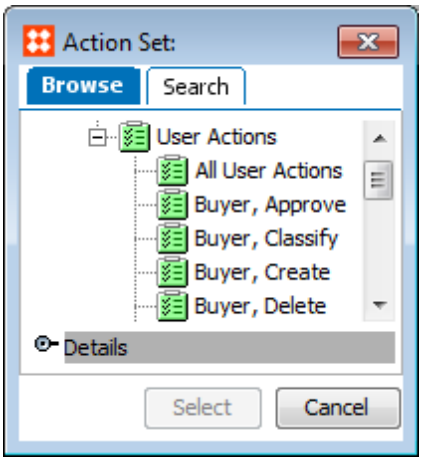
   A default “All user actions” appears, ready to edit.

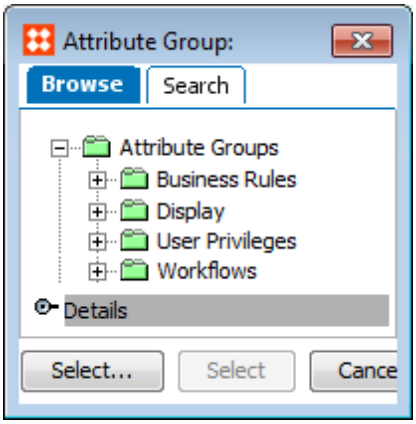
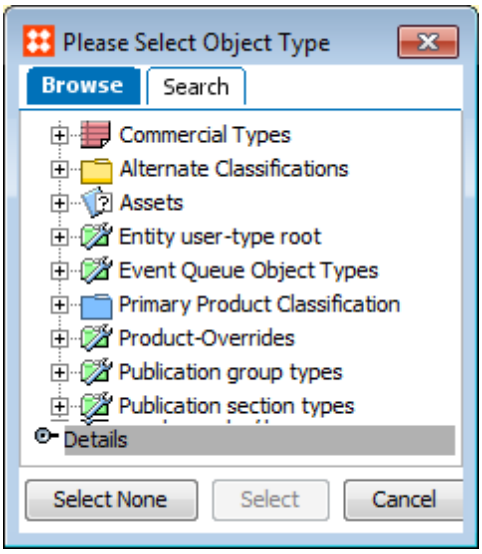
4. Right-click in relevant column, and then click **Edit Privilege Rule**.

   An **Edit Privilege Rule** dialog box appears.



5. Make the relevant changes, and then click **Save**.

Setting	Description
Apply to Node	Select the allowed classification or product node for the group. 
Action Set	Select the permitted Action Sets for the Group. For more information about Action Sets, see About Action Sets in System Setup 
Attribute Group	Optional. Select the permitted Attribute Group. Select if the Group should only be allowed to work with attributes in a specific Attribute Group. **Tip:** Consider creating attribute groups that will be used only for privileges and link allowable attributes into the user privilege groups, (approve, view and modify). Separate privilege rules can handle the Category Specific attribute group or sub-groups if they exist.

Setting	Description
	
Valid for Object type	Optional. Select the permitted Object type.  For more information about Object Types, see Object Types and Structures in System Setup.
Apply to Group	Select a Group. **Note:** If a Group is already selected, it will appear as default.
Market Language Country Price	Optional. Select the allowed Language, Price Group, Division or Market dimension.

Note: The **Object type**, **Attribute Group**, **Action set**, and **Apply to Node** fields are standard. The rest of the fields will reflect the currently set up Dimensions, e.g. **Language**, **Price** and **Market**. Also note that **<ANY>** will allow the Group to work with all Dimension Points. For more information about Dimension Points, see Contexts in System Setup.

A User Privilege rule has now been assigned to a Group. If a specific Object type and node is selected, the Users in the Group can only see Objects within that selection. E.g. a specific Classification or Product node, or Products of a certain Object type.

However, if a Product not included in the selection is linked to a Classification node to which the User has access, the User can see the Product, but not edit it.

Adding Setup Privileges for a Group

Group Privileges are maintained in [System Setup]: **Users & Groups**: **Privilege Rules** tab.

Setup privileges cover the Actions that are performed by members of the **Super user** Group in [System Setup].

1. In [System Setup], unfold **Users & Groups**, and click the relevant Group.

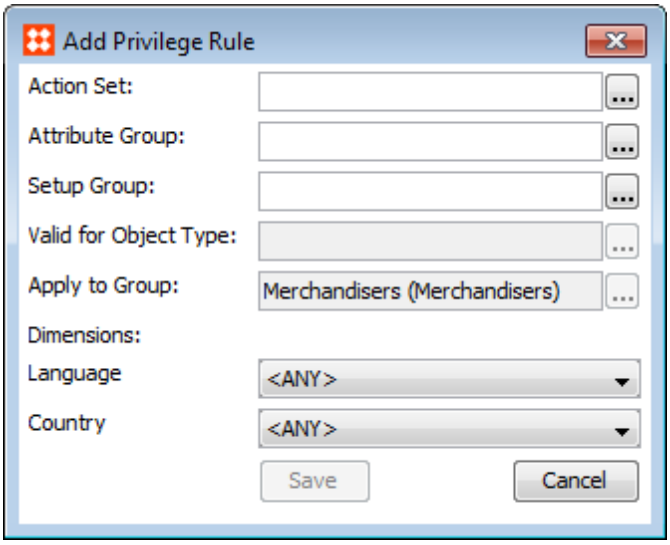
   The Group appears in the **Group** Editor.

2. Click **Privilege Rules** tab.
3. Under **Setup privileges**, click **Add Privilege**.

   A default "All setup actions" appears, ready to edit.

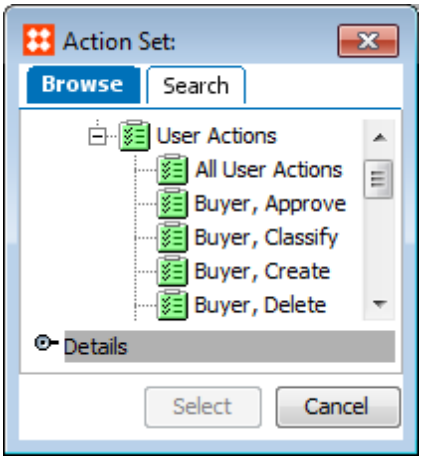
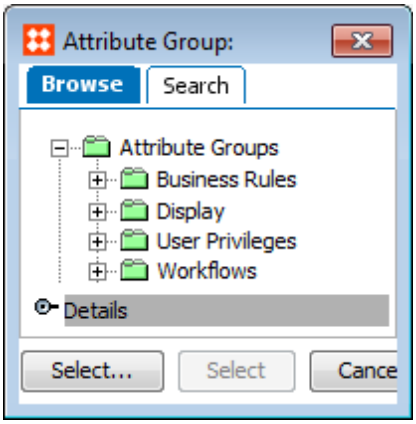
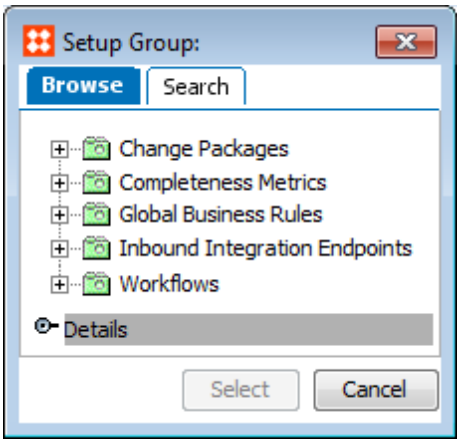
4. Right-click in the relevant **Action Set** field, and then click **Edit Privilege Rule**.

   An **Edit Privilege Rule** dialog box appears.



5. Make the relevant changes, and then click **Save**.

Setting	Description
Action Set	Select the permitted Action Sets for the Group. For more information about Action Sets, see About Action Sets in System Setup.

Setting	Description
	
Attribute Group	Optional.: Select the permitted Attribute Group. 
Setup Group	Optional: Select the permitted Setup Group. 

Setting	Description
Valid for Object type	Selecting an Object Type is not a part of a Setup privilege.
Language Country	Optional. Select the allowed Language, Price Group, Divison and/or Market etc.

Note: The **Attribute Group** field and **Action set** field are standard. The rest of the fields will reflect the currently set up Dimensions, e.g. **Language**, **Price** and **Division**. Typically a Super User should have permission to work with all Dimension Points. Also note that **<ANY>** will allow the Group to work with all Dimension Points. For more information about Dimension Points see About Contexts in System Setup.

Note: Setting Dimension Points is a very time consuming process.

A Setup Privilege rule has now been added to the Group.

