

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 cluster of three dots above it. 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 8.2

Table of Contents

Table of Contents	2
Introduction	19
About This Guide	19
Accessing Shortcut Menus on PC and Mac	19
System Setup	20
Action Sets	21
Recommendations for Action Sets	22
Setup Actions	23
Action Sets	23
Administration	23
Assets	23
Attributes	24
Background Processes	26
Bulk Updates	26
Business Rules	26
Change Packages	27
Contexts	27
Cryptographic Keys	27
Data Quality	27
Deduplication	28
Events	28
GDSN Data Pools	28
Integration Endpoints	30
Link Types	30
List of Values (LOVs)	31

Object Types	32
Print Publication	33
Purging	33
Reports	34
SDK / API Documentation	34
Tables	34
Tags	34
Units	35
Users & Groups and Privilege Rules	35
Value Search	36
Web UI	36
Workflows	36
Workspaces	38
User Actions	39
Assets	39
Collections	41
Commercial Lists	42
Data Quality Functionality	42
eCatalog	43
GDSN Data Pools	43
Products, Classifications, and Entities	44
Publications	51
Translations	55
Web UI	55
Maintaining Action Sets	56
Creating an Action Set	56
Adding Actions to an Action Set	57

Removing an Action from Action Set	59
Attributes	60
Attribute Display Sequence	61
Attribute Display Sequence Setup	61
Sorting Attributes	63
Selecting a Personal Attribute Display Sequence Attribute	65
Display Sequence Attribute and the Product Editor Display	66
Display Sequence Attribute and Classifications	67
Setting the Priority of a Product Variant Attribute	67
Display Sequence Attribute and Tables	68
Setting the Attribute Display Sequence on Publication (Green) Hierarchy Objects	69
Attribute Groups	71
Creating Attribute Groups	71
Creating a Customized View	72
Manually Sorting in Attribute Groups	76
Attribute Group Display Sequence	76
Sorting Attribute Groups	76
Attribute Links	77
Where to link an attribute:	77
Which hierarchy levels to link to:	77
Which product object types to link to:	77
Attribute inheritance:	77
Linking Specification Attributes	79
Linking Attributes in Product editor	79

Linking Attributes to Products or Classifications	80
Linking to a Source from Attribute	82
Linking Orphan Attributes	83
Sorting Linked Attributes	87
Setting a Display Sequence Number on the Attribute	87
Setting a Display Sequence Number under Linked Attributes	88
Setting the Priority of a Product Variant Attribute	90
Setting the Attribute Display Sequence on Publication (Green) Hierarchy Objects	91
Attribute Transformations	92
Attribute Transformation Options	93
Creating an Attribute Transformation	100
Create an Attribute Transformation Group ..	100
Create an Attribute Transformation	101
Configuring an Attribute Transformation ..	102
Calculated Attributes	107
Calculated Attribute Considerations	109
Deciding to Use a Calculated Attribute	110
Setting up a Calculated Attribute	112
Using Variables in a Calculated Attribute ..	115
Optimizing Calculations Using STEPXML ..	119
Limit Performance Penalties for Calculations	120
Calculated Attribute Use Cases	122
Introduction to Value Templates	123
Attribute Examples	126

Commercial Examples	129
Conditional Examples	131
Date Examples	135
List Examples	137
Number Examples	141
Other Examples	144
Classification Functions	144
Completeness Functions	145
Hierarchy Functions	145
Lookup Table Functions	148
Object Functions	149
Override Functions	150
Reference Functions	151
Revision Functions	152
Workflow Functions	153
Text Examples	154
Escape Functions	154
Identify Functions	154
Modify Functions	155
Regular Expression Functions	158
Representative Text Functions	160
Test Functions	160
Variable Examples	162
Concatenate Function with Variables	163
Find Function with Variables	164
Asset Push File Name Scenario	165
Solution	166

Classification Scenario	168
Get All Classifications	168
Get Classifications Based on Type	168
Get Metadata Value Located on the Link From Classification-to-Product	169
Get CPL Classifications Using Classification Product Links	170
Commercial Data Scenario	172
Price List in a Publication Structure	172
Price List in an eCatalog Structure	172
Comparing Localized Values Scenario	173
Solution	173
Identifying Child Objects Scenario	175
Solution	175
Including Parent Info on Child Scenario	177
Solution Option 1	177
Solution Option 2	178
New Product Indicator Scenario	180
Solution Options	180
Override Scenario	183
Get Overridden By Products	183
Get Overridden Product Objects	184
Get Product Override Parents	185
Product Description Scenario	187
Solution Option 1	188
Solution Option 2	190
Reference Scenario	191
Get List of References	191

Get Value From a Referenced Product	192
Get Metadata Value Located on the Link From Product-to-Product	193
Get Referenced By Using Reference Owner	194
Get Reference Type ID	195
Get Reference Source or Target Product Information on the Link	197
Total Cost of Materials Scenario	199
Solution	200
Units Scenario	202
STEP Functions	205
Comments	206
Use Cases	206
Comparing STEP and Excel Functions ...	207
Attribute Functions	208
Use Cases	209
Commercial Functions	210
Use Cases	211
Conditional Functions	212
Use Cases	212
Date Functions	213
Using a Locale	213
Use Cases	213
List Functions	214
Use Cases	216
Number Functions	217
Using a Locale	218
Use Cases	218

Other Functions	219
Classification Functions	219
Completeness Functions	220
Data Container Functions	220
Hierarchy Functions	221
Lookup Table Functions	222
Object Functions	222
Override Functions	223
Packaging Hierarchy Functions	223
Reference Functions	224
Revision Functions	225
Workflow Functions	226
Text Functions	227
Escape Functions	227
Identify Functions	227
Modify Functions	228
Regular Expression Functions	229
Representative Text Functions	230
Test Functions	231
Variable Functions	233
Use Cases	233
Function Editor	234
Attribute Editor	234
Bulk Update Wizard	237
Flatplanner	238
Using Function Editor	239
Inserting Attribute IDs in Functions	242

Inserting Templates	243
Creating Function Templates	244
Evaluating Functions	247
Creating a Calculated Attribute	248
Overriding and Restoring Calculated Attribute Values	254
Override a Calculated Attribute	254
Restrict Overrides of a Calculated Attribute Value	255
Remove Override Values	256
Restore a Calculated Value	256
Conditional Attribute Display	257
Terminology	257
Creating the Conditional Attribute	257
Activating a Conditional Attribute	259
Configuring Conditionally Valid Attributes ...	260
Description Attributes	260
Specification Attributes	262
Writing Conditional Expressions	263
Conditional Attribute Example	264
Description Attributes	267
Meta Attributes	267
Linking Description Attributes to an Object Type	268
Attribute Metadata on Attributes	272
Create a Metadata Attribute on an Attribute	272
Description Attributes and Object Types	275
Dimension Dependent Attributes	277
Dimension Dependent Attribute Values	277

Non-Dimension Dependent Attribute Values	278
Setting Dimension Dependency on Attributes	279
Value Inheritance in Dimension Dependent Attributes	279
Showing Source Context (Parent Dimensions)	281
Dimension Dependent Attribute and Attribute Group Best Practices	284
Dimension Dependent Attribute Do's	284
Dimension Dependent Attribute Don'ts	285
Which Attributes Need Dimension Dependencies?	285
Dimension Dependent Attribute Group Do's	286
Dimension Dependent Attribute Group Don'ts	286
Using Dimension Dependent LOVs With Attributes	287
Attribute With No Dependency, LOV With a Dependency	287
Attribute With No Dependency, LOV Without a Dependency	288
Attribute With a Dependency, LOV With a Dependency	288
Attribute With a Dependency, LOV Without a Dependency	288
Encrypted Attributes	289
Cryptographic Key	289
Password requirements	291
Creating Encrypted Attributes	291
Properties of attributes that cannot be used with encryption	292

Accessing encrypted data	292
Restrictions of Encrypted Data	293
Handling Duplicated Attributes in Web UI	294
Create Attribute Group(s) to Hold Attributes Known to Exist in Multiples Groups	294
Setting up Blacklisted Attribute Groups in Web UI	294
Maintaining Attributes	296
Attribute Types	297
Attribute Value Migration	299
Prerequisites	299
Technical Migration Details	300
Configuring Attribute Value Migration	301
Starting the Attribute Value Migration	305
Statistics	306
Completing the Migration	308
Converting Attributes from Date to ISO Date	311
Merging Attributes	311
Switching Attribute IDs	312
Creating Attributes	313
Externally Maintained Attributes	318
Revision Control	318
Sharing LOVs	318
Setting as Externally Maintained	318
Full Text Indexable Attributes	321
Hierarchical Filters on Attributes	324
Relation between LOV filters	324
Viewing and editing LOV Filters	325

Options affecting Performance	325
To setup a hierarchical filter:	325
Validity on Description Attributes	329
Validity on Specification Attributes	332
Reparenting Attribute Groups	334
Manual Reparenting	334
STEPXML Reparenting	338
Mandatory Attributes	353
Approval Mandatory Attributes (Global) / Visual Indicator in Workbench	353
Workbench Visual Indicators	353
Approval Mandatory Attributes	356
Business Rules for Mandatory Attributes	357
Smartsheet Mandatory Attributes	357
Web UI Screen Mandatory Attributes / Visual Indicator in Web UI	360
Web UI Visual Indicators	360
Enforcement of Mandatory Attributes	363
Workflow Mandatory Attributes	363
Merging Attribute Values	365
Verify Attributes to be Merged	365
Merge Values Using Bulk Update	368
Single and Multi-valued Attributes	370
Unlinking Attributes from Objects	371
Unlink Object from Attribute References Tab	371
Unlink Attribute from Object References Tab	371
Editing Attribute Values	373
Inherited and Local Values	375

View Filter Options	375
Attributes with Units	375
Editing in 'ISO Data' and 'ISO Date and Time' validation base type	376
Entering Values	376
Overwriting Values	377
Editing Multi-Valued Attributes	378
Specification Attributes	380
Specification Attribute Facts:	380
Inherited Attributes	382
Orphan Attributes	383
Validation Rules	384
Editing Validation Rules	390
Input Masks	393
Example Input Mask Use Cases	394
Definitions of Input Mask Characters	394
Static Mask Characters	396
Setting LOV Filters	397
Setting LOV Filter	398
Viewing Attribute Values	401
Background Process Queues	403
Prerequisites	403
Default Configuration	403
Changing the default configuration	404
Finding the Process Type ID	404
Editing the configuration file	406
Deleting Background Processes	406
Configuring Auto-Deletion of Background	406

Processes	
Retention of Background Processes	407
Auto-Deletion Timings	407
Deleting Background Processes Manually	408
Scheduled Process Properties	409
Data Containers	411
Composite Concept	411
Using Data Containers	411
Considerations and Limitations	412
Setting Up Data Containers in Workbench	413
Prerequisites	413
Adding and Configuring Data Container Types	414
Adding and Maintaining Data Containers	422
Dimensions, Dimension Points, and Contexts	424
Dimensions and Dimension Points	425
Dimensions	425
Dimension Points	425
Dimension Dependent Data	426
Dimension and Dimension Point Best Practices	428
Dimension Do's and Don'ts	428
Dimension Do's	428
Dimension Don'ts	428
Additional Attributes as an Alternative to Additional Dimensions	428
Language Dimension	429
Country Dimension	430

Business Unit Dimension	431
Market Dimension	431
Dimension Point Do's and Don'ts	432
Dimension Point Do's	432
Dimension Point Don'ts	433
Maintaining Dimensions and Dimension Points	434
Creating a Dimension	434
Creating a Dimension Point	436
Deleting a Dimension Point	437
Deleting a Dimension	438
Selecting a Dictionary for a Language Dimension Point	439
Adding a Dimension Dependency After Loading Data	440
Add a Dimension Dependency to an Attribute After Data Has Been Loaded	440
Add a Dimension Dependency to a Reference Type After Data Has Been Loaded	441
Contexts	442
Dimension Dependent Data	443
Context Best Practices	445
Context Do's	445
Context Don'ts	445
Maintaining Contexts	447
Creating a Context	447
Editing a Context	449
Deleting a Context	449
Hidden Values	449

Specifying a Default Context	450
Context Locales	453
Locale Do's	454
Events	455
Core Events	456
Prerequisites	456
Core Events and Event Types	456
Core Event Generation	457
Using Core Events	459
Change Flags for Events	459
Change Flag Example	462
Derived Events	466
Prerequisites	466
Add a Derived Event Type	466
Delete a Derived Event Type	467
Event Processors	469
Prerequisites	469
Setup Requirements	469
Event Processor Results	470
Processing Plugins	471
CASS Certification Report Processing Plugin Parameters and Triggers	473
Prerequisites	473
Parameters	474
Event Triggers	475
Execute Business Action Processing Plugin Parameters and Triggers	476
Prerequisites	476

Parameters	476
Event Triggers	478
Expand LOV Value Changes Processing Plugin Parameters and Triggers	480
Prerequisites	480
Parameters	480
Event Triggers	481
For Use with External Systems	482
Experian Email Validation Processing Plugin Parameters and Triggers	483
Prerequisites	483
Parameters	483
Event Triggers	485
Image Cache Processing Plugin Parameters and Triggers	488
Prerequisites	488
Parameters	489
Event Triggers	490
Matching Processing Plugin Parameters and Triggers	492
Prerequisites	492
Parameters	492
Event Triggers	494
Revision Management Processing Plugin Parameters and Triggers	496
Prerequisites	496
Parameters	496
Event Triggers	497
Initial Setup for Event Processors	499
Prerequisites	499

Create Setup Group Type for Event Processors	499
Link Event Processor Object Types to Setup Group Types	502
Create an Event Processor Setup Group	503
Configure Action Sets and Privileges	505
Create Event Processor Action Sets	505
Configure Event Processor Privileges	505
Creating an Event Processor	507
EPW - Identify Event Processor	509
EPW - Configure Event Processor	510
EPW - Configure Processing Plugin	512
EPW - Schedule Event Processor	514
EPW - Configure Error Reporter Processing Plugin	516
Running an Event Processor	519
Event Processor Status Options	519
Enable Event Processor	520
Invoke Event Processor	521
Disable Event Processor	522
Maintaining an Event Processor	524
Edit Event Processor	525
Delete Event Processor	526
Event Processor Tab	527
Prerequisites	527
Description Flipper	527
Configuration Flipper	528
Current Background Process Log Flipper ...	531
Event Processor Queue Status	533

Event Triggering Definitions Tab	534
Prerequisites	534
Event Triggering Definitions	534
Background Processes Tab	536
Prerequisite	536
Event Processor Background Processes	536
Background Processes in Different States	537
BG Processes Tab	537
Statistics Tab	539
Prerequisite	539
Statistics Flipper	539
Error Log Excerpts Tab	540
Prerequisites	540
Event Processor Log Flipper	540
Log Tab	541
Prerequisite	541
Logs	541
Monitoring an Event Processor via External System	542
Event-Based Event Processor Queued Events	543
Estimate Unread Events	543
View First Event Batch	543
List of Values (LOVs)	545
LOV Features	546
Values by Reference	546
Revision Control	546
Validation Base Type	546

Working with Large LOVs	546
Creating an LOVs Group	547
Creating an LOV	549
Adding Values to an LOV	555
Adding a Single LOV Value	555
Adding a Group of LOV Values	556
Exporting the Existing LOV to Create a STEPXML File	556
Modifying the File to Include the New Values.	559
Importing New Values Into STEP	562
Adding IDs to Values in LOV	565
Auto-generate IDs option	566
Import File option	567
Merge LOVs	572
Merging LOVs	572
Merge Values in LOV	572
Merging LOVs	573
Merging Values in an LOV	577
Prerequisites	577
Steps to Merge Values in an LOV	578
Merging LOV Values in Bulk	582
Prerequisites	582
Creating and Formatting the CSV File	582
Uploading the Merge LOV Values CSV Sheet	584
Monitoring LOV Value Merge Events	586
Deleting LOV Values in Bulk	587
Prerequisites	587

DeleteValue Tag and XML Attributes	587
Deleting LOVs	589
Initiating and Removing an LOV from Workflow	591
Steps to Initiate an LOV to a Workflow:	591
Steps to Remove an LOV from a Workflow:	593
Managing Revisions in STEP	594
Generating Revisions	594
Important considerations for revisions:	596
Setting the Timing Threshold for Automatic Revisions	596
Reverting to Past Revisions	597
Purging Revisions	598
Revision Maintenance and Cleanup	598
Comparing Revisions	600
Object Types and Structures	603
Attributes Valid for Object Types	607
Setting Validity	607
Autogenerated Using Name Pattern and ID Pattern	609
Defining the name and ID Pattern	609
Basic Object Types	612
Object Type Dimension Dependencies	615
Object Type Icons	616
Editing an Object Type Icon	616
Maintaining Object Types	619
Creating an Object Type	620
Steps to Create an Object Type	620
Deleting an Object Type	622

Unlinking Object Types	625
Updating Object Types from Another Object Type	627
Steps to Update an Object Type from Another Object Type	627
Maintaining Dimension Dependent Object Types	631
Making an Object Type Dimension Dependent	631
Removing Dimension Dependency from an Object Type	632
MIME Types	635
Setting MIME Types for Object Types	637
Editing MIME Types LOV	638
Modifying Manually Sorted	640
Object Types	642
Object Type Hierarchy	644
Classification Hierarchy Structures	645
Fixed Hierarchy Structure	647
Variation in Hierarchy Depths	648
Multiple Parents	649
Creating a Standard Publication Object Type Hierarchy	650
Creating a Publication Group Object Type	652
Creating a Publication Object Type	654
Creating a Publication Section Object Type	656
Product-Override Object Types and Commercial Object Types	658
Creating a Commercial List Type	659
Creating a Product-Override Object Type	662

Enabling a Product-Override	663
Owns Product Links on Alternate Classifications Object Type	664
Owns Product Links Affects STEPXML	665
Changing Owns Product Links	669
Steps to Change the Owns Product Links	669
Reference Target Lock Policy on Object Types	671
Revisability on Entity Object Type	673
Product Variants	675
Setting Up Product Variants	676
Add the object type to the component model	676
Set up a meta attribute	678
Product Variant Family Attributes	679
Specify variant and non-variant attributes	681
Using Variants List View	682
To view product variants in list view	682
To delete a variant	682
Product variants list in the Web UI	683
Using the Variants Matrix View	684
To view product variants in matrix view	684
To delete a variant	686
To create a new variant	686
Creating Variants Through Web UI	688
Business Condition: Validate Product Variant	691
Recycle Bin	693
Reference and Link Types	695

References	695
Product to Classification Links	696
Product Attribute Links	697
Classification Attribute Links	697
Classification Attribute Link Type	698
Creating a Classification Attribute Link	700
Setting Up an Attribute Link Type	703
Maintaining Attribute Link Metadata Values	706
Inherited Metadata Values	706
Local Metadata Values	707
Product Attribute Link Type	710
Creating a Product Attribute Link	712
Setting Up an Attribute Link Type	715
Maintaining Attribute Link Metadata Values	718
Inherited Metadata Values	718
Local Metadata Values	719
Product to Classification Link Types	722
Creating a Product to Classification Link Type	724
Product to Classification Link Type - Enter ID and Name	726
Product to Classification Link Type - Select Valid Product Types	727
Product to Classification Link Type - Select Valid Classification Types	728
Product to Classification Link Type - Apply Dimension Dependencies	729
Product to Classification Link Type - Advanced	730

Creating a Product to Classification Link	733
Maintaining a Product to Classification Link Type	736
Product to Classification Link Type Tab	736
Validity Tab	740
Log Tab	742
Attribute Groups for Reference and Link Types	743
Dimension Dependent Reference and Link Types	746
Dimension Dependency on the 'Reference-Type' Object Type	746
Dimension Dependency on a Reference Type	747
Inheritance Example for a Product to Classification Link Type	749
Metadata Attributes on Reference and Link Types	758
Add a Metadata Attribute	758
Edit a Metadata Attribute	760
Visibility of Reference and Link Types	762
Reference Types	766
Creating a Reference Type	768
Reference Type - Enter ID and Name	770
Reference Type - Select Valid Source Types	771
Reference Type - Select Valid Target Types	772
Reference Type - Apply Dimension Dependencies	773
Reference Type - Advanced	774
Creating a Reference	776

Maintaining a Reference Type	779
Reference Type Tab	779
Validity Tab	784
Log Tab	786
Attribute Groups for Reference and Link Types	787
Direction of a Reference	790
Dimension Dependent Reference and Link Types	791
Dimension Dependency on the 'Reference-Type' Object Type	791
Dimension Dependency on a Reference Type	792
Entity Reference Types	794
Parent/Child Relation Example	795
Inheritance Example for a Reference	797
Metadata Attributes on Reference and Link Types	800
Add a Metadata Attribute	800
Edit a Metadata Attribute	802
Multiple References for the Same Objects	804
Visibility of Reference and Link Types	805
Regular Expression	809
Versions	809
Regular Expression Syntax	810
Characters	810
Repetition Modifiers	812
Groups and Ranges	812
Escaping Special Characters	813
Anchors	813

Regular Expression Examples	815
Asset Importer	815
Attribute Transformations	816
Import Transformations	817
Search	818
Table Transformations	819
Validation Base Type	820
Setup Groups	822
Creating Setup Group Object Types	823
Defining Type of Objects to be Stored in Setup Groups	825
Creating Setup Groups in System Setup	828
Setting Up Privileges Example	830
Privilege Setup Steps	831
Single-Update Mode	836
Operations that Cause Single-Update Mode	837
Tags	839
Tag systems in STEP	839
Types of Tags in STEP	839
Working With Tags	841
Creating Tag Groups and Tags	842
Tag Groups	842
Tags	843
Configuring / Rendering Tags	848
Special Consideration When Rendering Tags:	850
Configuring Output Formatting	852
Output Formatting Flipper	854

'Pass Through' Setting	856
Tag Examples	857
Inserting a Tag	857
Unique Keys	861
Defining Keys	861
Key Example	862
Key Formula Example:	862
Key Formula Example	862
Important Notes:	863
Creating and Deleting Keys	864
To Create a Key	864
To Delete a Key	866
Activating and Deactivating Keys	868
Activate a Key	868
Deactivate a Key	869
Checking Keys	871
Creating Objects with Active Keys	872
STEP XML	872
Web Services APIs	873
Web UI Simple Importer	874
Workbench	874
Matching Keys During Import	875
Units	876
Unit Groups	877
Unit Conversions	877
Maintaining Unit Groups	879
Creating a Unit Group	879

Creating a Unit	880
Setting up a Unit Conversion	881
Removing a Unit Conversion	882
Removing a Unit or Unit Group	883
Users and Groups	884
Working with User Groups	886
Creating a User Group	886
Deleting a Group	887
Working with Users	888
Creating a User	888
Adding a User to a Group	889
Duplicating a User	890
Removing a User from a Group	891
Changing a User Password	893
Changing a User Password - System Administrator	893
Changing a User Password - User	893
From the Workbench:	894
From the Web UI:	894
System Settings	896
Classification Hierarchy Settings	897
Dimension Dependencies	897
Image and Document Settings	900
Dimension Dependencies	900
Store assets and DTP documents in	901
DTP asset source	901
Pregenerate thumbnail cache on upload	901
Disable auto-cleanup of thumbnail cache	901

Transformation Lookup Tables follow asset dimension dependency	901
Asset Import Compatibility Mode	902
Asset push attribute macro keys	903
Calculated Attribute Settings	904
Cryptographic Keys	905
Creating cryptographic keys	905
Password Requirements:	906
Validation Templates	908
To create a Validation Template:	908
Product Information Manager Default Settings	910
Enforce Mandatory Check for Attributes, References and Links	910
Product Editor, Group attributes by top group	911
Localize numbers with thousand delimiter when localizing exports	912
Localize dates when localizing exports	913
Report Save As CSV Character Set	913
Default Attribute to use as Display Sequence Attribute	914
Default Completeness Metric	915
Conditional Validity Attribute	915
Block Attribute Groups with more than 1000 attributes	915
Use full pathname for classes on Product References Tab	915
Pass through unconverted Special Characters and Tags (Y) or discard them on output (N)	916
Product Attribute Help metadata attribute ...	916

Attribute indicating the Priority of Product Variant Attributes	917
Show both name and ID in the PIM navigator	917
Comma separated list of node types for which to show both name and ID	917
Attribute Header Column Width	918
Reference Type Column Width	919
Translation Settings	920
Translation File Pattern	920
Should "Filter spreadsheet" option be shown in Translation Wizard.	920
Default value for "Filter spreadsheet" option in Translation.	921
Terms List Settings	922
Add the products from the selection list import file directly instead of looking up leaf products	922
Web Services	923
Tag format for Web Services and REST APIs	923
Web UI Settings	924
Default context	924
Default workspace	924
Web UI supplier classification object type	925
Web UI supplier products classification object type	925
Assets classification object type	926
Batches classification object type	926
Batch classification object type	926
Excel-template asset	927
Proof view stylesheet attribute	927

Step-users Web UI batches folder	927
Step-users Web UI assets folder	927
Context Help metadata attribute	927
Link type for vendor classification to product link	928
Enable all-view for users that are a member of multiple suppliers	928
Primary Image Type	929
Flatplanner Default Settings	930
Enable keep layouts for publications	930
Show frame information on flatplans with layout pickup	931
Reapply Flatplan rules when adding product to non-empty frame	932
DTP Settings	933
Use Publication Name prefix when naming DTP documents	933
Enable calculation of border rectangle space usage for actual pages	933
Calculated DTP Table width includes left and right rules	934
Attribute Display Sequence	934
Attribute Name Style Meta-attribute and Attribute Value Style Meta-attribute	934
Asset Order Meta-attribute	935
Table Cell Skip Empty Cell	935
JPEG DPI	935
DTP Page - preview image	935
Inherit Default Product Template through product hierarchy for Product Overrides.	936
Table Defaults	937

Index Word Settings	938
Revisability Settings	939
Threshold (hours) for changes by same user to generate a new revision	939
Business Rule Approve Overview	940
Security Policy	941
GUI Localization	943
Setting up a GUI Localization for a User	944
Privilege Rules	946
Adding User Privileges for a Group	950
Add Privilege Rule Options	951
Adding Setup Privileges for a Group	956
Add Setup Privilege Rule Options	956
Editing Privilege Rules	959
Edit Privilege Rule Options	960
Removing a Privilege Rule from a Group	965
Configuring Workflow Privileges	966
Workflow state selector	966
Editing Privilege Rules with focus on STEP Workflows	967
Limitations on Workflow privileges	969
Object Creation Privileges	972
Considerations When Creating New Object	973
Mass Creation Screen in Web UI	973
Smartsheets	973
Importer	973
Privileges and Security	973
Granting Access to the SDK and API Documentation	975

Grant Access to the SDK and API documentation	975
GUI Setup	978
GUI Setup on Groups	979
GUI Setup for Users	982
Web UI User Impersonation	984
Workspaces	986
Standard Workspaces	987
Maintaining Workspaces	988
Creating Workspaces	988
Workspace Log	990
Deleting Workspaces	991

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 Shortcut 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.

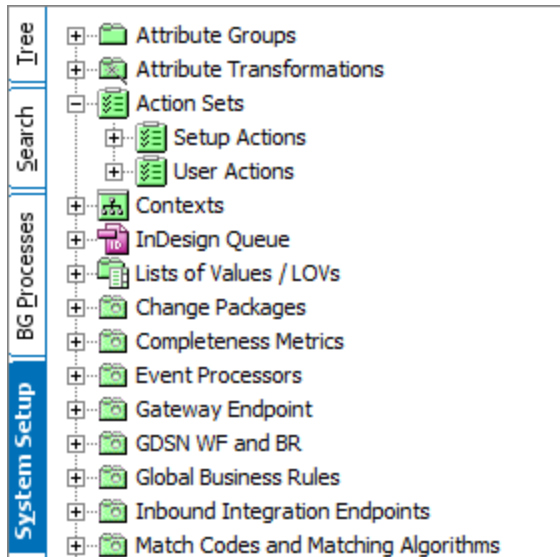
System Setup

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, see the **About Users and Groups** section of **Users and Groups** documentation.

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	Ability to remove (unlink) attributes that are linked to other attribute groups as well

Action	Privileges Granted
attribute group	as add attributes to a group.
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.

Action	Privileges Granted
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

Action	Privileges Granted
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)
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

Action	Privileges Granted
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

Action	Privileges Granted
	Note: In order to apply description attributes to link types, these actions are also needed: • View attribute • View attribute group • Modify valid (node) types for product attribute

List of Values (LOVs)

Action	Privileges Granted
Create domain	Ability to create LOVs Note: The actions 'Modify name/description of domain (also translate)' and 'Modify definition of domain (validator, etc.)' must also be included in order to create LOVs.
Create value in hard domain	Create values in LOVs where users are not allowed to add values ('hard' LOVs)
Create value in medium domain	Create values in LOVs where users are allowed to add values ('medium' LOVs)
Delete domain	Ability to delete LOVs
Delete value in hard/medium domain	Delete values in both types of LOVs: • Where users are not allowed to add values ('hard' LOVs) • Where users are allowed to add values ('medium' LOVs)
Maintain domain groups	Create and delete LOV groups
Merge entire domain	Ability to merge LOVs
Merge value in hard/medium domain	Merge values in both types of LOVs: • Where users are not allowed to add values ('hard' LOVs) • Where users are allowed to add values ('medium' LOVs)

Action	Privileges Granted
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 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.

Action	Privileges Granted
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	Ability to delete obsolete revisions:

Action	Privileges Granted
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 and the STEP API Documentation button on the STEP WebStart page.

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.

Action	Privileges Granted
	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
Access Web UI	Ability to access Web UIs on the system. Access to specific Web UI instances can be granted if a Setup Group is defined in the user (group) Setup Privileges, otherwise, this action allows access to ALL Web UI instances on the server.
Maintain Impersonation	Ability to impersonate another user for Web UI support purposes
Update Web UI configuration	Access to modify and configure any STEP Web UI
View context	In addition to granting access to the STEP workbench, 'View context' also grants the ability to access the Web UI.
Web UI Administration	Grants the ability to: • "Revert" / copy to front • Delete action context menu • Duplicate action context menu • Status tab • Edit Web UI through designer • Edit Web UI through Workbench • Change Web UI ID in the XML

Workflows

Action	Privileges Granted
Disable STEP	Ability to disable auto-initiation when importing data

Action	Privileges Granted
Workflow Auto-initiation in Imports	
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 Administrator	• View other users' tasks • Complete / submit tasks not assigned to the current user • Reassign tasks • Edit task deadlines The action can be applied to a setup group and will then only take effect for workflows in that setup group.
View 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.

Action	Privileges Granted
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 display 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

Action	Privileges Granted
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 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 display 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

Action	Privileges Granted
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

Action	Privileges Granted
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.

Action	Privileges Granted
Approve References	Approve reference links from products, i.e. links from products to products, links from products to assets, and links from products to classifications.
Approve Values	Approve the modification of specification and description attribute values of products and assets: This privilege is also subject to the attribute group setting of Privilege Rules on user groups. For example, if a user only has the privilege to approve attributes in a determined attribute group, this user can only approve certain attributes of a product or asset. Also, approving table values depends on the attribute group setting of the Privileges Rules on user groups.
Classify product (link product to classification)	Link products to classifications
Create classification	Create and duplicate classifications of specified object types Note: The action 'Modify name/description of classification (also translate)' must be included in order to create classifications.
Create classification 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

Action	Privileges Granted
	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
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

Action	Privileges Granted
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 display on all specification attributes that are linked to specified valid product object types.

Action	Privileges Granted
translate)	
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 display 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.

Action	Privileges Granted
	The 'Move entity to' action must also be applied.
Move entity to	Copy and paste entities to another entity node. The 'Move entity from' action must also be applied.
Move product from	Copy / paste and drag / drop products from another product node within the same object type hierarchy. 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

Action	Privileges Granted
	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 display on all specification attributes that are linked to specified valid classification object types.
View metadata for classification (property value)	View values of description attributes linked to a classification
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 display 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 display 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 display on all specification attributes that are linked to specified valid classification object type.
View metadata for	View link attribute values of attributes that have been applied to the link type marker edge

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

Publications

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

Action	Privileges Granted
Create DTP Template Document	Create product templates, page templates, and publication templates (master documents)
Create Generic Page Template	Create page templates
Create Planned Page	Add planned pages into sections

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

Action	Privileges Granted
Maintain Flatplan Content	Change the object assigned to an existing frame
Maintain Flatplan Frame	Add, move, delete, and resize frames
Maintain public basket view	Maintain public Flatlanner basket views
Maintain public financial view	Maintain public Financial Summary views (Flatplanner)
Maintain public 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
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:

Action	Privileges Granted
	• 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
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

Action	Privileges Granted
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 specific to the Web UI.

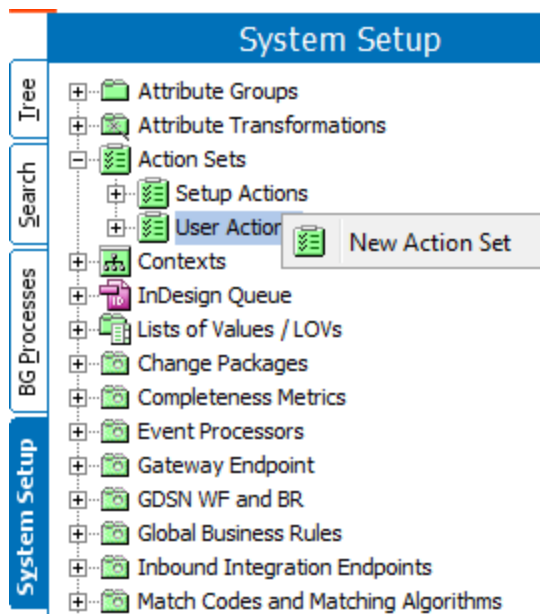
Action	Privileges Granted
Impersonate User	Impersonate a user in a targeted user group in order to troubleshoot issues
Share search with group	Share saved searches with specific user groups
Supplier Web UI functions	Use the Supplier Web UI functions in Supplier Web UI

Maintaining Action Sets

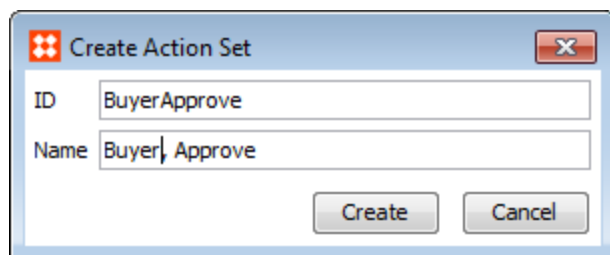
Creating an Action Set

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

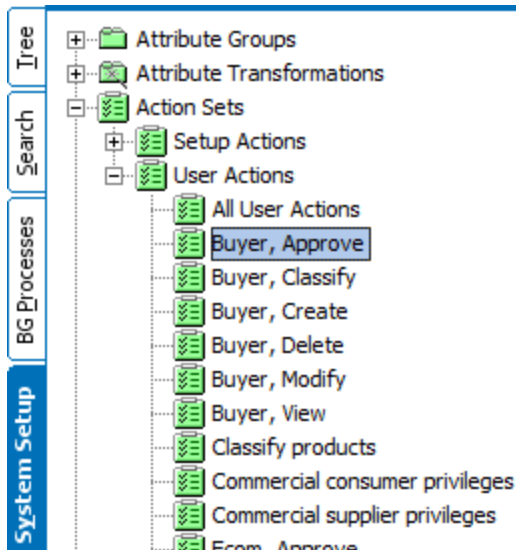
1. In [System Setup], unfold the **Action Set** group and select the relevant main Action Sets folder.
2. Right-click, and then click **New Action Set** (or in the **Maintain** menu, point to **Insert** and then click **Action Set**).



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**.

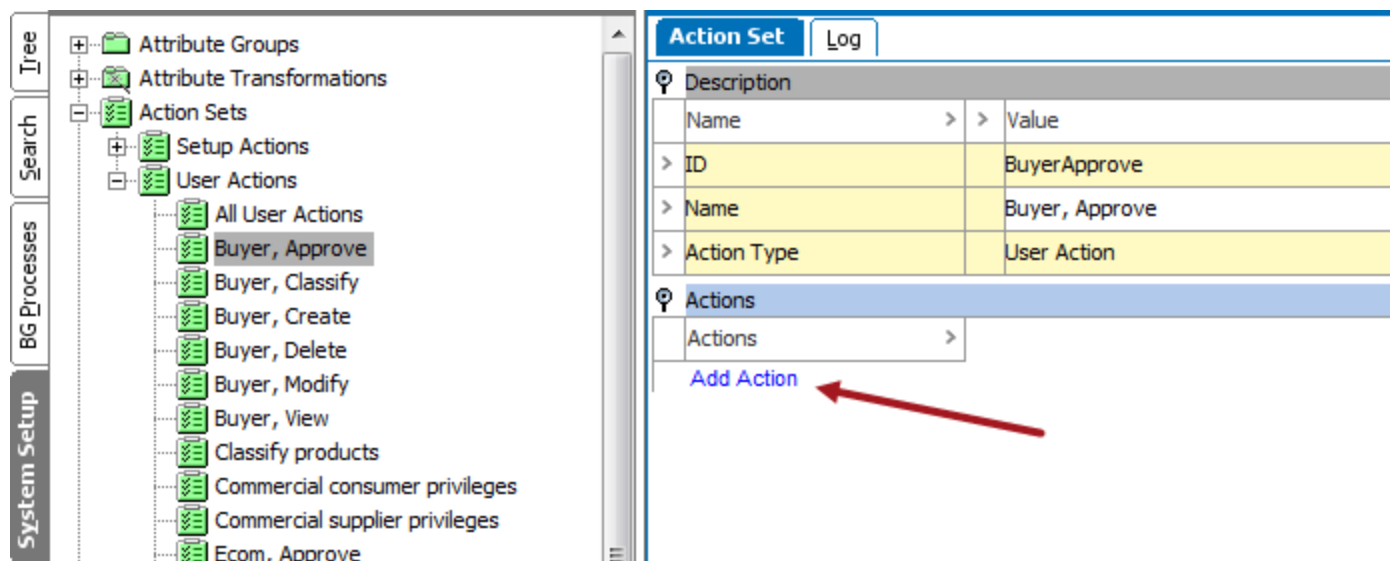


A new Action Set has now been created, and the next step will be to add the specific actions.

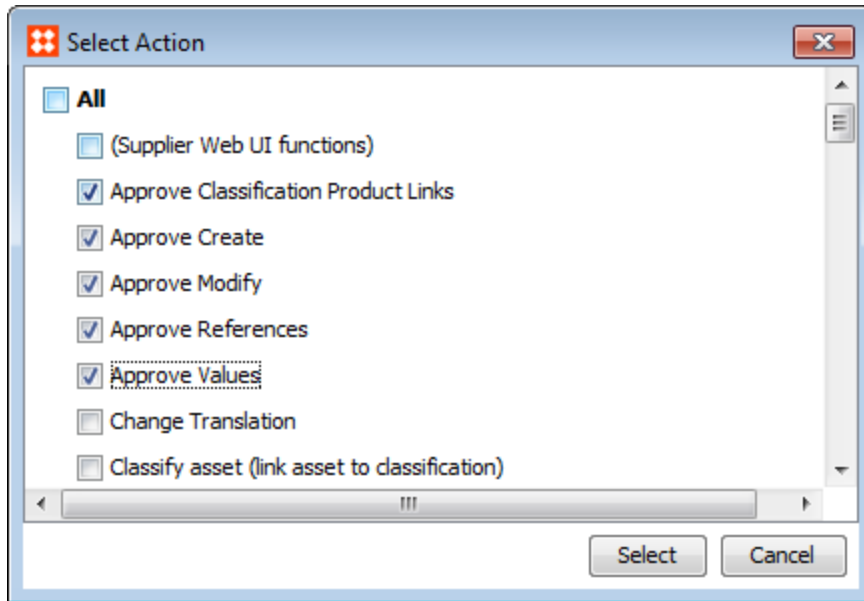
Adding Actions to an Action Set

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**.

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

Actions

- Approve Classification Product Links
- Approve Create
- Approve Modify
- Approve References
- Approve Values

[Add Action](#)

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 'Action Sets' folder is expanded, and 'Buyer, Approve' is selected. The right pane displays the 'Action Set' editor with a table of actions. A context menu is open over the 'Buyer, Approve' row, showing options like 'Hide', 'Show All Rows', 'Rotate Table', 'Add Action', and 'Remove Action'.

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

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

Remove the selected rows(s)

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

A STEP attribute is a characteristic or detailed piece of information related to STEP objects (i.e., Products, Image & documents, Classifications, Link types, Data containers).

There are two types of Attributes: Description and Specification. Both types of Attributes are stored within Attribute Groups. For more information, see the **Description Attributes**, **Specification Attributes**, and **Attribute Groups** sections of this guide.

Attributes and their values can be maintained in Workbench and/or Web UI. For more information, see the **Maintaining Attributes** section of this guide.

Attribute Display Sequence

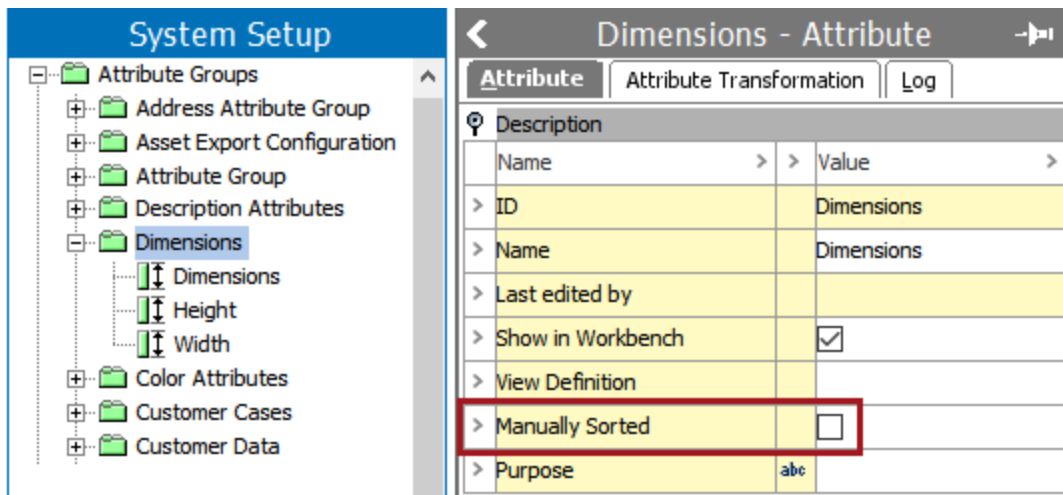
As per the default function in STEP, attributes shown in the Product Editor of a product are shown in alphabetical order. However, if an Attribute Display Sequence Attribute has been set up, the attributes shown in the Product Editor are sorted according to the order set in the Display Sequence attribute.

Important: The Display Sequence attribute only affects 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 display according to the order set in the Display Sequence attribute.

Before using display sequence attributes, you must visit the attribute groups in which the relevant attributes are contained and ensure that 'Manual Sorting' is not selected. If enabled on an attribute group, this option will overwrite any attribute sequences you configure. To check if the manual sorting is enabled, select the relevant Attribute Group in System Setup. The option will display there.

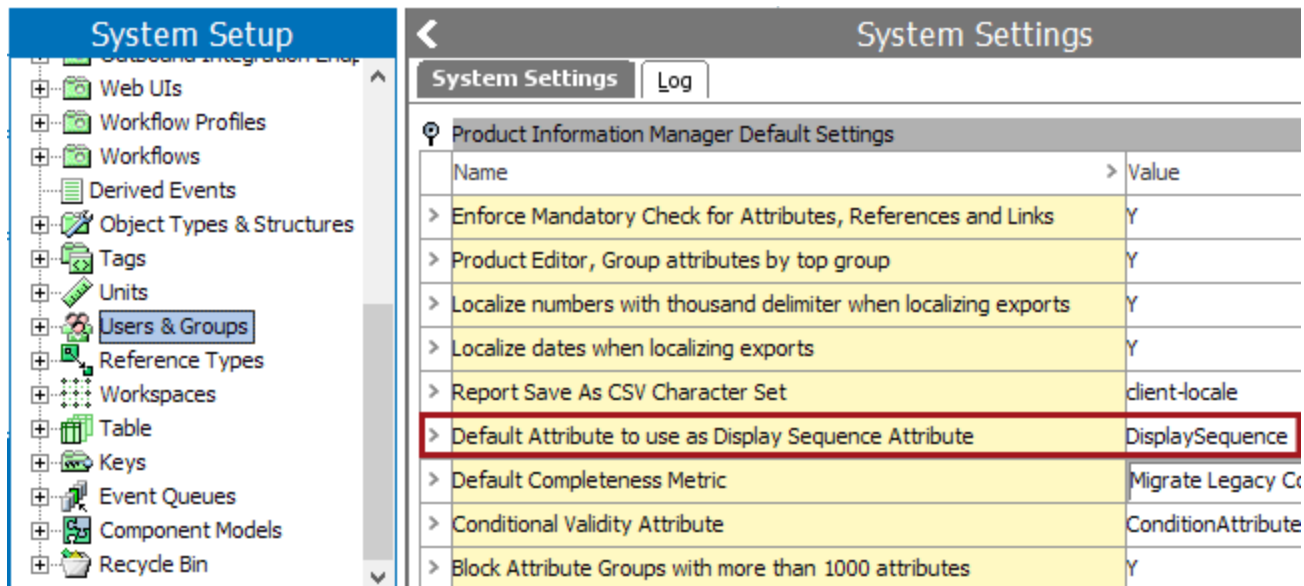


Set Up Display Sequence Attribute

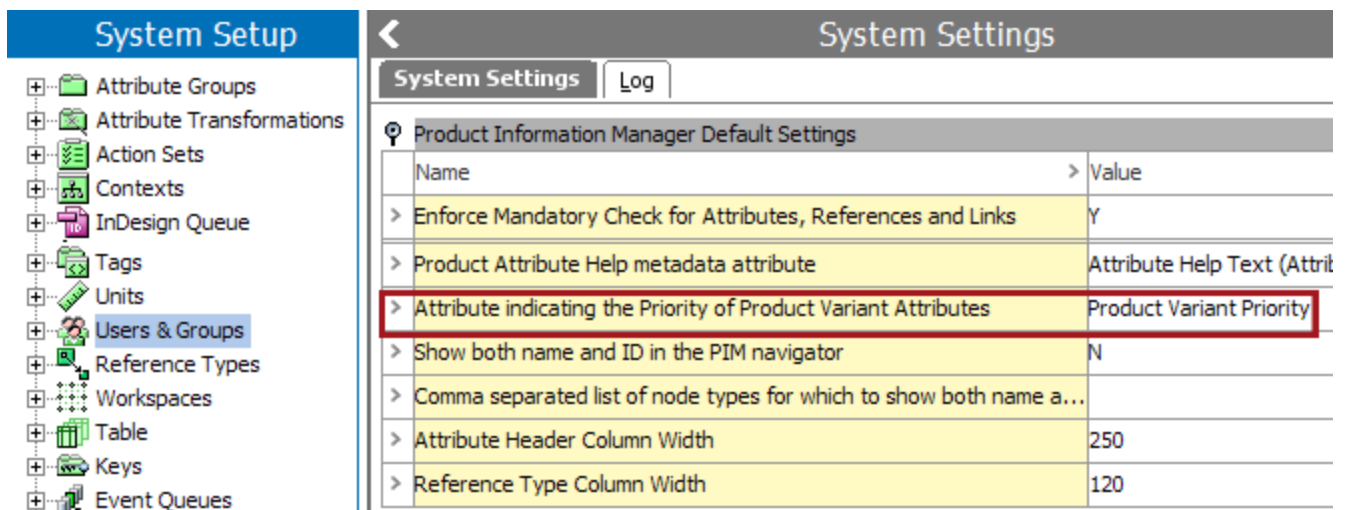
A Display Sequence Attribute is a description attribute that dictates how other attributes will be ordered on an object. To configure a Display Sequence Attribute, it must meet the following conditions:

- The attribute has to be a **Description** attribute
- The attribute has a Validation Base Type of 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**.



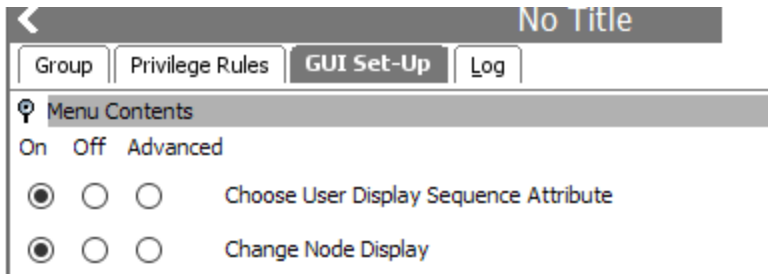
Similarly, the Display sequence for the Product variants object types can be set up by going to System Setup> Users & Groups > System Settings tab > Product Information Manager Default Settings flipper > **Attribute indicating the Priority of Product Variant Attributes**.



View Menu: Choose Attribute Display Sequence Attribute

By default, users are able to select a Display Sequence attribute. However, if admin users want to prevent specific users from making this selection, they may disable this privilege by making a change to the user group that contains the relevant users. To make this change, admin users may take the following steps:

1. In System Setup, expand **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. Set it to OFF.

For more information on how to use the Display Sequence Attribute, see the **Default Attribute to use as Display Sequence Attribute** heading within the **Product Information Manager Default Settings** topic of the **System Setup / Super User Guide** documentation.

Sorting Attributes

Once the attribute display sequence attribute has been set up, a user can establish 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, expand **Attribute Groups**, then click the relevant attribute. An **Attribute** editor displays.
2. In the Display Sequence field, type the desired number for each attribute in an attribute group. The lower the value, the higher priority it will have compared to other attributes. For example, an attribute with the value of 10 will always have priority over an attribute with the value of 20. Attributes without a display sequence value will display alphabetically after the prioritized attributes.

System Setup

- Attribute Groups
 - Attribute Group
 - brown
 - (green)
 - Attribute D
 - Attribute N
 - AttrSize
 - AttrTShirtSize
 - blue
 - Color, Interior
 - Condition
 - Hazmat
 - Hazmat Y N
 - Marketing Description
 - Reference Attribute
 - Released by

(brown) - Attribute

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

Description

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

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 the Tree, expand the Product Hierarchy and then 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 Fl
 - LIGHTS-10

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

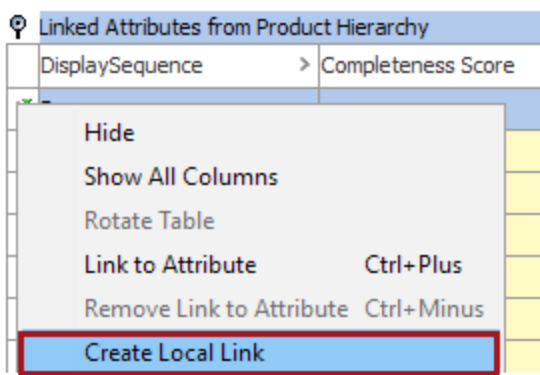
Publications

Linked Attributes from Product Hierarchy

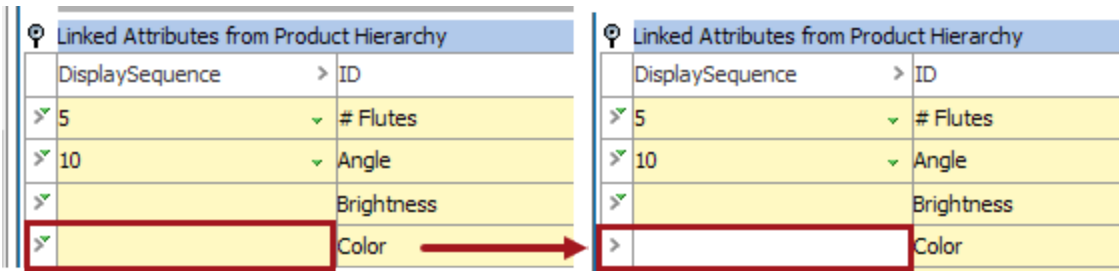
DisplaySequence	>	Completeness Score	>	ID	>	Name
5				BrandName		Brand Name
6				BrandOwner		Brand Owner
				Brightness		Brightness

Note: You are only able to edit the Display Sequence attribute column if the attribute is linked in at the Product Hierarchy level 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 you have selected, right-click the relevant row and click **Create Local link**. You are now able to type a local value in the Display Sequence column.



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.



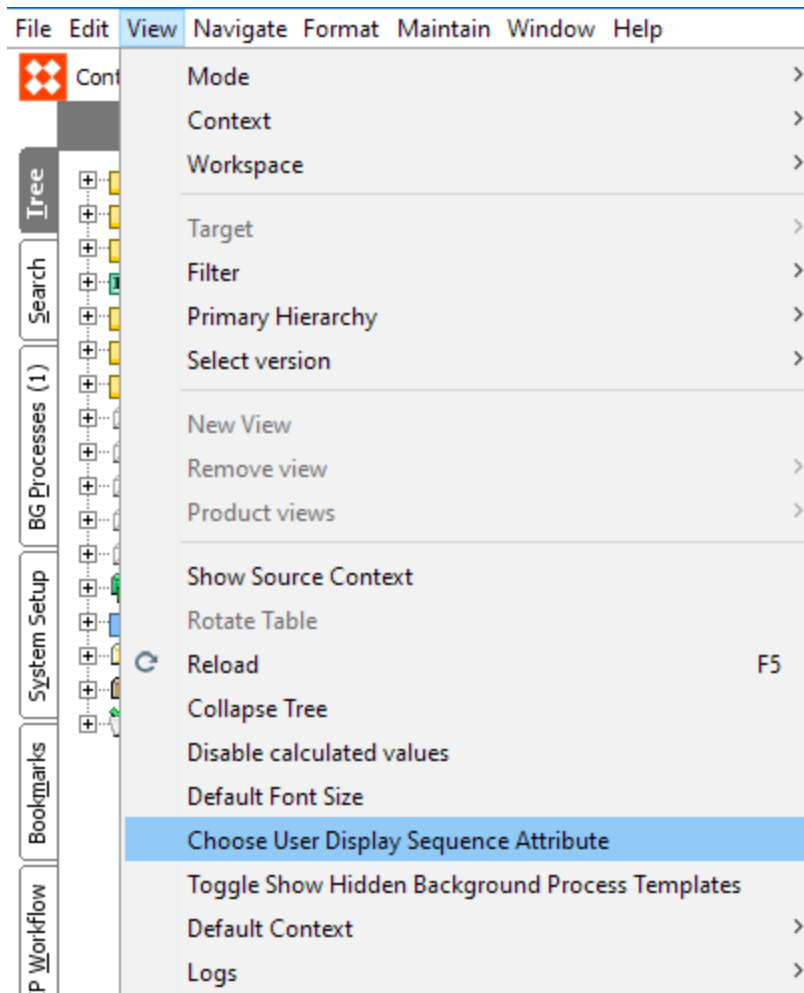
Attributes that have no sequence number, are therefore alphabetically sorted and will be placed last on the list.

Important: Information about the Display Sequence of attributes is exported in an XML file. 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.

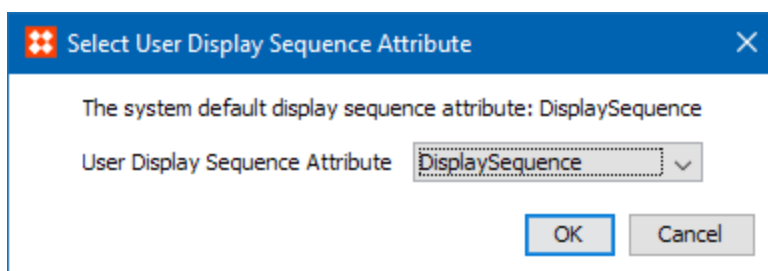
Selecting a Personal Attribute Display Sequence Attribute

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 displays.

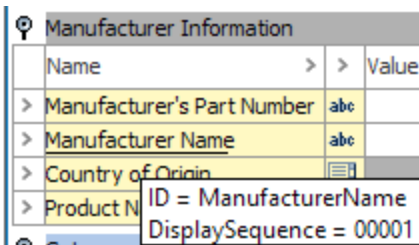


- If you now select a product folder /product, then select the **References** tab, the first column under **Linked Attributes** will now show your personal Display Sequence attribute.

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.

- 1. Set the cursor over the Attribute name and a tool tip will display.



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

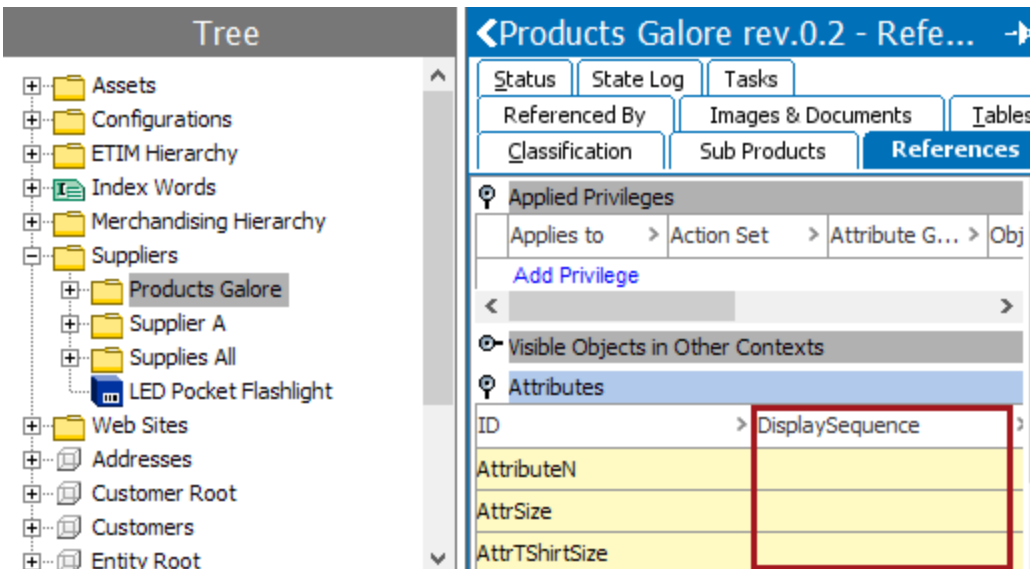
Note: If you select two or more products, the **Products** Editor will show the attributes in alphabetical order. The reason for this is that you can select two or more products that have the same attribute but with different values for the Display Sequence attribute.

Display Sequence Attribute and Classifications

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

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

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

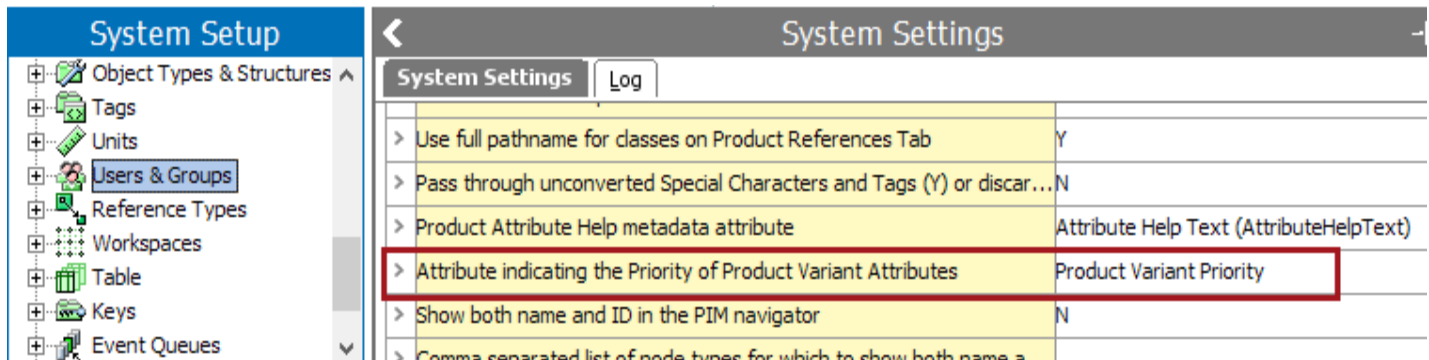


Important: Attribute Display sequences should not be maintained here, as they will have no effect in either the Classification editor or an export.

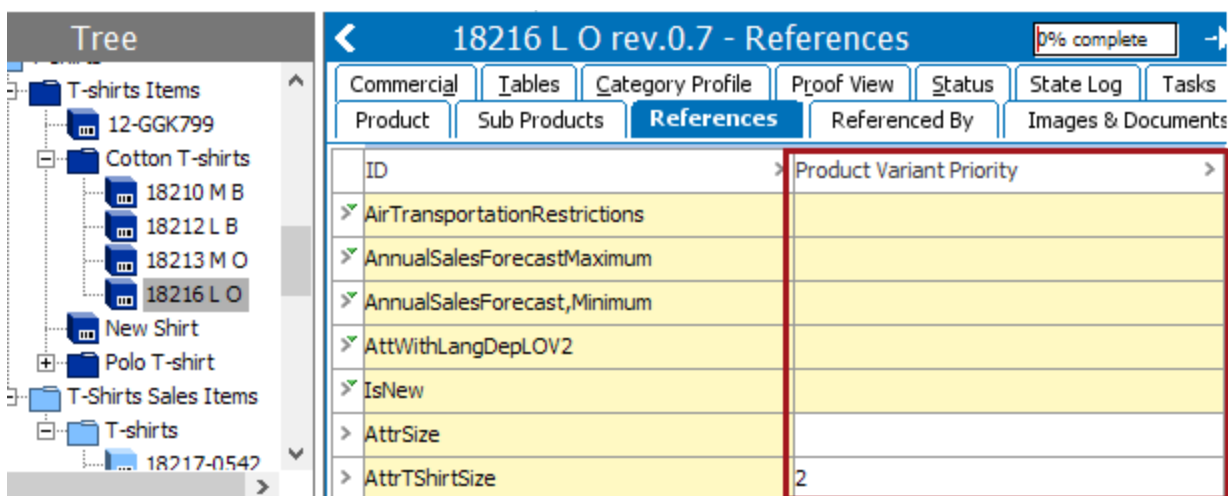
Setting the Priority of a Product Variant Attribute

- 1. In the Tree, select the parent containing the product variants.

- Click the References tab.
- 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 and 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.

Shown in the screenshot below is 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.

Row/Column Type & Content Definition

Column/Row Content definition: Attribute Group Attributes

Pull: Value & Unit

Browse Search

- Customer Data
- D&B Integration
- Data Containers for Entities
- Dates
- Description Attributes
- Dimensions
 - Height
 - (Length)
 - Width
- Display

Details

Transformation:

☐ Apply Recursively

☒ Limit Attributes (to only legal attributes and...)

☐ Include Attributes from Classifications

☐ Only non empty row/columns

Maximum number of rows/columns:

Sort on: Name ☒ Exclude if empty

OK Cancel

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

Metadata (or 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

Publication 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

Unique Identifiers

Multi Editor

Multi Editor

ID	Name	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 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.

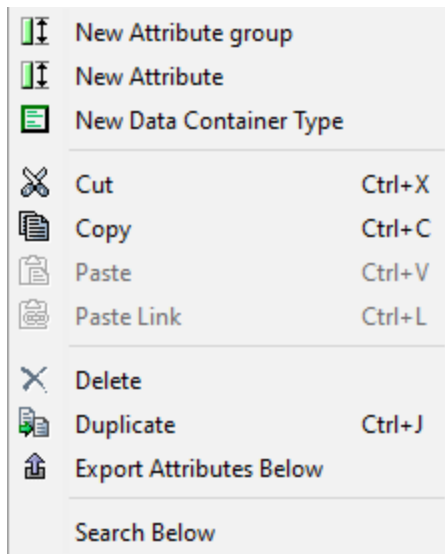
Note: Attribute groups can be specified in privilege rules. For example, specified users can change the values of attributes within the 'PriceGroup' attribute group.

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

Tree

Search

BG Processes (8)

Item 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

Attribute Group - Attribute

Attribute

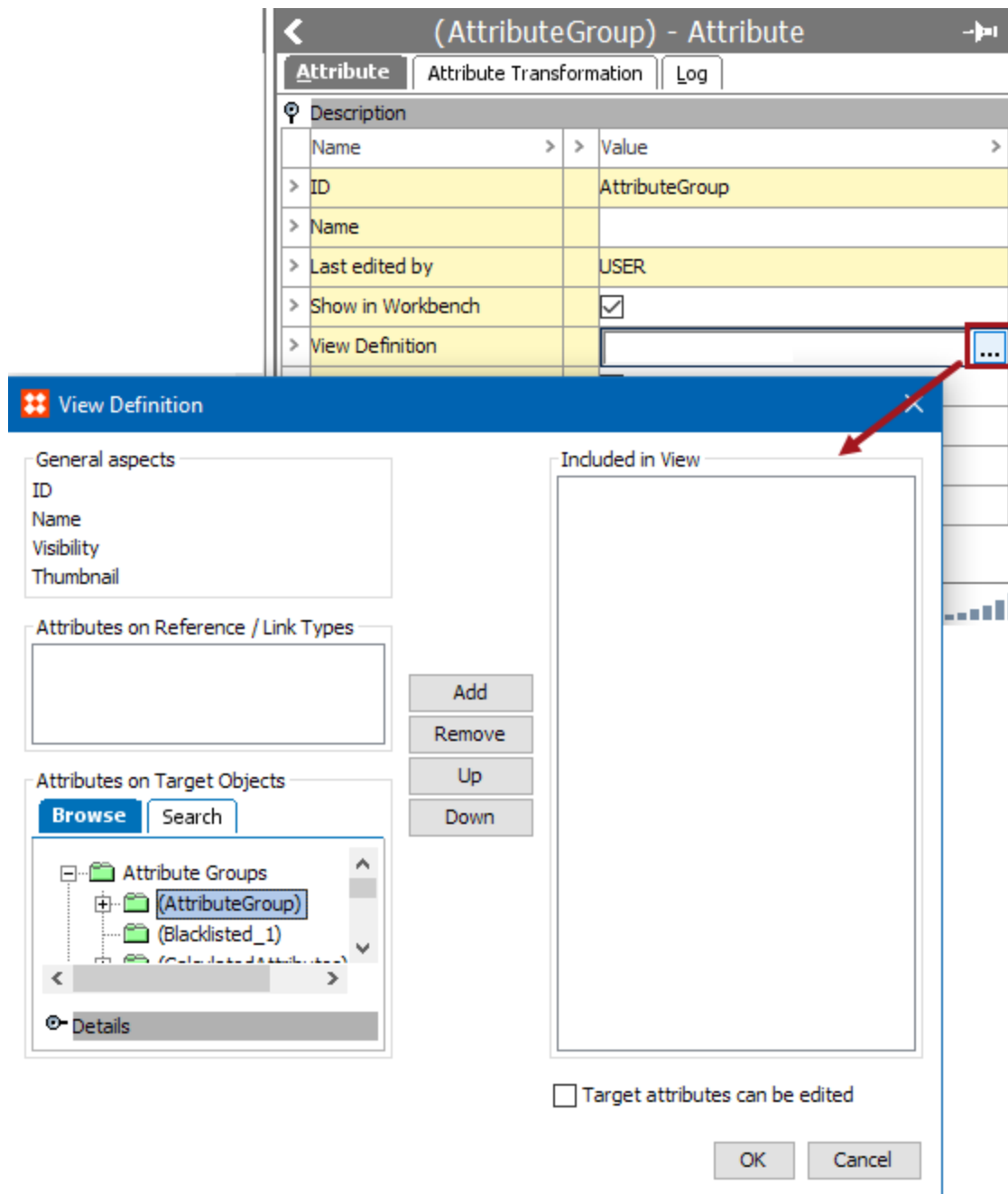
Attribute Transformation

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	

- 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.
- 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.

- 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 to 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.

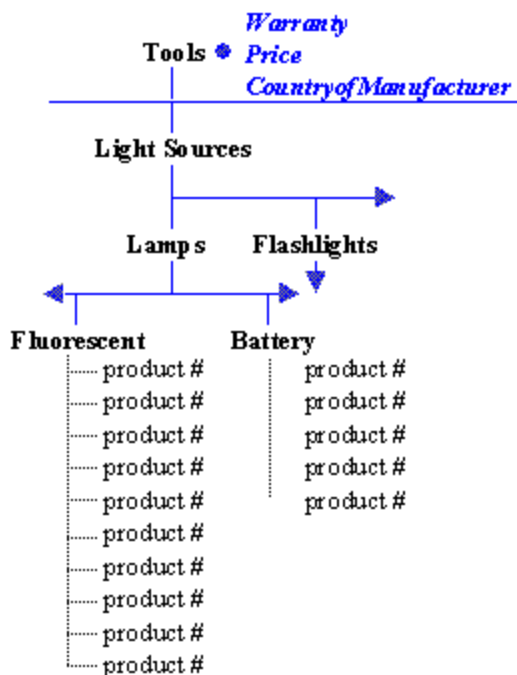


Figure 1 above: A part of a product hierarchy showing the principle of the inheritance of product attributes.

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.

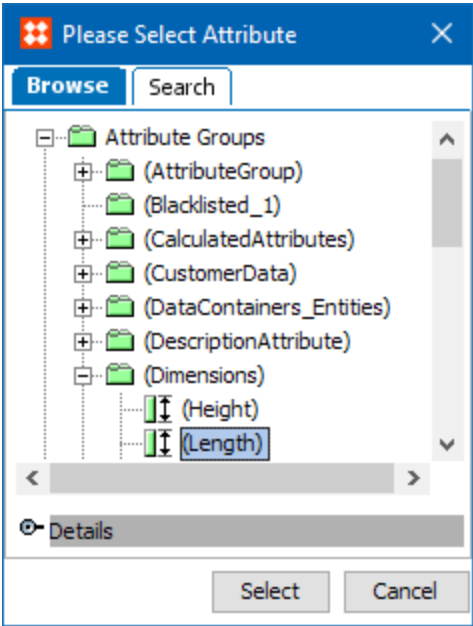
Note: If a specification attribute is not in the list of selectable attributes, it may not be available in the current context.

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)

Attributes

Products

Producting Hierarchy

Tools

Task Lighting

Flashlights

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

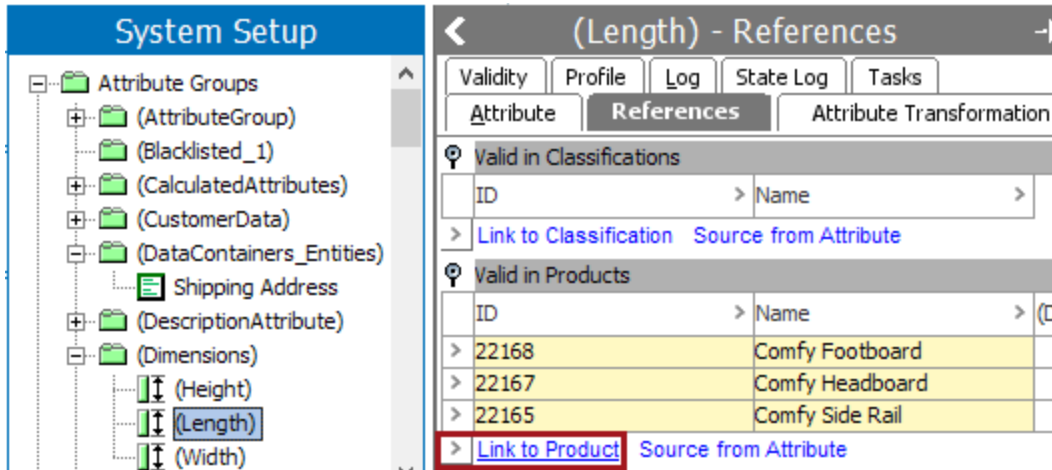
(DisplaySequence)	Completeness Score	ID	Name
2	▼	IsOrderableUnit	(IsOrderableUnit)
4	▼	LeadTime	(LeadTime)
>		Length	(Length)
1	▼	ListPrice	(ListPrice)

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.

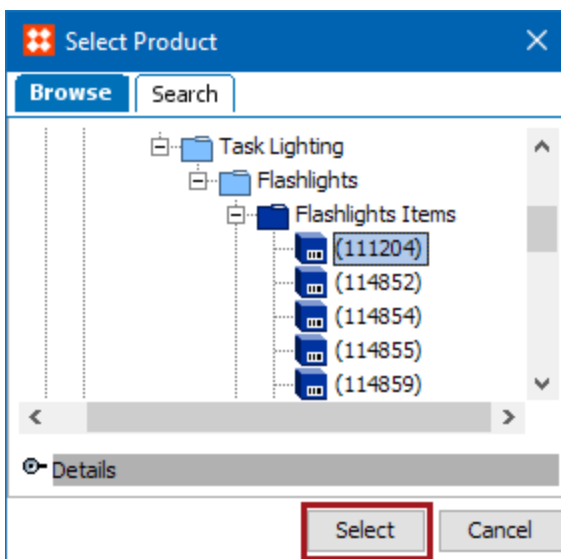
Linking Attributes to Products or Classifications

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.

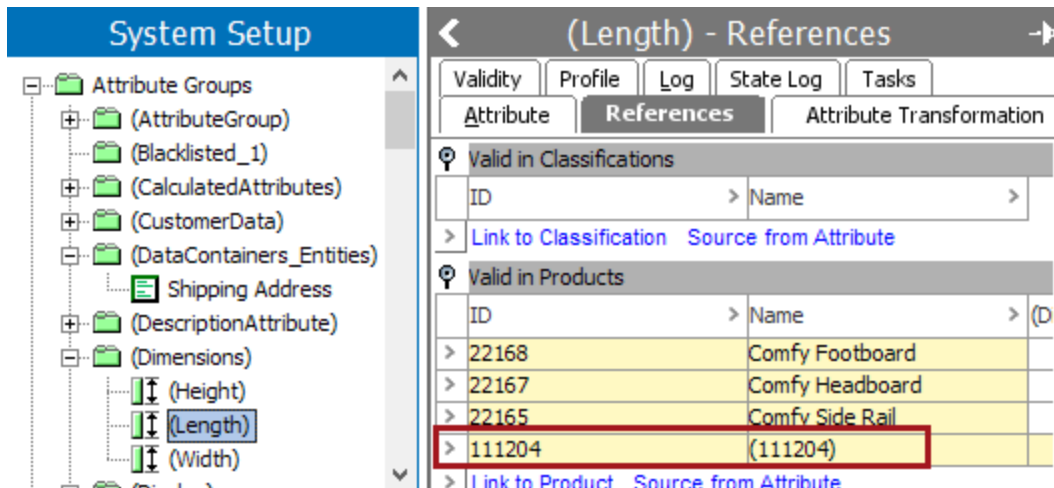
1. In System Setup, select the relevant attribute node, and click the **References** tab.
2. Under the 'Valid in Products' flipper, click **Link to Product** to link the object.



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**.

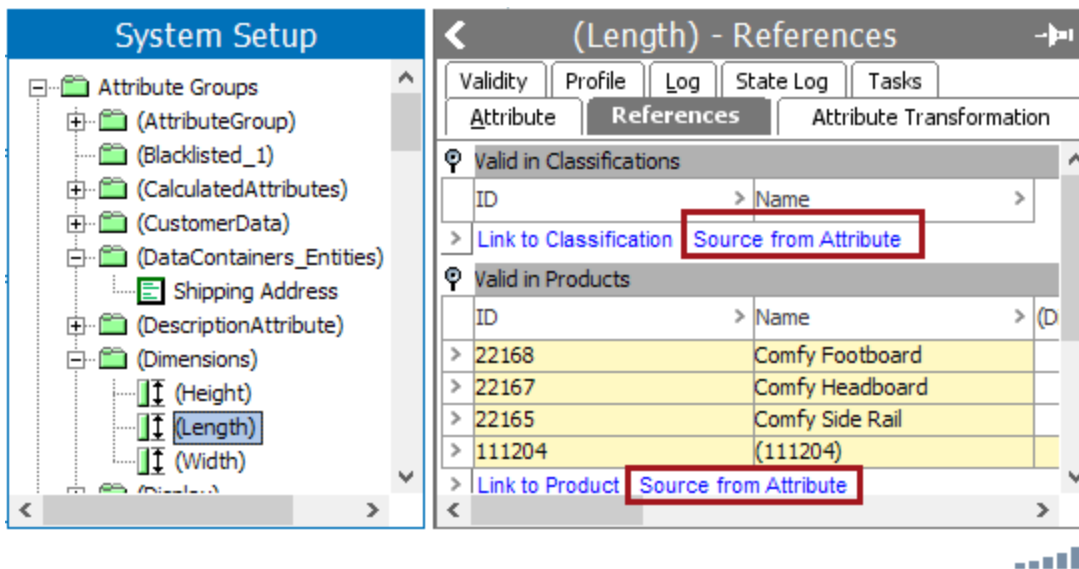


The selected product(s) will appear in the list of 'Valid in Products.'

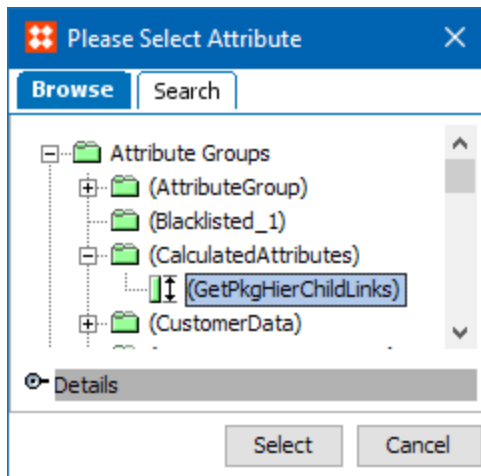


Linking to a Source from Attribute

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.



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 attribute link 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*.

The screenshot shows the Stibo Systems interface. On the left is a 'Tree' view with a hierarchy: T-shirts Items > (112456) > Bleu T-shirt > Cotton T-shirts > Polo T-shirt > T-Shirts Sales Items. The main area displays the 'Bleu T-shirt rev.0.17 - Product' editor. It has tabs for Category Profile, Proof View, Status, State Log, Referenced By, Images & Documents, and Cor. The 'Product' tab is active, showing a table with columns 'Name' and 'Value'. The table contains rows for '(Hazmat)' with value 'abc', '(Size)' with value 'abc', and '(Height)' with value '123 12 cm'. A red box highlights the '(Height)' row. A red arrow points from this row to the '(Height) - Attribute' editor. The attribute editor has tabs for Validity, Profile, Log, State Log, Tasks, Attribute, References, and Attribute Transformation. The 'Attribute' tab is active, showing a table with columns 'Name' and 'Value'. The table contains rows for 'ID' with value 'Height', 'Name', 'Last edited by' with value '2017-01-05 07:07:14 by USERL', 'Full Text Indexable' with value 'No', and 'Externally Maintained' with value 'No'.

4. In the attribute editor, click the **References** tab.
5. 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.

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 > Users & Groups > System Settings > Product Information Manager Default Settings flipper > verify the attribute under **Default Attribute to use as Attribute Display Sequence Attribute**. Alternatively, on a product editor > References tab > 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)				
✓		AttWithLangDepLOV2	(AttWithLangDepLOV2)				

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 the 'Linked Attributes from Product Hierarchy' table. The table has columns: Display Sequence, Completeness Score, and ID. The rows are sorted by ID. A context menu is open over a row, showing options: Hide, Show All Columns, Rotate Table, Link to Attribute (Ctrl+Plus), Remove Link to Attribute (Ctrl+Minus), and Create Local Link (highlighted). A red arrow points 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

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

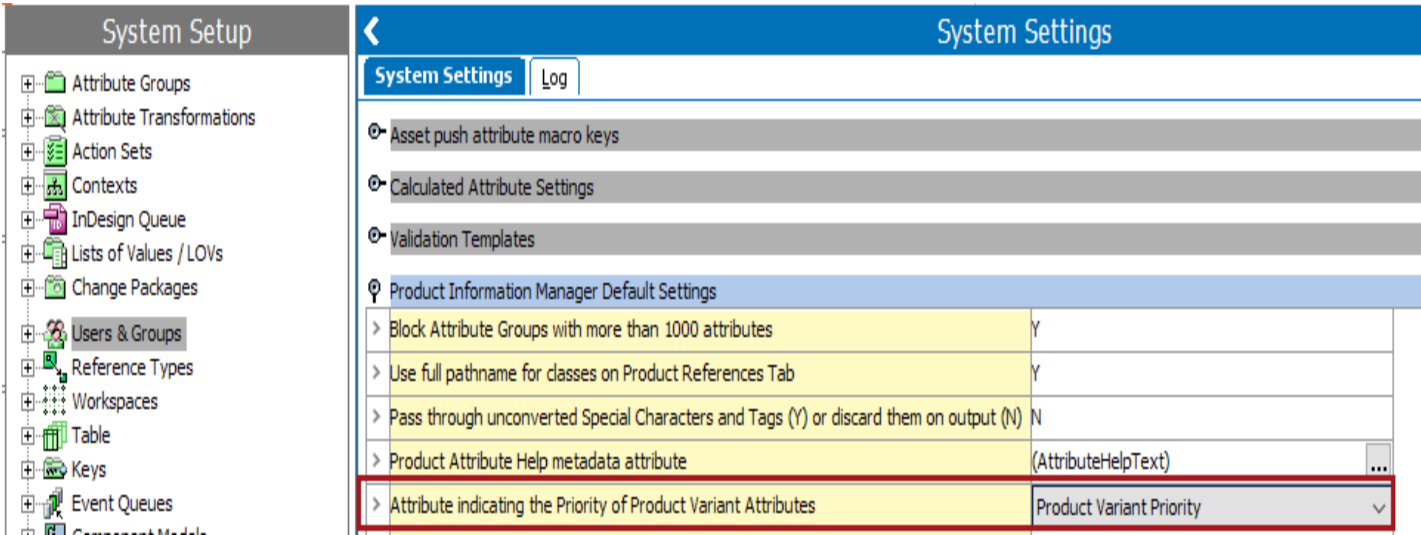
Important: Information about the Display Sequence of attributes is exported in an XML export. If the Display Sequence number is set on the attribute itself, the XML Export will contain the Attributes with Display Sequence attribute value. If the Display Sequence number is set on the Attribute Link, the 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 **System Setup / 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.

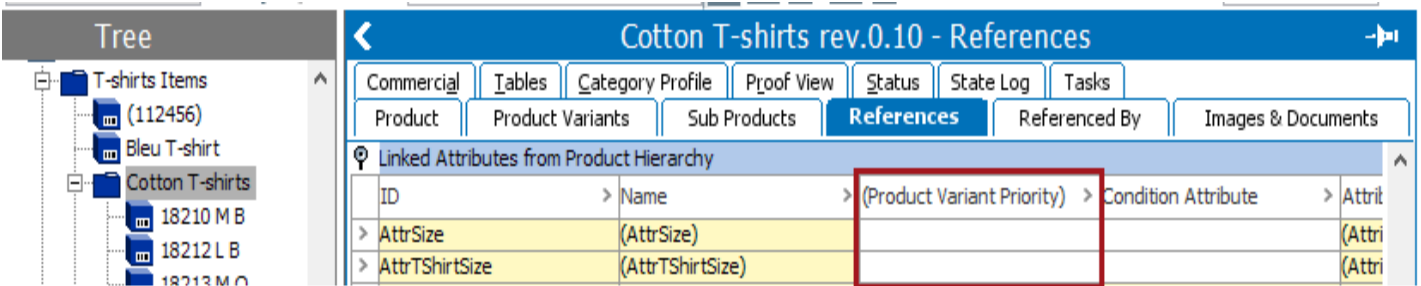
To check if it is selected go to System Setup > Users & Groups node > 'Product Information Manager Default Settings' flipper > **Attribute indication the Priority of Product Variant Attributes** field. In the field make sure that attribute set up to be the 'Product Variant Priority' is selected from 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. This is again specified in User & Groups under the '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, as this 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
 - Publication 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
 - Unique Identifiers

Multi Editor

ID	Name	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

Attribute transformations are maintained in System Setup > Attribute Transformations can be used to transform attribute values, product names, and terms lists. Transformations are not intended to be a substitute for a programming language but can give a non-technical user the opportunity to modify data, using series of standard modification operations.

As with all transformations, the original data is not modified, but the data delivered in an output, received during import, or displayed when mounted in STEP'n'design is altered by the transformations applied. 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.

The following information is available for attribute transformations:

- **Creating an Attribute Transformation** includes steps for making a new transformation.
- **Attribute Transformation Options** includes information about each available transformation.
- **Maintaining an Attribute Transformation** includes information about configuration.

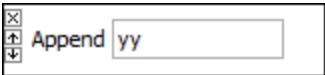
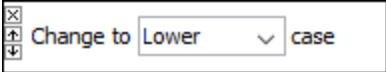
Attribute transformations can also be used to make a set of transformations, and then applied in the following additional areas where transformations are available:

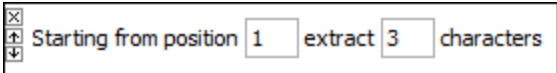
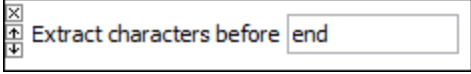
- Inbound and outbound data, when the selected format requires mapping, as discussed in the **Data Mapping** topic of the **Data Exchange** documentation.
- Tables for formatting, as discussed in the **Table Formatting Transformations** topic of the **Tables** documentation.

Attribute Transformation Options

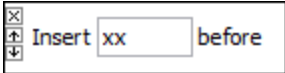
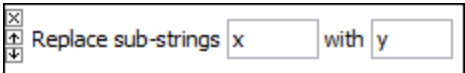
The following options are available after clicking the **Add Transformation** link on an Attribute Transformation:

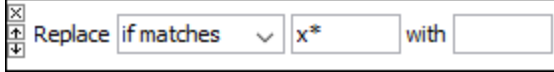
When applying multiple transformations, they are executed, in the order in which the transformations are applied. For more information, see the **Creating an Attribute Transformation** topic in the **System Setup / Super User Guide** documentation.

Transformation	Description
Append text	 Adds the specified text as a suffix to the end of the value. For an example using this transformation, see Concatenate Data Using Variables Example.
Change case	 Sets values to all upper case, all lower case, or mixed case. Mixed case sets the first letter of the value to upper case and the remainder to lower case. For an example using this transformation, see Split, Extract, and Convert Data Example.

Transformation	Description
Choose hash function	 Converts the attribute value to a cryptographic text using the selected cryptographic function: MD5 or SHA-1. This option is displayed when the STEPMatching licensing is applied. The function reads an attribute value, encrypts it, and saves the encrypted value as a standard attribute value in STEP. It could be used to encrypt (or 'hash') address field(s) so that by evaluating a single field, a changed address could trigger a revalidation process. While available for Attribute Transformation within System Setup, it is intended for use only when importing data using Import Manager or IIEP. No corresponding export function exists, nor is there any decryption functionality within STEP
Convert fractions to parsable tags	 Scans a text string for fractions. If a fraction is found, then the fraction is converted into parsable tags. A text string that contains the fraction 1/2 cm is transformed to: <StartNumerator/><bflidin/>1<FractionBar/><StartDenominator/>2<EndFraction/>cm You have to set up an output formatting to Indesign for <StartNumerator/>, <bflidin/>, <FractionBar/>, <StartDenominator/> and <EndFraction/> to mount the fractions.
Extract characters after a given text in the value	 Extracts a part of the value, starting after the occurrence of the specified text string.
Extract characters at position	 Extracts a specified number of characters after a specified position in a text string. For an example using this transformation, see Split, Extract, and Convert Data Example.
Extract characters before a given text in the value	 Extracts a part of the value, starting from the beginning of the text string and ending at the occurrence of the specified text string. For an example using this transformation, see Split, Extract, and Convert Data Example.

Transformation	Description
Extract characters between two pieces of text in the value	✕ ⬆ ⬇ Extract characters between ab and cd Extracts the texts between the specified starting text string and the specified ending text string. The start and end text string cannot be the same.
Extract prefix, suffix and value from an embedded number	✕ ⬆ ⬇ Extract prefix, suffix and value from an embedded number with a unit v Shows the prefix, suffix, and value of the embedded number with / without the unit. This is only relevant if the attribute you are mapping to is of the type 'embedded_number', otherwise, the transformation does not function.
Extract unit from an embedded value	✕ ⬆ ⬇ Extract unit from an embedded value Extracts only the unit from an embedded value. The prefix, suffix, and value are not extracted. This is only relevant if the attribute you are mapping to is of the type 'embedded_number', otherwise, the transformation does not function.
Extract unit part of value	✕ ⬆ ⬇ Extract unit part of value 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	✕ ⬆ ⬇ Extract the last 2 characters Extracts a specified number of characters of a value, counting from the end. For an example using this transformation, see Split, Extract, and Convert Data Example.
Format number	✕ ⬆ ⬇ Format number according to format #.00 . Localize False v Formats the number to only have a specified number of decimals, and enables you to localize the number.
Format the number with unit	✕ ⬆ ⬇ Format number according to format #.00 . Localize False v

Transformation	Description
	Formats the number with a unit to only have a specified number of decimals, and enables you to localize the number.
Generate Soundex code	Soundex code for the value. Soundex is a general algorithm which creates the same type of 4 alphanumeric code for the words with similar pronunciation.
Insert text at a given position	 Inserts a series of characters into the attribute value at the specified position.
Insert text before	 Adds the specified text as a prefix before the value.
Math operation (+, -, *, /)	 Performs the standard math operations on the value: addition, subtraction, multiplication, and division. Only valid for integer and number attributes.
Parse Address	 Splits an address contained in a single field into multiple fields using <multisep/>. This allows addresses to be normalized and validated. For example, the import of "1200 Mail Rd Hays KS 11111" should be transformed to "KS<multisep/>HAYS<multisep/>11111<multisep/>MAIL<multisep/>RD<multisep/>1200".
Remove unit from value	 Extracts the value of an attribute. If the value includes a unit, then the unit will be ignored. For an example using this transformation, see Split and Extract Data Example.
Replace substrings of the	

Transformation	Description
value	Changes the specified text string or empty value with the replacement text string or empty value. All matches on the value in the 'Replace sub-strings' text box are replaced. For example, all instances of 'Personal Computer' could be replaced with 'PC'.
Replace substrings of the value using a regular expression	 Uses regular expression to change the specified find value (in the first text box) with the replacement value (in the second text box). From the dropdown, select a replace option: 'if matches', 'first match of', or 'every match'. The if matches option must be set so that the find field defines the <u>entire</u> string to be replaced because it operates like the 'Replace the whole value' transformation. The first match of and every match options operate like the 'Replace substrings of the value' transformation and they replace only the text specified in the find field. For more information on regular expression and practical examples, see the Regular Expression topic in System Setup / Super User Guide documentation.
Replace the whole value	 Replaces the specified text or empty value with a different value. You must enter the entire value, or leave the Replace field blank. Only works on text-based attributes. For an example using this transformation, see Split, Extract, and Convert Data Example.
Replace the whole value using Lookup Table	 Replaces the specified value with another value from the selected Lookup Table. For setup of a transformation table, see the Creating a Transformation Lookup Table topic in this documentation. For an example using this transformation, see Transformation Lookup Table Example.
Replace words using Lookup Table	 Replaces the specified words with another value from the selected Lookup Table. For setup of a transformation table, see the Creating a Transformation Lookup Table topic in this documentation. For an example using this transformation, see Transformation Lookup Table Example.

Transformation	Description
Split and extract	 Split by x and extract field 1 Split the value where a given string is found and returns one of the created fields. You can, for example extract data from a value after the 4th occurrence of a hyphen, and before the 5th occurrence. So if the value is 179-245-Y-MYGR-TG105P-FDSJ-198, the result is TG105P. For an example using this transformation, see Split and Extract Data Example.
Transform date by locale	 Transform from Date Localized dates should follow the DateFormat.SHORT JavaScript formatting style. Times should be in the DateFormat.MEDIUM formatting style. Transforms Date, ISO Date, or ISO Date and Time attributes to localized formats. Inbound 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. Outbound 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. For step-by-step instructions using this transformation, see the Transform Date by Locale or by Pattern topic of the Transformation Examples section of the Data Exchange documentation.
Transform date by pattern	 Transform from Date to pattern yyyy-MM-dd 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). You 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. Inbound 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. Outbound transforms Date, ISO Date, or ISO Date and Time attributes to a patterned

Transformation	Description
	format (e.g., yyyy.MM.dd G 'at' HH:mm:ss z, EE, MMM d, "yy). You can use the prefilled 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. For step-by-step instructions using this transformation\, see the Transform Date by Locale or by Pattern topic of the Transformation Examples section of the Data Exchange documentation.
[Transform using attribute transformation]	 Applies an attribute transformation that is already defined in System Setup under Attribute Transformations. Makes it possible to apply a set of transformations to a value. The transformation takes a predefined attribute transformation as argument. For an example using this transformation, see Transform Using Attribute Transformation Example.

Creating an Attribute Transformation

Attribute Transformations are maintained in System Setup under the Attribute Transformations node using the following functionality:

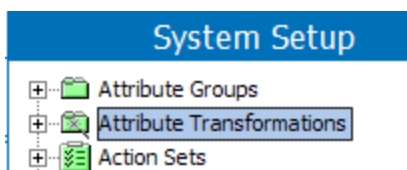
- **Create an Attribute Transformation Group** to organize the attribute transformations in your system.
- **Create an Attribute Transformation** to define one or more transformations to be used on your data.

These steps are defined below.

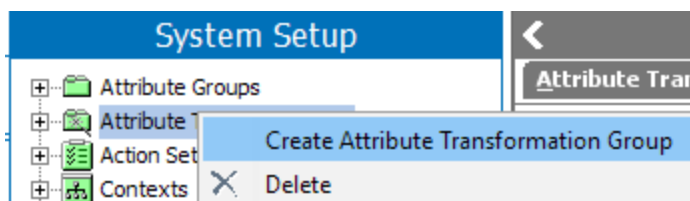
Create an Attribute Transformation Group

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. It has a blue title bar with the text 'Create Attribute Transformation Group' and a close button (X). The dialog contains two text input fields: 'ID' and 'Name'. Below the fields are two buttons: 'Create' and 'Cancel'.

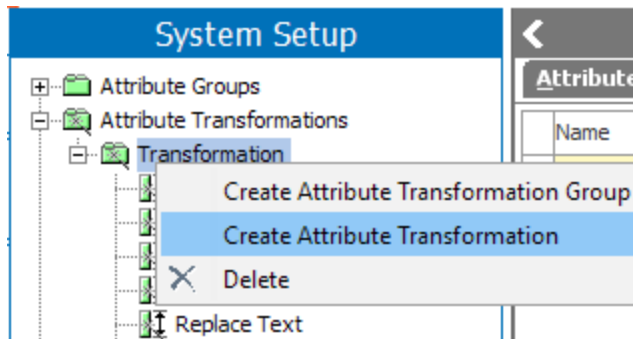
3. Type an ID and name for the attribute transformation group, and then click **Create**.

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

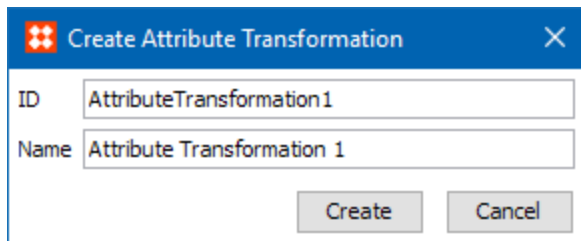
Create an Attribute Transformation

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**.



4. Once the attribute transformation is created, it will need to be set up with a transformation. Follow the steps in **Maintaining an Attribute Transformation** to complete the configuration.

Configuring an Attribute Transformation

Once an attribute transformation is created, it can be configured by adding transformations.

1. In System Setup, open the Attribute Transformation node, and then select an Attribute Transformation to display the editor.

The screenshot displays the 'System Setup' window on the left and the 'New Item - Attribute Transformation' editor on the right.

System Setup (Left Panel):

- Attribute Groups
- Attribute Transformations
 - Transformation
 - Commercial Price
 - Fraction
 - New Item** (selected)
 - Replace Text
- Action Sets
- Contexts
- InDesign Queue
- Lists of Values / LOVs
- Change Packages
- Gateway Endpoints
- Global Business Rules
- Inbound Integration Endpoints
- Match Codes and Matching A
- Outbound Integration Endpoints
- Web UIs
- Workflow Profiles
- Workflows
- Derived Events
- Object Types & Structures
- Tags
- Units

New Item - Attribute Transformation (Right Panel):

Attribute Transformation

Name	Value
ID	NewItem
Name	New Item

Transformations

- ☒ Replace Yes with New!
- ☒ Replace No with

[Add Transformation](#)

Multivalue Separator

Applies To

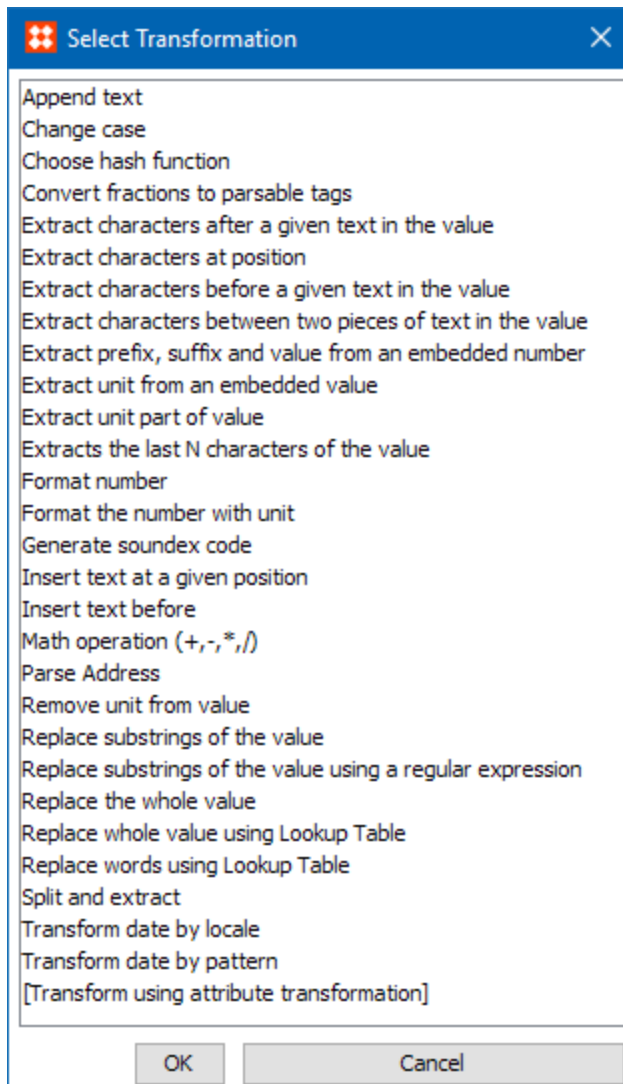
- ☐ Name
- ☐ ID

ID	Name
Add Attribute	

☐ All

☐ Availability

2. Under the Transformations flipper, click **Add Transformation** to display the Select Transformation dialog box appears with a list of rules to be used.



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

For a description of the transformations, see the **Attribute Transformation Options** topic.

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 Multi-valued Attribute should be mounted / extracted. If a character is keyed in, then the character will be inserted between each value in the multi-valued Attribute.

For 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. So using </return> would result in a list with each value separated by a return.

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** checkbox 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** checkbox 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.

Attribute Transformation

Please Select Attribute

Browse Search

(AttributeGroup)

- (AttributeN)
- (AttrSize)
- (AttrTShirtSize)
- (AttWithLangDepLOV2)
- (blue)
- (brown)
- (Color, Interior)

Select Cancel

The transformation can be used in

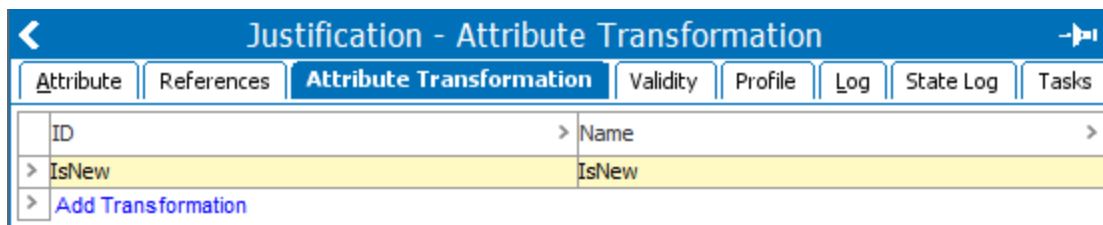
- Tables on columns or rows where attributes are extracted. The transformation can be applied columns or rows that list attribute values. For more information about applying attribute transformations, see STEP Table Component User Guide.
 - InDesign on product templates. The product template can be setup 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 STEP'n'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.

When an attribute is selected in the 'Applies To' flipper, the transformation is also displayed on the 'Attribute Transformation' tab when viewing the attribute editor.



Calculated Attributes

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. The foundation for a calculated attribute is a functional programming language very similar to the language used for defining functions in Excel.

For assistance in determining if a calculated attribute is the most efficient way to meet your requirements, see the **Calculated Attribute Considerations** topic in the **Calculated Attributes** section of the **System Setup / Super User Guide** documentation.

The value of a calculated attribute is determined by a value template which can be written in the Function Editor. Many functions can include an argument, which specifies the STEP data to be accessed. Functions can access information from references, compare data between objects, round numbers, modify text, and many other options. For a step-by-step guide to creating a calculated attribute, see the **Creating a Calculated Attribute** topic in the **Calculated Attributes** section of the **System Setup / Super User Guide** documentation.

As an 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 (3) 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 (2) attribute IDs are: 'Domestic Distribution ID' and 'International Distribution ID'.
- Assuming that the 'Domestic Distribution ID' value is 1234 and the 'International Distribution ID' value is 5678, the result of this 'Value template' would be 1234-5678.

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

For additional use cases for calculated attributes, see the **Calculated Attribute Use Cases** topic.

In Web UI, calculated attributes can be identified and their values can be overridden on the Attribute Management screen as described in the **Calculated Attributes in Web UI** section of the **Web User Interfaces** documentation.

Calculated Attribute Considerations

The decision to use a calculated attribute, and then the steps to create it to perform as expected, should include a consideration of the following topics:

- Deciding to Use a Calculated Attribute
- Setting up a Calculated Attribute
- Using Variables in a Calculated Attribute
- Optimizing Calculations using STEPXML
- Limit Performance Penalties for Calculations

Deciding to Use a Calculated Attribute

Although many functions can be accomplished using a calculated attribute, it is best to carefully consider when a calculated attribute is the most efficient way to address a requirement. Because a calculated attribute is recalculated each time it is viewed in an editor, and when it is exported (unless explicitly set to not calculate), overuse can lead to system inefficiency and user dissatisfaction. The type of data involved helps determine if a calculated attribute would be considered best practice, as defined in the sections below.

When deciding to use a calculated attribute or not, start with the following general rules:

- Static data, such as attribute values that rarely change, seldom benefits from a calculated attribute since the result is also static.
- Fluid data, such as referenced data where the target of the reference does change, often benefits from a calculated attribute since the result is rarely the same.

Static Data Example

Consider the following five (5) attributes that are used together to create a dynamic product description: Base Description, File Cabinet Color, File Cabinet Lock, File Cabinet Material, and File Cabinet No Drawers.

Products						
Products References Referenced By						
View: Description						
ID	Base Description	Calculated Description	File Cabinet Color	File Cabinet Lock	File Cabinet Material	File Cabinet No Drawers
> HON300SFC-MAH	File Cabinet	4-Drawer Steel File Cabinet in Mahogany, with Lock	Mahogany	with Lock	Steel	4
> HON300SFC-OAK	File Cabinet	4-Drawer Steel File Cabinet in Oak, without Lock	Oak	without Lock	Steel	4
> HON300SFC-WAL	File Cabinet	2-Drawer Steel File Cabinet in Walnut, with Lock	Walnut	with Lock	Steel	2
> HON300SFC-WHT	File Cabinet	2-Drawer Steel File Cabinet in White, without Lock	White	without Lock	Steel	2

In the image above, the calculated attribute (named Calculated Description) compiles each value into a single field. However, once an individual value is added for these attributes, they are not likely to change. If any one of the values did change, it would likely not be the same product. So using a calculated attribute in this case would mean that each time the data is displayed in an editor, or exported, the calculation runs but always returns the same result. Depending on the number of objects using this same calculation, the increased processing time could be significant. In this situation, it would be better to use a business rule to populate a standard specification attribute with the same information.

Fluid Data Example

Consider the following parts of a kit, where each includes product-to-product references. The requirement is to use the values on the target of the secondary reference.

In the example below, the 'Conference Room Starter Kit 1' References tab includes a reference to a single projector (with the ID JKL-5001-100) via the 'Part of a Kit' reference type. The projector, in turn, references three (3) other products via the 'Spare Part' reference type, as shown on the JKL-5001-100 References tab.

Conference Room Starter Kit 1

Product

Sub Products

References

Referenced By

Ungrouped Product References

Reference Type	Target
Part of a Kit	JKL-5001-100
Spare Part	

JKL-5001-100 rev.1.8 - References

Product

Sub Products

References

Referenced By

Image

Ungrouped Product References

Reference Type	Target
Part of a Kit	
Spare Part	Projector Cases Plastic Projector Lamp 100-750 Projector Screen 100" Folding

The screenshot below displays how these spare parts are displayed on the 'Conference Room Starter Kit 1' object Product tab using the calculated attribute named 'Spare Parts Calculation.'

Conference Room Starter Kit 1 rev.0.1 - Product

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Tab

Description

Name	Value
ID	Product-102356
Name	Conference Room Starter Kit 1
Object Type	Product
Revision	0.1 Last edited by USERJ on Tue Jul 18 10:22:11 EDT 2017
Approved	Never Been Approved
Translation	Not Translated
Path	Primary Product Hierarchy/Office Equipment/Conference Room Starter Kit 1
Spare Parts Calculation	Projector Lamp 100-750 --- [PL2752-750-2275] Projector Screen 100" Folding --- [PS4922-FL 100-2776] Projector Cases Plastic --- [PC6543-PL-5545B]

In this scenario, the referenced objects are expected to be updated regularly as new products are added to the line. This fluid data would benefit from a calculated attribute to ensure the updates are displayed.

Setting up a Calculated Attribute

Once you have determined that a calculated attribute is the most efficient way to address your requirement, review the following points to determine the necessary settings within the calculated attribute editor.

Description or Specification

Calculated attributes, like standard attributes, can be set to description or specification.

- **Calculated description attributes** can be made valid to objects based on the object type, just as any other description attribute. Thus a calculated description attribute can be made valid for an object, but as with all description attributes, the formula is not inherited. For more information, see the **Description Attributes** documentation.
- **Calculated specification attributes** allow the formula ('Function') to inherit down to subordinate products. The calculation will always be performed at each subordinate level. Thus, calculated attribute values are not inherited, only the formula is inherited. For more information, see the **Specification Attributes** documentation.

The image displays two screenshots of the Stibo Systems interface, illustrating the setup of a calculated attribute. Both screenshots show a tree view on the left with the following structure:

- Upper Body Wear
 - T-shirts
 - T-shirts Items
 - Cotton T-shirts (18210 M B)

The top screenshot shows the 'Cotton T-shirts' node selected. The right pane displays the 'Cotton T-shirts rev.0.11 - Product' details. The 'Calculated Attributes' section shows a table with the following data:

Name	Value
GetListOfRefs	HanesGrayT, HanesGreenT, Hanes Blue, Hanes Family, Hanes Orange

The bottom screenshot shows the same tree view, but the 'Cotton T-shirts' node is now expanded, showing its sub-nodes. The right pane displays the 'Cotton T-shirts rev.0.11 - Product' details. The 'Calculated Attributes' section shows a table with the following data:

Name	Value
GetListOfRefs	Hanes Blue

The following information identifies considerations when implementing a calculated attribute instead of a standard attribute.

Description Flipper

For all attributes, parameters describing the attribute are set in System Setup on the Attribute tab, under the Description flipper.

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

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

- **Mandatory** - Although an attribute with the parameter 'Calculated = Yes' can also have the parameter 'Mandatory = No', this is not advised. Since an actual value of a calculated attribute is not stored in the database (only the method to calculate the value), setting a calculated attribute to mandatory will prevent the attribute from being approved since it will never have a value.
- **Dimension Dependencies** - A calculated attribute only enforces the selected dimension dependencies when overriding the calculated value. 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. For more information about setting calculated attributes to be dimension dependent, see the **Calculated Attribute Settings** within the **Users and Groups** section of the **System Setup / Super User Guide** documentation.
- **Units** - When a calculated attribute uses the validation base type of Embedded Number, Fraction, Integer, Number, or Numeric Text, the **Unit template** field displays. This field can hold a static Unit ID and/or an expression that evaluates to a Unit ID.

Attribute Validation Flipper

For all attributes, parameters for Attribute Validation are set in System Setup on the Attribute tab, under the Attribute Validation flipper.

GetListOfRefs - Attribute	
Attribute	References Attribute Transformation Validity Profile
Description	
Attribute Validation	
Name	> Value
> Validation Base Type	Text
> List Of Values	N/A
> Multi Valued	No
> Mask	
> Minimum Value	N/A
> Maximum Value	N/A
> Maximum Length	100
Edit Validation Rule	

- **Validation Base Type** -Validation base type only takes effect when the calculated value is manually overridden, even when it is selected for a calculated attribute. 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 validity settings, unless 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.
- **Validation Base Type of List of Values (LOV)** - Calculated attributes cannot have a validation base type of List of Values (LOV). Changing this parameter on a calculated attribute is not allowed. Attempting to update an attribute with an LOV validation base type to be calculated displays an error.
- **Multi Valued** - Calculated attributes cannot have the Multi Valued parameter set to Yes. Changing this parameter on a calculated attribute is not allowed. Attempting to update a multi-valued attribute to be calculated displays an error.

Using Variables in a Calculated Attribute

Once you have determined that a calculated attribute is a good choice, next consider if one or more variables would be useful. A variable is used to separate each element of the calculation into it's own section. For more information on variables, see the **Variable Functions** documentation.

Note: When the calculation required involves many parts, it is a good idea to start with a small element, test and verify the results, before adding additional elements.

Functional and lengthy calculations without variables are harder to manage and troubleshoot later. To illustrate, the same calculation is written below; both without variables, and with variables.

The version that uses variables makes troubleshooting and maintenance easier for the following reasons:

- Identifying where a new attribute value should be inserted into the list is more obvious.
- Finding a missing parenthesis or comma is easier with fewer nested functions.
- Modifying a single element involves only the few defined lines in a variable, rather than having to first determine where the functions are in the larger calculation.

Example Input

The website requirement includes displaying the following 17 attribute values (when the exist) in a single block, including styling and introductory text to introduce the value.

HON300SFC-MAH		
Product	Sub Products	References
Build All		
Name	>	Value
> Fit Series	abc	Fits all series AH, AJ, and AK models
> For Matching	abc	Match with series AOK for side tables
> For Use With	abc	Computer laptops, desktops, or as a writing desk
> Includes	abc	Table, drawers, assembly kit, instructions
> Install With	abc	One Philips #2 Screwdriver
> Key Letter	abc	A
> Matching Accessories	abc	Hutch (AOK-2222) Footstool (AOK-2922)
> Optional Accessories	abc	Chrome plated drawer handle
> Order	abc	Order #: AOK-2290
> Product Name	abc	Utility Desk in Rosewood
> Recommended	abc	Home/Home Office
> Requires	abc	Available space of 36-3/4"x48-1/2"
> Shown In	abc	Rosewood
> SubTitle	abc	For Home or Home Office use
> Suite Inline References	abc	See also Utility Desk in Walnut
> Use With	abc	Use with: Desk Chair, Chair Mat.

Example Without Variables

Using the function editor, the following value template is created without variables. Notice that the first concatenate function wraps all of the attributes that should be included in the output. To make a change in one attribute, or the the formatting applied, requires locating and changing only the functions for that attribute.

Note that many character tags ([PN], [ST], [FT], etc.) are being used to style the text for output. For more information on tags, see the **Tags** documentation.

```
concatenate(
if(exact(prodval("KeyLetter"), ""), concatenate(" [PN] ", prodval("ProductName")),
concatenate(" [PN] ", prodval("KeyLetter"), ". ", prodval("ProductName")),
if(exact(prodval("SubTitle"), ""), "", concatenate("
[ST] ", prodval("SubTitle")),
if(exact(prodval("Features"), ""), "", concatenate("
[FI] ", prodval("Features")),
if(exact(prodval("MatchingAccessories"), ""), "", concatenate("
[FI]Matching Accessories:
[FI] ", prodval("MatchingAccessories")),
if(exact(prodval("Requires"), ""), "", concatenate("
[RQ]Requirements: ", prodval("Requires")),
if(exact(prodval("ForUseWith"), ""), "", concatenate("
[FI]Suitable for: ", prodval("ForUseWith")),
if(exact(prodval("ForMatching"), ""), "", concatenate("
[FI]For Matching: ", prodval("ForMatching")),
if(exact(prodval("InstallWith"), ""), "", concatenate("
[FI]Can be installed with: ", prodval("InstallWith")),
if(exact(prodval("Order"), ""), "", concatenate("
[FI] ", prodval("Order")),
if(exact(prodval("SuiteInlineReferences"), ""), "", concatenate("
[FI] ", prodval("SuiteInlineReferences")),
if(exact(prodval("UseWith"), ""), "", concatenate("
[FI] ", prodval("UseWith")),
if(exact(prodval("ShownIn"), ""), "", concatenate("
[SI]Shown in: ", prodval("ShownIn")),
if(exact(prodval("Recommended"), ""), "", concatenate("
[FI]Recommended Usage: ", prodval("Recommended")),
if(exact(prodval("FitSeries"), ""), "", concatenate("
[FS]Fit Series:
[FI] ", prodval("FitSeries")),
if(exact(prodval("OptionalAccessories"), ""), "", concatenate("
[FI]Optional: ", prodval("OptionalAccessories")),
if(exact(prodval("Includes"), ""), "", concatenate("
[FI]Includes: ", prodval("Includes"))
)
```

Example With Variables

Using the function editor, the following value template is created and includes variables. Notice that each attribute that should be output is addressed in a single variable at the top. The final concatenate function then calls each of the variables and separates them with a hard return ("n"). To make a change in one attribute, or the the formatting applied, simply locate the variable for that attribute.

```
{
pnnokey := concatenate("[PN]",prodval("ProductName")),
pnwkey := concatenate("[PN]",prodval("KeyLetter"),". ",prodval("ProductName")),
keyletter := if(exact(prodval("KeyLetter"),""),pnnokey,pnwkey),
subt := if(exact(prodval("SubTitle"),""),"",concatenate("[ST]",prodval("SubTitle"))),
feat := if(exact(prodval("Features"),""),"",concatenate("[FT]",prodval("Features"))),
match := if(exact(prodval("MatchingAccessories"),""),"",concatenate("[FT]Matching Accessories:", "\n", "[FT]", prodval("MatchingAccessories"))),
req := if(exact(prodval("Requires"),""),"",concatenate("[RQ]Requirements: ", prodval("Requires"))),
foruse := if(exact(prodval("ForUseWith"),""),"",concatenate("[FT]Suitable for: ", prodval("ForUseWith"))),
formatch := if(exact(prodval("ForMatching"),""),"",concatenate("[FT]For Matching: ", prodval("ForMatching"))),
install := if(exact(prodval("InstallWith"),""),"",concatenate("[FT]Can be installed with: ", prodval("InstallWith"))),
order := if(exact(prodval("Order"),""),"",concatenate("[FT]", prodval("Order"))),
suite := if(exact(prodval("SuiteInlineReferences"),""),"",concatenate("[FT]", prodval("SuiteInlineReferences"))),
usewith := if(exact(prodval("UseWith"),""),"",concatenate("[FT]", prodval("UseWith"))),
shown := if(exact(prodval("ShownIn"),""),"",concatenate("[SI]Shown in: ", prodval("ShownIn"))),
rec := if(exact(prodval("Recommended"),""),"",concatenate("[FT]Recommended Usage: ", prodval("Recommended"))),
fit := if(exact(prodval("FitSeries"),""),"",concatenate("[FS]Fit Series:", "\n", "[FT]", prodval("FitSeries"))),
option := if(exact(prodval("OptionalAccessories"),""),"",concatenate("[FT]Optional: ", prodval("OptionalAccessories"))),
includes := if(exact(prodval("Includes"),""),"",concatenate("[FT]Includes: ", prodval("Includes")))
}
concatenate(keyletter, "\n", subt, feat, "\n", match, "\n", req, "\n", foruse, "\n", formatch, "\n", install, "\n", order, "\n", suite, "\n", usewith,
"\n", shown, "\n", rec, "\n", fit, "\n", option, "\n", includes)
```

Example Output

Regardless of the method used to create the value template, the output contains the same static text and values as shown below in the following two resulting attributes.

- Build All Calculation - shows the result of the calculated attribute created without variables.
- Build All Calculation with Variables - shows the result of the calculated attribute created using variables.

HON300SFC-MAH rev.0.1 - Pro			
Product	Sub Products	References	Referenced By
Build All			
Name	>	>	Value
> Build All Calculation		fx	[PN]A. Utility Desk in Rosewood [ST]For Home or Home Office use [FT]Matching Accessories: [FT]Hutch (AOK-2222) Footstool (AOK-2922) [RQ]Requirements: Available space of 36-3/4"x48-1/2" [FT]Suitable for: Computer laptops, desktops, or as a writing desk [FT]For Matching: Match with series AOK for side tables [FT]Can be installed with: One Philips #2 Screwdriver [FT]Order #: AOK-2290 [FT]See also Utility Desk in Walnut [FT]Use with: Desk Chair, Chair Mat. [SI]Shown in: Rosewood [FT]Recommended Usage: Home/Home Office [FS]Fit Series: [FT]Fits all series AH, AJ, and AK models [FT]Optional: Chrome plated drawer handle [FT]Includes: Table, drawers, assembly kit, instructions
> Build All Calculation with Variables		fx	[PN]A. Utility Desk in Rosewood [ST]For Home or Home Office use [FT]Matching Accessories: [FT]Hutch (AOK-2222) Footstool (AOK-2922) [RQ]Requirements: Available space of 36-3/4"x48-1/2" [FT]Suitable for: Computer laptops, desktops, or as a writing desk [FT]For Matching: Match with series AOK for side tables [FT]Can be installed with: One Philips #2 Screwdriver [FT]Order #: AOK-2290 [FT]See also Utility Desk in Walnut [FT]Use with: Desk Chair, Chair Mat. [SI]Shown in: Rosewood [FT]Recommended Usage: Home/Home Office [FS]Fit Series: [FT]Fits all series AH, AJ, and AK models [FT]Optional: Chrome plated drawer handle [FT]Includes: Table, drawers, assembly kit, instructions

Optimizing Calculations Using STEPXML

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 rely on the value from another calculated attribute, since the value does not actually exist in the database.

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.

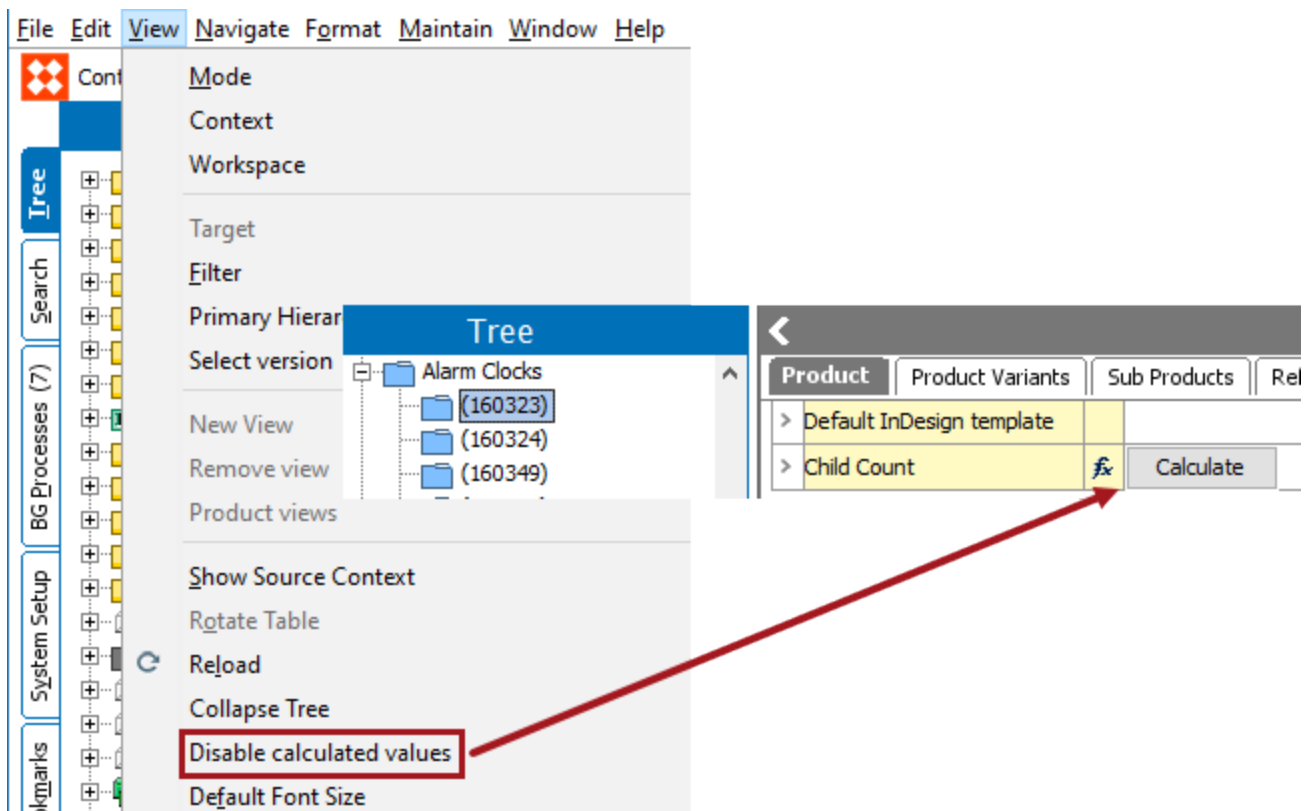
If these criteria are not met, then the calculation will be performed per context, which can increase the time to create the exported STEPXML file.

For more information on the STEPXML format, see the **STEPXML Format** topic in the **Data Exchange** documentation.

Limit Performance Penalties for Calculations

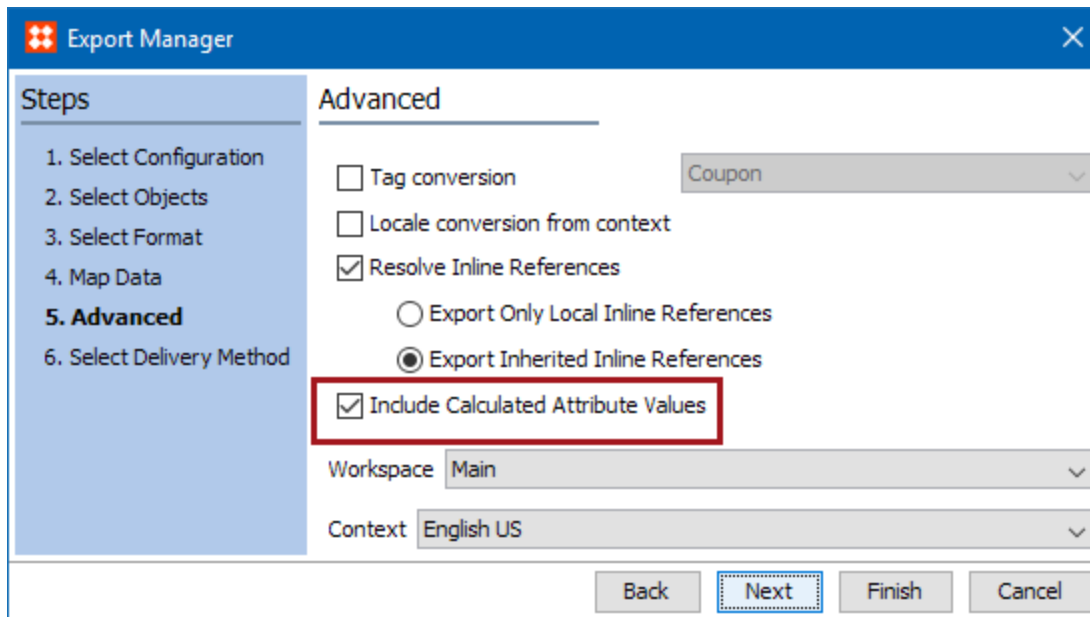
There is a performance penalty whenever the values are to be calculated. Complex calculations affect the time it takes to load editors and export objects.

To limit potential performance penalties in editors, it is possible to disable the calculated attributes in workbench by setting View > **Disable calculated values**. Once disabled, calculated attributes are displayed with a calculate button. Clicking the button displays the calculated value for only that attribute.



To limit potential performance penalties in outbound data, the Export Manager and OIEP configuration include the option to resolve (or ignore) Calculated Attributes during export.

- When the option is checked, the output includes the calculated value.
- When not checked, even if the attribute is mapped, the value is blank. This provides for a quicker output time.



Export Manager

Steps

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

Advanced

☐ Tag conversion Coupon

☐ Locale conversion from context

☒ Resolve Inline References

☐ Export Only Local Inline References

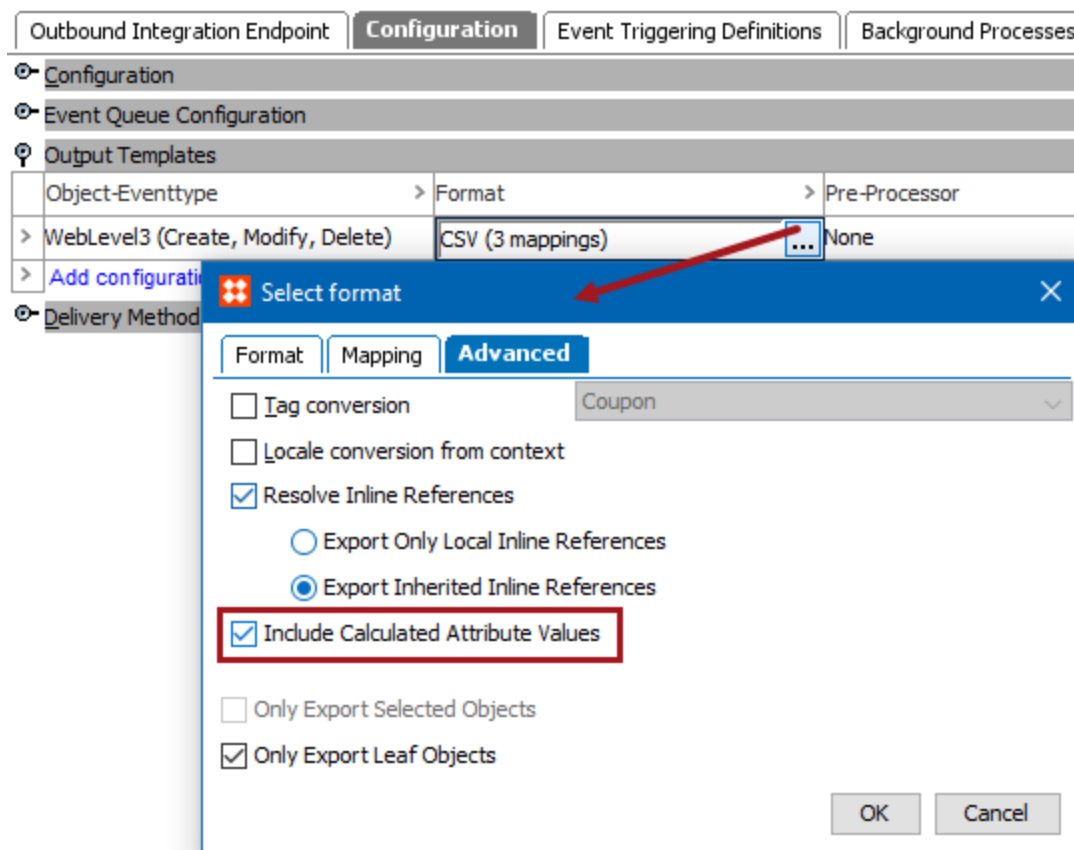
☒ Export Inherited Inline References

☒ Include Calculated Attribute Values

Workspace Main

Context English US

Back Next Finish Cancel



Outbound Integration Endpoint **Configuration** Event Triggering Definitions Background Processes

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor
WebLevel3 (Create, Modify, Delete)	CSV (3 mappings)	None

Select format

Format **Mapping** **Advanced**

☐ Tag conversion Coupon

☐ Locale conversion from context

☒ Resolve Inline References

☐ Export Only Local Inline References

☒ Export Inherited Inline References

☒ Include Calculated Attribute Values

☐ Only Export Selected Objects

☒ Only Export Leaf Objects

OK Cancel

Calculated Attribute Use Cases

Some of the countless ways to use a calculated attribute are defined in the following examples.

For a brief overview and introduction to some of the possibilities, see the **Introduction to Value Templates** topic.

Examples

For simple value templates that demonstrate the purpose and result of each function, see the following:

- Attribute
- Commercial
- Conditional
- Date
- List
- Number
- Other
- Text
- Variable

Scenarios

For more detailed scenarios, formula templates and results, see the following:

- Asset Push File Name
- Classification
- Commercial Data
- Comparing Localized Values
- Identifying Child Objects
- Including Parent Info on Child
- New Product Indicator
- Override
- Product Description
- Reference
- Total Cost of Materials
- Units

Introduction to Value Templates

This section provides a way to see the relationship between the attribute values, a value template, and the calculated attribute result.

Unit Considerations

In order to convert units relevant units must have a Base Conversion defined in the units area of System Setup, and the unit template field must be available when calculated attributes have a numeric validation base type.

In some of the examples below, the `prodval()` function includes the 'unitid'. This 'unitid' is only needed if a user wants to convert the values to a specific unit. If the unit is already the correct type of unit, there is no need to use `prodval("AttributeID", "Unitid")`, the user can just use `prodval("AttributeID")`.

Attribute Values

The functions 'prodval' and 'value' act the same in retrieving the attribute values in STEP.

The 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

Example Value Templates and Results

In the table below, the Description column describes some commonly used value template types, the Value Template column lists examples that can be copied and pasted into the Function Editor, and the Results column displays the resulting calculated attribute values when the example attribute values (listed above) are used with each value template.

Descripti on	Value Template	Result s
Box Shape: Displays if the bottom of the box is square or not.	<code>if(value('Length') = value('Width'), 'Square', 'Not Square')</code>	Not Square
Color:	<code>if(exact(value('Color Front'), value('Color Back')),</code>	Different

Description	Value Template	Results
Compares the front and back colors of the current product and displays the result.	'Same color front and back', 'Different colors front and back')	colors front and back
Dimensions: Displays the dimension of the current product in the form 'Height x Width x Length'	CONCATENATE (simplevalue('Height'), ' x ',simplevalue ('Width'), ' x ', simplevalue('Length'))	22 cm x 39 cm x 54 cm
Size for cabin baggage: An air carrier has a limit on the size of the cabin baggage, namely: 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.	<pre>IF (AND (OR (AND (PRODVAL ("Height", "unece.unit.CMT") <= 23, PRODVAL ("Length", "unece.unit.CMT") <= 40, PRODVAL ("Width", "unece.unit.CMT") <= 55), AND (PRODVAL ("Height", "unece.unit.CMT") <= 23, PRODVAL ("Length", "unece.unit.CMT") <= 55, PRODVAL ("Width", "unece.unit.CMT") <= 40), AND (PRODVAL ("Height", "unece.unit.CMT") <= 40, PRODVAL ("Length", "unece.unit.CMT") <= 55, PRODVAL ("Width", "unece.unit.CMT") <= 23)), 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")</pre>	Size OK for cabin baggage

Description	Value Template	Results
Size for special storage: The product must have either width or length greater than 40 cm, but not both.	<pre>IF (AND (OR (PRODVAL ("Length") >40, PRODVAL ("Width") > 40), NOT (AND (PRODVAL ("Length") > 40, PRODVAL ("Width") >40))), "OK - One and only one dimension > 40","Not OK - none or two dimensions > 40")</pre>	OK - One and only one dimension > 40
Size for standard package: Since a delivery service charges extra for oversized packages, this could be used to determine if a dimension exceeds 40 cm.	<pre>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")</pre>	Too big for standard package
Volume: Calculates and displays the volume of the current product.	<pre>prodval("Height", "unece.unit.CMT") *prodval("Width", "unece.unit.CMT") *prodval("Length", "unece.unit.CMT")</pre> Unit Template: <pre>iterate(stepurl2objid(stepurl("Unit", "unece.unit.CMQ")), 'stepid()')</pre>	46332 cm ³
Volume (Liter): Calculates the volume in Liter and displays with one decimal.	<pre>text(prodval("Height", "unece.unit.CMT") *prodval("Width", "unece.unit.CMT") *prodval("Length", "unece.unit.CMT")/1000,"0.0")</pre>	46.3

Attribute Examples

These example calculated attributes are written to use the data defined in the following tables.

For function syntax information and more detailed use cases, see the **Attribute Functions** documentation.

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

The example value templates require the **attributes** and values defined in the following table:

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

The example value templates require the **products** and values defined in the following table:

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

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor. Select the **Object** identified and click the Evaluate button to display the defined **Result**.

Value Template	Object	Result
<code>attrname('NUMBER1id')</code>	P-family-id	NUMBER1
<code>prodval('TEXT1id')</code>	P-family-id	family value
<code>prodval('NUMBER1id')</code>	P-family-id	2
<code>produnit('NUMBER1id')</code>	P-family-id	m
<code>prodval('NUMBER1id', 'unece.unit.MMT')</code>	P-family-id	2000
<code>prodvalsimple('NUMBER1id')</code>	P-family-id	2 m
<code>prodvalsimple('NUMBER1id', 'unece.unit.MMT')</code>	P-family-id	2000 mm
<code>getembeddedprefix(prodval('EMBEDDEDNUMBER1id'))</code>	P3id	prefix
<code>getembeddedvalue(prodval('EMBEDDEDNUMBER1id'))</code>	P3id	3
<code>getembeddedsuffix(prodval('EMBEDDEDNUMBER1id'))</code>	P3id	suffix
<code>prodvalsimple('EMBEDDEDNUMBER1id')</code>	P3id	prefix 3 m suffix
<code>prodvalsimple('EMBEDDEDNUMBER1id', 'unece.unit.MMT')</code>	P3id	prefix 3000 mm suffix

Value Template	Object	Result
<code>produnit('EMBEDDEDNUMBER1id')</code>	P3id	m
<code>unitconv('2','unece.unit.MTR','unece.unit.MMT')</code>	P3id	2000
<code>valuelovid('LOV1Attrid')</code>	P3id	LOV1id

Commercial Examples

These example calculated attributes are written to use the data defined in the following tables.

For function syntax information and more detailed use cases, see the **Commercial Functions** documentation.

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

The example value templates require the **Terms List Type** and **Terms List** defined in the following table:

Term List Type ID = PriceID			Terms List ID = TermsListID			
Product Name	Value	Unit - name(id)	Min Qty	Max Qty	Start	End
Pink Party Hat	\$7.79	\$ (iso4217.unit.USD)	1	11	Jan-1-15	Dec-31-15
Pink Party Hat	\$2.79	\$ (iso4217.unit.USD)	12	24	Jan-1-15	Dec-31-16
Blue Party Hat	£6.79	£ (iso4217.unit.GBP)	1	11	Jan-1-15	Dec-31-17
Blue Party Hat	£3.79	£ (iso4217.unit.GBP)	12	24	Jan-1-15	Dec-31-18

After creating the data above, open the Function Editor. Copy and paste any of the **Value Template** text below into the Function Editor, select the **Object** identified, and click the Evaluate button to display the defined **Result**.

Value Template	Object	Result
<code>termqtymax('PriceID', '1')</code>	Blue Party Hat	11
<code>termqtymin('PriceID', '2')</code>	Blue Party Hat	12
<code>termunit('PriceID', '1')</code>	Pink Party Hat	\$
	Blue Party Hat	£
<code>termunitid('PriceID', '1')</code>	Pink Party Hat	iso4217.unit.USD
	Blue Party Hat	iso4217.unit.GBP

Value Template	Object	Result
<code>termval('PriceID', '1', '2015-12-31 01:01:01')</code>	Pink Party Hat	\$7.79
<code>termval('PriceID', '2', '2015-12-31 01:01:01')</code>	Pink Party Hat	\$2.79
<code>termvalid('TermsListID', '2')</code>	Pink Party Hat	\$7.79

Conditional Examples

These examples do not require specific object data to produce the defined result.

For function syntax information and more detailed use cases, see the **Conditional Functions** documentation.

Copy and paste any of the **Value Template** text below into the Function Editor, select any object, and click the Evaluate button to display the defined **Result**.

Value Template	Result	Notes
<code>if (1=1, 'true', 'false')</code>	true	1 does equal 1 so the first result is returned
<code>if (not(1=1), 'true', 'false')</code>	false	NOT changes 'False to True' or 'True to False,' so the second result is returned
<code>if (exact('hello','you'), 'true', 'false')</code>	false	EXACT requires that the two values match, so the second result is returned
<code>if (or(1=1,exact('hello','you')), 'true', 'false')</code>	true	OR allows either 1=1 or an exact match, so the first result is returned
<code>if (and(1=1,exact('hello','you')), 'true', 'false')</code>	false	AND requires both functions to be true, so the second result is returned

Testing Using And, Or, and Not Functions

The example value templates require the **attributes** and values defined in the following table:

ATTRIBUTE Type	ID	Name	AttrDesc
Number-validated product attribute with length units	NUMBER1id	NUMBER1	NUMBER1 description
Text-validated product attribute	TEXT1id	TEXT1	TEXT1 description

The example value templates require the **products** and values defined in the following table:

PRODUCTS ID	Name	Parent	TEXT1id	NUMBER1id
P1id	P1	P-family-id	p1 value	4 m

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor. Select the **Object** identified and click the Evaluate button to display the defined **Result**.

Value Template	Object	Result	Notes
<code>and((value('TEXT1id')= 'p1 value'), (value('NUMBER1id')= '4'))</code>	P1id	1	Both conditions are true
<code>and((value('TEXT1id')= 'p0 value'), (value('NUMBER1id')= '3'))</code>	P1id	0	Both conditions are false
<code>or((value('TEXT1id')= 'p1 value'), (value('NUMBER1id')= '4'))</code>	P1id	1	One condition is true
<code>or((value('TEXT1id')= 'p0 value'), (value('NUMBER1id')= '3'))</code>	P1id	0	Both conditions are false
<code>not(value('TEXT1id')= 'p1 value')) // = 0 (FALSE) evaluated P1id</code>	P1id	0	Condition is true, NOT results in a return of 'false '
<code>not(value('TEXT1id')= 'p0 value')) // = 1 (TRUE) evaluated P1id</code>	P1id	1	Condition is false , NOT results in a return of 'true'

Testing Using If Function

These example calculated attributes are written to use a product with the attribute data defined in the following table.

Attribute	Length	Width	PowerCordLength
Value	75 cm	75 cm	2 m

Copy and paste any of the **Value Template** text below into the Function Editor, select the **Object** with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Result	Notes
<code>if(prodval('Length') = prodval('Width'), 'Product is Square.','Product is Oblong.')</code>	Product is Square.	Length does equal Width, so the first result is returned
<code>if(prodval("PowerCordLength") > 0,concatenate ('A ', prodvalsimple("PowerCordLength"), ' power cord is included. '), 'Power cord not included.')</code>	A 2 m power cord is included.	2 is greater than 0, so the first result is returned The value of '2 m' is extracted via the function PRODVALSIMPLE ("PowerCordLength"), and the surrounding text to that value was accomplished using the CONCATENATE function. If it is desired that the unit for meters (m) is not situated immediately after the value '2', then the function of PRODVALSIMPLE is not appropriate, but the following formula could be used instead.
<code>if(prodval("PowerCordLength") > 0,concatenate ('A ', prodval("PowerCordLength"), produnit("PowerCordLength"), ' power cord is included. '), 'Power cord not included.')</code>	A 2 m power cord is included.	The value of '2' is extracted via the function PRODVAL ("PowerCordLength"), and the unit of 'm' was extracted separately via the function PRODUNIT ("PowerCordLength").

Displaying a Value Using If Function

This example calculated attribute is written to use products with the attribute data defined in the following table.

	Weight
P2(P2id)	75
P3(P3id)	23
P4(P4id)	101

Copy and paste the **Value Template** text below into the Function Editor, select the **Object** identified, and click the Evaluate button to display the **Result**.

Value Template	Object	Result
If (value('Weight') > 100, 'A', if (value('Weight') < 50, 'B', 'C'))	P2id	C
If (value('Weight') > 100, 'A', if (value('Weight') < 50, 'B', 'C'))	P3id	B
If (value('Weight') > 100, 'A', if (value('Weight') < 50, 'B', 'C'))	P4id	A

Date Examples

These example calculated attributes are written to use the data defined in the following tables.

For function syntax information and more detailed use cases, see the **Date Functions** documentation.

These example calculated attributes are written to use a **product** with the attribute data defined in the following table.

Product ID	AttrDateID	StartDate (ISO Date)
P1id	2016-10-15	
P2id	1997-12-25	
P3id	2016-05-15	
P4id		2017-03-13

After creating the data above, open the Function Editor. Copy and paste any of the **Value Template** text below into the Function Editor, select the **Object** identified, and click the Evaluate button to display the defined **Result**.

Value Template	Object	Result
<code>formatdatetime(prodval('AttrDateid'), "EEE, MMM d, 'yy")</code>	P1id	Fri, Oct 15, '16
<code>formatdatetime(prodval('AttrDateid'), "yyyyy.MMMMM.dd")</code>	P2id	01997.December.25
<code>formatdatetime(prodval('AttrDateid'), "MMM d, yy")</code>	P3id	May 15, 16
<code>localizeddate(prodval('AttrDateid'))</code>	P3id when in Danish context	15-05-2016
<code>now()</code>	any when executed on 4 MAY 2016 just after midday	2016-05-04 12:06:25

Value Template	Object	Result
<pre>{d := value('StartDate'), x := localizedate(d, "en_GB"), y := localizedate(d, "en_US"), z := localizedate(d, "de_DE") } concatenate(d, "; ", x, "; ", y, "; ", z)</pre>	P4id	2017-03-13; 13-Mar-2017; Mar 13, 2017; 13.03.2017

List Examples

These example calculated attributes are written to use the data defined in the following tables.

For function syntax information and more detailed use cases, see the **List Functions** documentation.

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

The example value templates require the **attributes** and values defined in the following table:

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

The example value templates require the **products** and values defined in the following table:

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

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor. Select the **Object** identified and click the Evaluate button to display the defined **Result**.

Value Template	Object	Result
<code>list(iterate(subproducts(), 'prodval("TEXT1id")'), ', ')</code>	P-family-id	p1 value, p2 value, p3 value
<code>list(iterate(subproducts(), 'prodval("TEXT1id")'), ', ', ' & ')</code>	P-family-id	p1 value, p2 value & p3 value
<code>list(reverse(iterate(subproducts(), 'prodval("TEXT1id")')), ', ')</code>	P-family-id	p3 value, p2 value, p1 value
<code>list(iterate(subproducts(), 'concatenate(index, ": ", item)'), '\n')</code>	P-family-id	1. com.stibo.core.persistence.No dePO-343239, 2. com.stibo.core.persistence.No dePO-3432395, 3. com.stibo.core.persistence.No dePO-3432395 Note: 'com.stibo.core.persistence.NodePO-3432391,...' are internal representation of the nodes (sub products). This

Value Template	Object	Result
		happens when iterating subproducts and displaying them using the default variable 'item.'
list(iterate(subproducts(), 'concatenate(index," ",stepid())'), '\n')	P-family-id	P1id, P2id, P3id Note: Attribute values from the iterated nodes are displayed when called from inside the iterate function.
list(list2multivalue(value('AttributeA')))		value a<multisep/>value b<multisep/>value c Note: multivalue attribute with the values: "value a", "value b", and "value c"
list(multivalue2list(value('AttributeA')),', ')		value a, value b, value c Note: multivalue attribute with the values: "value a", "value b", and "value c"
list(sortnumeric(listconcatenate('2', '56', '11', '0', '21', '13')), ',')	{any}	0,2,11,13,21,56
list(unique(listconcatenate('a','b','a','c','b ')),', ')	{any}	a, b, c
listconcatenate(iterate(subproducts(), 'prodval("TEXT1id")'), iterate(subproducts(), 'prodval("NUMBER1id")'))	P-family-id	p1 valuep2 valuep3 value456
valuelovid('LOV1Attrid')	P3id	LOV1id
if(listcontains(iterate(subproducts(), 'prodval("TEXT1id")'), "p1 value"), "p1 is present", "p1 is NOT present")	any family <i>other than</i>	p1 is NOT present This is the same formula as shown previously except the list formula is written in full

Value Template	Object	Result
	P-family-id	within the LISTCONTAINS function
<code>listlen(iterate(subproducts(), 'prodval("TEXT1id")'))</code>	P-family-id	3
<code>listitem(iterate(subproducts(), 'prodval("TEXT1id")'),2)</code>	P-family-id	p2 value
<code>iterate(filter(subproducts(), 'prodval("number1id")>3'), 'concatenate("Name = ",stepname(),"; ID = ", stepid(), ".")')</code>	P-family-id	Name = P1; ID = P1id. Name = P2; ID = P2id. Name = P3; ID = P3id. extracts id of all child products where number1id is greater than 3
<code>{x := iterate(subproducts(), 'prodval("TEXT1id")') } if(listcontains(x,"p1 value"), "p1 is present","p1 is NOT present")</code>	P-family-id	p1 is present this is the same formula as shown following except the variable x is used to separate the list formula
<code>range(listconcatenate('1','5','8','6','7'), '-')</code>	{any}	1-8
<code>range(listconcatenate('1','5','8','6','7'), '-', ', ')</code>	{any}	1, 5-8
<code>sort(iterate(subproducts(), 'prodval("NUMBER1id")'))</code>	P-family-id	456

Number Examples

These examples do not require specific object data to produce the defined result.

For function syntax information and more detailed use cases, see the **Number Functions** documentation.

Copy and paste any of the **Value Template** text below into the Function Editor, select any object, and click the Evaluate button to display the defined **Result**.

Value Template	Result
<code>1.25*11</code>	13.75
<code>int(5.64)</code>	5
<code>int(-5.64)</code>	-6
<code>text(2.45, '000.0')</code>	002.5
<code>text(2.45, '.0')</code>	2.5
<code>text(2000.45, ',.0')</code>	2,000.5
<code>trunc(5.64)</code>	5
<code>trunc(-5.64)</code>	-5
<code>round(5.64, 0)</code>	6
<code>round(5.64, 1)</code>	5.6
<code>round(5.64, -1)</code>	10

Controlling Decimal Places

These example calculated attributes are written to use a product with the attribute data defined in the following table.

Attribute	height	width	length
Value	50.3 cm	135.26 cm	75.1 cm

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Result	Notes
<code>(value('length'))*(value('width'))</code>	10158.026	No formatting is applied to the resulting number.
<code>prodval('height', 'unece.unit.CMT') * prodval('width', 'unece.unit.CMT') * prodval('length', 'unece.unit.CMT')</code>	510948.7078	No formatting is applied to the resulting number.
<code>text(prodval('height', 'unece.unit.CMT') * prodval('width', 'unece.unit.CMT') * prodval('length', 'unece.unit.CMT'), "000,000.0")</code>	510,948.7	The TEXT function adds a thousands delimiter and reduces the decimal places to 1
<code>concatenate(prodvalsimple('height','unece.unit.CMT'), ' H x ', prodvalsimple('width','unece.unit.CMT'),' W x ', prodvalsimple('length','unece.unit.CMT'),' L')</code>	50.3 cm H x 135.26 cm W x 75.1 cm L	Static text strings provide a typical display for dimension information.
<code>concatenate(trunc(prodval('height')), " x ", trunc(prodval('width')), " x ", trunc(prodval('length')))</code>	50 x 135 x 75	The TRUNC function eliminates the decimal point and places.
<code>concatenate(int(prodval('height')), " x ", int(prodval('width')), " x ", int(prodval('length')))</code>	50 x 135 x 75	The INT function returns the integer part of the number.
<code>concatenate(round(prodval('height'),1), " x ",</code>	50.3 x 136 x	The ROUND

Value Template	Result	Notes
<pre>round(prodval('width'),0), " x ", round(prodval('length'),-1))</pre>	80	function, using each of the available precision values (1, 0, -1), controls the decimal places for each value being used.

Localized Number

These example calculated attributes are written to use a product with the attribute data defined in the following table.

Attribute	Diameter
Value	12,7
Locale Setting	'comma-is-decimal' method

Copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Result	Notes
<pre>{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)</pre>	<pre>12,7; 43.3578; NaN; 43,3578</pre>	If you attempt to make a math calculation on a number 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.

Other Examples

In the sections below, many other calculated attribute functions are grouped by purpose and are demonstrated in simple terms using a variety of value templates:

- **Classification Functions** can be used to access classification objects.
- **Completeness Functions** can be used to access the completeness score of an object.
- **Hierarchy Functions** can be used to access STEP hierarchy objects.
- **Lookup Table Functions** can be used to access lookup table data.
- **Object Functions** can be used to access object information.
- **Override Functions** can be used to access product override data.
- **Reference Functions** can be used to access object reference data.
- **Revision Functions** can be used to access revision information.
- **Workflow Functions** can be used to access workflow information.

For function syntax information and more detailed use cases, see the **Other Functions** documentation.

Classification Functions

These functions can be used to access classification objects.

The example calculated attribute is written to use a **product** with the classification data defined in the following table.

	References
Product1(P1)	ClassificationA ClassificationB

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Object	Result
<code>list(iterate(classifications(), "stepid()"), "/ ")</code>	P1	ClassificationA/ ClassificationB

Use Cases

For a scenario using the CLASSIFICATIONS function, see **Classification Scenario**.

Completeness Functions

These functions can be used to access the completeness score of an object.

For more information, see **Setting Up Completeness Metrics** in the **Data Quality** documentation.

Copy and paste the **Value Template** text below into the Function Editor, select an object set up for completeness metric, and click the Evaluate button to display the defined **Result**.

Value Template	Result
<code>completeness()</code>	51.282051282051284601948282215744
<code>round(completeness(),0)</code>	51

Hierarchy Functions

These functions can be used to access STEP hierarchy objects.

The example value templates require the **objects** and values defined in the following table:

Object	Parent	TEXT1id
P-family(P-family-id)	Product hierarchy root(PRoot)	product value
P1(P1id)	PFam(P-family-id)	p1 value
P2(P2id)	PFam(P-family-id)	p2 value
P3(P3id)	PFam(P-family-id)	p3 value
E-family(E-family-id)	Entity root(ERoot)	entity value
E1(E1id)	EFam(E-family-id)	e1 value
E2(E2id)	EFam(E-family-id)	e2 value

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Object	Result	Notes
<code>list(iterate(children(), 'concatenate(stepname())',', '))</code>	P-family-id	P1, P2, P3	These are products
<code>list(iterate(subproducts(), 'concatenate(stepname())',', '))</code>	P-family-id	P1, P2, P3	These are products
<code>list(iterate(children(), 'concatenate(stepname())',', '))</code>	E-family-id	E1, E2	These are entities
<code>list(iterate(subproducts(), 'concatenate(stepname())',', '))</code>	E-family-id		These are entities, not products
<code>parentid()</code>	P1id	P-family	
<code>list(iterate(path(), 'concatenate(stepname())',', '))</code>	P-family-id	Primary Product Hierarchy, Products, P-family	
<code>isbelow('step://product?id=P1id')</code>	P-family-id	0	FALSE
<code>isbelow('step://product?id=P-Family')</code>	P1id	1	TRUE
<code>{i:=path(), parent:=listitem(i,listlen(i)-1) } iterate(parent, 'concatenate("name=",stepname(),"; id=", stepid())')</code>	P1id	name=P-family; id=P-family-id	Use -2 (instead of -1) to get name / ID of grandparent, etc.

Evaluating Data on Child Objects

The example value templates require the **products** and values defined in the following table:

Parent	Products	BrandNameID	RemovalDate
Product hierarchy root(PRoot)	J-family(J-family-id)		
J-family-id	J1(J1id)	Alpha	2013-05-01
J-family-id	J2(J2id)	Alpha	2014-05-01
J-family-id	J3(J3id)	Alpha	2012-05-01
J-family-id	J4(J4id)	Zeta	
J-family-id	J5(J5id)	Zeta	2016-05-01

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Object	Result	Notes
<pre>{all:=iterate(subproducts(), "prodval('BrandNameID')"), 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), "")}</pre>	J-family-id	Alpha	Value for attribute ID 'BrandName' that occurs the most times in its children.
<pre>{attrid:='grandchildattrid' } listconcatenate(iterate(subproducts(),</pre>	J-family-id	AlphaAlphaAlphaZetaZeta	Use cautiously. This

Value Template	Object	Result	Notes
'iterate(subproducts(),"prodval(attrid)')')			operation can drain system CPU since all grandchildren are read for each product.
<pre> {msg := "Closest date that is less than the current date is: ", datenow := left(substitute(now(), "-", ""), 8), datelist := filter(iterate(children(), 'value ("RemovalDate)'), 'substitute(item, "-", "") < datenow'), neardate := listitem(reverse(sortnumeric(datelist)), 1) } if(exact(neardate, "N/A"), "None Newer", concatenate (msg, neardate)) </pre>	J-family-id	Closest date that is less than the current date is: 2016-05-01	Returns the value of 'RemovalDate' attribute on the child objects that is closest to the current date while still less than the current date. Returns 'None Newer' if no such child exists.

Lookup Table Functions

These functions can be used to access lookup table data.

For more information, see the **Creating a Transformation Lookup Table** topic or the **Transformation Lookup Table Example** topic of the **Data Quality** documentation.

The following templates use this lookup table data.

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

The examples in the following table do not require specific object data to produce the defined result. Copy and paste any of the **Value Template** text below into the Function Editor, select any object, and click the Evaluate button to display the defined **Result**.

Value Template	Result
<code>REPLACEVALUEBYLOOKUP('UOMLookupTable','EACH')</code>	Ea.
<code>REPLACEWORDBYLOOKUP('UOMLookupTable','EACH, DOZEN, CASE')</code>	Ea., Dz., Case

Object Functions

These functions can be used to access object information.

The example value templates require the **products** and values defined in the following table:

Product	Object Type	TEXT1id	TEXT1id Metadata
P-family(P-family-id)	Folder	Family value	Family meta value
P1(P1id)	Item	P1 value	P1 meta value

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Object	Result
<code>stepid()</code>	P1id	P1id
<code>stepname()</code>	P1id	P1

Value Template	Object	Result
<code>stepname('step://product?id=P1id')</code>	P1id	P1
<code>stepurl()</code>	P1id	step://product?id=P1id
<code>stepurl('Product','P1id')</code>	P1id	step://product?id=P1id
<code>stepobjecttype()</code>	P1id	Item
<code>stepobjecttype('step://product?id=P1id')</code>	P1id	Item
<code>iterate(stepurl2objid(stepurl('Product','Level1-109267')),'stepname()')</code>	{any}	P1id
<pre>{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)</pre>	P-family-id	'Family meta value' is the metadata on the 'TEXT1id' attribute which contains the value: Family value

Override Functions

These functions can be used to access product override data.

The example calculated attributes below are written to use the hierarchy objects defined in the following table.

Family	Folder	Overrides
Batteries(BatteriesID)	Battery Items(BatteryItemsID)	Override 1(Override1ID) Override 2(Override2ID) Override 3(Override3ID)

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Template	Object	Results
<code>overriddenproduct('Override1ID')</code>	Override1ID	Batteries
<code>overriddenproduct('Batteries')</code>	Override1ID	N/A
<pre>{x := listlen(overriddenby()), z := iterate(overriddenby(), 'stepname()'), z := list2multivalue(z), z := substitute(z, "<multisep/>", "") } concatenate("This product is overridden by ", x, " product override(s): ", z)</pre>	BatteriesID	This product is overridden by 3 product override(s): Override 2 Override 1 Override 3
<pre>{x := iterate(overriddenproductobject(), 'stepname()'), z := listitem(x,1) } concatenate('This PO overrides the product "', z, '"')</pre>	Override1ID	This PO overrides the product "Batteries"
<code>list(iterate(productoverrideparents(), "stepid()"), "/"</code> <code>)</code>	BatteryItemID	Override2ID/Override1ID/Override3ID

Reference Functions

These functions can be used to access object reference data.

The example calculated attributes below are written to use the referenced object defined in the following table.

Product	Reference Type	Reference Target
P1(P1id)	BillOfMaterials	P3(P3id)

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Object	Result
<code>list(iterate(iterate(references('product', 'BillOfMaterials'),'referencetarget()'),'stepname()'),'')</code>	P1id	P3

Use Cases

For a scenario using the REFERENCES function, see **Reference Scenario**.

For a scenario using the REFERENCES and REFERENCETARGET functions, see **Total Cost of Materials Scenario**.

Revision Functions

These functions can be used to access revision information.

The example calculated attributes below are written to use the revisions defined in the following table.

Product	Revision #	Create Date	Edit Date	Major	User	Comment
P-family(P-family-id)	1.1	Wed Jul 12 12:06:16 EDT 2017	Wed Jul 12 12:26:55 EDT 2017		USERZ	Complete Approval
P-family(P-family-id)	1.0	Mon May 08 11:34:51 EDT 2016	Tue May 09 15:09:14 EDT 2016	X	USERY	Comment added by user
P-family(P-family-id)	0.1	Mon Jun 15 16:27:04 EDT 2015	Mon Jun 15 16:27:04 EDT 2015		USERW	Auto Generated

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Object	Result
<code>list(iterate(revisions(), 'revisioncreatedate()'),'')</code>	P-family-id	2015-06-15 16:27:04, 2016-05-08 11:34:51, 2017-07-12 12:06:16
<code>revisionname()</code>	P-family-id	1.1

Value Template	Object	Result
revisioneditedby()	P-family-id	USERZ
revisioncomment()	P-family-id	Complete approval
revisionismajor()	P-family-id	0
revisioncreatedate()	P-family-id	2017-07-12 12:06:16
revisioneditdate()	P-family-id	2017-07-12 12:26:55
firstrevisioncreatedate()	P-family-id	2015-02-13 11:36:39

Workflow Functions

These functions can be used to access workflow information.

The example calculated attributes below are written to use the referenced object defined in the following table.

Product	Current Workflow	Current State
P1(P1id)	Workflow5	State-C
P1(P1id)	Workflow6	State-A

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Object	Result
workflowsids()	P1id	Workflow5Workflow6
currentworkflowstates('Workflow6')	P1id	State-A

Text Examples

In the sections below, the calculated attribute text functions are grouped by purpose and are demonstrated in simple terms using a variety of value templates:

- **Escape Functions** can be used to allow for special characters during text manipulation.
- **Identify Functions** can be used to display the defined data.
- **Modify Functions** can be used to modify defined data.
- **Regular Expression Functions** can be used to search and modify text based on a defined template.
- **Representative Text Functions** can be used to produce an alternative value for the defined text.
- **Test Functions** can be used to perform a test and return a result.

For function syntax information and more detailed use cases, see the **Text Functions** documentation.

Escape Functions

These functions can be used to allow for special characters during text manipulation.

The examples in the following table do not require specific object data to produce the defined result. Copy and paste any of the **Value Template** text below into the Function Editor, select any object, and click the Evaluate button to display the defined **Result**.

Value Template	Result	Notes
<code>escapemerge(';', '/', "Red", "Orange; Blue", "Violet")</code>	Red;Orange; Blue;Violet	Write semicolon (;) (split character) after each element and include slash (/) (escape character) when split character is found in text.
<code>escapesplit(';', '/', 'Red;Orange/; Blue;Violet')</code>	RedOrange; BlueViolet	Write elements without semicolon (;) (split character) unless it is escaped with a slash (/) (escape character).

Identify Functions

These functions can be used to display the defined data.

The examples in the following table do not require specific object data to produce the defined result. Copy and paste any of the **Value Template** text below into the Function Editor, select any object, and click the Evaluate button to display the defined **Result**.

Value Template	Result	Notes
<code>left('hello')</code>	h	1 character from left
<code>left('hello',3)</code>	hel	3 characters from left
<code>right('hello')</code>	o	1 character from right
<code>right('hello', 3)</code>	llo	3 characters from right
<code>mid('hello',2,3)</code>	ell	3 characters starting at character 2
<code>len('hello')</code>	5	number of characters in string

Modify Functions

These functions can be used to modify defined data.

The examples in the following table do not require specific object data to produce the defined result. Copy and paste any of the **Value Template** text below into the Function Editor, select any object, and click the Evaluate button to display the defined **Result**.

Value Template	Result	Notes
<code>concatenate("hello", "hallo")</code>	hellohallo	join multiple text strings into a single text string
<code>trim(' hello ')</code>	hello	remove spaces before and after text
<code>upper('hello')</code>	HELLO	convert to uppercase letters
<code>lower('HELLO')</code>	hello	convert to lowercase letters
<code>proper("hello, are you john's son?")</code>	Hello, Are You John'S Son?	capitalize the first letter of each word, including capitalization after punctuation

Value Template	Result	Notes
<code>firstnamefirst("Johnson, John S.")</code>	John S. Johnson	transpose words using comma as separator
<code>firstnamefirst("Joe Dalton")</code>	Joe Dalton	transpose words using comma as separator
<code>replace('hello',2,3,'el')</code>	helo	insert 'el' in place of 3 characters, starting at character 2
<code>substitute('hellohellohello', 'ell', 'al')</code>	halohalohalo	update all instances of 'ell' with 'al'
<code>substitute('hellohellohello', 'ell', 'al', 2)</code>	hellohalohello	update only instance 2 of 'ell' with 'al'

Formatting Text

The example calculated attributes below use an object with the attribute values defined in the following table.

Attribute>	height	width	length
Product1	50 cm	135 cm	75 cm
Product2	50 cm		75 cm

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

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Object	Result	Notes
<code>concatenate(prodval('height'), prodval('width'), prodval('length'))</code>	Product1	5013575	Concatenating only the values excludes the unit information and results in an unformatted number.

Value Template	Object	Result	Notes
<pre>concatenate(prodvalsimple('height','unece.unit.CMT'),' H x ', prodvalsimple('width','unece.unit.CMT'),' W x ', prodvalsimple('length','unece.unit.CMT'),' L')</pre>	Product1	50 cm H x 135 cm W x 75 cm L	Static text strings provide a typical display for dimension information.
<pre>list(listconcatenate(prodval("height"), prodval("width"), prodval("length")),'*')</pre>	Product2	50*75	Since the width value is empty, this needs to be concatenated without an extra separator '*' so the LISTCONCATENATE and LIST functions are used. Alternatively, the IF function can be used to check whether a value is empty or not before concatenating, but would require additional code to check for each concatenating value.

The example calculated attributes below use an object with the attribute values defined in the following table.

Attribute>	prodname
Product3	ACME-BP-1001 Super XBP

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

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Object	Result	Notes
<code>replace(prodval('prodname'), 6,2,'Boardroom Projector Series')</code>	Product3	ACME- Boardroom Projector Series-1001 Super XBP	Update the first occurrence of BP while leaving the second instance untouched. Note: If the position (6) or length (2) to be replaced was miscounted, whatever text is found at the location will be replaced.
<code>substitute(prodval('prodname'), '-BP-', ' Boardroom Projector Series ')</code>	Product3	ACME Boardroom Projector Series 1001 Super XBP	Update all occurrences of -BP-, so that the second instance of BP is not changed.
<code>substitute(prodval('prodname'), 'BP','Boardroom Projector Series')</code>	Product3	ACME- Boardroom Projector Series-1001 Super XBoardroom Projector Series	Update all occurrences of BP.
<code>substitute(prodval('prodname'), 'BP','Boardroom Projector Series',1)</code>	Product3	ACME- Boardroom Projector Series-1001 Super XBP	Update only the first occurrence of BP.

Regular Expression Functions

These functions can be used to search and modify text based on a defined template.

For more information on creating regular expressions, see the **Regular Expression** documentation.

The examples in the following table do not require specific object data to produce the defined result. Copy and paste any of the **Value Template** text below into the Function Editor, select any object, and click the Evaluate button to display the defined **Result**.

Value Template	Result	Notes
<code>regexpsearch('e.*he', 'hellohello')</code>	2	2 is the start point for 'ellohe' as a case sensitive match for the regexp
<code>regexpsearch('[ABC]\\d', 'HELLOHA2MSTER')</code>	7	7 is the start point for 'A' + any digit as a case sensitive match for the regexp. [] indicates only one of the enclosed letters is required. \\d indicates a digit.
<code>regexpsearchi('[efg] [ojk]', 'HELLOHELLO')</code>	2	2 is the start point for 'E' as a case insensitive match for the regexp looking for either 'e, f, g' or 'o, j, k'. [] indicates only one of the enclosed letters is required. indicates 'or.'
<code>regexpsearchi('[abc] [ojk]', 'HELLOHELLO', 3)</code>	5	Searching starting at position 3, 5 is the start point for 'O' as a case insensitive match for the regexp looking for either 'a, b, c' or 'o, j, k'. [] indicates only one of the enclosed letters is required. indicates 'or.'
<code>regexpsubstitute("kat;abe", "(.*)" ; (.*)", "\$2\$1")</code>	abekat	Save all characters before the semicolon in Group 1, save all characters after the semicolon in Group 2; display Group 2 then Group 1, without any space or the semicolon
<code>regexpsubstitute("kat abe", "(\\w*)\\s*(\\w*)", "\$2\$1")</code>	abekat	Find first word (and save it in Group 1) followed by a space, then find another word (and save it in Group 2), display Group 2 then Group 1 without the space. \\w indicates a word and \\s indicates a space.

Representative Text Functions

These functions can be used to produce an alternative value for the defined text.

The examples in the following table do not require specific object data to produce the defined result. Copy and paste any of the **Value Template** text below into the Function Editor, select any object, and click the Evaluate button to display the defined **Result**.

Value Template	Result
<code>soundex("hello")</code>	H400
<code>metaphone3("Joel Dalton")</code>	JLTLTN
<code>metaphone3alternate("Joel Dalton")</code>	ALTLTN

Test Functions

The functions can be used to perform a test and return a result.

The examples in the following table do not require specific object data to produce the defined result. Copy and paste any of the **Value Template** text below into the Function Editor, select any object, and click the Evaluate button to display the defined **Result**.

Value Template	Result	Notes
<code>exact("hello", "hallo")</code>	0	Exact match not found
<code>exact("hello", "hello")</code>	1	Exact match is found
<code>if (exact('hello','you'), 'true', 'false')</code>	false	Exact match is not found so the second result is returned
<code>find('llo','hellohello')</code>	3	The position of the first occurrence
<code>find('llo','hellohello', 4)</code>	8	The position of the first occurrence found after character 4
<code>find('llo','hellohello')</code>	N/A	Match is not found

Testing Text

The example calculated attributes below use an object with the attribute values defined in the following table.

Attribute>	PowerCordIncluded
Product4	Yes
Product5	yes

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Object	Result
<pre>if(exact(prodval('PowerCordIncluded'),'Yes'), 'A 2 meter power cord is included.', 'Power cord is not included.')</pre>	Product4	A 2 meter power cord is included.
	Product5	Power cord is not included.

Variable Examples

These example calculated attributes are written to use the data defined in the following tables.

For function syntax information on and more detailed use cases, see the **Variable Functions** documentation.

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

After creating the data above, copy and paste any of the **Value Template** text below into the Function Editor, select the object with the defined data, and click the Evaluate button to display the **Result**.

Value Template	Object	Result
<pre>{txt:=prodval('TEXT1id'), pos:=find('p3', txt)} if(pos='N/A', 'no P3', replace(txt, pos, 2, '??'))</pre>	P3 P2	?? value no P3
<pre>{total:=0, loop:=iterate(listconcatenate(1,4,8,13), 'total:=total+item') } concatenate(total)</pre>	{any}	26

Concatenate Function with Variables

Consider a calculated attribute that will concatenate two different attribute values, where they style of each should be different. Use character tags in STEP to format (or style) text attributes. Then convert those tags to either paragraph or character tags for the output media. For information on character tags, see **Tags**.

The example calculated attribute is written to use an object with the **attributes** defined in the following table.

Attribute>	ProductName	KeyLetter
Mandatory	Yes	No
Character Tag	[PN]	[KL]

The example calculated attributes below use an object with the attribute values defined in the following table.

	ProductName	KeyLetter	Required Result
Product1(P1)	Acme Widget	A	[KL]A. [PN]Acme Widget
Product2(P2)	Acme Widget		[PN]Acme Widget

A calculated attribute can produce the same result with or without variables as demonstrated in the table below.

Value Template With Variables	Value Template Without Variables
<pre> {pnnokey := concatenate(' [PN] ',prodval ("ProductName")), pnwkey := concatenate(' [KL] ',prodval("KeyLetter"), '. [PN] ',prodval("ProductName")) } if (exact (prodval ("KeyLetter"), ''),pnnokey,pnwkey) </pre>	<pre> if (exact (prodval ("KeyLetter"), ''), concatenate (' [PN] ',prodval ("ProductName")), concatenate (' [KL] ',prodval ("KeyLetter"), '. [PN] ',prodval ("ProductName"))) </pre>

Variables

In this example, two variables allow the actual value template (the last line) to be shorter and more legible to the reader:

- Variable 'pnnokey' calculates the result when no KeyLetter value exists.
- Variable 'pnwkey' calculates when there is a KeyLetter value.

In both examples, testing for a blank value is achieved by using the EXACT function to compare against two (2) single quote marks (') with no spaces between them.

Find Function with Variables

The example calculated attributes below use an object with the attribute values defined in the following table.

	ProductName(prodname)
Product1(P1)	ACME-BP-1001 Super XBP

A calculated attribute can produce the same result with or without variables as demonstrated in the table below.

Value Template	Result	Notes
<pre>{x := find('-BP-', prodval("prodname")), y := 'Boardroom Projector Series' } if(exact(x, 'N/A'), '', replace(prodval("prodname"), x+1, 2, y))</pre>	ACME- Boardroom Projector Series-1001 Super XBP	The IF function checks to see if the value of x is 'N/A' - If the value is 'N/A' the result is no output, as indicated by the blank value of two single quote marks. - If the value is not 'N/A', then the two (2) characters starting at position x+1 (6, in this example), are replaced by the value of y.

Variables

- Variable 'x' finds the position of the text string that will be replaced using the FIND function. In this example, the position of the text '-BP-' within the 'prodname' attribute value is needed. The function returns the numeric value of the position, in this example, 5 is returned since the first dash is at position 5.
- Variable 'y' supplies the replacement text string.

Asset Push File Name Scenario

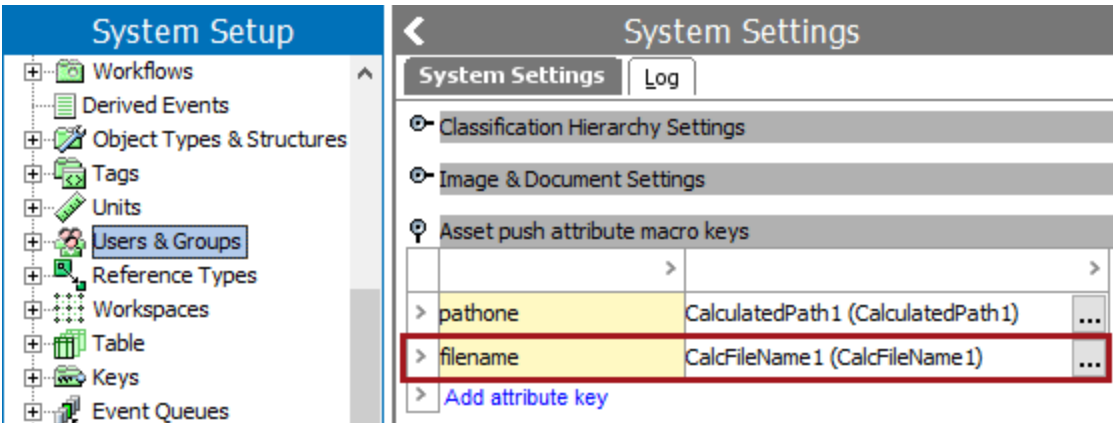
A calculated attribute can be used to create a custom file name for use with assets exported from STEP using the Asset Push functionality.

In this scenario, the file name of assets will be generated to contain the ID of the asset, appended with the date and timestamp of the upload. Of course, other attribute values may be included in the file name generation, including (but not limited to) Description Attributes values on the asset and any other attribute value from the asset's System Attribute flipper.

For information on Asset Push, see the **Asset Push** topic in the **Digital Asset Exchange** documentation.

Assumptions

- 1. An asset push attribute macro key with the Attribute Key value 'filename' is created to hold the calculated attribute 'CalcFileName1(CalcFileName1)'. A macro key attribute must be a description attribute. For more information on this macro, see **Asset push attribute macro keys** topic within the **Users and Groups** section of the **System Setup / Super User Guide** documentation.



- 2. The Asset Push Configuration 'Web85x85' will use the 'filename' macro key within its Relative Path Template, in order to construct the full path and file name of the assets pushed with this configuration. For details on creating a Relative Path Template, see the **Relative Path Template** documentation.

System Setup

- Event Queues
- Standard Configuration
 - raw-approved
 - raw-main
 - rawmain-assetname
 - Web150x150
 - Web300x300
 - Web300x300FixedS
 - Web85x85

Web85x85 - Asset Push Configuration

Asset Push Configuration
Asset Push Configuration Statistics
Log

Name	Value
ID	Web85x85
Name	Web85x85
Notification Email	
Workspace	Main
Image Conversion	<source>
Relative Path Template	\$configID\$/\$contentdimensionpointsID\$/\$IDpath\$/\$attribute:filename\$. \$extension\$

3. In this scenario, the asset being pushed is '0227658 11CC(0227658-11CC)' and the following screenshot includes some of the available description attribute values. In the Asset editor, the metadata system default attribute 'Upload Time(asset.uploaded)' is found under the System Properties flipper. The value of this asset is used to produce the date and time of the upload. The CalcFileName1 attribute displays the concatenation of the ID, upload date, and upload time, as described in the solution below.

0227658 11CC rev.1.2 - Images & Documents

Images & Documents
References
Referenced By
Status
State Log
Tasks

Description

Name	Value
ID	0227658-11CC
Name	0227658 11CC
Object Type	TIFF Image
Revision	1.2 Last edited by USERJ on Wed May 27 12:45:58 EDT 2015
Approved	✓ Approved on Wed May 27 12:45:58 EDT 2015
CalcFileName1	0227658-11CC_20150423_100752

System Properties:

Name	Value
Filename	abc 0227658-11CC.tif
Format	abc TIFF (Tagged Image File Format image)
MIME Type	abc image/tiff
Upload Time	abc 2015-04-23 10:07:52

Solution

Approach

A description calculated attribute 'CalcFileName1(CalcFileName1)' is made valid for all relevant assets, and includes two (2) variables:

- The **uploaddate** variable extracts the date of the upload, for example 2017-06-06, removing the dash or hyphen (-) character. For purposes of this example, assume that the requirement is not to have any separators in the date value.
- The **uploadtime** variable extracts the time of the upload, for example 14:53:28, removing the colon (:) character. Apart from the fact that this is an illegal character in a file name, again assume that the preference is not to use any separator in the time value.

Value Template

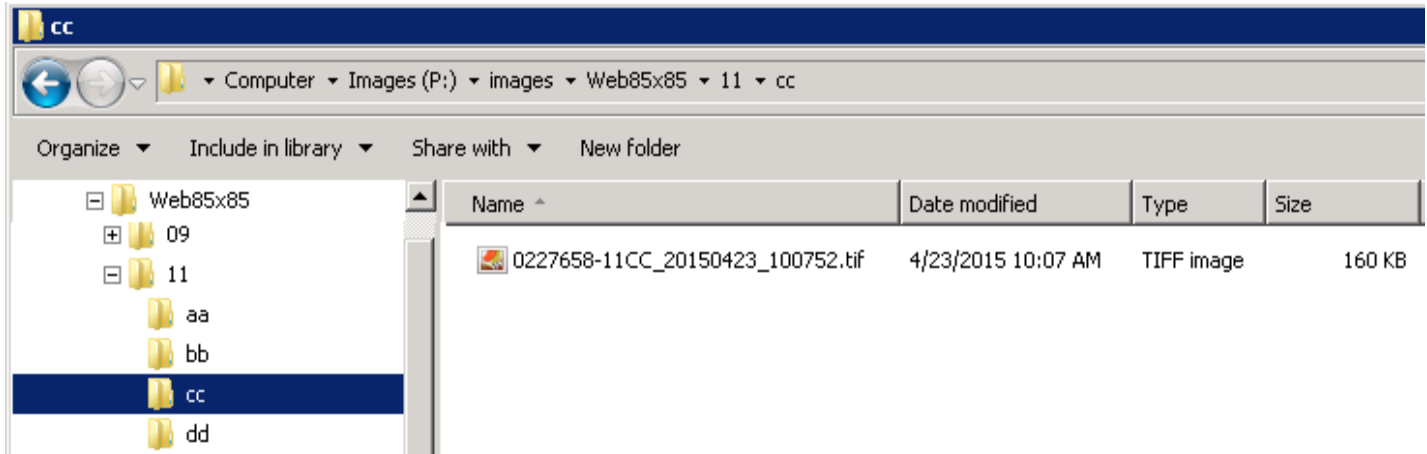
```
{uploaddate := substitute(left(value('asset.uploaded'), 10), "-", ""),
uploadtime := substitute(left(value('asset.uploaded'), 8), ":", "")
}

concatenate(stepid(), "_", uploaddate, "_", uploadtime)
```

Results

The pushed asset file is saved to the path generated by the Relative Path Template within the Asset Push Configuration, and includes the file name generated by the calculated attribute. The calculate attribute separates each of the elements of the file name listed below with an underscore (_):

- STEP ID = 0227658-11CC
- Upload date = 20150423
- Upload time = 100752



Classification Scenario

An asset is linked to two classifications.

Tree

Presentation & Home Theater Projectors

ABC-1001-103

DEF-7001-100

DEF-7001-103

GWM-1001-200

JEU-7001-103

Projector 15

Projector 15 rev.1.1 - Re

Images & DocumentsReferencesReferenced ByStatusState LogTasks

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.

Tree

JKL-5001-103

KJR-7001-103

KQM-1001-200

LMN-6001-100

LMN-6001-103

OPQ-1001-200

OPW-1001-200

PQR-3001-100

PQR-3001-103

Projector 15

Projector 6

Projector 15 rev.1.1 - Re

Images & DocumentsReferencesReferenced ByStatusState LogTasks

Description

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.

Tree

Office Equipment

Projectors

Pens, Pencils, Rollerballs, and Sets

Chairs

Specialty Office Chairs

Executive Chairs

Office Chairs

Office Chair

Office Chair rev.0.2

Product

References

Referenced By

Images & Documents

Commercial

Tables

Category Profile

Pr

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

Tree

Office Equipment

Projectors

Pens, Pencils, Rollerballs, and Sets

Chairs

Specialty Office Chairs

Executive Chairs

Office Chairs

Office Chair

Office

Product

References

Referenced By

Images & Documents

Commercial

Tables

Description

Name	Value
Name	Office Chair
Object Type	Product
GetClassificationsOfProducts	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.

Tree

Hand Tools

Office Equipment

Projectors

Classroom Type Projectors (Class A)

Home Theater

Office Projectors

Portable Projectors

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

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Tables

Category Profile

Proof View

Packaging Hi

Reference Type	Target	prod2prod2class
Default		
SupplierLink	Supplier Portal/My_Supplier/My_Supplier Products	
WebLink	Publication Structure/Web Structure/Chairs/Executive Chairs/Ergonomic Executive Chairs	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.

Get CPL Classifications Using Classification Product Links

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

The calculated attribute formula could be written as:

```
{meta := iterate(iterate(classificationproductlinks(), 'CPLClassification()'), 'value  
("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 multisep 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 P

Product

Sub Products

References

Referenced By

Images & Do

Name

>

>

Value

>

GetMetaDataFromClassesLinkedToProducts

fx

Peter Piper/Molly Malone

Commercial Data Scenario

This use case describes two ways to display commercial data (price list information) as an attribute value using a calculated attribute. These options allows you to use price lists and display a single price value per object in the object attributes.

Price List in a Publication Structure

The formula for this calculated attribute is:

```
(TermVal('SalesPrices', '0'))
```

In this example, object type of the commercial price list is 'SalesPrices'.

Place the commercial price list within the publication structure and select the appropriate Workbench menu, view, and version.

For more information on commercial data, see **Commercial Data** in the **Commercial Data** guide.

Price List in an eCatalog Structure

The formula for this calculated attribute is:

```
TermValid('PriceListID', '1')
```

In this example, the ID of the commercial price list is 'PriceListID'.

For more information on commercial data, see **eCatalogs** in the **eCatalogs** guide.

Comparing Localized Values Scenario

Calculated attributes can be used to identify and then compare an inherited value with a localized value. For example, an attribute named 'Warranty' is valid both at the product family level and at the item level (child of the product family). An export include objects with a value entered at the item level that is different from the value entered at the product family level. In other words, only when the value is localized.

Assumptions

1. Report only when the local value is different from the inherited value.
2. Do not report if a value has been entered on the item that is the same as the inherited value.

Solution

Approach

The approach follows the steps below:

1. A list of the product path is generated.
2. The value for the Warranty attribute is calculated from the parent of the child product (item).
3. The Warranty value for the child product is obtained.
4. The Warranty value for the child product is compared to the Warranty value for the parent of the child product.
5. If different, then both the inherited and local values are displayed.
6. If same, then the results value is blank.

Value Template

```
{i:=path(),
parent:=listitem(i,listlen(i)-1),
parentvalue := iterate(parent, 'value("Warranty)'),
localvalue := value("Warranty")
}
if(exact(parentvalue,localvalue), "", concatenate("Parent Value = ", parentvalue, ", Local
Value = ",localvalue))
```

Results

The value for the Warranty attribute on the parent node (Writing Pens) is 24.

The first child product has inherited the value for Warranty from the parent (indicated by the green triangle), so the calculated attribute result is blank.

Tree

Writing Pens

ABC-2500-1012

ABC-2500-1013

ABC-2500-1014

ABC-2500-1012

Images & DocumentsCommercialTablesCategory

ProductSub Products

Description

Item Brand Information

Name	>	>	Value
Solution		fx	
Warranty		abc	24

The second child product has a localized value, but it is the same as the value on the parent (indicated by the green triangle), so the calculated attribute result is blank.

Tree

Writing Pens

ABC-2500-1012

ABC-2500-1013

ABC-2500-1014

ABC-2500-1013

Images & DocumentsCommercialTablesCategory

ProductSub Products

Description

Item Brand Information

Name	>	>	Value
Solution		fx	
Warranty		abc	24

The third child product has a localized value that is different from the value on the parent, so the calculated attribute result shows both the parent and the local values.

Tree

Writing Pens

ABC-2500-1012

ABC-2500-1013

ABC-2500-1014

ABC-2500-1014 rev.0.1 - Pr

Images & DocumentsCommercialTablesCategory ProfileProof

ProductSub Products

Description

Item Brand Information

Name	>	>	Value
Solution		fx	Parent Value = 24, Local Value = 36
Warranty		abc	36

Identifying Child Objects Scenario

It can be beneficial to use a calculated attribute to easily identify if a product family has any children. For example, it can be beneficial to run an export of product families, and have a clear indication of whether the family has no child products, one single child product, or more than one child product.

Assumptions

- 1. The calculated attribute is valid for specific 'parental' product object types.
- 2. The subsequent export specifies the appropriate object type(s) to be exported.
- 3. This simple calculation will check all child products below the selected product family object type, which includes all subordinate levels.
- 4. If a calculation is required to restrict the levels to a specific number of children, a more sophisticated calculation will be required.

Solution

Approach

A simple list of child products will be obtained. The list will then be checked to see if it has 0 length (no children) or if it has a length of 1 (a single child product), and if neither of those two criteria are met then the family is assumed to have more than one child product.

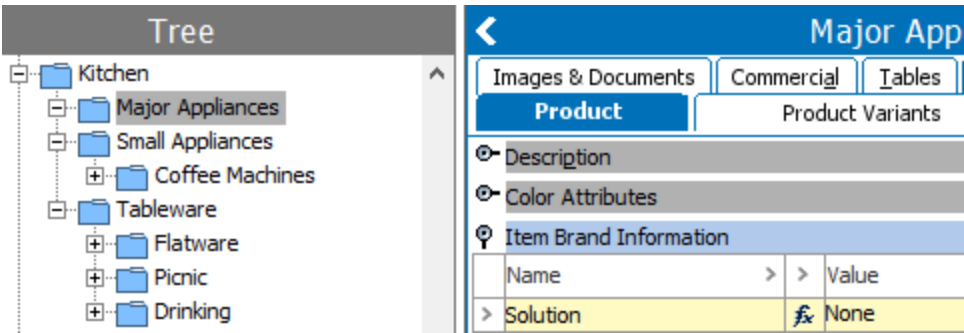
Value Template

```
{childlist := subproducts()}  
  
if(listlen(childlist)=0, "None", if(listlen(childlist)=1, "Single Child", "Multiple  
Children"))
```

For information on adding a value template, see the **Attribute Editor** section of the **Function Editor** documentation.

Results

The first node selected has no children, so 'None' is displayed.



The second node selected has one child, so 'Single Child' is displayed.

Tree

Kitchen

Major Appliances

Small Appliances

Coffee Machines

Tableware

Flatware

Picnic

Drinking

Small Appliance

Images & DocumentsCommercialTablesCatego

ProductProduct Variants

Description

Color Attributes

Item Brand Information

Name > > Value

> Solution & Single Child

The third node selected has more than one child, so 'Multiple Children' is displayed.

Tree

Kitchen

Major Appliances

Small Appliances

Coffee Machines

Tableware

Flatware

Picnic

Drinking

Tableware

Images & DocumentsCommercialTablesCate

ProductProduct Variants

Description

Color Attributes

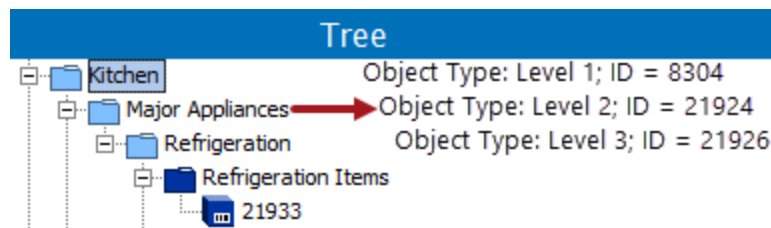
Item Brand Information

Name > > Value

> Solution & Multiple Children

Including Parent Info on Child Scenario

Calculated attributes can be used to automate the task of including parent information on a child. This topic addresses two options: when each individual child item should display the ID of the specified parent, and when the ID and name of each of the specified parent object types should display using multiple calculated attributes.



Assumptions

1. The Object Types in the path are all unique.
2. The Object Type requested in the calculation exists, otherwise the calculation results in the text 'N/A'.

Solution Option 1

Display only one parent using one calculated attribute.

Approach

The first three variables of the calculation will specify the object type name that is required, and will generate two lists, one for the IDs of the product nodes in the path, and the other for the object type names in the path. These two lists will have identical lengths. The 'getid' variable then locates the index number of the required object type name in the one list and returns the corresponding value of the product ID in the other list.

Value Template

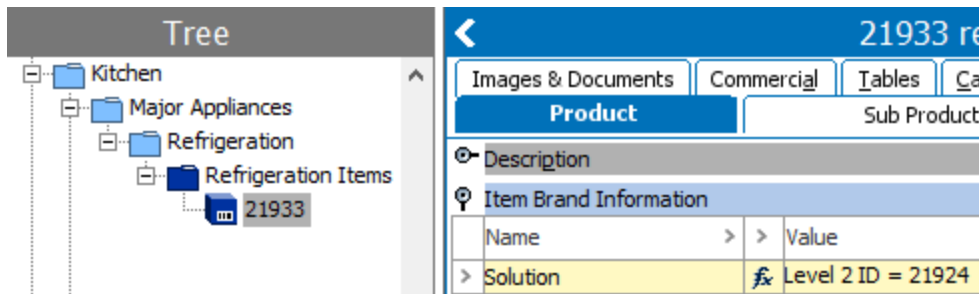
```

{objtype := "Level 2", //This is the object type NAME
plist := iterate(path(), 'stepid()'),
idlist := iterate(path(), 'stepobjecttype()'),
getid := listitem(filter(plist, 'exact(objtype, listitem(idlist,index))'),1)
}
concatenate("Level 2 ID = ",getid)

```

Results

The ID of the Level 2 object type that is parent to the item (ID = 21933) is reported by the calculated attribute.



Solution Option 2

Display information for multiple parents using the same calculated attribute copied on each level to display each object type.

Assumptions

1. A total of three calculated attributes are required (for Level 1, Level 2, and Level 3).
2. The value template for each would be nearly identical, but the first variable of each formula would identify the unique object type.

Approach

The following two additional variables are added to the previous value template: nlist and getname; and the concatenate formula has been modified accordingly.

Value Template

```
{objtype := "Level 1", //This is the object type NAME
plist := iterate(path(), 'stepid()'),
nlist := iterate(path(), 'stepname()'),
idlist := iterate(path(), 'stepobjecttype()'),
getid := listitem(filter(plist, 'exact(objtype, listitem(idlist,index))'),1),
getname := listitem(filter(nlist, 'exact(objtype, listitem(idlist,index))'),1)
}
concatenate("Level = ", getid, " (", getname,")")
```

Results

Each of the identified parent object types to the item (ID = 21933) are reported by separate calculated attributes.

Tree

Kitchen

Major Appliances

Refrigeration

Refrigeration Items

21933

21933 rev.0.21 - Product

Images & DocumentsCommercialTablesCategory ProfileProduct

ProductSub Products

Description

Item Brand Information

Name		Value
> Solution 1	fx	Level 1 = 8304 (Kitchen)
> Solution 2	fx	Level 2 = 21924 (Major Appliances)
> Solution 3	fx	Level 3 = 21926 (Refrigeration)

New Product Indicator Scenario

A calculated attribute can be used to automate the task of marking a product as new, or manually updating a field to indicate an item is new. For example, in a print application, a 'new' indicator needs to be generated if any of the products in a product family are designated as new. This could result in the display of a logo asset (such as the one in the image below) on a catalog section.

New2 rev.1.1 - Images & Documents		
Images & Documents	References	Referenced By
Status	State Log	Tasks
Description		
Name	>	Value
ID	>	232210
Name	>	New2
Object Type	>	Logo
Revision	>	1.1 Last edited by USERJ on Fri Jul 14 10:49:29 EDT 2017
Approved	>	✓ Approved on Fri Jul 14 10:49:29 EDT 2017



Assumptions

1. This calculated attribute will be used in a STEP'n'design product template.
2. The resulting value from this calculated attribute will either be changed via a transformation action into a call out for a logo, or a Stibo tag will be established that calls out the logo.
3. The calculated attribute will be made valid at the product family level (or product override if that is appropriate).

Approach

All the children in the family (or override if used) will be inspected for the value of a particular attribute (in this case the attribute is 'New Flag').

If any product has a value of 'Yes' as the New Flag value, the calculated attribute will return a value of 'Some New'. If none of the products has a 'Yes' as its New Flag value, the calculation will result in a value of 'None New'.

Solution Options

The following solutions are all valid for this approach.

Value Template 1

```
if(listlen(filter(subproducts(), 'exact("Yes",value("New Flag"))')) >0, "Some New","None New")
```

Value Template 2

```
{x := iterate(subproducts(), 'value("New Flag")="Yes")}  
if(listcontains(x,"1"), "Some New","None New")
```


Value Template 3

```
{x := iterate(subproducts(), 'value("New Flag")')}
if(listcontains(x,"Yes"), "Some New","None New")
```

Value Template 4

```
{x := iterate(subproducts(), 'value("New Flag")="Yes"')}
}
if(listcontains(x,"1"), "<NewLogo/>", "")
```

Notes

- Value Templates 1, 2, and 3 require some subsequent action, such as a transformation on the attribute value. For example, a transformation of Replace Whole Value could be established so that 'Some New' is transformed to an asset, as shown below; and the value of 'None New' would be transformed to nothing.

System Setup

- Attribute Groups
- Attribute Transformations
 - ObjectTransformations
 - Publishing Transformations
 - Append -A
 - Country Of Origin_Lowe
 - ExclusiveYN
 - NewLogo**
 - United States to USA
 - UOM Print
- Action Sets
- Contexts
- InDesign Queue
- Lists of Values / LOVs
- Asset Importer
- BusinessRuleMigration
- Change Packages
- Completeness Metrics
- D&B Integration
- Event Processors
- Experian Integration
- Gateway Endpoints
- GDSN
- Global Business Rules
- Inbound Integration Endpoints
- Integration Endpoints

NewLogo - Attribute Transformation

Attribute Transformation

Name	Value
ID	NewLogo
Name	NewLogo

Transformations

Replace ☒ Yes with

Add Transformation

Multivalue Separator

Applies To

☐ Name

☐ ID

ID	Name	Edited by
NewProduct	New Product	2016-09-01 11:27:56 by USER6

Add Attribute

☐ **All**

- ☒ Availability
- ☐ Back Cover Item (Y/N)
- ☐ Catalog Case Price

- Value Templates 1, 2, and 3 could have taken advantage of using a Stibo tag instead of creating a value of 'Some New' or 'None New'. This eliminates the need of a transformation, but does still require a setup for the

Stibo tag. In this example, the tag `<NewLogo/>` would be converted to the appropriate image call out, the same way as the transformation.

- Value Template 3 is very similar to Value Template 2.
- Value Template 4 is the same as Value Template 2, but uses a tag.

Override Scenario

A calculated attribute can be used to display information about a product override on the originating product. In the screenshot below, the product 'Diamond Cut End Mills (Table Guide) Series J' has been overridden several times. The sections below describe how to use calculated attributes to display on the originating product the name of the overriding products, display on the overriding product the name of the overridden product, and display on the overriding product the ID's of parent products.

Tree		Diamond Cut End Mills (Table Guide) Series J				
Product	Sub Products	References	Referenced By	Images & Documents	Co	
Used By Product Overrides						
ID	Name					
> TableGuide10	Table Guide 10 → Diamond Cut End Mills (Table Guide) Series J					
> DCEndMillsTableGuide	Table Guide 2 → Diamond Cut End Mills (Table Guide) Series J					
> DCTableGuide 3	Table Guide 3 → Diamond Cut End Mills (Table Guide) Series J					
> TableGuide4	Table Guide 4 → Diamond Cut End Mills (Table Guide) Series J					
> TableGuide5	Table Guide 5 → Diamond Cut End Mills (Table Guide) Series J					
> TableGuide6	Table Guide 6 → Diamond Cut End Mills (Table Guide) Series J					
> TableGuide7	Table Guide 7 → Diamond Cut End Mills (Table Guide) Series J					
> TableGuide8	Table Guide 8 → Diamond Cut End Mills (Table Guide) Series J					
> TableGuide9	Table Guide 9 → Diamond Cut End Mills (Table Guide) Series J					

Prerequisites

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 can list the names of the products that are overriding it. In this example, 'Diamond Cut End Mills (Table Guide) Series J' has multiple overrides.

Value Template

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.

Results

The screenshot displays the Stibo Systems interface. On the left, a tree view shows the hierarchy of products, with 'Diamond Cut End Mills (Table Guide) Series J' expanded to show its sub-products, 'Table Guide 2' through 'Table Guide 10'. On the right, the 'Diamond Cut End Mills (Table Guide) Series J' product details are shown. The 'Value' field is highlighted, displaying a list of overridden products: 'Table Guide 10', 'Table Guide 8', 'Table Guide 3', 'Table Guide 6', 'Table Guide 5', 'Table Guide 4', 'Table Guide 9', 'Table Guide 2', and 'Table Guide 7'.

Get Overridden Product Objects

Select a product override object to display the Overridden Product attribute where the value is the name and ID of the product being overridden. In this example, 'Table Guide 2' has overridden the product 'Diamond Cut End Mills (Table Guide) Series J' with ID 'DCEM-100.'

The screenshot displays the Stibo Systems interface. On the left, a tree view shows the hierarchy of products, with 'Diamond Cut End Mills (Table Guide) Series J' expanded to show its sub-products, 'Table Guide 2' through 'Table Guide 10'. On the right, the 'Table Guide 2' product details are shown. The 'Value' field is highlighted, displaying a list of overridden products: 'Diamond Cut End Mills (Table Guide) Series J (DCEM-100)'.

Value Template

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.

Results

Tree		Table Guide 2 → Diamond Cut End Mill	
- 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		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		Table Guide 2 → Diamond Cut End Mill	
- 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		Product	References
		Referenced By	Images & Documents
		Commercial	Tab
		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

Value Template

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()"), "/" "
```

Results

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

Commercial

Tables

Category Profile

Proof View

Packaging Hierarchy

Status

State Log

Tasks

Product

References

Referenced By

Image

Description

Name

>

>

Value

>

ID

8J5H1

>

Name

8J5H1

>

Object Type

Product

>

Approved

Never Been Approved

>

GetOverrideParents

TableGuide10/TableGuide8/DCTableGuide3/TableGuide6/TableGuide5/TableGuide4/TableGuide9/DCEnd MillsTableGuide/TableGuide7

Product Description Scenario

A calculated attribute can be used to automate the process of building dynamic product descriptions. Using existing attributes, and based on different product lines, in the example below we will generate a product description that is comprised of a basic 'stem' attribute, and concatenated with additional attributes.

