STIBO SYSTEMS
MASTER DATA MANAGEMENT



SUPER USER GUIDE

System Setup

Rel. 10.3-MP4 (March 29, 2022)

Table of Contents

Table of Contents **2**

System Setup **22**

Prerequisites 22

Accessing Shortcut Menus on PC and Mac 22

Action Sets **23**

Setup Actions **25**

Action Sets 25

Administration 25

Assets 26

Attributes 26

Background Processes 28

Bulk Updates 29

Business Modules 29

Business Rules 29

Change Packages 29

Contexts 29

Cryptographic Keys 30

Data Quality 30

Deduplication 30

Events 30

GDSN Data Pools 31

Integration Endpoints 33

Link Types 33

List of Values (LOVs) 33

Object Types 35

Print Publication 36

Purging 36

Reports 36

SDK / API Documentation 37

Tables 37

Tags 37

Units 37

Users & Groups and Privilege Rules 38

Value Search 38

Web UI 39

Workflows 39

Workspaces 41

User Actions **42**

Assets 42

Collections 44

Commercial Lists 45

Data Preparation Functionality 46

Data Quality Functionality 46

eCatalog 46

GDSN Data Pools 47

Products, Classifications, and Entities 47

Publications 55

Translations 58

Web UI 59

Maintaining Action Sets **60**

Creating an Action Set 60

Adding Actions to an Action Set 61

Removing an Action from Action Set 63
Action Set Log Tab 63
Attributes 65
Attribute Groups 66
Attribute Group Display Sequence 67
Manually Sorted 67
Creating a Customized View 69
Creating an Attribute Group 73
Attribute Links 74
Linking Orphan Attributes 76
Linking an Orphan Attribute 76
Linking Specification Attributes 79
Linking Attributes in Product editor 79
Linking Attributes to Products or Classifications 80
Linking to a Source from Attribute 82
Sorting Linked Attributes 84
Setting a Display Sequence Number on the Attribute 84
Setting a Display Sequence Number under Linked Attributes 85
Setting the Priority of a Product Variant Attribute 87
Setting the Attribute Display Sequence on Publication (Green) Hierarchy Objects 88
Attribute Transformations 89
Creating an Attribute Transformation 91
Create an Attribute Transformation Group 91
Create an Attribute Transformation 92
Configuring an Attribute Transformation 93
Attribute Types 98
Calculated Attributes 100
Calculated Attribute Considerations and Limitations 102
Deciding to Use a Calculated Attribute 103
Setting up a Calculated Attribute 105
Using Variables in a Calculated Attribute 108
Optimizing Calculations Using STEPXML 112
Limit Performance Penalties for Calculations 113
Calculated Attribute Use Cases 115
Introduction to Value Templates 116
Attribute Examples 119
Commercial Examples 122
Conditional Examples 124
Date Examples 128
List Examples 130
Number Examples 135
Other Examples 139
Classification Functions 139
Completeness Functions 140
Data Container Functions 140
Hierarchy Functions 141
Lookup Table Functions 145
Object Functions 146
Override Functions 147
Reference Functions 148
Revision Functions 149

Workflow Functions 150

Publishing Examples 152

Prerequisites 152

Sample Formulas and Values 153

Text Examples 160

Escape Functions 160

Identify Functions 160

Modify Functions 161

Regular Expression Functions 165

Representative Text Functions 166

Test Functions 167

Variable Examples 169

Concatenate Function with Variables 170

Find Function with Variables 171

Asset Push File Name Scenario 173

Solution 174

Classification Scenario 176

Get All Classifications 176

Get Classifications Based on Type 176

Get Metadata Value Located on the Link From Classification-to-Product 177

Get CPL Classifications Using Classification Product Links 178

Commercial Data Scenario 179

Price List in a Publication Structure 179

Price List in an eCatalog Structure 179

Comparing Localized Values Scenario 180

Solution 180

Identifying Child Objects Scenario 182

Solution 182

Including Parent Info on Child Scenario 184

Solution Option 1 184

Solution Option 2 185

New Product Indicator Scenario 187

Solution Options 187

Override Scenario 190

Get Overridden By Products 190

Get Overridden Product Objects 191

Get Product Override Parents 192

Product Description Scenario 193

Solution Option 1 194

Solution Option 2 196

Reference Scenario 197

Get List of References 197

Get Value From a Referenced Product 198

Get Metadata Value Located on the Link From Product-to-Product 199

Get Referenced By Using Reference Owner 200

Get Reference Type ID 201

Get Reference Source or Target Product Information on the Link 203

Total Cost of Materials Scenario 205

Solution 206

Units Scenario 208

Creating a Calculated Attribute 211

Overriding and Restoring Calculated 217

Attribute Values .....
Override a Calculated Attribute 217
Restrict Overrides of a Calculated Attribute Value 218
Remove Override Values 219
Restore a Calculated Value 220
Conditional Attribute Display **221**
Terminology 221
Creating the Conditional Attribute 221
Activating a Conditional Attribute 222
Configuring Conditionally Valid Attributes .223
Description Attributes 223
Specification Attributes 225
Writing Conditional Expressions 226
Conditional Attribute Example 227
Data Containers **231**
Composite Attribute Concept 231
Using Data Containers 232
Setting Up Data Container Types in Workbench **235**
Prerequisites 235
Adding Data Container Types 236
Configuring Data Container Types 237
Adding references to a data container type 241
Deleting Data Container Types 243
Data Container Keys 244
Data Container Keys **245**
Creating a Data Container Key in the Workbench 245
Check Duplicate Keys 248
Removing Duplicate Keys in Collections ...250
Adding and Maintaining Data Container Instances **253**
Edit a data container reference 255
Description Attributes **258**
Metadata Attributes 258
Linking Description Attributes to an Object Type 259
Attribute Metadata on Attributes **262**
Create a Metadata Attribute on an Attribute262
Description Attributes and Object Types 265
Validity on Description Attributes **267**
Dimension Dependent Attributes **270**
Dimension Dependent Attribute Values 270
Non-Dimension Dependent Attribute Values 271
Setting Dimension Dependency on Attributes 272
Value Inheritance in Dimension Dependent Attributes 274
Showing Source Context (Parent Dimensions) 276
Dimension Dependent Attribute and Attribute Group Recommended Practices **279**
Dimension Dependent Attribute Do's 279
Dimension Dependent Attribute Don'ts 280
Attributes That Need Dimension Dependencies 280

Dimension Dependent Attribute Group Do's 281

Dimension Dependent Attribute Group Don'ts 281

Using Dimension Dependent LOVs With Attributes 282

Attribute With No Dependency, LOV With a Dependency 282

Attribute With No Dependency, LOV Without a Dependency 283

Attribute With a Dependency, LOV With a Dependency 283

Attribute With a Dependency, LOV Without a Dependency 283

Display Sequence Attribute 284

Display Sequence Attribute Setup 284

Sorting Attributes 286

Selecting a Personal Display Sequence Attribute 288

Display Sequence Attribute and the Product Editor Display 289

Display Sequence Attribute and Classifications 290

Setting the Priority of a Product Variant Attribute 291

Display Sequence Attribute and Tables 291

Setting the Attribute Display Sequence on Publication (Green) Hierarchy Objects 292

Encrypted Attributes 294

Cryptographic Key 294

Password requirements 296

Creating Encrypted Attributes 296

Properties of attributes that cannot be used with encryption 297

Accessing encrypted data 297

Restrictions of Encrypted Data 298

Externally Maintained Attributes 299

Revision Control 299

Sharing LOVs 299

Setting as Externally Maintained 299

Full Text Indexable Attributes 302

List of Values (LOVs) 305

LOV Features 306

Values by Reference 306

Revision Control 306

Validation Base Type 306

Working with Large LOVs 306

Creating an LOV Group 309

Creating an LOV 311

Adding Values to an LOV 317

Adding a Single LOV Value 317

Adding a Group of LOV Values 318

Exporting the Existing LOV to Create a STEPXML File 318

Modifying the File to Include the New Values 321

Importing New Values Into STEP 323

Adding IDs to Existing Values in LOV 327

Allowing Value IDs to be Added to an LOV 334

Consolidating LOVs 337

Merging LOVs 337
Merge Values in LOV 337
Merging LOV Values in Bulk 337
Merging LOVs 338
Merging Values in an LOV 342
Merging LOV Values in Bulk 347
Creating and Formatting the CSV File 347
Uploading the Merge LOV Values CSV Sheet 349
Monitoring LOV Value Merge Events 351
Deleting LOV Values in Bulk 352
Prerequisites 352
DeleteValue Tag and XML Attributes 352
Deleting LOVs 354
Filtering LOVs 356
Filtering on Attribute and PPH 356
Filtering on Attribute and Classification Hierarchy 361
Combining PPH and Classification Hierarchy LOV Filtering 362
Filtering LOV Values 363
Filtering by Hierarchy on LOV Attributes 366
Filter an LOV Based on Hierarchy 366
Workflows and LOVs 370
Initiate an LOV in a Workflow 370
Remove an LOV from a Workflow 372
Maintaining Attributes 373
Attribute Value Migration 374
Prerequisites 374
Technical Migration Details 375
Configuring Attribute Value Migration 376
Starting the Attribute Value Migration 380
Statistics 381
Completing the Migration 382
Changing an Attribute ID 385
Converting Attributes from Date to ISO Date 390
Merging Attributes 390
Switching Attribute IDs 391
Creating Attributes 392
Attribute ID Considerations 396
Handling Duplicated Attributes in Web UI 397
Create Attribute Group(s) to Hold Attributes Known to Exist in Multiples Groups 397
Setting up Blacklisted Attribute Groups in Web UI 397
Merging Attribute Values 399
Verify Attributes to be Merged 399
Merge Values Using Bulk Update 402
Reparenting Attribute Groups 404
Manual Reparenting 404
STEPXML Reparenting 408
Viewing Attribute Values 421
Mandatory Attributes 424
Standard Mandatory Attributes / Visual Indicator in Workbench 424
Workbench Visual Indicators 424

Verify Mandatory Attribute Values Upon Approval 427
Business Rules for Mandatory Attributes 428
Smartsheet Mandatory Attributes 428
Web UI Mandatory Attributes 429
Workflow Mandatory Attributes 429
Single and Multi-Valued Attributes 430
Editing Attribute Values 432
Inherited and Local Values 434
View Filter Options 434
Attributes with Units 434
Editing 'ISO Date' and 'ISO Date and Time' validation base type 435
Entering Values 435
Overwriting Values 436
Editing Multi-Valued Attributes 437
Unlinking Attributes from Objects 440
Unlink Object from Attribute References Tab 440
Unlink Attribute from Object References Tab 440
Specification Attributes 442
Specification Attribute Facts 442
Inherited Attributes 444
Validity on Specification Attributes 445
Units 447
Unit Groups 447
Unit Conversions 448
Conversion Factor and Conversion Offset 448
Maintaining a Unit 450
Create a Unit 450
Delete a Unit 451
Maintaining a Unit Conversion 454
Creating a Unit Conversion 454
Deleting a Unit Conversion 456
Maintaining a Unit Group 457
Creating a Unit Group 457
Deleting a Unit Group 458
Validation Rules 460
Validation Base Type 464
Editing Validation Rules 471
Custom Validation Error Messages 474
Setting Up a Custom Validation Error Message 475
Using the Custom Validation Error Message 475
Input Masks 477
Example Input Mask Use Cases 477
Definitions of Input Mask Characters 478
Static Mask Characters 480
Background Processes and Queues 481
Evaluating BGPs 482
Parallel and Multi-Threading Properties 482
Single Application Server and Single Queue Example 484
Two Application Servers Example 484
Single Application Server and Two Queues Example 485

Understanding config.properties Settings . 485
BG Processes Execution Management .. 487
Background Process Execution Overview .488
Servers 489
Running Background Processes 489
Waiting Background Processes 489
Priority 490
Parallel Processes 491
BG Processes Failsafe 492
Integration Endpoints 492
Possible Values for Quarantined State 492
Searching for Quarantined Background Processes 494
BG Processes States 495
Queued Processes 495
Active Processes 495
Completed with Errors Processes 496
Ended Processes 497
Default Configuration for Background Process Queues 498
Modifying Background Process Queue Configuration 500
Update Queue Configuration 500
Scheduled Process Properties 502
Creating a Scheduled Background Process 504
Deleting a Scheduled Background Process 505
Managing Background Processes 508
Automatic Deletions 508
Manual Deletions 509
Business Module Manager 511
Component Models 513
Dimensions, Dimension Points, and Contexts 515
Dimensions and Dimension Points 516
Dimensions 516
Dimension Points 516
Dimension Dependent Data 517
Dimension and Dimension Points Recommended Practices 519
Dimension Do's and Don'ts 519
Dimension Do's 519
Dimension Don'ts 519
Additional Attributes as an Alternative to Additional Dimensions 519
Language Dimension 520
Country Dimension 521
Business Unit Dimension 522
Market Dimension 522
Dimension Point Do's and Don'ts 523
Dimension Point Do's 523
Dimension Point Don'ts 524
Maintaining Dimensions and Dimension Points 525
Creating a Dimension 525
Creating a Dimension Point 527
Deleting a Dimension Point 528
Deleting a Dimension 529

Selecting a Dictionary for a Language Dimension Point 530

Adding a Dimension Dependency After Loading Data 531

Add a Dimension Dependency to an Attribute After Data Has Been Loaded 531

Add a Dimension Dependency to a Reference Type After Data Has Been Loaded 532

Contexts 533

Dimension Dependent Data 534

Context Recommended Practices 536

Context Do's 536

Context Don'ts 536

Global Context 536

Deleting the Global Context 537

Context Locales 538

Locale Do's 539

Context Mode View 540

Maintaining Contexts 541

Creating a Context 541

Editing a Context 543

Deleting a Context 543

Hidden Values 543

Specifying a Default Context 544

Events 547

Core Events 548

Prerequisites 548

Core Events and Event Types 548

Core Event Generation 550

Using Core Events 551

Change Flags for Events 552

Change Flag Example 554

Derived Events 558

Prerequisites 558

Add a Derived Event Type 558

Use a Derived Event Type in an OIEP 559

Delete a Derived Event Type 561

Events Generated on Main Workspace 562

Configure Events for Unapproved Updates 562

Events Not Triggered 566

Event Processors 567

Prerequisites 567

Setup Requirements 567

Event Processor Results 568

Processing Plugins 569

Asset Publisher Processing Plugin Parameters and Triggers 572

Prerequisites 572

Parameters 572

Event Triggers 576

Republishing Assets 579

Asynchronous Translation Message Processor Processing Plugin Parameters and Triggers 580

Prerequisites 580

Parameters 580

Event Triggers 582

CASS Certification Report Processing 583

Plugin Parameters and Triggers .....
Prerequisites 583
Parameters 584
Event Triggers 585
Data Sufficiency Calculator Processing Plugin Parameters and Triggers **587**
Prerequisites 587
Parameters 587
Event Triggers 588
Elasticsearch Indexer Processing Plugin Parameters and Triggers **590**
Prerequisites 590
Parameters 590
Event Triggers 591
Execute Business Action for Event Batch Processing Plugin Parameters and Triggers **594**
Prerequisites 594
Parameters 594
Event Triggers 596
Execute Business Action Processing Plugin Parameters and Triggers **598**
Prerequisites 598
Parameters 598
Event Triggers 600
Expand LOV Value Changes Processing Plugin Parameters and Triggers **602**
Prerequisites 602
Parameters 602
Event Triggers 603
For Use with External Systems 604
Experian Email Validation Processing Plugin Parameters and Triggers **605**
Prerequisites 605
Parameters 605
Event Triggers 607
Image Cache Processing Plugin Parameters and Triggers **610**
Prerequisites 610
Parameters 611
Event Triggers 612
Matching Processing Plugin Parameters and Triggers **615**
Prerequisites 615
Parameters 615
Event Triggers 618
Troubleshooting 619
Revision Management Processing Plugin Parameters and Triggers **620**
Prerequisites 620
Parameters 620
Event Triggers 622
Source Record Data Management Processing Plugin Parameters and Triggers **624**
Prerequisites 624
Parameters 624
Event Triggers 626
Initial Setup for Event Processors **628**
Prerequisites 628

Create Setup Group Type for Event Processors 628
Link Event Processor Object Types to Setup Group Types 631
Create an Event Processor Setup Group 632
Configure Action Sets and Privileges 634
Create Event Processor Action Sets 634
Configure Event Processor Privileges 634
Creating an Event Processor 636
EPW - Identify Event Processor 638
EPW - Configure Event Processor 639
EPW - Configure Processing Plugin 641
EPW - Schedule Event Processor 643
EPW - Configure Error Reporter Processing Plugin 646
Running an Event Processor 649
Event Processor Status Options 649
Enable Event Processor 650
Invoke Event Processor 651
Disable Event Processor 652
Generating Events from Main 654
Maintaining an Event Processor 655
Edit Event Processor 656
Delete Event Processor 656
EP - Event Processor Tab 657
Prerequisites 657
Description Flipper 657
Configuration Flipper 658
Current Background Process Log Flipper 661
EP - Event Triggering Definitions Tab 664
Prerequisites 664
Event Triggering Definitions 664
EP - Background Processes Tab 666
EP - Statistics Tab 668
Statistics Flipper 668
EP - Error Log Excerpts Tab 669
Prerequisites 669
Event Processor Log Flipper 669
EP - Log Tab 670
Prerequisites 670
Logs 670
Monitoring an Event Processor via External System 671
Event-Based Event Processor Queued Events 672
Estimate Unread Events 672
View First Event Batch 672
Event Queues 674
Creating a New Event Queue 675
Maintaining Event Queues 678
Create or Edit Event Consumer 680
Creating or Editing an Event Consumer 680
Output 683
Metrics 685
Aggregator Metrics 686
Aggregator Metric Configuration 686
Adding a business function to an aggregator metric 688

Attribute Comparison Metric **691**
Attribute Comparison Metric configuration 691
Validation Errors 695
Attribute Value Analysis Metric **697**
Attribute Value Analysis Metric configuration 697
Validation Errors 702
Business Function Metrics **704**
Business Function Metric Configuration 704
Conditional Attribute Value Metric **707**
Conditional Attribute Value Metric configuration 707
Validation Errors 712
Initial Setup for Metrics **714**
Number of Referenced Target Objects Metric **717**
Number of Referenced Target Objects Metric Configuration 717
PDX Channel Metric **723**
PDX Channel Metric configuration 724
Creating and Editing Metrics **728**
Creating a Metric Object 728
Configuring a Metric 728
Set a Default Metric 730
Completeness Metrics **731**
Configuration 731
Migrating Completeness Scores to Description Attributes **734**
Migrate Completeness Scores 734
Value Metrics **737**
Configuration 737
Business Function Metrics **740**
Business Function Metric Configuration 740
Aggregator Metrics **743**
Aggregator Metric Configuration 743
Adding a business function to an aggregator metric 745
Metric Visualization **748**
View Quality Scores 748
User Privileges 751
Define Completeness Scores 751
Setting Completeness Scores for Data Containers 752
Export Completeness Values 754
Metric Visualization in Web UI **755**
Adding the Completeness Meter Component 755
Configuring the Completeness Meter Component 756
Object Types and Structures **757**
Attributes Valid for Object Types **761**
Setting Validity 761
Autogenerate Using Name Pattern and ID Pattern **762**
Defining the Name and ID Pattern 762
Basic Object Types **766**
Object Type Dimension Dependencies **769**
Object Type Icons **770**
Editing an Object Type Icon 770
Maintaining Object Types **773**

Creating an Object Type **774**
Create an Object Type 774
Deleting an Object Type **776**
Unlinking Object Types **779**
Updating Object Types from Another Object Type **781**
Steps to Update an Object Type from Another Object Type 781
Maintaining Dimension Dependent Object Types **785**
Making an Object Type Dimension Dependent 785
Removing Dimension Dependency from an Object Type 786
MIME Types **789**
Setting MIME Types for Object Types **791**
Editing MIME Types LOV 792
Modifying Manually Sorted **794**
Object Types **796**
Object Type Hierarchy **798**
Classification Hierarchy Structures **799**
Fixed Hierarchy Structure **801**
Variation in Hierarchy Depths **802**
Multiple Parents 803
Creating a Standard Publication Object Type Hierarchy **804**
Creating a Publication Group Object Type **806**
Creating a Publication Object Type **808**
Creating a Publication Section Object Type **810**
Product-Override Object Types and Commercial Object Types **812**
Creating a Commercial List Type **813**
Creating a Product-Override Object Type **816**
Enabling a Product-Override **817**
Owns Product Links on Alternate Classifications Object Type **818**
Owns Product Links Affects STEPXML 819
Changing Owns Product Links **824**
Steps to Change the Owns Product Links 824
Reference Target Lock Policy on Object Types **826**
Strict or Relaxed 826
Revisability on Entity Object Type **828**
Product Variants **830**
Setting Up Product Variants **832**
Add the Object Types to the Component Model 832
Create a Product Variant Priority Attribute 833
Specify Variant and Non-variant Attributes 837
Products Variants List in Workbench **840**
Use the List View 840
Create a Variant 841
Delete a Variant 841
Products Variants List in Web UI **843**
Configure Product Variants List 844
Dimensions 845
Enable Filtering 846
Headers 848

Hide Selection Buttons 848
Selection Screen 848
Sort Column 848
Title 848
Use Immediate Save 848
Actions 848
Prevent Duplicate Creation 849
Preview Variant Combinations 849
Use the List View 850
Create a Variant 851
Product Variants Matrix in Workbench .. 854
Use the Matrix View 854
Create a Variant 857
Delete a Variant 857
Product Variants Matrix in Web UI 859
Configure Product Variants Matrix 860
Initial Hide Empty 862
Title 862
Actions 862
Prevent Duplicate Creation 863
Preview Variant Combinations 863
Use the Matrix View 864
Create a Variant 865
Business Condition: Validate Product Variant 868
Recycle Bin for System Setup 870
Reference and Link Types 872
References 872
Product to Classification Links 873
Product Attribute Links 874
Classification Attribute Links 874
Classification Attribute Link Type 875
Creating a Classification Attribute Link . 877
Setting Up an Attribute Link Type 880
Maintaining Attribute Link Metadata Values 883
Inherited Metadata Values 883
Local Metadata Values 884
Mandatory Linked Attribute Values 886
Product Attribute Link Type 888
Creating a Product Attribute Link 890
Setting Up an Attribute Link Type 893
Maintaining Attribute Link Metadata Values 896
Inherited Metadata Values 896
Local Metadata Values 897
Mandatory Linked Attribute Values 899
Product to Classification Link Types 901
Creating a Product to Classification Link Type 903
Product to Classification Link Type - Enter ID and Name 905
Product to Classification Link Type - Select Valid Product Types 906
Product to Classification Link Type - Select Valid Classification Types 907
Product to Classification Link Type - Apply Dimension Dependencies 909

Product to Classification Link Type - Advanced **910**

Creating a Product to Classification Link **912**

Maintaining a Product to Classification Link Type **915**

Product to Classification Link Type Tab 915

Validity Tab 919

Log Tab 921

Attribute Groups for Reference and Link Types **922**

Dimension Dependent Reference and Link Types **925**

Dimension Dependency on the 'Reference-Type' Object Type 925

Dimension Dependency on a Reference Type 926

Dimension Dependent Reference Example 927

Inheritance Example for a Product to Classification Link Type **930**

Metadata Attributes on Reference and Link Types **939**

Add a Metadata Attribute 939

Edit a Metadata Attribute 941

Visibility of Reference and Link Types ... **943**

Reference Types **947**

Creating a Reference Type **949**

Reference Type - Enter ID and Name **951**

Reference Type - Select Valid Source Types **952**

Reference Type - Select Valid Target Types **953**

Reference Type - Apply Dimension Dependencies **954**

Reference Type - Advanced **955**

Creating a Reference **957**

Maintaining a Reference Type **960**

Reference Type Tab 960

Validity Tab 966

Log Tab 967

Attribute Groups for Reference and Link Types **968**

Direction of a Reference **971**

Dimension Dependent Reference and Link Types **973**

Dimension Dependency on the 'Reference-Type' Object Type 973

Dimension Dependency on a Reference Type 974

Dimension Dependent Reference Example 975

Entity Reference Types **978**

Parent/Child Relation Example 979

Inheritance Example for a Reference **981**

Metadata Attributes on Reference and Link Types **984**

Add a Metadata Attribute 984

Edit a Metadata Attribute 986

Multiple References for the Same Objects **988**

References on Data Containers **989**

Configuring a Reference Type with a Data Container Type as Source Object 989

Editing References on Data Containers 991
Viewing Data Container References via Referenced By Tab 991
Visibility of Reference and Link Types 992
Revisions 996
Comparing Revisions 997
Generating Revisions 1000
Maintaining Revisions 1002
Reverting to Past Revisions 1005
Timing for Automatic Revisions 1007
Setup Groups 1009
Creating Setup Group Object Types 1011
Defining Type of Objects to be Stored in Setup Groups 1013
Creating Setup Groups in System Setup 1016
Setting Up Privileges Example 1019
Privilege Setup Steps 1020
Single-Update Mode 1025
Operations that Cause Single-Update Mode 1026
Single-Update Mode Examples 1026
Guidance for Preventing SUM Activation 1028
Tags 1029
Tag Systems in STEP 1029
Types of STEP Tags 1029
Working With Tags 1030
Creating Tag Groups and Tags 1032
Tag Groups 1032
Tags 1033
Configuring Tag Definitions 1035
Rendering 1036
Configuring Tag Output Formatting 1040
Output Formatting Flipper 1040
InDesign Output Formatting 1042
Tag Conversion Parameter 1042
Special Character Tags 1046
Insert a Special Character 1046
Character Tags 1049
> and < Characters 1049
& and " Characters 1049
Insert a Character Tag 1049
Style Tags 1051
Insert a Style Tag 1051
Hyperlink Tags 1056
Insert a Hyperlink 1056
Unique Keys 1059
Defining Keys 1059
Key Attribute Considerations 1060
Key Example 1060
Key Formula Example 1061
Key Formula Example 1061
Important Notes 1061
Creating and Deleting Keys 1062
Create a Key 1062
Delete a Key 1064

Activating and Deactivating Keys **1066**
Activate a Key 1066
Deactivate a Key 1067
Checking Keys **1069**
Creating Objects with Active Keys **1070**
STEPXML 1070
Web Services APIs 1072
Web UI Simple Importer 1073
Workbench 1073
Matching Keys During Import **1074**
Users and Groups **1077**
Group Membership 1077
Working with User Groups **1079**
Creating a User Group 1079
Deleting a Group 1080
Working with Users **1082**
Creating a User 1082
Bulk Adding New Users 1083
Adding a User to a Group 1085
Duplicating a User 1086
Removing a User from a Group 1087
Deleting a User 1088
Considerations 1090
Changing a User Password **1091**
Changing a User Password - System Administrator 1091
Changing a User Password - User 1091
From the workbench 1092
From the Web UI 1092
User Anonymizer **1095**
Starting the Anonymization Process 1095
Example: Anonymizing a User 1097
System Settings **1102**
Classification Hierarchy Settings **1103**
Dimension Dependencies 1103
Image and Document Settings **1106**
Dimension Dependencies 1106
Store assets and DTP documents in 1107
DTP asset source 1107
Pregenerate thumbnail cache on upload 1107
Disable auto-cleanup of thumbnail cache 1107
Transformation Lookup Tables follow asset dimension dependency 1108
Asset Import Compatibility Mode 1108
Asset Push Attribute Macro Keys **1109**
Calculated Attribute Settings **1110**
Cryptographic Keys **1111**
Creating cryptographic keys 1111
Password Requirements 1112
Validation Templates **1113**
To create a Validation Template: 1113
Product Information Manager Default Settings **1115**
Enforce Mandatory Check for Attributes, References and Links 1115
Product Editor, Group attributes by top group 1116

Localize numbers with thousand delimiter when localizing exports 1117

Localize dates when localizing exports 1118

Report Save As CSV Character Set 1118

Default Attribute to use as Display Sequence Attribute 1119

Default Metric 1120

Conditional Validity Attribute 1120

Block Attribute Groups with more than 1000 attributes 1120

Use full pathname for classes on Product References Tab 1120

Pass through unconverted Special Characters and Tags (Y) or discard them on output (N) 1121

Product Attribute Help metadata attribute 1121

Attribute indicating the Priority of Product Variant Attributes 1122

Show both name and ID in the PIM navigator 1122

Comma separated list of node types for which to show both name and ID 1123

Attribute Header Column Width 1124

Reference Type Column Width 1125

Translation Settings 1126

Translation File Pattern 1126

Should "Filter spreadsheet" option be shown in Translation Wizard 1126

Default value for "Filter spreadsheet" option in Translation 1128

Always Excluded Attributes 1128

Terms List Settings 1130

Add the products from the selection list import file directly instead of looking up leaf products 1130

Web Services 1131

Tag format for Web Services and REST APIs 1131

Web UI Settings 1132

Control Override Meta-Attribute 1132

Default Context 1134

Default Workspace 1134

Web UI Supplier Classification Object Type 1134

Web UI Supplier Products Classification Object Type 1134

Assets Classification Object Type 1134

Batches Classification Object Type 1135

Batch Classification Object Type 1135

Supplier Privileges Entity to Classification Reference Type 1135

Web UI Supplier Entity Classification Object Type 1135

Proof View Stylesheet Attribute 1135

Step-users Web UI Batches Folder 1136

Step-users Web UI Assets Folder 1136

Context Help Metadata Attribute 1136

Link Type for Vendor Classification to Product Link 1136

Enable All-view for Users that are a Member of Multiple Suppliers 1136

Primary Image Type 1137

Flatplanner Default Settings 1138

Enable keep layouts for publications 1138

Show frame information on flatplans with layout pickup 1139

Reapply Flatplan rules when adding product to non-empty frame 1140

DTP Settings 1141

Use Publication Name prefix when naming DTP documents 1141

Enable calculation of border rectangle space usage for actual pages 1141

Calculated DTP Table width includes left and right rules 1142

Attribute Display Sequence 1142

Attribute Name Style Meta-attribute and Attribute Value Style Meta-attribute 1142

Asset Order Meta-attribute 1143

Table Cell Skip Empty Cell 1143

JPEG DPI 1143

DTP Page - preview image 1144

Inherit Default Product Template through product hierarchy for Product Overrides. 1144

Proof View Start Page Number 1144

Data Quality Monitoring Settings 1146

Selecting a Data Policies Default Context 1146

Table Defaults 1148

Index Word Settings 1149

Revisability Settings 1150

Threshold (hours) for changes by same user to generate a new revision 1150

Business Rule Approve Overview 1151

Security Policy 1152

GUI Localization 1156

Setting up a GUI Localization for a User 1157

Privilege Rules 1160

Privilege Rule Types 1161

Restricting Actions 1162

Unlimited Number of Privilege Rules 1162

Editing Linked Objects 1163

Access to Classifications and Products 1163

Read-Only Access 1163

Adding User Privileges for a Group 1164

Add Privilege Rule Options 1165

Adding Setup Privileges for a Group 1170

Add Setup Privilege Rule Options 1170

Editing Privilege Rules 1173

Edit Privilege Rule Options 1174

Removing a Privilege Rule from a Group 1179

Configuring Workflow Privileges 1180

Workflow state selector 1180

Editing Privilege Rules with focus on STEP Workflows 1181

Limitations on Workflow privileges 1183

Object Creation Privileges 1186

Considerations When Creating New Object 1187

Mass Creation Screen in Web UI 1187

Smartsheets 1187

Importer 1187

Privileges and Security 1188

Granting Access to the SDK and API Documentation 1189

Grant Access to the SDK and API documentation 1189

GUI Setup 1192

GUI Setup on Groups 1193

GUI Setup for Users 1197

Log for Users and Groups 1199

Web UI User Impersonation Setup and Reporting 1201

Value Generators 1206

Creating a Value Generator Configuration Type 1207

Algorithm Value Generator 1211

Prerequisites 1211

Configuring Business Functions to Generate a Value 1212

Configuring a Business Function to Release a Value 1215

Configuring an Algorithm Generator 1216

Configuring a Business Action to Generate / Release Values 1218

List Value Generator 1221

Configuring a List Value Generator 1221

Importing Values into the List Value Generator 1222

Configuring Business Actions for a List Value Generator 1224

Sequence Value Generator 1227

Configuring a Sequence Value Generator 1227

Configuring Business Actions for a Sequence Value Generator 1229

Wiki Metadata 1231

Setup Requirements 1231

Additional Information 1231

Initial Setup for Wiki Metadata 1232

Testing Wiki Metadata 1233

Creating Wiki Metadata Pages 1234

Bulk Wiki Page Creation 1234

Editing Wiki Metadata Pages 1236

Viewing Wiki Metadata Pages from Workbench 1239

Viewing Wiki Metadata Pages from Web UI 1242

Workspaces 1246

Standard Workspaces 1247

Maintaining Workspaces 1248

Creating Workspaces 1248

Workspace Log 1248

Deleting Workspaces 1248

System Setup

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

Prerequisites

The guide assumes the following:

- Users are familiar with the **Getting Started / User Guide** documentation.
- Users have read the **Platform and Software Support** requirements.
- Users have attended or have taken the **MDM Solution Fundamentals** training course.

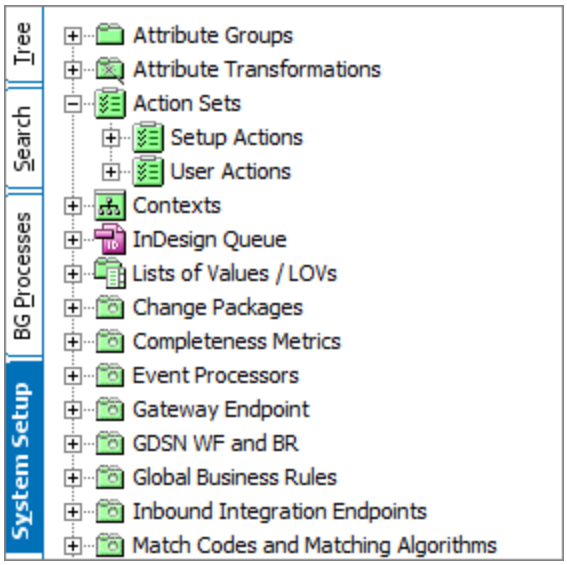
Accessing Shortcut Menus on PC and Mac

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.

Action Sets

Action Sets are used to define the actions or privileges a user or group of users are permitted to perform. Actions are grouped into Action Sets and made available for User Groups under the Users & Groups setup area. For more information, see the **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 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, and delete) the object or task in question. These actions can be added to Action Sets, as defined in the **Action Sets** topic.

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

All visible actions shown to users are for the global system. Some actions are related to licenses which may have not be applicable to your particular system.

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
System Administration	Ability to maintain configuration properties, prepare and deploy Extension API extensions, retrieve the STEP system status, and restart the STEP system via the remote system administration REST API. This functionality is strictly for Stibo Systems' STEP SaaS customers only. **Note: The 'System Administration' privilege should only be given to trusted system administration users.**
View Administration	Ability to access the STEP System Administration pages from the Start Page
View System Setup Logs	Ability to view the System Setup Log and Workspace Log (available from View > Logs in the workbench) and the Log tab (available in the editor on most System Setup objects). Without this privilege, the System Setup Log option in the View > Logs menu is grayed out, and the Log tab is not visible on System Setup objects.

Assets

Action	Privileges Granted
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 • 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, 'Delete attribute' global privilege must be added.

Action	Privileges Granted
	• To delete LOV attributes, the action 'Delete domain' must also be added.
Maintain validation templates	Create a validation template (validation base type) for attributes and LOVs
Remove attribute from attribute group	Ability to remove (unlink) attributes that are linked to other attribute groups as well as add attributes to a group.
Merge product attributes	Ability to merge attributes
Modify definition of product attribute (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.

Action	Privileges Granted
View attribute	Ability to view attributes
View attribute group	Ability to view attribute groups The 'View attribute group' action must be added for users to maintain attributes.
View metadata for attribute (property value)	View values of description attributes linked to an attribute

Background Processes

Action	Privileges Granted
Delete background-processes	Delete queued, started, and ended background processes
View Background Processes of Other Users	In workbench, while displaying other users' processes via the show all button () , the user can view and download files from background processes started by any other user. In Web UI, the user can view and download files from the background processes they started. For users without this privilege: • in the workbench, the show all button () on the BG Processes tab is disabled (For details, see the **BG Processes Displayed** section in the **BG Processes Tab** topic of the **Getting Started / User Guide** documentation.) • in the workbench and Web UI, processes started by other users are not visible • no IIEPs, OIEPs, and event queues are visible (Gateway integration endpoints are visible since 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 Modules

Action	Privileges Granted
Install Business Modules	Activates the Business Module Manager option on the File menu, and enables the ability to install business modules that have been activated by Stibo Systems.

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
Maintain Value Generator	• Create new value generator • Edit value generator • Run value generator
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

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

GDSN Data Pools

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

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

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

Integration Endpoints

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

Link Types

Action	Privileges Granted

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

List of Values (LOVs)

Action	Privileges Granted
Create domain	Ability to create LOVs **Note:** The actions 'Modify name/description of domain (also translate)' and 'Modify definition of domain (validator, etc.)' must also be included in order to create LOVs.

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

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

Print Publication

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

Purging

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

Reports

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

SDK / API Documentation

Action	Privileges Granted
View SDK Documentation	Ability to view the SDK / API documentation from server/sdk and the STEP API Documentation button on the Start 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.
View table types	Allows users to view tables and table types.

Tags

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

Units

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

Action	Privileges Granted
	• 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 Web UI
View context	In addition to granting access to the workbench, 'View context' also grants the ability to access the Web UI.
Web UI Administration	Grants the ability to: • "Revert" / copy to front • Delete action context menu • Duplicate action context menu • Status tab • Edit Web UI through designer • Edit Web UI through Workbench • Change Web UI ID in the XML

Workflows

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

Action	Privileges Granted
	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 also have the 'View STEP Workflow Profile' privilege 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.
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.
View tasks assigned to other users in my	Allows users to view all assigned tasks in any and all groups in which they are a member.

Action	Privileges Granted
group(s)	This privilege differs from what is currently provided by the 'STEP Workflow Administrator' setup action in that the 'STEP Workflow Administrator' setup action allows users to view all tasks across all groups.

Workspaces

Action	Privileges Granted
Maintain workspaces	Create and delete workspaces

User Actions

Once actions are added to a User Action Set, they are assigned to a user group via the Privilege Rules tab on a user group within the Users & Groups node of System Setup.

Note: All visible actions shown to users are for the global system. Some actions are related to licenses which may have not be applicable to your particular system.

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

Action	Privileges Granted
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.
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

Action	Privileges Granted
	In order to import and replace images and documents, these setup actions must be added to the privileges: • Create value in medium domain • Modify value in hard/medium domain
View asset	View assets **Note:** The 'View asset' action must be added in order to work with images and documents. To make any of the view actions listed below have an effect, the 'View asset' privilege must be a part of the action set. For instance, if the action 'View asset reference' is added and not 'View asset,' the 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

Action	Privileges Granted
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 Preparation Functionality

User actions for data preparation functionality allow users to run list processing.

Action	Privileges Granted
Run List Processing	Access to the list processing functionality.

Data Quality Functionality

User actions for data quality functionality allow users to configure, modify, and generate the available DQ profiles.

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

eCatalog

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

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

GDSN Data Pools

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

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

Products, Classifications, and Entities

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

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

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

Action	Privileges Granted
Approve Classification	Approve product classification links

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

Action	Privileges Granted
Create entity	Create, change, and duplicate entities **Note:** The action 'Modify name/description of entity (also translate)' must be included to create entities.
Create entity reference	Create entity references
Create product	• Create products of specified product object types • Change object types of specified products • Duplicate products The action 'Modify name/description of product (also translate)' must be included in order to create products.
Create product reference	Create product to product references
Declassify product (unlink product from classification)	Unlink products from classifications
Delete classification	Delete classifications of specified object types
Delete classification reference	Delete classification references
Delete data container	Allows the user to delete a data container
Delete entity	Ability to delete entities
Delete entity reference	Delete entity references
Delete product	Delete products of specified object types
Delete product	Delete product to product references

Action	Privileges Granted
reference	
Edit Data Container Key Values	Allows the user to edit data container keys within a data container after the data container keys are assigned. **Note: It is recommended that only select admin users have the 'Edit Data Container Key Values' privilege added to their user profile and that this privilege be used sparingly.**
Edit tables	Create and maintain tables in classification and product editors.
Legalise/unlegalise table types	Ability to legalize tables on classifications and products. 'Legalise/Unlegalise Table Types' does not grant a privilege to also create and maintain tables. This privilege is covered in the action 'Edit tables'.
Link (product) attribute into classification	Link specification attributes to classifications and products
Modify metadata for attribute to classification link (also translate)	Edit and translate description attribute values on specification attributes linked to classifications or products
Modify metadata for product (property value) (also translate)	Edit and translate the values of description attributes linked to a product
Modify metadata for classification (property value) (also translate)	Modify the metadata for classifications
Modify metadata for classification reference (link value) (also translate)	Modify the metadata for classification references

Action	Privileges Granted
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.
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

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

Action	Privileges Granted
View classification	Ability to view the classification nodes in the classification hierarchy. **Note:** The 'View classification' action must be added to work with classifications. To make any view action have an effect, the 'View classification' privilege must be a part of the action set. For example: If the action 'View classification reference' is added and not the 'View classification' action, the user will not be able to see the classification and therefore not the references either.
View classification reference	View classification references
View data containers	Allows the ability to view existing data containers on entity objects
View entity	Ability to view entities The 'View entity' action must be added to be able to work with entities. To make any of the view actions listed below have an effect, the 'View entity' privilege must be a part of the action set.
View entity reference	View entity references
View metadata for attribute to classification link (link attribute value)	View description attribute values that have been applied to the reference between the attribute and the classification. These attributes will 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.

Action	Privileges Granted
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 product reference (link attribute value)	View link attribute values of attributes that have been applied to the link type marker edge product. These attributes will 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. **Note:** This user action gives access to view products both when used for privilege rules applied to a product or applied to a classification. In the latter case, the user action will give access to view products linked in below the classification.

Action	Privileges Granted
View product attribute value	View the values of specification attributes linked to products
View product reference	View product to product references

Publications

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

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

Action	Privileges Granted
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
Maintain Flatplan Content	Change the object assigned to an existing frame
Maintain Flatplan Frame	Add, move, delete, and resize frames
Maintain public basket view	Maintain public Flatplanner basket views
Maintain public financial view	Maintain public Financial Summary views (Flatplanner)
Maintain public publication planner view	Maintain public Publication Planner views (Flatplanner)
Maintain public stickerbook view	Maintain public Sticker Book views (Flatplanner)
Manage autopage objects	Manage AutoPage objects

Action	Privileges Granted
Manage layout in page inspector	Manage layouts in page inspector (AutoPage)
Manage pagination rules	Manage pagination rules
Modify Basket for Planned Page	Modify Flatplanner baskets at the planned page level
Modify Basket for Publication	Modify baskets at the publication level
Modify Basket for Publication Section	Modify baskets at the section level
Modify DTP Template Document	Modify existing product templates, page templates, and publication templates (master documents)
Modify Generic Page Template	Modify page templates
Modify Planned Page	Add products or images to planned pages. The following menu actions are also controlled by this privilege: • Set page templates on planned pages • Reuse saved layouts • 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

Action	Privileges Granted
Modify Version	• Set version as master • Set context and workspace for a version • Set commercial list for a version
Perform autopage export actions	Perform AutoPage export actions
Perform autopage server write actions	Perform AutoPage server write actions.
Reorder Planned Pages	Reorder planned pages
View DTP Template Document	View publication templates (master documents), product templates, and page templates
View Generic Page Template	View page templates
View Planned Page	View planned pages
View Publication	View publications
View Publication Document	View actual pages **Note:** An actual page is the same thing as a mounted page—a page saved back to STEP from InDesign.
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 a supplier item Web UI

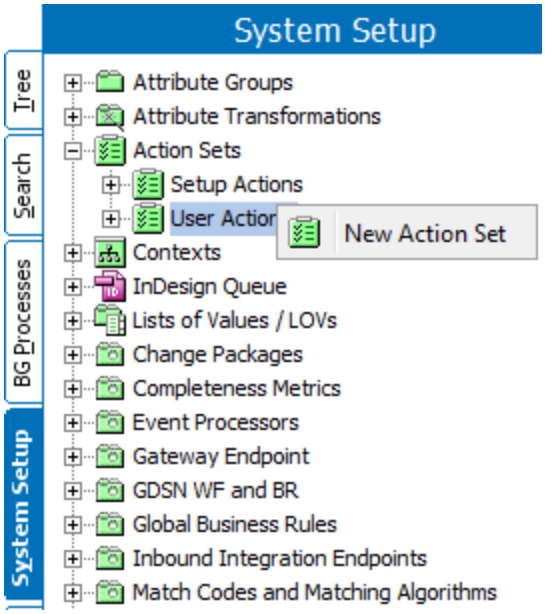
Maintaining Action Sets

An action set allows multiple actions to be added or removed as a group.

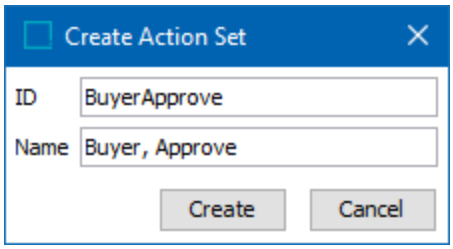
Creating an Action Set

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

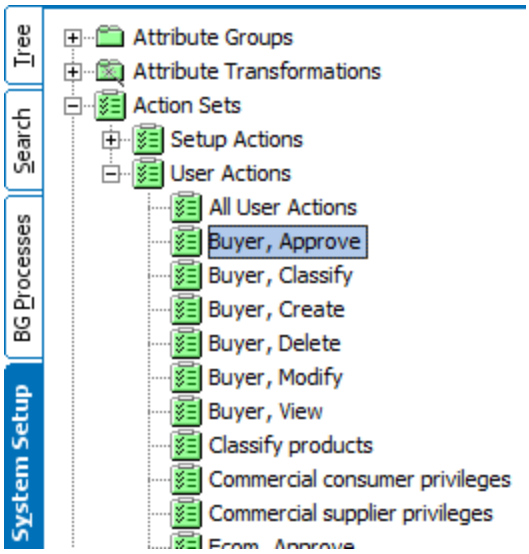
1. In System Setup, open the **Action Set** group and select the relevant main Action Sets folder.
2. Right-click, and then click **New Action Set** (or in the **Maintain** menu, point to **Insert** and then click **Action Set**).



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**.

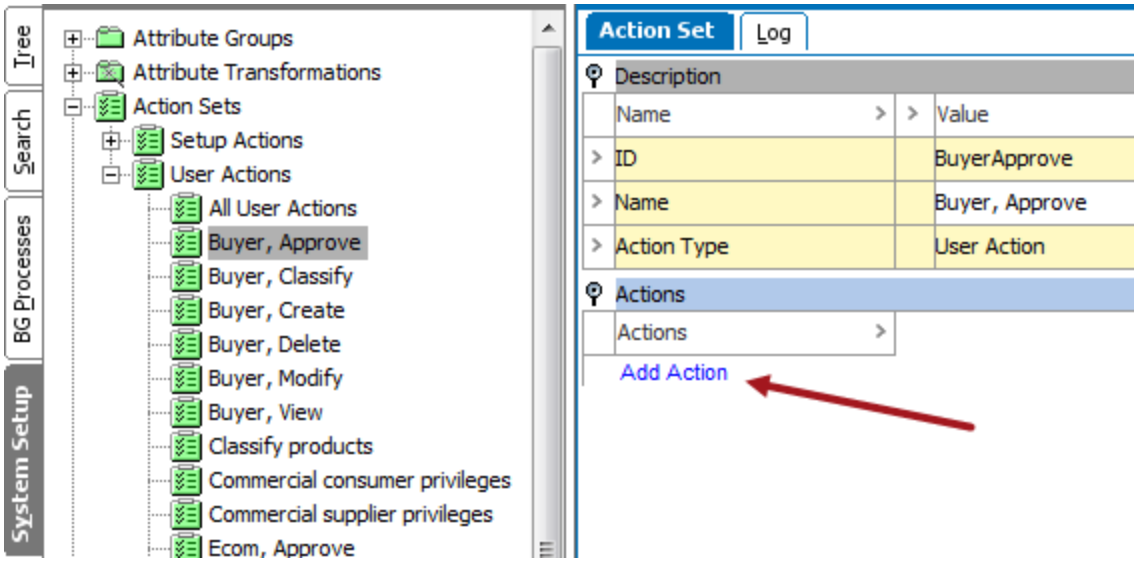


A new Action Set has now been created, and the next step will be to add the specific actions.

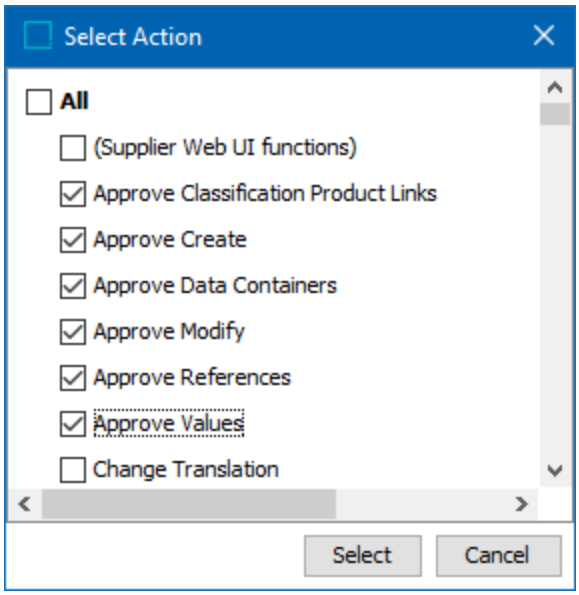
Adding Actions to an Action Set

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

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

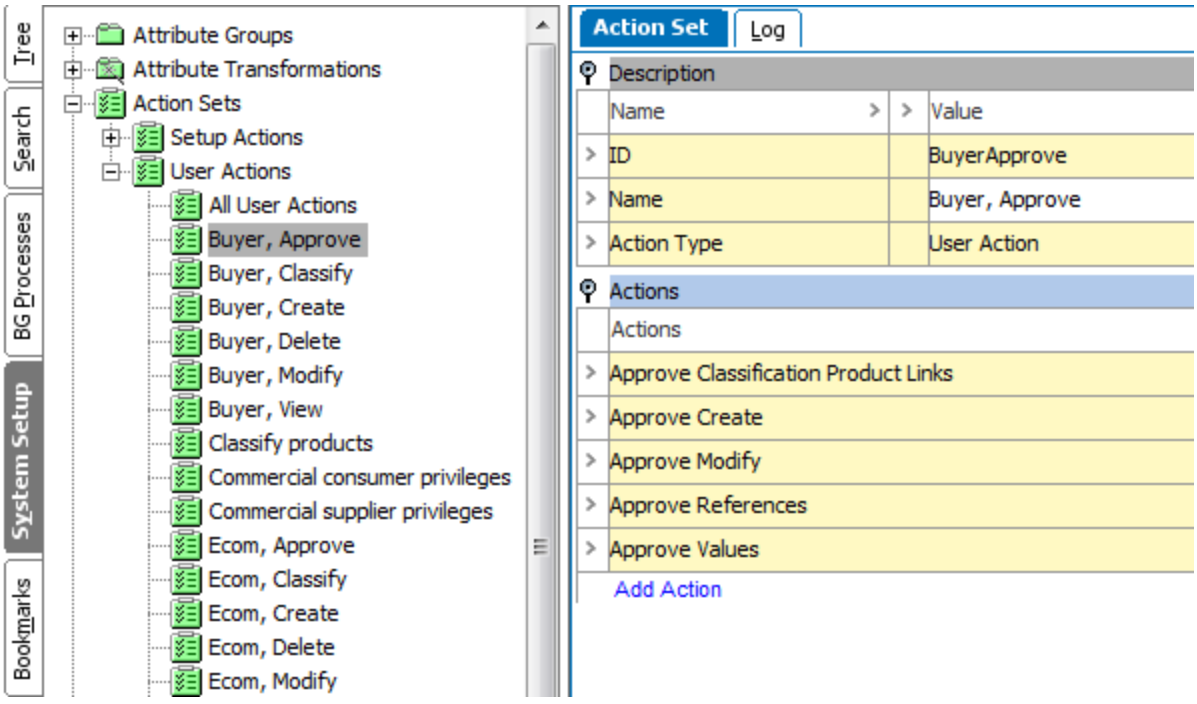


3. Under **Actions**, click **Add Action** and the **Select Action** dialog box appears.



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

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

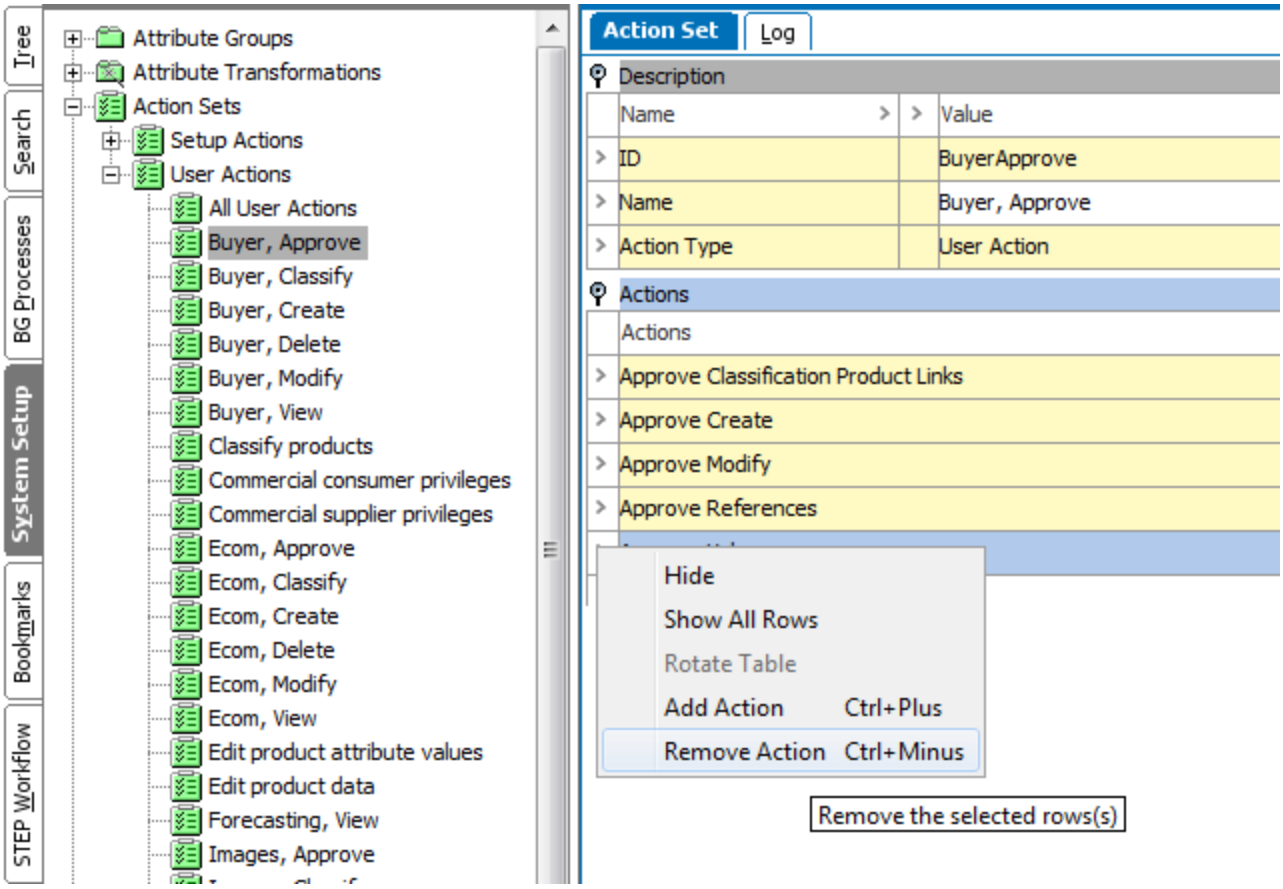


The selected Action Set will now include the added action(s), and when an action set is assigned to a user group, the members will be able to perform all actions in the actions list.

Removing an Action from Action Set

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

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



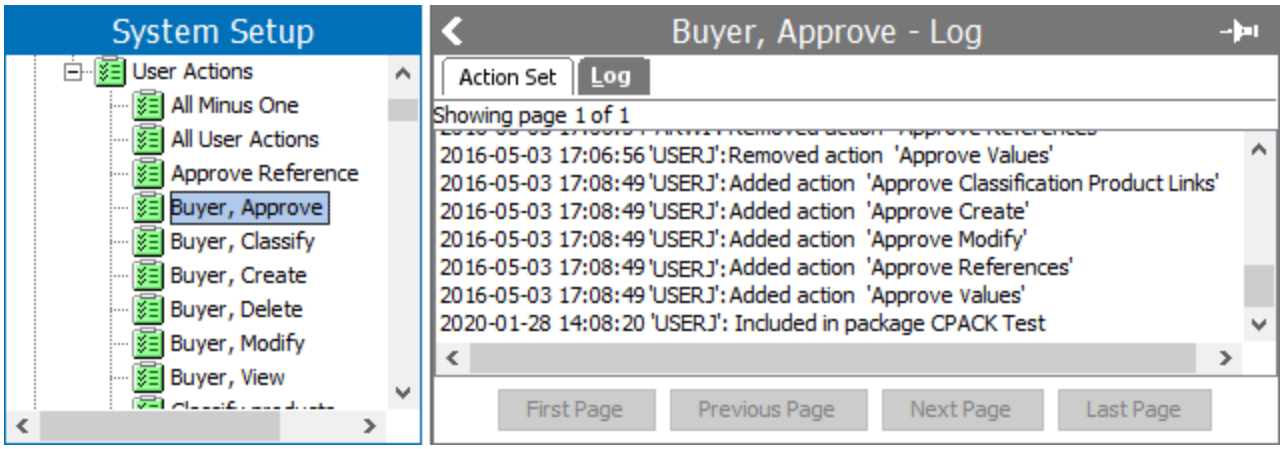
4. Press **Ctrl -** on the keyboard, or right-click the row header arrow, and click **Remove Action**.

   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.

Action Set Log Tab

Once an action set has been created, the Log tab will display specific details, including who created the action set, when the action set was created, and any changes that have taken place to the action set.

This information is essential for audit purposes, as well as aiding data stewards and/or system architects in setting up and updating the action sets.


System Setup
User Actions
All Minus One
All User Actions
Approve Reference
Buyer, Approve
Buyer, Classify
Buyer, Create
Buyer, Delete
Buyer, Modify
Buyer, View
Buyer, Approve - Log
Action Set
Log
Showing page 1 of 1
2016-05-03 17:06:56 'USERJ': Removed action 'Approve Values'
2016-05-03 17:08:49 'USERJ': Added action 'Approve Classification Product Links'
2016-05-03 17:08:49 'USERJ': Added action 'Approve Create'
2016-05-03 17:08:49 'USERJ': Added action 'Approve Modify'
2016-05-03 17:08:49 'USERJ': Added action 'Approve References'
2016-05-03 17:08:49 'USERJ': Added action 'Approve Values'
2020-01-28 14:08:20 'USERJ': Included in package CPACK Test
First Page
Previous Page
Next Page
Last Page

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 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.

Examples of main attribute groups could be:

- Technical group
- Price group

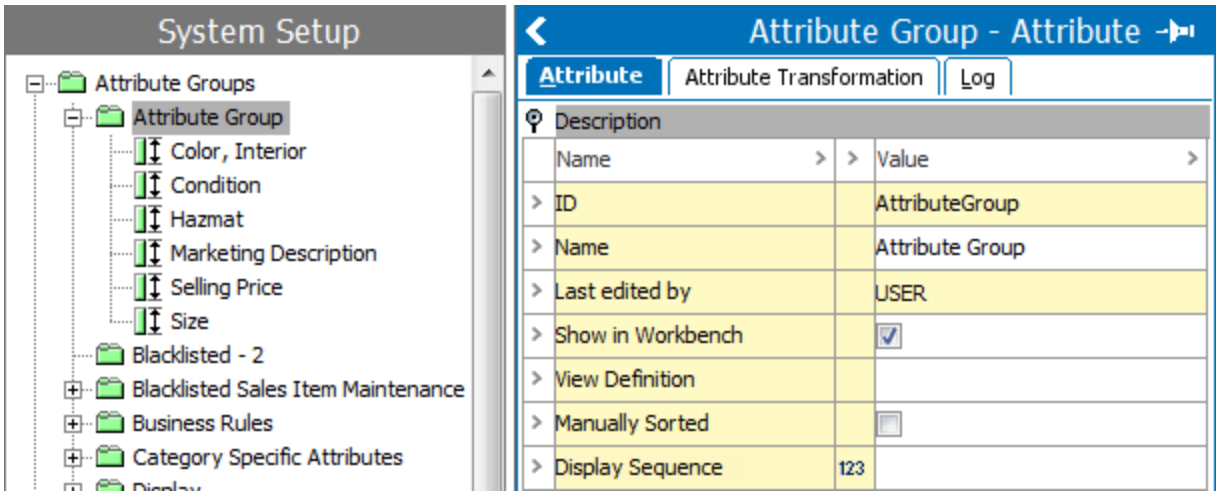
The following are examples of attribute subgroups:

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

In workbench, attribute groups are housed in System Setup as shown below.



For more information on attribute groups, see the following:

- **Attribute Group Display Sequence** (below) defines how to order attribute groups.
- **Creating an Attribute Group** defines how to create a new group.
- **Creating a Customized View** defines the 'Show in Workbench' and 'View Definition' parameters.
- **Manually Sorted** (below) defines how to order attributes with a group.

Attribute Group Display Sequence

By default, attribute groups of a product shown in the Product Editor are in alphabetical order. If an Attribute Display Sequence Attribute has been set on your system, the attribute groups shown in the Product Editor are sorted according to the order set in the Display Sequence attribute. For more information, see the **Display Sequence Attribute** topic.

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.

Manually Sorted

The 'Manually Sorted' parameter on an attribute group enables you to manually define the order of attributes within an attribute group.

Note: 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.

Manually Sorted is configured as follows:

- Not checked means attribute child nodes of this object are ordered alphabetically in the tree hierarchy. The sequence cannot be manually changed by a user.

- Checked means attribute child nodes of this object can have their sequence manually changed in the tree hierarchy. Drag and drop attributes to modify the order of the group.

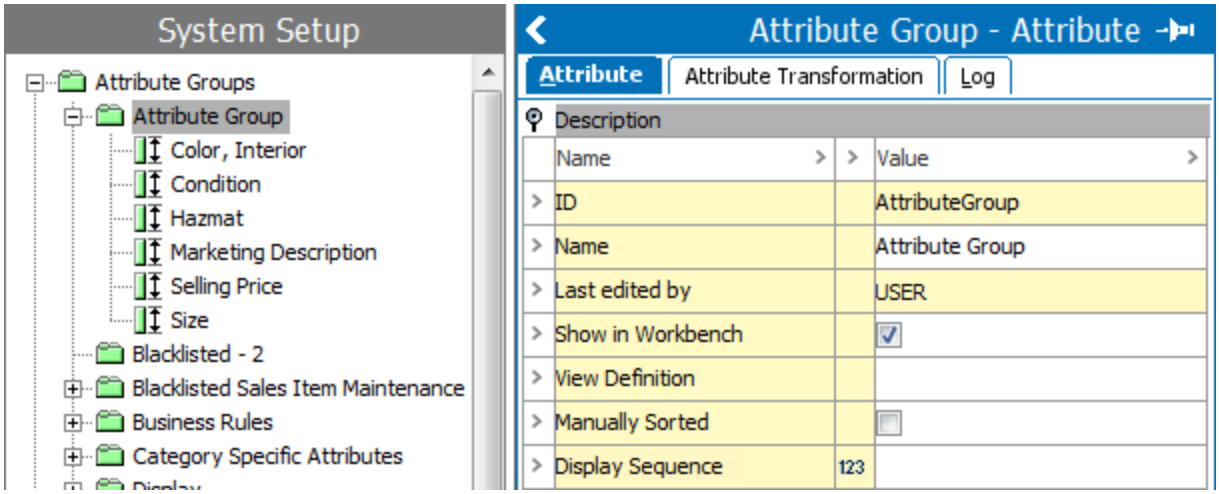
The 'Manually Sorted' order overrides any other attribute ordering where the attribute group is used, e.g., in an attribute link, when viewing the values of a product below the attribute group flipper for this attribute group. If the attribute is also included in another attribute group, the ordering inside that attribute group is independent of the ordering within this group.

Creating a Customized View

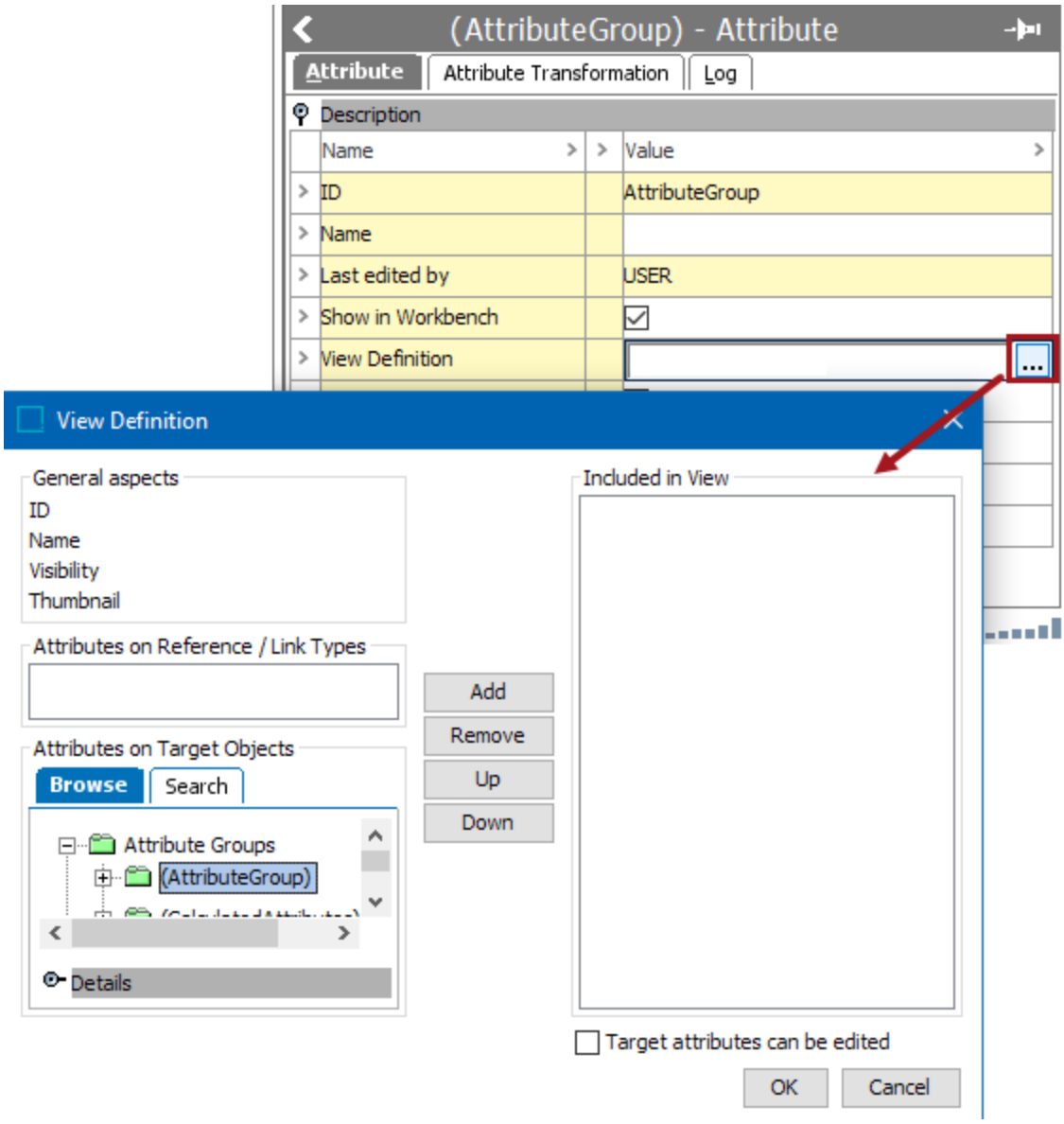
Attribute groups that contain reference types and product to classification link types can have their views defined. The customized view defines general aspects, meta data attributes and attributes on referenced objects. These referenced objects appear in the References editor on products, classifications,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



2. Check **Show in Workbench** if all reference types, product to classification link types, and attributes linked to the Attribute Group should appear on products, classifications, or assets. **Show in Workbench** is typically unchecked if a privilege needs to be applied to the attribute group to prevent the objects from appearing on products, classifications, and assets.
3. In **View Definition**, click into the field to display the ellipsis button (...). Then click the ellipsis button (...) to specify the appearance of reference types and product to classification link types linked into the attribute group. The **View Definition Dialog** appears.



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

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

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

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

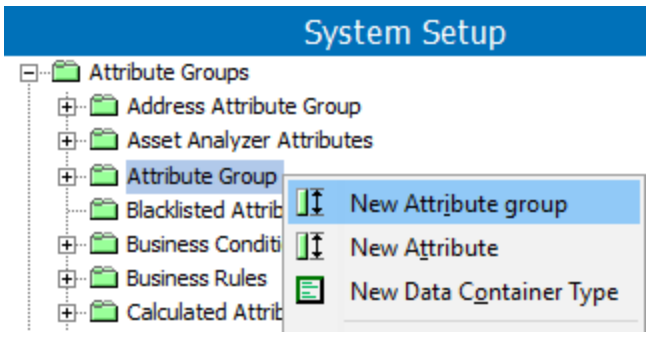
Attributes on Target Objects	Definition
Description Attributes or Specification Attributes	In Attributes on Target Objects you can search or browse for attributes. When you add an attribute to the view from this field, it displays the attribute value in the reference editor. Each reference type establishes 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.

5. 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.

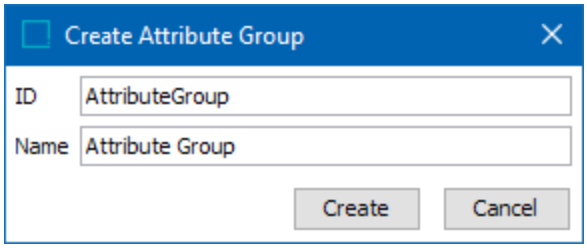
Creating an Attribute Group

Attribute groups allow you to group attributes, reference types, link types, and data container types according to their characteristics. Create attribute subgroups to limit the number of main groups.

1. In System Setup, expand Attribute Groups, and then select the relevant attribute main group folder.
2. Right-click and choose **New Attribute Group** to display the 'Create Attribute Group' dialog.



3. On the 'Create Attribute Group' dialog, enter an ID and a name for the attribute group.



4. Click **Create**. The new attribute group displays 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.

Attribute Links

Attributes linked to product and classification nodes are displayed on the References tab of the product and classification of 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 recommended 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 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

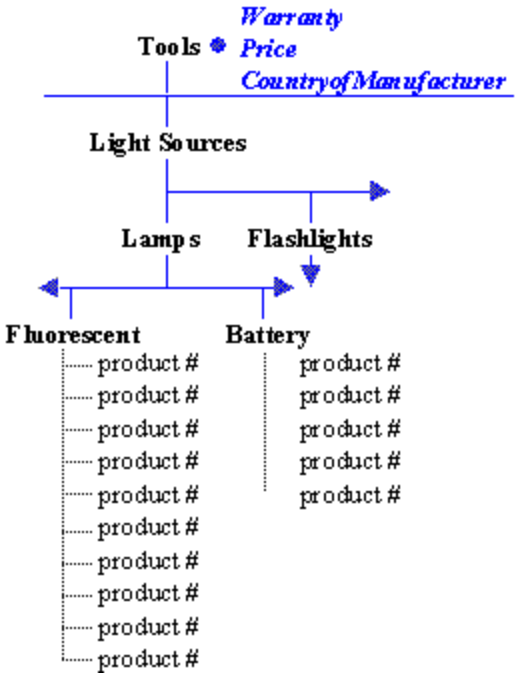
An attribute that is linked to a node in the hierarchy is inherited to the nodes below. This way, attributes are shared among nodes.

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, as shown in the image below, '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.



Linking Orphan Attributes

Orphan attributes are specification attributes that although have a value for a product or classification, are not specifically linked to that product or classification. To make an attribute no longer an orphan, the attribute must be linked to the product or classification for which it has a value. In the workbench, an orphan attribute displays in italicized text. In the Web UI, orphan attributes display with a warning indicator and explanatory text.

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 orphaned attribute.

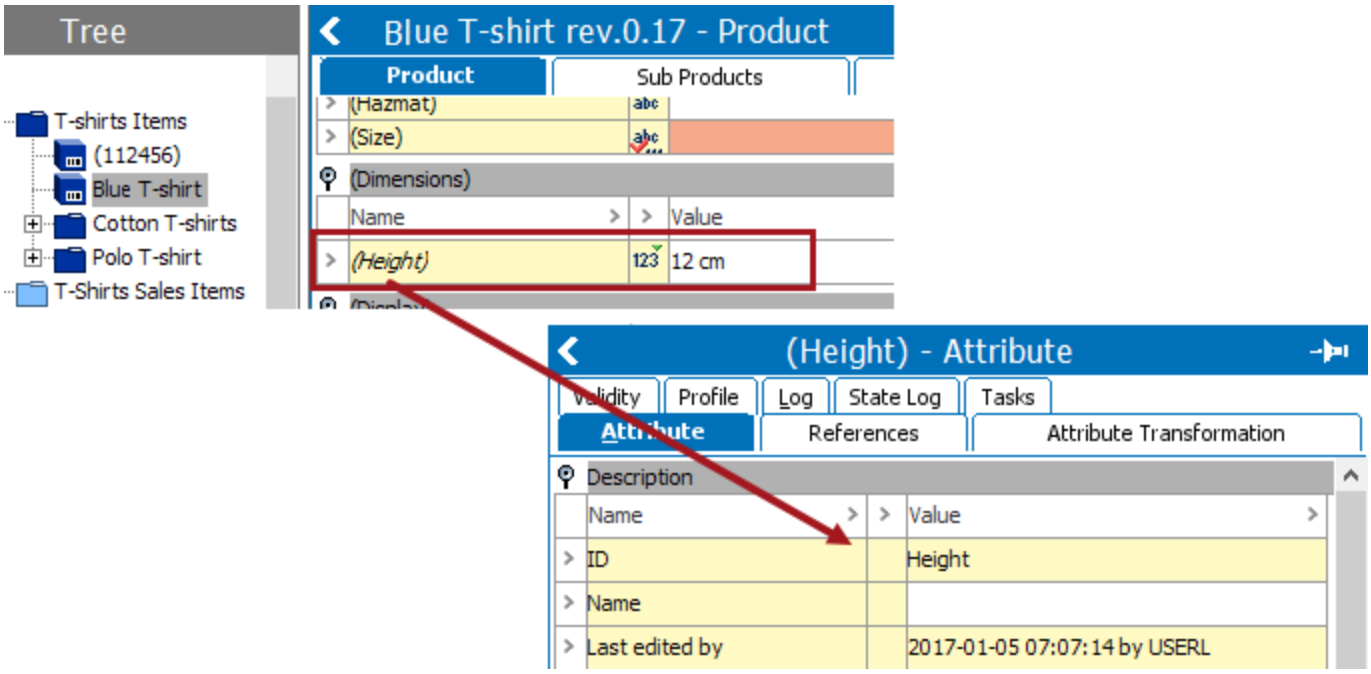
Linking an Orphan Attribute

Orphan attributes are linked in the attribute editor **References** tab. It is also possible to link orphan attributes from the product editor **Product** tab and the classification editor **Classification** tab.

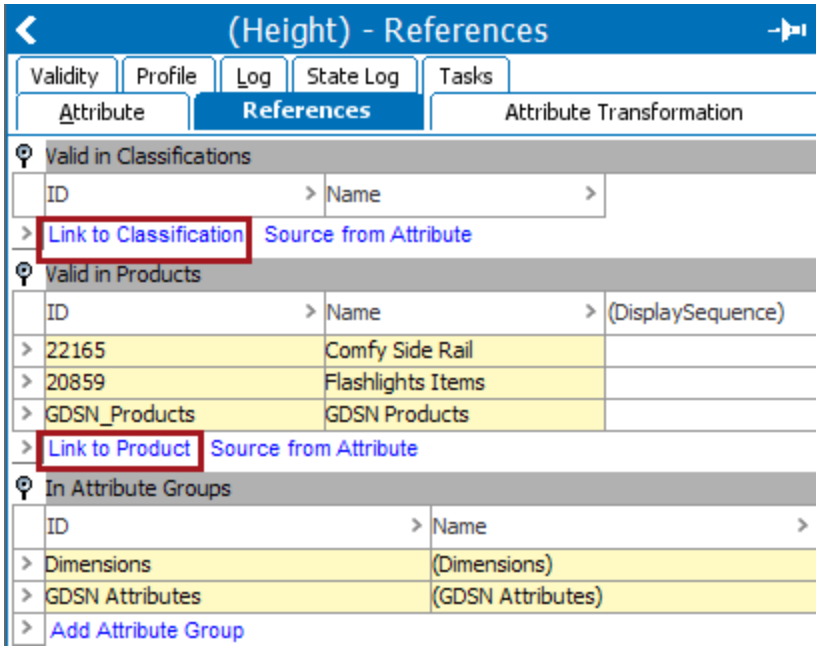
1. In Tree, select the product with the orphan attribute, or alternately, in the Search Result tab, search and select the attribute.

   The **Product** Editor appears.

2. In the product editor, click the **Product** tab.
3. Click the orphan attribute to be linked. The attribute editor appears.

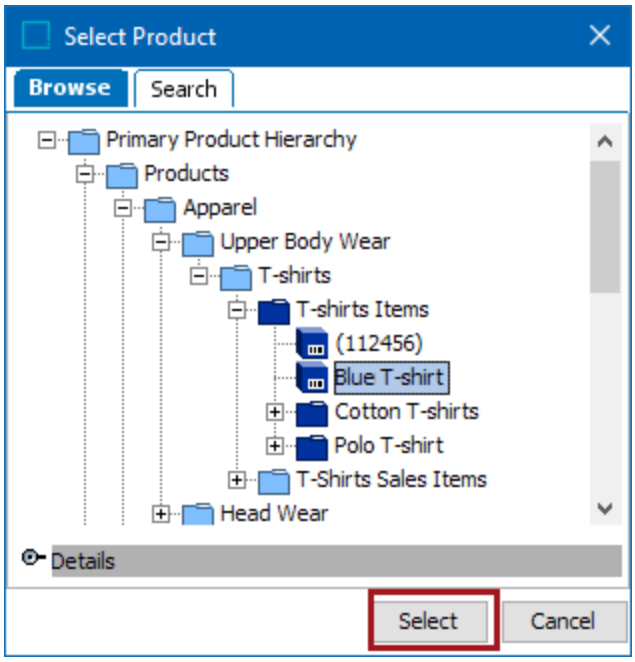


4. In the attribute editor, click the **References** tab.
5. Click **Link to Classification** or **Link to Product**.

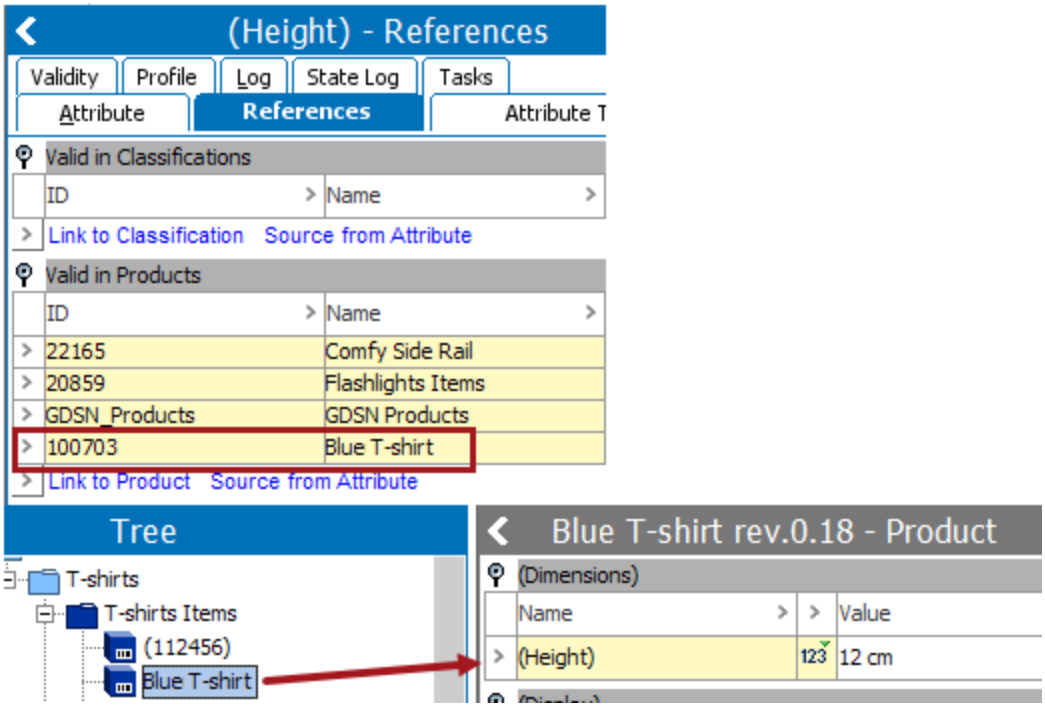


The **Select Classification** dialog box or **Select Product** dialog box appears.

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



The attribute displays under **Valid in Classification** or under **Valid in Products** flippers.



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.

For more information regarding linking specification attributes, see the **Linking Specification Attributes** topic.

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:

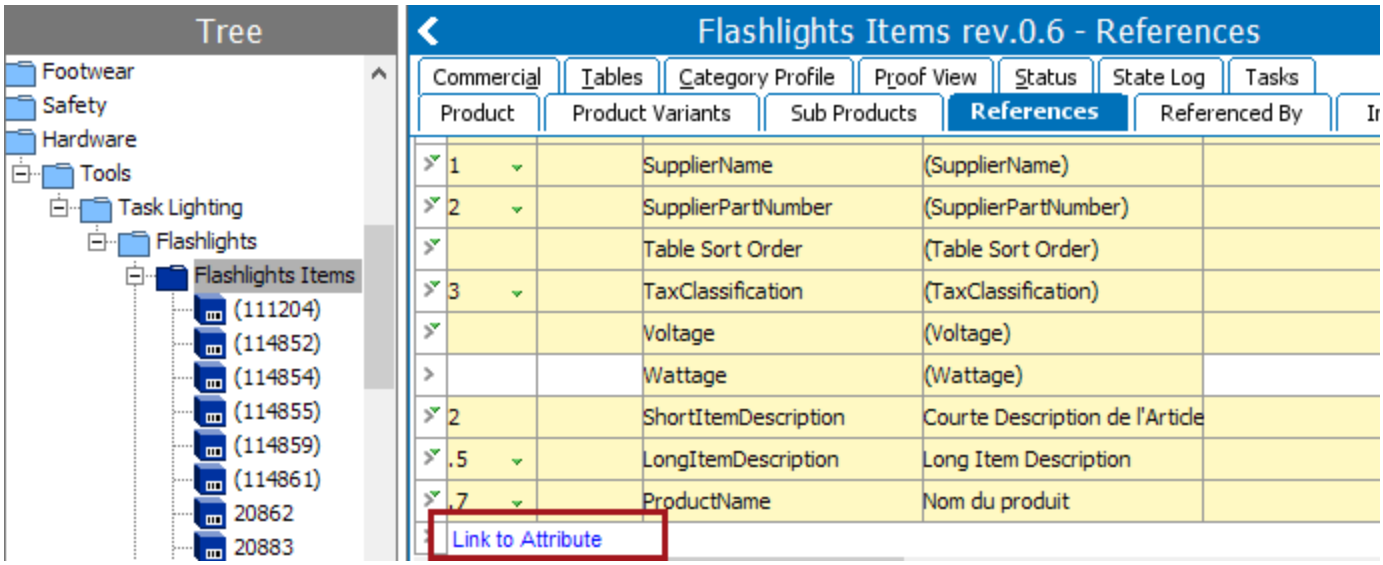
- Classifications
- Products

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

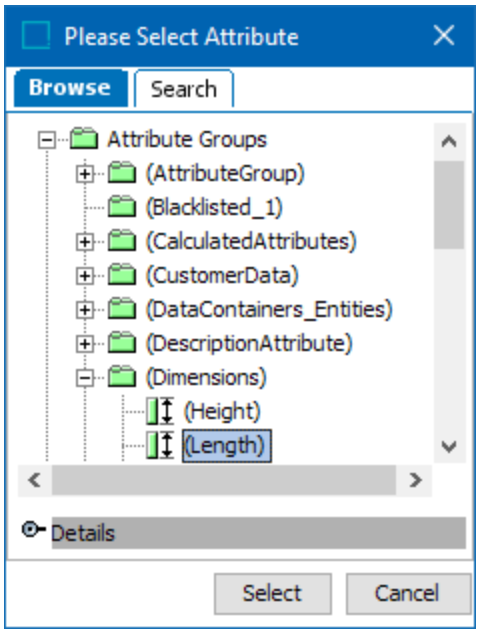
Note: If a specification attribute is not in the list of selectable attributes, it may not be available in the current context.

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**.

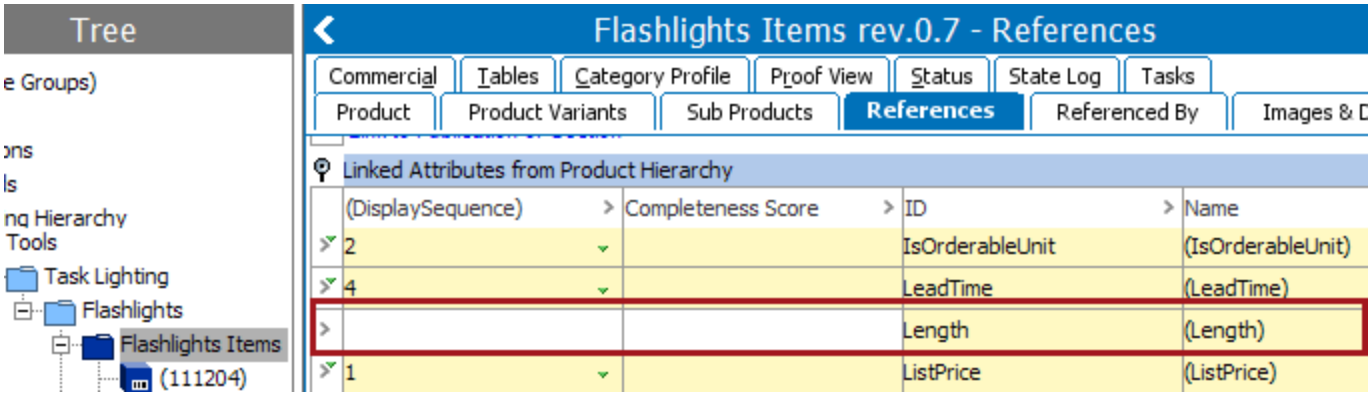


The **Select Attribute** dialog box appears.



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.

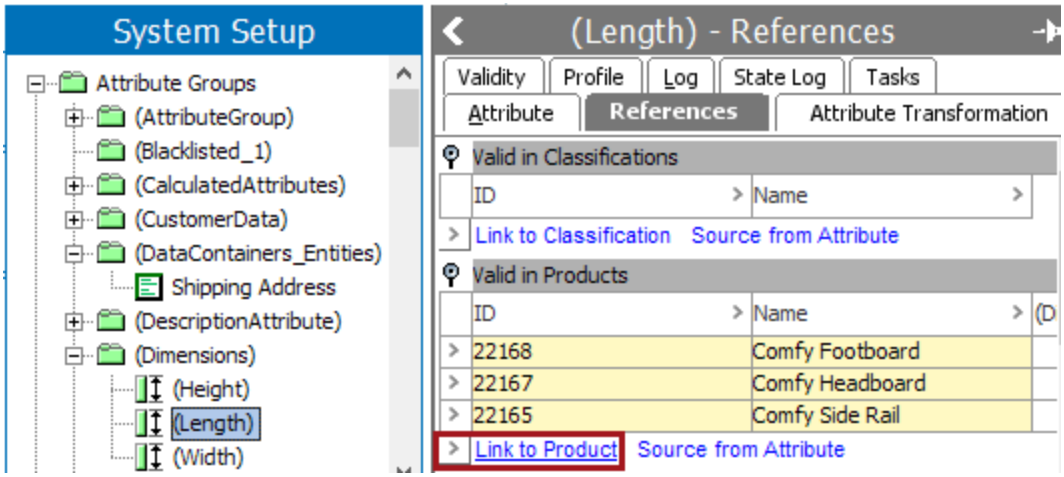


   One or more specification attributes have now been linked to a product or a classification node, and the attribute(s) will be available to the linked products and sub products.

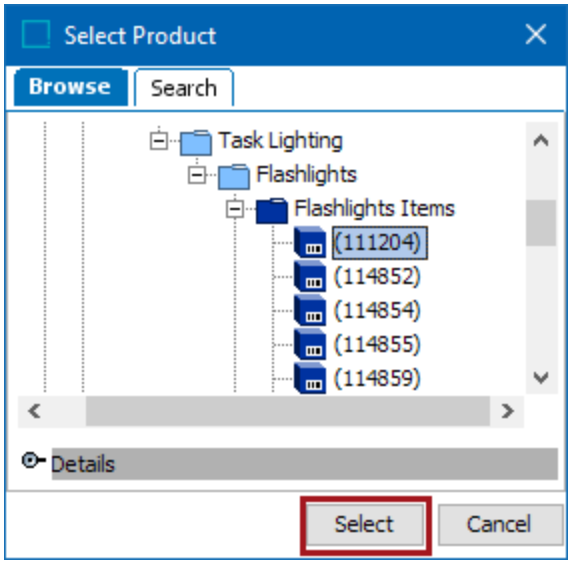
Linking Attributes to Products or Classifications

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.