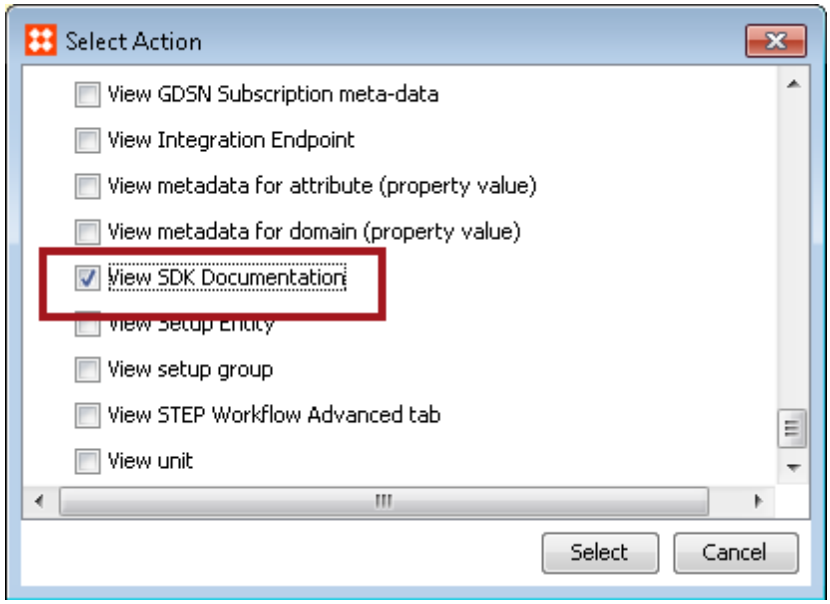
Granting Access to the SDK and API Documentation

To provide additional security within the STEP system, users must log in with user name and password to access the SDK and API documentation from [server]/sdk. Only users who are members of a user group that has been granted the **View SDK Documentation** setup action privilege are able to log in.

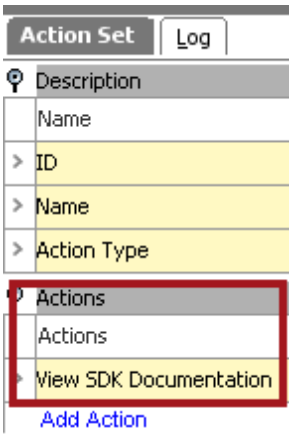
Assigning the privilege is a two-step process: First, you create an action set that includes the **View SDK Documentation** setup action privilege, and then you assign the action set to the relevant groups.

Grant Access to the SDK and API documentation

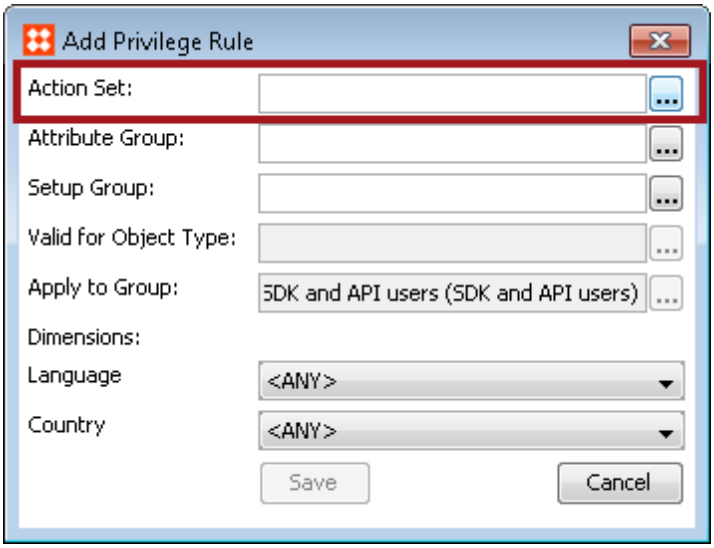
1. In **System Setup**, expand **Action Sets**, right-click **Setup Actions**, and then click **New Action Set**.
2. Enter an **ID** and a **Name** for the action set, and then click **OK**.
3. On the **Action Set** tab, click **Add Action**.
4. In the **Select Action** dialog, scroll down and select **View SDK Documentation**, and then click **Select**.



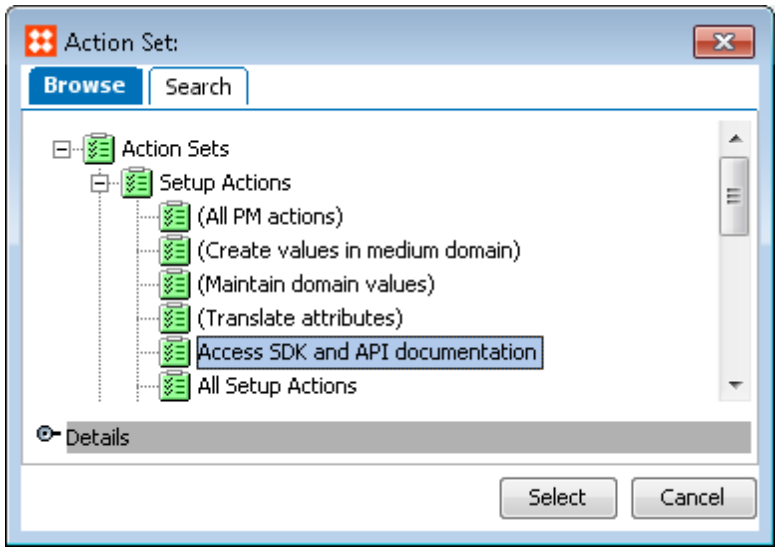
The action is now listed in the **Actions** area on the **Action Set** tab, and you can add the action set to the relevant user group.



5. Expand **Users & Groups**, and then select the relevant user group.
6. Click the **Privilege Rules** tab, and then click **Add Privilege**.
7. In the **Add Privilege Rule** dialog, next to the **Action Set** field, click the ellipsis button **(…)**.



8. In the **Action Set** dialog, expand **Action Sets**, and then expand **Setup Actions**.
9. Select the action set that grants access to the SDK and API documentation, click **Select**, and then in the **Add Privilege Rule** dialog click **Save**.



The privilege is granted to the selected user group.

About GUI Setup

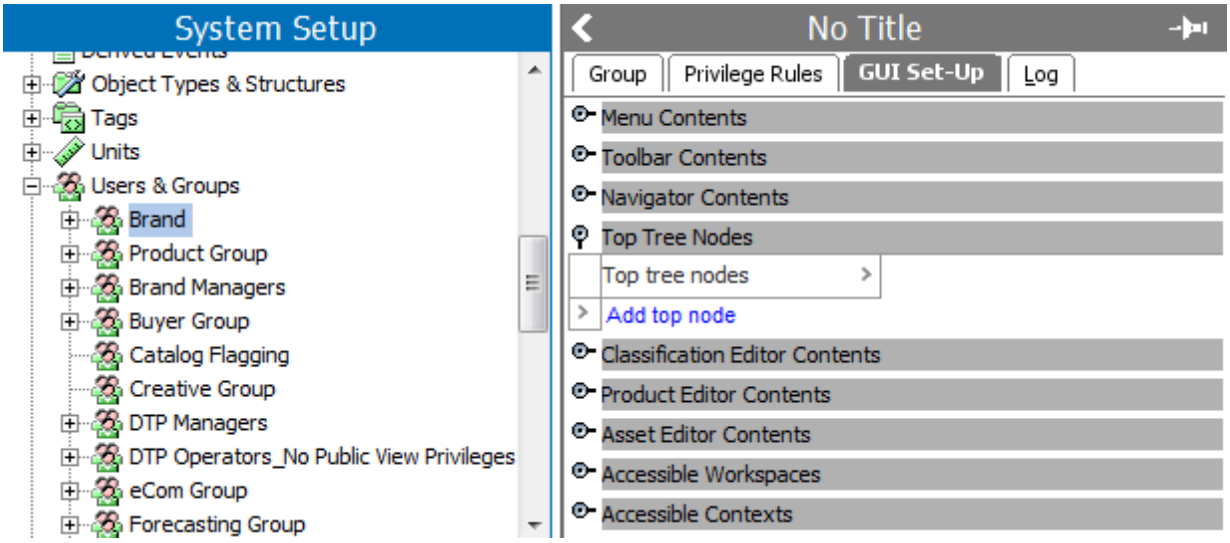
GUI Set-Up is maintained in [System Setup]: **Users & Groups** > **Groups** or **Users**

Note: Defining the Privilege Rules forms one part of the Privilege Control system settings. You can give users access to some STEP functionality but limit others. This will result in a STEP PIM GUI Interface, where some of the commands are made unavailable (appear as dimmed). The second part is defining the GUI Set-Up. You can consider defining the GUI Set-Up as cleaning up the user interface, based on the Privileges rules that are set up for a specific User Group.

GUI Set-Up makes it possible to:

- Customize contents in toolbars and menus. You can add and remove buttons and menus on toolbars, hide or display toolbars.
- Make specific Contexts and Workspaces accessible
- Set Navigator Contents
- Contents in Classification, Product and Asset Editors
- Set top level Tree nodes

Important: As the GUI Set-Up only has an impact on what you see in the User Interface, the GUI Set-Up must follow the privilege rules in order to create consistency between what you see in the GUI and the results you get when navigating around in the STEP workbench. GUI Set-Up settings apply **only** in the workbench.