Assumptions

1. The Primary Product Hierarchy is organized so that product attributes may be placed on different product nodes in that hierarchy that correlate to a 'product line.'
2. Up to three 'building' attributes are appended to a constant base (or 'stem') attribute to compose the complete product description.
3. The following attribute values exist for the product lines.

Product Line	Pencils	Writing Pens
Product Family Title ('stem')	BIC® Atlantis®	BIC® Grip Rollers
Point Size (PointSize)		Fine
Point Size MM (PointSizeMM)	0.5	
Ink Color (InkColor)		Blue
Package Qty (PkgQty)		10
Pencil Type (PencilType)	Mechanical	

The screenshot below displays the attribute values used to compose the calculated product description for the product lines in workbench:

Tree

- Pencils
 - ABC-2500-1015
 - ABC-2500-1016
 - ABC-2500-1017
- Writing Pens
 - ABC-2500-1012
 - ABC-2500-1013
 - ABC-2500-1014

Product | Sub Products | References | Re

ABC-2500-1017

Description

Item Brand Information

Name	>	Value
Qualifier 1	abc	PencilType
Qualifier 2	abc	PointSizeMM
Qualifier 3	abc	
Point Size	abc	
Ink Color	abc	
Package Qty	abc	
Pencil Type	abc	Mechanical
Point Size MM	abc	.05

Tree

- Pencils
 - ABC-2500-1015
 - ABC-2500-1016
 - ABC-2500-1017
- Writing Pens
 - ABC-2500-1012
 - ABC-2500-1013
 - ABC-2500-1014

Product | Sub Products | References | Ref

ABC-2500-1014

Description

Item Brand Information

Name	>	Value
Qualifier 1	abc	PointSize
Qualifier 2	abc	InkColor
Qualifier 3	abc	PkgQty
Point Size	abc	Fine
Ink Color	abc	Blue
Package Qty	abc	10
Pencil Type	abc	
Point Size MM	abc	

Results

The result of either solution 1 or solution 2 defined below are the same:

Tree

- Pencils
 - ABC-2500-1015
 - ABC-2500-1016
 - ABC-2500-1017
- Pilot Pens
 - Pilot Pens(2)
- Writing Pens
 - ABC-2500-1012
 - ABC-2500-1013
 - ABC-2500-1014

Product | Sub Products | References | Referenced By | In

ABC-2500-1017

Description

Item Brand Information

Name	>	Value
Qualifier 1	abc	PencilType
Qualifier 2	abc	PointSizeMM
Qualifier 3	abc	
Point Size	abc	
Ink Color	abc	
Package Qty	abc	
Pencil Type	abc	Mechanical
Point Size MM	abc	.05
Solution	fx	BIC® Atlantis®, Mechanical, .05
Product Family Title	abc	BIC® Atlantis®

Tree

- Pencils
 - ABC-2500-1015
 - ABC-2500-1016
 - ABC-2500-1017
- Pilot Pens
 - Pilot Pens(2)
- Writing Pens
 - ABC-2500-1012
 - ABC-2500-1013
 - ABC-2500-1014

Product | Sub Products | References | Referenced By | In

ABC-2500-1014 rev.0.1 - Product

Description

Item Brand Information

Name	>	Value
Qualifier 1	abc	PointSize
Qualifier 2	abc	InkColor
Qualifier 3	abc	PkgQty
Point Size	abc	Fine
Ink Color	abc	Blue
Package Qty	abc	10
Pencil Type	abc	
Point Size MM	abc	
Solution	fx	BIC® Grip Rollers, Fine, Blue, 10
Product Family Title	abc	BIC® Grip Rollers

Solution Option 1

Approach

1. Create three 'Qualifier' attributes that will reference and be made valid on the product nodes in the Primary Product Hierarchy. These attributes will hold the IDs of the attributes to be used to compose the final Product Description. In this case those attributes will have the ID of Q1, Q2, and Q3 (with names of Qualifier 1, 2, and 3).

- Place the qualifier attributes on the appropriate product line object type(s) and populate the values with the IDs of the attributes to use for the composed product description. For example, the pencils product line will have Q1 = PencilType, Q2 = PointSizeMM, Q3 = {empty}; and the writing pens product line will have Q1 = PointSize, Q2 = InkColor, Q3 = PkgQty

Value Template

The calculated attribute template includes a concatenation / if structure like this:

```
concatenate (value('ProductFamilyTitle')
,if(exact(value('Q1'),''),' ',concatenate(' ',value(value('Q1'))))
,if(exact(value('Q2'),''),' ',concatenate(' ',value(value('Q2'))))
,if(exact(value('Q3'),''),' ',concatenate(' ',value(value('Q3')))))
```

Notes

- The syntax **value(value('Q1'))** is used to get the value of the attribute ID pointed to by the attribute Q1 if that value is not blank. The same is repeated for Q2 and Q3.
- The concatenate function takes into account skipping over empty Qualifier attribute values as well as empty values in the attributes to be used to build the product description.
- This approach is not limited to only three attributes. As many Qualifier attributes can be used as are required by inserting an copy of the 'Q2' line before the 'Q3' line. Note that the 'Q3' row in the value template includes the final closing parenthesis and should not be duplicated.
- This approach requires only one calculated attribute to satisfy the needs of every product line. A more labor intensive (not recommended) approach would be to create a calculated attribute using fixed attributes for each product line.
- For the same result while simplifying the calculation and making it more readable, use variables to hold the values of each attribute, as shown below:

```
{GetQ1 := if(exact(value('Q1'),''),' ',concatenate(' ',value(value('Q1')))),
GetQ2 := if(exact(value('Q2'),''),' ',concatenate(' ',value(value('Q2')))),
GetQ3 := if(exact(value('Q3'),''),' ',concatenate(' ',value(value('Q3')))),
stem := value('ProductFamilyTitle')
}
concatenate (stem,GetQ1,GetQ2,GetQ3)
```

- When using auto-generated attribute IDs, if you prefer to see the name of the attribute in the Qualifiers attribute (for readability purposes), allow the qualifier values to include either name or id first, separating them with a delimiter character such as a colon (:). For example, if the Point Size attribute had an ID of 12734, and Ink Color an ID of 28276, and Package Quantity an ID of 47338, the qualifier values could allow for entries of 'Point Size:112734', 'Ink Color:28276', and 'Package Quantity:47338'. Then need to modify the value template to parse out the attribute ID from each qualifier entry, as demonstrated for Q1 below:

```
{GetID1 := trim(mid(value('Q1'),find(":",value('Q1'))+1,100)),
GetQ1 := if(exact(GetID1,''),' ',concatenate(' ',value(GetID1))),
....
}
concatenate (stem,GetQ1,GetQ2,GetQ3)
```

Solution Option 2

Approach

Instead of having separate qualifier attributes for each element in the product description, a single attribute could be used with all the required Qualifier IDs separated by a pre-designated delimiter character such as a comma.

1. Create one qualifier attribute with the ID of Qualifiers (name is also Qualifiers).
2. Reference the Qualifiers attribute on the appropriate product line object type(s), make it valid, and populate the values with the IDs of the attributes to use for the composed product description, separated by a comma delimiter. For example, PointSize,InkColor,PkgQty and PointSizemm,PkgQty

Value Template

This formula requires that the spaces before and after the delimiter must be accommodated.

```
{qattrs := iterate(ESCAPESPLIT(',', '/', value('Qualifiers')), 'trim(item)'),
  vals := filter(iterate(qattrs, 'value(item)'), 'not(exact(item, ""))')}
concatenate(value('ProductFamilyTitle'), ' ', list(vals, ', '))
```

Notes

- This approach first builds a list (label is 'qattrs') from the entries found in the attribute 'Qualifiers', separated by a comma.
- Then another list (label is 'vals') is formed by iterating through the qattrs list.
- Finally the list 'vals' is converted to a text string (with a comma / space as the separator between values) and is prepended to the ProductFamilyTitle attribute value.

Reference Scenario

A calculated attribute can be helpful when an object has references to multiple products. The scenarios described below are examples of how calculated attributes could be used.

- Get List of References
- Get Value from Referenced Product
- Get Metadata Value Located on the Link from Product-to-Product
- Get Referenced By Using Reference Owner
- Get Reference Type ID
- Get Reference Source or Target Product Information on the Link

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.

Tree	Classroom Type Projectors (Class A) rev.0.6 - Ref					
	Product	Sub Products	References	Referenced By	Images & Documents	Commercial
Hand Tools Office Equipment Projectors Classroom Type Projectors ABC-1001-100 ABC-1001-101 ABC-1001-102	Product Link Metadata					
	Reference Type	Target	ID	Name	Thumbnail	
	Alternative	Office Projectors	Office Projectors	Office Projectors		
		Portable Projectors	Portable Projectors	Portable Projectors		

Value Template

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)
```

Notes

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()')
```

Results

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

Tree

- Hand Tools
- Office Equipment
 - Projectors
 - Classroom Type Projectors**
 - ABC-1001-100
 - ABC-1001-101
 - ABC-1001-102
 - ABC-1001-103
 - ABC-1001-104
 - ABC-1001-105
 - Home Theater
 - Office Projectors

Classroom Type Pro

Product

Sub Products

References

Referenced By

Images & Documents

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
 - Home Theater
 - Office Projectors
 - Portable Projectors
 - Utility Projectors

Products

References

Referenced 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

Value Template

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))
```

Results

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	<	Class
Office Equipment	Product	Sub Products
Projectors	References	Referenced By
Classroom Type Projectors	Name	Value
Home Theater	Get Value From Ref Product	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:

Tree	<	Classroom Type Projectors (Class A) rev.0.6 - References
Office Equipment	Product	Sub Products
Projectors	References	Referenced By
Classroom Type Projectors	Images & Documents	Commercial
Home Theater	Tables	Category Profile
Office Projectors	Proof View	
Portable Projectors		
Utility Projectors		

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

Value Template

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)
```

Results

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.

Tree

- Hand Tools
- Office Equipment
 - Projectors
 - Classroom Type Projectors
 - Home Theater

Product

Sub Products

References

Referenced By

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.

Tree

Office Equipment

Projectors

Classroom Type Projectors

Home Theater

Office Projectors

Portable Projectors

Utility Projectors

3D Projectors

<

Home Theater rev.0.2 - Referenced By

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Tables

Product Link Metadata

Reference Type	Source	ID	Name
> Alternative +	3D Projectors	3D Projectors	3D Projectors
	Portable Projectors	Portable Projectors	Portable Projectors
	Utility Projectors	Utility Projectors	Utility Projectors

Tree

Office Equipment

Projectors

Classroom Type Projectors

Home Theater

Office Projectors

Portable Projectors

Utility Projectors

3D Projectors

Products

Products

References

Referenced By

View: Revisions view

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

Value Template

The following calculated attribute formula could be written:

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

```
concatenate(fin)
```

Notes

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.

Results

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

Tree		Home Theater rev.1					
		Product	Sub Products	References	Referenced By	Images & Documents	Commercial
Office Equipment							
Projectors							
Classroom Type Projectors							
Home Theater							
Office Projectors							
		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 - Home Theater - Office Projectors - Portable Projectors - Utility Projectors 3D Projectors - Pens, Pencils, Rollerballs, and S - Chairs		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

Value Template

The following calculated attribute formula could be written:

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

Results

The results display each of the reference type IDs.

Tree		< Classroom Type Projectors				
Office Equipment - Projectors - Classroom Type Projectors - Home Theater - Office Projectors - Portable Projectors - Utility Projectors 3D Projectors - Pens, Pencils, Rollerballs, and S - Chairs		Product	Sub Products	References	Referenced By	Images & Documents
		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.

Value Template

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)
```

Notes

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.

Results

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

Tree

Office Equipment

Projectors

Classroom Type Projectors

Home Theater

Office Projectors

Portable Projectors

Utility Projectors

3D Projectors

Pens, Pencils, Rollerballs, and S

Chairs

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

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Tables

Category Profil

Description

Name	>	>	Value
ID			Classroom Projectors
Name			Classroom Type Projectors (Class A)
Object Type			Product Family
GetProdRefNamesAndRefTypes			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

Get Reference Source or Target Product 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.

Value Template

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

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

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()')

Results

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

Tree

Safety

Hardware

Tools

Task Lighting

Flashlights

Flashlights Items

111204 Child of Overri

Flashlight Case

Flashlight Clip

114859

114861

20862

20883

LED Pocket Flashlight

LIGHTS-10

LIGHTS-10 rev.0.6 - References

0% complete

ProductSub ProductsReferencesReferenced ByImages & Documents

Owners Manual

Image References

Item References, Classification

Item References, Product

Packaging Hierarchy References

Sales Item References, Classification

Sales Item References, Product

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

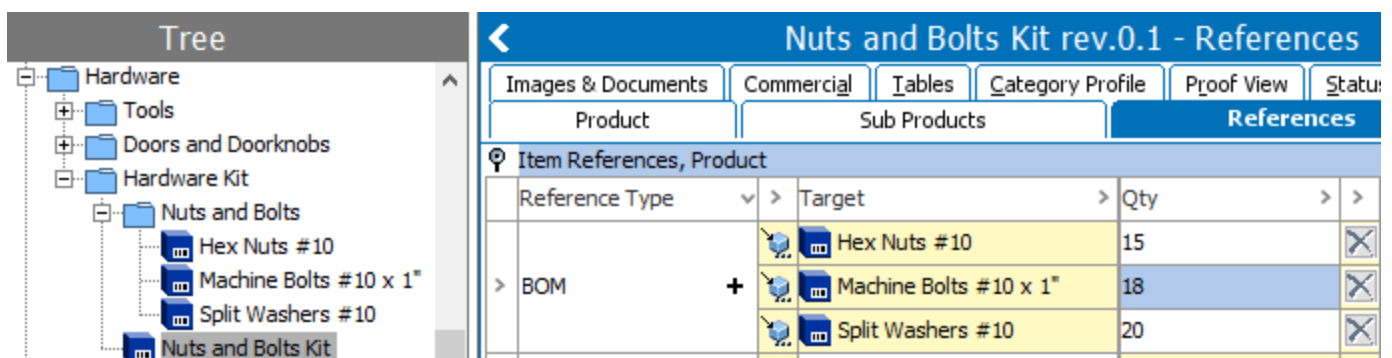
Total Cost of Materials Scenario

A calculated attribute can be used to find the total cost of a Bill of Materials (BOM), which is an 'assembled' product, that is itself a collection of individual products or 'part products.'

The most popular approach is to link the source product (i.e., the assembled product) to the various part products that go to make up that assembly (target product), using a specific reference type, for example, 'BOM.' Then, the cost of each of the part products is usually a standard product attribute, for example, 'Cost Price.' Finally, the quantity required for each of the part products in the assembly is a description attribute (metadata) on the link between the source product (the assembly) and each target product (the part products).

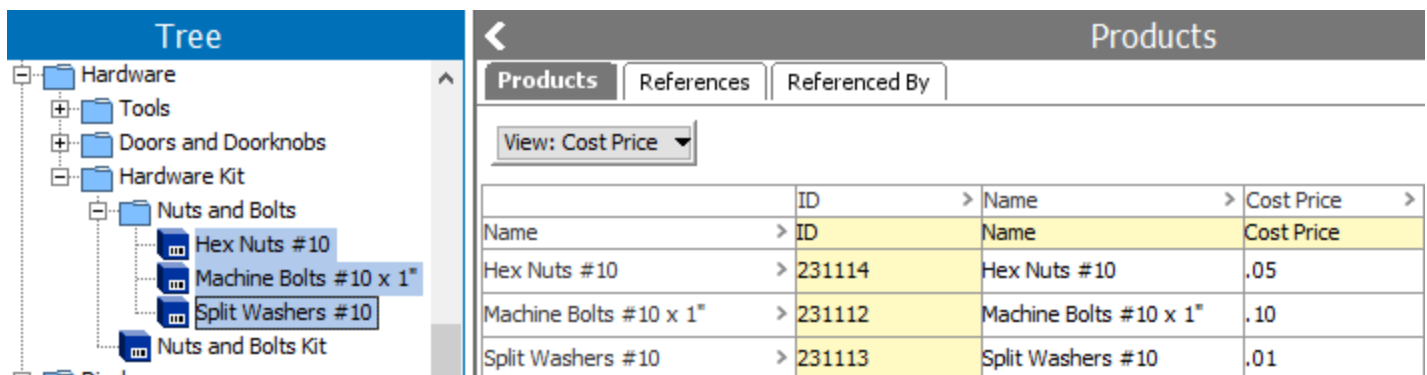
Assumptions

1. The reference type with ID = BOM exists and the assembled product is linked to the individual part products using this reference type.
2. A description attribute (metadata) with ID = Qty exists on the BOM reference type and the values are populated for how many of those part products are required in the assembled product.



Reference Type	Target	Qty	Status
BOM	Hex Nuts #10	15	
BOM	Machine Bolts #10 x 1"	18	
BOM	Split Washers #10	20	

3. An attribute with ID = CostPrice exists on the part products and the values are entered for the cost of each part product.



ID	Name	Cost Price
231114	Hex Nuts #10	.05
231112	Machine Bolts #10 x 1"	.10
231113	Split Washers #10	.01

Solution

Approach

In the calculation, three (3) lists are generated.

- The **CostPriceVal** list will hold the CostPrice of each of the part products referenced by the assembled products. By definition, these two lists will have the same number of entries and have values that are in the same order. That is, entry #1 in List #1 will have the corresponding cost in entry #1 of List #2.
- The **QtyVal** list will hold the Qty attribute values for each of the references from the assembled part to the part products.
- The **TotalCost** list will hold the list of total cost for each reference by multiplying entry #1 in List #1 with the value in entry #1 of List #2, and so on. The last part of the calculation will then summarize the entries of this third list, and thus calculate the total cost for the assembled product.

Value Template

```
{CostPriceVal := iterate(iterate(references("product", "BOM"), 'referencetarget()'), 'value("CostPrice")'),
QtyVal := iterate(references("product", ""), 'value("Qty")'),
TotalCost := iterate(CostPriceVal, 'item*listitem(QtyVal, index)'),
total := '0',
dummy := iterate(TotalCost, 'total:=item + total')
}
concatenate('Sum of BOM = ', total)
```

Notes

The summation of the entries in TotalCost is a little unusual in that the variable 'dummy' is not required at all, but the variable 'total' is what is needed. Although the contents of the variable 'dummy' are never used, in calculating that value, it is used to calculate the variable 'total' which is the required piece.

Results

The value template multiplies the CostPrice value with the Qty value for each item in the kit as follows:

- Hex Nuts: .05 x 15 = .75
- Machine Bolts: .10 x 18 = 1.80
- Split Washers: .01 x 20 = .20

Then sums the total for each part of the kit: $.75 + 1.80 + .20 = 2.75$

Tree

Hardware

Tools

Doors and Doorknobs

Hardware Kit

Nuts and Bolts

Hex Nuts #10

Machine Bolts #10 x 1"

Split Washers #10

Nuts and Bolts Kit

Nuts and Bolts Kit

Images & DocumentsCommercialTablesCategory I

ProductSub Products

Description

Item Purchasing Information

Name	>	>	Value
Solution		fx	Sum of BOM = 2.75

Item Brand Information

Item Manufacturer Information

Units Scenario

A calculated attribute can show multiple unit values using a single unit of measure. In this example, a product has several attributes with units, but the values entered are not using the same unit, the values of attributes height, length, and width have valid unit types of MM and CM.

Tree

Primary Product Hierarchy

Products

Apparel

Footwear

Displays

Furniture

Automotive

Building Products

Lumber

Pressure Treated Lumber

Pressure Treated Lum

20820

63220X0

<

Product

Sub Products

References

Referenced By

Description

Calculated Attributes

Name	>	>	Value
> Dimensions with CM Units		fx	25.5 x 105 x 65

Dimensions

Name	>	>	Value
> Height	123		255 mm
> Length	123		65 cm
> Width	123		105 cm

Prerequisites

For more information on units, see the **Units** topic in the **Getting Started / User Guide** documentation.

Approach

The units will need to be configured for conversion, and then the required display unit will need to be specified for the results.

Configure Units for Conversion

Using an appropriate conversion formula to associate multiple units allows the calculated attribute to make the conversions as needed. For example, meters can be converted to millimeters and centimeter as shown below.

System Setup

Units

Length

- µm
- Å
- cm
- ft
- in
- km
- m
- mile
- mm

mm - Units

UnitsReferencesLog

Description

Name	>	>	Value	>	s	Log
ID			unece.unit.MMT			
Name			mm			
Last edited			2017-01-12 08:24:24 by USERA			
Path			Length/mm			
Base Unit			Value(m) = .001 * Value(mm)			
Unit Description		abc	millimetre			

mm

Base Unit

Unit Description

abc

millimetre

cm - Units

Log

Description

Name	>	>	Value	>
ID			unece.unit.CMT	
Name			cm	
Last edited			2016-07-26 15:26:26 by USER	
Path			Length/cm	
Base Unit			Value(m) = 0.01 * Value(cm)	
Unit Description		abc	centimetre	

cm

Base Unit

Unit Description

abc

centimetre

Concatenate and Specify Required Unit

Different units were selected while adding values for the height, length, and width attributes.

Tree

Primary Product Hierarchy

- Products
 - Apparel
 - Footwear
 - Building Products
 - Sealers/Fillers/Adhesives
 - Lumber
 - Pressure Treated Lumber
 - Pressure Treated Lum

20820

Product

Sub ProductsReferencesRefer

Description

Calculated Attributes

Dimensions

Name	>	>	Value
Height	123		255 mm
Length	123		65 cm
Width	123		105 cm

The calculated attribute value template can specify the required display unit for the result. In the following value template, notice that the centimeter unit is applied to each attribute value. It also concatenates each of the values and separates them with 'x' (static text).

Note: To verify the value template formula is correct, use Evaluation Node to select an object with values in the attributes and click the Evaluate button. The result is displayed in the bottom text box.

Dimensions with CM Units - Attribute

Attribute	References	Attribute Transformation	Validity	Profile	Log	State Log	Tasks
Description							
Name	>	>	Value				
ID			DimWCMUnits				
Name			Dimensions with CM Units				
Last edited by			2017-06-23 14:21:04 by US				
Full Text Indexable			No				
Externally Maintained			No				
Hierarchical Filtering			None				
Calculated			Yes				
Type			Specification				
Dimension Dependencies							
Value template			concatenate(prodval("Height","unece.unit.CMT"), " x ", prodval("Width","unece.unit.CMT"), " x ", prodval("Length","unece.unit.CMT"))				
Mandatory			No				

Auto Indent | Insert Template | Insert Attribute ID | Highlighting ▾

concatenate (prodval("Height","unece.unit.CMT"), " x " ,
prodval ("Width","unece.unit.CMT"), " x " ,
prodval ("Length","unece.unit.CMT"))

Evaluation Node 20820 (20820) ... Evaluate

25.5 x 105 x 65

OK Cancel

Value Template

The following text was added to the value template:

```
concatenate (prodval ("Height","unece.unit.CMT")," x " ,
prodval ("Width","unece.unit.CMT")," x " ,
prodval ("Length","unece.unit.CMT"))
```

Note that the value entered as 255 mm was converted to 25.5 cm, using the Base Unit conversion formula.

STEP Functions

STEP functions allow you to access a variety of STEP data. Combining function results with static text and comments makes it possible to display a generated result as a value.

STEP functions are commonly used in:

- Calculated Attributes - where the value is not stored in the database, but is calculated on the fly, based on other data in STEP. For more information, see the **Creating a Calculated Attribute** section of **Calculated Attributes** in the **System Setup / Super User Guide** documentation.
- Bulk Updates - where mass editing functionality is available, for example, to dynamically calculate values or reference targets. For more information, see the **Set Value** section of **Attribute Values Operations for Bulk Updates** in the **Bulk Updates** documentation.
- Business Conditions - where data is tested against a condition, for example, to determine the path for a workflow. For more information, see the **Business Condition: Function** section of **Business Conditions** in the **Business Rules** documentation.
- Matching and Linking - where Match Codes uses Match Code Formulas. For more information, see **Configuring Match Codes** in the **Matching, Linking, and Merging** documentation.
- Key Definitions - where unique IDs (keys) can be defined by a key formula. For more information, see the **Creating and Deleting Keys** section of the **Unique Keys** documentation.

For information on the STEP functions that work in a similar or identical manner as functions in Excel, see the **Comparing STEP and Excel Functions** topic.

This documentation categorizes functions based on the data accessed. A description of each of the functions is included in the following sections:

- Attribute Functions
- Commercial Functions
- Conditional Functions
- Date Functions
- List Functions
- Number Functions
- Other Functions
- Text Functions
- Variable Functions

For information on adding comments to a calculated attribute, see the **Comments** topic

For information about overriding and restoring calculated attribute values, see the **Overriding and Restoring Calculated Attribute Values** topic.

Comments

Working with calculated attributes often results in using a combination of functions, and can include many lines of text. Comments allow the author to note the intention of the elements of the calculation and future readers to understand the reason for the element. Comments are ignored by the Function Editor and are only useful for a user reviewing the value template.

Comments can be added for the current line or can encompass multiple lines as defined the table below.

Description	Syntax	Example
Comment spans multiple lines	<code>/* */</code>	<code>prodval('NUMBERlid') /* This text is a comment and commented text continues on each line until the concluding comment indicator is reached.*/</code>
Comment for the current line	<code>//</code>	<code>prodval('TEXTlid') // This text is a comment</code>

Use Cases

For a scenario using comments, see **Including Parent Info on Child Scenario**.

For a scenario using comments, see the **Get Reference Source or Target Product Information on the Link** section of the **Reference Scenario**.

Comparing STEP and Excel Functions

Below is a list of functions available for use within STEP calculated attributes. These functions are either identical or similar to the matched Excel function. Additional data and tutorials are available for these functions in Excel.

STEP Function	Excel Function	Comparison; Additional STEP Documentation
CONCATENATE	CONCAT	Identical; see Text Functions
IF	IF	Similar; see Conditional Functions
INT	INT	Similar; see Number Functions
LEFT	LEFT	Identical; see Text Functions
LOWER	LOWER	Identical; see Text Functions
MID	MID	Identical; see Text Functions
PROPER	PROPER	Identical; see Text Functions
REPLACE	REPLACE	Identical; see Text Functions
RIGHT	RIGHT	Identical; see Text Functions
ROUND	ROUND	Identical; see Number Functions
SUBSTITUTE	SUBSTITUTE	Identical; see Text Functions
TRIM	TRIM	Similar; see Text Functions
TRUNC	TRUNC	Similar; see Number Functions
UPPER	UPPER	Identical; see Text Functions

Attribute Functions

Calculated Attribute expressions can be written using the following available functions. **Use Cases** follow the available functions table.

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

Function	Parameters	Description
ATTRNAME	('attribute-id')	Returns the name of the attribute (attribute-id).
GETEMBEDDEDPREFIX	(PRODVAL ('attribute-id'))	Returns 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). For more information about embedded numbers, see the Validation Rules in this guide.
GETEMBEDDEDSUFFIX	(PRODVAL ('attribute-id'))	Returns 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 in this guide.
GETEMBEDDEDVALUE	(PRODVAL ('attribute-id'))	Returns 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 in this guide.
PRODUNIT or UNITOFVALUE	('attribute-id')	Returns 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')	Returns the ID of the unit of the value of the supplied attribute (attribute-id) for the current object.

Function	Parameters	Description
		The expression cannot evaluate the Unit from other Calculated Attributes.
PRODVAL or VALUE	('attribute-id' [, 'unit-id'])	Returns 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. 'unit-id' is optional, and to convert units, the relevant units must have a Base Unit conversion defined in the System Setup > Units node. Returns 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. When unit is specified, it is automatically separated from the value with a single space.
PRODVALSIMPLE or SIMPLEVALUE	('attribute-id' [, 'unit-id'])	Returns 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. 'unit-id' is optional, and to convert units, the relevant units must have a Base Unit conversion defined in the System Setup > Units node. Returns 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. When unit is specified, it is automatically separated from the value with a single space.
UNITCONV	(value, unit-id, to-unit-id)	Returns number of 'value' in 'unit-id' converted into 'to-unit-id'.
VALUELOVID	(attribute-id)	Returns 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.

Use Cases

For examples of these functions, see **Attribute Examples**.

For a scenario using the VALUE function, see **Product Description Scenario**.

For a scenario using the PRODVAL with UNITS function, see **Units Scenario**.

Commercial Functions

Calculated Attribute expressions can be written using the following available functions. **Use Cases** follow the available functions table.

In addition to using commercial value functions for publications, they can also be used without actually publishing anything. For example, you can use price sheets stored in an eCatalog hierarchy to maintain price information.

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
TERMQTYMAX	('termslisttype-id', 'break-no'[, effective date])	Return the maximum quantity of the supplied 'termslist type id' for the current publication version or eCatalog 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 or eCatalog 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 or eCatalog 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 or eCatalog in the supplied break-no for the given effective date. [effective date] is optional.
TERMVAL	('termslisttype-id', 'break-no'[, effective date])	Returns the term value of the supplied 'termslist type id', ' for the current publication version or eCatalog hierarchy in the supplied 'break no' for the given effective date. [effective date] is optional.

Function	Parameters	Description
TERMVALID	('termslist-id', 'break-no', effectivedate])	Return the term value of the supplied 'termslist id' for the current publication version or eCatalog in the supplied break-no for the given effective date. [effective date] is optional.

Use Cases

For examples of these functions, see **Commercial Examples**.

For a scenario using commercial value functions, see **Commercial Data Scenario**.

Conditional Functions

Calculated Attribute expressions can be written using the following available functions. **Use Cases** follow the available functions table.

Conditional Function	Parameters	Description
AND	(condition1, condition2, ...)	Returns 1 (=TRUE), if all the supplied conditions are true; otherwise returns 0 (=FALSE).
IF	(condition, 'value_if_true', 'value_if_false')	If 'condition' is true, then 'value_if_TRUE' is returned, otherwise 'value_if_FALSE' is returned. The OR, AND, and NOT functions can be used with the IF function to check multiple conditions at a time. For more information, these are explained in the Other Functions documentation. The EXACT function can be used with the IF function to verify if two strings match exactly. For more information, see the Text Functions documentation.
NOT	(condition)	Returns 1 (=TRUE) if condition is false; otherwise returns 0 (=FALSE).
OR	(condition1, condition2, ...)	Returns 1 (=TRUE) if one of the supplied conditions is true; otherwise returns 0 (=FALSE).

Use Cases

For examples of these functions, see **Conditional Examples**.

For a scenario using the IF function, see **Comparing Localized Values Scenario**.

For a scenario using the IF function, see **Product Description Scenario**.

For a scenario using the IF function, see **New Product Indicator Scenario**.

For a scenario using the IF function, see **Identifying Child Objects Scenario**.

Date Functions

Calculated Attribute expressions can be written using the following available functions. **Use Cases** follow the available functions table.

Note: Date functions require that 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.

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.

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.

Use Cases

For examples of these functions, see **Date Examples**.

List Functions

Calculated Attribute expressions can be written using the following available functions. **Use Cases** follow the available functions table.

Function	Parameters	Description
FILTER	(list, expression)	Returns a list with the elements of 'list' where 'expression' evaluates to true. The 'index', 'item', and 'size' variables are also available. For more information, see the Special Variable Functions section of the Variable Functions documentation.
INDEX		The 'index' special variable is available for use within all ITERATE and FILTER functions. For more information, see the Special Variable Functions section of the Variable Functions documentation.
ITEM		The 'item' special variable is available for use within all ITERATE and FILTER functions. For more information, see the Special Variable Functions section of the Variable Functions documentation.
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. The 'index', 'item', and 'size' variables are also available. For more information, see the Special Variable Functions section of the Variable Functions documentation.
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.
LIST2MULTIVALUE	(list)	Returns multi-valued text, where the values equal the elements in the 'list'.

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.
LISTCONTAINS	(list, text)	Returns 1 (=TRUE) if 'list' contains the element 'text'; otherwise returns 0 (=FALSE).
LISTITEM	(list, indexno)	Returns the item identified by 'indexno' in 'list'. The index number of the first element is 1.
LISTLEN	(list)	Returns the number of elements in 'list'.
MULTIVALUE2LIST	(text)	Returns a list where the elements equal the values in the multi-valued 'text'.
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.
REVERSE	(list)	Returns a list with the elements of 'list' in reverse order.
SIZE		The 'size' special variable is available for use within all ITERATE and FILTER functions. For more information, see the Special Variable Functions section of the Variable Functions documentation. Note: The LISTLEN function provides the same results and is easier to use.
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.

Use Cases

For examples of these functions, see **List Examples**.

For a scenario using the LISTLEN and LISTITEM functions, see **Comparing Localized Values Scenario**.

For a scenario using the LISTLEN function, see **Identifying Child Objects Scenario**.

For a scenario using the FILTER, ITERATE, and LISTITEM functions, see **Including Parent Info on Child Scenario**.

For a scenario using the LISTLEN function, see **New Product Indicator Scenario**.

For a scenario using the FILTER and ITERATE functions, see **Product Description Scenario**.

For a scenario using the ITERATE function, see **Reference Scenario**.

For a scenario using the ITERATE and LISTITEM functions, see **Total Cost of Materials Scenario**.

Number Functions

Calculated Attribute expressions can be written using the following available functions. **Use Cases** follow the available functions table.

Number Function	Parameters	Description
Binary logical functions	<, >, <=, >=, =, !=	Returns 1 (=TRUE) or 0 (=FALSE). Note: These functions do NOT work on text strings since they attempt to convert to a number.
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.
INT	(number)	Returns the integer part of 'number' by rounding down to the nearest integer. Negative numbers are rounded down, away from 0. This function works exactly the same as the TRUNC function for positive numbers.
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
ROUND	(number, precision)	Returns 'number', rounded to a specified number of digits, controlling the number of decimal places. • 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.

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. See also the INT function in this topic.

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.

Use Cases

For examples of these functions, see **Number Examples**.

Other Functions

In the sections below, other calculated attribute functions are grouped by purpose:

- **Classification Functions** can be used to access classification objects.
- **Completeness Functions** can be used to access the completeness score of an object.
- **Data Container Functions** can be used to access data container objects.
- **Hierarchy Functions** can be used to access STEP hierarchy objects.
- **Lookup Table Functions** can be used to access lookup table data.
- **Object Functions** can be used to access object information.
- **Override Functions** can be used to access product override data.
- **Packaging Hierarchy Functions** can be used to access packaging hierarchy information.
- **Reference Functions** can be used to access object reference data.
- **Revision Functions** can be used to access revision information.
- **Workflow Functions** can be used to access workflow information.

For examples and scenarios using these functions, see the list of **Use Cases** at the end of each section or see **Other Examples**.

Classification Functions

These functions can be used to access classification objects.

Classification 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.
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, all are returned if not specified.

Classification Function	Parameters	Description
CPLCLASSIFICATION	()	Returns the classification object from a classification-product link.
CPLPRODUCT	()	Returns the product object from a classification-product link.

Use Cases

For a scenario using the CLASSIFICATIONS function, see **Classification Scenario**.

Completeness Functions

These functions can be used to access the completeness score of an object.

For more information, see **Setting Up Completeness Metrics** in the **Data Quality** documentation.

Completeness Function	Parameters	Description
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.

Data Container Functions

These functions can be used to access data container objects.

Data Container Function	Parameters	Description
DATACONTAINERCOMPOSITE	()	Returns composite object of current data container object.
DATACONTAINERS	([data-container-])	Returns all data container objects of current composite object.

Data Container Function	Parameters	Description
	type])	'data-container-type' is optional and returns all data containers of data container type with ID
DATACONTAINERTYPEID	()	Returns ID of data container type of current data container

Hierarchy Functions

These functions can be used to access STEP hierarchy objects.

Hierarchy Function	Parameters	Description
CHILDREN	()	Returns true children (not links) of an asset, a classification, an entity, a product, or a product-override. Similar to SUBPRODUCTS, but this function returns more than products. Note: This function must be used in combination with a LIST or ITERATE function. For more information, see the List Functions documentation.
ISBELOW	(url)	Returns TRUE (=1) if the object pointed to by 'url' is in the parent path of the current object.
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.
PARENTID	()	Returns the parent ID if node is a tree node.
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. For more information, see the List Functions documentation.

Use Cases

For a scenario using the PATH function, see **Comparing Localized Values Scenario**.

For a scenario using the PATH function, see **Including Parent Info on Child Scenario**.

For a scenario using the SUBPRODUCTS function, see **Identifying Child Objects Scenario**.

For a scenario using the SUBPRODUCTS function, see **New Product Indicator Scenario**.

Lookup Table Functions

These functions can be used to access lookup table data.

For more information, see the **Creating a Transformation Lookup Table** topic or the **Transformation Lookup Tables Example** topic of the **Data Quality** documentation.

Lookup Table Function	Parameters	Description
REPLACEVALUEBYLOOKUP	(asset-id, value)	Returns the 'value' substituted by using the transformation lookup table referred to in 'asset-id'.
REPLACEWORDBYLOOKUP	(asset-id, words)	Returns what each word in 'words' is substituted by, using the transformation lookup table referred to in 'asset-id'.

Object Functions

These functions can be used to access object information.

Object Function	Parameters	Description
STEPID	()	Returns the ID of the current object.
STEPNAME	(['url'])	Returns the name of the current object. 'url' is optional, and returns the name of the object of the URL.
STEPURL	(['object-type', 'object-id'])	Returns the STEP URL of the current object. 'object-type' and 'object-id' are optional, and return the STEP URL of the object with ID 'object-id' and of type 'object-type'. The object type could be product, classification, entity, asset, attribute, etc.
STEPURL2OBJID	('url')	Returns the internal object representation of the object of the URL Note: This function must be used in combination with a LIST or ITERATE function. For more information, see the List Functions documentation.

Object Function	Parameters	Description
STEPOBJECTTYPE	(['url'])	Returns the STEP name of the object type of the current object. 'url' is optional, and returns the STEP name of the object type of the object of the URL.

Override Functions

These functions can be used to access product override data.

Override Function	Parameters	Description
OVERRIDDENBY	()	Returns the product overrides for the current product. Returns an empty list if the object is not overridden.
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.
PRODUCTOVERRIDEPARENTS	()	Returns the product overrides having this product as child. Returns an empty list if the object is not linked into any product overrides.

Use Cases

For a scenario example using the OVERRIDES function, see **Override Scenario**.

Packaging Hierarchy Functions

These functions can be used to access packaging hierarchy information.

Packaging Hierarchy Function	Parameters	Description
PACKAGINGHIERARCHYCHILDLINKS	()	Returns the packaging hierarchy child links of the current product.
PACKAGINGHIERARCHYPARENTLINKS	()	Returns the packaging hierarchy parent links of the current product.
PHLCHILD	()	Returns the child product of the current packaging hierarchy link.
PHLPARENT	()	Returns the parent product of the current packaging hierarchy link.
PHLQUANTITY	()	Returns the quantity of the current packaging hierarchy link.

Reference Functions

These functions can be used to access object reference data.

Reference Function	Parameters	Description
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.
REFERENCEOWNER	()	Returns the owner of the reference.
REFERENCETO	([object-type [, reference-type]])	Returns the references pointing to this object from objects of type 'object-type' and with specified reference type. Object-type could be product, classification, entity, or asset. [object-type] and [reference-type] are optional, all are returned if not specified or blank.
REFERENCETARGET	()	Returns the target (destination) object of reference.
REFERENCETYPEID	()	Returns the ID of the reference type.

Use Cases

For a scenario using the REFERENCES function, see **Reference Scenario**.

For a scenario using the REFERENCES and REFERENCETARGET functions, see **Total Cost of Materials Scenario**.

Revision Functions

These functions can be used to access revision information.

Revision Function	Parameters	Description
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".
REVISIONCOMMENT	()	Returns the comment of the current revision (or the most recent revision of current object).
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".
REVISIONEDITEDBY	()	Returns the name of the user responsible for the change in the current revision (or most recent revision of current object).
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".
REVISIONISMAJOR	()	Returns the Boolean 1 (=TRUE) or 0 (=FALSE) to indicate if the current revision (or most recent revision of current object) is a major revision or not.
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.
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.

Revision Function	Parameters	Description
		Note: This function must be used in combination with a LIST or ITERATE function. For more information, see the List Functions documentation.

Workflow Functions

These functions can be used to access workflow information.

Workflow Function	Parameters	Description
CURRENTWORKFLOWSTATES	(workflow-id)	Returns the list of IDs for the STEP workflow states that include the current object for the supplied 'workflow-id'.
WORKFLOWIDS	()	Returns the list of workflow IDs for the STEP workflows that include the object.

Text Functions

In the sections below, the calculated attribute text functions are grouped by purpose:

- **Escape Functions** can be used to allow for special characters during text manipulation.
- **Identify Functions** can be used to display the defined data.
- **Modify Functions** can be used to modify defined data.
- **Regular Expression Functions** can be used to search and modify text based on a defined template.
- **Representative Text Functions** can be used to produce an alternative value for the defined text.
- **Test Functions** can be used to perform a test, and return a result.

For examples and scenarios using these functions, see the list of **Use Cases** at the end of each section or see **Text Examples**.

Escape Functions

These functions can be used to allow for special characters during text manipulation.

Escape Function	Parameters	Description
ESCAPEMERGE	(split-character, escape-character, text1, ...)	Merges text1...textN into a single string. The 'split-character' is used as a separator between the elements. If the 'split-character' is found in the text elements, then the 'escape-character' precedes it. Opposite function of ESACPESPLIT.
ESCAPESPLIT	(split-character, escape-character, text)	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.

Identify Functions

These functions can be used to display the defined data.

Identify Function	Parameters	Description
LEFT	(text [, len]):	Returns the first 'len' number of characters of the text. [len] is optional, the default is 1.
LEN	(text)	Returns number of characters in 'text'.
MID	(text, pos, len)	Returns the substring of 'text' starting at character position 'pos' with length 'len'.
RIGHT	(text[, len])	Returns the last 'len' number of characters of the text. [len] is optional, default is 1.

Modify Functions

These functions can be used to modify defined data.

Modify Function	Parameters	Description
CONCATENATE	(text1, text2, ...)	Returns the concatenated text of all the supplied texts in the parameter string. There is no limit to the number of strings that can be concatenated. Not typically used with values only; more often used with text strings to display values appropriately.
FIRSTNAMEFIRST	(text)	Function for reorganizing names formatted with family name first followed by a comma and then first name.
LOWER	(text)	Returns 'text' where all characters are in lower case.
PROPER	(text)	Returns '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.
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'. Also consider the SUBSTITUTE function below within this topic.

Modify Function	Parameters	Description
		This function is very strict in that the actual location and number of the characters to be replaced must be specified. Miscalculation of the position and length of replaced characters can result in an undesired outcome. It is a good idea to test your value template using the Evaluate feature within the Function Editor dialog.
SUBSTITUTE	(text, from, to [, indexno])	Returns 'text' where all instances of text 'from' are replaced with text 'to.' Comparable to typical 'find and replace' functionality in a standard text editor. Also consider the REPLACE function above within this topic. [indexno] is optional. If supplied, only that one instance is replaced. Typically only practical to use for the first instance.
TRIM	(text)	Returns 'text' where leading and trailing spaces are removed.
UPPER	(text)	Returns 'text' where all characters are in upper case.

Use Cases

For a scenario using the CONCATENATE function, see **Total Cost of Materials Scenario**.

Additional use cases for the CONCATENATE function are included in the **Test Functions** section below.

Regular Expression Functions

These functions can be used to search and modify text based on a defined template.

- 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 '\\' or to get a regexp string like 's', use double escaping of backslash character like '\\s'. 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.

For more information on regular expression, see the **Regular Expression** topic in **System Setup / Super User Guide** documentation.

Regular Expression Function	Parameters	Description
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.
REGEXPSEARCHI	(searchtext, text [, pos])	Same as function REGEXPSEARCH, but search is case insensitive. [pos] is optional, if not found 'N/A' is returned.
REGEXPSUBSTITUTE	(text, from, to)	Returns 'text' where all instances of text 'from' is replaced with text 'to'.

Representative Text Functions

These functions can be used to produce an alternative value for the defined text.

Representative Text Function	Parameters	Description
METAPHONE3	(text)	Returns 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)	Returns the alternate Metaphone 3 code.
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,

Representative Text Function	Parameters	Description
		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. 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.

Test Functions

The functions can be used to perform a test and return a result.

Text Function	Parameters	Description
EXACT	(text1, text2)	Returns 1 (true) if 'text1' equals 'text2', otherwise returns 0 (false). The fields text1 and text2 can be straightforward text , such as 'Yes,' or can be the result of using another function, such as PRODVAL. It is common to use this function along with the IF function to compare an attribute value to a fixed text string. See the example section below, and also the Conditional Functions topic.
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. [pos] is optional. If not found, 'N/A' is returned.

Use Cases

For a scenario using the CONCATENATE and EXACT functions, see **Comparing Localized Values Scenario**.

For a scenario using the CONCATENATE and EXACT functions, see **Including Parent Info on Child Scenario**.

For a scenario using the CONCATENATE and EXACT functions, see **Override Scenario**.

Variable Functions

When a calculated attribute becomes too large to easily modify, you can use one or more variables to separate out parts of it, making it easier to test and manage. Variables allow you to calculate portions of the overall calculation individually, so that it becomes easier to read and maintain. See the use cases below.

A variable is declared within a calculated attribute using a label to identify the purpose, the text ':=' , and then the necessary function(s), for example: [label] := [function]

Multiple variable declarations are added before a function call as a comma-separated list in curly brackets {}. The scope of the variables declared are only available inside the List function parameters that follow. (For more information, see the **List Functions** 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'.

Special Variable Functions

In addition to variables you create as described above, the following special variable functions are available for use within all ITERATE and FILTER function expressions: **index**, **item**, and **size**. These special variable functions can be typed into the Function Editor, or found by clicking the Insert Template button and navigating to Default Templates > All Functions > Special Variables.

Use Cases

For examples of these functions, see **Variable Examples**.

For a scenario using the INDEX function, see **Including Parent Info on Child Scenario**.

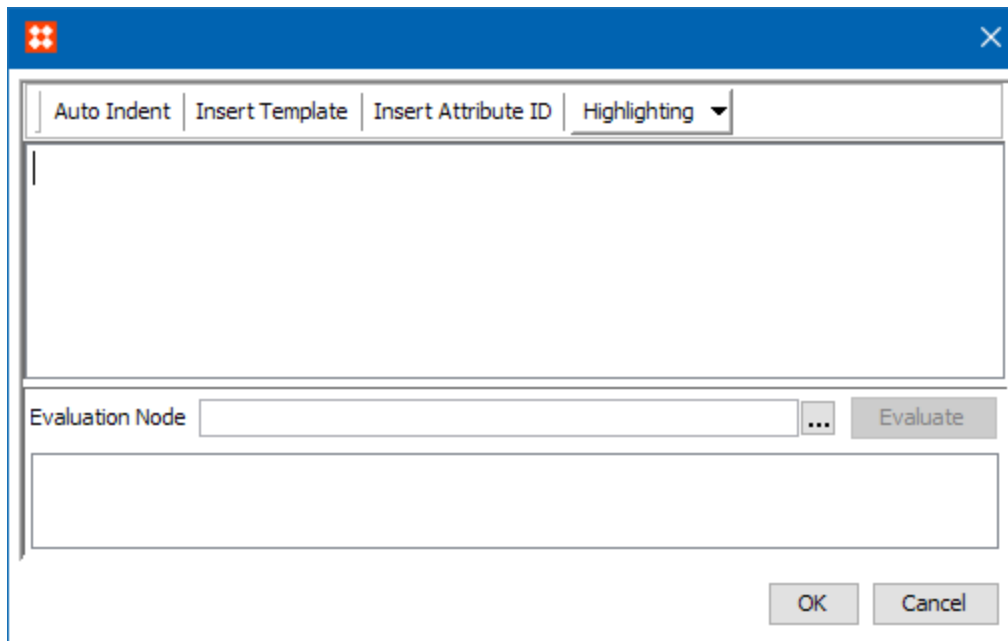
For a scenario using variables, see **Overrides Scenario**.

For a scenario using the INDEX and ITEM functions, see **Total Cost of Materials Scenario**.

Function Editor

The Function Editor (displayed below) can be used to insert STEP functions, for example when adding a key formula on a Key in System Setup, or when creating a value template for a calculated attribute. In addition to writing functions manually, you can create complex functions by using one or more predefined templates.

As described below, some of the places the Function Editor can be accessed include the attribute editor, the bulk update wizard, and in some Flatplanner views. For additional access areas, see the **STEP Functions** topic in **System Setup / Super User Guide** documentation.



Attribute Editor

1. In System Setup, navigate to an attribute that has been configured as calculated.
2. On the Attribute tab, under the Description flipper, locate the 'Value template' parameter.

Attribute	References	Attribute Transformation	Validity	Profile	Log	State Log	Tasks
Description							
Name	>	>	Value				
ID			ColorWidthHeight				
Name			Color Width Height				
Last edited by			2017-04-12 16:01:51 by USERJ				
Full Text Indexable			No				
Externally Maintained			No				
Hierarchical Filtering			None				
Calculated			Yes				
Type			Specification				
Dimension Dependencies							
Value template			Color = value("ProductColor"), Width = value("ProductWidth"), Height = value("ProductHeight")				
Mandatory			No				

- Until a function has been defined, the Value Template parameter displays EMPTY with a gray background. This means that no calculation will be performed (and no value displayed) in the object editor for the objects where the attribute is valid.

> Value template	EMPTY

- When a function exists, it is displayed with a gray background. This means that the calculation of the formula will be performed and displayed in the object editor for the objects where the attribute is valid.

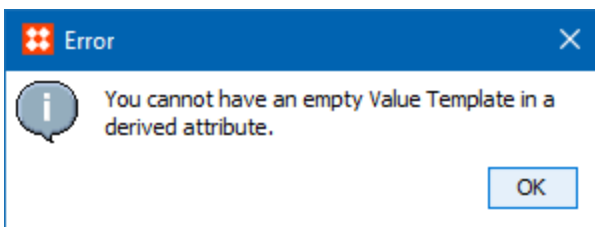
> Value template	list(iterate(classifications(), 'stepname()'), "\n")

- When static text has been defined, it is displayed with a white background. The text entered is displayed in the object editor for the objects where the attribute is valid. Static text can also be used in combination with functions.

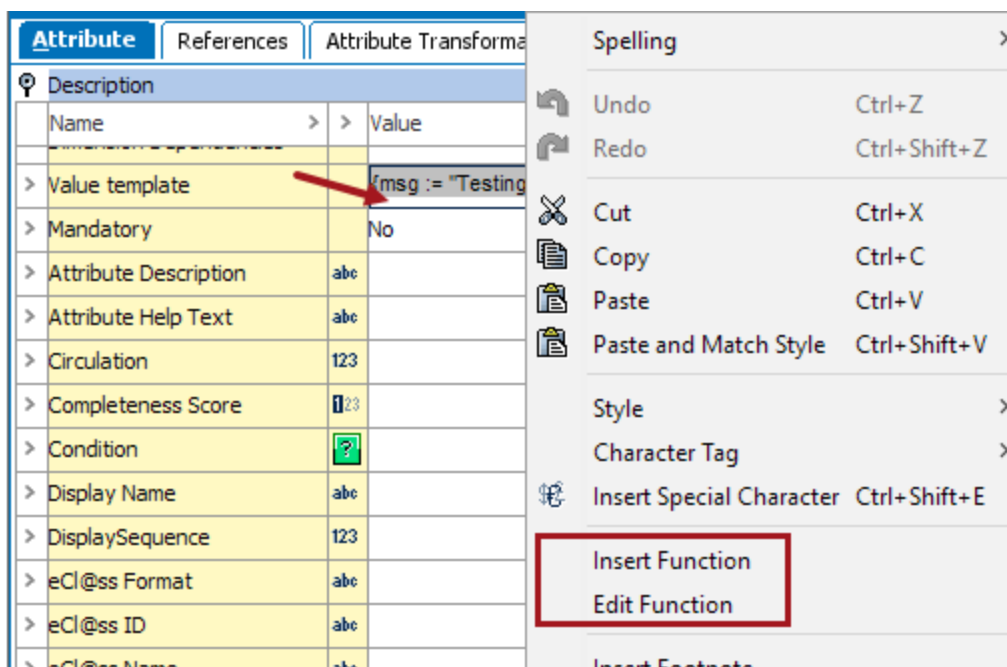
> Value template	static text

Important: No calculations are performed on static text.

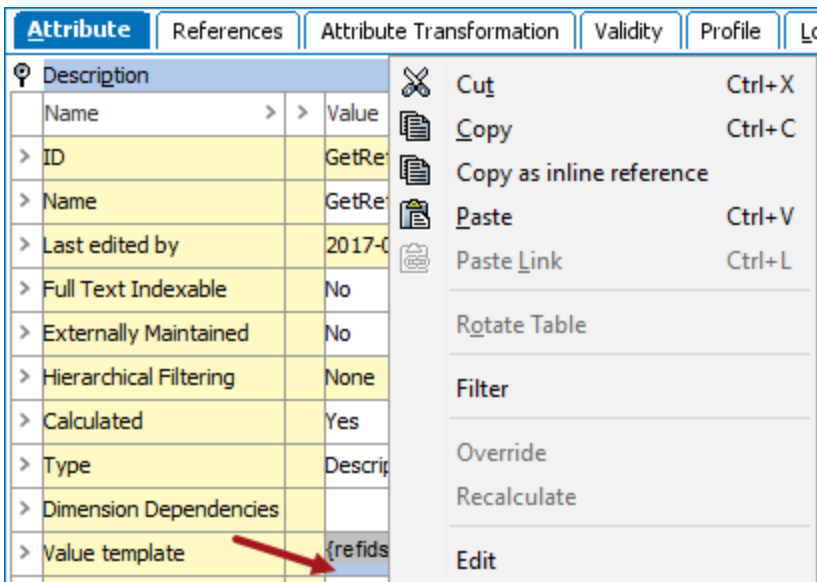
- Once an attribute is configured to be calculated, the Value Template cannot be blank. Enter either a function and/or static text to resolve the following error.



- Click the Value Template parameter value until the existing text is displayed with a white background. (Notice the white area below the text on the gray background.) Right-click to display the menu below with insert or edit function options:



- Right-clicking with a blue background (shown below) displays a menu with limited options and does not include Function options.



4. Edit the Value Template parameter as needed:

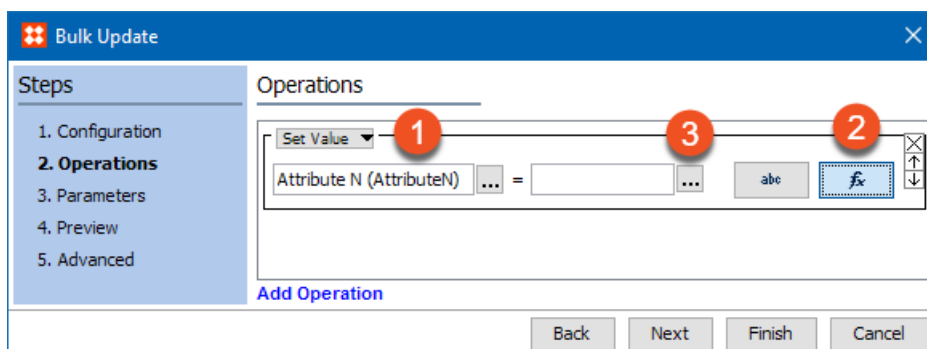
> Value template	Supplier: list(iterate(classifications("SupplierLink"), "stepid()"), "/") Image: list(iterate(classifications("ProductImage"), "stepid()"), "/")

- Click **into the field** and use the right-click **Insert Function** menu option to add a new function.
- Click **on an existing function** in the field and use the right-click **Edit Function** menu option to modify a function.
- Click **before or after an existing function** and type any required static text.

Bulk Update Wizard

The bulk update wizard can be accessed in a number of ways as described in the **Running Bulk Updates** topic of the **Bulk Updates** documentation.

1. On the **Operations** step of the Bulk Update wizard, select the **Set Value** operation from the dropdown.
2. Click the **Function** button ().
3. Then click the ellipsis button (...) to display the Function Editor.

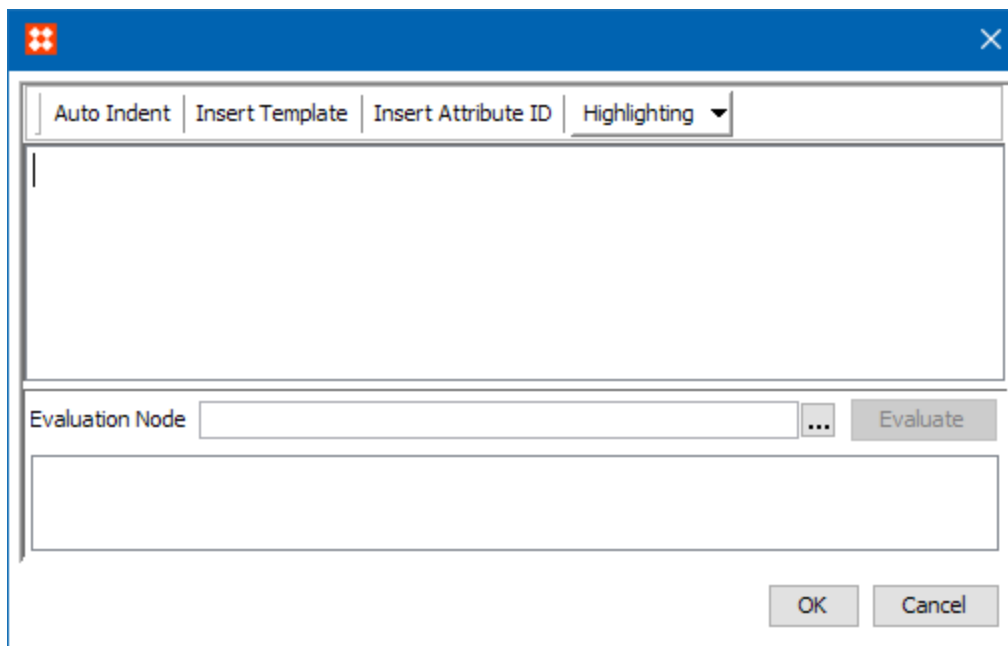


Flatplanner

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 Function Editor

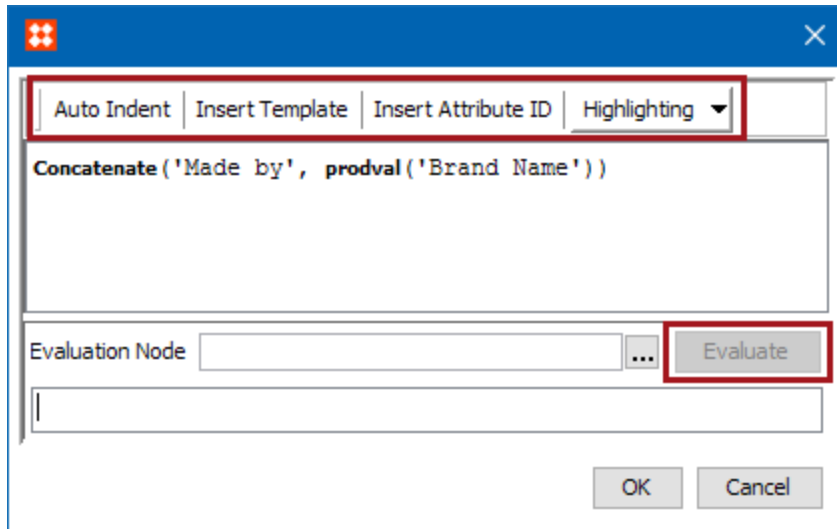
The Function Editor (displayed below) can be used to insert STEP functions, for example when adding a key formula on a Key in System Setup, or when creating a value template for a calculated attribute. In addition to writing functions manually, you can create complex functions by using one or more predefined templates.



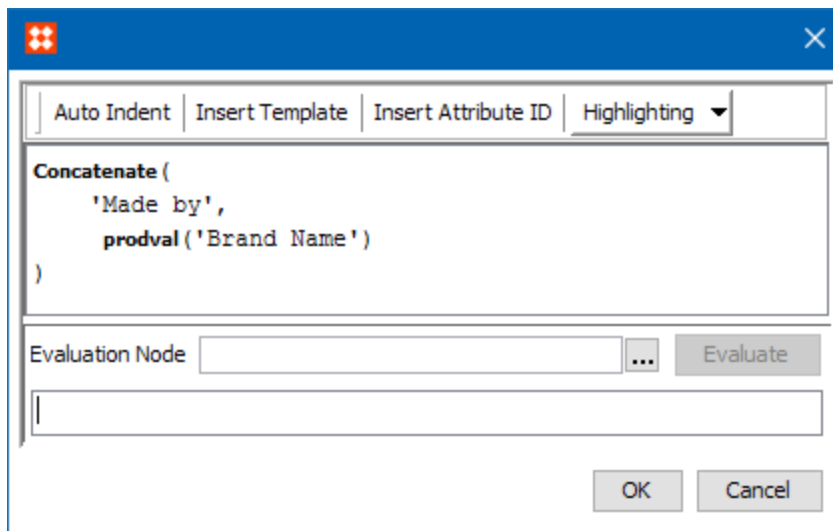
If you know the name and required arguments of the needed function, you can type directly into the upper text field. For more information on the available functions and their syntax, see the for **STEP Functions** topic and the **Inserting Templates** topic in the **Calculated Attributes** documentation.

Note: To display values in multiple lines or to add a new line, press the Enter key or add "\n" within the Function Editor template.

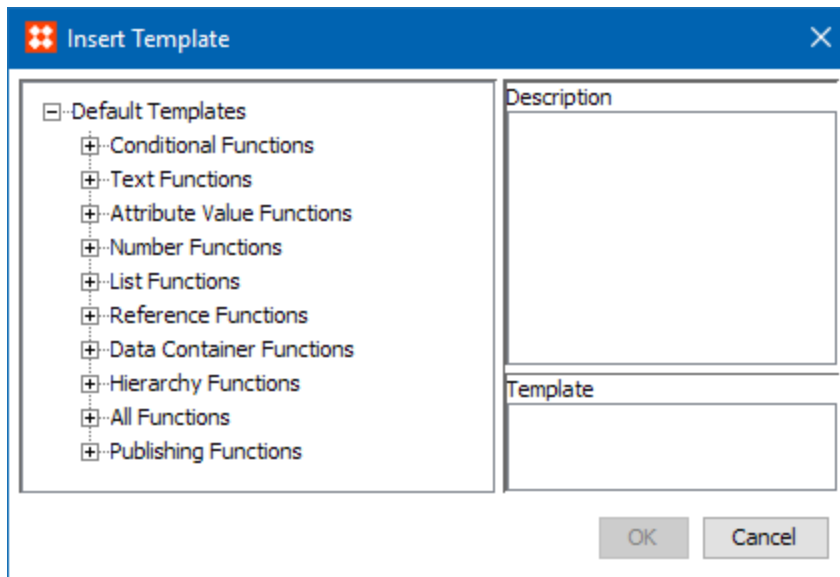
The Auto Indent, Insert Template, Insert Attribute ID, and Highlighting options can be accessed in the Function Editor either by clicking on the menu tabs across the top, or by right-clicking inside the upper text box and choosing from the popup menu. Each of these options are described below.



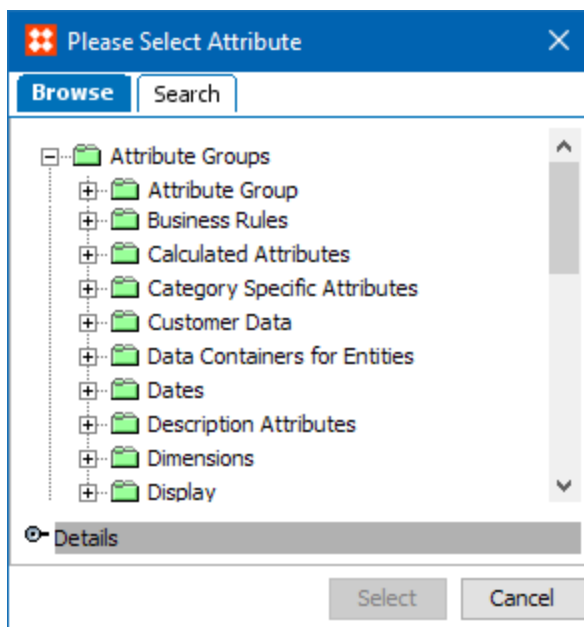
- **Auto Indent:** Displays the function in a typical coding layout. Indents and hard returns are added for an easier-to-read format, as shown below. When working on a complex formula, this can help ensure the parentheses are paired correctly (each open parenthesis is followed by a closing parenthesis).



- **Insert Template:** Allows a pre-configured calculated attribute function template to be inserted at the current location of the cursor. This is helpful when you do not remember the exact name of a function. 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. IDs are required for attributes and units. For more information, see the **Calculated Attributes** documentation for **Inserting Attribute IDs in Functions**.



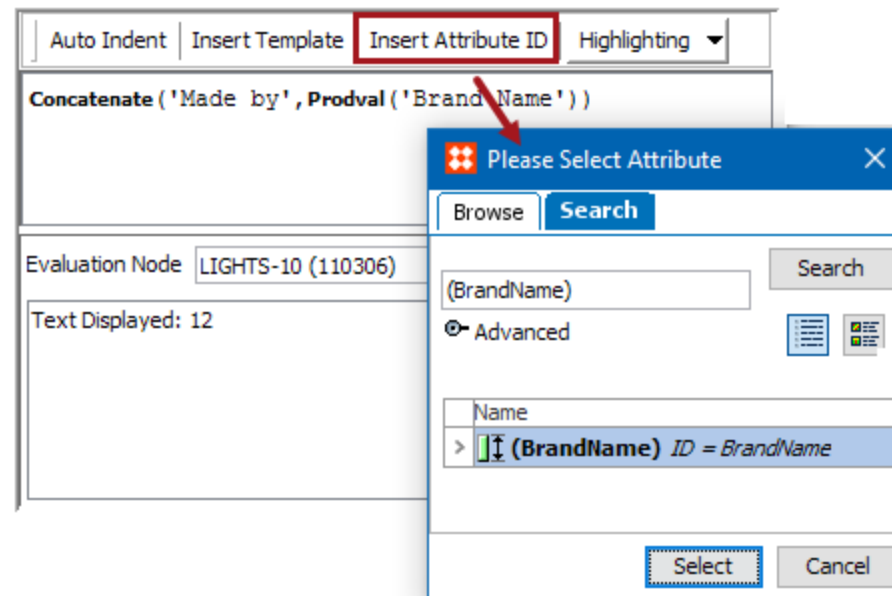
- **Highlighting:** Displays or removes highlighting for parentheses and/or bold for function names.
- **Evaluate:** Tests part of a function or an entire function against an object in the Tree. When writing a new formula, it is a good idea to have a test object prepared with the data you want to access. For more information, see the **Calculated Attributes** documentation for **Evaluating Functions**.

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.

Important: 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.



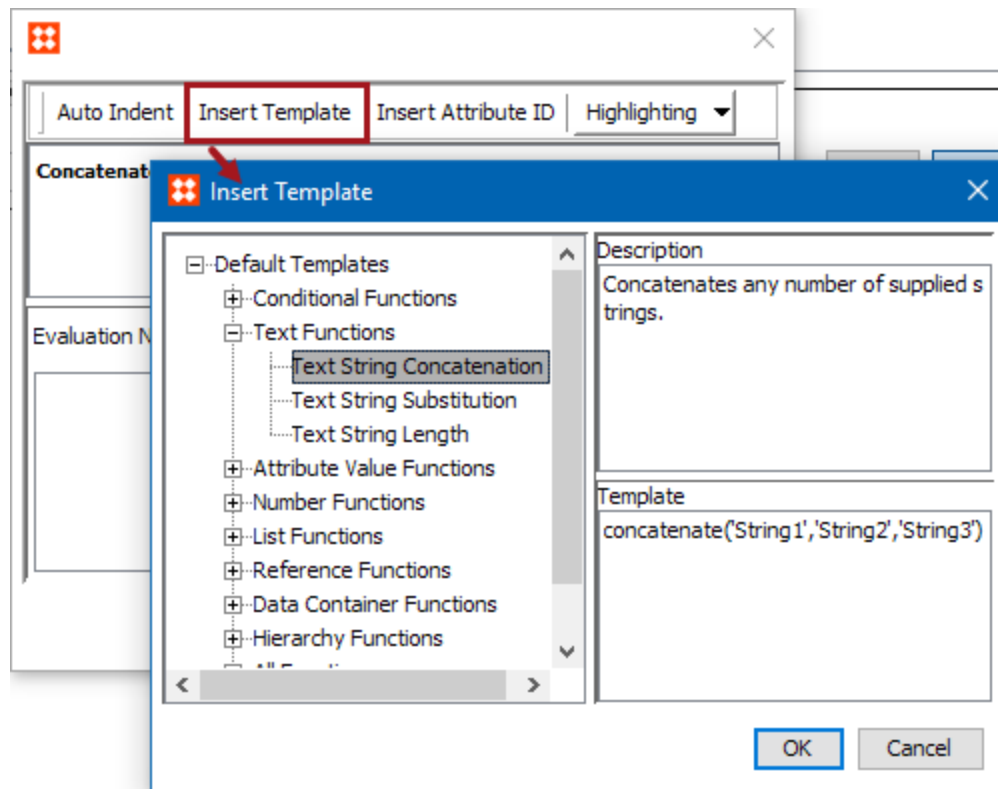
You may also perform some advanced search operations to find the right attribute, or use wildcards in searches. See the **Navigation and Searches** section in the **Getting Started / User Guide** documentation.

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** documentation.

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
```

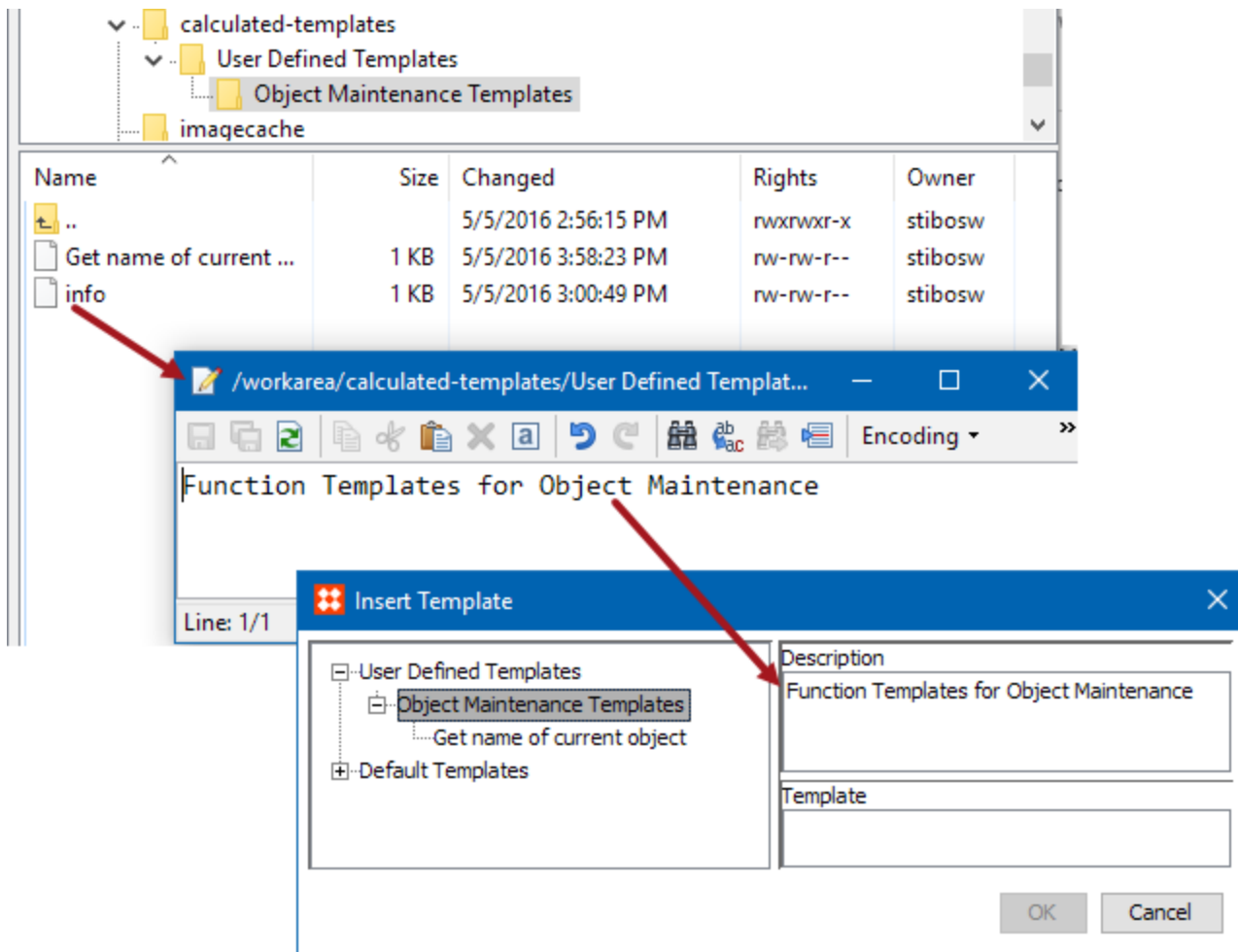
For Windows, the folder is normally found in a drive with the path similar to below:

```
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.

The screenshot displays the STEP Trailblazer 8.2 interface. At the top, a file explorer shows the directory structure: `calculated-templates` (expanded), `User Defined Templates` (expanded), `Object Maintenance Templates` (selected), and `imagecache`. Below this is a table listing files:

Name	Size	Changed	Rights	Owner
..		5/5/2016 2:56:15 PM	rw-rw-r-x	stibosw
Get name of current object	1 KB	5/5/2016 3:58:23 PM	rw-rw-r--	stibosw
info	1 KB	5/5/2016 3:00:49 PM	rw-rw-r--	stibosw

A red arrow points from the `Get name of current object` file in the table to a code editor window titled `/workarea/calculated-templates/User Defined Te...`. The code editor contains the following text:

```
#Function for getting NAME of current object  
stepname()  
  
Line: 1/12
```

Another red arrow points from the `Get name of current object` text in the code editor to the `Insert Template` dialog box. The dialog box shows a tree view of templates:

- User Defined Templates
 - Object Maintenance Templates
 - Get name of current object (selected)
- Default Templates

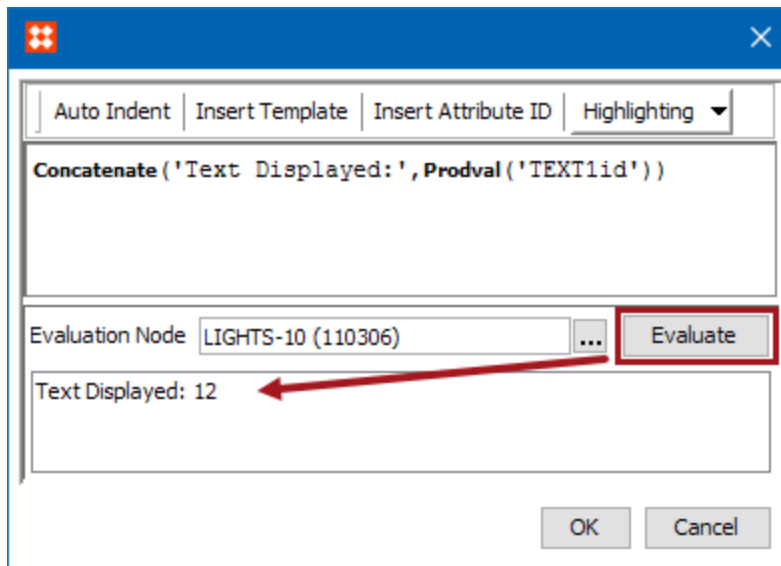
The `Description` field contains: `Function for getting NAME of current object`. The `Template` dropdown shows: `stepname()`. At the bottom are `OK` and `Cancel` buttons.

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 typeahead 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'

Note: To evaluate only a specific part of the function, highlight it and click the Evaluate button. Ensure that the parentheses are paired (opened and closed) for the selection. If there is any error in the code, or if the parentheses are not handled correctly, the evaluation result will be 'N/A'.



The Function Editor is available in several locations within STEP. For more information, see the **STEP Functions** documentation.

Creating a Calculated Attribute

Although a calculated attribute must be created in workbench, it is available for use in both workbench and Web UI.

Important: A calculated attribute is not always the best practice. Before creating a calculated attribute, it is important to be certain a different (i.e., more efficient) STEP method should not be used instead. For more information, see the **Calculated Attribute Considerations** topic within this guide.

1. In workbench > System Setup > select the attribute group that will hold your calculated attribute or create a new group.
2. Right-click the attribute group, and select the **New Attribute** option. The Create Attribute wizard displays and the selected group is provided for the Select Attribute Group step of the wizard.
3. On the **Enter ID and Name** step of the wizard, provide the following information:
 - Type an **ID** and **Name** for your new attribute.
 - Set **Validation Base Type**, keeping in mind that the assigned type only takes effect when the calculated value is manually overridden. For more information, see the **Attribute Validation Flipper** section of the **Setting up a Calculated Attribute** topic.
 - Set **Multi Valued** to **No**. This is required for a calculated attribute.
 - Select **Specification** or **Description** based on your requirements. For more information, see the **Description or Specification** section of the **Deciding Settings for a Calculated Attribute** topic.

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: ListOfRefs

Name: List Of References

Validation Base Type: Text

Multi Valued: No

☒ Specification

☐ Description

Back Next Finish Cancel

- Click the **Next** button.

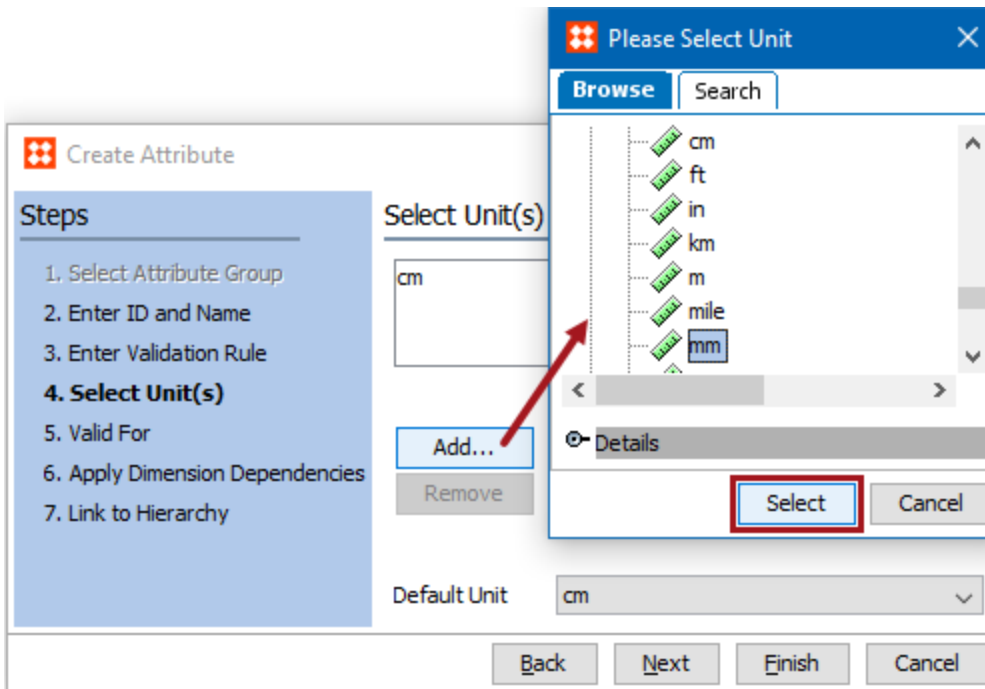
4. On the **Enter Validation Rule** step of the wizard, provide the required information, based on the validation base type selected (as shown in the image below):
 - Type an **Input Mask** if required. For more information, see the **Input Masks** section of the **Validation Rules** documentation and the **Restrict Override on a Calculated Value** section of **Overriding and Restoring Calculated Attribute Values**.
 - For Validation Base Types of Embedded Number, Fraction, Integer, Number, or Numeric Text, set a **Minimum Value** and/or **Maximum Value** if required. For more information, see the **Validation Rules** documentation.
 - For other Validation Base Types, set a **Maximum Length** if required. For more information, see the **Validation Rules** documentation.

The image displays two screenshots of the 'Create Attribute' wizard, specifically the 'Enter Validation Rule' step. The top screenshot shows a table with two rows: 'Input Mask' and 'Maximum Length' (with a value of 100). The bottom screenshot shows a similar table but with 'Minimum Value' and 'Maximum Value' instead of 'Maximum Length'. Both screenshots show a 'Steps' sidebar on the left with '3. Enter Validation Rule' highlighted. The bottom screenshot also includes 'Back', 'Next', 'Finish', and 'Cancel' buttons at the bottom.

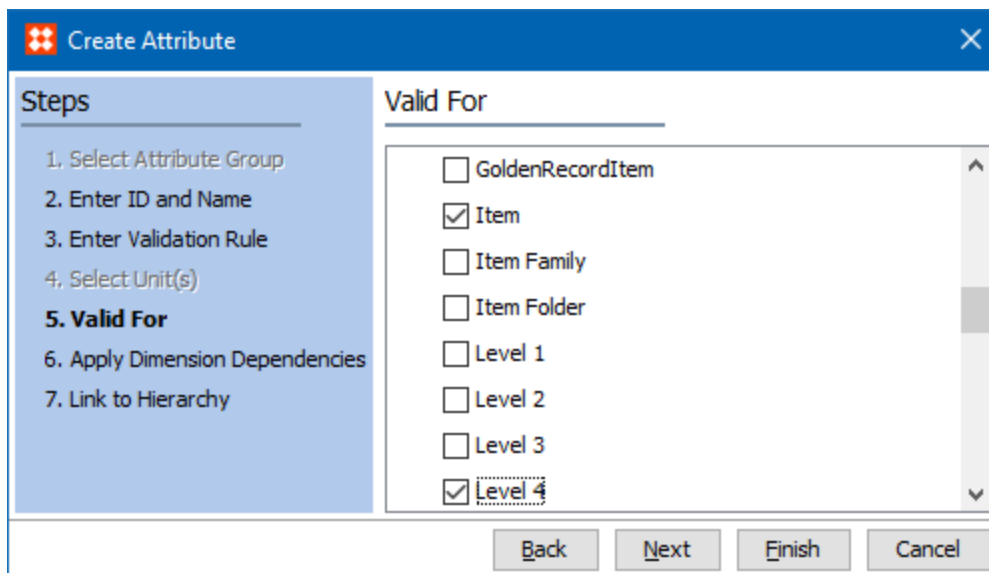
Name	Value
Input Mask	
Maximum Length	100

Name	Value
Input Mask	
Minimum Value	
Maximum Value	

- Click the **Next** button.
5. The **Select Unit(s)** step of the wizard is only enabled when the Validation Base Types is Embedded Number, Fraction, Integer, Number, or Numeric Text. Add the following data if required:
 - Click the **Add** button to display the Please Select Unit dialog. Choose a unit and click the **Select** button. The selected unit is displayed in the text box. Add as many units as are required.



- When a unit exists in the text box, the **Remove** button is enabled. Click it to remove the selected unit from the text box.
 - Use the **Default Unit** dropdown to set the unit that will be automatically applied to the calculated attribute value.
 - Click the **Next** button.
6. On the **Valid For** step of the wizard, select the object types where the calculated attribute can be edited. This sets the attribute editor Validity tab selections.



- Click the **Next** button.
7. On the **Apply Dimension Dependencies** step of the wizard, leave all checkboxes unchecked unless you have a reason to set a dimension. Dimension dependencies are only relevant when overriding the calculated value. For more information, see the **Deciding Settings for a Calculated Attribute** topic.

The screenshot shows the 'Create Attribute' wizard window. The title bar says 'Create Attribute'. On the left, a 'Steps' pane lists seven 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 (highlighted in blue), and 7. Link to Hierarchy. The main area is titled 'Apply Dimension Dependencies' and contains two unchecked checkboxes: 'Country' and 'Language'. At the bottom, there are four buttons: 'Back', 'Next' (highlighted with a blue border), 'Finish', and 'Cancel'.

- Click the **Next** button.
8. On the **Link to Hierarchy** step of the wizard, use the Browse and/or Search tab to find and select the hierarchy nodes where the new attribute can be viewed.

The screenshot shows the 'Create Attribute' wizard window at Step 7: 'Link to Hierarchy'. The 'Steps' pane on the left highlights step 7. The main area has two tabs: 'Browse' (selected) and 'Search'. Under the 'Browse' tab, a tree view shows a hierarchy starting with 'Products', which contains sub-items: 'Apparel', 'Footwear', 'Safety', 'Hardware', 'Displays', and 'Furniture'. The 'Apparel' item is selected and highlighted. Below the tree is a 'Details' section. At the bottom, there are four buttons: 'Back', 'Next' (highlighted with a blue border), 'Finish', and 'Cancel'.

This creates an entry on the attribute editor References tab, under the Valid in Products flipper as shown below.

<

List Of References - References

▶

▶

Attribute

References

Attribute Transformation

Validity

Profile

Log

State Log

Tasks

Valid in Classifications

Valid in Products

ID

>

Name

>

DisplaySequence

>

Product Variant Priority ..

>

18200

Apparel

>

Link to Product

Source from Attribute

In Attribute Groups

Values for Attribute

Value Distribution

- Click the **Finish** button on the wizard. The attribute editor is displayed for the new attribute in the attribute group. Continue with the manual configuration steps below to make the attribute calculated.
9. In the attribute editor, on the Attribute tab, within the Description flipper, double click the **Calculated** field to display the Change Calculated Setting dialog. Click the **Yes** radio button, and click the **Save** button.

<

List Of References - Attribute

▶

▶

Attribute

References

Attribute Transformation

Validity

Profile

Log

State Log

Tasks

Description

Name

>

>

Value

>

ID

ListOfRefs

>

Name

List Of Refere

>

Last edited by

2017-07-05 10

>

Full Text Indexable

No

>

Externally Maintained

No

>

Hierarchical Filtering

None

>

Calculated

No

>

Type

Specification

Change Calculated Setting

Is the attribute calculated:

Yes

No

Save

Cancel

- The 'Value template' parameter displays in the editor with the text 'EMPTY' and a gray background.
- >

Value template

EMPTY
- Based on the validation base type selected in the previous wizard step, the 'Unit template' parameter displays in the editor with no text.

>

Unit template

10. Open the Function Editor for the 'Value template' and 'Unit template' parameters to supply the calculation. For detailed steps, see the **Attribute Editor** section of the **Function Editor** documentation.
11. Use the Function Editor to update the 'Value template' and 'Unit template' parameters to construct the required functions. For details on available functions, see the **STEP Functions** section or the **Calculated Attribute Use Cases** section.

Overriding and Restoring Calculated Attribute Values

When a calculated attribute displays on an object, the value is read only. Occasionally, the calculated value for a specific product, even though correct, may not be what is desired. Below are options for overriding calculated attributes, restricting override, removing override values, and restoring a calculated value.

Note: For calculated attributes, the value is expected to be controlled by the resulting value within the actual calculation, so the settings for the attribute's validation base type, assigned unit, and dimension dependencies etc. are all ignored. It is only when a manual override is applied to the attribute value that those limits and setups are active.

Override a Calculated Attribute

Although a calculated attribute value is displayed as read-only in the workbench or Web UI Tree object editor, it can be overridden.

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 **Override**.

Tree

Primary Product Hierarchy

Products

Apparel

Upper Body Wear

T-shirts

T-shirts Items

T-Shirts Sales Items

T-shirts

18217-0542

18207-012

18214-012

18215-012

18217-012

Head Wear

Footwear

Safety

18217-0542 rev.0.27 - Product

ProductSub ProductsReferencesReferenced ByImages & Documents

Description

Name	Value
ID	100305
Name	18217-0542
Object Type	Sales Item
Revision	0.27 Last ed
Approved	✓ Approve
Translation	Not Translat
Path	Primary Proc
GetRefTypeID	PrimarySupp
Parent	T-shirts

Path

GetRefTypeID

Parent

Primary Product Hierarchy/Products/Apparel/Up

PrimarySupplierItem

T-shirts

Cut

Copy

Copy as inline reference

Paste

Paste Link

Rotate Table

Filter

Override

The parameter can then be edited, allowing you to enter the value to replace the calculated value. If values existed prior to clicking Override, then those values become static text. When static text has been defined, it is displayed with a white background. No calculations are performed on static text. The text entered is displayed in the object editor for the objects where the attribute is valid. Static text can also be used in combination with functions.

Another option is when a space added as a value is used to override a calculated attribute value. In the following image, Status 1 is a calculated value that uses data from Product Status, in this case the value of 'Active.' Status 2 adds the value of Status 1 to some text.

Name	>	>	Value	>
> Product Status	abc		Active	
> Status 1	fx		18216 L O is currently Active	
> Status 2	fx		Notes for 18216 L O include: 18216 L O is currently Active	

When Status 1 gets a space value for an override, that space is passed onto Status 2.

Name	>	>	Value	>
> Product Status	abc		Active	
> Status 1	abc			
> Status 2	fx		Notes for 18216 L O include:	

This method can be used to overwrite calculated values per object.

Name	>	>	Value	>
> Product Status	abc		Active	
> Status 1	fx		18216 L O is currently Active	
> Status 2	fx		Notes for 18216 L O include: 18216 L O is currently Active	

Name	>	>	Value	>
> Product Status	abc		Active	
> Status 1	abc			
> Status 2	fx		Notes for 18216 L O include:	

Restrict Overrides of a Calculated Attribute Value

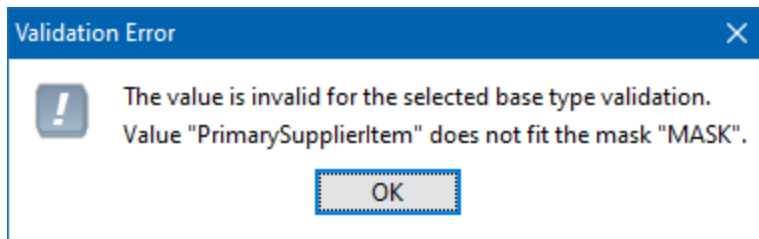
Calculated attribute values can be overridden as described above, and can also be overwritten via import.

For example, consider an export that includes the calculated attribute values (among other values). After making a change to non-calculated attributes, you then re-import the file. Since the file contains the value of the calculation, the calculation is overridden with an actual value, making it a local value. In this case, it might be beneficial to apply a mask to prevent the calculated value from ever being overridden unintentionally.

Important: Using an input mask places a restriction on the attribute override value. The only override allowed via keyboard entry or import is one that matches the mask. For more information, see the **Input Masks** section of the **Validation Rules** documentation.

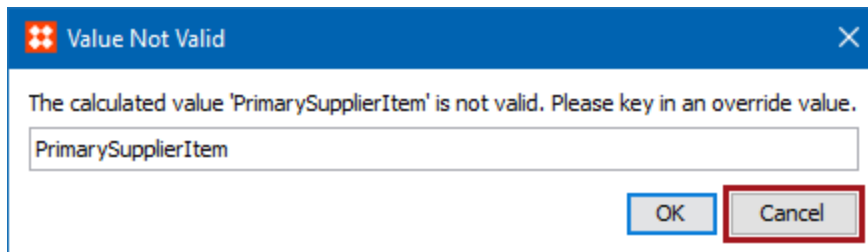
Continuing with the override example above, a mask of 'MASK' has been added to the calculated attribute named 'GetRefTypeID'.

1. Click the **Override** option for a calculated attribute value that includes a mask. The following validation error displays and compares the calculated value to the mask.



Note: This error is also displayed if override values already exist for a calculated attribute that is being modified to include a mask. In this case, follow the steps in the **Remove Override Values** section below, and then add the mask.

2. The Value Not Valid dialog only allows adding the mask text as an override. The user can either:
 - add the required mask text, or
 - click Cancel to resume using the calculated attribute value.



Remove Override Values

To delete existing override values:

1. Use Export Manager to create an export of the attribute (and any other attributes needed for identification) for all objects. For more information, see the **Creating a Data Export** documentation.
2. On the Export Manager wizard Advanced step, uncheck the 'Include Calculated Attribute Values' Advanced option. This outputs only local values. For more information, the **Export Manager - Advanced** documentation.
3. In the output export file, replace the attribute's local values with '[delete]' and save the file. For more information, see the **Delete Values - Map Inbound** documentation.
4. Use Import Manager to import the modified file. For more information, see the **Creating a Data Import** documentation.

Restore a Calculated Value

To restore the calculated value in workbench or Web UI, simply delete the manually entered value to have the calculated value display again.

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 in 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**.

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: ConditionAttribute

Name: Condition Attribute

Validation Base Type: Text

Multi Valued: ☐ Specification ☒ Description

Condition
Date
Embedded Number
Fraction
Integer
ISO Date
ISO Date and Time
List Of Values

Back Next Finish Cancel

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

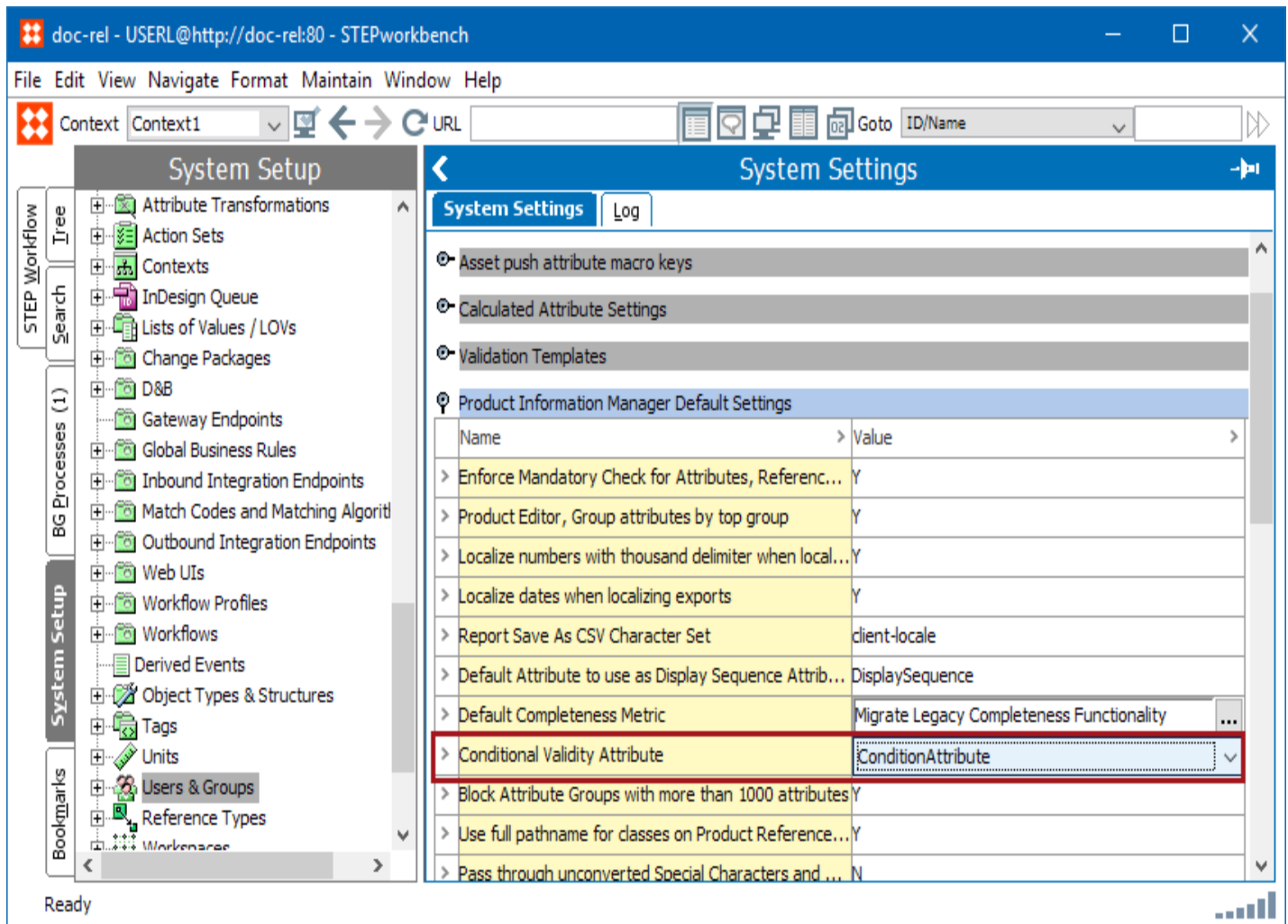
☐ All
☐ Classification attribute validation
☒ Product attribute validation

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.



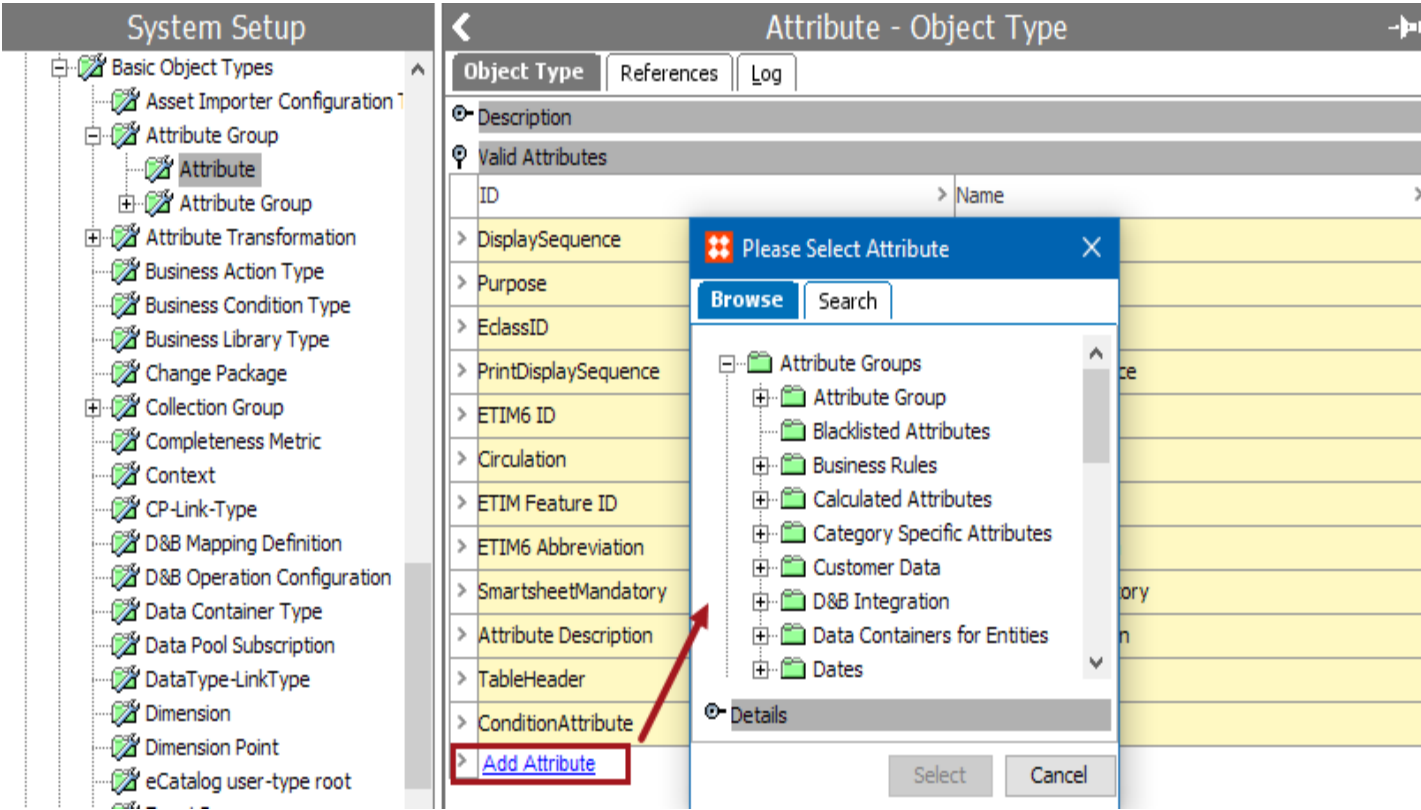
Configuring Conditionally Valid Attributes

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 browse or 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.

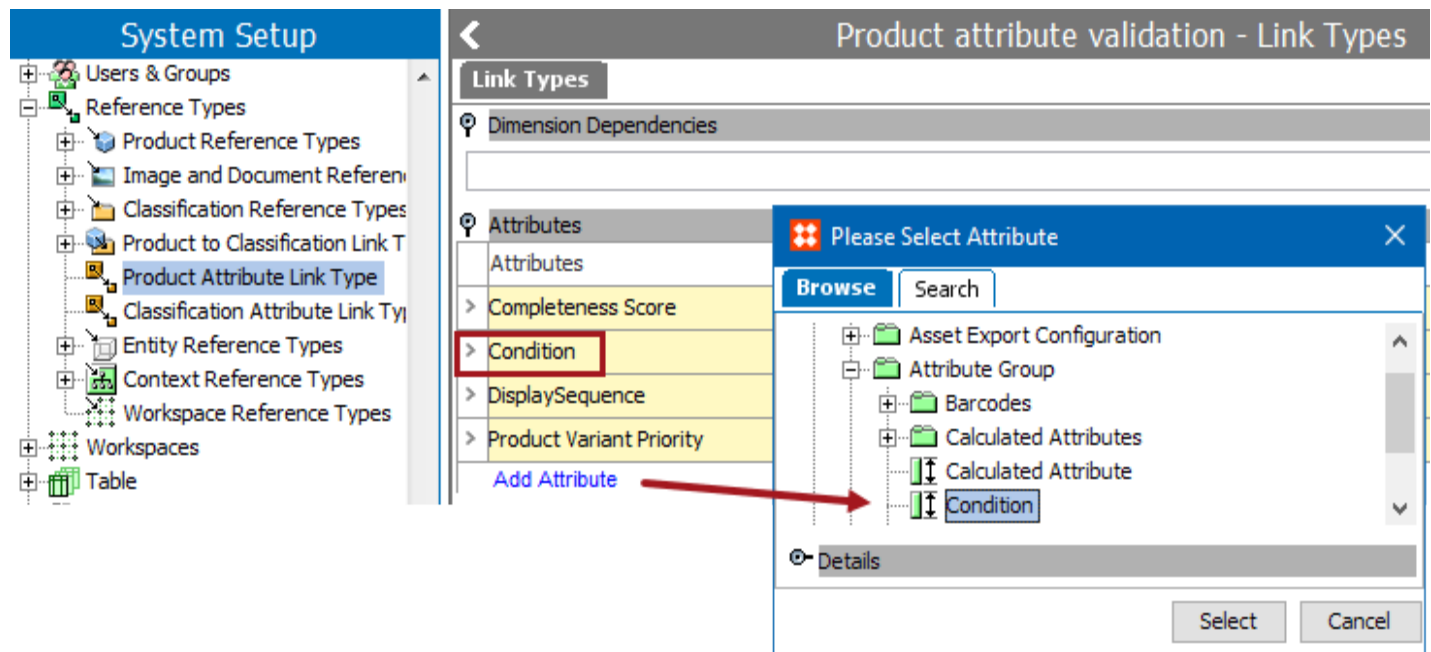
- 3. Next, navigate to **System Setup** and select the desired Description Attribute.
- 4. In the Conditional Attribute field, enter the desired condition rules for the Conditionally Valid Attribute to follow.

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 browse or select the Conditional Attribute.



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

Tree		Playstation 4 Games rev.0.3 - References					0% complete
- Furniture - Automotive - Building Products - Electrical and Electronics - Electronic Accessories - Wire & Cable - Game Software - Xbox One - Xbox One Games - Playstation 4 - Playstation 4 Games - Bloodborne - Destiny - Until Dawn - Food and Beverage - Kitchen - Wedding - Party Supplies - Discontinued Products - Product Overrides - Parking		Product					Referenced By
		References					
		Linked Attributes from Product Hierarchy					
		DisplaySequence	ID	Name	Completeness Score	Condition	
		5	AirTransportationRestricti...	Air Transportation Restriction			
		2	AnnualSalesForecastMaxi...	Annual Sales Forecast, Maxim			
		1	AnnualSalesForecast,Mini...	Annual Sales Forecast, Minim			
		5	AttributeA	Attribute A			
		5	AttributeB	Attribute B			
		5	Attribute1	Attribute C			
		5	AvailabilityEnd	Availability End			
		2	AvailabilityStart	Availability Start			
		1	BaseUnitOfMeasure	Base Unit of Measure			
		5	BrandName	Brand Name			
		6	BrandOwner	Brand Owner			
		5	ChildCount	ChildCount			
		>	Co-op	Co-op			Multiplayer = Yes

Writing Conditional Expressions

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

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

Syntax forms:

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

Rules that must be observed for the conditions:

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

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

For more information see the **Business Rules with 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

A number of Web UI components centered around the Task List and Node Editor components are enabled to deal with the conditional validity of attributes. These components honor the conditional validity settings and only allow for display and/or editing of attribute values if they are valid based on the driving conditions.

For 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').

Primary Product Image



ID	181120
Name	Acme Anvil
Object Type	Item
Approved	✖ Last approved 2/28/17 2:41 PM
Long Item Description	No anvil made, English or American, surpasses our Acme in shape, material or finish. It is solid forged of two pieces of best wrought iron, welded at waist; face is made of one piece of tool steel, electrically welded to the body and warranted not to come loose. Base has sufficient spread to insure stability and prevent tipping; has long perfectly shaped horn and heel; face is trued and shaped by a special machine so that there are no hollow or uneven places; edges are perfectly tempered and will not chip.

[Edit in Rich Text](#)

Has Lead	Yes

[Edit in Rich Text](#)

Hazmat	

[Edit in Rich Text](#)

Basic Information and references

Category Informations

Additional Information

Packaging Hierarchy

Primary Product Image



ID

181120

Name

Acme Anvil

Object Type

Item

Approved

✖ Last approved 2/28/17 2:41 PM

Long Item Description

No anvil made, English or American, surpasses our Acme in shape, material or finish. It is solid forged of two pieces of best wrought iron, welded at waist; face is made of one piece of tool steel, electrically welded to the body and warranted not to come loose. Base has sufficient spread to insure stability and prevent tipping; has long perfectly shaped horn and heel; face is trued and shaped by a special machine so that there are no hollow or uneven places; edges are perfectly tempered and will not chip.

[Edit in Rich Text](#)

Has Lead

[Edit in Rich Text](#)

Conditional attribute functionality can be used with Web UI elements:

- ParentValue
- TargetAttributeColumn
- TargetAttributeLinkColumn
- TargetValueHeader
- Value
- ValueHeader
- ValueGroup
- ValueLinkHeader
- ValueGroupHeader

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. There is no value inheritance to the nodes below.

Description attributes can also be applied to references, such as product to product references and also in the link between a product node and an attribute node.

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 attribute(s) linked to the product(s).

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.

Yes: Must have a value for the specified attribute linked to the Product

No: Not Mandatory, may have the value for the specific attribute linked to the product.

The screenshot displays the 'Attribute N - Attribute' configuration window. The left pane shows a tree view of 'Attribute Groups' with 'Attribute N' selected. The right pane shows the 'Attribute' tab with a table of attributes. The 'Mandatory' attribute is highlighted with a red box, and its value is set to 'No'.

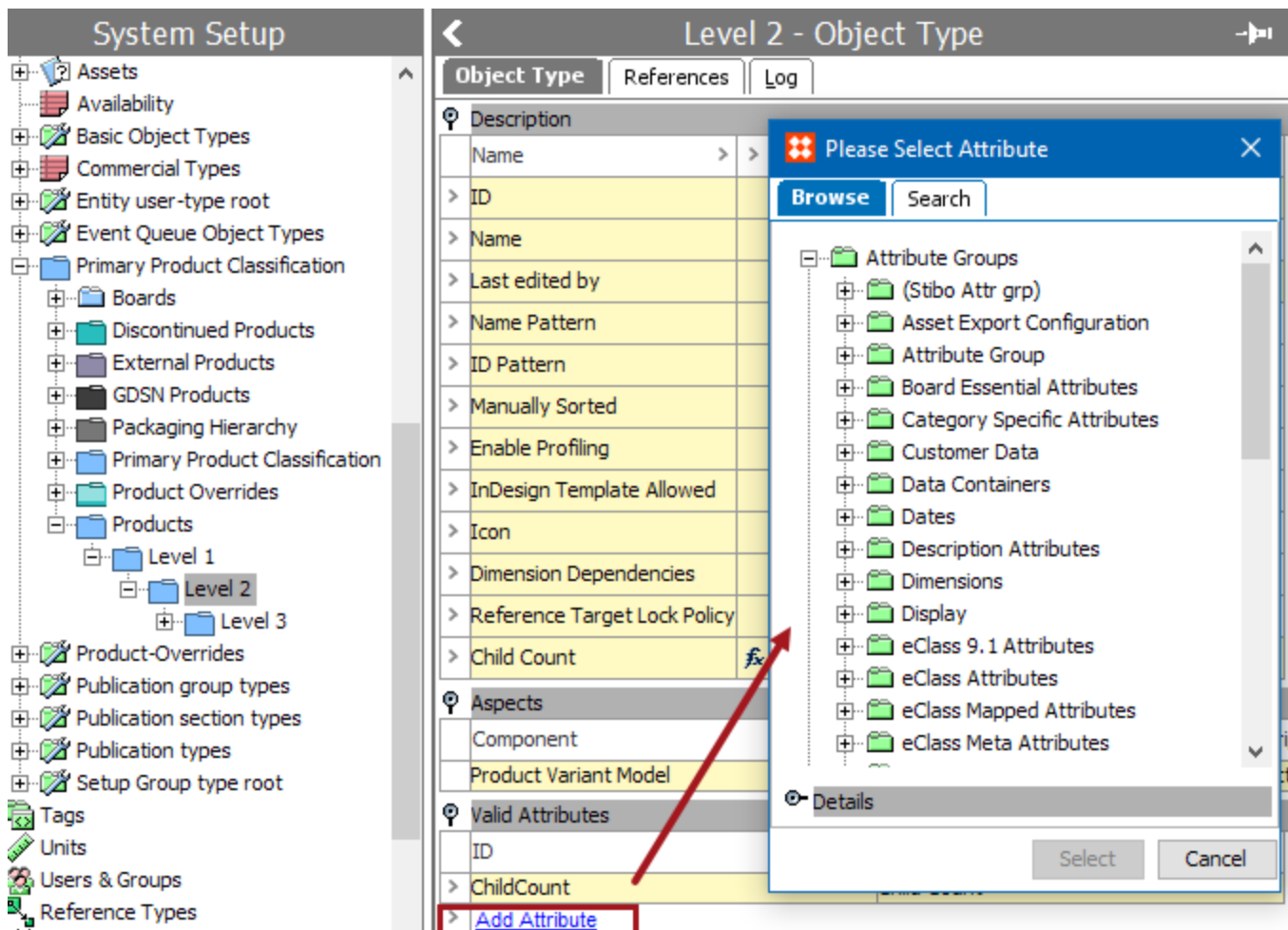
Name	Value
ID	AttributeN
Name	Attribute N
Last edited by	2017-01-26 11:09:28 by USERL
Full Text Indexable	No
Externally Maintained	Yes
Calculated	No
Type	Specification
Dimension Dependencies	
Mandatory	No
Attribute Description	abc
Circulation	123
Completeness Score	1/23

For more information about mandatory option please see the **Product Information Manager Default Settings** section of the **Super User Guide / System Setup** documentation.

Linking Description Attributes to an Object Type

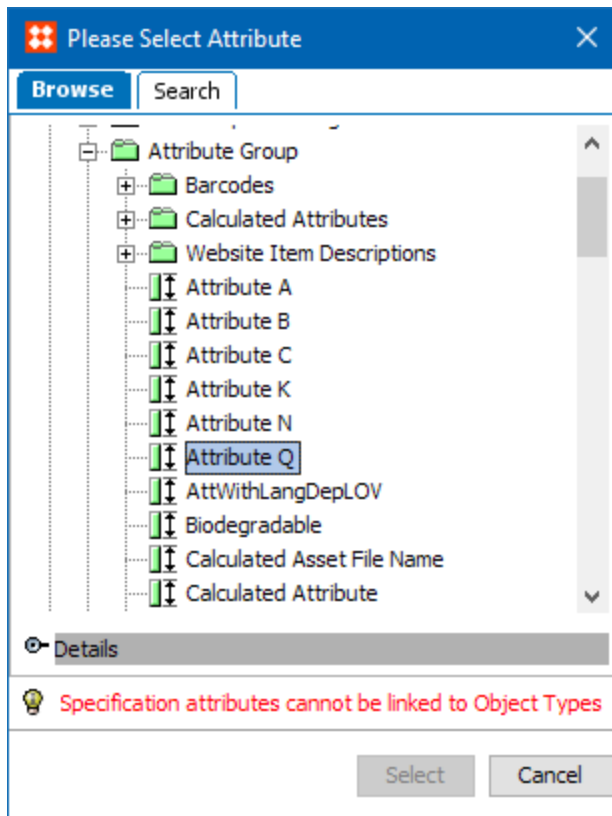
Once description attributes are created, they need to be linked to an object types. To do so, follow the steps below.

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. The 'Select' option will be grayed out for the specification attribute.



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

Tree

- Products
 - Apparel
 - Upper Body Wear
 - Head Wear
 - Hats and Caps
 - Hats and Caps
 - Baseball Cap
 - Baseball Hat
 - Gray Cap
 - Gray Hat
 - GRAY HAT
 - Industrial Hat
 - Orange Cap
 - Pink Cap
 - Red Cap
 - Red Hat
 - Ski Hat
 - Ski Helmet
 - Yellow Cap
 - Hats and Caps
- Footwear
 - Shoes
 - Pumps

Gray Hat rev.0.12 - Product

0% complete

Commercial
Tables
Category Profile
Proof View
Status
State Log
Tasks

Product
Sub Products
References
Referenced By
Images & Documents

Name	Value
ID	114509
Name	Gray Hat
Object Type	Item
Revision	0.12 Last edited by USERA on Tue Feb 21 13:30:2...
Approved	✓ Approved on Tue Feb 21 13:30:23 EST 2017
Translation	Not Translated
Path	Primary Product Hierarchy/Products/Apparel/Head...
Default InDesign template	Holiday Product Template (108803)
UPC	abc
EAN	GT3 0614141123452
Path	Products Apparel Head Wear Hats and Caps Hats and Caps Items
Planned Release Date	31
Print Display Sequence	abc
Product Image Description	abc
Qty of Next Higher Package	123

No Primary Image

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.

Create 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 / Super User Guide** 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

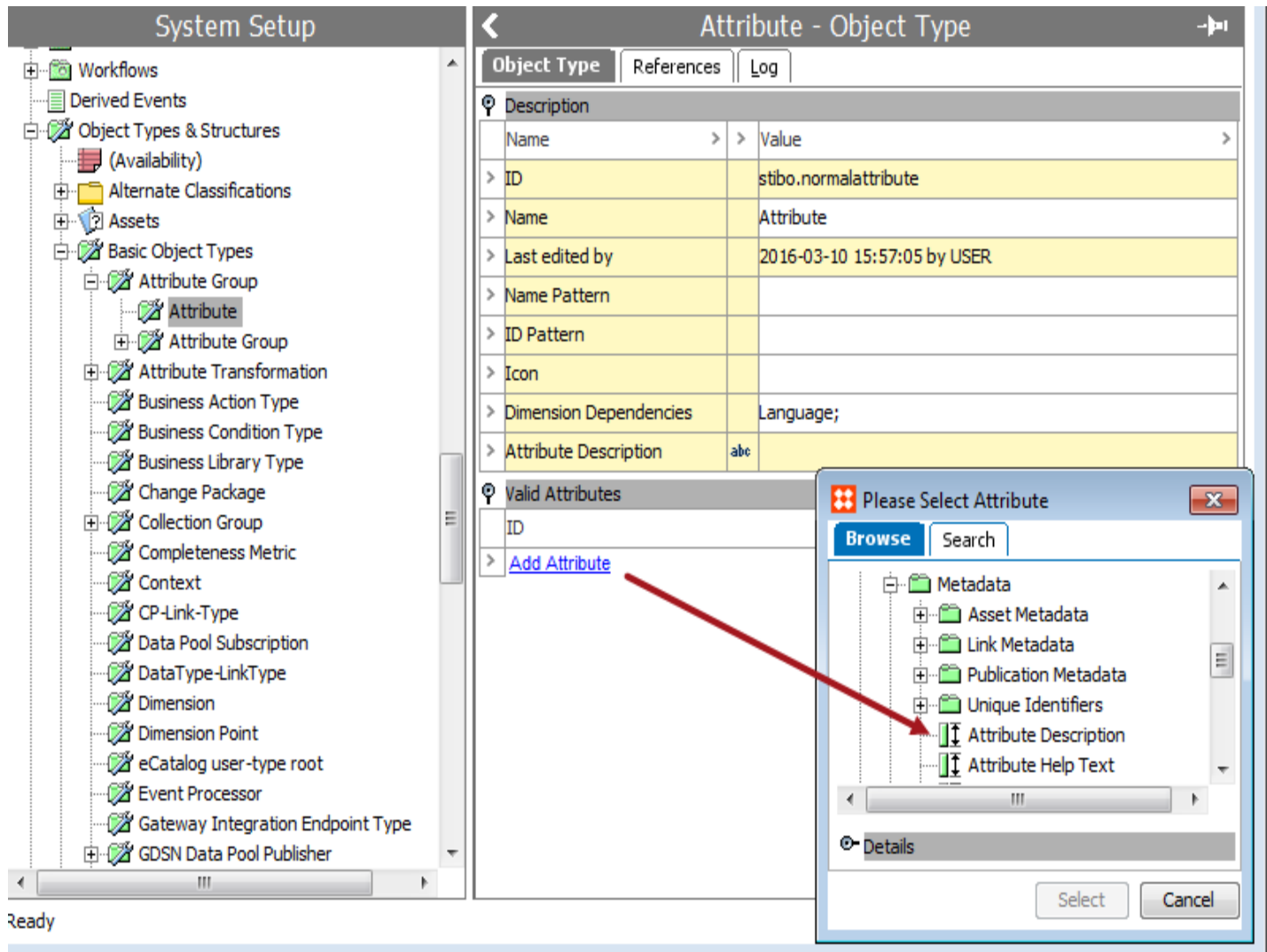
Name	Value
ID	Attribute Description
Name	Attribute Description
Last edited by	2016-03-14 10:22:45 by USER
Full Text Indexable	No
Externally Maintained	Yes
Hierarchical Filtering	None
Calculated	No
Type	Description
Dimension Dependencies	
Mandatory	No

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**.



Note: The dialog shows both description and specification attributes. A message is displayed when a specification attribute is selected. Select option will be grayed out for the specification attribute.

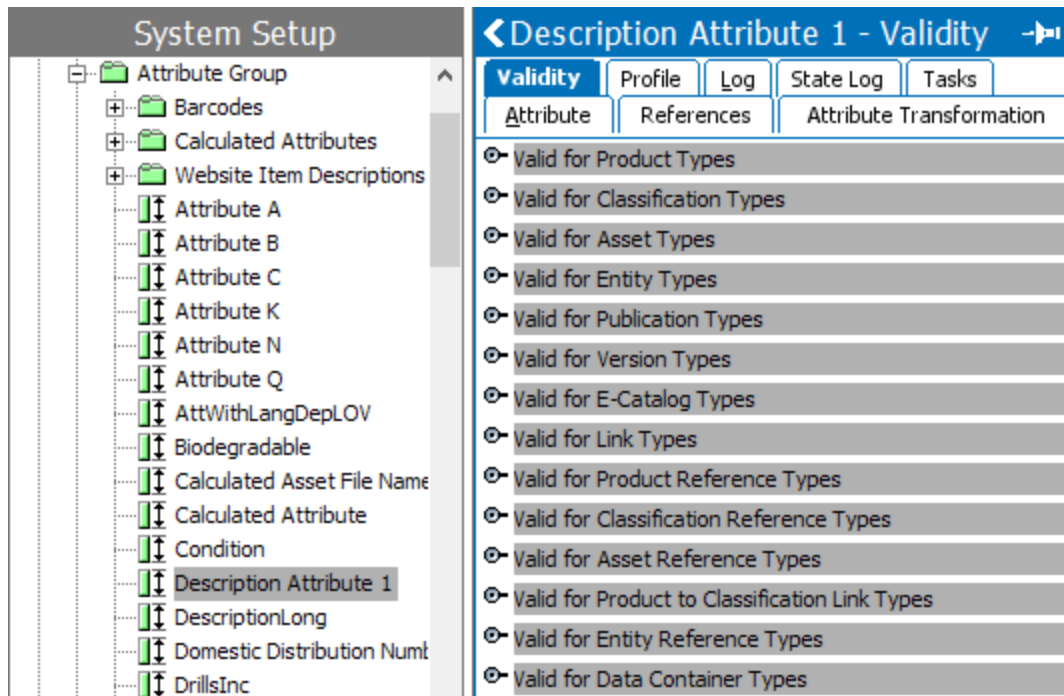
- The metadata attribute is now ready for use. Open the attribute editor for the attribute that requires help text, and add the text there.

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 display 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, under the Validity tab.



Some of the Object Types and linked Attributes are:

Object Type	Attribute
Product Object Types	The Attribute will display automatically for all objects of the specified set of Product Object Types. By selecting the 'All' option the Attribute will be valid for all Product Object Types. It is also possible to select specific Product Object types in the list.
Classification Object Types	The Attribute will display automatically for all Objects of the specified set of Classification Object Types by selecting the 'All' option. It is also possible to select specific Product Object types in the list.
Asset Object Types	The Attribute will display automatically for all Objects of the specified set of Asset Object Types (Images & Documents) by selecting the 'All' option. It is also possible to select specific Product Object types in the list.

For a description of all Object Types, see the **Basic Object types** section and **Object Types** topic in the **Super User / System Setup** documentation.

For more information on how to link description attributes to object types, see the **Description Attributes** topic in the **Super User / System Setup** documentation.

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: Dimension dependency can be applied to the attribute object type—and not just to individual attributes—if you need the attribute names translated. This is a global setting, which means that **all** attribute names should be translated. For more on this see the **Maintaining Dimension Dependent Object Types** topic in the System Setup / Super User documentation. Additionally see the **Dimension Dependent Attribute and Attribute Group Best Practices** topic for more information on how to work with dimension-dependent attribute object types.

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.

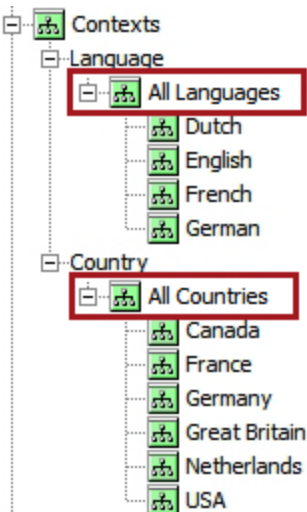
1. The attribute 'Long Description' is dimension dependent on **Language**, meaning that its content may differ based on what language dimension point is used in the context. The typical usage of dimension-dependent data is when content is translated, allowing an attribute to have different values for different languages. For more information on the translation functionality of STEP, see the **Translations** documentation.
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 shows the 'System Setup' interface for two attributes: 'Long Description' and 'Warranty'. On the left, a tree view shows the context hierarchy: Contexts > Language > All Languages (Dutch, English, French, German) and Contexts > Country > All Countries (Canada, France, Germany, Great Britain, Netherlands, USA). Red arrows and circles indicate the selection of 'All Languages' (labeled 1) and 'All Countries' (labeled 2) as the context levels for the respective attributes. On the right, the 'Long Description - Attribute' and 'Warranty - Attribute' configuration panels are shown. In the 'Long Description' panel, the 'Dimension Dependencies' field is set to 'Language;'. In the 'Warranty' panel, the 'Dimension Dependencies' field is set to 'Country;'. Both fields are highlighted with red boxes.

Non-Dimension Dependent Attribute Values

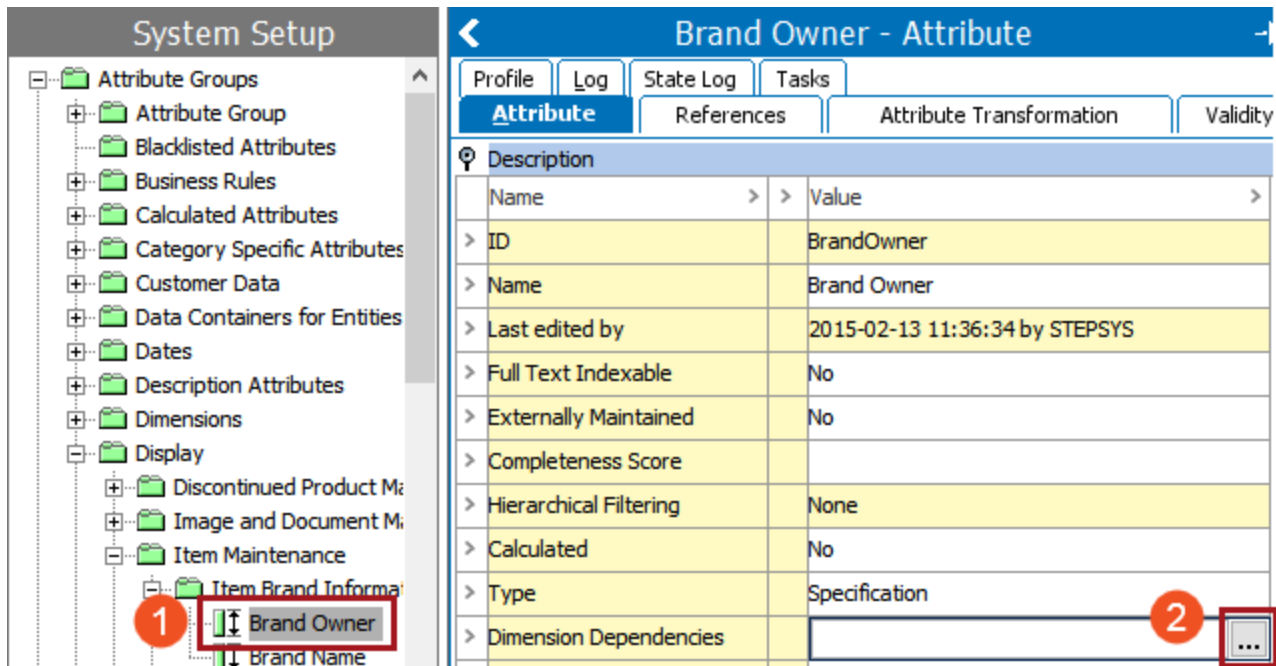
No dimension dependency means that data is stored in the **All** level (also known as the **Root** level) of all dimensions. The following also applies to values contained in STEP **names** of objects.



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 (...).

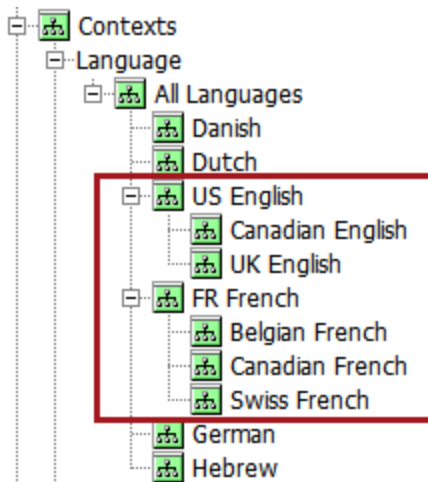


3. In the **Select Dimension Dependencies** dialog box that displays, check the box(es) for the relevant dimension(s), then click **Apply** to set the dimension dependency.

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

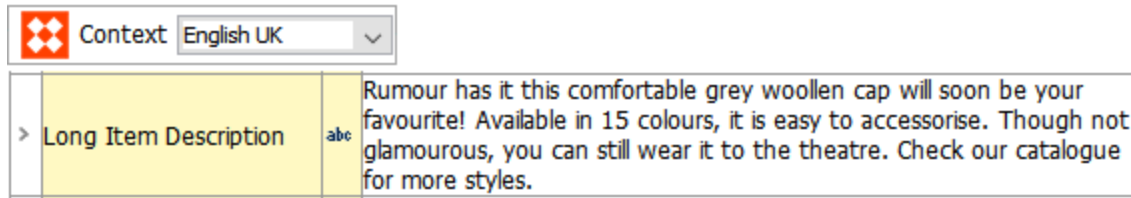
> Long Item Description	abc	Rumor has it this comfortable gray woolen cap will soon be your favorite! Available in 15 colors, it is easy to accessorize. Though not glamorous, you can still wear it to the theater. Check our catalog for more styles.

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	Text, inherited from other context 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.



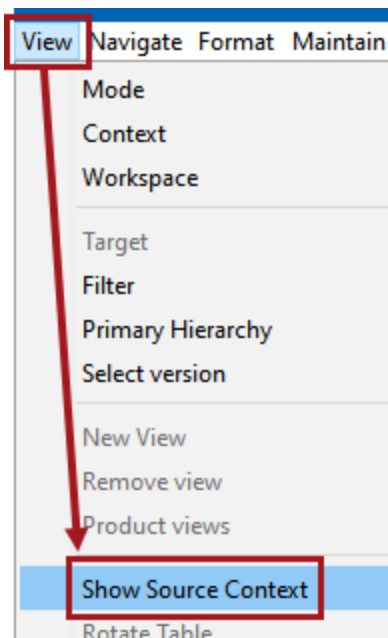
This can save on translation costs because, instead of sending US English content to a translation company to convert it to UK English, the US English content could be exported from STEP, a user could run a find and replace operation on the content to replace a certain subset of words within the content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

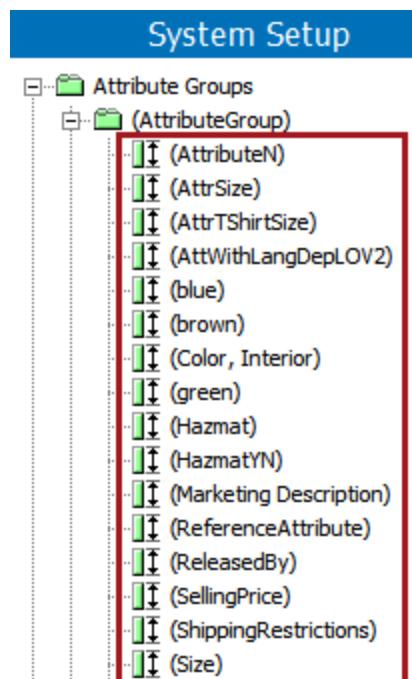
Dimension Dependent Attribute and Attribute Group Best Practices

When setting dimension dependency on attributes and attribute groups, certain considerations should be kept in mind. This topic explains several 'do's' and 'don'ts' when working with dimension dependent attributes and attribute groups.

Dimension Dependent Attribute Do's

1. Be consistent about which context you are in when you create the attributes.

The names of the attribute will not be visible in other contexts until they are translated and will instead display the attribute ID in parentheses, as pictured below.



2. Carefully consider dimension points when defining attributes.

Even if you are only publishing in one language or country today, it is much easier to remove a dimension dependency from an attribute than it is to add one later.

Dimension points should be applied to attributes such as description attributes, feature attributes, and benefit attributes, which truly contain data that could become dimension-dependent in the future. This may seem like an extra task in the beginning, but if business conditions change, it is much easier to remove dimension dependencies from attributes than it is to add a dimension dependency to attributes that already have values.

Note: Remember that if an attribute does not have a dimension dependency assigned, all data values are stored in the 'All' level.

For the step-by-step process for moving data from an All level to a language level, see the **Adding a Dimension Dependency After Loading Data** topic.

Dimension Dependent Attribute Don'ts

1. Do not create attributes with multiple dimension dependencies.

Only *language* is supported in translation tracking—no other tools exist in STEP to identify changes in other dimensions. For more information on the translation functionality of STEP, see the **Translations** documentation.

2. Do not make attribute links to categories dimension dependent.

Dimension-dependent attribute links can be handy but be aware that they are also hard to maintain. If for instance, if you want to create / modify these attribute links via STEPXML imports and some links should be visible in more than one context, you are forced to create a 'cross-context' import file with qualifiers, as you cannot make the links visible in one context at a time.

Which Attributes Need Dimension Dependencies?

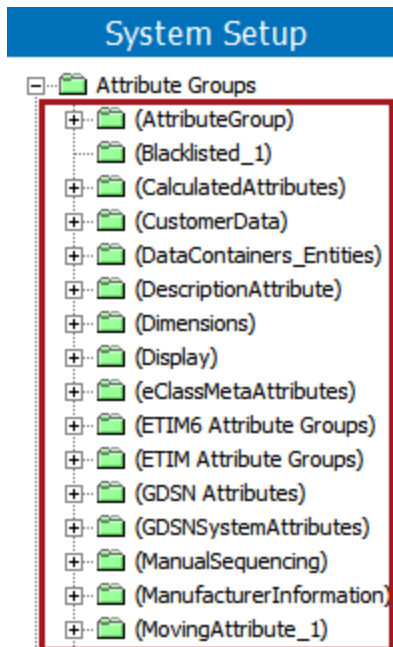
Questions to ask	Recommendation
Would the data be presented differently if it were published in another language (for example, long descriptions or feature bullets)?	If so, make the attribute language dependent, even if you only have one language today.
Would the data appear exactly the same if it were published in another language (for example, Y/N, T/F, and numeric values)?	If so, do not add a dependency
Would the data be presented differently if the product were sold in another country (for example, warranty)?	If so, make the attribute country dependent. If not, do not add a dependency.
If the attribute is owned by another system, does that system have the ability to store multiple language or country-dependent values?	If not, do not add dependencies.

Dimension Dependent Attribute Group Do's

Attribute **groups** should have a dimension dependency if you would like to see the attribute group **names** in their own language.

Note: Making attribute groups dimension dependent is a **global** setting and does not affect attributes within the group.

1. If you add a language dependency to attribute groups, be consistent about which context you are in when you create the groups. The names of the groups will not be visible in other contexts until they are translated.



2. Come up with a standard process for translating attribute groups and attributes. Always start in the same language and have a consistent process.

Dimension Dependent Attribute Group Don'ts

1. Do not make attribute groups dependent on anything other than language.
2. If there is no requirement to add a dimension dependency, do not add one.

Using Dimension Dependent LOVs With Attributes

Lists of Values (LOVs) can be set up to be dimension dependent, just as attributes. The most common scenario when using dimension-dependent LOVs is to use them along with *non*-dimension-dependent attributes, though other combinations are also applicable.

This topic explains four combinations of dimension dependency between attributes and LOVs, best practices in their setup, and use cases:

1. Attribute with no dimension dependency used with a dimension-dependent LOV
2. Attribute with no dimension dependency used with an LOV without a dependency
3. Attribute with a dimension dependency used with a dimension-dependent LOV
4. Attribute with a dimension dependency used with an LOV without a dependency

This topic does not explain how to create an LOV or how to make an LOV or attribute dimension dependent. For more information, see the **Creating an LOV** topic and the **Dimension Dependent Attributes** topic.

Note: It is important to set up dimension-dependent LOVs from the start to avoid issues that could arise from setting a dimension dependency later after values have already been added to the attributes that use them. For more information, see the **Adding a Dimension Dependency After Loading Data** topic in the **Dimensions, Dimension Points, and Contexts** documentation.

To illustrate the differences between the setups, the screenshots in this topic show attribute values viewed in the **Context** mode in the STEP Workbench

Attribute With No Dependency, LOV With a Dependency

The most typical scenario when using a dimension-dependent LOV is to use it along with an attribute that is **not** dimension dependent. This is used for descriptive information that is true for every language, where the value needs to appear in the local language.

This is a cost-saving benefit of 'pre-translation' when new products use the LOV. Customers who use a lot of LOVs for this purpose can save a lot of money.

	> English US	> Danish DK	> French FR	>
> ID	109011	109011	109011	
> Name	109011	20803-03	109011	
> Attribute not language dependent, LOV is	Red	Rød	Rouge	

When using an attribute without a dimension dependency and an LOV that has a language dependency, a user working in English can choose the color 'Red' for a new product. If the LOV has already been translated, the French value 'Rouge' will automatically show up, so there is no reason to retranslate this text. Every time Red is chosen, the translation is already in place.

Attribute With No Dependency, LOV Without a Dependency

Another commonly used setup is when neither the attribute nor the LOV it is using are dimension dependent. The value of the LOV is the same in every context

	> English US >	Danish DK >	French FR >
> ID	109011	109011	109011
> Name	109011	20803-03	109011
> Attribute not language dependent, LOV is not	Y	Y	Y

The typical use case for this scenario is for numbers or dates or even text that should never change, no matter which language or country is consuming the information. Many values like Y/N flags, availability dates, product weights, dimensions, and measures would not have any dimension dependencies.

Attribute With a Dependency, LOV With a Dependency

This setup is *not* commonly used because values in all dimension points are independent of each other due to there being separate values in each dropdown list. In this example, the value is completely different in English US, Danish DK, and French FR. 'Red' appears in English US, but the Danish word for 'Blue' and the French word for 'Black' appear in the Danish DK and French FR contexts.

	> English US >	Danish DK >	French FR >
> ID	109011	109011	109011
> Name	109011	109011	109011
> Attribute language dependent and LOV is too	Red	Blå	Noir

If an English US user chooses 'Red,' a value will not display in the French FR context until a French FR user chooses something from the dropdown, whereas if the attribute does *not* have a dimension dependency and the LOV *does* (per the first example in this topic), if a user chooses 'Red' for English US, then 'Rød' and 'Rouge' will automatically populate in the Danish DK and French FR contexts.

Attribute With a Dependency, LOV Without a Dependency

It is a rare setup to use an attribute with a dimension dependency along with an LOV that does *not* have a dimension dependency because the values can be different in every dimension. In this scenario, everyone uses the same drop down list—English US values mixed in with Danish DK and French FR values.

	> English US >	Danish DK >	French FR >
> ID	109011	109011	109011
> Name	109011	109011	109011
> Attribute language dependent, LOV is not	Y	N	Y

Although not typical, one use case for this might be an 'Available for Sale' attribute using a Y/N global LOV but with a country dependency on the attribute.

In this scenario, when an English US user chooses a value, nothing will appear in the Danish DK or French FR contexts until a user in one of those countries makes a choice from the same dropdown list.

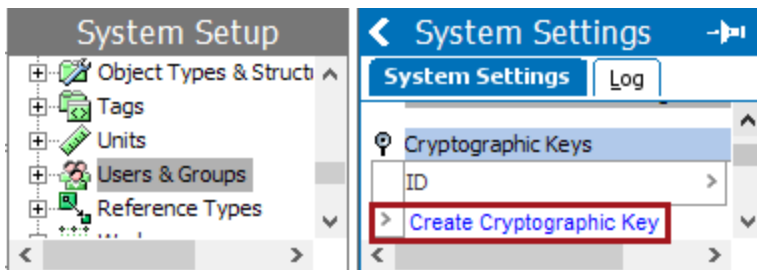
Encrypted Attributes

When dealing with customer data, some of the data is sensitive and needs extra protection. Encrypting an attribute is a way of making its values inaccessible to STEP users who are not allowed to view the values as well as obscuring the data for enhanced security. An attribute should only be encrypted when necessary since it prevents useful functionality such as searching, data profiling, exporting, and more.

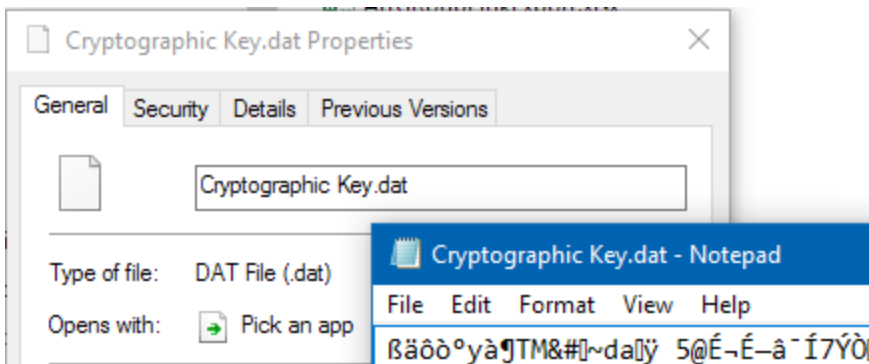
Encrypted attributes require a password to view or edit the values of the attribute. Before creating an encrypted attribute, create a cryptographic key.

Cryptographic Key

To create a cryptographic key, go to System Setup > Users and Groups > Cryptographic Keys flipper > click **Create Cryptographic Key**.



The attribute encryption system implemented in STEP is based on AES 256 bit encryption. It is a symmetric key algorithm which means that the same key is used for encryption and decryption. The system can hold multiple keys, and each key can have one unique password. This password then will be shared between users. For instance, the members of a specific group could be the only ones that should have access to the values of attributes Attr_A and Attr_B. In that case, only one key would be used for encryption of the values of Attr_A and Attr_B, and the password of the key would be shared among the users of the group.



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 will remain secure. In addition to the key file, an ID and a password must be supplied.

Note: Cryptographic keys are stored in STEP protected by a password that is neither the original key, nor the original password.

Create Cryptographic Key

ID:

Password: ✓

Retype Password: ✓

Select Key File

Look in:

Recent Items

Desktop

Documents

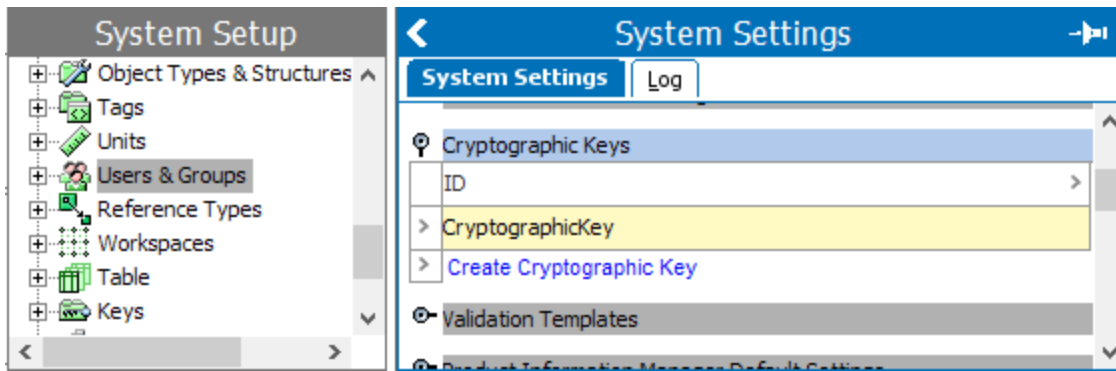
This PC

Network

☐ Cryptographic Key.dat

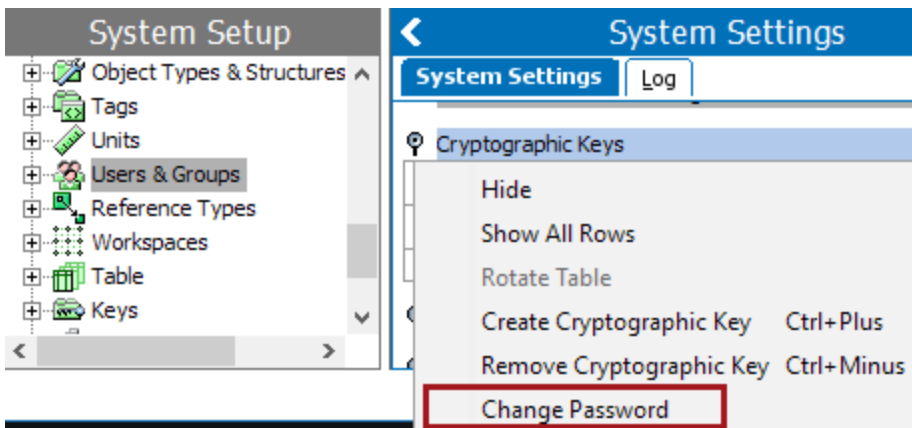
File name:

Files of type:



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. To do so, follow the steps below:

1. Go to System Setup and expand the attribute groups node.
2. Select and right click on the desired attribute group under which the new attribute is to be created, and click on 'New Attribute'.
3. On step 2, Enter ID and Name, there is an option called 'Cryptographic Key' with a dropdown list. Select the appropriate Cryptographic Key and proceed.

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: 112103

Name: Attribute N

Validation Base Type: Text

Multi Valued: No

☒ Specification
☐ Description

Cryptographic Key: <No Encryption>

Back Next Finish Cancel

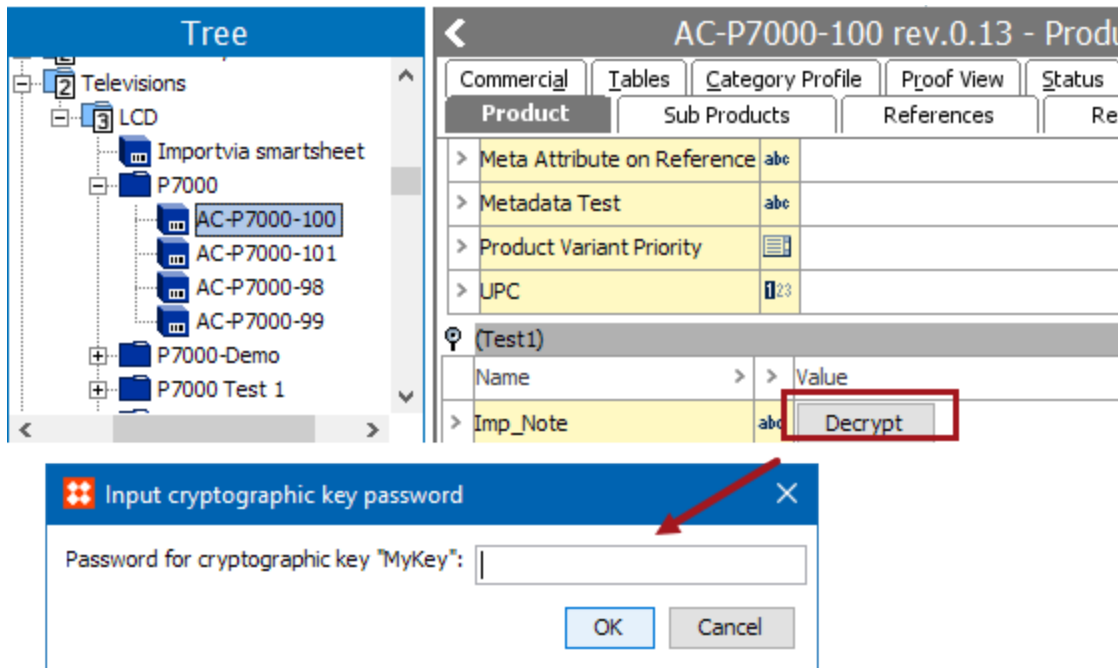
Note: The cryptographic key cannot be changed or set after creation.

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.

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 display multiple times when a Web UI screen is 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 in Web UI. A blacklisted attribute group can be used to hide this sensitive information.

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

If adding the group 'Blacklisted' to the parameter Blacklisted Attribute Group 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 'Blacklisted' is added to the parameter Blacklisted Attribute Group for both group 'X' and group 'Y', then neither group will show attribute 'A'. Alternatively, if nothing is chosen for the parameter Blacklisted Attribute Group for both group 'X' and group 'Y', then both groups will show attribute 'A'.

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

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

To avoid confusion, it is recommended that any attribute groups created specifically to group blacklisted attributes be named accordingly.

More information about attributes and creating attribute groups can be found in the **Attributes** section of the **System Setup** documentation.

Setting up Blacklisted Attribute Groups in Web UI

Within the designer, the Attribute Value Group component has a Blacklisted Attribute Group field which can be populated with an attribute group. The blacklisted attribute group indicates attributes that should not be displayed within the Web UI. Any attributes in the blacklisted attribute group will not display on the screen. This feature can be used when displaying multiple attribute groups on the screen or even when using a single group.

Note: It is not possible to add multiple attribute group to 'Blacklisted Attribute Group' field in Web UI. It only allows one attribute group to be selected.

1. Log in to the Web UI and go into design mode.
2. On a Node Editor (that is ideally on a screen designed to house object information) click 'Go to Component.'

3. Once there, click on the 'Add' button to add a Child component on the Node Editor Properties screen.
4. On the Add component screen, select the Attribute Value Group Component and then click 'Add.'
5. The 'Attribute Values Group Component Properties' will display. Select the attribute group and the 'blacklisted attribute group' as well as the remaining option, then click on 'Add.' For more information the Attribute Value Group Component Properties topic in the Web UI Setup and User Guide / Web User Interface documentation.

Add component - configure required properties

Attribute Group*

Blacklisted Attribute Group

Context Help

Display Mode
<Select a value>

Enable STEP Tags
☐

Enable Tag Conversion
☒

Exclude Filters

Note: Selecting a blacklisted attribute group in Attribute Value Group Component Properties has no impact on other attribute groups shown on a screen within Web UI. Each component operates independently of each other.

Maintaining Attributes

Once an attribute is created, there are changes done to that attribute based on user requirements. Such requirements could entail attribute validation, changing the attribute types, and many more.

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

When an attribute is created, it can be set to either a description attribute or a specification attribute based on the requirements. Similarly, changes can be done on existing attributes. For example, a certain description attribute is used to describe the product in detail. A user may now want to change the existing attribute from a description attribute, where the value entered is not for publishing, to a specification attribute, and publish the value entered for that attribute through different means.

For more on attribute types, see the **Description Attributes** topic and the **Specification Attributes** topic in the **System Setup / Super User Guide** documentation.

To set the attribute type, follow the directions below:

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

System Setup

Attribute Groups

Attribute Group

(green)

Attribute D

Attribute N

AttrSize

AttrTShirtSize

blue

brown

Color, Interior

Condition

Hazmat

Hazmat Y N

Marketing Description

Reference Attribute

Released by

Attribute N - Attribute

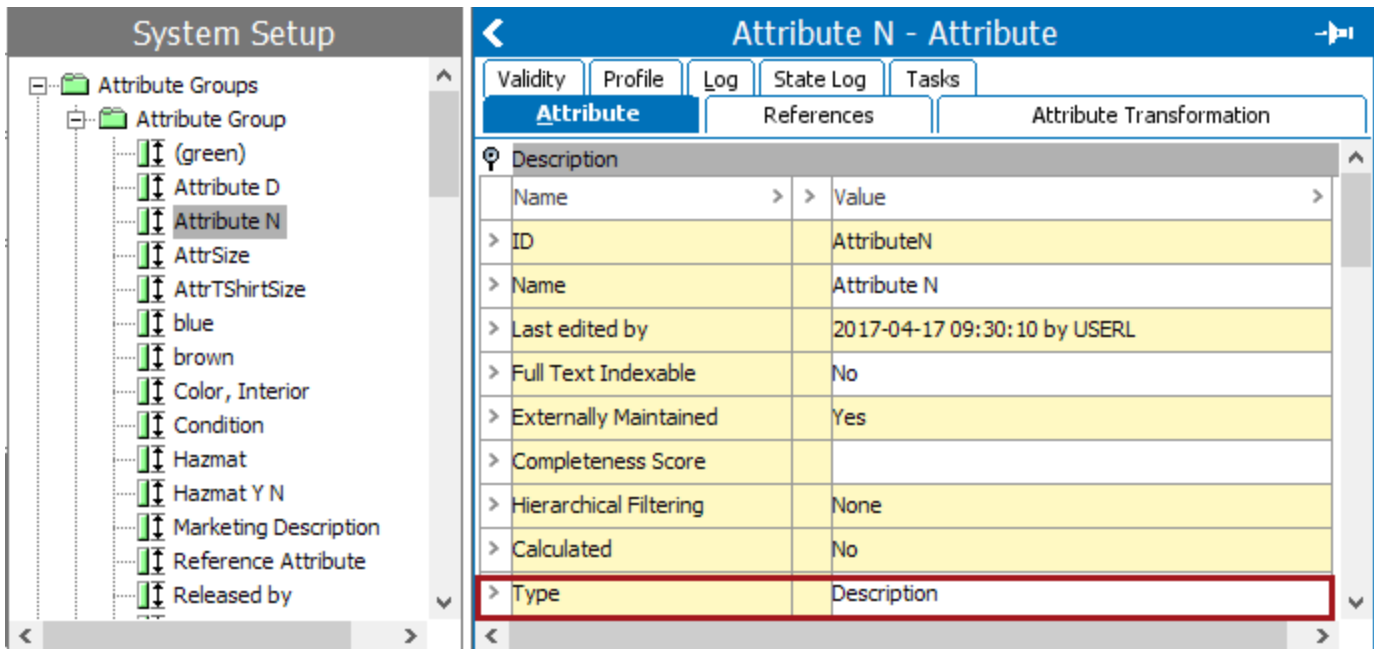
ValidityProfileLogState LogTasks

AttributeReferencesAttribute Transformation

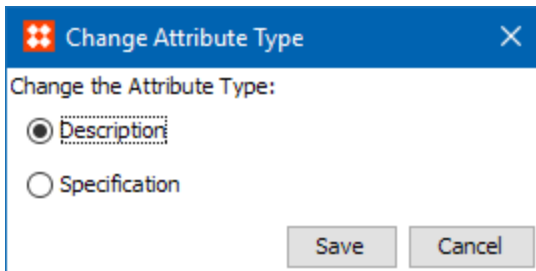
Description

Name	Value
ID	AttributeN
Name	Attribute N
Last edited by	2017-04-17 09:30:10 by USERL
Full Text Indexable	No
Externally Maintained	Yes
Completeness Score	
Hierarchical Filtering	None
Calculated	No
Type	Description

- 2. Under the **Description** flipper, double-click the **Type** field.



The **Change Attribute Type** dialog box appears.



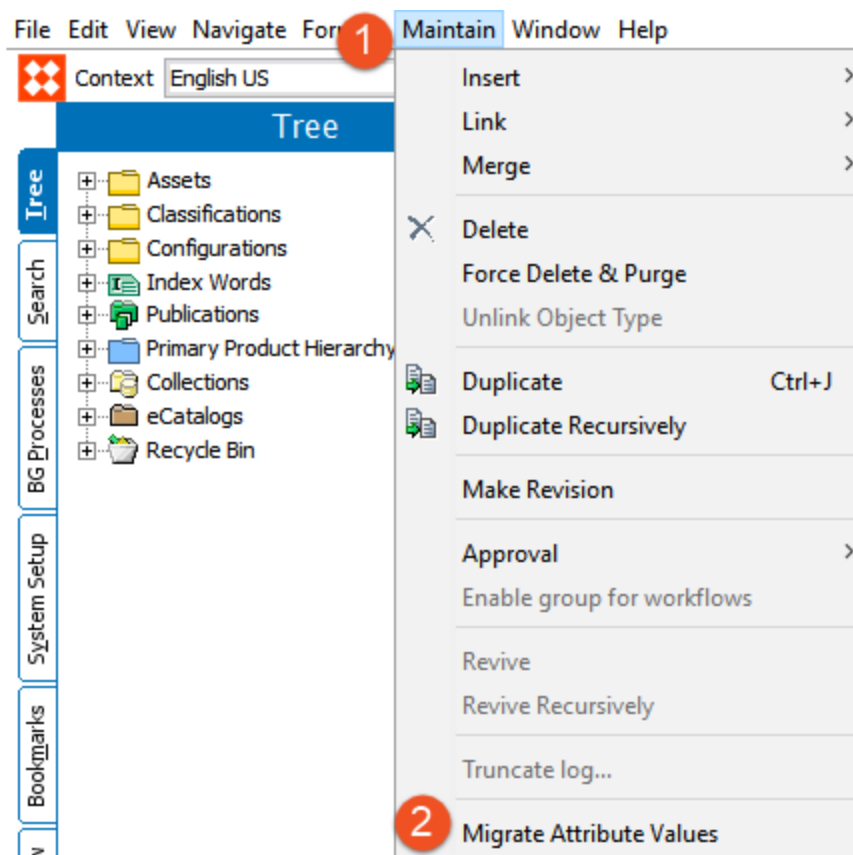
3. Select the relevant type, and click **Save**.

Note: When you are changing from a description type attribute to a specification type attribute, ensure that you link this attribute to the respective hierarchy. For more information, see the **Linking Specification Attributes** topic in the **System Setup / Super User Guide** documentation.

Attribute Value Migration

With Trailblazer 8.1, a new storage model was added for non-LOV attribute values called Compact Value Storage (CVS). The benefits of this storage model are that less space is required for storage and that the reading and writing of values can be performed faster. The CVS model is automatically used for attributes created in STEP Trailblazer 8.1 and newer systems.

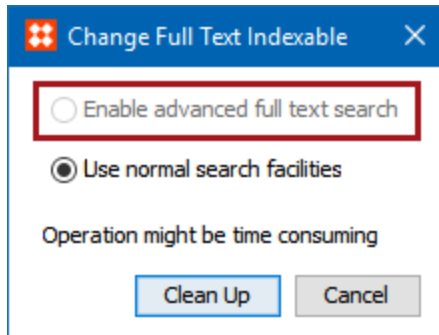
For those users who want to take advantage of CVS and need to migrate attribute values created on pre-8.1 STEP systems, there is a **Migrate Attribute Values** option in the **Maintain** menu of the workbench. See the **Technical Migration Details** section below for a detailed explanation of what the migration process entails.



Prerequisites

The attribute value migration should be configured and run by an admin or super user who has knowledge about the system utilization. The user configuring and running the migration must have the setup action privilege 'Modify definition of product attribute (domain and default unit)' for all attributes on the system. For more information on this privilege, see the **Attributes** section of the **Setup Actions** documentation.

- Your STEP system must have at least 5 percent free space in the 'STEPSYSDATA' tablespace in order to start the migration. A warning is displayed if there is less than 10 percent free space.
- Tests have shown that having the full text index ('value_ctxix') being enabled has a negative impact on the migration performance. If the full text index functionality is not critical to your business functions, it is recommended to drop the index before starting the migration.
- It is possible, via the workbench, to tell if the full text index is enabled. To do this, navigate to an attribute in the System Setup and double-click the “Full text indexable” field. If the “Enable advanced full text search” option is grayed out in the dialog that appears, the index is already disabled.



Important: If possible, it is advisable to perform the migration, at least partially, on a test system with the same hardware setup to validate the settings before starting the migration on a production system.

Technical Migration Details

The migration functionality will delete rows from the 'valuemap', 'valuemodel' and 'value7' database tables and will insert rows in the 'softvalue' table. Bulk deleting rows from Oracle tables, however, cause at least two issues:

- As data is deleted, blocks and segments will become sparsely populated. This may lead to Oracle selecting inefficient execution plans again causing an observable performance decrease.
- Tablespace is not released for 'reuse' automatically. This means that if nothing is done to prevent it, the migration will require as much additional tablespace as is required for the 'softvalue' table inserts.

To counter these issues, the migration functionality will not only delete and insert rows, but it will also compact tables and indexes during migration. Four different types of compaction are in play:

1. 'valuemap', 'valuemodel' and 'value7' index compaction – This type of compaction can be performed while the migration is running and does not require any restrictions to be imposed on the system usage.
2. Table compaction of 'valuemap' and 'valuemodel' – This type of compaction cannot be performed while the 'database mode importer' is running and therefore, when configuring the migration schedule, it is possible to specify windows where the database mode importer can be disabled. The 'database mode importer' is only available to super users with all data node dependent privileges. It allows import processes to batch insert unrevised data directly in the database and may provide performance benefits for very specific use cases. The database mode importer is not available on systems running In-Memory.

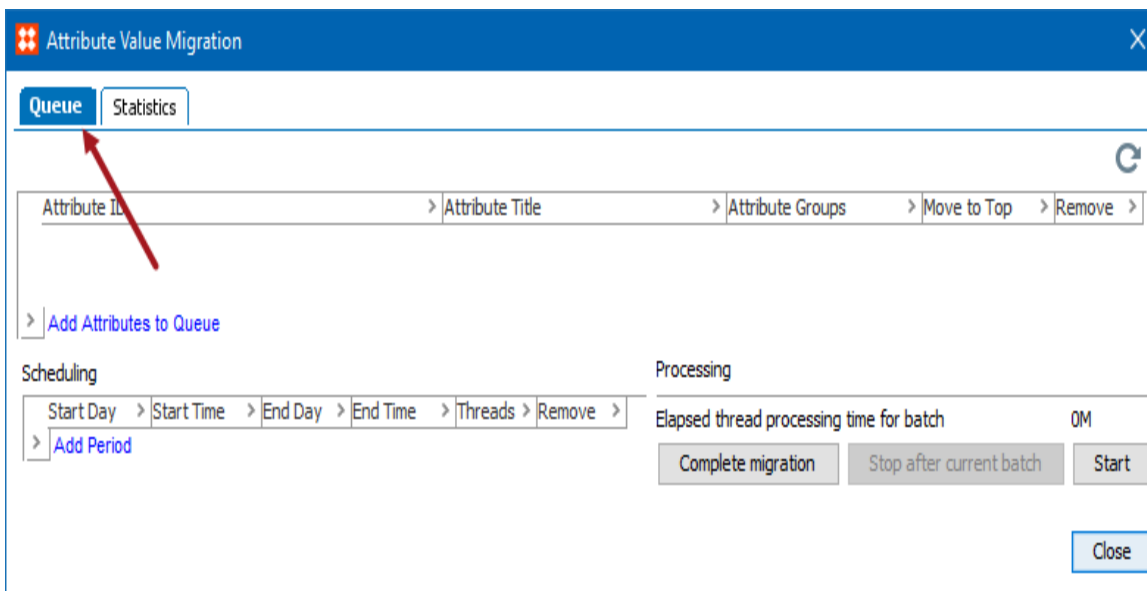
3. Shrinking 'valuemap' and 'valuemodel' – This operation requires a short-lived Single-Update Mode (SUM) lock. Therefore, when configuring the migration schedule, it is possible to specify windows where a SUM lock can be taken. See the **Single-Update Mode** section of the System Setup documentation.
4. Shrinking 'value7' – Due to the 'value7' table having a function-based index, the shrink space approach used for 'valuemap' and 'valuemodel' cannot be used here while STEP is running. Thus, when all values have been migrated, it is possible to schedule the required 'value7' operations for next system restart. What will happen here is that the function based index will be dropped, the table shrunk, and the index rebuilt. Refer to the next section, **Completing Attribute Value Migration**, for more information.

Configuring Attribute Value Migration

To start the value migration, add the desired attributes to be migrated to the queue, and then create a migration schedule.

Queuing Attribute Values for Migration

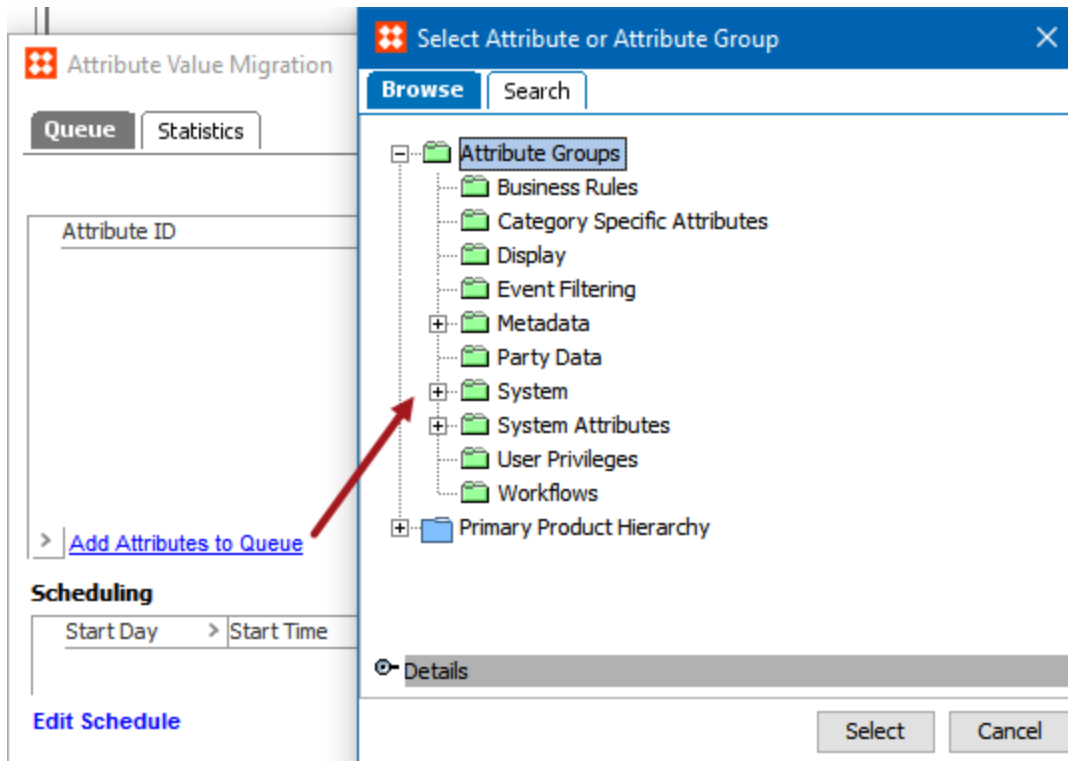
1. After selecting **Migrate Attribute Values** from the **Maintain** menu, the 'Attribute Value Migration' dialog will appear. Ensure the Queue tab is selected from this dialog.



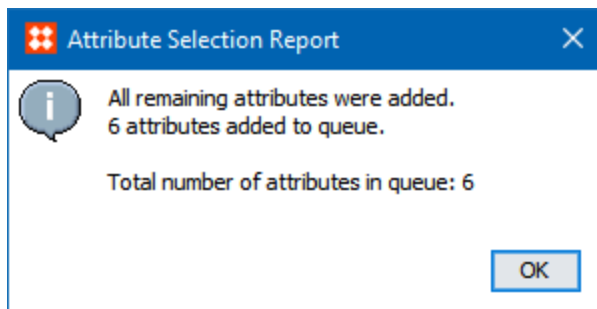
The screenshot shows the 'Attribute Value Migration' dialog box with the 'Queue' tab selected. A red arrow points to the 'Queue' tab. The dialog has a table with columns: Attribute ID, Attribute Title, Attribute Groups, Move to Top, and Remove. Below the table is a link 'Add Attributes to Queue'. There are two sections: 'Scheduling' and 'Processing'. The 'Scheduling' section has a table with columns: Start Day, Start Time, End Day, End Time, Threads, and Remove. Below it is a link 'Add Period'. The 'Processing' section has a label 'Elapsed thread processing time for batch' with a value '0M' and three buttons: 'Complete migration', 'Stop after current batch', and 'Start'. At the bottom right is a 'Close' button.

2. Click the **Add Attributes to Queue** link. A list of the migratable attributes will appear. Select **OK** after choosing the desired attributes to add to the queue.

Note: If an attribute group is selected, all attributes belonging to that group will be added to the queue. In this example, 'Attribute Groups' is selected, therefore, all the groups below it, as well as all the attributes under each attribute group, will be included.



A report of the number of added attributes will appear.



The attributes will be added to the queue.

Attribute Value Migration

Queue

Statistics

Attribute ID	Attribute Title	Attribute Groups	Move to Top	Remove
> UserInformation	User Information	Metadata	^	✕
> ProductPageNumber	Page Numbers	System Attributes	^	✕
> UnitDescription	Unit Description	Metadata	^	✕
> DisplaySequence	Display Sequence	Metadata	^	✕
> Purpose	Purpose	Metadata	^	✕
> AttributeHelpText	Attribute Help Text	Metadata	^	✕

> Add Attributes to Queue

As described above, it is possible to only queue a subset of the migratable attributes. It is also possible to remove attributes from the queue. The reasons for having these options are:

1. The CVS model requires different SQL API views to be used for querying values meaning that for customers using the SQL API it must be possible to postpone the migration of values for certain attributes until the SQL has been updated. For details, see the SQL API documentation by clicking the STEP API Documentation button on the STEP WebStart page.
2. It is not possible to perform schema maintenance actions like changing the validation base type or attribute dimension dependencies while values for an attribute are being migrated.

Scheduling the Attribute Value Migration

Before scheduling the migration and following the next steps, review the **Scheduling Considerations** section below.

1. From the scheduling section of the 'Attribute Value Migration' dialog, select **Edit Schedule**. The 'Edit Schedule' dialog will appear.

Scheduling

Start Day >	Start Time >	End Day >	End Time >	Threads >	Allow Disabl... >	Allow SUM >

[Edit Schedule](#)


Edit Schedule

Scheduling

Start Day >	Start Time >	End Day >	End Time >	Threads >	Allow Disabl... >	Allow SUM >	Remove >

> Add Period

OK

Cancel

2. Select **Add Period**, and a default time will be added. Configure by clicking on each option and selecting from the dropdown menu.


Edit Schedule

Scheduling

Start Day >	Start Time >	End Day >	End Time >	Threads >	Allow Disable DB Mode Imports >	Allow SUM >	Remove >
> Monday	00:00	Monday	00:00	1	Yes	No	X

> Add Period

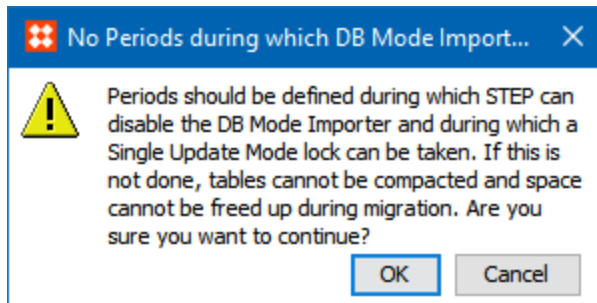
OK

Cancel

- **Start Day** - The day the scheduling period starts
- **Start Time** - The time on the starting day that the period starts
- **End Day** - The day the scheduling period ends
- **End Time** - The time on the ending day that the period ends
- **Threads** - How many threads the migration process may use during the period
- **Allow Disable DB Mode Imports** - Whether or not the process is allowed to disable the database mode importer within the period / window. In most cases, it is safe to set this value to 'Yes.' With In-Memory systems, it should always be set to yes since the database mode importer is disabled.
- **Allow SUM** - Whether or not the process is allowed to take a Single-Update Mode lock within the period / window. For more information about Single-Update Mode, see the **Single-Update Mode** topic.
- **Remove** - Selecting the X will delete the selected period

Scheduling Considerations

- The schedule uses server times.
- If the End Day and End Time are identical to the Start Day and Start Time, then the period will be a full week.
- If overlapping schedule periods are defined, the period with the fewest restrictions and the highest number of threads will be used.
- A warning will be shown if no periods are defined where the process is allowed to disable the database mode importer and/or take a SUM lock. It is not possible to complete the migration without having defined these schedule periods.



- The migration process will only run on a single application server in a clustered setup and this may be taken into account when specifying the number of threads for a processing period. How many threads to use depends on the system hardware and the system utilization. Typically one or two threads will be reasonable for periods where the system is heavily used while 16 threads is reasonable during off-peak hours. As mentioned earlier in this topic, it is advisable (if possible) to perform the migration, at least partially, on a test system with the same hardware setup to validate the settings before starting the migration on a production system.

Starting the Attribute Value Migration

1. After configuring your migration schedule, select **Start**.

Attribute Value Migration

Queue

Statistics

Attribute ID	Attribute Title	Attribute Groups	Move to Top	Remove
> UserInformation	User Information	Metadata	^	X
> ProductPageNumber	Page Numbers	System Attributes	^	X
> UnitDescription	Unit Description	Metadata	^	X
> DisplaySequence	Display Sequence	Metadata	^	X
> Purpose	Purpose	Metadata	^	X
> AttributeHelpText	Attribute Help Text	Metadata	^	X

> Add Attributes to Queue

Scheduling

Start Day	Start Time	End Day	End Time	Threads	Allow Disabl...	Allow SUM
> Tuesday	09:00	Tuesday	11:00	1	Yes	Yes

Edit Schedule

Processing Info

Status: Inactive

Queue attributes, configure a processing schedule and press "Start" to start migrating values.

Finalize migration

Stop after current batch

Start

Close

Your migration is now scheduled, and the process will begin at the earliest set period.

Scheduling

Start Day	Start Time	End Day	End Time	Threads	Allow Disable DB Mode Imports	Allow SUM
> Tuesday	09:00	Tuesday	11:00	1	Yes	Yes

Processing Info

Status: Waiting

Processing to start again Tuesday at 9:00

Finalize migration

Stop after current batch

Stop now

Statistics

The Statistics tab shows details about the migration. The following image shows how the Statistics tab looks before migration has begun.

Attribute Value Migration

Queue

Statistics

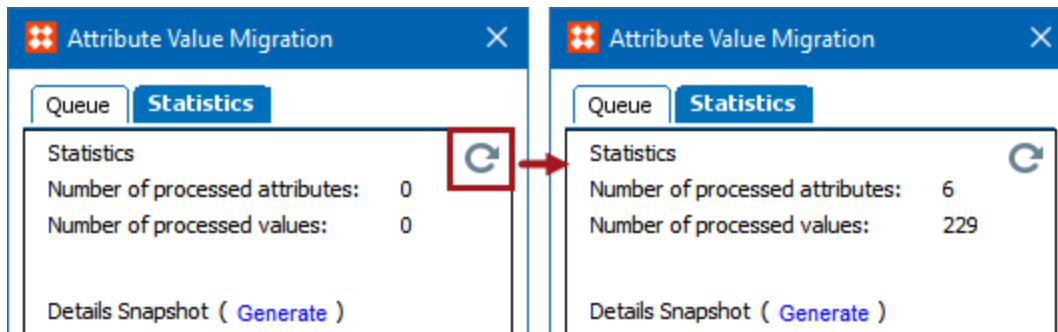
Statistics

Number of processed attributes: 0

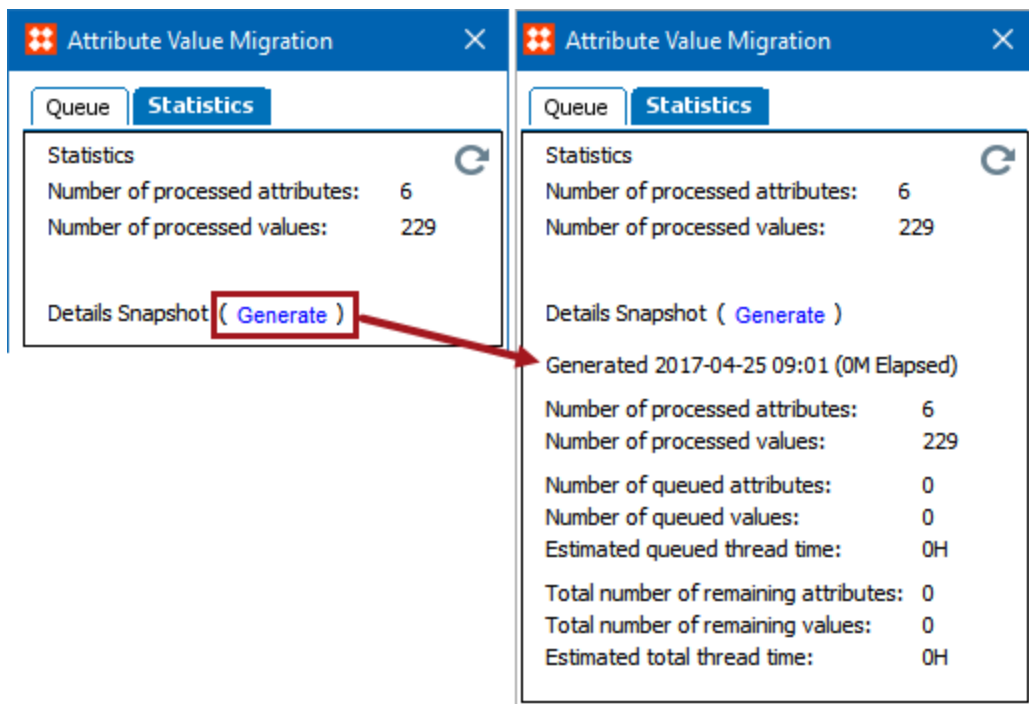
Number of processed values: 0

Details Snapshot (Generate)

- **Number of processed attributes** - The number of attributes that have been migrated
- **Number of processed values** - The number of values that have been migrated
- **Refresh icon** - Refreshes the GUI, allowing for data updated values to display



- **Details Snapshot** - Selecting the 'Generate' link will create a snapshot report. This report has information about attributes / values that have been processed as well as queued attributes / values to be processed. Finally, it will also display the estimates for the remaining processing time.



In the generated report, the 'Estimated queued thread time' and 'Estimated total thread time' are the estimated times to process all queued attributes and all migratable attributes, respectively. This does not take into account time spent doing other tasks such as the waiting period for the SUM lock.

Note: The estimates are based on processing using a single thread. Ideally, if the processing is run with two threads, the duration will be half of the estimate, and with four threads, it would be a quarter of the processing time, and so forth.

Completing the Migration

After the values for all migratable attributes on the system have been migrated, the final steps for the migration are ready to be initiated. This status can be noted on the 'Processing Info' section of the 'Attribute Value Migration' dialog.

When the button is pressed, a dialog will appear from where it is possible to schedule the “value7” operations for the next STEP system restart. To determine how long the offline operations will take, it is advisable to perform the migration on a test system (with similar data and hardware setup) prior to doing it on a production system.

If the full text index functionality is enabled on the system on which the migration is being performed, this index “value_ctxix” will be dropped and recreated during the system restart. The index will be populated asynchronously once the system has been started again and until this has happened users may experience that they get fewer search results for full text index searches.

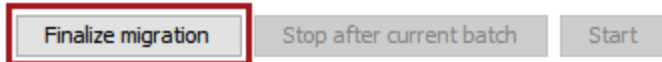
Important: The final processes of the migration require that the STEP system is restarted.

1. Click **Finalize migration** to generate a prompt noting that the migration is ready to be completed.

Processing Info

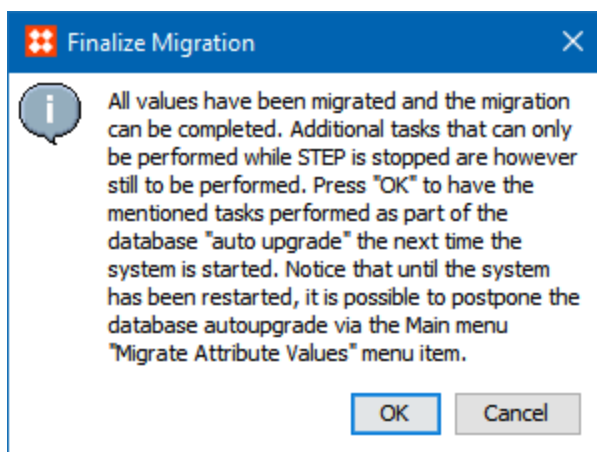
Status: Inactive

All attributes have been migrated. Press "Finalize migration" to complete the final migration steps.

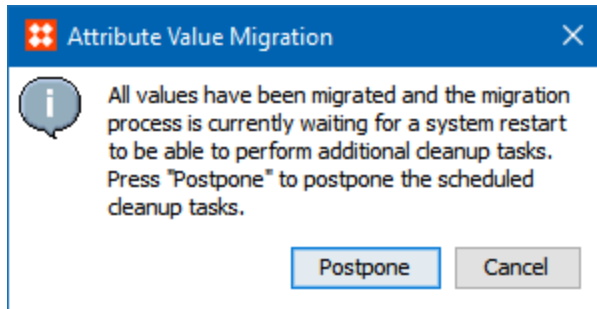


2. Press OK to schedule the final operations of the migration for the next system restart.

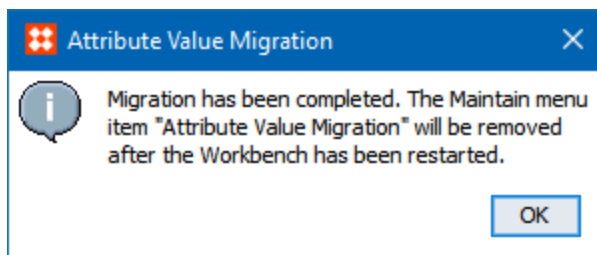
Important: The system may be offline for an extended period of time while the processes finish migrating. Please plan for a period of downtime before restarting STEP.



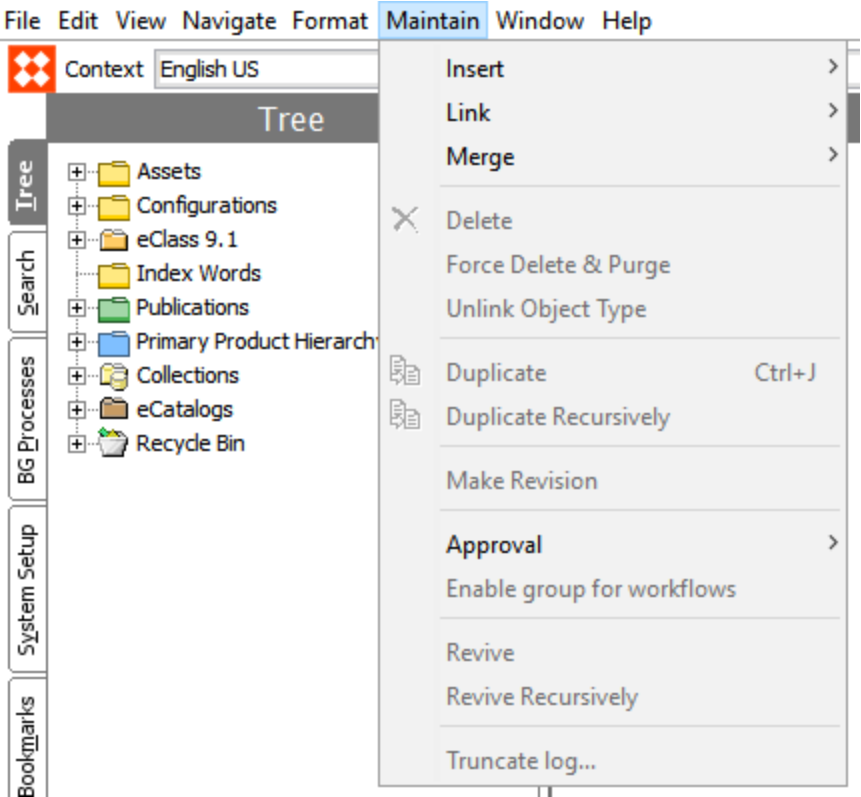
Should you need to postpone completing the migration for any reason, navigate to the **Migrate Attribute Values** option under the **Maintain** menu item. A dialog box will offer the option to postpone the final steps of this migration. Selecting this option will require you to finalize the migration again from the **Attribute Value Migration** tool.



3. Restart the STEP system, and then navigate once more to the **Migrate Attribute Values** option under the **Maintain** menu item to confirm that the migration was a success. Select the **Migrate Attribute Values** option to display a successful migration prompt.



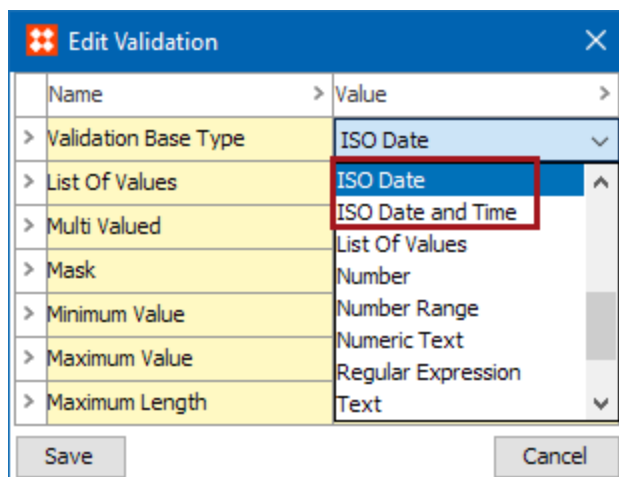
4. Restart the STEP Workbench, and verify the **Migrate Attribute Values** option no longer displays in the Maintain menu.



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 a search, comparing, or other 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.



There are 2 ISO Date formats available as shown above:

- **ISO Date:** When this option is used as a validation, the format in which the date will be entered is 2016-12-16.
- **ISO Date and Time:** When this option is used as a validation, both the date and time will be displayed such as 2016-12-16 17:35:45

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.

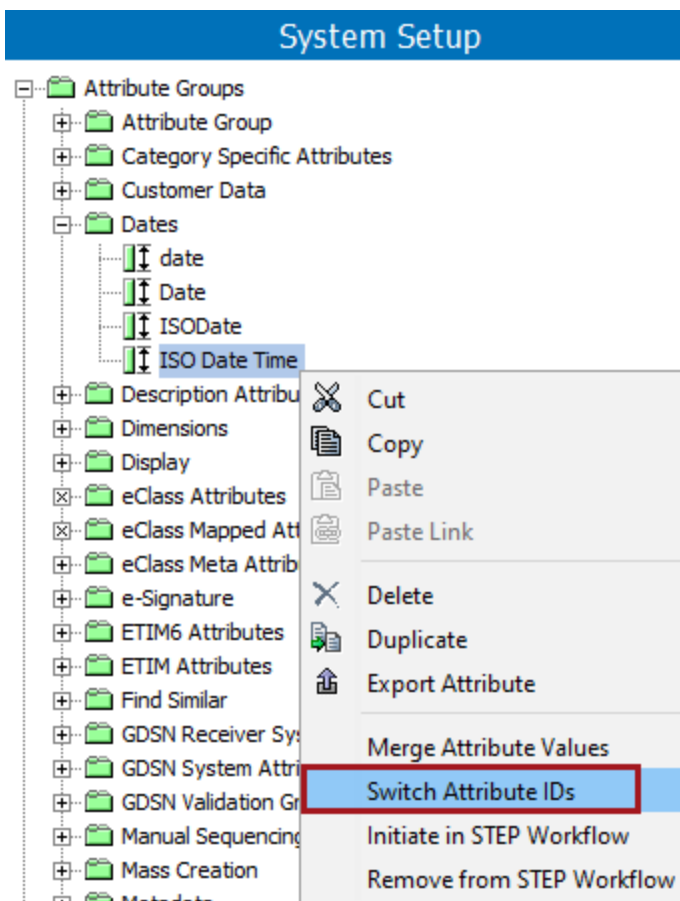
For more information about merging and the steps to merge attribute values, see **Merging Attribute Values** in the **Attributes** section of the **System Setup / Super User Guide** documentation.

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.

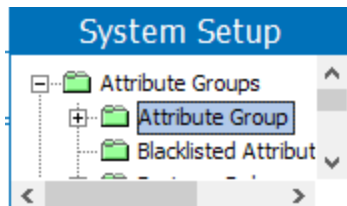
1. To begin this process, start by navigating to the *target* attribute name in the System Setup tab. Right-click on the attribute name and select **Switch Attribute IDs**.
2. Next, choose the source attribute from the tree shown on the 'Select Attribute for ID-Exchange' screen or use the Search tab 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.



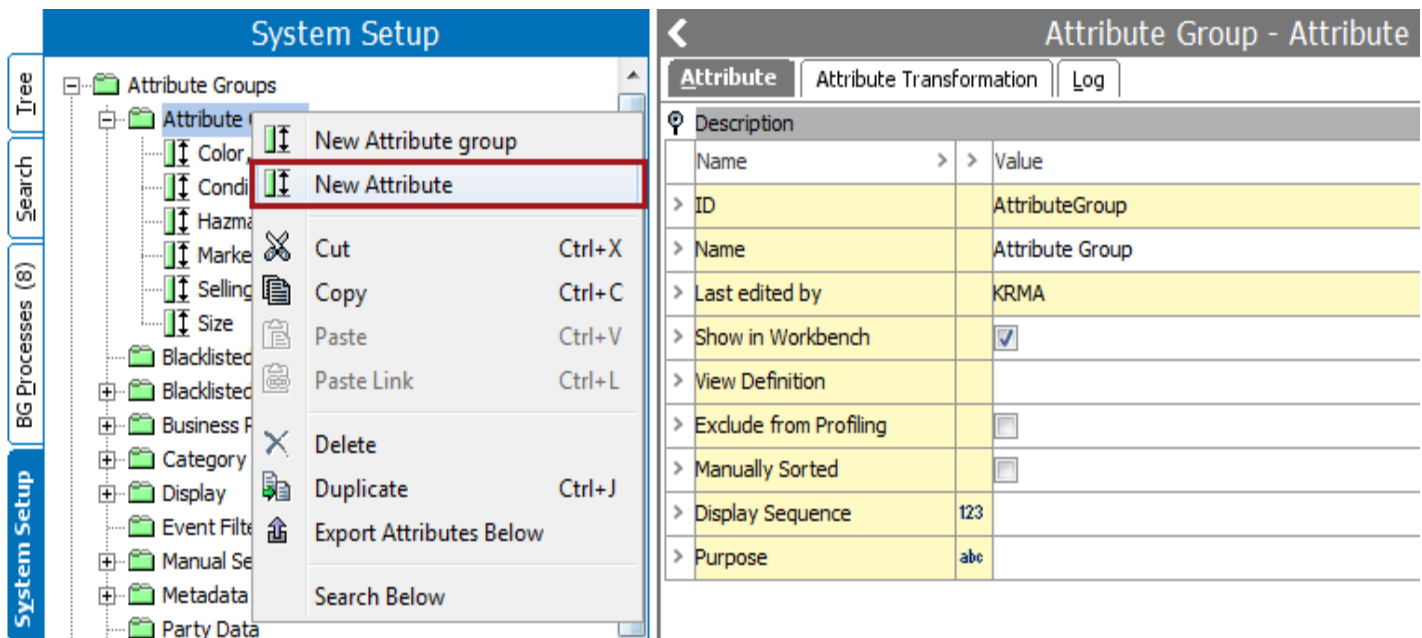
Creating Attributes

Attributes are maintained in System Setup, and are characteristic or a detailed pieces of information that relate to a particular object, or show relations between objects (Reference / Link). To create an attribute, follow the directions below.

1. In System Setup, expand **Attribute Groups**, and then select a group for the new attribute to live under.



2. Right-click on the attribute group, and select **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

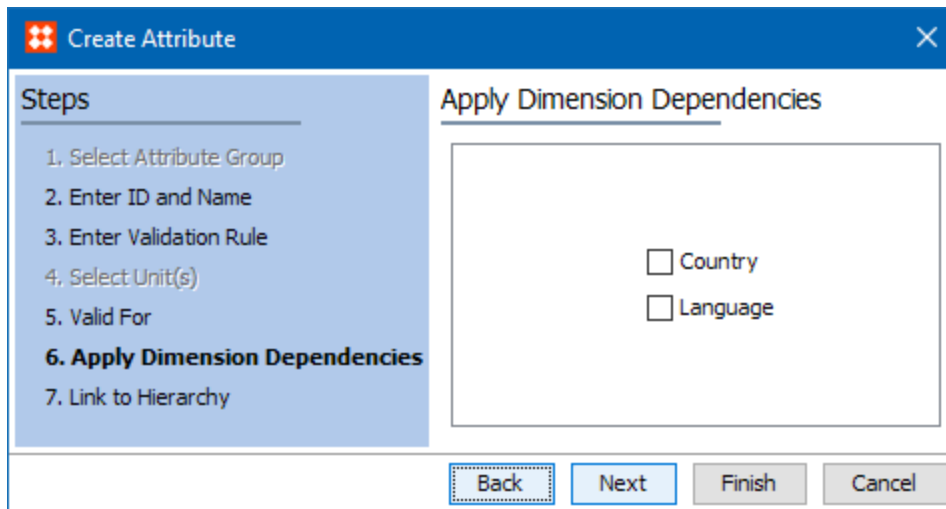
5. The **Select Units** step will only be visible if a numeric validation base type has been selected.
 Search or browse to add one or more relevant Units or Unit Groups.
 Select a unit to be selected as default unit. Click **Next**.

The screenshot shows the 'Create Attribute' dialog box with the 'Steps' panel on the left and the 'Select Unit(s)' panel on the right. The 'Steps' panel lists seven steps, with '4. Select Unit(s)' highlighted. The 'Select Unit(s)' panel has a text input field containing 'ft' and 'in'. Below the input field are 'Add...' and 'Remove' buttons. At the bottom, there is a 'Default Unit' dropdown menu set to 'ft'. The bottom of the dialog has 'Back', 'Next', 'Finish', and 'Cancel' buttons.

6. Select the product types the attribute is associated with. Click **Next**.

The screenshot shows the 'Create Attribute' dialog box with the 'Steps' panel on the left and the 'Valid For' panel on the right. The 'Steps' panel lists seven steps, with '5. Valid For' highlighted. The 'Valid For' panel has a search bar with the text 'Valid for Product Types'. Below the search bar is a list of product types with checkboxes: 'All', 'Board', 'Boards', 'Box', 'Buy Side', 'Carton', and 'Case'. The 'Next' button at the bottom is highlighted. The bottom of the dialog has 'Back', 'Next', 'Finish', and 'Cancel' buttons.

7. Apply Dimension Dependencies. 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

Apply Dimension Dependencies

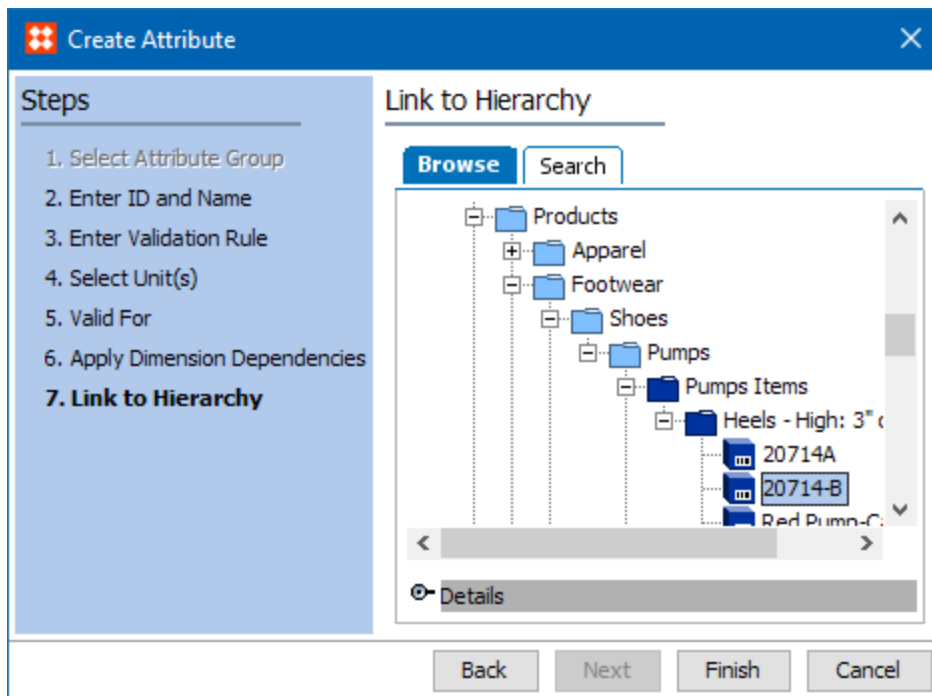
☐ Country

☐ Language

Back Next Finish Cancel

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.



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**

Link to Hierarchy

Browse Search

Products

- Apparel
- Footwear
 - Shoes
 - Pumps
 - Pumps Items
 - Heels - High: 3" c
 - 20714A
 - 20714-B
 - Red Pump-C

< Details >

Back Next Finish Cancel

Click **Finish** to create the attribute.

System Setup

Attribute Groups

Attribute Group

- Color, Interior
- Condition
- Hazmat
- Marketing Description
- Selling Price
- Size

Blacklisted - 2

Size - Attribute

Attribute

References

Attribute Transformation

Validity

Description

Name	Value
ID	Size
Name	Size
Last edited by	2015-08-10 10:47:32 by USER
Full Text Indexable	No

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). Another example is to use the externally maintained attribute for storing a unique ID (EAN, Manufacturer Part Number, etc.) that is not a STEP ID. It could be generated by other systems like an ERP. An example of an externally maintained attributes are 'unique keys'. To know more about unique keys, see the **Creating and Deleting Keys** topic in the **Super User/ Data Quality** documentation.

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 'Main' workspace for an externally maintained attribute, this value is automatically transferred to the other workspaces, e.g., 'Approved'.

When you change an existing internally maintained attribute to an externally maintained attribute (Externally Maintained field changed from No to Yes), you must select the workspace from which values should be taken from and kept on the system. 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.

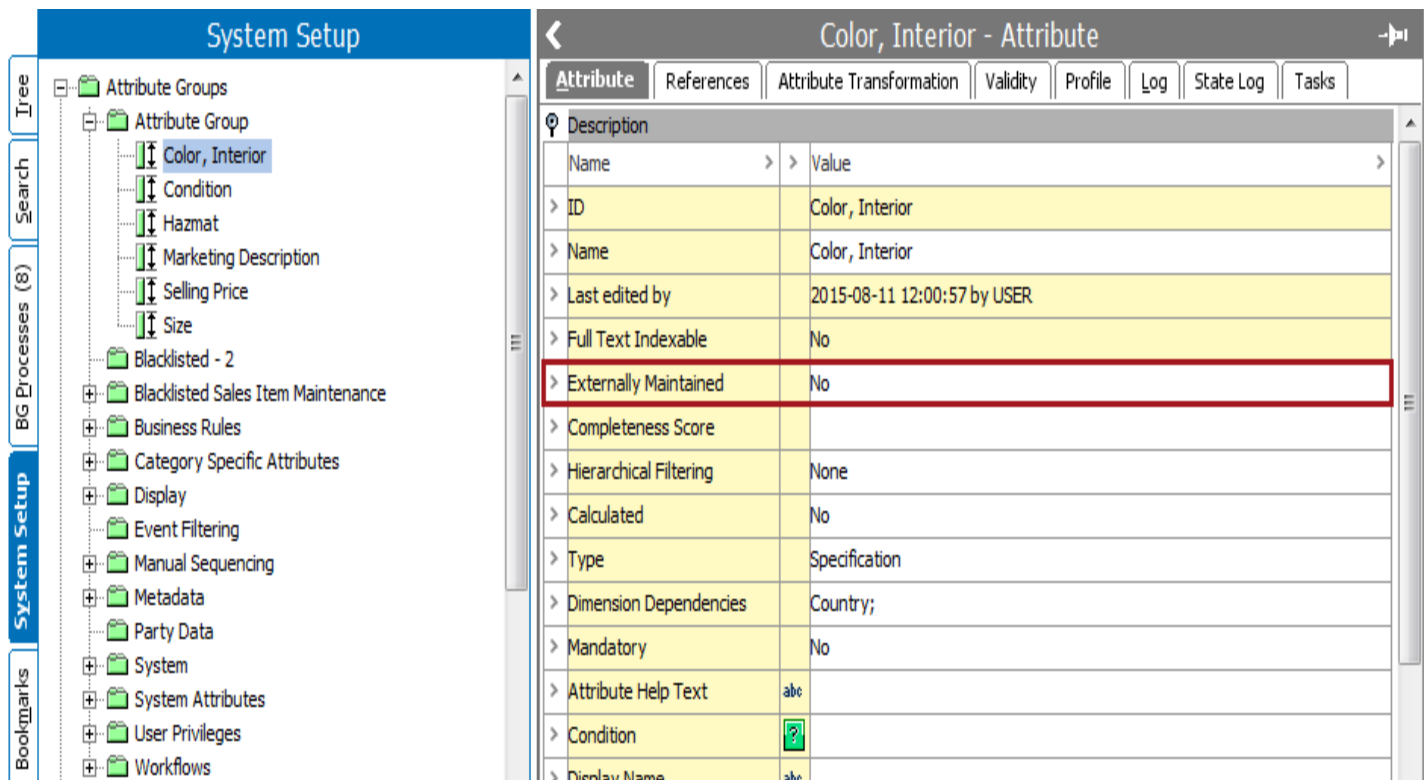
Sharing LOVs

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

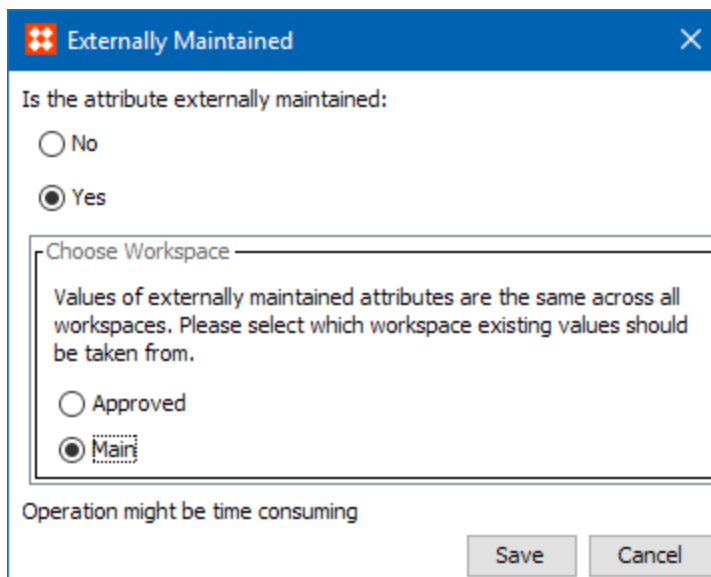
- Setting as externally maintained: If an LOV that is used by any internally maintained attributes is selected, and the attribute is then set as externally maintained, all attributes will become externally maintained.
- Setting as internally maintained: If an LOV that is used by one or more externally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

5. 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

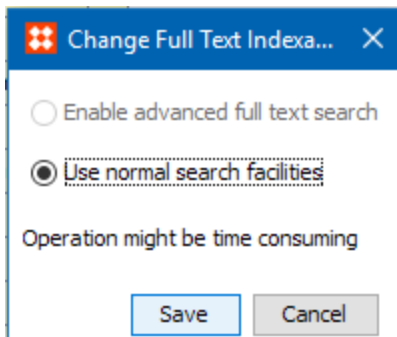
Marketing Description - Attribute

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

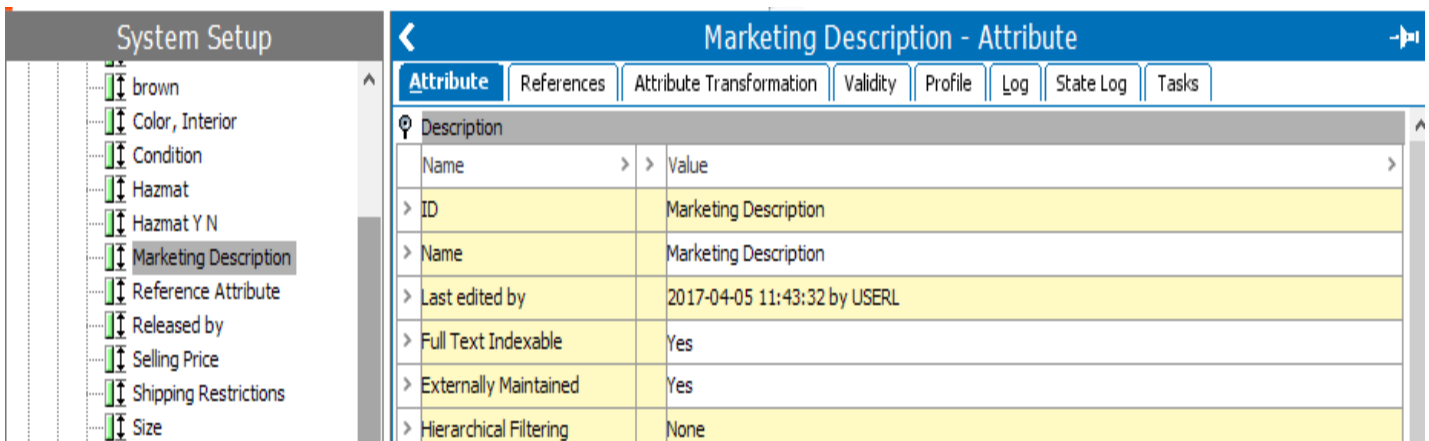
Description	
Name	Value
ID	Marketing Description
Name	Marketing Description
Last edited by	2015-08-11 14:05:14 by USER
Full Text Indexable	No
Externally Maintained	No
Hierarchical Filtering	None
Calculated	No
Type	Description
Dimension Dependencies	Country; Language;
Value template	EMPTY
Mandatory	No
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

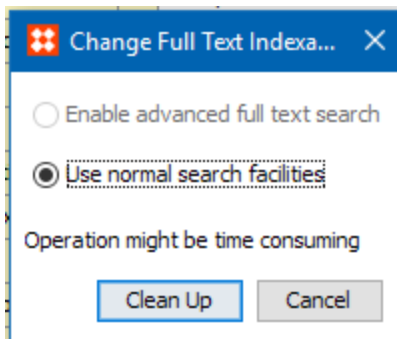
- Click the **Attribute** tab.
- Under **Description**, double-click the **Full text indexable** field. The **Change Full Text Indexable** dialog box appears.



- Click **Enable advanced full text search** and then **Save**.



If an attribute is having Full text Indexable as 'Yes,' and you do not need the Full text indexable option in an attribute, select this field and choose 'Use normal search facilities' and choose the 'Save' button. In some cases this operation might fail, then choose the same option, and select the 'Clean Up' button.



Important: If the **Enable advanced full text search** option is grayed out, and you want to use this 'Full text Indexable' functionality, please contact Stibo Systems. Changing Full Text Indexable option is time consuming, and it is based on the values present for that attribute. Without the full text indexable option you can still search for values. This option is only needed if you want to search for objects based on a word or set of words in a sentence in an attribute value. When an attribute is made as Full text indexable, there are some chances of having performance issues. Apply this option to attributes only if needed.

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.

Tree

Search

BG Processes (8)

System Setup

Bookmarks

TEP Workflow

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

Category Endpoint

Color, Interior - Attribute

AttributeReferencesAttribute TransformationValidityProfileLogState LogTasks

Description

Name	Value
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

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

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

Use of Value-filtering in Hierarchy

Determine if value-filtering may be applicable to

☒ Product and Classification Hierarchy

☐ Product Hierarchy

☐ None

Save

Cancel

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

Color, Interior - References

AttributeReferencesAttribute TransformationValidityProfileLogState LogTasks

Valid in Classifications

ID	>	Name	>

Link to ClassificationSource from Attribute

Valid in Products

ID	>	Name	>	Condition	>	Display Sequence
101585	>	Regional Products				

Link to ProductSource from Attribute

In Attribute Groups

ID	>	Name	>
AttributeGroup		Attribute Group	

Add Attribute Group

Values for Attribute

Value Distribution

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

Note: To modify filters you must have privileges to maintain Attributes. For more information see the Attribute Links topic in the Super User Guide / System Setup documentation.

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

Define Value Filter

☐ Do not use local filter
 ☒ Make local filter

Included in filter

Inherited filter values

Black
Orange

Inherited filters originates from

Name
>  Color, Interior ID = Color, Interior

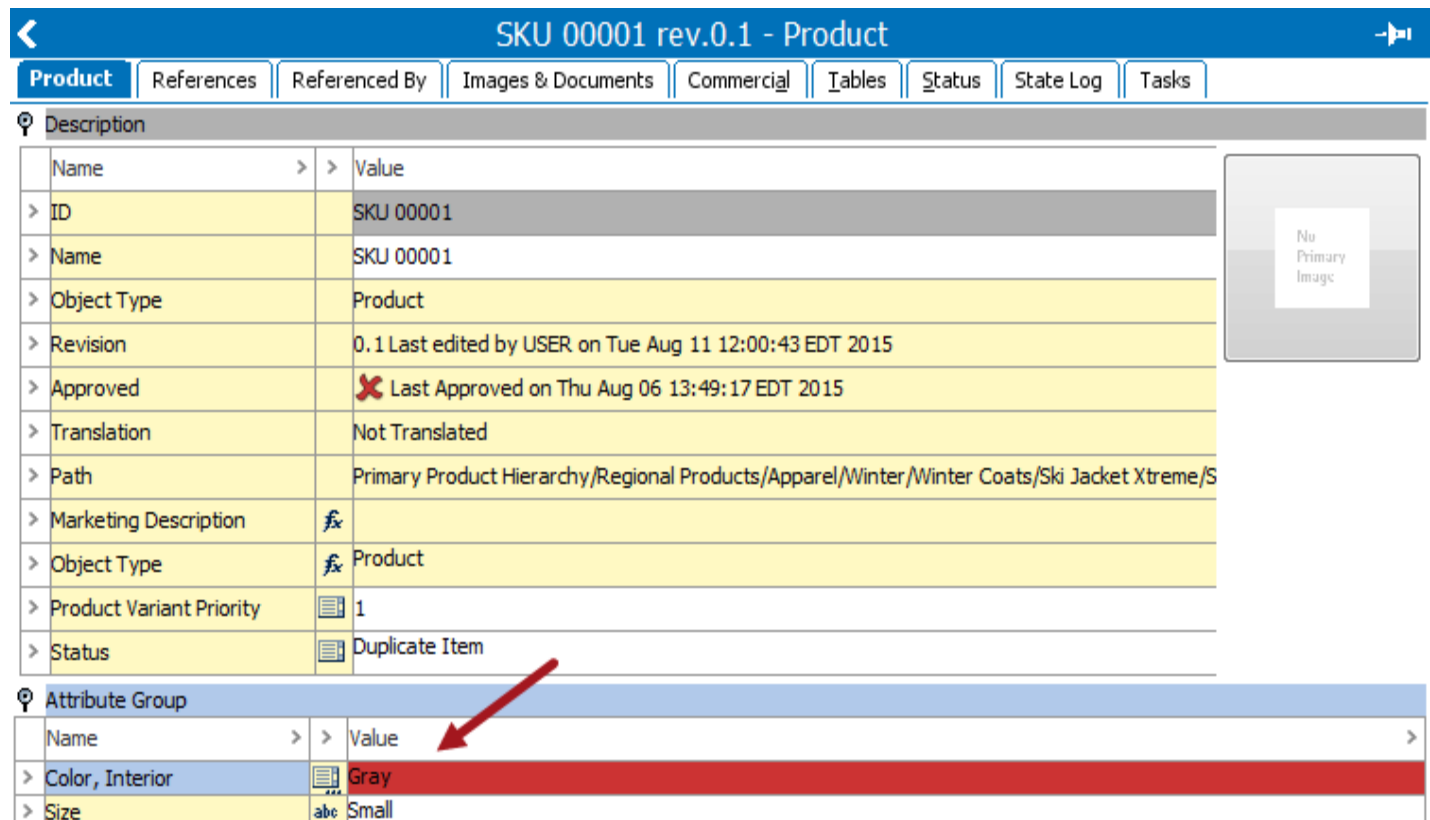
OKCancel

- 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

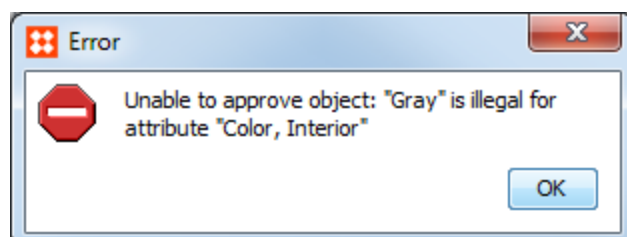
Description

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	fx
Object Type	fx Product
Product Variant Priority	1
Status	Duplicate Item

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 display automatically in the **Description** field of all objects of this object type.

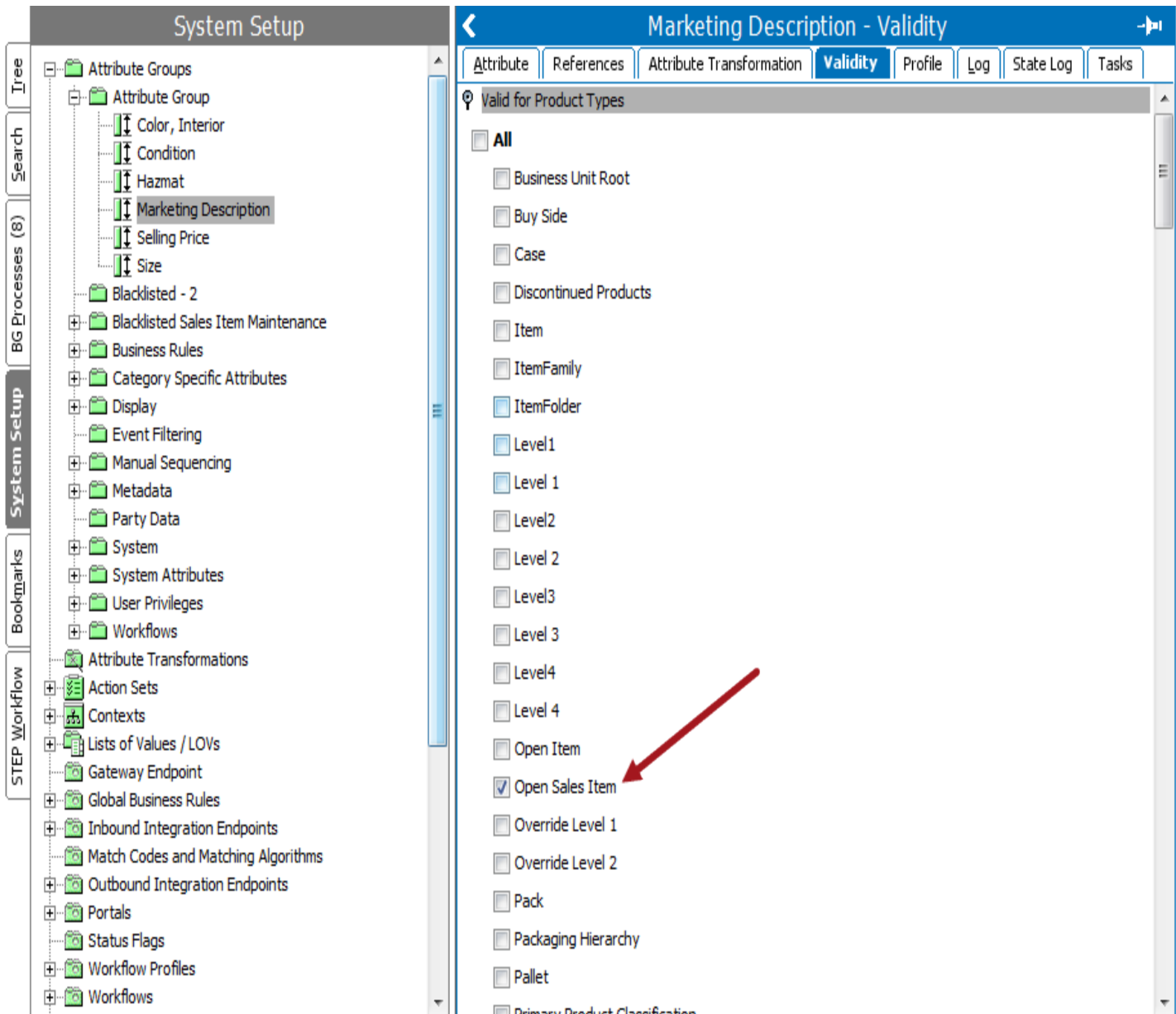
Object Type	Description
Product object types	The attribute will be valid for the selected product object types, and will display automatically in the Description field. Note: As default, the Attribute is set to be valid for all product object types.
Classification object types	The attribute will be valid for the selected classification object types, and will display automatically in the Description field.
Asset object types	The attribute will be valid for the selected asset object types (Images & Documents), and will display automatically in the Description field.
Entity object types	The attribute will be valid for the selected Entity object types, and will display automatically in the Description field.
Publication object types	The attribute will be valid for the selected publication object types, and will display automatically in the Description field.
Attribute Link Types	The attribute will be valid for Attribute link types and will display in Product to Attribute links or Classification to Attribute links
Reference types / Classification Link Types	The attribute will be valid for the selected link types, and will display automatically in the reference editors in product, classification and/or asset editors.
Version Types / E-catalog types	The attribute will be valid for the Publication Version objects / E-catalog objects based on the selection

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.

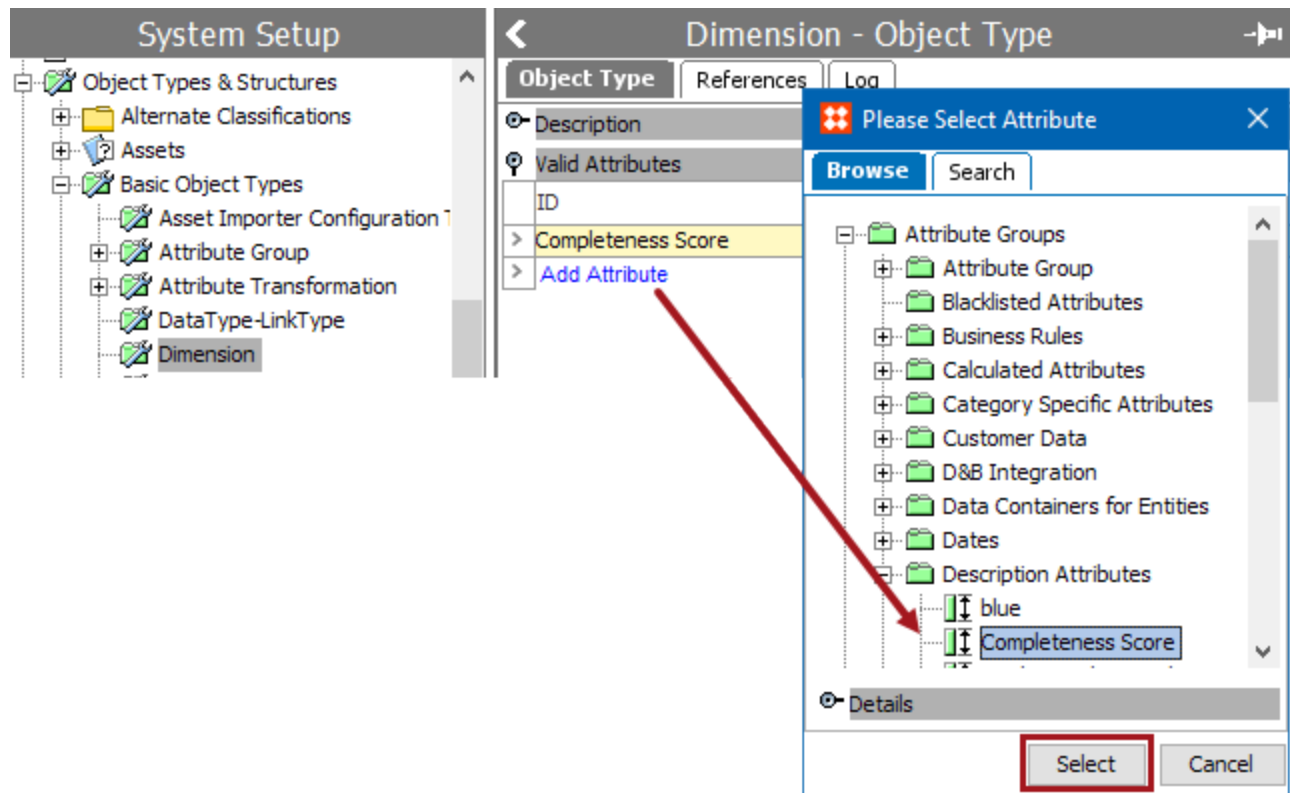
It is possible to select product, classification, entity, or asset object types. Selecting an object type will automatically let the attribute display in the **Description** field of all selected object types.

It is possible to select a link type. Selecting a link type will automatically let the attribute display on product, classification and/or asset references.

A user could select a product to attribute / classification to attribute link types. Selecting this will automatically make the attribute visible on the Link to Attribute portion of Products / Classification Reference tabs (or on the Attribute object's Reference tab).

For more on how to link description attributes to reference types and links, see the **Metadata Attributes on Reference and Link Types** documentation.

Additionally it is possible to add the description attribute on any object type, such as an: attribute, attribute groups, users, user groups, etc. To do so, go to System Setup > Object Type and Structures > the desired object type > add the desired description attribute.



For more on description attributes and how to link them, see the **Description Attributes** topic in the **System Setup / Super User** documentation.

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 only be made valid for:

- product object types

Setting validity for a specification attribute means specifying the product object types that can use the attribute.

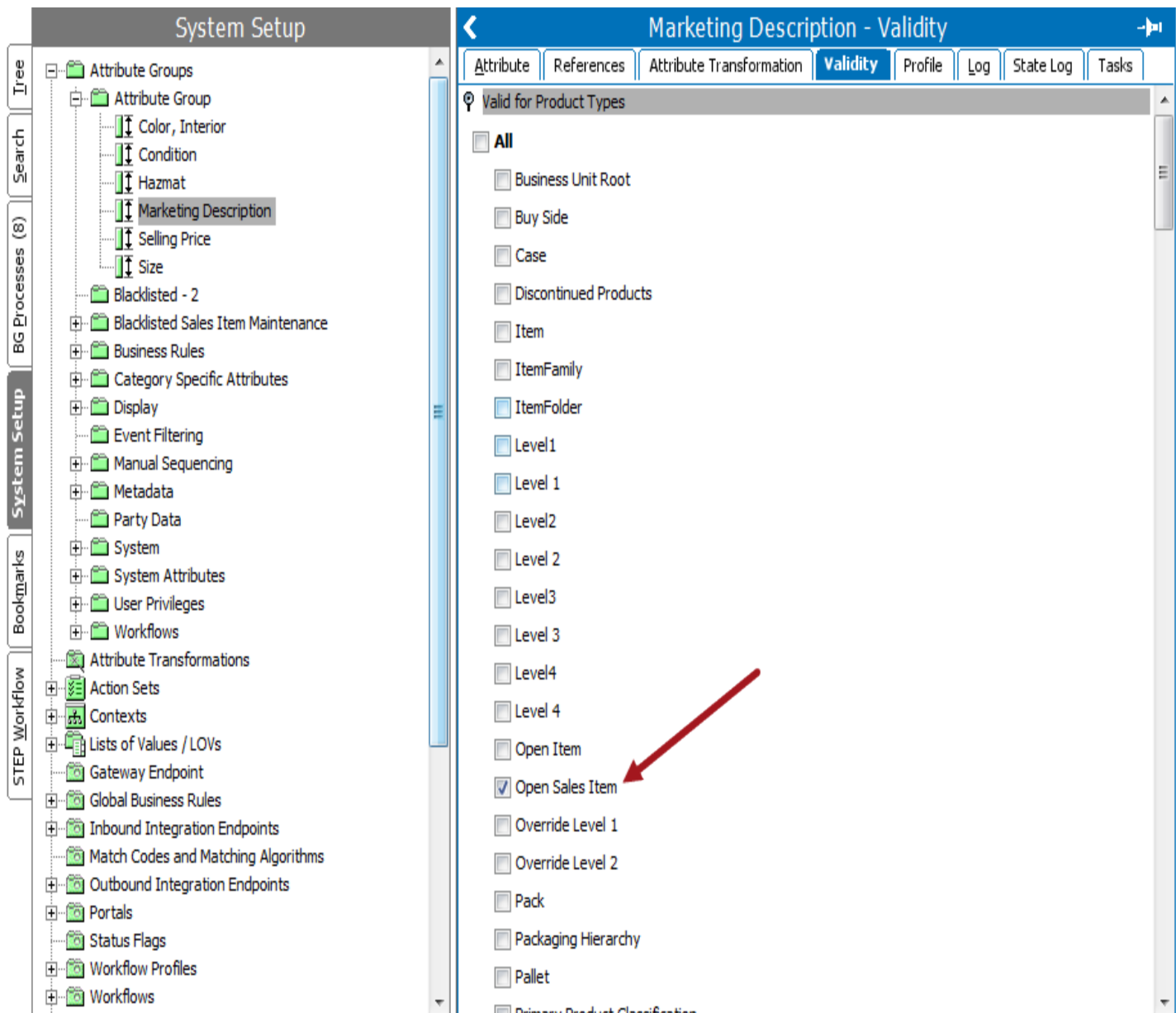
1. In System Setup, expand **Attribute Groups**, expand the relevant group, and then select the relevant attribute.

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.

Note: To make a specification attribute available and editable on a product, the attribute should be made valid on the product object type. It should be linked to the product, the product hierarchy, and classification hierarchy where the product is present under.

Reparenting Attribute Groups

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

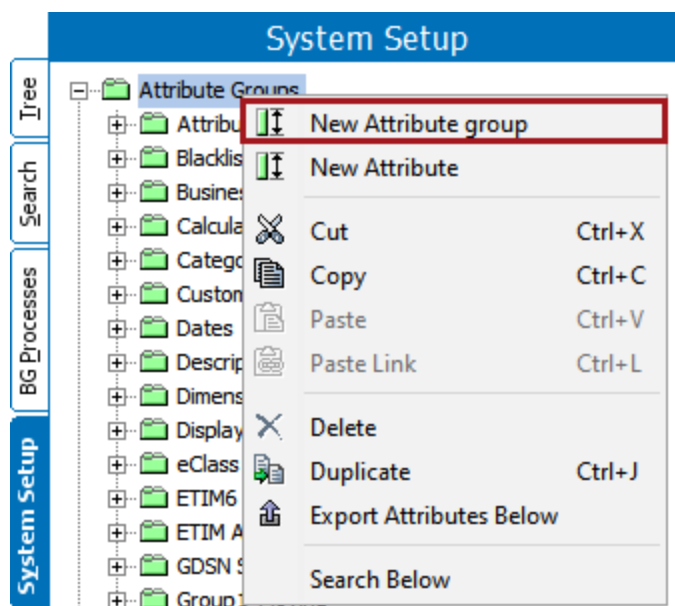
- Manual Reparenting - For smaller attribute groups with only a couple of attributes, users are able to easily drag and drop the attributes under a desired parent.
- STEPXML Reparenting - Using STEPXML, users can reparent an attribute group with numerous attributes inside.

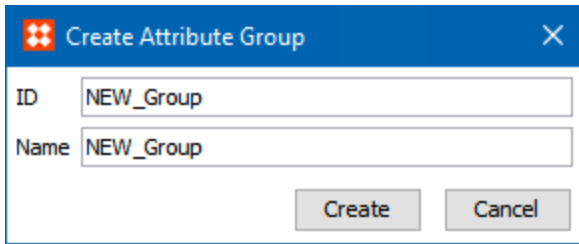
Manual Reparenting

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.

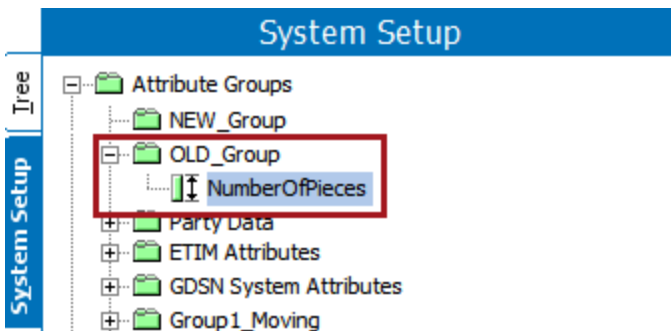
Note: If users wish to reuse the name of the existing attribute group inside of the new parent attribute, then that existing group must be deleted first. The reason for this process is because STEP will auto-generate a name based on the STEP ID, and only one instance of a STEP ID may exist.

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

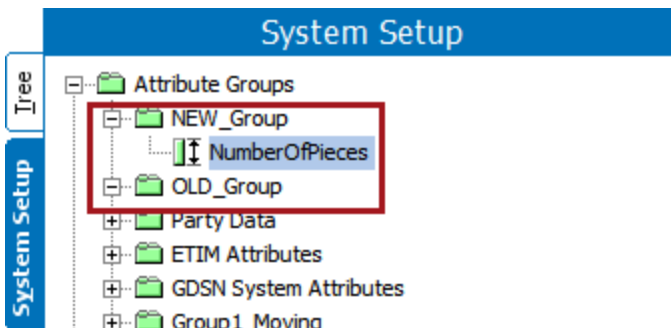




2. Open the attribute group containing the attribute to reparent by pressing the plus (+) icon next to the attribute group name.

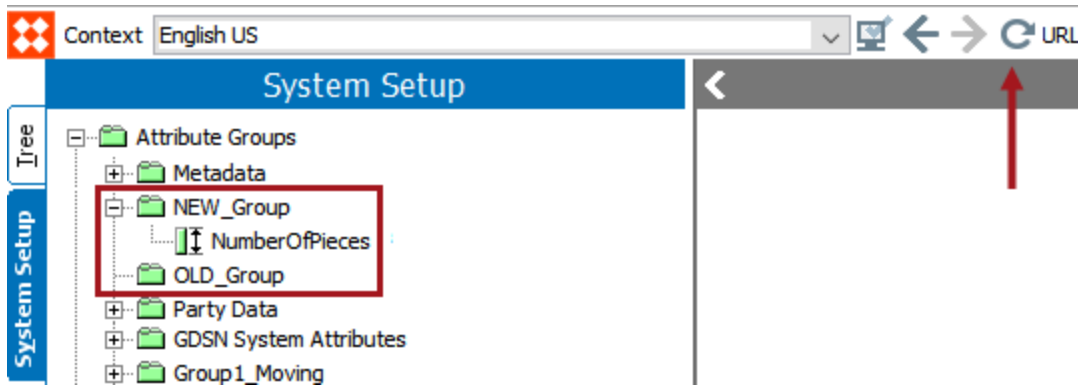


3. Drag the desired attribute(s) into the new attribute group.

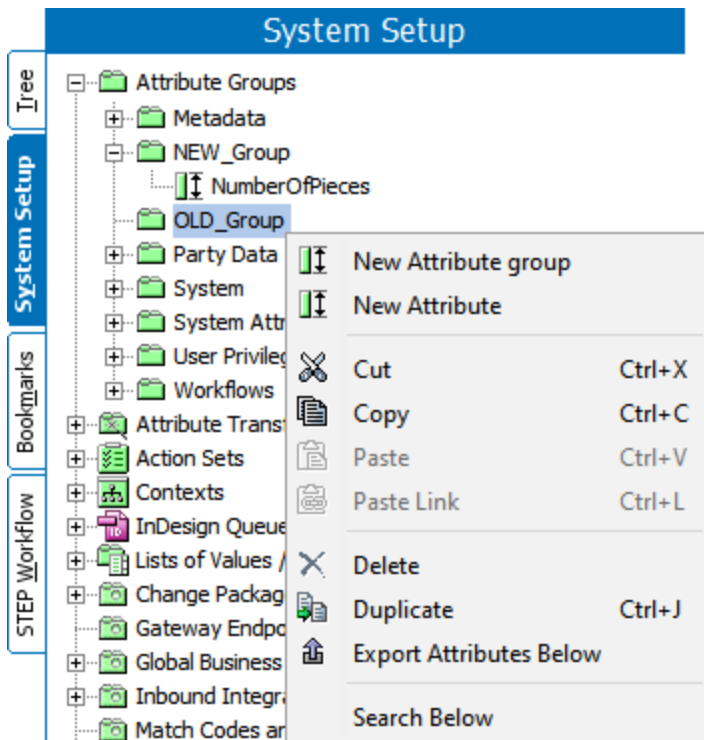


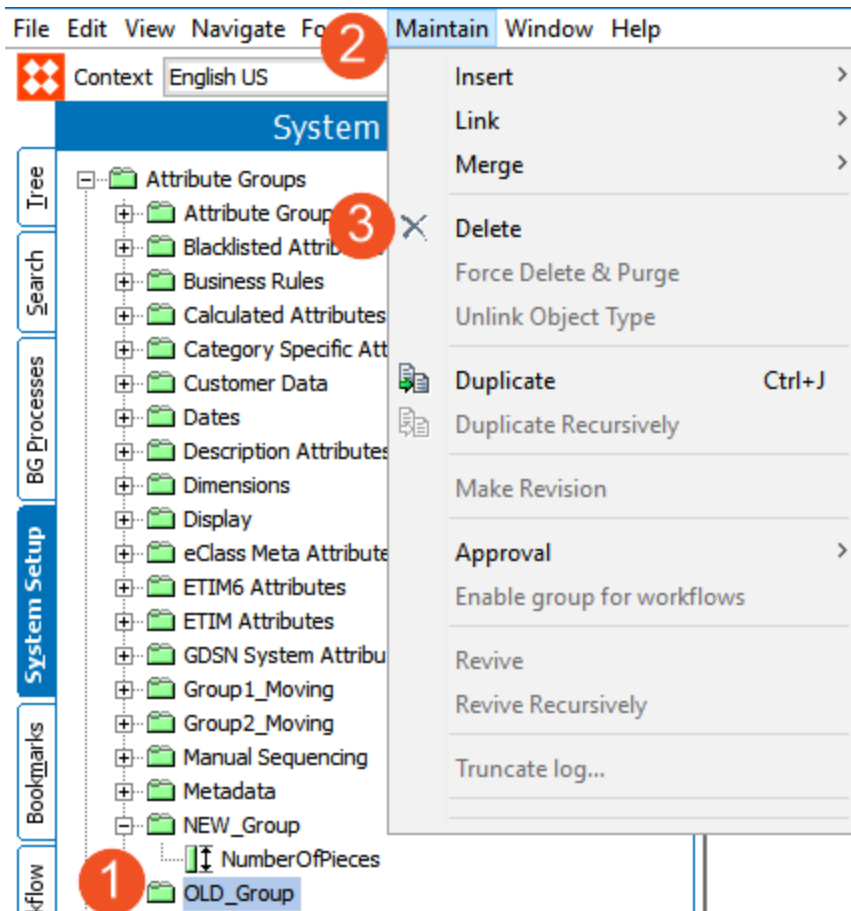
Note: When simply dragging the attribute, the old attribute group will retain the open icon, which is a minus (-) sign. This is due to the fact that the system has not been updated.

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

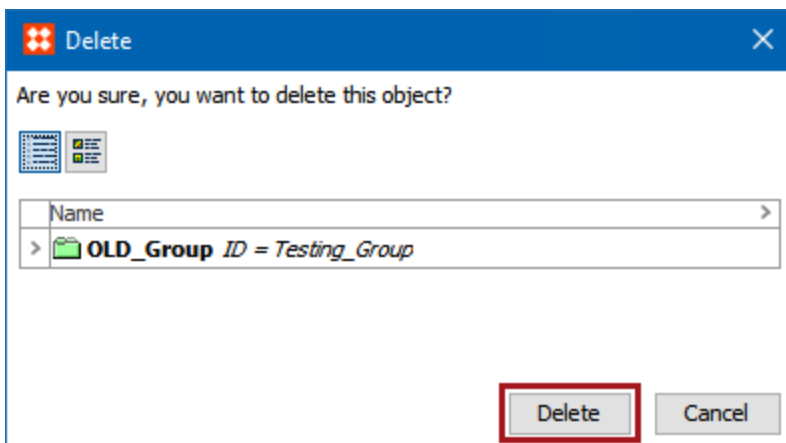


5. Delete your old group from the Tree tab by right-clicking on the attribute group and selecting **Delete**, or by going to the Maintain menu > **Delete**.

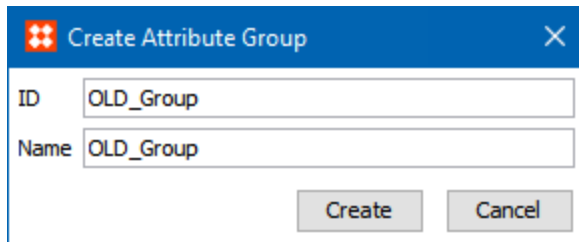
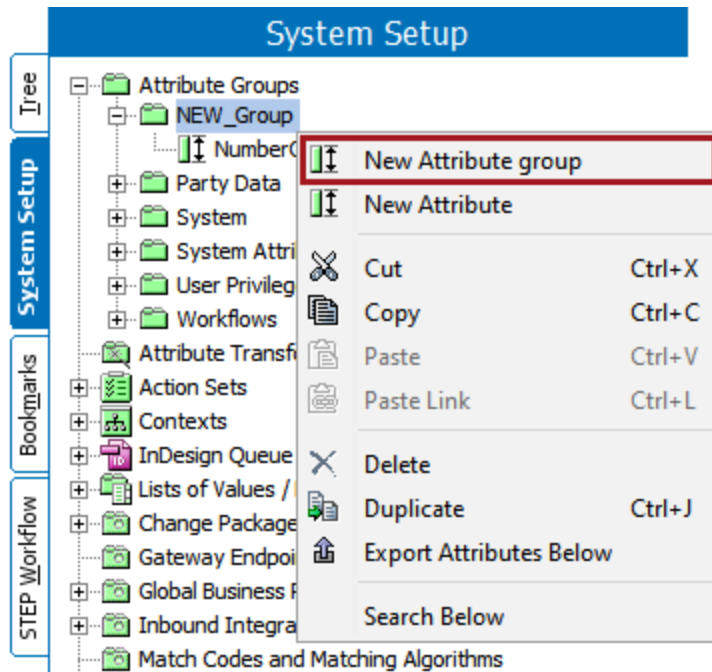




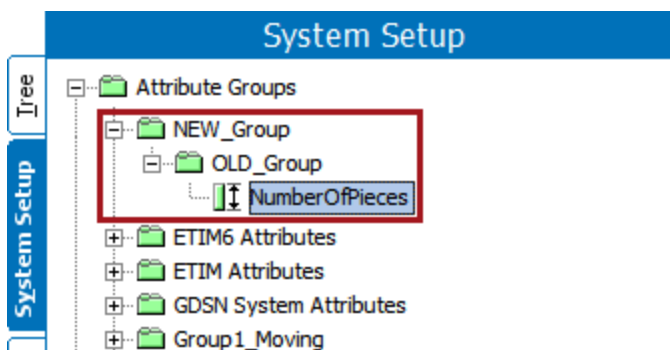
6. Click **Delete** to confirm the deletion on the pop-up prompt.



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.

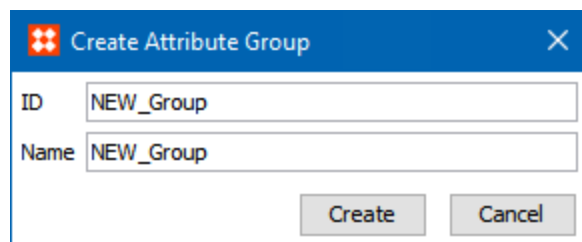
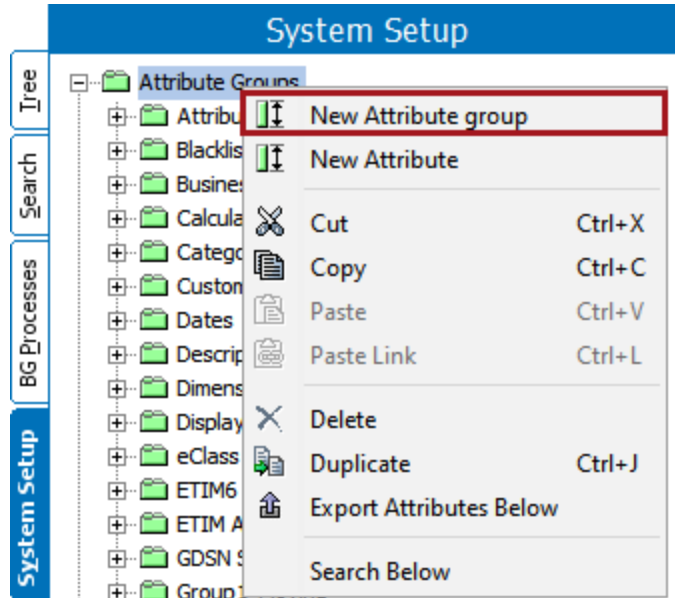


STEPXML Reparenting

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. Using STEPXML a user can link an attribute to new attribute group but cannot unlink an attribute from an old attribute group in which it is already linked.

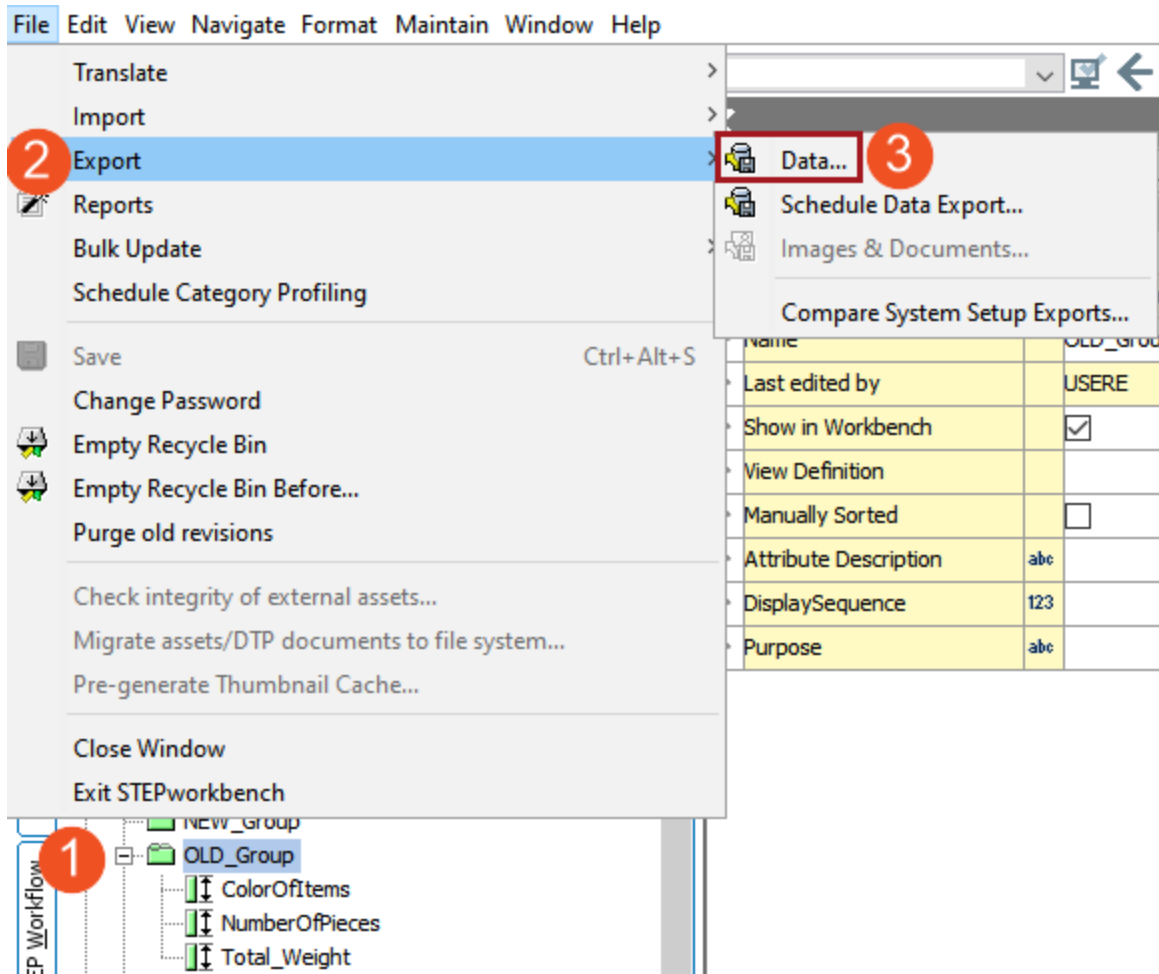
First, create the new group under which your old attribute group will be reparented.



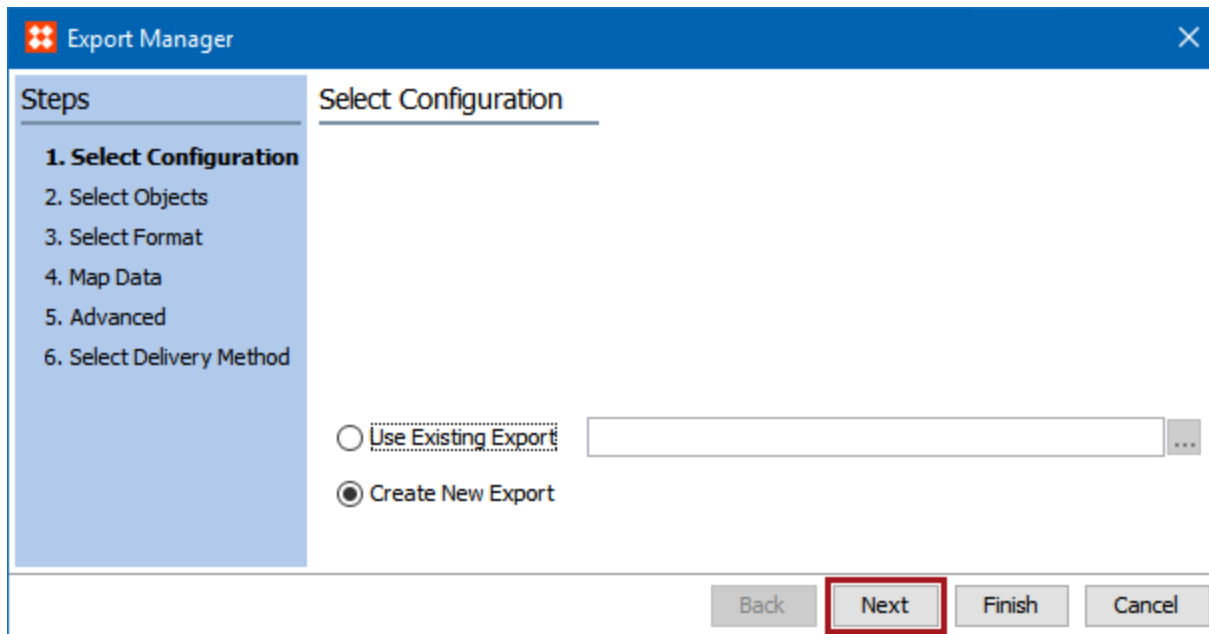
Export the Attribute Group

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....**



2. The Export Manager will display. Select **Next**, and on step 2 of the Export Manager, Select Objects, select **Attribute** from the Export dropdown menu.



Export Manager

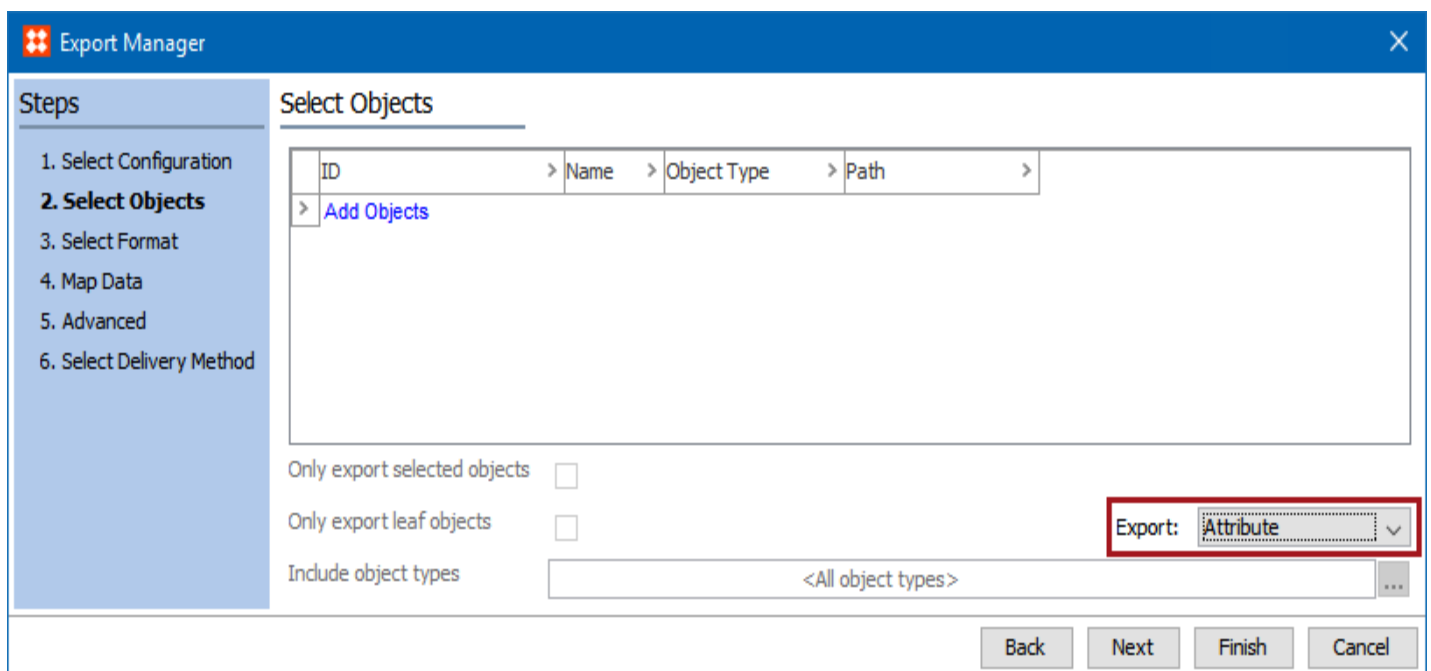
Steps

- 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



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			

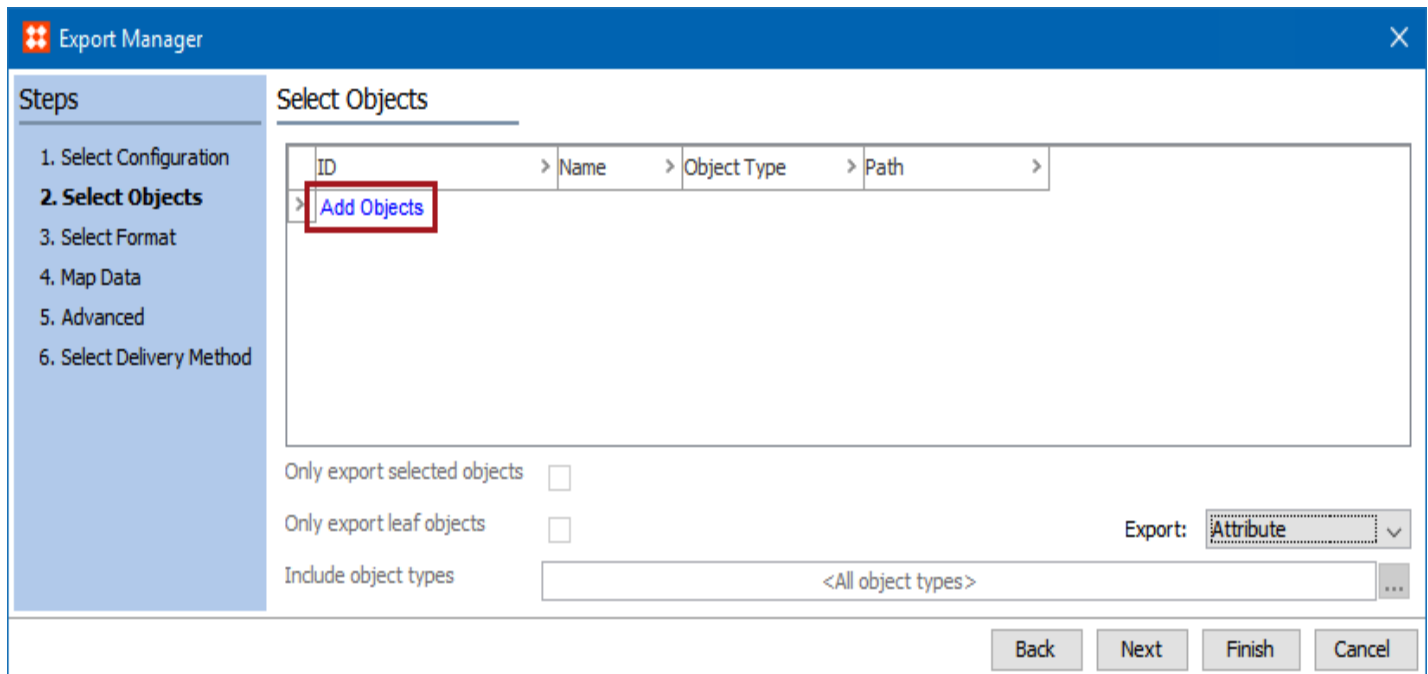
☐ Only export selected objects

☐ Only export leaf objects

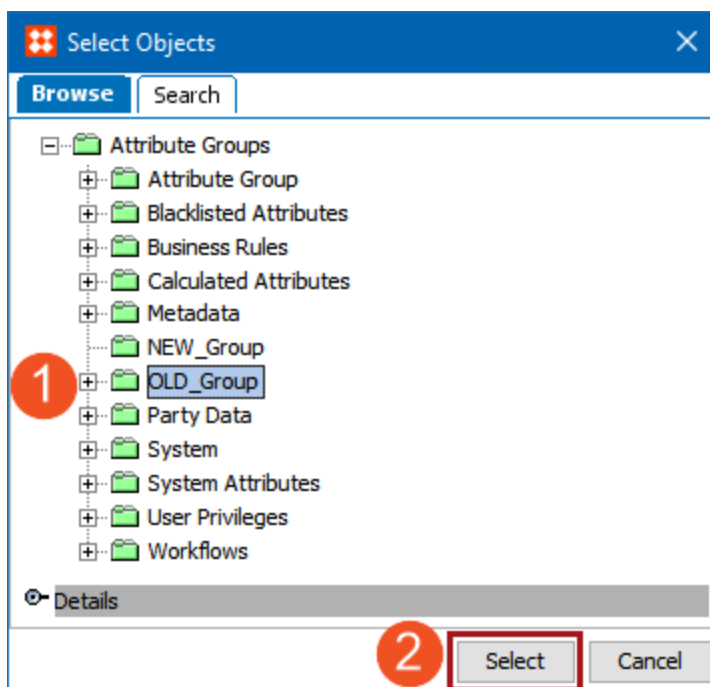
Include object types ...

Export:

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



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

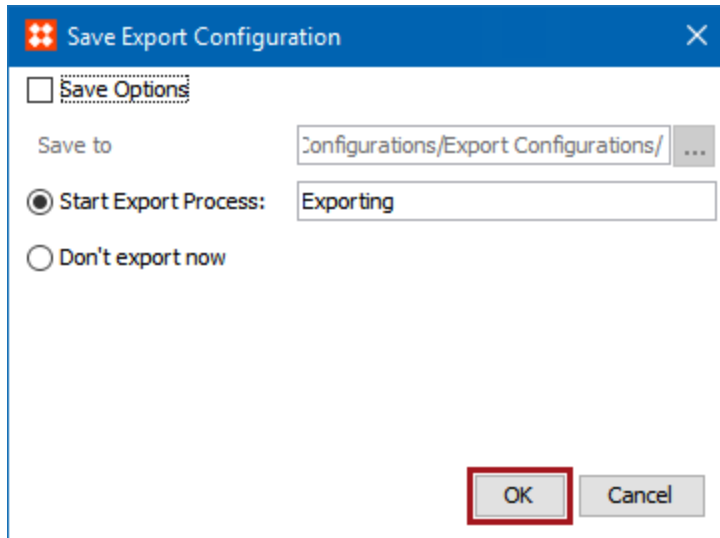
STEPXML

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

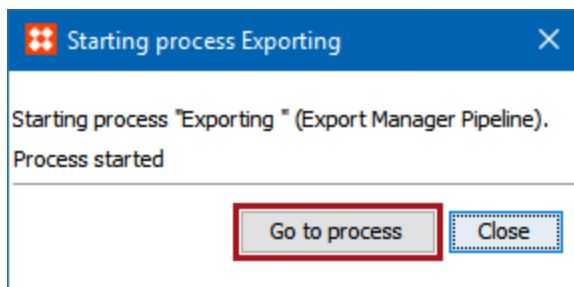
Include Validation	None
Include Type Definitions	None
Include Global Settings	None
Include Context Definitions	None
Include Unit Definitions	None
Include Collection Definitions	None
Include List Of Value Definitions	None
Include Attribute Group Definitions	Selected
Include Attribute Definitions	Selected
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.



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



Exporting - Background Process

Background Process Queue Info

Processing time	Errors
Time in Queue	0 m 1 s
# of warnings	0
# of errors	0

Execution Report

- Export process started (Thu Sep 22 11:53:47 EDT 2016)
- Logging on to PIM server doc-rel as USERE...
- Logged on
- Analysis started. (Thu Sep 22 11:53:47 EDT 2016)
- Analyzed 0 objects from initial object selection in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Analyzed 0 children objects (including aggregates) in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Analyzed 0 initial parents in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Analyzed 0 objects referenced from other objects in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Analyzed 0 referenced parents in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Removed duplicates in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Added 0 classifications to export processing area in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Added 0 assets to export processing area in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Added 0 products to export processing area in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Added 0 entities to export processing area in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Analysis completed in 0 seconds - a total of 0 objects needs processing by exporter. (Thu Sep 22 11:53:47 EDT 2016)
- Export started (resolve inline references=true, value calculation=true). (Thu Sep 22 11:53:47 EDT 2016)
- Starting "Select IDs" process. (Thu Sep 22 11:53:47 EDT 2016)
- "Select IDs" process started in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Starting "Select Values" process. (Thu Sep 22 11:53:47 EDT 2016)
- "Select Values" process started in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)
- Starting "Select Link Values" process. (Thu Sep 22 11:53:47 EDT 2016)
- "Select Link Values" process started in 0 seconds. (Thu Sep 22 11:53:47 EDT 2016)

1-36 of 36 Save... Truncate

Result

Exported file	exported.xml

Adding STEPXML

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

- 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>

