

SYSTEM SETUP SUPER USER GUIDE

The logo for StiboSystems, featuring the company name in a white sans-serif font. The 'i' in 'Stibo' has a small icon above it consisting of three small squares arranged in a triangle. The logo is positioned on the left side of the page, partially overlapping a large orange triangle that points towards the right.

StiboSystems

STEP Trailblazer 8.1

Table of Contents

Table of Contents	2
Introduction	15
About This Guide	15
Accessing Short-cut Menus on PC and Mac	15
System Setup	16
Action Sets	17
Recommendations for Action Sets	18
Setup Actions	19
Action Sets	19
Administration	19
Assets	19
Attributes	20
Background Processes	22
Bulk Updates	22
Business Rules	22
Change Packages	22
Contexts	23
Cryptographic Keys	23
Data Quality	23
Deduplication	23
Events	24
GDSN Data Pools	24
Integration Endpoints	26
Link Types	26
List of Values (LOVs)	26

Object Types	28
Print Publication	29
Purging	29
Reports	29
SDK / API Documentation	29
Tables	30
Tags	30
Units	30
Users & Groups and Privilege Rules	31
Value Search	31
Web UI	31
Workflows	32
Workspaces	33
User Actions	34
Assets	34
Collections	36
Commercial Lists	36
Data Quality Functionality	37
eCatalog	38
GDSN Data Pools	38
Products, Classifications, and Entities	39
Publications	45
Translations	48
Web UI	48
Maintaining Action Sets	50
Creating an Action Set	50
Adding Actions to an Action Set	51

Removing an Action from Action Set	52
Attributes	54
Attribute Groups	55
Creating Attribute Groups	55
Creating a Customized View	56
Manually Sorting in Attribute Groups	60
Attribute Group Display Sequence	60
Sorting Attribute Groups	60
Attribute Links	61
Linking Specification Attributes	63
Linking Attributes in Product editor	63
Linking Attributes to Products or Classifications	64
Linking to a Source from Attribute	66
Linking Orphan Attributes	67
Sorting Linked Attributes	71
Setting a Display Sequence Number on the Attribute	71
Setting a Display Sequence Number under Linked Attributes	72
Setting the Priority of a Product Variant Attribute	74
Setting the Attribute Display Sequence on Publication (Green) Hierarchy Objects	74
Attribute Transformations	76
About Attribute Transformations	76
Available Attribute Transformation Rules	81
Maintaining Attribute Transformations	81
Creating an Attribute Transformation Group	81

Creating an Attribute Transformation	81
Applying a Transformation to an Attribute Transformation	82
Date Attribute Transformations During Import and Export	88
Export Transformations	89
Transform date by locale	90
Transform date by pattern	91
Import Transformations	92
Transform date by locale	94
Transform date by pattern	94
Considerations and Limitations	95
Calculated Attributes	96
Differences Between Calculated Attributes and Standard Attributes	97
Value Template Expression Examples	100
Function Editor	103
Accessing the Function Editor	103
Using the Function Editor	106
Function Editor Options	106
Creating Function Templates	109
Dimension Dependencies	111
Inserting Attribute IDs in Functions	113
Inserting Templates	114
Evaluating Functions	115
Calculated Attribute Functions	116
Calculated Value Override	116
Available Function Types	116
Attribute Value Functions	117

Comments	120
Commercial Value Functions	121
Conditional Functions	123
Date Functions	124
List Functions	126
Number Functions	130
Other Functions	132
Text Functions	143
Variables	147
Calculated Attribute Use Case Examples	148
Use Case: Calculated Attribute for Classifications	149
Get All Classifications	149
Get Classifications Based on Type	149
Get Metadata Value Located on the Link From Classification-to-Product	150
Get CPL Classifications Using Classification Product Links	151
Use Case: Calculated Attribute for Overrides	153
Get Overridden By Products	153
Get Overridden Product Object	154
Get Product Override Parents	155
Use Case: Calculated Attribute for References	156
Get List of References	156
Get Value From a Referenced Product	157
Get Metadata Value Located on the Link From Product-to-Product	158
Get Referenced By Using Reference Owner	158

Get Reference Type ID	160
Getting the Reference Source or Target products' information on the link	161
Conditional Attribute Display	163
Terminology	163
Creating the Conditional Attribute	163
Activating a Conditional Attribute	165
Configuring Conditionally Valid Attributes	166
Description Attributes	166
Specification Attributes	166
Writing Conditional Expressions	167
Conditional Attribute Example	168
Description Attributes	170
Meta Attributes	170
Attribute Metadata on Attributes	171
Steps to Creating a Metadata Attribute on an Attribute	171
Description Attributes and Object Types	174
Linking Description Attributes to an Object Type	175
Dimension Dependent Attributes	176
Dimension Dependent Attribute Values	176
Non-Dimension Dependent Attribute Values	177
Setting Dimension Dependency on Attributes	178
Value Inheritance in Dimension Dependent Attributes	178
Showing Source Context (Parent Dimensions)	180
Attribute Display Sequence	183

Attribute Display Sequence Setup	183
Setup Display Sequence Attribute	183
View Menu: Choose Attribute Display Sequence Attribute	184
Sorting Attributes	184
Setting a Display Sequence number on the attribute	185
Setting a Display Sequence number under Linked Attributes	185
Selecting a Personal Attribute Display Sequence Attribute	187
Display Sequence Attribute and the Product Editor Display	188
Tooltip	188
Display Sequence Attribute & Classifications	188
Setting the Priority of a Product Variant Attribute	189
Display Sequence Attribute & Tables	190
Setting the Attribute Display Sequence on Publication (Green) Hierarchy Objects	191
Handling Duplicated Attributes in Web UI	193
Create Attribute Group(s) to Hold Attributes Known to Exist in Multiples Groups	193
Encrypted Attributes	193
Cryptographic Key	193
Password requirements	194
Creating Encrypted Attributes	195
Properties of attributes that cannot be used with encryption	195
Accessing encrypted data	195
Restrictions of Encrypted Data	195

Maintaining Attributes	196
Attribute Types	197
Converting Attributes from Date to ISO Date	198
Merging Attributes	198
Switching Attribute IDs	198
Creating Attributes	200
Externally Maintained Attributes	205
Revision Control	205
Sharing LOVs	205
Setting as Externally Maintained	205
Full Text Indexable Attributes	207
Hierarchical Filters on Attributes	209
Relation between LOV filters	209
Viewing and editing LOV Filters	210
Options affecting Performance	210
To setup a hierarchical filter:	210
Validity on Description Attributes	214
Validity on Specification Attributes	216
Reparenting Attribute Groups	218
Manual Reparenting	218
STEPXML Reparenting	222
Mandatory Attributes	238
Approval Mandatory Attributes (Global) / Visual Indicator in Workbench	238
Workbench Visual Indicators	238
Approval Mandatory Attributes	241
Business Rules for Mandatory Attributes	242

Smartsheet Mandatory Attributes	242
Web UI Screen Mandatory Attributes / Visual Indicator in Web UI	244
Web UI Visual Indicators	244
Enforcement of Mandatory Attributes	246
Workflow Mandatory Attributes	246
Merging Attribute Values	248
Verify Attributes to be Merged	248
Merge Values Using Bulk Update	251
Single and Multi-valued Attributes	251
Deleting Attributes from Objects	253
Delete Object from Attribute References Tab	253
Delete Attribute from Object References Tab	253
Editing Attribute Values	253
Inherited and Local Values	255
View Filter Options	255
Attributes with Units	255
Editing Multi-Valued Attributes	257
Entering and Overwriting Values	259
Examples	259
Entering Values	259
Overwriting Values	260
Example:	260
Specification Attributes	260
Inherited Attributes	262
Orphan Attributes	263
Validation Rules	264

Editing Validation Rules	268
Input Masks	270
Example Input Mask Use Cases	270
Definitions of Input Mask Characters	271
Static Mask Characters	272
Setting LOV Filters	274
Setting LOV Filter	275
Viewing Attribute Values	277
Background Process Queues	279
Default Configuration	279
Changing the default configuration	280
Finding the Process Type ID	280
Editing the configuration file	282
Scheduled Process Properties	284
Data Containers	285
Composite Concept	285
Using Data Containers	285
Considerations and Limitations	286
Setting Up Data Containers in Workbench	287
Prerequisites	287
Adding and Configuring Data Container Types	288
Adding and Maintaining Data Containers	294
Dimensions, Dimension Points, and Contexts	296
Dimensions and Dimension Points	297
Dimensions	297
Dimension Points	297

Dimension Dependent Data	298
Dimension and Dimension Point Best Practices	300
Dimension Do's and Don'ts	300
Dimension Do's	300
Dimension Don'ts	300
Additional Attributes as an Alternative to Additional Dimensions	300
Language Dimension	301
.....	302
Country Dimension	302
Business Unit Dimension	303
Market Dimension	303
Dimension Point Do's and Don'ts	304
Dimension Point Do's	304
Dimension Point Don'ts	305
Maintaining Dimensions and Dimension Points	306
Creating a Dimension	306
Creating a Dimension Point	308
Deleting a Dimension Point	309
Deleting a Dimension	310
Selecting a Dictionary for a Language Dimension Point	311
Adding a Dimension Dependency After Loading Data	312
Add a Dimension Dependency to an Attribute After Data Has Been Loaded	312
Add a Dimension Dependency to a Reference Type After Data Has Been Loaded	313

Contexts	314
Dimension Dependent Data	315
Context Best Practices	316
Context Do's	316
Context Don'ts	316
Maintaining Contexts	318
Creating a Context	318
Editing a Context	320
Deleting a Context	320
Hidden Values	320
Specifying a Default Context	321
Context Locales	324
Locale Do's	325
Events	326
Generating Events	326
Core Events	328
Derived Events	330
Prerequisites	330
Using a Derived Event	330
Using a Derived Event Type in an OIEP	330
Adding a Derived Event Type	332
Event Processors	333
Configuring an Event Processor	333
Initial Setup for Event Processors	334
Prerequisites	334
Create Setup Group Types for Event Processors	335
Link Event Processor Object Types to Setup	336

Group Types	
Create an Event Processor Setup Group	337
Configure Action Sets and Privileges	339
Create Event Processor Action Sets	339
Configure Event Processor Privileges	339
Creating Event Processors	341
Expand LOV Value Changes Processor Parameters and Triggers	347
Prerequisites	347
Parameters	347
Event Triggers	347
Image Cache Processor Parameters and Triggers	349
Prerequisites	349
Parameters	349
Event Triggers	350
Matching Processor Parameters and Triggers	353
Prerequisites	353
Parameters	353
Event Triggers	354
Revision Management Processor Parameters and Triggers	356
Prerequisites	356
Parameters	356
Event Triggers	357
Enabling Event Processors	358
Event Processor States	358
Editing an Event Processor	359

Deleting an Event Processor	359
List of Values (LOVs)	360
Values by Reference	360
Revision Control	360
Validation Base Type	361
Working with Large LOVs	361
Creating an LOV Group	362
Creating an LOV	363
Adding Values to an LOV	369
Adding a Single LOV Value	369
Adding a Group of LOV Values	370
Adding IDs to Values in LOV	376
Auto-generate Ids option	376
Import File option	377
Merge LOVs	382
Merging LOVs	382
Merge Values in LOV	382
Merging LOVs	383
Merging Values in an LOV	386
Deleting LOVs	390
Managing Revisions in STEP	391
Generating Revisions	391
Important considerations for revisions:	393
Setting the Timing Threshold for Automatic Revisions	393
Reverting to Past Revisions	394
Purging Revisions	395
Revision Maintenance and Cleanup	395

Object Types and Structures	398
Attributes Valid for Object Types	400
Autogenerate using Name Pattern and ID Pattern	401
Basic Object Types	403
Object Type Dimension Dependencies	405
Object Type Icons	406
Editing an Object Type Icon	406
Maintaining Object Types	408
Creating an Object Type	408
Deleting an Object Type	408
Unlinking Object Types	409
Updating Object Types from Another Object Type	409
Maintaining Dimension Dependent Object Types	410
Making an Object Type Dimension Dependent	410
Removing Dimension Dependency From an Object Type	411
MIME Types	414
Setting MIME Types for Object Types	415
Modifying Manually Sorted	418
Object Types for Classification, Product, Assets and Entities	420
Object Type Hierarchy	422
Classification and Product Hierarchy Structures	422
Fixed Hierarchy Structure	424
Variation in Hierarchy Depths	425

Multiple Parents	426
Flexible Hierarchy Structure	427
Creating a Standard Publication Object Type Hierarchy	428
Creating a Publication Group Object Type	430
Creating a Publication Object Type	432
Creating a Publication Section Object Type	434
Product-Override Object Types and Commercial Object Types	436
Creating a Commercial List Type	437
Creating a Product-Override Object Type	438
Enabling a Product-Override	439
Owns Product Links on Alternate Classifications Object Type	440
Owns Product Links Affects STEPXML	441
Changing Owns Product Links	441
Reference Target Lock Policy on Object Types	443
Revisability on Entity Object Type	445
Product Variants	447
Setting Up Product Variants	448
Add the object type to the component model	448
Set up a meta attribute	449
Product Variant Family Attributes	450
Specify variant and non-variant attributes	451
Using Variants List View	453
To view product variants in list view	453
To delete a variant	453

Product variants list in the portal	453
Using the Variants Matrix View	454
To view product variants in matrix view	454
To delete a variant	454
To create a new variant	454
Creating Variants Through Web UI	456
Business Condition: Validating Product Variant	458
Recycle Bin	459
Reference and Link Types	460
About References Types	460
Direction of a Reference	460
Metadata Attributes	462
Multiple references of different types between same two objects	464
Mandatory References	465
Externally Maintained References	465
Dimension Dependencies for Reference Types	466
Suppression of a Reference	466
Using Context Viewing Mode	466
Inheritance of References	467
Using the Parent / Child Relationship with Entity to Entity References	470
Reference Types in Attribute Groups	472
Creating Reference Types	473
Changing Reference Types Settings	479
Changing Valid Source and Target Object Types for a Reference Type	480
Add Reference Types to Attribute Groups ...	481

Adding References on the References and Referenced By Tabs	482
About Product to Classification Link Types ..	484
Metadata Attributes	485
Multiple Product to Classification Links	486
Restrict Source and Target Object Types .	486
Mandatory Links	486
Externally Maintained Links	486
Dimension Dependencies	486
Inheritance of Links, Specification Attributes and Tables	486
Product Classification Link Types in Attribute Groups	487
Creating Product to Classification Link Types	488
Changing Product to Classification Link Types Settings	493
Changing Valid Source and Target Object Type for a Product to Classification Link Type	495
About Product to Attribute Links and Classification to Attribute Links	496
Setting Dimension Dependency on Product to Attribute Link Type or Classification to Attribute Link Type	497
Applying Metadata Attributes to Product to Attribute Links or Classification to Attribute Links	497
About Reference Visibility	498
Editing Meta-attributes on References	498
To Specify Global Visibility for a Reference ..	498
To Specify Local Visibility for a Reference ...	500

To Suppress a Reference	503
Setup Groups	506
Creating Setup Group Object Types	507
Defining Type of Objects to be Stored in Setup Groups	509
Creating Setup Groups in System Setup	511
Setting Up Privileges Sample	513
Single User Mode	520
Operations that Cause Single Update Mode	521
Tags	523
Tag systems in STEP	523
Types of Tags in STEP	523
Special Character	524
Character Tags	524
Style Tags	524
Footnotes	524
Hyperlinks	525
Working With Tags	525
Creating Tag Groups and Tags	526
Tag Groups	526
Tags	527
Creating Tags	527
Considerations for Specific Tag types	528
Configuring / Rendering Tags	532
Name	532
Tag	532
Rendering	532
Show Character Tag	534

Keyboard Shortcut	534
Special Consideration When Rendering Tags:	534
Configuring Output Formatting	535
Output Formatting Flipper	537
'Pass Through' Setting	539
Tag Examples	540
Inserting a Tag	540
Unique Keys	544
Key Formula Example:	544
Key Formula Example	544
Important Notes:	545
Creating and Deleting Keys	546
To Create a Key	546
To Delete a Key	548
Activating and Deactivating Keys	549
Activate a Key	549
Deactivate a Key	549
Checking Keys	551
Creating Products with Active Keys	552
Matching Unique Keys During Import	553
Units	554
About Units	554
Unit Groups	555
Unit Conversions	555
Example of Conversion Factor and Conversion Offset	556
Maintaining Unit Groups	557

Creating a Unit Group	557	Disable auto-cleanup of thumbnail cache	580
Creating a Unit	558	Transformation Lookup Tables follow asset dimension dependency	580
Setting up a Unit Conversion	559	Asset push attribute macro keys	581
Removing a Unit Conversion	561	Calculated Attribute Settings	582
Removing a Unit or Unit Group	561	Cryptographic Keys	583
Users and Groups	563	Creating cryptographic keys	583
Working with User Groups	565	Password Requirements:	584
Creating a User Group	565	Validation Templates	586
Deleting a Group	566	To create a Validation Template:	586
Working with Users	567	Product Information Manager Default Settings	588
Creating a User	567	Enforce Mandatory Check for Attributes, References and Links	588
Adding a User to a Group	568	Product Editor, Group attributes by top group	589
Duplicating a User	569	Localize numbers with thousand delimiter when localizing exports	590
Removing a User from a Group	570	Localize dates when localizing exports	591
Changing a User Password	572	Report Save As CSV Character Set	591
Changing a User Password - System Administrator	572	Default Attribute to use as Display Sequence Attribute	592
Changing a User Password - User	572	Default Completeness Metric	593
From the Workbench:	573	Conditional Validity Attribute	593
From the Web UI:	573	Block Attribute Groups with more than 1000 attributes	593
System Settings	575	Use full pathname for classes on Product References Tab	593
Classification Hierarchy Settings	576	Pass through unconverted Special Characters and Tags (Y) or discard them on output (N)	594
Dimension Dependencies	576	Product Attribute Help metadata attribute ...	594
Image and Document Settings	579		
Dimension Dependencies	579		
Store assets and DTP documents in	580		
DTP asset source	580		
Pregenerate thumbnail cache on upload	580		

Attribute indicating the Priority of Product Variant Attributes	595
Show both name and ID in the PIM navigator	595
Comma separated list of node types for which to show both name and ID	595
Attribute Header Column Width	596
Reference Type Column Width	596
Translation Settings	598
Translation File Pattern	598
Should "Filter spreadsheet" option be shown in Translation Wizard.	598
Default value for "Filter spreadsheet" option in Translation.	599
Terms List Settings	600
Add the products from the selection list import file directly instead of looking up leaf products	600
Web Services	601
Tag format for Web Services and REST APIs	601
Web UI Settings	602
Default context	602
Default workspace	602
Web UI supplier classification object type	603
Web UI supplier products classification object type	603
Assets classification object type	604
Batches classification object type	604
Batch classification object type	604
Excel-template asset	605
Proof view stylesheet attribute	605

Step-users Web UI batches folder	605
Step-users Web UI assets folder	605
Context Help metadata attribute	605
Link type for vendor classification to product link	606
Enable all-view for users that are a member of multiple suppliers	606
Primary Image Type	607
Flatplanner Default Settings	608
Enable keep layouts for publications	608
Enable calculation of border rectangle space usage for actual pages	609
Show frame information on flatplans with layout pickup	609
DTP Settings	611
Use Publication Name prefix when naming DTP documents	611
Calculated DTP Table width includes left and right rules	611
Attribute Display Sequence	612
Attribute Name Style Meta-attribute and Attribute Value Style Meta-attribute	613
Asset Order Meta-attribute	613
Table Cell Skip Empty Cell	613
JPEG DPI	614
DTP Page - preview image	614
Inherit Default Product Template through product hierarchy for Product Overrides.	614
Reapply Flatplan rules when adding product to non-empty frame	615
Table Defaults	616

Index Word Settings	617
Revisability Settings	618
Threshold (hours) for changes by same user to generate a new revision	618
Business Rule Approve Overview	619
Security Policy	620
GUI Localization	622
Setting up a GUI Localization for a User	623
Privilege Rules	625
Adding User Privileges for a Group	629
Add Privilege Rule Options	630
Adding Setup Privileges for a Group	635
Add Setup Privilege Rule Options	635
Editing Privilege Rules	638
Edit Privilege Rule Options	639
Removing a Privilege Rule from a Group	644
Configuring Workflow Privileges	645
Workflow state selector	645
Editing Privilege Rules with focus on STEP Workflows	646
Limitations on Workflow privileges	647
Object Creation Privileges	650
Considerations When Creating New Object	651
Mass Creation Screen in Web UI	651
Smartsheets	651
Importer	651
Privileges and Security	651
Granting Access to the SDK and API Documentation	653

Grant Access to the SDK and API documentation	653
GUI Setup	656
GUI Setup on Groups	657
GUI Setup for Users	660
Web UI User Impersonation	662
Workspaces	664
Standard Workspaces	664
Creating Workspaces	666
Deleting Workspaces	669

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.

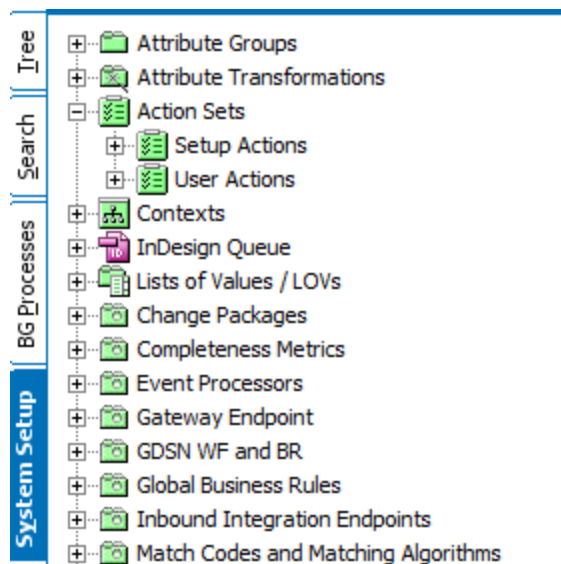
System Setup

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 WebStart 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
Maintain Asset Importer Configuration	Controls if a user is permitted to change an Asset Importer configuration.
Modify dimension dependency of all assets	Set dimension dependency of asset object types: • Add dimension to asset

Action	Privileges Granted
	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.

Action	Privileges Granted
Merge product attributes	Ability to merge attributes
Modify definition of product attribute (domain and default unit)	• Set and edit the validation base type of attributes • Edit the type of attributes (specification and description) • 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
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.

Action	Privileges Granted
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
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

Action	Privileges Granted
	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 Data Container Type	Create a data container type
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 Data Container Type	View a data container type
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
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.

Action	Privileges Granted
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
Maintain STEP Workflow Profile	Edit, create, and delete workflow profiles. The action can be applied to a setup group and will then only take effect for workflow profiles in that setup group. Users must have 'View STEP Workflow Profile' in order to access workflow profiles.
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	• View other users' tasks

Action	Privileges Granted
Administrator	• 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 STEP Workflow Profile	View 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 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

Action	Privileges Granted
	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

User actions for collections allow users to view, delete, and create.

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

User actions for Data Quality Functionality allow users to configure, modify, and generate the available DQ profiles.

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

User actions for eCatalog allow users to create, modify, and view.

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 Data Containers	Approve the modification of data containers
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)'

Action	Privileges Granted
	must be included in order to create classifications.
Create classification reference	Create classification references
Create data container	Adds the 'Add Data Container' action
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 data container	Allows the user to delete a data container
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

Action	Privileges Granted
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 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 data containers (property value) (also translate)	Edit description attribute values on data container attributes
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.

Action	Privileges Granted
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.
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.

Action	Privileges Granted
Move product from	Copy / paste and drag / drop products from another product node within the same object type hierarchy. 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 description attributes linked to classification references and links
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 data containers	Allows the ability to view existing data containers on entity objects
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.

Action	Privileges Granted
View metadata for classification (property value)	View values of description attributes linked to a classification
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 reference (link attribute value)	View link attribute values of attributes that have been applied to the reference type entities. These attributes will appear on all specification attributes that are linked to specified valid entity object type
View metadata for data container (property value)	View description attribute values for data container attributes
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.

Action	Privileges Granted
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

Action	Privileges Granted
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)
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 Flatplanner basket views
Maintain public financial view	Maintain public Financial Summary views (Flatplanner)
Maintain public publication planner view	Maintain public Publication Planner 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

Action	Privileges Granted
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 • 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

Action	Privileges Granted
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.
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

Users Action for Translations allows users to change the translation status, etc.

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

Web UI

Additional User Actions for WebUI component.

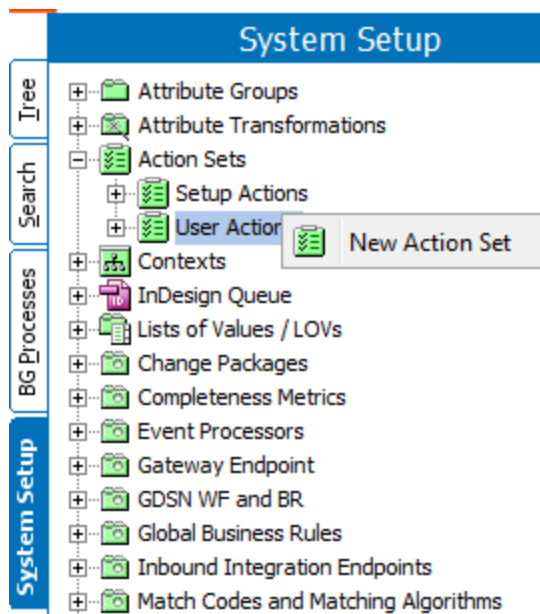
Action	Privileges Granted
Impersonate User	Impersonate a user in a targeted user group in order to troubleshoot issues
Share search with group	Share saved searches with specific user groups
Supplier Web UI functions	Use the Supplier Web UI functions in Supplier Web UI

Maintaining Action Sets

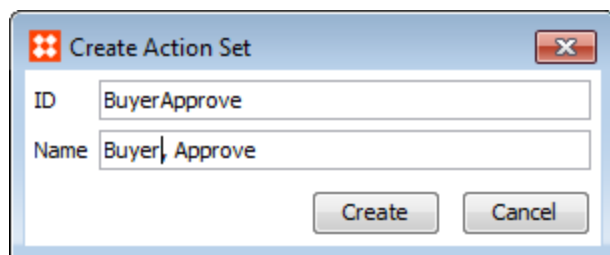
Creating an Action Set

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

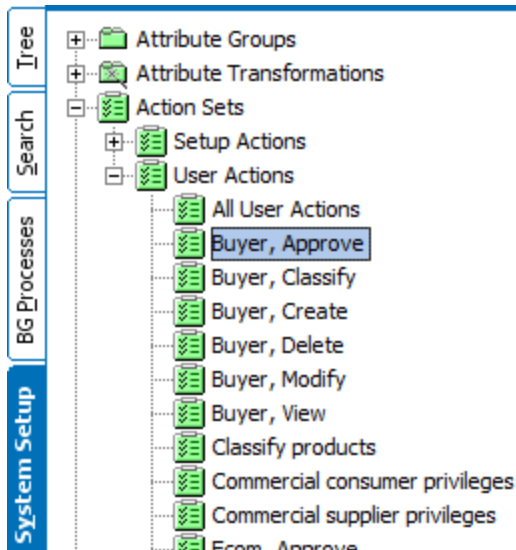
1. In [System Setup], unfold the **Action Set** group and select the relevant main Action Sets folder.
2. Right-click, and then click **New Action Set** (or in the **Maintain** menu, point to **Insert** and then click **Action Set**).



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**.

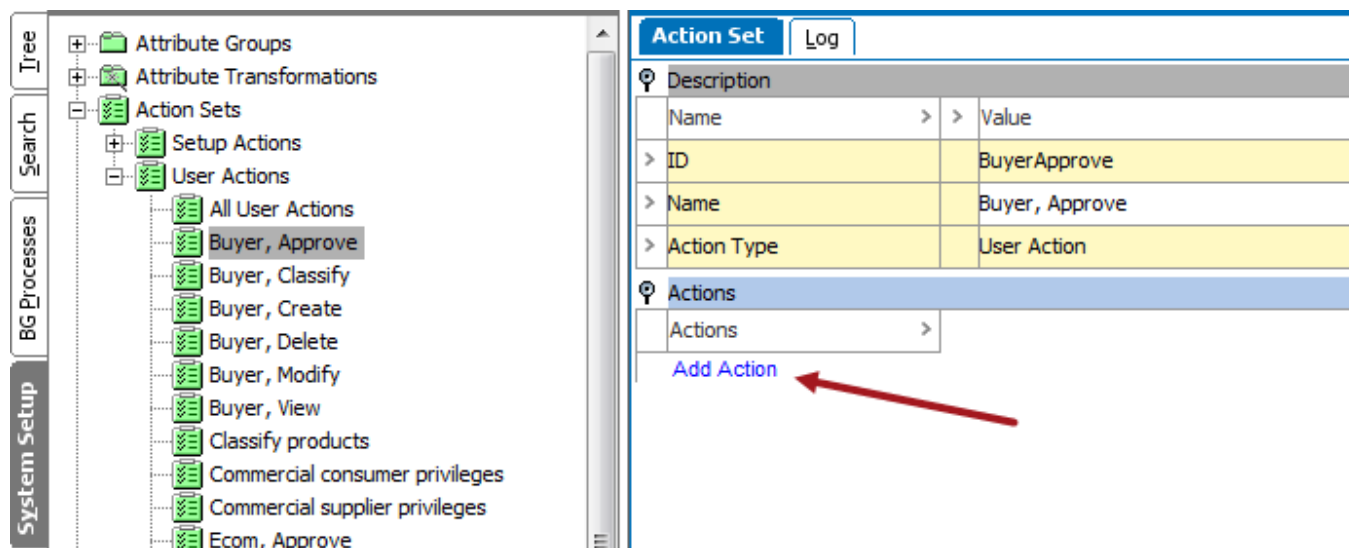


A new Action Set has now been created, and the next step will be to add the specific actions.

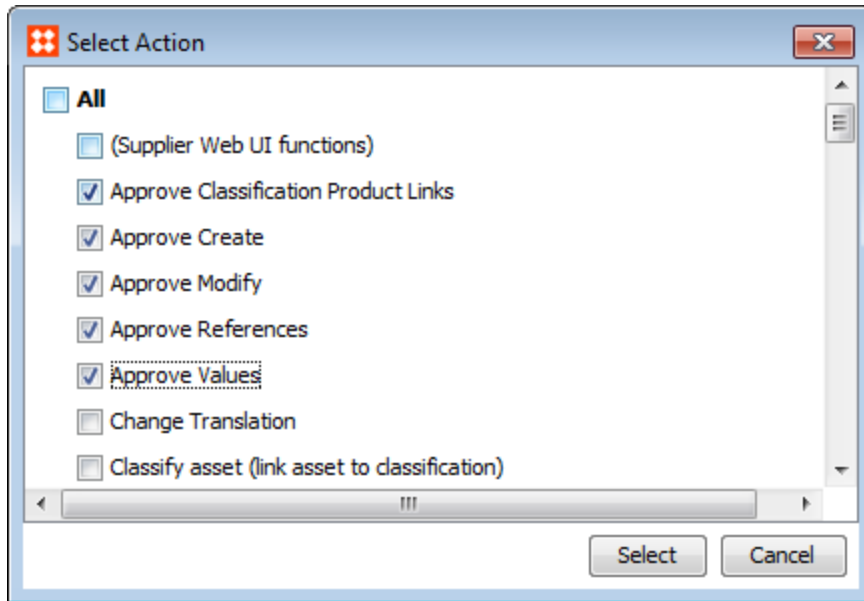
Adding Actions to an Action Set

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.

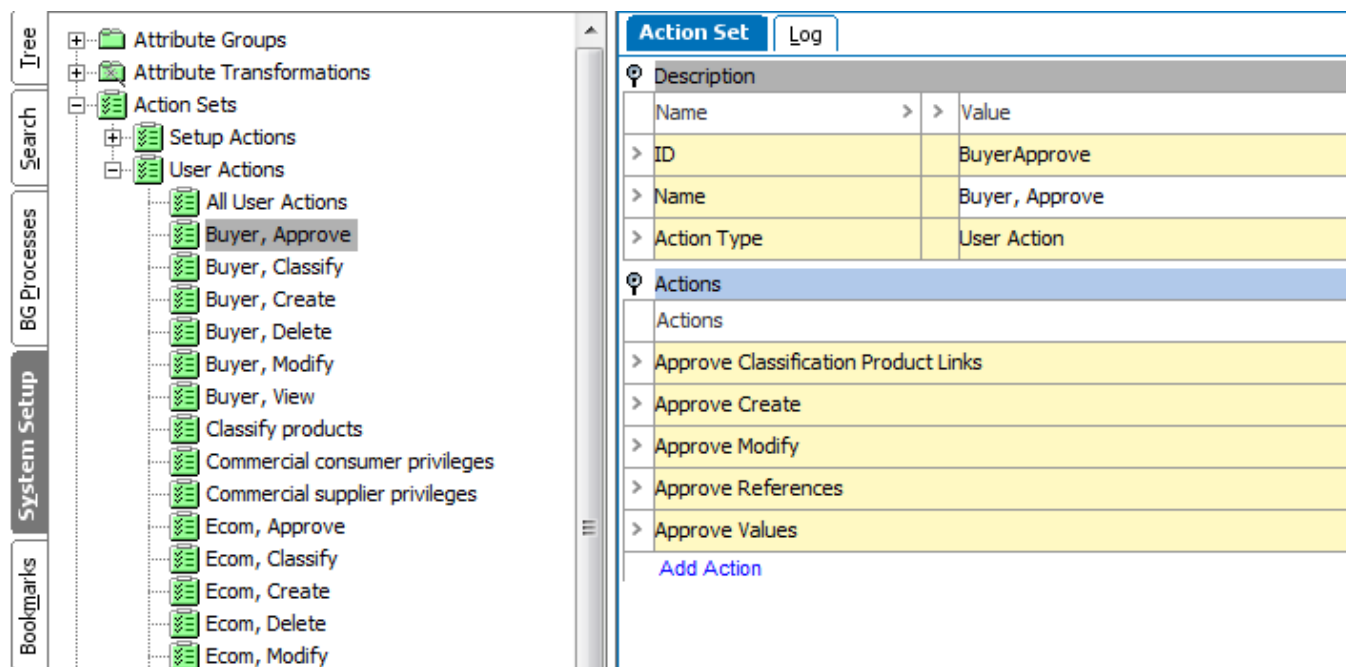


3. Under **Actions**, click **Add Action** and the **Select Action** dialog box appears.



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.

The screenshot shows the Stibo Systems Action Set Editor. On the left, the 'System Setup' tree is expanded, showing 'User Actions' > 'Buyer, Approve' selected. The main area displays the 'Action Set' editor for 'Buyer Approve'. It includes a 'Description' table and an 'Actions' table. A context menu is open over the 'Actions' table, showing options: 'Hide', 'Show All Rows', 'Rotate Table', 'Add Action Ctrl+Plus', and 'Remove Action Ctrl+Minus'. A button at the bottom right says 'Remove the selected rows(s)'.

Description	
Name	Value
ID	BuyerApprove
Name	Buyer, Approve
Action Type	User Action

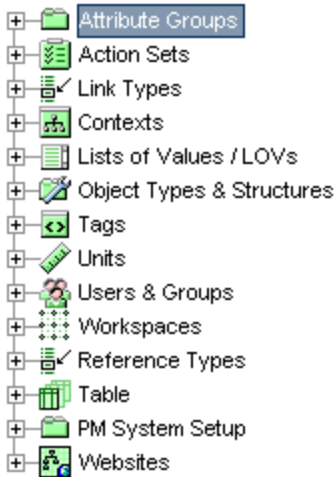
Actions	
Actions	
Approve Classification Product Links	
Approve Create	
Approve Modify	
Approve References	

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.

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, or a relation between objects (Reference / Link).

Objects can be:

- Products
- Image & documents
- Classifications
- Link types
- Data containers

Two types of attributes are available:

- Description attributes
- Specification attributes

Note that Specification attributes can become orphaned. More information can be found in the **Orphan Attributes** topic in the **Super User Guide** documentation.

Attribute Groups

Using attribute groups, you can group attributes, reference types, link types, and data container types according to their characteristics. An attribute group can also consist of a number of attribute subgroups. 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 attribute subgroups instead.

Examples of main attribute groups could be:

- Price group
- Technical group

The following are examples of attribute subgroups:

Attribute Main Group	Included Subgroups	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 subgroups makes it easier to organize and view related attributes. Typically, you will need to link all (or most) of the attributes in a subgroup, which is done by multi-selecting and linking the selected attributes.

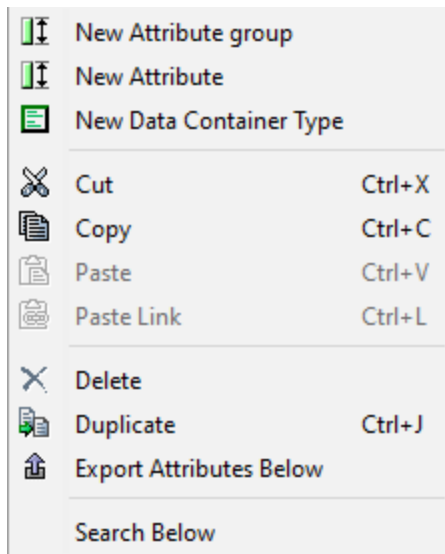
Note: Attribute groups can be specified in privilege rules. For example, specified users can change the values of attributes within the 'PriceGroup' attribute group.

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 attribute subgroups 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.

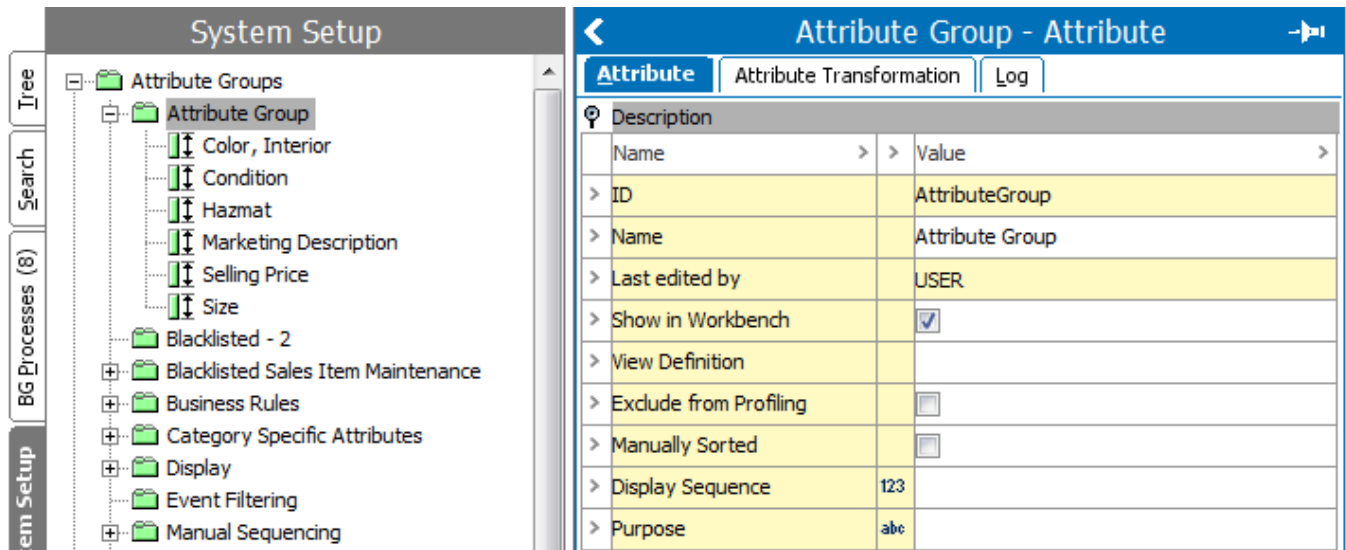
4. Click **Create**. The new attribute group will appear in **System Setup**, below the attribute group where you created it. You can now add attributes, reference types, and product to classification link types to the attribute group.

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.

Important: Approving a reference or product to classification link on a product, classification, or asset having a customized view defined will approve all general aspects and metadata attributes defined on the reference. Additionally, it also approves all general aspects and metadata attributes defined on the reference type or product to classification link type.

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



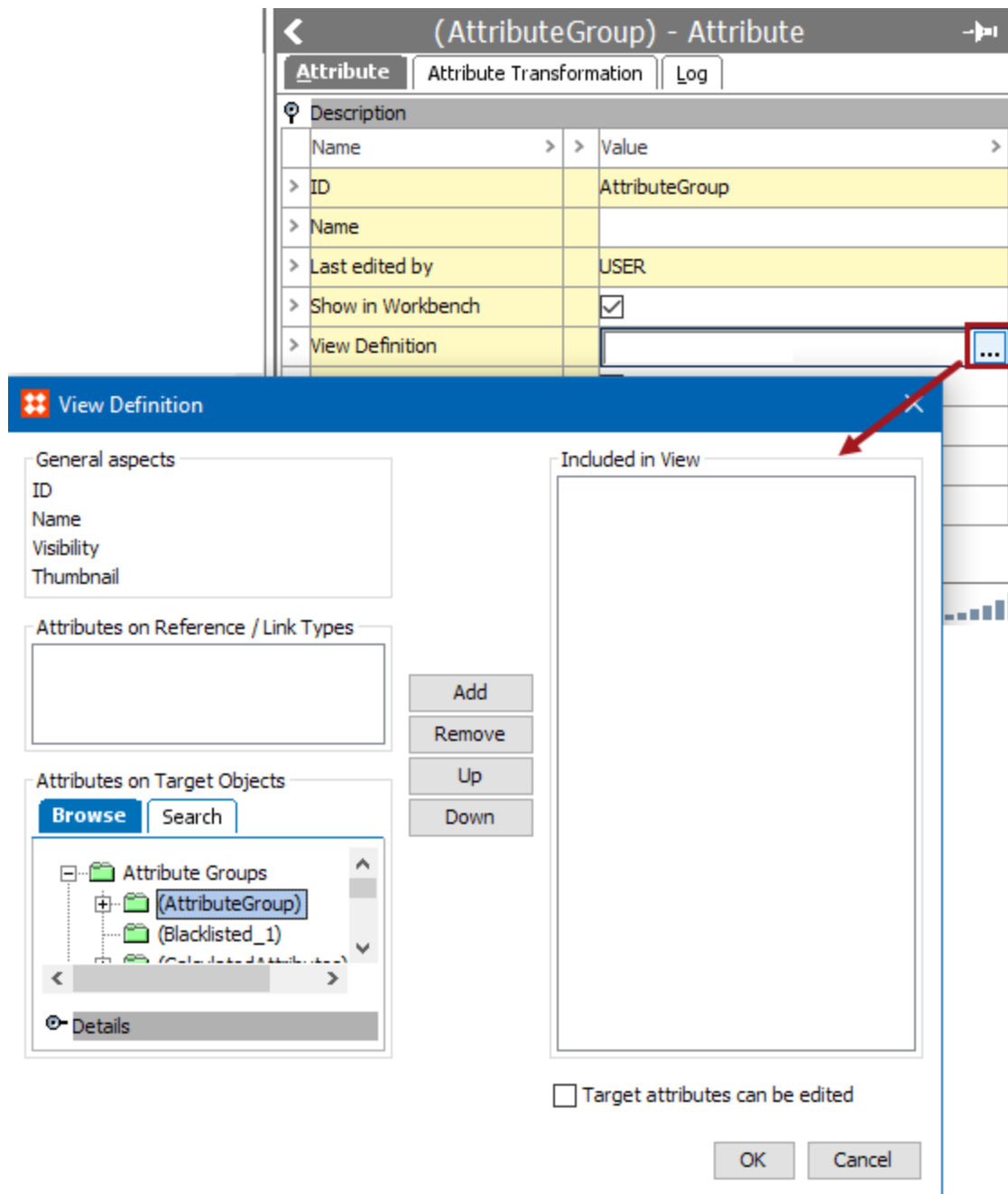
System Setup

Attribute Group - Attribute

Attribute | Attribute Transformation | Log

Description	
Name	Value
ID	AttributeGroup
Name	Attribute Group
Last edited by	USER
Show in Workbench	☒
View Definition	
Exclude from Profiling	☐
Manually Sorted	☐
Display Sequence	123
Purpose	abc

2. Check **Show in Workbench** if all reference types, product to classification link types, and attributes linked to 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 into the field to display the ellipsis button. Then click the ellipsis button (...) 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.

4. In the **View Definition** dialog, a number of choices are available to be included in the view. After selecting an option on the left, click the Add button to move it to the 'Included in View' pane. Use the Add, Remove, Up, and Down buttons to modify the items listed in the 'Included in View' pane.

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. 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. 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—manual sorting of attribute groups, reference types, data containers, 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 Group Display Sequence

As per default, attribute groups 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 attribute groups shown in the Product Editor are sorted according to the order set in the **Display Sequence** attribute.

Note: The **Display Sequence** attribute only has an impact on the attributes shown in the Product Editor. It does not affect the order in which attributes are exported.

Sorting Attribute Groups

The attribute groups shown in the Product Editor can be sorted according to the order set in the **Display Sequence** attribute. This can be done by setting a number in the **Display Sequence** field of the attribute itself.

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. Products 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.

Note: The inheritance feature is supported only in Specification Attributes.

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.

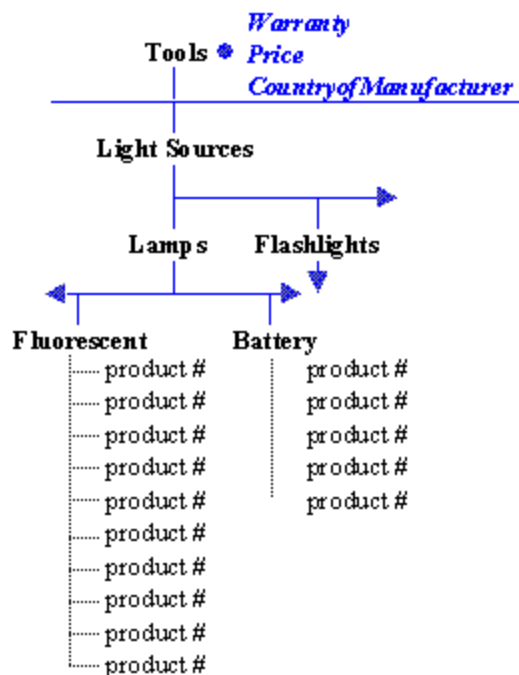


Figure 1 above: A part of a product hierarchy showing the principle of the inheritance of product attributes.

Linking Specification Attributes

Specification attributes are linked to products or classifications either from:

- Attribute editor **References** tab
- Product editor **References** tab: **Linked Attributes** from product hierarchy field
- Classification editor **References** tab: **Attributes** field

Linking Specification Attributes is available for:

- Classifications
- Products

Description attributes are linked globally to an object type. They will then be automatically displayed in the **Description** fields.

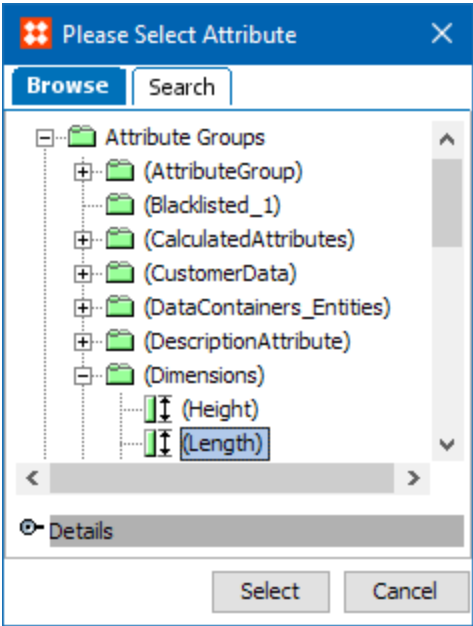
Note: If a specification attribute is not in the list of selectable attributes, it may not be available in the current context.

Linking Attributes in Product editor

1. In Tree, select the relevant node, click the **References** tab.
2. Under the Linked Attributes flipper, click **Link to Attribute**.

Product	Product Variants	Sub Products	References	Referenced By	Images & Documents
1			SupplierName	(SupplierName)	
2			SupplierPartNumber	(SupplierPartNumber)	
			Table Sort Order	(Table Sort Order)	
3			TaxClassification	(TaxClassification)	
			Voltage	(Voltage)	
			Wattage	(Wattage)	
2			ShortItemDescription	Courte Description de l'Article	
.5			LongItemDescription	Long Item Description	
7			ProductName	Nom du produit	

The **Select Attribute** dialog box appears.



3. 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.

Tree

Attribute Groups

Flashlights Items (111204)

Flashlights Items rev.0.7 - References

Commercial | Tables | Category Profile | Proof View | Status | State Log | Tasks

Product | Product Variants | Sub Products | **References** | Referenced By | Images & Documents

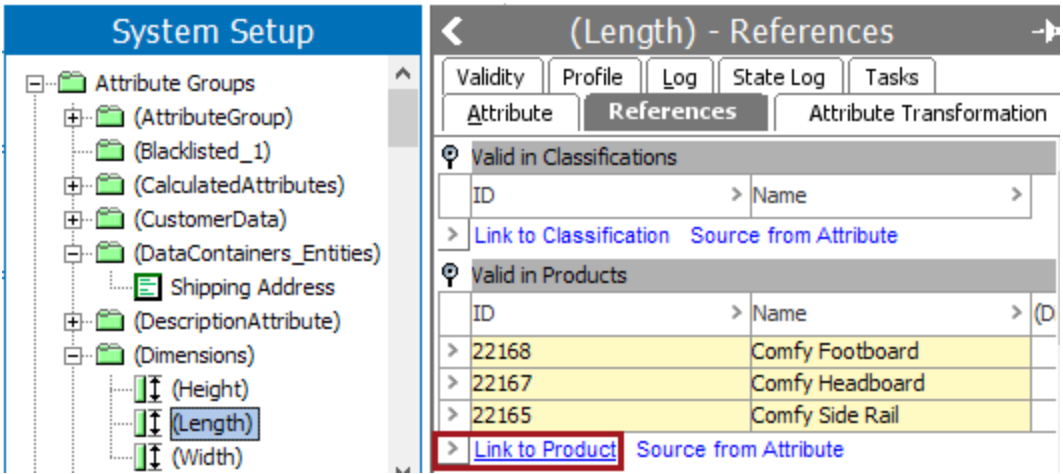
Linked Attributes from Product Hierarchy			
(DisplaySequence)	Completeness Score	ID	Name
2	▼	IsOrderableUnit	(IsOrderableUnit)
4	▼	LeadTime	(LeadTime)
>		Length	(Length)
1	▼	ListPrice	(ListPrice)

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.

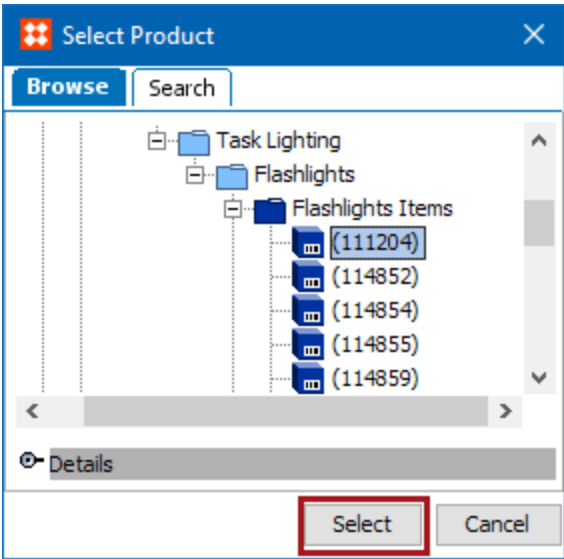
Linking Attributes to Products or Classifications

Linking a specification attribute to a classification or product can help in maintaining the object. For example, linking a specification attribute to a classification node could help in handling different price values of the object for different seasons. The example below shows an attribute being linked to a product, but the steps are the same for linking to a classification.

- 1. In System Setup, select the relevant attribute node, and click the **References** tab.
- 2. Under the 'Valid in Products' flipper, click **Link to Product** to link the object.



3. When the dialog box appears, search or browse for the product you want to link to. Select one or more products from the hierarchy, and then click **Select**.



The selected product(s) will appear in the list of 'Valid in Products.'

System Setup

- Attribute Groups
 - (AttributeGroup)
 - (Blacklisted_1)
 - (CalculatedAttributes)
 - (CustomerData)
 - (DataContainers_Entities)
 - Shipping Address
 - (DescriptionAttribute)
 - (Dimensions)
 - (Height)
 - (Length)
 - (Width)

(Length) - References

Validity Profile Log State Log Tasks

Attribute References Attribute Transformation

Valid in Classifications

ID	Name
> 22168	Comfy Footboard
> 22167	Comfy Headboard
> 22165	Comfy Side Rail
> 111204	(111204)

Valid in Products

Link to Classification Source from Attribute

Link to Product Source from Attribute

Linking to a Source from Attribute

This option is used to copy the classification or product nodes from one attribute and link it to the current attribute. You can link to the 'Source from Attribute' under the 'Valid in Classifications' flipper or the 'Valid in Products' flipper. Because both are linked the same way, an example using a classification will be used.

System Setup

- Attribute Groups
 - (AttributeGroup)
 - (Blacklisted_1)
 - (CalculatedAttributes)
 - (CustomerData)
 - (DataContainers_Entities)
 - Shipping Address
 - (DescriptionAttribute)
 - (Dimensions)
 - (Height)
 - (Length)
 - (Width)

(Length) - References

Validity Profile Log State Log Tasks

Attribute References Attribute Transformation

Valid in Classifications

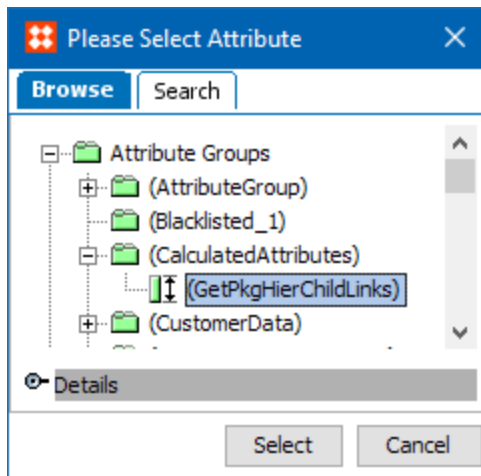
ID	Name
> 22168	Comfy Footboard
> 22167	Comfy Headboard
> 22165	Comfy Side Rail
> 111204	(111204)

Valid in Products

Link to Classification Source from Attribute

Link to Product Source from Attribute

1. To source from attribute, go to the desired attribute in System Setup and click on the **References** tab.
2. Open the 'Valid in classification' flipper or 'Valid in Products' flipper and click on **Source from Attribute**.
3. Select the source attribute you want to copy the classification node link from and link to the current (target) attribute, then click **Select**.



Now the classification node is linked to this attribute.

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.

Below are few of the reasons on how / why the orphan attributes are created:

1. During import an object has an attribute which is not valid to this object type.
2. If the validity of an attribute is removed, but the attribute still holds value(s).

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 attribute is that the orphan already has a value for specific products.

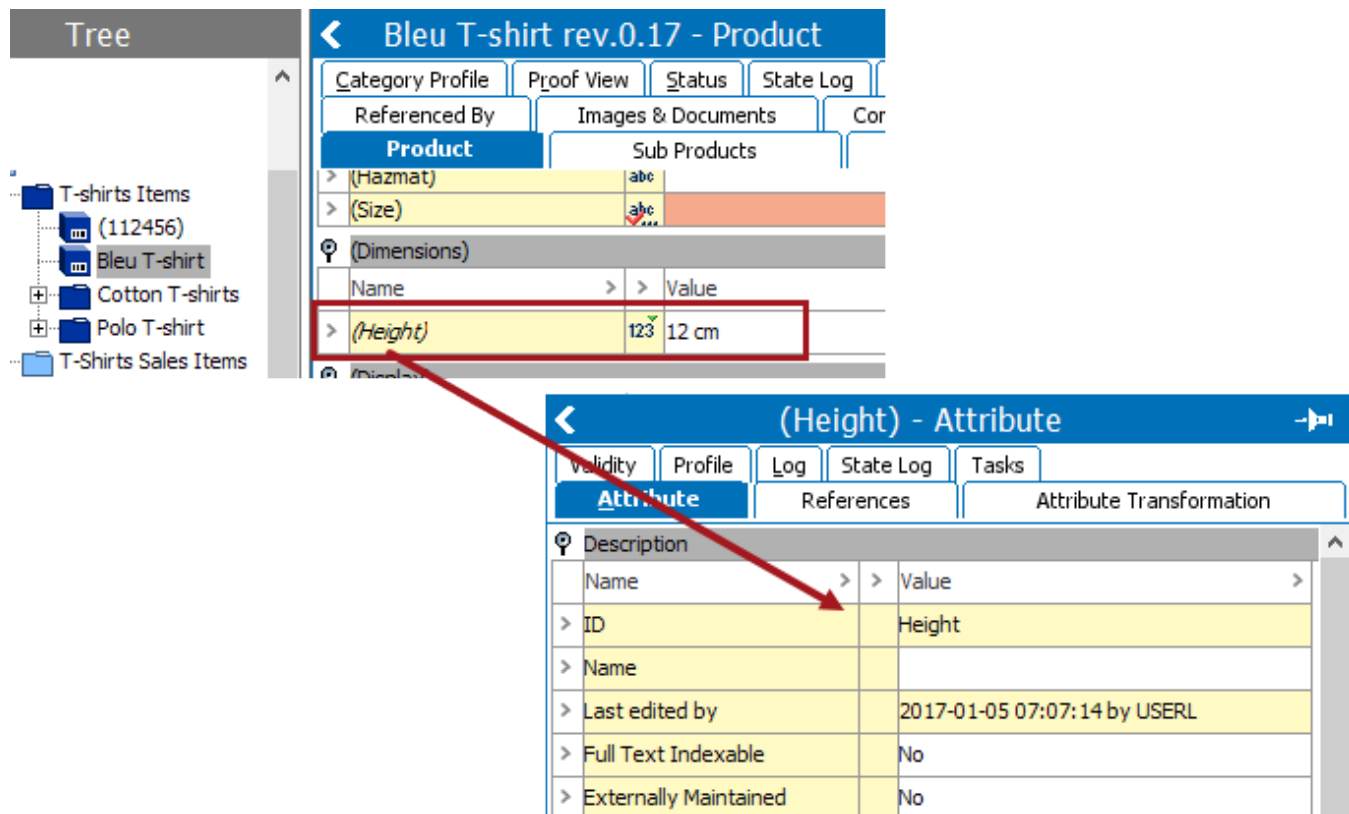
1. In Tree, select the product with the orphan attribute.

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.
3. Click the orphan attribute to be linked. The **Attribute** Editor appears.

Note: All orphan attributes are indicated by being in *italics*.



Tree

- T-shirts Items
 - (112456)
 - Bleu T-shirt
 - Cotton T-shirts
 - Polo T-shirt
 - T-Shirts Sales Items

Bleu T-shirt rev.0.17 - Product

Category Profile | Proof View | Status | State Log

Referenced By | Images & Documents | Cor

Product | Sub Products

> (Hazmat) | abc

> (Size) | abc

⚙ (Dimensions)

Name	Value
> (Height)	123 12 cm

(Height) - Attribute

Validity | Profile | Log | State Log | Tasks

Attribute | References | Attribute Transformation

⚙ Description

Name	Value
> ID	Height
> Name	
> Last edited by	2017-01-05 07:07:14 by USERL
> Full Text Indexable	No
> Externally Maintained	No

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

(Height) - References

Validity Profile Log State Log Tasks

Attribute **References** Attribute Transformation

Valid in Classifications

ID	Name
> Link to Classification	Source from Attribute

Valid in Products

ID	Name	(DisplaySequence)
> 114854	(114854)	
> 20883	20883	
> 22168	Comfy Footboard	
> 22167	Comfy Headboard	
> 22165	Comfy Side Rail	
> 20859	Flashlights Items	
> GDSN_Products	GDSN Products	
> Link to Product	Source from Attribute	

In Attribute Groups

ID	Name
> Dimensions	(Dimensions)
> GDSN Attributes	(GDSN Attributes)
> Add Attribute Group	

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

6. Search for or browse the Classification or Product hierarchy node to link to the needed object, and press **Select**.

Select Product

Browse Search

Primary Product Hierarchy

- Products
 - Apparel
 - Upper Body Wear
 - T-shirts
 - T-shirts Items
 - (112456)
 - Bleu T-shirt**
 - Cotton T-shirts
 - Polo T-shirt
 - T-Shirts Sales Items
 - Head Wear

Details

Select Cancel

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

(Height) - References

ValidityProfileLogState LogTasks

AttributeReferencesAttribute T

Valid in Classifications

IDName

Link to ClassificationSource from Attribute

Valid in Products

IDName

114854(114854)

2088320883

22168Comfy Footboard

22167Comfy Headboard

22165Comfy Side Rail

20859Flashlights Items

GDSN_ProductsGDSN Products

100703Bleu T-shirt

Link to ProductSource from Attribute

Tree

T-shirts

T-shirts Items

(112456)

Bleu T-shirt

Bleu T-shirt rev.0.18 - Product

(Dimensions)

NameValue

(Height)12312 cm

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.

© Stibo Systems - Confidential - STEP Trailblazer 8.1

70

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**.

Note: To determine which attribute on your system is set up as the Display Sequence attribute, go to System Setup > select **Users & Groups** > on System Settings tab open the **Product Information Manager Default Settings** flipper > and verify the attribute under **Default Attribute to use as Attribute Display Sequence Attribute**. Alternatively, on a product editor > References tab > open the **Linked Attributes from Product Hierarchy** flipper > the first column shows the Display Sequence attribute on your system.

Setting a Display Sequence Number on the Attribute

1. In System Setup, open the Attribute Groups node, and then click the relevant attribute. The **Attribute** editor appears.
2. In the display sequence field, type a number. 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.
3. Type the appropriate sequence number for each attribute in an attribute group.

System Setup

- Attribute Groups
 - (AttributeGroup)
 - (Blacklisted_1)
 - (CalculatedAttributes)
 - (GetPkgHierChildLinks)
 - (CustomerData)
 - (DataContainers_Entities)
 - Shipping Address
 - (DescriptionAttribute)
 - (Dimensions)
 - (Height)
 - (Length)
 - (Width)
 - (Display)
 - (eClassMetaAttributes)
 - (ETIM6 Attribute Groups)
 - (ETIM6 Meta Attributes)
 - Display Sequence
 - (ETIM6 Abbreviation)

(Length) - Attribute

Profile	Log	State Log	Tasks
Attribute	References	Attribute Transformation	Validity
ID	Length		
Name			
Last edited by	2017-01-05 07:07:07 by USERL		
Full Text Indexable	No		
Externally Maintained	No		
Completeness Score			
Hierarchical Filtering	None		
Calculated	No		
Type	Specification		
Dimension Dependencies			
Mandatory	No		
Condition Attribute			
Display Sequence	123	1	

Setting a Display Sequence Number under Linked Attributes

1. In Tree, open the product hierarchy folder, 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 that you are only able to edit the display sequence attribute column (white background color), only if the attribute is linked in at the particular node. You cannot edit the display sequence attribute column (yellow background color) if it is linked to the parent node, The **Inherited from** column shows where the attribute is linked in

Tree

- (ETIM Article Groups)
- Assets
 - Primary Product Hierarchy
 - Products
 - Apparel
 - Upper Body Wear
 - T-shirts
 - T-shirts Items
 - (112456)
 - Bleu T-shirt
 - Cotton T-shirt
 - Polo T-shirt

Cotton T-shirts rev.0.10 - References

Images & Documents	Commercial	Tables	Category Profile	Proof View	Status	State Log	Tasks
Product	Product Variants	Sub Products	References	Referenced By			
Linked Attributes from Product Hierarchy							
Display Sequence	Completeness Score	ID	Name	(Latitude)			
		AirTransportationRestricti...	(AirTransportationRestriction...				
		AnnualSalesForecast,Mini...	(AnnualSalesForecast,Minimu...				
		AnnualSalesForecastMaxi...	(AnnualSalesForecastMaximu...				
		AttrSize	(AttrSize)				
		AttrTShirtSize	(AttrTShirtSize)				
		AttrWithLangDepLOV2	(AttrWithLangDepLOV2)				

Note: An attribute can be linked in to one or more Attribute Groups. In the **Attribute Groups** column, you can see 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, and click **Create Local link**. The green arrow symbol will change in the inherited column, and the background color of the row will change from yellow to white. You are now able to type a local value in the display sequence column.

The screenshot shows a table titled "Linked Attributes from Product Hierarchy" with columns: Display Sequence, Completeness Score, and ID. The table contains several rows with attribute names like "AnnualSalesForecastMaxi...", "AttWithLangDepLOV2", "AvailabilityEnd", "AvailabilityStart", "BaseUnitOfMeasure", "Color", "Cost", "CostEffectiveDate", "CostExpirationDate", "CountryOfOrigin", and "DescriptionLong". A right-click context menu is open over the table, showing options: Hide, Show All Columns, Rotate Table, Link to Attribute (Ctrl+Plus), Remove Link to Attribute (Ctrl+Minus), and **Create Local Link** (highlighted with a red box). A red arrow points from the "Create Local Link" option to the "Color" row in the table below.

Display Sequence	Completeness Score	ID
		AnnualSalesForecastMaxi...
		AttWithLangDepLOV2
5		AvailabilityEnd
2		AvailabilityStart
1		BaseUnitOfMeasure
		Color
		Cost
		CostEffectiveDate
		CostExpirationDate
		CountryOfOrigin
		DescriptionLong

Display Sequence	Completeness Score	ID
		AirTransportationRestricti...
		AnnualSalesForecast,Mini...
		AnnualSalesForecastMaxi...
		AttWithLangDepLOV2
5		AvailabilityEnd
2		AvailabilityStart
1		BaseUnitOfMeasure
5		BrandName
6		BrandOwner
		Color

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

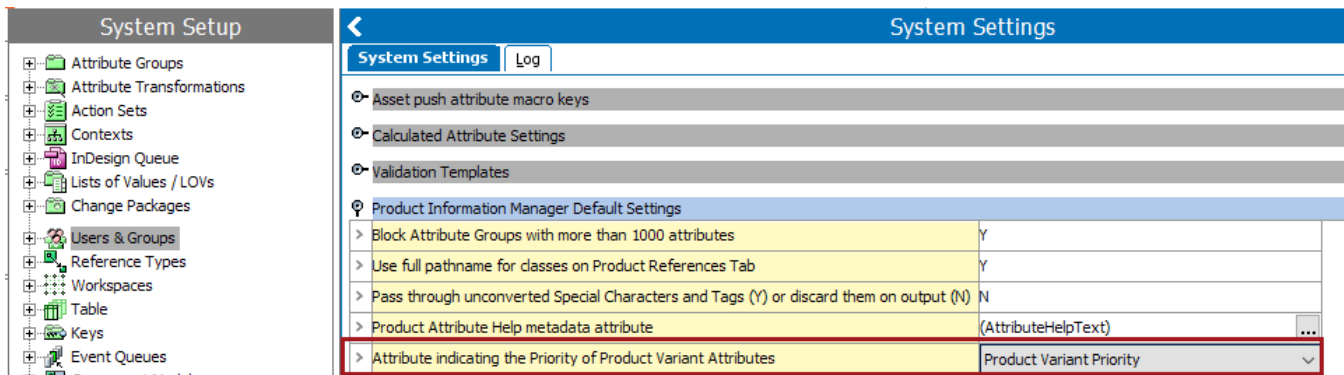
Important: Information about the Display Sequence of attributes is exported in an XML export. If the Display Sequence number is set on the attribute itself, the XML Export will contain the Attributes with Display Sequence attribute value. If the Display Sequence number is set on the Attribute Link, the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

For more information, see the Attribute Sequence topic in the Super User Guide documentation.

Setting the Priority of a Product Variant Attribute

In order to set the priority of a product variant attribute, a user must first make sure that they have this feature selected.

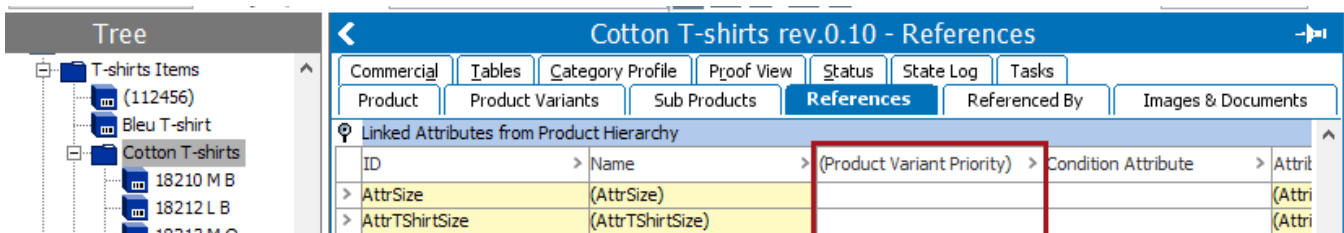
To check if it is selected go to System Setup > Users & Groups node > "Product Information Manager Default Settings" flipper > 'Attribute indicating the Priority of Product Variant Attributes' field. In the field make sure that attribute set up to be the Product Variant Priority is selected from the dropdown menu.



To set the priority of a product variant attribute, follow the steps below.

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'.

Display sequence values can be entered in the column related to the respective attributes.



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.

System Setup

Metadata

Asset Metadata

Catalog Group Theme

Catalog Theme

Circulation

Cover Photo Shot Due

Effective Date

Expiration Date

Get Inherited Pub Group Value

Get Inherited Publication Value

Get Inherited Section Value

Mail Date (same as Effective Date)

Object Name

Section Theme

Total Number of Items

Publication Metadata

Unique Identifiers

Multi Editor

Multi Editor

ID	Name	Completeness Score	Display Name	DisplaySequence
> CatalogGroupTheme	Catalog Group Theme			50
> CatalogTheme	Catalog Theme			50
> Circulation	Circulation			60
> CoverPhotoShotDue	Cover Photo Shot Due			40
> EffectiveDate	Effective Date			20
> ExpirationDate	Expiration Date			30
> GetInheritedPubGroupValue	Get Inherited Pub Group V...			1
> GetInheritedPublicationValue	Get Inherited Publication V...			1
> GetInheritedSectionValue	Get Inherited Section Value			1
> MailDate	Mail Date (same as Effecti...			10
> ObjectName	Object Name			25
> SectionTheme	Section Theme			50
> TotalNumberOfItems	Total Number of Items			15

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.

Transformation Rules	Description
Choose hash function	This will convert the Attribute Values to a cryptographic text using two cryptographic functions MD5 and SHA-1
Convert fractions to parsable tags	This transformation scans a text string for fractions. If a fraction is found, then the fraction will be converted into parsable tags. Example: A text string that contains the fraction 1/2 cm will be transformed to <pre><StartNumerator/><bfdin/> 1</pre> <pre><FractionBar/></pre> <pre><StartDenominator/>2</pre> <pre><EndFraction/></pre> cm Note: It is needed to setup an output formatting to InDesign for <StartNumerator/>, <bfdin/>, <FractionBar/>, <StartDenominator/> and <EndFraction/> in order to mount the fractions.
Extract characters after a given text in the value	This transformation extracts a part of the value, starting after the occurrence of the 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 a part of the value, before the occurrence of the specified text string.
Extract characters between two pieces of text in the value	This transformation extracts the texts between the specified starting text string and the specified ending text string.

Transformation Rules	Description
Extract prefix, suffix and value from an embedded number	This transformation shows the prefix, suffix, and value from an embedded number.
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.
Format the Number with the unit	This transformation makes it possible to format a number with the unit to only have a specified number of decimals.
Generate Soundex code	This will Generate the Soundex code to the Value. Soundex is a general algorithm which creates the same type of 4 alphanumeric codes for the words with similar pronunciation.
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	This transformation will replace specified Characters of a Value.

Transformation Rules	Description
Substrings of the 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.
Replace the whole value using Lookup Table	Replaces the specified value with another value with the Lookup Table. You must select the respective Lookup Table.
Replace words using Lookup Table	Replaces the specified words with another value with the Lookup Table. You must select the respective Lookup Table
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.

Transformation Rules	Description
	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 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.

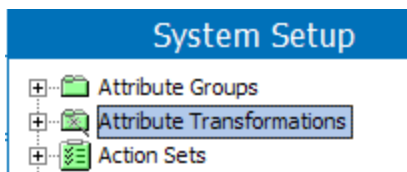
Maintaining Attribute Transformations

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

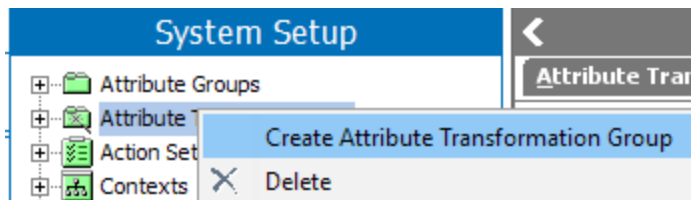
Creating an Attribute Transformation Group

To create an attribute transformation group, follow the steps below:

1. Go to System Setup, click on the **Attribute Transformation**, node.



2. Right-click on the top node, and then click **Create Attribute Transformation Group**.



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

 A screenshot of the 'Create Attribute Transformation Group' dialog box. The dialog has a title bar with the Stibo logo and the text 'Create Attribute Transformation Group' and a close button (X). Inside the dialog, there are two text input fields: 'ID' and 'Name'. Below the input fields are two buttons: 'Create' and 'Cancel'.

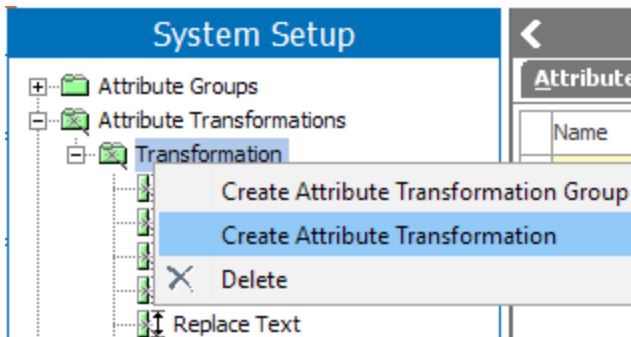
3. Type an ID and name for the attribute transformation group, and then click **Create**.

The attribute transformation group is created, and you can now define an attribute transformation for it.

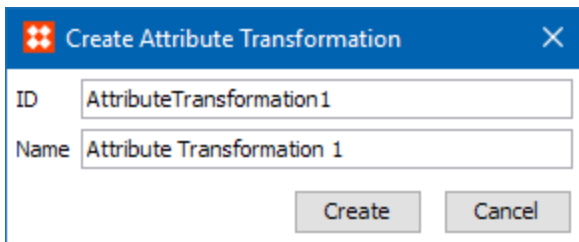
Creating an Attribute Transformation

To create an attribute transformation, follow the steps below.

1. In System Setup, go to the **Attribute Transformation** node, and then select an Attribute Transformation group.
2. Right-click on the group and select **Create Attribute Transformation**.



3. A **Create Attribute Transformation Group** dialog box appears. Type an ID and name for the attribute transformation, and then click **Create**.

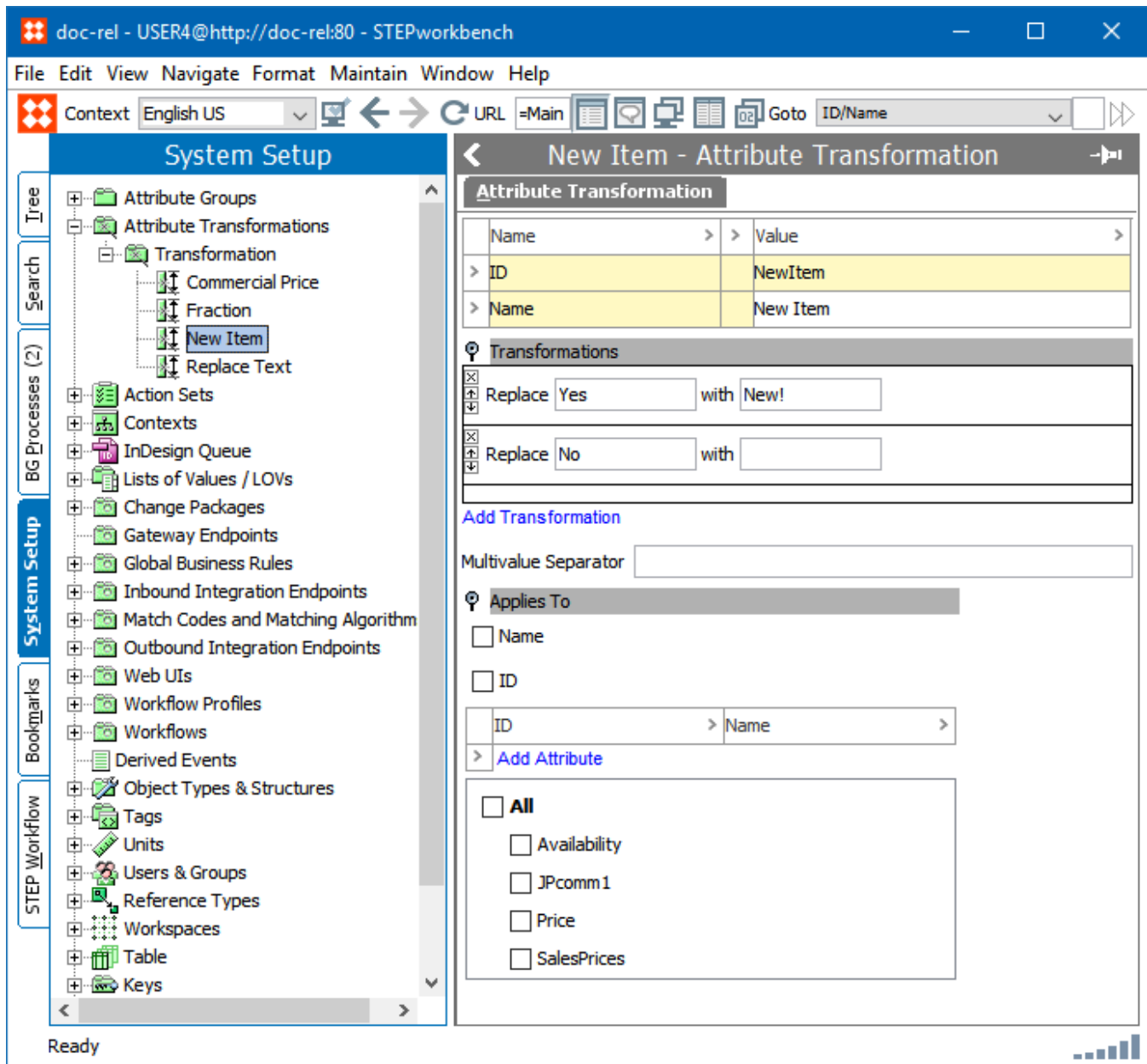


Once the attribute transformation is created, it will need to be set up with a transformation.

Applying a Transformation to an Attribute Transformation

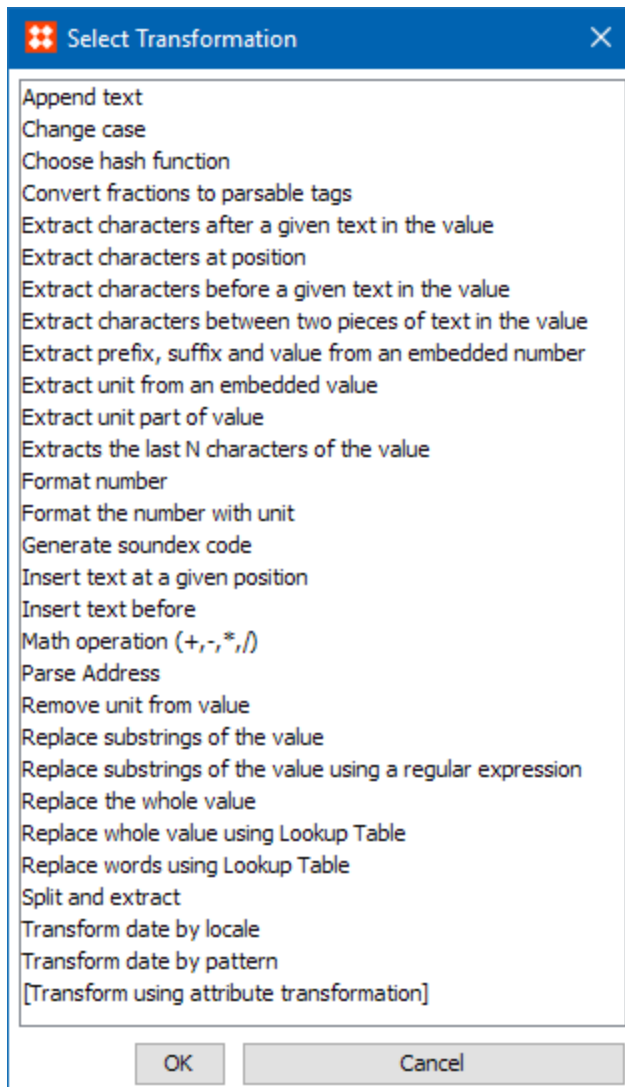
1. In System Setup, unfold the Attribute Transformation node, and then select an Attribute Transformation.

An Attribute Transformation Editor appears.



2. Under the Transformations flipper, 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**.

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 the Transformations flipper, 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.

Attribute Transformation	
Name	> > Value
> ID	NewItem
> Name	New Item
☐ Transformations	
☐ Replace	Yes with New!
☐ Replace	No with
Add Transformation	
Multivalue Separator	
☐ Applies To	

- Under the 'Applies To' flipper, 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.

Attribute Transformation	
Name	> > Value
> ID	NewItem
> Name	New Item
☒ Transformations	
☒ Applies To	
☒ Name	
☐ ID	
ID	> Name >
> IsNew	(IsNew)
Add Attribute	
☐ All	
☐ Availability	
☐ JPcomm1	
☐ Price	
☐ SalesPrices	

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 the 'Applies To' flipper, check the **ID** check box if it should be legal to use the Transformation on object IDs. For example, an object ID could be a product ID.

Name	Value
ID	NewItem
Name	New Item

Transformations

Applies To

☒ Name

☒ ID

ID	Name
IsNew	(IsNew)
Add Attribute	

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 the Applies To flipper, click **Add Attribute**, and select an attribute to legalize the transformation on an attribute. A user can select multiple attributes at one time.

Please Select Attribute

Browse Search

- (AttributeGroup)
- (AttributeN)
- (AttrSize)
- (AttrTShirtSize)
- (AttrWithLangDepLOV2)
- (blue)
- (brown)
- (Color, Interior)

Select Cancel

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 the 'Applies To' flipper, select a Terms List to legalize the transformation on a terms list.

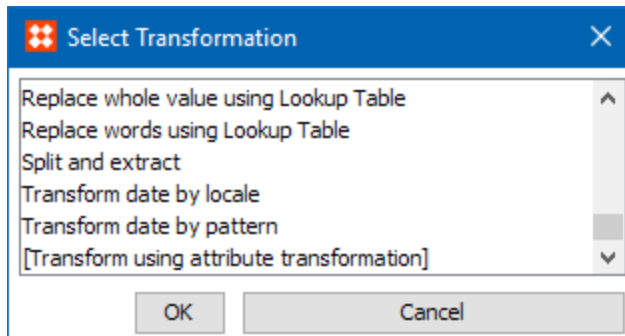
The transformation can be used in:

- 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.

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

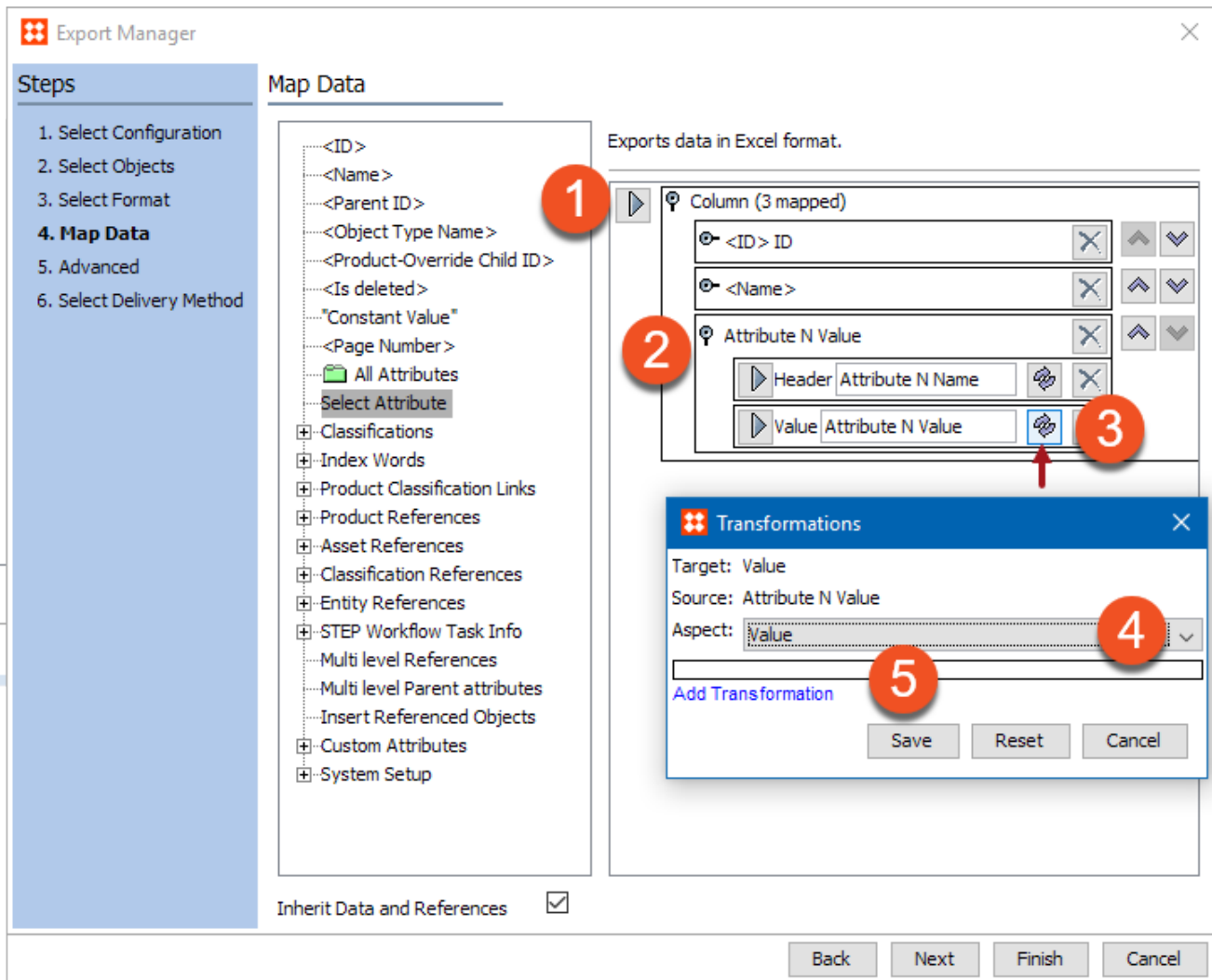
A few examples are shown in the following table. If time zone is included in a pattern used to export data, the time zone abbreviation is determined by the location of the STEP server.

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.

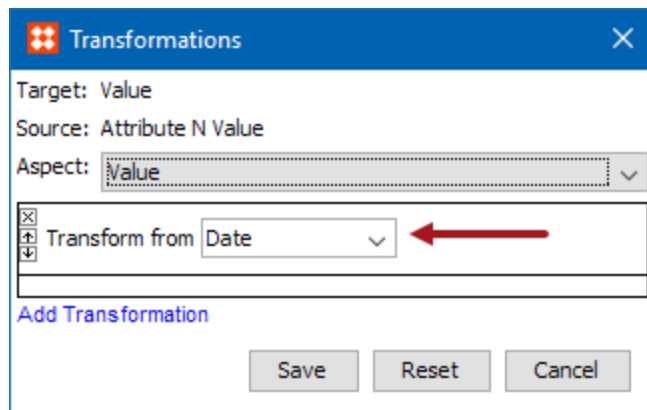
- Verify that the STEP Workbench Context is set correctly. This will impact locale transformations.
- Begin the export process by selecting all the applicable objects on the STEP Workbench Tree.
- Right-click while on a selected object, choose Export Data Below and then follow the export wizard steps. (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 'Attribute N' 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

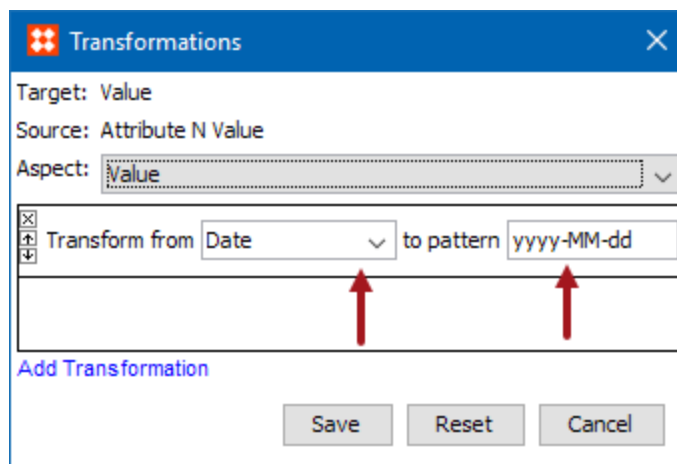
- Click on **Transform date by locale** and then click OK.
- From the Transformations screen, use the 'Transform from' dropdown to choose the validation base type of the

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.

Import Manager

Steps

1. Select Configuration
2. Select Data Source
3. Select Format
- 4. Map Data**
5. Identify Objects
6. Identify Destination
7. Select Business Rules
8. Advanced Settings

Map Data

Source:

<Name>	Manufacturing Expiration Date
EVN-06	15-10-2015

Result:

Map to: Product

Name=<Name> ✓	Date=Manufacturing Expiration Date
EVN-06	15-10-2015

Buttons: Auto Map, Map, Constant, Remove, Transform, Generate Profile

Navigation: Back, Next, Finish, Cancel

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.

Specify the sequence of transformations in the table below.

Target: Date

☐ Mandatory

Value - Source: Manufacturing Expiration Date

Transform to Date

Add Transformation

Preview

Date=Manufacturing Expiration Date ✓

15-Oct-2015

Buttons: Save, Reset, Cancel

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 Creating A Data Import section of the Data Exchange documentation. And, for additional help exporting data, see the Creating a Data Export section of the Data Exchnage 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** section of the **System Setup / Super User Guide**.

The foundation for a calculated attribute is a functional programming language very similar to the language used for defining functions in Excel. Calculated attributes are a special type of attributes whose values are not written in the database, but are calculated on the fly, based on other data in the system.

Both description attributes and specification attributes can be set as calculated attributes, and calculated attribute values can be used in Business Rules as any other attribute values.

When in workbench, calculated attributes are created by going to System Setup and clicking on the desired attribute. Then on the Attribute tab, click into the **Calculated** field, and select 'Yes' or 'No'. In Web UI this can be set on the Attribute Management screen. For information about Web UI, see the **Calculated Attributes in Web UI** section of the **Web User Interfaces** documentation.

When an attribute is set as calculated, a Value template field displays. Functions added to the value template 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).

For example, the following functions and arguments are included in the workbench image 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'.

Note: The functions 'prodval' and 'value' act the same in fetching the attribute values in STEP.

- 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.

System Setup

Attribute Groups

- ATTGroup
 - Attribute Group
 - Attribute A
 - Attribute B
 - Attribute C
 - Attribute Q
 - Attribute Z
 - Biodegradable
 - Calculated Asset File Name
 - Calculated Attribute**
 - Condition
 - Condition2
 - Description Attribute 1
 - Domestic Distribution Number
 - Family-Level Key Attribute
 - Has Lead
 - Hazmat

Calculated Attribute

Attribute | References | Attribute Transformation | Validity | Profile | Log | State Log | Tasks

Description

Name	Value
ID	CalcAttr
Name	Calculated Attribute
Last edited by	2015-11-05 15:29:33 by USER2
Full Text Indexable	No
Externally Maintained	No
Hierarchical Filtering	None
Calculated	Yes
Type	Description
Dimension Dependencies	
Value template	concatenate(prodval("Domestic Distribution ID"),",",prodval("International Distribution ID"))
Mandatory	No

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.

System Setup

Attribute Groups

- ATTGroup
 - Attribute Group
 - Calculated Attrs
 - Attribute A
 - Attribute B
 - Attribute C
 - Attribute Q
 - Attribute Z
 - Biodegradable
 - Calculated Asset File Name
 - Calculated Attribute
 - Calculated Unit**
 - Condition
 - Condition2
 - Description Attribute 1
 - Domestic Distribution Number
 - Family-Level Key Attribute
 - Has Lead

Calculated Unit

Attribute | References | Attribute Transformation | Validity | Profile | Log

Description

Name	Value
ID	CalcUnit
Name	Calculated Unit
Last edited by	2016-05-04 17:34:15 by USER
Full Text Indexable	No
Externally Maintained	No
Hierarchical Filtering	None
Calculated	Yes
Type	Specification
Dimension Dependencies	
Value template	value("Color")
Unit template	unitofvalue("NUMBER1id")
Mandatory	No

Differences Between Calculated Attributes and Standard Attributes

- The base validity that is assigned to a calculated attribute only takes effect when the calculated value is manually overridden. It is up to the formula in the calculation to control syntax of the resolved value, including any numeric min / max constraints, or the maximum length of text. A calculated value may in fact violate the settings in the validity. But if a calculated attribute is given the base validity of Integer with minimum and

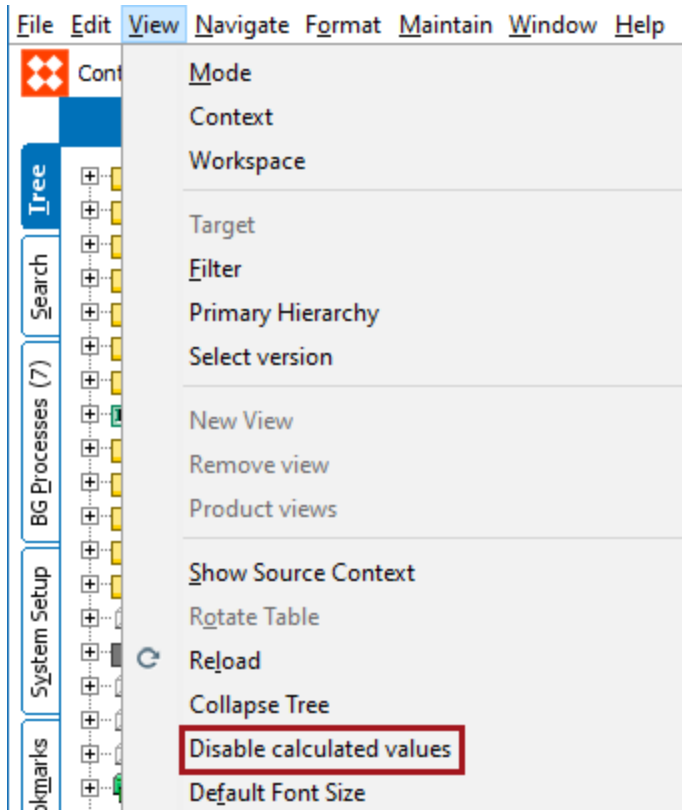
maximum values, then any override to the calculated value must conform to that validity.

- The dimension dependency of an attribute that is designated as Calculated is only relevant when an override has been entered. A calculated attribute may reference attribute values, or reference objects that are dimension dependent, so the result of a calculated attribute is not constrained by the dimension dependency set on the attribute itself, but an override to the calculated value must respect the dimension dependency.
- 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.
- There is a performance penalty whenever the values are to be calculated. Complex calculations affect the time it takes to load editors and export objects.
- When exporting calculated values in STEPXML files using cross-context mode, STEP will attempt to perform some optimization to avoid repetitive calculations per context. In this scenario, the following limitations exist:
 1. None of the attributes used in the calculation can be dimension dependent.
 2. There must be no iterations in the calculation that involve any kind of references or child products. Only attribute values that are directly on the selected object (or, if a product, an inherited value from a higher level) are allowed.
 3. The calculated attribute cannot make use of an attribute value that is itself a calculated attribute.

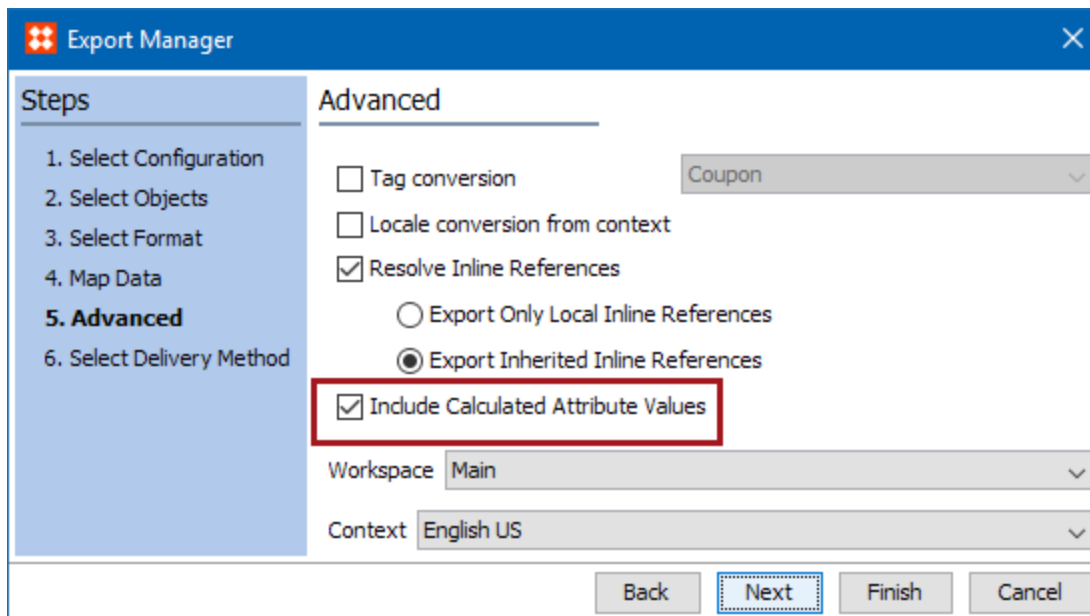
If these criteria are met, then the system will calculate the value once and will export that value for each of the selected contexts. This can result in a reduction of the time to create the exported STEPXML file.

If these criteria are not met, then the calculation will be performed per context, which can increase the time to create the exported STEPXML file.

- Due to potential performance penalties, it is possible to disable the calculated attributes in workbench by going to View > **Disable calculated values**.



- In the Export Manager it is possible to choose whether or not Calculated Attributes should be resolved during export.



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

Note that in order to convert units:

- Relevant units must have a Base Conversion defined in the units area of System Setup
- Calculated attributes should have some numeric validation base type, and then only unit template field will be available.

In the examples below, the `prodval()` function will have passed 'unitid'. This 'unitid' is only needed if a user wants to convert the values with other units to a specific unit. If the unit is already one type / the correct type of unit, there is no need of using `prodval("AttributeID", "Unitid")`. The user can just use `prodval("AttributeID")`.

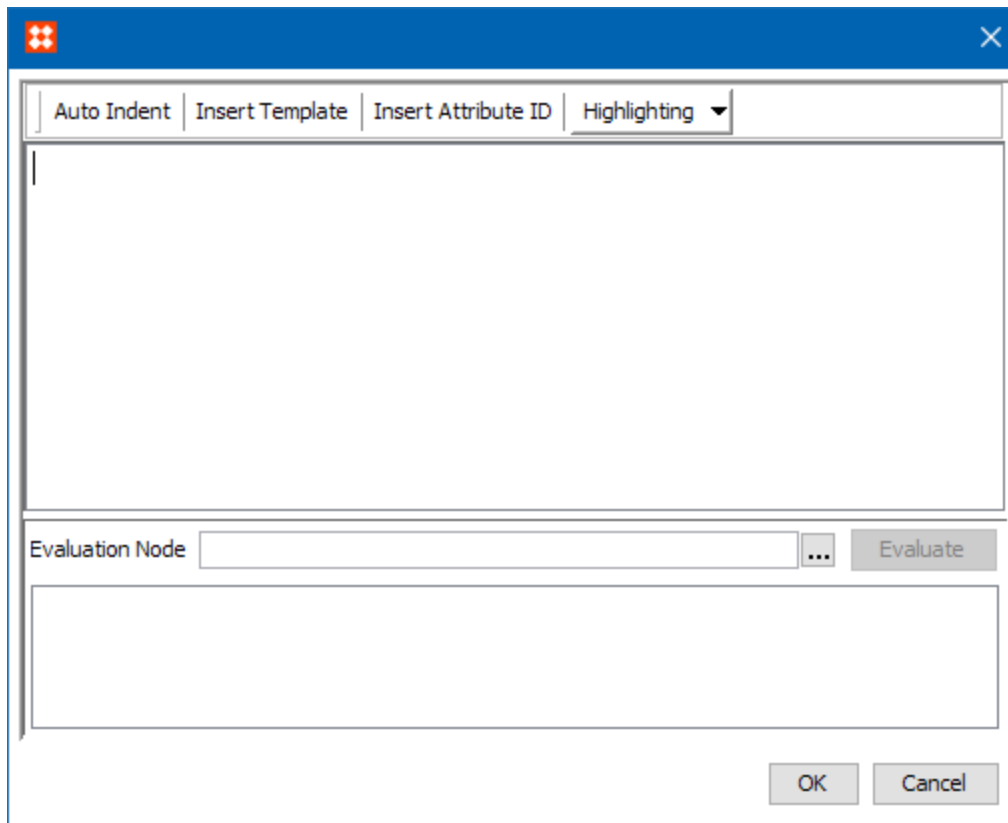
Attribute	Value Template - Expression	Description	Resulting Calculated Attribute Value
Dimensions	<code>CONCATENATE(simplevalue('Height'), ' x ', simplevalue('Width'), ' x ', simplevalue('Length'))</code>	Displays the dimension of the current product in the form 'Height x Width x Length'	22 cm x 39 cm x 54 cm
Volume	Value Template: <code>prodval("Height","unece.unit.CMT") * prodval("Width","unece.unit.CMT") * prodval("Length","unece.unit.CMT")</code> Unit Template: <code>iterate(stepurl2objid(stepurl("Unit","unece.unit.CMQ")), 'stepid()')</code>	Calculates and displays the volume of the current product.	46332 cm ³

Attribute	Value Template - Expression	Description	Resulting Calculated Attribute Value
Volume (Liter)	<code>text(prodval("Height","unece.unit.CMT")</code> <code>*prodval("Width","unece.unit.CMT")</code> <code>*prodval("Length","unece.unit.CMT")/1000,"0.0")</code>	Calculates the volume in Liter and displays with one decimal.	46.3
Form	<code>if(value('Length') = value('Width'), 'Square', 'Not Square')</code>	Displays if the bottom of the box is square or not.	Not Square
Color scheme	<code>if(exact(value('Color Front'), value('Color Back')), 'Same color front and back', 'Different colors front and back')</code>	Compares the front and back colors of the current product and displays the result.	Different colors front and back
Size for standard package	<code>IF (AND (PRODVAL ("Height", "unece.unit.CMT") <= 40, PRODVAL ("Width", "unece.unit.CMT") <= 40, PRODVAL ("Length", "unece.unit.CMT") <= 40), "Size OK for standard package", "Too big for standard package")</code>	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 cabin baggage	<code>IF (AND (OR (AND (PRODVAL ("Height", "unece.unit.CMT") <= 23, PRODVAL ("Length", "unece.unit.CMT") <= 40, PRODVAL ("Width", "unece.unit.CMT") <= 55),</code> <code>AND (PRODVAL ("Height", "unece.unit.CMT") <= 23, PRODVAL ("Length", "unece.unit.CMT") <= 55, PRODVAL ("Width", "unece.unit.CMT") <= 40),</code> <code>AND (PRODVAL ("Height", "unece.unit.CMT") <= 40, PRODVAL ("Length", "unece.unit.CMT") <= 55, PRODVAL ("Width", "unece.unit.CMT") <= 23),</code> <code>AND (PRODVAL ("Height", "unece.unit.CMT") <= 40, PRODVAL ("Length", "unece.unit.CMT") <= 23, PRODVAL ("Width", "unece.unit.CMT") <= 55),</code> <code>AND (PRODVAL ("Height", "unece.unit.CMT") <= 55, PRODVAL ("Length", "unece.unit.CMT") <= 23, PRODVAL ("Width", "unece.unit.CMT") <= 40),</code> <code>AND (PRODVAL ("Height", "unece.unit.CMT") <= 55, PRODVAL ("Length", "unece.unit.CMT") <= 40, PRODVAL ("Width", "unece.unit.CMT") <= 23)),</code> <code>PRODVAL ("Height", "unece.unit.CMT") + PRODVAL ("Length", "unece.unit.CMT") + PRODVAL ("Width", "unece.unit.CMT") <= 115), "Size OK for cabin baggage", "Too Big for cabin baggage")</code>	An air carrier has a limit on the size of the 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.	Size OK for cabin baggage
Size for	<code>IF (AND (OR (PRODVAL ("Length") >40, PRODVAL ("Width") ></code>	A special storage requires that the	OK - One

Attribute	Value Template - Expression	Description	Resulting Calculated Attribute Value
special storage	40), NOT (AND (PRODVAL ("Length") > 40, PRODVAL ("Width") >40))), "OK - One and only one dimension > 40","Not OK - none or two	product has either width or length greater than 40 cm, but not both.	and only one dimension > 40
storage	dimensions > 40")AND (PRODVAL ("Height", "unece.unit.CMT") <= 40, PRODVAL ("Length", "unece.unit.CMT") <= 23, PRODVAL ("Width", "unece.unit.CMT") <= 55), AND (PRODVAL ("Height", "unece.unit.CMT") <= 55, PRODVAL ("Length", "unece.unit.CMT") <= 23, PRODVAL ("Width", "unece.unit.CMT") <= 40), AND (PRODVAL ("Height", "unece.unit.CMT") <= 55, PRODVAL ("Length", "unece.unit.CMT") <= 40, PRODVAL ("Width", "unece.unit.CMT") <= 23)), PRODVAL ("Height", "unece.unit.CMT") + PRODVAL ("Length", "unece.unit.CMT") + PRODVAL ("Width", "unece.unit.CMT") <= 115), "Size OK for cabin baggage", "Too Big for cabin baggage")	the product has either Width or Length greater than 40 cm, but not both.	Dimension > 40

Function Editor

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.

Attribute	References	Attribute Transformation	Validity	Profile	Log	Stat
Description						
Name	>	>	Value			
ID			GetPkgHierChildLinks			
Name			Packaging Hierarchy Child Links			
Last edited by			2016-08-31 13:41:40 by USER6			
Full Text Indexable			No			
Externally Maintained			No			
Hierarchical Filtering			None			
Calculated			Yes			
Type			Specification			
Dimension Dependencies						
Value template			{msg := "Testing attribute for several pkg hier functions"}			
Mandatory			No			

- Double click and right-click the 'Value template' field and choose **Insert Function**.

Note: If there is a pre-existing function, select 'edit Function'.

AttributeReferencesAttribute TransformationValidityProfileLogState LogT

Description

Name	>	>	Value
> Value template			{msg := "Testing attribute for several pko hier functions" i := iterate/
> Mandatory			No
> Attribute Description		abc	
> Attribute Help Text		abc	
> Circulation		123	
> Completeness Score		123	
> Condition			
> Display Name		abc	
> DisplaySequence		123	
> eCl@ss Format		abc	
> eCl@ss ID		abc	
> eCl@ss Name		abc	
> eCl@ss Primary Key		abc	
> ETIM6 Abbreviation		abc	
> ETIM6 Description		abc	
> ETIM6 Feature ID		abc	
> ETIM6 ID		abc	
> ETIM Description		abc	
> ETIM Feature ID		abc	...
> ETIM ID		abc	...
> PrintDisplaySequence		123	

Spelling

UndoCtrl+Z

RedoCtrl+Shift+Z

CutCtrl+X

CopyCtrl+C

PasteCtrl+V

Paste and Match StyleCtrl+Shift+V

Style

Character Tag

Insert Special CharacterCtrl+Shift+E

Insert Function

Edit Function

Insert Footnote

Edit Footnote

Insert Hyperlink

Edit Hyperlink

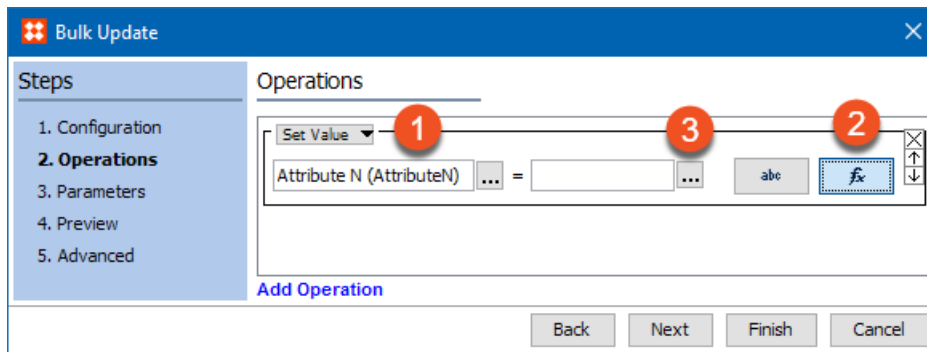
Follow Hyperlink

End Hyperlink

Note: The value template can consist of one or more functions, and it can also contain hardcoded text.

2. Bulk Update wizard

- On step 2 **Operations**, select the 'Set Value' operation, click the Function button () , and then click the ellipsis button (...).

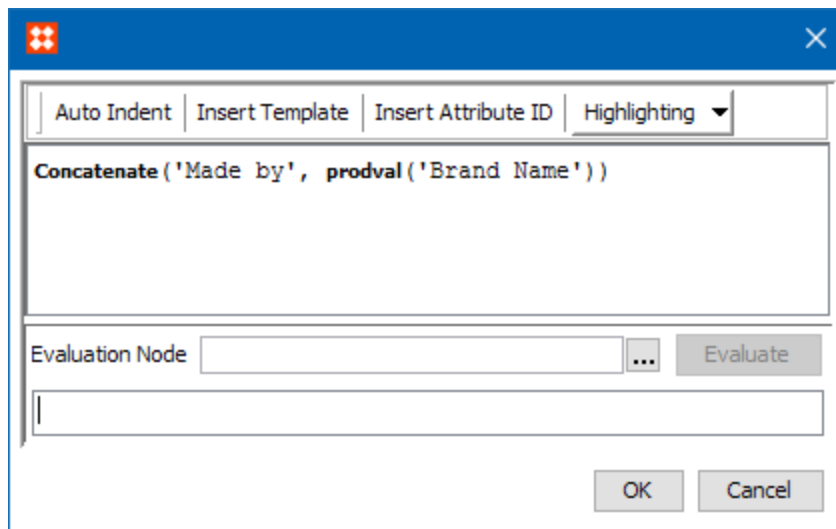


3. Flatplanner

- When working with Flatplanner 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

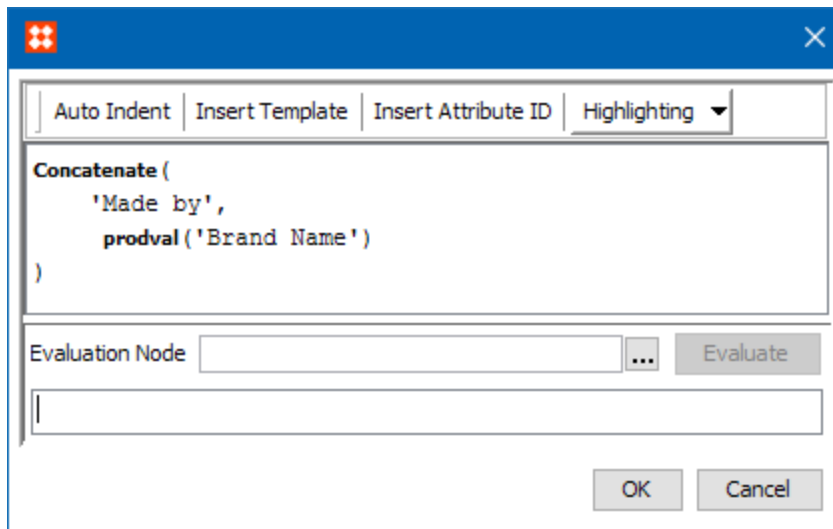
If you know the name and required arguments of the needed function, type directly into the upper text field. For more information on the available functions and their syntax, see the **Calculated Attributes** documentation for **Calculated Attributes Functions** or for **Inserting Templates**.



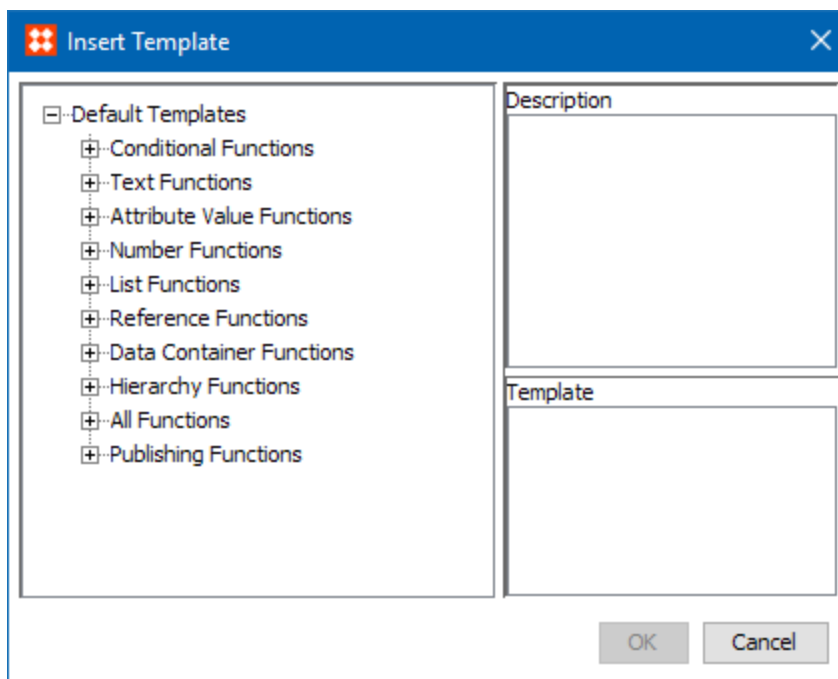
Function Editor Options

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.

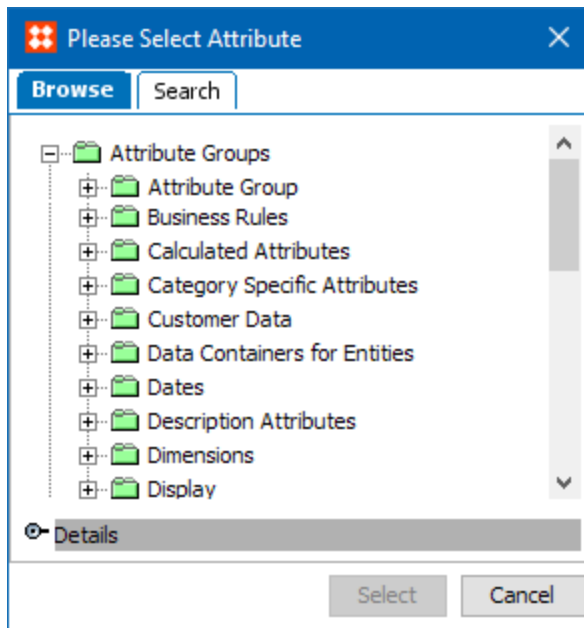
- Auto Indent:** Displays the function in a typical coding layout. Indents and hard returns are added for an easier-to-read format, as shown below.



- **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, which appear when 'Insert Template' is clicked on, but STEP 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
```

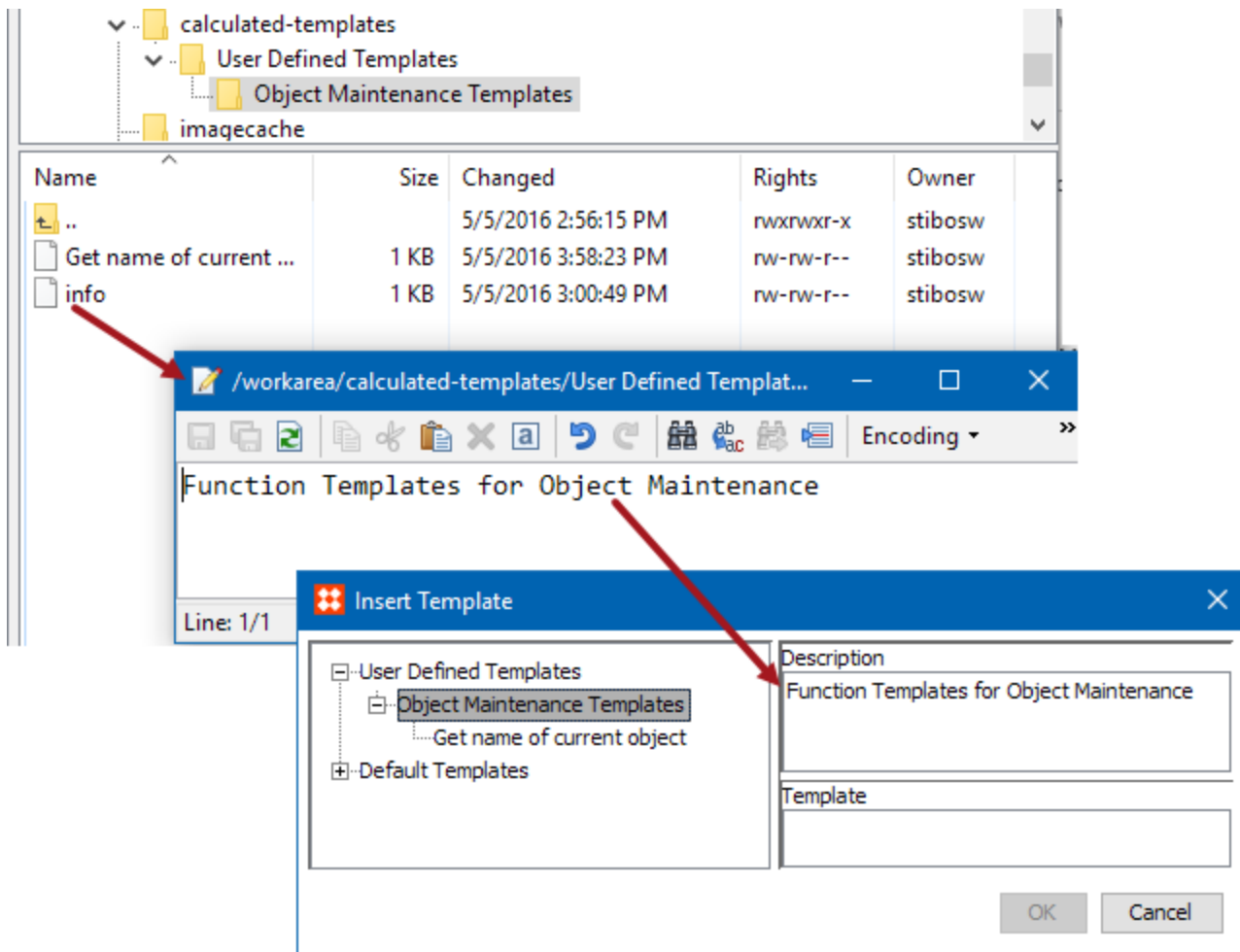
For Windows, the folder is normally found in a drive with the path similar to below:

E:\stiibo\step\calculated-templates

Note: The name of the drive that the path is found in may vary from system to system. In this example the drive 'E' was used.

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.

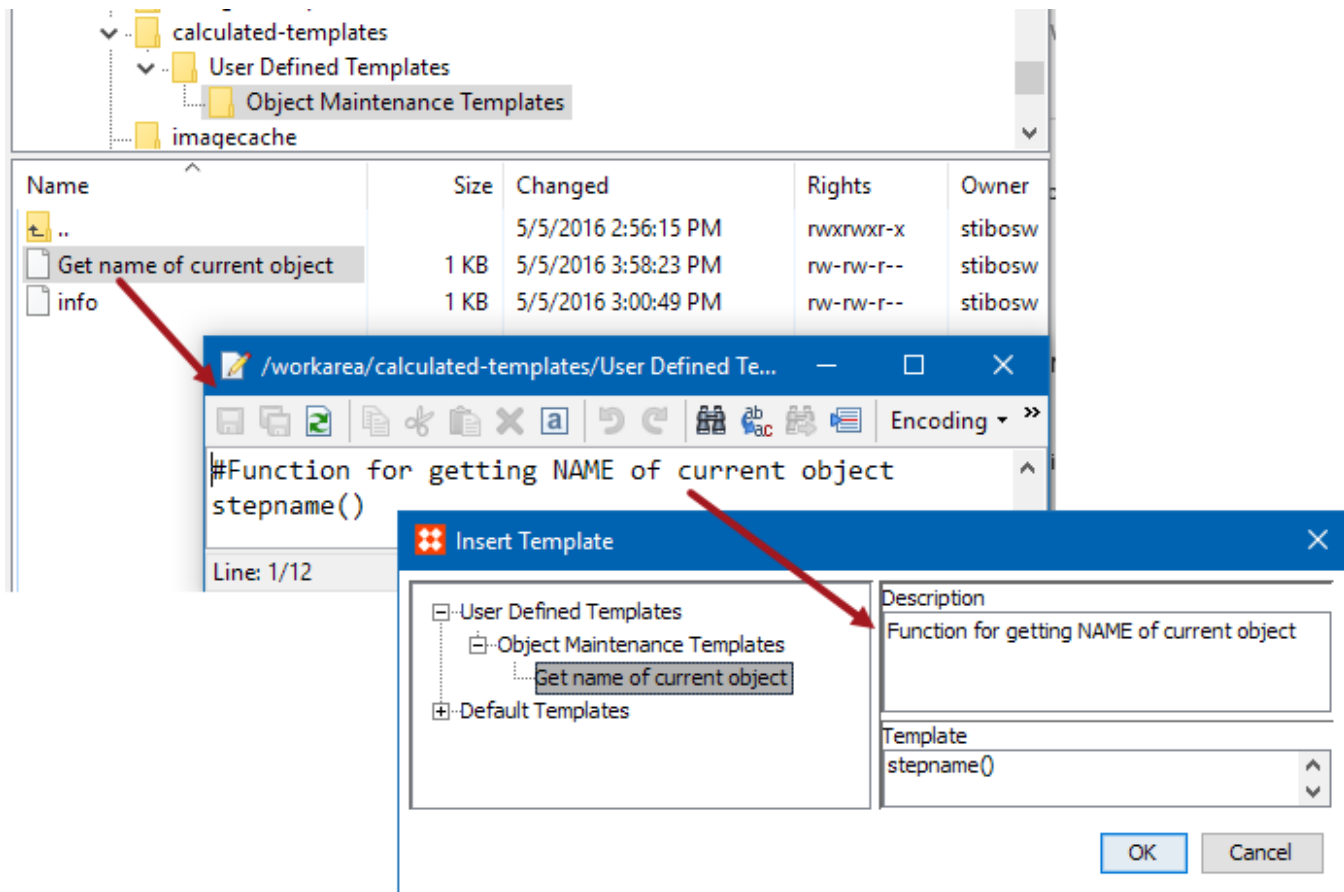
- To add a new level in the Insert Template dialog tree, create a new directory below the root, such as 'Object Maintenance Templates.'
- To display a **Description** for a *directory* in the Insert Template dialog, create a file named 'info' in the directory, and include the description text in it,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

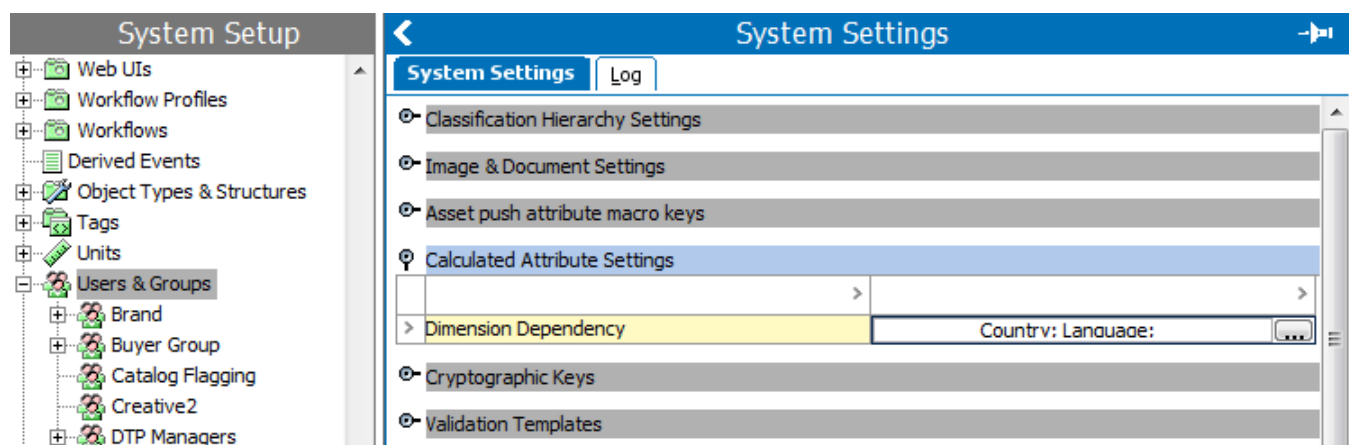
The following image shows the template setup and details in the Insert Template dialog.



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 preferred 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')
```

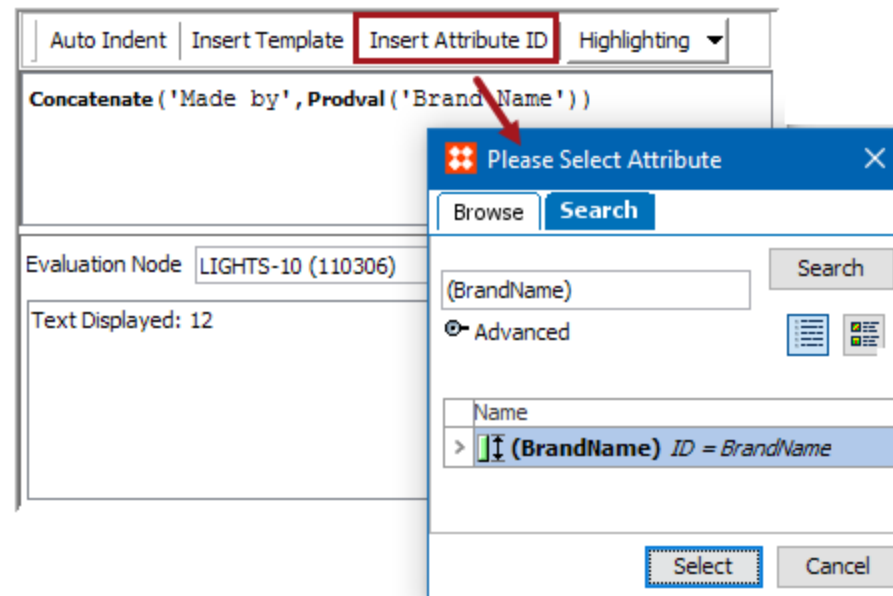
For more information about setting a calculated attribute to be dimension dependent, see the **Calculated Attribute Settings** within the Users and Groups > System Settings section of the **System Setup / Super User Guide** documentation.