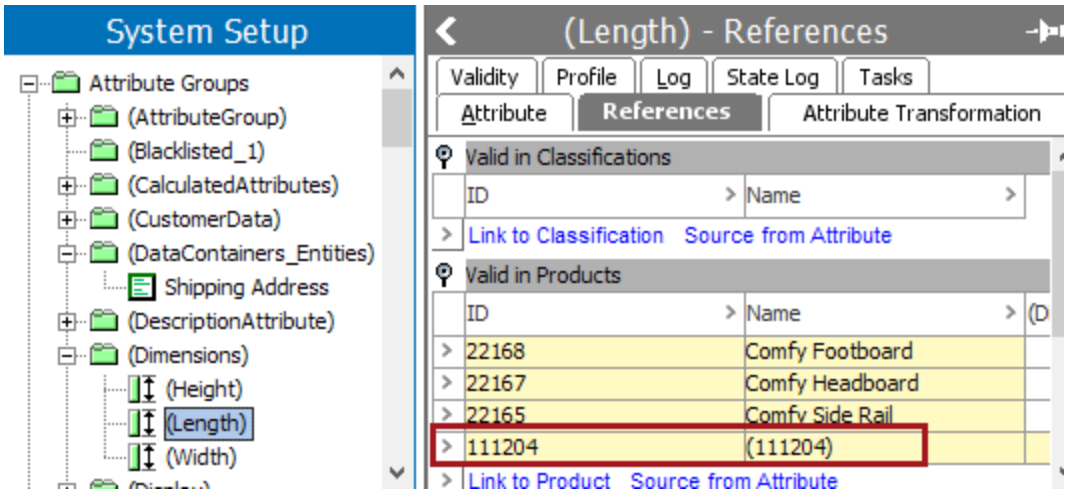
1. In System Setup, select the relevant attribute node, and click the **References** tab.
2. Under the 'Valid in Products' flipper, click **Link to Product** to link the object.



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**.



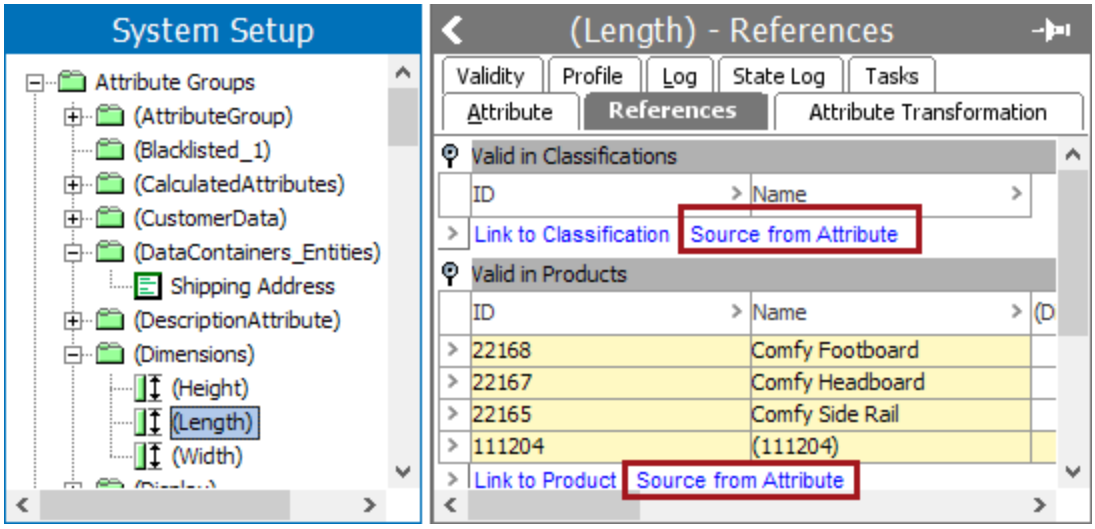
The selected product(s) will appear in the list of 'Valid in Products.'



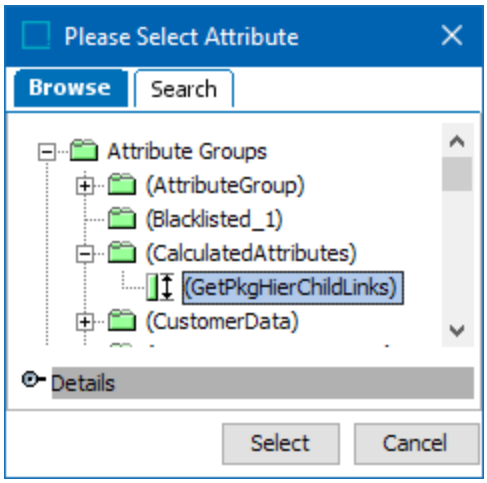
Note: While the maximum number of references to one object node is limited to 10,000 instances, this value can be changed; however, it is against Stibo Systems recommended practices to use this many references. Before changing this value, contact your Stibo Systems account manager or partner manager for assistance.

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.

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.

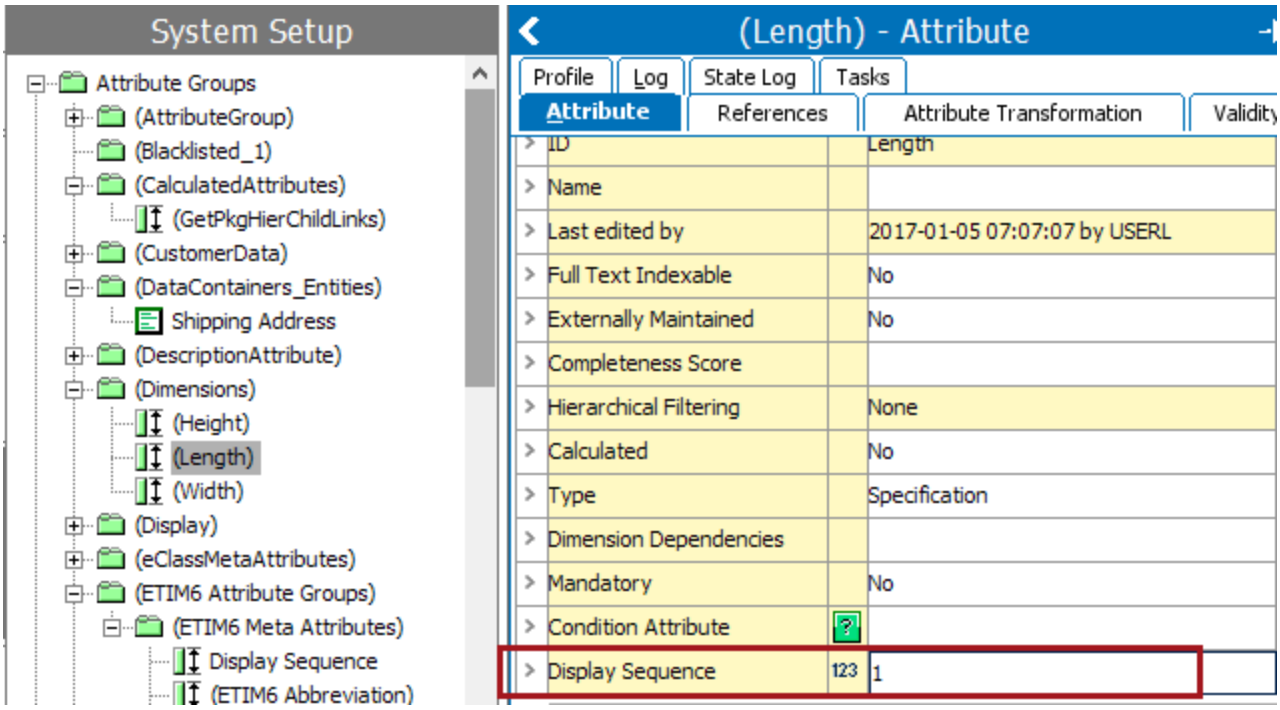
The attributes shown in the product editor can be sorted according to the order set in the 'Display Sequence' attribute. This can be done in two ways:

- 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.

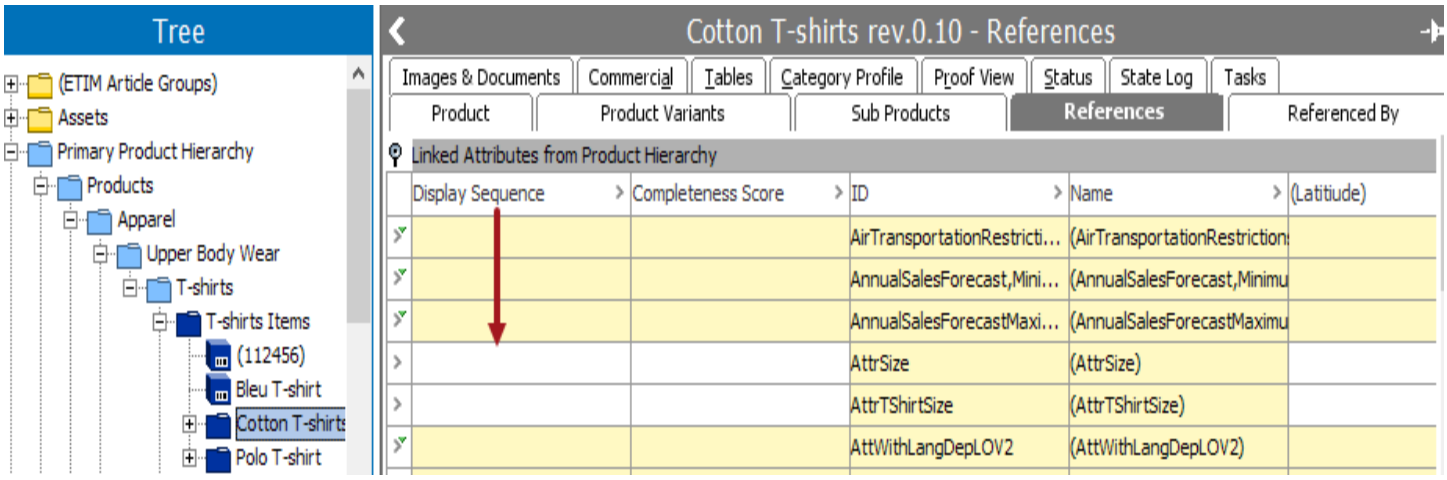
Setting a Display Sequence Number on the Attribute

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.



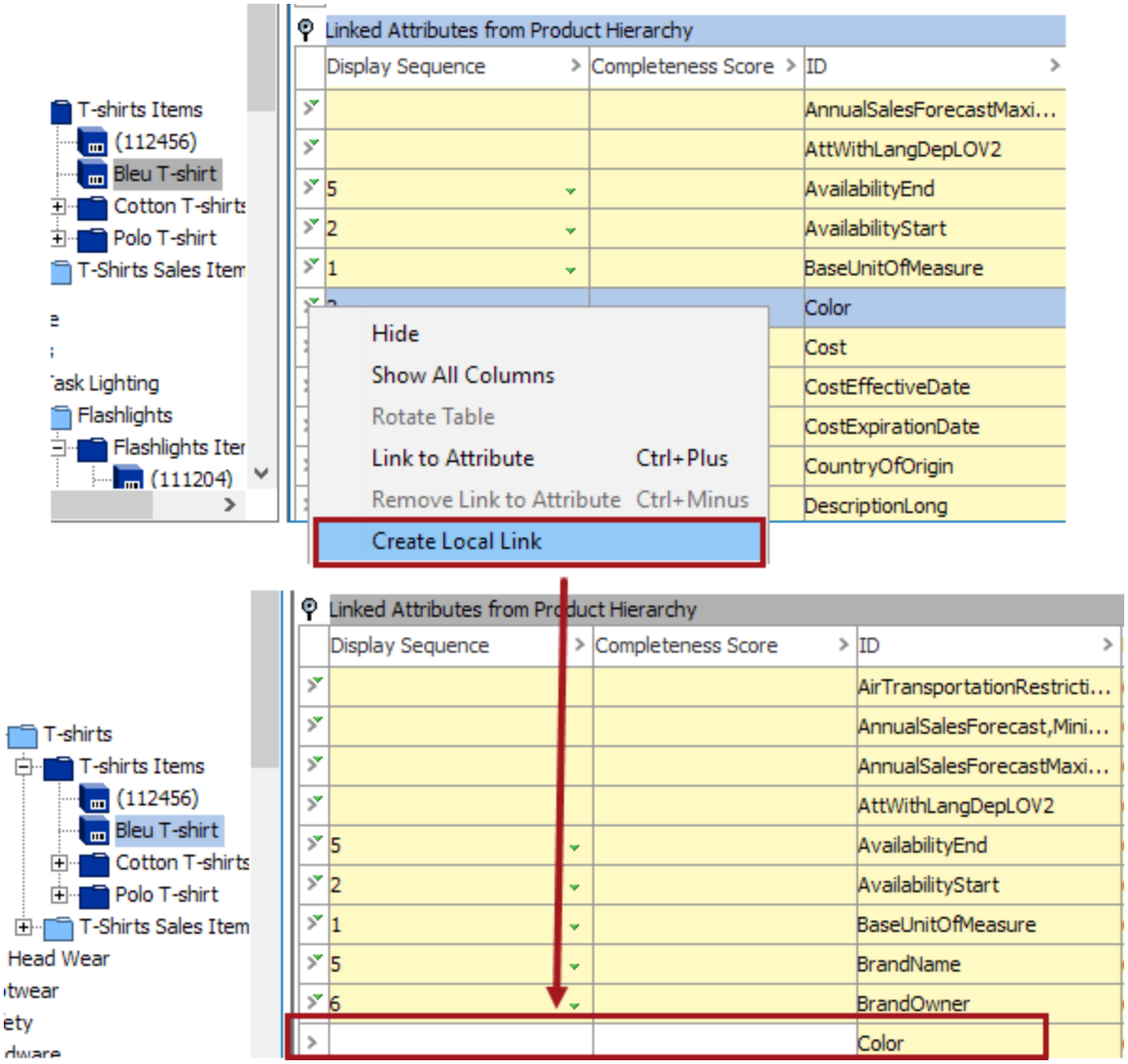
Setting a Display Sequence Number under Linked Attributes

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



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.



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

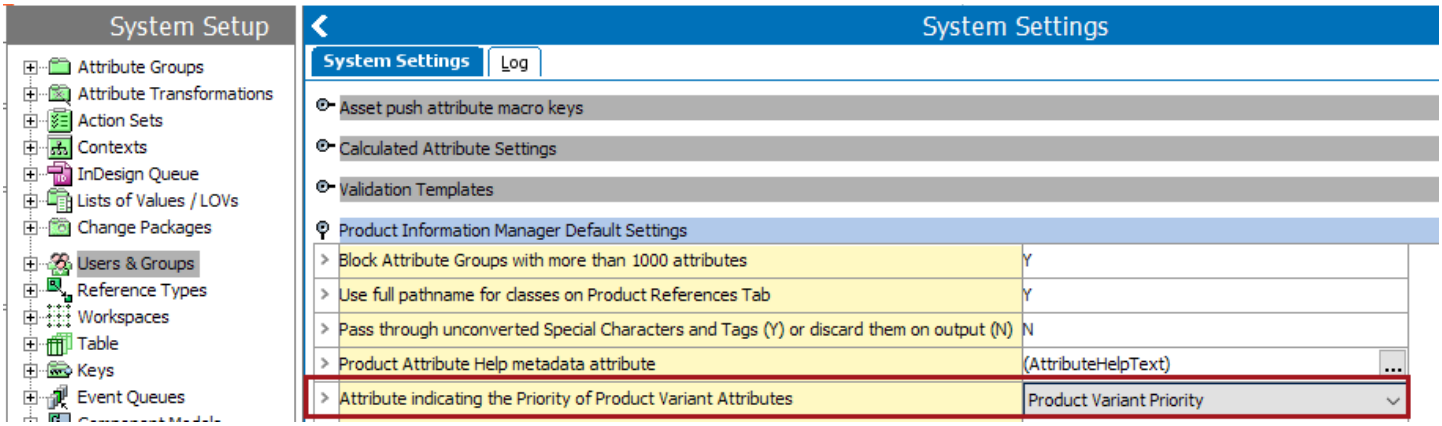
Important: Information about the Display Sequence of attributes is exported in an XML export. If the Display Sequence number is set on the attribute itself, the XML Export will contain the Attributes with Display Sequence attribute value. If the Display Sequence number is set on the Attribute Link, the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

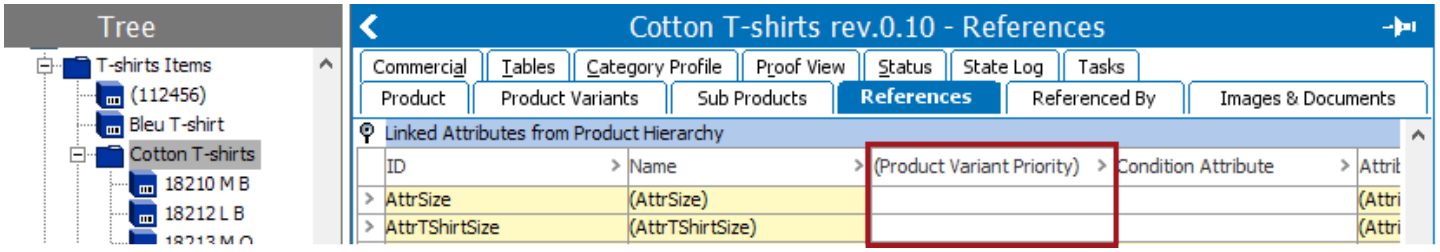
To check if it is selected go to System Setup > Users & Groups node > 'Product Information Manager Default Settings' flipper > **Attribute indication the Priority of Product Variant Attributes** field. In the field make sure that attribute set up to be the 'Product Variant Priority' is selected from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

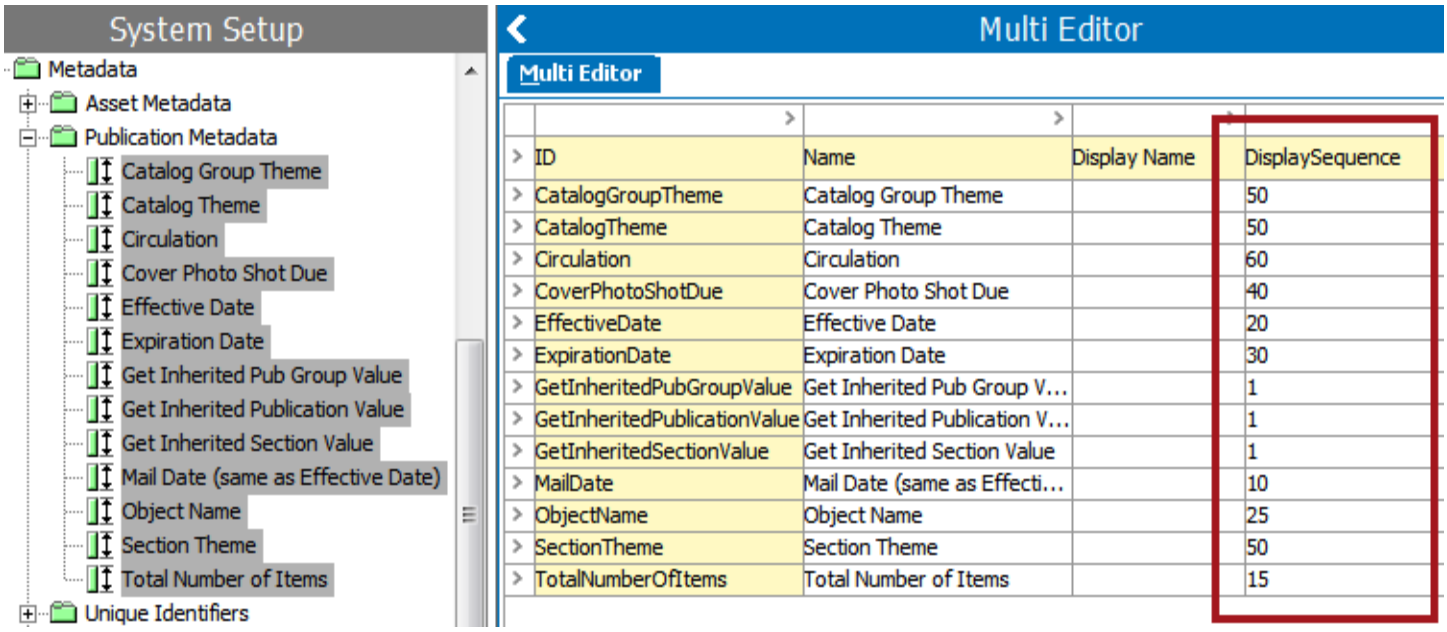
Display sequence values can be entered in the column related to the respective attributes.



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.

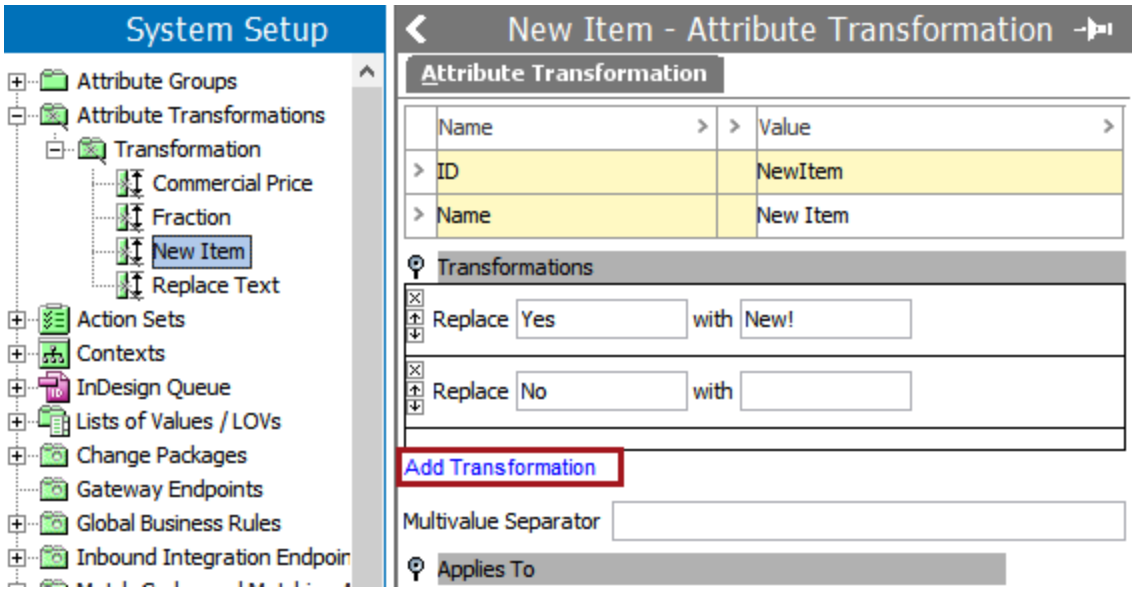
Because publication hierarchy objects do not use Reference Types, attribute sequencing cannot be set from a 'Linked Attributes' flipper on the References tab, as this is unavailable on publication hierarchy objects. Instead, metadata attribute sequencing must be set up at the attribute level. However,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



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.

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



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 or applying multiple transformations in the **System Setup / Super User Guide** documentation.
- **Maintaining an Attribute Transformation** includes information about configuration in the **System Setup / Super User Guide** documentation.

For details on each of the available transformation options, see the **Transformations** topic in the **Resource Materials** online help.

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.

Creating an Attribute Transformation

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

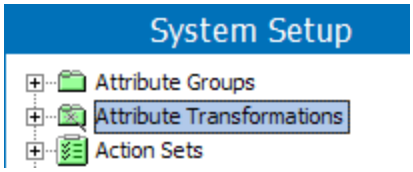
- **Create an Attribute Transformation Group** to organize the attribute transformations in your system.
- **Create an Attribute Transformation** to define one or more transformations to be used on your data.

These steps are defined below.

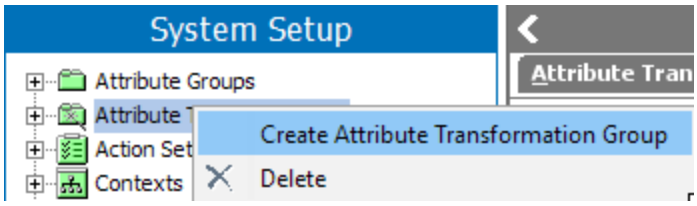
Create an Attribute Transformation Group

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

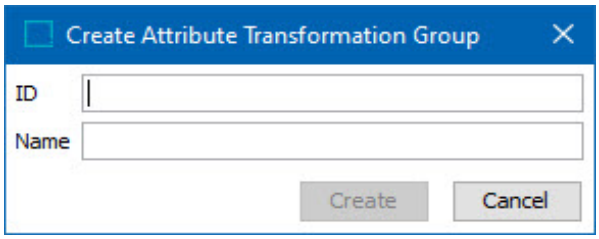
1. Go to System Setup, click on the **Attribute Transformation**, node.



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



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



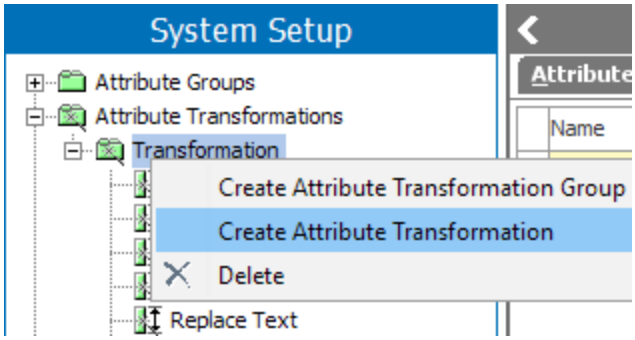
3. Type an ID and name for the attribute transformation group, and then click **Create**.

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

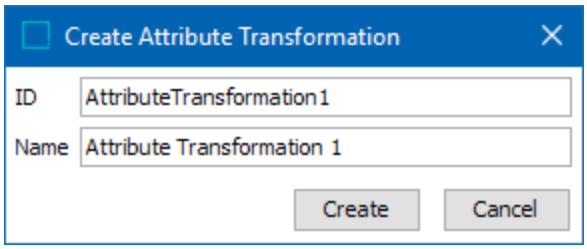
Create an Attribute Transformation

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**.

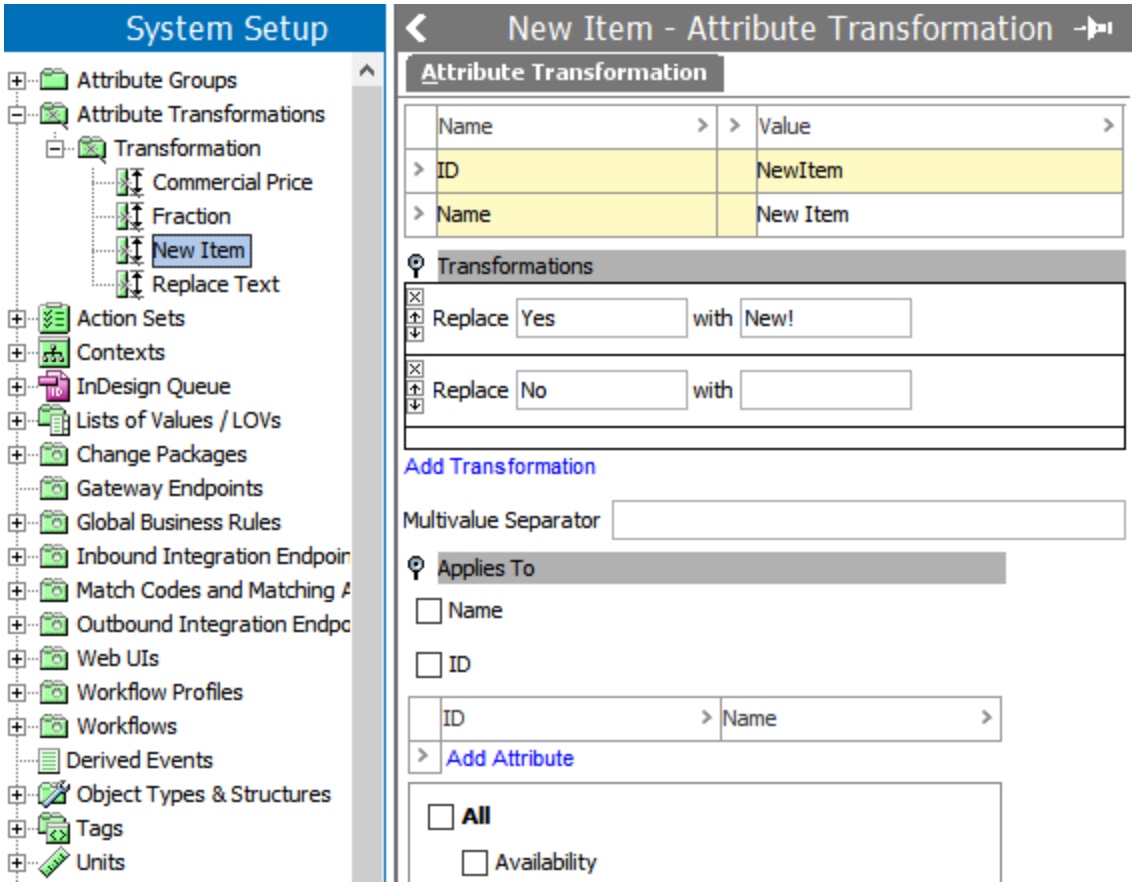


4. Once the attribute transformation is created, it will need to be set up with a transformation. Follow the steps in **Maintaining an Attribute Transformation** to complete the configuration.

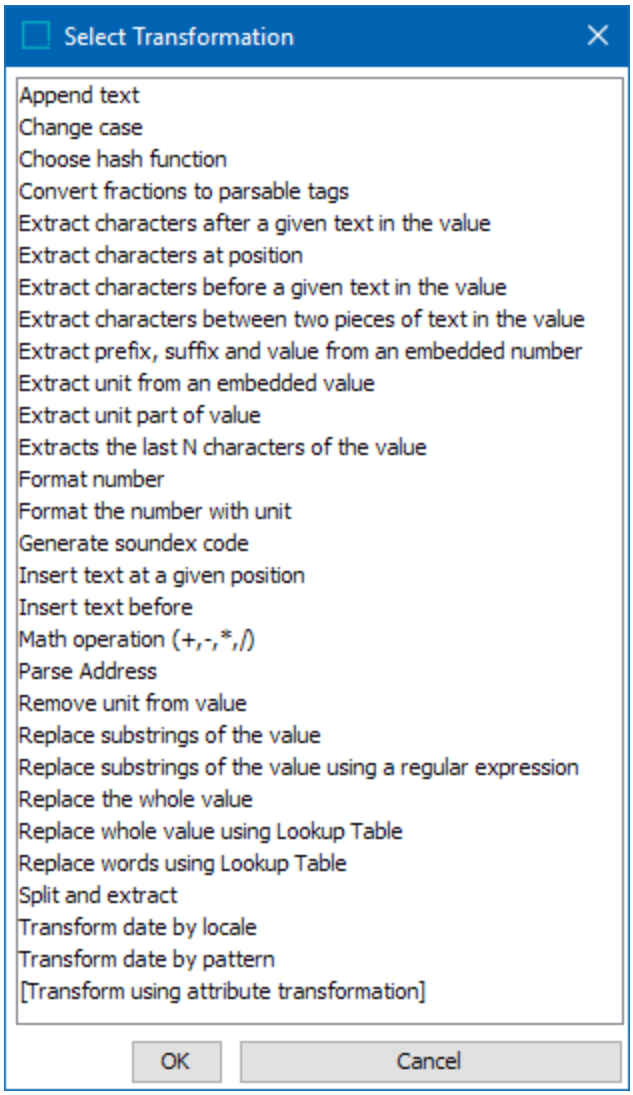
Configuring an Attribute Transformation

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.



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



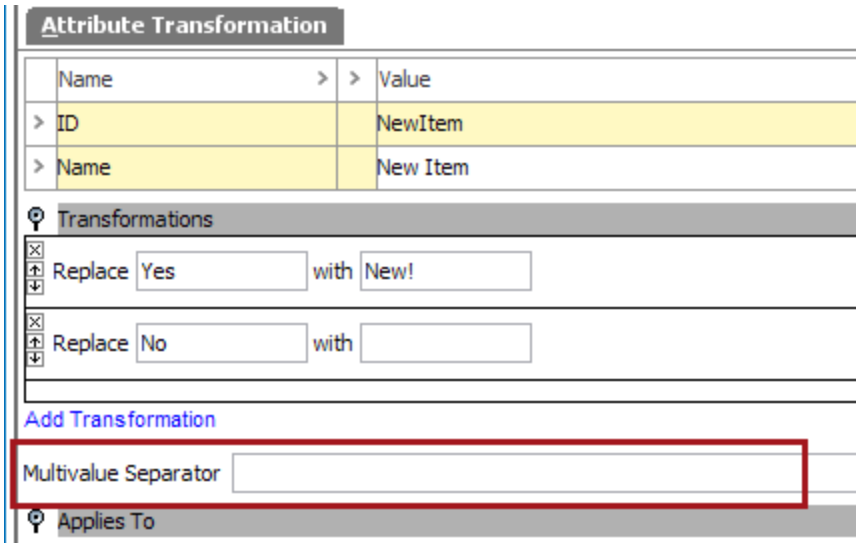
3. Select the relevant Transformations, and then click **OK**.

   For a description of the transformations, see the **Transformations** topic in the **Resource Materials** online help.

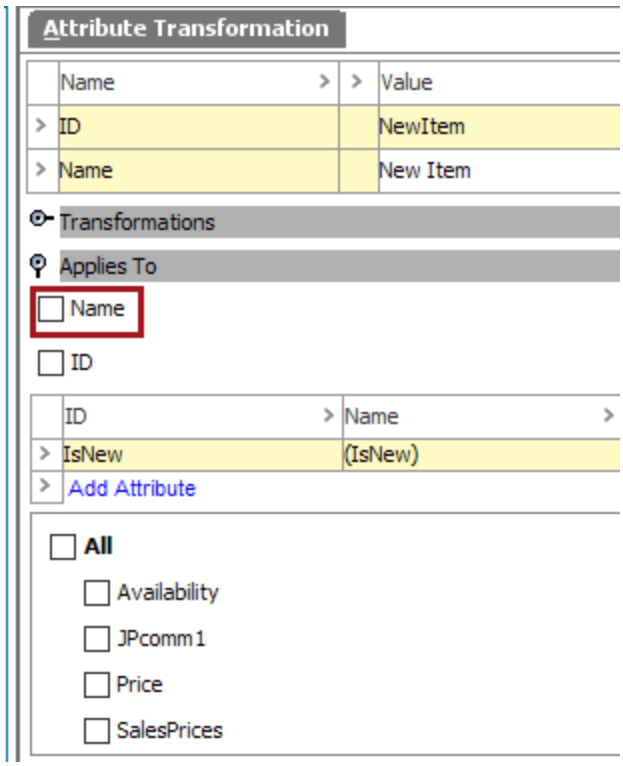
   Note: It is possible to create multiple sequential transformations as a rule, but you do have to be careful about the order in which the transformations are applied.

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.



5. 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.

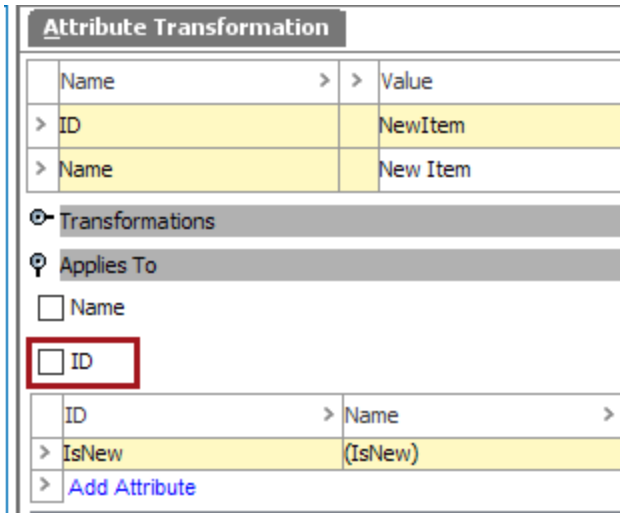


The transformation can be used in:

- 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 the **Applying Attribute Transformations to Tables** topic in the **Tables** documentation.

- InDesign on product templates. The product template can be set up to include the transformation, when an Object Name is mounted. See the **Attribute Transformations in InDesign** topic in the **STEP'n'design** documentation.

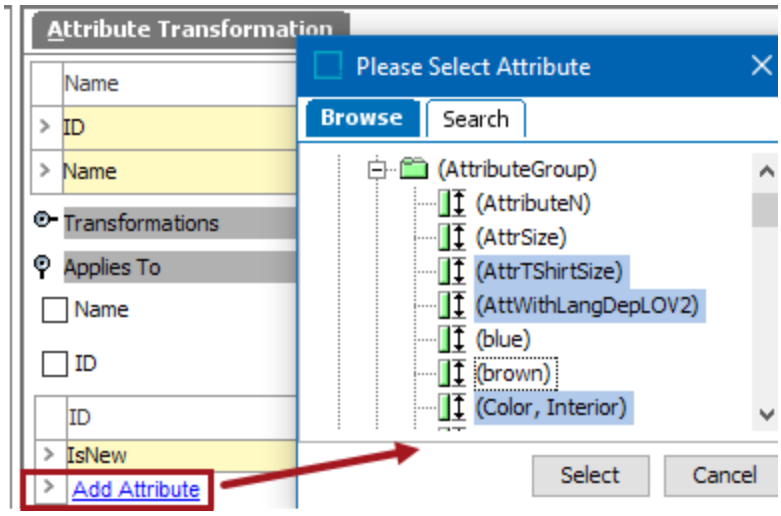
6. Under the 'Applies To' flipper, check the **ID** checkbox if it should be legal to use the Transformation on object IDs. For example, an object ID could be a product ID.



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 the **Applying Attribute Transformations to Table Content** topic in the **Tables** documentation.
- InDesign on product templates. The product template can be set up to include the transformation, when an Object ID is mounted. See the **Attribute Transformations in STEP Publisher** topic in the **STEP Publisher** documentation.

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.



The transformation can be used in:

- Tables on columns or rows where attribute values are extracted. The transformation can be applied to columns or rows that list attribute values. For more information about applying attribute transformations, see the **Applying Attribute Transformations to Tables** topic in the **Tables** documentation.
- InDesign on product templates. The product template can be set up to include the transformation when an attribute is mounted. See the **Attribute Transformations in InDesign** topic in the **STEP'n'design** documentation.

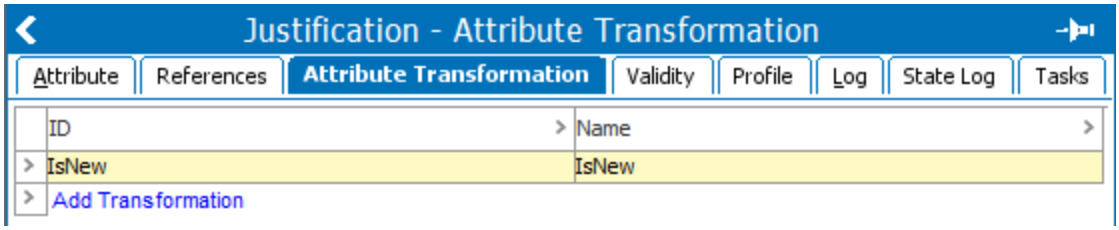
8. Under the 'Applies To' flipper, select a Terms List to legalize the transformation on a terms list.

   The transformation can be used in:

   - Tables on columns or rows where the contents of the specified terms list(s) are extracted. For more information about applying attribute transformations to tables, see the **Applying Attribute Transformations to Tables** topic in the **Tables** documentation.
   - InDesign on product templates. The product template can be set up to include the transformation when contents of the specific terms list(s) are mounted. See the **Attribute Transformations in InDesign** topic in the **STEP'n'design** documentation.

   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.



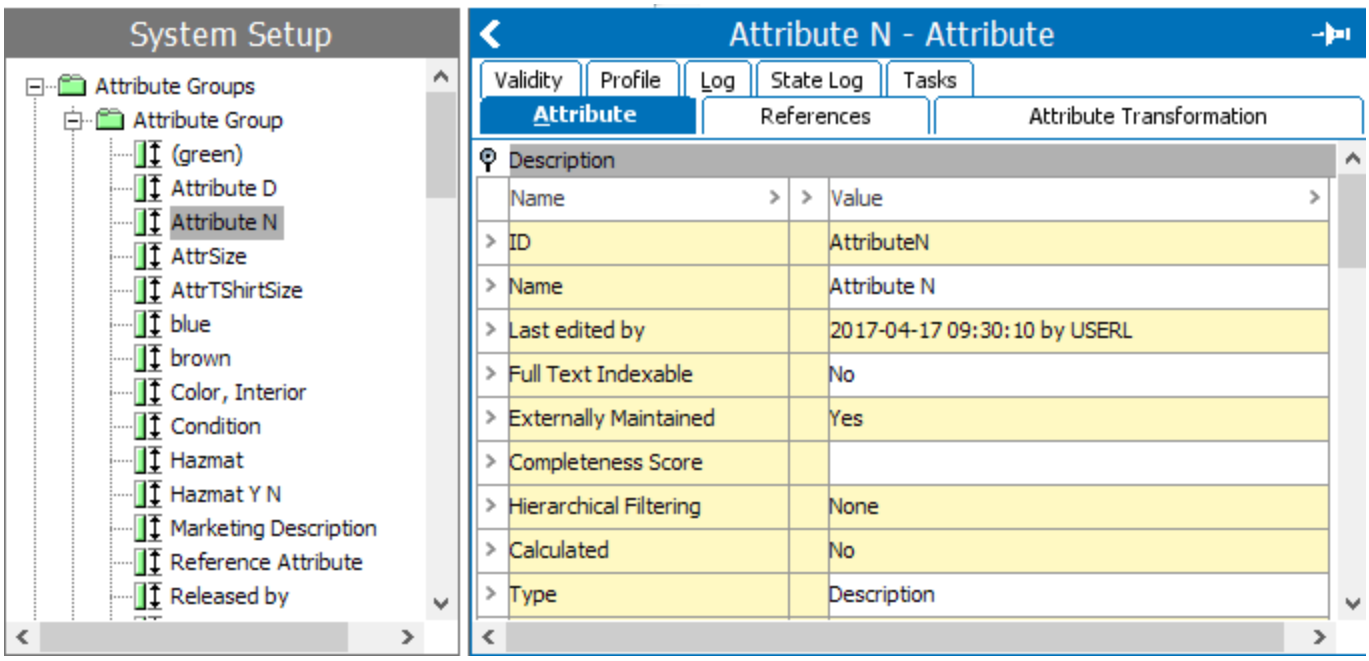
Attribute Types

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.

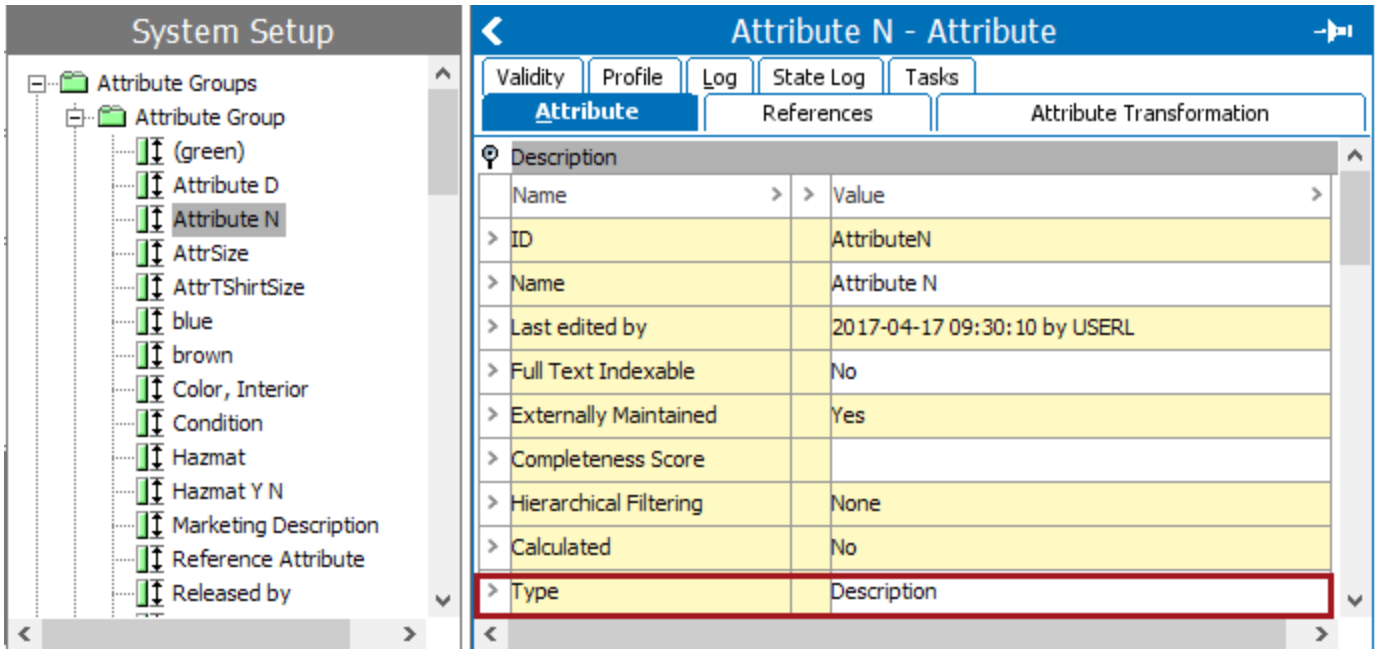
For more on attribute types, see the **Description Attributes** topic and the **Specification Attributes** topic in the **System Setup / Super User Guide** documentation.

To set the attribute type, follow the directions below:

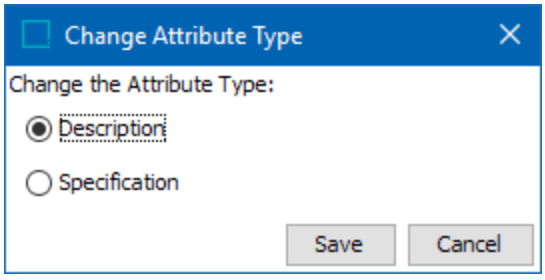
1. In System Setup, expand **Attribute Groups** node, expand the relevant attribute group, and then select the relevant attribute. The **Attribute** Editor appears.



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.

Calculated Attributes

Calculated attributes are a special type of attributes whose values are not written in the database, but are calculated on the fly. Calculated values are dynamic based on the context, version, product selected in the GUI or included in the export. Calculated values are generated when clicking on the object in GUI or when exported and are not stored in the database. The foundation for a calculated attribute is a functional programming language very similar to the language used for defining functions in Excel.

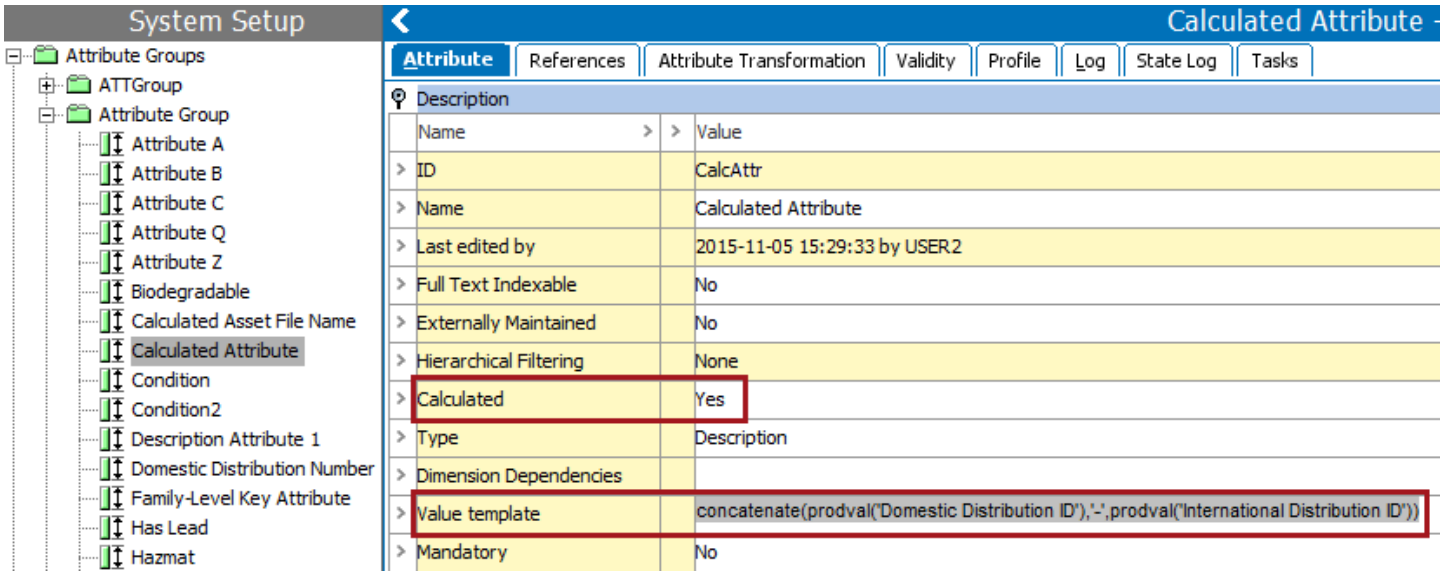
Note: Change Flags for events require that data is present in the database. Since calculated attribute values are not stored, no change flags are generated. For more information, see the **Limitations and Exceptions** section of the **Core Events** topic in the **System Setup / Super User Guide** documentation.

For assistance in determining if a calculated attribute is the most efficient way to meet your requirements, see the **Calculated Attribute Considerations and Limitations** 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.



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 and Limitations

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

Limitations

The following limitations exist because calculated attribute values are generated on-the-fly:

- Searching for the value of a calculated attribute is not possible since generating all possible calculations would cause substantial performance issues.
- Depending on the calculation formula, the revision history of an object does not necessarily display the calculated attribute value that was in effect at the time of the revision.

Deciding to Use a Calculated Attribute

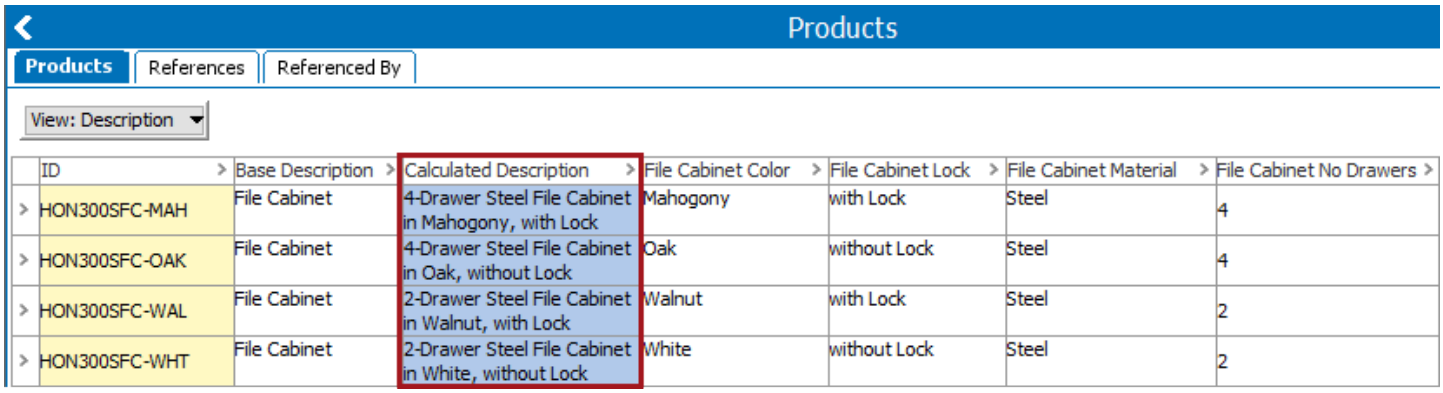
Although many functions can be accomplished using a calculated attribute, it is best to carefully consider when a calculated attribute is the most efficient way to address a requirement. Because a calculated attribute is recalculated each time it is viewed in an editor, and when it is exported (unless 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 recommended 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 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.



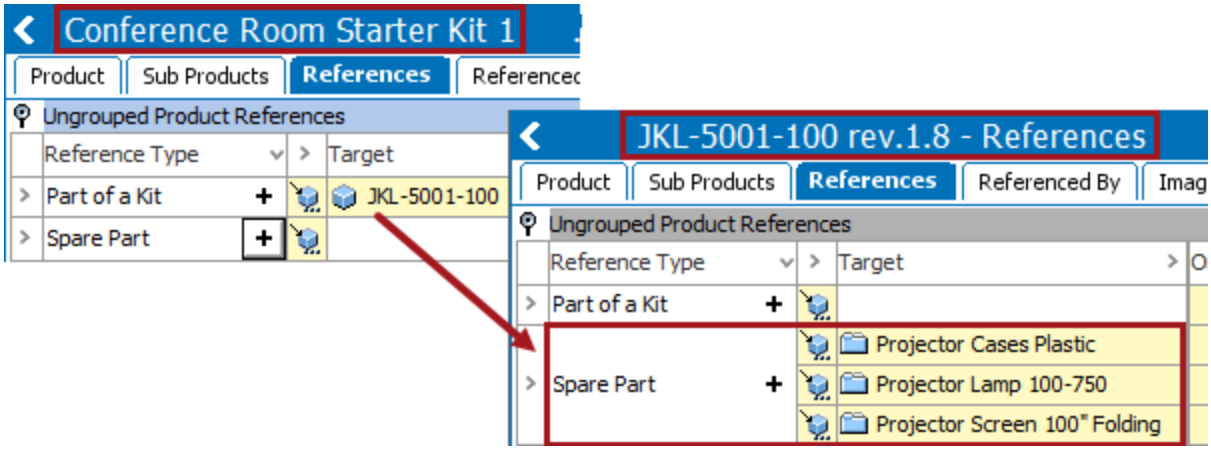
ID	Base Description	Calculated Description	File Cabinet Color	File Cabinet Lock	File Cabinet Material	File Cabinet No Drawers
HON300SFC-MAH	File Cabinet	4-Drawer Steel File Cabinet in Mahogony, with Lock	Mahogony	with Lock	Steel	4
HON300SFC-OAK	File Cabinet	4-Drawer Steel File Cabinet in Oak, without 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.

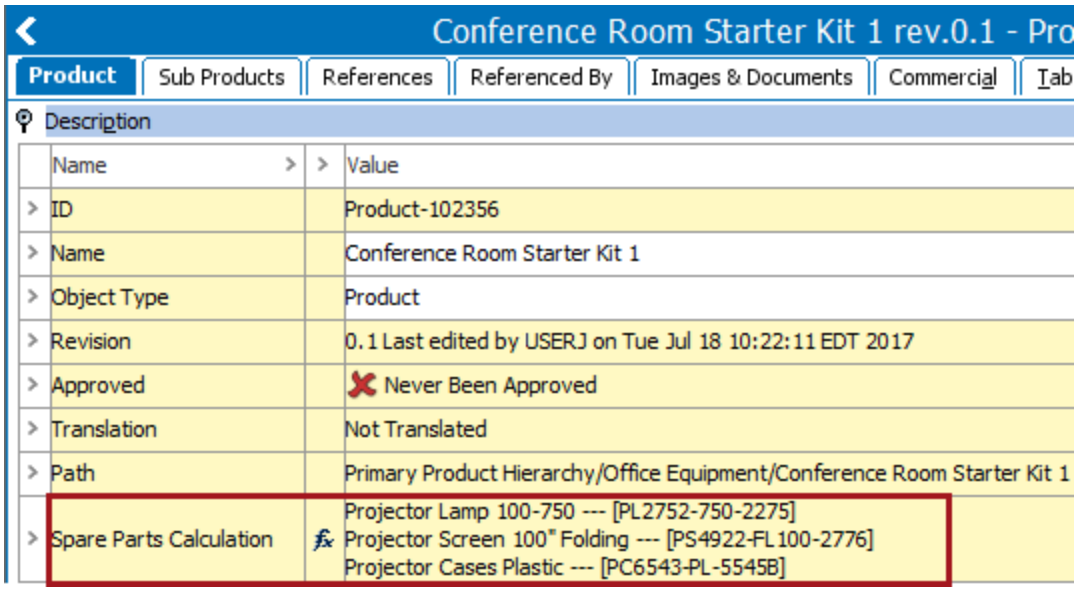
Fluid Data Example

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.



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.'



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.

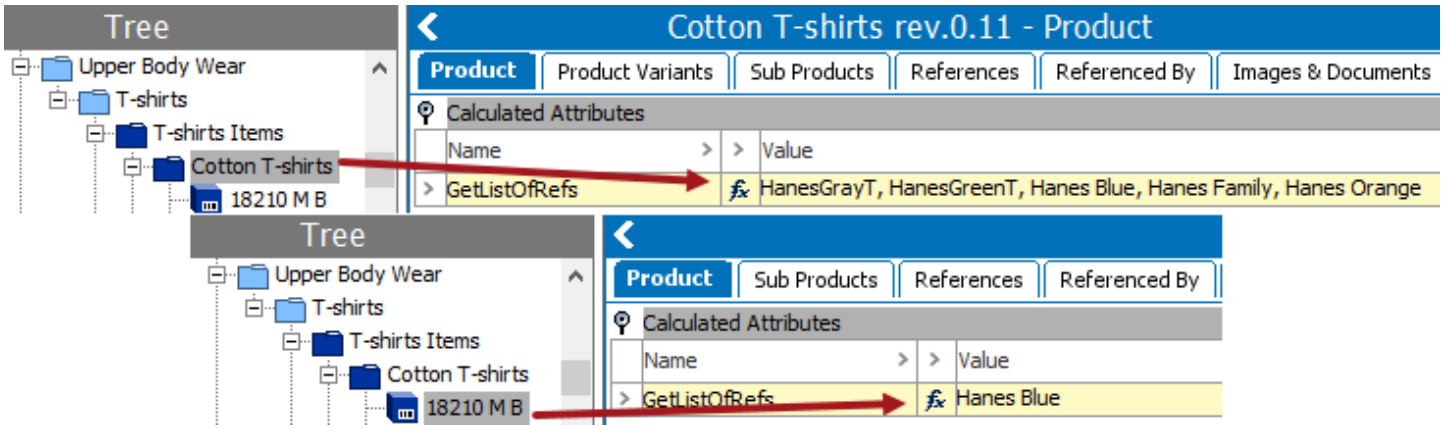
Setting up a Calculated Attribute

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.

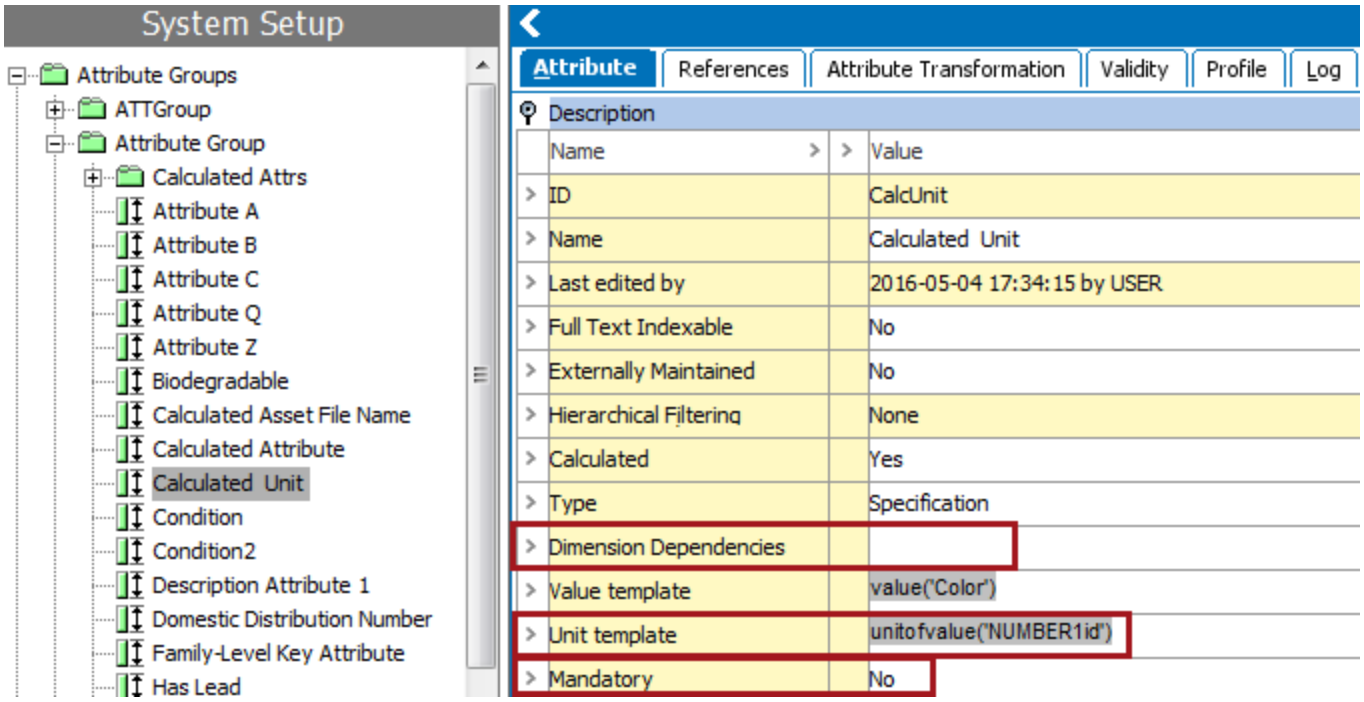
- **Calculated description attributes** can be made valid to objects based on the object type, just as any other description attribute. Thus a calculated description attribute can be made valid for an object, but as with all description attributes, the formula is not inherited. For more information, 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 following information identifies considerations when implementing a calculated attribute instead of a standard attribute.

Description Flipper

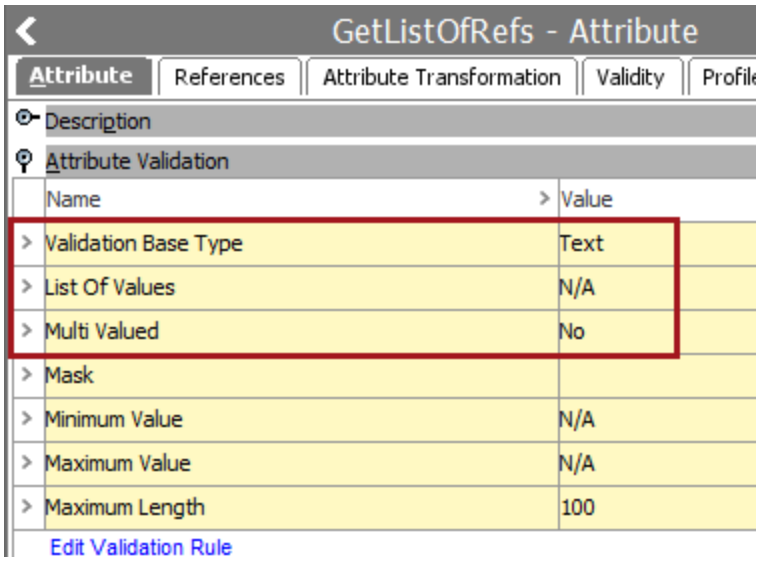
For all attributes, parameters describing the attribute are set in System Setup on the Attribute tab, under the Description flipper.



- **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, Numeric Text, or Numeric Text (exclude tags), 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.



- **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.

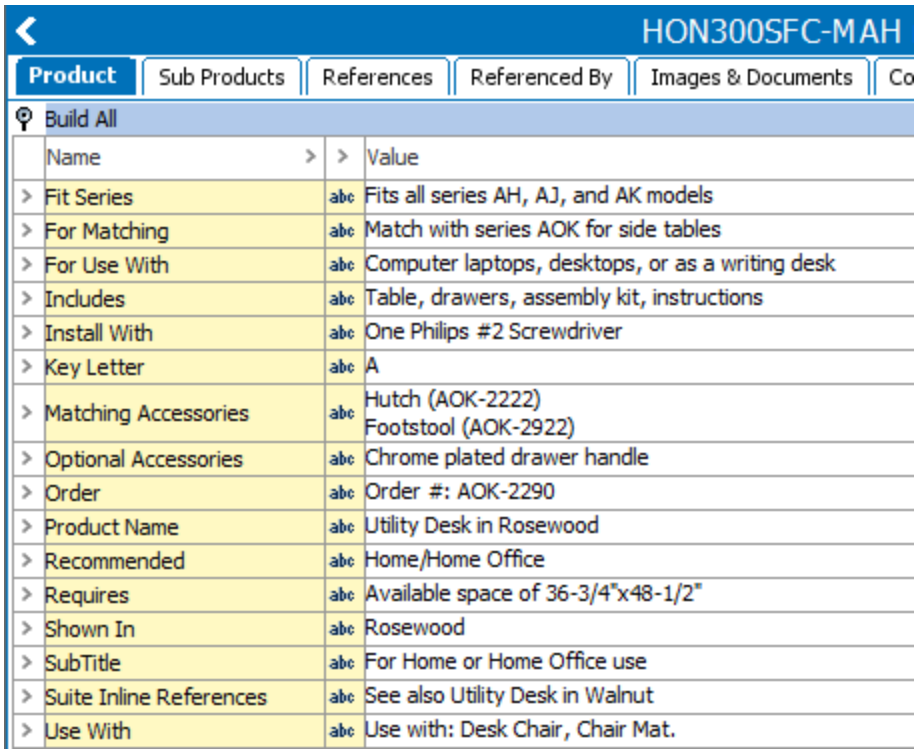
Functional and lengthy calculations without variables are harder to manage and troubleshoot later. To illustrate, the same calculation is written below; both without variables, and with variables.

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.



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

Example With Variables

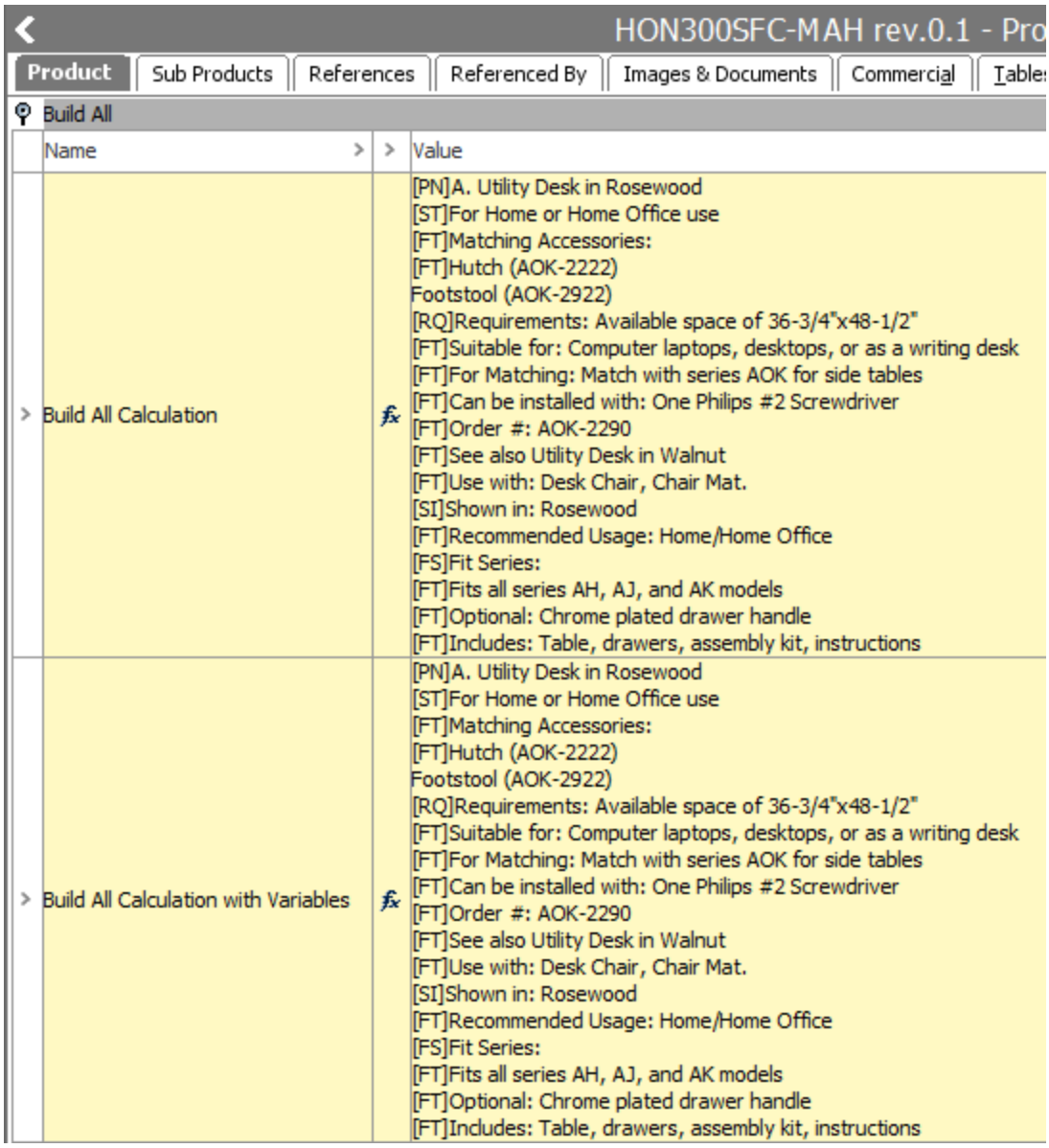
Using the function editor, the following value template is created and includes variables. Notice that each attribute that should be output is addressed in a single variable at the top. The final concatenate function then calls each of the variables and separates them with a hard return ("\n"). To make a change in one attribute, or to the formatting applied, 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
Images & Documents
Commercial
Tables
Build All
Name
Value
Build All Calculation
[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
[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.

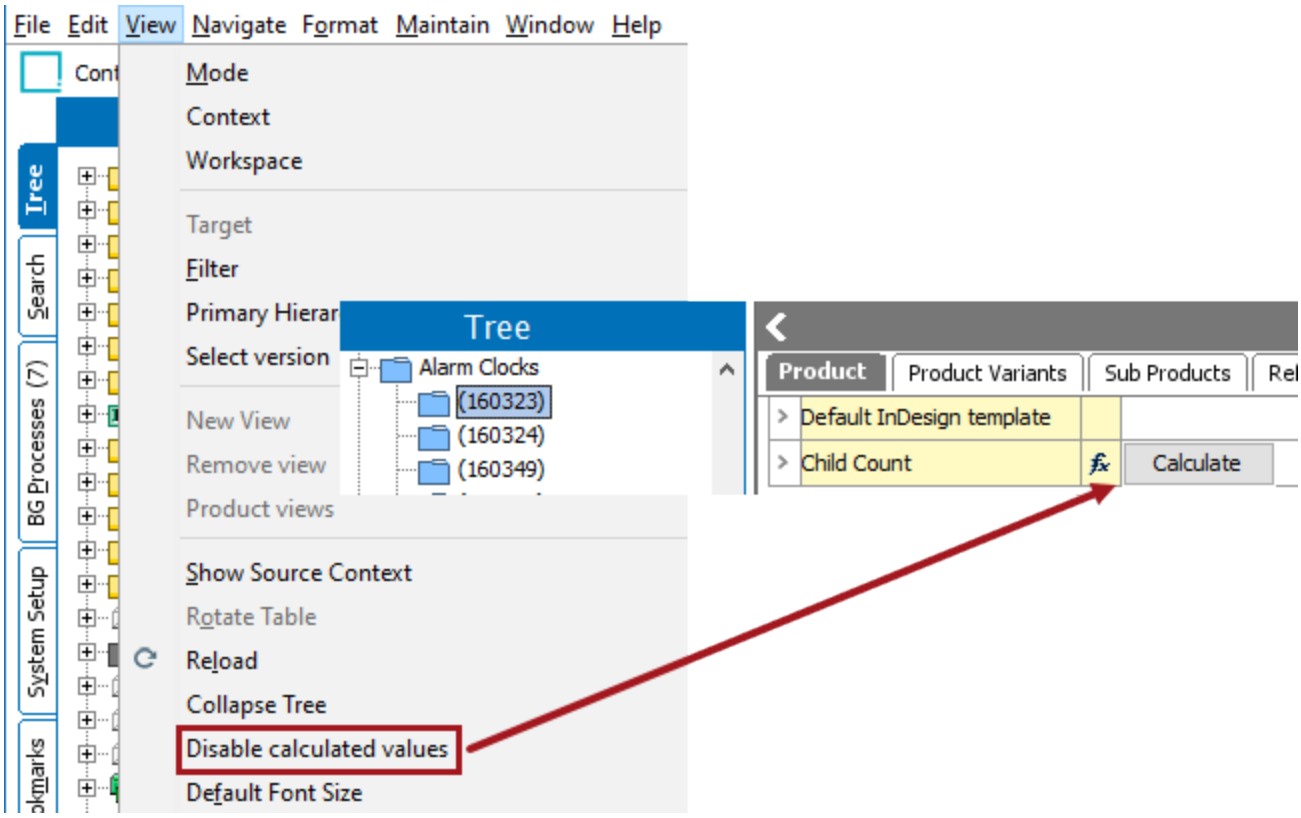
If these criteria are not met, then the calculation will be performed per context, which can increase the time to create the exported STEPXML file.

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

Limit Performance Penalties for Calculations

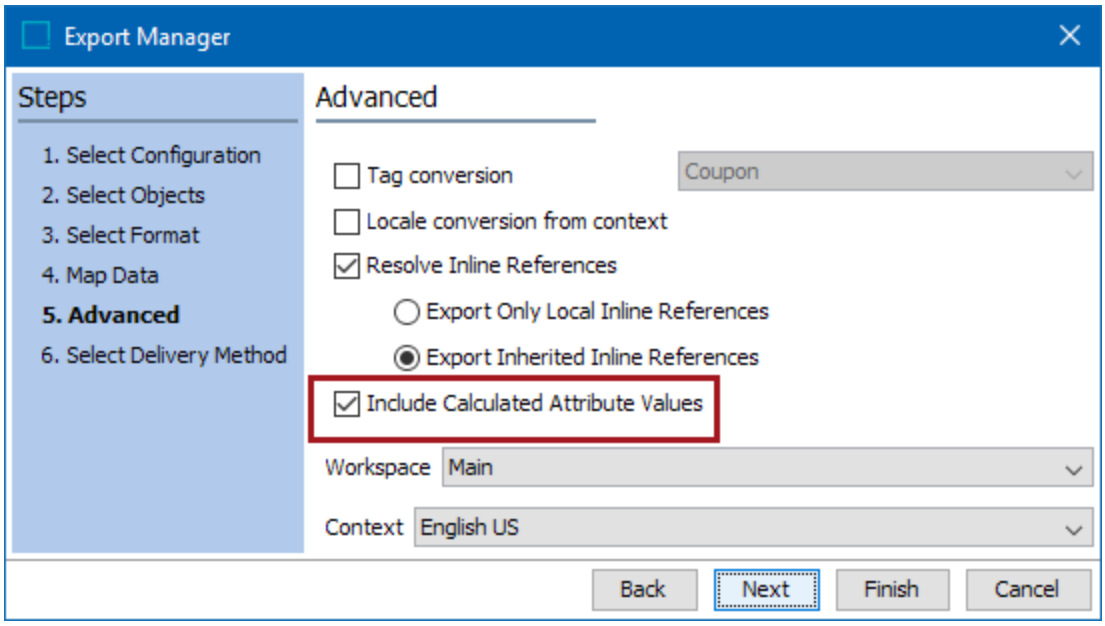
There is a performance penalty whenever the values are to be calculated. Complex calculations affect the time it takes to load editors and export objects.

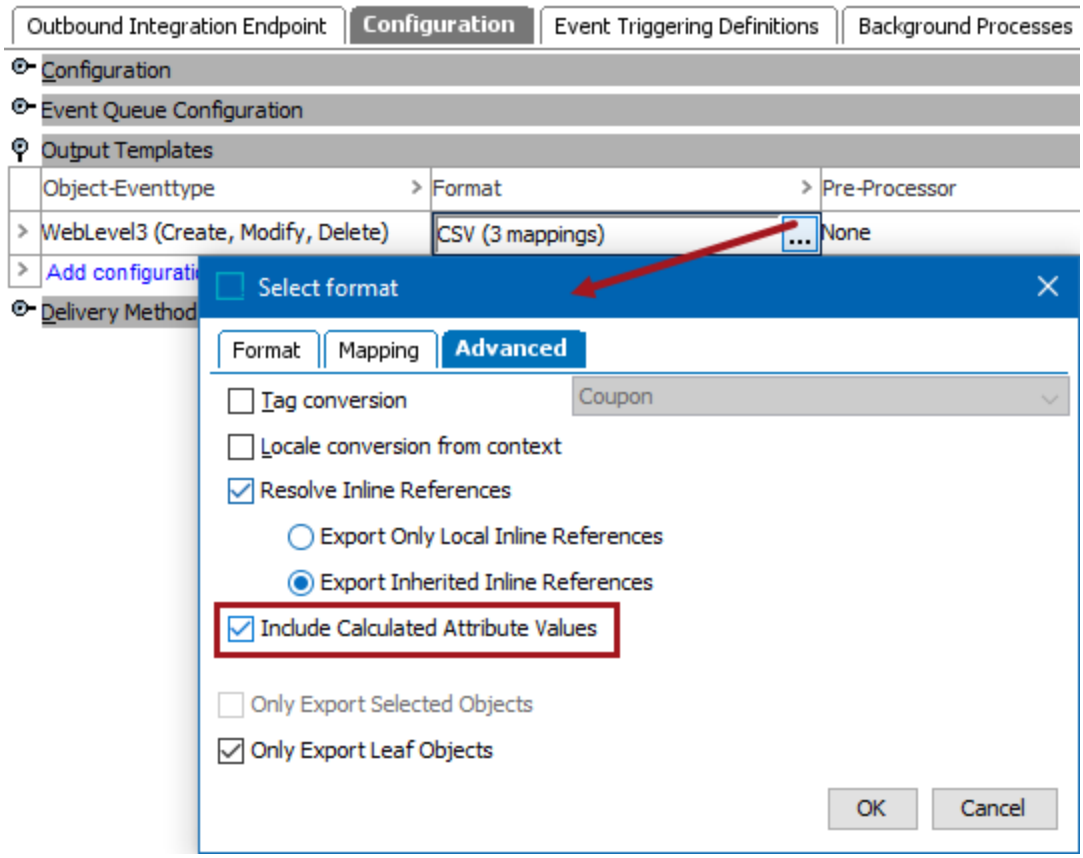
To limit potential performance penalties in editors, it is possible to disable the calculated attributes in workbench by setting View > **Disable calculated values**. Once disabled, calculated attributes are displayed with a calculate button. Clicking the button displays the calculated value for only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

Description	Value Template	Results
Box Shape: Displays if the bottom of the box is square or not.	if(value('Length') = value('Width'), 'Square', 'Not Square')	Not Square
Color: Compares the front and back colors of the current product and displays the result.	if(exact(value('Color Front'), value('Color Back')), 'Same color front and back', 'Different colors front and back')	Different colors front and back
Dimensions: Displays the dimension of the current product in the form	CONCATENATE(simplevalue('Height'), '	22 cm x 39

Description	Value Template	Results
'Height x Width x Length'	x ',simplevalue ('Width'), ' x ', simplevalue('Length'))	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.	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")	Size OK for cabin baggage
Size for special storage: The product must have either width or length greater than 40 cm, but not both.	IF (AND (OR (PRODVAL ("Length") >40, PRODVAL ("Width") > 40), NOT (AND (PRODVAL ("Length") > 40, PRODVAL ("Width") >40))), "OK - One and only one dimension >	OK - One and only one dimension > 40

Description	Value Template	Results
	40","Not OK - none or two dimensions > 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.	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")	Too big for standard package
Volume: Calculates and displays the volume of the current product.	prodval("Height","unece.unit.CMT") *prodval("Width", "unece.unit.CMT") *prodval ("Length","unece.unit.CMT") **Unit Template:** iterate(stepurl2objid(stepurl ("Unit","unece.unit.CMQ")),'stepid()')	46332 cm^3
Volume (Liter): Calculates the volume in Liter and displays with one decimal.	text(prodval("Height","unece.unit.CMT") *prodval("Width","unece.unit.CMT") *prodval ("Length","unece.unit.CMT")/1000,"0.0")	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
`attrname('NUMBER1id')`	P-family-id	NUMBER1
`prodval('TEXT1id')`	P-family-id	family value
`prodval('NUMBER1id')`	P-family-id	2
`produnit('NUMBER1id')`	P-family-id	m
`prodval('NUMBER1id', 'unece.unit.MMT')`	P-family-id	2000
`prodvalsimple('NUMBER1id')`	P-family-id	2 m
`prodvalsimple('NUMBER1id', 'unece.unit.MMT')`	P-family-id	2000 mm
`getembeddedprefix(prodval('EMBEDDEDNUMBER1id'))`	P3id	prefix
`getembeddedvalue(prodval('EMBEDDEDNUMBER1id'))`	P3id	3
`getembeddedsuffix(prodval('EMBEDDEDNUMBER1id'))`	P3id	suffix
`prodvalsimple('EMBEDDEDNUMBER1id')`	P3id	prefix 3 m suffix
`prodvalsimple('EMBEDDEDNUMBER1id', 'unece.unit.MMT')`	P3id	prefix 3000 mm suffix

Value Template	Object	Result
`produnit('EMBEDDEDNUMBER1id')`	P3id	m
`unitconv('2','unece.unit.MTR','unece.unit.MMT')`	P3id	2000
`valuelovid('LOV1Attrid')`	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-1-25	Dec-31-25
Pink Party Hat	$2.79	$ (iso4217.unit.USD)	12	24	Jan-1-25	Dec-31-26
Blue Party Hat	£6.79	£ (iso4217.unit.GBP)	1	11	Jan-1-25	Dec-31-27
Blue Party Hat	£3.79	£ (iso4217.unit.GBP)	12	24	Jan-1-25	Dec-31-28

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
`termqtymax('PriceID', '1')`	Blue Party Hat	11
`termqtymin('PriceID', '2')`	Blue Party Hat	12
`termunit('PriceID', '1')`	Pink Party Hat Blue Party Hat	$ £
`termunitid('PriceID', '1')`	Pink Party Hat Blue Party Hat	iso4217.unit.USD iso4217.unit.GBP

Value Template	Object	Result
`termval('PriceID', '1', '2015-12-31 01:01:01')`	Pink Party Hat	$7.79
`termval('PriceID', '2', '2015-12-31 01:01:01')`	Pink Party Hat	$2.79
`termvalid('TermsListID', '2')`	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
`if (1=1, 'true', 'false')`	true	1 does equal 1 so the first result is returned
`if (not(1=1), 'true', 'false')`	false	NOT changes 'False to True' or 'True to False,' so the second result is returned
`if (exact('hello','you'), 'true', 'false')`	false	EXACT requires that the two values match, so the second result is returned
`if (or(1=1,exact('hello','you')), 'true', 'false')`	true	OR allows either 1=1 or an exact match, so the first result is returned
`if (and(1=1,exact('hello','you')), 'true', 'false')`	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
`and((value('TEXT1id')= 'p1 value'), (value('NUMBER1id')= '4'))`	P1id	1	Both conditions are true
`and((value('TEXT1id')= 'p0 value'), (value('NUMBER1id')= '3'))`	P1id	0	Both conditions are false
`or((value('TEXT1id')= 'p1 value'), (value('NUMBER1id')= '4'))`	P1id	1	One condition is true
`or((value('TEXT1id')= 'p0 value'), (value('NUMBER1id')= '3'))`	P1id	0	Both conditions are false
`not(value('TEXT1id')= 'p1 value')) // = 0 (FALSE) evaluated P1id`	P1id	0	Condition is true, NOT results in a return of 'false '
`not(value('TEXT1id')= 'p0 value')) // = 1 (TRUE) evaluated P1id`	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
`if(prodval('Length') = prodval('Width'), 'Product is Square.','Product is Oblong.')`	Product is Square.	Length does equal Width, so the first result is returned
`if(prodval("PowerCordLength") > 0,concatenate ('A ', prodvalsimple("PowerCordLength"), ' power cord is included.'), 'Power cord not included.')`	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.
`if(prodval("PowerCordLength") > 0,concatenate ('A ', prodval("PowerCordLength"), produnit("PowerCordLength"), ' power cord is included.'), 'Power cord not included.')`	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
`formatdatetime(prodval('AttrDateid'), "EEE, MMM d, ''yy")`	P1id	Fri, Oct 15, '16
`formatdatetime(prodval('AttrDateid'), "yyyyy.MMMMM.dd" )`	P2id	01997.December.25
`formatdatetime(prodval('AttrDateid'), "MMM d, yy" )`	P3id	May 15, 16
`localizedate(prodval('AttrDateid'))`	P3id when in Danish context	15-05-2016
`now()`	any when executed	2016-05-04 12:06:25

Value Template	Object	Result
	on 4 MAY 2016 just after midday	
{d := value('StartDate'), x := localizedate(d, "en_GB"), y := localizedate(d, "en_US"), z := localizedate(d, "de_DE") } concatenate(d, "; ", x, "; ", y, "; ", z)	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
`list(iterate(subproducts(), 'prodval("TEXT1id")'), ', ')`	P-family-id	p1 value, p2 value, p3 value
`list(iterate(subproducts(), 'prodval("TEXT1id")'), ', ', ' & ')`	P-family-id	p1 value, p2 value & p3 value
`list(reverse(iterate(subproducts(), 'prodval("TEXT1id")')),', ')`	P-family-id	p3 value, p2 value, p1 value
`list(iterate(subproducts(), 'concatenate(index,": ",item)'), '\n')`	P-family-id	1. com.stibo.core.persistence.NodePO-343239, 2. com.stibo.core.persistence.NodePO-3432395, 3. com.stibo.core.persistence.NodePO-3432395 **Note:** 'com.stibo.core.persistence.NodePO-3432391,…' are internal

Value Template	Object	Result
		representation of the nodes (sub products). This 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

Value Template	Object	Result
`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 *other than* P-family-id	p1 is NOT present
		This is the same formula as shown previously except the list formula is written in full within the LISTCONTAINS function

`listlen(iterate(subproducts(), 'prodval("TEXT1id")'))`	P-family-id	3
`listitem(iterate(subproducts(), 'prodval ("TEXT1id")'),2)`	P-family-id	p2 value
`iterate(filter(subproducts(),'prodval ("number1id")>3'),` `'concatenate("Name = ",stepname(),"; ID = ", stepid(), ". ")')`	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

`{x := iterate(subproducts(), 'prodval("TEXT1id")')`
`}`
`if(listcontains(x,"p1 value"),`
`"p1 is present","p1 is NOT present")`
	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

`range(listconcatenate('1','5','8','6','7'), '-')`	{an	1-8

Value Template	Object	Result
	y}	
`range(listconcatenate('1','5','8','6','7'), '-', ', ')`	{any}	1, 5-8
`sort(iterate(subproducts(), 'prodval("NUMBER1id")'))`	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
`1.25*11`	13.75
`int(5.64)`	5
`int(-5.64)`	-6
`text(2.45, '000.0')`	002.5
`text(2.45, '.0')`	2.5
`text(2000.45, ',.0')`	2,000.5
`trunc(5.64)`	5
`trunc(-5.64)`	-5
`round(5.64, 0)`	6
`round(5.64, 1)`	5.6
`round(5.64, -1)`	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
`(value('length'))*(value('width'))`	10158.026	No formatting is applied to the resulting number.

`prodval('height', 'unece.unit.CMT') *`
`prodval('width', 'unece.unit.CMT') *`
`prodval('length', 'unece.unit.CMT')`
	510948.7078	No formatting is applied to the resulting number.

`text(prodval('height', 'unece.unit.CMT') *`
`prodval('width', 'unece.unit.CMT') *`
`prodval('length', 'unece.unit.CMT'), "000,000.0")`
	510,948.7	The TEXT function adds a thousands delimiter and reduces the decimal places to 1

`concatenate(prodvalsimple('height','unece.unit.CMT'), ' H x ',`
`prodvalsimple('width','unece.unit.CMT'),' W x ',`
`prodvalsimple('length','unece.unit.CMT'),' L')`
	50.3 cm H x 135.26 cm W x 75.1 cm L	Static text strings provide a typical display for dimension information.

Value Template	Result	Notes
concatenate(trunc(prodval('height')), " x ", trunc(prodval('width')), " x ", trunc(prodval('length')))	50 x 135 x 75	The TRUNC function eliminates the decimal point and places.
concatenate(int(prodval('height')), " x ", int(prodval('width')), " x ", int(prodval('length')))	50 x 135 x 75	The INT function returns the integer part of the number.
concatenate(round(prodval('height'),**1**), " x ", round(prodval('width'),**0**), " x ", round(prodval('length'),**-1**))	50.3 x 136 x 80	The ROUND 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

`{x := value('Diameter'),`
`y := 3.414,`
`num_us := localizenumber(x, "en_US"),`
`num_de := localizenumber(x, "de_DE"),`
`p := num_us*y, q := num_de*y,`
`num := localizenumber(p)`
`}`
`concatenate(x, "; ", p, "; ", q, "; ", num)`
	12,7; 43.3578; NaN; 43,3578	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.
- **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.
- **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 links to more detailed use cases, see the **Other Functions** topic in the **STEP Functions** section of the **Resource Materials** online help.

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
`list(iterate(classifications(), "stepid()"), "/ ")`	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 **Initial Setup for Metrics** in the **System Setup** 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
`completeness()`	51.28205128205128460194828221574 4
`round(completeness(),0)`	51

Data Container Functions

These functions can be used to access data container objects on products or entities.

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

Data Container Types (ID):	Non-Typed Data (244320)	Technical Specifications (244321)
Product: ALRM-01 (157618)	Non-Typed Data ID / Field Name / Supplier Cost / Value NonTypedData_286148 / Bells / 0.45 / 2 NonTypedData_286149 / Handles / 0.28 / 1 Add Data Container	Technical Specifications ID / Attribute Name / Value 286155 / Inrush VA / 12 286155 / Sealed VA / 2.8 286155 / Sealed Watts / 4w 286155 / Supplier Cost / 0.19
Product: ALRM-02 (157619)	{no Non-Typed Data (244320) data container exists}	Technical Specifications ID / Attribute Name / Value 286156 / Inrush VA / 10 286156 / Sealed VA / 1.9 286156 / Sealed Watts / 9w 286156 / Supplier Cost / 0.36

Value Template	ALRM-01 Result	ALRM-02 Result

{
dclist := datacontainers(),
dc_typeids := iterate(dclist, 'datacontainertypeid()')
}
list(sort(unique(dc_typeids)), "\n")
	244320
	
	244321
		244321

{
dc := datacontainers(),
dcid := list(iterate(dc, 'value("Supplier_Cost")'), "/"),
dcc := list(iterate(dc, 'datacontainertypeid()'), ", "),
dc1 := listitem(dc, 1),
dc2 := iterate(datacontainercomposite(), 'stepname()'),
msg := ""
}
concatenate(dcid, " --- ", dcc, " --- ", dc2)
	0.19/0.28/0.45 --- 244321, 244320, 244320 --- ALRM-01	0.36---244321---ALRM-02

{
dc := datacontainers(),
dcid := sort(unique(iterate(dc, 'value("Supplier_Cost")'))),
introtext := if(listlen(dcid) <gt/> 1, "Supplier Cost ranges
from ", "Supplier Cost is "),
septext := if(listlen(dcid) <gt/> 1, " to ", ""),
 	
firstval := listitem(dcid,1),
lastval := if(listlen(dcid) <gt/> 1, listitem(dcid,
listlen(dcid)), ""),
 	
dctypes := sort(unique(iterate(dc, 'datacontainertypeid()'))),
dclist := if(listlen(dctypes) <gt/> 1, list(dctypes, ", ",
" and "), listitem(dctypes,1)),
dctext := if(listlen(dctypes) <gt/> 1, " in Data Container
types ", " in Data Container type "),
msg := "Data Container examples"
}
concatenate(introtext, firstval, septext, lastval, dctext,
dclist)
	Supplier Cost ranges from 0.19 to 0.45 in data container types 244320 and 244321.	Supplier Cost is 0.36 in Data Container type 244321

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
`list(iterate(children(), 'concatenate(stepname())'),', ')`	P-family-id	P1, P2, P3	These are products
`list(iterate(subproducts(), 'concatenate(stepname())'),', ')`	P-family-id	P1, P2, P3	These are products
`list(iterate(children(), 'concatenate(stepname())'),', ')`	E-family-id	E1, E2	These are entities
`list(iterate(subproducts(), 'concatenate(stepname())'),', ')`	E-family-id		These are entities, not products
`parentid()`	P1id	P-family	

Value Template	Object	Result	Notes
`list(iterate(path(), 'concatenate(stepname())'),', ')`	P-family-id	Primary Product Hierarchy, Products, P-family	
`isbelow('step://product?id=P1id')`	P-family-id	0	FALSE
`isbelow('step://product?id=P-Family')`	P1id	1	TRUE
`{i:=path(), parent:=listitem(i,listlen(i)-1) } iterate(parent, 'concatenate("name=",stepname(),"; id=", stepid())')`	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

{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), "")
	J-family-id	Alpha	Value for attribute ID 'BrandName' that occurs the most times in its children.