GUI Setup on Groups

GUI Set-Up is maintained in [System Setup]: **Users & Groups** -> **Groups**

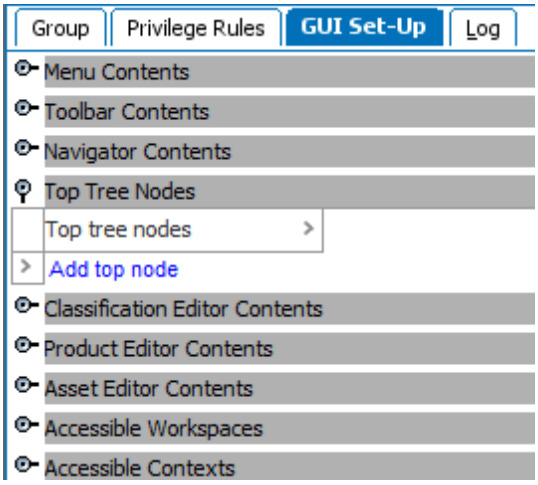
GUI Set-Up on User Groups makes it possible to customize menu contents, navigator contents, accessible workspaces and contexts and Classification, Product and Asset editor contents for Users in the Group.

1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group.

   A **Group** Editor appears.

2. Click the **GUI Set-Up** tab.

   An editor appears listing settings that can be applied on a Group.



Set-Up Options	Description
Menu Contents	In the field 'Menu Contents' it is possible to customize menu contents. There are three settings for each menu item: • **'On'** means that the menu item will be shown • **'Off'** means that the menu item will not be shown • **'Advanced'** means that the menu item will be shown in a sub menu item named "Advanced". "Advanced" will typically be selected for those menu items that are rarely used.
Toolbar	In the field "Toolbar Contents" it is possible to customize the icons or menus to be shown in

Set-Up Options	Description
Contents	the tool panel. There are two settings for each menu or Icon: • **'On'** means that the menu / icon will be shown in the toolbar • **'Off'** means that the menu / icon will not be shown in the toolbar
Navigator Contents	In the field 'Navigator Contents' it is possible to specify the tabs to be shown in the Navigator section of the STEP Workbench: • Tree Navigator • Search Navigator • Tasks Navigator • Process Navigator • System Setup Navigator • Bookmarks Navigator • Current Task Navigator There are two settings for each Navigator : • **'On'** means that the Navigator will be shown and can be accessed. • **'Off'** means that the Navigator is hidden and cannot be accessed
Top Tree Nodes	In the field 'Top Tree Nodes' it is possible to customize the tree nodes to be shown in the "Tree Navigator.
Classification Editor Contents	In the field 'Classification Editor Contents' it is possible to customize the fields and tabs to be shown in the Classification Editor.
Product Editor Contents	In the field 'Product Editor Contents' it is possible to customize the fields and tabs to be shown in the Product Editor.
Asset Editor	In the field 'Asset Editor Contents' it is possible to customize the fields and tabs to be shown

Set-Up Options	Description
Contents	in the Asset Editor.
Accessible Workspaces	In this field it is possible to limit the number of Workspaces that can be accessed.
Accessible Contexts	In this field it is possible to limit the number of Contexts that can be accessed.

GUI Setup for Users

GUI Set-Up for Users is maintained in [System Setup]: **Users & Groups**.

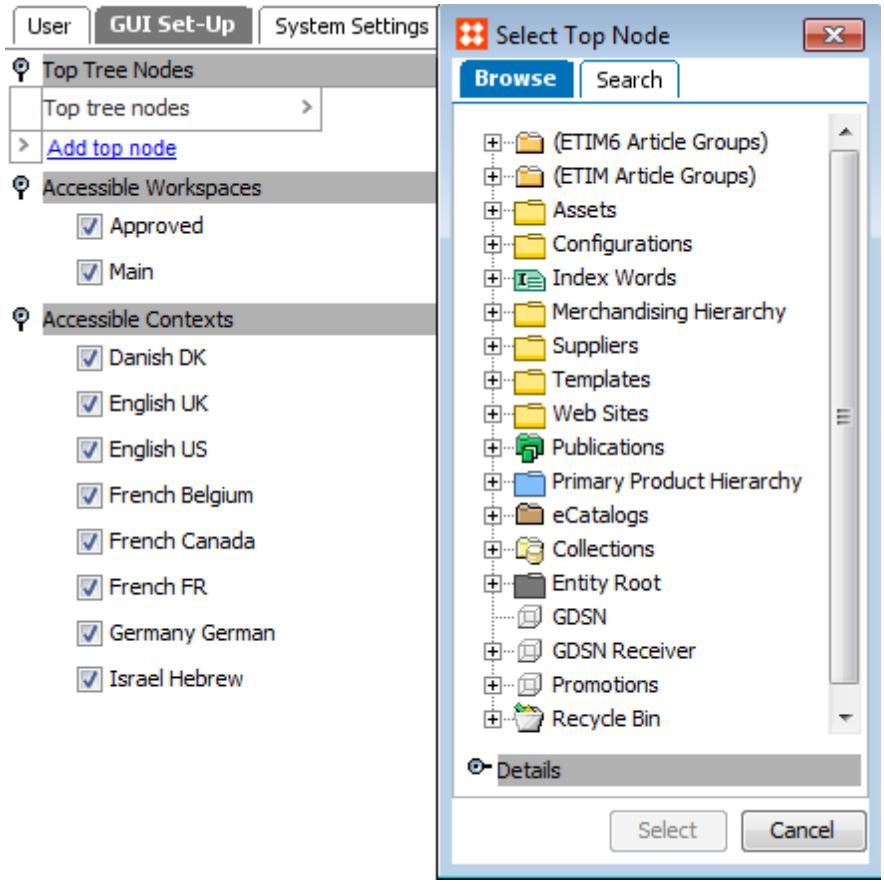
GUI Set-Up for a User makes it possible to customize top tree nodes for the user and provides an administrator with a way to make relevant workspaces and contexts accessible.

1. In [System Setup], unfold **Users & Groups**, unfold the relevant Group, and then click a User.

   A **User** Editor appears.

2. Click the **GUI Set-Up** tab.

   An editor appears listing settings that can be applied on a User. The **Add top node** was selected in this image example.



Set-Up Options	Description
Top Tree Nodes	The field 'Top Tree Nodes' allows you to customize the tree nodes to be shown in the 'Tree Navigator'. You are not required to select the Root Node in the hierarchy, for example: Primary Product Hierarchy, but you can select the specific nodes that the user needs to see in order to perform their tasks. **Important:** Remember that GUI set-up should align with the User's privileges.
Accessible Workspaces	In this field it is possible to limit the number of Workspaces that can be accessed.
Accessible Contexts	This Allows an administrator to select only those Contexts that are necessary for the User to work in.

Adding a User to a Group

Users are added to Groups in [System Setup]: **Users & Groups**.

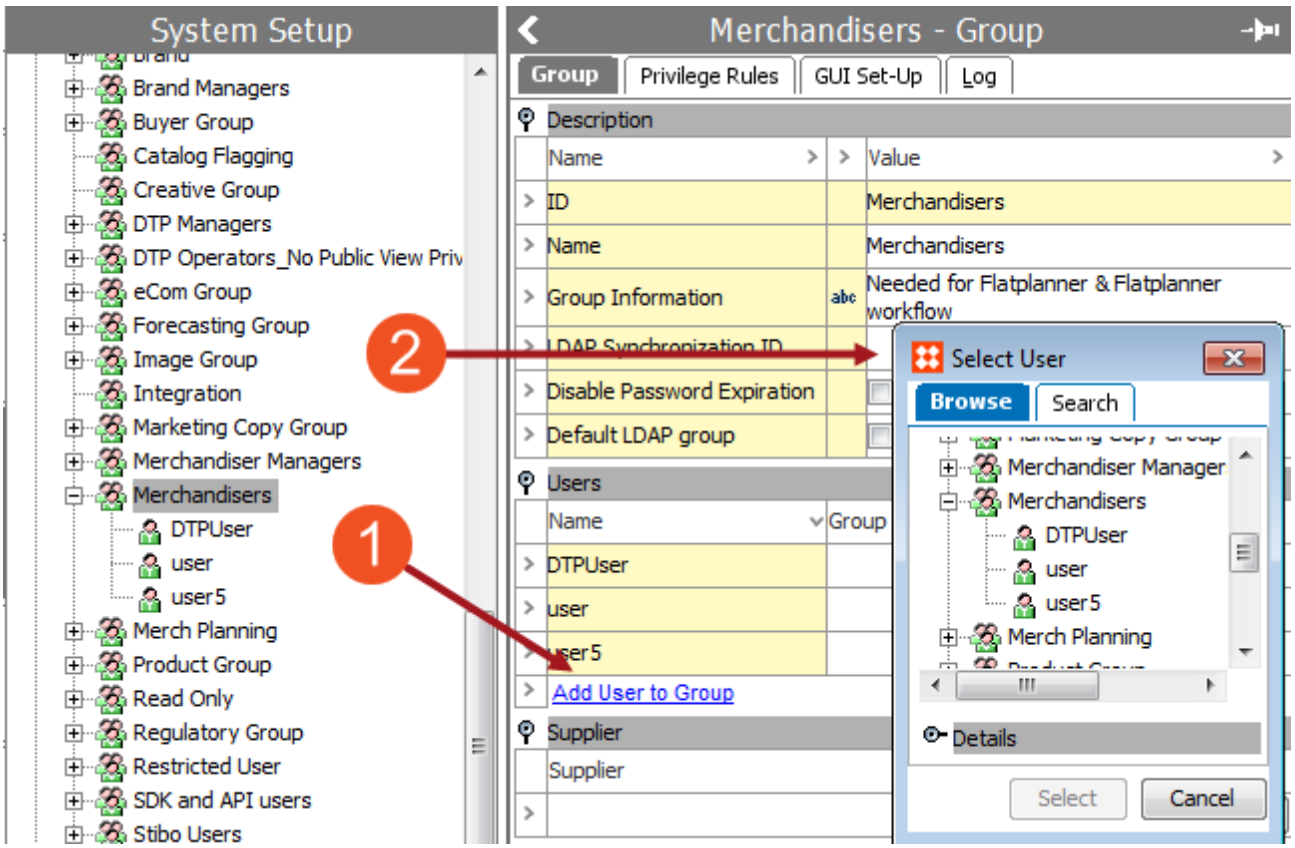
Adding a User to a Group will provide the User with the privileges assigned to this Group.