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.



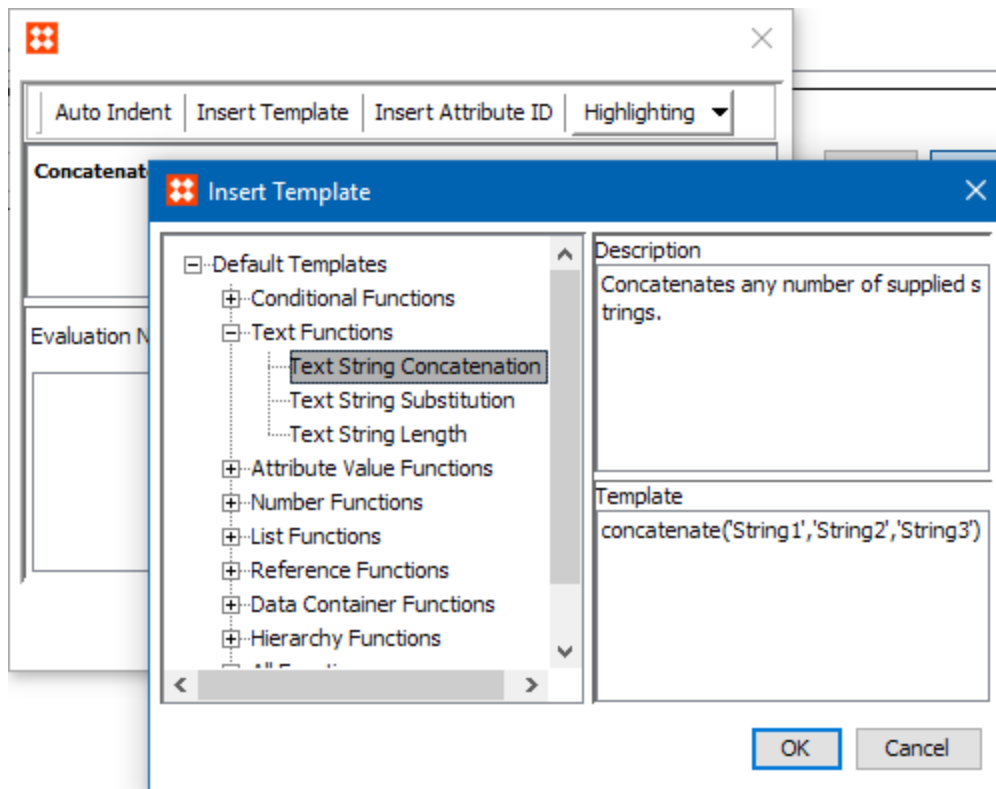
You may also perform some Advances Search operations to find the right attribute, or use wildcards in searches. Please see the **Navigation and Searches** section in the STEP User Guide documentation.

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.

Default Templates listed are arranged in a tree structure which is categorized on their usages on various functions that could be performed in STEP.

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.

In the picture above, the user is adding the Text String Concatenation to concatenate the values of 'height,' 'width,' and 'depth.'

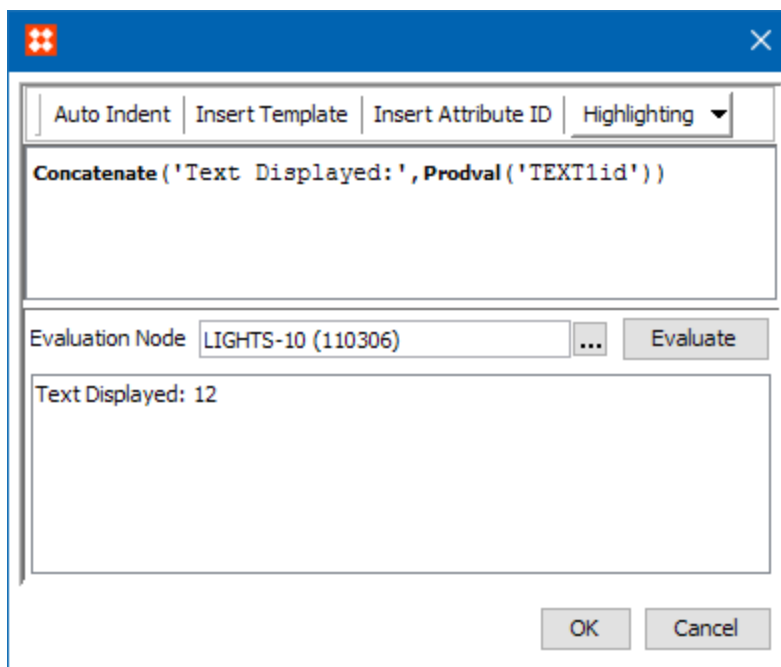
For information on creating user defined templates, see **Creating Function Templates** in the **System Setup / Super User Guide**.

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: 12'

To evaluate only a specific part of the function, highlight it and click the Evaluate button. Always ensure that the Parentheses are opened and closed for the selection. If there is any error in the code, or if the parentheses are not handled correctly, than the Evaluate result will be 'N/A'.



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

Calculated Attribute Functions

Calculated attributes can be set to Description or Specification, which depends on the requirement.

- 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 click on Override.

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' and 'unit-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
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
P3id	P3	P-family-id	p3 value	6 m

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
getembededsuffix(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', [effective date])	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', [effective date])	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', [effective date])	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', [effective date])	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', [effective date])	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', [effective date])	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

Terms List Object Type ID = PriceID			Terms List ID = TermsListID			
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
'iso4217.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('TermsListID', '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.

Note: The OR(), AND(), and NOT() functions can be used with IF function to check multiple conditions at a time. For more information, these are explained in the **Other Functions** documentation.

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 documentation for SimpleDateFormat for Oracle on the web. [locale] is optional; see Using a Locale below
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; see Using a Locale below
NOW	()	Returns current time on server in yyyy-MM-dd HH:mm:ss format.

Note: The above functions work only when the date values are entered in yyyy-MM-dd or yyyy-MM-dd HH:mm:ss formats. 'ISO Date' and 'ISO Date and Time' validation base type attributes accept only these formats.

Using a Locale

To obtain a date or date / time format for a specific locale, in order to override the locale that is specified by the current context, enter the locale's ID and ensure the following property is defined in the sharedconfig.properties file: Calculated.UseLegacyLocalization=true

For more information on locales, see **Context Locales** documentation in this guide.

Examples

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

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

The StartDate attribute is of the 'ISO Date' validation base type, and has the value: 2017-03-13. The following calculated attribute function returns the localized values displayed below for Great Britain, United States, and Germany.

```
{d := value('StartDate'), x := localizedate(d, "en_GB"), y := localizedate(d, "en_US"),
z := localizedate(d, "de_DE")} concatenate(d, " ", x, " ", y, " ", z)
// 2017-03-13   13-Mar-2017   Mar 13, 2017   13.03.2017
```

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).
INDEX		This variable hold 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.

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
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
P3id	P3	P-family-id	p3 value	6 m

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

list(reverse(iterate(subproducts(), 'prodval("TEXT1id")'), ', ' ) // = "p3 value, p2 value,
p1 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"

list(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

list(iterate(subproducts(), 'concatenate(index, ": ", item)'), ', \n') // =
```

1. com.stibo.core.persistence.NodePO-343239
2. com.stibo.core.persistence.NodePO-3432395
3. com.stibo.core.persistence.NodePO-3432395

evaluated on family “P-family-id”. Here “com.stibo.core.persistence.NodePO-3432391,...” are internal representation of the nodes (sub products). We have iterated the sub products and printed the iterated products as it is using the default variable “item”. So the internal representation of the nodes is displayed. If we need some attribute values or some other data from these iterated nodes, we should call it inside the iterate function. Example below:

```
list(iterate(subproducts(), 'concatenate(index, ": ", stepid())'), '\n') //=  
P1id  
P2id  
P3id
```

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; see Using a Locale below
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.

Using a Locale

When using calculations on numbers, use the `LOCALIZENUMBER()` and specify a locale ID that uses a 'dot-is-decimal' method (for example, 12.34) and not a 'comma-is-decimal' method (for example, 12,34). After the calculation, use the function again to convert the decimal point value to the required locale's format. Ensure the following property is defined in the `sharedconfig.properties` file: `Calculated.UseLegacyLocalization=true`

For more information on locales, see **Context Locales** documentation in this guide.

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
(value('AttID1'))*(value('AttID2')) // = the result of (attribute 1's value) multiplied by
(attribute 2's value)
```

In the example below, the Diameter attribute is in a context where a locale with a 'comma-is-decimal' method is used and has the value: 12,7

Evaluating the following value template shows that if you attempt to make a math calculation on a number that is using the comma-is-decimal method, the calculation does not work, as indicated by the NaN ('not a number') result. Instead, perform calculations in a 'dot-is-decimal' locale (`en_US` or `en_GB`, for example) and then convert back to the preferred locale after the calculation has been done. In this example, this is done by the statement `'num := localizenumber(p),'` which reverts the calculated number to the format according to the locale that is specified in the currently selected context.

```
{x := value('Diameter'), y := 3.414,
num_us := localizenumber(x, "en_US"), num_de := localizenumber(x, "de_DE"),
p := num_us*y, q := num_de*y,
num := localizenumber(p) }
concatenate(x, " ", p, " ", q, " ", num)
//12,7      43.3578      NaN      43,3578
```

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 / Super User Guide** documentation.

Also refer to the Insert Templates topic in the **System Setup / Super User Guide** documentation for more information.

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

Other Function	Parameters	Description
		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'.
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.

Other Function	Parameters	Description
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.
DATACONTAINERS	([data-container-type])	Return data container objects of current composite object. Return all data containers of data container type with ID 'data-container-type' (all if not specified).
DATACONTAINERCOMPOSITE	()	Return composite object of current data container object.
DATACONTAINERTYPEID	()	Return ID of data container type of current data container
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

Other Function	Parameters	Description
		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 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.
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).

Other Function	Parameters	Description
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.
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

Other Function	Parameters	Description
		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 Creating a Transformation Lookup Table 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

Other Function	Parameters	Description
		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'.

Use Cases

For an example using overrides, see **Use Case: Calculated Attribute for Overrides**.

For an example using classifications, see **Use Case: Calculated Attribute for Classifications**.

For an example using classifications, see **Use Case: Calculated Attribute for References**.

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
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
P3id	P3	P-family-id	p3 value	6 m

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 Plid
or((value('TEXT1id')= 'p0 value'), (value('NUMBER1id')= '3')) // = 0 (FALSE) evaluated Plid
and((value('TEXT1id')= 'p1 value'), (value('NUMBER1id')= '4')) // = 1 (TRUE) evaluated Plid
and((value('TEXT1id')= 'p0 value'), (value('NUMBER1id')= '3')) // = 0 (FALSE) evaluated Plid
not(value('TEXT1id')= 'p1 value')) // = 0 (FALSE) evaluated Plid
not(value('TEXT1id')= 'p0 value')) // = 1 (TRUE) evaluated Plid
stepid() // = 'Plid' evaluated Plid
stepname() // = 'P1' evaluated Plid
stepname('step://product?id=Plid') // = 'P1' evaluated Plid
stepurl() // = 'step://product?id=Plid' evaluated Plid
stepurl('Product','Plid') // = 'step://product?id=Plid' evaluated Plid
stepobjecttype() // = 'Item' evaluated Plid
stepobjecttype('step://product?id=Plid') // = 'Item' evaluated Plid
iterate(stepurl2objid(stepurl('Product','Level1-109267')),'stepname()')
// = 'Plid' evaluated on some product node.

list(iterate(children(),'concatenate(stepname())'),' ',' ') // = 'P1, P2, P3' evaluated P-
Family since these are products
list(iterate(subproducts(),'concatenate(stepname())'),' ',' ') // = 'P1, P2, P3' evaluated P-
Family 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
three children, none of which are products
list(iterate(iterate(references('product', 'BillOfMaterials'),'referencetarget
()'),'stepname()'),' ',' ') // = 'P3' evaluated against P1 when P1 has a BillOfMaterials
reference to P3
parentid() // = 'P-family' evaluated Plid
list(iterate(path(),'concatenate(stepname())'),' ',' ') // = 'Primary Product Hierarchy,
Products, P-family' evaluated P-Family

```

```
isbelow('step://product?id=Plid') // = 0 (FALSE) evaluated P-Family
isbelow('step://product?id=P-Family') // = 1 (TRUE) evaluated Plid
list(iterate(revisions(), 'revisioncreatedate()'), ', ') // = '2016-04-20 17:18:42, 2016-04-22 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() // = '51.282051282051284601948282215744' evaluated P1
round(completeness(),0) // = '51' evaluated P1
workflowsids() // = 'ItemCreationOrderProcessWorkflow' evaluated P1 (2 workflows returned: Item Creation and Order Process Workflow)
currentworkflowstates('OrderProcessWorkflow') // = 'State-2' evaluated P1
list(iterate(classifications(), "stepid()"), "/" ) // = Classification1/ Classification2 evaluated 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 Description")'),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 value: 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
<code>overriddenproduct('BatteriesOverride1ID')</code>	Evaluated on Batteries Override 1: Batteries
<code>overriddenproduct('Batteries')</code>	Evaluated on Batteries Override 1: N/A
<pre>{ x := listlen(overriddenby()), z := iterate(overriddenby(), 'stepname()'), z := list2multivalue(z), z := substitute(z, "<multisep/>", " ") } concatenate("This product is overridden by ", x,</pre>	Evaluated on Batteries: This product is overridden by 3 product override(s): Batteries Override 2 Batteries Override 1 Batteries Override 3

Template	Results
<pre>" , product override(s): " , z)</pre>	
<pre>{ x := iterate(overriddenproductobject(), 'stepname ()'), z := listitem(x,1) } concatenate('This PO overrides the product "', z, '")</pre>	Evaluated on Batteries Override 1: This PO overrides the product "Batteries"
<pre>list(iterate(productoverrideparents(), "stepid ()"), "/")</pre>	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, "-", "")
<lt/> 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'.
REGEXP SUBSTITUTE	(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 ESAPESPLIT.
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('llo','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") //= "Joe 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

```

Note: list(listconcatenate(prodval("height"),prodval("width"),prodval("length")),'*') // width value is empty, and this needs to be concatenated without an extra separator '*', for that listconcatenate and list functions are used. IF function can be used to check whether a value is empty or not before concatenating, but it needs lot of code to check for each concatenating value. The above method is easy to write in this scenario.

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
Text-validated product attribute	TEXT1id	TEXT1

PRODUCTS ID	Name	Parent	TEXT1id
P-family-id	P-family	Product hierarchy root	family value
P1id	P1	P-family-id	p1 value
P2id	P2	P-family-id	p2 value
P3id	P3	P-family-id	p3 value

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.

ID	Name	Object Type
> Office Furniture	Office Furniture	Folder
> Presentation & Home Theater Projectors	Presentation & Home Theater Projectors	Folder

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()'), "\n")
```

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.

Name	Value
> ID	4404
> Name	Projector 15
> Object Type	TIFF Image
> Approved	Last Approved on Tue Mar 06 07:49:45 EST 2007
> GetClassificationsOfAnAsset	Office Furniture Presentation & Home Theater Projectors

Get Classifications Based on Type

Multiple types of classifications exist.

Reference Type	Target
> SupplierLink	Supplier Portal/Stibo systems/Products
> WebLink	Publication Structure/Web Structure/Chairs/Executive Chairs/Uncomfortable Chairs
> WebLink	Publication Structure/Web Structure/Chairs/New Chairs
> WebLink	Publication Structure/Web Structure/Chairs/Utility Chairs

A calculated attribute on the product 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 "/".

The screenshot shows the Stibo Systems interface. On the left is a 'Tree' view showing a hierarchy: Office Equipment > Projectors > Pens, Pencils, Rollerballs, and Sets > Chairs > Specialty Office Chairs > Executive Chairs > Office Chairs > Office Chair. On the right is the 'Office Chair' product details view. The 'Product' tab is active, showing a table with columns 'Name' and 'Value'. The row for 'GetClassificationsOfProducts' is highlighted with a red box, showing the value 'New Chairs/ Uncomfortable Executive Chairs/ Utility Chairs'.

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

The product is linked to a classification folder with a product to classification reference type, and includes metadata on that particular product to classification reference type link.

For example, Classroom Type Projectors has a product to classification reference type called WebLink reference (ID = 'WebInclude'), that has the metadata attribute prod2prod2class on it.

The screenshot shows the Stibo Systems interface. On the left is a 'Tree' view showing a hierarchy: Hand Tools > Office Equipment > Projectors > Classroom Type Projectors (Class A). On the right is the 'Classroom Type Projectors (Class A) rev.0.6 - References' view. The 'References' tab is active, showing a table with columns 'Reference Type' and 'Target'. The row for 'WebLink' is highlighted with a red box, showing the target 'Publication Structure/Web Structure/Chairs/Executive Chairs/Ergonomic Executive Chairs'. A tooltip is visible over the 'WebLink' row, stating 'This is a meta-data value on the link from product to classification'.

The calculated attribute formula could be written as:

```
list (iterate(classificationproductlinks("WebInclude"), 'value("prod2prod2class")'), '\n')
```

This will display the meta attribute value from all classification links with type WebInclude and the separator between the multiple links values is "\n" (new line).

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

The screenshot shows the Stibo Systems interface. On the left is a 'Tree' view showing a hierarchy: Office Equipment > Projectors > Classroom Type Projectors (Class A). On the right is the 'Classroom Type Projectors (Class A) rev.0.6' view. The 'Product' tab is active, showing a table with columns 'Name' and 'Value'. The row for 'GetMetaDataFromLinkBetweenProdToClass' is highlighted with a red box, showing the value 'This is a meta-data value on the link from product to classification'.

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.

Tree

- Office Equipment
 - Projectors
 - Classroom Type Projectors (Class A)
 - Home Theater
 - Office Projectors
 - Portable Projectors

Classroom Type Projectors (Class A) rev.0.6 - Ref

Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Tables	Categories
		Reference Type	Target				
> SupplierLink		+	Supplier Portal/My_Supplier/My_Supplier Products				
> WebLink		+	Publication Structure/Web Structure/Chairs/Executive Chairs/Ergonomic Executive Chairs				

Tree

- Comfortable Executive Chairs
- Ergonomic Executive Chairs
- Uncomfortable Chairs
- New Chairs
- Utility Chairs
- Sample Data & Images
- Supplier Portal
 - My_Supplier
 - My_Supplier Assets
 - My_Supplier Batches
 - My_Supplier Products

Ergonomic Executive Chairs rev.0.5 - Multi Editor

Multi Editor	References	Referenced By
> ID	Web_Ergonomic Executive Chairs	My_Supplier Products
> Name	Ergonomic Executive Chairs	My_Supplier Products
> Object Type	Web Group	Supplier Products
> Path	Classification 1 root/Publication Structure/Web Structure/Chairs/Executive Chairs/Ergonomic Executive Chairs	Classification 1 root/Supplier Portal/My_Supplier/My_Supplier Products
> Approved	Last Approved on Wed Jul 22 09:04:08 EDT 2015	Last Approved on Thu Jul 03 11:08:35 EDT 2014
> Translation	Not Translated	Not Translated
> Class Owner	Peter Piper	Molly Malone

The calculated attribute formula could be written as:

```
{
  meta := iterate(iterate(classificationproductlinks() , 'CPLClassification()'), 'value
  ("ClassOwner") '),
  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 multiseper character tag as the list was converted to a multivalue list before being displayed.

Tree

- Office Equipment
 - Projectors
 - Classroom Type Projectors (Class A)
 - Home Theater

Classroom Type Projectors (Class A) rev.0.6 - Ref

Product	Sub Products	References	Referenced By	Images & Documents
Name		Value		
> GetMetaDataFromClassesLinkedToProducts		fx	Peter Piper/Molly Malone	

Should a user want to access the following options, please use the corresponding formulas:

Desired Outcome	Formula
Access the classification object directly	classification([Link-type]). [Link-type] is optional.
Access the link attributes present on the product to classification links	classificationproductlinks([Link-type]). [Link-type] is optional.
Access the product	CPLproduct()
Access the classification	CPLclassification()
Access the values in product object	classificationproductlinkis([Link-type]) and CPLproduct() in combination [Link-type] is optional.

Use Case: Calculated Attribute for Overrides

A product has been overridden several times.

Tree		Diamond Cut End Mills	
+	Diamond Cut End Mills (Table Guide) Series J	Product	Sub Products
+	Table Guide 2 → Diamond Cut End Mills (Table Guide) Series J	References	Referenced By
+	Table Guide 3 → Diamond Cut End Mills (Table Guide) Series J	Images & Documents	Co
+	Table Guide 4 → Diamond Cut End Mills (Table Guide) Series J	Used By Product Overrides	
+	Table Guide 5 → Diamond Cut End Mills (Table Guide) Series J	ID	Name
+	Table Guide 6 → Diamond Cut End Mills (Table Guide) Series J	> Table Guide 10	Table Guide 10 → Diamond Cut End Mills (Table Guide) Series J
+	Table Guide 7 → Diamond Cut End Mills (Table Guide) Series J	> DC End Mills Table Guide	Table Guide 2 → Diamond Cut End Mills (Table Guide) Series J
+	Table Guide 8 → Diamond Cut End Mills (Table Guide) Series J	> DC TableGuide 3	Table Guide 3 → Diamond Cut End Mills (Table Guide) Series J
+	Table Guide 9 → Diamond Cut End Mills (Table Guide) Series J	> Table Guide 4	Table Guide 4 → Diamond Cut End Mills (Table Guide) Series J
+	Table Guide 10 → Diamond Cut End Mills (Table Guide) Series J	> Table Guide 5	Table Guide 5 → Diamond Cut End Mills (Table Guide) Series J
+	Diamond Cut End Mills (Table Guide) Series L	> Table Guide 6	Table Guide 6 → Diamond Cut End Mills (Table Guide) Series J
+	Diamond Cut End Mills (Table Guide) Series M	> Table Guide 7	Table Guide 7 → Diamond Cut End Mills (Table Guide) Series J
		> Table Guide 8	Table Guide 8 → Diamond Cut End Mills (Table Guide) Series J
		> Table Guide 9	Table Guide 9 → Diamond Cut End Mills (Table Guide) Series J

For more information on product overrides, see the **Product Overrides** topic in the **Getting Started / User Guide** documentation.

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 := list(iterate(overriddenby(), 'stepname()'), '\n')
}
concatenate("This product is overridden by ", x, " product override(s): \n", 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 final output is assembled using a text string, the list count, and the STEPNAME of the overrides with new lines for each name.

Tree

Diamond Cut End Mills (Table Guide) Series J

Table Guide 2 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 3 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 4 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 5 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 6 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 7 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 8 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 9 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 10 → Diamond Cut End Mills (Table Guide) Series J

Diamond Cut End Mills (Table Guide) Series L

Diamond Cut End Mills (Table Guide) Series J

Product

Sub Products

References

Referenced By

Images & Documents

Cor

Name

>

>

Value

This product is overridden by 9, product override(s):

Table Guide 10

Table Guide 8

Table Guide 3

Table Guide 6

Table Guide 5

Table Guide 4

Table Guide 9

Table Guide 2

Table Guide 7

Get Overridden Product Object

A calculated attribute on the product override object 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.

Tree

Diamond Cut End Mills (Table Guide) Series J

Table Guide 2 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 3 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 4 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 5 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 6 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 7 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 8 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 9 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 10 → Diamond Cut End Mills (Table Guide) Series J

Diamond Cut End Mills (Table Guide) Series L

Table Guide 2 → Diamond Cut End Mills (Table Guide) Series J

Product

Sub Products

References

Referenced By

Images & Documents

Cor

Description

Name

>

>

Value

ID

DC End Mills Table Guide

Name

Table Guide 2

Object Type

Standard Product-Override

Approved

Last Approved on Fri Feb 13 12:43:05 EST 2015

Overridden Product

Diamond Cut End Mills (Table Guide) Series J (DCEM-100)

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

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

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, the list is displayed directly.
- The final output is assembled using a text string and the name of the product that is overridden.

Tree

Diamond Cut End Mills (Table Guide) Series J

Table Guide 2 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 3 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 4 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 5 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 6 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 7 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 8 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 9 → Diamond Cut End Mills (Table Guide) Series J

Table Guide 10 → Diamond Cut End Mills (Table Guide) Series J

Diamond Cut End Mills (Table Guide) Series L

Table Guide 2 → Diamond Cut End Mills (Table Guide) Series J

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Tables

F

Name

>

>

Value

GetOverriddenProductObject

This PO overrides the product "Diamond Cut End Mills (Table Guide) Series J"

Get Product Override Parents

A product that is used in one or more product overrides can have multiple parents.

Tree

Diamond Cut End Mills (Table Guide) Series J

Table Guide 2 → Diamond Cut End Mills (Table Guide) Series J

8J5H1

8J5H2

8J5H3

8J5H4

8J5H5

8J5H6

8J5F4

8J5F5

8J5F6

8J5F7

8J5H1

ProductReferencesReferenced ByImages & DocumentsCommercialTablesCa

Linked into Product Overrides

ID	Name
Table Guide 10	Table Guide 10 → Diamond Cut End Mills (Table Guide) Series J
DC End Mills Table Guide	Table Guide 2 → Diamond Cut End Mills (Table Guide) Series J
DC TableGuide 3	Table Guide 3 → Diamond Cut End Mills (Table Guide) Series J
Table Guide 4	Table Guide 4 → Diamond Cut End Mills (Table Guide) Series J
Table Guide 5	Table Guide 5 → Diamond Cut End Mills (Table Guide) Series J
Table Guide 6	Table Guide 6 → Diamond Cut End Mills (Table Guide) Series J
Table Guide 7	Table Guide 7 → Diamond Cut End Mills (Table Guide) Series J
Table Guide 8	Table Guide 8 → Diamond Cut End Mills (Table Guide) Series J
Table Guide 9	Table Guide 9 → Diamond Cut End Mills (Table Guide) Series J

A simple metadata calculated attribute on the main product can display the IDs of the parents for all product override objects. This could be written with the following formula:

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

The result of this formula is:

Tree

Diamond Cut End Mills (Table Guide) Series J

Table Guide 2 → Diamond Cut End Mills (Table Guide) Series J

8J5H1

8J5H2

8J5H3

8J5H4

8J5H5

8J5H6

8J5F4

8J5F5

8J5F6

8J5F7

8J5H1 rev.0.1 - Product

CommercialTablesCategory ProfileProof ViewPackaging HierarchyStatusState LogTasks

ProductReferencesReferenced ByImages & Docu

Description

Name	Value
ID	8J5H1
Name	8J5H1
Object Type	Product
Approved	Never Been Approved
GetOverrideParents	Table Guide 10/ Table Guide 8/ DC TableGuide 3/ Table Guide 6/ Table Guide 5/ Table Guide 4/ Table Guide 9/ DC End Mills Table Guide/ Table Guide 7

Use Case: Calculated Attribute for References

A calculated attribute is helpful for when an object has references to multiple other products. The following scenarios are examples of how calculated attributes could be used.

Get List of References

Calculated attributes could be used to get a list of references, and display them on the Product tab. 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.

Reference Type	Target	ID	Name	Thumbnail
Alternative	Office Projectors	Office Projectors	Office Projectors	
Alternative	Portable Projectors	Portable Projectors	Portable Projectors	

The formula for this calculated attribute could look like the formula below:

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

To break this formula down, the first 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'.

Because this list of labels is not needed, having the first ITERATE() function surround the second ITERATE() function, makes it so that the names of the referenced products are retrieved from the labels:

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

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

Tree

- Hand Tools
- Office Equipment
 - Projectors
 - Classroom Type Projectors (Class A)
 - ABC-1001-100
 - ABC-1001-101
 - ABC-1001-102
 - ABC-1001-103
 - ABC-1001-104
 - ABC-1001-105
 - Home Theater
 - Office Projectors

Classroom Type Projectors (Class A)

Product		Sub Products	References	Referenced By	Images & Documents	Cor
Description						
Name	>	>	Value			
ID	>		Classroom Projectors			
Name	>		Classroom Type Projectors (Class A)			
Object Type	>		Product Family			
Approved	>		Last Approved on Wed Feb 11 11:37:08 EST 2015			
Default InDesign template	>		ProductTemplate1 (7396)			
GetListOfRefs	>		Office Projectors, Portable Projectors			

Get Value From a Referenced Product

Using this formula for a calculated attribute allows a user to look to a reference on an object, get a value that is on the reference, and display it on the object. 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:

Tree

- Hand Tools
- Office Equipment
 - Projectors
 - Classroom Type Projectors (Class A)
 - Home Theater
 - Office Projectors
 - Portable Projectors
 - Utility Projectors

Products

ProductsReferencesReferenced By

View: PowerCordLength view

	ID	>	Name	>	PowerCordLength	>
Name	>	ID	Name	>	PowerCordLength	
Office Projectors	>	Office Projectors	Office Projectors	>	12 ft	
Portable Projectors	>	Portable Projectors	Portable Projectors	>	9 ft	

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.

Tree

- Office Equipment
 - Projectors
 - Classroom Type Projectors (Class A)
 - Home Theater

Classroom Type Projectors (Class A)

Product		Sub Products	References	Referenced By
Name	>	>	Value	
GetValueFromRefProduct	>		Cord Lengths = 12/9	

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

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

Reference Type	Target	ID	Name	Thumbnail	EquivalenceRating
Alternative	Office Projectors	Office Projectors	Office Projectors		Functional Equivalent
	Portable Projectors	Portable Projectors	Portable Projectors		Downsell Equivalent

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/>", "\n ")
}
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.

Name	Value
GetMetaDataFromLinkBetween	Functional Equivalent Downsell Equivalent

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.

Reference Type	Source	ID	Name
Alternative	3D Projectors	3D Projectors	3D Projectors
	Portable Projectors	Portable Projectors	Portable Projectors

Tree

- Office Equipment
 - Projectors
 - Classroom Type Projectors (Class A)
 - Home Theater
 - Office Projectors
 - Portable Projectors
 - Utility Projectors
 - 3D Projectors

Products

ProductsReferencesReferenced By

View: Revisions view

ID	Name	Revision
Portable Projec...	Portable Projectors	0.4 Last edited by USER on Thu May 12 09:10:18 EDT 2016
Utility Projectors	Utility Projectors	0.1 Last edited by USER on Wed Jan 06 17:10:46 EST 2016
3D Projectors	3D Projectors	0.1 Last edited by USER on Thu May 12 16:43:12 EDT 2016

The following calculated attribute formula could be written:

```
{
  refnames := iterate(iterate(referencesto("product", "Alternative"), 'referenceowner()'),
    'stepname()'),
  editdates := iterate(iterate(referencesto("product", "Alternative"), 'referenceowner()'),
    'revisioneditdate()'),
  combo := iterate(refnames, 'concatenate(item, " Last edited on ", listitem
    (editdates,index))'),
  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.

Tree

- Office Equipment
 - Projectors
 - Classroom Type Projectors (Class A)
 - Home Theater
 - Office Projectors

Home Theater rev.

ProductSub ProductsReferencesReferenced ByImages & DocumentsCommercial

Name	Value
GetAttrValueFromReferenced	3D Projectors Last edited on 2016-05-12 16:43:12
	Portable Projectors Last edited on 2016-05-12 09:10:18
	Utility Projectors Last edited on 2016-01-06 17:10:46

Get Reference Type ID

In this example, 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.

Tree

Office Equipment

Projectors

Classroom Type Projectors (Class A)

Home Theater

Office Projectors

Portable Projectors

Utility Projectors

3D Projectors

Pens, Pencils, Rollerballs, and Sets

Chairs

Training

New Supplier Products

Product

Sub Products

References

Referenced By

Product Link Metadata

Reference Type	Target
Alternative	Office Projectors
	Portable Projectors

Ungrouped Product References

Reference Type	Target
Spare Part	Hammers
	Pliers

The following calculated attribute formula could be written:

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

concatenate(refids)
```

The results display each of the reference type IDs.

Tree

Office Equipment

Projectors

Classroom Type Projectors (Class A)

Home Theater

Office Projectors

Portable Projectors

Utility Projectors

3D Projectors

Pens, Pencils, Rollerballs, and Sets

Chairs

Classroom Type Projectors

Product

Sub Products

References

Referenced By

Images & Documents

Co

Description

Name	Value
ID	Classroom Projectors
Name	Classroom Type Projectors (Class A)
Object Type	Product Family
Approved	Last Approved on Wed Feb 11 11:37:08 EST 2015
GetRefTypeID	Alternative, Spare Part

Although accurate, the display could be more informative with some modifications. You could filter the list so that each of the names and/or IDs of the referenced products together with the reference type displays.

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
Reference 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.

Tree

Office Equipment

Projectors

Classroom Type Projectors (Class A)

Home Theater

Office Projectors

Portable Projectors

Utility Projectors

3D Projectors

Pens, Pencils, Rollerballs, and Sets

Chairs

Training

Classroom Type Projectors (Class A) rev.0.6 - Product

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Tables

Category Profile

Description

Name	>	>	Value
ID	>		Classroom Projectors
Name	>		Classroom Type Projectors (Class A)
Object Type	>		Product Family
GetProdRefNamesAndRefTypes	>	fx	Product Office Projectors is being referenced with the Reference Type of Alternative Product Portable Projectors is being referenced with the Reference Type of Alternative Product Hammers is being referenced with the Reference Type of Spare Part Product Pliers is being referenced with the Reference Type of Spare Part

Getting the Reference Source or Target products' information on the link

The calculated attribute can also be used to show metadata on the References link. Here a user could use it to show the Source product and the Target product for the link. This can be done like below.

```
iterate(referenceowner(),'stepid()') // to get the Source product ID
iterate(referencetarget(),'stepname()') // to get the Target product ID
```

System Setup

Metadata

Asset Metadata

Link Metadata

Publication Metadata

Unique Identifiers

Attribute Description

Attribute Help Text

Category

Display Name

DisplaySequence

GetSourceProductID

GetTargetProductName

Group Information

Justification

Landing Page Copy

linkbetweenproducts

Parent

Path

GetSourceProductID - Attribute

ValidityProfileLogState LogTasks

AttributeReferencesAttribute Transformation

Description

Name	Value
ID	GetSourceProductID
Name	GetSourceProductID
Last edited by	2017-03-22 07:33:09 by USERL
Full Text Indexable	No
Externally Maintained	No
Hierarchical Filtering	None
Calculated	Yes
Type	Description
Dimension Dependencies	
Value template	iterate(referenceowner(),'stepid()')

When configured properly, the metadata calculated attributes appear like below.

Tree

Footwear

Safety

Hardware

Tools

Task Lighting

Flashlights

Flashlights Items

111204 Child of Overrides Produ

Flashlight Case

Flashlight Clip

114859

114861

20862

20883

LED Pocket Flashlight

LIGHTS-10

LIGHTS-10 rev.0.6 - References

CommercialTablesCategory ProfileProof ViewStatusState LogTasks

ProductSub ProductsReferencesReferenced ByImages & Documents

0% complete

Owners Manual

Image References

Item References, Classification

Item References, Product

Packaging Hierarchy References

Sales Item References, Classification

Sales Item References, Product

Reference Type	GetTargetProductName	GetSourceProductID
Accessory Required	Flashlight Case	110306
	Flashlight Clip	110306
Cross Reference		

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**.

Condition - Validity

AttributeReferencesAttribute TransformationValidityProfileLogState LogTasks

☒ Valid for Product Types

☒ Valid for Classification Types

☒ Valid for Asset Types

☒ Valid for Entity Types

☒ Valid for Publication Types

☒ Valid for Version Types

☒ Valid for E-Catalog Types

☒ Valid for Link Types

☐ All

☐ Classification attribute validation

☒ Product attribute validation

☒ Valid for Product Reference Types

☒ Valid for Classification Reference Types

☒ Valid for Asset Reference Types

☒ Valid for Product to Classification Link Types

☒ Valid for Entity Reference Types

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.

System Setup

System Settings

System SettingsLog

Validation Templates

IDRegular Expression

Create Validation Template

Product Information Manager Default Settings

NameValue

> Enforce Mandatory Check for Attributes, References and Linksnone

> Product Editor, Group attributes by top groupN

> Localize numbers with thousand delimiter when localizing exportsY

> Localize dates when localizing exportsY

> Report Save As CSV Character Setclient-locale

> Default Attribute to use as Display Sequence AttributeDisplaySequence

> Default Completeness MetricCompleteness Metric B

> Conditional Validity AttributeConditionAttribute

> Block Attribute Groups with more than 1000 attributesY

> Use full pathname for classes on Product References TabY

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.

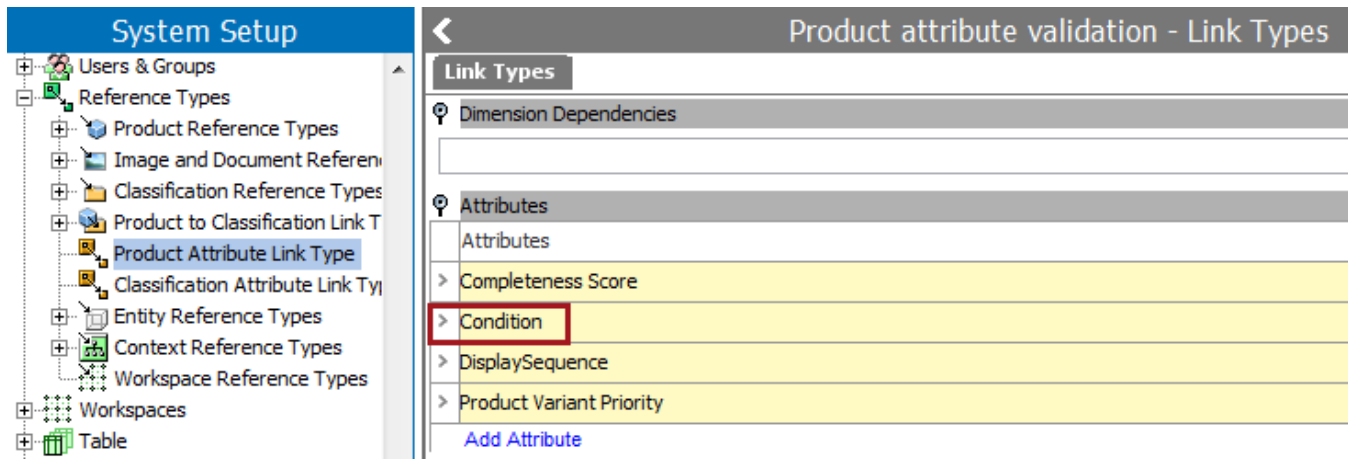
3. Next, navigate to **System Setup** and select the desired Description Attribute.
4. In the Conditional Attribute field, enter the desired condition rules for the Conditionally Valid Attribute to follow.

Attribute		References	Attribute Transformation
Description			
Name	>	>	Value
> ID			Hazmat
> Name			Hazmat
> Last edited by			2015-06-09 22:04:38 by USER
> Full Text Indexable			No
> Externally Maintained			No
> Calculated			No
> Type			Description
> Dimension Dependencies			
> Mandatory			No
> Condition Attribute			HasLead = Yes

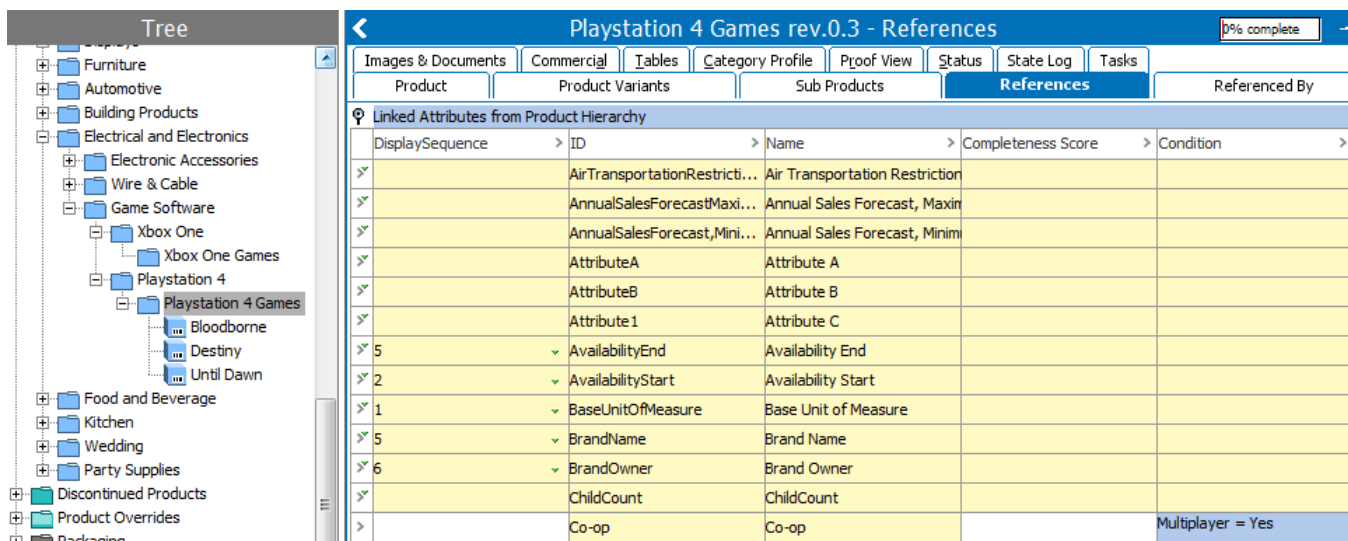
Specification Attributes

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.



- Next, navigate to the **Tree** and open the relevant node in the product hierarchy.
- 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 'Hazard' 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 on creating Smartsheets exports, see the **Smartsheet Data and Template Configurations** section of the **Smartsheets** 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

Ir

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.

System Setup

- Category Specific Attributes
- Customer Data
- Dates
- DescriptionAttribute
- Dimensions
- Display
- ETIM6 Attributes
- ETIM Attributes
- Event Filtering
- GDSN Receiver System Attribute
- GDSN System Attributes
- GDSN Validation Group
- Manual Sequencing
- Metadata
 - Asset Metadata
 - Link Metadata
 - Publication Metadata
 - Unique Identifiers
 - Attribute Description
 - Attribute Help Text
 - Category
 - Configuration Description
 - Discontinued
 - Display Name
 - DisplaySequence
 - External Item Description
 - Group Information
 - Justification
 - Landing Page Copy

Attribute Description - Attribute

Validity
Profile
Log
State Log
Tasks

Attribute
References
Attribute Transformation

Description

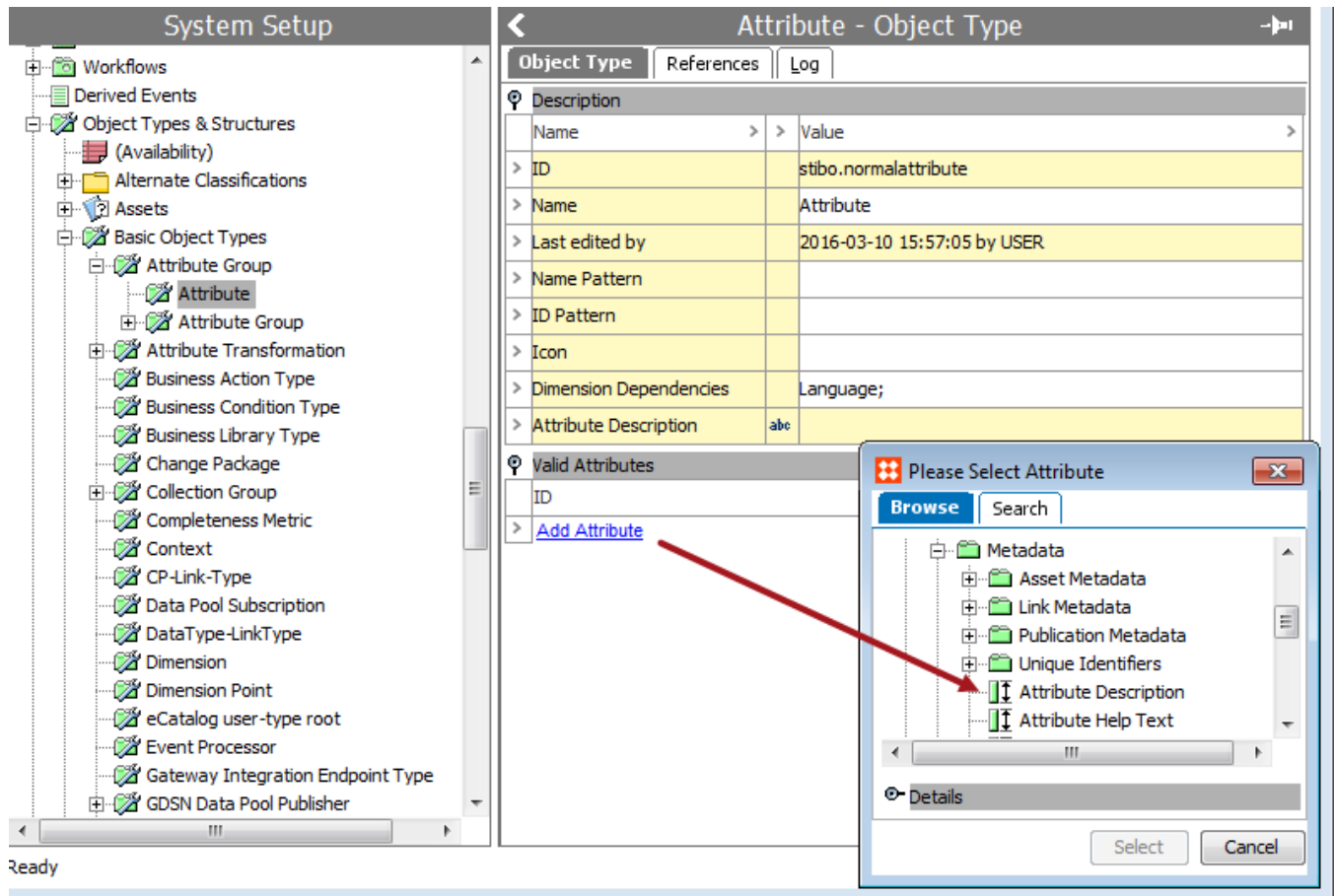
Name	Value
ID	Attribute Description
Name	Attribute Description
Last edited by	2016-03-14 10:22:45 by USER
Full Text Indexable	No
Externally Maintained	Yes
Hierarchical Filtering	None
Calculated	No
Type	Description
Dimension Dependencies	
Mandatory	No

Attribute Validation

Name	Value
Validation Base Type	Text
List Of Values	N/A
Multi Valued	No
Minimum Value	N/A
Maximum Value	N/A
Maximum Length	500

[Edit Validation Rule](#)

- 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.
- In the **Please Select Attribute** dialog, browse to or search for the Description attribute that was just created, then click **Select**.



- 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.

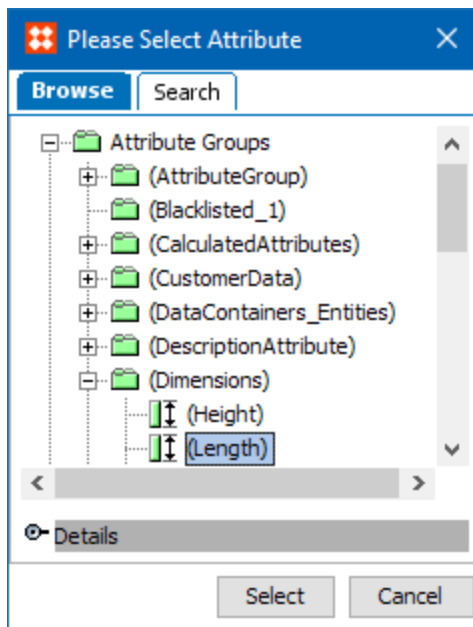
Some of the Object Types and linked Attributes are:

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.

Dimension Dependent Attributes

When attribute values need to be dimension dependent—typically for translation purposes—they are made dimension dependent on a one-by-one basis on the attributes themselves.

Note: Making the attribute **object type** dimension dependent only impacts the ability of the attribute **names** to be translated.

Dimension Dependent Attribute Values

An example of an attribute where the values are dimension dependent on language could be 'Primary Color.' The following also applies to values contained in STEP **names** of objects.

- For **English**, the values would 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'

The below screenshot shows two more examples of dimension-dependent attributes, one dependent on a language dimension and one dependent on a country dimension.

1. The attribute 'Long Description' is dimension dependent on **Language**, meaning that its content may differ based on what language dimension point is used in the context. The typical usage of dimension-dependent data is when content is translated, allowing an attribute to have different values for different languages.
2. By contrast, the 'Warranty' attribute is dimension dependent on **Country**, meaning that its content may differ based on what country dimension point is used in the context, since a product's warranty period may depend on the country in which it is sold.

A **context** that uses the language dimension point of **English** and the country **dimension point** of **USA** would contain an **English**-language long description and **USA**-specific warranty content. Additionally, Warranty could be made *both* country and language dependent to ensure that the warranty text is always in English.

The screenshot displays the 'System Setup' interface for two attributes: 'Long Description' and 'Warranty'. On the left, a tree view shows the hierarchy of contexts: 'Contexts' (circled in red), 'Language' (with 'All Languages' circled in red), and 'Country' (with 'All Countries' circled in red). Red arrows indicate the configuration process: arrow 1 points from 'All Languages' to the 'Long Description' attribute, and arrow 2 points from 'All Countries' to the 'Warranty' attribute. The right side shows the 'Attribute' configuration table for each attribute.

Attribute	References	Att
> Calculated	No	
> Type	Specification	
> Dimension Dependencies	Language;	
> Mandatory	No	

Attribute	References	Att
> Calculated	No	
> Type	Specification	
> Dimension Dependencies	Country;	

Non-Dimension Dependent Attribute Values

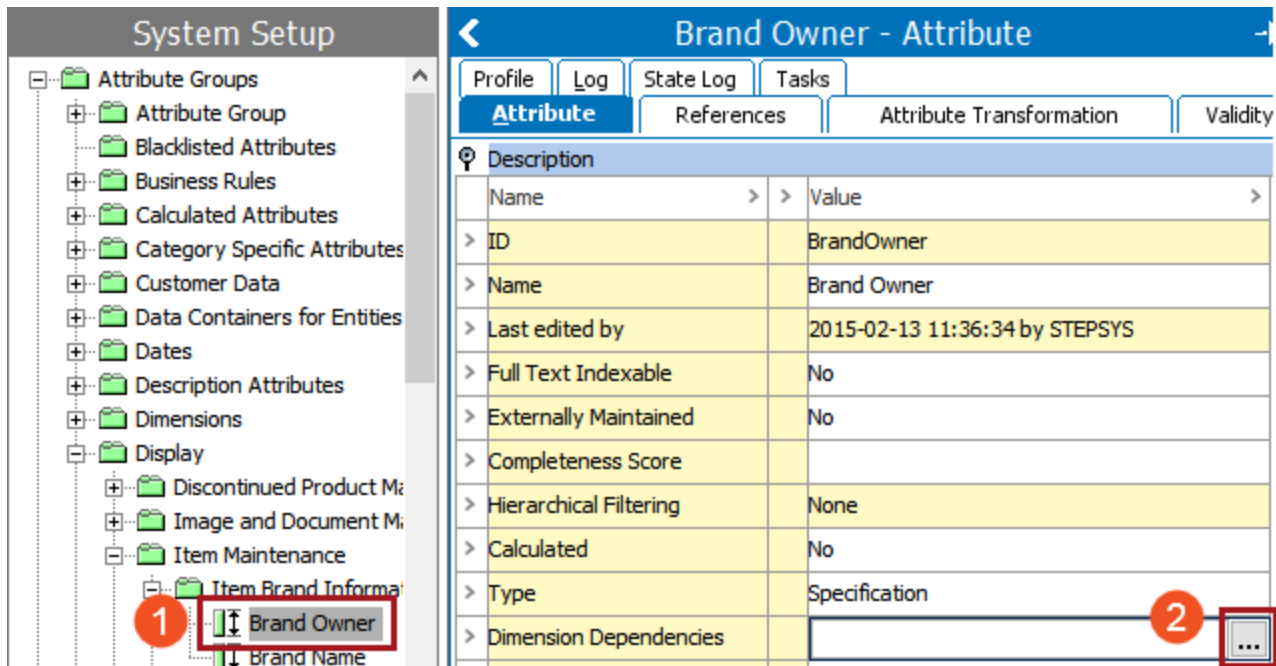
No dimension dependency means that data is stored in the **All** level (also known as the **Root** level) of all dimensions. The following also applies to values contained in STEP **names** of objects.



Note: If you have information that will never vary by language or country, you should **never** assign a dimension dependency on it. When you **do** have information that truly will be different by language or country, assign the dimension dependency right away. If you later apply a dimension dependency to an attribute, new data will be stored in the dimension point that you specified and existing data will remain at the 'All' level and should be moved. This can be very difficult to correct later on. For more information on this problem and how to correct it, see the **Adding a Dimension Dependency After Loading Data** topic.

Setting Dimension Dependency on Attributes

1. In System Setup, navigate to the relevant attribute under Attribute Groups.
2. On the **Attribute** tab, double-click in the **Dimension Dependencies** field, then click the ellipsis button (...).

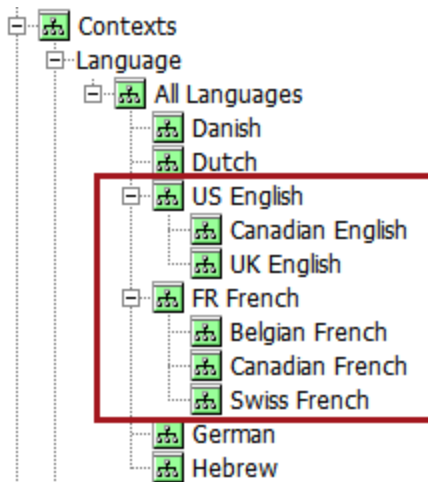


3. In the **Select Dimension Dependencies** dialog box that displays, check the box(es) for the relevant dimension(s), then click **Apply** to set the dimension dependency.

Value Inheritance in Dimension Dependent Attributes

There sometimes may be a need for parent / child relationships between dimensions if you need to maintain many versions of an 'almost the same' language, as pictured in the below screenshot.

Note: This is not recommended. In most circumstances, dimension points should be created at the same level. For more information on best practices with regard to dimensions, see the **Dimension and Dimension Points Best Practices** topic.



There may be valid use cases to maintain several versions of the same language, for example, US English, Canadian English, and UK English. Since maintaining translations requires effort and carries a cost, it may be of benefit to maintain a base version of English and then only translate a small percentage of the content for Canadian English and UK English. In these cases, there must be a clear owner of the parent language.

The most simple explanation of inheritance of values from dimension points can be explained using the example of a **US English** parent language dimension and a **UK English** child language dimension. Inheritance within dimension points behaves exactly like inheritance in other nodes in STEP, such as the inheritance of attribute values from parent to child product objects.

For example, the value of Long Item Description in the parent **US English** language dimension (which is used by the English US context) looks as follows:

Context
English US

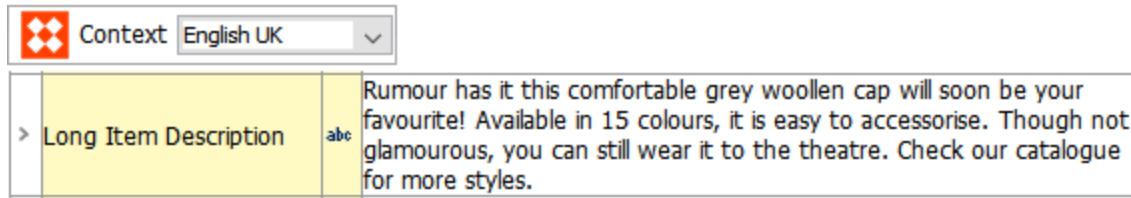
> Long Item Description	abc	Rumor has it this comfortable gray woolen cap will soon be your favorite! Available in 15 colors, it is easy to accessorize. Though not glamorous, you can still wear it to the theater. Check our catalog for more styles.

Before translation, the content will inherit to the child context of **UK English** as-is. Inheritance of the value is indicated by a purple triangle. When the triangle is hovered over, the help text 'Text, inherited from other context' displays.

Context
English UK

> Long Item Description	abc	Rumor has it this comfortable gray woolen cap will soon be your favorite! Available in 15 colors, it is easy to accessorize. Though not glamorous, you can still wear it to the theater. Check our catalog for more styles.

After translation, the content for UK English appears as follows. The purple triangle is gone, as the value is no longer inherited but has been overwritten by the inherited content. If the UK English content is deleted, the US English content will reappear.



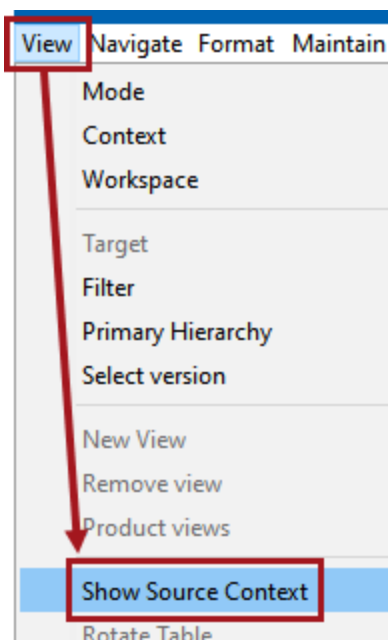
This can save on translation costs because, instead of sending US English content to a translation company to convert it to UK English, the US English content could be exported from STEP, a user could run a find and replace operation on the content to replace a certain subset of words within the content (such as changing color to colour), then the newly 'translated' content can be reimported into the UK English context to overwrite the US English content.

Showing Source Context (Parent Dimensions)

For an inherited attribute value, you can check to see what parent dimension it is inherited from by using the **Show Source Context** feature. This is useful for translation purposes to confirm what language owns the content.

Continuing from the example used in the previous section of this topic, the following steps illustrate the content of the Long Item Description attribute as viewed from the English Canada context and from within the English UK context.

1. With your product object selected in the STEP Workbench **Tree**, click on the Product tab.
2. Choose the relevant context from the **Context** dropdown list in the upper left corner of the GUI.
3. Navigate to View > **Show Source Context**.



4. A **Source** column displays to the right of the Value column that indicates the source dimension point of the values.

In this example, the content that exists in the English Canada context is inherited from the US English language dimension point.

 Context English Canada

Gray Cap rev.0.6 - Product

0% complete

Tables

Category Profile

Proof View

Status

State Log

Tasks

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Description

Item Description Information

Name	Value	Source
> Base Unit of Measure		
> Description Source	abc	
> Description Target	abc	
> Primary Color		
> Secondary Color		
> Short Item Description	abc	
> Exclusive?		
> Long Item Description	Rumor has it this comfortable gray woolen cap will soon be your favorite! Available in 15 colors, it is easy to accessorize. Though not glamorous, you can still wear it to the theater. Check our catalog for more styles.	US English

In the next example, the content that exists in the English UK context is not inherited from the US English language dimension point, since it has been placed directly into the UK English language dimension point and has overwritten the content that would have been inherited from the US English language dimension point.

Context

English UK

<

Grey Cap rev.0.6 - Product

0% complete

>

Tables

Category Profile

Proof View

Status

State Log

Tasks

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Description

Item Description Information

Name	Value	Source
> Base Unit of Measure		
> Description Source	abc	
> Description Target	abc	
> Primary Color		
> Secondary Color		
> Short Item Description	abc	
> Exclusive?		
> Long Item Description	abc	Rumour has it this comfortable grey woollen cap will soon be your favourite! Available in 15 colours, it is easy to accessorise. Though not glamorous, you can still wear it to the theatre. Check our catalogue for more styles.

Attribute Display Sequence

As per the default function in STEP, attributes shown in the Product Editor of a product are shown in alphabetical order. However, if an Attribute Display Sequence Attribute has been set up, 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.

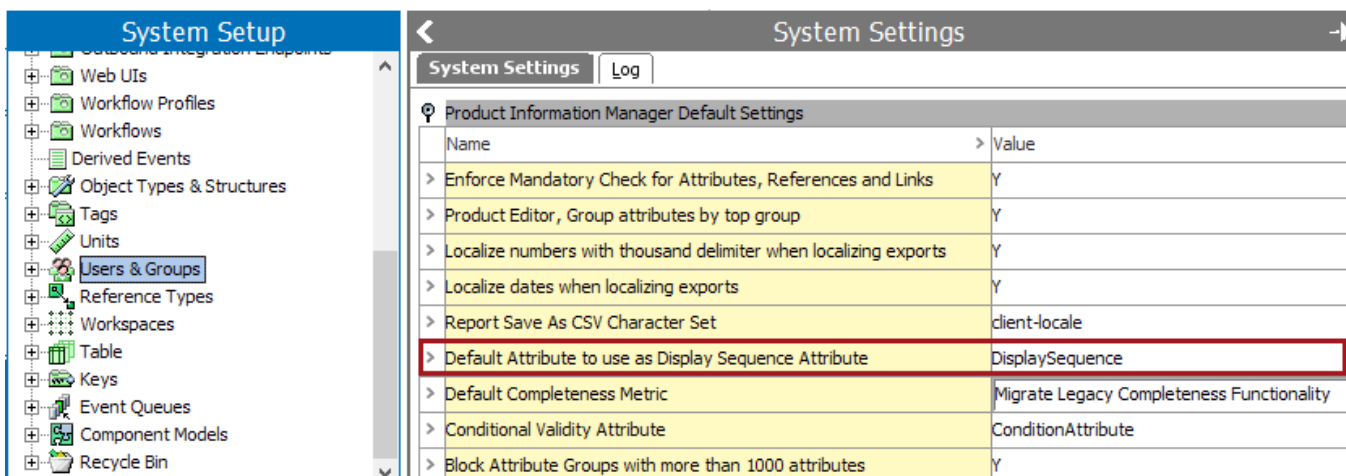
Attribute Display Sequence Setup

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

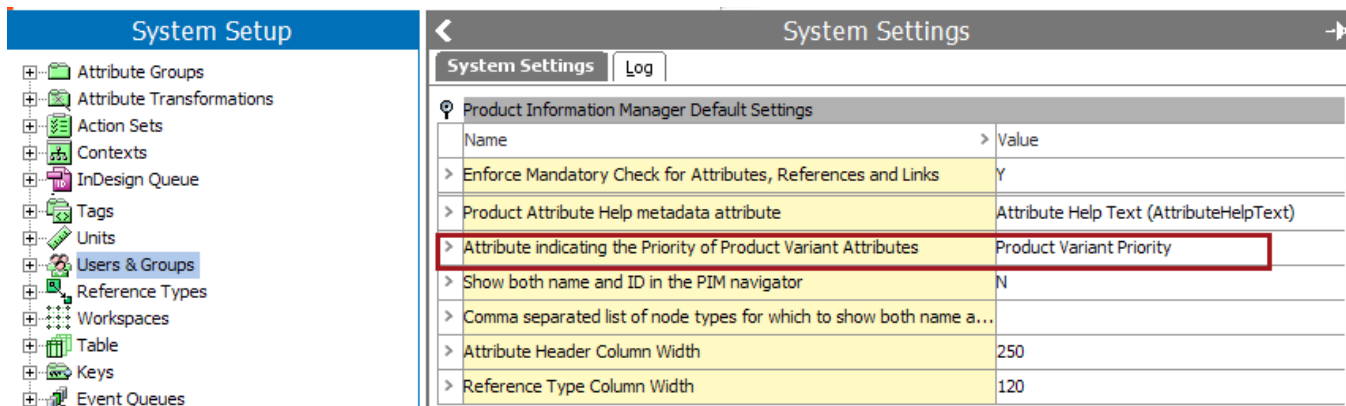
A Display Sequence Attribute has to meet the following conditions:

- The attribute has to be a **Description** attribute
- The attribute has a Validation Base Type or either **number** or **integer**
- The attribute has to be valid for the Link Type **Product attribute validation** and/or valid for the Object Type **Attribute**
- The display sequence attribute is selected in System Setup > Users & Groups > System Settings tab > Product Information Manager Default Settings flipper > **Default Attribute to use as Attribute Display Sequence Attribute**



- Similarly, the Display sequence for the Product variants object types can be setup by going to System Setup> Users & Groups > System Settings tab > Product Information Manager Default Settings flipper > **Attribute**

indicating the Priority of Product Variant Attributes

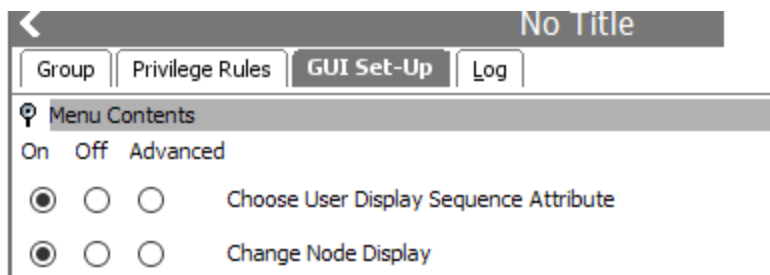


View Menu: Choose Attribute Display Sequence Attribute

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, and under the Menu Contents flipper locate the **Choose User Display Sequence Attribute** option.



Per default, the **Choose User 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.

Sorting Attributes

Once the attribute display sequence has been set up, a user can set the sorting. 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.

Name	Value
ID	brown
Name	brown
Last edited by	2017-01-30 13:42:48 by USERL
Full Text Indexable	No
Externally Maintained	No
Hierarchical Filtering	None
Calculated	No
Type	Specification
Mandatory	No
DisplaySequence	123 2

3. Type the appropriate sequence number for each attribute in an attribute group. You can choose to set a sequence by using example 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 a Linked Attributes flipper, type a number in the cell related to the Display Sequence attribute column in front of the relevant attribute. In the example below, the 'Linked Attributes from Product Hierarchy' is being used.

Tree

Flashlights

Flashlights Items

111204 Child

114859

114861

20862

20883

Flashlight Cas

Flashlight Clip

LED Pocket Fi

LIGHTS-10

Flashlights Select

<

LIGHTS-10 rev.0.6 - References

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Tables

Ungrouped Product References

Ungrouped Image & Document References

Index Words

Publications

Linked Attributes from Product Hierarchy

DisplaySequence	Completeness Score	ID	Name
> 5		BrandName	Brand Name
> 6		BrandOwner	Brand Owner
>		Brightness	Brightness

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.

Linked Attributes from Product Hierarchy

DisplaySequence	Completeness Score
>	

Hide

Show All Columns

Rotate Table

Link to Attribute Ctrl+Plus

Remove Link to Attribute Ctrl+Minus

Create Local Link

The symbol will change in the inherited column and also the background color of the row will change from yellow to white. You are now able to type a local value in the Display Sequence column.

Linked Attributes from Product Hierarchy

DisplaySequence	ID
> 5	# Flutes
> 10	Angle
>	Brightness
>	Color

→

Linked Attributes from Product Hierarchy

DisplaySequence	ID
> 5	# Flutes
> 10	Angle
>	Brightness
>	Color

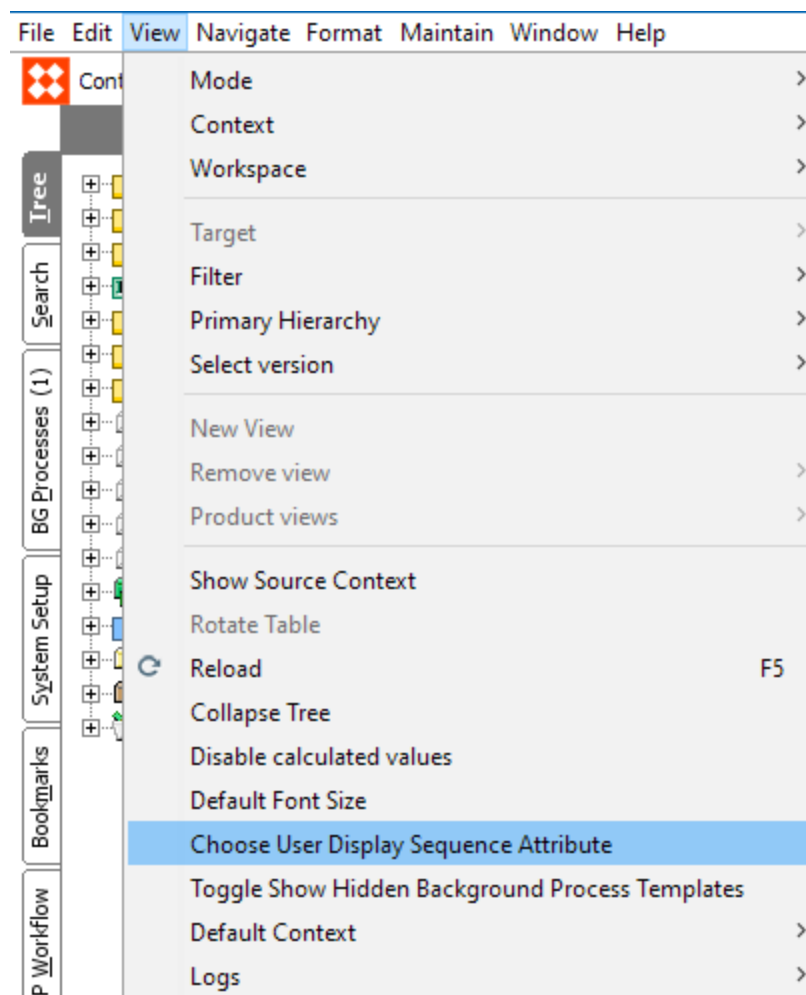
Note: Attributes that have no sequence number, and therefore are alphabetically sorted, will be placed last on the list.

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.

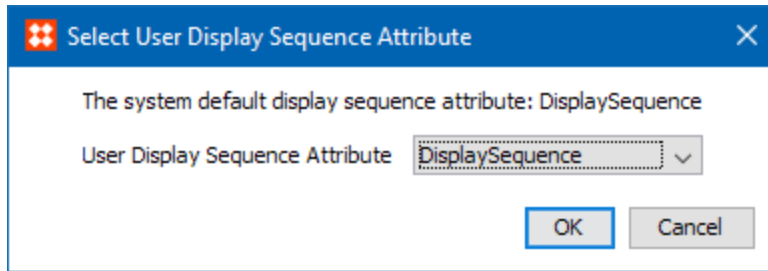
Selecting a Personal Attribute Display Sequence Attribute

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

1. Go to **View > 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.

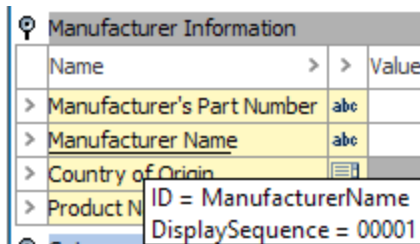
Display Sequence Attribute and the Product Editor Display

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.

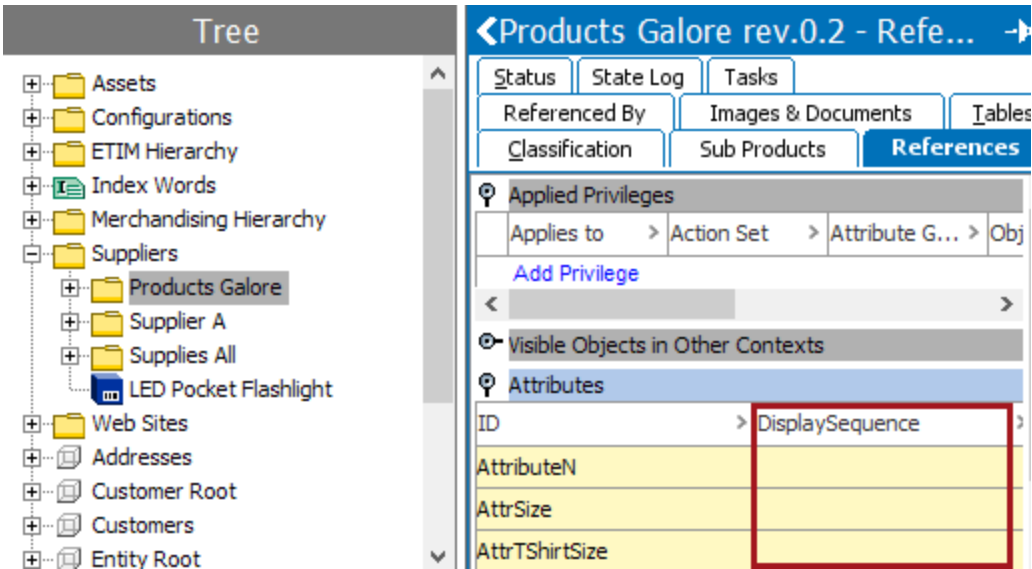
Note: If you select two or more products, the **Products** Editor shows the attributes always in alphabetical order. The reason for this is that you can select two or more products, that have the same attribute, but with different values for the Display Sequence attribute.

Display Sequence Attribute & Classifications

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

1. In Tree, navigate to a classification, select a folder, then click the **References** tab.

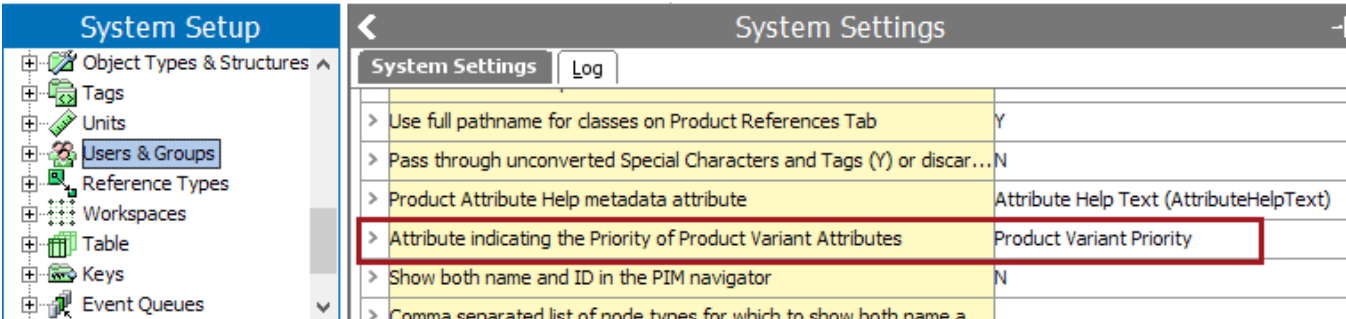
Under **Attributes** a Display Sequence attribute column is shown.



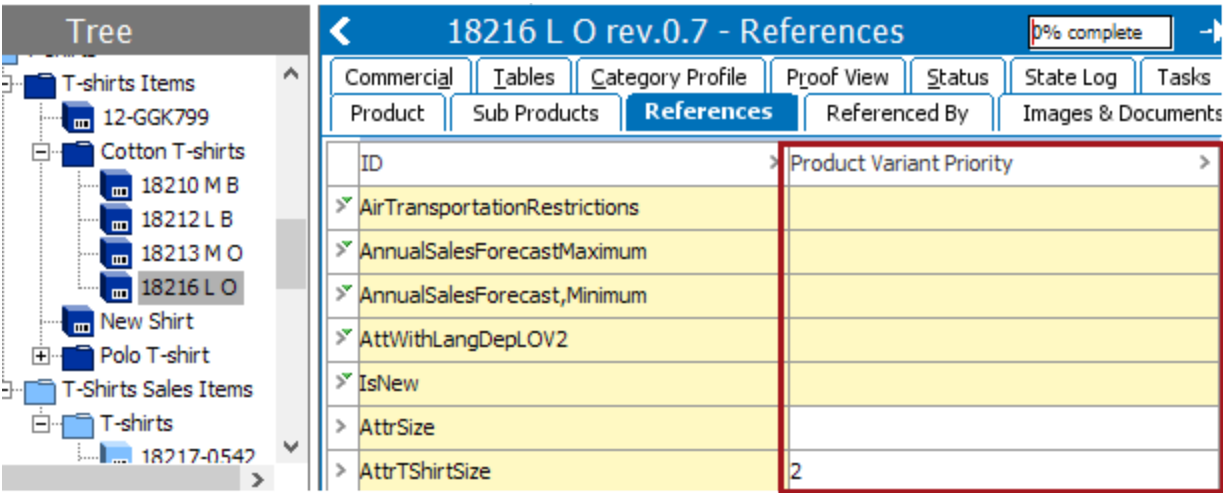
Important: It is of no use though to maintain the Attribute Display sequence here, as it will have no effect anywhere, not in the Classification editor, nor in an export.

Setting the Priority of a Product Variant Attribute

1. In Tree, select 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 the User and Groups under the **Product Information Manager Default Settings** flipper.



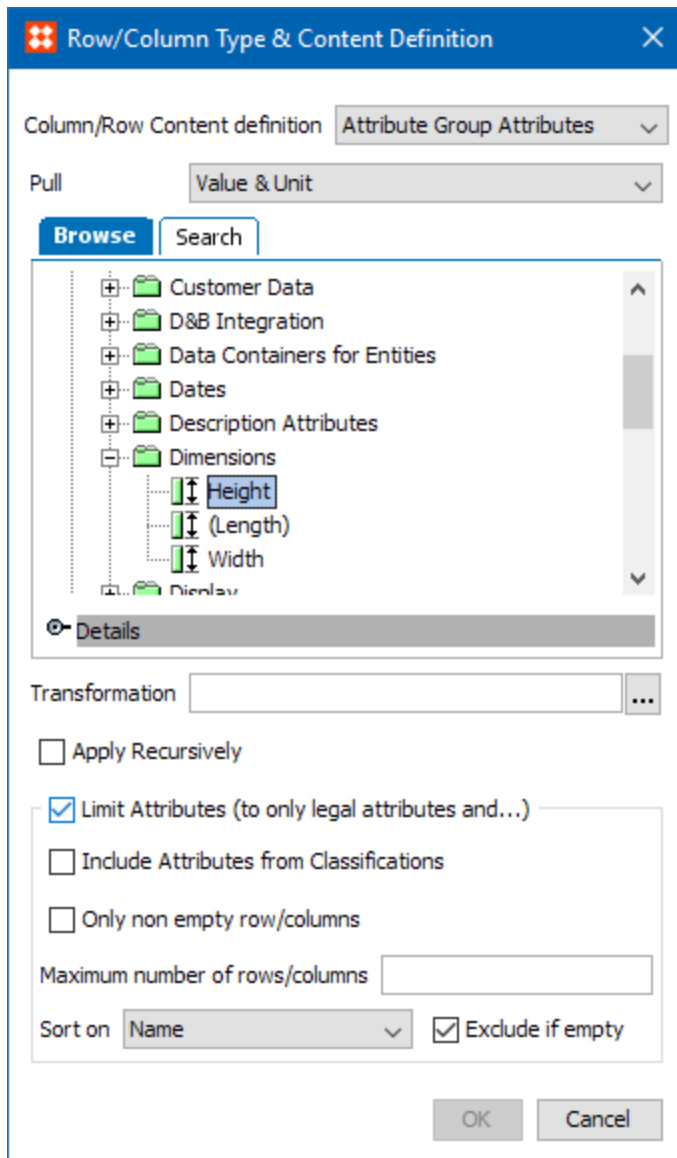
Display sequence values can be entered in the column related to the respective attributes.



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 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 multiselect to allow sequencing from a single location.

System Setup

Metadata

Asset Metadata

Catalog Group Theme

Catalog Theme

Circulation

Cover Photo Shot Due

Effective Date

Expiration Date

Get Inherited Pub Group Value

Get Inherited Publication Value

Get Inherited Section Value

Mail Date (same as Effective Date)

Object Name

Section Theme

Total Number of Items

Publication Metadata

Unique Identifiers

Multi Editor

ID	Name	Completeness Score	Display Name	DisplaySequence
> CatalogGroupTheme	Catalog Group Theme			50
> CatalogTheme	Catalog Theme			50
> Circulation	Circulation			60
> CoverPhotoShotDue	Cover Photo Shot Due			40
> EffectiveDate	Effective Date			20
> ExpirationDate	Expiration Date			30
> GetInheritedPubGroupValue	Get Inherited Pub Group V...			1
> GetInheritedPublicationValue	Get Inherited Publication V...			1
> GetInheritedSectionValue	Get Inherited Section Value			1
> MailDate	Mail Date (same as Effecti...			10
> ObjectName	Object Name			25
> SectionTheme	Section Theme			50
> TotalNumberOfItems	Total Number of Items			15

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.

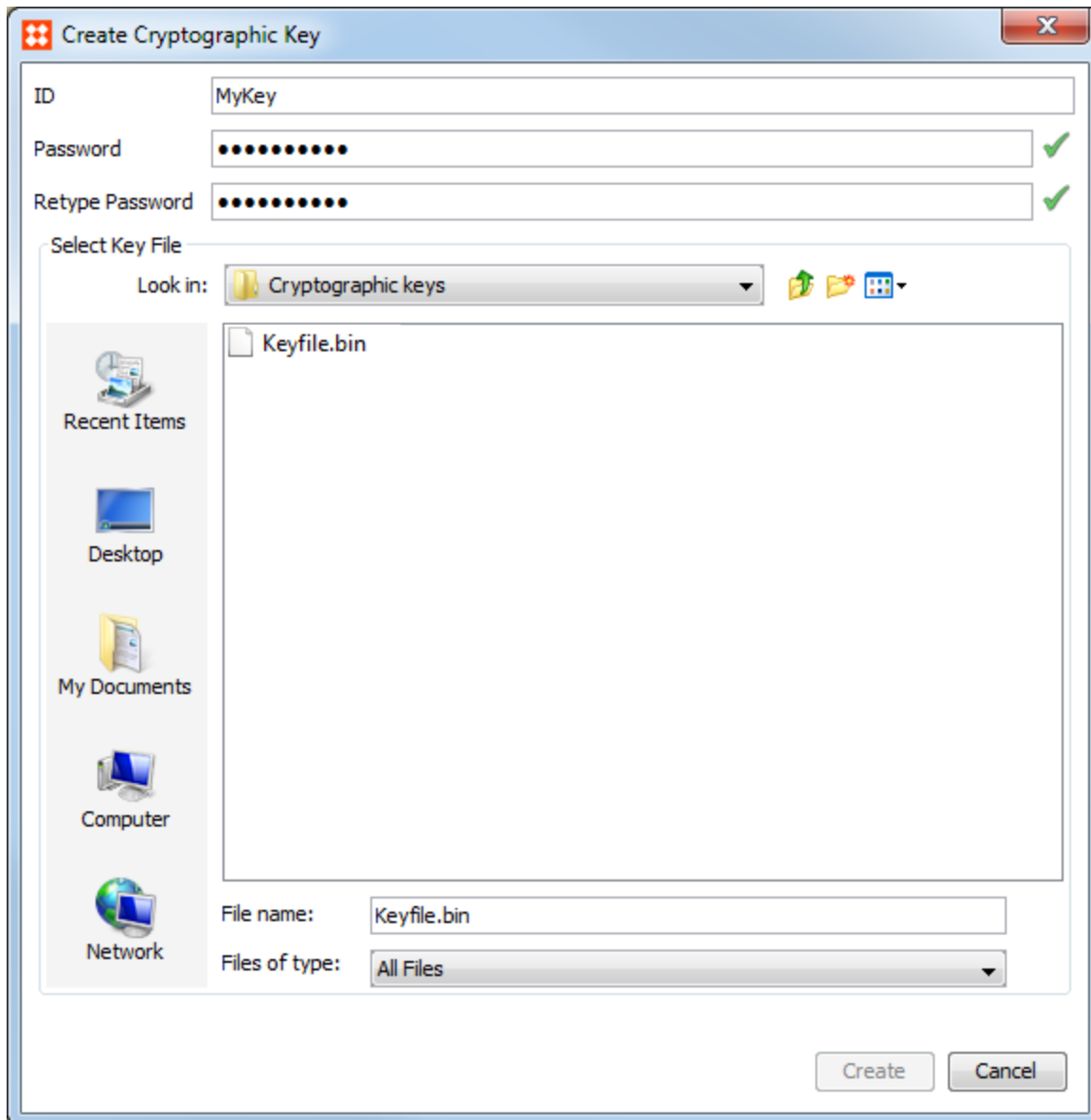
More information about attributes and creating attribute groups is found under System Setup in STEP Help.

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.

Maintaining Attributes

This section discusses how to create an attribute and then modify it based on your requirements. You will find information including setting validity, using inline references for values, filtering in the hierarchy, and converting for certain scenarios.

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.

The screenshot shows the 'System Setup' window with a tree view on the left and a detailed editor on the right. The tree view shows 'Attribute Groups' expanded, with 'Marketing Description' selected. The right pane shows the 'Marketing Description - Attribute' editor with the 'Attribute' tab active. The 'Type' field is highlighted with a red box, showing 'Description'.

Name	Value
ID	Marketing Description
Name	Marketing Description
Last edited by	2015-08-11 16:32:46 by USER
Full Text Indexable	No
Externally Maintained	No
Hierarchical Filtering	None
Calculated	Yes
Type	Description
Dimension Dependencies	Country; Language;
Value template	EMPTY
Mandatory	No
Attribute Help Text	abc
Condition	[?]
Display Name	abc
Display Sequence	123
Purpose	abc
Translate Item Data	123

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

The 'Change Attribute Type' dialog box is shown with two radio buttons: 'Description' and 'Specification'. The 'Specification' radio button is selected. There are 'Save' and 'Cancel' buttons at the bottom.

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

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

In STEP workbench, merging attributes can be done from System Setup, using the right-click menu option 'Merge Attribute Values' for any attribute. This option starts the Bulk Update 'Merge attribute' operation with the selected attribute as the source, and allows you to choose 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.

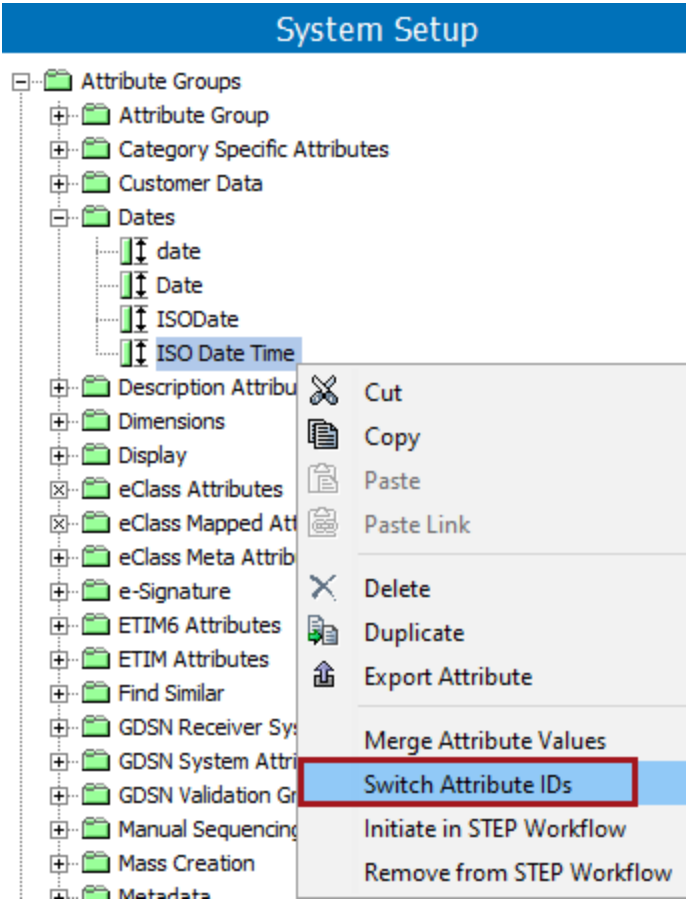
For more information about merging and the steps to merge attribute values, see **Merging Attribute Values** in the **Attributes** section of the **System Setup / STEP Super User** documentation.

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.

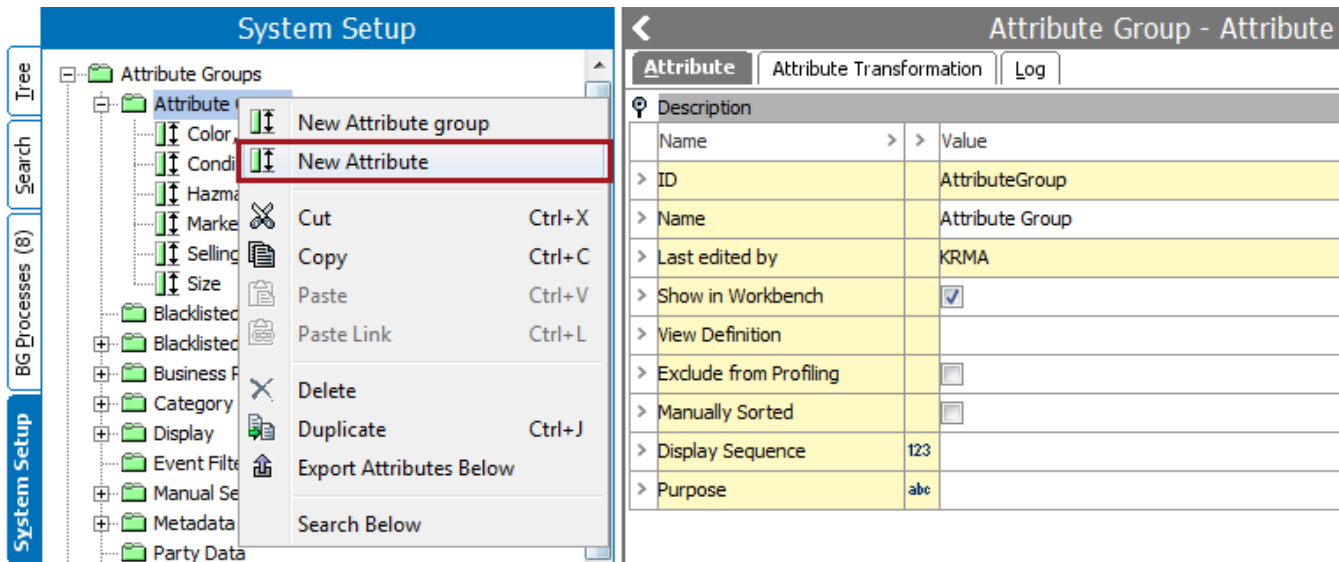


Creating Attributes

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.



The screenshot shows the 'System Setup' window with the 'Attribute Groups' tree on the left. A right-click context menu is open over the 'Attribute Groups' folder, with 'New Attribute' highlighted. The 'Attribute Group - Attribute' dialog is open on the right, showing the 'Attribute' tab. The dialog has a table with the following data:

Description	
Name	Value
ID	AttributeGroup
Name	Attribute Group
Last edited by	KRMA
Show in Workbench	☒
View Definition	
Exclude from Profiling	☐
Manually Sorted	☐
Display Sequence	123
Purpose	abc

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**.

Create Attribute

Steps

1. Select Attribute Group
- 2. Enter ID and Name**
3. Enter Validation Rule
4. Select Unit(s)
5. Valid For
6. Apply Dimension Dependencies
7. Link to Hierarchy

Enter ID and Name

ID:

Name:

Validation Base Type:

Multi Valued:

☒ Specification
☐ Description

Back Next Finish Cancel

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

Click **Next**.

Create Attribute

Steps

1. Select Attribute Group
2. Enter ID and Name
- 3. Enter Validation Rule**
4. Select Unit(s)
5. Valid For
6. Apply Dimension Dependencies
7. Link to Hierarchy

Enter Validation Rule

Name	Value
Input Mask	
Minimum Value	
Maximum Value	

Back Next Finish Cancel

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**.

Create Attribute

Steps

1. Select Attribute Group
2. Enter ID and Name
3. Enter Validation Rule
- 4. Select Unit(s)**
5. Valid For
6. Apply Dimension Dependencies
7. Link to Hierarchy

Select Unit(s)

- ft
- Picas

Add...

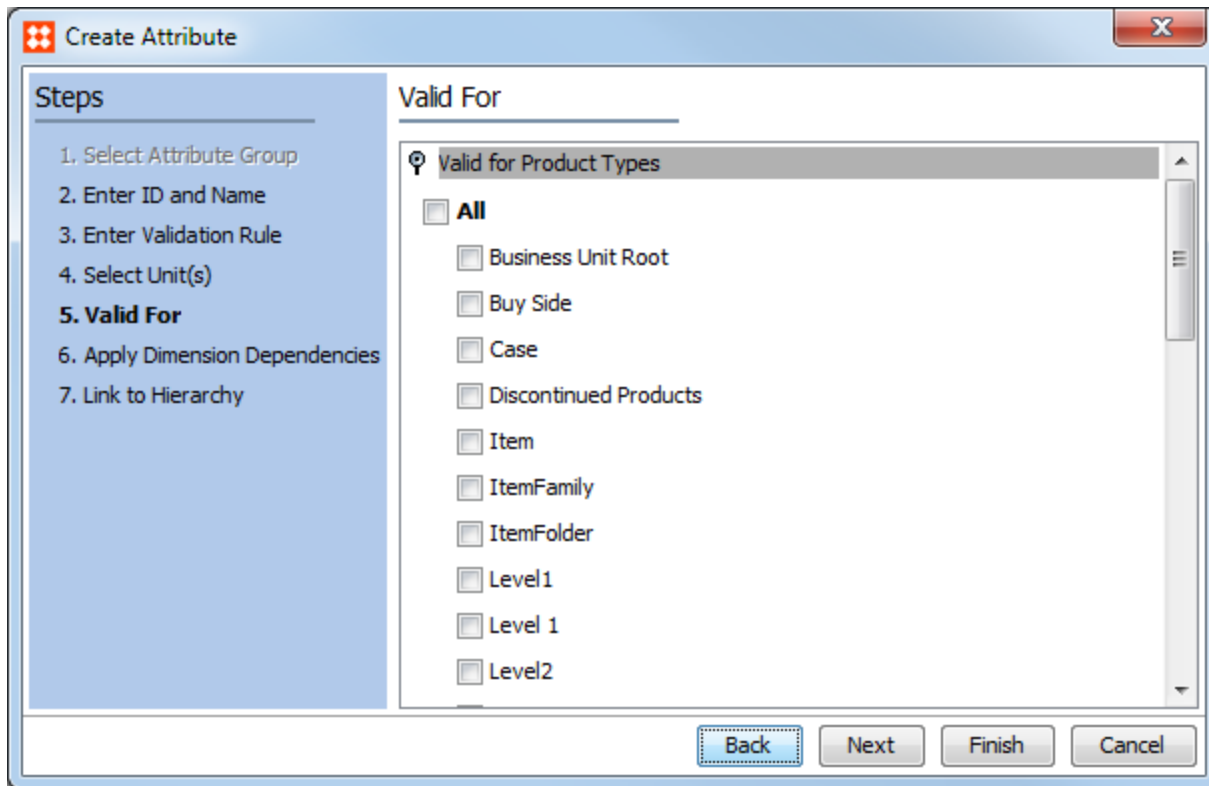
Remove

Default Unit: ft

Back Next Finish Cancel

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

Click **Next**.

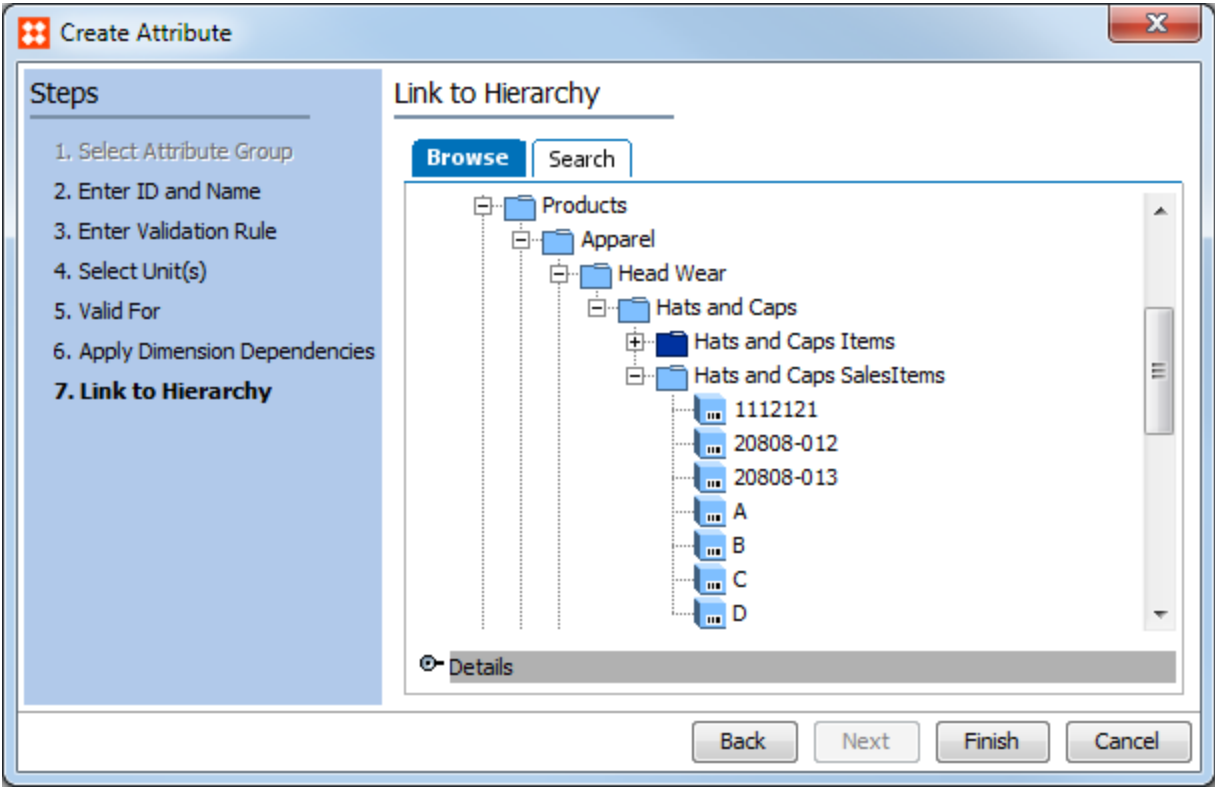


7. Apply Dimension Dependencies.

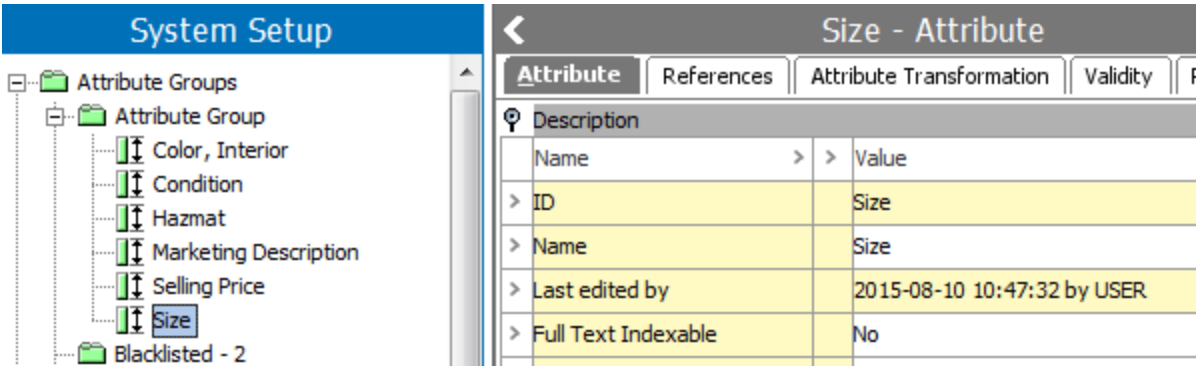
Click **Next**.

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** to create the attribute.



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

An externally maintained attribute 'lives' outside of STEP because the values are housed in an external system. A typical example is the prices that are maintained in the ERP environment and transferred to STEP for publishing and distribution or addresses maintained in an electronic medical record (EMR).

Revision Control

Externally maintained attributes are not included in revision control. This means that when a value is imported to one of the workspaces, the new value is automatically applied to all other workspaces. For example, if a supplier enters a value in the 'Staging' workspace for an externally maintained attribute, this value is automatically transferred to the other workspaces, e.g., 'Main' and 'Approved'.

If you update an existing internally maintained attribute that has values to be externally maintained, you must select the workspace that contains the values that will be kept. Typically this will be the workspace that contains the most recent updates, e.g., 'Main'.

Note: No revisions are made due to a change on an externally maintained attribute, even if another user modifies it. The data change does generate an event, however, a revision is only generated if some internally maintained attribute is changed.

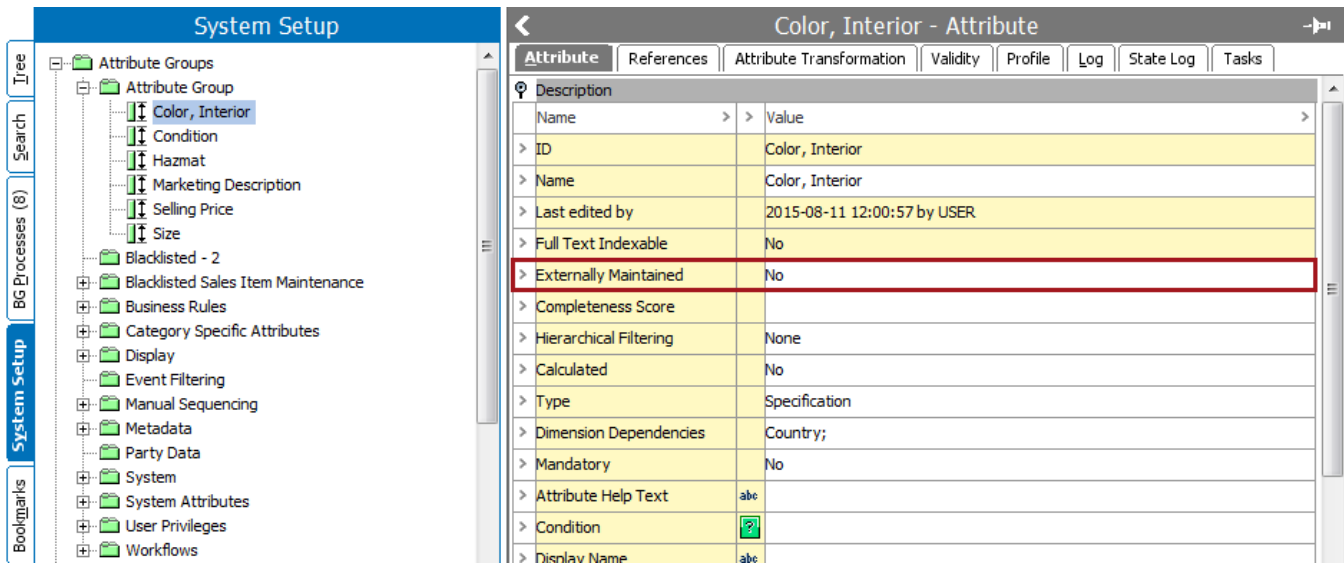
Sharing LOVs

Externally maintained attributes cannot share LOVs with internally maintained attributes, as values cannot be maintained both internally and externally.

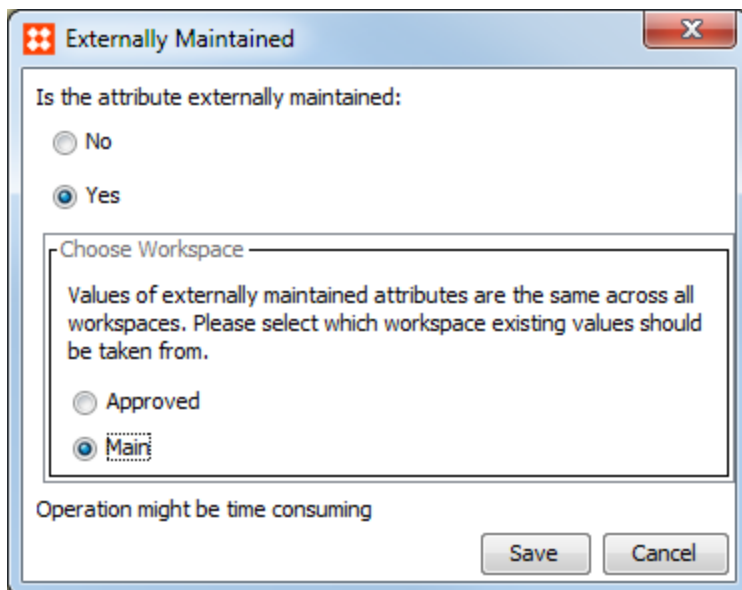
- Setting as externally maintained: If an LOV that is used by any internally maintained attributes is selected, and the attribute is then set as externally maintained, all attributes will become externally maintained.
- Setting as internally maintained: If an LOV that is used by one or more externally maintained attributes is selected, and the attribute is then set as internally maintained, all attributes will become internally maintained.

Setting as Externally Maintained

1. In System Setup, expand **Attribute Groups**, expand the relevant group, and then select the attribute you want to set as externally maintained.



- On the attribute editor > Attribute tab > Description flipper > double-click the **Externally Maintained** parameter to display the Externally Maintained dialog.



- For 'Is the attribute externally maintained:' radio button, choose **Yes** to enable the Choose Workspace parameter.
- Select the workspace that contains the existing values that will be used across all workspaces.
- Click **Save**.

Note: The number of values that already exist for the attribute affects the time required to complete this operation.

Full Text Indexable Attributes

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.

System Setup

- Attribute Groups
 - Attribute Group
 - Color, Interior
 - Condition
 - Hazmat
 - Marketing Description
 - Selling Price
 - Size
 - Blacklisted - 2
 - Blacklisted Sales Item Maintenance
 - Business Rules
 - Category Specific Attributes
 - Display
 - Event Filtering
 - Manual Sequencing
 - Metadata
 - Party Data
 - System
 - System Attributes
 - User Privileges
 - Workflows
 - Attribute Transformations
- Action Sets
- Contexts
- Lists of Values / LOVs
- Gateway Endpoint
- Global Business Rules
- Inbound Integration Endpoints
- Match Codes and Matching Algorithms
- Outbound Integration Endpoints
- Portals
- Status Flags
- Workflow Profiles
- Workflows

Marketing Description - Attribute

Attribute | References | Attribute Transformation | Validity | Profile | Log | State Log | Tasks

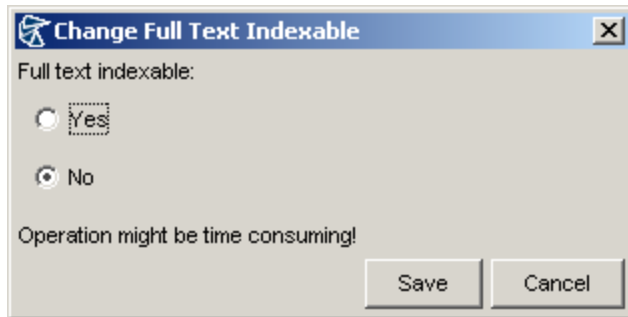
Description

Name	Value
ID	Marketing Description
Name	Marketing Description
Last edited by	2015-08-11 14:05:14 by USER
Full Text Indexable	No
Externally Maintained	No
Hierarchical Filtering	None
Calculated	Yes
Type	Description
Dimension Dependencies	Country; Language;
Value template	EMPTY
Mandatory	No
Attribute Help Text	abc
Condition	?
Display Name	abc
Display Sequence	123
Purpose	abc
Translate Item Data	123

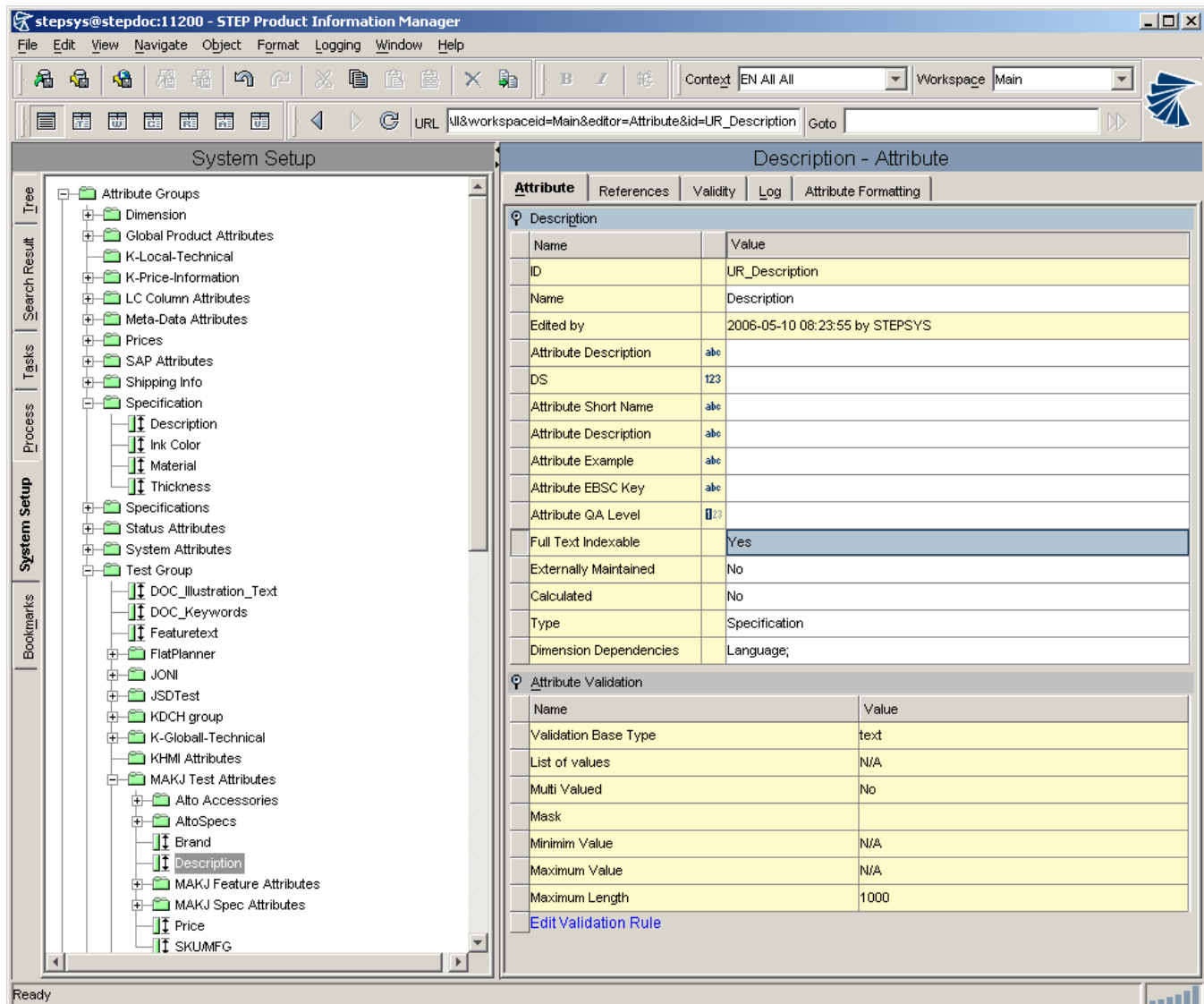
Attribute Validation

Name	Value
Validation Base Type	Text
List Of Values	N/A
Multi Valued	No
Mask	
Minimum Value	N/A

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**.



Hierarchical Filters on Attributes

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

Important: Care should be taken in using LOV filters to enforce data population as the functionality is intended to support filtered end user selection rather than globally restricted data population. Specifically, when LOV filters are used, users are prevented from manually selecting values outside of the applicable filter. However, when data is imported, the values being written are evaluated against the complete LOV rather than the filtered lists. Therefore, it is possible to import "illegal" values.

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:

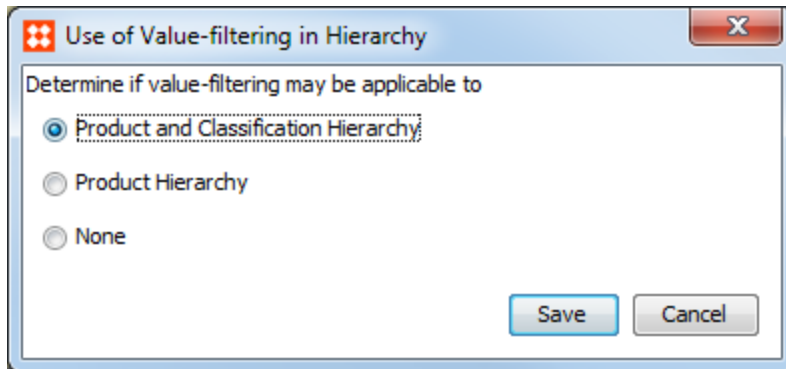
1. In [System Setup], expand **Attribute Groups**, expand the relevant Group, and then select the Attribute to be set to hierarchical filtering.

The screenshot shows the 'System Setup' window with the 'Attribute Groups' tree expanded. The 'Color, Interior' group is selected, and the 'Attribute' dialog is open. The dialog has tabs for 'Attribute', 'References', 'Attribute Transformation', 'Validity', 'Profile', 'Log', 'State Log', and 'Tasks'. The 'Attribute' tab is active, showing a table of attributes for 'Color, Interior'.

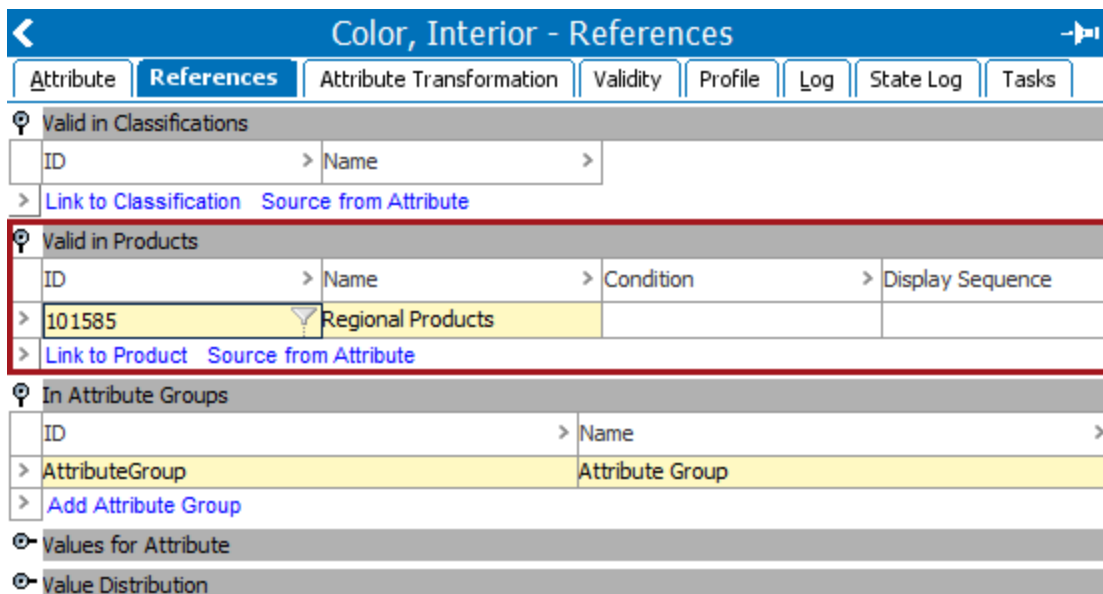
Attribute	Value
Name	Color, Interior
ID	Color, Interior
Name	Color, Interior
Last edited by	2015-08-11 12:00:57 by USER
Full Text Indexable	No
Externally Maintained	No
Completeness Score	
Hierarchical Filtering	Product and Classification Hierarchy
Calculated	No
Type	Specification
Dimension Dependencies	Country;
Mandatory	No
Attribute Help Text	abc
Condition	[?]
Display Name	abc
Display Sequence	123
Purpose	abc
Translate Item Data	123

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

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



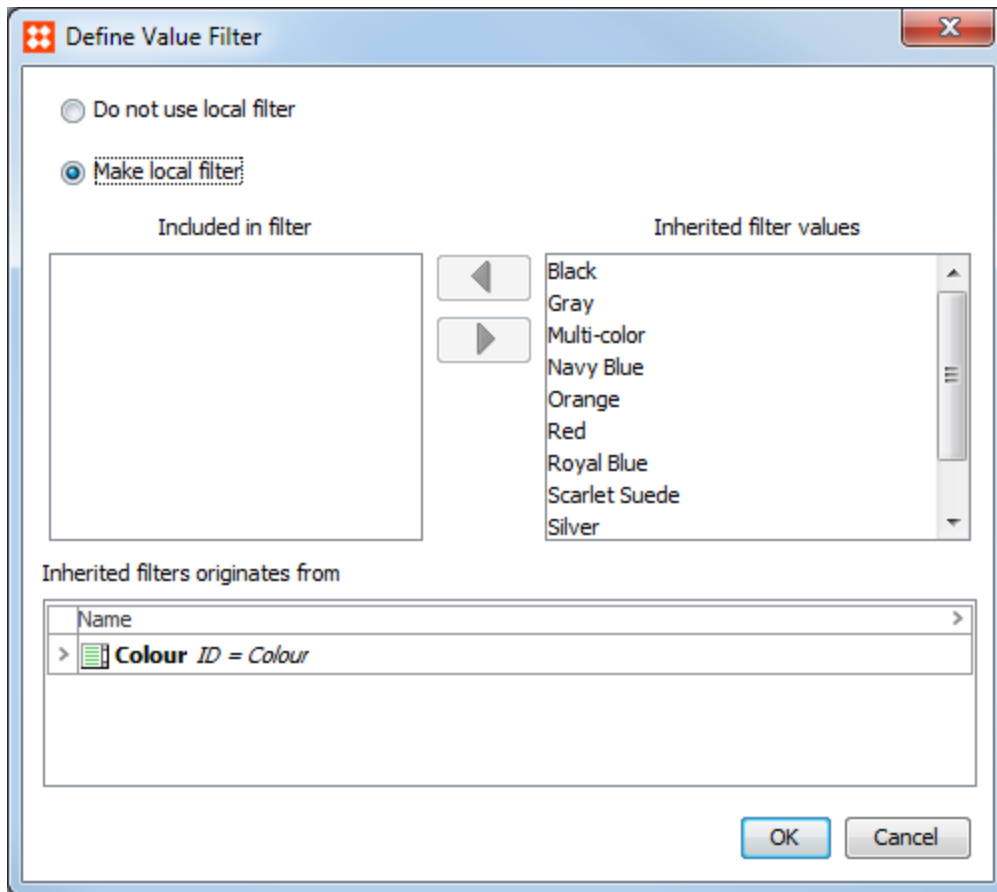
- Click **Product and Classification Hierarchy** or **Product Hierarchy**, and then click **Save**.
- Click the **References** tab.



- 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 [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.

SKU 00001 rev.0.1 - Product

Product

References

Referenced By

Images & Documents

Commercial

Tables

Status

State Log

Tasks

Description

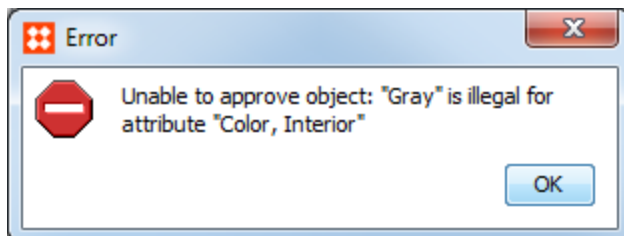
Name	>	>	Value
> ID			SKU 00001
> Name			SKU 00001
> Object Type			Product
> Revision			0.1 Last edited by USER on Tue Aug 11 12:00:43 EDT 2015
> Approved			Last Approved on Thu Aug 06 13:49:17 EDT 2015
> Translation			Not Translated
> Path			Primary Product Hierarchy/Regional Products/Apparel/Winter/Winter Coats/Ski Jacket Xtreme/S
> Marketing Description			
> Object Type			Product
> Product Variant Priority			1
> Status			Duplicate Item

No Primary Image

Attribute Group

Name	>	>	Value
> Color, Interior			Gray
> Size			Small

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



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

Validity on 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.

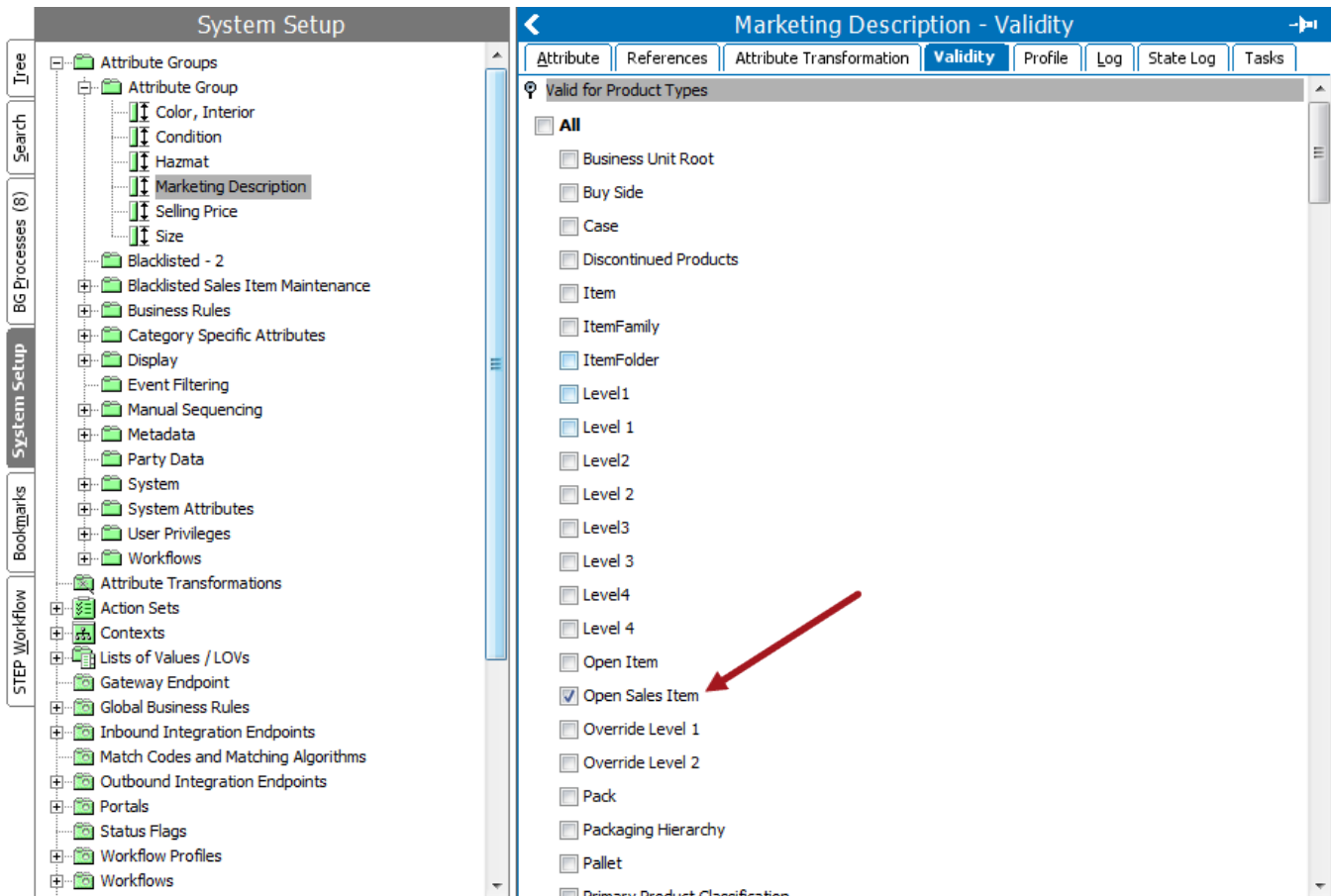
Object Type	Description
Product object types	The attribute will be valid for the selected product object types, and will appear automatically in the Description field. Note: As default, the Attribute is set to be valid for all product object types.
Classification object types	The attribute will be valid 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 setting the check boxes on / off.

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.

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.

Validity on 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.

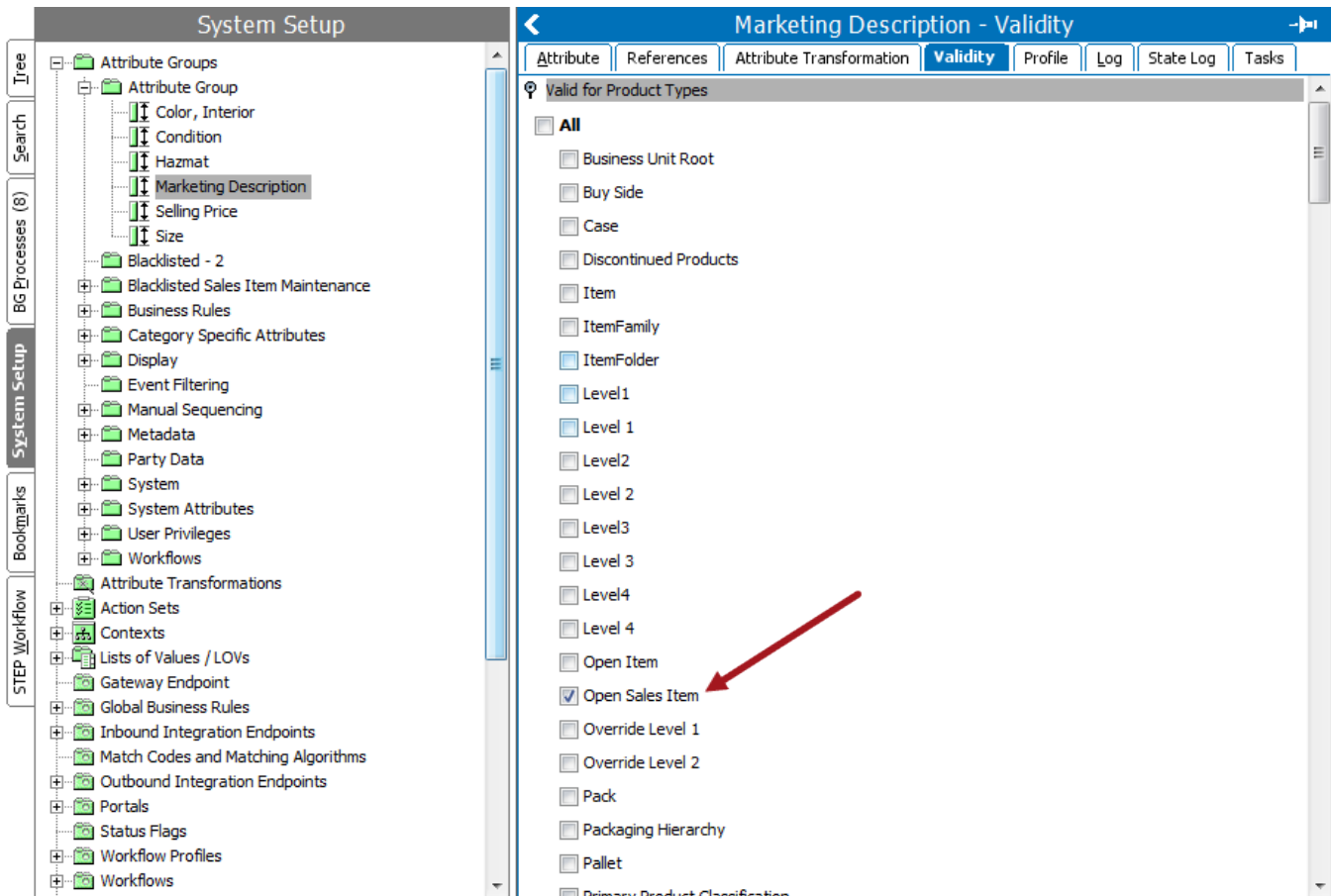
1. In [System Setup], expand **Attribute Groups**, expand the relevant Group, and then select the relevant Attribute.

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.

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

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

Reparenting Attribute Groups

To reorganize attributes under new groups, you will need to reparent the groups. There are two methods of reparenting attribute groups.

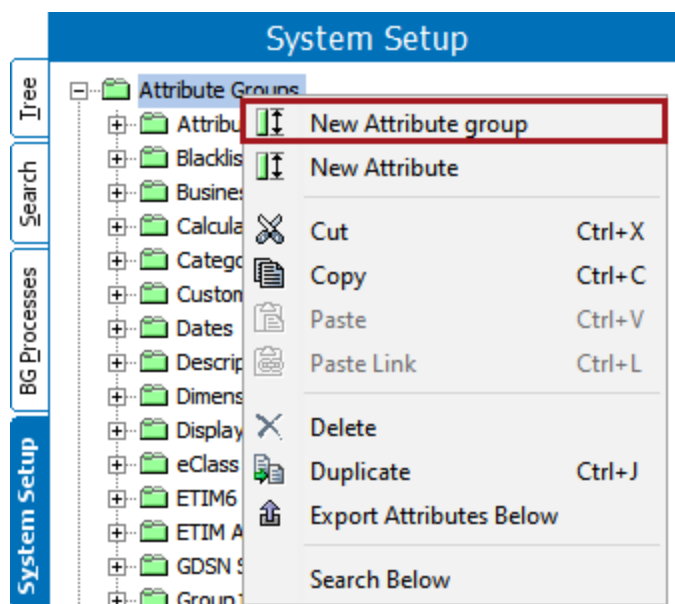
- Manual Reparenting - For smaller attribute groups with only a couple of attributes, users are able to easily drag and drop the attributes under a desired parent.
- STEPXML Reparenting - Using STEPXML, users can reparent an attribute group with numerous attributes inside.

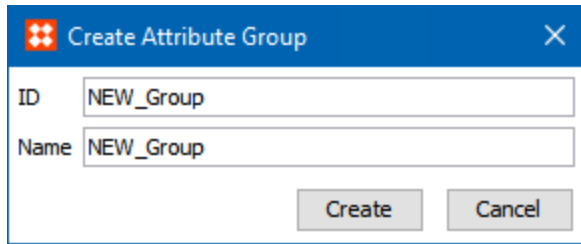
Manual Reparenting

To manually reparent attribute groups with attributes: create a new attribute group, drag the attributes from the existing attribute group into the newly created attribute group, delete the old attribute group, and recreate it inside of the new one.

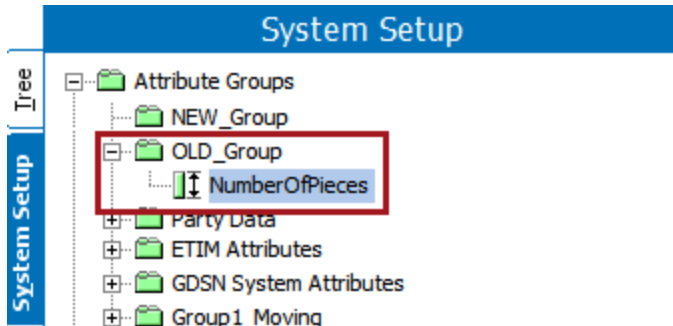
Note: If users wish to reuse the name of the existing attribute group inside of the new parent attribute, then that existing group must be deleted first. The reason for this process is because STEP will auto-generate a name based on the STEP ID, and only one instance of a STEP ID may exist.

1. In System Setup, create a new attribute group that will become the new parent for the attributes in the existing group.

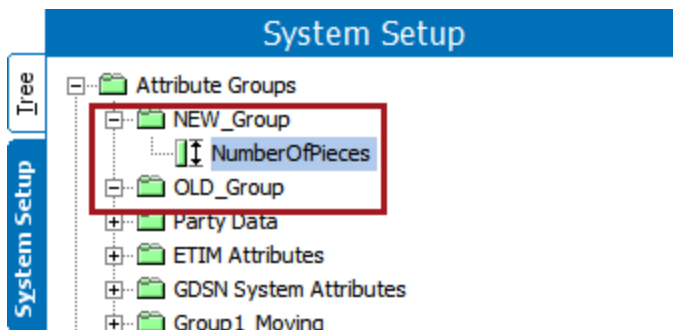




2. Open the attribute group containing the attribute to reparent by pressing the plus (+) icon next to the attribute group name.

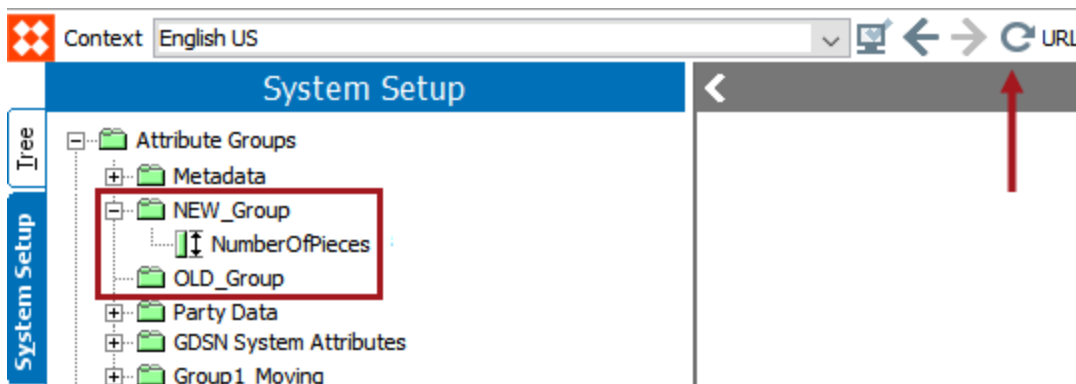


3. Drag the desired attribute(s) into the new attribute group.

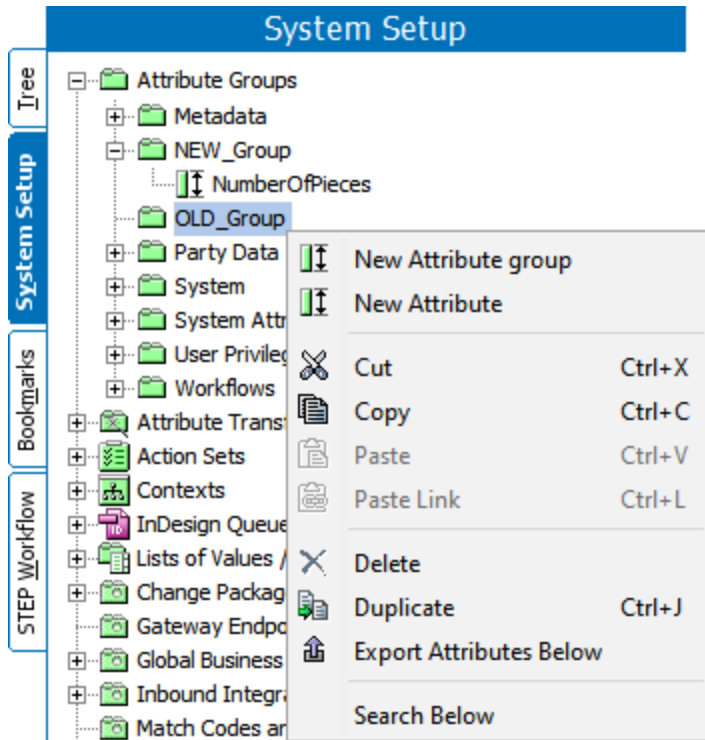


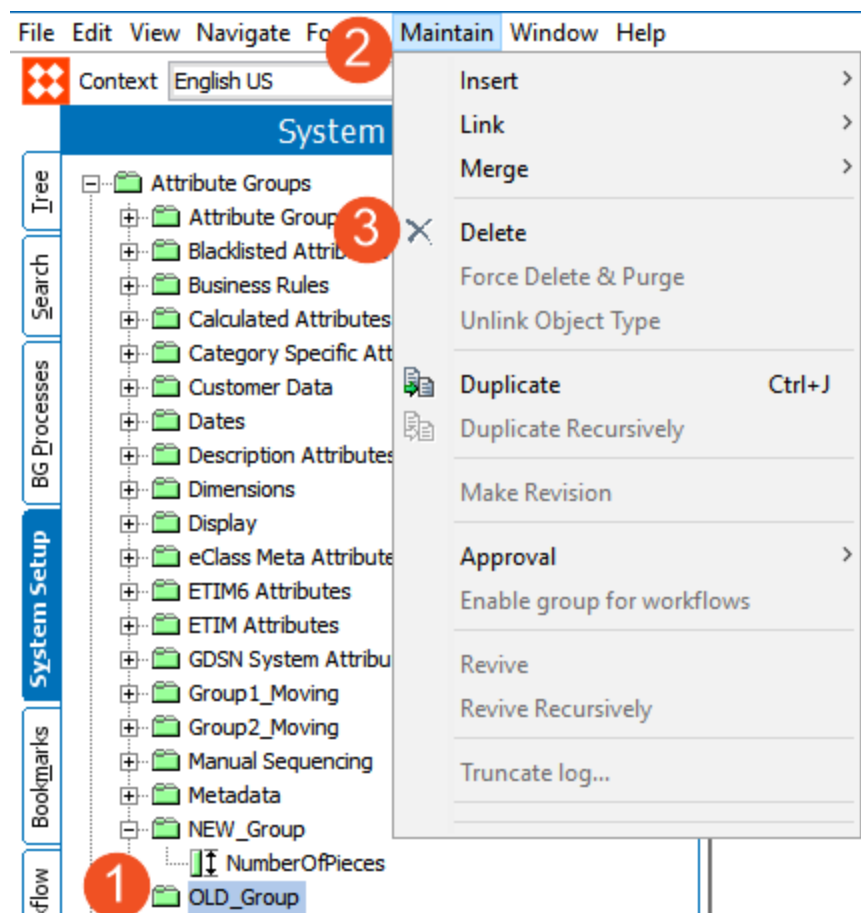
Note: When simply dragging the attribute, the old attribute group will retain the open icon, which is a minus (-) sign. This is due to the fact that the system has not been updated.

4. Select **Refresh** at the top navigation in STEP. The original attribute group refreshes to show it is empty.

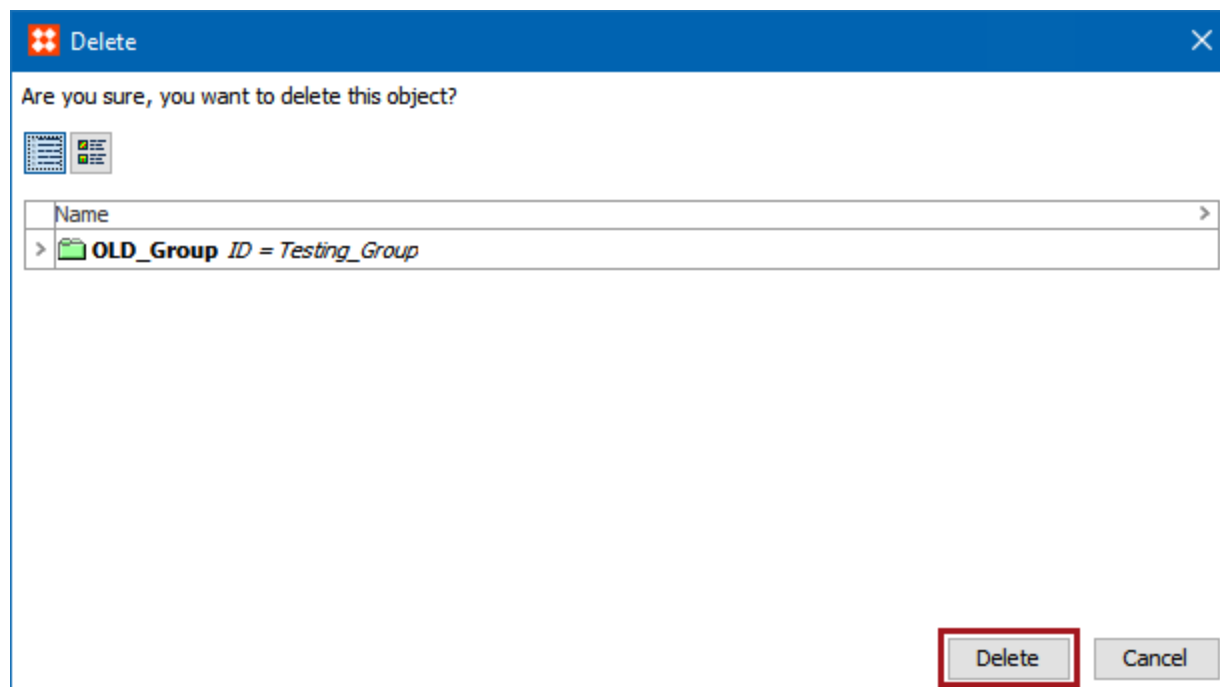


5. Delete your old group from the Tree tab by right-clicking on the attribute group and selecting **Delete**, or by going to the Maintain menu > **Delete**.

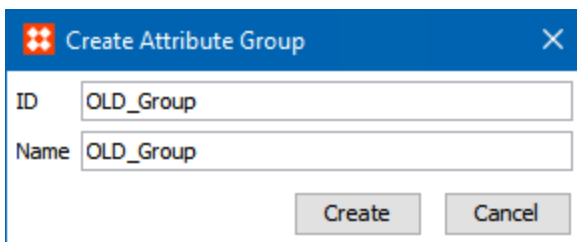
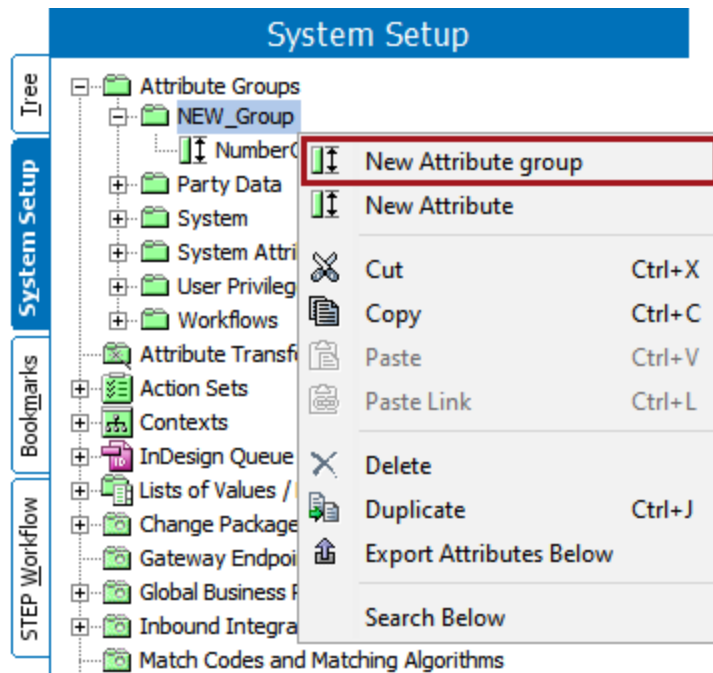




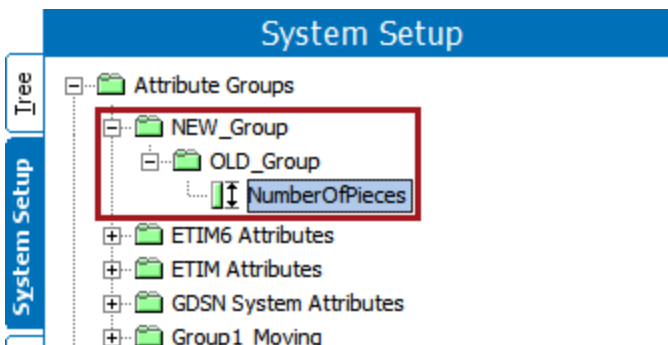
6. Click **Delete** to confirm the deletion on the pop-up prompt.



7. Create the old group under the new attribute group that was created in step 1.



8. Drag the attribute from the attribute group created in step 1 and move it to the recreated old attribute group from step 7.

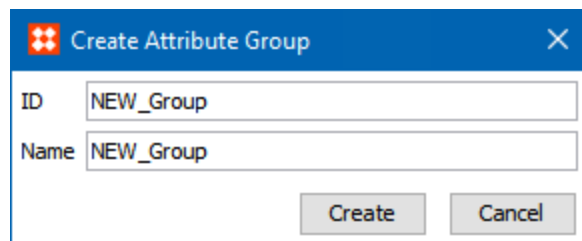
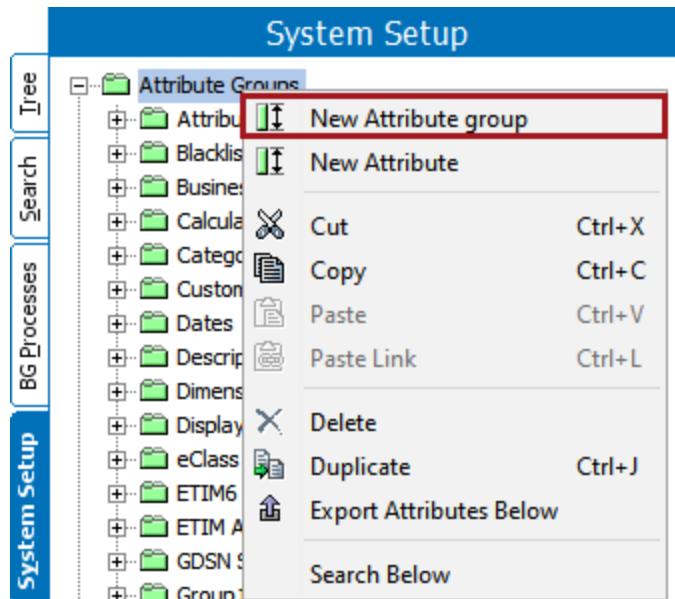


STEPXML Reparenting

An easier method for multiple attribute reparenting is to use STEP XML. In this process, users will export all the attributes to reparent, apply some STEP XML, and then import the updated attributes.

Note: Using the STEP XML method is best when dealing with a larger number of attributes. If only a few are being reparented, the manual method is fine.

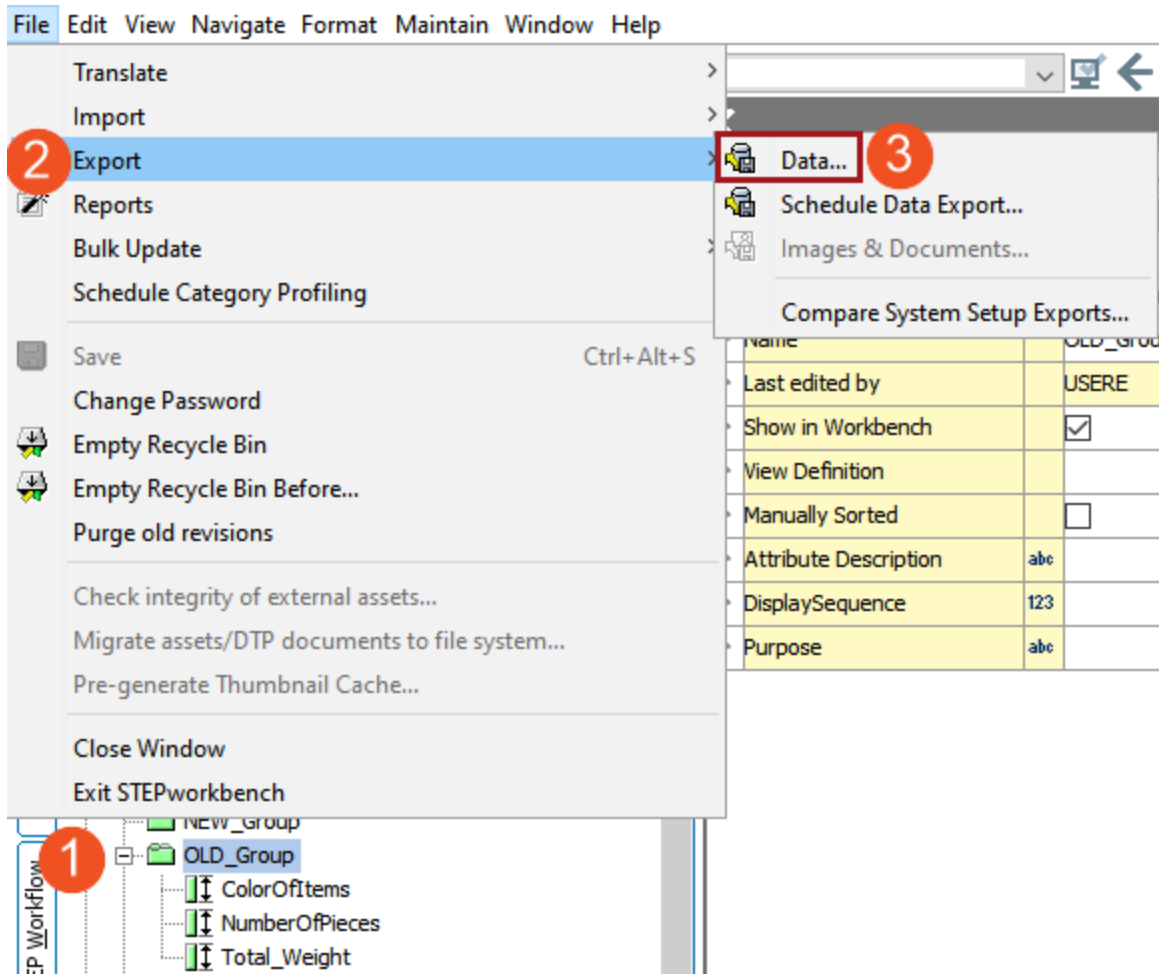
First, create the new group under which your old attribute group will be reparented.



Export the Attribute Group

Next, export the attribute group that needs reparenting. To properly export an attribute group for reparenting:

1. Select the old attribute group to reparent from the Tree tab. Then, select File > Export > **Data....**



- The Export Manager will display. Select **Next**, and on **Step 2. Select Objects**, select Attribute from the Export dropdown.

Export Manager

Steps

1. **Select Configuration**
2. Select Objects
3. Select Format
4. Map Data
5. Advanced
6. Select Delivery Method

Select Configuration

☐ Use Existing Export

☒ Create New Export

Back

Next

Finish

Cancel

Export Manager

Steps

1. Select Configuration
2. **Select Objects**
3. Select Format
4. Map Data
5. Advanced
6. Select Delivery Method

Select Objects

ID	Name	Object Type	Path
Add Objects			

☐ Only export selected objects
☐ Only export leaf objects
 Include object types <All object types>

Export: Attribute

Back

Next

Finish

Cancel

© Stibo Systems - Confidential - STEP Trailblazer 8.1

225

3. Finally, select **Add Objects** from the data field.

Export Manager

Steps

1. Select Configuration
- 2. Select Objects**
3. Select Format
4. Map Data
5. Advanced
6. Select Delivery Method

Select Objects

ID	Name	Object Type	Path
>	Add Objects		

Only export selected objects ☐

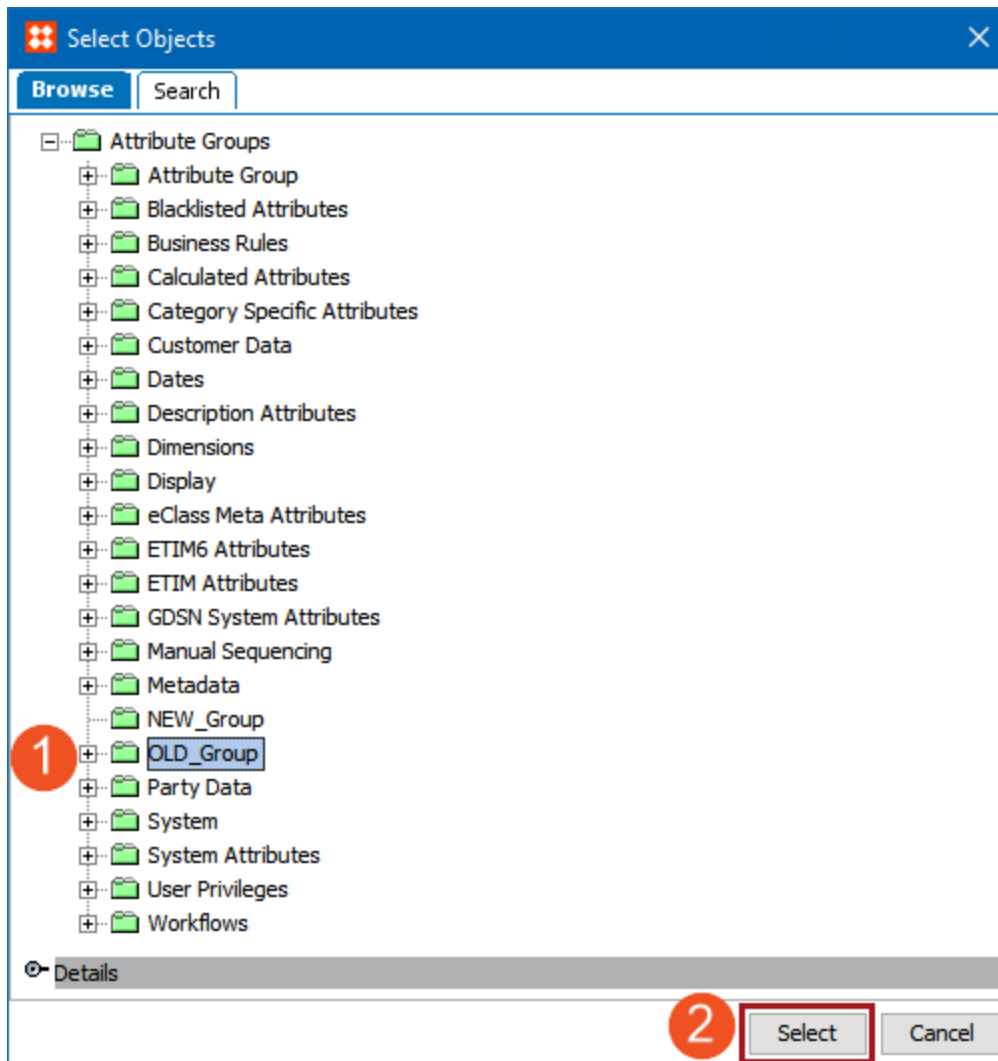
Only export leaf objects ☐

Include object types

Export: Attribute ▼

Back Next Finish Cancel

4. Select the desired attribute group to reparent by clicking the group then pressing **Select**. Select **Next** once the attribute group is added to the Select Objects list.



5. On **Step 3. Select Format**, first select STEPXML from the dropdown menu. Then, navigate to **Include Attribute Group Definitions** and **Include Attribute Defintions** and set both options to **Selected**. Select **Finish** to export the file.

Export Manager

Steps

1. Select Configuration
2. Select Objects
3. **Select Format**
4. Map Data
5. Advanced
6. Select Delivery Method

Select Format

Exports data in a STEP Product Information XML format. Note that this format ignores the leaf products only setting.

Include Validation	None
Include Type Definitions	None
Include Global Settings	None
Include Context Definitions	None
Include Unit Definitions	None
Include Collection Definitions	None
Include List Of Value Definitions	None
Include Attribute Group Definitions	Selected
Include Attribute Definitions	Selected
Include Assets	None
Include Classifications	None
Include Products	None
Include Entities	None
Include Product Attribute Values	All

Back Next **Finish** Cancel

6. Press **OK** on the export configuration pop-up. If you wish to save this configuration for future use, select the Save Options box and select a location. Otherwise, the export will begin and give a status update.

Save Export Configuration

☒ **Save Options**

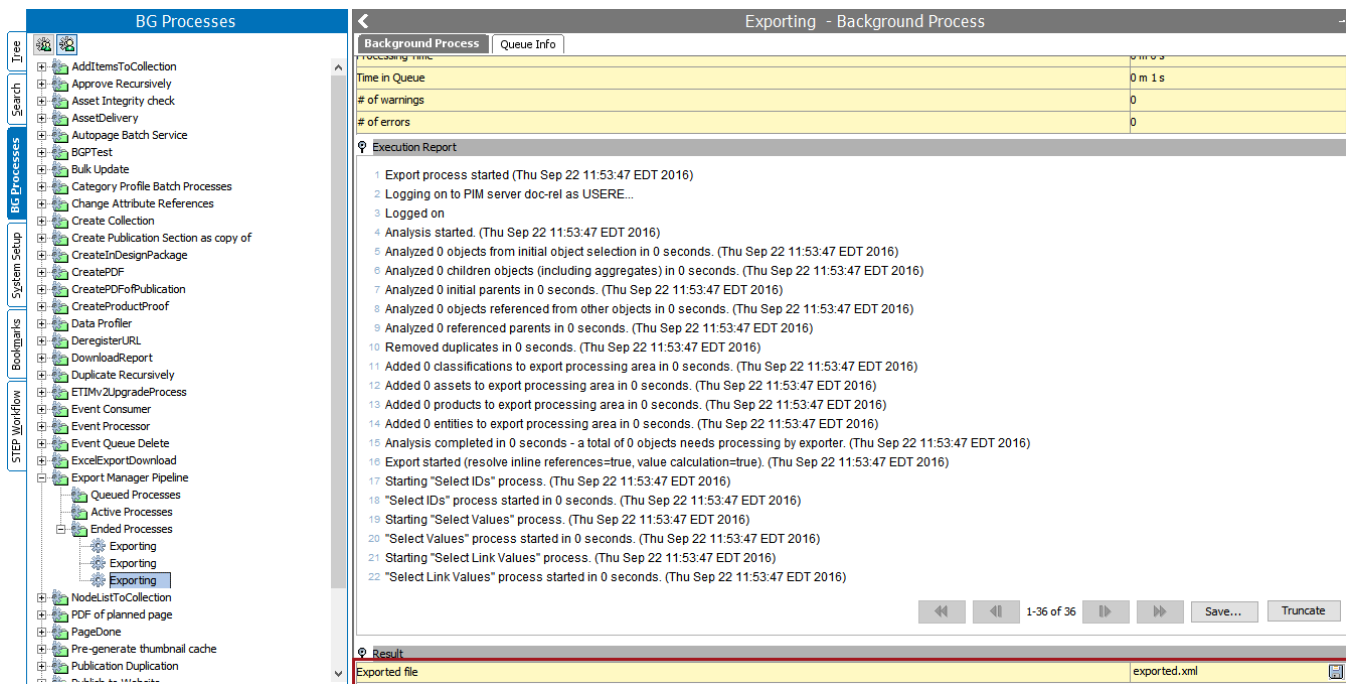
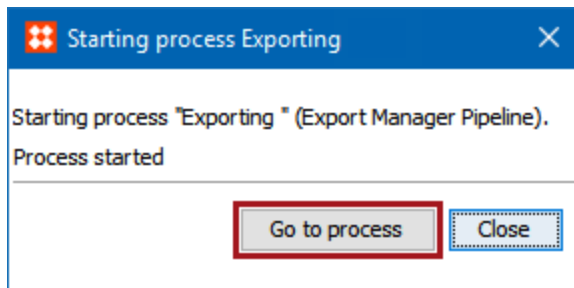
Save to: Configurations/Export Configurations/ ...

☒ **Start Export Process:** Exporting

☐ Don't export now

OK Cancel

7. Select **Go to Process** and scroll down to the exported file. Open the XML file.



Adding STEPXML

Next, STEPXML is added to the exported file. The process for adding STEPXML to reparent an attribute group is as follows:

1. After double-clicking the exported file, it will open in the default XML browser. Locate the AttributeGroup ID=" [Your old group here]" tag.