```

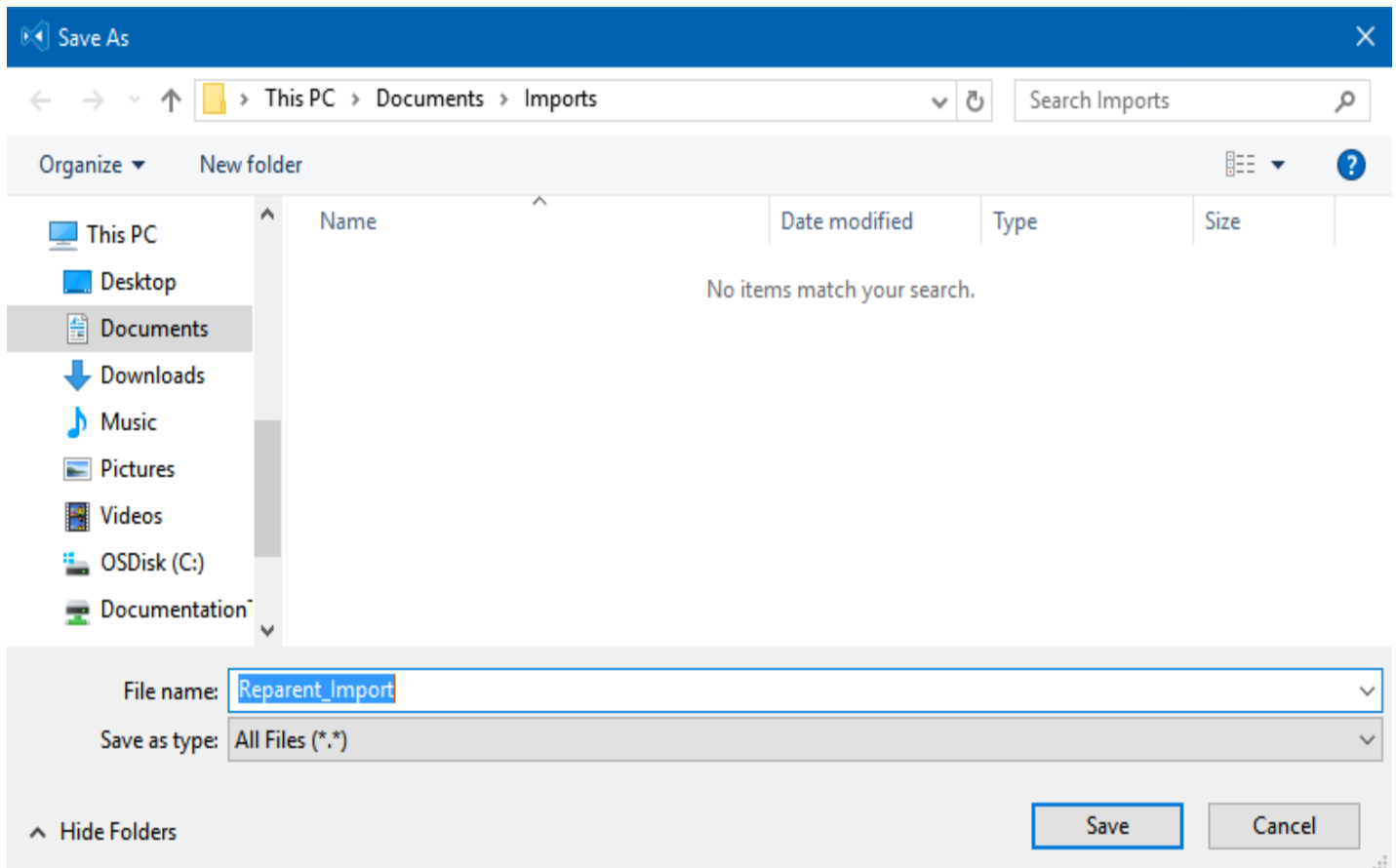
2. Inside the XML brackets for AttributeGroup ID, add ParentID="[Your new group name]". In the braces, use the ID of the parent attribute group where the old attribute group will use as a parent.

```

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>

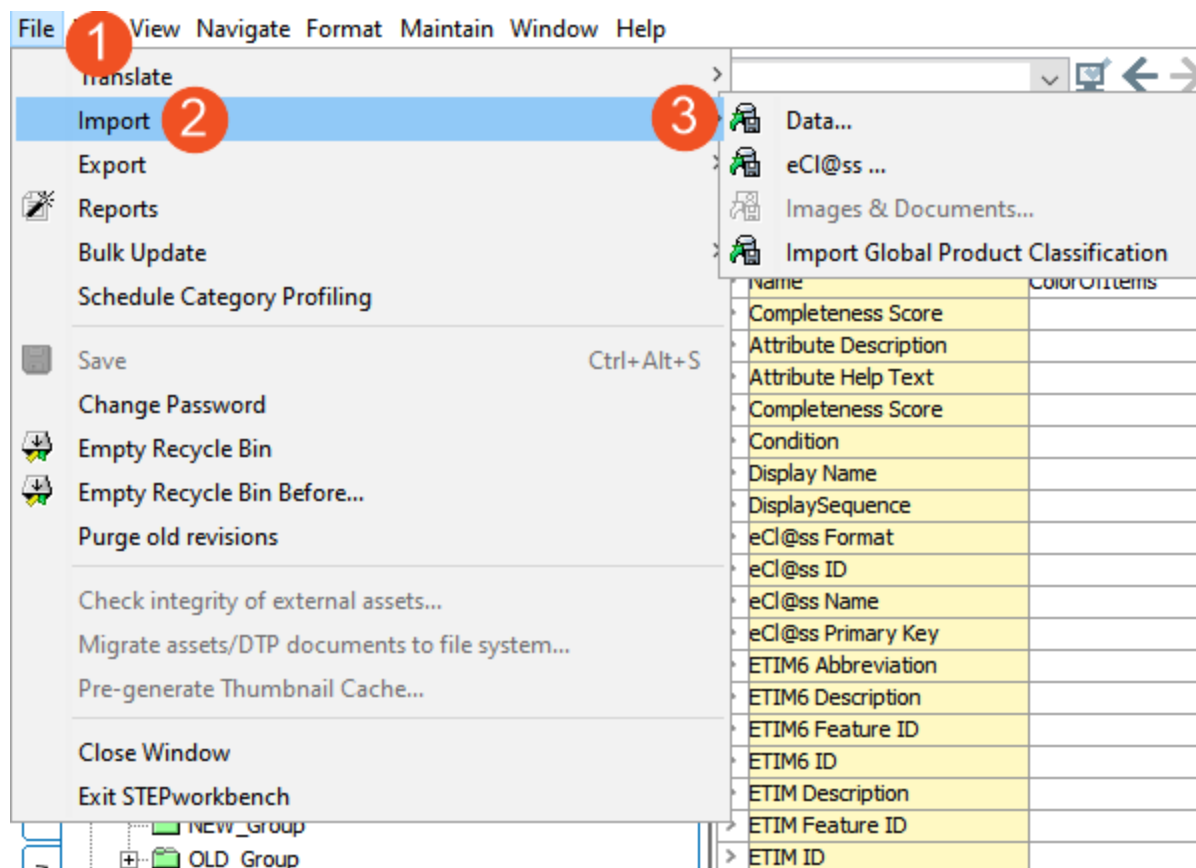
```

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....**

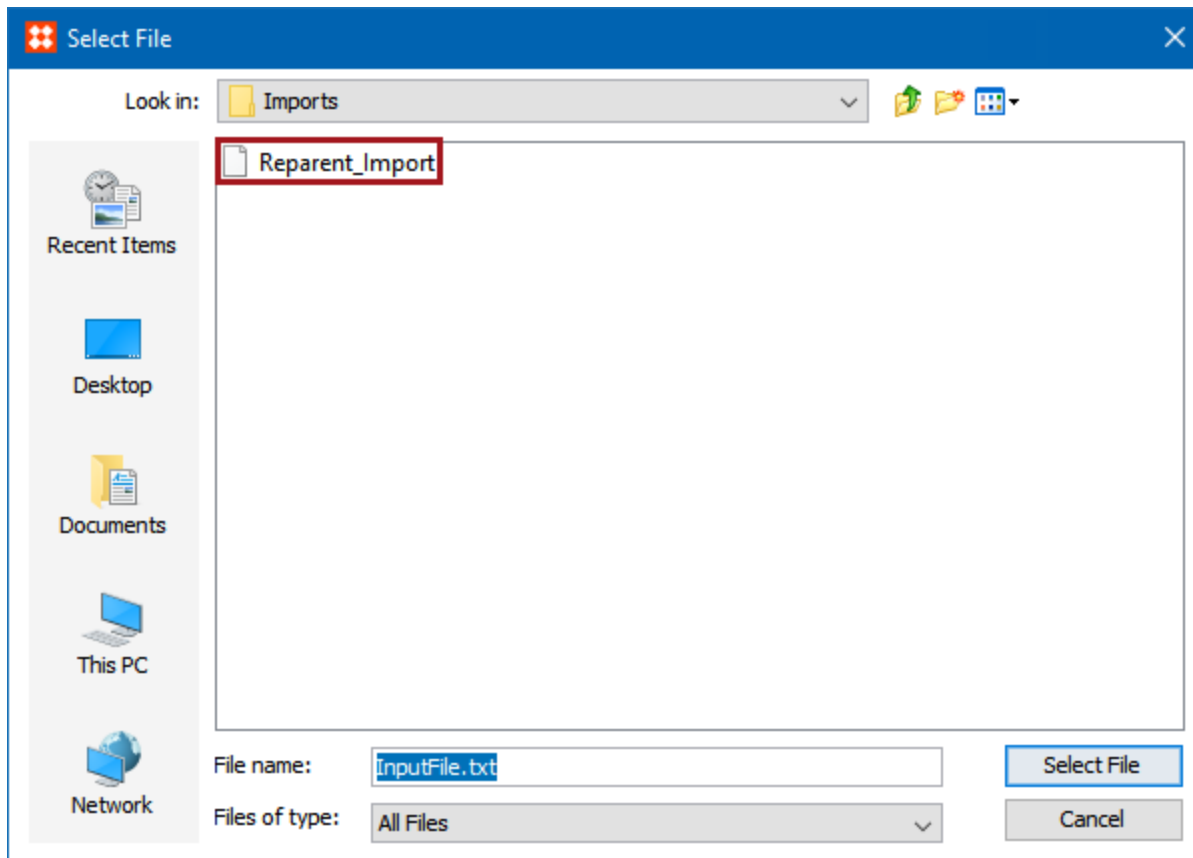


2. The Import Manager will load. Select **Next**.

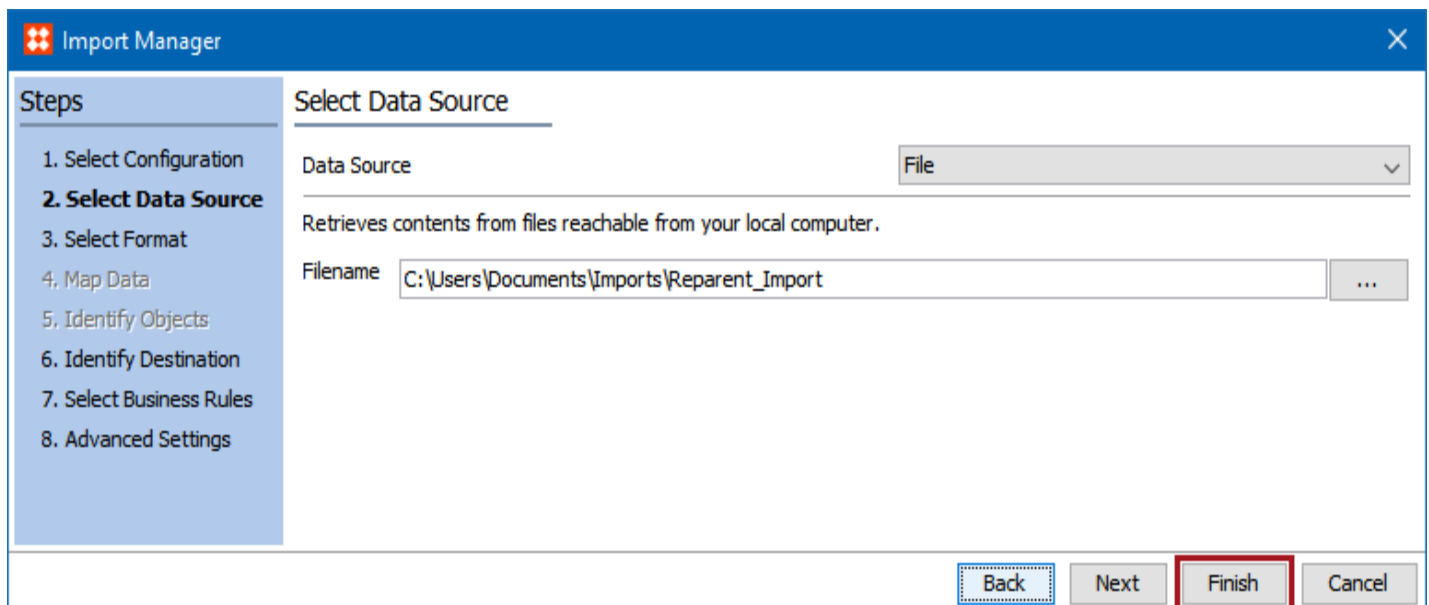
The screenshot shows the 'Import Manager' dialog box with the title bar and a close button. On the left, a 'Steps' list contains eight items, with '1. Select Configuration' highlighted. The main area is titled 'Select Configuration' and contains two radio buttons: 'Use Existing Import' (unselected) and 'Create New Import' (selected). Next to the 'Use Existing Import' button is a text field with a browse button (three dots). At the bottom, there are four buttons: 'Back', 'Next' (highlighted with a red box), 'Finish', and 'Cancel'.

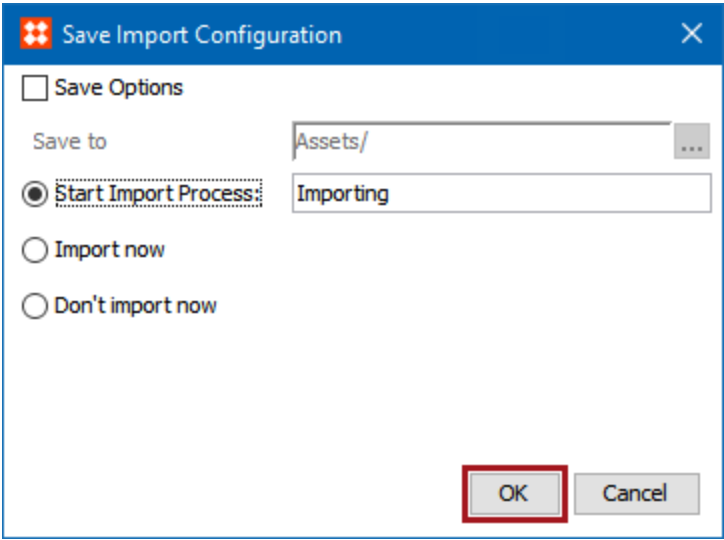
3. On **Step 2. Select Data Source**, select the previously modified STEPXML file from the last step.

The screenshot shows the 'Import Manager' dialog box with the title bar and a close button. On the left, a 'Steps' list contains eight items, with '2. Select Data Source' highlighted. The main area is titled 'Select Data Source' and contains a 'Data Source' dropdown menu set to 'File'. Below this, it says 'Retrieves contents from files reachable from your local computer.' There is a 'Filename' text field containing '/InputFile.txt' and a browse button (three dots) to its right, which is highlighted with a red box. At the bottom, there are four buttons: 'Back', 'Next' (highlighted with a blue dashed box), 'Finish', and 'Cancel'.

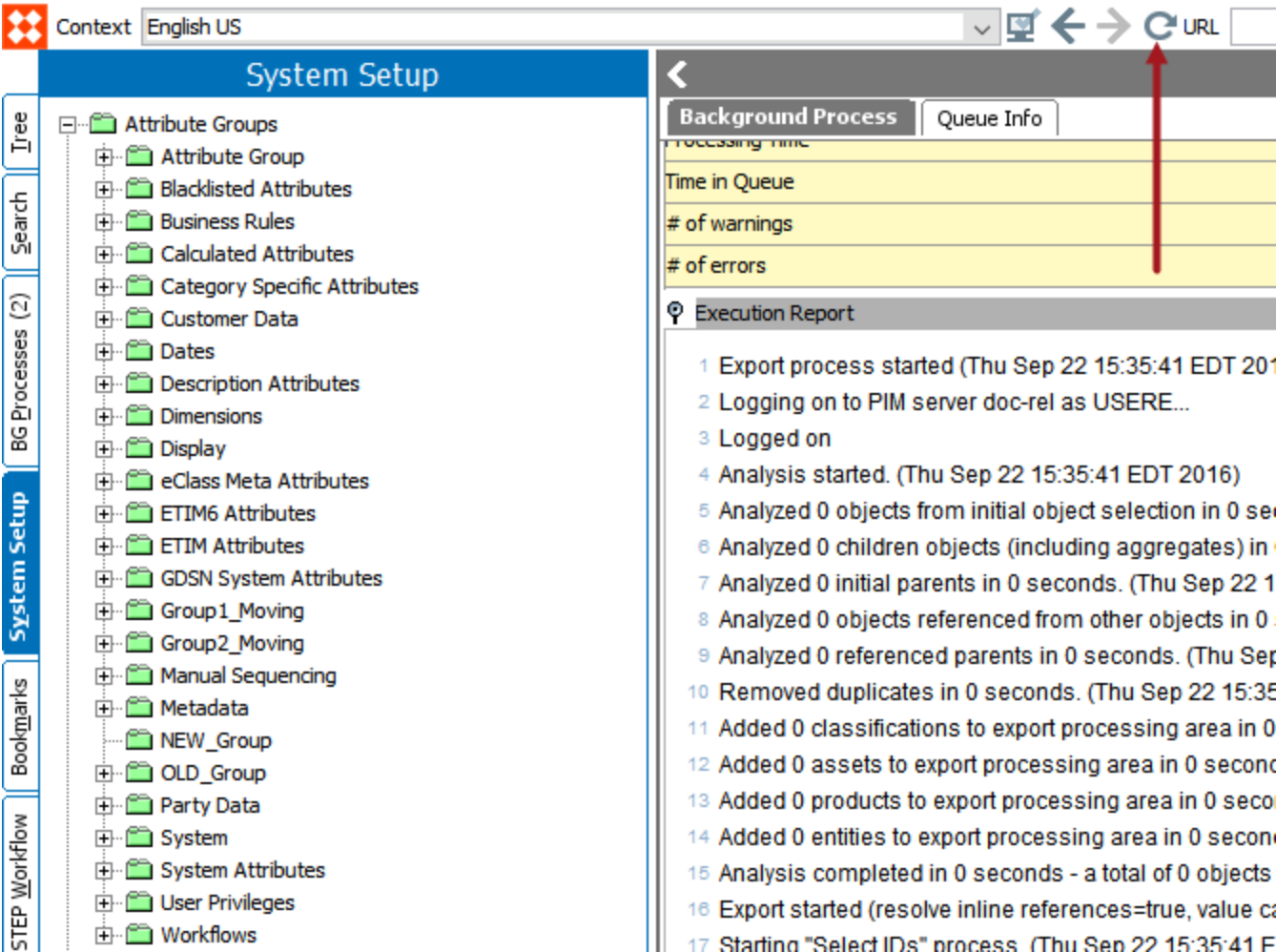


4. Select **Finish**. A Save Import Configuration dialog will pop-up. Select **OK** to import the STEPXML script.





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



System Setup

Tree

Search

BG Processes (2)

System Setup

Bookmarks

STEP Workflow

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:

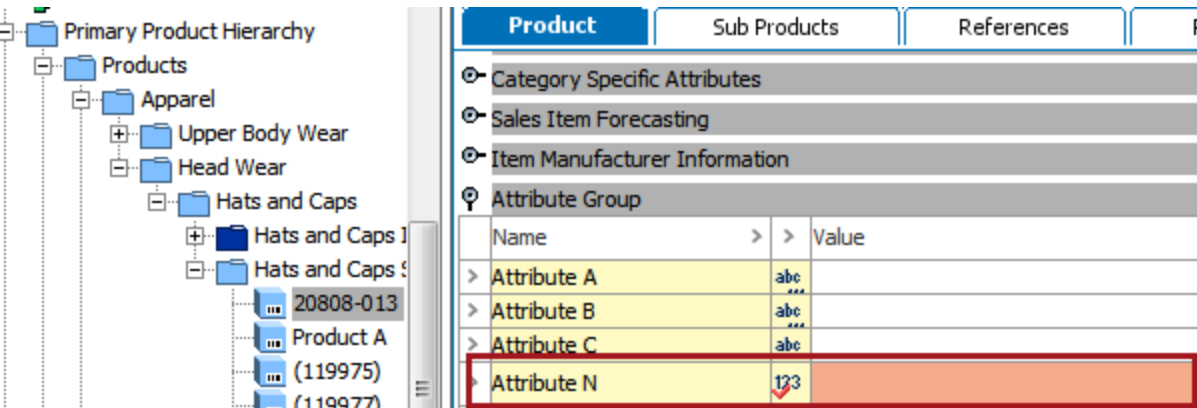
- The options for making attributes mandatory
- Detailed configuration and end user information for the various settings, or links to other topics that provide this information
- Considerations for how the different mandatory options behave across the varying interfaces (workbench, Web UI, and Smartsheets)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



The attribute indicator has a red check mark 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**.

System Setup

Attribute Groups

Attribute Group

Calculated Attributes

Approve

Attribute A

Attribute B

Attribute C

Attribute N

Attribute Q

Attribute Z

AttWithLangDepLOV

Biodegradable

Calculated Asset File Name

Calculated Attribute

Condition

Condition2

Description Attribute 1

Attribute N - A

AttributeReferencesAttribute TransformationValidity

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

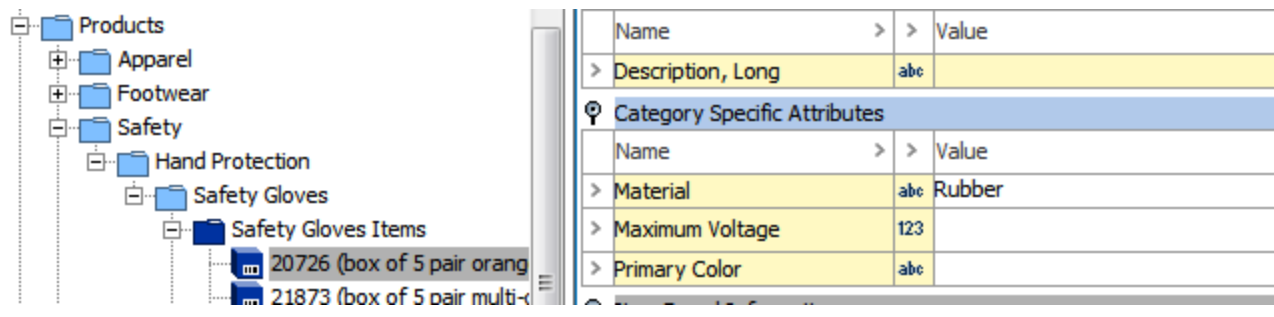
20881 C Battery 2 Pad

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

© Stibo Systems - Confidential - STEP Trailblazer 8.2

355

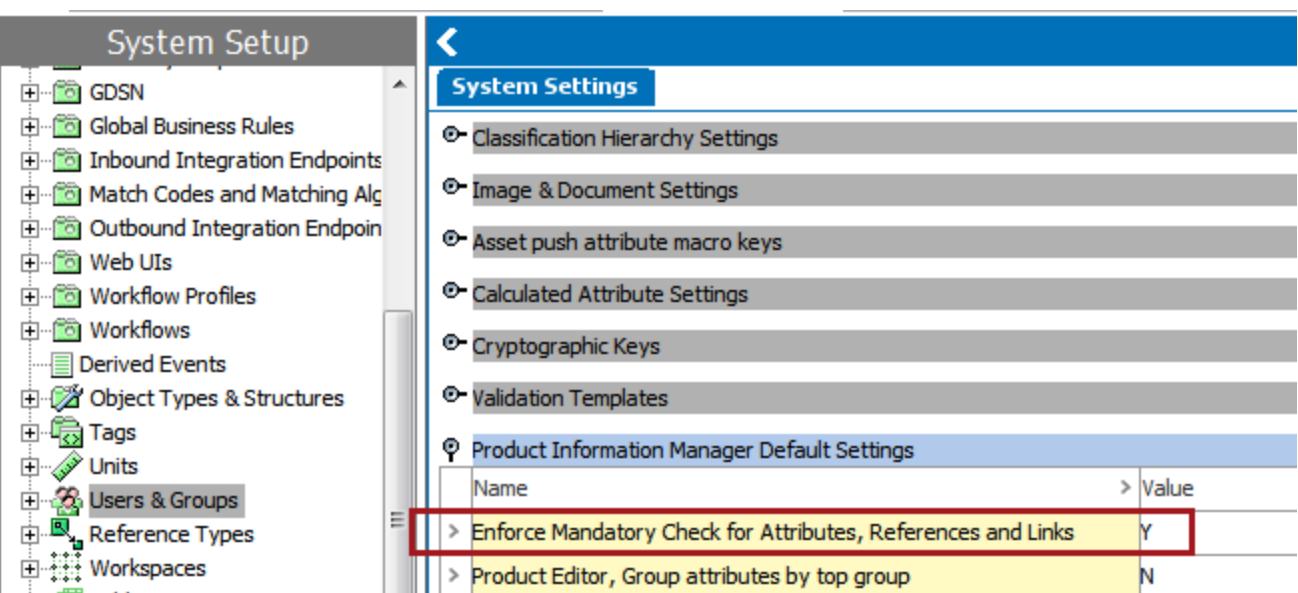


Approval Mandatory Attributes

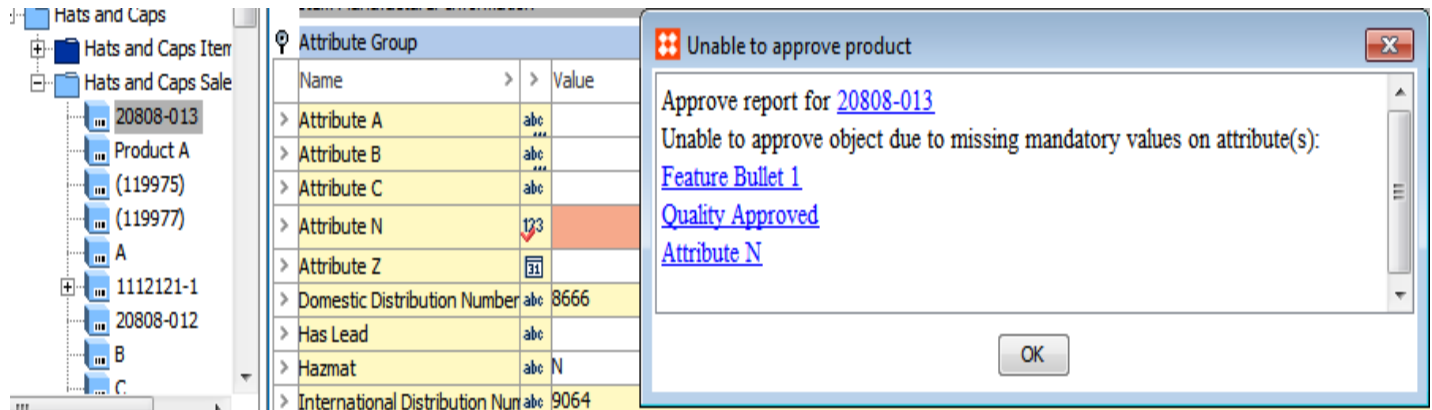
In addition to visual indicators, the system can be configured to *enforce* that the attribute is populated prior to the attribute being approved. This option is most often used only as a final "safety check" and is applied only to attributes that are required for ALL objects of a particular type or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

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

Export Manager

Steps

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

Select Format

Excel Smartsheet

Exports hierarchical data in interactive Excel format

Excel version: Excel 2007

Smartsheet type: Multiple level. Hierarchical structure flattened to sheet

Smartsheet usage: Data export (for maintaining product data)

Object Types: Select a product object type for each desired level.

1: Item (Item)

2: Sales Item (SalesItem)

3:

Use Cross-Context Export: No

Smartsheet import configuration:

Mandatory metadata attribute: Smartsheet Mandatory (SmartsheetMandatory)

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.

In addition to the above option to make attributes mandatory , it is also possible to do so in the export manager. See the **Smartsheet Data and Template Configurations** topic in the Data Exchange documentation.

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

Packaging Hierarchy

Name	18216 L O
ID	181951
Primary Product Image	
Product Category <i>fx</i>	Apparel Upper Body Wear T-shirts T-shirts Items Cotton T-Shirts 
Approved	✗ Last approved 6/14/17 2:08 PM
Primary Color*	Amber 

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

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*	PrimaryColor
Enable STEP Tags	☐
Enable Tag Conversion	☒
Label	
Label Css Class	
Mandatory	☒
Max Suggestions	
Read Only	☐

▶ LOV Options

▶ Advanced

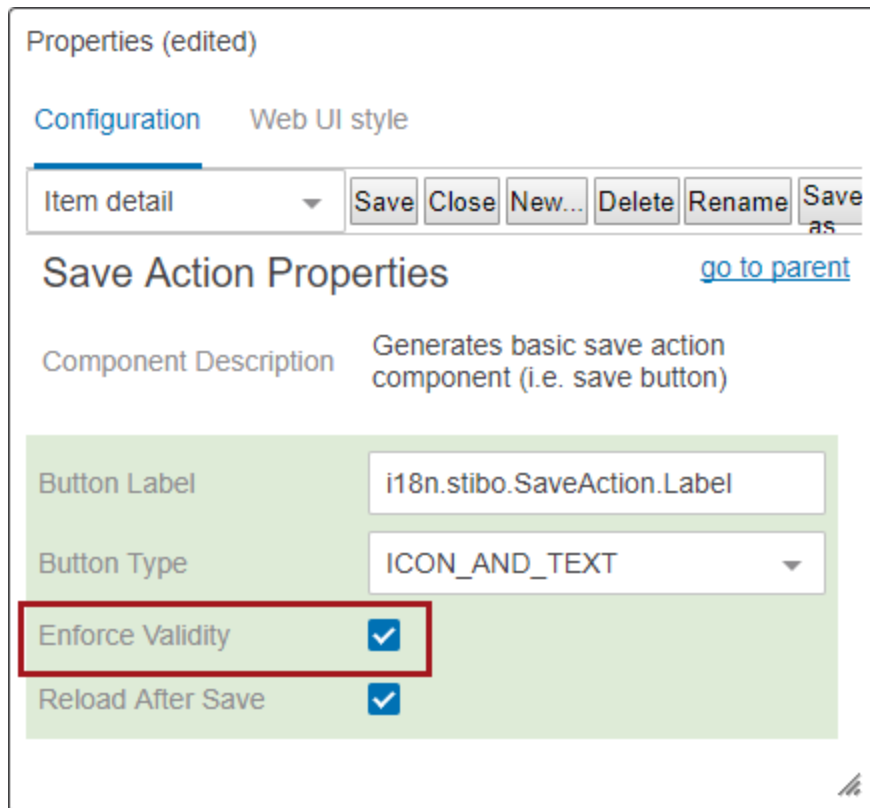
Add
Cancel

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 Component** and **Attribute Value Group** 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.



Properties (edited)

Configuration Web UI style

Item detail Save Close New... Delete Rename Save as

Save Action Properties [go to parent](#)

Component Description Generates basic save action component (i.e. save button)

Button Label i18n.stibo.SaveAction.Label

Button Type ICON_AND_TEXT

Enforce Validity ☒

Reload After Save ☒

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. An example of why this might occur could be due to data loads that happened from different systems and carry the same value in different patterns. 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 redundant attribute values can be deleted.

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.

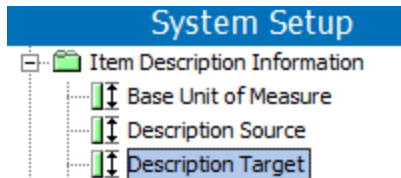
Verify Attributes to be Merged

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.

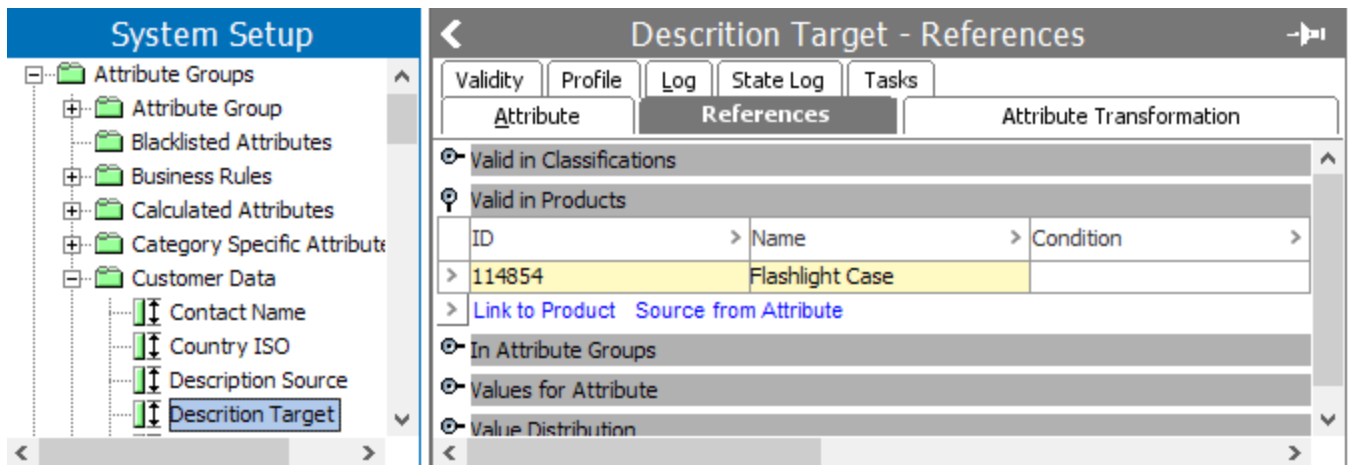
- Ensure that the upstream systems that feed the data to STEP are configured to send the data to the target attribute as the source attribute will be deleted.
- Ensure that if there are any existing import or export configurations using the source attribute, that the configuration is modified. This is to avoid errors when trying to run the configuration after the source attribute is deleted from STEP.

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

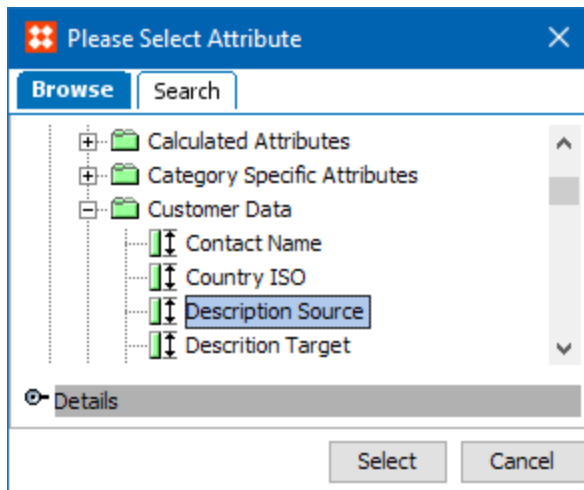


2. On the References tab, if it exists, open the 'Valid in Classifications' flipper, and click the **Source from Attribute** link.

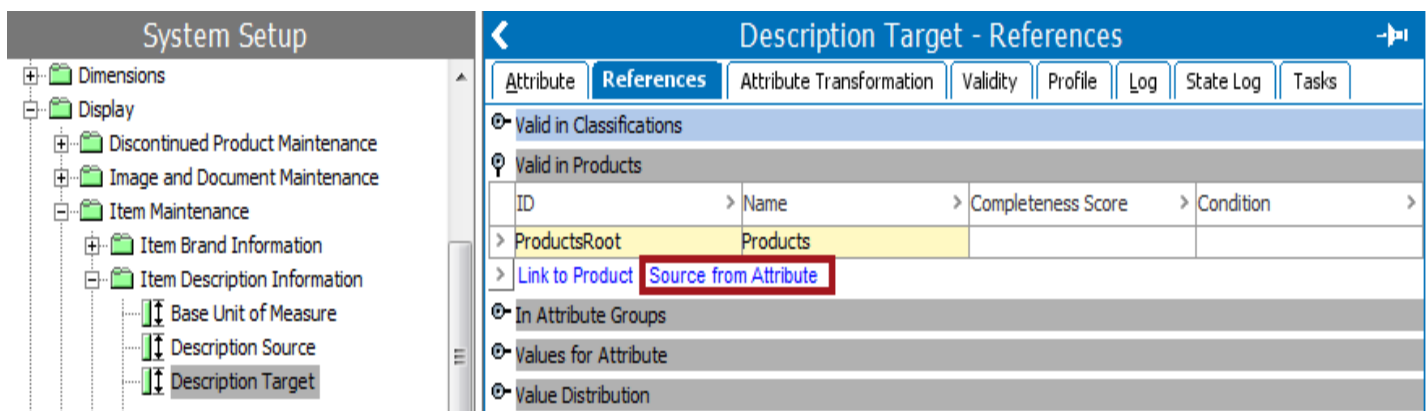
Note: If the attribute is not valid for any classifications, verify that the source attribute is also not valid for classifications.



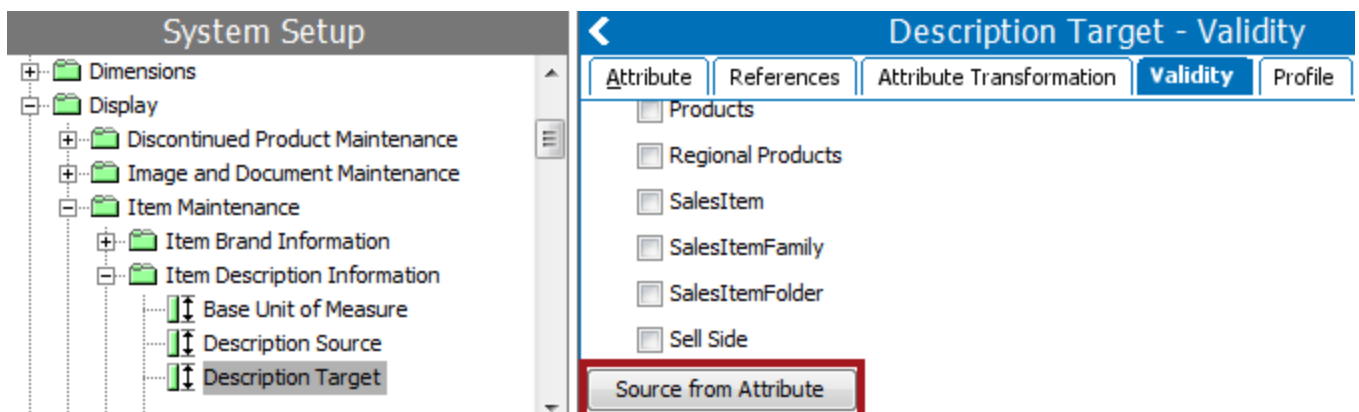
3. Use Browse or Search to find the source attribute and click **Select**. The classification links of the source attribute are copied to the target attribute.



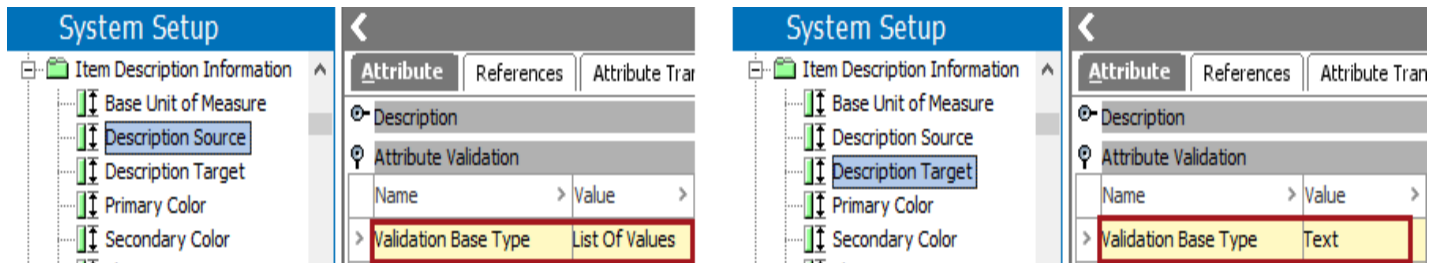
- On the References tab open the 'Valid in Products' flipper (if it exists), and click the **Source from Attribute** link.



- 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.
- On the Validity tab, under the Valid for Product Types flipper, click the **Source from Attribute** button.



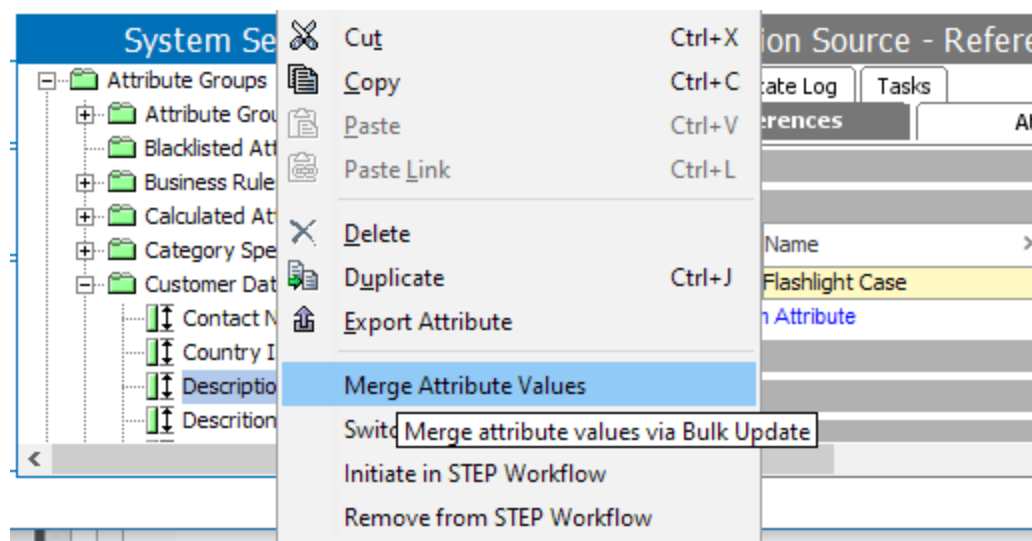
7. Use Browse or Search to find the **Source Attribute** and click Select. The validity of the source attribute is copied to the target attribute.
8. In System Setup, on the Attribute tab under the Attribute Validation flipper, compare the Validation Base Type for the **Target Attribute** and the **Source Attribute**.



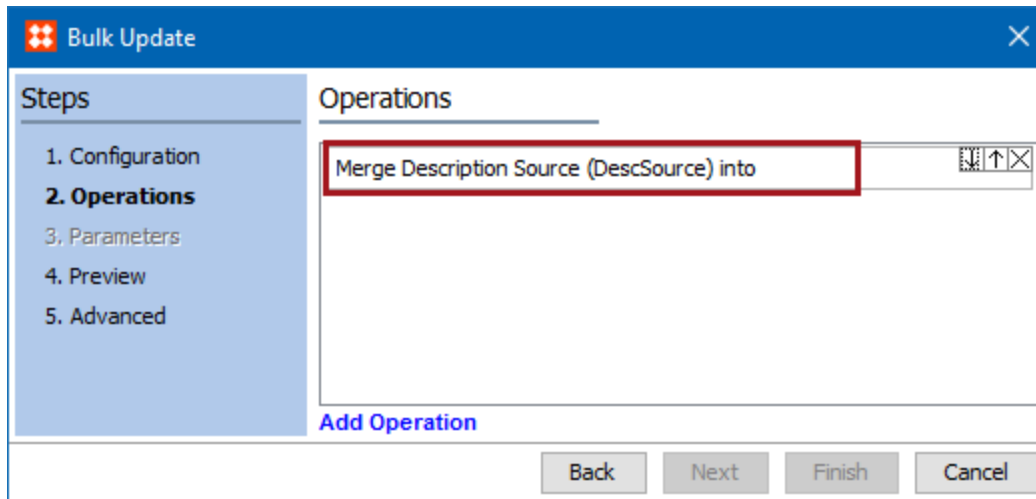
- If the Validation Base Types are the same and are not LOVs, continue to **Merge Values Using Bulk Update**.
- If the Validation Base Types are not the same, determine if they are compatible by manually adding an existing source attribute value to the target attribute successfully. If the manual test 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 a 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 **Editing 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	Icons examples	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	 Represented by three dots at the bottom	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.

Unlinking 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 / unlinked from an Object.

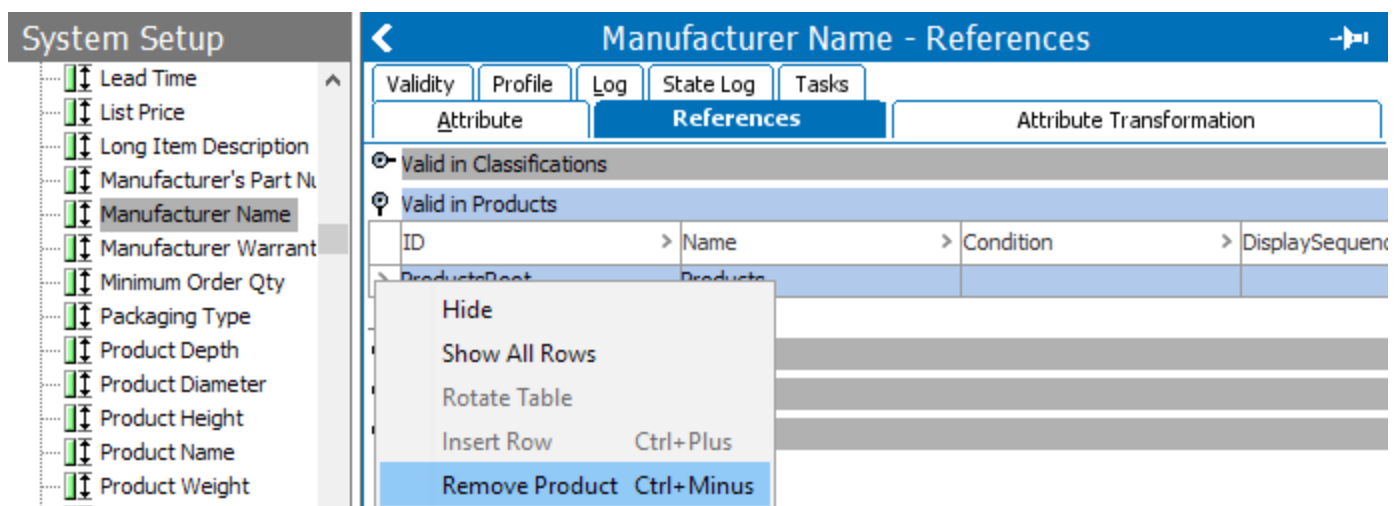
If an attribute with a value is deleted, or unlinked, from a product node, the attribute becomes an 'Orphan Attribute', and will keep values for the attribute. If the attribute is linked to the product again, the value will also appear again.

Attributes can be unlinked from objects in 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.

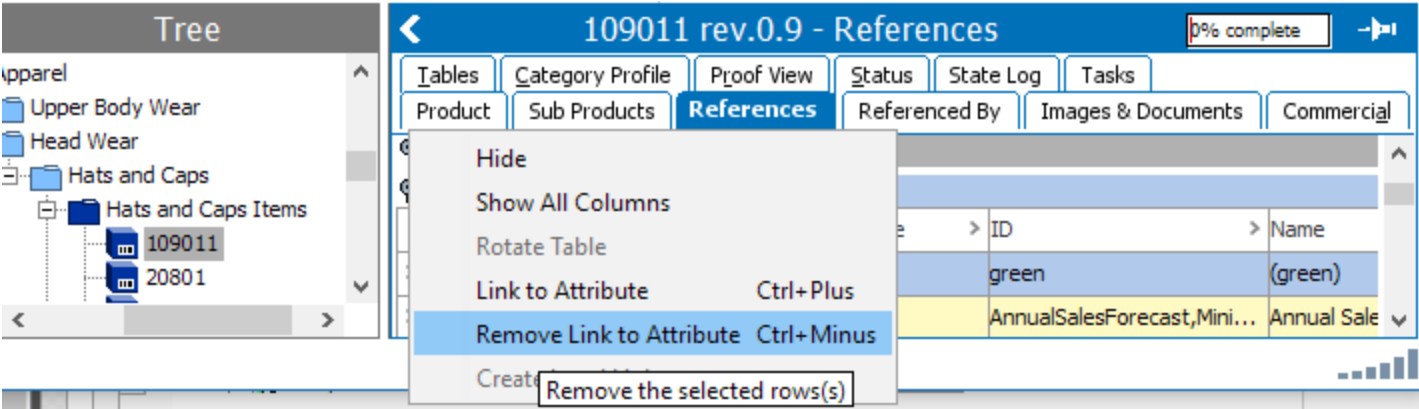
Unlink Object from Attribute References Tab

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**.



Unlink 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 unlinked from an object. If the object has any linked products, the attribute will also disappear from those. If the unlinked 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 unlinking an attribute from an object, the user must make sure that the attribute is 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.

Specification attributes can be found under their respective attribute group's flipper, and descriptive attributes can be found under the 'Description' flipper at the top of the value editor.

Attribute values can be edited either by selecting from a list, typing a value, or overwriting an existing value.

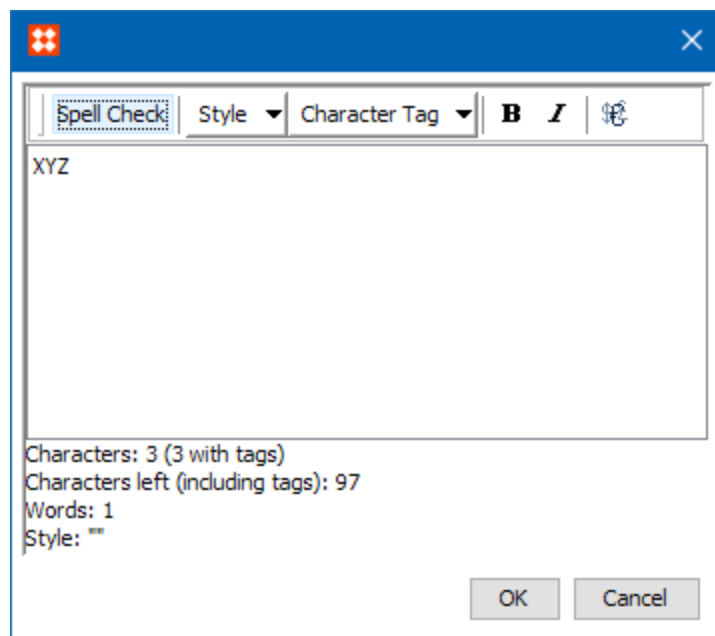
1. In **Tree**, select the relevant node to open the editor.
2. Depending on the object type, click the **Product** tab, **Classification** tab or **Images & Documents** tab.
3. Double click the value field of the relevant attribute and enter the new value.

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, IOS Date and Time or Numbers with units, and so on).

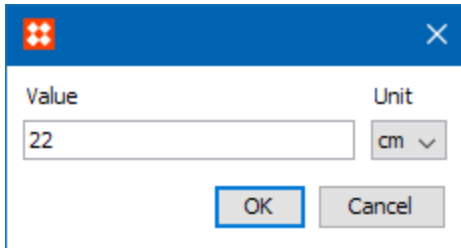
Text attribute editor:



In addition to addition values to the attribute in the editor, the below operations can also be performed:

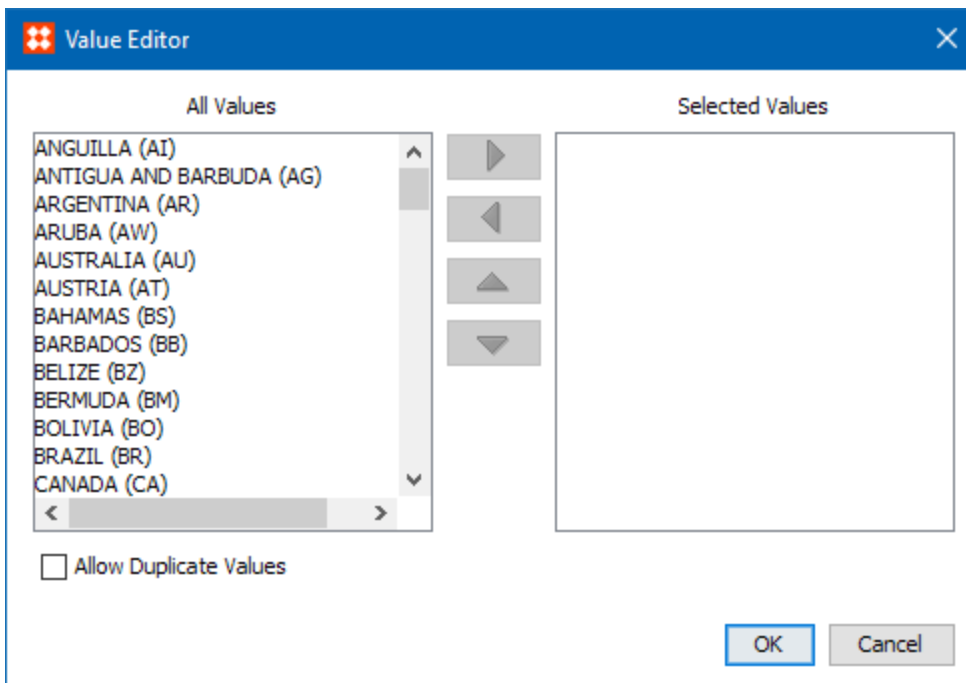
- Perform Spell Check on the values being entered
- Apply style tags to the text. For more information see the **Tags** topic in the **System Setup / Super User** documentation.
- Apply character styles to the text. For more information see the **Tags** topic in the **System Setup / Super User** documentation.
- Apply basic formatting to the text

If it was a 'Number' validation attribute, the editor would look like the picture below:



A small dialog box with a blue header bar containing a red cross icon and a close button. The main area has two labels: 'Value' and 'Unit'. Below 'Value' is a text input field containing the number '22'. Below 'Unit' is a dropdown menu showing 'cm' with a downward arrow. At the bottom are two buttons: 'OK' and 'Cancel'.

For a multi-valued or LOV attribute, this would be the editor:



A larger dialog box titled 'Value Editor' with a blue header bar containing a red cross icon and a close button. The main area is divided into two sections: 'All Values' on the left and 'Selected Values' on the right. The 'All Values' section contains a list of country names with their ISO codes: 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), and CANADA (CA). Below the list is a checkbox labeled 'Allow Duplicate Values' which is currently unchecked. Between the two sections are four arrow buttons: a right arrow, a left arrow, an up arrow, and a down arrow. The 'Selected Values' section is an empty rectangular box. At the bottom are two buttons: 'OK' and 'Cancel'.

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.

Note: In a multi-valued attribute with LOV validation base type, if you need the same value to be added more than once, you will have to click on 'Allow Duplicate Values' in the Value Editor.

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. To learn more on how to set filters, see the **Editing Objects in Tree** topic in the **Getting Started / User Guide** documentation.

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 its 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.

To quickly add values to multiple objects, it is possible to multi select products using the Ctrl key on the keyboard and then adding values to the products in the Mult Editor.

Tree

Apparel

Upper Body Wear

Head Wear

Hats and Caps

Hats and Caps Items

109011

20801

20805

Red Baseball Cap

Hats and Caps Sales Items

Footwear

Safety

Hardware

Displays

Furniture

Automotive

Building Products

Electrical & Electronics

Food and Beverage

Products

References

Referenced By

View: Show all

Name	Long Item Description	Manufacturer's Part Number	Manufacturer Name
109011	Baseball style cap, adjustable write back	20803-03	
20801	Baseball style cap, adjustable write back	20801	
20805	Epuda	20805	
Red Baseball Cap	Epuda	20803-112	Products Galore, Inc.

Editing in 'ISO Data' and 'ISO Date and Time' validation base type

To edit in the data / time selector, double click on the attribute value to get the data selector to make selections. This applies for both 'ISO Date' and 'ISO Date and Time' validation base types. Below is an example of an ISO Date and Time data selector.

Date Picker

Year: 2014

Month: oct

Day: 01

Hours: 11

Minutes: 41

Seconds: 23

OK

Cancel

Entering Values

There are different ways of entering values according to the validation base type of the Attribute:

1. 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.

2. Values *can* be in a list of values (LOV). 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.

3. Values are 'free'. Option is:

- 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 **System Setup / Super User Guide** 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

• Specification Attributes:

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.

For 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.

• Description Attributes:

For descriptive attributes, a value may be used and saved on the top node, and could be set to not be shown or allowed to be inherit on any children.

See the **Validation Rules** in the **Attributes** documentation for more information.

Editing Multi-Valued Attributes

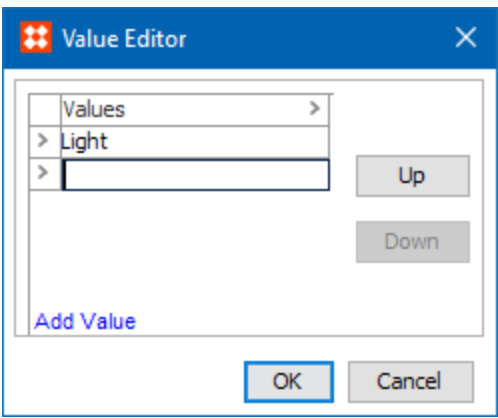
To edit the value of an attribute on an object, navigate to the desired object in **Tree** and select the value field of the relevant attribute.

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

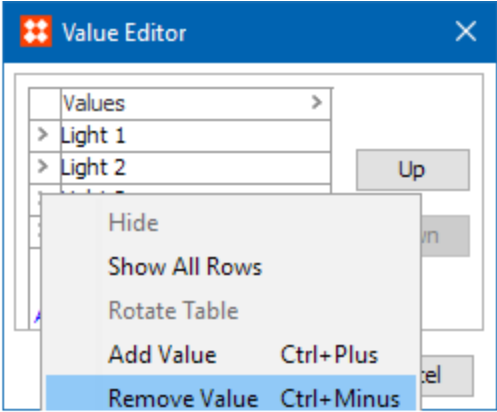
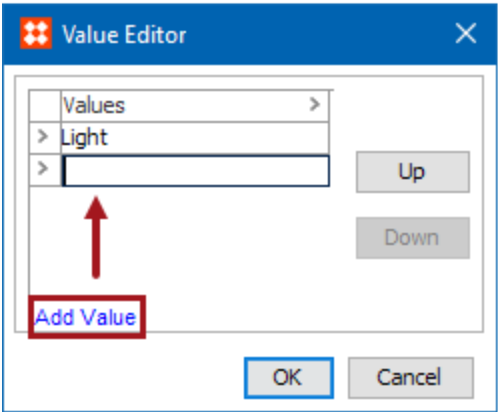
Note: Multi-valued attributes are available for specification and description attributes. Multi-valued attributes are represented by three dots at the bottom of the icon, for example: 

- 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.

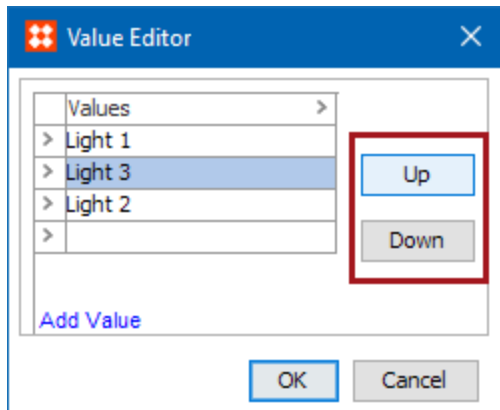
A **Value Editor** dialog box appears.



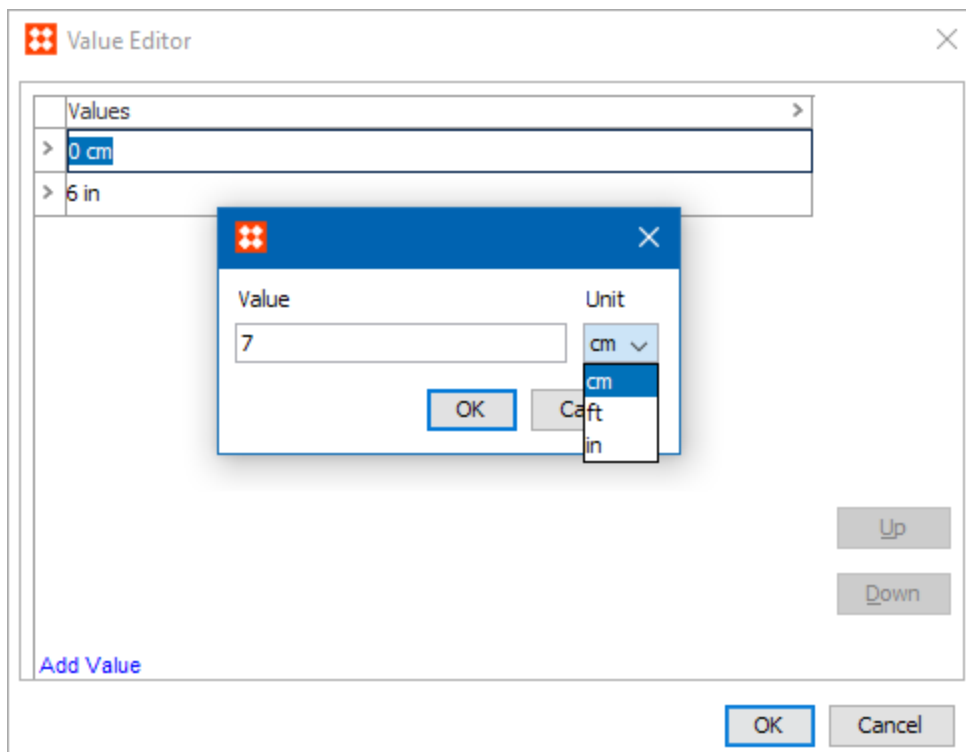
- 4. Click **Add Value** to add a new line, or click into the line to change the value. To delete a value, highlight it right click on the arrow and press 'Remove Value.'



- 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.



- Click **OK** when finished. 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. If the attribute has units applied, the user can either key in the relevant units, or press <Alt> + <F2> to select a unit from the list. The example below is with a number validation base type with units applied.



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 Attribute Facts:

- Specification attributes are only available to products (compared to description attributes, which also are available to classifications, images, documents and link types).
- Specification attributes and their values are always inherited by child objects in the same hierarchy.
- Specification Attributes are always displayed under their 'Attribute Group Flippers' as which appear under the Description.

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

Tree

Products

Apparel

Footwear

Safety

Hardware

Tools

Task Lighting

Flashlights

Flashlights Items

111204 Child of Over

114859

114861

20862

20883

Flashlight Case

Flashlight Clip

LED Pocket Flashlight

LIGHTS-10

Flashlights SalesItems

Displays

Furniture

Automotive

Building Products

Electrical & Electronics

Food and Beverage

Flashlight Clip rev.0.4 - Product

0% complete

Images & DocumentsCommercialTablesCategory ProfileProof ViewStatusState LogTasks

Product

Sub Products

References

Referenced By

Description

Name	Value
ID	114855
Name	Flashlight Clip
Object Type	Item
Revision	0.4 Last edited by USERL on Fri Apr 07 15:33:46 EDT 2017
Approved	Never Been Approved
Translation	Not Translated
Category	Primary Product Hierarchy Products Hardware Tools Task Lighting Flashlights Flashlights Items Flashlight Clip Flashlight Clip
Parent	Flashlights Items
Path	Hardware Tools Task Lighting Flashlights Flashlights Items

No Primary Image

Manufacturer Information

Name	Value
Manufacturer's Part Number	9
Manufacturer Name	Flashlight Clips Inc.
Country of Origin	ARGENTINA
Product Name	

Category Specific Attributes

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. Without Strict Validation, the stored values are not guaranteed to be searchable.

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
GLN		Validates the length and check digit of the thirteen digit GLN.	1111111111116
GTIN		Validates the lengths check digits for multiple GTIN types. For instance, if an attribute is valid for either an EAN-8 or an EAN-13, this attribute can be configured to validate against either of them. By default, the GTIN Validation Base Type will validate the GTIN-8, 12, 13 and 14 digit numbers. If the requirement is to validate only the GTIN-8 and 13 digit numbers and not the GTIN-12 and 14 digit numbers, an input mask will have to be set so that 12 and 14 digit number are not accepted. For more information please see	22222220 or 44444444444444

Validation Base Type	Icons	Description	Example
		the Input Masks section.	
GTIN-8		Validates the length and check digit of the eight digit GTIN.	22222220
GTIN-12		Validates the length and check digit of the twelve digit GTIN.	333333333331
GTIN-13		Validates the length and check digit of the thirteen digit GTIN.	44444444444444
GTIN-14		Validates the length and check digit of the fourteen digit GTIN.	55555555555555
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
LOV		LOV must be selected containing the valid values.	An LOV (List Of Values) will contain the valid values of an

Validation Base Type	Icons	Description	Example
			attribute, e.g.: • Red • Green • Blue
Number	123	Allows any numeric characters.	Valid values: 78, 1, .34, and 384.5. Invalid values: Fractions such as 50 1/2
Number Range	1-23	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. Web UI users and those using the setSimpleValue() method in the business rules API: If using units and a user types a value in the field that matches a unit, but the user omits typing an explicit unit, then the text in the field becomes the value and the unit becomes null. If the attribute has a default unit, that unit is appended even if it matches the value string. The only exception to this behavior is if the text ends with a digit followed by a matching unit, e.g., '5m.' Examples to explain the concept described above are shown below (with 'm' being a valid unit and 'cm' being the default unit): • '5' becomes '5 cm' (value: '5', unit: 'cm') • '5m' becomes '5 m' (value: '5', unit: 'm') • '5 m' becomes '5 m' (value: '5', unit: 'm') • 'medium' becomes 'medium cm' (value: 'medium', unit: 'cm') • 'medium m' becomes 'medium m' (value: 'medium', unit: 'm') • 'm' becomes 'm cm' (value: 'm', unit: 'cm')	Valid values: Converter 20 bit, Cable 2.2 mm

Validation Base Type	Icons	Description	Example
		'cm' becomes 'cm cm' (value: 'cm', unit: 'cm')	
Regular Expression		Allows for the applying of regular expressions as a validation base type. For more information, see the Regular Expression topic in System Setup / Super User Guide documentation. Regular expressions used as validation base type can be stored as a Validation Template, allowing you to create an expression once and then apply it to multiple attributes by selecting the template from the Validation Base Type dropdown. For more information, see the Validation Templates topic under the System Settings section of Users and Groups documentation in this guide.	[a-z]{3} ([0-9]{3})
Text		All characters (numeric and alpha) are valid.	
URL		Allows applying a URL.	
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. For more information about creating Validation Templates, see the Validation Templates section of System Settings.	

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.

Icons used for single valued attributes	Description
	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.

For an attribute, open the Attribute Editor and use the Edit Validation Rule link on the Attribute tab.

System Setup

- GDSN Attributes
 - Brand
 - Depth
 - Description Long
 - EAN
 - GDSNgrossWeight
 - GDSNHeight
 - GDSNIsBaseUnit
 - GDSNIsInvoiceUnit
 - GDSNIsOrderableUnit
 - GDSNWidth
 - GTIN
 - GTIN0
 - GTIN12
 - GTIN13
 - GTIN8
 - GTIN Name
 - Input Mask**
 - Qty of Next Higher Package
 - Qty Of Next Lower Package
 - Regular expression
 - UPC
- GDSN Receiver System Attributes
- GDSN System Attributes
- GDSN Validation Group
- IDOC
- Locale
- Manual Sequencing
- Mass Creation
- Match_and_Merge_Attribute_Group

Input Mask - Attribute

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

Description

Name	Value
ID	Input Mask
Name	Input Mask
Last edited by	2017-03-28 12:28:06 by USERA
Full Text Indexable	No
Externally Maintained	Yes
Hierarchical Filtering	None
Calculated	No
Type	Specification
Dimension Dependencies	
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](#)

Aspects

For a List of Values / LOVs, edit directly on the List of Values tab.

System Setup

- Lists of Values / LOVs
 - Boards List Of Values
 - CMDM
 - Computers
 - Customer MDM LOVs
 - eClass LOV Group
 - e-Signature
 - ETIM6 List Of Values
 - ETIM List Of Values
 - GDSN List of Values
 - GDSN Receiver List of Values
 - Item Creation Workflow
 - Match_and_Merge_LOV_Group
 - Print Publishing LOVs
 - ProductVariants
 - Sales Item Creation Workflow
 - Battery Type
 - Battery Type AV
 - Color
 - CountryISOCODE
 - CountryLOV
 - DimensionDependentLOV
 - Harmonization Code
 - Indoor/Outdoor
 - Item Status

Indoor/Outdoor - List of Values

List of Values
References
Log
State Log
Tasks

Description

Name	Value
ID	IndoorOutdoorLOV
Name	Indoor/Outdoor
Edited by	2015-02-13 11:36:33 by STEPSYS
Path	Lists of Values / LOVs/Indoor/Outdoor
Dimension Dependencies	Language;
Use Ids on values	Yes
Use Ids for sorting	No
Value-ID Pattern	

In Attribute Groups

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

Sometimes it is 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 Lists of Values are merged into one. Prior to merging, it would be necessary for the Attribute or Lists of Values to share the same validation rules. The validation rules are explained in the **Validation Rules** section. 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.

For a non-LOV attribute, the following changes are allowed, although some restrictions apply as noted below:

- Text can be changed to Numeric Text
- Text can be changed to ISO Date or ISO Date and Time only if all current values conform to the ISO Date or ISO Date and Time format
- ISO Date or ISO Date and Time can be changed to Text
- Number or Integer can be changed to Text only if no units are involved
- Number or Integer can be changed to Numeric Text

- Fraction can be changed to Text only if no units are involved
- Fraction can be changed to Numeric Text
- Integer can be changed to Number

For a non-LOV attribute, the following changes are not allowed if the attribute already has values:

- ISO Date cannot be changed to ISO Date and Time when values already exist
- ISO Date and Time cannot be changed to ISO Date when values already exist
- Any validation base type with units cannot be changed to Text if the attribute already has values
- Fraction cannot be changed to Number if any fractional value has a hyphen (-) or slash (/)

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**.

The screenshot displays the 'System Setup' window with a tree view on the left and a main configuration area on the right. The tree view shows 'GDSN Attributes' expanded, with 'Input Mask' selected (indicated by a red circle 1). The main area shows the 'Input Mask - Attribute' configuration (indicated by a red circle 2). The 'Attribute' tab is active, showing a table of attributes with 'ID' and 'Name' having 'Input Mask' values. The 'Edit Validation Rule' link is visible at the bottom (indicated by a red circle 3). An 'Edit Validation' dialog box is open, showing the 'Mask' field set to '00000-0000' (indicated by a red circle 4). The dialog also shows other validation settings like 'Validation Base Type' (Number), 'List Of Values' (N/A), and 'Maximum Length' (N/A).

Name	Value
ID	Input Mask
Name	Input Mask
Last edited by	2017-02-23 08:00
Externally Maintained	No
Type	Specification

Name	Value
Validation Base Type	Number
List Of Values	N/A
Multi Valued	No
Mask	00000-0000
Minimum Value	
Maximum Value	
Maximum Length	N/A

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 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)
A GTIN attribute needs to be configured to allow for more than one type of GTIN.	An example might be to allow for the EAN-8 or the EAN-13 to be entered. An example for this would be: 00000000 00000000000000 This mask would allow for an eight digit or a 13 digit number to be entered.

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

Mask	Description	Input Mask Example
		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
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: aaaaa 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: AAAAA Valid values: abcde, a1b2c, 12345 Invalid values: B1, ab de, 45%

Mask	Description	Input Mask Example
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.


Edit Validation



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:

- Attributes
- Product Hierarchies
- Product and Classification Hierarchies

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.

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
- Derived Events
- Object Types & Structures
- Tags
- Units
- Users & Groups
- Reference Types
- Workspaces
- Table

Color, Interior - Attribute

Attribute
References
Attribute Transformation
Validity
Profile
Log
State Log
Tasks

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](#)

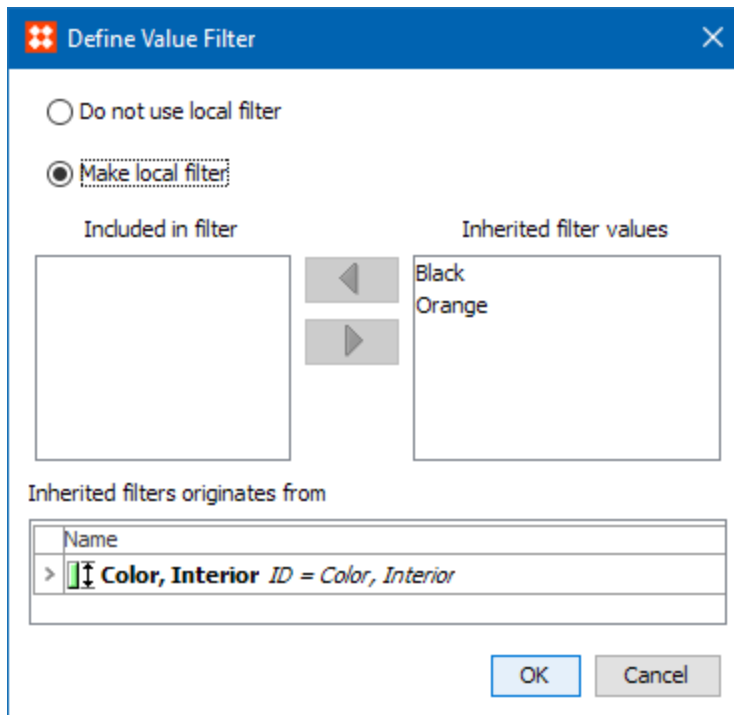
Aspects

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.



The **Define Value Filter** dialog box has a blue title bar with a close button. It contains two radio buttons: **Do not use local filter** (unselected) and **Make local filter** (selected). Below the radio buttons are two list boxes. The left list box is labeled **Included in filter** and is currently empty. The right list box is labeled **Inherited filter values** and contains the text "Black" and "Orange". Between the two list boxes are two arrow buttons: a left-pointing arrow and a right-pointing arrow. Below the list boxes is a section labeled **Inherited filters originates from** containing a table with one row:

Name
>  Color, Interior ID = Color, Interior

. At the bottom right are **OK** and **Cancel** buttons.

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

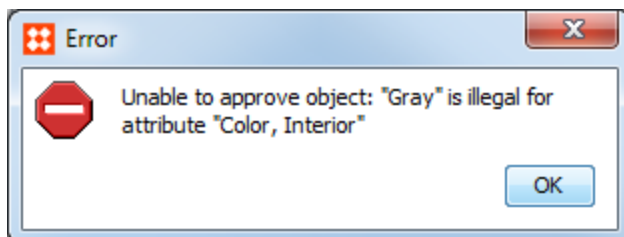
Name	>	>	Value
> ID			SKU 00001
> Name			SKU 00001
> Object Type			Product
> Revision			0.1 Last edited by USER on Tue Aug 11 12:00:43 EDT 2015
> Approved			✖ Last Approved on Thu Aug 06 13:49:17 EDT 2015
> Translation			Not Translated
> Path			Primary Product Hierarchy/Regional Products/Apparel/Winter/Winter 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 for all products can be viewed in the Attribute Editor on the References tab.

Please note that, if the attribute is valid for a lot of products across hierarchies, it may take some time to populate the values.

System Setup

Attribute Groups

Attribute Group

- (green)
- Attribute D
- Attribute N
- AttrSize
- AttrTShirtSize
- blue
- brown
- Color, Interior
- Condition
- Hazmat
- Hazmat Y N
- Marketing Description
- Reference Attribute
- Released by
- Selling Price
- Shipping Restrictions
- Size
- Alternate Supplier Item
- Hyperlinks
- Style Tags

Size - References

ProfileLogState LogTasks

AttributeReferencesAttribute TransformationValidity

Valid in Classifications

Valid in Products

In Attribute Groups

Values for Attribute

Type	Name	Value
Sales Item	B	M
Product-override	104061-Override	Medium
Item	Red Baseball Cap	8
Item	18213 M O	Medium
Sales Item	20808-012	Medium
Sales Item	D	XL
Sales Item	18207-012	Medium
Sales Item	C	L
Sales Item	1112121	Small
Sales Item	A	S

Value Distribution

Viewing the values of an Attribute means getting a list of the objects' names 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.

To make edits to values, follow the instructions below to open the corresponding Editors:

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

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.

Filters can be added to the **Value of Attributes** flipper on any column. To filter, right click on any column header and select **Filter**. Filter will then appear in all column headers.

The left screenshot shows the 'Values for Attribute' flipper with columns: Type, Name, and Value. A right-click context menu is open over the 'Name' column header, showing options: Hide, Show All Columns, Sort, Rotate Table, and Filter. A tooltip 'Changes filter on the column.' points to the 'Filter' option. The right screenshot shows the same flipper with the 'Filter' option selected for the 'Name' column. A dropdown menu is open, showing filter options: '- All -', '- Empty Values -', '- Non Empty Values -', '- Equals -', '- Begins With -', '- Ends With -', '- Contains -', and '< -'. The 'Name' column header now displays '- All -'.

Type	Name	Value
Item	Red Baseball Cap	8
Item	18213 M O	Medium
Product-overrid	104061-Override	Medium
Sales Item	B	M
Sales Item	20808-012	Medium
Sales Item	D	XL
Sales Item	18207-012	Medium
Sales Item	C	L
Sales Item	1112121	Small
Sales Item	A	S

Additionally, the values can be sorted in different orders. Using filters can help with finding desired attribute values, allowing the user to perform various tasks on those values.

To sort, right click on any column header and select **Sort** to get options.

The screenshot shows the 'Values for Attribute' flipper with columns: Type, Name, and Value. A right-click context menu is open over the 'Name' column header, showing options: Hide, Show All Columns, Sort, Rotate Table, and Filter. The 'Sort' option is selected, and a submenu is open showing options: None, Ascending, and Descending. A tooltip 'Sort current column ascending.' points to the 'Ascending' option. The 'Name' column header now displays 'None'.

Type	Name	Value
Item	Red Baseball Cap	8
Item	18213 M O	Medium
Product-overrid	104061-Override	Medium
Sales Item	B	M
Sales Item	20808-012	Medium
Sales Item	D	XL
Sales Item	18207-012	Medium
Sales Item	C	L
Sales Item	1112121	Small
Sales Item	A	S

Background Process Queues

There are various background process queues that are standard with the STEP system. Queues help organize the various STEP activities for easy access and monitoring, and hold information on scheduled processes, active processes, and ended processes.

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.

Prerequisites

It is expected that anyone dealing with background process queues is first familiar with the **Background Processes Tab** topic within the **STEP Workbench UI** guide.

Default Configuration

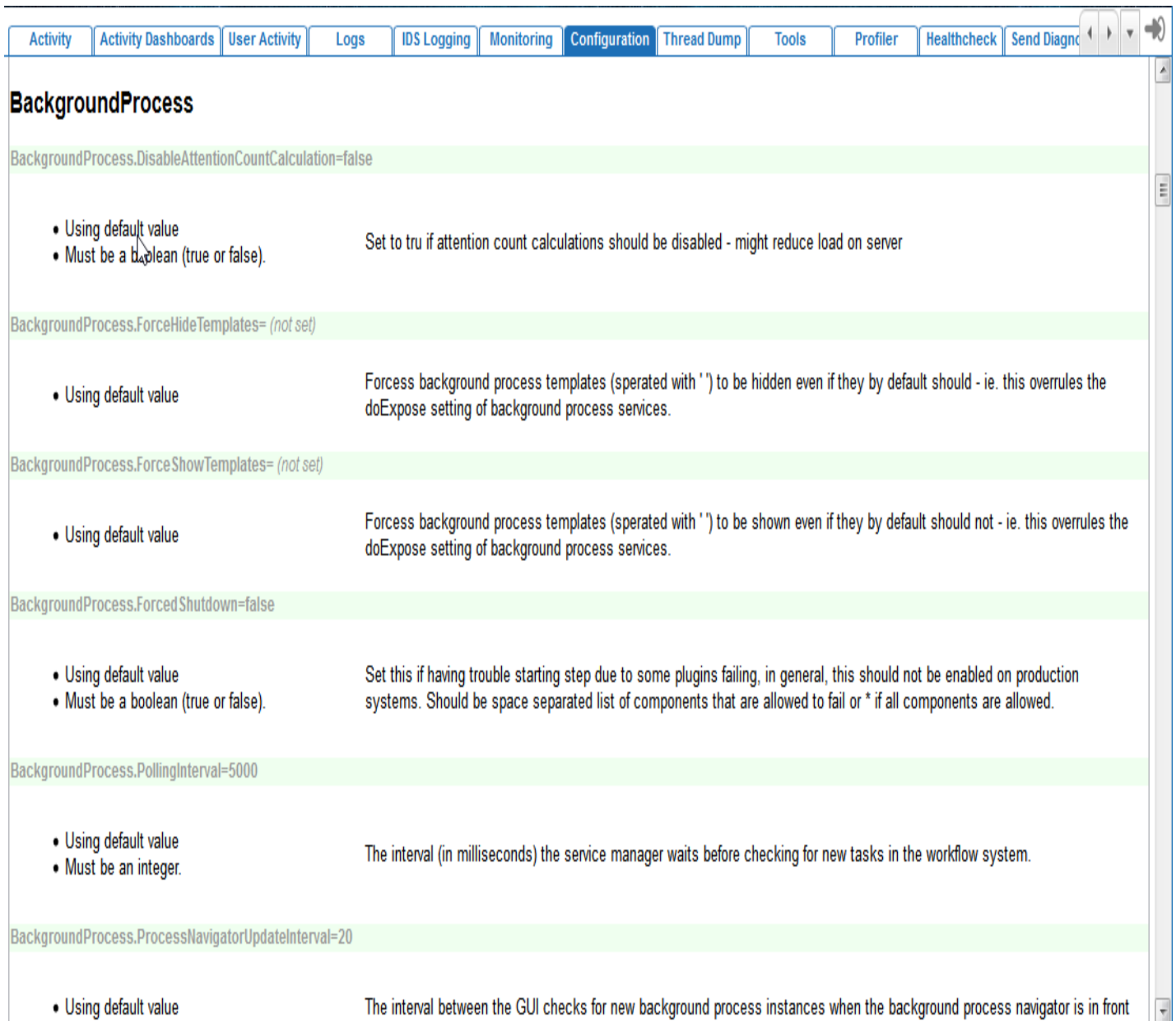
In the default configuration, import processes have their own queue called IN. Export and web publisher processes share a queue called OUT. Scheduled processes have their own queue called SCHEDULE. All other background processes share a queue called MISC.

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. 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. 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.

Below are the STEP background processes properties:

- 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 Stiibo Systems interface. The 'BackgroundProcess' section is expanded, displaying several configuration items with their default values and descriptions.

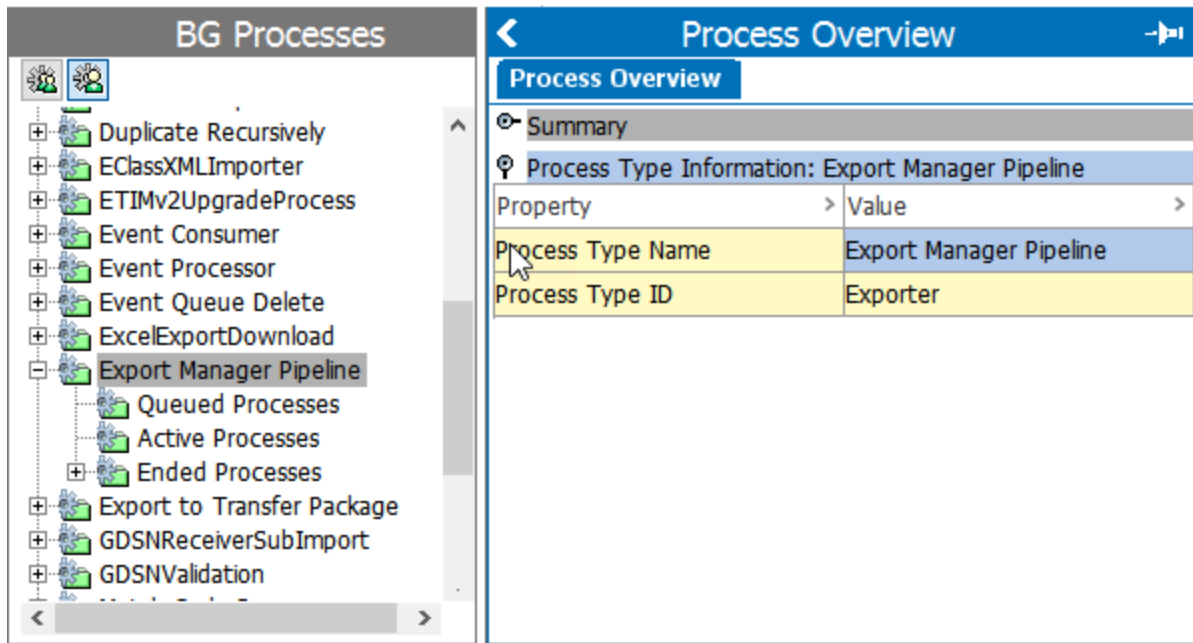
Configuration Item	Default Value	Description
BackgroundProcess.DisableAttentionCountCalculation	false	Set to true if attention count calculations should be disabled - might reduce load on server
BackgroundProcess.ForceHideTemplates	(not set)	Forcess background process templates (seperated with ' ') to be hidden even if they by default should - ie. this overrules the doExpose setting of background process services.
BackgroundProcess.ForceShowTemplates	(not set)	Forcess background process templates (seperated with ' ') to be shown even if they by default should not - ie. this overrules the doExpose setting of background process services.
BackgroundProcess.ForcedShutdown	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	The interval (in milliseconds) the service manager waits before checking for new tasks in the workflow system.
BackgroundProcess.ProcessNavigatorUpdateInterval	20	The interval between the GUI checks for new background process instances when the background process navigator is in front

Changing the default configuration

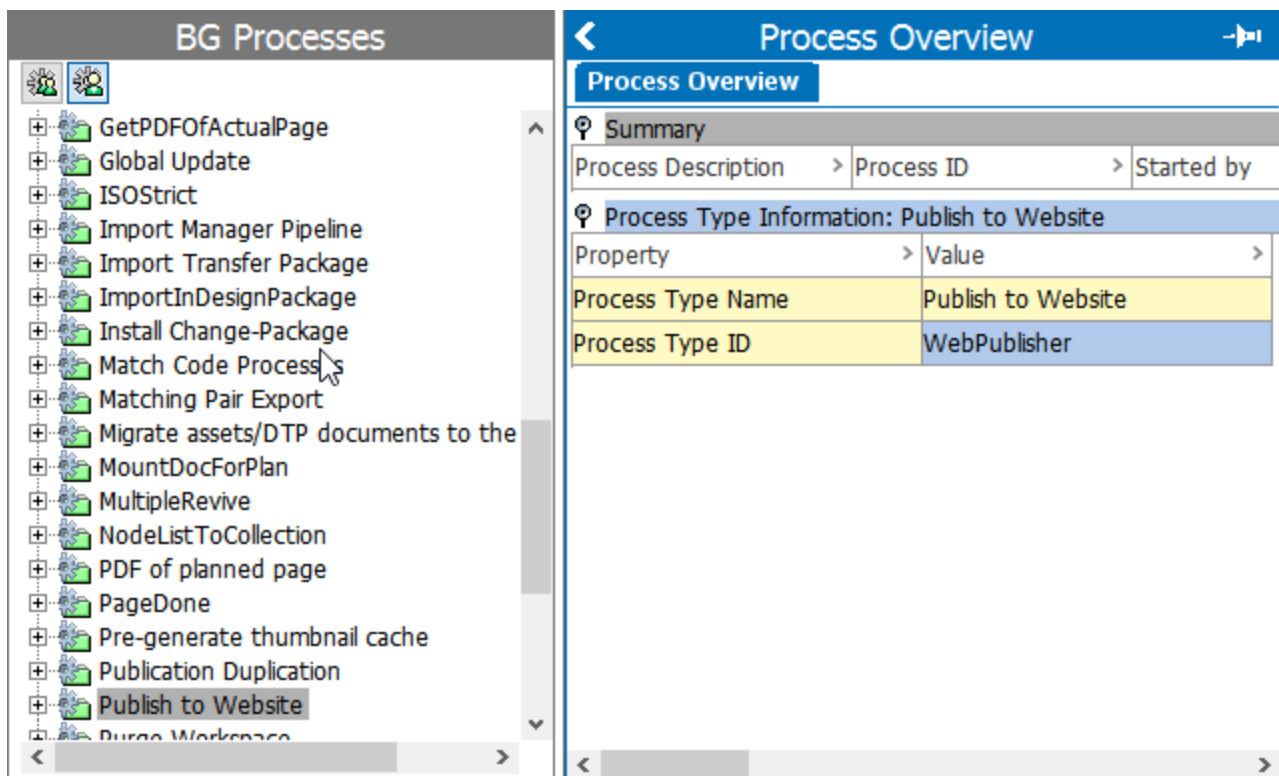
The default configuration can be change by following the steps below. In the example below, the default configuration is changed to allow for the export and web publisher processes to be run on separate queues. Each queue will be configured to run two processes simultaneously.

Finding the Process Type ID

1. In the workbench, click the **BG Processes** tab.
2. Click **Export Manager Pipeline**, and a Process Overview dialog will display.



3. Expand the **Process Type Information: Export Manager Pipeline** flipper.
4. Copy the value of the **Process Type ID**. (For this example 'Exporter' is used.)
5. Paste the value in a temporary text file.
6. Click the **BG Processes** tab, then click **Publish to Website**, and a Process Overview dialog will display.



7. Expand the **Process Type Information: Publish to Website** flipper.
8. Copy the value of the **Process Type ID**. (For this example 'WebPublisher' is used.)
9. Paste the value to a 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.

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.

Deleting Background Processes

Configuring Auto-Deletion of Background Processes

Also part of a default STEP configuration are properties that enable STEP's auto-deletion function for background processes that have ended in the 'Aborted', 'CompletedWithErrors', 'Succeeded', or 'Failed' states.

Retention of Background Processes

These three properties determine how long, measured in hours, background processes will be retained by STEP for export, import, and web publishing processes:

`AutoDeleteBackgroundProcesses.AgeInHours.Exporter = 168`

`AutoDeleteBackgroundProcesses.AgeInHours.Importer = 168`

`AutoDeleteBackgroundProcesses.AgeInHours.WebPublisher = 168`

By default, all ended background processes are set to auto-delete after 168 hours (one week). To make changes to that default, these properties can be edited to direct STEP to retain background processes for less or more time by changing the number value.

Using these properties, the retention period for background processes can be individually adjusted for exports, imports, and web publishing tasks.

Auto-Deletion Timings

Aspects of the auto-delete functionality related to when ended background processes are exposed to the auto-deletion process can be configured to meet the requirements of the specific user.

AutoDeleteBackgroundProcesses

`AutoDeleteBackgroundProcesses.DelayInHours=1`

- Using default value
- Must be an integer.

`DelayInHours`, specifies the frequency of running the delete background processes check.

`AutoDeleteBackgroundProcesses.OperatingHours= (not set)`

- Using default value

`OperatingHours`, specifies the operating hours to run default is all hours. This is specified as a comma separated list of intervals. Eg. 10-16;22-02

`AutoDeleteBackgroundProcesses.OperatingHoursTimeZone= (not set)`

- Using default value

`OperatingHoursTimeZone`, specifies the time zone of `OperatingHours`. Eg. Europe/Berlin or GMT+0200. Default is timezone set on app server

1. `AutoDeleteBackgroundProcesses.DelayInHours=1`

Using this property, users can dictate the frequency at which STEP runs a check to assess which ended background processes are newly subject to auto-deletion. The value must be an integer and, if no property is set, defaults to one [1].

2. `AutoDeleteBackgroundProcesses.OperatingHours`

This property can be used to set the interval(s) during a 24-hour period in which the auto-delete function should be run against ended background processes. This property is often set for users who want the auto-delete functionality to only run during business hours. Using the 24-hour clock, the value for this property should be

set in this format: "[XX]-[YY]", with [XX] being the starting hour and [YY] the ending hour. Multiple intervals can be configured using the same format, separating the hour pairs with a semi-colon instead of a dash, e.g., 10-16;22-02. If this property is not set, the auto-delete functionality is set to the default, which is to run at all hours.

3. AutoDeleteBackgroundProcesses.OperatingHoursTimeZone

If the time zone during which the auto-delete functionality runs should be different than the default time zone set on the app server, then users can configure the appropriate time zone using this property. Using the 24-hour clock, the value for this property should be set in this format: "GMT[+/-][XXYY]", with [XX] being the hour and [YY] being the minutes following (+) or preceding (-) Greenwich Mean Time (GMT), e.g., GMT+0200. For example, if the time zone to be configured using this property is Eastern Standard Time (EST), then the value for this property should be GMT-0400. For Pacific Standard Time (PST), the value would be GMT-0700. If this property is not set, the auto-delete functionality is set to the default, which is to run according to the time zone set on the app sever.

Important: It is important to note that this property should not be set unless the AutoDeleteBackgroundProcesses.OperatingHours property has also been set.

With these properties in place in the sharedconfig.properties file, background processes that have ended in the 'Aborted', 'CompletedWithErrors', 'Succeeded', or 'Failed' states that are older than the number of hours specified in the AutoDeleteBackgroundProcesses.AgeInHours properties will be automatically deleted from STEP. It should be noted that, in this context, deletion means removal not just from the application server but also the file server.

Deleting Background Processes Manually

Background processes may also be deleted manually. To delete a background process, click the **BG Processes** tab and then select the background process to be deleted.

From the Toolbar, click the Maintain dropdown, and then select **Delete**.

The background process is now removed from STEP.

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 or the schedule is completed.

A scheduled process starts in the Queues Processes folder until the scheduled time arrives. Then the process appears in the Active Processes folder until it is completed. The same process again goes into the Scheduled Processes folder until the next scheduled time. The process is moved to Ended Processes after successful completion of schedule or if it is aborted while not running. If a user aborts the process before the schedule is complete, or if there is an error in the BGP, the process moves to the Active Processes folder with red text indicating an error execution.

The screenshot shows the 'BG Processes' editor. On the left, a tree view shows the hierarchy: BG Processes > Scheduled Processes > Queued Processes > 'Export using 'Collection of Hats''. The main panel displays the properties of this process in a table format.

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 Address Component Model, as well the Loqate, CASS, 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

The screenshot displays the 'System Setup' interface. On the left, a tree view shows the hierarchy: 'System Setup' > 'Setup Actions' > 'DataContainers_Privileges_SetUp'. The main panel is titled 'DataContainers_Privileges_SetUp - Action Set'. It contains a table with the following data:

Description	
Name	Value
ID	DataContainers_Privileges_SetUp
Name	DataContainers_Privileges_SetUp
Action Type	Setup Action

Below the table, there is an 'Actions' section with a list of actions:

- Maintain Data Container Type
- View Data Container Type

The 'Maintain Data Container Type' and 'View Data Container Type' actions are highlighted with a red box. An 'Add Action' link is visible below the list.

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

System Setup

- Attribute Groups
- Attribute Transformations
- Action Sets
 - Setup Actions
 - User Actions
 - All User Actions
 - Buyer, Approve
 - Buyer, Classify
 - Buyer, Create
 - Buyer, Delete
 - Buyer, Modify
 - Buyer, View
 - Classify products
 - Commercial consumer privilege
 - Commercial supplier privileges
 - DataContainers_Privileges**
 - Data Steward, Create
 - Data Steward, Modify
 - Data Steward, View

DataContainers_Privileges - Action Set

Action Set Log

Description

Name	Value
ID	DataContainers_Privileges
Name	DataContainers_Privileges
Action Type	User Action

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)

Add Action

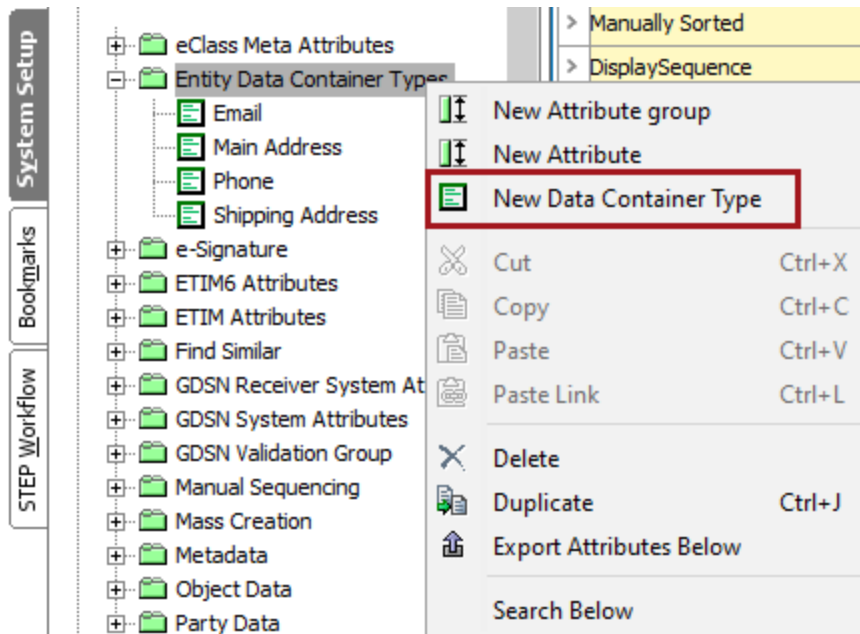
For more information, see the **Action Sets** section of the **System Setup / Super User** documentation.

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

Note: If a data container is made invalid for an entity type, and it had data container instances on entities of that type, those instances will be purged by a background process.

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,' and two or more container instances are created for an entity, it is no longer possible to set it to 'No'.

However, the system can assist the user in identifying entities that have more than one instance so that they may be deleted, and the setting ultimately reverted. If multiple instances of a data container type exist on one entity, attempting to change this configuration will prompt the user with an optional auto search.

Find Objects with Data Container Type Shipping Address 2

? Do you want to search (possibly in multiple workspaces) for objects that contain more than one instance of Shipping Address 2 in order to delete all instances but one?

OK Cancel

After clicking OK, the user is taken to a Search tab with the relevant criteria already filled out.

By referencing the search results, the user can easily identify any entities with multiple instances of the same data container type and delete them. Once all of the required instances have been deleted the user can change the setting to single-valued.

For more information on executing searches, see the **Search Overview** section of the **Getting Started** documentation.

- 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.

Alternatively, attributes can be added to a data container via the attribute's 'Validity' tab, under the 'Valid for Data Container Types' flipper.

System Setup

- Attribute Groups
 - (Stibo Attr grp)
 - Asset Export Configuration
 - Attribute Group
 - Board Essential Attributes
 - Category Specific Attributes
 - Customer Data
 - Address
 - Street
 - City
 - State
 - Zip

<
Attribute
References
Attribute Transformation
Validity

Valid for Data Container Types

- ☐ All
- ☐ Contacts
- ☐ Email
- ☒ Main Address
- ☐ Main Address
- ☐ Phone
- ☐ Satellite JAN

Note: If existing values are found on a data container type when attempting to remove the validity of an attribute, the user will be asked to confirm or reject the deletion of existing values.

- 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](#)

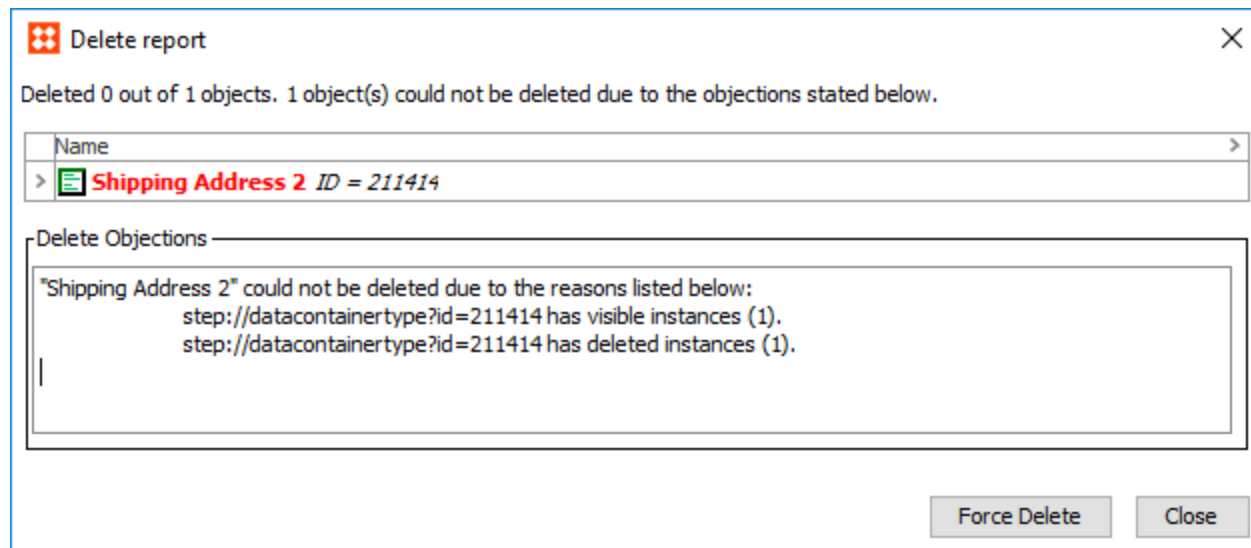
Valid Attributes

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

[Add Attribute](#)

Deleting Data Container Types

Data container types can be force deleted, even if they have visible or deleted instances. Deleting a data container type starts a background process that also purges any associated data container instances.



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

Shipping Address - Validity

Data Container Type **Validity** Log

Valid for Entity Types

- ☐ All
- ☐ Account
- ☐ Address
- ☐ Address Root
- ☐ All Customers
- ☐ Contact
- ☐ Contacts Root
- ☒ Customer
- ☐ Customer Data Root
- ☐ Entity Root
- ☐ GDSN CIC
- ☐ GDSN CICs

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

Shipping Address - Log

Data Container Type Validity **Log**

Showing page 1 of 1

2016-10-10 15:00:12 'USER8': Created : Added to attribute group 'EntityDataContainers'

2016-10-10 15:00:12 'USER8': Name modified from 'null'

2016-10-10 15:33:41 'USER8': Added valid node 'Street'

2016-10-10 15:33:41 'USER8': Added valid node 'City'

2016-10-10 15:33:41 'USER8': Added valid node 'State'

2016-10-10 15:33:41 'USER8': Added valid node 'CountryISOCode'

2016-10-10 15:33:41 'USER8': Added valid node 'Country'

2016-10-10 15:33:41 'USER8': Added valid node 'Zip'

2016-10-10 15:49:15 'USER8': Added valid node 'Address'

2016-10-10 15:49:18 'USER8': Added valid node 'Customer'

2016-10-10 15:49:18 'USER8': Added valid node 'Customer Data Root'

2016-10-10 15:51:57 'USER8': Removed valid node 'Address'

2016-10-10 15:51:58 'USER8': Removed valid node 'Address Root'

2016-10-10 15:51:58 'USER8': Removed valid node 'All Customers'

2016-10-10 15:52:00 'USER8': Removed valid node 'Customer Data Root'

2016-10-10 15:52:01 'USER8': Removed valid node 'Contacts Root'

2016-10-10 15:52:01 'USER8': Removed valid node 'Contact'

2016-10-10 15:52:02 'USER8': Added valid node 'Contact'

2016-10-10 15:52:07 'USER8': Removed valid node 'Contact'

2016-10-13 13:07:40 'USER8': Property modified : Modified validation rule for Multi Valued 'No'

2016-10-27 09:01:00 'USER8': Added valid node 'Address'

2016-10-27 09:01:20 'USER8': Removed valid node 'Address'

First Page Previous Page Next Page Last Page Download...

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 Stibo Systems interface. On the left is the 'Tree' view with a hierarchy: Entity Root, Address Root, Contacts Root, and Customer Hierarchy. Under Customer Hierarchy, 'Customer A' is selected. On the right is the 'Customer A re' record view. It has tabs: Customer, Data Containers (highlighted with a red box), References, and Referenced By. The Data Containers tab shows a table with columns 'Name' and 'Value'. The rows are: ID (CUS_101545), Name (Customer A), Object Type (Customer), Revision (0.5 Last edited by USER8 on Thu), and 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.

This screenshot shows the 'Data Containers' tab for 'Customer A'. It has tabs: Customer, Data Containers (active), References, Referenced By, Status, State Log, and Tasks. There are two sections: 'Main Address' and 'Shipping Address'. Each section has a table with columns 'ID' and 'Value'. Under 'Main Address', there is one row with ID 'CUS_101545' and a value, and a link '> Add Data Container'. Under 'Shipping Address', there are three rows with IDs 'CUS_101545', 'CUS_101546', and 'CUS_101547', and values 'Toronto', 'Paris', and 'United States' respectively. Each row has a link '> Add Data Container'.

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.

This screenshot shows the 'Data Containers' tab for 'Customer D rev.0.3'. It has tabs: Customer, Data Containers (active), References, Referenced By, Status, State Log, and Tasks. There are two sections: 'Main Address' and 'Shipping Address'. Each section has a table with columns 'ID' and 'Value'. Under 'Main Address', there is one row with ID 'CUS_101545' and a value, and a link '> Add Data Container'. Under 'Shipping Address', there are three rows with IDs 'CUS_101545', 'CUS_101546', and 'CUS_101547', and values 'Toronto', 'Paris', and 'United States' respectively. Each row has a link '> Add Data Container'. A dialog box titled 'ID for Data Container' is open, showing a text input field with 'MainAddress' and buttons 'Create' and 'Cancel'.

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

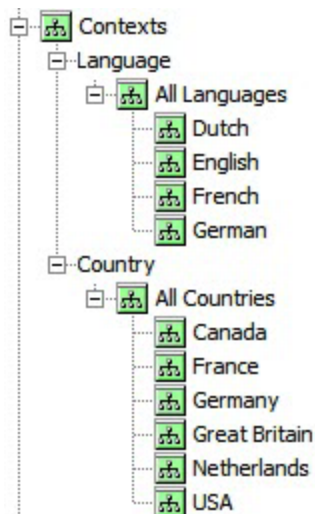
For more information about working with and maintaining data containers in Web UI, see the **Working with Data Containers in Web UI** topic in the **Web User Interfaces** documentation.

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

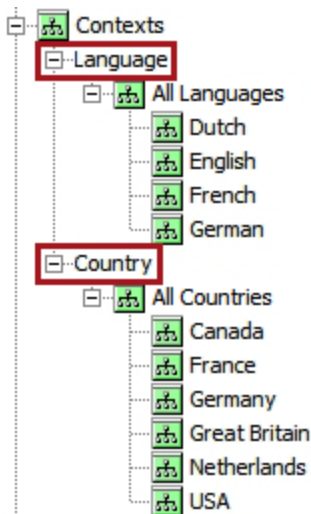
Dimensions and Dimension Points

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

Dimensions

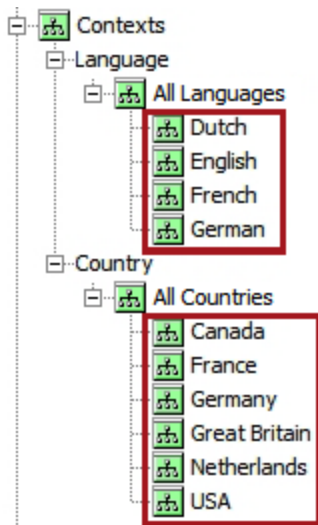
A **dimension** is a definition of a context-sensitive domain in STEP, such as language or country, which allows for different layers of information in which content can vary. A dimension is presented in STEP as a hierarchy of dimension points. Each system must have at least one dimension.

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 **Dimension Dependent 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.

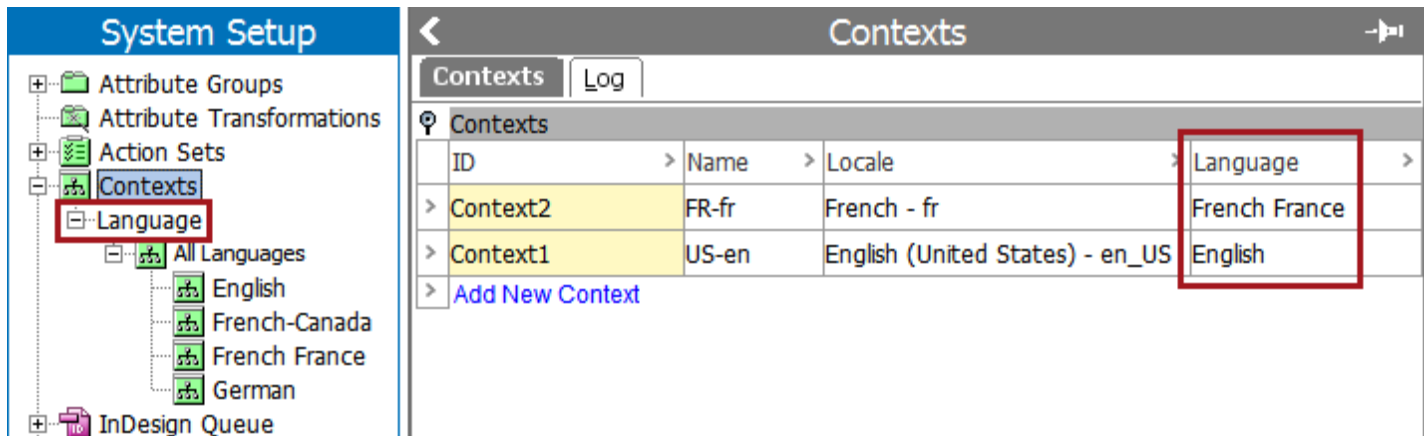
Arguments for Additional Attributes	Arguments for Additional Dimensions
Users can compare values for all versions side-by-side in an Excel file.	Users can compare values for a single attribute across dimensions using the STEP context mode. Select this icon from the workbench to switch to STEP context 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

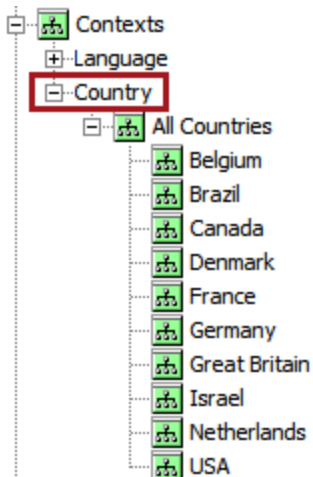
Contexts

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:

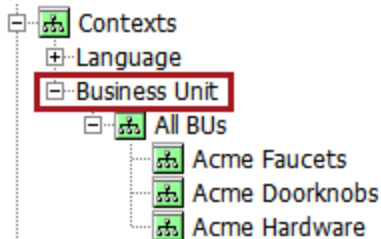
- You have products that are sold in more than one country and there are a *lot* of **attributes** with different values because of the country they are sold in.
- You have products that are sold in more than one country and there are a *lot* of **references** that are different because of the country 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 Dependent 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

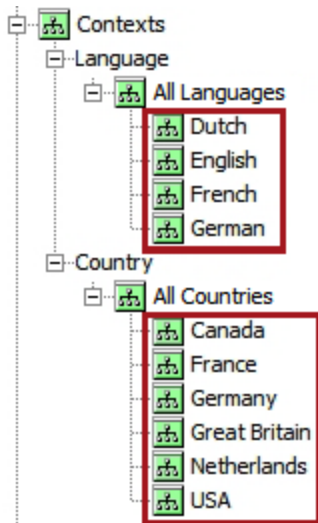
Alternatives to market dimensions are:

- 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

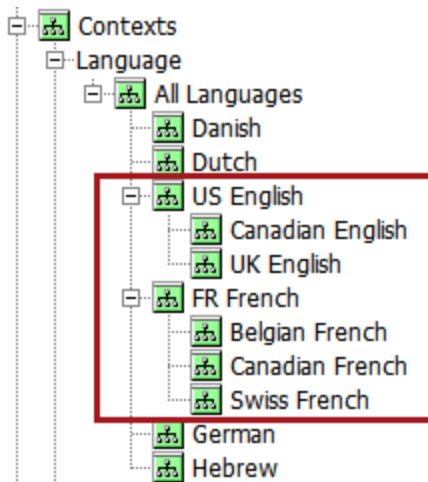
Dimension Point Do's and Don'ts

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.

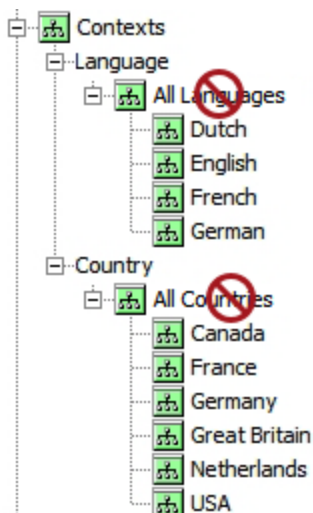


For more information on use cases and the value inheritance of parent / child dimensions and attribute values, see the **Dimension Dependent Attributes** topic.

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

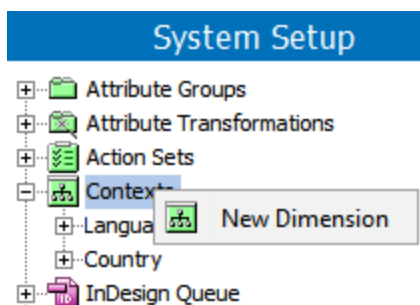
This topic describes how to:

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

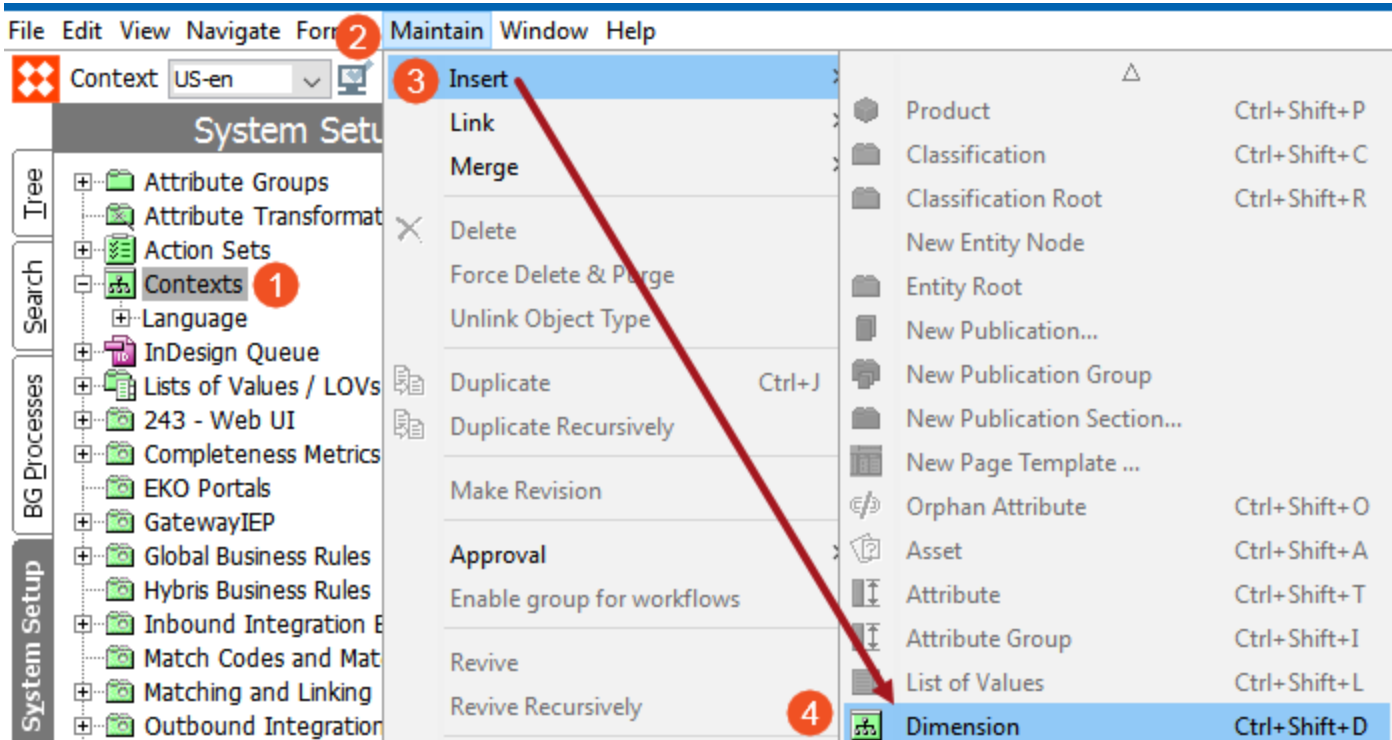
Creating a Dimension

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**.

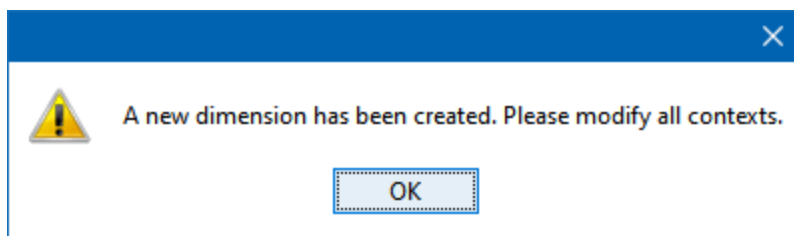


- Alternatively, select Contexts and go to Maintain > Insert > **Dimension**.