1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group.

   The Group appears in the **Group** Editor.

2. Under **Users**, click **Add User to Group**.

   A **Select User** dialog box appears.



3. Select the relevant User, and then click **Select**.

   A User has now been added to the selected Group, and the User appears under **Users**.

Note: A User can be a member of as many Groups as needed.

Changing User Password

User passwords are changed in [System Setup]: **Users & Groups**.

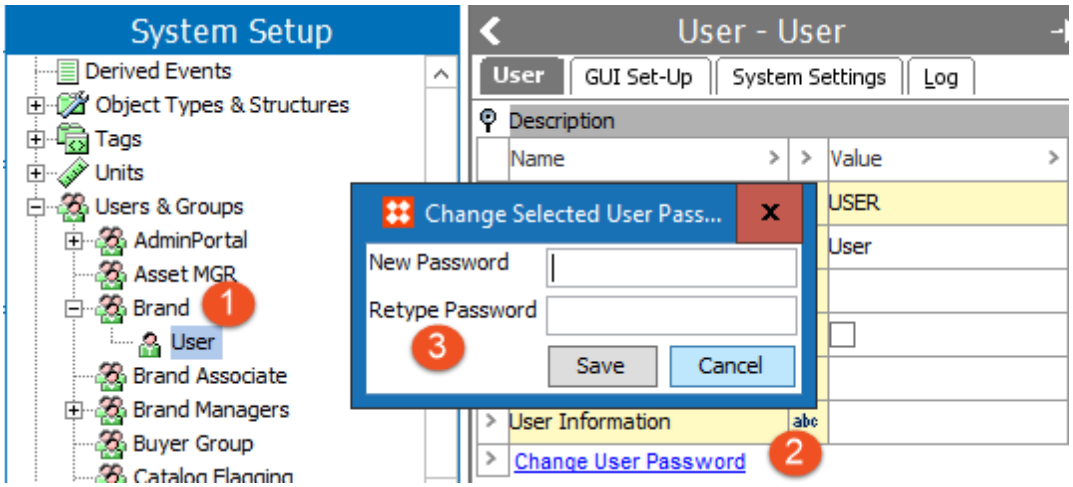
Note: Privileges to change passwords is required.

1. In [System Setup], unfold **Users & Groups**, unfold the relevant Group, and then click the User for the password to change.

   The **User** Editor appears.

2. Under **Description**, click **Change User Password**.

   A **Change Selected User Password** dialog box appears.



3. Type the new password.

On a standard STEP system passwords may contain up to 24 characters. Special characters are not allowed, such as asterisk (*) and percentage (%), etc. Passwords are case sensitive. Only a top-level User has the permissions to set a User's password. For security reasons, the password is stored encrypted in the system so that even an administrator cannot view it.

Note: On Users and Groups root node you can define a password security policy to be used.

4. Type the new password again.
5. Click **Save**.

   A password has now been changed for a User.

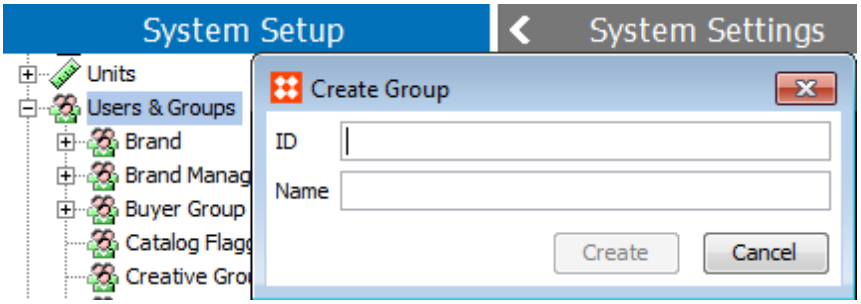
Note: A User may change his or her own password, if privileged to do so. For more information, see Online Help - Common Functionality.

Creating a Group

Groups are created in [System Setup]: **Users & Groups**.

1. In [System Setup], click **Users & Groups**.
2. In the **Object** menu, point to **Create** and then click **Group**, or right-click on **Users & Groups**, and then click **New Group**.

   A **Create Group** dialog box appears.



3. Type a name for the new group, and then click **OK**.

   The new Group appears in the Group Editor.

Creating a User

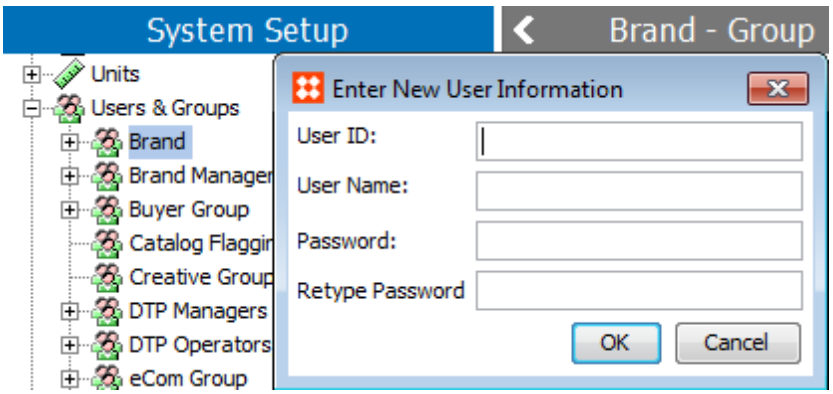
Users are created in [System Setup]: **Users & Groups**.

Note: If information fields should be available for Users, they must be applied as Attributes on the Object Type "Operator user-type root", such as e-mail address and telephone number.

For more information about Object Types, see Object Types and Structures in System Setup.

1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group.
2. In the **Object** menu, point to **Create** and then click **User**, or right-click on a Group and select **New User** from menu.

   An **Enter New User Information** dialog box appears.



3. Type a User ID and a User Name.

   Note: User ID and Name must be unique. Values are case insensitive and special characters should not be used.

4. Type the Password.

   Note: Passwords may have up to 24 characters. Special characters are not allowed, e.g. asterisk (*) and percentage (%), etc. Passwords are case sensitive.

5. Type the Password again.
6. Click **OK**.

   The new User has now been created in the Group and appears in the Group Editor.

Deleting a User or a Group

Users and Groups are deleted in [System Setup]: **Users & Groups**.

Note: It is not possible to delete a Group that contain Users. These must be removed from the Group first. For more information, see Removing a User from a Group in Users and Groups.

1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group or User to be deleted.
2. In the **Object** menu, click **Delete**.

   A **Delete** dialog box appears.

3. Click **Delete**.

   The selected User or Group has now been deleted.

   Note: A deleted User will no longer have access to the workbench or Web UI. A deleted Group will no longer be available for the Users.

Duplicating a User

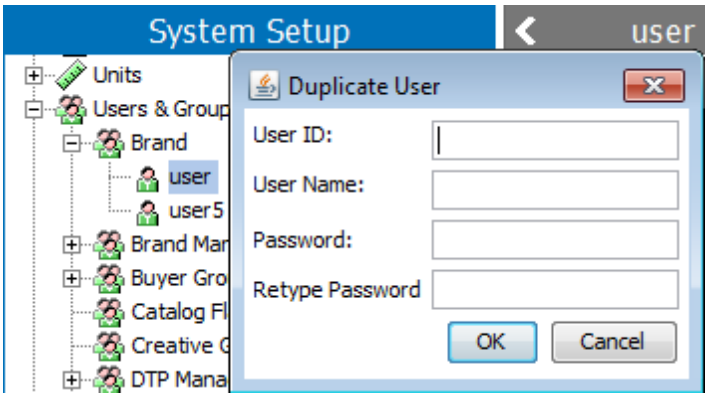
Duplicating a User is done in [System Setup]: **Users & Groups**.