```

14 <STEP-ProductInformation ExportTime="2016-09-22 15:35:41" ExportContext="Context1" ContextID="Context1" WorkspaceID="Main" UseContextLocale="false">
15   <AttributeGroupList>
16
17     <AttributeGroup ID="OLD_Group" ShowInWorkbench="true" ManuallySorted="false" Selected="true" Referenced="true">
18       <Name>OLD_Group</Name>
19     </AttributeGroup>
20   </AttributeGroupList>
21 </STEP-ProductInformation>

```

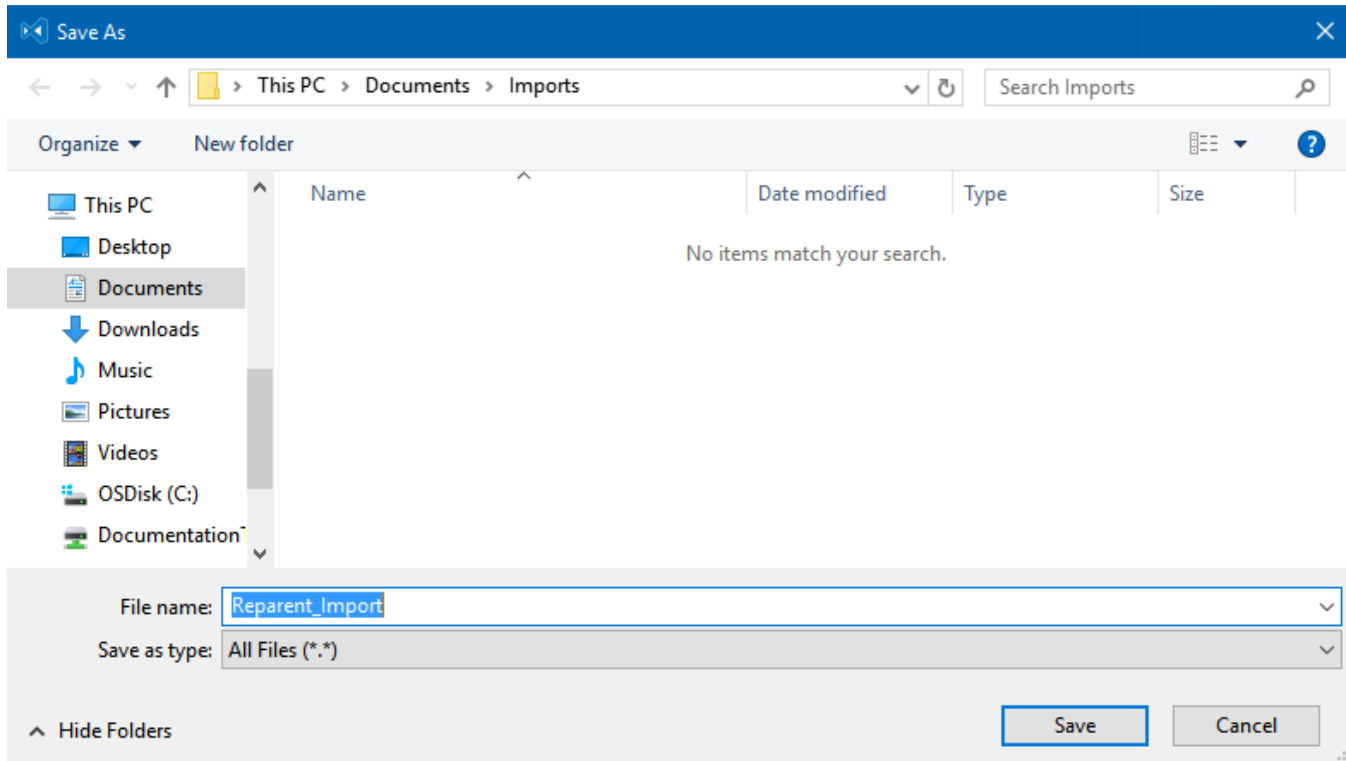
2. Inside the XML brackets for AttributeGroup ID, add ParentID="[Your new group name]". In the braces, use the ID of the parent attribute group where the old attribute group will use as a parent.

```

14 <STEP-ProductInformation ExportTime="2016-09-22 15:35:41" ExportContext="Context1" ContextID="Context1" WorkspaceID="Main" UseContextLocale="false">
15 <AttributeGroupList>
16
17 <AttributeGroup ID="OLD_Group" ShowInWorkbench="true" ManuallySorted="false" Selected="true" Referenced="true" ParentID="NEW_Group">
18 <Name>OLD_Group</Name>
19 </AttributeGroup>
20 </AttributeGroupList>
21 </STEP-ProductInformation>