{attrid:='grandchildattrid' } listconcatenate(iterate(subproducts(), 'iterate(subproducts(),"prodval(attrid)")'))	J-family-id	AlphaAlphaAlphaZetaZeta	Use cautiously. This operation can drain system CPU since all grandchildren are read for each product.
{msg := "Closest date that is less than the current date is: ", datenow := left(substitute(now(), "-", ""), 8),	J-family-id	Closest date that is less than the current date is: 2016-05-01	Returns the value of

Value Template	Object	Result	Notes

`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))`
			'Removal Date' 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 **Transformation Lookup Tables** topic in the **Resource Materials** online help.

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
`REPLACEVALUEBYLOOKUP('UOMLookupTable','EACH')`	Ea.
`REPLACEWORDBYLOOKUP('UOMLookupTable','EACH, DOZEN, CASE')`	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
`stepid()`	P1id	P1id
`stepname()`	P1id	P1
`stepname('step://product?id=P1id')`	P1id	P1
`stepurl()`	P1id	step://product?id=P1id
`stepurl('Product','P1id')`	P1id	step://product?id=P1id
`stepobjecttype()`	P1id	Item

Value Template	Object	Result
`stepobjecttype('step://product?id=P1id')`	P1id	Item
`iterate(stepurl2objid(stepurl('Product','Level1-109267')),'stepname()')`	{any}	P1id

`{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)`
	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
`overriddenproduct('Override1ID')`	Override1ID	Batteries

Template	Object	Results
`overriddenproduct('Batteries')`	Override1 ID	N/A

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

`{x := iterate(overriddenproductobject(), 'stepname()'),`
`z := listitem(x,1)`
`}`
`concatenate('This PO overrides the product "', z, '"')`
	Override1 ID	This PO overrides the product "Batteries"

`list(iterate(productoverrideparents(), "stepid()"), "/ ")`	BatteryIte msID	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
`list(iterate(iterate(references('product', 'BillOfMaterials'),'referencetarget()'),'stepname()'),', ')`	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
`list(iterate(revisions(), 'revisioncreatedate()'), ', ')`	P-family-id	2015-06-15 16:27:04, 2016-05-08 11:34:51, 2017-07-12 12:06:16

Value Template	Object	Result
`revisionname()`	P-family-id	1.1
`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
`workflowids()`	P1id	Workflow5Workflow6
`currentworkflowstates('Workflow6')`	P1id	State-A

Publishing Examples

Publishing functions can be used to pull information from objects within the publication hierarchy, such as:

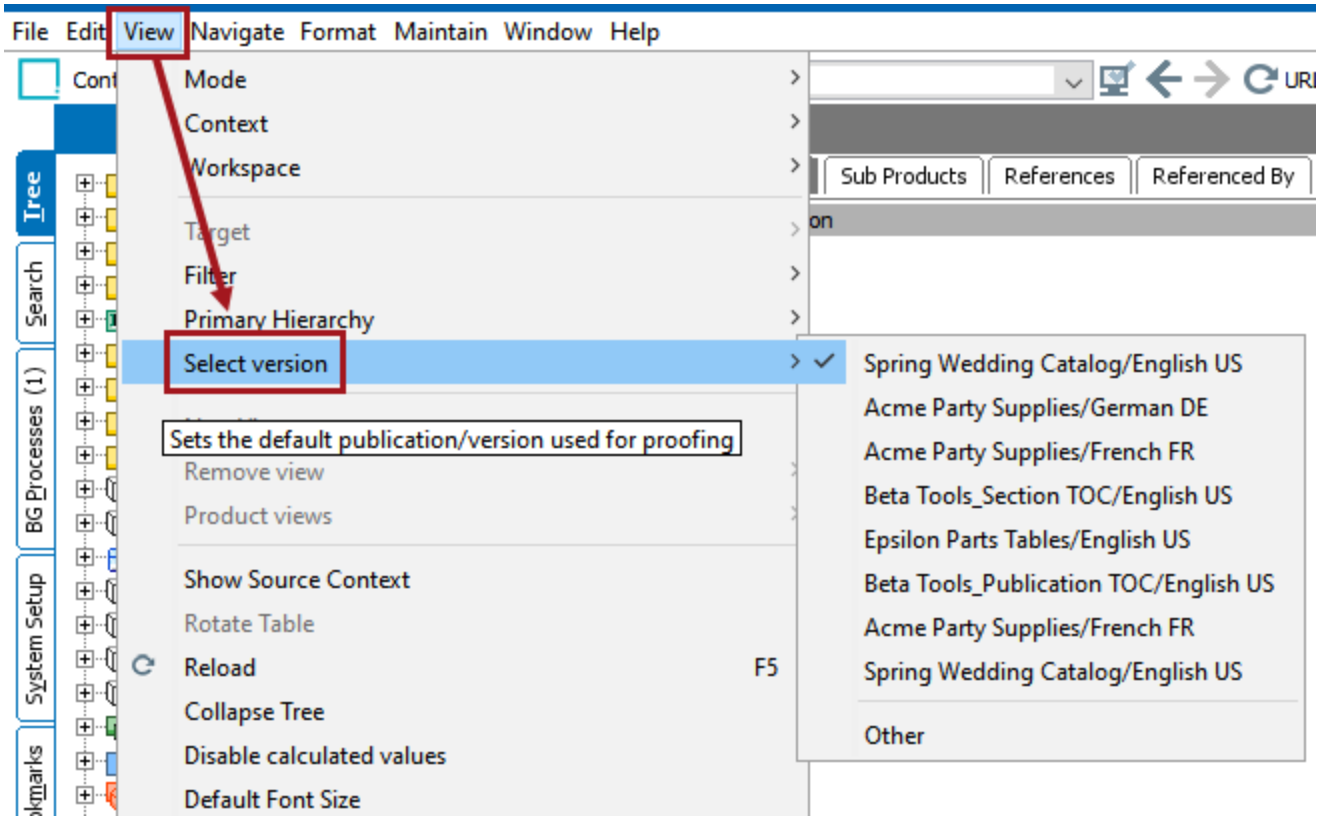
- The values of metadata attributes on publication group, publication, and section objects
- The name of a publication
- The name and/or ID of a currently selected publication version
- The first and last page numbers of the DTP pages saved to a section
- The page number(s) on which a product appears in a publication

These functions are most commonly used within calculated attributes whose values display on InDesign pages created with STEP Publisher. They can also be used in formulas that display in publication-related locations in the workbench, such as the Tables preview and in Flatplanner Sticker Book views, baskets, and financial reports.

Prerequisites

A publishing function will not return a value unless a publication **version** is selected. Additionally, the selected version must typically be relevant to the object on which the calculated attribute appears. One example would be for a product that appears on a DTP page that is saved in a particular publication. If a version from a *different* publication is chosen, then values such as page numbers will not display on the product, since it does not appear on any pages within that publication.

The examples in this topic show how calculated values from publishing functions display on **product** objects in the STEP Workbench Tree. To choose a version for testing purposes, go to View > **Select version** in the menu bar, then choose the relevant publication version.



Sample Formulas and Values

This section provides basic examples of calculated attribute formulas that use publishing functions, along with illustrations of the values they return. Where applicable, links are provided to more detailed topics that appear in other sections of the STEP Online Help. Though the examples provided are basic, they can be embedded within more complex calculations.

For the full list of publishing functions and brief descriptions of their intended purpose, see the **Publishing Functions** section of the **STEP Functions** documentation.

getInheritedPublicationValue and getInheritedPublicationHierarchyValue

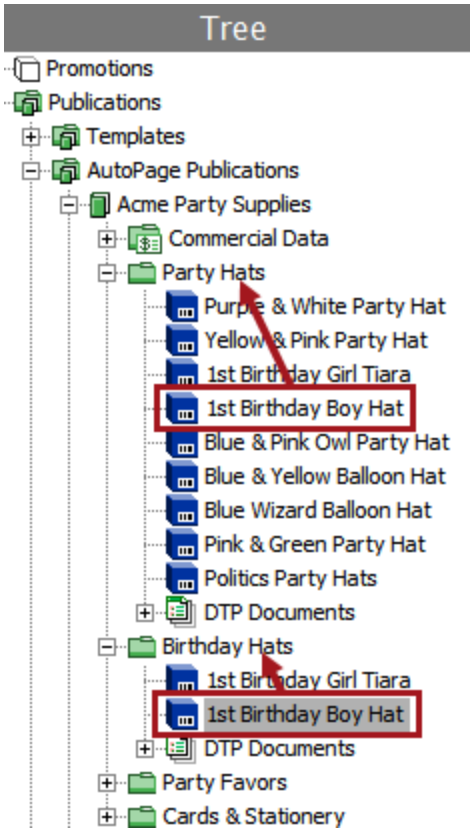
The getInheritedPublicationValue and getInheritedPublicationHierarchyValue functions are very similar. When used for product objects, both return metadata values from attributes on publication groups and publications, whereas the getInheritedPublicationHierarchyValue also returns values from sections.

Since the getInheritedPublicationValue function returns values from publication groups, publications, and sections when used to mount metadata using **publication templates**, it is more commonly used for mounted pages, whereas getInheritedPublicationHierarchyValue is more commonly used for products. The publication template scenario is only relevant to the Flatplanner and AutoPage components. For more information, see:

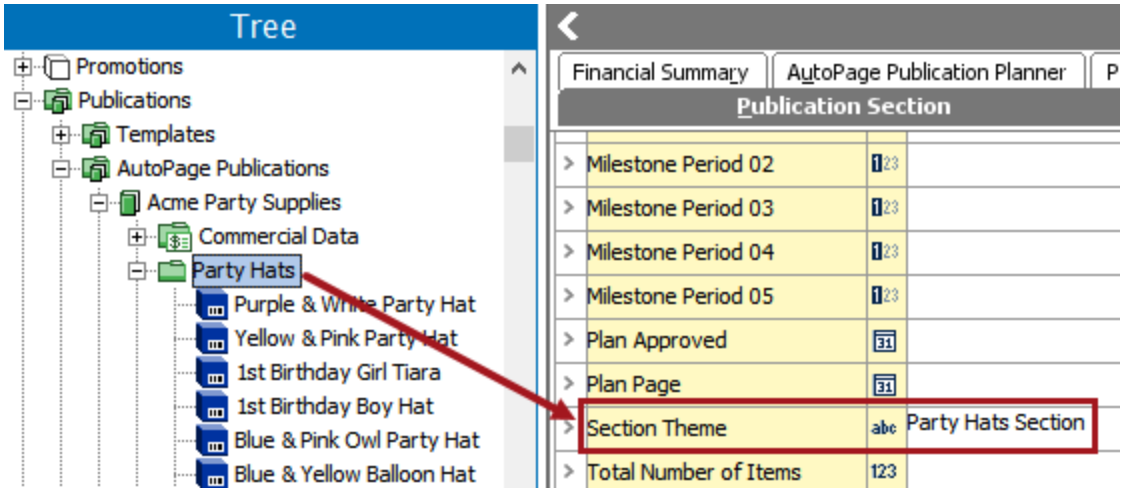
- **Using the getInheritedPublicationValue Function in Flatplanner** in the **Flatplanner** documentation
- **Using the getInheritedPublicationValue Function in AutoPage**in the **AutoPage** documentation

The following example uses the getInheritedPublicationHierarchyValue function to pull metadata from a section onto a product, using the formula `getInheritedPublicationHierarchyValue('SectionTheme')`. 'SectionTheme' is the ID of the source attribute on the section.

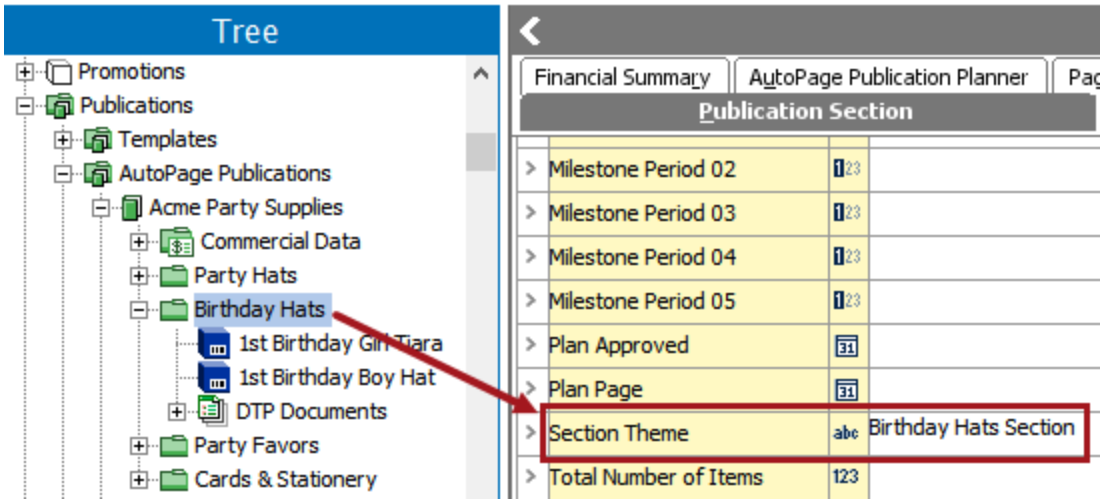
The selected product is linked into two sections of the publication—'Party Hats' and 'Birthday Hats.'



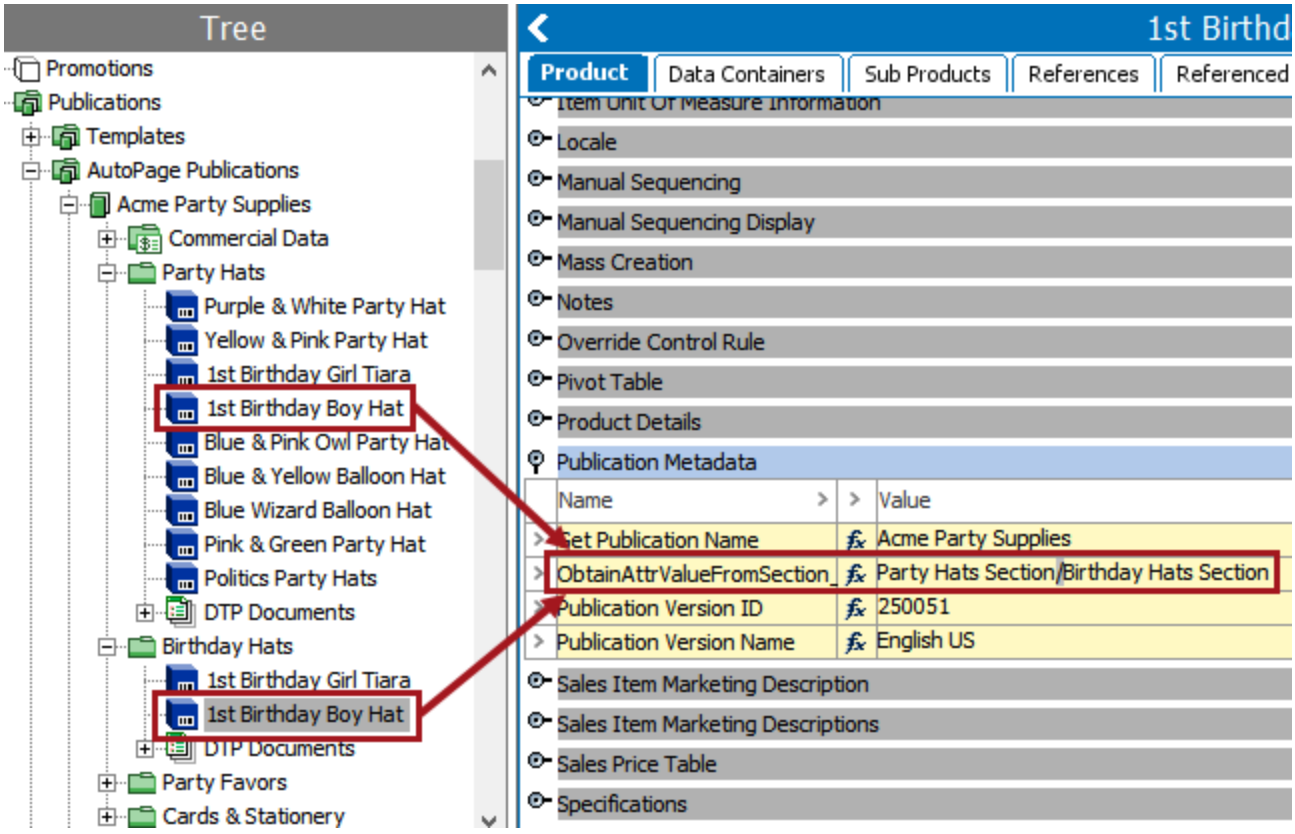
On 'Party Hats,' the value of Section Theme (SectionTheme) is 'Party Hats Section.'



On 'Birthday Hats,' the value of Section Theme (SectionTheme) is 'Birthday Hats Section.'



With the product selected in the Tree, the value of Section Theme (SectionTheme) from both sections displays in the ObtainAttrValueFromSection calculated attribute field. Values are sorted as they appear from top to bottom in the publication hierarchy, and are separated with a slash (/).

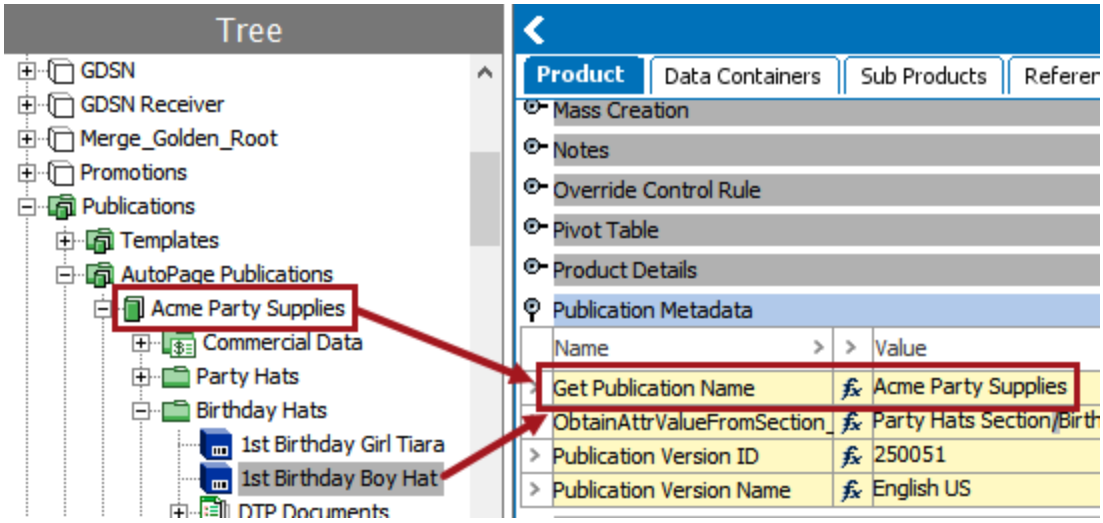


Similar values will be returned if the calculated attribute formula includes the ID of a metadata attribute that appears on a publication group or publication.

getPublication

Use this formula to return the name of a parent publication using the getPublication function:

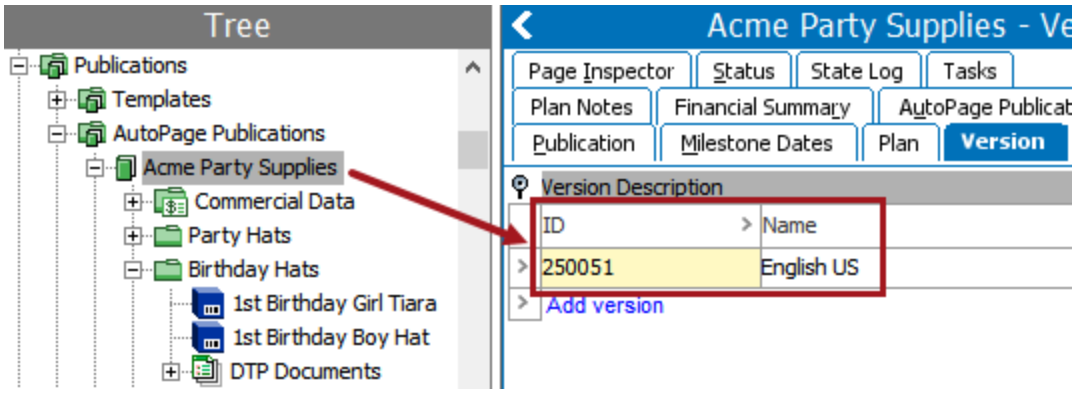
```
iterate(getPublication(),"stepname()")
```



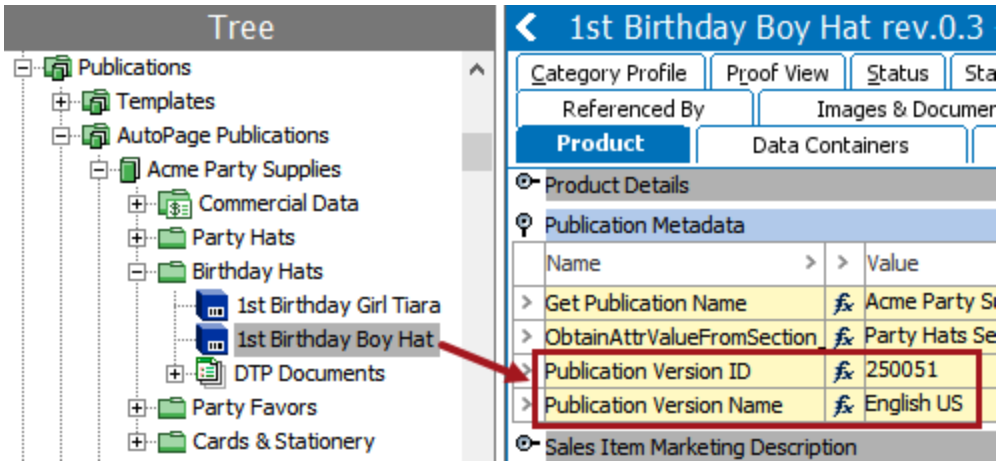
getPublicationVersionID and getPublicationVersionName

The below example uses the simple formulas of `getPublicationVersionID()` and `getPublicationVersionName()` to return the ID and Name of the current publication version.

The following image shows the ID (250051) and Name (English US) of the selected publication on the **Version** tab.



Below, the product is selected and the values of the publication ID and Name display in the calculated attribute fields.

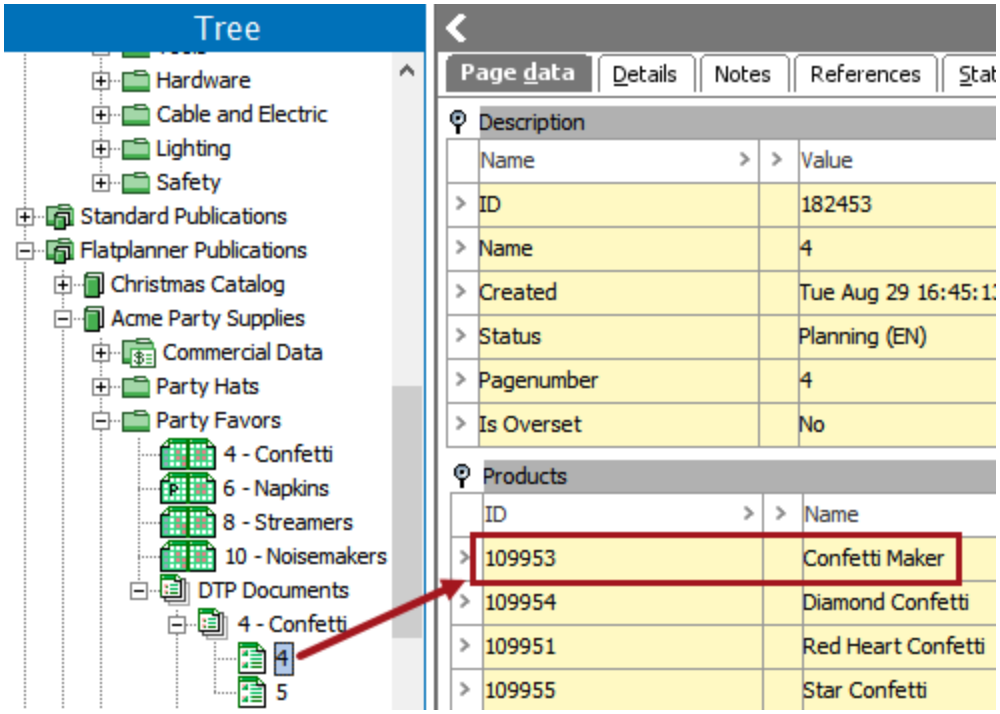


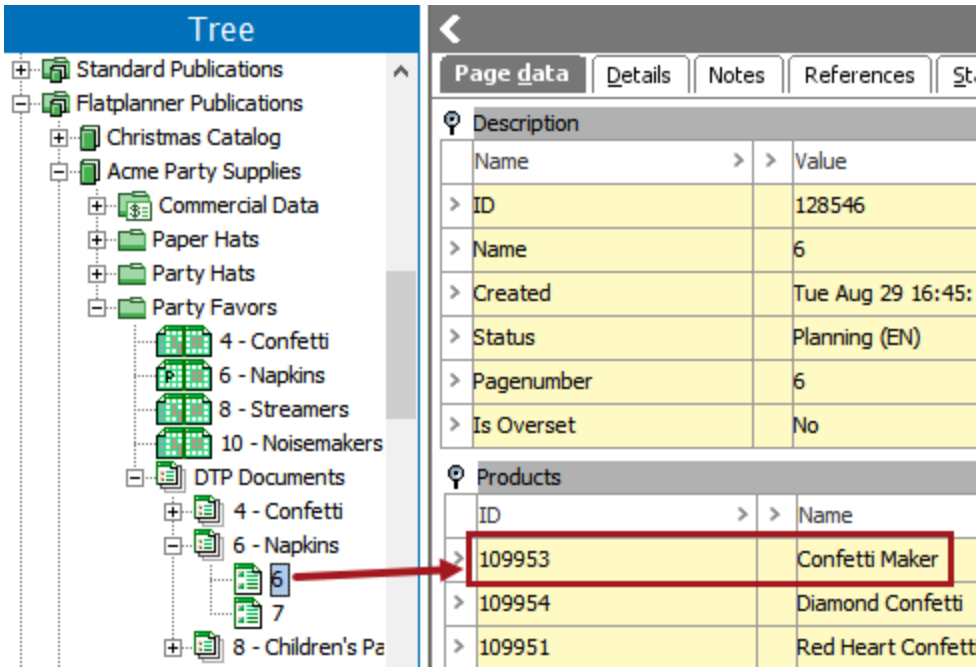
pagenumbers

This formula uses the pagenumbers function to return a list of page numbers (separating each with a comma and a space) on which a product appears in a publication:

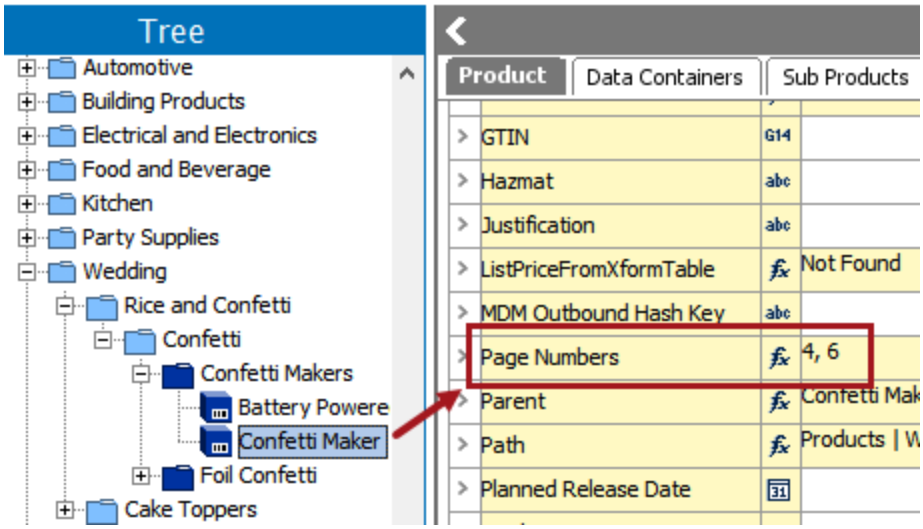
```
list(sortnumeric(pagenumbers()), ", ")
```

The below screenshot shows the **Page data** tab of two selected DTP pages. The products that appear on the pages are listed under the **Products** flipper. This example uses a product named 'Confetti Maker,' which appears on both page 4 and 6 of the publication.



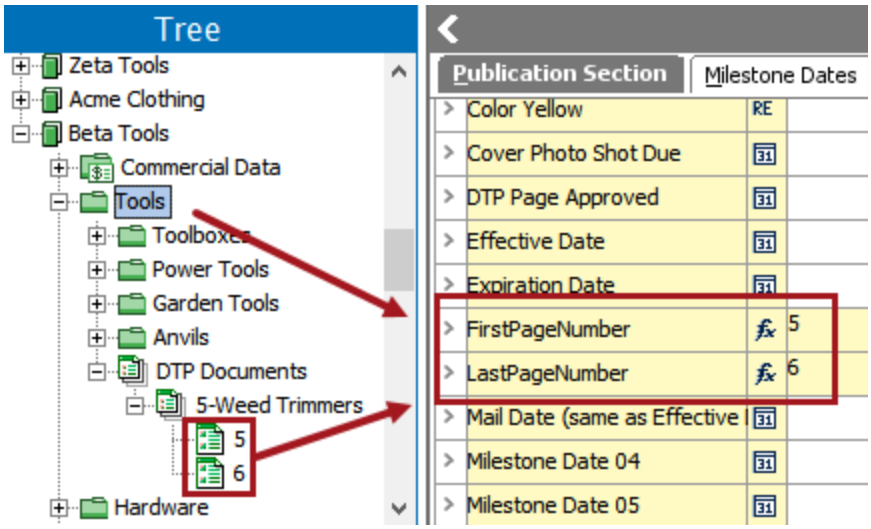


When the Confetti Maker product is selected in the Tree, the page numbers on which it appears display in the calculated attribute field, e.g., pages 4 and 6.



sectionFirstPageNumber and sectionLastPageNumber

Use the formulas `sectionFirstPageNumber()` and `sectionLastPageNumber()` to return the values of the first and last page numbers of the DTP pages saved within a section. Typically there will be more than two pages, but for this basic example, only two DTP pages have been saved into the Tools section.



For a scenario using the SectionFirstPageNumber and SectionLastPageNumber functions to create a table of contents, see **Creating a Table of Contents Product Template** in the **STEP'n'design** documentation.

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
`escapemerge(';', '/', "Red", "Orange; Blue", "Violet")`	Red;Orange/; Blue;Violet	Write semicolon (;) (split character) after each element and include slash (/) (escape character) when split character is found in text.
`escapesplit(';', '/', 'Red;Orange/; Blue;Violet')`	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
`left('hello')`	h	1 character from left
`left('hello',3)`	hel	3 characters from left
`right('hello')`	o	1 character from right
`right('hello', 3)`	llo	3 characters from right
`mid('hello',2,3)`	ell	3 characters starting at character 2
`len('hello')`	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
`concatenate("hello", "hallo")`	hellohallo	join multiple text strings into a single text string
`trim('  hello  ')`	hello	remove spaces before and after text
`upper('hello')`	HELLO	convert to uppercase letters
`lower('HELLO')`	hello	convert to lowercase letters
`proper("hello, are you john's son?")`	Hello, Are You John'S Son?	capitalize the first letter of each word, including capitalization after

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

`concatenate(`
`prodval('height'),`
`prodval('width'),`
`prodval('length'))`
	Product 1	5013575	Concatenating only the values excludes the unit information and results in an unformatted number.

`concatenate(`
`prodvalsimple('height','unece.unit.CMT'),' H x ',`
`prodvalsimple('width','unece.unit.CMT'),' W x ',`
`prodvalsimple('length','unece.unit.CMT'),' L')`
	Product 1	50 cm H x 135 cm W x 75 cm L	Static text strings provide a typical display for dimension information.

`list(listconcatenate(`
`prodval("height"),`
`prodval("width"),`
`prodval("length")),'*')`
	Product 2	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
`replace(prodval('prodname'), 6,2,'Boardroom Projector Series')`	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.
`substitute(prodval('prodname'), '-BP-',' Boardroom Projector Series ')`	Product3	ACME Boardroom Projector Series 1001 Super XBP	Update all occurrences of -BP-, so that the second instance of BP is not changed.
`substitute(prodval('prodname'), 'BP','Boardroom Projector Series')`	Product3	ACME-Boardroom Projector Series-1001 Super XBoardroom	Update all occurrences of BP.

Value Template	Object	Result	Notes
		Projector Series	
`substitute(prodval('prodname'), 'BP','Boardroom Projector Series',1)`	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
`regexpsearch('e.*he', 'hellohello')`	2	2 is the start point for 'ellohe' as a case sensitive match for the regexp
`regexpsearch('[ABC]\\d', 'HELLOHA2MSTER')`	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.
`regexpsearchi('[efg]|[ojk]', 'HELLOHELLO')`	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

Value Template	Result	Notes
		letters is required. | indicates 'or.'
`regexpsearchi('[abc]|[ojk]', 'HELLOHELLO', 3)`	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.'
`regexpsubstitute("kat;abe", "(.*);(.*)", "$2$1")`	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
`regexpsubstitute("kat  abe", "(\\w*)\\s*(\\w*)", "$2$1")`	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
`soundex("hello")`	H400
`metaphone3("Joel Dalton")`	JLTLTN
`metaphone3alternate("Joel Dalton")`	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
`exact("hello", "hallo")`	0	Exact match not found
`exact("hello", "hello")`	1	Exact match is found
`if (exact('hello','you'), 'true', 'false')`	false	Exact match is not found so the second result is returned
`find('llo','hellohello')`	3	The position of the first occurrence
`find('llo','hellohello', 4)`	8	The position of the first occurrence found after character 4
`find('lllo','hellohello')`	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
`if(exact(prodval('PowerCordIncluded'),'Yes'), 'A 2 meter power cord is included.', 'Power cord is not included.')`	Product4 Product5	A 2 meter power cord is included. 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

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

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

`{pnnokey := concatenate('[PN]',prodval ("ProductName")),`
`pnwkey := concatenate('[KL]',prodval ("KeyLetter"),`
`'. [PN]',prodval("ProductName"))`
`}`
`if(exact(prodval ("KeyLetter"),''),pnnokey,pnwkey)`
	`if(exact(prodval("KeyLetter"),''),`
	`concatenate('[PN]',prodval ("ProductName")),`
	`concatenate('[KL]',prodval ("KeyLetter"),`
	`'. [PN]',prodval("ProductName")))`

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

`{x := find('-BP-', prodval("prodname")),`
`y := 'Boardroom Projector Series'`
`}`
`if(exact(x,'N/A'),'',replace(prodval("prodname"),x+1,2,y))`
	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

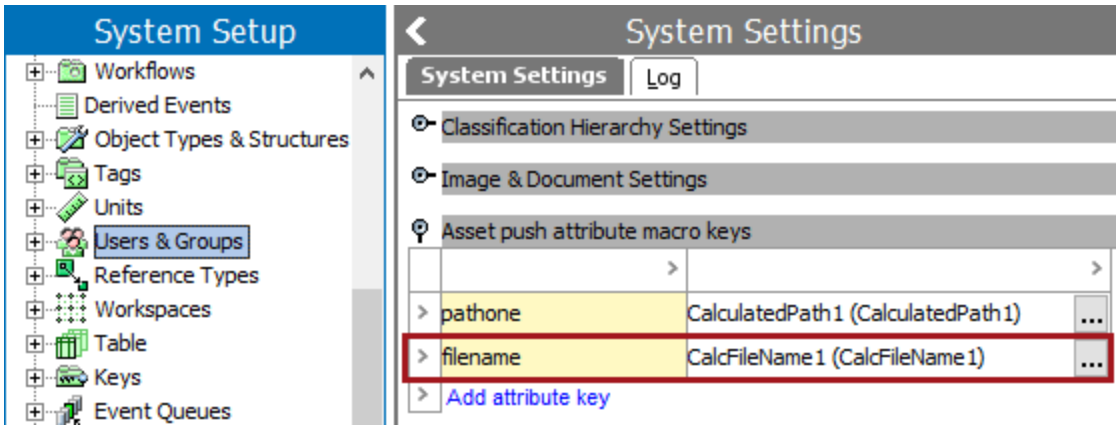
A calculated attribute can be used to create a custom file name for use with assets exported from STEP using the Asset Push functionality.

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.

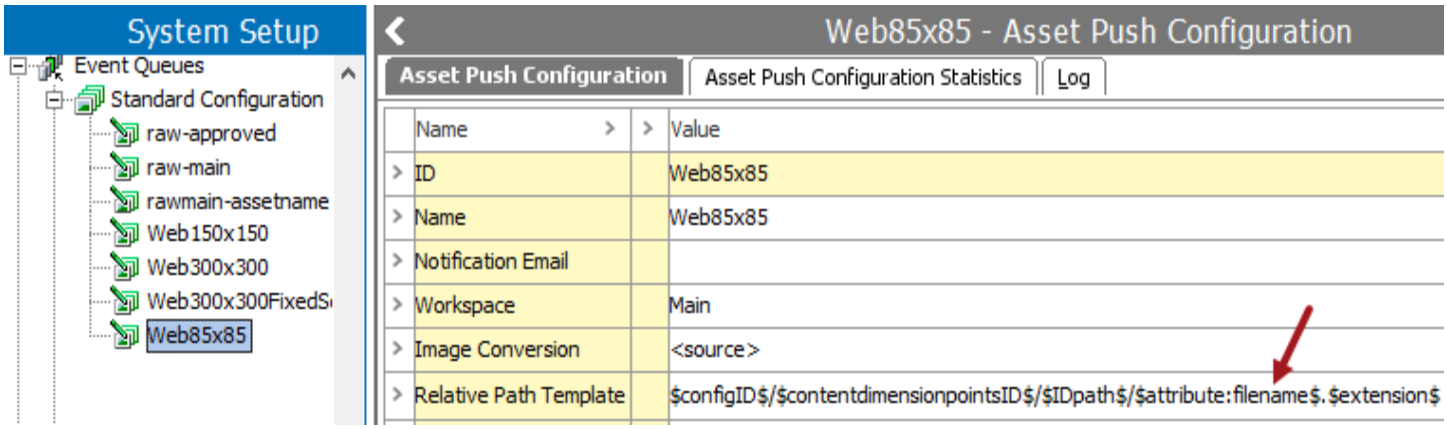
For information on Asset Push, see the **Asset Push** topic in the **Digital Assets** documentation.

Assumptions

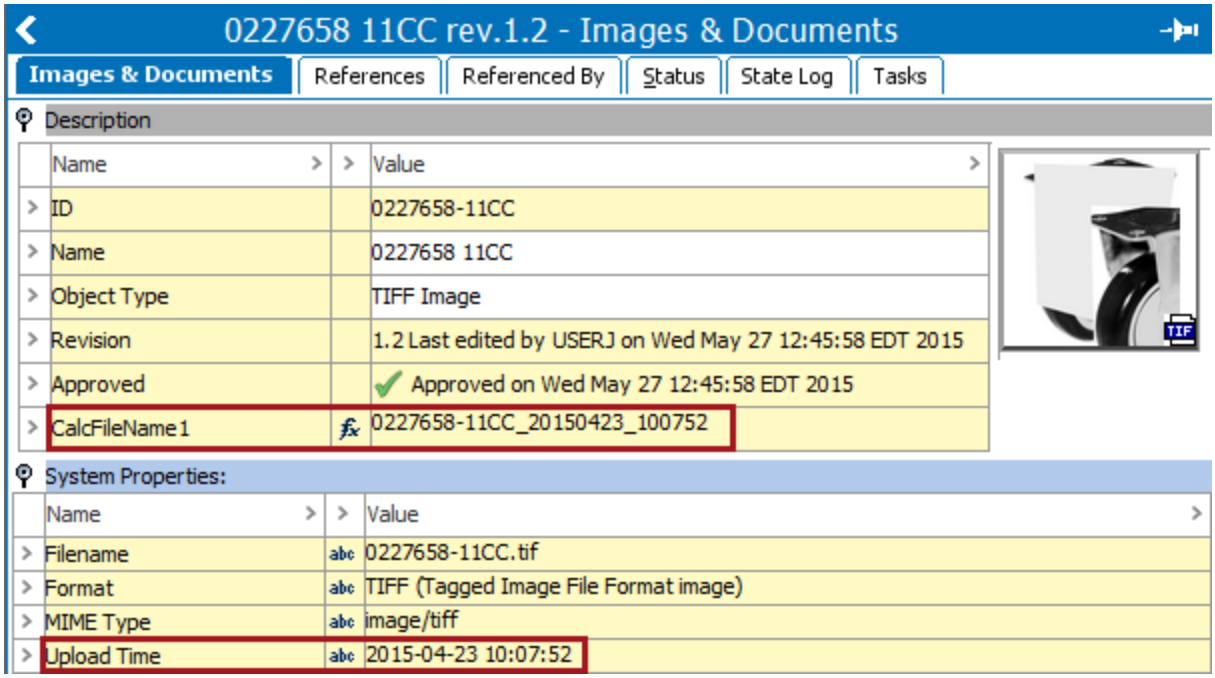
1. An asset push attribute macro key with the Attribute Key value 'filename' is created to hold the calculated attribute 'CalcFileName1(CalcFileName1)'. A macro key attribute must be a description attribute. For more information on this macro, see **Asset Push Attribute Macro Keys** topic within the **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.



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.



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.

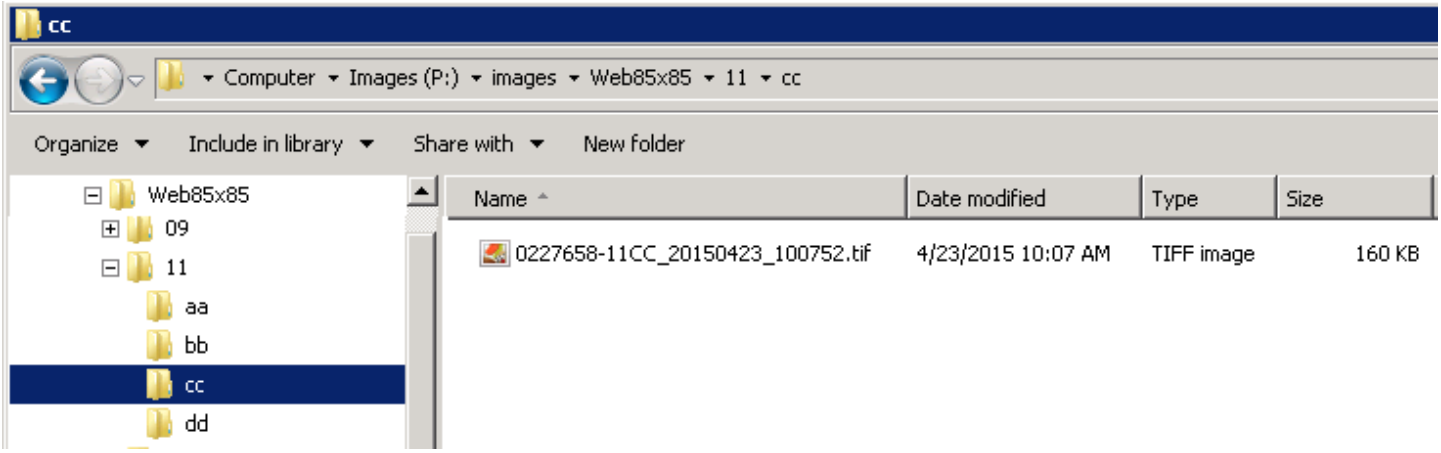
Value Template

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

Results

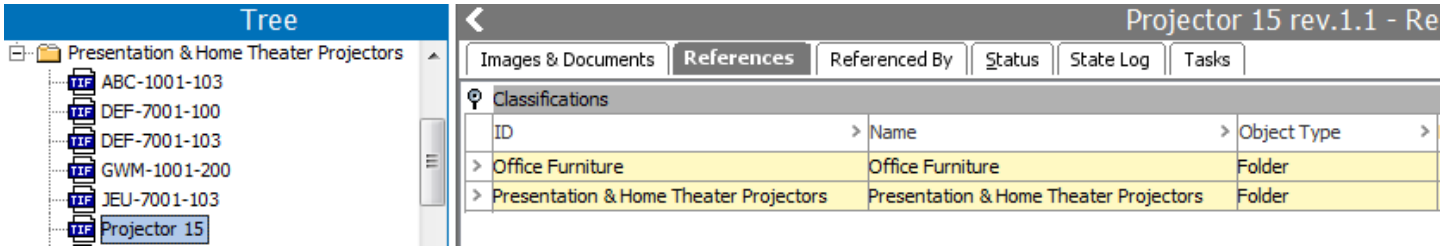
The pushed asset file is saved to the path generated by the Relative Path Template within the Asset Push Configuration, and includes the file name generated by the calculated attribute. The calculate attribute separates each of the elements of the file name listed below with an underscore (_):

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



Classification Scenario

An asset is linked to two classifications.

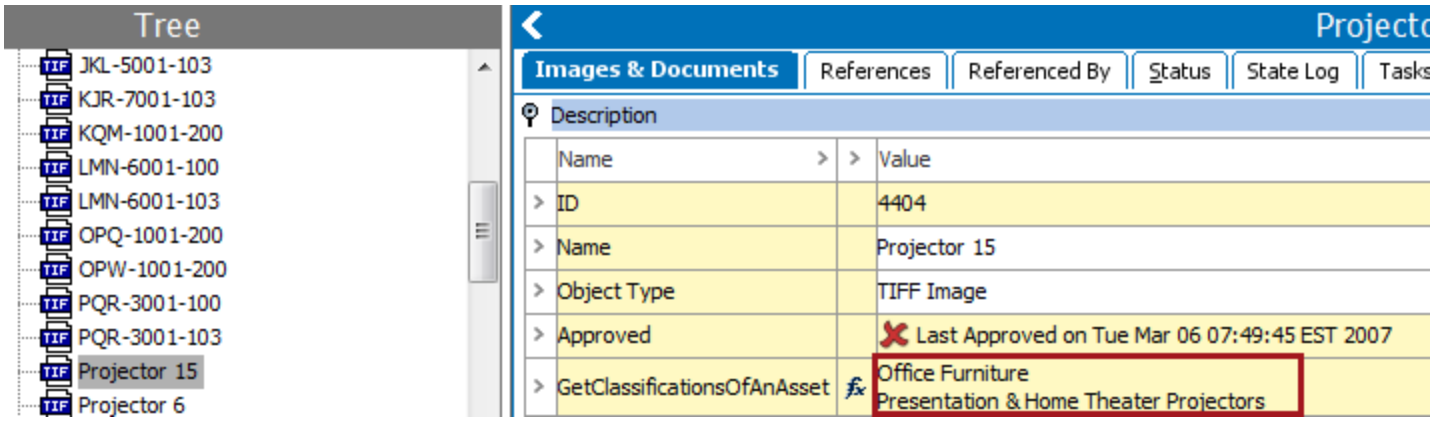


Get All Classifications

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:

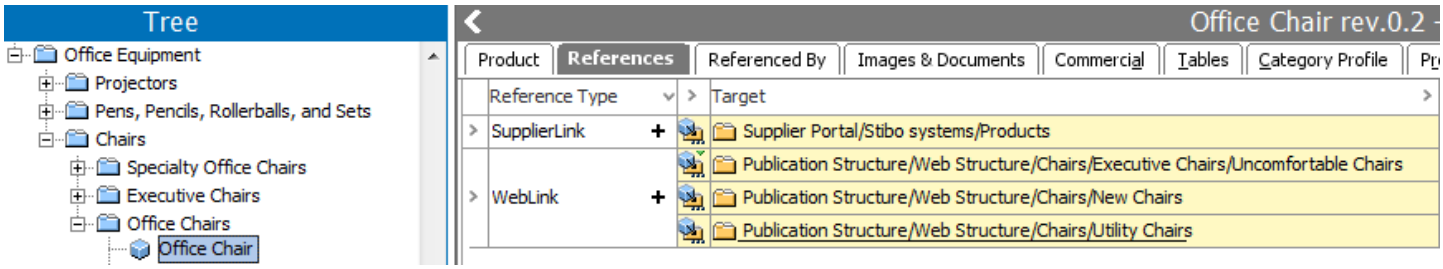
```
list(iterate(classifications(), 'stepname()'), "\n")
```

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.



Get Classifications Based on Type

Multiple types of classifications exist.

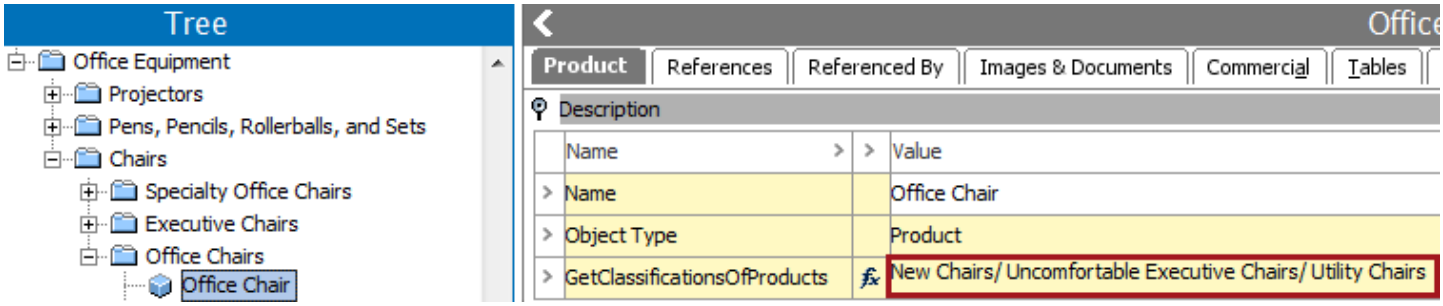


A calculated attribute on the product could display a limited number of classifications. For example, only include the 'WebLink' references type, which have an ID of 'WebInclude'.

The calculated attribute formula could be written as:

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

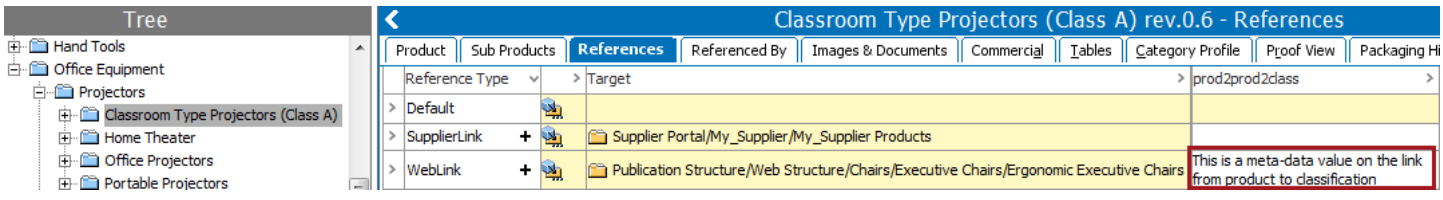
The result displays the references separated by the text string "/ ".



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

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

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

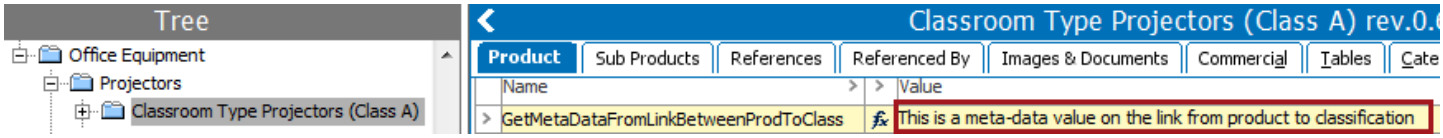


The calculated attribute formula could be written as:

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

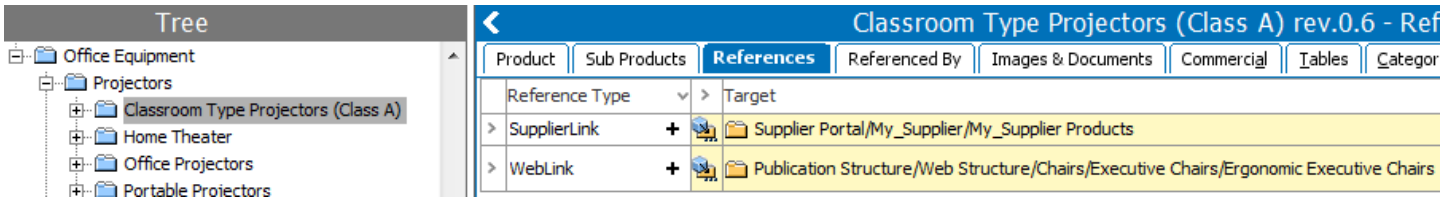
This will display the meta attribute value from all classification links with type WebInclude and the separator between the multiple links values is “\n” (new line).

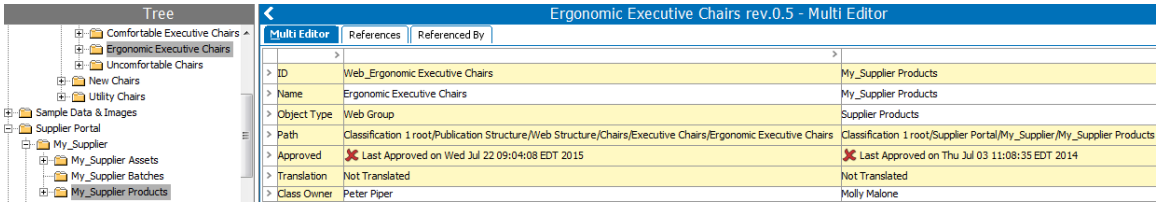
The result displays the value of the linked metadata on the original product.



Get CPL Classifications Using Classification Product Links

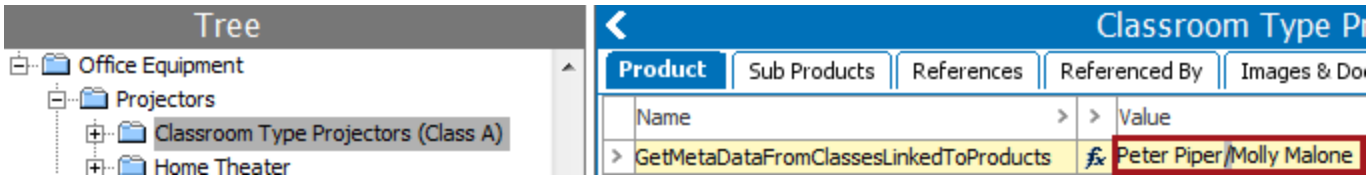
A product is linked into one or more classifications and metadata from each classification is needed. For example, for the Classroom Type Projectors product, linked using SupplierLink and WebLink, the calculated attribute will display the Class Owner attribute (ID = 'ClassOwner') from the classification.





The 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.



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 the **Commercial Data** section of the **STEP Publisher (InDesign)** 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

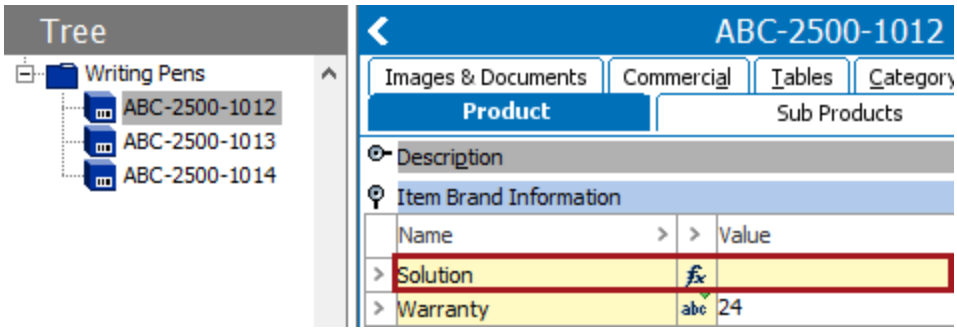
```
{i:=path(),
parent:=listitem(i,listlen(i)-1),
parentvalue := iterate(parent, 'value("Warranty")'),
localvalue := value("Warranty")

}
if(exact(parentvalue,localvalue), "", concatenate("Parent Value = ",
parentvalue, ", Local Value = ",localvalue))
```

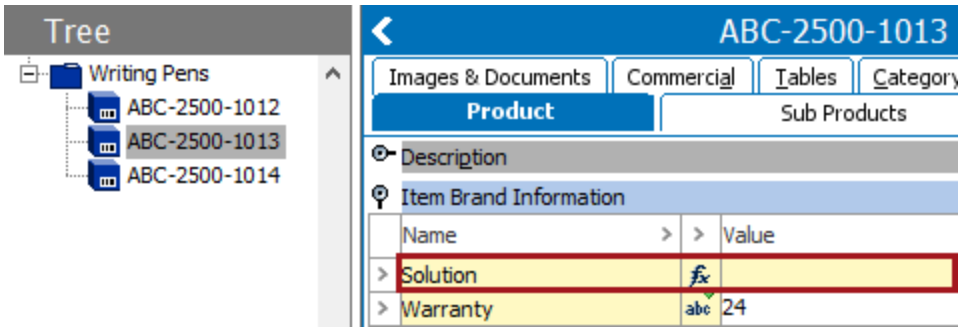
Results

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

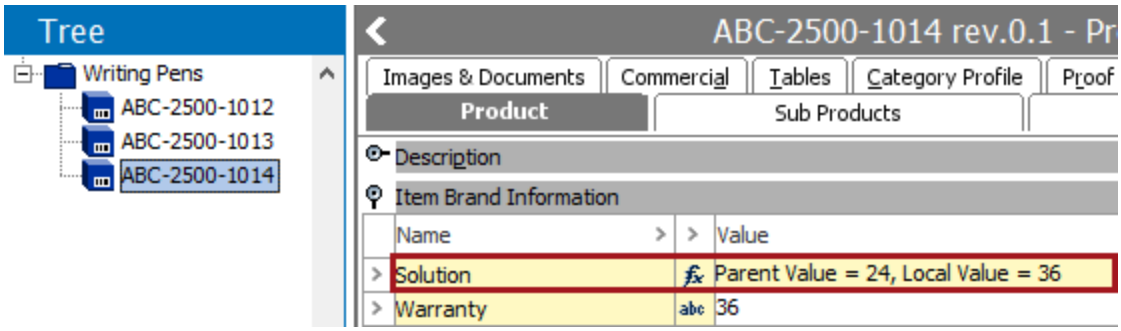
The first child product has inherited the value for Warranty from the parent (indicated by the green triangle), so the calculated attribute result is blank.



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.



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.



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

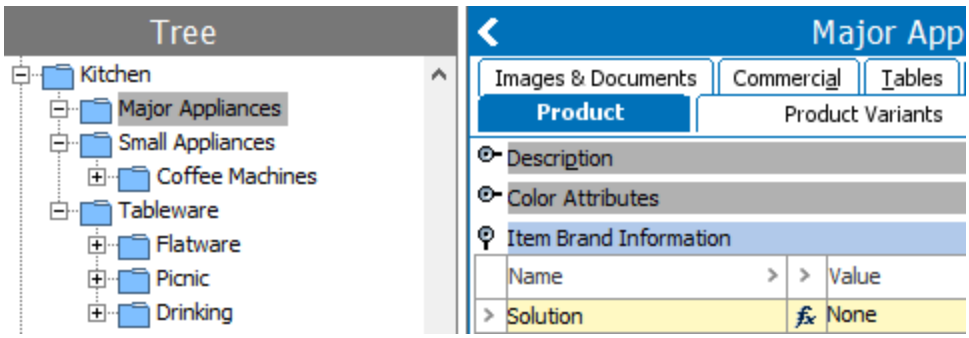
```
{childlist := subproducts()}

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

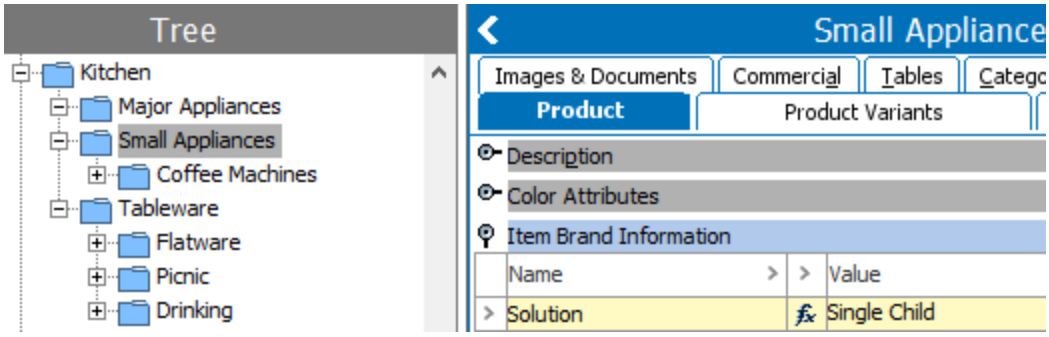
For information on adding a value template, see the **Attribute Editor** section of the **Function Editor** documentation.

Results

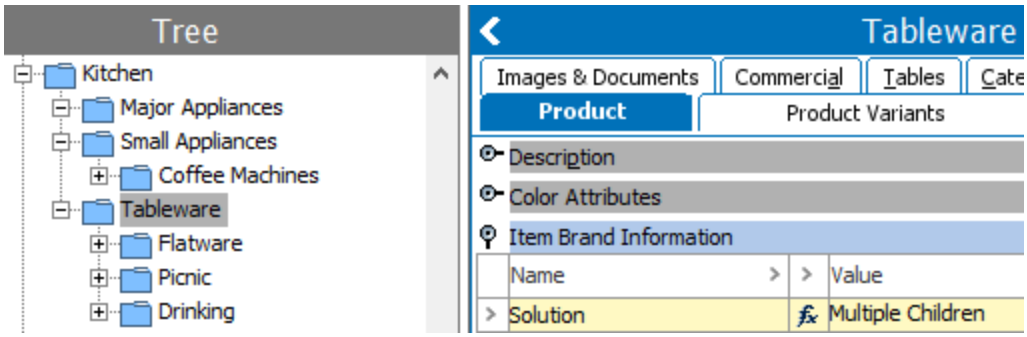
The first node selected has no children, so 'None' is displayed.



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

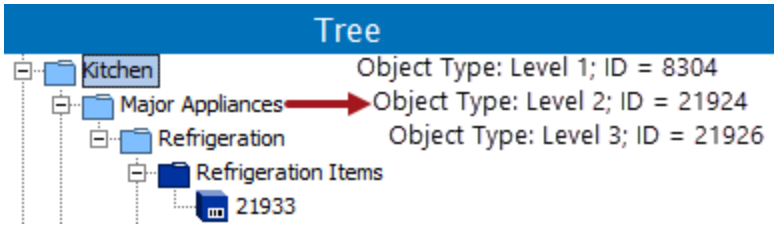


The third node selected has more than one child, so 'Multiple Children' is displayed.



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()'),

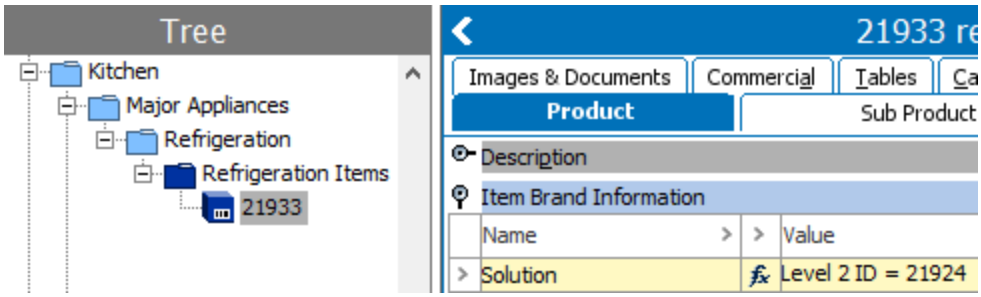
getid := listitem(filter(plist, 'exact(objtype, listitem
(idlist,index))'),1)

}

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

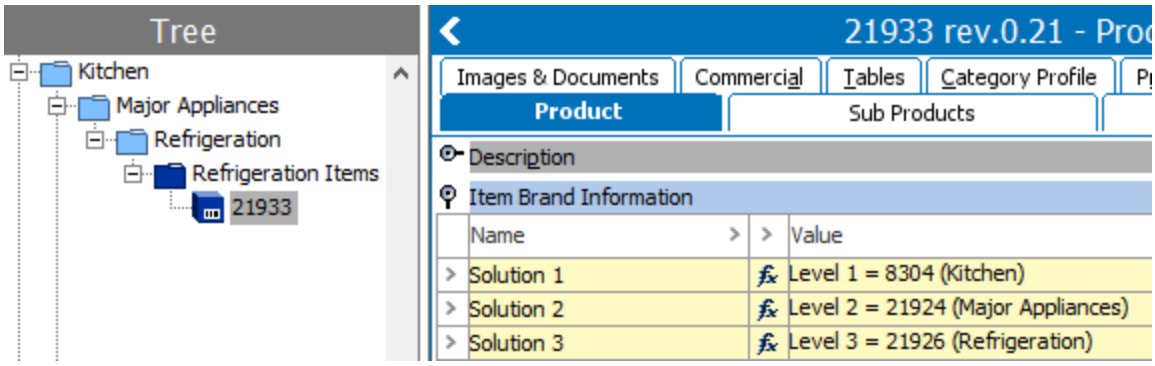
The following two additional variables are added to the previous value template: nlist and getname; and the concatenate formula has been modified accordingly.

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,")")
```

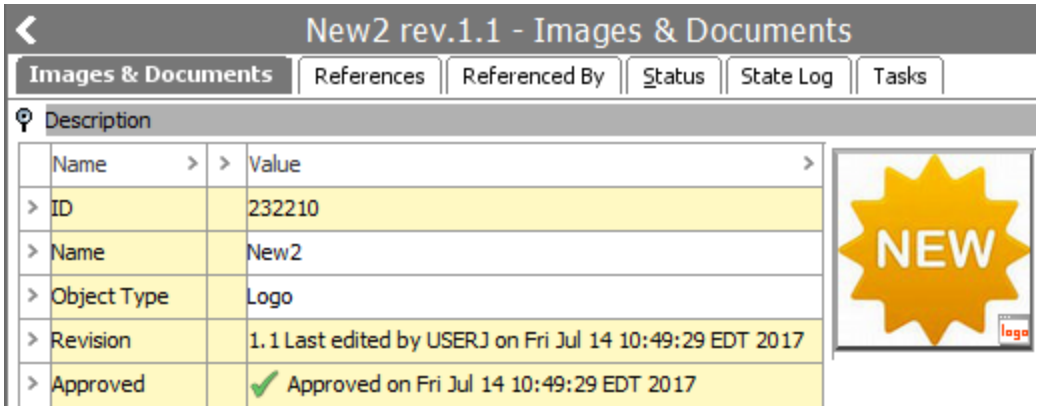
Results

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 - Proc
Images & Documents
Commercial
Tables
Category Profile
Product
Sub Products
Description
Item Brand Information
Name
Value
Solution 1
Level 1 = 8304 (Kitchen)
Solution 2
Level 2 = 21924 (Major Appliances)
Solution 3
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.



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 Systems 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

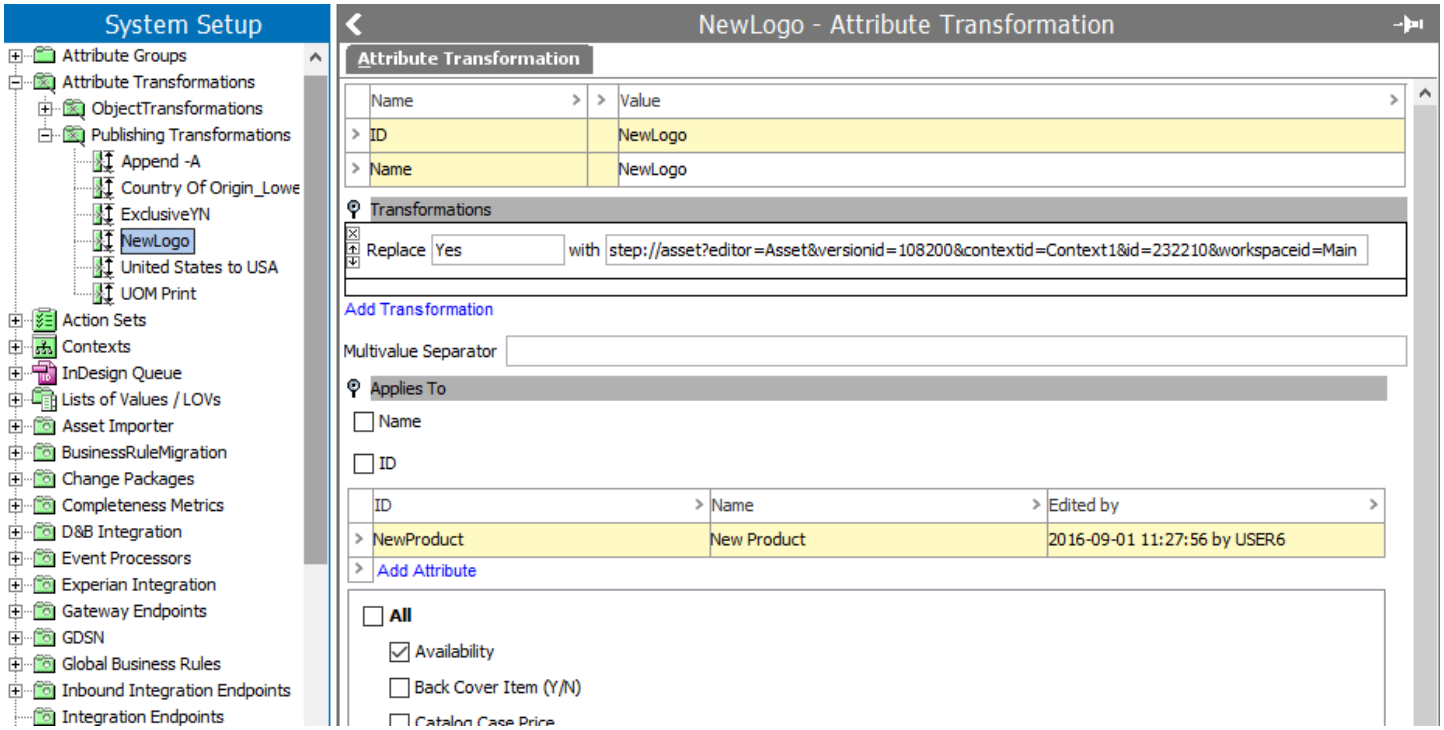
```
{x := iterate(subproducts(), 'value("New Flag")')}
if(listcontains(x,"Yes"), "Some New","None New")
```

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.



- Value Templates 1, 2, and 3 could have taken advantage of using a Stibo Systems 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 Systems tag. In this example, the tag <NewLogo/> would be converted to the appropriate

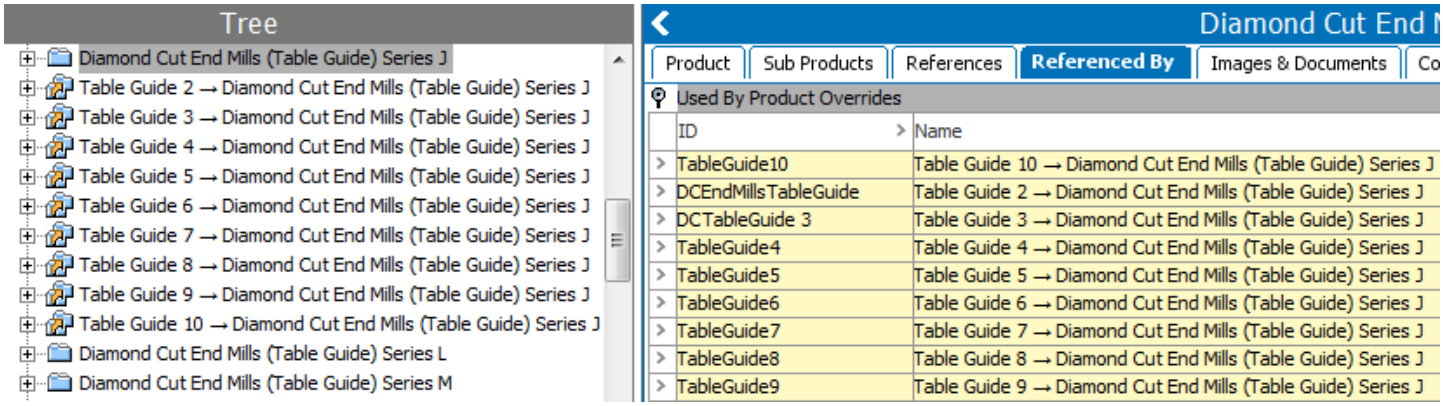
image call out, the same way as the transformation.

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

Note: Alternatively, attribute transformations can be applied to text within STEP Publisher product templates to transform text into an image, such as changing the word 'New' to a logo. This method does not require calculated attributes. See the **Attribute Transformations in STEP Publisher** topic in the **STEP Publisher** documentation for more information.

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 IDs of parent products.



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

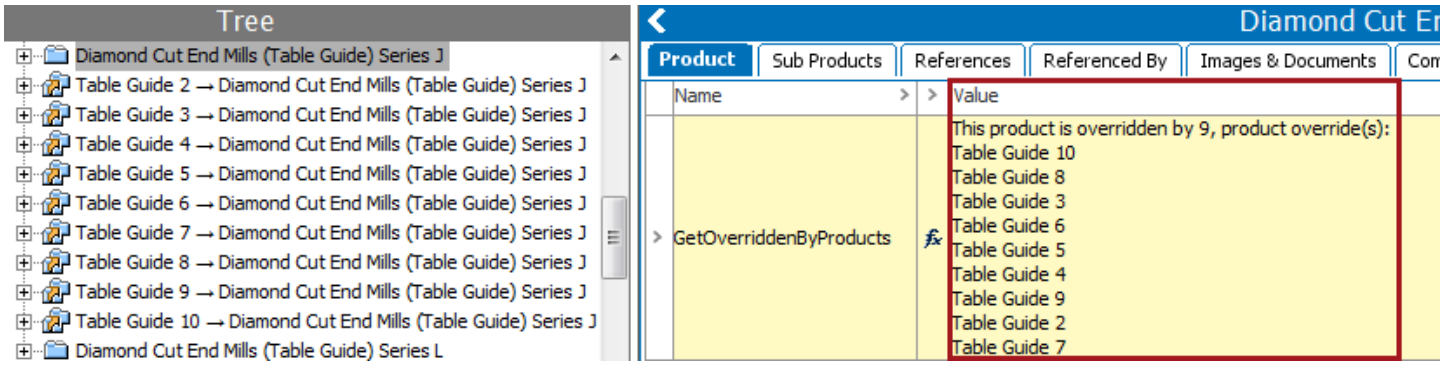
The formula for this calculated attribute could be written:

```
{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:

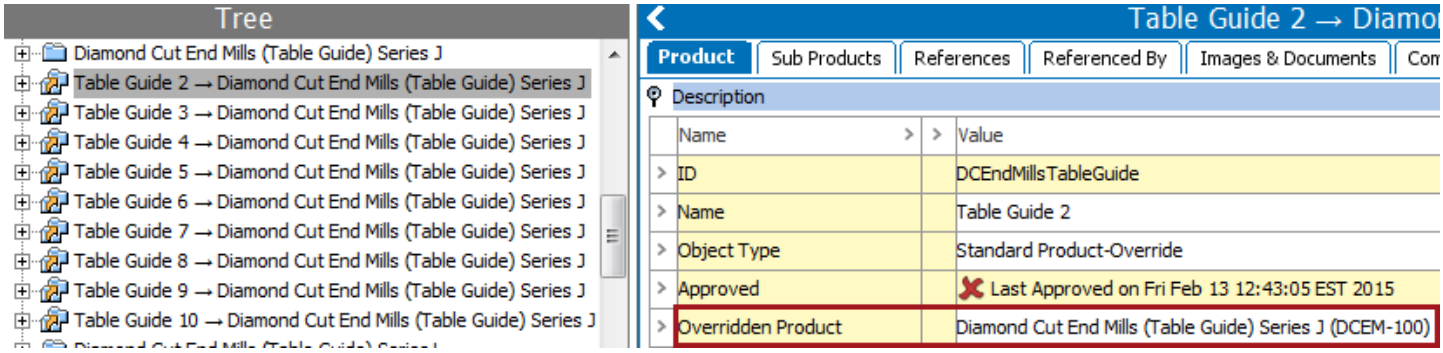
- Adding a new line using the Enter key or adding \n" allows the results to be displayed on individual lines.
- The LISTLEN() function gets the number of override items.
- The LISTLEN() function, used with STEPNAME(), gets the names of the override items.
- The final output is assembled using a text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

Results



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.'



Value Template

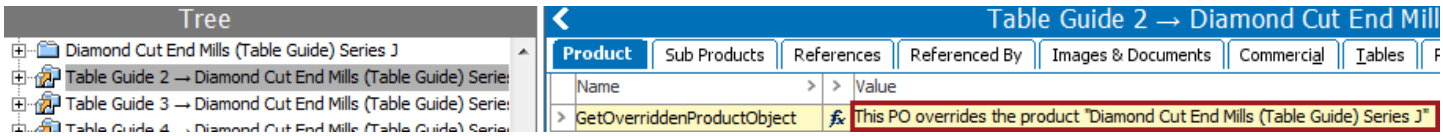
A calculated attribute on the product override could be written with the following formula:

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

This formula uses the following elements:

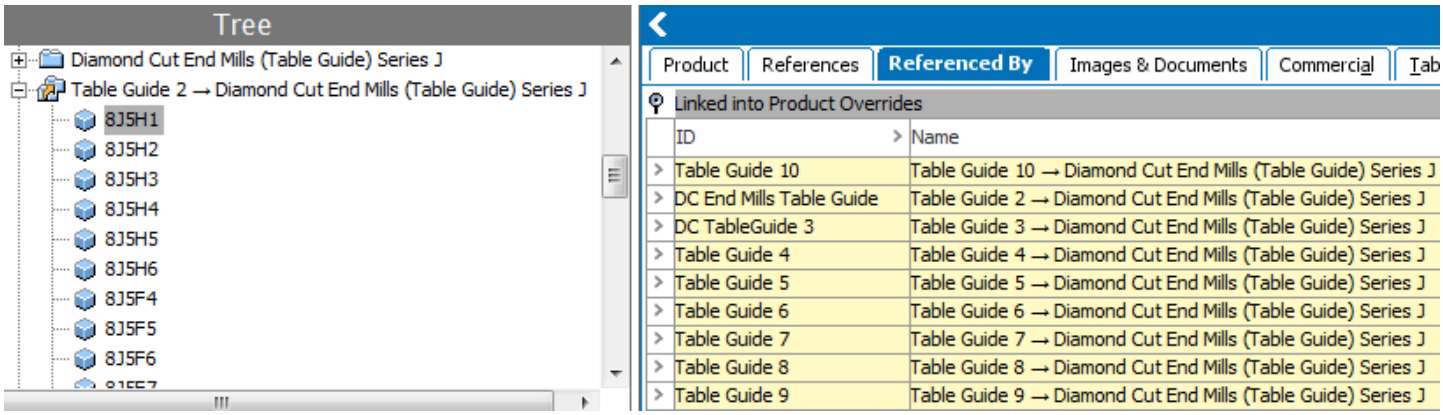
- The ITERATE() function gets information from an object other than the currently selected object.
- Since the maximum list length is 1, the list is displayed directly.
- The final output is assembled using a text string and the name of the product that is overridden.

Results



Get Product Override Parents

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

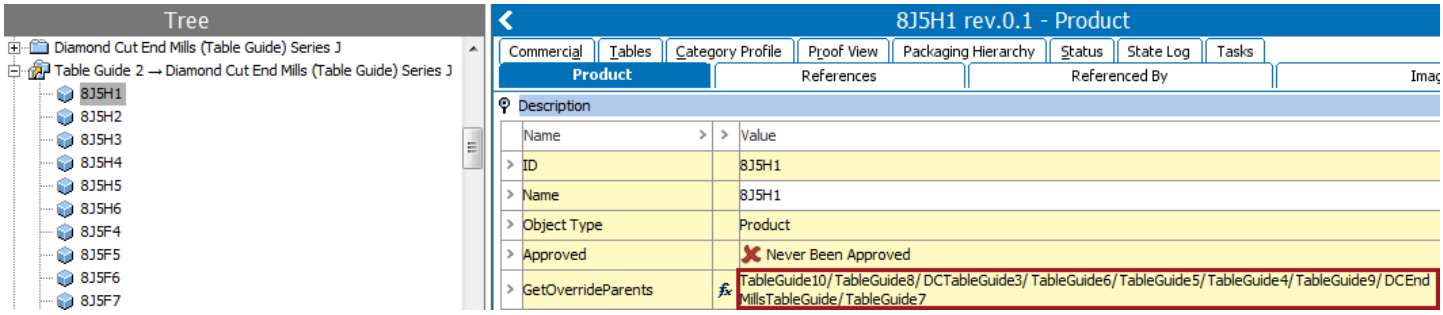


Value Template

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



Product Description Scenario

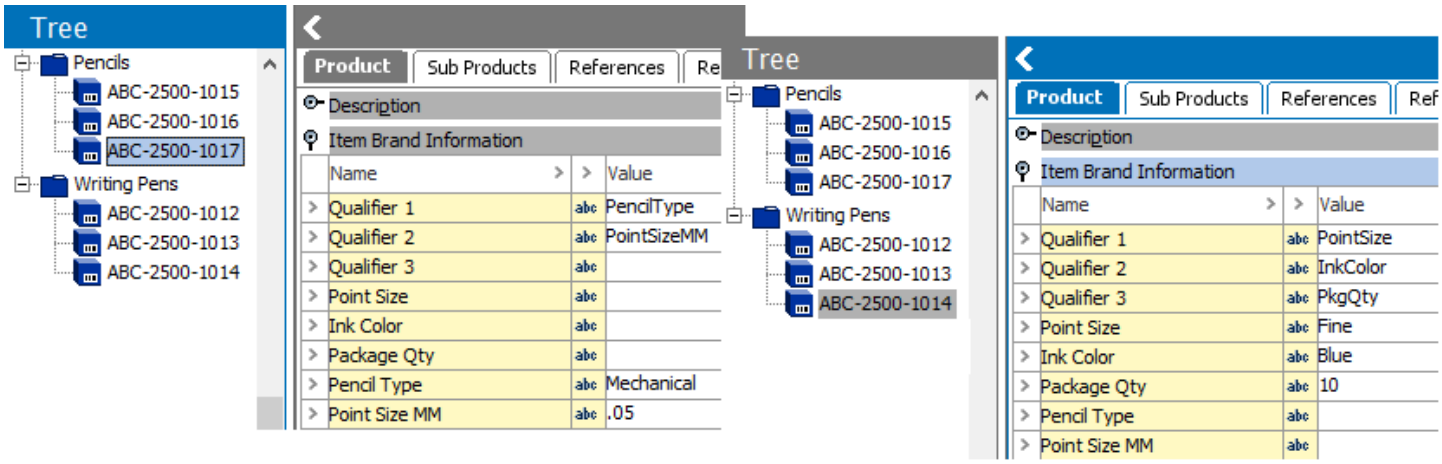
A calculated attribute can be used to automate the process of building dynamic product descriptions. Using existing attributes, and based on different product lines, in the example below we will generate a product description that is comprised of a basic 'stem' attribute, and concatenated with additional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

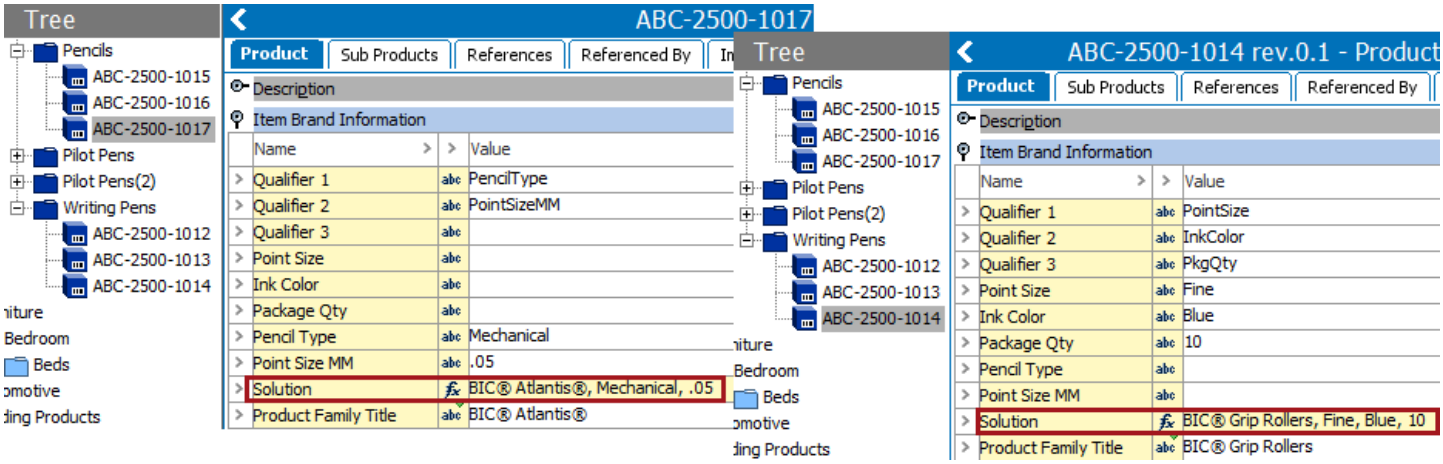
Product Line	Pencils	Writing Pens
Product Family Title ('stem')	BIC® Atlantis®	BIC® Grip Rollers
Point Size (PointSize)		Fine
Point Size MM (PointSizeMM)	0.5	
Ink Color (InkColor)		Blue
Package Qty (PkgQty)		10
Pencil Type (PencilType)	Mechanical	

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



Results

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



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).
2. 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.

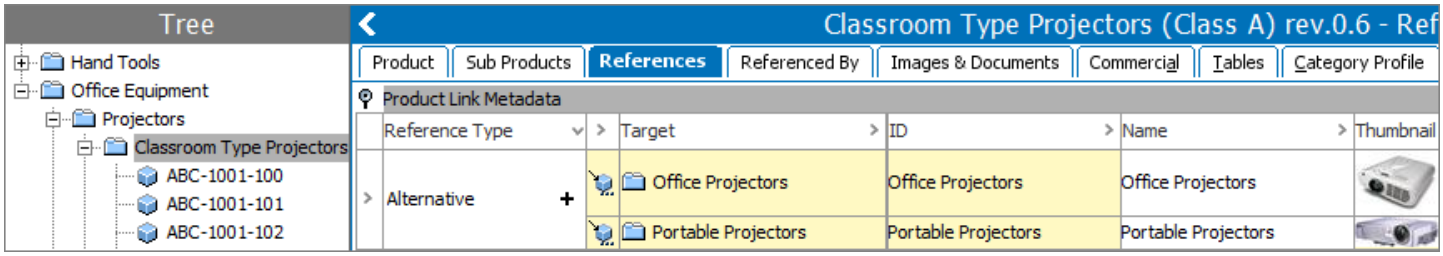
Reference Scenario

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.



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

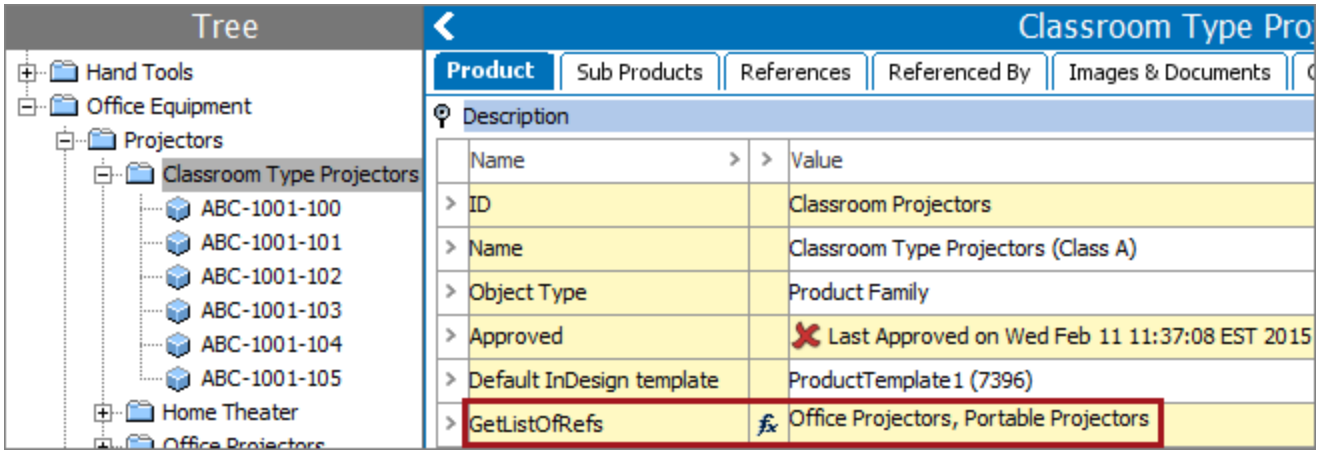
To break this formula down, the first ITERATE() function is used to pull information that is not on the current product. The second ITERATE() function (shown as bold above) returns a list of 'labels' for all of the references that Classroom Type Projectors has made to other products (i.e., reference 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()')
```

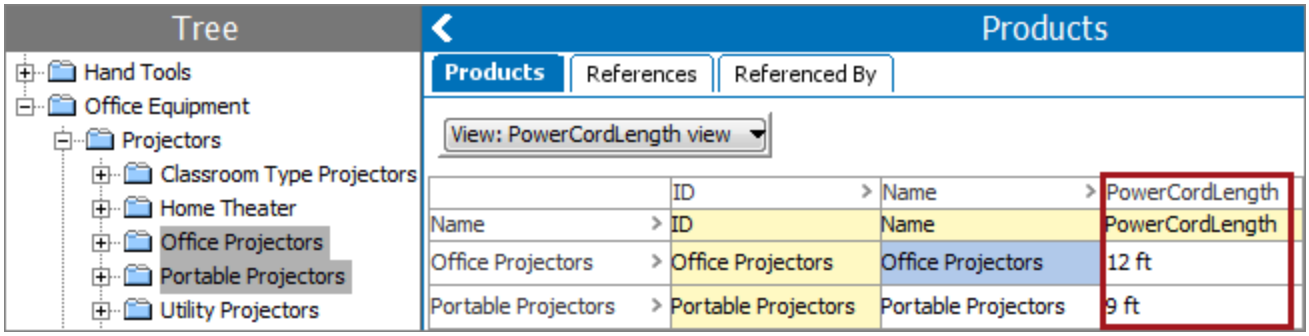
Results

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



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:



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.