Duplicating a User is making a copy of a User including all STEP privileges and settings.

This function is useful, when a lot of Users are created within the same Group.

1. In [System Setup], unfold **Users & Groups**, and then click the relevant User to be duplicated.
2. Click the **User** tab.
3. Right-click the User to be duplicated, and then click **Duplicate**.

   A Duplicate User dialog box appears.



4. Type a User ID and User Name, and then click **OK**.

Note: User ID and Name must be unique. Values are case insensitive and special characters should not be used.

The User will be copied and pasted into the same Group. Now a User setup has been duplicated to a new User sharing STEP and Staffware privileges and settings.

Note: The duplicated User will have the same password as the original User. If a new password is required, you can Change User Password. For more information, see Changing User Password in Action Sets.

Note: A User may change his or her own password, if privileged to do so. For more information, see Online Help - Common Functionality.

Removing a User from a Group

Users are removed from Groups in [System Setup]: **User & Groups**.

Removing a User from a Group will deprive the User of the privileges assigned to this Group.

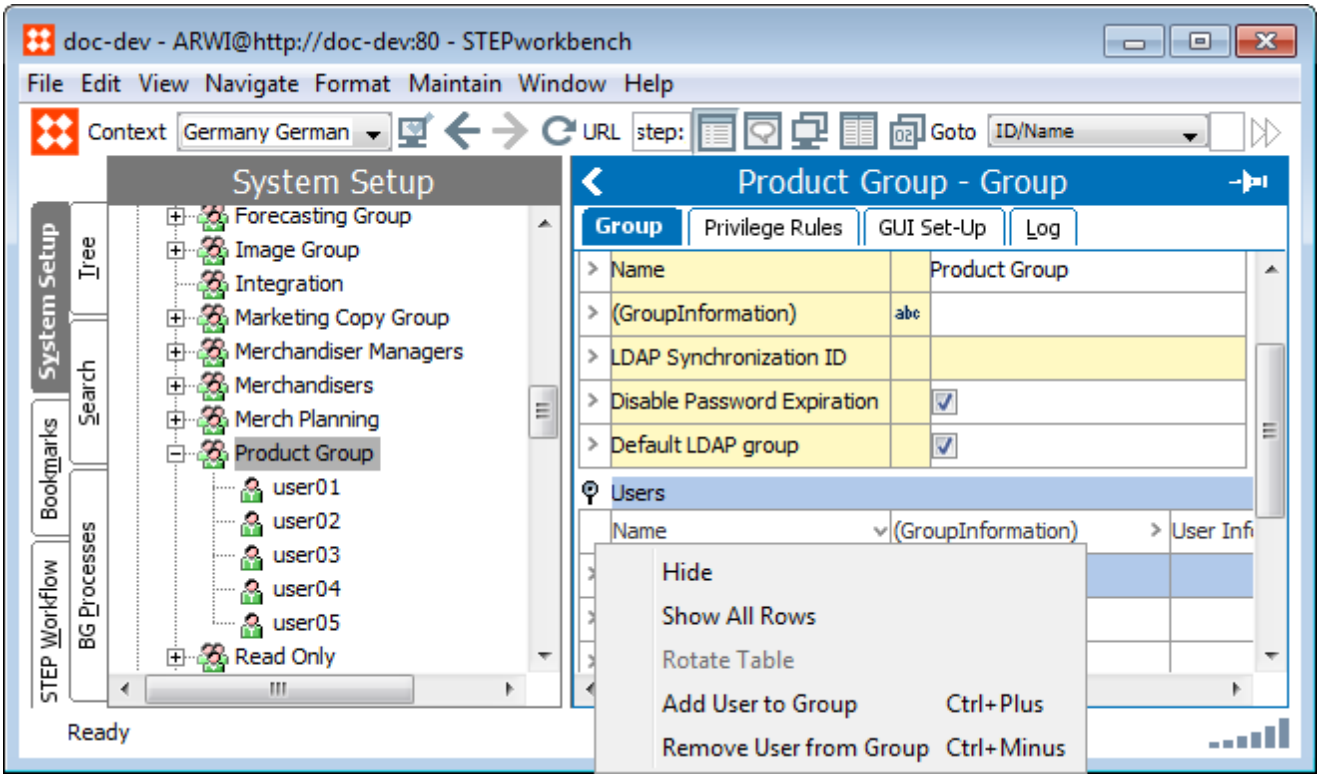
Note: A User can be a member of more Groups, and will still have the privileges assigned to these Groups.

1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group.

   The Group appears in the **Group** Editor.

2. Select the User that needs to be removed from the Group
3. Right-click, and then click **Remove User from Group**.

   A User has now been removed from the selected Group.



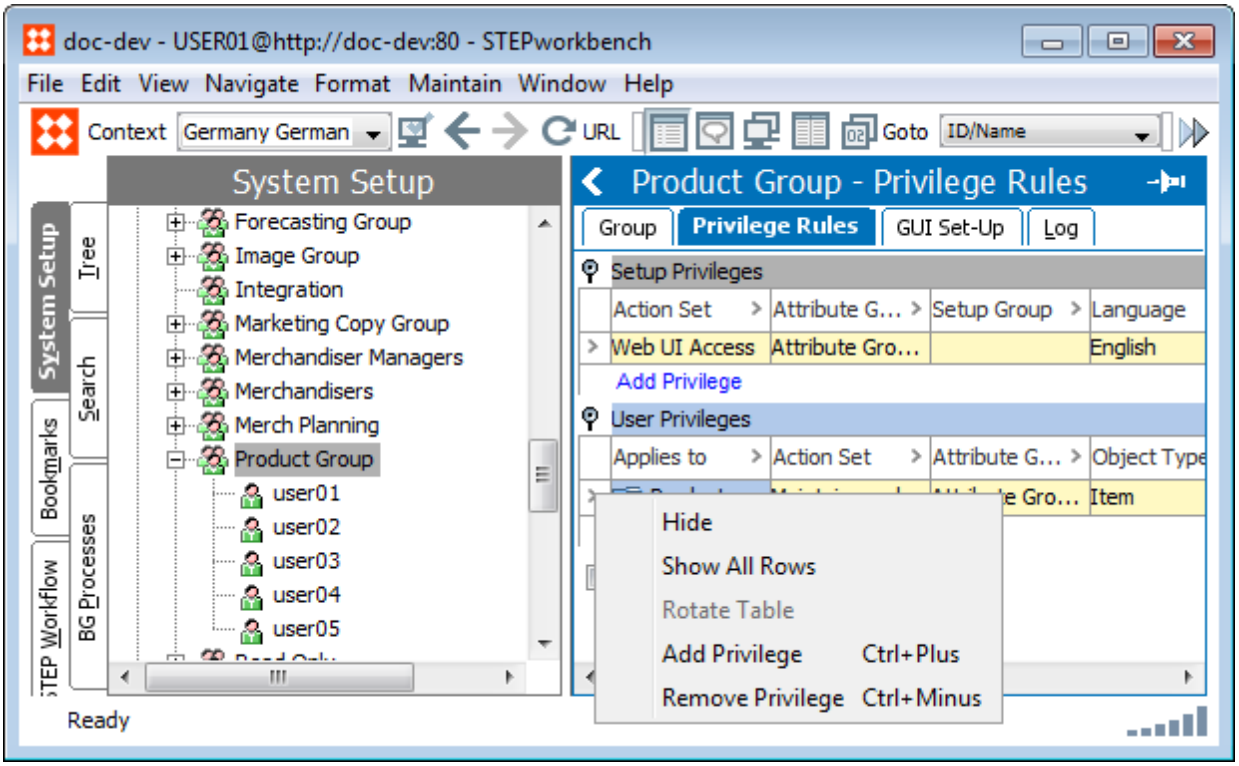
Removing a Privilege Rule from a Group

1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group.

   The Group appears in the **Group** Editor.

2. Click the **Privilege Rules** tab.
3. Click the row header of the Privilege rule to be removed, and then click **Remove Privilege**.

   The Privilege rule will now be removed from the list and will no longer apply to the members of the selected Group.