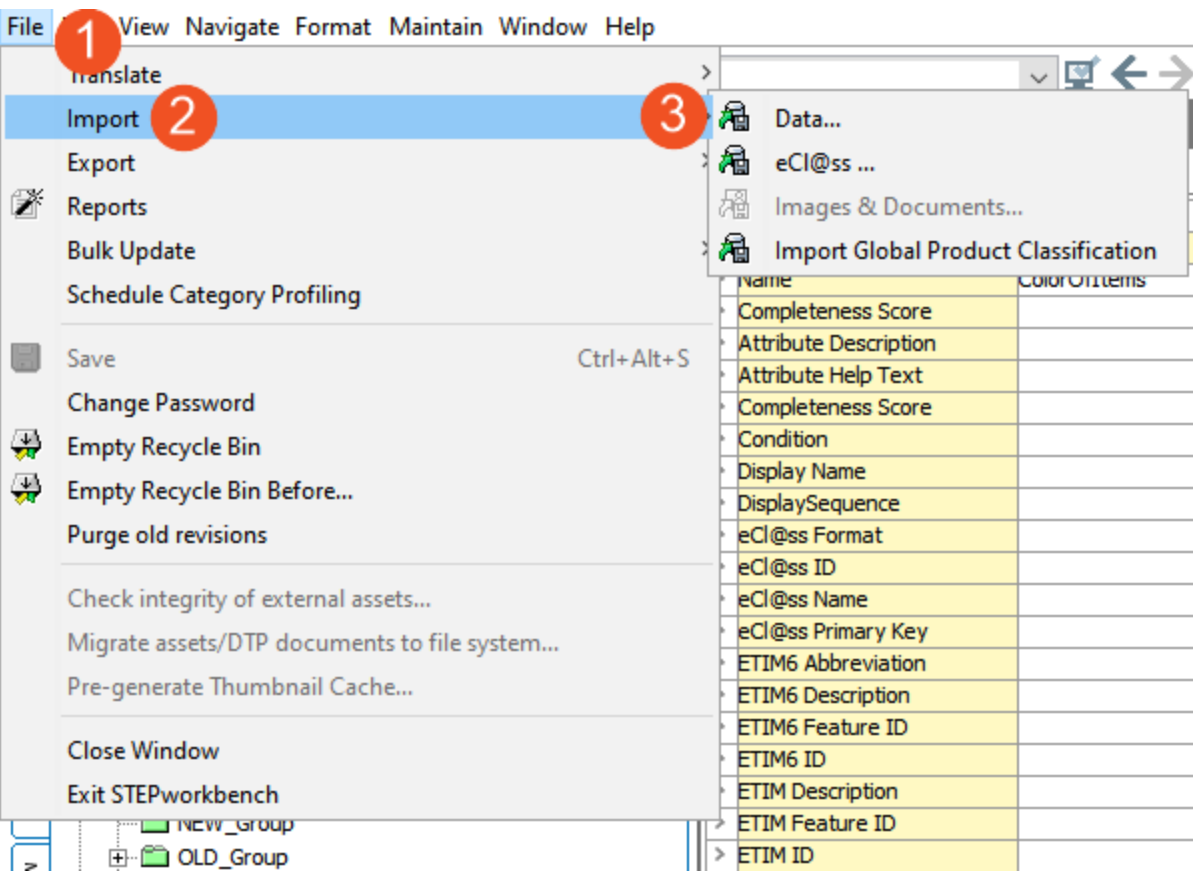
```

3. Select **Save**, and remember where the file is saved for uploading in the STEP Workbench.



Upload the Modified File

1. In the STEP Workbench, select File > Import > **Data...**



2. The Import Manager will load. Select **Next**.

The screenshot shows the 'Import Manager' dialog box with the title bar 'Import Manager' and a close button. The 'Steps' pane on the left lists eight steps, with '1. Select Configuration' highlighted. The main area is titled 'Select Configuration' and contains two radio buttons: 'Use Existing Import' (unselected) and 'Create New Import' (selected). A text input field is positioned to the right of the 'Use Existing Import' radio button. At the bottom right, there are four buttons: 'Back', 'Next' (highlighted with a red rectangle), 'Finish', and 'Cancel'.

3. On **Step 2. Select Data Source**, select the previously modified STEPXML file from the last step.

Import Manager

Steps

1. Select Configuration

2. Select Data Source

3. Select Format

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Select Data Source

Data Source

File

Retrieves contents from files reachable from your local computer.

Filename

/InputFile.txt

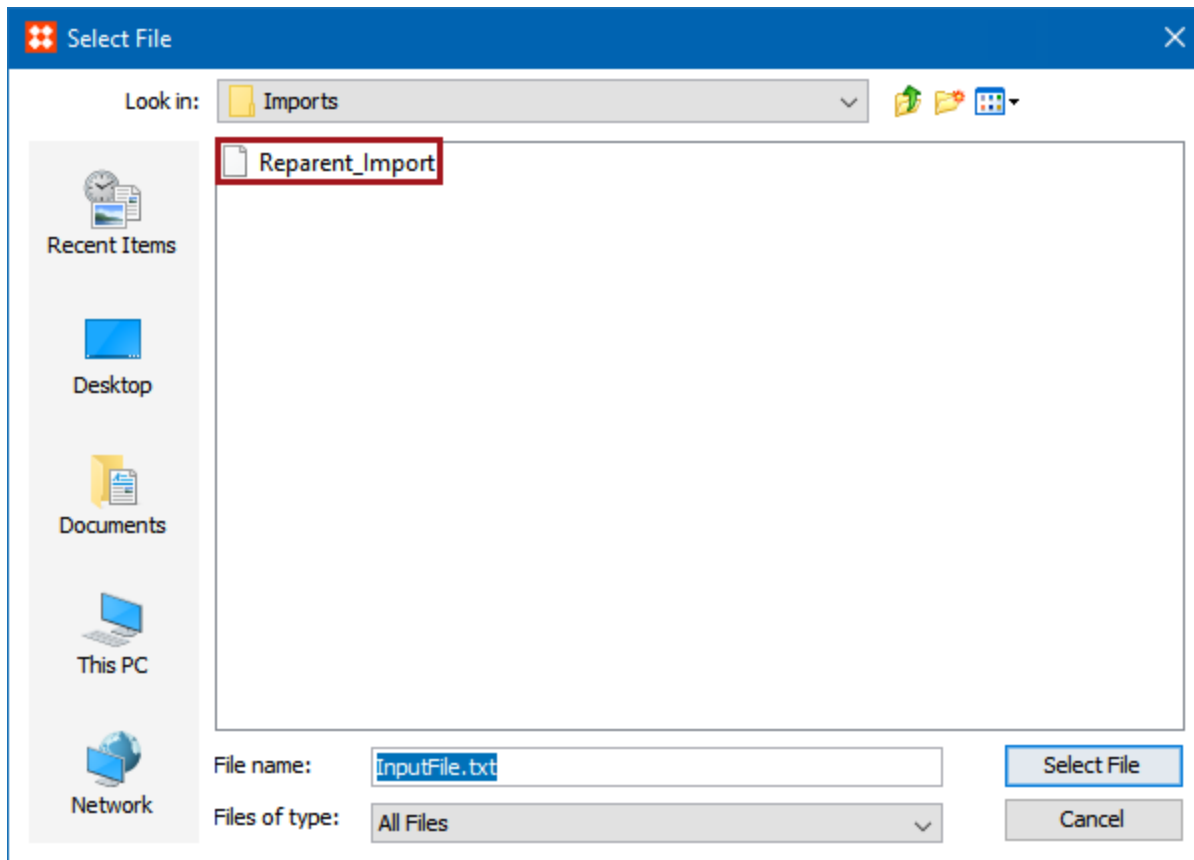
...

Back

Next

Finish

Cancel



4. Select **Finish**. A Save Import Configuration dialog will pop-up. Select **OK** to import the STEPXML script.

Import Manager

Steps

1. Select Configuration
- 2. Select Data Source**
3. Select Format
4. Map Data
5. Identify Objects
6. Identify Destination
7. Select Business Rules
8. Advanced Settings

Select Data Source

Data Source: File

Retrieves contents from files reachable from your local computer.

Filename: C:\Users\Documents\Imports\Reparent_Import

Buttons: Back, Next, **Finish**, Cancel

Save Import Configuration

☐ Save Options

Save to: Assets/

☒ **Start Import Process:** Importing

☐ Import now

☐ Don't import now

Buttons: **OK**, Cancel

5. Refresh your STEP Workbench. The old attribute group will be a child under the new attribute group.

Context English US

System Setup

Tree

- Attribute Groups
 - Attribute Group
 - Blacklisted Attributes
 - Business Rules
 - Calculated Attributes
 - Category Specific Attributes
 - Customer Data
 - Dates
 - Description Attributes
 - Dimensions
 - Display
 - eClass Meta Attributes
 - ETIM6 Attributes
 - ETIM Attributes
 - GDSN System Attributes
 - Group1_Moving
 - Group2_Moving
 - Manual Sequencing
 - Metadata
 - NEW_Group
 - OLD_Group
 - Party Data
 - System
 - System Attributes
 - User Privileges
 - Workflows

Search

BG Processes (2)

System Setup

Bookmarks

STEP Workflow

Background Process Queue Info

Processing time

Time in Queue

of warnings

of errors

Execution Report

- 1 Export process started (Thu Sep 22 15:35:41 EDT 2016)
- 2 Logging on to PIM server doc-rel as USER...
- 3 Logged on
- 4 Analysis started. (Thu Sep 22 15:35:41 EDT 2016)
- 5 Analyzed 0 objects from initial object selection in 0 seconds
- 6 Analyzed 0 children objects (including aggregates) in 0 seconds
- 7 Analyzed 0 initial parents in 0 seconds. (Thu Sep 22 15:35:41 EDT 2016)
- 8 Analyzed 0 objects referenced from other objects in 0 seconds
- 9 Analyzed 0 referenced parents in 0 seconds. (Thu Sep 22 15:35:41 EDT 2016)
- 10 Removed duplicates in 0 seconds. (Thu Sep 22 15:35:41 EDT 2016)
- 11 Added 0 classifications to export processing area in 0 seconds
- 12 Added 0 assets to export processing area in 0 seconds
- 13 Added 0 products to export processing area in 0 seconds
- 14 Added 0 entities to export processing area in 0 seconds
- 15 Analysis completed in 0 seconds - a total of 0 objects
- 16 Export started (resolve inline references=true, value count=0)
- 17 Starting "Select IDs" process. (Thu Sep 22 15:35:41 EDT 2016)

Tree

Search

BG Processes (2)

System Setup

Bookmarks

STEP Workflow

System Setup

Attribute Groups

- Attribute Group
- Blacklisted Attributes
- Business Rules
- Calculated Attributes
- Category Specific Attributes
- Customer Data
- Dates
- Description Attributes
- Dimensions
- Display
- eClass Meta Attributes
- ETIM6 Attributes
- ETIM Attributes
- GDSN System Attributes
- Group1_Moving
- Group2_Moving
- Manual Sequencing
- Metadata

NEW_Group

- OLD_Group
 - ColorOfItems
 - NumberOfPieces
 - Total_Weight

Party Data

System

System Attributes

User Privileges

Workflows

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:

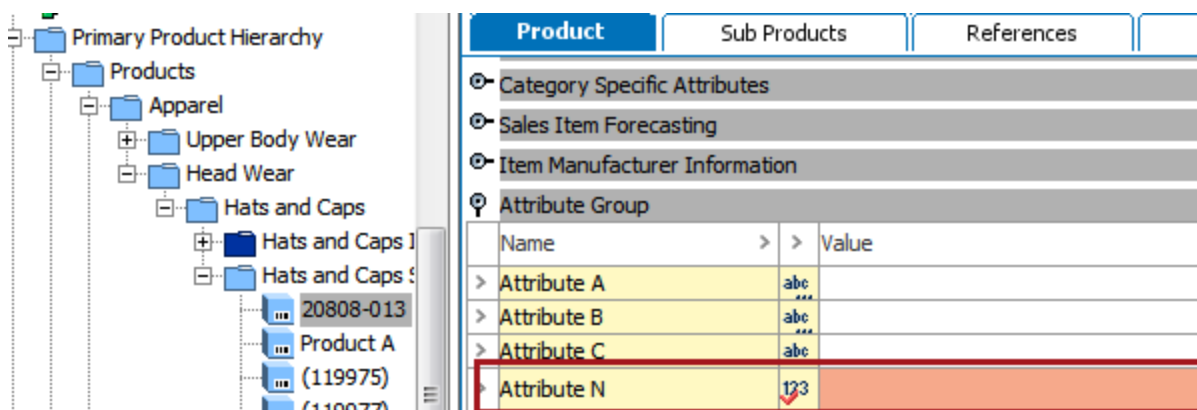
- The options for making attributes mandatory
- Detailed configuration and end user information for the various settings, or links to other topics that provide this information
- Considerations for how the different mandatory options behave across the varying interfaces (workbench, Web UI, and Smartsheets)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



Product		Sub Products	References
Category Specific Attributes			
Sales Item Forecasting			
Item Manufacturer Information			
Attribute Group			
Name	>	>	Value
> Attribute A		abc	
> Attribute B		abc	
> Attribute C		abc	
> Attribute N		123	

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**.

The screenshot shows the 'System Setup' window. On the left, a tree view under 'Attribute Groups' shows 'Attribute N' selected. On the right, the 'Attribute' tab for 'Attribute N' is active, displaying a table of settings. The 'Mandatory' setting is highlighted with a red box and set to 'Yes'.

Description	
Name	Value
ID	Attribute N
Name	Attribute N
Last edited by	2016-07-28 13:19:09 by USER6
Full Text Indexable	No
Externally Maintained	No
Hierarchical Filtering	None
Calculated	No
Type	Specification
Dimension Dependencies	
Mandatory	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.

Tree

Primary Product Hierarchy

Products

Apparel

Footwear

Safety

Hardware

Displays

Furniture

Bedroom

Beds

Beds Items

Comfy Bed

Comfy Footboard

Comfy Headboard

Comfy Side Rail

Beds Sales Items

Automotive

Building Products

Electrical & Electronics

Electronic Accessories

Power

Batteries

Batteries Items

20881 AA Battery 2 Pk

20881 C Battery 2 Pad

20888 AAA Batteries

Batteries Deco Items

Batteries Recharge Items

Batteries SalesItems

Wire & Cable

Food and Beverage

Electrical & Electro

Product Sub Products References Referenced By Images & Documents Commercial Tables Category P

Add Index Word

Publications

ID Name

Link to Publication or Section

Linked Attributes from Product Hierarchy

ID	Name	Mandatory	Inherited from
AvailabilityEnd	Availability End		Products
AvailabilityStart	Availability Start		Products
Cost	Cost		Products
CostEffectiveDate	Cost Effective Date		Products
CostExpirationDate	Cost Expiration Date		Products
CountryISO	CountryISO		Primary Product Hierarchy
CountryOfOrigin	Country of Origin		Products
DescriptionWeb	Description, Web		Products
Footnote	Footnote		Products
GroundTransportationRestrictions	Ground Transportation Restrictions		Products
Hazmat	Hazmat		Primary Product Hierarchy
IsBaseUnit	Is Base Unit		Products
IsInvoiceUnit	Is Invoice Unit		Products
IsOrderableUnit	Is Orderable Unit		Products
ListPrice	List Price		Products
ManufacturersPartNumber	Manufacturer's Part Number		Products
ManufacturerWarranty	Manufacturer Warranty		Products
MaximumVoltage	Maximum Voltage	☒	

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.

Electrical & Electronics

Electronic Accessories

Power

Batteries

Batteries Items

20881 AA Battery 2 Pk

20881 C Battery 2 Pad

20888 AAA Batteries

Name	Value
CSA Listed	Yes
ETL Listed	Yes
Maximum Voltage	123
Primary Color	abc
UL Listed	Yes
Voltage	123 1.5 V

Products

Apparel

Footwear

Safety

Hand Protection

Safety Gloves

Safety Gloves Items

20726 (box of 5 pair orang

21873 (box of 5 pair multi

Name	Value
Description, Long	abc

Category Specific Attributes

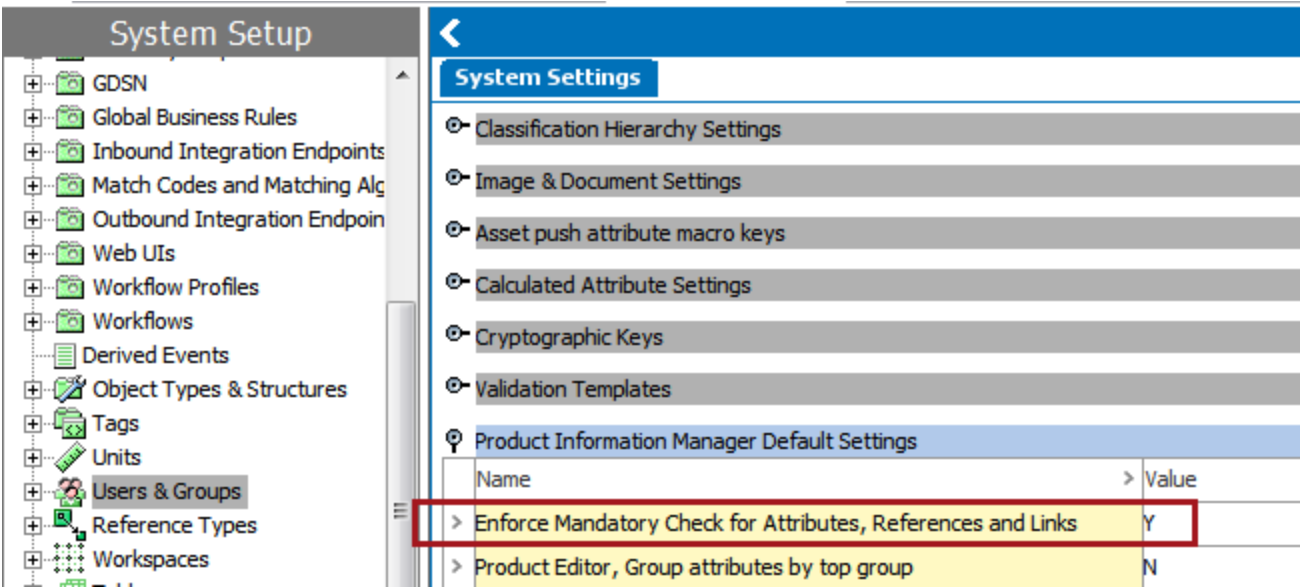
Name	Value
Material	abc Rubber
Maximum Voltage	123
Primary Color	abc

Approval Mandatory Attributes

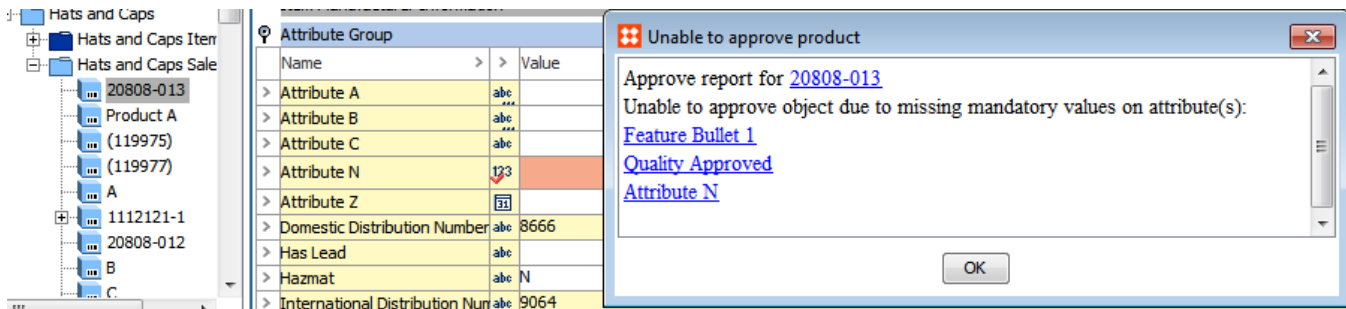
In addition to visual indicators, the system can be configured to *enforce* that the attribute is populated prior to the attribute being approved. This option is most often used only as a final "safety check" and is applied only to attributes that are required for ALL objects of a particular type or 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.

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

Serverside validation

☐ Validate only changed products
☐ Validate all product sheets
Validation result
Products:
Examined products: 3
Total errors in validation: 3
Missing required values: 3

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.

System Setup

- Attribute Groups
 - (Dates)
 - Attribute Group
 - Attribute D
 - Attribute N**
 - Color, Interior
 - Condition
 - Condition2
 - Hazmat
 - Marketing Description
 - Reference Attribute
 - Selling Price
 - Size
 - Bill of Materials
 - Multi-supplier Item
 - Blacklisted - 2
 - Blacklisted Sales Item Maintenance
 - Business Rules
 - Calculated Attributes
 - Category Specific Attributes
 - Customer Data
 - Description Attributes
 - Dimensions
 - Display
 - eClass Meta Attributes
 - ETIM6 Attributes
 - ETIM Attributes
 - Event Filtering
 - GDSN System Attributes
 - Manual Sequencing
 - Metadata
 - Asset Metadata
 - Link Metadata

Attribute N -

- Attribute
- References
- Attribute Transformation
- Validity
- Profile
- Log
- State Log
- Tasks

Description

Name	Value
ID	AttributeN
Name	Attribute N
Last edited by	2016-08-01 14:14:16 by USER6
Full Text Indexable	No
Externally Maintained	No
Completeness Score	
Hierarchical Filtering	None
Calculated	No
Type	Specification
Dimension Dependencies	
Mandatory	Yes
Cryptographic Key	<No Encryption>
Attribute Description	abc
Attribute Help Text	abc
Completeness Score	123
Condition	
Display Name	abc
DisplaySequence	123
Purpose	abc
Smartsheet Mandatory	1

Attribute Validation

Name	Value
Validation Base Type	

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

The screenshot shows the 'Export Manager' dialog box with the 'Steps' pane on the left. Step 3, 'Select Format', is active. The 'Select Format' dropdown is set to 'Excel Smartsheet'. Below this, the 'Exports hierarchical data in interactive Excel format' section contains several configuration options: 'Excel version' (Excel 2007), 'Smartsheet type' (Multiple level. Hierarchical structure flattened to sheet), 'Smartsheet usage' (Data export (for maintaining product data)), and 'Object Types' (Select a product object type for each desired level). The 'Object Types' section has three dropdowns: '1: Item (Item)', '2: Sales Item (SalesItem)', and '3:'. Below these are 'Use Cross-Context Export' (No), 'Smartsheet import configuration' (empty), and 'Mandatory metadata attribute' (Smartsheet Mandatory (SmartsheetMandatory)). The 'Mandatory metadata attribute' field is highlighted with a red box.

More information on configuring Smartsheet exports is available in the **Smartsheet Data and Template Configurations** section of 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.

Item

Basic Information and references

Category Information

Additional Information

Product Category^{fx}
Apparel | Upper Body Wear | T-shirts | T-shirts Items | Cott

Name
18210 M B

Primary Product Image


Approved
✗ Last approved 6/15/15 4:27 PM

UPC*

Product Name[✓]
Beefy-T

Short Item Description
T-shirt, short sleeve, Beefy-T, Mens M, Royal Blue

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

Add component - configure required properties

The Attribute Value Component component has required properties
Required properties (*) must be set before the component can be added to the configuration.

Attribute Value Component Properties

Component Description

The Attribute Value component is used for displaying the value of an attribute for a selected object. Used in combination with a Node Details screen.

Attribute*

UPC

Enable STEP Tags

☐

Enable Tag Conversion

☒

Label

Label Css Class

Mandatory

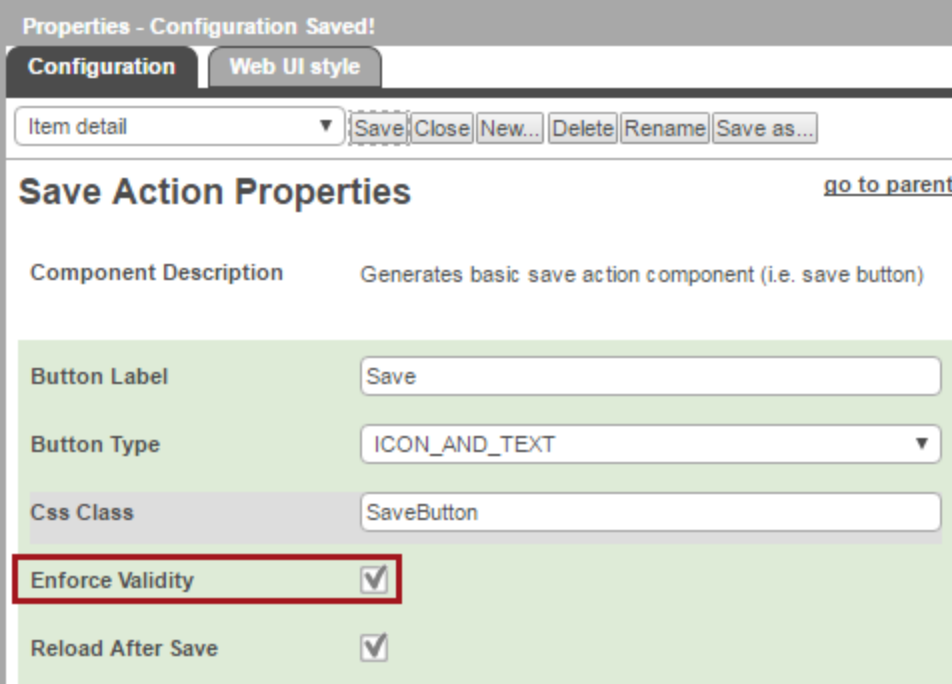
☒

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.

Merging Attribute Values

Redundant attributes have values that may technically be different, but that are logically tracking the same data and are valid for the same set of objects. Merging redundant attribute values improves data quality by establishing a distinct set of attribute values. The bulk update 'Merge Attribute Values' operation allows you to copy values from one attribute to another, so that ultimately you can delete a redundant attribute.

The 'Merge Attribute Values' operation works on two attributes:

- The source attribute is the redundant attribute that will be deleted after the merge.
- The target attribute is the preferred attribute and will be retained after the merge.

For example, consider a single object that includes the following two attributes and values to represent a clothing size.

Products				
Products References Referenced By				
View: SizeAttrs				
ID	Name	T-shirt Size	Size	
18210	18210 M B	S	small	

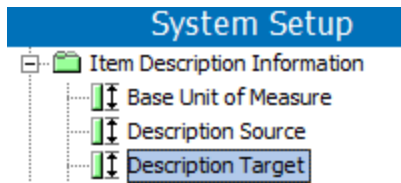
Assume that you want to keep the 'Size' attribute, but want to use the value currently stored in the 'T-shirt Size' attribute. Using the 'Merge Attribute Values' operation with 'T-shirt Size' as the source and 'Size' as the target, the preferred value ('S' for this object) is copied to the preferred attribute ('Size') for this object and for all other identified objects via a background process. Then, after the merge, the redundant attribute ('T-shirt Size') can be deleted manually while retaining all the data it previously held.

Verify Attributes to be Merged

For a successful merge, before running the bulk update, use the steps below to verify / update the target attribute and the source attribute so that:

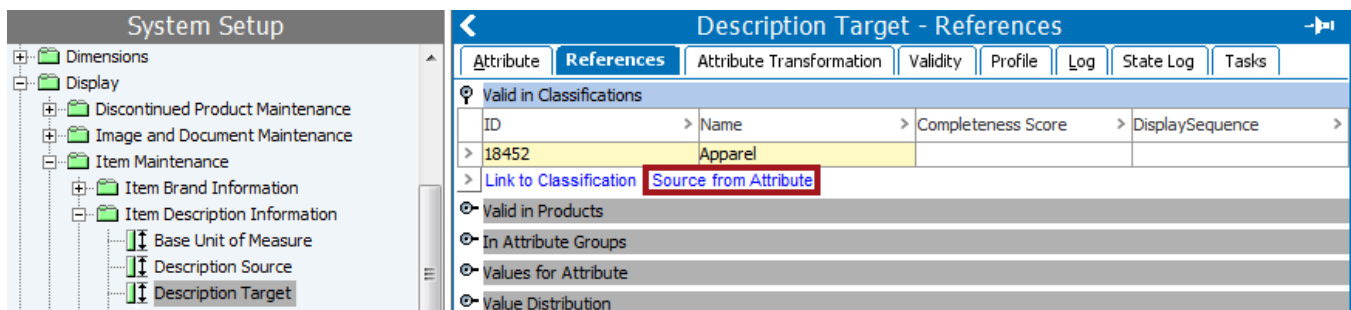
- Both have the same classification links to prevent orphaned target attributes after the merge.
- Both have the same product links to prevent orphaned target attributes after the merge.
- Both are valid for the same object types to ensure all source attribute values can be written to the target attribute.
- Both have the same or compatible Validation Base Types to prevent validation errors during the merge. For example, when an LOV attribute is the source, the values can be written to a target Text attribute, however, with a target LOV attribute, depending on the LOV setup, the allowed values may be restricted.

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

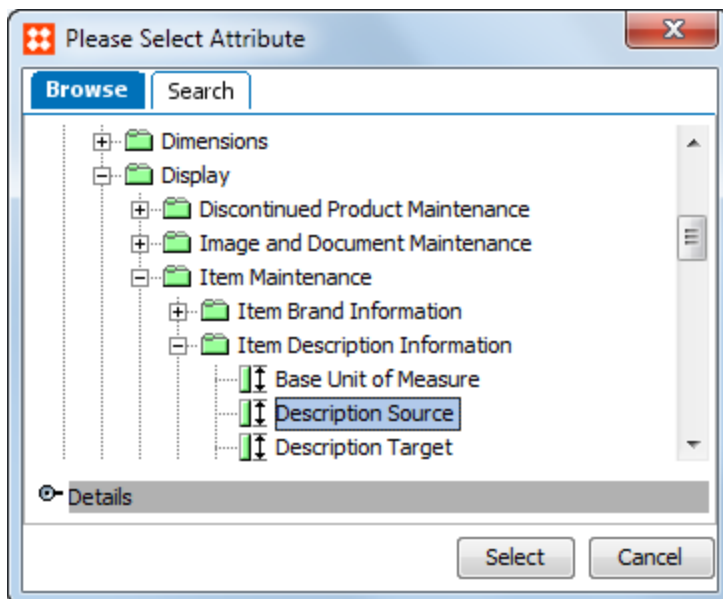


- On the References tab, if it exists, open the 'Valid in Classifications' flipper, and click the **Source from Attribute** link.

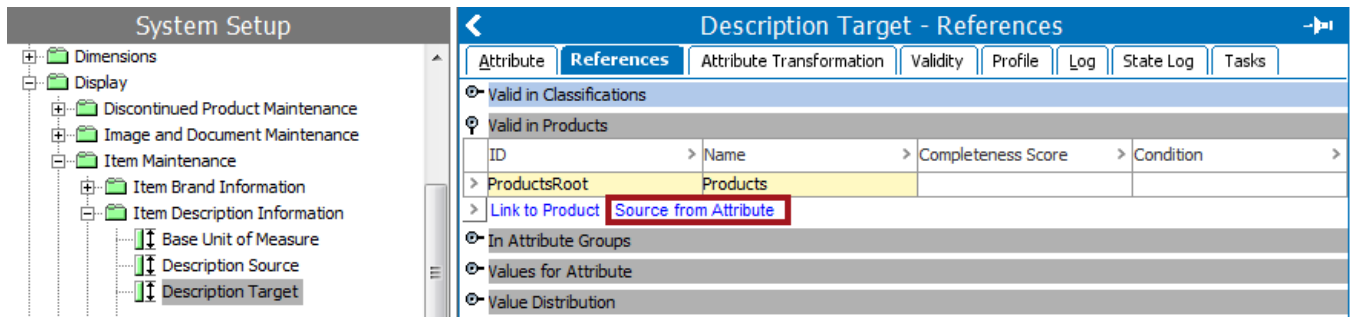
Note: If the attribute is not valid for any classifications, verify that the source attribute is also not valid for classifications.



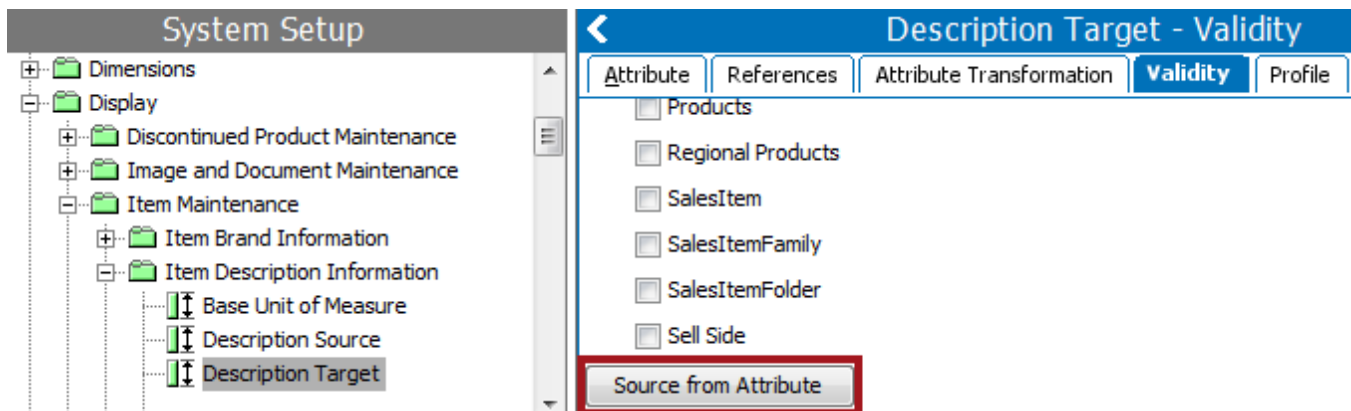
- Use Browse or Search to find the source attribute and click **Select**. The classification links of the source attribute are copied to the target attribute.



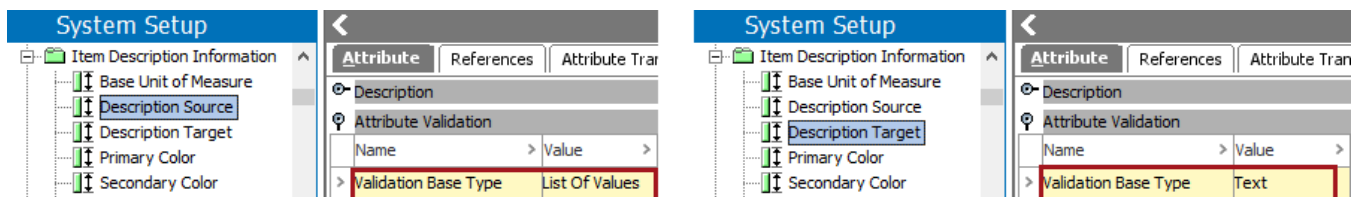
- On the References tab open the 'Valid in Products' flipper (if it exists), and click the **Source from Attribute** link.



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, under the Valid for Product Types flipper, 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, on the Attribute tab under the Attribute Validation flipper, compare the Validation Base Type for the **Target Attribute** and the **Source Attribute**.



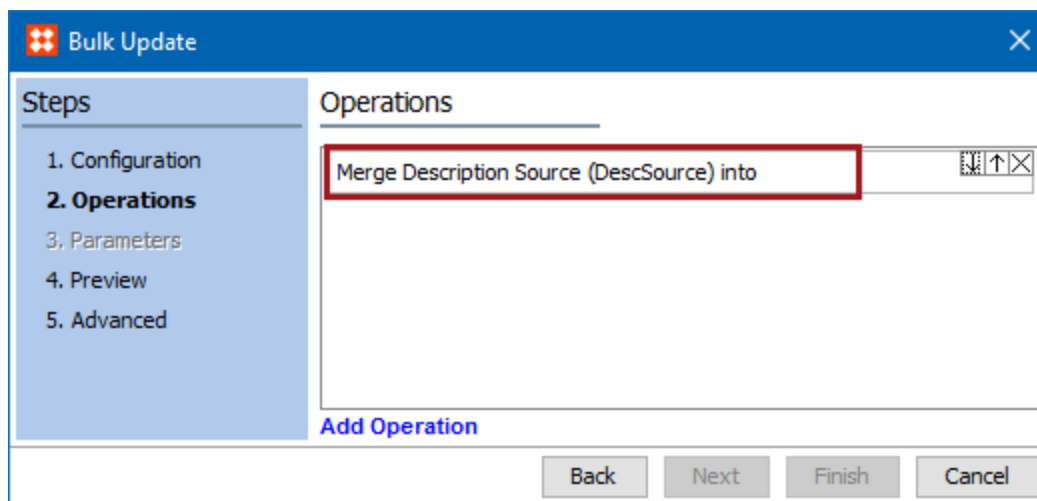
- If the Validation Base Types are the same and are not LOVs, continue to **Merge Values Using Bulk Update**.
- If the Validation Base Types are not the same, determine if they are compatible by manually adding an existing source attribute value to the target attribute successfully. If the manual test works, continue to **Merge Values Using Bulk Update**.

- If the Validation Base Types are not the same and they are not compatible, in the case of an target LOV, setting the 'Allow Users to Add Values' option to 'Yes' allows the new values to be written to the target LOV. Otherwise, see **Changing Validation Rules** section and change the validation type if possible, then continue to **Merge Values Using Bulk Update**.
- If the Validation Base Types are not the same, are not compatible, and cannot be changed, validation errors will be displayed in during the Bulk Update process. Values cannot be merged in this scenario.

Merge Values Using Bulk Update

After verifying that the data is ready to be merged, select the **Source Attribute**, right-click and choose **Merge Attribute Values**.

The Bulk Update wizard is displayed with the source attribute already selected for the Merge operation. Click the operation to supply the target attribute and display additional parameters available for the merge.



For details about the bulk update parameters, see **Attribute Values Operations for Bulk Updates** section of the **Getting Started / STEP User Guide** 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.

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.

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.

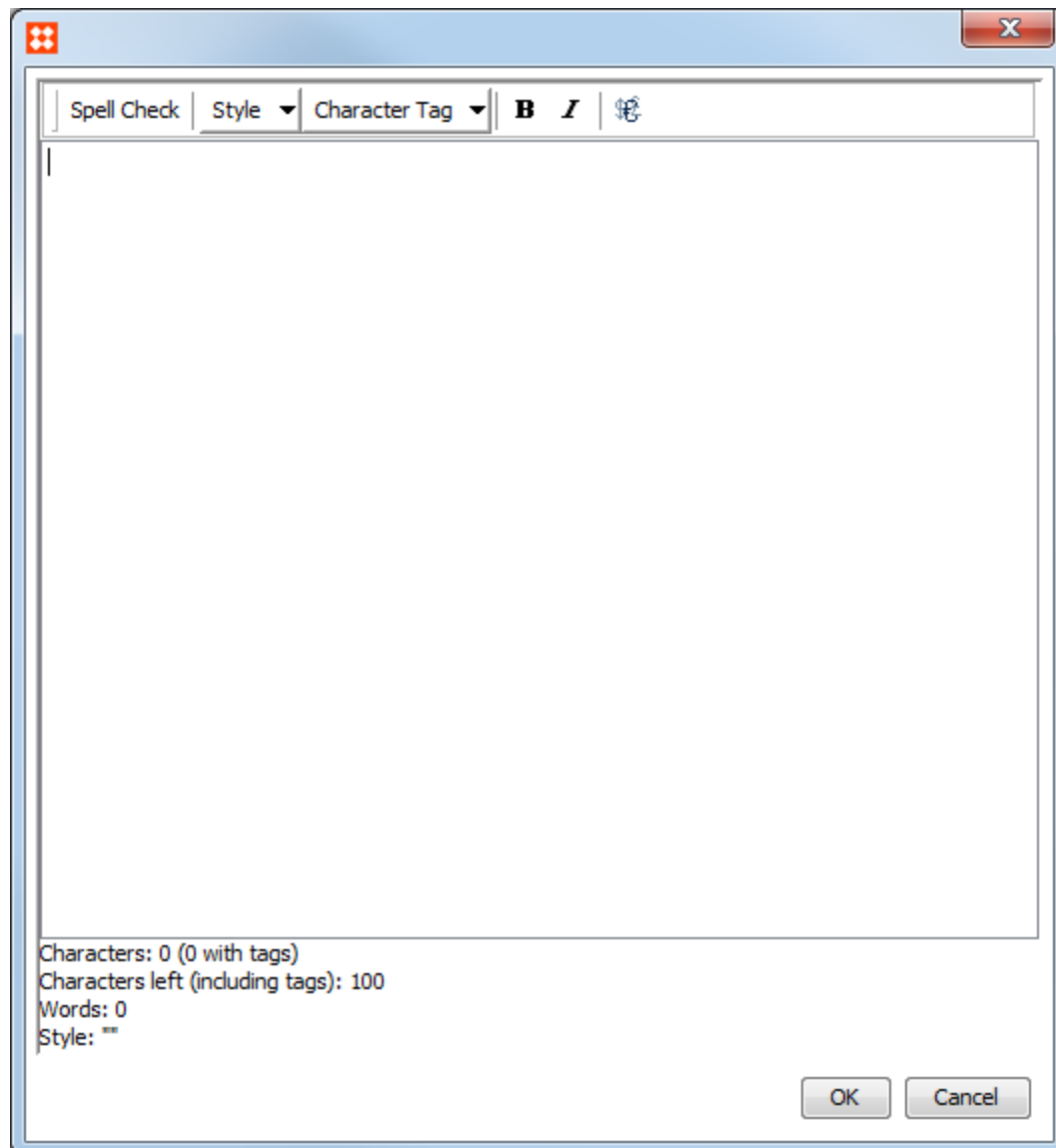
2. Depending on the object type, click the **Product** tab, **Classification** tab or **Images & Documents** tab.
3. Double click the value field of the relevant attribute and enter the new value.

Note: Only white fields are editable.

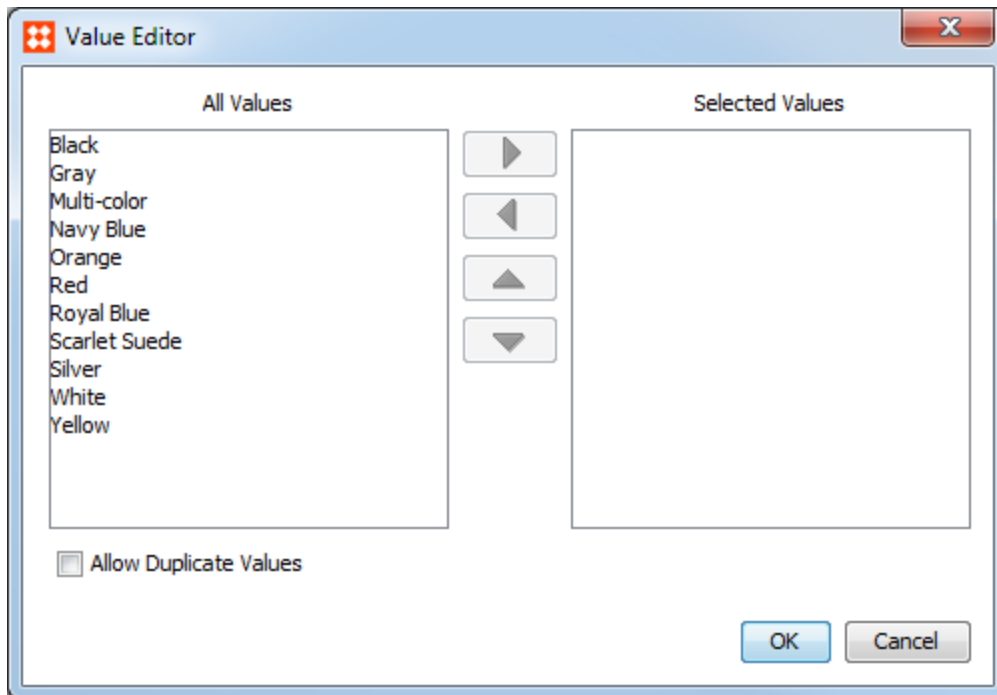
4. For additional options when editing a value field, press **Alt + F2** on the keyboard.

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

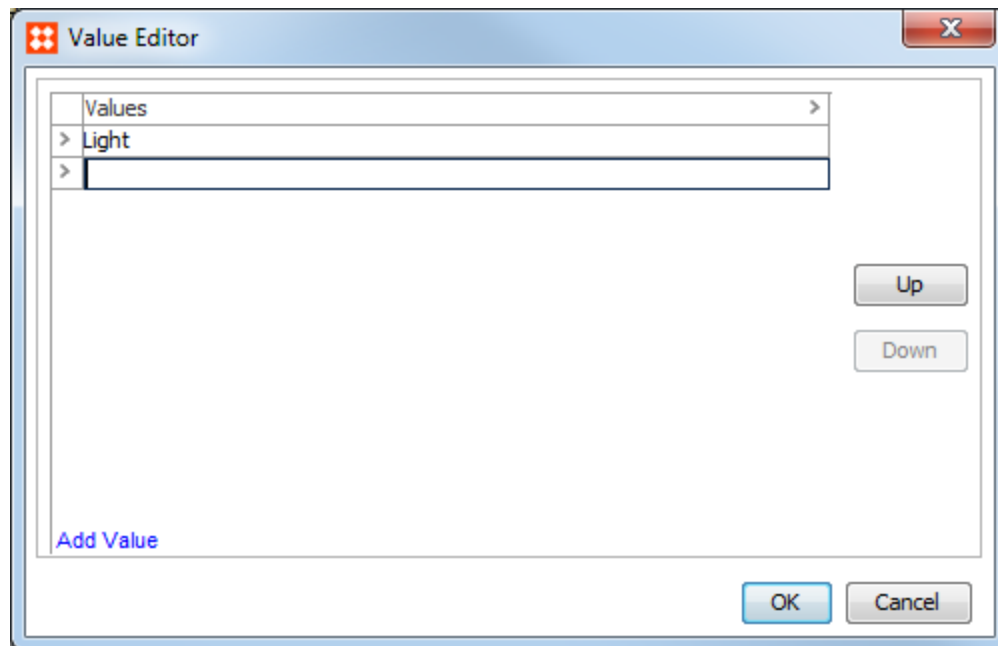
To edit the value of an attribute on an object, navigate to the desired object in **Tree** and select the value field of the relevant attribute.

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.

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.

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. If the attribute is also made valid on the child node, inherited values can be localized (e.g. overwritten directly on the child object).

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. See the **Linking Orphan Attributes** topic in this guide for steps on how to "correct" orphaned attributes.

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.
Strict Validation	If Validation base type is 'ISODate' or 'ISODATE and Time', setting this field to 'Yes' make it so that only the desired Mask and way of entering a date is accepted. E.g.: For an ISODate, the only way to enter in a valid and accepted date would be in the following format with dashes between numbers: YYYY-MM-DD. This is set after the attribute is created on the Attribute tab under the Description flipper.

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
Condition		Allows applying conditions.	[AttributeID] = [Value1];[Value2]
Date		Allows a date to be entered according to a defined input mask, indicating the valid pattern of the value.	The mask is: DD-MON-YYYY. A valid date is e.g.: 15-JAN-2008
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 (-) and 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
ISO Date		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
ISO Date and Time		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

Validation Base Type	Icons	Description	Example
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
Number		Allows any numeric characters.	Valid values: 78, 1, .34, and 384.5. Invalid values: Fractions such as 50 1/2
Number Range		Allows a range of numbers, separated by a dash (-). First value must be less than or equal to second value.	Valid values: 1.2-3.6, 100-100, 5-7 Invalid values: 7-5, 76
Numeric Text		All characters (numeric and alpha) are valid. It is also possible to link a Unit.	Valid values: Converter 20 bit, Cable 2.2 mm
Regular Expression		Allows for the applying of regular expressions as a validation base type.	[a-z]{3} ([0-9]{3})
Text		All characters (numeric and alpha) are valid.	
URL		Allows applying a URL.	

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.

Editing Validation Rules

Validation rules for Attributes and List of Values, (LOVs), are changed in [System Setup]: **Attribute Group: Attribute** or the **List of Values / LOVs: List of Values** tab.

System Setup

- GDSNgrossWeight
- GDSNHeight
- GDSNIsBaseUnit
- GDSNIsInvoiceUnit
- GDSNIsOrderableUnit
- GDSNWidth
- GTIN
- GTIN0
- GTIN12
- GTIN13
- GTIN8
- GTIN Name
- Input Mask**
- Max Width/Min width v
- Qty of Next Higher Pac
- Qty Of Next Lower Pac
- Regular expression
- UPC
- GDSN Receiver System A
- GDSN System Attributes
- GDSN Validation Group
- IDOC
- Locale
- Manual Sequencing
- Mass Creation
- Metadata
- Object Data
- Party Data
- Person
- Specifications
- System
- System Attributes

Input Mask - Attribute

Validity
Profile
Log
State Log
Tasks

Attribute
References
Attribute Transformation

Description

Name	Value
ID	Input Mask
Name	Input Mask
Last edited by	2017-03-28 07:39:09 by ...
Full Text Indexable	No
Externally Maintai...	Yes
Hierarchical Filter...	None
Calculated	No
Type	Specification
Dimension Depe...	
Mandatory	No

Attribute Validation

Name	Value
Validation Base Type	Date
List Of Values	N/A
Multi Valued	No
Mask	
Minimum Value	N/A
Maximum Value	N/A
Maximum Length	N/A

Edit Validation Rule

Sometimes it might be necessary to edit the validation rule for an Attribute or List of Values. For instance, a change of validation rules may be necessary when two or more Attributes or List of Values are merged into one. Prior to merging, it would be necessary for the Attribute or List of Values to share the same validation rules. The validation rules are explained in the **Validation Rules** section [here](#). This section is used to describe some of the limitations encountered when changing validation rules; specifically the 'Validation Base Type'.

Attributes and LOVs with values: If the attribute being changed has values, it can only be changed to a validation rule. For instance, it may be possible to change the Validation Base Type from 'Date' to 'ISO Date' but you would not be able to change 'Text' to 'Date' if text values were being used.

Workspace Values: Sometimes the 'Validation Base Type' cannot be changed on an attribute that is not 'Externally Maintained' due to values being different across multiple workspaces. This can be overcome by changing the attribute to **Externally Maintained** and choosing which workspace the existing values should be taken from as shown in the image below. It can even be changed back to not Externally Maintained but it is important to note that in doing so, all of the revision history will be lost.

Externally Maintained

Is the attribute externally maintained:

☐ No

☒ Yes

Choose Workspace

Values of externally maintained attributes are the same across all workspaces. Please select which workspace existing values should be taken from.

☐ Approved

☐ Main

Operation might be time consuming

Save Cancel

Input Masks

Input masks provide a set format for data entry in an attribute or list of values by using characters and symbols. When you apply an input mask to a field, anyone who inputs data in that field must follow the specific pattern defined by the input mask.

Input masks are a part of the Validation rules that are set in **Attributes** and **List of Values** in the [System Setup].

1. Create attribute or LOV.
2. Select the Attribute or List of Values tab.
3. Click on the **Edit Validation Rule** link.
4. Edit the **Mask**.

Example Input Mask Use Cases

Example	Description
An attribute contains values for the 5+4 U.S. zip codes.	A pattern could be set that would require entering: • 00000-0000

Example	Description
	The valid value for this attribute would have to be patterned as 12345-1234. Any other pattern, such as 123-1234 or 12345 would result in an error message appearing and the entry would not be accepted.
A LOV for an attribute contains values for a unique color code. Examples: BK (01) RD (02) Y (03) GR (04)	The requirement for this LOV input mask are: • The first character must contain an alpha character. • There may or may not be a second alpha character. • A space must exist between the letters and the numbers. • There must be two corresponding numerical characters and they must be enclosed in parenthesis. Thus the input mask would look something like this: L? (00)

Definitions of Input Mask Characters

The following table lists the different restrictions that may be imposed on a value through the use of an input mask. Remember, this list focuses on each of the values for an input mask. They can be combined to create a combination of patterns for an attribute or LOV.

Mask	Description	Input Mask Example
9	The corresponding character must be numeric, (0-9), a space, or an empty value.	Input Mask: 99999 Valid values: 1, 12345, 12 45 Invalid values: 12-45, A45, +23
#	The corresponding character must be one from the following set: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, <space>, -, +}	Input Mask: ##### Valid values: 12345, -23 5, 1234+ Invalid values: AA 45, 123#5
?	The corresponding character must be an alphabetic letter up to the length of the mask. Empty values are OK at the end of the entry but no spaces otherwise.	Input Mask: ?????? Valid values: ABCDE, AbCd, a Invalid values: AB12, A+, AB DE

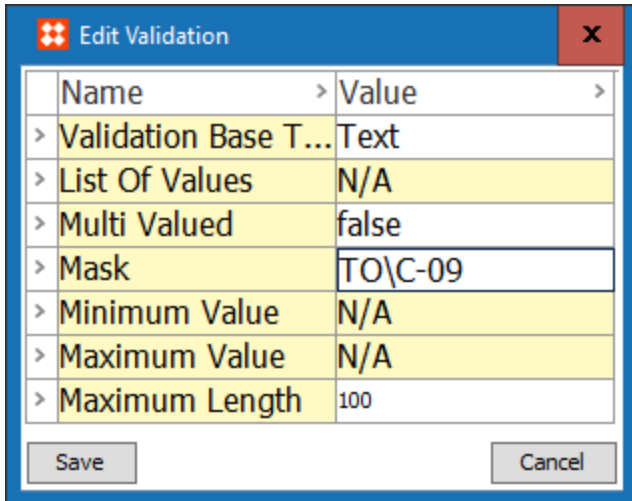
Mask	Description	Input Mask Example
a	The corresponding character may be either alpha or numeric but no special characters. Empty values are OK at the end of the entry but no spaces otherwise.	Input Mask: aaaaaa Valid values: aB123, 1234, ab1 Invalid values: AB 12, +GT
C	This will allow any character in the corresponding position, including spaces and empty values at the end of the entry.	Input mask: CCCCC Valid values: ABC, ab 12, 1, +45, 23% Invalid values: 123456 as the last digit exceeds the limit of the mask.
&	This will allow any character in the corresponding position, including spaces but no empty values at the end of the entry.	Input mask: &&&& Valid values: 123ab, 12 AB, -1Xy% Invalid values: 123, abcd
L	The corresponding character must be an alphabetic letter up to the length of the mask. No spaces or empty values at the end of the entry are permitted.	Input mask: LLLLL Valid values: ABcDE, ABCDE, abcde Invalid values: ABC, AB123, +abcd
A	The corresponding character may be either alpha or numeric but no special characters or spaces and no empty values at the end of the entry.	Input mask: AAAAAA Valid values: abcde, a1b2c, 12345 Invalid values: B1, ab de, 45%
0	The corresponding character must be numeric, (0-9), with no spaces or empty values.	Input mask: 00000 Valid values: 12345 Invalid values: 12, 12 34, 12abc

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 'T' and the '-' are always required to be entered but the '99' are masks for optional number per the definition in the table above. This means that the values T-1 and T-12 would be valid values but the values R-12 or T=1 would not be valid values.

If the static mask requires the usage of one of the mask characters mentioned in the mask character table above, it is possible to release or 'escape' the character rules by typing a '\' in front of the character. For example, if you wanted a static mask like 'TOC-09' meaning a static prefix of 'TOC-' followed by a required number '0' followed by an optional number '9', the 'C' would allow any character to be entered under the mask rules for 'C'. However, by entering the '\' in front of the 'C' as shown in the image below, the 'C' no longer follows the mask rules and would require the 'C' be entered as a valid value.



Name	Value
> Validation Base T...	Text
> List Of Values	N/A
> Multi Valued	false
> Mask	TO\C-09
> Minimum Value	N/A
> Maximum Value	N/A
> Maximum Length	100

Save Cancel

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 [Hierarchical Filters on Attributes](#). 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.

The screenshot displays the 'Color, Interior - Attribute' configuration page. The left sidebar shows the 'System Setup' tree with 'Attribute Groups' expanded. The main panel shows the 'Attribute' tab for 'Color, Interior'. The 'Attribute Validation' section is visible, showing a 'List Of Values' field with a filtering icon.

Description	
Name	Value
ID	Color, Interior
Name	Color, Interior
Last edited by	2015-08-10 10:06:26 by USER
Full Text Indexable	No
Externally Maintained	No
Completeness Score	
Hierarchical Filtering	None
Calculated	No
Type	Specification
Dimension Dependencies	Country;
Mandatory	No
Attribute Help Text	abc
Condition	
Display Name	abc
Display Sequence	123
Purpose	abc
Translate Item Data	123

Attribute Validation	
Name	Value
Validation Base Type	List Of Values
List Of Values	Colour
Multi Valued	Yes
Mask	
Minimum Value	N/A
Maximum Value	N/A
Maximum Length	N/A

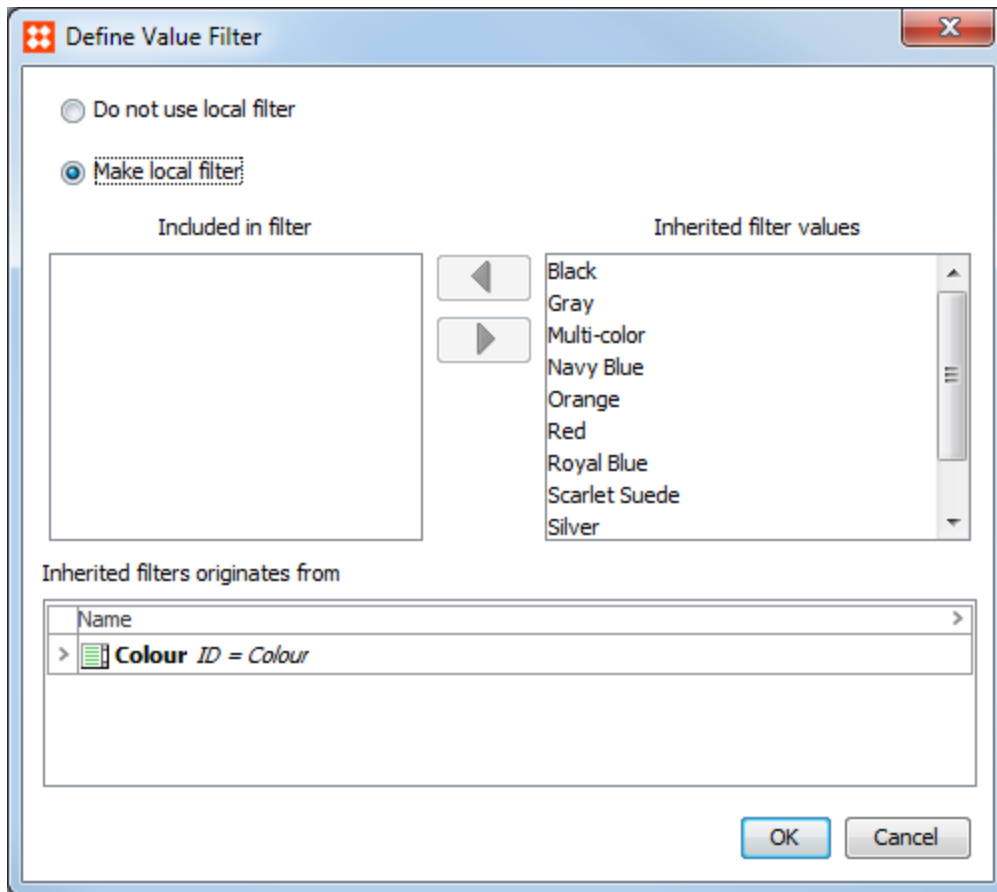
Edit Validation Rule

Aspects		
Component	Name	Description

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 [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 [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.

SKU 00001 rev.0.1 - Product

Product
References
Referenced By
Images & Documents
Commercial
Tables
Status
State Log
Tasks

Description

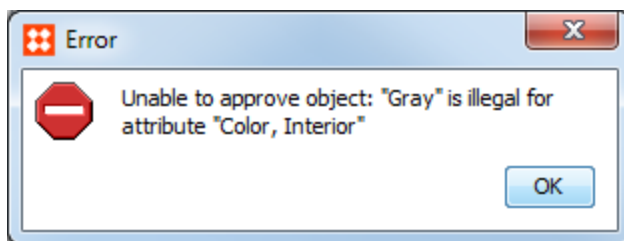
Name	>	>	Value
ID			SKU 00001
Name			SKU 00001
Object Type			Product
Revision			0.1 Last edited by USER on Tue Aug 11 12:00:43 EDT 2015
Approved			✖ Last Approved on Thu Aug 06 13:49:17 EDT 2015
Translation			Not Translated
Path			Primary Product Hierarchy/Regional Products/Apparel/Winter/Winter Coats/Ski Jacket Xtreme/S
Marketing Description			fx
Object Type			fx Product
Product Variant Priority			1
Status			Duplicate Item

No Primary Image

Attribute Group

Name	>	>	Value
Color, Interior			Gray
Size			Small

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



Click **OK** in the warning dialog box, then select a legal value.

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.

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 an integrated part of the STEP code and therefore cannot be found in a separate file. However, most entries can be verified by accessing the admin portal. For more information on reviewing system information found in the admin portal, see the **Accessing the Admin Portal** documentation and the **Configuration** documentation, both of which can be found in the Admin Portal guide. If the user has the necessary privileges to access the admin portal, the list of properties that dictate the behavior of background processes can be found there under the Configuration tab. Listed below are the properties related to configuration of background processes on a STEP system:

BackgroundProcess.ProcessType.Exporter.Queue=OUT

BackgroundProcess.ProcessType.Importer.Queue=IN

BackgroundProcess.ProcessType.Schedule.Queue=SCHEDULE

BackgroundProcess.ProcessType.WebPublisher.Queue=OUT

BackgroundProcess.Queue.IN.Size=1

BackgroundProcess.Queue.MISC.Size=2

BackgroundProcess.Queue.OUT.Size=1

BackgroundProcess.Queue.SCHEDULE.Size=1

The screenshot shows the 'Configuration' tab in the Stibo Systems interface. The 'BackgroundProcess' section is expanded, displaying several configuration items with their default values and descriptions:

- BackgroundProcess.DisableAttentionCountCalculation=false**
 - Using default value
 - Must be a boolean (true or false).
 - Set to true if attention count calculations should be disabled - might reduce load on server
- BackgroundProcess.ForceHideTemplates= (not set)**
 - Using default value
 - Forcess background process templates (sperated with ' ') to be hidden even if they by default should - ie. this overrules the doExpose setting of background process services.
- BackgroundProcess.ForceShowTemplates= (not set)**
 - Using default value
 - Forcess background process templates (sperated with ' ') to be shown even if they by default should not - ie. this overrules the doExpose setting of background process services.
- BackgroundProcess.ForcedShutdown=false**
 - Using default value
 - Must be a boolean (true or false).
 - Set this if having trouble starting step due to some plugins failing, in general, this should not be enabled on production systems. Should be space separated list of components that are allowed to fail or * if all components are allowed.
- BackgroundProcess.PollingInterval=5000**
 - Using default value
 - Must be an integer.
 - The interval (in milliseconds) the service manager waits before checking for new tasks in the workflow system.
- BackgroundProcess.ProcessNavigatorUpdateInterval=20**
 - Using default value
 - The interval between the GUI checks for new background process instances when the background process navigator is in front

Changing the default configuration

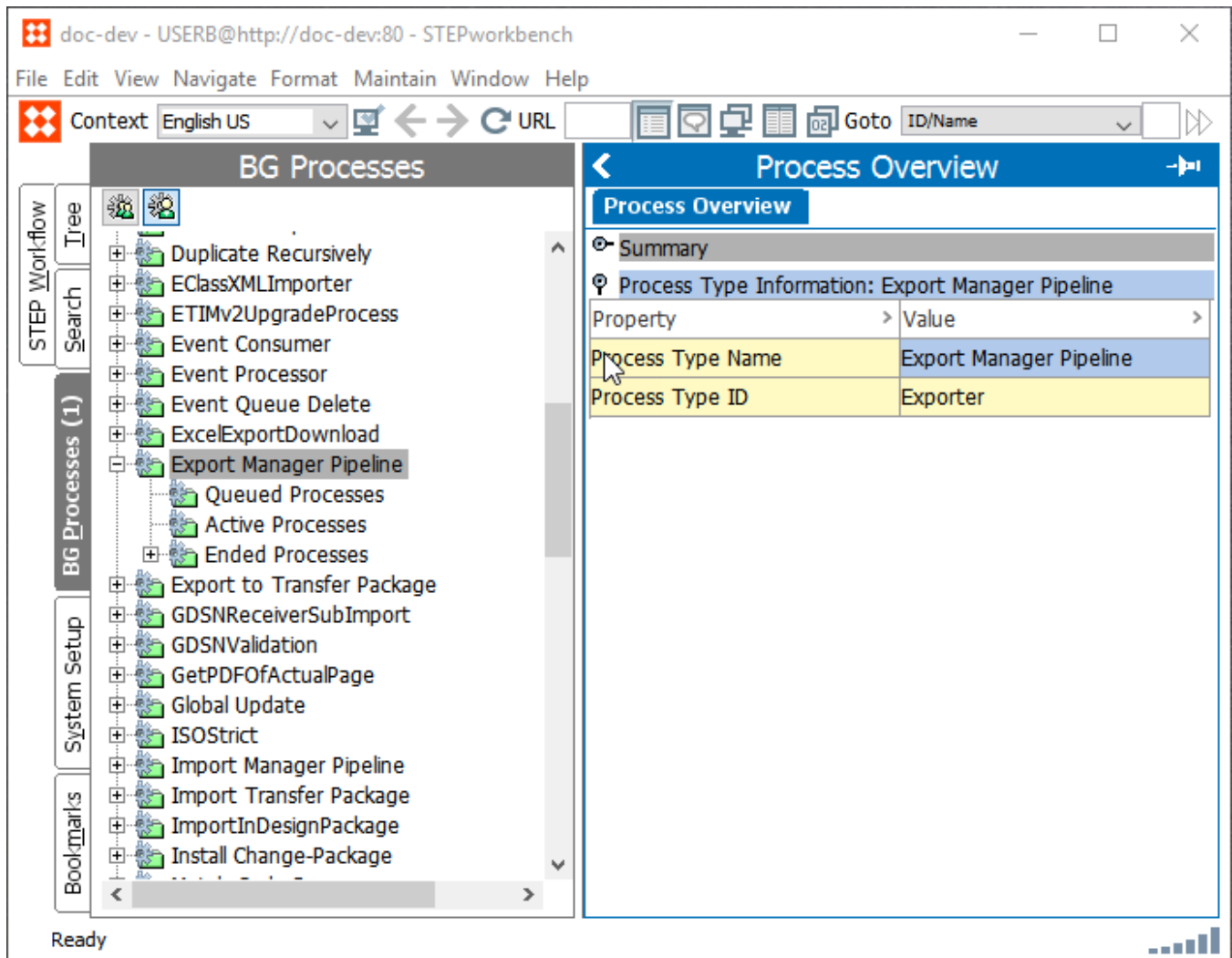
It is possible to change the default configuration.

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

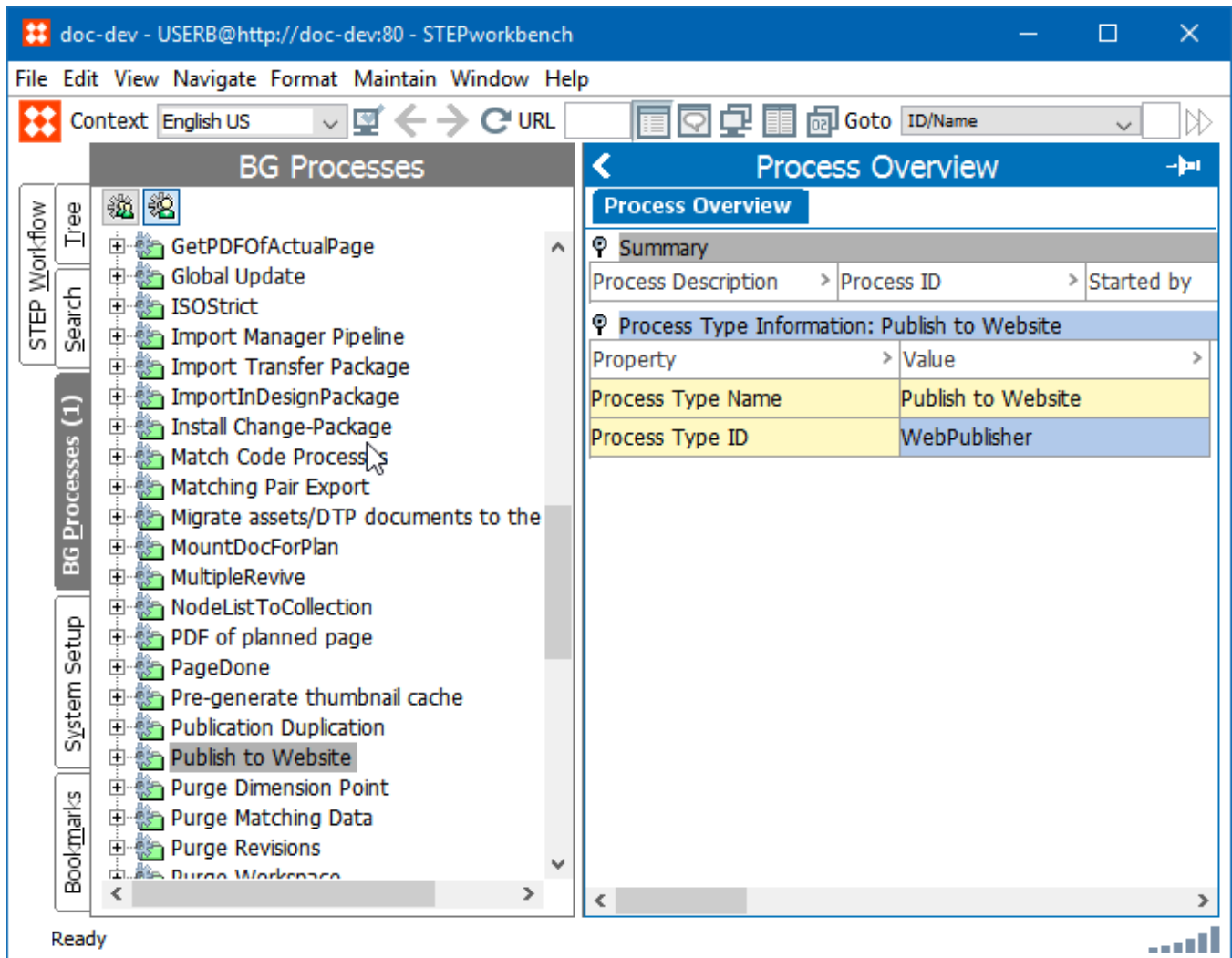
1. Click the [Background Processes] tab.
2. Click **Export Manager Pipeline**.

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.

Scheduled Process Properties

Once created, a scheduled process displays on the Background Processes tab under the 'Scheduled Processes' folder, and then in a folder that indicates the status:

- **Queued Processes** are waiting for the next scheduled run.
- **Active Processes** are currently running.
- **Ended Processes** have been aborted.

The screenshot shows the 'BG Processes' window. On the left is a tree view with folders for 'Queued Processes', 'Active Processes', and 'Ended Processes'. The 'Export using 'Collection of Hats'' process is selected under 'Queued Processes'. On the right, the 'Background Process' properties are displayed in a table.

Property	Value
Started by	USERJ
Id	BGP_185495
Description	Export using 'Collection of Hats' - Start every 60 minutes, starting Mon Sep 19 12:20:00 EDT 2016
Schedule	Every 60 minutes, starting Mon Sep 19 12:20:00 EDT 2016
Execution Server	doc-dev
Status	waiting
Created	Mon Sep 19 12:20:48 EDT 2016
Started	Mon Sep 19 12:20:56 EDT 2016
# of warnings	0
# of errors	0
Next Run	Mon Sep 19 13:20:51 EDT 2016

The Background Processes editor tab displays the following data about the selected export:

- **Started by** shows the user who scheduled the export process.
- **ID** shows the background process ID.
- **Description** is the text added in the 'Scheduled Process Name' field when the scheduled export was created.
- **Schedule** defines the frequency that the export is scheduled to run.
- **Execution Server** shows the name of the database server running the export.
- **Status** can be waiting, active, or aborted, depending on the current situation.
- **Created** shows the day, date, and time that the scheduled export was created.
- **Started** is the day, date, and time that the scheduled export first ran.
- **# of warnings** is the number of warning messages included in the execution report.
- **# of errors** is the number of error messages included in the execution report.
- **Next Run** is the day, date, and time that the export will run again.

Data Containers

This documentation provides guidance on how to represent and structure complex entity data within STEP through the use of data containers. This topic provides an overview of complex data modeling as well as basic information regarding data container functionality available within workbench and Web UI.

Composite Concept

Customer data can be complex with a single record that comprises many different pieces of information. A single customer record is typically linked to multiple objects via references representing addresses (e.g., main / physical, shipping, mailing), contact names, contact information, email addresses, phone numbers and much more. The customer record acts as a 'composite,' which is a data structure that is comprised of other data structures that need to be seen and treated as a whole.

By using Data Container Types, one is able to model a composite entity (e.g., customer record) containing Data Containers (addresses - e.g., main / physical, shipping- contact names, contact information, email addresses, phone numbers, etc.) with relevant attributes. Composite entities have many benefits, including simplified revision handling, a collective process to save and approve data, and easier management of container attributes in workflows.

Using Data Containers

Users can begin working with data containers and their associated attributes in various situations once basic setup is complete. Data container modeling is intended to ensure that all create, read, update, and delete operations can complete or fail as a whole as the operations are being done on an entity containing the data containers.

- Data Container Type is a Basic Object Type found under System Setup > Basic Object Types.
- New Data Container Types are created in workbench within System Setup > Attribute Groups and are distinguishable via the  icon. One data container can reside in multiple attribute groups.
- Data container types can contain description attributes, and these attributes can be used in multiple data container types.
- Data containers have (three) tabs when viewed in workbench: Data Container Type, Validity, and Log.
- Once the entity type validity is set, a Data Containers tab will display when clicking on the valid entity objects from the Tree. Once configured to display, the data containers will be visible on entities in the Web UI, also.
- Data containers can be imported into and exported from workbench. STEPXML, Advanced STEPXML, CSV, and Excel formats are supported. Data container setups are also transferable through change packages.
- Attribute values in data containers are searchable, and the entities are returned as results. The workbench and Web UI Advanced Search contain a Data Containers search criteria, while both the workbench Search and Web UI simple search allows for searching via attribute values in Data Containers.

- Matching can be done for objects that use data containers, making use of survivorship rules for data containers (Data Container: Trusted Source, Data Container: Most Recent).
- Event handling for data containers is supported.
- Data containers are supported in calculated functions, and API support for access to data containers in JavaScript business rules is provided.
- Data Containers are supported in the Simple Address Component model, as well the Google Map API, Loqate, and Trillium integrations used for address standardization.
- Data containers, and the attributes within them, are supported in data profiling.
- Completeness statistics take the attributes in the data containers into account.
- Revisions made to data containers, including creating, deleting, and editing attributes, are created on the applicable entity object.
- Data containers can be maintained in Web UI by using the Data Container Editor component.

Considerations and Limitations

Please note the following while working with data containers:

- Data containers can only be made valid for entities, thereby only description attributes can be used in data containers.
- An entity can only contain 100 data container instances.
- Data containers cannot be externally maintained.
- Data containers are not dimension dependent; however, the individual attributes in data containers are supported with regards to dimension dependencies.
- Data containers cannot be translated via the standard translation tools.
- Data containers cannot be initiated into workflows; they follow the workflow of their owning entities.
- Data Containers are not supported in Web Services.

For more information on configuring data containers and using Web UI, see the following topics:

- **Setting up Data Containers in Workbench**
- **Working with Data Containers in Web UI** in the **Web User Interfaces** documentation

Additional information can be found in the relevant STEP documentation for a particular functionality (i.e., Data Exchange, Navigation and Searches / Advanced Search (Web UI), Data Profiling, and more).

Setting Up Data Containers in Workbench

This topic provides basic information on how to set up data containers in STEP Workbench. Once system setup is complete, users can begin working with data containers in workbench and start configuring Web UI screens and components.

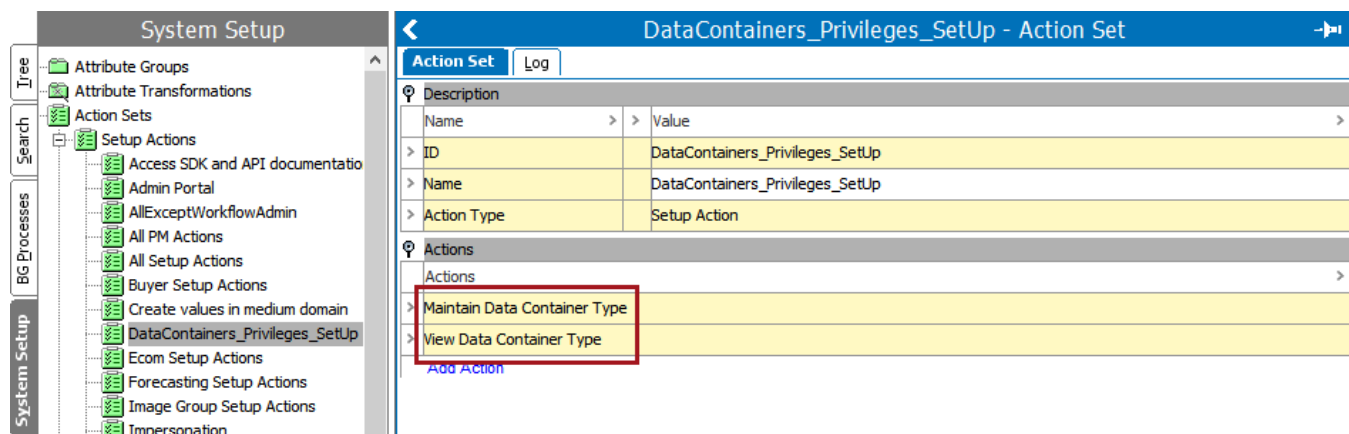
Prerequisites

Before setup begins, users should be familiar with basic information regarding using data containers and associated limitations. This information is available in the **Data Containers** topic in **System Setup / Super User** documentation.

To setup new object types and work with attributes and entity objects, you must have all the proper privileges.

There are two setup actions relating to data containers:

1. Maintain Data Container Type
2. View Data Container Type



There are six user actions relating to data containers:

1. Approve data containers
2. Create data containers
3. Delete data containers
4. Modify metadata for data container
5. View data containers
6. View metadata for data containers

The screenshot displays the 'System Setup' window with a tree view on the left and a detailed view on the right. The tree view shows the hierarchy: System Setup > Action Sets > User Actions > DataContainers_Privileges. The right pane is titled 'DataContainers_Privileges - Action Set' and contains a table with the following data:

Description	
Name	Value
ID	DataContainers_Privileges
Name	DataContainers_Privileges
Action Type	User Action

Below the table is a section titled 'Actions' with a list of actions:

- Approve Data Containers
- Create data container
- Delete data container
- Modify metadata for data container (property value) (also translate)
- View data container
- View metadata for data container (property value)

An 'Add Action' link is visible at the bottom of the actions list.

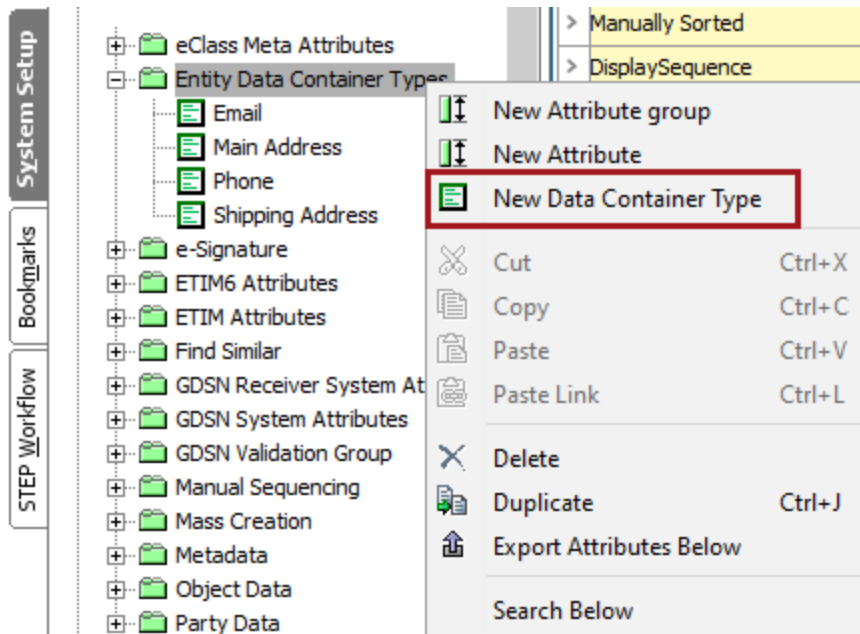
For more information, see the **Action Sets** section of the **System Setup / Super User** documentation.

Adding and Configuring Data Container Types

New data container types are created in workbench within System Setup > Attribute Groups. Insert data containers into new or existing attribute groups using the same functionality used to create new attribute groups and new attributes. Data container types are distinguishable via the  icon.

Creating Data Container Types and Setting Validity

1. Right click on an attribute group.
2. Select **New Data Container Type**, as shown below.



3. Give the Data Container Type an ID and Name. Click **Create**.

Data Containers IDs are global/unique across entities.

When configuring a data container, you will see three tabs in the data container editor: Data Container Type, Validity, and Log. Each one is described in detail later in this topic.

4. Next, select the validity for the data container type just created. Data containers are only valid on entities, so you will only see entity types listed on the Validity tab. Once the entity type validity is set, a Data Containers tab will display when clicking on the valid entity objects from the Tree.

Data Container Type
Validity
Log

Valid for Entity Types

☐ All
☐ Account
☐ Address
☐ Address Root
☐ All Customers
☐ Contact
☐ Contacts Root
☒ Customer

Configuring Data Container Types

Data container types can contain description attributes, and these attributes can be used in multiple data container types. Also, a data container can reside in multiple attribute groups.

- Enter an ID Pattern. (If nothing is entered, then users will be able to determine their own data container ID when adding them to an entity object.)
- Determine if multiple data containers of the type in question can be added per Data Container Type. The setting is 'No' by default. Click in the value field to change to 'Yes.'

It is very important to note that once this parameter is set to 'Yes,' you cannot set it back to 'No.' If you set it to 'Yes' by mistake, you will need to remove the validity, delete the data container type, and recreate it.

- To add attributes, click the **Add Attribute** link on the Data Container Type tab. Individual attributes need to be selected (no attribute groups); however, multiple attributes can be selected and added at the same time.

Note: Formatting of attributes is not supported. If formatting is used, the attributes are displayed as read only in the Web UI and, in the case of multi-valued data containers, locked for editing. Calculated attributes can be used. And, if using inline references in attribute values, the received value is displayed as read only.

Attributes within a data container are shown in alphabetical order in workbench. They can be set up to display in a different order in Web UI via the Data Container Editor component properties. See the **Data Container Editor** topic in Using a Web UI documentation.

- To add the data container type to additional attribute groups, click the **Add Attribute Group** link.

System Setup

- Attribute Groups
 - Attribute Group
 - Category Specific Attributes
 - Customer Data
 - Data Containers
 - Dates
 - Description Attributes
 - Dimensions
 - Display
 - eClass Attributes
 - eClass Mapped Attributes
 - eClass Meta Attributes
 - Entity Data Container Types
 - Email
 - Main Address
 - Phone
 - Shipping Address
 - e-Signature
 - ETIM6 Attributes
 - ETIM Attributes
 - Find Similar
 - GDSN Receiver System Attribu
 - GDSN System Attributes
 - GDSN Validation Group
 - Manual Sequencing

Shipping Address - Data

Data Container Type | Validity | Log

Description

Name	Value
ID	ShippingAddress
Name	Shipping Address
Last edited by	2016-10-13 13:07:40 by USER8
ID Pattern	
Allow multiple data containers	Yes

In Attribute Groups

ID	Name
EntityDataContainers	Entity Data Container Types

[Add Attribute Group](#)

Valid Attributes

ID	Name
Street	Street
Country	Country
Zip	Zip
State	State
CountryISOCODE	Country ISO Code
City	City

[Add Attribute](#)

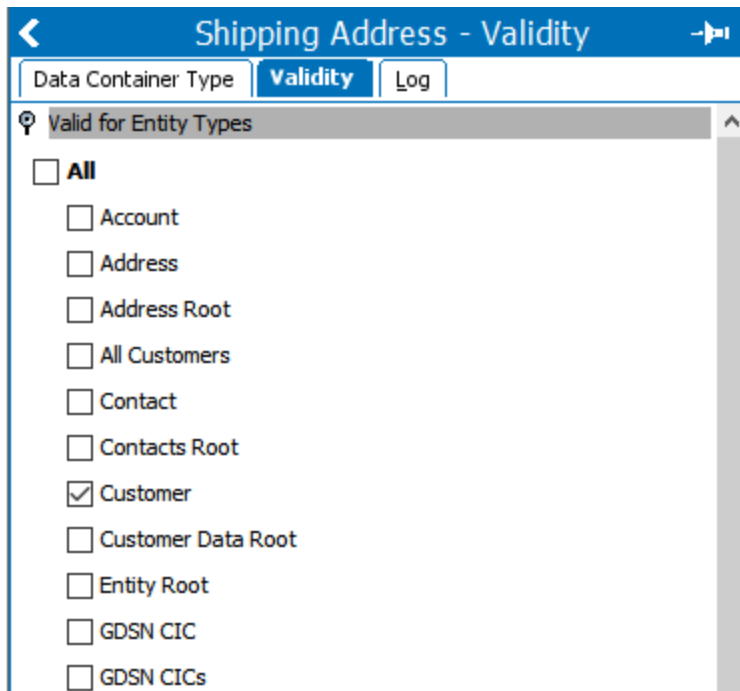
Data Container Tabs

When configuring a data container, you will see three tabs in the data container editor.

1. Data Container Type

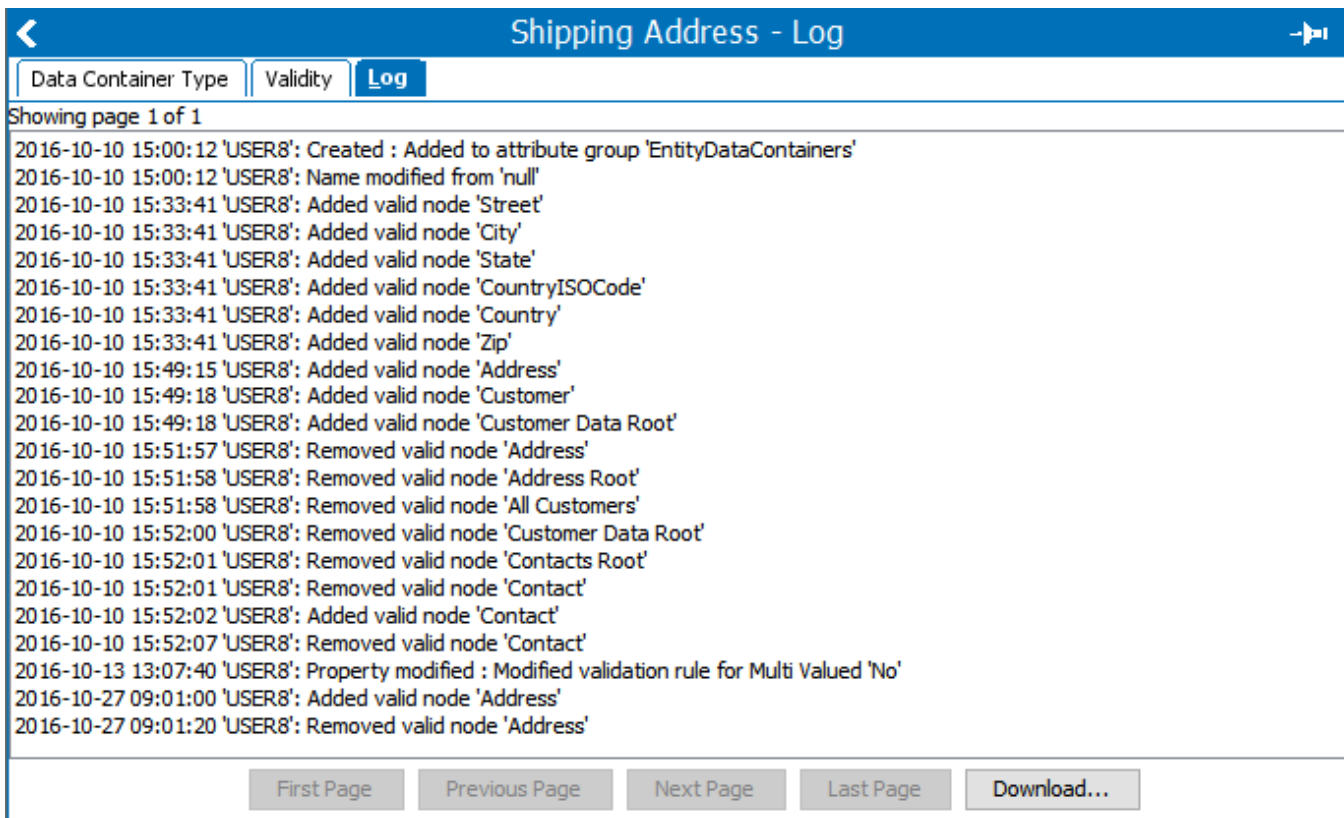
The Data Container Type (shown in the previous screenshot) tab is where you add attributes for building a data container. In this section, an ID pattern can be specified, creating an automatic method for naming new data containers of this type. Another attribute that can be set is the ability to allow multiple data containers of this type, such as multiple shipping addresses.

2. Validity



The Validity tab, like with other validity options in the workbench, sets the validity for the Data Container Type in relation to Entity Types. Valid entities will also display the Data Containers tab within the object editor.

3. Log



The Log tab provides a record of actions involving the selected Data Container Type. Users may download this record via the 'Download...' button.

Adding and Maintaining Data Containers

In the Tree, when selecting an entity object in which data container types are valid, you will see a Data Containers tab on the screen. This is where you can maintain the attribute values for each container.

The screenshot shows the 'Tree' on the left with a hierarchy: Entity Root, Address Root, Contacts Root, and Customer Hierarchy. Under Customer Hierarchy, 'Customer A' is selected. On the right, the 'Customer A re' details are shown with tabs for 'Customer', 'Data Containers', 'References', and 'Referenced By'. The 'Data Containers' tab is active, showing a table with columns 'Name' and 'Value'. The table contains the following data:

Name	Value
ID	CUS_101545
Name	Customer A
Object Type	Customer
Revision	0.5 Last edited by USER8 on Thu
Path	Entity hierarchy root/Entity Root/

1. Click on the Data Containers tab to see data container types that are valid for that particular object.

The screenshot shows the 'Data Containers' tab for 'Customer A re'. It has tabs for 'Customer', 'Data Containers', 'References', 'Referenced By', 'Status', 'State Log', and 'Tasks'. The 'Data Containers' tab is active, showing a table with columns 'Name' and 'Value'. The table contains the following data:

Name	Value
ID	CUS_101545
Name	Customer A
Object Type	Customer
Revision	0.5 Last edited by USER8 on Thu
Path	Entity hierarchy root/Entity Root/

2. Add data container entries to each data container type via the **Add Data Container** link. Alternatively, you may click on the > icon at the beginning of each row to access the Add Data Container option.

The screenshot shows the 'Data Containers' tab for 'Customer D rev.0.3'. It has tabs for 'Customer', 'Data Containers', 'References', 'Referenced By', 'Status', 'State Log', and 'Tasks'. The 'Data Containers' tab is active, showing a table with columns 'Name' and 'Value'. The table contains the following data:

Name	Value
ID	CUS_101545
Name	Customer A
Object Type	Customer
Revision	0.5 Last edited by USER8 on Thu
Path	Entity hierarchy root/Entity Root/

A dialog box titled 'ID for Data Container' is open, showing 'MainAddress' as the ID. The dialog has 'Create' and 'Cancel' buttons.

If a data container type does not allow for multiple data containers, then the Add Data Container link will no longer display once a data container is added. In the example screenshot shown in the next step, Main Address allows for only one data container while multiple data containers are allowed for Shipping Address.

3. Enter attribute values for each data container.

Customer	Data Containers	References	Referenced By	Status	State Log	Tasks

🔍	Main Address

ID	Attribute Name	Value
> MainAddress_Denmark	City	Højbjerg
	Country	Denmark
	Country ISO Code	DK
	State	
	Street	Axel Kiers Vej 11
	Zip	8270

🔍	Shipping Address

ID	City	Country	..	State	Street	Zip
> ShippingAddress_Canada	Toronto	Canada	CA	ON	116 Spadina Ave	M5V 2K6
> ShippingAddress_France	Paris	France	FR	Île-de-France	43 Boulevard Haussmann	75009
> ShippingAddress_Kennesaw	Kennesaw	United States	US	GA	3550 George Busbee Pkwy NW	30144
Add Data Container						

4. To delete data containers, click the > icon at the beginning of the data container row. Select **Remove Row(s)** to delete the existing data container.

🔍	Shipping Address

ID	City
> ShippingAddress_Canada	Toronto
> ShippingAddress_France	Paris
> ShippingAddress_Kennesaw	Kennesaw
>	

Hide

Show All Rows

Rotate Table

Add Data Container Ctrl+Plus

Remove Row(s) Ctrl+Minus

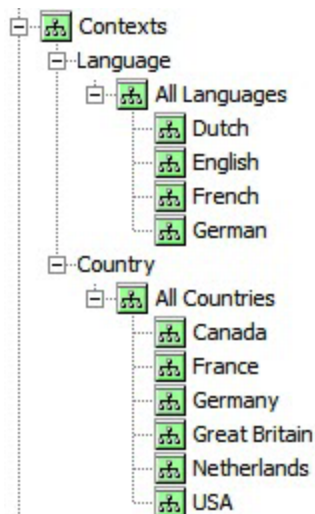
For more information about working with and maintaining data containers in Web UI, see the **Working with Data Containers in Web UI** topic in the **Web User Interfaces** documentation.

Dimensions, Dimension Points, and Contexts

This documentation section explains dimensions, dimension points, and contexts in STEP. In addition to describing what they are, this section also explains why they are used, best practices for their configuration, and examples on how to create and configure them.

One of the core features of STEP is its ability to store multi-dimensional data. Dimension-dependent data is presented differently based on the dimension in which it is stored, such as language or country. Most often, data is dimension dependent for purposes of translation when content needs to be stored in multiple languages, though sometimes data is made dimension-dependent if it must vary in different countries, markets, or business units.

Dimensions, dimension points, and contexts are maintained in the STEP Workbench under System Setup > **Contexts**.



This documentation section addresses the following topics:

- Dimensions and Dimension Points
- Maintaining Dimensions and Dimension Points
- Dimension and Dimension Points Best Practices
- Adding a Dimension Dependency After Loading Data
- Contexts
- Maintaining Contexts
- Context Best Practices
- Context Locales

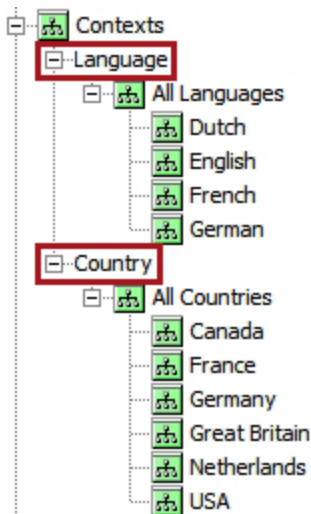
Dimensions and Dimension Points

This documentation section explains dimensions and dimension points, which are maintained in the STEP workbench under System Setup > **Contexts**.

Dimensions

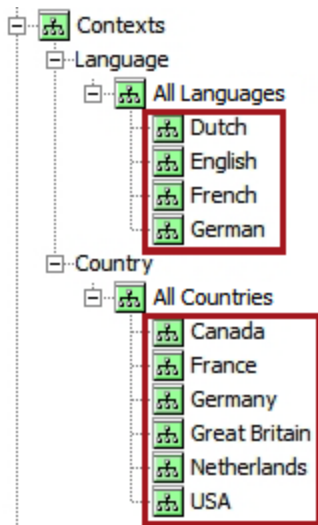
A **dimension** is a definition of a context-sensitive domain in STEP, such as language or country, which allows for different layers of information in which content can vary. A dimension is presented in STEP as a hierarchy of dimension points. Each system must have at least one dimension.

Common examples of dimensions are 'Language' and 'Country,' as shown in the below screenshot.



Dimension Points

Contained within dimensions are **dimension points**. A dimension point is, for example, a specific language or country. Data is stored in STEP in dimension points.



Dimensions are *general* and dimension points are *specific*.

Note: Though 'All Languages' and 'All Countries' are also dimension points, they should never be used when creating a context. See the **Dimension and Dimension Points Best Practices** topic for more information.

Dimension Dependent Data

Data is stored in STEP in **dimension points**. In order to place data into different dimensions, the object type, attribute, and/or reference type that will either contain or link the data together must be made dimension dependent.

Note: Data is not stored in contexts.

- **Object types** are made dimension dependent if they should have different STEP **names** in different contexts. In most cases, this is because the names are translated into different languages, though they may also have different names in different markets, for example.

For information on making object types dimension dependent, see the **Maintaining Dimension Dependent Object Types** topic.

- **Attributes** are made dimension dependent when their **values** should be different in different contexts. This also includes lists of values (LOVs).

For information on making attributes dimension dependent, see the **Dimension Dependent Attributes** topic in the **Getting Started / STEP User Guide** documentation.

- **Reference types** are made dimension dependent when different linked objects (typically assets) should be visible in different contexts.

For information on making reference types dimension dependent, see the **Reference and Link Types** topic.

- **Assets** can be made dimension dependent but this is a global setting applied under System Setup > Users & Groups > **Image & Document Settings**. For more information, see the **Image and Document Settings**

topic.

Note: In almost all circumstances, assets can be 'swapped out' in different dimensions by use of dimension-dependent reference types.

Dimension and Dimension Point Best Practices

Though STEP is highly configurable with regard to the creation and maintenance of dimension and dimension points, certain implementation standards apply when they are created. This topic outlines some considerations to keep in mind when determining how to configure your system to handle dimension-dependent data.

Dimension Do's and Don'ts

Dimension Do's

- Use as few dimensions as possible—dimensions affect usability.
 - Dimension-dependent data is a good concept but should not be taken too far.
 - The more dimensions you introduce, the more likely it is that users will have to flip back and forth between contexts. At some point, users will forget to change contexts, and then mistakes happen, such as adding data in the wrong dimension for a long time before the mistake is realized. There is no easy way to find out what should be corrected.
 - When lot of data must be entered while working in one context, and then some of it has to be entered in another context, there is a risk that the user will forget to break up the information into separate load files, or they will forget to change contexts for part of it.

Dimension Don'ts

- Do not introduce a dimension unless there is strong justification for it
- Do not use a dimension for any aspect of data that will be temporary, such as a website dimension, a print publication dimension, or promotions.

Note: Never use more than three dimensions. Typically, only two should be used, though it is possible to even use only one—**language**.

Additional Attributes as an Alternative to Additional Dimensions

As an alternative to dimensions, consider using additional attributes or additional reference types, which are acceptable if you only have a small amount of information that will vary by language or country.

Arguments for Additional Attributes	Arguments for Additional Dimensions
This is a simple approach for users to understand.	Using the same attributes in layers enables re-use of integrations, saved imports / exports, and Web UI screens.

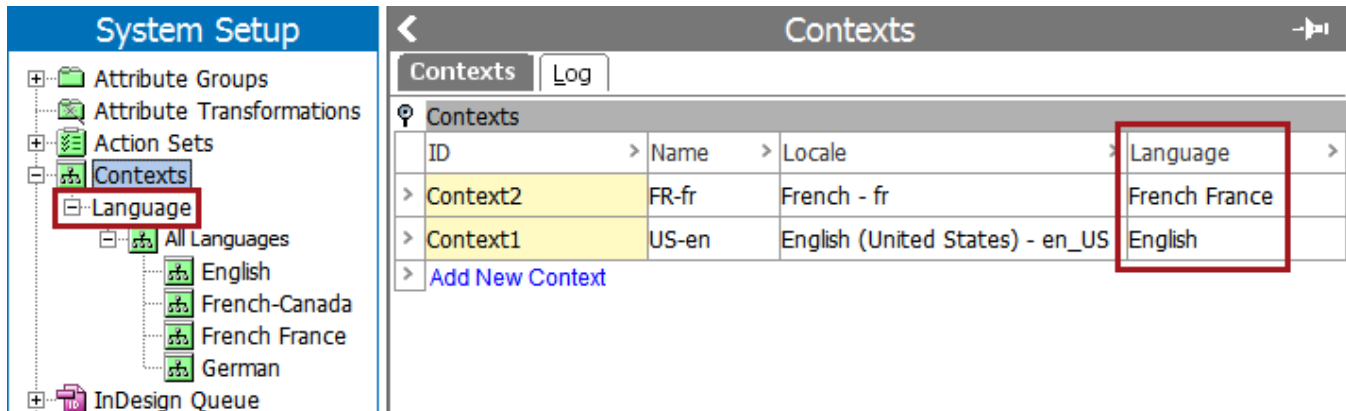
Arguments for Additional Attributes	Arguments for Additional Dimensions
Users can compare values for all versions side-by-side in an Excel file.	Users can compare values for a single attribute across dimensions using the STEP context mode. Select this icon from the workbench to switch to STEP context mode. 
More dimensions will drive more contexts. More contexts means users have to be aware of which context they should work in, causing a high risk that people will make mistakes and put data in the wrong context.	There will be fewer attributes. Separate attributes for each dimension could mean many hundreds more attributes.
More dimensions will drive multi-dimensional attributes, which are to be avoided.	
To load via Excel, it is easier to add more attribute columns to Excel than it is to separate Excel files when some of the data has to be loaded to a different context.	

Language Dimension

There **must** be a language dimension. This is the only dimension that is supported by the STEP translation tracking and translation workflow features. For more information on the translation functionality of STEP, see the **Translations** documentation.

Even if you do not use multiple languages, you must have a language dimension for the language in which you are maintaining your data. Choose a real language so that it is very clear what belongs there. The language dimension is special in STEP because it tracks changes between a source language and its translations. This means that there are no STEP tools to help users find instances where content in a country-dependent attribute value has changed and may need to be updated for other countries.

In many installations, *only* the language dimension point is required. Additional dimensions, such as country, are not needed if there is no country-specific data, such as warranty information that varies depending on which country a product is sold. The below screenshot shows a dimension hierarchy in System Setup and two contexts that only use the Language dimension.



System Setup

- Attribute Groups
- Attribute Transformations
- Action Sets
- Contexts**
 - Language**
 - All Languages
 - English
 - French-Canada
 - French France
 - German
- InDesign Queue

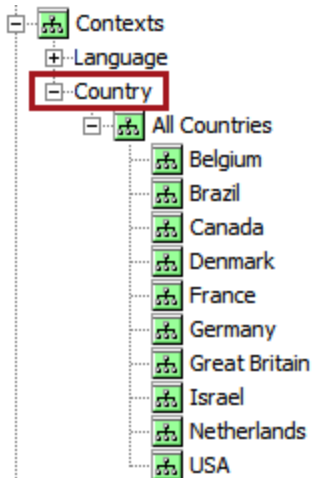
Contexts

ID	Name	Locale	Language
Context2	FR-fr	French - fr	French France
Context1	US-en	English (United States) - en_US	English

[Add New Context](#)

Country Dimension

The Country dimension comes standard with STEP, but if you do not need it, remove it until you do.



Contexts

- Language
- Country**
 - All Countries
 - Belgium
 - Brazil
 - Canada
 - Denmark
 - France
 - Germany
 - Great Britain
 - Israel
 - Netherlands
 - USA

You should only use a country dimension if the following conditions apply:

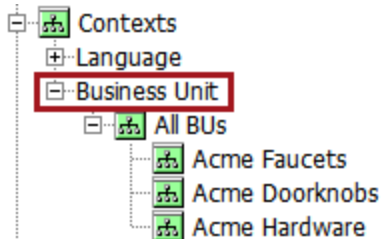
- You have products that are sold in more than one country and there are *a lot* of **attributes** with different values because of the country they are sold in.
- You have products that are sold in more than one country and there are *a lot* of **references** that are different because of the country they are sold in.
- You are using the Stibo GDSN solution, which requires that each target market references a context containing a language and country dimension. For more information on GDSN target markets, see **Creating GDSN Provider Target Markets** in the **GDSN Provider** documentation.

If there are only a handful of attributes and products are only sold in a few countries, use separate attributes to hold this information—for example, 'Description, English,' 'Description, German,' and so forth. Additional references might also be used if they are relating information that varies by country. For example, a product could reference a PDF document and the document contains different information in some countries. Instead of making the reference dimension-dependent, you could make separate reference types ('Warranty, United States,' 'Warranty,

Canada'). For information on dimension-dependent references, see the 'Dimension Dependencies for Reference Types' section of the **Reference and Link Types** topic.

Business Unit Dimension

Business unit dimensions are sometimes used in larger enterprise implementations as well as in cases where customers start out with a STEP implementation, then acquire another business.



Business unit dimensions may be needed for enterprise-wide implementations if the following conditions apply:

- The exact same products, with the same internal ID, are sold by more than one business unit.
- Each business unit must share the same pool of attributes, and there are no clear 'owners' of the attributes. For example, an attribute like 'Description' is used for products across all business units instead of there being separate attributes for each business unit—'Acme Shoes Description,' 'Acme Electronics Description,' or 'Acme Clothing Description.'

Even if these conditions apply, it should not be an automatic decision to use a business unit dimension. Different customer-facing part numbers, for example, can be stored in separate attributes. If part numbers are different based on the business unit selling the products, you *might* need a business unit dimension, but this is not enough justification on its own to do so.

Market Dimension

Market dimensions are not recommended; however, this section addresses them mainly to provide alternatives to their use.



Any requirements for market-dependent data are typically for a small number of attributes or references, which can be handled by a less-complex approach of additional attributes and/or references.

- Most market-specific attributes typically are separate attributes
- Market-specific asset references can be handled by separate reference types
- People who maintain market-specific attributes would only be able to partially approve their own content, which is the case whether they use dimensions or separate attributes

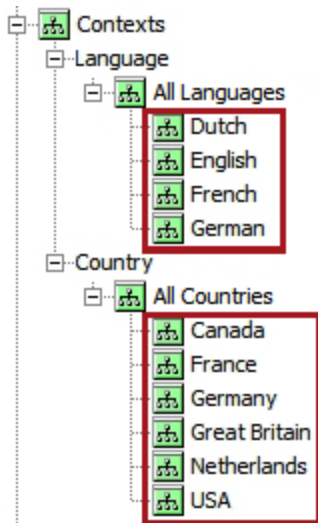
Alternatives to market dimensions are:

- Use separate attributes for web copy, print copy, and so forth
- Use separate reference types for different markets
- Group the reference types for visual separation

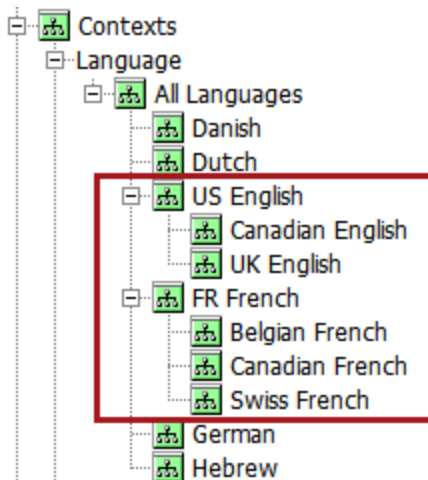
Dimension Point Do's and Don'ts

Dimension Point Do's

1. In most circumstances, dimension points should be created at the same level:



However, sometimes there may be a need to have parent / child relationships between dimensions if you need to maintain many versions of an 'almost the same' language, as pictured in the below screenshot. In this type of setup, parent values from the top-level dimension will be inherited by the child dimensions. If this is implemented, there **must** be clear ownership of the parent language. Otherwise, do not use one.

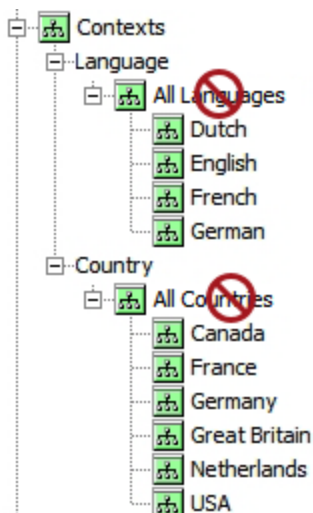


For more information on use cases and the value inheritance of parent / child dimensions and attribute values, see the **Dimension Dependent Attributes** topic.

2. Be consistent across all dimension points and use IDs based on ISO standards. More information about these standards can be found on the web.
 - For languages, use IDs based on the **ISO 639-1** Language Code (2 letter) + **ISO 3166** alpha-2 Country Code. For example, US English: **en-US**.
 - For countries, use IDs based on the **ISO 3166** alpha-2 Country Codes. For example, United States: **US**

Dimension Point Don'ts

Do not use an **All** dimension point in a context. Nobody speaks All Languages, nobody lives in All Countries, and nobody will maintain All.



Note: When content is **not** dimension dependent it is already stored in the All Languages and All Countries dimension points.

Maintaining Dimensions and Dimension Points

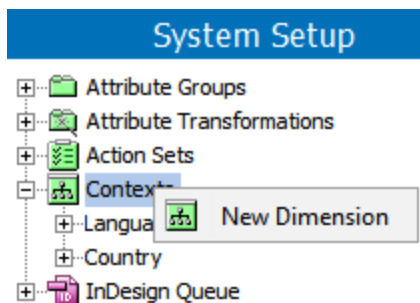
This topic describes how to:

- Create dimensions and dimension points
- Delete dimensions and dimension points
- Select a dictionary for a language dimension point

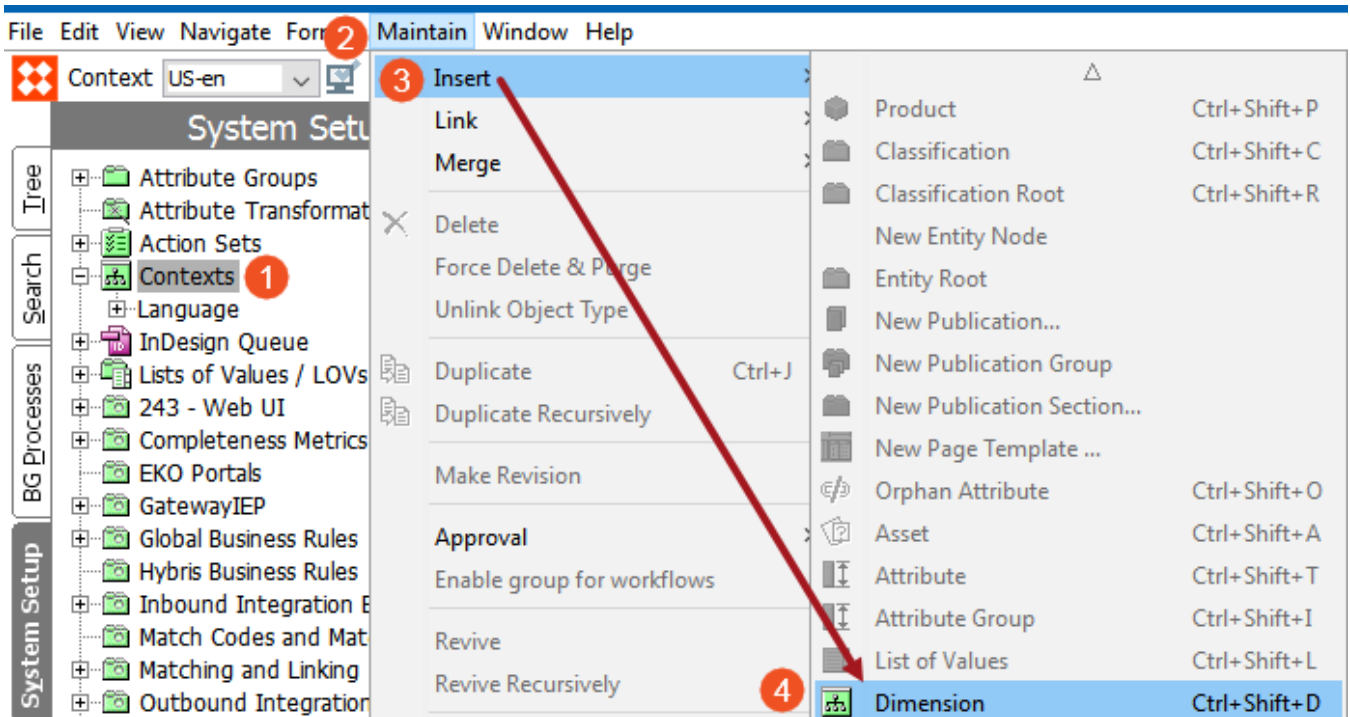
Creating a Dimension

If the need arises to create a new dimension, follow these steps. A new dimension may be needed if, for example, content that was once only language dependent also needs to become country dependent.

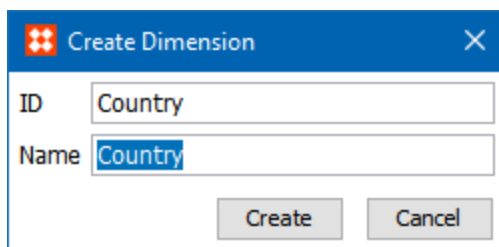
1. Navigate to System Setup > **Contexts**.
2. With Contexts selected, right-click and select **New Dimension**.



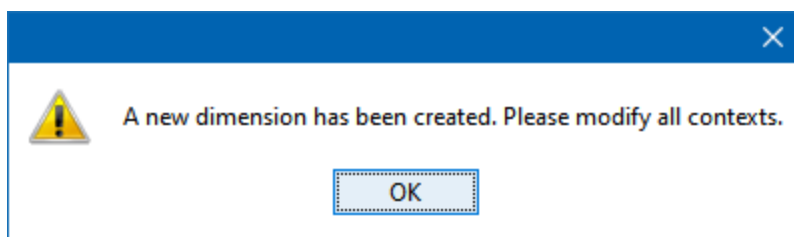
- Alternatively, select Contexts and go to Maintain > Insert > **Dimension**.



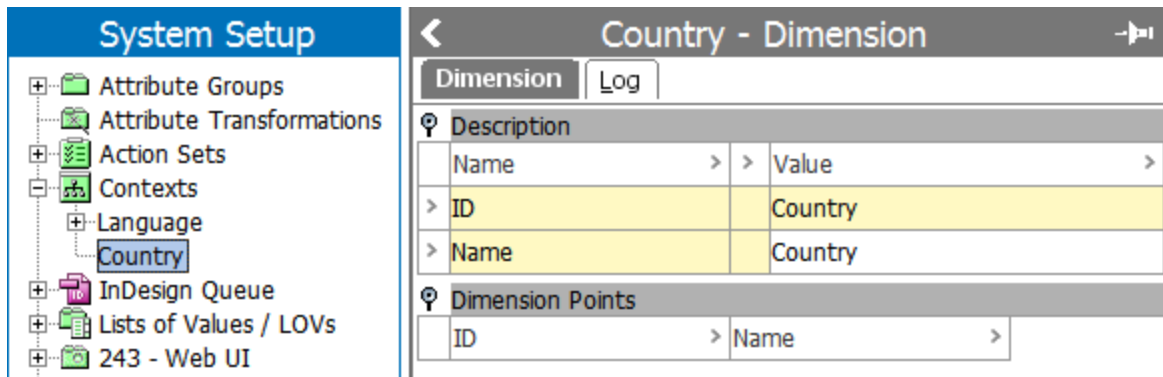
2. In the **Create Dimension** dialog that displays, enter an ID and Name for the new dimension, then click **Create**. If the ID and Name should be identical, press 'Enter' on the keyboard after typing the ID and it will repeat in the 'Name' field.



3. A dialog displays that says 'A new dimension has been created. Please modify all contexts.' Click **OK**.



4. The **Dimension** editor displays. At this point, no additional configuration is possible on the dimension until at least one **dimension point** is created below it.

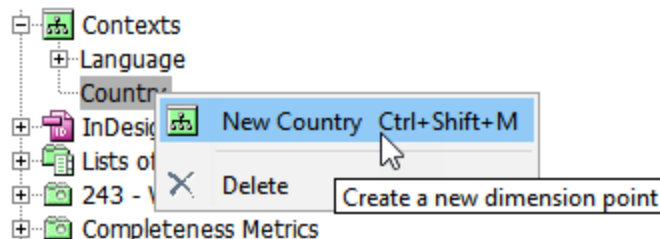


Creating a Dimension Point

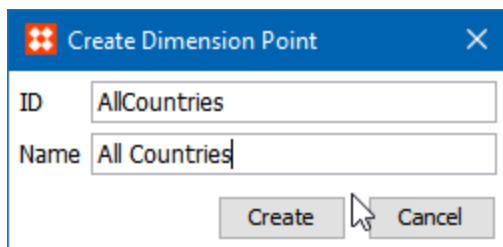
After a new dimension has been created, the next step is to create at least one dimension point below it.

1. Navigate to the relevant dimension in System Setup; in this example, Country.
2. Select the dimension, right-click, and then click **New** (for example, New Country).

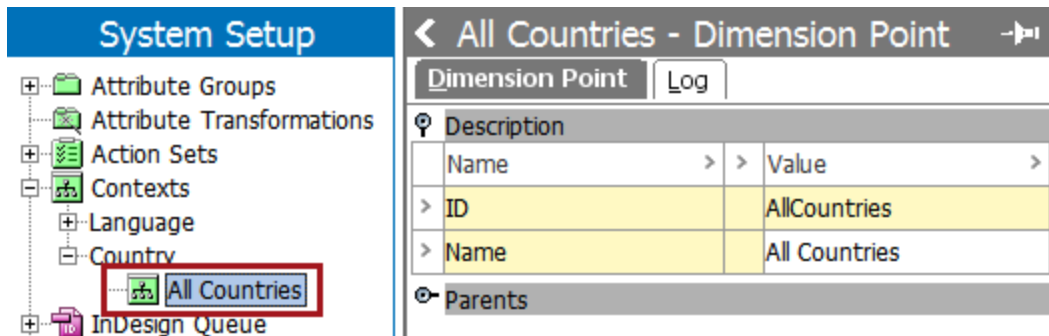
Note: The first dimension point created beneath a dimension must be a top-level **All** level, which is where *non*-dimension-dependent values are stored. Specific dimension points are created below the All level.



3. In the **Create Dimension Point** dialog that displays, type the ID and Name for the new dimension point, then click **Create**. If the ID and Name should be identical, press 'Enter' on the keyboard after typing the ID and it will repeat in the 'Name' field.

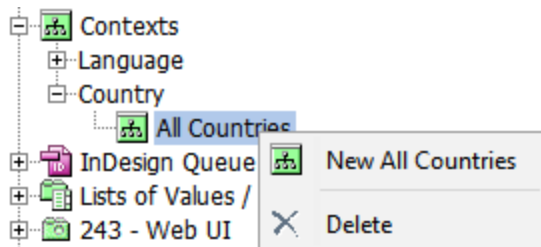


4. The new dimension point is created.

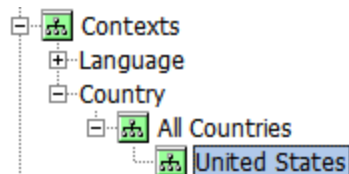


- To create additional dimension points below the All level, select the All level, right click, and click **New** (for example, New All Countries).

Note: These *specific* dimension points (*not* the All level) will be used when creating contexts.



- Follow steps 1-4 above to create the new dimension point.

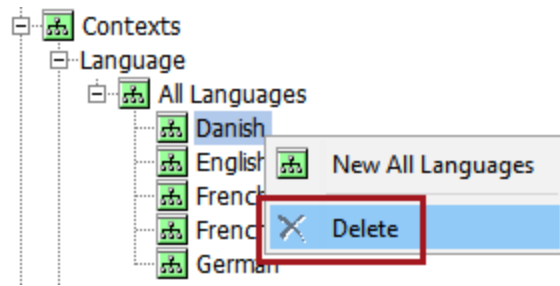


Deleting a Dimension Point

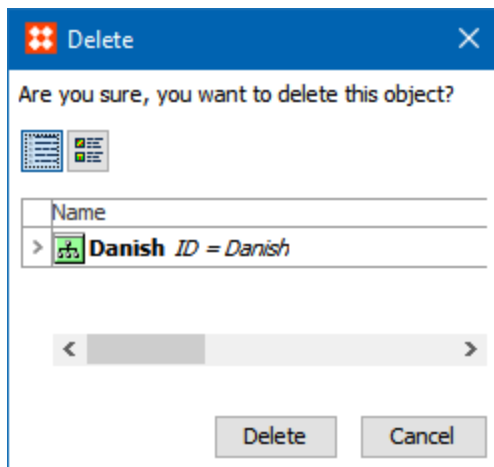
If you need to delete a dimension or dimension point, you must first delete all dimension points that exist below it.

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.

- Navigate to the dimension point that you would like to delete under System Setup > **Contexts**.
- With the dimension point selected, right-click and select **Delete**. Alternatively, select the dimension point and go to Maintain > **Delete**.



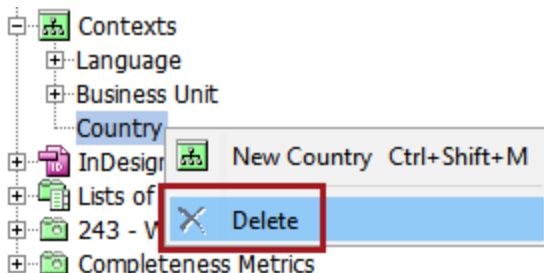
3. In the Delete dialog that displays, click **Delete** to remove the dimension point.



Deleting a Dimension

Once all dimension points have been deleted beneath a dimension, then the dimension can be deleted. To delete a dimension:

1. Navigate to the dimension that you would like to delete under System Setup > **Contexts**.
2. With the dimension selected, right-click and select **Delete**. Alternatively, select the dimension and go to Maintain > **Delete**.



3. In the Delete dialog that displays, click **Delete** to remove the dimension.

Selecting a Dictionary for a Language Dimension Point

A **dictionary** can be selected for each dimension point in the language dimension. The dictionary is used to check spelling when values are typed. 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.

1. In System Setup, navigate to Contexts > **Language**.
2. Navigate to the language dimension point under All Languages where a dictionary should be added.
3. On the **Dimension Point** tab, click inside the **Dictionary** field to access a dropdown list of dictionaries. tab, and then click the relevant dictionary from the list.

The screenshot shows the 'System Setup' sidebar on the left with a tree view containing 'Attribute Groups', 'Attribute Transformations', 'Action Sets', 'Contexts', 'Language', and 'Country'. Under 'Language', there is a sub-tree for 'All Languages' including 'Danish', 'Dutch', 'English', 'French', and 'German'. The 'English' item is selected. The main window is titled 'English - Dimension Point' and has a 'Dimension Point' tab selected. It displays a table with the following data:

Description	
Name	Value
ID	en-US
Name	English
Dictionary	English (United States)
GDSN mapping	English (United Kingdom)
Parents	English (United States)
Name	Français Classique (France)
	German (Germany)
std.lang.all	German (Germany) neu 08/2006 (frami)

Adding a Dimension Dependency After Loading Data

Though it is always recommended to have your dimension dependencies sorted out and in place before loading data, sometimes you may need to add a dimension dependency to an **attribute** or **reference type** after the fact.

Add a Dimension Dependency to an Attribute After Data Has Been Loaded

1. Temporarily create a context that has the 'All' level of the dimension you want to apply to the attribute.



ID	Name	Locale	Language
> All	All		All Languages
> Context6	Danish DK	Danish (Denmark) - da...	Danish

2. In the Export Manager, select all products in the database and map the attributes that you want to apply the dimension to.
3. Save the export configuration.
4. Save two copies of the exported Excel file.
5. In STEP, add the dimension dependency to the attribute(s).
6. If an attribute uses an LOV, see the **Using Dimension Dependent LOVs With Attributes** topic in the **List of Values (LOVs)** documentation for advice about setting them up.
7. In one copy of Excel, load the values back to the real working context (not the context that has the 'All' level).
8. In the second copy of Excel, replace all of the values with '[delete]' (no quotes)
9. Load the [delete] values to the temporary context that has the 'All' level of the dimension you applied to the attribute(s). This will remove the values.
10. Verify that the values have been removed from the 'All' level by re-running the export from the temporary context.
11. Verify that the values are now in the context with the dimension point you applied to the attribute.
12. It is recommended to make a small test of steps 8-10 before loading the entire file.

Add a Dimension Dependency to a Reference Type After Data Has Been Loaded

1. Temporarily create a context that has the 'All' level of the dimension you want to apply to the reference type.
2. In the Export Manager, select all products in the database and map the reference types that you want to apply the dimension to.
3. Save the export configuration.
4. Save two copies of the exported Excel file.
5. In one copy of Excel, load the values back to the real working context (not the context that has the 'All' level)
6. In the second copy of Excel, delete all values.
7. Import the file with deleted values to the temporary context that has the 'All' level of the dimension you applied to the reference type(s).
8. In the Advanced Settings (step 8) of the Import Manager, select the type of reference in the 'Remove Un-Mapped References' box.

Remove Un-Mapped References

☐ Classification References

Product Reference T... >
> Add Product Reference Type

Asset Reference Type >
> Add Asset Reference Type

Entity Reference Type >
> Add Entity Reference Type

Classification Product... >
> Add Classification Product Link Type

9. Verify that the values have been removed from the 'All' level by re-running the export from the temporary context.
10. Verify that the references are now in the context with the dimension point you applied to the reference type.
11. It is recommended to make a small test of steps 7-9 before loading the entire file.

Contexts

A **context** is a specific filter placed on data in STEP which groups a set of dimension points, allowing data to vary based on context. Each context is a combination of different dimensions, such as language and country. One and only one dimension point from each dimension can be associated with a specific context. In the below screenshot, the Language and Country columns contain the relevant dimension points.

System Setup

Attribute Groups

Attribute Transformations

Action Sets

Contexts

Language

All Languages

Country

All Countries

Lists of Values / LOVs

Global Business Rules

Inbound Integration Endpoints

Contexts

Log

ID	Name	Locale	Language	Country
> Context1	English US	English - en	English	USA
> Context4	French Belgium	French (Belgium) - fr_BE	English	USA
> Context3	French Canada	French (Canada) - fr_CA	French	Canada
> Context2	French France	French - fr	French	France
> Context5	Germany German	German - de	English	USA
> Add New Context				

When selecting a context from the **Context** dropdown list in the upper left of the STEP workbench interface, the product data being viewed will change and only data that is valid within the selected context will be displayed. The below screenshot shows a list of contexts available within this specific STEP instance and the view of French data.

If the data being viewed is a dimension-dependent STEP **name** and no name has been provided in the context you are viewing, the ID, in parentheses, will appear in the name field instead. For example, if the ID is 123456, the name will appear as (123456).

doc-dev - USER4@http://doc-dev:80 - STEPworkbench

File Edit View Navigate Format Maintain Window Help

Context: French France

French France

English US

Germany German

Israel Hebrew

Danish DK

English UK

French Belgium

French Canada

Kitchen

Party Supplies

Party Hats

Paper Hats

Balloon Hats

Paper Hats

Pet Hats

Assortiment de chapeaux de fête chien

Chapeau fête chouette rose & bleu

Rose et Chapeau de Fête Girafe Bleu

Assortiment de chapeaux de fête chien rev.0.9 - Product

Tables Category Profile Proof View Status State Log Tasks

Product Sub Products References Referenced By Images & Doc

(SalesItemMarketingDescriptions)

Description de l'objet de l'information

Name	Value
> Unité de Base de Mesure	CASE
> Description Source in German	abc
> Description Target	abc
> Couleur Primaire	Noir
> Couleur Secondaire	Bleu
> Brève Description Article	abc C'est l'anniversaire de votre chien - lever le « wouf »
> Exclusif?	Oui
> Longue Description de l'Article	abc Série de cinq. Violet, or, bleu, rose et vert. Spécifier ou grandes.
> Marketing Blurb	abc

Dimension Dependent Data

Data is stored in STEP in **dimension points**. In order to place data into different dimensions, the object type, attribute, and/or reference type that will either contain or link the data together must be made dimension dependent.

Note: Data is not stored in contexts.

- **Object types** are made dimension dependent if they should have different STEP **names** in different contexts. In most cases, this is because the names are translated into different languages, though they may also have different names in different markets, for example.

For information on making object types dimension dependent, see the **Maintaining Dimension Dependent Object Types** topic.

- **Attributes** are made dimension dependent when their **values** should be different in different contexts. This also includes lists of values (LOVs).

For information on making attributes dimension dependent, see the **Dimension Dependent Attributes** topic in the **Getting Started / STEP User Guide** documentation.

- **Reference types** are made dimension dependent when different linked objects (typically assets) should be visible in different contexts.

For information on making reference types dimension dependent, see the **Reference and Link Types** topic.

- **Assets** can be made dimension dependent but this is a global setting applied under System Setup > Users & Groups > **Image & Document Settings**. For more information, see the **Image and Document Settings** topic.

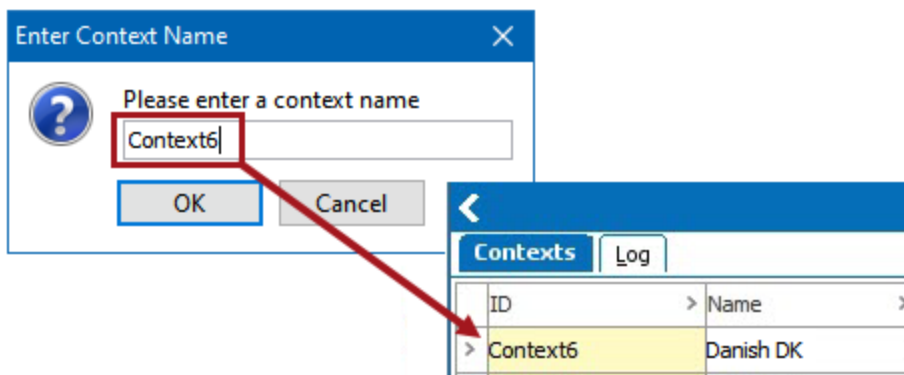
Note: In almost all circumstances, assets can be 'swapped out' in different dimensions by use of dimension-dependent reference types.

Context Best Practices

Though STEP is highly configurable with regard to the creation and maintenance of contexts, certain implementation standards apply to how they are created and implemented. This topic outlines some considerations to keep in mind when deciding how to configure your system to handle dimension-dependent data.

Context Do's

- Plan for future change—use IDs that will still make sense when situations change. The language + country will not make sense if you add another dimension later on.
- Use a generic ID and do not use the ID as the name. Change the name to be readable by users. Unlike most objects in STEP, there is no option to auto-generate Context IDs.



- Suggest IDs that are future safe. For example, if you only start out with a language and then a country dimension is added at some point in the future, 'EN' will no longer make sense. You could delete the EN context and create a new one (EN US), but that could mean that downstream systems that are receiving feeds may need to make adjustments to accept the new files, or, for print implementations, there may have been created hundreds and hundreds of InDesign pages created that have the context ID stored in the page files.

Context Don'ts

- Do not use a **global** context. If one already exists on your system, *delete it*. Nobody should work in global, which results in unnecessarily inherited values. See the **Dimension Dependent Attributes** topic for more information on value inheritance in dimension-dependent attributes.

A global context is where the All Languages + All Countries dimensions would be used. Nobody speaks All and nobody lives in All, so no one should maintain data there.

Problems that could be caused by using a global context include:

- A user might log in to a global context to create some attributes. They also do some sort of data maintenance in STEP. They may forget to switch contexts before entering language-dependent data in English and, if so, they have stored that information in the 'All' level instead of the 'English' level. That is not where it belongs, and it should be moved.
- A user may forget that they are in the global context and exports data to send to someone. This file would not contain any of the language- or country-dependent data, so the file is incomplete.
- Do not create a context that will not be used right away. Create only the combination of *dimensions* that you need.
- Do not use an **All** level in a context. Pick a real language and a real country.
- Avoid asking users to work in multiple contexts. There is a very high risk that someone will accidentally enter content in the wrong context, and it is very difficult to find these situations and correct them.

Maintaining Contexts

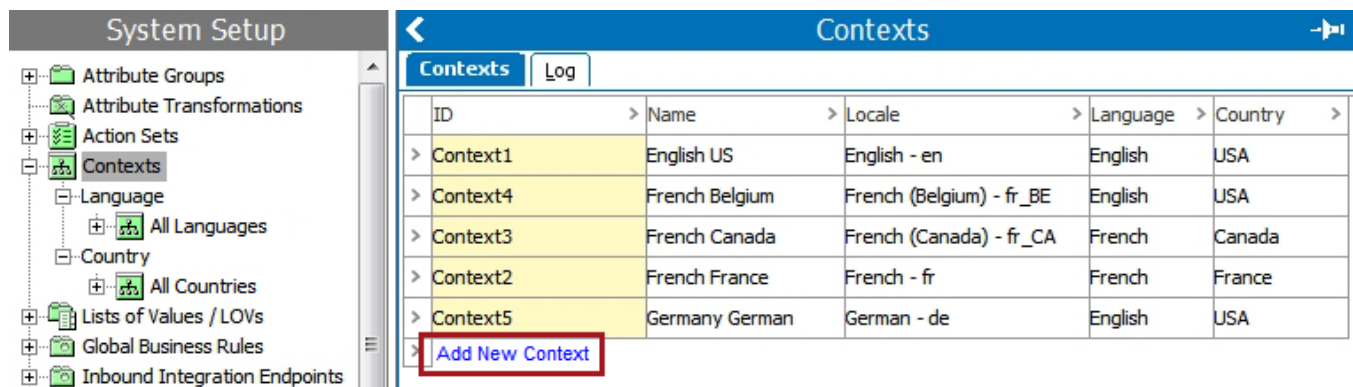
This section describes how to create, edit, and delete contexts. Also explained is the concept of hidden values, which can be caused when a context is deleted, and how to specify a default context.

Creating a Context

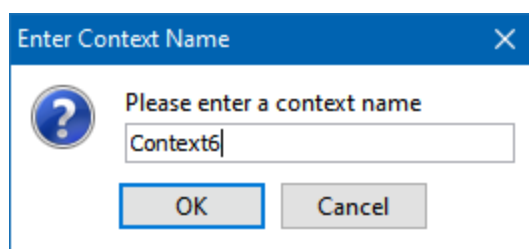
Contexts are maintained in System Setup > **Contexts**. A context must contain at minimum one language dimension point; for example, English.

Note: Before creating a context, the associated dimension and dimension points must be created. For more information, see the **Maintaining Dimensions and Dimension Points** topic.

1. Navigate to System Setup > **Contexts** to display the Contexts editor, then click on **Add New Context**.



2. In the **Enter Context Name** dialog that displays, enter a name for the context, then click **OK**. By default, the name entered in this field also becomes the context ID. See the **Context Best Practices** topic in this documentation for suggestions on what ID format to use when naming your contexts.



3. A new 'default' context appears with a blank locale and English as the language. If other dimensions exist, such as Country, a 'default' value will also appear in that column, such as 'All Countries' or 'USA.'
4. In the **Name** column, replace the auto-created name with a new name for the context, which should be different than the ID. See the **Context Best Practices** topic in this documentation for suggestions on how to name your contexts.

- 5. Select the relevant dimension point from the dropdown list for each dimension, for example, Language and Country.

Contexts

Log

ID	Name	Locale	Language	Country
> Context6	Danish DK		English	USA
> Context1	English US	English - en	Danish	USA
> Context4	French Belgium	French (Belgium) - fr_BE	Dutch	USA
> Context3	French Canada	French (Canada) - fr_CA	English	Canada
> Context2	French France	French - fr	French	France
> Context5	Germany German	German - de	German	USA
Language Root				
English				
> Add New Context				

- 6. Click the blank field in the **Locale** column, then click the ellipsis button (...) to select the locale that will be used for data in the context.
- 7. In the **Select Locale** dialog that displays, choose the relevant locale from the dropdown list, then click **OK**.

Select Locale

Danish (Denmark) - da_DK

▼

Locale info

Danish (Denmark) - da_DK

-1.234,56

13-05-2009

13-05-2009 13:10:48

OK

Cancel

- 8. The locale now displays in the Locale column. For more information on locales and how they are used in contexts, see the **Context Locales** section of this documentation.

Contexts				
Contexts Log				
ID	Name	Locale	Language	Country
> Context6	Danish DK	Danish (Denmark) - da_DK	Danish	Denmark

- The context has been created and will be available in the dropdown list of available contexts in the upper left corner of the STEP Workbench GUI as well as in the context menu in the Web UI.

Editing a Context

When editing a context, the following options are available:

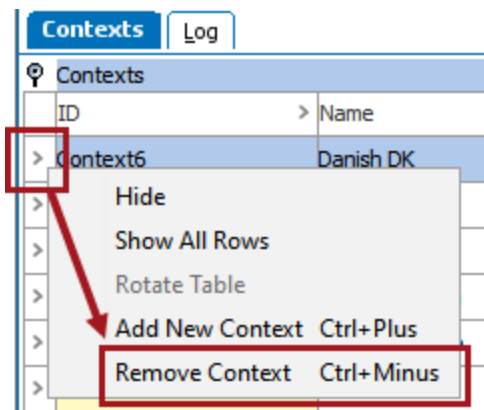
- Enter a new name in the Name field
- Choose a new locale
- Select a new dimension point from the dropdown menu in one of the dimension columns (such as Language or Country)

The ID cannot be edited; a new context must be created to generate a new ID.

Deleting a Context

To delete a context:

- Locate the context that you want to delete, then right-click on the arrow directly to the left of the context ID.

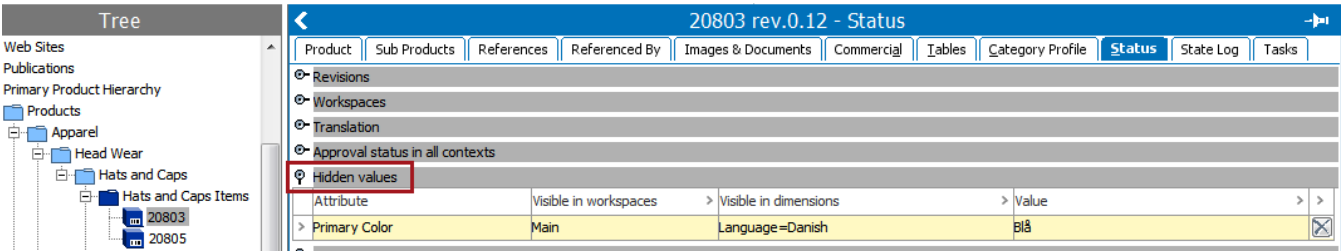


- Click **Remove Context**, or press Ctrl + Minus on your keyboard
- The context is deleted.

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.



Within a standard editor, values inherited from another dimension will have a purple indicator in the attribute icon, as shown below:

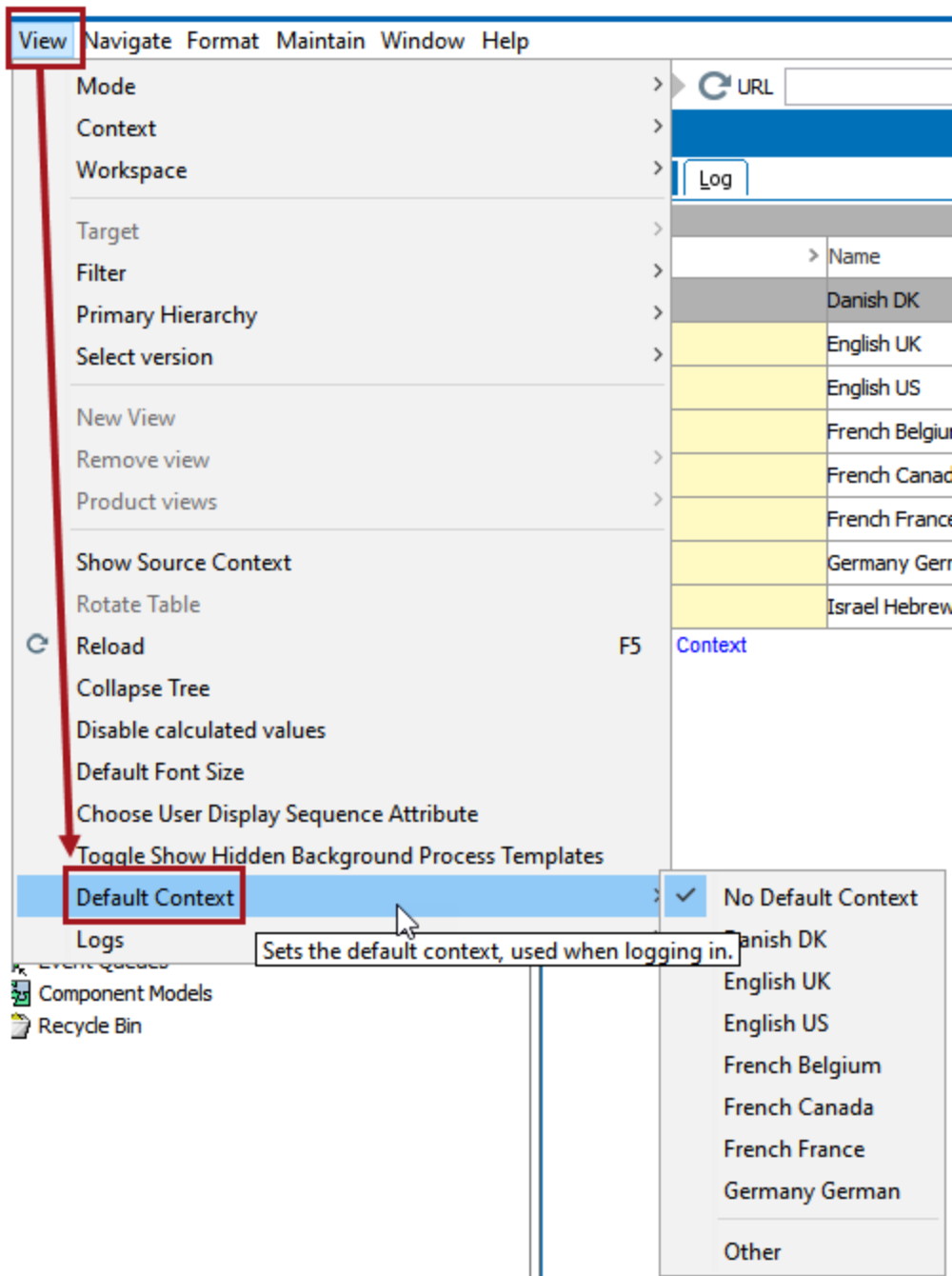
Item Description Information		
Name		Value
> Base Unit of Measure		
> Product Name	abc	Baseball Cap
> Description Source in English	abc	Text, Inherited from other context
> Description Target	abc	

Note: Deleting a context can also cause a product to not be fully approved even if it is approved in all contexts. This is because there are still existing values in a deleted context.

Specifying a Default Context

To ensure that the STEP Workbench always loads in your preferred context, you can specify a default context by following these steps.

1. Navigate to View > **Default Context**.

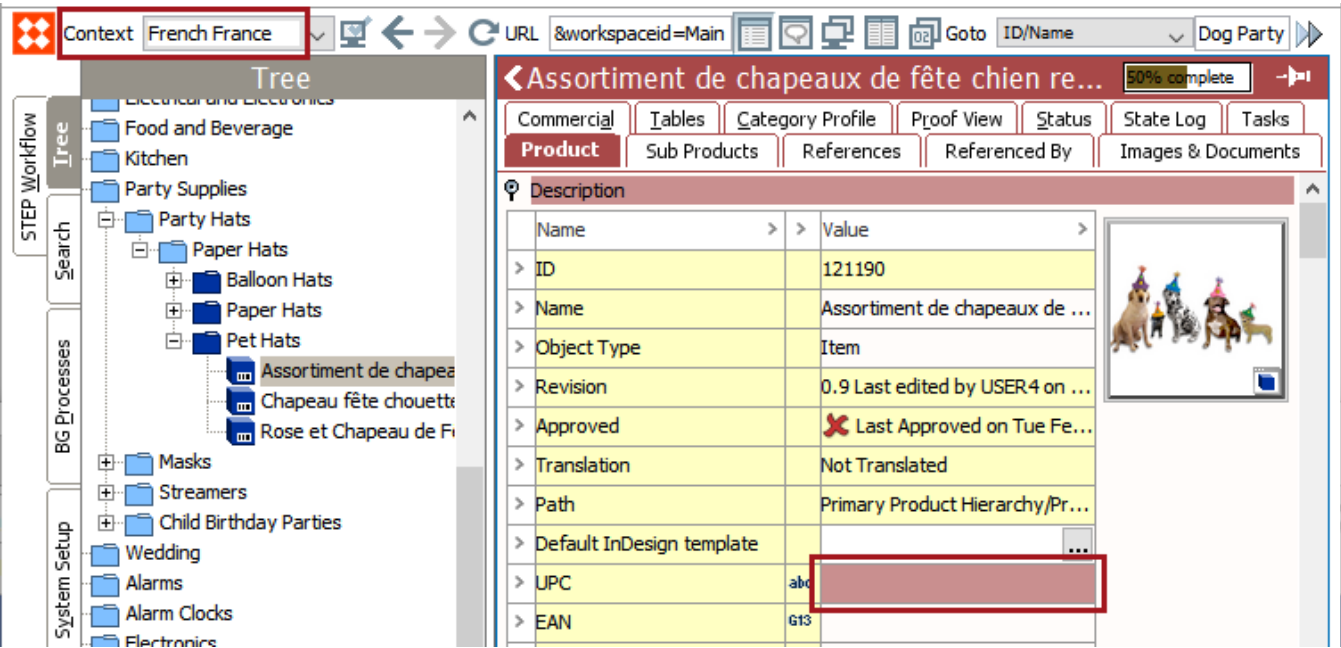


2. Select the preferred context from the list of contexts that displays. If you do not see your context, click **Other**.
3. Choose your preferred context from the list in the 'Other Contexts' dialog, then click **OK**.



Going forward, STEP will always open in this context unless you set the default context to something else or choose 'No Default Context.'

You can still work in other contexts, but if you select a context that is not the default context, the toolbar turns dark pink to indicate that you are not working in the default context. Additionally, attribute editor fields will turn pink when selected.

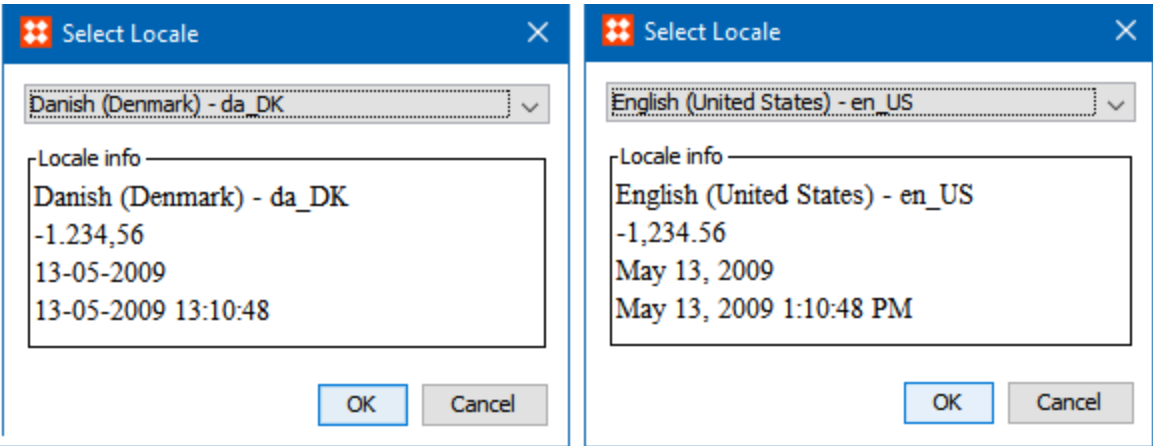


Context Locales

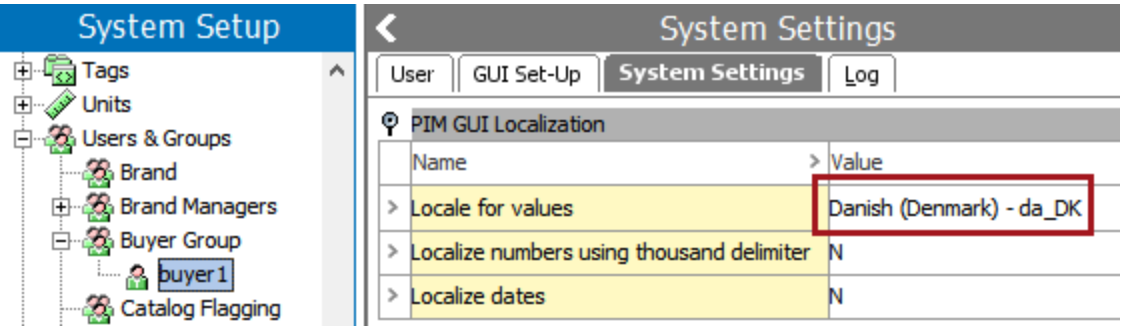
A locale is a part of a context. Locales control the display for users in the workbench, as well as Web UI buttons, tabs, and so forth.

Note: Locale settings on the workbench should not be confused with Web UI locale. The Web UI locale allows users to select a locale when logging in and, if there is a localized version of a Web UI, they will see the localized screens. For more information on Web UI locale, see the **Localization** topic in the **Administration Portal** documentation.

Locales control the decimal separator in attributes that are numbers, fractions, and embedded numbers. Additionally, locales control the appearance of dates. Attributes with the validation of numeric text are really text attributes and cannot be accommodated with the locale functionality.



Locales can be set on individual users in the 'Locale for values' field on the System Settings tab, located under System Setup > **Users & Groups**. Setting a locale for a user controls how the user sees data in the workbench, regardless of what context they are working in.



For more information on using locale with users in System Setup > System Settings, see the **GUI Localization** topic in the **System Setup / STEP Super User** documentation.

Locales are also used in attribute transformations, STEP table transformations, and in STEP'n'design.

Locale Do's

Although the locale can be left blank, it is recommended is to set the locale for the country in each context.

Contexts

Contexts

Log

ID	Name	Locale	Language
> Context6	Danish DK	Danish (Denmark) - da_DK	Danish
> Context7	English UK	English	
> Context1	English US	English	
> Context4	French Belgium	French	
> Context3	French Canada	French	
> Context2	French France	French	
> Context5	Germany German	German	
> GL	GL		
> Context8	Israel Hebrew	Hebrew	

Add New Context

Select Locale

Danish (Denmark) - da_DK

Locale info

Danish (Denmark) - da_DK

-1.234,56

13-05-2009

13-05-2009 13:10:48

OK

Cancel

Events

Within STEP, an event indicates that data has been changed. Events can be monitored and used to kick-off additional processing via a business rule or an event processor, or to generate an export via an OIEP. Events are generated based on two factors: database transactions and revisability.

Database Transactions

Each time an update is made in STEP, a transaction is used to write the changed data to the STEP database. The method used to make the change is a factor in determining if each transaction generates its own event, or if multiple transactions are collected to create a single event. The available methods for changing STEP data are:

- **STEP Workbench** generates a transaction each time a field is changed, since no Save button is available or required.
- **Web UI** generates a transaction when the Save button is clicked, meaning several changes may be included in a single transaction.
- **Import** generates a transaction on the success of the import, and may include several changes in a single transaction.

Revisability

Revisability determines when it is possible to include different values per workspace and affects the generation of events as follows:

- **Global revisable** objects do not require approval (values are the same for all workspaces) and an event is generated when the database transaction is written. Workflows, business rules, integration endpoints, event processors, and all publication structures are always globally revised. Entities can be set to be globally revisable.
- **Workspace revisable** objects do require approval (values can differ per workspace) and an event is generated when the object is approved. Products, classifications, and assets are always workspace revised. Entities can be set to be workspace revisable.

For more information on revisability of entities, see the **Revisability on Entity Object Type** section of the **System Setup / Super User Guide** documentation.

Note: It is important to note that the event system cannot be used as a transactional request / response integration with other applications. Events are not reflected by transaction in the exported data (which would, in many cases, cause an overflow of data). For more information, see **Change Flags for Events** below.

Generating Events

For many system actions an event is automatically generated based on the method used to change STEP data and the Revisability parameter on the object type. This information impacts when an OIEP or an Event Processor is triggered, and the rules are outlined below.

For a **global revisable** object, the transaction and event are created, based on the change method, as follows:

	Workbench	Web UI	Import
Transaction and Event	Change to a single field	Click the Save button	Successful Import

For a **workspace revisable** object, the transaction and event are created, based on the change method, as follows:

	Workbench	Web UI	Import
Transaction	Change to a single field	Click the Save button	Successful Import
Event	Approve change	Approve change	Approve change

Considering the event generation rules, for example, changing data in Web UI on an entity object that is workspace revisable would not trigger an OIEP export until that object is approved. However, changing data in Web UI on a similar entity object that is global revisable would trigger the OIEP as soon as the change was saved. Note that both scenarios assume that an event-based OIEP exists that is set up listen for appropriate events on the applicable object types. For more information, see the **Core Events** section.

Additionally, events can be generated in a manual way based on some system action or setup. For example, business rules, a post-processor on an OIEP, a derived event, and some event handlers can cause events to be generated. For more information, see the **Derived Events** section.

For an example of manually generating events on demand, see the **Generating Match Codes and Running a Matching Algorithm** section of the **Matching and Linking** documentation.

Core Events

You can use different output templates (export configurations) within the OIEP framework to export data depending on the event type (create, modify, or delete for core events) and for the type of object for which the event is generated. This means that with the core events, you can use different formats for different types of objects when a create, modify, or delete event is generated.

Except for the Republish event, core events are all generated automatically by STEP when certain actions are performed.

Core events are always tied to an object in STEP, meaning an event can include data about its creation time, the object for which the event was generated, and the event type.

Core Event	Event Type	Event Is Generated When
APPROVECREATED	Create	A revisable object is approved for the first time.
APPROVEMODIFIED	Modify	A revisable object already in Approved state is approved again.
DELETEAPPROVAL	Delete	The deletion of a revisable object in the recycle bin is approved .
GLOBALATTRIBUTECHANGE	Modify	The value for an Externally Maintained Attribute is changed.
GLOBALEVENT	Create	One of these System Setup objects is created: attribute, attribute group, list of values group, list of values, unit group, unit, product to classification link type, and reference type objects.
GLOBALEVENT	Delete	One of these System Setup objects is deleted: attribute, attribute group, list of values group, list of values, unit group, unit, product to classification link type, and reference type objects.
GLOBALEVENT	Modify	One of these System Setup objects is modified: attribute, attribute group, list of values group, list of values, unit group, unit, product to classification link type, and reference type objects.

Core Event	Event Type	Event Is Generated When
GLOBALREFERENCETYPECHANGE	Modify	An Externally Maintained Reference or Link is modified.
GLOBALTERMCHANGE	Modify	A Terms List is modified.
REPUBLISH	Create	An on-demand, manual command is sent using the 'Send Republish Event' operation on a business rule or from bulk update, or the 'Republish' command is used for a Collection. For more on Republish events, see Event-Based OIEP Forward, Rewind, Purge, and Republish .
REVIVED	Create	A revisable object is revived from the recycle bin.

To create multiple output templates under conditions other than core event conditions, see the **Derived Events** section.

Derived Events

When the trigger you need is not available as a core event, use a derived event, which is an event generated by a JavaScript-based business rule. Another case for derived events is when you want to use different output templates under conditions different from the core conditions. For example, when publishing data to a website (if the website categories are modeled using a classification in STEP) you may want to send files formatted in different ways or include different data when a product is first linked into a category, when a product linked to a category is modified, and when a product is unlinked from a category.

This topic addresses:

- Using a Derived Event
- Using a Derived Event Type in an OIEP
- Adding a Derived Event Type

Prerequisites

A thorough understanding of business rules is needed to create successful derived events. For more information, see the **Business Rules** section of the **Business Rules** documentation.

Using a Derived Event

Although core event functionality can be used to publish data to a downstream system when an object is created or updated via an import, or when an object reaches a specific state in a workflow, these situations will not normally trigger any of the core events. Since an event-based OIEP cannot discriminate the republish event from other types of events, it would also be registering events and publishing data, for example, based on an object being approved for the first time.

You may also want to generate derived events on referenced images, referenced products, parent product, or on products that are linked, removed, or modified in a specific classification structure.

When triggering object types are set up on an event-based OIEP, business rules can be applied to generate derived events, based on the original object type triggering. Derived events are triggered based on business rules, and can be set up depending on the need.

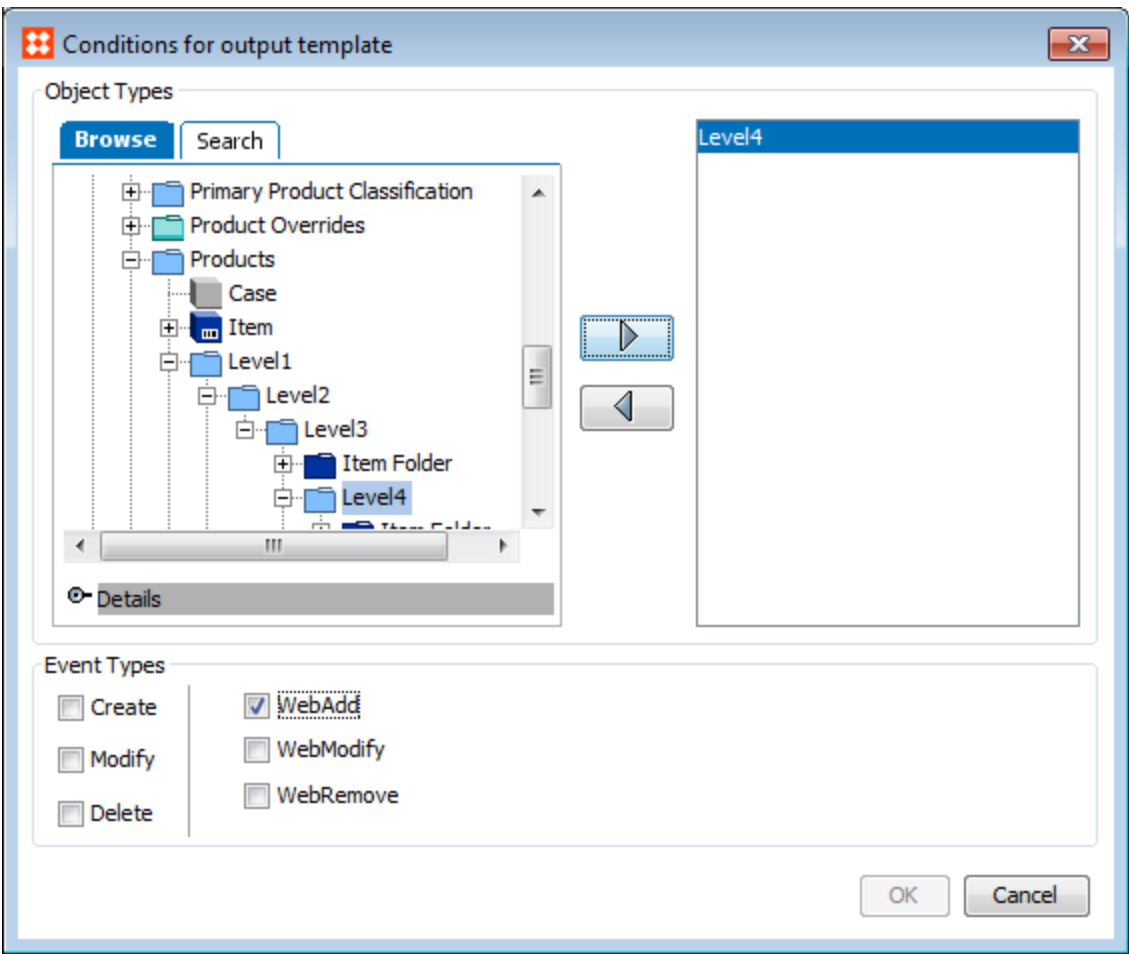
For more information about using derived events, see the **Event-Based OIEP Example Business Rules for Derived Events** section of the **Data Exchange** documentation.

Using a Derived Event Type in an OIEP

Derived event types can be added to an OIEP output template, and from within a business condition to filter objects that trigger the OIEP. For example, consider the following derived event types:

- WebAdd
- WebModify

- WebRemove
1. In an event-based OIEP , Configuration > Output Templates flipper > click the **Add configuration** link.
 2. On the **Conditions for Output Template** dialog, select the object type and the derived event type WebAdd.



3. Repeat the previous steps to create an additional output template for the WebRemove and WebModify derived events.

Output Templates				
Object-Eventtype	Format	Pre-Processor	Post-Processor	
> Level4 (WebAdd)	Excel (3 mappings)	None	None	
> Level4 (WebModify)	Excel (3 mappings)	None	None	
> Level4 (WebRemove)	Excel (3 mappings)	None	None	
> Add configuration				

In this example, the format for all output templates is Excel but the mapping differs in each template. It is also possible to specify a different output format for each output template, such as Generic XML, STEPXML, or CSV, depending on the specific requirements.

Adding a Derived Event Type

Before using a derived event, create a derived event type.

1. In System Setup, click **Derived Events**.
2. On the Derived Events tab, click the **Add Derived Event Type** link.
3. Type an ID and click the **Add** button. The derived event type is now available for use in an output template for an endpoint.

Event Processors

An event processor includes a queue of generated events and an engine for processing the events. For information about how and when events are generated, see the **Events** section of the **System Setup / STEP Super User Guide** documentation.

An event processor has no delivery method, but is otherwise similar to an event-based outbound integration endpoint (OIEP). Event processors can be used alone when the results are only required internally to STEP. However, if the delivery of the results of the event processor is required externally, the event processor can be used in conjunction with an OIEP. For more information, see the **Creating an Event-Based Outbound Integration Endpoint** documentation.

As described in the **Configuring an Event Processors** topic referenced below, event processors can be used to automatically perform actions based on triggering events, including:

- Managing revisions on objects via the **Revision Management** processor.
- Propagating updated LOV values to the objects that use the LOV via the **Expand LOV Value Changes** processor.
- Caching assets via the **Image Cache** processor.
- Generating new match code values and/or running a matching algorithm via the **Matching** processor.

Configuring an Event Processor

To configure an event processor, complete the following steps:

1. Perform one-time setup tasks to allow for the creation of event processors as described in **Setting Up for Event Processors**.
2. Create an event processor as described in **Creating Event Processors**.
3. Enable the event processor as described in **Enabling Event Processors**.

Important: If an event processor is being used in the process of moving data outside of STEP, an OIEP is required as a delivery method.

Initial Setup for Event Processors

Before creating an event processor, one or more setup groups must be created that are allowed to hold event processors. You must also specify the setup group(s) in which event processors can be created. This setup only needs to be performed once, and most systems will already have it completed.

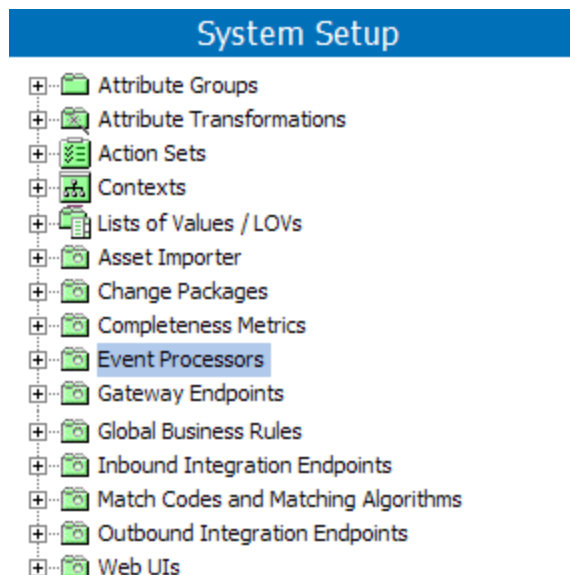
Each of the steps necessary to setup for event processors are described below.

- Create setup group types for event processors.
- Link event processor object types to setup group types.
- Create an event processor setup group.
- Configure action sets and privileges.

Prerequisites

Only users with the relevant privileges can view or maintain event processors. For detailed information, see the **Action Sets** section and the **Users and Groups** section in the **System Setup / STEP Super User** documentation.

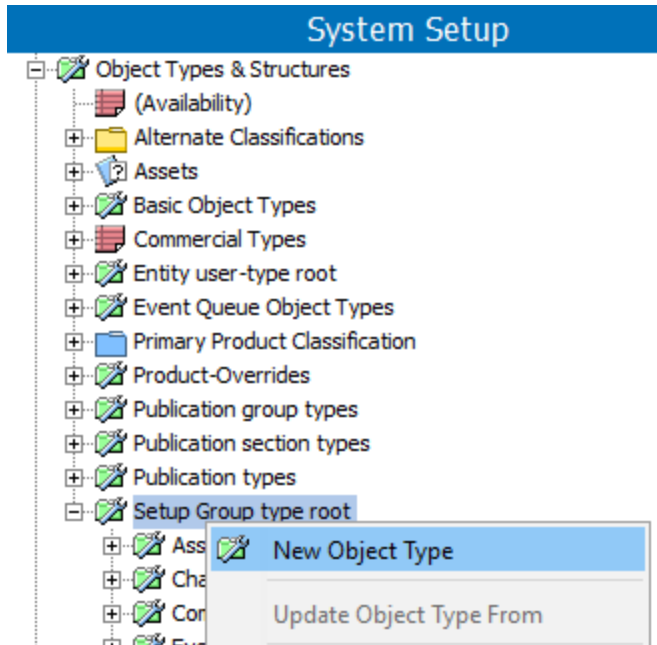
Review your System Setup tab to determine if one or more event processor nodes already exist. The name of the node on your system is not required to match the one in the image below.



Once the setup has been completed, the steps in this section are only needed if additional levels of organization are desired.

Create Setup Group Types for Event Processors

1. Go to System Setup > Object Types & Structures > Right-click **Setup Group type root**.
2. Click **New Object Type**.



3. Enter an **ID**.
4. Enter a **Name**.
5. Select any required Dimension Dependencies.
6. Click **Create**.

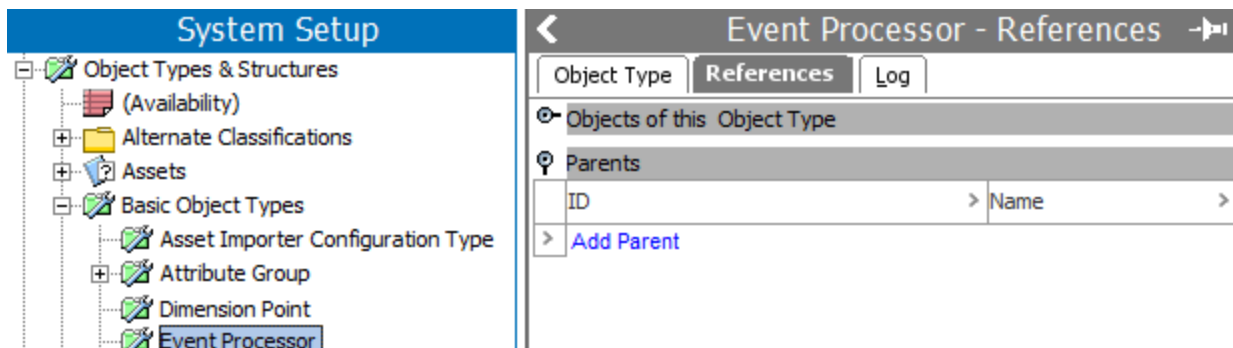
The screenshot shows the 'Create Object Type' dialog box. It has a title bar with the Stibo logo and a close button. The dialog contains three main sections: 'ID' with a text input field, 'Name' with a text input field, and 'Dimension Dependency' with two checkboxes: 'Country' and 'Language'. At the bottom, there are two buttons: 'Create' and 'Cancel'. The 'Create' button is highlighted in a darker shade.

The new setup group appears in System Setup under 'Object Types & Structures' as a child in the Setup Group type root.

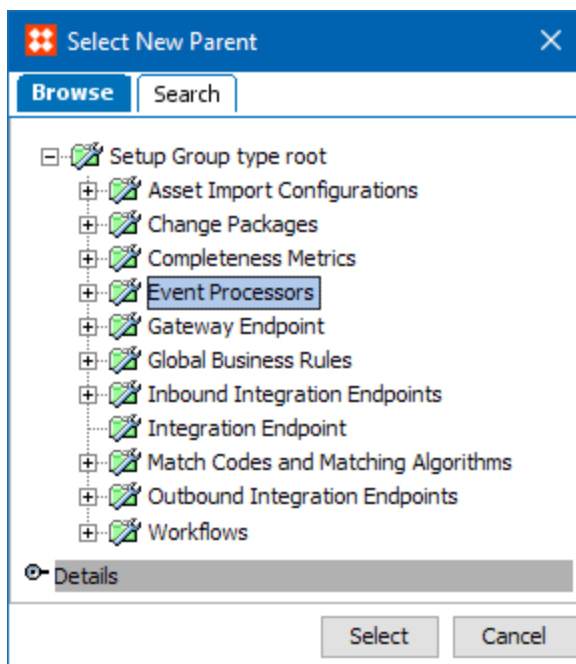
Link Event Processor Object Types to Setup Group Types

Linking determines the object types that can be displayed at each level of a hierarchy.

1. Go to System Setup> Object Types & Structures > **Basic Object Types**.
2. Select **your event processor object type** to display the editor.



3. Click the **References** tab.
4. Open the **Parents** flipper.
5. Click the **Add Parent** link, and the Select New Parent dialog will display.



6. Browse or search to select **the relevant setup group type**.
7. Click **Select**.

For example, in this image of an event processors setup group type hierarchy, the main setup group type named 'Event Processors' contains:

- A setup group type named 'Matching'
- Itself as a parent so that additional event processors group types can be added below the main level

The screenshot shows the 'System Setup' hierarchy on the left with 'Event Processors' selected. The right pane, titled 'Event Processors - References', has the 'References' tab active. It displays a table of references for the 'EventProcessors' object type.

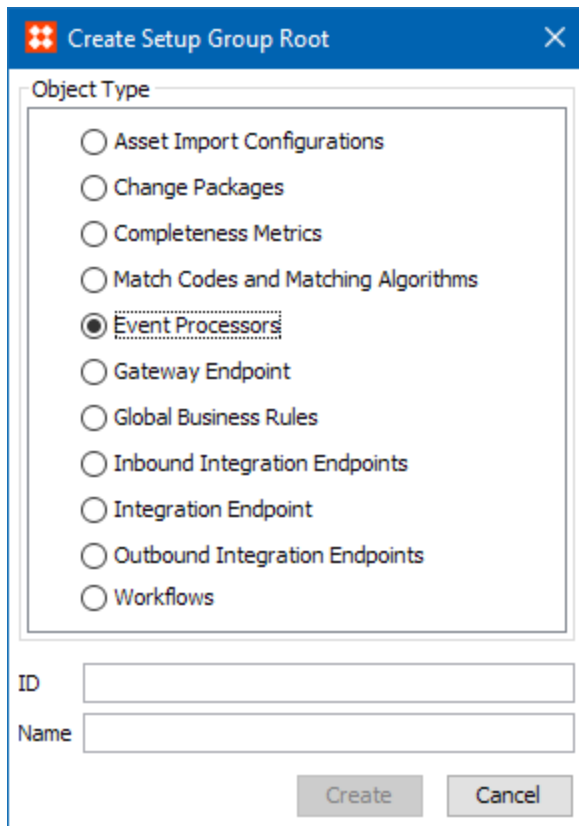
ID	Name
EventProcessors	Event Processors
Setup Group user-type root	Setup Group type root

An 'Add Parent' link is visible at the bottom of the table.

Create an Event Processor Setup Group

Creating a setup group allows your event processor setup group type to appear as a node in the System Setup hierarchy.

1. Go to System Setup > Select **any object in the hierarchy**.
2. On the menu bar, Select **Maintain > Insert > Setup Group Root**, and the Create Setup Group Root dialog will display.



The dialog box is titled "Create Setup Group Root" and contains a list of object types. The "Event Processors" option is selected. Below the list are input fields for "ID" and "Name", and "Create" and "Cancel" buttons.

Create Setup Group Root

Object Type

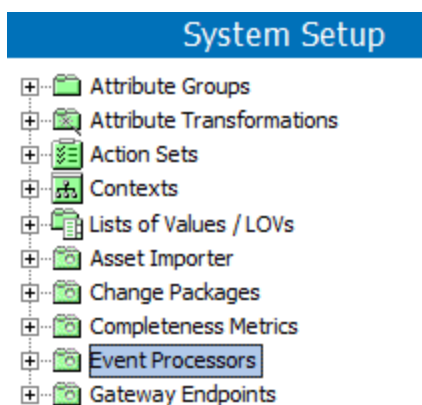
- ☐ Asset Import Configurations
- ☐ Change Packages
- ☐ Completeness Metrics
- ☐ Match Codes and Matching Algorithms
- ☒ Event Processors
- ☐ Gateway Endpoint
- ☐ Global Business Rules
- ☐ Inbound Integration Endpoints
- ☐ Integration Endpoint
- ☐ Outbound Integration Endpoints
- ☐ Workflows

ID

Name

3. Select **your event processors object type**.
4. Enter an **ID**.
5. Enter a **Name**.
6. Click **Create**.

Your setup group is created and appears as a node in the System Setup hierarchy. The name of the setup group on your system is not required to match the one in the image below.



Configure Action Sets and Privileges

When you configure 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 actions the members of the user group are allowed to perform on the event processors.

In some situations, it can be useful to create a 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. Go to System Setup > Action Sets > Select **Setup Actions**.
2. Right-click **Setup Actions**.
3. Click **New Action Set**, and the Create Action Set dialog will display.
4. Enter an **ID**.
5. Enter a **Name**.
6. Click **Create**.



7. Go to the new Action Set > Action Set tab > Click **Add Action**, and the Select Action dialog will display.
8. Select one or more of the listed actions that are relevant for event processors that you want to be part of this action set.
9. Click **Select**.

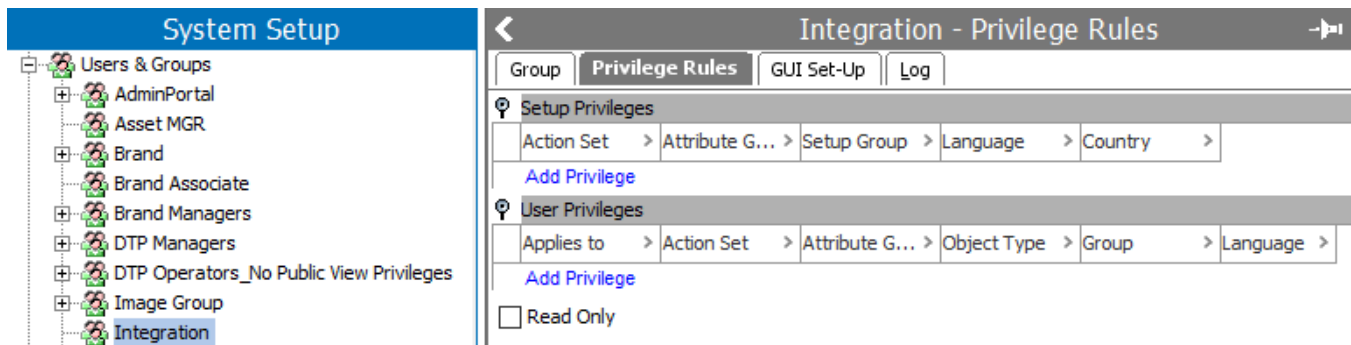
Some commonly used Actions are described below:

- 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.

Configure Event Processor Privileges

1. Go to System Setup > Users & Groups > select the relevant user group that will have the new event processor action(s).

- Click the **Privilege Rules** tab.



- Under the Setup Privileges flipper, click the **Add Privilege** link to display the Add Privilege Rule dialog.

The 'Add Privilege Rule' dialog box is shown. It has a title bar with a red icon and a close button. The fields are as follows:

- Action Set:** View Event Processors (ViewEventProcessors) with an ellipsis button.
- Attribute Group:** Empty field with an ellipsis button.
- Setup Group:** Event Processors (EventProcessors) with an ellipsis button.
- Valid for Object Type:** Empty field with an ellipsis button.
- Apply to Group:** Integration (Integration) with an ellipsis button.
- Dimensions:**
 - Language:** <ANY> dropdown menu.
 - Country:** <ANY> dropdown menu.

 At the bottom are 'Save' and 'Cancel' buttons.

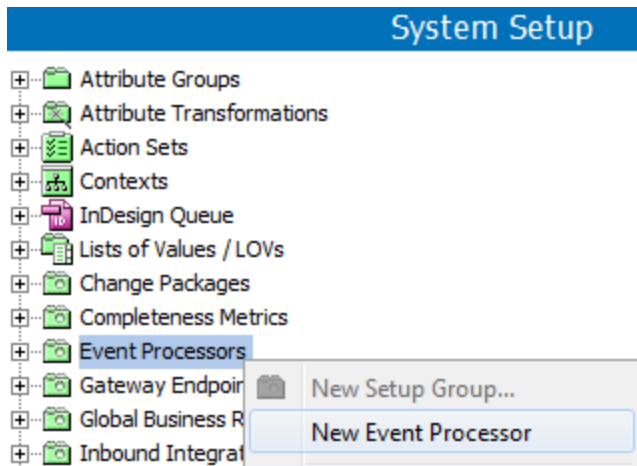
- In Action Set, click the ellipsis button (...), and then select the relevant event processor action set.
- In Setup Group, click the ellipsis button (...), and then select the relevant event processor setup group.
- Click **Save**.

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. Go to **System Setup** > Right-click the **Event Processors** setup group.



2. Select **New Event Processor**, and the Event Processor Wizard will display step '1. Identify Event Processor.'

The screenshot shows the 'Event Processor Wizard' dialog box. The 'Identify Event Processor' step is active, showing fields for Event Processor ID, Event Processor Name, and Object type (set to Event Processor). The 'Next' button is highlighted.

3. Enter an **ID**.
4. Enter a **Name**.
5. Confirm the object type parameter is set to the default **Event Processor**.
6. Click **Next**, and the Event Processor Wizard will display step '2. Configure Event Processor.'

Event Processor Wizard

Steps

1. Identify Event Processor
- 2. Configure Event Processor**
3. Configure Processing Plugin
4. Schedule Event Processor
5. Configure Error Reporter Processing Plugin

Configure Event Processor

User running event processor plugin	user (USER) ...
Number of events to batch	1000
Days to retain events	0
Queue for event processor	EVPROC
Maximum number of old processes	100
Maximum age of old processes in hours	168
Limit of lines in execution report	1000
Select Processor	Matching
Select Error Reporter	Mail Error Report

Back Next Finish Cancel

7. 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.
 - **User running event processor plugin:** To search or browse for a user, click the ellipsis button (...). The privileges of the user selected must include the objects being processed since the selected user's privileges are applied to the event processor. Common setup is to create a separate user for each event processor to allow tracking of actions taken by each 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 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. Below are details for configuring each of the 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.
 - **Image Cache** allows the event processor to generate images for required conversions asynchronously and saves the resulting files to the image cache. Common setup is for all STEP systems to use this event processor as a means of optimizing asset export time to external systems. Only image conversion configurations with the 'Cache generated images' option checked are handled by this event processor. It is recommended that all systems using image conversions have imaging caching processors set up. For more information, see the **Image Conversion Configurations** section of the **Export Manager** documentation.
 - **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 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. It is strongly recommended that all STEP systems have this event processor set up as a means of managing the STEP database size.
 - **Select Error Reporter:** Choose either Do nothing or Mail Error Report. If error reporting is selected, additional information must be supplied in step 5 of the wizard.
8. Once any required changes are made, click **Next**, and the Event Processor Wizard will display step '3. Configure Processing Plugin.'

Important: The options displayed within the Event Processor Wizard step '3. Configure Processing Plugin' will vary depending on the processor selected in step '2. Configure Event Processor.'

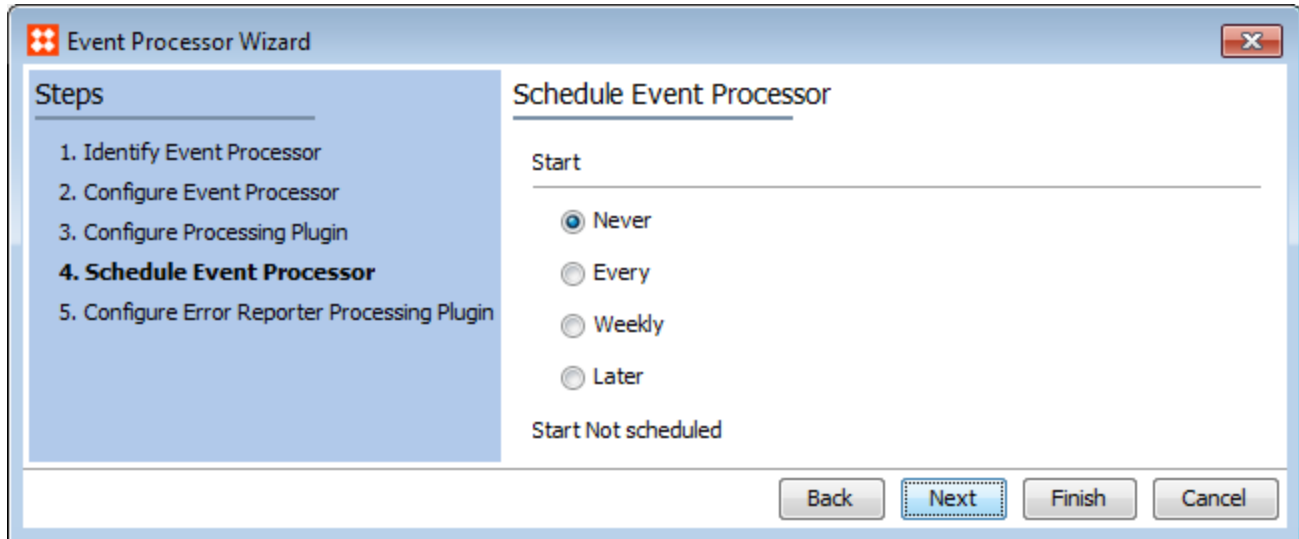
9. Configure the processing plugin if required.

Below are details for configuring each of the standard core processors:

- For the **Expand LOV Value Changes** processor, no further configuration is required. However, considerations for the event triggering definitions are provided in the **Expand LOV Value Changes Processor Parameters and Triggers** section.
- For the **Image Cache** processor, see information in the **Image Cache 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.

10. Once any required configurations are made, click **Next**, and the Event Processor Wizard will display step '4. Schedule Event Processor.'

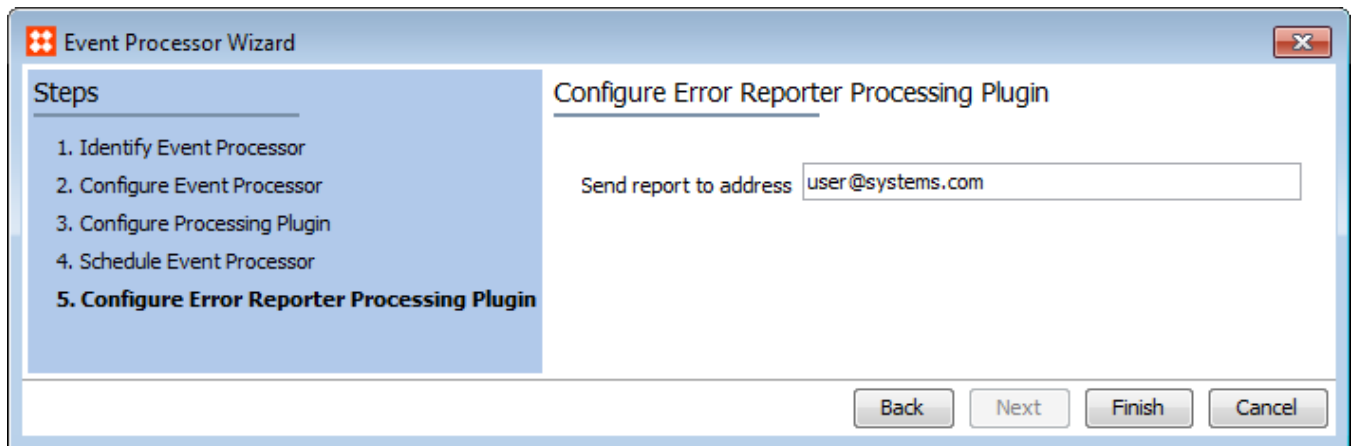


11. Schedule the event processor by selecting a start option.

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 and also the start and end dates for the schedule.
- **Later:** A specific time and date that the process will run only once.

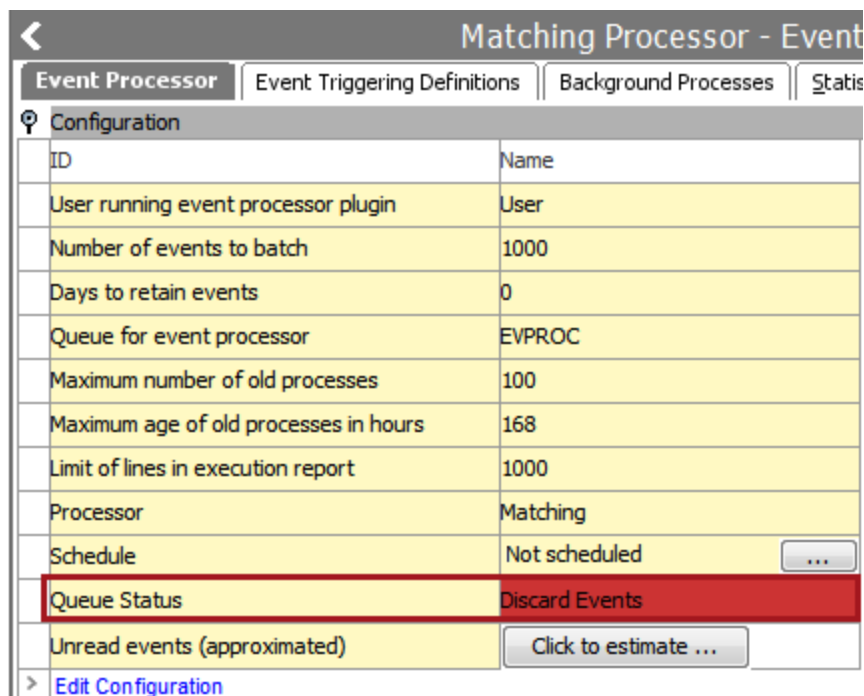
12. Once any required configurations are made, click **Next**, and the Event Processor Wizard will display step '5. Configure Error Reporter Processing Plugin.'



13. Optionally enter an email address to send error reports to, otherwise click **Finish** to complete the wizard.

Important: During step 2. Configure Event Processor, if the 'Select Error Reporter' option was set to 'Mail Error Report', then the 'Send report to address' option will display.

14. 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.
15. 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.



16. 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 **OIEP - Event-Based - Event Triggering Definitions Tab** documentation.

For information on common triggering definitions setups for the core event processors, see the following sections:

- **Expand LOV Value Changes Processor Parameters and Triggers**
- **Image Cache Processor Parameters and Triggers**
- **Matching Processor Parameters and Triggers**
- **Revision Management Processor Parameters and Triggers**

The screenshot displays the 'System Setup' interface with a tree view on the left and a configuration panel on the right. The tree view includes categories like 'Event Processors', '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', and 'Table'. The 'Matching Processor' is selected under 'Event Processors'. The right panel, titled 'Matching Processor - Event Triggering Definitions', has four tabs: 'Event Processor', 'Event Triggering Definitions' (active), 'Background Processes', and 'Statistics'. The 'Event Triggering Definitions' tab contains several sections: 'Triggering Object Types' with a dropdown for 'Object Types' and buttons for 'Event Filter' and 'Generate Event'; 'Triggering Attributes' with a dropdown for 'Name' and an 'Add Attribute' button; 'Triggering Table Types' with a dropdown for 'Table Types' and an 'Add Table Type' button; 'Reference Type Triggers' with a dropdown for 'Reference Types' and an 'Add Reference Type' button; and 'Miscellaneous Triggers' with four checked checkboxes: 'Names enabled', 'Parent links enabled', 'Attribute-links enabled', and 'Index-word Hierarchy enabled'.

Expand LOV Value Changes Processor Parameters and Triggers

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.

The Parameters and Event Triggers sections below contain important information on settings that should be considered when creating an event processor for expanding LOV value changes.

Prerequisites

This section of documentation describes configuration steps for a specific processor that is only one part of configuring an event processor. For instructions on configuring an event processor see the **Event Processors** documentation.

Parameters

On the Event Processor Wizard 'Configure Processing Plugin' step, 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.

The screenshot displays the 'System Setup' interface on the left, with a tree view containing categories like Event Processors, 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, Table, and Keys.

The main panel is titled '(stibo.2207160) - Event Triggering Definitions' and features several tabs: Event Processor, Event Triggering Definitions (selected), Background Processes, and Statistics. The 'Event Triggering Definitions' tab is active, showing a configuration table with columns for Object Types, Event Filter, and Generate Event. Below this table are sections for adding object types, attributes, table types, and reference types. A red box highlights the 'Miscellaneous Triggers' section, which includes checkboxes for 'Names enabled', 'Parent links enabled', 'Attribute-links enabled', and 'Index-word Hierarchy enabled'.

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** section of the **Outbound Integration Endpoint** documentation.

Image Cache Processor Parameters and Triggers

Image Cache allows the event processor to generate images for required conversions asynchronously, and saves the resulting files to the image cache. Common setup is for all STEP systems to use this event processor as a means of optimizing asset export time to external systems. Only image conversion configurations with the 'Cache generated images' option checked are handled by this event processor. It is recommended that all systems using image conversions have imaging caching processors set up. For more information, see the **Image Conversion Configurations** section of the **Export Manager** documentation.

Although caching images is optional, a cache improves export performance, especially for those assets that are exported frequently. Image caching is automatically defined for STEP with no setup required. All image caching is controlled by the **ImageCache.*** properties in the sharedconfig.properties file. Contact Stibo Systems Technical Support for assistance when modifying the default settings to meet your needs.

The Parameters and Event Triggers sections below contain important information on settings that should be considered when creating an event processor for Image Cache.

Prerequisites

This section of documentation describes configuration steps for a specific processor that is only one part of configuring an event processor. For instructions on configuring an event processor see the **Event Processors** documentation.

An image conversion that uses caching must have been previously created. Only Image conversion configurations with the 'Cache generated images' option checked are handled by this event processor. For more information on creating these conversions, see the **Image Conversion Configurations** section of the **Export Manager** documentation.

Parameters

On the Event Processor Wizard 'Configure Processing Plugin' step, configure the processing plugin.

1. For **Classification**, click the ellipsis button (...) to display the Select Classification dialog.
2. Choose the classification folder that will be monitored for images to be cached and click the **Select** button. Each time an asset is added to the folder, or an existing asset is modified, that corresponds to the triggering definitions (below), an event will be generated for the image cache processor.

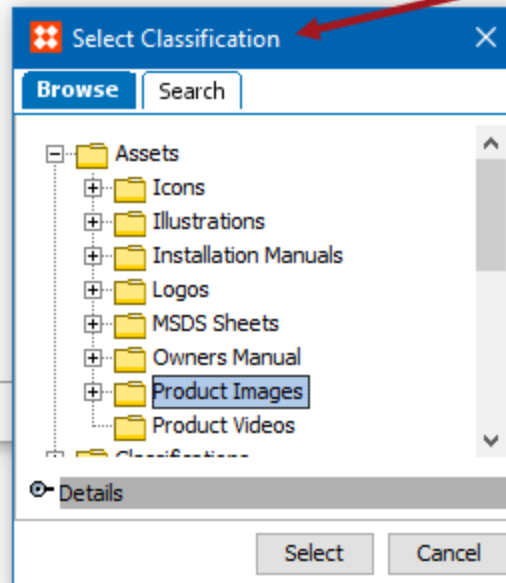
 Event Processor Wizard

Steps

1. Identify Event Processor
2. Configure Event Processor
- 3. Configure Processing Plugin**
4. Schedule Event Processor
5. Configure Error Reporter Processing Plugin

Configure Processing Plugin

Classification



Cancel

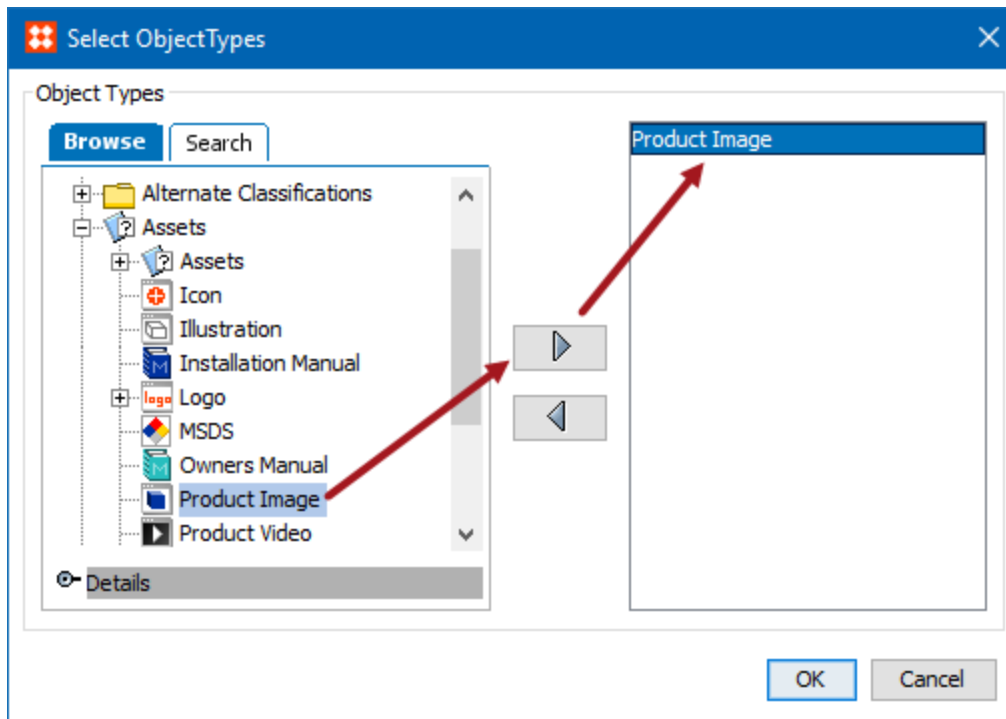
Select

Cancel

Event Triggers

For the event processor to take effect, it must be configured to only listen for changes on asset type objects housed within the same previously selected classification folder.

1. Click the **Add Object Type** link to display the Select Object Types dialog.
2. Select one or more asset object types that will be monitored for changes. When business rules are not required in the Event Filter and Generate Event fields, multiple object types can be selected within the same triggering object types row.



3. Click OK to add the selection to the Triggering Object Types table.

Important: You must select at least one asset type for Triggering Object Types or nothing will be processed by the event processor.

4. Add any required business rules in the Event Filter and Generate Event fields.

System Setup

- Event Processors
 - External Item Async Matching
 - Find Existing
 - Image Cache**
 - MatchCodeProcess
 - Matching Event
 - Matching Processor
 - Person Match
 - 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

Image Cache - Event Triggering Definitions

Event Processor
Event Triggering Definitions
Background Processes
Statistics

Triggering Object Types

Object Types	Event Filter	Generate Event
> Product Image	...	...

[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

Matching Processor Parameters and Triggers

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 Parameters and Event Triggers sections below contain important information on settings that should be considered when creating an event processor for matching.

Prerequisites

This section of documentation describes configuration steps for a specific processor that is only one part of configuring an event processor. For instructions on configuring an event processor see the **Event Processors** documentation.

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. For details about match codes and matching algorithms, see **Matching and Linking Components** in the **Matching and Linking** documentation.

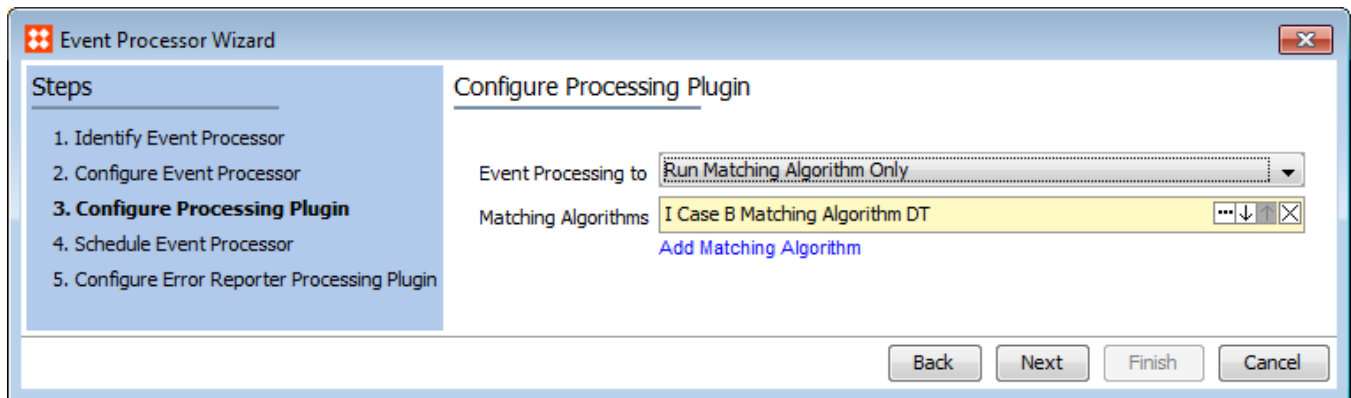
Important: The 'Run Matching Algorithm' functionality is controlled by System Setup > Action Sets > Setup Actions > Maintain Deduplication Configurations.

Parameters

On the Event Processor Wizard 'Configure Processing Plugin' step, 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.

System Setup

- Event Processors
 - External Item Async Matching
 - Matching Event**
 - Matching 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
- Table
- Keys
- Event Queues
- Component Models
- Recycle Bin

Matching Event - Event Triggeri

Event Processor
Event Triggering Definitions
Background Processes
Statistics

Triggering Object Types

Object Types	>	Event Filter	>	Generate Event	>
> External Item					

[Add Object Type](#)

Triggering Attributes

Name
> OEM
> OEM Part Number

[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

Revision Management Processor Parameters and Triggers

Revision Management allows automatic purging of object revisions 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. It is strongly recommended that all STEP systems have this event processor set up as a means of managing the STEP database size.

The Parameters and Event Triggers sections below contain important information on settings that should be considered when creating an event processor for revision management.

Prerequisites

This section of documentation describes configuration steps for a specific processor that is only one part of configuring an event processor. For instructions on configuring an event processor see the **Event Processors** documentation.

Parameters

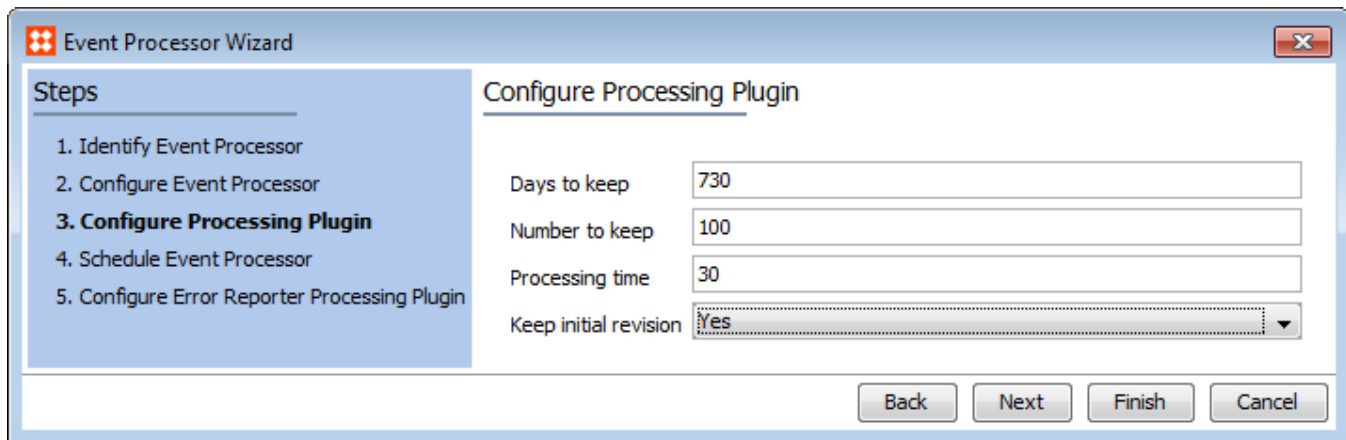
On the Event Processor Wizard 'Configure Processing Plugin' step, configure the Revision Management processor.

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 Processor Wizard

Steps

1. Identify Event Processor
2. Configure Event Processor
- 3. Configure Processing Plugin**
4. Schedule Event Processor
5. Configure Error Reporter Processing Plugin

Configure Processing Plugin

Days to keep: 730

Number to keep: 100

Processing time: 30

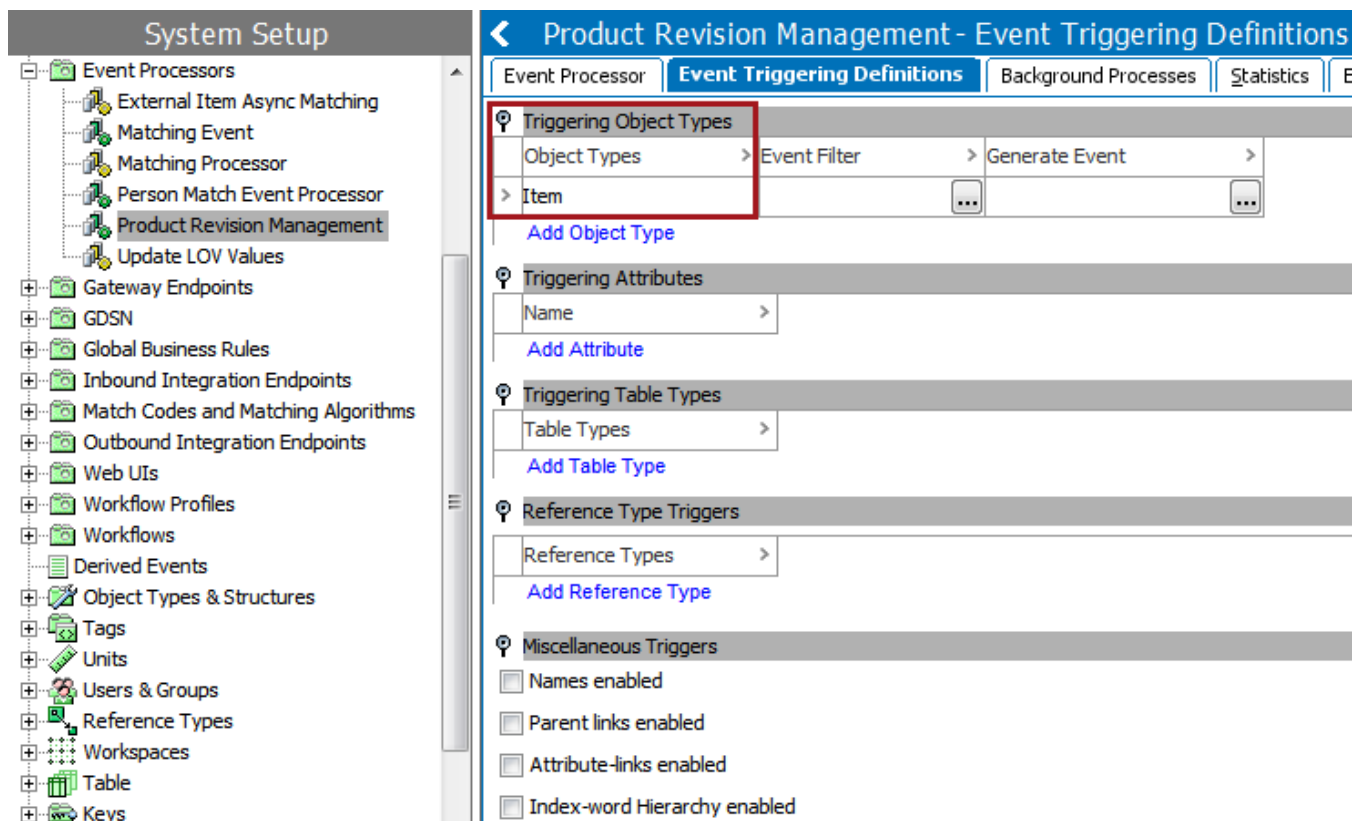
Keep initial revision: Yes

Buttons: Back, Next, Finish, Cancel

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.



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
- Table
- Keys

Product Revision Management - Event Triggering Definitions

Event Processor | **Event Triggering Definitions** | Background Processes | Statistics

Triggering Object Types

Object Types	Event Filter	Generate Event
> Item	...	...

[Add Object Type](#)

Triggering Attributes

Name >

[Add Attribute](#)

Triggering Table Types

Table Types >

[Add Table Type](#)

Reference Type Triggers

Reference Types >

[Add Reference Type](#)

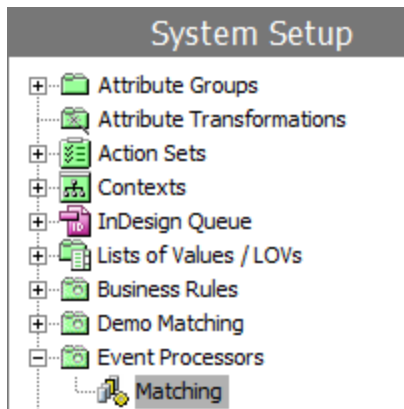
Miscellaneous Triggers

- ☐ Names enabled
- ☐ Parent links enabled
- ☐ Attribute-links enabled
- ☐ Index-word Hierarchy enabled

Enabling Event Processors

To enable an event processor, complete the following steps:

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		
Event Triggering Definitions		
Background Process		
Description		
Name	>	Value
ID	>	mltEventProcessor
Name	>	mltEventProcessor
Type	>	Event Processor
Last Edited by	>	2014-09-12 10:51:30 by STEPSYS
Enabled	>	Yes
Processor Status	>	Running

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.
	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. LOVs are also sometimes referred to as 'Domains.'

Values	Value ID
Alkaline	Alkaline
Lithium-ion	LithiumIon
NiMH	NickelMetalHydride
Plutonium	Plutonium

Name	Value
Battery Type	Alkaline (Alkaline)
Capacity	123
CSA Listed	123
ETL Listed	123
Maximum Voltage	123
Primary Color	123

Ideally, each LOV value is unique. For example:

- Battery Size—AA, AAA, D, C, 9V
- Material—Brick, Steel, 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.

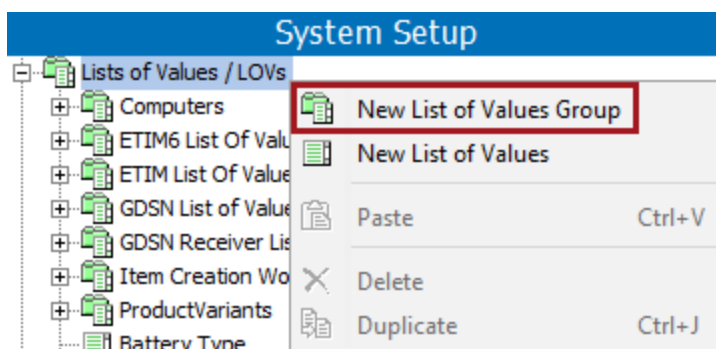
To see all LOV values, type an asterisk (*) to initially display most of the values, and after fully loading, display the complete, scrollable list of values.

Creating an LOV Group

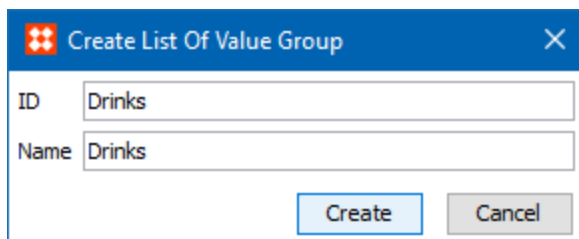
Individual, but related, Lists of Values can be kept together using a List of Values Group. The group can be linked into an attribute and used in combination with privileges to restrict users in maintaining certain List of Values.

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.

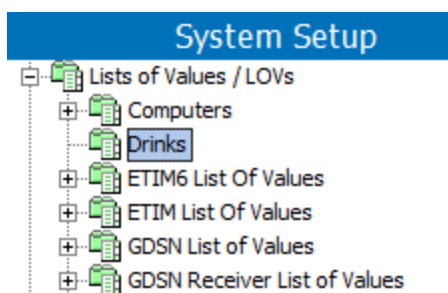
1. In **System Setup**, right-click List of Values (LOVs) and select the **New List of Value Group** option.



2. Key in **ID** and **Name**.



3. Click **Create**.

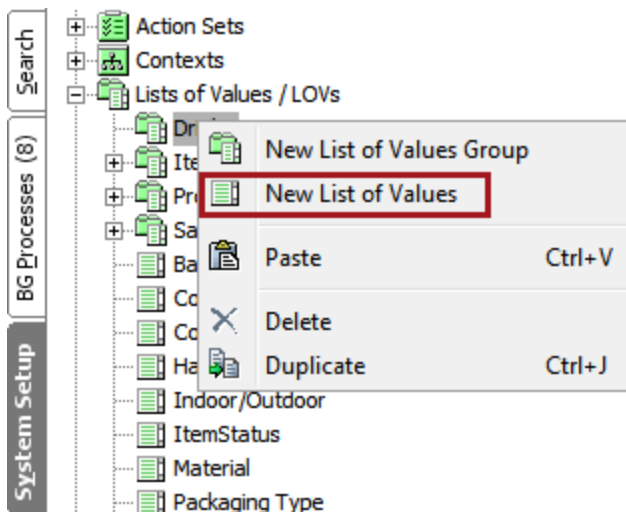


4. Add an LOV to the LOV Group by following the steps in **Creating a List of Values**.

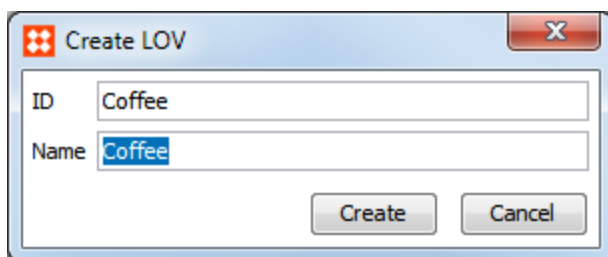
Creating an LOV

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.

<
Coffee - List of Values

List of Values
References
Log
State Log
Tasks

Description	
Name	> > Value
ID	Coffee
Name	Coffee
Edited by	2015-08-11 13:58:09 by USER
Path	Lists of Values / LOVs/Drinks/Coffee
Dimension Dependencies	
Use Ids on values	No
Use Ids for sorting	No
Value-ID Pattern	

In Attribute Groups

ID	> Name	>

[Add Attribute Group](#)

List of Values Validation

Name	Value
Validation Base Type	Text
Allow Users to Add Values	No
Mask	
Minimum Value	N/A
Maximum Value	N/A
Maximum Length	100

Values

Values	>

[Add Value](#)

4. Under **List of Values Validation**, select validation base type.

List of Values Validation

Name	Value
Validation Base Type	Text
Allow Users to Add Values	ISO Date
Mask	ISO Date and Time
Minimum Value	Number
Maximum Value	Number Range
Maximum Length	Numeric Text

Values

Values	>

Regular Expression

Text

URL

>

5. Under **List of Values Validation**, determine when users can add values.

In Attribute Groups	
ID	> Name >
> Add Attribute Group	
List of Values Validation	
Name	> Value >
> Validation Base Type	Text
> Allow Users to Add Values	No
> Mask	Yes
> Minimum Value	No
> Maximum Value	N/A
> Maximum Length	100

- **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.

<
Coffee - List of Values

List of Values
References
Log
State Log
Tasks

☰ Description

Name	>	>	Value
> ID			Coffee
> Name			Coffee
> Edited by			2015-08-11 14:41:19 by USER
> Path			Lists of Values / LOVs/Drinks/Coffee
> Dimension Dependencies			
> Use Ids on values			Yes
> Use Ids for sorting			No
> Value-ID Pattern			[id]

☰ In Attribute Groups

ID	>	>	Name
> Add Attribute Group			

☰ List of Values Validation

Name	>	Value
> Validation Base Type		Text
> Allow Users to Add Values		No
> Mask		
> Minimum Value		N/A
> Maximum Value		N/A
> Maximum Length		100

☰ Values

Values	v	Value ID	>

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.

List of Values Validation

Name	Value
> Validation Base Type	Number
> Allow Users to Add Values	No
> Mask	
> Minimum Value	
> Maximum Value	
> Maximum Length	N/A

Units

ID	Name	Edited by	Path	Unit Proper Name	Default Unit
> Add Unit					

Values

Values	Value ID
Add Value	

8. **Optional:** in the **Dimension Dependencies** field, click the ellipsis button (...), and select one or more dimensions.

Setting a dimension dependency of Language allows your LOV values to be translatable and thus appear differently when viewed in different contexts. For more information on making objects and object types dimension dependent, see the **Maintaining Dimension Dependent Object Types** topic. For more information on using attributes with dimension-dependent LOVs, see the **Using Dimension Dependent LOVs With Attributes** topic.

List of Values | References | Log | State Log | Tasks

Description

Name	Value
> ID	Coffee
> Name	Coffee
> Edited by	2015-08-11 15:02:45 by USER
> Path	Lists of Values / LOVs/Drinks/Coffee
> Dimension Dependencies	...
> Use Ids on values	Yes

Select Dimension Dependencies

☐ Country

☐ Language

Level of difficulty of applying changes: 0

Operation might be time consuming

Apply Cancel

Note: To make the LOV object type *itself* dimension dependent so the **names** of your LOVs can be translated, you must set a language dimension dependency on the **Domains** object type, located here: System Setup > Object Types & Structures > Basic Object Types > List Of Values Group Type > **Domains**.

The screenshot shows the 'System Setup' window on the left and the 'Domains - Object Type' configuration window on the right. In the 'System Setup' window, the 'Domains' object type is selected under the 'List Of Values Group Type' category. A red arrow points from this selection to the 'Domains - Object Type' window. In the 'Domains - Object Type' window, the 'Object Type' tab is active, showing a table of attributes. The 'Dimension Dependencies' attribute is highlighted with a red box, and its value 'Language;' is also highlighted with a red box.

Description	
Name	Value
ID	Domain user-type root
Name	Domains
Last edited by	2017-03-03 13:39:57 by
Name Pattern	
ID Pattern	
Icon	
Dimension Dependencies	Language;

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.

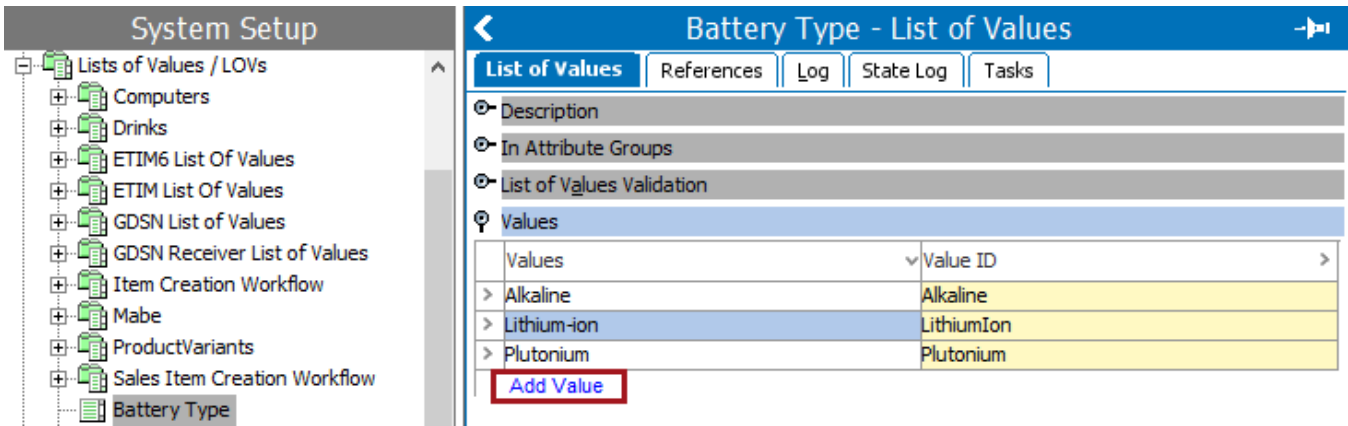
Adding Values to an LOV

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.



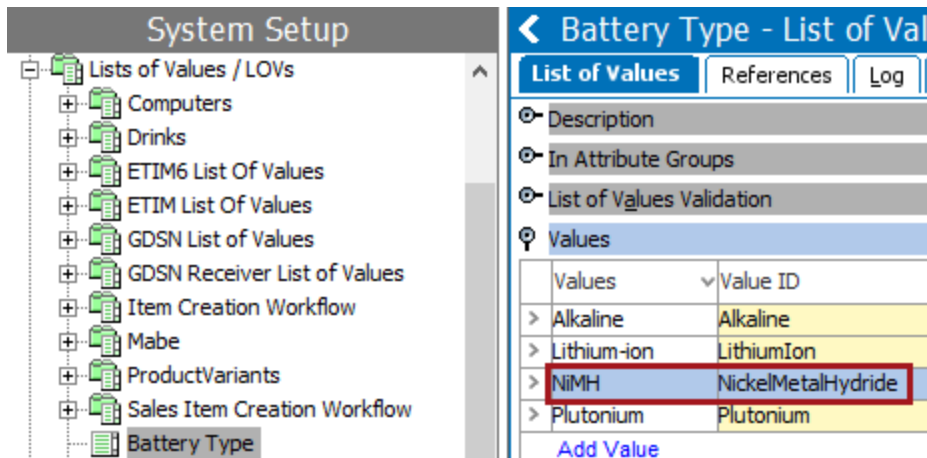
- 2. Enter a **Value ID** (if required by the LOV's Use ValueID parameter) and a **Value**.

If units are defined for the LOV, select the appropriate unit for the value from the available dropdown.

This screenshot shows a dialog box for adding a value. The 'Value ID' field contains the text 'NickelMetalHydride'. The 'Value' field contains the text 'NiMH'. Below the 'Value' field, there are statistics: 'Characters: 4 (4 with tags)', 'Characters left (including tags): 96', 'Words: 1', and 'Style: ""'. There are 'OK' and 'Cancel' buttons at the bottom.This screenshot shows a dialog box for adding a value. The 'Value ID' field contains the text '184939'. The 'Value' field contains the text '10'. Below the 'Value' field, there are statistics: 'Characters: 2 (2 with tags)', 'Words: 1', and 'Style: ""'. A unit dropdown menu is open, showing options 'yd', 'in', and 'ft'. There are 'OK' and 'Cancel' buttons at the bottom.

and click the **OK** button to add it to the LOV

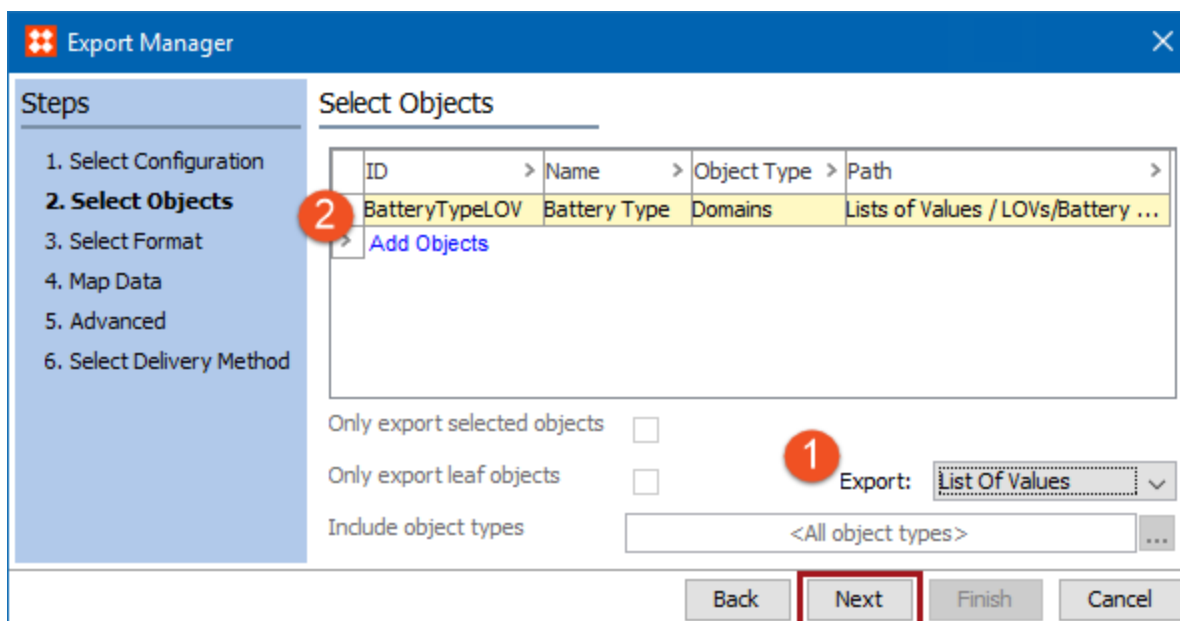
3. Verify the new value is displayed for the LOV.



Adding a Group of LOV Values

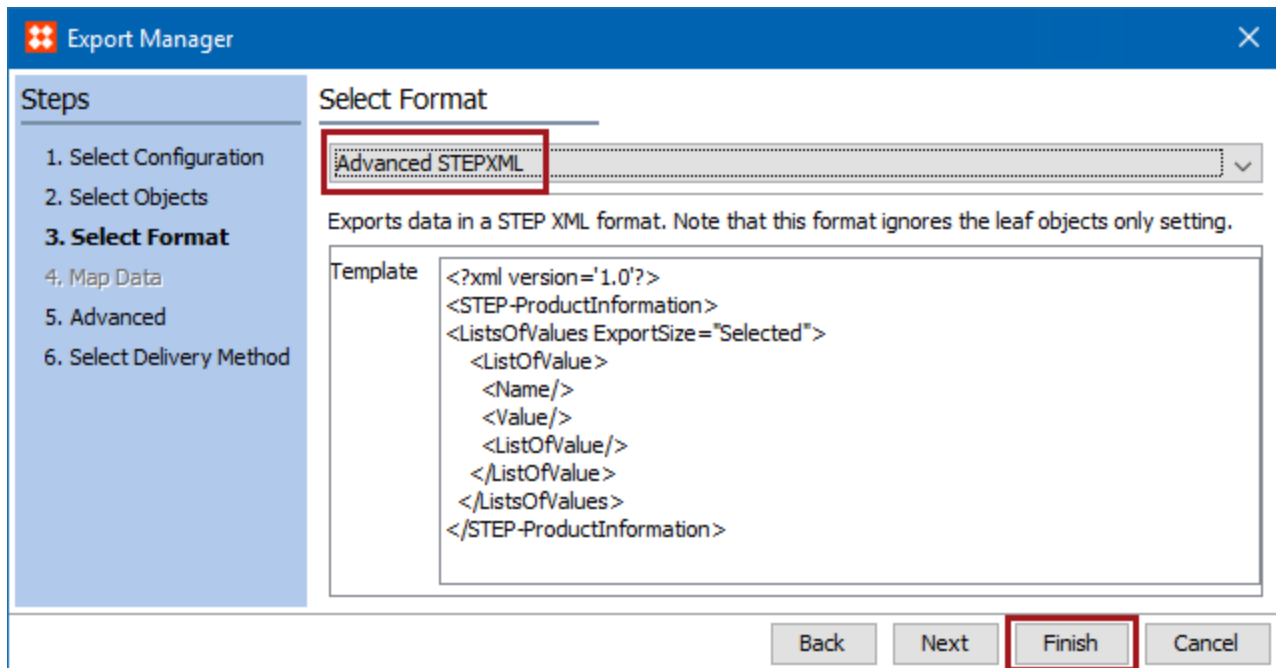
When many new values are required, it may be easier to import the values using STEPXML. This process involves exporting the existing LOV to create a STEPXML file, modifying the file to include the new values, and then importing those new values back into STEP.

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.

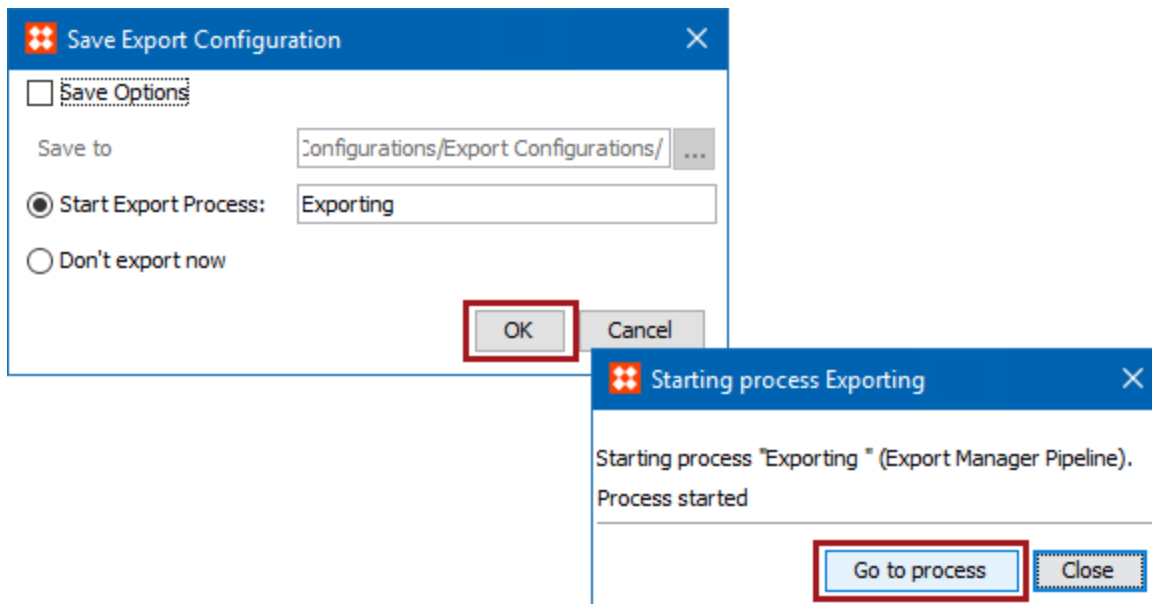


For details about each step and parameter of the Export Manager wizard, see the **Creating a Data Export** section of the **Data Exchange** documentation.

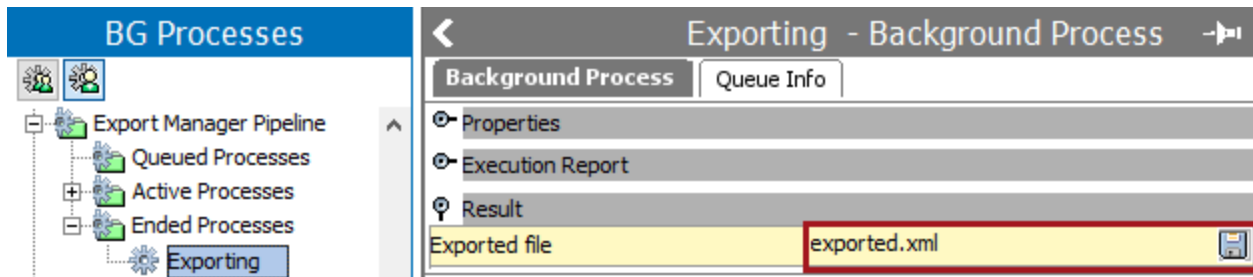
- On step 3 Select Format, select **Advanced STEPXML** and click the **Finish** button.



- On the Save Export Configuration dialog, click **OK** to start the export, and on the Starting process dialog, click **Go to process**.



- 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">
<ListOfValues ExportSize="Selected">
  <ListOfValue>
    <Name/>
    <Value/>
  </ListOfValue>
</ListOfValues>
</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">

  <ListOfValues>
    <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>
      </Validation>
    </ListOfValue>
  </ListOfValues>
</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">

  <ListOfValues>
    <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>
  </ListOfValues>
</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.

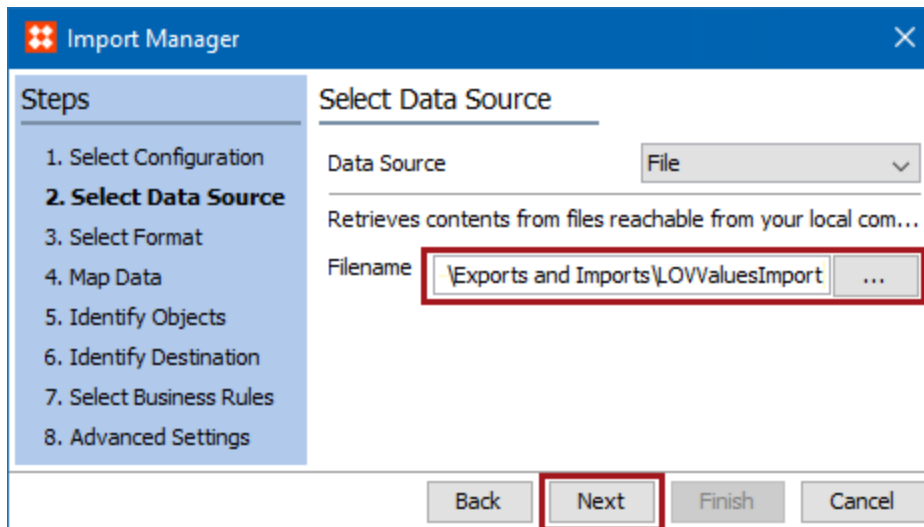
	A	B	C	D	E	F	G	H	I
1	<Value ID="	Var1 "	UnitID="	unece.unit.INH ">	32	</Value>			<Value ID="Var1" UnitID="unece.unit.INH">32</Value>
2	<Value ID="	Var2 "	UnitID="	unece.unit.FOT ">	45	</Value>			<Value ID="Var2" UnitID="unece.unit.FOT">45</Value>
3	<Value ID="	Var3 "	UnitID="	unece.unit.INH ">	58	</Value>			<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>

- 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>
```

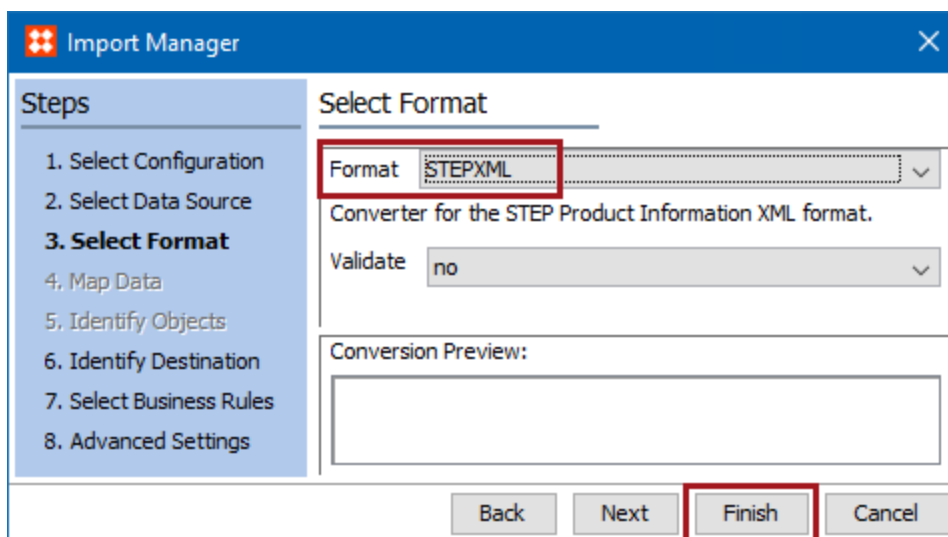
- File > Import > Data opens Import Manager, on step 2 Select Data Source:

- In the Filename field, select the **import file name**.
- Click the **Next** button.