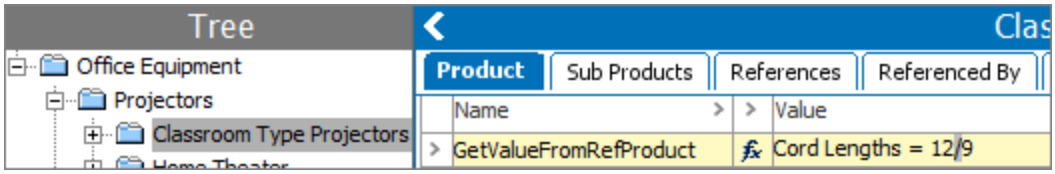
```
{i := iterate(iterate(references("product", "Alternative"),
'referencetarget()'), 'value("PowerCordLength")')

}
```

```
concatenate("Cord Lengths = ",list2multivalue(i))
```

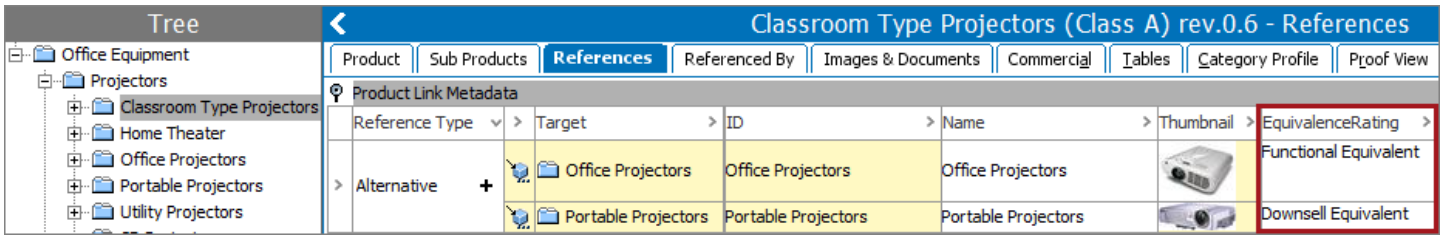
Results

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.



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:



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),

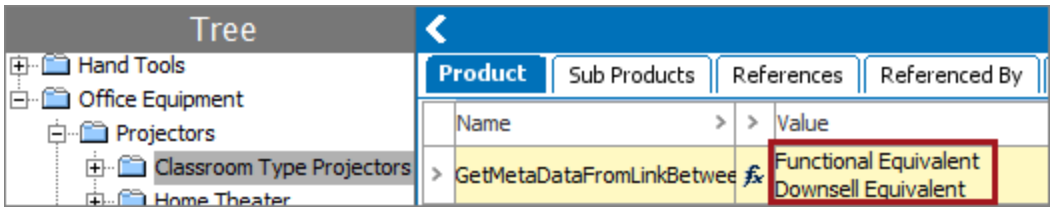
z := substitute(z, "<multisep/>", "\n ")

}

concatenate(z)
```

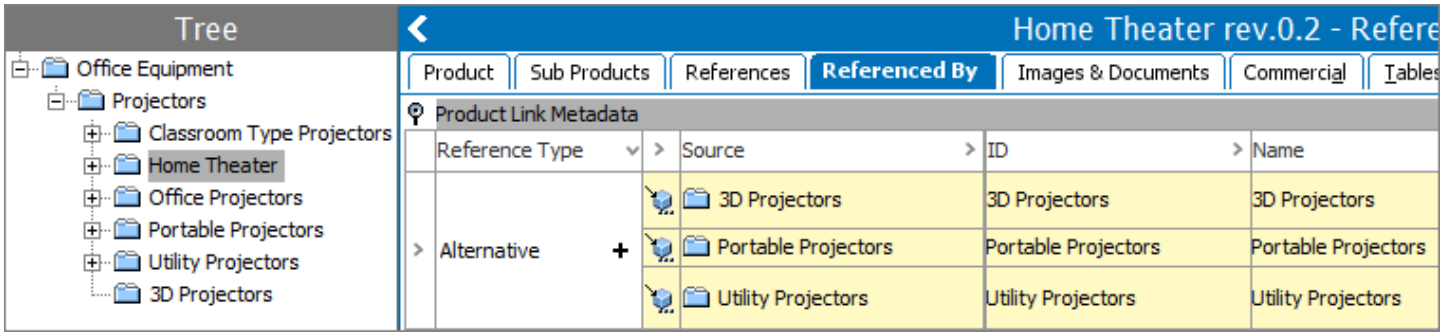
Results

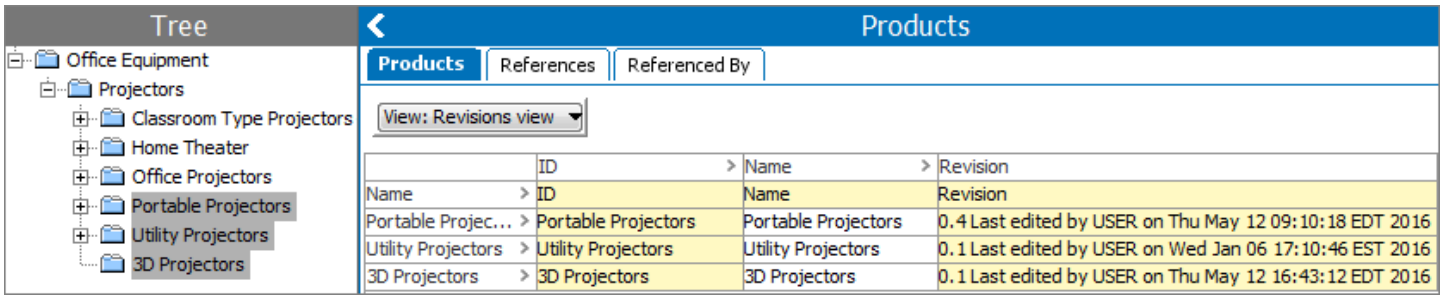
The default separator used by the LIST2MULTIVALUE() function is converted to a carriage return, so that the workbench displays a new line for each entry within the same value field.



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.





Value Template

The following calculated attribute formula could be written:

```
{refnames := iterate(iterate(referencesto("product", "Alternative"),
'referenceowner()'), 'stepname()'),

editdates := iterate(iterate(referencesto("product", "Alternative"),
'referenceowner()'), 'revisioneditdate()'),

combo := iterate(refnames, 'concatenate(item, " Last edited on ",
listitem(editdates,index))'),

fin := substitute(list2multivalue(combo), "<lt/>multisep/<gt/>", "")

}

concatenate(fin)
```

Notes

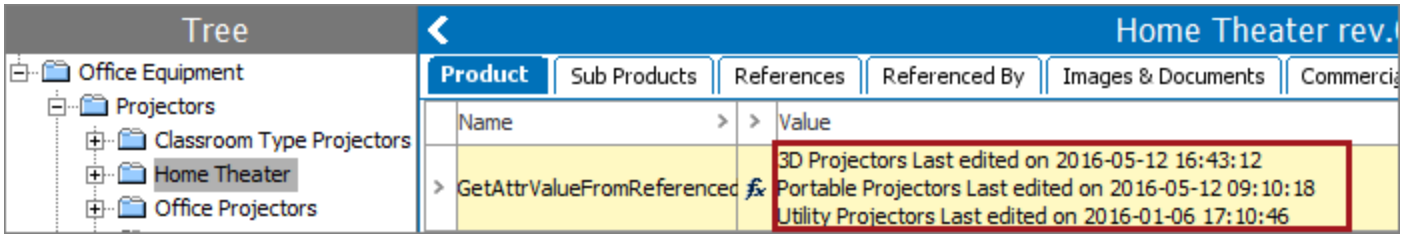
This formula uses the following elements:

- 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.

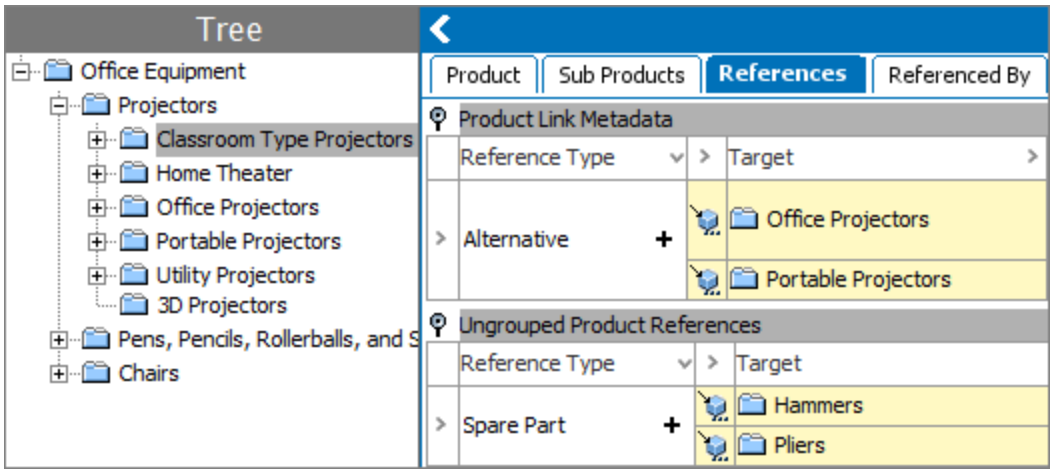
Results

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



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.



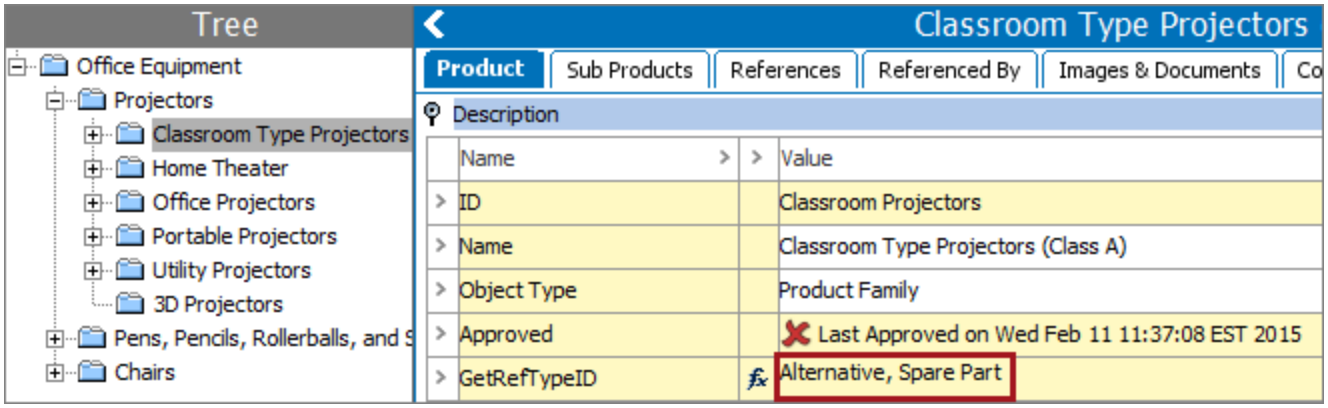
Value Template

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.



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/>", "")

}

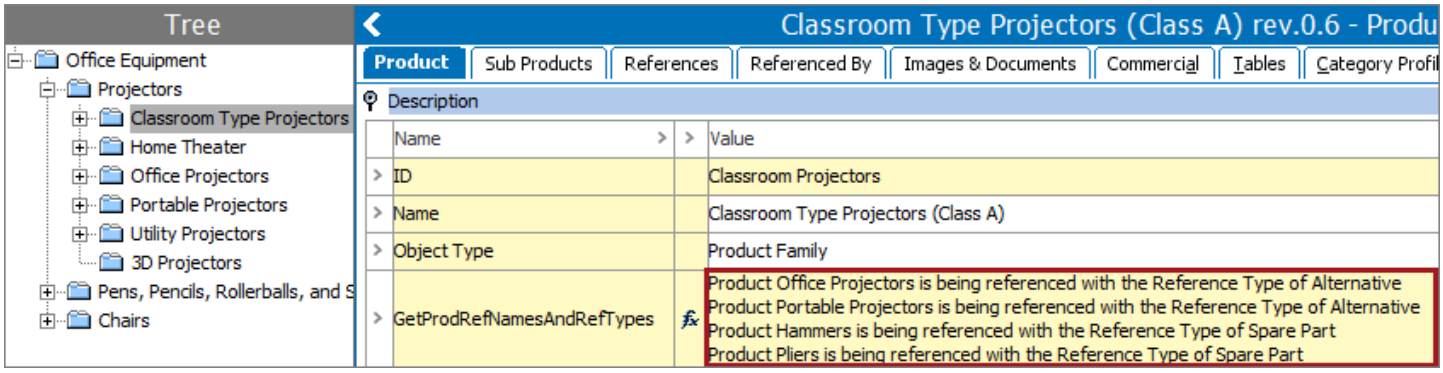
concatenate(fin)
```

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.



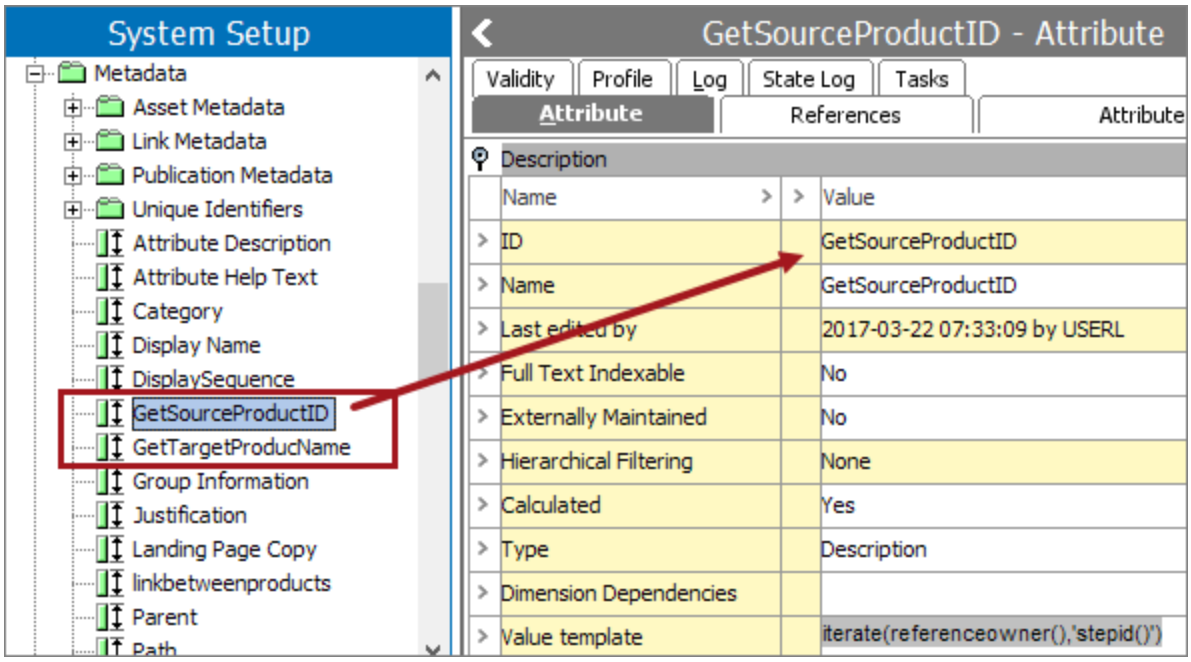
Get Reference Source or Target Product Information on the Link

The calculated attribute can also be used to show metadata on the References link. A user could use it to show the Source product and the Target product for the link.

Value Template

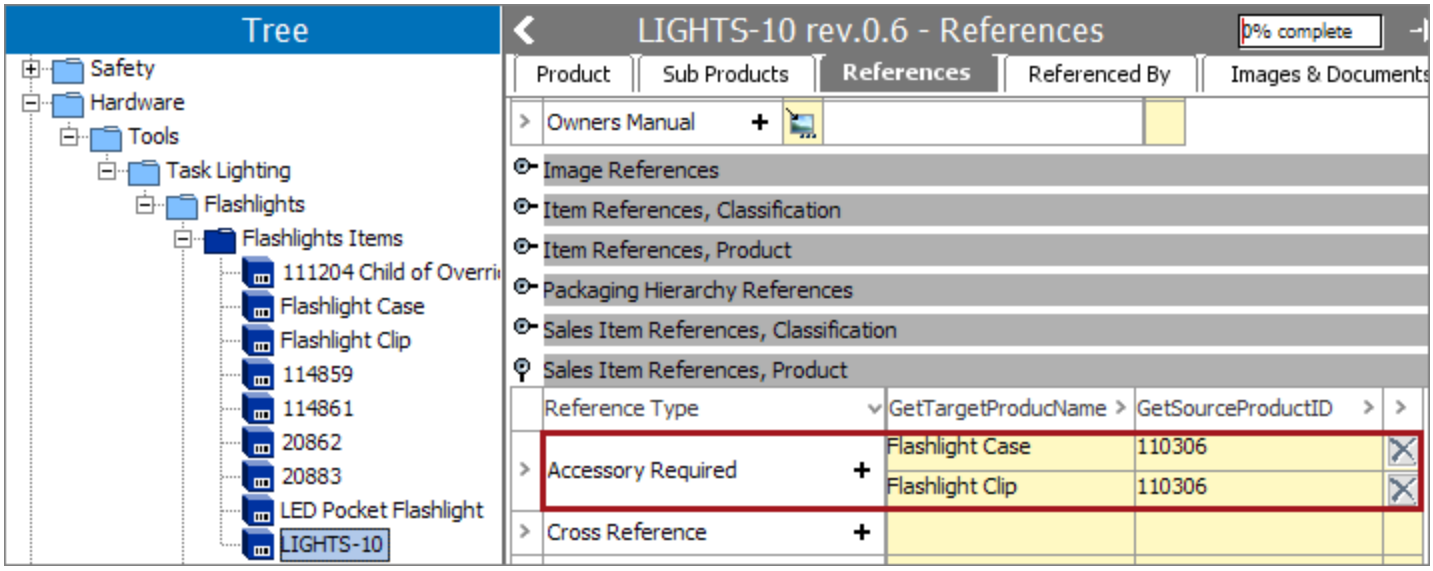
```
iterate(referenceowner(),'stepid()') // to get the Source product ID

iterate(referencetarget(),'stepname()') // to get the Target product
name
```



Results

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



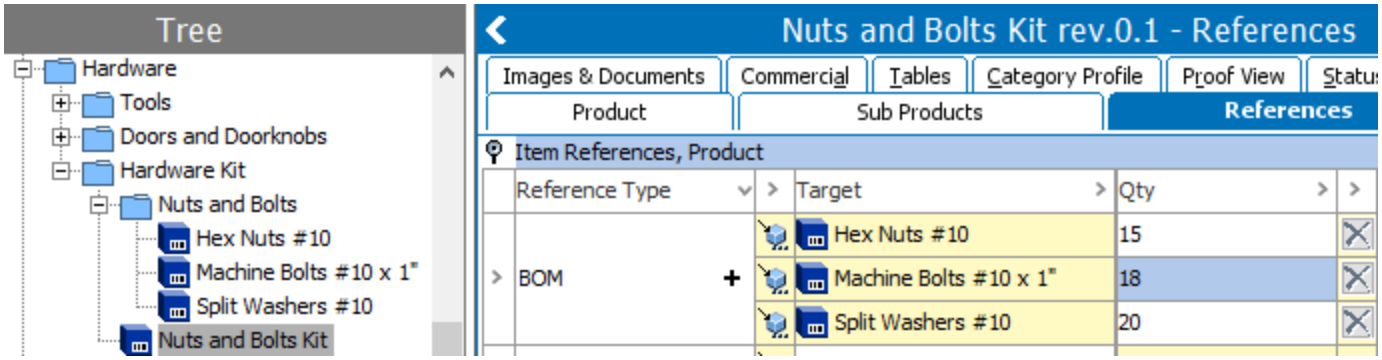
Total Cost of Materials Scenario

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.'

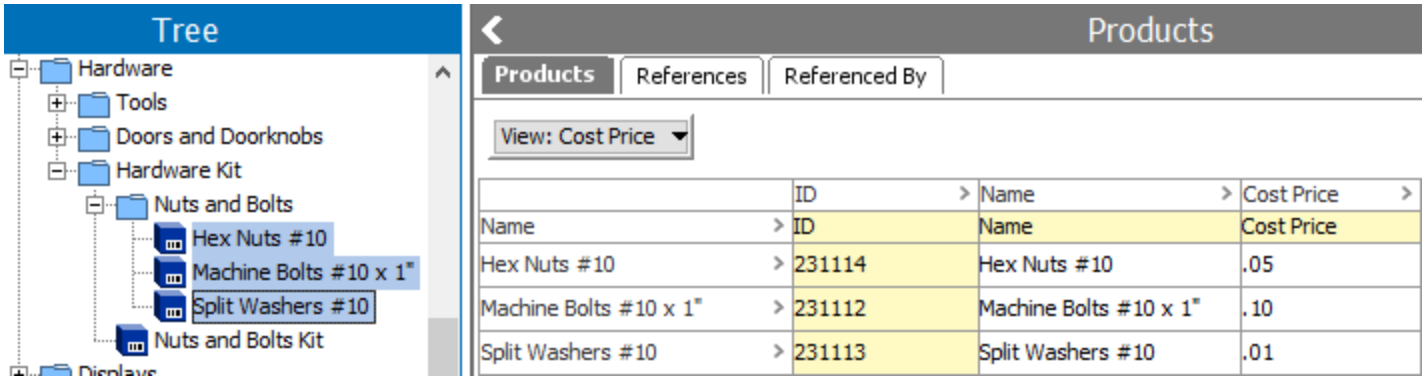
The most popular approach is to link the source product (i.e., the assembled product) to the various part products that go to make up that assembly (target product), using a specific reference type, for example, 'BOM.' Then, the cost of each of the part products is usually a standard product attribute, 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.



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



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

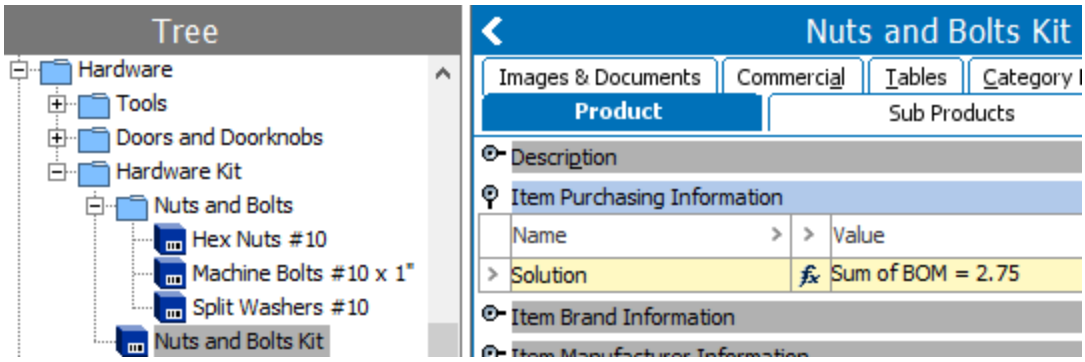
The summation of the entries in TotalCost is a little unusual in that the variable 'dummy' is not required at all, but the variable 'total' is what is needed. Although the contents of the variable 'dummy' are never used, in calculating that value, it is used to calculate the variable 'total' which is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

Results

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

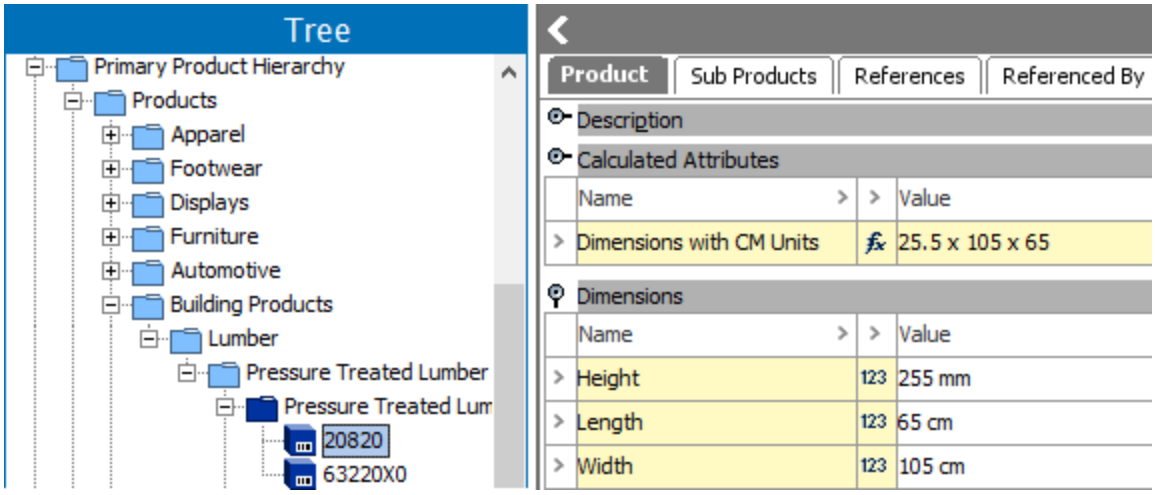
- Hex Nuts: .05 x 15 = .75
- Machine Bolts: .10 x 18 = 1.80
- Split Washers: .01 x 20 = .20

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 & Documents
Commercial
Tables
Product
Sub Products
Description
Item Purchasing Information
Name
Value
Solution
Sum of BOM = 2.75
Item Brand 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.



Prerequisites

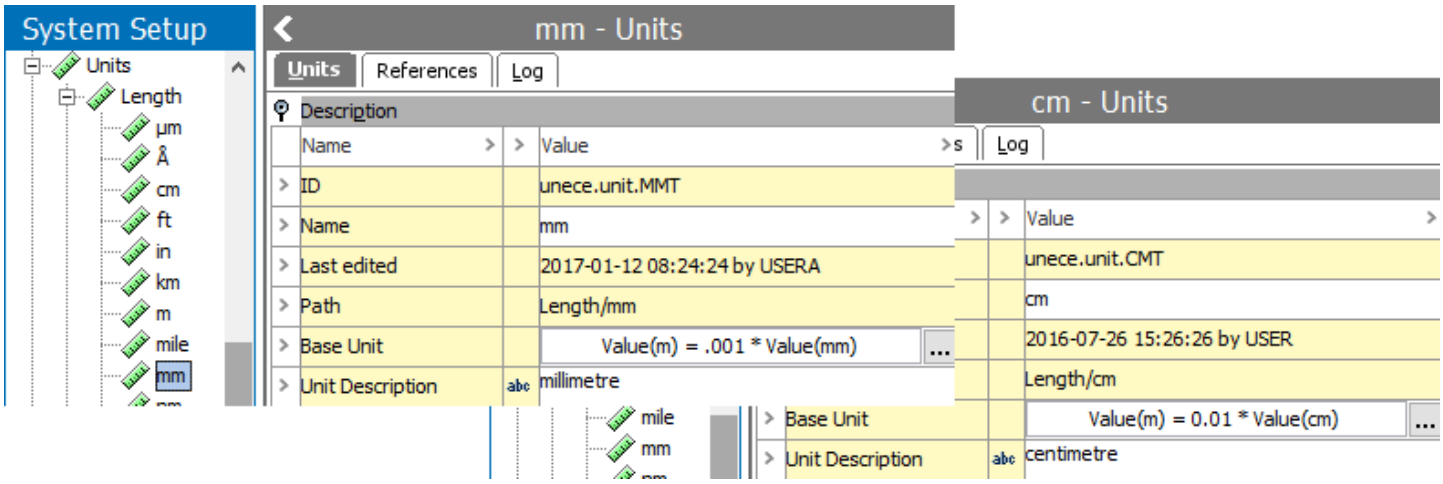
For more information on units, see the **Units** topic in the **Getting Started / User Guide** documentation.

Approach

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

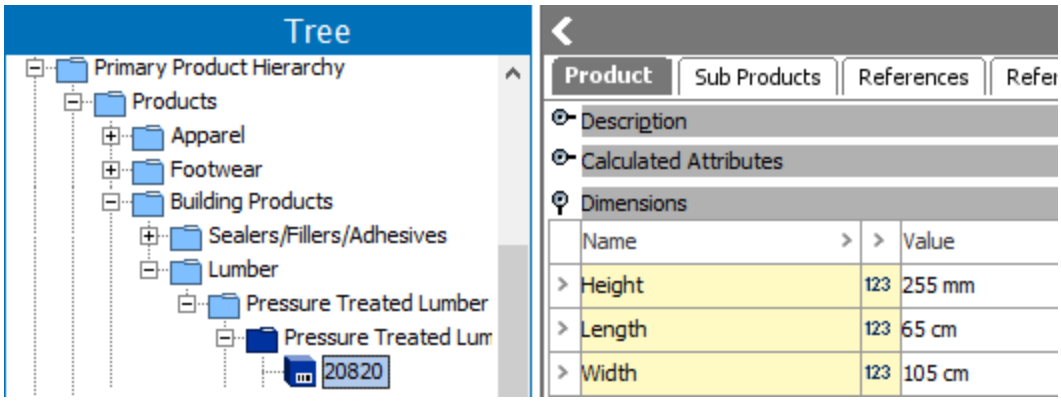
Configure Units for Conversion

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.



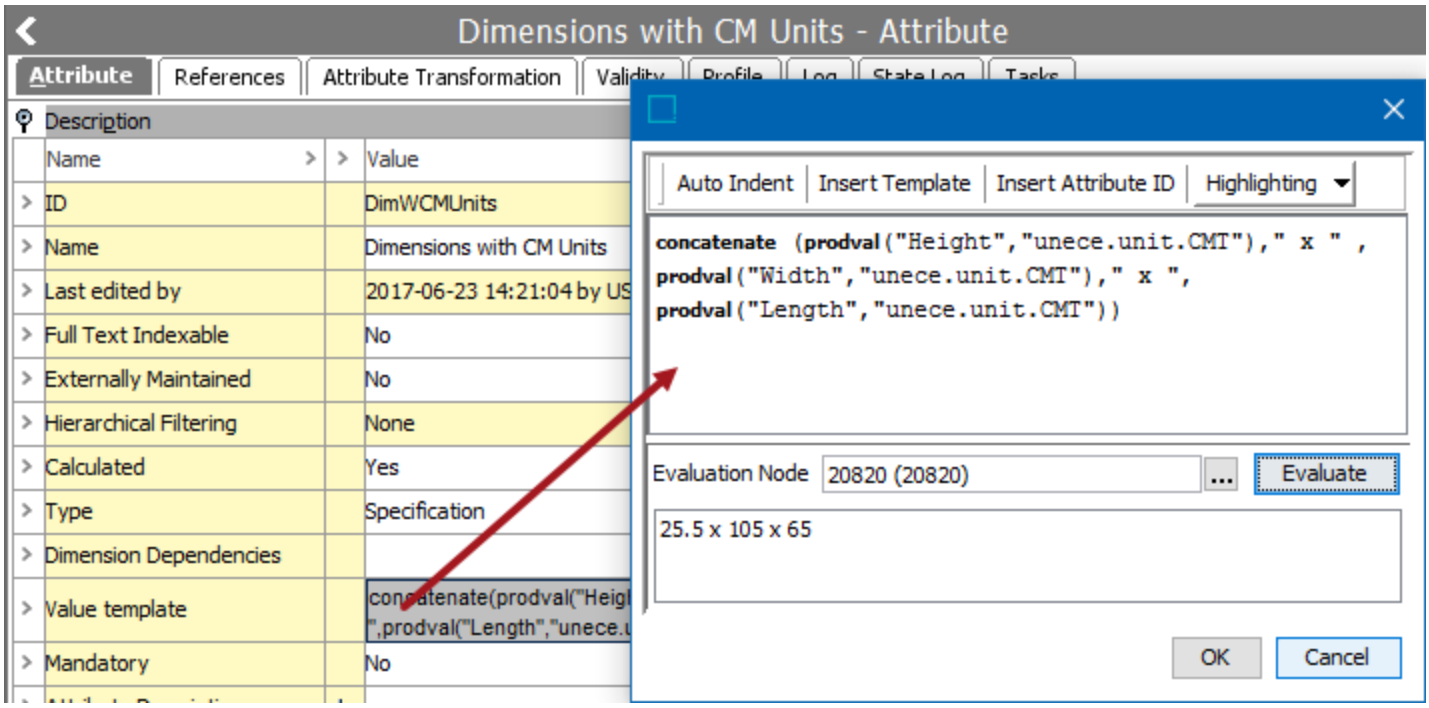
Concatenate and Specify Required Unit

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



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.



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.

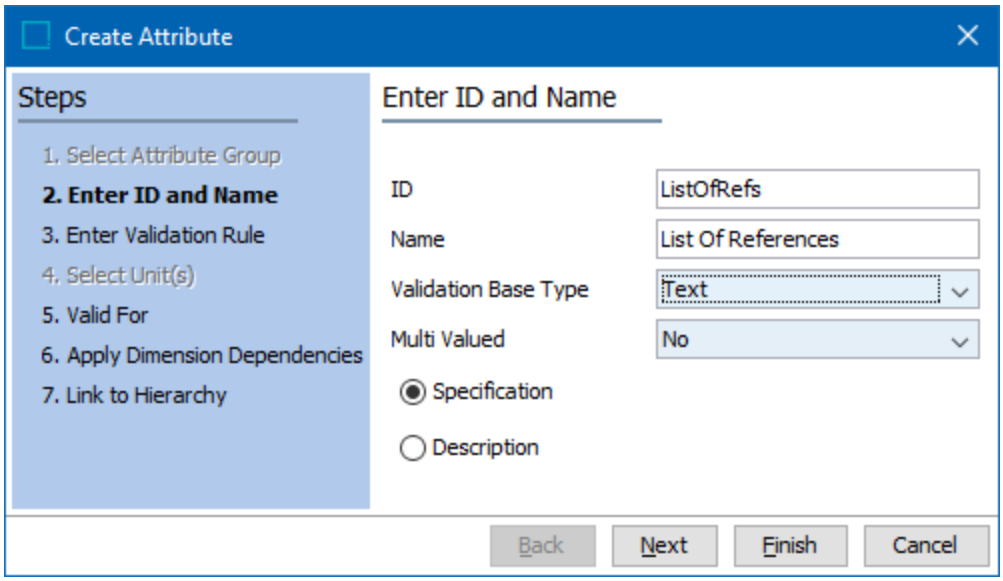
Creating a Calculated Attribute

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 tool. 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:

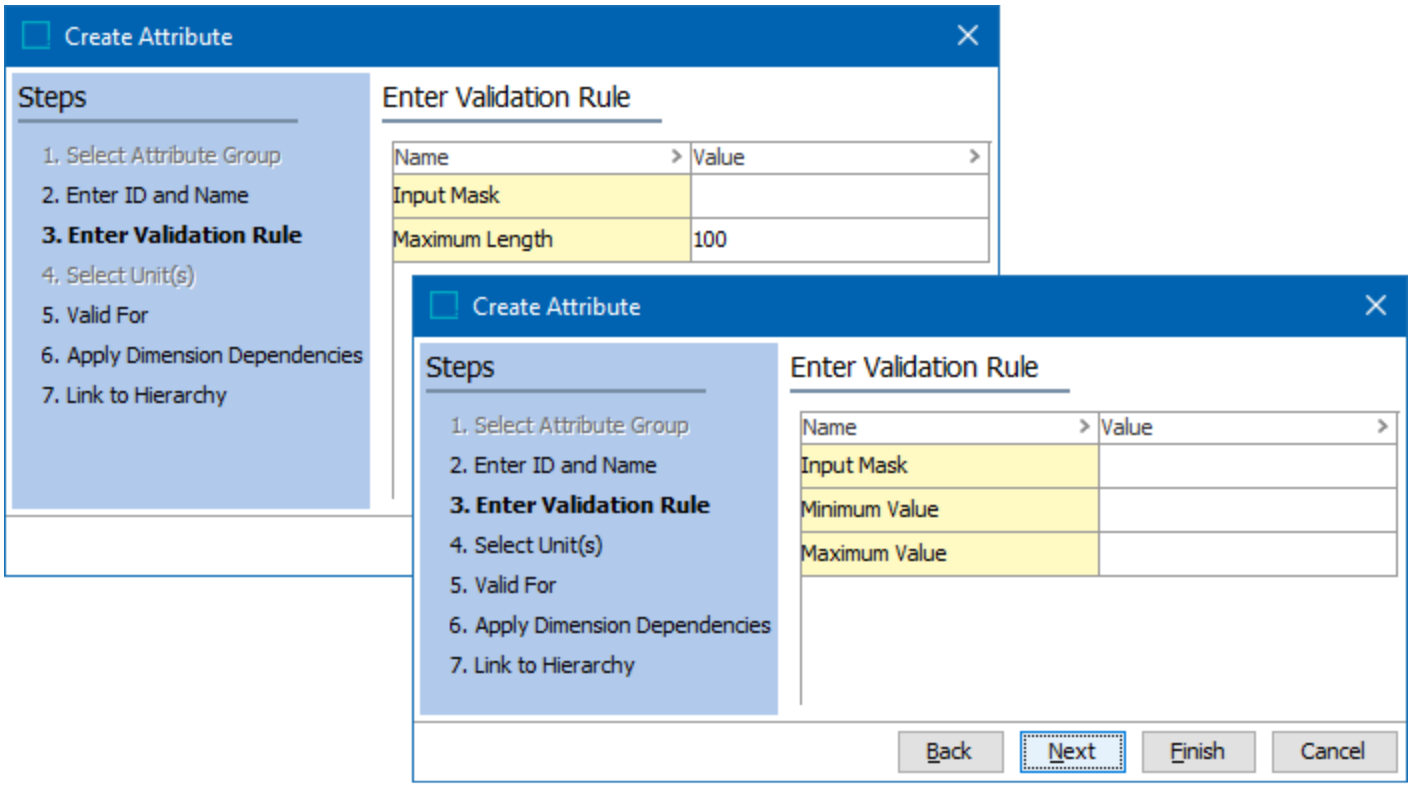
   - Type an **ID** and **Name** for your new attribute.
   - Set **Validation Base Type**, keeping in mind that the assigned type only takes effect when the calculated value is manually overridden. For more information, see the **Attribute Validation Flipper** section of the **Setting up a Calculated Attribute** 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.



   - 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):

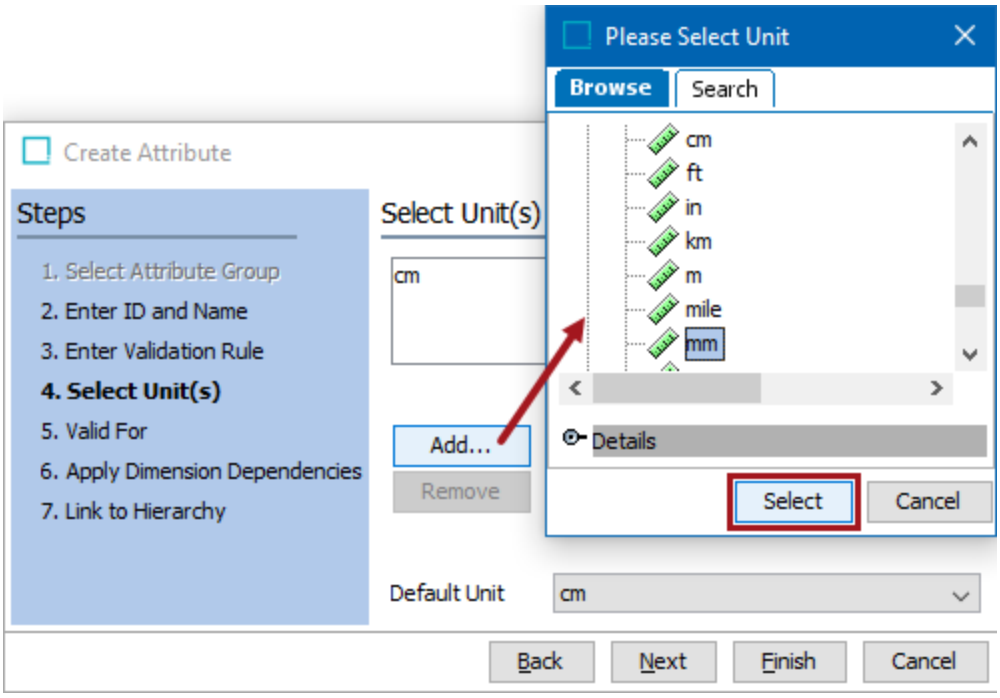
   - Type an **Input Mask** if required. For more information, see the **Input Masks** section of the **Validation Rules** documentation and the **Restrict Override on a Calculated Value** section of **Overriding and Restoring Calculated Attribute Values**.
   - For Validation Base Types of Embedded Number, Fraction, Integer, Number, Numeric Text, or Numeric Text (exclude tags), 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.



   - Click the **Next** button.

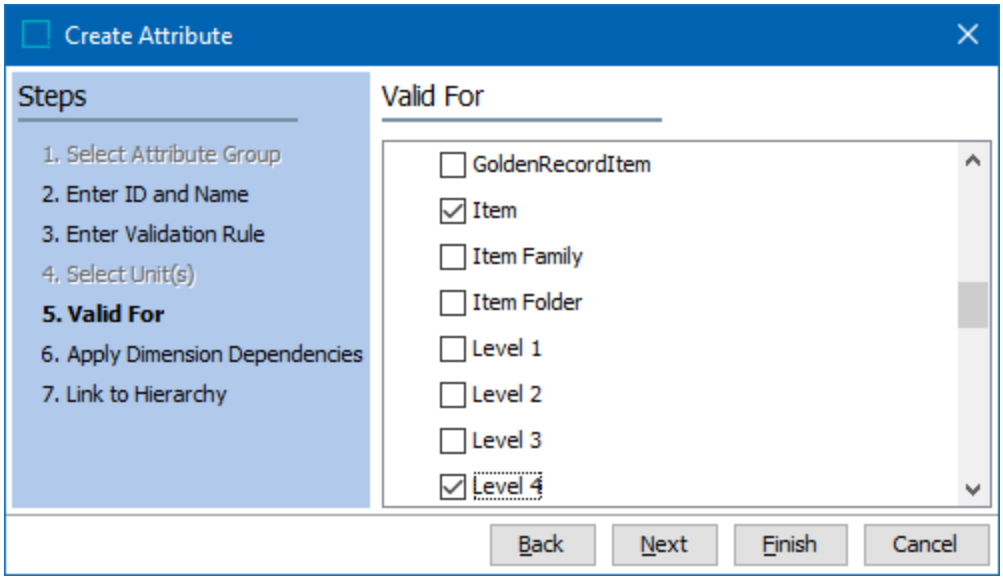
5. The **Select Unit(s)** step of the wizard is only enabled when the Validation Base Types is Embedded Number, Fraction, Integer, Number, Numeric Text, or Numeric Text (exclude tags). Add the following data if required:

   - 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.



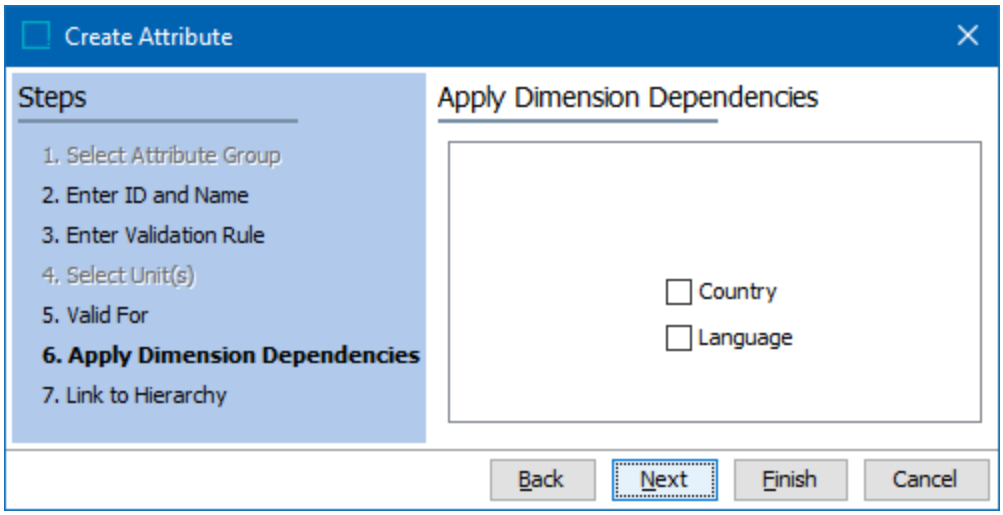
- When a unit exists in the text box, the **Remove** button is enabled. Click it to remove the selected unit from the text box.
- Use the **Default Unit** dropdown to set the unit that will be automatically applied to the calculated attribute value.
- Click the **Next** button.

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.



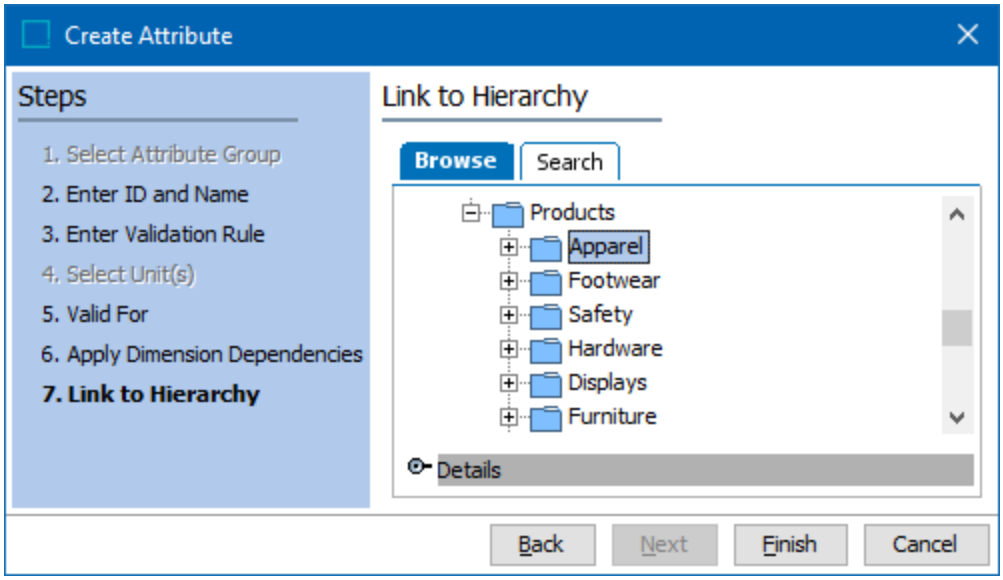
- Click the **Next** button.

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.

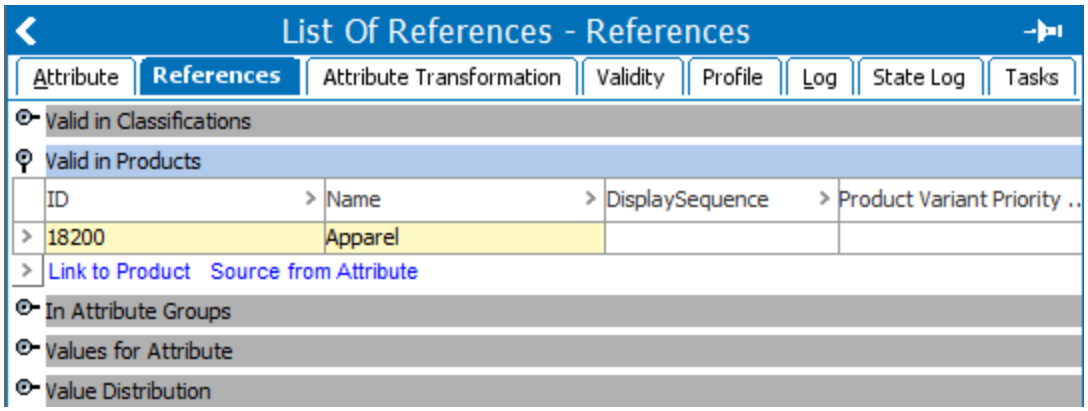


- Click the **Next** button.

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.

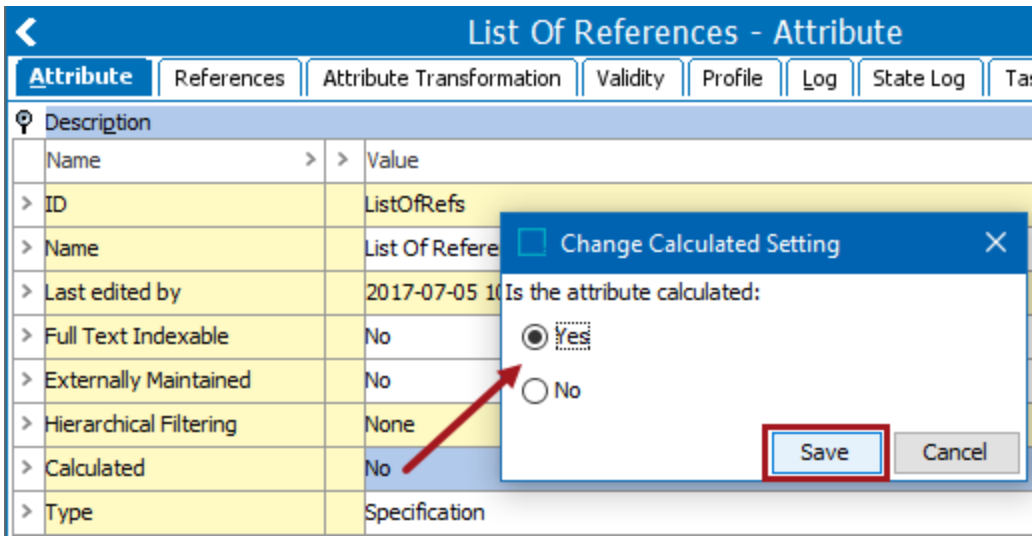


This creates an entry on the attribute editor References tab, under the Valid in Products flipper as shown below.

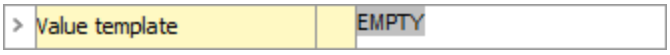


- Click the **Finish** button on the wizard. The attribute editor is displayed for the new attribute in the attribute group. Continue with the manual configuration steps below to make the attribute calculated.

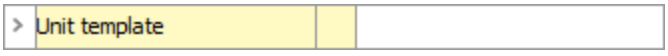
9. In the attribute editor, on the Attribute tab, within the Description flipper, double click the **Calculated** field to display the Change Calculated Setting dialog. Click the **Yes** radio button, and click the **Save** button.



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



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



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

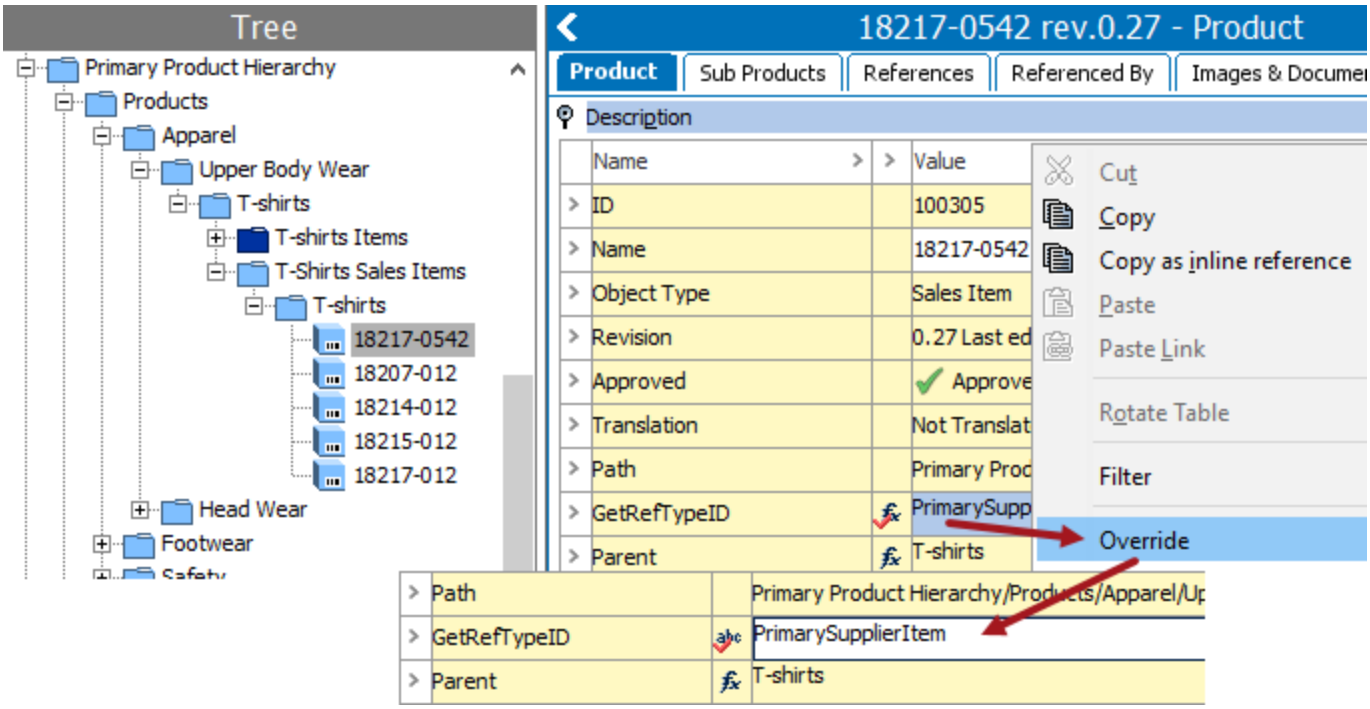
When a calculated attribute displays on an object, the value is read only. Occasionally, the calculated value for a specific product, even though correct, may not be what is desired. Below are options for overriding calculated attributes, restricting override, removing override values, and restoring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

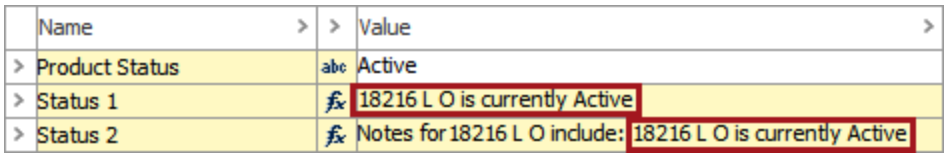
To manually override the value on specific product (and child products for specification attributes, if the value is inherited), right-click **the calculated value** and click **Override**.



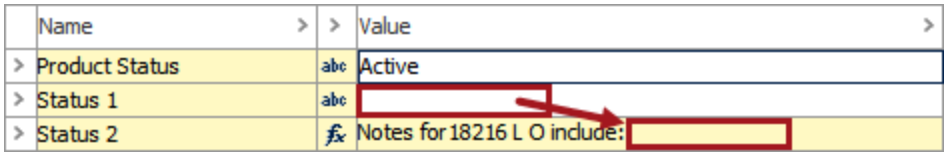
The parameter can then be edited, allowing you to enter the value to replace the calculated value. If values existed prior to clicking Override, then those values become static text. When static text has been defined, it is

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.

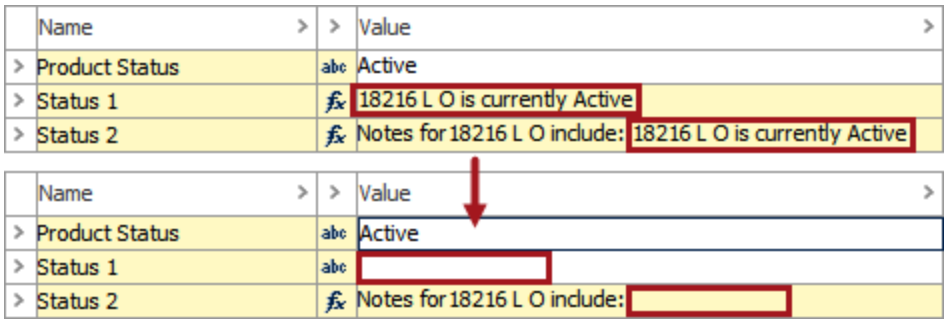
Another option is when a space added as a value is used to override a calculated attribute value. In the following image, Status 1 is a calculated value that uses data from Product Status, in this case the value of 'Active.' Status 2 adds the value of Status 1 to some text.



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



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



Restrict Overrides of a Calculated Attribute Value

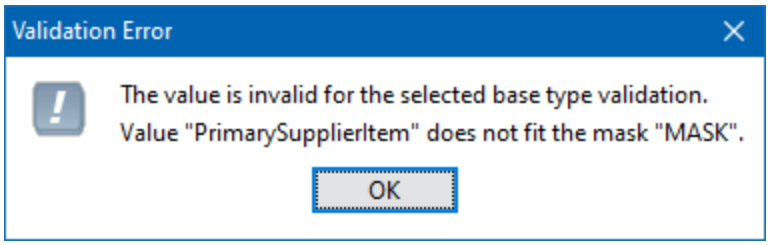
Calculated attribute values can be overridden as described above, and can also be overwritten via import.

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.

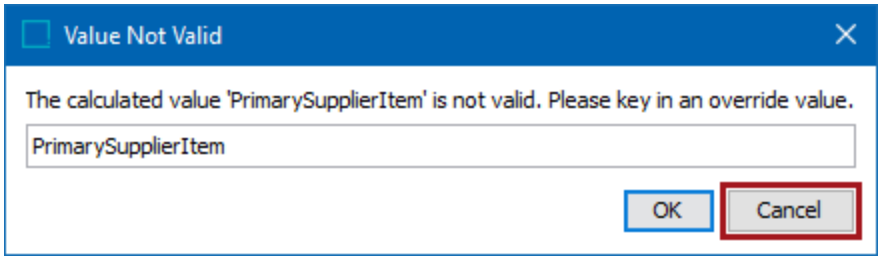
Continuing with the override example above, a mask of 'MASK' has been added to the calculated attribute named 'GetRefTypeID'.

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** topic in the **Data Exchange** 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, 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:

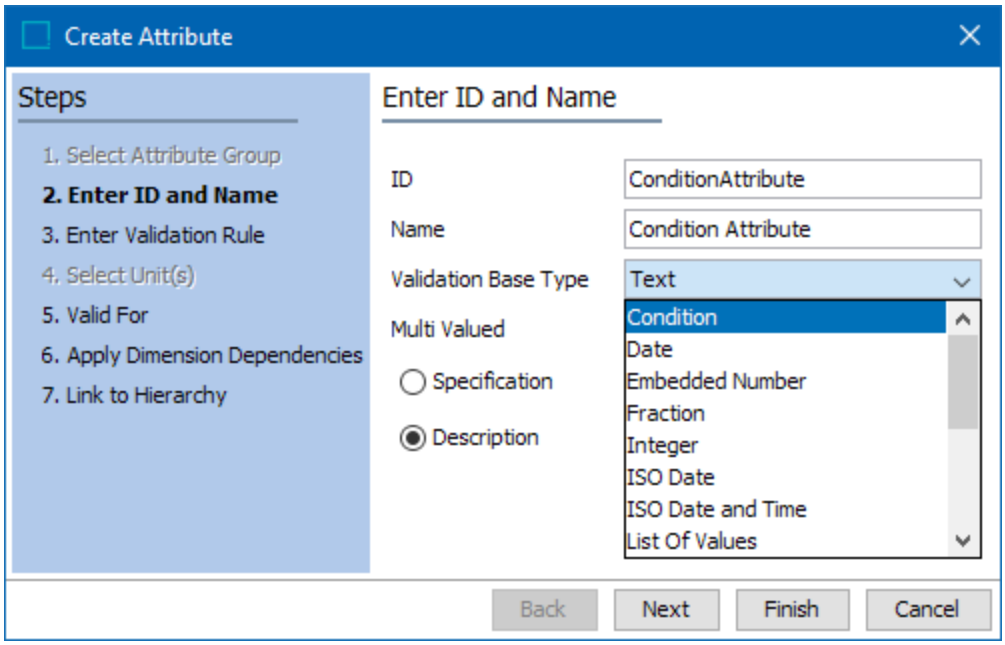
- **Conditional Attribute** - This attribute appears as a metadata field on Description attributes and on Specification attribute links in the Product Hierarchy. Conditional expressions are stored in this attribute.
- **Conditional Expression** - Validity rules for the Conditionally Valid Attribute. 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 the attribute that holds the logic for the Conditionally Valid Attributes. Therefore, any number of Conditionally Valid Attributes and their associated expressions can be configured using a single Conditional Attribute.

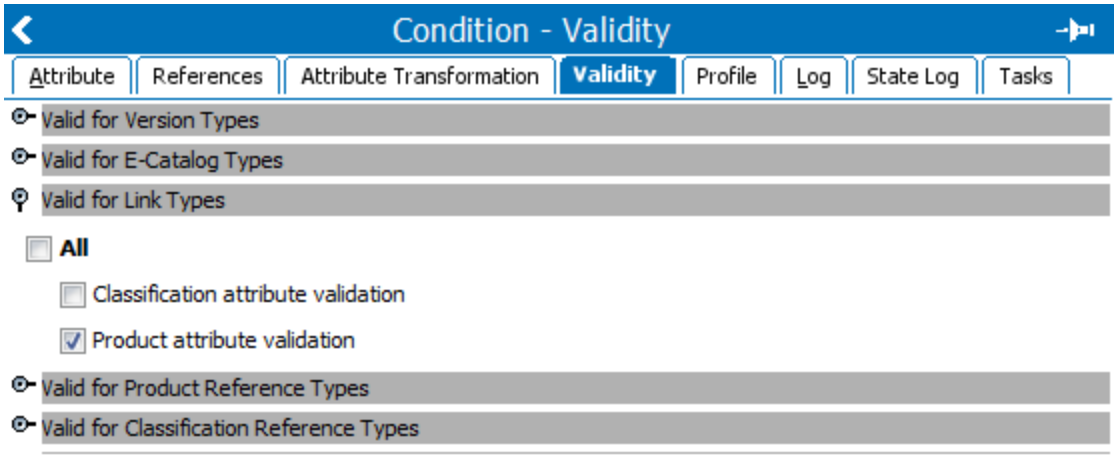
1. In a relevant **Attribute Group** in **System Setup**, create a **Description Attribute** with the **Base Validation Type** called **Condition**.



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

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

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

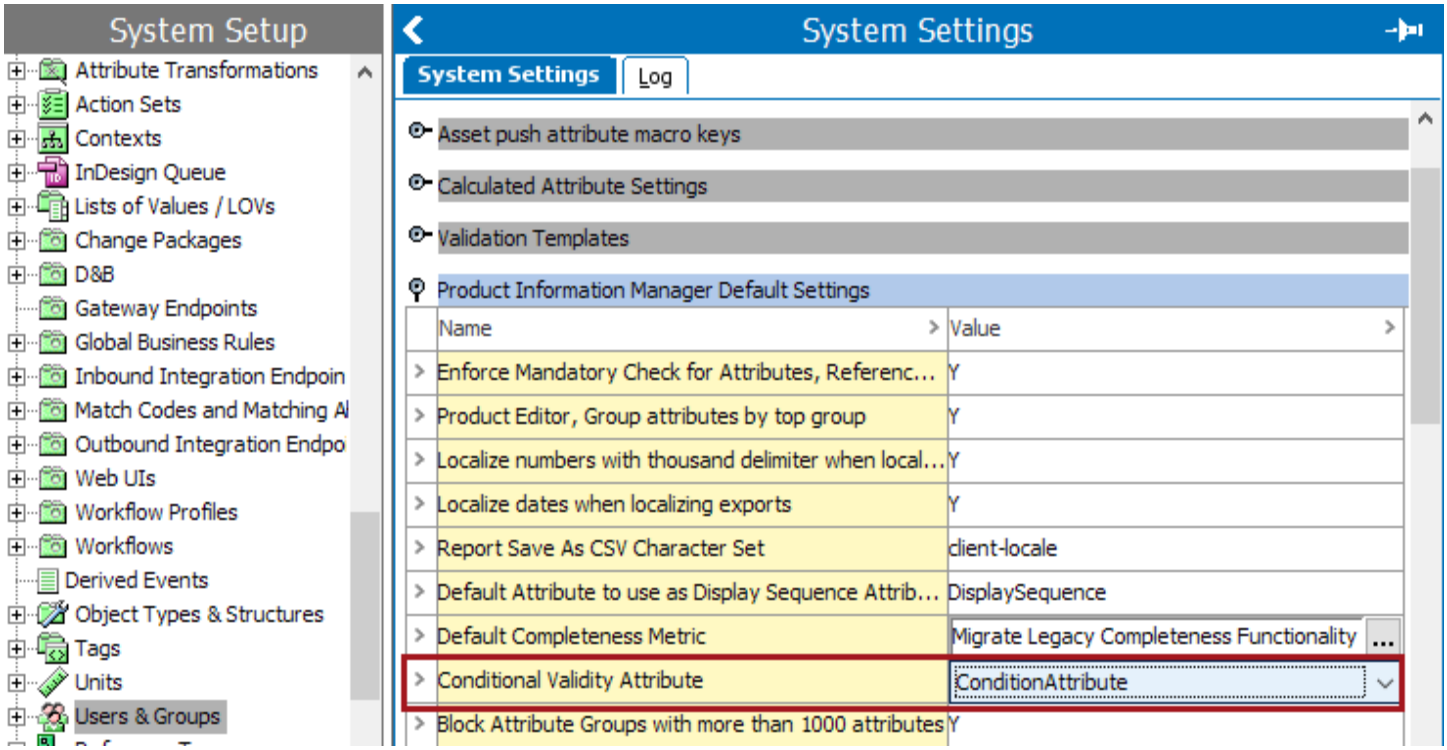


Activating a Conditional Attribute

Once created, the Conditional Attribute must be activated.

1. Navigate to **System Settings** and select **Users & Groups**.
2. Locate the **Product Information Manager Default Settings** section.

3. In the **Conditional Validity Attribute** field, select the Conditional Attribute configured in the previous section.



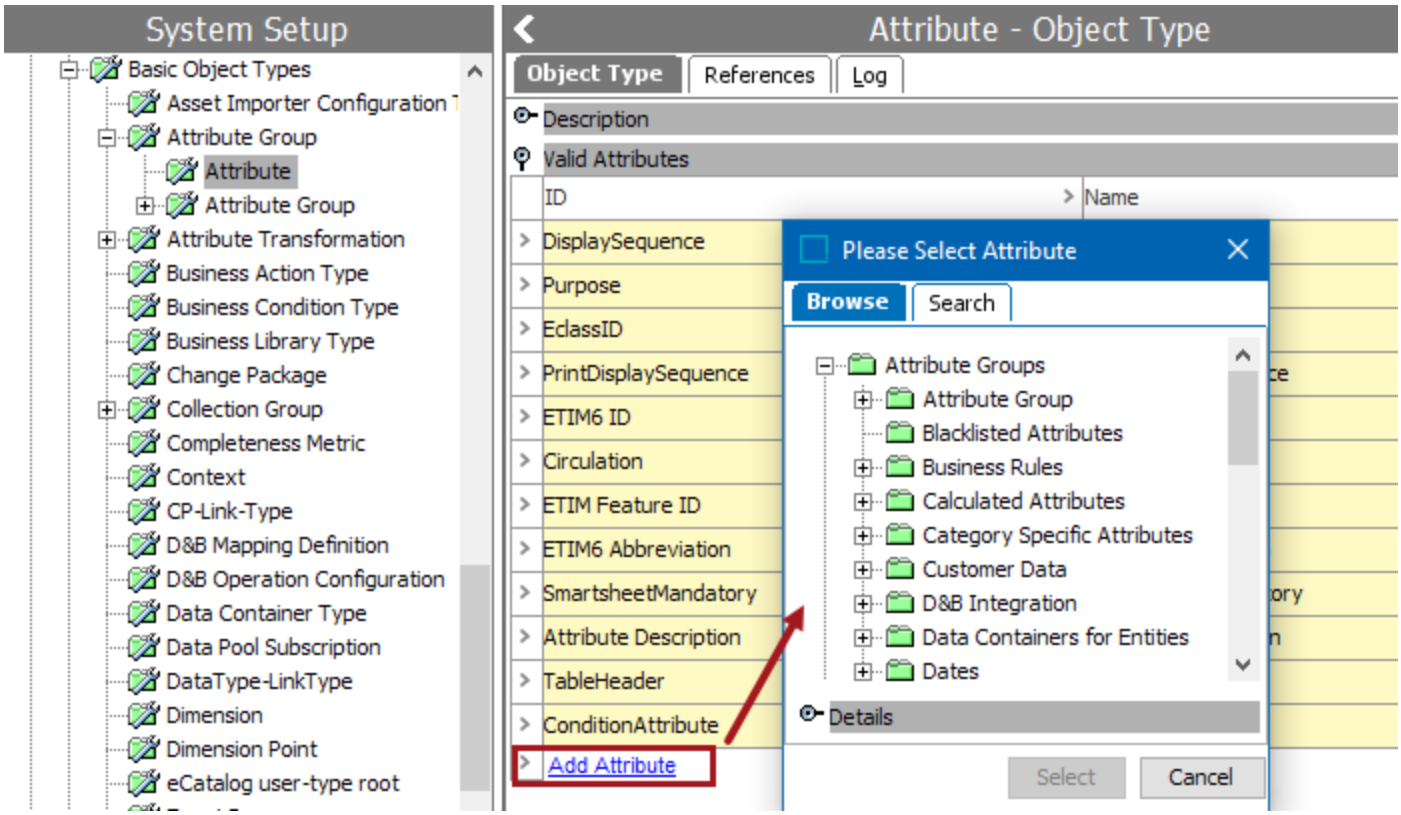
Configuring Conditionally Valid Attributes

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

Description Attributes

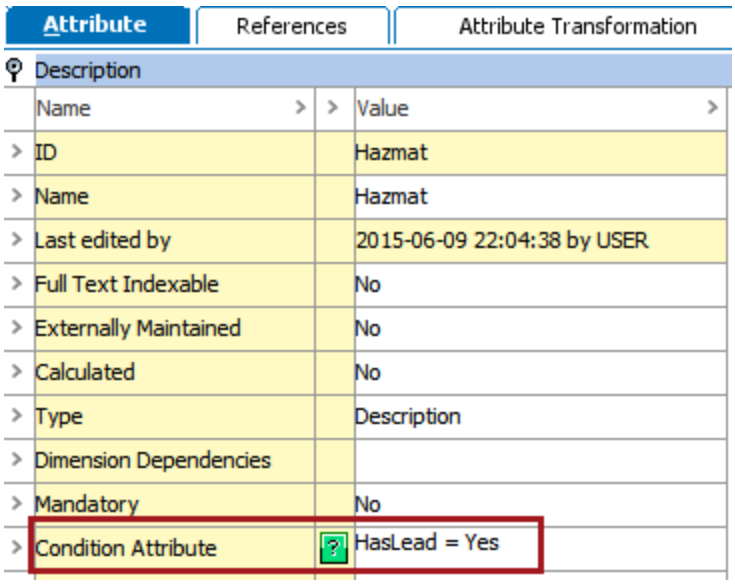
The condition for a Description Attribute is configured as metadata on the attribute itself. Before you can add it to an attribute, however, the Conditional Attribute must be made valid for the Attribute object type.

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

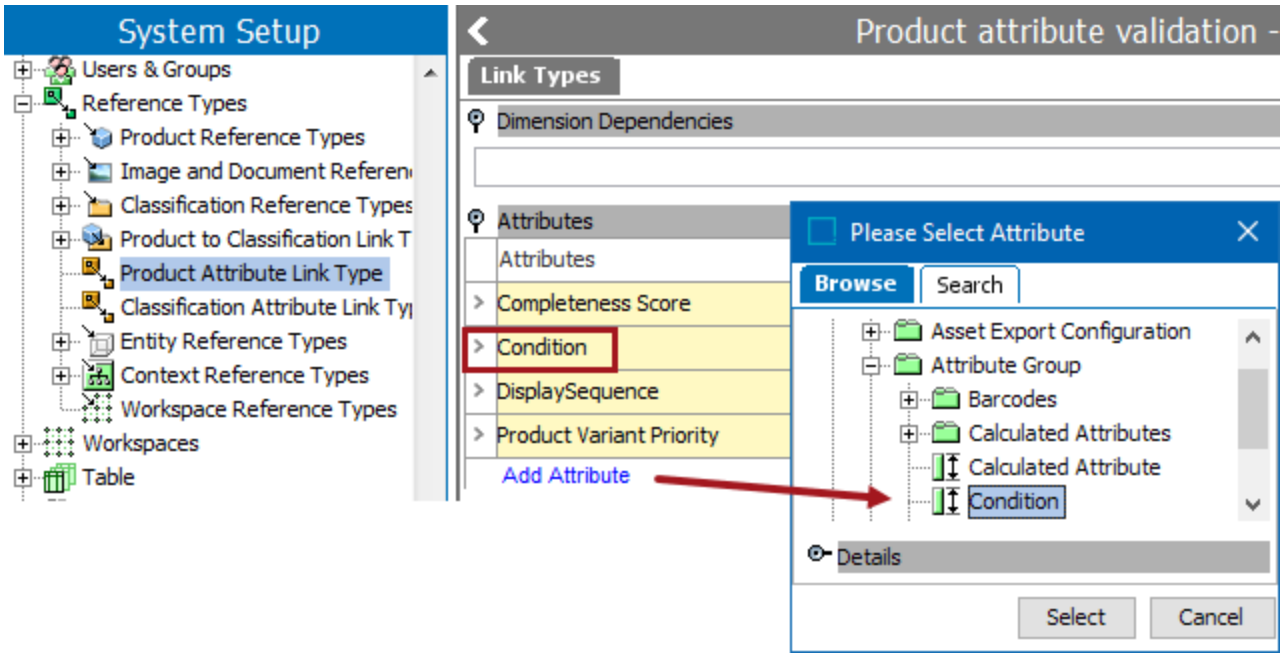
3. Navigate to **System Setup** and select the desired Description Attribute.
4. In the Conditional Attribute field, enter the desired condition rules for the Conditionally Valid Attribute to follow.



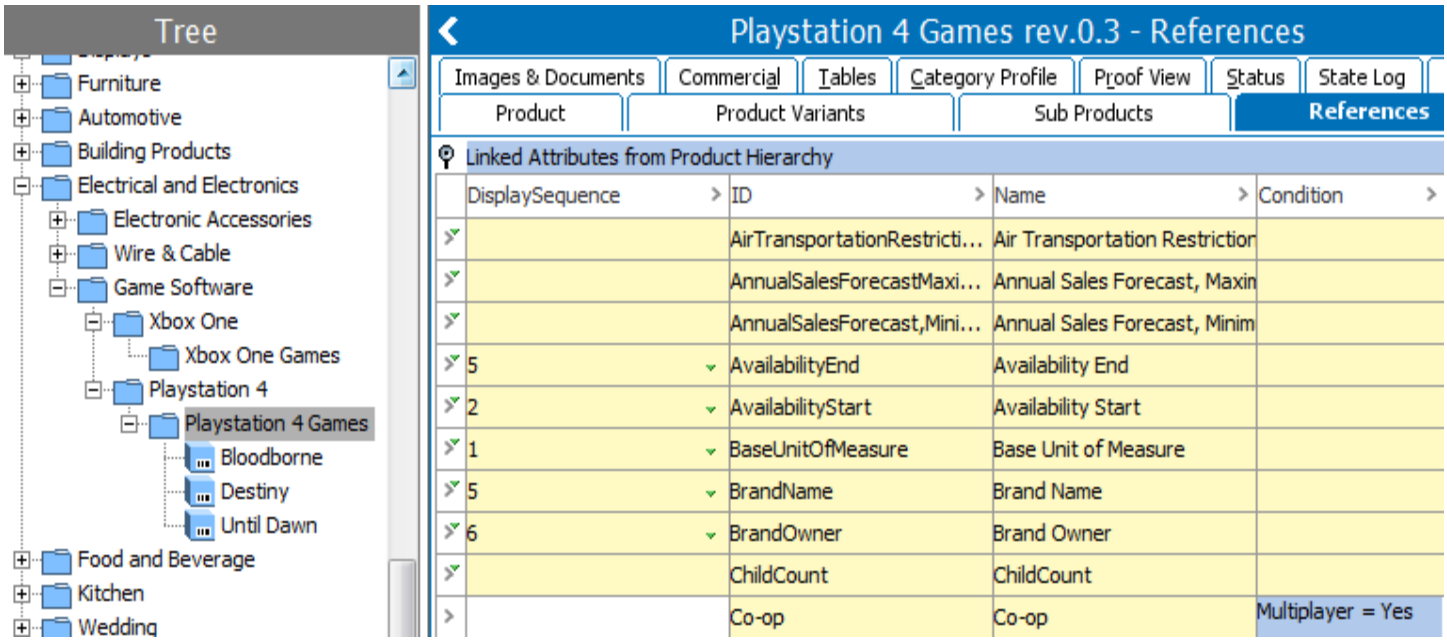
Specification Attributes

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

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



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').

Basic Information and references | Category Informations | Additional Information | Packaging Hier

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.

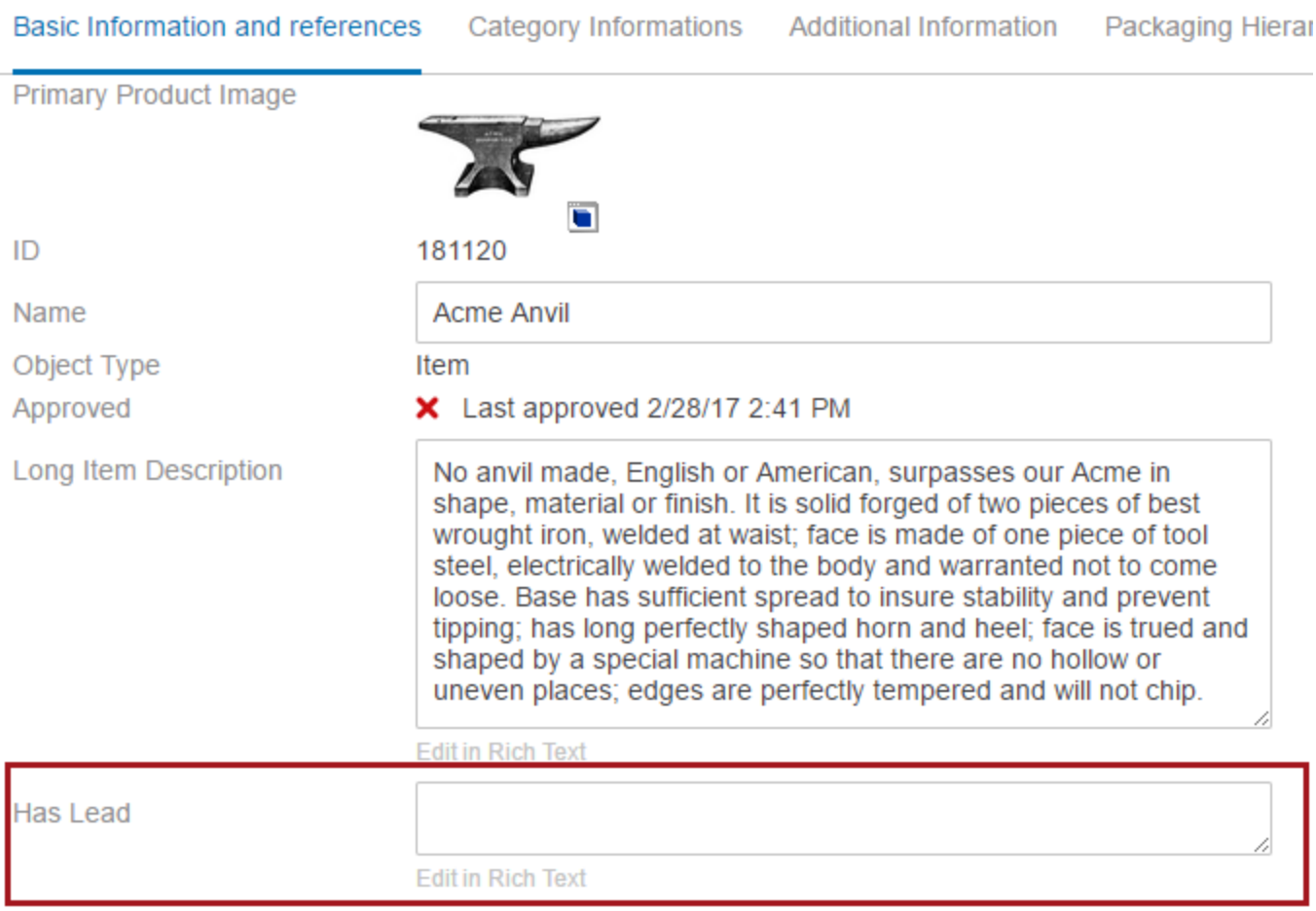
Edit in Rich Text

Has Lead	Yes

Edit in Rich Text

Hazmat	

Edit in Rich Text



Note: Hazmat attribute will not be available, If 'Has Lead' attribute is set to 'No' or 'Left Blank.'

Conditional attribute functionality can be used with Web UI elements:

- ParentValue
- TargetAttributeColumn
- TargetAttributeLinkColumn
- TargetValueHeader
- Value
- ValueHeader
- ValueGroup
- ValueLinkHeader
- ValueGroupHeader
- Attribute Value Component
- Attribute Value Group Component

Note: Before working on Conditional Attributes, it is important to make sure that attributes have validity and are linked on the object where conditions should match. In our example, Has Lead and Hazmat attributes are linked and have validity on the Item object. If the attribute is linked on the parent level, specify the condition on the parent level. Validity must be available on object.

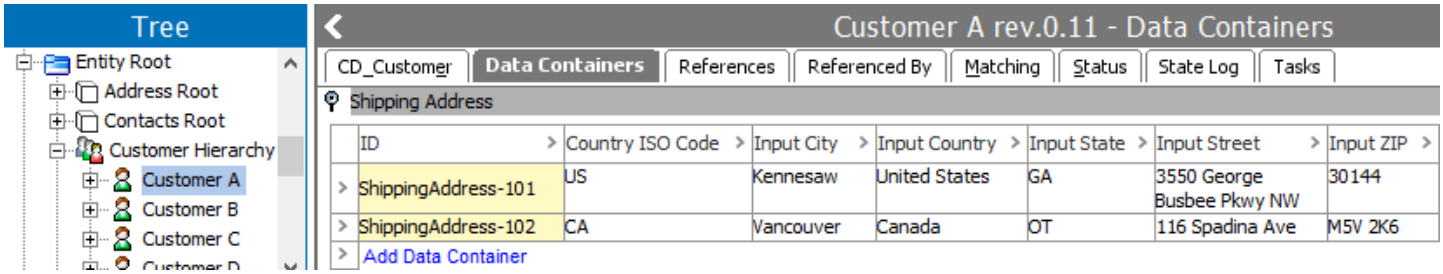
Data Containers

A data container allows you to represent and structure complex entity or product data through the use of composite attribute objects. This topic is an overview of complex data modeling as well as basic information regarding data container functionality available within workbench and Web UI.

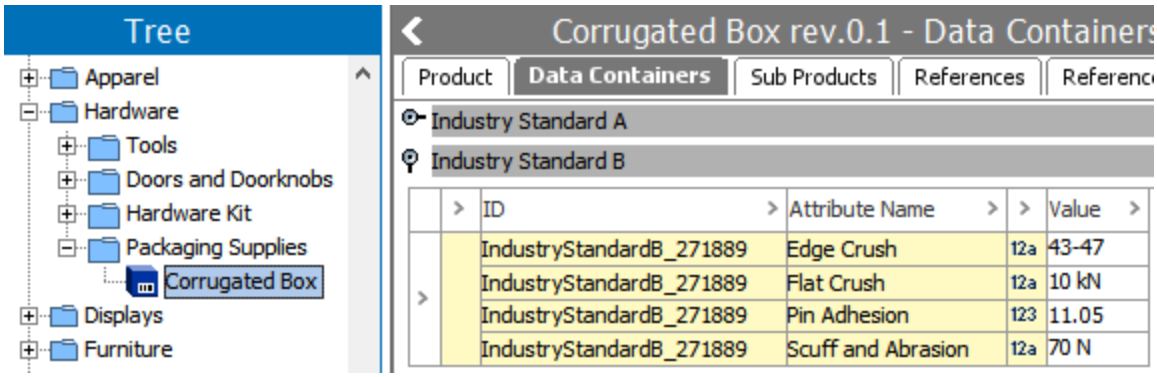
Composite Attribute Concept

A data container type allows modeling complex data with a single record that comprises many different pieces of information. Composite attribute objects have many benefits, including simplified revision handling, a collective process to save and approve data, and easier management of composite attribute data in workflows.

For example, a composite attribute object for a customer record could use entity data containers for addresses (e.g., main / physical, shipping), contact names, contact information, email addresses, phone numbers, etc., with relevant attributes and references. The customer record acts as a 'composite attribute,' which is a data structure that is composed of other data structures that need to be viewed and treated as a whole. While this type of complex data can be modeled using references, it provides challenges with various actions and processes. The following image shows an entity data container that allows multiple instances (the 'Allow multiple data containers' parameter is set to 'Yes'), so the individual data container attributes / references are displayed as columns.



Another example is a composite attribute object that captures data for multiple tests on a single product. Each data container holds values for a single industry standard test (e.g., tensile strength, specific gravity, water absorption). All tests must pass before the product can be approved. Data containers allow a value to indicate pass / fail for each test to be recorded, and workflows / business rules can be used to verify that the product is not approved until all tests pass. The following image shows a product data container that does not allow multiple instances (The 'Allow multiple data containers' parameter is set to No.), so the individual data container attributes are displayed as rows.



Using Data Containers

Once basic setup is complete, users can begin working with data containers and their associated attributes and references. Data container modeling is intended to ensure that all create, read, update, and delete operations can succeed or fail as a whole since the operations are being performed on an object holding the data container.

The following rules apply to 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 distinguished via the icon. One data container can reside in multiple attribute groups.
- Data container types can only contain description attributes and references, and these attributes / references can be used in multiple data container types.
- Data container types have four tabs when viewed in workbench: Data Container Type, References, Validity, and Log.
- Once the data container type validity is set in System Setup, selecting a valid object from the Tree displays a Data Containers tab in the editor. Data containers configured in workbench are then available for configuration in the Web UI designer, allowing users to add, edit, delete, or just view data container values in Web UI.

Support for Data Containers

- Data container setups are transferable through change packages.
- Attribute values in data containers are searchable, and the entities or products are returned as results. The workbench Search tab, the Web UI Advanced Search functionality, and the Query API include a Data Containers search criteria. 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 via the survivorship rules for data containers ('Data Container: Trusted Source' and 'Data Container: Most Recent').
- Data containers are supported in calculated functions.
- Event handling is supported for data containers.
- The API supports access to data containers in JavaScript business rules.
- Data Containers are supported in the Address Component Model, as well the Loqate, CASS, and Trillium integrations used for address standardization.

- Completeness statistics take into account the attributes in the data containers.
- Revisions made to data containers are created on the applicable entity or product object, including creating, deleting, and editing attributes / references.
- Data containers can be maintained in Web UI by using one of the Data Container components.

Considerations and Limitations

Review the following items while working with data containers:

- Data containers can be made valid for either entities or products, and can only include description attributes and references.
- Product and entity data containers can be imported and exported using STEPXML, Advanced STEPXML, CSV, and Excel.
- Only product data containers allow inheritance, restriction to specified hierarchy structures, and the ability to make the data container mandatory.
- An entity or product can only contain 1000 data container records.
- Data containers cannot be externally maintained.
- Data containers are not dimension dependent; however, the individual attributes in data containers do support dimension dependencies.
- Data containers are not supported for product overrides.
- Data containers cannot be translated via the standard translation tools.
- Data containers cannot be initiated into workflows; they follow the workflow of their owning objects.
- Data containers are not supported in REST / SOAP calls (web services).
- Importing a product data container completely replaces the data container records, instead of making updates.
- Data containers are not supported for print output.
- Some business actions cannot run on data inside a data container. In general, when mapping to an attribute, it must be read directly from the node it runs on, not from the data container on the node.
- Data container sort order cannot changed for viewing in Workbench. It is, however, possible to change the sorting order that is displayed in Web UI by changing the data container to manually sort the attributes in Workbench. For more information, see the **Additional Web UI Functionality** section of **Data Containers in Web UI** topic in this documentation.
- There is no support for searching for inherited data containers and attributes.

For information on using data containers in workbench, see the **Adding and Maintaining Data Container Instances** topic in this guide.

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

- **Setting Up Data Container Types in Workbench**
- **Data Containers in Web UI** in the **Web User Interfaces** documentation

Additional information can be found in the relevant documentation for a particular functionality (i.e., Data Exchange, Navigation and Searches / Advanced Search (Web UI), Data Profiling, and more). In online help, search for 'Data Container' to review the available topics.