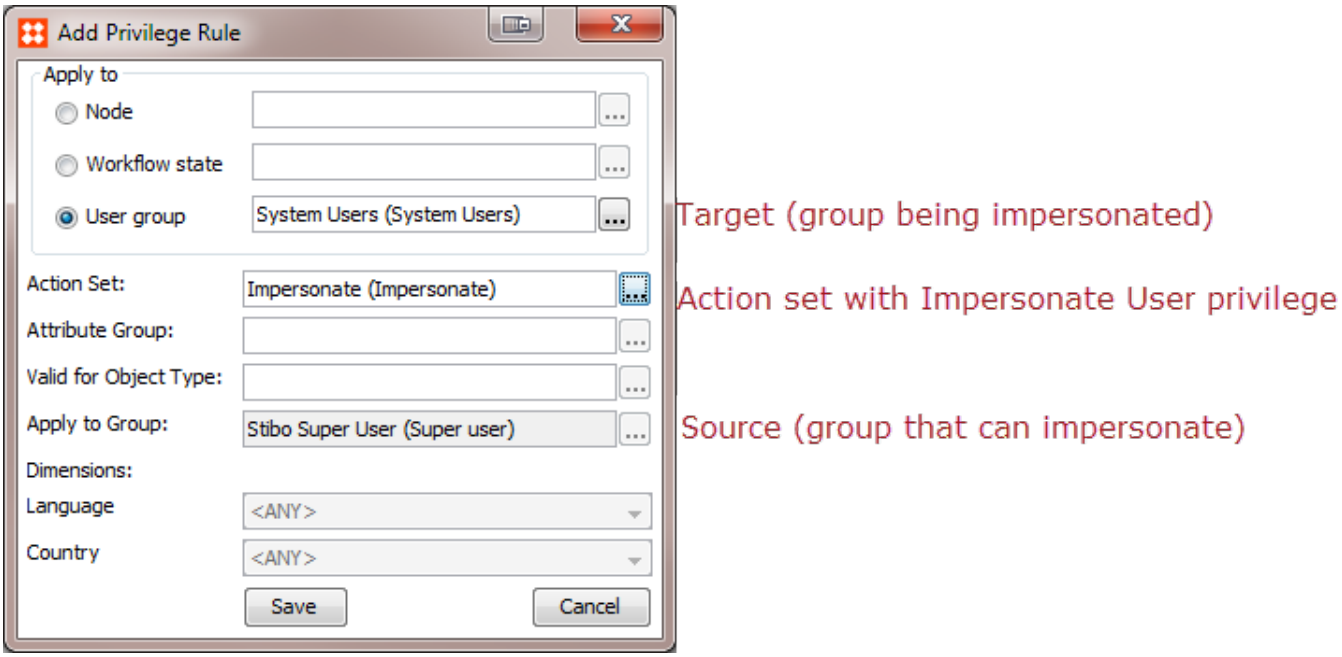
Web UI User Impersonation

In support situations it is often relevant to be able to see exactly what the user experiencing the issue is seeing, including data and error messages. Portal user impersonation enables a privileged user to act as another user, but using their own password so that the impersonated user's password is not revealed. When impersonating someone, all privileges, views, and access are exactly matched to the user being impersonated.

Specific privileges have been created to support impersonation:

- Setup action **Maintain Impersonation** indicates that the user may maintain the impersonation privileges of others. Users with maintain impersonation privileges can assign the Impersonate User privilege to other groups, including specifying the target group that can be impersonated.
- User action **Impersonate User** enables any user with this privilege assigned to be able to impersonate the selected user group identified in the privilege setup.

Note: Applying any privileges to a user group (rather than node or workflow state) will have no effect; impersonation privileges are the only ones applicable to user groups.



Any user with the Impersonate User privilege can impersonate any user in the target user group for which the privilege is configured. Impersonate functionality is added to the Web UI (Portal) via the following components:

- ImpersonateUserWidget can be inserted in the homepage
- CornerBarImpersonateUser can be inserted in the corner bar

The impersonation components are only visible if the user is a member of a user group with impersonation rights for another group.

The impersonator impersonates another user by selecting the user in one of the user impersonation components. The Web UI (Portal) will then start on the homepage, and the impersonator can now act as the impersonated. The impersonator can continue switching between impersonated users and return to act as the impersonator user as needed.

All actions made by the impersonator are logged as actions taken by the user being impersonated. However, for the sake of data governance it is logged in the system log, the impersonator user log, and the impersonated user log when an impersonator starts or stops impersonation.

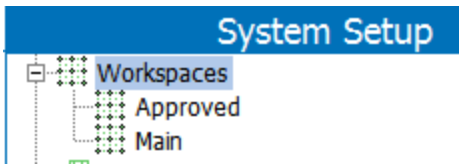
Note: Impersonation can only be used in Web UI.

About Workspaces

A workspace can be perceived as a set of independent 'copies' of workspace revisable database objects. A workspace is thus an area in the database that provides a snapshot of all objects in the database. Objects that are subject to approval can be maintained within one workspace without affecting the snapshot in other workspaces.

Standard STEP installations come with two workspaces, Main and Approved, where Main is the editable "draft" workspace and Approved is a non-editable workspace representing the version of data exposed to downstream systems. Data is reflected in the Approved workspace via the approval functionality.

Workspaces are maintained in [System Setup]: **Workspaces**.



Important: Data maintained in [System setup], such as Attributes, LOVs, Units, etc. are valid for all Workspaces. Objects that are not subject to approval are present identically in all workspaces.

Standard Workspaces

There are two standard workspaces:

- Main
- Approved

The following describes the basic Workspace types:

Workspace	Description	Example
Main	Maintenance area (editable)	Used as a the standard maintenance area for objects.
Approved	Approved area (uneditable)	Used for objects that have been formally approved and are ready for e.g. translation or publishing. Objects are copied to the Approved Workspace from the menu option: Object ->Approve object, or automatically upon system approval events.

Note the following:

- The Approved and Main Workspace are system defined workspaces and cannot be deleted or modified. A system can only hold one Main workspace and one Approved workspace. It is possible to create new Workspace Types based on another Workspace.
- Data in different workspaces can be synchronized. For more information about updating Workspaces, see Online Help about - Updating.

Creating Workspaces

Workspaces are maintained in [System Setup]: **Workspaces**.

When creating a new Workspace, the contents of the selected Workspace will be copied to the new Workspace.

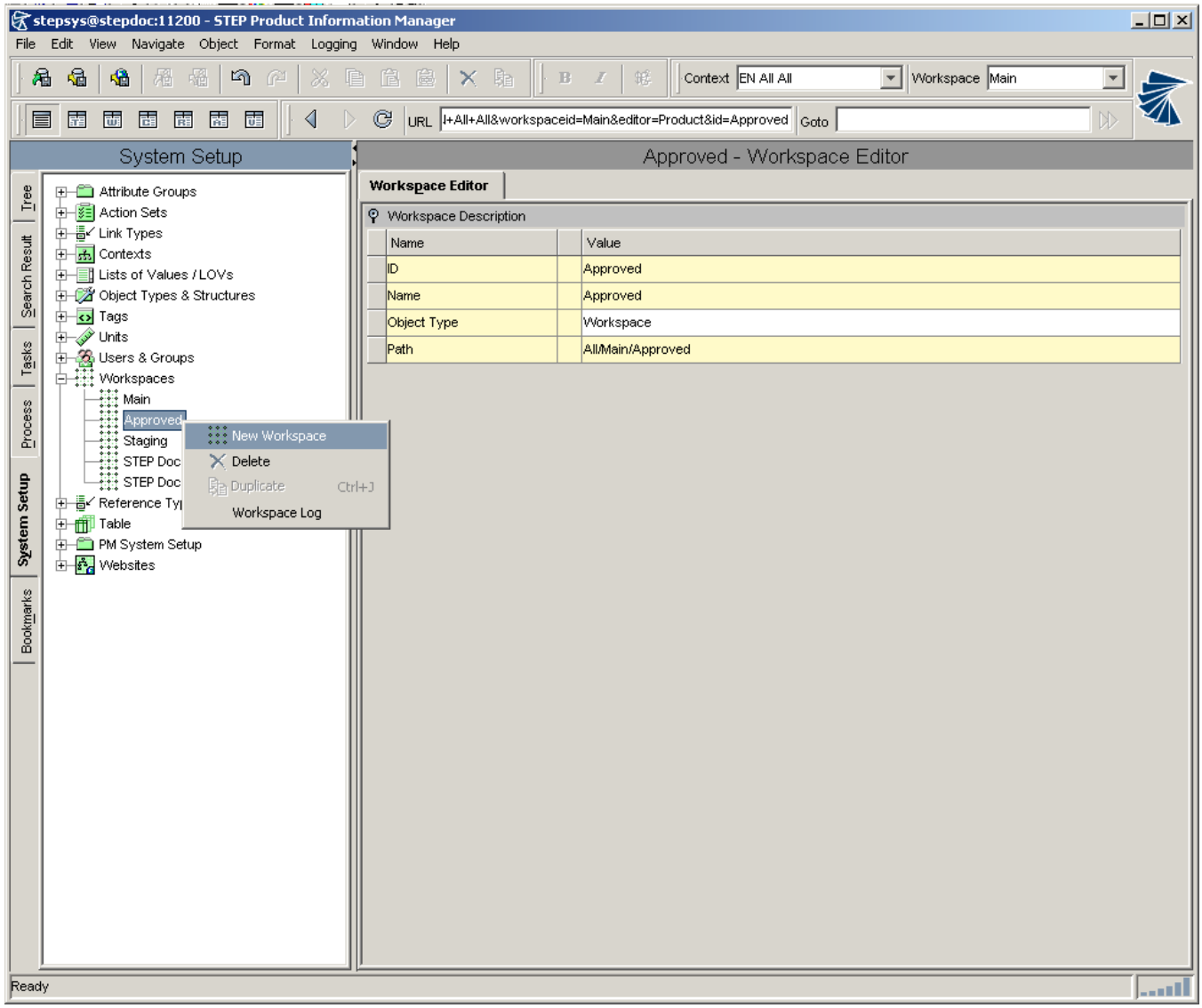
Note: Typically, new Workspaces should not be created as the initially set up Workspaces will be sufficient for the cycle of data maintenance.