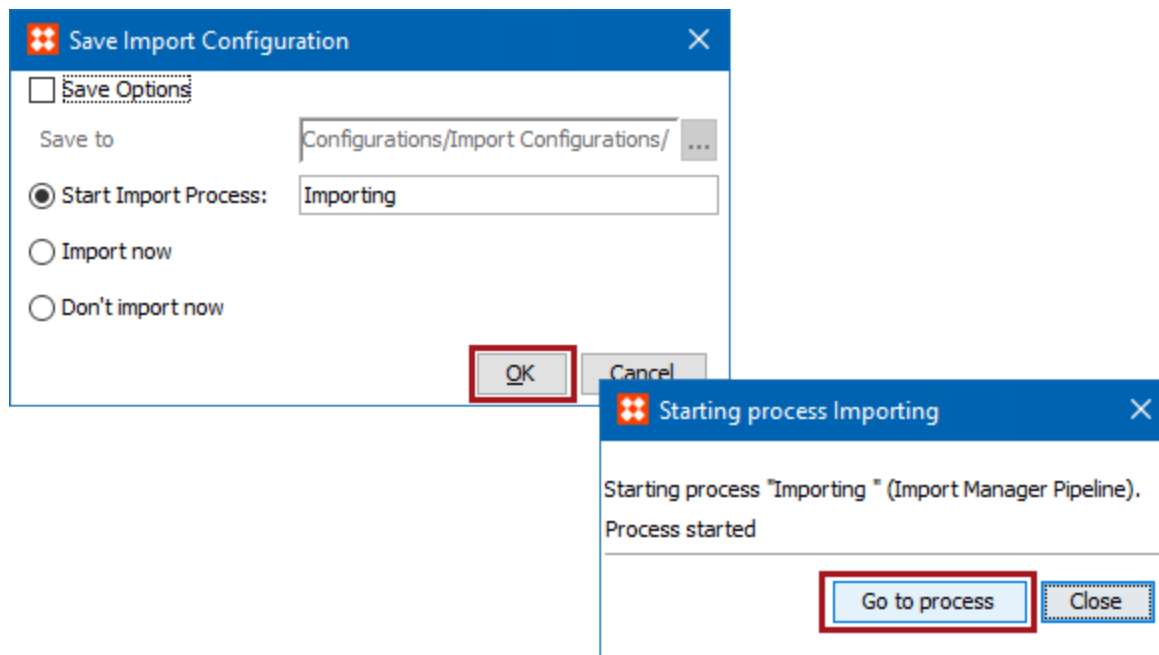


For details about each step and parameter of the Import Manager wizard, see the **Creating a Data Import** section of the **Data Exchange** documentation.

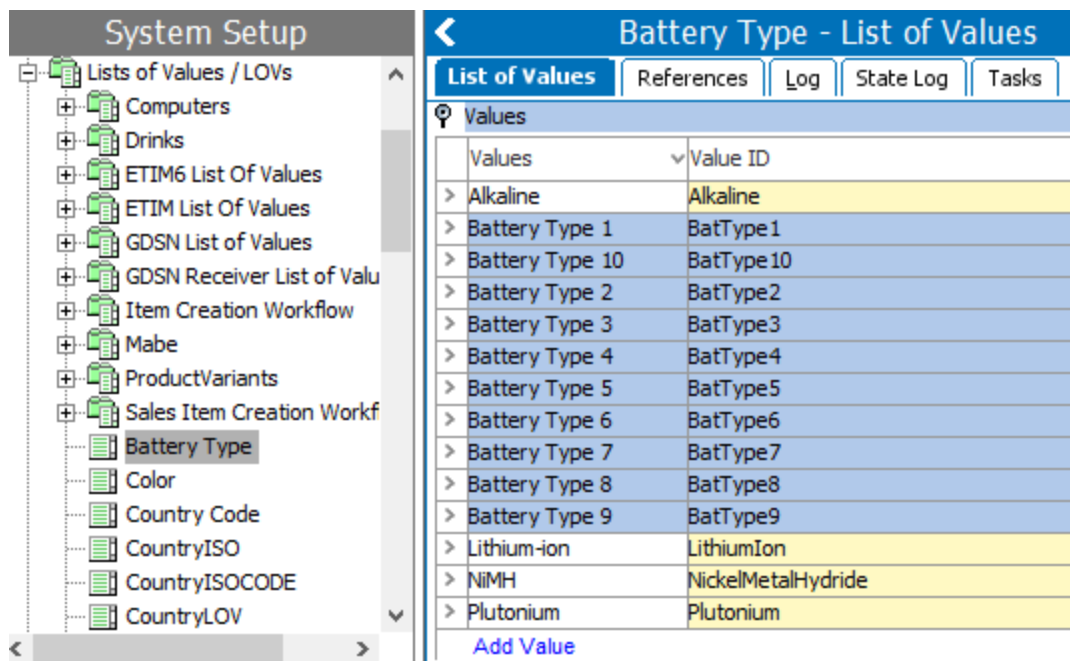
10. On step 3 Select Format, select **STEPXML** and click the **Finish** button.



11. On the Save Import Configuration dialog, click **OK** to start the import, and on the Starting process dialog, click **Go to process**.



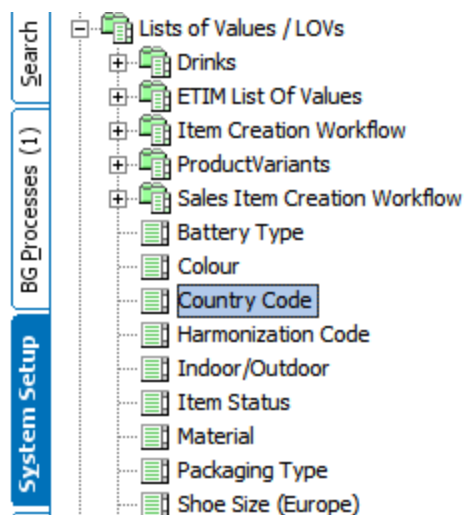
12. On the displayed background process editor, verify the import succeeded.
13. In System Setup, verify the new values are displayed for the LOV.



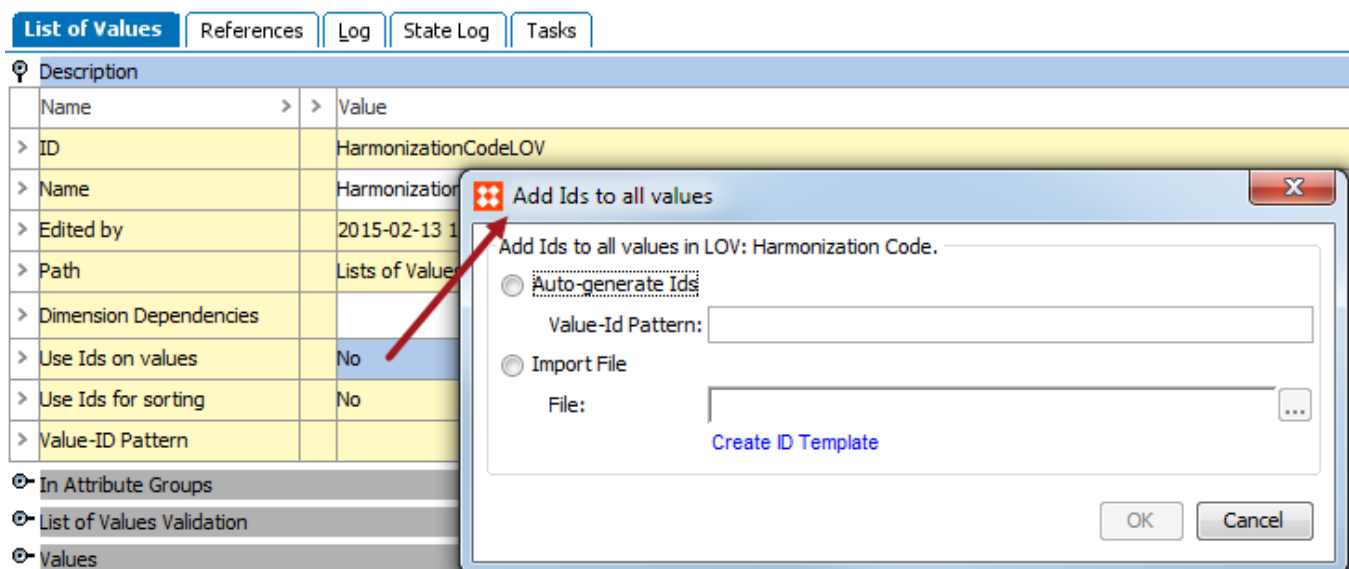
Adding IDs to Values in LOV

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.

Add Ids to all values

Add Ids to all values in LOV: Material.

☒ Auto-generate Ids

Value-Id Pattern:

☐ Import File

File: ...

[Create ID Template](#)

OK Cancel

3. Click **OK** to apply the Value-ID Pattern on existing values.

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		prefix [id]		

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.

Add Ids to all values

Add Ids to all values in LOV: Country Code.

☐ Auto-generate Ids

Value-Id Pattern:

☒ Import File

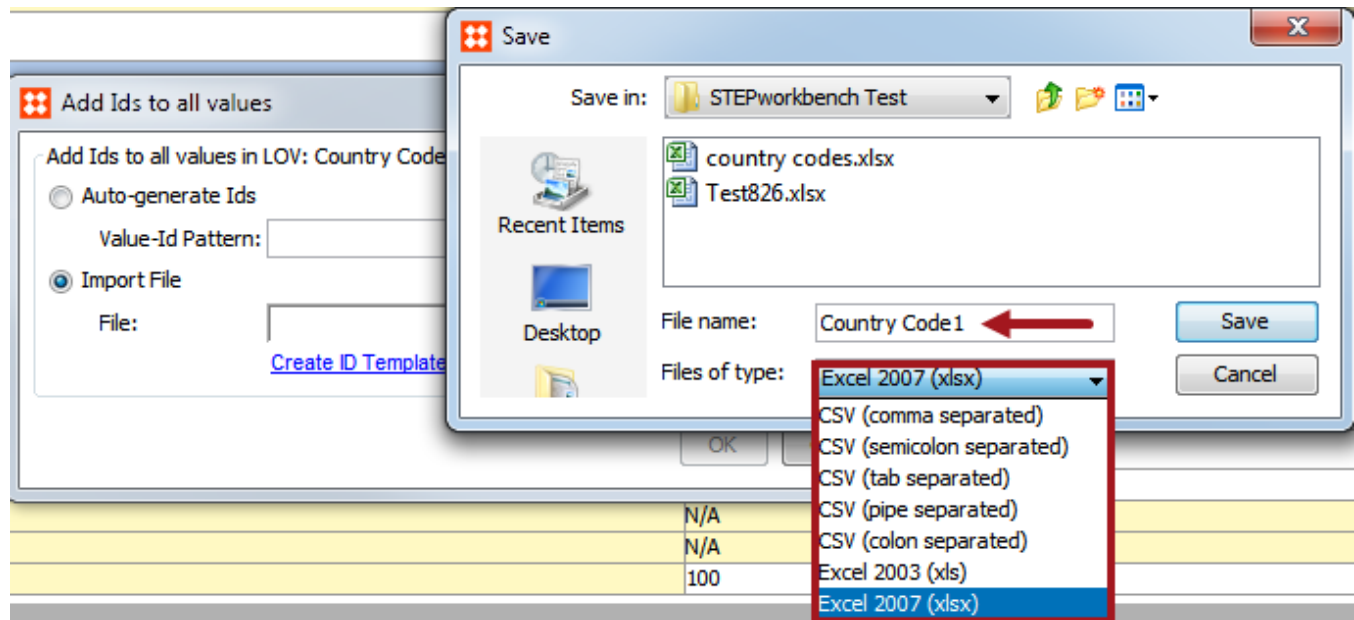
File: ...

[Create ID Template](#)

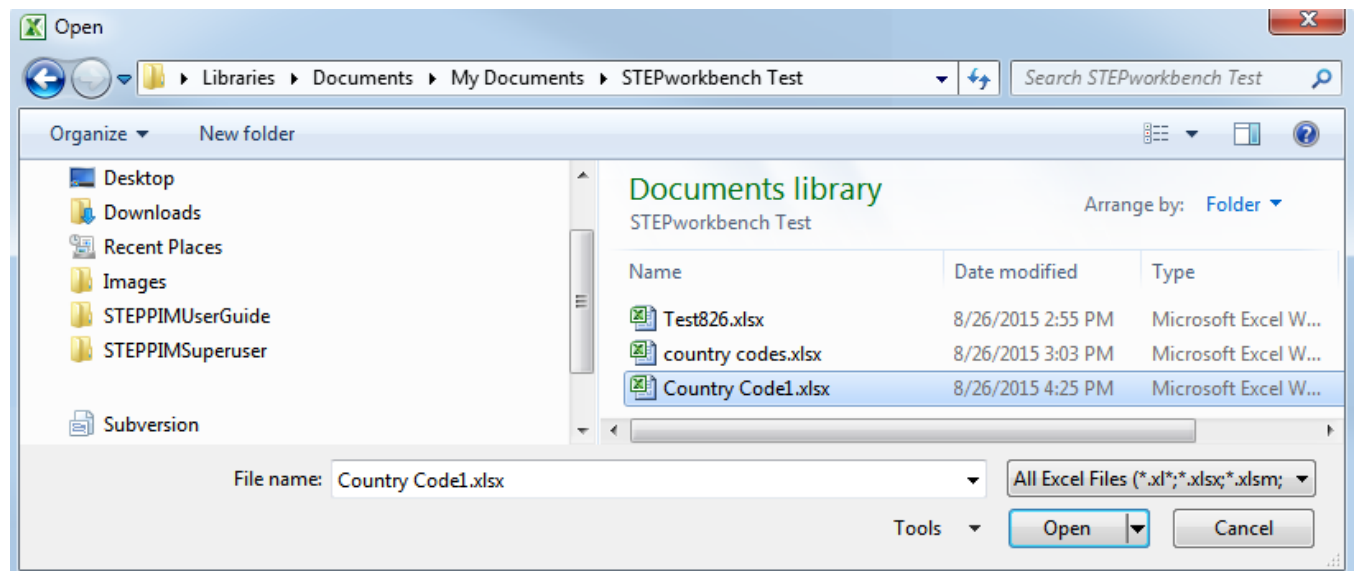
OK Cancel

- Choose destination of the saved template and name the template.

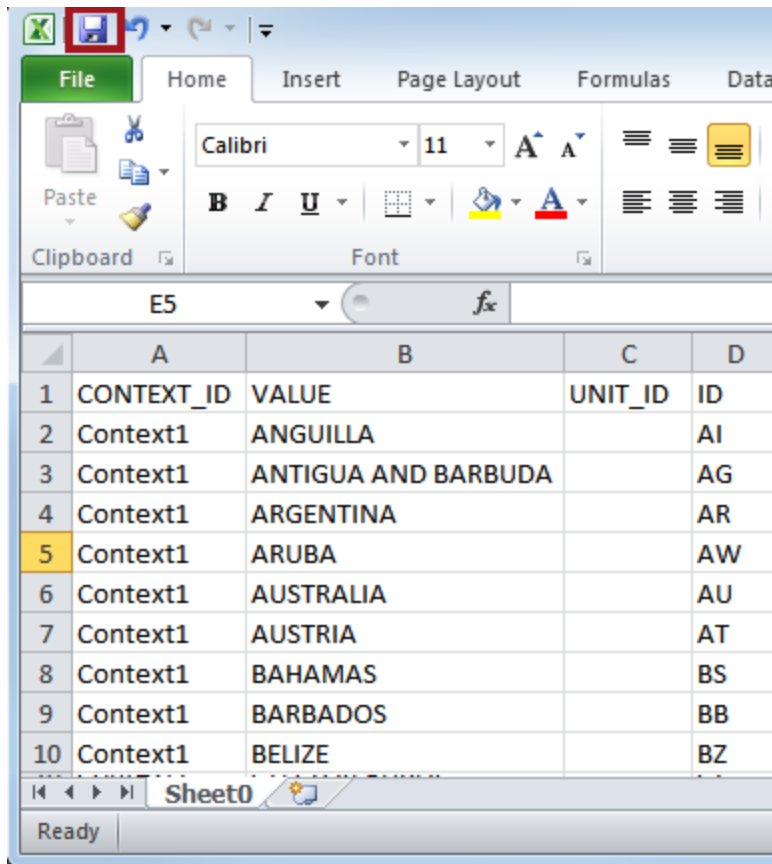
Note: An ID template can be saved in CSV (comma, semicolon, tab, pipe or color separated) or Excel (2003 or 2007) format.



- Click **Save**, leaving the **Add Ids to all values** dialog open.
- Open your template using to the program you used to create it.

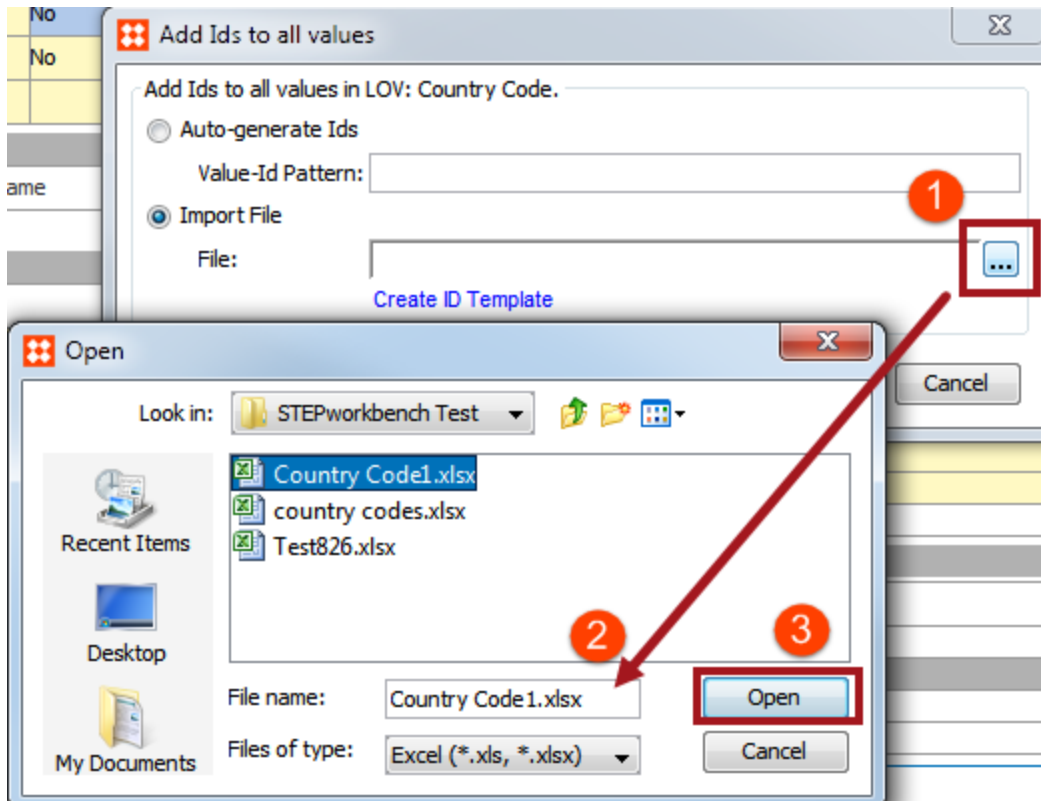


- Once the template is open, fill in and name your ID column accordingly, and save your file.

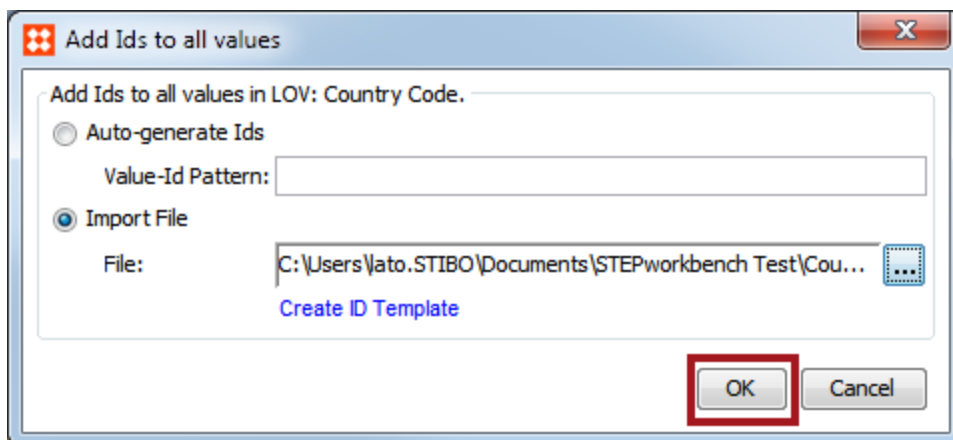


Note: Unit ID column is only filled if there are units, such as inches, pounds, centimeters, etc.

- 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**.

Merging LOVs

1. In **System Setup, Lists of Values / LOVs**, ensure that the **Dimension Dependencies** match on the LOVs to be merged.

The image shows two screenshots of the Stibo Systems interface, illustrating the process of merging LOVs. The left screenshot shows the 'Color - List of Values' configuration page, and the right screenshot shows the 'Colour - List of Values' configuration page. Both pages have a 'List of Values' tab selected. The 'Color' LOV is highlighted in the left-hand 'System Setup' tree, and the 'Colour' LOV is highlighted in the right-hand 'System Setup' tree. Red arrows point from these LOVs to their respective configuration panels.

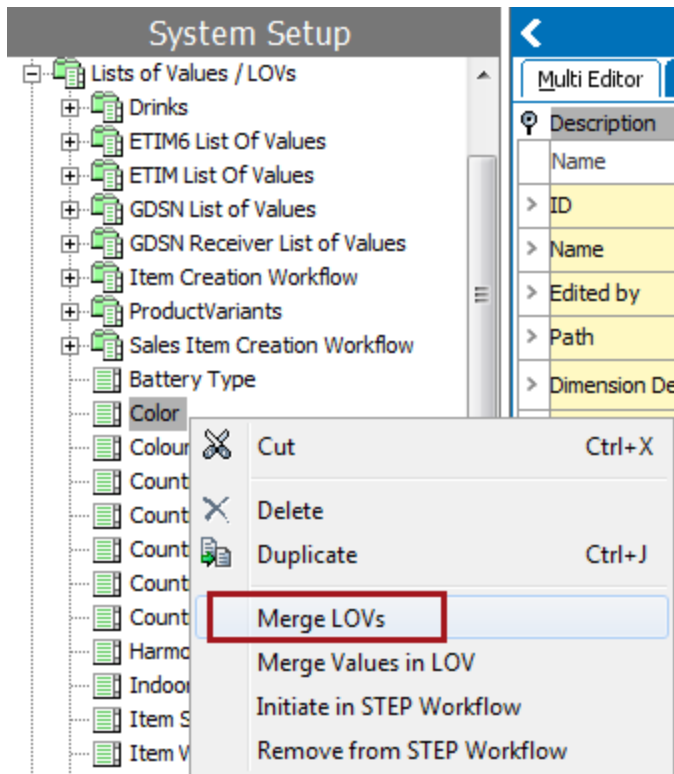
Color - List of Values Configuration:

Name	Value
ID	Color
Name	Color
Edited by	2016-03-28 14:45:34 by USER
Path	Lists of Values / LOVs/Color
Dimension Dependencies	Country;
Use Ids on values	Yes

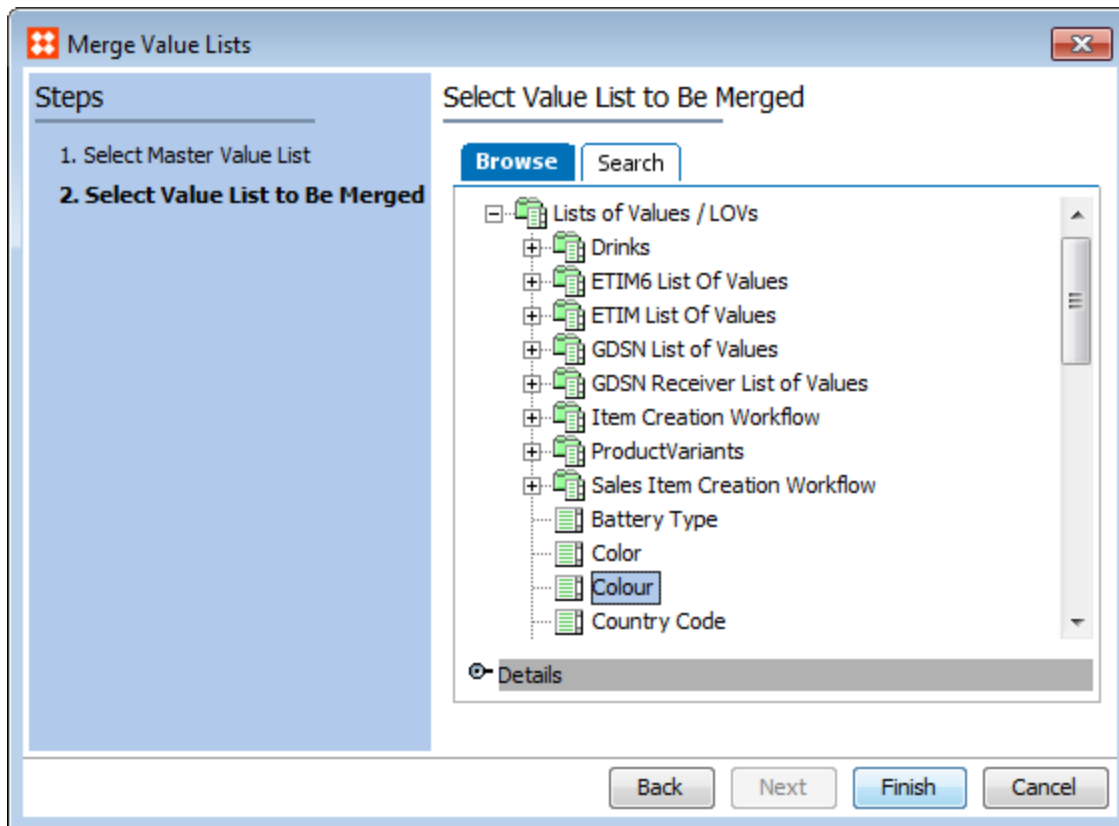
Colour - List of Values Configuration:

Name	Value
ID	Colour
Name	Colour
Edited by	2016-03-28 14:58:49 by USER
Path	Lists of Values / LOVs/Colour
Dimension Dependencies	Country;
Use Ids on values	Yes
Use Ids for sorting	No
Value-ID Pattern	[id]

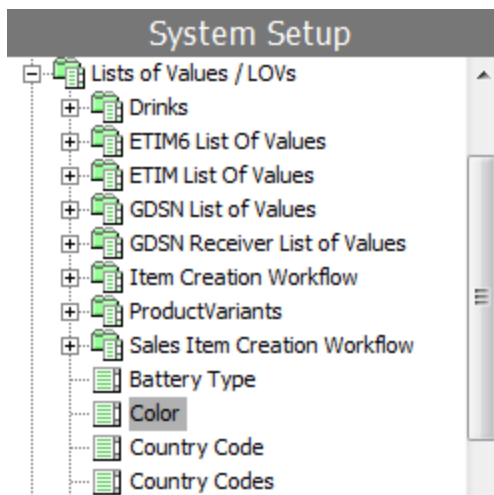
2. Highlight the master LOV (the LOV that will remain after the merge), right-click and select **Merge LOVs**.



3. In the Merge Value Lists wizard, use **Browse** or **Search** to find and select the LOV to be merged (the LOV that will be deleted) with the master LOV.



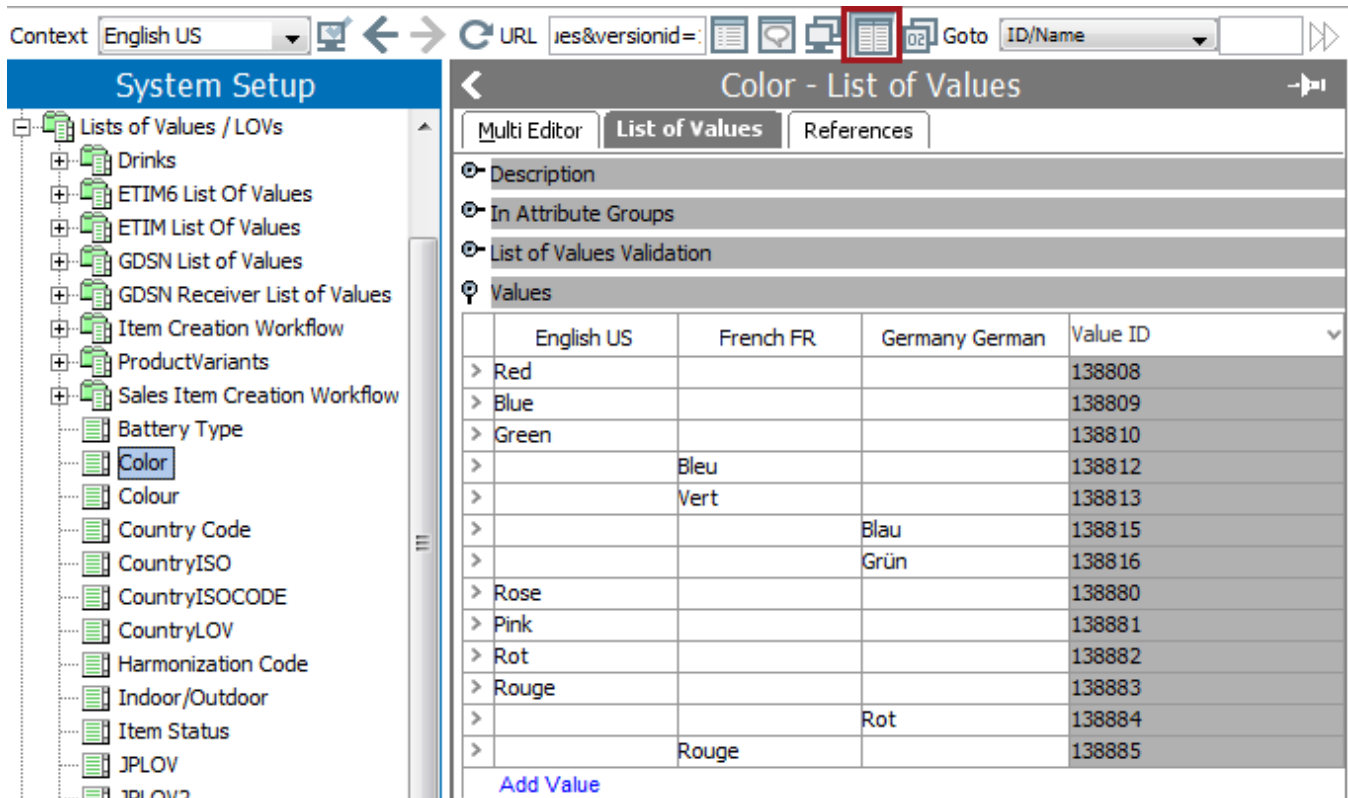
4. Click the **Finish** button to complete the merge process. The master LOV remains, the merged LOV has been deleted.



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.



The screenshot displays the Stibo Systems workbench interface. On the left, the 'System Setup' tree shows 'Lists of Values / LOVs' expanded, with 'Color' selected. The main pane shows the 'Color - List of Values' configuration. The 'List of Values' tab is active, displaying a table of color values across different languages (English US, French FR, Germany German) and their corresponding Value IDs. The table includes values like Red, Blue, Green, Bleu, Vert, Blau, Grün, Rose, Pink, Rot, and Rouge, each with a unique Value ID ranging from 138808 to 138885. A red box highlights the 'Context' dropdown menu at the top, which is set to 'English US'.

	English US	French FR	Germany German	Value ID
> Red				138808
> Blue				138809
> Green				138810
>		Bleu		138812
>		Vert		138813
>			Blau	138815
>			Grün	138816
> Rose				138880
> Pink				138881
> Rot				138882
> Rouge				138883
>			Rot	138884
>		Rouge		138885

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.

System Setup

- Lists of Values / LOVs
 - Drinks
 - ETIM6 List Of Values
 - ETIM List Of Values
 - GDSN List of Values
 - GDSN Receiver List of Values
 - Item Creation Workflow
 - ProductVariants
 - Sales Item Creation Workflow
 - Battery Type
 - Color**
 - Colour
 - Country Code
 - CountryISO

Color - List of Values

List of Values | References | Log | State Log | Tasks

Description	
Name	Value
ID	Color
Name	Color
Edited by	2016-03-16 09:55:25 by USER
Path	Lists of Values / LOVs/Color
Dimension Dependencies	Country;
Use Ids on values	Yes
Use Ids for sorting	No
Value-ID Pattern	[id]

- 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.

Color - List of Values

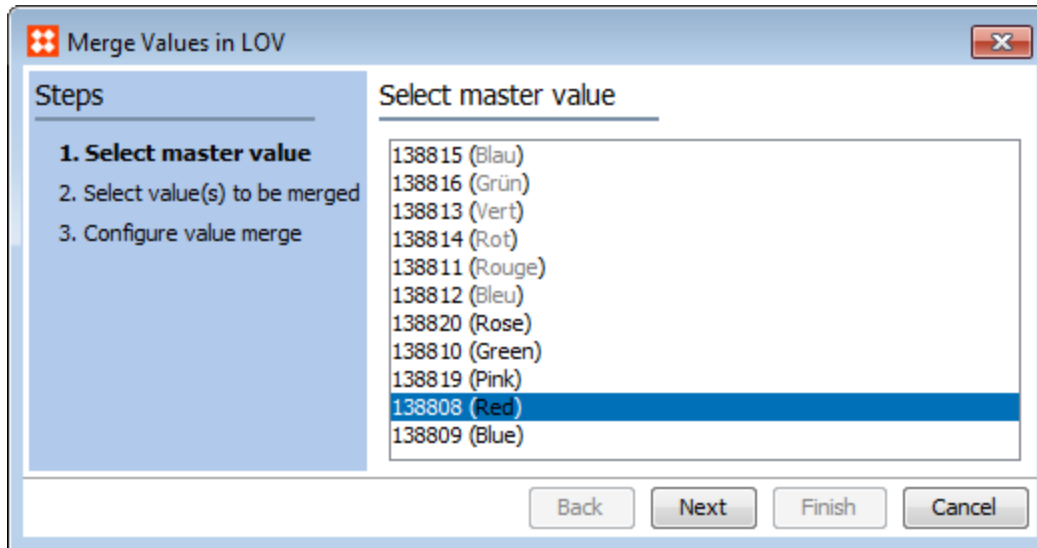
Multi Editor | **List of Values** | References

Description
In Attribute Groups
List of Values Validation
Values

English US	French FR	Germany German	Value ID
> Red			138808
> Blue			138809
> Green			138810
>	Bleu		138812
>	Vert		138813
>		Blau	138815
>		Grün	138816
> Rose			138880
> Pink			138881
> Rot			138882
> Rouge			138883
>		Rot	138884
>	Rouge		138885

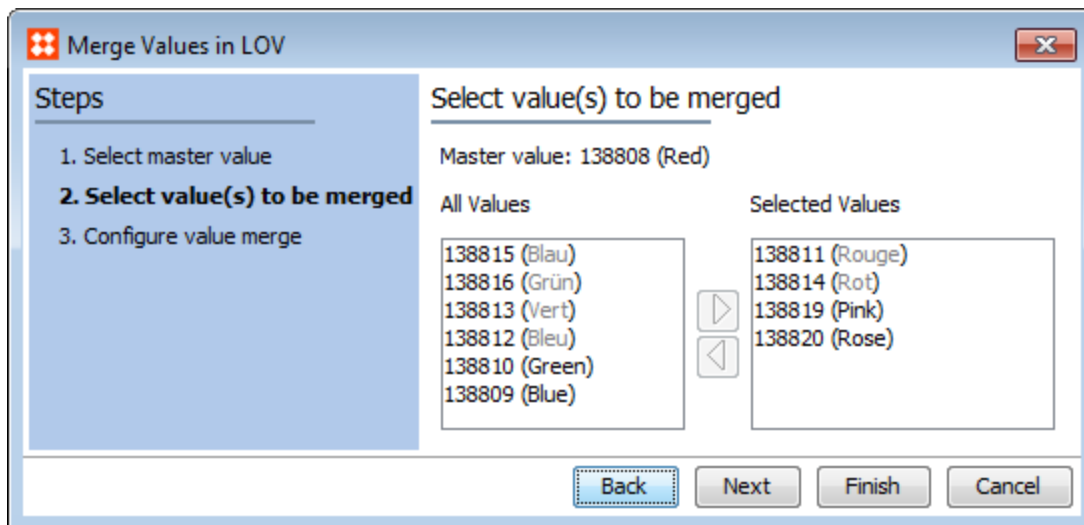
Add Value

- Select the LOV, right-click and select **Merge Values in LOV**.
- 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.

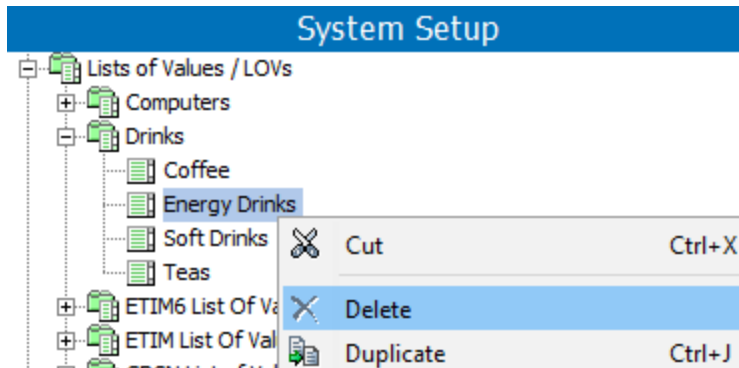
Repeat this process for all values to be merged into the master value and click the **Next** button.



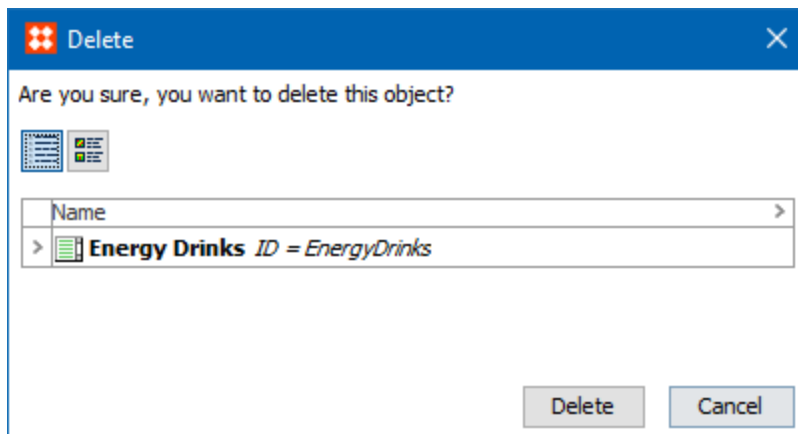
7. In the 'Merged/Deleted LOV values' table, use the handles to drag LOV values and reorder as necessary.

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

In STEP, historical versions of objects are stored as revisions. A revision thus represents a historical "snapshot" of an object. 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.

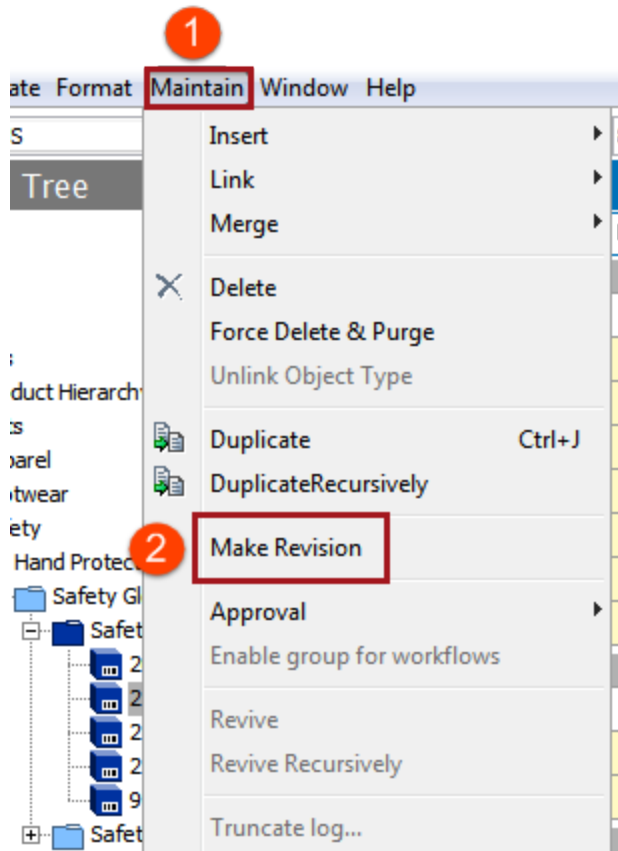
It is important to note the difference between *revisions* and *approvals*. The two concepts are related, but should not be confused. Objects that are *workspace revisable* are subject to approvals, and approving an object automatically generates a revision of it. However, objects that are *globally 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 Approval of Objects section of the Getting Started / STEP User Guide. Also note that most object types are either strictly workspace revisable, or strictly globally revisable, with no option to change the revisability handling. The exception to this is entity object types, which have a Revisability parameter that allows for configuration of the revisability type. More information on this is available in the Revisability on Entity Object Type topic within this 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.

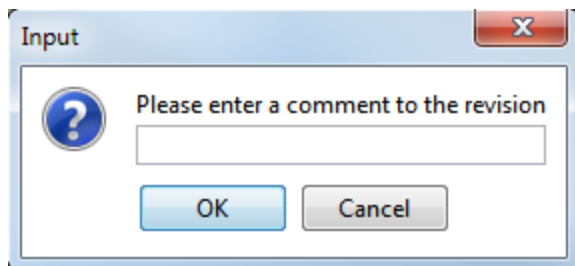
Generating Revisions

A revision can be created in one of five ways:

1. Upon creation of a new workspace revisable object.
2. Automatically, each time a change is made by a different user, or when any change is made following an approval. Note that for assets, when the digital media file associated with an asset in STEP is changed, this generates a revision on the asset object.
3. 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.



4. 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).
5. Upon approval of an object (applicable only to workspace revisable objects).

Note that a new revision is *not* created for every change to an object, thus STEP does not have traceability on the field level. This means that not *all* historical values on an object are stored as they may have differed *between* revisions, and the values are only stored *at* revisions. As stated above, this means that if the same user is making continuous edits to an object, these are not captured as a revision until the object is approved, the automatic revision threshold expires, or another user makes changes to the object.

Important considerations for revisions:

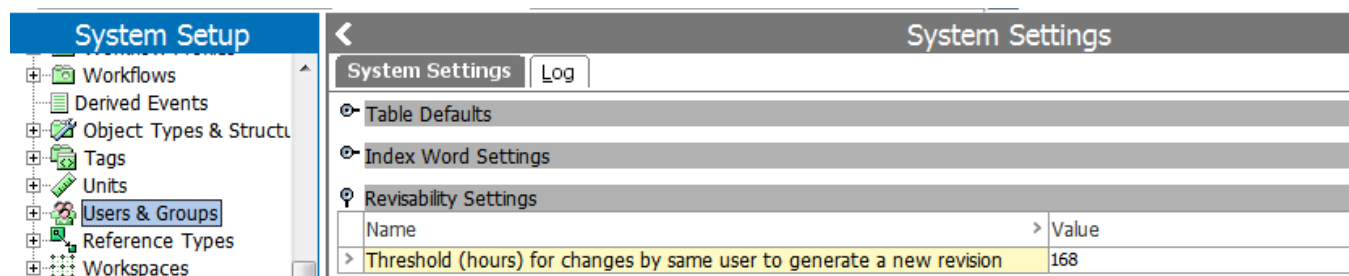
- Externally maintained data and calculated data are exceptions to the revision generation options mentioned above. Changes to the value of an externally maintained or calculated attribute value on an object do *not* generate a revision for that object, do *not* change the approval status for the object, and are *not* retained as part of the revision history of the object.
- When objects that are part of the revision history for another object are deleted, these are *not* retained with the revision history. For example, if an object had an attribute value populated at the time of a revision, and subsequently that attribute is deleted from STEP, reverting to the previous revision will *not* include a value for that attribute as the attribute no longer exists. The same applies for, e.g. workflows that included business rules that are subsequently deleted, objects that referenced other objects or assets that were subsequently deleted, etc.

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.

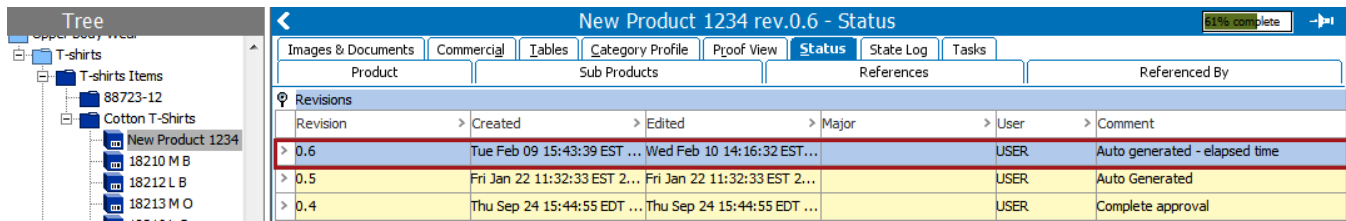
This number can be changed by first selecting the **System Setup** tab, navigating to the **Users and Groups** node, and finding the **Revisability Settings** flipper. The interval can be changed by editing the value in the Threshold parameter.



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.

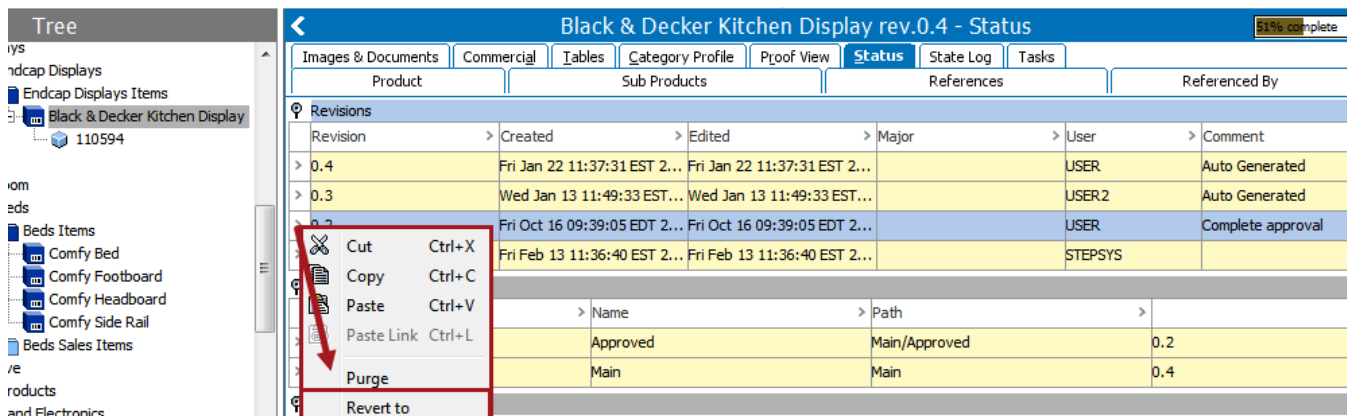


Revision	Created	Edited	Major	User	Comment
> 0.6	Tue Feb 09 15:43:39 EST ...	Wed Feb 10 14:16:32 EST...		USER	Auto generated - elapsed time
> 0.5	Fri Jan 22 11:32:33 EST 2...	Fri Jan 22 11:32:33 EST 2...		USER	Auto Generated
> 0.4	Thu Sep 24 15:44:55 EDT ...	Thu Sep 24 15:44:55 EDT ...		USER	Complete approval

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.

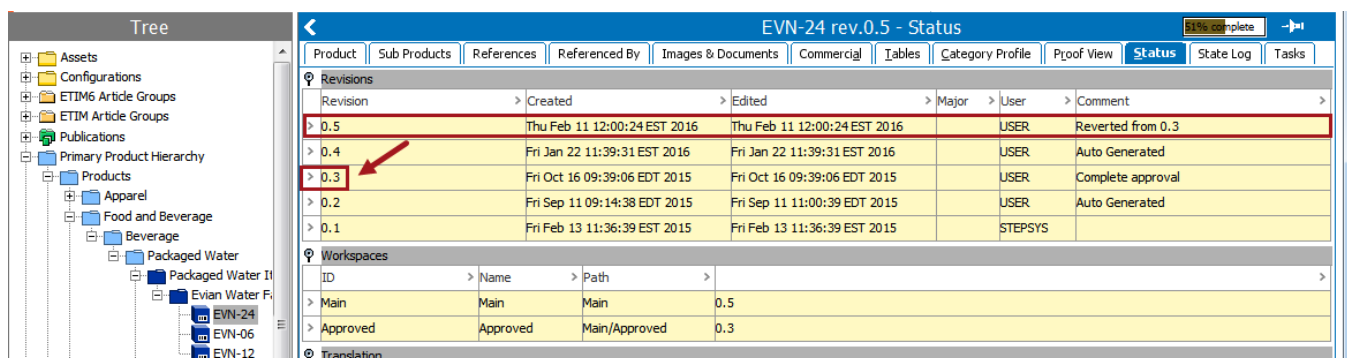
Reverting to Past Revisions

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.



Revision	Created	Edited	Major	User	Comment
> 0.4	Fri Jan 22 11:37:31 EST 2...	Fri Jan 22 11:37:31 EST 2...		USER	Auto Generated
> 0.3	Wed Jan 13 11:49:33 EST...	Wed Jan 13 11:49:33 EST...		USER2	Auto Generated
> 0.2	Fri Oct 16 09:39:05 EDT 2...	Fri Oct 16 09:39:05 EDT 2...		USER	Complete approval
> 0.1	Fri Feb 13 11:36:40 EST 2...	Fri Feb 13 11:36:40 EST 2...		STEPSYS	

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.



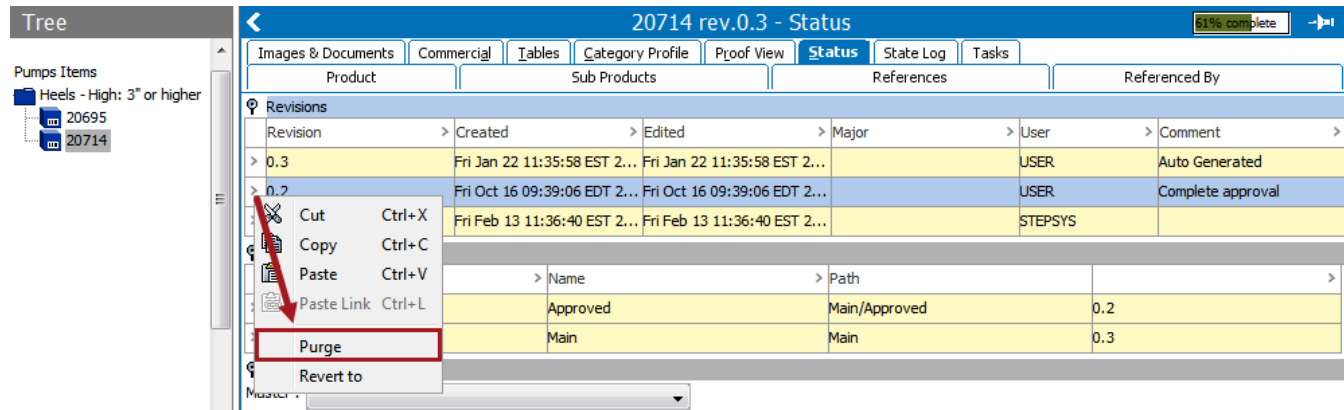
Revision	Created	Edited	Major	User	Comment
> 0.5	Thu Feb 11 12:00:24 EST 2016	Thu Feb 11 12:00:24 EST 2016		USER	Reverted from 0.3
> 0.4	Fri Jan 22 11:39:31 EST 2016	Fri Jan 22 11:39:31 EST 2016		USER	Auto Generated
> 0.3	Fri Oct 16 09:39:06 EDT 2015	Fri Oct 16 09:39:06 EDT 2015		USER	Complete approval
> 0.2	Fri Sep 11 09:14:38 EDT 2015	Fri Sep 11 11:00:39 EDT 2015		USER	Auto Generated
> 0.1	Fri Feb 13 11:36:39 EST 2015	Fri Feb 13 11:36:39 EST 2015		STEPSYS	

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**.

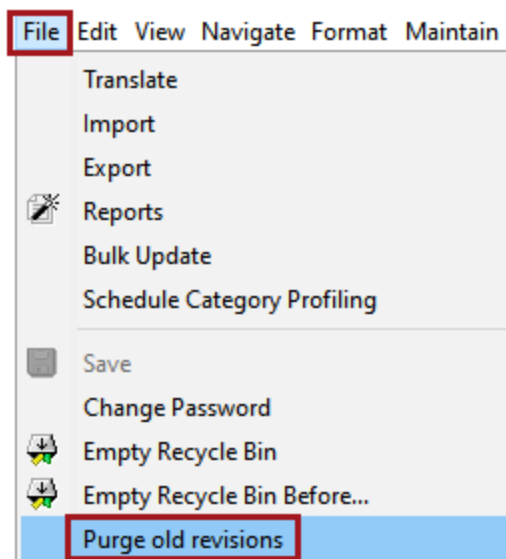


Revision Maintenance and Cleanup

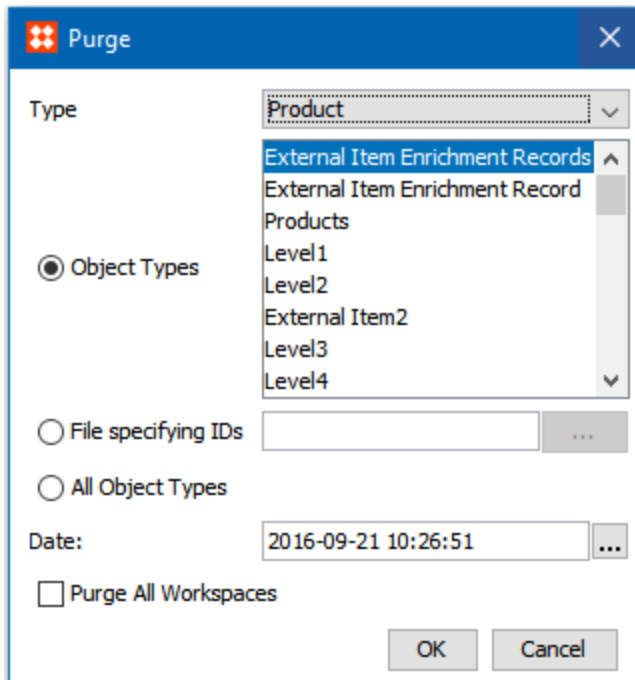
Planning for maintenance and cleanup of revisions is a critical step for all implementations and is best to be carried out early on in the setup of any system.

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.



The Purge dialog allows users to specify parameters for the revisions to be purged.



1. First select which Type of object should be purged.

Type: Select the type of object for which revisions should be purged. Depending on what is selected, will determine what object type is purged no matter which of the three radio button options is selected.

The available selections for 'Type' are self-explanatory: Product, Classification, Asset, DTP Documents, and System Configuration. Selecting any type filters the subsequent Object Types selection accordingly.

2. Next select one of the following three options to specify the objects that should have their revisions purged:
 - **Object Types:** Select this radio button and one or more object types to have revisions purged for all objects of that type. Multiple object types can be selected using Ctrl + Click. The available selections are filtered using the Type dropdown above.
 - **File specifying IDs:** Select this to load a file with the IDs of the objects that should have revisions purged. Revision purging will be carried out only for those objects where a matching STEP ID can be found. As STEP IDs are unique within object super types (e.g. Products, Classifications, etc), care should be taken using this option. For example: If there is an asset and a product that both have the same STEP ID, but the 'Product' object type was selected from the Type dropdown, then only the product would have its revisions purged. However, because both the asset and the product share the same ID, the user needs to pay attention to the 'Type' they choose, or they could accidentally purge the wrong revisions. Files should be a .TXT file type with one ID per row.
 - **All Object Types:** Select this option to have revisions purged across all objects in STEP within the 'Type' chosen from the dropdown menu.
3. **Date:** This allows the user to specify up to what date / time the revisions should be purged. This should be set after selecting one of the three radio button options. If there is no revision that meets the date exactly, it will

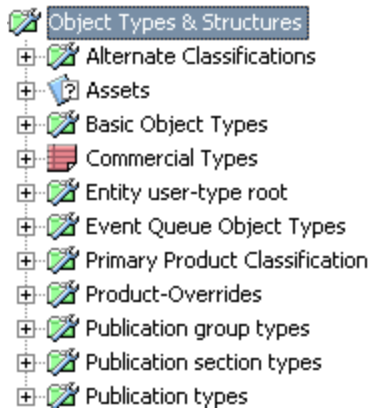
keep the most recent revision. Ex: User selects today, but the last revision was yesterday. The system would keep yesterdays revision because it is most recent.

4. Decide if **Purge All Workspaces** should be checked. If this is checked, this would get rid of revisions made in all workspaces, ex: Approved, Main. If it is left unchecked, it will only delete the revisions in the current workspace. For more on workspaces see the Workspaces topic in the System Setup / STEP Super User Guide documentation.

Object Types and Structures

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 / System Setup Object Type	A Basic Object Type is applied to nodes that are used for structuring the <i>system data</i>. 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 Object Type / Tree Object Type	A non-basic object type applies to nodes that are used for structuring the product data hierarchies including:

Object Type kind	Description
	• 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.

Open Sales Item - Object Type	
Object Type References Log	
Description	
Name	Value
ID	OpenSalesItem
Name	Open Sales Item
Last edited by	2015-07-31 10:34:00 by USER
Name Pattern	1 SalesItem [id]
ID Pattern	2 [id]
Icon	
Dimension Dependencies	
Condition	
Aspects	
Valid Attributes	

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.

Tree

- Assets
- Configurations
- Index Words
- Merchandising Hierarchy
- Suppliers
- Web Sites
- Entity Root
- Publications
- Primary Product Hierarchy
 - Products
 - Food and Beverage
 - Beverage
 - Packaged Water
 - Packaged Water Items
 - Packaged Water Sales Items
 - Evian Water Family
 - 7133-12
 - 7134-24
 - SalesItem 107807

SalesItem 107807 rev.0.1 - Product

Images & Documents

Commercial

Tables

Category Profile

Status

State Log

Tasks

Product

References

Description

Name	>	>	Value
ID		2	107807
Name		1	SalesItem 107807
Object Type			Open Sales Item
Revision			0.1 Last edited by USER on Tue Aug 11 14:34:40 EDT 2015
Approved			✗ Never Been Approved
Translation			Not Translated
Path			Primary Product Hierarchy/Products/Food and Beverage/Beverage/P
Group Information			abc
Marketing Description			fx

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.

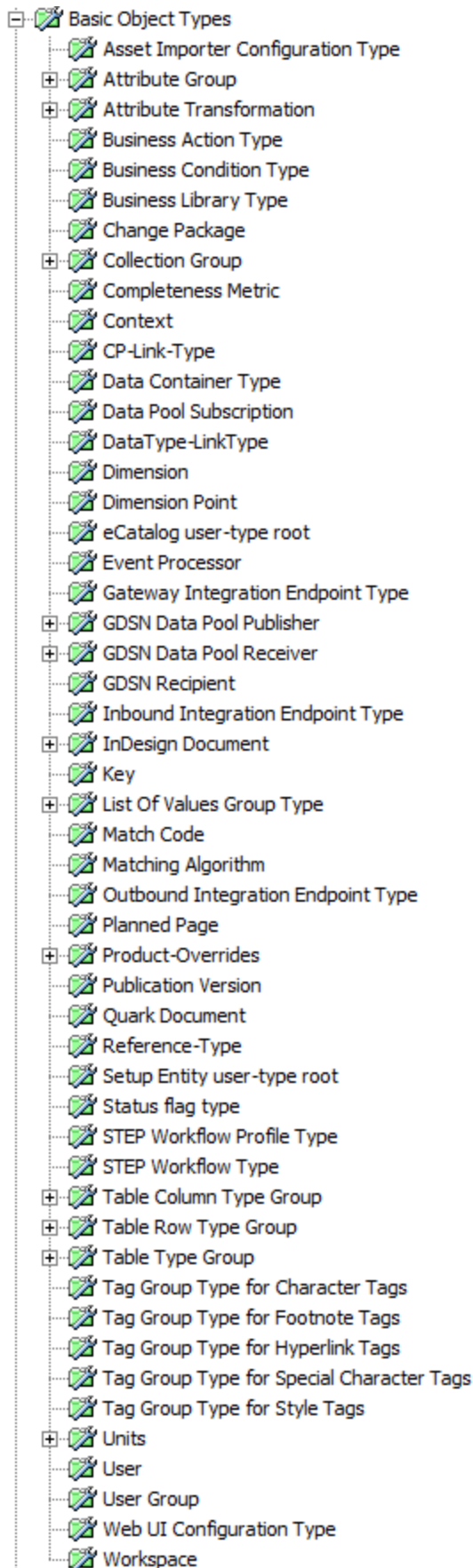
Basic Object Types

A Basic Object Type is a node used for structuring the system data, for example, Attributes, Contexts, Users, Tables.

Whenever an Object is created using a Basic Object Type, the relevant Object Type is automatically applied.

Basic Object Types cannot be manually created, however when additional components are purchased and added to STEP, additional Basic Object Types may appear.

Below is one example of Basic Object Types in a STEP instance.



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.

<

Open Sales Item - Object Type

→▶

Object Type			References	Log
Description				
	Name	>	>	Value
>	ID			OpenSalesItem
>	Name			Open Sales Item
>	Last edited by			2015-08-11 14:23:17 by USER
>	Name Pattern			SalesItem [id]
>	ID Pattern			[id]
>	Icon			
>	Dimension Dependencies			Language;
>	Group Information		abc	

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.

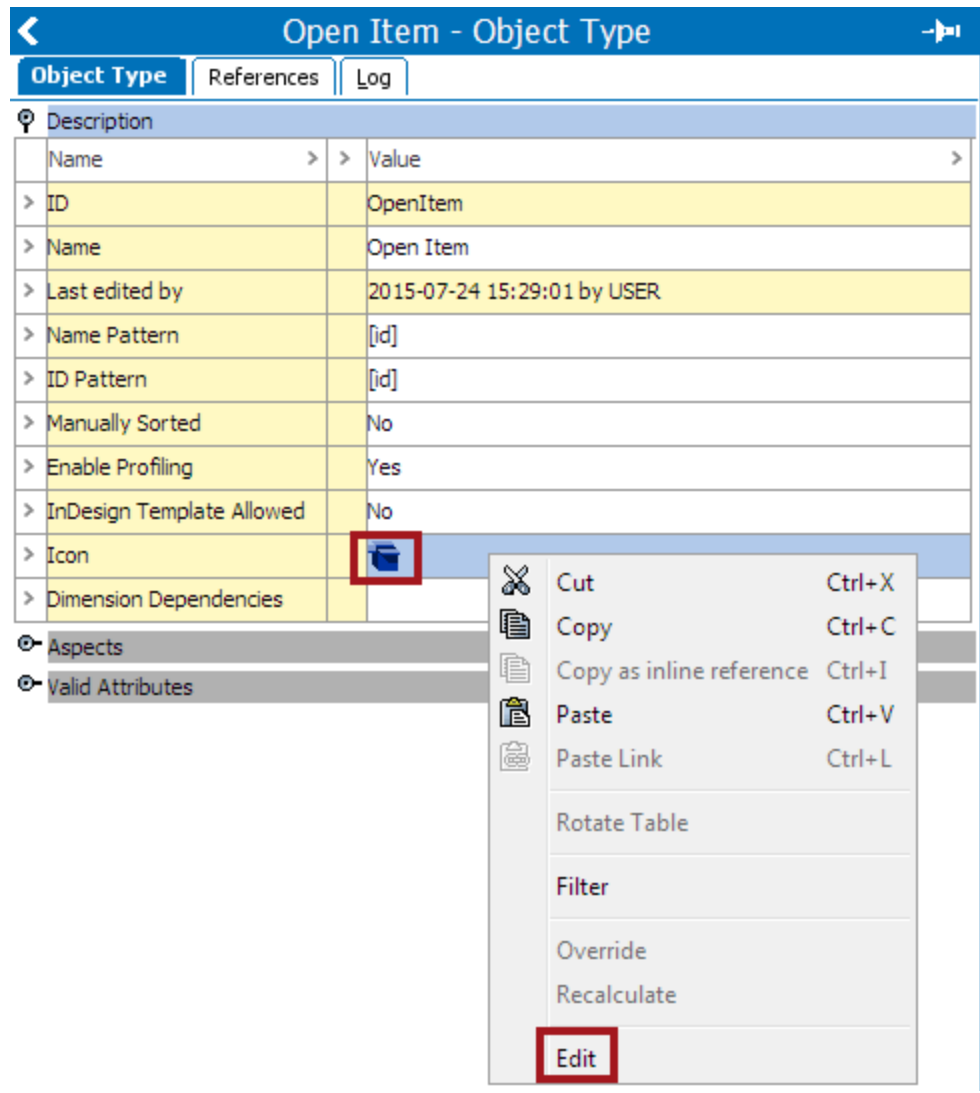
For more information, see **Maintaining Dimension Dependent Object Types** in the **Dimensions, Dimension Points, and Contexts** documentation.

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.

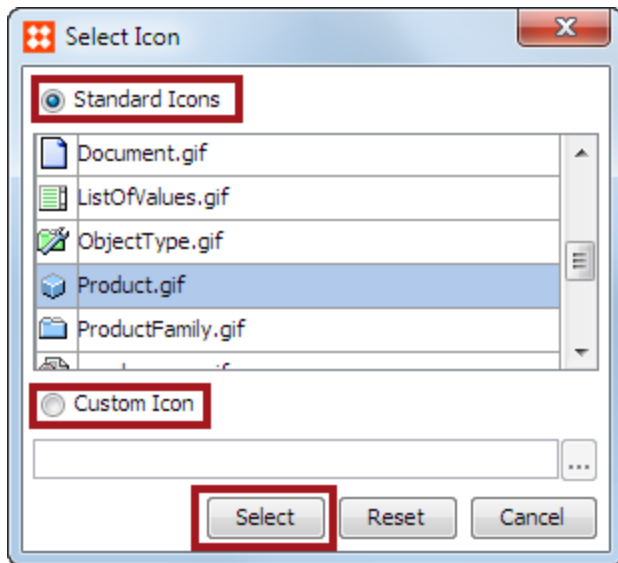
Editing an Object Type Icon

- 1. On the System Setup tab, select the Icon field of an Object, right-click and select the **Edit** option.

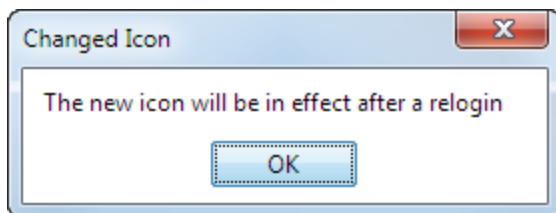


- 2. Choose a standard or custom icon and click **Select**.

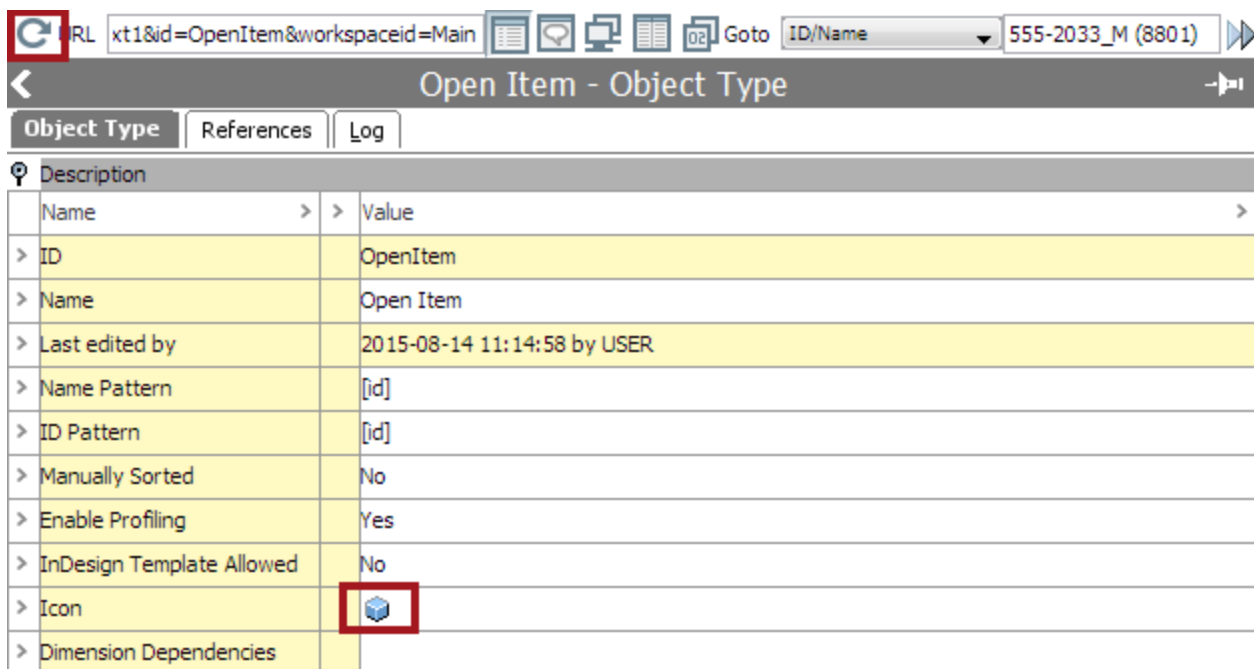
Note: The maximum icon size is 16 x 16 pixels.



- Click **OK** on the Changed Icon confirmation dialog.



- On the Object Type, click the **Refresh** button to see the new icon. Logging out is not necessary.



Maintaining Object Types

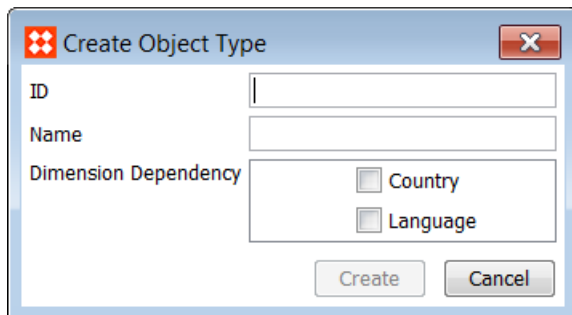
Object Types are maintained in System Setup under **Object Types & Structures**.

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 **Maintaining Dimension Dependent Object Types** topic.
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

Maintaining Dimension Dependent Object Types

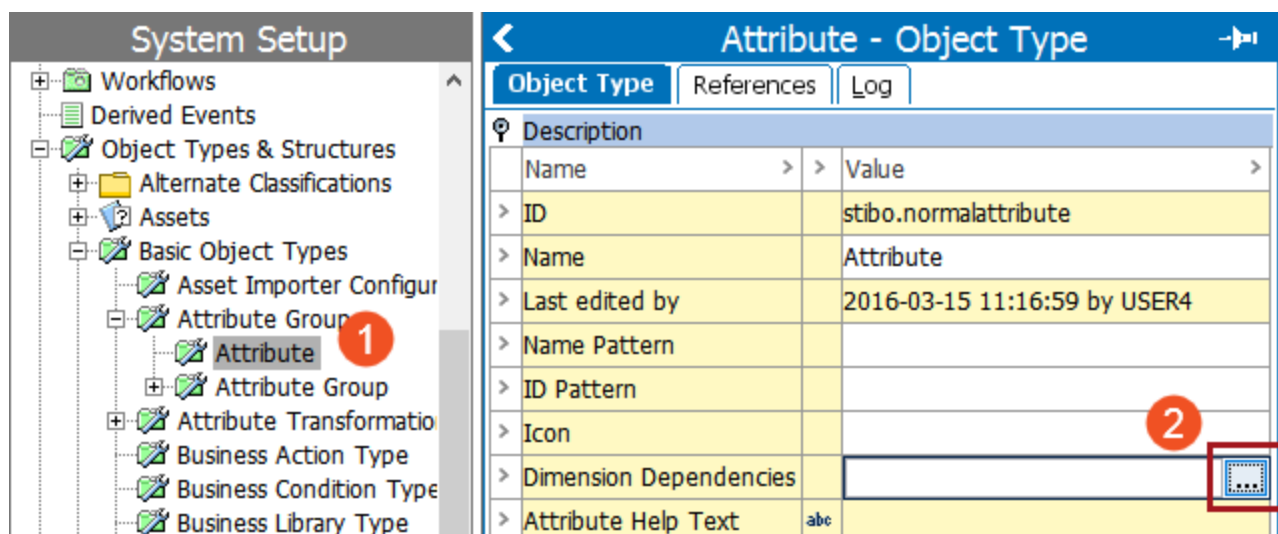
Dimension dependency is set for objects on their respective object types in **System Setup**. Almost all object types in STEP—both **basic** and **non-basic**—can be made dimension dependent so that their **names** will display in different languages in different contexts.

- **Basic object types:** Including attribute groups, attributes, contexts, dimensions, users, and more (located under Object Types & Structures > Basic Object Types)
- **Non-basic object types:** Including alternate classifications, primary product classifications, assets, reference types, LOVs, entities, commercial types, product-overrides, publication object types, and more

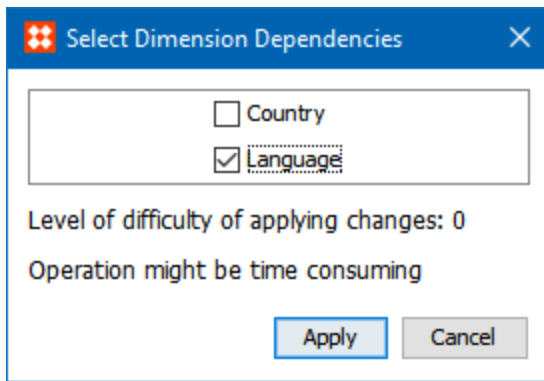
Making an Object Type Dimension Dependent

The following example uses the **Attribute** object type (located under Object Types & Structures > Basic Object Types > Attribute) to show how to make an object type dimension dependent in STEP. The same steps apply for any object type, and as such, directions will not be provided for every object type.

1. In System Setup, navigate to object type that you would like to make dimension dependent.
2. On the **Object Type** tab, double-click in the **Dimension Dependencies** field, then click the ellipsis button (...).



3. In the **Select Dimension Dependencies** dialog box that displays, check the box(es) for the relevant dimension(s), then click **Apply**.



A dialog box titled "Select Dimension Dependencies" with a close button (X) in the top right corner. It contains two checkboxes: "Country" (unchecked) and "Language" (checked). Below the checkboxes, it states "Level of difficulty of applying changes: 0" and "Operation might be time consuming". At the bottom, there are "Apply" and "Cancel" buttons.

- The object type is now dimension dependent, and the STEP names of any object using this object type are now dimension dependent.

Attribute - Object Type		
Object Type References Log		
Description		
Name	>	Value
ID	>	stibo.normalattribute
Name	>	Attribute
Last edited by	>	2017-02-27 16:31:50 by USER4
Name Pattern	>	
ID Pattern	>	
Icon	>	
Dimension Dependencies	>	Language;
Attribute Help Text	abc	

Removing Dimension Dependency From an Object Type

Dimension dependency is removed from an object in a near-identical fashion that it is applied.

Note: The dimension dependency should normally not be changed. Be aware of possible conflicts if a dimension dependency is removed.

- Navigate to the relevant object type in System Setup.
- Double-click in the **Dimension Dependencies** field, then click the ellipsis button (...).

Attribute - Object Type		
Object Type References Log		
Description		
Name	> >	Value
ID		stibo.normalattribute
Name		Attribute
Last edited by		2017-02-27 16:31:50 by SU1
Name Pattern		
ID Pattern		
Icon		
Dimension Dependencies		Language; 
Attribute Help Text	abc	

- In the **Select Dimension Dependencies** dialog that displays, uncheck the box(es) for the relevant dimension (s), then click **Apply**.

 Select Dimension Dependencies

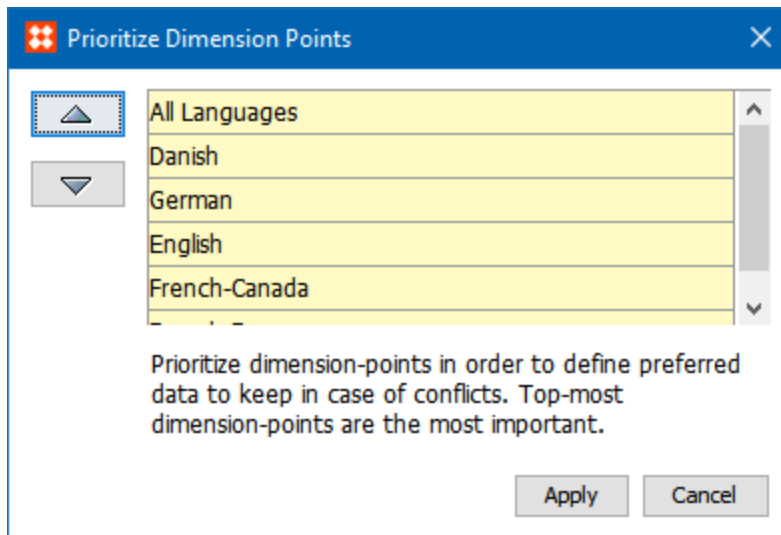
☐ Country
☒ Language

Level of difficulty of applying changes: 0

Operation might be time consuming

Note: The dialog box indicates the complexity of removing the dimensions. If values exist in multiple dimensions and dimension points, it can be time consuming to remove the dimension dependency. Difficulty 0-5 indicates the complexity of the operation, where a difficulty of 5 is a very complex operation.

- If there is a conflict when removing a dimension dependency, a dialog box displays 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.



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. Otherwise, the system will look at the dimension points with lower priority to find a value to keep.

- Click **Apply** to remove the dimension dependency from the object type.

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.

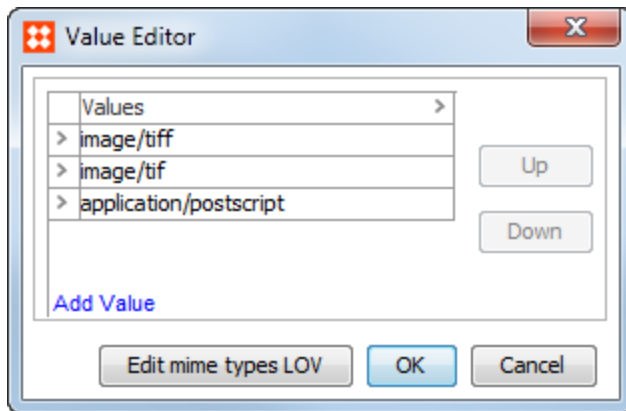
Note: STEP determines that a file is an 'image' if it has a MIME type of **image/***.

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

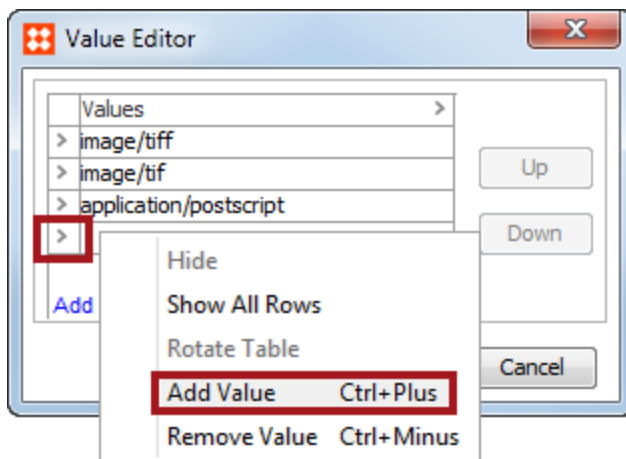
MIME Type	Definition
image/gif	GIF Image
image/jpeg	JPEG JFIF image
image/jpg	JPEG JFIF image
image/png	Portable Networks Graphic
image/tif	TIF Image
image/tiff	TIF Image
text/*	All Text Documents
text/html	Textual data in HTML format
text/rtf	Textual data in rich text format
Text/ plain;charset=ISO-8859-1	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

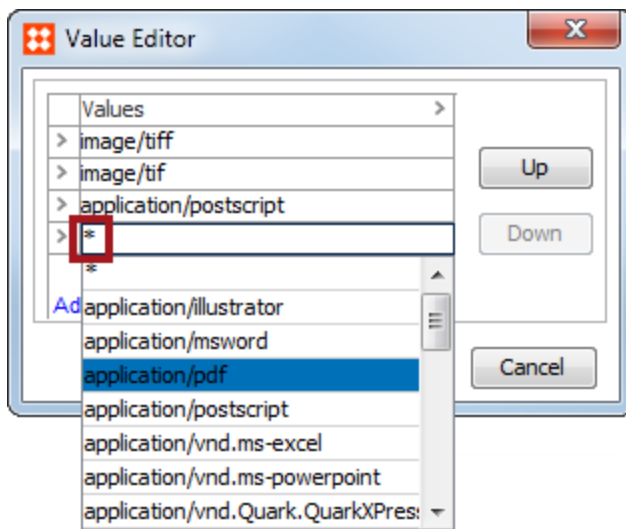
1. In System Setup, select **Object Types & Structures**, then **Asset Type Root**, and select an Image or Document Object Type. Mime Types are not valid for any other object types.
The corresponding editor appears.
2. Double-click in the **MIME Types** field to display the **Value Editor** dialog.



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**.



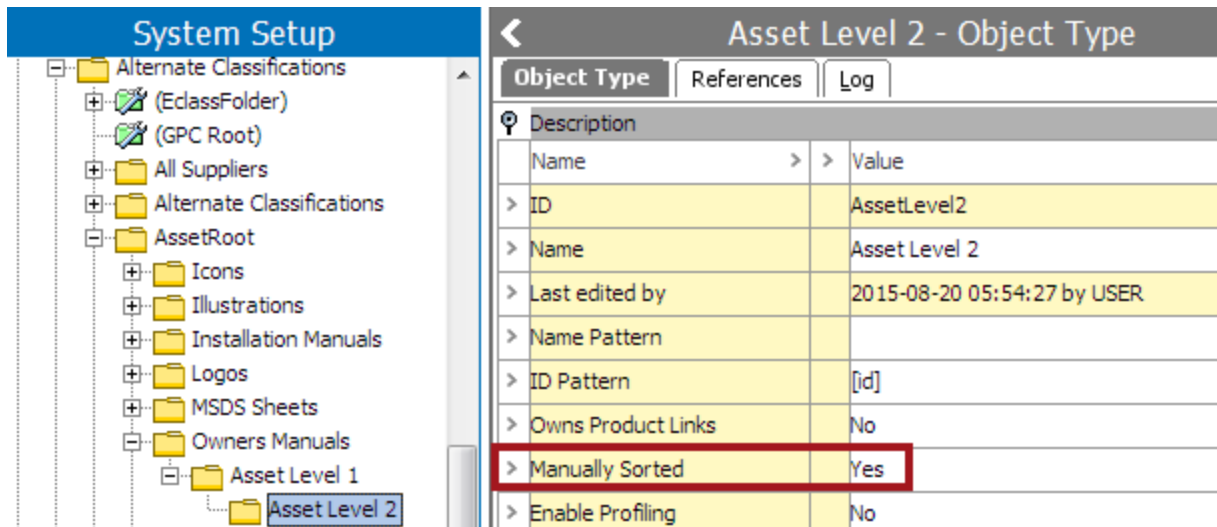
5. Repeat steps 3 and 4 to select more MIME Types.
6. Click **OK** to display the Object Type Editor assigned MIME Type(s).

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.

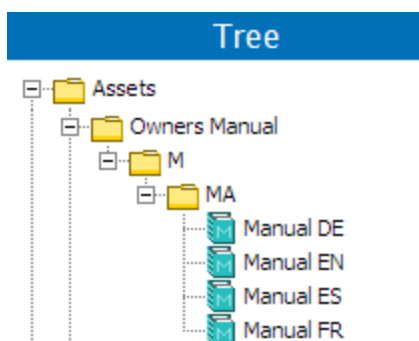


The screenshot shows the 'System Setup' window with the 'Asset Level 2 - Object Type' configuration. The 'Object Type' tab is selected, and the 'Description' section is visible. The 'Manually Sorted' field is highlighted with a red box, indicating it is set to 'Yes'.

Name	Value
ID	AssetLevel2
Name	Asset Level 2
Last edited by	2015-08-20 05:54:27 by USER
Name Pattern	
ID Pattern	[id]
Owns Product Links	No
Manually Sorted	Yes
Enable Profiling	No

2. Set the **Manually Sorted** field:

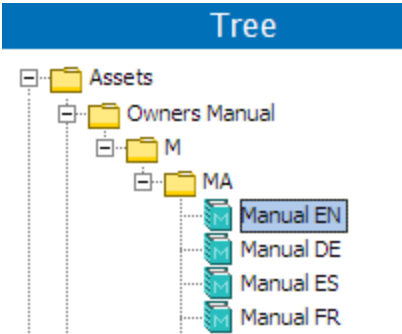
- **No** means children of the object are ordered alphabetically in the Tree. Their sequence cannot be manually changed by a user.



The screenshot shows the 'Tree' view with the following hierarchy:

- Assets
 - Owners Manual
 - M
 - MA
 - Manual DE
 - Manual EN
 - Manual ES
 - Manual FR

- **Yes** means a user can manually change the sequence of child objects in the tree hierarchy by dragging an object to another position.



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

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.

The screenshot displays the Stibo Systems interface for configuring Object Type Hierarchy. On the left, the 'System Setup' tree shows various object types, with 'Workflows' highlighted. On the right, the 'Workflows - References' tab is active, showing a table of parents for the 'Workflows' object type. The table has columns for 'ID' and 'Name'. The 'Workflows' object type is listed as its own parent, indicating a recursive hierarchy.

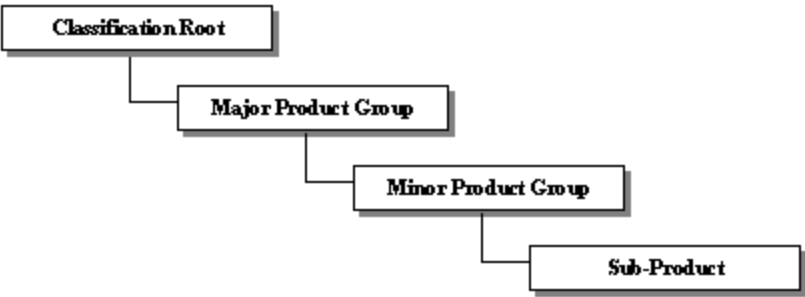
ID	Name
Workflows	Workflows
Setup Group user-type root	Setup Group type root

Classification and Product Hierarchy Structures

The two hierarchy structures that may be defined are:

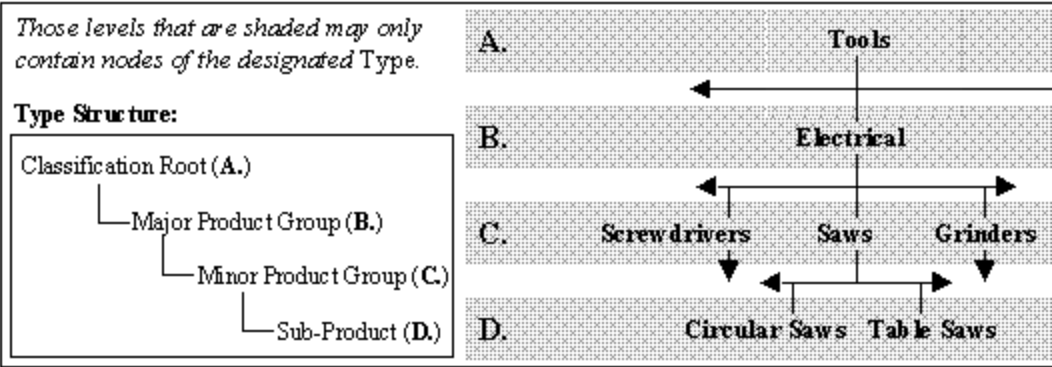
- Classification hierarchy
- Product hierarchy

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.

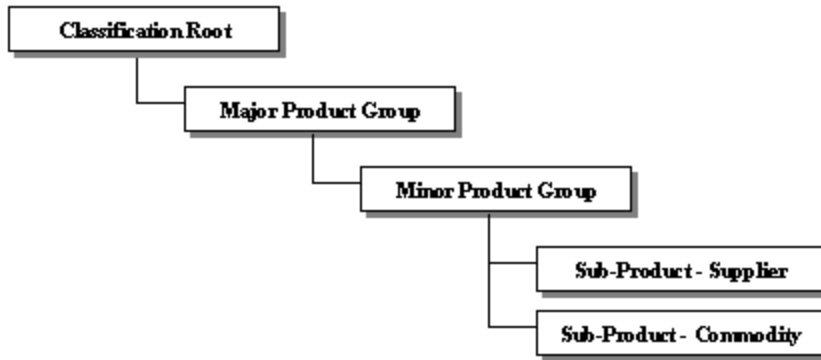


In this example, only a **Major Product Group** may be a child of a **Classification Root**, a **Minor Product Group** may only be a child of a **Major Product Group**, etc. As defined by this structure, it is not possible to create a **Sub-Product** under the **Classification Root**. Any nodes of the 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:



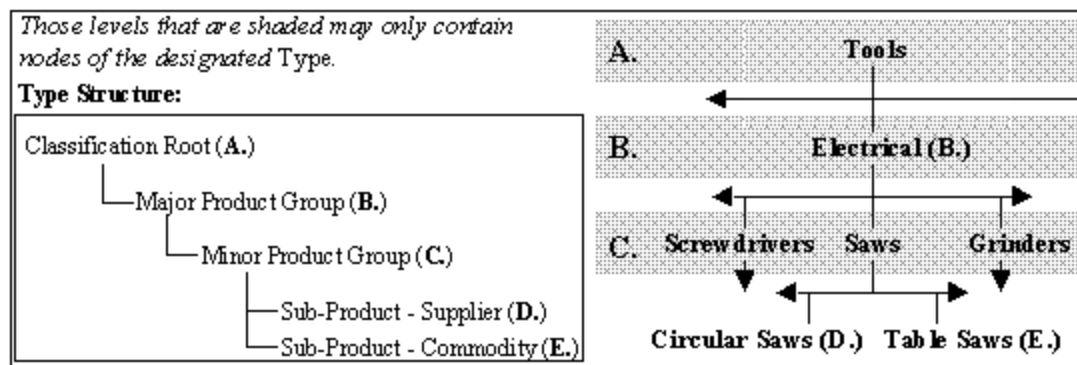
The first example illustrates a simple structure where each Object Type defines a level in the hierarchy. To build on this illustration, the second example shows it is also possible to create an Object Type structure in which a parent Object Type may have multiple children.



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**

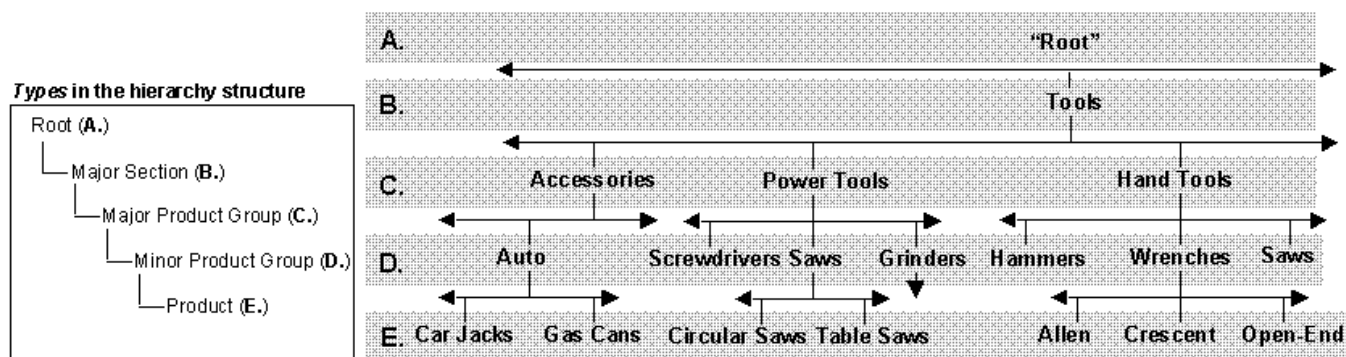
Fixed Hierarchy Structure

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.

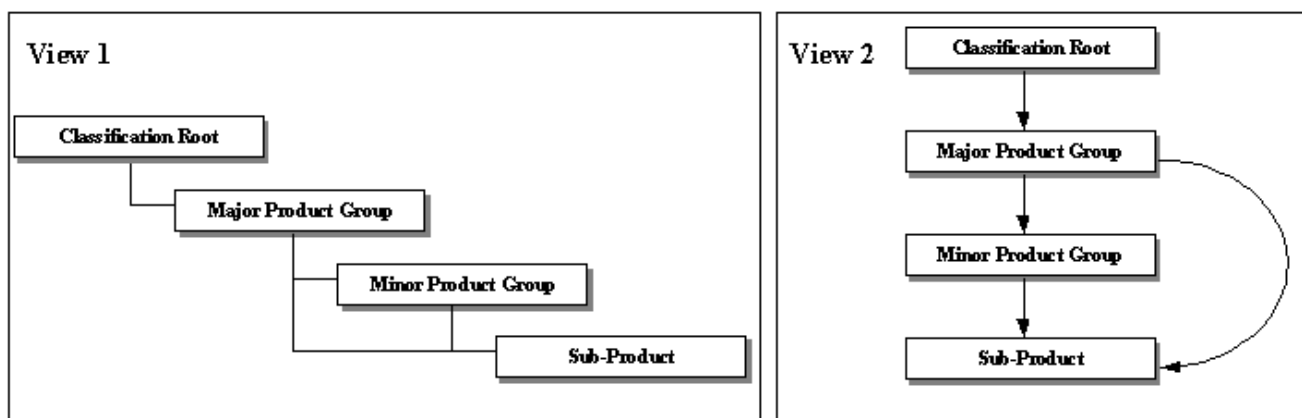
In this example, the 'Screwdrivers' node is of the Object Type called 'Minor Product Group' and the 'Table Saws' node has been given 'Product' as its Object Type. In the hierarchy structure, then, the setup is such that there is a fixed number of levels, and that at each level there is only one 'child' 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

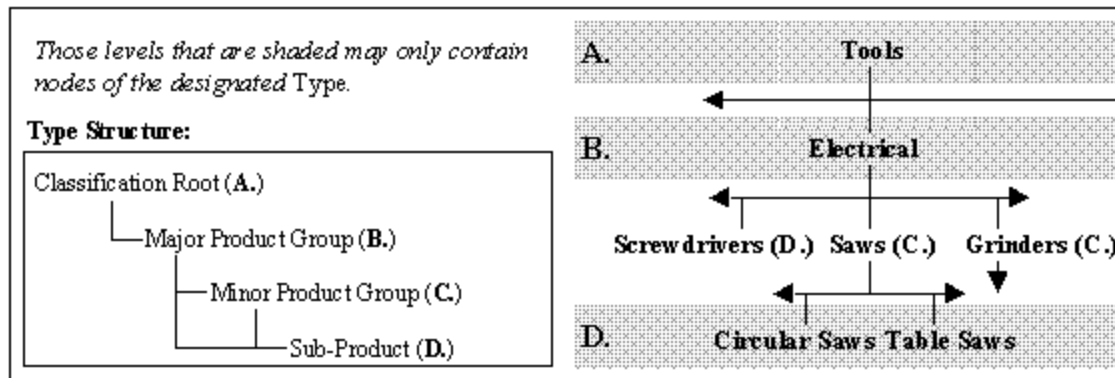
In some instances, the depth of the hierarchy may vary. For example, the leaf nodes are of the same Object Type, but may be created in different levels. This is accomplished by allowing an Object Type to have multiple parents and displayed in the figure below.



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**.

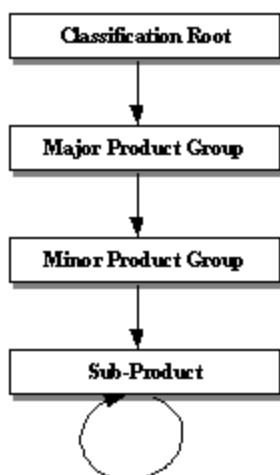
The following figure represents a hierarchy based on a varying Object Type structure.



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.

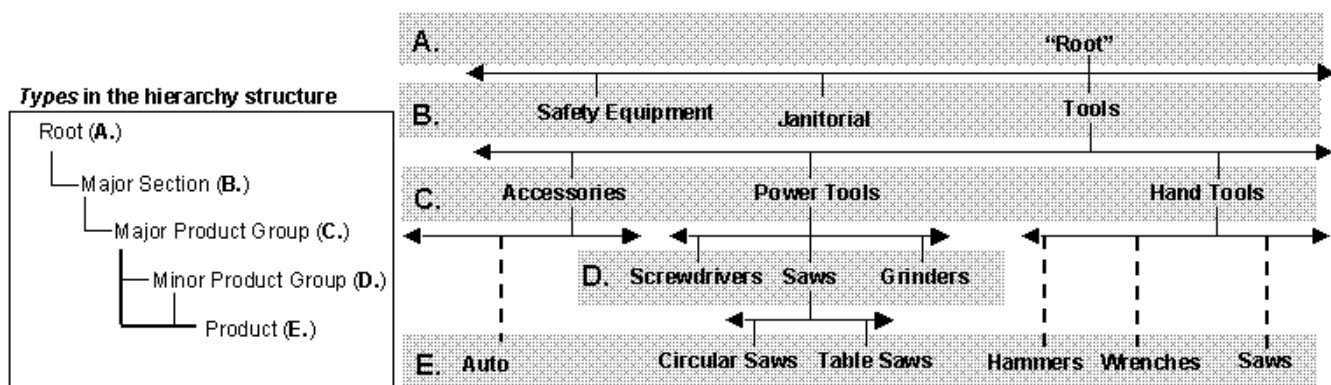


Flexible Hierarchy Structure

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'.

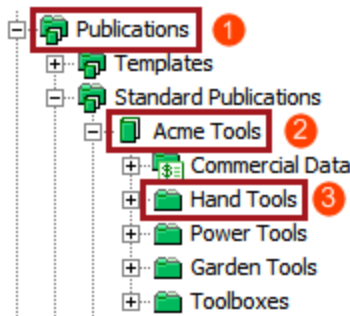


Creating a Standard Publication Object Type Hierarchy

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.

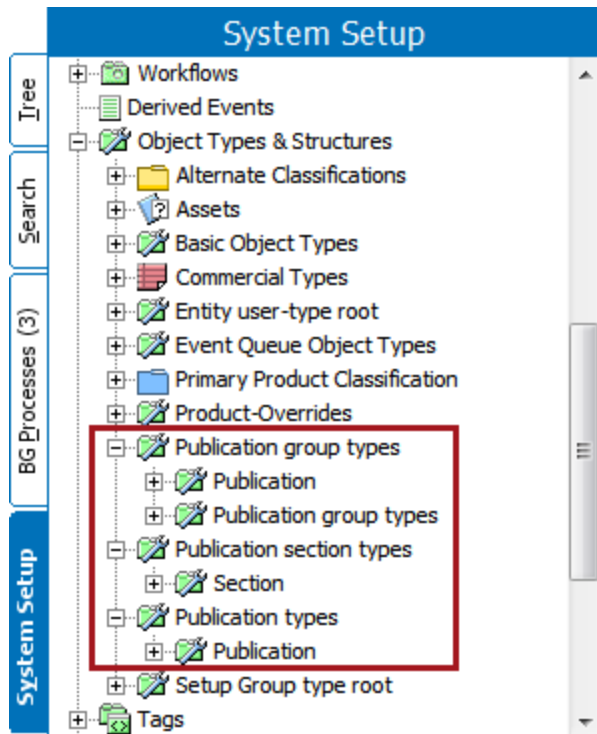
The following screenshot shows what a basic publication hierarchy looks like in the Tree. The publication object types addressed in this documentation are used to build the following objects:

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

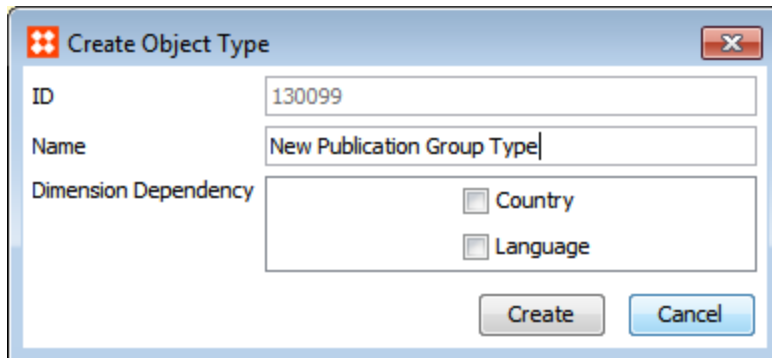
The following sections of this documentation will guide you through the process of creating new publication hierarchy object types as well as configuring valid parent / child relationships between these object types.

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.

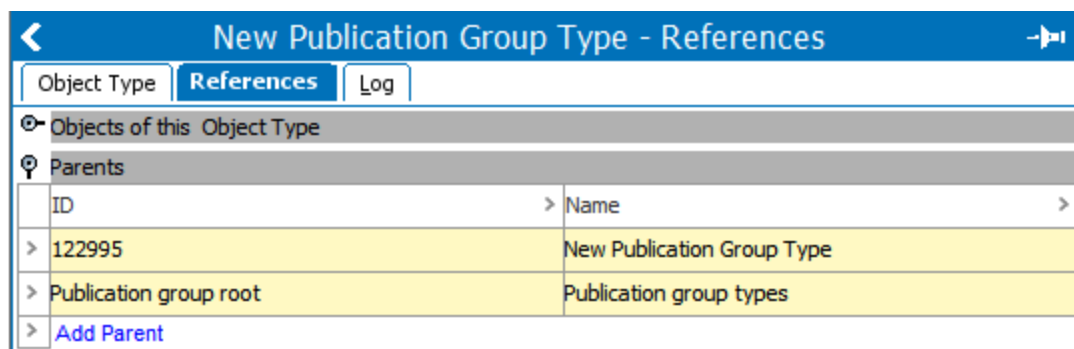
Creating a Publication Group Object Type

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.



The 'Create Object Type' dialog box is shown. It has a title bar with a red 'X' button. The 'ID' field contains '130099'. The 'Name' field contains 'New Publication Group Type'. The 'Dimension Dependency' section has two checkboxes: 'Country' and 'Language', both of which are unchecked. At the bottom are 'Create' and 'Cancel' buttons.

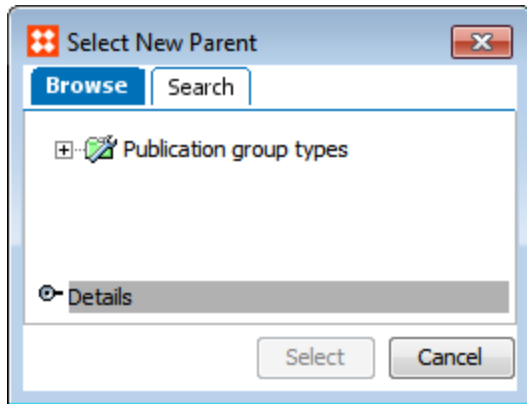
3. If the STEP ID for the **Publication group root** object has not been set to autogenerate with an ID Pattern, type an ID in the **ID** field.
4. In the **Name** field, type a name.
5. Check the relevant **Dimension Dependency** check boxes, if applicable. (Dimension dependencies are optional, and 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.



The 'New Publication Group Type - References' dialog box is shown. It has a title bar with a back arrow and a forward arrow. The 'References' tab is selected. Below the tabs are three sections: 'Objects of this Object Type', 'Parents', and a table. The 'Parents' section shows a list of parent objects. The table has two columns: 'ID' and 'Name'. The first row shows '122995' as the ID and 'New Publication Group Type' as the Name. The second row shows 'Publication group root' as the ID and 'Publication group types' as the Name. At the bottom is an 'Add Parent' button.

ID	Name
> 122995	New Publication Group Type
> Publication group root	Publication group types

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.

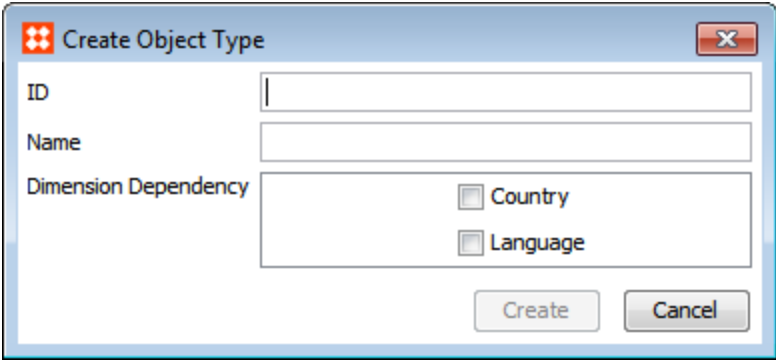


9. Search or browse for the relevant publication group type, then click **Select**.

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

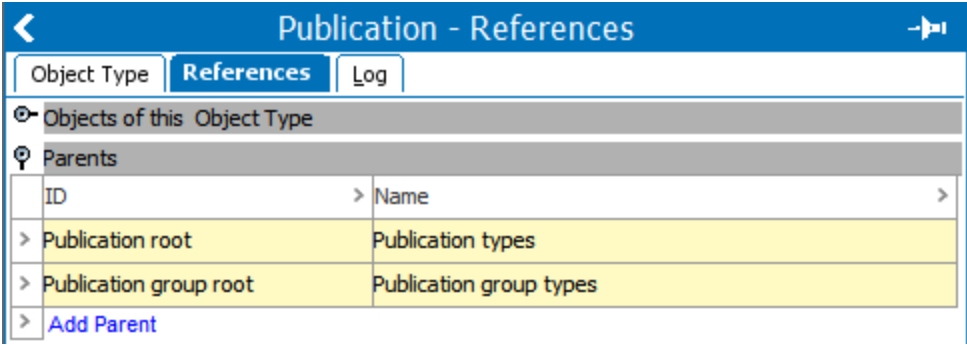
- 1. In System Setup, open the **Object Types & Structure** hierarchy.
- 2. Click **Publication types**, right-click, and then click **New Object Type**. A **Create Object Type** dialog box appears.



The 'Create Object Type' dialog box contains the following fields and controls:

- ID**: A text input field.
- Name**: A text input field.
- Dimension Dependency**: A section containing two checkboxes: ☐ **Country** and ☐ **Language**.
- Create** and **Cancel** buttons at the bottom right.

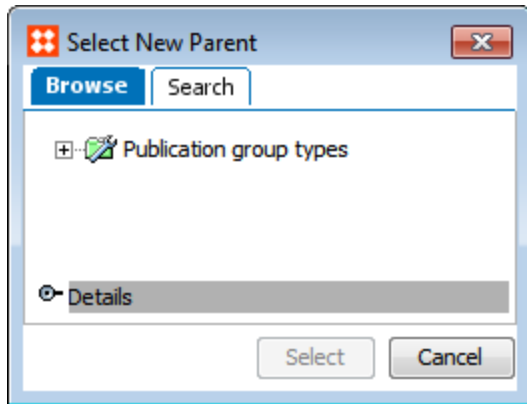
- 3. If the STEP ID for the 'Publication root' object has not been set to autogenerate with an ID Pattern, type an ID in the **ID** field.
- 4. In the **Name** field, type a name.
- 5. Check the relevant **Dimension Dependency** check boxes, if applicable. (Dimension dependencies are optional, and are 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.



The 'Publication - References' tab shows a table of parent objects:

ID	Name
Publication root	Publication types
Publication group root	Publication group types
Add Parent	

- 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.

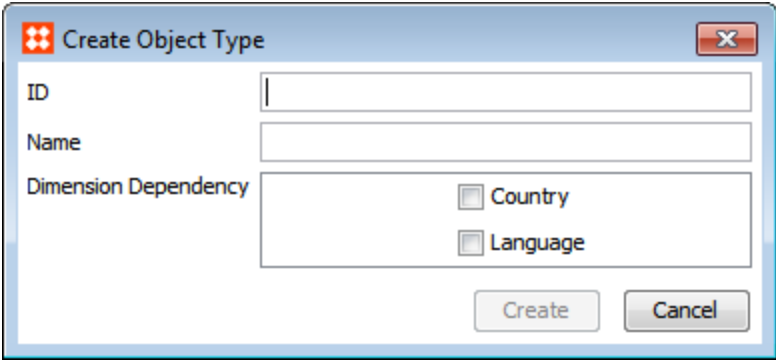


9. Search or browse for the relevant publication group type, then click **Select**.

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

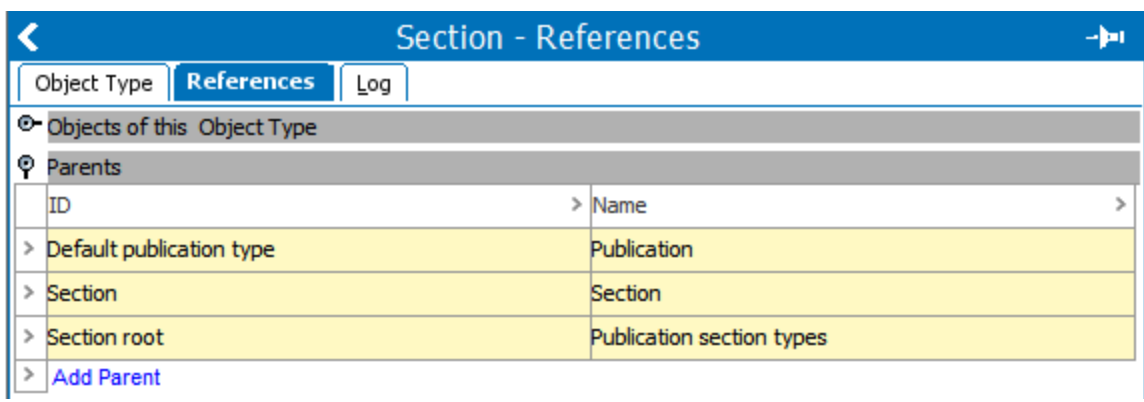
- 1. In System Setup, open **Object Types & Structures**.
- 2. Click **Publication section types**, right-click, and then select **New Object Type**. The **Create Object Type** dialog appears.



The 'Create Object Type' dialog box contains the following fields and controls:

- ID**: A text input field.
- Name**: A text input field.
- Dimension Dependency**: A container with two checkboxes:
 - ☐ Country
 - ☐ Language
- Create**: A button.
- Cancel**: A button.

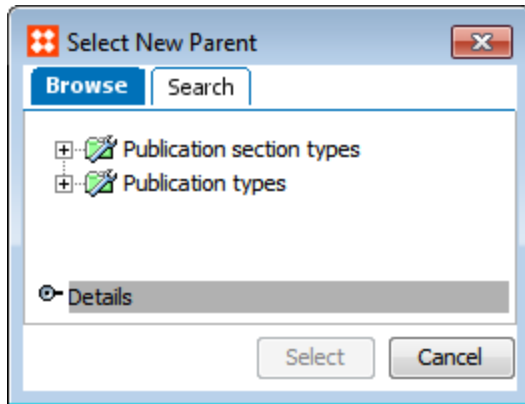
- 3. If the STEP ID for the 'Section root' object has not been set to autogenerate with an ID Pattern, type an ID in the **ID** field.
- 4. In the **Name** field, type a name.
- 5. Check the relevant **Dimension Dependency** check boxes, if applicable. (Dimension dependencies are optional, and are not 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.



The screenshot shows the 'Section - References' editor with the 'References' tab selected. It displays a table of parent objects.