Setting Up Data Container Types in Workbench

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

This topic includes the following information:

- Prerequisites
- Adding Data Container Types
- Configuring Data Container Types
- Deleting Data Container Types

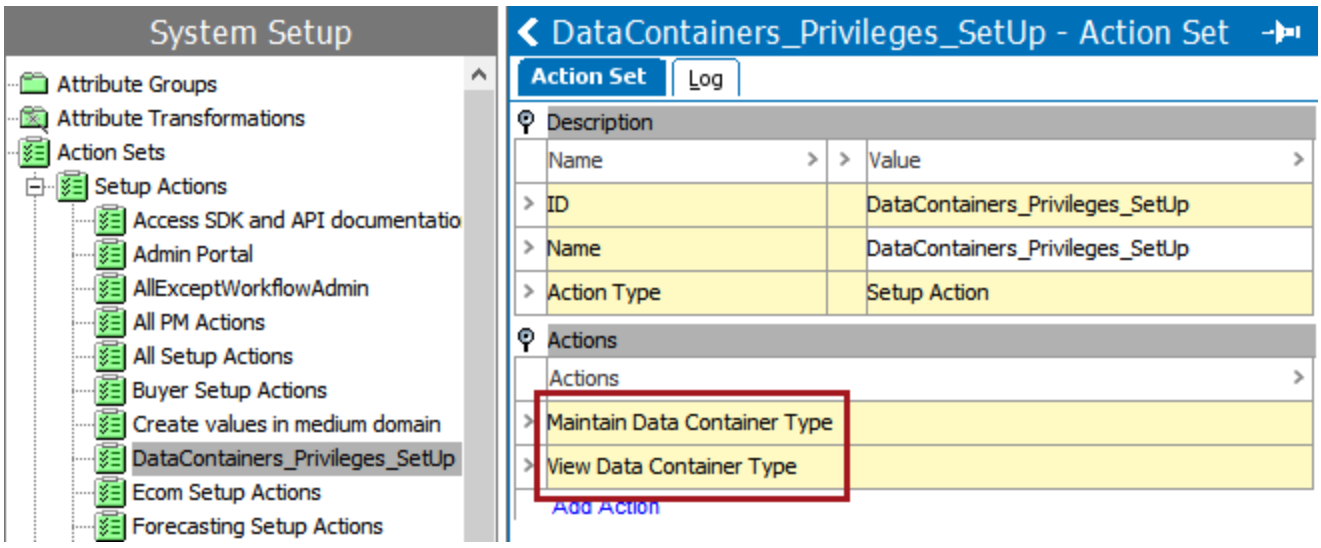
Prerequisites

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 create a new data container object type and work with related attributes, you must have all the proper privileges.

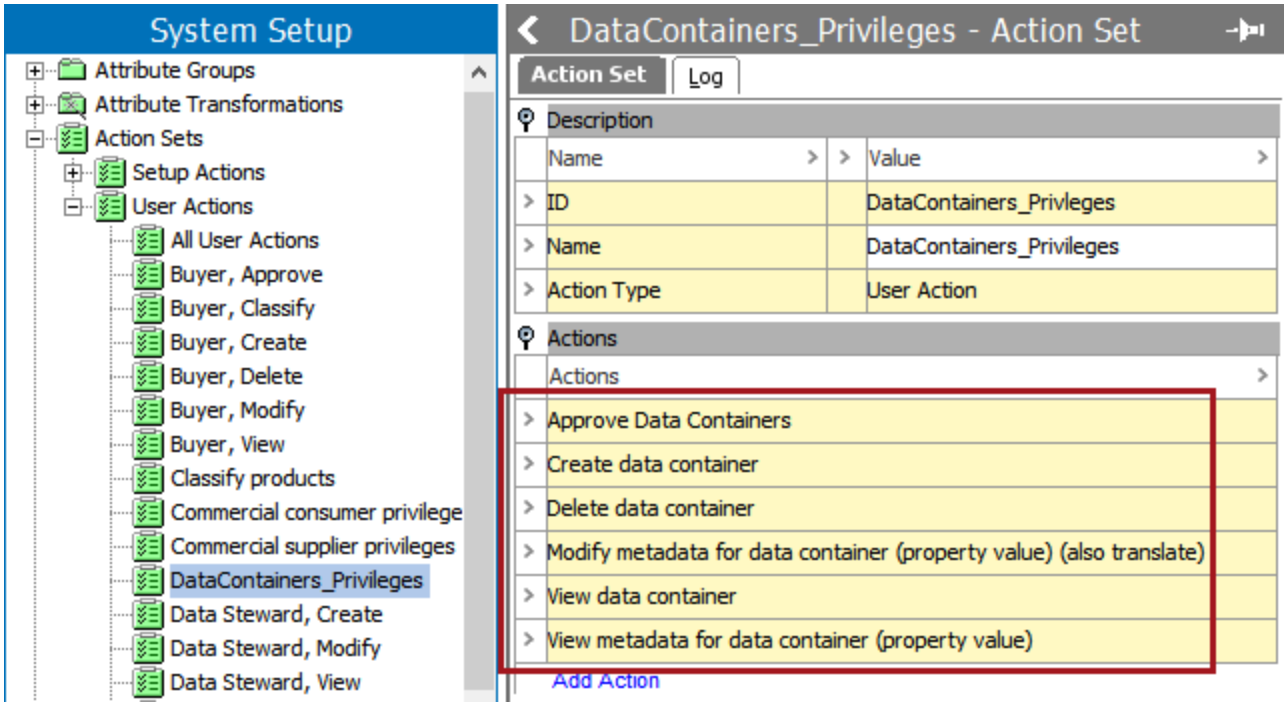
The following setup actions are related to data containers:

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



The following user actions are related 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

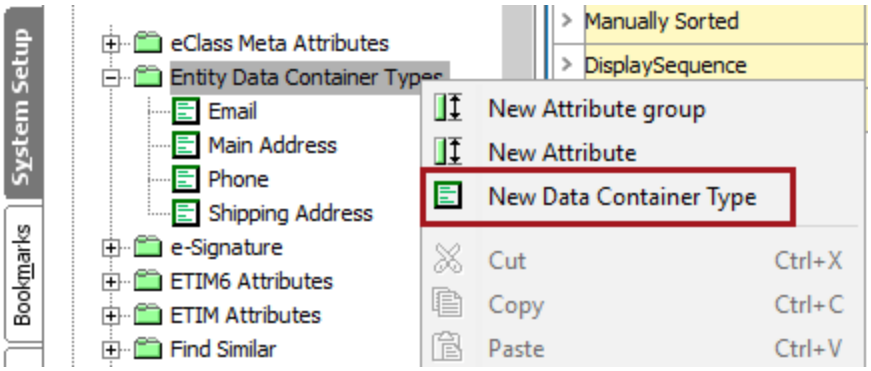


For more information, see the **Action Sets** section of the **System Setup / Super User** documentation.

Adding Data Container Types

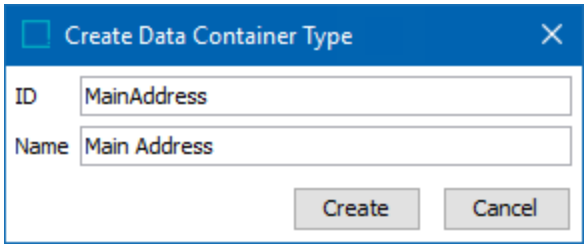
New data container types, distinguished via the icon, are created in workbench within System Setup > Attribute Groups. Insert data containers into new or existing attribute groups using the same functionality used to create new attribute groups and new attributes. For more information, see the **Attributes** and **Attribute Groups** topics.

1. In System Setup, right-click the attribute group that will hold the data container type, and select **New Data Container Type**, as shown below.



2. Enter the data container type ID and Name, and click **Create**.

> **Note:** Data Containers IDs are global and unique across objects.



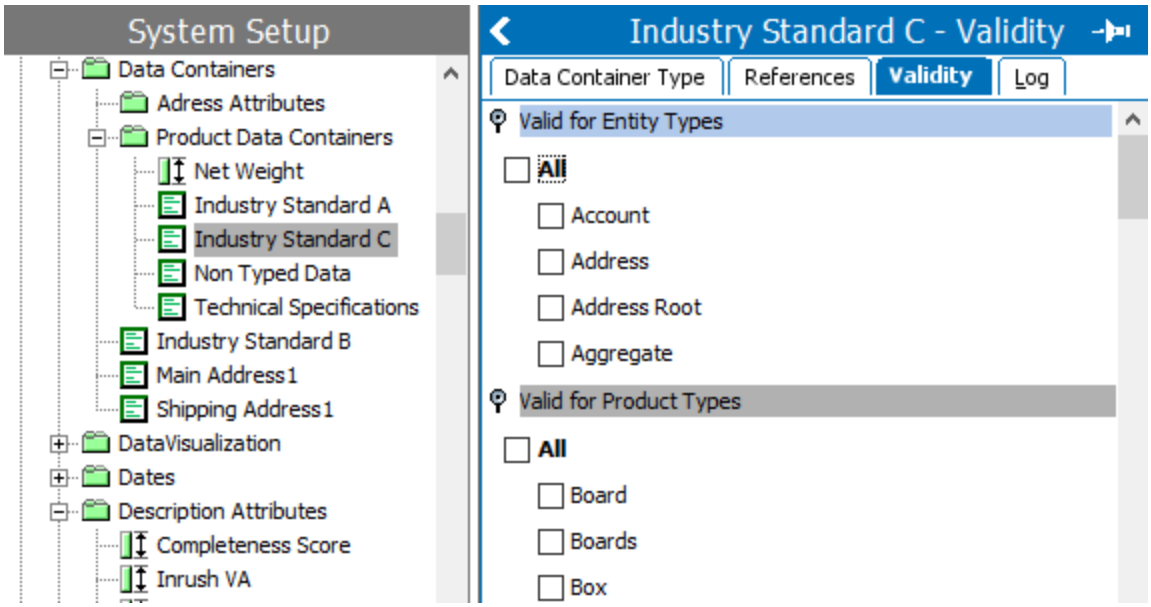
The new data container type editor displays four tabs: Data Container Type, References, Validity, and Log. Each one is described in the **Data Container Tabs** section later in this topic.

Configuring Data Container Types

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

1. In System Setup, select the Data Container Type and click the **Validity** tab to set valid object types. A data container can only be valid for entity or product object types, not both.

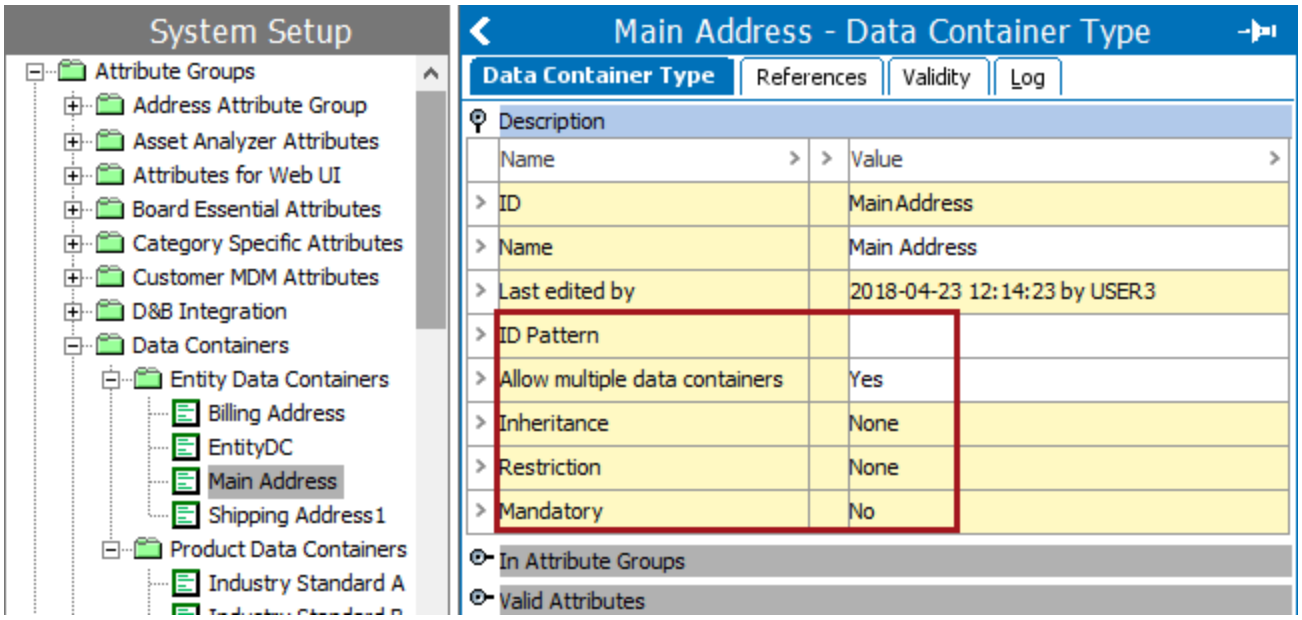
> **Note:** To change the object type validity of a data container, you must first clear all current settings before choosing the other object type. If existing values are found on a data container type when attempting to remove the validity of an attribute, the user is asked to confirm or reject the deletion of existing values.



- Open the **Valid for Entity Types** flipper and select the valid types of entities. Selecting an entity object type makes the Inheritance, Restriction, and Mandatory product-related parameters on the Data Container Type editor tab read only. If those parameters have already been modified, they must be restored to the default settings before the validity can be modified.
- Open the **Valid for Product Types** flipper and select the valid types of products. The Inheritance, Restriction, and Mandatory product-related parameters on the Data Container Type editor tab can now be edited.

Note: If a data container is made invalid for an object type, and it had data container instances of that type, those instances are orphaned by a background process. The values are still displayed in the Tree, but only the delete functionality is available while the object type is not valid.

2. Select the **Data Container Type** tab, and on the **Description** flipper, set the following parameters:

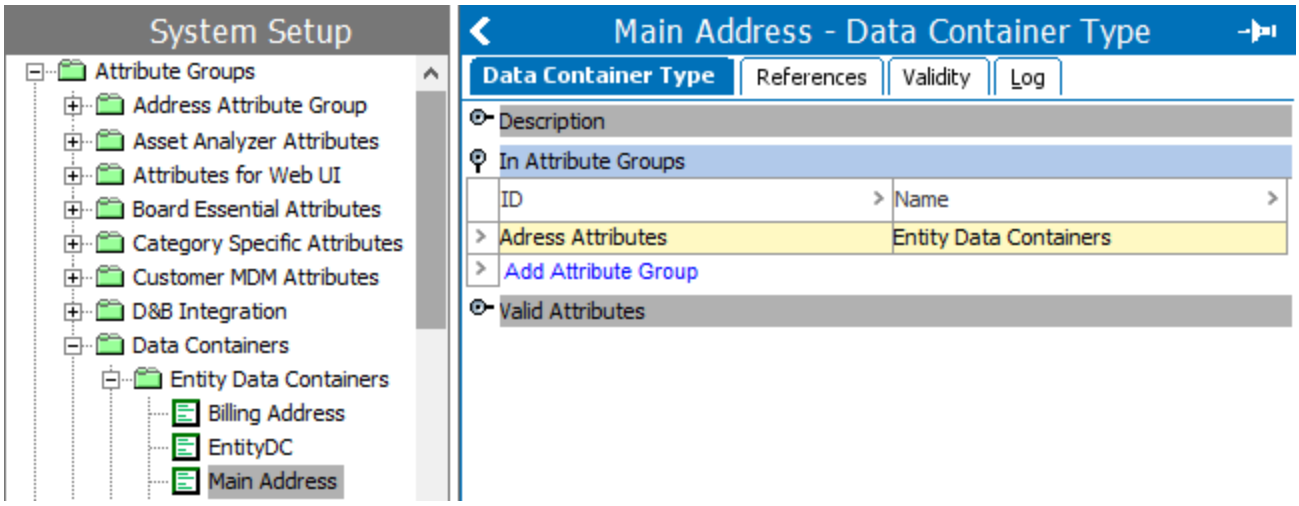


- For **ID Pattern**, which is required to add, edit, and delete the data container in Web UI, enter a pattern. If nothing is entered, users are able to determine their own data container ID when adding them to an object.
- For **Allow multiple data containers**, the default setting of 'No' means only one of this data container type can exist on the same object in the tree and the individual data container attributes are displayed as rows. Change this parameter to 'Yes' if two or more container instances should be allowed on the same object in the tree and to display individual data container attributes as columns

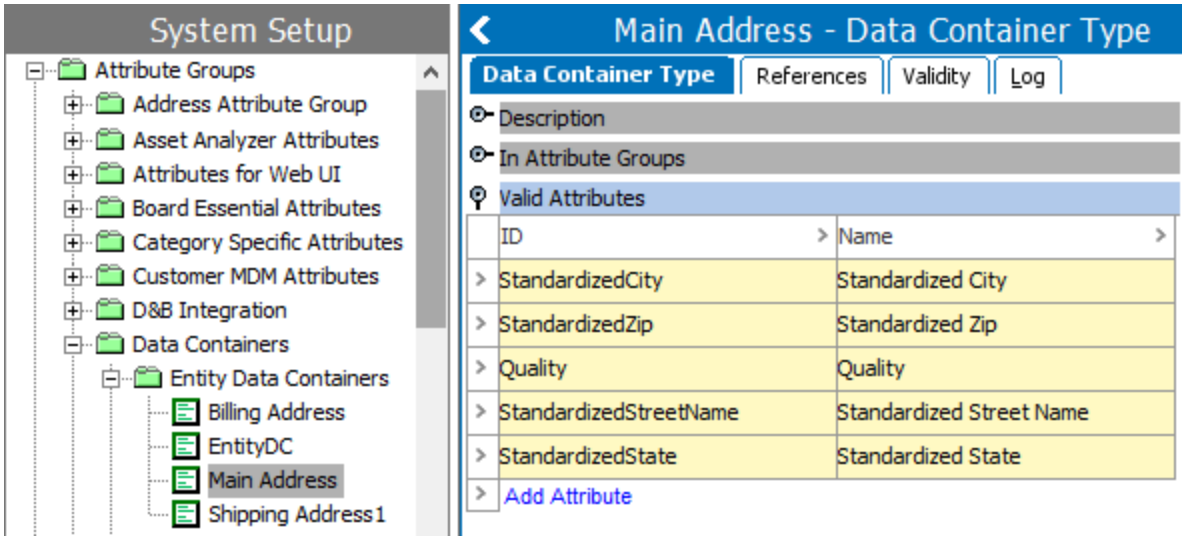
  When set to 'Yes', if multiple instances of the data container type exists on a single object, reverting to 'No' requires some data clean up, as defined in the **Changing 'Allow multiple data containers' back to 'No'** section below.

- For **Inheritance**, for product data containers only, the default setting of 'None' means values of the valid attributes are not inherited. Change this parameter to 'Inherited' if values should be inherited as they would be on a standard specification attribute. For entity data containers, this parameter is read only.
- For **Restriction**, for product data containers only, the default setting of 'None' means the data container can be used at all levels of the hierarchy. Change this parameter to 'Validity restricted to hierarchies' if the data container usage should be limited to specified nodes in the hierarchy, as determined by the References tab. For entity data containers, this parameter is read only.
- For **Mandatory**, for product data containers only, the default setting of 'No' means the data container attribute values can be left blank. Change this parameter to 'Yes' if the data container attribute values are required. For entity data containers, this parameter is read only.

3. On the **In Attribute Groups** flipper, click the **Add Attribute Group** link to display the data container type in one or more attribute groups.



4. On the **Valid Attributes** flipper, click the **Add Attribute** link to add attributes to the data container type. Individual attributes must be selected (no attribute groups); however, multiple attributes can be selected and added at the same time.

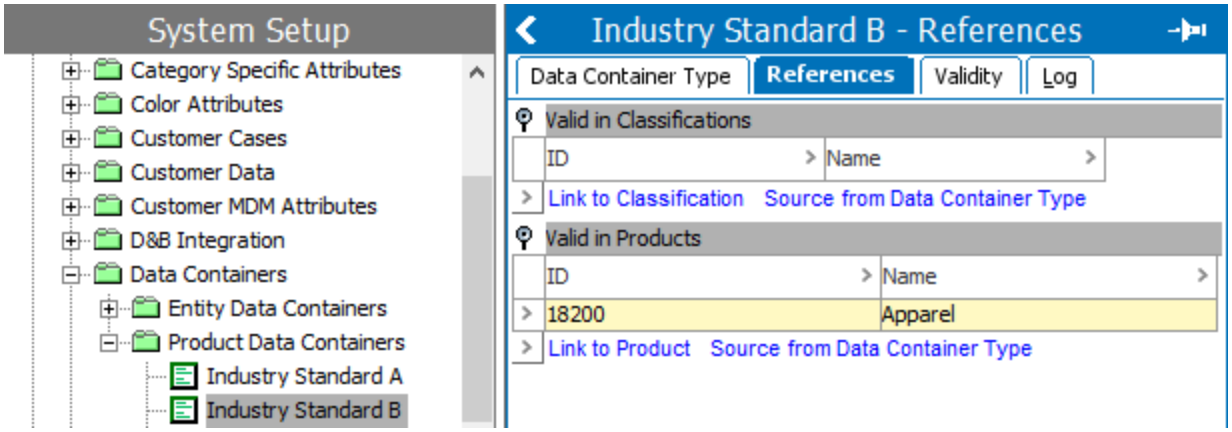


Note: Formatting of attributes is not supported. If formatting is used, the attributes are displayed as read only in the Web UI and, in the case of multi-valued data containers, locked for editing. Calculated attributes can be used. And, if using inline references in attribute values, the received value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

Alternatively, attributes can be added to a data container from the attribute's 'Validity' tab, under the 'Valid for Data Container Types' flipper.

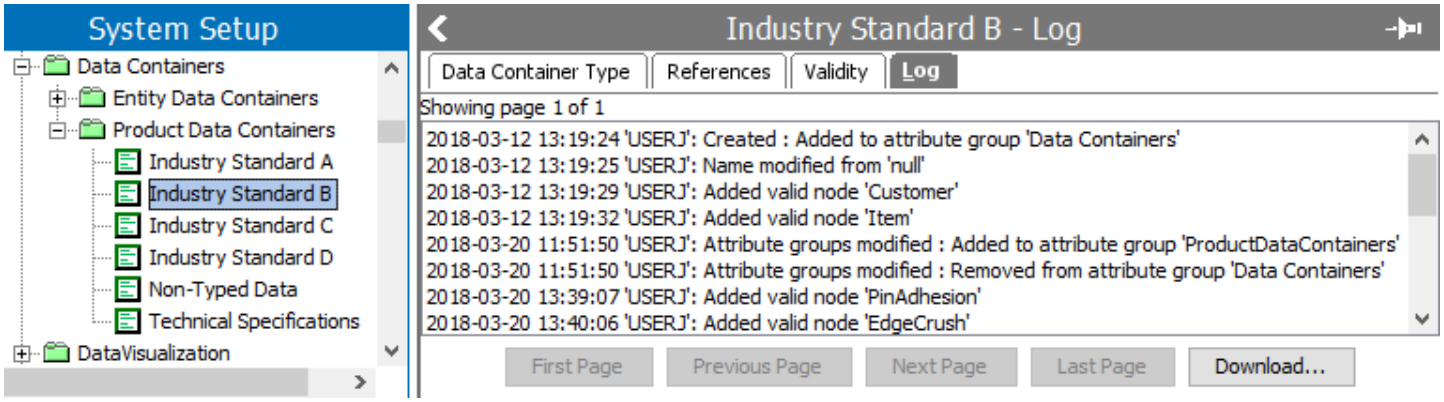
5. If product object types are valid, and the Restrictions parameter on the Data Container Type editor tab is set to 'Validity restricted to hierarchies', then click the **References** tab, and set the node in the hierarchy where

the data container can be used.



- Click the 'Link to Classifications' or 'Link to Product' links to define classification or product nodes for the data container.

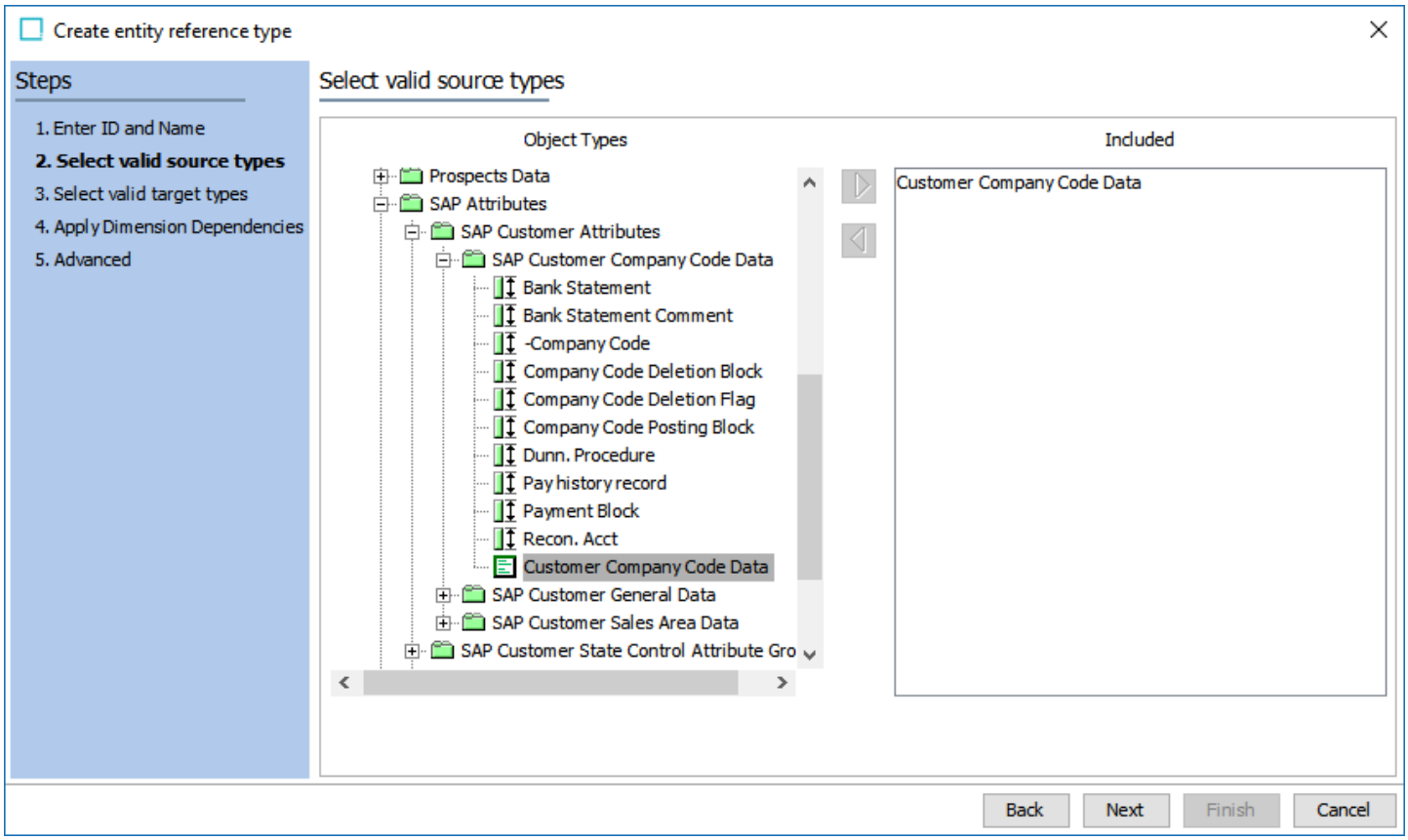
6. Select the **Log** tab to display the history of changes made to the data container. Users may download this record via the 'Download...' button.



See the **Adding and Maintaining Data Container Instances** topic to work with data containers in the Tree.

Adding references to a data container type

To add a reference to another object on a data container, a new reference type must be created in System Setup. Specifically, when configuring the new reference type, select the relevant data container as a valid source type.



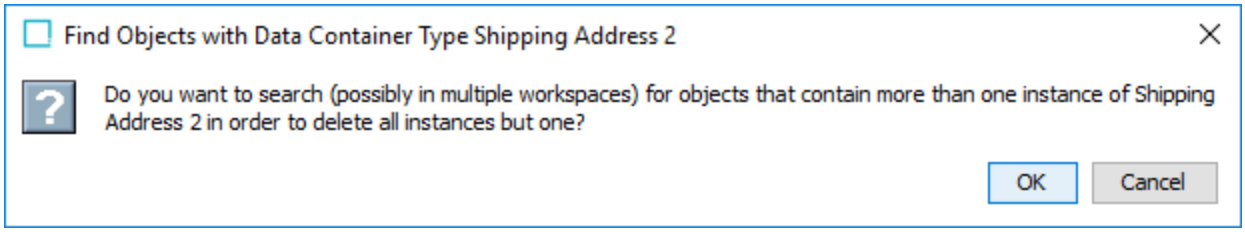
For more information on creating reference types, see the **Creating a Reference Type** section of the **System Setup** documentation. For more detailed information on configuring references on data containers, see the **References on Data Containers** section of the **System Setup** documentation.

Changing 'Allow multiple data containers' back to 'No'

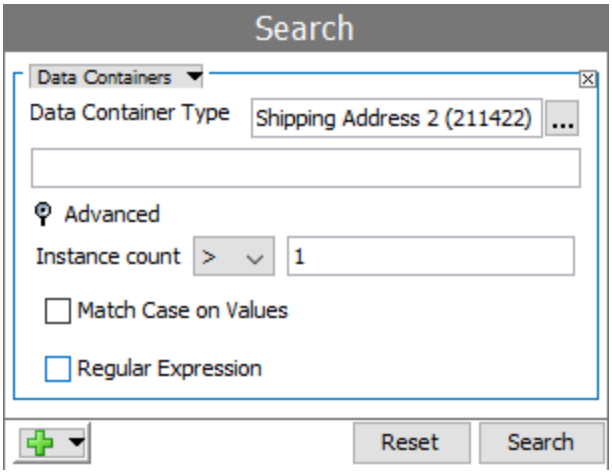
If the 'Allow multiple data containers' parameter is set to 'Yes,' and two or more container instances exist for any object, it is no longer possible to set it to 'No' until all the objects include a maximum of one instance.

To identify objects that have more than one instance of a data container type:

1. In System Setup, select the data container type, change the 'Allow multiple data containers' parameter from 'Yes' to 'No'.
2. If multiple instances exist on any single object, a dialog similar to the following is displayed for an optional auto search.



3. Choose an option:
   - Click **Cancel** to close the dialog and leave the data container set to allow multiple instances.
   - Click **OK** to display a Search tab with the relevant criteria is already supplied.

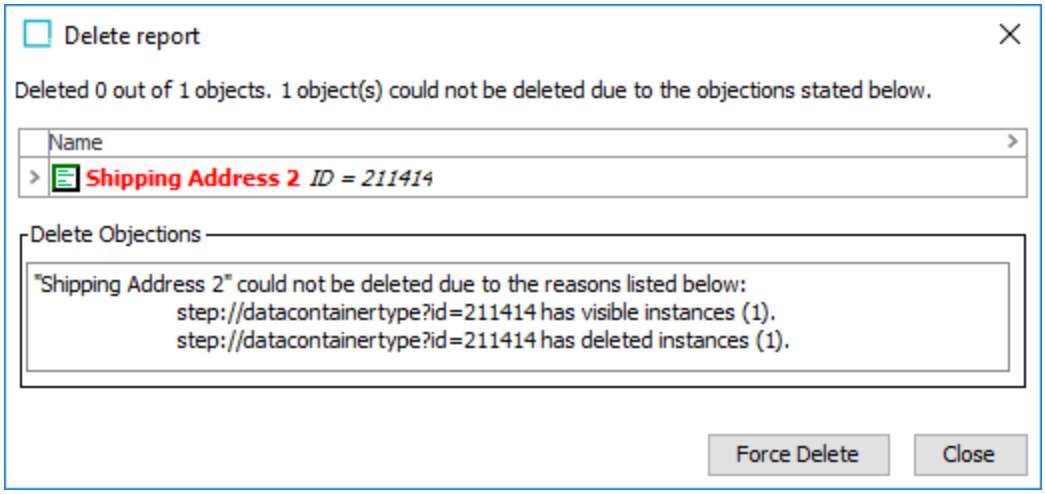


   For more information on executing searches, see the **Search** section of the **Getting Started / User Guide** documentation.

4. To search for multiple instances, click the **Search** button to display the results.
5. Review the objects with multiple instances of the same data container type and delete as necessary to leave a maximum of one for all reported objects.
6. Back in System Setup, select the data container type again, change the 'Allow multiple data containers' parameter from 'Yes' to 'No'. This indicates it is a single-valued object type.

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 Keys

Data containers can be identified and modified by using key definitions, which can be created using a combination of attributes and/or references. For more information, see the **Data Container Keys** topic in the **Data** Containers documentation.

Data Container Keys

Data container objects can be uniquely identified via a key. This allows users to create and update data containers via imports. The data container key serves as a unique identifier and can be based on the values of selected attributes and/or the target object IDs of selected reference types. These unique identifiers are necessary in order to identify data containers when that data is imported (e.g., from an external system that cannot identify an internal STEP ID) and to ensure that duplicate data container objects are not created.

In addition to creating data container keys within the workbench, there are pertinent aspects of data container keys that are addressed in other topics:

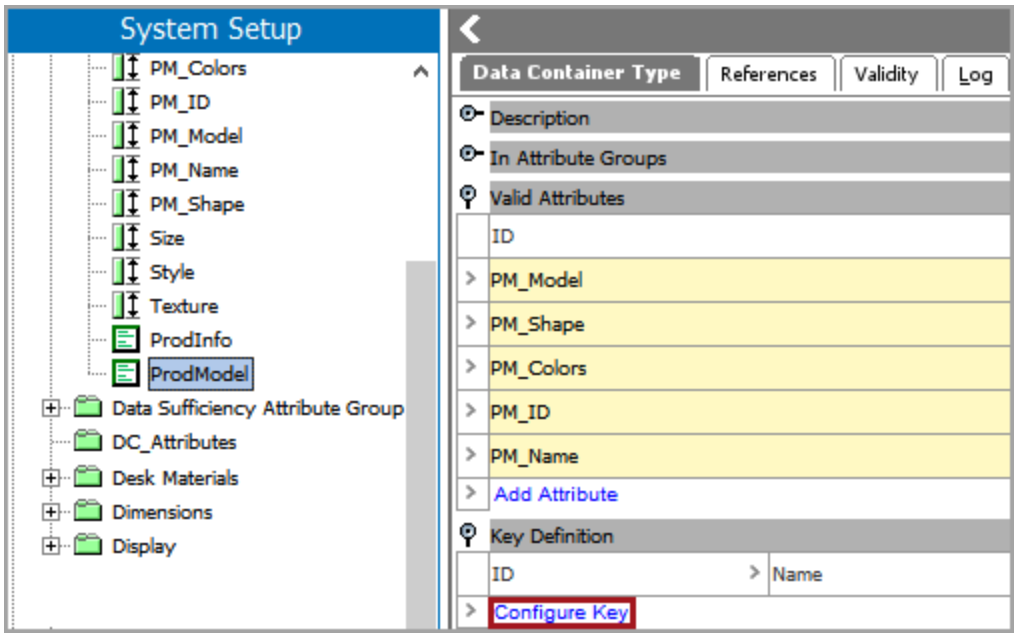
- **Data Container Keys in STEPXML** - Users can modify data container keys using the <KeyDefinition> tag. For more information regarding data container keys in STEPXML, see the **Data Containers in STEPXML** topic in the **STEPXML Format** documentation.
- **Mapping Data Containers** - Users can specify the mapping of attributes and reference types for data container keys. For more information, see the **Data Container - Map Inbound** topic in the **Data Exchange** documentation.
- **Data Containers in Web UI** - Users working with data containers in Web UI are able to add values to attributes and/or reference types that make up data container keys. For more information, see the **Data Containers in Web UI** topic in the **Using Web UI** documentation.

Creating a Data Container Key in the Workbench

Users can create data container keys within the workbench using the Configure Key feature, located within the Key Definition flipper parameter.

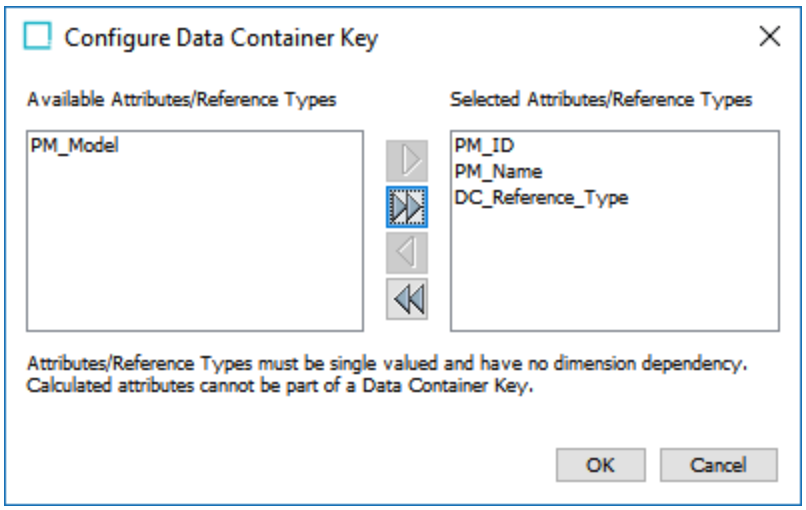
Note: Within the Data Container Type editor, the Inheritance parameter (located under the Description flipper) must be set to 'None' in order to create data container keys. If the parameter is set to 'Inherited,' the key definition cannot be configured. Additionally, if a user attempts to set the Inheritance parameter to 'Inherited' after a key definition has been configured, they will receive an error.

1. Select the data container to which you want to apply a unique key, then open the Key Definition flipper within the Data Container Type editor.
2. Click **Configure Key**.



3. In the Configure Data Container Key dialog, select the attributes and reference types that the key will consist of from the Available Attributes/Reference Types column and click the right single arrow to place those options into the Selected Attributes/Reference Types column. Only attributes and reference types that fit the specific criteria to be part of a key definition will be available for selection. The specific criteria is detailed later in this topic.

   In this example, the user has selected the attributes PM_ID and PM_Name, and the reference type DC_Reference_Type. These attributes and the reference type will be used to create the data container key.



Note: By clicking the right double arrow, all options in the Available Attributes/Reference Types column will be moved to the Selected Attributes/Reference Types column. By clicking the left double arrow, all

> options in the Selected Attributes/Reference Types column will be moved back to the Available Attributes/Reference Types column.

Attributes and reference types must meet specific criteria to be used as data container keys.

Attributes **cannot** be:

- inherited
- dimension-dependent
- calculated
- multi-valued

> **Note:** If an attribute that has List of Values (LOV) as a Validation Base Type is used as a data container key definition, the LOV must have 'Yes' selected as the 'Use Ids on values' setting if the LOV is language dependent.

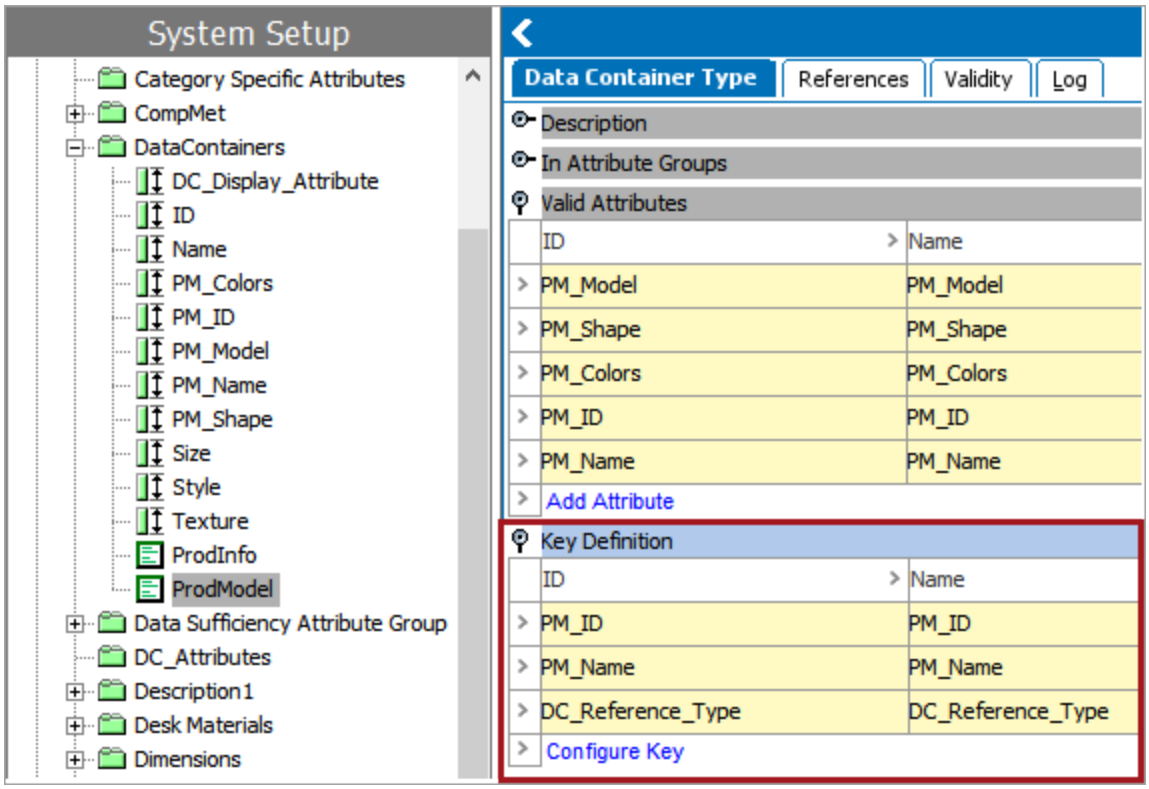
Reference types **cannot** be:

- inherited
- dimension-dependent
- multi-valued

Additionally, attributes and reference types must be valid for the selected data container.

4. Once the desired attributes and/or reference types are selected, click OK to close the dialog.

   The selected data container keys are now visible in the Key Definition parameter within the workbench.



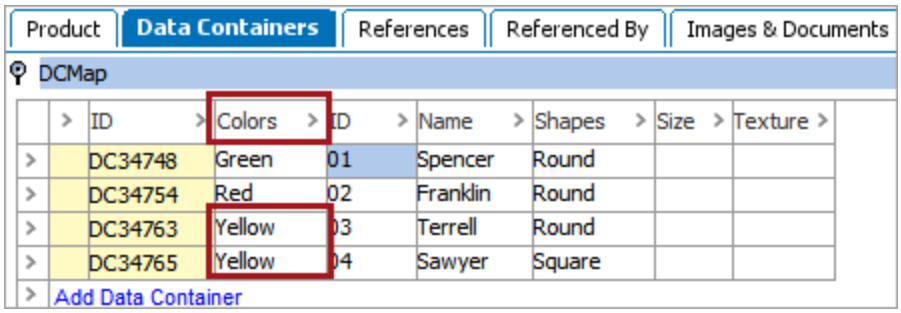
Note: Only users with the 'Edit Data Containers' privilege can make edits to the attributes /reference types after they have been assigned to data containers. It is recommended that only select admin users have this privilege added to their user profile and that this privilege be used sparingly to avoid creating duplicate keys. For more information regarding privileges, see the **User Actions** topic in the **Action Sets** documentation.

Note: Users can edit the target of a reference type that has been used as part of the data container key by right-clicking within the data container cell that holds the reference type and selecting 'Edit.' This will open up the node picker where another target for the reference type can be selected.

Check Duplicate Keys

Users are able to check duplicate keys on the same data container type using the Check Duplicate Keys feature. Duplicate keys, which can occur when a user selects an attribute or reference type to be used as a key definition that already has duplicate values, undermine the overall value of applying uniquely identifiable keys for data containers.

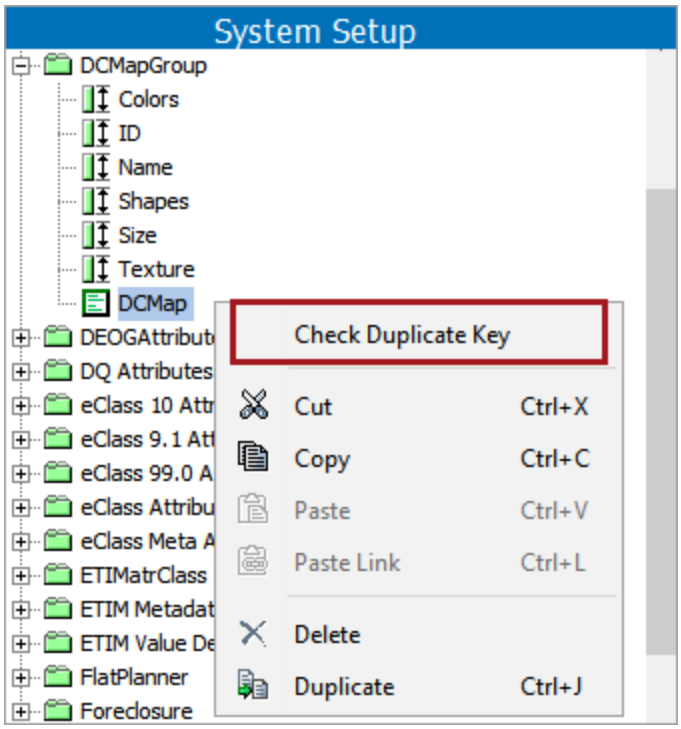
For example, in the image below, there are two data containers within the DCMap data container type that include the value 'Yellow' for the attribute 'Colors.'



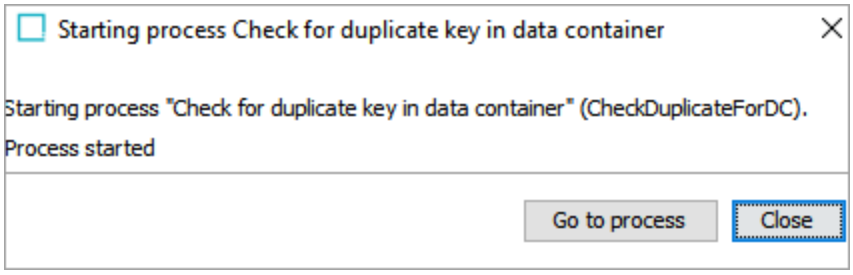
If the user creates a data container key using the attribute 'Colors' as the key definition to be part of the data container key after these data containers have already been created, this will result in duplicate keys.

To check for duplicate keys on a data container type:

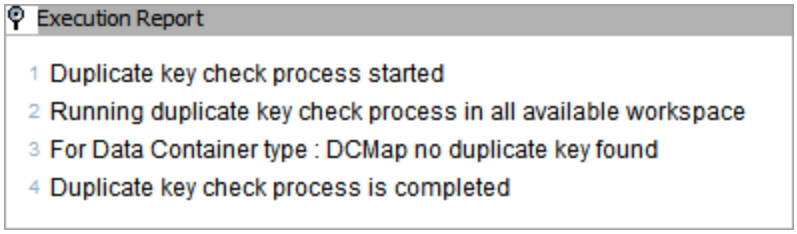
1. Right-click the data container type and select Check Duplicate Key.



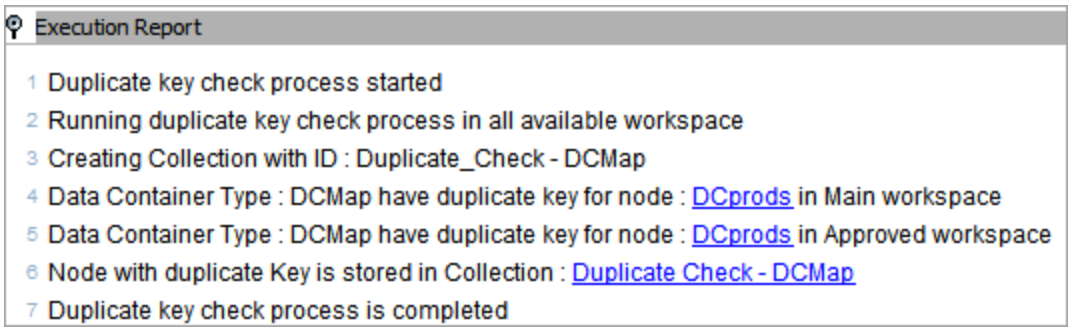
2. Click 'Go to process' to see if any duplicate keys were identified (and if so, which data containers they are in), or click 'Close' to shut dialog.



If there are no duplicate keys, the execution report message will state that no duplicate keys are found.



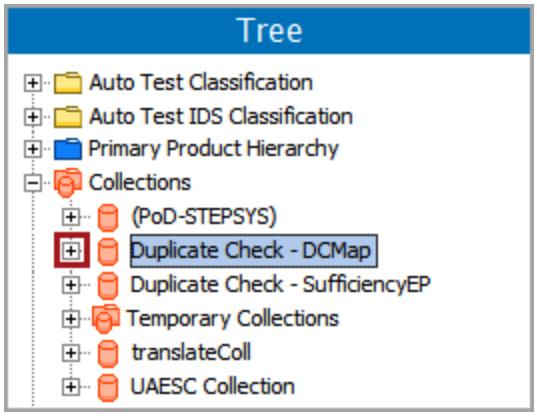
If there are duplicate keys, the execution report message will include the name and link of the collection that is created to store the objects that contain the data containers with duplicate keys. In this example, the name of the relevant collection is Duplicate Check - DCMap.



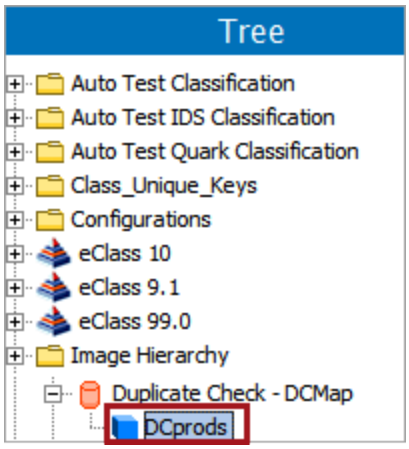
Removing Duplicate Keys in Collections

Once objects that contain data containers with duplicate keys are placed into a collection, the data containers that are attached to those objects can be modified to remove the duplicate key values.

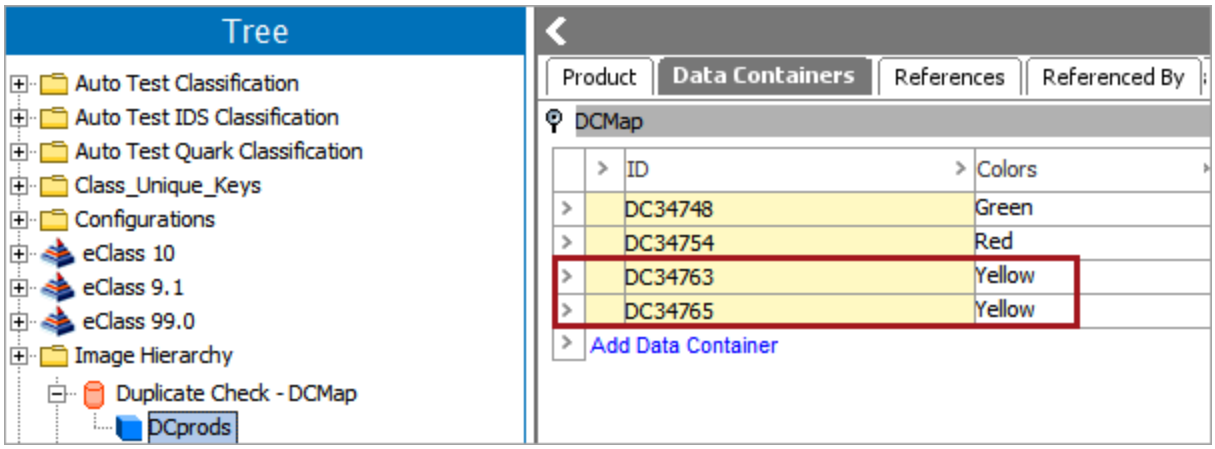
1. In Tree, expand the Collections node, locate the collection which houses the objects holding the data containers with duplicate key values, and click on the plus sign next to the selected collection.



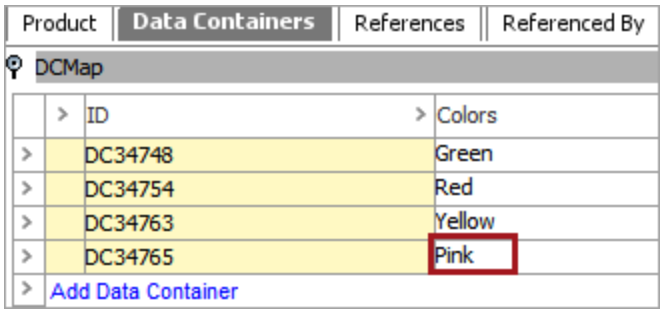
2. Select the product that contains the data container with duplicate keys.



3. With the product selected, click the Data Containers tab. In this example you can see two duplicate values for the attribute 'Colors.'



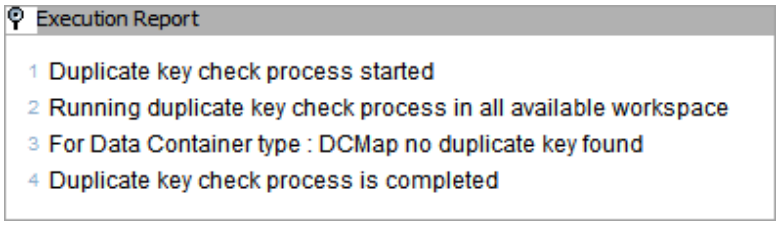
4. Change one of the 'Colors' attribute values to a different value.



Note: Only users with the 'Edit Data Containers' privilege can make edits to data container keys after they have been assigned to data containers.

5. Run the Check Duplicate Keys on the selected data container type (DCMap) again.

   The execution report returns stating that there are no duplicate keys found.



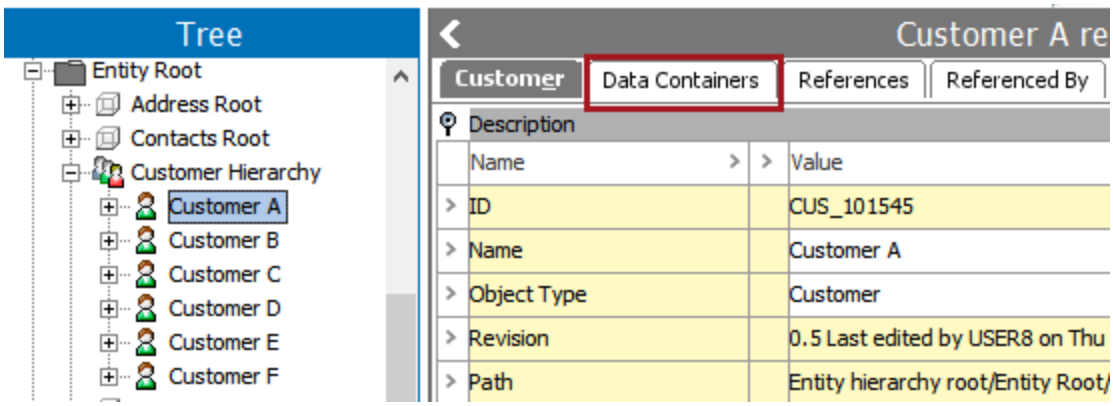
For more information regarding collections, see the **Collections** topic of the **Object Maintenance in Tree** section of the **Getting Started / User Guide** documentation.

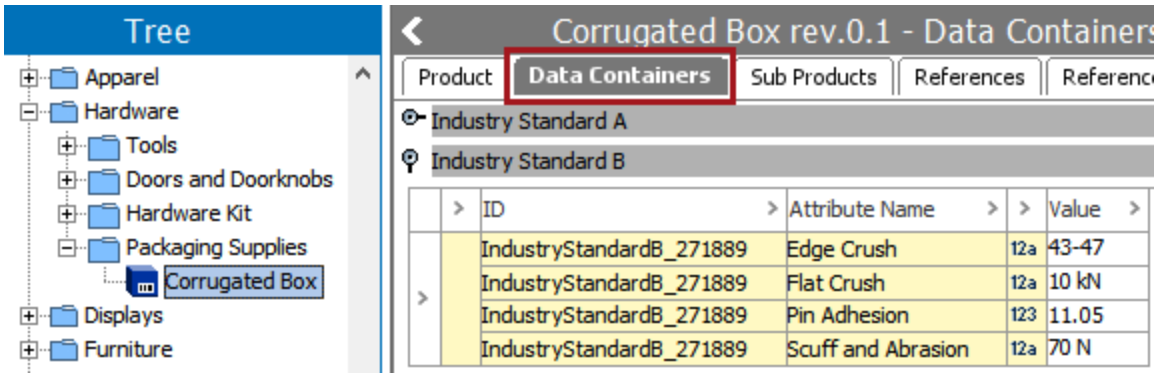
Adding and Maintaining Data Container Instances

After data container setup, a data container instance can be created on entities or products in the Tree. The data container type Validity tab settings determines the object types that display the data container. For more information, see the **Setting Up Data Container Types in Workbench** topic.

When a data container is valid for entity object types (a customer in the image below) or for product object types, selecting one of the valid objects in the Tree displays the Data Containers tab shown below. This is where you can add and maintain the attribute values and references for each container.

Note: If a data container instance exists in the Tree editor and has values, removing validity for the object type in System Setup does not remove the data container values in the Tree editor, even though the object type is no longer valid. Instead, the delete button is enabled, which allows you to remove the values, and the 'Add Data Container' link is disabled since the validity has been modified.

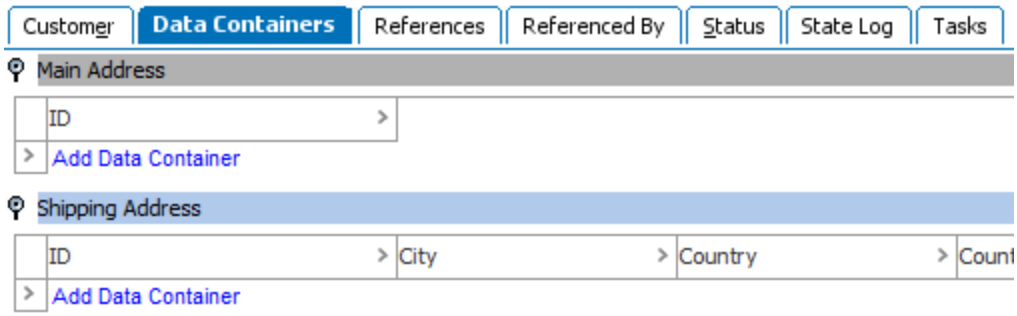


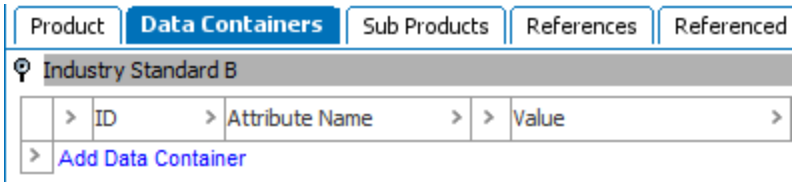


For information about working with and maintaining data containers in Web UI, see the **Data Containers in Web UI** topic in the **Web User Interfaces** documentation.

Add a data container instance

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

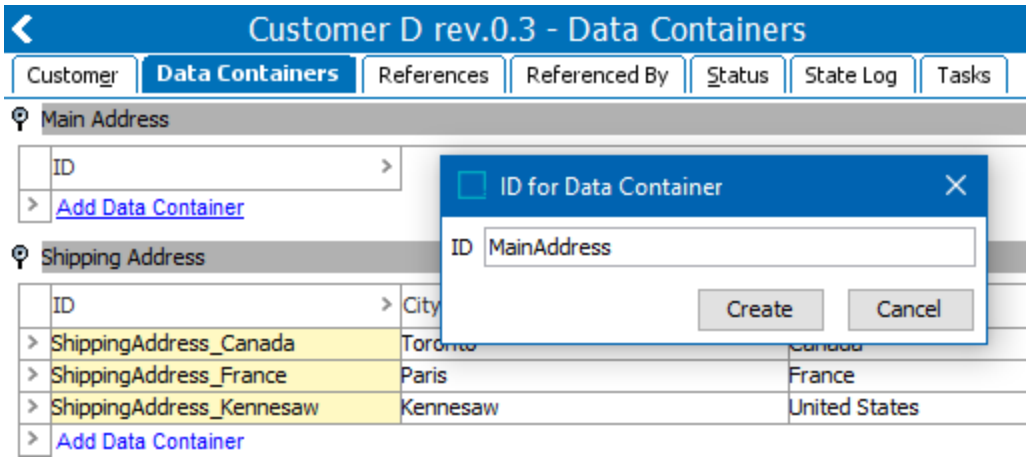


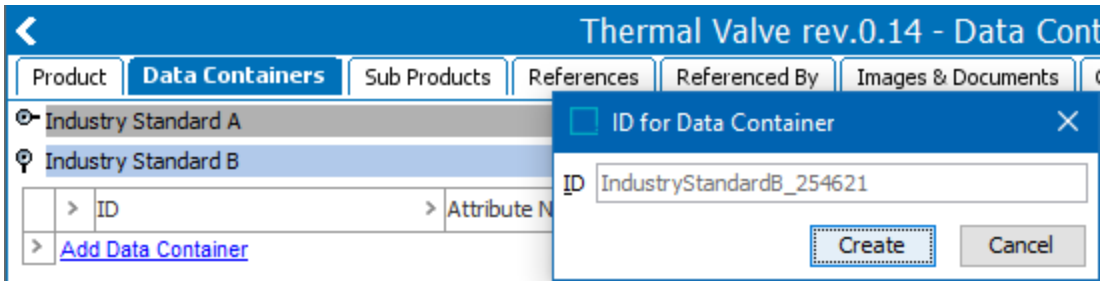


2. Click the **Add Data Container** link below the desired data container type, and supply an **ID** for a new data container instance, and click the **Create** button. If there is an auto-ID pattern set up, then no ID needs to be entered.

   Alternatively, click the **>** button at the beginning of a row to access the Add Data Container option.

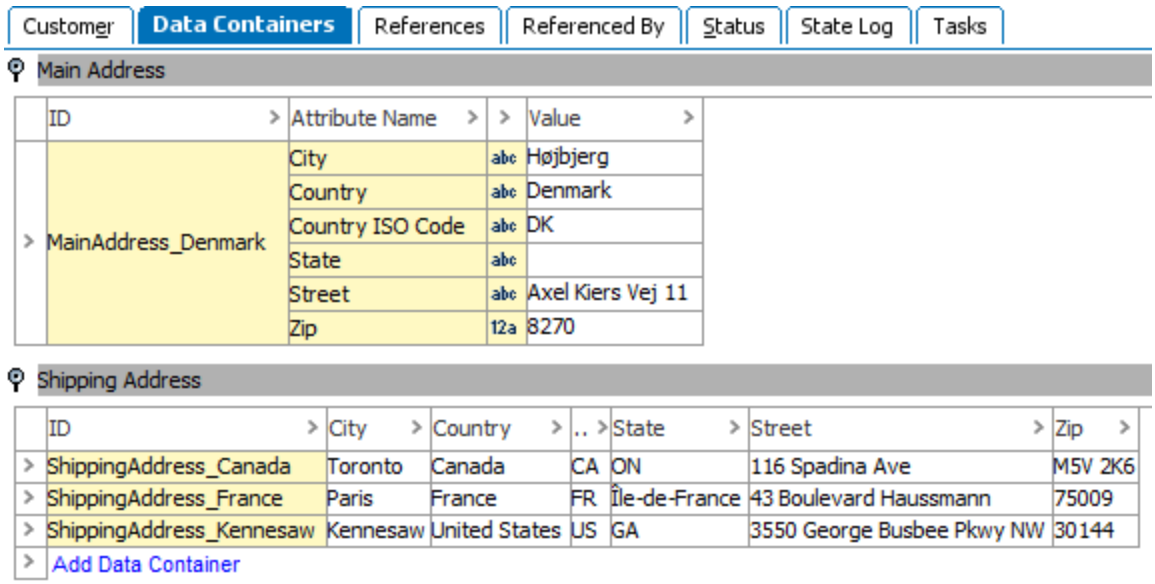
   In the example below, the 'Main Address' data container instance is created to hold values for each of the address attributes.

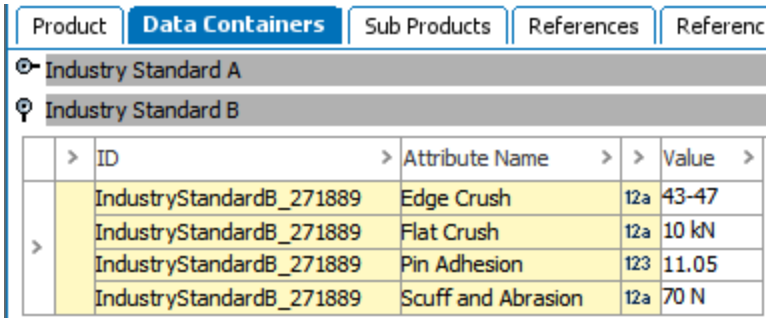




- If a data container type allows for multiple data containers as shown for 'Shipping Address' above, the Add Data Container link continues to display after the first data container is added.
- If multiple data containers are not allowed, as is the case with 'Main Address' and 'Industry Standard B' in the screenshots below, the link is removed after the first one is added.

3. Enter attribute values for each data container attribute.

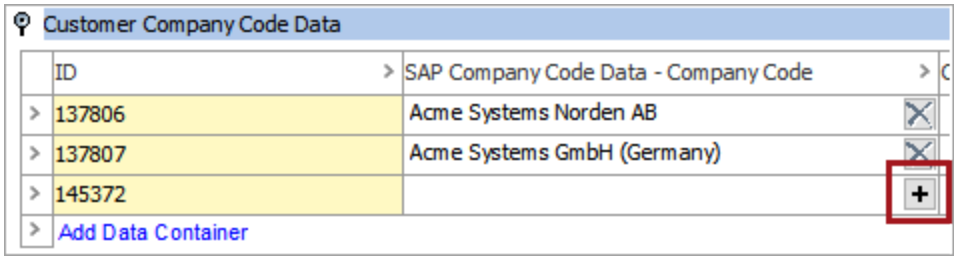




Edit a data container reference

References can also be maintained within the data container workbench tab.

1. To add a reference, navigate to the cell of the relevant reference type and click the + button.

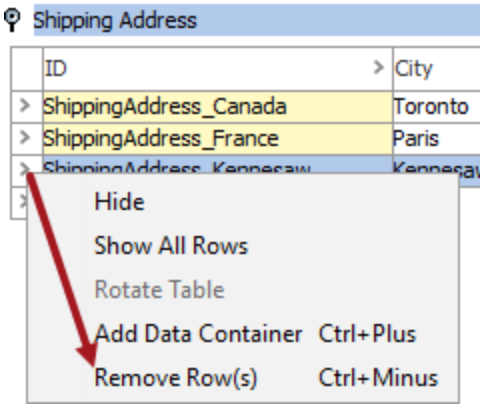


2. In the window that appears, browse or search for the desired object and click **Select**.

To delete a reference, click the ✕ button and confirm the deletion.

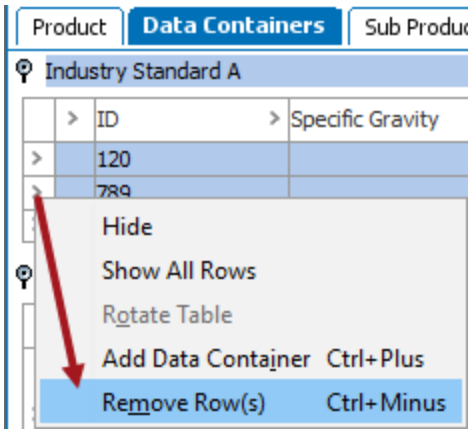
Delete a data container instance

1. Click the **>** button at the beginning of the data container row.
2. Select the **Remove Row(s)** option.



Delete multiple data container instances

1. Click the **>** button at the beginning of the data container row, press the Ctrl key on the keyboard, and continue to click the > button at the beginning of each row to be deleted.
2. Select the **Remove Row(s)** option.



Note: If a data container that was originally created using a manually input ID is deleted, that same ID will not be able to be used again for another data container. To avoid this issue, it is recommended that data containers are created using auto-generated IDs, or in the case where data containers need to be identified using specific details (e.g., a name or number that describes what the data container entails), using data container keys. For information regarding creating auto-generated IDs for data containers, see the **Setting Up Data Container Types in Workbench** topic. For information regarding data container keys, see the **Data Container Keys** topic.

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

Metadata Attributes

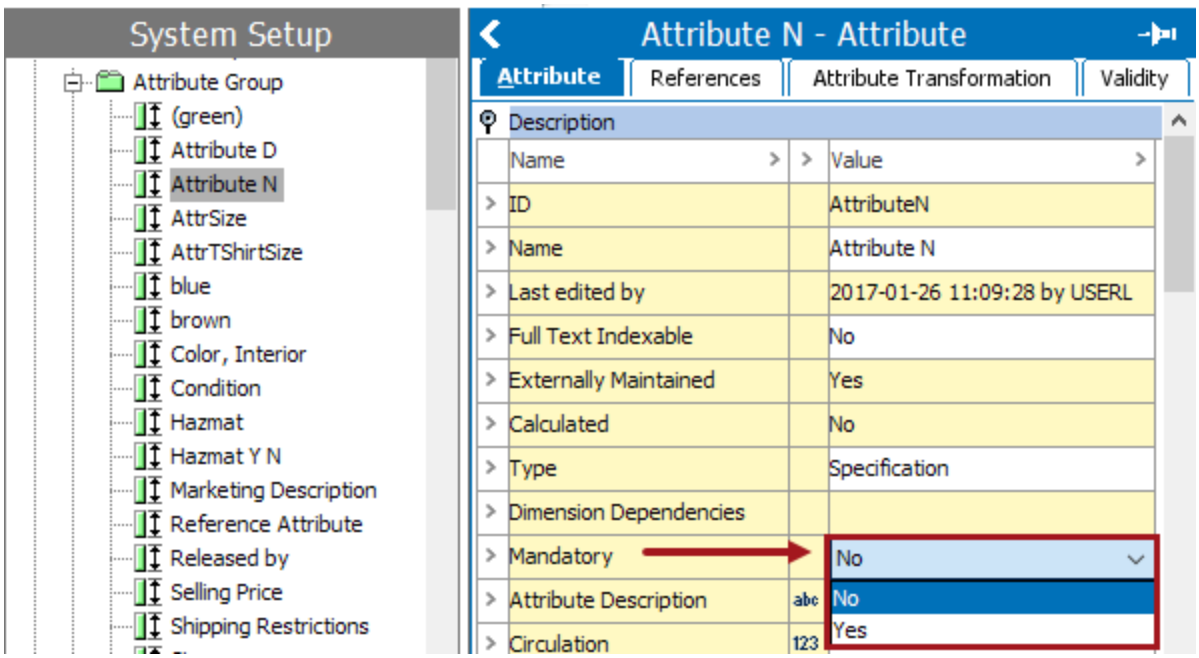
A metadata attribute is a description attribute that can be applied to the link between two objects. A link is, for example, a product to product reference or attribute(s) linked to the product(s). Metadata attributes hold information specific to a link between two nodes.

The following are metadata 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.

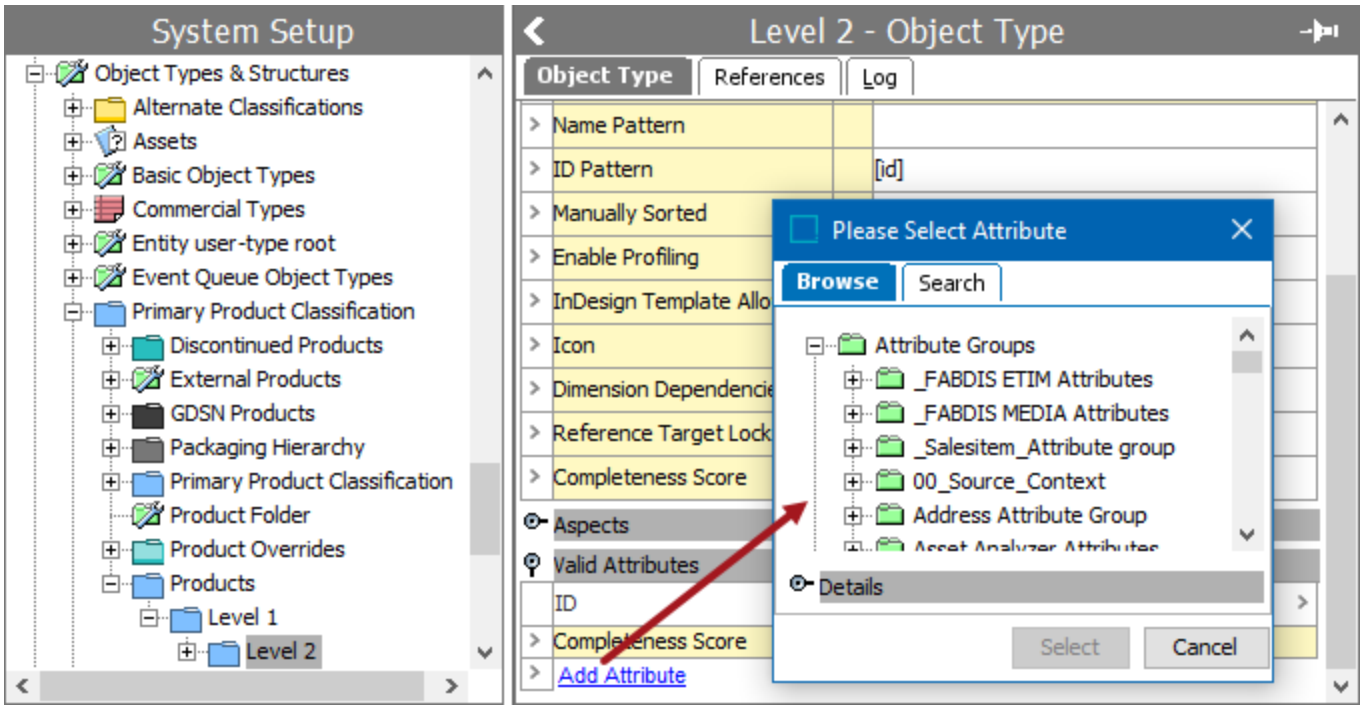


For more information about mandatory option, see the **Product Information Manager Default Settings** section of the **System Setup / Super User Guide** documentation.

Linking Description Attributes to an Object Type

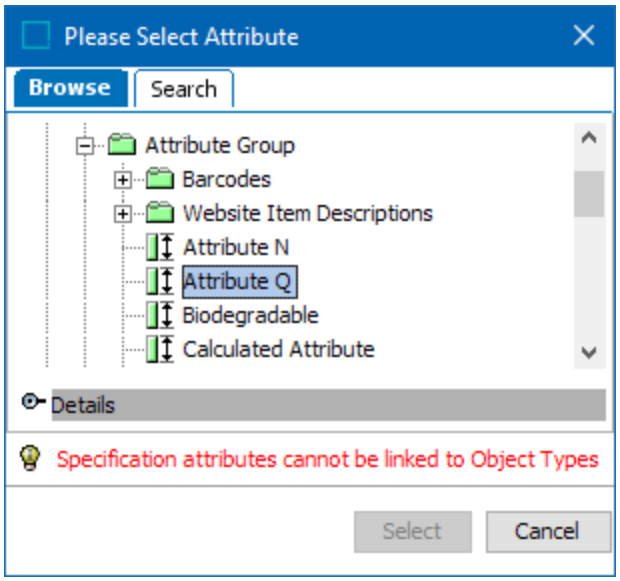
Once description attributes are created, they need to be linked to an object type, as follows.

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.

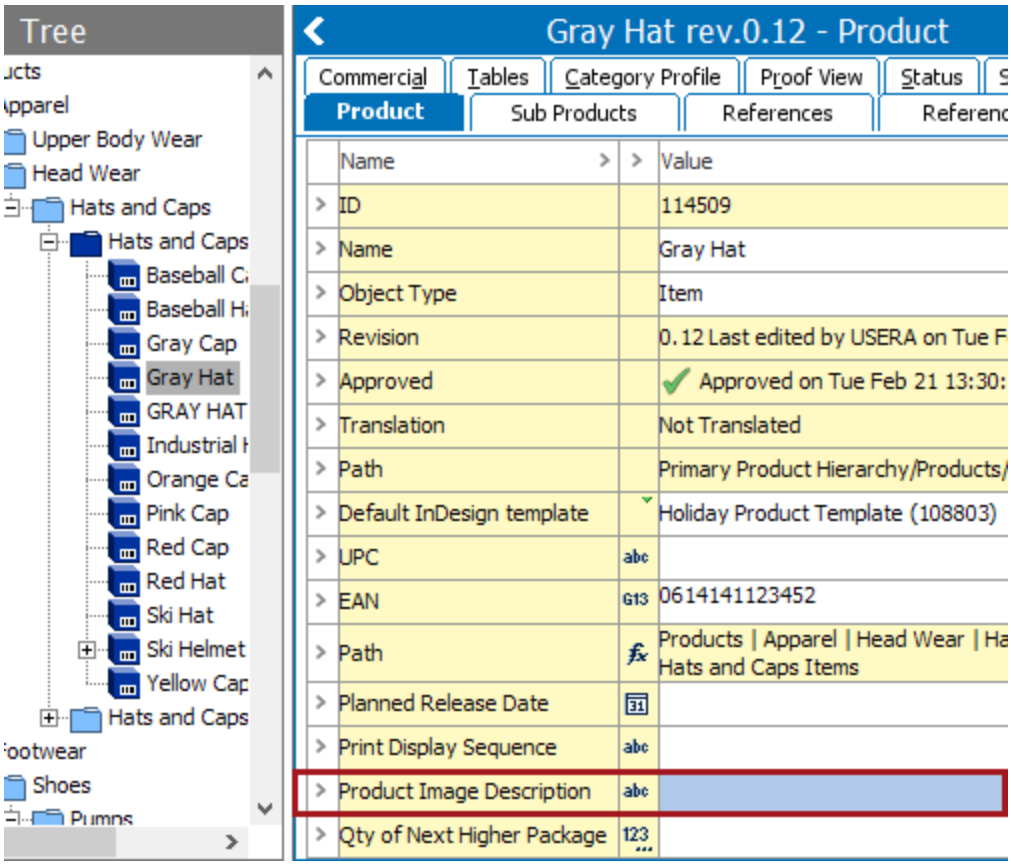


3. Search or browse to the relevant description attribute, select it and click the **Select** button.

> **Note:** The dialog shows both description and specification attributes. A message is displayed when a specification attribute is selected. The 'Select' option will be disabled 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.

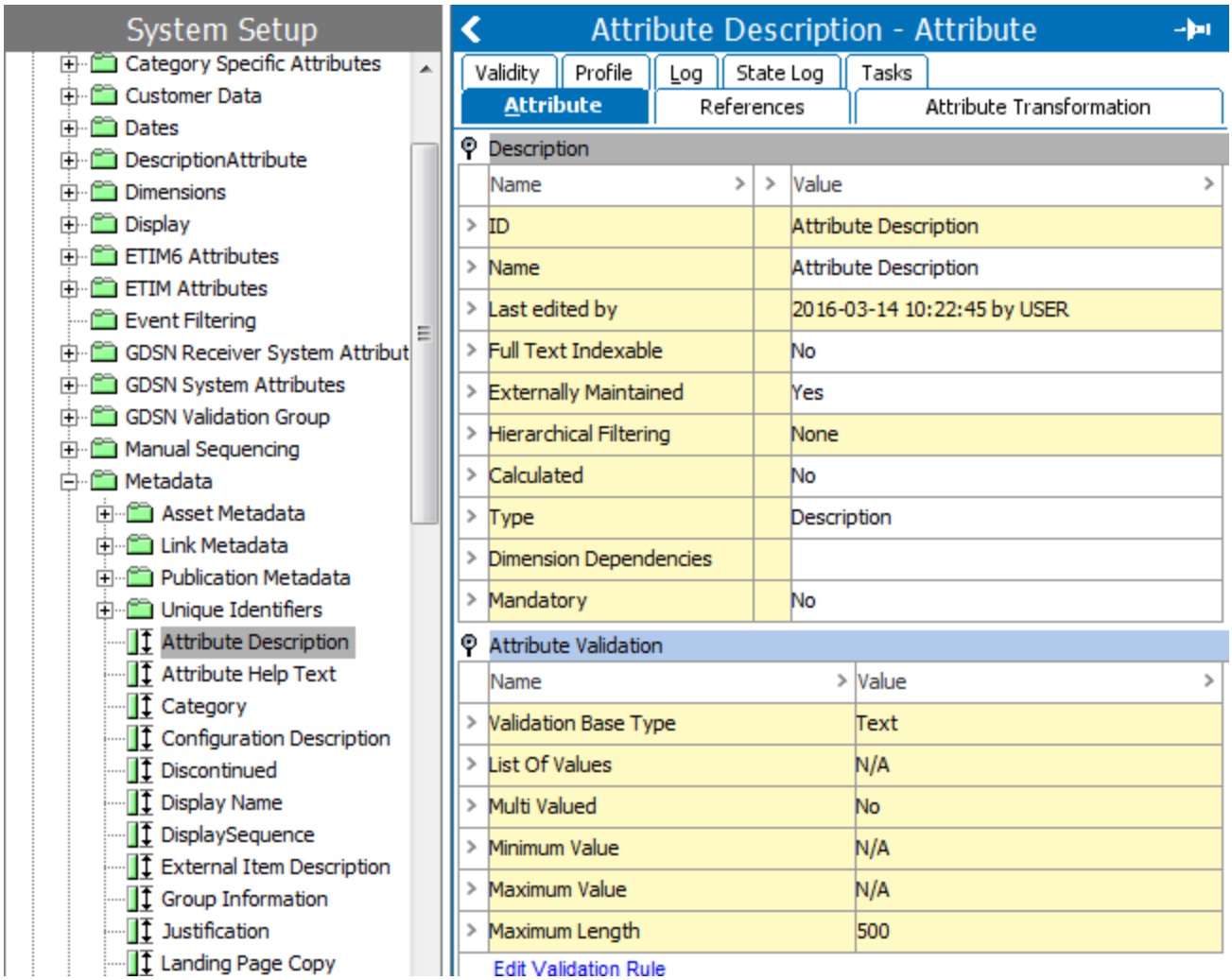

Tree
Upper Body Wear
Head Wear
Hats and Caps
Hats and Caps
Gray Cap
Gray Hat
GRAY HAT
Pink Cap
Red Cap
Red Hat
Ski Hat
Ski Helmet
Yellow Cap
Hats and Caps
Shoes
Gray Hat rev.0.12 - Product
Commercial
Tables
Category Profile
Proof View
Status
Product
Sub Products
References
Name
Value
ID
114509
Name
Gray Hat
Object Type
Item
Revision
0.12 Last edited by USERA on Tue F
Approved
Approved on Tue Feb 21 13:30:
Translation
Not Translated
Path
Primary Product Hierarchy/Products/
Default InDesign template
Holiday Product Template (108803)
UPC
EAN
0614141123452
Path
Products | Apparel | Head Wear | Ha
Hats and Caps Items
Planned Release Date
Print Display Sequence
Product Image Description
Qty of Next Higher Package

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.

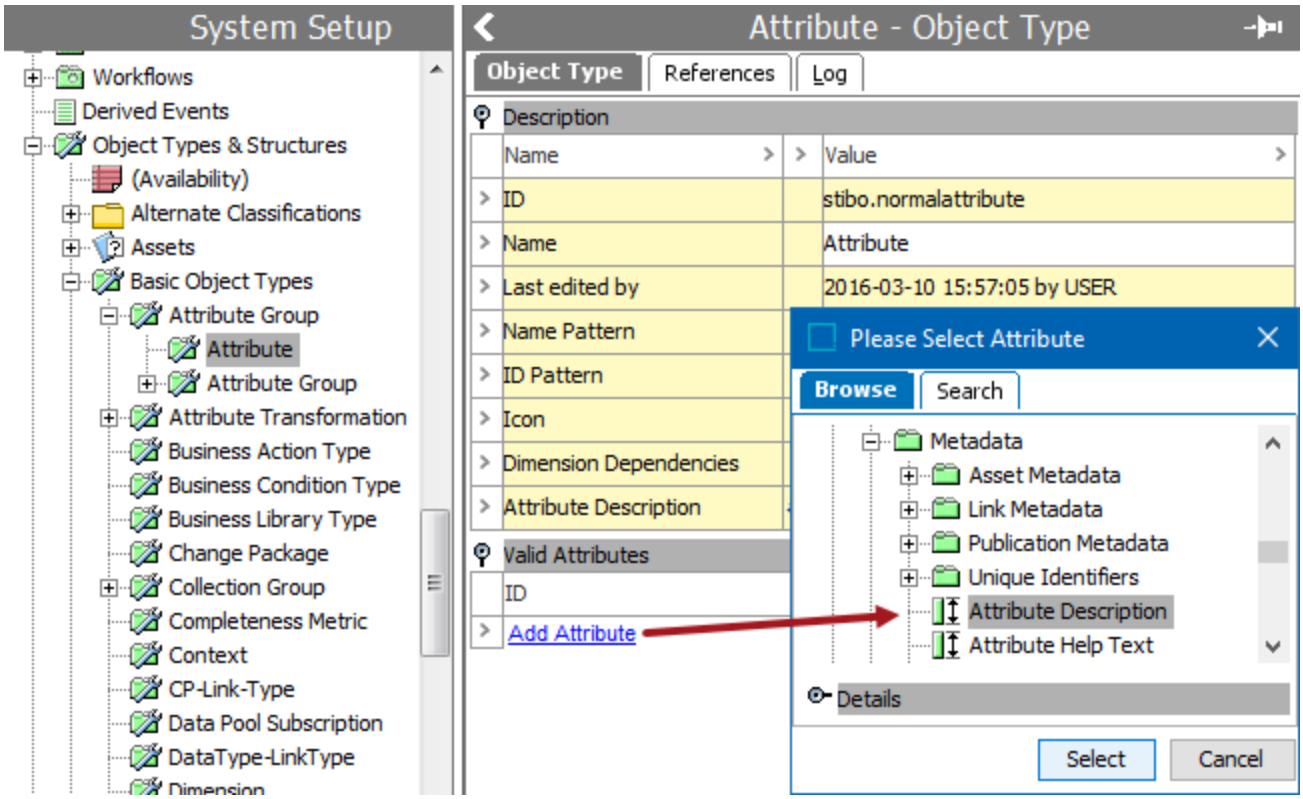
Create a Metadata Attribute on an Attribute

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.



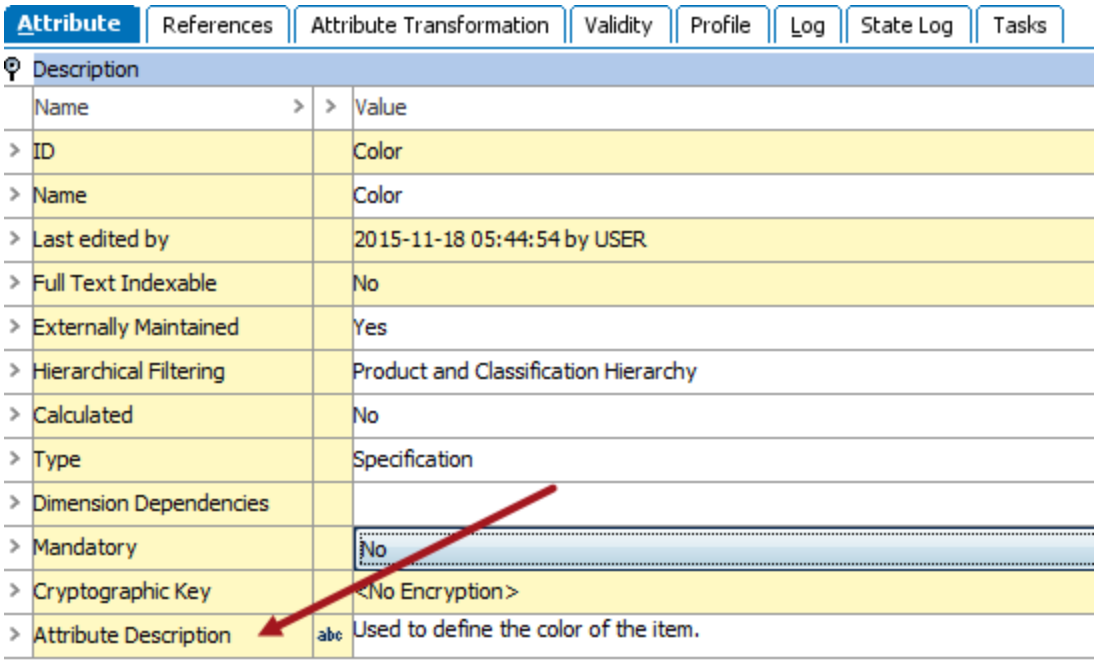
2. After creating the metadata attribute, the attribute must be made valid for the **attribute** object type. Navigate to the attribute object type under System Setup > Object Types & Structures > Basic Object Types > Attribute Group. Select the attribute object type, then click **Add Attribute**. The **Please Select Attribute** dialog displays.

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



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

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

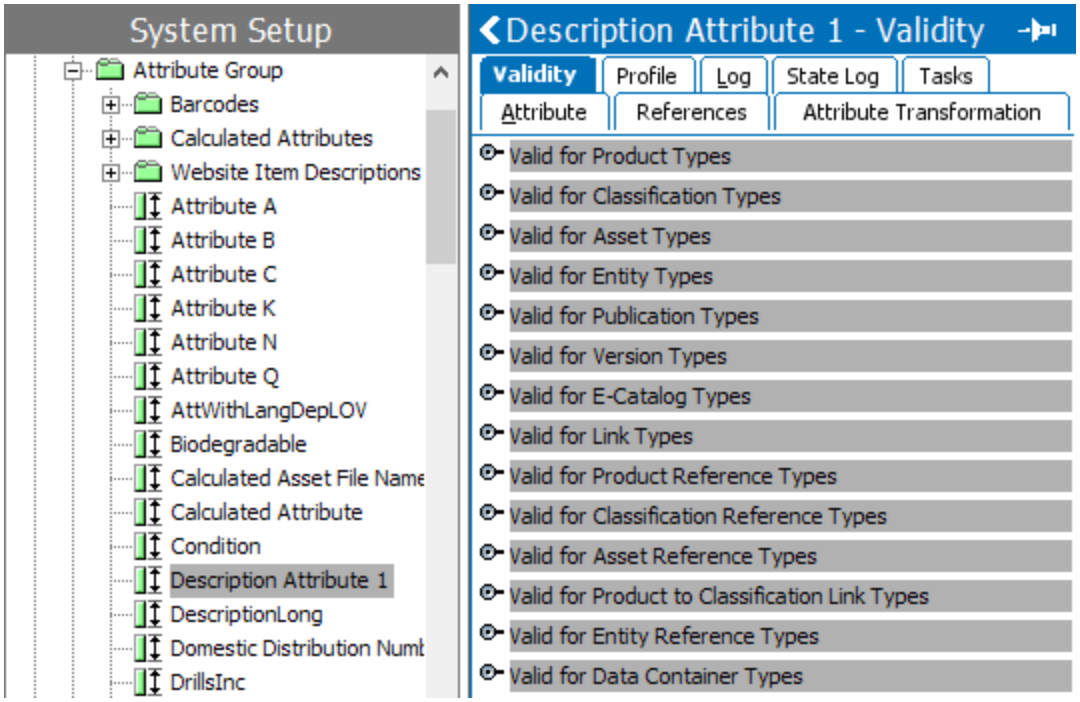


For more information on implementing global help text for attributes, see the **Attribute Help Text for Web UI** topic.