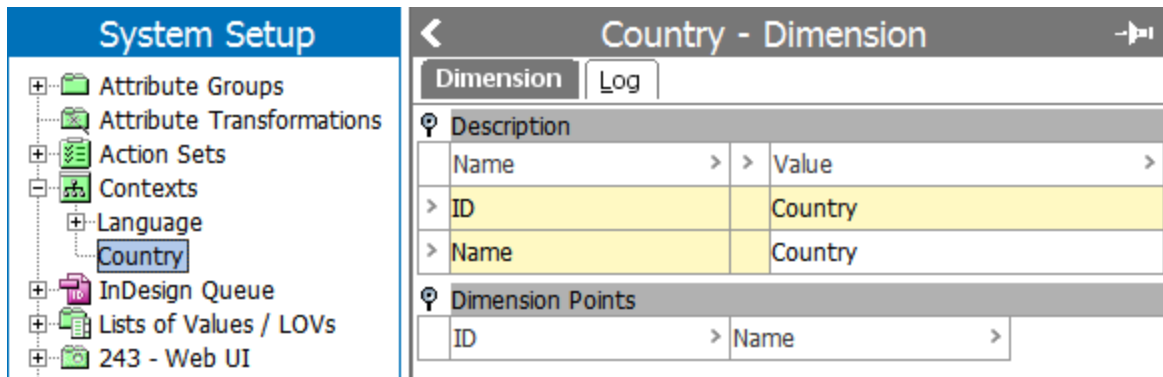


2. In the **Create Dimension** dialog that displays, enter an ID and Name for the new dimension, then click **Create**. If the ID and Name should be identical, press 'Enter' on the keyboard after typing the ID and it will repeat in the 'Name' field.

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.

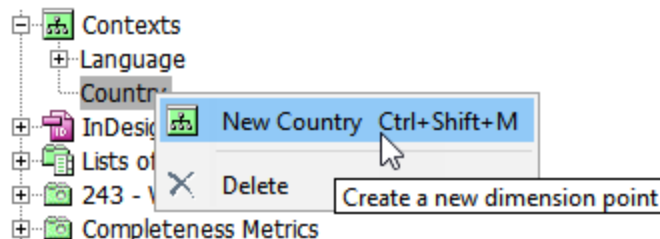


Creating a Dimension Point

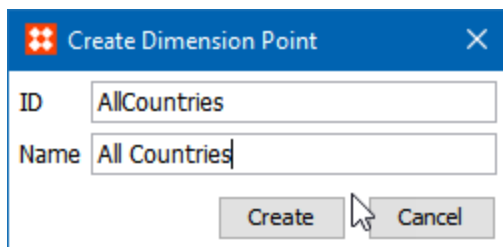
After a new dimension has been created, the next step is to create at least one dimension point below it.

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).

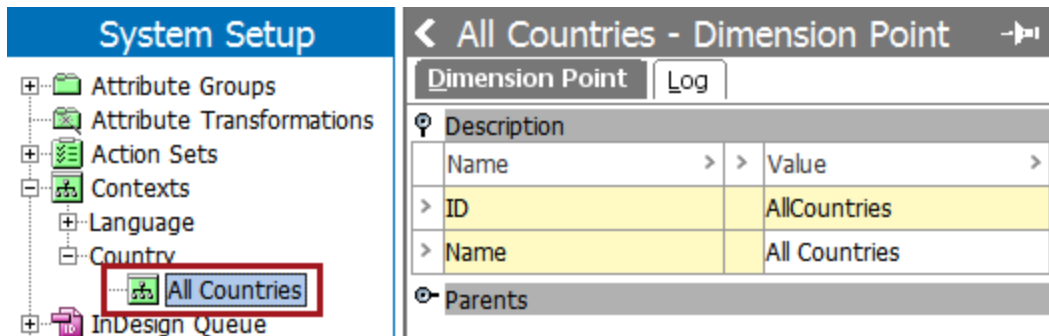
Note: The first dimension point created beneath a dimension must be a top-level **All** level, which is where *non*-dimension-dependent values are stored. Specific dimension points are created below the All level.



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.

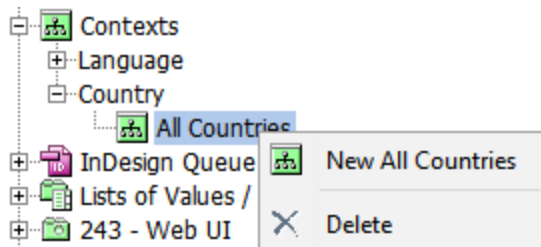


4. The new dimension point is created.

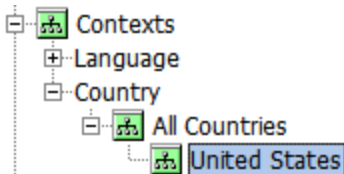


- To create additional dimension points below the All level, select the All level, right click, and click **New** (for example, New All Countries).

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.

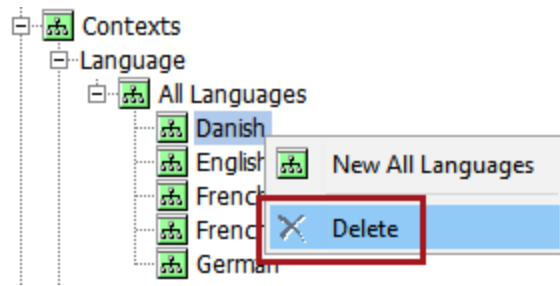


Deleting a Dimension Point

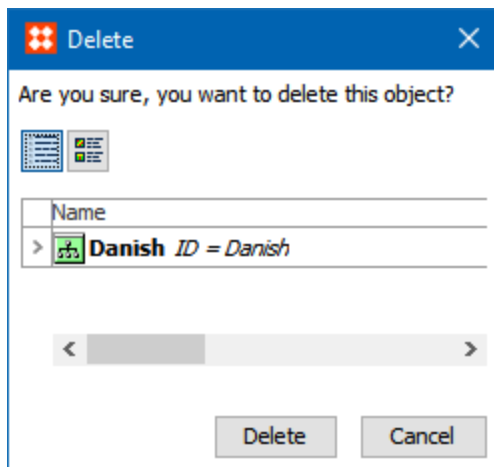
If you need to delete a dimension or dimension point, you must first delete all dimension points that exist below it.

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**.



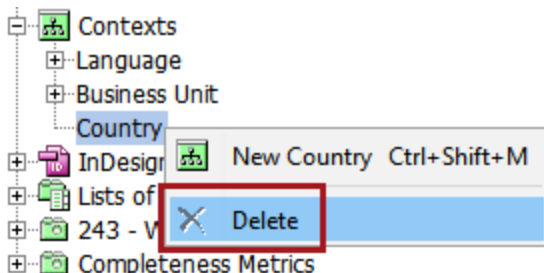
3. In the Delete dialog that displays, click **Delete** to remove the dimension point.



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. Contact your Stibo Systems account manager for more information.

- 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.

System Setup

Attribute Groups

Attribute Transformations

Action Sets

Contexts

Language

All Languages

Danish

Dutch

English

French

German

Country

All Countries

Canada

English - Dimension Point

Dimension PointLog

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.

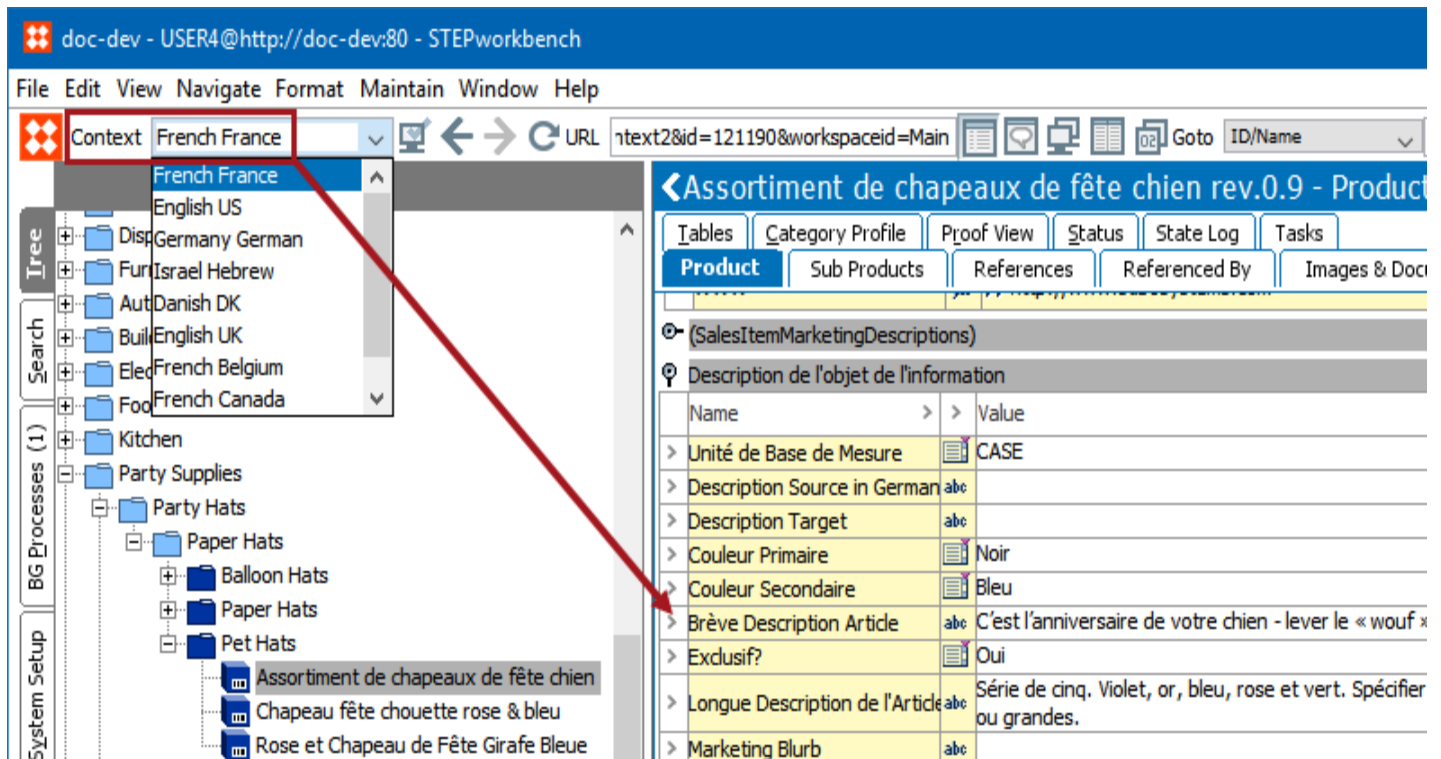
Contexts

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.

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

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).



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 / 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 **Dimension Dependent Reference and Link Types** topic in the **Getting Started / User Guide** documentation.

- **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.

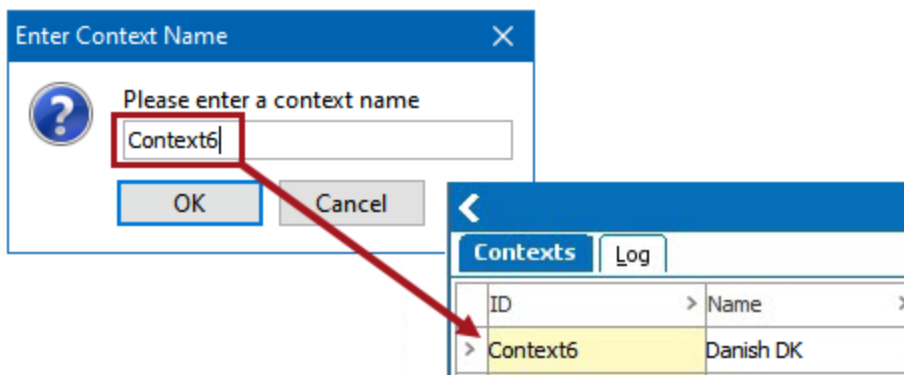
Note: In almost all circumstances, assets can be 'swapped out' in different dimensions by use of dimension-dependent reference types.

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

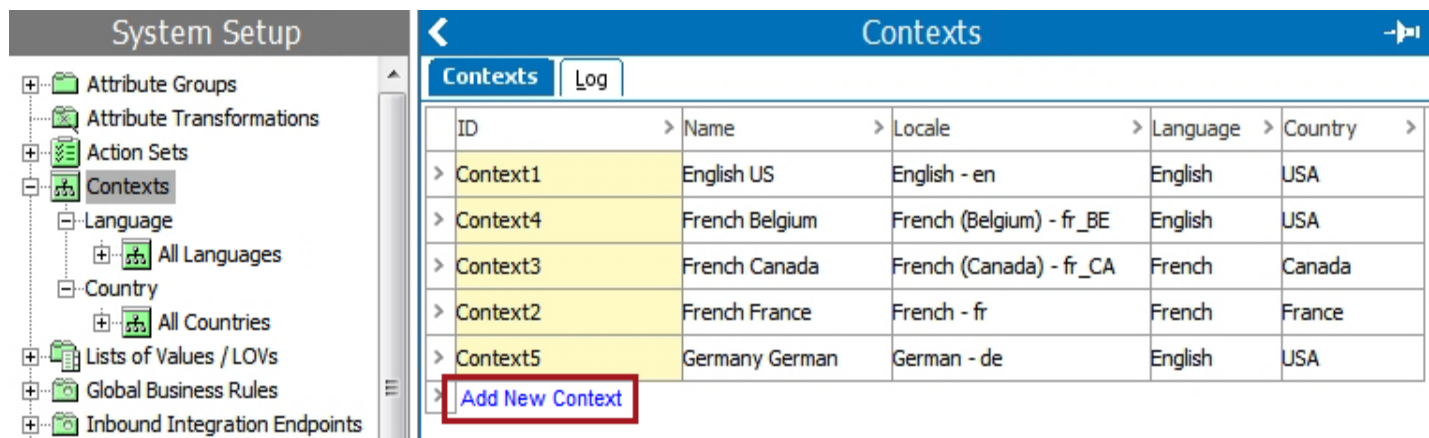
This section describes how to create, edit, and delete contexts. Also explained is the concept of hidden values, which can be caused when a context is deleted, and how to specify a default context.

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.

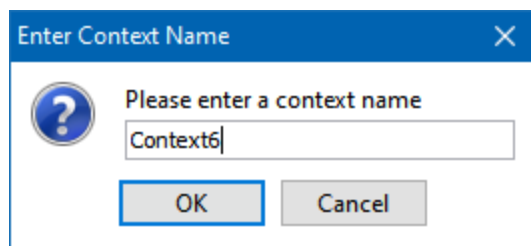
1. Navigate to System Setup > **Contexts** to display the Contexts editor, then click on **Add New Context**.



The screenshot shows the 'System Setup' interface with a sidebar on the left containing a tree view of system setup categories: Attribute Groups, Attribute Transformations, Action Sets, Contexts, Language, All Languages, Country, All Countries, Lists of Values / LOVs, Global Business Rules, and Inbound Integration Endpoints. The 'Contexts' category is selected. The main area displays the 'Contexts' editor with a table of existing contexts and an 'Add New Context' button at the bottom.

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				

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.



The screenshot shows the 'Enter Context Name' dialog box. It has a blue header bar with the title 'Enter Context Name' and a close button (X). Below the header is a question mark icon and the text 'Please enter a context name'. There is a text input field containing 'Context6'. At the bottom are two buttons: 'OK' and 'Cancel'.

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.'

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

Contexts					
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	
> Add New Context			Language Root		
			English		

- 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.
- 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

- 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:

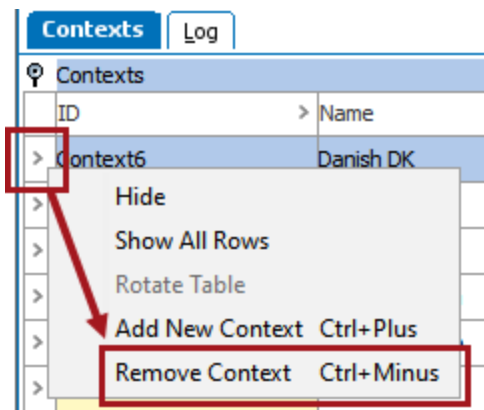
- Enter a new name in the Name field
- Choose a new locale
- Select a new dimension point from the dropdown menu in one of the dimension columns (such as Language or Country)

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.

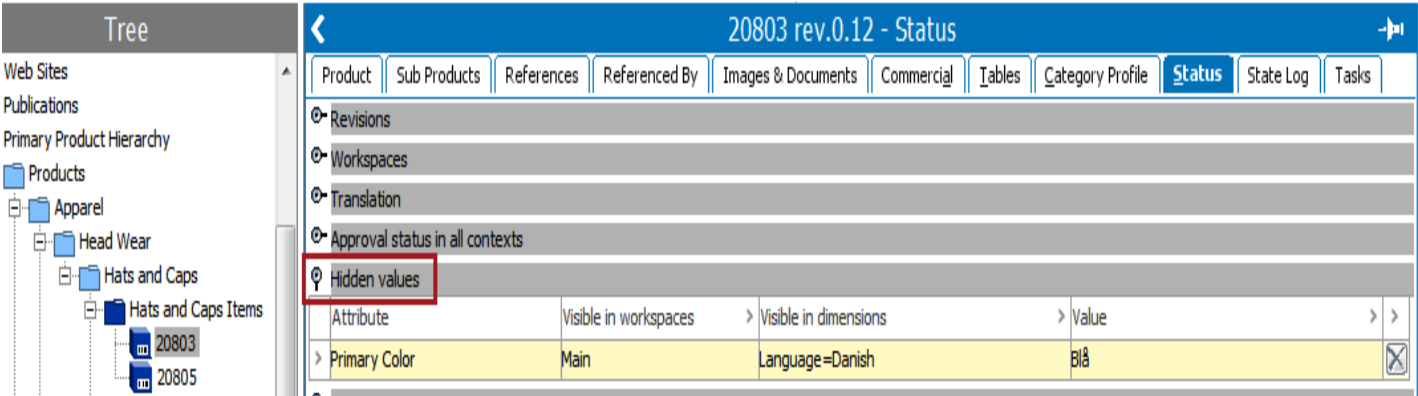


- Click **Remove Context**, or press Ctrl + Minus on your keyboard
- The context is deleted.

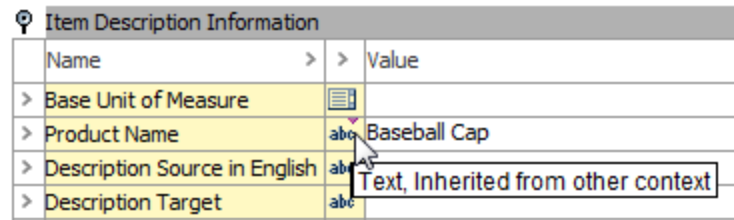
Hidden Values

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:

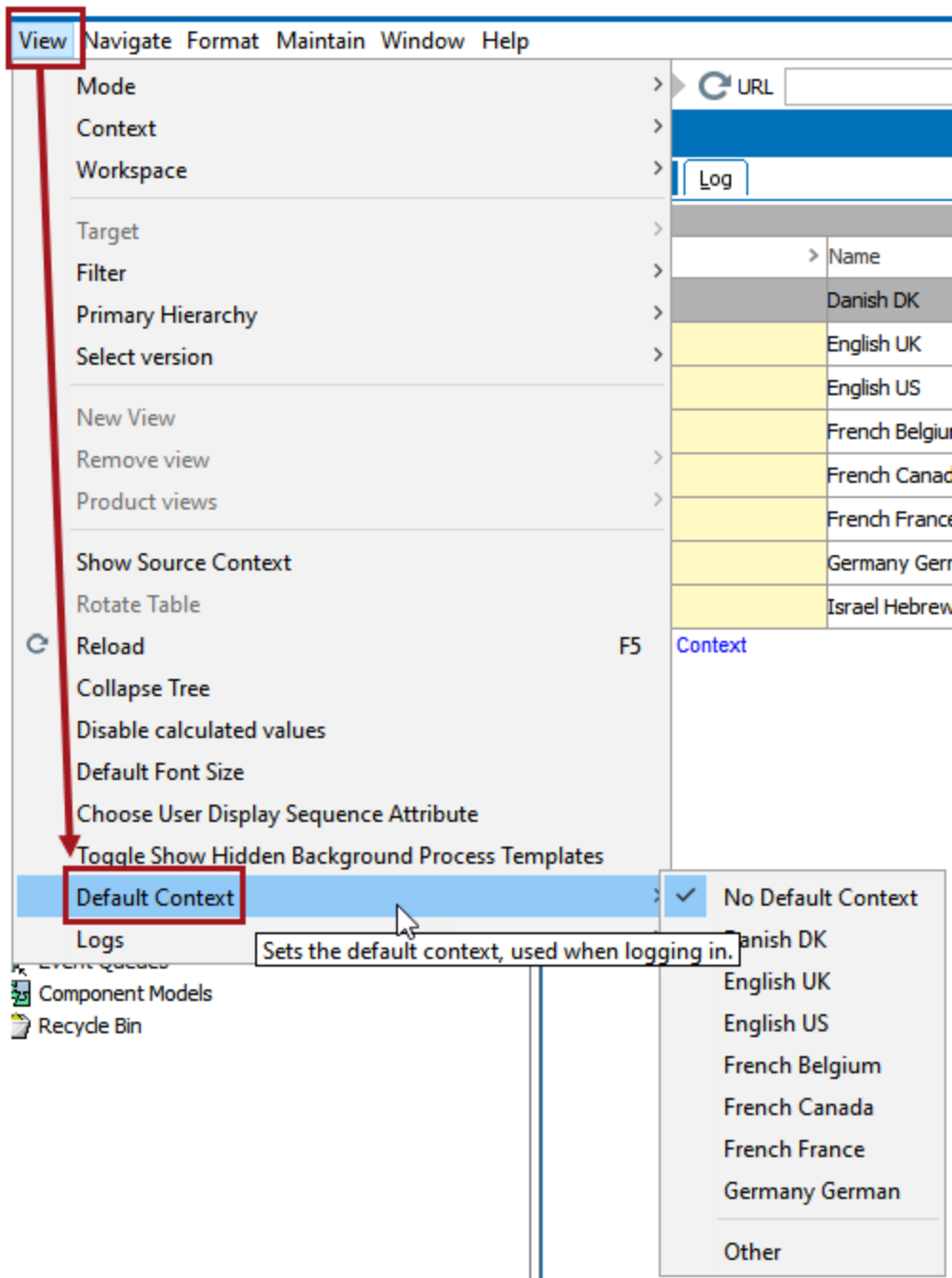


Note: Deleting a context can also cause a product to not be fully approved even if it is approved in all contexts. This is because there are still existing values in a deleted context.

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**.

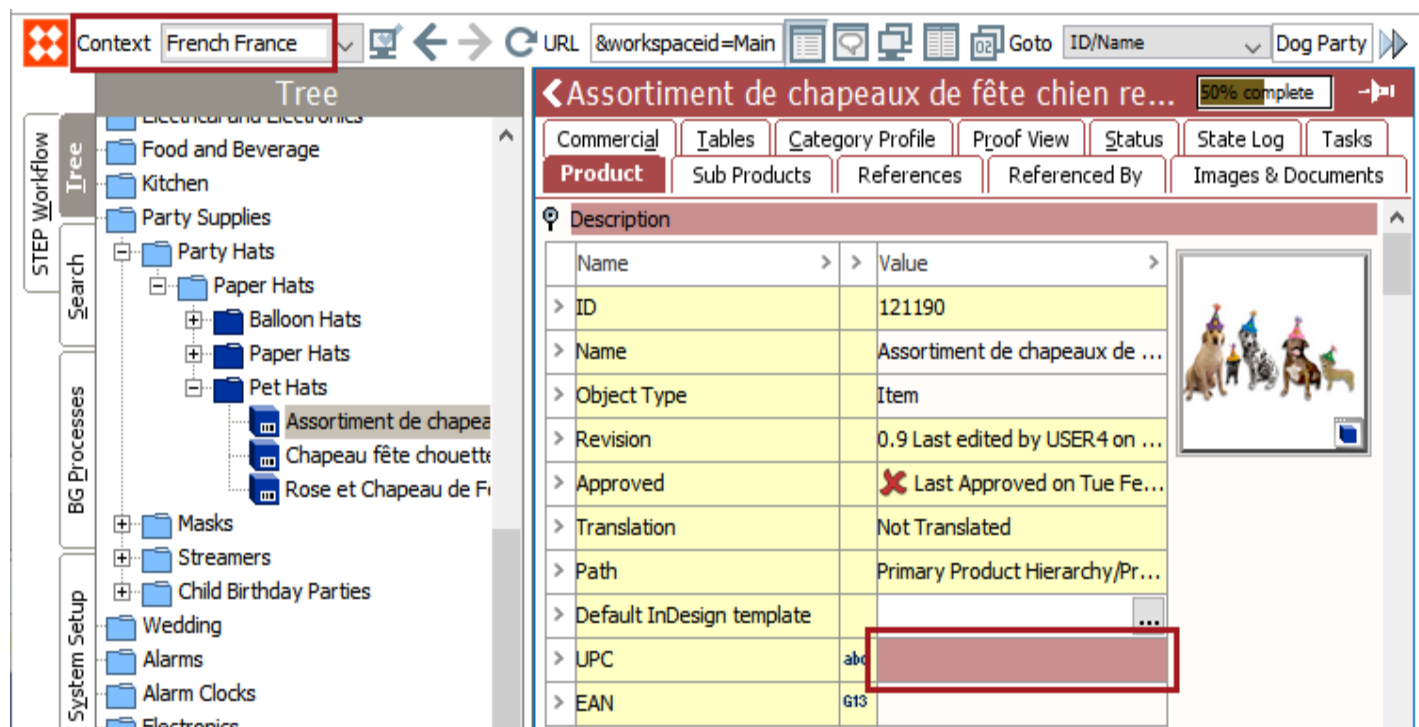


2. Select the preferred context from the list of contexts that displays. If you do not see your context, click **Other**.
3. Choose your preferred context from the list in the 'Other Contexts' dialog, then click **OK**.



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.

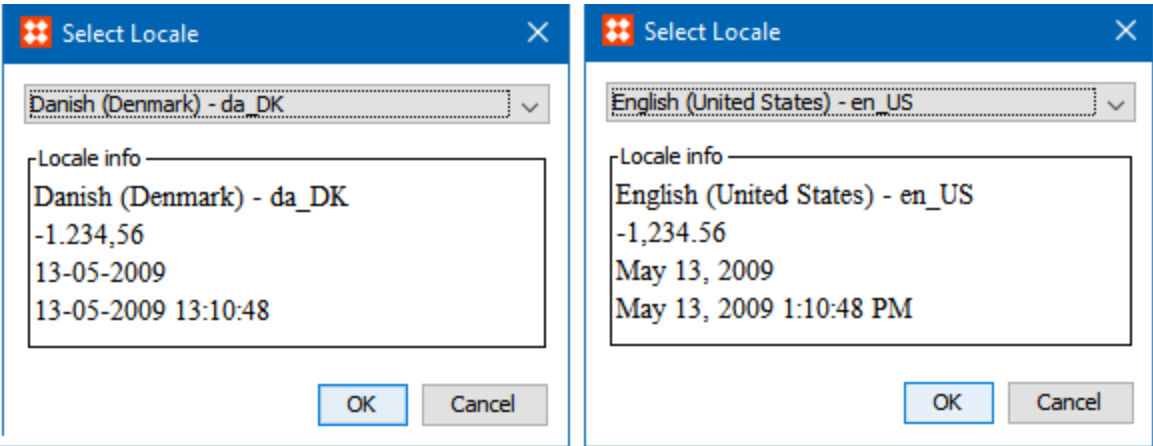


Context Locales

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.



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

Log

Contexts

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 data has been changed. An event triggered automatically is known as a Core Event. For more information, see the **Core Events** section.

Optionally, events can be manually generated based on some system action or setup (e.g., 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.

Core Events

Within STEP, an event indicates data has been changed. An event triggered automatically is known as a 'core event'. 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. Except for the Republish event, core events are all generated automatically by STEP when the actions defined below are performed.

This topic addresses:

- Core Events and Event Types
- Core Event Generation
- Using Core Events
- Change Flags for Events

Prerequisites

It may be helpful to review the **Workspaces** and **Managing Revisions in STEP** sections of the **System Setup / Super User Guide** documentation prior to this topic.

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

Core Events and Event Types

The table below lists core events, the associated event type, and the action performed to generate the core event.

Core Event	Event Type	Action Performed to Generate the Core Event
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,

Core Event	Event Type	Action Performed to Generate the Core Event
		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.
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.

Important: The timing or action performed to generate a core event is especially important when dealing with an OIEP or Event Processor, because it impacts when an OIEP or an Event Processor is triggered. For more information related to OIEPs see the **OIEP - Event-Based - Event Triggering Definitions Tab** section of the **Outbound Integration Endpoints** documentation. For more information related to Event Processors, see the **Event Triggering Definitions Tab** section of the Event Processors documentation.

Core Event Generation

Core events are automatically generated based on both the **database transaction method** and the **revisability** used. Details for each are below.

Database Transaction Methods

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.

Available database transaction methods:

- **STEP Workbench**
 - Generates a transaction each time a field is changed.
 - Does not provide a Save button.
- **Web UI**
 - Generates a transaction when the Save button is clicked.
 - May include several changes in a single transaction.
- **Import**
 - Generates a transaction on the success of the import.
 - May include several changes in a single transaction.

Revisability

Revisability determines when it is possible to include different values per workspace. Revisability can be either global or workspace dependent. Below are details of how revisability works with each of the database transaction methods to affect the generation of events.

- **Global Revisable Objects**
 - Do not require approval (values are the same for all workspaces), thus an event is generated when the database transaction is written.
 - Are system setup objects (workflows, business rules, integration endpoints, event processors, and all publication structures), externally maintained attributes, and entities with the revisability parameter set as global revisable.
 - Create a transaction and event based on the change method, as follows:

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

For example, assuming an event-based OIEP is set up to listen for appropriate events on the applicable object types, 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. Whereas on Workbench, if a value of an externally maintained attribute is changed, the value gets reflected automatically in all the workspaces.

• Workspace Revisable Objects

- Do require approval (values can differ per workspace), thus an event is generated when the object is approved.
- Are products, classifications, assets, and entities set as workspace revisable.
- Create a transaction and event based on the change method, as follows:

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

For example, assuming an event-based OIEP is set up to listen for appropriate events on the applicable object types, changing data in Web UI or Workbench on an entity object that is workspace revisable would not trigger an OIEP export until that object is approved.

Using Core Events

Core events can be monitored and used to kick-off additional processing (e.g., a derived event, business rules, event processors, a post-processor on an OIEP, and some event handlers can cause events to be generated). Additionally, events can be manually generated based on some system action or setup.

To learn more about:

- Derived events, see the **Derived Events** documentation.
- Business rules, see the **Business Rules** documentation.
- Event processors, see the **Event Processors** documentation.
- How core events can be used to control the output and format of an OIEP (Outbound Integration Endpoint), see the **OIEP - Event-Based - Output Templates Flipper** topic in the **Outbound Integration Endpoints** section of the **Data Exchange** documentation
- Manually generating events on demand, see the **Generating Match Codes and Running a Matching Algorithm** section of the **Matching, Linking, and Merging** documentation.

Change Flags for Events

Although actions can trigger events as they occur, a record of the individual actions taken are not stored in the database—only the end result. This means that it is not possible to review a list of actions that comprised an event, and also that STEP does not track the series of individual actions taken by an event, outside of any revision created. For information on revisions, see **Managing Revisions in STEP**.

When exporting event data, to determine if a change has happened, STEP compares the current values with the values in the revision prior to the event (based on the timestamp). The XML output then reports the current truth along with a 'change flag' or 'changed marker', which is output as **Changed="true"**.

For externally maintained data, since revisions are not created, a change to one externally maintained object results in change flags being included on all externally maintained objects, also referred to as a 'false positive'. This is because STEP cannot determine exactly which externally maintained data on the object has changed. False positives are likely, especially when a large number of externally maintained attributes or references exist on an object. However, you can be certain that when a change flag is present, data has changed. In the event that an externally maintained value of an attribute or reference is deleted, no change flag is output. For more information on externally maintained data, see the 'Important considerations for revisions' heading within the **Managing Revisions in STEP**.

Limitations and Exceptions

In addition to the caveats on externally maintained data mentioned above, the following limitations exist for change flags:

- No change flags are produced when republishing events. For more information, see **Event-Based OIEP Forward, Rewind, Purge, and Republish**.
- No change flags are produced when using the context splitter post processor on a OIEP Output Template. For more information, see **OIEP - Event-Based - Output Templates Flipper**.
- No change flags are produced for hierarchy inherited values. Change flags are only applied on the node that 'owns' the data. For more information, see **Inherited Attributes**.
- If there are multiple edits on an attribute within a scheduled Event publish time, it is not possible to record all the edits and publish the events at a scheduled time as STEP gives the latest change available during the publish of the event.
- No change flags are produced for deleted values.
- Use caution when relying on change flags in combination with dimension inheritance or suppression of references. The complexity of change flags is much greater in these scenarios.
- Derived events are always compared between the first approval of a node and the last approval of the node. This results in an excessive number of change flags.
- When using the Expand LOV Value Changes event processor, 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. For more information, see **Expand LOV Value Changes Processing Plugin Parameters and Triggers**.
- Data Container tags are marked **Changed=true** on the data container object itself, and not on that of the individual attribute value tags inside the data container.

Note: To eliminate any concern about when change flags are included (or are not included), you should consider each export as a full replacement.

For an example of a number of transactions and how they are represented by the change flag, see the **Change Flag Example** documentation.

Change Flag Example

When processing events via an event processor or an OIEP, a change flag is displayed in XML output as Changed="true". A Change flag indicates the current truth, namely that data in the event being processed is different compared to the approved data currently in STEP.

Consider the transactions (Txn) shown in the table below, and the resulting output, to understand how change flags work. The data was generated using the following setup:

- The exported data is a single, internally maintained attribute (attrWarranty), that has a dimension dependency of Country, and is visible in both contexts being exported.
- The values are modified in two contexts: one that includes the Country dimension point = USA, and a second that includes the Country dimension point = Canada.
- The OIEP is configured for 'No event batching'. This means the OIEP will fully process a single event (run an export, create a file, and deliver the file) before processing the next event in the queue.

Note: In the table, the 'Event XML Processed' column output has been edited to show only the attribute values for product ID, object type (UserTypeID), parent ID, Name, and attrWarranty.

Txn	Action	Event XML Processed
1	attrWarranty value originally set to Y in the USA and Canada dimension points is updated to N in both dimension points. While viewing both contexts using Context Mode in workbench, the object is approved for both contexts, two events are generated, and a revision is created. OIEP is invoked and processes Event 1 and Event 2. Export compares current attrWarranty approved values to those in the prior revision: • Prior revision values: Y in	Event 1: <pre><Products> <Product ID="181120" UserTypeID="Item" ParentID="181119"> <Name QualifierID="USA">Acme Anvil</Name> <Values> <ValueGroup AttributeID="attrWarranty" Changed="true"> <Value QualifierID="USA" Changed="true">N</Value> <Value QualifierID="Canada" Changed="true">N</Value> </ValueGroup> </Values> </Product> </Products></pre> Event 2: <pre><Products> <Product ID="181120" UserTypeID="Item" ParentID="181119"> <Name QualifierID="USA">Acme Anvil</Name> <Values> <ValueGroup AttributeID="attrrWarranty" Changed="true"> <Value QualifierID="USA">N</Value> <Value QualifierID="Canada" Changed="true">N</Value> </ValueGroup> </Values> </Product> </Products></pre>

Txn	Action	Event XML Processed
	USA and Y in Canada • Main values: N in USA and N in Canada • Approved values: N in USA and N in Canada For Event 1, the value in the approved workspace for each context is different from the previous revision, so both dimension points are marked with Changed="true" in the XML. For Event 2, because two contexts were approved, a false positive flag is included as the second event, although the change had already been reported in the previous event.	
2	attrWarranty value is cleared in the USA and Canada dimension points. Object is NOT approved, NO revision is created. Since the object is not approved in either context, no event is generated. • Prior revision values: N in USA and N in Canada • Main values: {blank} in USA and {blank} in Canada • Approved values: N in USA and N in Canada	
3	attrWarranty value is set to Y	Event 3:

Txn	Action	Event XML Processed
	for USA. Object is approved in the context with USA dimension only, an event is generated, and a revision is created. OIEP is invoked and processes Event 3. - Prior revision values: N in USA and N in Canada - Main values: Y in USA and {blank} in Canada - Approved values: Y in USA and N in Canada The value for USA has been changed to Y and approved, so the change flag is true. The removal of the value for Canada has not yet been approved, so it continues to export as N.	<pre> <Products> <Product ID="181120" UserTypeID="Item" ParentID="181119"> <Name QualifierID="USA">Acme Anvil</Name> <Values> <ValueGroup AttributeID="attrWarranty" Changed="true"> <Value QualifierID="USA" Changed="true">Y</Value> <Value QualifierID="Canada">N</Value> </ValueGroup> </Values> </Product> </Products> </pre>
4	attrWarranty value is set to Y for Canada. Object is approved in the context with Canada dimension only, an event is generated, and a revision is created. OIEP is invoked and processes Event 4. - Prior revision values: Y in USA and N in Canada - Main values: Y in USA and Y in Canada	Event 4: <pre> <Products> <Product ID="181120" UserTypeID="Item" ParentID="181119"> <Name QualifierID="USA">Acme Anvil</Name> <Values> <ValueGroup AttributeID="attrWarranty" Changed="true"> <Value QualifierID="USA">Y</Value> <Value QualifierID="Canada" Changed="true">Y</Value> </ValueGroup> </Values> </Product> </Products> </pre>

Txn	Action	Event XML Processed
	Approved values: Y in USA and Y in Canada Compared to the previous revision, only the Canada value has changed, so the change flag is true.	
5	attrWarranty value is cleared from the USA and Canada dimension points. Object is approved in both contexts, two events are generated, and a revision is created. OIEP is invoked and processes Event 5 and Event 6. - Prior revision values: Y in USA and Y in Canada - Main values: {blank} in USA and {blank} in Canada - Approved values: {blank} in USA and {blank} in Canada The attrWarranty attribute is excluded from the XML output because neither of the contexts in the Approved workspace contains a value.	Event 5: <pre><Products> <Product ID="181120" UserTypeID="Item" ParentID="181119"> <Name QualifierID="USA">Acme Anvil</Name> <Values> </Values> </Product> </Products></pre> Event 6: <pre><Products> <Product ID="181120" UserTypeID="Item" ParentID="181119"> <Name QualifierID="USA">Acme Anvil</Name> <Values> </Values> </Product> </Products></pre>

Changing the OIEP to use batching would cause multiple events to be delivered in a single file. This would lessen the number of occurrences of a 'false positive' change flag, but would also require that the downstream system be able to process multiple objects per file. For more information, see the **Event-Based OIEP Event Batching** topic in the **Data Exchange** documentation.

Derived Events

A derived event is an event that is triggered based on the triggering of another event. When the trigger you need is not available as a core event, use a derived event.

For example, it is common to configure a derived event to generate when:

- An object reaches a specific state in a workflow.
- An event-based OIEP cannot discriminate the republish event from other types of events because it would also be registering events and publishing data.
- Referenced images, referenced products, parent products, or products that are linked, removed, or modified in a specific classification structure.
- The triggering object types are set up on an event-based OIEP, and then the business rules are applied to generate derived events based on the original object type triggering.
- An object does not have a change in approval status, but the inherited value changes on the children need to be published.

Prerequisites

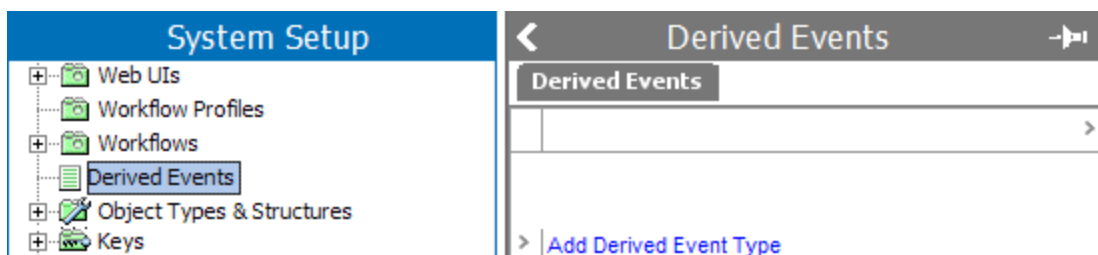
Derived Events is an advanced topic, it is expected that anyone reading this documentation is familiar with Core Events. For more information see the **Core Events** section of this guide.

Also, a thorough understanding of business rules is needed to create successful derived events. For more information, see the **Business Rules** section, and see the **Event-Based Example Business Rules for Derived Events** section of the **Data Exchange** documentation.

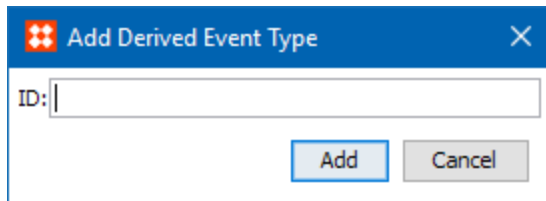
Add a Derived Event Type

To begin using derived events, a derived event type must exist. Below are the steps to add a derived event type.

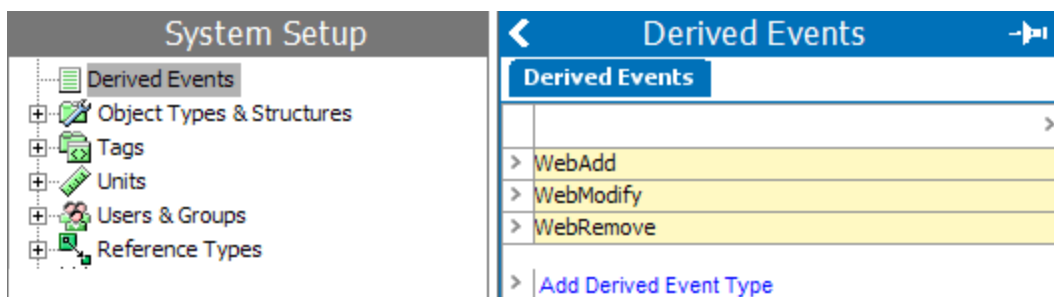
1. Go to System Setup.
2. Click **Derived Events**.



- On the Derived Events tab, click the **Add Derived Event Type** link, and the Add Derived Event Type dialog will display.



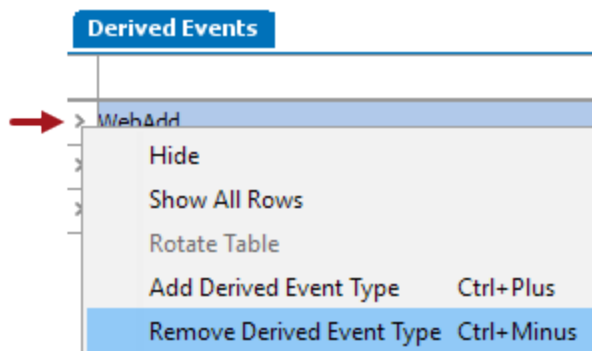
- Type an ID, click the **Add** button, and the Derived Events tab will list the newly added derived event type. In the example below the commonly used derived event types are listed. Once the derived event type is listed, it is available for use.



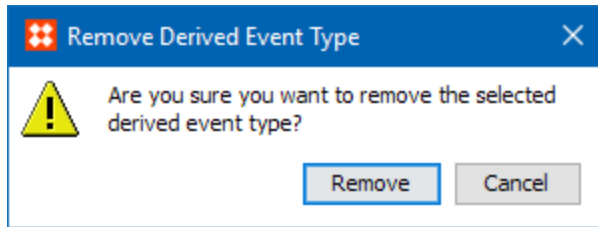
Delete a Derived Event Type

It is possible to delete a derived event type. Below are the steps to remove a derived event type.

- Go to System Setup.
- Click **Derived Events**.
- Find the derived event type to be removed, select the row icon, and the available options will display.



- Click **Remove Derived Event Type** and the Remove Derived Event Type dialog will appear.



5. Click **Remove** and the Derived Events tab will no longer list the deleted derived event type.

Event Processors

An event processor can be used to monitor events within STEP. Each event processor has many parameters but most notably the collective configuration of a processing plugin, a processor queue, and event triggering definitions. These allow for the collected events to be logically processed, and actions automatically performed, based upon each event processor's specific configuration.

Event processors can be used to automatically perform actions based on triggering events, however much of what an event processor can do is controlled by the processing plugin. Before creating an event processor, you should know the processing plugin to be used. For a complete list of STEP processing plugins, see the **Processing Plugins** documentation.

Prerequisites

Before an Event Processor can be created / configured the one-time setup tasks described in the **Initial Setup for Event Processors** must be completed to allow for the creation of event processors.

It is expected that anyone configuring an event processor is familiar with how and when events are generated. For more information, see the **Events** section of the **System Setup / Super User Guide** documentation.

Setup Requirements

Setting up and using an event processor involves the following steps:

1. Perform one-time setup tasks to allow for the creation of event processors as described in **Initial Setup for Event Processors**.
2. Launch the event processor wizard as described in **Creating an Event Processor**.
3. In the wizard, add an ID, name, and object type as described in **EPW - Identify Event Processor**.
4. In the wizard, specify the processing and error handling as described in **EPW - Configure Event Processor**.
5. In the wizard, specify the information specific to the processing plugin selected in the previous step as described in **EPW - Configure Processing Plugin**.
6. In the wizard, specify how often the event processor should start as described in **EPW - Schedule Event Processor**.
7. In the wizard, populate the send report to address parameter as described in **EPW - Configure Error Reporter Processing Plugin**.
8. Confirm / Update triggering definitions as described in **Event Triggering Definitions** documentation.
9. Set Queue Status Events to 'Read Events' as described in **Event Processor Queue Status**.
10. Enable the event processor as described in the **Enable Event Processor** section of the **Running an Event Processor** documentation.

Event Processor Results

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 **Outbound Integration Endpoints** section within the **Data Exchange** documentation.

Monitoring the progress of an event processor from an external system is available by way of a STEP Monitoring Sensor. For details, see the **Monitoring an Event Processor via External System** documentation.

Processing Plugins

An event processor consists of an outer Event Processor Framework for finding and reacting on events, and an inner processing mechanism (processing plugin) that deals with the actual event batch. The Event Processor Framework is the same for all event processors, whereas there are many different types of processing plugins.

For more information on Event Processors, see the **Event Processors** documentation.

When creating an event processor to monitor events within STEP, a processing plugin must be selected and configured.

Below is a list of available processing plugins and links to their configuration details. Custom processors can be created for various business requirements. For more information, contact your Stibo Systems account manager.

Processing Plugin Name	Description
CASS Certification Report	Generates CASS (Coding Accuracy Support System) certification reports. Only available for users in the US who have both a Loqate Local integration and a CASS license. For more information, see the CASS Certification Report Processing Plugin Parameters and Triggers documentation.
Execute Business Action	Allows a business action to be run from an event processor, so that when changes occur, a custom business action on objects can be asynchronously triggered. See Execute Business Action Processing Plugin Parameters and Triggers documentation.
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. See Expand LOV Value Changes Processing Plugin Parameters and Triggers documentation.
Experian Email Validation	Allows the event processor to be configured with an Experian Email Validation Configuration, and a business condition used to revalidate email information (Email Revalidation). Common setup is to be configured to listen for 'Account' events made by changes in the 'Email Field' and the 'Experian Email Validation' processing plugin. See Experian Email Validation Processing Plugin 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. See Image Cache Processing Plugin Parameters and Triggers documentation.

Processing Plugin Name	Description
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. See Matching Processing Plugin Parameters and Triggers documentation.
Revision Management	Allows automatic purging of object revisions to limit the total number of revisions retained. See Revision Management Processing Plugin Parameters and Triggers documentation.

CASS Certification Report Processing Plugin Parameters and Triggers

The CASS Certification Report processing plugin allows for an event processor to be configured to generate CASS certification reports. CASS certification reports are text files stored as assets in the STEP Workbench classification hierarchy. These documents serve as a record of addresses that are validated through the CASS address standardization service and are provided to the USPS, upon request. These reports can be downloaded from within the workbench or by using standard Web UI components.

CASS (Coding Accuracy Support System) is a certification program run by the United States Postal Service (USPS) and is offered to all mailers, service bureaus, and software vendors that would like the USPS to evaluate the quality of their address matching. Mailers who use CASS-certified software to check their mailing addresses are able to qualify for discounted postage rates from the USPS. CASS is only valid for US addresses.

Note: It is strongly recommended to create a CASS certification report event processor by using the 'Easy setup of CASS Component Model' wizard, which is available by right-clicking on the CASS Address Component Model object in System Setup. Using the easy setup wizard is the most user-friendly and straightforward method for creation of a CASS event processor. However, the below sections of this documentation topic provide information for users who may choose to create this event processor manually. For more information, see the **CASS Address Component Model** topic in the **Data Integration** documentation.

The Parameters and Event Triggers sections below contain important information on settings that should be considered when creating an event processor using this processing plugin.

Prerequisites

This section of documentation describes configuration steps for this specific processor, but that is only one part of configuring an event processor. For the full set of instructions on configuring an event processor, see the **Event Processors** documentation.

Before configuration of an event processor using the CASS Certification Report processor can occur, the following conditions must be met.

- Users must be based in the United States, as CASS is not valid outside of the US.
- Users must be connected to a Loqate Local server installation and have purchased a CASS license. The CASS components will not work with a Loqate Cloud API installation. For more information, see the **Loqate Integration** topic in the **Data Integration** documentation.
- The Address Component Model must already be configured. It is strongly recommended to configure this component model using the 'Easy setup of Address Component Model' wizard. For more information, see the **Address Component Model** section of the **Loqate** documentation.

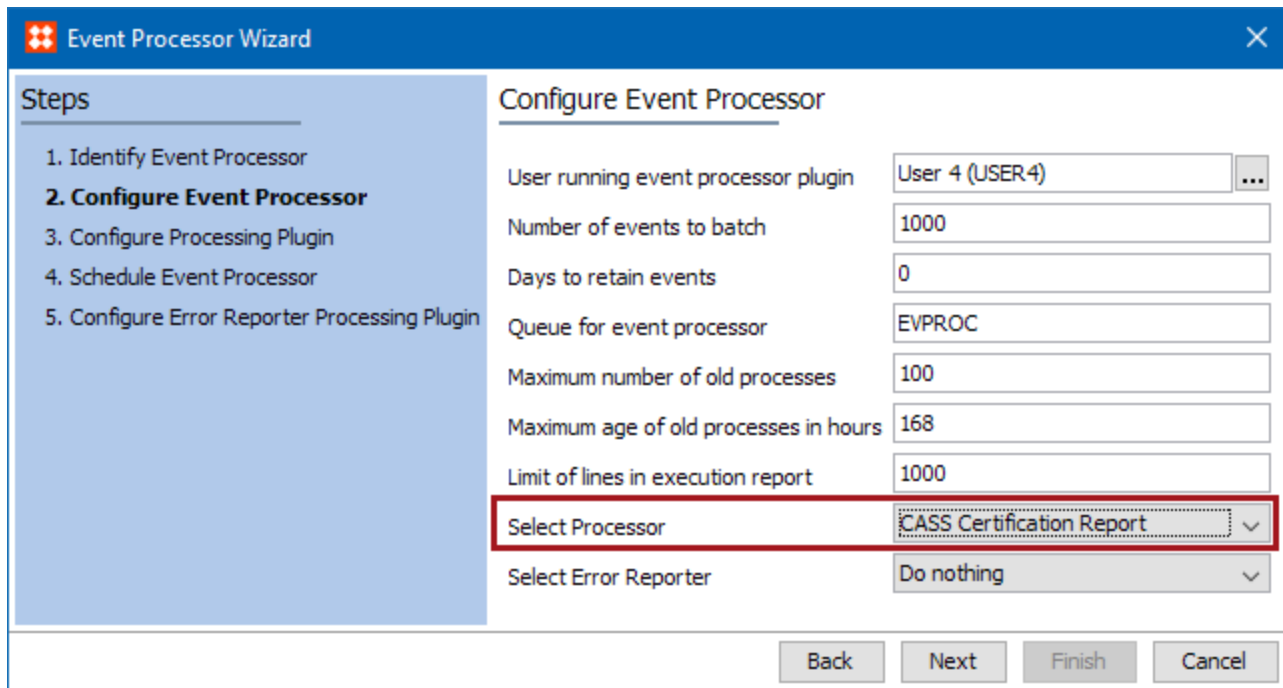
Note: Ideally, the CASS Address Component Model should also be previously configured. However, if the CASS component model has already been configured through the easy setup method, a CASS certification report event processor should have already been created automatically during that process. If so, the below sections of this

documentation simply provide additional information on manual setup of event processors with some CASS-specific details highlighted, as users may need to return to the event processor and make manual changes later.

Parameters

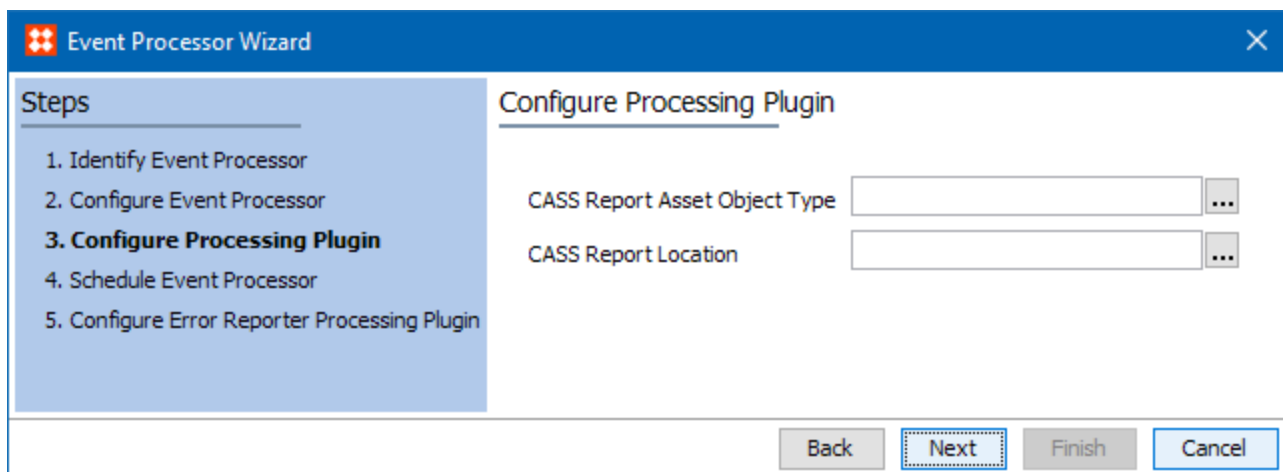
Each of the parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below.

To access the 'Configure Processing Plugin' parameters as shown below, the **CASS Certification Report** processing plugin must be selected within the Select Processor parameter during the wizard step 'Configure Event Processor.'



The screenshot shows the 'Event Processor Wizard' window. On the left, the 'Steps' list includes: 1. Identify Event Processor, 2. **Configure Event Processor**, 3. Configure Processing Plugin, 4. Schedule Event Processor, and 5. Configure Error Reporter Processing Plugin. The main area is titled 'Configure Event Processor' and contains several input fields: 'User running event processor plugin' (User 4 (USER4)), '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' (CASS Certification Report), and 'Select Error Reporter' (Do nothing). The 'Select Processor' dropdown is highlighted with a red border. At the bottom are buttons for 'Back', 'Next', 'Finish', and 'Cancel'.

Once the CASS Certification Report processing plugin has been selected, click the **Next** button to display the 'Configure Processing Plugin' wizard screen.



The screenshot shows the 'Event Processor Wizard' window at the 'Configure Processing Plugin' step. The 'Steps' list on the left now highlights step 3: 'Configure Processing Plugin'. The main area contains two input fields: 'CASS Report Asset Object Type' and 'CASS Report Location', both with empty text boxes and dropdown arrows. At the bottom, the 'Next' button is highlighted with a blue border, while 'Back', 'Finish', and 'Cancel' are in grey.

- **CASS Report Asset Object Type:** Click the ellipsis button (...) to display the 'Select Objecttype' dialog, then select the asset object type that should be used for the CASS certification reports. CASS certification reports are assets generated as .txt files by the CASS certification report event processor.
- **CASS Report Location:** Click the ellipsis button (...) to display the 'Select Classification' dialog, then select the classification folder where the CASS certification report asset objects should be stored after being generated by the event processor.

Once any required configurations are made, click the **Next** button to display the wizard step 'Schedule Event Processor.'

See the **EPW - Schedule Event Processor** topic within the **Creating an Event Processor** section of the **Event Processors** documentation.

Event Triggers

Unlike other event processors, no triggering events are required to be configured on the Event Triggering Definitions tab for CASS certification report event processors. CASS certification report event processors are typically triggered when addresses are CASS validated through an 'Address Standardization' bulk update or business action. After validation and the return of information from the Loqate server, the actioned objects (e.g. address objects or entity objects that carry address data containers) are sent as events to the event processor.

To standardize US addresses with CASS, two options must be enabled when configuring the 'Standardize Address' operation for business actions and bulk updates—'Turn on CASS validation for US addresses' and 'CASS Certification Report Event Processor.' The following screenshot shows these options as they appear in the 'Operations' step of the Bulk Update wizard and on the 'Edit Operation' dialog of the Business Rule Editor.

For more information on the Standardize Address operation for bulk updates, see the **Data Quality Operations for Bulk Updates** topic in the **Bulk Updates** documentation. For more information on the Standardize Address operation for business actions, see the **Business Action: Standardize Address** topic in the **Business Rules** documentation.

Important: By default, events are discarded. For more information about the optional configuration to enable the Queue Status, see the **Queue Status for an Event Processor** documentation.

Execute Business Action Processing Plugin Parameters and Triggers

The Execute Business Action processing plugin allows for an event processor to be configured with a predefined custom business action. The custom business action determines when and how to act upon the events from the event processor. This is useful when changes occur on objects and custom actions need to occur.

For example, when changes to relationships between objects occur, an email may be sent to a specific user or attribute values may be recalculated. Another example is when an email object is changed, the email object can then be sent to an external service for validation. By design, business actions are highly customizable.

The Parameters and Event Triggers sections below contain important information on settings that should be considered when creating an event processor using this processing plugin.

Prerequisites

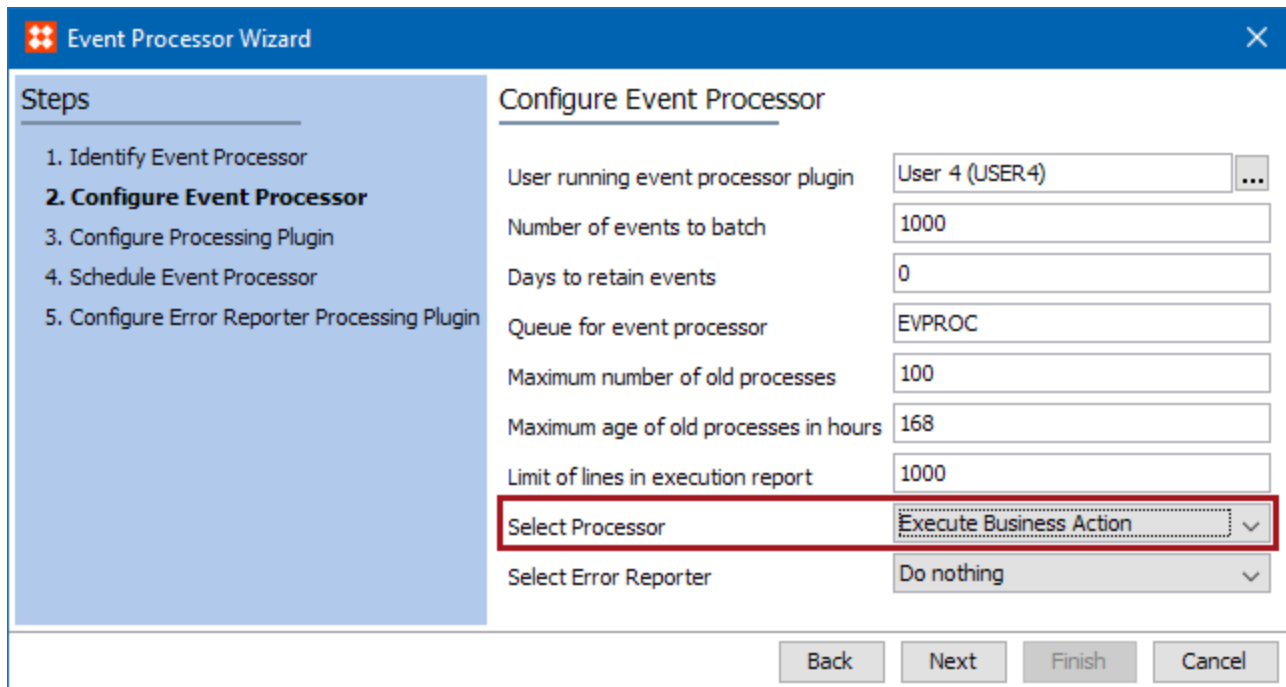
This section of documentation describes configuration steps for this specific processor, but that is only one part of configuring an event processor. For the full set of instructions on configuring an event processor, see the **Event Processors** documentation.

Before configuration of an event processor using the execute business action processor can occur, a business action must be created. For more information, see the **Business Actions** section within the **Business Rules** documentation.

Parameters

Each of the parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below.

To access the 'Configure Processing Plugin' parameters as shown below, the **Execute Business Action** processing plugin must be selected within the Select Processor parameter during the wizard step 'Configure Event Processor.'



Event Processor Wizard

Steps

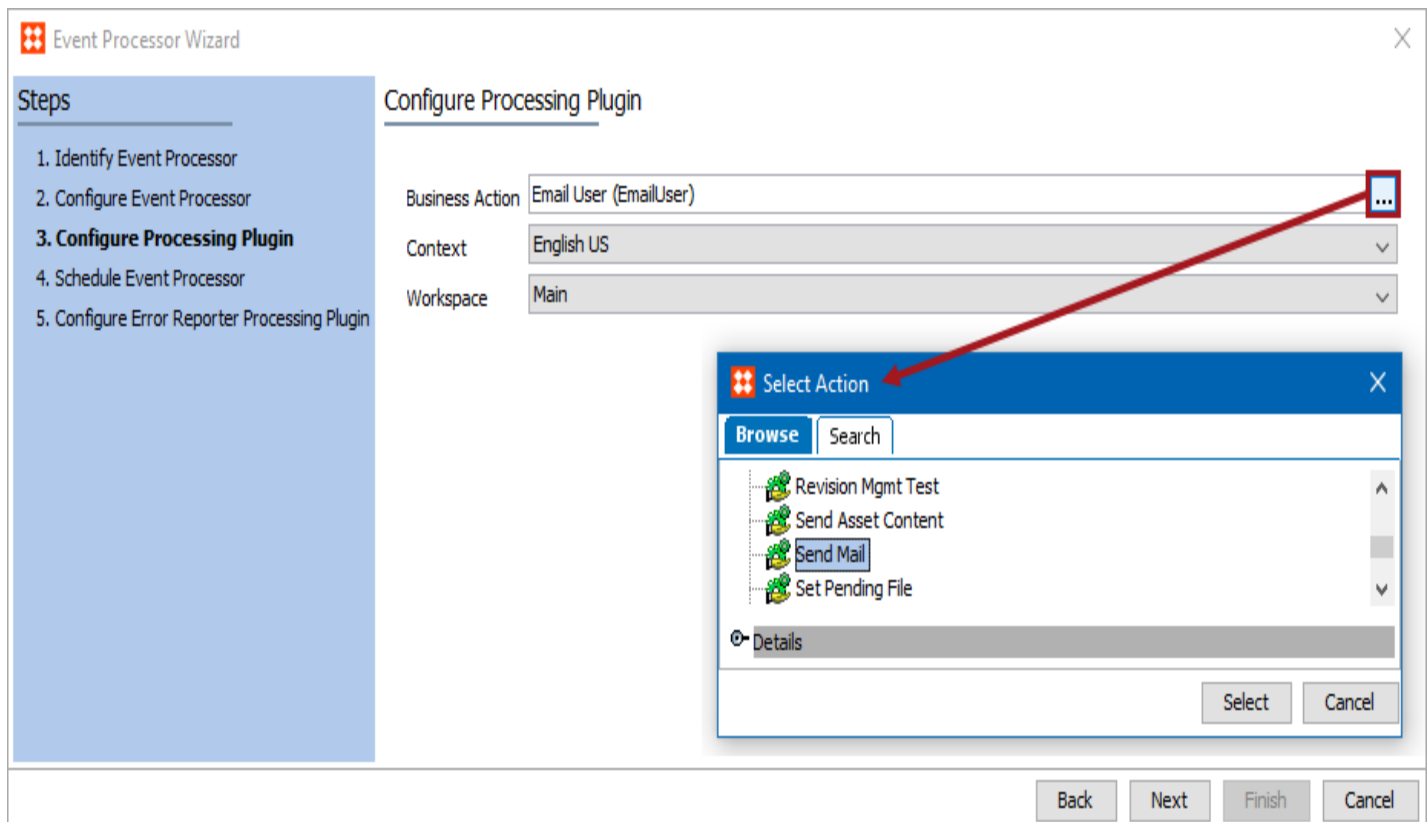
1. Identify Event Processor
- 2. Configure Event Processor**
3. Configure Processing Plugin
4. Schedule Event Processor
5. Configure Error Reporter Processing Plugin

Configure Event Processor

User running event processor plugin	User 4 (USER4) ...
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	Execute Business Action
Select Error Reporter	Do nothing

Back Next Finish Cancel

Once the Execute Business Action processing plugin has been selected, click the **Next** button, and the wizard step 'Configure Processing Plugin' will display.



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

Business Action	Email User (EmailUser) ...
Context	English US
Workspace	Main

Back Next Finish Cancel

Select Action

Browse Search

- Revision Mgmt Test
- Send Asset Content
- Send Mail
- Set Pending File

Details

Select Cancel

- **Business Action:** Click the ellipsis button (...) to display the Select Action dialog, and select the **necessary business action**.
- **Context:** Select a context from the dropdown. This will determine the context in which the business action will run.
- **Workspace:** Select a workspace from the dropdown. This will determine the workspace in which the business action will run.

Important: The Business Action will always read from the configured Context / Workspace independently of how the events are actually triggered.

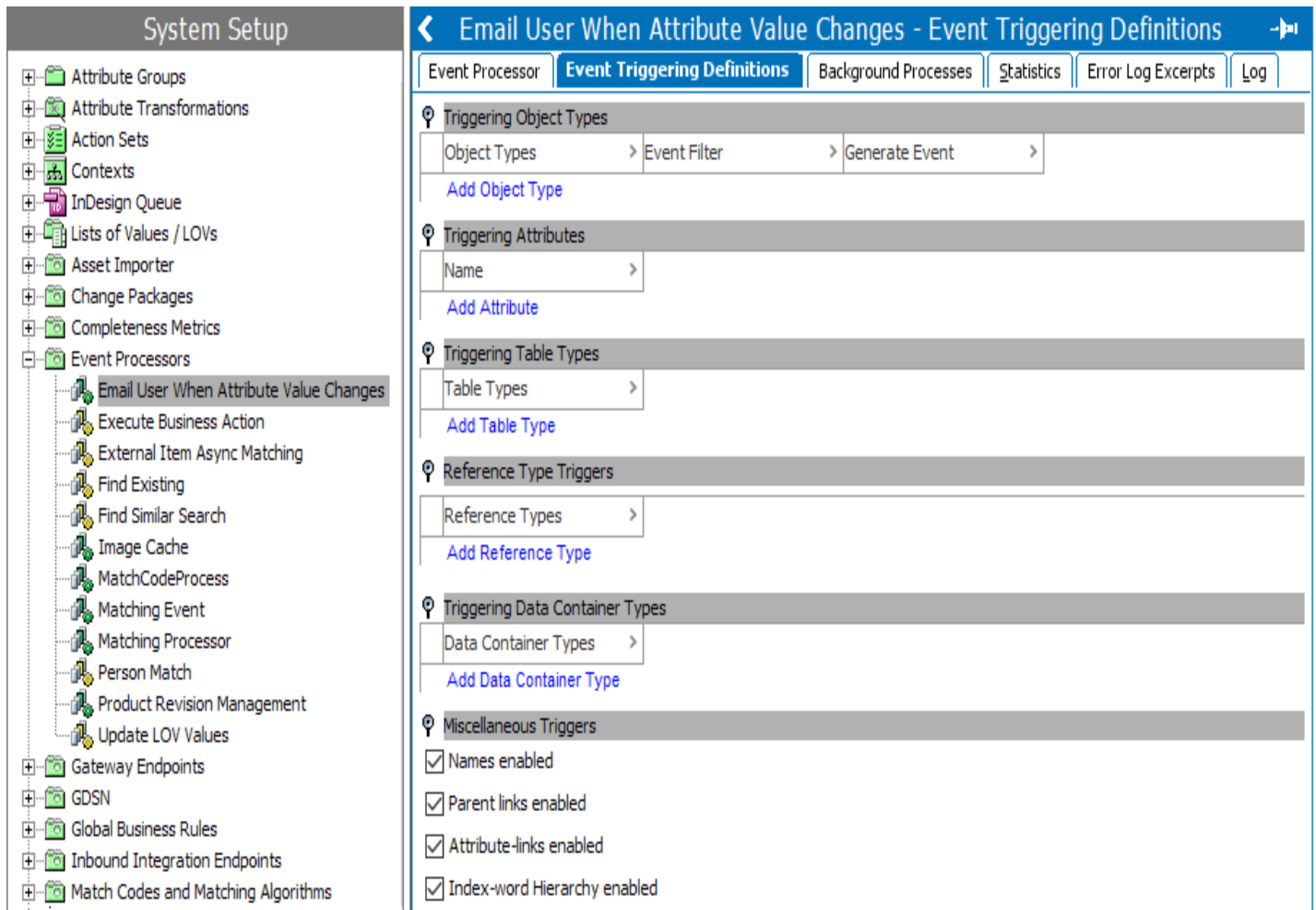
Once any required configurations are made, click the **Next** button to display the wizard step 'Schedule Event Processor.'

See the **EPW - Schedule Event Processor** topic within the **Creating an Event Processor** section of the **Event Processors** documentation.

Event Triggers

For the event processor to execute the business action, the Event Triggering Definitions tab must be correctly configured.

Select an event processor configured to use the Execute Business Action processor, and click on the Event Triggering Definitions tab. Use the various triggering flippers to configure the event processor to listen for changes on anything that should cause the event processor to execute.



- Upon creation, the parameters within the Miscellaneous Triggers flipper are enabled.
- When a configured trigger causes an event to reach the event processor queue, the business action will execute.
- When an event batch is successfully processed, no message is written into the execution report of the corresponding background process (BGP).
- When JavaScript is used to write the business action, the originating object of the event will be available via the 'Current Object' bind.
- If the event processor encounters a problem processing a batch, the event processor will retry the business actions on each event in the batch until all events have been processed or the failing event is identified, in which case the event processor will enter the failed state and stop processing new events.
- There is no guarantee that actions will not run multiple times, even when batch size is equal to one. However, the likelihood is reduced by the having a batch size of one, but performance can then be effected.

Important: By default, events are discarded. For more information about the optional configuration to enable the Queue Status, see the **Event Processor Queue Status** documentation.

Expand LOV Value Changes Processing Plugin 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 processing plugin handles LOV value updates only, not additions or deletions.

The Parameters and Event Triggers sections below contain important information on settings that should be considered when creating an event processor using this processing plugin.

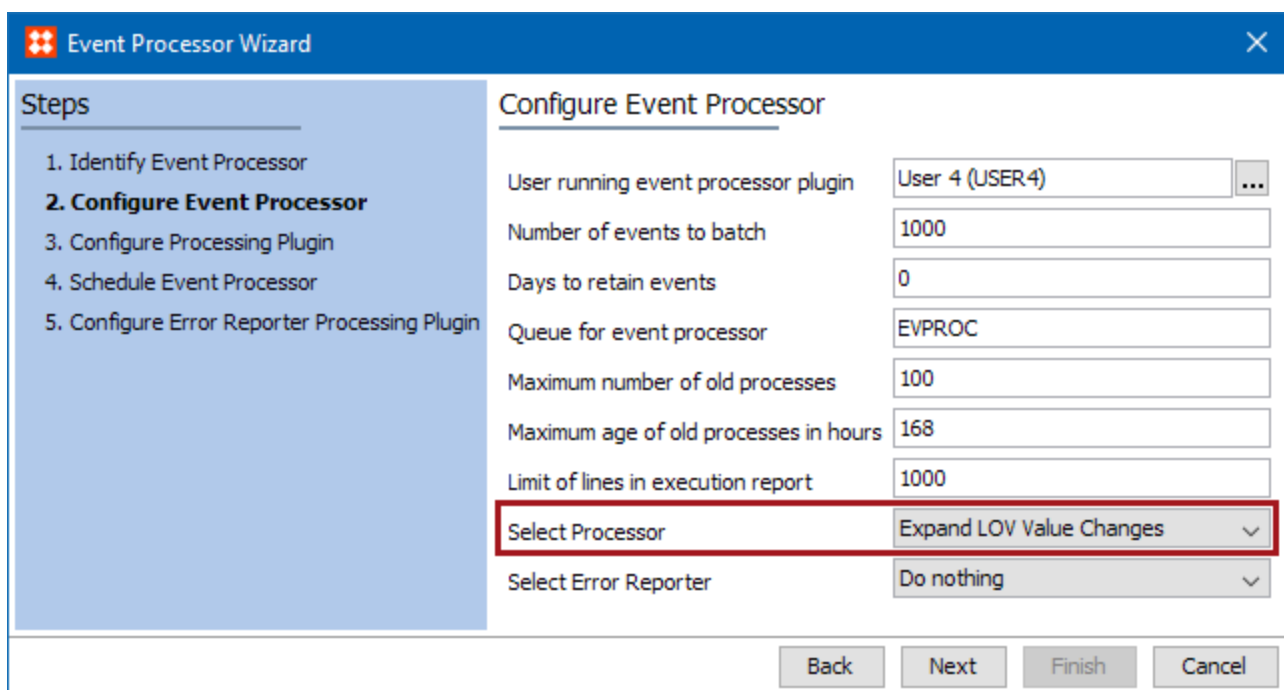
Prerequisites

This section of documentation describes configuration steps for this specific processor, but that is only one part of configuring an event processor. For the full set of instructions on configuring an event processor, see the **Event Processors** documentation.

It is expected that anyone configuring the Expand LOV Value Changes processing plugin is familiar with the **List of Values** documentation within the **System Setup** guide.

Parameters

To access the 'Configure Processing Plugin' parameters as shown below, the **Expand LOV Value Changes** processing plugin must be selected within the Select Processor parameter during the wizard step 'Configure Event Processor.'

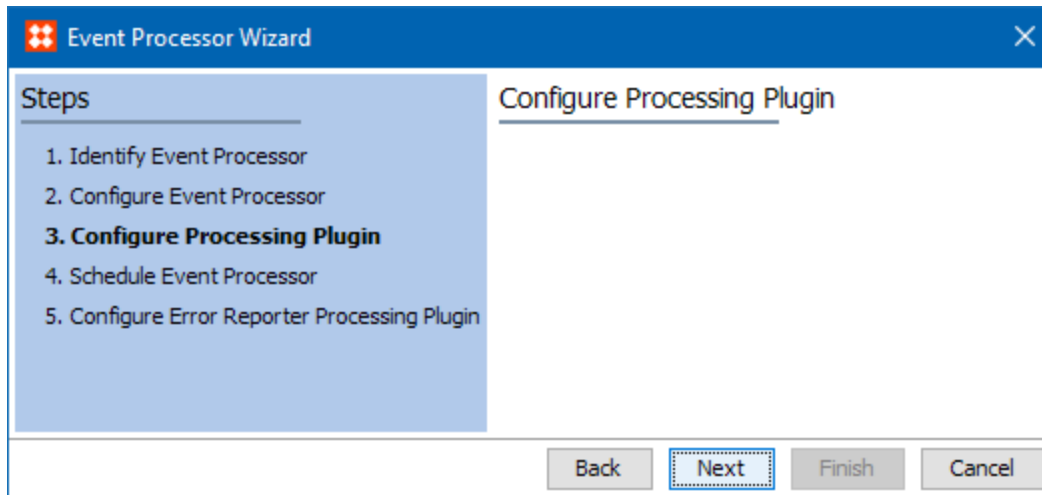


The screenshot shows the 'Event Processor Wizard' dialog box. On the left, a 'Steps' panel lists five steps: 1. Identify Event Processor, 2. **Configure Event Processor** (highlighted), 3. Configure Processing Plugin, 4. Schedule Event Processor, and 5. Configure Error Reporter Processing Plugin. The main area is titled 'Configure Event Processor' and contains several configuration fields:

- User running event processor plugin: User 4 (USER4)
- 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: **Expand LOV Value Changes** (highlighted with a red box)
- Select Error Reporter: Do nothing

At the bottom right, there are four buttons: Back, Next, Finish, and Cancel.

Once the Expand LOV Value Changes processing plugin has been selected, click the **Next** button, and the wizard step 'Configure Processing Plugin' will display.



On the Event Processor Wizard 'Configure Processing Plugin' step, no further configuration is required for the Expand LOV Value Changes processing plugin, therefore no parameters are displayed.

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.

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

(stibo.2207160) - Event Triggering Definitions

Event Processor | **Event Triggering Definitions** | Background Processes | Statistics

Triggering Object Types

- Object Types > Event Filter > Generate Event >
- > Domains ...
- [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

For Use with External Systems

An OIEP can be configured to listen for changes to LOV values and generate events for objects using the modified values.

For more information, see the **Event-Based Outbound Integration Endpoint** section of the **Outbound Integration Endpoint** documentation.

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.

For more information, see the **Change Flags for Events** section of the **Core Events** documentation.

Important: 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.

Experian Email Validation Processing Plugin Parameters and Triggers

Experian Email Validation processing plugin allows an event processor to be configured with an Experian Email Validation Configuration, and a business condition used to revalidate email information (Email Revalidation). Common setup is to configure the event processor to listen for events made by changes in the 'Email Field.'

The Parameters and Event Triggers sections below contain important information on settings that should be considered when creating an event processor using this processing plugin.

Prerequisites

This section of documentation describes configuration steps for this specific processor, but that is only one part of configuring an event processor. For the full set of instructions on configuring an event processor, see the **Event Processors** documentation.

The configuration requires an understanding of the Experian Email Validation Integration. For more information, see the **Experian Email Validation Integration** section of the **Data Integration** documentation.

Parameters

Each of the parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below.

To access the 'Configure Processing Plugin' parameters as shown below, the **Experian Email Validation** processor must be selected within the Select Processor parameter during the wizard step 'Configure Event Processor.'

The screenshot shows the 'Event Processor Wizard' window, specifically the 'Configure Event Processor' step. On the left, a 'Steps' pane lists five steps: 1. Identify Event Processor, 2. **Configure Event Processor**, 3. Configure Processing Plugin, 4. Schedule Event Processor, and 5. Configure Error Reporter Processing Plugin. The main area contains several configuration fields: 'User running event processor plugin' (USER NAME (USER NAME)), 'Number of events to batch' (10000), 'Days to retain events' (0), 'Queue for event processor' (EVPROC), 'Maximum number of old processes' (100), 'Maximum age of old processes in hours' (168), and 'Limit of lines in execution report' (1000). Below these are two dropdown menus: 'Select Processor' (set to 'Experian Email Validation') and 'Select Error Reporter' (set to 'Do nothing'). The 'Select Processor' dropdown is highlighted with a red rectangle. At the bottom are 'Back', 'Next', 'Finish', and 'Cancel' buttons.

Once the Experian Email Validation processing plugin has been selected, click the **Next** button, and the wizard step 'Configure Processing Plugin' will display.

The screenshot shows the 'Event Processor Wizard' window, specifically the 'Configure Processing Plugin' step. On the left, the 'Steps' pane lists five steps: 1. Identify Event Processor, 2. Configure Event Processor, 3. **Configure Processing Plugin**, 4. Schedule Event Processor, and 5. Configure Error Reporter Processing Plugin. The main area contains two configuration fields: 'Email Revalidation Condition' (Email Revalidation Condition (EmailRevalidationCondition)) and 'Experian Email Validation Configuration' (Experian Email Validation Configuration (ExperianEmailValidationConfiguration)). Both fields have ellipsis buttons to the right. At the bottom are 'Back', 'Next', 'Finish', and 'Cancel' buttons.

Note: Optionally, use the Email Component Model Easy Setup wizard to automatically configure these options. For more information, see the **Email Component Model** section of the **Experian Email Validation Integration**.

- **Email Revalidation Condition:** Click the ellipsis button (...) to display the Select Condition dialog, select your email revalidation condition. For more information, see the **Email Revalidation Business Condition** section of the **Experian Email Validation Integration**.

- **Experian Email Validation Configuration:** Click the ellipsis button (...) to display the Select Experian Email Validation Configuration dialog, select your configuration.

Once any required configurations are made, click the **Next** button to display the wizard step 'Schedule Event Processor.'

See the **EPW - Schedule Event Processor** topic within the **Creating an Event Processor** section of the **Event Processors** documentation.

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 made on the configured Triggering Object Types in the 'Email Field.'

If the Email Component Model Easy Setup wizard was used, then the Object Types selected during the wizard step 'Create New Email Data Container,' will automatically display within the Triggering Object Types flipper. Any email attributes selected or created during the wizard step 'Configure Email Fields,' will automatically display within the Triggering Attributes flipper.

For more information, see the **Configuring Experian Email Validation Integration Using the Easy Setup Option** section of the **Experian Email Validation Integration** documentation.

In the example below, during the Email Component Model Easy Setup wizard step 'Create New Email Data Container,' the Object Types 'Sales Manager' and 'Employee' were selected, and during the wizard step 'Configure Email Fields' the 'Email Field' attribute was automatically created.

System Setup

Event Processors

- BestEP
- CASS Certification Report
- Email User When Attribute Value Changes
- Experian Email Validation**
- External Item Async Matching
- Find Existing
- Find Similar Search
- Image Cache
- Loqate
- LOV Value Merge Events
- MatchCodeProcess
- Matching Event
- Matching Processor
- Person Match
- Product Revision Management
- Update LOV Values

- Experian Integration
- Gateway Endpoints
- GDSN
- Global Business Rules
- Inbound Integration Endpoints
- Integration Endpoints
- Match Codes and Matching Algorithms
- Merge GR
- Merge GR IIEP
- Outbound Integration Endpoints
- Web UIs
- Web UIs
- Workflow Profiles

Experian Email Validation - Event Triggering Definitions

Event Processor
Event Triggering Definitions
Background Processes
Statistics
Error Log Excerpts
Log

Triggering Object Types

Object Types	Event Filter	Generate Event
> Sales Manager	...	...
> Employee	...	...

Add Object Type

Triggering Attributes

Name	>
> Email Field	

Add Attribute

Triggering Table Types

Table Types	>

Add Table Type

Reference Type Triggers

Reference Types	>

Add Reference Type

Triggering Data Container Types

Data Container Types	>

Add Data Container Type

Miscellaneous Triggers

☒ Names enabled
☒ Parent links enabled
☐ Attribute-links enabled
☒ Index-word Hierarchy enabled

Once configured, the Experian Email Validation processing plugin will do the following for each event processor batch:

1. For each account object in the event processor batch, the processor will deduce the emails that need to be validated based upon the configuration of the Email Revalidation business condition. If no email revalidation condition is set, all email addresses (within the batch) will be validated.
2. Build an Experian batch request of the deduced emails.
3. The 'Experian Email Validation Integration Status' field will be updated with the value 'Processing.'
4. The event processor plugin will now wait, and after each poll period (configured in the Experian Configuration object) make a request for each Experian batch. No new events are picked up until all emails have been processed.

Note: The Experian service is an asynchronous service, aimed at validating large batches of email data. Therefore it will not return immediate responses. The response times depend on the number of emails sent,

and email data quality. The guaranteed response times are from 2 (less than 1,000 email addresses) to 20 hours (less than 100,000 email addresses).

5. When a response is received:

- One of the following result values will be written into the 'Email Status' email data quality field.
 - **(blank)**: Email has never been sent for validation.
 - **Disposable**: Domain is administered by a disposable email provider (e.g., Mailinator).
 - **Illegitimate**: Seed, spam trap, black hole, technical role account or inactive domain.
 - **Malformed**: The email is identified by STEP as having an illegal format.
 - **Undeliverable**: Mailbox or domain does not exist, or mailbox is full, suspended, or disabled.
 - **Unknown**: Experian service is unable to conclusively verify or invalidate this address.
 - **Unreachable**: Domain has no reachable mail exchangers.
 - **Verified**: Mailbox exists, is reachable, and not known to be illegitimate or disposable.
- The 'Last Validated Email' field will be updated with the email address last sent for validation.
- The 'Validation Timestamp' field will be updated with the date and time the email address was last received from the Experian service.
- The 'Experian Email Validation Integration Status' field will display one of the following values:
 - **(blank)**: Indicates the email address has never been sent for validation.
 - **Success**: Indicates the email address has been successfully validated.
 - **Processing**: Indicates the email address is in the process of being validated.
 - **Failed**: Indicates the email address validation has failed. This allows a search to be performed for accounts where an error has occurred during validation, and a collection is automatically created.

Note: If too many revisions occur (causing performance issues) for the 'Experian Email Validation Integration Status' field, then it is recommended that the attribute be set to externally maintained. For more information, see the **Externally Maintained Attributes** section of the **Attributes** documentation.

- The corresponding events will be marked as read.
- The steps described above will continue until a timeout occurs. The timeout setting ensures that all emails will get an updated status, and STEP will log the timeout in the execution report of the corresponding background process.
- If a fundamental error occurs that prevents STEP from obtaining a status from Experian, the event processor may go into a failed state, and stop.

Note: In case of a STEP system app server restart, the event processor will restart in the same queue, and this can potentially cause the same emails to be resent for validation.

Important: By default, events are discarded. For more information about the optional configuration to enable the Queue Status, see the **Event Processor Queue Status** documentation.

Image Cache Processing Plugin 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.

To access the 'Configure Processing Plugin' parameters as shown below, the **Image Cache** processor must be selected within the Select Processor parameter during the wizard step 'Configure Event Processor.'

The screenshot shows the 'Event Processor Wizard' window. On the left, a 'Steps' sidebar lists five steps: 1. Identify Event Processor, 2. **Configure Event Processor**, 3. Configure Processing Plugin, 4. Schedule Event Processor, and 5. Configure Error Reporter Processing Plugin. The main area is titled 'Configure Event Processor' and contains several configuration fields:

- User running event processor plugin: User 4 (USER4)
- 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**: Image Cache (highlighted with a red box)
- Select Error Reporter: Do nothing

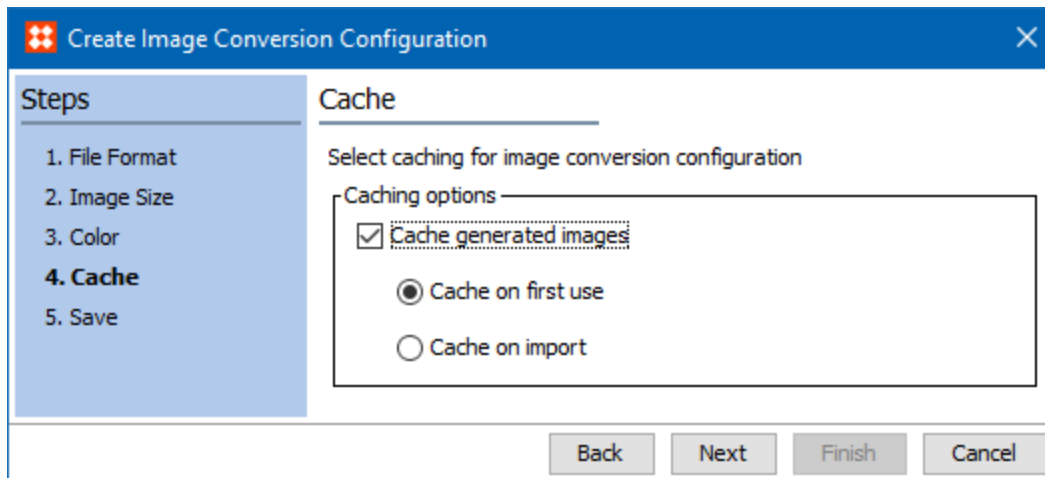
At the bottom of the window are four buttons: Back, Next, Finish, and Cancel.

The Parameters and Event Triggers sections below contain important information on settings that should be considered when creating an event processor using this processing plugin.

Prerequisites

This section of documentation describes configuration steps for this specific processor, but that is only one part of configuring an event processor. For the full set of 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.

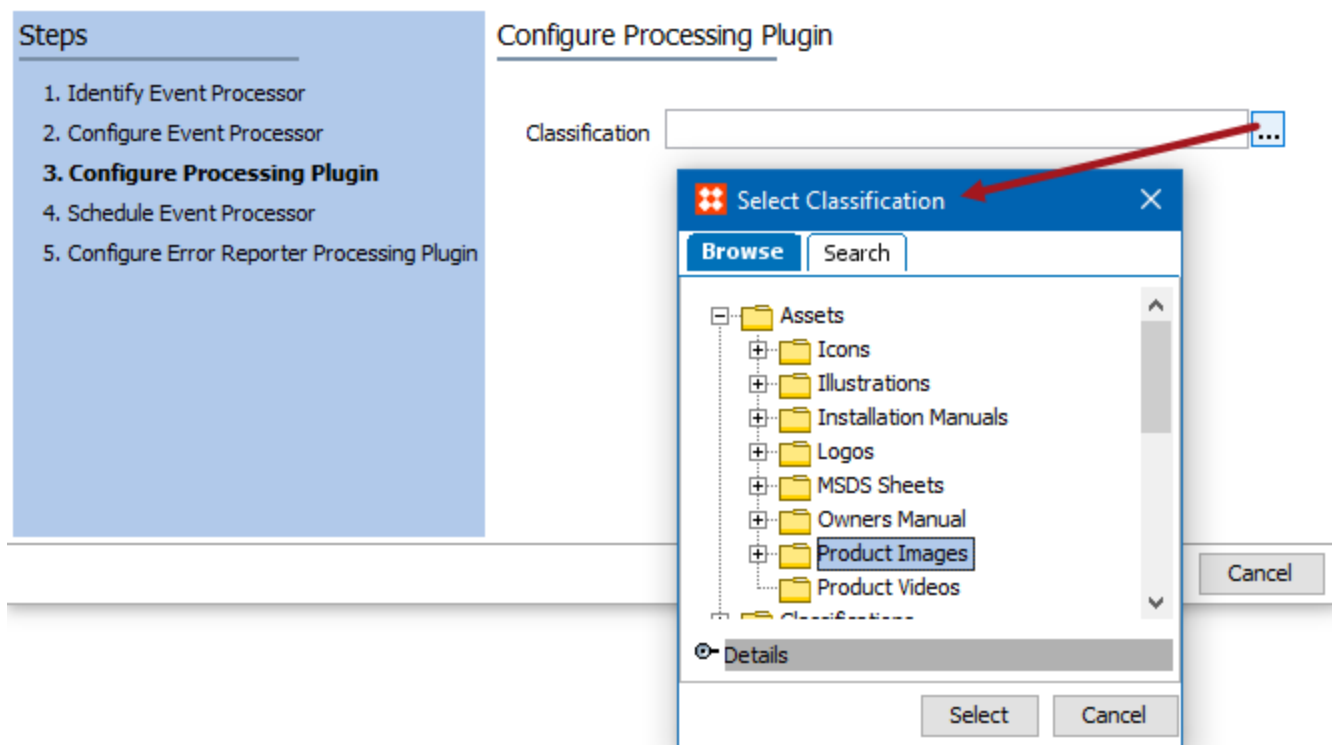


Parameters

Each of the parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below.

- **Classification:** click the ellipsis button (...) to display the Select Classification dialog. 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

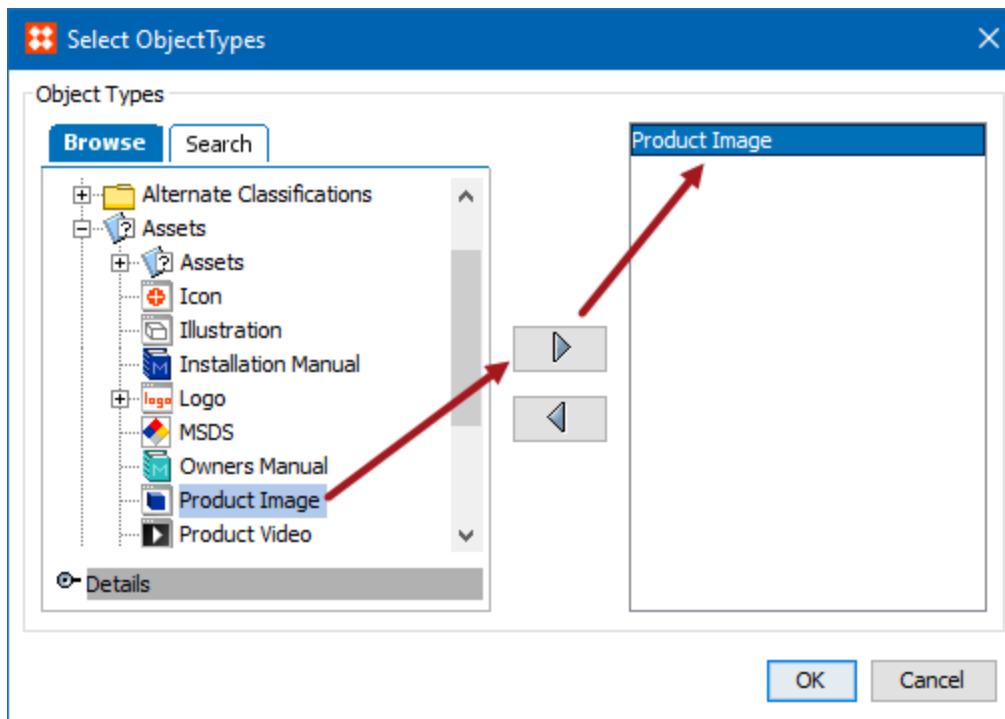


Important: Although caching images is optional, a cache improves export performance, especially for those assets that are exported frequently. Image caching is automatically defined for STEP with no setup required. All image caching is controlled by the ImageCache.* properties in the sharedconfig.properties file (any property that begins with 'ImageCache.' regardless of what text follows). Contact Stibo Systems Technical Support for assistance when modifying the default settings to meet your needs.

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.



Important: You must select at least one asset type for Triggering Object Types or nothing will be processed by the event processor.

3. Click the **OK** button to add the selection to the Triggering Object Types table.

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

- Click the ellipsis button (...) in the respective Event Filter and Generate Event fields to add any required business rules.

Matching Processing Plugin 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.

The Parameters and Event Triggers sections below contain important information on settings that should be considered when creating an event processor using this processing plugin.

Prerequisites

This section of documentation describes configuration steps for this specific processor, but that is only one part of configuring an event processor. For the full set of 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, Linking, and Merging Components** in the **Matching, Linking, and Merging** documentation.

Important: The 'Run Matching Algorithm' functionality is controlled by System Setup > Action Sets > Setup Actions > Maintain Deduplication Configurations.

Parameters

Each of the parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below.

To access the 'Configure Processing Plugin' parameters as shown below, the **Matching** processing plugin must be selected within the Select Processor parameter during the wizard step '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 4 (USER4) ...
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	Do nothing ▼

Back Next Finish Cancel

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 NAME (USER NAME) ...
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	Do nothing ▼
Number of events to batch	1000

Back Next Finish Cancel

Once the Matching processing plugin has been selected, click the **Next** button, and the wizard step 'Configure Processing Plugin' will display.

- **Event Processing to:** Select an option from the dropdown
 - **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.
- **Matching Algorithms:** Specify which matching algorithm(s) to run by clicking 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, expand the Triggering Attributes flipper and click the Add Attribute link to select the attributes used by the Match Code definition.
- If the event processor will run a matching algorithm, select the respective 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 Processing Plugin 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 using this processing plugin.

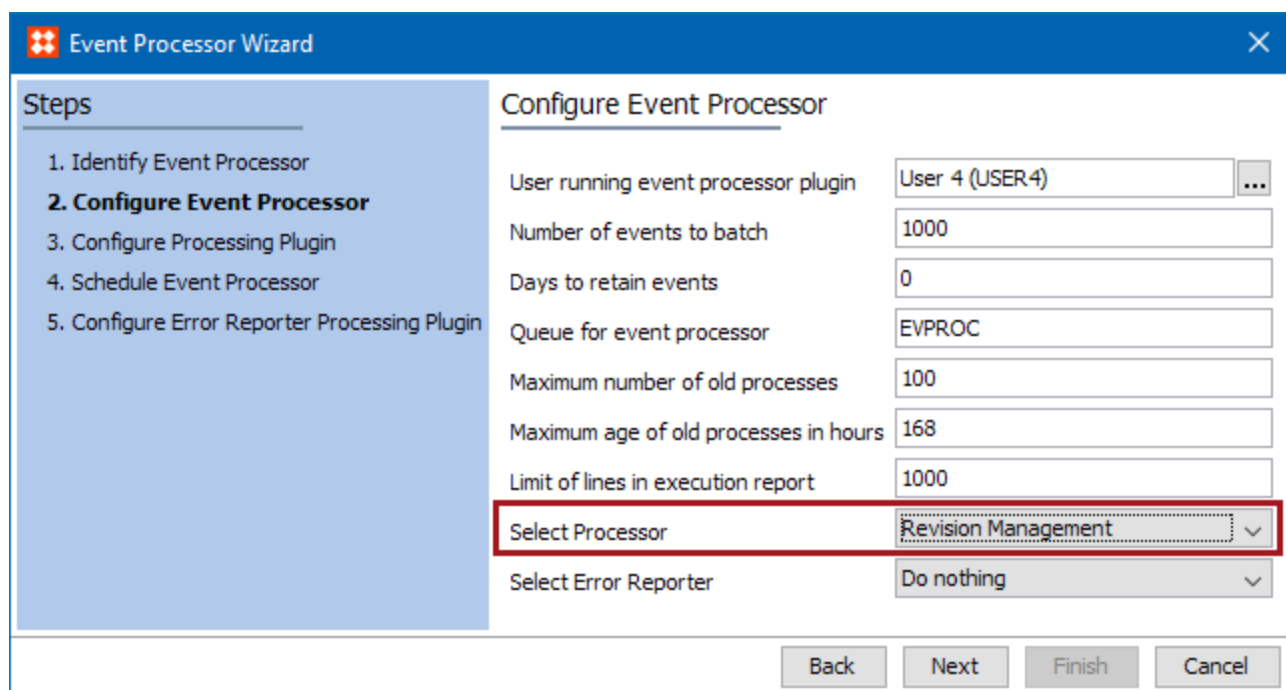
Prerequisites

This section of documentation describes configuration steps for this specific processor, but that is only one part of configuring an event processor. For the full set of instructions on configuring an event processor, see the **Event Processors** documentation.

Parameters

Each of the parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below.

To access the 'Configure Processing Plugin' parameters as shown below, the **Revision Management** processing plugin must be selected within the Select Processor parameter during the wizard step 'Configure Event Processor.'



The screenshot shows the 'Event Processor Wizard' window with the 'Configure Event Processor' step selected. The 'Steps' list on the left includes: 1. Identify Event Processor, 2. **Configure Event Processor**, 3. Configure Processing Plugin, 4. Schedule Event Processor, and 5. Configure Error Reporter Processing Plugin. The main configuration area contains the following parameters:

Parameter	Value
User running event processor plugin	User 4 (USER4)
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	Revision Management
Select Error Reporter	Do nothing

At the bottom of the window are buttons for 'Back', 'Next', 'Finish', and 'Cancel'.

Once the Revision Management processing plugin has been selected, click the **Next** button, and the wizard step 'Configure Processing Plugin' will display.

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

Back Next Finish Cancel

- **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.

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. For example, considering the values in the image above, a revision is deleted if:

1. The revision being processed is at least 731 days old, **and**
2. More than 100 revisions will remain after deletion of the revision being processed, **and**
3. The processor has not run for 30 seconds on the revision being processed, **and**
4. The revision being processed is NOT the initial revision.

Event Triggers

Select the object type(s) that should have revisions deleted according to the configured limits.

If a different number of revisions is needed based on the object type, create an additional event processor with the necessary configuration limits.

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
E

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

Initial Setup for Event Processors

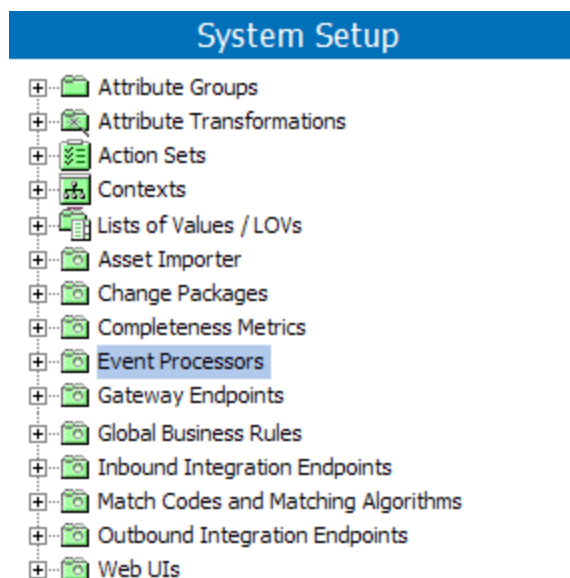
Before you can create an event processor, the following one-time setup steps must be completed. Details for each of these are provided below.

1. Create setup group type for event processors.
2. Link event processor object types to setup group types.
3. Create an event processor setup group.
4. 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 / 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.

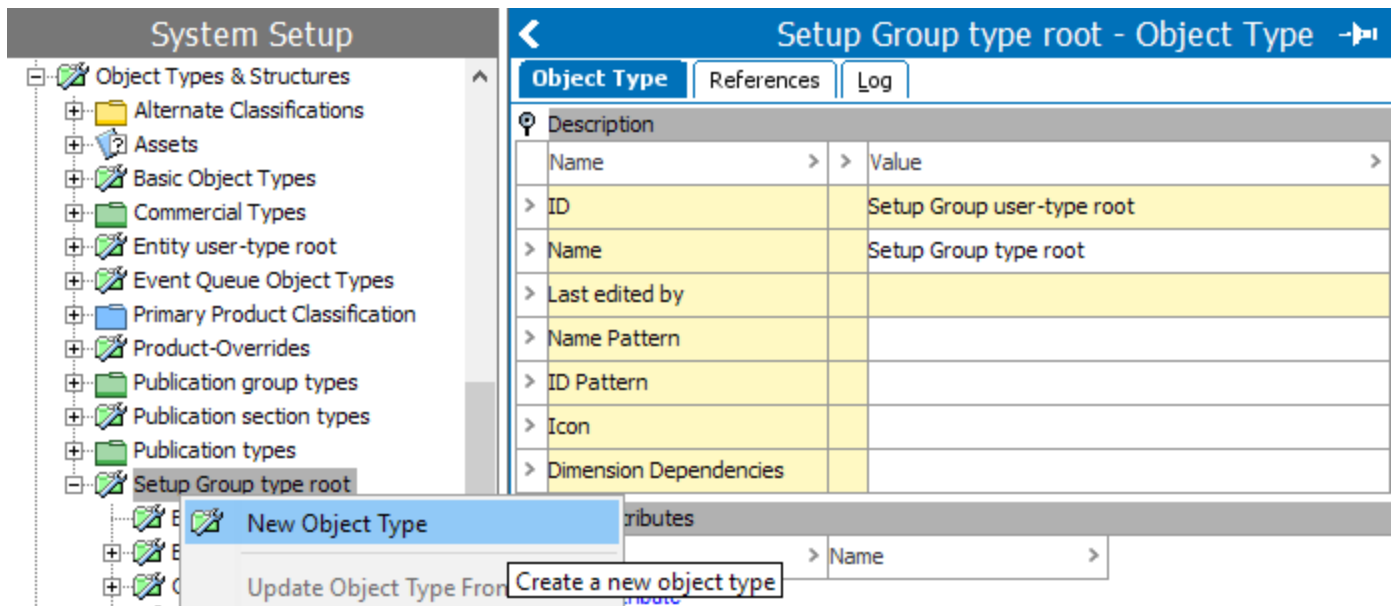


Once the setup has been completed, the steps in this section are only needed if additional levels of organization are desired.

Create Setup Group Type for Event Processors

An event processor setup group type defines the structure and allowed locations of an Event Processor.

1. Go to System Setup > Object Types & Structures > select **Setup Group type root**.
2. Right-click **Setup Group type root**, and the New Object Type option will display.



3. Click **New Object Type**, and the Create Object Type dialog will display.

Create Object Type

ID: EventProcessors

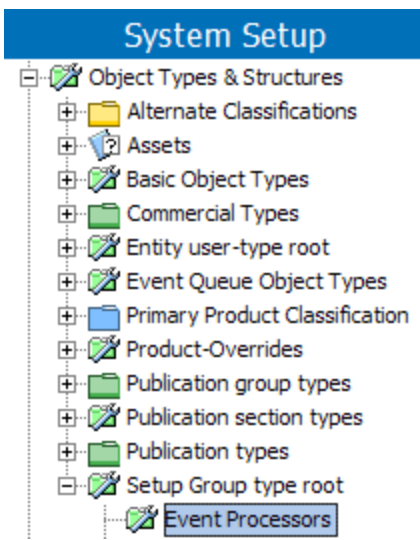
Name: Event Processors

Dimension Dependency: ☐ Country ☐ Language

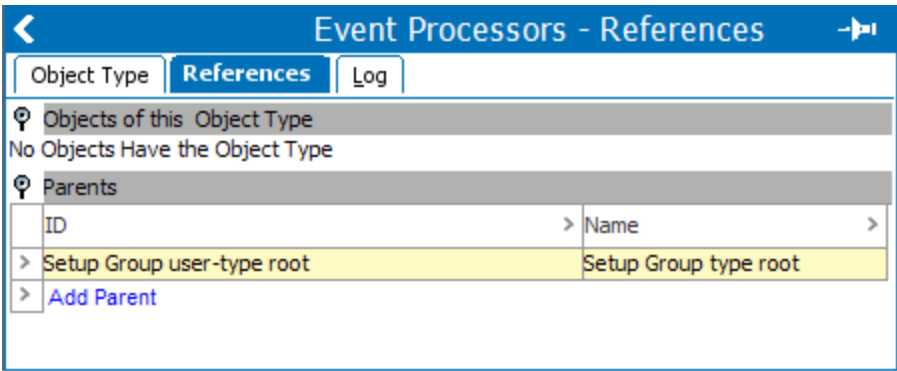
Create Cancel

4. Enter an **ID**.
5. Enter a **Name**.
6. Select any required **Dimension Dependency**.
7. Click **Create**.

The Create Object Type dialog will close, and the newly created object type for the event processors will display beneath the Setup Group type root.



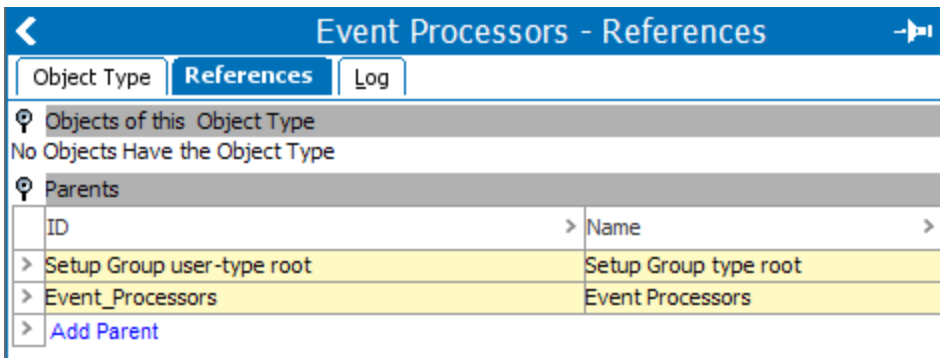
8. Select the newly added Setup Group type > References tab > open the Parents flipper.



Important: By default the Setup Group type root is listed as the parent. Optionally add the newly created setup group type as a parent of itself so that additional event processors group types can be added below the main level.

- 9. Click **Add Parent**, and the Select New parent dialog will display.
- 10. Browse or search to select **the relevant setup group type**.
- 11. Click the **Select** button.

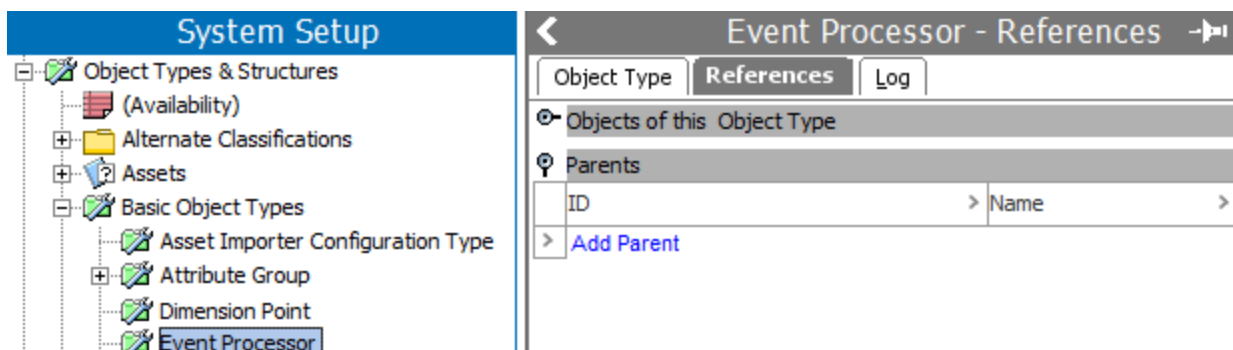
The dialog will close, and the newly created setup group type (i.e., Event Processors) will be listed as a parent along with the Setup group user-type root.



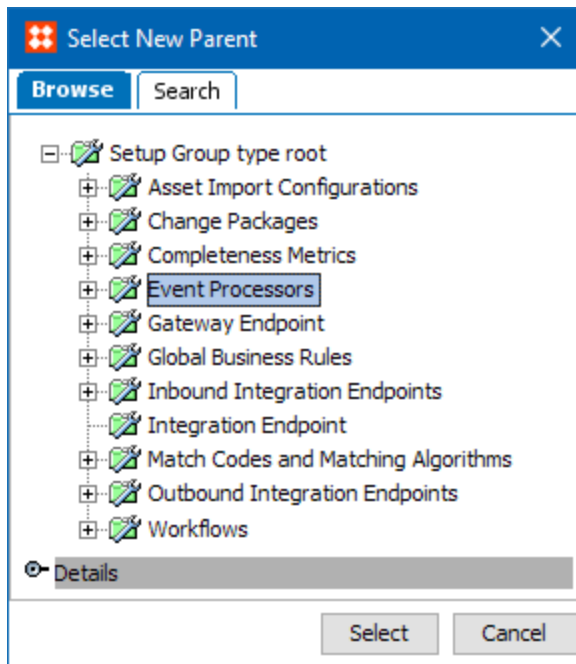
Link Event Processor Object Types to Setup Group Types

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.



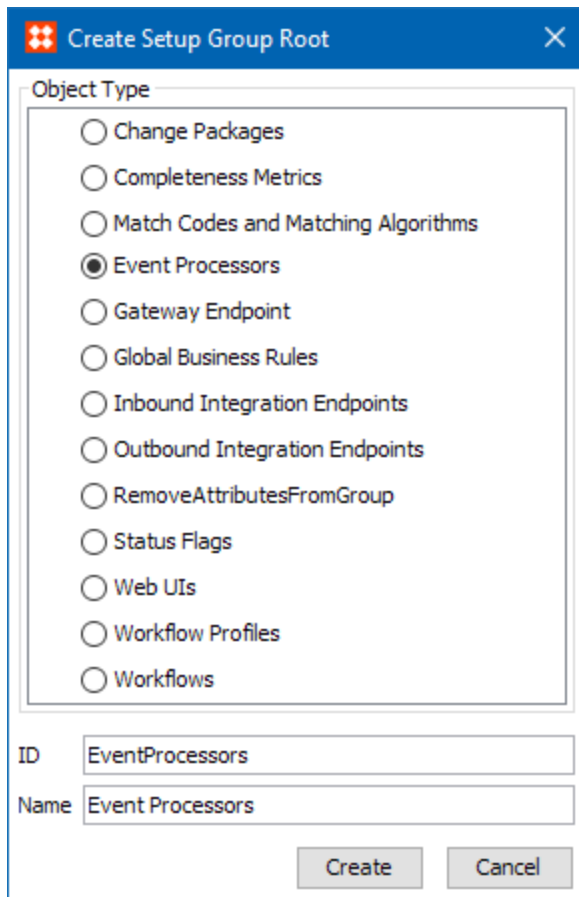
3. Click the **References** tab.
4. Open the **Parents** flipper.
5. Click the **Add Parent** link, and the Select New Parent dialog will display.
6. Browse or search to select **the relevant setup group type**.
7. Click the **Select** button.



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. "Event Processors" is selected. Below the list are fields for "ID" and "Name", both containing the text "EventProcessors". At the bottom are "Create" and "Cancel" buttons.

Create Setup Group Root

Object Type

- ☐ Change Packages
- ☐ Completeness Metrics
- ☐ Match Codes and Matching Algorithms
- ☒ Event Processors
- ☐ Gateway Endpoint
- ☐ Global Business Rules
- ☐ Inbound Integration Endpoints
- ☐ Outbound Integration Endpoints
- ☐ RemoveAttributesFromGroup
- ☐ Status Flags
- ☐ Web UIs
- ☐ Workflow Profiles
- ☐ Workflows

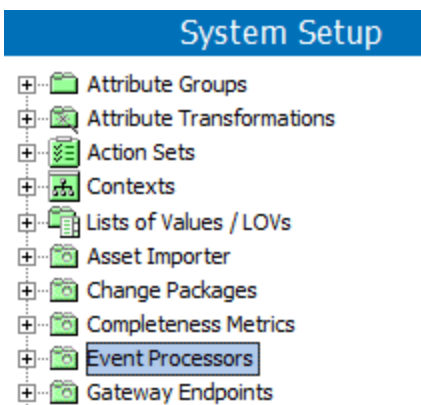
ID: EventProcessors

Name: Event Processors

Create Cancel

3. Select **your event processors object type**.
4. Enter an **ID**.
5. Enter a **Name**.
6. Click **Create**.

The setup group is created and appears as a node in the System Setup hierarchy.



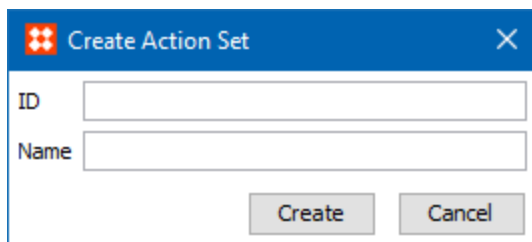
Configure Action Sets and Privileges

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 an event processor specific action set. For example, if you have users who are allowed to maintain event processors 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 the **Select** button.

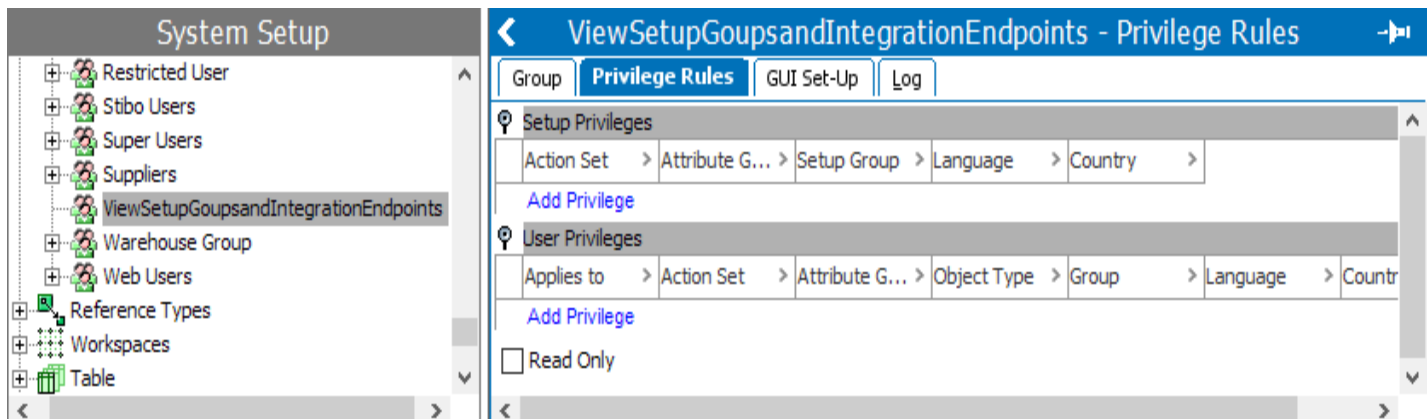
Some commonly used Actions are described below:

- **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.

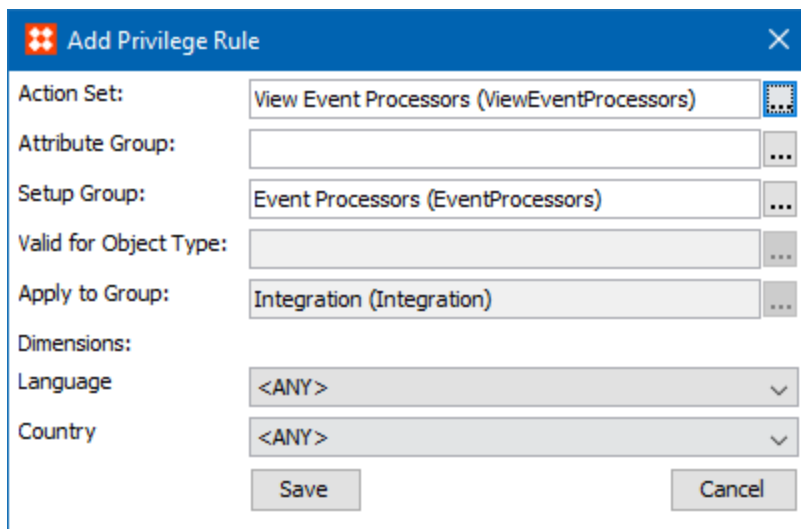
Configure Event Processor Privileges

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.



- 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 **Action Sets**.

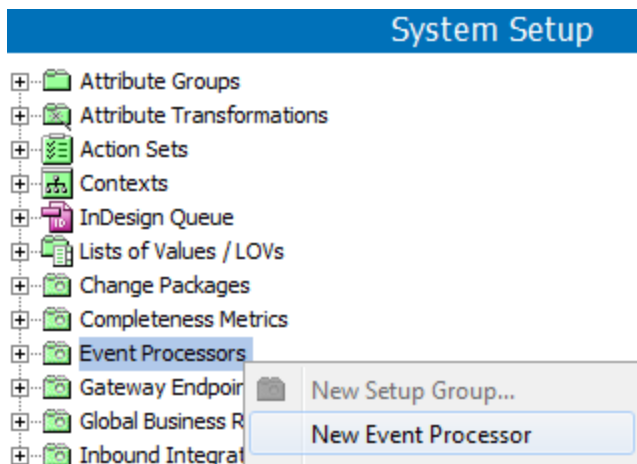
Creating an Event Processor

Prior to creating an event processor, the one-time setup tasks (as described in **Initial Setup for Event Processors** documentation) must be preformed to allow for the creation of event processors. After creating a setup group for event processors, create an event processor to monitor changes in STEP, and utilize the functionality of a specific processor.

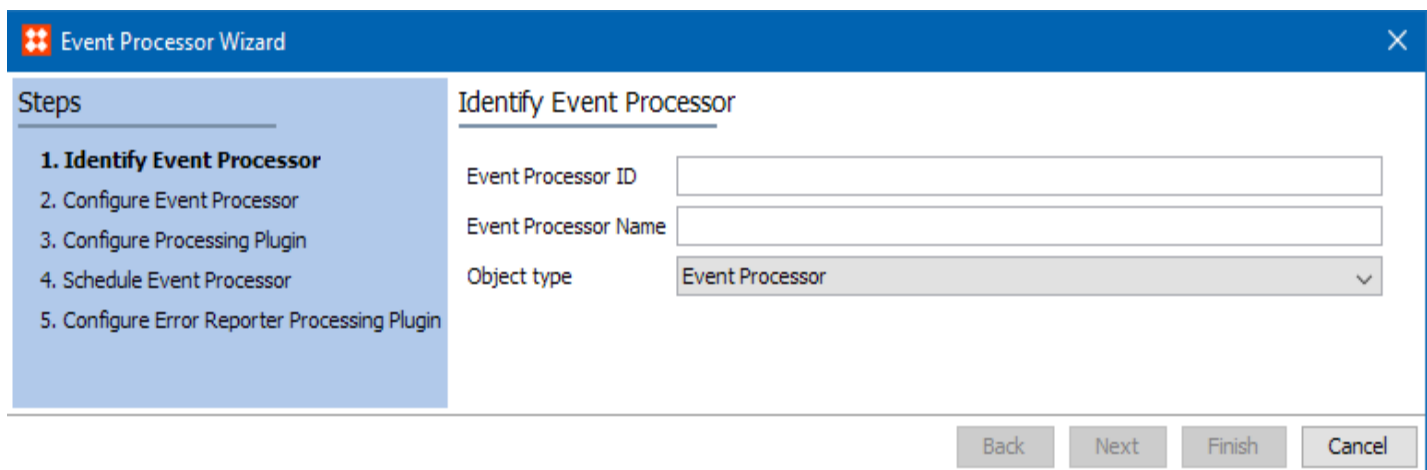
An event processor is created in the workbench System Setup using an event processor wizard. Below are general steps for creating a new event processor. However, the event processor wizard 'Configure Processing Plugin' dialog will vary based upon the processing plugin selected. Information for configuring each of the processing plugins can be found in the **Processing Plugins** documentation.

To create a new event processor:

1. Go to **System Setup**, right-click **your event processors setup group**, and click **New Event Processor**.



2. The Event Processor Wizard displays and involves the following steps:



Complete setup requires providing data in the wizard as well as some manual configuration. All setup includes the following sections:

1. In the wizard, specify an ID, name, and object type as described in **EPW - Identify Event Processor**.
2. In the wizard, specify the processing and error handling as described in **EPW - Configure Event Processor**.
3. In the wizard, specify the information specific to the event processor selected in the previous step as described in **EPW - Configure Processing Plugin**.
4. In the wizard, specify how often the event processor should start as described in **EPW - Schedule Event Processor**.
5. In the wizard, populate the send report to address parameter as described in **EPW - Configure Error Reporter Processing Plugin**.
6. Confirm / Update triggering definitions as described in **Event Triggering Definitions Tab** documentation.
7. Set Queue Status Events to 'Read Events' as described in **Event Processor Queue Status**.
8. Enable the event processor as described in the **Enable Event Processor** section of the **Running an Event Processor** documentation.

EPW - Identify Event Processor

To complete the Event Processor Wizard Identify Event Processor steps:

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

Identify Event Processor

Event Processor ID

Event Processor Name

Object type

Back Next Finish Cancel

1. Enter an ID without spaces. (IDs should never contain spaces, especially leading or trailing spaces)
2. Enter a Name for the event processor. The name will be displayed as the name of the background process.
3. Confirm the object type parameter is set to the default event processor object type.
4. Once any required changes are made, click the **Next** button to display **EPW - Configure Event Processor**.

EPW - Configure Event Processor

Configure the new event processor by populating the available parameters. Common setup is to use the default settings unless there is a valid reason to change them.

For more information on each of the parameters, see the **Maintaining an Event Processor** documentation.

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 L (USERL) ...
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	Do nothing

Back Next Finish Cancel

- **User running event processor plugin:** Click the ellipsis button (...) to search or browse for a user.

Important: The privileges of the selected user determine which actions the Event processor can perform and what data can be processed. Common setup is to create a special system user for this purpose so that the effects of the event processor are easily identified.

- **Number of events to batch:** Specify the batch size. The events are handled one by one, but will be more effective and ensure increased performance by committing changes in larger batches. If you are experiencing slow processing times, try increasing your batch size.
- **Days to retain events:** Optionally enter 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. 'EVPROC' is the default, however entering the name of a new queue will result in the new queue automatically being created upon completion of the wizard.
- **Maximum number of old processes:** Specify the number of ended processes the system will keep.
- **Maximum age of old processes in hours:** Specify the maximum age of ended processes that the system will keep.
- **Limit of lines in execution report:** Specify the maximum number of lines to store from the execution report in the log.
- **Limit of lines in execution report:** Specify the maximum number of lines to store from the execution report in the log.
- **Select Processor:** Choose from the available options, for more information, see the **Processing Plugins** documentation.
- **Select Error Reporter:** Choose either 'Do nothing' or 'Mail Error Report.'
 - If 'Mail Error Report' is selected, additional information must be supplied in step 5 of the wizard.
 - If 'Do Nothing' is selected, then step '5. Configure Error Reporter Processing Plugin.' will appear blank.

Once any required configurations are made, click the **Next** button to display the wizard step 'Configure Processing Plugin'. See the **EPW - Configure Processing Plugin** topic.

EPW - Configure Processing Plugin

When creating an event processor to monitor events within STEP, a processor must be selected and then the processing plugin must be configured.

The options / parameters displayed within the Event Processor Wizard step 'Configure Processing Plugin' will vary depending on the processor selected during the previous step, 'Configure Event Processor.'

The screenshot below displays an example of the Event Processor Wizard step where the Select Processor parameter displays a dropdown list of available processors.

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 NAME (USER NAME) ...

Number of events to batch: 1

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: Expand LOV Value Changes ▼

Select Error Reporter:

Expand LOV Value Changes

Business Action Event Processor

Experian Email Validation

Image Cache

Matching

Revision Management

Back Next Finish Cancel

Once the Select Processor parameter is populated, and the Next button is clicked, the Event Processor Wizard step 'Configure Processing Plugin' relative to the selected processor will display.

The screenshot below displays an example of the Event Processor Wizard step 'Configure Processing Plugin' that displays when the Revision Management processor is selected.

The screenshot shows the 'Event Processor Wizard' window. On the left, a 'Steps' list contains five items: '1. Identify Event Processor', '2. Configure Event Processor', '3. Configure Processing Plugin' (which is highlighted in blue), '4. Schedule Event Processor', and '5. Configure Error Reporter Processing Plugin'. The main area is titled 'Configure Processing Plugin' and contains four input fields: 'Days to keep' with the value '730', 'Number to keep' with the value '100', 'Processing time' with the value '30', and 'Keep initial revision' with a dropdown menu set to 'Yes'. At the bottom, there are four buttons: 'Back', 'Next' (which is highlighted with a dashed border), 'Finish', and 'Cancel'.

Information on each of the processors and their respective important parameters and event triggers that should be considered when configuring the event processor and processing plugin can be found below.

- **CASS Certification Report Processing Plugin Parameters and Triggers**
- **Execute Business Action Processing Plugin Parameters and Triggers**
- **Expand LOV Value Changes Processing Plugin Parameters and Triggers**
- **Experian Email Validation Processing Plugin Parameters and Triggers**
- **Image Cache Processing Plugin Parameters and Triggers**
- **Matching Processing Plugin Parameters and Triggers**
- **Revision Management Processing Plugin Parameters and Triggers**

Once any required configurations are made, click the **Next** button to display the wizard step 'Schedule Event Processor.'

See the **EPW - Schedule Event Processor** topic.

EPW - Schedule Event Processor

The wizard step 'Schedule Event Processor' will display with the following Start options.

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

Schedule Event Processor

Start

☒ Never

☐ Every

☐ Weekly

☐ Later

Start Not scheduled

Back Next Finish Cancel

- **Never** - invoke the event processor manually, no additional parameters are required, and no schedule is applied. This is the default setting and should be used while testing your endpoint.
- **Every** - automatically run the event processor repeatedly, every selected number of minutes. One (1) minute is the shortest interval allowed and is closest to real time. Enter the number of minutes in the text box. The selection is summarized at the bottom of the dialog.

☐ Never

☒ Every

☐ Weekly

☐ Later

1 minutes

Start every minute

- **Weekly** - automatically run the endpoint repeatedly, based on the selected time, start and end dates, and days of the week. Use this option if a daily schedule is needed. The 'Start at' parameter determines the time of day that the endpoint will run. The 'Start on' parameter determines the date the endpoint will first run, while the 'End on' parameter determines the date of the endpoint's final run. The 'Every' checkboxes determine the days of the week when the endpoint will run. The selections are summarized at the bottom of the dialog.

☐ Never	Start at (hh:mm):	21:10
☐ Every	Start on (yyyy-mm-dd):	2017-07-06
☒ Weekly	End on (yyyy-mm-dd):	-
☐ Later	Every:	☒ Mon ☒ Sat ☐ Tue ☐ Sun ☒ Wed ☐ Thu ☐ Fri

Start every Mon, Wed, Sat 21:10:00 EDT, starting Thu Jul 06 2017

- **Later** - automatically run the event processor only once, at the time and date specified. The selections are summarized at the bottom of the dialog.

☐ Never	Start at (hh:mm):	20:30
☐ Every	Start on (yyyy-mm-dd):	2017-07-06
☐ Weekly		
☒ Later		

Start at Thu Jul 06 20:30:00 EDT 2017

Important: Selecting a 'Schedule Event Processor' setting will **not** enable / start the event processor. Additional steps are required after completing the wizard to enable / start the event processor. The selection within this step can be edited later. Choosing an option here will simply set the schedule for the event processor *after* the additional configuration steps have been completed. Optionally, you can invoke the event processor manually, if you need to immediately publish the events to a downstream system and do not want to wait for the schedule.

Once any required configurations are made, click the **Next** button to display the wizard step 'Configure Error Reporter Processing Plugin.'

See the **EPW - Configure Error Reporter Processing Plugin** topic.

EPW - Configure Error Reporter Processing Plugin

If the 'Select Error Reporter' option was set to 'Mail Error Report,' then the wizard step 'Configure Error Reporter Processing Plugin' will display with the following options.

The screenshot shows the 'Event Processor Wizard' dialog box. The title bar is blue with the Stibo logo and the text 'Event Processor Wizard'. The main area is divided into two panes. The left pane, titled 'Steps', contains a list of five steps: 1. Identify Event Processor, 2. Configure Event Processor, 3. Configure Processing Plugin, 4. Schedule Event Processor, and 5. Configure Error Reporter Processing Plugin. Step 5 is highlighted with a blue background. The right pane, titled 'Configure Error Reporter Processing Plugin', contains a single text input field labeled 'Send report to address' with the value 'user@systems.com'. At the bottom of the dialog are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

Note: If the 'Select Error Reporter' option was not set to 'Mail Error Report,' then the 'Send report to address' option will not display. Use the Back button to return to the 'Configure Event Processor' step, and edit the selection.

- **Send report to address:** Optionally enter an email address (use a semi-colon to add more than one email address) or distribution list that can be used to receive error reports.
-

Important: For the Error reporter plugin to send email, the SMTP server must be configured on the application server.

Click the **Finish** button to complete the wizard, and the workbench will display the event processors grouping expanded with the newly created event processor selected, thus displaying the event processor editor.

System Setup

- Attribute Groups
- Attribute Transformations
- Action Sets
- Contexts
- InDesign Queue
- Lists of Values / LOVs
- Asset Importer
- Change Packages
- Completeness Metrics
- Event Processors
 - Email User When Attribute Value Changes
 - Execute Business Action
 - External Item Async Matching
 - Find Existing
 - Find Similar Search
 - 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

Email User When Attribute Value Changes - Event Processor

Event Processor
Event Triggering Definitions
Background Processes
Statistics
Error Log Excerpts
Log

Description

Name	Value
ID	EmailUserWhenAttributeValueChanges
Name	Email User When Attribute Value Changes
Type	Event Processor
Last edited by	2017-01-31 15:10:20 by USERM
Enabled	No
Processor Status	Stopped

Configuration

ID	Name
User running event processor plugin	USER NAME
Number of events to batch	1
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
Processor	Execute Business Action
Schedule	Not scheduled
Queue Status	Discard Events
Unread events (approximated)	Click to estimate ...

[Edit Configuration](#)

Current Background Process Log

No matter the selections made during the wizard, new event processors are created with the following 'Event Processor' tab parameter settings:

- **Enabled:** No
- **Processor status:** Stopped
- **Queue Status:** Discard Events

Also, it is important to note that all event processors are created with the following settings enabled within the 'Miscellaneous Triggers' flipper on the 'Event Triggering Definitions' tab:

- Names enabled
- Parent links enabled

- Attribute-links enabled
- Index-word Hierarchy enabled

< Email User When Attribute Value Changes - Event Triggering Definitions **→**

Background Processes | **Statistics** | Error Log Excerpts | Log

Event Processor | **Event Triggering Definitions**

Triggering Object Types

Object Types > Event Filter > Generate Event >

[Add Object Type](#)

Triggering Attributes

Name >

[Add Attribute](#)

Triggering Table Types

Table Types >

[Add Table Type](#)

Reference Type Triggers

Reference Types >

[Add Reference Type](#)

Miscellaneous Triggers

☒ Names enabled

☒ Parent links enabled

☒ Attribute-links enabled

☒ Index-word Hierarchy enabled

Important: Before the event processor can be used, the event triggering definitions must be set, see the **Event Triggering Definitions Tab** documentation.

Running an Event Processor

After configuring an event processor, it must be enabled before it can run on schedule or be invoked manually. The status of an event processor is very important. This section addresses the different event processor status options and how to enable, invoke, or disable an event processor.

Event Processor Status Options

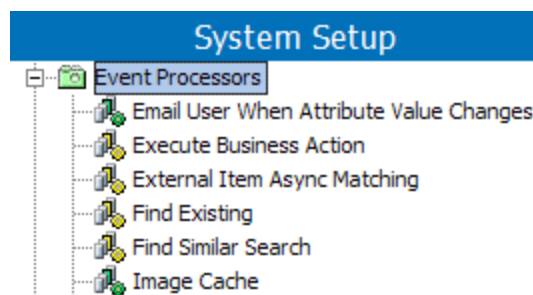
An event processor can be in one of three statuses: Enabled, Disabled, and Stopped.

The table below displays each icon and a description of each event processor status.

Icons	Description
	Enabled: 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.

When viewing a list of event processors within STEP Workbench > System Setup the status of each processor can easily be identified by the color of the icon.

In the example below, a list of event processors is displayed within the Workbench > System Setup > Event Processors. Using the event processor status icons makes it easy to decipher the status of each of the event processors.



The event processor status information is also displayed in text on the Event Processor tab of the editor. For more information see the **Event Processor Tab** documentation.

When configuring and managing event processors, it is important to monitor their status, because an event processor only functions when it is enabled.

Enable Event Processor

When an event processor is created, it is in a stopped or disabled status, and must be enabled to be of use.

The steps below describe how to enable an event processor when it has been disabled or stopped because of failure.

1. Go to System Setup > expand the Setup Group created for event processors > select the **Event Processor you want to enable**.
2. Right-click the event processor, and click **Enable Event Processor**.

The screenshot shows the 'System Setup' tree on the left with 'Event Processors' expanded. The 'Email User When Attribute Value Changes' event processor is selected. The right pane shows the details for this event processor, including its ID, Name, Type, Last edited by, and Enabled status (No). A context menu is open over the event processor, with 'Enable Event Processor' highlighted. The menu also includes options like 'Disable Event Processor', 'Edit Event Processor...', 'Invoke Event Processor', 'View First Event Batch...', 'Forward Events', 'Rewind Events', 'Purge Events', 'Republish', 'Cut', 'Copy', 'Paste', and 'Delete'.

Name	Value
ID	EmailUserWhenAttributeValueChanges
Name	Email User When Attribute Value Changes
Type	Event Processor
Last edited by	2017-01-31 15:10:20 by USERM
Enabled	No
Status	Stopped

plugin	Name
1	USER NAME
0	
	EVPROC
100	
168	
1000	
	Execute Business Action
	Not scheduled
	Discard Events
	Click to estimate ...

3. Confirm the event processor is now enabled by viewing the

- Enabled icon  next to the specific event processor.
- Enabled parameter is set to 'Yes' within the event processor editor.
- Processor Status parameter is set to 'Running', and highlighted green within the event processor editor.

The screenshot shows the 'System Setup' window. On the left, under 'Event Processors', the 'Email User When Attribute Value Changes' processor is selected, indicated by a red arrow. On the right, the configuration details for this processor are displayed. The 'Event Processor' tab is active, showing a table with the following information:

Event Processor	
ID	EmailUserWhenAttributeValueChanges
Name	Email User When Attribute Value Changes
Type	Event Processor
Last edited by	2017-01-31 15:10:20 by USERM
Enabled	Yes
Processor Status	Running

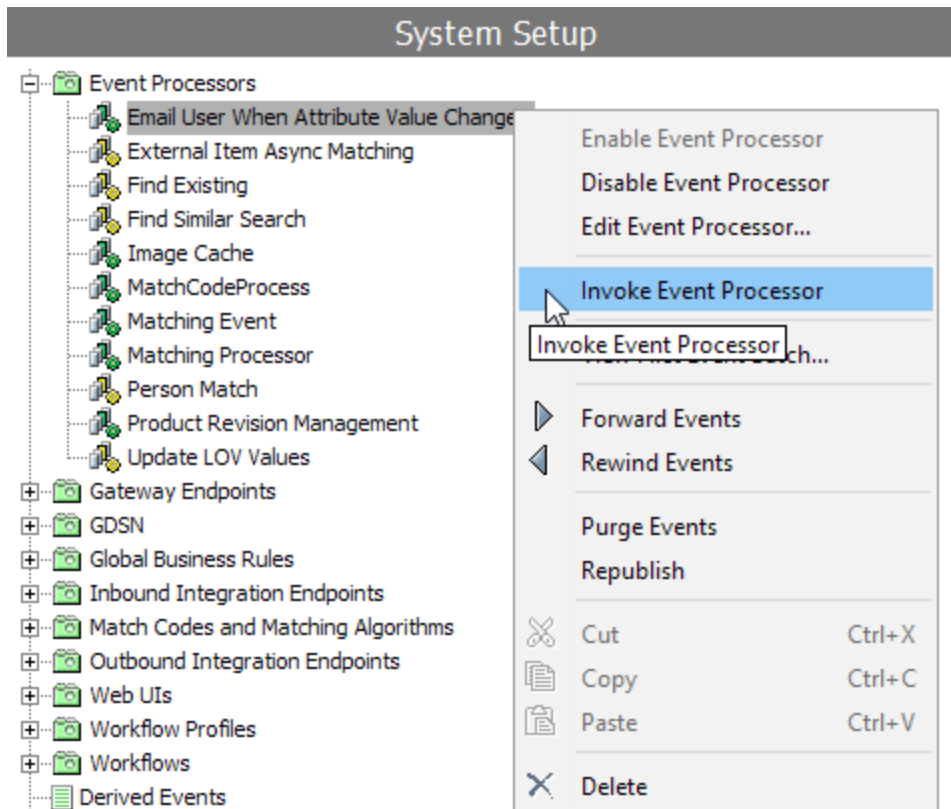
Important: If you are ready for the processor to apply the actions based on the events processed, make sure the Queue Status parameter is also set to Read Events, otherwise the enabled event processor will not function. For more information see the **Event Processor Queue Status** documentation.

Invoke Event Processor

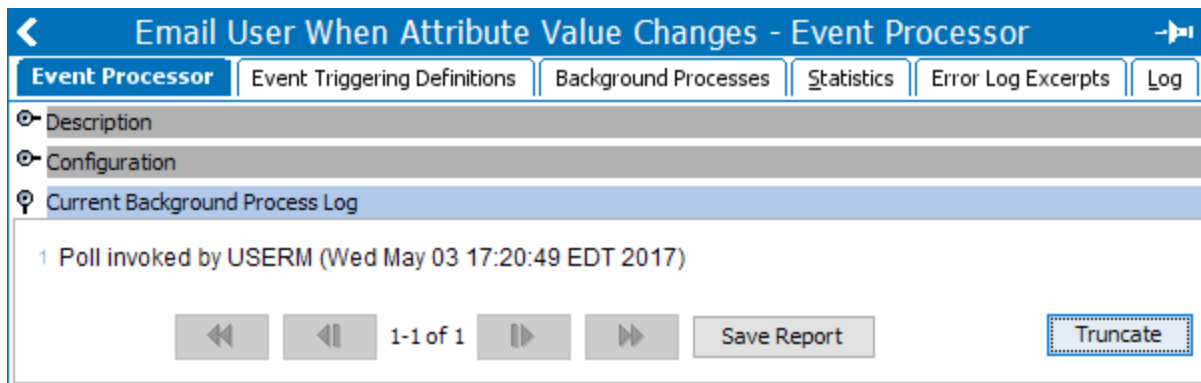
An event processor can only be invoked if its status is Enabled.

Steps to invoke an event processor are below.

1. On the System Setup tab, select the event processor to be invoked.
2. Right-click to display and select the **Invoke Event Processor** option.



3. If successful, the Current Background Process Log will display the following 'Poll invoked by (user name) (Date and Time Stamp)'.



Disable Event Processor

If an event processor is no longer necessary, or needs to be stopped for a period of time, then a user can disable it, and if necessary enable it at a later date. The option to disable an event processor is only available when an event processor is in the enabled status.

The steps below describe how to disable an event processor in an enabled status.

1. Go to System Setup > expand the Setup Group created for Event processors > select the **Event Processor you want to enable**.
2. Right-click the event processor, and click **Disable Event Processor**.

System Setup

- Event Processors
 - Email User When Attribute Value Changes
 - Execute Business Action
 - External Item Async Matching
 - Find Existing
 - Find Similar Search
 - 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

Email User When Attribute Value Changes - Event Processor

Event Processor | Event Triggering Definitions | Background Processes | Statistics | Error Log Excerpts | Log

Enable Event Processor
Disable Event Processor
 Edit Event Processor...
 Invoke Event Processor
 View First Event Batch...
 Forward Events
 Rewind Events
 Purge Events
 Republish
 Cut (Ctrl+X)
 Copy (Ctrl+C)
 Paste (Ctrl+V)
 Delete

EmailUserWhenAttributeValueChanges
Email User When Attribute Value Changes
2017-01-31 15:10:20 by USERM
Yes
Running

Name
USER NAME
1
0
EVPROC
100
168
1000
Execute Business Action

3. Confirm the event processor is now disabled by viewing the:
 - Disabled icon next to the specific event processor.
 - Enabled parameter within the event processor editor is set to 'No.'
 - Processor Status parameter within the event processor editor is set to 'Stopped,' and highlighted yellow.

System Setup

- Event Processors
 - Email User When Attribute Value Changes
 - Experian Email Validation
 - External Item Async Matching
 - Find Existing
 - Find Similar Search
 - Image Cache
 - Loqate
 - LOV Value Merge Events
 - MatchCodeProcess
 - Matching Event
 - Matching Processor

Email User When Attribute Value Changes - Event Processor

Event Processor | Event Triggering Definitions | Background Processes

Description

Name	Value
ID	EmailUserWhenAttributeValueChanges
Name	Email User When Attribute Value Changes
Type	Event Processor
Last edited by	2017-05-30 14:35:24 by USER3
Enabled	No
Processor Status	Stopped

Maintaining an Event Processor

Once an event processor is created, it can be viewed and edited via the event processor editor. This section describes how to access an event processor editor, details for each of the event processor editor tabs, and how to edit and delete an event processor.

To access an event processor editor go to workbench > System Setup > Event Processors and select **your event processor**.

The event processor editor will display with the following tabs:

- Event Processor
- Event Triggering Definitions
- Background Processes
- Statistics
- Error Log Excerpts
- Log

System Setup

Event Processors

- Email User When Attribute Value Changes
- Execute Business Action
- External Item Async Matching
- Find Existing
- Find Similar Search
- Image Cache
- MatchCodeProcess
- Matching Event
- Matching Processor
- Person Match
- Product Revision Management

Email User When Attribute Value Changes - Event Processor

Event Processor | Event Triggering Definitions | Background Processes | Statistics | Error Log Excerpts | Log

Description

Name	>	>	Value	>
ID	>		EmailUserWhenAttributeValueChanges	
Name	>		Email User When Attribute Value Changes	
Type	>		Event Processor	
Last edited by	>		2017-01-16 15:09:24 by USERM	
Enabled	>		Yes	
Processor Status	>		Running	

Configuration

ID	Name
User running event processor plugin	USER NAME
Number of events to batch	1
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
Processor	Execute Business Action
Schedule	Start every minute
Queue Status	Read Events
Unread events (approximated)	Click to estimate ...

[Edit Configuration](#)

Current Background Process Log

1	Next poll scheduled for Tue Jan 31 02:31:35 EST 2017 (Tue Jan 31 02:30:35 EST 2017)
2	Checked and found no new events. (Tue Jan 31 02:31:38 EST 2017)

Edit Event Processor

There are many reasons and methods to edit an event processor.

Steps to edit an event processor are below.

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:

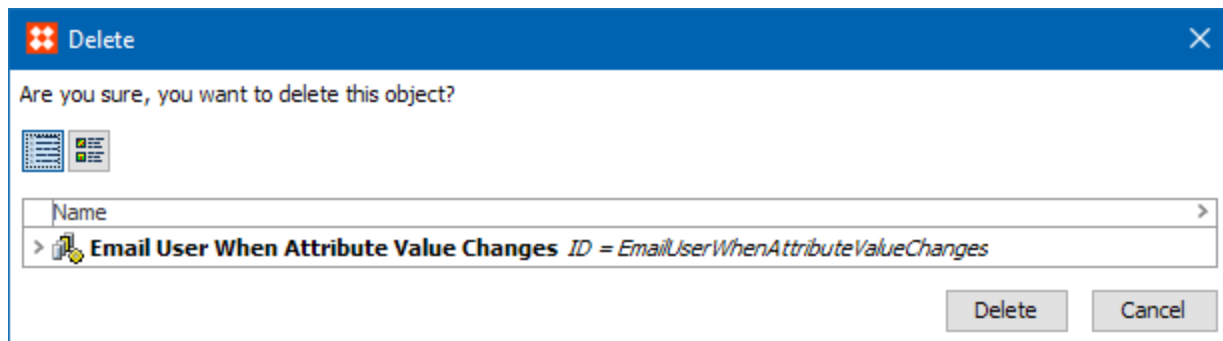
- Right-click the event processor and select **Edit Event Processor**.
- Expand the Configuration flipper, click the **Edit Configuration** link, and the event processor wizard will display.
- Change Event Triggering Definitions on the editor's **Event Triggering Definitions** tab.

Delete Event Processor

An event processor can only be deleted if its status is Disabled. Once the event processor is deleted, it does **not** move to the recycle bin, it is completely deleted from the system.

Steps to delete an event processor are below.

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, select **Delete**, and the Delete confirmation will display



3. Verify the correct event processor is being deleted, and click the **Delete** button.

Event Processor Tab

The Event Processor tab contains information about the description, configuration, and current background process log.

Prerequisites

The Event Processor tab is part of the event processor editor. Information about the event processor editor can be found in the **Maintaining an Event Processor** documentation.

Description Flipper

The Description flipper includes basic information to identify the event processor. The name can be edited. This data is originally set up in the event processor wizard, see **EPW - Identify Event Processor** documentation.

For information on the processor status, see **Running an Event Processor** documentation.

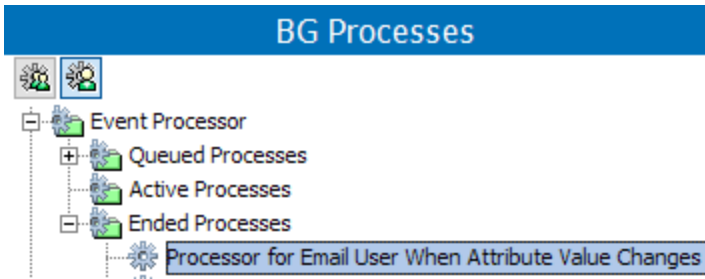
Event Processor		Event Triggering Definitions	Background Processes	Statistics	Error Log Excerpts	Log
Description						
	Name	>	>	Value		>
>	ID			EmailUserWhenAttributeValueChanges		
>	Name			Email User When Attribute Value Changes		
>	Type			Event Processor		
>	Last edited by			2017-01-16 15:09:24 by USERM		
>	Enabled			Yes		
>	Processor Status			Running		

ID

- Established during the event processor creation, using the Event Processor Wizard.
- Cannot be edited.

Name

- Established during the event processor creation, using the Event Processor Wizard.
- Can be edited.
- Appears as the processor description when viewing background processes.



Type

- The object type for the event processor in this example is 'Event Processor'.
- The object type for your instance of STEP may be different, because the event processor object type is configured during the one-time setup tasks described in **Initial Setup for Event Processors**.
- Cannot be edited.

Last edited by

- Displays the Date, Time, and User name responsible for the last edit.
- Cannot be edited.

Enabled

- Displays the current event processor state.
- When enabled the parameter displays 'Yes', the Processor Status parameter automatically displays 'Running'.
- When disabled the parameter displays 'No', the Processor Status parameter automatically displays 'Stopped'.
- A processor must be enabled for it to be used in STEP.
- More information about event processor states and how to enable / disable them can be found in the **Running an Event Processor** documentation.

Processor Status

- Displays the event processor's current state.

Configuration Flipper

Details for each of the Configuration flipper parameters are below. To edit any of the parameters within the 'Configuration' flipper, select **Edit Configuration**, located at the bottom of the list, and the Event Processor Wizard will display step '2. Configure Event Processor.' The navigation buttons in the wizard can be used to access the other steps, if necessary.

Event Processor		Event Triggering Definitions	Background Processes	Statistics	Error Log Excerpts	Log
Description						
Configuration						
ID	Name					
User running event processor plugin	USER NAME					
Number of events to batch	1					
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					
Processor	Execute Business Action					
Schedule	Not scheduled ...					
Queue Status	Discard Events					
Unread events (approximated)	Click to estimate ...					
Edit Configuration						

User running event processor plugin

- 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.
- 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

- 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.
- Specify the number of days to keep events once processed.

Queue for event processor

- During the create event processor wizard, the name of a queue is entered determining where to process data from the event processor.
- 'EVPROC' is the default, however entering the name of a new queue will result in the new queue automatically being created upon completion of the wizard.

Maximum number of old processes in hours

- Indicates 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 usage.

Processor

- Displays the processing plugin currently selected for use by the event processor.
- For more information, see the **Processing Plugins** documentation.

Schedule

- Indicates the current schedule for running the event processor.
- 'Not scheduled' indicates the event processor is not currently scheduled to run automatically.

Queue status

- 'Read Events' allows the processor to register events as they occur, based on the event processor configuration and event triggering definitions.
- 'Discard Events' ignores generated events that meet the configuration and event triggering definitions. This is useful when testing a processor without modifying data.
- By default, when a new event processor is created, events are discarded.
- Select **Read Events** if you are ready for the processor to apply the actions based on the events processed.

Unread events (approximated)

- Displays the 'Click to estimate ...' button.
- Selecting the 'Click to estimate ...' button will estimate the number of unread events and the button will be replaced with the results and a date / time stamp.
- Refreshing the editor will allow for the 'Click to estimate ...' button to display again.

Current Background Process Log Flipper

The Current Background Process Log flipper is blank until the event processor state is enabled. Once enabled, information about when the event processor is expected to run and/or what occurred when the processor ran will display.

Using the example below, line 1 indicates the event processor was enabled, but not scheduled to run. Line 2 indicates the event processor is scheduled, and the subsequent lines indicate the processor is running per the Schedule parameter settings of 'Start every minute'.

Email User When Attribute Value Changes - Event Processor

Event Processor | Event Triggering Definitions | Background Processes | Statistics | Error Log Excerpts | Log

Description

Configuration

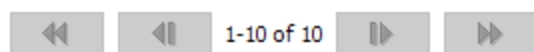
Current Background Process Log

- 1 Next poll not scheduled. (Tue Feb 07 10:34:35 EST 2017)
- 2 Next poll scheduled for Tue Feb 07 10:37:21 EST 2017 (Tue Feb 07 10:36:21 EST 2017)
- 3 Checked and found no new events. (Tue Feb 07 10:37:21 EST 2017)
- 4 Checked and found no new events. (Tue Feb 07 10:38:21 EST 2017)
- 5 Checked and found no new events. (Tue Feb 07 10:39:21 EST 2017)
- 6 Checked and found no new events. (Tue Feb 07 10:40:21 EST 2017)
- 7 Checked and found no new events. (Tue Feb 07 10:41:22 EST 2017)
- 8 Checked and found no new events. (Tue Feb 07 10:42:22 EST 2017)
- 9 Checked and found no new events. (Tue Feb 07 10:43:22 EST 2017)
- 10 Checked and found no new events. (Tue Feb 07 10:44:22 EST 2017)

Navigation buttons: << < 1-10 of 10 > >> Save Report Truncate

Navigating the Current Background Process Log

After the first entry, the Current Background Process Log will display the following navigation buttons:



- Selecting the  button will navigate back one page.
- Selecting the  button will navigate to the beginning of the log.
- Selecting the  button will navigate forward one page.
- Selecting the  button will navigate to the end of the log.
- The buttons are will not be functional when until the log collects over 100 lines.

Save Report

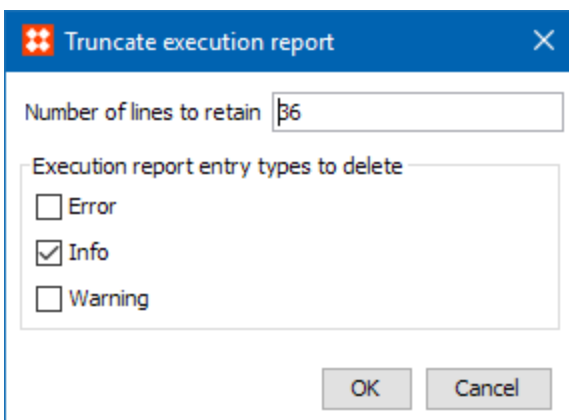
The Current Background Process Log can be saved as an 'HTML Only' file type.

To save an HTML Only version of the log, select the **Save Report** button, and the file explorer dialog for your system will display. Enter a name for the log report and click **Save**.

Truncate

The Current Background Process Log can be truncated. This can be useful when logs become cluttered or only specific information is needed.

To truncate a log, select the **Truncate** button, and the Truncate execution report dialog will display with the following options:



The image shows a dialog box titled "Truncate execution report" with a close button (X) in the top right corner. Inside the dialog, there is a text input field labeled "Number of lines to retain" with the value "36" entered. Below this is a section titled "Execution report entry types to delete" which contains three checkboxes: "Error" (unchecked), "Info" (checked), and "Warning" (unchecked). At the bottom of the dialog are two buttons: "OK" and "Cancel".

- Number of lines to retain
 - By default, the current number of log entries will display.
 - Either enter the number of lines to retain from the end of the log (if you wish to only save the last 'X' number of lines) or leave the current number and select from the 'Execution report entry types to delete.'
- Execution report entry types to delete
 - Up to three entry types can be deleted; Error, Info, and Warning.
 - By default 'Info' is selected.
 - Each entry type selected will be deleted from the log after selecting **OK**.

After making the necessary selections, click **OK** and the event processor editor will display with the Current Background Process Log truncated.

Event Processor Queue Status

By default, when a new event processor is created, events are discarded. Use the event processor editor to update the **Queue Status**.

1. Go to System Setup > your event processor > Select **Event Processor** tab.
2. Open the **Configuration** flipper.
3. In **Queue Status**, select **Read Events** if you are ready for the processor to apply the actions based on the events processed.
 - **Read Events** allows the processor to register events as they occur, based on the event processor configuration and event triggering definitions.
 - **Discard Events** ignores generated events that meet the configuration and event triggering definitions. This is useful when testing a processor without modifying data.

The screenshot shows the 'Matching Processor - Event Processor' configuration window. The 'Event Processor' tab is selected. The 'Configuration' section is expanded, showing a table of settings. The 'Queue Status' is currently set to 'Discard Events'.

ID	Name
User running event processor plugin	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
Processor	Matching
Schedule	Not scheduled ...
Queue Status	Discard Events
Unread events (approximated)	Click to estimate ...

[Edit Configuration](#)

Important: If you are ready for the processor to apply the actions based on the events processed, make sure the event processor Status parameter is also set to Enabled, otherwise the event processor will not function. For more information see the **Enable Event Processor** section of the **Running an Event Processor** documentation.

Event Triggering Definitions Tab

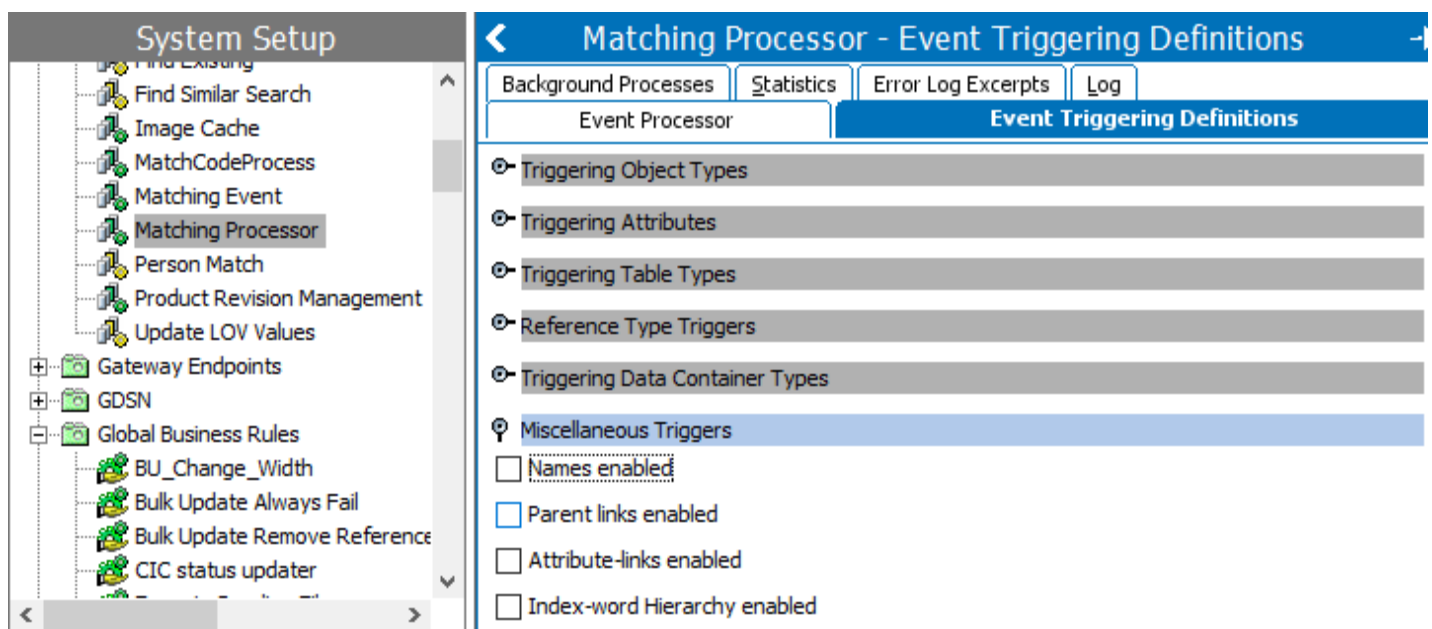
The Event Triggering Definitions Tab contains information about the configuration to be set to trigger events as per the requirement.

Prerequisites

The Event Triggering Definitions tab is part of the event processor editor. Information about the event processor editor can be found in the **Maintaining an Event Processor** topic in the **Event Processors** documentation.

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. Use the event processor editor 'Event Triggering Definitions' tab to configure trigger definitions.



Triggering object types are used to filter the data being monitored by the event processor, whereas processor plugins (standard core processors) provide direction on what to do next.

Note: The functionality of the Event Triggering Definitions tab is the same for event processors and integration endpoints. For details about the options, see the **OIEP - Event-Based - Event Triggering Definitions Tab** documentation.

To help streamline setup for event processors using processing plugins, the information for common parameters and triggering definition setups for each of the processing plugins is covered in the following sections:

- **CASS Certification Report Processing Plugin Parameters and Triggers**
- **Execute Business Action Processing Plugin Parameters and Triggers**
- **Expand LOV Value Changes Processing Plugin Parameters and Triggers**
- **Experian Email Validation Processing Plugin Parameters and Triggers**
- **Image Cache Processing Plugin Parameters and Triggers**
- **Matching Processing Plugin Parameters and Triggers**
- **Revision Management Processing Plugin Parameters and Triggers**

Important: If you are ready for the processor to apply the actions based on the events processed, make sure the Queue Status parameter is also set to Read Events, otherwise the enabled event processor will not function. For more information see the **Event Processor Queue Status** documentation.

Background Processes Tab

An active Event Processor uses an associated background process to handle the scheduled invocation. The Background Processes tab contains information about the Queued Process, Active Processes, Completed with Errors and Ended Processes. Hover over an ID to display the background process (BGP) link, click it to open the individual Importer Background Processes.

Matching Processor - Background Processes						
Event Processor Event Triggering Definitions Background Processes Statistics Error Log Excerpts Log						
🔍 Queued Processes (1)						
Id	Description	Status	Progress	Start Date	Started By	Errors
BGP_184017	Processor for Matching Pr...	waiting	0%	Tue Sep 13 11:49:33 EDT ...	USERJ	6
🔍 Active Processes						
Id	Description	Status	Progress	Start Date	Started By	Errors
🔍 Completed with Errors						
Id	Description	Status	Progress	Start Date	Started By	Errors
🔍 Ended Processes						
Id	Description	Status	Progress	Start Date	Started By	Errors

Prerequisite

The Background Processes tab is part of the event processor editor. Information about the event processor editor can be found in the **Maintaining an Event Processor** topic in the **Event Processors** documentation.

Event Processor Background Processes

An active event processor uses an associated background process to handle the scheduled invocation. The event processors Background Processes tab holds the following information related to the selected event processor:

- **ID:** Provides ID of the Background Process. Hovering over it displays the background process (BGP) link. Click on the link to open the individual Import Background Process.
- **Description:** Describes the background process as specified by the user.
- **Status:** Provides the status of the background process. The Status field can possess any one of these statuses at a given time: *Waiting*, *Failed*, *Suspended*, or *Succeeded*.
- **Progress:** Displays the percentage of the BGP complete on a progress bar.
- **Start Data:** Displays the timestamp of when the Event Processor was invoked to run the BGP.
- **Started By:** Shows the user name of who initiated the background process by invoking the Event Processor.

- **Errors:** Displays the number of errors that occurred in the BGP
- **Created:** Shows the timestamp of when the queue was created for the first time.

Background Processes in Different States

Below are a list of the possible states that a background process can be in, as well as explanations as to what that state means.

Queued Processes

- Background processes that are started in the 'waiting' state.
- The maximum number of background processes that can be started in the 'waiting' state is specified in the Event Processor configuration.

Active Processes

- Background processes that are currently being processed.
- By default the data that is processed has one active background process at a time, though it is possible to apply more background process queues to work in parallel. In this case the Event Processor can have more active background processes at a time.

Failed Processes

- Background processes that are no longer running.

For more information, see the **Handling Failed IIEP Background Processes** section of the **Inbound Integration Endpoints** documentation.

Completed with Errors Processes

- When a process has run, but there are errors that have been detected.

Ended Processes

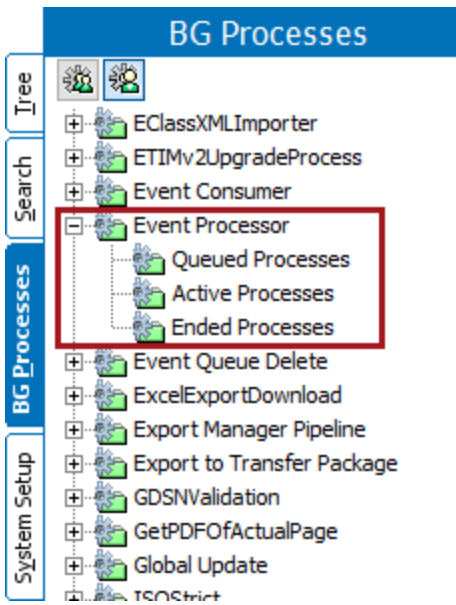
- The process has been completed successfully. The number of ended processes is determined by the configuration.

For more information, see the **Monitoring an IIEP via Background Process** topic in the **Data Exchange** documentation.

Note: The functionality of the Background Processes tab is the same for event processors and integration endpoints.

BG Processes Tab

Users may also navigate to the BG Processes tab > Event Processor and get all the information of all the background processes related to all the event processors in the system.



Statistics Tab

The Statistics tab provides an overview of the related background processes with the ability to click and see details about the number of requests handled, run times, and wait times. The information in the Statistics tab can be used to monitor the Event Processor.

Prerequisite

The Statistics tab is part of the event processor editor. Information about the event processor editor can be found in the **Maintaining an Event Processor** topic in the **Event Processors** documentation.

Statistics Flipper

Statistic data includes Name and Value field for:

- Last run
- Next run
- Started
- Total number of handled events
- Total processing time (seconds)
- Current number of handled events
- Current processing time (seconds)
- Number of unhandled events
- Estimated remaining time (seconds)

System Setup

Event Processors

BestEP

Email User When Attribute Value

External Item Async Matching

Find Existing

Find Similar Search

Image Cache

Loqate and CASS

LOV Value Merge Events

MatchCodeProcess

Matching Event

Matching Processor

Person Match

Product Revision Management

Update LOV Values

MatchCodeProcess - Statistics

Event Processor

Event Triggering Definitions

Background Processes

Statistics

Error Log Excerpts

Log

Statistics

Name	Value
> Last run	Wed Sep 14 11:47:44 EDT 2016
> Next run	Mon Jan 12 00:31:00 EST 9998
> Started	Wed Sep 14 11:46:17 EDT 2016
> Total number of handled events	1
> Total processing time (seconds)	0
> Current number of handled events	1
> Current processing time (seconds)	0
> Number of unhandled events	651
> Estimated remaining time (seconds)	393

Error Log Excerpts Tab

The Error Log Excerpts tab shows data from the main Java log file related to failed background processes with Log Level > 'Info'.

Prerequisites

The Event Processor tab is part of the event processor editor. Information about the event processor editor can be found in the **Maintaining an Event Processor** topic in the **Event Processors** documentation.

Event Processor Log Flipper

In this flipper, a user can click on a hyperlink to a failed background process and correct the cause of the failed background process.

Matching Processor - Error Log Excerpts		
Event Processor	Event Triggering Definitions	Background Processes
Statistics	Error Log Excerpts	Log
Event Processor Log		
Background Pro...	Log Item No	Text
> BGP_184017	120	Caught ClassCastException at Mon Aug 29 13:41:08 EDT 2016: com.stiibo.core.domain.impl.FrontProductImpl\$\$Generated\$\$70 cannot be cast to com.stiibo.matching.domain.rankscorable.RankScorable

- **Background Process:** Displays the ID of a Background process throwing the error. Clicking on the ID redirects the user to the Background process.
- **Log Item No:** Displays the Log Item No
- **Text:** Displays a brief text specifying an error message. For example: Caught IndexOutOfBoundsException at Tue Mar 14 10:44:08 CET 2017: Index: 0, Size: 0 "Failed to add 1 files to delivery due to Failed to create directory: Failed to create directory: "

Note: The functionality of the Error Log Excerpts tab is the same for event processors and integration endpoints. For more information, see **Handling Failed OIEP Background Processes** topic in the **Data Exchange** documentation.

Log Tab

The Log tab provides information about changes to the Event Processor configuration.

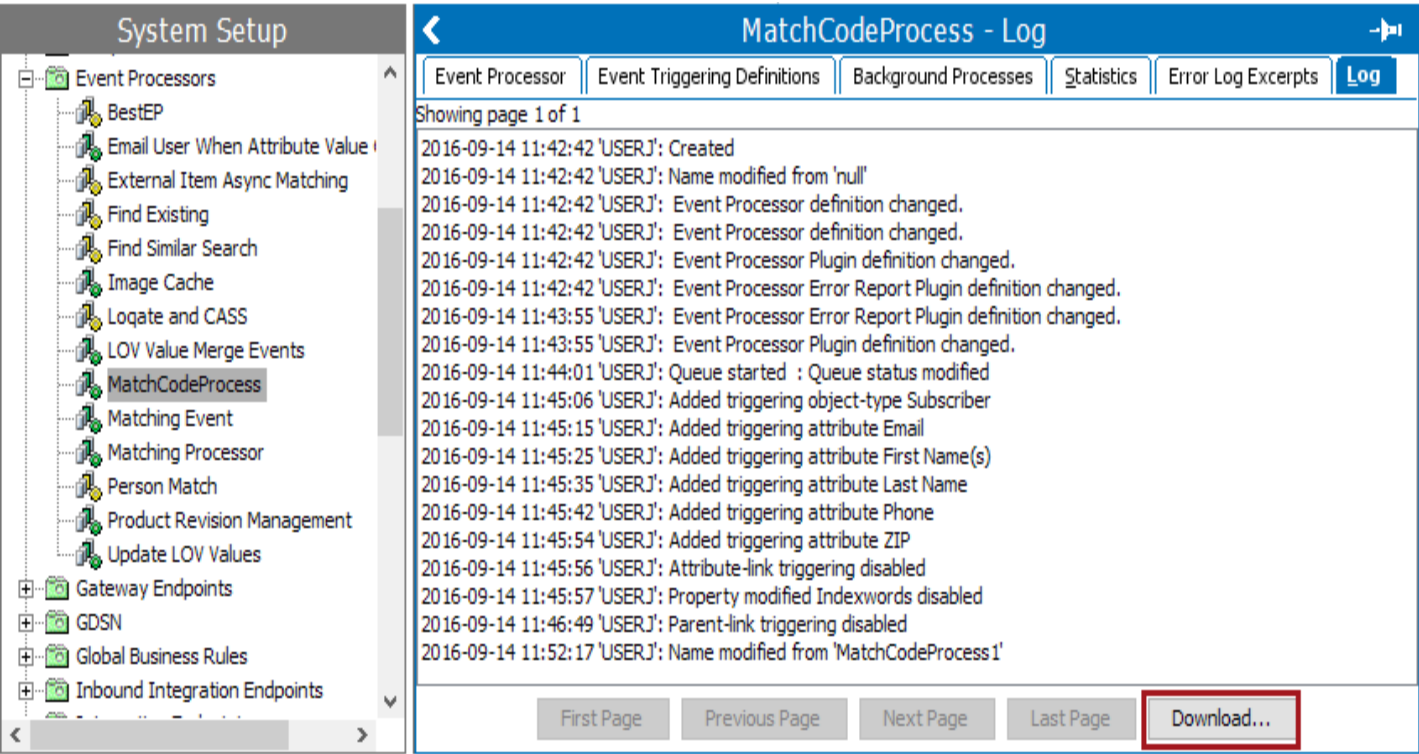
Prerequisite

The Log tab is part of the event processor editor. Information about the event processor editor can be found in the **Maintaining an Event Processor** topic in the **Event Processors** documentation.

Logs

Information included on the Log tab are timestamps, ID of the user who performed the action, and a brief description of the changes done on the event processor.

Should a user wish to download the information kept in the log, they can do so by clicking the Download button.



Monitoring an Event Processor via External System

Event processors can be monitored from external systems using Monitoring Sensors.

For each event processor created in STEP, a Monitoring Sensor is automatically created. Monitoring Sensors allow external systems to query the status of individual event processors via HTTP, without authentication. In the following sample URLs, substitute your own system URL and event processor ID to access your own event processor.

A Monitoring Sensor can return the following information:

1. **Simple traffic light response** returns OK, WARNING, CRITICAL, or UNKNOWN for event processors via:

```
http://[System URL]/admin/monitoring/EventProcessorStatus-[event processor ID]/status
```

2. **Nagios friendly response** returns detailed statistics in Nagios friendly format for event processors via:

```
http://[System URL]/admin/monitoring/EventProcessorStatus-[event processor ID]/nagios
```

3. **Full detailed XML response** returns XML with detailed statistics for event processors via:

```
http://[System URL]/admin/monitoring/EventProcessorStatus-[event processor ID]/xml
```

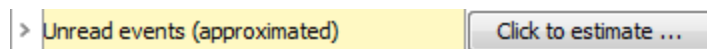
Event-Based Event Processor Queued Events

As events are found that meet the criteria set in the event processor configuration and event triggering definitions, and estimate and details are available for review in System Setup.

Estimate Unread Events

An estimate of the current number of unread events in the Event Queue Configuration area 'Unread events (approximated)' parameter.

On the OIEP's Configuration tab, Click the 'Click to estimate' button to display the data.



The 'Click to estimate' button is replaced with the number of events and the time of the estimate.

The screenshot shows the 'System Setup' window with the 'Event Processor' tab selected. The left sidebar lists various event processors, with 'Email User When Attribute Value Changes' highlighted. The main panel displays the configuration for the selected event processor. The 'Unread events (approximated)' field is highlighted with a red box, showing the value '4 (2017-07-18 17:16:28)'.

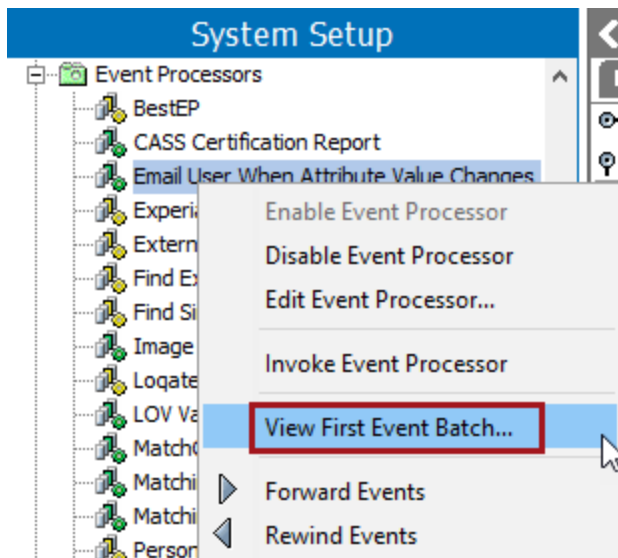
ID	Name
User running event processor plugin	USER NAME
Number of events to batch	1
Days to retain events	0
Queue for event processor	EVPROC
Maximum number of old processes	100
Maximum age of old processes in hours	168
Limit of lines in execution report	1000
Processor	Business Action Event Processor
Schedule	Start every minute
Queue Status	Read Events
Unread events (approximated)	4 (2017-07-18 17:16:28)

Using the forward, rewind, purge, or republish events options causes the number of unread events on the queue to change. For more information, see **Event-Based OIEP Forward, Rewind, Purge, and Republish**.

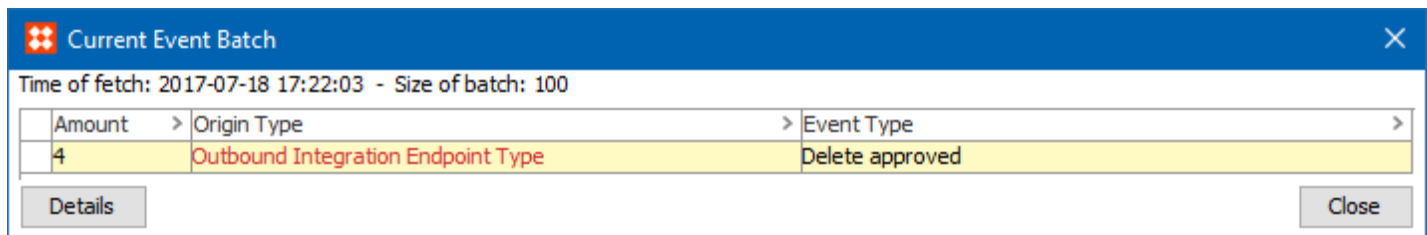
View First Event Batch

This option allows you to view more than just a count of the events.

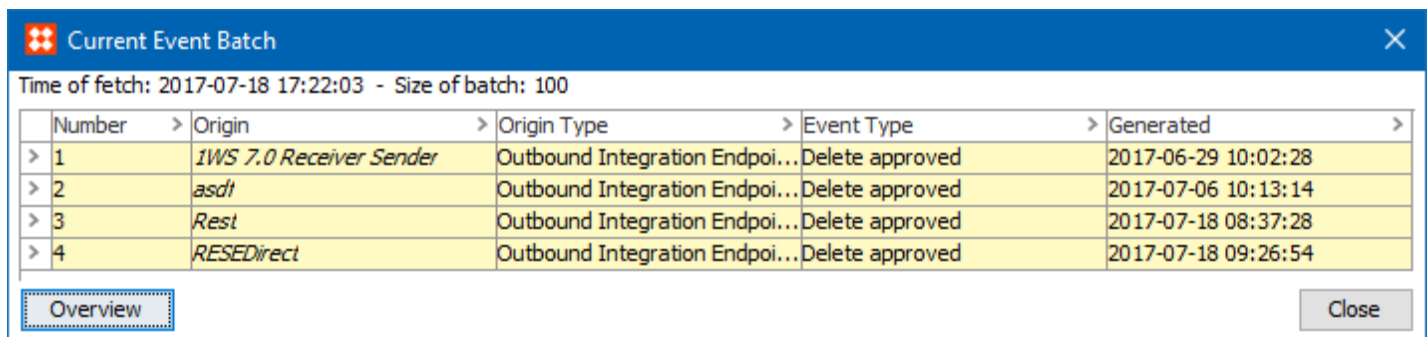
1. Right-click the event processor and select **View First Event Batch**.



2. The Current Event Batch dialog displays the first 100 unread events on the queue. Derived events are represented by a number instead of by their ID.



3. For more information about the events, click the **Details** button. The Origin column includes a link to the object being reported by the event.



4. Click the **Overview** button to return to the previous view.

List of Values (LOVs)

A List of Values (LOVs) 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.

- Attributes of this type will reference a separate 'List of Values' or 'LOV' object that holds a list of possible values to choose from.
- LOVs come in two variants; one that allows users to add new values when populating attributes that use the LOV (a medium domain LOV), and one that does not allow this (a hard domain LOV).
- LOVs are also sometimes referred to as 'Domains.'
- The values that exist on an LOV are available for selection on an attribute via a dropdown, as shown below.

The screenshot displays the Stibo Systems interface. On the left, the 'System Setup' sidebar shows a tree structure with 'Lists of Values / LOVs' expanded, listing various LOVs including 'Battery Type'. The main window is titled 'Battery Type - List of Values' and shows a table of values:

Values	Value ID
Alkaline	Alkaline
Lithium-ion	LithiumIon
NiMH	NickelMetalHydride
Plutonium	Plutonium

Below this, a 'Tree' view shows the product hierarchy: Electrical and Electronics > Electronic Accessories > Power > Batteries > Batteries Rechargeable Item > Recharge AA (123943). The right pane shows the 'Recharge AA (123943) rev.0.19 - P' product configuration. The 'Battery Type' attribute is highlighted, and its dropdown menu is open, showing the same list of values: Alkaline (Alkaline), Lithium-ion (LithiumIon), NiMH (NickelMetalHydride), and Plutonium (Plutonium).

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

LOV Features

- LOV objects have a Validation Base Type setting of their own, and values must comply to this.
- A single LOV can be used by multiple Attributes.
- For each Attribute, a filter can be applied so only a subset of the values is selectable for the Attribute.
- If the Attribute is a Specification Attribute it is furthermore possible to filter legal values on the links from the instantiated hierarchy so only specific values are legal for specific branches.

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.

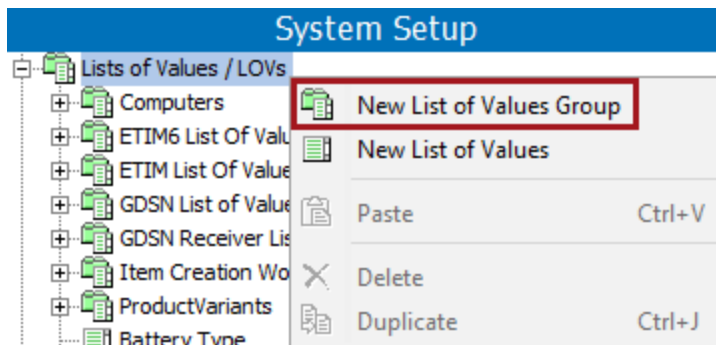
To see all LOV values, type an asterisk (*) to initially display most of the values, and after fully loading, display the complete, scrollable list of values.

Creating an LOVs Group

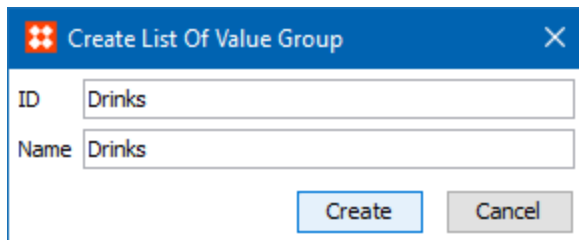
A Lists of Values / LOVs Group can be used to collect individual, but related, LOVs. The LOVs group can be linked to an attribute group, and used in combination with privileges to restrict users in maintaining certain List of Values.

Below are steps to create a new Lists of Values / LOVs Group.

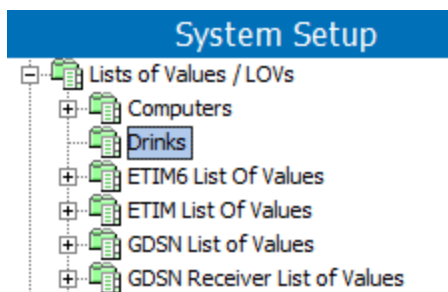
1. Go to System Setup > Lists of Values / LOVs, and right-click **List of Values /LOVs** and click the **New List of Values Group** option.



2. Enter an **ID** and **Name**, then click the **Create** button.



3. The newly created LOV group will display within the List of Values / LOVs, as shown below.



4. To add an LOV to the LOV group, follow the steps in the **Creating a List of Values** topic within this guide.
5. To link an LOVs group to an attribute group, click the **Add Attribute Group** link within the 'In Attribute Groups' flipper of the List of Values Group editor, as shown below.

The screenshot displays the 'System Setup' interface for 'Drinks - List of Values Group'. On the left, a tree view shows the hierarchy: Lists of Values / LOVs > Boards List Of Values > CDM > Computers > Customer MDM LOVs > D&B LOVs > Drinks. The main panel shows the 'List of Values Group' editor. The 'In Attribute Groups' flipper is open, showing a table with columns 'ID' and 'Name'. The 'Add Attribute Group' link is highlighted with a red box. A red arrow points from this link to a 'Select Attribute Group' dialog box. The dialog box has a 'Browse' tab and a list of attribute groups: Display, Email Attribute Group, Entity Data Container Types, Find Similar, Flavors (selected), GDSN Attributes, and GDSN Product System Attributes. The 'Select' button is highlighted with a red box.

Name	Value
ID	Drinks
Name	Drinks
Last edited	2015-08-11 13:51:15 by USER
Path	Lists of Values / LOVs/Drinks

ID	Name
>	Add Attribute Group

Select Attribute Group

Browse Search

- Display
- Email Attribute Group
- Entity Data Container Types
- Find Similar
- Flavors**
- GDSN Attributes
- GDSN Product System Attributes

Details

Select Cancel

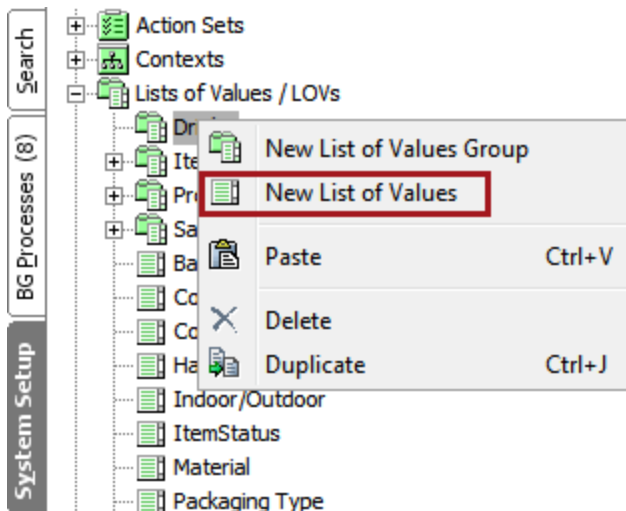
Creating an LOV

Creating an LOV (List of Values) involves:

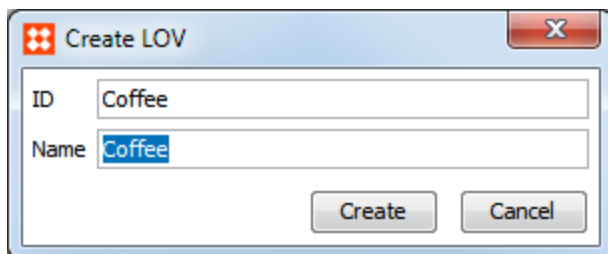
- Setting validation rules
- Selecting valid units
- Selecting dimension dependencies
- Adding values

Below are the steps to create an LOV.

1. Go to System Setup > Lists of Values / LOVs, right-click the List of Values / LOVs folder (or a desired LOVs group folder where you want to group the new LOV), and click **New List of Values**.



2. The Create LOV dialog will display. Enter an **ID** and a **Name**.



3. Click the **Create** button, and the List of Values editor will display (as shown below).

<
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](#)

- Expand the **List of Values Validation** flipper, and click the **Validation Base Type** parameter field to select the desired validation base type from the dropdown menu.

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
	Regular Expression

Values

Values	>

5. Within the **List of Values Validation** flipper, click the **Allow Users to Add Values** parameter field to select either **Yes** or **No** from the dropdown menu (as shown below).

☐ 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:** allows the user to add values to the existing list from the object editor.
 - **No:** does not allow the user to add values, thus requiring them to select a value that already exists in the list. New values can only be added from System Setup.
6. **Use IDs on values:** Optionally allows the use on an ID for each value within the LOV. Can be useful when importing object data and applying a certain value with a specific ID.
 - 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 auto generate a unique id for each value created in the LOV. The pattern [id] can be combined with a prefix or suffix if needed.
 - Setting Use IDs on Values to Yes also enables the Use IDs on Sorting field. Yes means that the value are sorted based on the IDs. No means that the value are sorted based on the Values, not based on the IDs
 - If Allow Users to add Values is set to No, set Use IDs on Values to Yes.

<
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. For more information, see the **Adding IDs to Values in LOV** topic within this section.

7. **Units:** Optionally allows for the use of units. The Unit field is only shown in the 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. **Dimension Dependencies:** Optionally allows for the use of more than one dimension. To enable, 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**.

System Setup

- Dimension
- Dimension Point
- eCatalog user-type root
- Event Processor
- Experian Configuration Setu
- Gateway Integration Endpc
- Inbound Integration Endpo
- InDesign Document
- Key
- List Of Values Group Type
 - Domains**
 - List Of Values Group Ty
- Match Code

Domains - Object Type

Object Type | References | Log

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;

- Add all required values to the LOV by following the steps in the **Adding Values to an LOV** topic within this guide.
- 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. This topic addresses adding a single LOV value, and adding a group of LOV values.

Adding a Single LOV Value

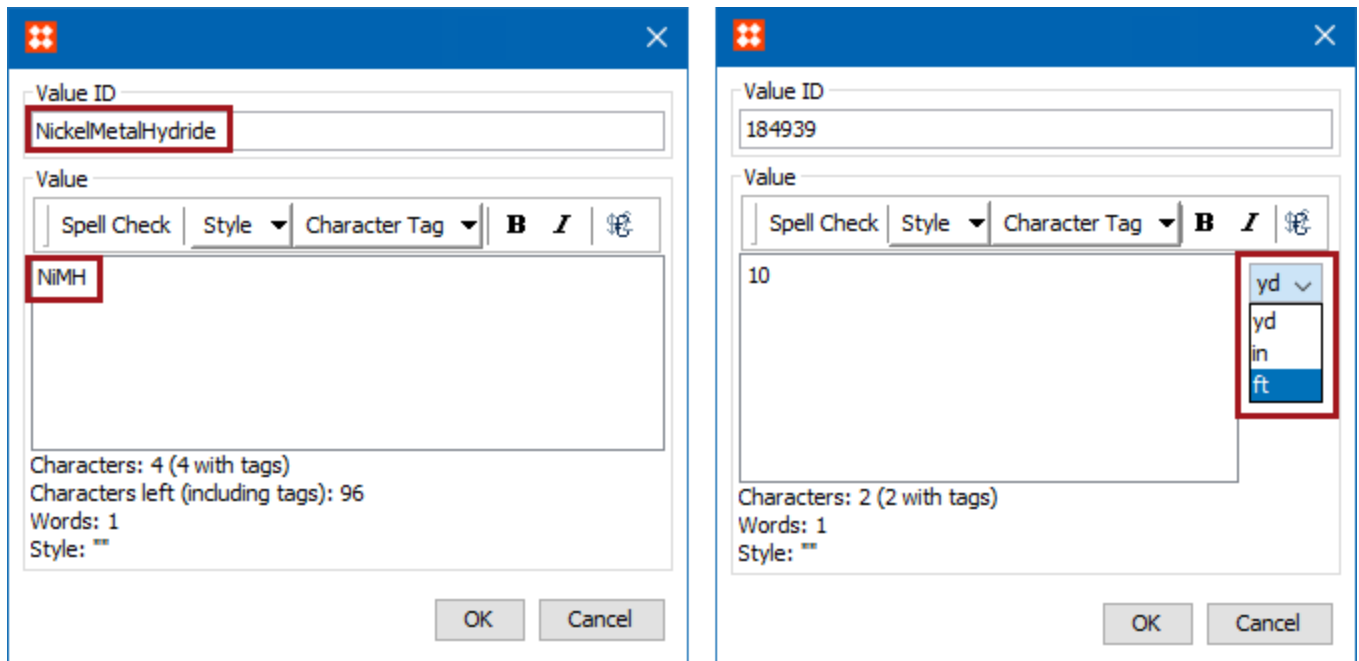
When only a few new values are required, it is simple to add them individually from the System Setup tab.

1. In **System Setup**, select the relevant LOV, open the Values flipper and click the **Add Value** link.

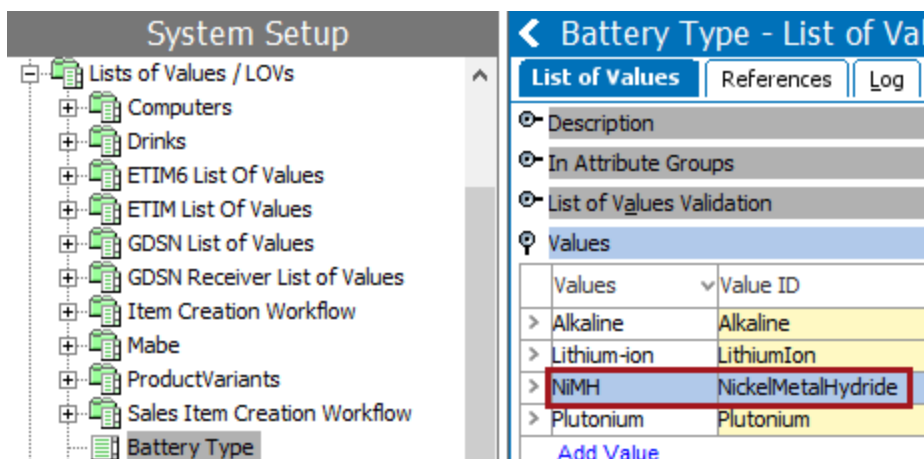
The screenshot shows the 'System Setup' tab on the left with a tree view of various LOVs. The 'Battery Type' LOV is selected. On the right, the 'Battery Type - List of Values' flipper is open, showing the 'List of Values' tab. The table lists existing values: Alkaline, Lithium-ion, and Plutonium. The 'Add Value' link is highlighted in a red box at the bottom of the table.