ID	Name
> Default publication type	Publication
> Section	Section
> Section root	Publication section types
> Add Parent	

- 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.

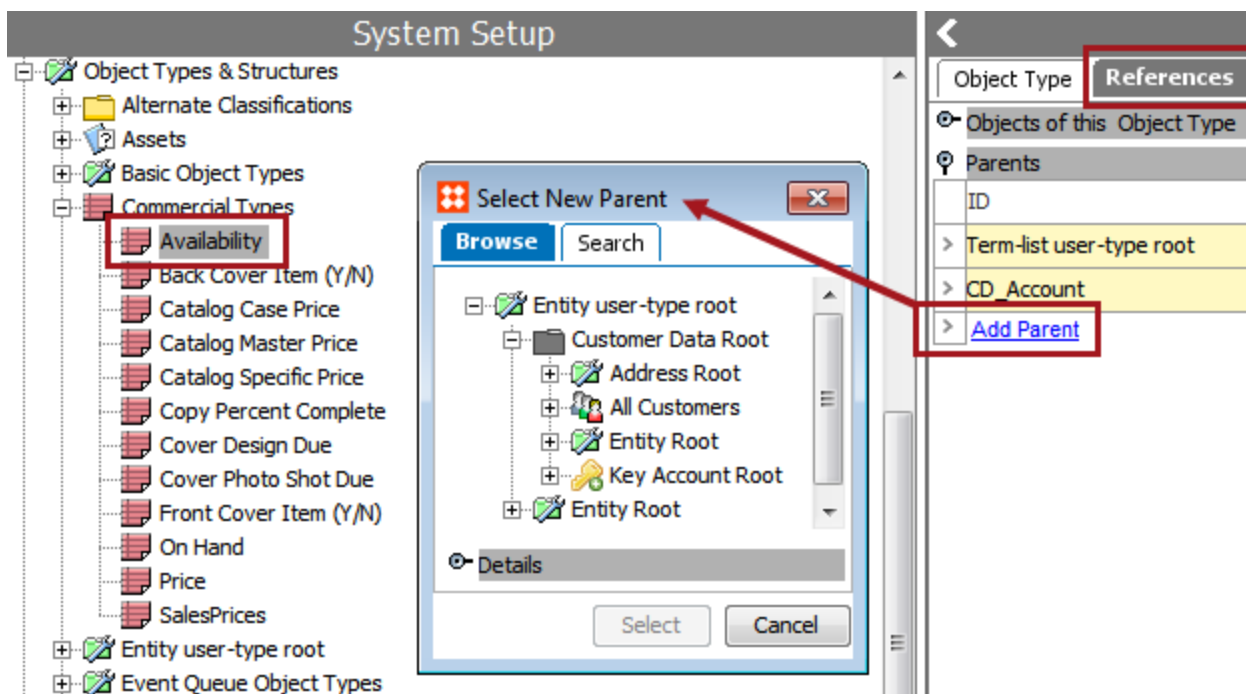
Product-Override Object Types and Commercial Object Types

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.)

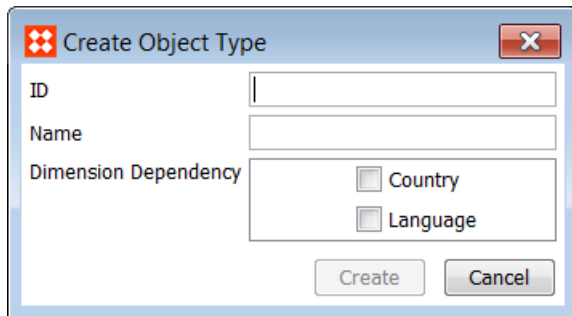
Link the Object Types from the References tab on either a Commercial Type or the References tab on a Product-Override by clicking the **Add Parent** link. A node selector displays a list of legal Object Types for linking.



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.



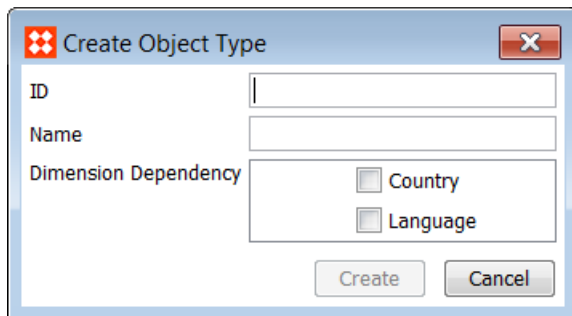
The screenshot shows a 'Create Object Type' dialog box. It has a title bar with a red 'X' button. The dialog contains three input fields: 'ID', 'Name', and 'Dimension Dependency'. The 'Dimension Dependency' field has two checkboxes: 'Country' and 'Language'. At the bottom are 'Create' and 'Cancel' buttons.

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.



The screenshot shows a Windows-style dialog box titled "Create Object Type". It has a standard title bar with a close button (X). The dialog contains three input fields: "ID", "Name", and "Dimension Dependency". The "Dimension Dependency" field is a group box containing two checkboxes: "Country" and "Language". At the bottom of the dialog are two buttons: "Create" and "Cancel".

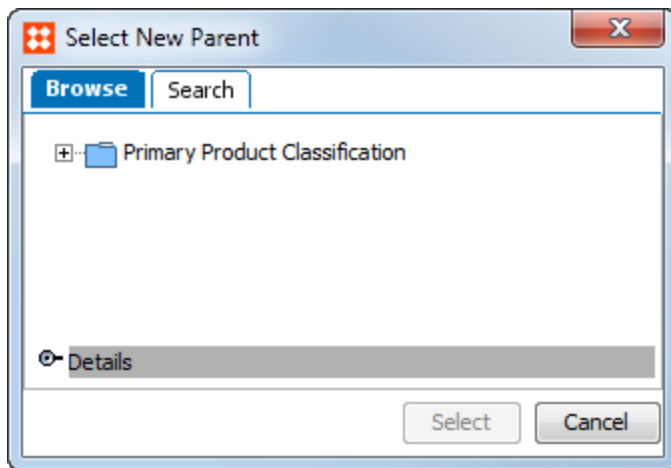
3. Type an **ID** and **Name** and optionally, check Dimension Dependencies.
4. Click **Create** and a new Product-Override Object Type is created.

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.

<
Asset Level 2 - Object Type

Object Type
References
Log

Description

Name	>	>	Value
ID			AssetLevel2
Name			Asset Level 2
Last edited by			2015-08-20 04:03:25 by USER
Name Pattern			
ID Pattern			[id]
Owns Product Links			No
Manually Sorted			No
Enable Profiling			No
InDesign Template Allowed			No
Icon			
Dimension Dependencies			
Purpose		abc	
URL		URL	

Aspects

Valid Attributes

ID	>	Name
URL		URL
Purpose		Purpose

Add Attribute

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:

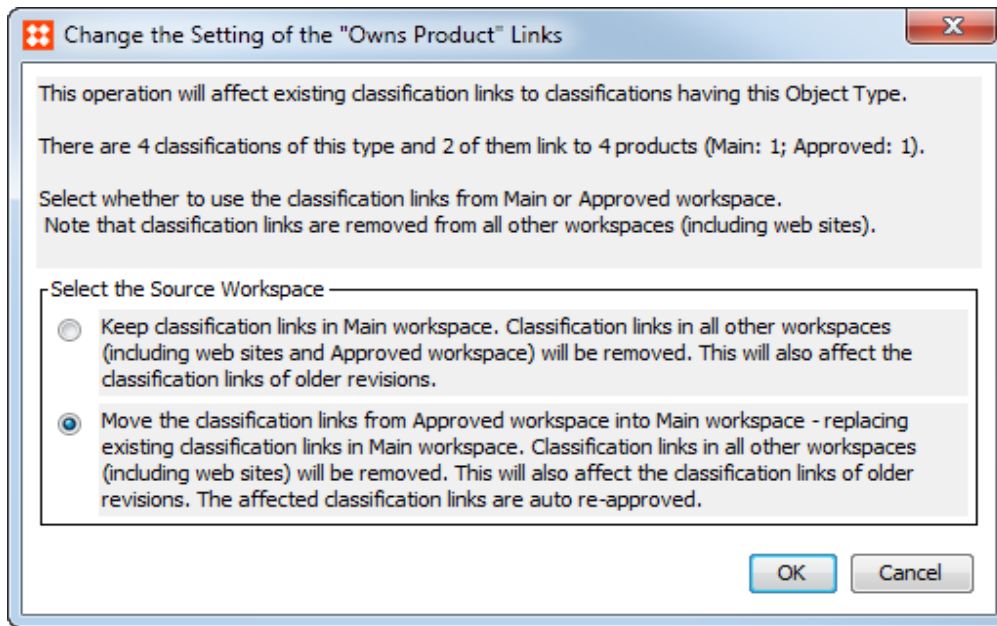
```
<STEP-ProductInformation>
<Products IncludeClassificationOwned="false">
<Product>
</Product>
</Products>
</STEP-ProductInformation>
```

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.

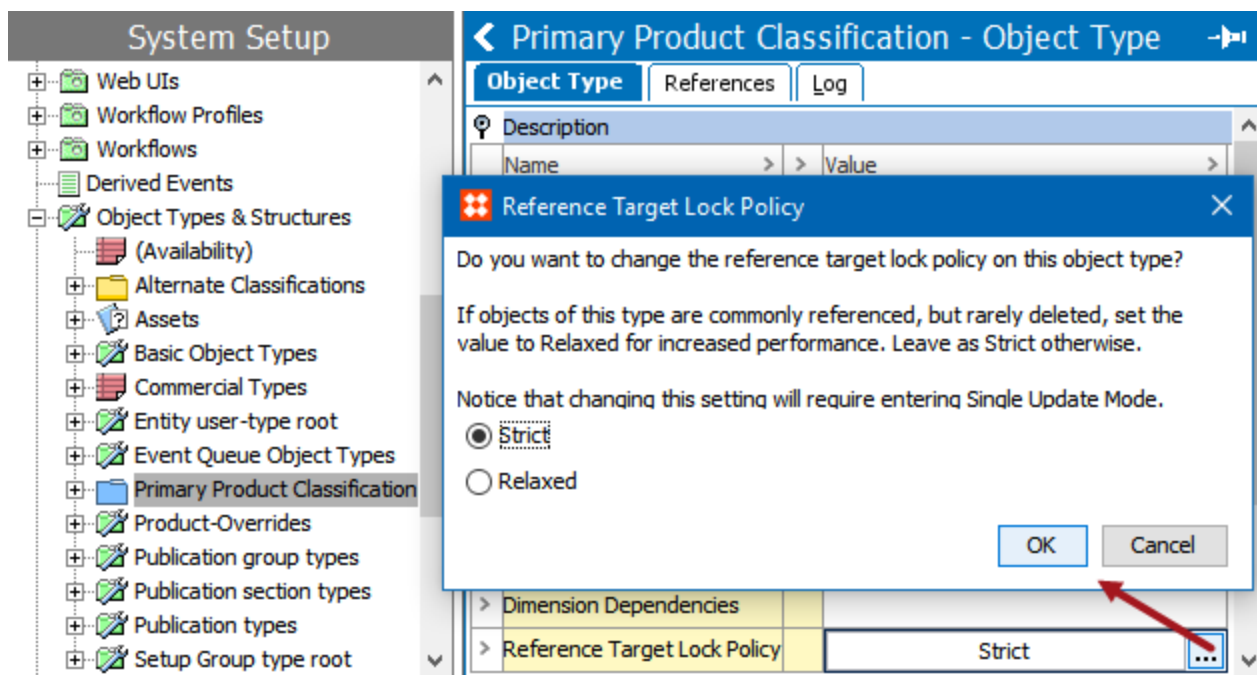
- **Keep classification links in Main Workspace:** the Classification References in the Main Workspace remain untouched. Classification References in all other Workspaces are removed. Since the references in the Approved Workspace are replaced with the references from Main Workspace, the Approved Workspace 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.

Reference Target Lock Policy on Object Types

The 'Reference Target Lock Policy' is used for the Assets, Entities, Classifications, and Products object types to manage how objects should be locked while they are being referenced. When a reference between two objects is being created in STEP, the target object is locked to ensure that it is not being deleted while the reference is being created, meaning only one process or user can edit it. This can be an issue when running parallel imports where the first import locks the object type being referenced and the second import eventually stops running because it cannot access the locked object. The default setting for this property is set to 'Strict' unless changed to 'Relaxed', and 'Relaxed' should NOT be used on an object type for objects that are often deleted – this could result in poor performance, and risk of locks and deadlocks.

When a reference between two objects is being created in STEP, the target object will be 'locked' to ensure that it is not being deleted while the reference is being created. In most of these cases, deletion of the object is not a deletion from the database, but the deletion that occurs due to revision control when an insert of a new history entry occurs. This can be a major issue when running parallel imports where the first import locks the object type being referenced, and the second import eventually stops running because it cannot access the locked object. Although STEP continues to retry the import, this can cause inbound feeds to be negatively affected.

The 'Relaxed' setting on the property puts a less restrictive lock on the reference target objects being edited so that they can be updated concurrently by more than one process and/or user, with a full lock being employed only in the case that a deletion is attempted. This setting, when used with object types that are frequently referenced but rarely deleted, will improve the performance and stability of parallel inbound imports, bulk updates, and users concurrently creating references to the same objects.



These Relaxed / Strict settings can be changed back and forth as needed, but changing the setting requires the system to enter Single Update Mode for a short time, only as long as it takes to update the setting. Keep in mind, however, that changing this setting during peak production hours can have a negative impact on performance.

Revisability on Entity Object Type

Unlike other objects which are always workspace revisable or globally revisable, the entity object type editor includes a Revisability parameter that can be set to Global Revisable or Workspace Revisable.

The screenshot shows the 'System Setup' tree on the left with 'Customer Data Root' selected. The right pane shows the 'Customer Data Root - Object Type' editor. The 'Revisability' field is highlighted with a red box, showing 'Global Revisable'.

Description	
Name	Value
ID	CD_Customer_Root
Name	Customer Data Root
Last edited by	2016-09-01 11:38:11 by USER6
Name Pattern	
ID Pattern	
Enable Profiling	No
Icon	
Dimension Dependencies	
Revisability	Global Revisable
Enforce Reference Target...	Yes

- Global revisable entity values display identically in all workspaces, are not under revision control, and therefore do not require approval.

The screenshot shows the '116 Spadina Avenue rev.0.1 - Address' object type editor. The 'Address' tab is selected, and the 'Revisability' field is highlighted with a red box, showing 'Global Revisable'.

Description	
Name	Value
ID	ADD_184139
Name	116 Spadina Avenue
Object Type	Address
Revision	0.1 Last edited by USER8 on Wed Aug 31 13:48:51 EDT 2016
Path	Entity hierarchy root/Entity Root/Address Root/116 Spadina Avenue

- Workspace revisable entity values display in the workspace where created, are under revision control, and must be approved before they can display in the Approved workspace.

Baseball Hat rev.0.20 - Product		
Product	Sub Products	References
Referenced By	Images & Documents	Comments
Description		
Name	>	Value
ID	>	20805
Name	>	Baseball Hat
Object Type	>	Item
Revision	>	0.20 Last edited by USER8 on Fri Sep 09 15:07:36 EDT 2016
Approved	>	 Last Approved on Fri Oct 16 09:39:04 EDT 2015
Translation	>	Not Translated

The revisability setting also plays a part in determining when events are generated. For more information, see the **Events** section of the **System Setup / STEP Super User Guide** documentation.

Note: When building an entity object structure, a top entity object type can be global revisable, while child entity object types are workspace revisable. However, a top entity object type that is workspace revisable can only have child entity object types that are also workspace revisable.

Product Variants

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. For example, clothing and shoes both offer good use cases for product variants as shoes or clothing items may come in multiple sizes and colors, but are otherwise identical. The product family holds all data for the item, e.g. a particular shoe, while the variants would hold only the various combinations of the color and size varieties (e.g. size 7 red, size 8 blue, etc). For example:

Variant	Shoe Size (Europe)	Shoe Size (US)	Color
> 115121	36	6	Black
> 115122	36	6	Navy Blue
> 115123	36	6	Multi-color
> 20695	36	6	Red
> 20714	37	6.5	Red

Color selection dropdown options: Black, Gray, Multi-color, Navy Blue, Orange, Red, Royal Blue.

The Product Variants tab is not available by default as there is configuration required to determine where variants should be enabled and what attributes should be considered variant attributes for each category of objects.

To configure a system for product variants, see the **Setting Up Product Variants** topic in this guide.

For working with variants, see the following topics:

- Using Variants List View
- Using Variants Matrix View
- Creating Variants Through Web UI

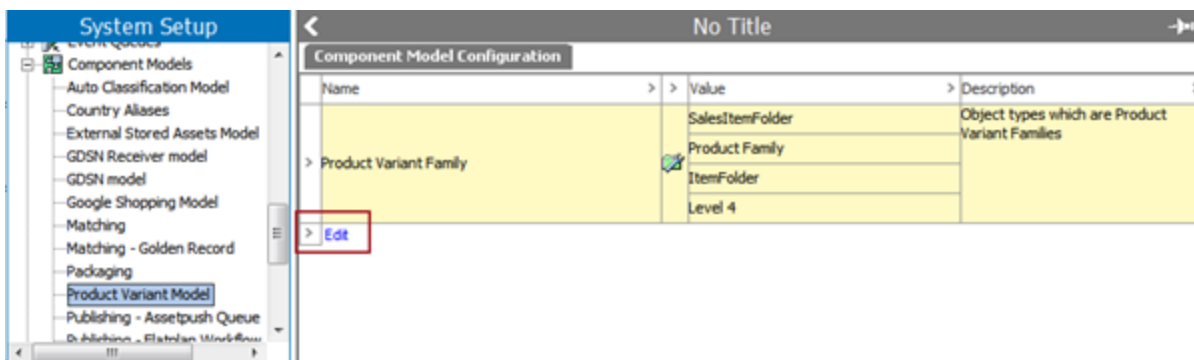
Setting Up Product Variants

To set up product variants, complete the following steps:

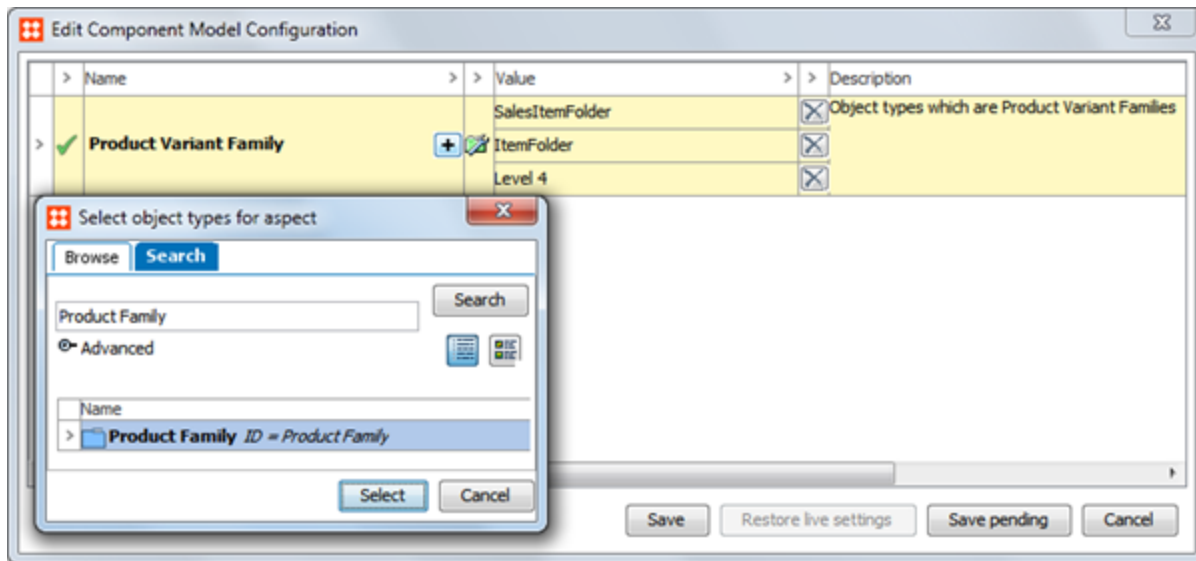
1. Add the object type of the parent folder to the component model **Product Variant Model** to enable the product variant capabilities.
2. Create a meta-data attribute and make it valid on the links from the product folder to the related attributes. This enables you to specify which attributes are variant 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**.

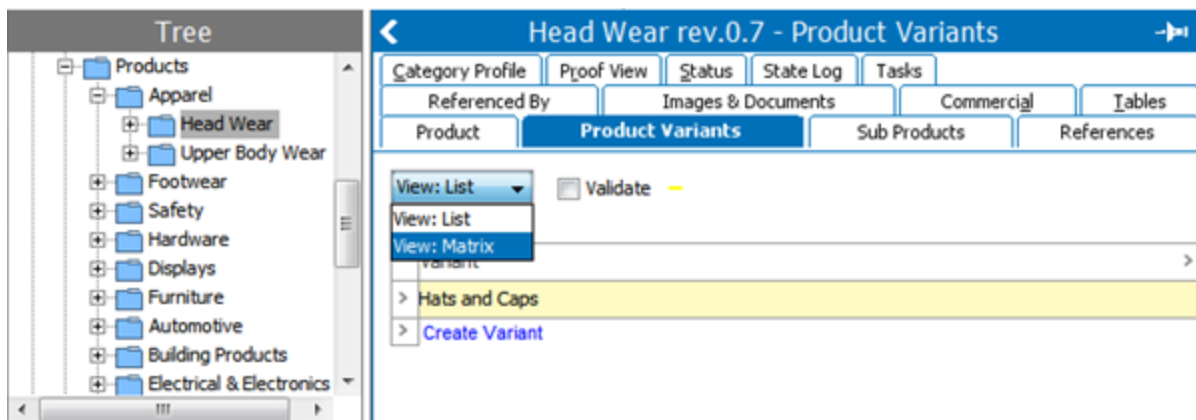


3. In the **Edit Component Model Configuration** dialog, click the plus sign, choose the object type of the parent folder, and then click **Save**.



After having completed this step, a **Product Variants** tab is available on all products of the selected object types.

- To view the **Product Variants** tab, in the **Tree** in the **Primary Product Hierarchy** select an object. The tab has two views for managing product variants: **List** view and **Matrix** view.



Note that no attributes are available for editing at this point. To complete the setup, a metadata attribute has to be configured and the priority of variant attributes must be assigned.

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.

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.

We recommend 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 Product Variant Priority, and it is validated using the List Of Values ProductVariantPriority

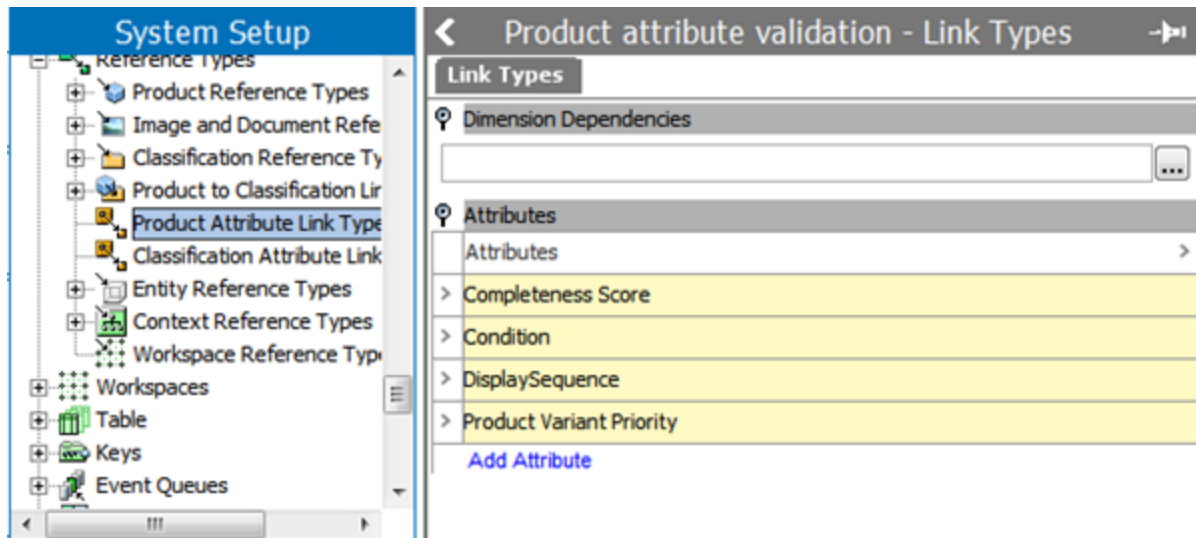
System Setup		Product Variant Priority - Attribute	
		Attribute	References
Pressure Relief Valve		Attribute Transformation	
Primary Color		Validity	
Product Variant Priority		Profile	
Programmable		Log	
Rated Strength (lb.)		State Log	
Seasonal		Tasks	
Shoe Size (Europe)			
Shoe Size (US)			
Stranded or Solid			
Temper			
T-shirt Size			
UL Listed			
URL			
Voltage			
Wattage			

Attribute Validation	
Name	Value
> Validation Base Type	List Of Values
> List Of Values	ProductVariantPriority
> Multi Valued	No
> Mask	
> Minimum Value	N/A
> Maximum Value	N/A
> Maximum Length	N/A

[Edit Validation Rule](#)

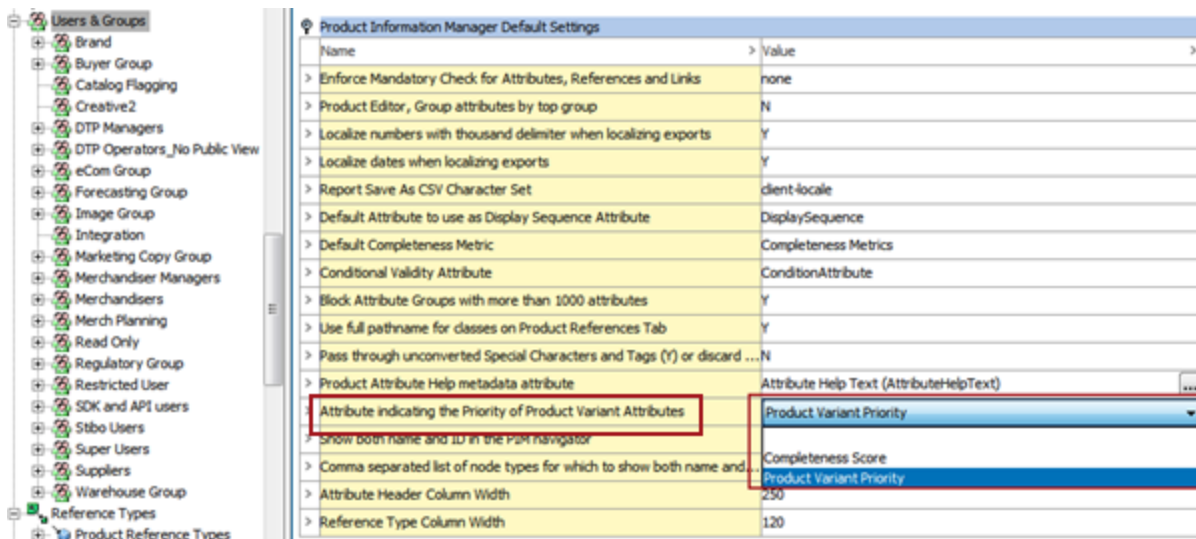
Next, make the attribute valid on product to attribute links.

3. In **System Setup**, expand **Reference Types**, select **Product Attribute Link Type**, and then add the attribute you just created.



Now, 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, Footwear is the parent folder and the variant attributes are Color, Shoe Size (Europe), and Shoe Size (US).

The screenshot displays the 'Footwear rev.0.3 - References' window. On the left, a 'Tree' pane shows the 'Primary Product Hierarchy' with 'Footwear' as the parent folder. The right pane shows a table of linked attributes from the product hierarchy. The table has columns for 'DisplaySequence', 'Completeness Score', 'ID', 'Name', 'Condition', 'Latitude', and 'Product Variant Priority'. The attributes listed include 'AirTransportationRestricti...', 'AnnualSalesForecastMaxi...', 'AnnualSalesForecast, Mini...', 'AvailabilityEnd', 'AvailabilityStart', 'BaseUnitOfMeasure', 'BrandName', 'BrandOwner', 'Color', 'ShoeSizeEurope', 'ShoeSizeUS', 'ShortItemDescription', 'Stackable', 'StackedHeight', 'Status', 'SupplierName', 'SupplierPartNumber', 'Table Sort Order', and 'TaxClassification'.

DisplaySequence	Completeness Score	ID	Name	Condition	Latitude	Product Variant Priority
✓			AirTransportationRestricti...	Air Transportation Restriction		
✓			AnnualSalesForecastMaxi...	Annual Sales Forecast, Maxim		
✓			AnnualSalesForecast, Mini...	Annual Sales Forecast, Minim		
✓	5		AvailabilityEnd	Availability End		
✓	2		AvailabilityStart	Availability Start		
✓	1		BaseUnitOfMeasure	Base Unit of Measure		
✓	5		BrandName	Brand Name		
✓	6		BrandOwner	Brand Owner		
>			Color	Color		3
>			ShoeSizeEurope	Shoe Size (Europe)		1
>			ShoeSizeUS	Shoe Size (US)		2
✓	2		ShortItemDescription	Short Item Description		
✓	6		Stackable	Stackable?		1
✓	7		StackedHeight	Stacked Height		2
✓			Status	Status		3
✓	1		SupplierName	Supplier Name		4
✓	2		SupplierPartNumber	Supplier Part Number		5
✓			Table Sort Order	Table Sort Order		6
✓	3		TaxClassification	Tax Classification		7

Repeat the above steps as appropriate for other product categories. Note that variant priority should be set as high in the hierarchy as possible to reduce the setup and maintenance effort. For example, in the above, the variant priority is set on the Footwear node. And whether we look at objects in the Shoes or Boots subcategory, the variant editor will be set up to have Color, Shoe Size (Eurpoe), and Shoe Size (US) as the variant attributes.

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.

Tree

Primary Product Hierarchy

Products

Apparel

Head Wear

Hats and Caps

Hats and Caps Items

(111777)

(111778)

(120170)

(120666)

109308

20803

20805

Gray Hat

Hats and Caps Items rev.0.4 - Product Variants

Images & DocumentsCommercialTablesCategory ProfileProof ViewStatusState LogTasks

ProductProduct VariantsSub ProductsReferencesReferenced By

View: List☒ Validate☐

Variant	Hat Size	Color
> (111777)		
> (111778)		Gray
> (120170)		
> (120666)		
> 109308	7	Gray
> 20803	7	Black
> 20805		Black
> Gray Hat		
> Create Variant		

ID = 114509

Products have identical values for variant attributes

Duplicated by: (120666), (111777), (120170)

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**.

Variant Info	Black	Gray
7	20803 20805	109308
8	(111777) (120170) (120666) Gray Hat	

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.

Create variants

ObjectType

Shoe

Size (7)

120

105

130

133

134

107

106

104

121

108

131

Gender (1)

Male

Female

Color (2)

Black

Green

Pink

Purple

Red

White

Orange

Blue

Number of combinations: 14

OK

Cancel

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.

Preview			
	Gender	Size	Color
☒	Female	107	Orange
☒	Female	106	Orange
☒	Female	104	Orange
☒	Female	121	Orange
☒	Female	108	Orange
☒	Female	131	Orange
☒	Female	132	Orange
☒	Female	107	Blue
☒	Female	106	Blue
☒	Female	104	Blue

1-10 of 14

OK Cancel

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.

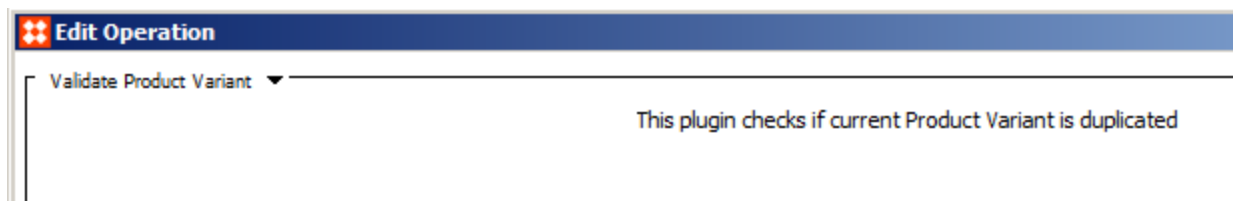
Business Condition: Validating Product Variant

It is possible to use a business condition to perform validation for product variant duplicates.

For more information about business conditions, see Business 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**.

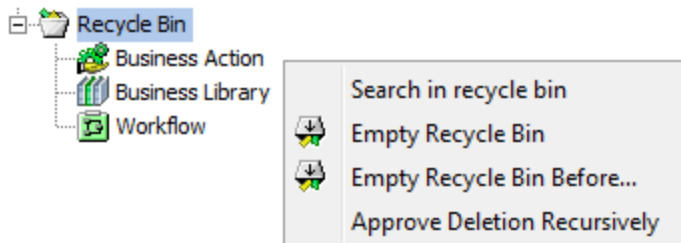


The Validate Product Variant condition uses the configuration you created when you set up Product Variant Families, so there is no configuration to do. The action works on the Product Variant level, and returns false if the product variant is a duplicate.

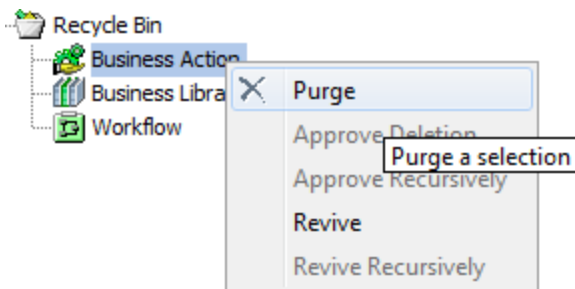
Note: Running the Validate Product Variant business rule may affect performance depending on how many siblings and variant attributes that have to be checked.

Recycle Bin

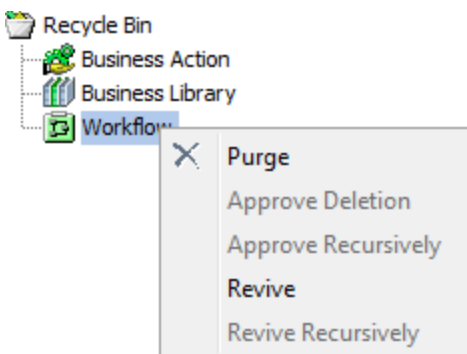
System Setup contains a recycle bin.



Deleted business rules can be seen in the Recycle Bin and can be revived from that location. Similarly they can be purged.

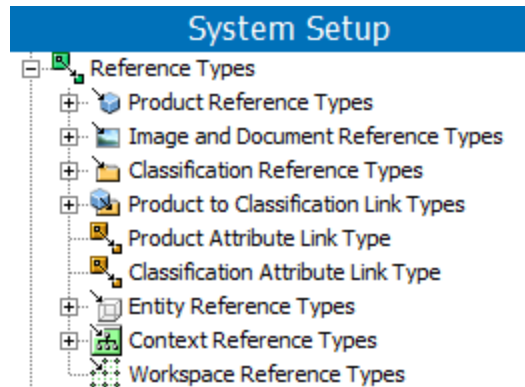


Deleted workflows can be seen in the System Setup Recycle Bin and can be revived from this location. Similarly they can be purged from the bin.



Reference and Link Types

This section describes how to set up and work with 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 define relationships between objects. 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

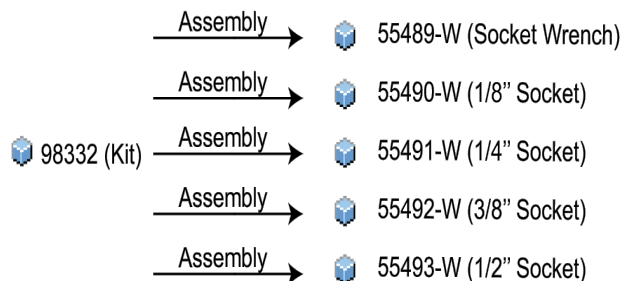
Direction of a Reference

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.

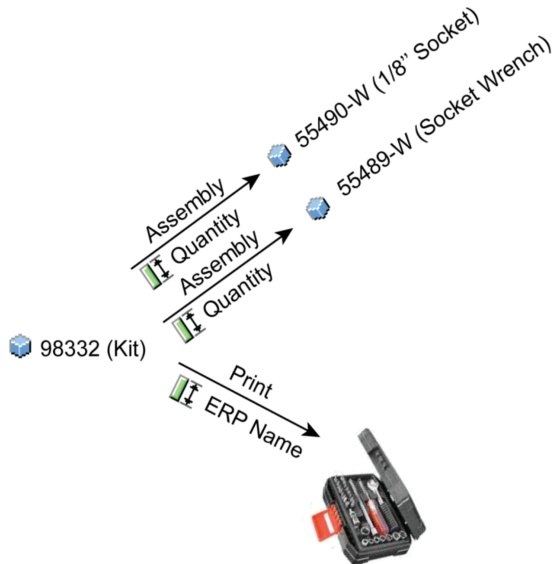
Icons used for References	Description
	Entity, Product, Image, Classification Reference Types
	Inherited Reference Type
	Mandatory Reference Type
	Inherited and Mandatory Reference Type

Note: The approval check for mandatory product to classification links is determined by the ownership of the link. The ownership of a link is set up on the classification object type using the 'Owns Product Links' parameter.

Metadata Attributes

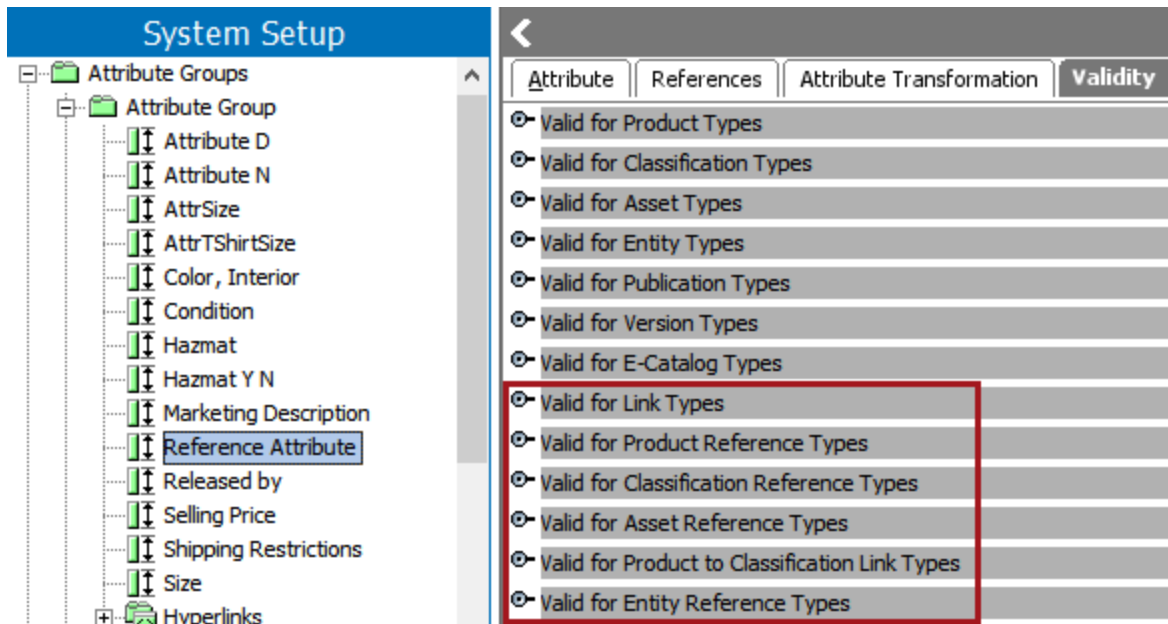
Description Attributes can be applied to a **Reference Type**.

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.

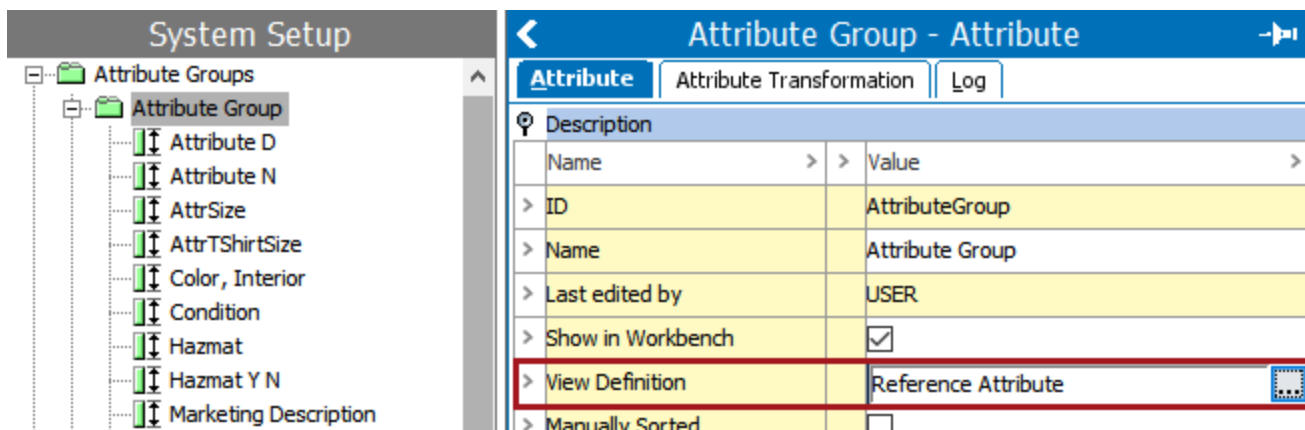


- A **Description Attribute** named **Quantity** has been applied the Reference Type **Assembly**. This Attribute holds number of Sockets to be included in the KIT.
- A **Description Attribute** named **ERP Name** has been applied the Reference Type **Print**. This Attribute holds the name of the Reference used in an ERP System.

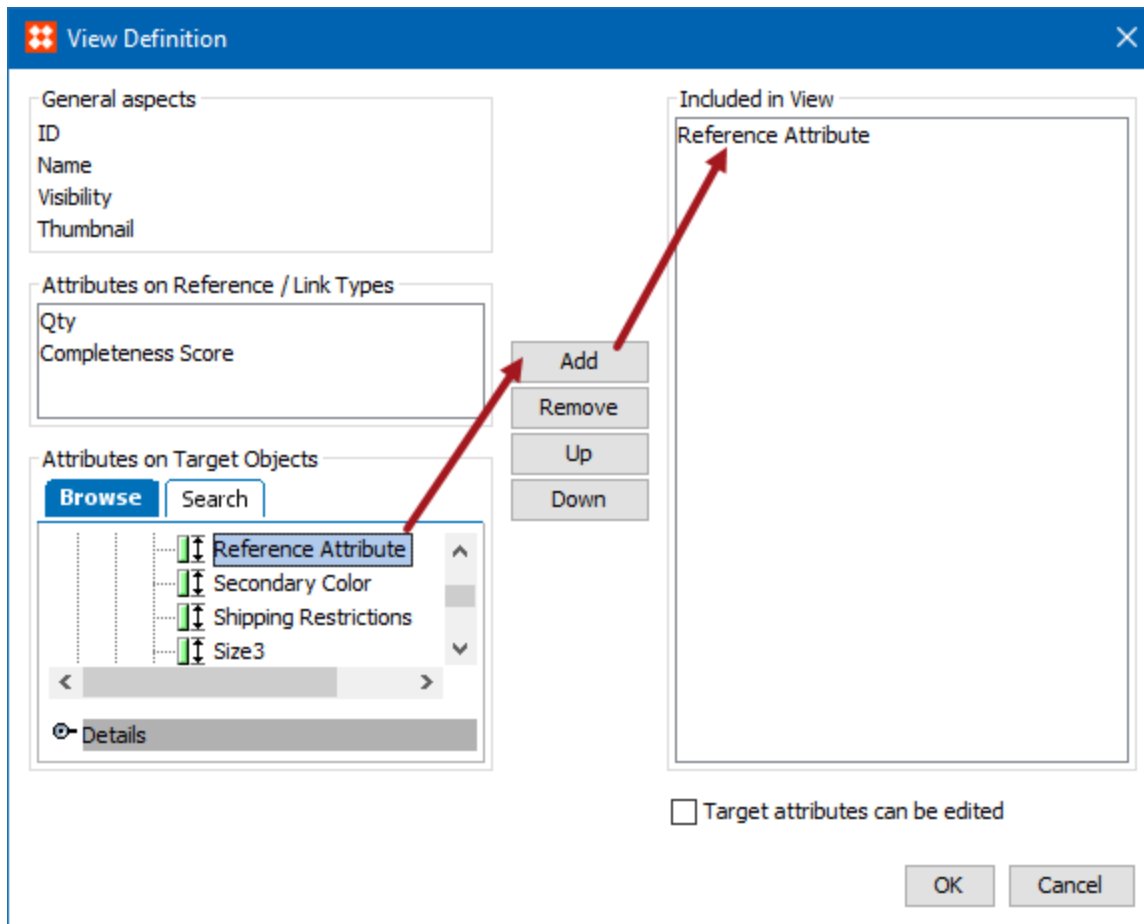
Metadata Attributes are applied to a Reference Type from the Validity tab in the Attribute Editor, by selecting one or more reference types from the reference flippers:



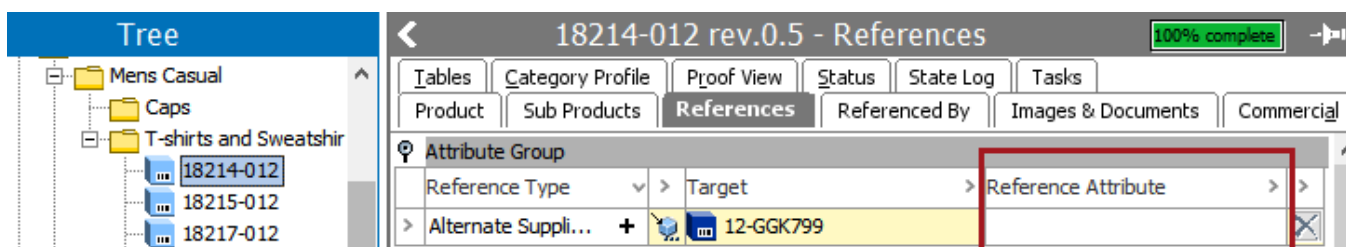
After an attribute has been valid for the reference type, it must also be added to the view for the attribute group for it to appear in the object editors. On the **Attribute** tab of the attribute, click in the **View Definition** field to expose an ellipsis (...).



Click the ellipsis to open the View Definition dialog. Search or browse for the attribute, select it, and click Add to add it to the view.



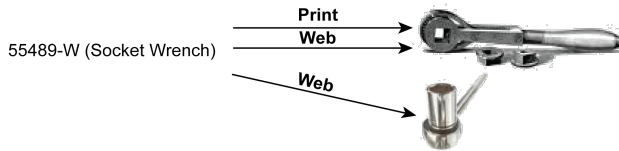
The attribute will then be available on the reference.



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 be 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.

Compare Contexts		References	Referenced By
Data Group			
Shipping Info			
Technical Sheets			
Reference Type: Data Sheets			
Context	Reference Type	Target	Visibility
DE All All	Data Sheets	X9-123_SH	All Languages
EN All All	Data Sheets	X9-123_SH	Suppressed
FR All All	Data Sheets	X9-123_SH	All Languages
Global	Data Sheets	X9-123_SH	All Languages
SP All All	Data Sheets	X9-123_SH	All Languages
TCR/ReferenceGroup			
Un grouped Classification References			
Un grouped Product References			
Index Words			
Un grouped Image & Document References			
Linked Attributes			

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:

System Setup

Inherited, not valid on child

Inherited, valid on child

LastParent

Multi-supplier Item

Open Item

Pack To Child

Pallet To Child

PrimaryDataSource

PrimaryFamily

PrimarySupplierItem

Inherited, not valid on child - Validity

Reference Type

Validity

Log

Valid Source Types

ID

Name

ItemFolder

Item Folder

Modify Source Types

Valid Target Types

ID

Name

Item

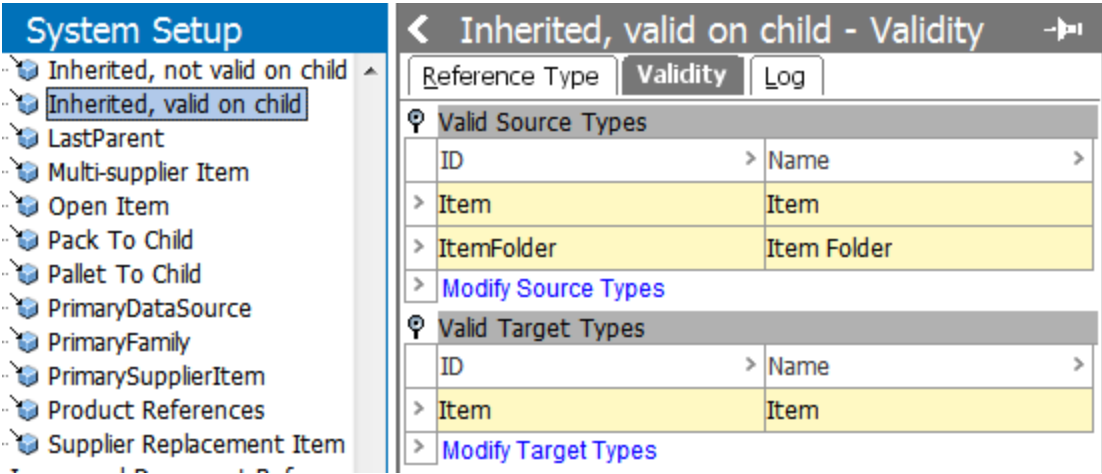
Item

Modify Target Types

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:

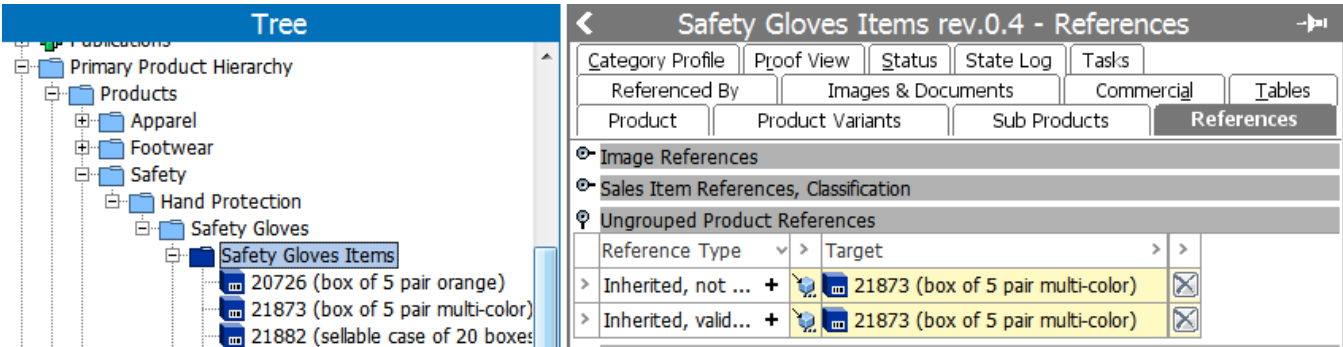
© Stibo Systems - Confidential - STEP Trailblazer 8.1

467

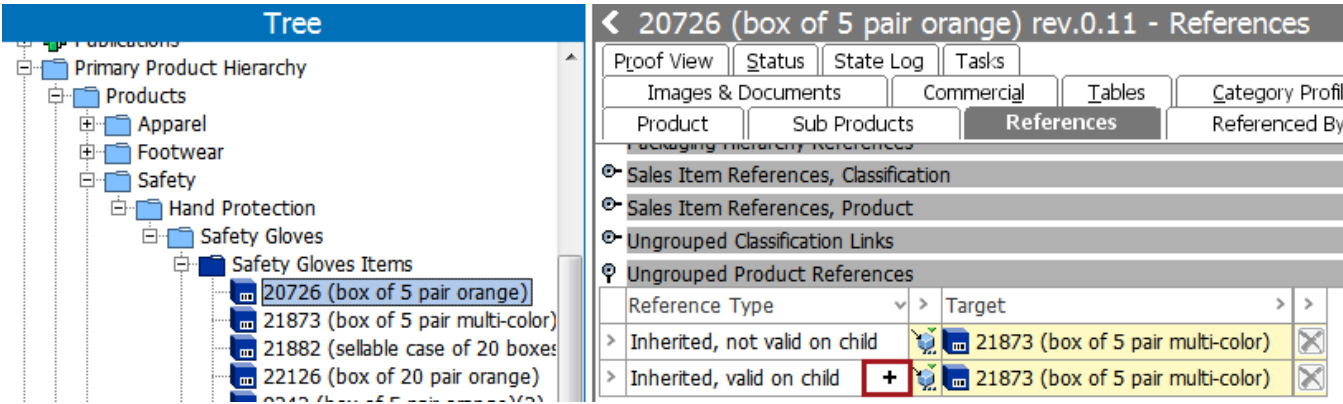


When both references are set with some type of inheritance, any reference added to the parent (Item Folder object) will also be present on the child (Item object).

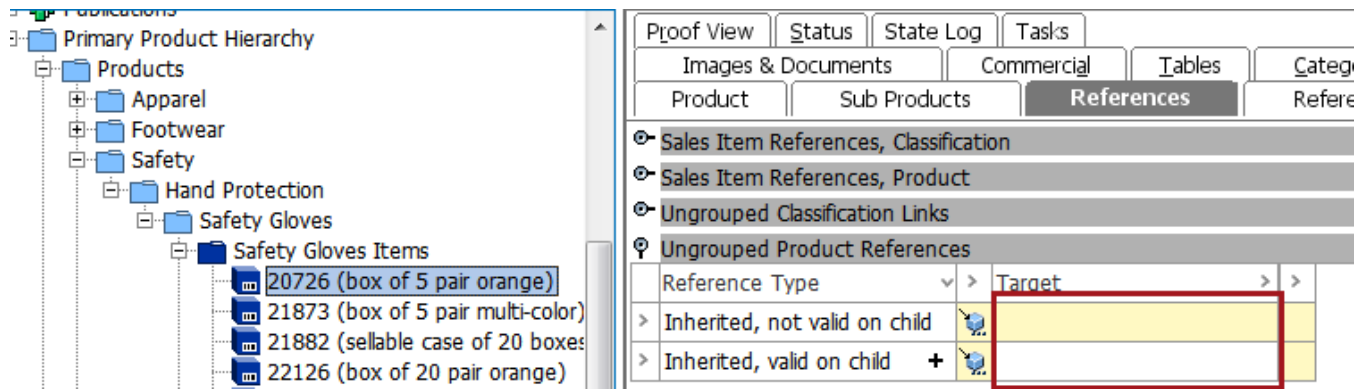
Parent object:



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.

Reference Type	Validity	Log
Description		
Name	>	Value
ID		SendTo
Name		Send to
Last edited by		2017-03-22 08:00:57.437 by USER
Externally Maintained		No
Dimension Dependencies		
Completeness Score		
Allow multiple references		Yes
Mandatory		No
Parent/Child relation		None
Inheritance		None

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 super type (e.g., products or classifications), the parameter will not be present.

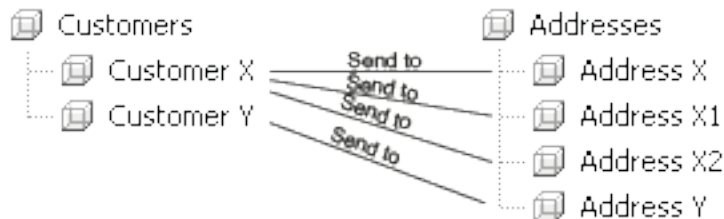
Both the parent and the child objects must be of the Entity object type. On the Entity Reference Types Editor, the Parent/Child Relation field includes these options:

- **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).

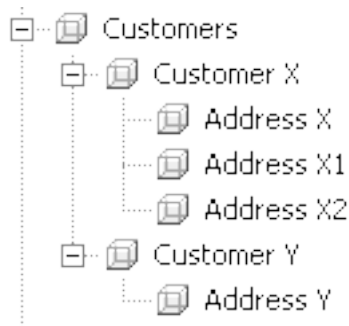
The following example includes:

- 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:

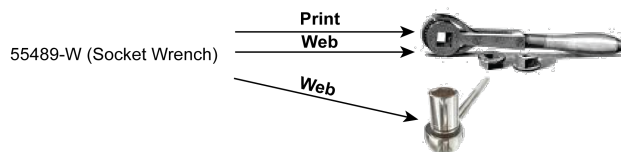
- Different icons for each Entity make the hierarchy overview easier to understand. See the **Object Type Icons** section of the **System Setup / Super User Guide**.
- Aggregated references enables **Copy**, **Paste**, **Drag** and **Drop** between Entity hierarchies. STEP will automatically apply the 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**.

Reference Type		Validity	Log
Description			
Name	>	>	Value
ID			Web
Name			Web
Last edited by			2017-02-23 16:00:31.107 by USERJ
Externally Maintained			No
Dimension Dependencies			
Allow multiple references			Yes
Mandatory			No
Inheritance			None
Completeness Score			10
Purpose			Images that should be used for the web.
In Attribute Groups			
ID	>	Name	>
WebReferenceTypes			Web Reference Types
Add Attribute Group			
Valid Attributes			
ID	>	Name	>
Add Attribute			

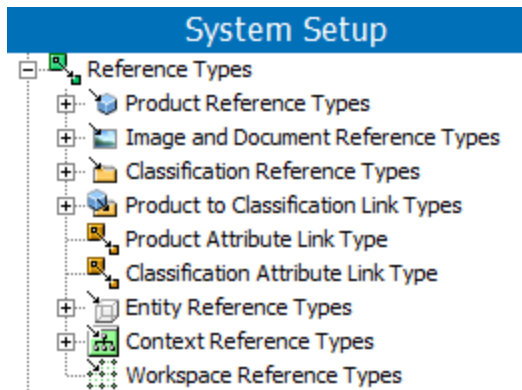
A User Privilege is setup to include the **Approve Reference** Privilege for Reference Types in the Attribute Group **Web Reference Types**. For more information about setting up privileges, see **About Action Sets** in the **System Setup / Super User Guide**.

User Group2 - Privilege Rules					
Group	Privilege Rules	GUI Set-Up	Log		
Setup Privileges					
User Privileges					
Applies to	>	Action Set	>	Attribute Group	>
Primary Product Hierarchy		Approve Reference		Web Reference Types	
Add Privilege					
Read Only					

A User with this privilege setup will only be allowed to Approve References of Reference Types Web.

Creating Reference Types

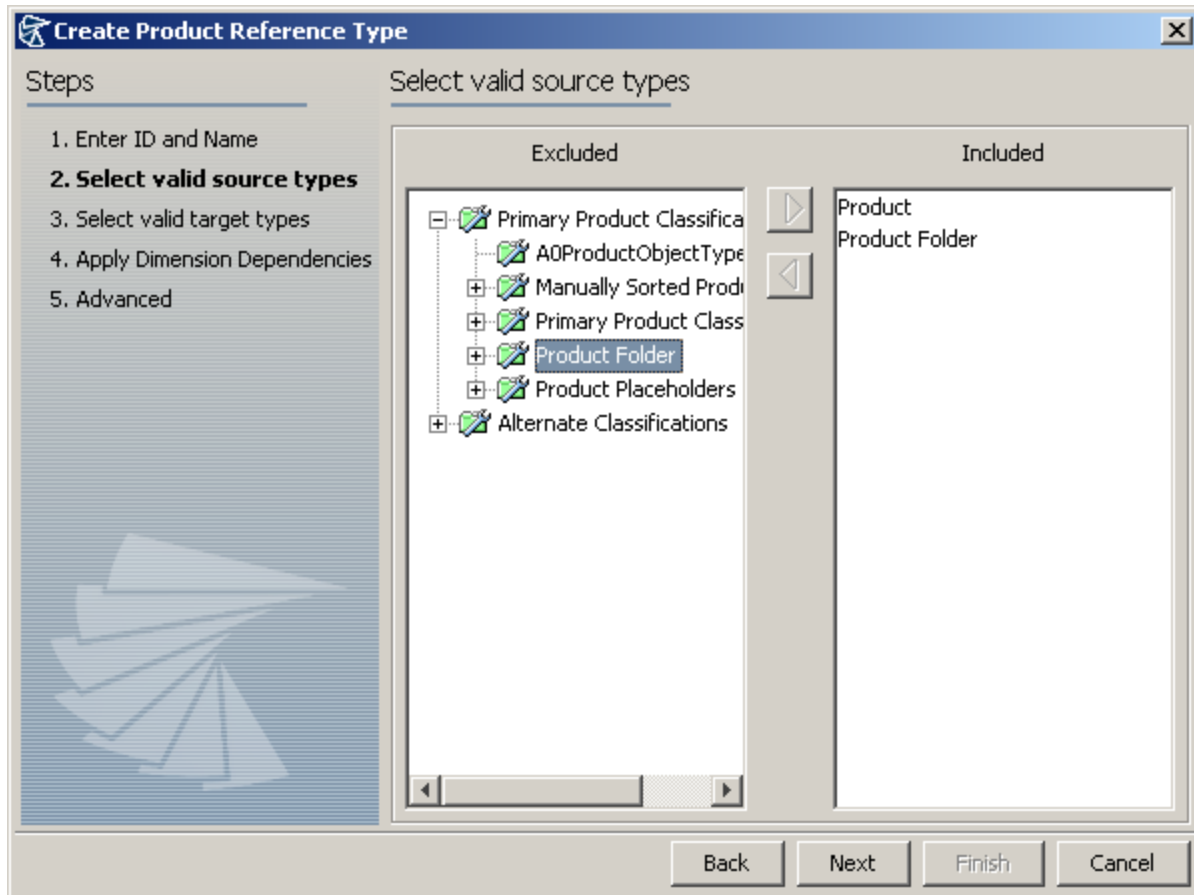
1. In **System Setup**, click the **Reference Types** Hierarchy.



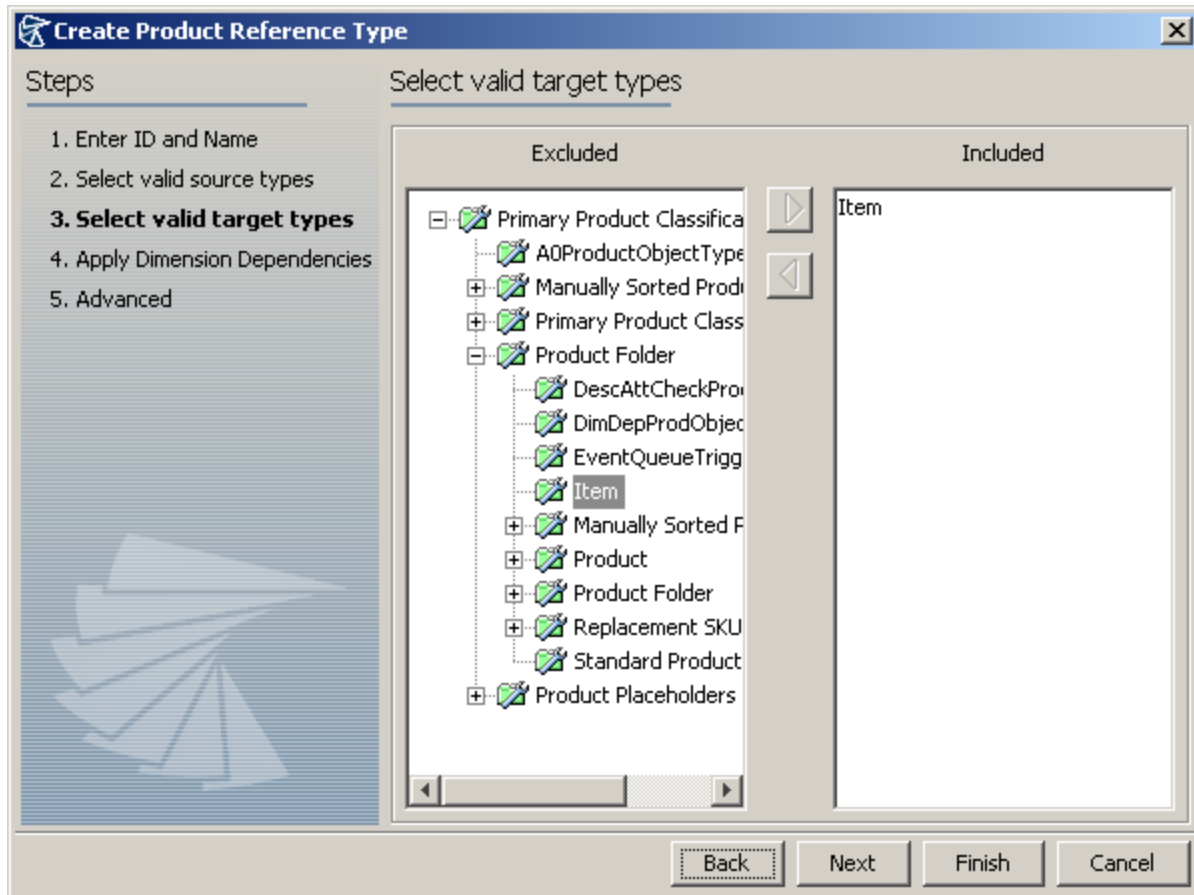
2. Right-Click relevant **Reference Type** to be created, click **New Reference Type**.

The screenshot shows a dialog box titled "Create Product Reference Type". It has a "Steps" pane on the left and a main area on the right. The "Steps" pane lists five steps: 1. Enter ID and Name, 2. Select valid source types, 3. Select valid target types, 4. Apply Dimension Dependencies, and 5. Advanced. The first step is selected. The main area is titled "Enter ID and Name" and contains two input fields: "ID" and "Name". At the bottom of the dialog are four buttons: "Back", "Next", "Finish", and "Cancel".

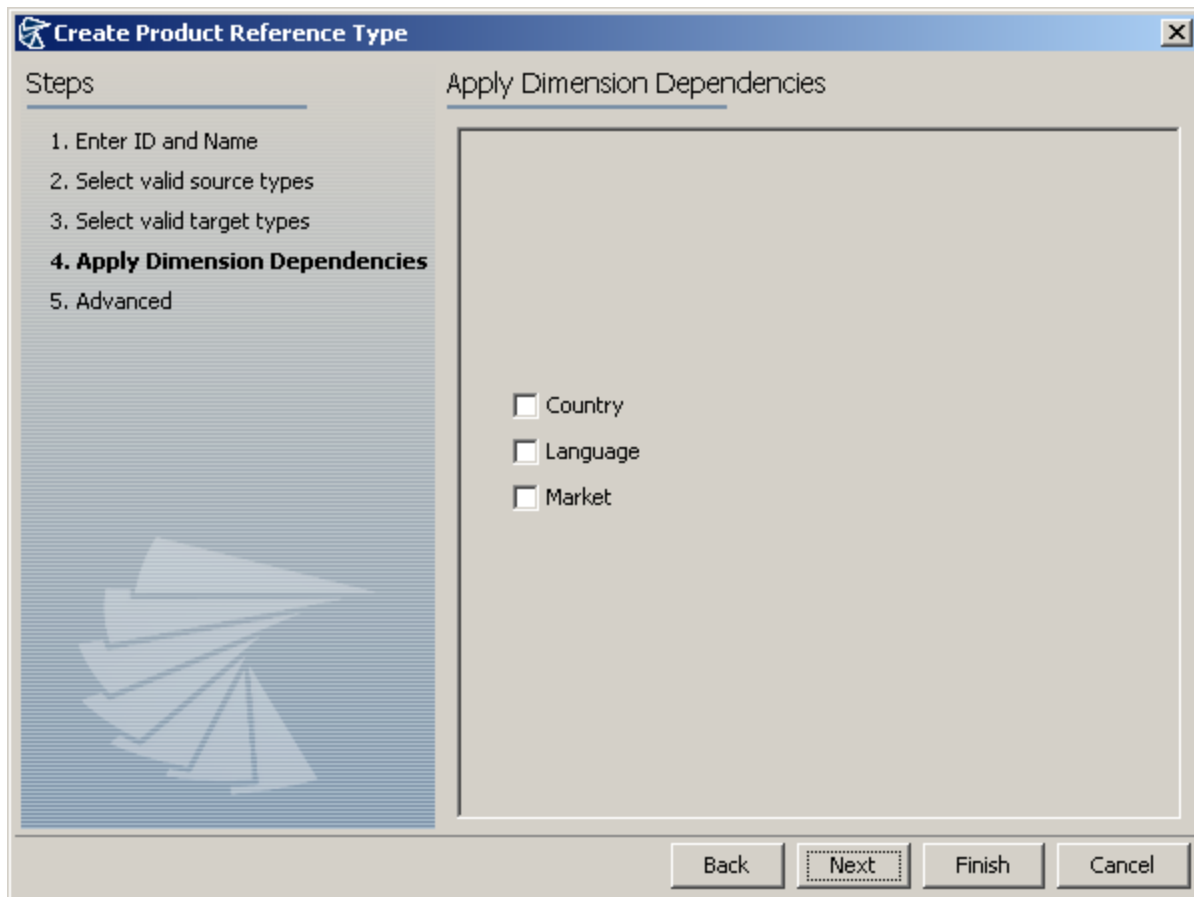
3. Key in **ID** and **Name**, and click **Next**.
4. Select **Valid Source Object Types** for the Reference Type to be created, and click **Next**.



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.

Create Product Reference Type

Steps

1. Enter ID and Name
2. Select valid source types
3. Select valid target types
4. Apply Dimension Dependencies
- 5. Advanced**

Advanced

☒ Allow multiple references

☐ Externally Maintained

☐ Mandatory

☐ Inheritance

Back Next **Finish** Cancel

- **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.

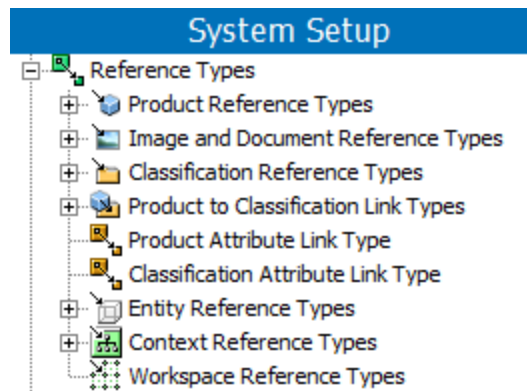
• Inheritance

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**

Accessory - Reference Type	
Reference Type	Validity Log
Description	
Name	Value
ID	Accessory
Name	Accessory
Last Edited by	2010-05-05 11:41:09.0 by STEPSYS
Externally Maintained	No
Dimension Dependencies	
Allow multiple references	Yes
Mandatory	No
Inheritance	No
In Attribute Groups	
ID	Name
Add Attribute Group	

- 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.

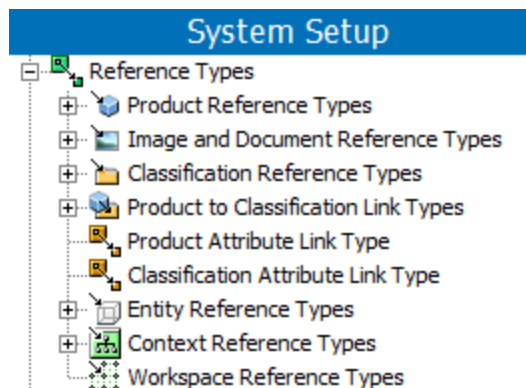
- Optionally, set the **Dimension Dependency** field to a **Dimension**, if the specific Reference Type should be dimension Dependent. It is recommended to select only one Dimension.
- Optionally, set the **Allow multiple references** field to **Yes**, if the specific Reference Type should be used to 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.
- 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.

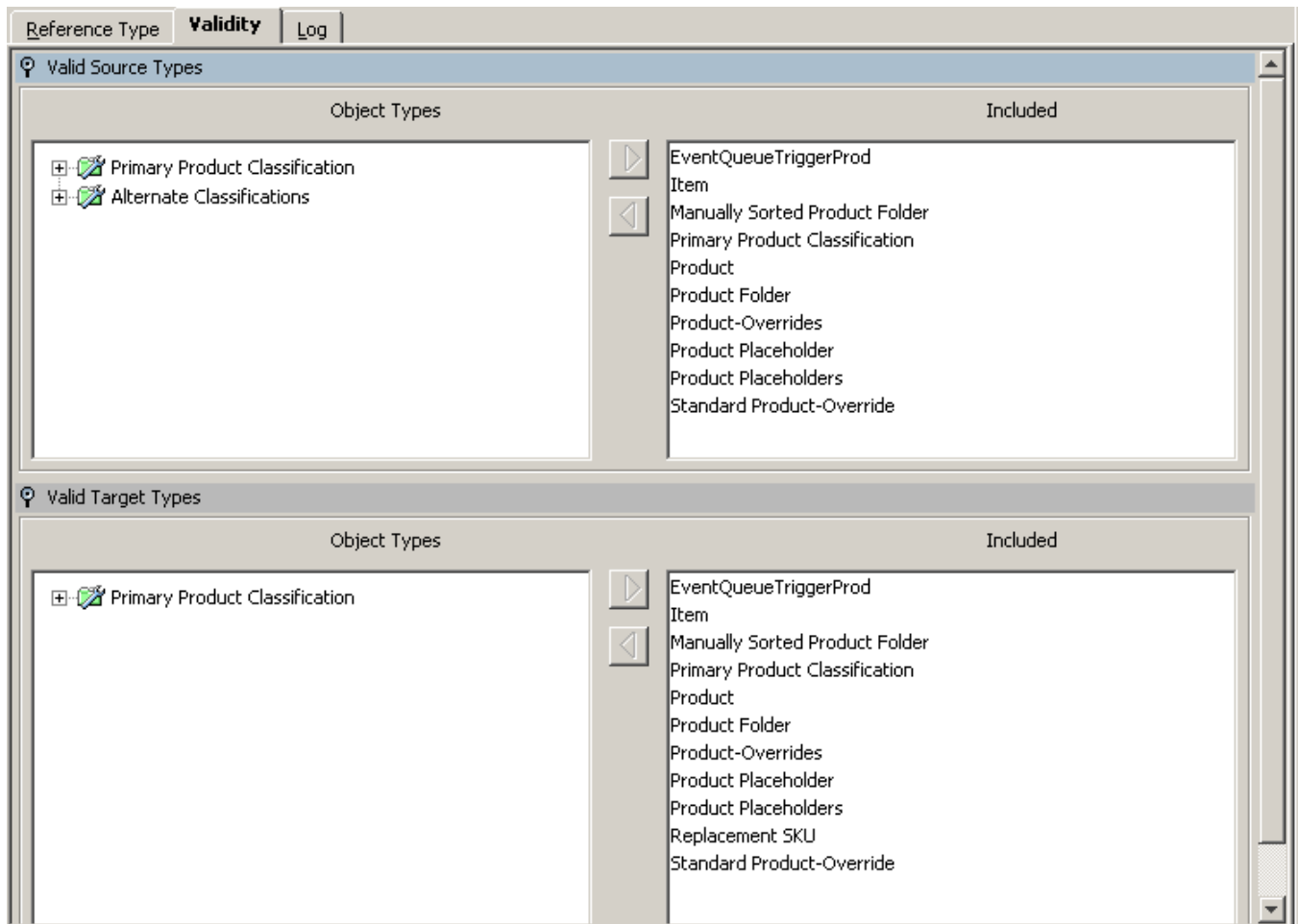
- Optionally, set the **Inheritance** field to **Yes**, if the specific Reference Type should be inheritable. When creating a Reference from a Source (Master Object) to a Target (Subordinate), the Reference will be inherited on all Objects below the Source Object. Each inherited Reference will reference 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

- In **System Setup**, click the **Reference Types** Hierarchy.



- Select the relevant **Reference Type**
- Select the **Validity** tab



4. In the **Valid Source Types** field, use the Add and Remove buttons to specify valid **Source Object Types**.

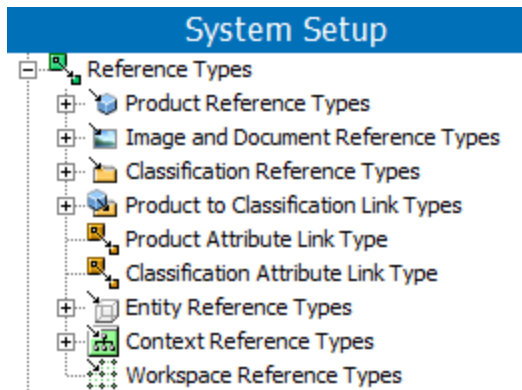
Important: If an Object Type is removed from the Included Object Types, and References already exist in the database, References will be deleted.

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**.

Accessory - Reference Type

Reference Type | Validity | Log

Description

Name	Value
ID	Accessory
Name	Accessory
Last Edited by	2010-05-05 11:41:09.0 by STEPSYS
Externally Maintained	No
Dimension Dependencies	
Allow multiple references	Yes
Mandatory	No
Inheritance	No

In Attribute Groups

ID	Name
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.

Product	Sub Products	References	Referenced By
Item References, Product			
Reference Type		Source	
> Bill of Materials	+		
> Supplier Replace...	+		

2. Find the reference that is to be added and click Select.

Product

Sub Products

References

Referenced By

Item References, Product

Reference Type		Source
> Bill of Materials	+	
> Supplier Replace...	+	

Select Object (Reference Type = Bill of Mat...

Browse

Search

Primary Product Hierarchy

Products

Apparel

Footwear

Shoes

Pumps

Pumps Items

Heels - High: 3" or higher

20714-A

20714-B

20714-C

20714-D

20714-E

20714-F

Details

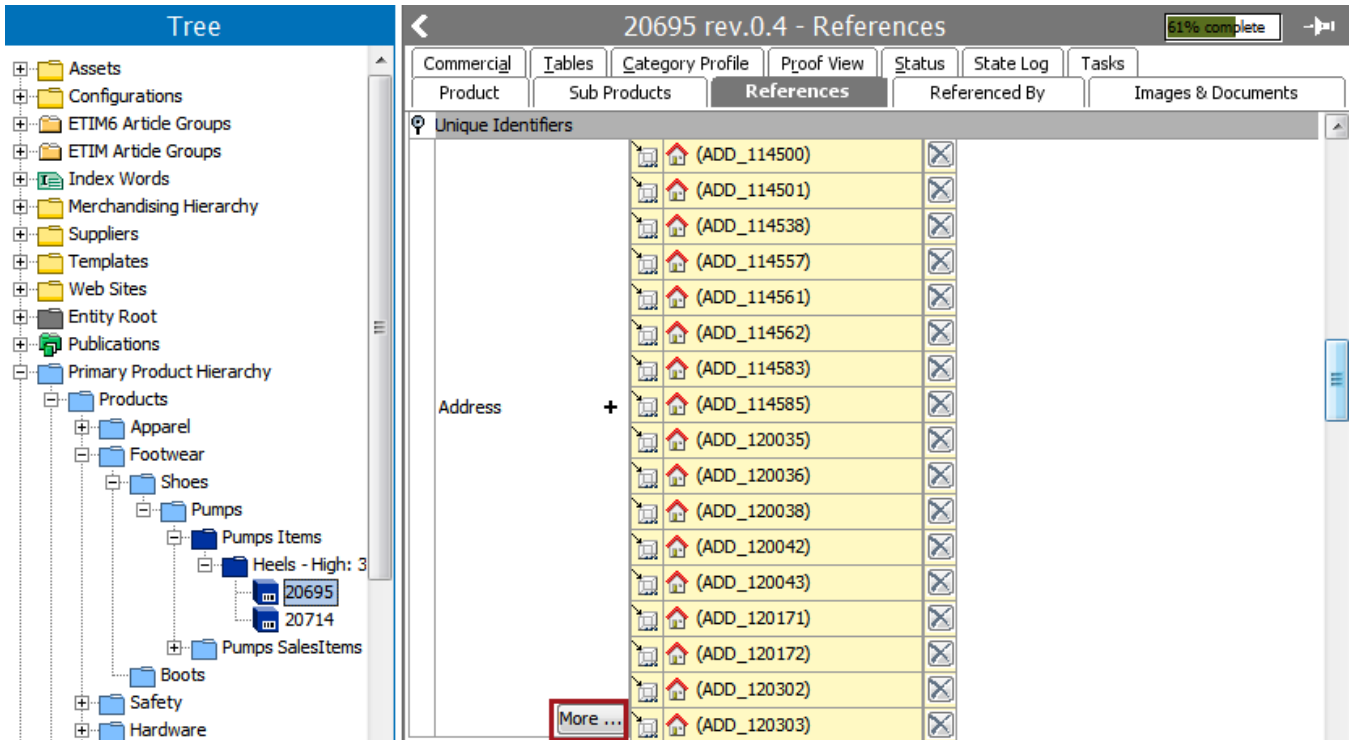
Select

Cancel

3. Once selected, the object will appear and be added as a reference.

Product	Sub Products	References	Referenced By	Im
Attribute Group				
Item References, Product				
Reference Type		Source		
> Bill of Materials	+	20714-B		
> Supplier Replace...	+			

4. If more than 50 references are added, a button entitled '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.

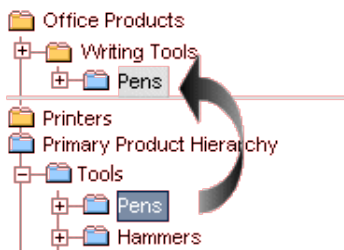
About Product to Classification Link Types

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

Icons used for Link Types	Description
	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.

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](#).

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](#).

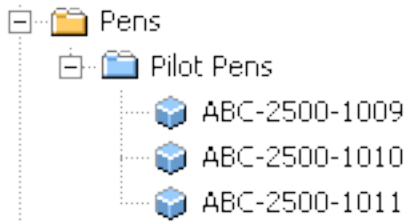
Inheritance of Links, Specification Attributes and Tables

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](#).

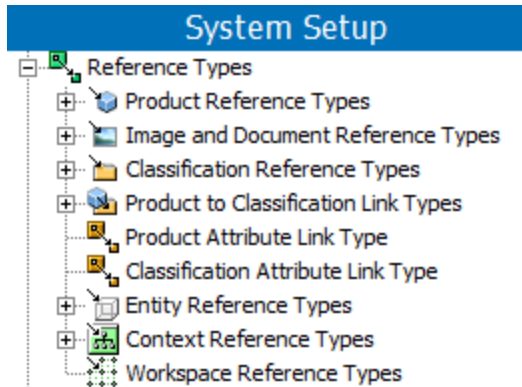
Product Classification Link Types in Attribute Groups

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

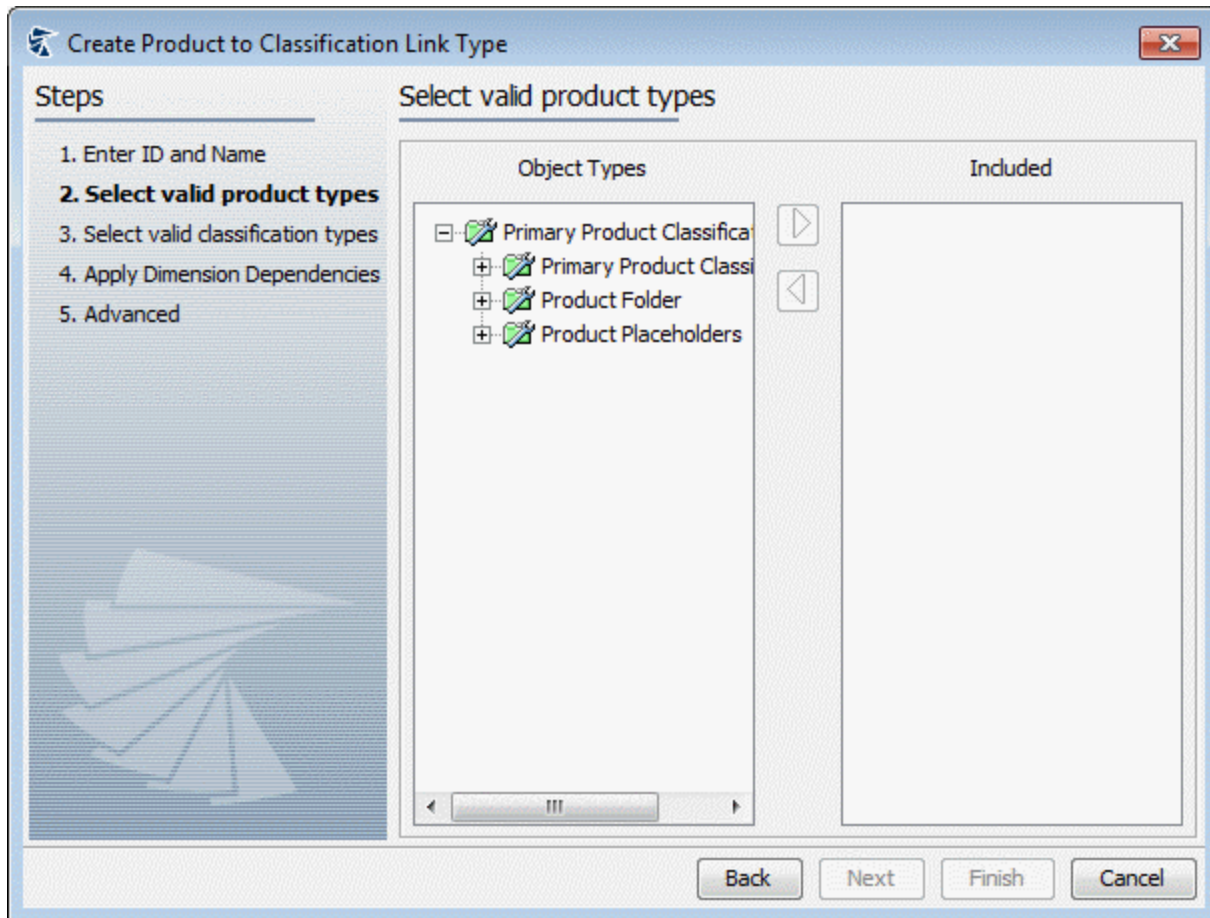
1. In **System Setup**, click the **Reference Types** Hierarchy.



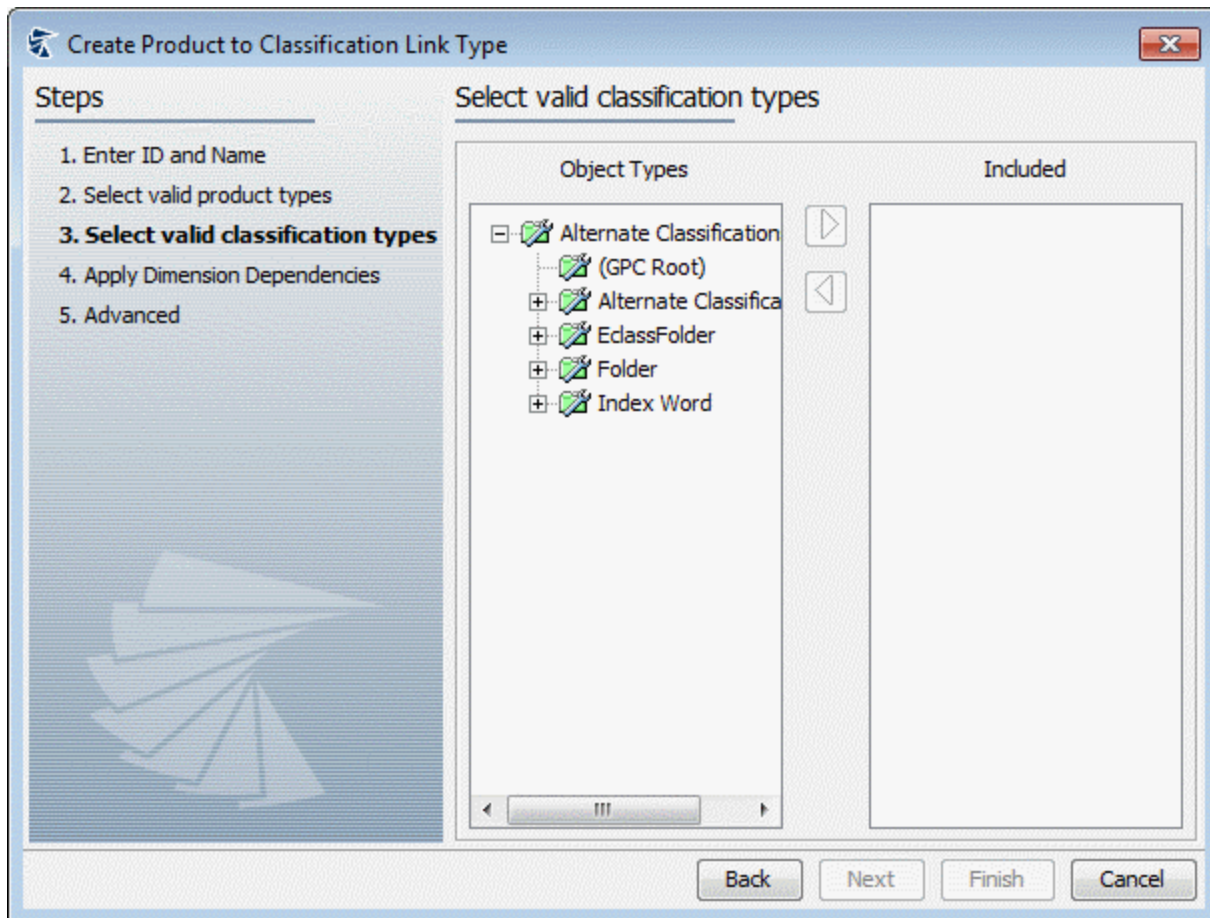
2. Right-Click **Product to Classification Link Types** and click **New Product to Classification Link Type**.

The screenshot shows the 'Create Product to Classification Link Type' dialog box. It has a title bar with a close button. The dialog is divided into two main sections: 'Steps' on the left and 'Enter ID and Name' on the right. The 'Steps' section lists five steps: 1. Enter ID and Name (highlighted), 2. Select valid product types, 3. Select valid classification types, 4. Apply Dimension Dependencies, and 5. Advanced. The 'Enter ID and Name' section contains two input fields: 'ID' and 'Name'. At the bottom of the dialog are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

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**.

Create Product to Classification Link Type

Steps

1. Enter ID and Name
2. Select valid product types
3. Select valid classification types
- 4. Apply Dimension Dependencies**
5. Advanced

Apply Dimension Dependencies

☐ Country

☐ Language

☐ Market

Back Next Finish Cancel

Note: It is recommended that only one Dimension is applied for a Link Type.

7. Apply **Advanced Settings** for the Link Type.

Create Product to Classification Link Type

Steps

1. Enter ID and Name
2. Select valid product types
3. Select valid classification types
4. Apply Dimension Dependencies
- 5. Advanced**

Advanced

☒ Allow multiple links

☐ Externally Maintained

☐ Mandatory

Inheritance

- No Inheritance
- No Inheritance of Tables and Specification Attributes
- Inherit Links and Tables and Specification Attributes

Buttons: Back, Next, Finish, Cancel

- **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](#).

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

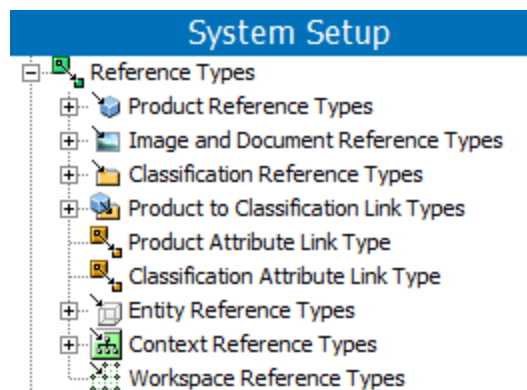
No Inheritance: Link from Product Family to Classification will not be inherited to Leaf Nodes below linked Product Family. Specification Attributes and Table Definitions applied on Classification will not be inherited to any Product below the Classification.

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.

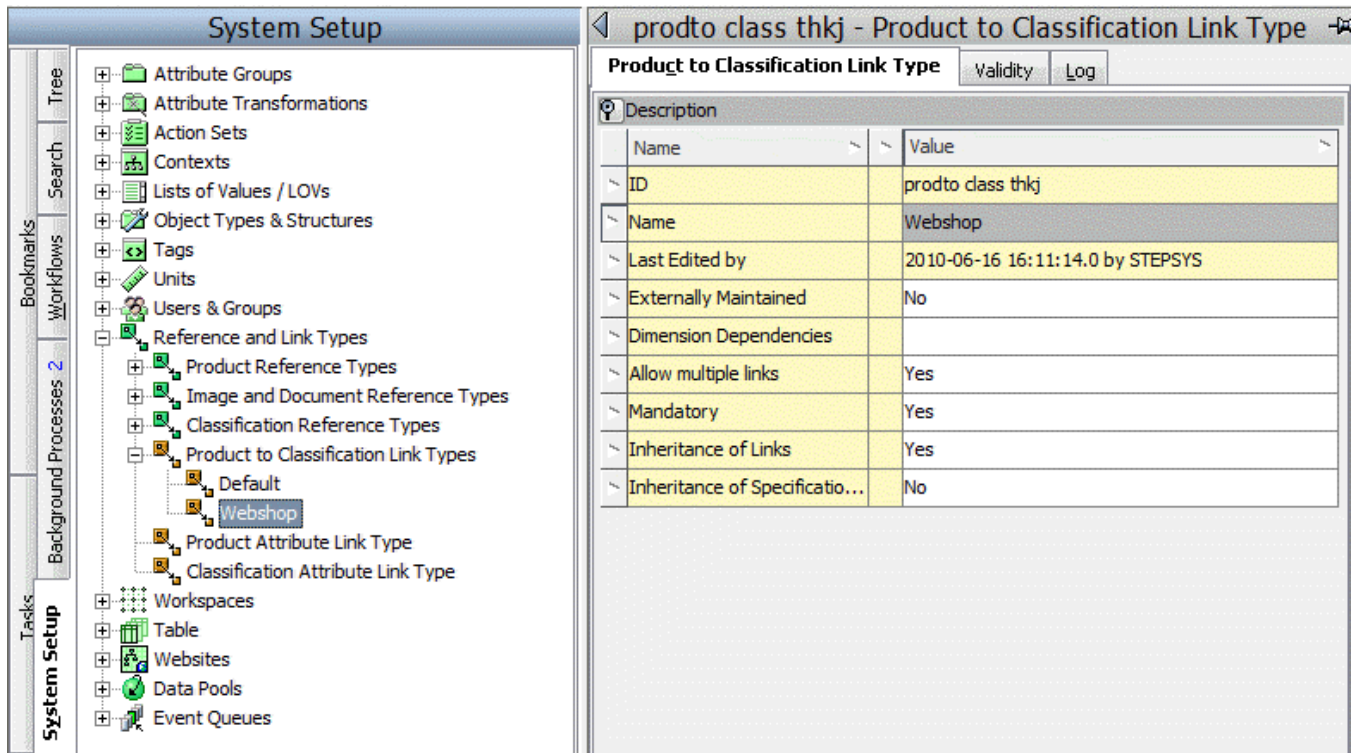
Inherit Links, Tables and Specification Attributes: Link from Product Family to Classification will be inherited to all Products below the Product Family. Specification Attributes and Tables applied the Classification will be inherited to all Products below the Classification.

Changing Product to Classification Link Types Settings

1. In **System Setup**, click the **Reference Types** Hierarchy.



2. Select relevant **Product to Classification Link Type**



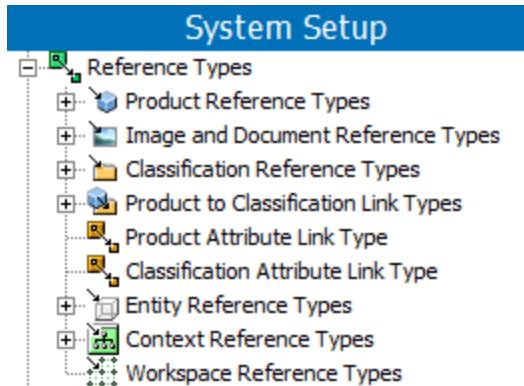
- Optionally, double-click the **Externally Maintained** field, to specify if **Product to Classification Links** of this type should be **Externally Maintained**.

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.

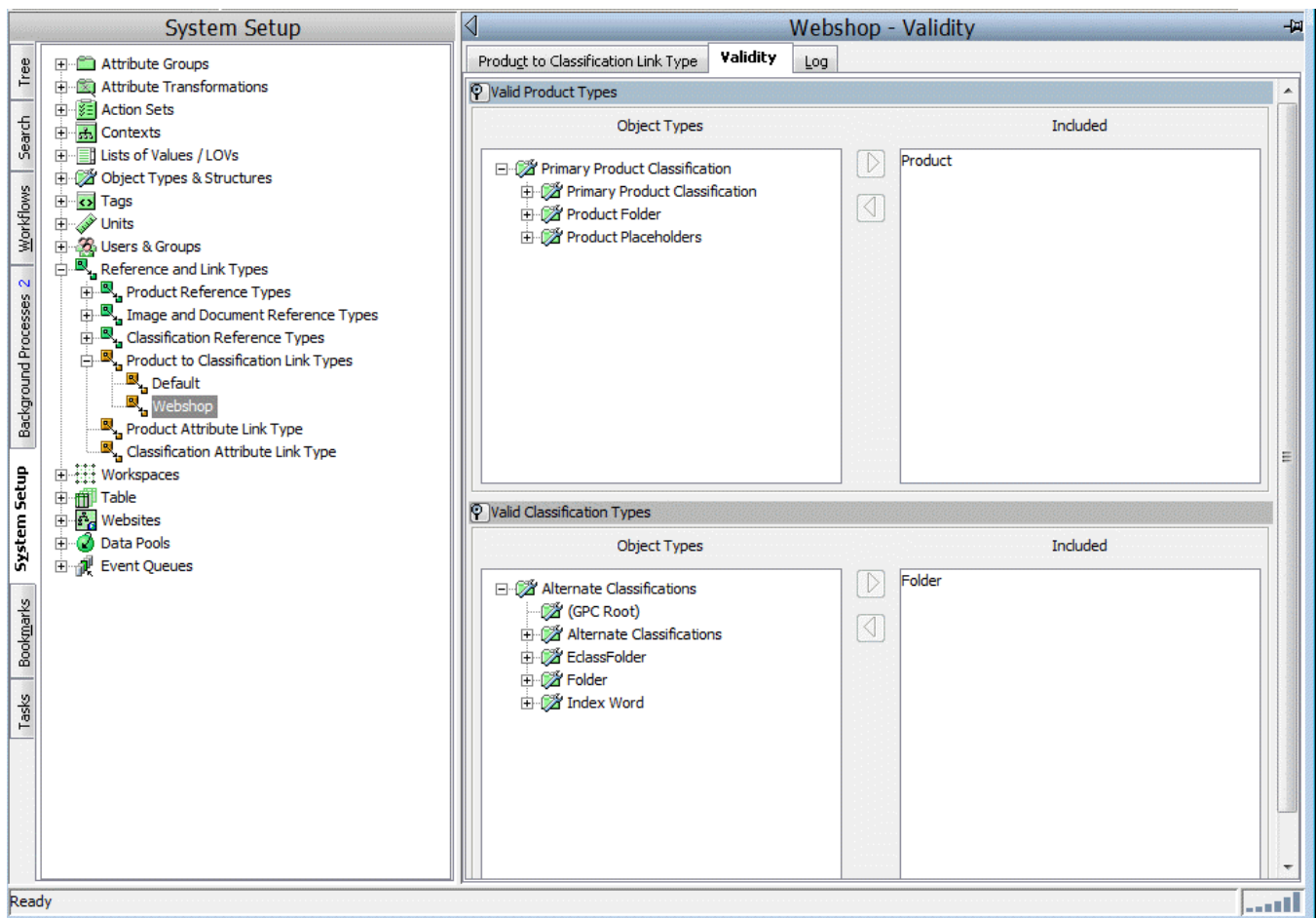
- Optionally, set the **Dimension Dependency** field to a **Dimension**, if the Link Type should be Dimension Dependent. It is to be recommended to only select one Dimension.
- Optionally, set the **Allow Multiple links** field to **Yes**, if the Link Type should be used to reference one Product to more Classifications. If **Allow multiple links** is set to **No**, a Product can be linked to one Classification only.
- 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.
- 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.
- 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**.

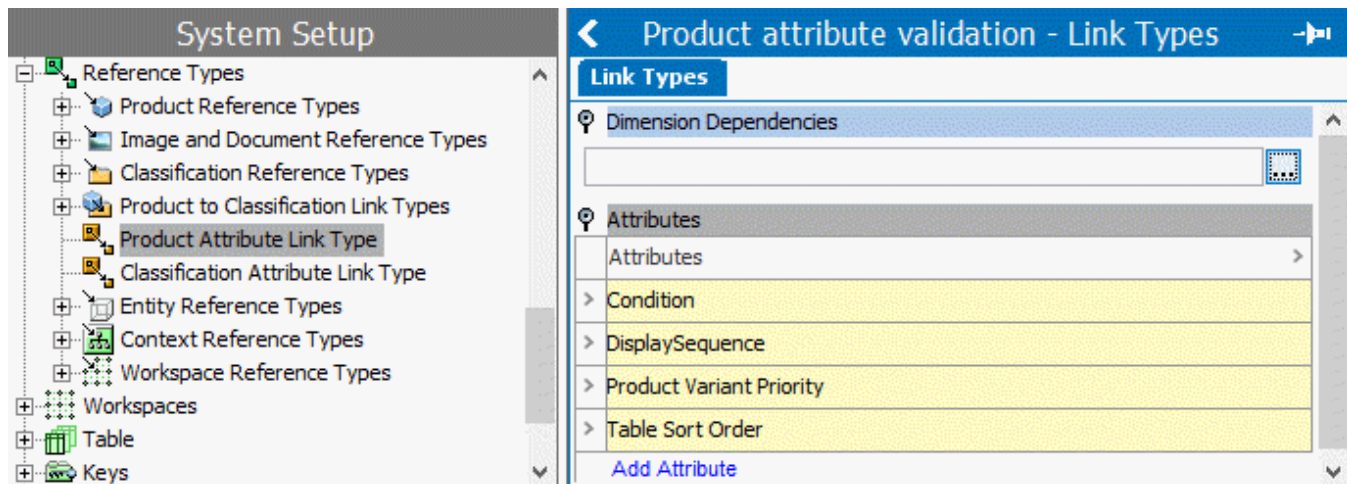
Important: If an Object Type is removed from the Included Object Types, and Product to Classification Links with this type already exist, Links will be deleted.

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.



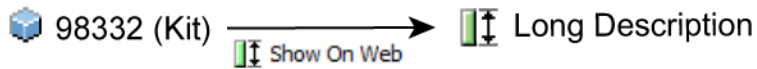
Dimension Dependency for Product to Attribute Links or Classification to Attribute Links

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.

- 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**.
- If a **Value Editor** opens, click **Add Value**.
- 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.

Product	References	Referenced By	Images & Documents	Commercial	Tables	Proof View	Packaging Hierarchy	Status	State Log	Tasks
Reference Group										
Reference Type	Target	Visibility	Thumbnail	Meta Attribute	Meta Attribute 1					
Part of Kit	RefProd2	All Languages	No Primary Image							
Secondary Image	New	All Languages	new							

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

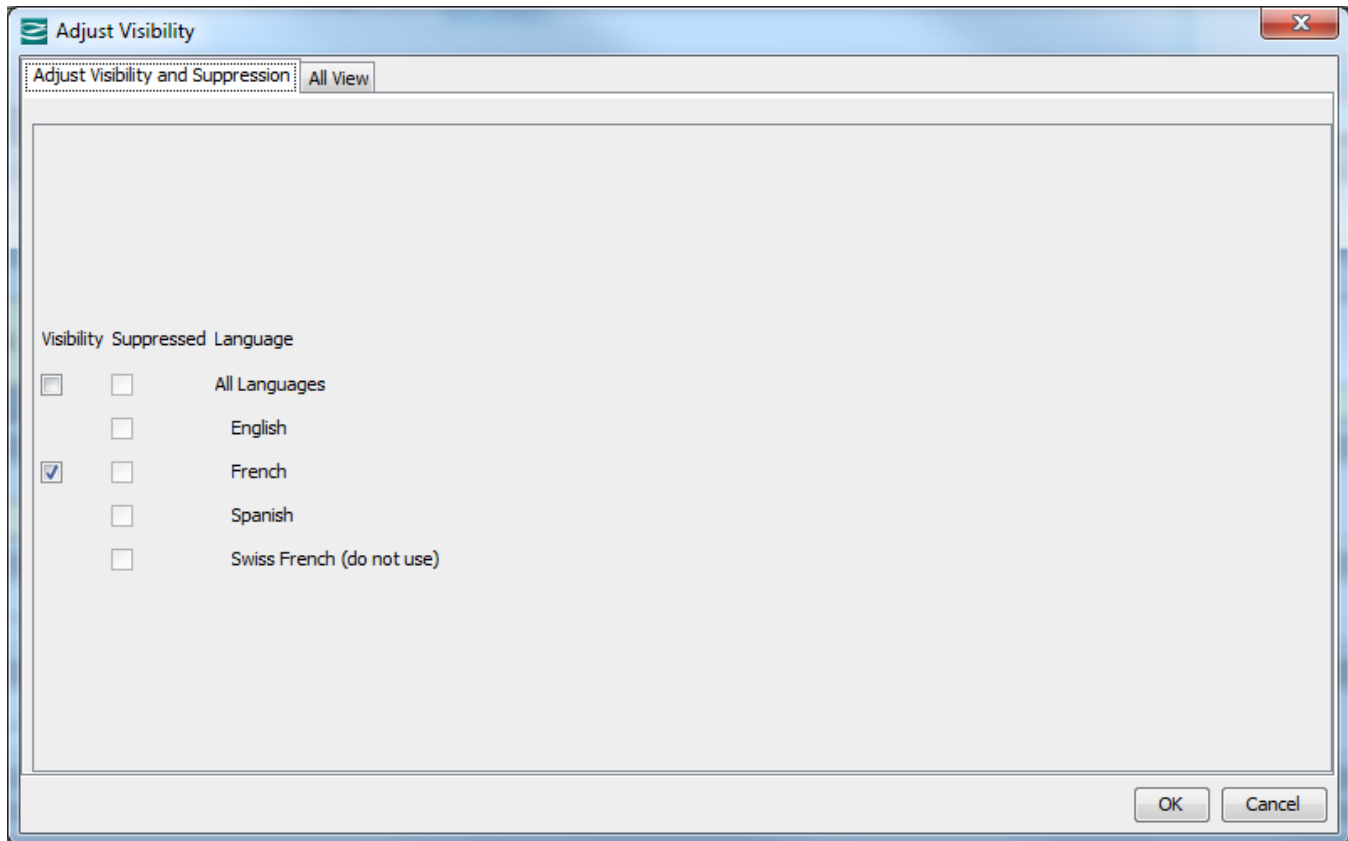
If a Reference is created in a local Context, it is visible in the local Context and no other Contexts.

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.

- In the **Product Hierarchy**, select the relevant product, and then click the **References** tab.

RefAttributeGroupProductEditorTest									
Reference Type	Target	ID	Name	Visibility	Thumbnail	Meta Attribute	Meta Attribute 1		
Part of Kit	+	RefProd2	RefProd2	French	No Primary Image				

- Click the **Visibility** field of the Reference whose visibility you want to change, and then click the ellipsis button (...). The **Adjust Visibility** dialog appears.

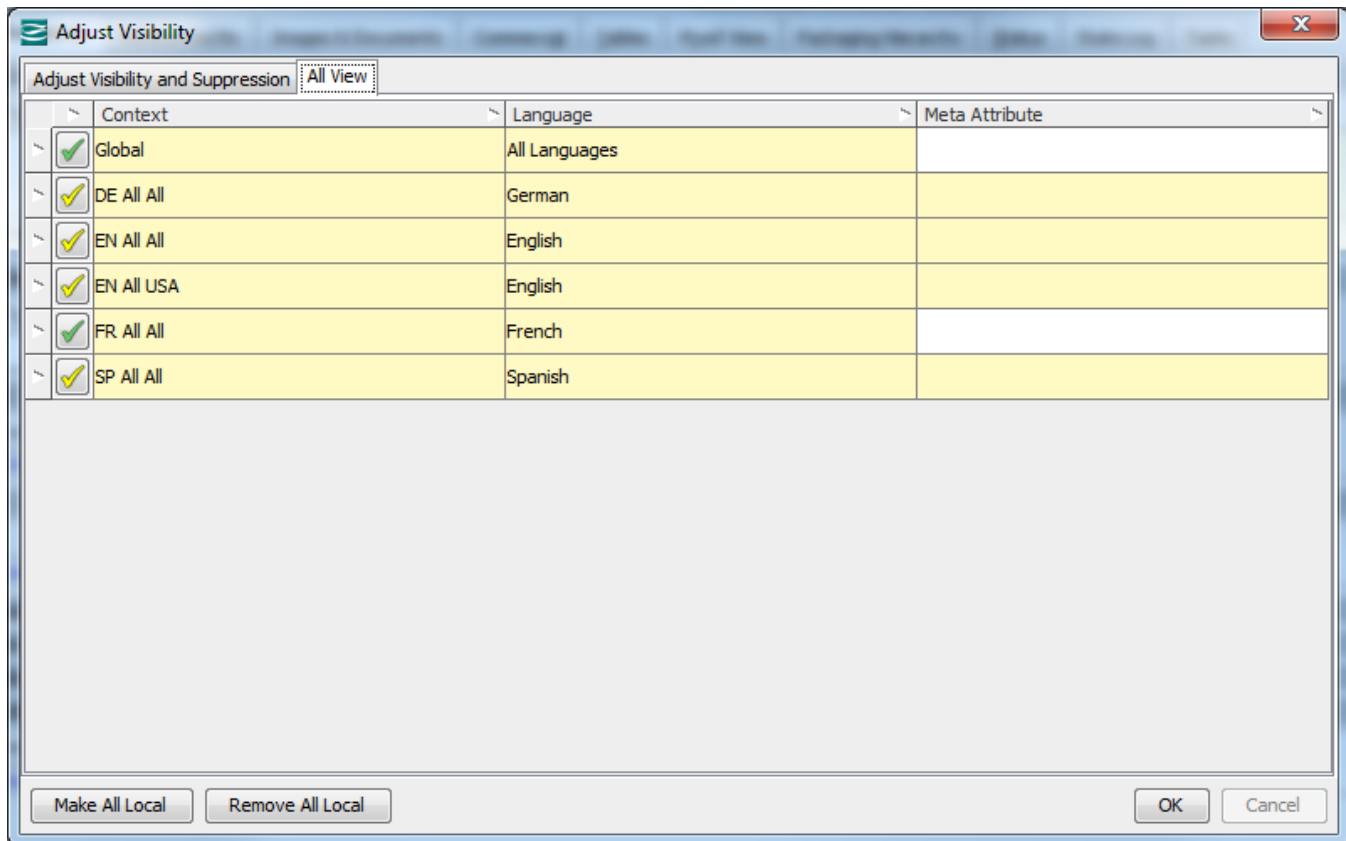


The **Adjust Visibility** dialog box is shown. It has a title bar with the Stibo logo and the text "Adjust Visibility". Below the title bar is a tabbed interface with two tabs: "Adjust Visibility and Suppression" (selected) and "All View". The main area of the dialog is a list box titled "Visibility Suppressed Language". It contains five entries, each with a checkbox and a label:

- ☐ All Languages
- ☐ English
- ☒ French
- ☐ Spanish
- ☐ Swiss French (do not use)

At the bottom right of the dialog are two buttons: "OK" and "Cancel".

- 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.



- To make the visibility of a Reference global, click the green **check mark** next to the relevant Context. The check mark turns yellow.
- 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.

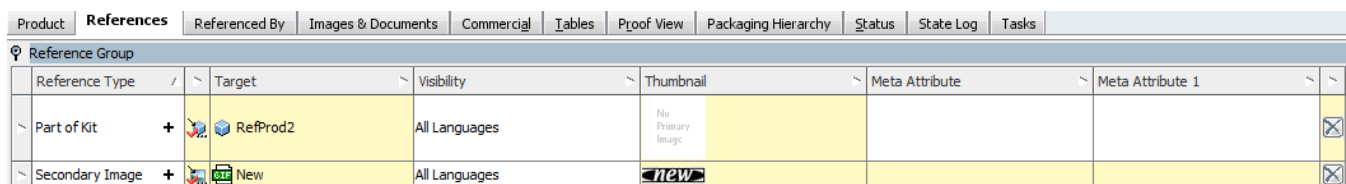
When the visibility of a Reference is set to global it inherits its settings from the Global Context.

To Specify Local Visibility for a Reference

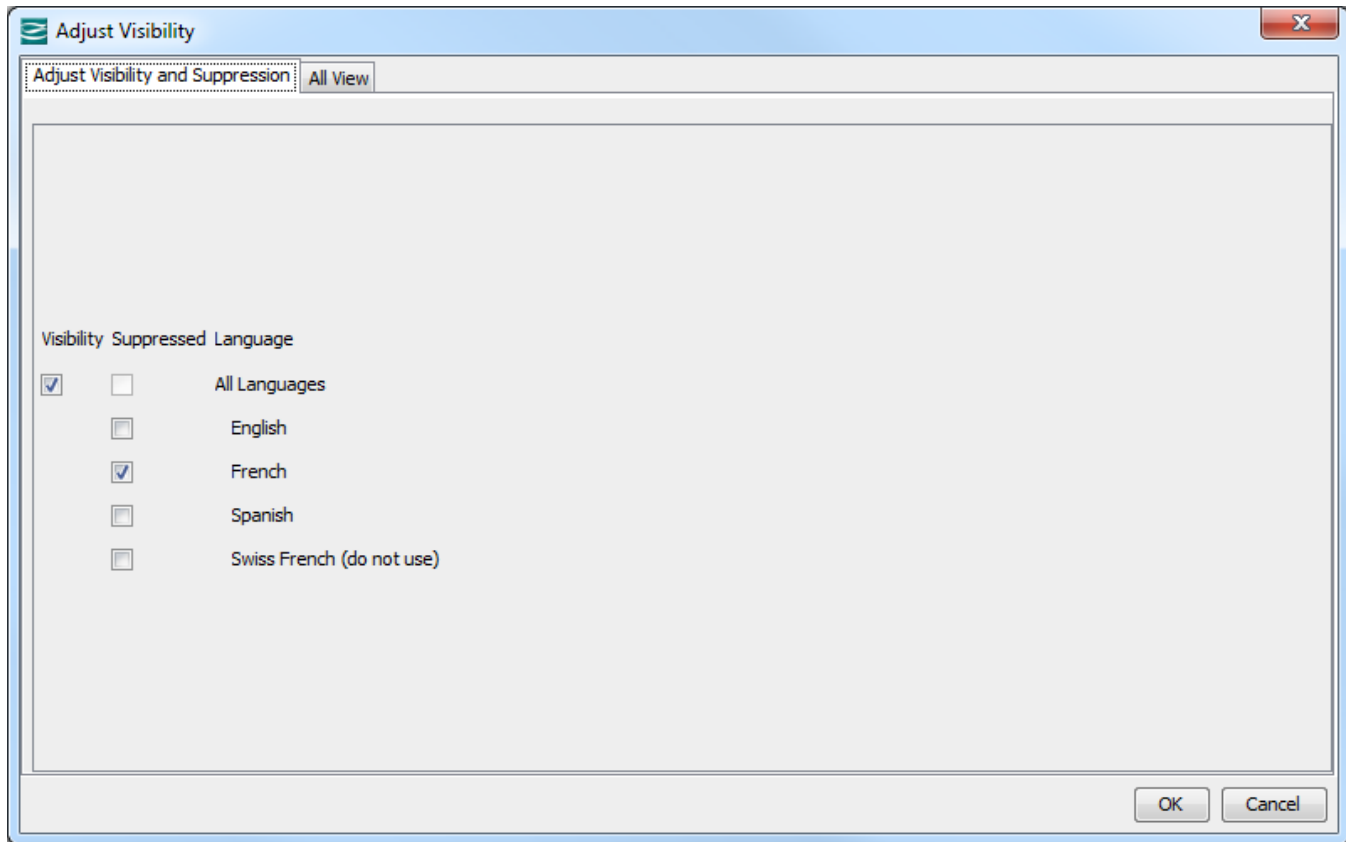
If a Reference is created in a Global Context, it is visible in a Global Context and in all other Contexts. In other Contexts, the Reference inherits its settings from the Global context settings.

However, you can specify that you only want a Reference to be visible in a local Context.

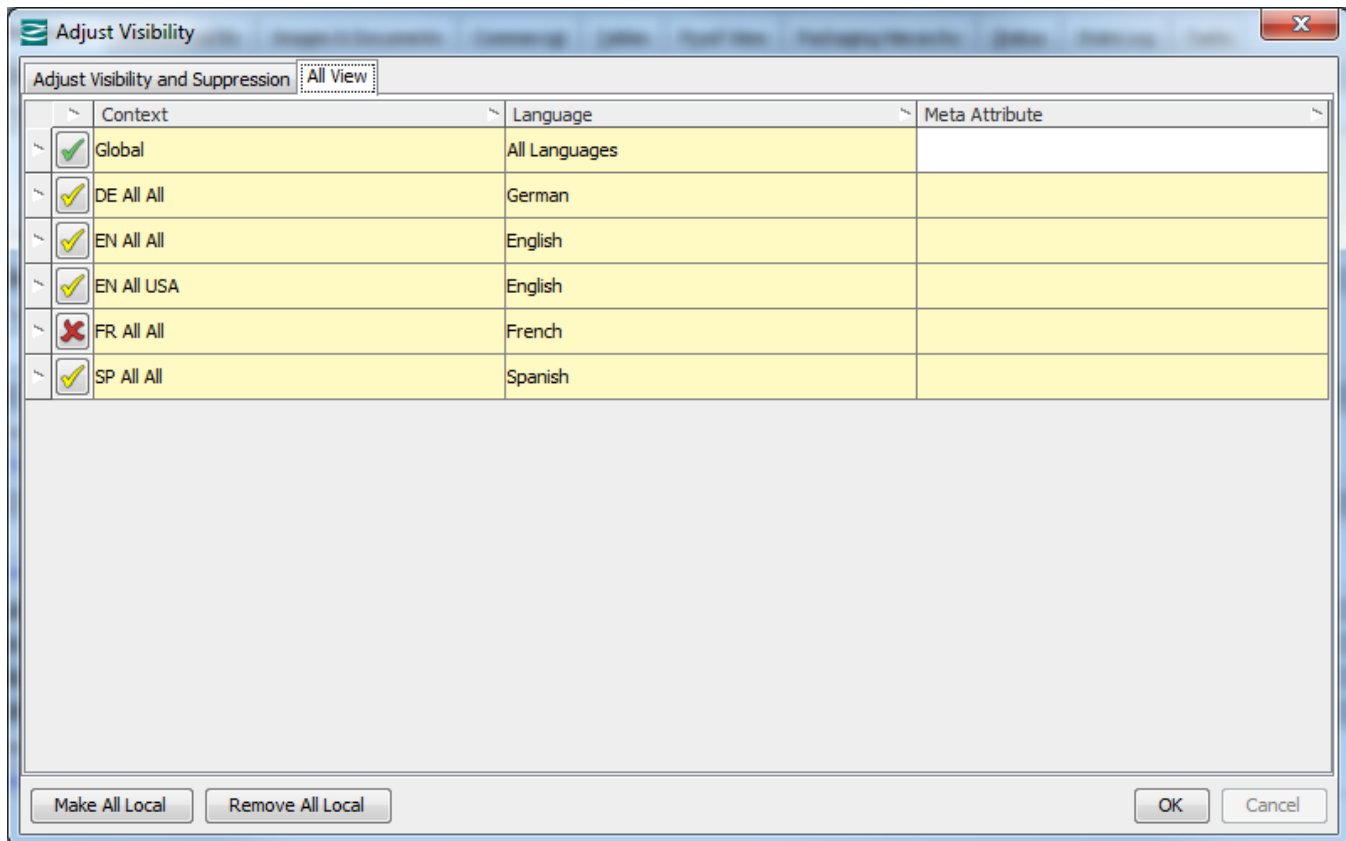
- In the **Product Hierarchy**, select the relevant product, and then click the **References** tab.



- Click the **Visibility** field of the Reference whose visibility you want to change, and then click the ellipsis button (...). The **Adjust Visibility** dialog appears.



- 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 its settings from the Global settings, and you can specify, for example, attribute values that only apply in the local Context.

To Suppress a Reference

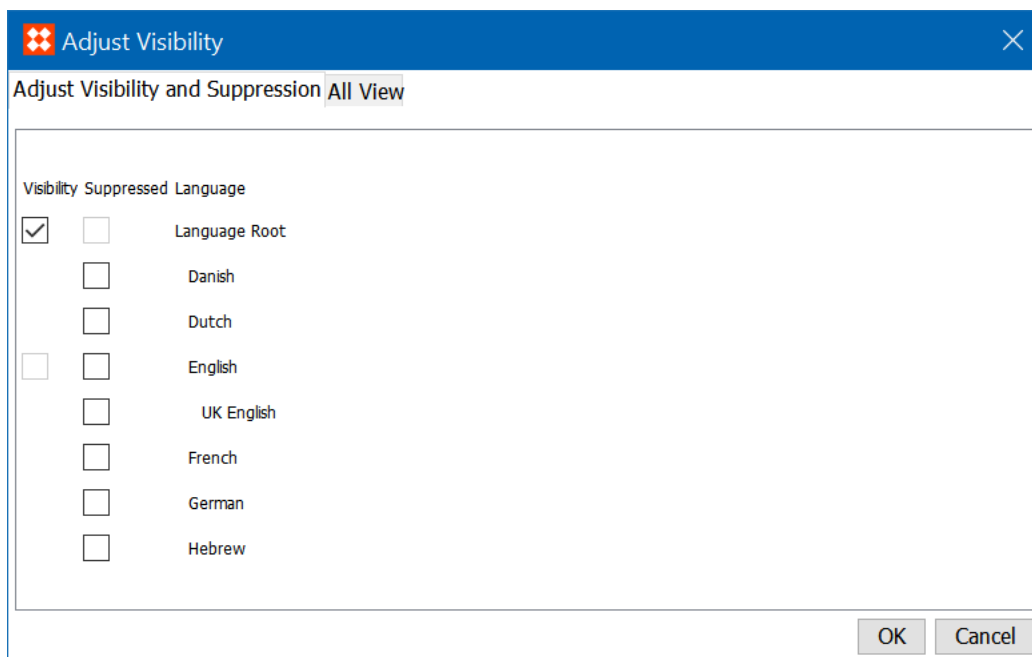
Users may suppress references so that they are not visible in specific contexts. As an example of a scenario where this might be useful, a use case: let us say a t-shirt product is being sold in many parts of the world. In STEP that t-shirt exists in multiple contexts organized by the languages spoken in the various countries in which that t-shirt is sold. An image of a person modeling the t-shirt is referenced by the t-shirt product. The image is acceptable in all contexts except one in which a specific country may consider the image to be culturally insensitive. In that instance, a user could access the References tab on that product in STEP, click the ellipses in the Visibility cell, and then suppress that image reference in the relevant context.

It is important to note that references may only be suppressed if they are dimension dependent, meaning that a dependency has been applied giving users the ability to make changes to that reference specific to context.

What follows are instructions on how to suppress references.

1. In the **Product Hierarchy**, select the relevant product, and then click the **References** tab.
2. If the reference to be suppressed is dimension dependent, then the **Visibility** field for that Reference will be clickable. Users must first click inside the Visibility cell, and then click the ellipsis button (). The **Adjust Visibility** dialog appears.

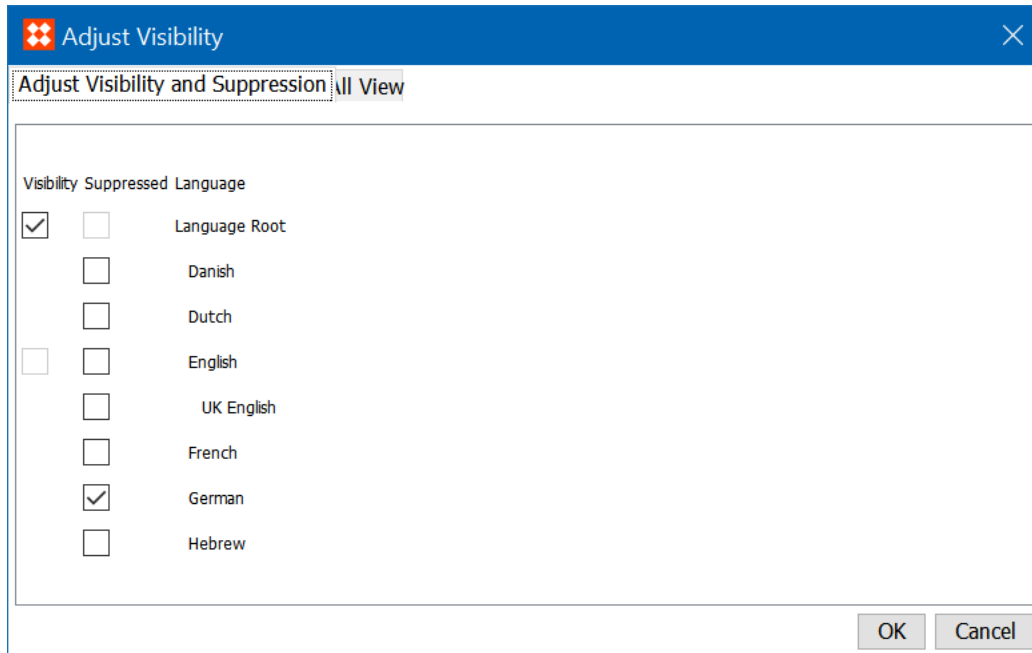
Important: If the reference has not been made dimension dependent, the ellipsis will not display when a user clicks into the Visibility cell. Only dimension dependent references may be suppressed.



The image shows a screenshot of the 'Adjust Visibility' dialog box. The dialog has a blue header bar with the Stibo logo and the text 'Adjust Visibility'. Below the header, there is a tab labeled 'Adjust Visibility and Suppression' and a sub-tab labeled 'All View'. The main content area is titled 'Visibility Suppressed Language' and contains a list of languages with checkboxes. The 'Language Root' checkbox is checked, while all other language checkboxes (Danish, Dutch, English, UK English, French, German, Hebrew) are unchecked. At the bottom right of the dialog, there are 'OK' and 'Cancel' buttons.