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.

Object Type	Attribute
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.

Validity on Description Attributes

Setting validity for a description attribute identifies when it will display for the selected object types, as defined in the following table.

Object Type	Description
Product	The attribute will be valid for the selected product object types and will display in the Description flipper. **Note:** By default, the attribute is set to be valid for all product object types.
Classification	The attribute will be valid for the selected classification object types and will display in the Description flipper.
Asset	The attribute will be valid for the selected asset object types (Images & Documents) and will display in the Description flipper.
Entity	The attribute will be valid for the selected entity object types and will display in the Description flipper.
Publication	The attribute will be valid for the selected publication object types and will display in the Description flipper.
Attribute Link Types	The attribute will be valid for attribute link types and will display in the Product Attribute Link Type or the Classification Attribute Link Type in the 'Attributes' flipper.
Reference Types	The attribute will be valid for the selected link types and will display 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.

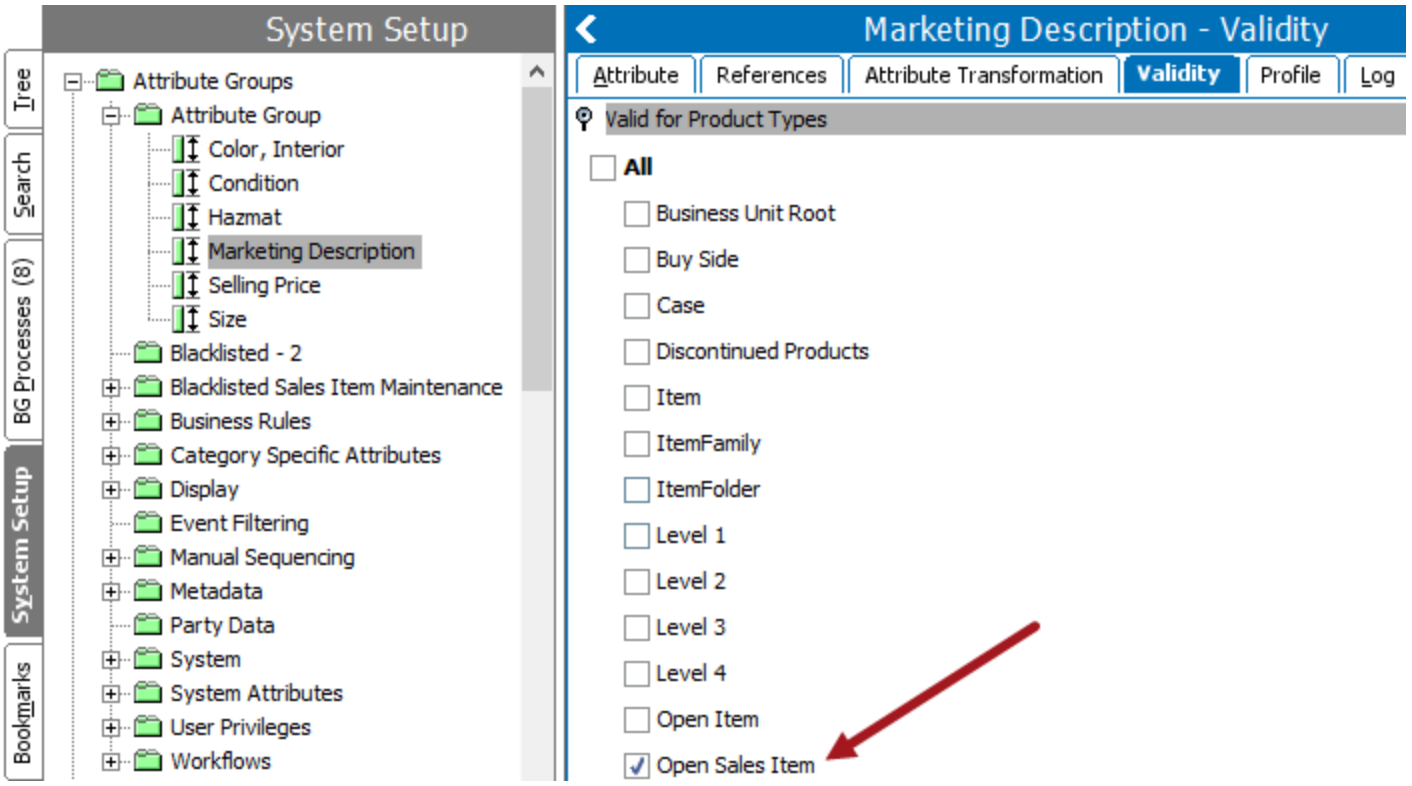
Note: The STEPXML format identifies selected Validity object types within the UserTypeLink tag using the UserTypeID element.

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.

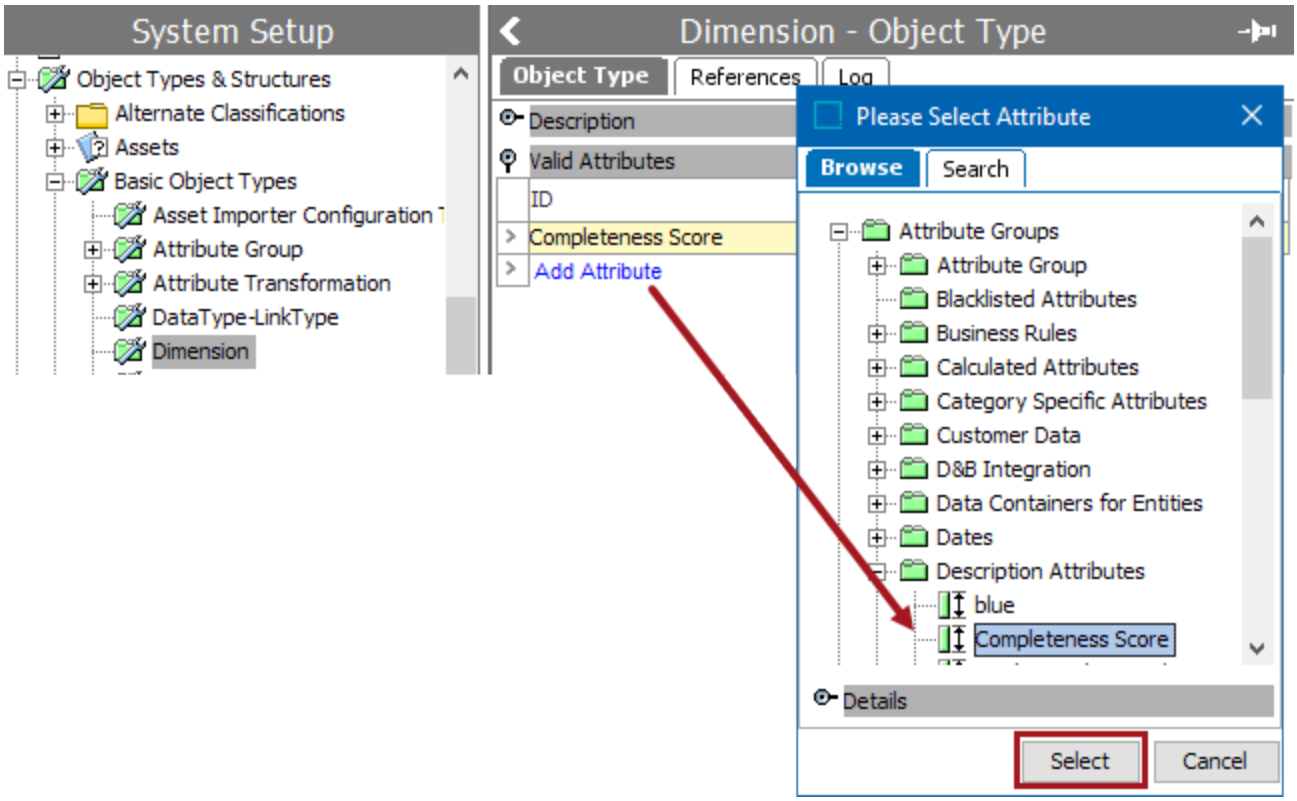


3. Use the checkboxes to select the object types for which the attribute should be valid.

   - Selecting a product, classification, entity, or asset object type displays the attribute in the **Description** flipper of all selected object types.
   - Selecting a link type displays the attribute on product, classification, and/or asset references.
   - Selecting a product / classification attribute link type makes the attribute visible on the Linked Attributes flippers of product / 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.

4. If needed, add the description attribute on any object type (attribute, attribute groups, users, user groups, etc.).

   - As shown below, in System Setup > Object Type and Structures > select 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.

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 Recommended Practices** topic for more information on how to work with dimension-dependent attribute object types.

Dimension Dependent Attribute Values

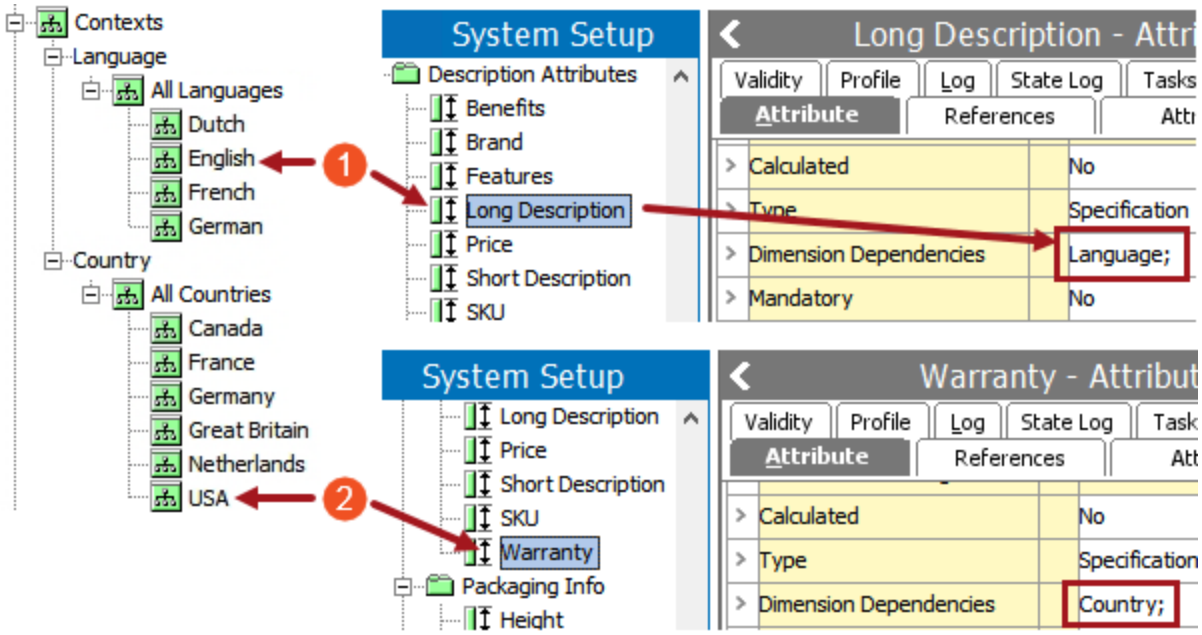
An example of an attribute where the values are dimension dependent on language could be 'Primary Color.' The following also applies to values contained in STEP **names** of objects.

- 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.

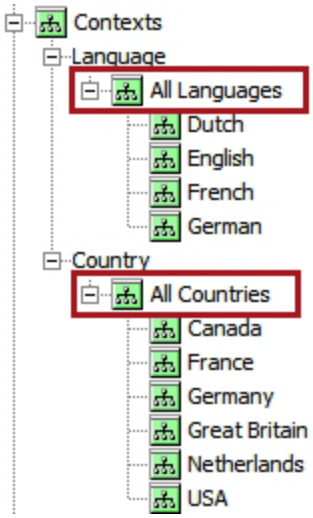
1. The attribute 'Long Description' is dimension dependent on **Language**, meaning that its content may differ based on what language dimension point is used in the context. The typical usage of dimension-dependent data is when content is translated, allowing an attribute to have different values for 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.



Non-Dimension Dependent Attribute Values

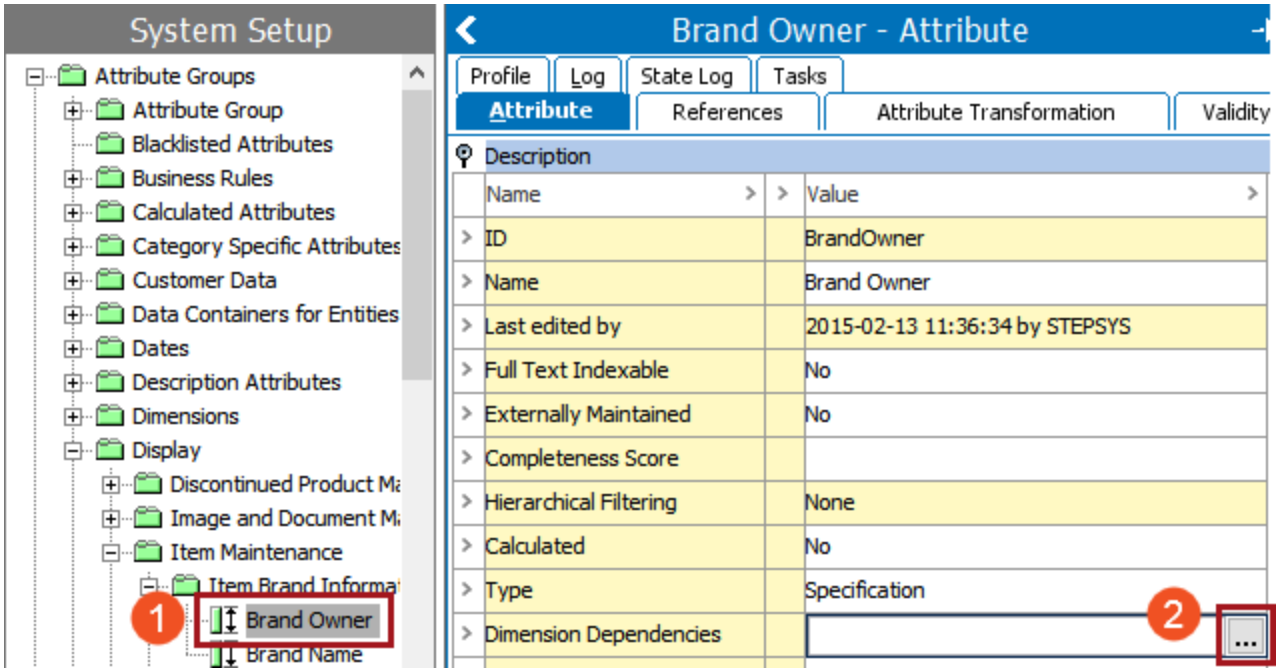
No dimension dependency means that data is stored in the **All** level (also known as the **Root** level) of all dimensions. The following also applies to values contained in STEP **names** of objects.



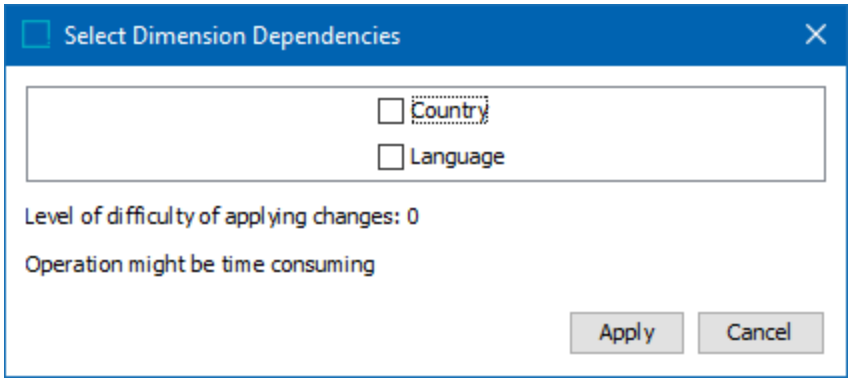
Note: If you have information that will never vary by language or country, you should **never** assign a dimension dependency on it. When you **do** have information that truly will be different by language or country, assign the dimension dependency right away. If you later apply a dimension dependency 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

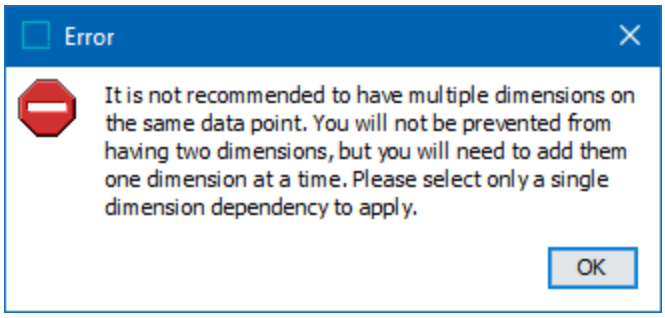
1. In System Setup, navigate to the relevant attribute under Attribute Groups.
2. On the **Attribute** tab, double-click in the **Dimension Dependencies** field, then click the ellipsis button (...).



3. A **Select Dimension Dependencies** dialog box displays showing the configured dimensions accompanied with checkboxes.

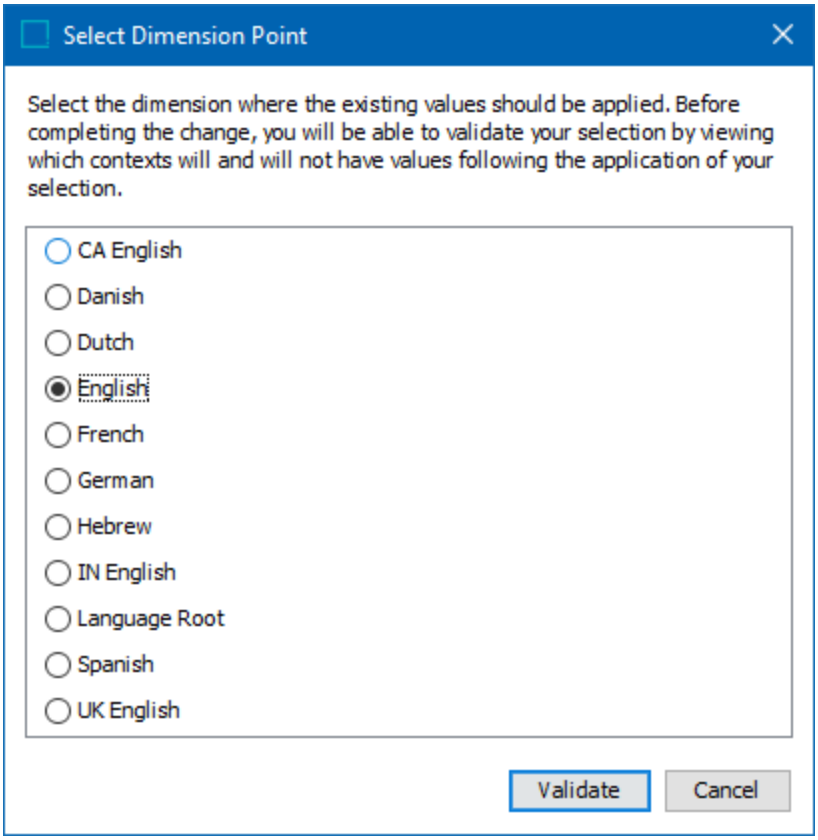


Only one dimension dependency can be selected in a single action, with only two dimension dependencies applied in total. If a user attempts to check two dimensions at once, an error will display:



4. Check the desired box for the relevant dimension(s), then click **Apply** to set the dimension dependency.
5. When adding a dimension dependency to an attribute or LOV, a dialog box will display that enables the user to select the one target dimension to which the existing attribute's or LOV's data should be applied.

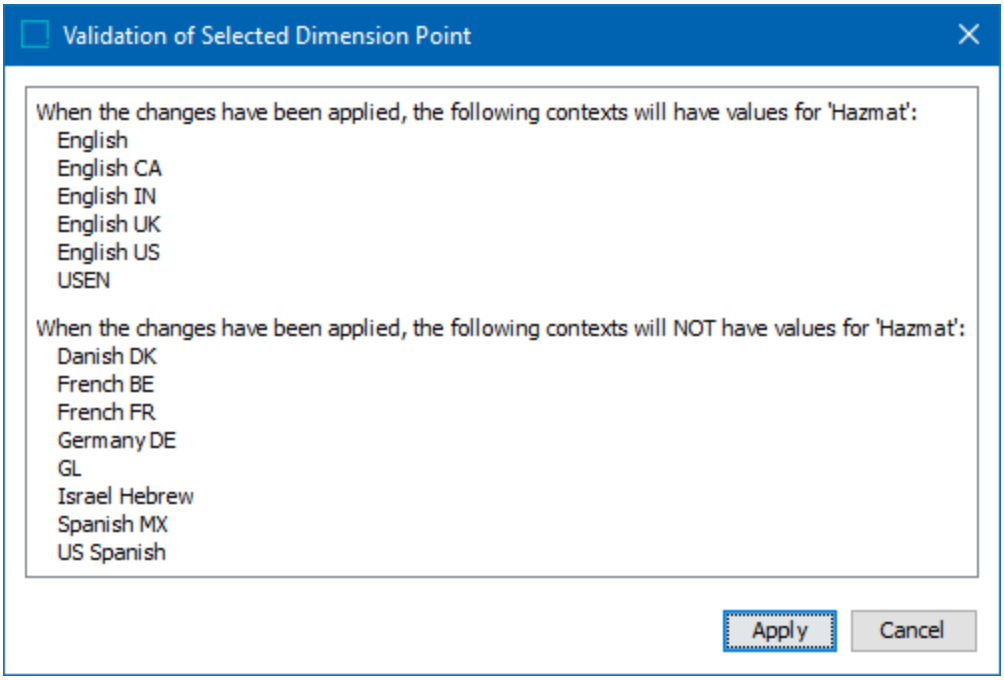
   When a language dimension dependency is being applied, a 'Select Dimension Point' window will display showing all configured dimension points specific to the selected dimension.



Note: Recommended practice for the majority of use cases is not to select the 'All' or 'Global' dimension point (or, as shown in the screenshot above, 'Language Root'). The reason for this is that when the global dimension point is selected, the attribute or LOV data is stored on the global dimension point. However,

he user might expect that by selecting the global dimension point, the data is being stored on all listed dimension point, which is not necessarily the case.

Once the desired dimension point is selected, the 'Validate' button will become active. Once clicked, a 'Validation of Selected Dimension Point' popup will display:

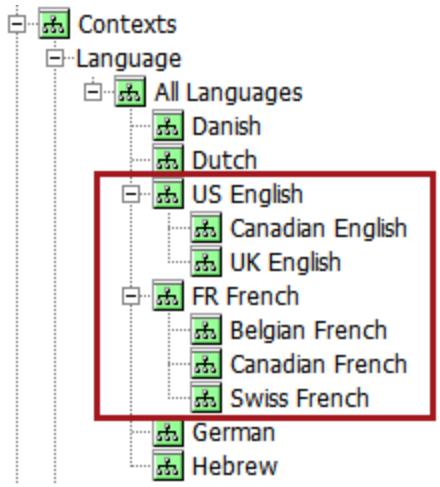


As is made clear by the above screenshot, this validation pop-up lists first which dimension points the data will be applied to, and then lists the dimension points to which no data will be applied. If additional contexts need values, those values can be applied via export from the context that contains the values and then imported into the contexts to which the values should also be applied.

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.

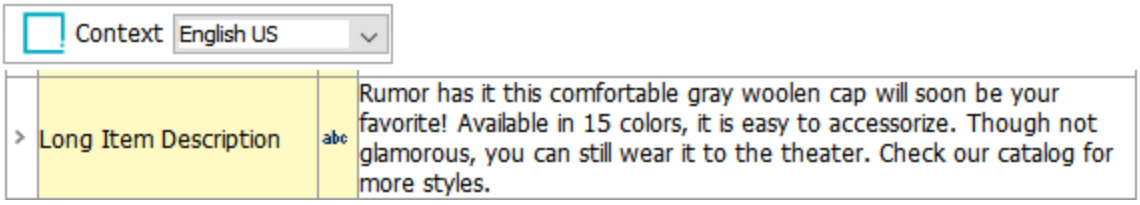
Note: This is not recommended. In most circumstances, dimension points should be created at the same level. For more information on recommended practices with regard to dimensions, see the **Dimension and Dimension Points Recommended Practices** topic.



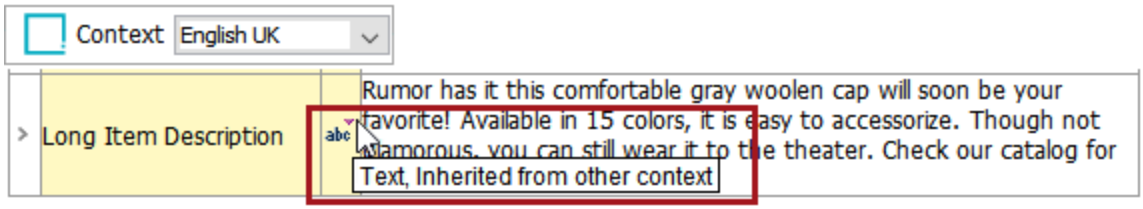
There may be valid use cases to maintain several versions of the same language, for example, US English, Canadian English, and UK English. Since maintaining translations requires effort and carries a cost, it may be of benefit to maintain a base version of English and then only translate a small percentage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

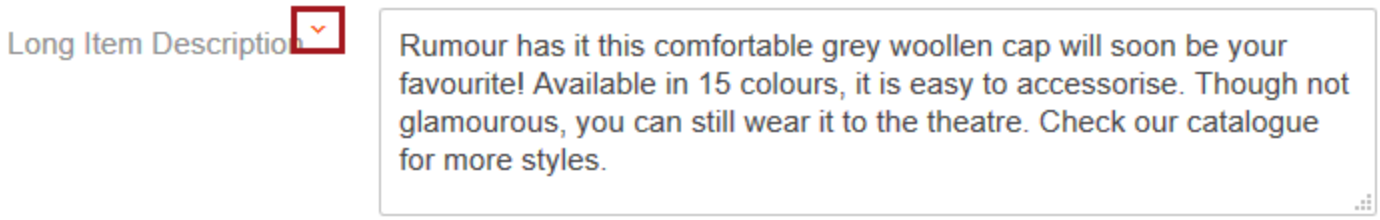
For example, the value of Long Item Description in the parent **US English** language dimension (which is used by the English US context) looks as follows:



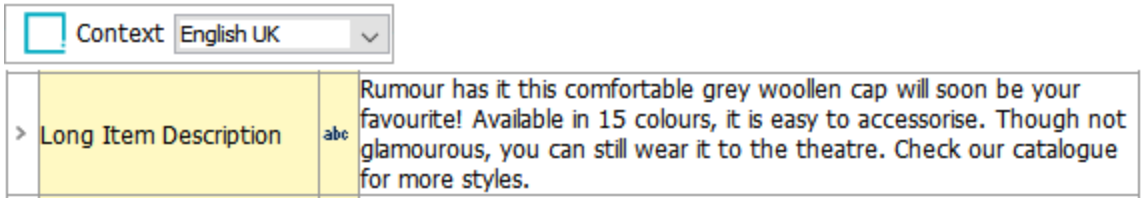
Before translation, the content will inherit to the child context of **UK English** as-is. Inheritance of the value is indicated by a purple triangle. When the triangle is hovered over, the help text 'Text, inherited from other context' displays.



It is worth noting that, in the Web UI, the indicator of an attribute with a value inherited from another context is a downward-pointing red arrow.



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.



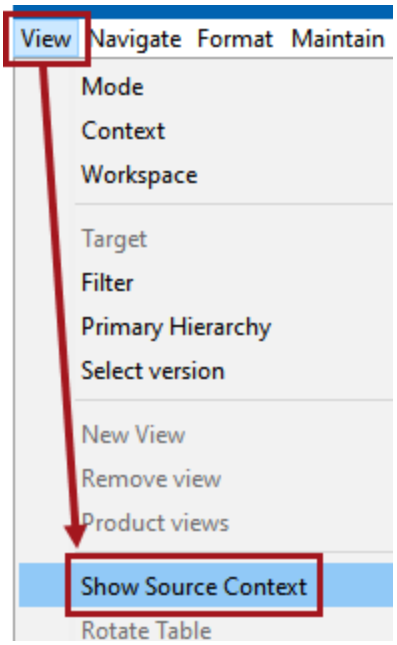
This can save on translation costs because, instead of sending US English content to a translation company to convert it to UK English, the US English content could be exported from STEP, a user could run a find and replace operation on the content to replace a certain subset of words within the content (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.

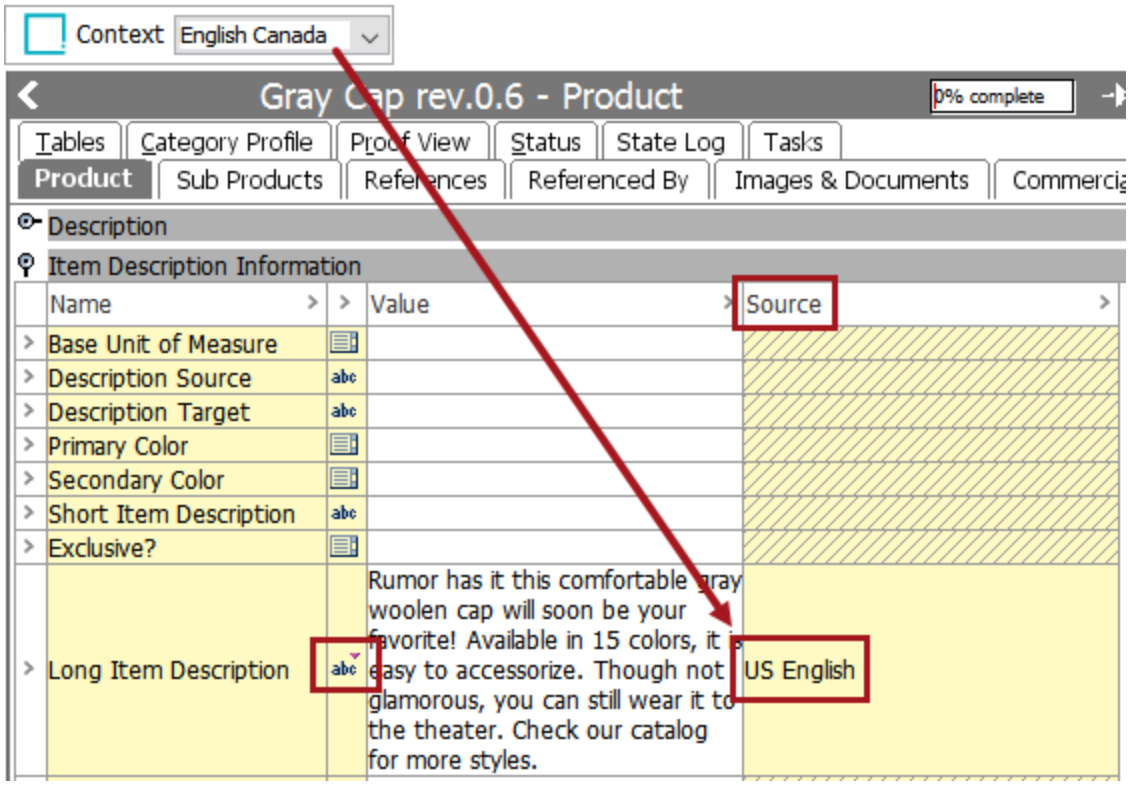
Continuing from the example used in the previous section of this topic, the following steps illustrate the content of the Long Item Description attribute as viewed from the English Canada context and from within the English UK context.

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**.

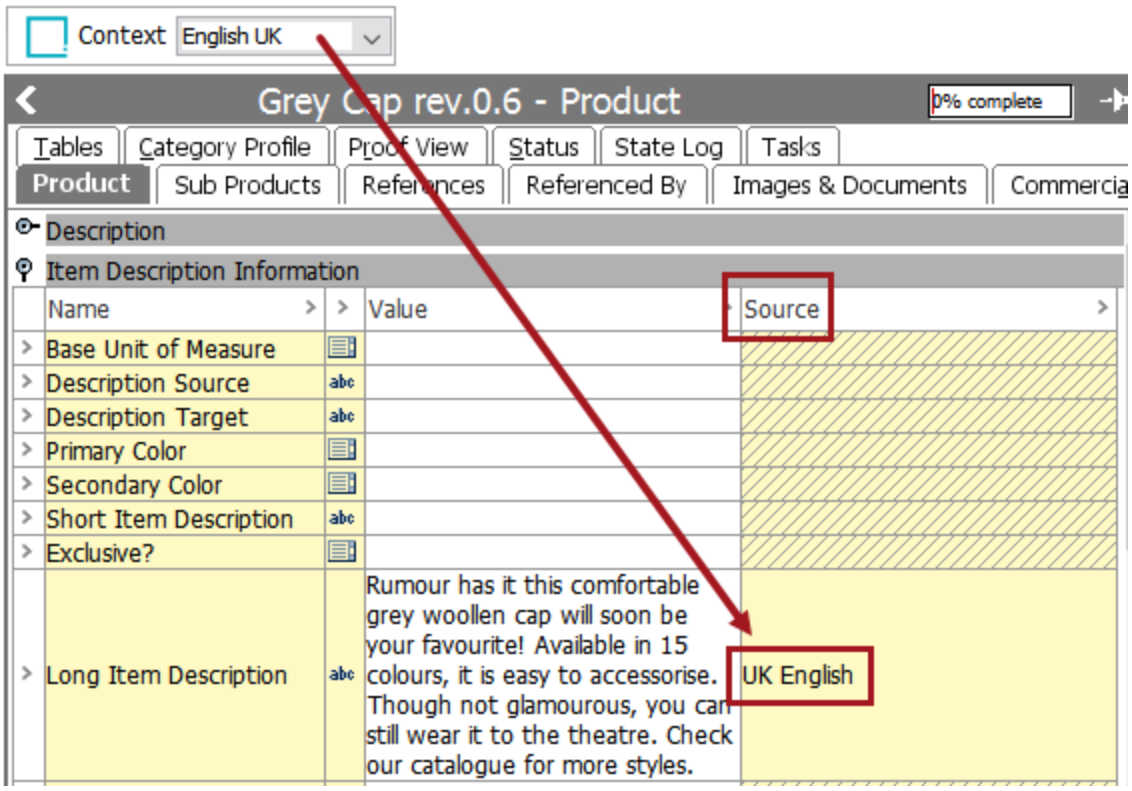


4. A **Source** column displays to the right of the Value column that indicates the source dimension point of the values.

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



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.



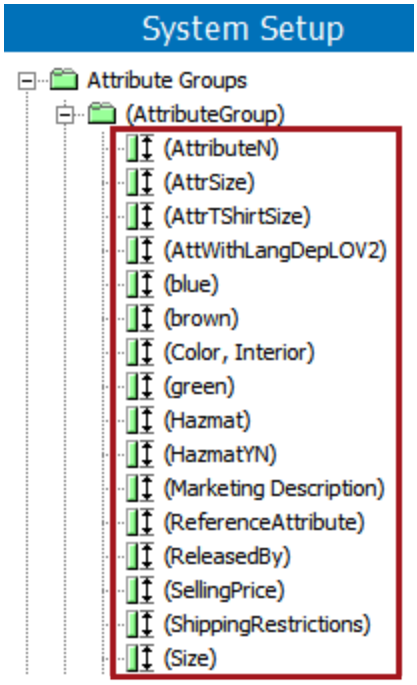
Dimension Dependent Attribute and Attribute Group Recommended Practices

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.

Attributes That Need Dimension Dependencies

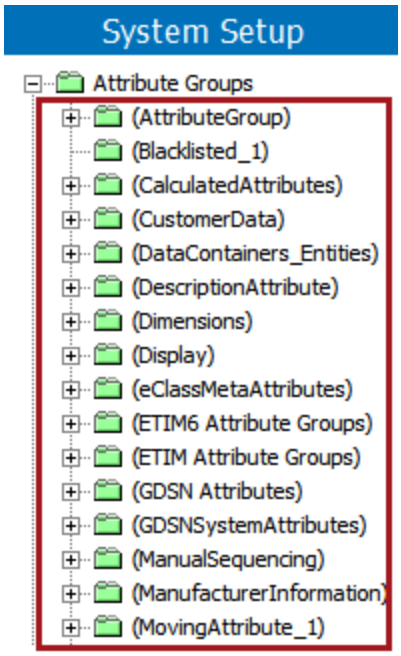
Questions to ask	Recommendation
Would the data be presented differently if it were published in another language (for example, long descriptions or feature bullets)?	If so, make the attribute language dependent, even if you only have one language today.
Would the data appear exactly the same if it were 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, recommended 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 dropdown 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.

Display Sequence Attribute

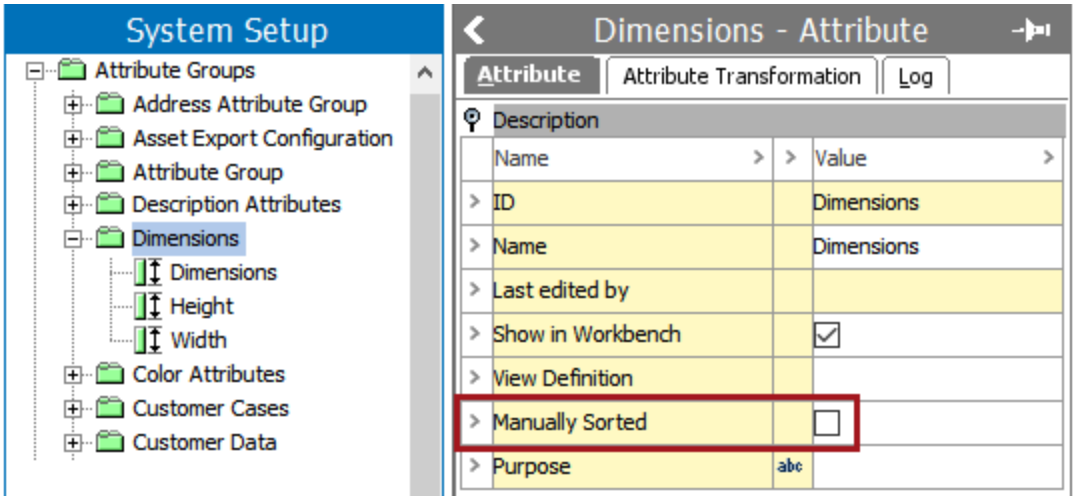
The default function in STEP is that attributes shown in the editor of a product are displayed in alphabetical order. However, if a Display Sequence Attribute has been set up, the attributes shown in the editor are sorted according to the value of the Display Sequence attribute.

Important: The Display Sequence Attribute has no impact on the order in which attributes are exported.

Display Sequence Attribute 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 a display sequence attribute, you must verify that 'Manual Sorting' is not selected on the attribute groups in which the relevant attributes are contained. If enabled on an attribute group, the Manual Sort option will overwrite any attribute sequences you configure. To check if the manual sorting is enabled, select the relevant attribute group in System Setup as shown below.



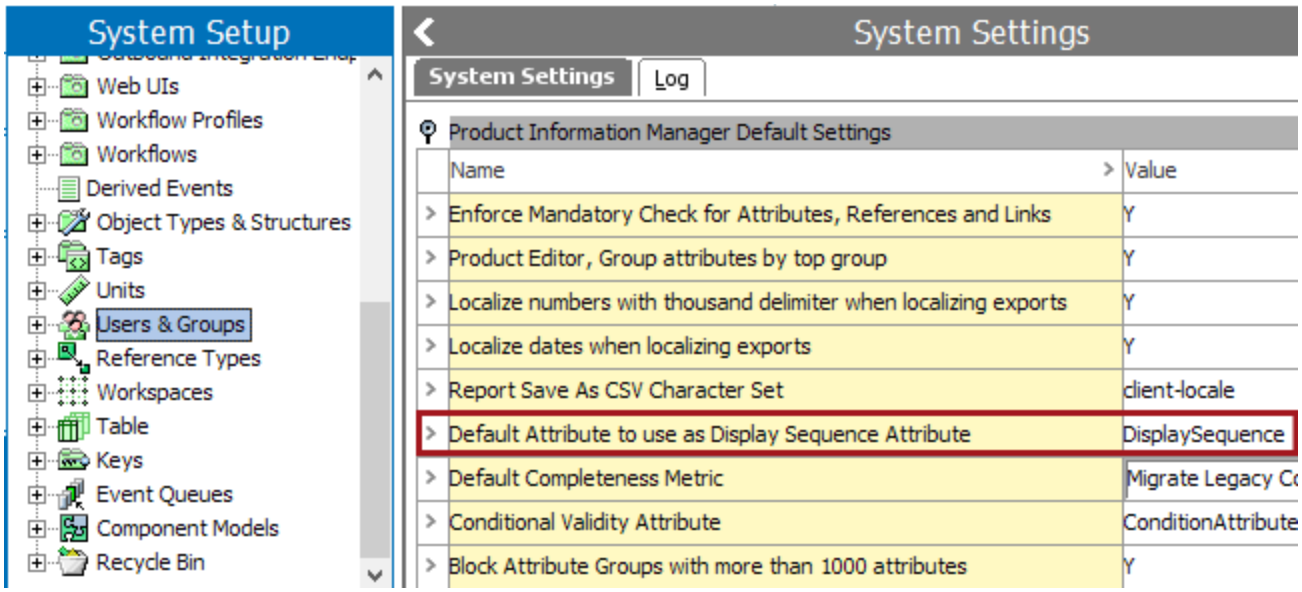
Set Up Display Sequence Attribute

A Display Sequence Attribute is a description attribute that dictates how other attributes will be ordered in an object editor. To configure a Display Sequence Attribute, it must:

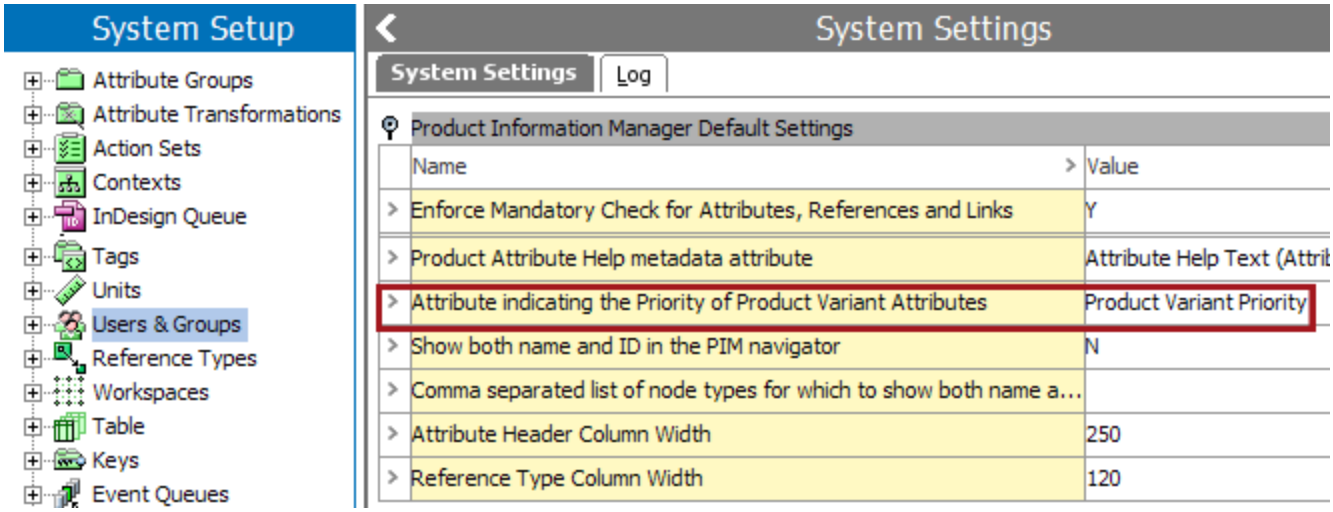
- be a **Description** attribute.
- have a Validation Base Type of either **number** or **integer**.
- be valid for the link type **Product attribute validation** and/or valid for the object type **Attribute**.

To create this type of attribute, see the **Creating Attributes** topic.

Select the display sequence attribute in System Setup > Users & Groups > System Settings tab > Product Information Manager Default Settings flipper > **Default Attribute to use as Attribute Display Sequence Attribute**.



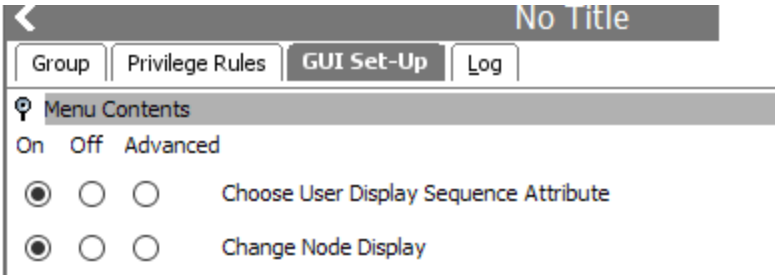
Similarly, select the display sequence for the product variant object types in System Setup> Users & Groups > System Settings tab > Product Information Manager Default Settings flipper > **Attribute indicating the Priority of Product Variant Attributes**.



View Menu: Choose 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 as follows:

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.



3. Set the **Choose User Display Sequence Attribute** radio button to Off'. By default it is set to 'On'.

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.

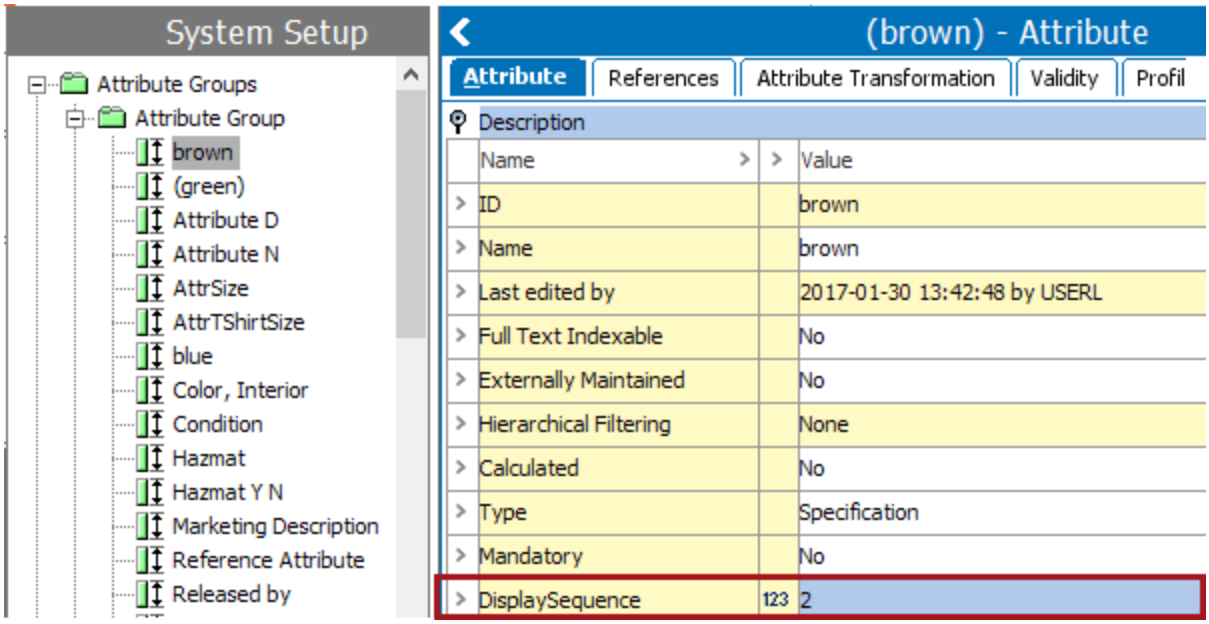
Sorting Attributes

Once the display sequence attribute has been set up, a user can establish the sorting. The attributes shown in the product editor are sorted according to the order set in the display sequence attribute. This can be done in two ways:

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

Setting a display sequence number on the attribute

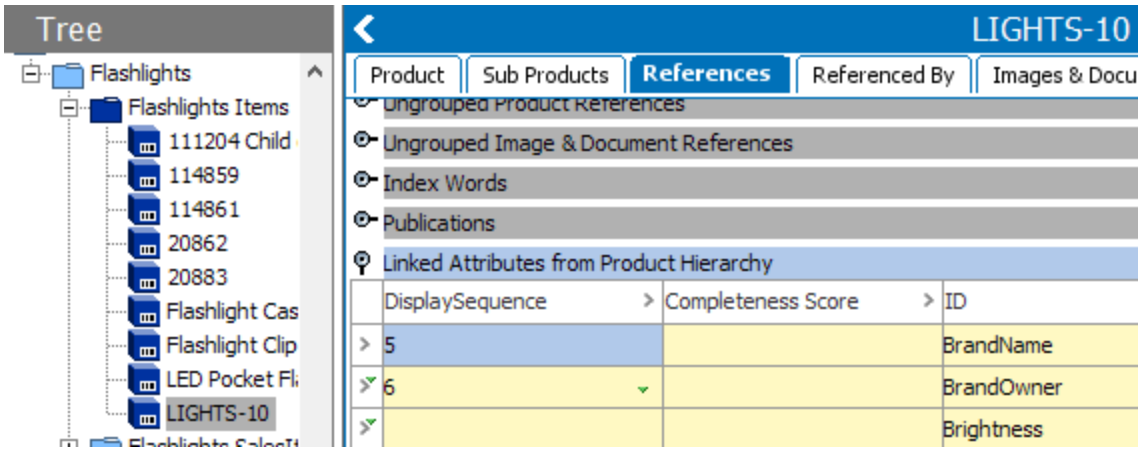
1. In System Setup, expand **Attribute Groups**, then click the relevant attribute. An **Attribute** editor displays.
2. In the Display Sequence field, type the desired number for each attribute in an attribute group. The lower the value, the higher priority it will have compared to other attributes. 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.



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

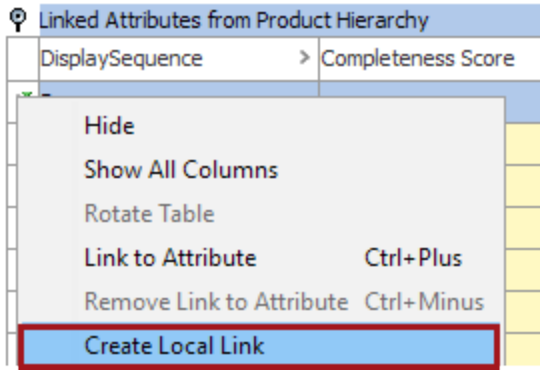
Setting a display sequence number under Linked Attributes

1. In the Tree, expand the Product Hierarchy and then click the relevant product folder or product.
2. Click the **References** tab.
3. Open the Linked Attributes from Product Hierarchy flipper, type a number in the cell related to the display sequence attribute column in front of the relevant attribute, as shown below.

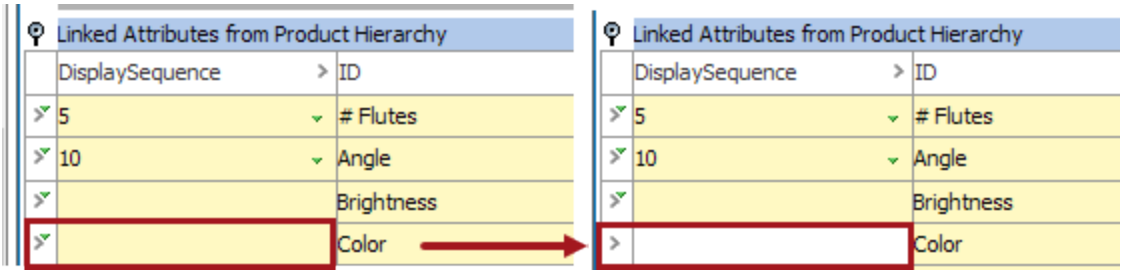


Note: The display sequence attribute column can only be edited if the attribute is linked in at the product hierarchy level you have selected. The **Inherited from** column shows where the attribute is linked in. An attribute can be linked in to one or more attribute groups. In the **Attribute Groups** column, you can see to which attribute group(s) the attribute belongs.

4. Optionally, if you want to override the display sequence displayed 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.



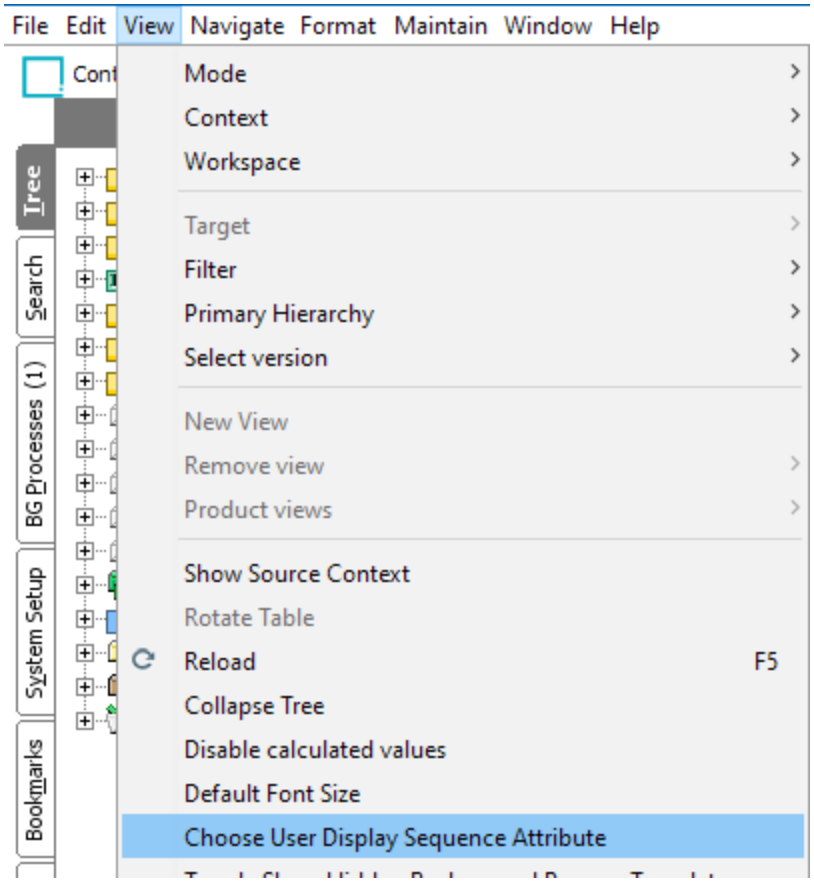
Attributes that have no sequence number, are alphabetically sorted and will be placed last on the list.

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.

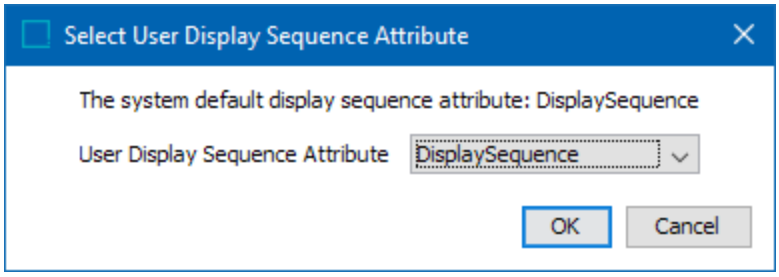
Selecting a Personal Display Sequence Attribute

If you are interested in sorting attributes in the product editor on a different display sequence attribute than the one set in System Setup >Users and Groups> Default Attribute to use as Display Sequence Attribute, do the following:

1. Go to **View** > **Choose User Display Sequence Attribute**.



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



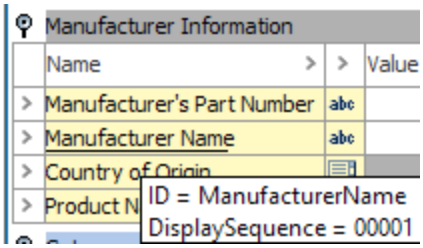
2. Select an attribute and click OK.

   In Tree, select a product object and select the **References** tab. The first column under **Linked Attributes** displays your personal display sequence attribute.

Display Sequence Attribute and the Product Editor Display

In the product editor, within an Attribute Group, the attributes are sorted on the display sequence attribute. If no display sequence attribute is set, the attributes are shown in alphabetical order.

1. Hover over the attribute name and a tool tip displays.



2. The Attribute **ID** and the display sequence attribute is shown.

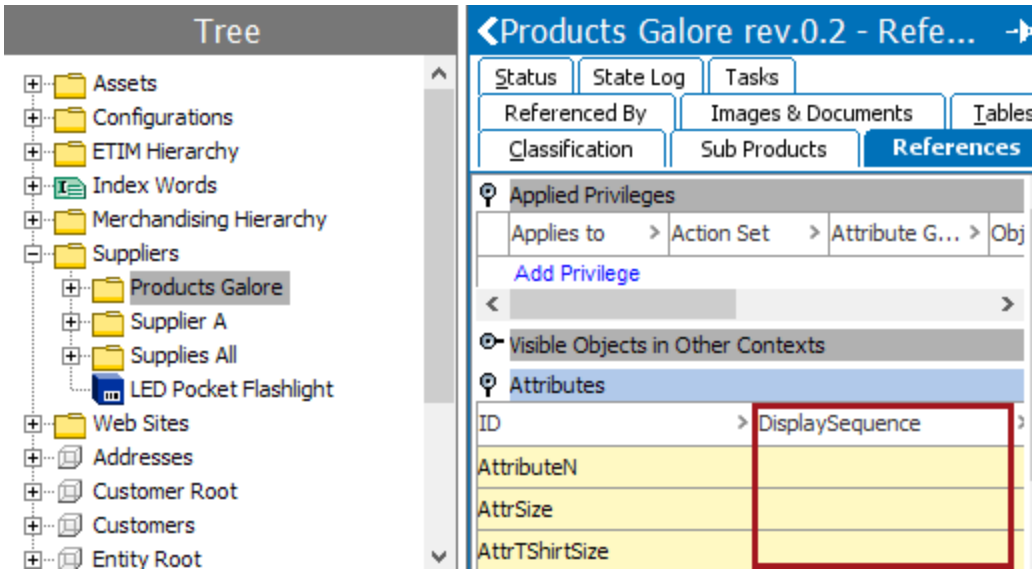
Note: When selecting two or more products, the product editor will show the attributes in alphabetical order since it is possible to select two or more products with the same attributes but with different values for the display sequence attribute.

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 and select a folder.
2. Click the **References** tab.

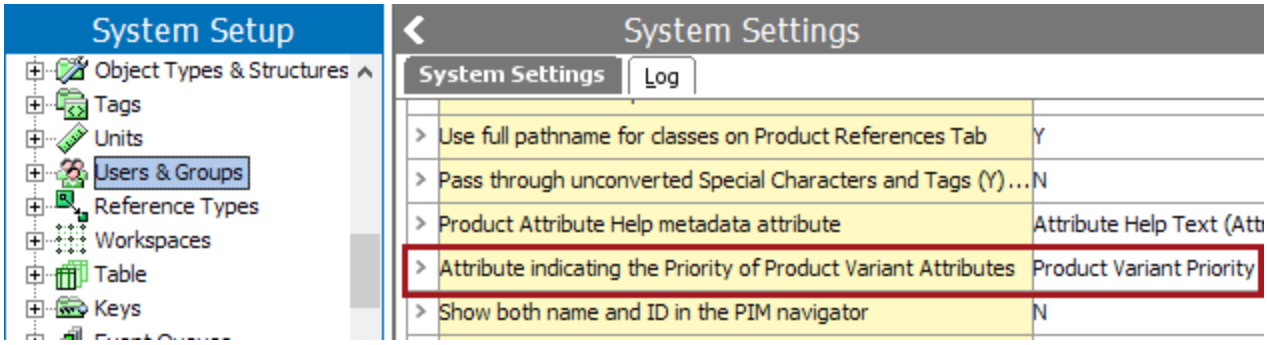
   A display sequence attribute column is shown in the Attributes flipper.



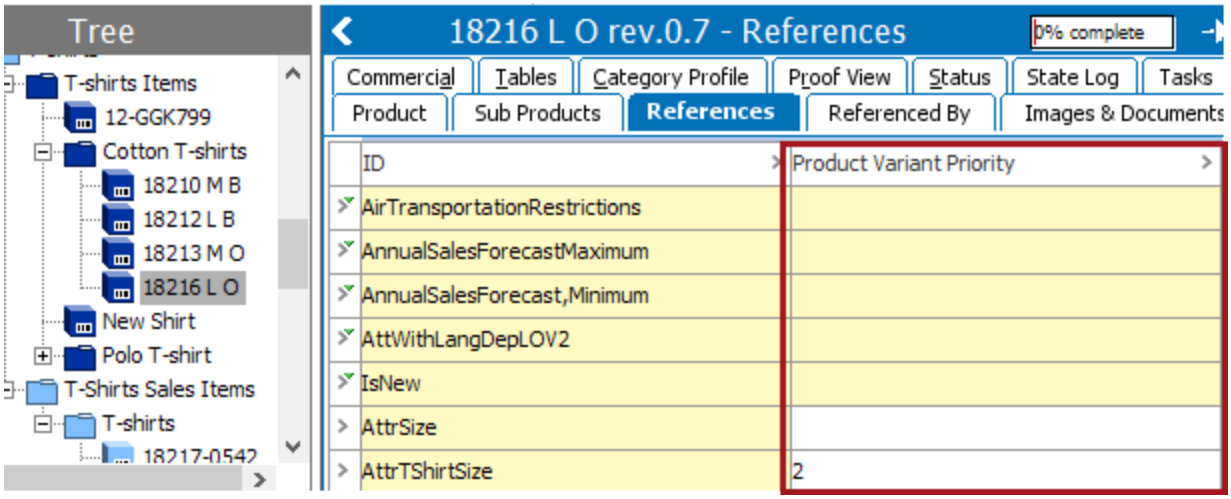
Important: Maintaining attribute display sequence in this area as no effect in either the classification editor or an export.

Setting the Priority of a Product Variant Attribute

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



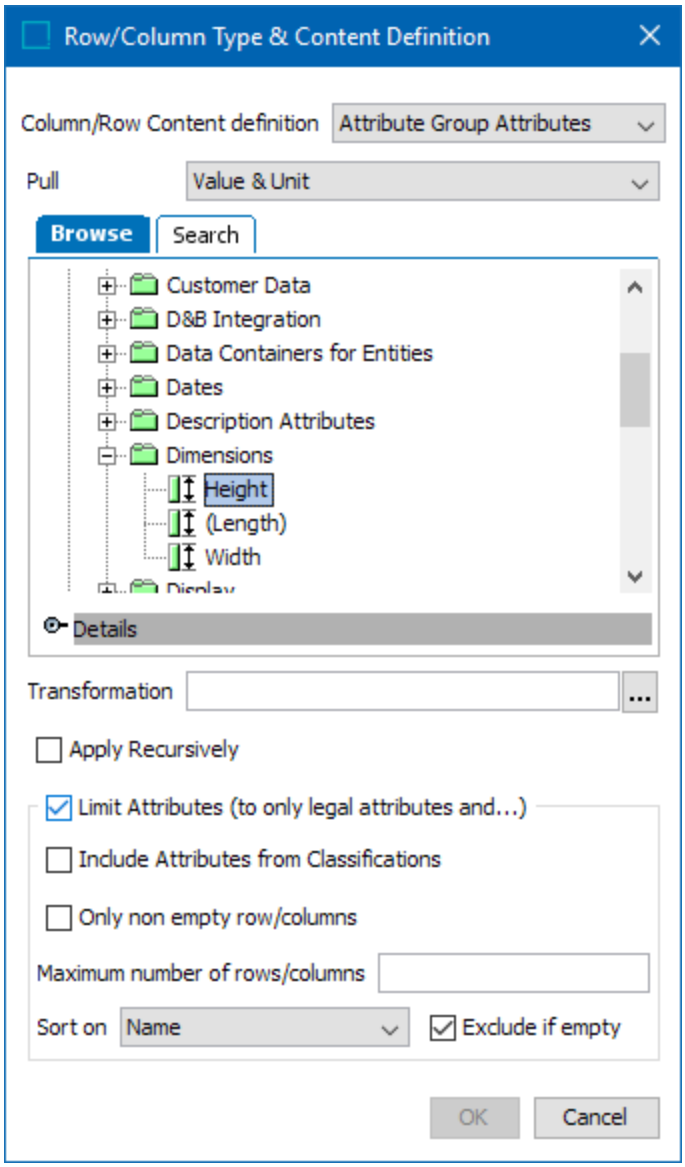
4. Enter the display sequence values in the column related to the respective attributes.



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 **Tables** documentation.

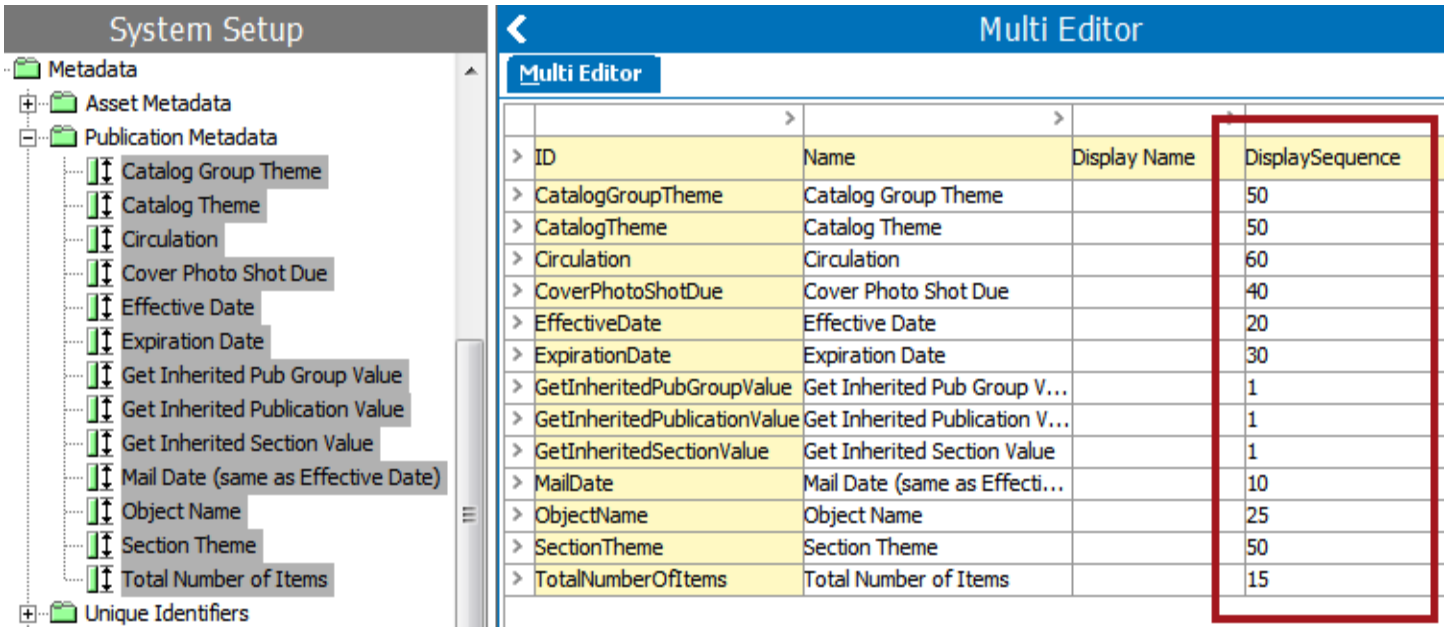
Shown in the screenshot below, in the Sort On parameter, select the display sequence attribute that will determine the order in which the attributes within an attribute group will show in a table.



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

Metadata (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. 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

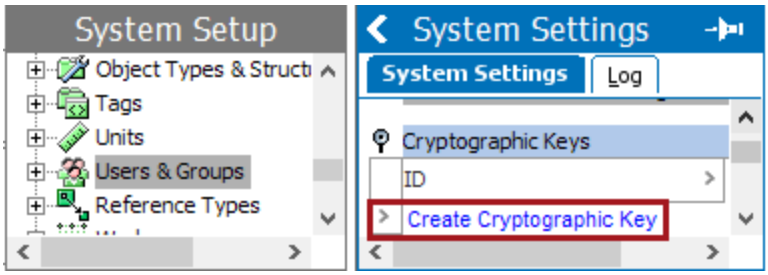
Encrypted Attributes

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.

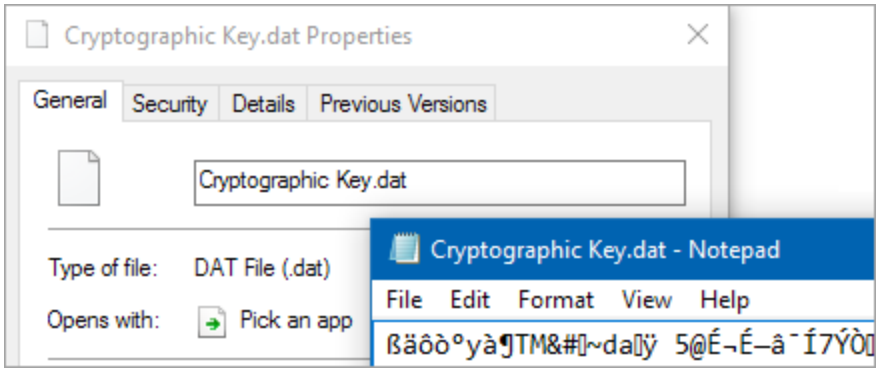
Encrypted attributes require a password to view or edit the values of the attribute. Before creating an encrypted attribute, create a cryptographic key.

Cryptographic Key

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

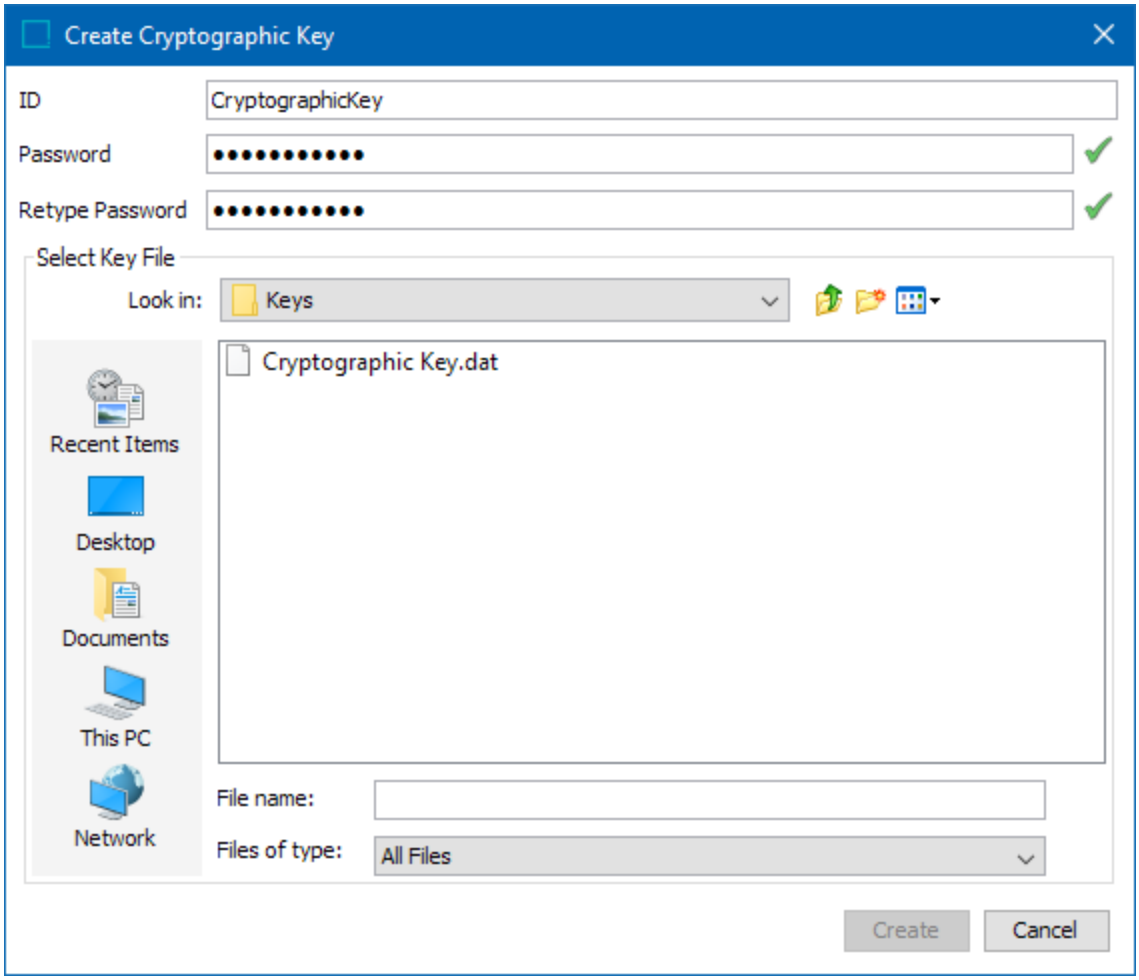


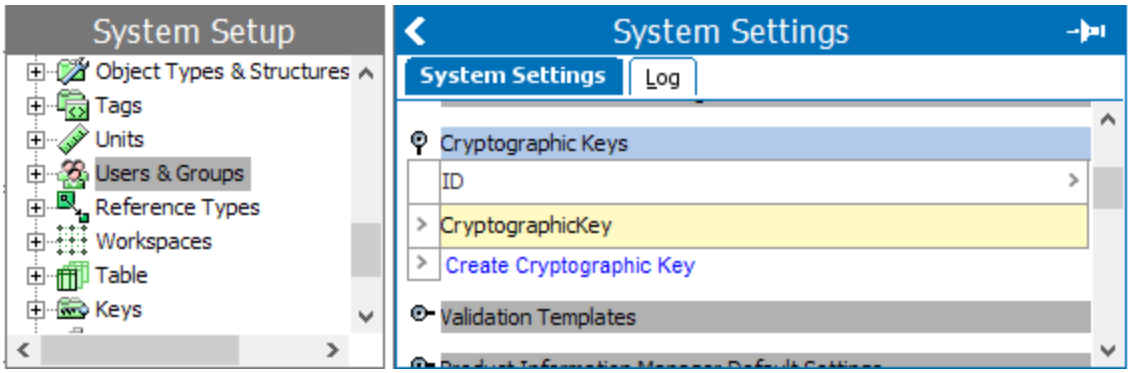
The attribute encryption system implemented in STEP is based on AES 256 bit encryption. It is a symmetric key algorithm which means that the same key is used for encryption and decryption. The system can hold multiple keys, and each key can have one unique password. This password then will be shared 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.

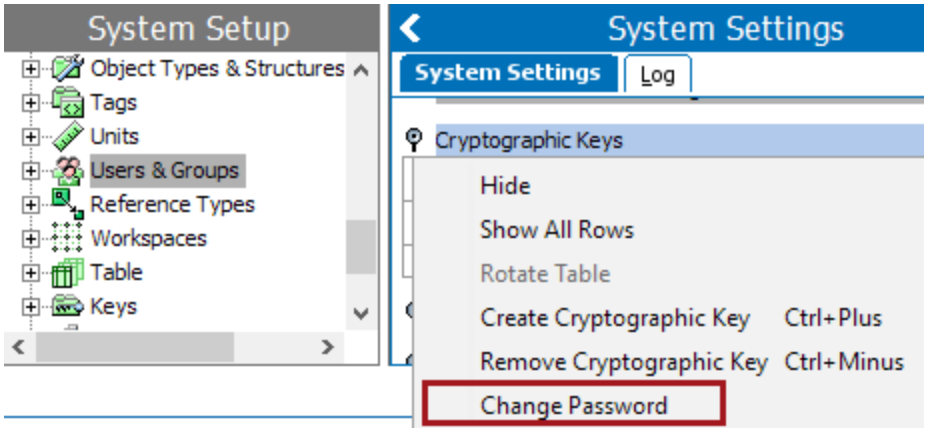
Note: Cryptographic keys are stored in STEP protected by a password that is neither the original key, nor the original password.





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:

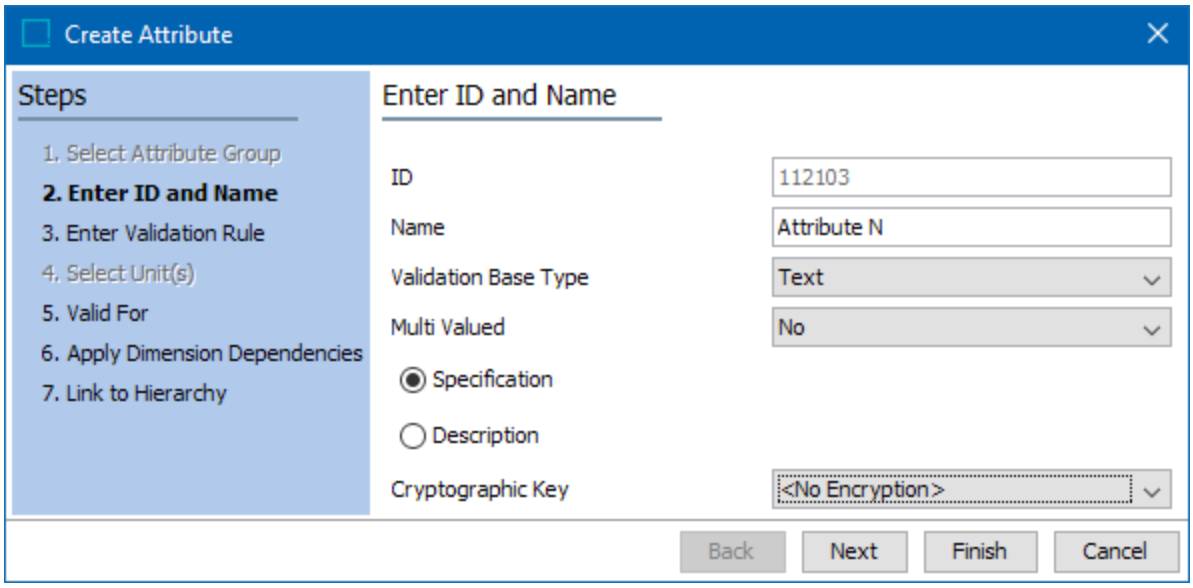
- Lower case characters a-z
- Upper case characters A-Z
- Digits 0-9
- Symbols

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 ad 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.



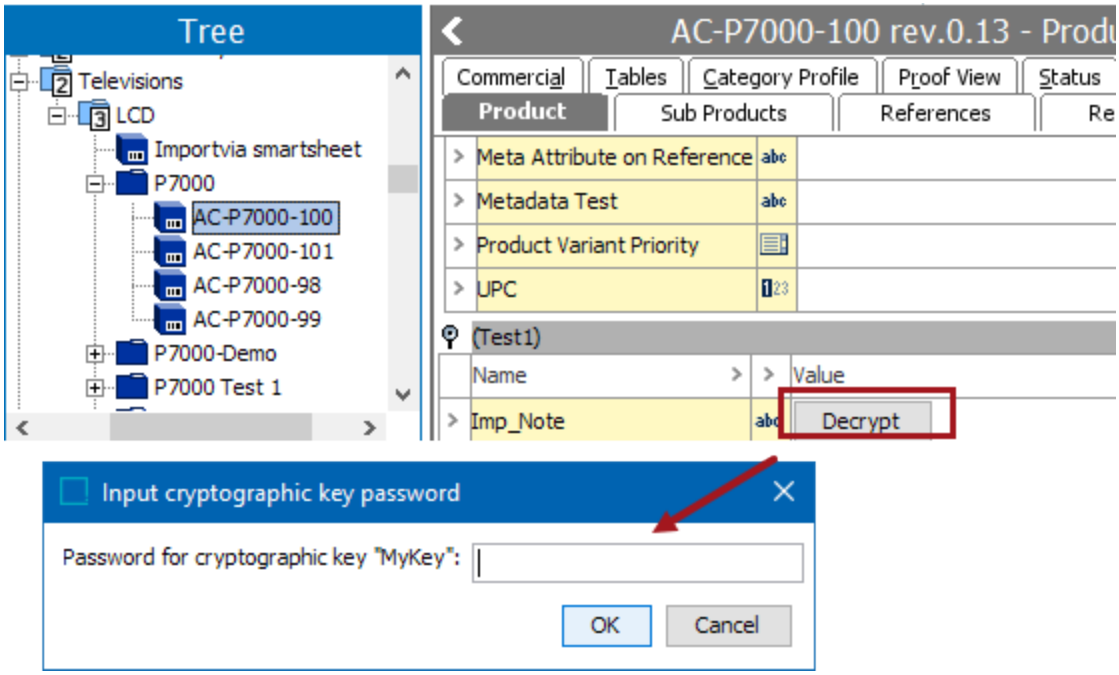
Note: The cryptographic key cannot be changed or set after creation.

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 Setup Log (available via workbench).

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.

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 **System Setup / Super User Guide** 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.

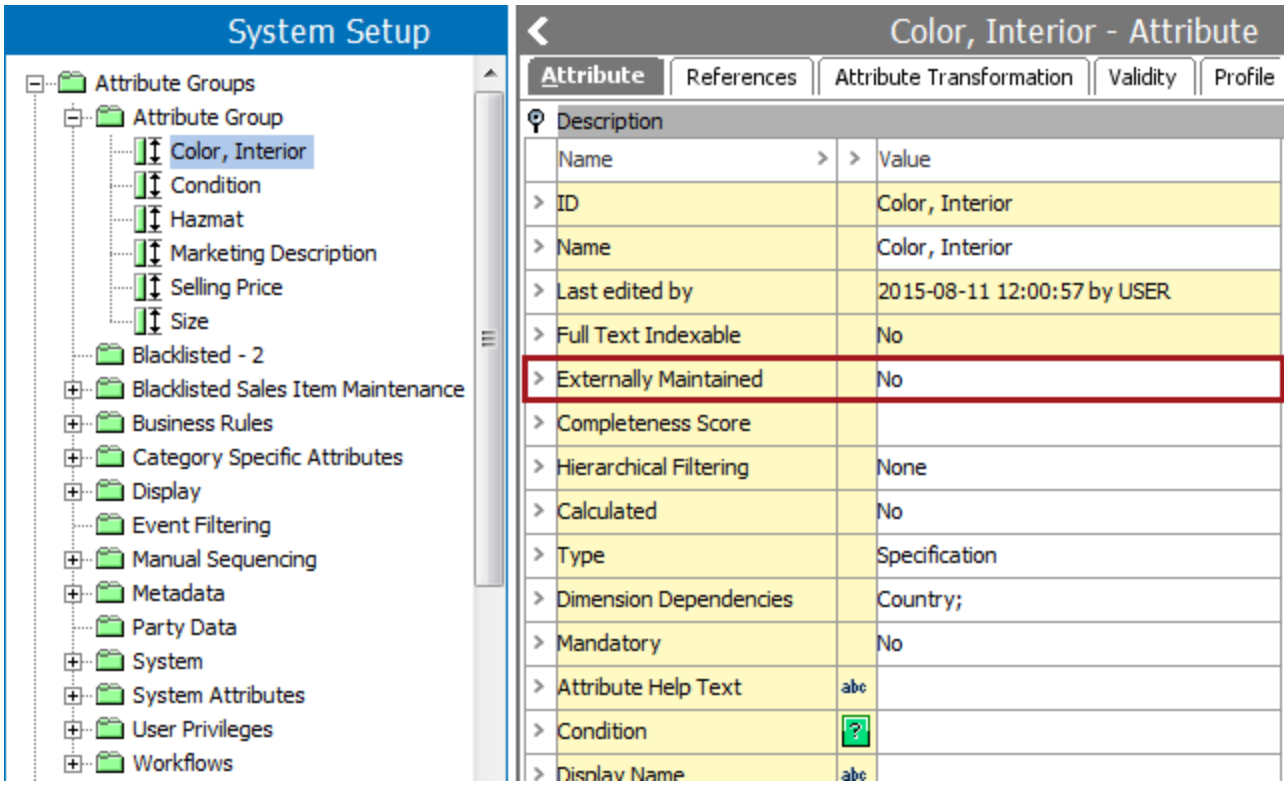
Sharing LOVs

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

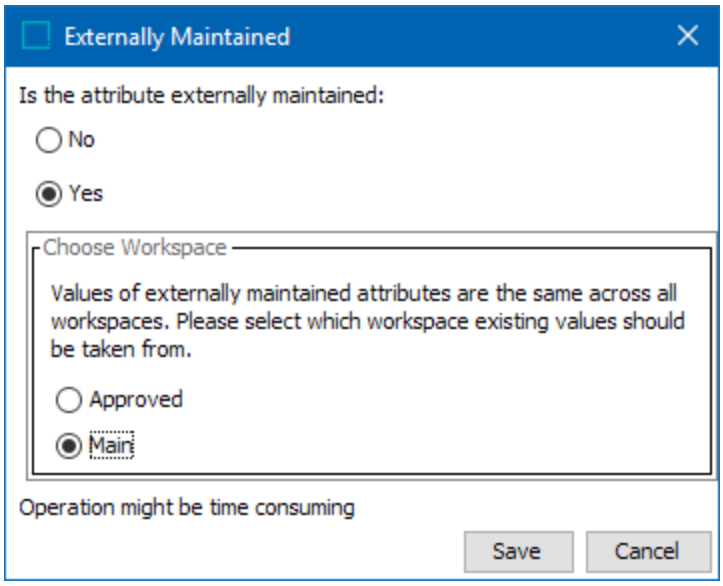
- Setting as externally maintained: If an LOV that is used by any internally maintained attributes is selected, and the attribute is then set as externally maintained, all attributes will become externally maintained.
- Setting as internally maintained: If an LOV that is used by one or more externally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



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



3. For 'Is the attribute externally maintained:' radio button, choose **Yes** to enable the Choose Workspace parameter.
4. 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

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.

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.

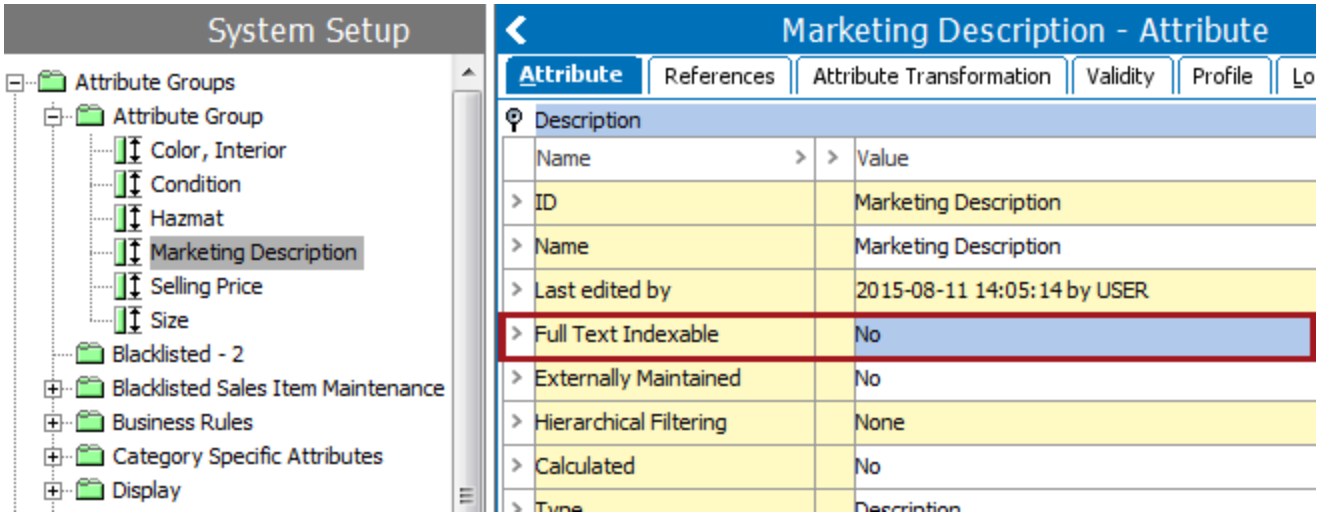
Important: Changing the Full Text Indexable parameter is time consuming, and is based on the values present for that attribute. Making an attribute full text indexable may create performance issues. **Only apply this option to attributes if absolutely necessary, and all other options have been explored.**

It is recommended to only set Full Text Indexable parameter to 'Yes" on an attribute if full text searches on values of the attribute are absolutely necessary.

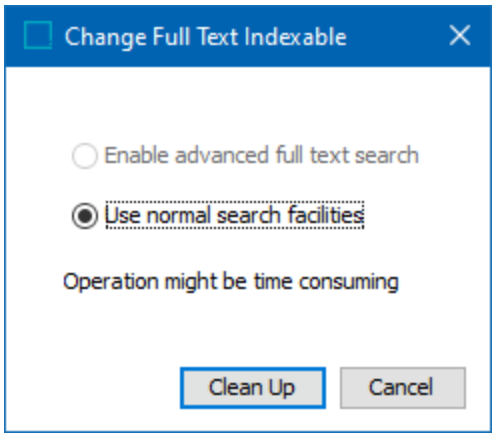
Important: Enabling the Full Text Indexable parameter can have a negative system-wide impact on performance because it is time-intensive to read, write, and search for such values.

To modify the setting:

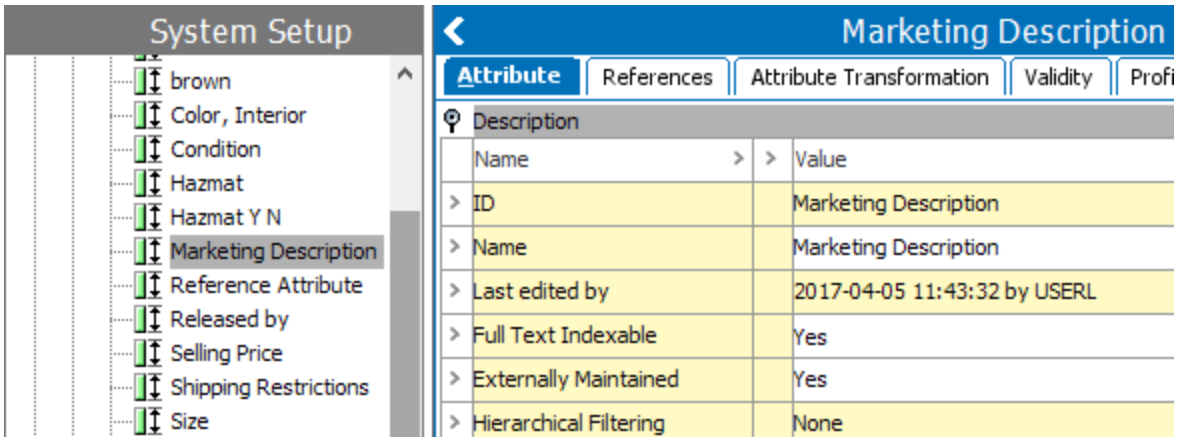
1. In System Setup, expand **Attribute Groups**, expand the relevant Group, and then select the Attribute to be set as full text indexable.