Values	Value ID
Alkaline	Alkaline
Lithium-ion	LithiumIon
Plutonium	Plutonium

2. Enter a **Value ID** (if required by the LOV's use ValueID parameter) and a **Value**.
 - If units are not defined for the LOV, then simply enter a value (as shown in the screenshot below on the left).
 - If units are defined for the LOV, select the appropriate unit for the value from the available dropdown (as shown in the screenshot below on the right).



3. Once all the proper selections are made, click the **OK** button to add it to the LOV.
4. Verify the new value is displayed for the LOV.

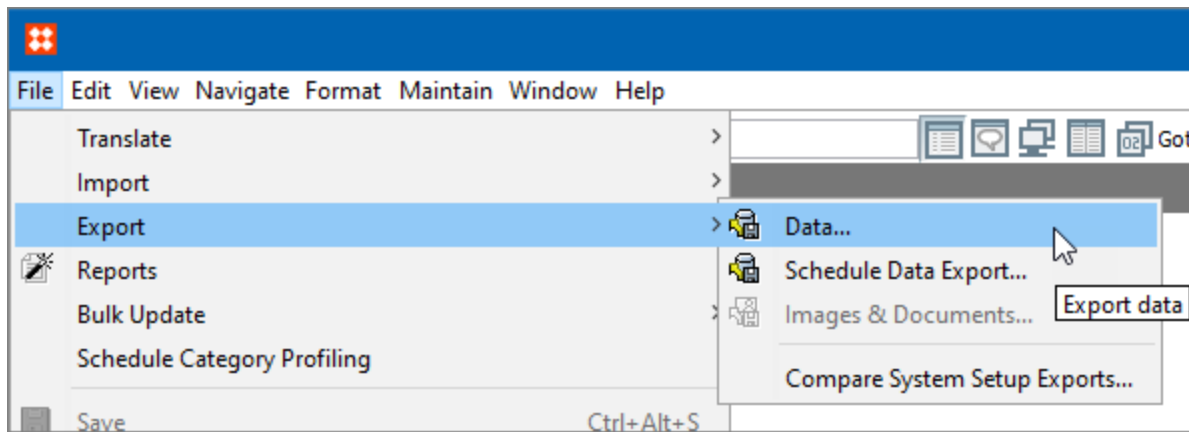


Adding a Group of LOV Values

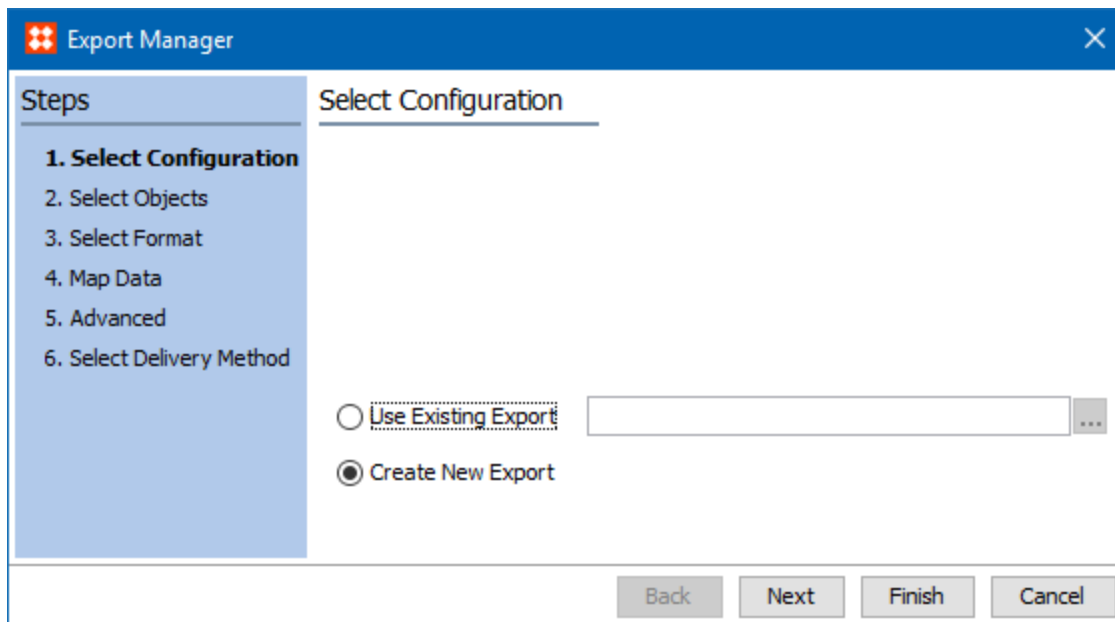
When many new values are required, it may be easier to import the values using STEPXML. This process involves exporting the existing LOV to create a STEPXML file, modifying the file to include the new values, and then importing those new values back into STEP.

Exporting the Existing LOV to Create a STEPXML File

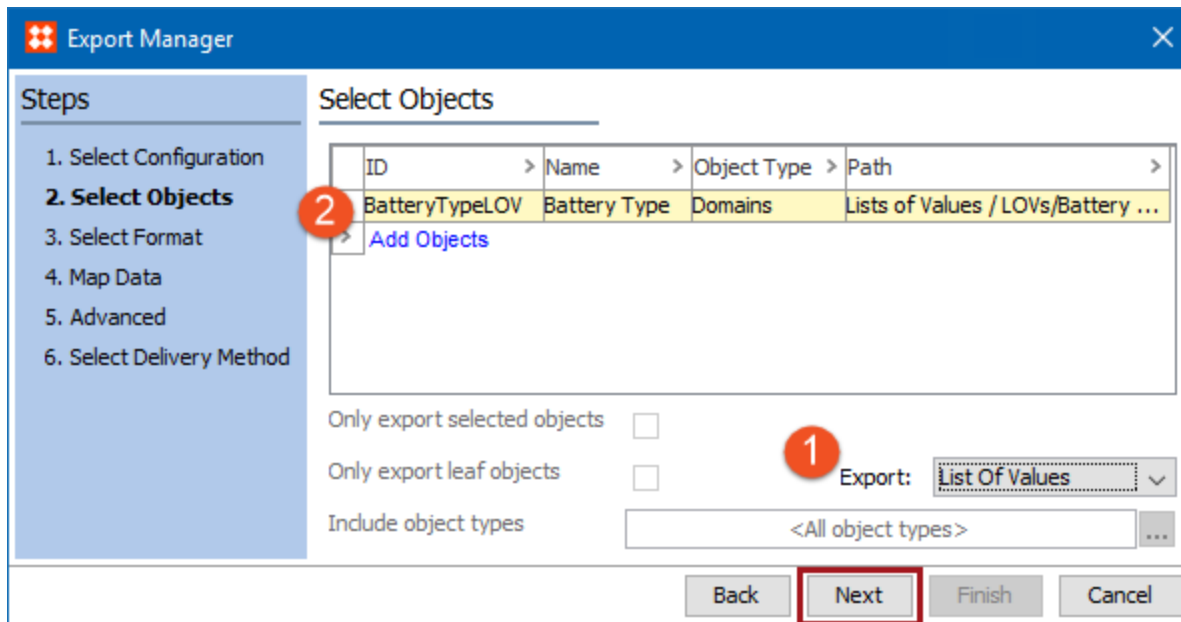
1. Go to Workbench > File > Export > Data



2. The Export Manager dialog will display, as shown below.



3. Click the **Next** button and step 2 Select Objects will display.



The 'Export Manager' dialog box is shown in the 'Select Objects' step. The 'Steps' list on the left includes: 1. Select Configuration, 2. Select Objects (highlighted), 3. Select Format, 4. Map Data, 5. Advanced, and 6. Select Delivery Method. The main area displays a table with columns: ID, Name, Object Type, and Path. The table contains one row: BatteryTypeLOV, Battery Type, Domains, Lists of Values / LOVs/Battery ... A red circle with the number '2' is next to the 'Add Objects' link below the table. Below the table, there are three checkboxes: 'Only export selected objects' (unchecked), 'Only export leaf objects' (unchecked), and 'Include object types' (unchecked). To the right of these checkboxes is an 'Export:' dropdown menu set to 'List Of Values', with a red circle and the number '1' next to it. Below the dropdown is a text field containing '<All object types>'. At the bottom, there are four buttons: 'Back', 'Next' (highlighted with a red box), 'Finish', and 'Cancel'.

ID	Name	Object Type	Path
BatteryTypeLOV	Battery Type	Domains	Lists of Values / LOVs/Battery ...

Only export selected objects ☐

Only export leaf objects ☐

Include object types ☐

Export: List Of Values

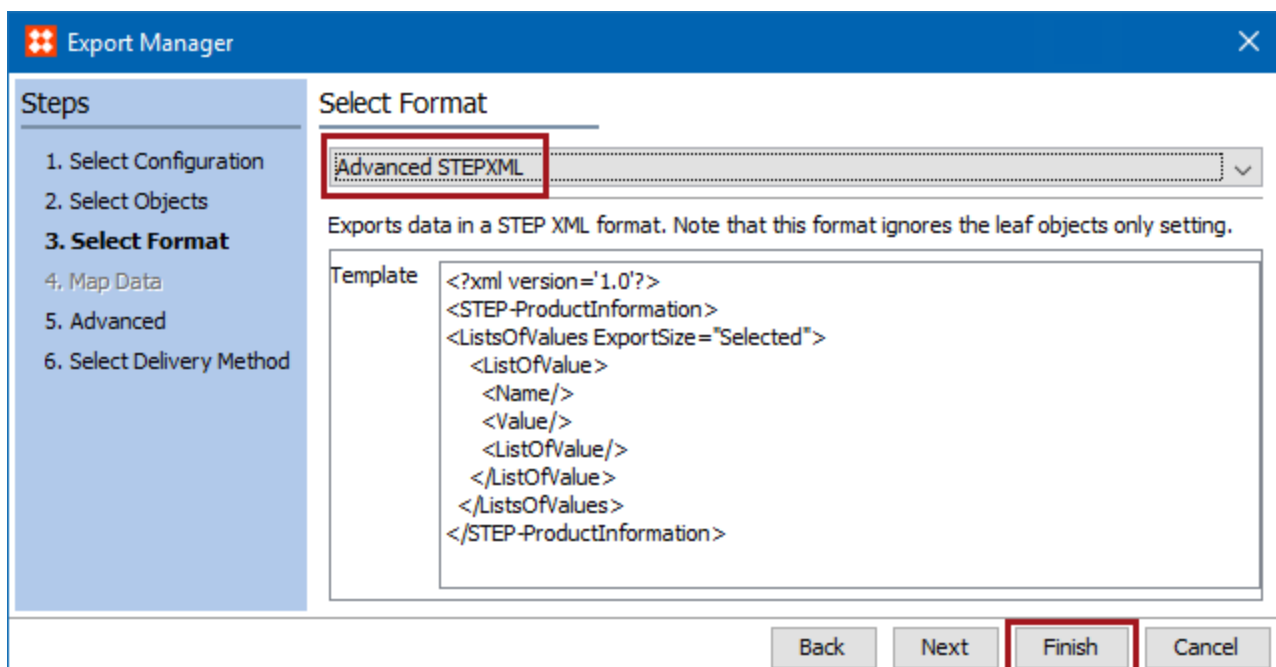
<All object types>

Back Next Finish Cancel

- 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.



The 'Export Manager' dialog box is shown in the 'Select Format' step. The 'Steps' list on the left includes: 1. Select Configuration, 2. Select Objects, 3. Select Format (highlighted), 4. Map Data, 5. Advanced, and 6. Select Delivery Method. The main area displays a dropdown menu for 'Select Format' with 'Advanced STEPXML' selected, highlighted with a red box. Below the dropdown, there is a note: 'Exports data in a STEP XML format. Note that this format ignores the leaf objects only setting.' Below the note is a 'Template' text area containing the following XML code:


```
<?xml version='1.0'?>
<STEP-ProductInformation>
<ListsOfValues ExportSize="Selected">
  <ListOfValue>
    <Name/>
    <Value/>
  </ListOfValue>
</ListOfValue>
</ListsOfValues>
</STEP-ProductInformation>
```

 At the bottom, there are four buttons: 'Back', 'Next', 'Finish' (highlighted with a red box), and 'Cancel'.

Advanced STEPXML

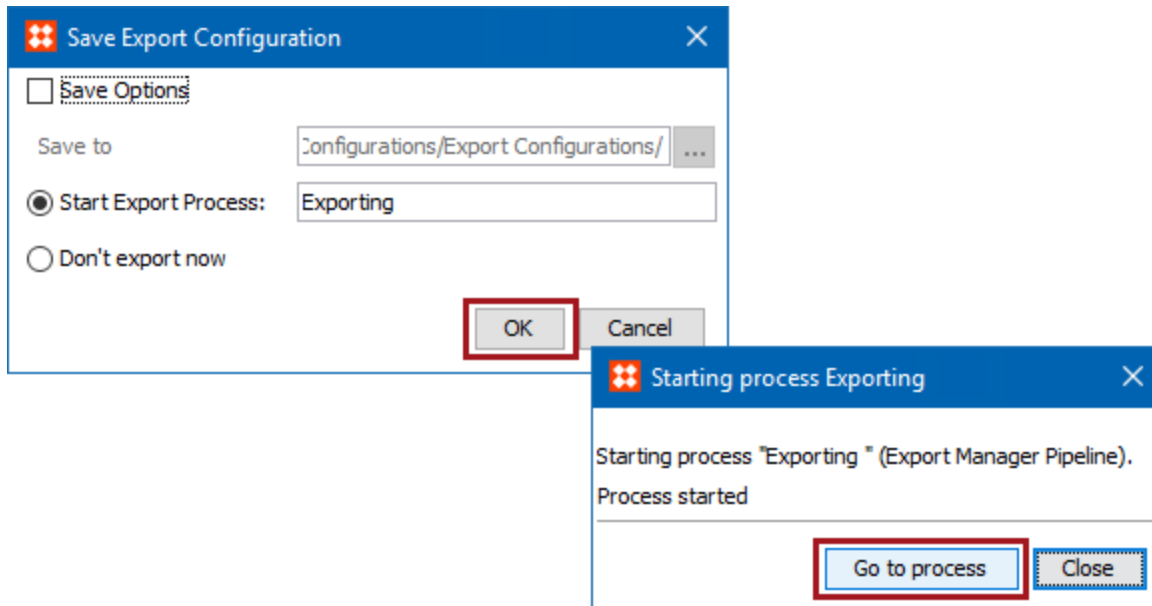
Exports data in a STEP XML format. Note that this format ignores the leaf objects only setting.

Template

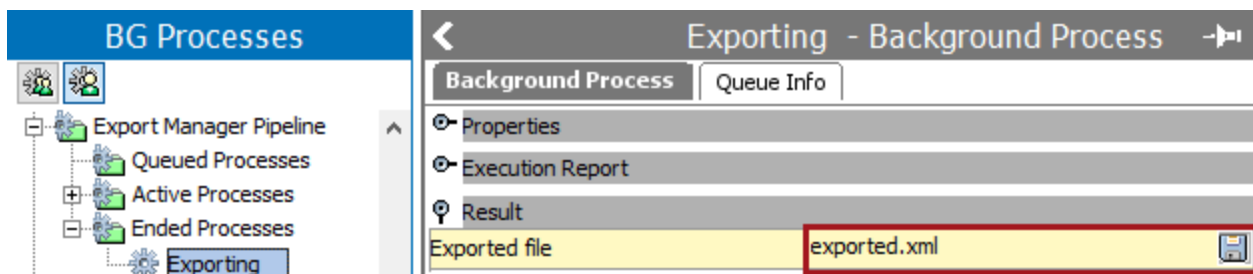
```
<?xml version='1.0'?>
<STEP-ProductInformation>
<ListsOfValues ExportSize="Selected">
  <ListOfValue>
    <Name/>
    <Value/>
  </ListOfValue>
</ListOfValue>
</ListsOfValues>
</STEP-ProductInformation>
```

Back Next Finish Cancel

- On the Save Export Configuration dialog, click **OK** to start the export, and on the Starting process dialog, click the **Go to process** button.



- On the displayed background process editor, open the output file.



Modifying the File to Include the New Values.

- In the output file, notice the **UseValueID** setting and the **Value** tags.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Configuration:
<?xml version='1.0'?>
<STEP-ProductInformation ResolveInlineRefs="true">
<ListsOfValues ExportSize="Selected">
  <ListOfValue>
    <Name/>
    <Value/>
    <ListOfValue/>
  </ListOfValue>
</ListsOfValues>
</STEP-ProductInformation>
Export from Primary Product Hierarchy
Classifications All
Products All
Assets All
-->
<STEP-ProductInformation ExportTime="2016-09-08 16:27:37" ExportContext="Context1" ContextID="Context1" WorkspaceID="Main" UseContextLocale="false">

  <ListsOfValues>
    <ListOfValue ID="BatteryTypeLOV" ParentID="List Of Values group root" AllowUserValueAddition="false" UseValueID="true" Selected="true" Referenced="true">
      <Name>Battery Type</Name>
      <Validation BaseType="text" MinValue="" MaxValue="" MaxLength="100" InputMask=""/>
      <Value ID="Plutonium">Plutonium</Value>
      <Value ID="LithiumIon">Lithium-ion</Value>
      <Value ID="NickelMetalHydride">NiMH</Value>
      <Value ID="Alkaline">Alkaline</Value>
    </ListOfValue>
  </ListsOfValues>
</STEP-ProductInformation>
```

- Additionally, if **Units** are being used by the LOV, notice the valid options.

```
<STEP-ProductInformation ExportTime="2016-09-12 16:47:34" ExportContext="Context1" ContextID="Context1" WorkspaceID="Main" UseContextLocale="false">

  <ListsOfValues>
    <ListOfValue ID="Variants" ParentID="List Of Values group root" AllowUserValueAddition="false" UseValueID="true" IDPattern="[id]" Selected="true" Referenced="true">
      <Name>Variants</Name>
      <Validation BaseType="number" MinValue="" MaxValue="" MaxLength="" InputMask=""/>
      <UnitLink UnitID="unece.unit.INH"/>
      <UnitLink UnitID="unece.unit.YRD"/>
      <UnitLink UnitID="unece.unit.FOT"/>
    </Validation>
    <Value ID="184941" UnitID="unece.unit.FOT">15</Value>
    <Value ID="184940" UnitID="unece.unit.YRD">10</Value>
    <Value ID="184942" UnitID="unece.unit.INH">27</Value>
  </ListOfValue>
</ListsOfValues>
</STEP-ProductInformation>
```

2. Copy the output text file and save it with a **new name** to create an **import** file.
3. 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.

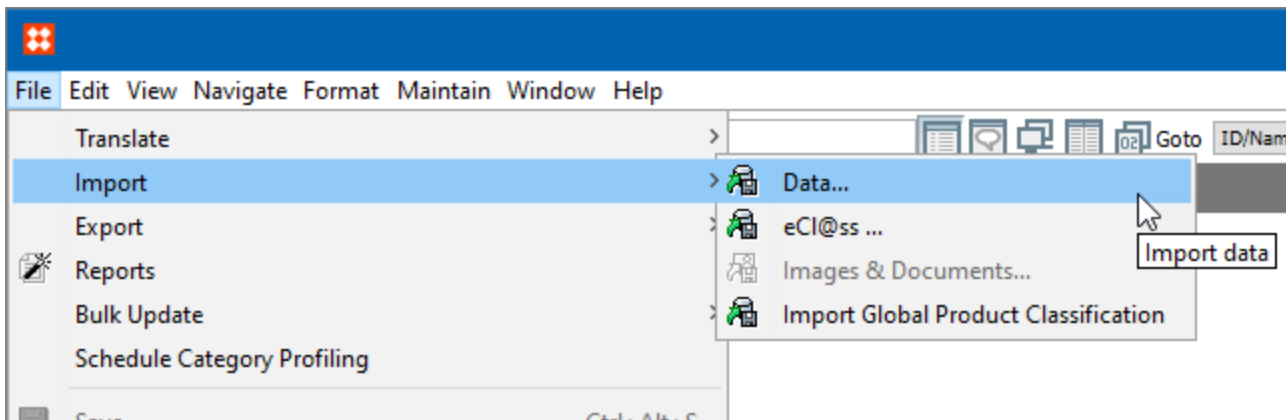
```

<STEP-ProductInformation ExportTime="2016-09-08 16:27:37" Export
  <ListOfValues>
    <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>
  </ListOfValues>
</STEP-ProductInformation>

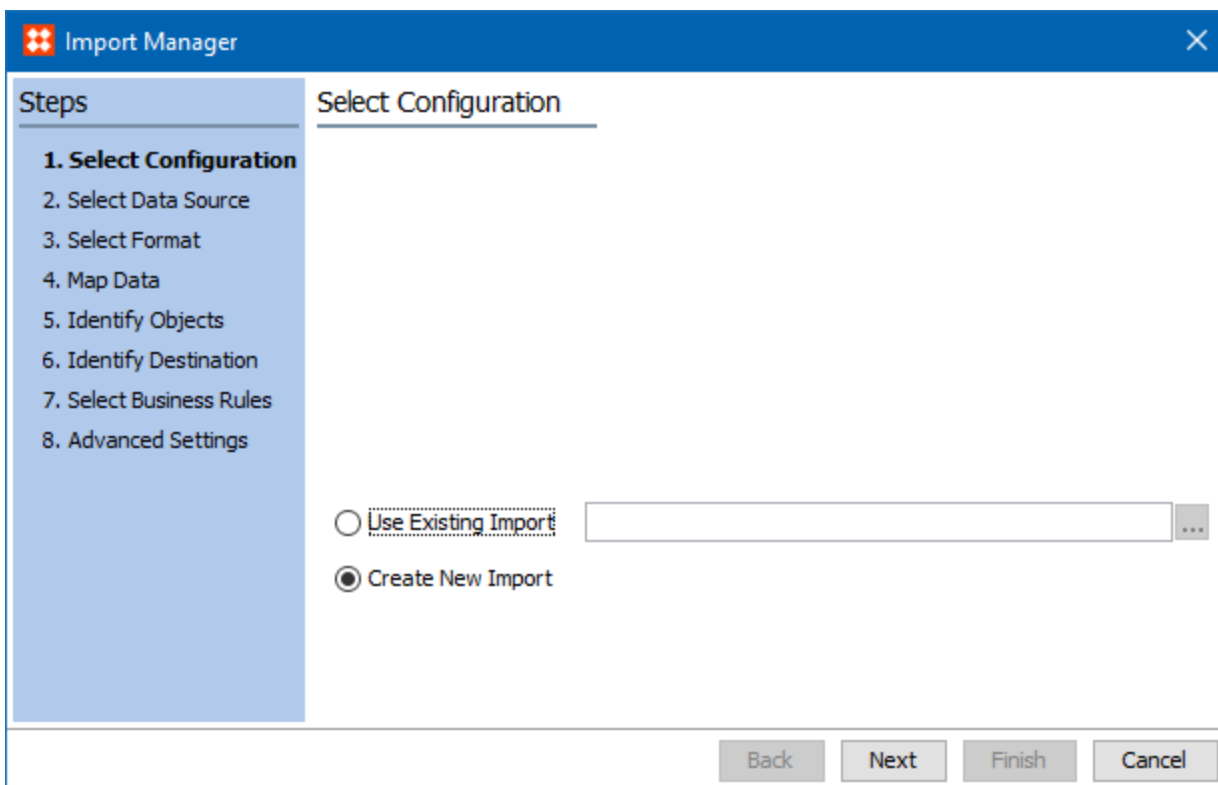
```

Importing New Values Into STEP

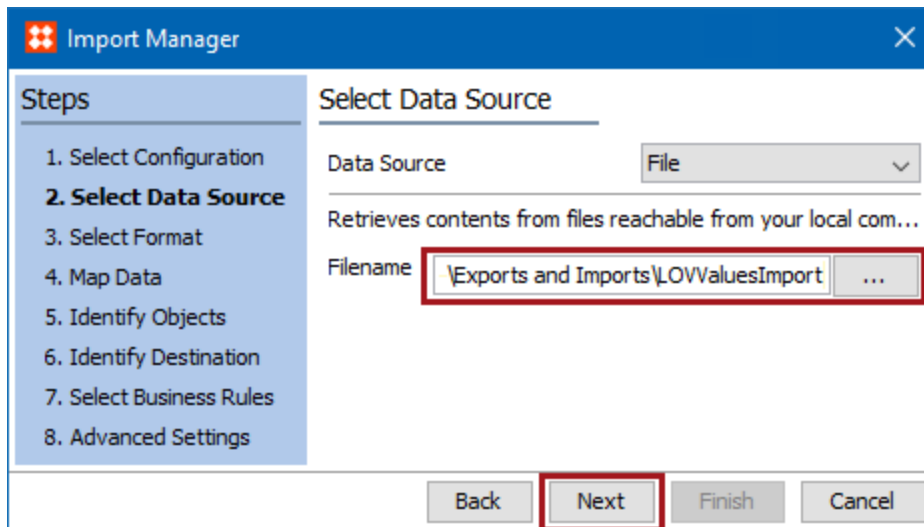
1. Go to Workbench > File > Export > Data



2. The Import Manager dialog will display, as shown below.



3. Click the **Next** button and step 2 Select Data Source will display.

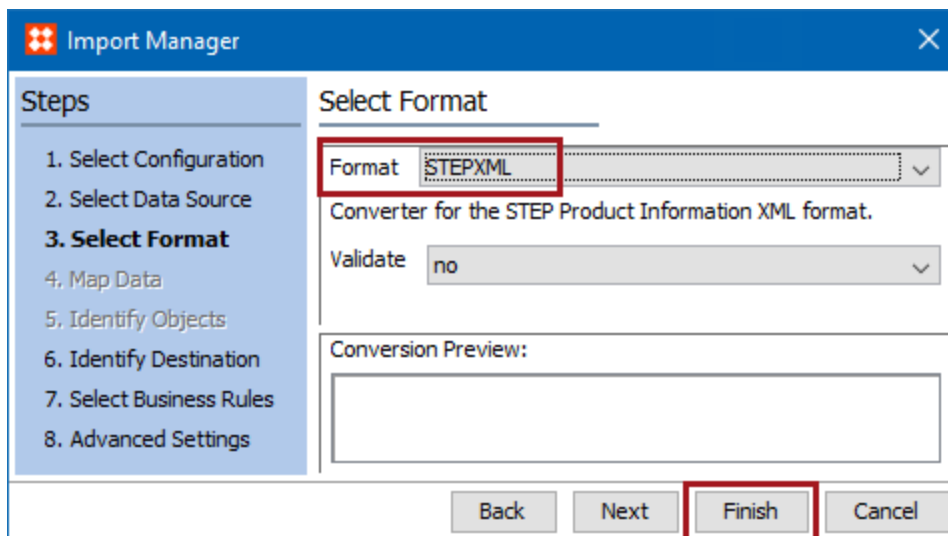


4. On step 2 Select Data Source:

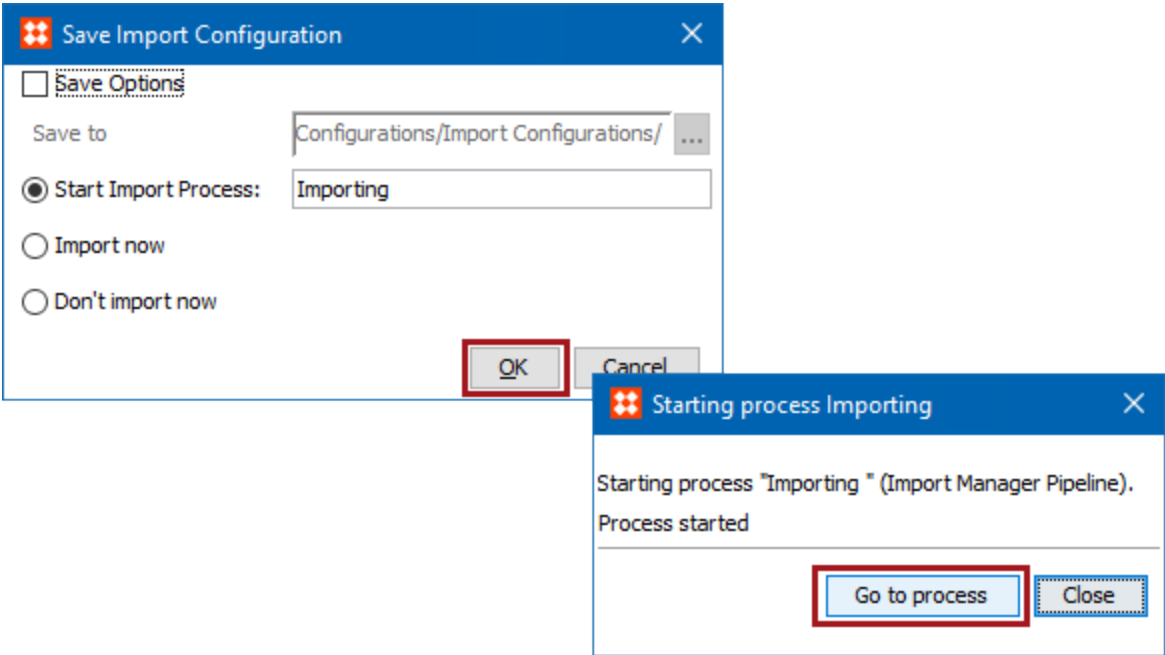
- In the Filename field, select the **import file name**.
- Click the **Next** button.

For details about each step and parameter of the Import Manager wizard, see the **Creating a Data Import** section of the **Data Exchange** documentation.

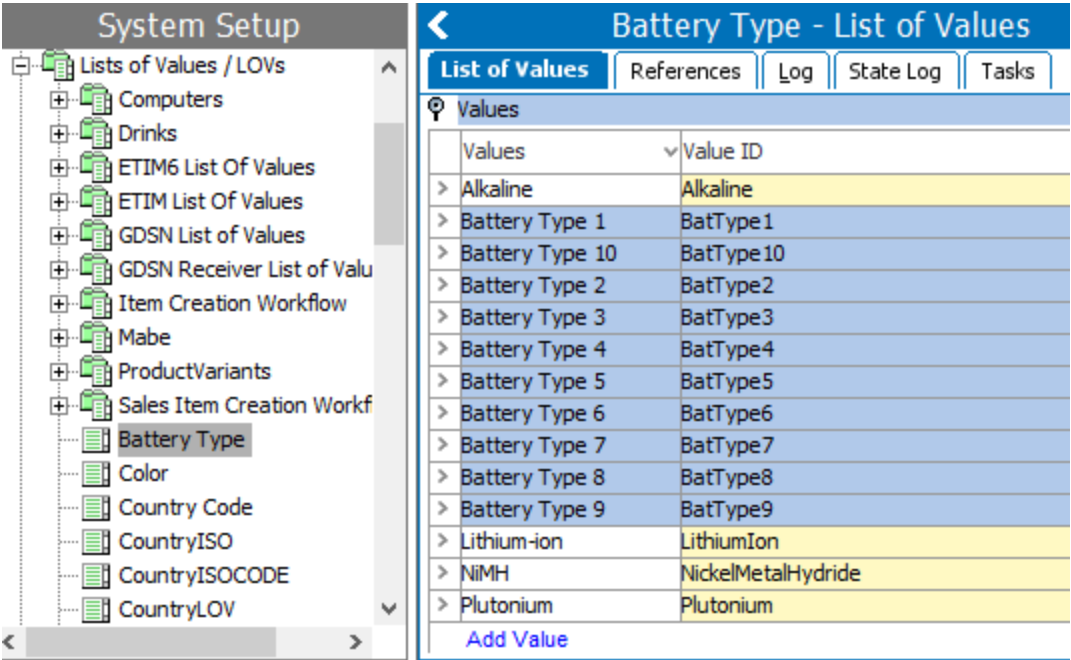
5. On step 3 Select Format, select **STEPXML** and click the **Finish** button.



6. On the Save Import Configuration dialog, click the **OK** button to start the import, and on the Starting process dialog, click **Go to process**.



- 7. On the displayed background process editor, verify the import succeeded.
- 8. In System Setup, verify the new values are displayed for the LOV.

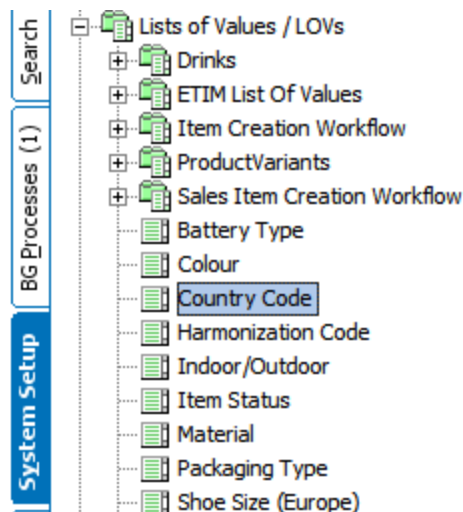


Adding IDs to Values in LOV

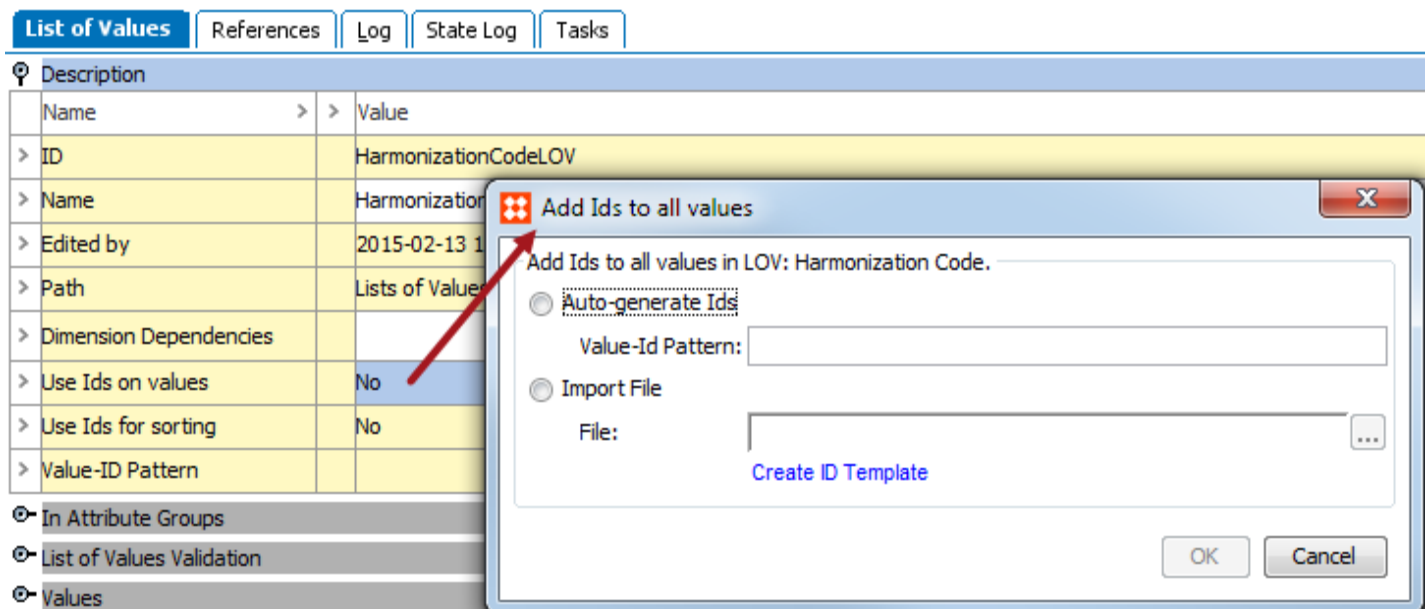
For LOVs that already contain values, Value ID generation adds an ID to each value in the LOV. For example, it is possible to map the LOV values to the international standard codes as 'value ID' (i.e., color code, abbreviations of country or cities, ISO codes).

Below are the steps to add IDs to values in LOV using the auto-generate IDs or the import file options.

1. Go to System Setup > Lists of Values / LOVs > Click on the relevant LOV.



2. On the List of Values tab, double-click the value column of the **Use Ids on values** field.

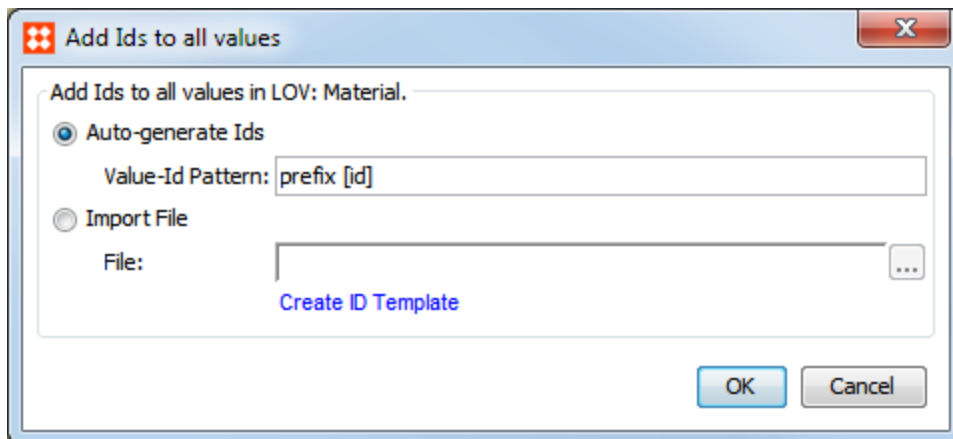


3. On the Add IDs to all values dialog, select the radio button to determine how IDs are created. Each option is addressed below.

Auto-generate IDs option

This option allows customization of the ID by using a specified Value-ID Pattern with an auto-generated ID.

1. Select the Auto-generate IDs radio button.
2. In the Value-ID Pattern field, add the desired value with '[id]'. '[id]' is case sensitive. Additional text can precede or follow the [id], if needed.



3. Click the **OK** button to apply the Value-ID Pattern on existing values. The LOV editor will display the newly applied ID added to the existing value according to the ID pattern and the 'Value ID' column will display the respective IDs for the present values within the Values flipper (as shown below).

List of Values		References	Log	State Log	Tasks
Description					
Name	>	>	Value	>	
ID			HarmonizationCodeLOV		
Name			Harmonization Code		
Edited by			2017-08-08 10:27:24 by USERM		
Path			Lists of Values / LOVs/Harmonizati		
Dimension Dependencies					
Use Ids on values			Yes	←	
Use Ids for sorting			No		
Value-ID Pattern			prefix [id]	←	
In Attribute Groups					
List of Values Validation					
Values					
Values			Value ID	>	
> 3208200000			prefix 233427		
> 3304995000			prefix 233418		
> 3403195000			prefix 233431		

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**

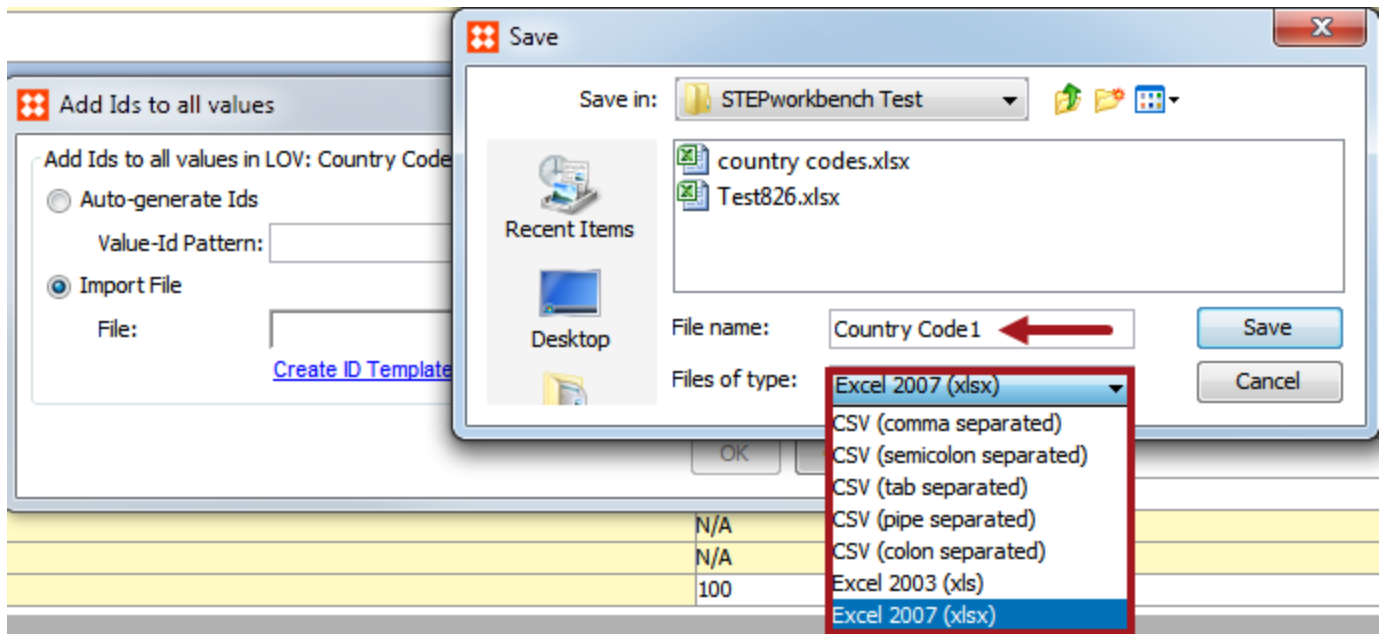
File: ...

Create ID Template

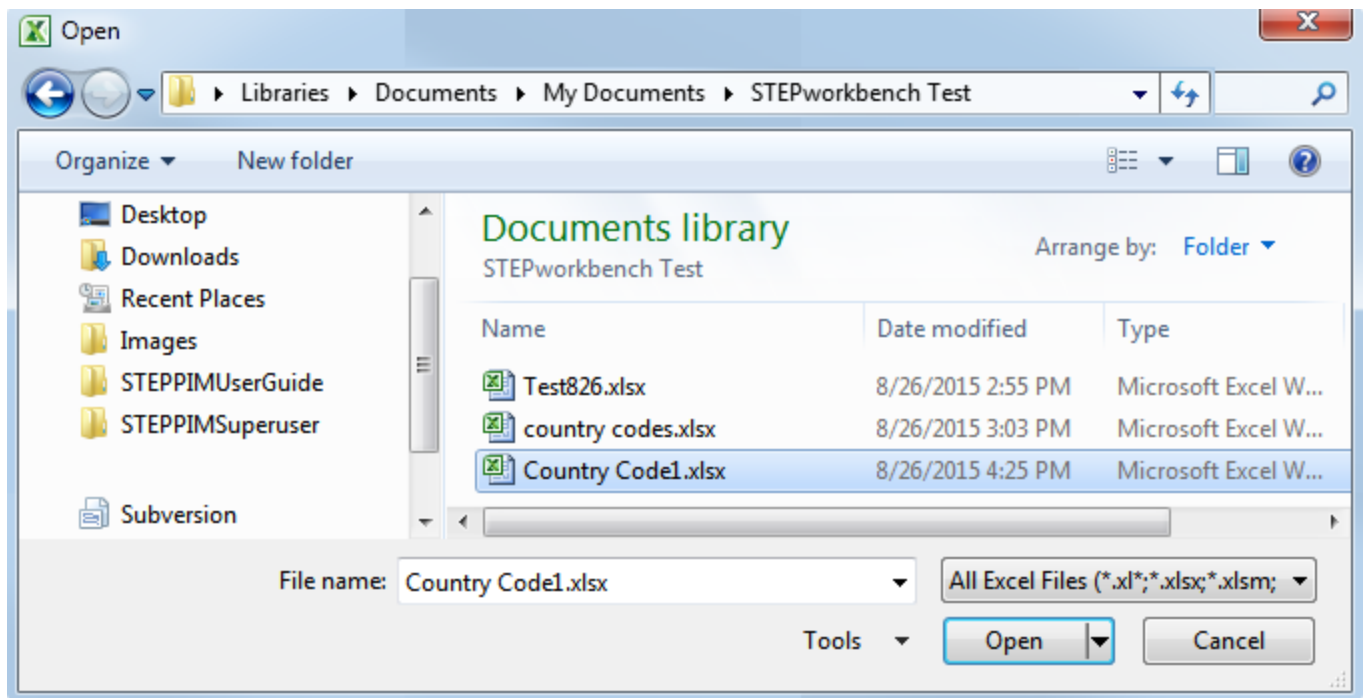
OK Cancel

3. Choose destination of the saved template and name the template.

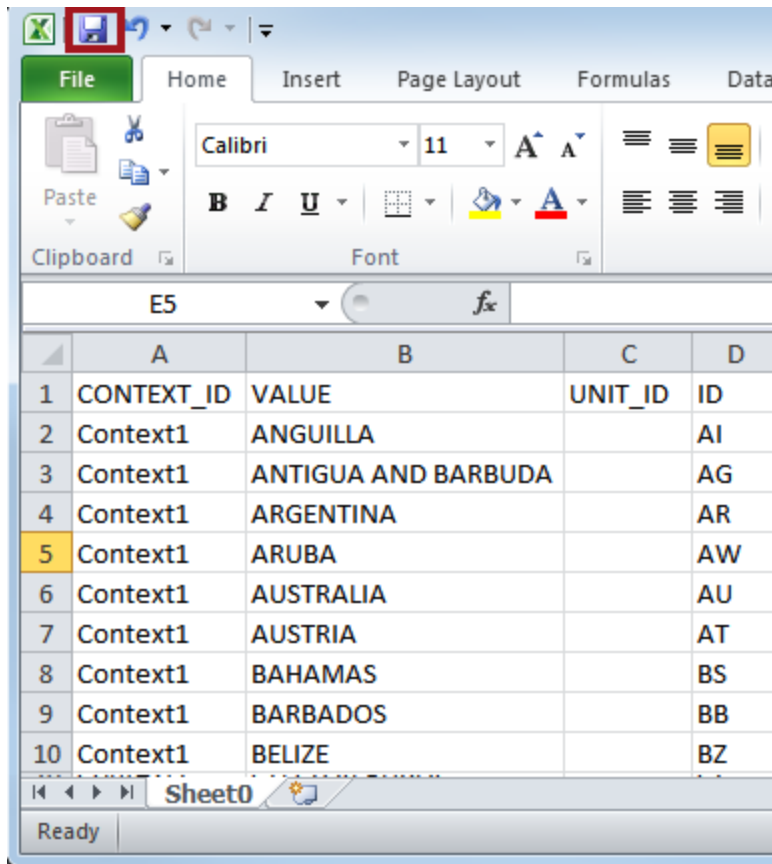
Note: An ID template can be saved in CSV (comma, semicolon, tab, pipe or color separated) or Excel (2003 or 2007) format. It is recommended that users use the newest, supportable version of Excel.



4. Click **Save**, leaving the **Add IDs to all values** dialog open.
5. Open your template using to the program you used to create it.

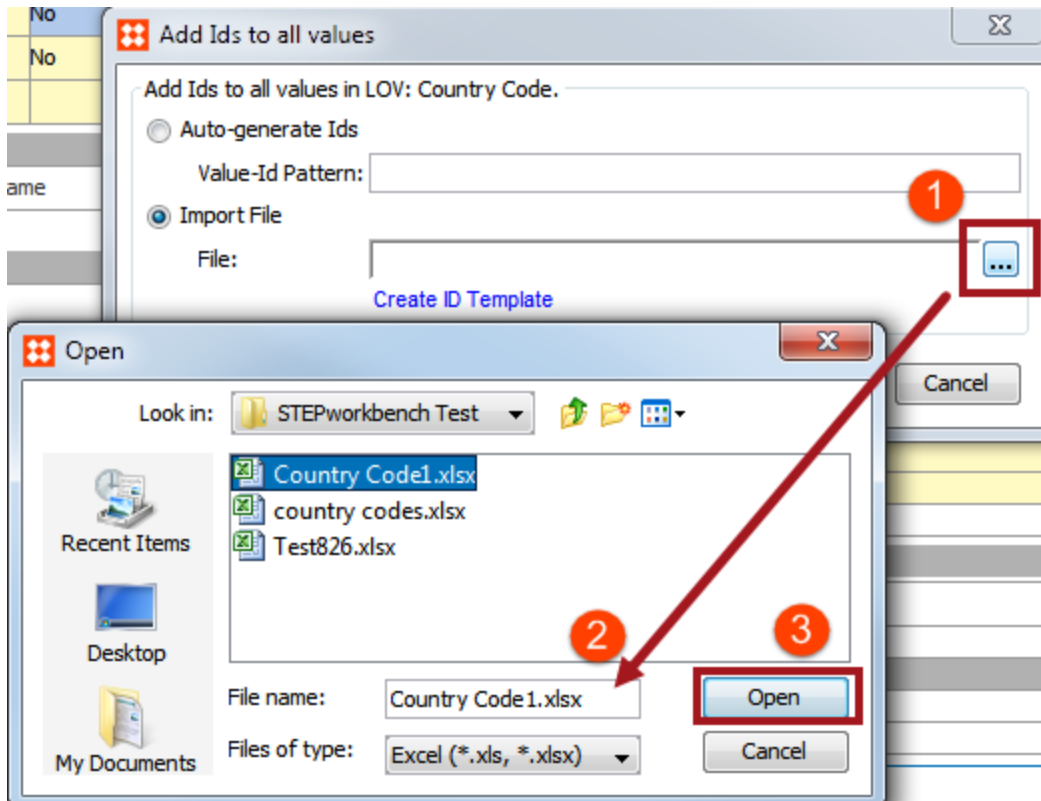


6. Once the template file is open, populate the name and ID column accordingly, and save your file.

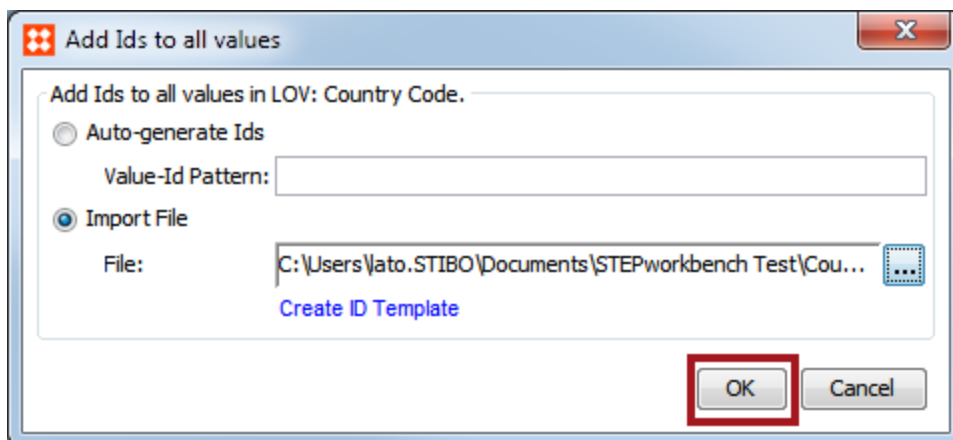


Note: Unit ID column is only filled if there are units (i.e., inches, pounds, centimeters).

- 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 the **OK** button.



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, case sensitive, 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 duplicated values 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

The Merge LOVs option can be used when LOVs with similar characteristics can be combined into one LOV, easing the maintenance burden.

Below are Steps to Merge LOVs:

1. Go to System Setup > Lists of Values / LOVs > click on **each of the LOVs to be merged** and ensure the settings within the Dimension Dependencies parameter match. For the example below, the two attributes 'Color' and 'Colour' have the same value for the Dimension Dependencies parameter.

The image shows two screenshots of the Stibo Systems System Setup interface, illustrating the configuration of two LOVs: 'Color' and 'Colour'.

Top Screenshot (Color LOV):

- System Setup** (Left Sidebar): Lists of Values / LOVs > Color
- Multi Editor** (Right Panel): List of Values tab
- Description** (Table):

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

Bottom Screenshot (Colour LOV):

- System Setup** (Left Sidebar): Lists of Values / LOVs > Colour
- Multi Editor** (Right Panel): List of Values tab
- Description** (Table):

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]

For this example, prior to merging the LOVs, the values for each of the LOVs can be seen in the screenshot below.

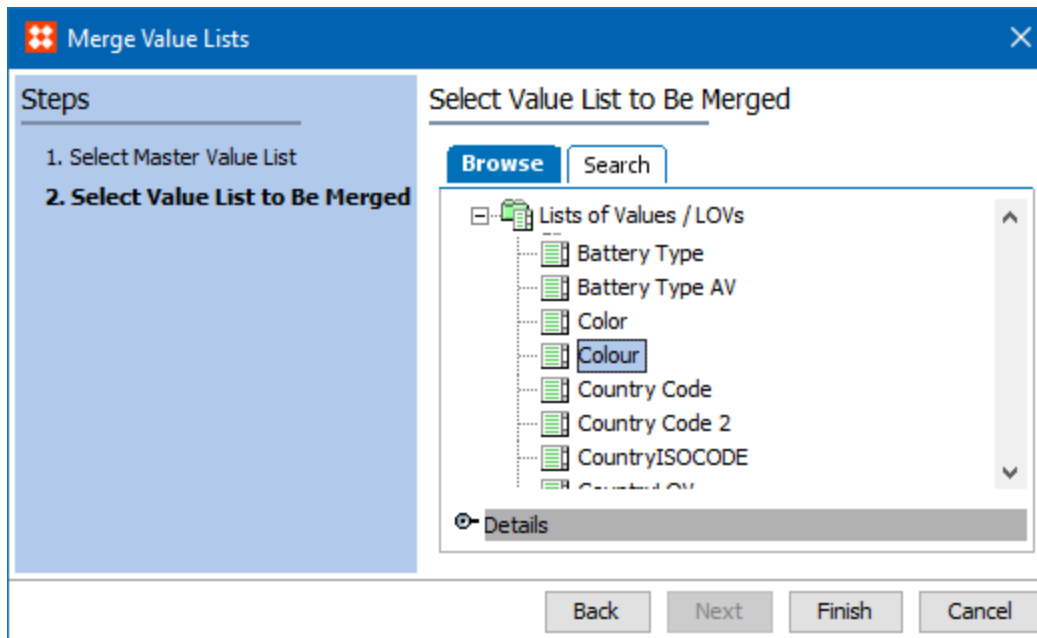
Color - List of Values	
Log	State Log
Tasks	
List of Values	References
Description	
In Attribute Groups	
List of Values Validation	
Values	
Values	Value ID
> Black	179464
> Bluish	138809
> Gray	138698
> Green	179462
> Multi-color	138702
> Orange	138700
> Red	138808
> White	138699
> Yellow	179461
Add Value	

Colour - List of Values	
Log	State Log
Tasks	
List of Values	References
Description	
In Attribute Groups	
List of Values Validation	
Values	
Values	Value ID
> chartreuse	233435
> Multi-colour	233434
> Pink	233433
Add Value	

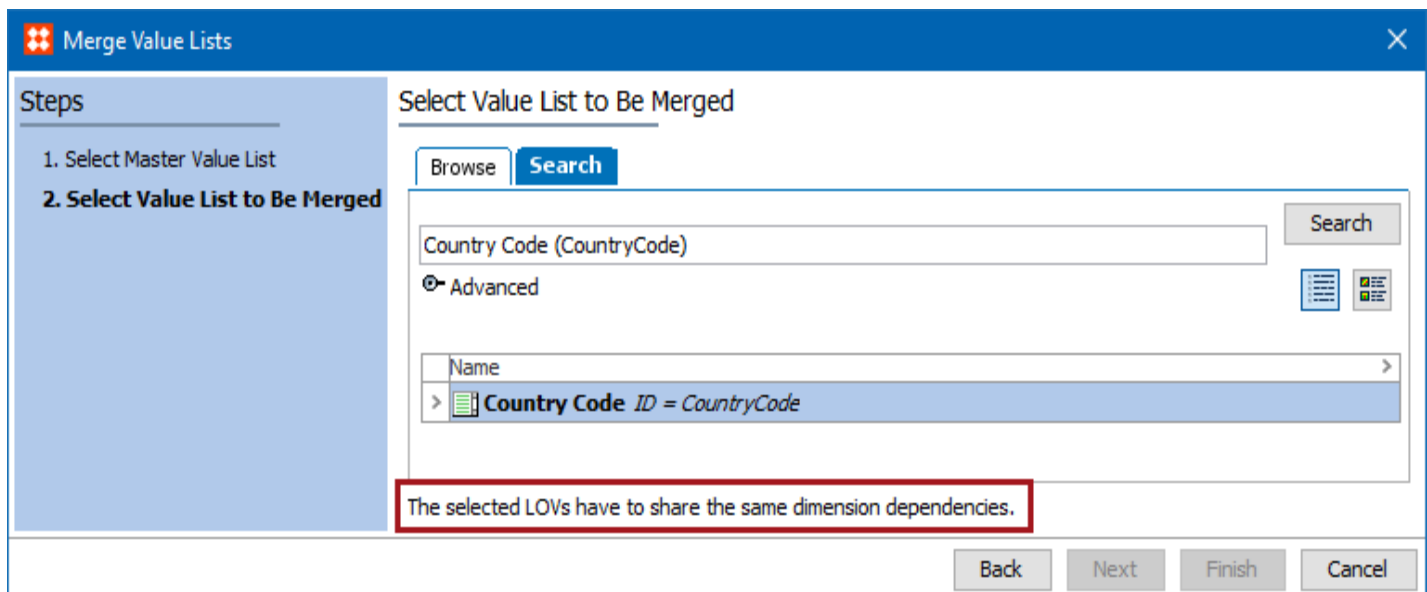
- Highlight the master LOV (the LOV that will remain after the merge), right-click and select **Merge LOVs**.

The screenshot shows the 'System Setup' window with a tree view of Lists of Values / LOVs. The 'Color' LOV is selected. A right-click context menu is displayed over the 'Color' LOV, with the 'Merge LOVs' option highlighted by a red rectangle. The menu also includes options like Cut, Delete, Duplicate, Merge Values in LOV, Initiate in STEP Workflow, and Remove from STEP Workflow.

- The Merge Value Lists wizard will display. Use the Browse or Search tabs to find and select the LOV to be merged (the LOV that will be deleted) with the master LOV.



If an LOV with dimension dependency values that are different from the master LOV is selected, then a warning will display and the Finish button will not be enabled (as shown below).



4. Click the **Finish** button to complete the merge process. The master LOV remains with all the values from the two LOVs, and the merged LOV has been deleted.

System Setup

Lists of Values / LOVs

Battery Type

Color

Country Code

Country Codes

Color - List of Values

LogState LogTasks

List of ValuesReferences

Description

In Attribute Groups

List of Values Validation

Values

Values	Value ID
> Black	179464
> Bluish	138809
> chartreuse	233435
> Gray	138698
> Green	179462
> Multi-color	138702
> Multi-colour	233434
> Orange	138700
> Pink	233433
> Red	138808
> White	138699
> Yellow	179461

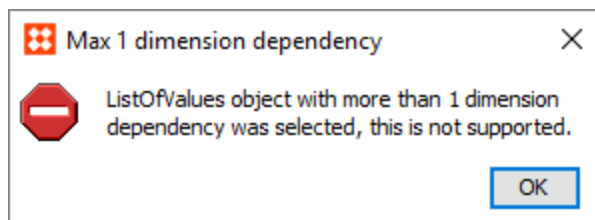
Add Value

Merging Values in an LOV

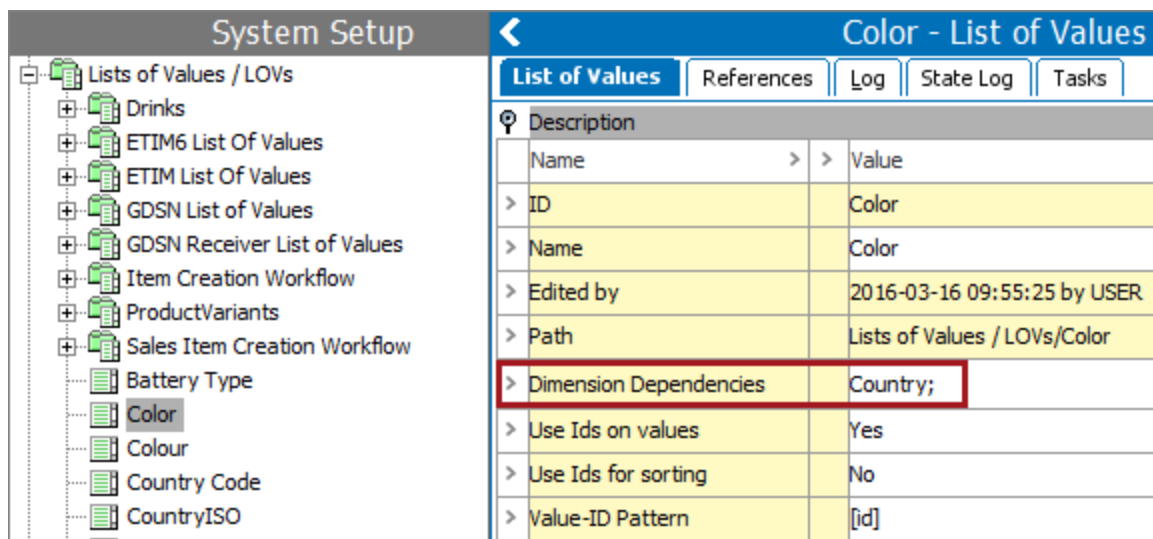
If multiple sets of duplicate values exist within a single LOV, run the merge operation multiple times to eliminate each of the duplicate sets. This also applies to dimension dependent LOVs.

Prerequisites

To use the Merge Values in LOV function, the LOVs must have less than two dimension dependencies, otherwise a warning dialog will display as shown below.

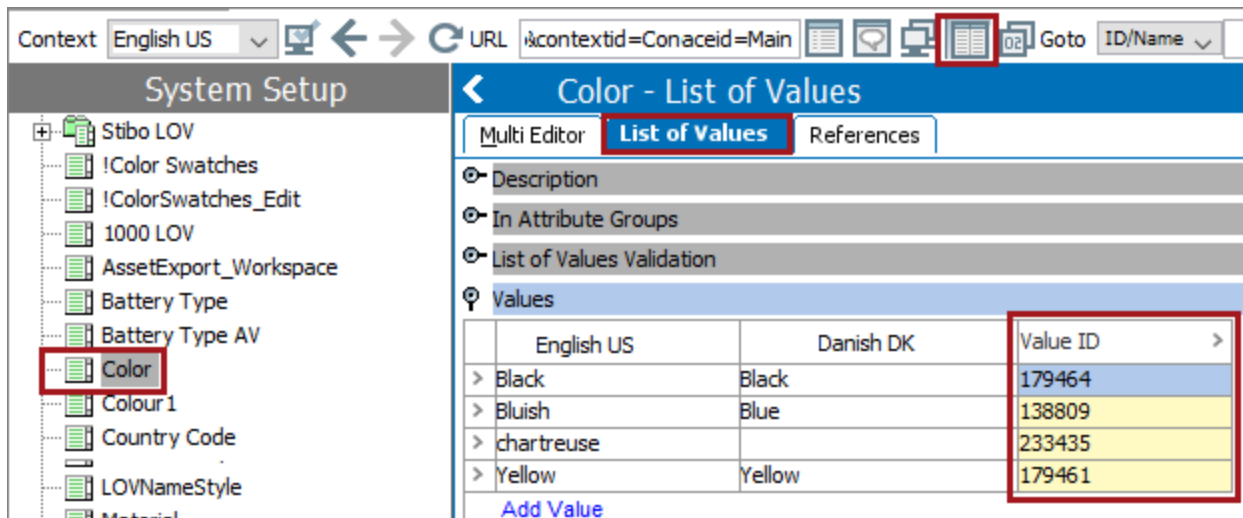


To confirm the LOV value has less than two dimension dependencies go to System Setup > Lists of Values / LOVs > select the desired LOV > Click the List of Values Tab in the editor and confirm the Dimension Dependencies parameter lists no more than one dimension dependency.



To use the Merge Values in LOV function, the LOVs must use value IDs, otherwise the menu option to Merge Values in LOV will be grayed out.

To confirm the LOV uses value IDs, go to Workbench > System Setup > Select the desired LOV > Select the List of Values tab > Click the **Context Mode** button in the toolbar > Expand the Values flipper > Look for the Value ID column as shown below.



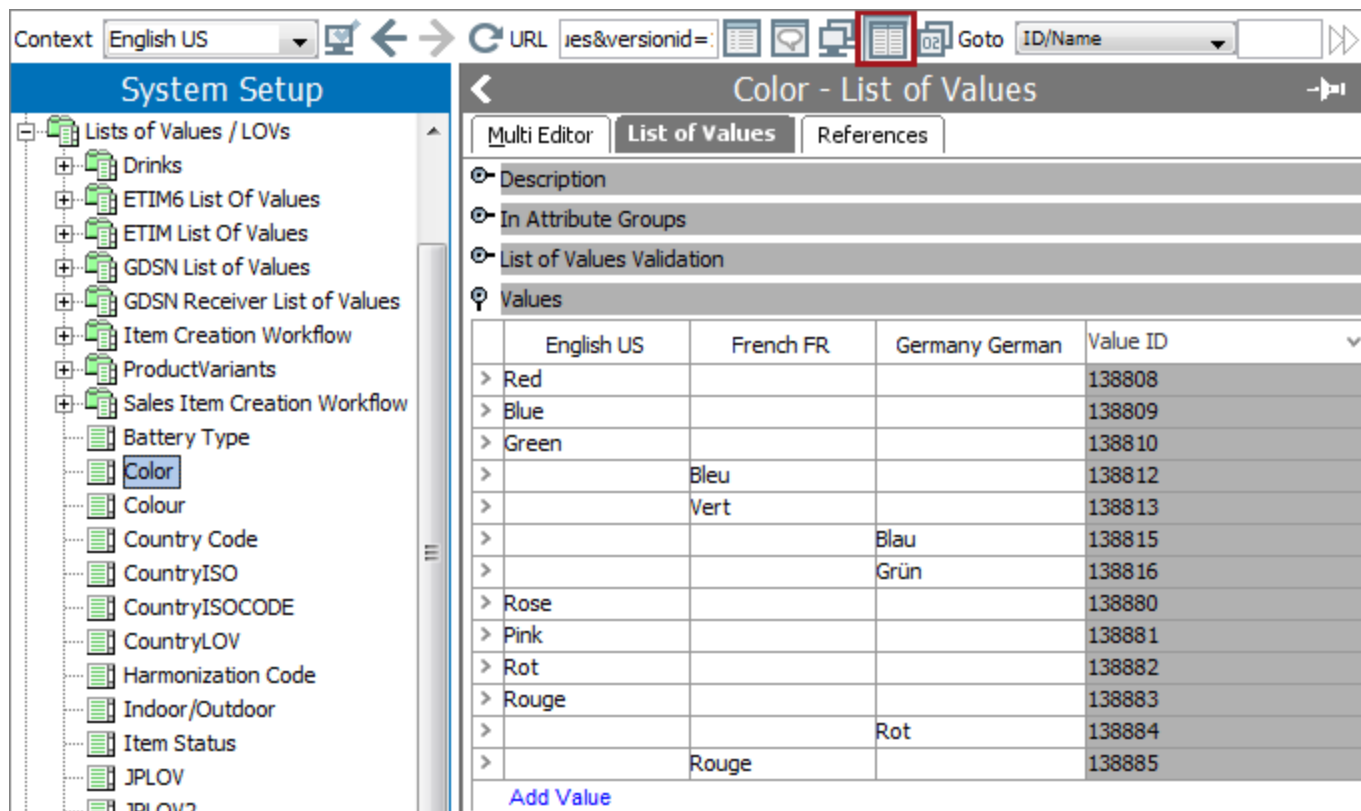
Note: Because the LOVs are dimension dependent each LOV value has a different 'Value ID'. This is important to notice for later steps.

If the desired LOV needs to have value IDs applied, see the **Adding IDs to Values in LOV** topic within this guide.

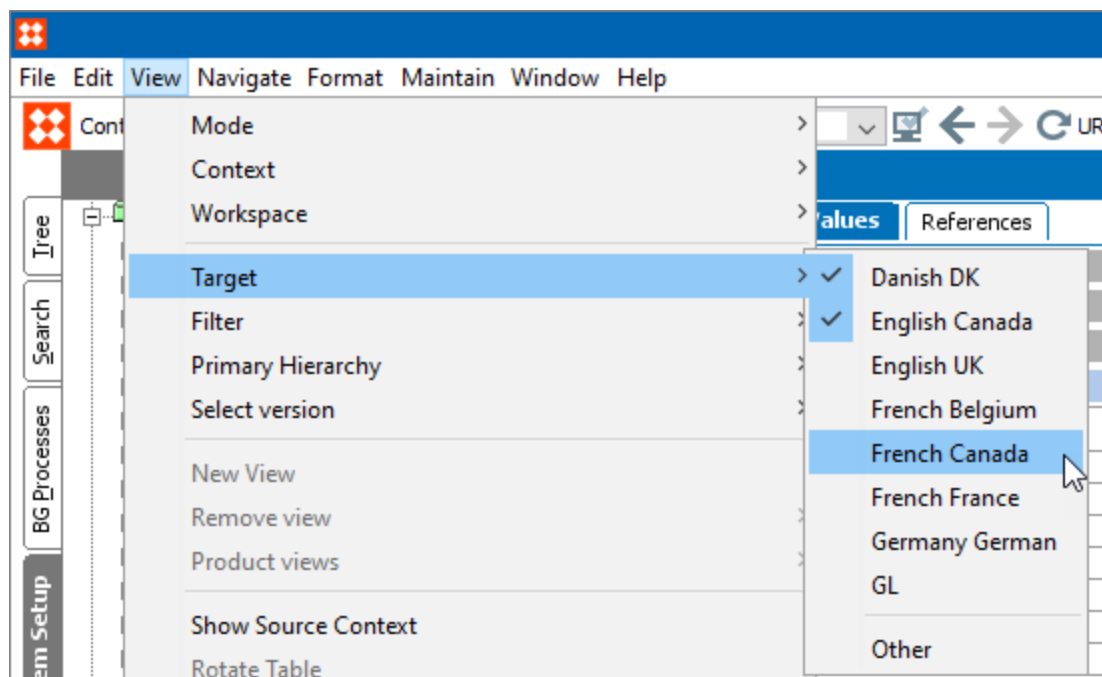
Steps to Merge Values in an LOV

Once the requirements have been confirmed for the LOV, follow the steps below to merge values in an LOV.

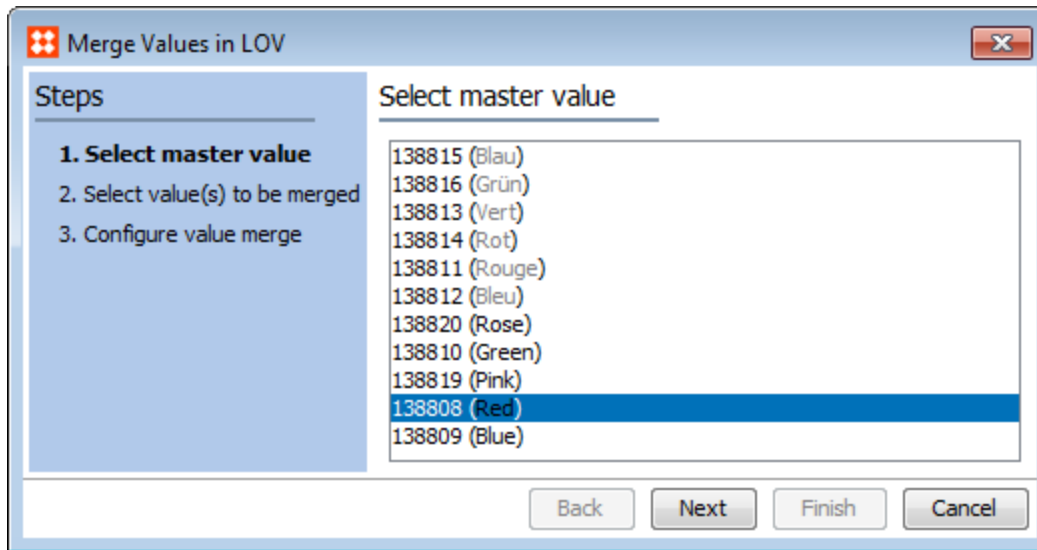
1. Go to Workbench > System Setup > Select the desired LOV > Select the List of Values tab > Click the **Context Mode** button in the toolbar to view all available values for an LOV in other contexts.



- Optionally, choose the desired Target Contexts to display within the editor. Go to Workbench > Menu bar > View > Target > Click on the desired contexts to display.

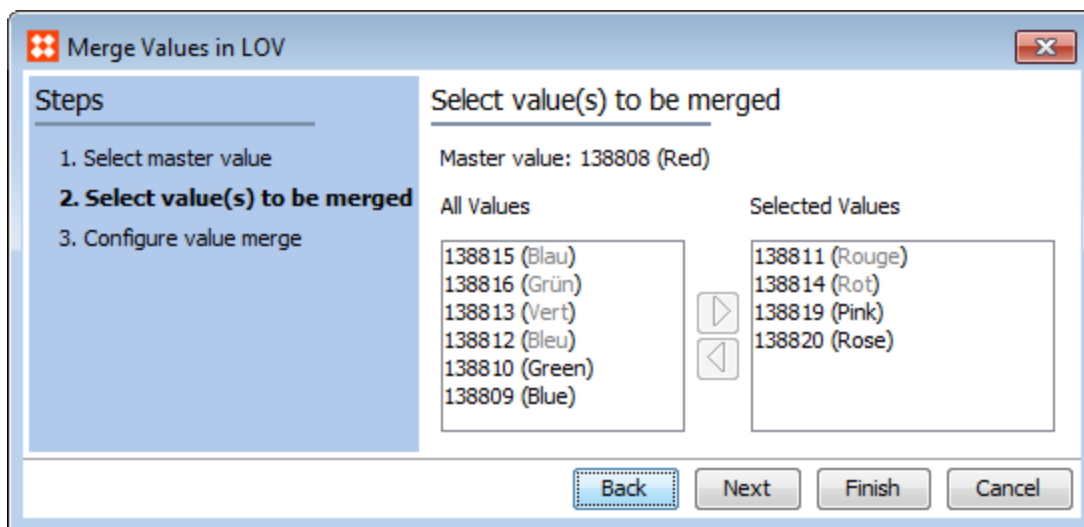


- Go to the System Setup > Lists of Values / LOVs > Select the desired LOV > Right click the selected LOV > select **Merge Values in LOV**. The Merge Values in LOV dialog will display.



- Select the value to be the master (the value that will remain after the merge), and click the **Next** button.
- 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.



- The Configure value merge dialog will display. Within the Merged / Deleted LOV values table, use the handles to drag LOV values and reorder as necessary. The topmost value in the respective column will be the master for the respective context. For example, in the screenshot below, when viewing the LOV using the 'France'

context the value ID '138808' will display as 'Rouge,' whereas for the 'Germany' context the same value ID will display as 'Rot.'

Merge Values in LOV

Steps

1. Select master value
2. Select value(s) to be merged
3. **Configure value merge**

Configure value merge

Value ID	France	Germany	USA
138808	Rouge	Rot	Red

Merged/Deleted LOV values

Drag and drop to prioritize LOV values

Value ID	France	Germany	USA
138808			Red
138811	Rouge		
138814		Rot	
138820			Rose
138814			Rot
138819			Pink

Back

Next

Finish

Cancel

- Click the **Finish** button to complete the merge process. In the example below, the duplicate values that were in the 'USA' context have been merged, and the values for the other context are aligned with the single ID (138808).

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	Rouge	Rot	138808
> Blue			138809
> Green			138810
>	Bleu		138812
>	Vert		138813
>		Blau	138815
>		Grün	138816

Add Value

Merging LOV Values in Bulk

LOV values across multiple LOVs can be merged in bulk through the 'Merge LOV Values' feature, which is available as a right-click action on LOV groups in System Setup.

When 'Merge LOV Values' is selected, a dialog displays that prompts users to upload a semicolon-separated CSV file containing the IDs of the LOVs in which values should be merged, the ID of the source value, and the ID of the target value. Multiple LOVs can be actioned at once, with a limit of 1,000 total values (i.e. 1,000 lines in the file). This functionality greatly eases the maintenance of LOVs that contain misaligned, repeated, or incorrect values across multiple contexts.

Note: This functionality can only be used on LOVs that use value IDs. If your LOVs do not already contain value IDs, see the **Adding IDs to Values in LOV** topic for more information.

Prerequisites

Users must be granted the following setup actions in order to merge LOV values in bulk:

- View domain
- Merge value in hard/medium domain
- Delete value in hard/medium domain

Creating and Formatting the CSV File

The Merge LOV Values functionality accepts a CSV with the following file format. Any other format will result in an error. The file must not contain headers.

Note: The file must either have a .csv or .txt file extension. STEP will not accept Excel file extensions (e.g. .xlsx)

LOVID	SourceValueID	TargetValueID
LOV1	VAL1	Value1
LOV1	VAL2	Value1
LOV2	VAL3	Value2

- The **LOVID** column accepts an LOVID
- The **SourceValueID** column accepts a LOV Value ID, which is to be used as source
- The **TargetValueID** column accepts a LOV Value ID, which is to be used as target for the merge

Note: The 'Merge LOV Values' CSV sheet must contain *semicolon*-separated values, not *comma*-separated values. If you are a user in a region where commas are not used as decimal separators in numbers—such as the United States—you may need to adjust your advanced Excel options or the regional settings in your operating system to change the default list separator from a comma to a semicolon. Additional information on how to adjust these settings can be found on the web.

The below screenshot shows a sample file that can be used to merge values within two different LOVs at one time—one with the ID of PrimaryColors and a second with the ID of ColorSwatches. Column B contains the IDs of the LOV values that will be removed (source values) when the file is imported. Column C contains the IDs of the LOV values that will remain (target values) after the file is imported. (Even though multiple LOVs can be included in a single sheet, values are only merged within the same LOV, not across multiple LOVs.)

	A	B	C
1	PrimaryColors	166545	Absolute_Zero
2	PrimaryColors	7887456	Acid_Green
3	PrimaryColors	385676	Aero
4	PrimaryColors	17118686	Aero_Blue
5	PrimaryColors	43155254	African_Violet
6	ColorSwatches	16146545	Absolute_Zero
7	ColorSwatches	787456	Acid_Green
8	ColorSwatches	37785676	Aero
9	ColorSwatches	178686	Aero_Blue
10	ColorSwatches	431254	African_Violet

A .txt file containing semicolon-separated values may also be used.

```

LOV_Value_Merge.txt - Notepad
File Edit Format View Help
PrimaryColors;166545;Absolute_Zero
PrimaryColors;7887456;Acid_Green
PrimaryColors;385676;Aero
PrimaryColors;17118686;Aero_Blue
PrimaryColors;43155254;African_Violet
ColorSwatches;16146545;Absolute_Zero
ColorSwatches;787456;Acid_Green
ColorSwatches;37785676;Aero
ColorSwatches;178686;Aero_Blue
ColorSwatches;431254;African_Violet

```

The following screenshot shows the same sample CSV from above but with a highlighted example of two values that will be merged. After the file is imported, the 'Absolute Zero' value with the ID of 16146545 will be removed and the 'Absolute Zero' value with the ID of Absolute_Zero will remain.

	A	B	C
1	PrimaryColors	166545	Absolute_Zero
2	PrimaryColors	7887456	Acid_Green
3	PrimaryColors	385676	Aero
4	PrimaryColors	17118686	Aero_Blue
5	PrimaryColors	43155254	African_Violet
6	ColorSwatches	16146545	Absolute_Zero
7	ColorSwatches	787456	Acid_Green
8	ColorSwatches	37785676	Aero
9	ColorSwatches	178686	Aero_Blue
10	ColorSwatches	431254	African_Violet

System Setup

- GDSN List of Values
- GDSN Receiver List of Values
- Item Creation Workflow
- Mabe
- Match_and_Merge_LOV_Group
- Print Publishing LOVs
- ProductVariants
- Sales Item Creation Workflow
- Stibo LOV
- Color Swatches
- Primary Colors
- 1000 LOV
- AssetExport_Workspace
- Battery Type
- Battery Type AV

Color Swatches - List of Values

List of Values | References | Log | State Log | T

List of Values Validation

Values

Values	Value ID
Absolute Zero	16146545
Absolute Zero	Absolute_Zero
Acid Green	787456
Acid Green	Acid_Green
Aero	Aero
Aero	37785676
Aero Blue	Aero_Blue
Aero Blue	178686
African Violet	African_Violet
African Violet	431254

After the sheet is uploaded, the ColorSwatches LOV in this example will look as follows, with only the values in column C of the CSV sheet remaining.

Color Swatches - List of Values

List of Values | References | Log | State Log

Description

In Attribute Groups

List of Values Validation

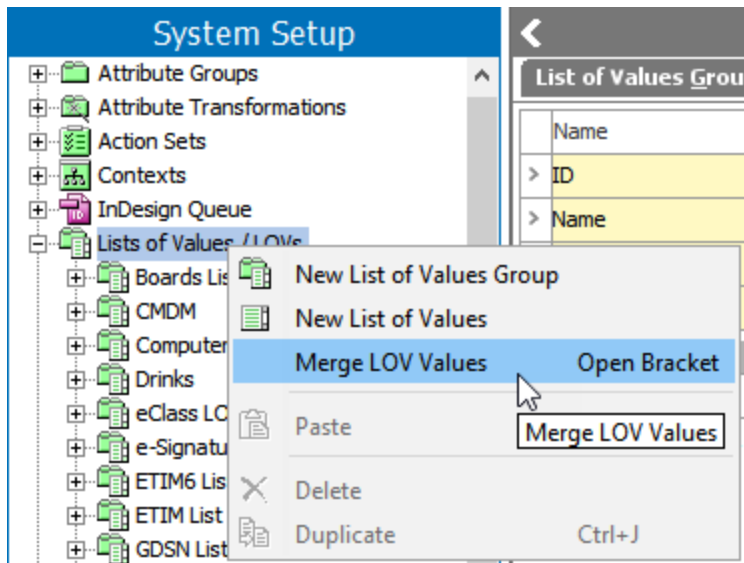
Values

Values	Value ID
> Absolute Zero	Absolute_Zero
> Acid Green	Acid_Green
> Aero	Aero
> Aero Blue	Aero_Blue
> African Violet	African_Violet

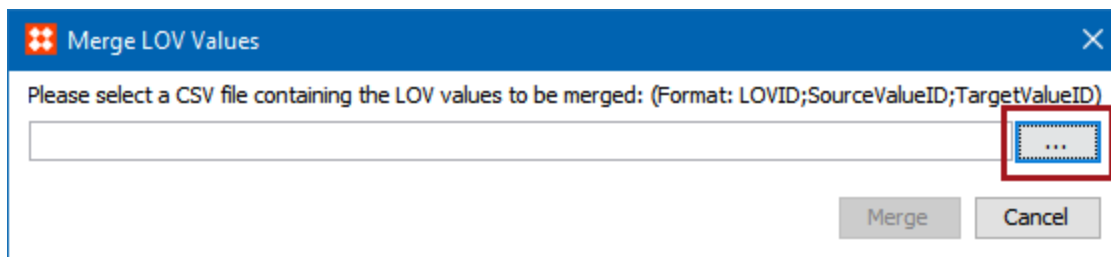
Uploading the Merge LOV Values CSV Sheet

1. Navigate to System Setup > Lists of Values / LOVs, then right-click on any LOV group (including the root node) and select **Merge LOV Values**.

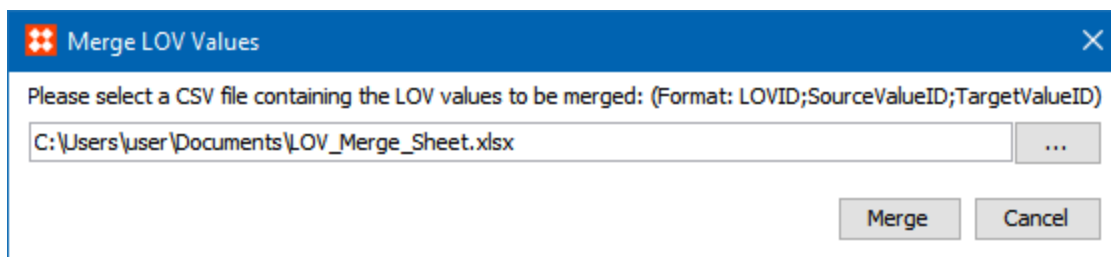
Note: 'Merge LOV Values' should not be confused with 'Merge Values in LOV,' which is the manual process for merging values within an individual LOV and is only available when right-clicking on an LOV, not on an LOV group. For more information on this function, see the **Merging Values in an LOV** topic.



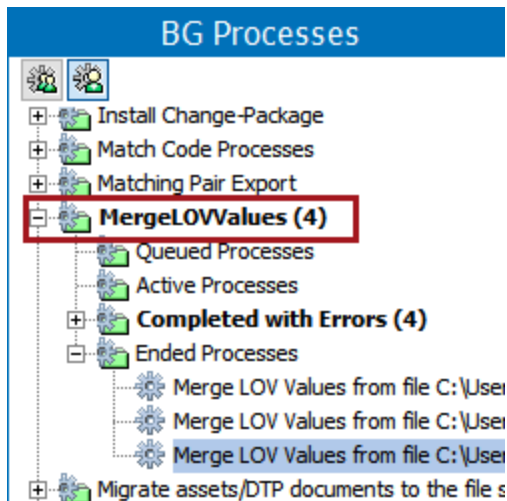
2. Click the ellipsis button (...) on the **Merge LOV Values** dialog to select the CSV file containing the LOV values to be merged.



3. Navigate to your CSV file in the Select File dialog, then click **Select File**.
4. After selecting your file, click **Merge** on the Merge LOV Values dialog.



5. A 'Background process status' dialog displays. Click 'Go to process' if you would like to view the background process for the LOV value merge.
6. The background process will display on the BG Processes tab within the 'MergeLOVValues' queue.



Monitoring LOV Value Merge Events

When values in an LOV are merged, STEP generates events that can be monitored in an event processor that uses the 'Expand LOV Value Changes' event processor plugin. For more information, see the **Expand LOV Value Changes Processing Plugin Parameters and Triggers** section of the **Event Processors** documentation.

Deleting LOV Values in Bulk

Values can be deleted in bulk from a List of Values (LOV) by importing a STEPXML file that contains a **<DeleteValue/>** tag nested within the parent **<ListOfValue></ListOfValue>** tag.

This file is imported into STEP like any other STEPXML file, by navigating to File > Import > Data. For more information on how to import files into STEP, see the **Creating a Data Import** section of the **Import Manager** documentation.

Note: This functionality can only be used on LOVs that use value IDs. If your LOVs do not already contain value IDs, see the **Adding IDs to Values in LOV** topic for more information.

Prerequisites

Users must be granted the following setup actions in order to delete LOV values in bulk:

- View domain
- Delete value in hard/medium domain

DeleteValue Tag and XML Attributes

The **<DeleteValue/>** tag can contain three XML attributes: **ID**, **Force**, and **GenerateEvents**. Of these three, **ID** is mandatory, as it identifies the value to be deleted from the LOV. The **Force** and **GenerateEvents** attributes are optional.

```
<ListsOfValues>
  <ListOfValue ID="ColorSwatches" ParentID="List Of Values group root"
    AllowUserValueAddition="false" UseValueID="true" Selected="true" Referenced="true">
    <Name>Color Swatches</Name>
    <Validation BaseType="text" MinValue="" MaxValue="" MaxLength="100" InputMask=""/>
    <DeleteValue Force="true" GenerateEvents="true" ID="Red"/>
  </ListOfValue>
</ListsOfValues>
```

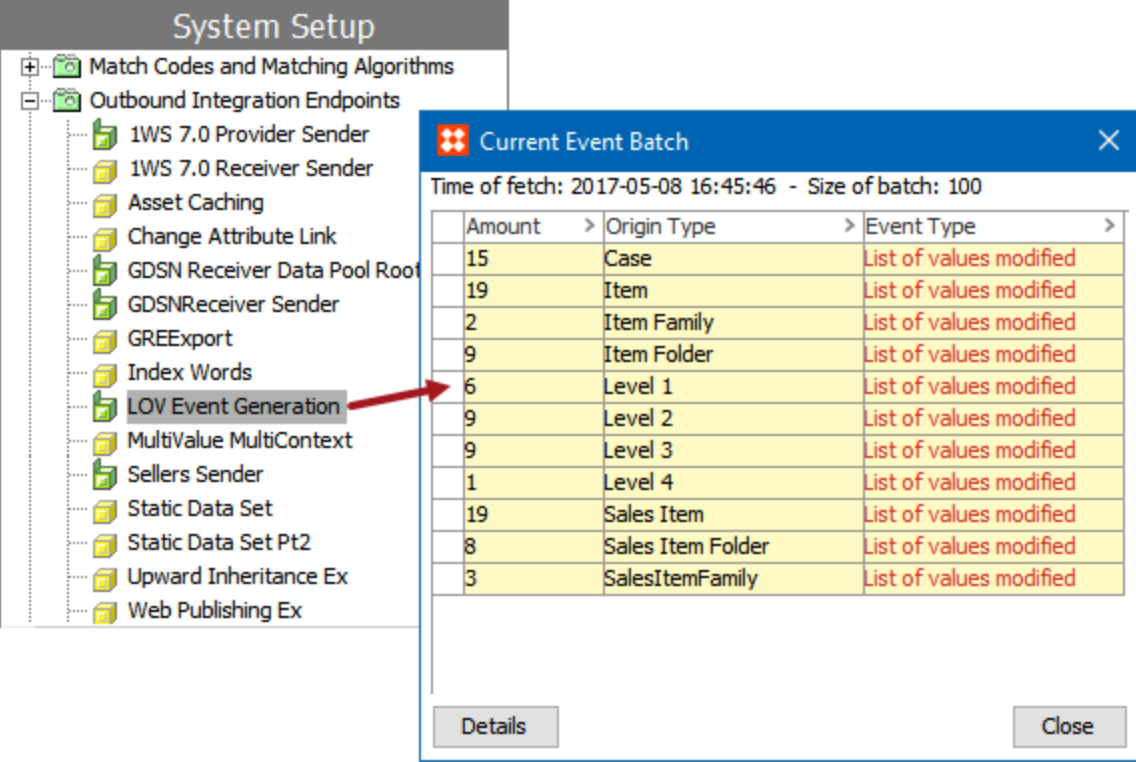
Force can have a value of 'true' or 'false.' If set to true, the value will be deleted from the LOV even if there are attributes using the value. If approved objects in STEP are using the attribute that contained the deleted value, the approval status of these objects will not be affected, i.e. the objects will remain approved.

If **Force** is set to 'false,' or if the **Force** attribute is excluded altogether, the value will be deleted as long as there are no attributes using the value. If attributes are using the value, the import will fail, and the value will not be deleted from the LOV. The default value for 'Force' is false.

GenerateEvents also can have a value of 'true' or 'false.' If set to true, events will be generated on objects that are using the attribute that contained the force-deleted value. If set to false, no events will be generated. The default value for 'GenerateEvents' is false.

An example of how to tell if events have been generated is to configure an event-based outbound integration endpoint (OIEP) to listen for 'Modify' events on object types that are affected when the LOV value is deleted. The configuration also specifies which attributes will trigger the events; in this case, the attribute(s) that use the LOV(s) from which the values have been deleted. As with this example, any object that had an attribute value removed when the value was force deleted from the LOV will display as a 'List of values modified' event.

Note: Approve events cannot be generated for deletions of LOV values. I.e. if a product uses an attribute based on the LOV and contained the value that was deleted from the LOV, the deletion of the LOV values through the STEPXML import does not unapprove the product.



The screenshot shows the 'System Setup' window with the 'Outbound Integration Endpoints' tree. The 'LOV Event Generation' endpoint is selected, indicated by a red arrow. A 'Current Event Batch' window is open, displaying a table of events generated during a fetch operation on 2017-05-08 at 16:45:46, with a batch size of 100.

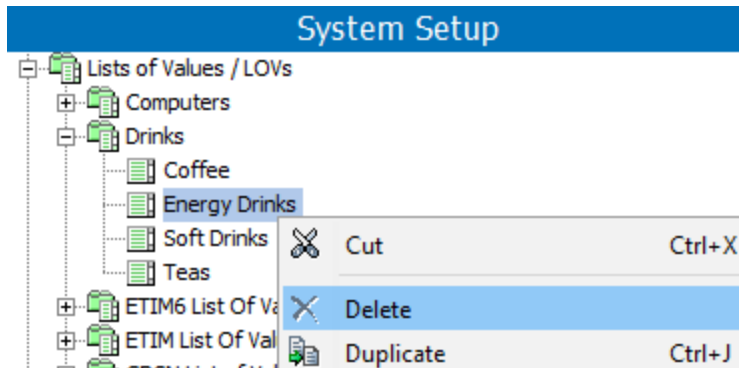
Amount	Origin Type	Event Type
15	Case	List of values modified
19	Item	List of values modified
2	Item Family	List of values modified
9	Item Folder	List of values modified
6	Level 1	List of values modified
9	Level 2	List of values modified
9	Level 3	List of values modified
1	Level 4	List of values modified
19	Sales Item	List of values modified
8	Sales Item Folder	List of values modified
3	SalesItemFamily	List of values modified

The 'Current Event Batch' window also includes 'Details' and 'Close' buttons at the bottom.

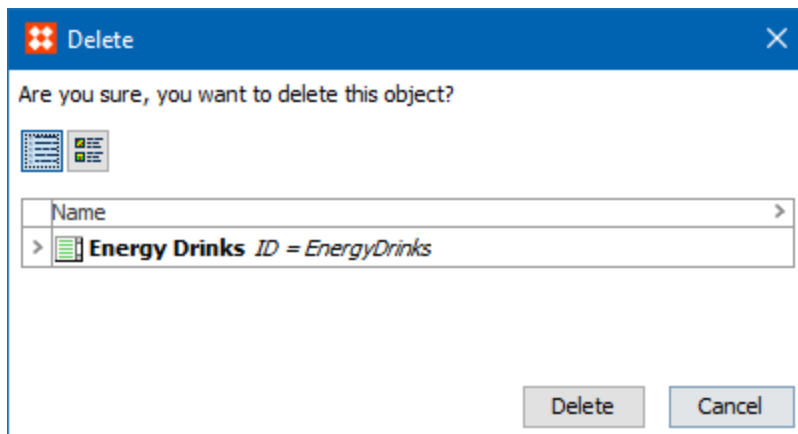
For more information on event generation, see the **Events** section of the **System Setup / Super User Guide** documentation. For more information on event-based OIEPs, see the **Creating an Event-Based Outbound Integration Endpoint** section of the **Data Exchange** documentation.

Deleting LOVs

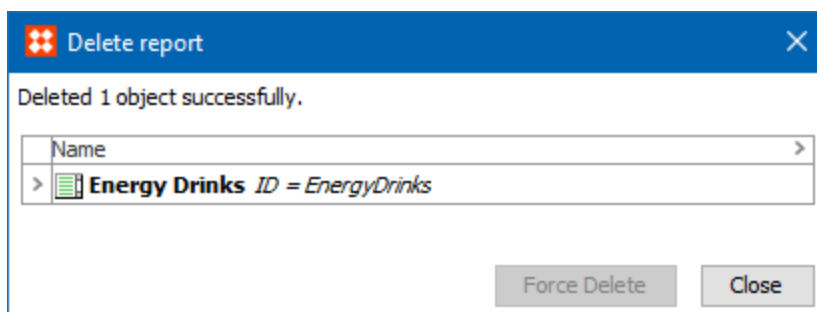
1. Go to System Setup > List of Values / LOVs > Select **the LOV to be deleted**.
2. Right-click **the LOV to be deleted**, and the menu will display as shown below.



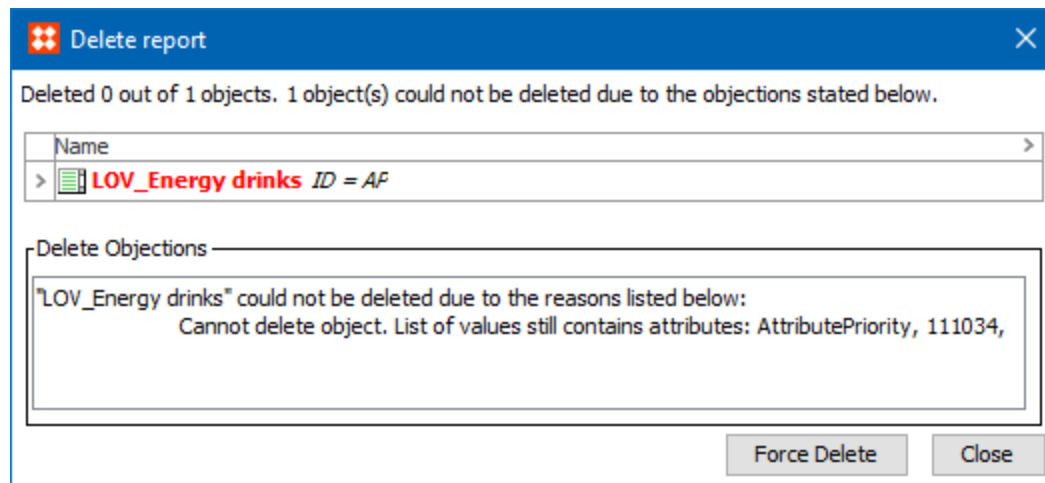
3. Click **Delete** and a Delete dialog displays.



4. Click the **Cancel** button to return to System Setup without deleting. Click the **Delete** button to delete the List of Values, and display the Delete report dialog.



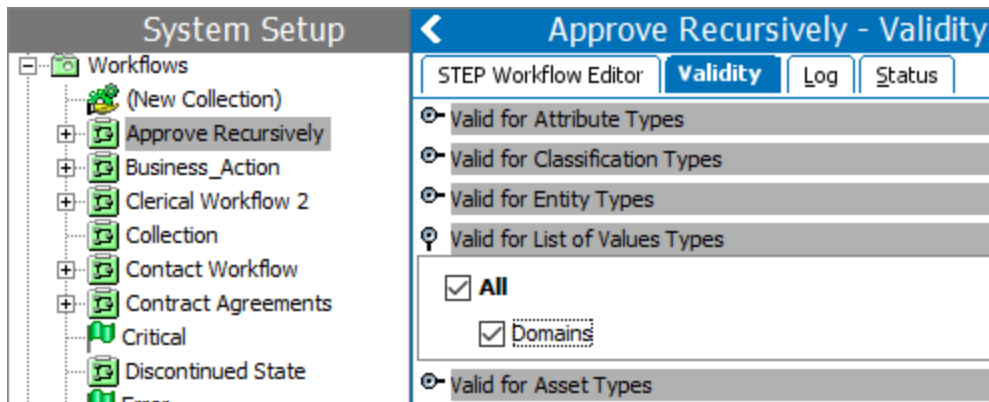
Note: If the LOV being deleted contains values used for an object, then the LOV cannot be deleted and the Delete report will display the Delete Objections (as shown below).



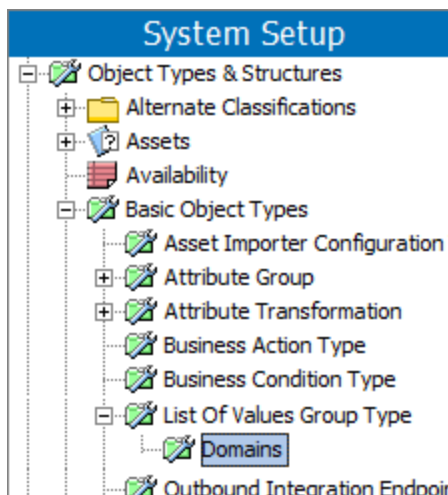
Initiating and Removing an LOV from Workflow

An LOV can be initiated into and/or removed from a workflow the same way as Products, Classification, Entities, and Assets. The LOVs can be assigned to a respective user or user group to update the LOV values, similar to the enrichment of product attributes.

The workflow should be valid for the LOV object type 'Domains' as shown in the image below.



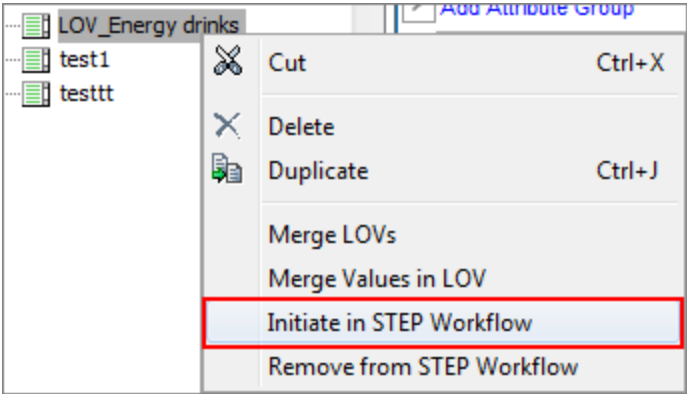
Domains are the only Object Type for LOVs and you can find them in the Workbench > System Setup > Object Types and Structures > Basic Object Types > List Of Values Group Type (as shown in the image below).



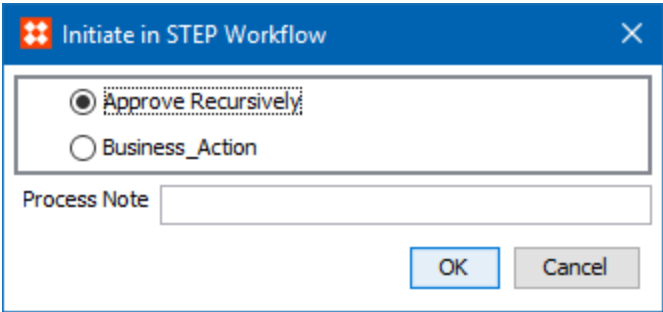
Steps to Initiate an LOV to a Workflow:

Below are steps to initiate one LOV to a Workflow, however more than one LOV can be initiated to the save workflow.

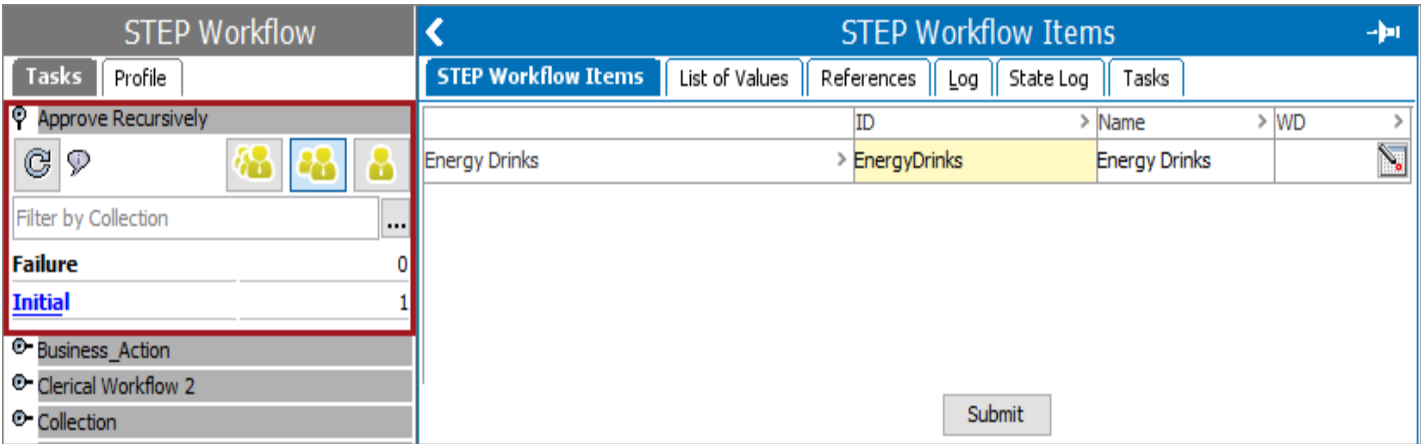
1. Right-click on an LOV, and select 'Initiate in STEP Workflow.'



- 2. The Initiate in STEP Workflow dialog will display listing only those workflows configured with validity for Domains. Select the appropriate workflow, optionally enter text within the Process Note parameter, and click the **OK** button.



The LOV has now been initiated into the workflow. This can be viewed by going to STEP Workflow > Expanding the respective workflow flipper > Selecting the workflow state and the workflow editor will display. Click the STEP Workflow Items tab and the LOV will display.

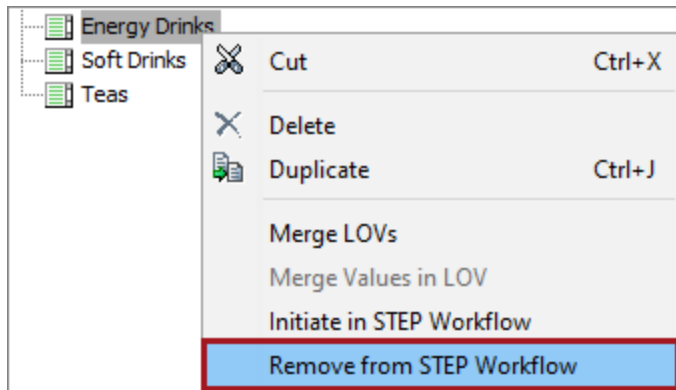


For more information, see the **Working with Tasks in Workflows** section of the **Workflows** documentation.

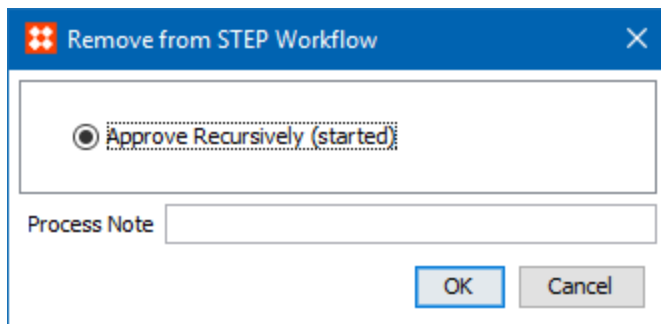
Steps to Remove an LOV from a Workflow:

Below are steps to remove an LOV from a Workflow.

1. Right-click on the LOV which is already in a workflow, and select the option 'Remove from the STEP Workflow' as shown below.



2. The Remove from STEP Workflow dialog will display with the workflow in which the LOV has been initiated for. If required add text to the Process Note parameter, then click the **OK** button.



The LOV has now been removed from the workflow. For more information on workflows, see the **Workflows** guide.

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.

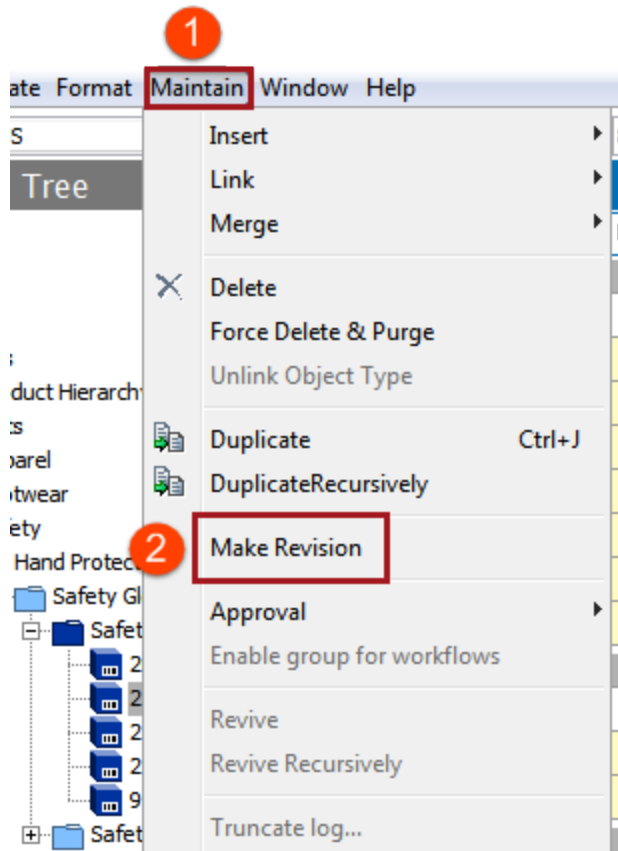
It is important to note the difference between *revisions* and *approvals*. The two concepts are related, but should not be confused. Objects that are *workspace revisable* are subject to approvals, and approving an object automatically generates a revision of it. However, objects that are *globally 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 / 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.

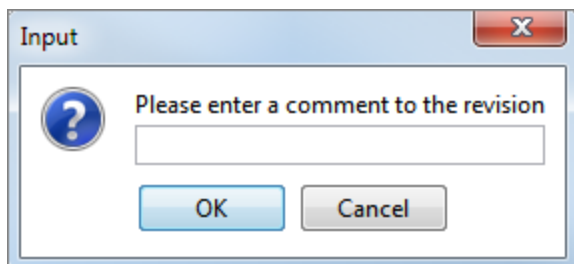
Generating Revisions

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.



Note: The manual creation of revisions is used when the data looks absolutely perfect, however there may be future changes that could be unexpected, and you need to revert back to when the data was perfect. Manual creation of revisions are not possible in Bulk, and should be done for each node at a time.

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. The same applies to references and link types when deleted.

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.

The screenshot shows the 'System Setup' interface. On the left is a navigation tree with icons and labels: Workflows, Derived Events, Object Types & Structures, Tags, Units, Users & Groups (highlighted), Reference Types, and Workspaces. The main panel is titled 'System Settings' and contains a 'System Settings' tab and a 'Log' button. Below these are three expandable sections: 'Table Defaults', 'Index Word Settings', and 'Revisability Settings'. The 'Revisability Settings' section is expanded, showing a table with two columns: 'Name' and 'Value'. The table contains one entry: 'Threshold (hours) for changes by same user to generate a new revision' with a value of '168'.

Name	Value
Threshold (hours) for changes by same user to generate a new revision	168

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.

Tree

- Upper body wear
 - T-shirts
 - T-shirts Items
 - 88723-12
 - Cotton T-Shirts
 - New Product 1234
 - 18210 M B
 - 18212 L B

New Product 1234 rev.0.6 - Status

61% complete

Images & DocumentsCommercialTablesCategory ProfileProof ViewStatusState LogTasks

Revisions

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.

Tree

- ys
 - ndcap Displays
 - Endcap Displays Items
 - Black & Decker Kitchen Display
 - 110594

Black & Decker Kitchen Display rev.0.4 - Status

Images & DocumentsCommercialTablesCategory ProfileProof ViewStatus

Revisions

Revision	Created	Edited	Major
> 0.4	Fri Jan 22 11:37:31 EST 2...	Fri Jan 22 11:37:31 EST 2...	
> 0.3	Wed Jan 13 11:49:33 EST...	Wed Jan 13 11:49:33 EST...	
> 0.2	Fri Oct 16 09:39:05 EDT 2...	Fri Oct 16 09:39:05 EDT 2...	
> 0.1	Fri Feb 13 11:36:40 EST 2...	Fri Feb 13 11:36:40 EST 2...	

CutCtrl+XCopCtrl+CCopCtrl+VasteCtrl+VasteLinkCtrl+LPurgeRevert to

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.

<

EVN-24 rev.0.5 - Status

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Tables

Category Profile

Proof View

Status

Revisions

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 cannot be recovered.

To purge a revision, click on the arrow next to the desired revision and select **Purge**.

20714 rev.0.3 - Status

Images & Documents

Commercial

Tables

Category Profile

Proof View

Status

State Log

Tasks

Revisions

Revision	Created	Edited	Major	User	Comment
0.3	Fri Jan 22 11:35:58 EST 2...	Fri Jan 22 11:35:58 EST 2...		USER	Auto Generated
0.2	Fri Oct 16 09:39:06 EDT 2...	Fri Oct 16 09:39:06 EDT 2...		USER	Complete approval
	Fri Feb 13 11:36:40 EST 2...	Fri Feb 13 11:36:40 EST 2...		STEPSYS	

Cut

Copy

Paste

Paste Link

Purge

Revert to

Ctrl+X

Ctrl+C

Ctrl+V

Ctrl+L

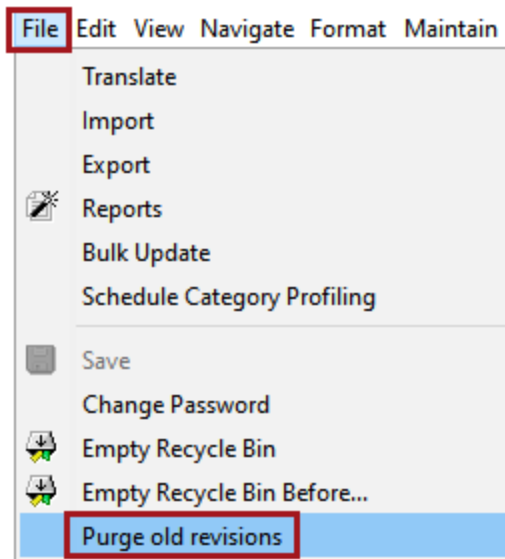
Name	Path	
Approved	Main/Approved	0.2
Main	Main	0.3

Revision Maintenance and Cleanup

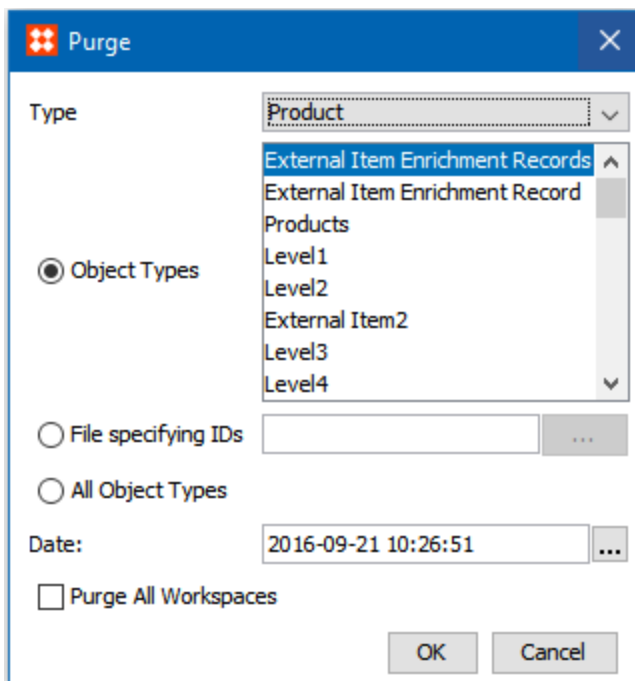
Planning for maintenance and cleanup of revisions is a critical step for all implementations and is best to be carried out early on in the setup of any system.

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 / Super User Guide documentation.

Comparing Revisions

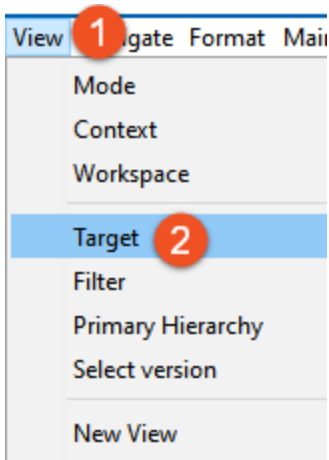
Users may compare revisions on an object within the Revision view. For instance, if a user needs to see how an attribute on a object has changed over the course of five revisions, the five relevant product revisions can be viewed at once on the same screen.

To enable revision comparing:

1. With the object selected, click the Revision mode button from the workbench menu bar.

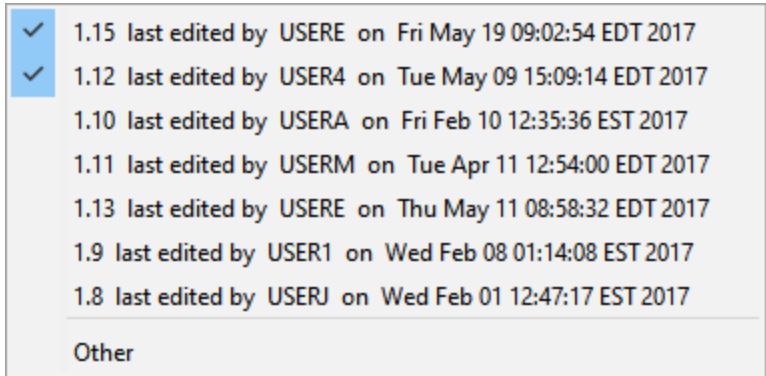


2. The two most recent revisions for that object will be visible. To view additional revisions, select 'View' from the toolbar, and then 'Target'.

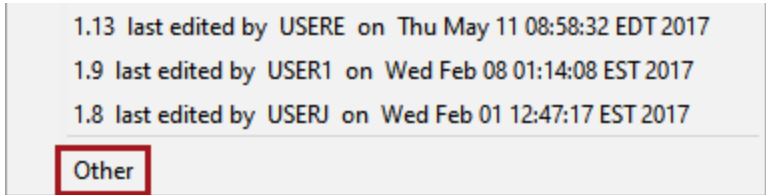


Note: Target will not be a selectable option unless the object is being viewed in the appropriate mode.

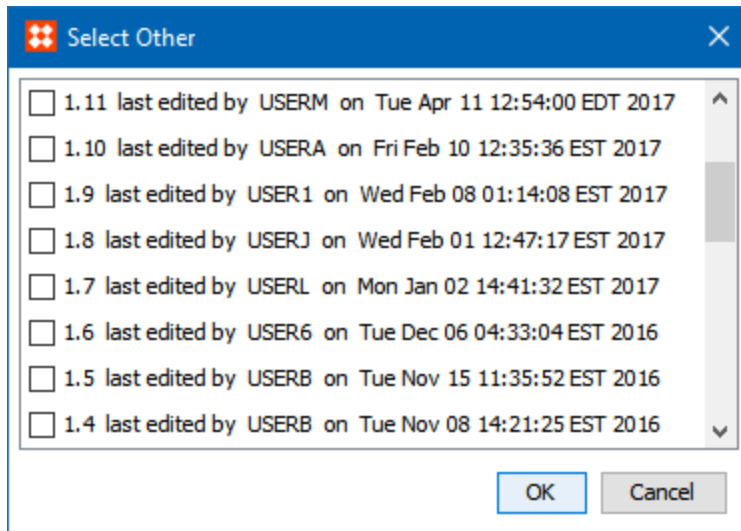
3. The eight most recent revisions are listed in a dropdown with check marks displaying beside the two most recent. To view additional revisions, click any unchecked revision that should also display. That revision will display to the right of revisions already displaying. Only one revision can be added at a time.



4. When more revisions are added than can be displayed on the screen, a scrollbar will appear allowing users access to all revisions via a horizontal scroll.
5. Revisions that are older than the eight available in the 'Target' dropdown can also be selected for comparison display by clicking 'Other' in the 'Target' dropdown.



When 'Other' is selected, a window displays listing all revisions for that object. To include any of these revisions in the STEP window's comparison view, click a revision listed in the 'Select Other' window (shown below). Multiple revisions can be added simultaneously using this method.



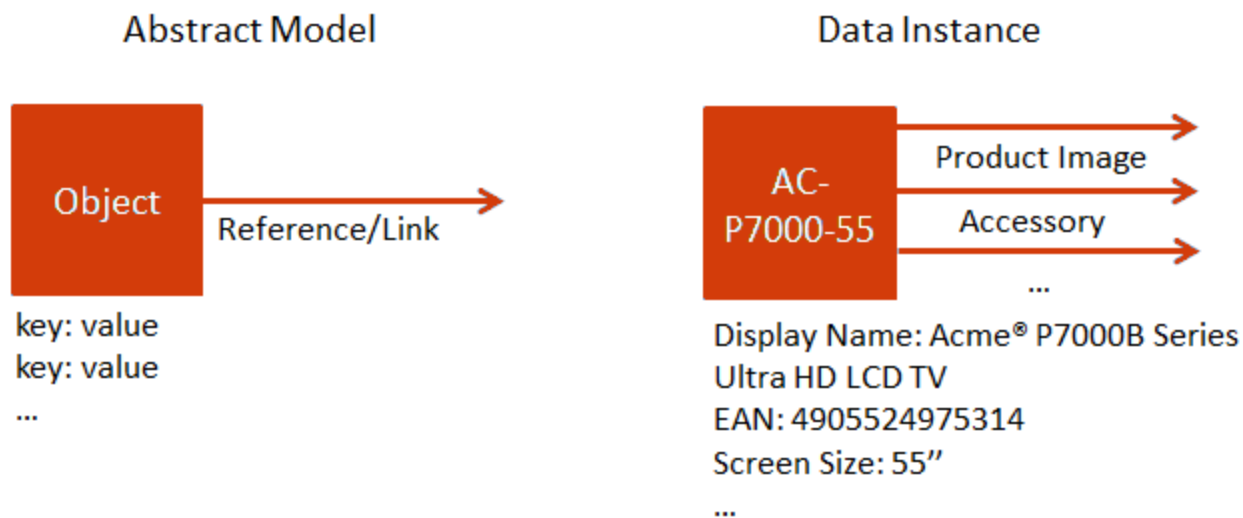
When all relevant revisions have been selected, they display together for a comparison view.

Compare Revisions		References	Referenced By										
View: Show all													
	>	1.16	>	1.15	>	1.13	>	1.12	>	1.11	>	1.10	>
> ID		100914		100914		100914		100914		100914		100914	
> Name		88723-12		88723-12		88723-12		88723-12		88723-12		88723-12	
> Object Type		Item Family		Item Family		Item Family		Item Family		Item Family		Item Family	
> Revision		1.16 Last e...		1.15 Last ...		1.13 Last ...		1.12 Last ...		1.11 Last ...		1.10 Last ...	
> Path		Primary Pro...		Primary Pr...		Primary Pr...		Primary Pr...		Primary Pr...		Primary Pr...	
> Approved		✔ Approv...		N/A		N/A		N/A		N/A		N/A	
> Translation		Master		Master		Master		Master		Master		Master	
> Category		Classification 1 root Suppliers Products Galore Products 88723-12		Classificatio n 1 root Suppliers Products Galore Products 88723-12		Classification 1 root Suppliers Products Galore Products 88723-12		Classification 1 root Suppliers Products Galore Products 88723-12		Classificatio n 1 root Suppliers Products Galore Products 88723-12		Classification 1 root Suppliers Products Galore Products 88723-12	
> Child Count		0		0		0		0		0		0	
> Discontinued													
> EAN		0025896541239		0025896541239		0025896541239		0025896541239		0025896541239		0025896541239	
> Family-Level ...													
> Hazmat													

Object Types and Structures

In a traditional relational database model, with concepts such as tables, columns, and rows to form a normalized data model, STEP uses the concept of objects to achieve the same goal.

In a STEP system, all entered items are modeled as objects with properties; the properties being metadata like IDs, attributes with values, and references / links to other objects.



Object Types allow a user 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 a user to identify a required item more easily. This object model is used throughout STEP and applies to objects themselves, configurations, attribute definitions, and reference type definitions.

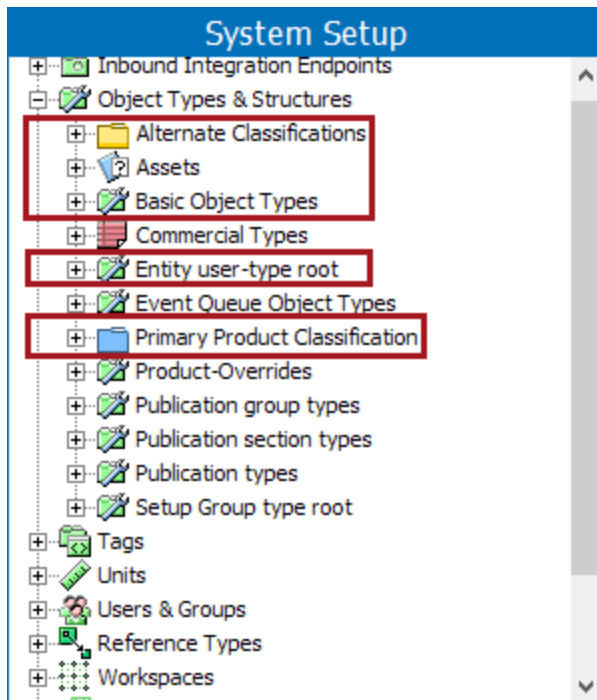
Object Type Facts in STEP

- All objects in a STEP system are instances of an 'Object Type' that constrains the behavior of the instances. For example, via the object type, a user decides which attributes and references should be legal for the instantiated objects.
- STEP comes with a number of predefined, but still is to some degree configurable, object types for modeling configuration objects like attributes. However, it is up to the system designer to determine data modeling, to define the required object types, and apply the desired behavior.
- The user defined object types that are listed below, are super types / node types used for modeling data.

Super Type / Node Type	Explanation	User-Defined Object Type Example	Instance Example
Products	Super type for modeling product data	• SKU • Sales Item • Supplier Item	AC-P70900-55 HD TV
Entities	Super type for modeling non-product data	• Supplier • Address • Customer • Location	• Joe Smith • World Trade Holdings
Assets	Super type for modeling digital data	• User Manual • Bill of Materials • Product Image • Situation Image	AC-P7000-ee User Manual
Calcification	Super type for modeling product and asset classification hierarchies	• Web Category • Asset Folder	Televisions

Object Type Location in STEP

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 section.

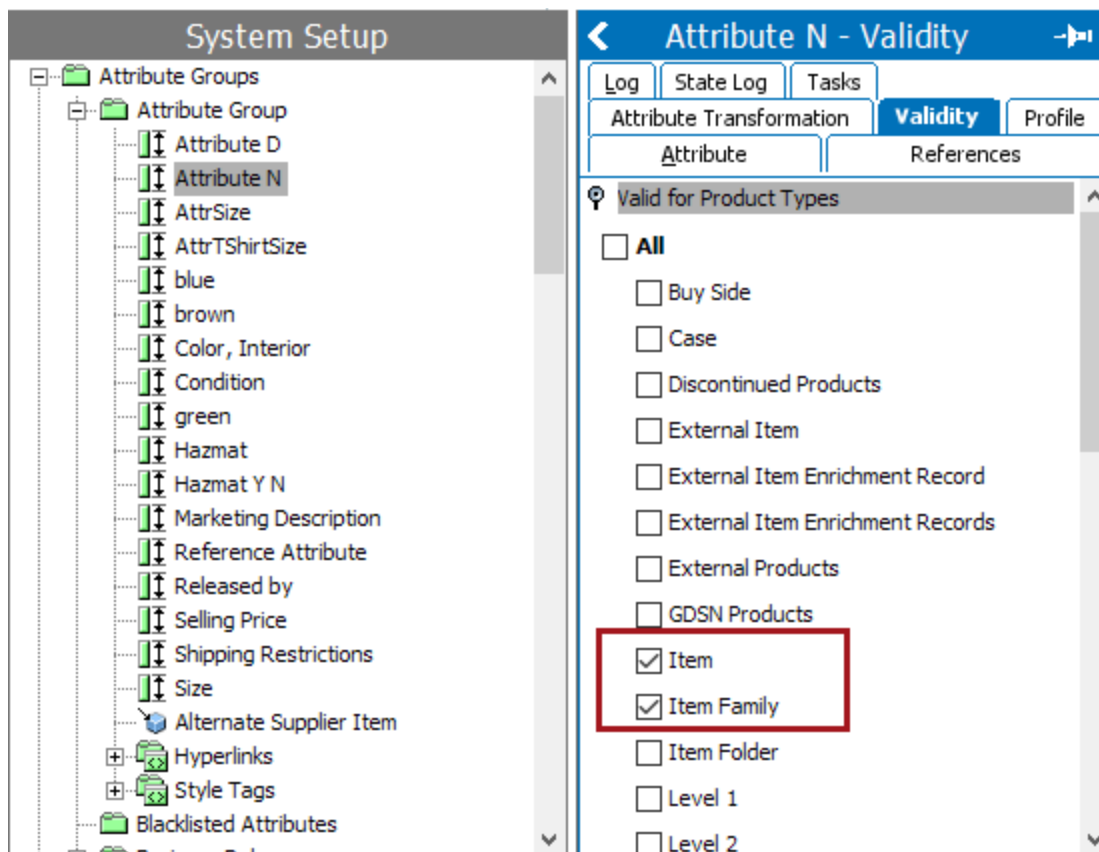
Attributes Valid for Object Types

An attribute is a characteristic or a detailed piece of information related to a particular object, or a relation between objects, such as references / links.

- Attributes can be made globally-valid for all objects of a specific object type. Description attributes can be applied to all object types, and will appear on all instances of the object type, as an editable field, no matter where in the hierarchy the object type is instantiated.
- A description attribute made valid for an object type will appear as a field in **Description**. Description attributes can be made valid for all user-defined object types, the majority of the predefined basic types and for user-defined reference and link types.
- Specification attributes can only be made valid for product object types and can thus only have values for product object type instances.

Setting Validity

To set the validity, go to the desired attribute > select the validity tab > choose required object types to give the validity as shown.



For more information, see the Description Attributes topic.

Autogenerated Using Name Pattern and ID Pattern

All data and configuration objects in STEP have an ID. The ID cannot be omitted, it cannot be changed, and since it serves to identify an object, it will always be the same across all 'Contexts' and 'Workspaces'.

ID's are however not unique identifiers in STEP, as they are not unique across super types or node types. Thus, you cannot have two products with the same ID, even when they are of different Object Types, but you can have a product, a classification and an attribute that all have the same ID.

Because IDs cannot be changed, care should be taken when assigning them, especially for objects that are likely to have their data and metadata changed routinely. Also it is advisable to avoid having spaces in the IDs.

In STEP it is possible to have IDs automatically generated for objects of specific object types and this feature should be used for:

- Objects that are created often.
- Objects that are created by end users
- Objects whose names / core properties are likely to change over time, and the name of the product doesn't hold much importance in identifying the product.

For example, an auto ID pattern is generally used when the products are created automatically, either by import or from an upstream system. This could be if there was a legacy system where the products were created and sent to STEP. To get the products created in STEP, an auto ID pattern should be used.

Defining the name 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.

To specify an ID / Name pattern, stand on the object type in System Setup / Object Types and Structure and enter the pattern in the field 'ID Pattern' or 'Name Pattern'. 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				
Translation			Not Translated	
Path			Primary Product Hierarchy/Products/Food and Beverage/Beverage/P	
Group Information			abc	
Marketing Description			fx	

Some important factors to keep in mind when using autogenerated IDs and names:

1. When an ID pattern has been specified for an object type, it is not possible to manually enter the ID. Objects with user defined IDs can however still be created via data imports. This should generally be avoided as mixing user defined and auto generated IDs typically will lead to errors.
2. The auto generated name would be a suggestion and can be changed.
3. There is no guarantee that the generated ID isn't already used by an existing object.
4. There can be gaps in the sequence

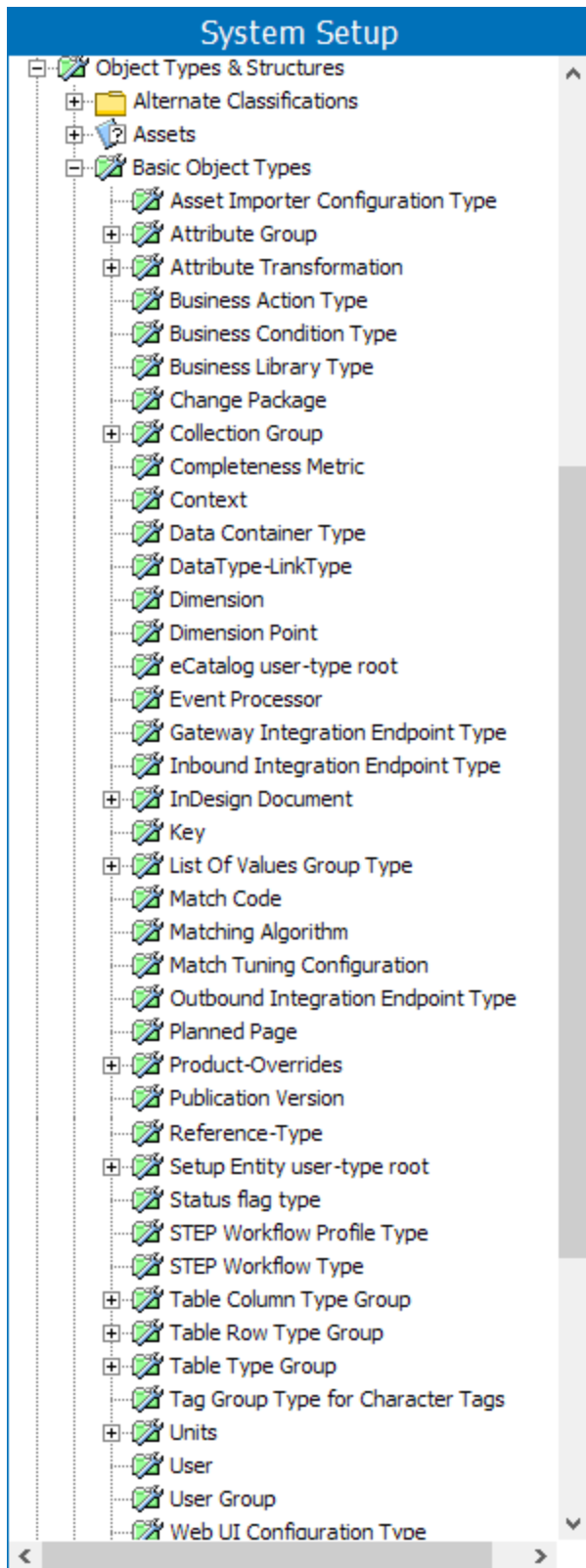
Basic Object Types

A basic object type is a node used for structuring the system data, for example, attributes, contexts, users, tables, etc. They are also called as predefined object types and structures.

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. Basic objects can be viewed under System Setup > Object Types and Structures > Basic Object Types:



Note: There is no delete access to any of the basic object types. User can define the automatic ID pattern and name pattern even for the basic objects.

Basic object types are used while creating the setup roots. For more information see the Setup Group topics in this documentation.

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. An example may be if you have the same product line in two country, but called different names.

To add a dependency on the object types, go to System Setup > Object Types and Structures > Object Type tab > Description flipper > Dimension Dependencies field.

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 **System Setup / Super User Guide** documentation.

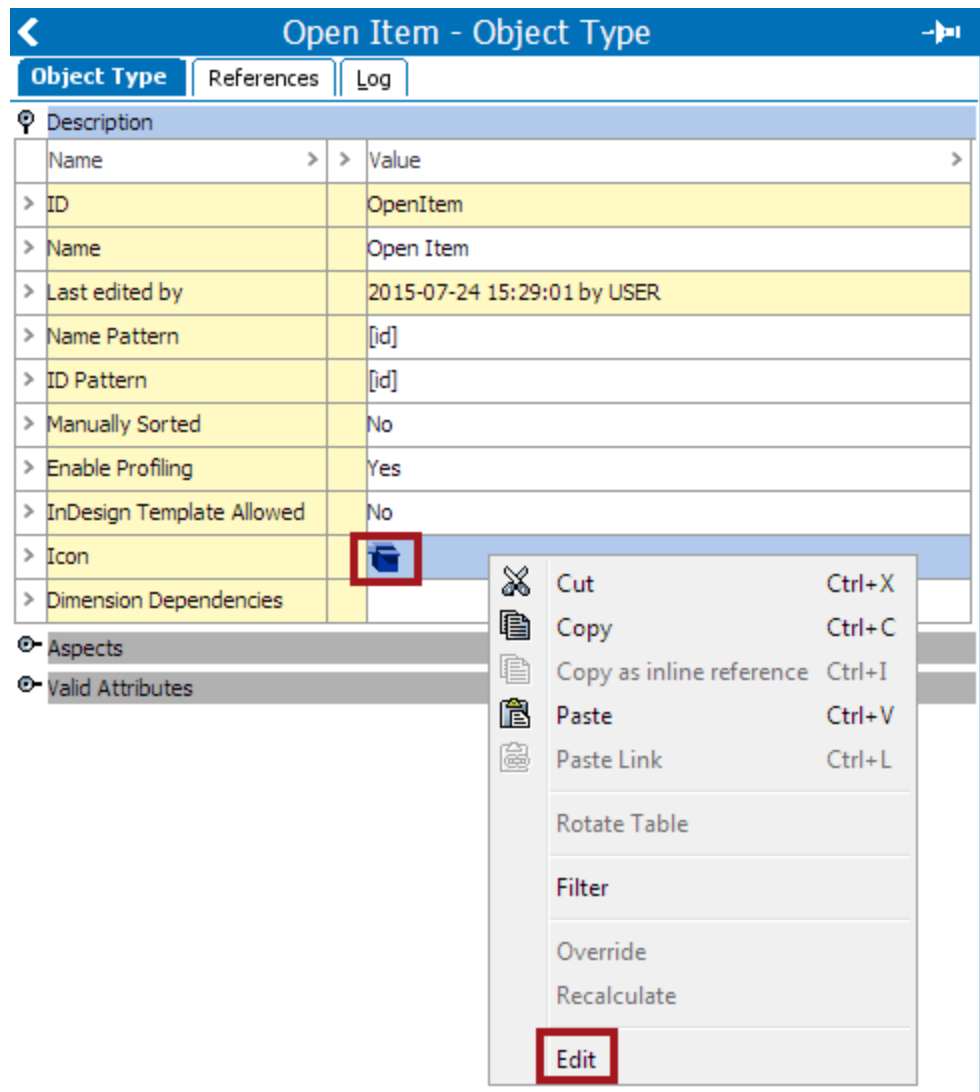
Object Type Icons

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.

Note: By default STEP has Icons for the products in the tree tab. However there is an option to change it.

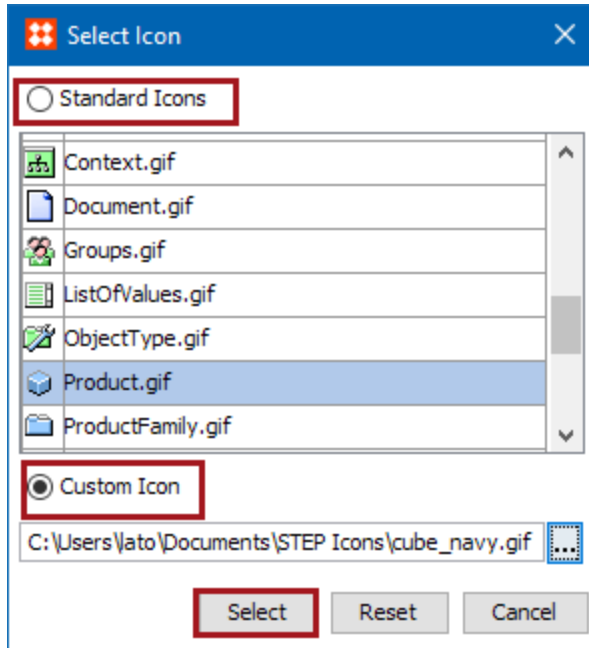
Editing an Object Type Icon

- 1. On the System Setup tab, select the Icon field of an Object, right-click and select the **Edit** option.

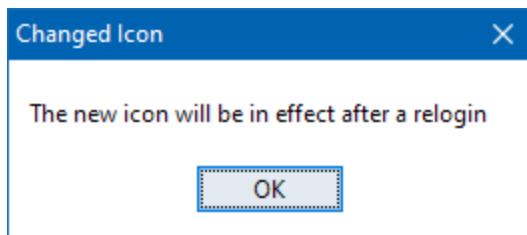


- Choose a standard or custom icon and click **Select**. If choosing a custom icon, select Custom Icon option, and click on the eclipses button. The image should be saved on the local machine to browse and select.

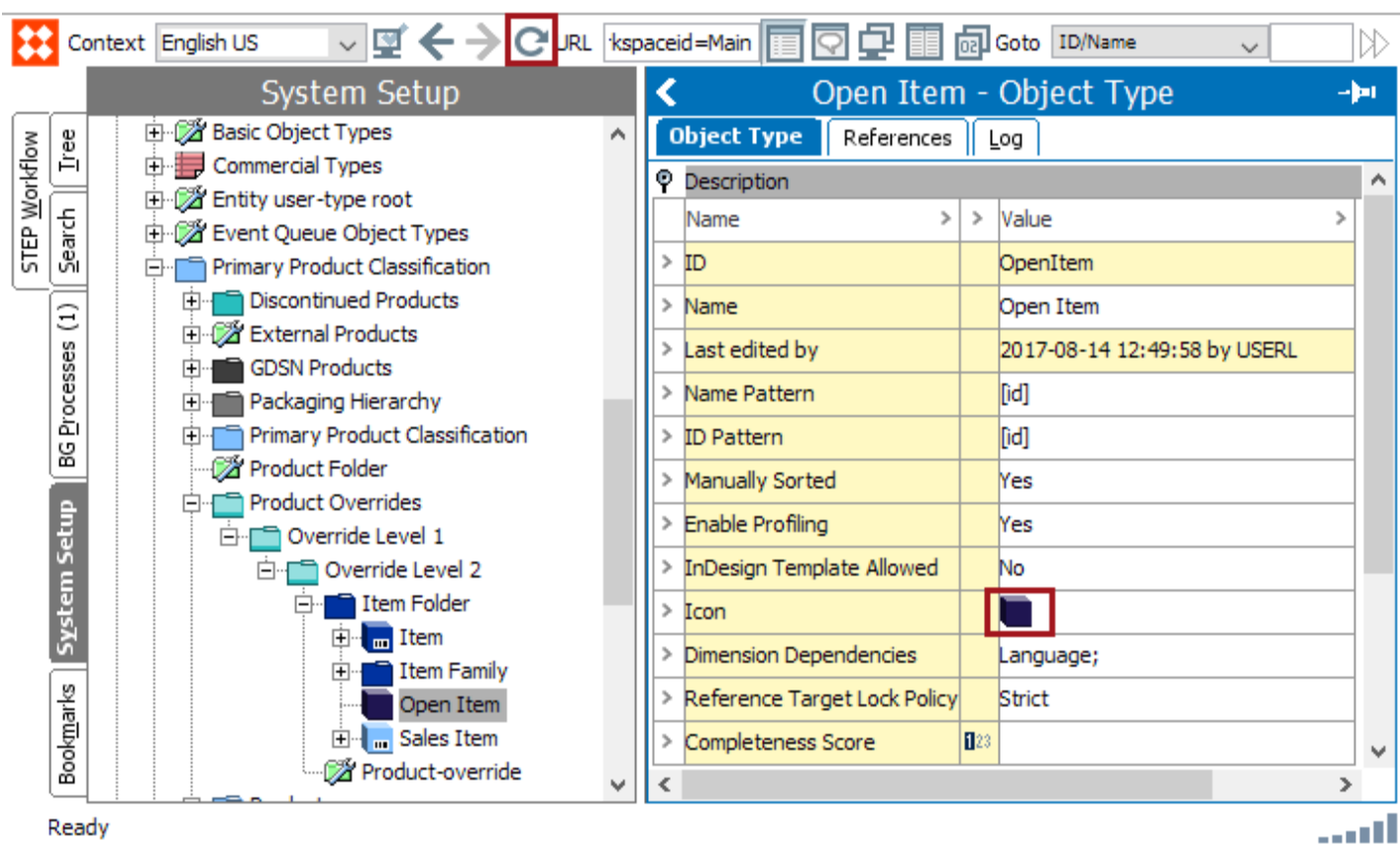
Note: The maximum icon size is 16 x 16 pixels. Sometimes when a user selects a bigger image as Icon, it appears as is on the products in the Tree nodes which will make the navigation difficult.



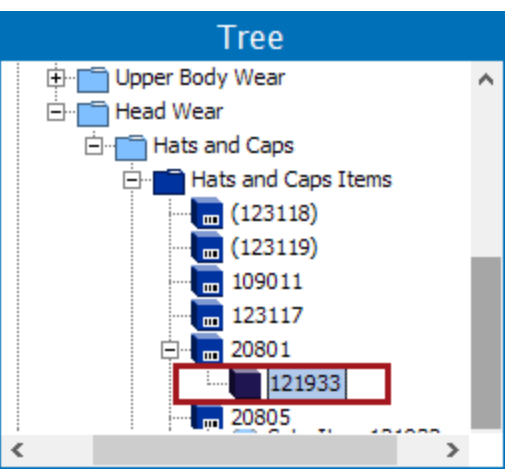
- Click **OK** on the Changed Icon confirmation dialog.



- On the Object Type, click the **Refresh** button to see the new icon rendered for the respective products in the Tree tab. Logging out is not necessary.



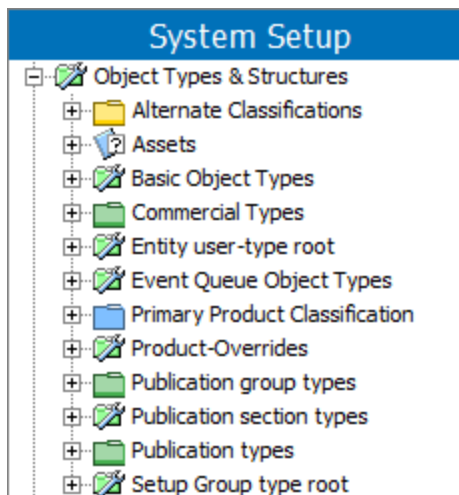
The icons will be reflected in the Tree tab on the products for the respective object type as shown below:



Note: Icons can be customized not only for products, but also classifications, assets, entities, publication etc.

Maintaining Object Types

- All nodes are of a specific Object type.
- An Object Type specifies the characteristics of a group of nodes in the database structure.
- Object Types can be created, deleted, linked and/or unlinked and edited.
- Object Types are maintained in System Setup under **Object Types & Structures**.



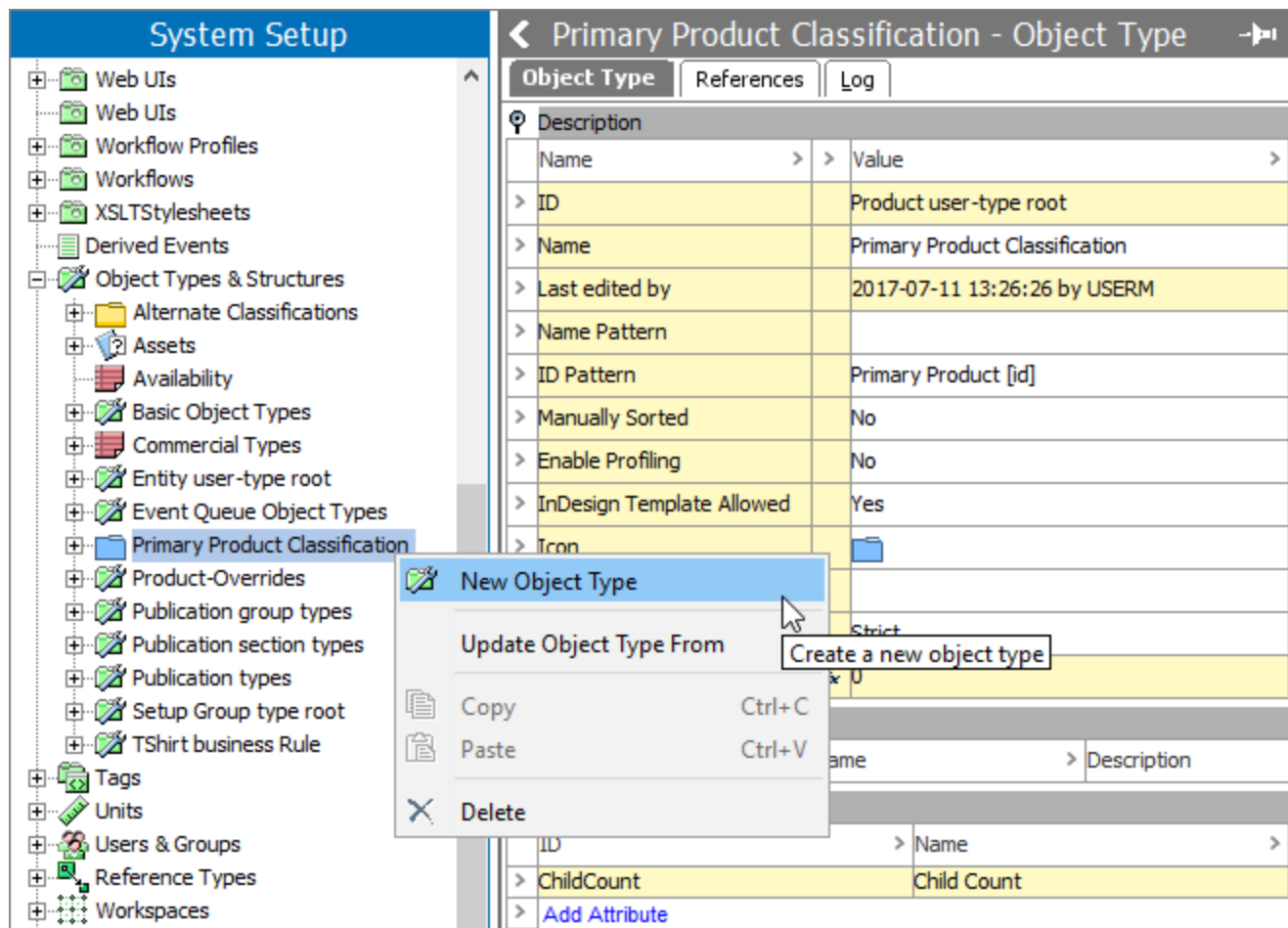
Creating an Object Type

STEP includes the following standard root object types that cannot be deleted: Assets, Classifications, Entities, Products. All new user defined object types must be defined below these.

New object types are created via the parent object type menu as shown in the steps below. As with all STEP objects, the new object types must be given an ID and a display name.

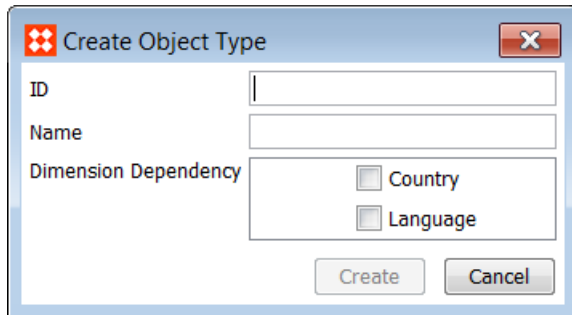
Steps to Create an Object Type

1. Go to **System Setup** > Expand **Object Types & Structures** > Select the relevant standard object type > Right-click the relevant standard object type > Click the **New Object Type** selection.



Note: Standard objects (provided automatically in STEP) can be easily identified in the workbench, because their ID will contain 'user-type root' (as shown in the example above).

2. The Create Object Type dialog will display.



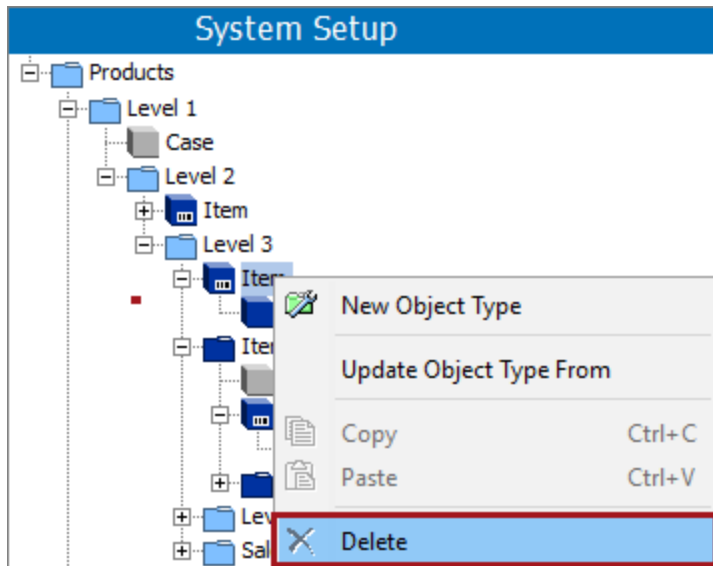
Note: If the Product root object type has an ID Pattern enabled, then the ID will display auto populated, and not editable. For more information see the **Autogenerate Using Name Pattern and ID Pattern** topic within this guide.

3. Enter an **ID** for the new object type. If ID and Name should be identical, press either the Enter or Tab keys to move the cursor to the **Name** parameter and automatically display the ID value. Otherwise, enter a name within the Name parameter.
4. If necessary, select one or more dimension dependencies. For more information, see the **Maintaining Dimension Dependent Object Types** topic.
5. Click the **Create** button, and the Object Type Editor will display with the newly created object type. The new object type will also be in the list of options when creating a new object of this type.
6. A desired hierarchy of objects can now be created using the new object type, and instantiating the objects in the Tree Tab. For more information, see the **Object Types and Structures** section of this guide.

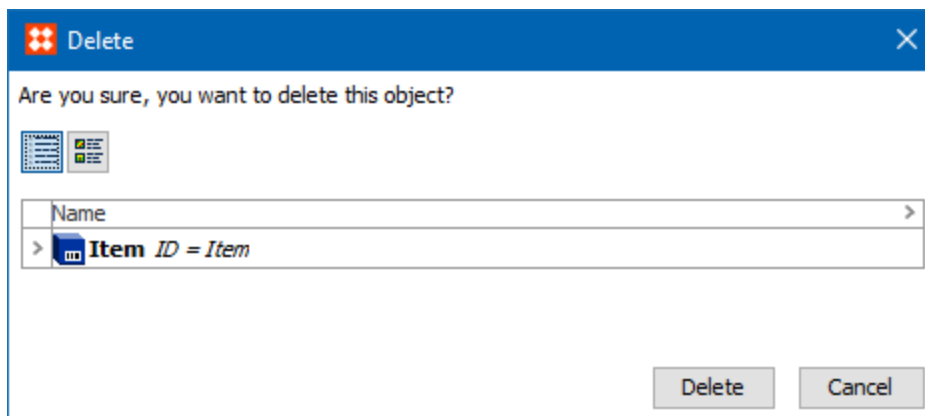
Deleting an Object Type

Below are the steps to delete an object type, that does not have children, in System Setup.

1. Go to **System Setup** > Expand **Object Types & Structures** > Select the **object type to delete** > Right-click the **object type to delete** > Select **Delete** from the menu.

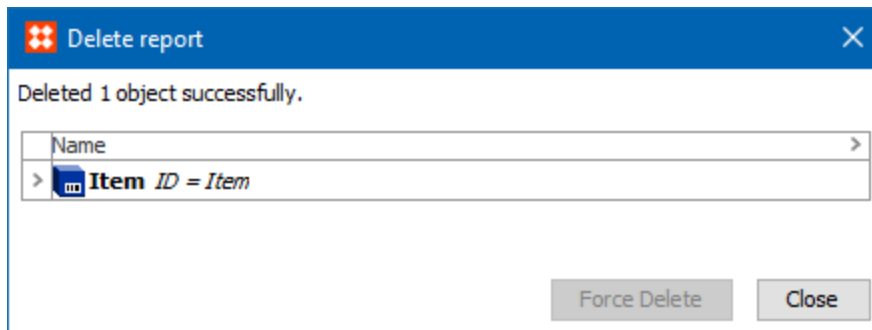


2. The Delete dialog will display asking you to confirm the deletion.

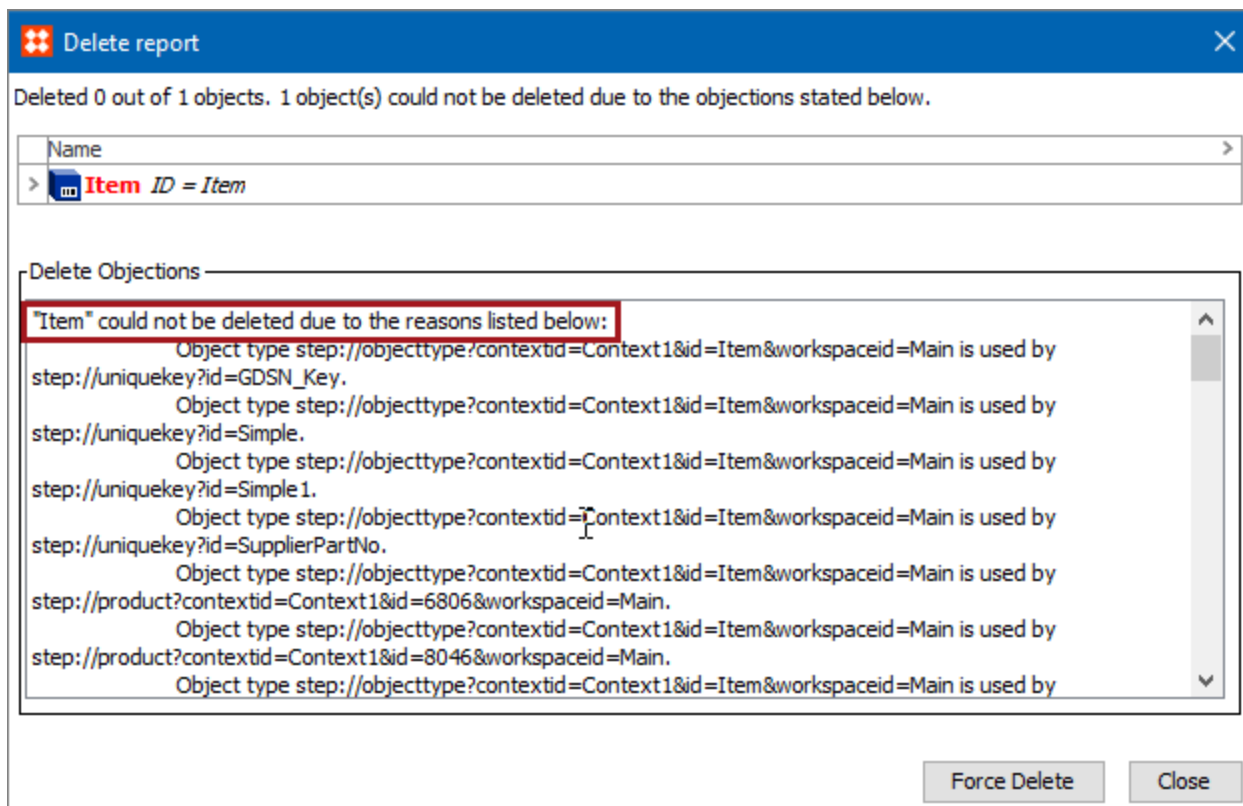


3. Click the **Delete** button, and the Delete report dialog will display.

If the object is successfully deleted, then the Delete report will display as shown below, and the object type is removed from the list of options displayed when creating a new object of that type.



If the object is not successfully deleted, then the Delete report will display with Delete Objections as shown below.



Clicking the **Force Delete** button will result in the Force delete report displaying. However, an object type that is used (instantiated in the tree), or has child object types, cannot be deleted.

Clicking the **Force Delete** button when the object type being deleted is used (instantiated in the tree) or has a child, then the Force delete report dialog will display as shown below.

Force delete report

Deleted 0 out of 1 objects. 1 object(s) could not be deleted due to the objections stated below.

Name

>  **Item** ID = Item

Delete Objections

"Primary Item" could not be deleted due to the reasons listed below:
Object type step://objecttype?contextid=Context1&id=Primary+Product+233614&workspaceid=Main is
used by step://product?contextid=Context1&id=PrimaryItem&workspaceid=Main.

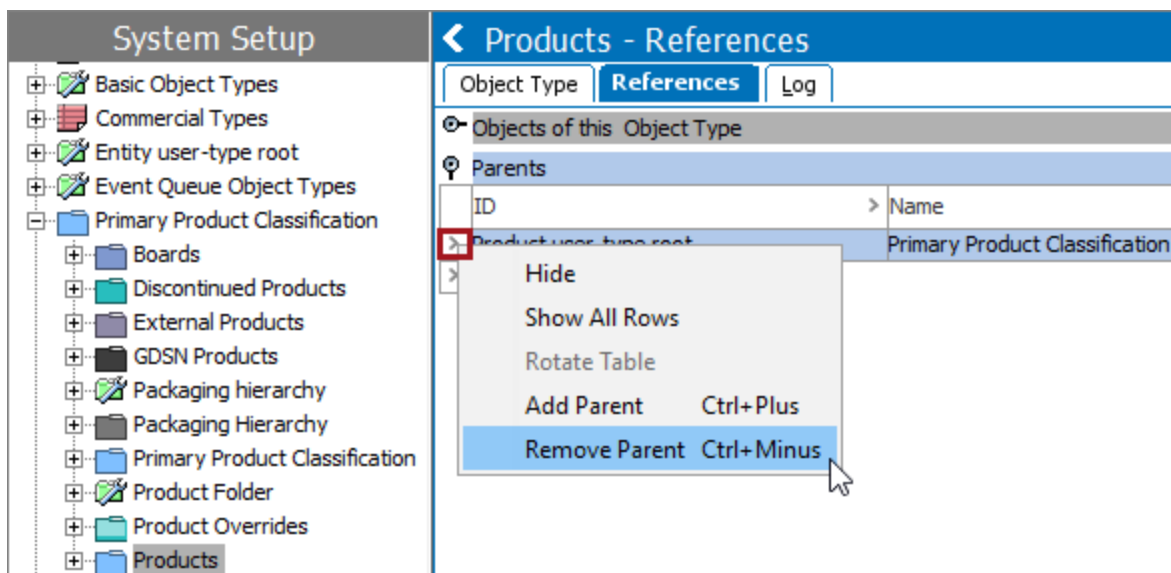
Close

Unlinking Object Types

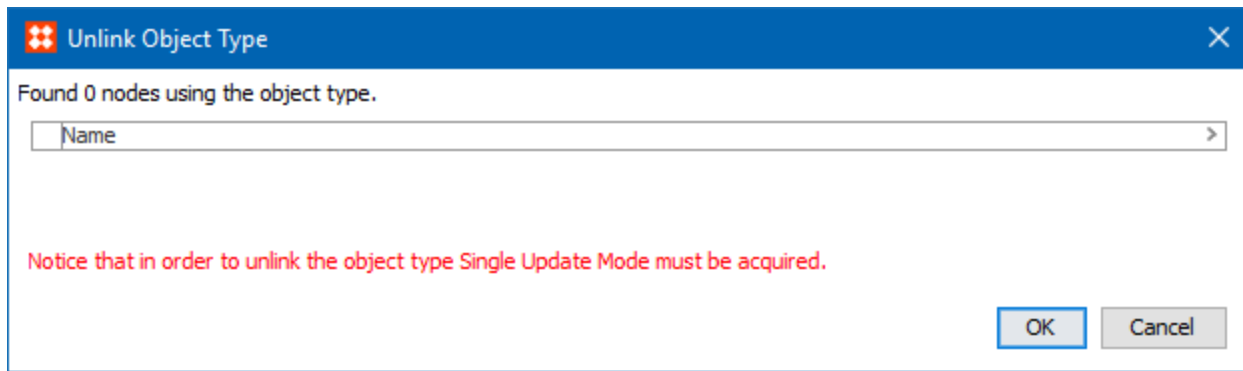
Unlinking an object type requires the object maintains at least one parent, all children must be removed from parents of this parent type in all Workspaces, and the system must enter Single-Update Mode. For more information, see the **Single-Update Mode** topic within the **System Setup / Super User Guide**.

Below are the steps to unlink an object type in System Setup.

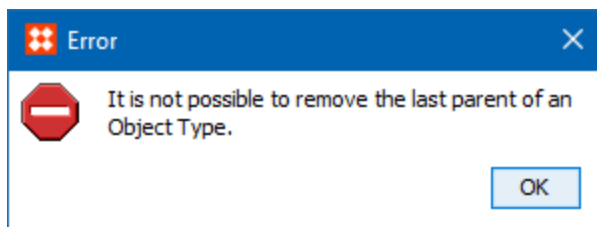
1. Go to **System Setup** > Expand **Object Types & Structures** > Select a relevant **object type** and the **Object Type Editor** will display.
2. Click the **References** tab.
3. Expand the **Parents** flipper, right-click the row '>,' and click the **Remove Parent** menu option (as shown below).



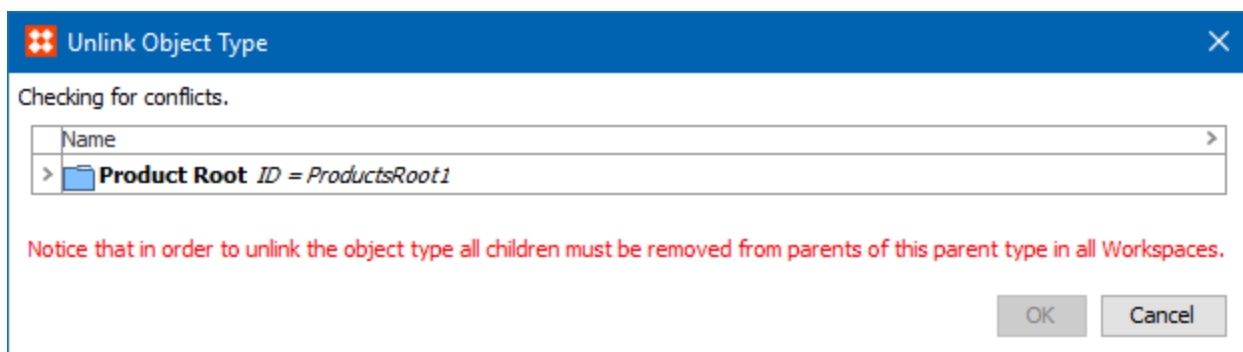
4. If more than one parent exists, and no children exist, then the Unlink Object Type dialog will display as shown below. Click the **OK** button and the Object Type Editor will display with the parent removed. Click the **Cancel** button to display the Object Type Editor with no changes.



If only one parent exists, then the Error dialog will display as shown below, because at least one parent must exist in the object type hierarchy (the last parent cannot be removed). Click the **OK** button to return to the Object Type Editor.



If children of the parent being removed are instantiated, the Unlink Object Type dialog will display with conflicts (as shown below). The instantiated children listed as conflicts must be moved to a valid parent (or deleted) before the parent can be unlinked. Click the **Cancel** button to return to the Object Type Editor. Once the conflicting children are removed, repeat the steps above to unlink the object type.



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.

Important: Existing settings on the object type are not removed / overwritten, but rather the original object type settings are added to the updated object types existing settings.

An example of this is when you have products that are obsolete, and you would like to save the obsolete product information for future reference, yet easily distinguish them from the active products. You could create a new obsolete products object type, save it in a different hierarchy, and update the obsolete products object type with the validation settings from the active products object type.

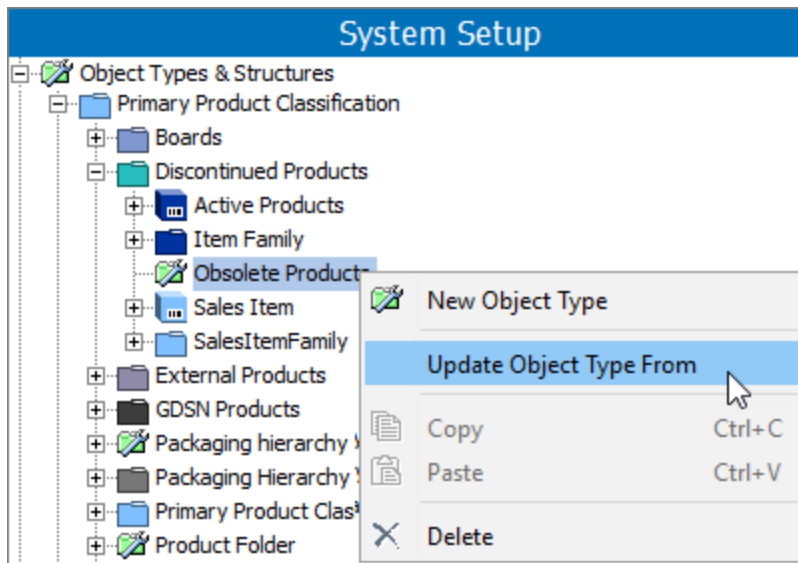
This concept could also be used for Onboarding products, so that there would be a temporary object type being assigned until the product is enriched and ready to be published, at which time its object type would be changed to the actual object type. This would defend against publishing incomplete data to downstream systems until the product is properly enriched.

Note: Dimension dependency will not be updated from the source object type, if needed then it should be done manually. For more information, see the **Maintaining Dimensions and Dimension Points** topic within the **Dimensions, Dimension Points, and Contexts** section of the **System Setup / Super User Guide**.

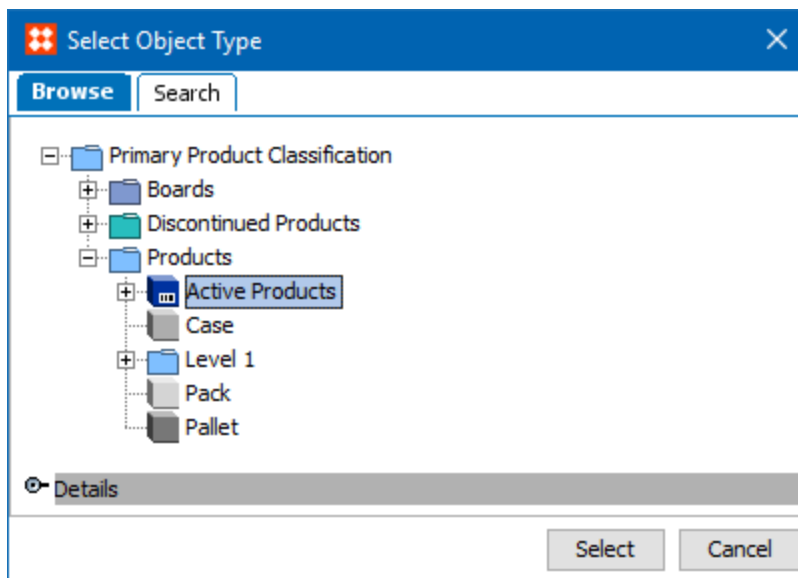
Steps to Update an Object Type from Another Object Type

Below are the steps to update an object types Attribute Validity, Reference Types Validity, Object Type Parent Links and Object Type Child Links from another object type.

1. Go to **System Setup** > Expand **Object Types and Structures** > Select the **relevant object type to be updated from another object type**. For this example, the 'Obsolete Products' object type is selected.
2. Right-Click the **relevant object type to be updated from another object type** > Click the **Update Object Type from** menu option.

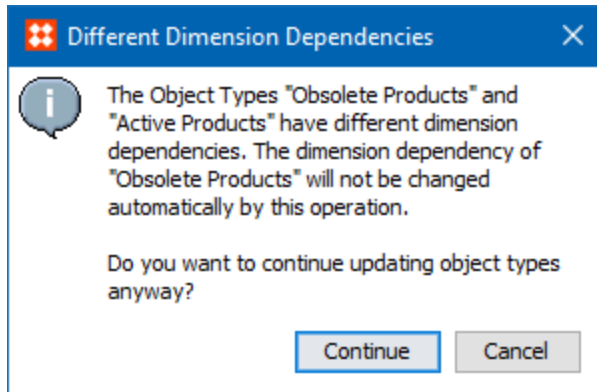


3. The Select Object Type dialog will display. Use the Browse or Search tabs to find the **object type with the validity settings that need to be applied**. For this example, the 'Active Products' object type is selected.



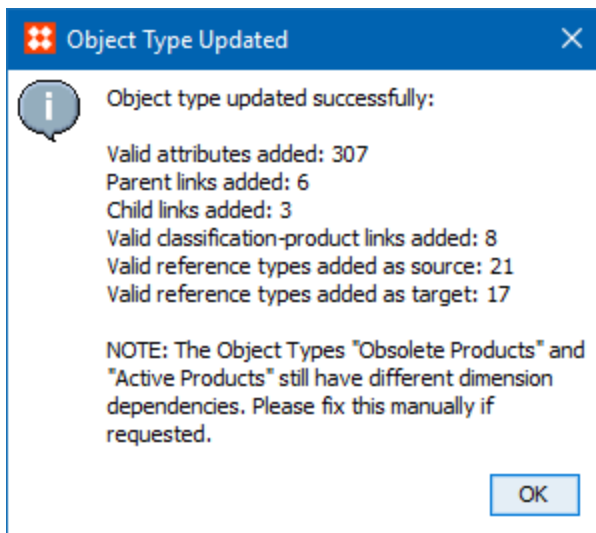
4. Click the **Select** button.

If the object types have different dimension dependencies then the Different Dimension Dependencies dialog will display as shown below. Click the **Continue** button to continue. Click the **Cancel** button to cancel and return to the workbench.

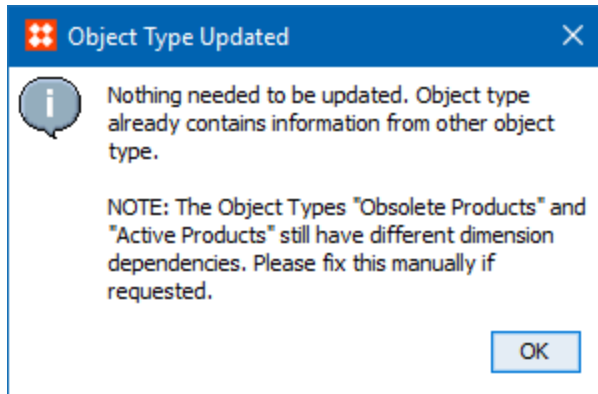


5. The Object Type Updated status dialog will display indicating how the selected object type was updated by listing the number of:

- Valid Attributes added
- Parent links added
- Child links added
- Valid classification-product links added
- Reference Types added as Source
- Reference Types added as Target



If the object types already match, then the Object Type Updated dialog will display with 'Nothing needed to be updated. Object type already contains information from other object type.' as shown below.



Once the object types is updated, optionally verify the updated object type results by comparing the Object Type and References tabs for each object type and/or look for the 'Updated from object type with ID...' text within the Log tab of the newly updated object type.

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.

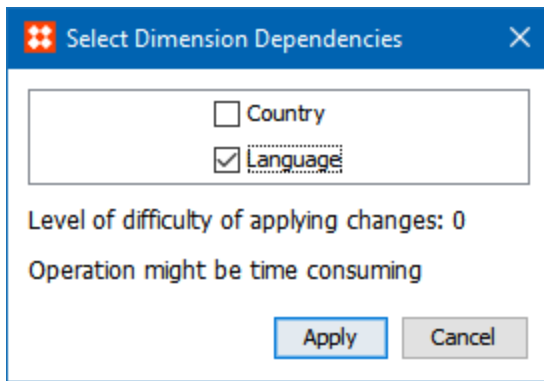
1. Go to System Setup, and navigate to object type that you would like to make dimension dependent.
2. On the **Object Type** tab, double-click in the **Dimension Dependencies** field, then click the ellipsis button (...).

The screenshot shows the 'System Setup' interface. On the left, the 'System Setup' tree is expanded to 'Object Types & Structures' > 'Basic Object Types' > 'Attribute', which is highlighted with a red circle and the number 1. On the right, the 'Attribute - Object Type' configuration window is open. It has tabs for 'Object Type', 'References', and 'Log'. The 'Object Type' tab is active, showing a table with the following data:

Description	
Name	Value
ID	stibo.normalattribute
Name	Attribute
Last edited by	2016-03-15 11:16:59 by USER4
Name Pattern	
ID Pattern	
Icon	
Dimension Dependencies	
Attribute Help Text	abc

The 'Dimension Dependencies' field is highlighted with a red box and a red circle with the number 2, indicating where to 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 the **Apply** button.



Select Dimension Dependencies

☐ Country

☒ Language

Level of difficulty of applying changes: 0

Operation might be time consuming

Apply **Cancel**

- 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

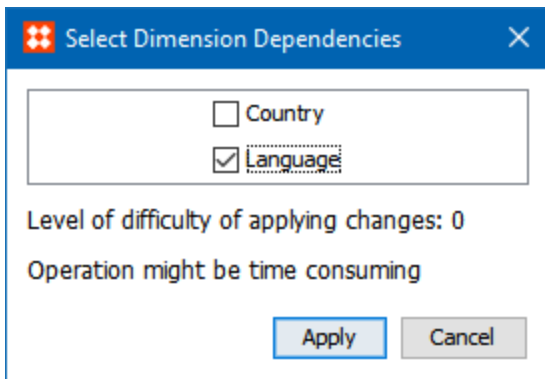
Dimension dependency is removed from an object in a near-identical fashion that it is applied.

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	

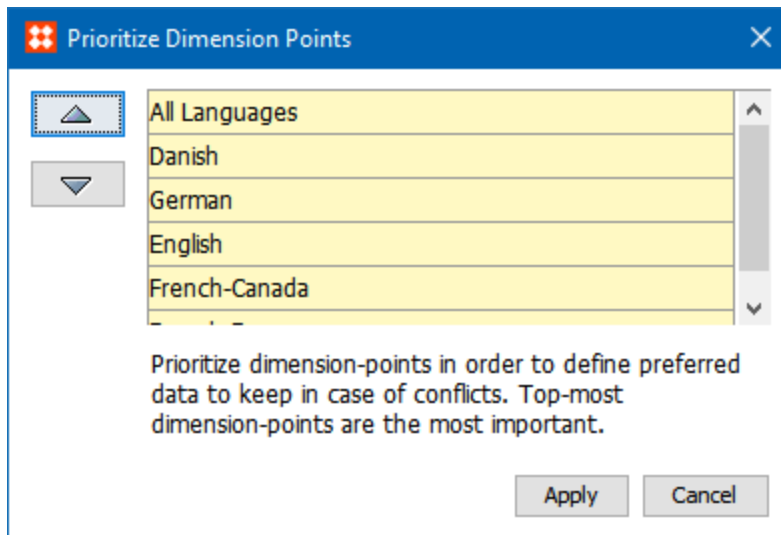
3. In the **Select Dimension Dependencies** dialog that displays, uncheck the box(es) for the relevant dimension (s), then click the **Apply** button.



The dialog box titled "Select Dimension Dependencies" contains a list of dimensions with checkboxes. "Country" is unchecked, and "Language" is checked. Below the list, it states "Level of difficulty of applying changes: 0" and "Operation might be time consuming". At the bottom are "Apply" and "Cancel" buttons.

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. Also, the system may enter Single-Update Mode, stalling other users from making updates / changes.

4. 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 the **Apply** button 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. However, STEP does not support all file formats, even if there is a MIME type available for it. Those most commonly used are listed in the table below.

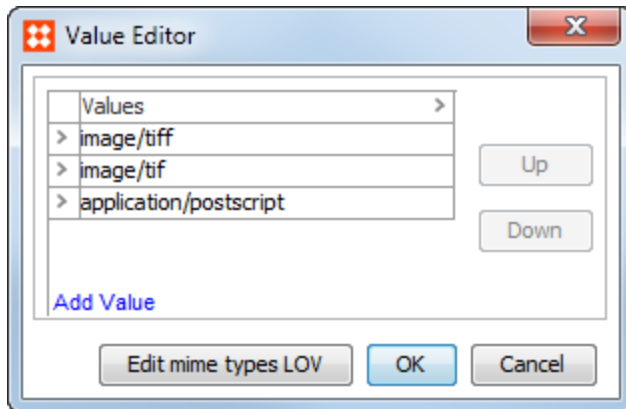
Note: STEP determines that a file is an 'image' if it has a MIME type of **image/***.

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

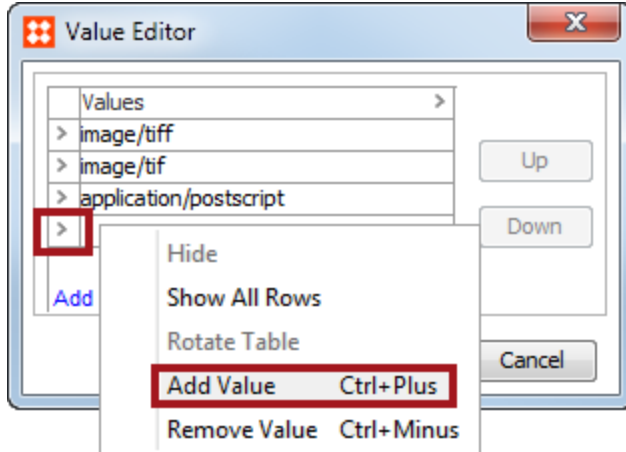
MIME Type	Definition
image/gif	GIF Image
image/jpeg	JPEG JFIF image
image/jpg	JPEG JFIF image
image/png	Portable Networks Graphic
image/tif	TIF Image
image/tiff	TIF Image
text/*	All Text Documents
text/html	Textual data in HTML format
text/rtf	Textual data in rich text format
Text/ plain;charset=ISO-8859-1	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

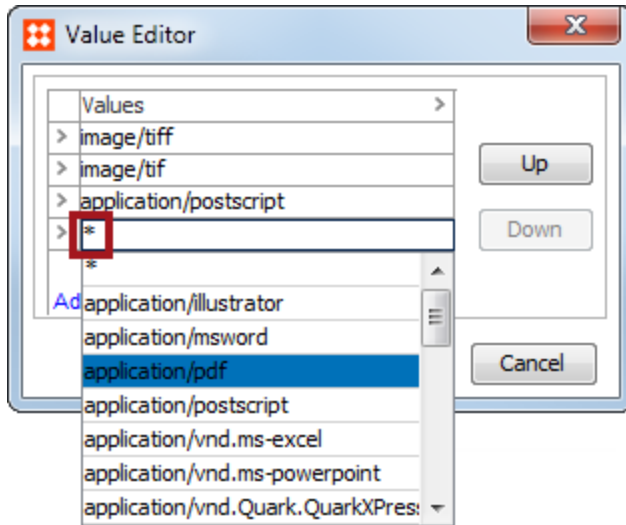
1. Go to **System Setup** > Select **Object Types & Structures** > **Assets** > select an **Image** or **Document Object Type**. (Mime Types are not valid for any other object types.) The corresponding editor will display.
2. Double-click in the **MIME Types** field to display the **Value Editor** dialog.



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**.

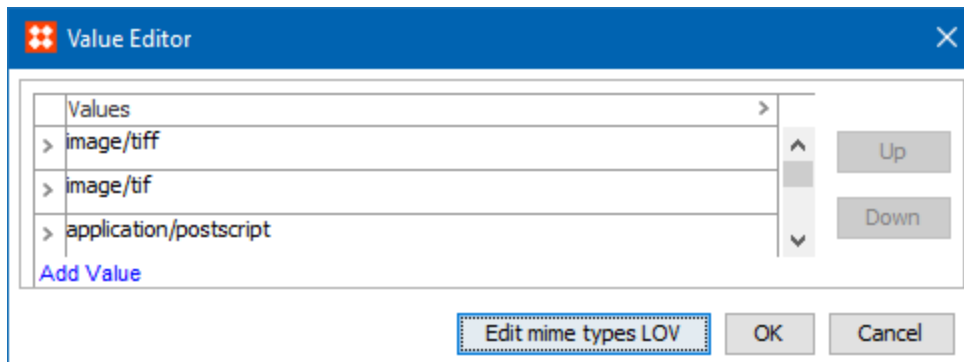


5. Repeat steps 3 and 4 to select more MIME Types.
6. Click **OK** to display the Object Type Editor assigned MIME Type(s).

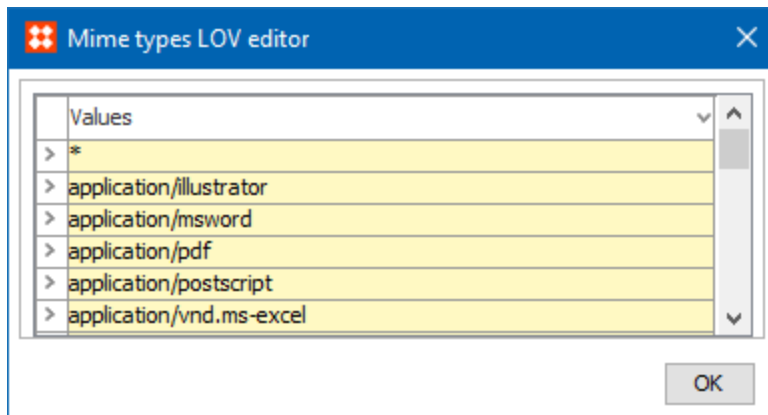
When importing an image or a document, it is automatically assigned the Object Type with the specific MIME Type.

Editing MIME Types LOV

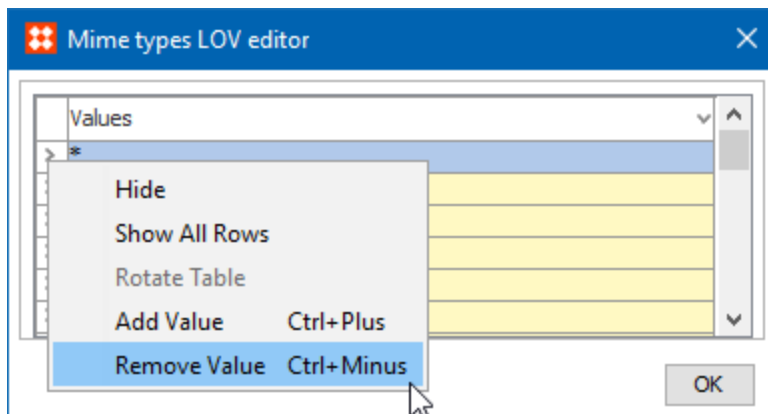
1. In Value Editor click on the 'Edit MIME types LOV' button.



2. The Mime types LOV editor dialog will display allowing the MIME list of values to be managed.



3. Navigate to an existing MIME type to be edited, and click the > button to display a menu as shown below.

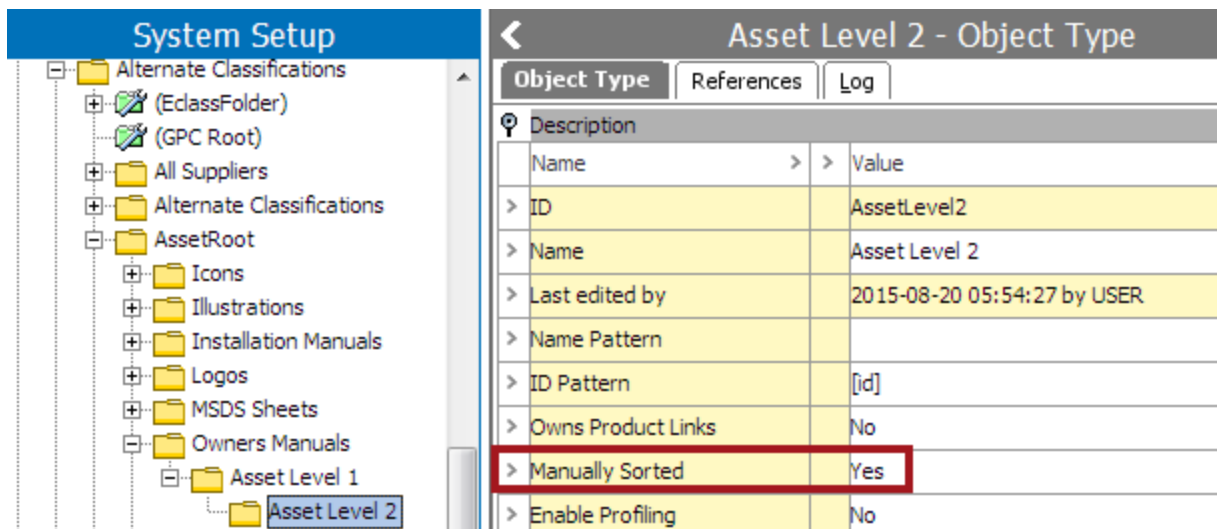


- Click Hide to no longer display a MIME type.
- Click Show All Rows to display all existing MIME types.
- Click Remove Value to delete the MIME type.

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. Go to **System Setup** > Select an **Alternate Classification, Primary Product Classification or Publication to be edited** > Select the **Object Type** tab on the editor and view the **Manually Sorted** parameter.

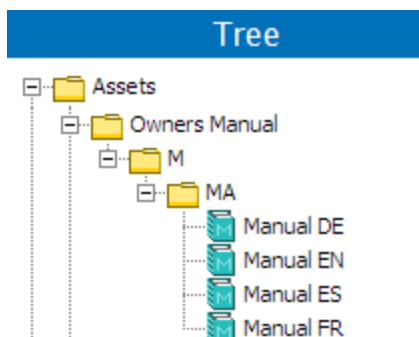


The screenshot shows the 'System Setup' window on the left with a tree view of folders including 'Alternate Classifications', 'All Suppliers', 'AssetRoot', 'Icons', 'Illustrations', 'Installation Manuals', 'Logos', 'MSDS Sheets', 'Owners Manuals', 'Asset Level 1', and 'Asset Level 2'. The 'Asset Level 2' folder is selected. On the right, the 'Asset Level 2 - Object Type' configuration window is open, showing the 'Object Type' tab. The 'Manually Sorted' field is highlighted with a red box, indicating its value is 'Yes'.

Asset Level 2 - Object Type	
Object Type	
Description	
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

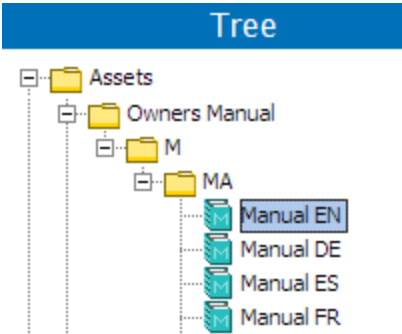
2. Set the **Manually Sorted** field:

- **No** means that child nodes of this object, that are holding this current object type, are ordered alphabetically in the Tree. Their sequence cannot be manually changed by a user.



The screenshot shows the 'Tree' view with a hierarchy of folders. The 'Assets' folder is expanded, showing 'Owners Manual', which is expanded to show 'M', which is expanded to show 'MA'. Under 'MA', there are four sub-folders: 'Manual DE', 'Manual EN', 'Manual ES', and 'Manual FR'.

- **Yes** means that child nodes of this object, that are holding this current object type, can have their sequence manually changed in the tree hierarchy. This can be done by dragging an object to another position.



Object Types

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.

The logs on all of the respective object types shows the changes made on them.

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

Object Type	Definition	Example
Entities	Defines the structure of the Entity hierarchy.	• Customers • Addresses
Publications	Defines the structure of the Publication hierarchy. Includes Publication Object Types, Publication Group Types, and Publication Sections.	• Printed Catalogs • Printed Sale Fliers
Basic Object Types	Defines structuring the system data Note that basic objects cannot be edited, or can new objects be added.	• Attributes • Contexts • Users

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.

System Setup

Object Types & Structures

(Availability)

Alternate Classifications

Assets

Basic Object Types

Commercial Types

Entity user-type root

Event Queue Object Types

Primary Product Classification

Product-Overrides

Setup Group type root

Change Packages

Status Flags

Uncategorized Setup Group Type

Workflow Profiles

Workflows

Business Action Type

Business Condition Type

Workflows

Workflows - References

Object Type

References

Log

Objects of this Object Type

Parents

ID	Name
Workflows	Workflows
Setup Group user-type root	Setup Group type root

Add Parent

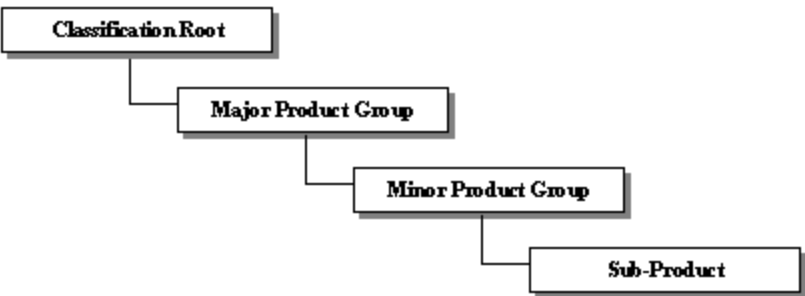
Classification Hierarchy Structures

The two hierarchy structures that may be defined are:

- Classification hierarchy
- Product hierarchy

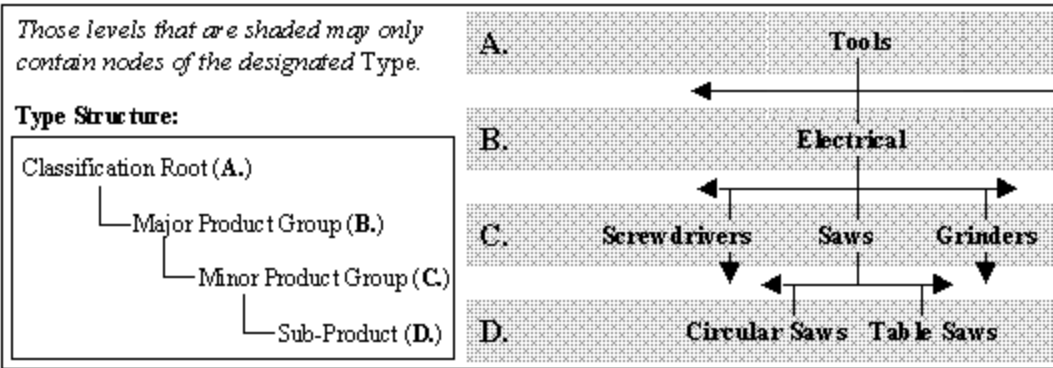
Note: There is no hierarchy that can be created for assets and entities.

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 type reside in the same level.

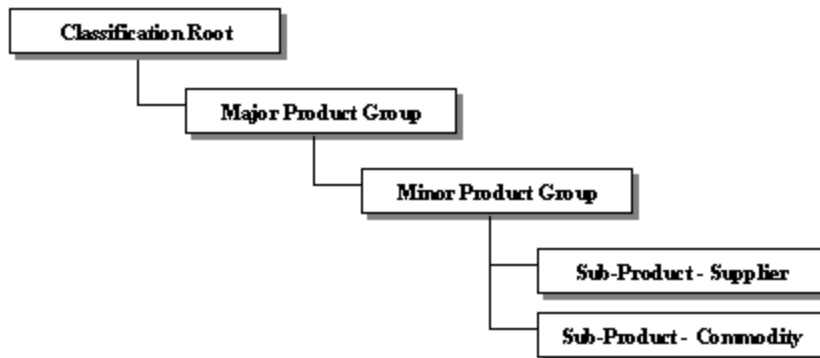


In this example, only a **Major Product Group** may be a child of a **Classification Root**, a **Minor Product Group** may only be a child of a **Major Product Group**, etc. As defined by this structure, it is not possible to create a **Sub-Product** under the **Classification Root**. Any nodes of the 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:



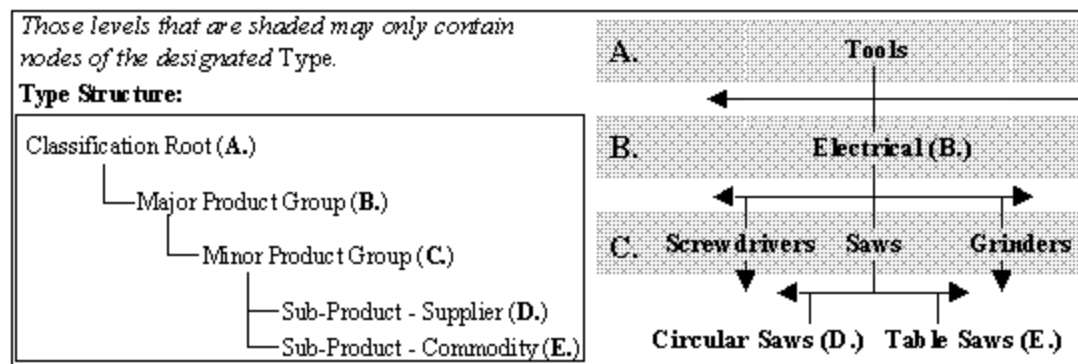
The first example illustrates a simple structure where each object type defines a level in the hierarchy. To build on this illustration, the second example shows it is also possible to create an object type structure in which a parent object type may have multiple children.



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 type 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 type. Therefore, object type do not necessarily represent 'levels' within a hierarchy. Also, both of these object type 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 type 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**

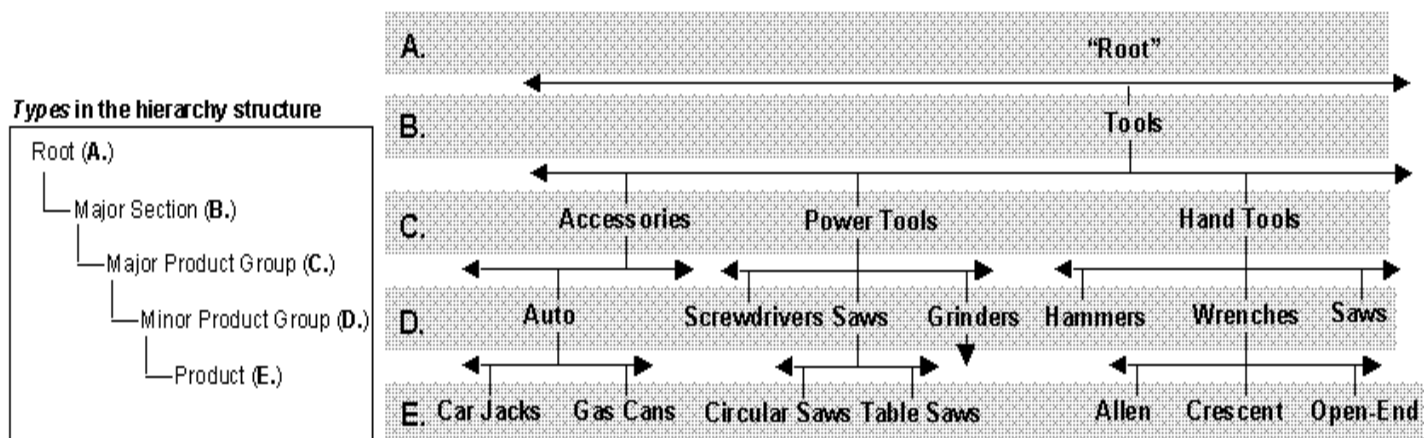
Fixed Hierarchy Structure

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'.

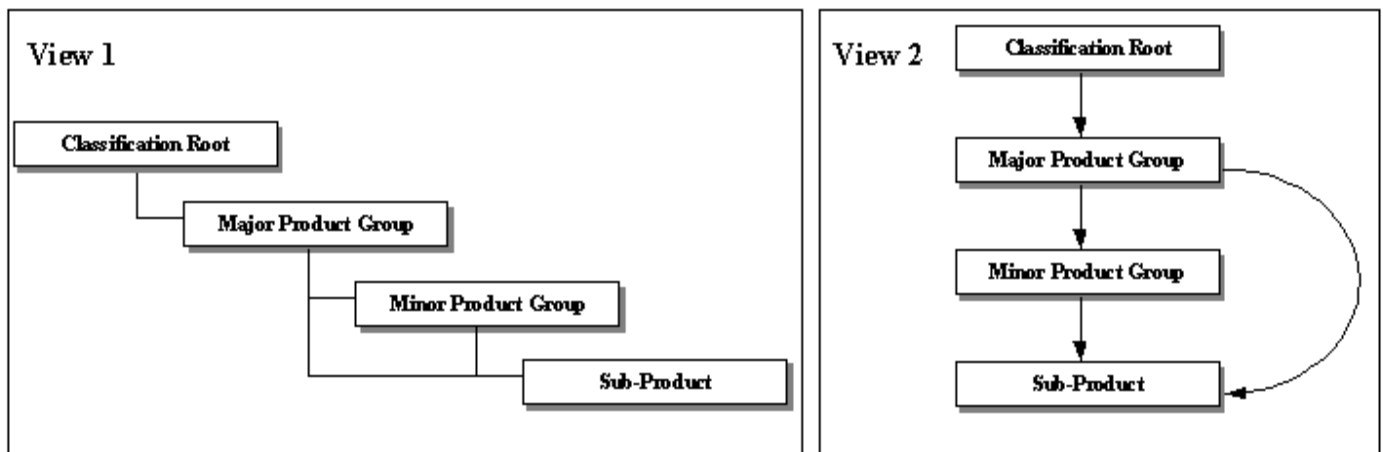


For information, see the Flexible Hierarchy Structure section.

Variation in Hierarchy Depths

In some instances, the depth of the hierarchy may vary. Not all hierarchies are suited to a formal type of structure discussed in the **Fixed Hierarchy Structure** topic in this documentation. 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.

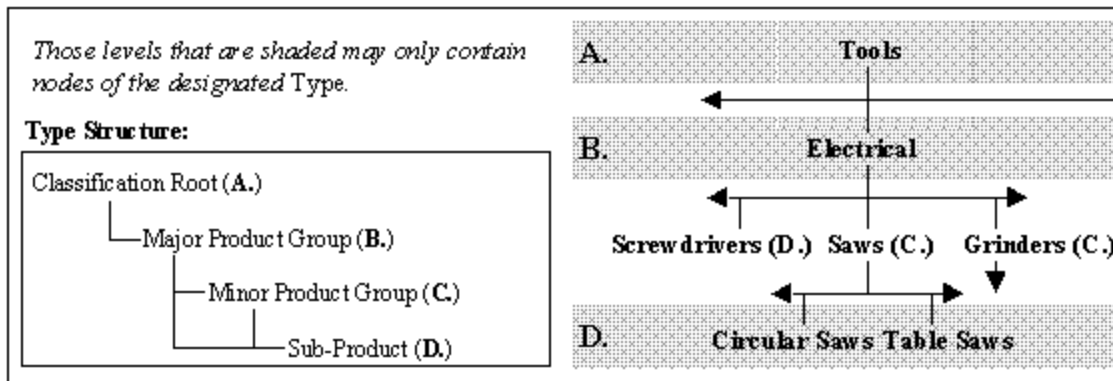
For example, the leaf nodes are of the same object type, but may be created in different levels. This is accomplished by allowing an object type to have multiple parents and displayed in the figure below.



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**.

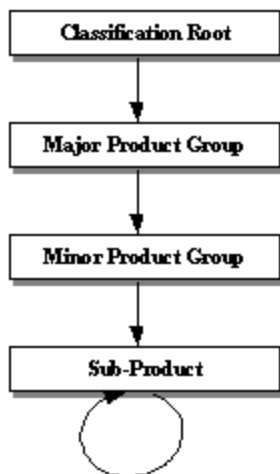
The following figure represents a hierarchy based on a varying object type structure.



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.

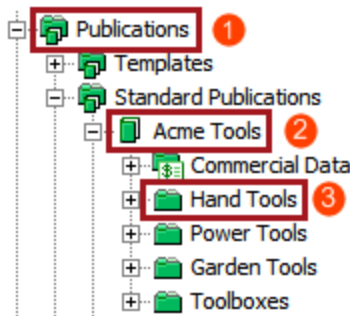


Creating a Standard Publication Object Type Hierarchy

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.

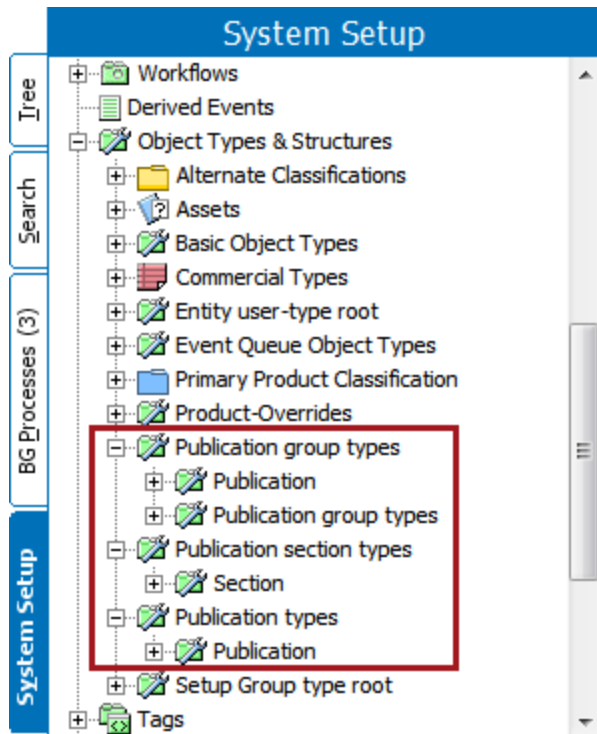
The following screenshot shows what a basic publication hierarchy looks like in the Tree. The publication object types addressed in this documentation are used to build the following objects:

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

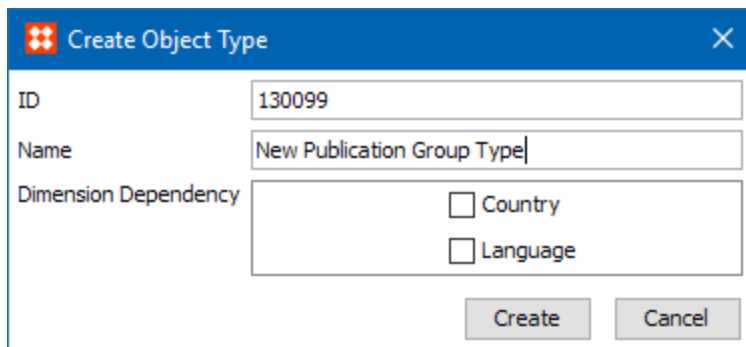
The following sections of this documentation will guide you through the process of creating new publication hierarchy object types as well as configuring valid parent / child relationships between these object types.

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

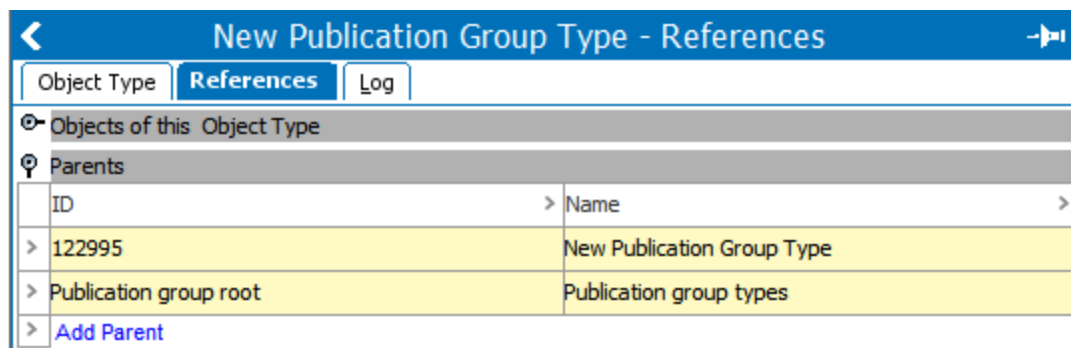
1. In System Setup, open the **Object Types & Structures** hierarchy.
2. Select **Publication group types**, right-click, and then select **New Object Type**. A **Create Object Type** dialog box appears.



The 'Create Object Type' dialog box is shown. It has a blue header with the Stibo logo and the title 'Create Object Type'. The fields are as follows:

ID	130099
Name	New Publication Group Type
Dimension Dependency	☐ Country ☐ Language
Create Cancel	

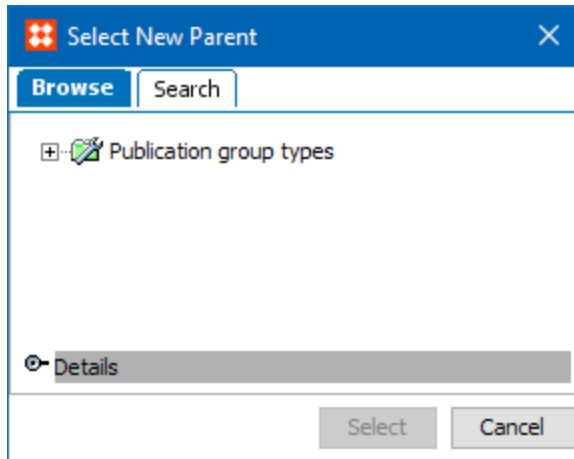
3. If the STEP ID for the **Publication group root** object has not been set to autogenerate with an ID Pattern, type an ID in the **ID** field.
4. In the **Name** field, type a name.
5. Check the relevant **Dimension Dependency** check boxes, if applicable. (Dimension dependencies are optional, and 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' editor is shown. It has a blue header with a back arrow, the title 'New Publication Group Type - References', and a right arrow. Below the header are three tabs: 'Object Type', 'References' (selected), and 'Log'. The main content area is divided into two sections: 'Objects of this Object Type' and 'Parents'. The 'Parents' section contains a table with the following data:

ID	Name
> 122995	New Publication Group Type
> 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 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.

Create Object Type

ID

Name

Dimension Dependency

☐ Country

☐ Language

Create

Cancel

- 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.

Publication - References

Object Type

References

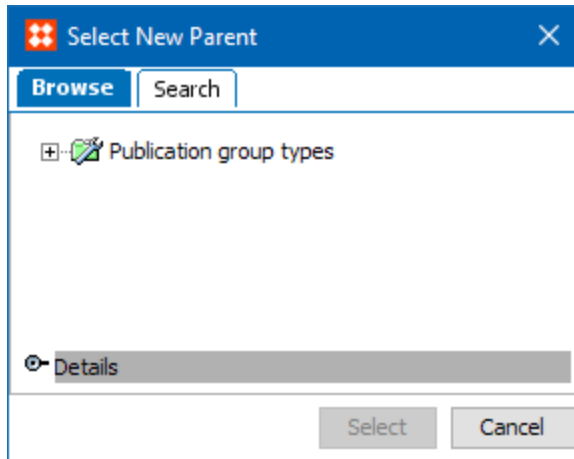
Log

Objects of this Object Type

Parents

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.

Create Object Type

ID

Name

Dimension Dependency

☐ Country

☐ Language

Create

Cancel

- 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.

<

Section - References

Object Type

References

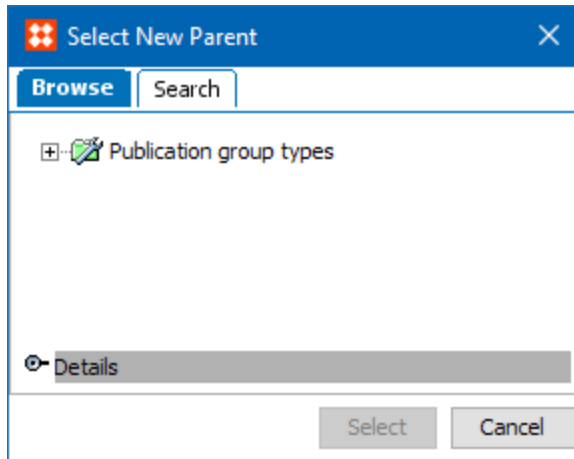
Log

Objects of this Object Type

Parents

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.

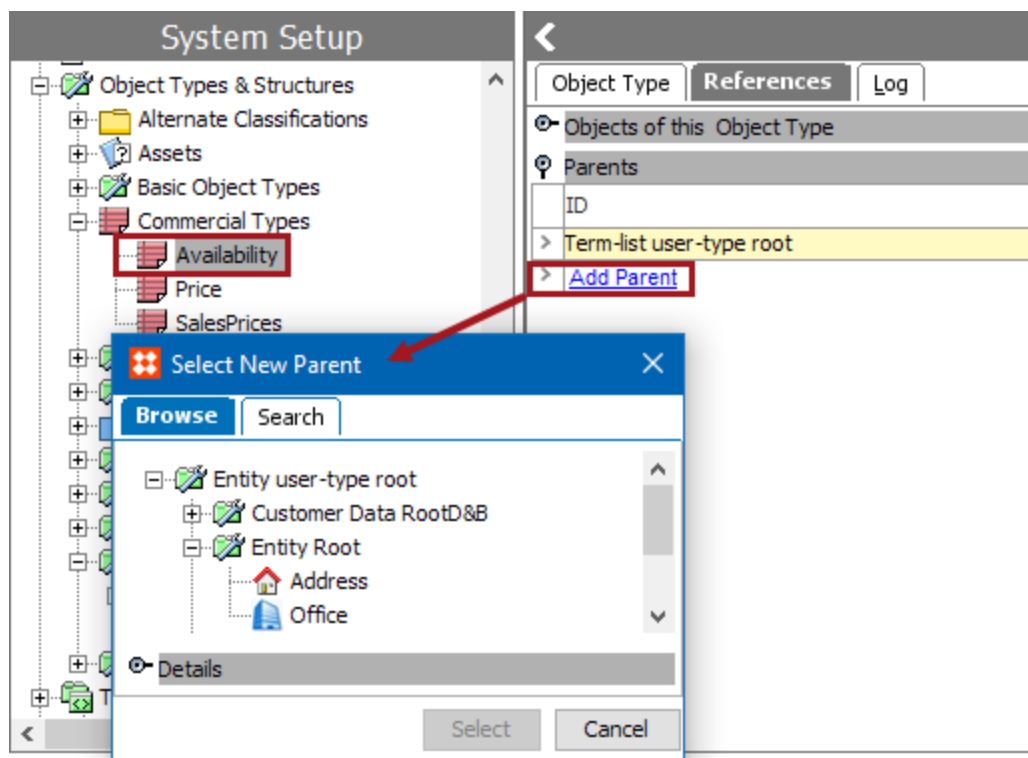
Product-Override Object Types and Commercial Object Types

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.)

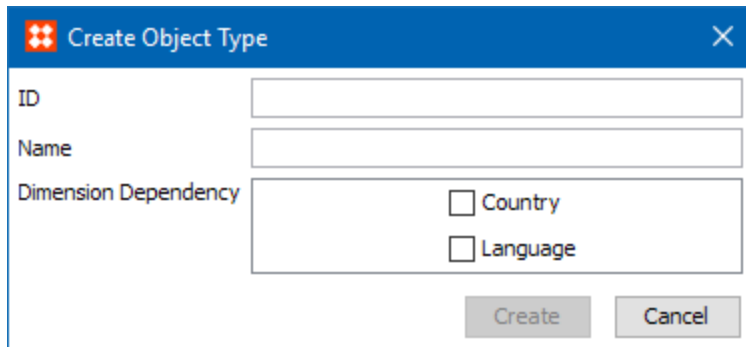
Link the Object Types from the References tab on either a Commercial Type or the References tab on a Product-Override by clicking the **Add Parent** link. A node selector displays a list of legal Object Types for linking.



Note: When Commercial Types must be used in Entity Hierarchies, manually link the Commercial Type into the Entity Hierarchy.

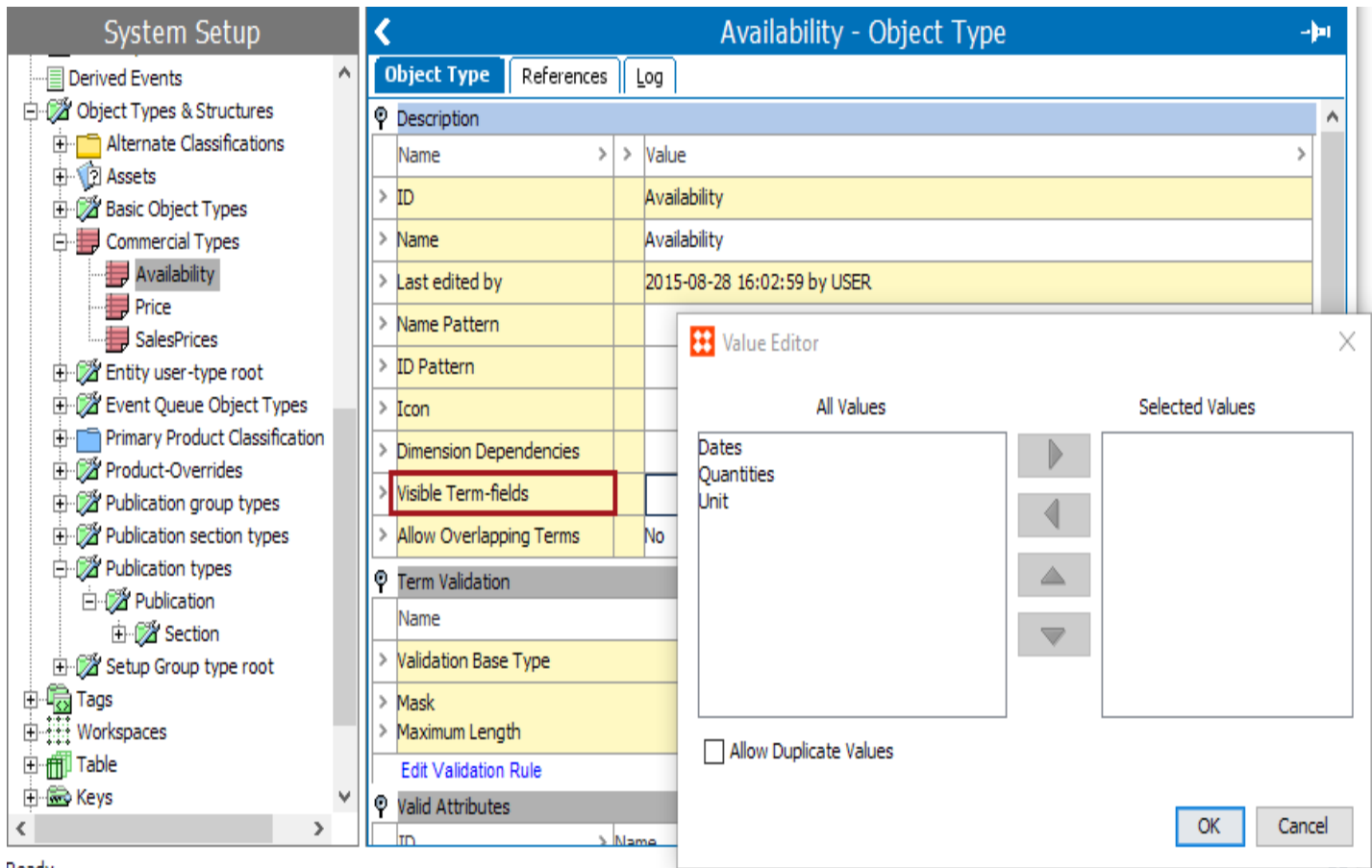
Creating a Commercial List Type

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 the 'Create Object Type' dialog box. It has a blue header bar with the Stibo logo on the left and a close button (X) on the right. Below the header, there are three input fields: 'ID', 'Name', and 'Dimension Dependency'. The 'Dimension Dependency' field is a larger box containing two checkboxes: 'Country' and 'Language'. At the bottom of the dialog, there are two buttons: 'Create' and 'Cancel'.

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. Select one or more values in the left hand side under the 'All Values,' and click on the selection button, , to move the values to the right hand side under 'Selected Values.'
7. Check the box for 'Allow duplicate Values,' if required and select 'OK' when finished.
8. Selected values will be populated in the **Visible Term-fields** value field.

System Setup

Derived Events

Object Types & Structures

Alternate Classifications

Assets

Basic Object Types

Commercial Types

Availability

Price

SalesPrices

Entity user-type root

Event Queue Object Type

Primary Product Classificat

Product-Overrides

Publication group types

Publication section types

Publication types

Availability - Object Type

Object TypeReferencesLog

Description

Name	Value
ID	Availability
Name	Availability
Last edited by	2015-08-28 16:02:59 by USER
Name Pattern	
ID Pattern	
Icon	
Dimension Dependencies	
Visible Term-fields	Dates Unit
Allow Overlapping Terms	No

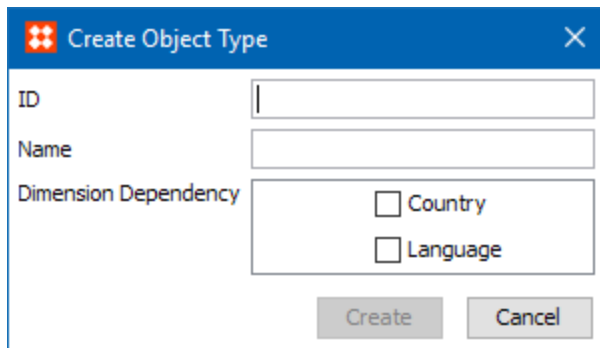
Term Validation

For more information, see the Commercial Data documentation.

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 'Create Object Type' dialog box. It has a blue header bar with the Stibo logo and the text 'Create Object Type'. Below the header, there are three input fields: 'ID', 'Name', and 'Dimension Dependency'. The 'Dimension Dependency' field has two checkboxes: 'Country' and 'Language'. At the bottom of the dialog are 'Create' and 'Cancel' buttons.

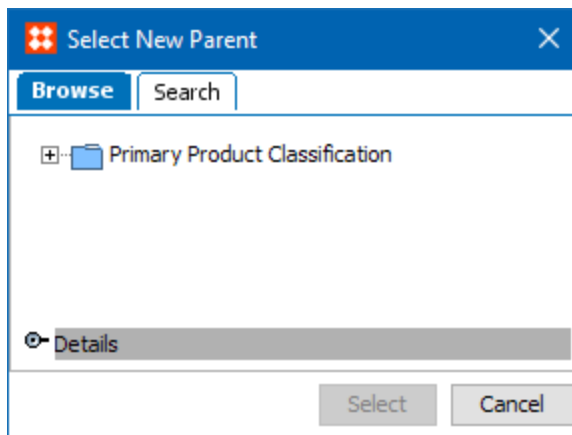
3. Type an **ID** and **Name** and optionally, check Dimension Dependencies.
4. Click **Create** and a new Product-Override Object Type is created.

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.'

Note: By default the Owns Product Links is set to 'No' whenever a new classification object type is created. You can instantly set the option to 'Yes' if the classification object type is not used as a target in any of the classification reference Links.

Owns Product Links Affects STEPXML

When Owns Product Links is set to 'Yes', Advanced STEPXML exports these references as both 'ProductReferences' under the Classification and as 'ClassificationReferences' under the Product.

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>
```