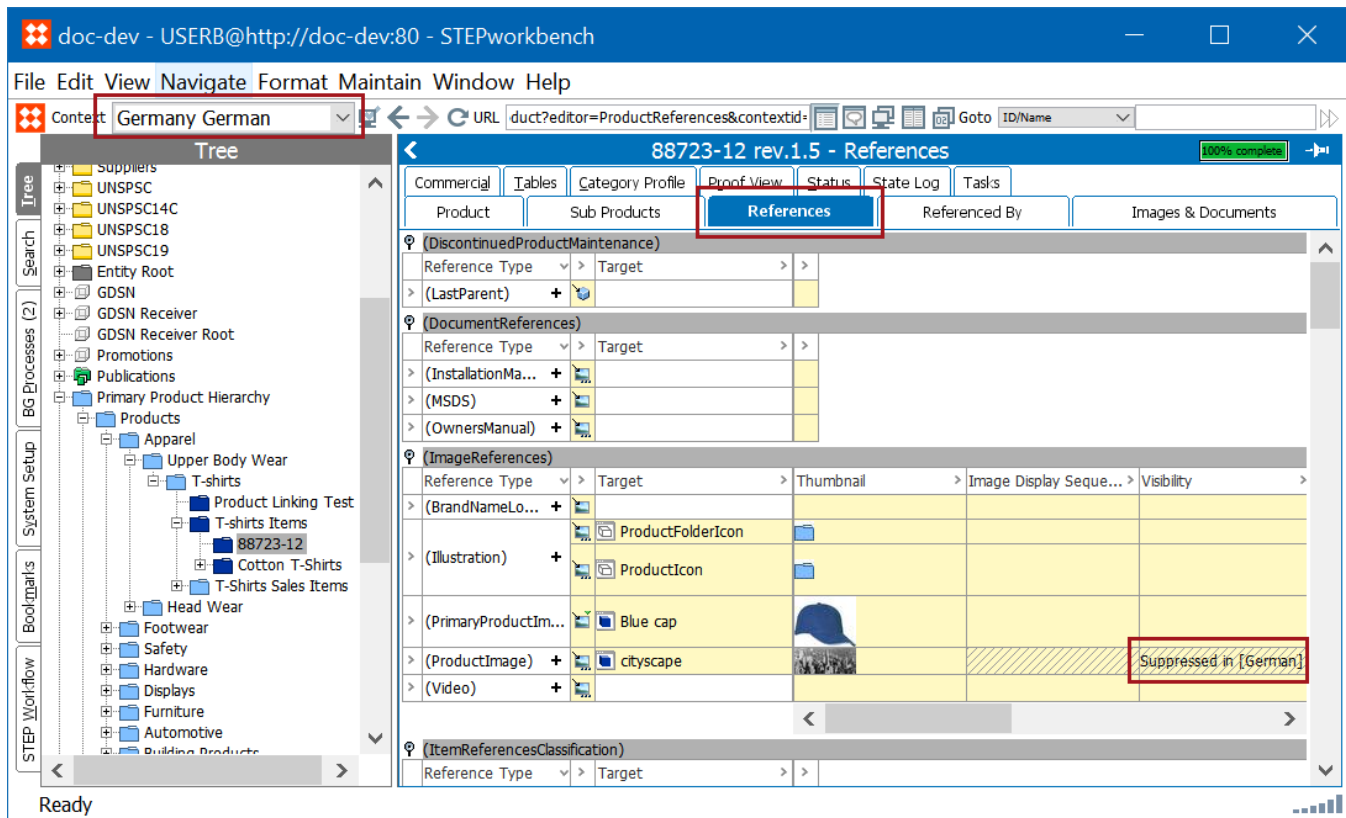
Visibility Suppressed	Language
☒	Language Root
☐	Danish
☐	Dutch
☐	English
☐	UK English
☐	French
☐	German
☐	Hebrew

- On the **Adjust Visibility and Suppression** tab of the **Adjust Visibility** window, specify if you want to suppress the visibility for one or more of the available contexts. To suppress the reference's visibility, select the desired contexts by checking the box beside the context in the **Suppressed** column. In this example, the reference is being suppressed in the German context.



- When all contexts in which the reference should be suppressed have been selected, click the **OK** button.

When a reference's visibility is successfully suppressed for a specific context, the reference's attributes are grayed out and the Visibility field displays the reference's status as Suppressed. As shown below, when the product is viewed in the German context, the suppressed reference's status is shown as Suppressed in [German], and its attributes are grayed out.



- If you want to change the visibility of a Reference from Suppressed to Visible, click the **Unsuppress Link** icon located at the far right of the reference attribute columns (highlighted below).

(ImageReferences)				
Reference Type	Target	Thumbnail	Image Display Sequen...	Visibility
(BrandNameLo...	ProductFolderIcon			
(Illustration)	ProductIcon			
(PrimaryProductIm...	Blue cap			
(ProductImage)	cityscape			Suppressed in [German]
(Video)				

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 ellipsis button (...).

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.

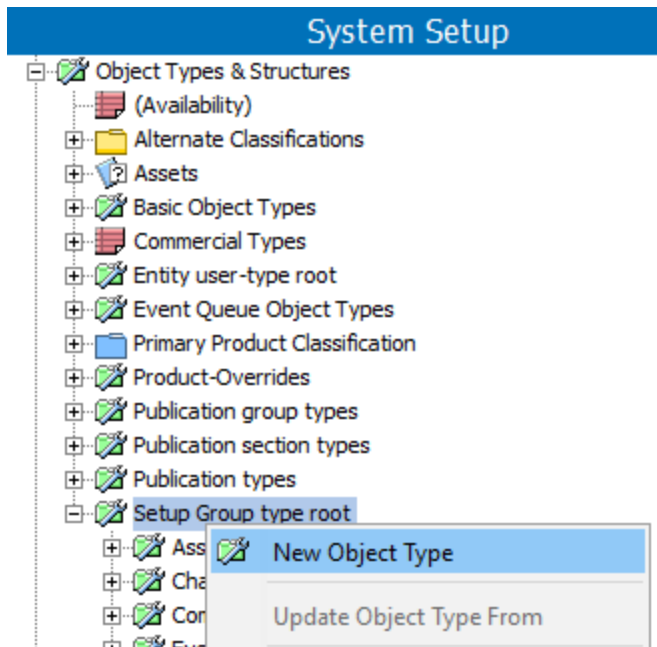
Setup Actions	Description
View Setup Groups	Users with this action applied will be able to View Setup Groups.
Maintain Setup Groups	Users with this action applied will be able to Create and Delete Setup Groups. They are also able to use the cut, copy and paste actions.

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**.

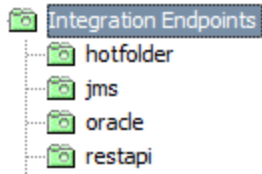


2. Key in **ID**, **Name** and Click **Create**.

The 'Create Object Type' dialog box is shown. It has fields for 'ID' (containing 'GatewayEndpoints') and 'Name' (containing 'Gateway Endpoints'). Below these is a 'Dimension Dependency' section with two checkboxes: 'Country' and 'Language', both of which are unchecked. At the bottom are 'Create' and 'Cancel' buttons. The 'Create' button is highlighted.

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.



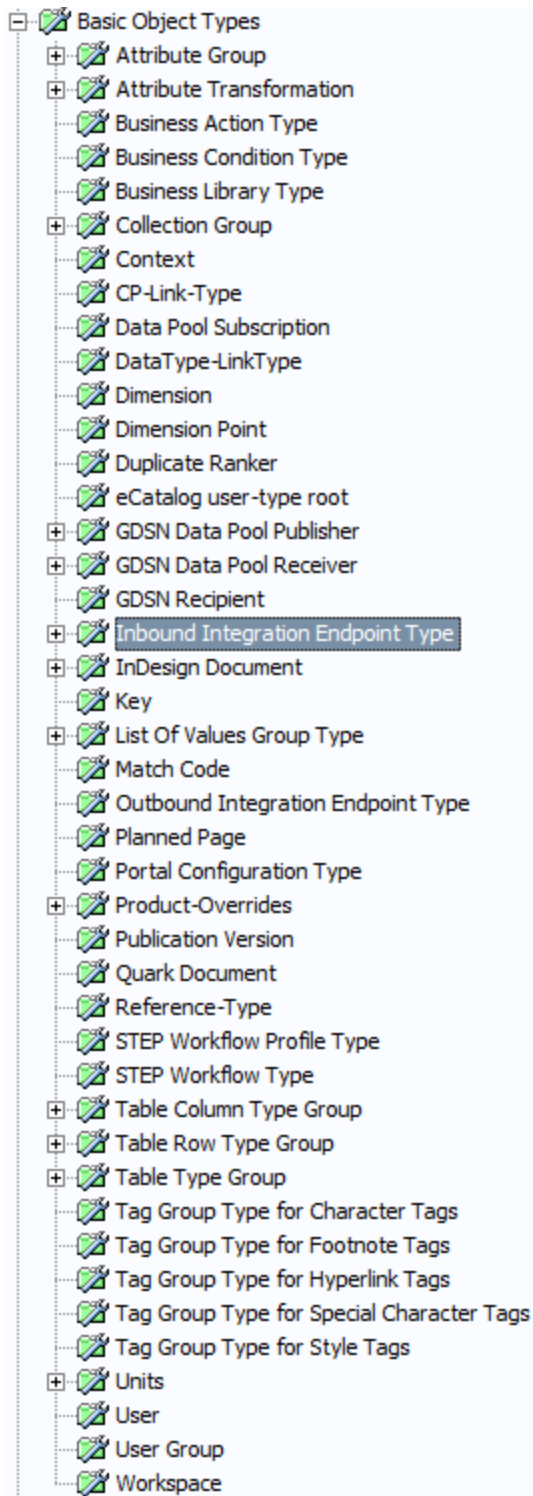
Defining Type of Objects to be Stored in Setup Groups

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:

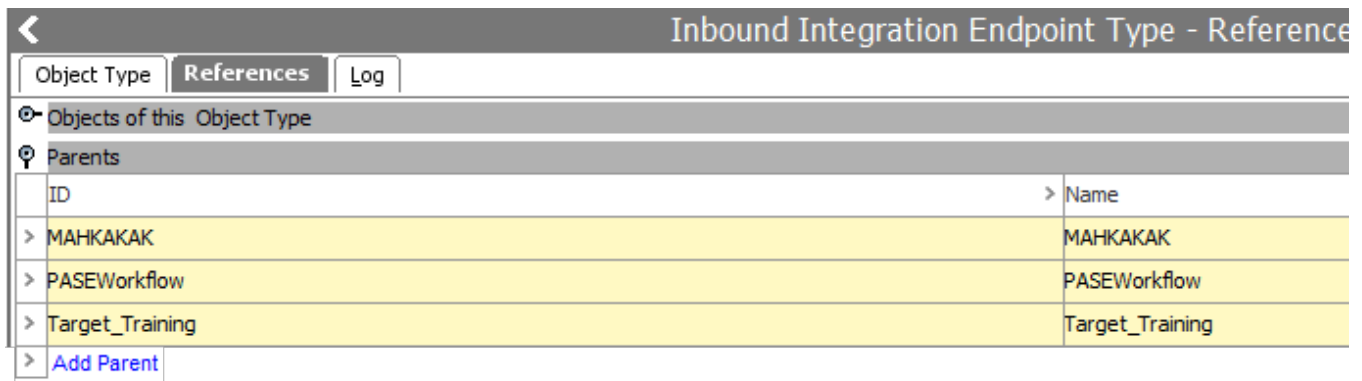
- Business Action Type
- Business Condition Type
- Business Library Type
- Inbound Integration Endpoint Type
- Outbound Integration Endpoint Type
- Portal Configuration Type
- STEP Workflow Type

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.

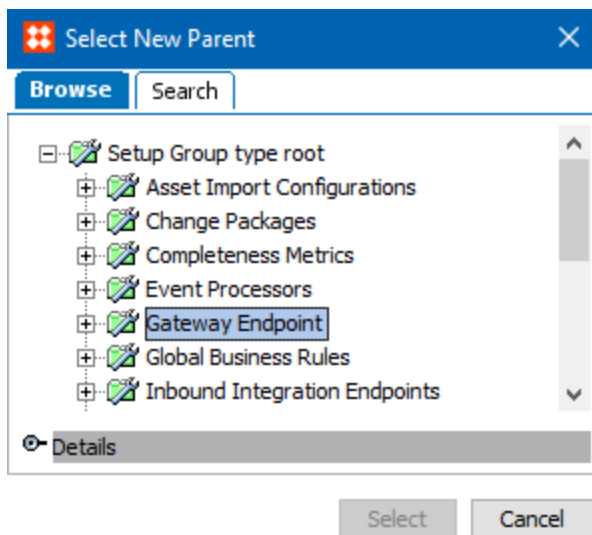
1. In **Object Types & Structures**, in **Basic Object Types**, click **Inbound Integration Endpoints**.



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**.

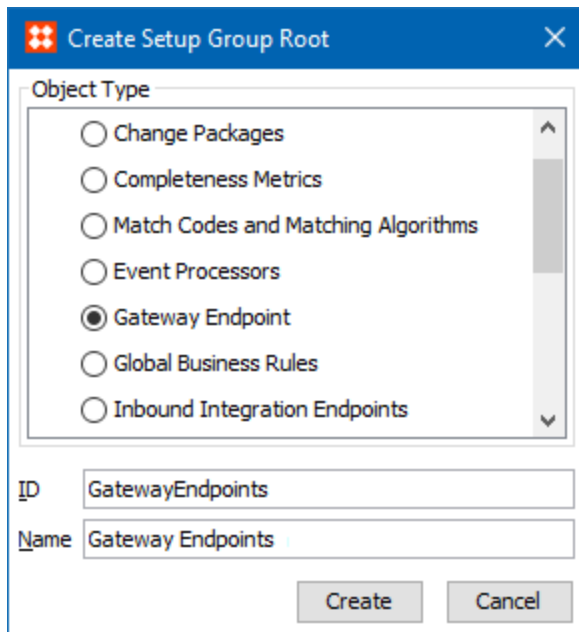
Creating Setup Groups in System Setup

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**.

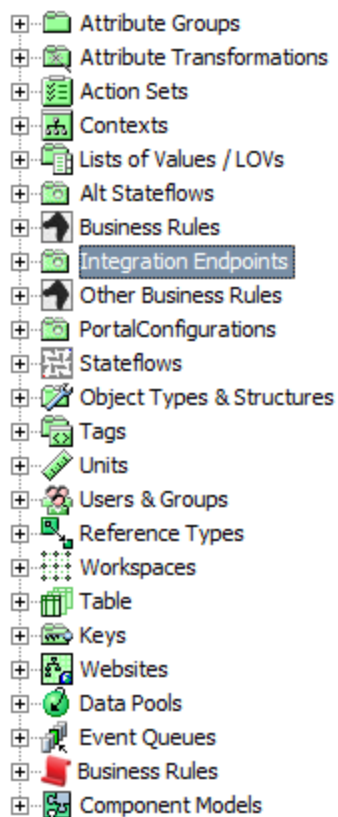
A **Create Setup Group Root** dialog appears.



The dialog box is titled "Create Setup Group Root". It contains a list of "Object Type" options: Change Packages, Completeness Metrics, Match Codes and Matching Algorithms, Event Processors, Gateway Endpoint (selected), Global Business Rules, and Inbound Integration Endpoints. Below the list are two text fields: "ID" with the value "GatewayEndpoints" and "Name" with the value "Gateway Endpoints". At the bottom are "Create" and "Cancel" buttons.

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.



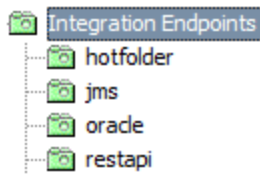
Setting Up Privileges Sample

In this example, two User Groups will be created.

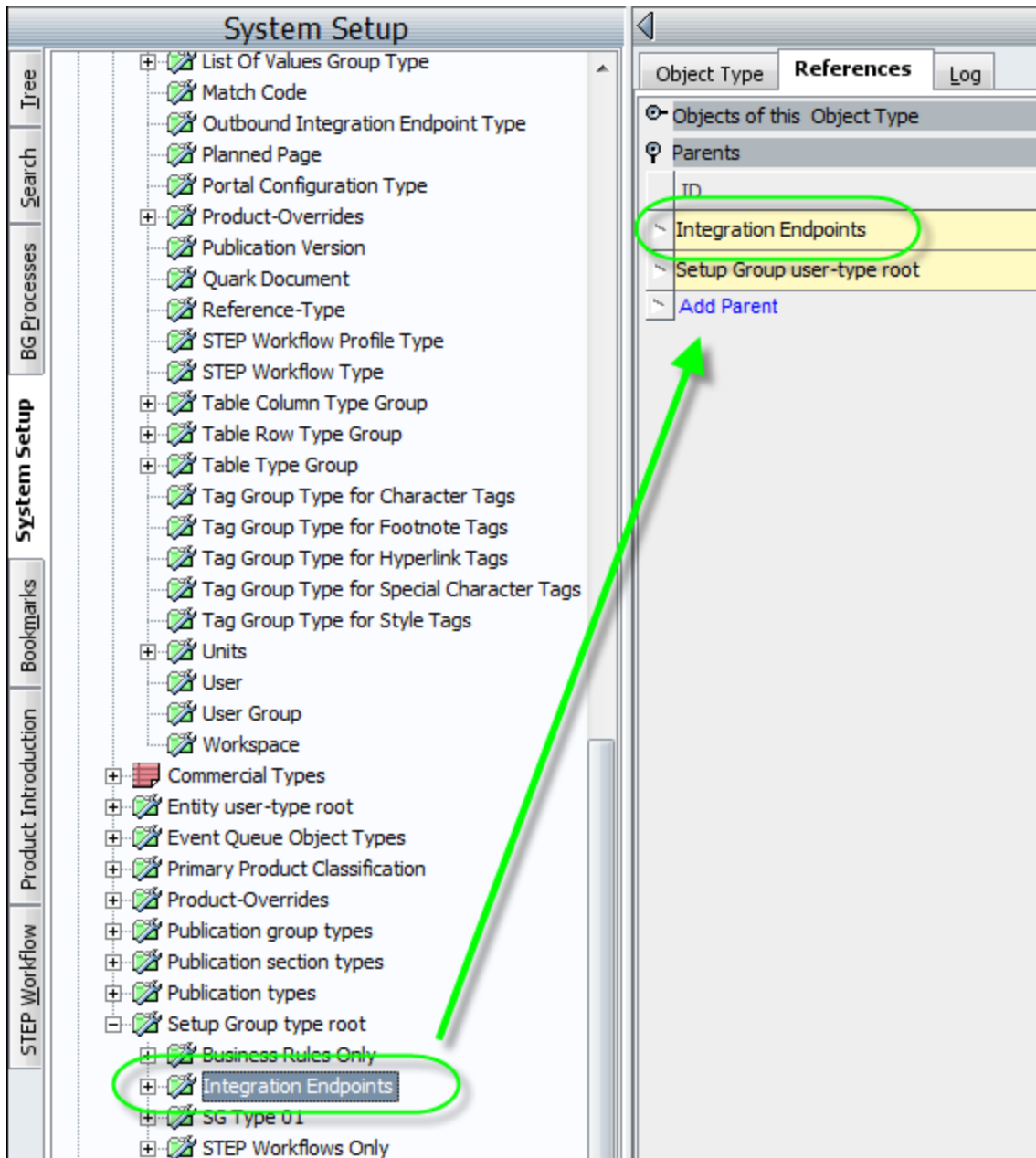
- A User Group named **View Setup Group and Integration Endpoints** will be created. Users linked into this group will be granted the privilege to view Setup Groups and Integration Endpoints.
- A User Group named **Maintain Setup Group and Integration Endpoints** will be created. It will inherit the 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.

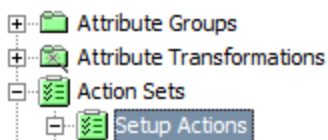
A structure of the Setup Groups could be as illustrated below.



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.

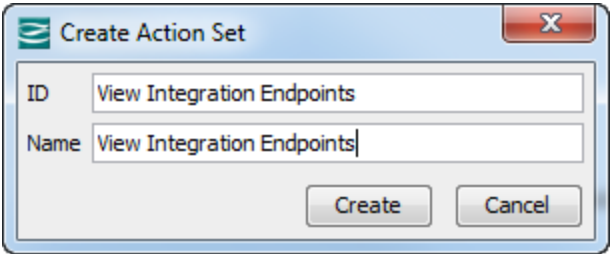


1. In **System Setup**, Select **Setup Actions** in Action Sets.

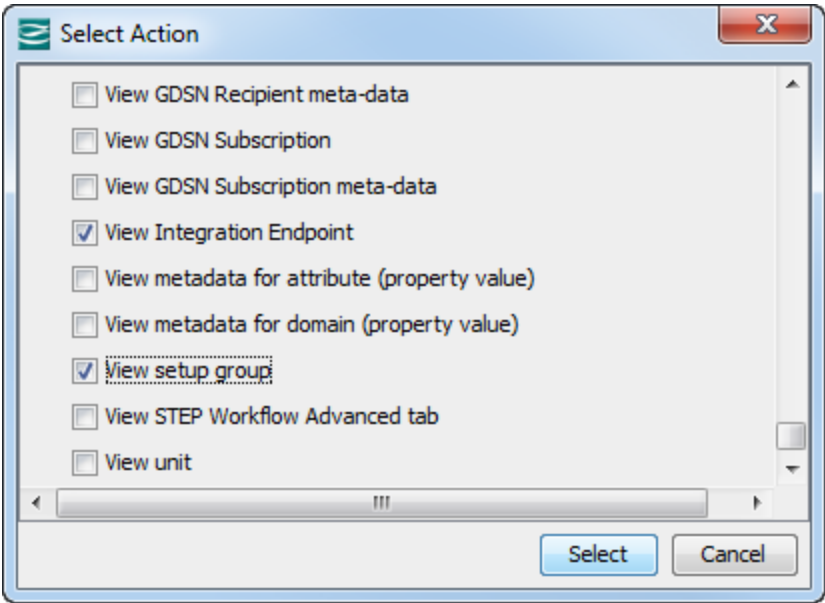


2. Right-Click and select **New Action Set**.

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**.



Action Set		Log
Description		
Name	Value	
ID	View Integration Endpoints	
Name	View Integration Endpoints	
Action Type	Setup Action	
Actions		
Actions		
View Integration Endpoint		
View setup group		
Add Action		

- 6. Repeat step 1-5 and create an Action Set containing the actions **maintain setup group** and **maintain integration endpoint**.

Action Set [Log](#)

Description

Name	Value
ID	Maintain Integration Endpoints
Name	Maintain Integration Endpoints
Action Type	Setup Action

Actions

Actions
Maintain Integration Endpoint
Maintain setup group

[Add Action](#)

7. In **Users & Groups**, right-click and select **New Group**.

Create Group

ID: 1234

Name: View Setup Groups and Integration Endpoints

[Create](#) [Cancel](#)

8. Key in **ID** and **Name** and click **Create**.

9. In **User Group editor**, click **Privilege Rules** tab.

System Setup

- Users & Groups
 - AdminPortal
 - Asset MGR
 - Brand
 - Brand Associate
 - Brand Managers
 - DTP Managers
 - DTP Operators_No Public View Privileges
 - Image Group
 - Integration

Integration - Privilege Rules

Group **Privilege Rules** GUI Set-Up Log

Setup Privileges

Action Set	Attribute G...	Setup Group	Language	Country

[Add Privilege](#)

User Privileges

Applies to	Action Set	Attribute G...	Object Type	Group	Language

[Add Privilege](#)

☐ Read Only

10. In **Setup Privileges**, click **Add Privilege**. The Edit Privilege Rule dialog will appear:

Edit Privilege Rule

Apply to Node:

Action Set: View Integration Endpoints (View Integration Endpoints)

Attribute Group:

Setup Group: Integration Endpoints (Integration Endpoints)

Valid for Object Type:

Apply to Group: View Setup Groups and Integration Endpoints (1234)

Dimensions:

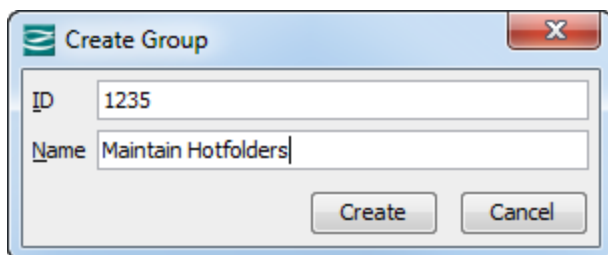
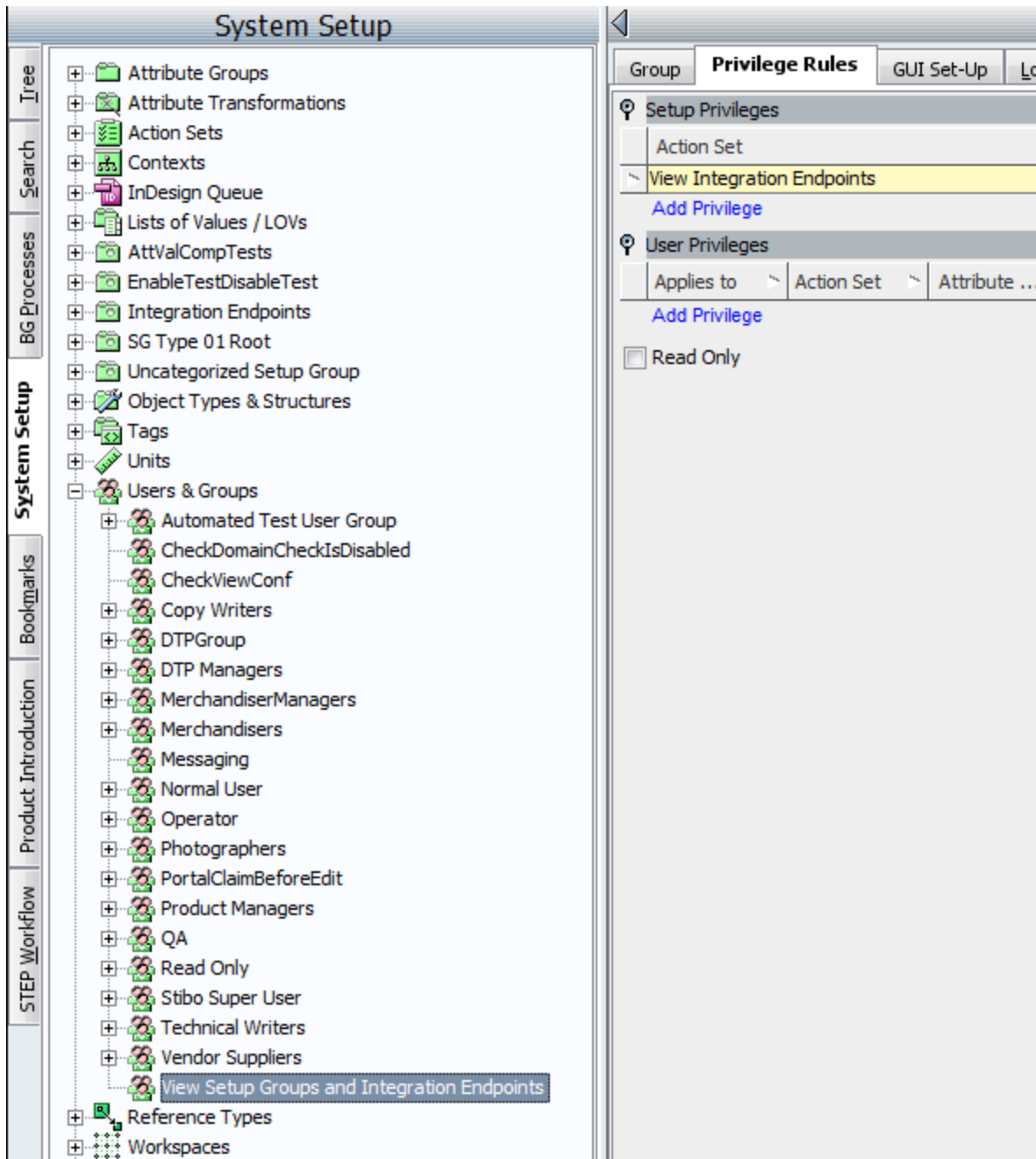
Market: <ANY>

Language: <ANY>

Country: <ANY>

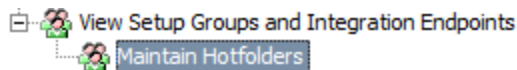
Save Cancel

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**.

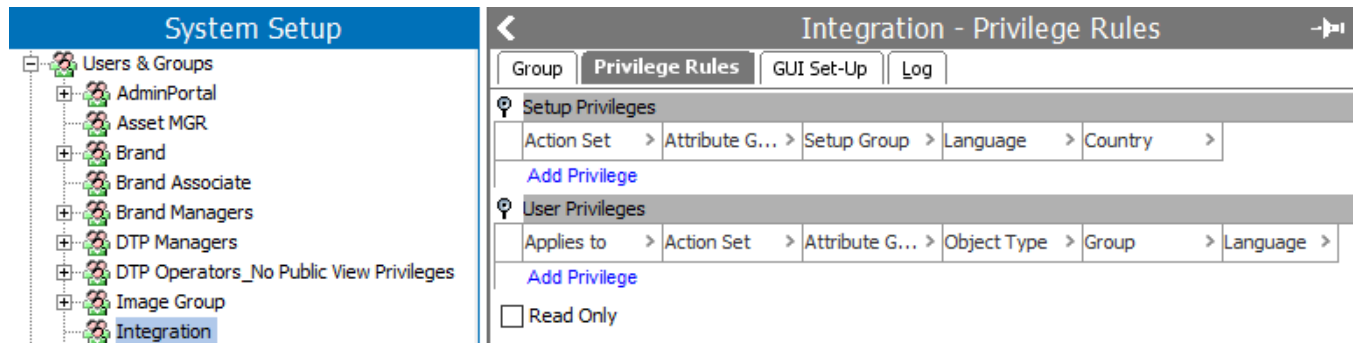


14. Key in **ID**, **Name** and click **Create**.

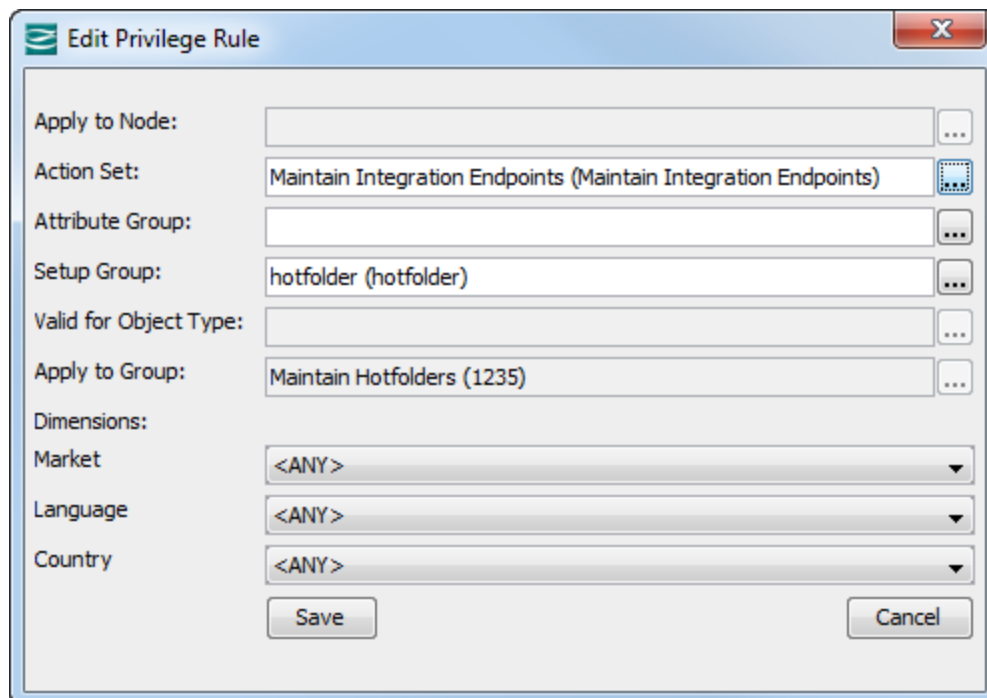
The **User Group** will appear below the User Group **View Setup Groups and Integration Endpoints**.



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**.

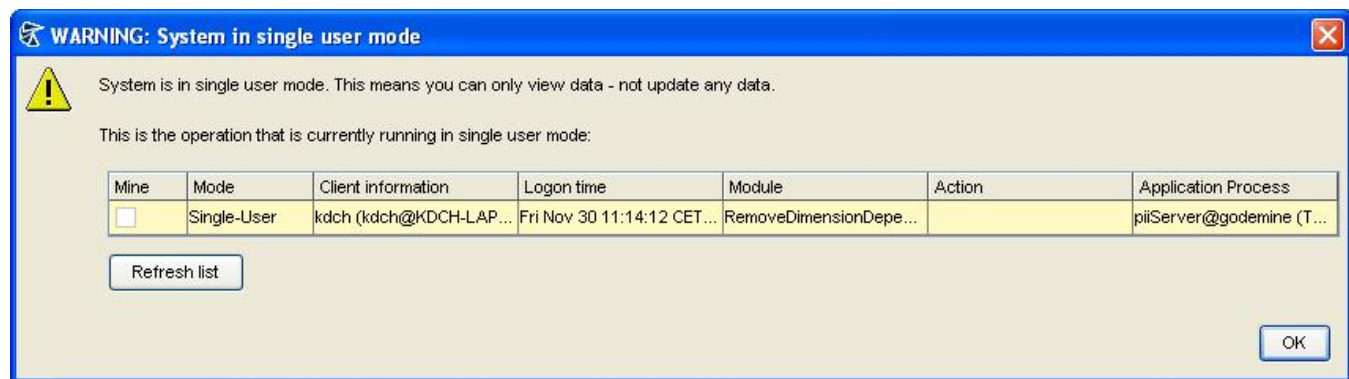
18. In Setup Group, apply the setup group **hotfolder**.

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".

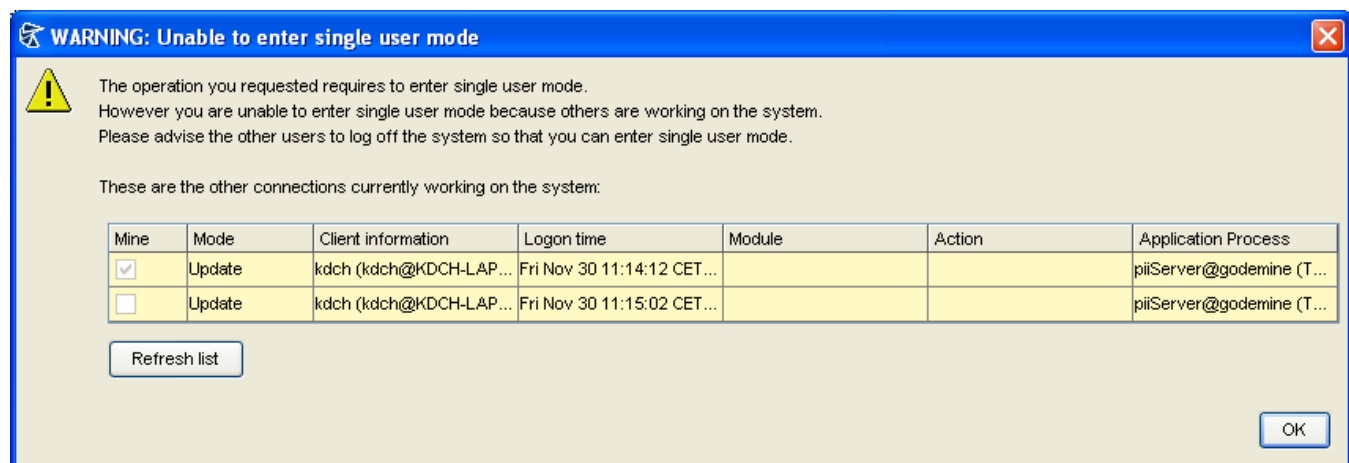
Single User Mode

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.

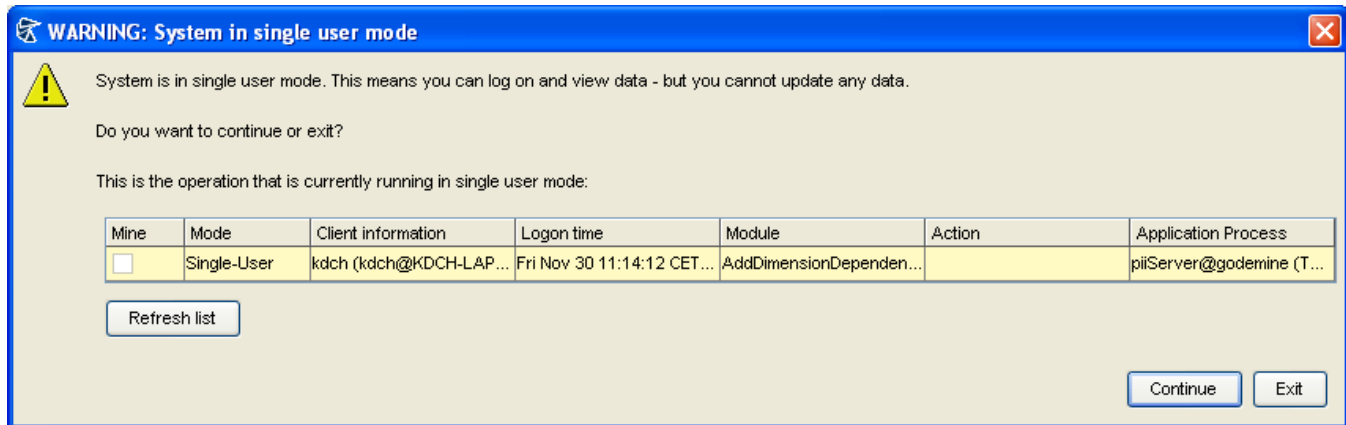


You can still view data when the system is in Single Update Mode. However, if you make changes and try to update when another connection is in Single Update Mode, a warning appears.



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

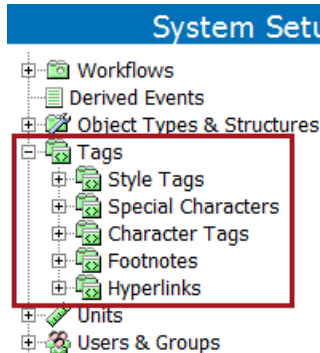
To avoid entering Single User mode when changing attribute definitions, the following procedure can be used:

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

A **tag** is a markup notation used to indicate specific formatting or special characters. A tag is identified by a unique tag name, and is available to be inserted into text based attributes in both workbench and Web UI.

Tags are maintained in **System Setup** in the **Tags** folder.



All tags in STEP are 'Media Neutral.' This means that data entered into and maintained in the STEP system is independent of whatever media it delivers exported files to. There are no embedded codes within the data that are specific to any one type of media, such as HTML. This means that STEP solves the issue of users having to enter media-specific command codes within the data stream. In this way, the same attribute could be output to any media and provide the necessary command codes that are relevant to that output media.

STEP supports several different types of output, including but not limited to InDesign, Microsoft Word, web applications, and any number of custom applications.

Tag systems in STEP

There are two systems of tags in STEP:

- **'Paired tags'** have a separate start and stop tag, e.g. `<bold>xxxx</bold>`. An example in STEP would be Style Tags.
- **'Unpaired tags'** do not have a separate start and stop tag. Instead they are 'self-contained', e.g. `<in/>` or `<ft/>`. An example in STEP would be Character Tags.

For every STEP tag created the user can define what the appropriate corresponding command codes are for all the different media outputs. For example, the `<bold>` `</bold>` tags may be converted to `<b>` and `</b>` for an HTML3 output format, `<ct:Bold>` and `<ct:>` for the InDesign output format, or the tags could be stripped out entirely for a custom output format.

Types of Tags in STEP

There are five main types of tag:

Special Character

Special Characters are Unicode characters, which in some cases need to be formatted. It is important to know that not all Unicode characters are available in all fonts. Consult the web if there is a question about a particular font style.

Two 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
Line Feed (LF)	Line Feed (LF) is a hard return

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.

Character Tags

Character Tags allows users to maintain STEP tags within the system and convert them to one or more characters used in special fonts upon exporting the data. Sometimes these characters are known as Dingbats. For example, they can be used to export non-Unicode characters such as a telephone  or a special arrow  (or any other character or characters that are not Unicode characters).

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. Consult the web for more available special characters.

Style Tags

Style tags apply changes to how text looks. For example, it can make plain black text bold, blue, or with bullets. Style Tags must have an opening and closing tag, for example, <bolditalic> </bolditalic>. The system automatically inserts the symbols < />. Style Tags are used with STEP Tables in order to apply styling to the contents of table cells.

Footnotes

Footnotes are used to add an additional piece of information to attribute values when the attributes are used in STEP Tables only. If footnotes are added to attribute values outside of a STEP Table, they will not output expected result. Footnotes are added / edited within the Rich Text Editor, so using these options will open that dialog. They are also required in order to use the Footnote Transformation within STEP tables.

Hyperlinks

Hyperlinks are used to define the formatting of hyperlinks when they are used in attributes with the value base type as 'text'. The hyperlink can either precede, come in the middle of, or come after the designated text. The types of hyperlink tags are: FILE, MAILTO, FTP, HTTP, and HTTPS. For print, these tags may be used on PDF documents created in InDesign. If the reader opens the PDF and clicks on a hyperlink in the PDF, they are then directed to the relevant external website.

Note: Hyperlink tags are not attributes with the validation base type of a 'URL'. Attributes with the validation base type of a URL do not allow for any other text in the value field, except for a URL.

Working With Tags

To create tags in STEP, a user will need to do the following:

1. Create tags- See **Creating Tag Groups and Tags** in this documentation.
2. Render tags properly- See **Configuring / Rendering Tags** in this documentation.
3. Establish proper output formatting- See **Configuring Output Formatting** in this documentation.

To make tags visible in Web UI, see the **Attribute Value Components** topic in the **Web UI Getting Started / Web UI Setup and User Guide** documentation.

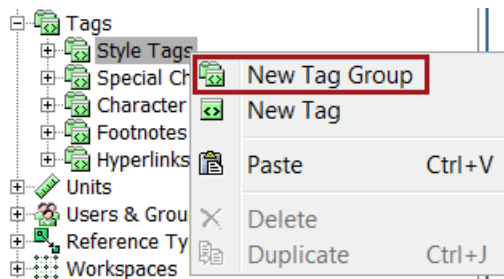
Creating Tag Groups and Tags

When creating tags, it is often helpful to first put them in tag groups. This can allow for better organization and stronger control when combined with user privileges.

Tag Groups

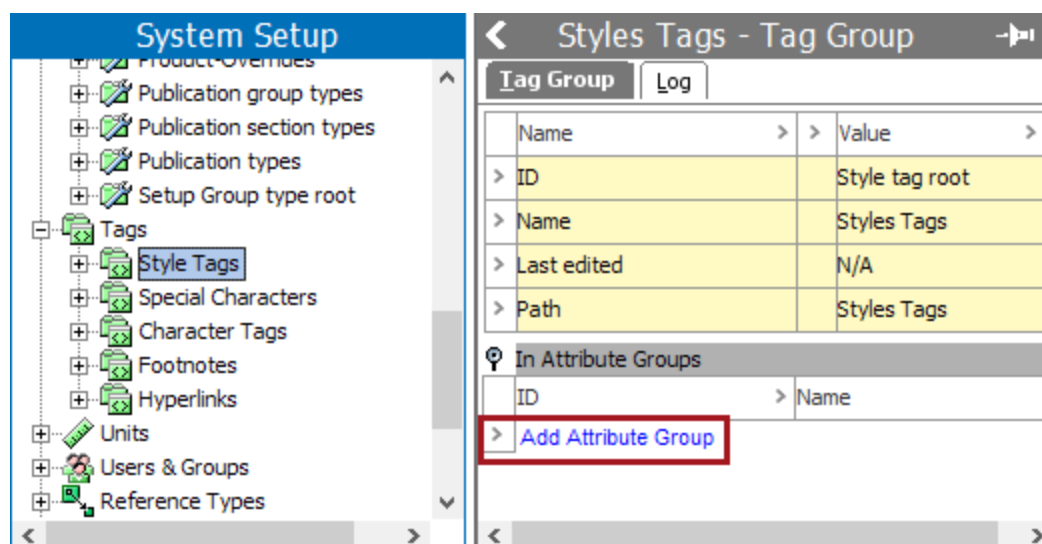
To create a tag group under one of the five types of tags, follow the steps below:

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**.

The Group can be linked into an attribute group and used in combination with privileges to restrict users in maintaining and/or using certain tags.



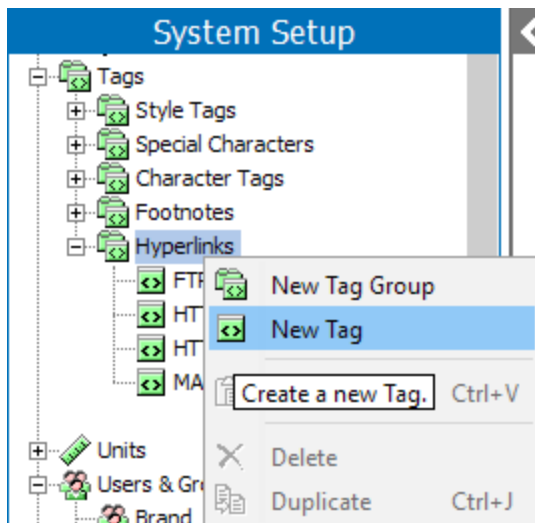
The specific privileges used to have tighter control over tag usage are 'Use tag' and 'Maintain tags'. See the **Users and Groups / Super User Guide** documentation for more information on how to set up privilege groups.

Tags

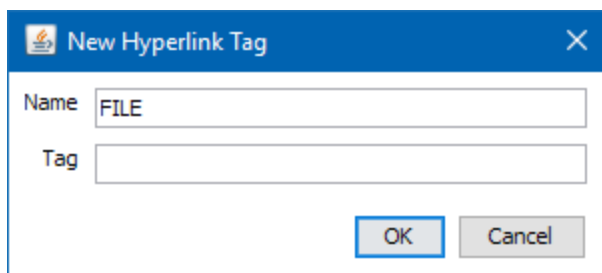
The way to create the various tags are overall the same, though for special characters, character tags, style tags, and hyperlinks, there are some special consideration to keep in mind when creating them. Both how to create tags and any special considerations are described below.

Creating Tags

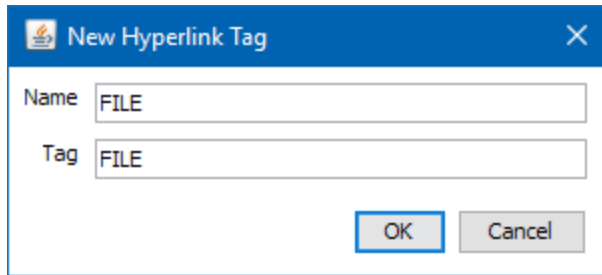
1. Go to the **System Setup** tab, expand **Tags**.
2. Right-click desired tag type, then select **New Tag**.



3. In the dialog box that appears, enter a display **Name** for the tag. The name can be up to 80 characters long, and there can be spaces between words.

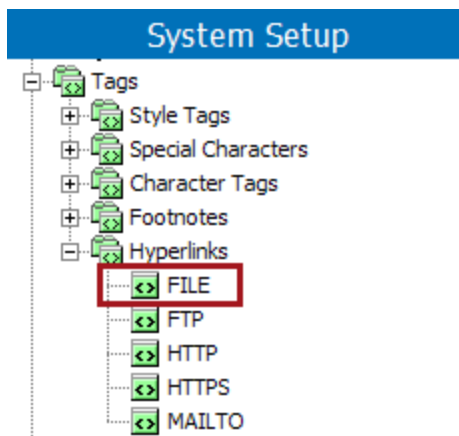


4. In the **Tag** field, enter a tag. The tag will be used for example, in a document when importing files using STEP tags. Because it can get confusing if the name of the hyperlink, which is what is displayed in the UI, is different than the tag, it is recommended to keep the name of the hyperlink and the tag itself the same. Spaces or the special characters: less than, <, greater than, >, or a forward slash, /, are not permitted to be used in the tag.



Note: All STEP tags are case-sensitive. When loading files with attribute values that include STEP tags, they must exactly match how the tag was set up.

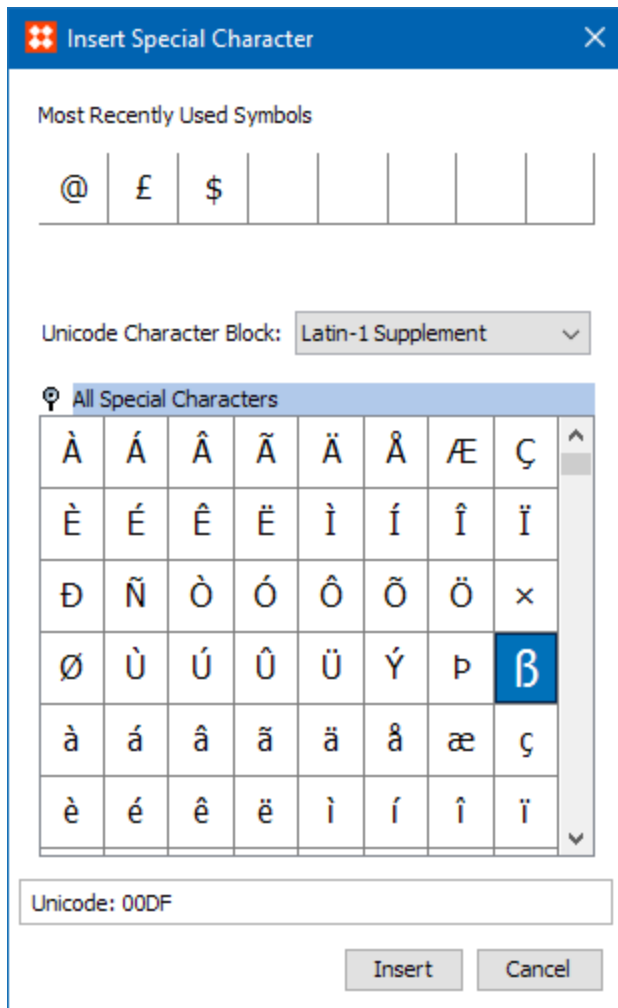
5. Click **OK**. The tag is created, and is selectable in the hierarchy.



Once you have created tags, proper configuration and rendering of the tags needs to take place. See **Configuring / Rendering Tags** in this documentation.

Considerations for Specific Tag types

- **Special Character Tags:** When creating special characters, an 'Insert Special Character' dialog will appear. Choose a Unicode Character Block from the dropdown, and select a special character. Click 'Insert' when finished.



- **Character Tag:** It is best practice to set up tag entries for <lt/> , less-than, and <gt/> , greater-than, characters if they are not already set up. As is the case with all character tags, the character(s) entered in the Rendering area will display in the workbench GUI with a gray background. Example: ➤

Once they are set up, for these two specific tags *only* can a user enter the < or > directly on the keyboard. STEP will take this single keystroke and immediately convert that single keystroke to the STEP tag <lt/> or <gt/> and it will show in the workbench as that single character (background of gray). However, this is only useful for those cases where a user wishes to display the characters of greater than or less than as a symbol within a attribute values. Example >40mg

In order to use the greater than, >, and less than, <, characters, tags must be set up for them using <gt/> and <lt/>. As is the case with all Character Tags, the character(s) entered in the Rendering area will display in the workbench GUI with a gray background.

The ampersand character, &, and the double straight quote character, ", are also considered 'automatic' when converting to on output. Again, it is an accepted industry standard to use the entities of & and " when these characters occur in a text stream in XML. However do not enter these entities of & etc. directly into

STEP attribute values in the workbench or Web UI. Entering these in will not give the desired results. You should enter the & and " directly from the keyboard.

- **Style Tag:** 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 be converted to 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 (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 (this tag will surround the entire bullet list).
- Click 'Apply' when finished.

Rendering

Style

Font: Inherit Size: Inherit

Text Color: Inherit

Background Color: Inherit

On Inherit Off

Bold ☐ ☒ ☐

Italic ☐ ☒ ☐

Underline ☐ ☒ ☐

Subscript ☐ ☒ ☐

Superscript ☐ ☒ ☐

☒ Paragraph Style

Prefix: • List Tag: bulletlist

Preview: The Quick Brown Fox Jumps Over The Lazy Dog

Apply Cancel

Using this setup supports only a one level list, as STEP does not support any further levels, so you cannot produce multiple level lists on HTML output.

This setup for creating bullets is a unique accommodation in STEP to allow for lists to be produced correctly for HTML output. Since InDesign has no such stringent requirements, the special handling is just ignored.

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.

- **Hyperlink Tags:** When creating hyperlink tags, the setup depends greatly upon the user's needs. A user may find it useful to create a hyperlink for each of the five format types of hyperlinks: FILE, MAILTO, FTP, HTTP, and HTTPS. This way the hyperlinks are all able to be rendered differently on-screen to help with visual recognition, as STEP does not have a 'check' to make sure that the proper type of hyperlink was applied.

Configuring / Rendering Tags

It is possible to configure and render tags differently so that users are more visually able to tell the difference between the different tag types used while in STEP. Because configuring tags are generally the same, a hyperlink is used as an example for how to properly configure tags.

On the Hyperlink Tag tab, under the Special Character Definition flipper, modify the relevant fields:

<

hyperlink - Hyperlink Tag

→|

Hyperlink Tag		
Special Character Definition		
Name	>	Value
Name		hyperlink
Edited By		
Tag		hyperlink
Rendering		The Quick Brown Fox Jumps Over The Lazy Dog
Show Style		Yes
Keyboard Shortcut		
Output Formatting		
Format	Formatting Start	Formatting Stop
Add Formatting		

Name

The name field displays the name of the tag. The name can have a maximum of 80 characters. Spaces between words are allowed.

Tag

In the Tag field you can enter only one word. Spaces and the special characters: Less than, '<', Greater than, '>', and the forward slash '/' are not allowed in the tag field. The tag will be used for example, in a document when importing files using STEP tags. Because it can get confusing if the name field, which is what is displayed in the UI, is different than the tag, it is recommended to keep both the name and the tag itself the same.

Rendering

In the Rendering field, click the ellipses [...] to open the Rendering dialog. The settings in this dialog define how a tag is displayed in the user interface in STEP. When creating a tag, it is best practice to use some kind of rendering of the tag. This way when it is viewed in the UI, it is easier to tell which part of the attribute value is text and which part is the tag.

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
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'

Style	Description
Subscript	Select this to render the style tag 'subscript'
Superscript	Select this to render the style tag 'superscript'

Show Character Tag

In the Show Style field, specify whether or not the tag is shown in the drop down list when a user is editing an attribute value. Select 'Yes' to make the tag available and 'No' when you do not want the tag to be present in the list.

Keyboard Shortcut

In the Keyboard Shortcut field, specify a keyboard shortcut that you wish to use to insert the tag.

After the tag has been rendered properly, the output needs to be formatted. See **Configuring Output Formatting** in this documentation for more information.

Special Consideration When Rendering Tags:

- A **Character tag** is rendered by entering in a character directly into the field, there are no ellipsis buttons to select. If a **Character** tag does not have a rendering specified, the tag is rendered 'Undefined'.

The screenshot shows the 'System Setup' interface. On the left, a tree view lists 'Character Tags' including 'BeforeLastDigitInNumerator', 'CharacterTag', 'EndFraction', 'FractionBar', 'ft', 'gt', 'in', 'IntegerToFractionSeparator', and 'LineBreak'. The 'CharacterTag' is selected. On the right, the 'Character Tag Definition' table is displayed with the following rows:

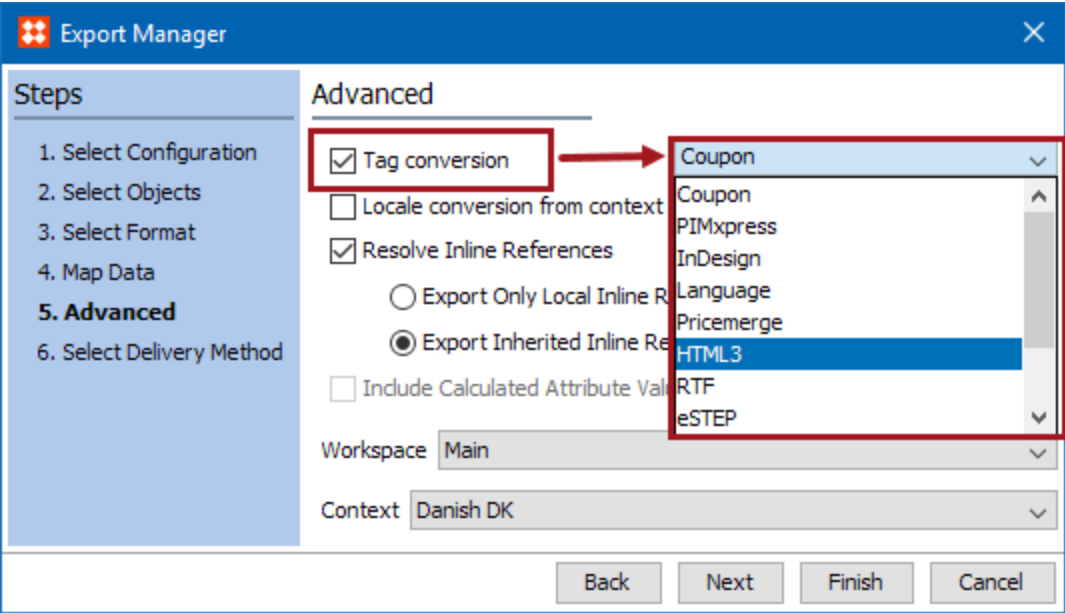
Name	Value
Name	CharacterTag
Edited By	
Character Tag	CharacterTag
Rendering	UNDEFINED
Show Character Tag	Yes
Keyboard Shortcut	

This means that each time the tag is used in Rich Text Editors, then the rendering [Undefined] is shown.

- A **Style tag** is how bullet lists are created. If a bullet list for HTML output is needed, please see the **Tags Examples** topic in this documentation.

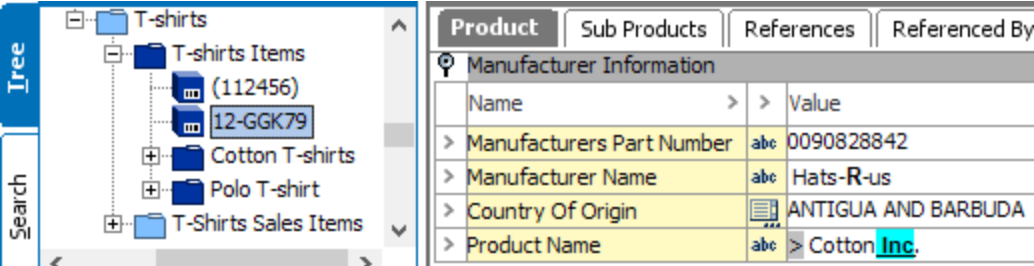
Configuring Output Formatting

Because all STEP tags themselves are media neutral, a conversion of the STEP tag can be set up to accommodate the requirements of downstream systems. This allows the tags to be processed by downstream systems when they are exported, provided that in the Export Manager, or Outbound Integration Endpoint Wizard, the **Tag Conversion** is checked and the appropriate tag output format is selected from the dropdown.



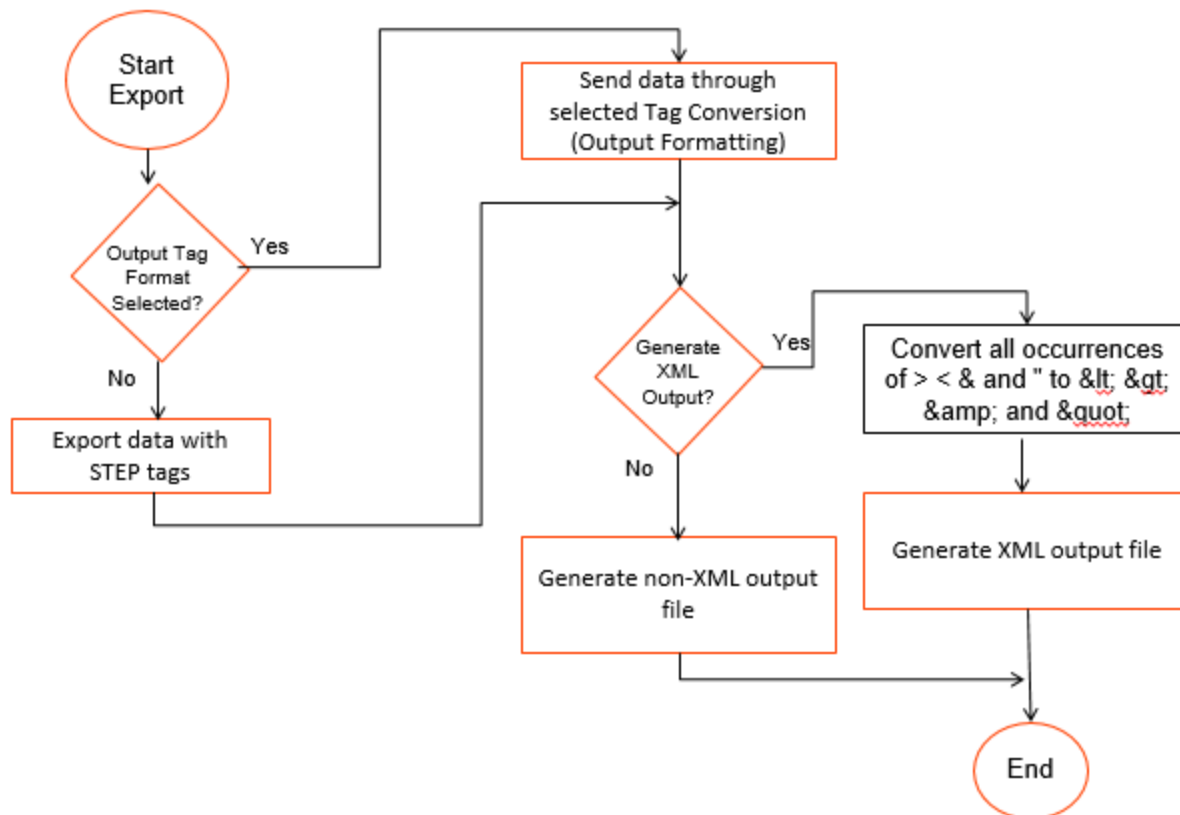
For more on exporting, see the **Export Manager** and the **Outbound Integration Endpoints** topics in the **Data Exchange** documentation.

The picture and chart below shows an example of the difference if the tag conversion is checked or not checked when exported to excel using HTML3 formatting in this case.



Tag Conversion is:	Manufacturer Name	Product Name
Checked	Hats-R-us	> Cotton Inc.
Unchecked	Hats-<Bold_C>R</Bold_C>-us	<gt/> Cotton<FILE target="http://google.com"> Inc</FILE>.

Note that when outputting XML files the conversion of < and > characters etc. comes after it has gone through tag conversion. So for XML output you will never be able to force a < or a > through the system in any way (unless a custom solution is in effect).



The same goes for importing data into STEP. If the data being imported into STEP does not have valid STEP tags, the data will not import correctly. This includes if the data has some tags that are valid in STEP and other tags that are not. If there is one tag in the attribute value that is being imported into STEP that is not valid, then the whole attribute value will not be imported properly.

Note: Using HTML names for special characters, such as ™ or ©, is not supported with four exceptions. The five exceptions are >, <, &, and ". See the chart below for more details.

The XML specification allows for HTML Names to be used (usually to denote characters such as ™ or © -- trademark and copyright symbols). STEP supports only four HTML Names withing an XML file. They are as follows:

Name	Symbol	Predefined Entities / Character Entity Reference	Unicode number
double quotation mark	"	"	Unicode 0022
ampersand	&	&	Unicode 0026
less-than sign	<	<	Unicode 003C
greater-than sign	>	>	Unicode 003E

Although STEP does not support HTML names other than those listed above, it does support HTML codes on importing data files that use the format &#nnnn; or &#xhhhh; where *nnnn* is the code point in decimal form, and *hhhh* is the code point in hexadecimal form. The 'x' must be lowercase in XML documents. The *nnnn* or *hhhh* may be any number of digits and may include leading zeros. The *hhhh* may mix uppercase and lowercase, though uppercase is the usual style.. But using characters in the ranges &#0; to &#31; and &#127; to Ÿ should be avoided since they are not Unicode characters.

Output Formatting Flipper

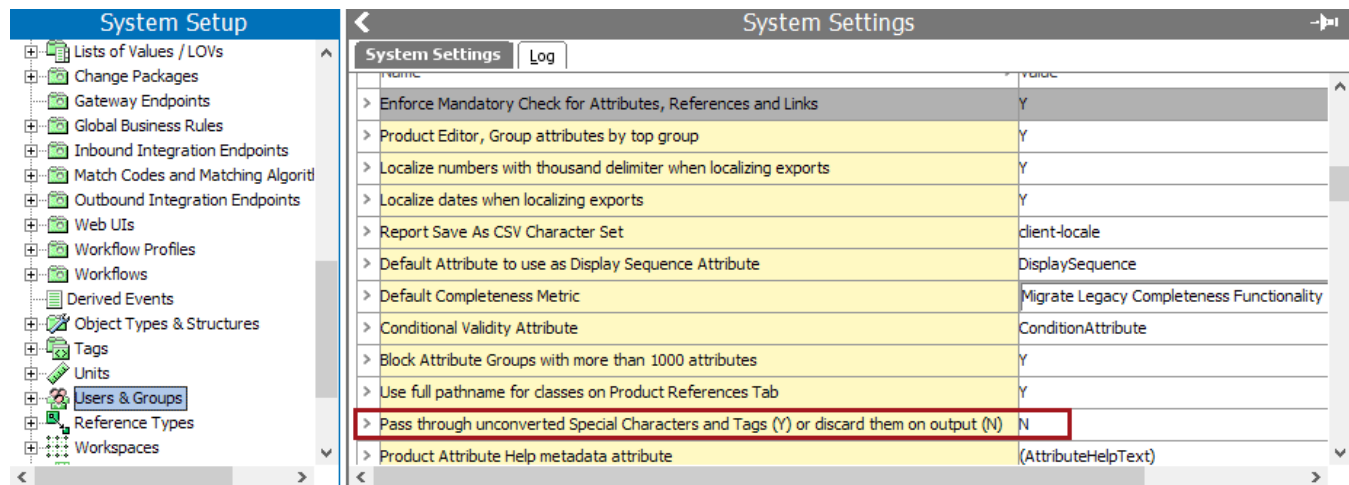
Multiple output formats can be added to each tag to accommodate the requirements of the different systems being exported to. With valid STEP tags formatted, data coming into STEP will import correctly as well. The steps below apply to all STEP tags when formatting.

1. Choose an output format from the dropdown menu. Note that the output format is really just a name. No predefined set of codes is actually being chosen here. It is just a name to allow the user to set up one or more different tag conversions to meet the requirements of different downstream systems or applications. For a web application the output format name of 'HTML3' is usually selected, and output formatting is set up to convert all the STEP media neutral tags to appropriate HTML tags.

Be sure that the 'Pass through' setting under the Users and Groups node is set up accordingly to the desired output for tags.

'Pass Through' Setting

Once all output formatting is set up, it is also important to check the setting in System Setup in the Users and Groups node, under the 'Enforce Mandatory Check for Attributes, References and Links' flipper called, **Pass through unconverted Special Characters and Tags (Y) or discard them on output (N)**.



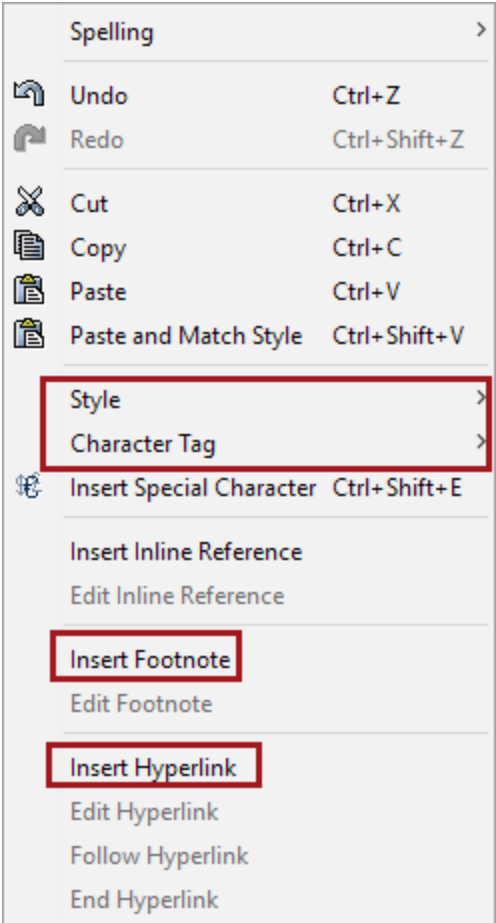
If it is set to N, then any style tags, character tags or special characters that do not have an entry in the output formatting area for the format selected will all be stripped out on output.

If it is set to Y, then any style tags, character tags or special characters that do not have an entry in the output formatting area for the format selected will all be output as STEP Tags.

See the **Product Information Manager Default Settings** in the **Super User / System Setup** documentation for more information.

Tag Examples

How to implement a tag is similar for all tags. A user highlights and right-clicks on the text in valid attribute, and selects the appropriate tag.



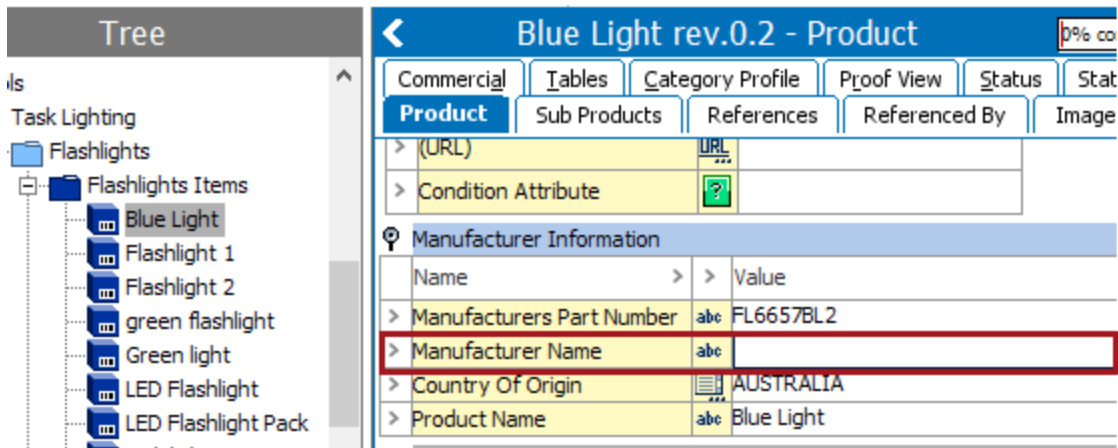
Note: Footnotes are used to add an additional piece of information to attribute values when the attributes are used in STEP Tables only. If footnotes are added to attribute values outside of a STEP Table, they will not output expected result.

Below is an detailed example of how to implement a tag using a hyperlink.

Inserting a Tag

When a user wishes to implement a tag, they can do so by following the steps below.

1. Navigate to the desired object in STEP, and click into the attribute with the value base as text.



Tree

- Task Lighting
 - Flashlights
 - Flashlights Items
 - Blue Light
 - Flashlight 1
 - Flashlight 2
 - green flashlight
 - Green light
 - LED Flashlight
 - LED Flashlight Pack

Blue Light rev.0.2 - Product 0% complete

Commercial Tables Category Profile Proof View Status Stat

Product Sub Products References Referenced By Image

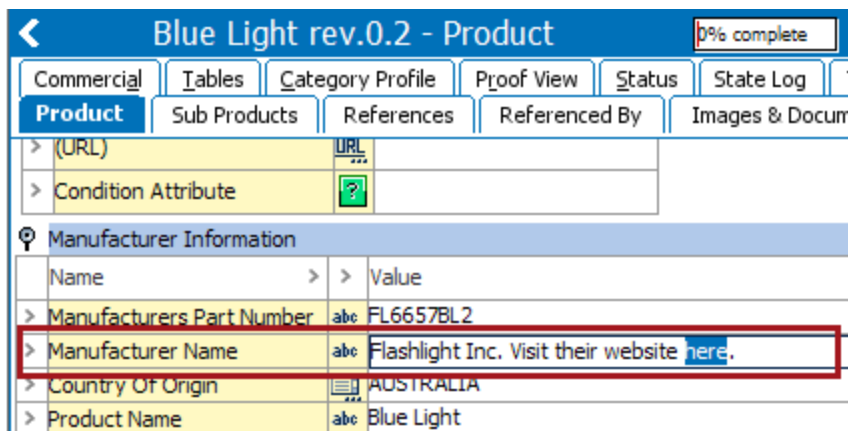
> (URL) URL

> Condition Attribute ?

Manufacturer Information

Name	Value
> Manufacturers Part Number	abc FL6657BL2
> Manufacturer Name	abc
> Country Of Origin	ABC AUSTRALIA
> Product Name	abc Blue Light

2. Enter in any text that needs to be added in the attribute field. Then highlight section of the entered text that needs to have a tag and right-click.



Blue Light rev.0.2 - Product 0% complete

Commercial Tables Category Profile Proof View Status State Log

Product Sub Products References Referenced By Images & Docum

> (URL) URL

> Condition Attribute ?

Manufacturer Information

Name	Value
> Manufacturers Part Number	abc FL6657BL2
> Manufacturer Name	abc Flashlight Inc. Visit their website here.
> Country Of Origin	ABC AUSTRALIA
> Product Name	abc Blue Light

3. Select the desired tag option. In this example for hyperlinks, an 'Insert Hyperlink' dialog box comes up. Here next to the Hyperlink Type field, select the name of the desired hyperlink from the dropdown menu, and then type the needed URL. Click OK.



Insert Hyperlink

Hyperlink type FILE

URL http://www.flashlightinc.com

OK Cancel

Note that this URL could be pointing not just to an internet site outside of the system, but it could also point to a public folder in the user's Intranet. In this case the URL to the folder would need to be entered.

4. The hyperlink should now be in place, and show up with the rendering that was specified for that hyperlink. It is best practice to then click right after the hyperlink and select, 'End Hyperlink'. This ensures that any text added after the hyperlink does not then become part of the hyperlink.

Manufacturer Information

Name		Value
> Manufacturers Part Number	abc	FL6657BL2
> Manufacturer Name	abc	Flashlight Inc. Visit their website here
> Country Of Origin		AUSTRALIA
> Product Name	abc	Blue Light

Spelling

Cut

Ctrl+X

Copy

Ctrl+C

Paste

Ctrl+V

Paste and Match Style

Ctrl+Shift+V

Style

Character Tag

Insert Special Character

Ctrl+Shift+E

Insert Footnote

Edit Footnote

Insert Hyperlink

Edit Hyperlink

Fd

Edit a hyperlink.

End Hyperlink

5. You can check to make sure that the hyperlink works by right clicking on it and selecting **Follow Link**.

Manufacturer Information

Name		Value
> Manufacturers Part Number	abc	FL6657BL2
> Manufacturer Name	abc	Flashlight Inc. Visit their website here.
> Country Of Origin		AUSTRALIA
> Product Name	abc	Blue Light

Spelling

Cut

Ctrl+X

Copy

Ctrl+C

Paste

Ctrl+V

Paste and Match Style

Ctrl+Shift+V

Style

Character Tag

 Insert Special Character

Ctrl+Shift+E

Insert Footnote

Edit Footnote

Insert Hyperlink

Edit Hyperlink

Follow Hyperlink

End Hyperlink

6. If the link is correct, the user will be directed to the designated URL.

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 **Data Exchange** 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)

Important Notes:

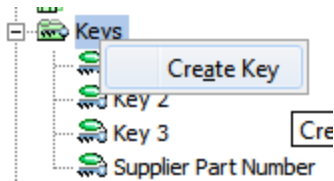
- If values for attributes used in the key definition are missing on objects, those objects are ignored and no key is generated for them.
- Attributes used in keys must have values assigned to the objects which should be identified via key - inherited values cannot be used. That is, if you have an attribute value entered at the parent level of a product, that inherited value cannot be used to create a key for a child product. The attribute value must be assigned to the child product directly.

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.

A screenshot of the 'Key 4 - Key' dialog box. The 'Key' tab is selected. The 'Definition' section shows fields for Name, ID, State, and Last edited by. The 'Used For Object Types' section shows a table with columns ID and Name. The 'Key Attributes' section shows a table with columns ID and Name. The 'Key Formula' section is empty.

Name	Value
ID	Key 4
Name	Key 4
State	Inactive
Last edited by	2015-12-02 04:56:04 by USER

ID	Name
Add Object Type	

ID	Name
Add Attribute	

Key Formula

- 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.

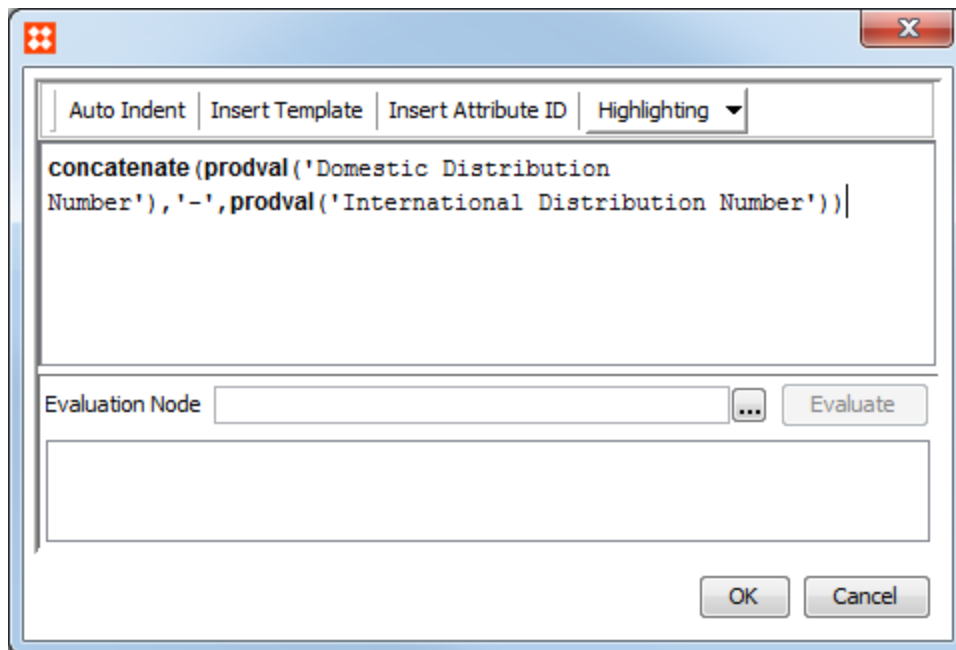
Note: The key functionality does not work with product overrides.

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.

For more information on deactivating keys, see the **Activating and Deactivating Keys** section of the **System Setup / Super User 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** section of the **System Setup / Super User Guide** documentation.

To Delete a Key

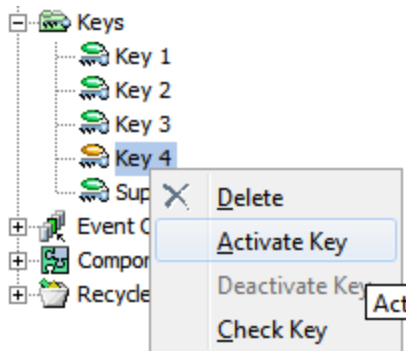
In **System Setup**, expand **Keys**, right-click the relevant key, and then choose **Delete Key**.

Note: A key must be deactivated before it can be deleted.

Activating and Deactivating Keys

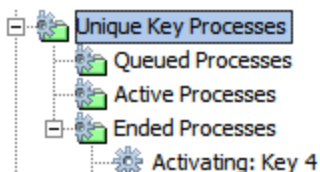
Activate a Key

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.

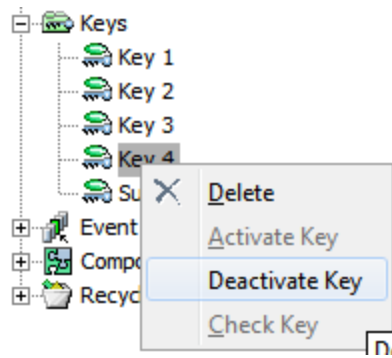


Keys are not generated for objects with missing key attribute values. Once values are supplied for all key attributes on a product, keys are generated on the fly.

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**.

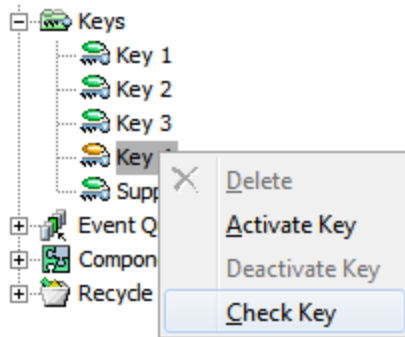


2. Make the preferred changes, and then activate the key again.

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.

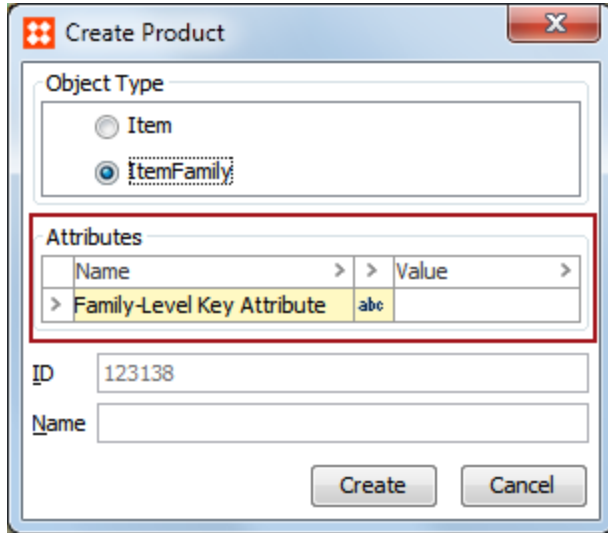
If identical keys are generated for two or more products, this is reported as errors in the Execution report. The generated key and the conflicted products are listed in the report.

- 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.



Create Product

Object Type

☐ Item

☒ ItemFamily

Attributes

Name	Value
> Family-Level Key Attribute	abc

ID: 123138

Name:

Create Cancel

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">

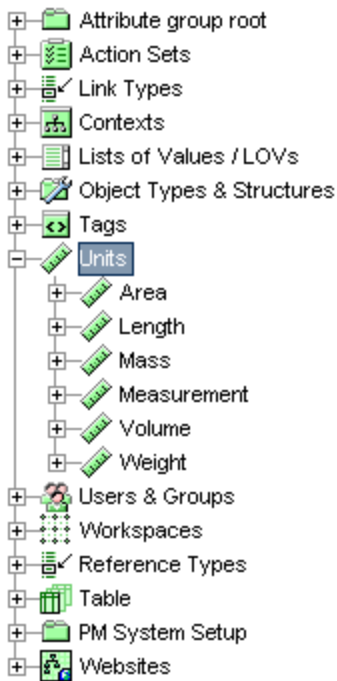
  <Products>
    <Product UserTypeID="KeyBProdType" ParentID="UniqueKeyImport">
      <KeyValue KeyID="KeyB">ValAvalBvalC</KeyValue>
      <Name>keyprodXMLImpChanged</Name>
      <Values>
        <Value AttributeID="UniqueKeyImportAttC">valC</Value>
        <Value AttributeID="ImportAtt">valA</Value>
        <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 554.

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.

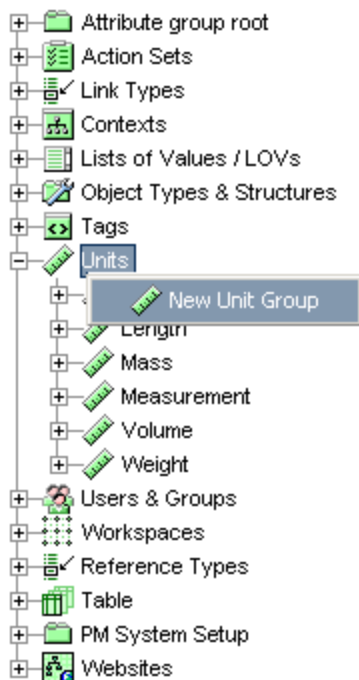
Note: The conversion factor and conversion offset should never be empty. In case one of them is “0” (zero), “0.0” has to be typed.

Maintaining Unit Groups

Creating a Unit Group

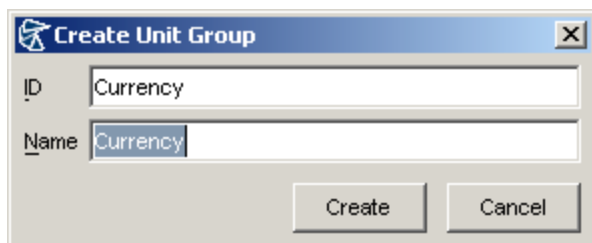
Unit Groups are created in [System Setup]: **Units**.

1. In [System Setup], click **Units**.

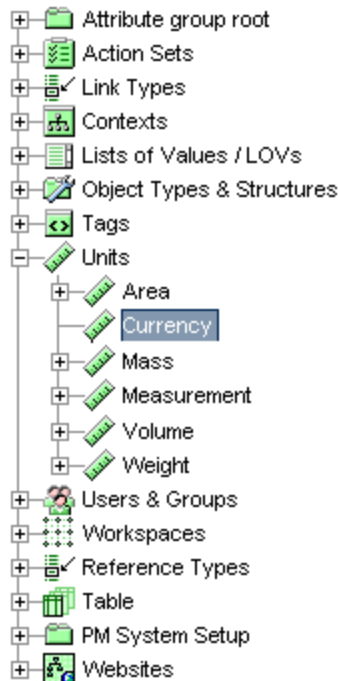


2. In the **Object** menu, point to **Create** and then click **Unit Group**, or right-click and click **New Unit Group**.

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

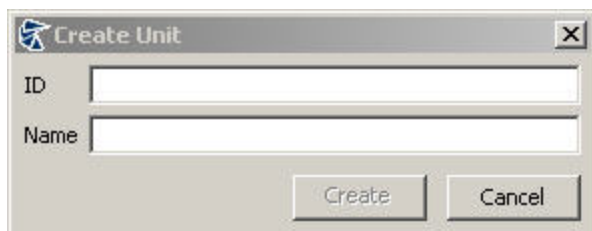
Units are created in [System Setup]: **Units**.

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.

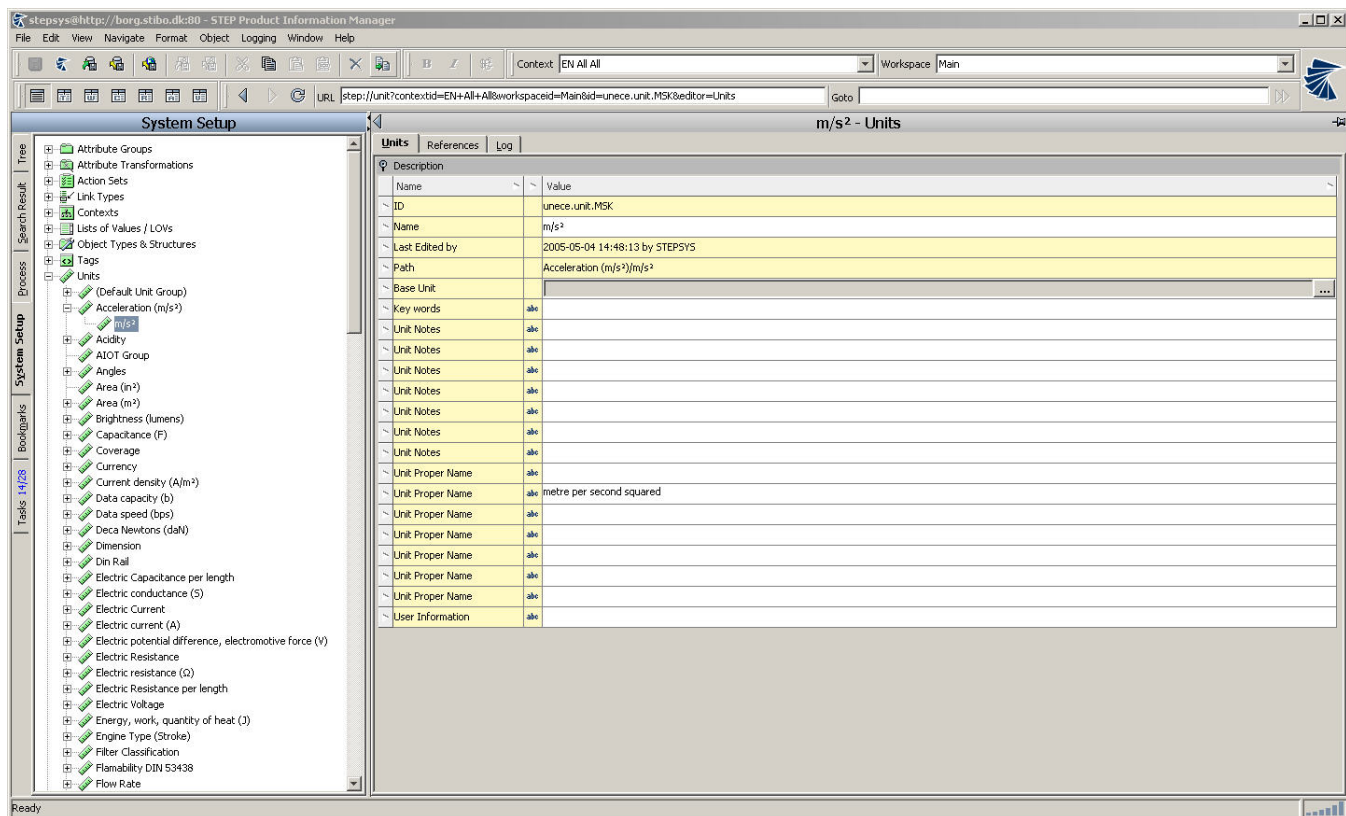
Setting up a Unit Conversion

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.

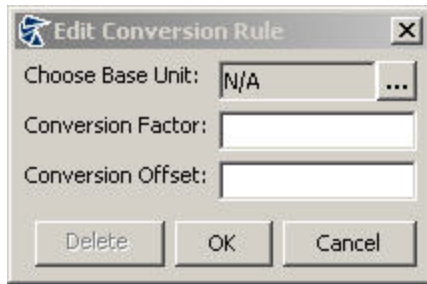
1. In [System Setup], unfold **Units**, unfold the relevant Unit Group, and then click a Unit.

The **Unit Editor** appears.



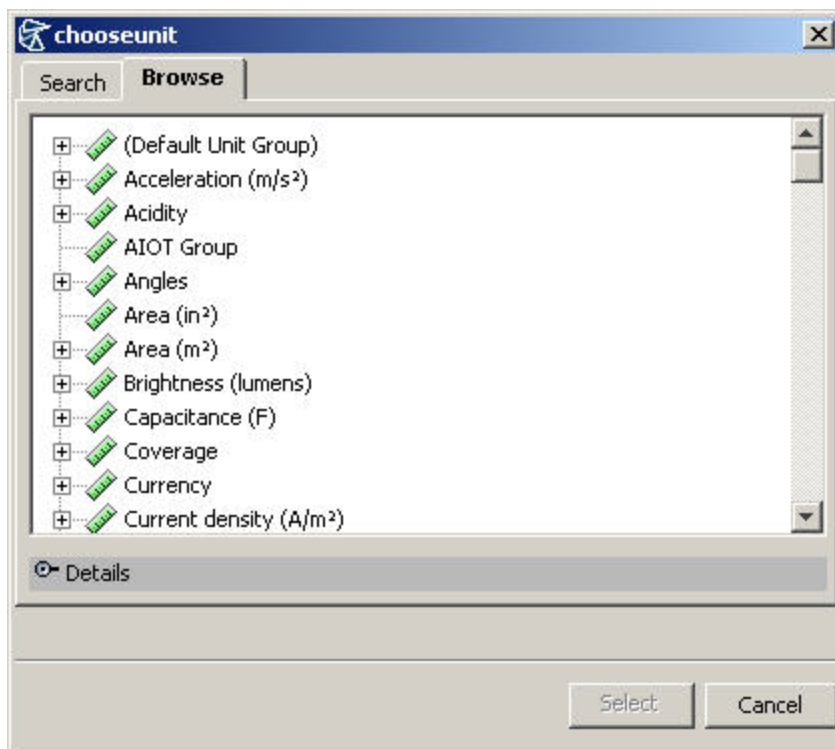
2. Click the **Units** tab.
3. In the **Base Unit** field, click the **[...]** button.

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 554
7. In the **Conversion Offset** field, type the offset. (, see ["Units"](#) on page 554
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

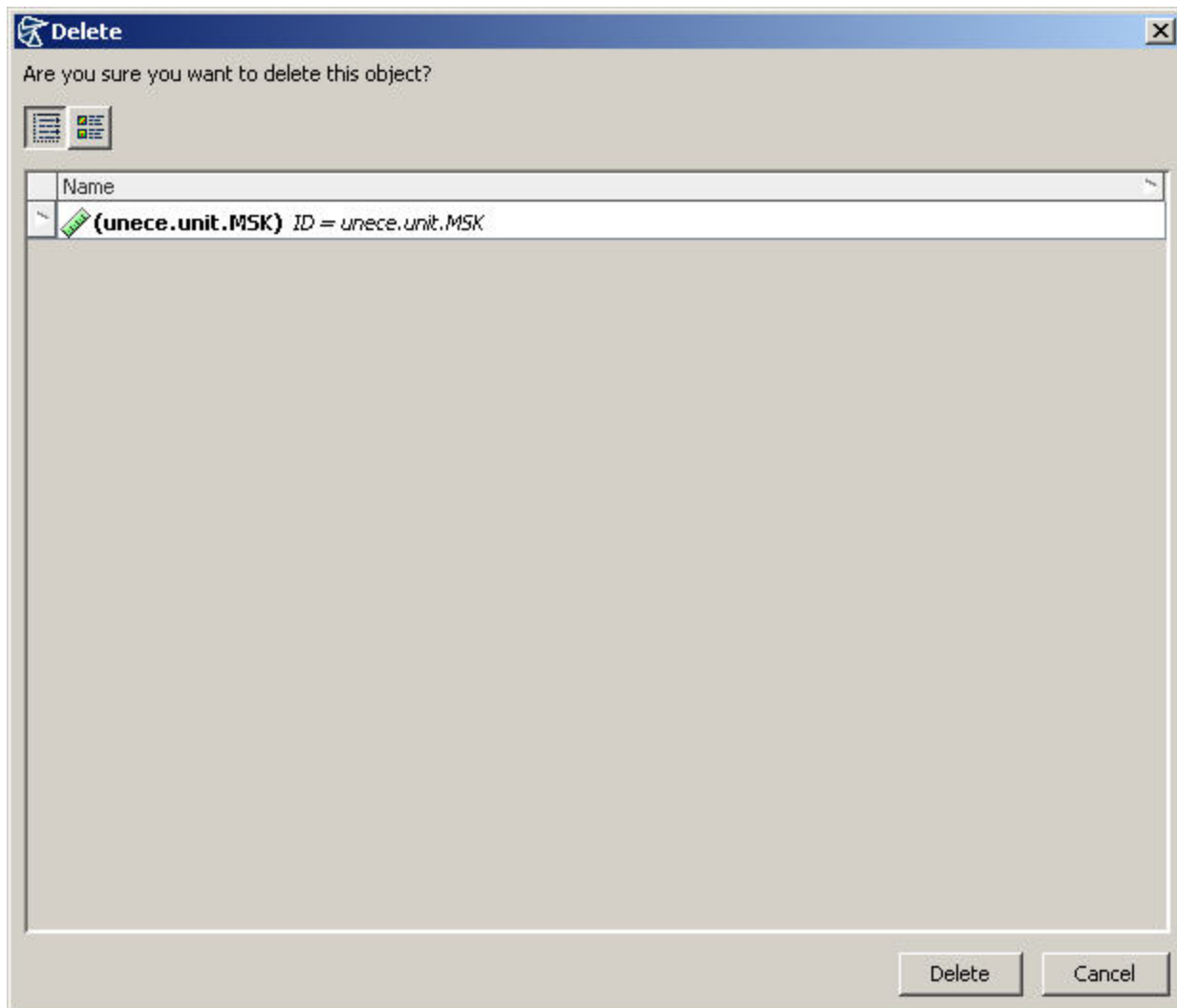
Units are deleted in [System Setup]: **Units**.

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.

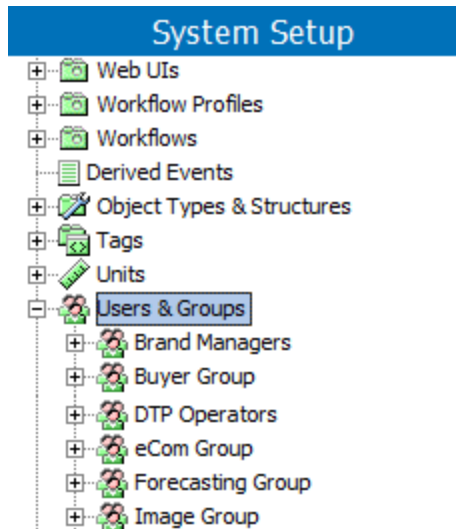


3. Click **OK** to delete.

The Unit or Unit Group will be deleted.

Users and Groups

Users and Groups are maintained from the Workbench in System Setup > **Users & Groups**.



In the Users & Groups hierarchy, you can create STEP Roles and Authorizations to determine who can access the STEP system and what they are able to do or not do as a User in the Workbench, Web UI, or DTP client. A Group containing these roles and authorizations is a set of one or more Users. An example of User Groups set-up based on roles and authorizations can be seen in the image above. All Users that are members of a specific Group share the Privilege Rules that are applied to that Group. While it is possible to assign a User to more than one Group, privileges are accumulative so this is not considered best practice as it makes it difficult to define what Privileges a User actually has.

Some of the reasons STEP Roles and Authorizations are created in Users and Groups are:

- It allows for the proper set-up of system access for Users that will be using STEP so that Users only have the rights and privileges they need.
- It gives others the ability to add Users to existing groups based on the type of access required.
- It ensures data quality and integrity by preventing accidental or malicious changes made by people who should not have access.

Note: Every User must be a member of a Group because all permissions are obtained via the Groups.

Group Membership

The members of a Group ideally perform the same or very similar roles within the database environment. 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 User Groups could be:

- 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 operator who in addition to data entry also is working with system setup and higher-level maintenance.

- Read-Only

The **Read-Only** Group will typically include operators that have an interest in viewing information but have no permissions to perform any editing operations.

Working with User Groups

This section explains how to create and delete User Groups. While always containing the ID and Name attributes, the User Group tab will look different based upon the what system settings and attributes are configured and set up on your STEP system. Information about Privilege Rules can be found in the **About Privilege Rules** section of Users and Groups. Information about GUI Set-up can be found in the **About GUI Set-up** section of Users and Groups.

Read Only - Group

Group

Privilege Rules

GUI Set-Up

Log

Description

Name	Value
ID	ReadOnly
Name	Read Only
Group Information	abc
LDAP Synchronization ID	
Disable Password Expiration	☐
Default LDAP group	☐

Users

Name	Group Information	User Information
guest		

Add User to Group

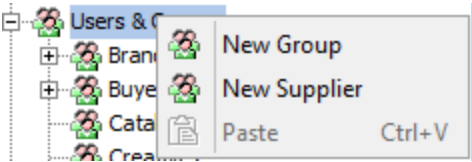
Supplier

Supplier

Creating a User Group

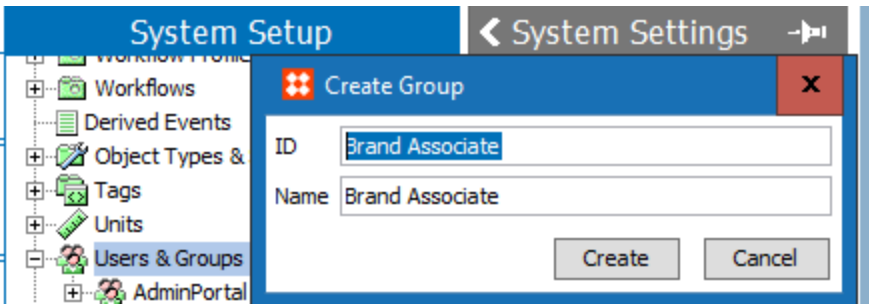
Groups are created in [System Setup]> **Users & Groups**.

User Groups can be created for Internal Users and Suppliers by right-clicking the Users and Groups root node. It is important to create and name User Groups by the function and roles the Users under that group will be performing.

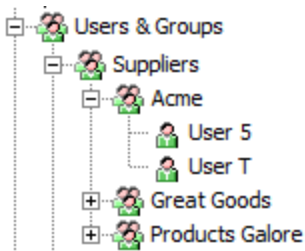


To create a New Group, right-click on the Users and Groups root not and select **New Group**. Enter the ID and Name and click 'Create'. The new group appears in the Group Editor. At this point, Privilege Rules will need to be

defined for the group and Users added. For more information about Privilege Rules, please see the **About Privilege Rules** section under Users and Groups. For more information about creating a user, please see the **Creating a User** section under Working with Users.



Creating a Supplier Group is done the same way, but it is more common to create a Supplier Group 'root node' for the supplier groups to be brought under as shown in the example below. The Privilege Rules would be applied at this Supplier Group 'root node,' and these Privilege Rules would apply to all Supplier Sub-Groups and Users residing within the hierarchy.

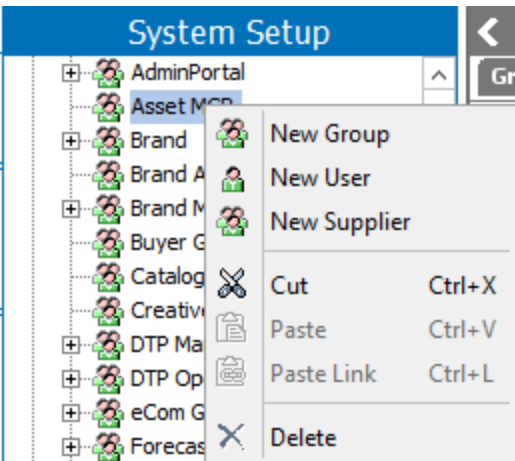


Note: If a supplier user is linked to multiple supplier groups, the supplier can see all of the supplier hierarchies that they are a member of.

Deleting a Group

If a Group needs to be deleted, all that is required is to right-click on that group and select **Delete**.

Note: All Users must be removed from the Group to be deleted in order to delete that Group.



Working with Users

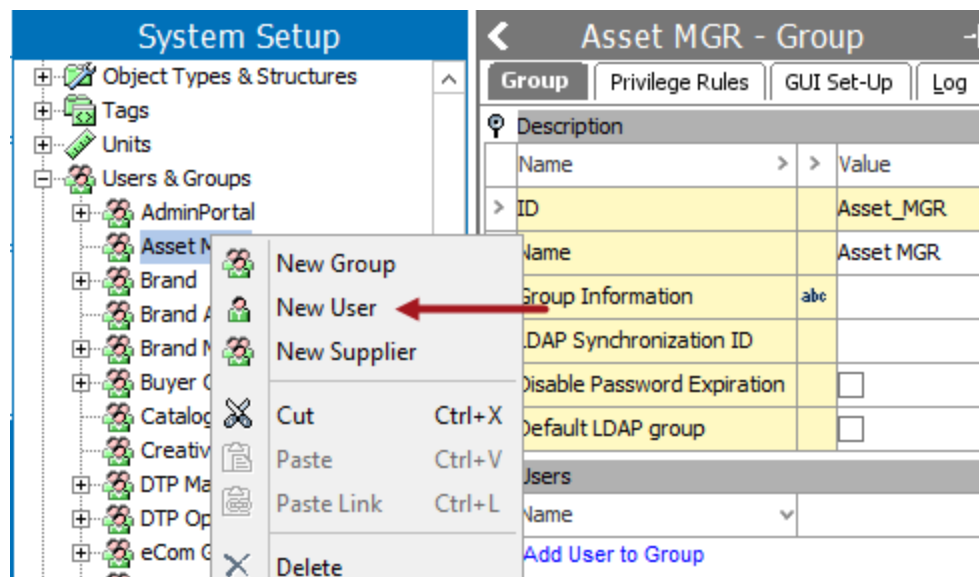
STEP can only be accessed by authenticated Users. This section contains information needed to work with Users in Users and Groups from the STEP Workbench. Information detailed in this topic include:

- Creating a user
- Adding a user to a group
- Duplicating a user
- Removing a user from a group

Information regarding how to change a user password can be found in **Changing User Password**.

Creating a User

Once User Groups and their associated Privilege Rules have been defined, Users can be created by right-clicking on the User Group and selecting **New User**.



A dialog box will appear, and the following information must be completed for the User to be created:

- **User ID** - The ID that the User will use to sign into the STEP Workbench or Web UI with. The User ID is not case sensitive, and it is recommended that special characters are not used. All User IDs must be unique as this is the ID that is used to track everything that is done in the STEP Workbench or the Web UI.
- **Name** - Enter the name to be used for the user.
- **Password** - Enter the password that the user will be using to sign on with. Passwords may have up to 24 characters and are case sensitive.
- **Retype Password** - Retype the password just entered.

Enter New User Information

X

User ID:

User Name:

Password:

Retype Password

OK

Cancel

At this point, the User has been created and additional setup specific to the user can be completed if necessary. If necessary, the User can be added to multiple groups although this is not recommended as it makes it difficult to track which privileges a User has.

User

GUI Set-Up

System Settings

Log

Description

Name	Value
ID	USERZ1
Name	Johnny Bench
E-Mail	
Force Authentication via S...	☐
Group Information	abc
User Information	abc
Change User Password	

Groups

Name	Group Information
Super Users	

[Add User to Group](#)

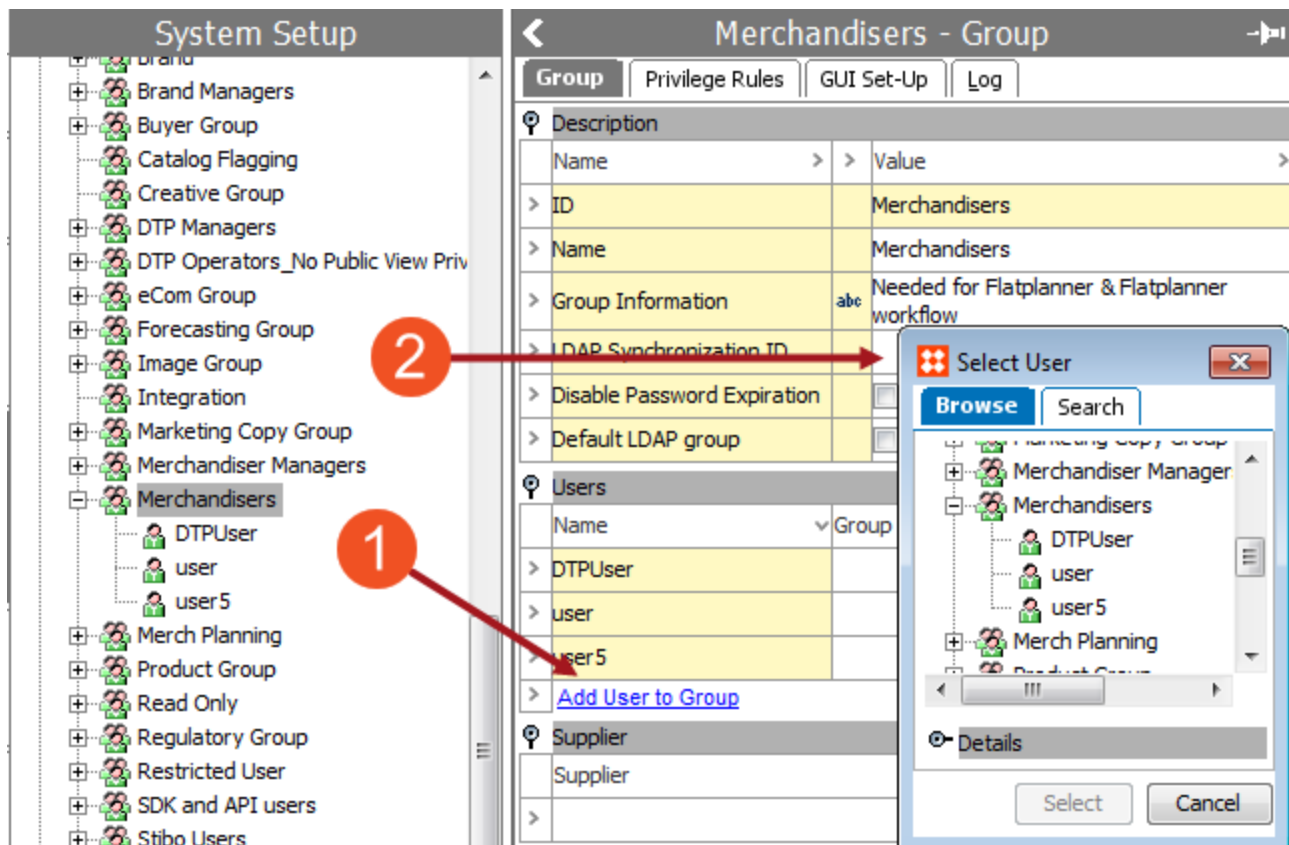
Note: If information fields should be available for Users, they must be applied as Attributes on the Object Type 'Operator user-type root', such as E-mail, Group Information, and User Information as shown in this example.

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.

- In [System Setup], unfold **Users & Groups**, and then click the relevant Group.
The Group appears in the **Group** Editor.
- Under Users, click **Add User to Group**.
A **Select User** dialog box appears.



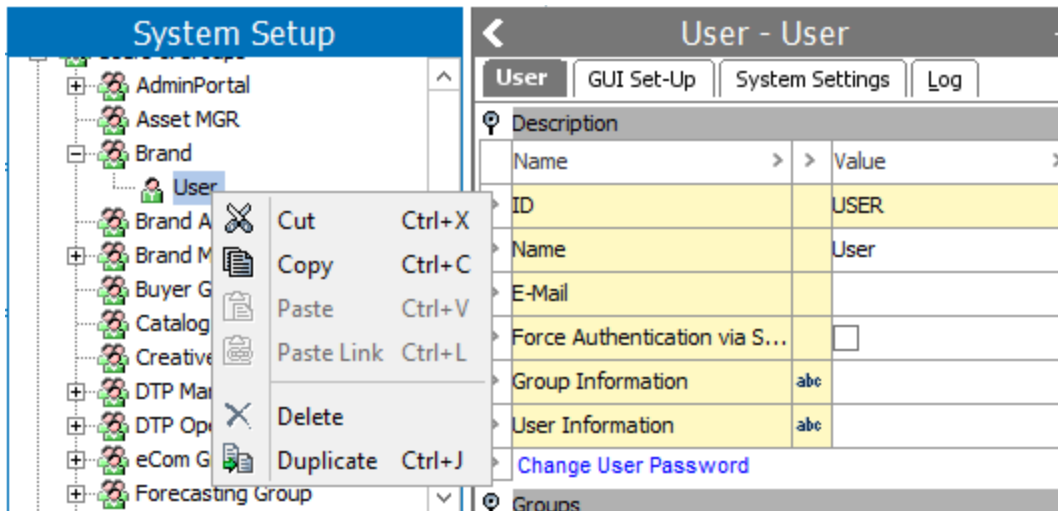
3. Select the relevant User, and then click **Select**.

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, but this is not considered best practice as it makes it difficult to define what Privileges a user actually has.

Duplicating a User

Duplicating a User can be very useful when it is necessary to add several users to the same group that need to adhere to the same privileges. For instance, if there are a hundred users that need to be added to the same group, simply create the first user following the process in **Creating a User**, right-click the user you just created, and select 'Duplicate'.

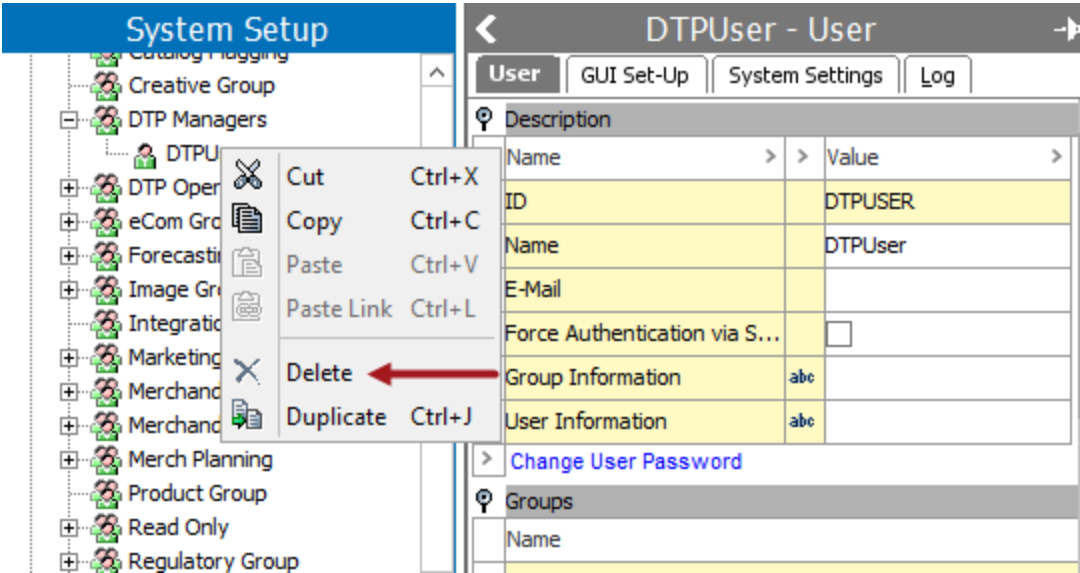


A pop-up window will appear as shown in the image below, and from here, you would follow the steps outlined in **Creating a User**. The duplicated user will have all of the same privileges, GUI Set-Up, and System Settings as the user that was duplicated.

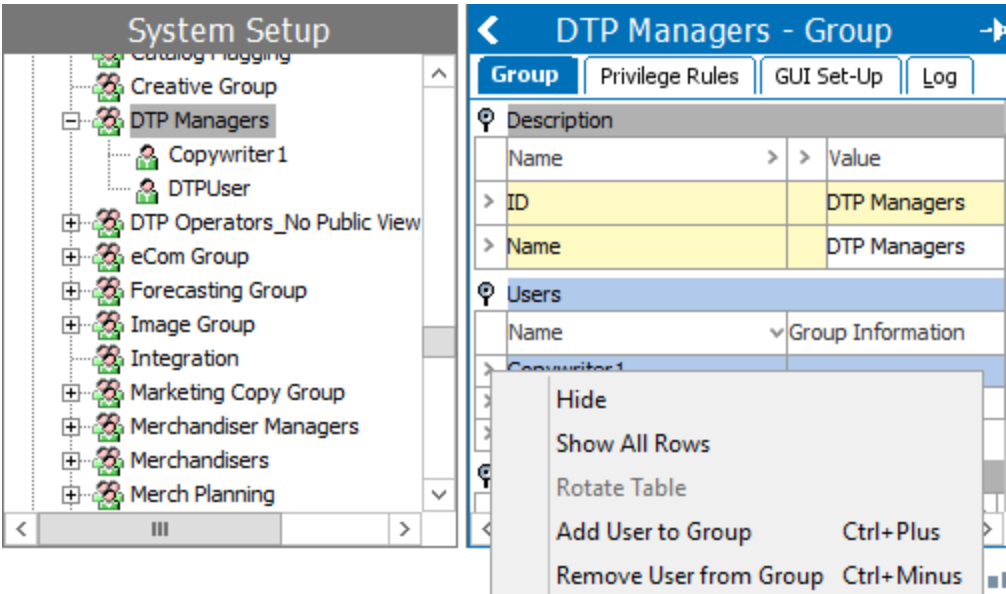
Important: Duplicating a User can be very effective, but if you are using metadata attributes on the duplicated user, the object type values on those will be duplicated as well. So for instance, if a user e-mail address value is in the user being duplicated, that value will be carried over to the new user object and will need to be changed.

Removing a User from a Group

There are two ways a User can be removed from a group. If a User is no longer permitted to have any access to STEP whether in the Workbench or in the Web UI, the User ID can simply be deleted by finding the User in the group where the User resides, right-clicking on the User, and selecting **Delete** in the pop-up window. This will delete the User from the system whether the User is a member of one group or multiple groups.



Sometimes there are circumstances where a User is a member of multiple groups but may need to be removed from one or more groups while remaining in another group or groups. In these circumstances, find the User Group you wish to remove the User from, right-click the arrow next to the User ID, and select **Remove User from Group**. If the User needs to be removed from multiple groups, this process will need to occur for each group.



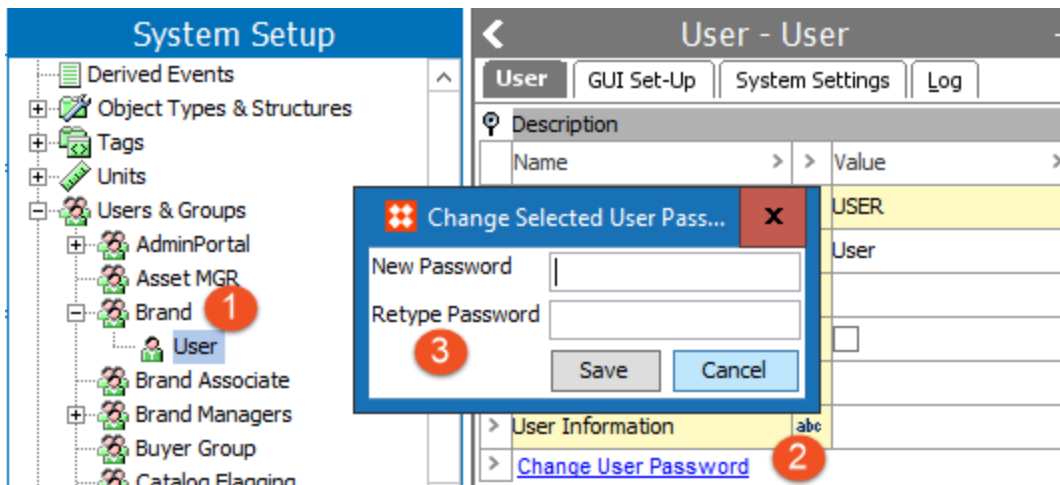
Changing a User Password

This section outlines the two most common Changing a User Password scenarios. In the first scenario, the user forgets their user password, and they need a system administrator to change it. In the second scenario, a user may want to change their own user password.

Changing a User Password - System Administrator

To change a user password as a System Administrator simply:

1. From System Setup, under Users & Groups, select the user requiring the password change.
2. Select **Change User Password** in the under the Description section of the user information.
3. Enter and retype the new password and click Save.



On a standard STEP system, passwords may contain up to 24 characters. Special characters are not allowed, such as an asterisk (*) and percentage (%), etc. Passwords are case sensitive. Only an administrative-level User should have the permissions to re-set a User's password from the Users and Groups node. Options for Users to change their own passwords are outlined below. For security reasons, the password is stored encrypted in the system so that even an administrator cannot view it.

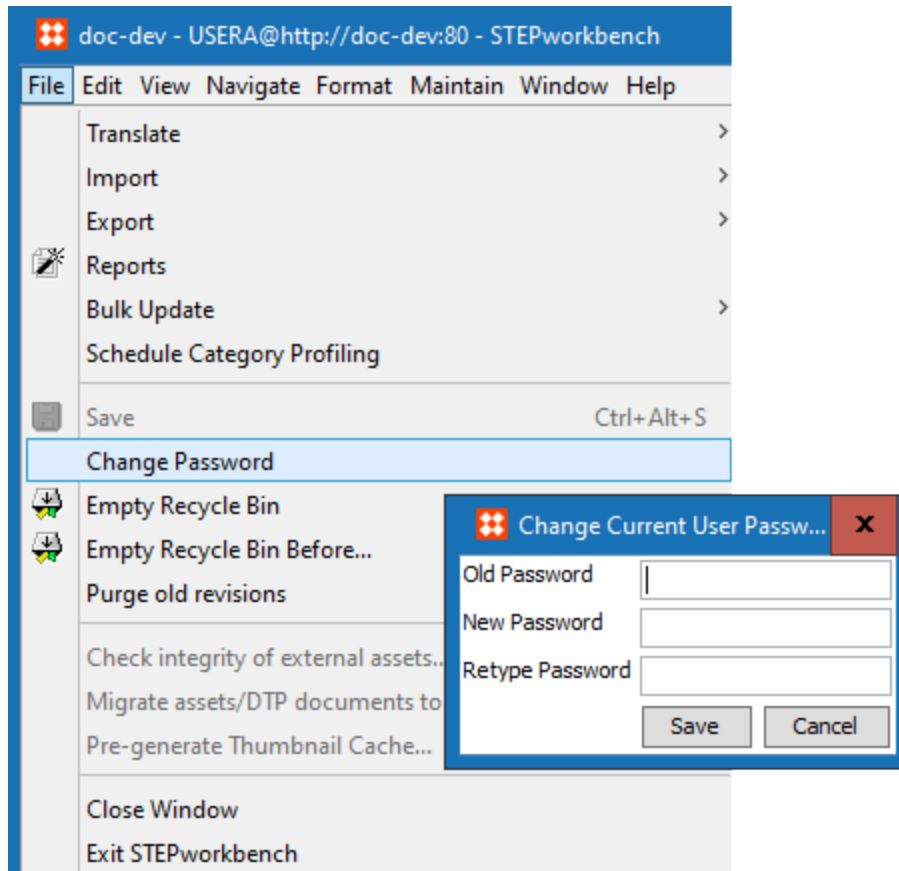
Note: On the Users and Groups root node, you can define a password security policy to be used. Information about setting security policies can be found in the **Security Policy** section of Users and Groups.

Changing a User Password - User

If privileged to do so, a User may change their own password. There are two ways the User can do this depending upon where the User is working from: the Workbench or the Web UI.

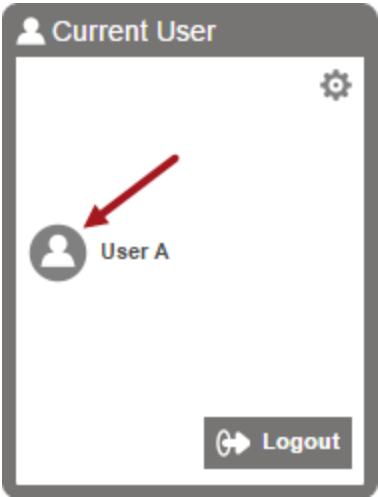
From the Workbench:

From the File menu, select **Change Password** and a **Change Current User Password** dialog box will appear. Type in the old password and then the new password twice. Click 'Save', and the new password will take effect at the next system sign-on.



From the Web UI:

Click on the User icon in the Current User widget, and the User profile will appear. Type in the old password and the new password twice and select 'Save'. The new password will take effect at the next system sign-on.



User

English US

Main

English US

User ID	USERA
Name	User A
Email Address*	
Old password	
New password	
Repeat new password	

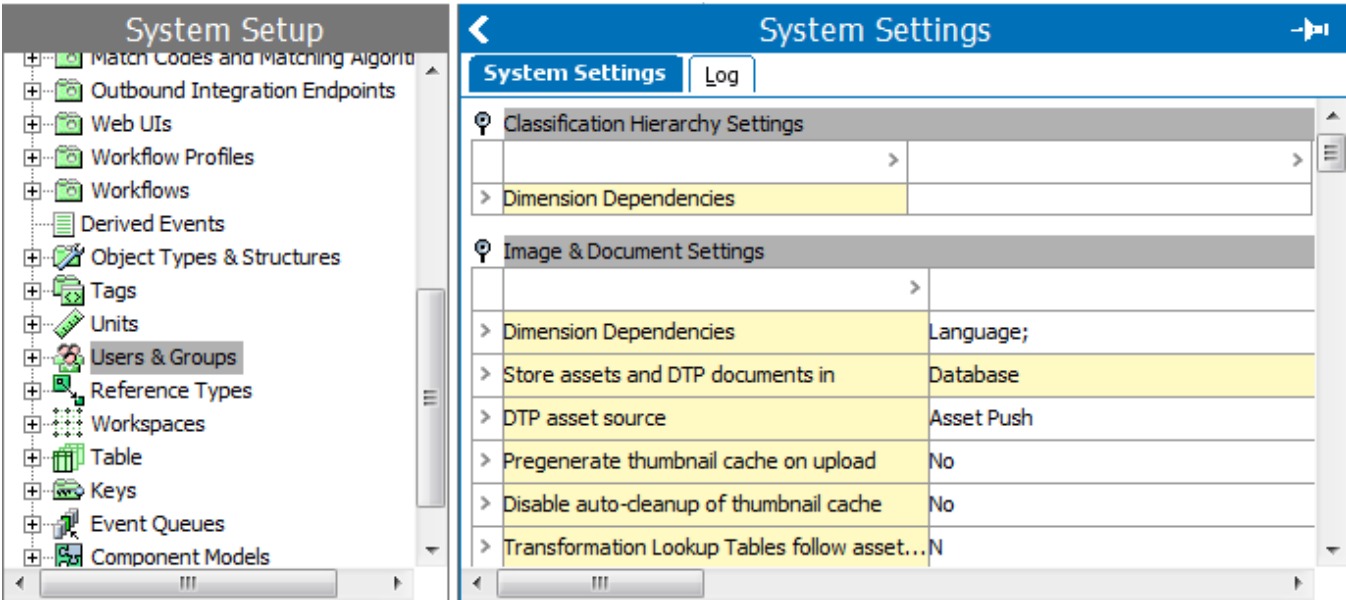
Save

Reset

System Settings

Global system settings are maintained in the **System Setup** tab, under the **Users & Groups** root node.

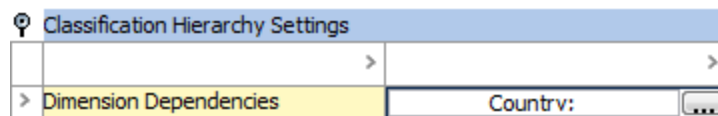
In System Setup, click **Users & Groups**. A **System Settings** editor appears. It lists a number of general and global settings on the system. Each of these settings will be described throughout this section.