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

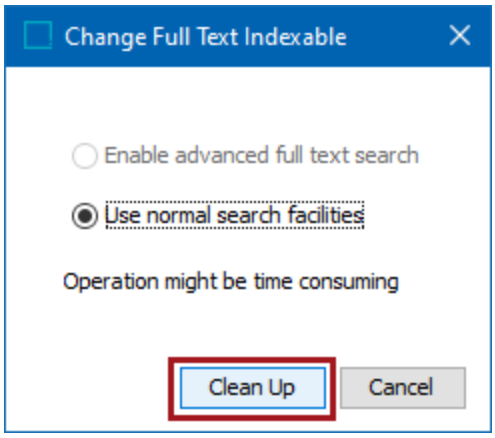


4. Click **Enable advanced full text search** and then **Save** to display Full Text Indexable = Yes.



Important: If the **Enable advanced full text search** option is disabled, you must contact Stibo Systems' Technical Support via the customer portal (Jira) to enable the full text index functionality. This is not standard functionality, so the Technical Support team will go over your options for running a SQL script (either by Stibo Systems or by the customer), which will activate 'Enable advanced full text search' option. In the background these attribute values will be indexed in value7 of the database, helping users to search by the respective attribute values. See the 'Important' message at the beginning of this topic for potential issues you may encounter upon enabling, and why it is recommended that you explore other options prior to enabling.

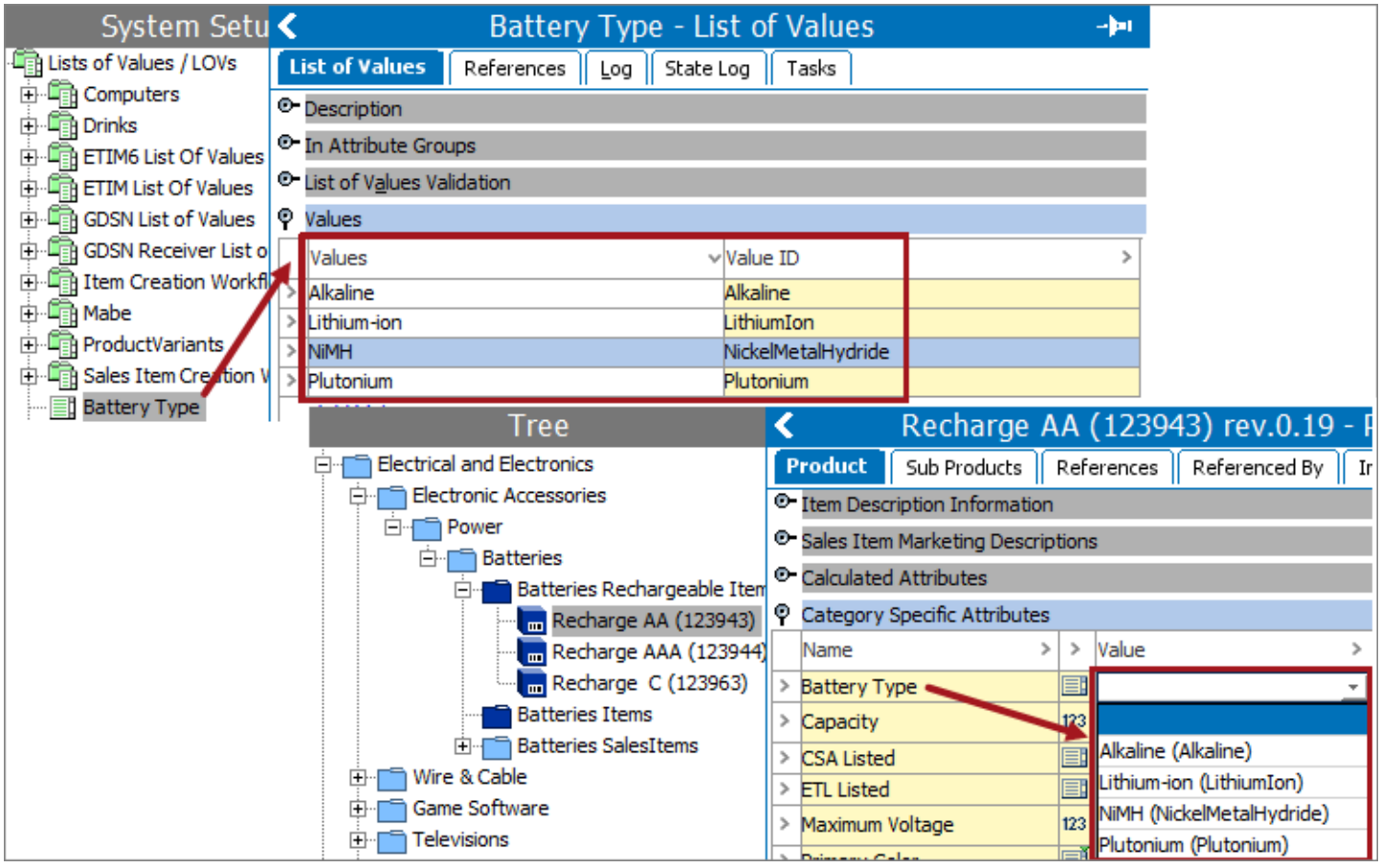
To remove the Full text Indexable = 'Yes' setting, select the field and choose 'Use normal search facilities' and click the 'Save' button. If this operation fails, choose it again, and click the 'Clean Up' button.


Change Full Text Indexable
Enable advanced full text search
Use normal search facilities
Operation might be time consuming
Clean Up
Cancel

List of Values (LOVs)

A List of Values (LOVs) is a defined 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. As described below, an LOV is only used after it has been selected on an attribute.

- LOV attributes reference a separate 'List of Values' (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 a user to add new values (a hard domain LOV).
- LOVs are 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.



Ideally, each LOV value is unique. Unique values prevent duplicate data and the confusion about the "correct" selection. 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 an list of values must be selected.
- A single LOV can be used by multiple attributes. For example, a single list of colors could hold all possible colors for your products.
- When used for multiple attributes, a filter can be applied so only a subset of the values is selectable for the attribute. For example, a user would be limited to only the colors that are valid for a specific product, and not the entire list of colors.
- If the attribute is a specification attribute it is possible to filter legal values on the links from the instantiated hierarchy so only specific values are legal for specific branches. For example, the colors available in the clothing line of products may not be the same as those in the tools line.

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.

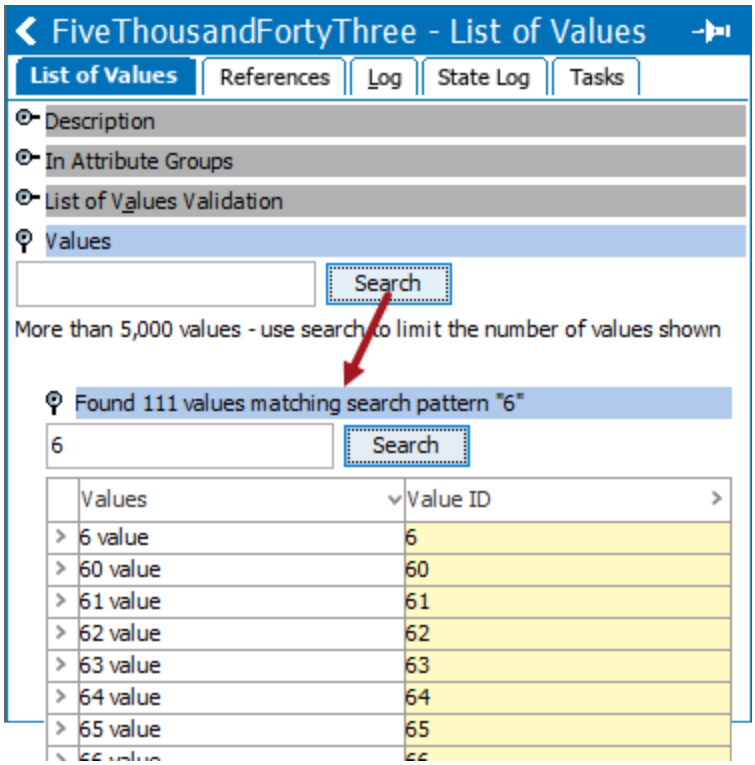
Validation Base Type

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. To find a specific LOV value, type a few characters to search for a partial or complete match.

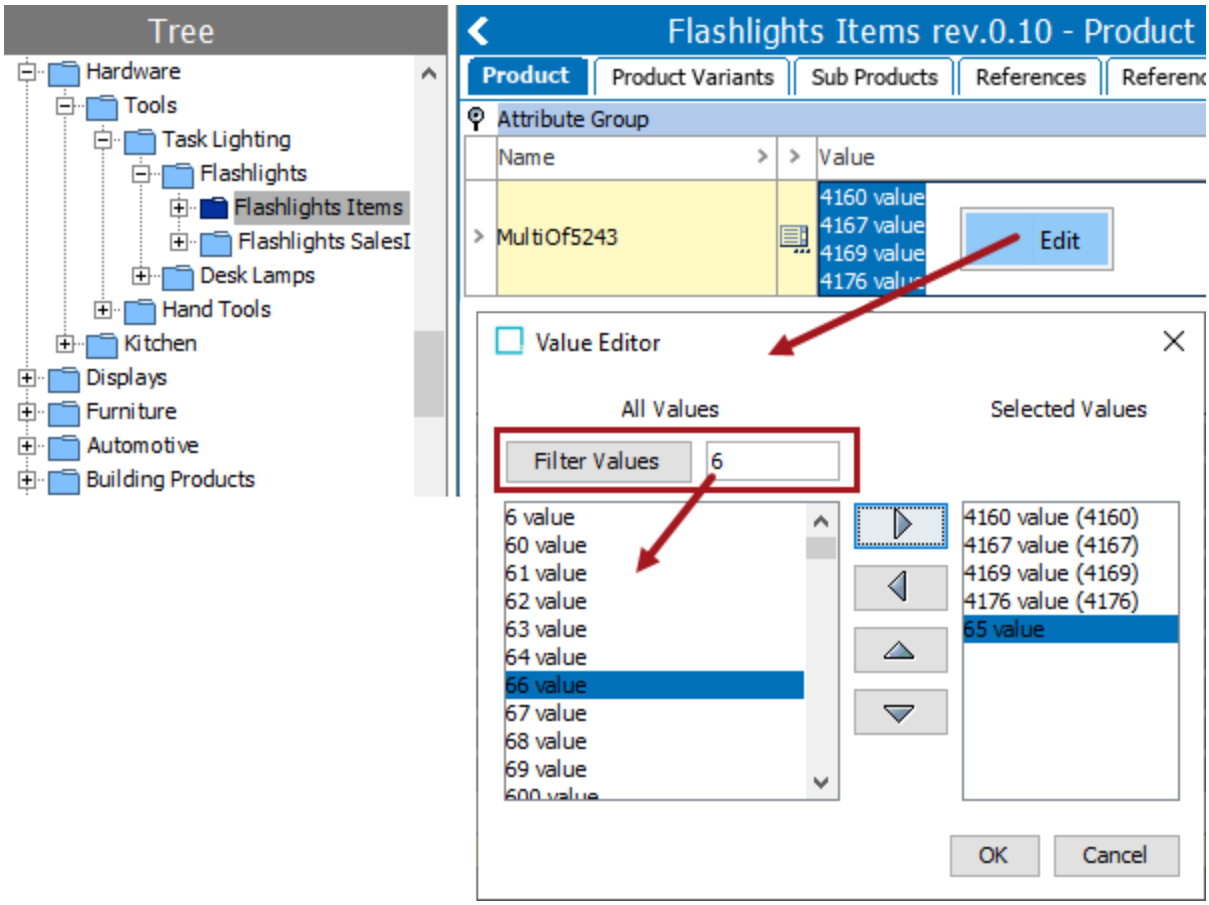
In System Setup, on the List of Values editor tab, open the Values flipper. Enter characters in the text box and click Search to display the results. The flipper label changes to show the values that match the search, and the values are displayed below the search field.



In Tree, when editing an attribute that uses a large list of values, click into the value field, right-click to display and click the Edit option (or click Alt+F2). On the Value Editor dialog:

- Enter a character in the text box to limit the listed available values. Click the Filter Values button to update the available values list.
- Double click a value in the left column to move it to the Selected Values list, or select multiple values and click the right arrow button ( ).
- Double click a value in the Selected Values column to return it to the available values list and remove it from the attribute. You can also select an item in the Selected Values list and click the left arrow to remove an item from the list ( ).
- Use the up ( ) and down ( ) arrow buttons to change the order of the highlighted value in the Selected Values list.
- Click OK to update the values in the product editor.

The values shown in the Selected Values list will be displayed in the attribute editor.

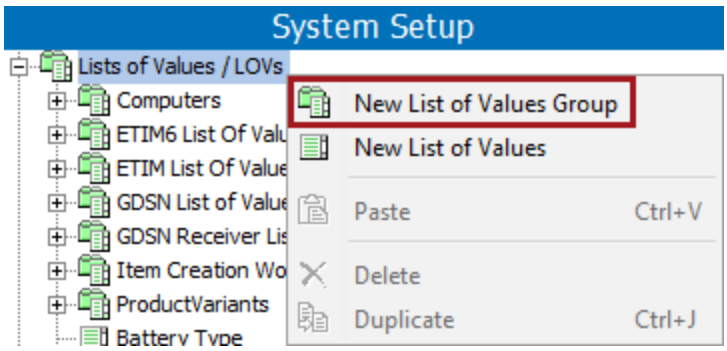

Tree
Hardware
Tools
Task Lighting
Flashlights
Flashlights Items
Flashlights SalesI
Desk Lamps
Hand Tools
Kitchen
Displays
Furniture
Automotive
Building Products
Flashlights Items rev.0.10 - Product
Product
Product Variants
Sub Products
References
Attribute Group
Name
Value
MultiOf5243
4160 value
4167 value
4169 value
Edit
Value Editor
All Values
Selected Values
Filter Values
6
6 value
60 value
61 value
62 value
63 value
64 value
66 value
67 value
68 value
69 value
4160 value (4160)
4167 value (4167)
4169 value (4169)
4176 value (4176)
65 value
OK
Cancel

Creating an LOV Group

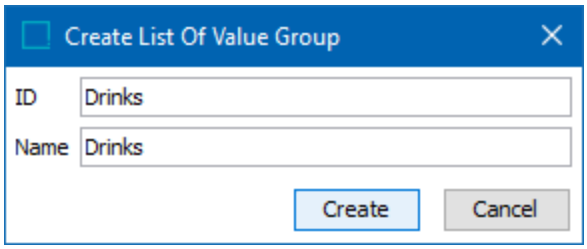
A Lists of Values / LOV group can be used to collect individual, but related, LOVs. The LOV group can be linked to an attribute group, and used in combination with privileges to restrict users in maintaining certain lists of values.

Below are steps to create a new lists of values group.

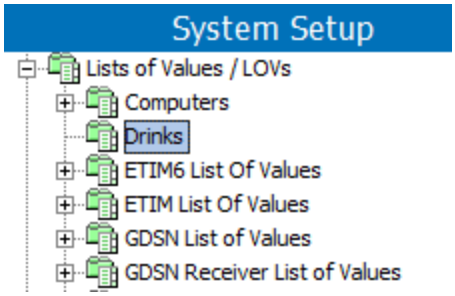
1. Go to System Setup > Lists of Values / LOVs, right-click **List of Values /LOVs**, and click the **New List of Values Group** option.



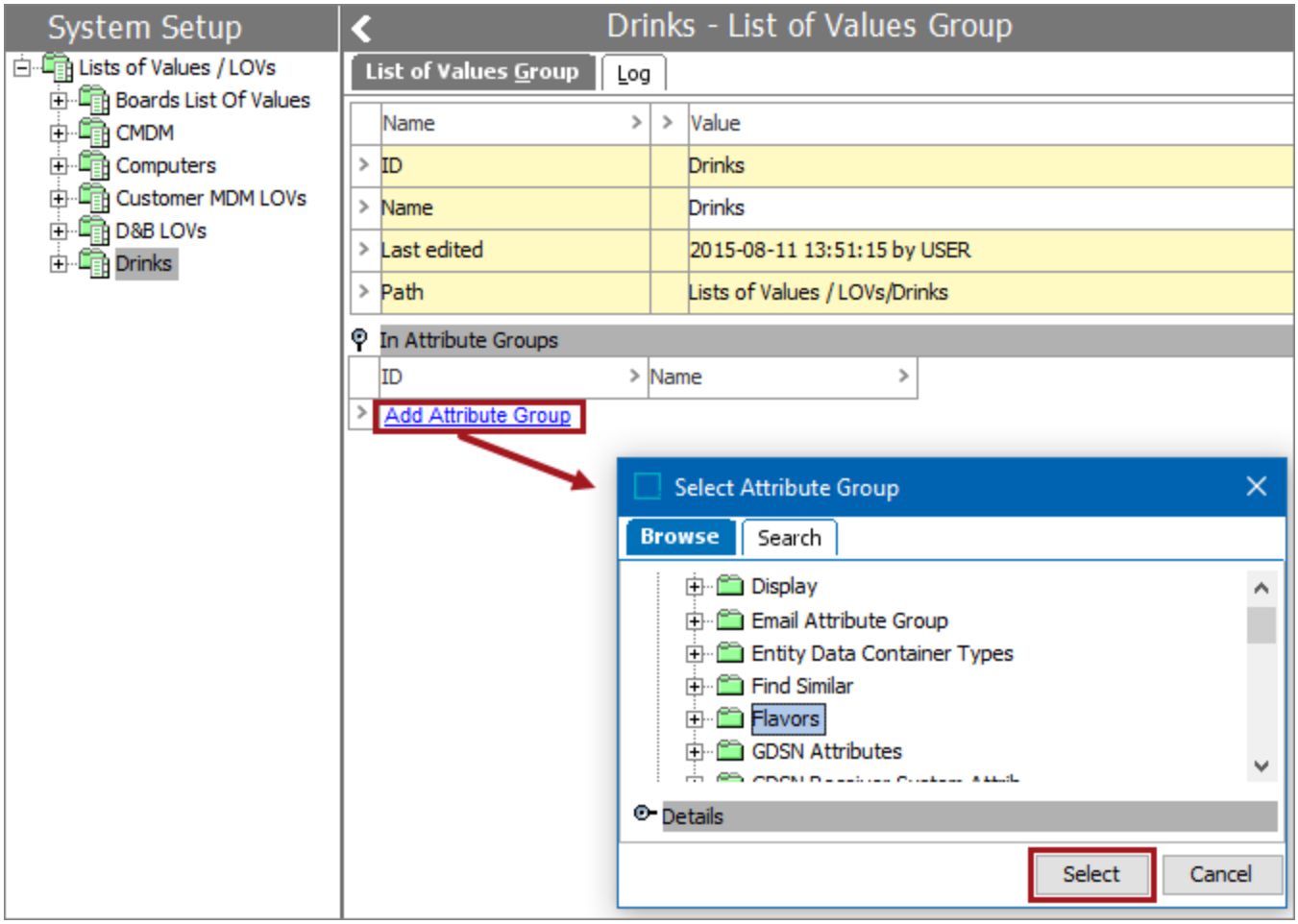
2. Enter an **ID** and **Name**, then click the **Create** button.



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 LOV 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.



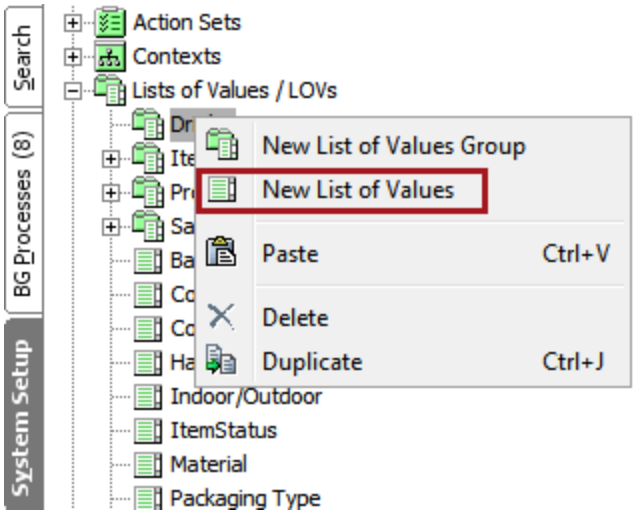
Creating an LOV

Creating an LOV (List of Values) involves:

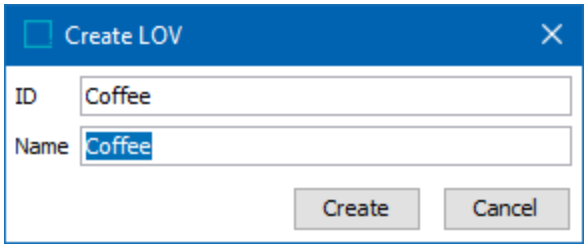
- Setting validation rules
- Selecting valid units
- Selecting dimension dependencies
- Adding values

Below are the steps to create an LOV.

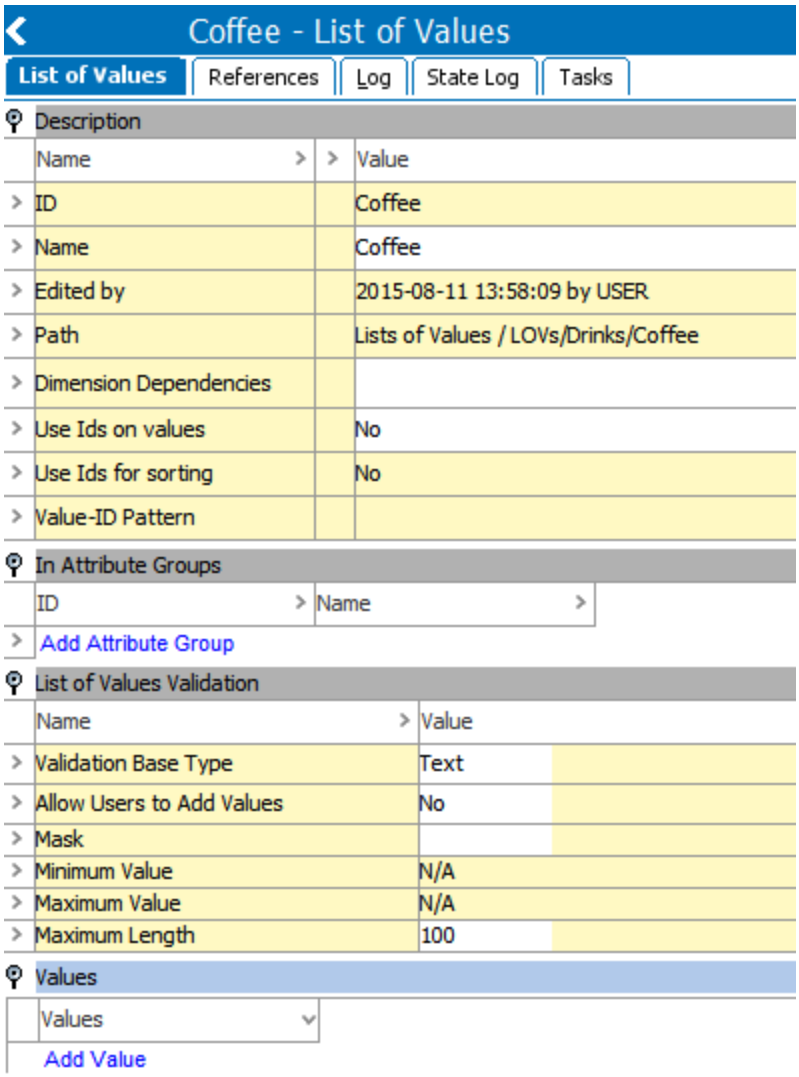
1. Go to System Setup > Lists of Values / LOVs, right-click the List of Values / LOVs folder (or a desired LOVs group folder where you want to group the new LOV), and click **New List of Values**.



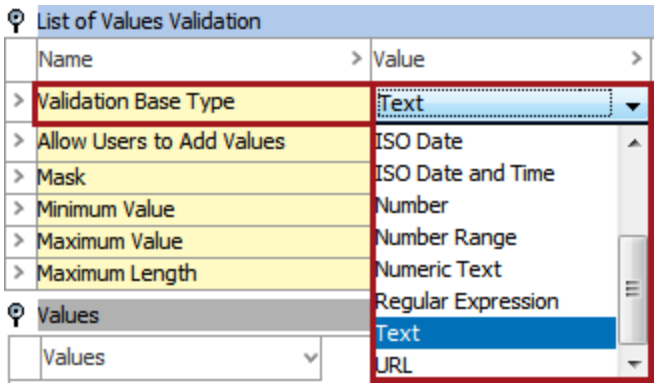
2. The Create LOV dialog will display. Enter an **ID** and a **Name**.



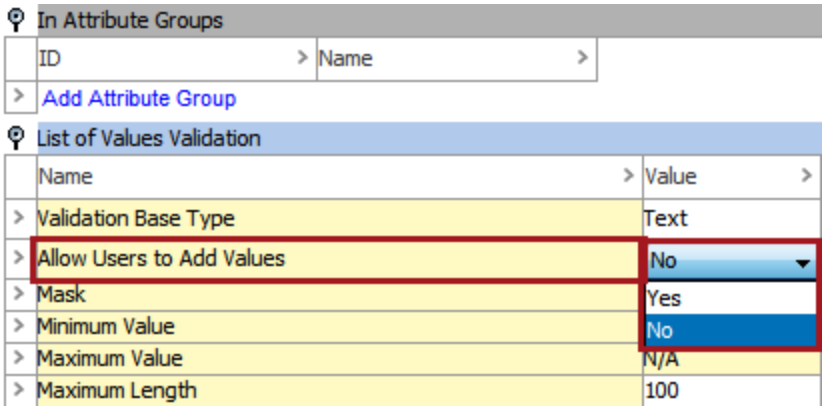
3. Click the **Create** button, and the List of Values editor will display (as shown below).



4. 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.



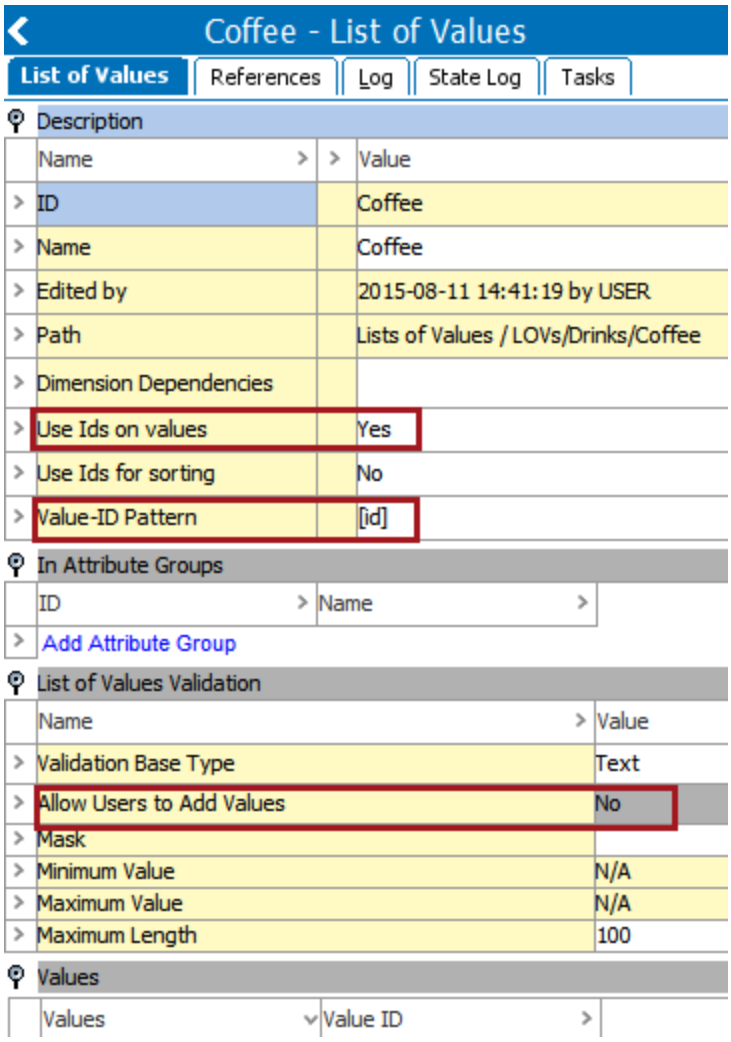
5. Within the **List of Values Validation** flipper, click the **Allow Users to Add Values** parameter field to select either **Yes** or **No** from the dropdown menu (as shown below).



- **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.

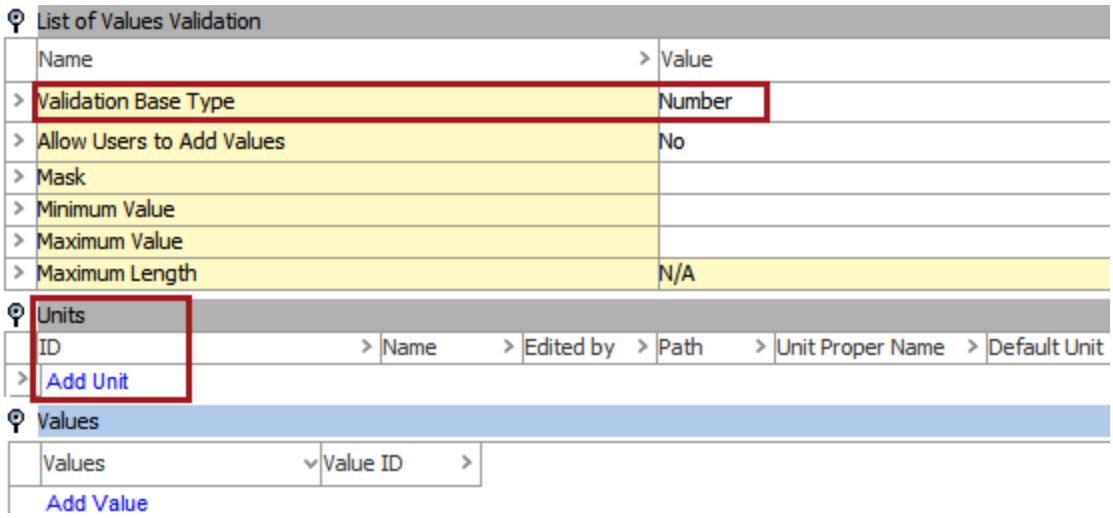
6. **Use IDs on values:** Optionally allows the use on an ID for each value within the LOV. Can be useful when importing object data and applying a certain value with a specific ID.

- 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.



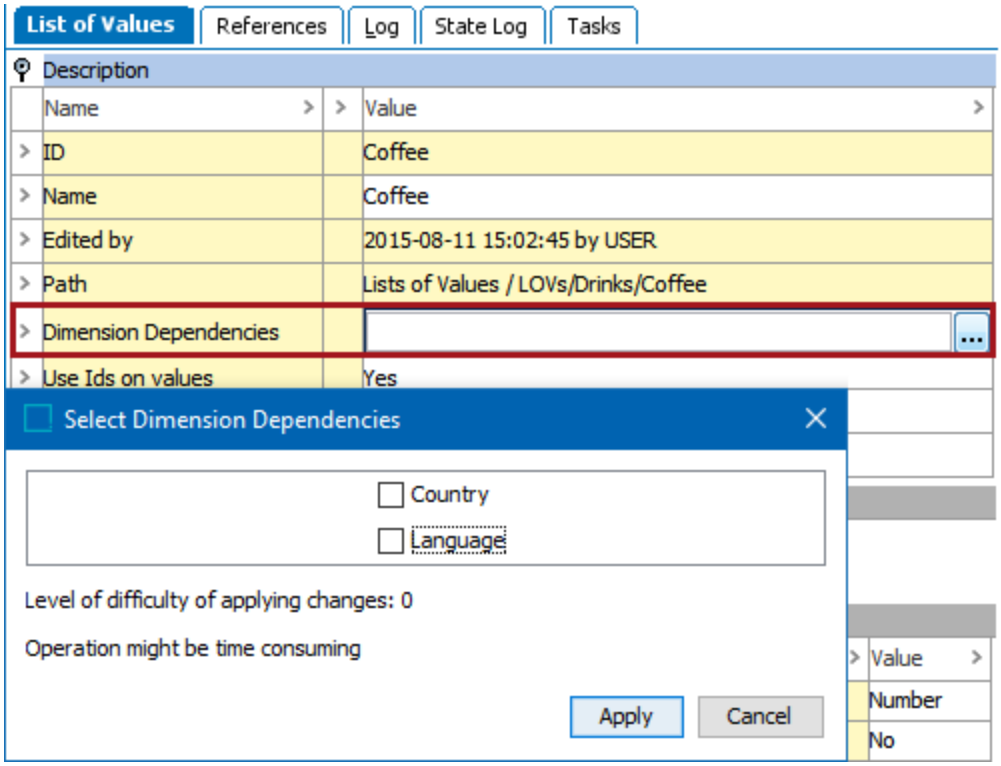
Note: If you set **Use Ids on Values** to **Yes** on an LOV that already contains values, a dialog displays with options to generate IDs on existing values. For more information, see the **Adding IDs to Values in LOV** topic within this section.

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.

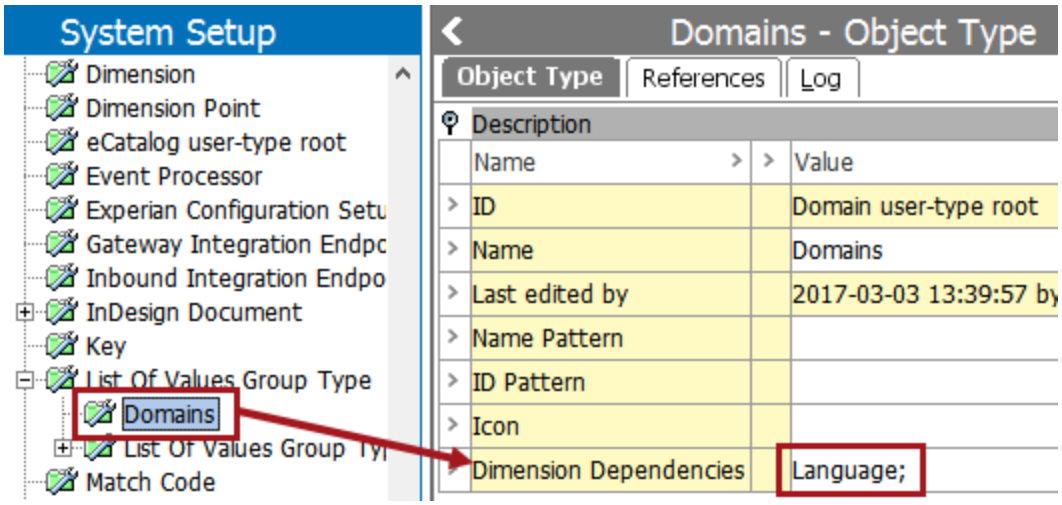


8. **Dimension Dependencies**: Optionally allows for the use of more than one dimension. To enable, click the ellipsis button (...), and select one or more dimensions.

   - 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.



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 at: System Setup > Object Types & Structures > Basic Object Types > List Of Values Group Type > **Domains**.



9. Add all required values to the LOV by following the steps in the **Adding Values to an LOV** topic within this guide.
10. Assign the LOV to the relevant attributes to allow selection of the values.

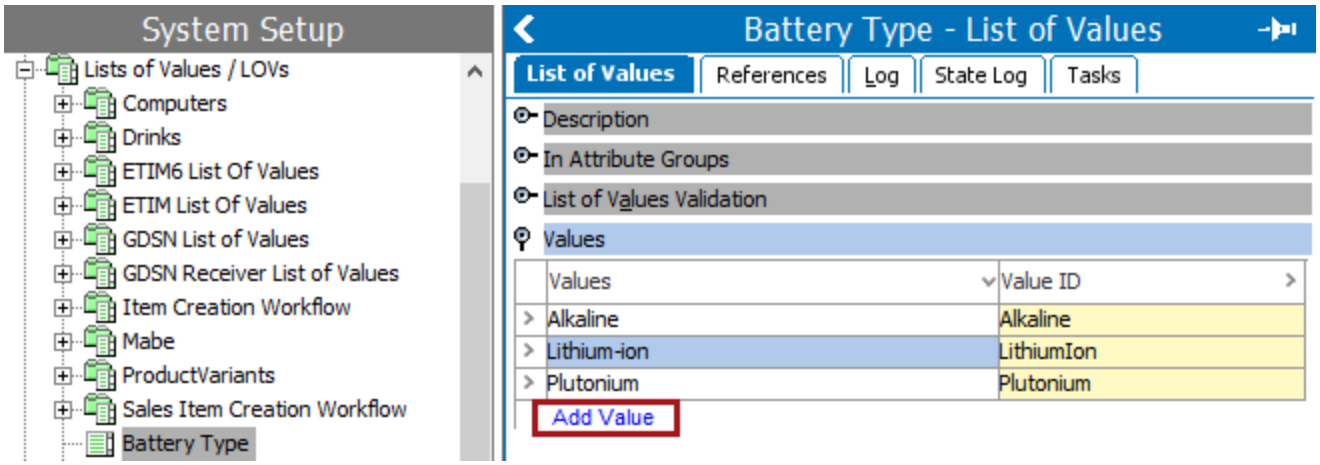
Adding Values to an LOV

Adding new values to an LOV can be done one-at-a-time or in bulk. However, with the necessary privileges, values can always be added to the LOV from the System Setup tab. This topic addresses adding a single LOV value, and adding a group of LOV values.

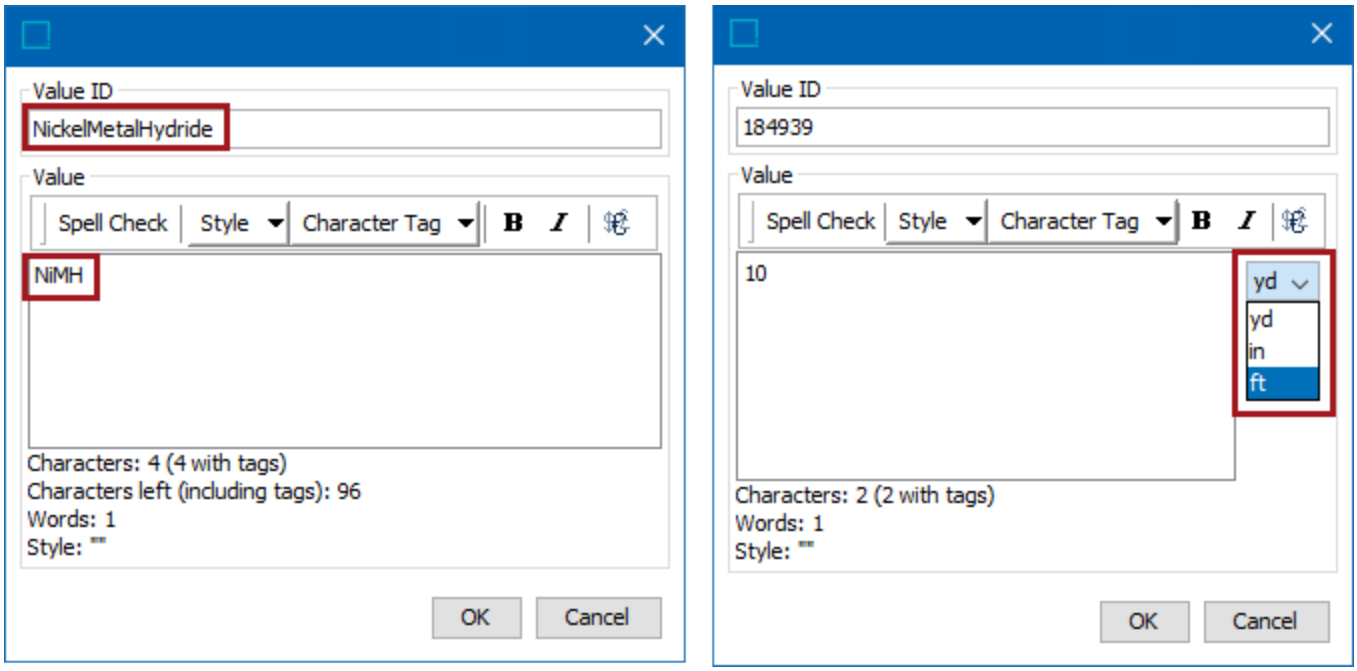
Adding a Single LOV Value

When only a few new values are required, it is simple to add them individually from the System Setup tab.

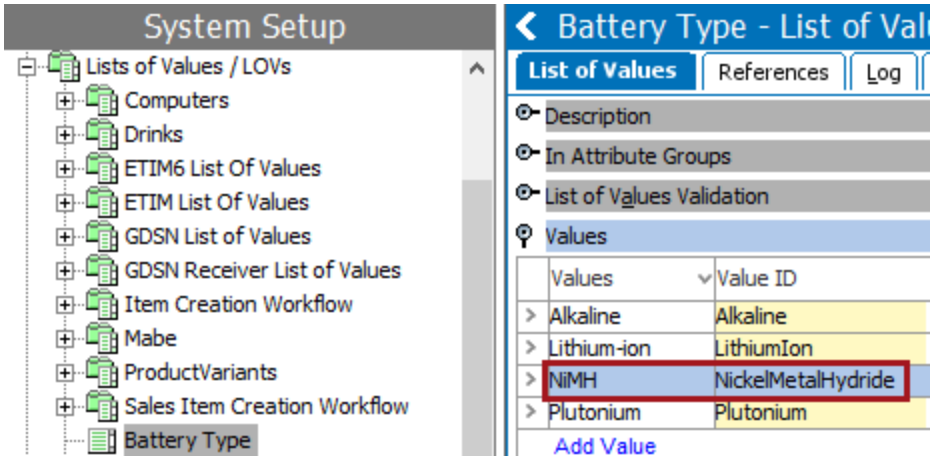
1. In System Setup, select the relevant LOV, open the Values flipper and click the **Add Value** link.



2. Enter a **Value ID** (if required use ValueID parameter) and a **Value**.
   - If units are not defined for the LOV, then 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.

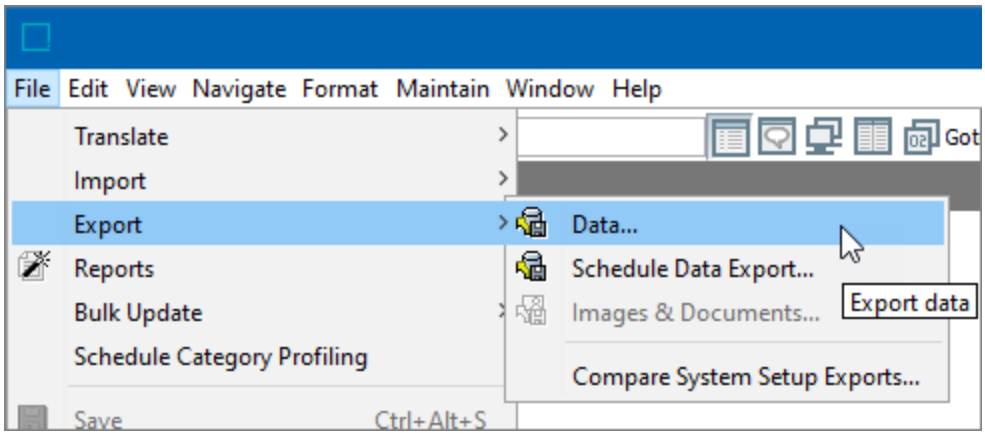


Adding a Group of LOV Values

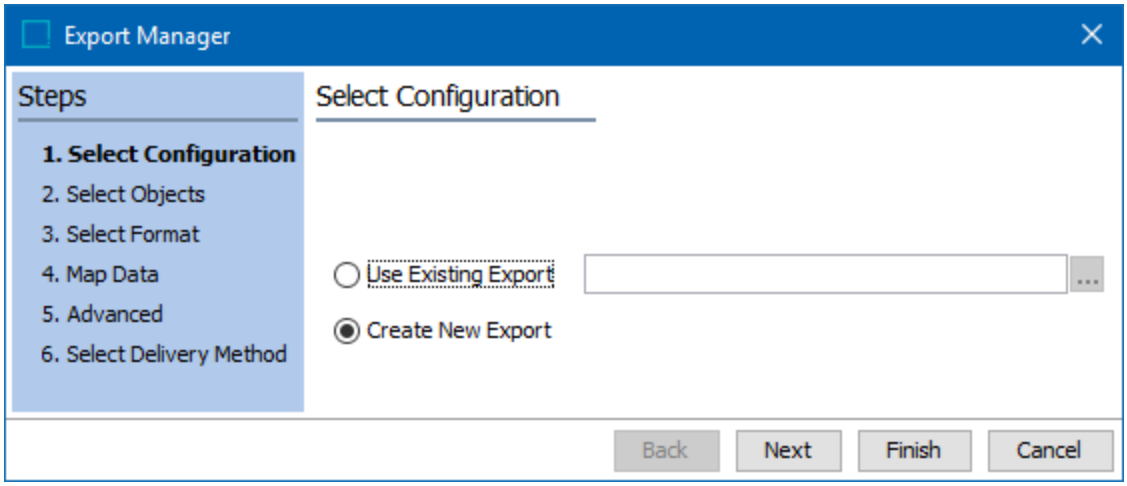
When many new values are required, it may be easier to import the values using STEPXML. This process involves exporting the existing LOV to create a STEPXML file, modifying the file to include the new values, and then importing those new values back into STEP.

Exporting the Existing LOV to Create a STEPXML File

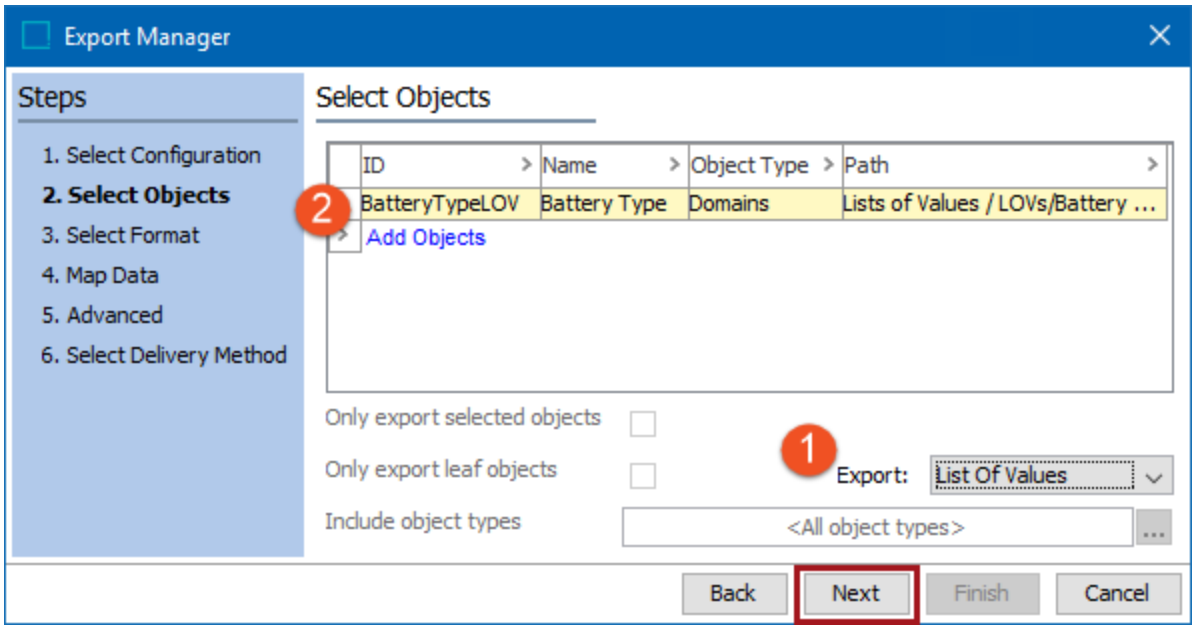
1. Go to Workbench > File > Export > Data



2. The Export Manager dialog will display, as shown below.



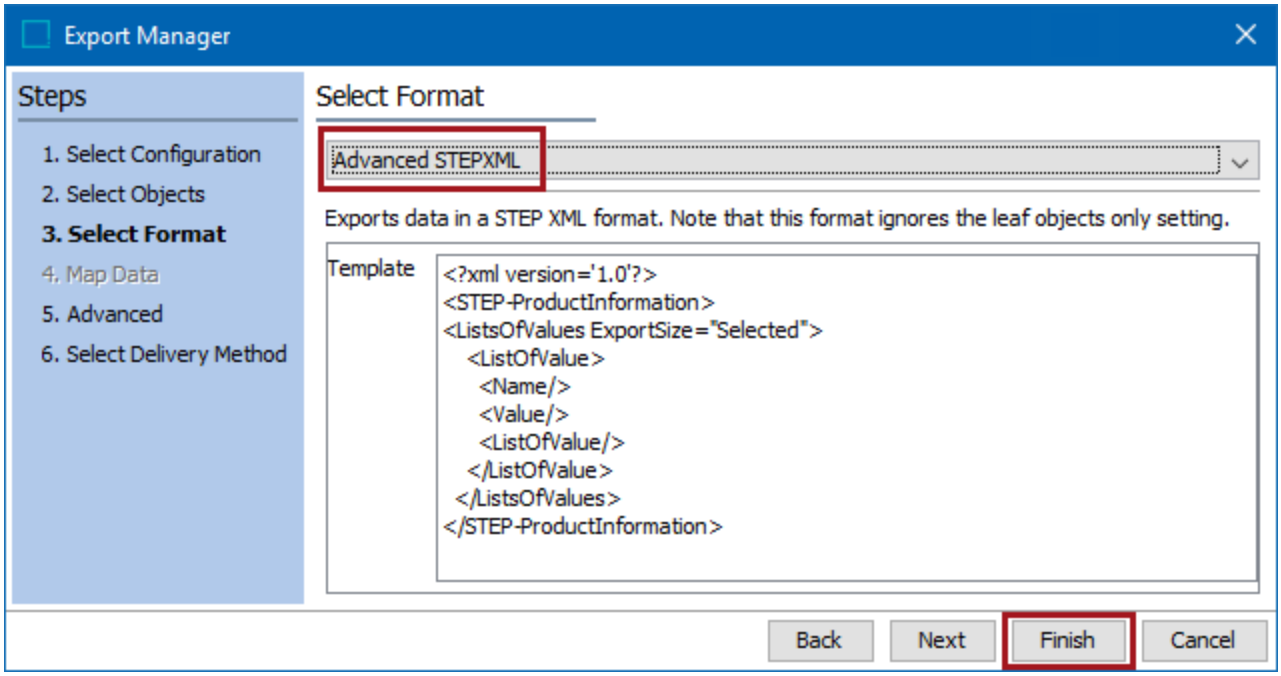
3. Click the **Next** button and Select Objects will display.



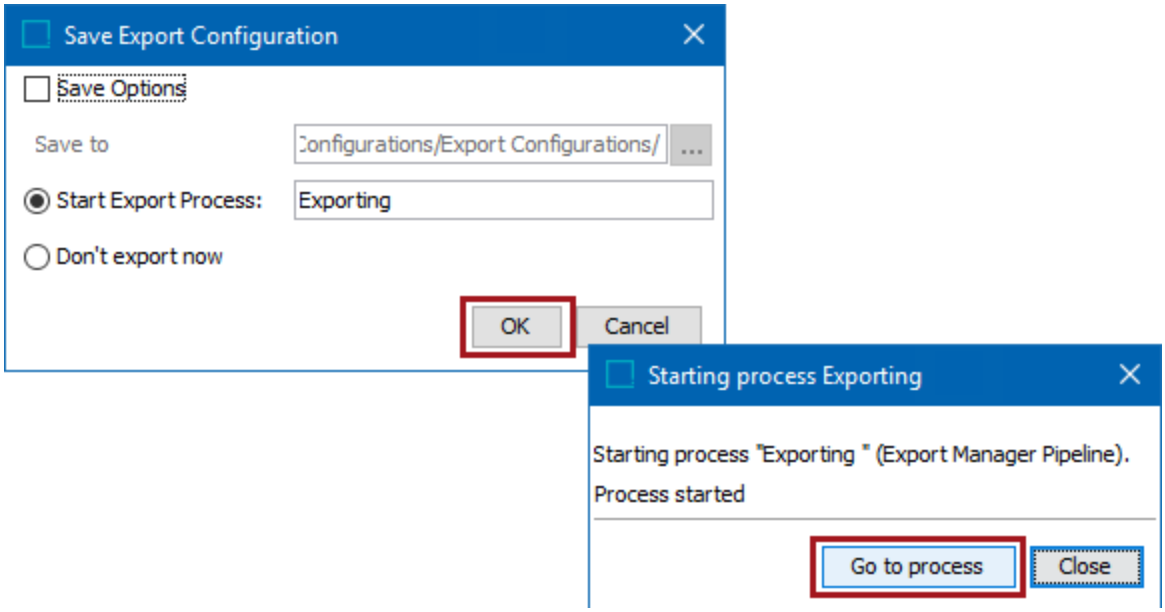
- In the Export field, select **List of Values**.
- In the objects list, use the **Add Objects** link to select the LOV (object type of Domain) that will get the new values.
- Click the **Next** button.

For details about each step and parameter of the Export Manager wizard, see the **Creating a Data Export** section of the **Data Exchange** documentation.

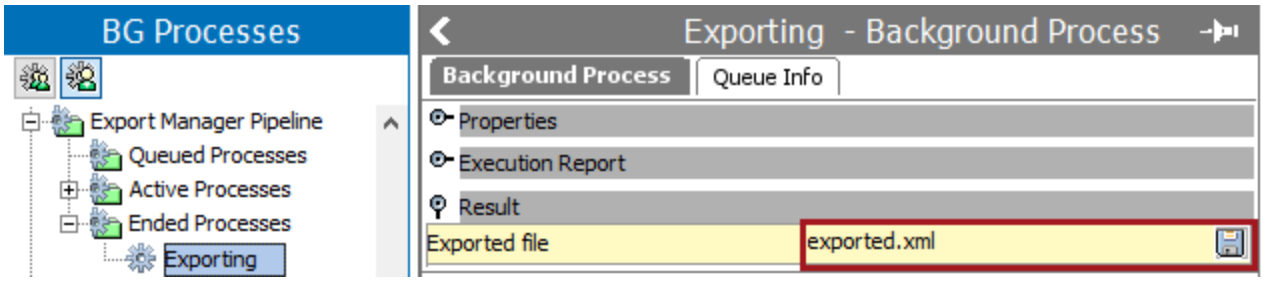
4. On Select Format, select **Advanced STEPXML** and click the **Finish** button.



5. On the Save Export Configuration dialog, click **OK** to start the export, and on the Starting process dialog, click the **Go to process** button.



6. On the displayed background process editor, open the output file.



Modifying the File to Include the New Values

1. 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>

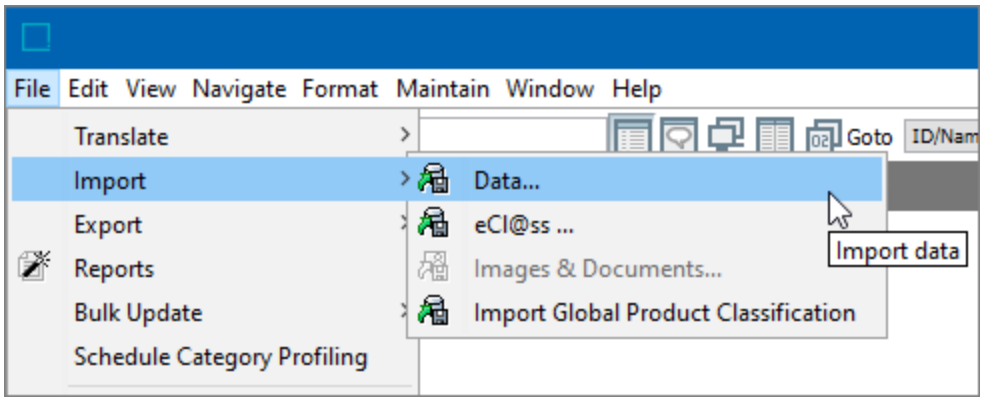
4. Copy and paste the new value data into your import file and save it.

```
<STEP-ProductInformation ExportTime="2016-09-08 16:27:37" Export

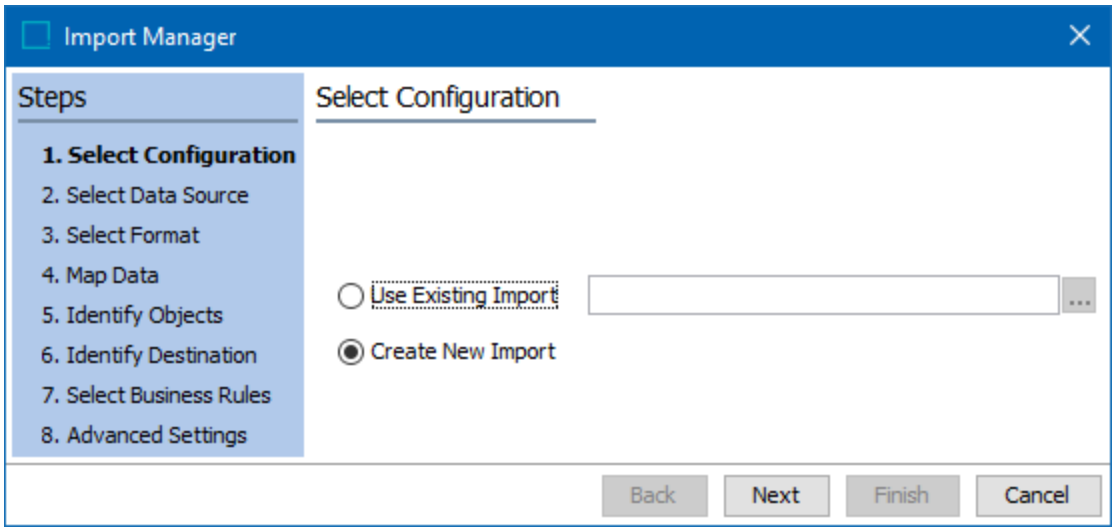
  <ListsOfValues>
    <ListOfValue ID="BatteryTypeLOV" ParentID="List Of Values gr
      <Name>Battery Type</Name>
      <Validation BaseType="text" MinValue="" MaxValue="" MaxLer
<Value ID="BatType1">Battery Type 1</Value>
<Value ID="BatType2">Battery Type 2</Value>
<Value ID="BatType3">Battery Type 3</Value>
<Value ID="BatType4">Battery Type 4</Value>
<Value ID="BatType5">Battery Type 5</Value>
<Value ID="BatType6">Battery Type 6</Value>
<Value ID="BatType7">Battery Type 7</Value>
<Value ID="BatType8">Battery Type 8</Value>
<Value ID="BatType9">Battery Type 9</Value>
<Value ID="BatType10">Battery Type 10</Value>
    </ListOfValue>
  </ListsOfValues>
</STEP-ProductInformation>
```

Importing New Values Into STEP

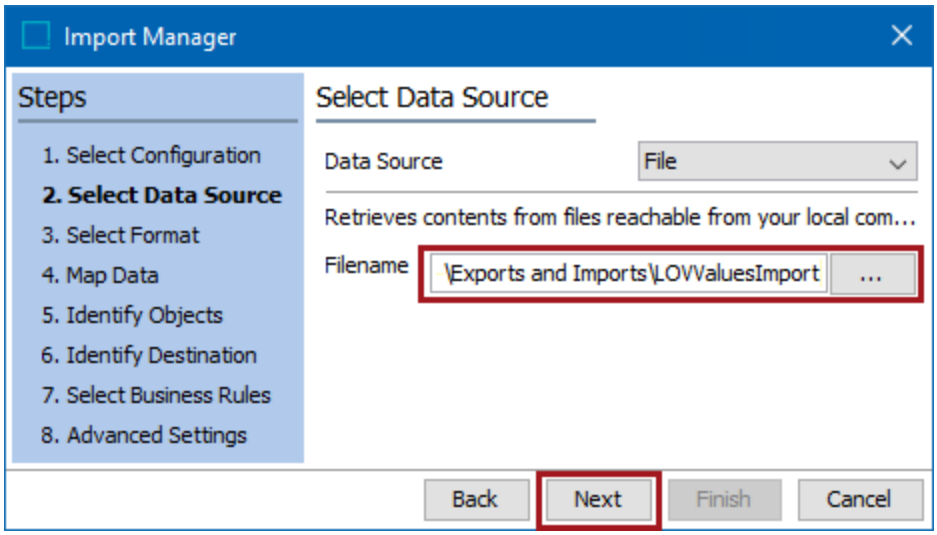
1. Go to Workbench > File > Import > Data



2. The Import Manager dialog will display, as shown below.



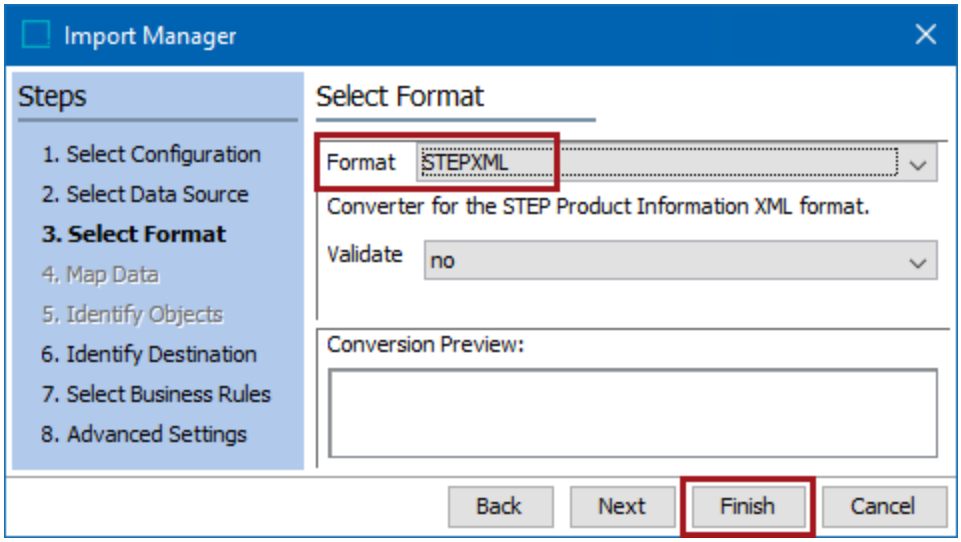
3. Click the **Next** button and Select Data Source will display.



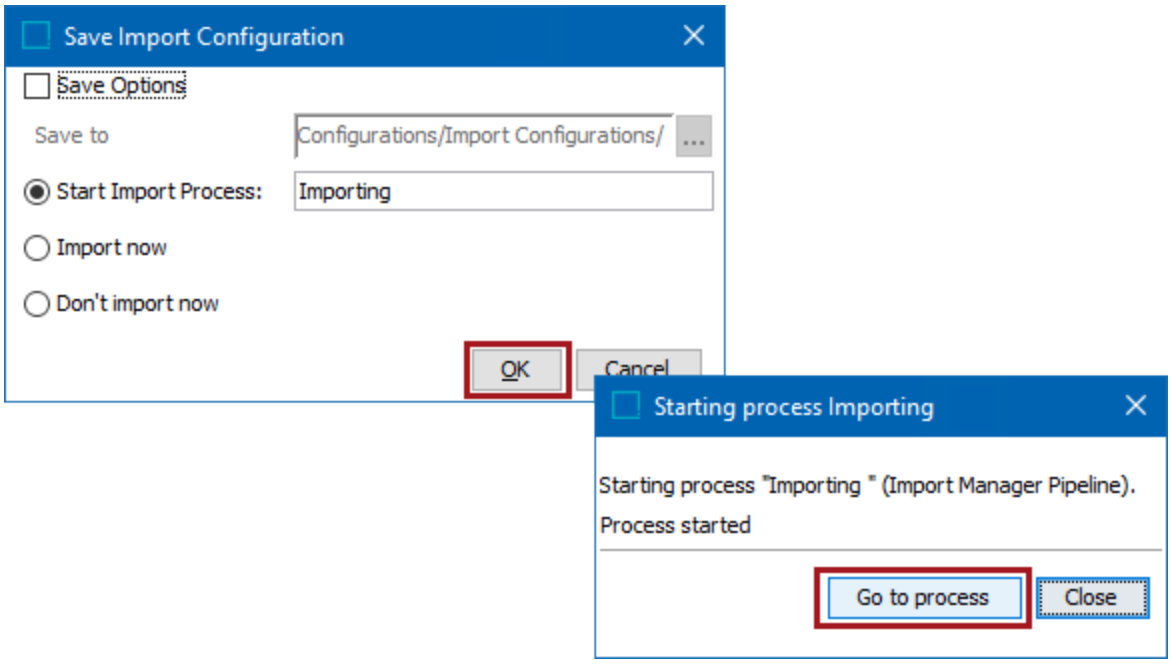
4. On Select Data Source:
   - In the Filename field, select the **import file name**.
   - Click the **Next** button.

   For details about each step and parameter of the Import Manager wizard, see the **Creating a Data Import** section of the **Data Exchange** documentation.

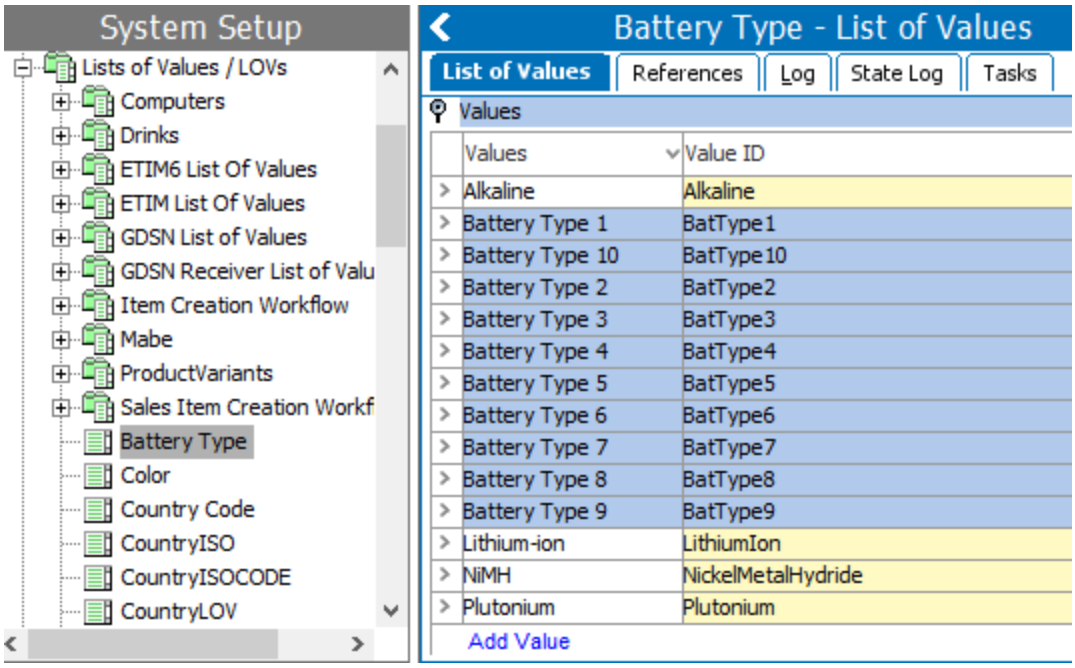
5. On Select Format, select **STEPXML** and click the **Finish** button.



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.



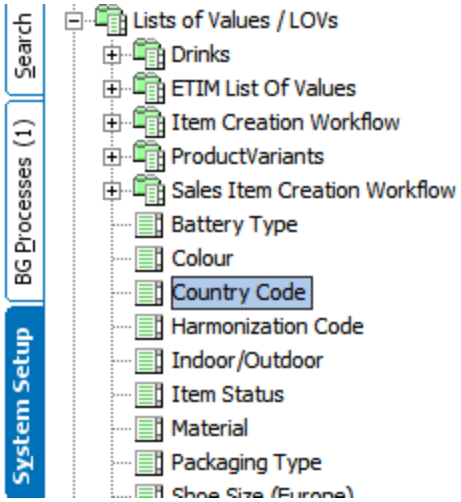
Adding IDs to Existing Values in LOV

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).

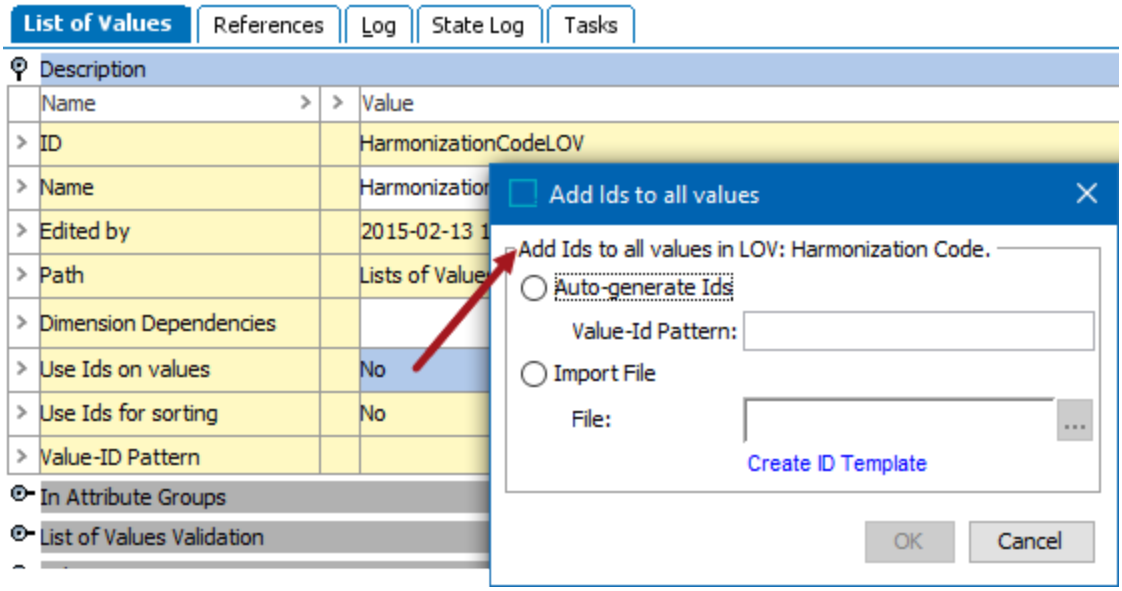
Configuring IDs for Existing LOV Values

IDs can be added to existing LOV values one of two ways: 'auto-generated IDs' or 'import files.' The following configuration is required for either option.

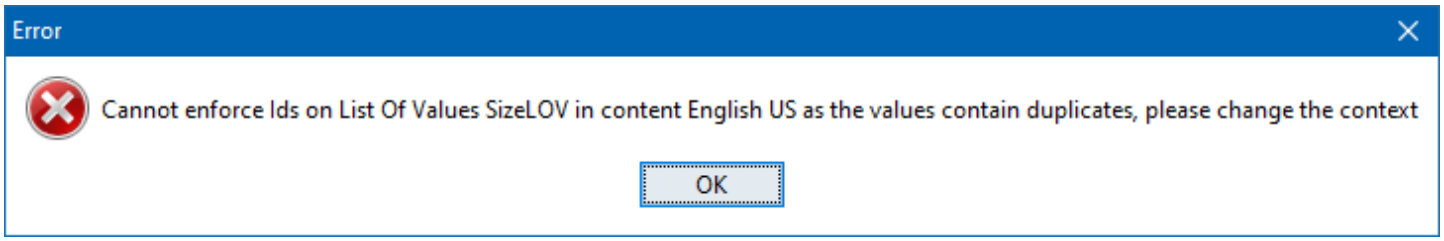
1. Go to System Setup > Lists of Values / LOVs > click on the relevant LOV.



2. On the List of Values tab, double-click the value column of the **Use Ids on values** parameter.
    - If the 'Add IDs to all values' dialog displays, select a radio button to determine how IDs are created. Each option is addressed below in the **Using the Auto-generate IDs option** section or the **Using the Import File option** section.



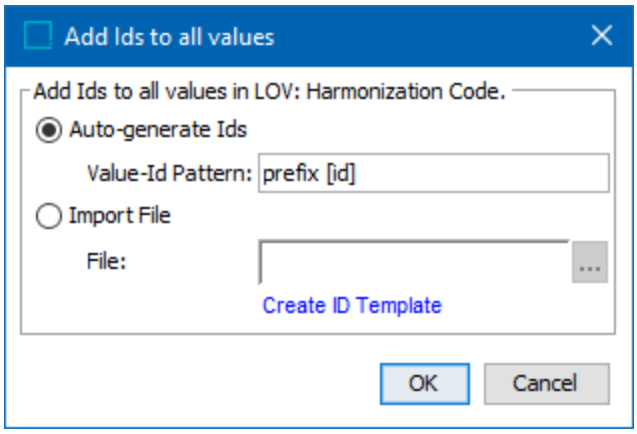
- If an error similar to the following is displayed, click **OK** to close the error and see the **Allowing Value IDs to be Added to an LOV** topic for additional required configuration.



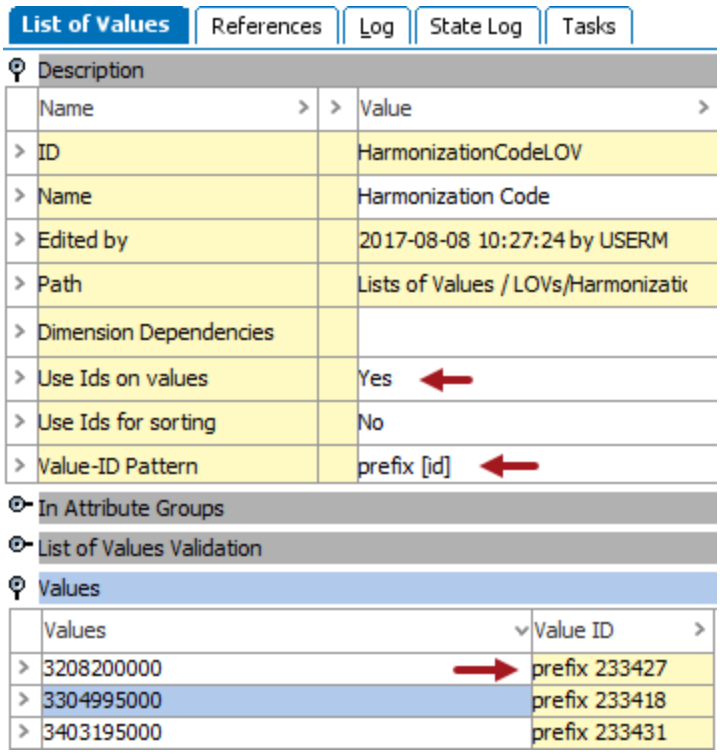
Using the Auto-generate IDs option

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 case-sensitive ID pattern '[id]' and any other additional desired value. The additional text can precede or follow the [id] as needed.



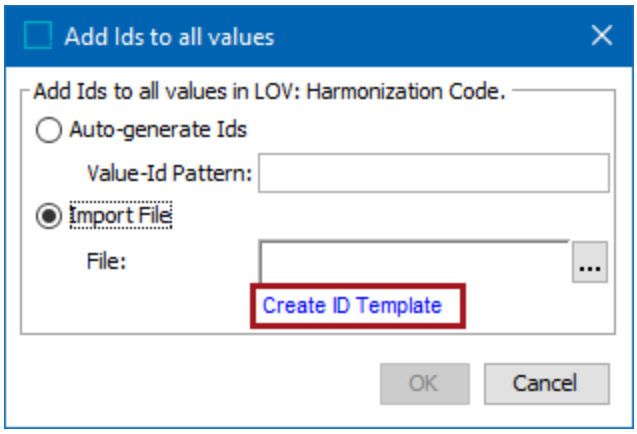
3. Click the **OK** button to apply the Value-ID Pattern on existing values. The LOV editor will display the newly applied ID added to the existing value according to the ID pattern and the 'Value ID' column will display the respective IDs for the present values within the Values flipper (as shown below).



Using the Import File option

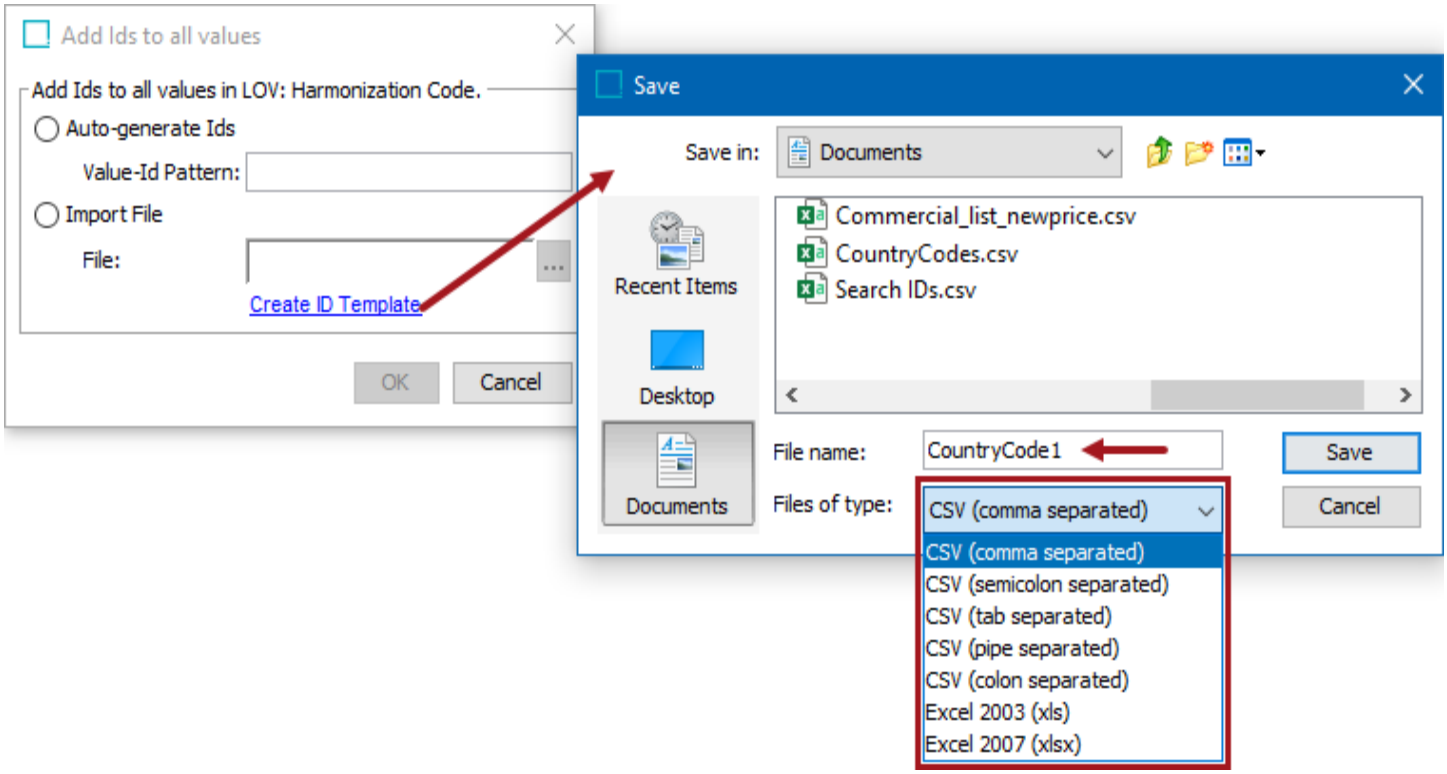
This option uses the selected import file to determine the value for each ID.

1. Select the **Import File** radio button.
2. Click the **Create ID Template** link.

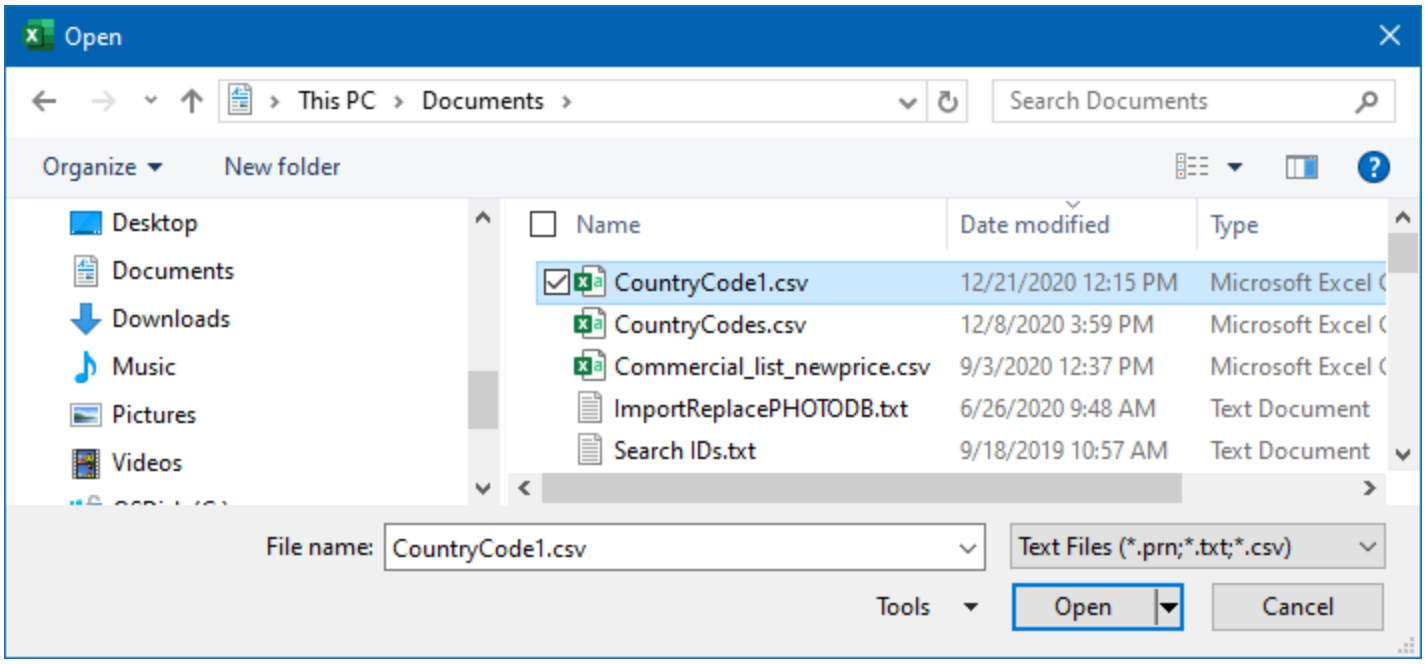


3. Choose the destination of the saved template, and name the template.

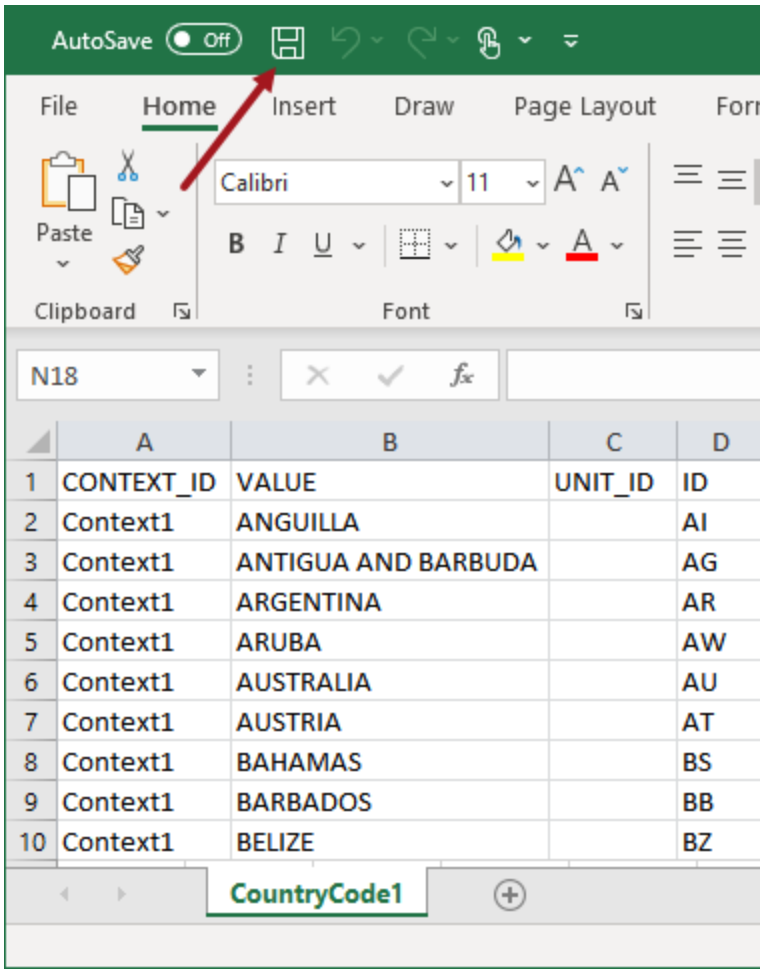
Note: An ID template can be saved in CSV (comma, semicolon, tab, pipe, or color separated) or Excel 2007 format. It is recommended to use the newest, supportable version of Excel.



4. Click **Save**, leaving the **Add IDs to all values** dialog open.
5. Open your template using to the program used to create the template.



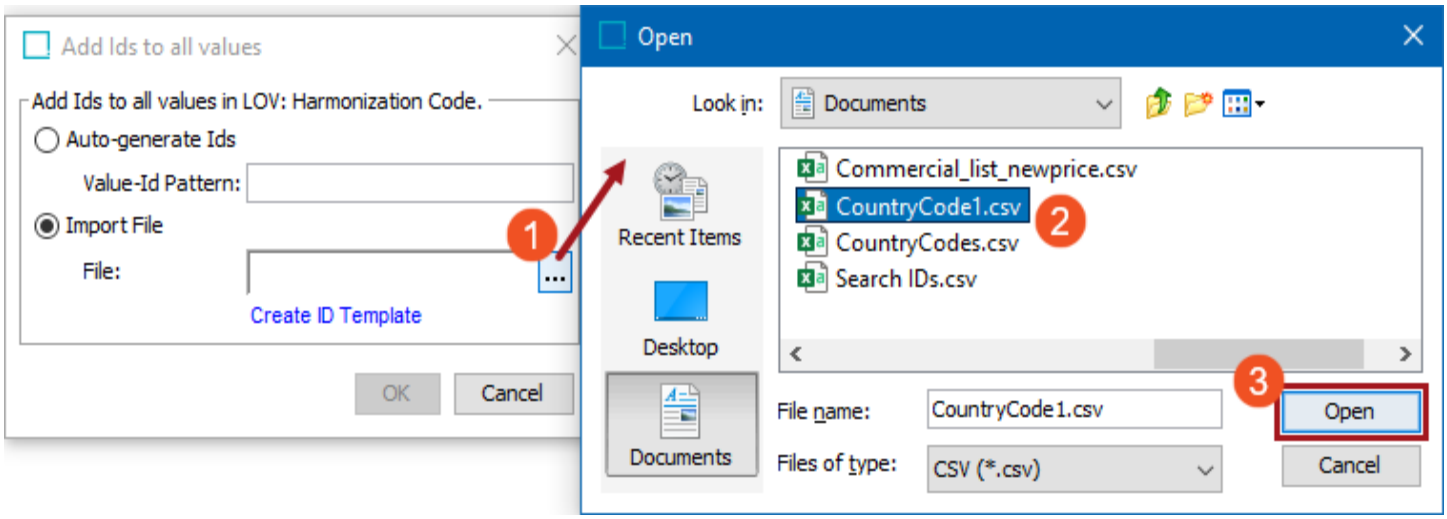
6. Once the template file is open, populate the name and ID column accordingly, and save the file.



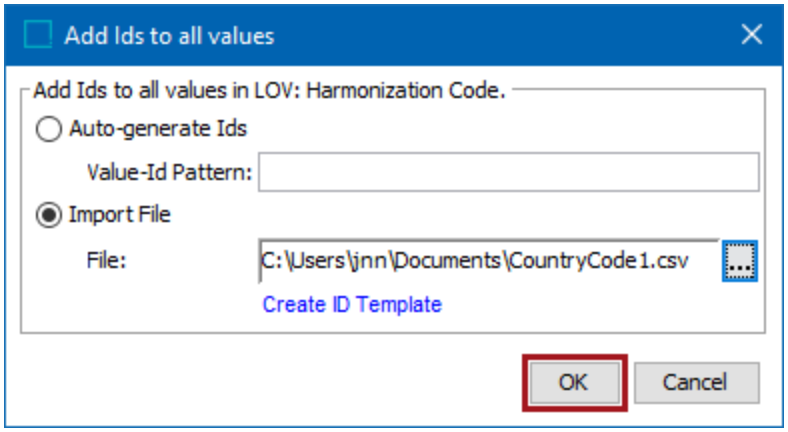
	A	B	C	D
1	CONTEXT_ID	VALUE	UNIT_ID	ID
2	Context1	ANGUILLA		AI
3	Context1	ANTIGUA AND BARBUDA		AG
4	Context1	ARGENTINA		AR
5	Context1	ARUBA		AW
6	Context1	AUSTRALIA		AU
7	Context1	AUSTRIA		AT
8	Context1	BAHAMAS		BS
9	Context1	BARBADOS		BB
10	Context1	BELIZE		BZ

Note: Unit ID column is only filled if there are units (i.e., inches, pounds, centimeters).

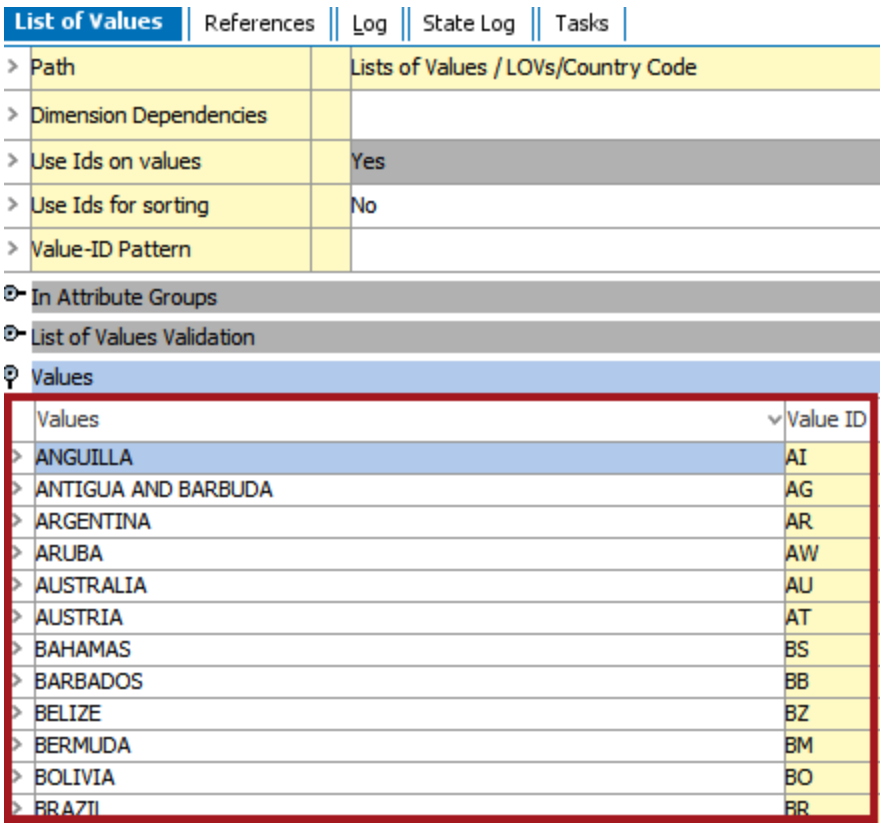
7. On the **Add IDs to all values** dialog, click the ellipsis button (...) and select your file. Click the **Open** button.