Consider the following example where a product is linked to a classification. Looking at the Object Type tab of this classification, OffersCategory, the 'Owns Product Links' is set to 'Yes'.

System Setup

- Object Types & Structures
 - Alternate Classifications
 - (GPC Root)
 - All Suppliers
 - Alternate Classifications
 - Asset Root
 - Configurations Root
 - E Class Folder
 - eClass 10 Root
 - ETIM 5 Article Groups Type
 - ETIM 6 Article Groups Type
 - Index
 - Merchandising Hierarchy Root
 - OffersCategory**
 - Web Hierarchy Root
- Assets
- Basic Object Types
- Commercial Types
- Entity user-type root
- Event Queue Object Types
- Primary Product Classification
- Product-Overrides
- Publication group types
- Publication section types

OffersCategory - Object Type

Object Type | References | Log

Description

Name	Value
ID	OffersCategory
Name	OffersCategory
Last edited by	2017-08-17 04:06:06 by STEPSYS
Name Pattern	
ID Pattern	
Owns Product Links	Yes
Manually Sorted	No
Enable Profiling	No
InDesign Template Allowed	No
Icon	
Dimension Dependencies	
Reference Target Lock Policy	Strict

Aspects

Component	Name	Description

Valid Attributes

ID	Name

[Add Attribute](#)

Product 'Polo A' is linked to classification 'Offers' through Classification Link 'OfferProducts.'

Tree

Primary Product Hierarchy

Products

Apparel

Upper Body Wear

T-shirts

T-shirts Items

Cotton T-shirts

InDesign

18210 M B

18212 L B

18213 M O

18216 L O

Polo T-shirt

Polo A

12-GGK799

12-GGK899

New Shirt

Polo A rev.0.2 - References

75% complete

Commercial

Tables

Category Profile

Proof View

Status

State Log

Tasks

Product

Sub Products

References

Referenced By

Images & Documents

Document References

Image References

Item References, Classification

Item References, Product

Packaging Hierarchy References

Sales Item References, Classification

Sales Item References, Product

Ungrouped Classification Links

Reference Type	Target
> Classifications	
> Object Creation ...	
> OfferProducts	Offers

Ungrouped Product References

Tree

Assets

Configurations

eClass 10

ETIM Hierarchy

Index Words

Merchandising Hierarchy

Offers

Polo A

Suppliers

Web Sites

Addresses

Customer Root

Customers

Entity Root

CCM

Offers rev.0.3 - Classification

Images & Documents

Tables

Status

State Log

Tasks

Classification

Sub Products

References

Referenced By

Description

Name	Value
> ID	Offers
> Name	Offers
> Object Type	OffersCategory
> Revision	0.3 Last edited by USERY on Thu Aug 17 04:31:33 EDT 2017
> Approved	Last Approved on Thu Aug 17 04:31:17 EDT 2017
> Translation	Not Translated
> Path	Classification 1 root/Offers
> Visibility	

To export an object like this via Advanced STEPXML, use a template similar to the one below. In this example 'Polo A' will be used:

© Stibo Systems - Confidential - STEP Trailblazer 8.2

666

Select Format

Advanced STEPXML

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

Template

```

<?xml version='1.0'?>
<STEP-ProductInformation>
  <Classifications>
    <Classification/>
  </Classifications>
  <Products>
    <Product/>
  </Products>
</STEP-ProductInformation>

```

Back

Next

Finish

Cancel

When the user has finished going through the wizard and pressed 'Finished,' part of the complete output will look like the following:

```

<Classifications>
  <Classification ID="Offers" UserTypeID="OffersCategory"
ParentID="Classification 1 root" Selected="false"
Referenced="true">
    <Name>Offers</Name>
    <ProductReference ProductID="113202" Type="OfferProducts"/>
  </Classification>
</Classifications>
<Products>
  <Product ID="113202" UserTypeID="Item" ParentID="113201">
    <Name>Polo A</Name>
    <ClassificationReference ClassificationID="Offers"
Type="OfferProducts"/>
  </Product>
</Products>

```

If the export of 'Polo A' via Advanced STEPXML with a template is complete, but with the recorder option 'IncludeClassificationOwned' included in Product element the output will look similar to the following:

Select Format

Advanced STEPXML

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

Template

```
<?xml version='1.0'?>
<STEP-ProductInformation>
  <Classifications>
    <Classification/>
  </Classifications>
  <Products IncludedClassificationOwned="false">
    <Product/>
  </Products>
</STEP-ProductInformation>
```

Back

Next

Finish

Cancel

When the user has finished going through the wizard and pressed 'Finished,' part of the complete output will look like the following:

```
<Classifications>
  <Classification ID="Offers" UserTypeID="OffersCategory"
ParentID="Classification 1 root" Selected="false"
Referenced="true">
    <Name>Offers</Name>
    <ProductReference ProductID="113202"
Type="OfferProducts"/>
  </Classification>
</Classifications>
<Products>
  <Product ID="113202" UserTypeID="Item" ParentID="113201">
    <Name>Polo A</Name>
```


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** topic within this guide.

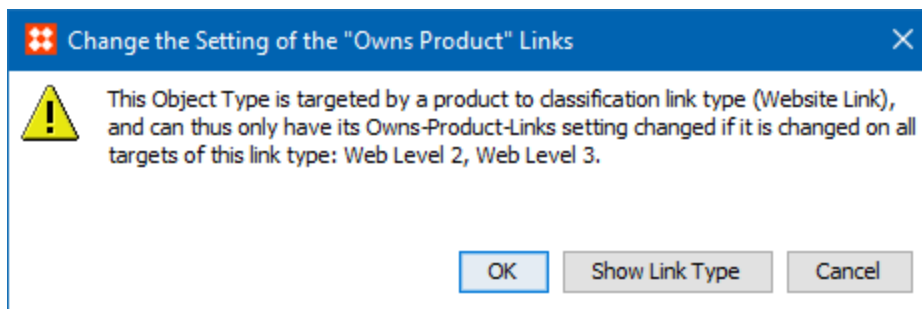
Note: Changing the setting of Owns Product Links can be a time-consuming operation.

Steps to Change the Owns Product Links

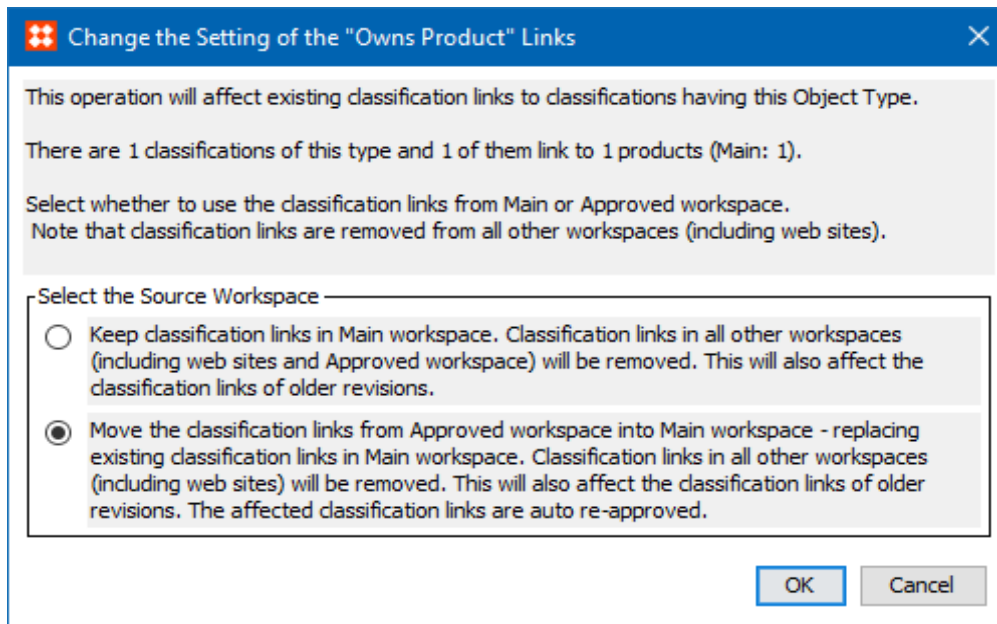
Changing the Ownership for an existing Classification Object Type which is being used as Target in various Classification Link Types can be done by following the steps below:

1. Go to **System Setup** > Expand **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.

If the object type is targeted by a product to classification link type then the dialog will display as shown in the screenshot below.



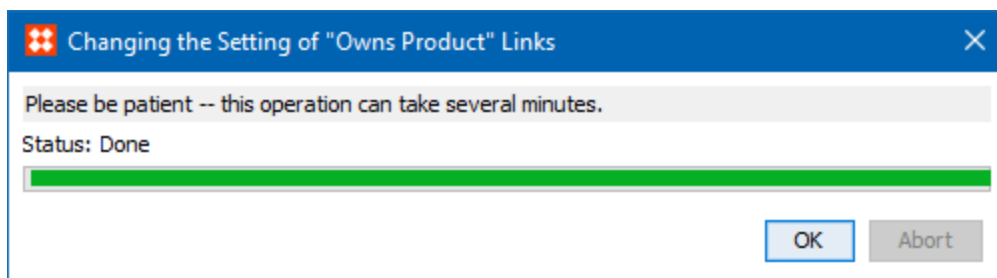
- Clicking the **Show Link Type** button will close the dialog and display the specified link type editor.
- Clicking the **Cancel** button will close the dialog and display the classification object type editor.
- Clicking the **OK** button will close the dialog and display the dialog as shown below.



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 the **OK** button to start the process and display the Changing the Setting of "Owns Product" Links status dialog, as shown below.



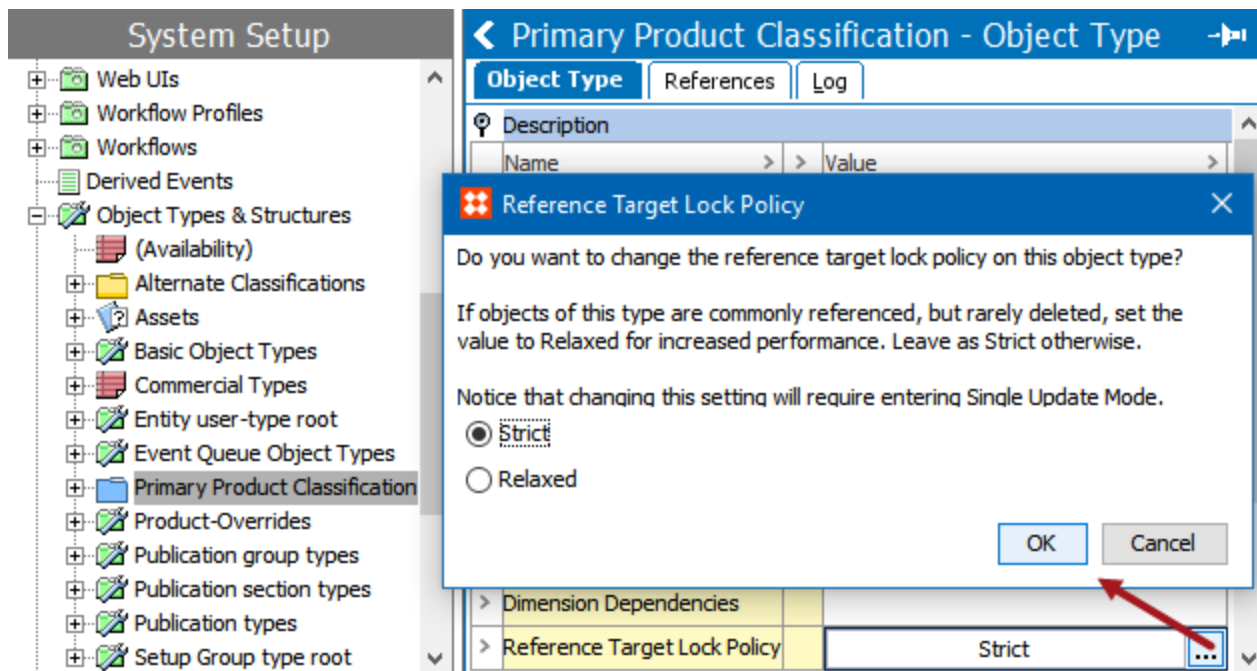
Reference Target Lock Policy on Object Types

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.

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. STEP no longer locks the object. Instead the object type ID receives a shared 'user lock.' This means that multiple imports in parallel can reference the same objects without issues.

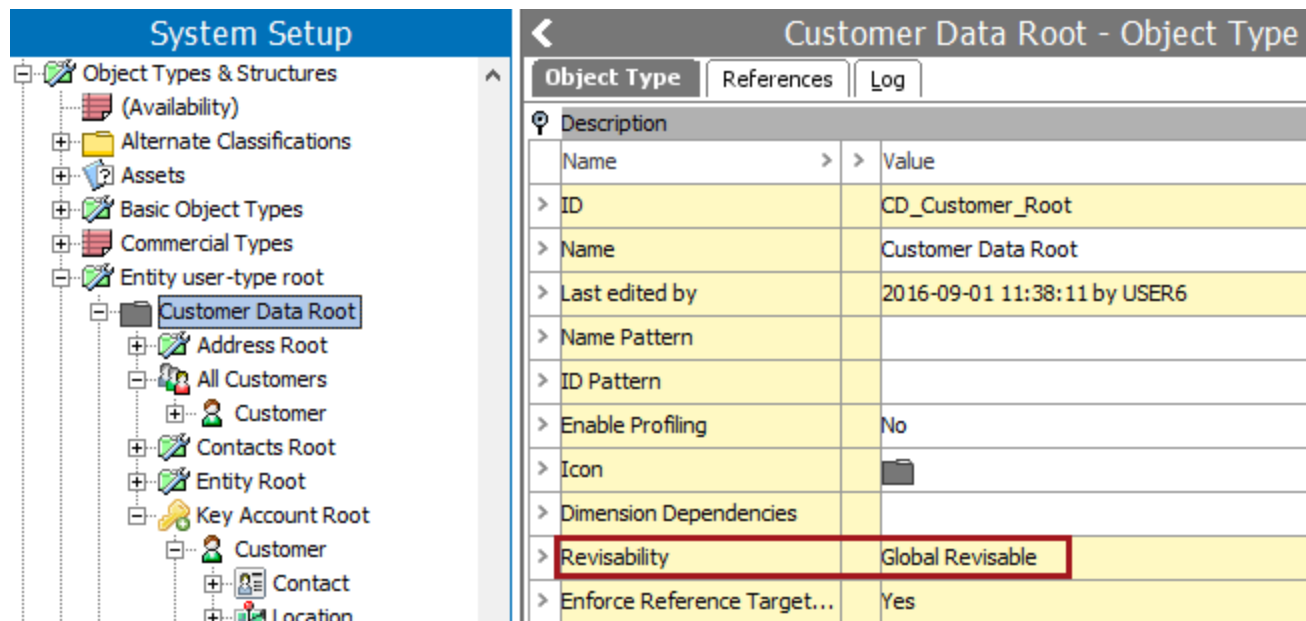
This setting, when used with object types that are frequently referenced but rarely deleted, will improve the performance and stability of parallel inbound imports, bulk updates, and users concurrently creating references to the same objects.



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

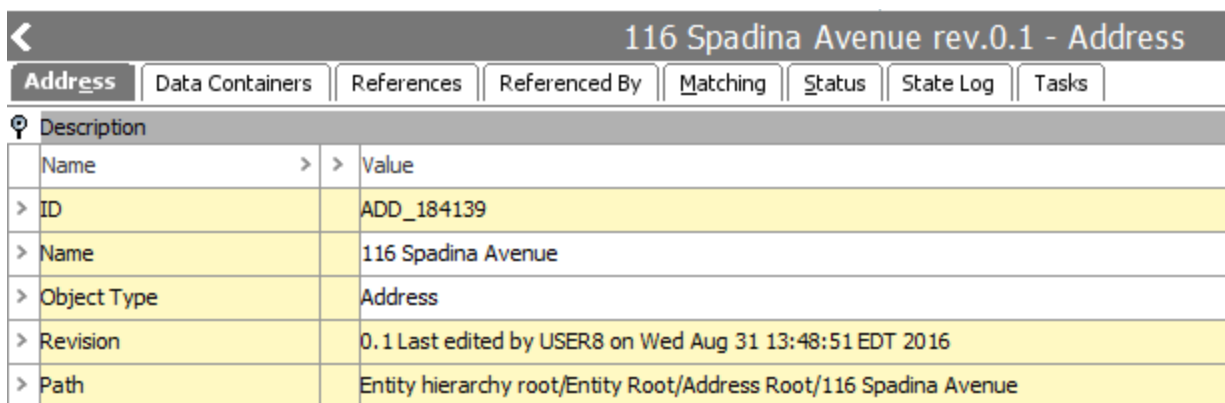
Unlike other objects which are always workspace revisable or globally revisable, the entity object type editor includes a Revisability parameter that can be set to Global Revisable or Workspace Revisable.



The screenshot shows the 'System Setup' interface with the 'Customer Data Root - Object Type' configuration. The 'Revisability' field is highlighted in red, indicating it is set to 'Global Revisable'.

Customer Data Root - Object Type		
Object Type	References	Log
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' configuration. The 'Address' tab is selected, displaying various details about the address object.

116 Spadina Avenue rev.0.1 - Address							
Address	Data Containers	References	Referenced By	Matching	Status	State Log	Tasks
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.

Note: Workflow revisable is normally selected when the party data is imported into STEP from an upstream system after being reviewed and approved and you don't wish to revisit the data again

<

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 / 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:

Tree

Primary Product Hierarchy

Products

Apparel

Footwear

Shoes

Pumps

Pumps Items

2 inch heels

114613

114614

3 inch heels

115121

115122

115123

20695

20714

Pumps Sales Items

3 inch heels

20709-012

3 inch heels

ProductProduct VariantsSub ProductsReferencesReferenced ByImages & DocumentsCommercialTablesCategory Profile

View: ListValidate

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
Create Variant			
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
- Validating Product Variants in the Business Rules documentation

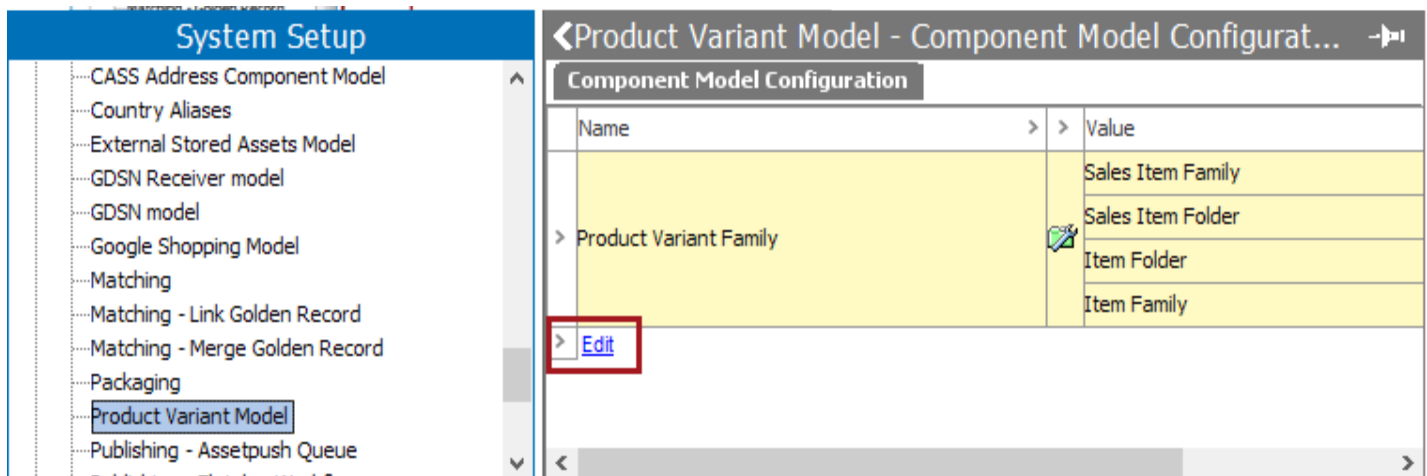
Setting Up Product Variants

To set up product variants, complete the following steps:

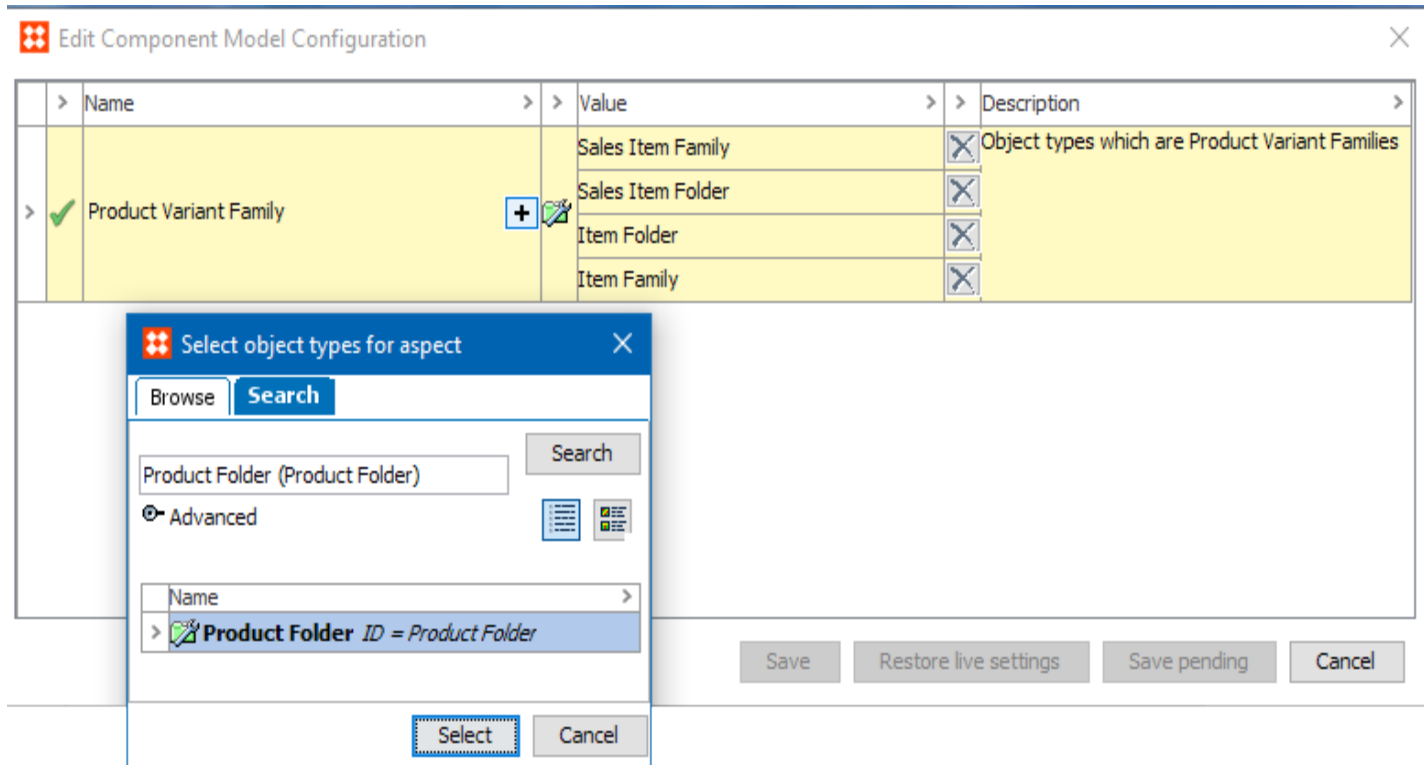
1. Add the object type of the parent folder to the component model **Product Variant Model** to enable the product variant capabilities.
2. Create a metadata attribute and make it valid for the links from the product folder to the related attributes. This enables you to specify which attributes are variant attributes.
3. Add values to the metadata 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**.

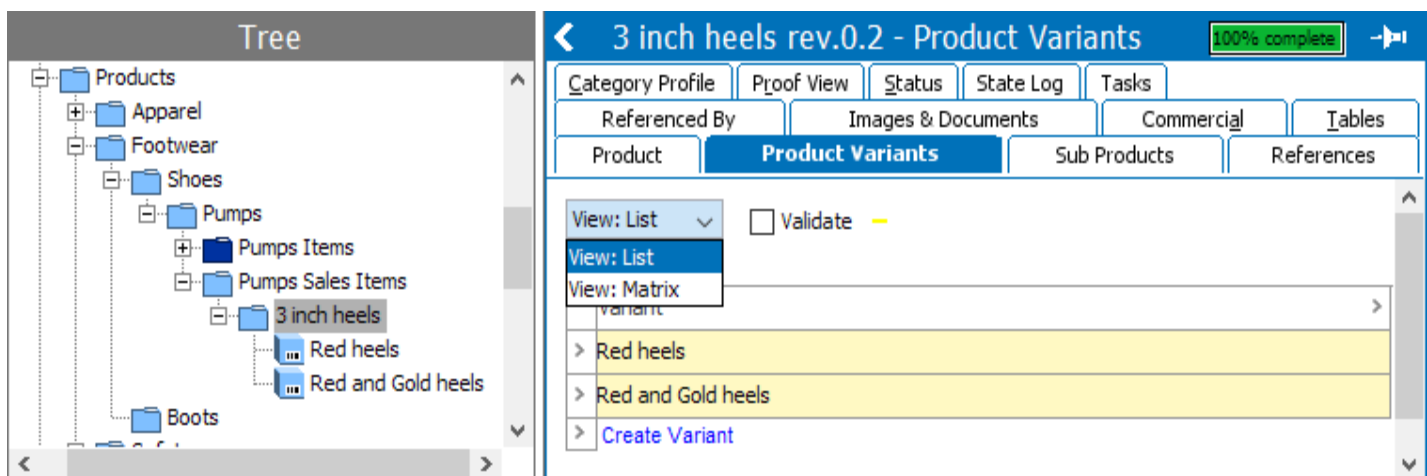


3. In the **Edit Component Model Configuration** dialog, click the plus sign, choose the object type of the parent folder, and then click **Save**.



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 'n2'). 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 example below with Shoes, there is an attribute 'color' linked from the Shoes folder with variant priority attribute value 'n1'. This indicates that the price of the shoes is shown in the views, but it is not used to distinguish variants.

Tree

Footwear

Shoes

Pumps

Pumps Items

2 inch heels

3 inch heels

Pumps Sales Items

3 inch heels

Red heels

3 inch heels rev.0.5 - References

100% complete

Category Profile

Proof View

Status

State Log

Tasks

Referenced By

Images & Documents

Commercial

Tables

Product

Product Variants

Sub Products

References

Linked Attributes from Product Hierarchy

ID	Name	Completeness Score
BrandOwner	Brand Owner	2
Color	Color	n1
Cost	Cost	1

In the following screenshot, the 'color' attribute is treated as a **non-variant attribute**, while the 'size' attribute is treated as a variant attribute based upon the values present in the variant priority attribute (metadata attribute). It is showing that the products are duplicating based on the values present only on the 'size' attribute.

© Stibo Systems - Confidential - STEP Trailblazer 8.2

678

Tree

- Primary Product Hierarchy
 - Products
 - Apparel
 - Upper Body Wear
 - Head Wear
 - Hats and Caps
 - Hats and Caps Items
 - 109011
 - 20801
 - 20805
 - Red Baseball Cap
 - Hats and Caps Sales Items

<Hats and Caps Items rev.0.9 - Product Variants

Category Profile
Proof View
Status
State Log
Tasks

Referenced By
Images & Documents
Commercial
Tables

Product
Product Variants
Sub Products
References

View: List
☐ Validate

Variant	Color	Size
> 109011	Red	8
> 20801	Red	7
> 20805	Red	7
> Red Baseball Cap	Blue	7
> Create Variant		

Product Variant Family Attributes

Attributes linked from a product variant family is categorized into three groups:

Variant attributes: A 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.

- 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.
- 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

- Pressure Relief Valve
- Primary Color
- Product Variant Priority**
- Programmable
- Rated Strength (lb.)
- Seasonal
- Shoe Size (Europe)
- Shoe Size (US)
- Stranded or Solid
- Temper
- T-shirt Size
- UL Listed
- URL
- Voltage
- Wattage

< Product Variant Priority - Attribute

Attribute
References
Attribute Transformation
Validity
Profile
Log
State Log
Tasks

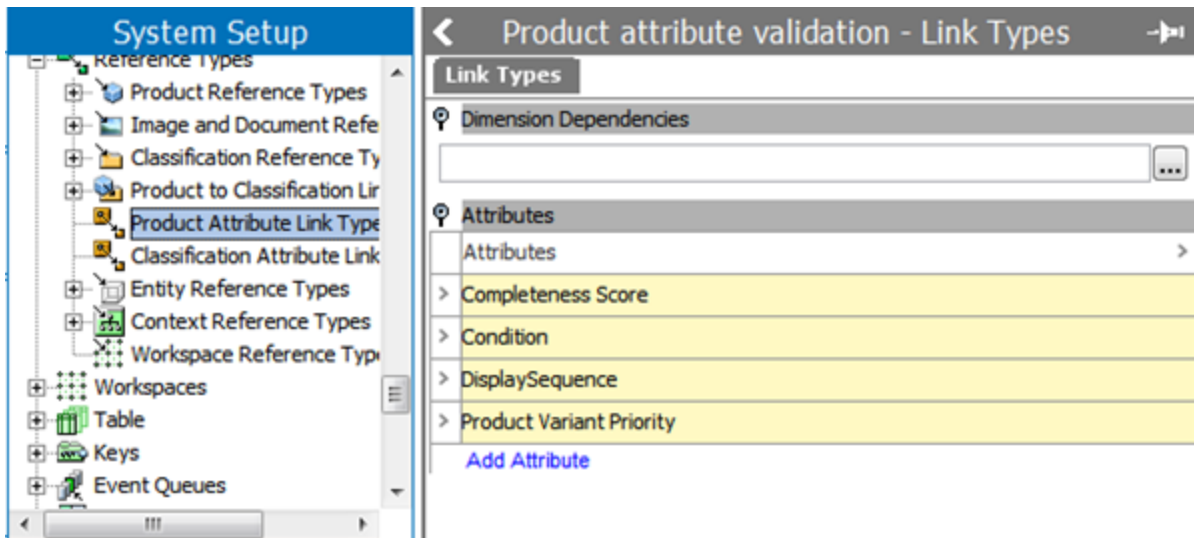
Description
Attribute Validation

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

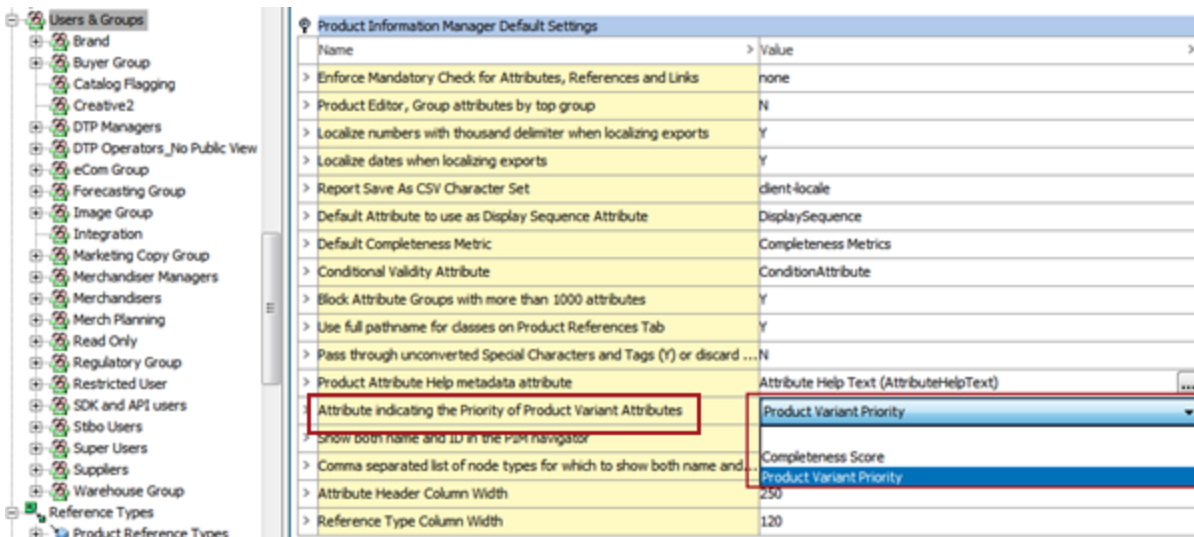
Next, make the attribute valid on the product to attribute links.

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).

Tree

- Primary Product Hierarchy
 - Products
 - Apparel
 - Footwear
 - Shoes
 - Pumps
 - Pumps Items
 - 3 inch heels
 - 115121
 - 115122
 - 115123
 - 20695
 - 20714
 - 2 inch heels
 - 114613
 - 114614
 - Pumps Sales Items
 - 3 inch heels
 - 20709-012
 - 20738-012
 - Translation
 - Boots
 - Safety
 - Hardware
 - Displays
 - Furniture
 - Automotive

Footwear rev.0.3 - References

Product Sub Products References Referenced By Images & Documents Commercial Tables Category Profile Proof View Status State Log Tasks

Index Words

Publications

ID Name

Link to Publication or Section

Linked Attributes from Product Hierarchy

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

Link to Attribute

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 (Europe), and Shoe Size (US) as the variant attributes.

See the **Using Variants List View** topic in this documentation for more information.

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: ListValidate

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

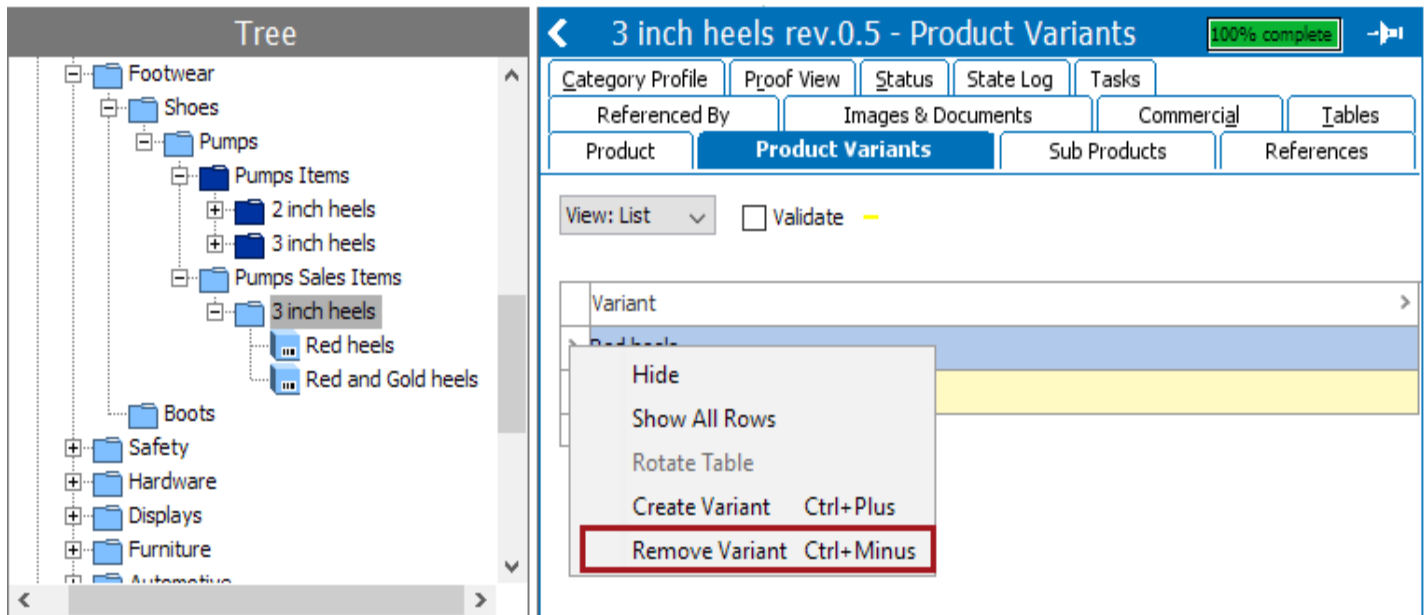
Duplicated by: (120666), (111777), (120170)

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. Even the null value is considered as duplicates here.

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 Web UI

In Web UI, the **VariantsListTabPage** corresponds to the Product Variants list. You can insert it into a **TabControl** on a **NodeDetails** screen.

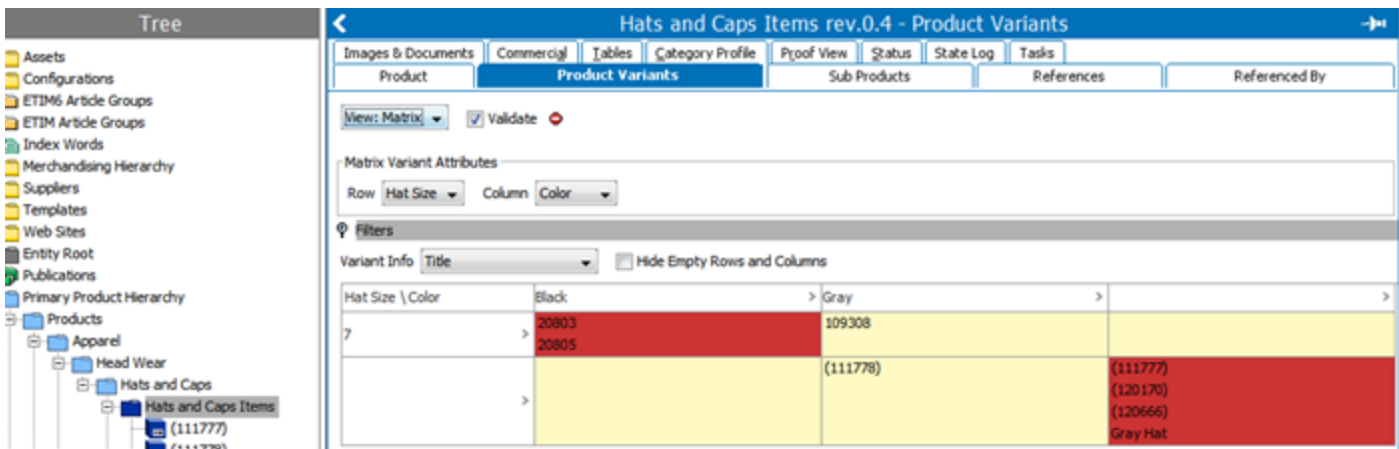
Using the Variants Matrix View

The Matrix view is organized into columns and rows with a variant attribute as row and another variant attribute as column.

Each product variant is placed in a cell with attributes that match the selected row and column attributes. Because a cell contains all product variants that match the selected attributes, it is easier to spot possible duplicates.

To view product variants in matrix view

- 1. In the **Tree**, expand the **Primary Product Hierarchy**, and then select the relevant product.
- 2. Click the **Product Variants** tab, and then from the **View** list select **Matrix**.



- 3. From the **Row** and **Column** lists, select the attributes you want to use as row and column attributes.
- 4. Check **Validate** if you want to view product variants with possible duplicates. Variants are possible duplicates if the variant attributes are identical. The possible duplicate variants are highlighted in red.
- 5. In the **Filters** area, you can filter the view on the product variant attributes that are not used as row or column

When you hover the mouse over a product variant, information is displayed about its variant and non-variant values and possible duplicates.

Variant Info

In this dropdown, there are two default options available, **Title** and **Primary Image and Title**, as well as any available variant attributes.

Filters

Variant Info Title

Color \ Size

black

red

Attribute: Color

Attribute: Size

- **Title:** This will display the results only with the name of the product.

Filters

Variant Info Title ☐ Hide Empty Rows and Columns

Color \ Size	12	13	14	
black	AC-P7000-100			
red		AC-P7000-98		
			AC-P7000-99	(SalesItem-111040)

- **Primary Image and Title:** This will display the result with the name and primary image of the product. If there is no image, it displays only the name for the product.

Filters

Variant Info Primary Image and Title ☐ Hide Empty Rows and Columns

Color \ Size	12	13	14	
black	 AC-P7000-100			
red		 AC-P7000-98		
			AC-P7000-99	(SalesItem-111040)

- **Variant Attributes:** If a variant attribute was selected in the drop-down list, it will display the values. If there are any blank values, it will display as '(empty)'. In the example below the 'Size' attribute values are displayed across the Color attribute values.

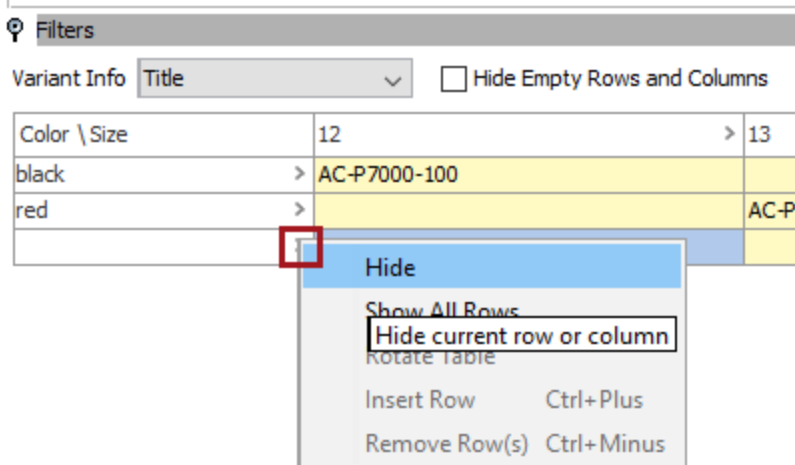
Filters

Variant Info Attribute: Size ☐ Hide Empty Rows and Columns

Color \ Size	12	13	14	
black	12			
red		13		
			14	(empty)

If the user checks the 'Hide Empty Rows and Columns checkbox, the empty rows / columns will be hidden in the view.

Additionally, a user can also hide the rows / columns by clicking the arrow [>] left to the row / column and then choosing the Hide option to temporarily hide them from view.



When a user hovers 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.

Product

Product Variants

Sub Products

References

Referenced By

Images & Documents

Comments

View: Matrix

☐ Validate

Matrix Variant Attributes

Row

Color

Column

Size

Filters

Variant Info

Title

☐ Hide Empty Rows and Columns

Color \ Size	data for auto page	
red	AC-P7000-100	AC-P7000-99
		AC-P7000-98

New Product Variant

Delete Product Variant

In Web UI, 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** (VariantsListTabPage component available for List view). This allows you to generate a variety of combinations of variants over the variant attributes.

This section gives an overview, however for detailed information about variants in the Web UI, see the Web UI documentation.

A user can create new variants by clicking the icon, , on the screen.

In the **Create Variants** dialog, choose the variant attributes that need to be combine.

Create variants

×

Object type

Active Products

▼

Hat Size

Source

7

4

6

▲

▼

Result

6.5

▲

▼

➤

➤

➤

Add value

Color

Source

Blue

Orange

▲

▼

Result

Green

▲

▼

➤

➤

➤

Add value

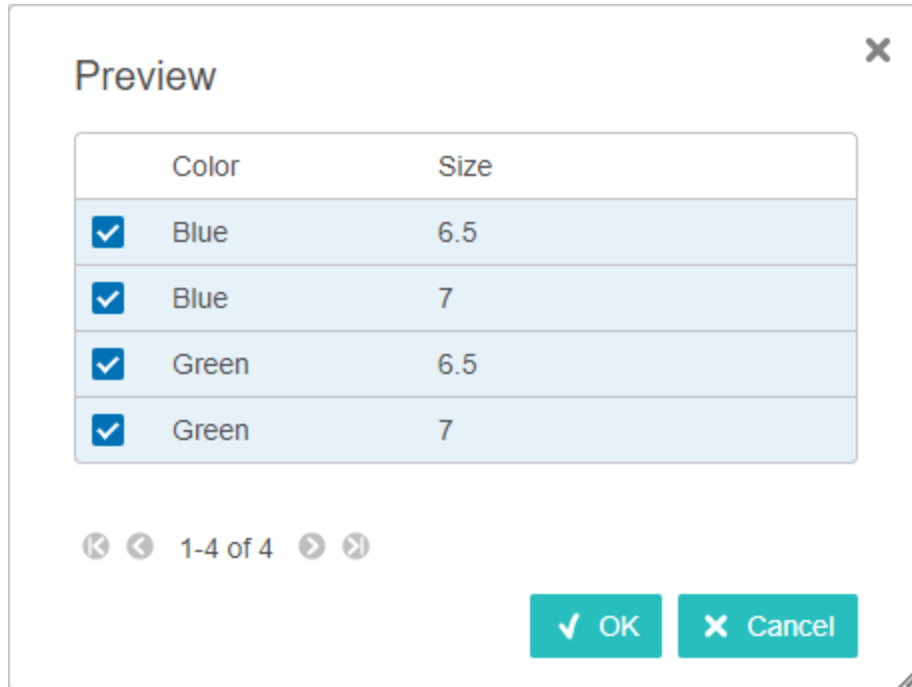
1 variants

✓ OK

✗ Cancel

Notice that if there are any duplicates, then the number above the cancel button displays the number duplicated.

If the parameter **PreviewVariantsCombinations** has been checked on the **CreateVariantsAction** in design mode, a preview dialog is displayed before the chosen combinations are created. This prevents specific combinations from being created.



On the CreateVariantsAction in design mode, there is also a **PreventDuplicateCreation** check box. If the parameter is checked, the combinations that match an existing variant will not be created when the action is applied.

Properties

Configuration

Web UI style

productdetails

Save

Close

New...

Delete

Rename

Save as

Create Variants Action Properties [go to parent](#)

Cancel Button Label

i18n.stibo.portal.server.component

Dialog Title

i18n.stibo.portal.server.component

Label

i18n.stibo.portal.server.component

Ok Button Label

i18n.stibo.portal.server.component

Prevent Duplicate Creation

☒

Preview Page Size

50

Preview Variant Combinations

☒

Child Components

Business Condition: Validate Product Variant

A product variant family is a group of products where the members are considered to be the same product except for variations in the predefined attributes. Variations are, for example, different sizes or colors. This condition works on the Product Variant level, and returns False if the product variant is a duplicate. For more information, see the **Product Variants** topic in the **System Setup / Super User Guide** documentation.

For example, a product folder named 'Shoes' contains children products and has one variant attribute: Size. This condition can be used to identify the duplicate sizes (variants) within the product variant family.

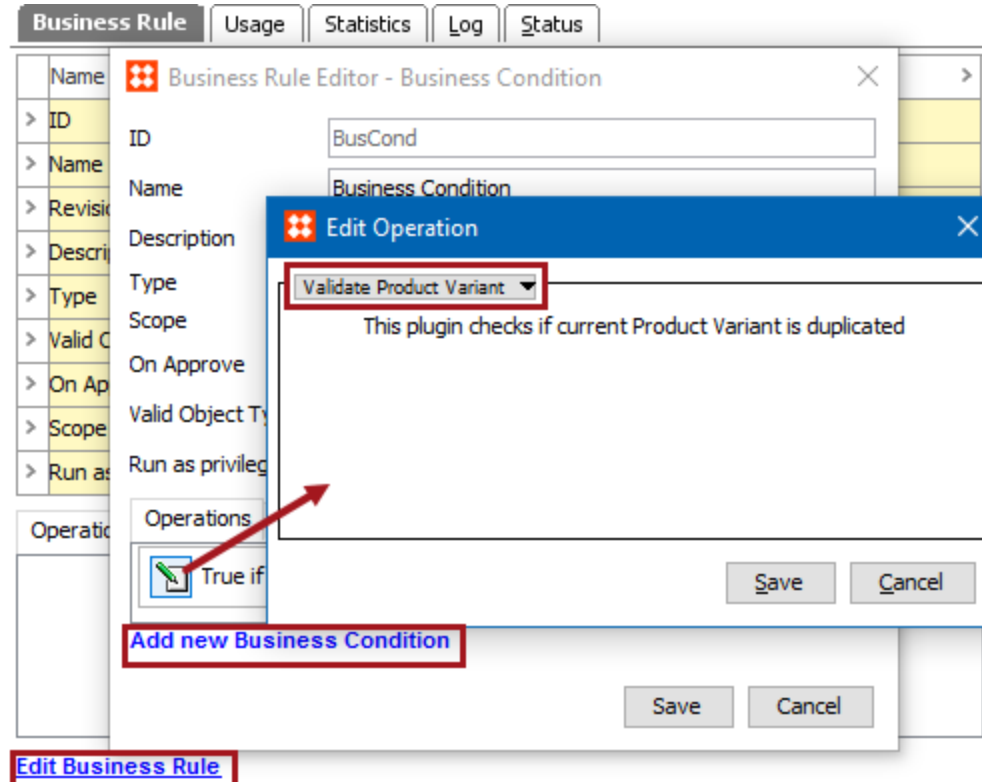
Note: This condition can affect performance based on the number of siblings and variant attributes to be checked.

Prerequisites

Before using this business condition:

1. Verify the configuration on the Product Variant Families is correct since it is used by this condition.
2. Create a business rule as defined in the **Creating a Business Rule or Library** topic.
3. Edit the business rule to configure the operation as defined in **Editing a Business Rule** topic.

Configuration



1. On the Edit Operation dialog, select **Validate Product Variant** from the dropdown.
2. Click the **Save** button to add the operation to the business rule editor.

Recycle Bin

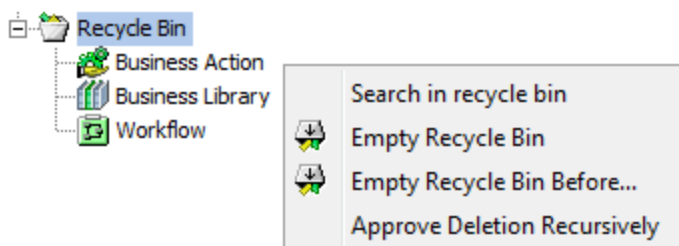
The deletion concept in STEP is quite similar to what you know from Windows and Mac OS. Thus, if you delete the objects in System Setup, they will be moved to the Recycle Bin in the System Setup tab. However, not all objects will display in the Recycle Bin.

Below are the objects that will display in the recycle bin when deleted from System Setup:

- Business Rules (Business Condition and Business Action)
- Business Rule Libraries
- Integration Endpoints
- Workflows

Note: The Recycle Bin in System Setup is different from the one in Tree tab. For more information about the Recycle Bin for the Tree tab, see the **Recycle Bin** topic with the **Object Maintenance in Tree** documentation.

The Recycle Bin for System Setup can be found in the System Setup tab. Right-clicking on the Recycle Bin will display a menu, as shown below.

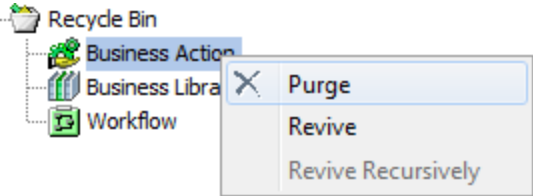


- **Search in recycle bin:** Allows the search for deleted objects in the recycle bin.

Note: Items / objects in the Recycle Bin are not searched when doing STEP searches nor when using the workbench Goto functionality.

- **Empty Recycle Bin:** When selected, the contents of the Recycle Bin are permanently deleted.
- **Empty Recycle Bin Before...:** When selected, the contents of the Recycle Bin before a specified date are permanently deleted.
- **Approve Deletion Recursively:** Only works for the Recycle Bin in the Tree tab, as objects in system setup cannot be approved.

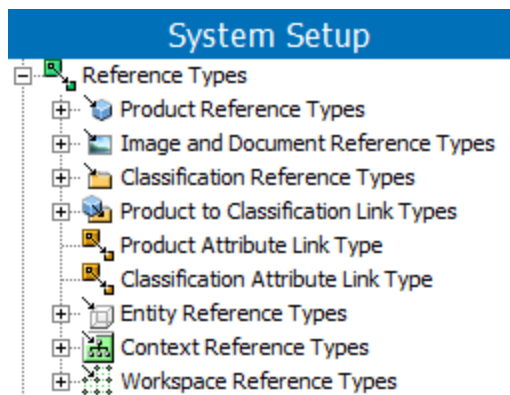
Deleted business rules, workflows, and integration endpoints can be seen in the Recycle Bin and can be revived from that location. Similarly they can be purged.



Reference and Link Types

References and links allow you to establish a relationship between objects. Fundamentally, references and links are similar to attributes in that they can be internally or externally maintained, can have dimension dependencies, can be inherited, and can allow multiple values. References and links are different from attributes in that they model relationships between objects, that the relationships between objects have directionality, and that they can also include metadata in the form of attributes that live on the reference or link itself.

Referencing or linking objects allows those objects to be used for different purposes, while avoiding data duplication and minimizing data maintenance. For example, a single stock image could be referenced a for a whole family of products. Reference and link types are housed and created in System Setup under the Reference Types node.



STEP includes the following options for object relationships: References, Product to Classification Links, Product Attribute Links, and Classification Attribute Links. Each is defined below.

References

References allow you to define a connection between two objects in STEP, essentially they are a set of rules for relationships between objects. For example, a reference can associate an image with a product, or join multiple addresses with a single company, or connect a classification to an object, among other uses.

Prerequisites for Reference Types

Before creating a reference type, consider the available options, and determine the type most suited for your data as described in **Reference Types**.

Setup Requirements for References

Setting up and using any reference involves the following steps:

1. Create the appropriate reference type as described in **Creating a Reference Type**.
2. Apply the reference to the necessary objects as described in **Creating a Reference**.

Additional Information About References

- Understand the benefits of attribute groups as described in **Attribute Groups for Reference and Link Types**
- Understand the effects of dimension dependencies as described in **Dimension Dependent Reference and Link Types**
- Understand how visibility affects dimension-dependent references as described in **Visibility of Reference and Link Types**
- Understand the importance of designating the source and target as described in **Direction of a Reference**
- Review effects of the Parent/Child Relation as described in **Entity Reference Types**
- Modify settings on existing reference types as described in **Maintaining a Reference Type**
- Understand metadata functionality as described in **Metadata Attributes on Reference and Link Types**
- Understand the uses for multiple references as described in **Multiple References for the Same Objects**
- Review effects of inheritance as described in **Inheritance Example for a Reference**

Product to Classification Links

Product to classification links allow you to define a connection between a product and a classification, in addition to showing the product as a child to the classification and are described in **Product to Classification Link Types**.

Setup Requirements for Product to Classification Links

Setting up and using a product to classification link involves the following steps:

1. Create a link type as described in **Creating a Product to Classification Link Type**.
2. Apply the link to the necessary objects as described in **Creating a Product to Classification Link**.

Additional Information About Product to Classification Links

- Understand the benefits of attribute groups as described in **Attribute Groups for Reference and Link Types**
- Understand the effects of dimension dependencies as described in **Dimension Dependent Reference and Link Types**
- Understand how visibility affects dimension-dependent link types as described in **Visibility of Reference and Link Types**
- Modify settings on existing link types as described in **Maintaining a Product to Classification Link Type**
- Understand metadata functionality as described in **Metadata Attributes on Reference and Link Types**
- Review effects of inheritance as described in **Inheritance Example for a Product to Classification Link Type**

Product Attribute Links

A product attribute link allows a specification attribute to be available when the product is a child of the selected level of the primary product hierarchy. A product attribute link can also include dimension dependencies and metadata attributes. This enables you to control when the attribute is available and/or published based on its location in the product hierarchy.

Using the Product Attribute Link

The product attribute link is pre-configured and ready to use as described in **Creating a Product Attribute Link**.

Optional Setup for the Product Attribute Link

Set up is only necessary when dimension dependencies and metadata attributes are required as described in **Setting Up an Attribute Link Type**.

Additional Information About Product Attribute Links

- Understand metadata functionality as described in **Metadata Attributes on Reference and Link Types**
- Understand the available functionality as described in **Product Attribute Link Type**

Classification Attribute Links

A classification attribute link allows a specification attribute to be available when the object is also linked into the specified classification hierarchy via a Product to Classification Link Type. Classification attributes links can also include dimension dependencies and metadata attributes. This enables you to control when the attribute is available and/or published based on the classification hierarchy.

Using the Classification Attribute Link

The classification attribute link is pre-configured and ready to use as described in **Creating a Classification Attribute Link**.

Optional Setup for the Classification Attribute Link

Set up is only necessary when dimension dependencies and metadata attributes are required as described in the **Setting Up an Attribute Link Type**.

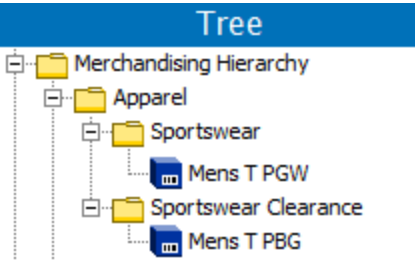
Additional Information About Classification Attribute Links

- Understand metadata functionality as described in **Metadata Attributes on Reference and Link Types**
- Understand the available functionality as described in **Classification Attribute Link Type**

Classification Attribute Link Type

The Classification Attribute Link Type is pre-configured and allows a specification attribute to be available when the product is linked into the specified classification hierarchy via a Product to Classification Link Type.

For example, consider the attribute 'Marked Down %' that is only needed for clearance items. Using two classifications, one for general items and another for clearance items, allows you to display the clearance attribute only on the group of products where it is required.



With additional setup, the Classification Attribute Link Type can also include dimension dependency and metadata attributes as described below.

Dimension Dependency

Dimension dependencies on the Classification Attribute Link Type enable you to use a variety of values for the linked attribute based on the selected dimension.

For example, when the language dimension is set, the same attribute could hold the value in English and in French, and be used appropriately as required by a website or export.

Mens T PBG rev.0.8 - Compare Contexts			0% complete
Compare Contexts			References
View: Show all			Referenced By
	English US	French France	
ID	179916	179916	
Name	Mens T PBG	Mens T PBG	
Object Type	Item	Item	
Revision	0.8 Last edited by USERJ on Wed Mar 15 15:56:17 EDT 2017	0.8 Last edited by USERJ on Wed Mar 15 15:56:17 EDT 2017	
Path	Primary Product Hierarchy/Products/Apparel/Upper Body W...	Primary Product Hierarchy/Products/Apparel/Upper Body W...	
Approved	✗ Last Approved on Thu Aug 11 12:15:42 EDT 2016	✗ Last Approved on Thu Aug 11 12:15:42 EDT 2016	
BuyItNowText US	This offer won't last long!	Cette offre ne durera pas longtemps !	

For more information, see **Dimension Dependent Attributes**.

Metadata Attributes

Metadata (description) attributes enable you to control when the linked attribute is published based on an additional setting, or provide additional data for any other purpose.

For example, a description attribute named 'Show on Web' has been added to the Classification Attribute Link Type. This description attribute holds the value 'yes' or 'no' and indicates if the specification attribute 'BuyItNowText' should be displayed on the web.

Tree

Merchandising Hierarchy

Apparel

Sportswear

Sleepwear

T-shirts

Mens T PGW

Sportswear Clearance

Mens T PBG

Sportswear Clearance rev.0.1

ClassificationSub ProductsReferencesReferenced ByImages & Document

Applied Privileges

Visible Objects in Other Contexts

Attributes

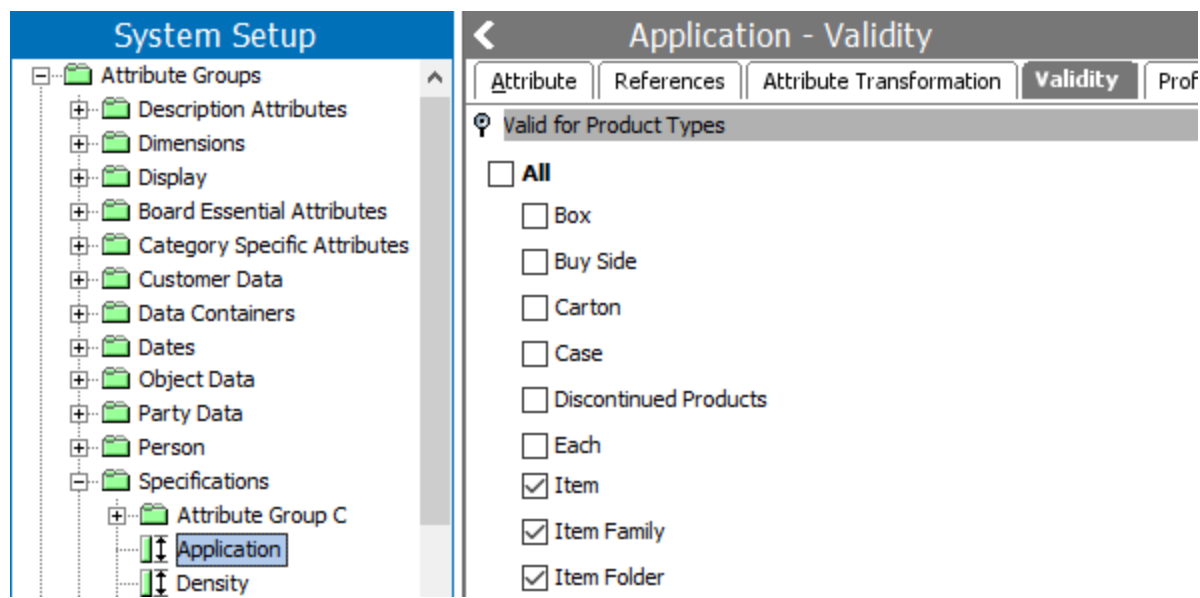
ID	Name	Show On Web	Attribu
> BuyItNowText	BuyItNowText US	Y	
> MarkedDown%	Marked Down		

For more information, see **Description Attributes**.

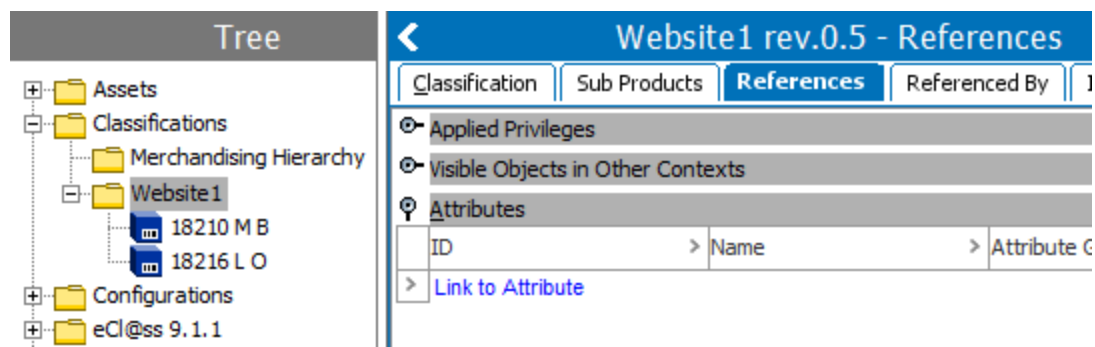
Creating a Classification Attribute Link

Adding a classification attribute link allows a specification attribute to be available when a product is linked into the specified classification hierarchy via a Product to Classification Link Type.

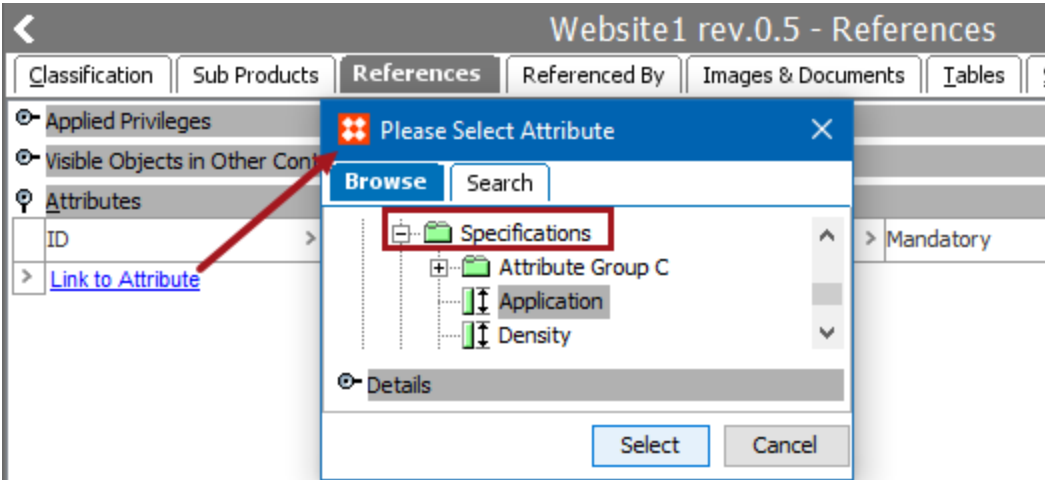
- 1. In System Setup, verify that the specification attribute to be linked has Validity that includes the products in the classification.



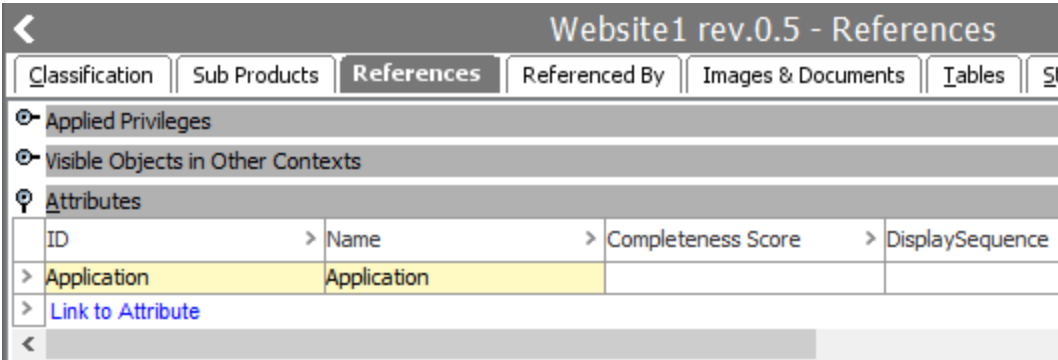
- 2. On Tree, select the classification that contains the products (via a Product to Classification Link Type) that should display the specification attribute.



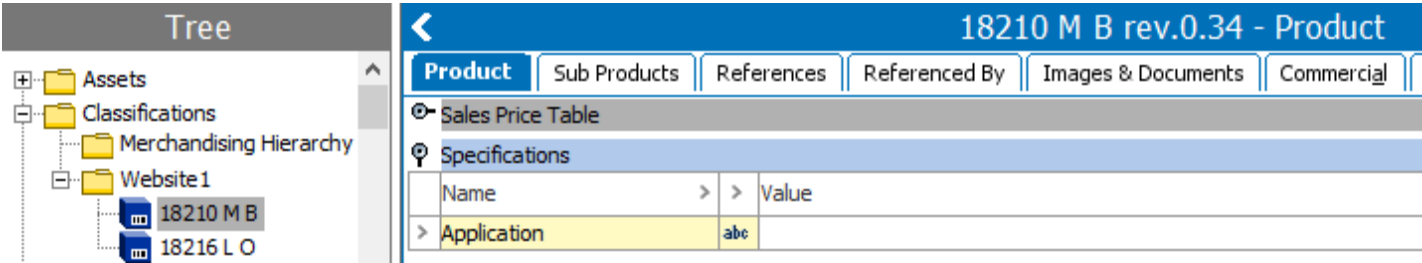
- 3. On the References tab > Attributes flipper> click the 'Link to Attribute' link to display the Select Attribute dialog.



4. Note the attribute group of the desired specification attribute, choose the attribute, and click the Select button. The specification attribute is displayed under the Attributes flipper.



5. Select a product currently displayed in the classification, open the attribute group flipper identified previously, and verify that the linked attribute is available.



Note that other products of the same type that are not linked into the classification do not include the specification attribute (the flipper is absent).

Tree

Cotton T-Shirts

18210 M B

18212 L B

18213 M O

18216 L O

< 18212 L B rev.0.22 - Product

Product

Sub Products

References

Referenced E

Mass Creation

Product Details

Sales Price Table

Website Item Descriptions

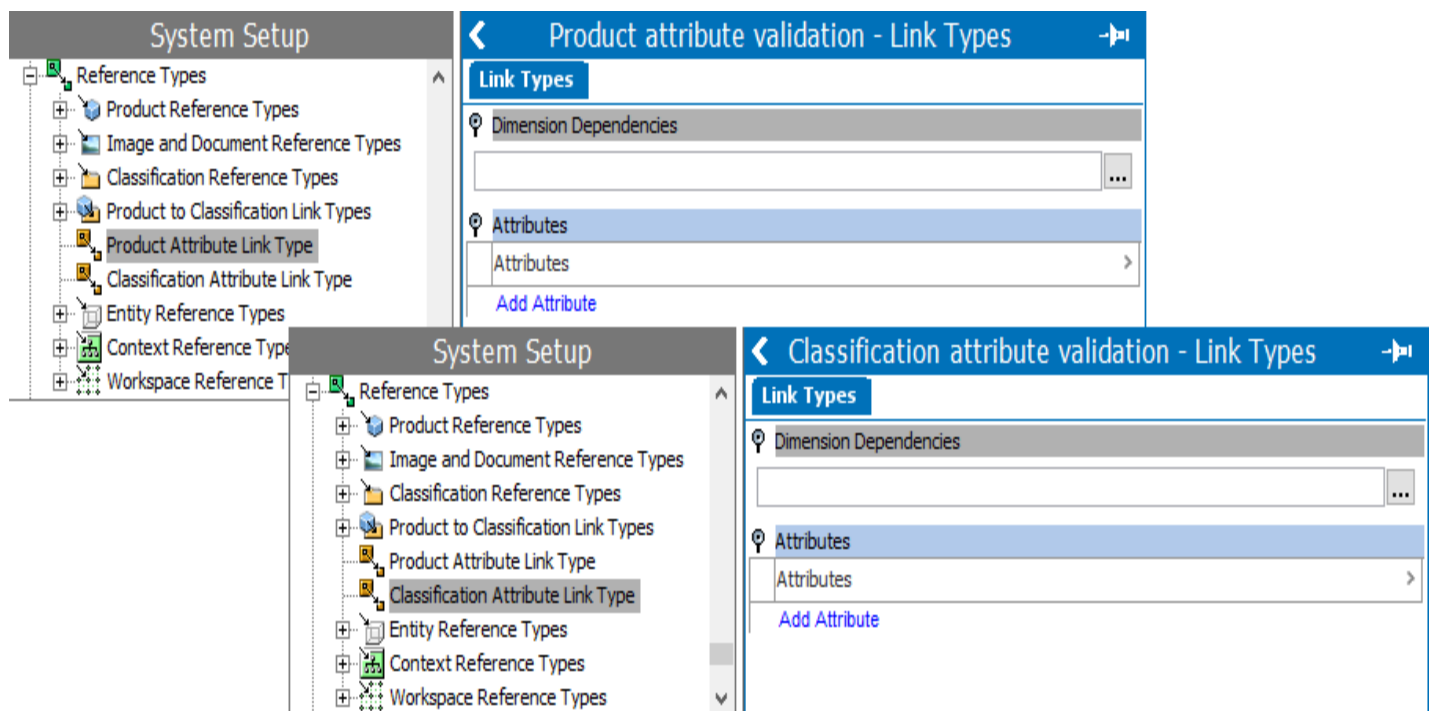
Setting Up an Attribute Link Type

The 'Product Attribute Link Type' and the 'Classification Attribute Link Type' are pre-configured. By default, linked attributes are displayed for the selected product hierarchy or classification hierarchy. If dimension dependency or metadata attributes are required, use the following steps to further configure the appropriate link type.

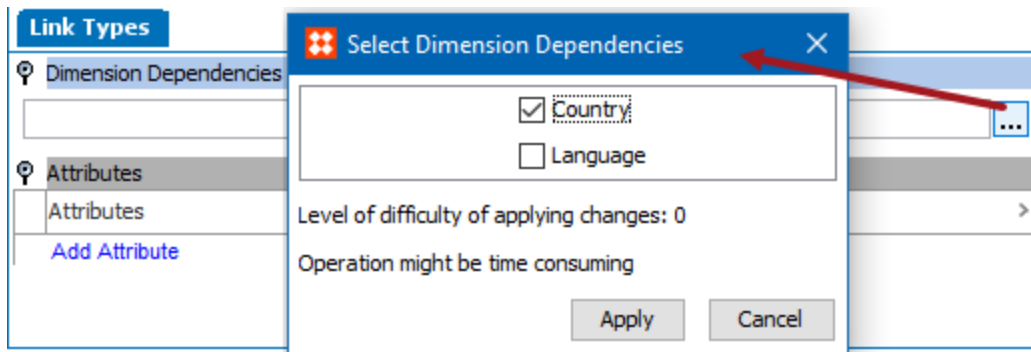
Only the default attribute link types can exist in STEP. Unlike references, user-created types are not allowed. The settings on the default attribute link types are applied to all 'Product Attribute Link Type' and 'Classification Attribute Link Type' used in the database.

Note: Setup is the same for both the Product Attribute Link Type and the Classification Attribute Link Type.

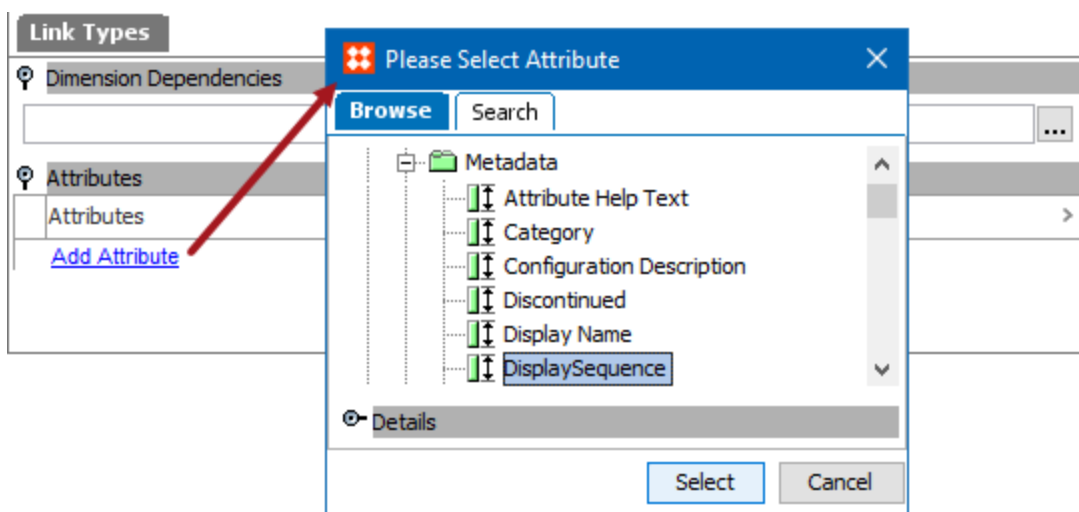
1. To set up an attribute link type for dimension dependencies or metadata attributes, in System Setup > Reference Types > click the desired attribute link type to display the editor.



2. If required, set the dimension dependencies on the Link Types tab, under the Dimension Dependencies flipper. Only one dimension is recommended if any, due to the increased complexity of managing more than one. Setting an attribute link type to be dimension dependent will allow variations of values on the linked specification attribute across dimension points, and by extension, contexts.



- Click the ellipsis button (...) to display the Select Dimension Dependencies dialog.
 - Choose the required dimension(s).
 - Click **Apply**.
3. If required, set metadata attributes on the Link Types tab, under the Attributes flipper. Specification attributes and groups cannot be added.



- Click the **Add Attribute** link to display the Select Attribute dialog.
 - Browse or search for the required description attribute and choose it.
 - Click **Select**.
4. The link type displays the selected dimension dependencies and metadata attributes.

Link Types

Dimension Dependencies

Country;

...

Attributes

Attributes

DisplaySequence

Add Attribute

If the same metadata attribute is set for both attribute links, the product's References tab shows the attribute in both the 'Linked Attributes from Product Hierarchy' and the 'Linked Attributes from Classification Hierarchy' flippers.

Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Ta
Linked Attributes from Product Hierarchy						
DisplaySequence	ID	Name	Condition			
	Color	Color				
	LongDescription	Long Description				
Link to Attribute						
Linked Attributes from Classification Hierarchy						
ID	Name	DisplaySequence	Show On Web			
BuyItNowText	BuyItNowText US		Y			
MarkedDown%	Marked Down					

- Add metadata values as described in **Maintaining Attribute Link Metadata Values**.

Maintaining Attribute Link Metadata Values

When a metadata attribute has been added to the **Product Attribute Link Type**, it is displayed on the References or Referenced By tabs under the 'Linked Attributes from Product Hierarchy' flipper for all linked attributes. In the image below, the 'Show on Web' description attribute is metadata on the product attribute link.

Tree

Products

Apparel

Upper Body Wear

Head Wear

Hats and Caps

Hats and Caps Items

Industrial Hard Hat

Industrial Hard Hat rev.0.9 - Reference

ProductSub ProductsReferencesReferenced ByImages & DocumentsCommercialTablesCategories

Linked Attributes from Product Hierarchy

DisplaySequence	ID	Name	Condition	Product Variant Priority	Show On Web
9	ManufacturerWarranty	attrWarranty		10	Y
5	AvailabilityEnd	Availability End			
2	AvailabilityStart	Availability Start			

When a metadata attribute has been added to the **Classification Attribute Link Type**, it is displayed on the References or Referenced By tabs under the 'Linked Attributes from Classification Hierarchy' flipper for all linked attributes. In the image below, the 'Show on Web' description attribute is metadata on the classification attribute link.

Tree

Merchandising Hierarchy

Apparel

Sportswear

Sleepwear

T-shirts

Mens T PGW

Sportswear Clearance

Mens T PBG

Mens T PBG rev.0.8 - References

ProductSub ProductsReferencesReferenced ByImages & DocumentsCommercialTablesCategories

Linked Attributes from Product Hierarchy

Linked Attributes from Classification Hierarchy

ID	Name	DisplaySequence	Show On Web
BuyItNowText	BuyItNowText US		Y
MarkedDown%	Marked Down		

To add a new metadata attribute for an attribute link type, see the **Setting Up an Attribute Link Type**.

Values can be added for a metadata attribute on an attribute link type and be inherited down, or values can be written at the local level.

Inherited Metadata Values

A value being inherited is indicated by the small green arrow (↗) in the first column of the list. Inherited values must be edited at the parent level, and are displayed as read-only for all children.

Use inheritance when the value is the same from all or most of the products from the 'Inherited from' level and below.

Edit an Inherited Value

Note: Although the following images show the Product Attribute Link, the process is the same for Classification Attribute Link maintenance.

1. Click the link in the 'Inherited from' column to display the parent object where the value is being held.

Industrial Hard Hat rev.0.9 - References							
Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Tables	Category Profile
Linked Attributes from Product Hierarchy							
DisplaySequence	ID	Name	Show On Web	Attribute Groups	Mandatory	Inherited from	
9	ManufacturerWarranty	attrWarranty	Y	Buyer, View, Item Descrip...		Products	
5	AvailabilityEnd	Availability End		Buyer, Approve, Buyer, Vi...		Products	
2	AvailabilityStart	Availability Start		Buyer, Approve, Buyer, Vi...		Products	

2. Edit the value on the parent object. Also note that the 'Inherited from' column is not displayed for a parent object.

Products rev.0.40 - References							
Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Tables	Category Profile
Linked Attributes from Product Hierarchy							
DisplaySequence	ID	Name	Show On Web	Attribute Groups	Mandatory	Inherited from	
9	ManufacturerWarranty	attrWarranty	N	Buyer, View, Item Description I...			
5	AvailabilityEnd	Availability End		Buyer, Approve, Buyer, View, ...			
2	AvailabilityStart	Availability Start		Buyer, Approve, Buyer, View, ...			

All children display the new value.

Industrial Hard Hat rev.0.9 - References							
Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Tables	Category Profile
Linked Attributes from Product Hierarchy							
DisplaySequence	ID	Name	Show On Web	Attribute Groups	Mandatory	Inherited from	
9	ManufacturerWarranty	attrWarranty	N	Buyer, View, Item Descrip...		Products	
5	AvailabilityEnd	Availability End		Buyer, Approve, Buyer, Vi...		Products	
2	AvailabilityStart	Availability Start		Buyer, Approve, Buyer, Vi...		Products	

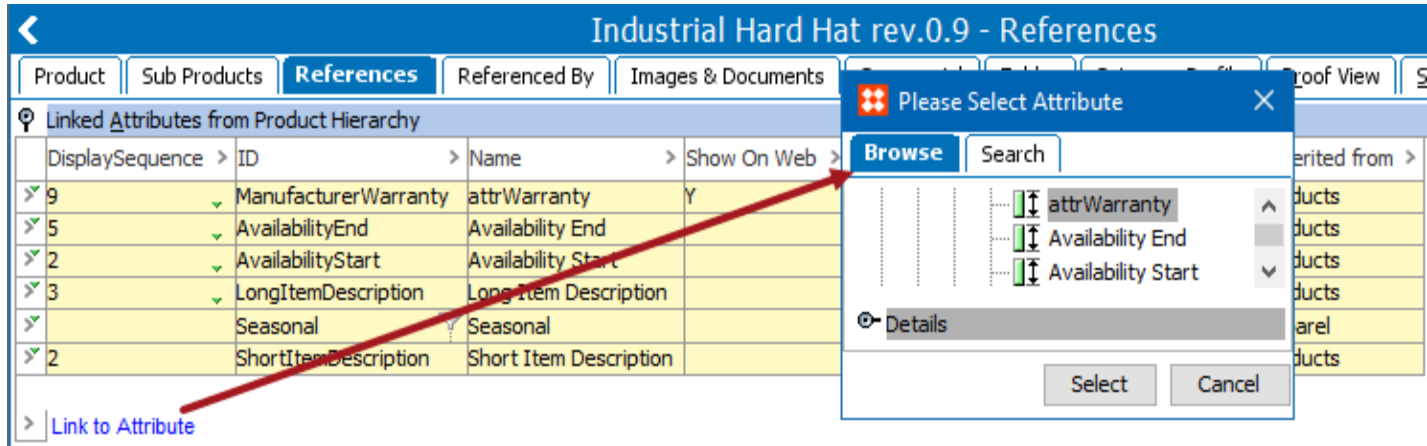
Local Metadata Values

A local value overwrites any inherited value from a higher level and can be edited. It is indicated by the absence of the small green arrow (>) in the first column of the list and the absence of the data in the 'Inherited from' column.

Add a Local Value

Note: Although the following images show the Product Attribute Link, the process is the same for Classification Attribute Link maintenance.

1. On the product References or Referenced By tabs, open the 'Linked Attributes from Product Hierarchy' (or 'Linked Attributes from Classification Hierarchy') flipper.
2. Click the **Link to Attribute** link to display the Select Attribute dialog.
3. Choose a specification attribute (description attributes are not allowed) and click the **Select** button.



4. Now that the attribute can be edited, add a local value.

Industrial Hard Hat rev.0.9 - References							
Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Tables	Category Profile
Linked Attributes from Product Hierarchy							
DisplaySequence	ID	Name	Show On Web	Attribute Groups	Mandatory	Inherited from	
> 9	ManufacturerWarranty	attrWarranty		Buyer, View, Item Descrip...	☐		
> 5	AvailabilityEnd	Availability End		Buyer, Approve, Buyer, Vi...	☐	Products	
> 2	AvailabilityStart	Availability Start		Buyer, Approve, Buyer, Vi...	☐	Products	

The local value shows at the level where it is added, and at any lower levels, via inheritance.

Industrial Hard Hat rev.0.9 - References							
Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Tables	Category Profile
Linked Attributes from Product Hierarchy							
DisplaySequence	ID	Name	Show On Web	Attribute Groups	Mandatory	Inherited from	
> 9	ManufacturerWarranty	attrWarranty	N	Buyer, View, Item Descrip...	☐		
> 5	AvailabilityEnd	Availability End		Buyer, Approve, Buyer, Vi...	☐	Products	
> 2	AvailabilityStart	Availability Start		Buyer, Approve, Buyer, Vi...	☐	Products	

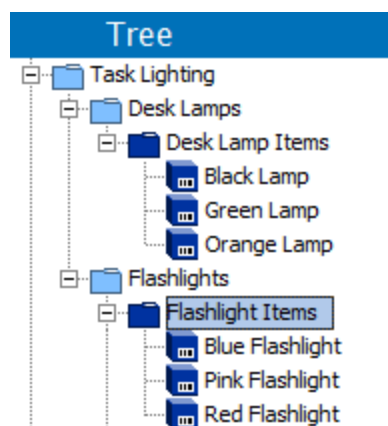
Restore an Inherited Value

Note: Although the following images show the Product Attribute Link, the process is the same for Classification Attribute Link maintenance.

Product Attribute Link Type

The Product Attribute Link Type is pre-configured, and allows a specification attribute to be available when the product is a child of the selected level of the PPH (primary product hierarchy). This reduces the number of unnecessary attributes displayed in the product editor.

For example, consider a PPH where one set of products requires batteries and the other set does not. In the following PPH, the 'Desk Lamps' level does not need battery values, but the 'Flashlights' level does. Instead of making battery information available at all levels via the object type, the product attribute link is used to display the attribute only where it is needed.



With additional setup, the Product Attribute Link Type can also include dimension dependency and metadata attributes as described below.

Dimension Dependency

Dimension dependencies enable you to use a variety of values for the linked attribute based on the value of the selected dimension.

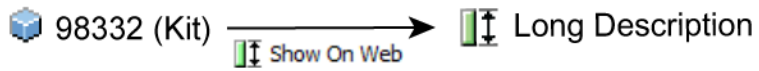
For example, if the language dependency is selected for the Product Attribute Link, the attribute 'Long Description' can be written in English for the English US context and written in French for the French FR context.

For more information, see **Dimension Dependent Attributes**.

Metadata Attributes

Metadata (description) attributes enable you to control when the linked attribute is published based on an additional setting.

For example, a description attribute named 'Show on Web' has been added to the Product Attribute Link Type. This description attribute holds the value 'yes' or 'no' and indicates if the specification attribute 'Long Description' should be displayed on the web. The description attribute 'Show on Web' will be available on all attributes on the References tab under the 'Linked Attributes from Product Hierarchy' flipper.



For more information, see **Description Attributes**.

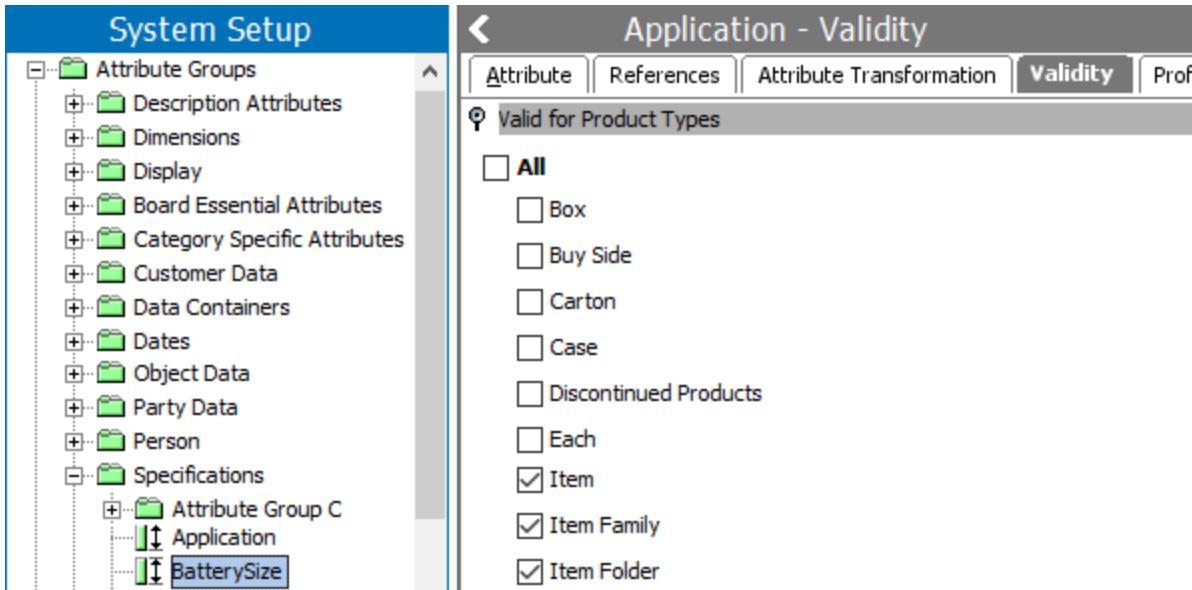
For information on creating a product attribute link type, see **Setting Up an Attribute Link Type**.

For information on creating a product attribute link, see **Creating a Product Attribute Link**.

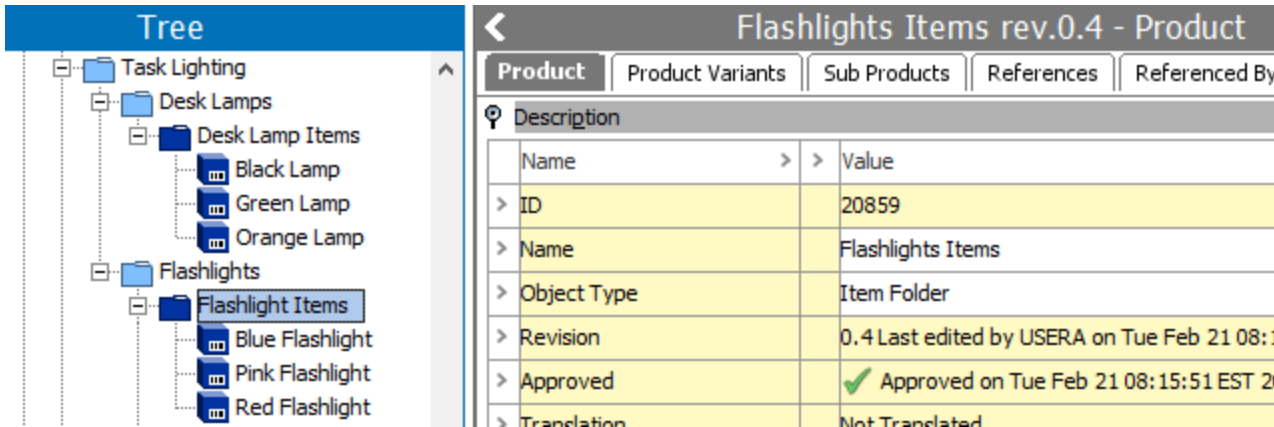
Creating a Product Attribute Link

Adding a product attribute link allows a specification attribute to be available when the object is a child of the selected level of the PPH (primary product hierarchy).

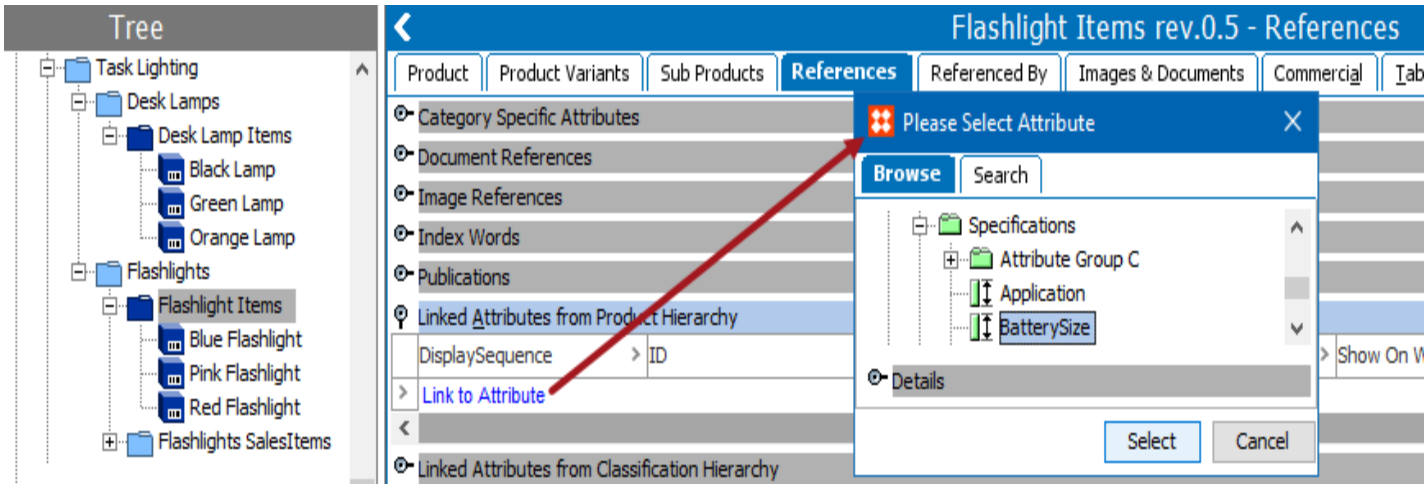
- 1. In System Setup, verify that the specification attribute to be linked has Validity that includes the objects in the intended levels of the PPH.



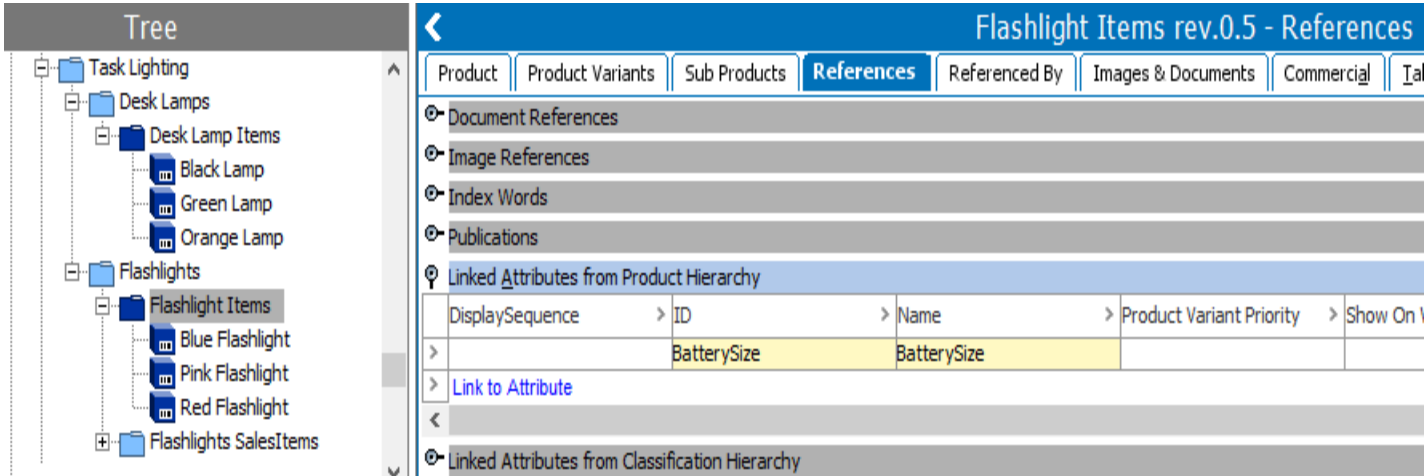
- 2. On Tree, select the PPH that should display the specification attribute.



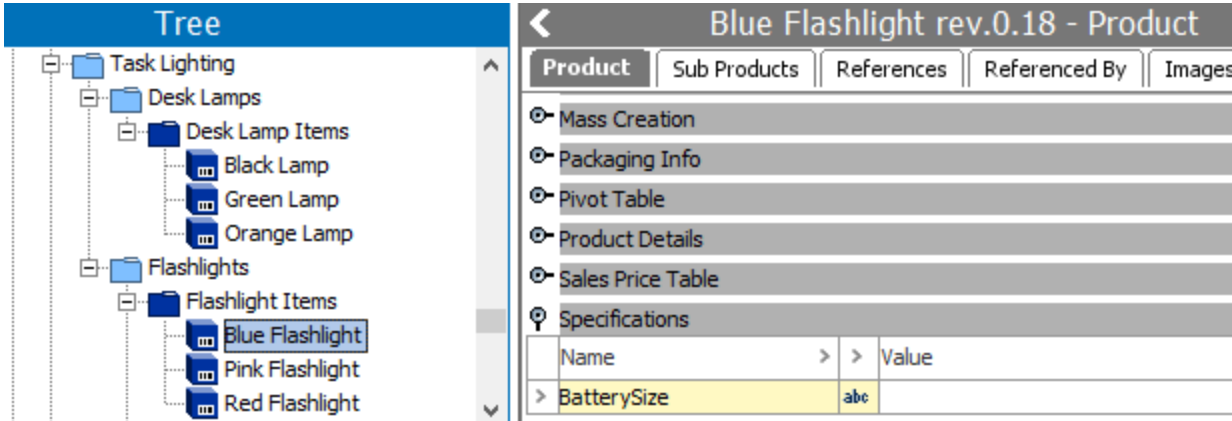
- 3. On the References or Referenced By tabs > 'Linked Attributes from Product Hierarchy' flipper> click the 'Link to Attribute' link to display the Select Attribute dialog.



4. Note the attribute group of the desired specification attribute, choose the attribute, and click the Select button. The specification attribute is displayed under the Attributes flipper.



5. Select an object in the linked PPH level, open the attribute group flipper identified previously, and verify that the linked attribute is available.



Note that other objects of the same type, but that are linked into a different PPH level, do not include the specification attribute (the flipper is absent).

Tree

Task Lighting

Desk Lamps

Desk Lamp Items

Black Lamp

Green Lamp

Orange Lamp

Flashlights

Flashlight Items

Blue Flashlight

Pink Flashlight

Red Flashlight

Orange Lamp rev.0.1 - Product

Product

Sub Products

References

Referenced By

Image

⊖ eClass Mapped Attributes

⊖ Manual Sequencing

⊖ Manual Sequencing Display

⊖ Mass Creation

⊖ Packaging Info

⊖ Pivot Table

⊖ Product Details

⊖ Sales Price Table

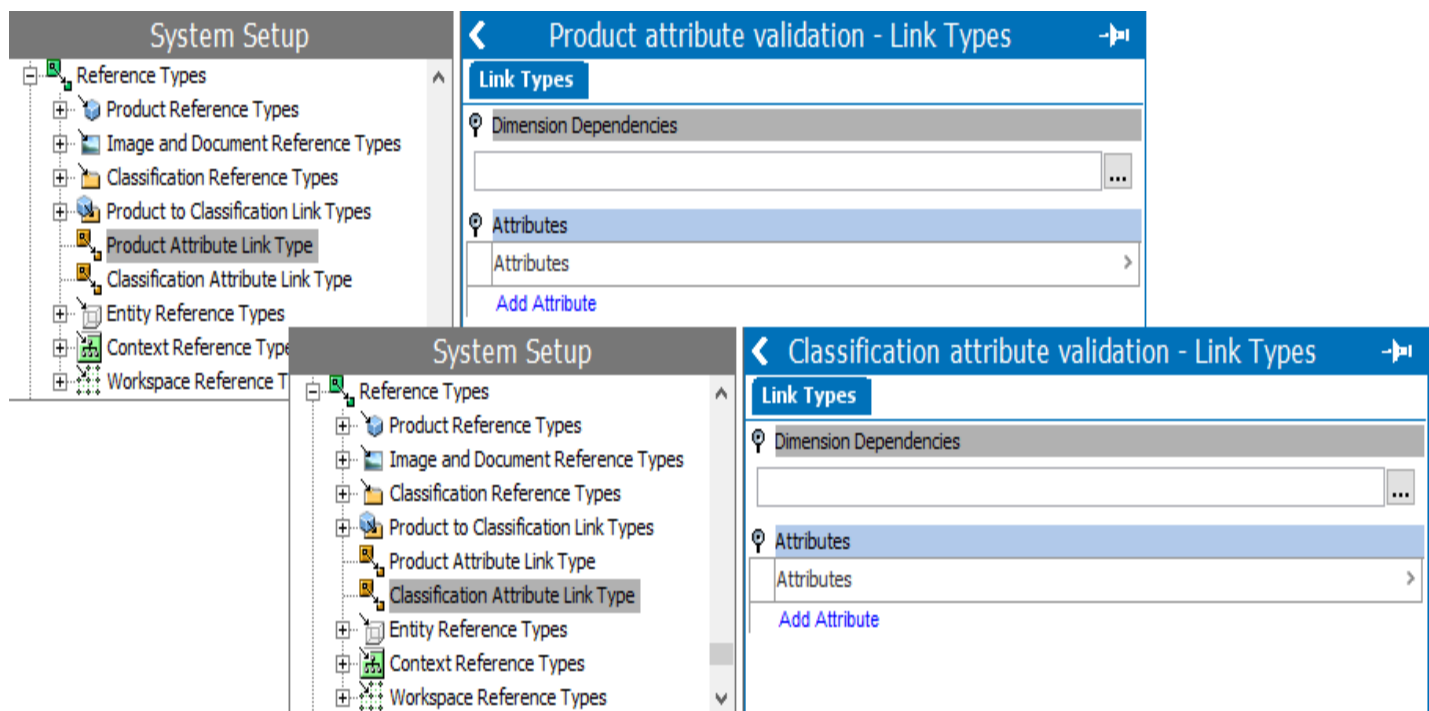
Setting Up an Attribute Link Type

The 'Product Attribute Link Type' and the 'Classification Attribute Link Type' are pre-configured. By default, linked attributes are displayed for the selected product hierarchy or classification hierarchy. If dimension dependency or metadata attributes are required, use the following steps to further configure the appropriate link type.

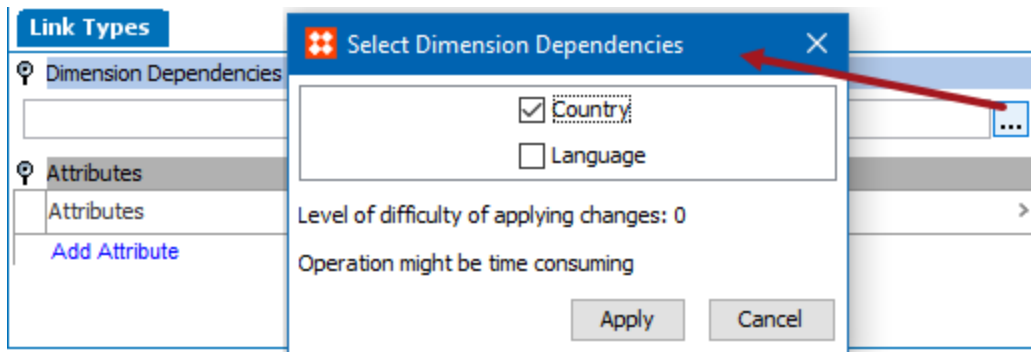
Only the default attribute link types can exist in STEP. Unlike references, user-created types are not allowed. The settings on the default attribute link types are applied to all 'Product Attribute Link Type' and 'Classification Attribute Link Type' used in the database.

Note: Setup is the same for both the Product Attribute Link Type and the Classification Attribute Link Type.

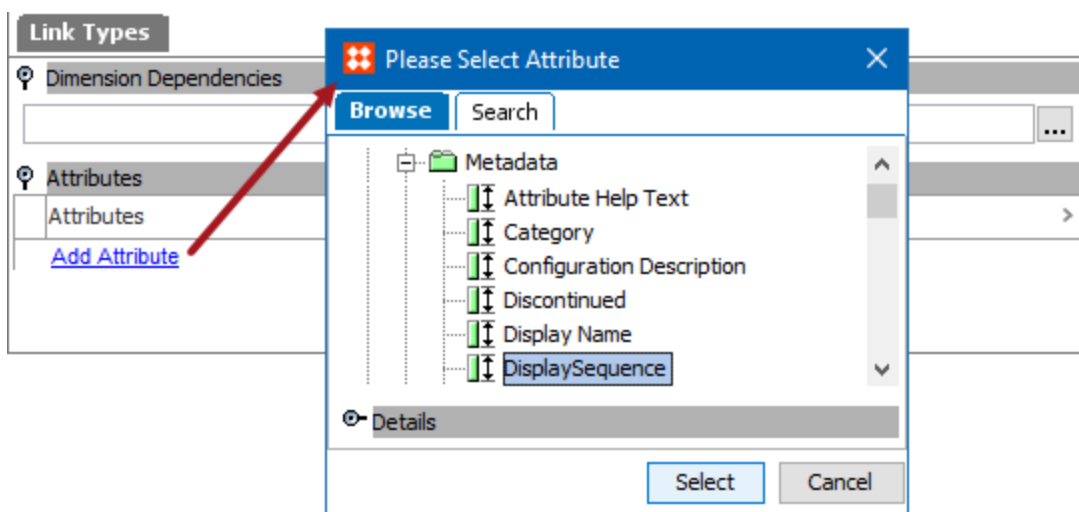
1. To set up an attribute link type for dimension dependencies or metadata attributes, in System Setup > Reference Types > click the desired attribute link type to display the editor.



2. If required, set the dimension dependencies on the Link Types tab, under the Dimension Dependencies flipper. Only one dimension is recommended if any, due to the increased complexity of managing more than one. Setting an attribute link type to be dimension dependent will allow variations of values on the linked specification attribute across dimension points, and by extension, contexts.



- Click the ellipsis button (...) to display the Select Dimension Dependencies dialog.
 - Choose the required dimension(s).
 - Click **Apply**.
3. If required, set metadata attributes on the Link Types tab, under the Attributes flipper. Specification attributes and groups cannot be added.



- Click the **Add Attribute** link to display the Select Attribute dialog.
 - Browse or search for the required description attribute and choose it.
 - Click **Select**.
4. The link type displays the selected dimension dependencies and metadata attributes.

Link Types

Dimension Dependencies

Country; ...

Attributes

Attributes >

> DisplaySequence

Add Attribute

If the same metadata attribute is set for both attribute links, the product's References tab shows the attribute in both the 'Linked Attributes from Product Hierarchy' and the 'Linked Attributes from Classification Hierarchy' flippers.

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Is

Linked Attributes from Product Hierarchy

DisplaySequence	>	ID	>	Name	>	Condition
>		Color		Color		
>		LongDescription		Long Description		

[Link to Attribute](#)

<

Linked Attributes from Classification Hierarchy

ID	>	Name	>	DisplaySequence	>	Show On Web
>	BuyItNowText	BuyItNowText US				Y
>	MarkedDown%	Marked Down				

5. Add metadata values as described in **Maintaining Attribute Link Metadata Values**.

Maintaining Attribute Link Metadata Values

When a metadata attribute has been added to the **Product Attribute Link Type**, it is displayed on the References or Referenced By tabs under the 'Linked Attributes from Product Hierarchy' flipper for all linked attributes. In the image below, the 'Show on Web' description attribute is metadata on the product attribute link.

Tree

Products

Apparel

Upper Body Wear

Head Wear

Hats and Caps

Hats and Caps Items

Industrial Hard Hat

Industrial Hard Hat rev.0.9 - Reference

ProductSub ProductsReferencesReferenced ByImages & DocumentsCommercialTablesCategories

Linked Attributes from Product Hierarchy

DisplaySequence	ID	Name	Condition	Product Variant Priority	Show On Web
9	ManufacturerWarranty	attrWarranty		10	Y
5	AvailabilityEnd	Availability End			
2	AvailabilityStart	Availability Start			

When a metadata attribute has been added to the **Classification Attribute Link Type**, it is displayed on the References or Referenced By tabs under the 'Linked Attributes from Classification Hierarchy' flipper for all linked attributes. In the image below, the 'Show on Web' description attribute is metadata on the classification attribute link.

Tree

Merchandising Hierarchy

Apparel

Sportswear

Sleepwear

T-shirts

Mens T PGW

Sportswear Clearance

Mens T PBG

Mens T PBG rev.0.8 - References

ProductSub ProductsReferencesReferenced ByImages & DocumentsCommercialTablesCategories

Linked Attributes from Product Hierarchy

Linked Attributes from Classification Hierarchy

ID	Name	DisplaySequence	Show On Web
BuyItNowText	BuyItNowText US		Y
MarkedDown%	Marked Down		

To add a new metadata attribute for an attribute link type, see the **Setting Up an Attribute Link Type**.

Values can be added for a metadata attribute on an attribute link type and be inherited down, or values can be written at the local level.

Inherited Metadata Values

A value being inherited is indicated by the small green arrow (↗) in the first column of the list. Inherited values must be edited at the parent level, and are displayed as read-only for all children.

Use inheritance when the value is the same from all or most of the products from the 'Inherited from' level and below.

Edit an Inherited Value

Note: Although the following images show the Product Attribute Link, the process is the same for Classification Attribute Link maintenance.

1. Click the link in the 'Inherited from' column to display the parent object where the value is being held.

Industrial Hard Hat rev.0.9 - References							
Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Tables	Category Profile
Linked Attributes from Product Hierarchy							
DisplaySequence	ID	Name	Show On Web	Attribute Groups	Mandatory	Inherited from	
9	ManufacturerWarranty	attrWarranty	Y	Buyer, View, Item Descrip...		Products	
5	AvailabilityEnd	Availability End		Buyer, Approve, Buyer, Vi...		Products	
2	AvailabilityStart	Availability Start		Buyer, Approve, Buyer, Vi...		Products	

2. Edit the value on the parent object. Also note that the 'Inherited from' column is not displayed for a parent object.

Tree		Products rev.0.40 - References						
Products		Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Tables
Apparel		Linked Attributes from Product Hierarchy						
Upper Body Wear		DisplaySequence	ID	Name	Show On Web	Attribute Groups	Mandatory	
Head Wear		9	ManufacturerWarranty	attrWarranty	N	Buyer, View, Item Description I...		
Hats and Caps		5	AvailabilityEnd	Availability End		Buyer, Approve, Buyer, View, ...		
Hats and Caps Items		2	AvailabilityStart	Availability Start		Buyer, Approve, Buyer, View, ...		
Industrial Hard Hat								

All children display the new value.

Industrial Hard Hat rev.0.9 - References							
Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Tables	Category Profile
Linked Attributes from Product Hierarchy							
DisplaySequence	ID	Name	Show On Web	Attribute Groups	Mandatory	Inherited from	
9	ManufacturerWarranty	attrWarranty	N	Buyer, View, Item Descrip...		Products	
5	AvailabilityEnd	Availability End		Buyer, Approve, Buyer, Vi...		Products	
2	AvailabilityStart	Availability Start		Buyer, Approve, Buyer, Vi...		Products	

Local Metadata Values

A local value overwrites any inherited value from a higher level and can be edited. It is indicated by the absence of the small green arrow (>) in the first column of the list and the absence of the data in the 'Inherited from' column.

Add a Local Value

Note: Although the following images show the Product Attribute Link, the process is the same for Classification Attribute Link maintenance.

Click the arrow button on the row to display a menu and select **Remove Link to Attribute**. Other display options are also available on this menu.

Industrial Hard Hat rev.0.10 - References

Product Sub Products **References** Referenced By Images & Documents Commercial Tables Category Profile Proof View

Linked Attributes from Product Hierarchy

DisplaySequence	ID	Name	Show On Web	Attribute Groups	Mandatory	Inherited from
9	ManufacturerWarranty	attrWarranty	N	Buyer, View, Item Descrip...	☐	
		Availability End		Buyer, Approve, Buyer, Vi...	☐	Head Wear
		Availability Start		Buyer, Approve, Buyer, Vi...	☐	Products
		Use Unit of Measu		Buyer, View, Item Descrip...	☐	Products
		and Name		Buyer, View, Image Revie...	☐	Products
		and Owner		Buyer, View, Item Brand I...	☐	Products
		st		Buyer, Approve, Buyer, Vi...	☐	Products
		st Effective Dat		Buyer, Approve, Buyer, Vi...	☐	Products
		st Expiration Da		Buyer, Approve, Buyer, Vi...	☐	Products

Hide

Show All Columns

Rotate Table

Link to Attribute Ctrl+Plus

Remove Link to Attribute Ctrl+Minus

Create Local Link

The inherited value is displayed again.

Industrial Hard Hat rev.0.9 - References

Product Sub Products **References** Referenced By Images & Documents Commercial Tables Category Profile Proof View

Linked Attributes from Product Hierarchy

DisplaySequence	ID	Name	Show On Web	Attribute Groups	Mandatory	Inherited from
9	ManufacturerWarranty	attrWarranty	Y	Buyer, View, Item Descrip...	☐	Products
5	AvailabilityEnd	Availability End		Buyer, Approve, Buyer, Vi...	☐	Products
2	AvailabilityStart	Availability Start		Buyer, Approve, Buyer, Vi...	☐	Products

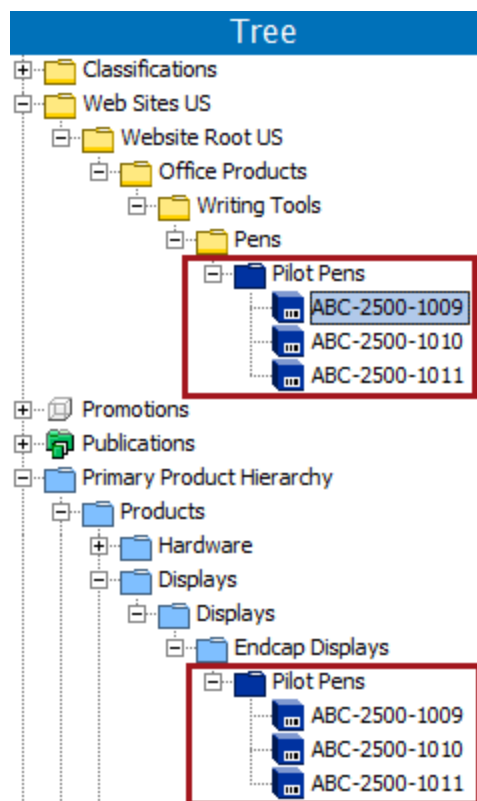
Product to Classification Link Types

A Product to Classification Link Type defines a set of rules for linking products into classifications. This link type displays the products under the linked classification folder as a parent / child relationship, which is how it differs from a classification reference type where the source is a product and the target is a classification.

Products should only be added to the PPH (primary product hierarchy) once. A product displayed within a classification folder is a shortcut to the actual object in the primary product hierarchy. Regardless of the number of classification folders that display the product, they all have the same ID and are the same object. Edits made to the product are displayed in all places that display that ID.

Note: Deleting an object from a classification folder deletes every instance of the object. To remove a product from a classification folder, while continuing to display it in the PPH, only remove the product to classification link from the product's References or Referenced By tabs.

For example, in the image below, the pen products classified into 'Writing Tools' are the same products that are displayed in the primary product hierarchy. Since a Product to Classification Link Type controls which products are valid to be linked into the 'Writing Tools' classification, it can be configured to allow pens to be linked into 'Writing Tools,' and also prevent products like hammers from being linked.



Consider the following elements when setting up a product to classification link type:

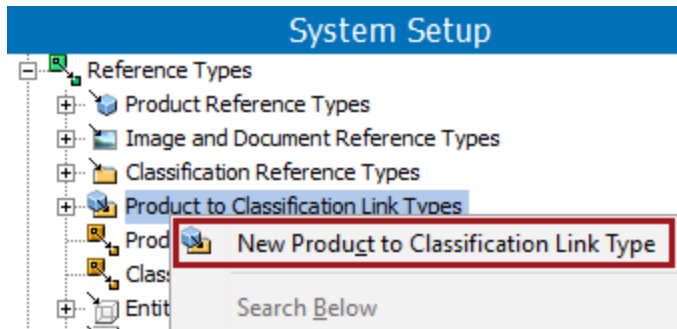
- Adding a link to an attribute group allows grouping of references and links, customized views, and added security via user group privileges. For more information, see **Attribute Groups for Reference and Link Types**.
- Dimension dependency determines if the link is displayed in a context, based on the selected dimension point. Metadata is not shared between dimension points, so values must be added for each link. For more information, see **Dimension Dependent Reference and Link Types**.
- Metadata attributes on a link allow description attributes to supply additional information about the link itself. For more information, see **Metadata Attributes on Reference and Link Types**.
- Dimension dependent links are displayed based on the Visibility setting. For more information, see **Visibility of Reference and Link Types**.

To create a product to classification link type, follow the steps outlined in **Creating a Product to Classification Link Type**.

Creating a Product to Classification Link Type

Adding a product-to-classification link type allows a product to be associated with a classification.

1. In System Setup > Reference Types > select **Product to Classification Link Type** > right-click, and choose the **New** option from the menu.



For information on the attribute link types, see the **Product Attribute Link Type** topic or the **Classification Attribute Link Type** topic.

For information on reference types, see the **Reference Types** topic.

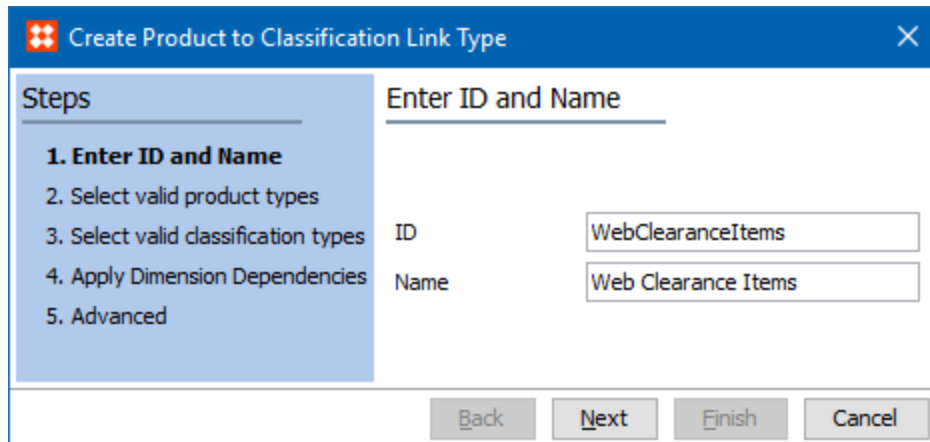
2. The Create Product to Classification Link Type wizard displays and includes the following steps:

- **Enter ID and Name** allows you to identify the link.
- **Select Valid Product Types** allows you to choose the types of products valid for this link.
- **Select Valid Classification Types** allows you to choose the alternate classifications valid for this link.
- **Apply Dimension Dependencies** allows you to define the dimension points that are affected by this link.

- **Advanced** allows you to configure multiple references, set externally maintained, mandatory, and inheritance for the link.

Once the reference type exists, create a reference between the two objects as outlined in **Creating a Product to Classification Link**.

Product to Classification Link Type - Enter ID and Name



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

Enter ID and Name

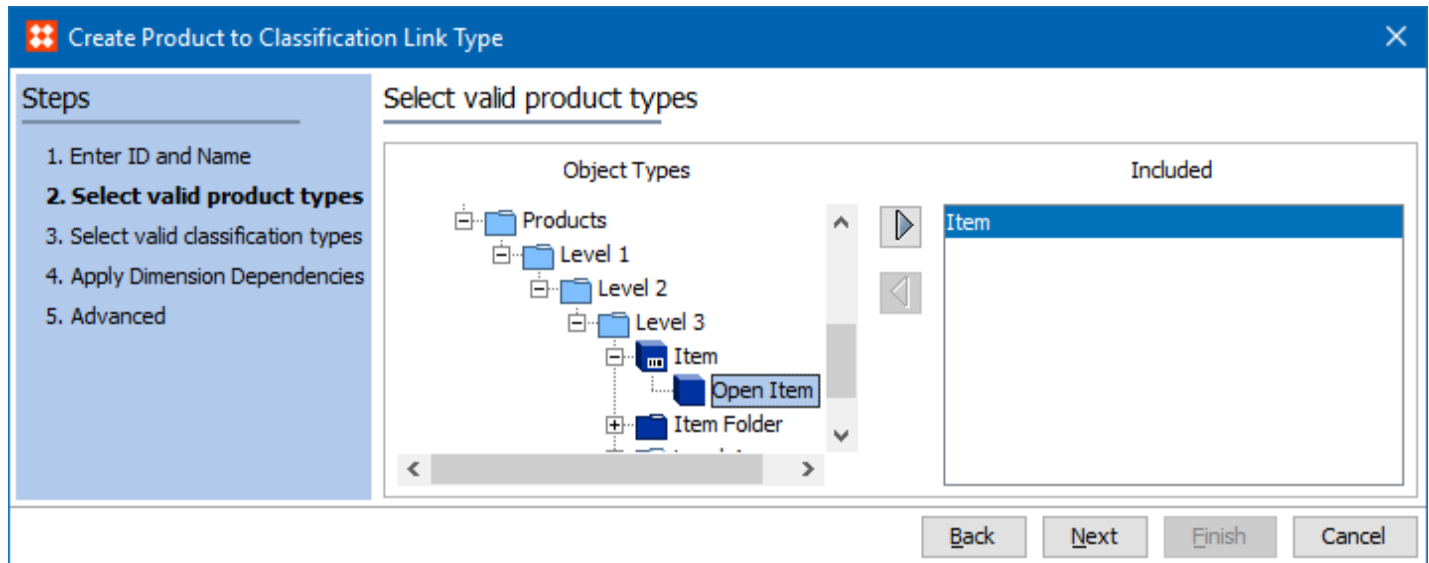
ID: WebClearanceItems

Name: Web Clearance Items

Back Next Finish Cancel

1. Enter an **ID** for the link. Common setup is to use no spaces or punctuation.
2. Enter a **Name** for the link. Common setup is to repeat the ID with added spaces for readability.
3. Click the **Next** button to display **Product to Classification Link Type - Select Valid Product Types**.

Product to Classification Link Type - Select Valid Product Types

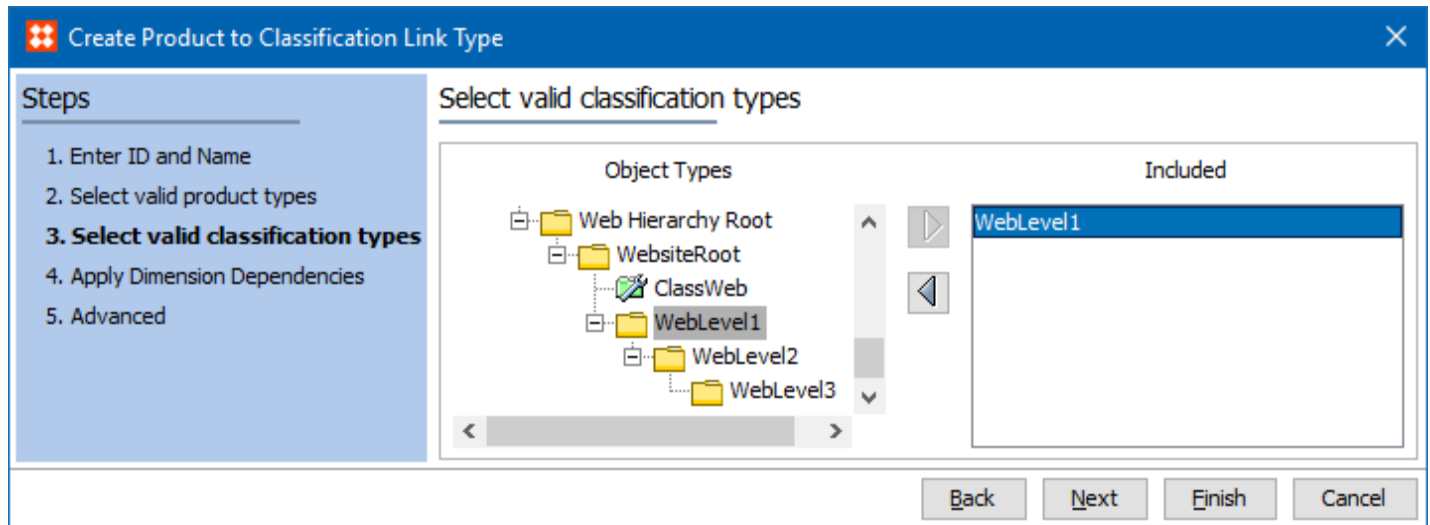


1. Use the displayed hierarchy to browse to and select one or more product object types of the reference. This is where the reference starts. For more information, see **Direction of a Reference**. Selecting an item from the Object Types pane enables the right arrow button.
2. Click the right arrow button () to move each required selected object type to the Included pane. All objects in the Included pane will be valid for the Product to Classification Link type being created.

Note: To remove an object type from the Included pane, first select the Included item, and then click the left arrow button ().

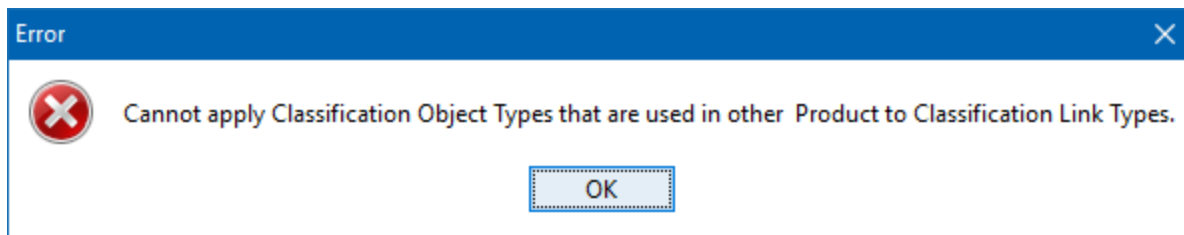
3. Click the **Next** button to display **Product to Classification Link Type - Select Valid Classification Types**.

Product to Classification Link Type - Select Valid Classification Types



1. Use the displayed hierarchy to browse to and select one or more classification object types of the reference. This is where the reference ends. For more information, see **Direction of a Reference**. Selecting an item from the Object Types pane enables the right arrow button.

Each classification object type can only be used in one Product to Classification Link Type. A warning is displayed if you choose a classification that has already been selected for a product -to-classification link type.



2. Click the right arrow button (▶) to move the selected object type to the Included pane. All objects in the Included pane will be valid for the Product to Classification Link type being created.

Note: To remove an object type from the Included pane, first select the Included item, and then click the left arrow button (◀).

3. Click the **Next** button to display **Product to Classification Link Type - Apply Dimension Dependencies**.

Product to Classification Link Type - Apply Dimension Dependencies

The screenshot shows a software dialog box titled "Create Product to Classification Link Type". On the left, a "Steps" sidebar lists five steps: "1. Enter ID and Name", "2. Select valid product types", "3. Select valid classification types", "4. Apply Dimension Dependencies" (which is highlighted in blue), and "5. Advanced". The main area of the dialog is titled "Apply Dimension Dependencies" and contains two checkboxes: "Country" (unchecked) and "Language" (checked). At the bottom of the dialog, there are four buttons: "Back", "Next", "Finish", and "Cancel".

1. If the link type should be dimension dependent, select a dimension checkbox. Only one dimension is allowed for a link type. If multiple dimension dependencies are needed, create a separate link type.

For more information about dimension dependencies, see the **Dimension Dependent Reference and Link Types** topic.

2. Click the **Next** button to display **Product to Classification Link Type - Advanced**.

Product to Classification Link Type - Advanced

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

Sub Products Inheritance Settings

No Inheritance

No Inheritance

No Inheritance of Tables and Specification Attributes

Inherit Links and Tables and Specification Attributes

Accumulative inheritance without Tables and Specification Attributes

Accumulative inheritance with Tables and Specification Attributes

Back Next Finish Cancel

1. **Allow Multiple Links** determines if more than one of this link type is allowed on a product and is indicated by the  icon.
 - **Checked** means the link type can be used to link the same product of the valid product types to more than one classification of the valid classification types.
 - **Unchecked** means the link type is restricted to a single link a product of the valid product types to one classification of the valid classification types.
2. **Externally Maintained** determines if adding or modifying the link affects the Approved parameter on the product. When using an externally maintained link between product and classification, the link will automatically be included in all workspaces and the product will remain in its current approval status when the link is modified.
 - **Checked** means adding a link from a product to a classification will not require approved to display in the Approved workspace.

- **Unchecked** means the link type is controlled by revision control and adding a link from a product and/or changing values on attributes applied to the link on the product will required approval to display in the Approved workspace.

Note: The product's approved status is reset (requires approval to display in the Approved workspace) if metadata attribute values are changed on the link and the metadata attributes are not externally maintained.

3. **Mandatory** determines if the link type is required for approval of the product and is indicated by the  icon. The approval check for mandatory 'Product to Classification Link Type' is determined by the ownership of the link, which is set on the classification object types. For more information, see **Owns Product Links on Alternate Classifications Object Type**.
 - **Checked** means the link type is required from a product to a classification before the product can be approved.
 - **Unchecked** means the link type for a product to a classification is optional and does not affect approval of the product.
4. **Sub Products Inheritance Settings** determines if and how links are displayed on valid child objects below the product with respect to the Validity settings, and is indicated by the  icon. For an illustration of the effects of each setting, see **Inheritance Example for a Product to Classification Link Type**.
 - **No Inheritance** means the product to classification link is not inherited to leaf nodes below the linked product. Specification attributes and table definitions applied on the classification are not inherited to any product linked to the classification.
 - **No Inheritance of Tables and Specification Attributes** means the link from product to classification is inherited to all leaf nodes below the product. Specification attributes and tables applied to the classification are not inherited to all products below the classification.
 - **Inherit Links and Tables and Specification Attributes** means the link from product to classification is inherited to all products below the product. Specification attributes and tables applied to the classification are inherited to all products below the classification. Additionally, if multiple links are allowed, and a new link of this type is added to a sub product, the inherited link is removed and only the local link remains.
 - **Accumulative Inheritance without Tables and Specification Attributes** (requires that 'Allow multiple links' is checked) means the link from product to classification will be inherited to all leaf nodes below the product. Additionally, leaf nodes can be linked to valid classifications with collective additions to the existing targets. Specification attributes and tables applied to the classification are excluded from the inheritance to any products below the classification. When a new link of this type is added to a sub product, the inherited link remains and the local link is added to the list.
 - **Accumulative Inheritance with Tables and Specification Attributes** (requires that 'Allow multiple links' is checked) means the link from product to classification will be inherited to all leaf nodes below the product. Additionally, leaf nodes can be linked to valid classifications with collective additions to the existing targets. Specification attributes and tables applied to the classification are also inherited. When a new link of this type is added to a sub product, the inherited link remains and the local link is added to the list.

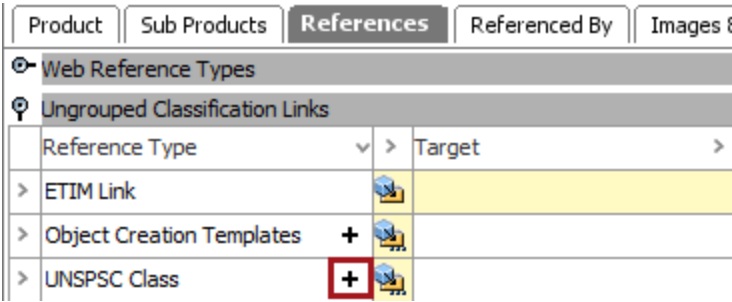
5. Click the **Finish** button to create the link type.
6. To apply the new link type to an object, see **Creating a Product to Classification Link**.

Creating a Product to Classification Link

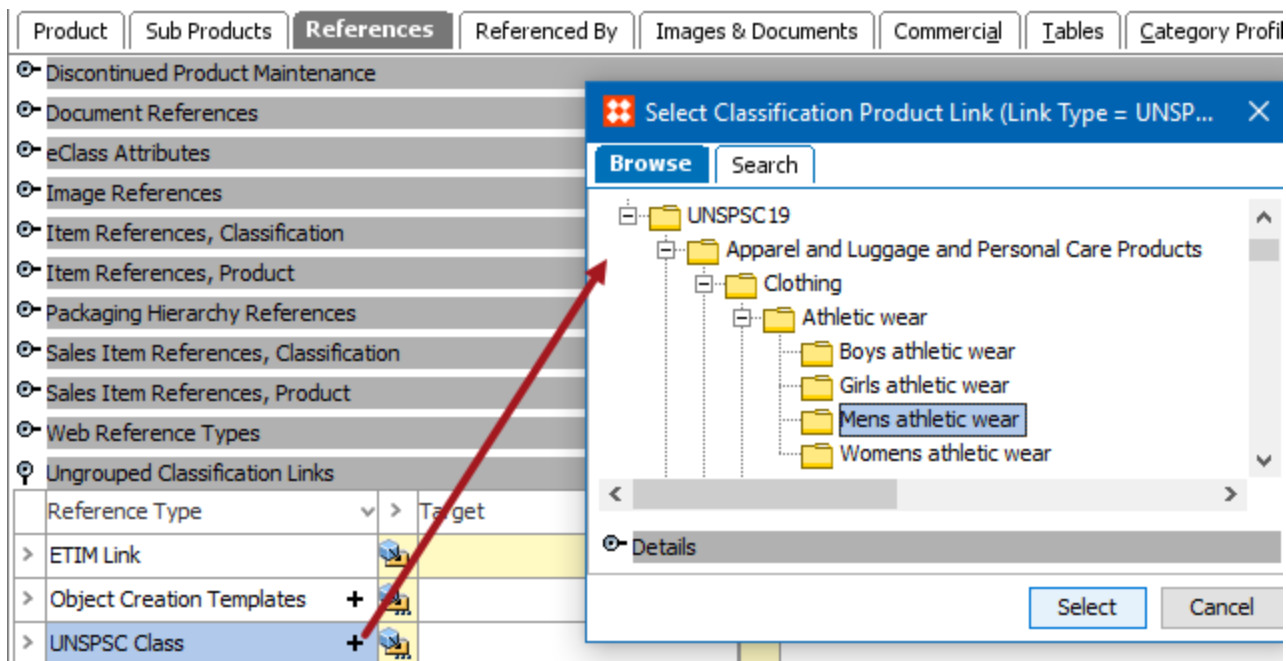
After creating a product to classification link type, you can add that type on an object's References or Referenced By tabs.

To maintain data on the References and Referenced By tabs, see the **References Tab** topic and the **Referenced By Tab** topic in **Getting Started / User Guide**.

1. To add a product to classification link to either the **References** or **Referenced By** tab of an object, use one of the following ways to find the product to classification link type in a flipper:
 - If the link type has been added to an attribute group, it appears under a flipper that is named the same as the attribute group.
 - If the link type is not in an attribute group, it appears under an 'Ungrouped Classification Links' flipper, as shown in the image in the next step.
2. Click the plus sign button in the desired link field. If more than one link is allowed for the link type, the plus sign button continues to display after the first link is added, otherwise it is removed.



3. In the Select Classification Product Link dialog, find the object that is to be linked, and click the Select button. If you know the target ID / Name of the object, use the Search tab to find it quickly using the typeahead functionality to display the options as a dropdown.



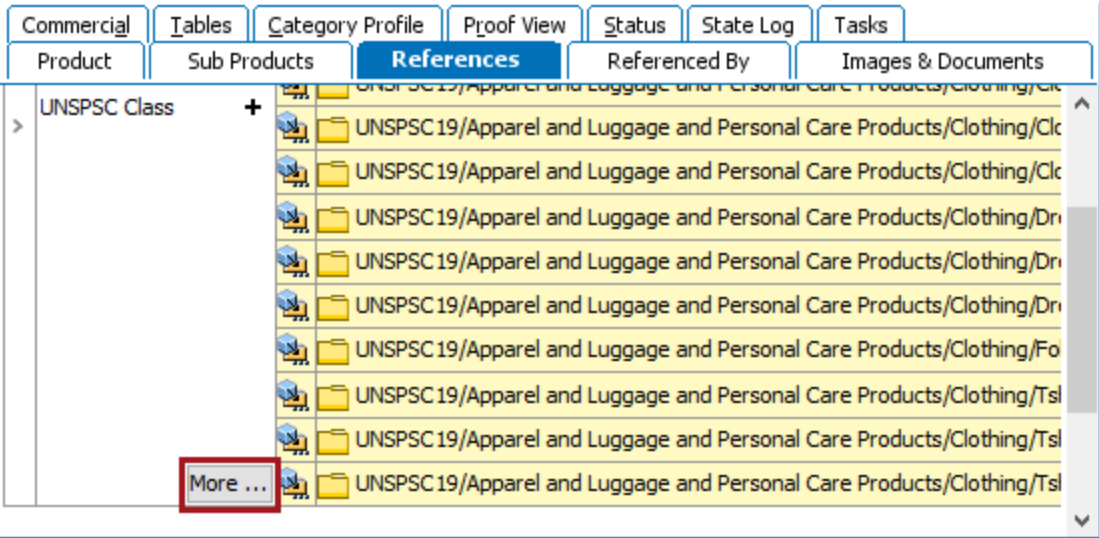
Once selected, the object displays as an added link.

Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Tables	Category Profile	Proof View	Status
Ungrouped Classification Links									
Reference Type		Target							
> ETIM Link									
> Object Creation Templates	+								
> UNSPSC Class	+	UNSPSC 19/Apparel and Luggage and Personal Care Products/Clothing/Athletic wear/Mens athletic wear							

- To remove a link, click the X button.

Displaying a large number of links

If more than 50 links of the same type are added, a button labeled 'More...' ([More ...](#)) displays, allowing users to expand the list, and view 50 additional links.

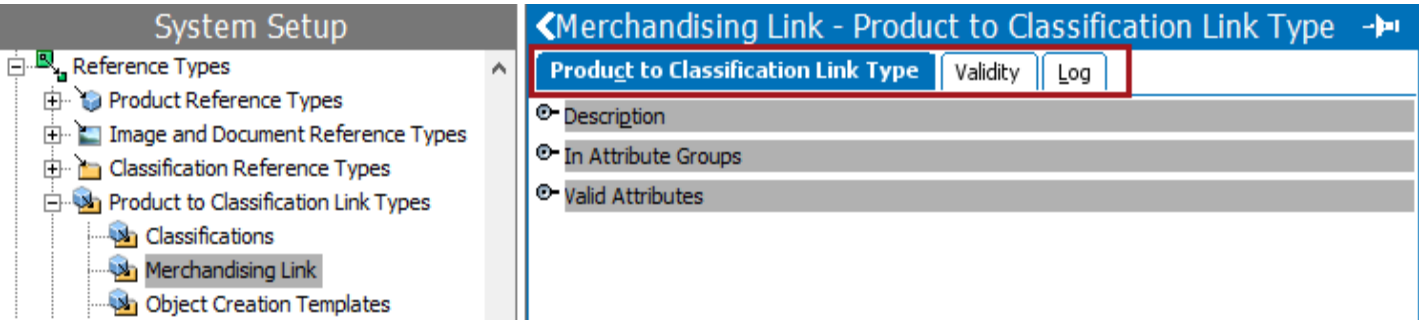


Repeatedly clicking the button doubles the number of links shown each time, until the full set is displayed. Note that if new links are added, they display on the list immediately indicating that they were added properly, though normal functionality will resume if the user presses the 'More...' button again. This means that items may become hidden until the list is fully expanded.

After the list has been fully expanded for the given link type, it stays expanded until the user refreshes the page, or clicks away to another object in workbench.

Maintaining a Product to Classification Link Type

Select a Product to Classification Link Type (🔗) to display the Product to Classification Link Type editor and make updates to the link type settings. Each tab on the editor and the parameters are defined below.



Product to Classification Link Type Tab

This tab holds basic information within the Description, In Attribute Groups, and Valid Attribute flippers.

Description Flipper

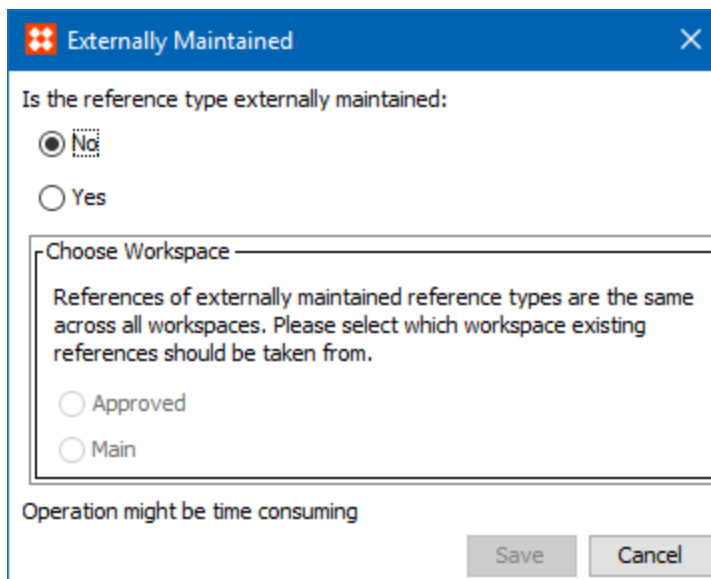
The Description flipper includes basic information to identify the link. This data is originally set up in the Product to Classification Link Type wizard, see **Creating a Product to Classification Link Type** documentation.

Product to Classification Link Type			Validity	Log
Description				
	Name	>	>	Value
>	ID			SupplierLink
>	Name			Supplier Link
>	Last edited by			2016-11-21 06:41:41.0 by USERL
>	Externally Maintained			No
>	Dimension Dependencies			Language;
>	Allow multiple links			Yes
>	Mandatory			No
>	Inheritance of Links			Inherited
>	Inheritance of Specification Attributes			Yes
>	Ignore LOV Filter definitions on inherited Attributes for links			No
In Attribute Groups				
Valid Attributes				

All fields except the ID and 'Last edited by' can be edited. The method required to edit a parameter in the link editor is determined by the parameter type:

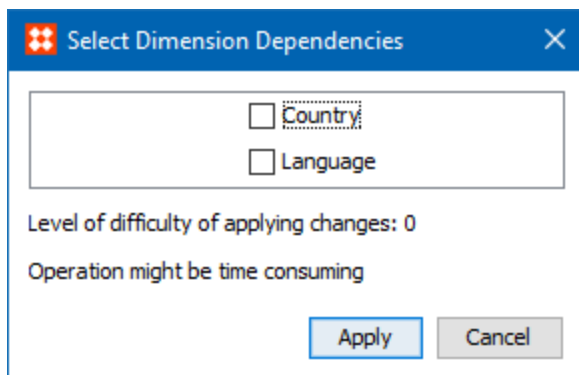
- For **Name**, click into the text box and make the desired changes. The ID and name parameters are discussed in the **Product to Classification Link Type - Enter ID and Name** documentation.
- For **Externally Maintained**, double-click to display a dialog. The externally maintained parameter is discussed in the **Product to Classification Link Type - Advanced** documentation.

If a link type is edited to be Externally Maintained = Yes, and the link type has been used to link objects, you must select a workspace. All existing links will be synchronized across all workspaces based on the value in the selected workspace.



The 'Externally Maintained' dialog box has a blue title bar with the Stibo logo and a close button. The main content area contains the text 'Is the reference type externally maintained:' followed by two radio buttons: 'No' (selected) and 'Yes'. Below this is a section titled 'Choose Workspace' with a text box containing the instruction: 'References of externally maintained reference types are the same across all workspaces. Please select which workspace existing references should be taken from.' This section includes two radio buttons: 'Approved' and 'Main'. At the bottom, there is a warning 'Operation might be time consuming' and two buttons: 'Save' and 'Cancel'.

- For **Dimension Dependencies**, double-click to display an ellipsis button. Click the ellipsis button (...) to display a dialog that includes the dimension points on your system. Only one dimension point is allowed. If more are needed, create a separate link type. The dimension dependencies parameter is set in the **Product to Classification Link Type - Apply Dimension Dependencies** documentation.



The 'Select Dimension Dependencies' dialog box has a blue title bar with the Stibo logo and a close button. The main content area contains a list box with two items: 'Country' and 'Language', each with an unchecked checkbox. Below the list box, it says 'Level of difficulty of applying changes: 0' and 'Operation might be time consuming'. At the bottom, there are two buttons: 'Apply' and 'Cancel'.

- For **Allow multiple links**, click the parameter to display the dropdown list of options. Three (3) small dots displayed below the icon indicates multiple links are allowed (🔴). This parameter is discussed in the **Product to Classification Link Type - Advanced** documentation.
- For **Mandatory**, click the parameter to display the dropdown list of options. The red check mark displayed below the icon indicates the links is required for full approval (🔴). This parameter is discussed in the **Product to Classification Link Type - Advanced** documentation.

The approval check for mandatory product to classification links is determined by the ownership of the link. The ownership of a link is set on the classification object type editor using the 'Owns Product Links' parameter. For more information, see **Owns Product Links on Alternate Classifications Object Type**.

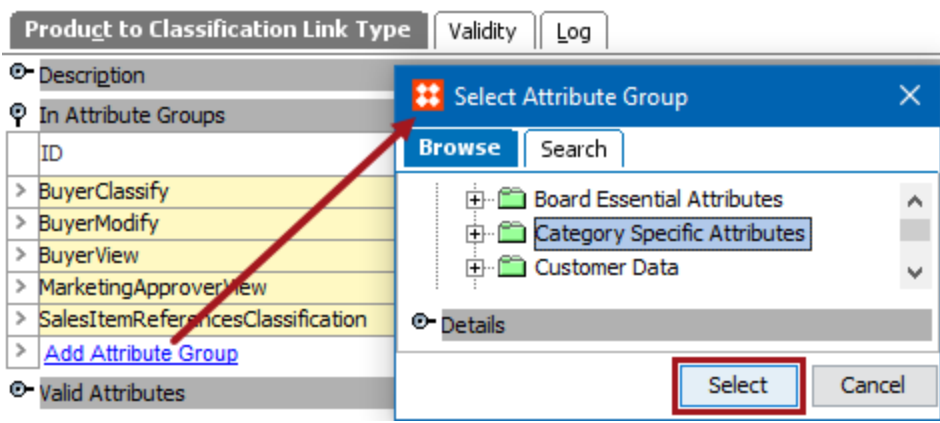
- For **Inheritance of Links** and **Inheritance of Specification Attributes**, click the parameter to display the dropdown list of options. The small green down arrow displayed above the icon indicates the value is inherited (🔴). This parameter is discussed in the **Product to Classification Link Type - Advanced** documentation. The inheritance of tables is indicated by the 'Inheritance of Specification Attribute' parameter setting.
- For **Ignore LOV Filter definitions on inherited Attributes for links**, click the parameter to display the dropdown list of options. This parameter determines if the LOV Filter definition is inherited when the attributes on the link are inherited and is only available when Inheritance of Specification Attributes = Yes. This option defaults to No, and must be manually changed to Yes in the link editor, if required. For more information on LOV filtering, see **Hierarchical Filters on Attributes**.

In Attribute Groups Flipper

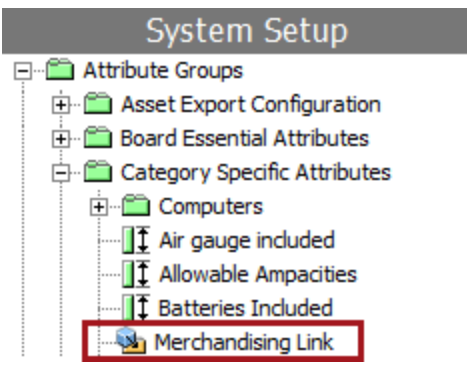
The 'In Attribute Groups' flipper includes all attribute groups that include the link. Adding a link type to one or more attribute groups determines the label of the flipper that displays the link in an object editor, enables you to set up user privileges to restrict approval of the link type, and to define a customized view of the link types in the group. For more information, see **Attribute Groups for Reference and Link Types** documentation.

Product to Classification Link Type		Validity	Log
Description			
In Attribute Groups			
ID	>	Name	
> BuyerClassify		Buyer, Classify	
> BuyerModify		Buyer, Modify	
> BuyerView		Buyer, View	
> MarketingApproverView		Marketing Approver, View	
>	Add Attribute Group		
Valid Attributes			

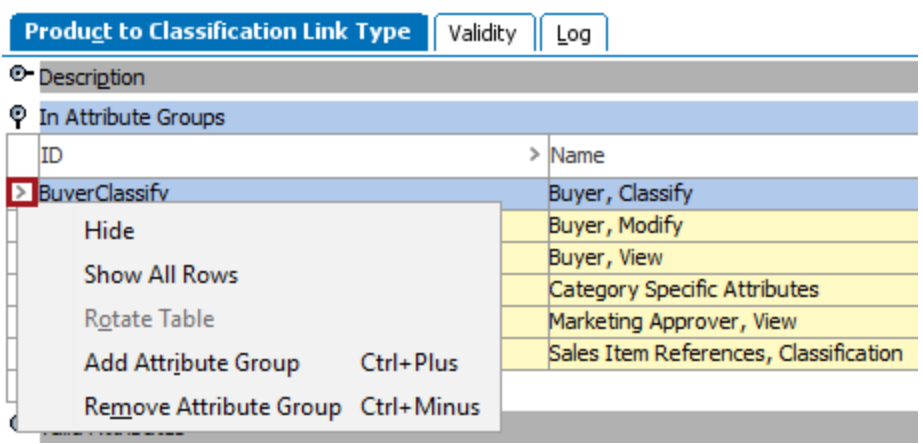
- To add an attribute group, click the **Add Attribute Group** link to display the Select Attribute Group dialog. Choose a group and click the Select button to add the link type to the attribute group.



Once added to an attribute group, the link type is also displayed within the System Setup > Attribute Groups node.



- To modify the attribute groups included or displayed, right-click the arrow to the left of the group ID and select an option from the menu.



Valid Attributes Flipper

Metadata (description) attributes allow information to be captured on the link itself, to determine when the target object should be used, or to provide additional information about the connection between the objects.

Product to Classification Link Type		Validity	Log
Description			
In Attribute Groups			
Valid Attributes			
ID	>	Name	>
>	Completeness Score	Completeness Score	
>	URL	URL	
>	Add Attribute		

To modify the attributes included or displayed, right-click the arrow to the left of the ID and select an option from the menu.

Product to Classification Link Type		Validity	Log
Description			
In Attribute Groups			
Valid Attributes			
ID	>	Name	>
>	Completeness Score	Completeness Score	
>		URL	
Hide Show All Rows Rotate Table Add Attribute Ctrl+Plus Remove Attribute Ctrl+Minus			

For more information, see **Metadata Attributes on Reference and Link Types** documentation.

Validity Tab

The Validity tab stores the source product types and target classification types allowed on the product to classification link type.

Product to Classification Link Type	Validity	Log
Valid Product Types		
ID	>	Name >
> Item		Item
> ItemFamily		Item Family
> ItemFolder		Item Folder
>		Modify Product Types
Valid Classification Types		
ID	>	Name >
> MerchandisingClass		MerchandisingClass
> SubDepartment		SubDepartment
>		Modify or Move Classification Types

- For **Valid Product Types**, any number of product object types can be added. Click the 'Modify Product Types' link to add or remove a product object type. When an object type is removed from the list, if Product to Classification Link of this type already exists for the product object type, the links are deleted. This parameter is discussed in the **Product to Classification Link Type - Select Valid Product Types** documentation.

To modify the object types included or displayed, right-click the arrow to the left of the ID and select an option from the menu.

Product to Classification Link Type	Validity	Log
Valid Product Types		
ID	>	Name
> Item		Item
		Item Family
		Item Folder
Hide Show All Rows Rotate Table Modify Product Types Ctrl+Plus Remove Row(s) Ctrl+Minus		

- For **Valid Classification Types**, any number of classification object types can be added, however, the same object type cannot be added to more than one product to classification link type. Click the 'Modify Product Types' link to add or remove a classification object type. This parameter is discussed in the **Product to Classification Link Type - Select Valid Classification Types** documentation.

To modify the object types included or displayed, right-click the arrow to the left of the ID and select an option from the menu.

Product to Classification Link Type **Validity** Log

Valid Product Types

Valid Classification Types

ID	Name
MerchandisingClass	MerchandisingClass

Hide

Show All Rows

Rotate Table

Modify or Move Classification Types Ctrl+Plus

Remove Row(s) Ctrl+Minus

Log Tab

The Log tab stores the information about the changes that have taken place on the product to classification link type since it was created.

Product to Classification Link Type Validity **Log**

Showing page 1 of 1

2016-04-11 09:32:51 'USER2': Included in package CPACK_11Apr2016
2016-04-11 09:36:30 'USER2': Excluded from package CPACK_11Apr2016
2016-05-06 13:43:27 'USER': Property modified ItemFolder added to valid source types
2016-05-06 13:45:47 'USER': Property modified Item added to valid source types
2016-05-20 16:04:20 'USER7': Property modified ItemFamily added to valid source types
2016-07-13 09:52:33 'USER1': Property modified std.AdaptorType added to valid source types
2016-07-26 15:20:27 'USER6': Included in package SampleWF
2016-07-26 15:20:52 'USER6': Sealed with 'SampleWF' (cpk-e4e39b10-2cf4-45a2-a598-de81d326eb7a) from doc-dev
2016-09-08 12:17:14 'USER6': Sealed with 'ChangePack1' (6d21091c-36b9-4ae7-ba46-3f4e7f6753ac) from doc-dev
2016-11-21 07:56:59 'USERY': Property modified MerchandisingHierarchyRoot added to valid target types
2017-03-15 15:14:05 'USERJ': Property modified Make reference type multi-valued
2017-03-15 15:16:20 'USERJ': Property modified SubDepartment added to valid target types

First Page

Previous Page

Next Page

Last Page

Download...

As additional log entries become available and the page is filled, the pagination buttons are enabled. Additionally, you can download the complete log as a zipped text file.

Attribute Groups for Reference and Link Types

Adding a Reference or Product to Classification Link type to an attribute group provides easy identification, a customized view, and control of approvals via user privileges.

Easy Identification

The name of the attribute group is used as the name of the flipper that displays the reference or link within the object editor. This provides a logical grouping for the references or links.

System Setup

The screenshot shows the 'System Setup' interface. On the left, under 'Attribute Groups', the 'Display' group is expanded, showing 'Web Reference Types' and 'Web'. On the right, the 'Tree' view shows 'Desk Lamps' > 'Desk Lamp Items' containing 'Black Lamp', 'Green Lamp', and 'Orange Lamp'. A red arrow points from 'Web Reference Types' to the 'References' tab of the 'Black Lamp rev.0.1 - References' flipper. Another red arrow points from 'Web' to the 'Web' reference entry in the flipper's table.

Black Lamp rev.0.1 - References				
Product	Sub Products	References	Referenced By	Images & Documents
Web Reference Types				
Reference Type		Target		
Web	+			

Otherwise, the reference or link displays in a flipper with the 'Ungrouped' label.

The screenshot shows the 'Tree' view with 'Desk Lamps' > 'Desk Lamp Items' containing 'Black Lamp', 'Green Lamp', and 'Orange Lamp'. The flipper 'Black Lamp rev.0.1 - References' is open, showing the 'References' tab. A red box highlights the 'Ungrouped Classification Links' section at the top of the flipper's table.

Black Lamp rev.0.1 - References				
Product	Sub Products	References	Referenced By	Images & Documents
Ungrouped Classification Links				
Reference Type		Target		
Basic Products	+	Web Sites US/Website Root US/Office Products		

Customized View

A customized view can be defined on the attribute group of all reference or link types in the group to display additional information on the reference or link. For more information, see the **Creating a Customized View** section in the **Attribute Groups** topic of the **System Setup / Super User Guide** documentation.

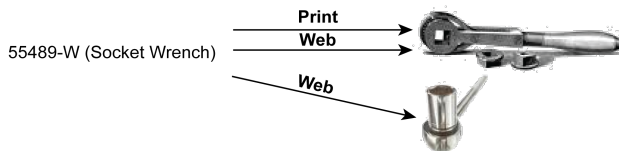
The screenshot shows the 'Tree' view with 'Desk Lamps' > 'Desk Lamp Items' containing 'Black Lamp', 'Green Lamp', and 'Orange Lamp'. The flipper 'Black Lamp rev.0.1 - References' is open, showing the 'References' tab. A red box highlights the 'Image References' section at the top of the flipper's table. Below it, a row for 'Brand Name Logo' is shown with a custom view displaying a logo, the name 'Acme', and other details.

Black Lamp rev.0.1 - References							
Product	Sub Products	References	Referenced By	Images & Documents	Commercial	Images & Documents	Images & Documents
Image References							
Reference Type		Target	Thumbnail	ImagePriority	ID	Name	
Brand Name Logo		Acme			168241	Acme	

Control of Approvals via User Privileges

Perhaps most importantly, an attribute group allows you to control user privileges for approval of the reference or link. This means that approval of specific types of references and links is based on the user's access to the attribute group, so that users will only be allowed to approve references or links of the types in the attribute group.

For example, consider a user who should only be allowed to approve references used for the web. As shown below, approving Product 55489-W (Socket Wrench) should also allow approving the images used for web.



To implement this, the reference type named 'Web' is added to an attribute group named 'Web Reference Types.'

A reference or link type can be added to as many attribute groups as needed by different users with different privileges.

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 set up to include the 'Approve Reference' privilege for reference types in the attribute group 'Web Reference Types.' For more information about setting up privileges, see **Action Sets** in the **System Setup / Super User Guide** documentation.

Dimension Dependent Reference and Link Types

A Reference or Product to Classification Link type can be set up to be dependent on a dimension. Dimension dependency is used to specify the dimension points, and by extension, the contexts, where a reference or link should display or should be suppressed. A reference or link can only be dependent on a single dimension point. If additional dimensions are required, create a new type for the same source and target objects.

When deciding on the need for dimension dependency, consider the following points:

- You are strongly encouraged to give this topic a lot of thought when setting up the STEP system and before a lot of data is migrated in. It is easier to remove a dependency than to add the dependency after references have been created. See the **Adding a Dimension Dependency After Loading Data** topic for more information on what it takes to correct this or to move dimension-dependent references from the All level to a real language or country.
- If you set dimension dependency on an asset reference type, do not use it in conjunction with dimension-dependent assets. Although STEP can handle this scenario, maintaining and troubleshooting is complicated when multiple dependencies are in use. For more information on dimension-dependent assets, see the **Asset Dimension Dependencies** section of **Maintaining Assets** in the **Getting Started / User Guide** documentation.
- Consider using Users and Groups privileges to hide objects by Reference Type, instead of using dimension dependency. For more information, see the **Attribute Groups for Reference and Link Types** topic.

Dimension Dependency on the 'Reference-Type' Object Type

Dimension dependency can be added on the 'Reference-Type' object type, which is located under System Setup > Object Types & Structures > Basic Object Types.

The screenshot displays the 'Reference-Type - Object Type' configuration window. On the left, the 'System Setup' tree shows 'Reference-Type' selected. On the right, the 'Object Type' configuration table is visible, showing various attributes and their values. The 'Dimension Dependencies' attribute is highlighted with a red box, showing the value 'Language;'.

Description	
Name	Value
ID	Reference-Type user-type root
Name	Reference-Type
Last edited by	2016-10-11 11:34:15 by USER4
Name Pattern	
ID Pattern	
Icon	
Dimension Dependencies	Language;

Using the Dimension Dependencies parameter on the Reference-Type object type is a global setting that enables translation of reference type names and requires that all reference type names be translated. Translated reference type names are useful on websites where viewers want to see information in their own language. For example, 'Installation Guide' could be the reference type and the link that appears when someone needs to download a PDF version of a document.

Setting a Reference-Type object type dimension dependency does not affect the dimension dependencies between referenced objects.

Dimension Dependency on a Reference Type

Setting the Dimension Dependencies parameter on a reference type adds dimension dependency on the reference between objects. This can be useful for a global website structure where not all products are sold in all countries, or where documents vary by language or country and users only want to see the appropriate versions.

The screenshot shows the 'System Setup' interface. On the left, a tree view under 'Reference Types' shows 'Primary Product Image' selected. On the right, the 'Primary Product Image - Reference Type' configuration window is open. It has tabs for 'Reference Type', 'Validity', and 'Log'. The 'Reference Type' tab is active, showing a table of properties.

Name	Value
ID	PrimaryProductImage
Name	Primary Product Image
Last edited by	2016-12-19 10:20:21.0 by USER3
Externally Maintained	No
Dimension Dependencies	Language;
Allow multiple references	No
Mandatory	No
Inheritance	Inherited

This is not a typical setup because the References / Referenced By tabs can become difficult to manage due to the layout required when viewing multiple contexts.

One scenario where this setup would be useful might be when there are 20 documents in STEP, one for each of the 20 countries where a product is sold. A dimension-dependent reference would make it possible for a user working in a country context to only see the document that is used for that country since the rest would be filtered from view.

For information on managing when a dimension dependent reference is available, see the **Visibility of Reference and Link Types** section.

Viewing Dimension Dependent Values

When dimension dependency is applied to a reference type, values can be applied based on the selected dimension. For example, the following reference is dependent on the Country dimension, so each country can have a different setting. The Context view displays all values for each country, based on the targets selected from the View > Target menu option.

Context English UK

URL I&contextid=Context7&id=20866&workspaceid=Main

Goto ID/Name

Tree

Primary Product Hierarchy

Products

Apparel

Hardware

Tools

Task Lighting

Flashlights

Flashlights Items

Flashlights SalesItems

20866-012

No Title

Compare ContextsReferencesReferenced By

Attribute Group

Reference Type: Alternate Supplier Item

Context	Reference Type	Target	Visibility	
> French FR	+ Alternate Supplier Item	(114861)	France	X
> English UK	+ Alternate Supplier Item	20862	Great Britain	X
> Danish DK	+ Alternate Supplier Item	Green light	Denmark	X

Discontinued Product Maintenance

Inheritance Example for a Product to Classification Link Type

A Product to Classification Link Type has a product for the source, a classification for the target, and displays linked products as children of the selected classification. This type of link includes the inheritance options demonstrated below.

System Setup

The 'Basic Products' product to classification link type is created to allow products of the object type Item or ItemFamily to be linked into classifications of the object type WebLevel3.