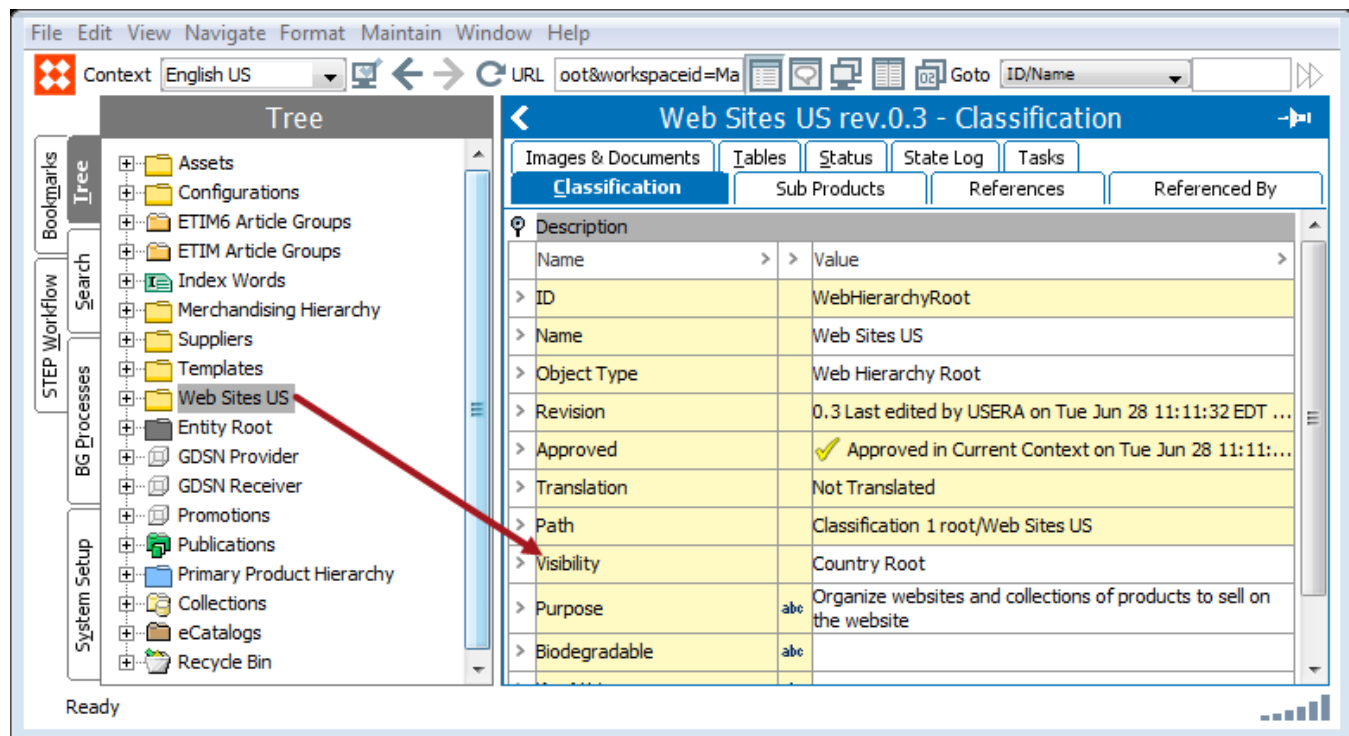
Classification Hierarchy Settings

Dimension Dependencies

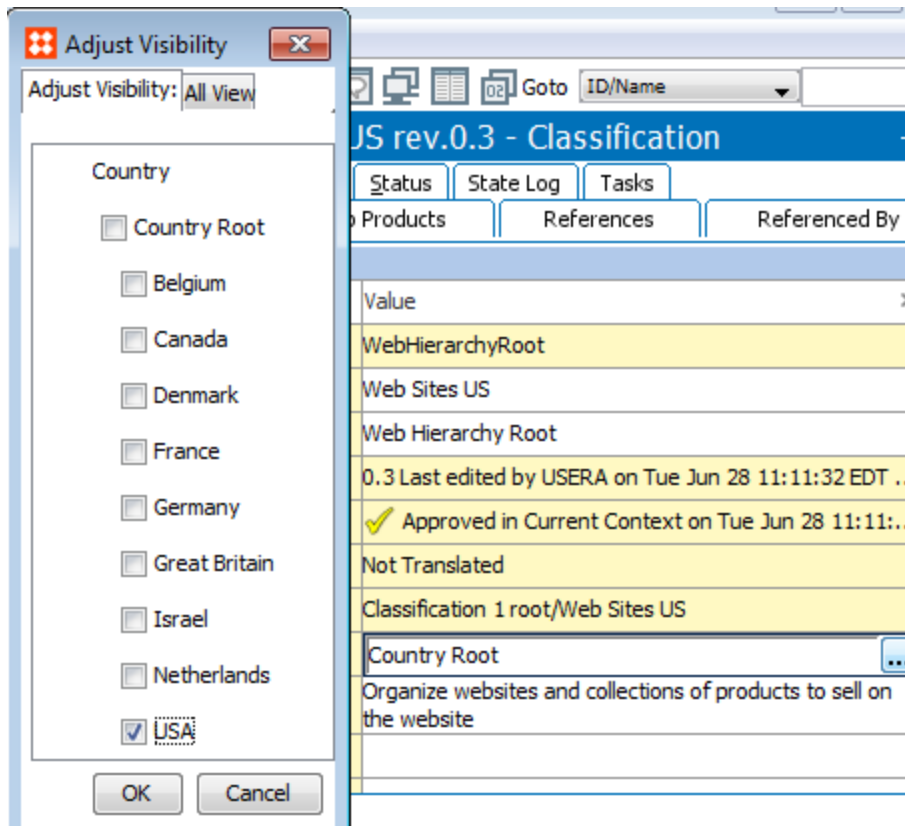
This option allows for **folders** in classification structures to be dimension-dependent. This is different than the dimension dependencies that are set on the object type. Selecting dimension dependencies here will determine whether classification structure will or will not be shown in specific languages or countries. In the example below, the dimension dependency has been set for country by clicking on the ellipses [...] button and selecting Country.



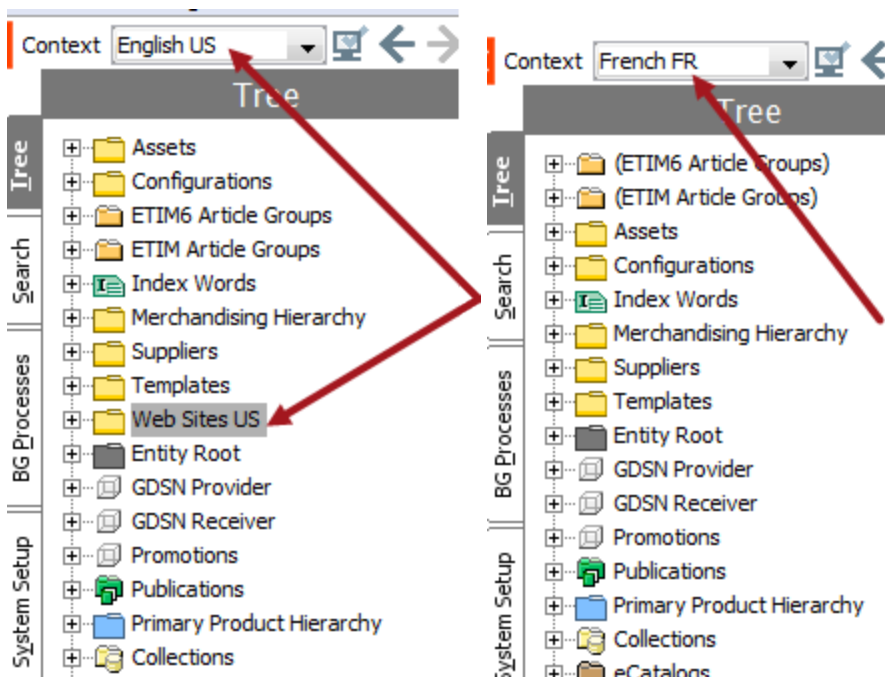
Selecting this dimension dependency added the Visibility metadata attribute to the classification objects in Tree.



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 that are dimension-dependent (e.g. presence of subfolders is dimension-dependent rather than the hierarchy itself). 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.

When sub-classifications are visible in another context, this can be seen on the References tab of the parent. For example, the Mens Tops classification has Visibility = Denmark. In the Danish DK context, we can see this classification in the hierarchy.

The screenshot shows the Stibo Systems interface with the context set to 'Danish DK'. The left sidebar shows a tree view of the classification hierarchy. The 'Mens Tops' folder is highlighted under 'Apparel' > 'Mens Casual'. The right pane shows the 'Mens Tops rev.0.1 - Classification' details. The 'References' tab is active, showing a table of references.

Classification	Sub Products	References	Referenced By	Images & Documents	Tables	Status	State
Description							
Name	>	>	Value				
ID	>	>	115215				
Name	>	>	Mens Tops				
Object Type	>	>	Web Level 3				
Revision	>	>	0.1 Last edited by USER6 on Wed Jan 25 05:54:10 EST 2017				
Approved	>	>	Never Been Approved				
Translation	>	>	Not Translated				
Path	>	>	Classification 1 root/Web Sites/Acme Retail Web Site/Apparel/Mens Casual/Mens Tops				
Visibility	>	>	Denmark				

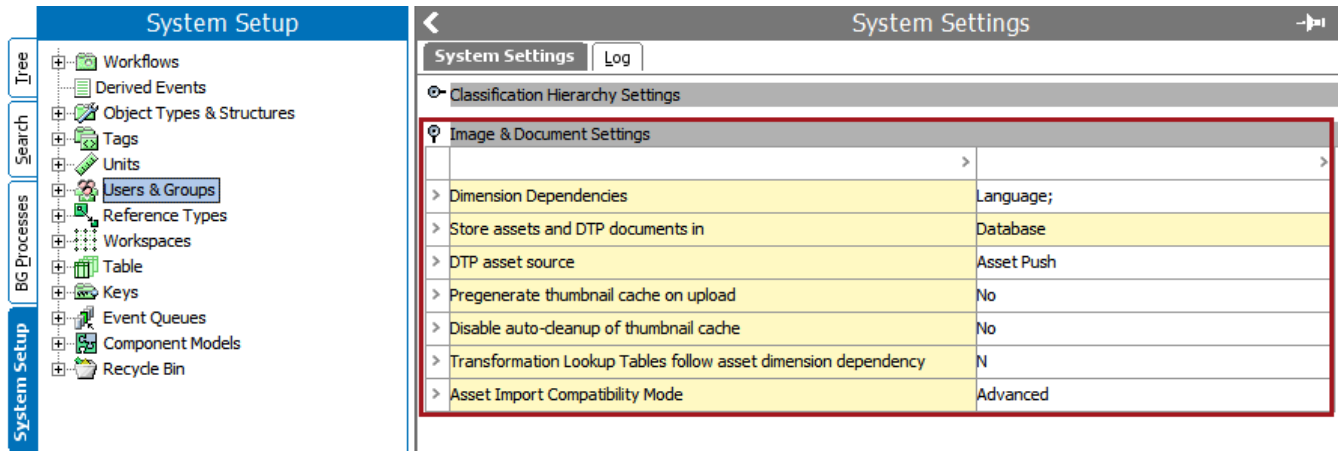
When we switch to the English US context, the Mens Tops classification is not visible in the Tree. However, if we view the References tab of the parent folder (Mens Casual), we can see the Mens Tops classification listed under the 'Visible Objects in Other Contexts' flipper.

The screenshot shows the Stibo Systems interface with the context set to 'English US'. The left sidebar shows a tree view of the classification hierarchy. The 'Mens Casual' folder is highlighted under 'Apparel'. The right pane shows the 'References' tab of the 'Mens Casual' folder, which lists the 'Mens Tops' classification under the 'Visible Objects in Other Contexts' section.

Classification	Sub Products	References
Display		
Reference Type	>	Target
Classification Reference	+	
Applied Privileges		
Applies to	>	Action Set > Attribute G...
Add Privilege		
Visible Objects in Other Contexts		
ID	>	Name
115215	>	Mens Tops
Attributes		

Image and Document Settings

This section is used to describe the various options available in the **System Setup > Users & Groups > Image & Document Settings**.

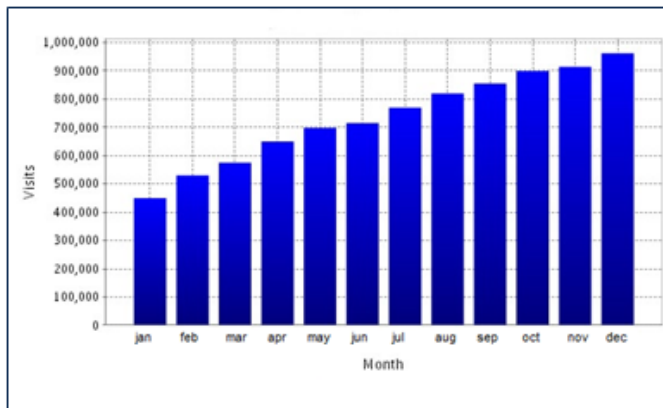


The screenshot shows the 'System Setup' interface. On the left is a navigation tree with categories: Tree, Search, BG Processes, and System Setup. Under 'System Setup', the 'Users & Groups' option is selected. The main panel displays 'System Settings' with a 'Log' button. Below this is the 'Classification Hierarchy Settings' section, which includes the 'Image & Document Settings' sub-section. This sub-section contains a table with the following settings:

Setting	Value
Dimension Dependencies	Language;
Store assets and DTP documents in	Database
DTP asset source	Asset Push
Pregenerate thumbnail cache on upload	No
Disable auto-cleanup of thumbnail cache	No
Transformation Lookup Tables follow asset dimension dependency	N
Asset Import Compatibility Mode	Advanced

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

Asset push attribute macro keys		
	>	>
> DescAtt	Description Attribute 1 (DescAtt1)	...
> assetfilename	Calculated Asset File Name (CalculatedAssetFileName)	...
> KeyAtt2	KeyAtt2 (KeyAtt2)	...
> Add attribute key		

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).

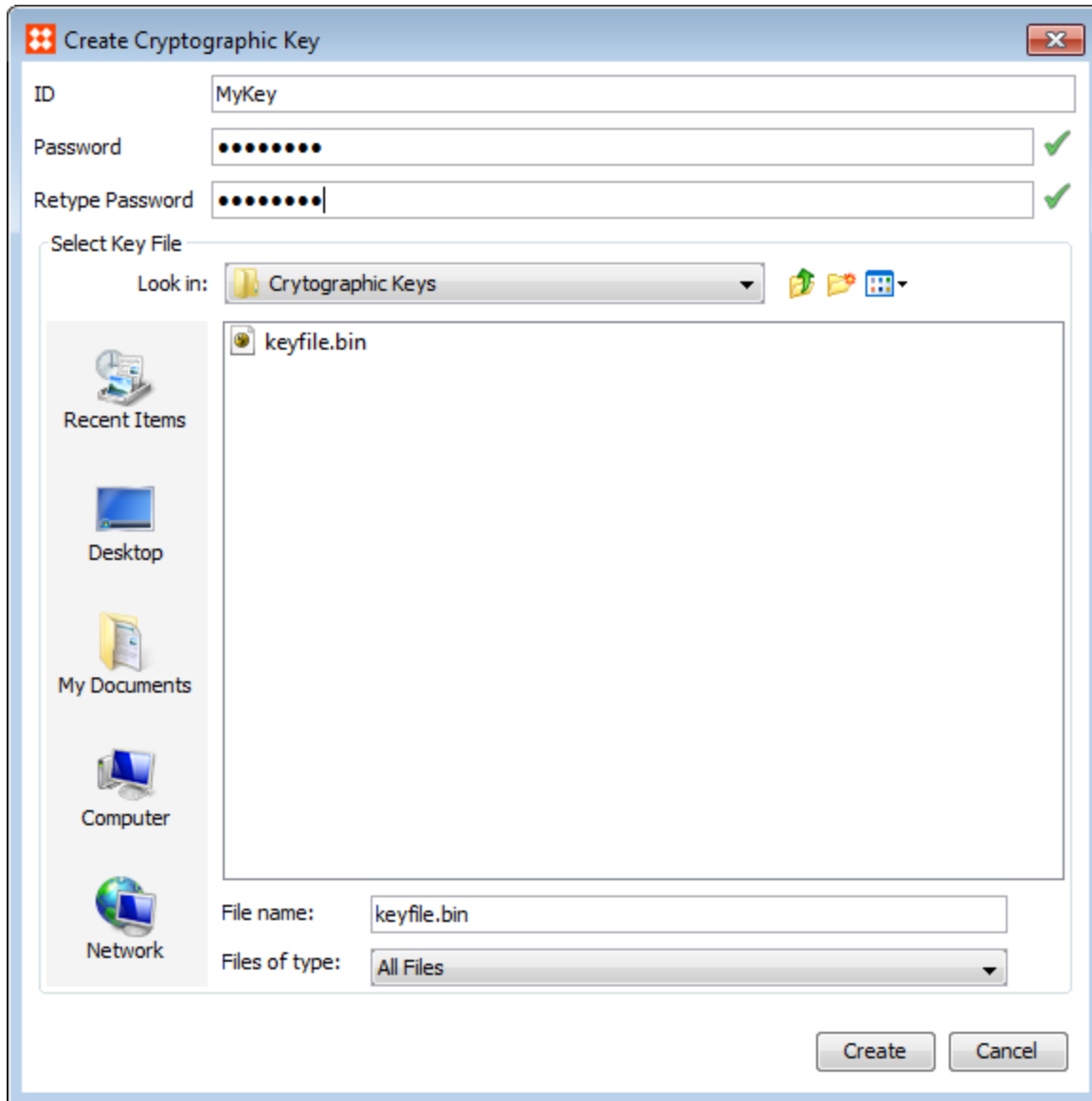
Calculated Attribute Settings	
> Dimension Dependency	Language: ...

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).

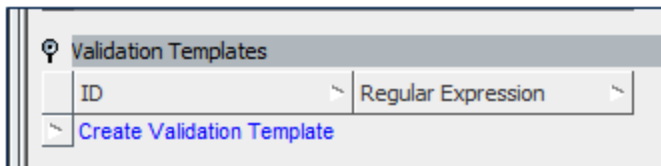
In STEP it is possible to make customized validation base types to be used on Attributes and LOVs. In this field you can create validation templates to be used when creating Attributes or LOVs.

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.

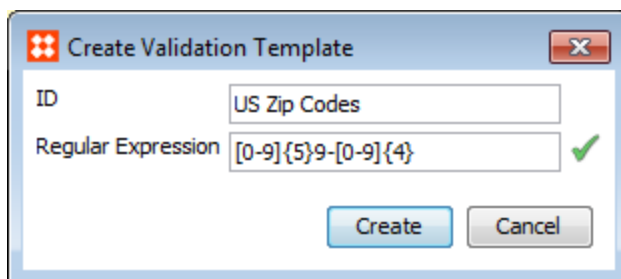
The example below shows how a new Validation Template may be created and how that new template subsequently displays in the drop-down list of validation base types when a new attribute is created (or when changing an existing attribute's validation).

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.

ID	ZipCode
Name	ZipCode
Validation Base Type	Text
Multi Valued	List Of Values
☒ Specification	Number
☐ Description	Number Range
Cryptographic Key	Numeric Text
	Regular Expression
	Text
	URL
	Template: US Zip Codes

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.

Description

Name	Value
ID	133734
Name	Car Tire Kit
Object Type	Item
Revision	0.8 Last edited by USERA on Mon Jul 11 15:39:41 EDT 2016
Approved	Last Approved on Mon Jul 11 15:38:07 EDT 2016
Translation	Not Translated
Path	Primary Product Hierarchy/Produ
UPC	abc
EAN	abc
GTIN	abc
Provider GLN	abc
(AttrCalcDesc)	
Category	Primary Product Hierarchy Prod Tire Care Tire Care Items Car Tire Kit Car Tire Kit



Unable to approve product

Approve report for [Car Tire Kit](#)

Unable to approve object due to missing mandatory values on attribute(s):

[UPC](#)

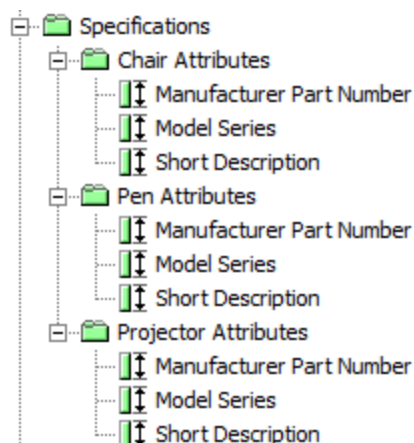
OK

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.

For 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.

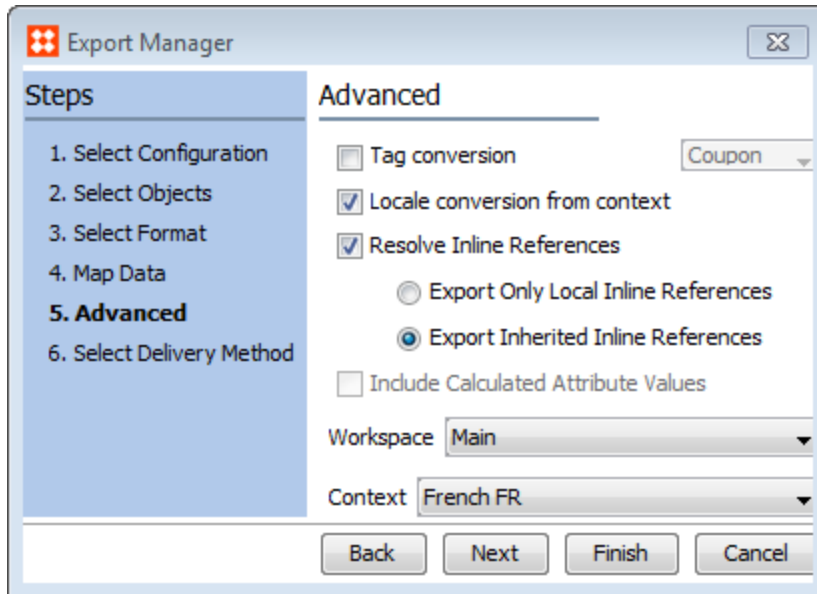
Chair Attributes		
Name	>	Value
> Manufacturer Part Number	abc	
> Model Series	abc	
> Short Description	abc	
Pen Attributes		
Name	>	Value
> Manufacturer Part Number	abc	
> Model Series	abc	
> Short Description	abc	
Projector Attributes		
Name	>	Value
> Manufacturer Part Number	abc	
> Model Series	abc	
> Short Description	abc	

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.

Specifications		
Name	>	Value
> Manufacturer Part Number	abc	
> Model Series	abc	
> Short Description	abc	

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.

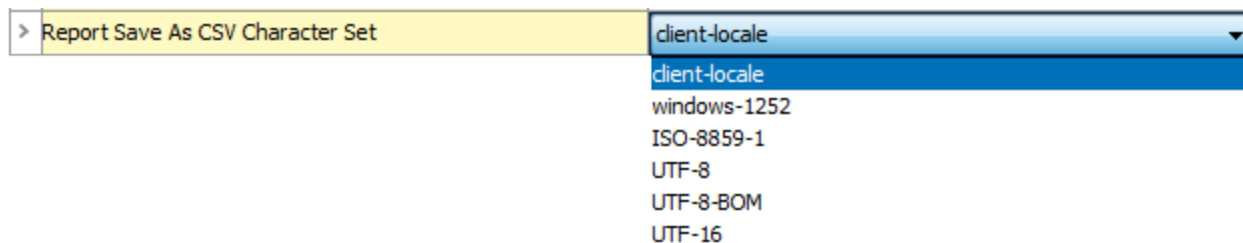
Note: Attributes with the validation of 'Numeric Texts' will not be localized with this setting as they are actually text attributes.

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	abc	
> Density	abc	
> Gross Weight	abc	
> HazMat	abc	
> Manufacturer Part Number	abc	
> Model Series	abc	
> Net Weight	abc	
> Plug Included	abc	
> Short Description	abc	

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		
	Name	DisplaySequence
> 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.

Specifications		
Name	>	Value
> Model Series	abc	
> HazMat	abc	
> Application	abc	
> Short Description	abc	
> Net Weight	abc	
> Gross Weight	abc	
> Density	abc	
> Manufacturer Part Number	abc	
> Plug Included	abc	

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.

Reference Type	▼	>	Target	>	>
> Merchandising Link			 Merchandising Hierarchy/Automotive/Tires/Tire Accessories		
> WebsiteLink	+		 Web Sites US/Acme Retail Web Site/Automotive/Exterior		

Reference Type	▼	>	Target	>	>
> Merchandising Link			 Tire Accessories		
> WebsiteLink	+		 Exterior		

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. See the topic on **Tags** in the **System Setup / Super User Guide** documentation.

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.

Specifications		
Name	>	Value
> Model Series	abc	
> HazMat	abc	
> Application	abc	
> Short Description	abc	
> Net Weight	abc	
> Gross Weight	ID = Short Description	
> Density	DisplaySequence = 40	
> Manufacturer Part Number	abc	
> Plug Included	abc	

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 description of the product.

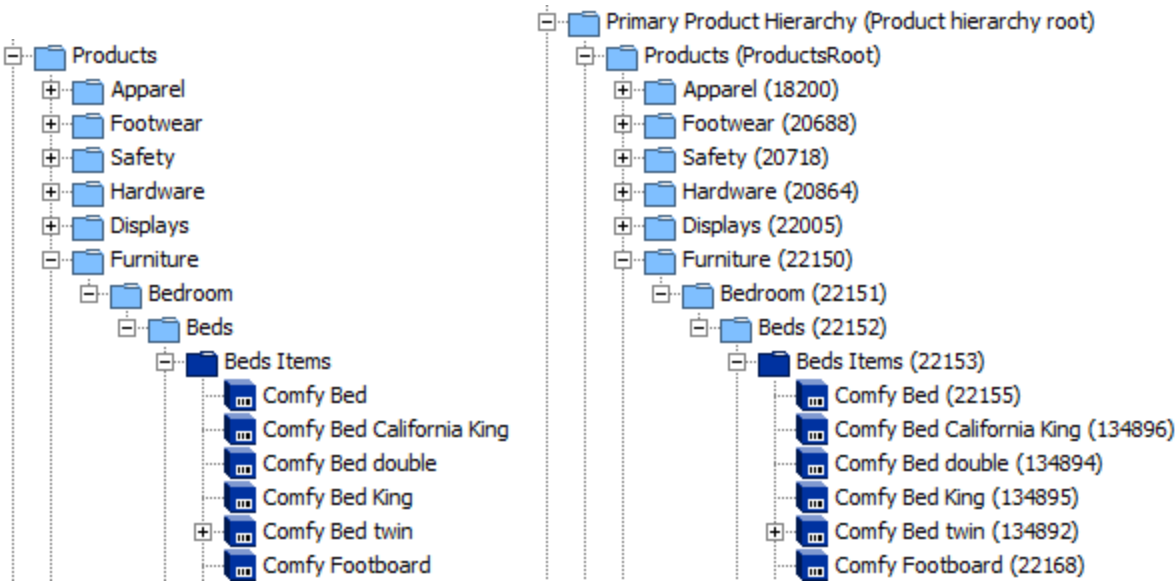
Specifications		
Name	>	Value
> Model Series	abc	
> HazMat	abc	
> Application	abc	
> Short Description	abc	
> Net Weight	abc	
> Gross Weight	ID = Short Description	
> Density	DisplaySequence = 40	
> Manufacturer	Provide a brief description of the product.	
> Plug Included	abc	

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	+		

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 Structured Translation** section of the **Translation** documentation).

These options are available under System Setup > Users & Groups > **Translation Settings**.

System Settings		Log
Translation Settings		
Name	Value	
> Translation File Pattern (procName, caseNum, src and tgt are supported; i.e. 'translation-[caseNum][tgt].xml')	translation-[caseNum]-[tgt].xml	
> Should "Filter spreadsheet" option be shown in Translation Wizard.	Y	
> Default value for "Filter spreadsheet" option in Translation.	N	

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

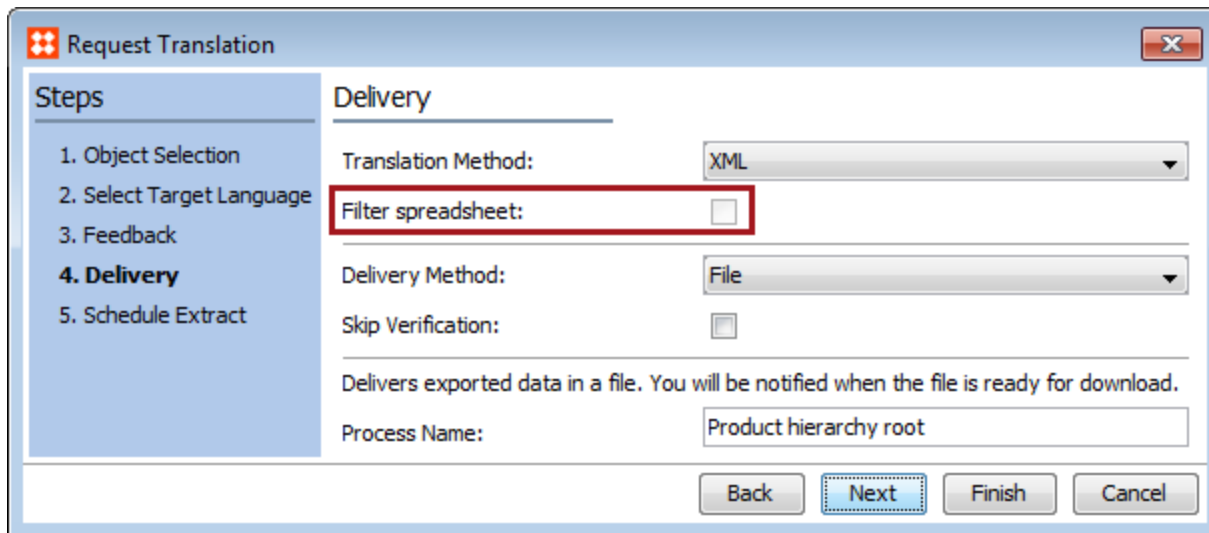
Would create a file with a name such as:

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.



Request Translation

Steps

1. Object Selection
2. Select Target Language
3. Feedback
- 4. Delivery**
5. Schedule Extract

Delivery

Translation Method: XML

Filter spreadsheet: ☐

Delivery Method: File

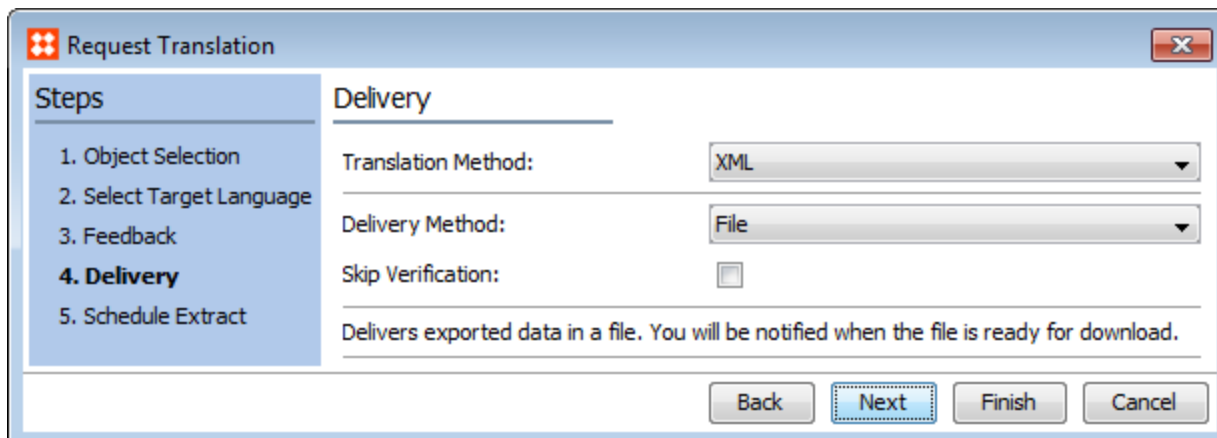
Skip Verification: ☐

Delivers exported data in a file. You will be notified when the file is ready for download.

Process Name: Product hierarchy root

Back Next Finish Cancel

If set to **N**, the 'Filter spreadsheet:' option will not be available on this screen.



Request Translation

Steps

1. Object Selection
2. Select Target Language
3. Feedback
- 4. Delivery**
5. Schedule Extract

Delivery

Translation Method: XML

Delivery Method: File

Skip Verification: ☐

Delivers exported data in a file. You will be notified when the file is ready for download.

Back Next Finish Cancel

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.

Web Services	
Name	Value
> Tag format for Web Services and ReST APIs	HTML3

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.

System Settings		Log
Web UI Settings		
Name	Value	
> Default context	Context1	
> Default workspace	Main	
> Web UI supplier classification object type	SuppliersRoot	
> Web UI supplier products classification object type	SuppliersProducts	
> Assets classification object type	SuppliersAssets	
> Batches classification object type		
> Batch classification object type		
> Excel-template asset	QuickSheetTemplate	
> Proof view stylesheet attribute		
> Step-users Web UI batches folder	SuppliersBatches	
> Step-users Web UI assets folder	SuppliersAssets	
> Context Help metadata attribute	Attribute Description	
> Link type for vendor classification to product link	SupplierLink	
> Enable all-view for users that are a member of multiple suppliers	Y	

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

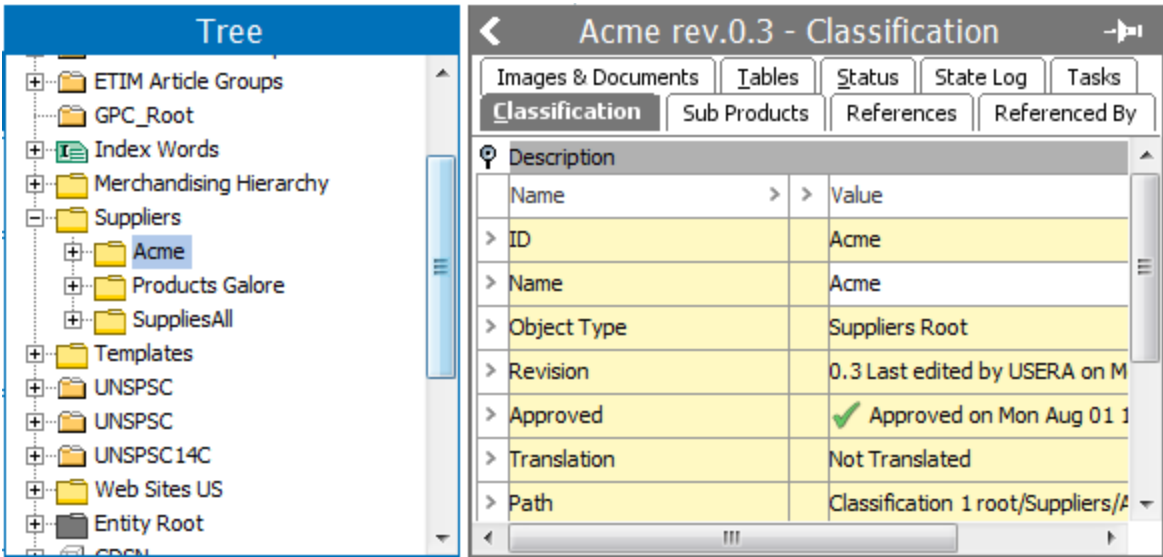
This option determines the Object Type that is 'assigned' to a Web UI user via the Supplier User Group and its associated Classification(s). Filling in this information is only required if you are setting up a Web UI that will include a supplier hierarchy. In the images below, an object type of Suppliers 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, please see the **Setup of STEP Supplier Web UI** documentation.

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, please see the **Setup of STEP Supplier Web UI** documentation.

Note: Enter the ID here and not the Name of the Object Type.

> Web UI supplier products classification object type

SuppliersProducts

Tree

+

 eClass_9_1_0

+

 ETIM6 Article Groups

+

 ETIM Article Groups

+

 GPC_Root

+

 Index Words

+

 Merchandising Hierarchy

-

 Suppliers

+

 Acme

-

 Products Galore

+

 Assets

+

 Batches

-

 Products

+

 (111777)

+

 (111777)(2)

+

 (111778)

+

 109308

+

 129707

+

 129997

+

 129999

+

 190 Lumen LED Cyber

+

 20-68204

Products rev.0.3 - Classification

Status

State Log

Tasks

Referenced By

Images & Documents

Tables

Classification

Sub Products

References

Description

Name	Value
ID	ProductsGaloreProducts
Name	Products
Object Type	Suppliers Products
Revision	0.3 Last edited by USERA ...
Approved	Last Approved on Fri ...
Translation	Not Translated
Path	Classification 1 root/Suppli...
Visibility	
Biodegradable	abc
KeyAtt1	abc
KeyAtt2	abc
Supplier	abc Products Galore

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, please see the **Setup of STEP Supplier Web UI** documentation.

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, please see the **Setup of STEP Supplier Web UI** documentation.

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, please see the **Setup of STEP Supplier Web UI** documentation.

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.

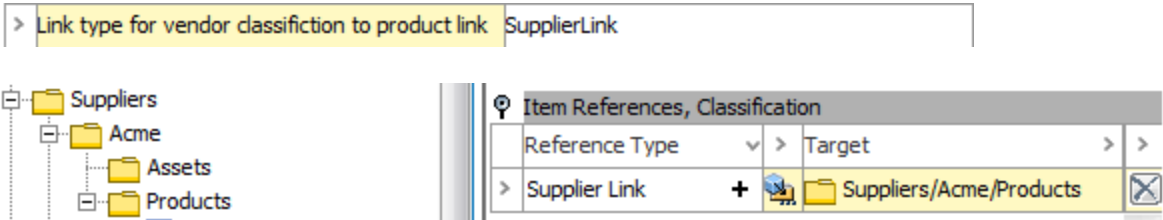
Context Help metadata attribute

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.

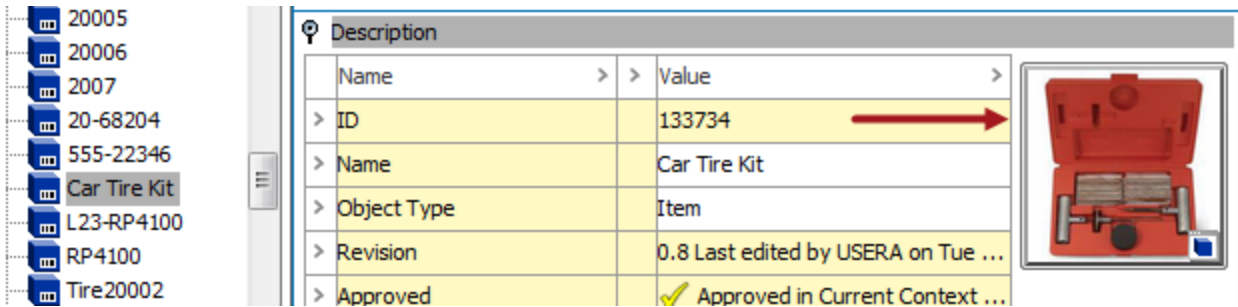


Enable all-view for users that are a member of multiple suppliers

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 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.

Primary Image Type	
Name	Value
Primary Image ReferenceType	PrimaryProductImage

System Setup

- GDSN Receiver CIN Sample
- GDSN Receiver Format
- Illustration
- Installation Manual
- MSDS
- Owners Manual
- Primary Product Image
- Product Image
- Video

Primary Product Image - Reference Type

Reference Type

Validity

Log

Description	
Name	Value
ID	PrimaryProductImage
Name	Primary Product Image
Last edited by	2016-07-28 17:00:28.0 by USERA
Purpose	Used for the single product image that is the primary image to show in STEP and potentially in downstream systems

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 for actual pages	Y
> Show frame information on flatplans with layout pickup	N

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).

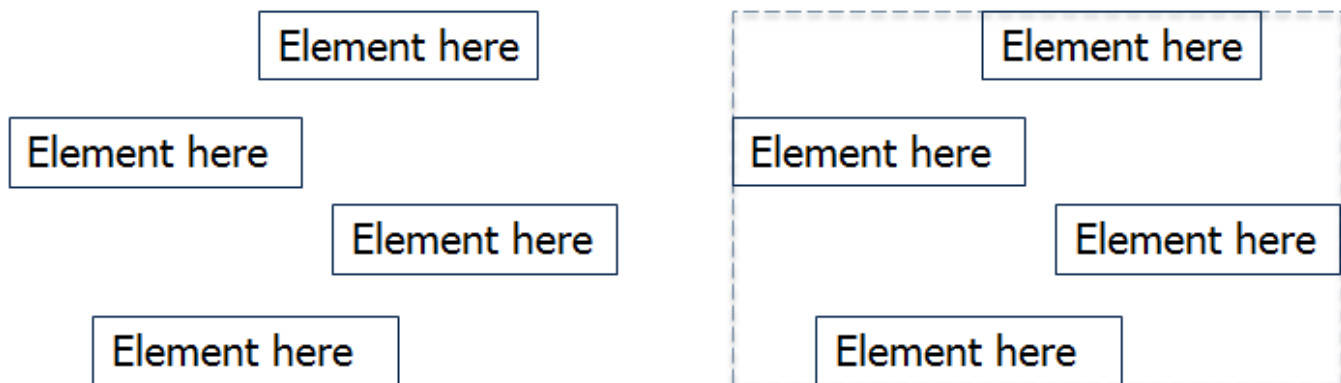
(Note that a 'kept' layout will be removed from the Flatplan if the corresponding mounted [actual] pages are deleted from STEP.)

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.



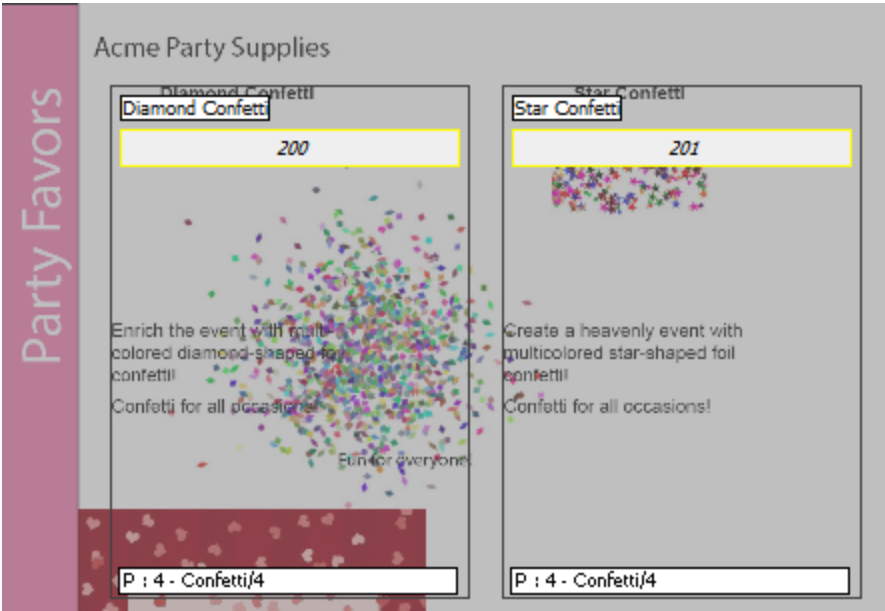
The above example shows the difference between the two methods of measuring space usage. On the left, each of the four elements' individual areas will be used to create the total space usage. On the right, the bounding box is the smallest box that could be drawn to encompass all the individual elements. 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.

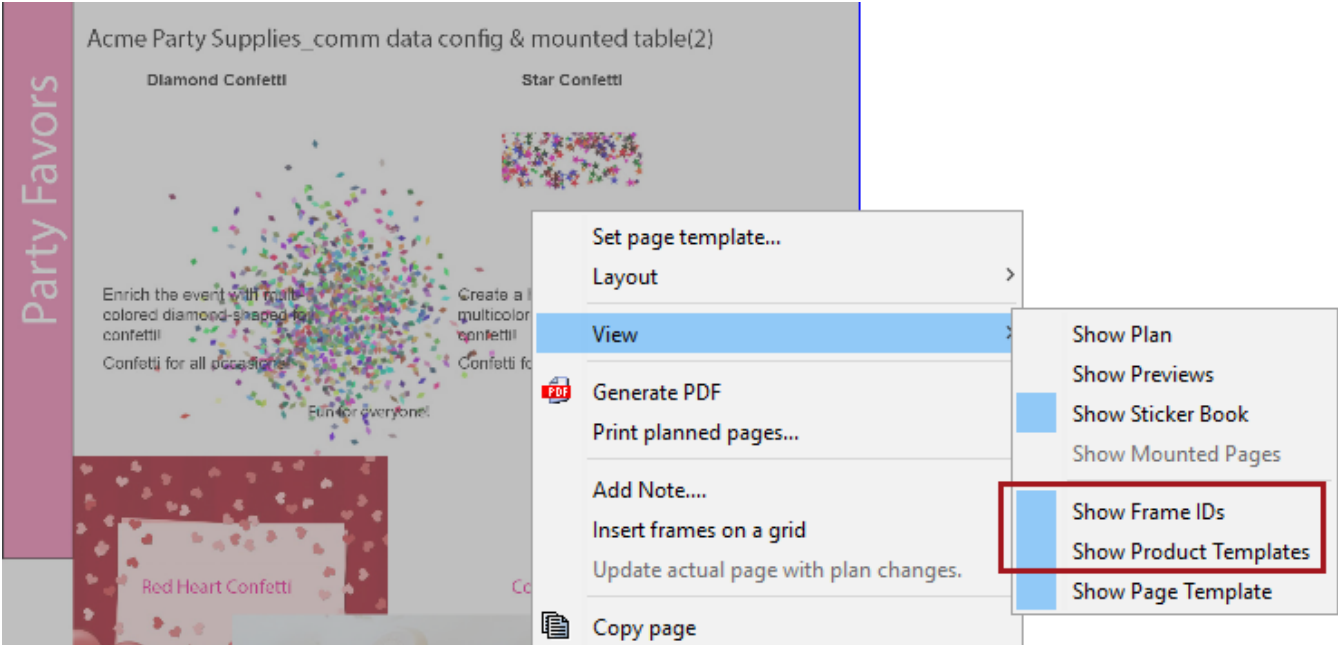
Note that this setting is global for all publications and the decision to set it to Y or N should be made before building a publication. Changing the setting during production could cause space usage to be calculated differently on different pages.

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 Product Overrides.	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, TM), 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 'Y', the table width will exactly match the frame width, as the side rules will be fit inside the frame.

Attribute Display Sequence

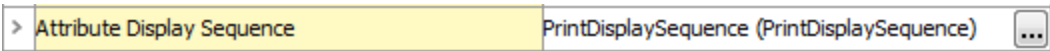
The 'Attribute Display Sequence' setting under DTP Settings is different than the 'Default Attribute to use as Display Sequence Attribute' setting under Product Information Manager Default settings. Whereas 'Default Attribute to use as Display Sequence' is used to sort the order of attributes as they 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.

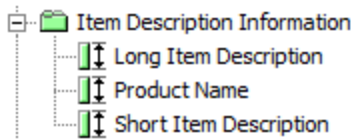
```
<STEPGROUP AG="ItemDescriptionInformation" RS="\n"></STEPGROUP>
```

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.

Item Description Information		
Name	>	Value
Product Name	abc	Husky 120-Volt Portable Tire Inflator
Short Item Description	abc	120-Volt Portable Tire Inflator
Long Item Description	abc	The Husky 120-Volt Inflator inflates vehicle tires, bicycle tires, air mattresses and sports equipment.

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.

120-Volt Portable Tire Inflator The Husky 120-Volt Inflator inflates vehicle tires, bicycle tires, air mattresses and sports equipment Husky 120-Volt Portable Tire Inflator

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.

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.

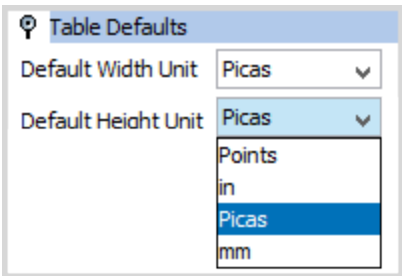
Reapply Flatplan rules when adding product to non-empty frame

This option is only relevant for Galley Flatplans, which allow multiple products to be placed in the same frame.

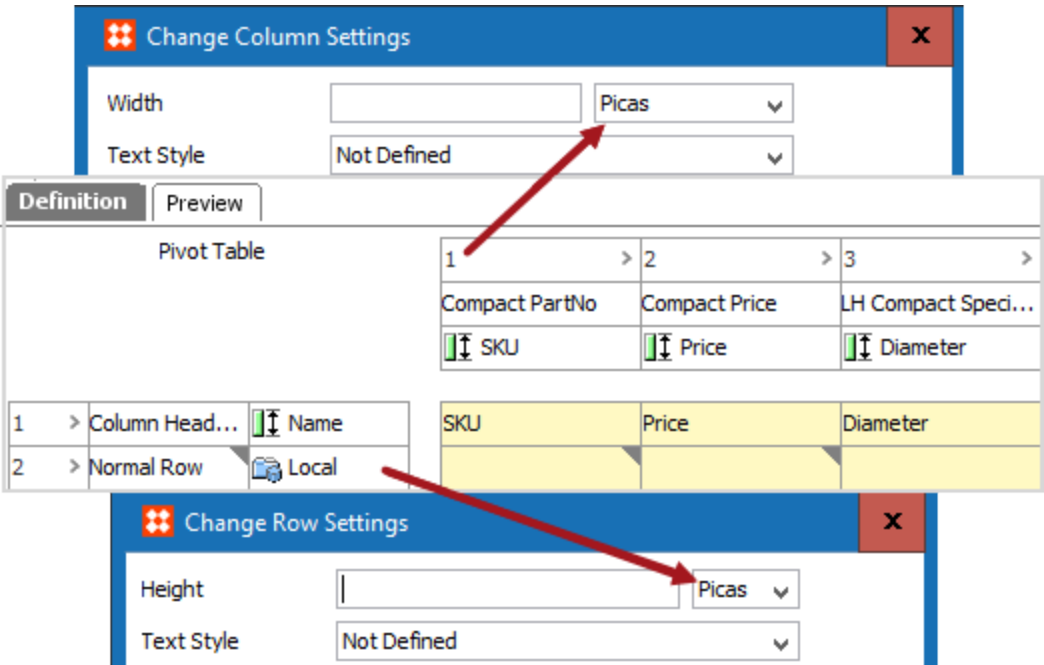
Normally, Flatplanner does not check Flatplan publication rules if a template is already assigned to a frame. Setting this option to 'Y' forces it to do so.

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.



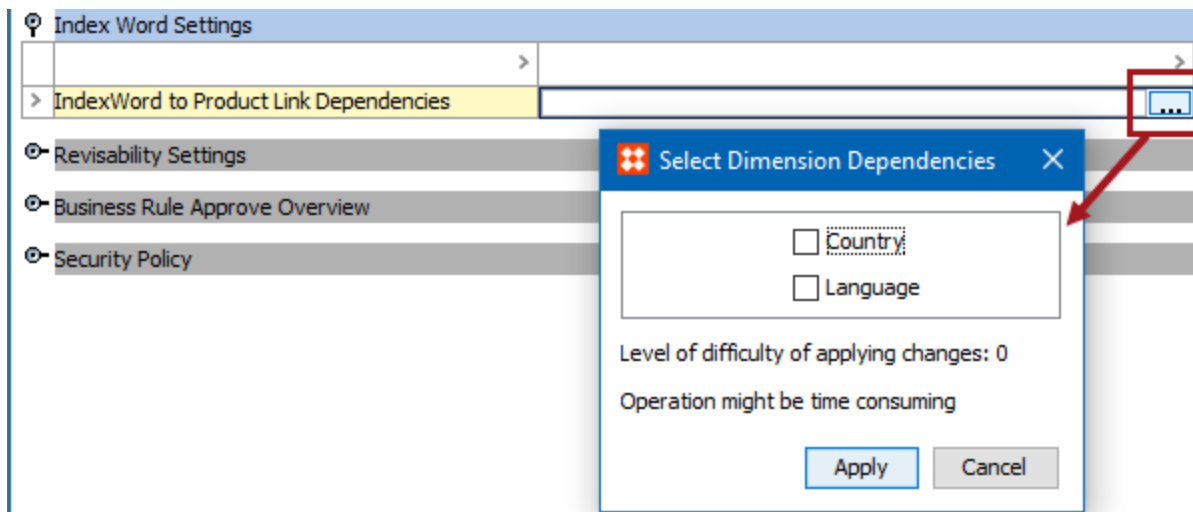
Once the default units have been set, the units will be the default units in the table editor as shown in the image below. If no default unit is selected, the default units will be blank and must be manually selected every time.



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

Revisability Settings	
Name	Value
Threshold (hours) for changes by same user to generate a new revision	168

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:

Security Policy

Property	> Enable
> Password Strength Validation	☐
> Password Expiration	☐ Expires after 0 days since last time the password was changed
> Prohibit Password Reuse	☐ Reusable after 0 days
> Lock out User after 3 consecutive invalid login attempts	☐ Lock out for 0 seconds
> Inactivity timeout period	☐ Time out after 0 minutes.

Edit

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:

> Lock out User after 3 c

> Inactivity timeout peri

Edit

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.

Property	> Enable
> Password Strength Validation	☒

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.

> Password Expiration	☐ Expires after	0 days since last time the password was changed

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.

> Prohibit Password Reuse	☐ Reusable after	0 days

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.

> Lock out User after 3 consecutive invalid login attempts	☐ Lock out for	0 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.

> Inactivity timeout period	☐ Time out after	0 minutes.

GUI Localization

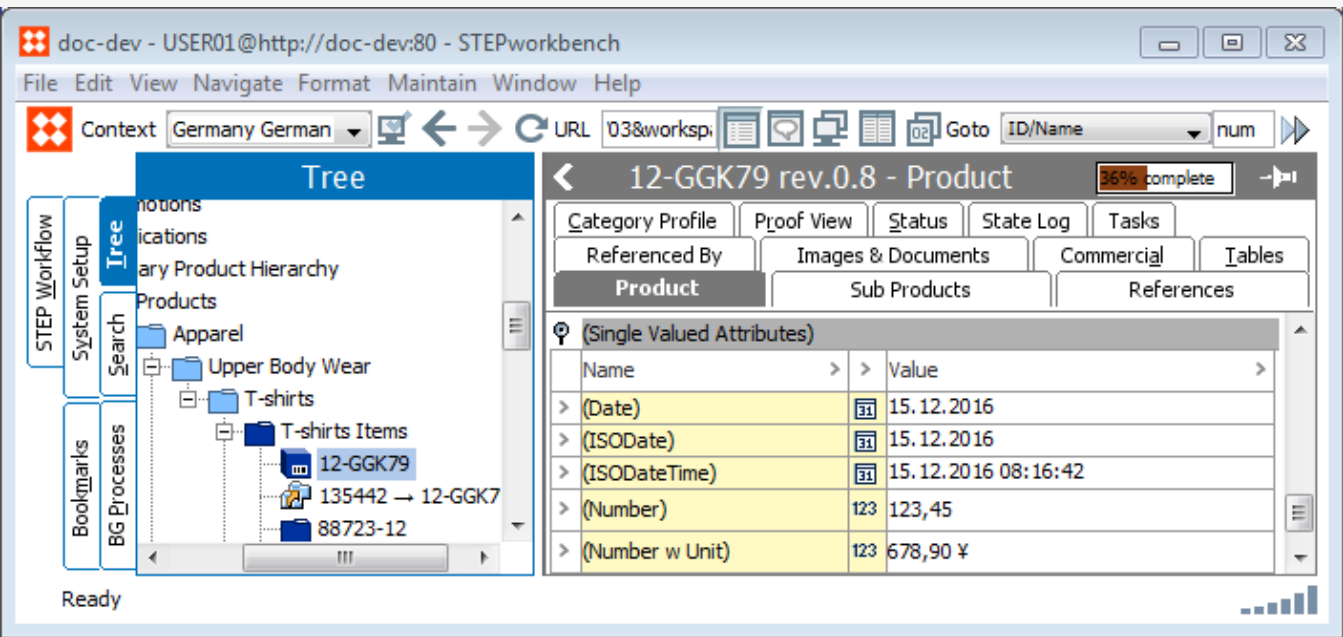
GUI Localization is maintained in System Setup > **Users & Groups** > **Users** on the **System Settings** tab.

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

The screenshot shows the STEPworkbench interface for User02. The 'Tree' view on the left displays a hierarchy: Products > Apparel > Upper Body Wear > T-shirts > T-shirts Items > 12-GGK79. The main panel shows the '12-GGK79 rev.0.8 - Product' details. The 'Product' tab is active, showing a table of 'Single Valued Attributes'.

Name	Value
(Date)	15-dec-2016
(ISODate)	2016-12-15
(ISODatetime)	2016-12-15 08:16:42
(Number)	123 123.45
(Number w Unit)	123 678.90 ¥

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.

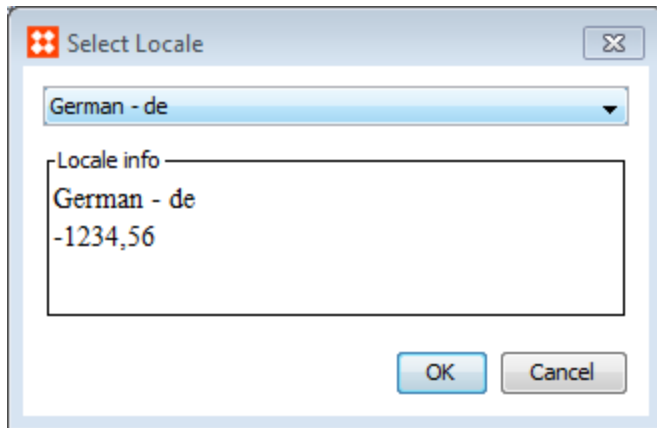
Setting up a GUI Localization for a User

GUI Localization is maintained in [System Setup]: **Users & Groups** -> **Users** in the **System Settings** tab.

The screenshot shows the STEPworkbench interface for User02. The 'System Setup' view on the left displays a hierarchy: Merch Planning > Product Group > user01, user02, user03, user04, user05. The main panel shows the 'System Settings' tab, which is active. The 'PIM GUI Localization' section is visible, showing a table of 'Single Valued Attributes'.

Name	Value
Locale for values	
Localize numbers using thousand deli...	N
Localize dates	N

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.

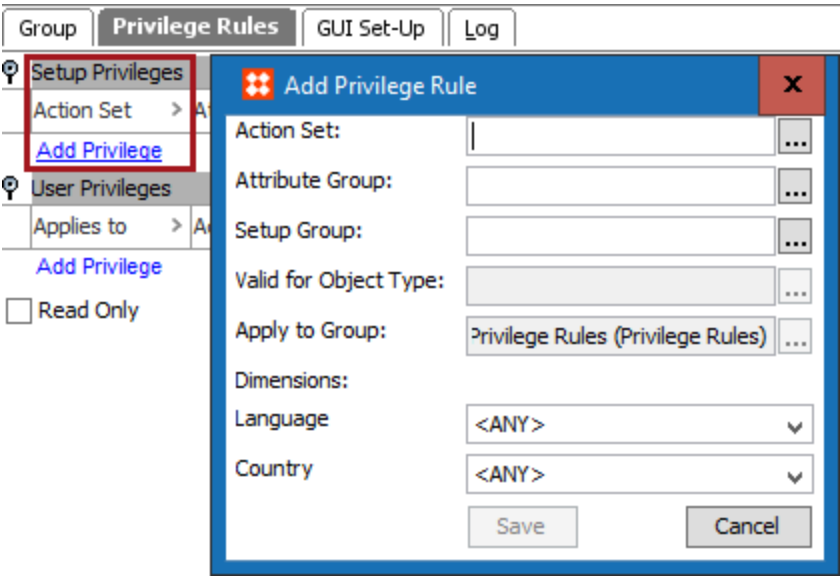
4. In **Localize dates**, click Y to localize dates in the Workbench. In this example a German -de locale is selected for User 01. For User 01, isodates will be shown as 13.05.2009 and isodatetime will be shown as 13.05.2009 19:10:48.

applied a top User Group will not be inherited to any sub User Group. It is to be recommended to apply all general Privilege Rules as high as possible in the User Groups Hierarchy to simplify management of the Privilege Rules.

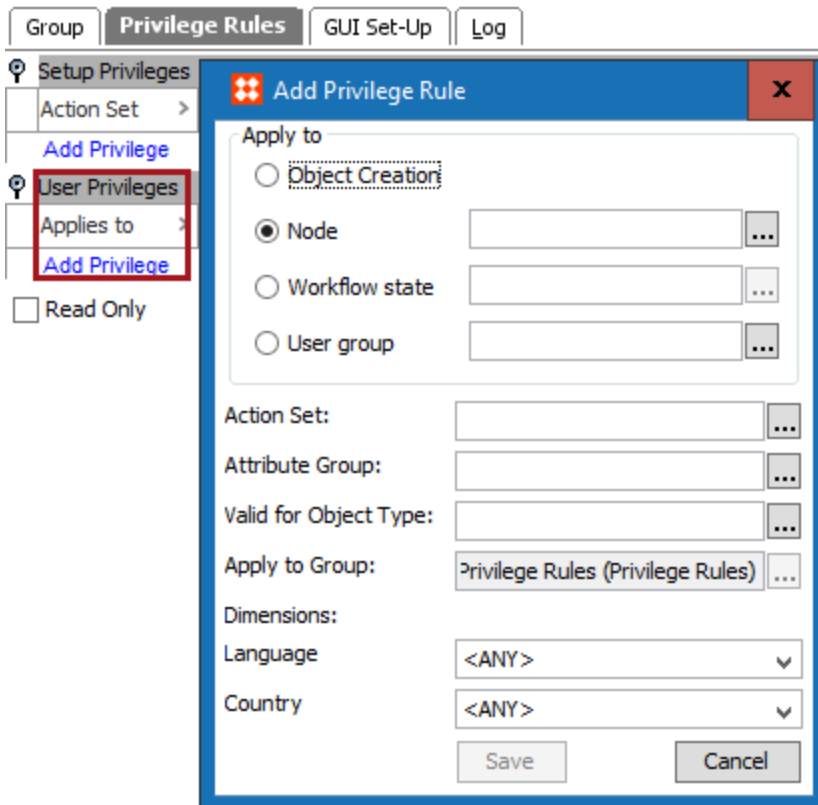
Privilege Rule Types

There are two types of Privilege rules:

Setup Privileges: These are usually actions that are usually performed by a System administrator.



User Privileges: For actions that should be performed by the users of a specific group.



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 an Attribute Group(s) is not specified, the Action is permitted for all Attributes.
- Workflows, Integration Endpoints, and 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, 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 accumulate the Privilege rules from all these Groups, so a restriction from one Group is ignored if the corresponding Action is permitted via another Group membership.

Editing linked Objects

If 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

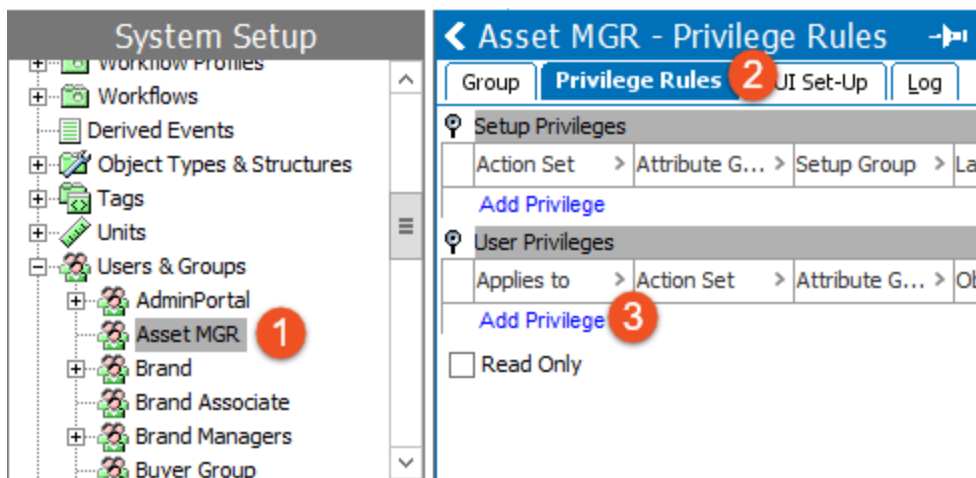
If a Group has permission to work with a specific Classification node, Attribute Values can be edited for all Products linked to (or below) that node. However, access to the Classification node does not grant the privilege to link new Products in the Product hierarchy. To do that, the Group must have permission to work with the specific Product node as well.

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 be applied to the same Group so you can be as granular as needed regarding specific actions for specific objects, attributes and so on. Remember, the Privilege Rules applied at the group level will affect all users and sub-groups within that level.

1. From the System Setup menu, unfold the Users and Groups node and select the relevant User Group.
2. Click the **Privilege Rules** tab.
3. In the User Privileges field, click **Add Privileges**.



4. The **Add Privilege Rule** dialog box will appear as shown in the image below.

 Add Privilege Rule

☐ Object Creation

☒ Node

☐ Workflow state

☐ User group

Language

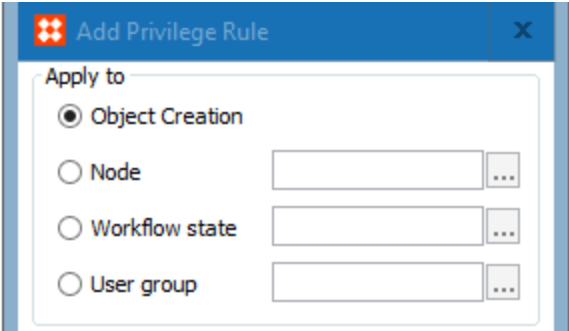
Country

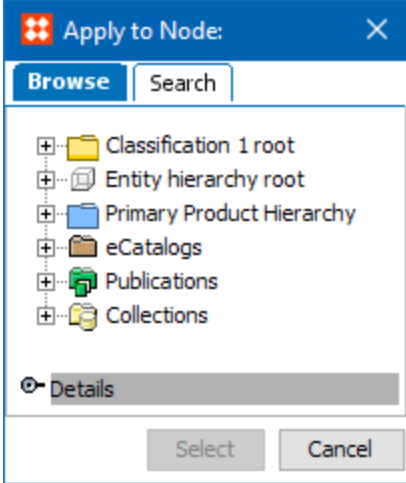
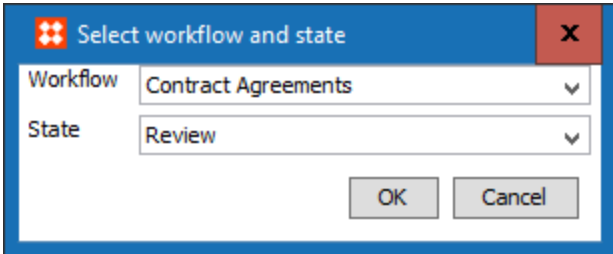
Save

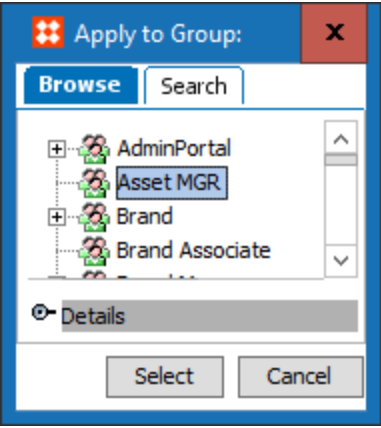
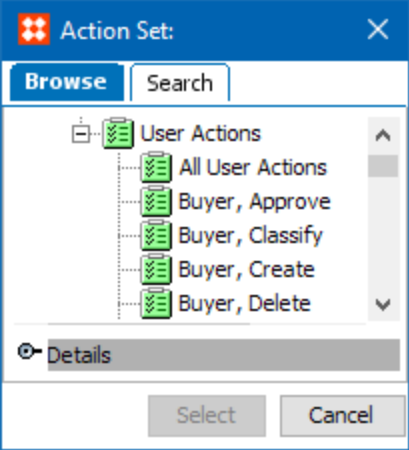
Cancel

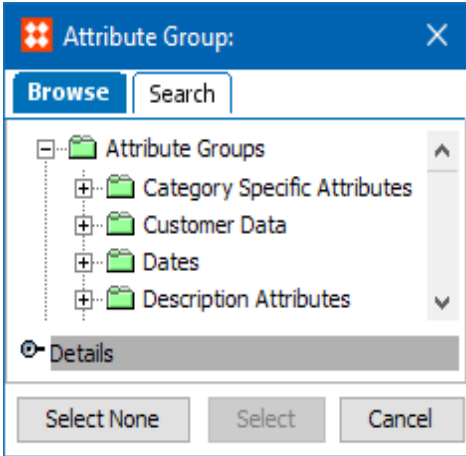
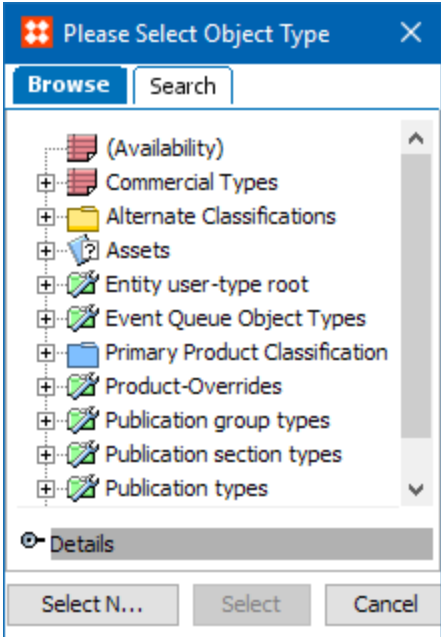
5. Make the relevant changes, and then click **Save**.

Add Privilege Rule Options

Setting	Description
Object Creation	Administrators can set permissions for users when creating new objects in STEP via the Object Creation parameter which is available when configuring user privileges. More information about this parameter can be found in Object Creation Privileges. 
Apply to Node	Select the allowed classification or product node for the group. This would include specific sub-nodes as well.

Setting	Description
	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. 
Workflow state	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. More information about this parameter can be found in Configuring Workflow Privileges. 
User Group	Web UI 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. This is done through selecting a User Group. More information about this parameter can be found in Web UI User Impersonation.

Setting	Description
	
Action Set	Select the permitted Action Sets for the Group. Remember to create Action Sets that are very specific to the privileges that you want the User Group to have. 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. Note that if are attributes within a group that should not be valid for a User to have access to, a separate group should be maintained containing the attributes the Users should have access to. 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 the System Setup / STEP Super User Guide documentation.
Apply to Group	Select the User Group that the Privilege Rules should be applied to. This is usually not necessary if you are already on the group that you are applying Privilege Rules for.
Dimensions	Optional. Select the permitted Language or Country dimension.

Setting	Description
	If a restriction to a dimension is required, you should know: • If a Privilege Rule to modify an attribute value is restricted to a dimension, user will only have the ability to modify the dimension dependent attribute values. • They are not able to modify values that have no dimension dependency. Note: The default of Any will allow the Group to work with all dimension points. For more information about dimension points, see the Dimensions and Dimension Points topic in the System Setup / STEP Super User Guide documentation. Dimensions: Language <ANY> Country <ANY> Save Cancel

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].

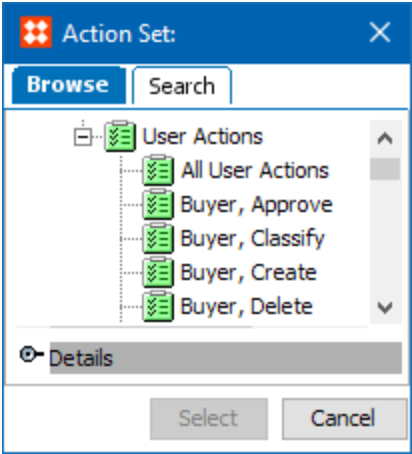
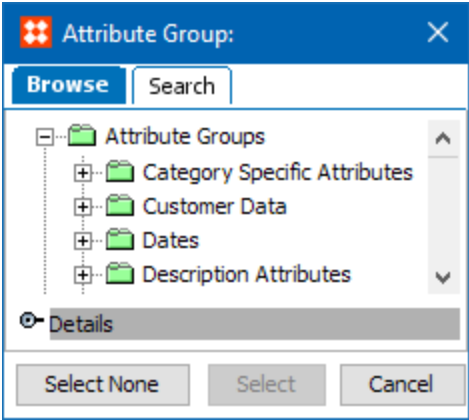
1. From the System Setup menu, unfold the Users and Groups node and select the relevant User Group.
2. Click the **Privilege Rules** tab.
3. In the User Privileges field, click **Add Privileges**.

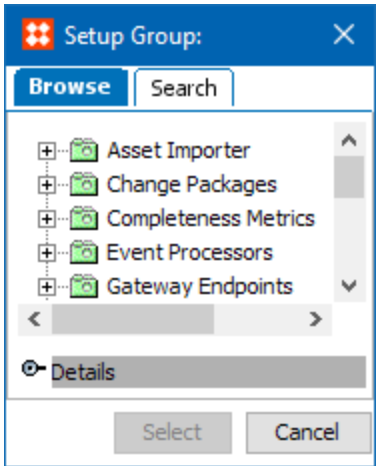
An **Add Privilege Rule** dialog box appears.

4. Make the relevant changes, and then click **Save**.

Add Setup Privilege Rule Options

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 Dimension to be accessed.

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 the **Dimensions and Dimension Points** topic in the **System Setup / STEP Super User** documentation.

Note: Setting dimension points is a very time consuming process.

A Setup Privilege rule has now been added to the Group.

Editing Privilege Rules

Privileges are maintained in [System Setup]: **Privilege Rule** tab in the **User Group** Editor.

System Setup

- Image Group
- Integration
- Marketing Copy Group
- Merchandiser Managers
- Merchandisers
- Merch Planning
- Privilege Rules
- Product Group
- Read Only
- Regulatory Group
- Restricted User
- SDK and API users
- Stibo Users**
- Super Users
- Suppliers
- User Group
- Warehouse Group
- Web UI Group

Stibo Users - Privilege Rules

Group | **Privilege Rules** | GUI Set-Up | Log

Setup Privileges

Action Set	Attribute Group	Setup Group	Language	Country
> All Setup Actions			<ANY>	<ANY>

[Add Privilege](#)

User Privileges

Applies to	Action Set	Attribute Group	Object Type	Group	Language	Country
> Classification 1 root	All User Actions			Stibo Users	<ANY>	<ANY>
> Collections	All User Actions			Stibo Users	<ANY>	<ANY>
> eCatalogs	All User Actions			Stibo Users	<ANY>	<ANY>
> Entity hierarchy root	All User Actions			Stibo Users	<ANY>	<ANY>
> Primary Product Hierarchy	All User Actions			Stibo Users	<ANY>	<ANY>
> Publications	All User Actions			Stibo Users	<ANY>	<ANY>

[Add Privilege](#)

☐ Read Only

For more information about Privileges Rules and the Privileges Rule Types, see **Privilege Rules** in the **Users and Groups** documentation.

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.

Add Privilege Rule

X

Apply to

☐ Object Creation

☒ Node

☐ Workflow state

☐ User group

Action Set:

Attribute Group:

Valid for Object Type:

Apply to Group:

Privilege Rules (PrivilegeRules)

Dimensions:

Language

<ANY>

Country

<ANY>

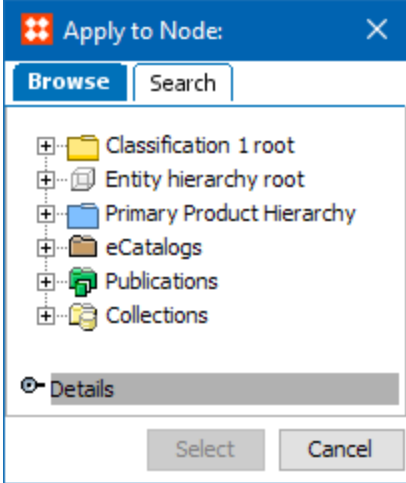
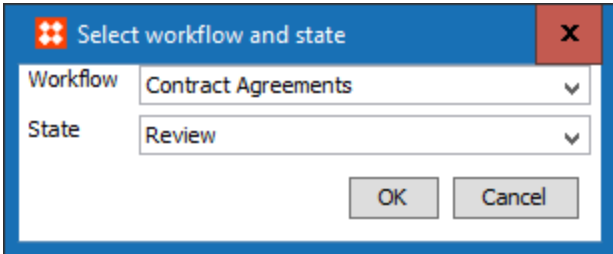
Save

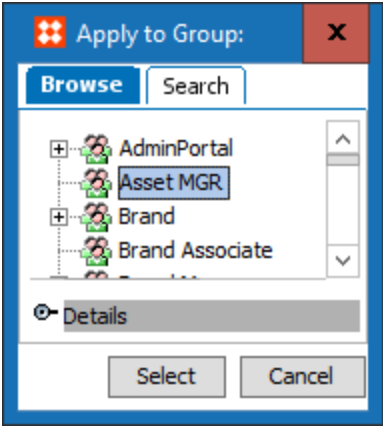
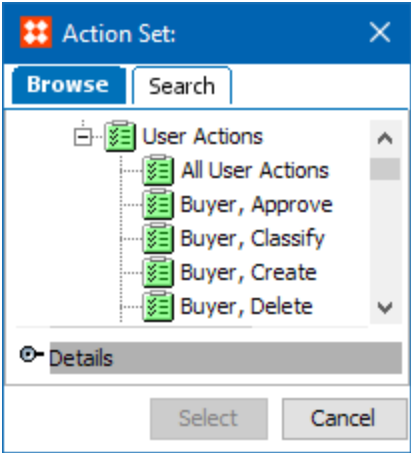
Cancel

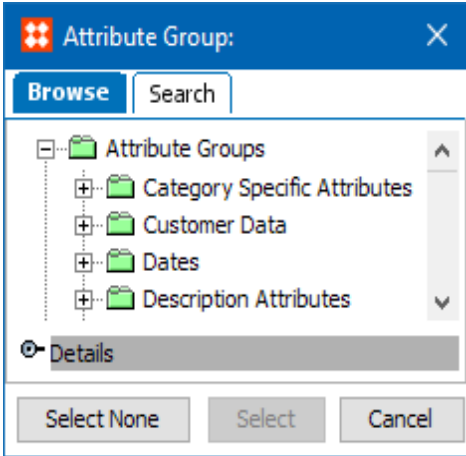
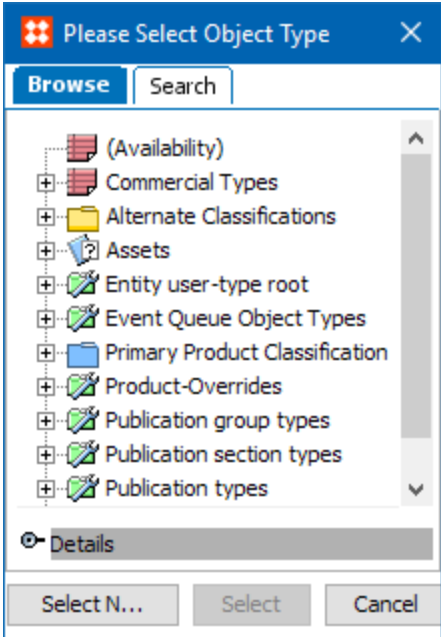
4. Make the relevant changes, and then click **Save**.

Edit Privilege Rule Options

Setting	Description
Object Creation	Administrators can set permissions for users when creating new objects in STEP via the Object Creation parameter which is available when configuring user privileges. More information about this parameter can be found in Object Creation Privileges. Add Privilege Rule X Apply to ☒ Object Creation ☐ Node ☐ Workflow state ☐ User group

Setting	Description
	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. 
Workflow state	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. More information about this parameter can be found in Configuring Workflow Privileges. 
User Group	Web UI 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. This is done through selecting a User Group. More information about this parameter can be found in Web UI User Impersonation.

Setting	Description
	
Action Set	Select the permitted Action Sets for the Group. Remember to create Action Sets that are very specific to the privileges that you want the User Group to have. 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. Note that if are attributes within a group that should not be valid for a User to have access to, a separate group should be maintained containing the attributes the Users should have access to. 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 the User Group that the Privilege Rules should be applied to. This is usually not necessary if you are already on the group that you are applying Privilege Rules for.
Dimensions	Optional. Select the permitted Language or Country dimension. If a restriction to a dimension is required, you should know:

Setting	Description
	• If a Privilege Rule to modify an attribute value is restricted to a dimension, user will only have the ability to modify the dimension dependent attribute values. • They are not able to modify values that have no dimension dependency. Note: The default of Any will allow the Group to work with all dimension points. For more information about Dimension Points, see the Contexts topic in the System Setup / STEP Super User documentation. Dimensions: Language <ANY> ▼ Country <ANY> ▼ Save Cancel

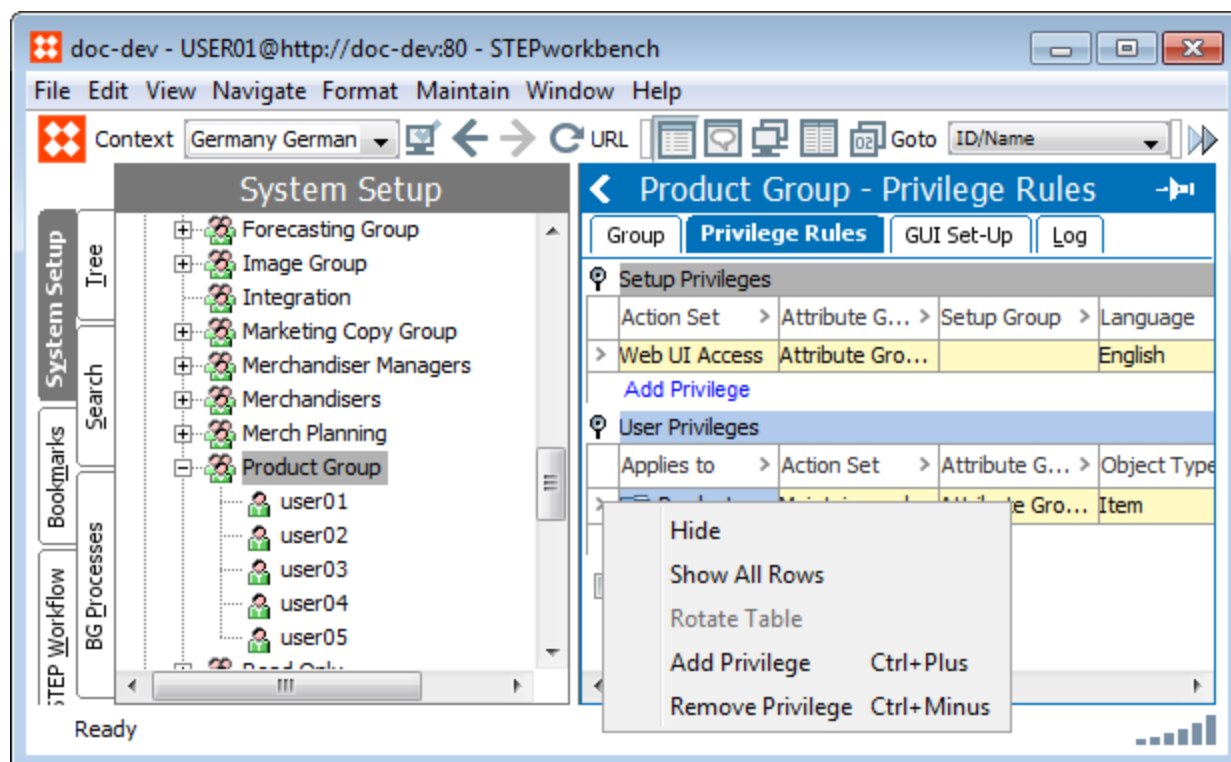
Removing a Privilege Rule from a Group

1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group.

The Group appears in the **Group Editor**.

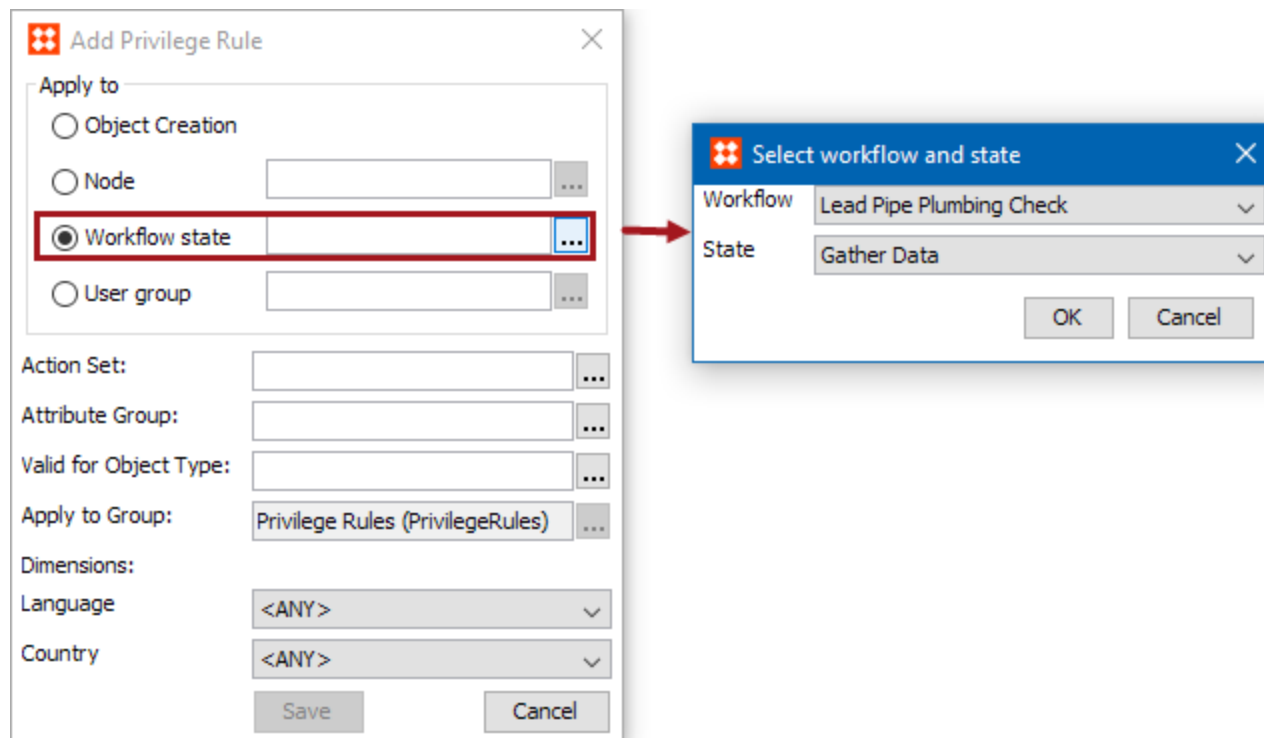
2. Click the **Privilege Rules** tab.
3. Click the row header of the Privilege rule to be removed, and then click **Remove Privilege**.

The Privilege rule will now be removed from the list and will no longer apply to the members of the selected Group.



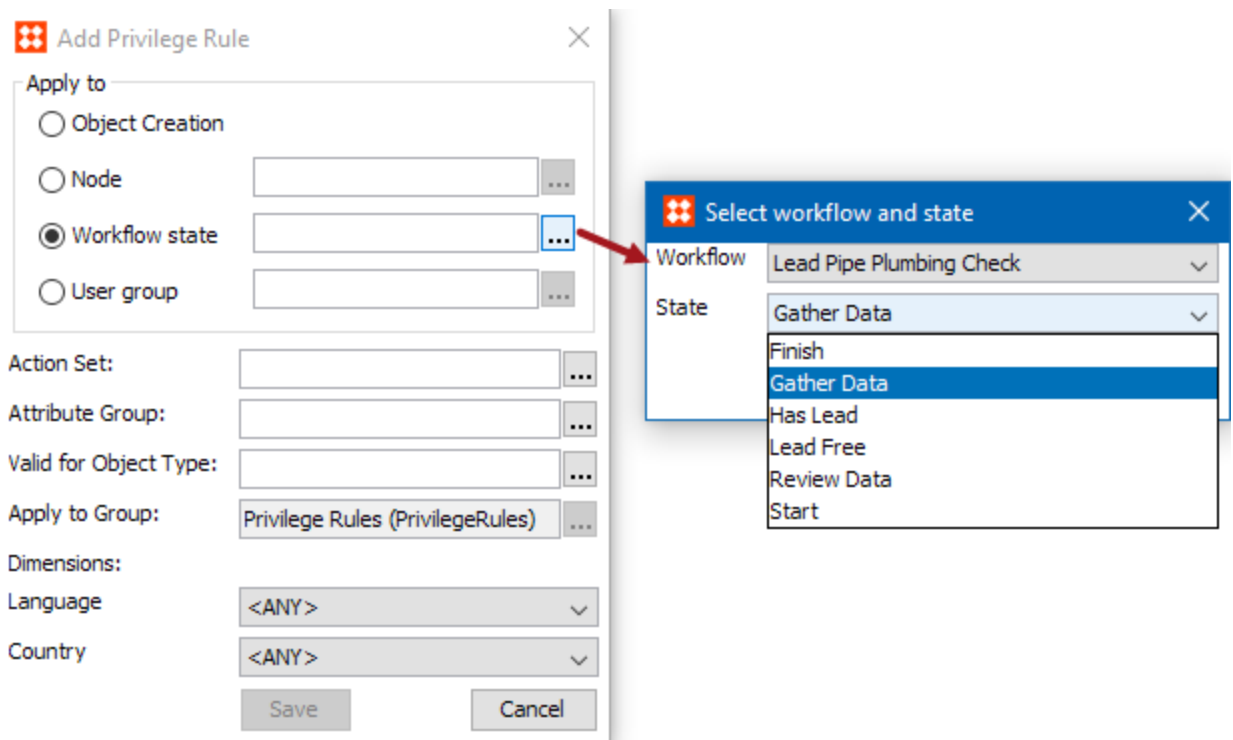
Configuring Workflow Privileges

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:



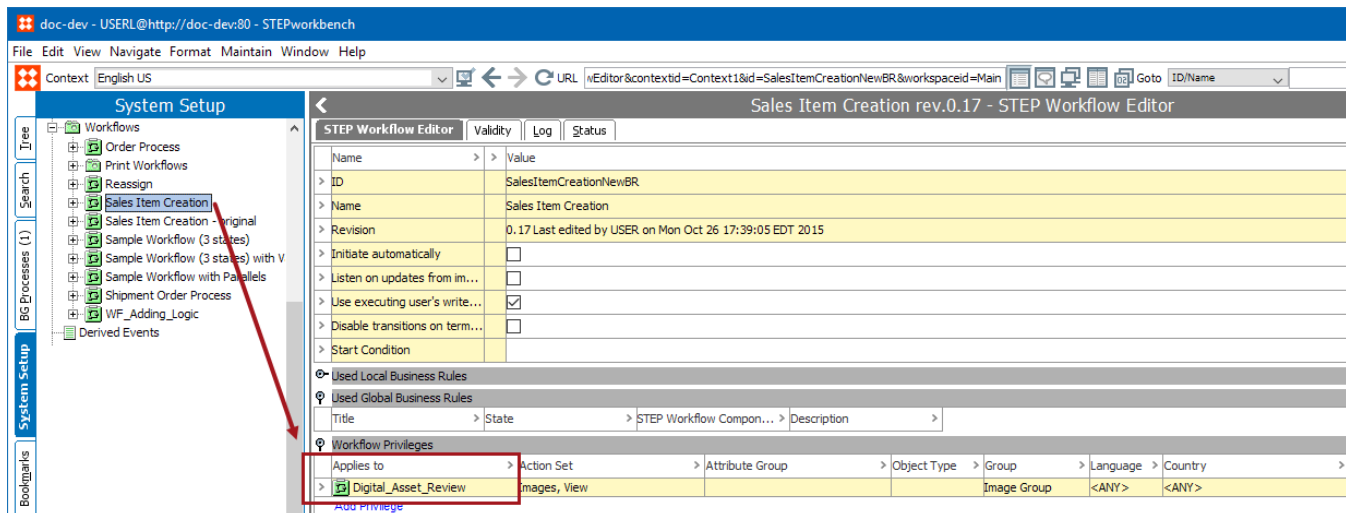
Workflow state selector

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.

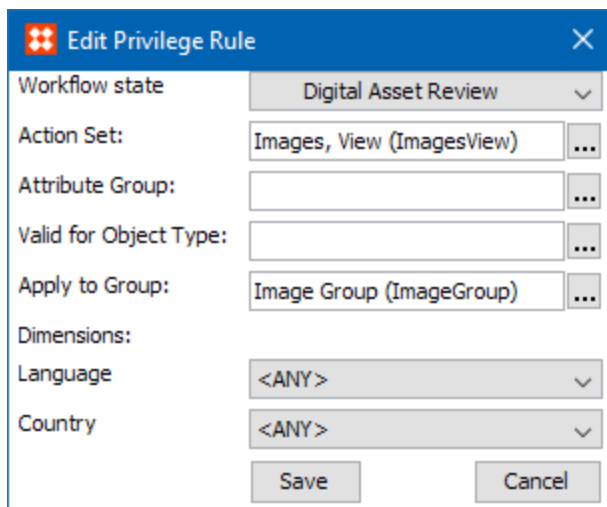


Editing Privilege Rules with focus on STEP Workflows

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

Object Creation Privileges

Administrators can set permissions for users when creating new objects in STEP via the Object Creation parameter which is available when configuring user privileges. The Object Creation parameter is intended for use on Initiate Item, Mass Creation, and Simple Importer in Web UI, as well as when using Smartsheets and/or the STEP Importer.

Add Privilege Rule

Apply to

☒ Object Creation

☐ Node

☐ Workflow state

☐ User group

Action Set:

Attribute Group:

Valid for Object Type:

Apply to Group: Privilege Rules (PrivilegeRules)

Dimensions:

Language: <ANY>

Country: <ANY>

Save Cancel

Selecting this allows users to create either new assets, classifications, entities, or products only, and populate specific data values on the new object that they have been given permission to populate. For example, when creating a product, a user may be able to populate certain attribute values, but once the object is created, they no longer have access to these data fields, and thus can't change them. This allows for more granular control over data so that users can be given access to data during the creation process only, but not to the same data after the object has been created. Following creation of the object, any global or workflow specific permissions will take over.

For more on how to set up privileges and how they work, see the topic on Users and Groups in the System Setup / STEP Super User Guide documentation.

For more on how to setup global and business rules, see the Local and Global Business Rules topic in the Business Rules documentation.

For more on workflow specific permissions see the, Configuring Workflow privileges topic in the Super User Guide documentation.

It is worth noting that these object creation privileges do not apply to any new setup type nodes. Also, if an administrator selects to apply the object creation permissions to an unsupported object type, such as eCatalogs, publication sections, pages, attributes, etc., there will not be any kind of warning message.

Note: The Object Creation privilege is not recommended for use when using Web Services, Quicksheets, and creation in workbench outside of an importer. This is because users do not populate values for new objects in these cases until after creation of the object.

Considerations When Creating New Object

When setting up permissions for Object Creation in STEP, administrators need to keep in mind certain aspects when their users create new objects through the different means: Mass Creation in Web UI, as well as Smartsheets and/or the STEP Importer (e.g. Import Manager or inbound integration endpoints).

Mass Creation Screen in Web UI

If a field for entering an attribute value is shown on the Mass Creation screen that the user does not have privileges to populate, this will result in a 'privileges' error that will not be detected until after the 'Create' button is clicked. Administrators should take care that the Mass Creation screen is configured appropriately so that only fields that the end user has permission to edit are visible.

For more on mass creation, see [Onboarding Multiple Objects Using Web UI Screens in the Web User Interface / STEP Web UI Setup and User Guide](#) documentation.

Smartsheets

Like the Mass Creation screen, care should be given when creating the Smartsheet template. If fields for attributes, references, etc. are editable that should not be to the end users, this will give a privilege error when attempting to validate the Smartsheet or importing into STEP. Additionally, the author of any business rules that will be executed during a validation or import process will need to consider the 'Object Creation' and other create-privileges rules.

For more on Smartsheets, see the [Excel Smartsheet Format](#) documentation.

Importer

This is not recommended for use with imports where the importer is creating references to objects that are being created as part of the same import, as this may result in errors. Furthermore, administrators should ensure that only the attributes that the privilege is applied to are available in the user interface, else the user will receive a privilege error. The author of any business rules that will be executed during an import process will need to consider the Object Creation and other create-privileges rules.

For more about importing, see the [Data Exchange](#) documentation.

Privileges and Security

Unlike workflow state-specific privileges, the Object Creation privilege is not associated with any state in any workflow, so it works similar to global privileges, however only while the object is in the process of being created. After the object is created, any global or workflows specific privileges will take place.

The list of actions that can be associated with an Object Creation privilege are limited to the following list:

Object (node) Type	Action
Asset	• Classify asset (link asset to classification) • Create asset reference • Modify name / description of asset (also translate) • Modify metadata for asset reference (link attribute value, and also translate) • Modify metadata for asset (property value, and also translate)
Classification	• Create classification reference • Modify name / description of classification (also translate) • Modify metadata for classification reference (link value, and also translate) • Modify metadata for classification (property value, and also translate)
Entity	• Create entity reference • Modify name / description for entity (also translate) • Modify metadata for entity reference (link value, and also translate) • Modify metadata for entity (property value, and also translate)
Product	• Classify product (link product to classification) • Create entity reference • Modify name / description of product (also translate) • Modify product attribute value (also translate) • Modify metadata for product classification (link value, and also translate) • Modify metadata for product reference (link attribute value, and also translate)

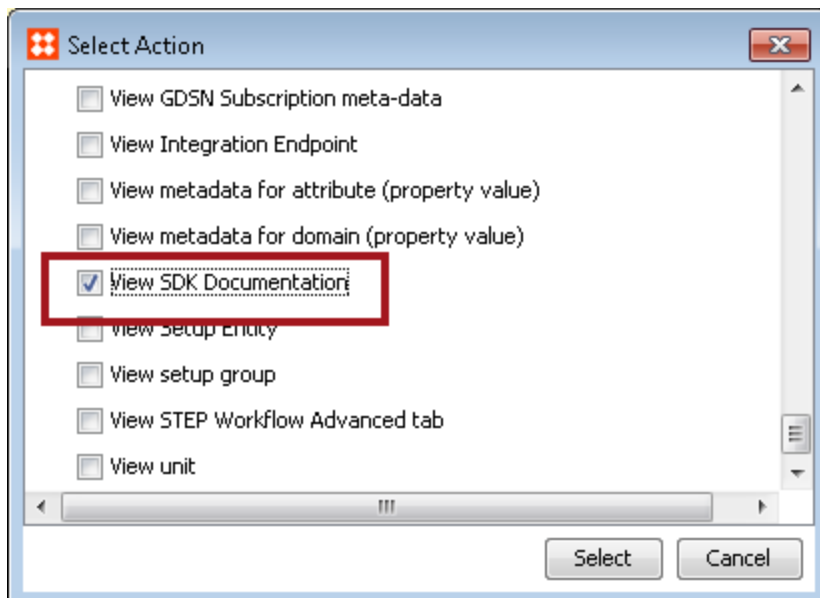
Granting Access to the SDK and API Documentation

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.

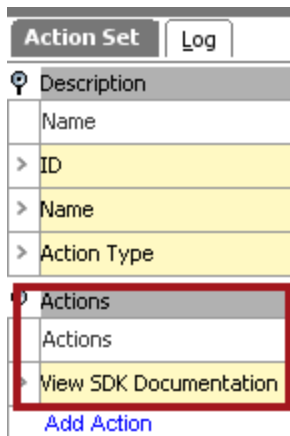
Assigning the privilege is a two-step process: First, you create an action set that includes the **View SDK Documentation** setup action privilege, and then you assign the action set to the relevant groups.

Grant Access to the SDK and API documentation

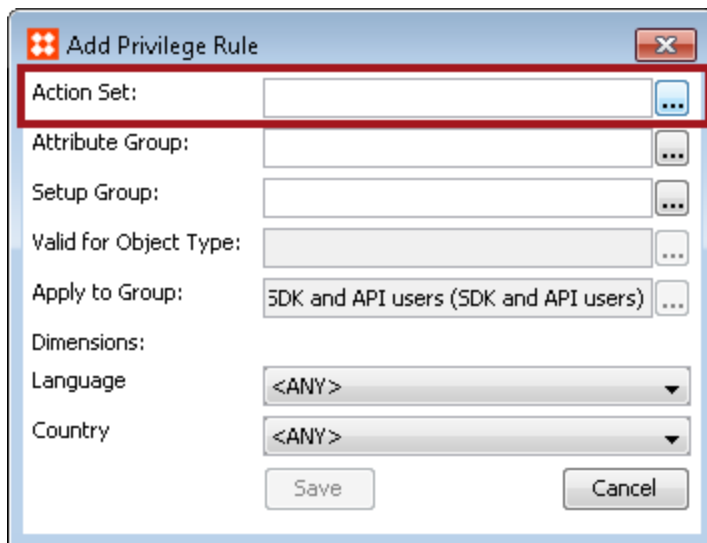
1. In **System Setup**, expand **Action Sets**, right-click **Setup Actions**, and then click **New Action Set**.
2. Enter an **ID** and a **Name** for the action set, and then click **OK**.
3. On the **Action Set** tab, click **Add Action**.
4. In the **Select Action** dialog, scroll down and select **View SDK Documentation**, and then click **Select**.



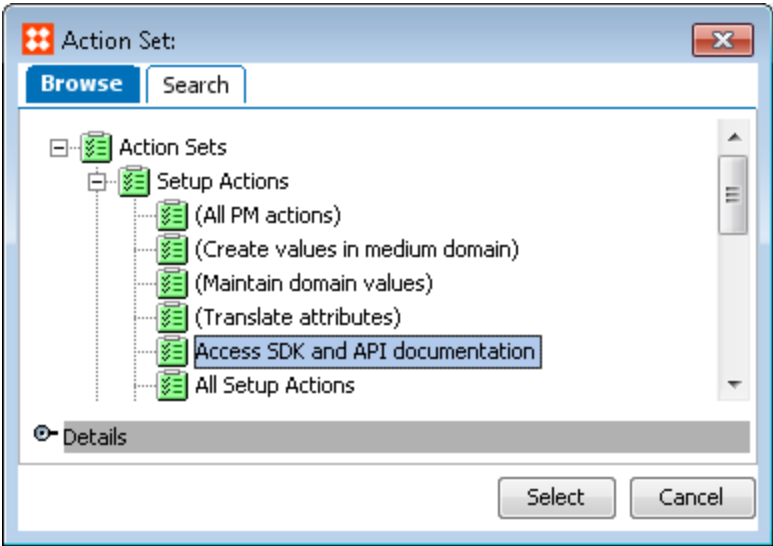
The action is now listed in the **Actions** area on the **Action Set** tab, and you can add the action set to the relevant user group.



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.



GUI Setup

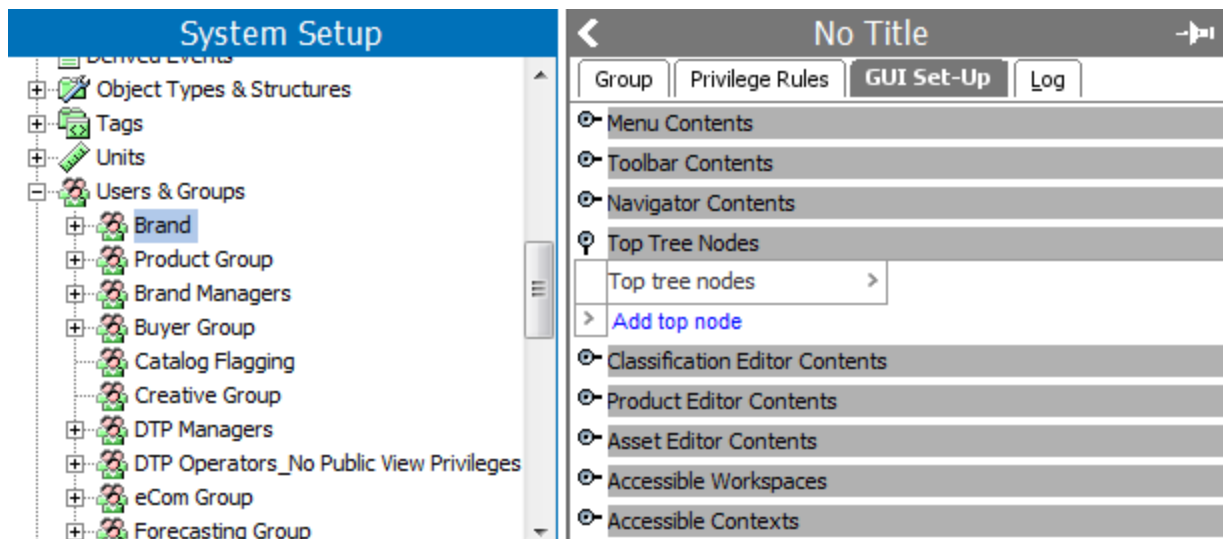
GUI Set-Up is maintained in [System Setup]: **Users & Groups > Groups or Users**

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

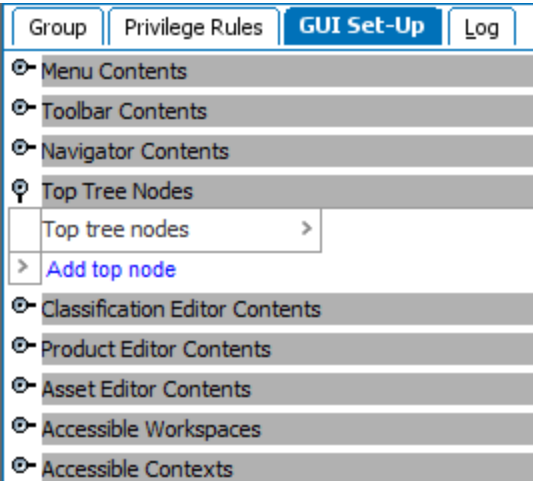
The GUI settings at the user group level applied to all the users that get created under it.

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.

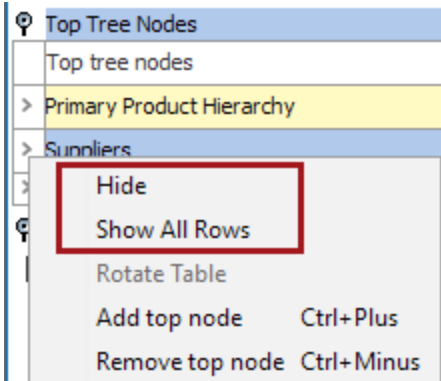
- 1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group.
A **Group** Editor appears.

- 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

Set-Up Options	Description
	'Advanced'. 'Advanced' will typically be selected for those menu items that are rarely used. For easy accessibility, 'Enable All' and 'Disable All' buttons are available.
Toolbar Contents	In the field 'Toolbar Contents' it is possible to customize the icons or menus to be shown in 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 For easy accessibility, 'Enable All' and 'Disable All' buttons are available.
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 • Product Introduction Navigator • STEP Workflow 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. For easy accessibility, 'Enable All' and 'Disable All' buttons are available.
Top Tree Nodes	In the field 'Top Tree Nodes' it is possible to customize the tree nodes to be shown in the Tree Navigator.

Set-Up Options	Description
	You can select any top node for Classification, Products, Collections etc. Once a top node is selected, if you want to avoid other users from making changes to it, you can 'Hide' the Selection. To make it visible again, select 'Show All Rows.' 
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 Contents	In the field 'Asset Editor Contents' it is possible to customize the fields and tabs to be shown in the Asset Editor.
Accessible Workspaces	In this field it is possible to limit the number of Workspaces that can be viewed and accessed.
Accessible Contexts	In this field it is possible to limit the number of Contexts that can be viewed and accessed.

Note: STEP Privileges are different from GUI Set-Up. GUI Set-up applies only on workbench and not on Web UIs or even web services. If the user has privileges to access a product hierarchy which is restricted by the GUI Set-up, he can still access the same via Web UIs or web services.

GUI Setup for Users

GUI Set-Up for Users is maintained in [System Setup]: **Users & Groups**.

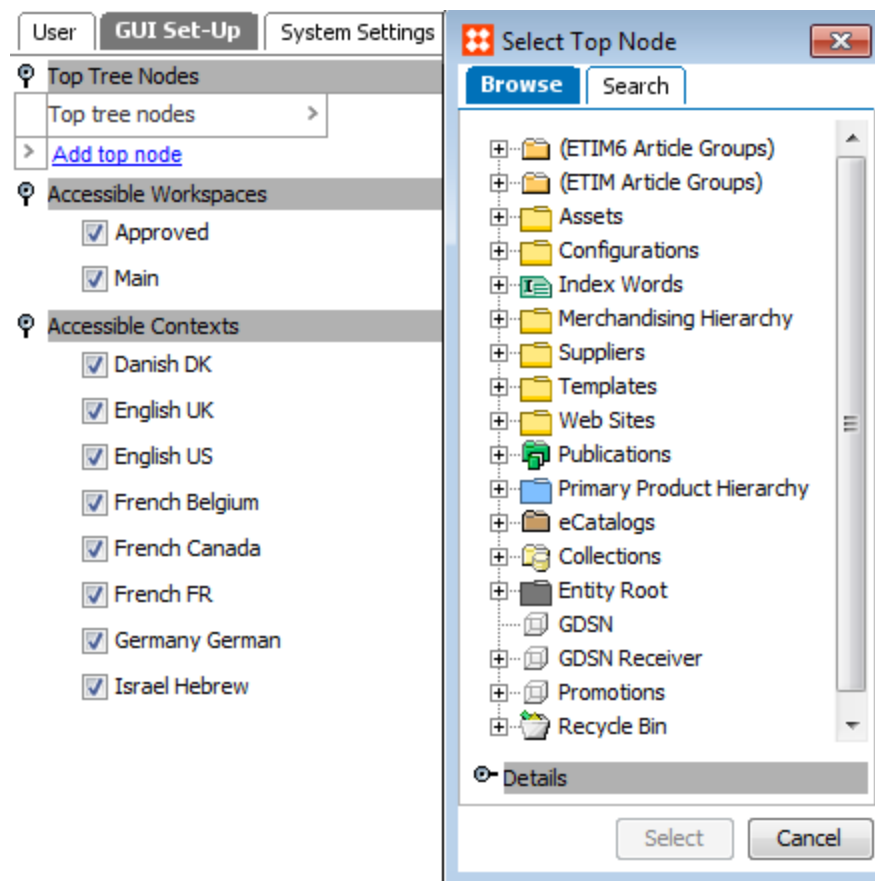
GUI Set-Up for a User makes it possible to customize top tree nodes for the user and provides an administrator with a way to make relevant workspaces and contexts accessible.

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.

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. Web UI 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.

Target (group being impersonated)

Action set with Impersonate User privilege

Source (group that can impersonate)

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 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 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.

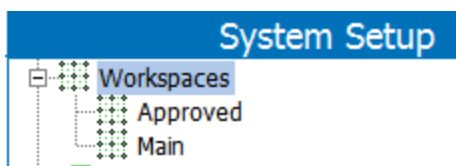
Note: Impersonation can only be used in Web UI.

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, and so forth are valid for all workspaces. Objects that are not subject to approval appear 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, however, 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 one.

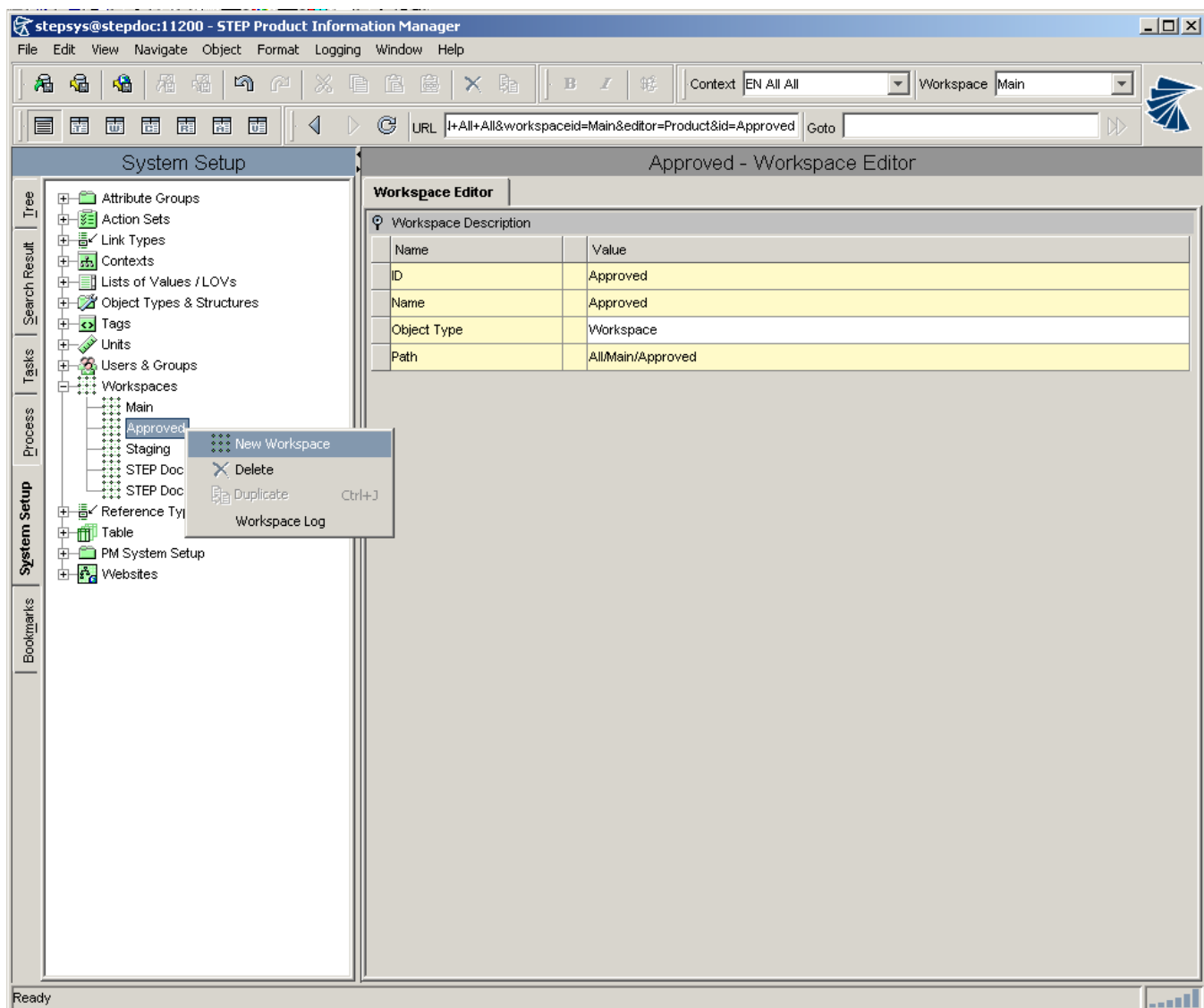
Note: Typically, new workspaces should not be created as the initially set up workspaces (Main and Approved) will be sufficient for the cycle of data maintenance.

For more information, see the **Workspaces** documentation.

1. In System Setup, unfold **Workspaces**, and then click the Workspace of which you need create another workspace.

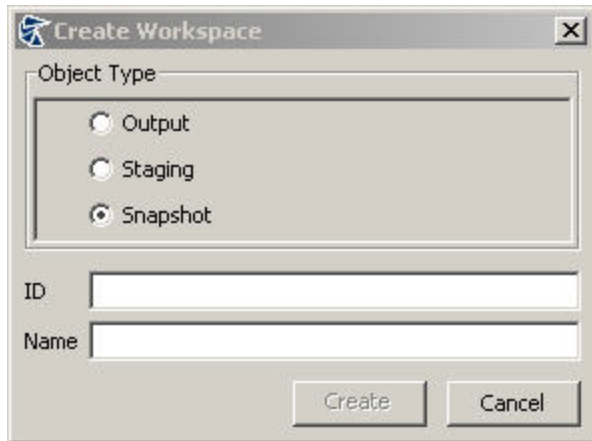
In this example, a Snapshot workspace is created.

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.

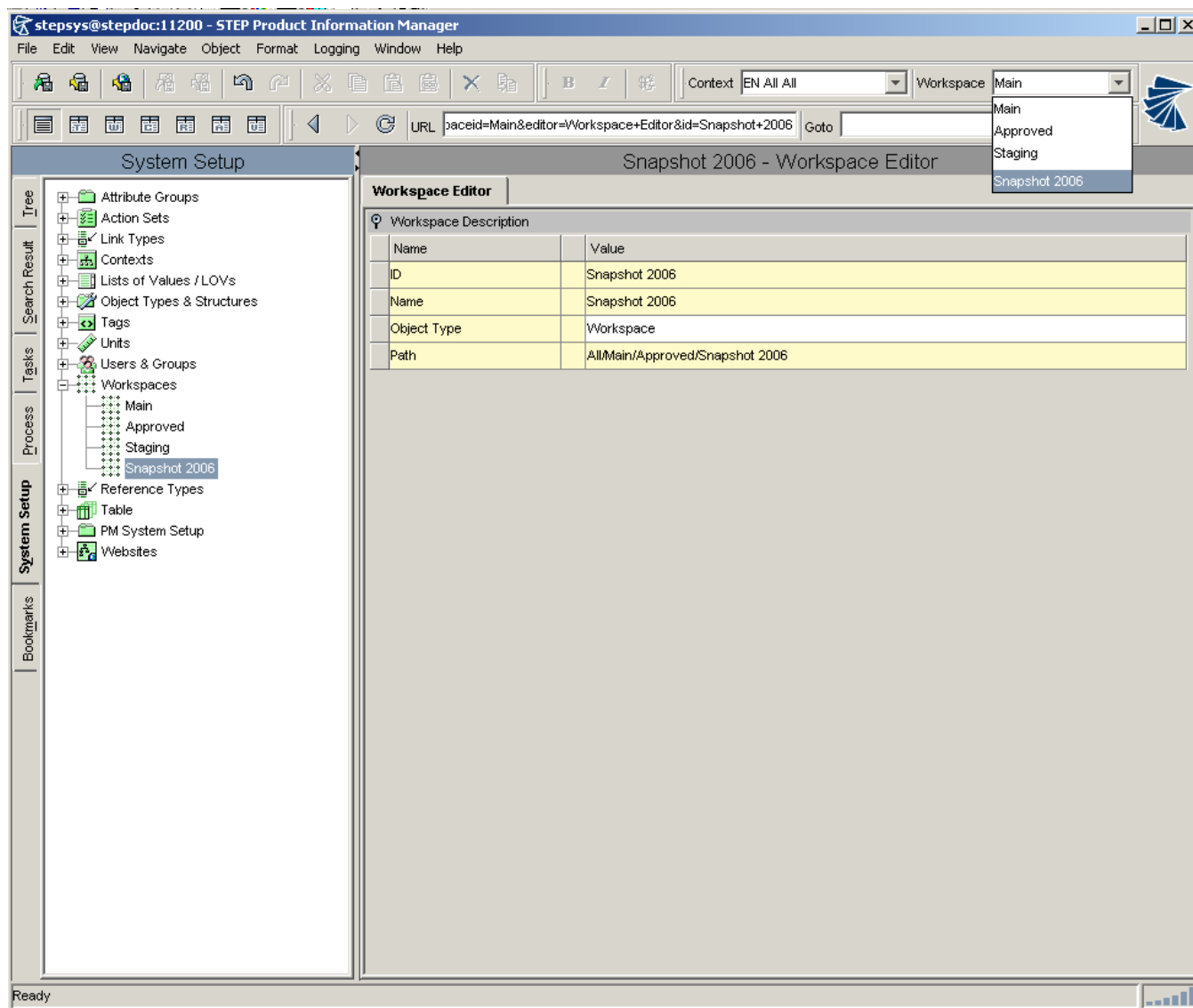


3. Click **Snapshot**.

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 workbench.



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.

Deleting Workspaces

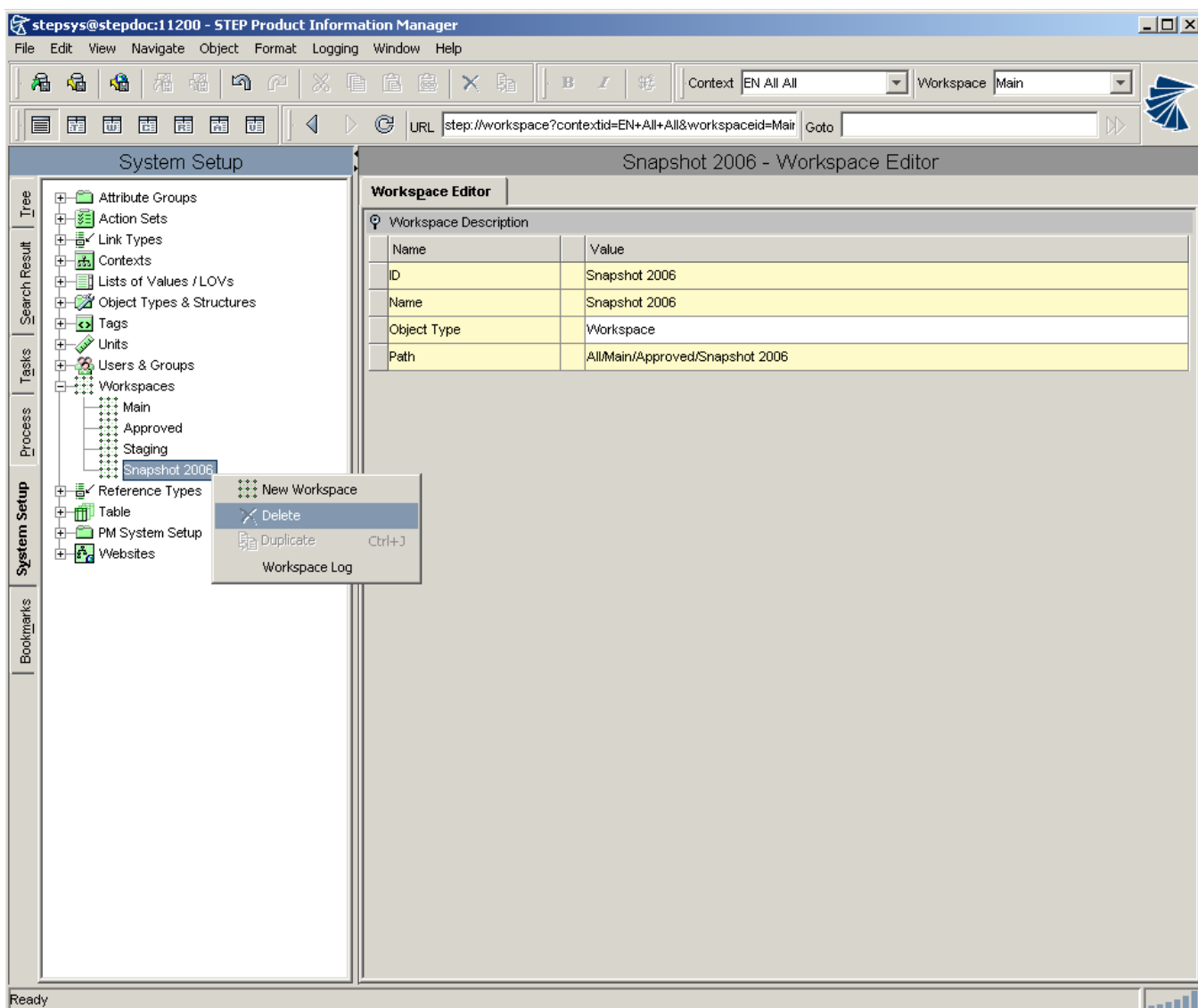
Workspaces are maintained in System Setup > **Workspaces**.

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

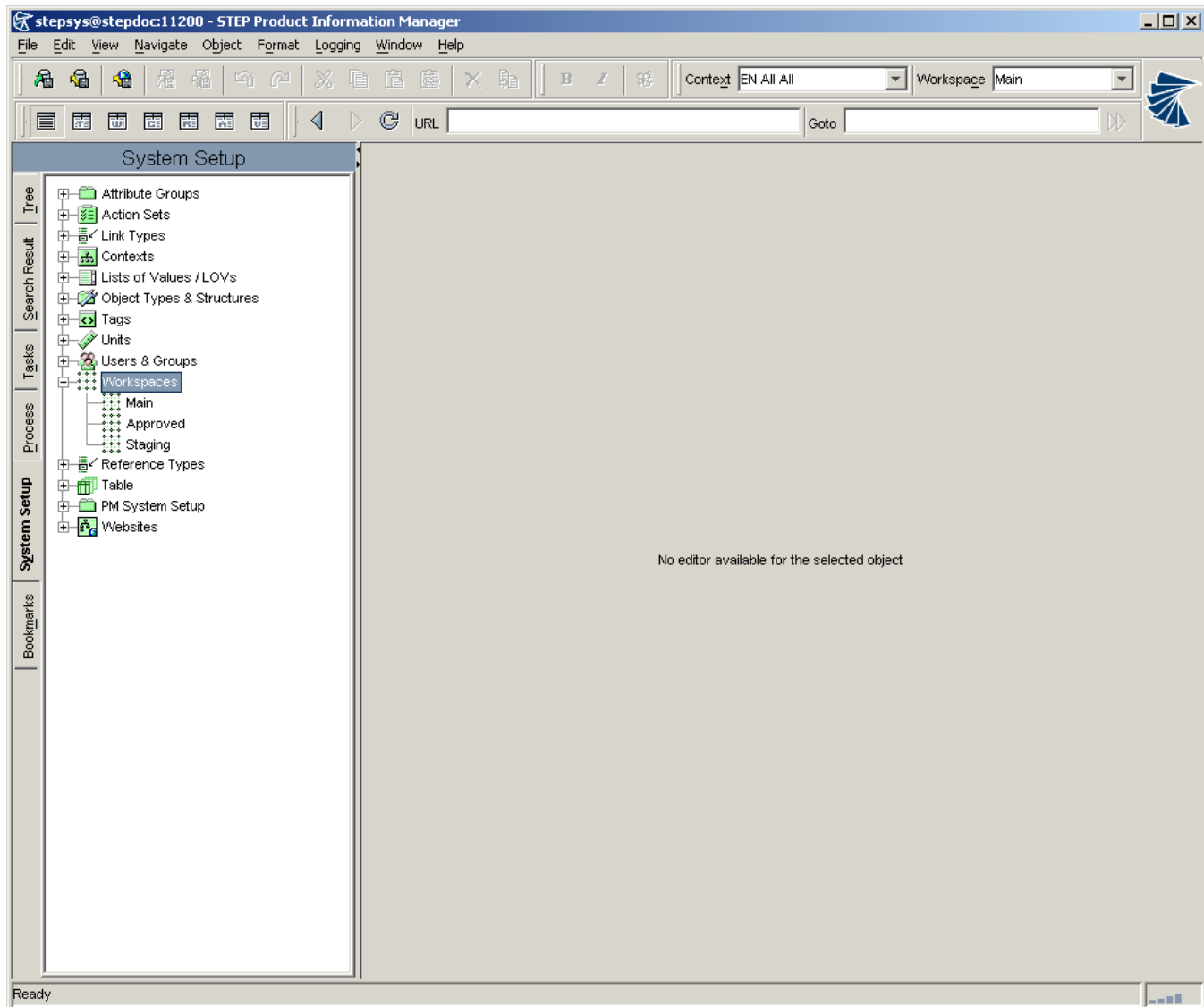
Note: Deleting a Workspace is very time consuming.

1. In System Setup, unfold **Workspaces**, and then click the Workspace to be deleted.
2. Right-click, and then click **Delete**.



3. A **Delete Warning** dialog box appears. 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.