For more information about Workspaces, see About Workspaces.

1. In [System Setup], unfold **Workspaces**, and then click the Workspace of which you need create another workspace.

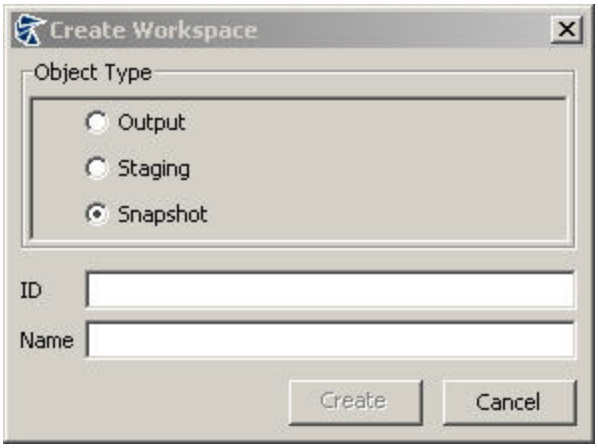
   In this example a Snapshot Workspace is created.

   Note: A snapshot will typically be made of the **Approved** Workspace.



2. In the **Object** menu, point to **Create**, and then click **Workspace**, or right-click and click **New Workspace**.

   A **Create Workspace** dialog box appears.

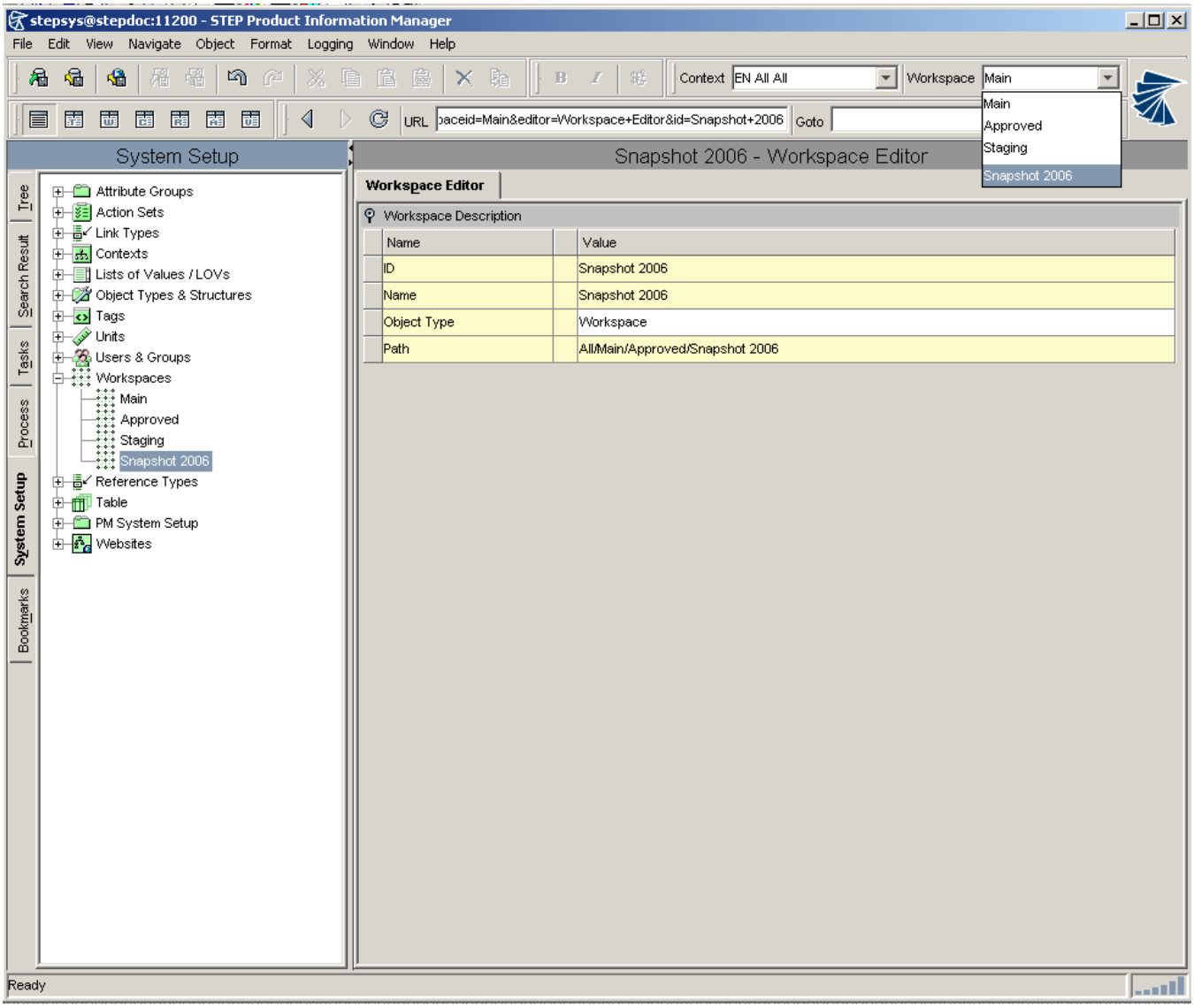


3. Click **Snapshot**.

   Note: When clicking either **Outputs** or **Snapshot**, a terminal (uneditable) Workspace will be created.

4. Type the ID and Name of the snapshot Workspace, and then click **Create**.

   A new Workspace appears in the Workspace list in System Setup and the Workspace selector in the STEP PIM.



A snapshot Workspace has now been created of the data in the selected Workspace. In this example all data from Approved Workspace has been copied to the Snapshot Workspace “Snapshot 2006”. All data in “Snapshot 2006” is read-only and cannot be edited.

The workspace is available for selection in the Workspace list.

Deleting Workspaces

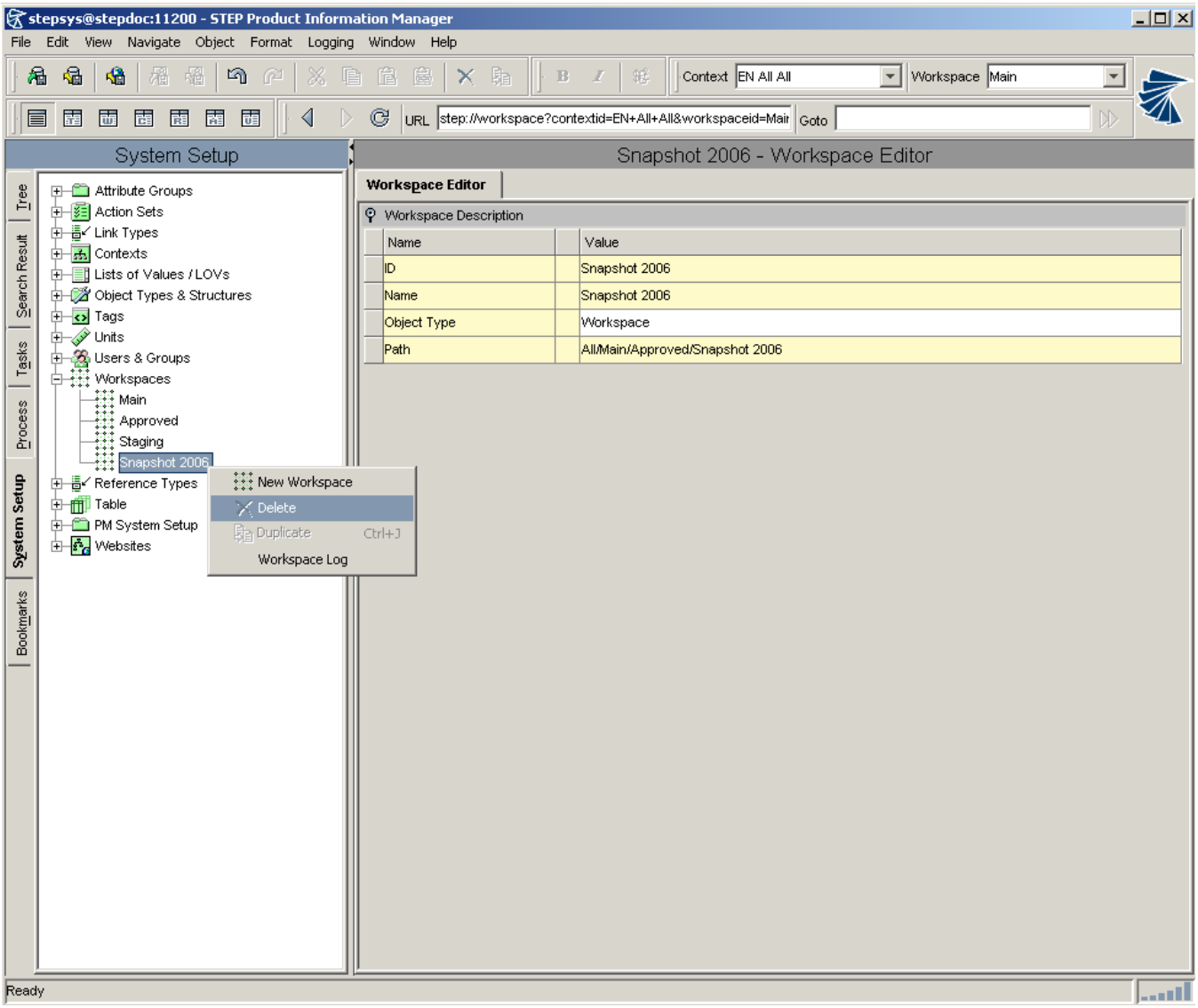
Workspaces are maintained in [System Setup]: **Workspaces**.