System Setup

Reference Types

Product Reference Types

Image and Document Reference Types

Classification Reference Types

Product to Classification Link Types

Basic Products

Classifications

eCl@ss

eClassSEG

ETIM Link

GPC

Merchandising Link

Basic Products - Validity

Product to Classification Link Type

Validity

Log

Valid Product Types

ID	Name
Item	Item
ItemFamily	Item Family

Modify Product Types

Valid Classification Types

ID	Name
WebLevel3	WebLevel3

Modify or Move Classification Types

Tree

The 'Basic Products' product to classification link type is used to link a classification named 'Pens' (object type Weblevel3) to a product family named 'Pilot Pens' (object type ItemFamily).

Tree

Pens

Pilot Pens

ABC-2500-1009

ABC-2500-1010

ABC-2500-1011

Pilot Pens rev.0.1 - References

Product

Product Variants

Sub Products

References

Referenced By

Images & Documents

Comme

Ungrouped Classification Links

Reference Type	Target
Basic Products	Web Sites US/Website Root US/Web Level 1 US/Web Level 2 US/Pens

Results

The 'Basic Products' link type would have the following inheritance effects on the linked products via the 'Sub Products Inheritance Settings' parameter on the wizard or the separate parameter settings on the editor 'Inheritance of Links' and 'Inheritance of Specification Attributes.' Note that the inheritance of tables is indicated by the 'Inheritance of Specification Attribute' parameter setting.

Basic Products - Product to Classification Link Type

Description

Name

ID

Name

Last edited by

Externally Maintained

Dimension Dependencies

Allow multiple links

Mandatory

Inheritance of Links

Inheritance of Specification Attributes

Ignore LOV Filter definitions on inherited Attributes for links

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

Sub Products Inheritance Settings

No Inheritance

No Inheritance

No Inheritance of Tables and Specification Attributes

Inherit Links and Tables and Specification Attributes

Accumulative inheritance without Tables and Specification Attributes

Accumulative inheritance with Tables and Specification Attributes

Inheritance of Links

None

Inheritance of Specification Attributes

No

Ignore LOV Filter definitions on inherited Attributes for links

No

No Inheritance

This option in the wizard displays as 'Inheritance of Links = None' and 'Inheritance of Specification Attributes = No' in the editor.

The value of the Basic Products link is not inherited from the product family (Pilot Pens) to the linked products (ABC-2500-1009, ABC-2500-1010, and ABC-2500-1011).

Tree

Pens

Pilot Pens

ABC-2500-1009

ABC-2500-1010

ABC-2500-1011

ABC-2500-1009 rev.0.1 - References

Product

Sub Products

References

Referenced By

Images & Documents

Ungrouped Classification Links

Reference Type

Target

Basic Products

Specification attributes applied to the classification are not inherited to products below the classification. In the example below, if available, the Application attribute would display along with the Net Weight attribute in the Specifications attribute group flipper.

Tree

Pens

Pilot Pens

ABC-2500-1009

ABC-2500-1010

ABC-2500-1011

Pens rev.0.15 - References

ClassificationSub ProductsReferencesReferenced ByImages & DocumentsTables

Attributes

IDNameCompleteness ScoreDisplaySequen

ApplicationApplication

Tree

Pens

Pilot Pens

ABC-2500-1009

ABC-2500-1010

ABC-2500-1011

Wooden Pencils

ABC-2500-1009

ProductSub ProductsReferencesReferenced ByImages & Documents

Specifications

NameValue

Net Weightabc

Website Item Descriptions

Tables applied to the classification are not inherited to products below the classification.

Tree

Pens

Pilot Pens

ABC-2500-1009

ABC-2500-1010

ABC-2500-1011

Pens rev.0.1 - Tables

ClassificationSub ProductsReferencesReferenced ByImages & DocumentsTablesSta

Tables

NameDefined At

Pivot TableLocal

Add Table Type

Tree

Pens

Pilot Pens

ABC-2500-1009

ABC-2500-1010

ABC-2500-1011

ABC-2500-1009 rev.0.1 - Tables

ProductSub ProductsReferencesReferenced ByImages & DocumentsCon

Tables

NameDefined At

Price TableProducts

Sales Price TableSuppressed

Add Table Type

No Inheritance of Tables and Specification Attributes

This option in the wizard displays as 'Inheritance of Links = Inherited' and 'Inheritance of Specification Attributes = No' in the editor.

The value of the Basic Products link is inherited from the classification (Pens) to the linked products (ABC-2500-1009, ABC-2500-1010, and ABC-2500-1011).

Note: If multiple links are allowed, and a new link of this type is added to a linked product (sub product), the inherited link is removed and only the local link remains.

Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

ABC-2500-1009 rev.0.1 - References

ProductSub ProductsReferencesReferenced ByImages & DocumentsCommercialTables

Ungrouped Classification Links

Reference Type	Target
Basic Products	Web Sites US/Website Root US/Web Level 1 US/Web Level 2 US/Pens

Specification attributes applied to the classification are not inherited to products below the classification. In the example below, if available, the Application attribute would display along with the Net Weight attribute in the Specifications attribute group flipper.

Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

Pens rev.0.15 - References

ClassificationSub ProductsReferencesReferenced ByImages & DocumentsTables

Attributes

ID	Name	Completeness Score	DisplaySequen
Application	Application		

Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011
 - Wooden Pencils

ABC-2500-1009 rev.0.1 - Specifications

ProductSub ProductsReferencesReferenced ByImages & Documents

Specifications

Name	Value
Net Weight	abc

Website Item Descriptions

Tables applied to the classification are not inherited to products below the classification.

Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

Pens rev.0.1 - Tables

ClassificationSub ProductsReferencesReferenced ByImages & DocumentsTablesSta

Tables

Name	Defined At
Pivot Table	Local

Add Table Type

Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

ABC-2500-1009 rev.0.1 - Tables

ProductSub ProductsReferencesReferenced ByImages & DocumentsCon

Tables

Name	Defined At
Price Table	Products
Sales Price Table	Suppressed

Add Table Type

Inherit Links and Tables and Specification Attributes

This option in the wizard displays as 'Inheritance of Links = Inherited' and 'Inheritance of Specification Attributes = Yes' in the editor.

The value of the Basic Products link is inherited from the classification (Pens) to the linked products (ABC-2500-1009, ABC-2500-1010, and ABC-2500-1011).

Note: If multiple links are allowed, and a new link of this type is added to a linked product (sub product), the inherited link is removed and only the local link remains.

Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

ABC-2500-1009 rev.0.1 - References

ProductSub ProductsReferencesReferenced ByImages & DocumentsCommercialTables

Ungrouped Classification Links

Reference Type	Target
Basic Products	Web Sites US/Website Root US/Web Level 1 US/Web Level 2 US/Pens

Specification attributes applied to the classification are inherited to products below the classification. In the example below, the Application attribute displays along with the Net Weight attribute in the Specifications attribute group flipper.

Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

Pens rev.0.15 - References

ClassificationSub ProductsReferencesReferenced ByImages & DocumentsTables

Attributes

ID	Name	Completeness Score	DisplaySequen
Application	Application		

Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011
 - Wooden Pencils

ABC-2500-1009

ProductSub ProductsReferencesReferenced ByImages & Do

Specifications

Name	Value
Application	abc
Net Weight	abc

Tables applied to the classification are inherited to products below the classification.

Tree

Pens

Pilot Pens

ABC-2500-1009

ABC-2500-1010

ABC-2500-1011

Pens rev.0.1 - Tables

ClassificationSub ProductsReferencesReferenced ByImages & DocumentsTablesSta

Tables

Name

Defined At

Pivot Table

Local

Add Table Type

Tree

Pens

Pilot Pens

ABC-2500-1009

ABC-2500-1010

ABC-2500-1011

ABC-2500-1009 rev.0.1 - Tables

ProductSub ProductsReferencesReferenced ByImages & DocumentsCon

Tables

Name

Defined At

Pivot Table

Pens

Price Table

Products

Sales Price Table

Suppressed

Accumulative Inheritance without Tables and Specification Attributes

This option in the wizard displays as 'Inheritance of Links = Accumulative' and 'Inheritance of Specification Attributes = No' in the editor.

The value of the Basic Products link is inherited from the classification (Pens) to the linked products (ABC-2500-1009, ABC-2500-1010, and ABC-2500-1011).

Note: If multiple links are allowed, and a new link of this type is added to a linked product (sub product), the inherited link is retained and the local link is added.

Tree

Pens

Pilot Pens

ABC-2500-1009

ABC-2500-1010

ABC-2500-1011

ABC-2500-1009 rev.0.1 - References

ProductSub ProductsReferencesReferenced ByImages & DocumentsCommercialTables

Ungrouped Classification Links

Reference Type

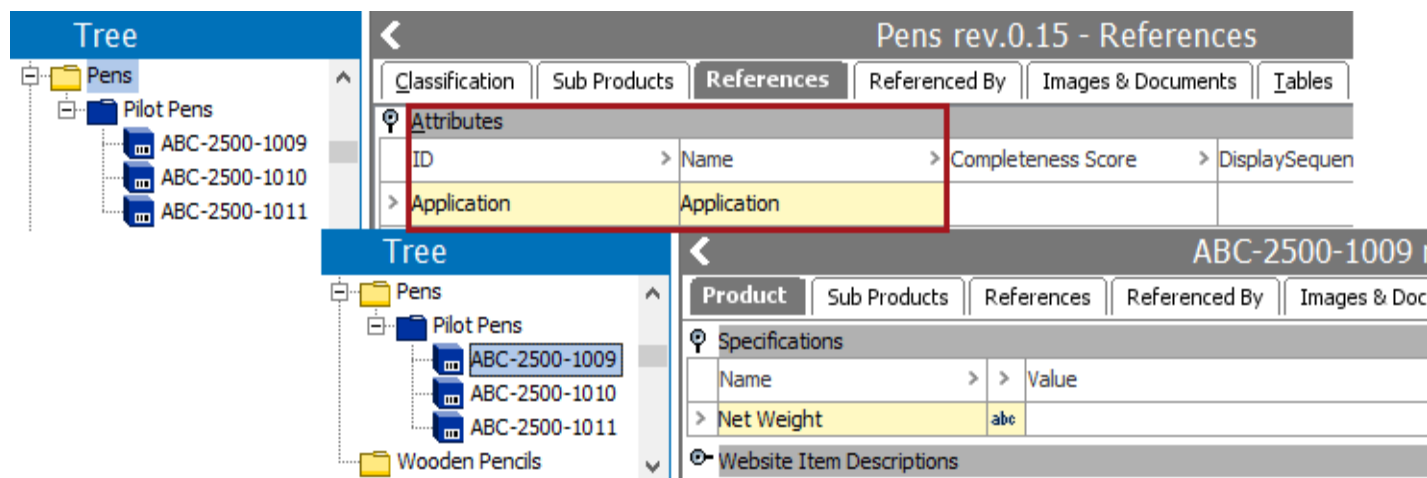
Target

Basic Products

+

Web Sites US/Website Root US/Web Level 1 US/Web Level 2 US/Pens

Specification attributes applied to the classification are not inherited to products below the classification. In the example below, if available, the Application attribute would display along with the Net Weight attribute in the Specifications attribute group flipper.



Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

Pens rev.0.15 - References

Classification	Sub Products	References	Referenced By	Images & Documents	Tables
Attributes					
ID	Name	Completeness Score	DisplaySequen		
Application		Application			

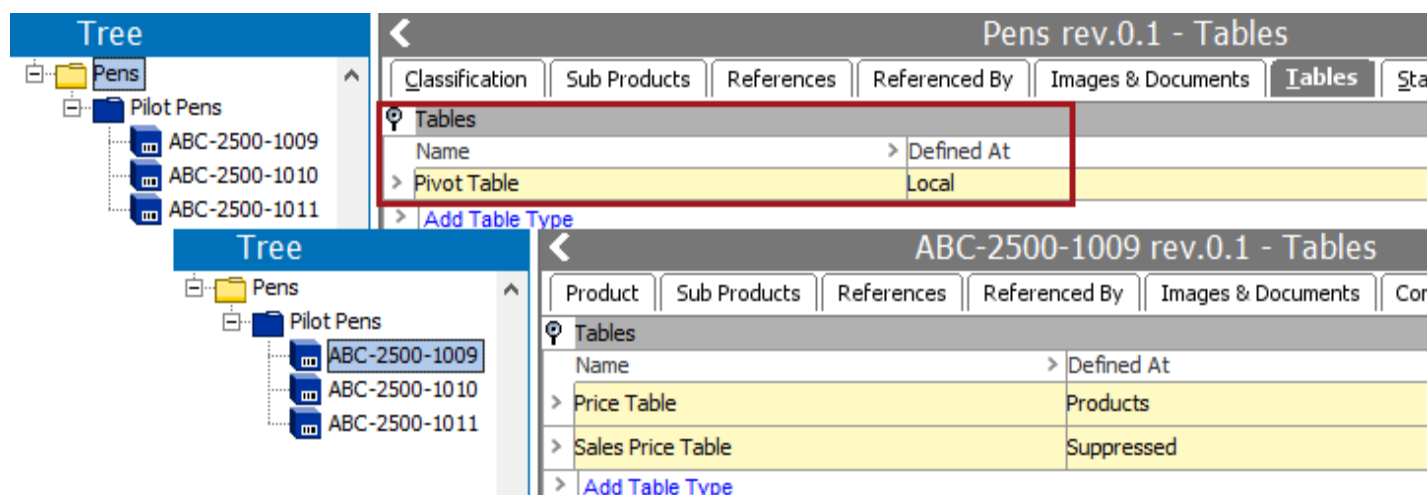
Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011
 - Wooden Pencils

ABC-2500-1009 rev.0.15 - References

Product	Sub Products	References	Referenced By	Images & Documents
Specifications				
Name	Value			
Net Weight	abc			
Website Item Descriptions				

Tables applied to the classification are not inherited to products below the classification.



Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

Pens rev.0.1 - Tables

Classification	Sub Products	References	Referenced By	Images & Documents	Tables	Sta
Tables						
Name	Defined At					
Pivot Table	Local					
Add Table Type						

Tree

- Pens
 - Pilot Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

ABC-2500-1009 rev.0.1 - Tables

Product	Sub Products	References	Referenced By	Images & Documents	Con
Tables					
Name	Defined At				
Price Table	Products				
Sales Price Table	Suppressed				
Add Table Type					

Accumulative Inheritance with Tables and Specification Attributes

This option in the wizard displays as 'Inheritance of Links = Accumulative' and 'Inheritance of Specification Attributes = Yes' in the editor.

The value of the Basic Products link is inherited from the classification (Pens) to the linked products (ABC-2500-1009, ABC-2500-1010, and ABC-2500-1011).

Note: If multiple links are allowed, and a new link of this type is added to a linked product (sub product), the inherited link is retained and the local link is added.

Tree

- Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

ABC-2500-1009 rev.0.1 - References

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Tables

Ungrouped Classification Links

Reference Type	Target
Basic Products	Web Sites US/Website Root US/Web Level 1 US/Web Level 2 US/Pens

Specification attributes applied to the classification are inherited to products below the classification. In the example below, the Application attribute displays along with the Net Weight attribute in the Specifications attribute group flipper.

Tree

- Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

Pens rev.0.15 - References

Classification

Sub Products

References

Referenced By

Images & Documents

Tables

Attributes

ID	Name	Completeness Score	Display Sequen
Application	Application		

Tree

- Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011
- Wooden Pencils

ABC-2500-1009

Product

Sub Products

References

Referenced By

Images & Do

Specifications

Name	Value
Application	abc
Net Weight	abc

Tables applied to the classification are inherited to products below the classification.

Tree

- Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

Pens rev.0.1 - Tables

Classification

Sub Products

References

Referenced By

Images & Documents

Tables

Sta

Tables

Name	Defined At
Pivot Table	Local

Add Table Type

Tree

- Pens
 - ABC-2500-1009
 - ABC-2500-1010
 - ABC-2500-1011

ABC-2500-1009 rev.0.1 - Tables

Product

Sub Products

References

Referenced By

Images & Documents

Con

Tables

Name	Defined At
Pivot Table	Pens
Price Table	Products
Sales Price Table	Suppressed

In the wizard, you can specify any of the settings for the inheritance of the product to classification link type as defined in **Product to Classification Link Type - Advanced**.

The wizard selections are reflected in the reference editor Inheritance of Links and 'Inheritance of Specification Attributes' parameters as defined in **Maintaining a Product to Classification Link Type** documentation.

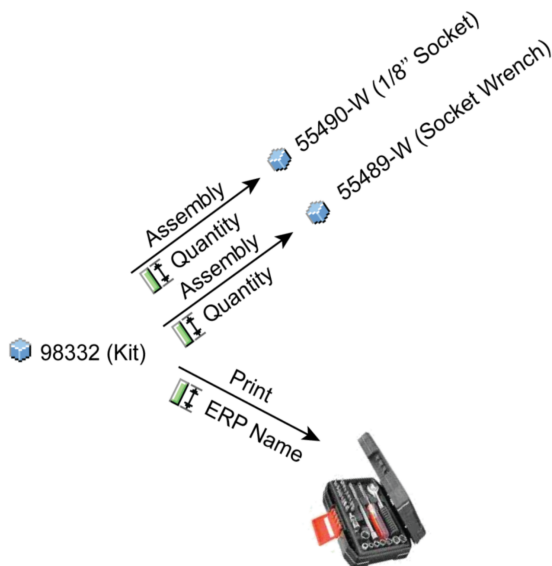
Metadata Attributes on Reference and Link Types

Metadata (description) attributes can be applied to all objects in the System Setup > Reference Types folder, including a Reference type, a Product to Classification Link type, the Classification Attribute link type, and the Product Attribute link type.

Metadata is maintained on the References or Referenced By tabs of the object where the reference or link is applied. Different attributes can be selected for each type of reference or link type.

Example of metadata on a reference or link

Consider product 98332 (KIT), which includes two sockets and is referencing an image to be used for Print. Different metadata attributes are needed for references applied to images and the references applied to products.



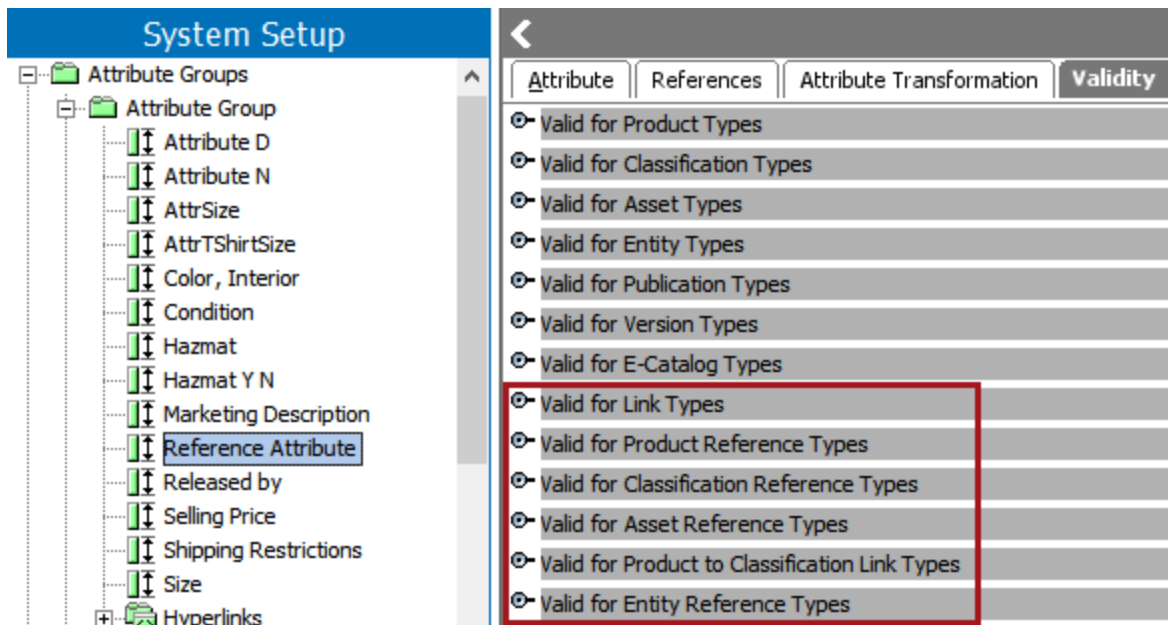
- A description attribute named Quantity has been applied the reference type Assembly. This attribute holds number of sockets to be included in the kit.
- A description attribute named 'ERP Name' has been applied the reference type Print. This attribute holds the name of the reference used in an ERP System.

For information on adding metadata to the attribute link types, see **Product Attribute Link Type** or **Classification Attribute Link Type**.

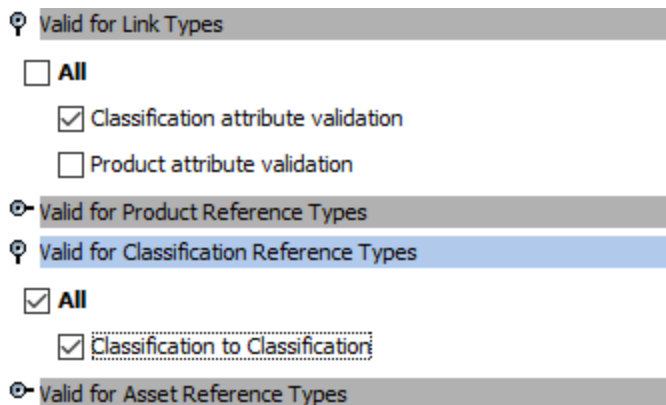
Add a Metadata Attribute

Although the images below show a reference type, the same setup is used for adding metadata on a Product to Classification link type.

1. In System Setup > Attribute Groups > select the description attribute that should be added as metadata on the reference or link. Specification attributes and groups cannot be added as metadata.
2. Open one (or more) reference types flippers or the link types flippers.

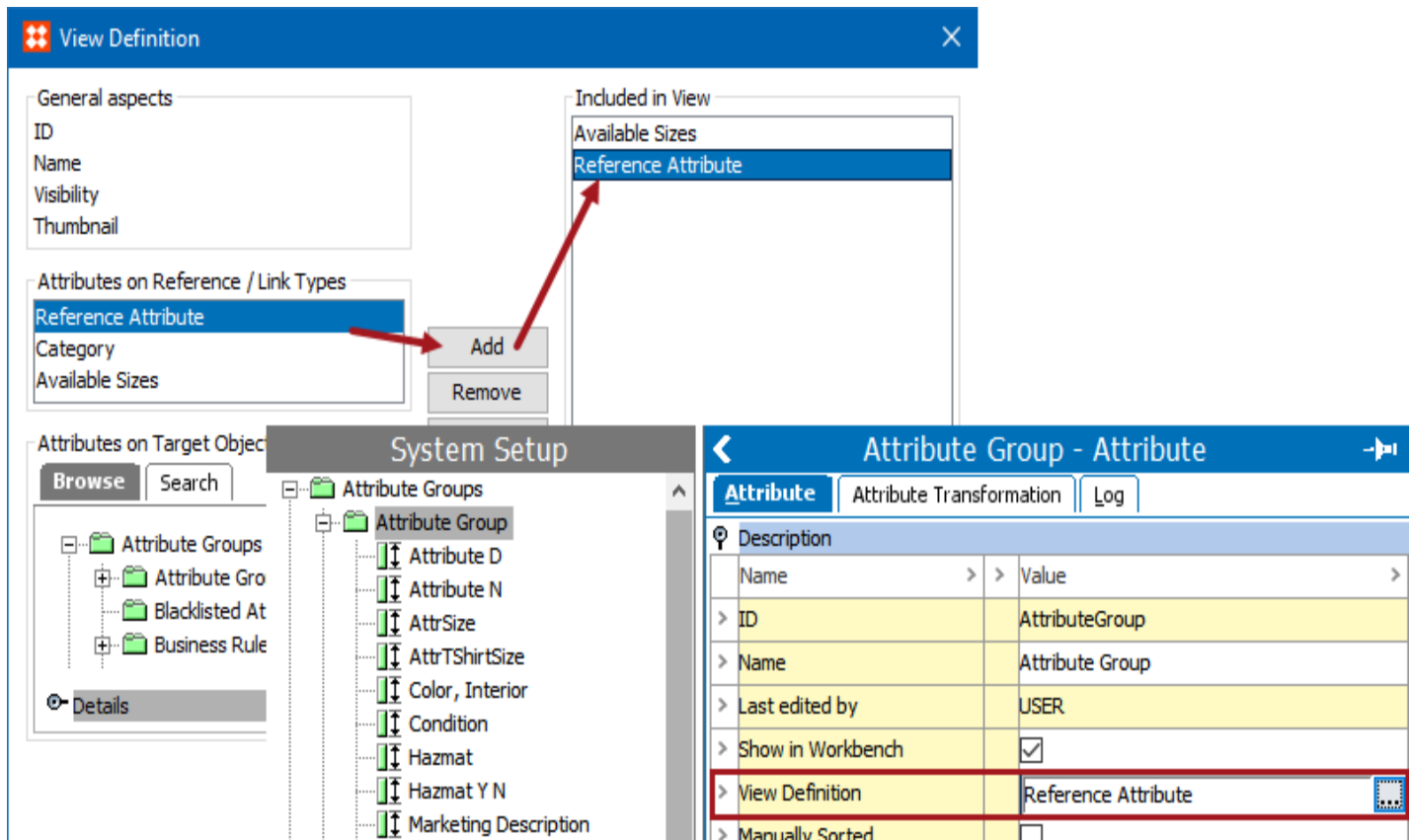


3. Select all reference types and link types that should display the metadata attribute. Checking an option here also updates the Attributes flipper of the selected reference or link type.

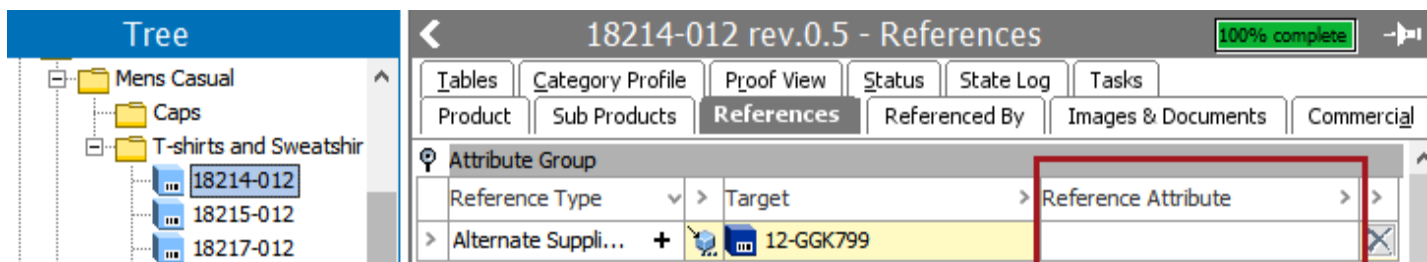


For more information, see **Validity on Description Attributes** in the **System Setup / Super User Guide**.

4. Add the description attribute from the 'Attributes on Reference / Link Types' section to the attribute group's View Definition so it will display in the object editors as metadata on any reference or link in the group. For detailed steps, see **Creating a Customized View** section in the **Attribute Groups** topic of the **System Setup / Super User Guide** documentation.



5. Verify that the metadata attribute is available on the reference.



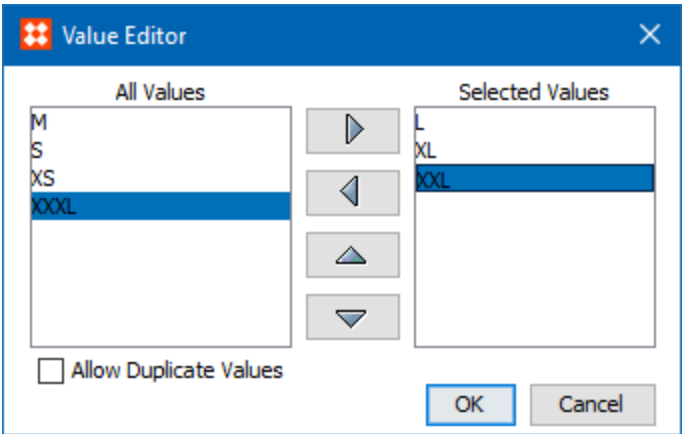
Edit a Metadata Attribute

Although the images below show a reference type, the same setup is used for editing metadata on a Product to Classification link type.

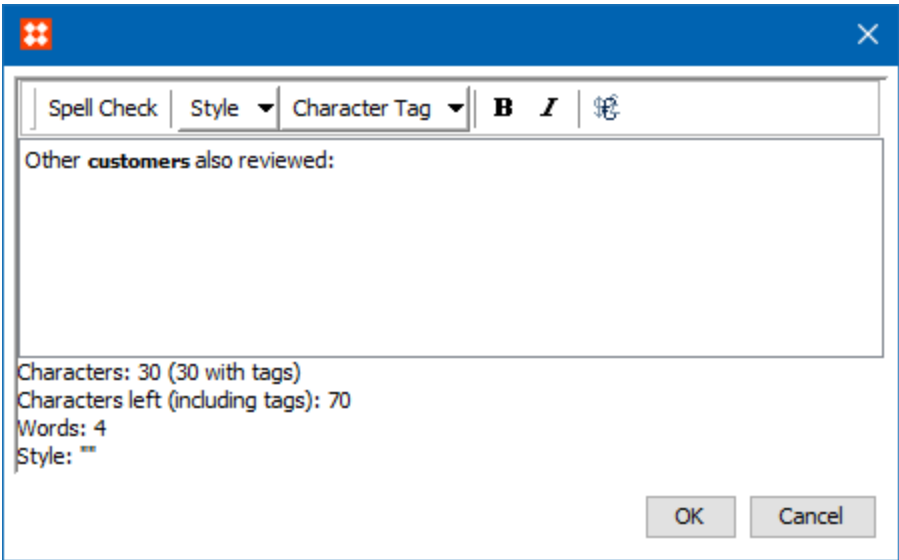
On the **References** tab, double-click an editable field (any white field as shown above) and edit using the available functionality based on the type of the metadata attribute.

- Modify the value directly in the field by adding text or selecting from a dropdown.
- Double-click (or right-click > Edit) to display a value editor.

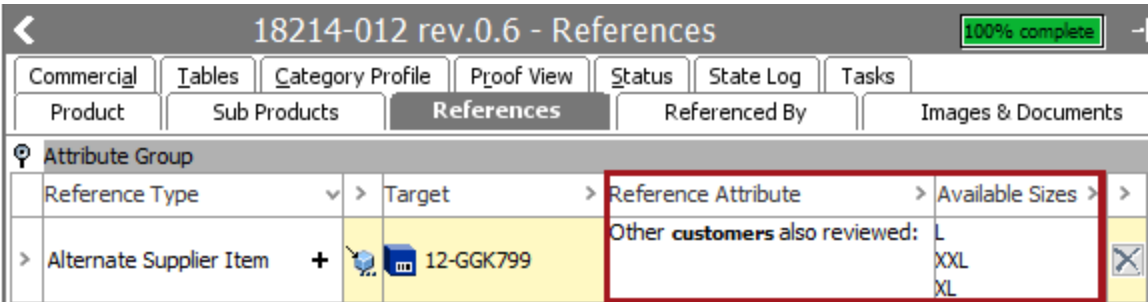
Choose a value and use the necessary arrow button to include it in the Selected Values list, or to rearrange the order of the values.



Add or modify text, including styles, character tags, and spell check.



Edits made to metadata on a reference or link are displayed:



Visibility of Reference and Link Types

On dimension dependent references and link types, you can specify the visibility of a reference or link in a specific context. Based on the visibility, you can make a single setting and allow it to be inherited by other contexts, make local settings on some contexts, make all contexts have a local setting, or suppress the reference completely for a context.

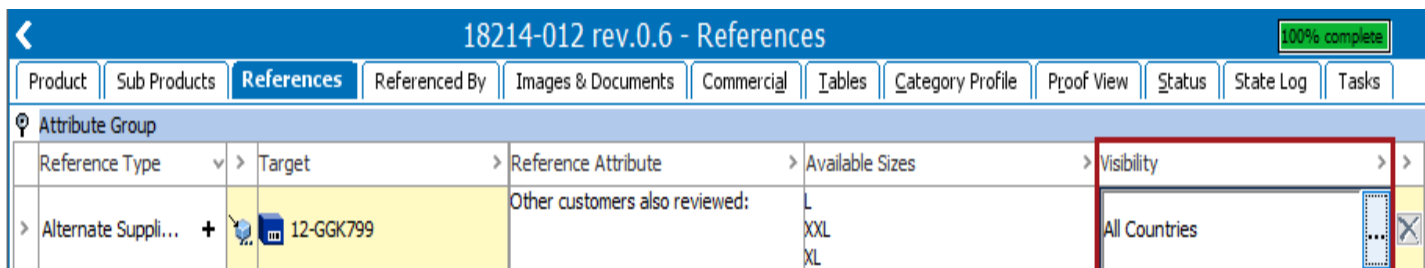
- Making the visibility global means the reference settings are inherited from the global context. For example, when all contexts require the same setting.
- Making the visibility local means the reference does not inherit its settings from the global context. For example, if you want to define attributes on a language-by-language basis.
- Suppressing the visibility means it will not be used in a specific context.

Metadata 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.

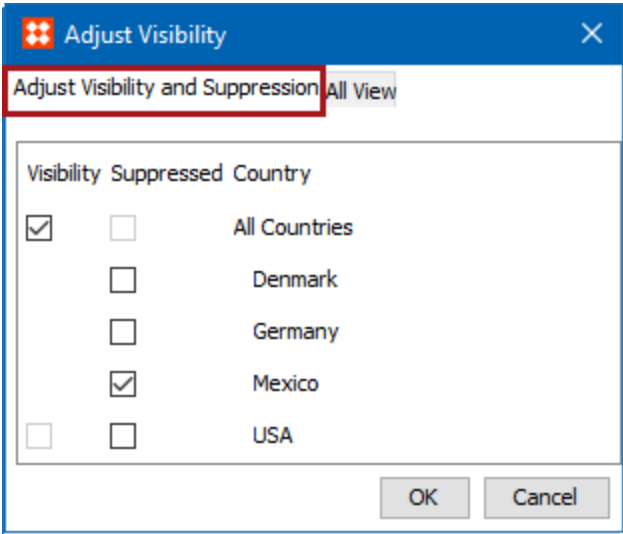
Note: The Visibility column is displayed on a reference when the General Aspect option 'Visibility' is added to the 'Included in View' panel on the View Definition of the attribute group. For more information, see the **Add a Metadata Attribute** section of the **Metadata Attributes on Reference and Link Types** documentation.

Setting Reference and Link Type Visibility

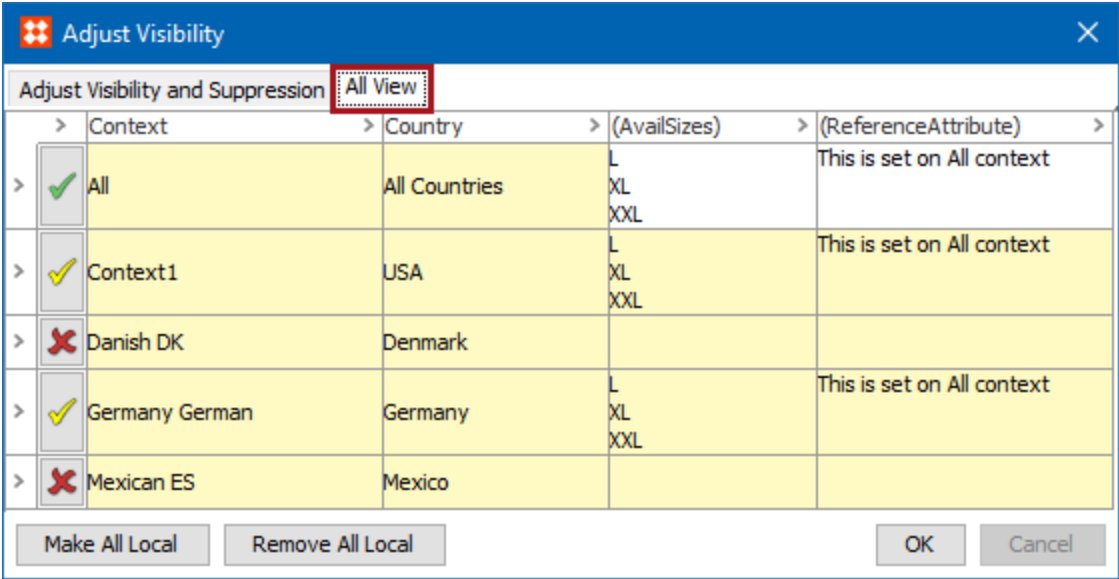
1. In the Tree, select the relevant node, click the **References** tab, and open the flipper that contains the dimension-dependent reference to be edited.



2. Double-click the **Visibility** field to display the ellipsis button, and then click the ellipsis button (...). The **Adjust Visibility** dialog displays with the 'Adjust Visibility and Suppression' tab active.



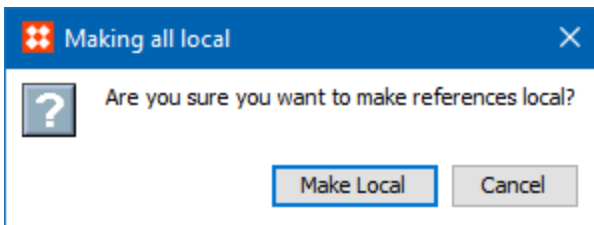
- 3. In the Suppressed column, check any contexts that should not display the reference.
- 4. Click the **All View** tab and note the status of the visibility of each context:
 - indicates that the metadata attribute is local and not being inherited (All). The setting is inherited by the contexts that display a yellow check.
 - indicates that the contexts inherits the settings of the context with the green check (USA and Germany).
 - indicates that the reference is suppressed (Mexico), or does not exist in the context (Denmark).



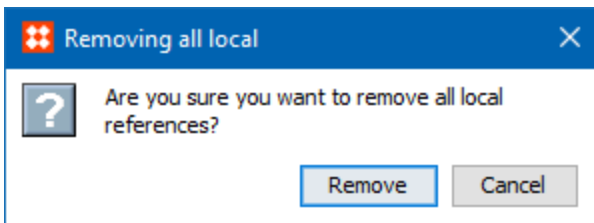
- 5. Modify the visibility of each required context as needed:
 - Click or to make the visibility of a reference local for only the selected context. Assuming the user has privileges to create the reference, the indicator changes to , the reference value field changes to a white

background, indicating that the setting can be modified on the selected context.

- Click  to remove the local reference for only the selected context. The indicator changes to , and inheritance is restored from the global context, and the reference value field changes to a yellow background, indicating that the setting cannot be modified on the selected context.
- Click **Make All Local** to make the visibility of the reference local for all contexts. In the 'Making all local' dialog, confirm that you want to make all local by clicking the **Make Local** button. All check marks turn green except the icon next to the contexts that were previously local (green). Previously local contexts are set with a red X, indicating that the reference no longer appears in those contexts. When the visibility of a reference is set to local, it no longer inherits its settings from the global context. This allows you to specify, for example, attribute values that only apply in the local context.



- Click **Remove All Local** to restore inheritance for all contexts currently local. In the 'Removing all local' dialog, confirm that you want to restore inheritance by clicking the **Remove** button. If a global context is local, all check marks turn yellow except the icon for the contexts that were previously local (green). Previously local contexts are set with a red X, indicating that the reference no longer appears in those contexts. When the visibility of a reference is set to global, it inherits its settings from the global context.



- Verify the results of the visibility by checking the References page on the object and make any additional changes. Use the Context Mode by clicking  in the toolbar.

Note: Navigate to View > Target to select the Contexts for the comparison view to display.

Compare Contexts

References

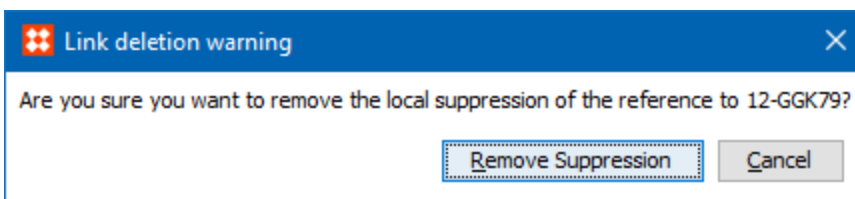
Referenced By

(AttributeGroup)

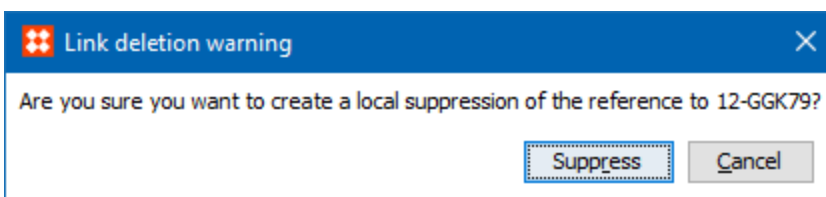
Reference Type (Alternate Supplier Item) ▾

Context	Reference Type	Target	(ReferenceAttribute)	(AvailSizes)	Visibility	
> Germany German	+ (Alternate Supplier Item)	12-GGK79	This is set on All context	L XL XXL	All Countries	
> All	+ (Alternate Supplier Item)	12-GGK79	This is set on All context	L XL XXL	All Countries	
> Mexican ES	+ (Alternate Supplier Item)	12-GGK79			Suppressed in [Mexico]	
> Context1	+ (Alternate Supplier Item)	12-GGK79	This is set on All context	L XL XXL	All Countries	
> Danish DK	(Alternate Supplier Item)					

- indicates the reference is allowed for the context. Note that reference is not available for the Danish DK context in the image above.
- indicates that the reference is local and values can be added for metadata.
- indicates that the reference is inherited and values cannot be added for metadata.
- indicates that the reference has been suppressed.
- allows you to remove the suppression for the selected context. In the 'Link deletion warning' dialog, confirm that you want to remove the suppression by clicking the **Remove Suppression** button. Previously set inheritance is restored. Return to the Adjust Visibility dialog to change visibility to local if necessary.



- allows you to remove the current setting and suppress the reference for the selected context. In the 'Link deletion warning' dialog, confirm that you want to create a local suppression by clicking the **Suppress** button.



Reference Types

References allow you to define a connection between two objects in STEP, essentially creating a set of rules for the relationship between the objects. For example, a reference can associate an image with a product, or join multiple addresses with a single company, or connect an object to a classification, among other uses.

The reference type name refers to the target of the reference. For example, Product Reference Types all have a product objects as the target. The reference types available allow connections between the objects defined in the table below.

Reference Type	Connections Allowed
Product Reference Types 	Classification-to-product Product-to-product Entity-to-product
Image and Document Reference Types 	Classification-to-asset Product-to-asset Entity-to-asset
Classification Reference Types 	Classification-to-classification Product-to-classification Entity-to-classification
Entity Reference Types 	Classification-to-entity Product-to-entity Asset-to-entity Entity-to-entity Publication-to-entity
Context Reference Types 	Entity-to-context
Workspace Reference Types 	Entity-to-workspace

Consider the following elements when setting up any type of reference:

- Adding a reference to an attribute group allows grouping of references and link, customized views, and added security via user group privileges. For more information, see **Attribute Groups for Reference and Link Types**.
- Dimension dependency determines if the reference is available based on the selected dimension point. For more information, see **Dimension Dependent Reference and Link Types**.
- Direction identifies one object as the 'source' and the referenced object as the 'target'. For more information, see **Direction of a Reference**.
- The only reference type that can display objects as parent / child in Tree is an entity reference type when both objects are entities. For more information, see **Entity Reference Types**.
- Metadata attributes on a reference allow description attributes to supply additional information about the reference itself. For more information, see **Metadata Attributes on Reference and Link Types**.
- Multiple references are allowed between the same two objects, using the same type of reference, when multiple reference types exist. For more information, see **Multiple References for the Same Objects**.
- Inheritance determines if a reference added to a parent object will also be visible on the child object. For more information, see **Inheritance Example for a Reference**.
- Dimension dependent references are displayed based on the Visibility setting. For more information, see **Visibility of Reference and Link Types**.

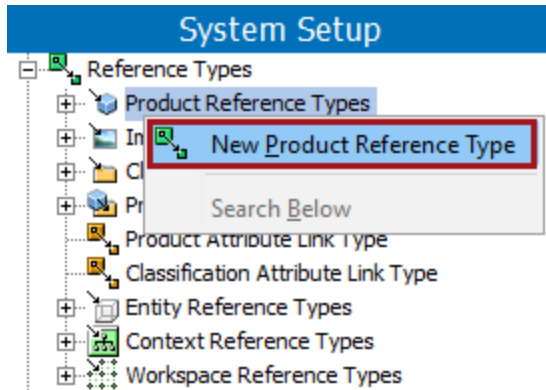
To create a reference type, follow the steps outlined in **Creating a Reference Type**.

To maintain a reference type, follow the steps outlined in **Maintaining a Reference Type**.

Creating a Reference Type

1. In System Setup > Reference Types > select the relevant reference type node > right-click, and choose the **New** option from the menu.

The following image shows creating a Product Reference Type, but the process is the same for all reference types.



You can create the following types of references using the right-click menu option identified below:

- Product Reference Type > New Product Reference Type
- Image and Document Reference Type > New Asset Reference Type
- Classification Reference Type > New Classification Reference Type
- Entity Reference Type > New Entity Reference Type
- Context Reference Type > New Context Reference Type
- Workspace Reference Type > New Workspace Reference Type

For information on Product to Classification Link Types, see the **Creating a Product to Classification Link Type** topic.

For information on the attribute link types, see the **Product Attribute Link Type** topic or the **Classification Attribute Link Type** topic.

2. Regardless of the reference type selected, the Create Reference Type wizard displays.

The image below shows a Product Reference Type, but the steps are the same for each 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

Enter ID and Name

ID

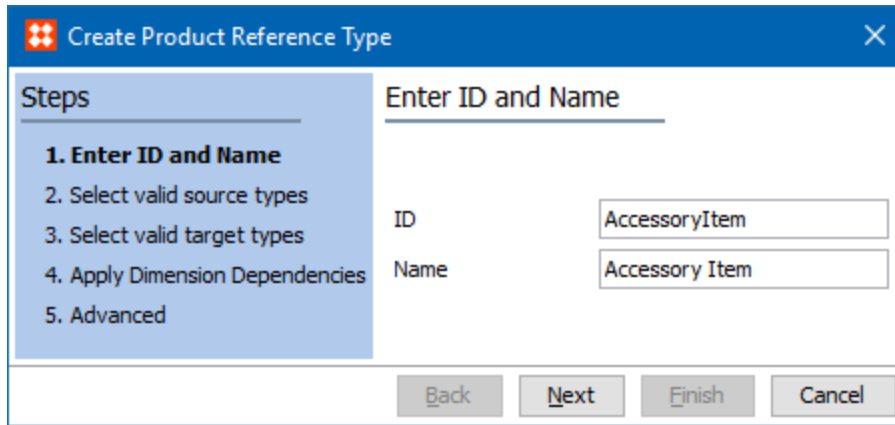
Name

Each reference wizard includes the following steps:

- **Enter ID and Name** allows you to identify the reference.
- **Select Valid Source Types** allows you to choose the types of products valid for this reference.
- **Select Valid Target Types** allows you to choose the alternate classifications valid for this reference.
- **Apply Dimension Dependencies** allows you to define the dimension points that are affected by this reference.
- **Advanced** allows you to configure multiple references, set externally maintained, mandatory, and inheritance for the reference.

Once the reference type exists, create a reference between the two objects as outlined in **Creating a Reference**.

Reference Type - Enter ID and Name

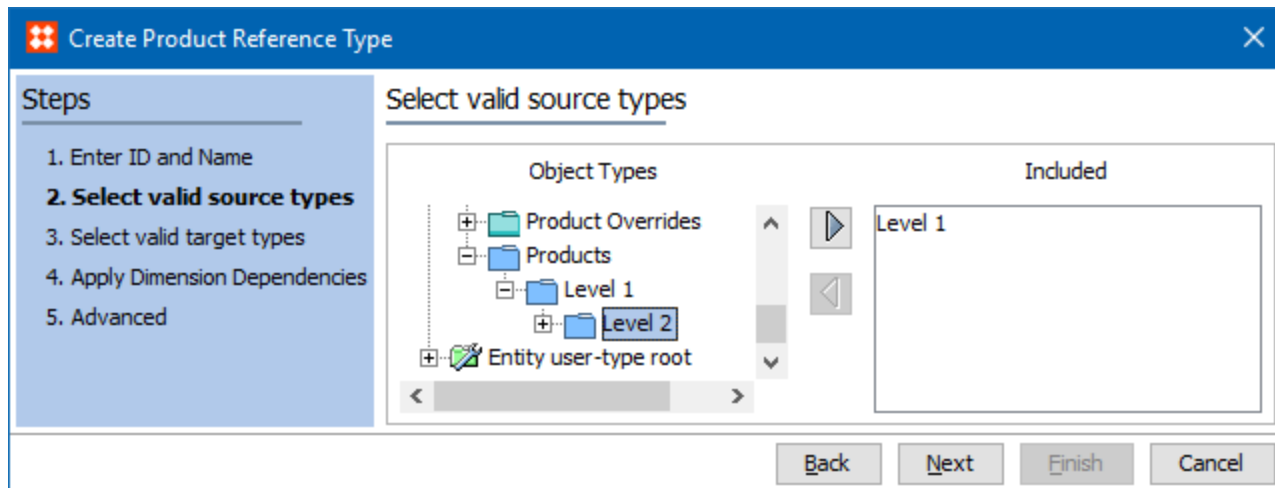


The screenshot shows a wizard window titled "Create Product Reference Type". On the left, a "Steps" panel lists five steps: 1. Enter ID and Name (highlighted), 2. Select valid source types, 3. Select valid target types, 4. Apply Dimension Dependencies, and 5. Advanced. The main area is titled "Enter ID and Name" and contains two input fields: "ID" with the value "AccessoryItem" and "Name" with the value "Accessory Item". At the bottom, there are four buttons: "Back", "Next", "Finish", and "Cancel".

This wizard step is the same for all types of references: Product, Image and Document, Classification, Entity, Context, and Workspace.

1. Enter an **ID** for the reference. Common setup is to use no spaces or punctuation.
2. Enter a **Name** for the reference. Common setup is to repeat the ID with added spaces for readability.
3. Click the **Next** button to display **Reference Type - Select Valid Source Types**.

Reference Type - Select Valid Source Types



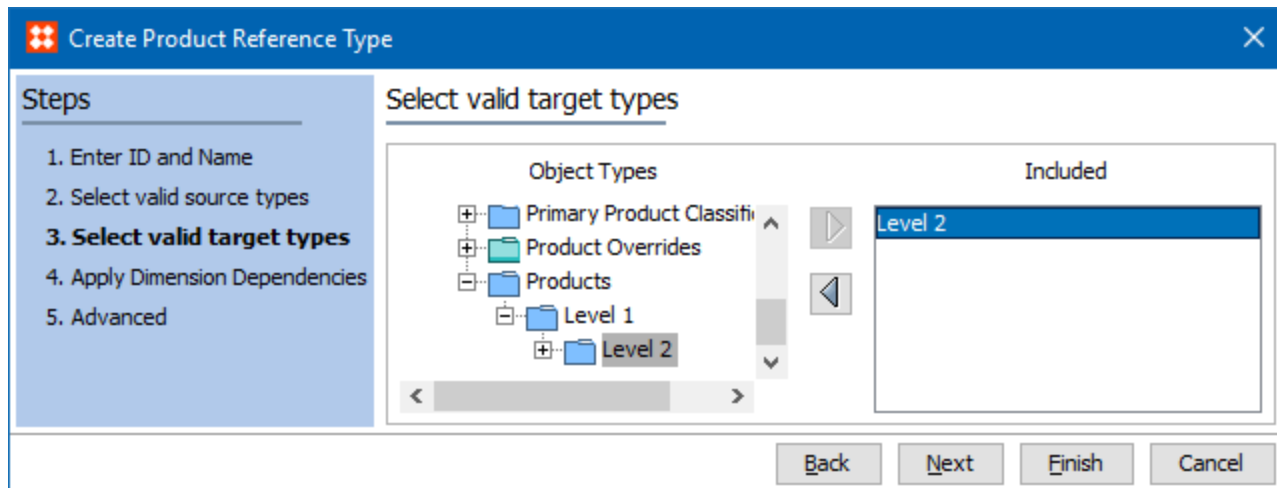
The contents of the Object Types pane in this wizard step displays only valid object types based on the selected reference type (Product, Image and Document, Classification, Entity, Context, and Workspace).

1. Use the displayed hierarchy to browse to and select one or more source object type of the reference. The 'source' is where the reference starts. For more information, see **Direction of a Reference**. Selecting an item from the Object Types pane enables the right arrow button.
2. Click the right arrow button (▶) to move the selected object type to the Included pane. All objects in the Included pane will be valid for the reference type being created.

Note: To remove an object type from the Included pane, first select the Included item, and then click the left arrow button (◀).

3. Click the **Next** button to display **Reference Type - Select Valid Target Types**.

Reference Type - Select Valid Target Types



The contents of the Object Types pane in this wizard step displays only valid object types based on the selected reference type (Product, Image and Document, Classification, Entity, Context, and Workspace).

1. Use the displayed hierarchy to browse to and select one or more target object types of the reference. The 'target' is where the reference ends. For more information, see **Direction of a Reference**. Selecting an item from the Object Types pane enables the right arrow button.
2. Click the right arrow button (▶) to move the selected object type to the Included pane. All objects in the Included pane will be valid for the reference type being created.

Note: To remove an object type from the Included pane, first select the Included item, and then click the left arrow button (◀).

3. Click the **Next** button to display **Reference Type - Apply Dimension Dependencies**.

Reference Type - Apply Dimension Dependencies

The screenshot shows a wizard window titled "Create Product Reference Type" with a close button (X) in the top right corner. On the left, a "Steps" panel lists five steps: "1. Enter ID and Name", "2. Select valid source types", "3. Select valid target types", "4. Apply Dimension Dependencies" (which is highlighted in blue), and "5. Advanced". The main area of the wizard is titled "Apply Dimension Dependencies" and contains a large rectangular box. Inside this box, there are two checkboxes: "Country" (which is checked) and "Language" (which is unchecked). At the bottom of the wizard, there are four buttons: "Back", "Next", "Finish", and "Cancel".

This wizard step is the same for all types of references: Product, Image and Document, Classification, Entity, Context, and Workspace.

1. If the reference type should be dimension dependent, select a dimension checkbox. Only one dimension is allowed for a reference type. If multiple dimension dependencies are needed, use a separate reference type. For more information, see **Dimension Dependent Reference and Link Types**.
2. Click the **Next** button to display **Reference Type - Advanced**.

Reference Type - Advanced

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

Sub Products Inheritance Settings

No Inheritance

No Inheritance

Normal Inheritance

Accumulative Inheritance

Back Next Finish Cancel

This wizard step is the same for all types of references: Product, Image and Document, Classification, Entity, Context, and Workspace.

1. **Allow Multiple References** determines the number of references of the type can be applied to the source object. For more information, see the **Maintaining a Reference Type** documentation.
 - **Checked** means the reference type can reference more than one target from same source.
 - **Unchecked** means the reference type is restricted to a single reference from the same source.
2. **Externally Maintained** determines if adding or modifying a reference of this type affects the Approved parameter on the source object. For more information, see the **Maintaining a Reference Type** documentation.
 - **Checked** means adding a reference from a source object to a target object will not required approved to display in the Approved workspace.
 - **Unchecked** means the reference type is controlled by revision control and adding a reference from a source object and/or changing values on attributes applied to the reference on the source object will require approval to display in the Approved workspace.

Note: The source object's approved status is reset (requires approval to display in the Approved workspace) if metadata attribute values are changed on the reference and the metadata attributes are not externally maintained.

3. **Mandatory** determines if the reference type is required for approval of the source object. For more information, see the **Maintaining a Reference Type** documentation.
 - **Checked** means the reference type is required from a source object to a target object before the source object can be approved.
 - **Unchecked** means the reference type for a source object to a target object is optional and does not affect approval of the source object.
4. **Sub Products Inheritance Settings** determines if and how references are displayed on valid child objects below the source, with respect to the Validity settings. For details on inheritance of references, see **Inheritance Example for a Reference**.
 - **No Inheritance** means the reference type will not be inherited by the sub products. The reference will only appear on the two referenced objects.
 - **Normal Inheritance** means when creating a reference from a source object to a target object, the reference will be inherited on all valid objects below the source object. Each inherited reference will reference the same target. If multiple references are allowed, and a new reference of this type is added to a sub product, the inherited reference is removed and only the local reference remains.
 - **Accumulative Inheritance** means when creating a reference from a source object to a target object, the reference will be inherited on all valid objects below the source object. Each inherited reference will reference the same target. If multiple references are allowed, and a new reference of this type is added to a sub product, the inherited reference is retained and the local reference is added to the list.
5. Click the **Finish** button to create the reference type.
6. To apply the new reference type to a pair of objects, see **Creating a Reference**.

Creating a Reference

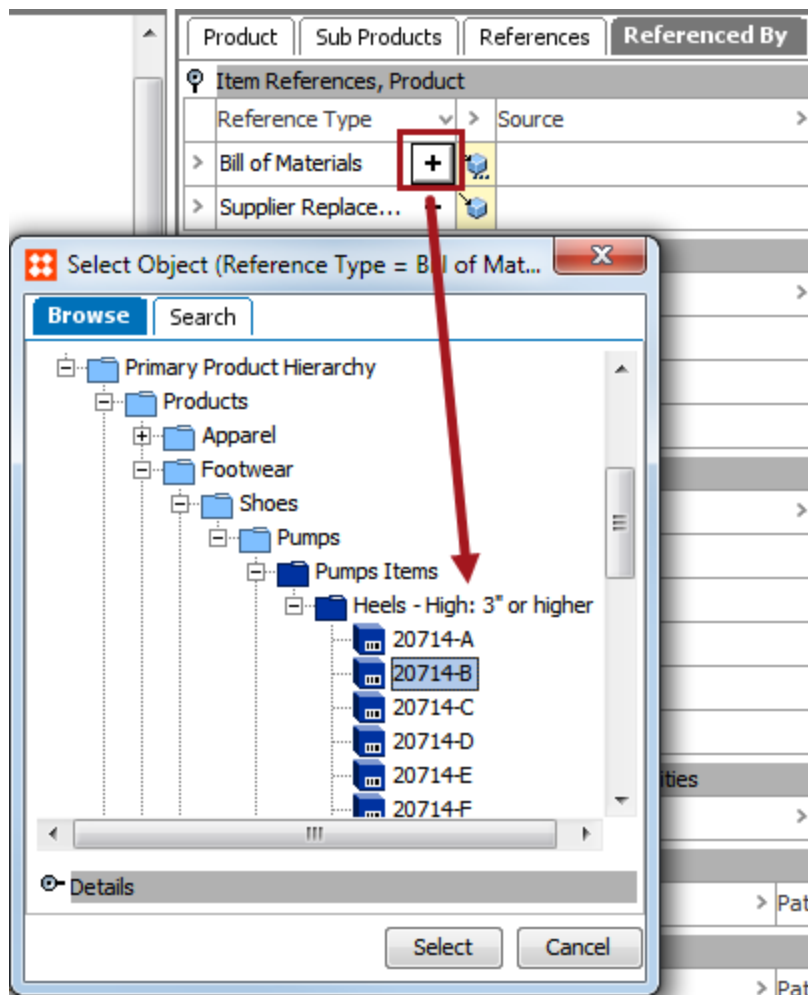
After creating a reference type, you can add that type on an object's References or Referenced By tabs.

To maintain data on the References and Referenced By tabs, see the **References Tab** topic and the **Referenced By Tab** topic in the **Getting Started / User Guide** documentation.

1. To add references to either the **References** or **Referenced By** tab of an object, use one of the following ways to find the reference type in a flipper:
 - If the reference type has been added to an attribute group, it appears under a flipper that is named the same as the attribute group, as shown in the image in the next step.
 - If the reference type is not in an attribute group, it appears under an 'Ungrouped References' flipper, based on the type of reference. For example, product references are under a flipper with the name 'Ungrouped Product References.'
2. Click the plus sign button in the desired reference field. If more than one reference is allowed for the reference type, the plus sign button continues to display after the first reference is added, otherwise it is removed.

Product	Sub Products	References	Referenced By
Item References, Product			
Reference Type	>	Source	>
> Bill of Materials	+		
> Supplier Replace...	+		

3. In the Select Object dialog, find the object that is to be added, and click the Select button. If you know the target ID / Name of the object, use the Search tab to find it quickly using the typeahead functionality to display the options as a dropdown.



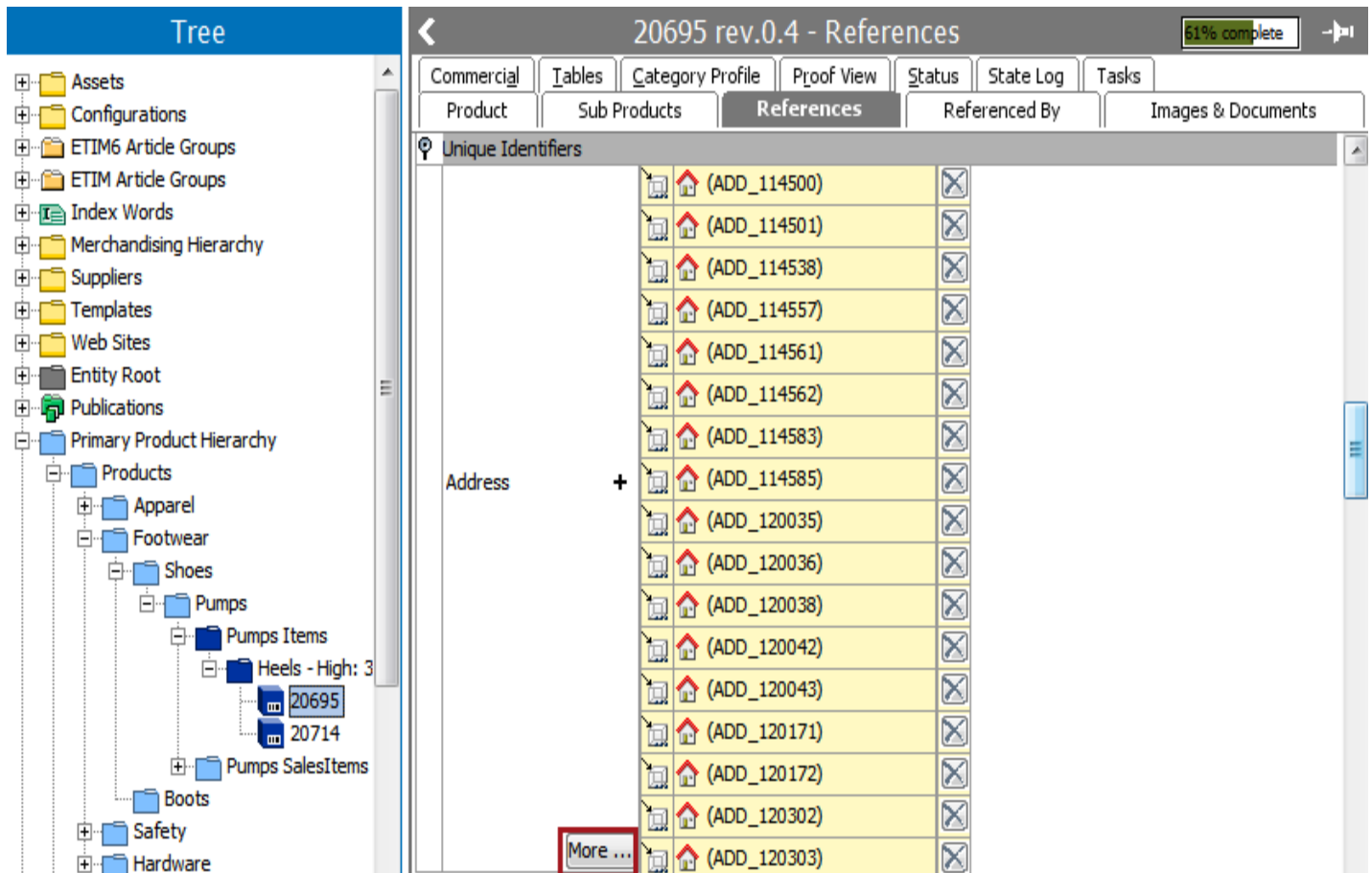
4. Once selected, the object displays as an added reference.

Product	Sub Products	References	Referenced By	Ima
Attribute Group				
Item References, Product				
Reference Type		Source		
> Bill of Materials	+	20714-B		X
> Supplier Replace...	+			

- To remove a reference, click the X button.

Displaying a large number of references

If more than 50 references of the same type are added, a button labeled 'More...' ([More ...](#)) displays, allowing users to expand the list and view 50 additional references.

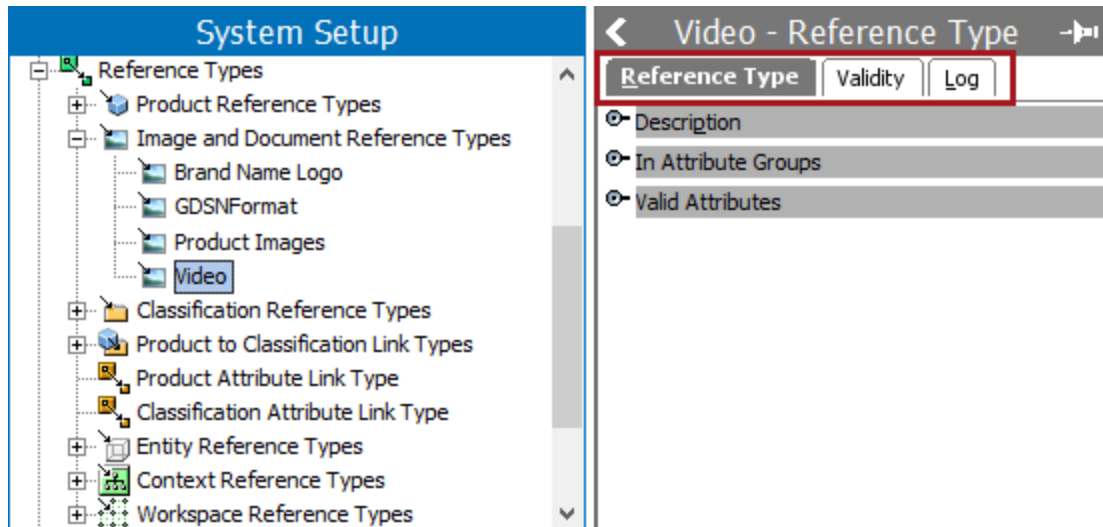


Repeatedly clicking the button doubles the number of references shown each time, until the full set is displayed. Note that if new references are added, they display on the list immediately indicating that they were added properly, though normal functionality will resume if the user presses the 'More...' button again. This means that items may become hidden until the list is fully expanded.

After the list has been fully expanded for the given reference type, it stays expanded until the user refreshes the page or clicks away to another object in workbench.

Maintaining a Reference Type

Select a reference to display the Reference Type editor and make updates to the reference type settings. With the few exceptions defined below, the tabs and parameters on a reference are the same, regardless of the type. Each tab on the editor is defined below and includes parameters to maintain a Product, Image and Document, Classification, Entity, Context, or Workspace reference type.



Reference Type Tab

This tab is displayed for the Product, Image and Document, Classification, Entity, Context, and Workspace reference types. It holds basic information within the Description, In Attribute Groups, and Valid Attribute flippers.

Description Flipper

The Description flipper includes basic information to identify the reference. This data is originally set up in the reference type wizard, see **Creating a Reference Type** documentation.

Reference Type		Validity	Log
Description			
Name	>	>	Value
ID			PrimaryProductImage
Name			Primary Product Image
Last edited by			2016-12-19 10:20:21.0 by USER3
Externally Maintained			No
Dimension Dependencies			
Allow multiple references			No
Mandatory			No
Inheritance			Inherited
In Attribute Groups			
Valid Attributes			

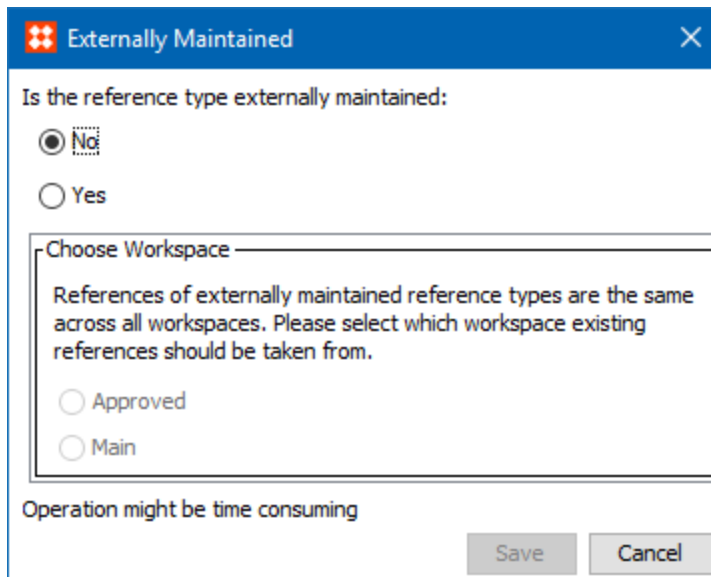
All fields except the ID and 'Last edited by' fields can be edited. The method required to edit a parameter in the references editor is determined by the parameter type:

- For **Name**, click into the text box and make the desired changes. The ID and name parameters are set in the **Reference Type - Enter ID and Name** wizard step.
- For **Externally Maintained**, double-click to display a dialog. When using an externally maintained reference between two objects, the reference will automatically be included in all workspaces and the source object will remain in its current approval status.

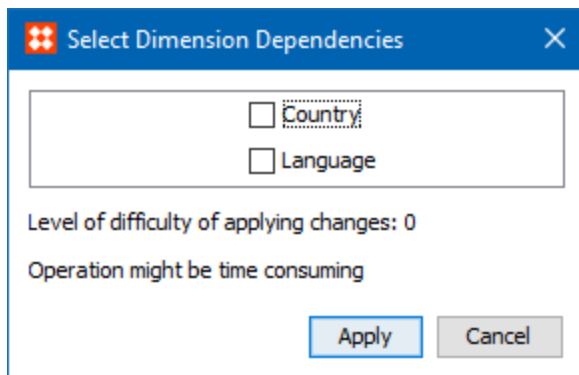
For example, references to vendor logos are imported from an ERP system and are not under revision control. To address this case, an image reference type named 'Vendor Logo' is set to be externally maintained, which means that:

- 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 unchanged.
- Changing metadata attribute values on attributes not set to be externally maintained and applied the reference type 'Vendor Logo' will trigger a status change on the products where the vendor logos are referenced from. Approval is required to see the changes in all workspaces.

If a reference type is changed to be Externally Maintained = Yes, 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 value. The externally maintained parameter is set in the **Reference Type - Advanced** wizard step.



- For **Dimension Dependencies**, double-click to display an ellipsis button. Click the ellipsis button (...) to display a dialog that includes the dimension points on your system. Only one dimension can be selected. If more are needed, create separate reference type. The dimension dependencies parameter is set in the **Reference Type - Apply Dimension Dependencies** wizard step.



- For **Allow multiple references**, click the parameter to display the dropdown list of options. Three (3) small dots displayed below the reference icon indicates multiple references are allowed (🔍). This parameter is set in the **Reference Type - Advanced** wizard step. For an example of using multiple references, see **Multiple References for the Same Objects**.
- For **Mandatory**, click the parameter to display the dropdown list of options. A red check mark displayed below the reference icon indicates it is mandatory for full approval (🔴). This parameter is set in the **Reference Type - Advanced** wizard step.

For example, a product reference type named 'Part of Kit' is set to be mandatory and is valid for source object types 'Product Family' and target object types 'Articles,' which means that:

- A reference of type 'Part of Kit' can be established from products with object type 'Product Family' to products with object type 'Articles.'
- A reference of type 'Part of Kit' must exist in order to approve a product with object type 'Product Family.'
- For **Inheritance**, click the parameter to display the dropdown list of options. A green down arrow displayed above the reference icon indicates it is inherited (👇). This parameter is set in the **Reference Type - Advanced** wizard step. For an illustration of inheritance on a reference type, see the **Inheritance Example for a Reference** topic.

Aspects Flipper

When an Entity, Context, or Workspace reference is used by a component model, the Aspects flipper is displayed. Component models, like solutions for GDSN or Matching, are housed in the System Setup > Component Models node.

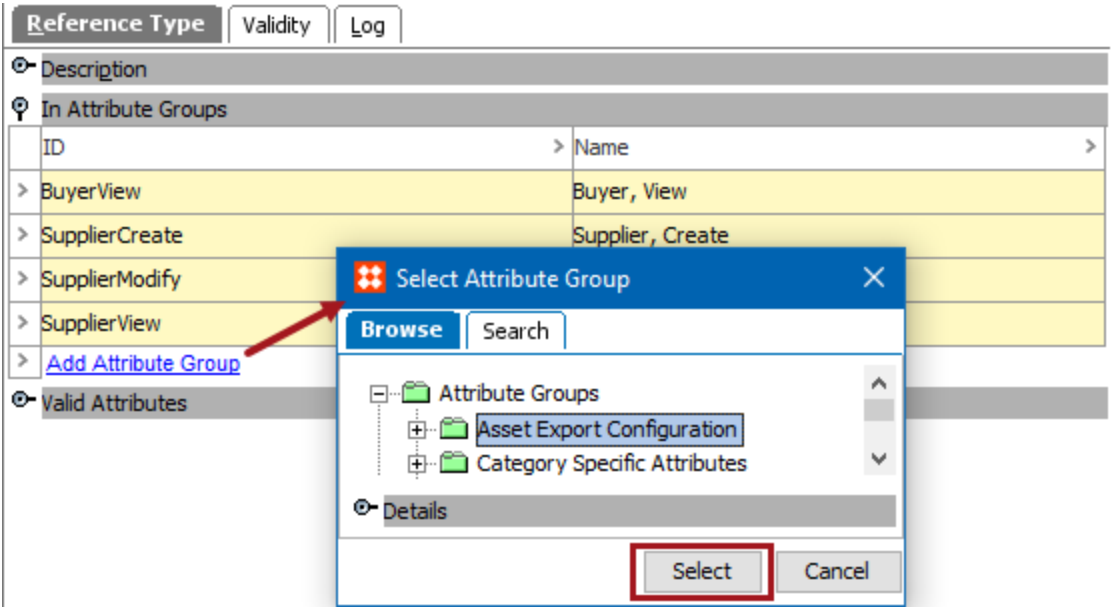
No edit option is available on the Aspects flipper. Changes to component model references must be made directly on the component model. Click the link in the Component column to display the component model using the reference.

Reference Type			Validity	Log
Description				
Aspects				
	Component	>	Name	>
	GDSN model		Target Market to Context Reference	Reference type that relates a Target Market with a Context
In Attribute Groups				
Valid Attributes				

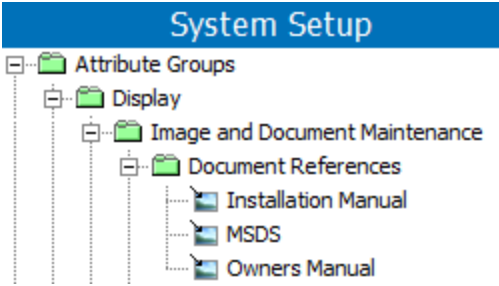
In Attribute Groups Flipper

The 'In Attribute Groups' flipper includes all attribute groups that include the reference. Adding a reference types to one or more attribute groups determines the label of the flipper that displays the reference in an object editor, enables you to set up user privileges to restrict approval of the reference type, and to define a customized view of the reference types in the group. For more information, see **Attribute Groups for Reference and Link Types** documentation.

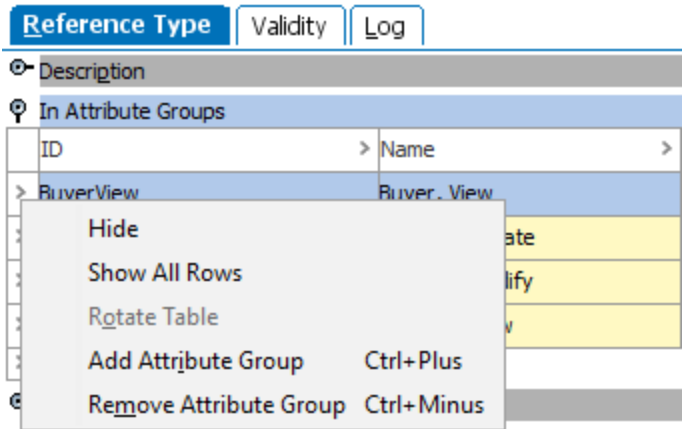
- To add an attribute group, click the **Add Attribute Group** link to display the Select Attribute Group dialog. Choose a group and click the Select button to add the reference to an attribute group.



The reference type is also displayed within the Attribute Groups node on System Setup.



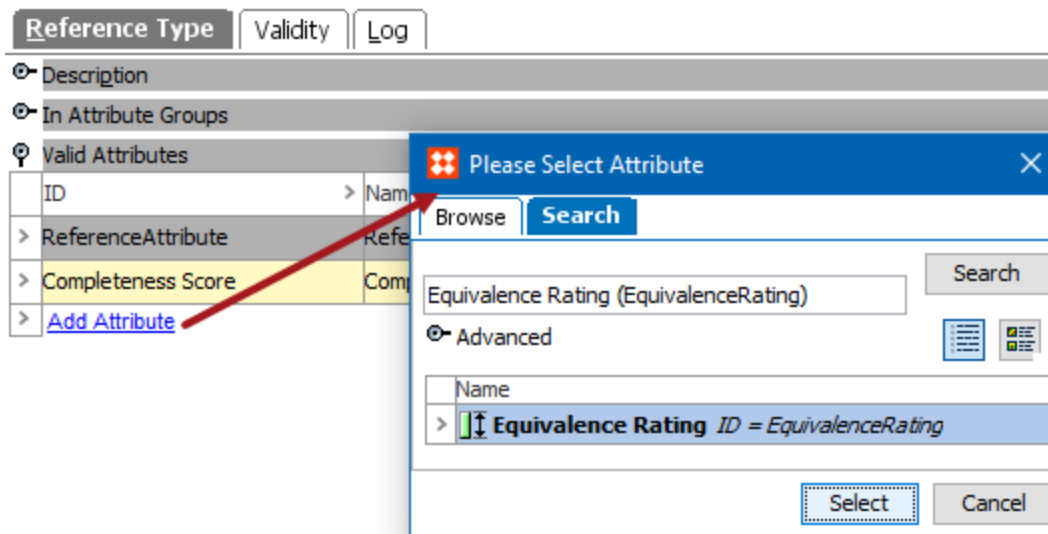
- To modify the attribute groups included or displayed, right-click the arrow to the left of the ID and select an option from the menu.



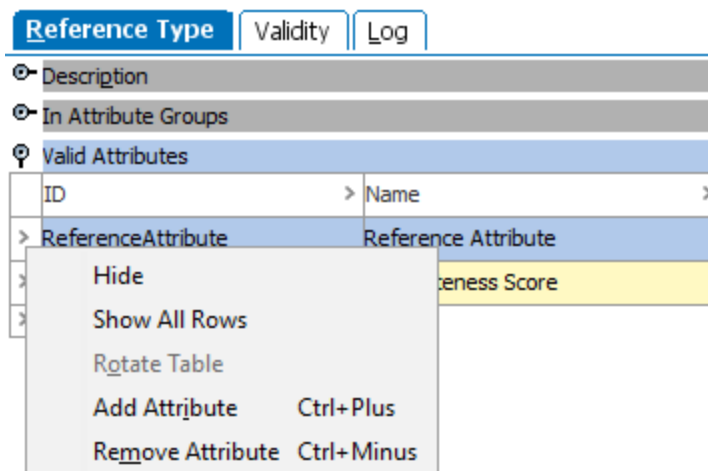
Valid Attributes Flipper

Metadata (description) attributes allow information to be captured on the reference itself to determine when the target object should be used, or to provide additional information about the connection between the objects.

- To add a metadata attribute, click the **Add Attribute** link to display the Select Attribute dialog. Choose a description attribute and click the Select button. The Search tab only returns results for description attributes. A specification attribute cannot be selected.



- To modify the metadata attributes included or displayed, right-click the arrow to the left of the ID and select an option from the menu.



For more information, see **Metadata Attributes on Reference and Link Types** documentation.

Validity Tab

The Validity tab stores the source object types and the target object types allowed on the selected reference type.

Reference Type	Validity	Log
Valid Source Types		
ID	>	Name >
> Item		Item
> ItemFolder		Item Folder
> SalesItem		Sales Item
> SalesItemFamily		SalesItemFamily
>		Modify Source Types
Valid Target Types		
ID	>	Name >
> ProductImage		Product Image
>		Modify Target Types

- For **Valid Source Types**, any number of source object types can be added, but the object types available are restricted based on the selected reference type. Click the 'Modify Source Types' link to add or remove an object type. This parameter is discussed in the **Reference Type - Select Valid Source Types** documentation.

Note: When an object type is removed from the list, if a reference of this type already exists for the deleted object type, the links are deleted.

To modify the object types included or displayed, right-click the arrow to the left of the ID and select an option from the menu.

Reference Type	Validity	Log
Valid Source Types		
ID	>	Name
> Item		Item
		Item Folder
		Sales Item
		SalesItemFamily

Hide
Show All Rows
Rotate Table
Modify Source Types Ctrl+Plus
Remove Row(s) Ctrl+Minus

- For **Valid Target Types**, any number of target object types can be added, but the object types available are restricted based on the selected reference type. Click the 'Modify Target Types' link to add or remove an object type. This parameter is discussed in the **Reference Type - Select Valid Target Types** documentation.

Note: When an object type is removed from the list, if a reference of this type already exists for the deleted object type, the links are deleted.

To modify the object types included or displayed, right-click the arrow to the left of the ID and select an option from the menu.

Reference Type	Validity	Log
Valid Source Types		
Valid Target Types		
ID	Name	
ProductImage	Product Image	
Hide Show All Rows Rotate Table Modify Target Types Ctrl+Plus Remove Row(s) Ctrl+Minus		

Log Tab

The Log tab stores the information about the changes that have taken place on the reference type since it was created.

Reference Type	Validity	Log
Showing page 1 of 1		
2016-09-22 12:12:37 'USERB': Dimension dependency modified : Removed 'Language' 2016-09-22 12:16:43 'USERB': Dimension dependency modified : Added 'Language' 2016-09-28 13:32:11 'USER4': Property modified 'TableHeader' from 'Image' 2016-09-28 13:33:31 'USER4': Dimension dependency modified : Removed 'Language' 2016-09-28 15:48:20 'USER4': Property modified Make reference type not-multi-valued 2016-09-28 15:59:11 'USER4': Property modified 'Purpose' from 'Used for the single product image that is the primary image to show in STEP and potentially in downstream systems' 2016-10-25 11:52:31 'USER7': Property modified Level4 added to valid source types 2016-12-09 08:49:00 'USER7': Name modified from 'Primary Product Image' 2016-12-09 08:49:52 'USER7': Name modified from 'PrimaryProductImage' 2016-12-19 10:18:17 'USER3': Property modified Make reference type multi-valued 2016-12-19 10:20:21 'USER3': Property modified Make reference type not-multi-valued 2017-02-28 21:07:22 'USER4': Dimension dependency modified : Added 'Language' 2017-03-21 14:24:46 'USERJ': Property modified InstallationManual added to valid target types		
First Page Previous Page Next Page Last Page Download...		

As additional log entries become available and the page is filled, the pagination buttons are enabled. Additionally, you can download the complete log as a zipped text file.

Attribute Groups for Reference and Link Types

Adding a Reference or Product to Classification Link type to an attribute group provides easy identification, a customized view, and control of approvals via user privileges.

Easy Identification

The name of the attribute group is used as the name of the flipper that displays the reference or link within the object editor. This provides a logical grouping for the references or links.

System Setup

The screenshot shows the 'System Setup' interface. On the left, under 'Attribute Groups', the 'Web Reference Types' group is selected. In the center 'Tree' view, 'Desk Lamps' is expanded, showing 'Desk Lamp Items' which includes 'Black Lamp', 'Green Lamp', and 'Orange Lamp'. On the right, the 'Black Lamp rev.0.1 - References' flipper is open, showing the 'References' tab. The flipper displays a table with columns: Product, Sub Products, References, Referenced By, and Images & Documents. The 'References' column shows 'Web Reference Types' and 'Web' with a plus icon.

Otherwise, the reference or link displays in a flipper with the 'Ungrouped' label.

The screenshot shows the 'Black Lamp rev.0.1 - References' flipper with the 'References' tab selected. A red box highlights the 'Ungrouped Classification Links' section. Below this, a table lists reference types and targets. The first row shows 'Basic Products' with a plus icon and a target 'Web Sites US/Website Root US/Office Products'.

Customized View

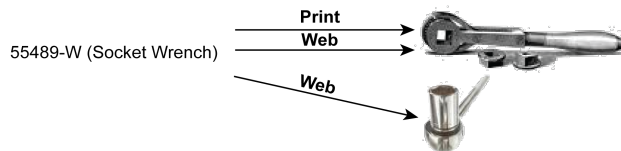
A customized view can be defined on the attribute group of all reference or link types in the group to display additional information on the reference or link. For more information, see the **Creating a Customized View** section in the **Attribute Groups** topic of the **System Setup / Super User Guide** documentation.

The screenshot shows the 'Black Lamp rev.0.1 - References' flipper with the 'References' tab selected. A red box highlights the 'Image References' section. Below this, a table lists reference types and targets. The first row shows 'Brand Name Logo' with a plus icon and a target 'Acme'. The table also displays a thumbnail of the 'Acme' logo, an 'ImagePriority' field, an 'ID' field with the value '168241', and a 'Name' field with the value 'Acme'.

Control of Approvals via User Privileges

Perhaps most importantly, an attribute group allows you to control user privileges for approval of the reference or link. This means that approval of specific types of references and links is based on the user's access to the attribute group, so that users will only be allowed to approve references or links of the types in the attribute group.

For example, consider a user who should only be allowed to approve references used for the web. As shown below, approving Product 55489-W (Socket Wrench) should also allow approving the images used for web.



To implement this, the reference type named 'Web' is added to an attribute group named 'Web Reference Types.'

A reference or link type can be added to as many attribute groups as needed by different users with different privileges.

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 set up to include the 'Approve Reference' privilege for reference types in the attribute group 'Web Reference Types.' For more information about setting up privileges, see **Action Sets** in the **System Setup / Super User Guide** documentation.

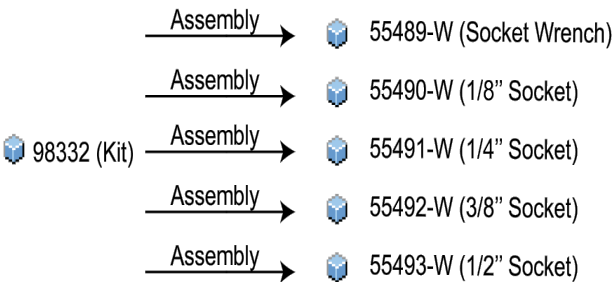
Direction of a Reference

Each reference type requires a direction that will identify one object as the 'source,' and another object as the 'target.'



For internally maintained references, directionality is important because it impacts the approval status of the objects involved. The source of the reference gets a change in approval status when the reference is added or removed, while the target approval status remains unchanged. This means that a source object can only be fully approved when all of the objects it references are also approved.

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 item in the set of products may also have an order code since they also could be sold as individual products.



In this example, a buyer could purchase only the socket wrench (order code 55489-W), only an individual socket (order code 55491-W), or a kit that includes the socket wrench and five individual sockets (order code 99332). 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.

In this case, order code 98332 is the source object, and the component parts order codes (e.g., 55489-W, 55490-W, 55491-W...) are the target objects.

Important: When creating a reference, verify that the 'from' and 'to' are assigned correctly. For example, 55489-W is an 'Assembly' of 98332, but 98332 is not an 'Assembly' of 55489-W.

The direction scheme for references makes it possible to build large kits or assemblies, the component parts of which are themselves assemblies, which themselves comprise of smaller assemblies, and so on. By always having a direction of a reference, i.e., a source / target scheme, the system can support rather complex assemblies.

Dimension Dependent Reference and Link Types

A Reference or Product to Classification Link type can be set up to be dependent on a dimension. Dimension dependency is used to specify the dimension points, and by extension, the contexts, where a reference or link should display or should be suppressed. A reference or link can only be dependent on a single dimension point. If additional dimensions are required, create a new type for the same source and target objects.

When deciding on the need for dimension dependency, consider the following points:

- You are strongly encouraged to give this topic a lot of thought when setting up the STEP system and before a lot of data is migrated in. It is easier to remove a dependency than to add the dependency after references have been created. See the **Adding a Dimension Dependency After Loading Data** topic for more information on what it takes to correct this or to move dimension-dependent references from the All level to a real language or country.
- If you set dimension dependency on an asset reference type, do not use it in conjunction with dimension-dependent assets. Although STEP can handle this scenario, maintaining and troubleshooting is complicated when multiple dependencies are in use. For more information on dimension-dependent assets, see the **Asset Dimension Dependencies** section of **Maintaining Assets** in the **Getting Started / User Guide** documentation.
- Consider using Users and Groups privileges to hide objects by Reference Type, instead of using dimension dependency. For more information, see the **Attribute Groups for Reference and Link Types** topic.

Dimension Dependency on the 'Reference-Type' Object Type

Dimension dependency can be added on the 'Reference-Type' object type, which is located under System Setup > Object Types & Structures > Basic Object Types.

The screenshot displays the 'Reference-Type - Object Type' configuration window. On the left, the 'System Setup' tree has 'Reference-Type' selected. The main area shows a table with the following data:

Description	
Name	Value
ID	Reference-Type user-type root
Name	Reference-Type
Last edited by	2016-10-11 11:34:15 by USER4
Name Pattern	
ID Pattern	
Icon	
Dimension Dependencies	Language;

Using the Dimension Dependencies parameter on the Reference-Type object type is a global setting that enables translation of reference type names and requires that all reference type names be translated. Translated reference type names are useful on websites where viewers want to see information in their own language. For example, 'Installation Guide' could be the reference type and the link that appears when someone needs to download a PDF version of a document.

Setting a Reference-Type object type dimension dependency does not affect the dimension dependencies between referenced objects.

Dimension Dependency on a Reference Type

Setting the Dimension Dependencies parameter on a reference type adds dimension dependency on the reference between objects. This can be useful for a global website structure where not all products are sold in all countries, or where documents vary by language or country and users only want to see the appropriate versions.

The screenshot shows the 'System Setup' interface. On the left, a tree view under 'Reference Types' shows 'Primary Product Image' selected. On the right, the 'Primary Product Image - Reference Type' configuration window is open. It has tabs for 'Reference Type', 'Validity', and 'Log'. The 'Reference Type' tab is active, showing a table of properties.

Name	Value
ID	PrimaryProductImage
Name	Primary Product Image
Last edited by	2016-12-19 10:20:21.0 by USER3
Externally Maintained	No
Dimension Dependencies	Language;
Allow multiple references	No
Mandatory	No
Inheritance	Inherited

This is not a typical setup because the References / Referenced By tabs can become difficult to manage due to the layout required when viewing multiple contexts.

One scenario where this setup would be useful might be when there are 20 documents in STEP, one for each of the 20 countries where a product is sold. A dimension-dependent reference would make it possible for a user working in a country context to only see the document that is used for that country since the rest would be filtered from view.

For information on managing when a dimension dependent reference is available, see the **Visibility of Reference and Link Types** section.

Viewing Dimension Dependent Values

When dimension dependency is applied to a reference type, values can be applied based on the selected dimension. For example, the following reference is dependent on the Country dimension, so each country can have a different setting. The Context view displays all values for each country, based on the targets selected from the View > Target menu option.

Context English UK

URL I&contextid=Context7&id=20866&workspaceid=Main

Goto ID/Name

Tree

Primary Product Hierarchy

- Products
 - Apparel
 - Hardware
 - Tools
 - Task Lighting
 - Flashlights
 - Flashlights Items
 - Flashlights SalesItems
 - 20866-012

No Title

Compare ContextsReferencesReferenced By

Attribute Group

Reference Type: Alternate Supplier Item

Context	Reference Type	Target	Visibility	
> French FR	+ Alternate Supplier Item	(114861)	France	X
> English UK	+ Alternate Supplier Item	20862	Great Britain	X
> Danish DK	+ Alternate Supplier Item	Green light	Denmark	X

Discontinued Product Maintenance

Entity Reference Types

When an entity reference type has only entity object types for source and only entity object types for target, the 'Parent/Child relation' parameter is available. This allows an aggregated display of all entity-to-entity reference types in the Tree hierarchy navigator.

Note: If another super type (e.g., products or classifications) is present on the Validity tab of the reference, the parameter is not available.

Send to - Reference Type		
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

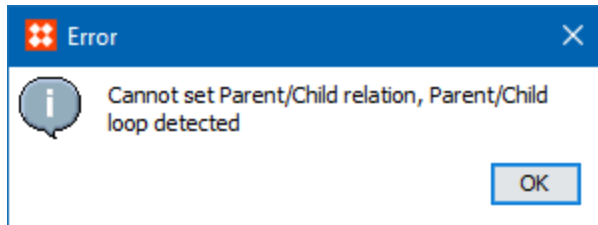
The **Parent/Child Relation** field includes these options, which are illustrated in the 'Parent/Child Relation Example' section below:

- **None** means entity-to-entity references are not aggregated.
- **Source as parent, Target as Child** means entity-to-entity references are aggregated below the source entity (where references originate).
- **Target as parent, Source as Child** means entity-to-entity references are aggregated below the target entity (where references end).

When setting the Parent/Child Relation field, be aware of these items:

- Use different icons for each entity object type to make the hierarchy overview easier to understand. See the **Object Type Icons** section of the **System Setup / Super User Guide**.
- Aggregating references enables **Copy**, **Paste**, **Drag**, and **Drop** between entity hierarchies. STEP will automatically apply the reference at the new position.

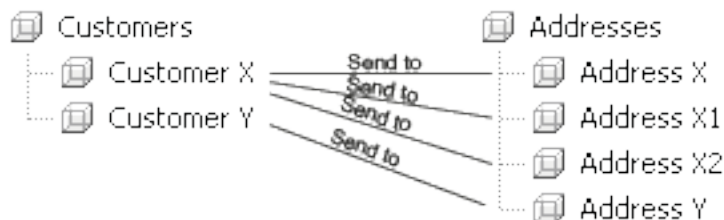
- Entity objects types can be set up to be globally revisable or internally revisable. For more information about this functionality, see the **Revisability on Entity Object Type** documentation.
- STEP prevents you from setting the Parent/Child Relation field if the objects that are parent and child of the reference already have a true parent / child relationship via the data model. If either object type appears under the object type References tab > Parents flipper, an error is displayed and a parent/child relation is not allowed.



Parent/Child Relation Example

Consider an example that includes the following data: a Customers entity hierarchy, an Addresses entity hierarchy, and a 'Send to' reference type which is used to link each customer to its relevant address(es).

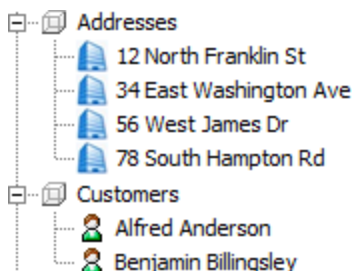
The image below shows the following relationships: Customer X has three Addresses, and Customer Y has one Address. Each of the Addresses (target) are referenced from Customer (source) using the 'Send to' reference type.



This same data can be displayed differently on the Tree, based on the 'Parent/Child Relation' parameter setting in the 'Send to' reference editor. Customer is the source and Address is the target since a reference is created from a customer to an address.

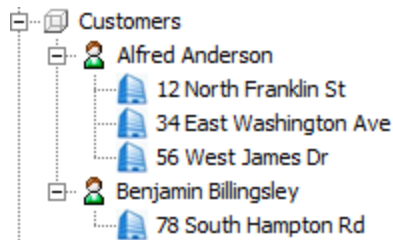
None

On the Tree, expanding the Customers entity hierarchy displays only customer objects, and expanding the Addresses entity hierarchy displays only address objects, as shown in the image below:



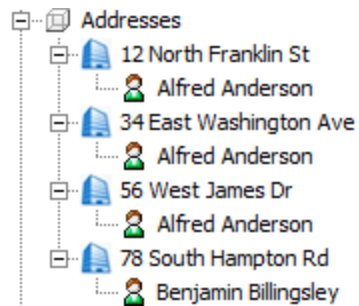
Source as parent, target as child

On the Tree, expanding the Customers entity hierarchy displays all referenced Address entities, as shown in the image below:



Target as parent, source as child

On the Tree, expanding the Customer entity hierarchy displays all referenced Address entities, as shown in the image below:



Inheritance Example for a Reference

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:

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

Product References

Supplier Replacement Item

< Inherited, valid on child - Validity

Reference Type

Validity

Log

Valid Source Types

ID

Name

> Item

Item

> ItemFolder

Item Folder

> Modify Source Types

Valid Target Types

ID

Name

> Item

Item

> Modify Target Types

When both references are set with inheritance, any reference added to the parent (Item Folder object) will also be present on the child (Item object).

Note: Inheritance is indicated by the small green down arrow above the reference type icon (📁). The three (3) dots below the icon indicates that multiple references of this type are allowed.

Parent object:

Tree

- Publications
 - Primary Product Hierarchy
 - Products
 - Apparel
 - Footwear
 - Safety
 - Hand Protection
 - Safety Gloves
 - Safety Gloves Items
 - 20726 (box of 5 pair orange)
 - 21873 (box of 5 pair multi-color)
 - 21882 (sellable case of 20 boxes)

Safety Gloves Items rev.0.4 - References

Category ProfileProof ViewStatusState LogTasks

Referenced ByImages & DocumentsCommercialTables

ProductProduct VariantsSub ProductsReferences

Image References

Sales Item References, Classification

Ungrouped Product References

Reference Type	Target
Inherited, not ...	21873 (box of 5 pair multi-color)
Inherited, valid...	21873 (box of 5 pair multi-color)

Child object:

Tree

- Publications
 - Primary Product Hierarchy
 - Products
 - Apparel
 - Footwear
 - Safety
 - Hand Protection
 - Safety Gloves
 - Safety Gloves Items
 - 20726 (box of 5 pair orange)
 - 21873 (box of 5 pair multi-color)
 - 21882 (sellable case of 20 boxes)
 - 22126 (box of 20 pair orange)

20726 (box of 5 pair orange) rev.0.11 - References

Proof ViewStatusState LogTasks

Images & DocumentsCommercialTablesCategory Profile

ProductSub ProductsReferencesReferenced By

Ungrouped Product References

Sales Item References, Classification

Sales Item References, Product

Ungrouped Classification Links

Ungrouped Product References

Reference Type	Target
Inherited, not valid on child	21873 (box of 5 pair multi-color)
Inherited, valid on child	21873 (box of 5 pair multi-color)

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:

Tree

- Publications
 - Primary Product Hierarchy
 - Products
 - Apparel
 - Footwear
 - Safety
 - Hand Protection
 - Safety Gloves
 - Safety Gloves Items
 - 20726 (box of 5 pair orange)
 - 21873 (box of 5 pair multi-color)
 - 21882 (sellable case of 20 boxes)
 - 22126 (box of 20 pair orange)

20726 (box of 5 pair orange) rev.0.11 - References

Proof ViewStatusState LogTasks

Images & DocumentsCommercialTablesCategory Profile

ProductSub ProductsReferencesReferenced By

Sales Item References, Classification

Sales Item References, Product

Ungrouped Classification Links

Ungrouped Product References

Reference Type	Target
Inherited, not valid on child	
Inherited, valid on child	

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.

In the wizard, you can specify different settings for the inheritance of reference types (No Inheritance, Normal Inheritance, and Accumulative Inheritance) as defined in **Reference Type - Advanced**.

The wizard selections are reflected in the reference editor Inheritance parameter (None, Inherited, Accumulative) as defined in **Maintaining a Reference Type** documentation.

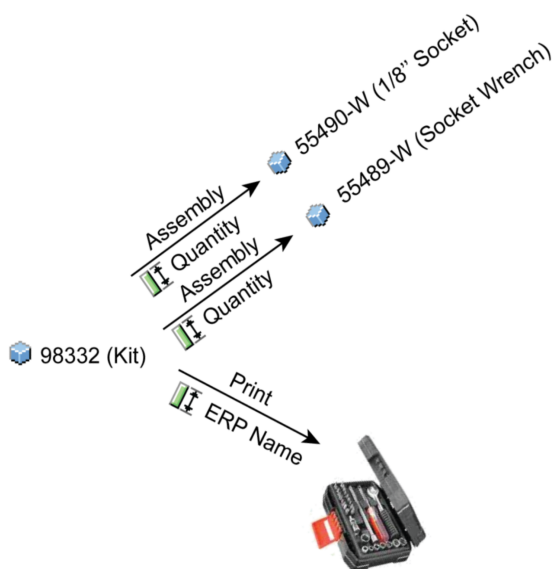
Metadata Attributes on Reference and Link Types

Metadata (description) attributes can be applied to all objects in the System Setup > Reference Types folder, including a Reference type, a Product to Classification Link type, the Classification Attribute link type, and the Product Attribute link type.

Metadata is maintained on the References or Referenced By tabs of the object where the reference or link is applied. Different attributes can be selected for each type of reference or link type.

Example of metadata on a reference or link

Consider product 98332 (KIT), which includes two sockets and is referencing an image to be used for Print. Different metadata attributes are needed for references applied to images and the references applied to products.



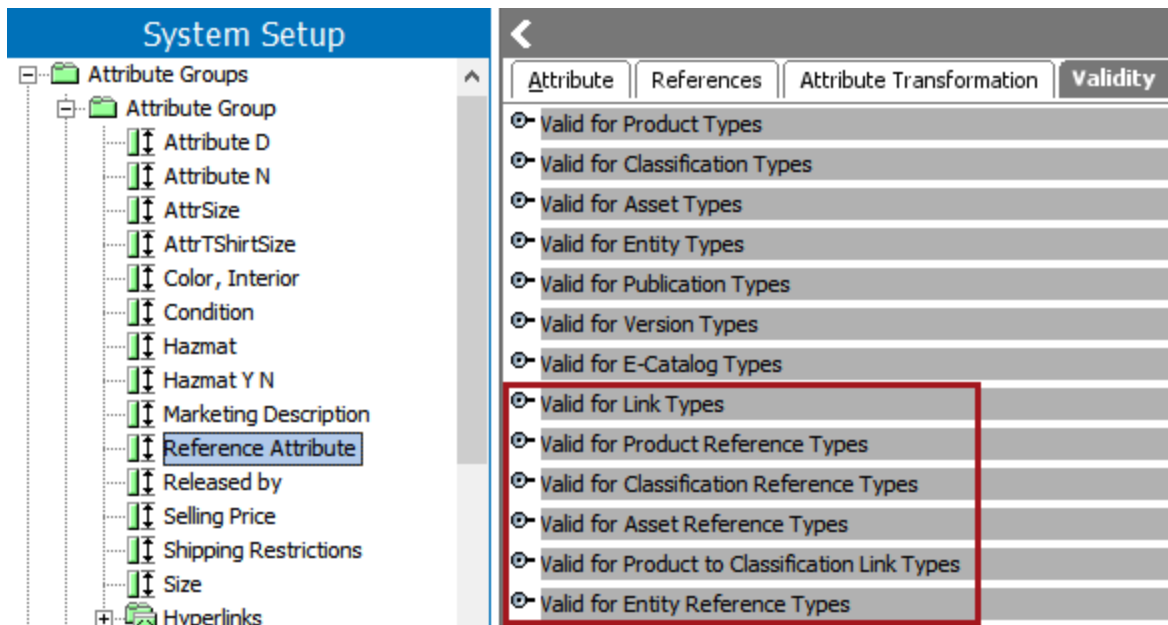
- A description attribute named Quantity has been applied the reference type Assembly. This attribute holds number of sockets to be included in the kit.
- A description attribute named 'ERP Name' has been applied the reference type Print. This attribute holds the name of the reference used in an ERP System.

For information on adding metadata to the attribute link types, see **Product Attribute Link Type** or **Classification Attribute Link Type**.

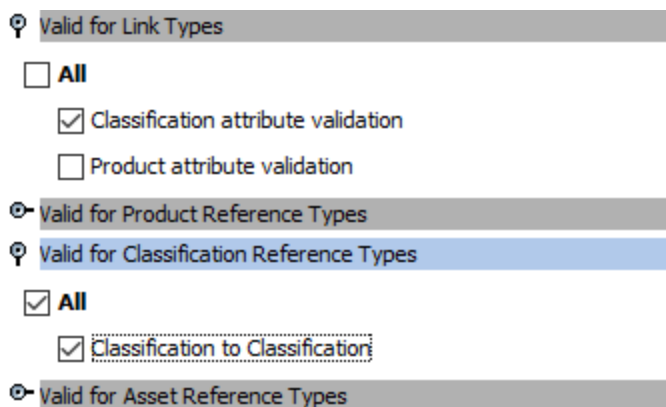
Add a Metadata Attribute

Although the images below show a reference type, the same setup is used for adding metadata on a Product to Classification link type.

1. In System Setup > Attribute Groups > select the description attribute that should be added as metadata on the reference or link. Specification attributes and groups cannot be added as metadata.
2. Open one (or more) reference types flippers or the link types flippers.

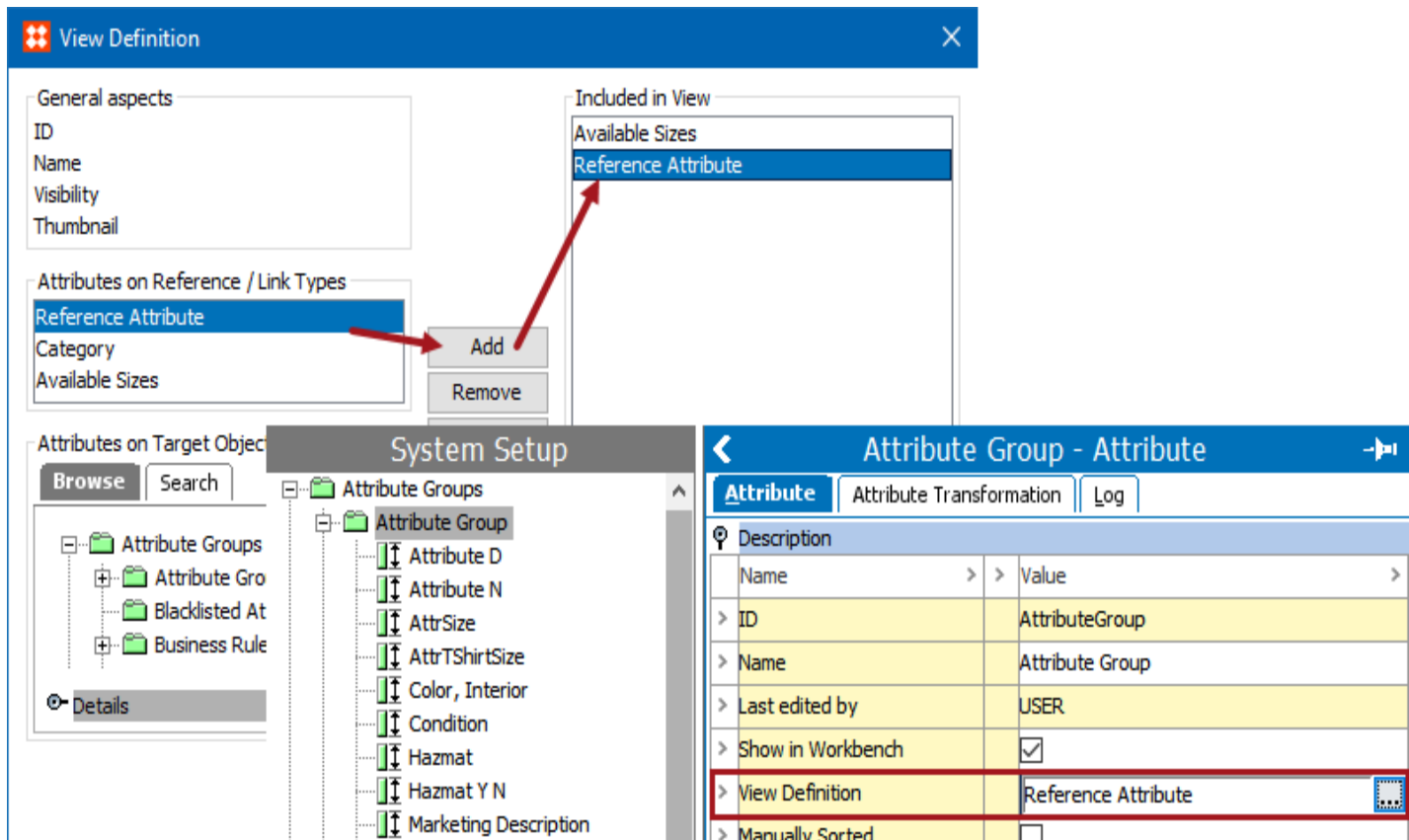


3. Select all reference types and link types that should display the metadata attribute. Checking an option here also updates the Attributes flipper of the selected reference or link type.

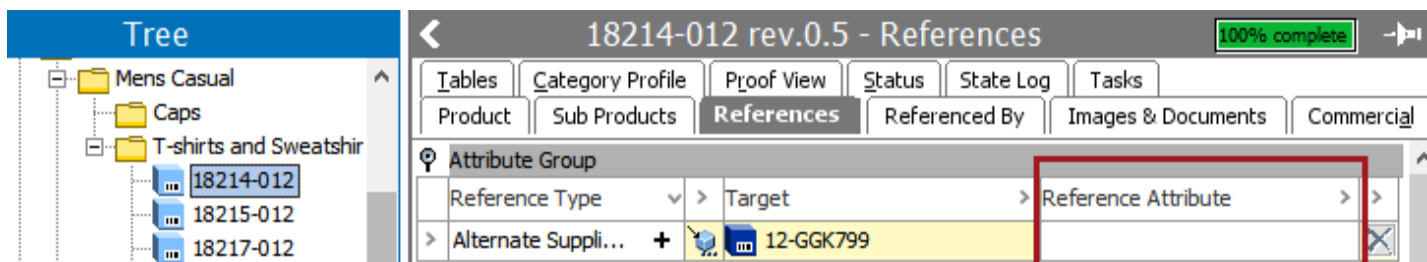


For more information, see **Validity on Description Attributes** in the **System Setup / Super User Guide**.

4. Add the description attribute from the 'Attributes on Reference / Link Types' section to the attribute group's View Definition so it will display in the object editors as metadata on any reference or link in the group. For detailed steps, see **Creating a Customized View** section in the **Attribute Groups** topic of the **System Setup / Super User Guide** documentation.



5. Verify that the metadata attribute is available on the reference.



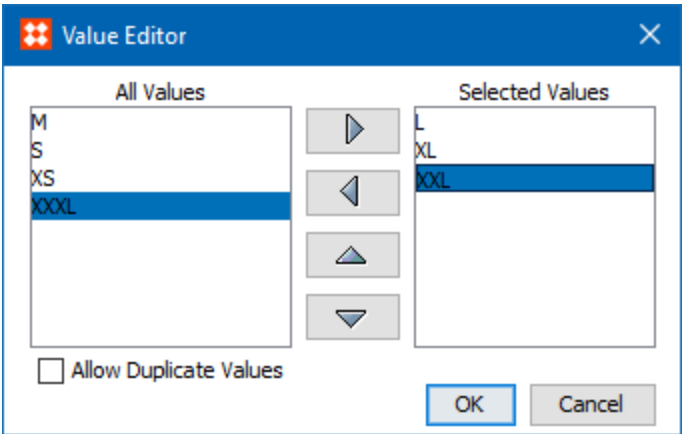
Edit a Metadata Attribute

Although the images below show a reference type, the same setup is used for editing metadata on a Product to Classification link type.

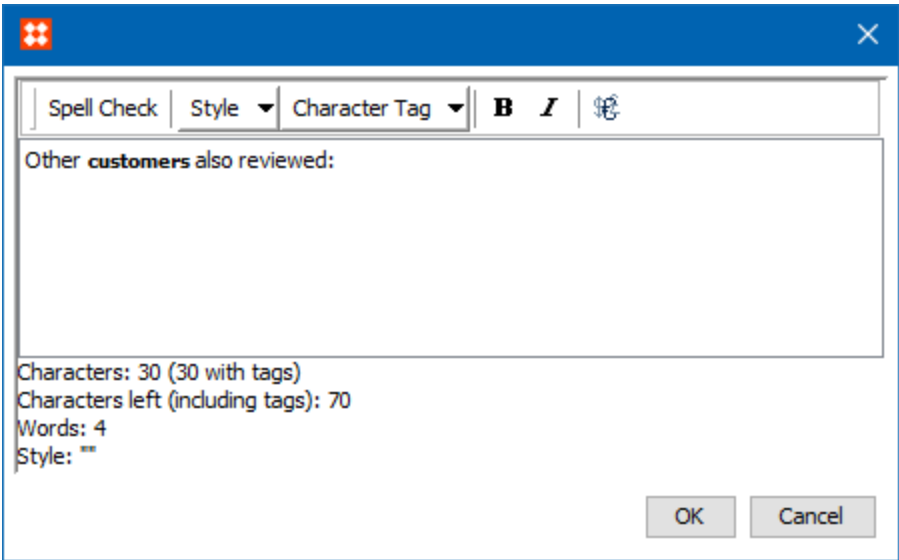
On the **References** tab, double-click an editable field (any white field as shown above) and edit using the available functionality based on the type of the metadata attribute.

- Modify the value directly in the field by adding text or selecting from a dropdown.
- Double-click (or right-click > Edit) to display a value editor.

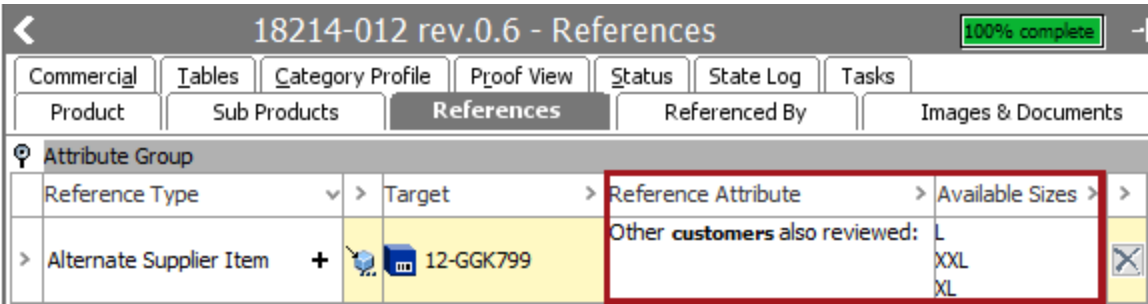
Choose a value and use the necessary arrow button to include it in the Selected Values list, or to rearrange the order of the values.



Add or modify text, including styles, character tags, and spell check.



Edits made to metadata on a reference or link are displayed:



Multiple References for the Same Objects

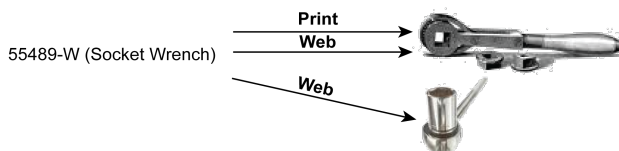
It is possible to have multiple references of the same type between the same two objects, by creating a separate reference type for each need. Typically, such references may also have different restrictions applied. In this way, the same product could be linked to the same image by multiple product-to-asset references.

For example, a socket wrench with order code 55489-W should be represented by a single image for print, and should be represented by multiple images for web.

To establish such restrictions, two (2) asset reference types are created:

- The asset reference type named 'Web' allows multiple references to any number of images.
- The asset reference type named 'Print' only allows a single reference to a single image.

As shown in the following image, the same wrench image is referenced with asset reference type Print and with asset reference type Web from the socket wrench product 55489-W.



For more information, see the **Allow multiple references** parameter section of **Maintaining a Reference Type** documentation.

Visibility of Reference and Link Types

On dimension dependent references and link types, you can specify the visibility of a reference or link in a specific context. Based on the visibility, you can make a single setting and allow it to be inherited by other contexts, make local settings on some contexts, make all contexts have a local setting, or suppress the reference completely for a context.

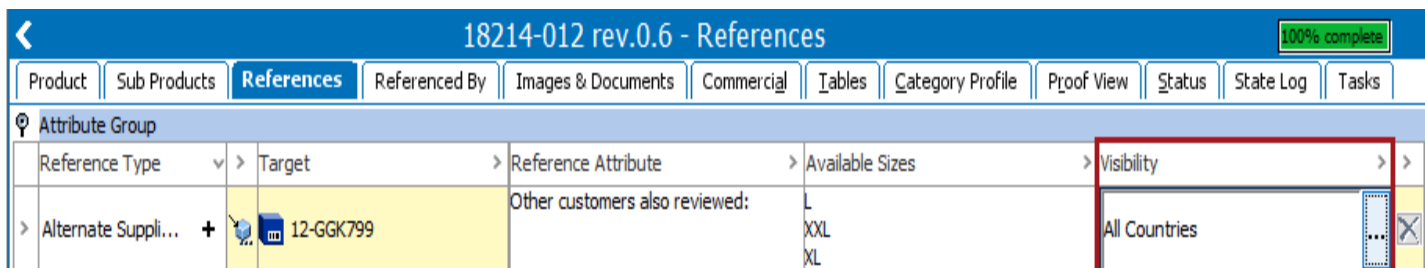
- Making the visibility global means the reference settings are inherited from the global context. For example, when all contexts require the same setting.
- Making the visibility local means the reference does not inherit its settings from the global context. For example, if you want to define attributes on a language-by-language basis.
- Suppressing the visibility means it will not be used in a specific context.

Metadata 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.

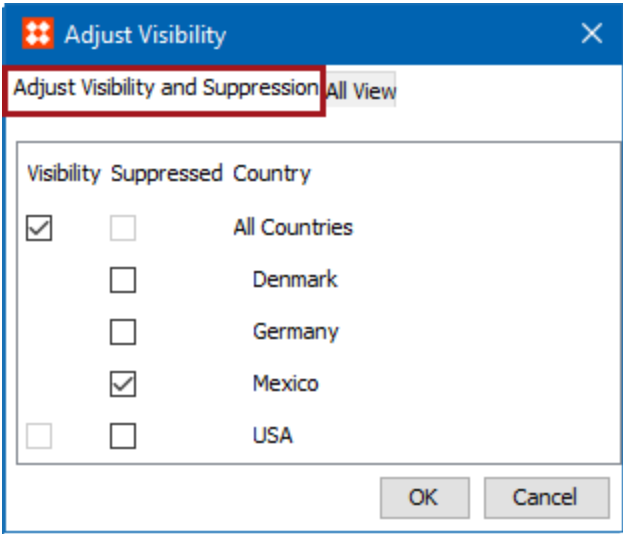
Note: The Visibility column is displayed on a reference when the General Aspect option 'Visibility' is added to the 'Included in View' panel on the View Definition of the attribute group. For more information, see the **Add a Metadata Attribute** section of the **Metadata Attributes on Reference and Link Types** documentation.

Setting Reference and Link Type Visibility

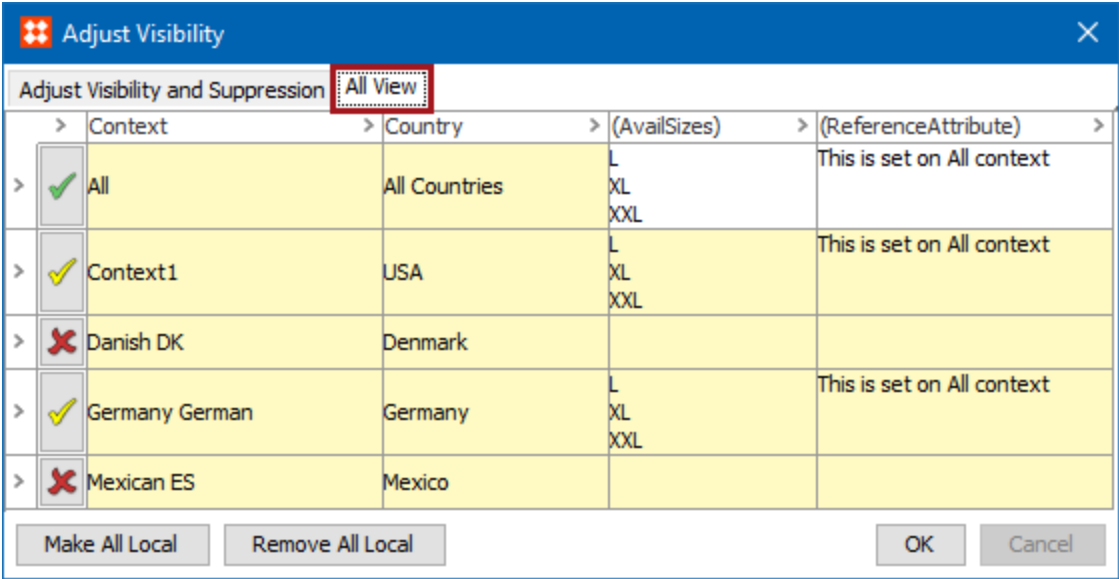
1. In the Tree, select the relevant node, click the **References** tab, and open the flipper that contains the dimension-dependent reference to be edited.



2. Double-click the **Visibility** field to display the ellipsis button, and then click the ellipsis button (...). The **Adjust Visibility** dialog displays with the 'Adjust Visibility and Suppression' tab active.

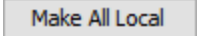


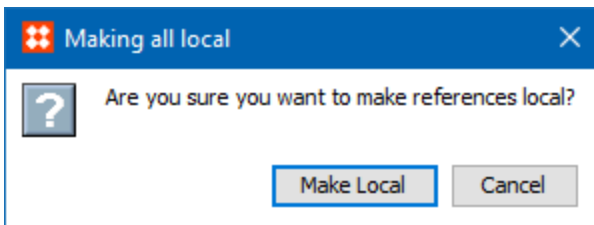
- 3. In the Suppressed column, check any contexts that should not display the reference.
- 4. Click the **All View** tab and note the status of the visibility of each context:
 - indicates that the metadata attribute is local and not being inherited (All). The setting is inherited by the contexts that display a yellow check.
 - indicates that the contexts inherits the settings of the context with the green check (USA and Germany).
 - indicates that the reference is suppressed (Mexico), or does not exist in the context (Denmark).

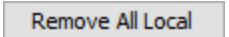


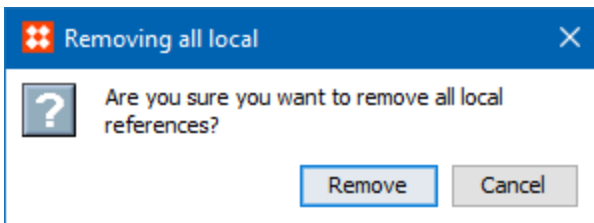
- 5. Modify the visibility of each required context as needed:
 - Click or to make the visibility of a reference local for only the selected context. Assuming the user has privileges to create the reference, the indicator changes to , the reference value field changes to a white

background, indicating that the setting can be modified on the selected context.

- Click  to remove the local reference for only the selected context. The indicator changes to , and inheritance is restored from the global context, and the reference value field changes to a yellow background, indicating that the setting cannot be modified on the selected context.
- Click  to make the visibility of the reference local for all contexts. In the 'Making all local' dialog, confirm that you want to make all local by clicking the **Make Local** button. All check marks turn green except the icon next to the contexts that were previously local (green). Previously local contexts are set with a red X, indicating that the reference no longer appears in those contexts. When the visibility of a reference is set to local, it no longer inherits its settings from the global context. This allows you to specify, for example, attribute values that only apply in the local context.



- Click  to restore inheritance for all contexts currently local. In the 'Removing all local' dialog, confirm that you want to restore inheritance by clicking the **Remove** button. If a global context is local, all check marks turn yellow except the icon for the contexts that were previously local (green). Previously local contexts are set with a red X, indicating that the reference no longer appears in those contexts. When the visibility of a reference is set to global, it inherits its settings from the global context.



- Verify the results of the visibility by checking the References page on the object and make any additional changes. Use the Context Mode by clicking  in the toolbar.

Note: Navigate to View > Target to select the Contexts for the comparison view to display.

Compare Contexts

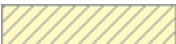
References

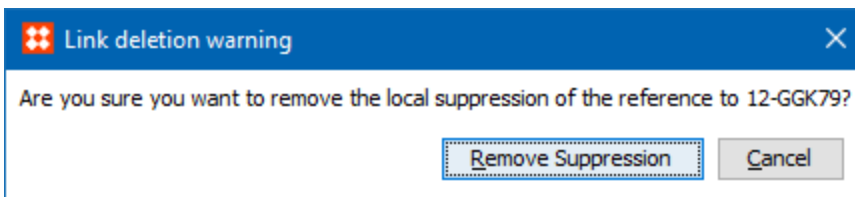
Referenced By

 (AttributeGroup)

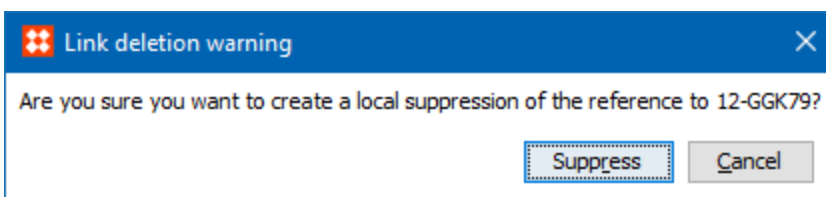
Reference Type (Alternate Supplier Item) ▾

Context	Reference Type	Target	(ReferenceAttribute)	(AvailSizes)	Visibility	
> Germany German	+ (Alternate Supplier Item)	 12-GGK79	This is set on All context	L XL XXL	All Countries	
> All	+ (Alternate Supplier Item)	 12-GGK79	This is set on All context	L XL XXL	All Countries	
> Mexican ES	+ (Alternate Supplier Item)	 12-GGK79			Suppressed in [Mexico]	
> Context1	+ (Alternate Supplier Item)	 12-GGK79	This is set on All context	L XL XXL	All Countries	
> Danish DK	(Alternate Supplier Item)					

-  indicates the reference is allowed for the context. Note that reference is not available for the Danish DK context in the image above.
-  indicates that the reference is local and values can be added for metadata.
-  indicates that the reference is inherited and values cannot be added for metadata.
-  indicates that the reference has been suppressed.
-  allows you to remove the suppression for the selected context. In the 'Link deletion warning' dialog, confirm that you want to remove the suppression by clicking the **Remove Suppression** button. Previously set inheritance is restored. Return to the Adjust Visibility dialog to change visibility to local if necessary.



-  allows you to remove the current setting and suppress the reference for the selected context. In the 'Link deletion warning' dialog, confirm that you want to create a local suppression by clicking the **Suppress** button.



Regular Expression

Several areas in STEP allow the use of a regular expression including validation templates, transformations, gateway endpoint configuration, and asset importer. While it is not required to be an expert, having some familiarity with the concept will help users to better understand the configuration options available.

To put it simply, a regular expression is shorthand for describing patterns that can be matched by text.

This can be extremely useful when describing, for example, how a file name should be used to match an existing asset or product object, which part of a folder structure should be processed or ignored, and many other scenarios related to matching text patterns.

The following additional information is available:

- The **Regular Expression Syntax** topic includes information on the syntax required and used in STEP regular expressions.
- The **Regular Expression Examples** topic includes practical examples using regular expressions in STEP.

Versions

Within STEP, different versions of regular expression are used, as follows:

STEP Area	RegEx Version	Detailed Information
Asset Importer Calculated Attributes Search Table Transformations Transformations Validation Rules	Java	https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html
Business Rules	JavaScript	https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions

Regular Expression Syntax

A regular expression is a very powerful (and sometimes complicated) tool, and a complete understanding of their capabilities and applications goes well beyond this primer. Instead, some standard expressions that are typically used and the rules below must be understood in order to read a regular expression, including:

- Characters
- Repetition modifiers
- Groups and ranges
- Escaping special characters
- Anchors

Characters

The following parts of a regular expression allow you to indicate both matches for characters and character classes.

Matches

To specify that a certain character or character set can be matched, use the following options.

- **Literals:** Any character matches itself.
Example: 'a' matches the character 'a'.
- **Dot / Period (.):** Matches one single character (any character).
Example: a..d
Matches: 'abcd', 'a12d', and 'aad'
Does Not Match: 'abbb', 'abcde', or 'ba1d'
- **Character Sets:** Uses brackets ([]) to match any one of a set of characters.
Example: [ae]
Matches: 'a' and 'e'
Does Not Match: 'b', 'c', or 'd'
Example: [12345]
Matches: '1', '2', '3', '4', and '5'
Does Not Match: '0', '6', '7', '8', or '9'
- **Ranges:** Use a dash (-) to indicate ranges within a Character Set.

Example: [a-e]

Matches: 'a', 'b', 'c', 'd', and 'e'

Does Not Match: '1' or 'f'

Example: [1-5]

Matches: '1', '2', '3', '4', and '5'

Does Not Match: '0', '6', '7', '8', or '9'

- **Combinations**

Example: [a-z][a-z0-9][a-z0-9]

Matches any lowercase letter followed by two other lowercase letter or numbers

Matches: 'a11', 'bt9', and 'xyz'

Does Not Match: '1ab' or 'abc1'

Classes

To specify that the text must be of a certain class, use the following options.

- **Digit:** Any number character 0-9.

\d indicates a digit character is required.

\D indicates a digit character must not be present.

Example: '\d' matches the number '5' in the value '5 = V'.

Example: '\D' matches the equal sign '=' in the value '5 = V'.

- **Word:** Any word character: a-z, A-Z, or 0-9.

\w indicates a word character is required.

\W indicates a word character must not be present.

Example: '\w' matches the number '5' in the value '5 = V'.

Example: '\W' matches the equal sign '=' in the value '5 = V'.

- **White space:** Any non-whitespace character.

\s indicates space character is required.

\S indicates space character must not be present.

Example: '\s' matches the space between the two words in the value 'this is'.

Example: '\S' matches the character 't' in the value ' this is'.

Repetition Modifiers

To specify that a certain character or character set can be matched more than once, you can add repetition modifiers after the character or character set. Some common repetition modifiers are described below.

- **+**: One (1) or more; means 'there must be at least one of these'

Example: `[a-z]+[0-9]`

Matches anything starting with one or more lowercase letters, followed by a single number

Matches: 'abc1', 'a1', and 'abcxyz9'

Does Not Match: '9a', 'a22', or 'a-4'

- **?**: Zero (0) or one (1); means 'this is optional'

Example: `[a-z]+[0-9]?`

Matches anything starting with one or more lower-case letters, optionally followed by a single number

Matches: 'abc', 'abc2', 'x', 'y3', and 'abcdefgh8'

Does Not Match: '9', 'a99', or 'a-4'

- *****: Zero (0) or more; means 'have as many as you want of these, or none at all'

Example: `.*`

Any number (0 or more, due to the '*') of any character (due to the '.')

Will match any text

Matches: 'a1b2c3', 'aaaa', and '111'

- **{min, max}**: Between the minimum and maximum occurrences; means 'have between X and Y of the indicated character / set / range'; if max is omitted, this means exactly min number of times

Example: `[0-9]{5}-[0-9]{4}`

Simple zip code + 4 matcher

Matches: '12345-0123'

Does Not Match: '12345' or '6789012'

Groups and Ranges

Grouping is not related to matching, but is used to store certain matched sets of characters for use later, typically in a substitution.

- **Grouping:** Done using parenthesis () to isolate groups for storage

Example: `([0-9]{5})-([0-9]{4})`

Stores the first five digits in group 1

Stores the last four digits in group 2

The dash (-) character is not stored as it is not inside parenthesis

- **Referencing Groups:** Done using a dollar sign (\$) to indicate a group

Example: \$1

Use the content from group 1

With the expression '([0-9]{5})-([0-9]{4})' and value '12345-0123' the results are \$1 = '12345' and \$2 = '0123'

- **|:** Or condition; means either must be true

Example: [a-z][0-9]

Matches anything with one lowercase letter OR one numeric digit

Matches: 'c' and '8'

Does Not Match: 'X' or '22'

Escaping Special Characters

There are several characters that have special meanings, such as `?`, `.` (dot), `*`, `+`, `(, )`, `{, }`, and more. This prompts the question: how can you match a literal example of one of these characters? The key is to escape the special character by putting a backslash (\) in front of it.

- **Escaping** Done using backslash (\) to indicate retention of the literal; if you need to match a backslash, escape it in the same way: \\

Example: \.

Retain the dot (.)

Example: .*\\.*

Match anything that has a number of characters, followed by a period, followed by any number of characters

Used to match file names with extensions (e.g. abc123.pdf)

Can also be used to match any text (e.g. 'Hello there this is matched')

Anchors

The following characters indicate the beginning or end of an expression.

- **^:** Start of a string, or start of a line in a multi-line pattern
- **\$:** End of a string, or end of a line in a multi-line pattern

Example: ^[1-5][0-9]\$|^[0-9]\$

Two strings are being considered for matching. The first string allows a single digit from 1 - 5, and a second digit from 0 - 9. The pipe (|) symbol indicates an 'or' condition, meaning either the first or second string can be matched. The second string allows for a single digit from 0 - 9.

Regular Expression Examples

A regular expression can be used in several places throughout STEP. The examples below are some of the common uses and are divided into the following areas:

- Asset importer
- Attribute transformations
- Import transformations
- Search
- Table Transformations
- Validation base type

Asset Importer

The following examples involve requirements for file name matching as used in Asset Importer.

For more information on asset importer, see the **Asset Importer** section of the **Digital Asset Exchange** documentation.

Match a file name and save the part before the extension

- **Expression:** `(.*)\..*`
- **Value:** 'abc123.pdf'
- **Result:** `$1 = 'abc123'`

New Asset Import Configuration

Steps

1. Identify Config
2. Import Validator
3. Hierarchy Builder
- 4. Asset Matcher**
5. Content Importer
6. Metadata Importer
7. Product Linker
8. Approver
9. Auto Purger
10. Workflow Handler
11. Business Rules

Asset Matcher

Asset Matcher: Asset Matcher

Asset Matcher Hierarchy Root *: 9 (20138)

Match on *: File Name

File Name Match Expression *: (.*)\\.*

Match Template *: \$1

ID Template: \$1

Name Template: \$1

Multiple Matches Found: Error on record

* Mandatory Fields

Back Next Finish Cancel

Match the last folder in a path

- **Expression:** .*/(.*)
- **Value:** '/net/share/imageupload/product_images'
- **Result:** \$1 = 'product_images'
- **Notes**
 - To understand why it doesn't yield \$1 = 'share/load/product_images', it is necessary to understand that by default, the repetition modifiers are 'greedy' and try to match as far as they can while still resulting in an overall match. There are modifiers to make the match 'lazy,' but that is beyond the scope of this primer.
 - The initial '.*' will match all the path separators (/) except the last one that has to be saved to be matched by the literal '/' in the expression.

Attribute Transformations

The following examples involve requirements for attribute values that should be modified when mounted in STEP'n'design.

For more information on attribute transformations, see the **Attribute Transformations** documentation.

Transform attribute value based on the value

Consider the requirement that objects containing greater than 60% recycled material get an icon informing the customer that recycled materials were used. Using an Attribute Transformation (created in System Setup), it is possible to use a regular expression along with the transformation 'Replace substrings of the value using regular

expressions' to modify the value of the attribute 'recycled percentage' (with ID RecycledPcent) only under the specified condition. Then, when mounted in STEP'n'design, the 'recycled' icon (with ID 232810) provided by the transformed value is displayed. For this example, two (2) regular expression transformations are used.

- **Transformation 1:** Replace if matches `^[1-5][0-9]$|^[0-9]$` with
- **Transformation 2:** Replace if matches `.*` with
`step://asset?editor=Asset&contextid=Context1&id=232810&workspaceid=Main`
- **Notes**
 - The first regular expression transformation determines if the value of the 'recycled percentage' attribute is less than 60 percent by testing for any two-character value where the first character is 1-5, and the second character is 0-9, OR a single character value that is a number 0 - 9. If a match is found, any existing value is deleted. In actuality, the STEP value is not altered at all, rather the transformed value will be sent for use by the external system.
 - The second regular expression transformation replaces any actual value for the 'recycled percentage' attribute after the first transformation, with the URL to an asset which is a recycling icon. Again, the STEP value is not actually altered.

RegExp - Attribute Transformation

Attribute Transformation			
>	Name	>	Value
>	ID		RegExp
>	Name		RegExp

Transformations

X + - Replace if matches ^[1-5][0-9]\$|^*[0-9]\$ with

X + - Replace if matches .* with step://asset?editor=Asset&contextid=Context1&id=232810&workspaceid=Main

[Add Transformation](#)

Multivalue Separator

Applies To

☐ Name

☐ ID

ID	Name	Edited by	Completeness Score	WMMandate
>	RecycledPcent	Recycled Percent	2017-07-24 13:12:27 by ...	
>	Add Attribute			

Import Transformations

The following examples involve requirements for attribute values that should be modified upon import.

For more information on import transformations, see the **Modifying Data** section of the **Inbound Map Data Options** section of the **Data Exchange** documentation.

Format US Phone Numbers

Phone numbers in the United States are a total of ten (10) digits. The common formatting is three (3) digits within a set of parentheses, followed by three (3) digits, followed by a dash (-), and then a final four (4) digits. During import, the regular expression transformation can enforce the correct number of digits, as well as the formatting of this data.

- **Transformation:** Replace first match of `(\d{3})(\d{3})(\d{4})` with `($1) $2-$3`
- **Notes**
 - The regular expression is looking for a match of a total of ten (10) numbers and the transformation will be performed on the first match.
 - Groups are created using sets of 3 numbers, 3 numbers, and 4 numbers, and their original order is retained.
 - Punctuation is added to the groups, including a set of parentheses around group 1, followed by a space, and then a dash (-) between groups 2 and 3.

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:
☐ Mandatory

Value - Source:Phone

Replace first match of `(\d{3})(\d{3})(\d{4})` with `($1) $2-$3`

Add Transformation

Preview

Phone=Phone ✓
(624) 571-2437
(963) 589-0030
(355) 736-8956

Result:

ID=<ID> ✓	Name=<Name> ✓	Phone=Phone
I-Subscriber_0042	Winter Haney	6245712437
I-Subscriber_0059	Zeph Pope	9635890030
I-Subscriber_0175	Yoshio Burks	3557368956

Auto Map Map Constant Remove Transform Generate Profile

Back Next Finish Cancel

Search

The following example shows that STEP Search allows for matching on a regular expression.

For more information on using a regular expression to search STEP values, see the **Advanced Options** topic in the **Getting Started / User Guide** documentation.

Find Text

Using a regular expression via the Advanced search option in workbench, allows a text string to be located in STEP regardless of the object type in which it appears.

- **Expression:** [a-z]{6}8888
- **Values Allowed:** trader8888, Customer8888
- **Values Rejected:** Trader8888, Custom8888
- **Notes**
 - The regular expression is looking for a match of six (6) lower case letters followed by four (4) instances of the number '8'.

Search

Search

[a-z]{6}8888

Advanced

☐ Match Case on Names and Values

☐ Include Inherited Values

☐ Exclude values

☒ Regular Expression

Reset

Search

Displaying 3 of 3 results

Show Details

Name

>

(160323) ID = 160323

Customer8888 ID = D&B08888

Customer8888(2) ID = CON_225853

<

(160323) rev.0.8 - Product

Product

Product Variants

Sub Products

Refer

Description

Color Attributes

Item Brand Information

Item Description Information

Item Manufacturer Information

Mass Creation

Name	>	>	Value
Product Family Title	abc		trader8888

Override Control Rule

Sales Item Marketing Descriptions

Table Transformations

The following example involves a requirement for a table where text will be modified not only in a Product Template in STEP'n'design, but also in the Table Definition when setting up a table in STEP Workbench.

For more information on transformations in tables, see the **Applying Attribute Transformations to Tables** documentation.

Transform row / column text formatting

Once a table is defined in workbench, transformations can be added on the Preview tab. The Add Transformation link gives access to many transformations, including a formatting option that uses a regular expression to modify the text.

- **Transformation:** Replace every match of `^\d\d(.*)` with `$1`
- **Notes**
 - The regular expression is looking for a match of a string that starts with exactly two (2) digits, followed by any number of characters (or no other characters).
 - The value is updated so that the initial two characters are stripped out.

The screenshot shows the 'Audi A8 rev.0.1 - Tables' interface. The 'Tables' tab is active, and the 'NAPA Table Type 1' is selected. The 'Definition' tab is open, showing a table with columns 'Part', 'Year', and 'Model/Application'. The table contains data for 'Brake Pads' for the years 2017 and 2016-12. A 'Row/Column Text Formatting' dialog box is open, showing the transformation rule: Replace every match of `^\d\d(.*)` with `$1`. The dialog also shows options for 'Ignore Headings' and 'Add Transformation'.

Validation Base Type

The following examples involve requirements for restricting the values that are allowed for an attribute.

For more information on using a regular expression as a validation base type, see the **Validation Rules** documentation.

Format US Zip Code

Zip codes in the United States are five (5) digits, with an additional optional dash (-) and four (4) more digits. In STEP, the regular expression validation base type can enforce the correct formatting of this data, for example, 30144-5646.

- **Expression:** `[0-9]{5}(-[0-9]{4})?`
- **Values Allowed:** 30144, 30144-5646
- **Values Rejected:** 301445646, 30144-88

- **Notes**

- The regular expression is looking for a match of five (5) numeric characters, and optionally allows a dash (-) followed by exactly four (4) more numeric characters.

Format US Social Security Numbers

Social Security Numbers (SSN) in the United States are a total of nine (9) digits. The common formatting is three (3) digits, followed by a dash (-), then two (2) digits, followed by another dash (-), and then a final three (3) digits. In STEP, the regular expression validation base type can be used to enforce the correct number of digits as well as formatting of this data, for example, 123456789 or 123-45-6789.

- **Expression:** `^\d{3}((-?\d{2})?)(\d{2})\d{4}$`

- **Values Allowed:** 123456789, 123-45-6789

- **Values Rejected:** 1234567891, 12-3456-789

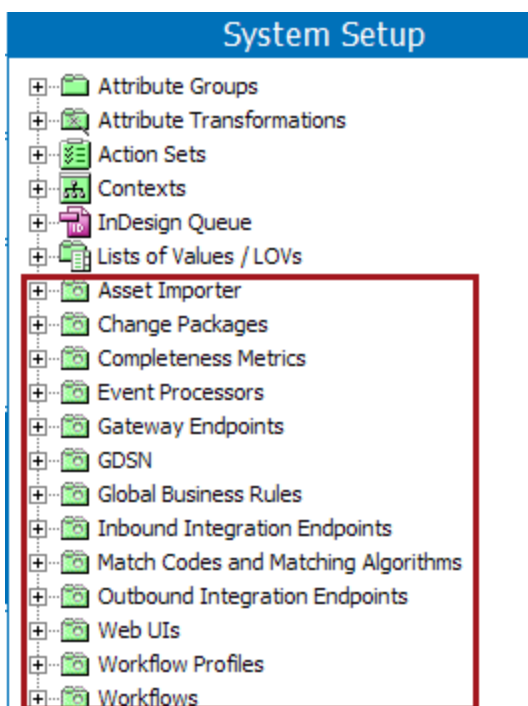
- **Notes**

- The regular expression is looking for a match of three (3) numeric characters. It optionally allows a dash (-) followed by exactly two (2) numbers and another dash (-), or two (2) numbers without the dashes. And finally, is looking for a match of four (4) more numbers, for a total of 9 numbers.

Setup Groups

Setup groups are special folders, which need to be configured to allow for holding of Integration Endpoints, business rules, Web UI configurations, workflow configurations, etc.

They are essentially the root nodes to create the system setup configurations. You can define and refine your setup groups to group the system setup configurations. These are helpful when you want all the Integration Endpoints, business rules, Web UI, workflows etc., to be grouped under respective parent groups (or folders). Additionally, STEP does come with some predefined setup groups which could be edited. These are explained in later sections. An example of some setup group folders are pictured below:



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.

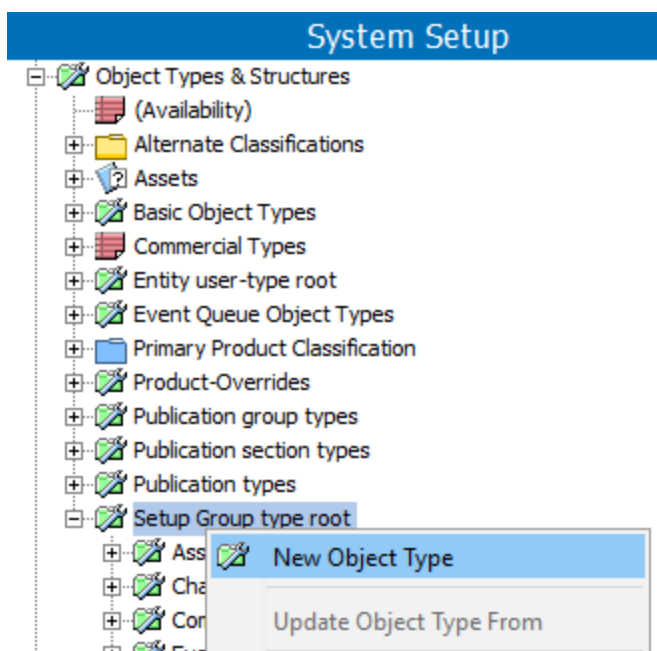
Setup Actions	Description
View Setup Groups	Users with this action applied will be able to view setup groups.
Maintain Setup Groups	Users with this action applied will be able to create and delete setup groups. They are also able to use the cut, copy and paste actions.

Creating Setup Group Object Types

Before you can create any configuration in System Setup, you need to create one or more Setup Groups and define in which Setup Group a specific object type (e.g., Integration Endpoints, Business Rules, Web UI Configurations, Status Flags, Workflow Configurations) is 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**.



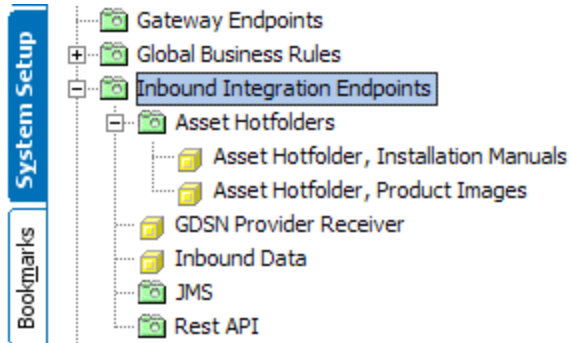
2. Key in **ID**, **Name** and Click **Create**.

The screenshot shows a 'Create Object Type' dialog box. It has a title bar with a close button. Inside, there are three input fields: 'ID' with the value 'IntegrationEndpoints', 'Name' with the value 'Integration Endpoints', and 'Dimension Dependency'. The 'Dimension Dependency' section has two checkboxes: 'Country' and 'Language', both of which are unchecked. At the bottom right, there are two buttons: 'Create' and 'Cancel'. The 'Create' button is highlighted.

Note: The IDs can be auto generated for setup group creation. See the Autogenerate Using Name Pattern and ID Pattern topic in the System Setup / Super User Guide documentation.

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 a setup group object type named Asset Hotfolder, JMS, and Rest API.



By creating different Setup Groups, it is easy to maintain all the respective integrations under respective setup groups.

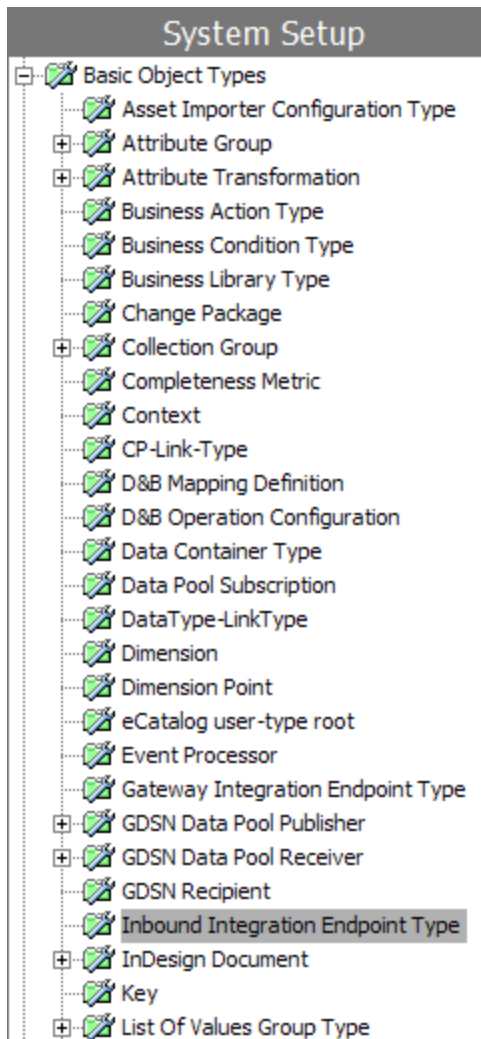
Defining Type of Objects to be Stored in Setup Groups

Before a Setup Group can hold object types, you need to select an object type to create a setup group. Object types are located under System Setup > Object Types & Structures > **Basic Object Types**. A few of the object types that could be selected to make setup groups from are listed below:

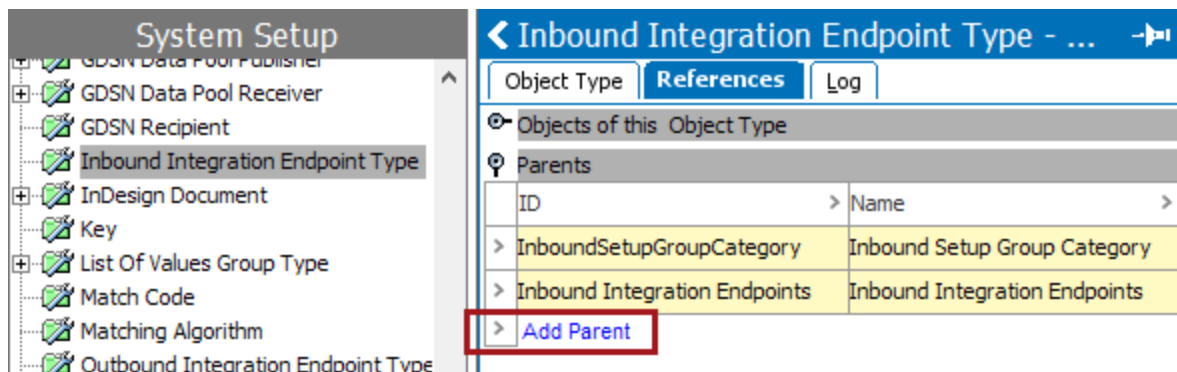
- Business Action Type
- Business Condition Type
- Business Library Type
- Inbound Integration Endpoint Type
- Outbound Integration Endpoint Type
- Web UI Configuration Type
- STEP Workflow Type

In the example below, it illustrates how you set up a setup group to hold Inbound Integration Endpoints. The same steps can be used for all of the other object types that can be created as setup groups to hold other child objects.

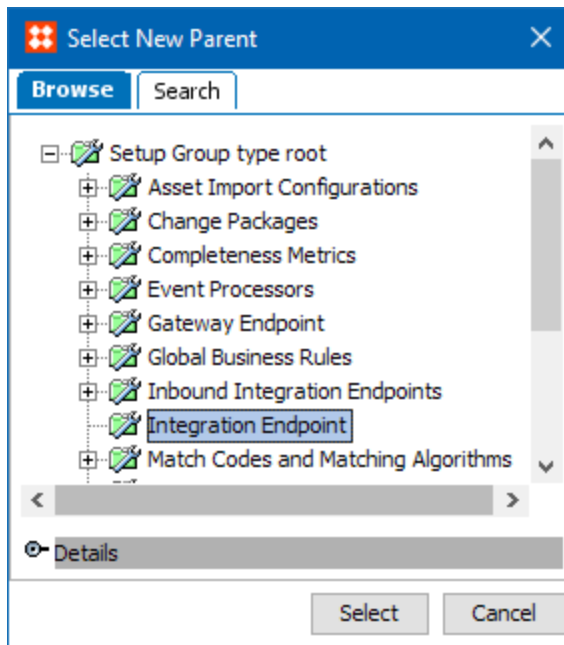
1. In **Object Types & Structures > Basic Object Types** > click **Inbound Integration Endpoints**.



2. In the **Inbound Integration Endpoint Type** editor, click the **References** tab and **Add Parent**.



3. A dialog appears where you can search or browse for 'Setup Group Types' to link the object type into. Select the relevant **Setup Group** and click **Select**.



Below is an example list of Basic Object Types that could be added as parents for any setup group depending on the requirements:

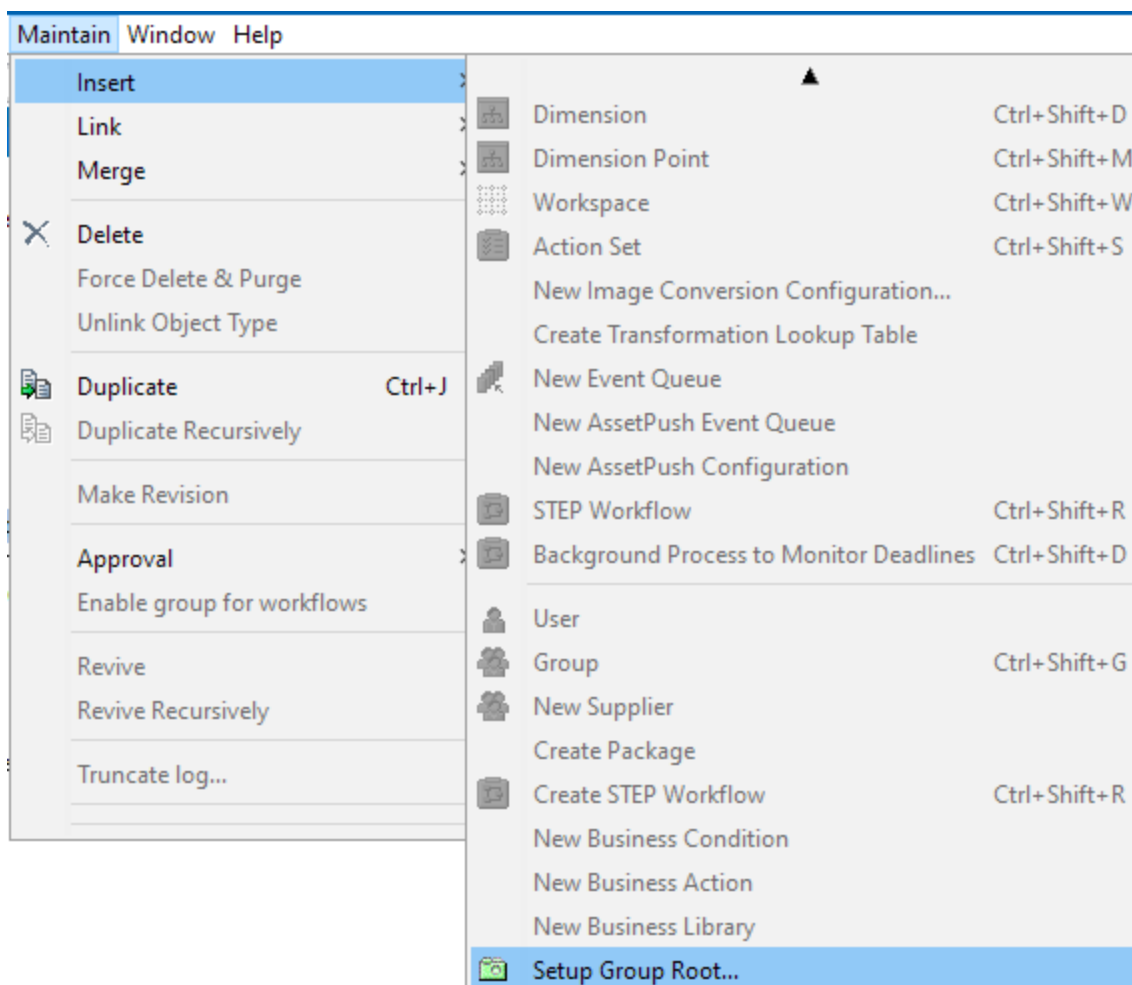
- Asset Importer Configuration Type
- Business Action Type
- Business Condition Type
- Business Library Type
- Change Package
- Completeness Metric
- Event Processor
- Gateway Integration Endpoint Type
- Inbound Integration Endpoint Type
- Match Code
- Matching Algorithm
- Outbound Integration Endpoint Type
- Web UI Configuration Type

Creating Setup Groups in System Setup

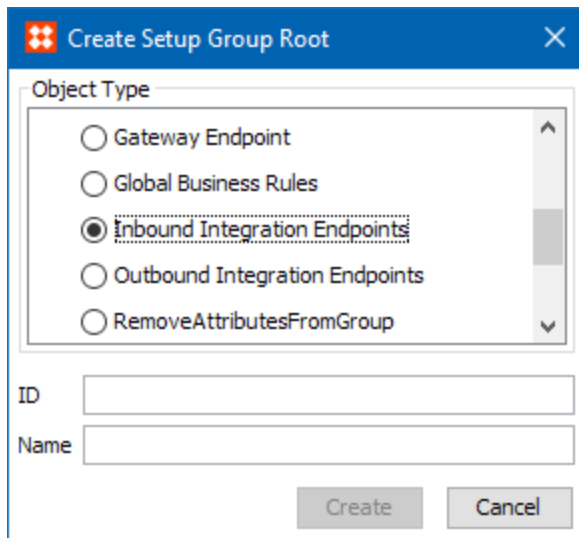
Setup Groups can be used to hold Integration Endpoints, Business Rules and/or Web UI configurations. Before you can create any of those objects, you need to create a Setup Group in System Setup.

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 the **System Setup** tab, and select any object in the System Setup hierarchy.
2. In main menu, click **Maintain**, and **Insert -> Setup Group Root**.

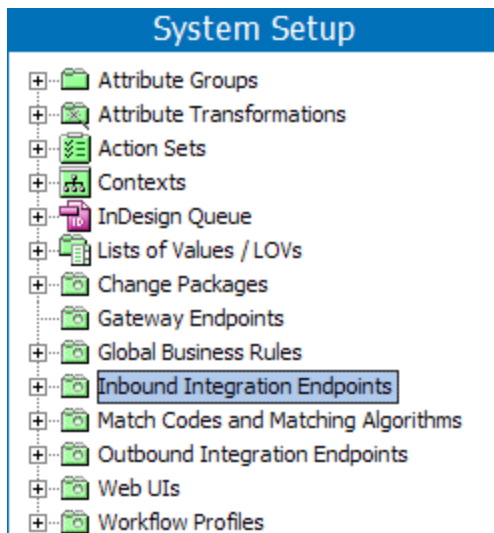


A **Create Setup Group Root** dialog appears.



3. Key in **ID**, **Name** and click **Create**.

A Setup Group is created and will appear as a node in the System Setup hierarchy. The Setup Group is ready to be used for Integration Endpoints.



Note: An auto ID pattern can be specified for the setup group. See the Autogenerate Using Name Pattern and ID Pattern topic in the System Setup / Super User Guide documentation.

Once the setup group root is created, right click and create the respective configuration under it.

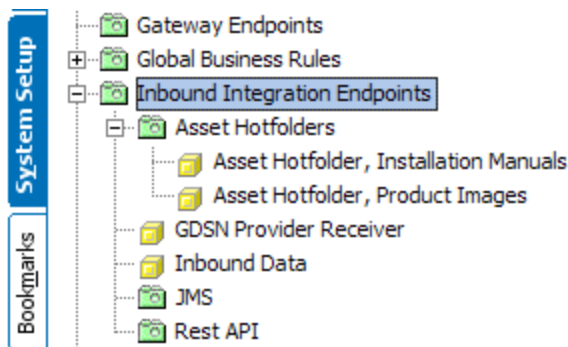
Setting Up Privileges Example

In this example, two user groups will be created.

- One user group created will be named **View Setup Group and Integration Endpoints**. Users linked into this group will be granted the privilege to view Setup Groups and Integration Endpoints.
- The other user group created will be named **Maintain Setup Group and Integration Endpoints**. 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.



To be able to create additional setup groups that are able to hold Integration Endpoints below the **Integration Endpoint** setup group folder, the **Setup Group type root** 'Integration Endpoints' must have itself added as a parent. This same concept applies to all setup groups. Please see the picture below.

System Setup

+

 D&B Type

+

 D&B Type

+

 Event Processors

+

 Gateway Endpoint

+

 Global Business Rules

+

 Inbound Integration Endpoints

+

 Match Codes and Matching Algori

< Inbound Integration Endpoints - References

Object TypeReferencesLog

Objects of this Object Type

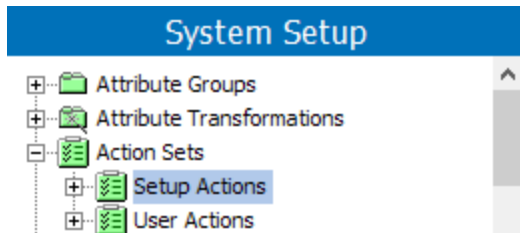
Parents

ID	Name
Setup Group user-type root	Setup Group type root
Inbound Integration Endpoints	Inbound Integration Endpoints

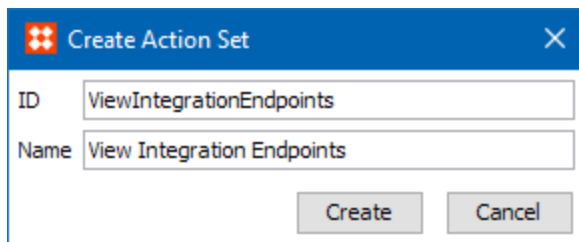
For more on how to maintain object types, see the **Maintaining Object Types** documentation in the **System Setup / Super User** documentation.

Privilege Setup Steps

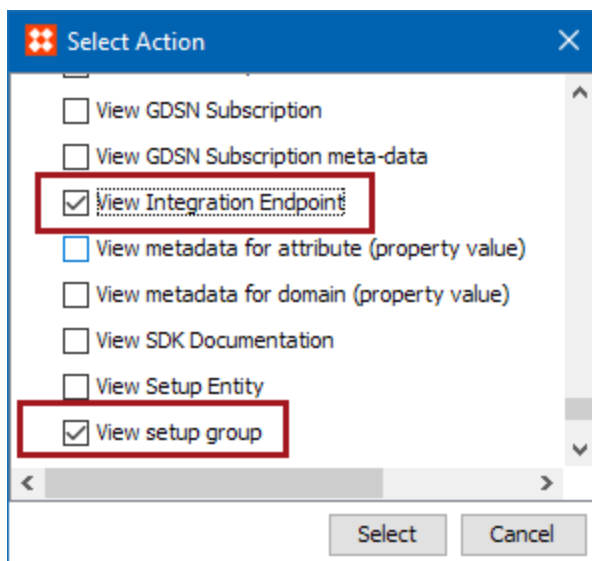
1. Go to System Setup > Action Sets > **Setup Actions**.

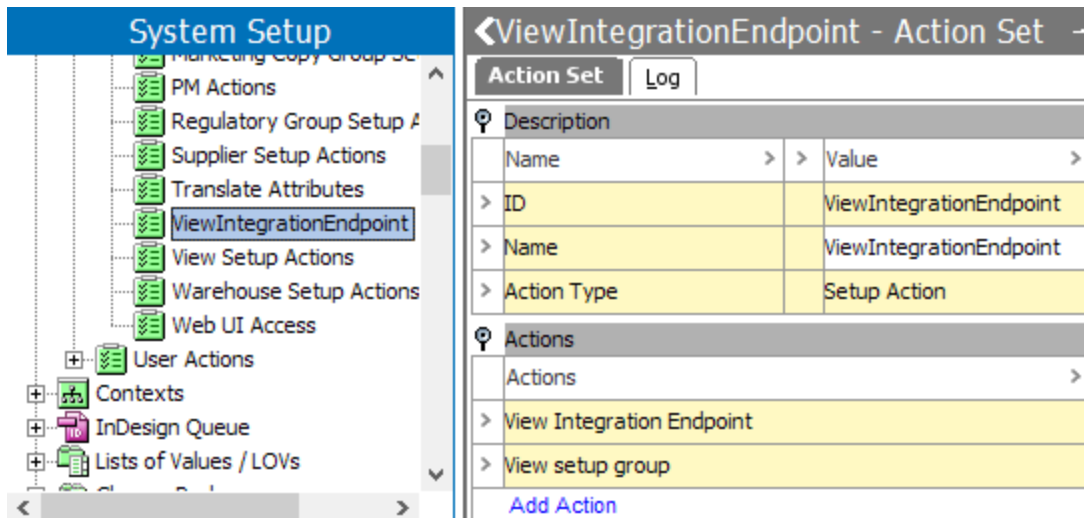


2. Right-Click and select **New Action Set**. Key in ID, Name, and click Create.

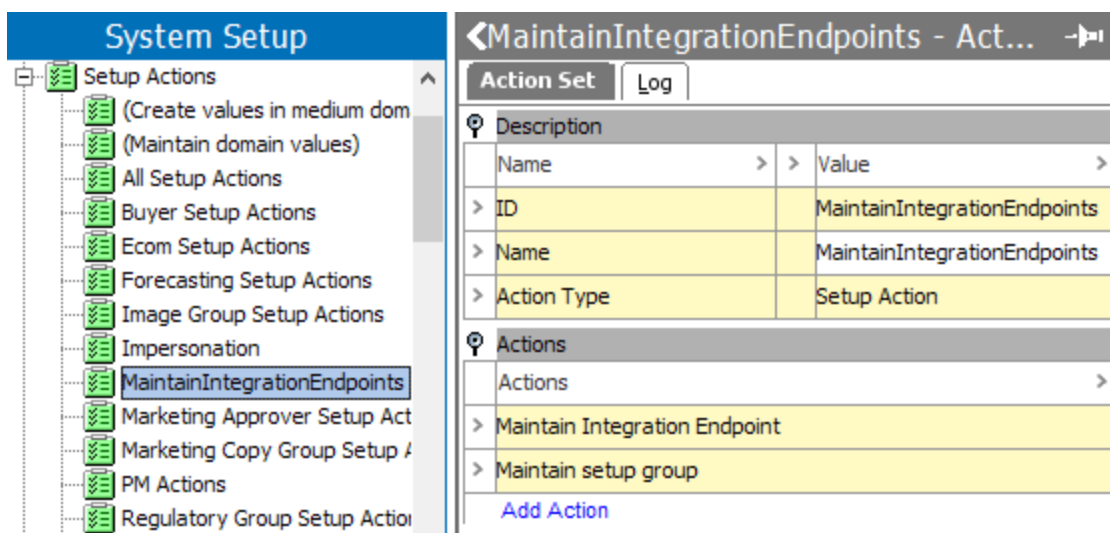


3. Select the newly created action and in the Action Set tab, click **Add Action**.
4. In the Select Action dialog, check **View Integration Endpoint** and **View Setup Group** and click **Select**.

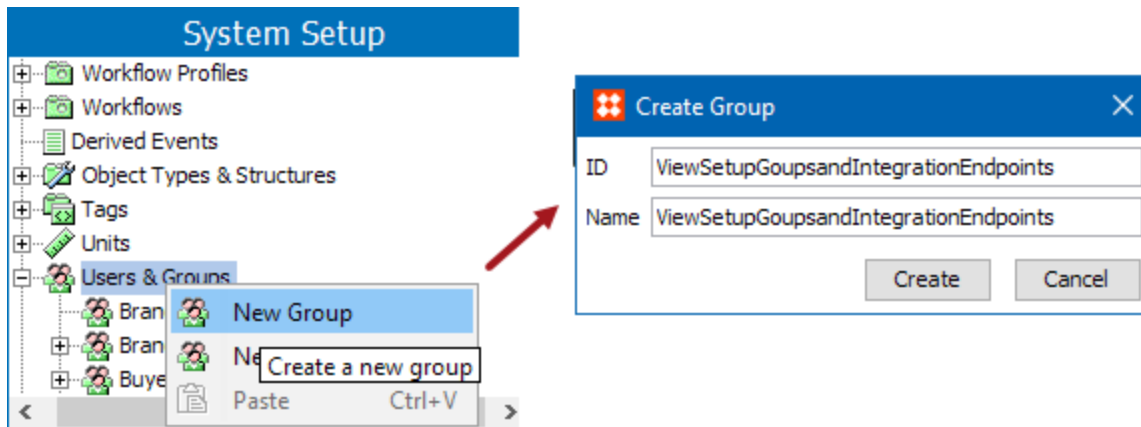




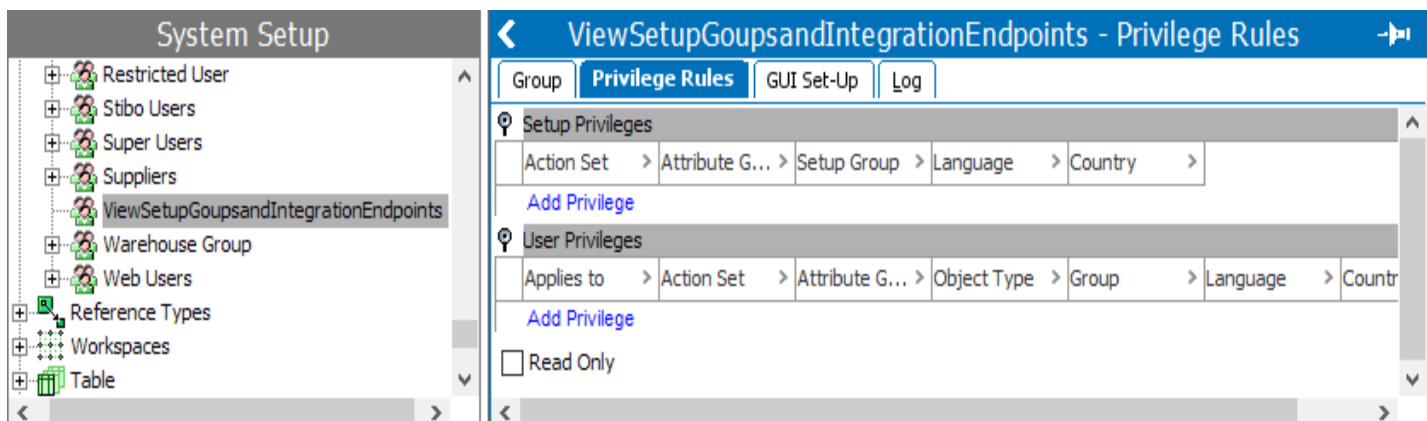
- Repeat step 1-5 and create another Action Set containing the actions **maintain setup group** and **maintain integration endpoint**.



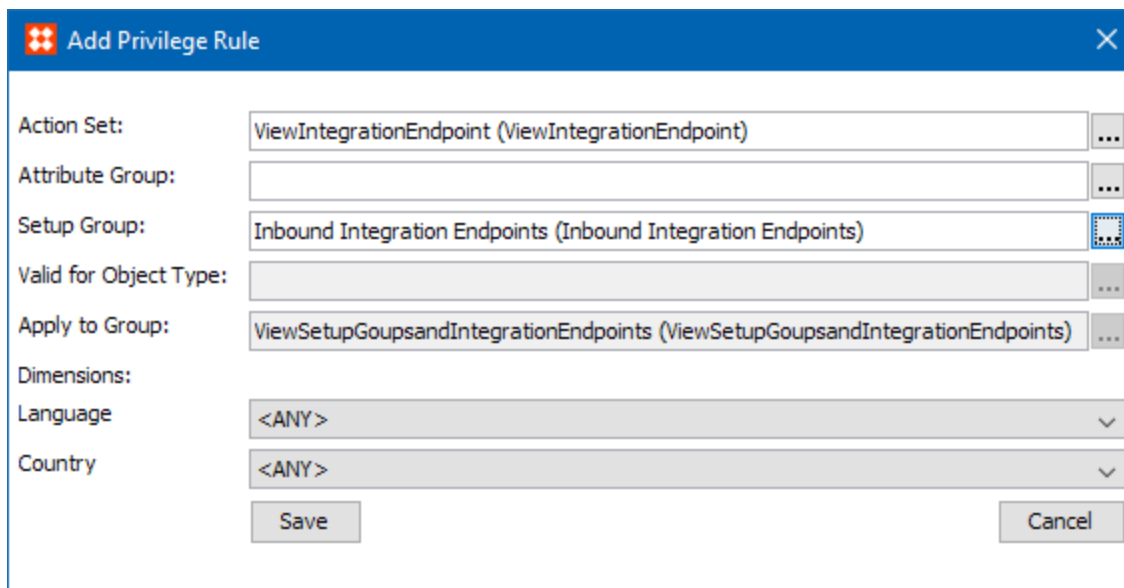
- Go to the **Users & Groups** node, right-click and select **New Group**. Key in ID, Name, and click Create.



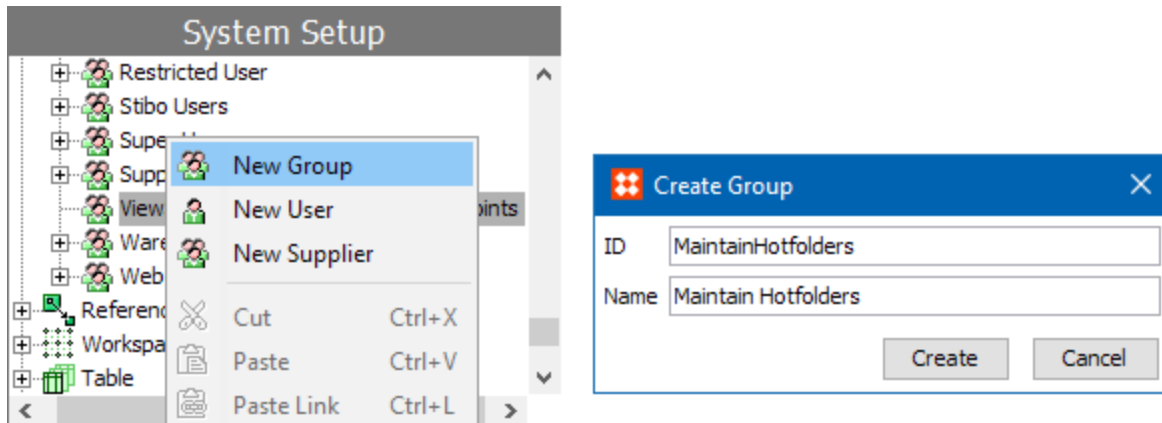
7. In **User Group editor**, click **Privilege Rules** tab.



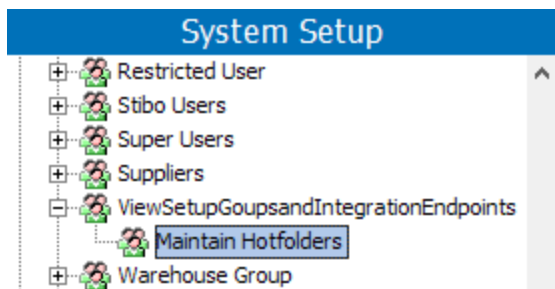
8. Under the **Setup Privileges** flipper, click **Add Privilege**. The Edit Privilege Rule dialog will appear:



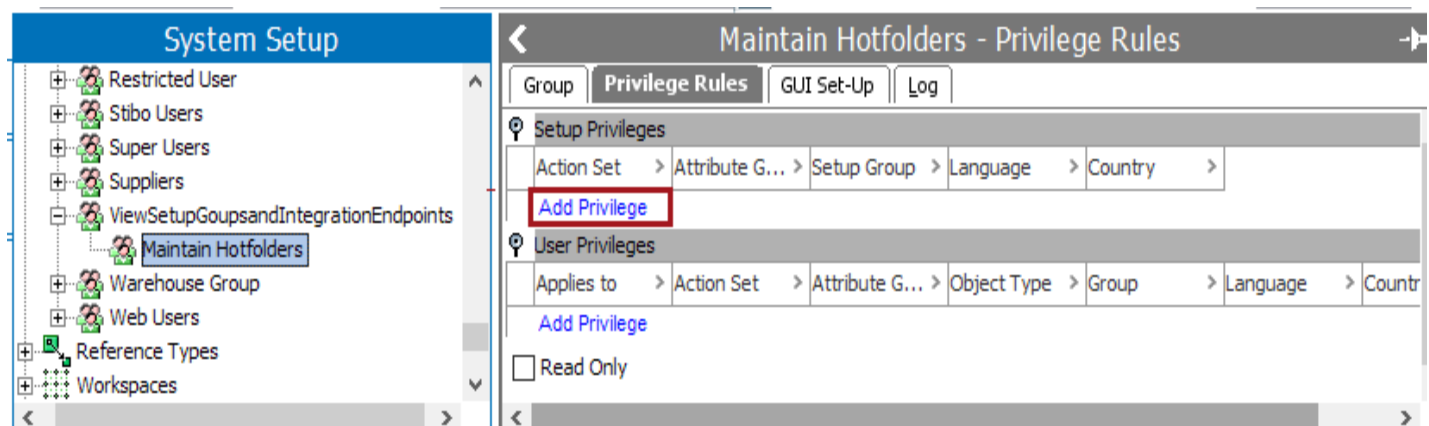
9. In the **Action Set** field, apply the **View Integration Endpoint** Action Set.
10. In the **Setup Group** field, apply the Setup Group named **Inbound Integration Endpoints**. Click Save when finished.
11. Next, in Users & Groups, select and right-click the group just created: **View Setup Groups and Integration Endpoints**. Select **New Group**. Key in the ID, Name, and click Create.



The new user group will appear below the user group **View Setup Groups and Integration Endpoints**.



12. Click the Privilege Rules tab. Under the **Setup Privileges** flipper, click **Add Privilege**.



13. In the **Action Set** field, apply the Action Set **Maintain Integration Endpoints**.

Add Privilege Rule

Action Set: MaintainIntegrationEndpoints (MaintainIntegrationEndpoints) ...

Attribute Group: ...

Setup Group: Asset Hotfolders (AssetHotfolders) ...

Valid for Object Type: ...

Apply to Group: Maintain Hotfolders (MaintainHotfolders) ...

Dimensions:

Language: <ANY> v

Country: <ANY> v

Save Cancel

14. 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'.

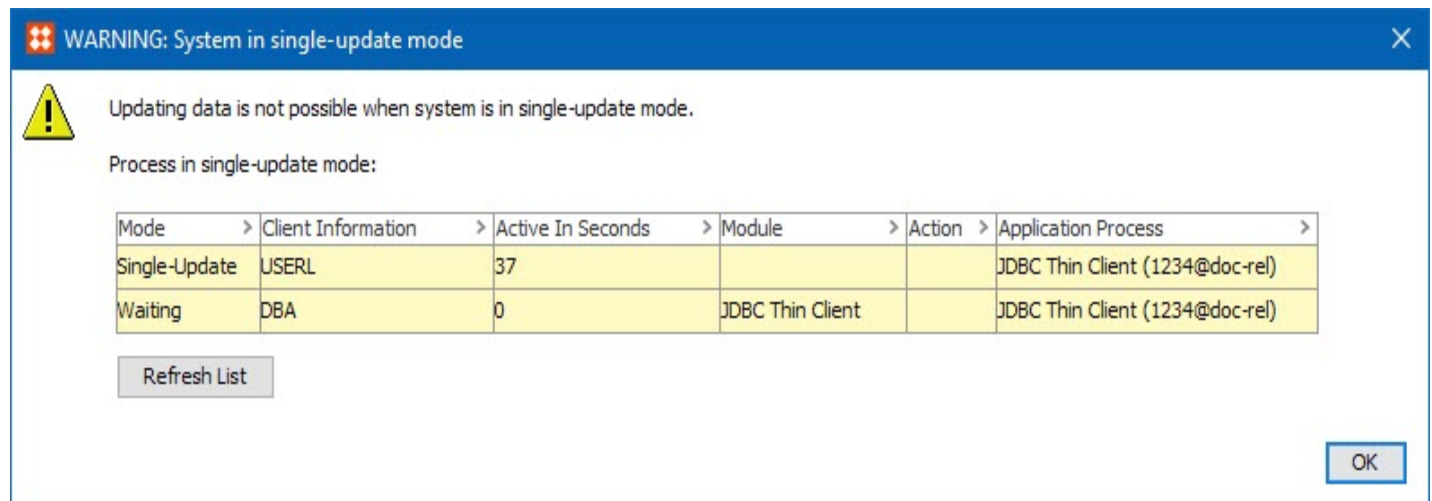
Note: The above is an example to set the User Privileges and necessarily doesn't mean that the only actions mentioned in the example are sufficient for a user to work on STEP. The user should also have other actions defined to work.

Single-Update Mode

Certain STEP operations (i.e., GUI updates, data import) can corrupt data if they are run at the same time someone else is updating the database

STEP enters single-update mode when operations that can affect data elsewhere in the system are performed. When STEP 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 single-update mode. If it is not possible for the system to enter single-update mode, or if a user tries to make changes or update when another connection is in single-update mode, a warning message appears.



WARNING: System in single-update mode

Updating data is not possible when system is in single-update mode.

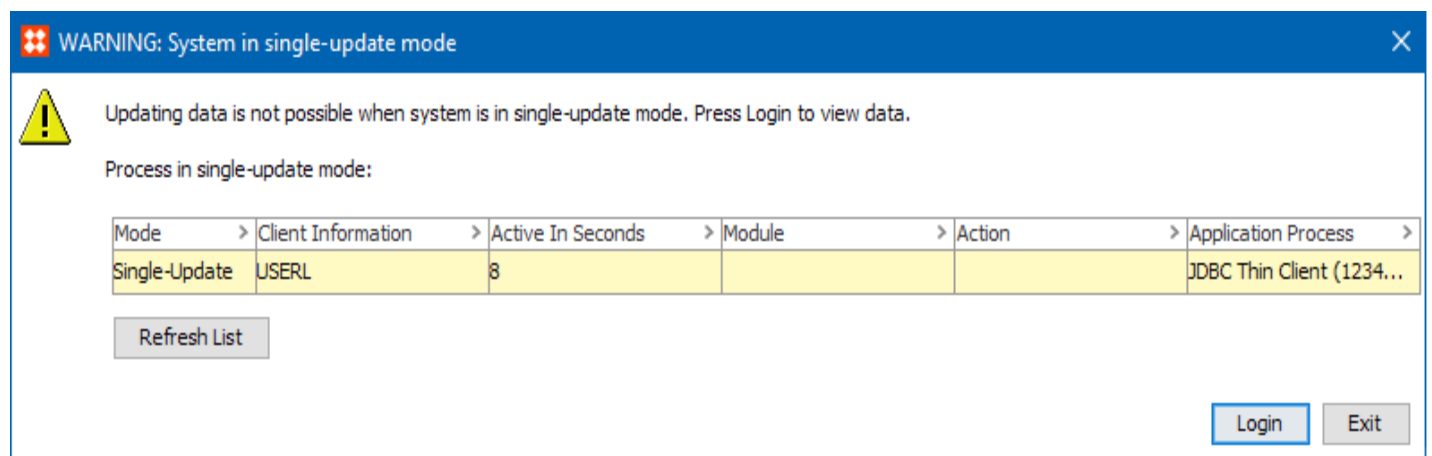
Process in single-update mode:

Mode	Client Information	Active In Seconds	Module	Action	Application Process
Single-Update	USERL	37			JDBC Thin Client (1234@doc-rel)
Waiting	DBA	0	JDBC Thin Client		JDBC Thin Client (1234@doc-rel)

Refresh List

OK

Additionally, if you log on to the system while it is in single-update mode, a warning appears.



WARNING: System in single-update mode

Updating data is not possible when system is in single-update mode. Press Login to view data.

Process in single-update mode:

Mode	Client Information	Active In Seconds	Module	Action	Application Process
Single-Update	USERL	8			JDBC Thin Client (1234...)

Refresh List

Login Exit

The warning displays information about which connection is currently running in single-update mode. While the user not in single-update mode can't change any data, they can still view data.

Note: Importer will only allow entering single-update mode if attribute 'SingleUpdateMode' on root tag 'STEP-ProductInformation' is 'Y'. Otherwise single-update mode operations would be ignored during import.

Operations that Cause Single-Update Mode

The following operations will cause the system to enter single-update mode.

- Deleting a workspace. For more information, see the **Maintaining Workspaces** topic in the **System Setup / Super User** documentation.
- 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
- Changing the validity of a Data Container Type for an Entity Type.
- Changing Data Container Type parameter **Allow multiple values** to 'No'.

To avoid entering single-update mode when changing attribute definitions, the following procedure can be used:

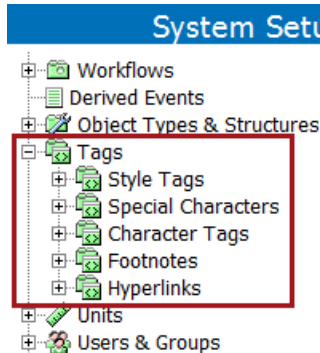
1. Create a new attribute (Attribute B) with the same desired attribute definition (including the changes from the origin attribute, Attribute A).
2. Use Bulk Update in the STEP Workbench to copy all attribute values from Attribute A to Attribute B. For more information on using bulk update functionality, 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.

If a user attempts the above actions without entering into single-update mode then errors in data could occur.

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.

Special characters are characters that are pasted into a text element or a data cell, but where the character as it displays on the screen is not available in a normal font on the print output device.

Examples are the Greek characters α β γ. These characters are valid Unicode characters and can be entered into any text element or into product data. To make these characters output to a printer or to output properly in, for example, InDesign, a formatting has to be set up.

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 two 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 to 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 Component** topic in the **Web UI Getting Started / Web UI Setup and User Guide** documentation.

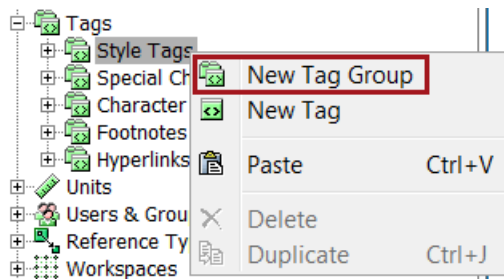
Creating Tag Groups and Tags

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

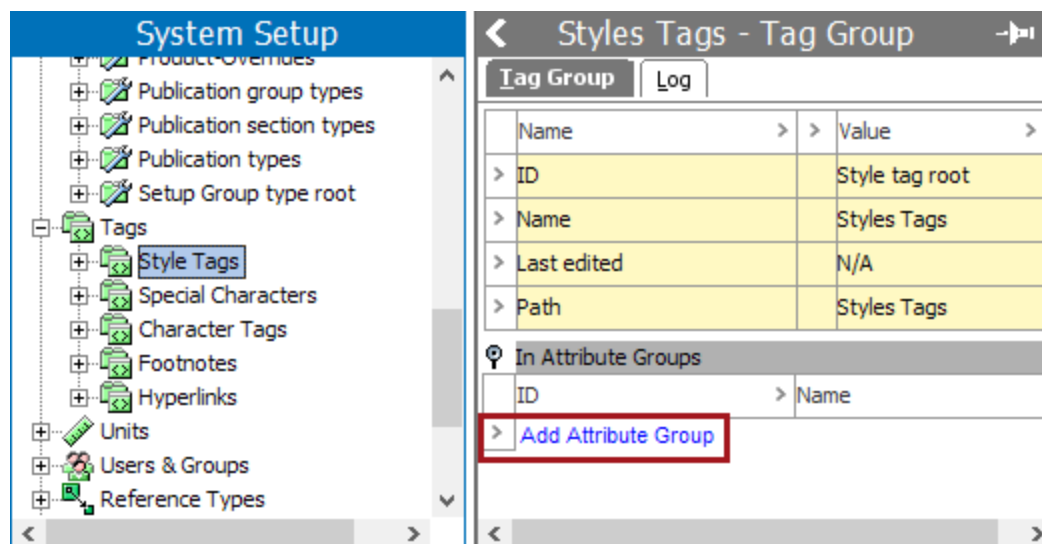
To create a tag group under one of the five types of tags, follow the steps below:

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.



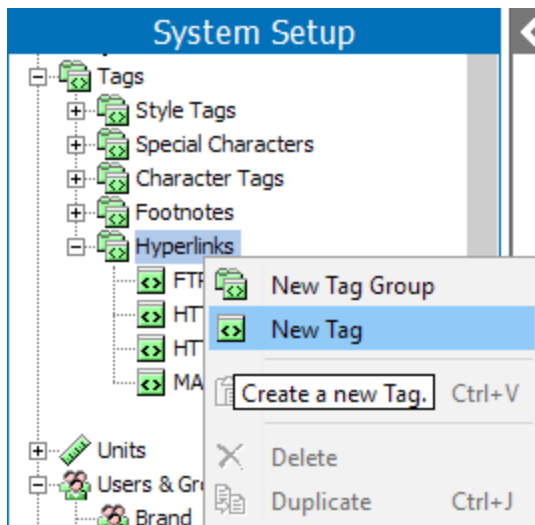
The specific privileges used to have tighter control over tag usage are 'Use tag' and 'Maintain tags'. See the **Users and Groups / Super User Guide** documentation for more information on how to set up privilege groups.

Tags

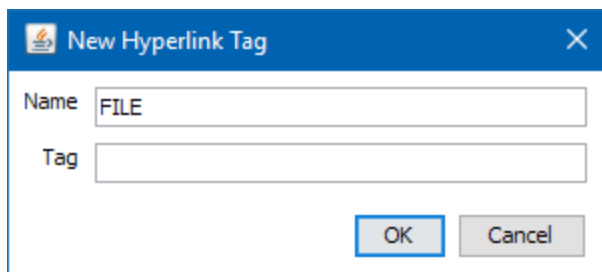
The way to create the various tags are overall the same, though for special characters, character tags, style tags, and hyperlinks, there are some special consideration to keep in mind when creating them. Both how to create tags and any special considerations are described below.

Creating Tags

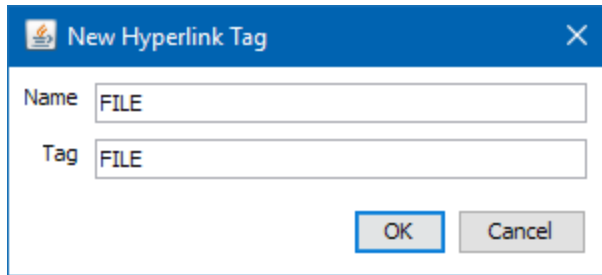
1. Go to the **System Setup** tab, expand **Tags**.
2. Right-click desired tag type, then select **New Tag**.



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.

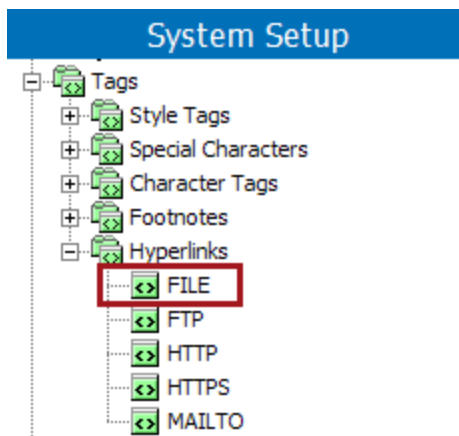


4. In the **Tag** field, enter a tag. The tag will be used for example, in a document when importing files using STEP tags. Because it can get confusing if the name of the hyperlink, which is what is displayed in the UI, is different than the tag, it is recommended to keep the name of the hyperlink and 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.

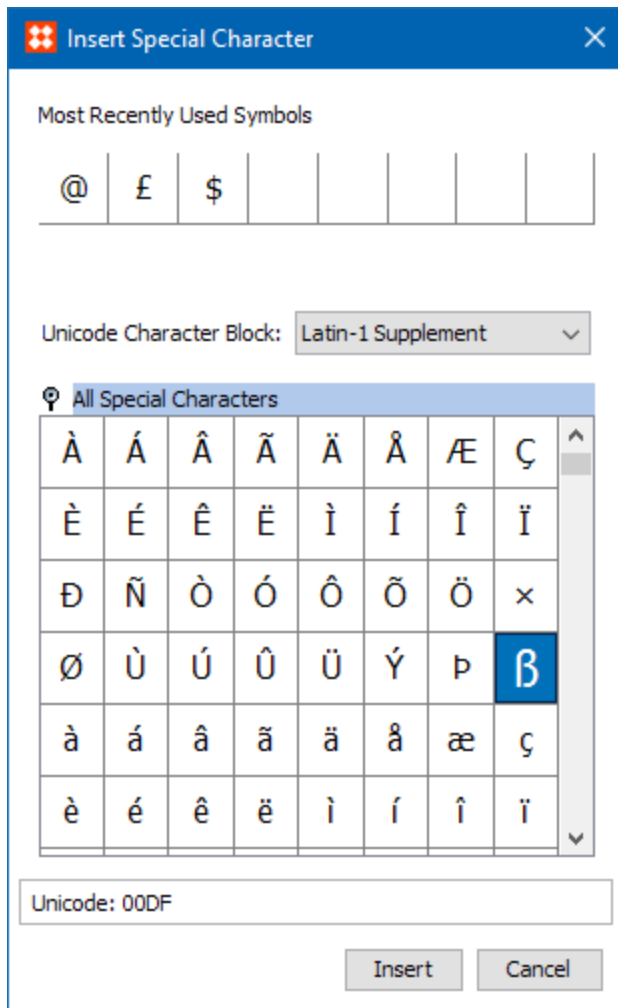


Once you have created tags, proper configuration and rendering of the tags needs to take place. See **Configuring / Rendering Tags** in this documentation.

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.



Note: There is a small subset of characters that may be used in object IDs. In the ASCII character set from characters Decimal 32 (the space character) to Decimal 126 (The tilde character), or for hexadecimal characters, it is from character 20 to 7E.

Character Tag:

1. 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

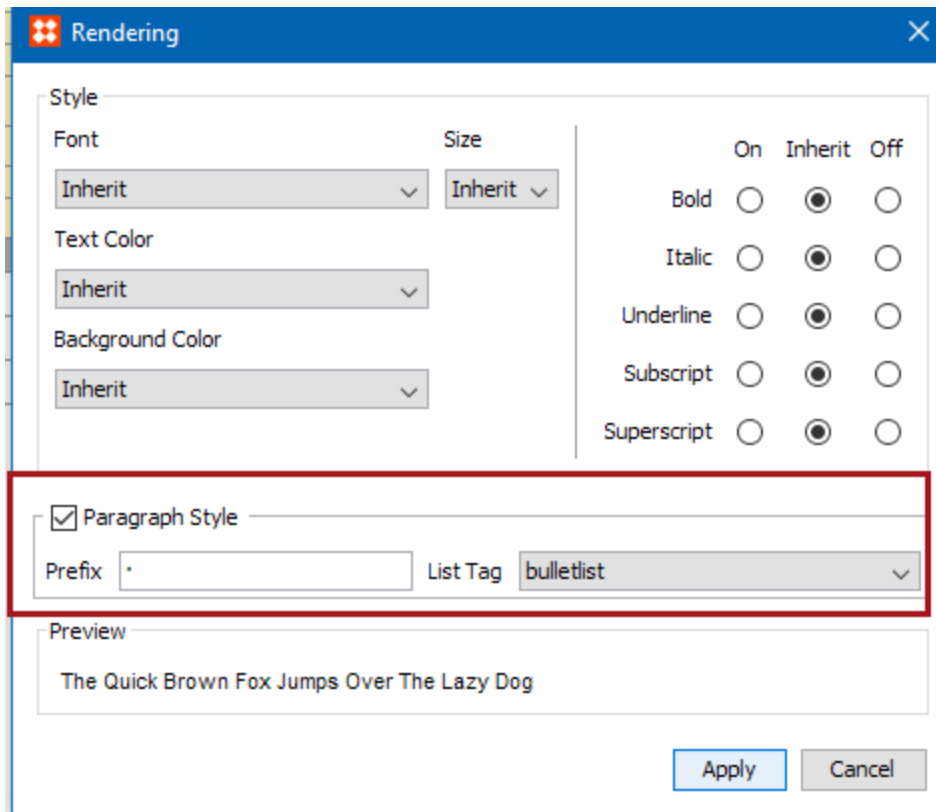
Note: Although the < and > characters are allowed in an object ID, it is not considered 'Best Practice.'

If these two character tags are entered within an object's name, the STEP will still accept them and save them as the STEP character tags <lt/> and <gt/>. However, this may cause some confusion when using the GoTo or Search functions in STEP. Additionally, if data is imported via an XLS file, and these characters are used in the object names, the system will throw an error, though it will still load the data and will transform the < and > characters to an <lt/> or <gt/>. For these reasons it is not considered 'Best Practice' to enter these two characters into an object name.

2. 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:

1. Style tags can not be used within an object's name.
2. If you have defined a style tag that will be used to define bullet lists for HTML output, enable the 'Paragraph Style' checkbox. 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.



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.

Rendering

Style

Font

Inherit

Size

Inherit

Text Color

Red

Background Color

Inherit

On

Inherit

Off

Bold

☐

☒

☐

Italic

☐

☒

☐

Underline

☒

☐

☐

Subscript

☐

☒

☐

Superscript

☐

☒

☐

Preview

The Quick Brown Fox Jumps Over The Lazy Dog

Apply

Cancel

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 Style

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 selecting 'No' will make the tag unavailable for selection on all dropdown lists available in STEP.

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.

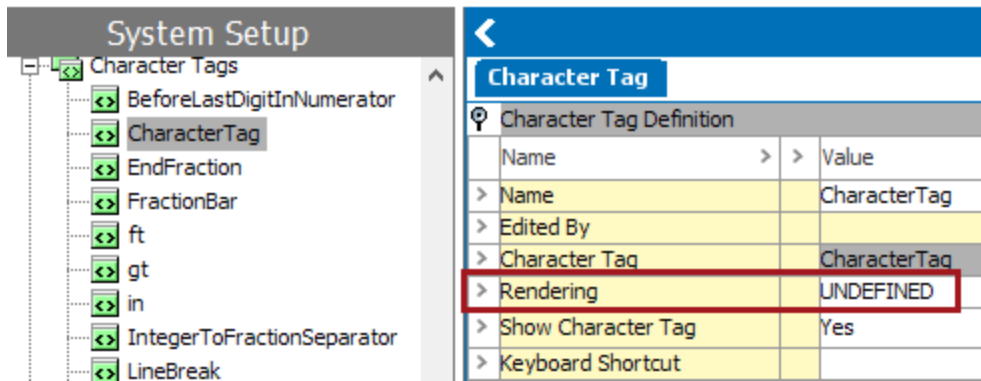
For style tags, one more field will appear in the Special Characters Definition flipper: List Tag.

Style Tag		
Special Character Definition		
Name	>	Value
Name		bulletlist
Edited By		
Tag		bulletlist
Rendering		The Quick Brown Fox Jumps Over The Lazy Dog
Show Style		No
Keyboard Shortcut		
List Tag		Yes

In the List Tag field, select 'Yes' if the style tag should be used to define bullet lists for HTML output. (E.g., if the style will be used with the HTML tags and .)

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'.

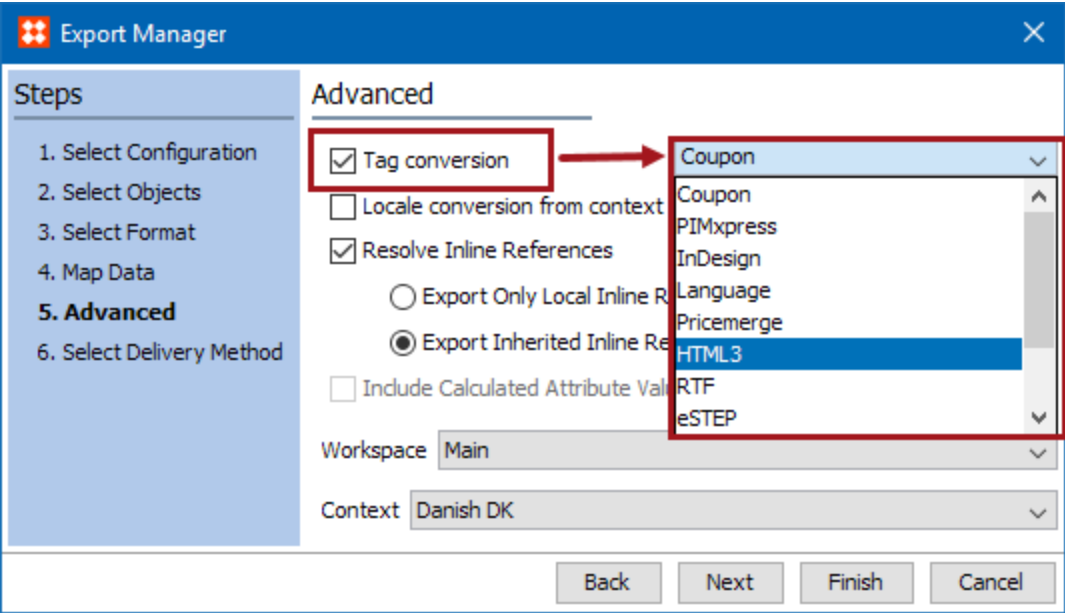


This means that each time the tag is used in Rich Text Editors, then the rendering [Undefined] is shown.

- 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.

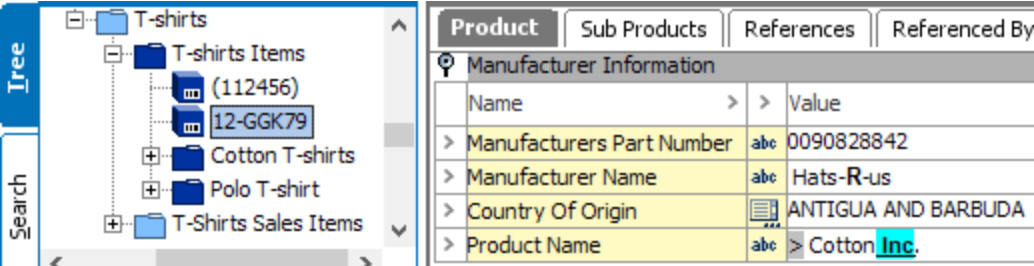
Configuring Output Formatting

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.



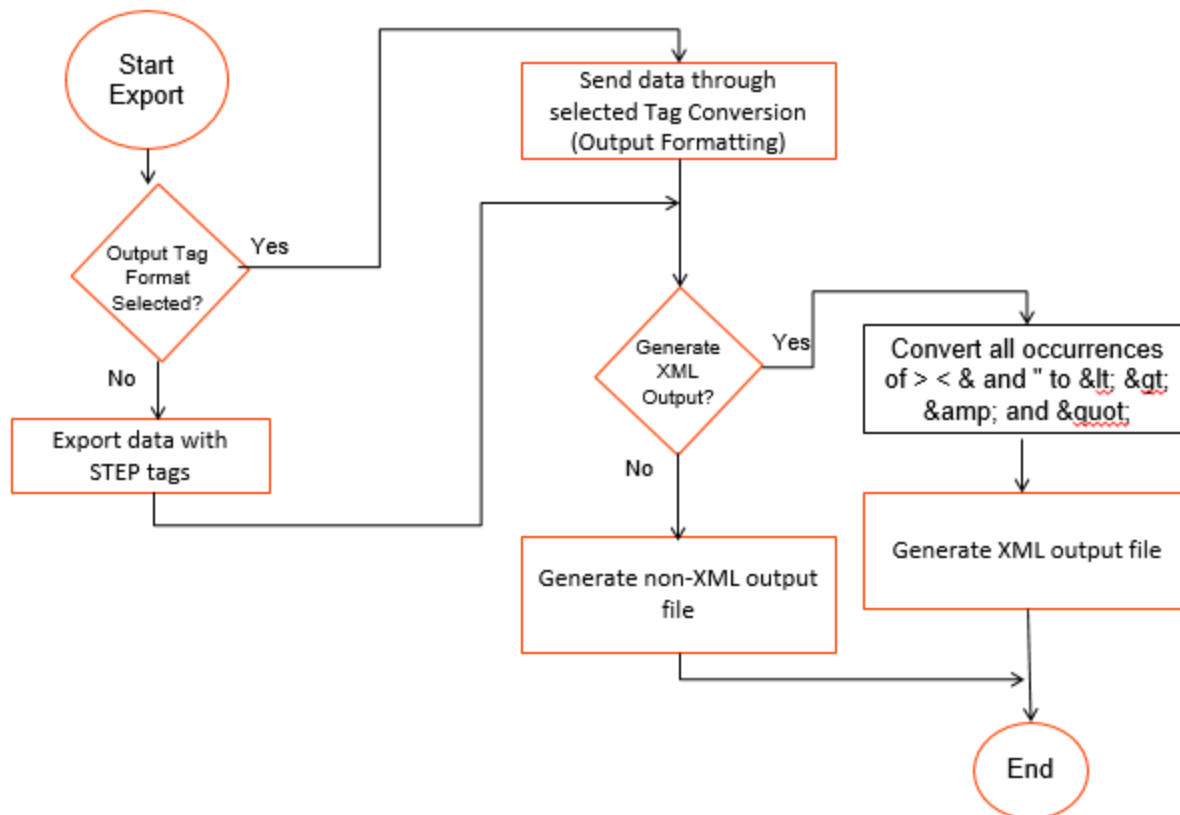
For more on exporting, see the **Export Manager** and the **Outbound Integration Endpoints** topics in the **Data Exchange** documentation.

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 four 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:

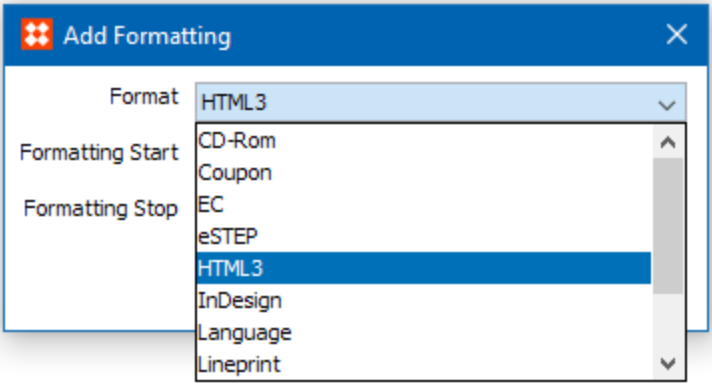
Name	Symbol	Predefined Entities / Character Entity Reference	Unicode number
double quotation mark	"	"	Unicode 0022
ampersand	&	&	Unicode 0026
less-than sign	<	<	Unicode 003C
greater-than sign	>	>	Unicode 003E

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.

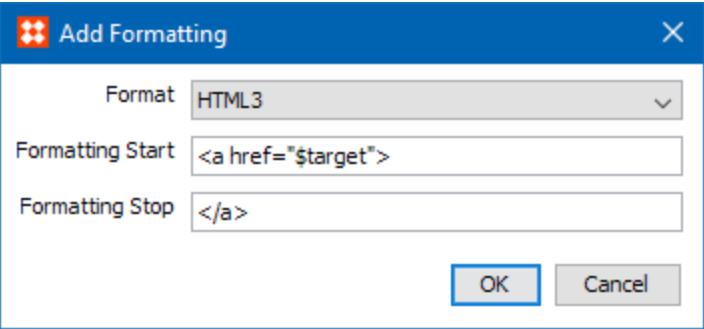
Output Formatting Flipper

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.



2. Enter the start tag for the selected output format name in the **Formatting Start** field and the end tag in the **Formatting Stop** field if adding formatting for Style Tags, Footnotes, and Hyperlinks. For Special Characters and Character Tags, only Formatting is needed to be entered. In this example, because it is common to have a website link as a hyperlink, and users want the hyperlink to the website to work when the attribute is exported, the following start and stop tags are used. Press **OK**.



3. The format and its start and stop tags, or formatting if a Special Character or Character Tag were selected, will appear in the formatting table.

FILE - Hyperlink Tag		
Hyperlink Tag		
Special Character Definition		
Name	>	Value
Name		FILE
Edited By		
Tag		FILE
Rendering		The Quick Brown Fox Jumps Over The Lazy Dog
Show Style		Yes
Keyboard Shortcut		
Output Formatting		
Format	>	Formatting Start
Format	>	Formatting Stop
> HTML3		
> HTML3		
> Add Formatting		

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)**.

The screenshot shows the 'System Setup' window with a left-hand navigation pane and a main 'System Settings' table. The 'Users & Groups' node is selected in the navigation pane. The 'System Settings' table has two columns: 'Name' and 'Value'. The setting 'Pass through unconverted Special Characters and Tags (Y) or discard them on output (N)' is highlighted with a red box, and its value is 'N'.

Name	Value
> Enforce Mandatory Check for Attributes, References and Links	Y
> Product Editor, Group attributes by top group	Y
> 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	Migrate Legacy Completeness Functionality
> Conditional Validity Attribute	ConditionAttribute
> Block Attribute Groups with more than 1000 attributes	Y
> Use full pathname for classes on Product References Tab	Y
> Pass through unconverted Special Characters and Tags (Y) or discard them on output (N)	N
> Product Attribute Help metadata attribute	(AttributeHelpText)

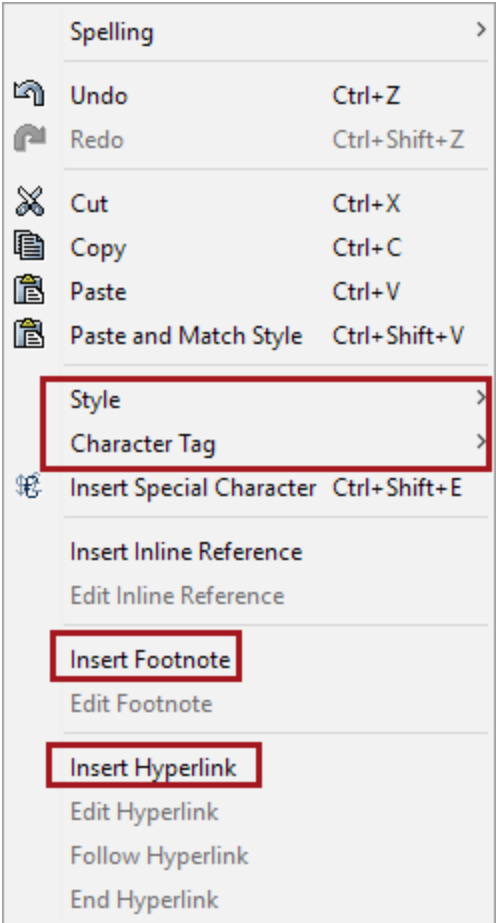
If it is set to N, then any style tags, character tags or special characters that do not have an entry in the output formatting area for the format selected will all be stripped out on output.

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.



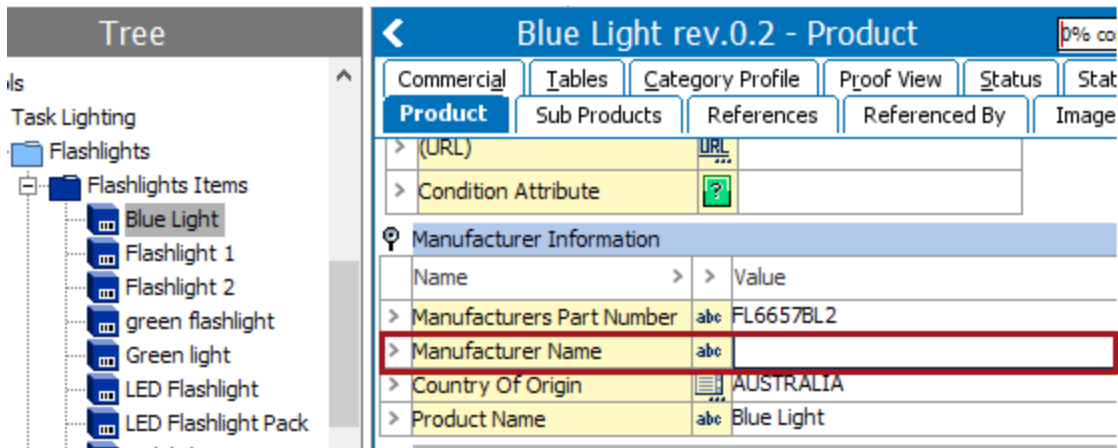
Note: Footnotes are used to add an additional piece of information to attribute values when the attributes are used in STEP Tables only. If footnotes are added to attribute values outside of a STEP Table, they will not output expected result.

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% co

Commercial Tables Category Profile Proof View Status Stat

Product Sub Products References Referenced By Image

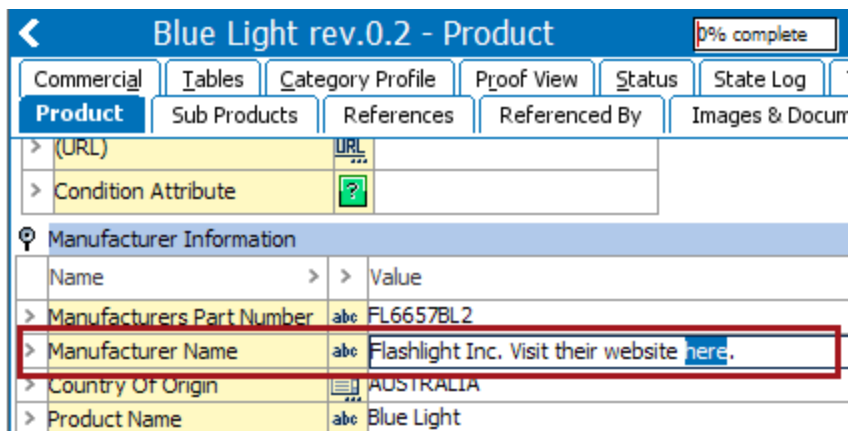
> (URL) URL

> Condition Attribute ?

Manufacturer Information

Name	>	>	Value
> Manufacturers Part Number	abc		FL6657BL2
> Manufacturer Name	abc		
> Country Of Origin			AUSTRALIA
> Product Name	abc		Blue Light

- 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 1

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			AUSTRALIA
> Product Name	abc		Blue Light

- 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

When a STEP ID is not available (such as with imported data) or is not an ideal unique identifier (i.e., when external systems need to update data in STEP), STEP Keys can be implemented. A key is a unique database entity composed of (transformed) object data, and represents a specific object in the STEP system. Because no two objects can have the same key, they provide an alternative method for identifying objects. If STEP ID isn't available (such as with imported data), a key can be used in its place.

The idea behind the STEP Key concept is simply that an external system can identify an object in STEP, not via its ID or URL but on one or more Attribute values. Thus, a Key in STEP is basically a string representation of an object that is unique for a specific key definition.

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 information on exporting STEPXML, see the **STEPXML Format** section of the **Data Exchange** documentation.

Defining Keys

Each key has a formula that defines what object data is used and how it is combined and transformed.

A key definition has an ID, a state, and specifies the following:

- One or more Object Types that the key applies to (Product, Entity, Classification and Asset Object Types only).
- One or more attributes whose values are used in constructing keys for instances of the selected Object Type (s).
- A STEP Function used for generating / calculating the actual keys.

Key Attribute Considerations

As there needs to be no more than one calculated key (string) per object covered by the key definition, a number of restrictions apply to the attributes used for calculating the keys.

Because a key must be the same in all contexts and workspaces, the selected attributes must comply with the following:

- must be valid for the selected object types.
- must be externally maintained (values should be the same across all workspaces).
- must be single valued, and when specification attributes are used, the values must be local.
- may not be dimension dependent (values should be the same across all contexts).
- may not be calculated, as the update of a single value in STEP could lead to the system potentially having to re-calculate millions of keys.

- length must not exceed 1,000 characters.
- if a validation base type list of values (LOV) is used, then 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 cannot 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.

Key Example

See below for examples of key formulas and their resulting values.

In this example, two products object types 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 concatenated 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 a 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.
- For example, if you have data for books and you determine that the books could be uniquely identified with one attribute, ISBN, then you could have a calculation for this particular attribute value.

Creating and Deleting Keys

When creating a key, the user defines what object data is used and how it is combined and transformed. Multiple keys can be created and inactive or active in the system. More than one key can be valid for the same object type at the same time.

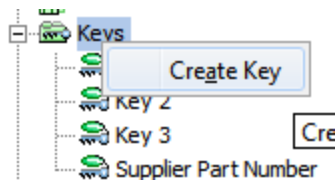
Prerequisites

It is expected that anyone creating a STEP Key is familiar with the **Defining Keys** section of the **Keys** topic within this guide.

To Create a Key

Below are steps to create a key.

1. Go to **System Setup** > Right-click **Keys**, and then click **Create Key**.



2. In the Create Key dialog enter an **ID** and a **Name** for the key, and then click the **OK** button.

3. Navigate to the **Key tab** of the newly created key.

Key 4 - Key

Key Log

Definition

Name	Value
ID	Key 4
Name	Key 4
State	Inactive
Last edited by	2015-12-02 04:56:04 by USER

Used For Object Types

ID	Name
Add Object Type	

Key Attributes

ID	Name
Add Attribute	

Key Formula

- Expand the **Used For Object Types** flipper, click the **Add Object Type** link. Browse or search for the object types to be valid for the key.

Note: Key functionality does not work with product overrides. If a product override object type is selected, then an Error dialog will display.

- Expand the **Key Attributes** flipper > Click the **Add Attribute** link > Browse or search for the attributes to be used in the key formula.

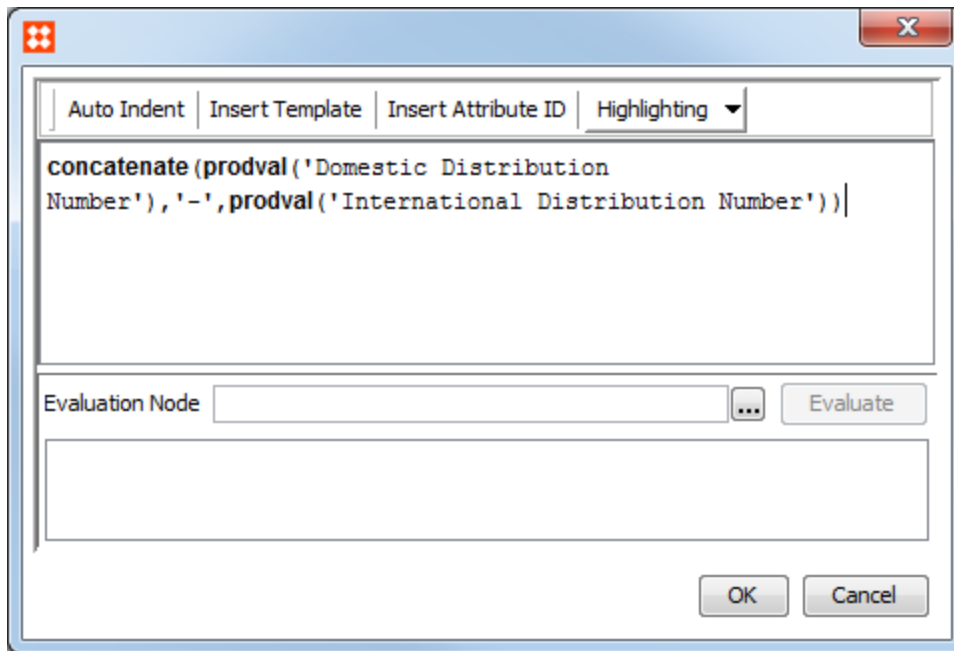
Note: A number of restrictions apply to the attributes used for calculating the keys. For more information, see the **Defining Keys** section of the **Keys** topic within this guide.

- In the 'Key Formula' field, click the ellipsis button (...) and the Function editor dialog will display. Define the key formula so that the object data will be properly 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

Important: When writing the STEP Function, only the value extraction and string manipulation functions will be available. Thus, you can combine and manipulate the key attribute values but not e.g. let the key include the ID of a parent or similar.



Note: It is best practice to always evaluate your formula on the Function editor by using a sample object on which the key could be implemented using the “Evaluate Node” option before activating the Key. For more information about the Function editor, see the **Using Function Editor** section of the **System Setup / Super User Guide** documentation.

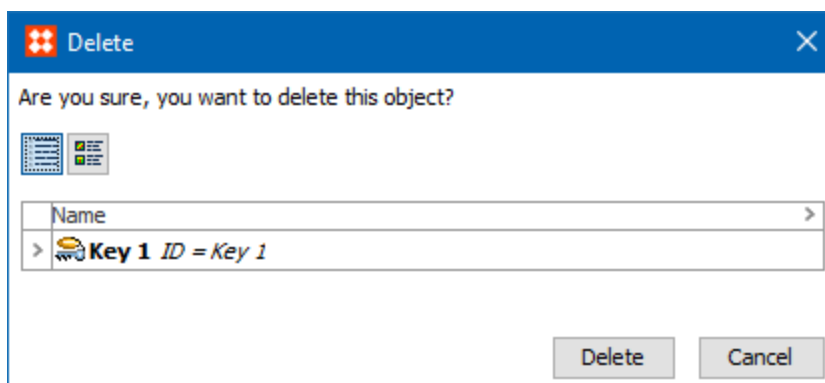
Once a key is created, it must be activated before it can be used. For more information, see the **Activating and Deactivating Keys** topic within this guide.

Optionally, before activating the key, use the Check Key function to confirm the setup of the STEP Key. For more information, see the **Checking Keys** topic within this guide.

To Delete a Key

A key must be deactivated before it can be deleted. Below are steps to delete a key, after it has been deactivated.

1. Go to System Setup > Expand Keys > Right-click the relevant key > Click **Delete Key**, and the Delete dialog will display.



2. Click the **Delete** button.

Activating and Deactivating Keys

When a Key is first configured, it will be inactive and must be activated to take effect. The activation is performed by a background process that calculates and stores keys for the affected objects, and only allows the key to become active if the keys can be created for all specified object types.

Keys will only be generated for objects that have local values for all key attributes. For the objects that do not have local values for all attributes and for new objects added later, keys will be generated once all key attributes have local values.

Once a key has been activated, in the workbench, key attribute fields with values will be read-only and only users with a special 'Modify unique key value' privilege can update the value (Edit menu > 'Edit Unique Key Values'). It will not be possible to update the value via an import or via the API's.

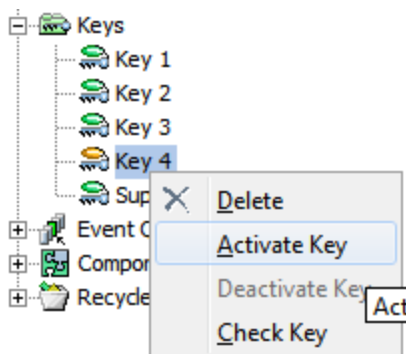
Prerequisites

It is expected that anyone activating a STEP Key is familiar with the **Defining Keys** section of the **Keys** topic within this guide.

Activate a Key

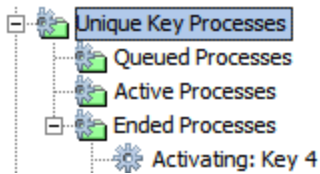
Below are steps to activate a key.

1. Go to System Setup > Expand Keys > Right-click the relevant key > Click **Activate Key**.



A background process is started that tries to generate keys for all objects of the specified object types. If identical keys are generated for two or more objects, this is reported as errors in the Execution Report. The generated key and the conflicted objects are listed in the report.

2. To view the Execution Report, in **Background Processes**, expand **Unique Key Processes**, and then select the relevant completed process.

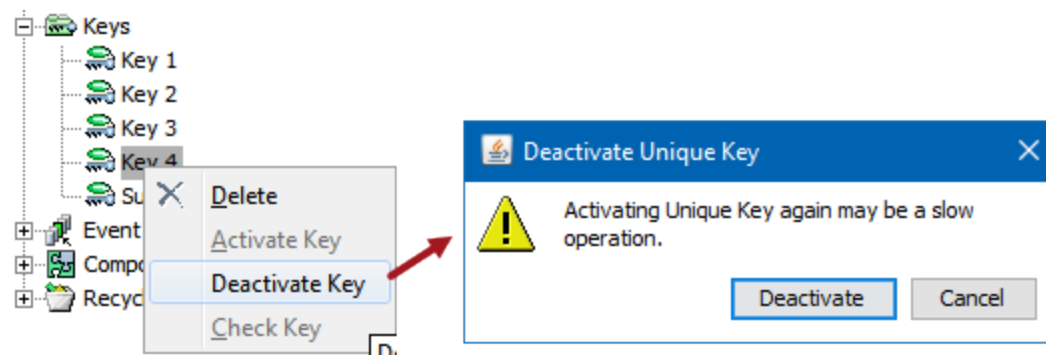


Keys are not generated for objects with missing key attribute values. Once values are supplied for all key attributes of an object, keys are generated on the fly.

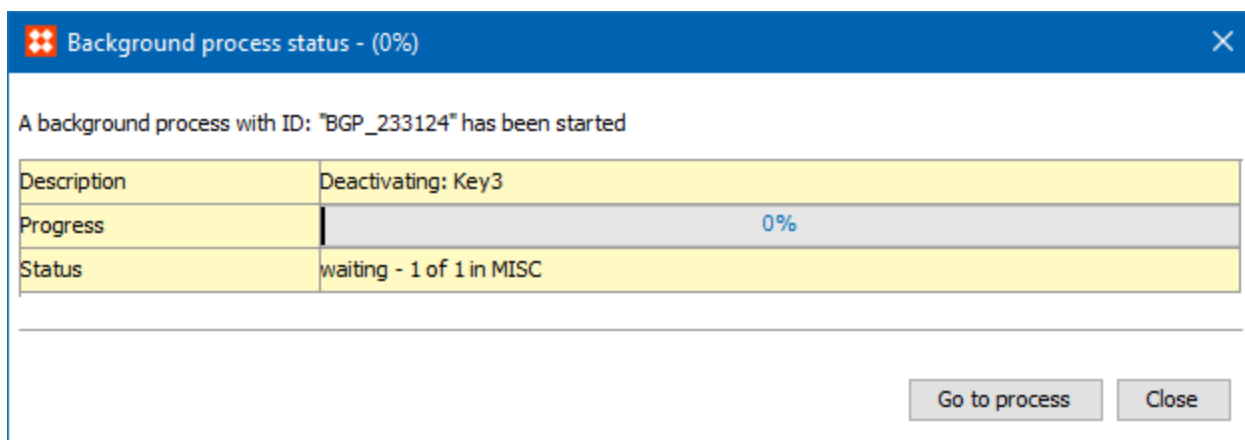
When a key has been activated, it is no longer possible to edit the values on the key attributes, unless you have set up special user privileges. However, you must deactivate the key first.

Deactivate a Key

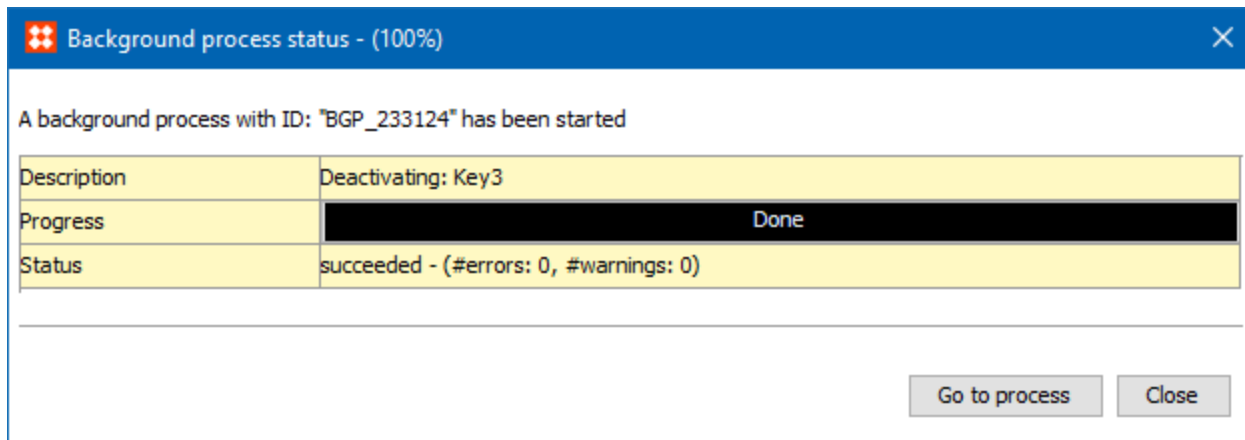
1. In **System Setup**, expand **Keys**, right-click the **relevant key**, select **Deactivate Key**, and the Deactivate Unique Key dialog will display.



2. Click the **Deactivate** button, and the Background process status dialog will display. A BGP will start and attempt to deactivate the key.



3. Once the BGP has completed, the Background process status dialog will display as 'Done' (as shown below).



4. Optionally, click the **Go to process** button to view the Background Process Execution Report.

BG Processes

- Report Process
- Republish nodes
- Revive Recursively
- RuleSetExporter
- RuleSetImporter
- STEP Workflow Profiling
- Scheduled Processes
- Seal Change-Package
- Smartsheet Export
- Smartsheet Import
- StateflowDeadline
- Subscription import
- SuspendBGPs
- Translation
- Translation Import
- Unique Key Processes
 - Queued Processes
 - Active Processes
 - Ended Processes
 - Activating: Key3
 - Deactivating: Key3**
- Unlink Object Type
- Update actual page from plan
- UploadDTPDocument
- Web UI Asset Pipeline Download
- Web UI Download
- Web UI Upload

Deactivating: Key3 - Background Process

Background Process
Queue Info

Properties

Property	Value
Started by	USERM
Id	BGP_233124
Description	Deactivating: Key3
Execution Server	doc-dev
Progress	Done
Status	succeeded
Created	Fri Aug 04 18:24:56 EDT 2017
Started	Fri Aug 04 18:24:59 EDT 2017
Finished	Fri Aug 04 18:24:59 EDT 2017
Processing Time	0 m 0 s
Time in Queue	0 m 3 s
# of warnings	0
# of errors	0

Execution Report

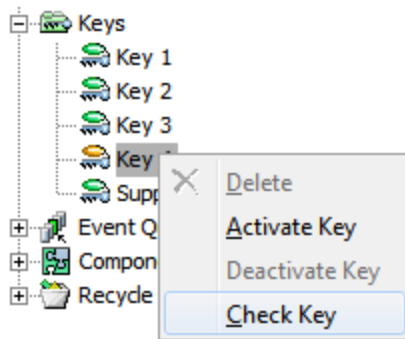
- 1 Deactivation of unique key with ID Key3 started. (Fri Aug 04 18:24:59 EDT 2017)
- 2 Synchronized. Deactivation continues. (Fri Aug 04 18:24:59 EDT 2017)
- 3 Successfully deactivated unique key with ID Key3. (Fri Aug 04 18:24:59 EDT 2017)

1-3 of 3
Save...
Truncate

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 object types.

If identical keys are generated for two or more objects, this is reported as errors in the Execution report. The generated key and the conflicted objects 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 object data.

Creating Objects with Active Keys

Once a key has been activated the following options are available for creating and/or updating objects:

- STEP XML
- Web Services APIs
- Web UI Simple Importer
- Workbench

STEP XML

Once a key has been activated, external systems can identify objects in STEP (with generated Keys) via their resolved key values or by providing values for the key attributes. In the example below, the key 'EAN' is active, and Sales Item Product objects can now be updated by external systems that only know the EAN of an object and not its STEP ID.

In STEP XML a special 'KeyValue' element is available but it is also possible to identify a Sales Item simply by providing the value for the EAN attribute. Both STEP XML snippets shown below would update the existing Sales Item with EAN 6725372937625.

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="GL" WorkspaceID="Main">
  <Products>
    <Product>
      <KeyValue KeyID="EAN">6725372937625</KeyValue>
      <Values>
        <Value AttributeID="HE Color">Black</Value>
      </Values>
    </Product>
  </Products>
</STEP-ProductInformation>
```

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="GL" WorkspaceID="Main">
  <Products>
    <Product>
      <Values>
        <Value AttributeID="EAN">6725372937625</Value>
        <Value AttributeID="HE Color">Black</Value>
      </Values>
    </Product>
  </Products>
</STEP-ProductInformation>
```

```

    </Product>
  </Products>
</STEP-ProductInformation>

```

Note: If no object with the specified key exist in advance, STEP will try to create a new object and thus Object Types selected for key definitions should have Auto ID configured. Apart from this, successful creation depends on parent information and Object Type information being supplied in the file. The KeyValue element data is only used for matching and you cannot set a key using it. Also, if there is a mismatch between the KeyValue value and actual attribute values, the values take precedence.

For Generic XML and the tabular formats, STEP IDs can likewise be left out and an object identified via values for key attributes. Notice here that the 'Identify Destination' step of the Import Manager currently does not support Keys and objects therefore in this step incorrectly can be identified as being new.

For more information about creating objects using STEPXML, see the **Create Objects in STEPXML** topic within the **STEPXML Format** documentation.

Web Services APIs

Both SOAP APIs and REST APIs have key requests available. See the respective sections below for details.

SOAP APIs

For the SOAP API a 'queryByKey' request is available:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://stibo.com/step/ws/step/1.0"
  xmlns:ns1="http://stibo.com/step/ws/types/1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:queryByKeyRequest>
      <ns:accessContext>
        <ns1:userName>*****</ns1:userName>
        <ns1:password>*****</ns1:password>
      </ns:accessContext>
      <ns:keyConditions>
        <ns1:keyExternalID>EAN</ns1:keyExternalID>
        <ns1:key>6725372937625</ns1:key>
        <ns1:type>product</ns1:type>
      </ns:keyConditions>
    </ns:queryByKeyRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

REST APIs

Similarly, for REST, an "objectbykey" resource is available:

GET http://[host]/restapi/objectbykey/EAN/6725372937625?context=Context1&workspace=Main

Web UI Simple Importer

New objects with active keys can be created and/or updated using the Simple Importer Widget within Web UI. For more information, see the **Simple Importer Widget** topic within the **Using a Web UI** documentation.

Workbench

When you use the workbench to create new products of an object type for which active keys exist, the Create Product dialog includes editors for supplying values for key attributes.

Name	Value
Family-Level Key Attribute	abc

The attribute value field can be left blank in which case no key will be generated. As mentioned above, a Key will be generated (and the editor field locked) once a value has been set.

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.

Note: A product can be created without entering the key attribute value, and the value can be added later.

Matching Keys During Import

When you import updates to an object type with active keys, 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.

Currently, the Identify Products step of the Import Manager wizard supports this functionality and reports the matched products with the respective STEP IDs in green (as shown below).

Source:	
ID KeyID=EAN-Key=SU6-EAN	>
33123	
12314425325	

Result:	
Match Result	> ID KeyID=EAN-Key=SU6-EAN
SU6-SalesItem-100877	33123
SU6-SalesItem-100878	12314425325

Buttons: Match, Remove, Back, Next, Finish, Cancel

STEP XML Sample:

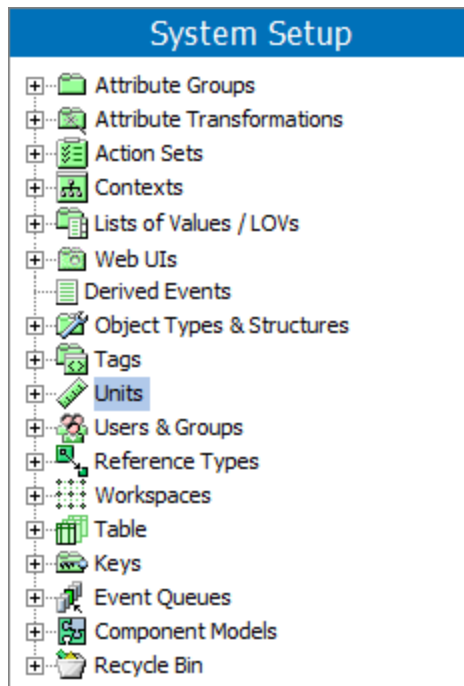
```
<STEP-ProductInformation ExportTime="2013-09-15 16:16:26" ExportContext="GL" ContextID="GL"
WorkspaceID="Main">
```

```
<Products>
  <Product UserID="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

A unit is a representation of measurement. In STEP, 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 a Value and a unit (or unit of measure). This section describes how to setup and maintain units.

Units are maintained in System Setup > **Units**.



The separation of the unit and the value of a measurement means:

- 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 measure.
- More presentation options, e.g. in 'specifications tables.'
- Consistency can be enforced in the use of units.

Note: Units can be applied to Attributes and/or LOVs of all validation base types except: Condition, Text, Date, ISO Date, ISO Date and Time, URL, and Regular Expression.

Example Units:

- Cm
- Meters
- Inches
- Feet
- Yards
- Celsius
- Fahrenheit

Unit Groups

A unit group is a grouping of compatible units. 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 and List of Values (LOVs).

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 the 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

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.

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:

1. Multiply the Celsius temperature reading by 1.8 to get 180.
2. Then add 32 to 180 and get 212 degrees Fahrenheit.

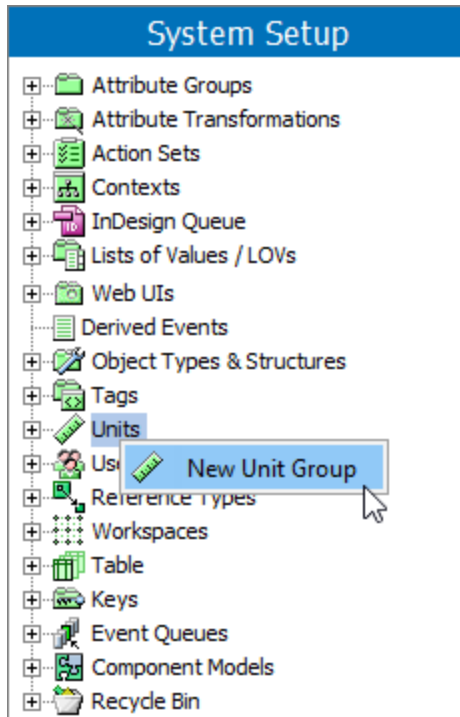
Note: A conversion factor and conversion offset must be typed with minimum one decimal. The conversion factor and conversion offset should never be empty. In case one of them is '0' (zero), '0.0' has to be typed.

Maintaining Unit Groups

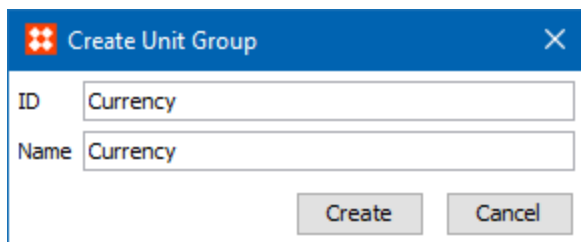
Creating a Unit Group

Unit groups are created in System Setup > **Units**.

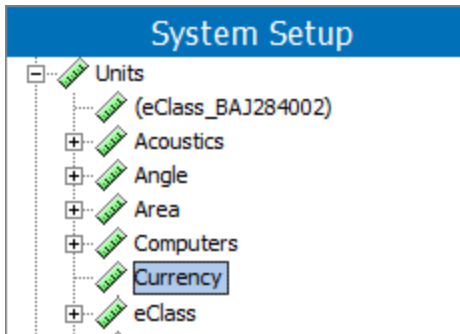
1. Go to System Setup > Units > Right click Units > Click **New Unit Group**.



2. The Create Unit Group dialog will display. Enter an **ID** and **Name** for the new unit, and click the **Create** button.



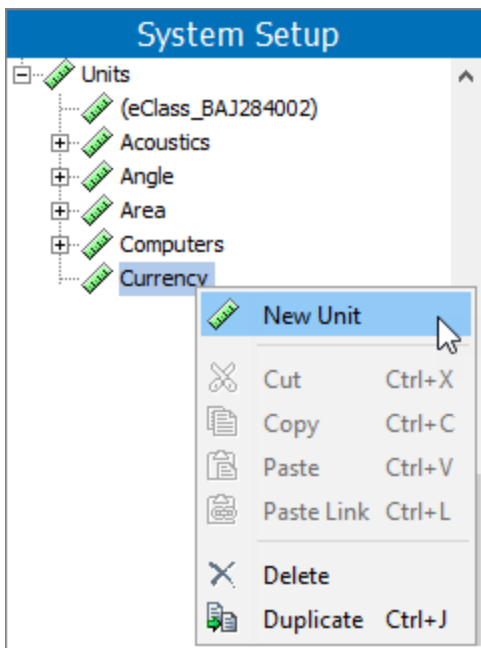
3. The newly created unit group will display in System Setup (as shown below).



Creating a Unit

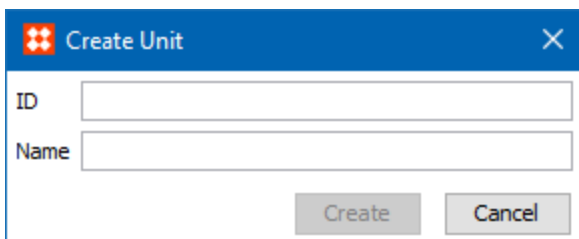
Units are created in System Setup > **Units**.

1. Go to System Setup > Units > Click the relevant Unit Group > Right click the relevant Unit Group > Click **New Unit**.



Note: You may have to create a new unit group for a new unit.

2. A Create Unit dialog will display.



3. Enter the **ID** and **Name** of the new unit, click the **Create** button, and the Unit Editor will display the newly created unit within a unit group. The unit can now be selected for attributes.

Setting up a Unit Conversion

Unit conversions are maintained in System Setup > **Units**.

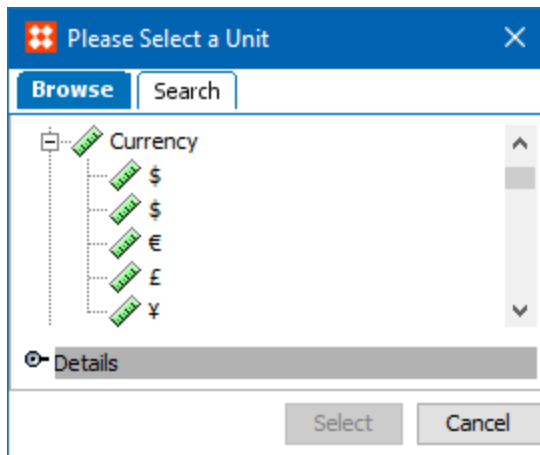
A Unit conversion is a conversion script ('translation') from one unit to another. Standard units are defined with standard conversions.

1. Go to System Setup > Units > Expand the relevant unit group > Click a unit and the Unit Editor will display as shown below.

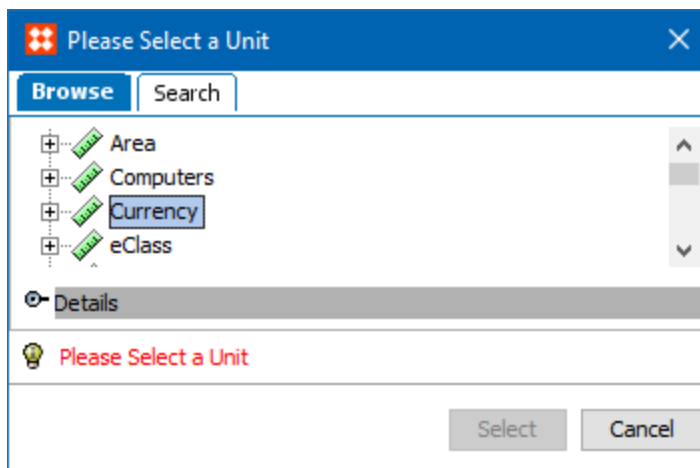
Name	Value
ID	iso4217.unit.USD
Name	\$
Last edited	2008-02-25 08:28:25 by STEPSYS
Path	Currency/\$
Base Unit	
eCl@ss Description	abc
eCl@ss ID	abc
eCl@ss Import Version	abc
eCl@ss Primary Key	abc
GDSN unit map (publisher)	abc
Unit Description	abc
Unit Description	abc US Dollar

2. Click the **Units** tab.
3. In the Base Unit field click the ellipsis button (...), and an **Edit Conversion Rule** dialog displays.

4. In the **Choose Base Unit** field, click the ellipsis button (...), and a **Please Select a Unit** dialog displays.



5. Use the Browse or Search tabs to find and select the relevant unit, and then click the **Select** button. When a unit group is selected, then the user will be warned with red text to 'Please Select a Unit' (as shown below).



6. In the **Conversion Factor** field, type the factor. (See the **Example of Conversion Factor and Conversion Offset** section of this topic.)
7. In the **Conversion Offset** field, type the offset. (See the **Example of Conversion Factor and Conversion Offset** section of this topic.)
8. Click the **OK** button, and the conversion rule that has been created between two units displays in the **Base Unit** field.

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

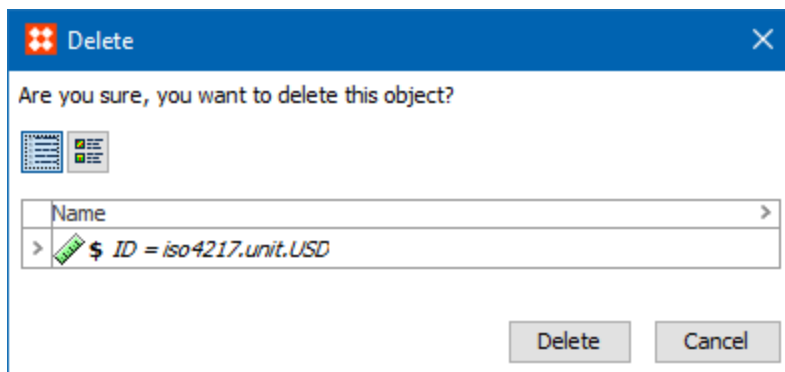
Unit conversions are removed in System Setup > **Units**.

1. Go to System Setup > Units > Expand the relevant unit group > Click the relevant Base Unit where an assigned conversion rule has to be removed. An Edit Conversion Rule dialog displays.
2. Click the **Delete** button, and the conversion rule will now be removed from the base unit.

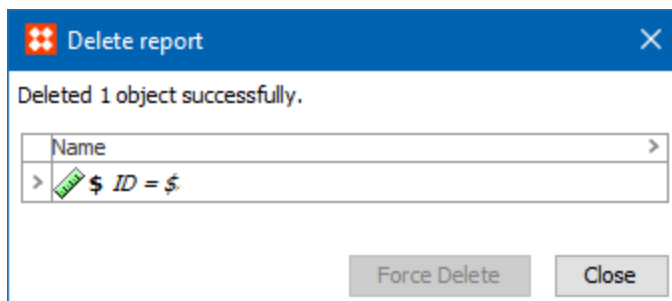
Removing a Unit or Unit Group

Units are deleted in System Setup > **Units**. A unit group containing a unit cannot be deleted. All units must be removed from the group before the unit group can be deleted. 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. Go to System Setup > Units > Click the relevant unit or unit group to be deleted > Right click the relevant unit or unit group > Click **Delete**, and a Delete warning dialog displays.

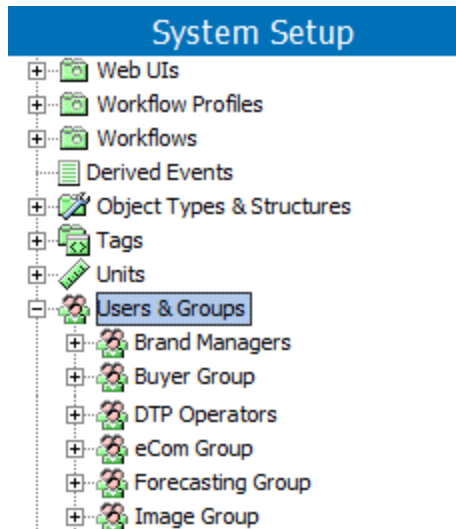


2. Click the **OK** button to delete. The unit or unit group will be deleted, and a Delete report dialog will display. Click the **Close** button to close the dialog.



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

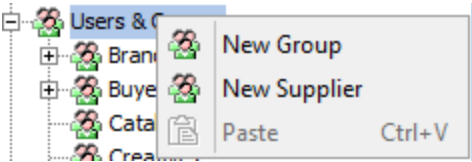
Supplier

Supplier

Creating a User Group

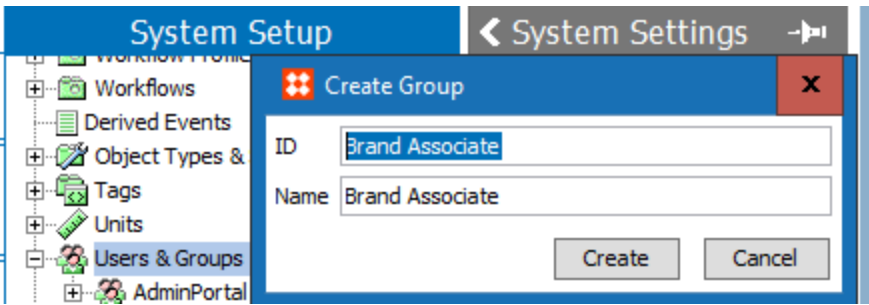
Groups are created in [System Setup]> **Users & Groups**.

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.

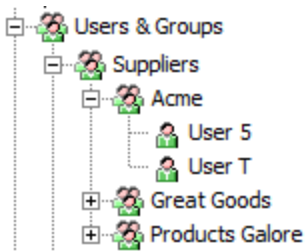


To create a New Group, right-click on the Users and Groups root not and select **New Group**. Enter the ID and Name and click 'Create'. The new group appears in the Group Editor. At this point, Privilege Rules will need to be

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.

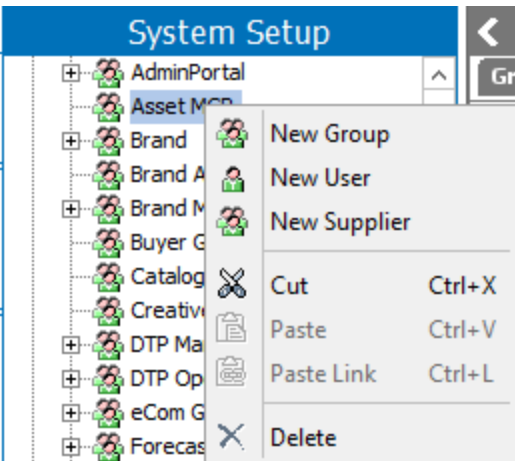


Note: If a supplier user is linked to multiple supplier groups, the supplier can see all of the supplier hierarchies that they are a member of.

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

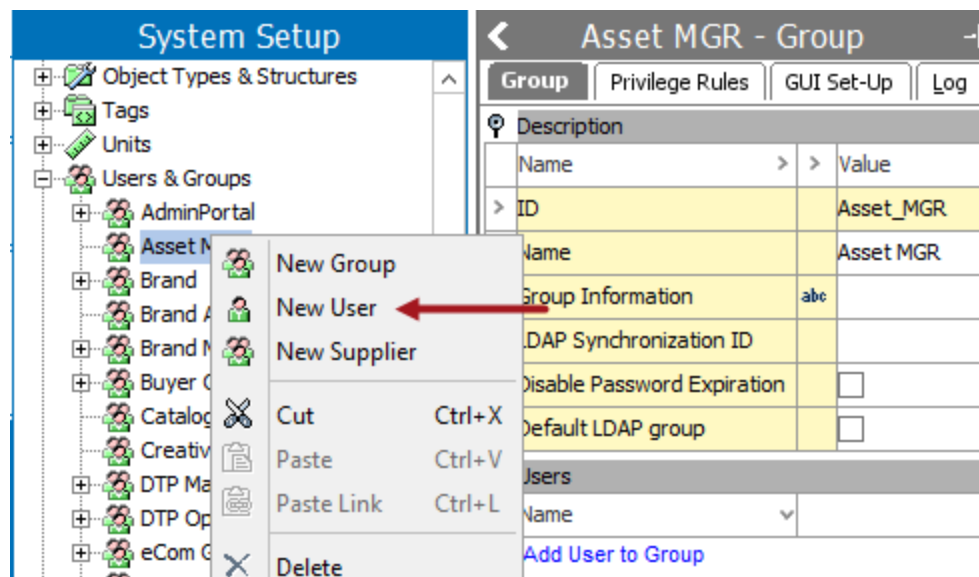
STEP can only be accessed by authenticated Users. This section contains information needed to work with Users in Users and Groups from the STEP Workbench. Information detailed in this topic include:

- 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](#)

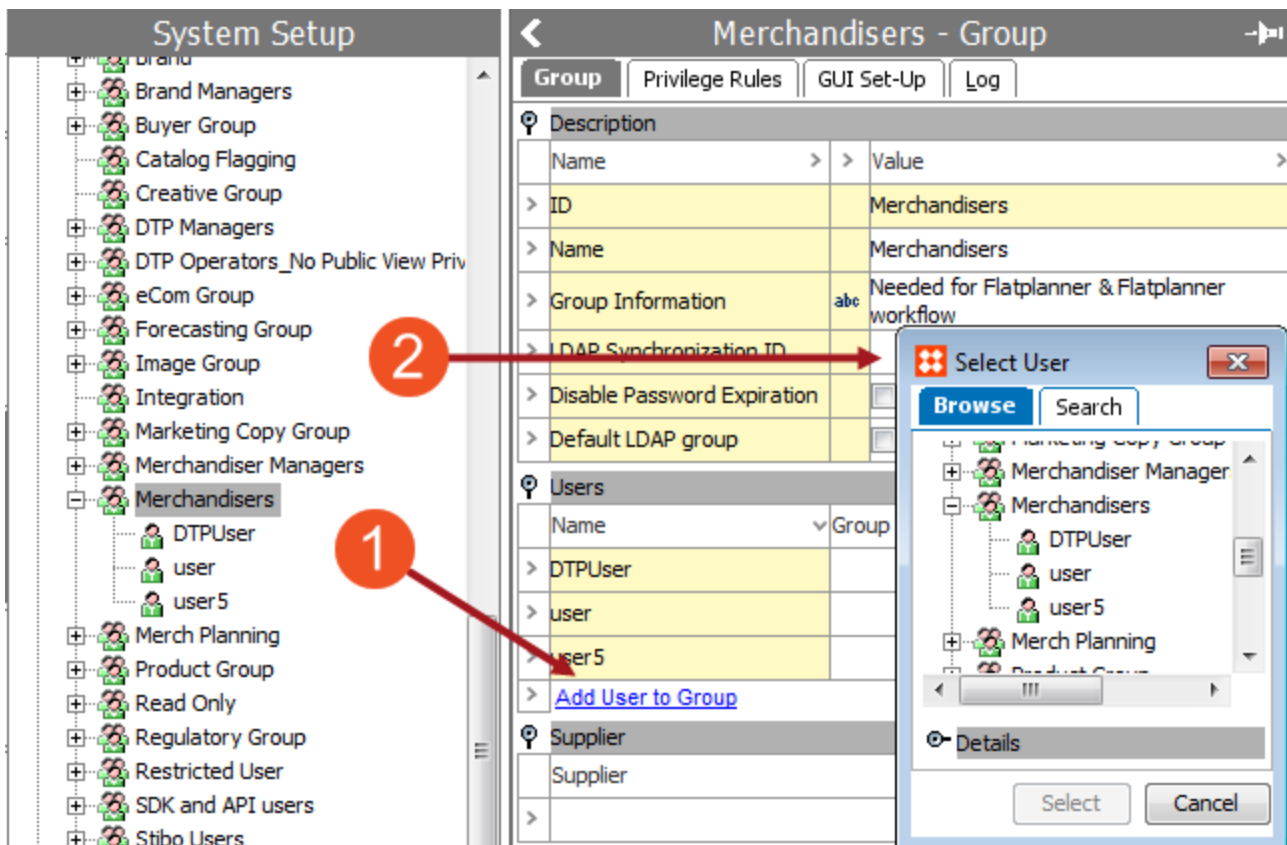
Note: If information fields should be available for Users, they must be applied as Attributes on the Object Type 'Operator user-type root', such as E-mail, Group Information, and User Information as shown in this example.

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.



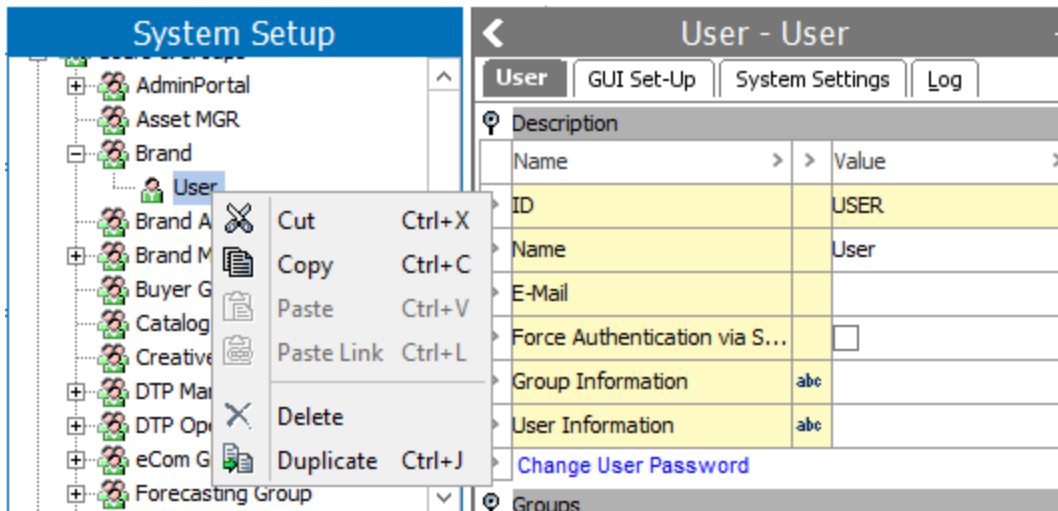
3. Select the relevant User, and then click **Select**.

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'.

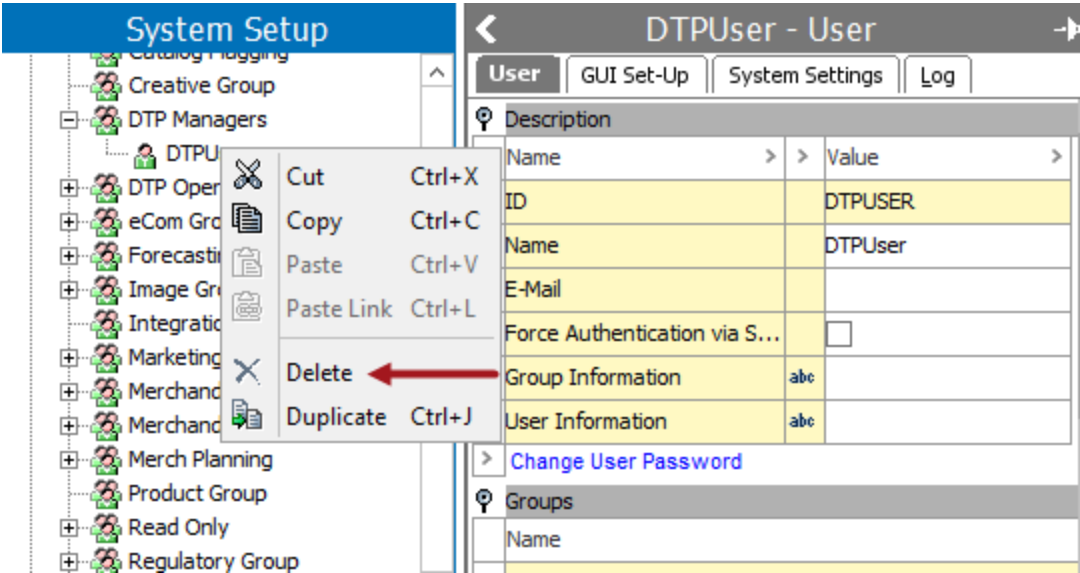


A pop-up window will appear as shown in the image below, and from here, you would follow the steps outlined in **Creating a User**. The duplicated user will have all of the same privileges, GUI Set-Up, and System Settings as the user that was duplicated.

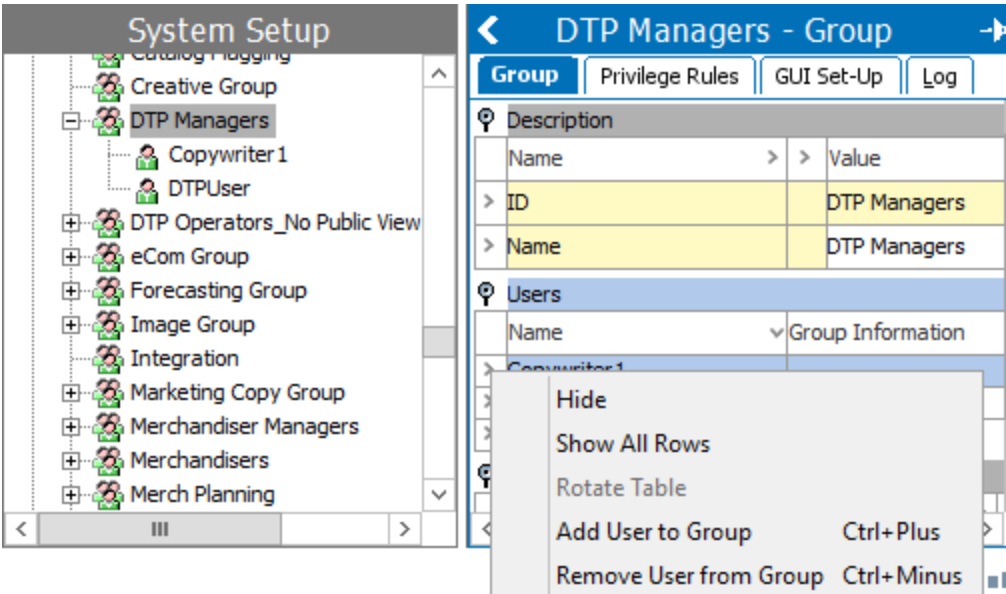
Important: Duplicating a User can be very effective, but if you are using metadata attributes on the duplicated user, the object type values on those will be duplicated as well. So for instance, if a user e-mail address value is in the user being duplicated, that value will be carried over to the new 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.



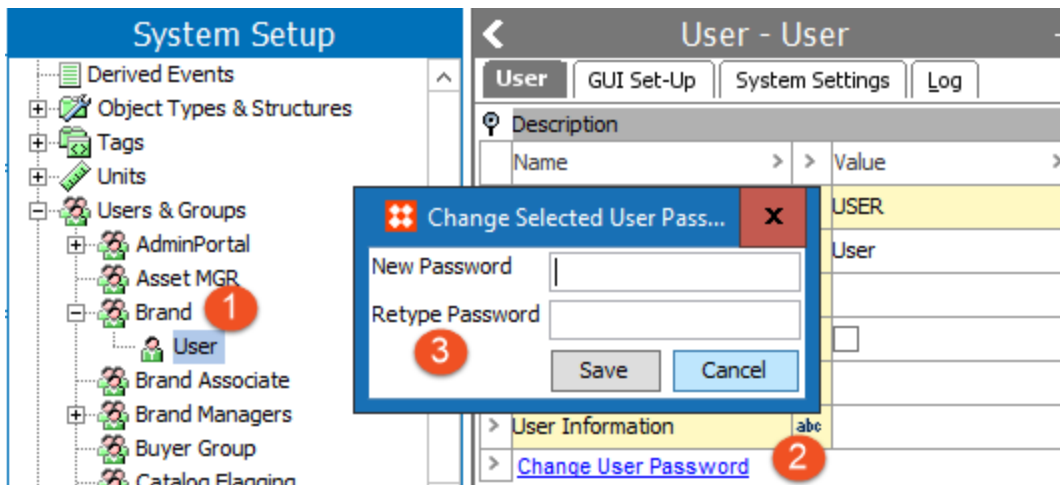
Changing a User Password

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.

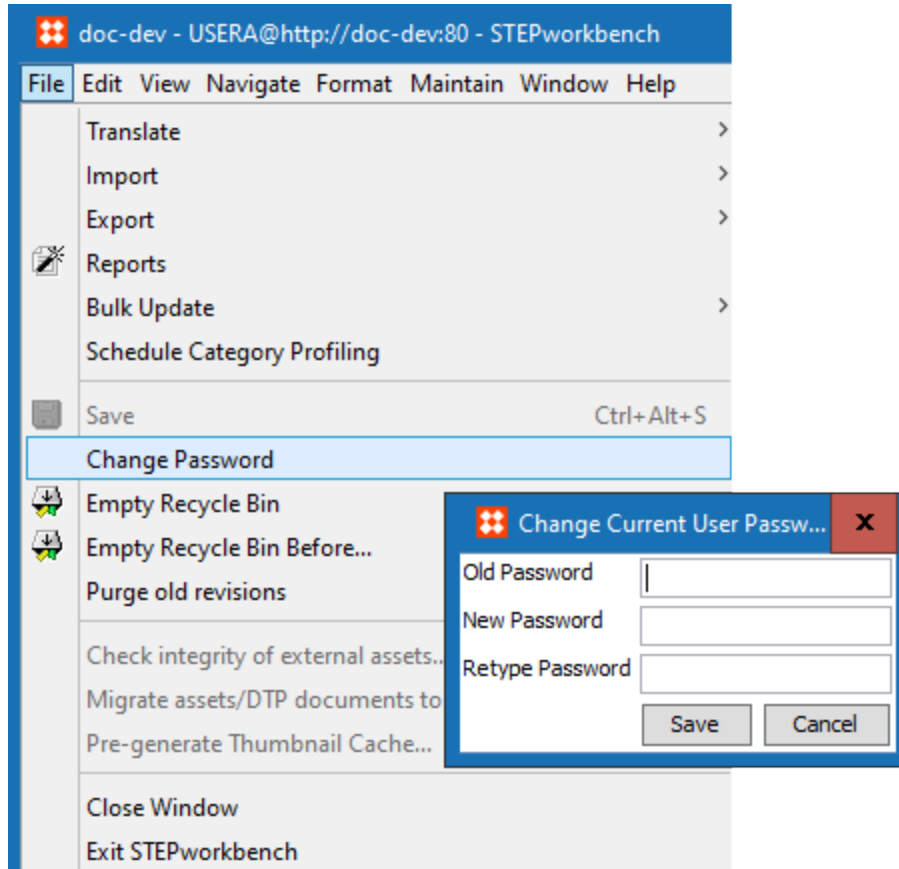
Note: On the Users and Groups root node, you can define a password security policy to be used. Information about setting security policies can be found in the **Security Policy** section of Users and Groups.

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.

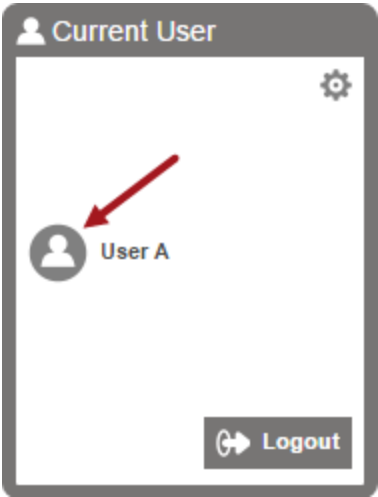
From the Workbench:

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

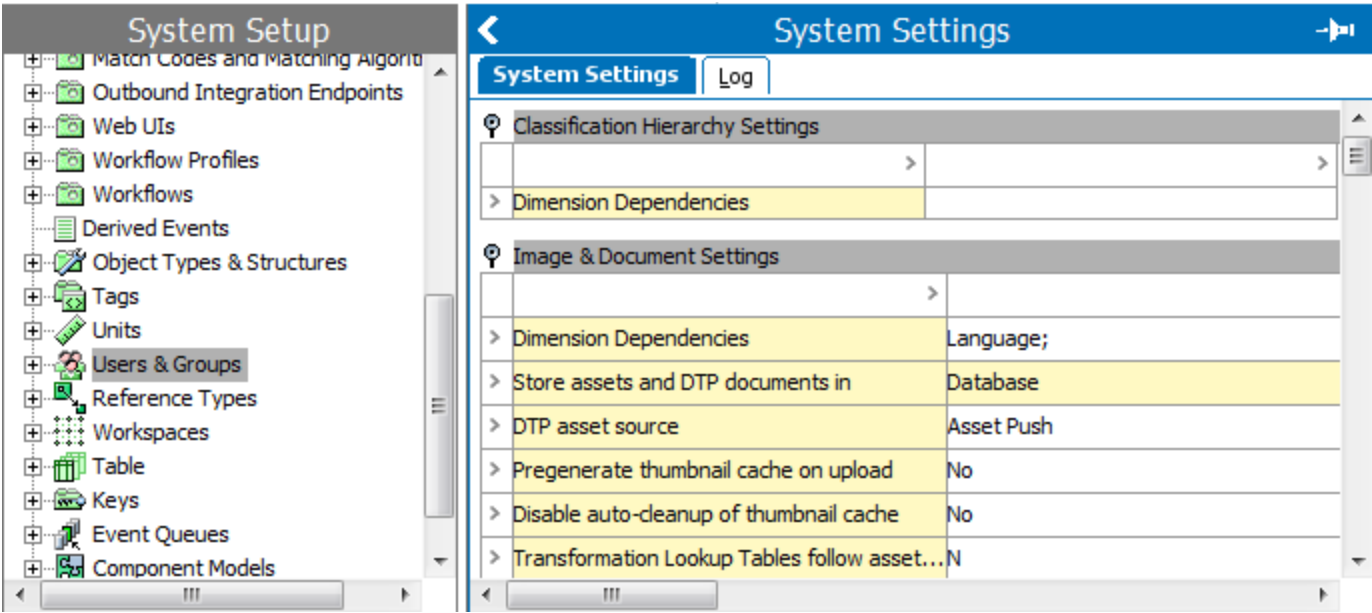
Save

Reset

System Settings

Global system settings are maintained in the **System Setup** tab, under the **Users & Groups** root node.

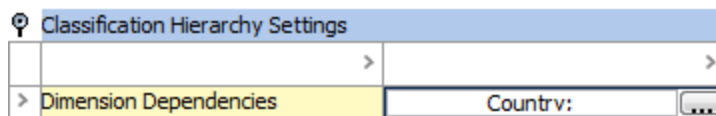
In System Setup, click **Users & Groups**. A **System Settings** editor appears. It lists a number of general and global settings on the system. Each of these settings will be described throughout this section.



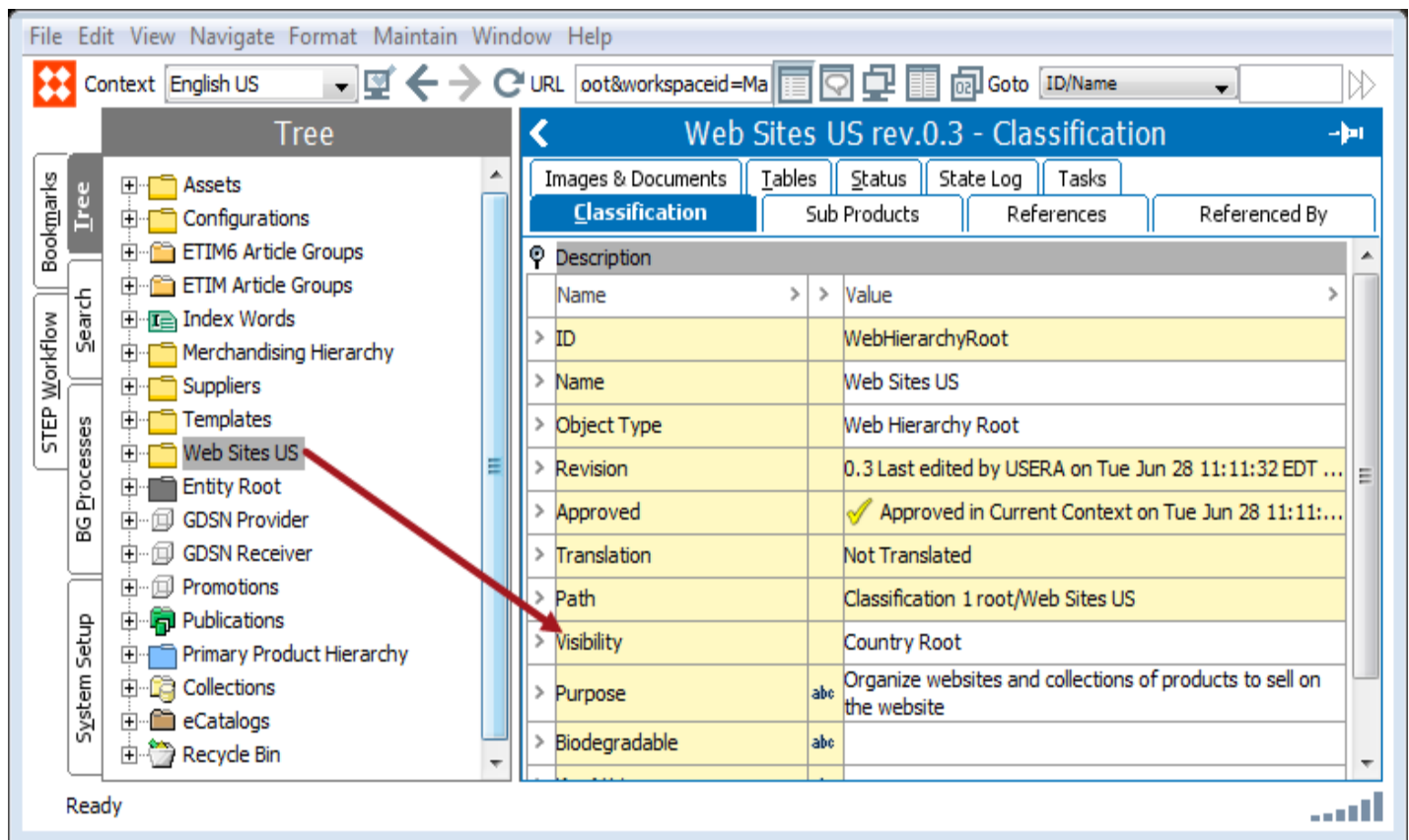
Classification Hierarchy Settings

Dimension Dependencies

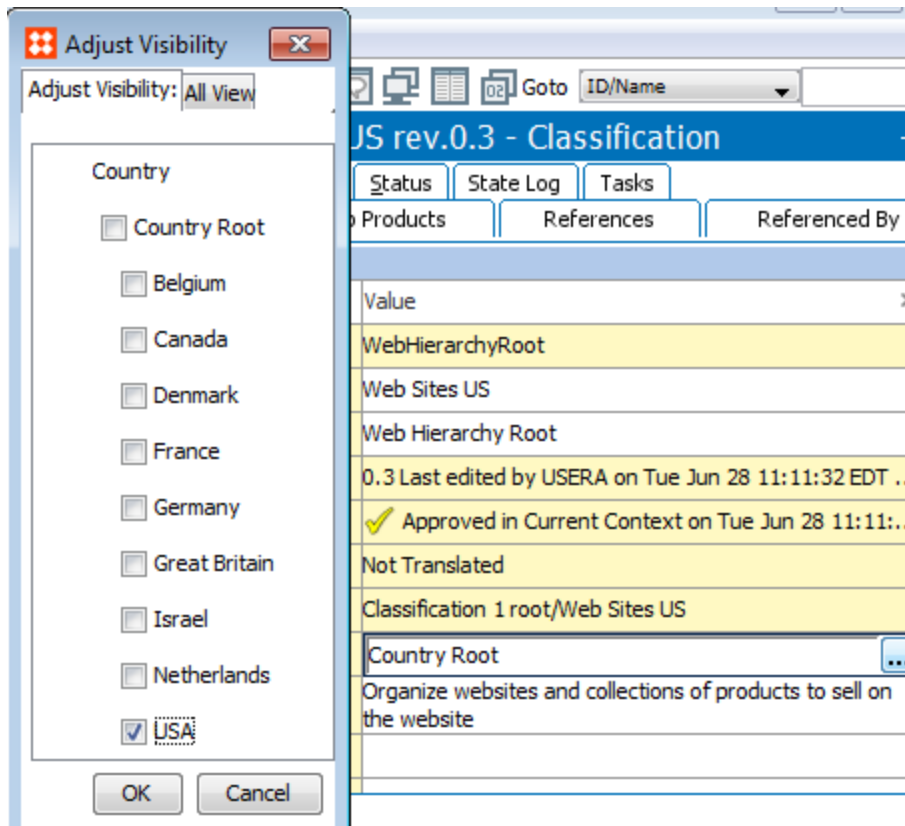
This option allows for **folders** in classification structures to be dimension-dependent. This is different than the dimension dependencies that are set on the object type. Selecting dimension dependencies here will determine whether classification structure will or will not be shown in specific languages or countries. In the example below, the dimension dependency has been set for country by clicking on the ellipses [...] button and selecting Country.



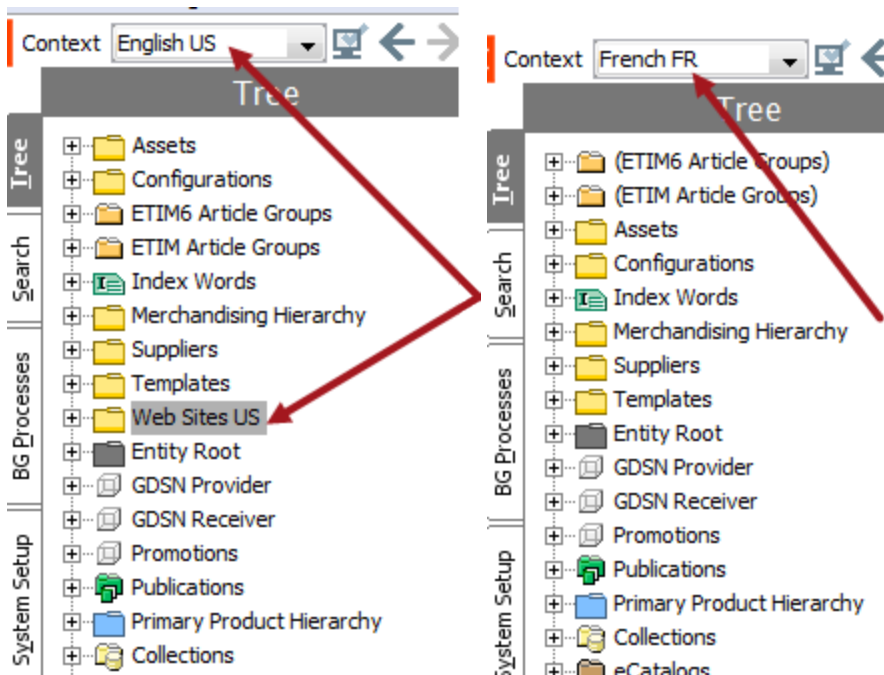
Selecting this dimension dependency added the Visibility metadata attribute to the classification objects in Tree.



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.

Context: Danish DK

Tree

- (ETIM Article Groups)
 - Assets
 - Configurations
 - Index Words
 - Merchandising Hierarchy
 - Suppliers
 - Web Sites
 - Acme Retail Web Site
 - Apparel
 - Mens Casual
 - Caps
 - Mens Tops**
 - T-shirts and Sweatshirts
 - Women's Apparel

Mens Tops rev.0.1 - Classification

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.

Context: English US

Tree

- Assets
- Configurations
- ETIM Hierarchy
- Index Words
- Merchandising Hierarchy
- Suppliers
- Web Sites
 - Acme Retail Web Site
 - Apparel
 - Mens Casual**
 - Caps
 - T-shirts and Sweatshirts
 - Women's Apparel

References

ID	Name
115215	Mens Tops

Image and Document Settings

This section is used to describe the various options available in the **System Setup > Users & Groups > Image & Document Settings**.

System Setup

Tree

Search

BG Processes

System Setup

+

 Workflows

+

 Derived Events

+

 Object Types & Structures

+

 Tags

+

 Units

+

 Users & Groups

+

 Reference Types

+

 Workspaces

+

 Table

+

 Keys

+

 Event Queues

+

 Component Models

+

 Recycle Bin

System Settings

Log

Classification Hierarchy Settings

Image & Document Settings

>

 Dimension Dependencies

>

 Store assets and DTP documents in

>

 DTP asset source

>

 Pregenerate thumbnail cache on upload

>

 Disable auto-cleanup of thumbnail cache

>

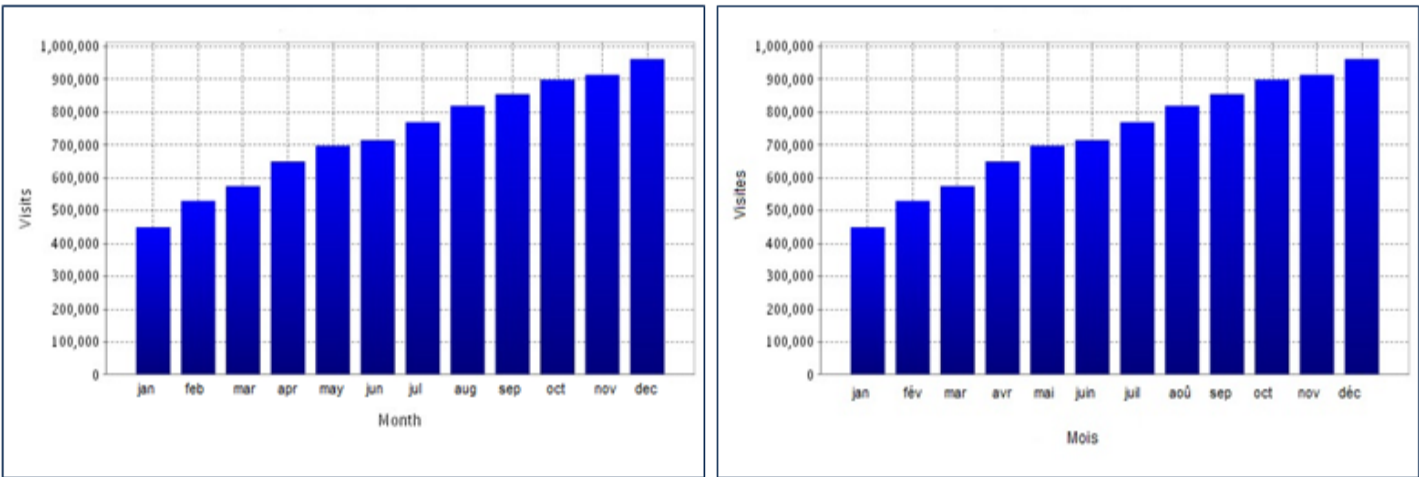
 Transformation Lookup Tables follow asset dimension dependency

>

 Asset Import Compatibility Mode

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 Import Compatibility Mode

Two options are available for this setting: 'Simple' and 'Advanced'. 'Advanced' is the default setting on new STEP systems, and enables the Asset Importer functionality. Selecting 'Simple' will enable the legacy image import functionality and disable the Asset Importer.

Asset push attribute macro keys

The **Asset push attribute macro keys** are user defined keys that are designed to extract values from any Description attribute on an asset so they can be used in place of the file name in the asset push path. These are created when a user has opted to use a **Relative Path Template** (RPT) which is a folder structure that is not based using the asset ID. For more information regarding Relative Path Templates including the use of the **Asset push attribute macro keys**, please see the **Relative Path Template** section of the **Asset Push** documentation.

Asset push attribute macro keys		
	>	>
> DescAtt	Description Attribute 1 (DescAtt1)	...
> assetfilename	Calculated Asset File Name (CalculatedAssetFileName)	...
> KeyAtt2	KeyAtt2 (KeyAtt2)	...
> Add attribute key		

For an example using an attribute macro key in a calculated attribute, see the **Asset Push File Name Scenario** documentation.

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. Dimension Dependency 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 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')
```

Calculated Attribute Settings

> Dimension Dependency	Language:	

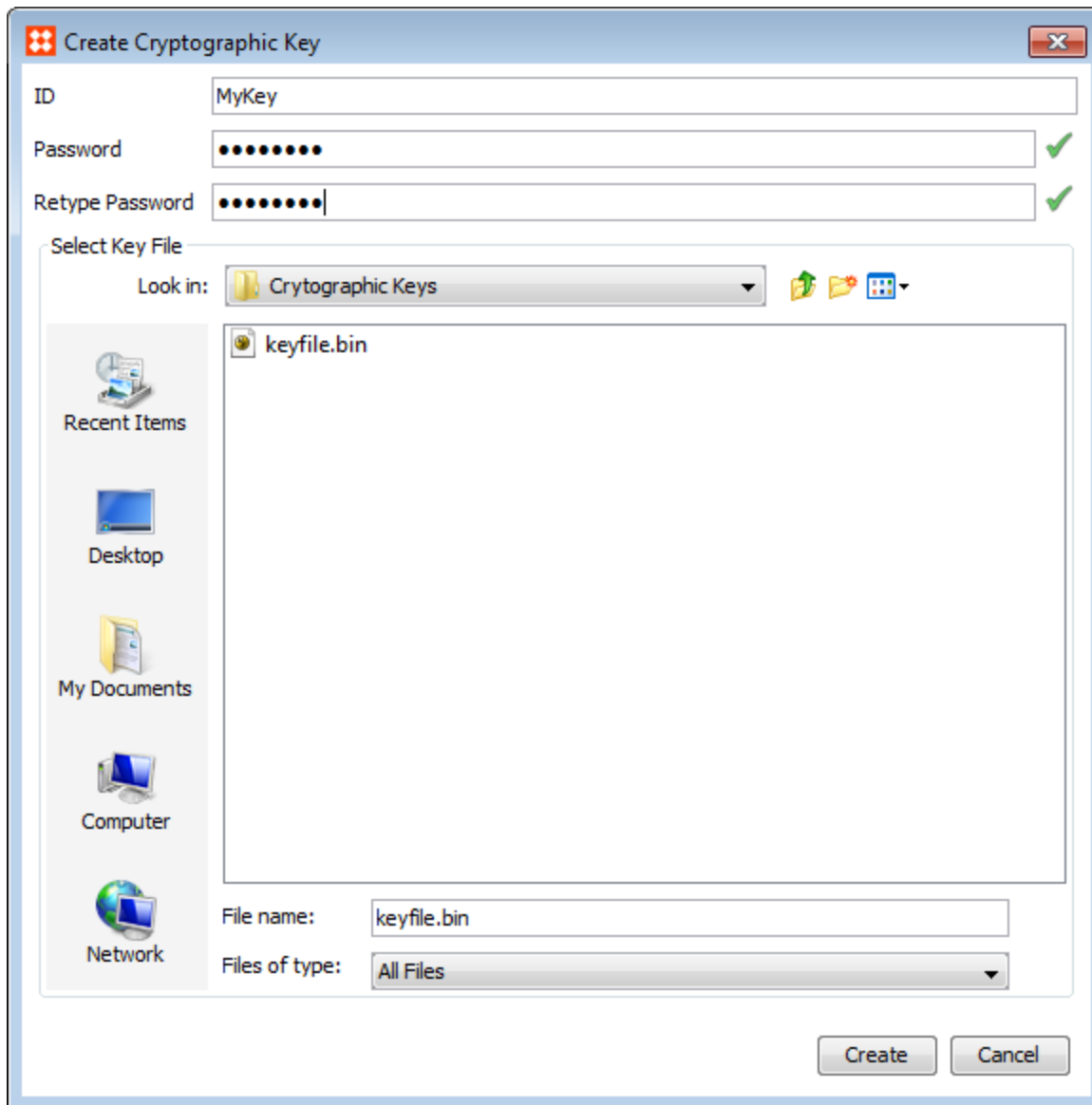
For information about calculated attributes, see the **Calculated Attribute** section of the **System Setup / Super User Guide** documentation.

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. Note that the use of cryptographic keys requires a license.

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).

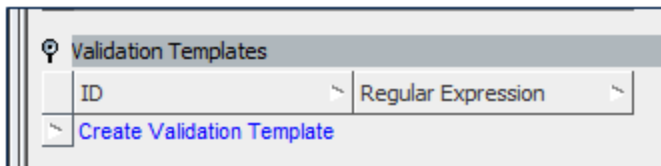
In STEP it is possible to make customized validation base types to be used on Attributes and LOVs. In this field you can create validation templates to be used when creating Attributes or LOVs.

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. For more information on regular expression, see the **Regular Expression** topic in **System Setup / Super User Guide** documentation.

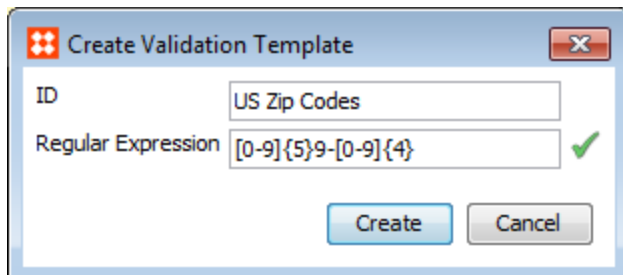
The example below shows how a new Validation Template may be created and how that new template subsequently displays in the drop-down list of validation base types when a new attribute is created (or when changing an existing attribute's validation).

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 Number Number Range Numeric Text Regular Expression Text URL Template: US Zip Codes
☒ Specification ☐ Description	
Cryptographic Key	

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](#)

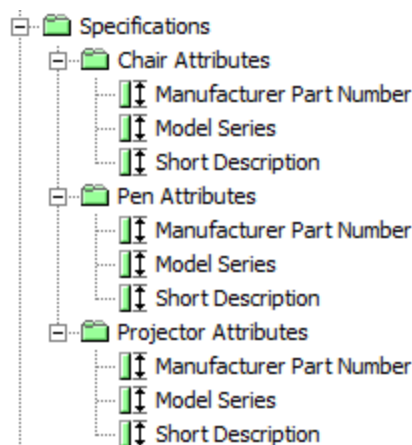
OK

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	

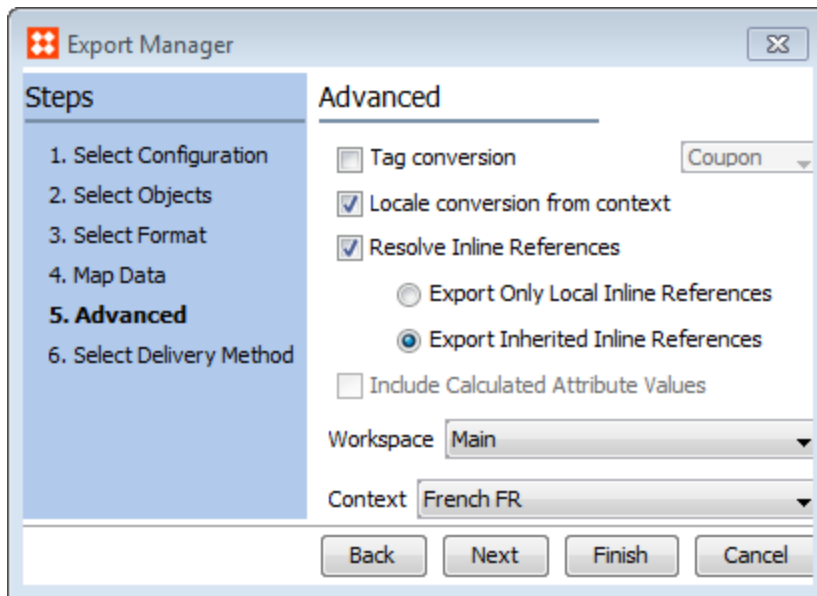
When the value for this setting is set to 'Y', the attributes would roll up into the top group. The user would still have privileges to update 'Chair Attributes' but 'Pen Attributes' and 'Projector Attributes' do not appear.

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. Attributes with the validation of 'Numeric Texts' will not be localized with this setting as they are actually text attributes.

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.

Important: This setting affects the values presented in the table preview for print applications. It also adversely affects any calculated attribute in the table that accesses attributes with **Numeric**, **ISO Date**, or **ISO Date and Time** validations. Common setup for print applications is to keep this setting at 'N' and use Export Manager to control the localization of exports.

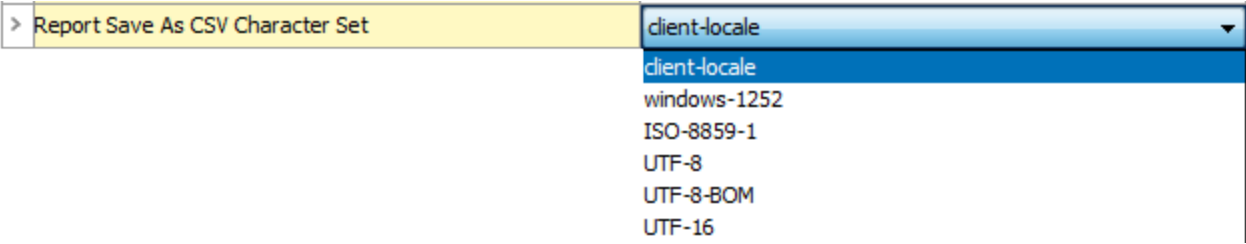
Localize dates when localizing exports

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 (considering note below) and optionally in the STEP Export Manager Advanced screen as shown in the 'Localize numbers with thousand delimiter when localizing exports' section above.

Important: This setting affects the values presented in the table preview for print applications. It also adversely affects any calculated attribute in the table that accesses attributes with **Numeric**, **ISO Date**, or **ISO Date and Time** validations. Common setup for print applications is to keep this setting at 'N' and use Export Manager to control the localization of exports.

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.

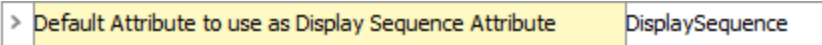


Default Attribute to use as Display Sequence Attribute

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 metadata 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 metadata attribute with the ID 'Display Sequence' was created and selected as the default attribute.



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		
> 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.

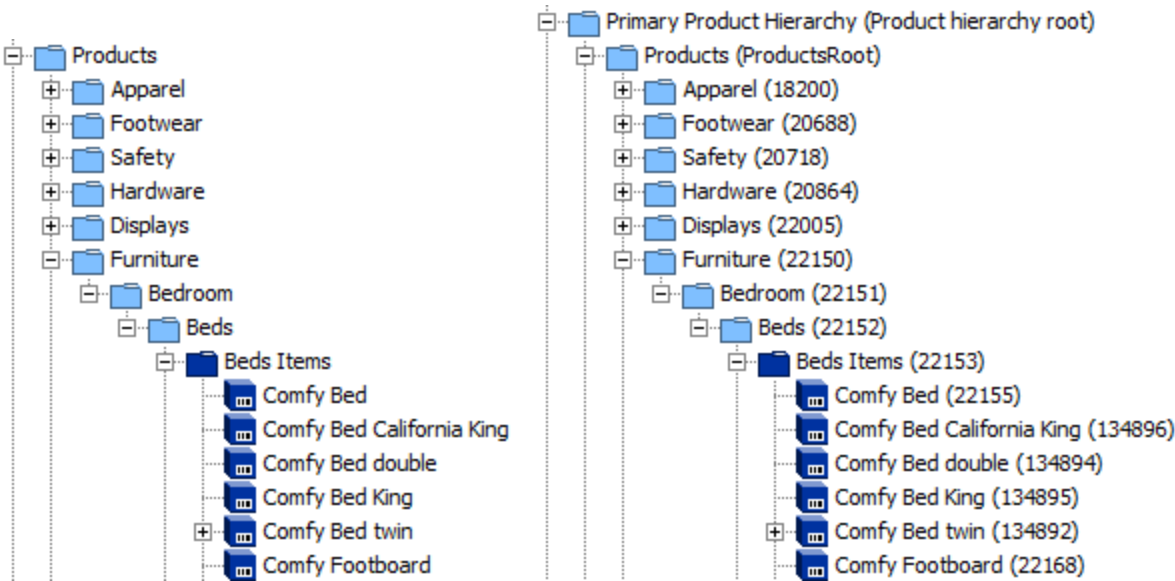
Specifications		
Name	>	Value
> Model Series	abc	
> HazMat	abc	
> Application	abc	
> Short Description	abc	
> Net Weight	abc	
> Gross Weight	ID = Short Description	
> Density	DisplaySequence = 40	
> Manufacturer	Provide a brief description of the product.	
> Plug Included	abc	

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

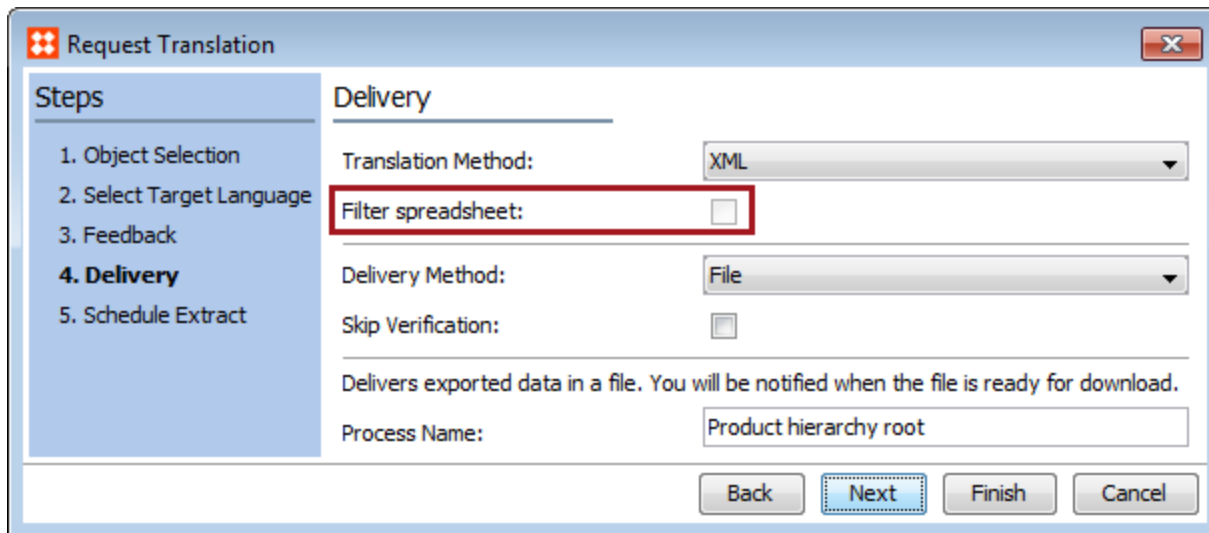
Would create a file with a name such as:

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

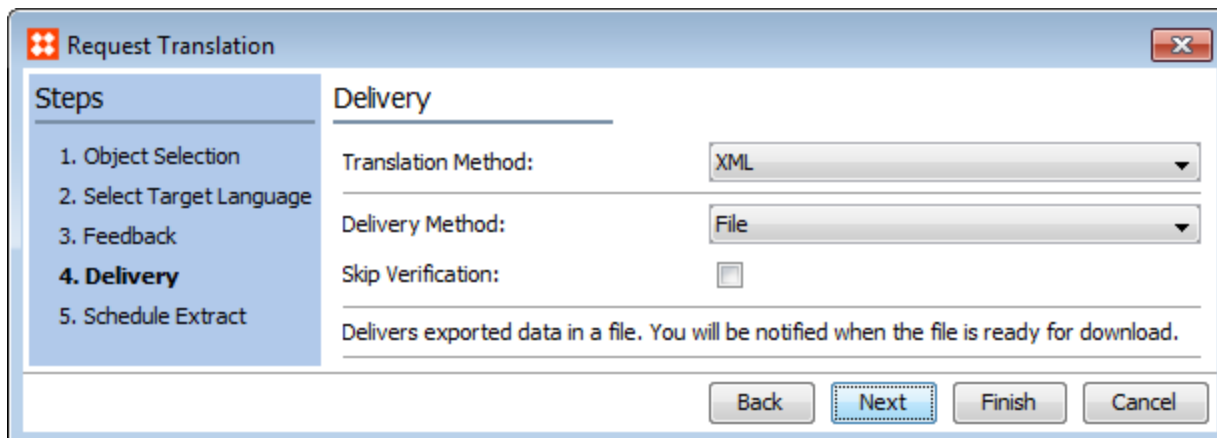
Skip Verification: ☐

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

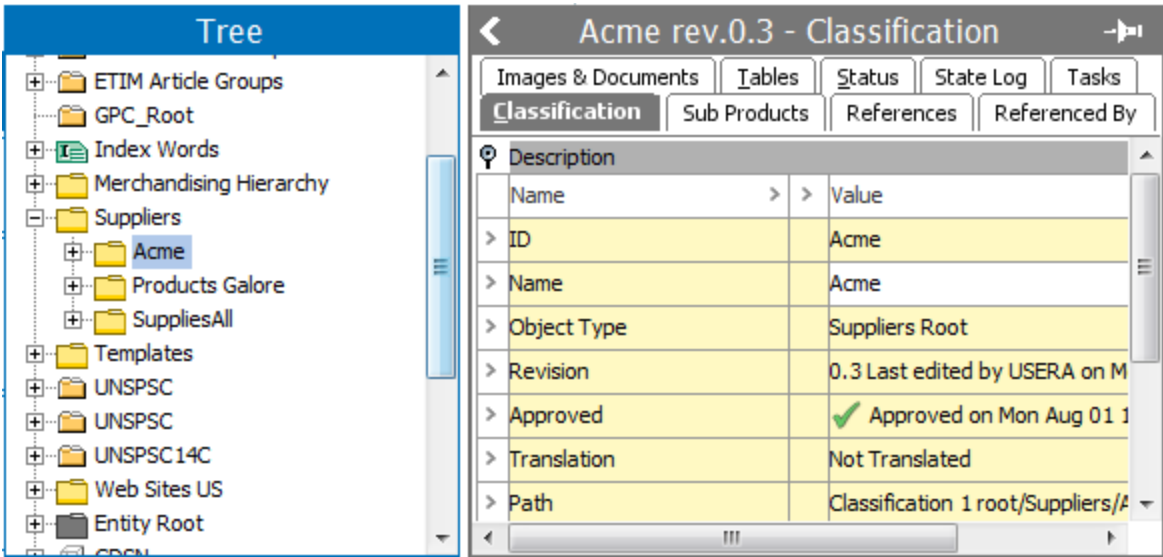
This option determines the Object Type that is 'assigned' to a Web UI user via the Supplier User Group and its associated Classification(s). Filling in this information is only required if you are setting up a Web UI that will include a supplier hierarchy. In the images below, an object type of Suppliers 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

+

+

-

+

-

+

+

-

(111777)

(111777)(2)

(111778)

109308

129707

129997

129999

190 Lumen LED Cyber

20-68204

Products rev.0.3 - Classification

StatusState LogTasks

Referenced ByImages & DocumentsTables

ClassificationSub ProductsReferences

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.

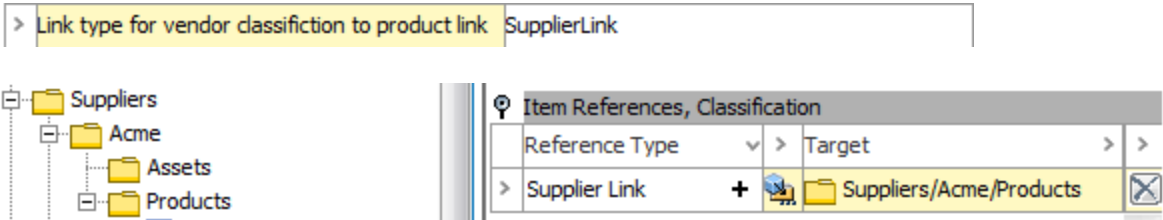
Context Help metadata attribute

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.

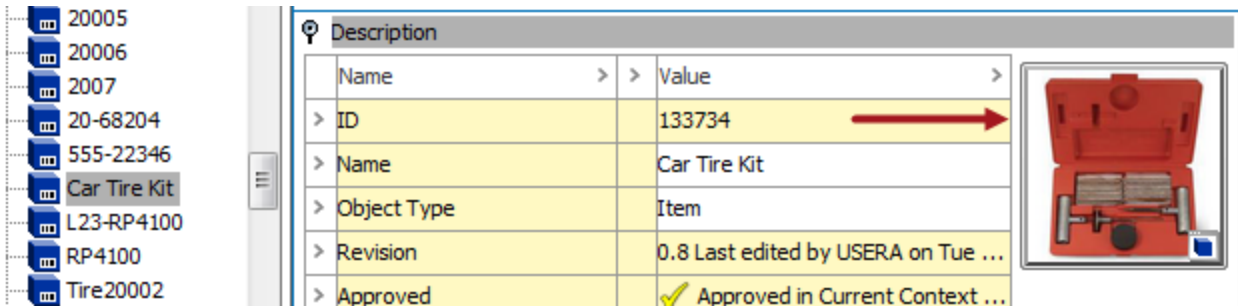


Enable all-view for users that are a member of multiple suppliers

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
>	Show frame information on flatplans with layout pickup	Y
>	Reapply Flatplan rules when adding product to non-empty frame	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.

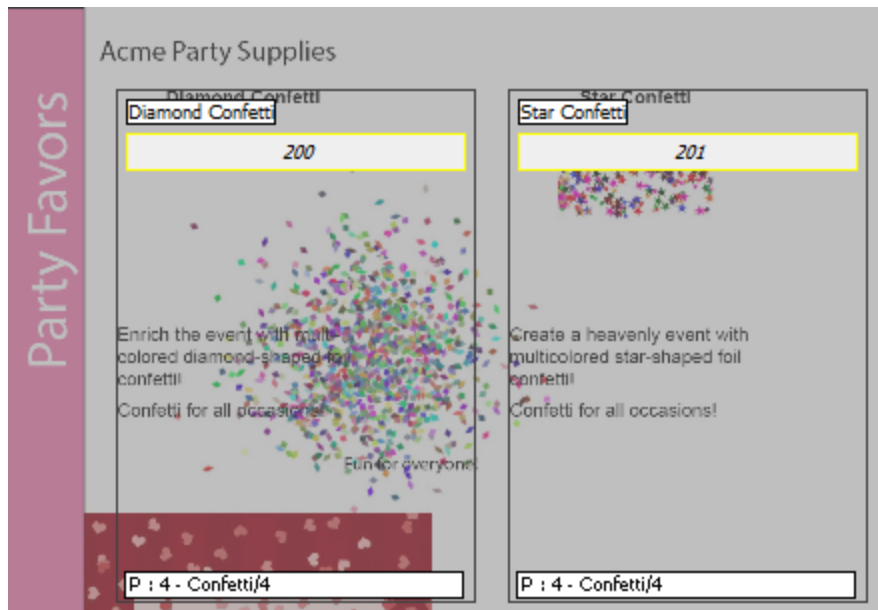
If 'Enable keep layouts for publications' is set to **N**, users are free to save manually adjusted pages back to STEP, but the next time a page is mounted from the corresponding Flatplan, it will mount with the layout from the original product template(s).

(Note that a 'kept' layout will be removed from the Flatplan if the corresponding mounted [actual] pages are deleted from STEP.)

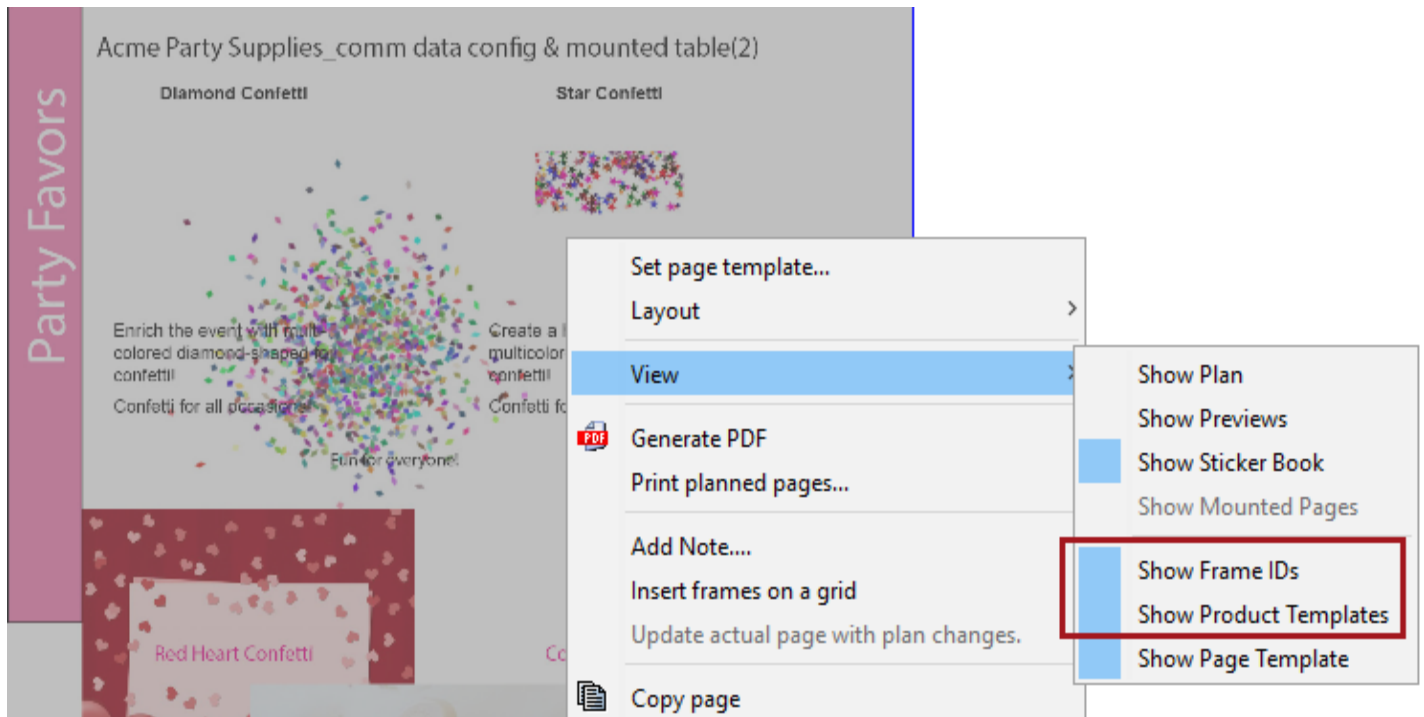
For more information, see **Reusing Saved Layouts** in the **Flatplanner** documentation.

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.



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.

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
> Enable calculation of border rectangle space usage for actual pages	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

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.

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.

For more information, see the **Space Usage in STEP** topic in the **STEP'n'design** documentation.

Calculated DTP Table width includes left and right rules

This option tells STEP to make an adjustment to correct an anomaly in InDesign tables when vertical rules are placed on the left and right side of a table. InDesign does not normally include the width of a left or right vertical rule in the overall width of a table. Instead, it effectively oversets the table's width. Setting 'Calculated DTP Table width includes left and right rules' to 'Y' will readjust the left and right vertical rules so they are within the overall table width.

To further explain, when 'Fit calculated Table Width to Frame Width' is checked in the STEP Preferences menu in InDesign, STEP'n'design will fit the mounted table to the width of the frame on the product template that contains the table tag. If 'Calculated DTP Table width includes left and right rules' is set to 'N,' the table will be slightly wider than the frame because the rule lines are not 'fit' inside the boundaries of the frame. When set to 'N', the table width will exactly match the frame width, as the side rules will be fit inside the frame.

Attribute Display Sequence

The 'Attribute Display Sequence' setting under DTP Settings is different than the 'Default Attribute to use as Display Sequence Attribute' setting under Product Information Manager Default settings. Whereas 'Default Attribute to use as Display Sequence' is used to sort the order of attributes as they display in the STEP Workbench (for example, on the 'Product' tab of a product object), the 'Attribute Display Sequence' setting controls the order of mounted attribute values from an attribute group on mounted InDesign pages.

On product templates, a tag to mount values from an entire attribute group can be placed into a frame (as opposed to the more common practice of placing single attributes at a time into frames), as pictured below.

```
<STEPGROUP AG="ItemDescriptionInformation" RS="\n"></STEPGROUP>
```

When an object is mounted onto an InDesign page using a product template containing this type of tag, the values from all attributes within this group are mounted. The 'Attribute Display Sequence' attribute will determine what order the attributes mount on the page. By default, these values would display in the order in which they reside under the attribute group (which is typically in alphabetical order).

For more information, see the **Mounting Multi-Valued Attributes and Attribute Groups** section of the **STEP'n'design** documentation.

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 paragraph styles in InDesign that are applied to the name and values of attributes, respectively, when mounted as part of an attribute group.

For more information, see the **Mounting Multi-Valued Attributes and Attribute Groups** section of the **STEP'n'design** documentation.

Asset Order Meta-attribute

This feature enables a sequence order to be set on assets (usually images) when products are mounted to InDesign pages. This sequence is set by the use of a metadata attribute that is valid on the link between products and image reference types. If there are multiple references from a product to various images using the same reference type (for example, 'Product Images'), and the product template calls out the Product Images, then all these images mount to the page. However, without this feature, the order that they mount would be in the same order as they are seen in the workbench—by their alphabetically sorted name. By creating a metadata attribute and placing it on the link between the product and asset (reference type = Product Images in this case), then the user may establish a numeric order to the images. If this system setting specifies this metadata attribute, then the images will mount to the InDesign page in the order specified by this 'sequence' attribute.

For more information, see the **Sequencing Images in InDesign** section of the **STEP'n'design** documentation.

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.

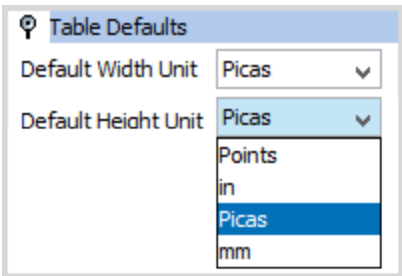
This setting is to maintain backwards compatibility with prior versions of STEP.

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.

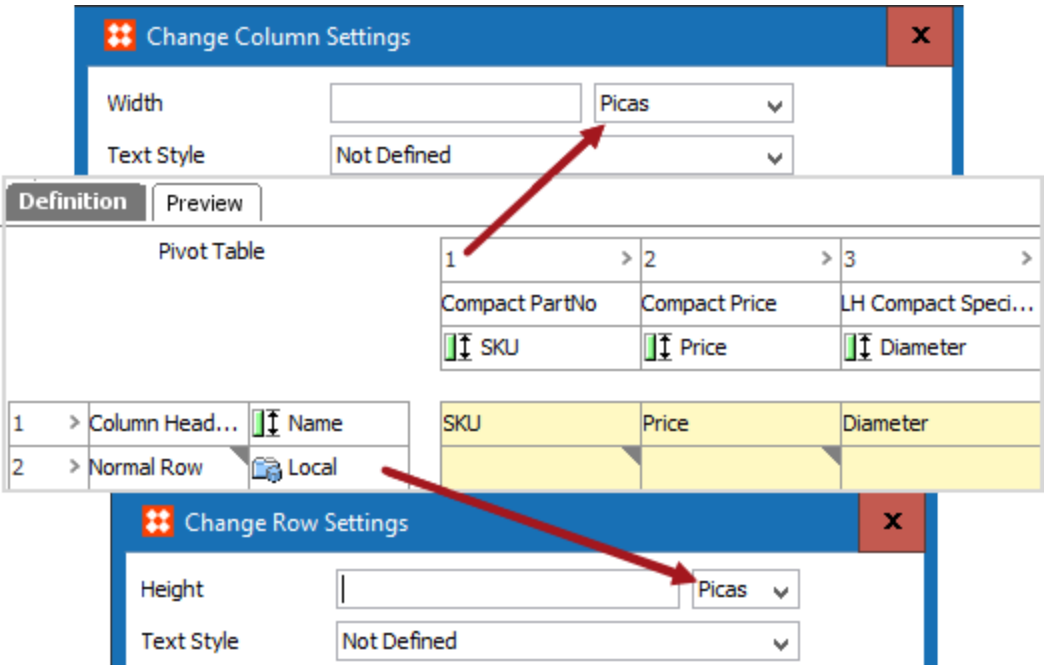
This is the only type of inheritance that is done from a product node 'downwards' to the product overrides below it.

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.



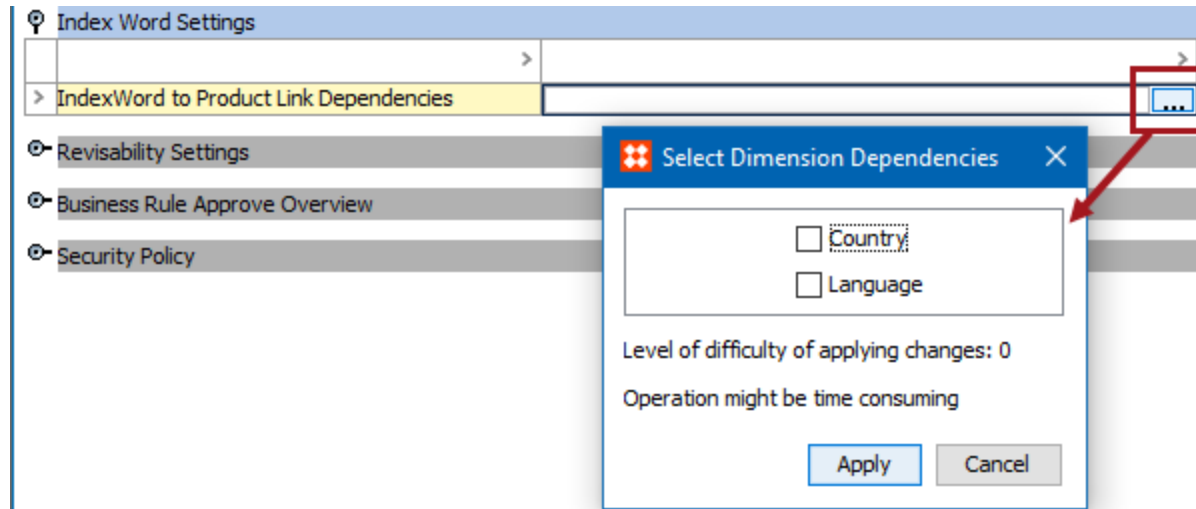
Once the default units have been set, the units will be the default units in the table editor as shown in the image below. If no default unit is selected, the default units will be blank and must be manually selected every time.



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. Making the links between products and index words dimension dependent can be useful for a situation where not all products are sold in all countries.



For more information on index words, see the **Creating Document Indexes** section of the **STEP'n'design** documentation.

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 **Business Rules on Approval** 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

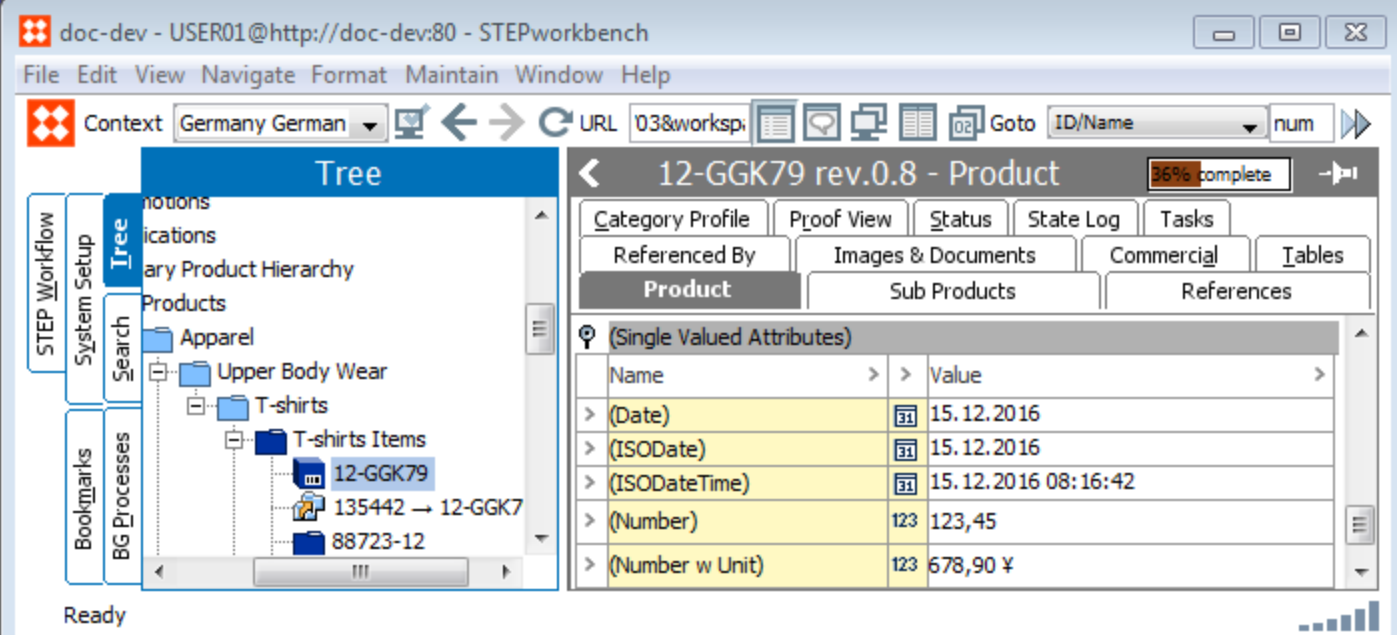
GUI Localization is maintained in System Setup > **Users & Groups** > **Users** on the **System Settings** tab.

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

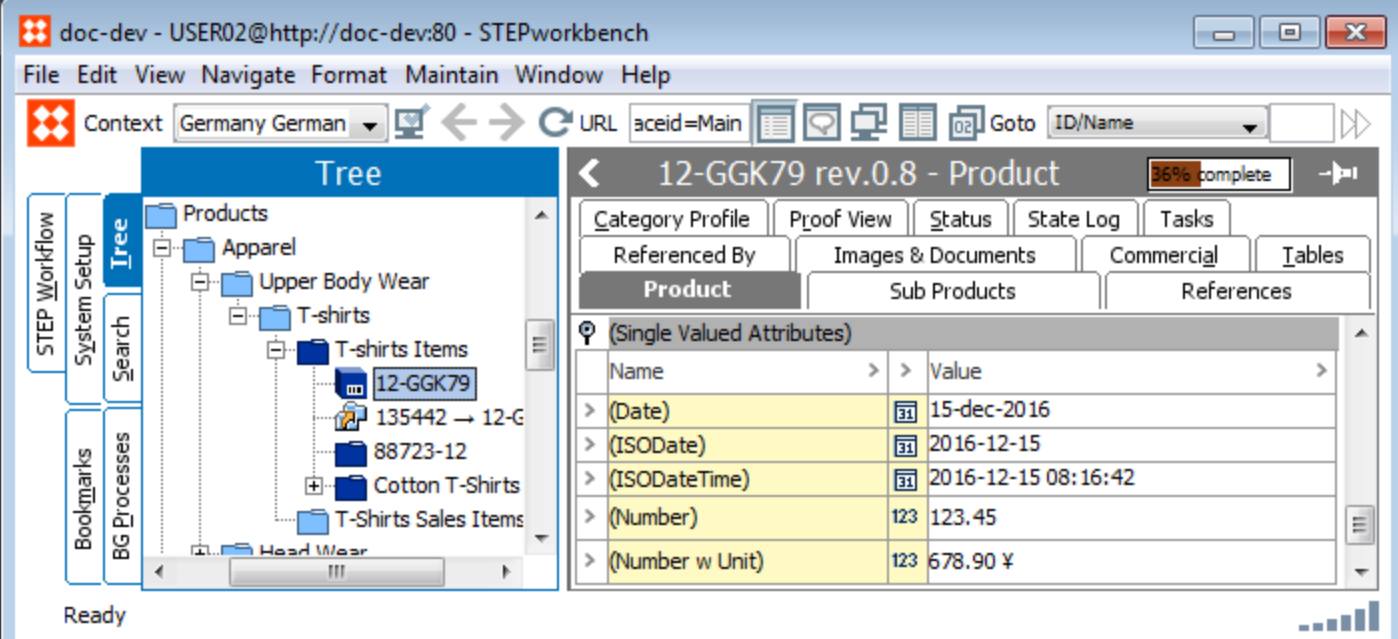


The screenshot shows the STEPworkbench interface. The top menu bar includes File, Edit, View, Navigate, Format, Maintain, Window, and Help. The context is set to 'Germany German'. The main window displays '12-GGK79 rev.0.8 - Product' with a progress bar at 96% complete. The 'Single Valued Attributes' table shows values formatted for the German locale, such as '15.12.2016' for dates and '123,45' for numbers.

Name	Value
(Date)	15.12.2016
(ISODate)	15.12.2016
(ISODateTime)	15.12.2016 08:16:42
(Number)	123,45
(Number w Unit)	123,678,90 €

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



doc-dev - USER02@http://doc-dev:80 - STEPworkbench

File Edit View Navigate Format Maintain Window Help

Context: Germany German URL: aceid=Main Goto: ID/Name

Tree

- Products
 - Apparel
 - Upper Body Wear
 - T-shirts
 - T-shirts Items
 - 12-GGK79
 - 135442 → 12-G
 - 88723-12
 - Cotton T-Shirts
 - T-Shirts Sales Items

12-GGK79 rev.0.8 - Product 36% complete

Category Profile Proof View Status State Log Tasks

Referenced By Images & Documents Commercial Tables

Product Sub Products References

(Single Valued Attributes)

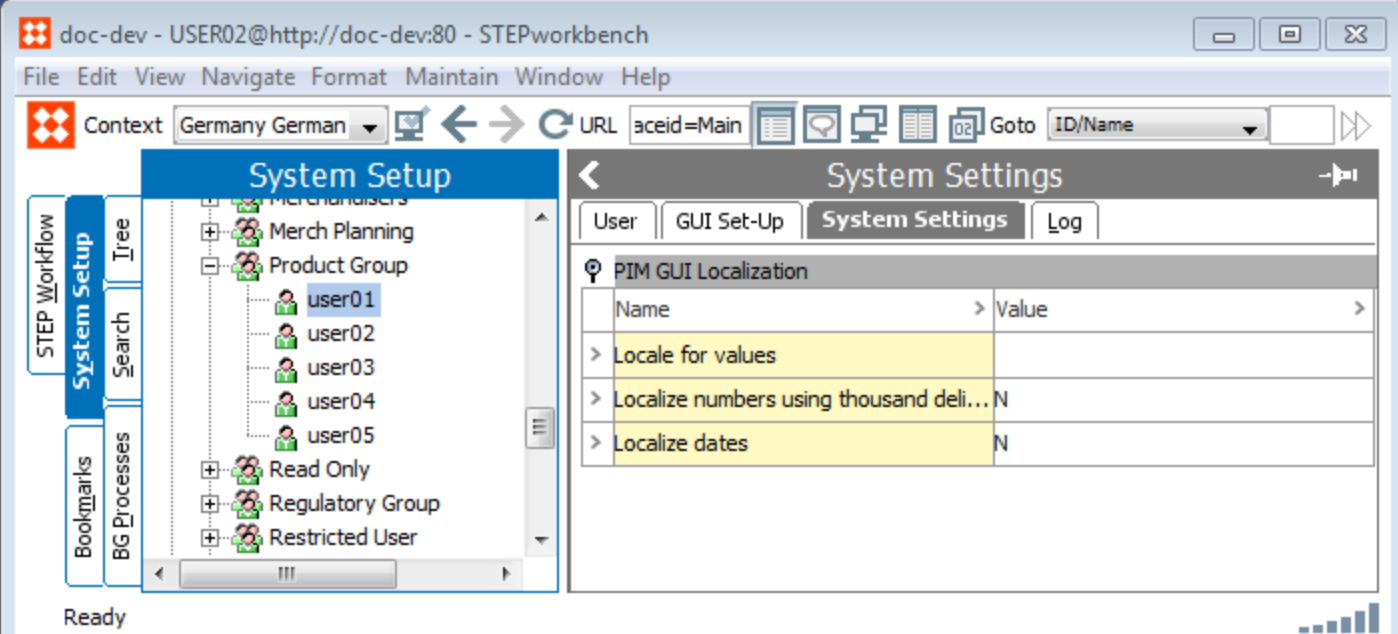
Name	Value
> (Date)	15-dec-2016
> (ISODate)	2016-12-15
> (ISODateTime)	2016-12-15 08:16:42
> (Number)	123 123.45
> (Number w Unit)	123 678.90 ¥

Ready

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.



doc-dev - USER02@http://doc-dev:80 - STEPworkbench

File Edit View Navigate Format Maintain Window Help

Context: Germany German URL: aceid=Main Goto: ID/Name

System Setup

- Merchandisers
- Product Group
 - user01
 - user02
 - user03
 - user04
 - user05
 - Read Only
 - Regulatory Group
 - Restricted User

System Settings

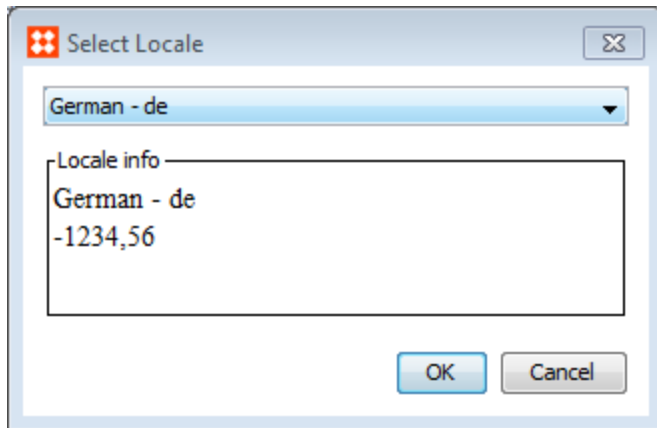
User GUI Set-Up System Settings Log

PIM GUI Localization

Name	Value
> Locale for values	
> Localize numbers using thousand deli...	N
> Localize dates	N

Ready

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.

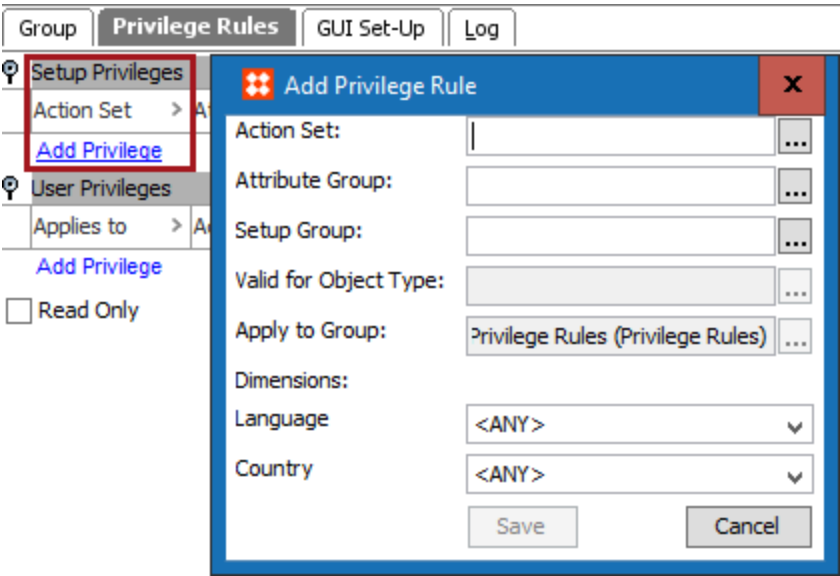
4. In **Localize dates**, click Y to localize dates in the Workbench. In this example a German -de locale is selected for User 01. For User 01, isodates will be shown as 13.05.2009 and isodatetime will be shown as 13.05.2009 19:10:48.

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.

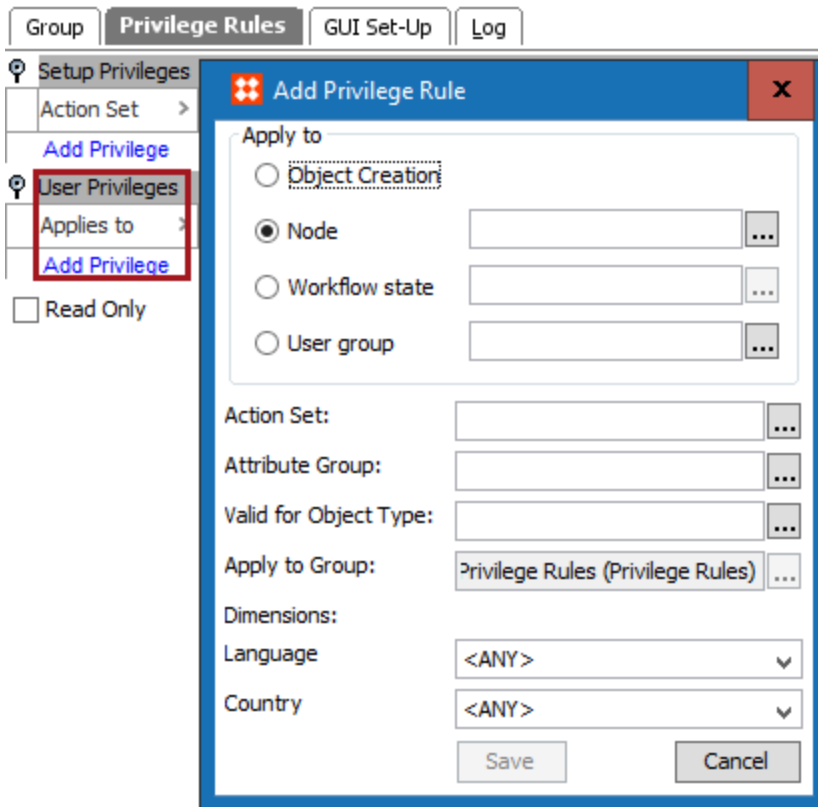
Privilege Rule Types

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

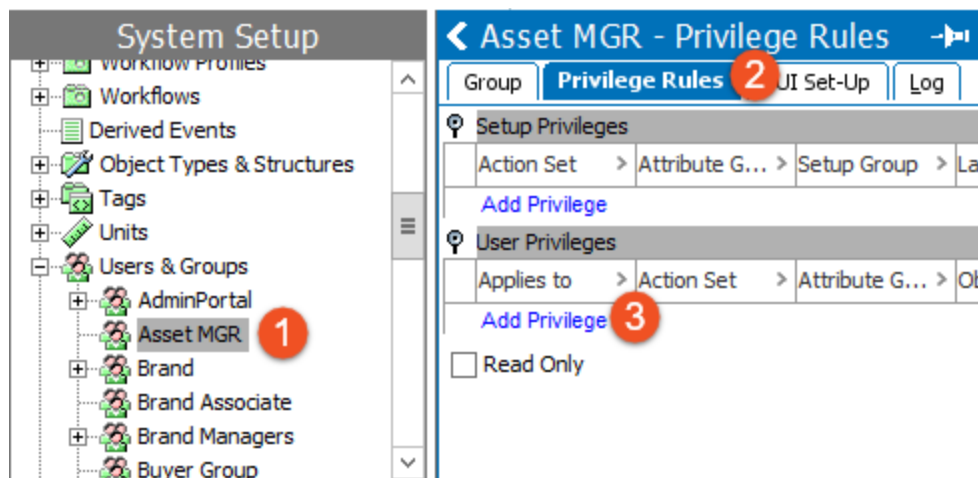
If a Group has permission to work with a specific Classification node, Attribute Values can be edited for all Products linked to (or below) that node. However, access to the Classification node does not grant the privilege to link new Products in the Product hierarchy. To do that, the Group must have 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



Apply to

☐ Object Creation

☒ Node

☐ Workflow state

☐ User group

Action Set:

Attribute Group:

Valid for Object Type:

Apply to Group:

Asset MGR (Asset_MGR)

Dimensions:

Language

<ANY>

Country

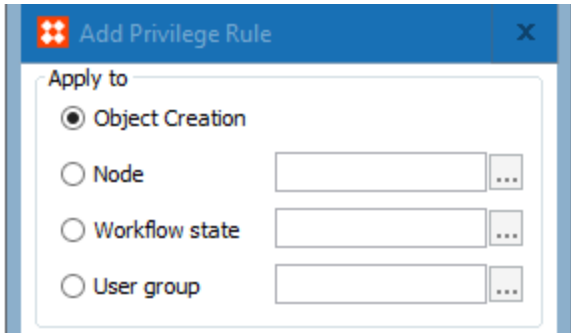
<ANY>

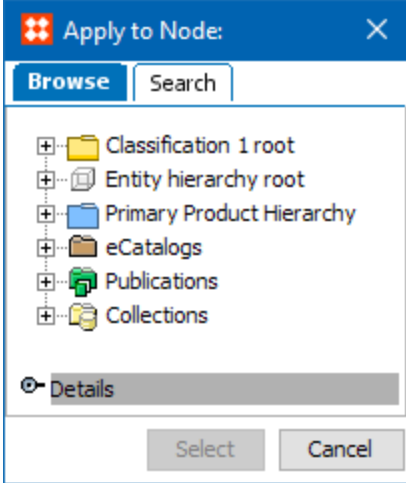
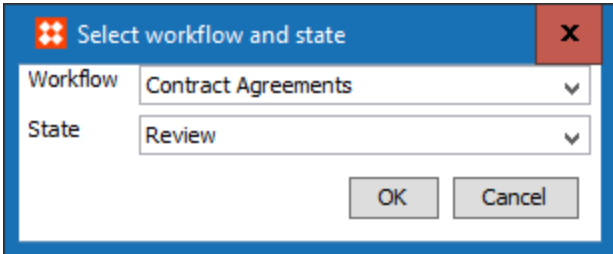
Save

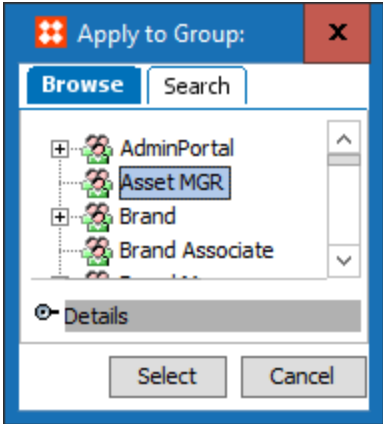
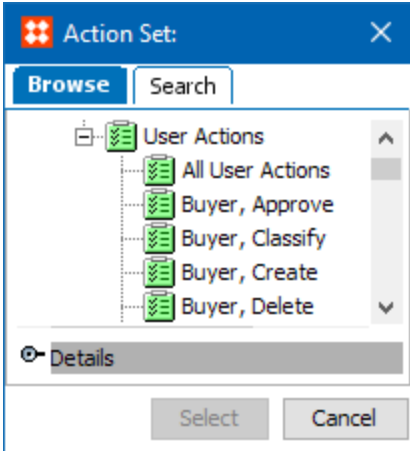
Cancel

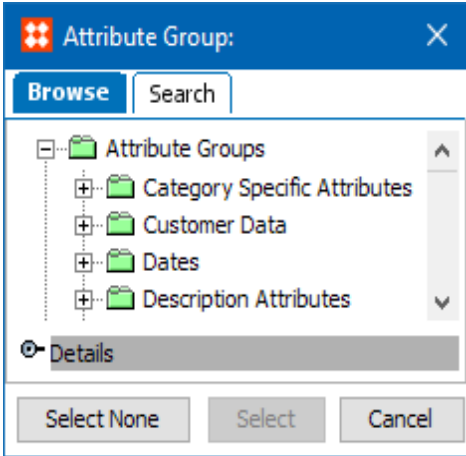
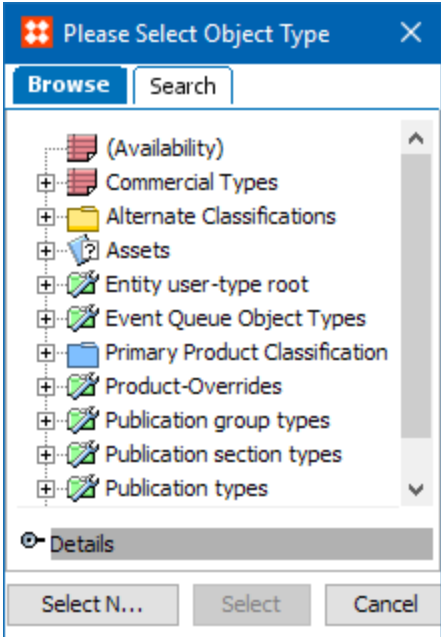
5. Make the relevant changes, and then click **Save**.

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 Action Sets in System Setup / Super User Guide. 
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. 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 / 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 / Super User Guide documentation. Dimensions: Language <ANY> Country <ANY> Save Cancel

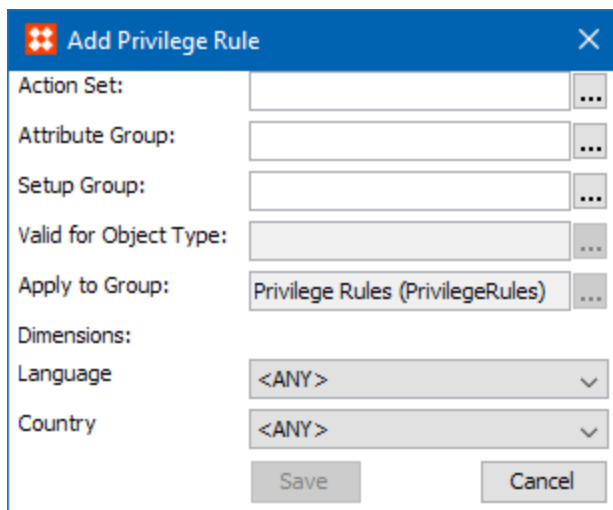
Adding Setup Privileges for a Group

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**.

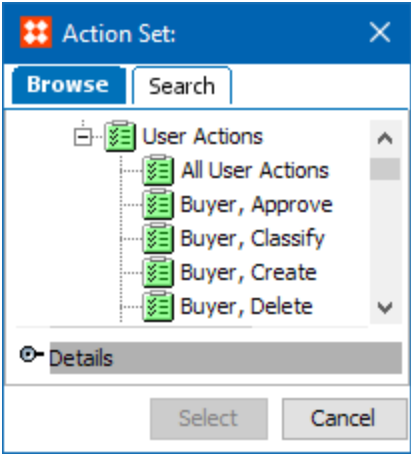
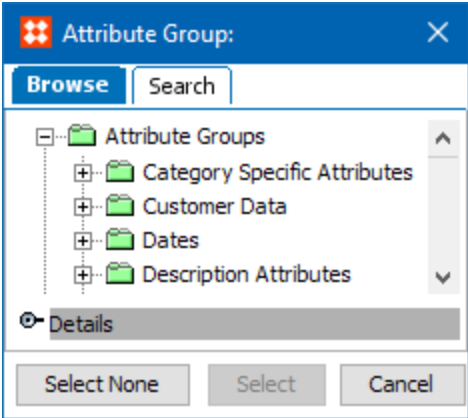
An **Add Privilege Rule** dialog box appears.

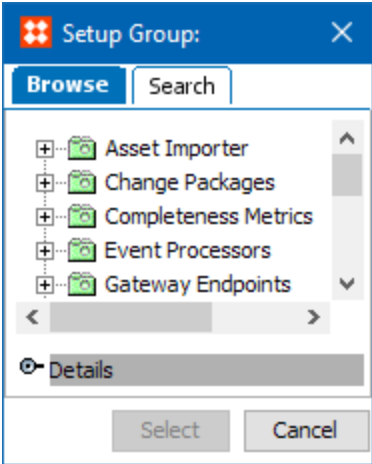


4. Make the relevant changes, and then click **Save**.

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 / Super User Guide** 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
- Merchandise 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>

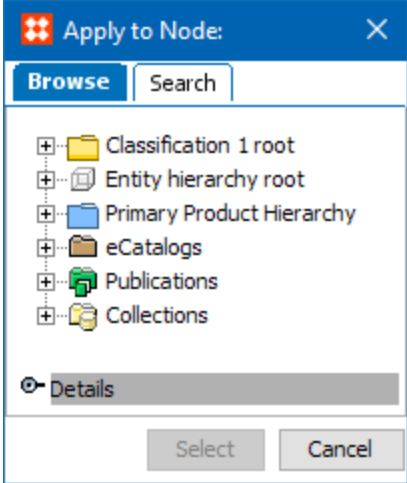
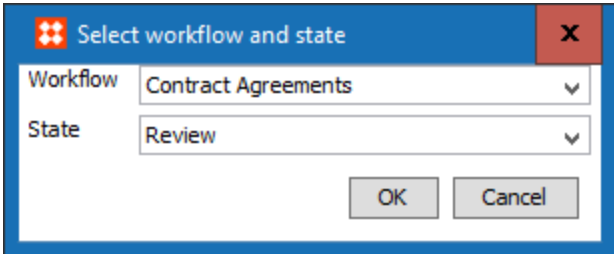
Save

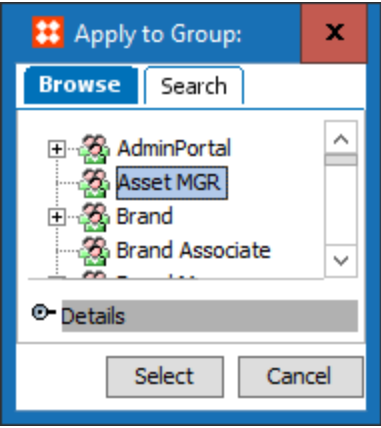
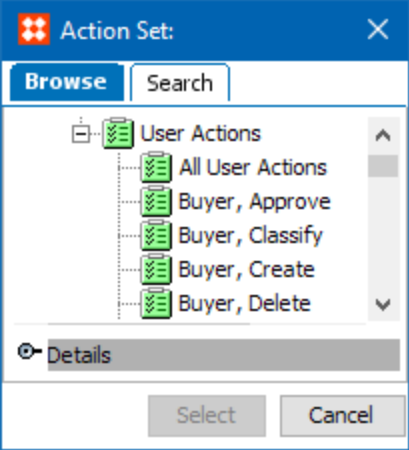
Cancel

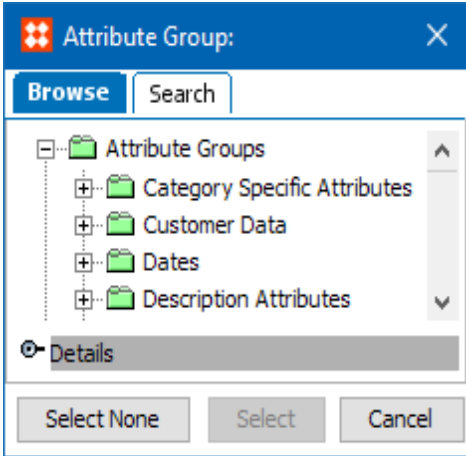
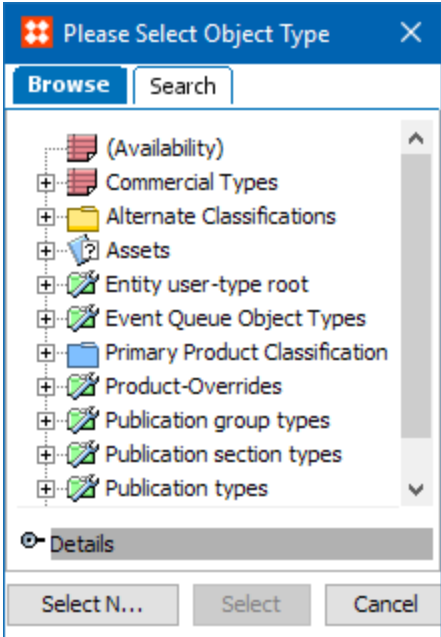
4. Make the relevant changes, and then click **Save**.

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. 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 / Super User Guide documentation. Dimensions: Language <ANY> ▼ Country <ANY> ▼ Save Cancel

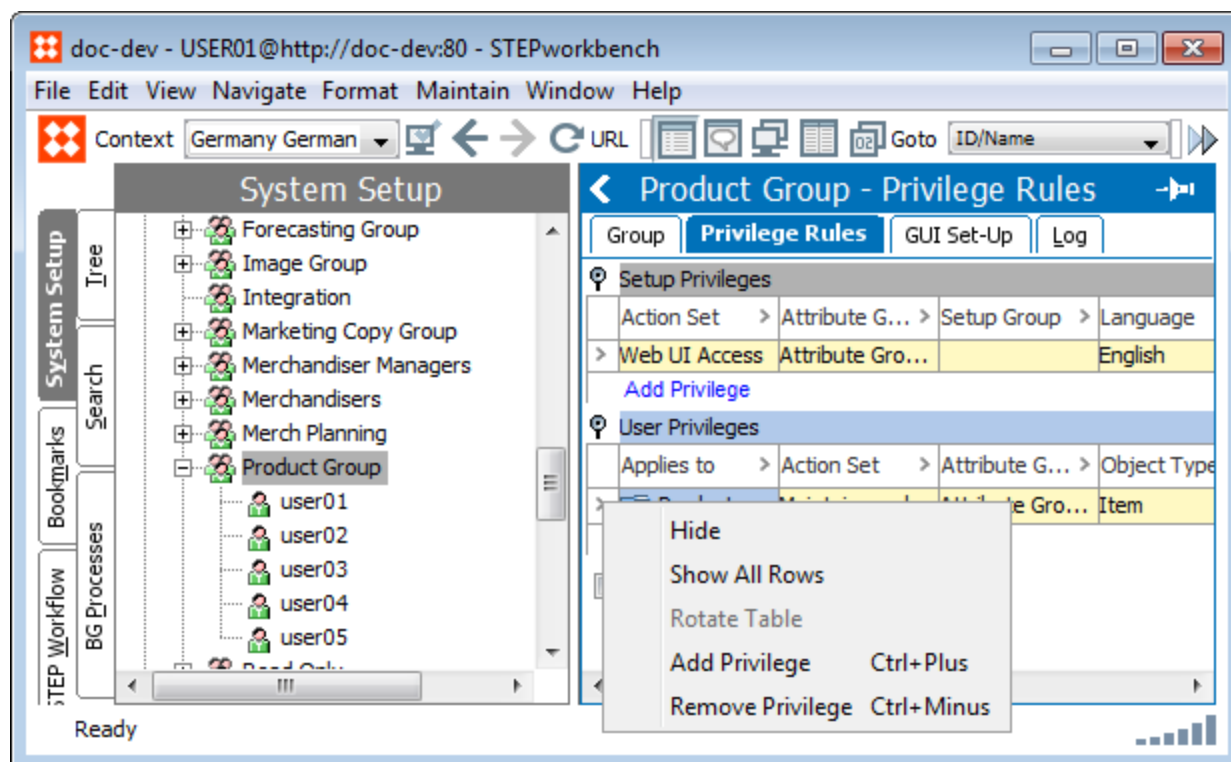
Removing a Privilege Rule from a Group

1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group.

The Group appears in the **Group Editor**.

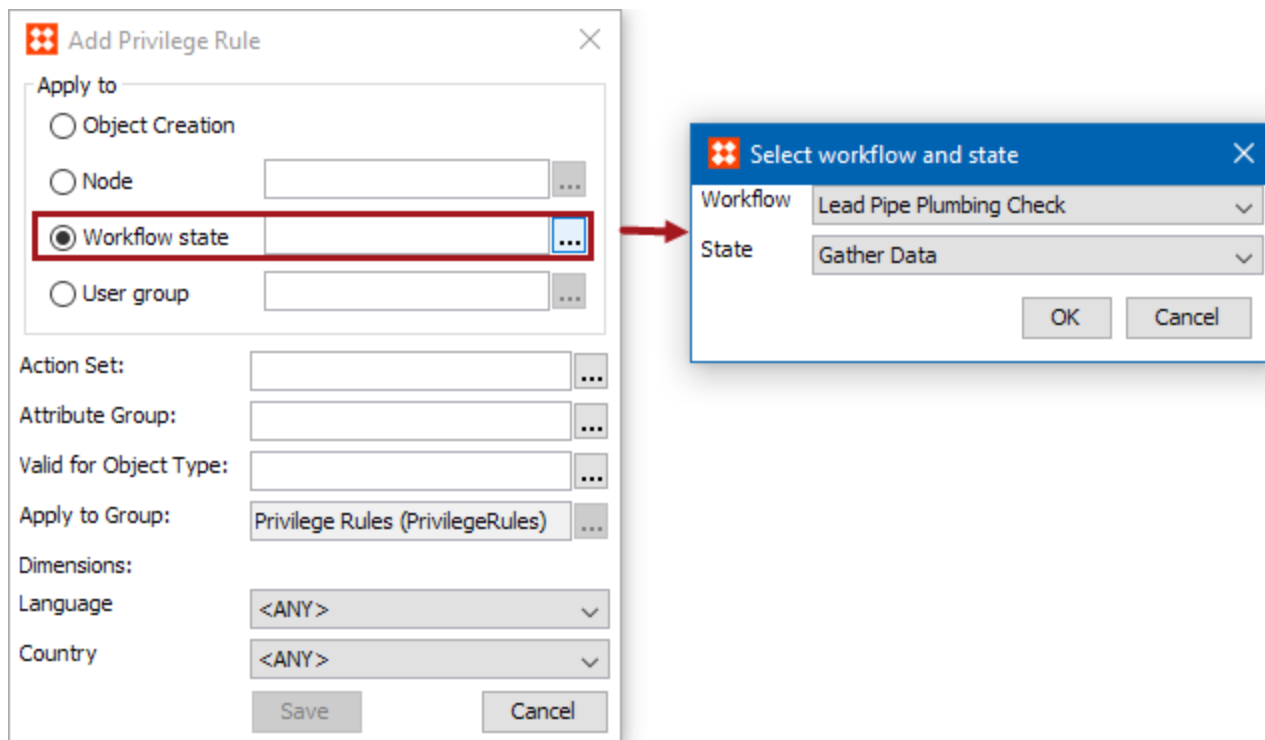
2. Click the **Privilege Rules** tab.
3. Click the row header of the Privilege rule to be removed, and then click **Remove Privilege**.

The Privilege rule will now be removed from the list and will no longer apply to the members of the selected Group.



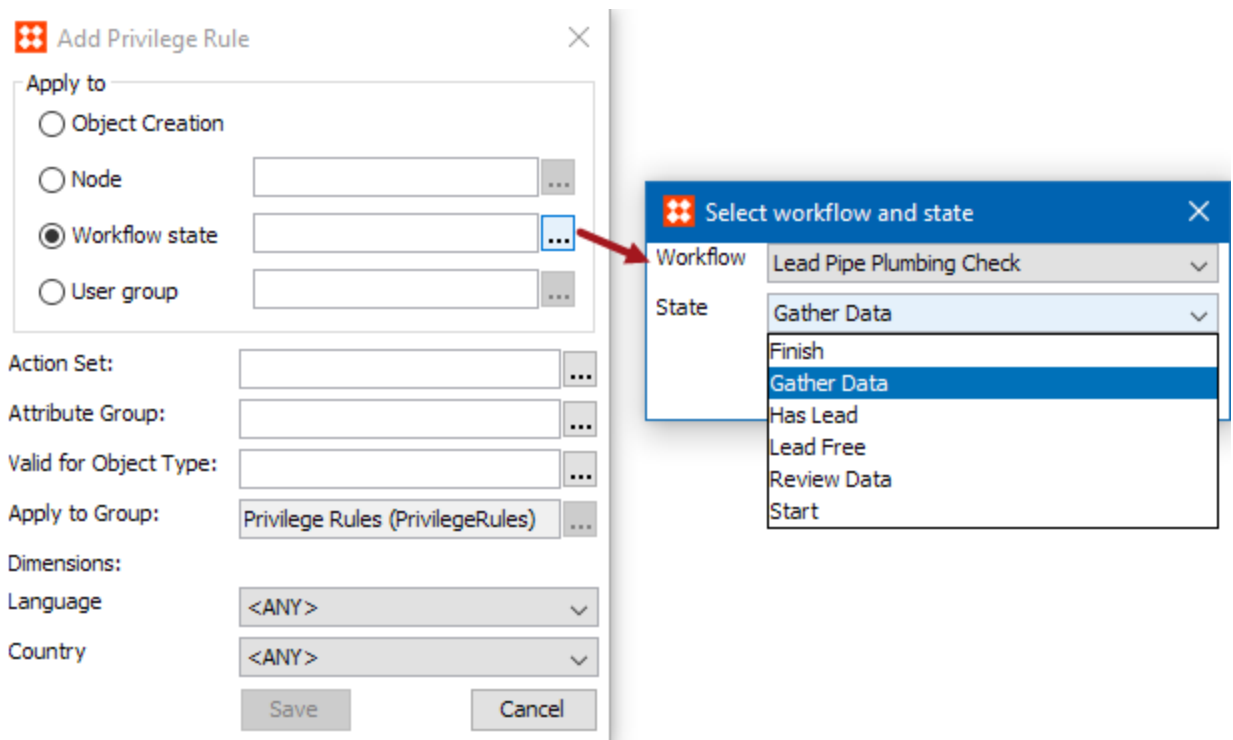
Configuring Workflow Privileges

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.

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

File Edit View Navigate Format Maintain Window Help

Context: English US

URL: /Editor&contextid=Context1&id=SalesItemCreationNewBR&workspaceid=Main

Goto: ID/Name

System Setup

Sales Item Creation rev.0.17 - STEP Workflow Editor

STEP Workflow Editor | Validity | Log | Status

Name	Value
ID	SalesItemCreationNewBR
Name	Sales Item Creation
Revision	0.17 Last edited by USER on Mon Oct 26 17:39:05 EDT 2015
Initiate automatically	☐
Listen on updates from im...	☐
Use executing user's write...	☒
Disable transitions on term...	☐
Start Condition	

Used Local Business Rules

Used Global Business Rules

Title	State	STEP Workflow Compon...	Description
Workflow Privileges			
Applies to	Action Set	Attribute Group	Object Type
	Group	Language	Country
Digital_Asset_Review	Images, View		Image Group
			<ANY>
			<ANY>

Add Privilege

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.

Edit Privilege Rule

Workflow state: Digital Asset Review

Action Set: Images, View (ImageView)

Attribute Group:

Valid for Object Type:

Apply to Group: Image Group (ImageGroup)

Dimensions:

Language: <ANY>

Country: <ANY>

Save Cancel

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 / 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)

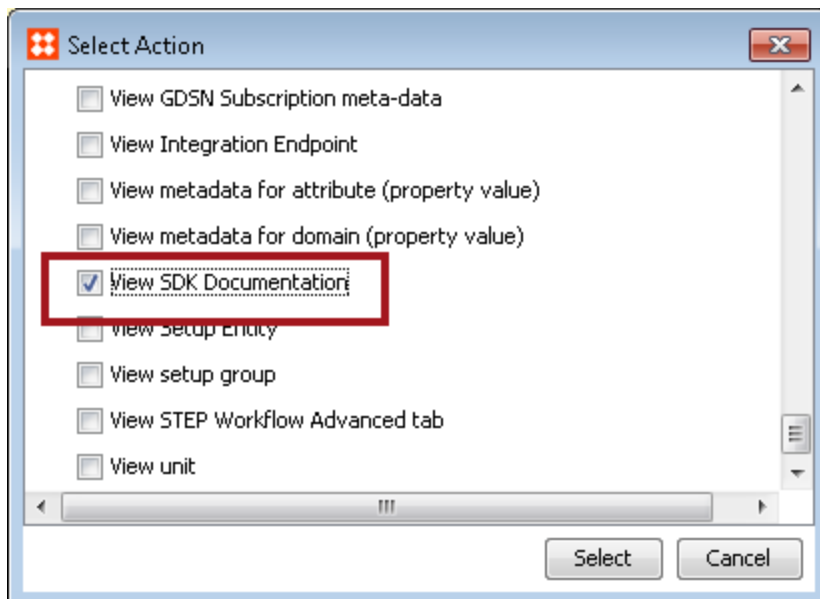
Granting Access to the SDK and API Documentation

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 or by clicking the STEP API Documentation button on the STEP WebStart page. 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.

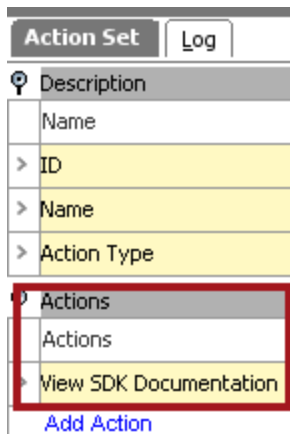
Assigning the privilege is a two-step process: First, you create an action set that includes the **View SDK Documentation** setup action privilege, and then you assign the action set to the relevant groups.

Grant Access to the SDK and API documentation

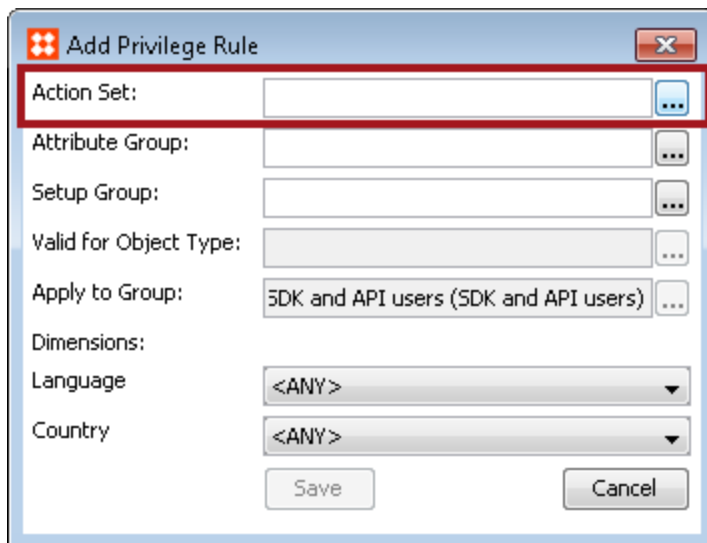
1. In **System Setup**, expand **Action Sets**, right-click **Setup Actions**, and then click **New Action Set**.
2. Enter an **ID** and a **Name** for the action set, and then click **OK**.
3. On the **Action Set** tab, click **Add Action**.
4. In the **Select Action** dialog, scroll down and select **View SDK Documentation**, and then click **Select**.



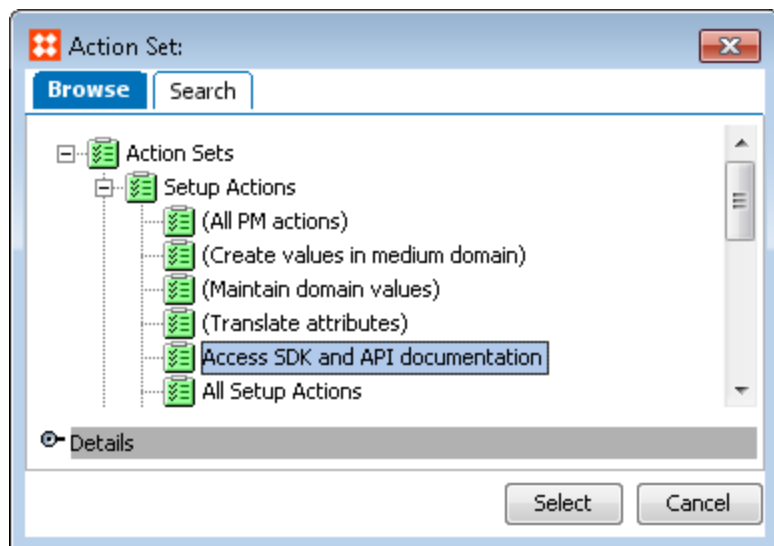
The action is now listed in the **Actions** area on the **Action Set** tab, and you can add the action set to the relevant user group.



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

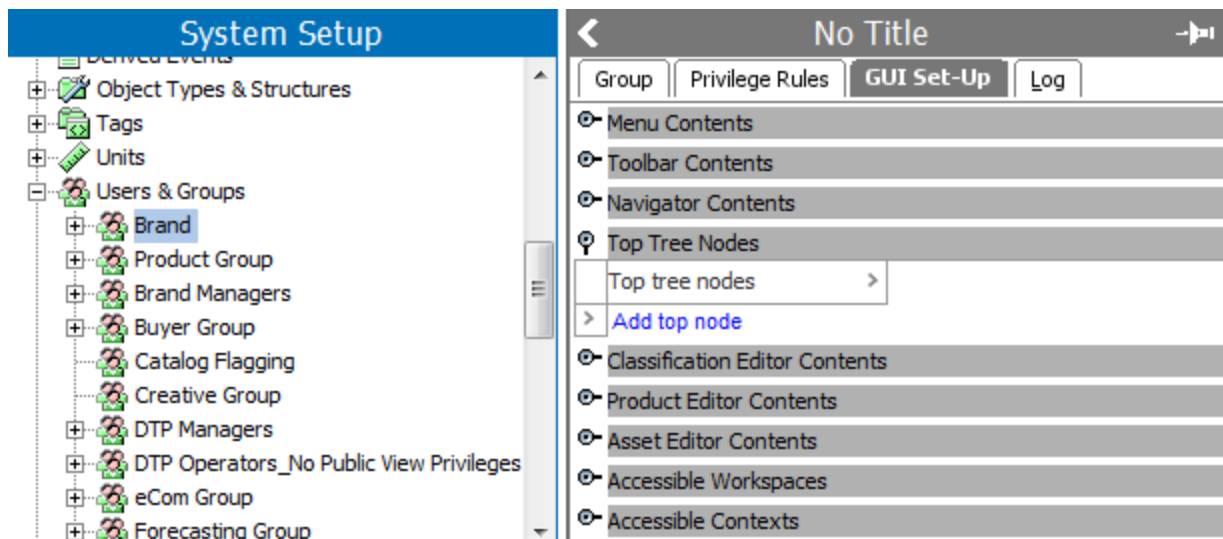
GUI Set-Up is maintained in [System Setup]: **Users & Groups > Groups or Users**

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

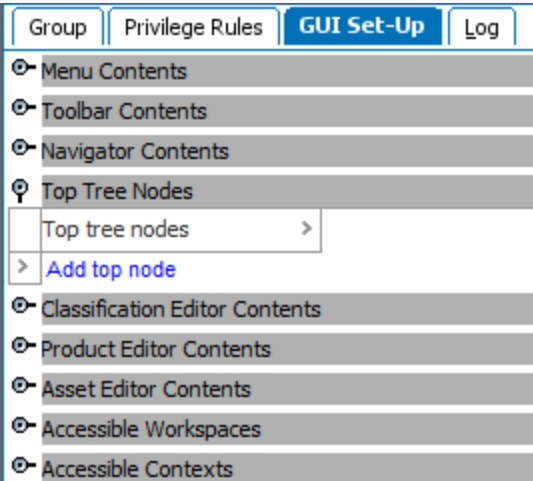
The GUI settings at the user group level applied to all the users that get created under it.

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.

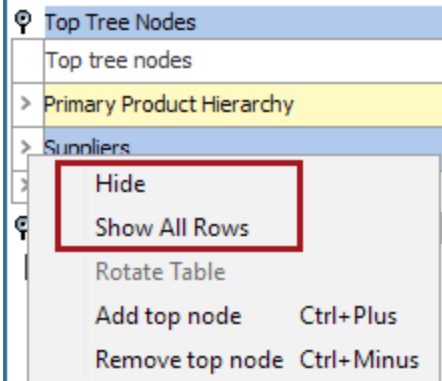
- 1. In [System Setup], unfold **Users & Groups**, and then click the relevant Group.
A **Group** Editor appears.

- 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**.

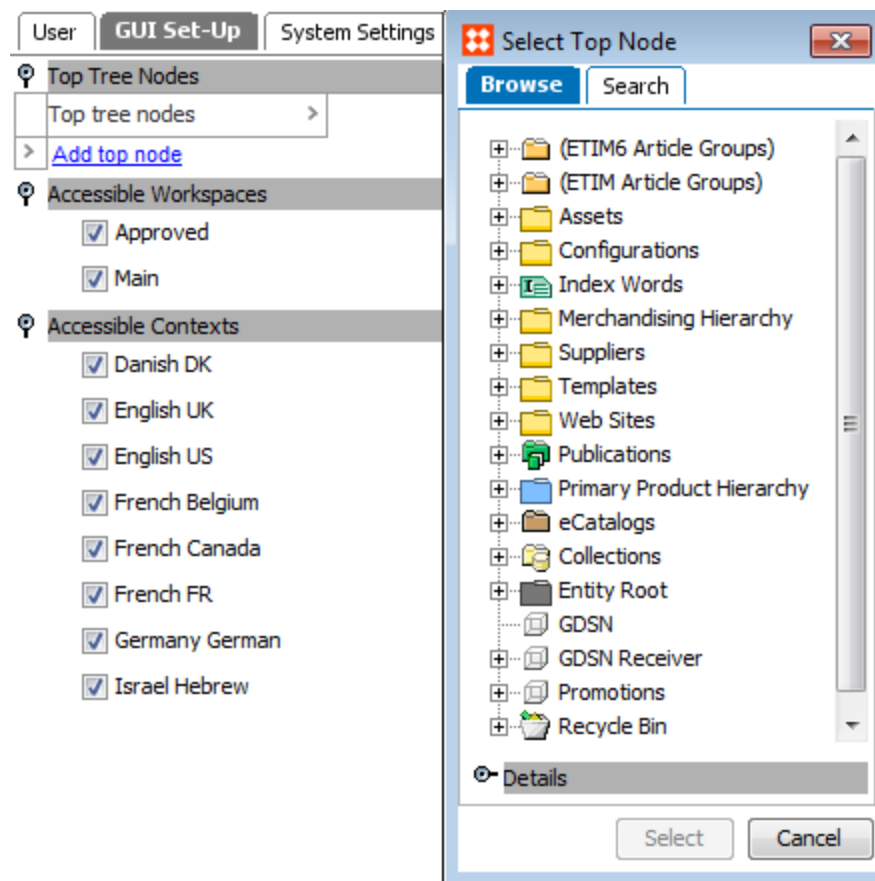
GUI Set-Up for a User makes it possible to customize top tree nodes for the user and provides an administrator with a way to make relevant workspaces and contexts accessible.

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

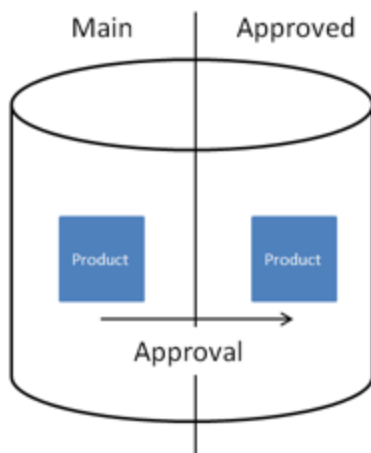
A workspace can be perceived as a set of independent 'copies' of workspace revisable database objects. A workspace is thus an area in the database that provides a snapshot of all objects in the database. Objects that are subject to approval can be maintained within one workspace without affecting the snapshot in other workspaces.

The data in STEP is logically divided in to 'workspaces'. Standard STEP installations come with two workspaces which are named Main and Approved. The Main workspace represents unapproved revisions of the data and the Approved workspace represents approved revisions of the same data.

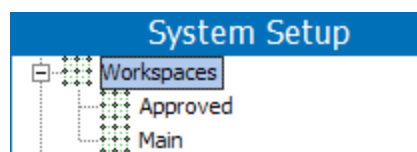
Hence, the Main workspace serves as an area for acquiring and maintaining data and the relation between data. Importing, enriching and categorizing data is done in the Main Workspace. The Approved workspace stores data revisions that are deemed fit for production purposes - data that is ready to be published / exported to third party systems.

The transition of data revisions from the Main workspace to the Approved workspace is referred to as Approval. Approvals can be performed manually or programmatically.

Data revisions that are approved will be marked as 'Approved' in the Main workspace. If approved data changes in the Main workspace e.g. is edited by a user, it becomes unapproved and must be re-approved to have the latest data made available in the Approved workspace. Hence, validation of data prior to approval becomes of main concern.



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.
- 'Externally Maintained' attributes do not need approval. The values get populated in the Approved workspace as soon as the value is added / edited in the Main workspace.

Maintaining Workspaces

The following information is about how to properly maintain workspaces created outside of the Main and Approved workspaces. All workspaces are maintained in System Setup under the Workspaces node.

Creating Workspaces

When creating a new workspace, the contents of the selected workspace will be copied to the new one.

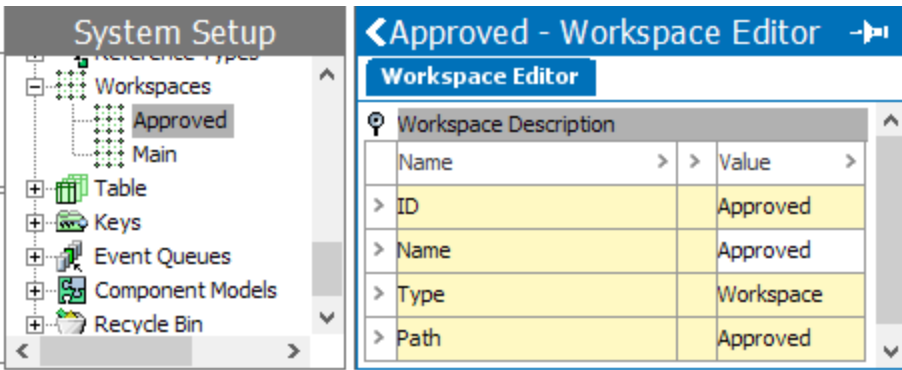
Note: Typically, new workspaces should not be created as the initially set up workspaces (Main and Approved) will be sufficient for the cycle of data maintenance. Sometimes unexpected issues can occur with more than the Main and Approved workspace created.

For more information, see the **Workspaces** documentation.

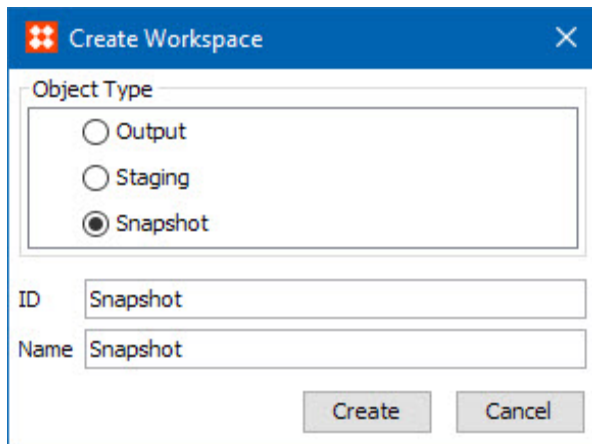
1. In System Setup, click on **Workspaces**, and then click the workspace of which you need to create another workspace.

In this example, a Snapshot workspace is created.

Note: A snapshot will typically be made of the **Approved** Workspace.



2. In the dialog that appears, select the Object Type, in this case it is 'Snapshot', and enter in an ID and Name. Click Create.



Create Workspace

Object Type

☐ Output

☐ Staging

☒ Snapshot

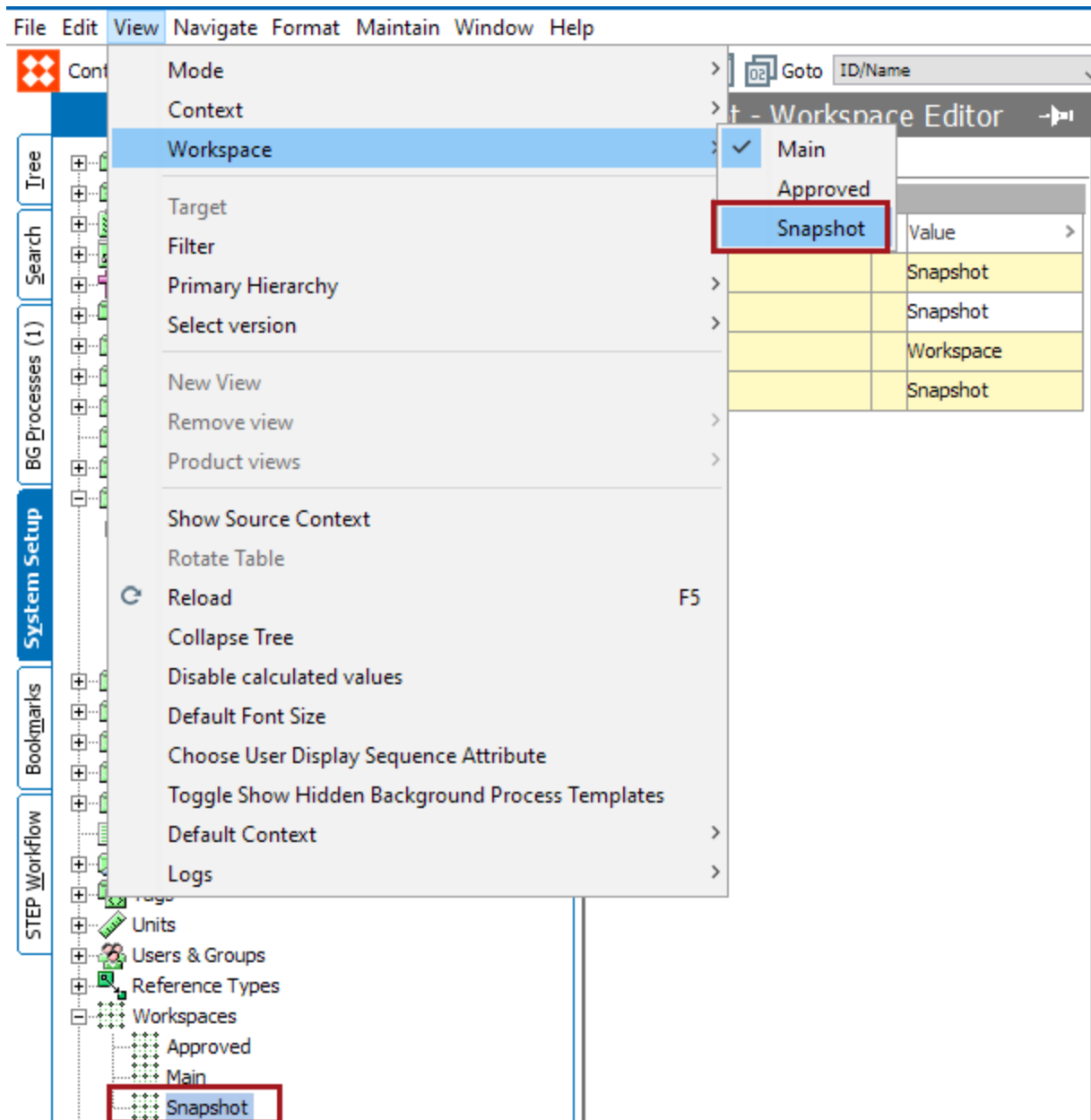
ID: Snapshot

Name: Snapshot

Create Cancel

Note: When clicking either **Outputs** or **Snapshot**, a terminal (uneditable) workspace will be created.

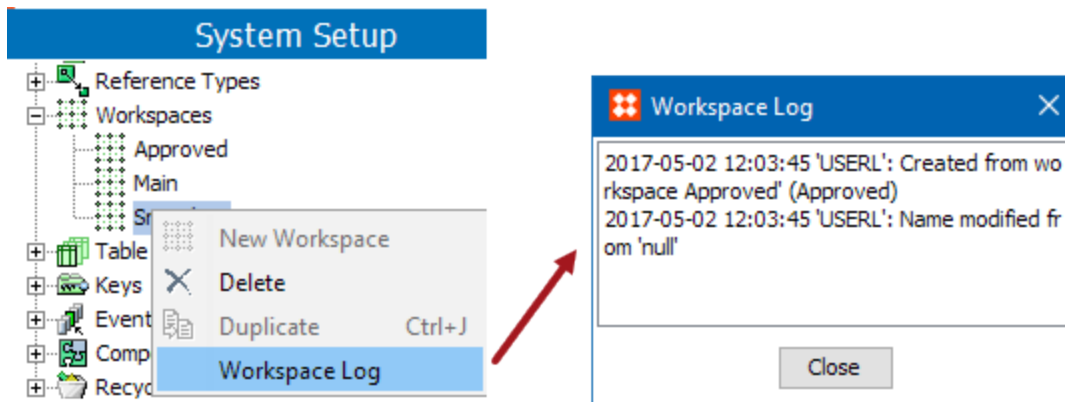
3. A new workspace appears in the Workspace list in System Setup, and the Workspace selector in the workbench by going to View > Workspace > Snapshot.



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.' All data in 'Snapshot' is read-only and cannot be edited.

Workspace Log

To view the activity and changes that has taken place in a workspace, go to System Setup and right click on the workspace to view the logs for the particular workspace selected.

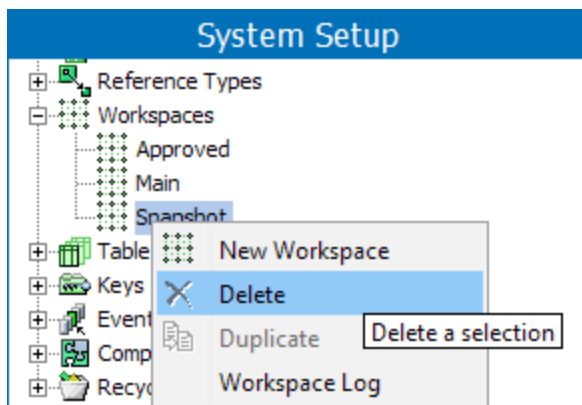


Deleting Workspaces

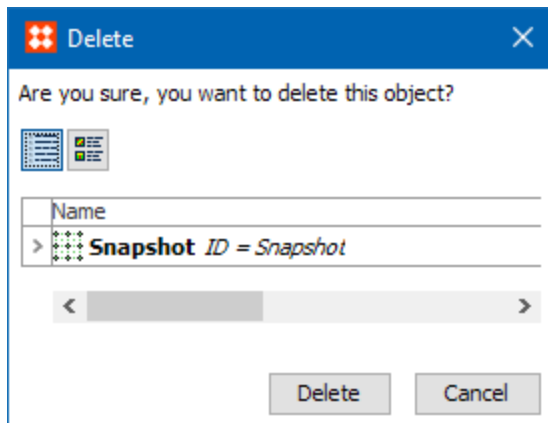
If a workspace is no longer needed, it can be deleted. Before deleting a workspace though, make sure to transfer data that should be kept to another workspace before deleting. The initially set up Workspaces cannot be deleted. Deleting a workspace is very time consuming.

Note: The Workspace for a current Object is selected from the **Workspaces** list (upper right corner).

1. In System Setup, unfold **Workspaces**, and then click the workspace to be deleted.
2. Right-click, and select **Delete**.



3. A **Delete Warning** dialog box appears. Click **Delete** to delete the workspace.



Note: It is not possible to delete the Main Workspace and the Approved workspace.