Note: The Workspace for a current Object is selected from the **Workspaces** list (upper right corner).

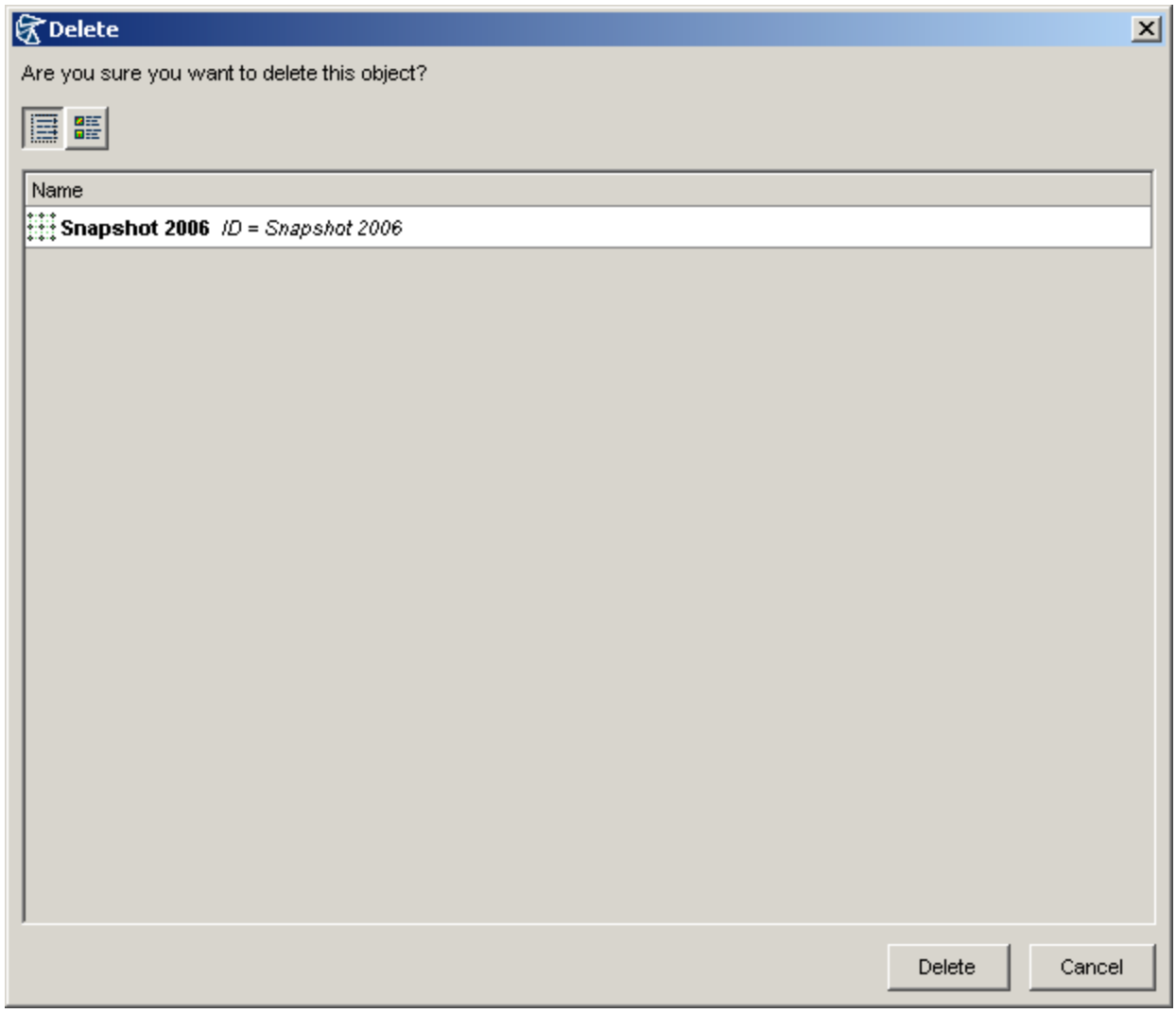
Important: Make sure to transfer data that should be kept to another Workspace before deleting. The initially set up Workspaces cannot be deleted

Note: Deleting a Workspace is very time consuming.

1. In [System Setup], unfold **Workspaces**, and then click the Workspace to be deleted.
2. Right-click, and then click **Delete**.

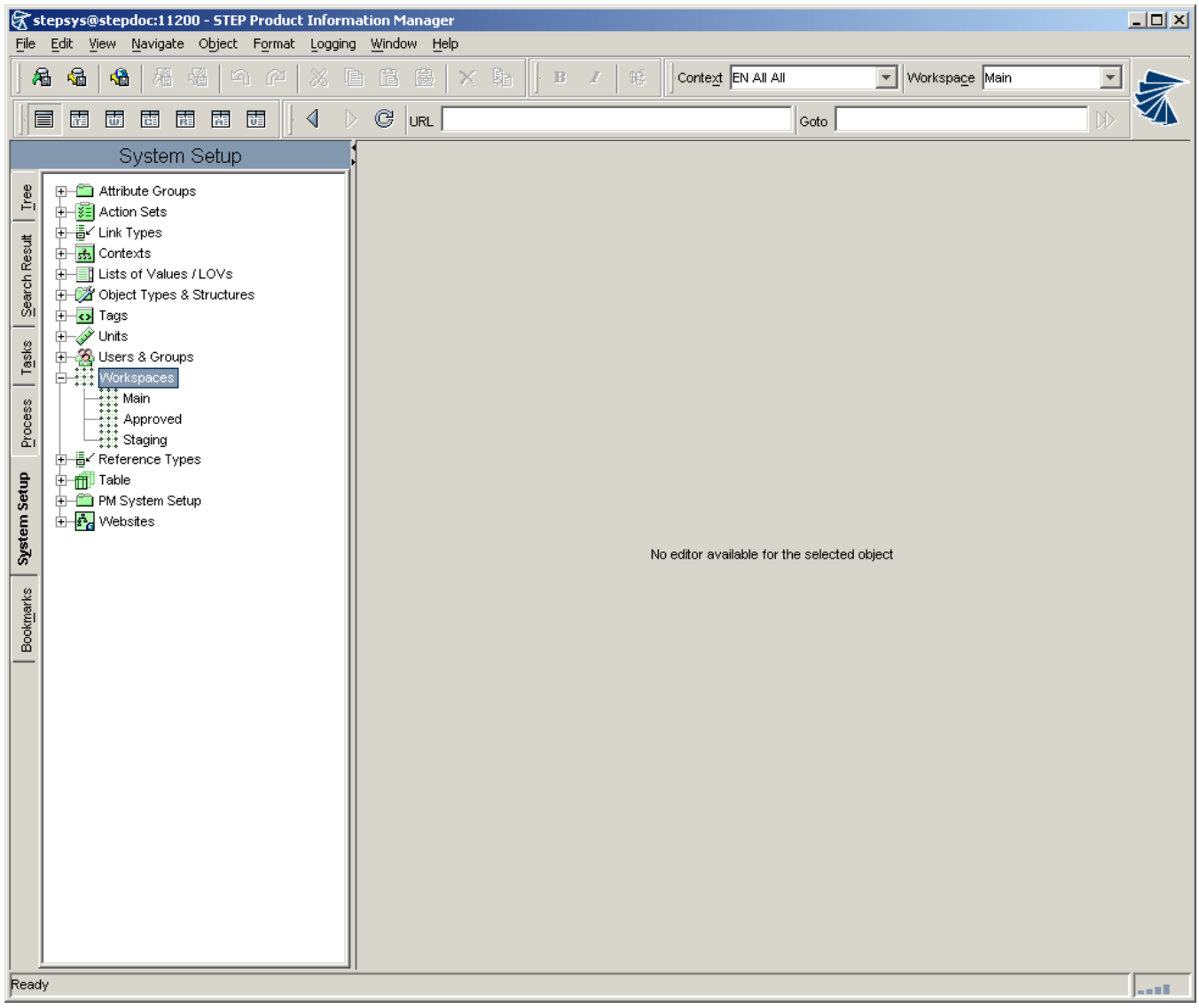


A **Delete Warning** dialog box appears.



3. Click **Delete** to delete the Workspace.

   The Workspace will be removed.



Note: It takes some time for the system to delete an entire workspace.

Note: It is not possible to delete the Main Workspace and the Approved workspace.