8. On the **Add IDs to all values** dialog click the **OK** button.

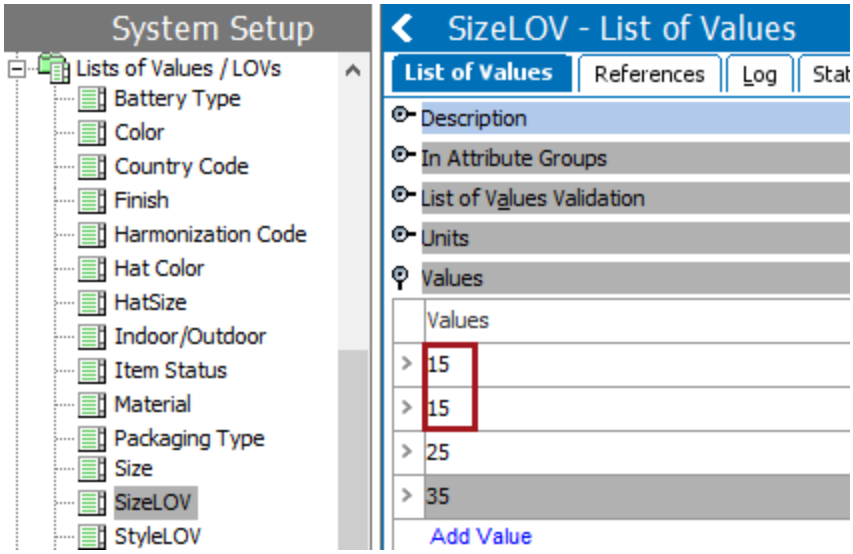


9. The imported data displays in the Values flipper with the new IDs.

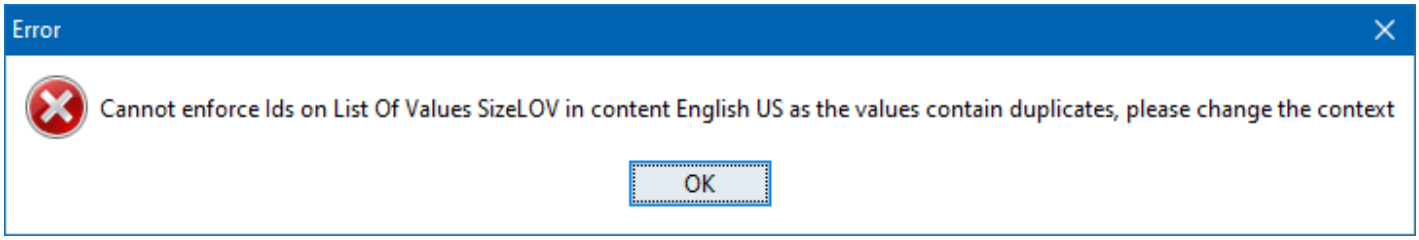


Allowing Value IDs to be Added to an LOV

Merging values in an LOV requires that Value IDs are used. Adding Value IDs requires that no duplicate LOV values exist. For example, the SizeLOV list of values has a Language dimension dependency and contains duplicate values for the English US context.



Attempting to set the 'Use Ids on values' parameter (in the Description flipper) to Yes, results in the displaying the following error:

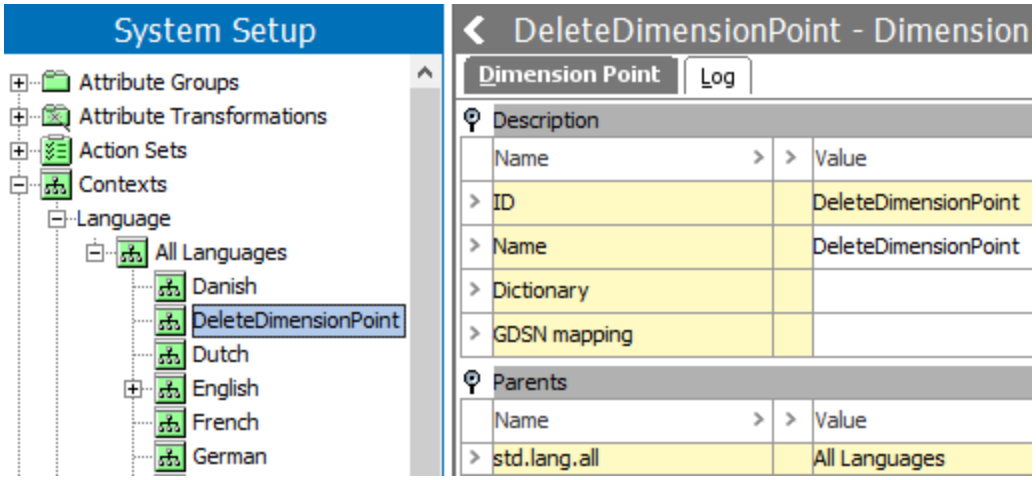


To resolve this issue and allow Value IDs to be used, first a new (temporary) dimension point and context are created. Unique values are then added for the temporary context and the Value IDs are applied. Finally, the temporary context is deleted. This process allows unique Value IDs to be assigned to the duplicate values (in other contexts), which then allows those duplicate values to be merged successfully.

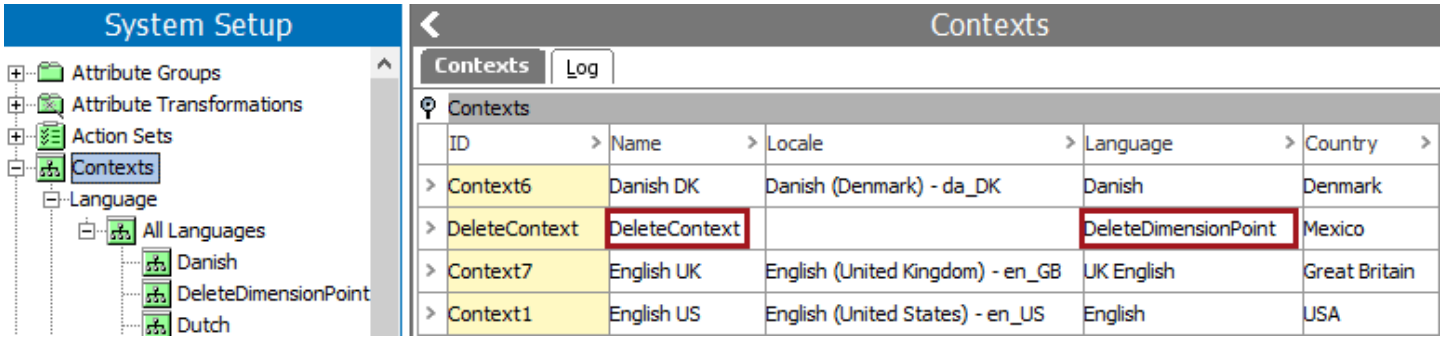
Important: These steps require care since they involve creating and deleting a temporary dimension point and context.

Configuration

1. In System Setup, open the Contexts node, right-click a dimension point (Language in the example below), and add an ID and Name for a new (temporary) dimension point. Click **Create**. For example, the following image shows a dimension point named 'DeleteDimensionPoint' so it is not confused with the active dimension points.

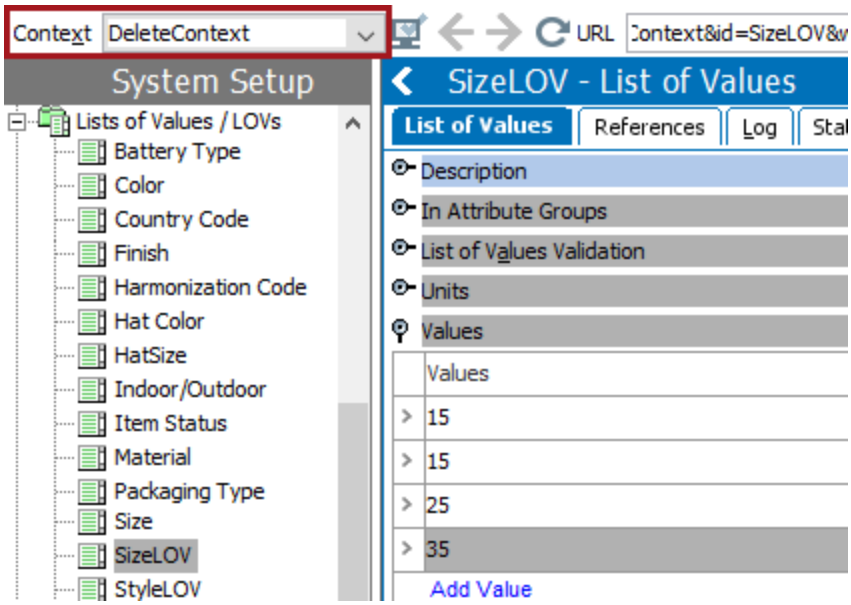


2. Select the Contexts node, click the **Add New Context** link to create a temporary context. Add a context name and click **OK**. In this example, the new context is named 'DeleteContext'.

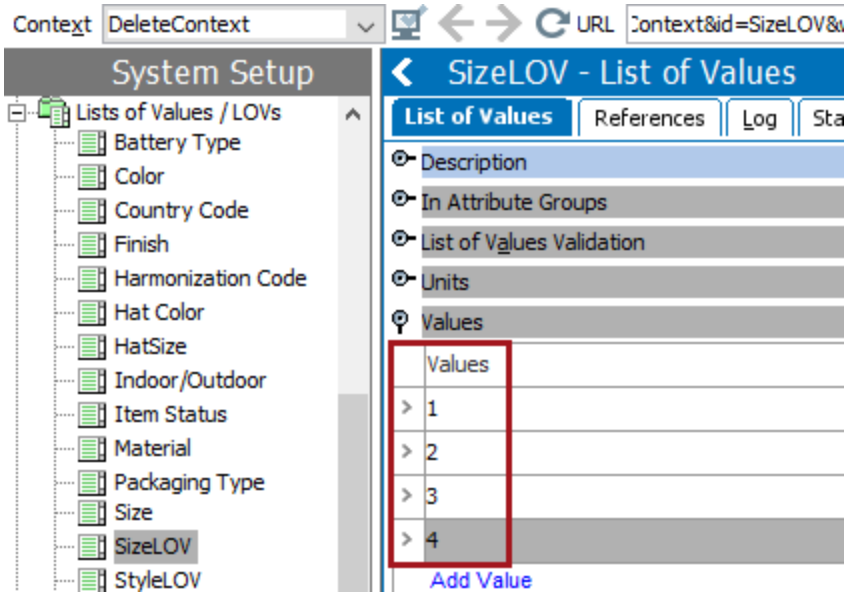


3. For the new temporary context (DeleteContext), select the new temporary dimension point. In this example, DeleteDimensionPoint is selected for Language. The other settings are irrelevant.

4. Click on the LOV, and select the temporary context 'DeleteContext'.



5. Add unique values for each row of the temporary context, including replacing inherited values if displayed.



6. While continuing to display the temporary context (DeleteContext), proceed to add IDs to all values as described in the Configuration steps of the **Adding IDs to Values in LOV** topic. During this configuration, setting the LOV 'Use Ids on values' parameter applies IDs to values in all contexts.
7. Once IDs are successfully added to the LOV values, delete the temporary dimension point and the temporary context.

> **Note:** For steps to merge duplicate values in an LOV, see the **Merging Values in an LOV** topic.

Consolidating 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 LOV Values in Bulk

The 'Merge LOV Values' option greatly eases the maintenance of LOVs that contain misaligned, repeated, or incorrect values across multiple contexts. This functionality allows users to upload a semicolon-separated CSV file containing the IDs of the LOVs in which values should be merged, the Value ID of the source value, and the Value 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).

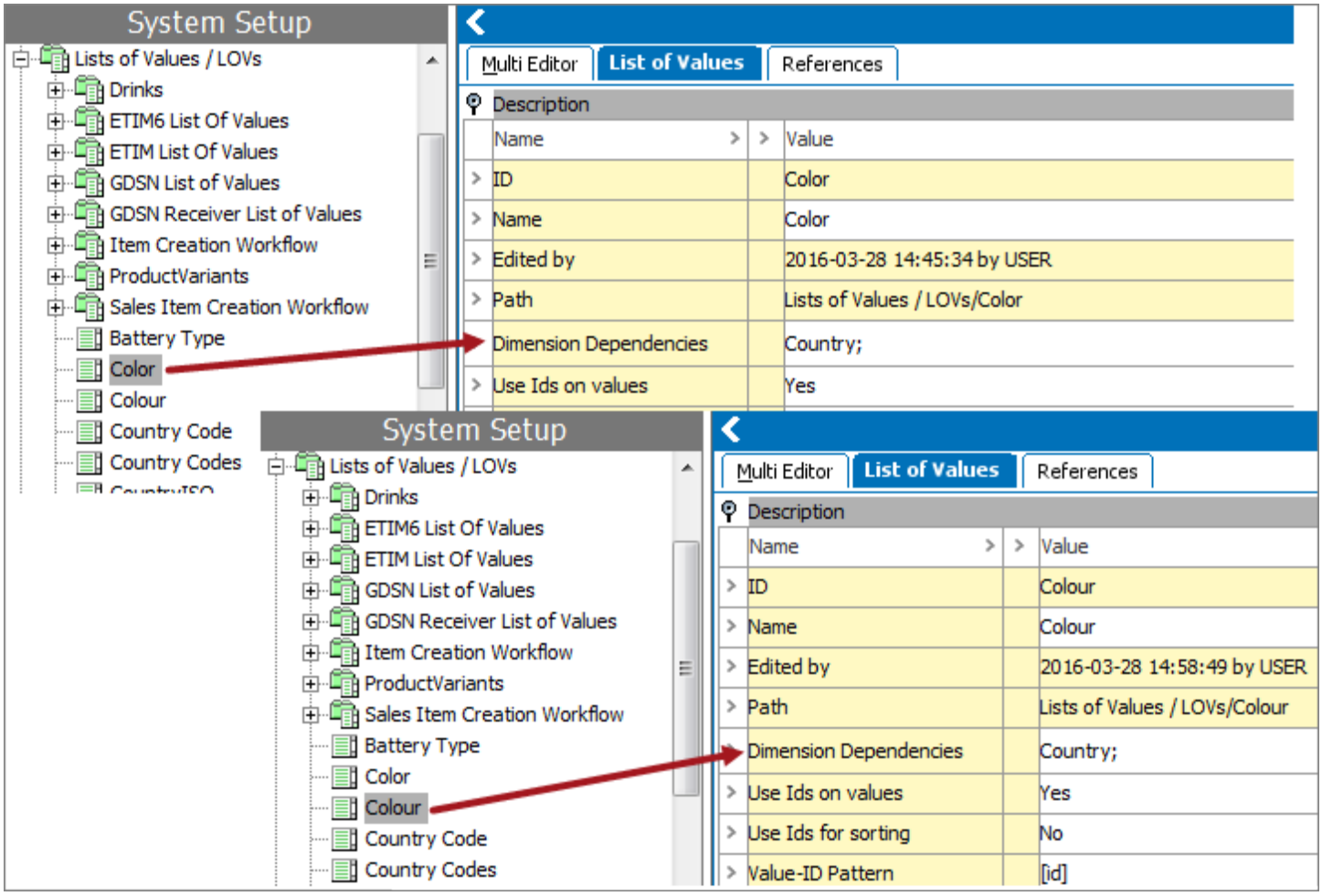
For more information, see **Merging LOV Values in Bulk**.

Merging LOVs

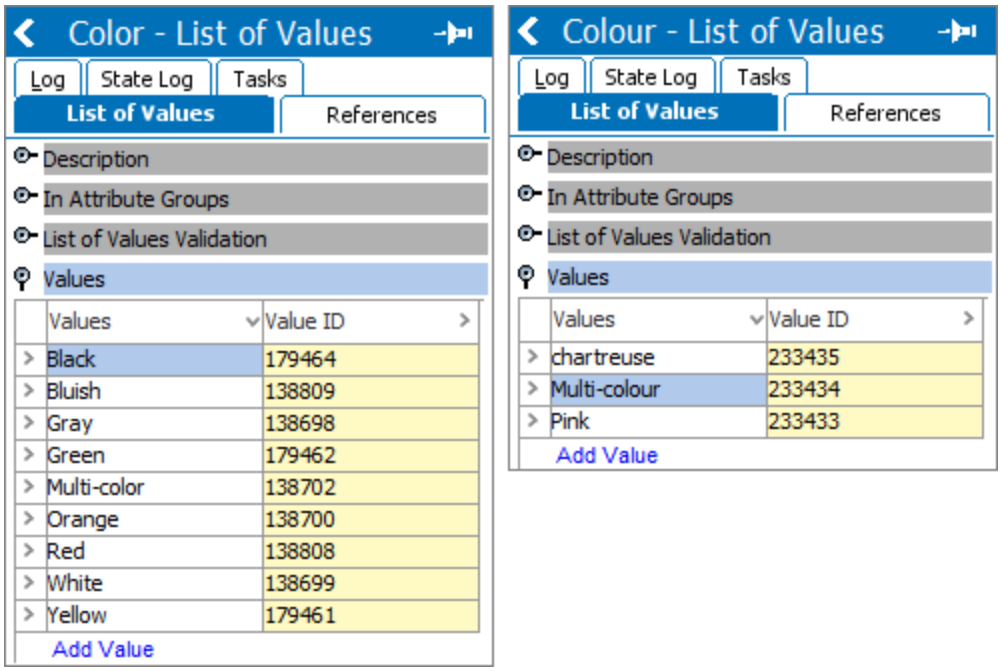
The Merge LOVs option can be used when LOVs with similar characteristics can be combined into one LOV, easing the maintenance burden.

Configuration

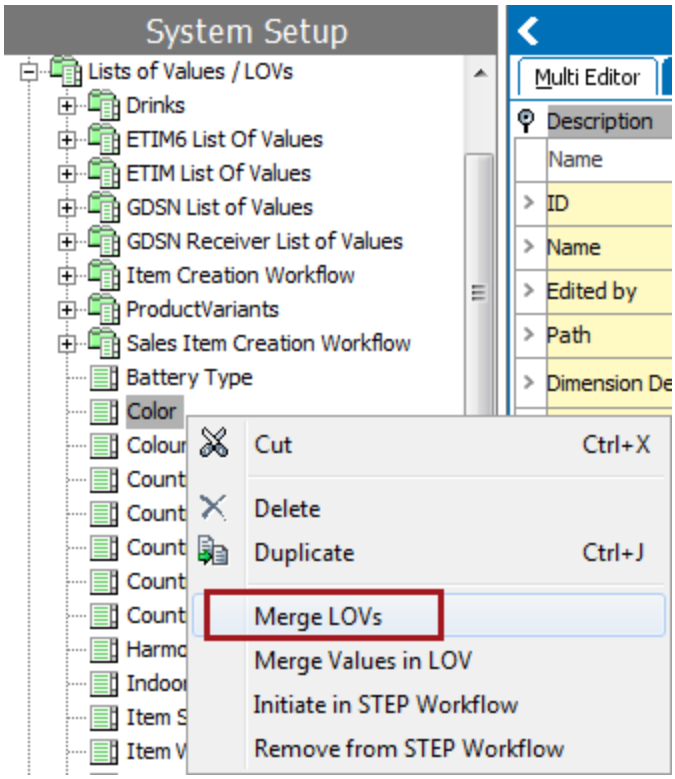
1. Go to System Setup > Lists of Values / LOVs > click on **each of the LOVs to be merged** and ensure the settings within the Dimension Dependencies parameter match. For the example below, the two attributes 'Color' and 'Colour' have the same value for the Dimension Dependencies parameter.



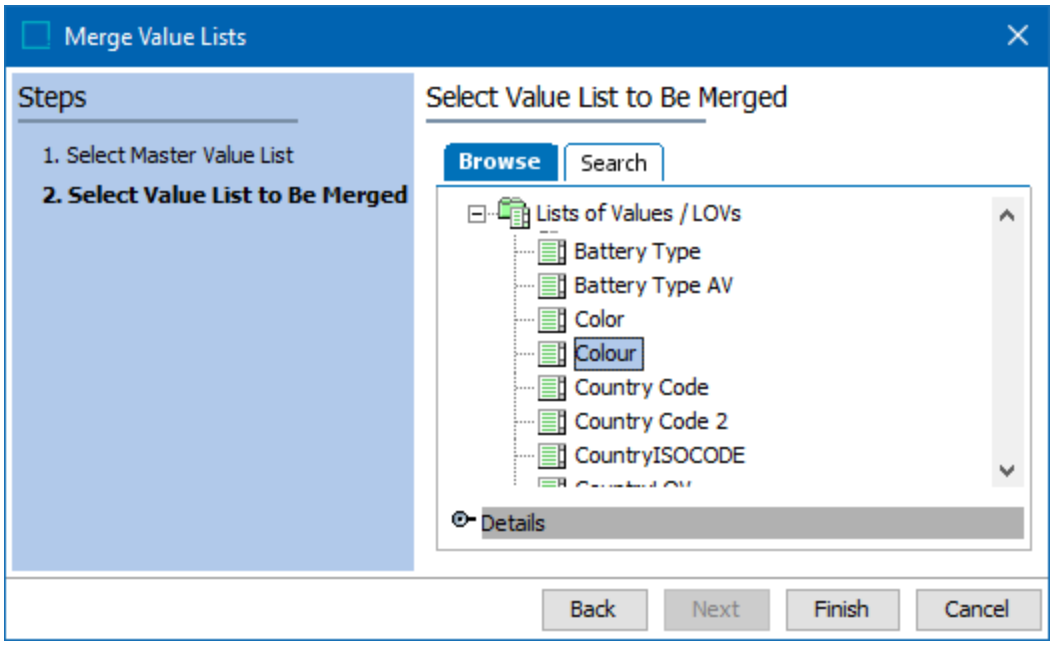
For this example, prior to merging the LOVs, the values for each of the LOVs can be seen in the screenshot below.



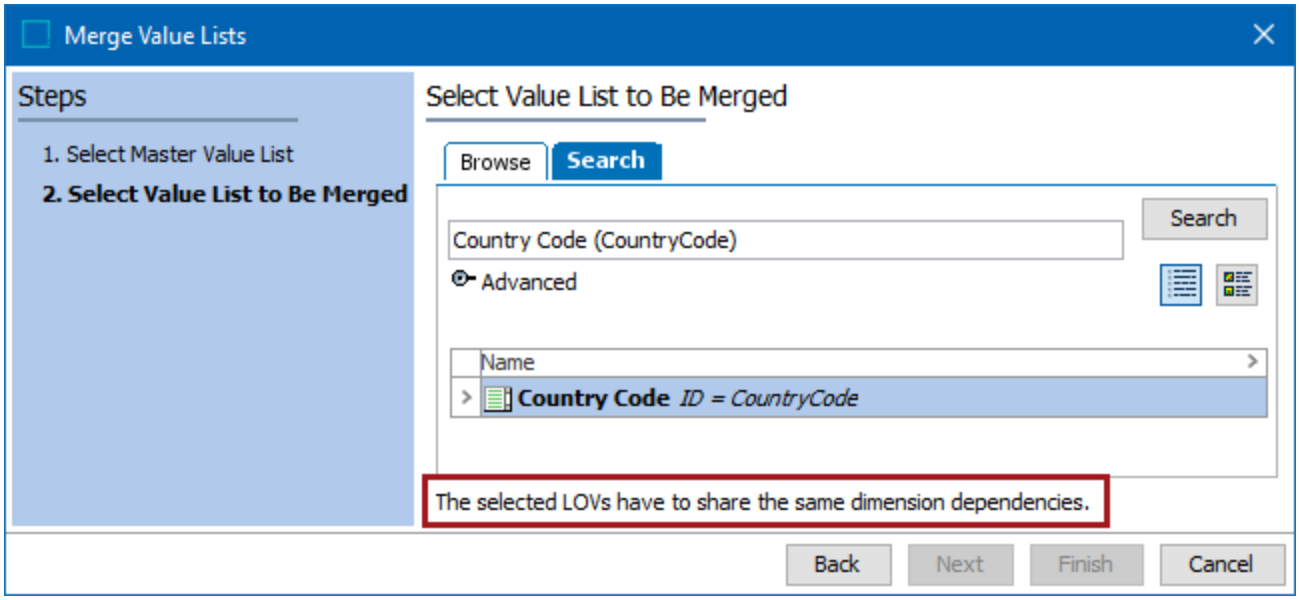
2. Highlight the master LOV (the LOV that will remain after the merge), right-click and select **Merge LOVs**.



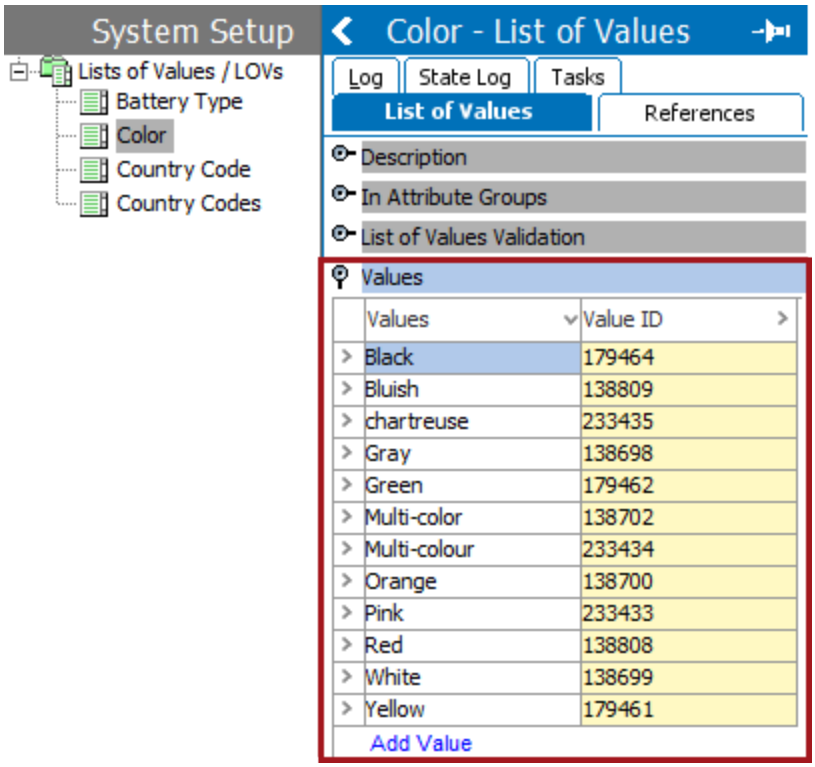
3. The Merge Value Lists wizard will display. Use the Browse or Search tabs to find and select the LOV to be merged (the LOV that will be deleted) with the master LOV.



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
Log
State Log
Tasks
List of Values
References
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

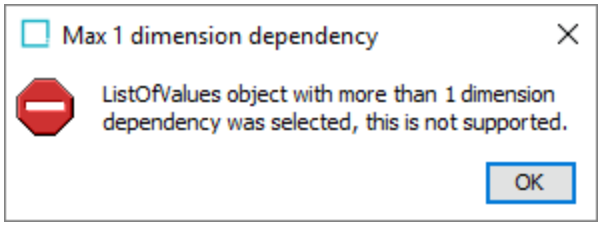
Merging Values in an LOV

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.

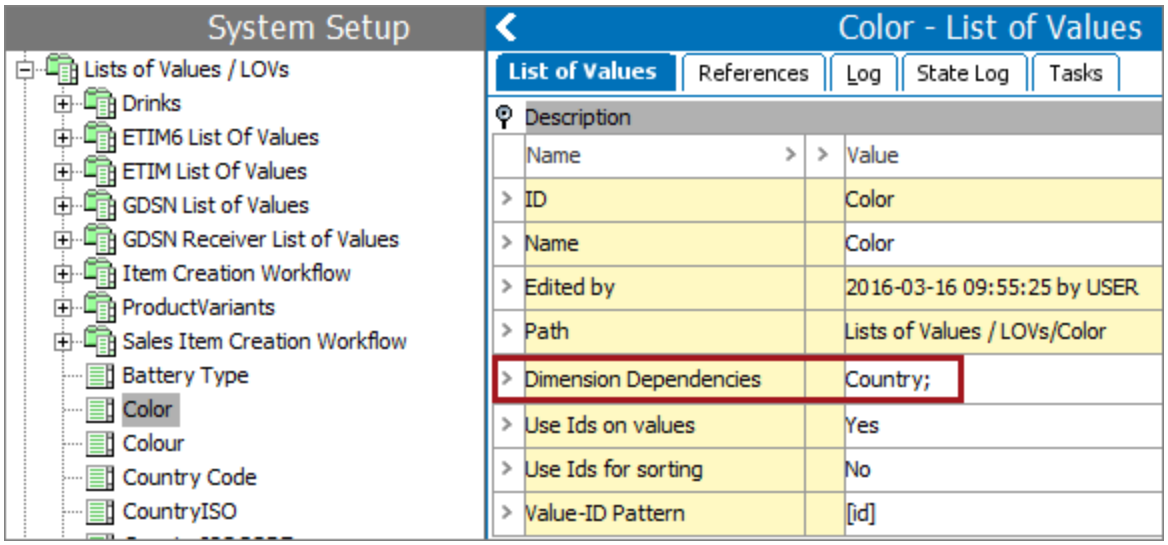
Verify the following prerequisites have been met before completing the configuration.

Prerequisites

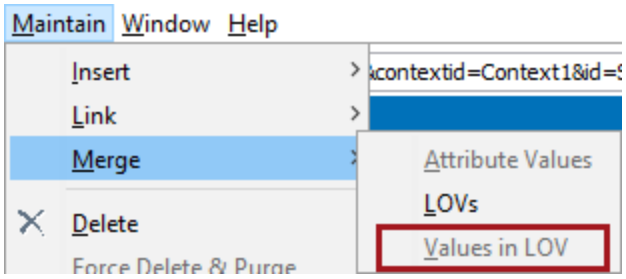
1. The LOV must have fewer than two dimension dependencies, otherwise the following warning dialog is displayed:



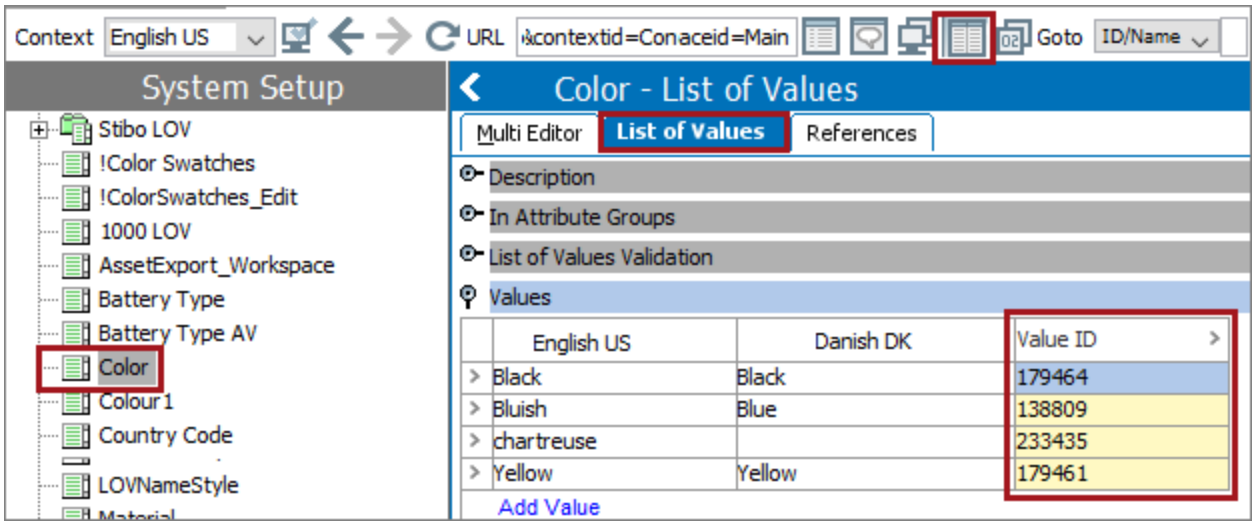
   To confirm the dimension dependencies of an LOV, in workbench System Setup > Lists of Values / LOVs > select the desired LOV to display the List of Values tab in the editor. Confirm the Dimension Dependencies parameter lists no more than one dimension dependency.



2. The LOV must use Value IDs, otherwise the menu option to Maintain > Merge > Values in LOV option is disabled:



To confirm the LOV uses value IDs, in workbench System Setup > Lists of Values / LOVs > select the desired LOV to display the List of Values tab in the editor. Click the **Context Mode** button ( ) in the toolbar and open the Values flipper to display the **Value ID** column as shown below.



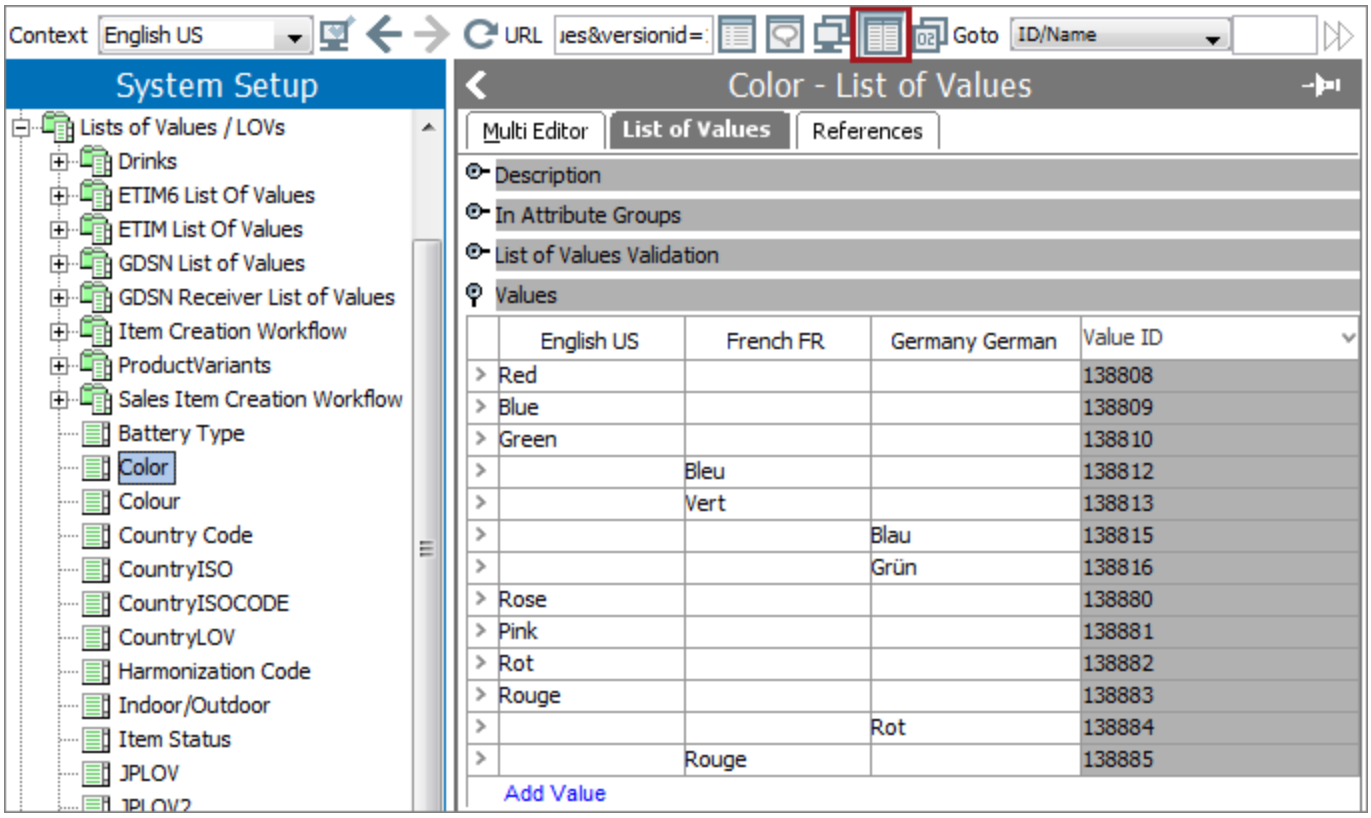
Note: Because the LOVs are dimension dependent each LOV value has a different 'Value ID'. This is important to notice for later steps.

If the Values in LOV option is disabled and the LOV does not have Value IDs, see the **Adding IDs to Values in LOV** topic.

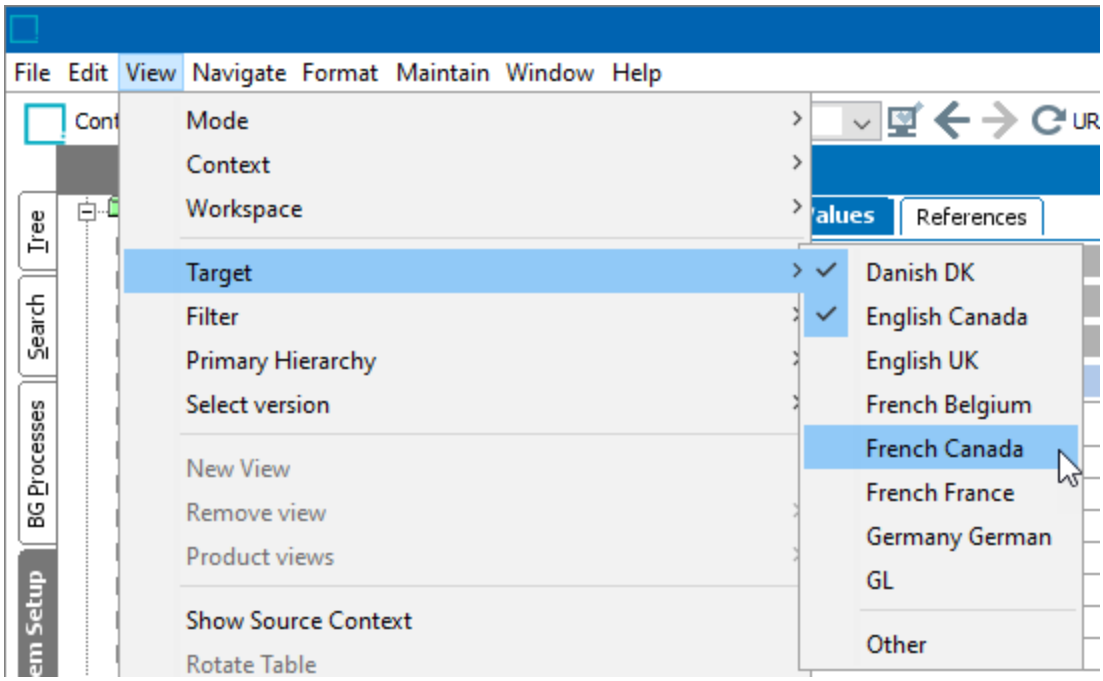
Configuration

Once the prerequisites have been confirmed, follow the steps below to merge values in an LOV.

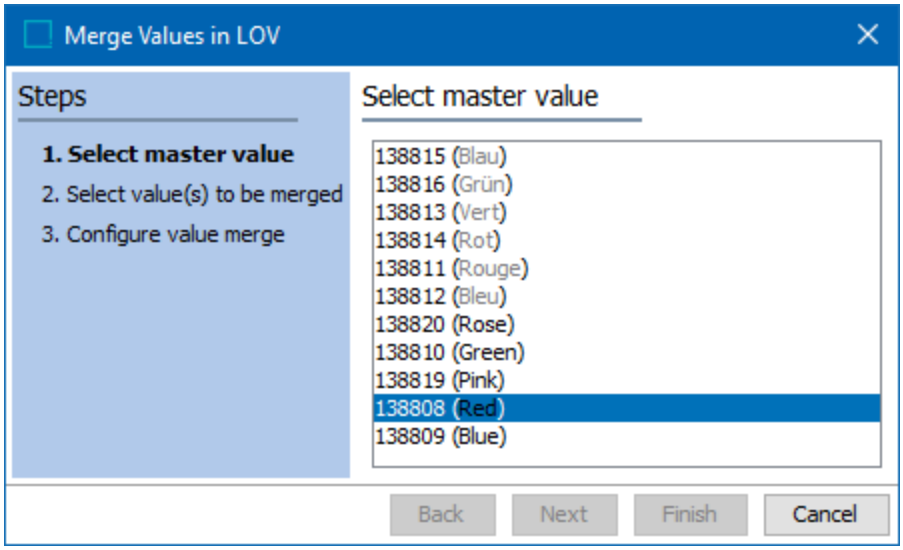
1. In workbench > System Setup > select the desired LOV > select the List of Values tab > click the **Context Mode** button in the toolbar to view all available values for an LOV in other contexts.



2. Optionally, choose the desired Target Contexts to display within the editor. From the menu bar, click View > Target > and click on the desired contexts to display.

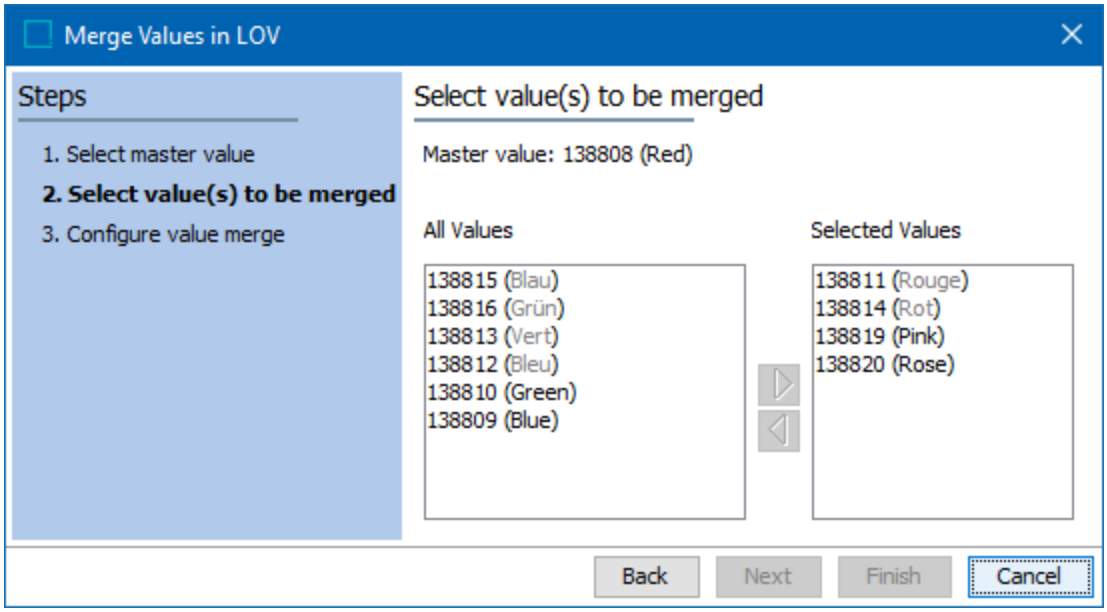


3. On System Setup > Lists of Values / LOVs > select the desired LOV > right-click the selected LOV > select **Merge Values in LOV**. The Merge Values in LOV dialog will display.



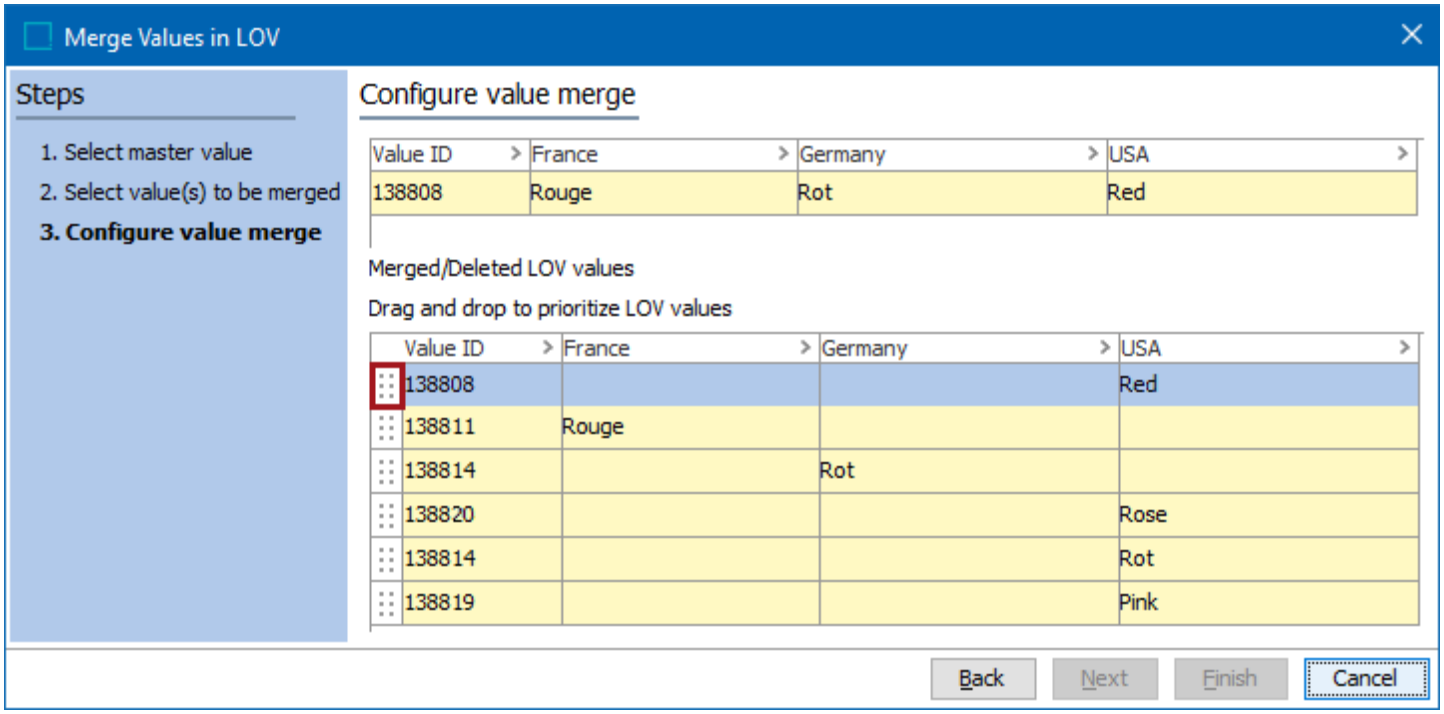
4. Select the value to be the master (the value that will remain after the merge), and click the **Next** button.
5. Identify the values to be deleted by highlighting a value in the **All Values** column, clicking the right arrow button and moving it to the **Selected Values** column. All Selected Values will be deleted.

   Repeat this process for all values to be merged into the master value and click the **Next** button.

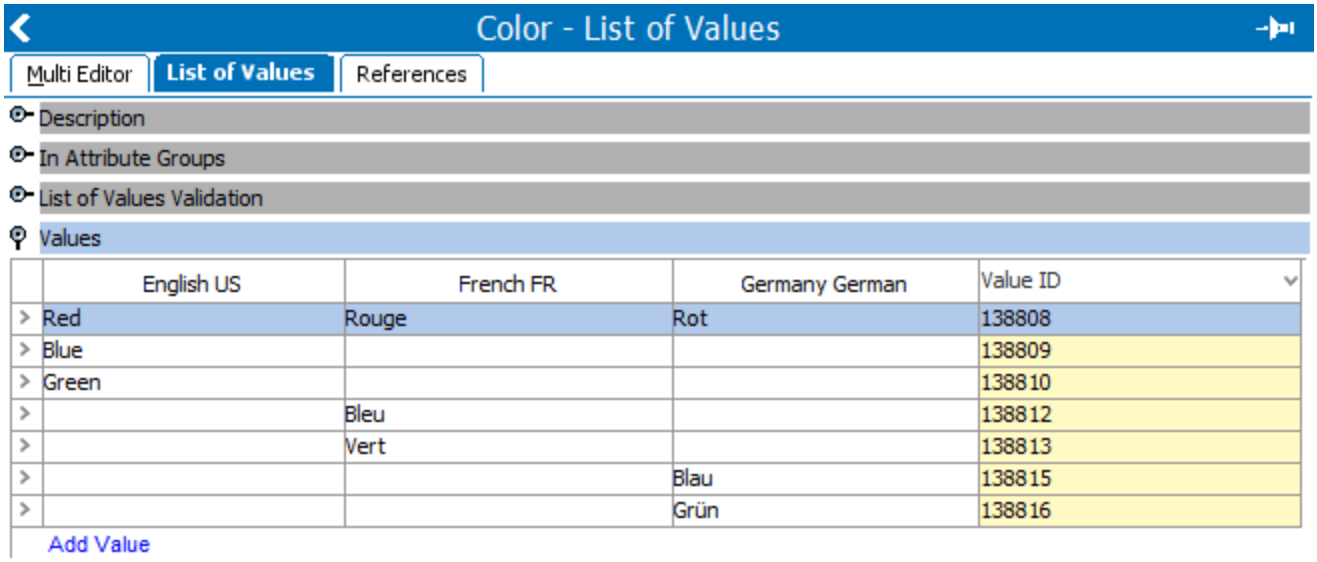


Note: Remove a value from the 'Selected Values' list by selecting it and clicking the left arrow button.

6. The 'Configure value merge' step 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 image 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.'



7. 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).



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., .xslx)

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

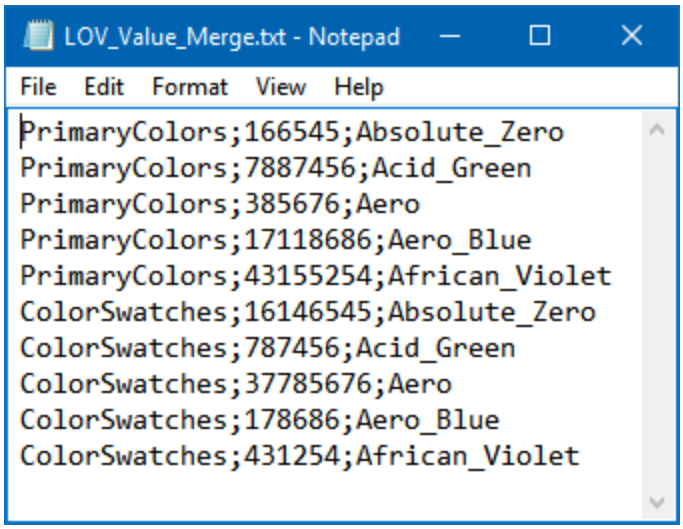
• The **TargetValueID** column accepts a LOV Value ID, which is to be used as target for the merge

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.

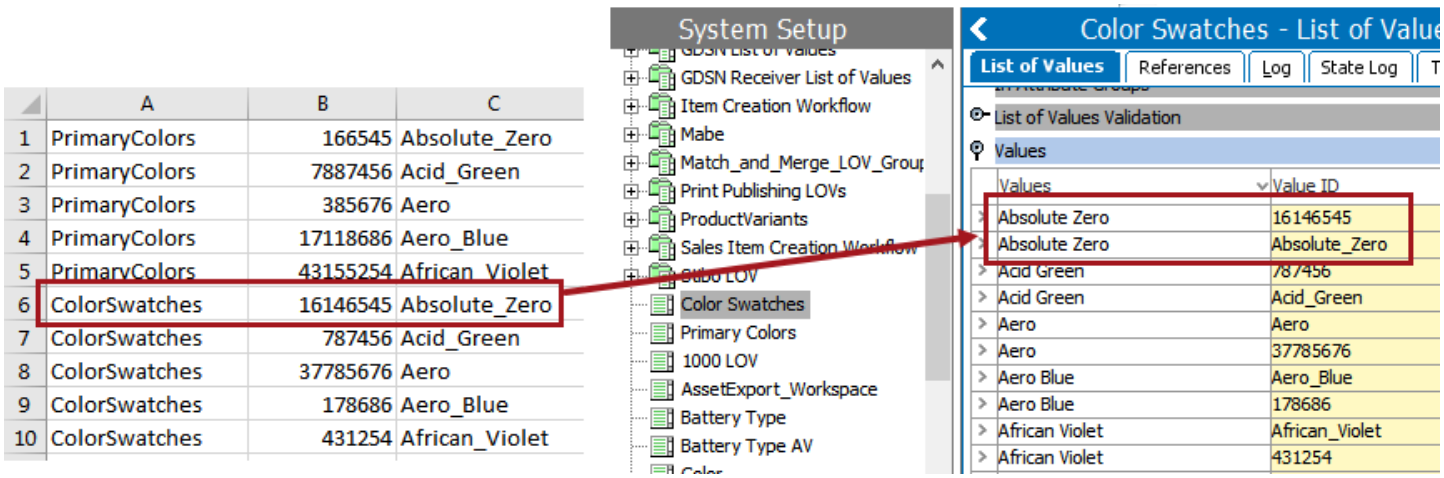
The below screenshot shows a sample file that can be used to merge values within two different LOVs at one time—one with the ID of PrimaryColors and a second with the ID of ColorSwatches. Column B contains the IDs of the LOV values that will be removed (source values) when the file is imported. Column 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

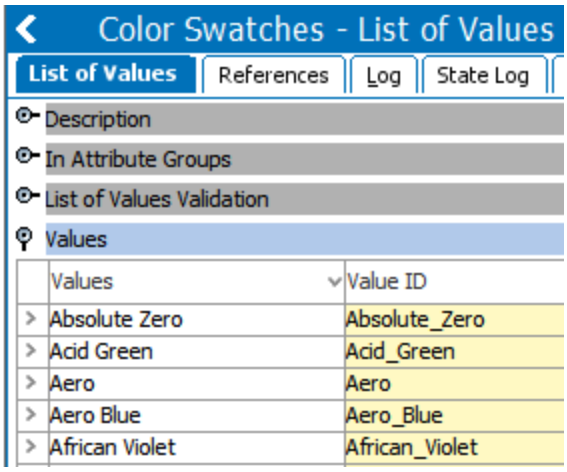
A .txt file containing semicolon-separated values may also be used.



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.



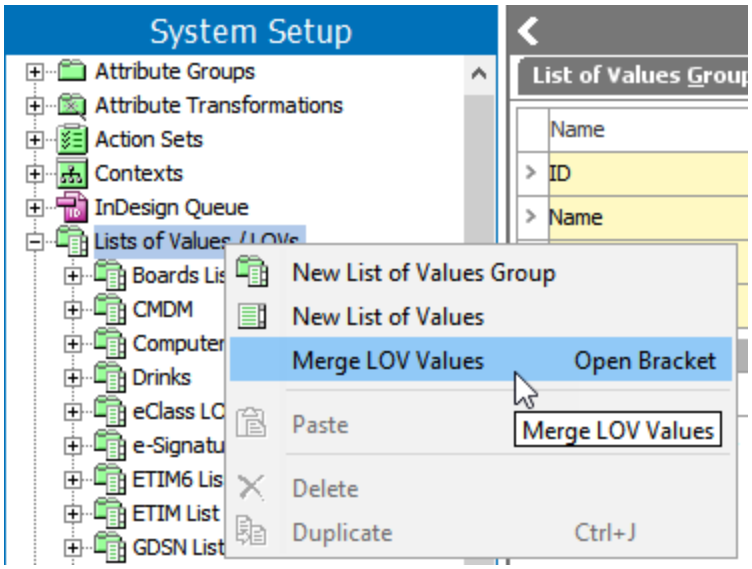
After the sheet is uploaded, the ColorSwatches LOV in this example will look as follows, with only the values in column C of the CSV sheet remaining.



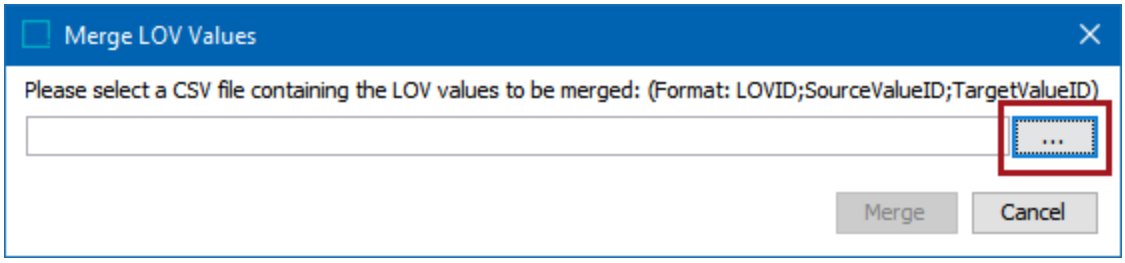
Uploading the Merge LOV Values CSV Sheet

1. Navigate to System Setup, open Lists of Values / LOVs, then right-click on any LOV group (including the root node) and select **Merge LOV Values**. Or use the shortcut

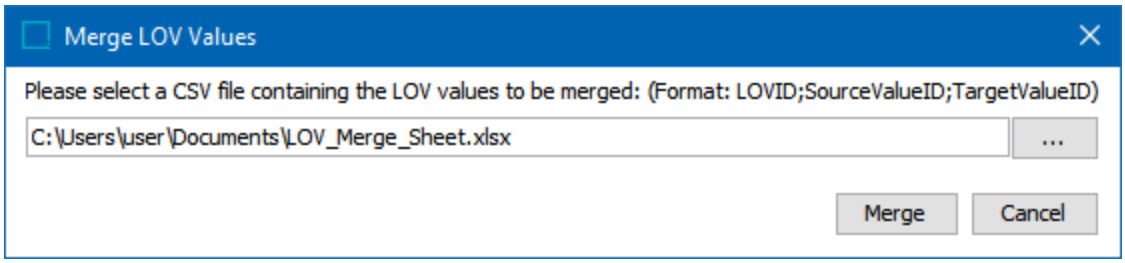
> **Note:** 'Merge LOV Values' should not be confused with 'Merge Values in LOV,' which is the manual process for merging values within an individual LOV and is only available when right-clicking on an LOV, not on an LOV group. For more information on this function, see the **Merging Values in an LOV** topic.



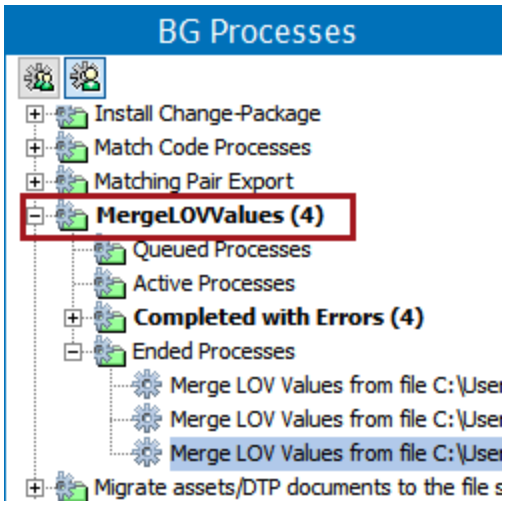
2. Click the ellipsis button (...) on the **Merge LOV Values** dialog to select the CSV file containing the LOV values to be merged.



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>
```

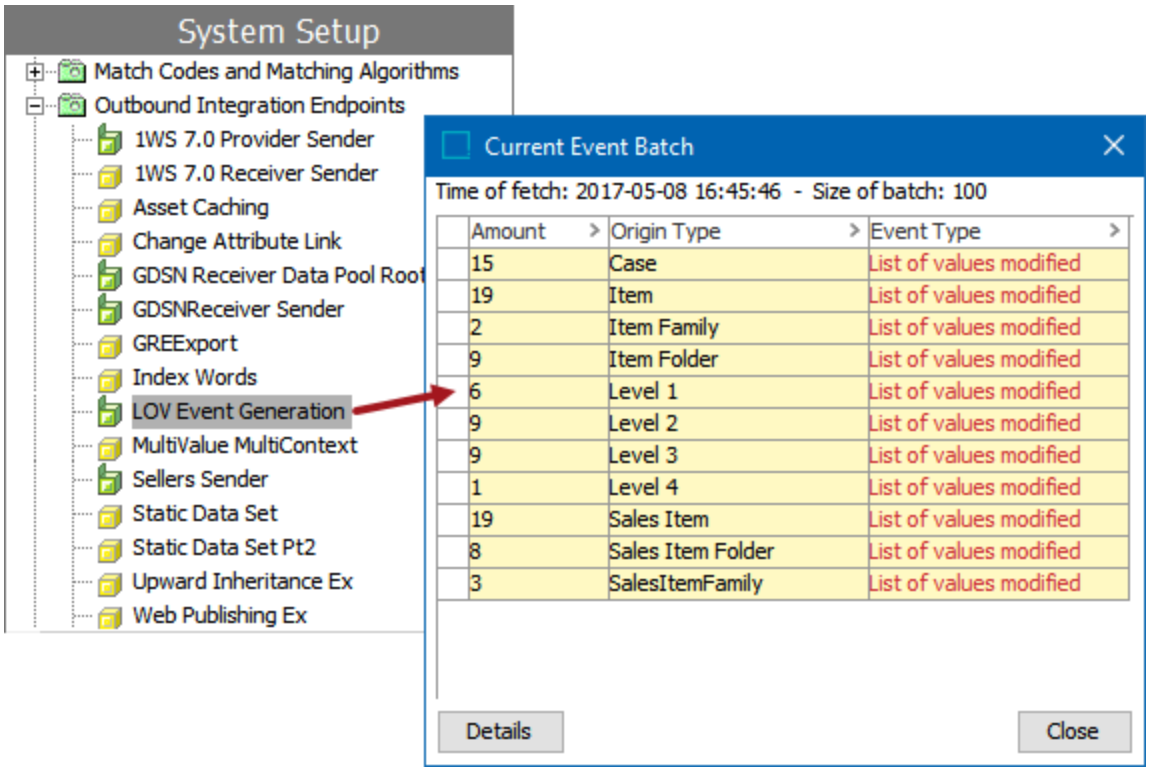
Force can have a value of 'true' or 'false.' If set to true, the value will be deleted from the LOV even if there are attributes using the value. If approved objects in STEP are using the attribute that contained the deleted value, the approval status of these objects will not be affected, i.e., the 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.

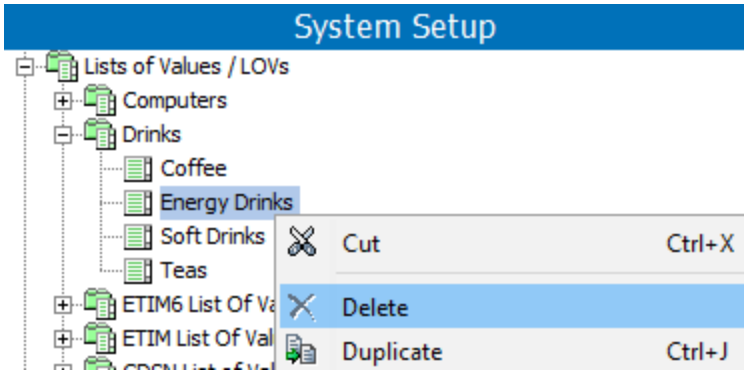
Note: *Approve* events cannot be generated for deletions of LOV values. i.e., if a product uses an attribute based on the LOV and contained the value that was deleted from the LOV, the deletion of the LOV values through the STEPXML import does not unapprove the product.



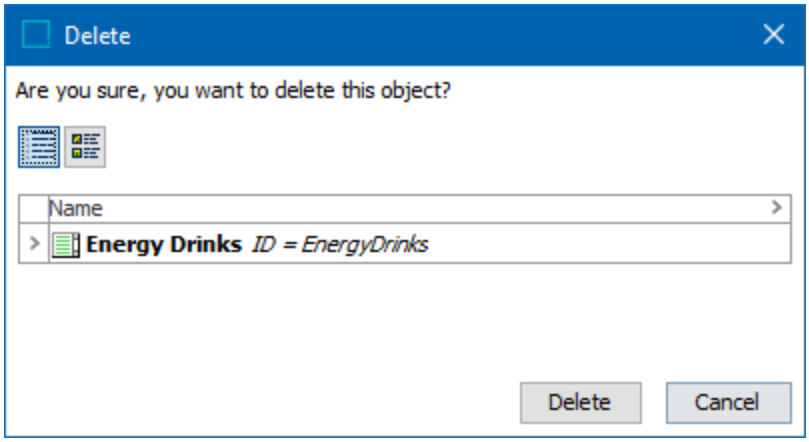
For more information on event generation, see the **Events** section of the **System Setup / Super User Guide** documentation. For more information on event-based OIEPs, see the **Creating an Event-Based Outbound Integration Endpoint** section of the **Data Exchange** documentation.

Deleting LOVs

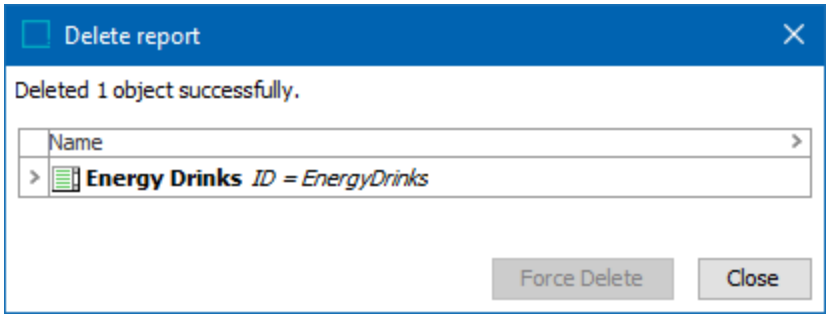
1. Go to System Setup > List of Values / LOVs > Select **the LOV to be deleted**.
2. Right-click **the LOV to be deleted**, and the menu will display as shown below.



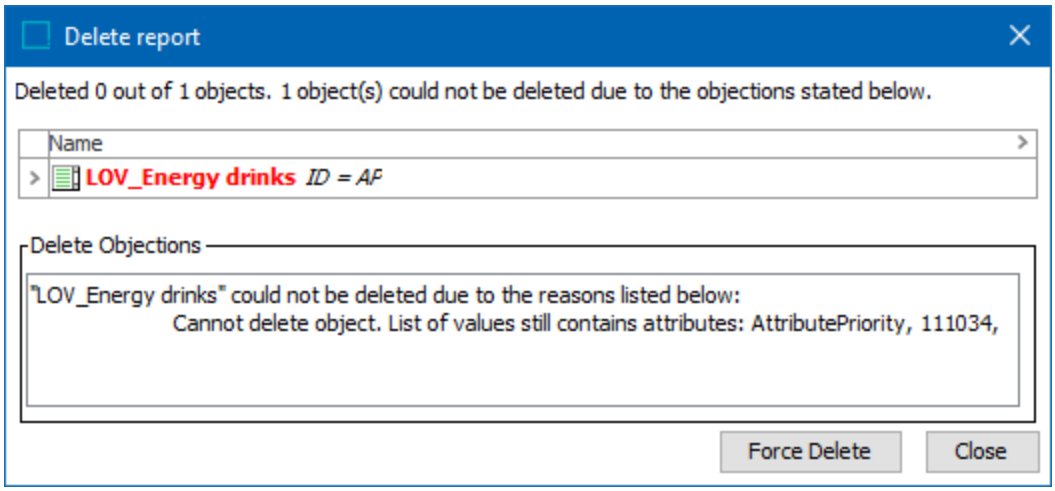
3. Click **Delete** and a Delete dialog displays.



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).



Filtering LOVs

The following LOV filtering options are cumulative and work together to provide only the necessary values at each point of filtering:

1. **Attribute filtering** - used when not all of the values that exist in the LOV are needed by all attributes using the LOV. This is important since an LOV can be used by more than one attribute.
2. **Hierarchy node filtering** - used when the need for the LOV values is determined by the location of the object in the hierarchy (product hierarchy or classification hierarchy).

Consider an LOV with 18 values. When used by attribute A, only 11 values are legal; the seven (7) other LOV values are not relevant for attribute A. When the same LOV is used on attribute B, 15 LOV values are legal. This can be achieved with attribute level LOV filtering.

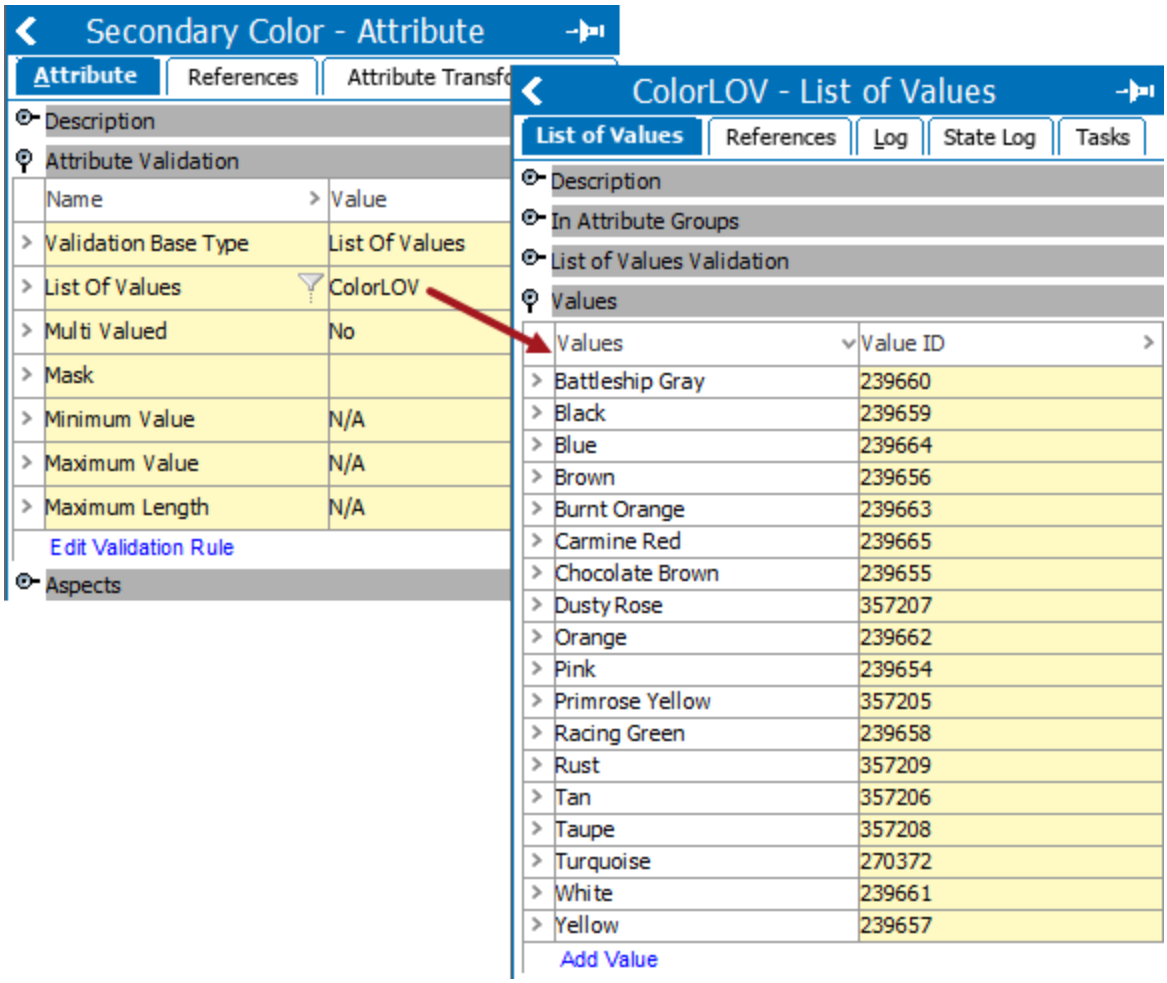
Considering the same 18 LOV values above, the 11 legal LOV values for attribute A can be further constrained when hierarchy filtering is applied at different levels in the primary product hierarchy (PPH). This allows attribute A to display only five (5) legal LOV values within the 'Shirts' family, while displaying four (4) legal LOV values for the 'Pants' family.

Filtering on Attribute and PPH

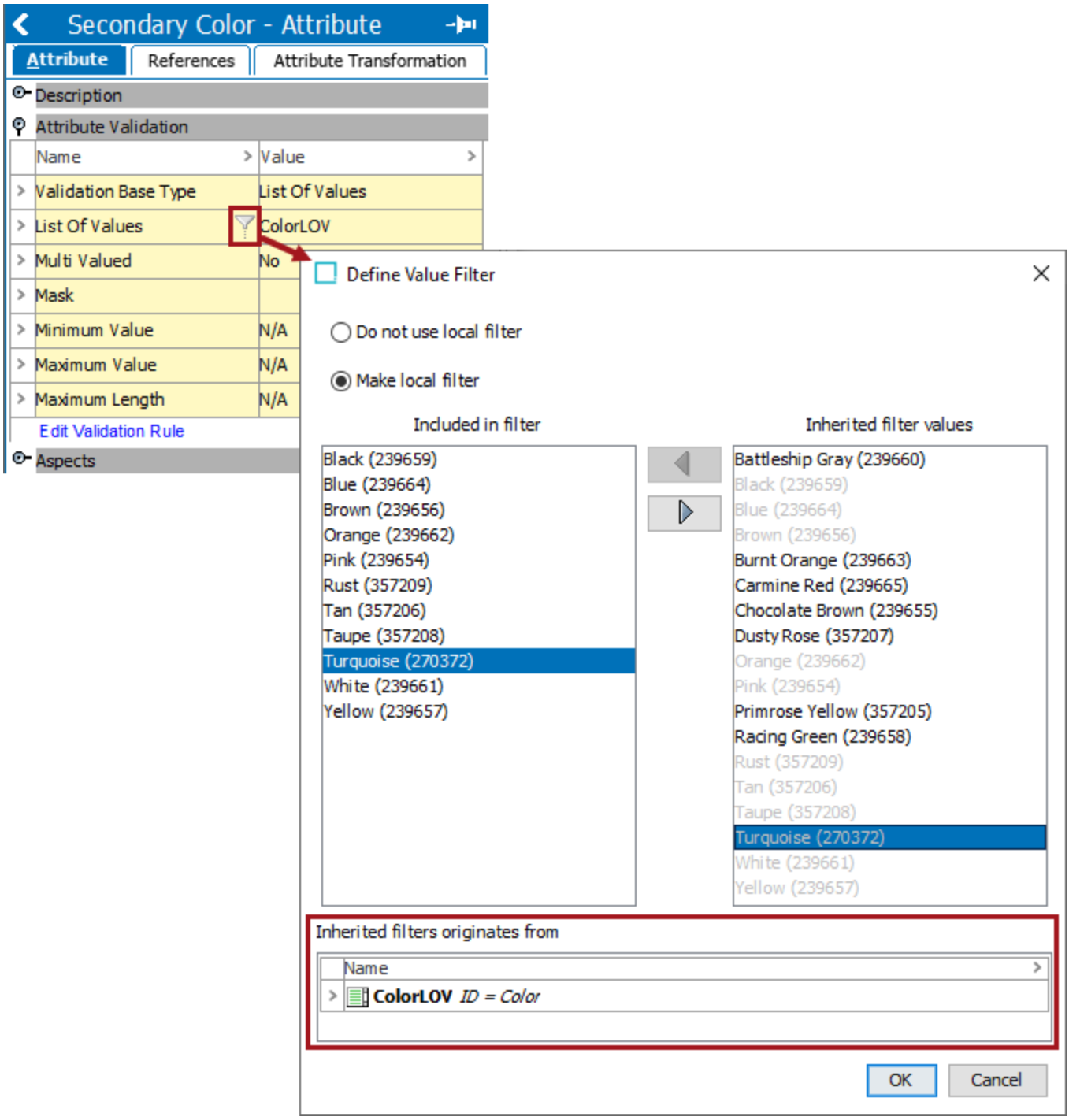
The cumulative effect of applying filters at both the attribute and the primary product hierarchy means increasingly more filtering can be added at lower levels of the PPH.

Note: An LOV value cannot be made available at a lower level of the PPH if it has been filtered out higher up in the hierarchy.

The 'Secondary Color' attribute uses the 'ColorLOV' list of values, which has 18 values:



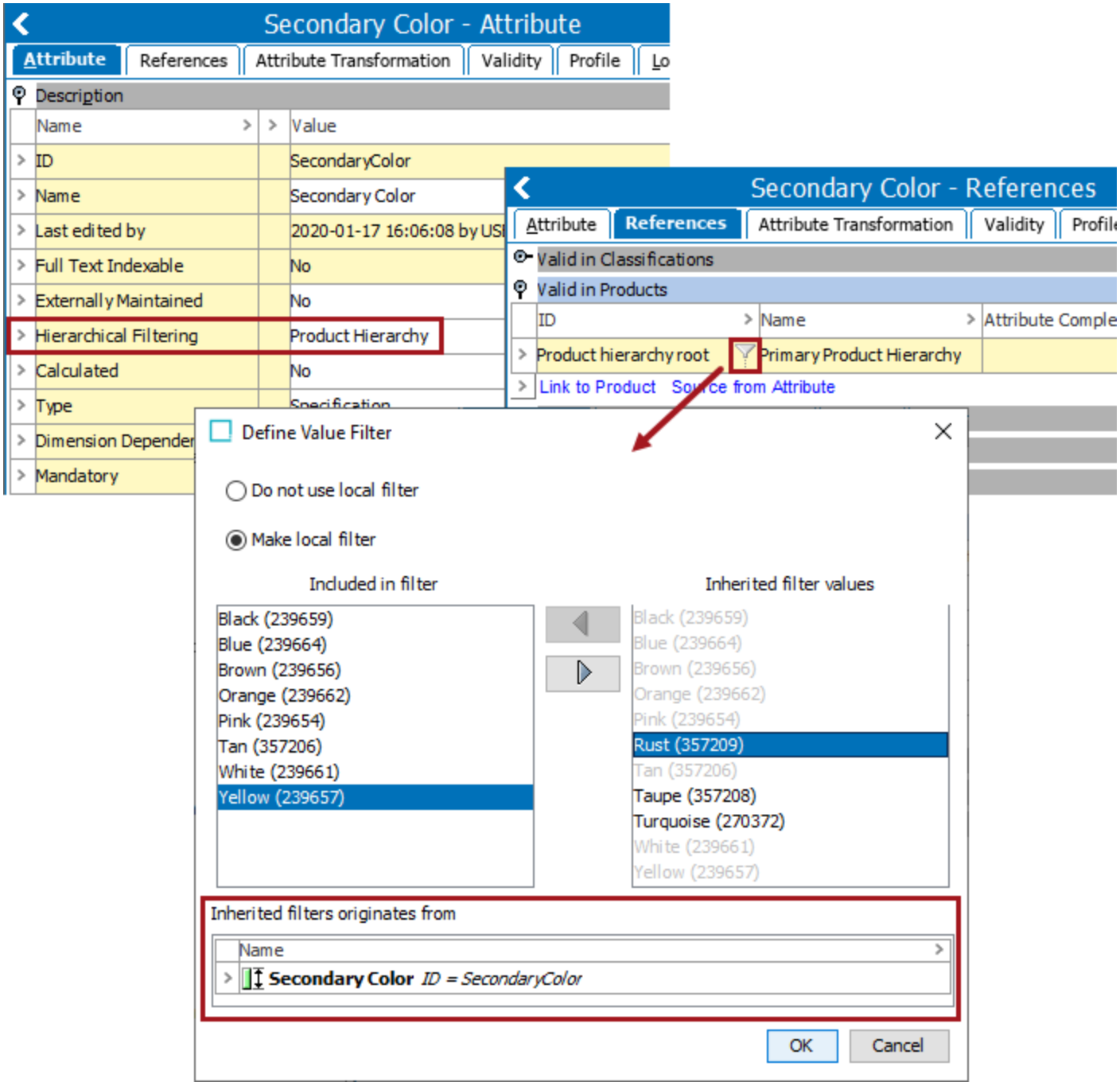
On the 'Secondary Color' attribute, display the attribute filter by clicking the List Of Values filter and show the 'Define Value Filter' dialog. The 'Inherited filters originates from' section shows the corresponding LOV, which in this example is the 'ColorLOV' list of values.



In the case of the removed values (shown as gray text) in the above image, this considers that the primary product hierarchy is organized according to similar product specifications. This shows that certain LOV values are not applicable for some product lines, while the same values are applicable in other product lines.

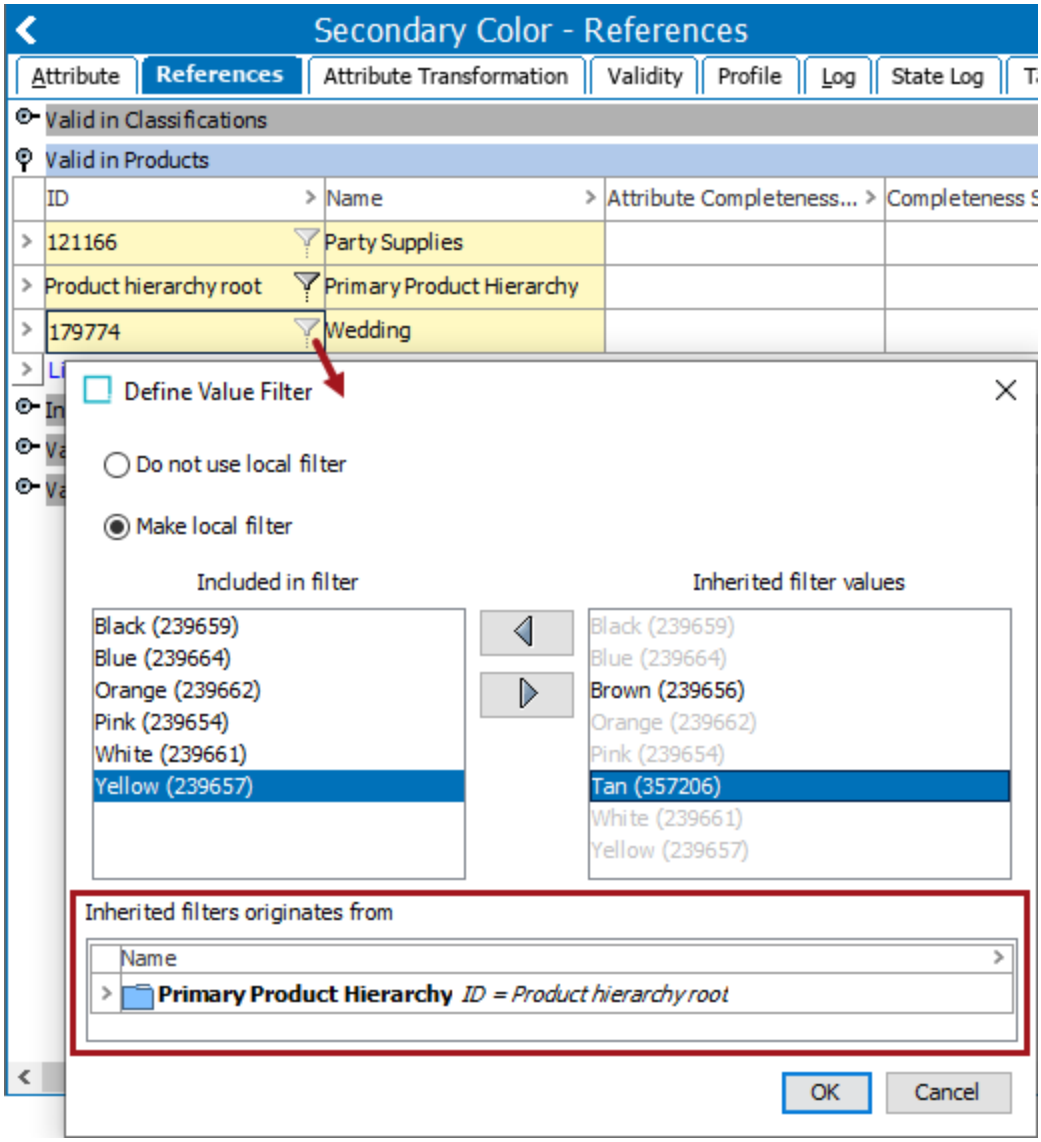
If the 'Secondary Color' attribute is linked to the PPH at the top level it can further be filtered, as shown below. With this filter, two additional LOV values ('Rust' and 'Taupe') have been removed from the list of valid values,

and the 'Inherited filters originates from' section shows the values are derived from the 'Secondary Color' attribute.



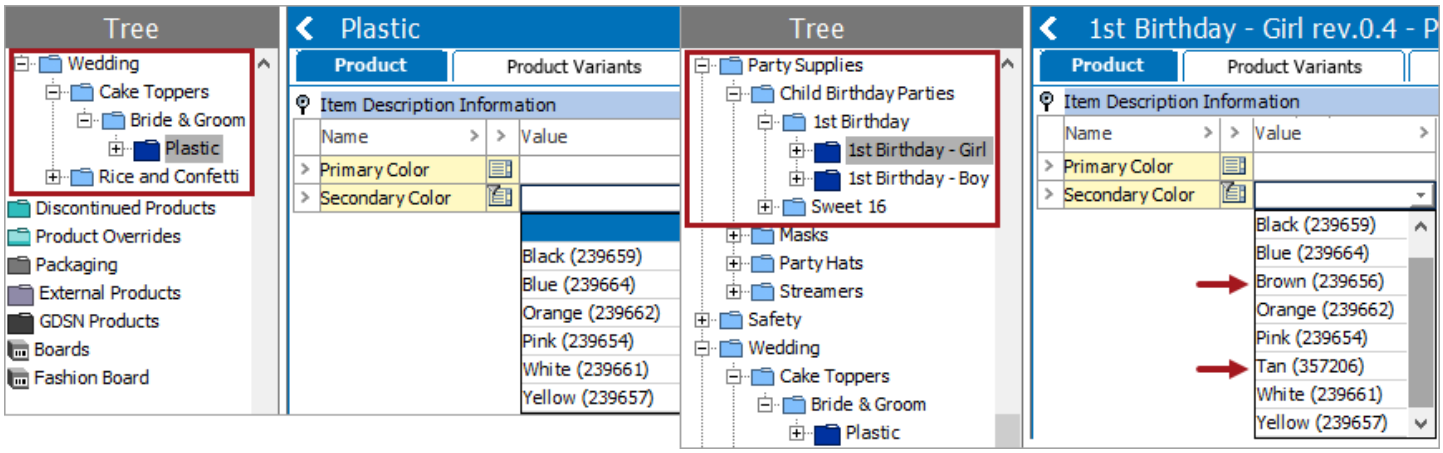
Consider that when the attribute is used in a particular product line, the values are further limited. On the 'Secondary Color' References tab, the filter will be set on the 'Wedding' node. The gray color of the filter buttons indicates that no filter is applied at the 'Party Supplies' nor the 'Wedding' nodes.

For the 'Wedding' node, two further values have been removed from the valid selection of values, and the 'Inherited filters originates from' section shows the values are derived from the 'Primary Product Hierarchy' (PPH) node.



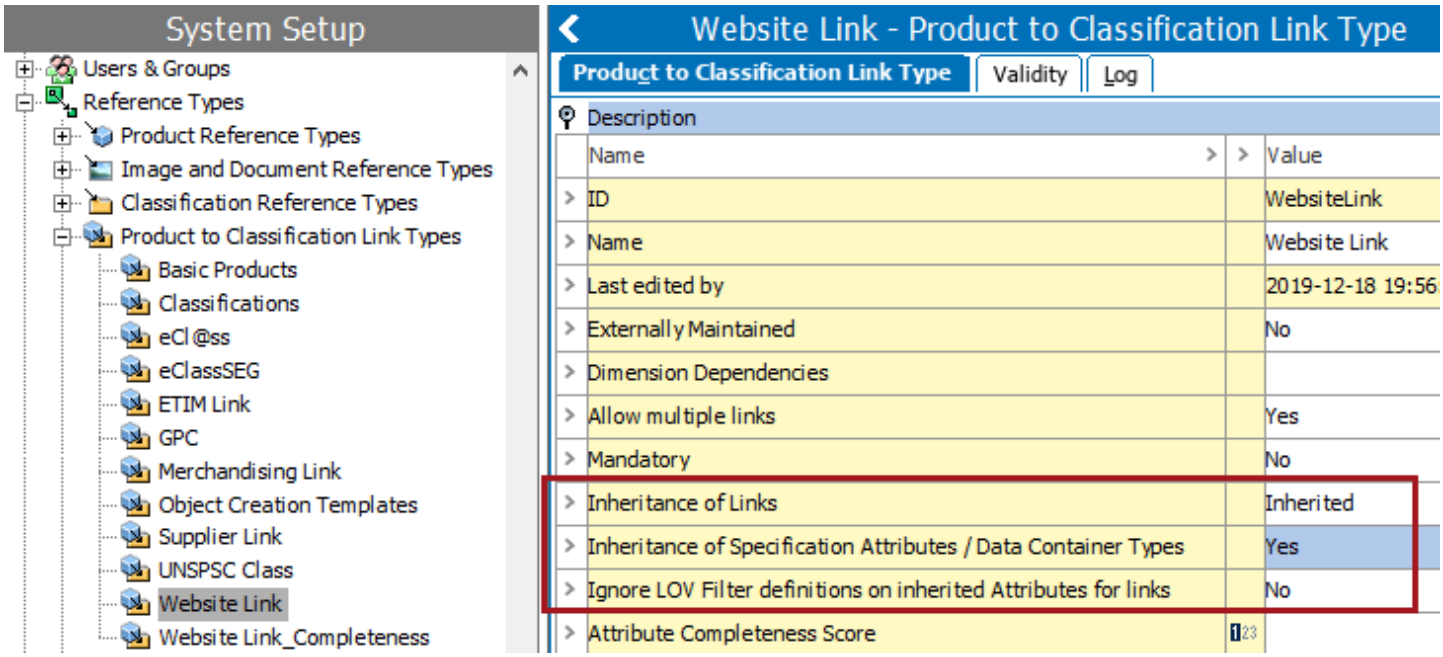
This shows that while at any point in the hierarchy a further filter may be applied, the available values for a level are based on the selection made at the higher level in the hierarchy. For example, you cannot reintroduce the value of 'Rust' at this lower level because it has already been filtered out at the top level of the hierarchy.

The result of these filters are seen in Tree by comparing the LOV values available on products within the filtered 'Wedding' and unfiltered 'Party Supplies' nodes.



Filtering on Attribute and Classification Hierarchy

Filtering on classification hierarchies works in the same way as the product hierarchy. However, configuration on the Product to Classification Link Type is required to define the behavior.



The parameters identified control the effects of the LOV filtering applied:

- **Inheritance of Links** - set to either Inherited or Accumulative.
- **Inheritance of Specification Attributes / Data Container Types** - set to Yes.
- **Ignore LOV Filter definitions on inherited attributes for links** - set to No to enable LOV filtering on the Classification for the selected Product to Classification Link Type.

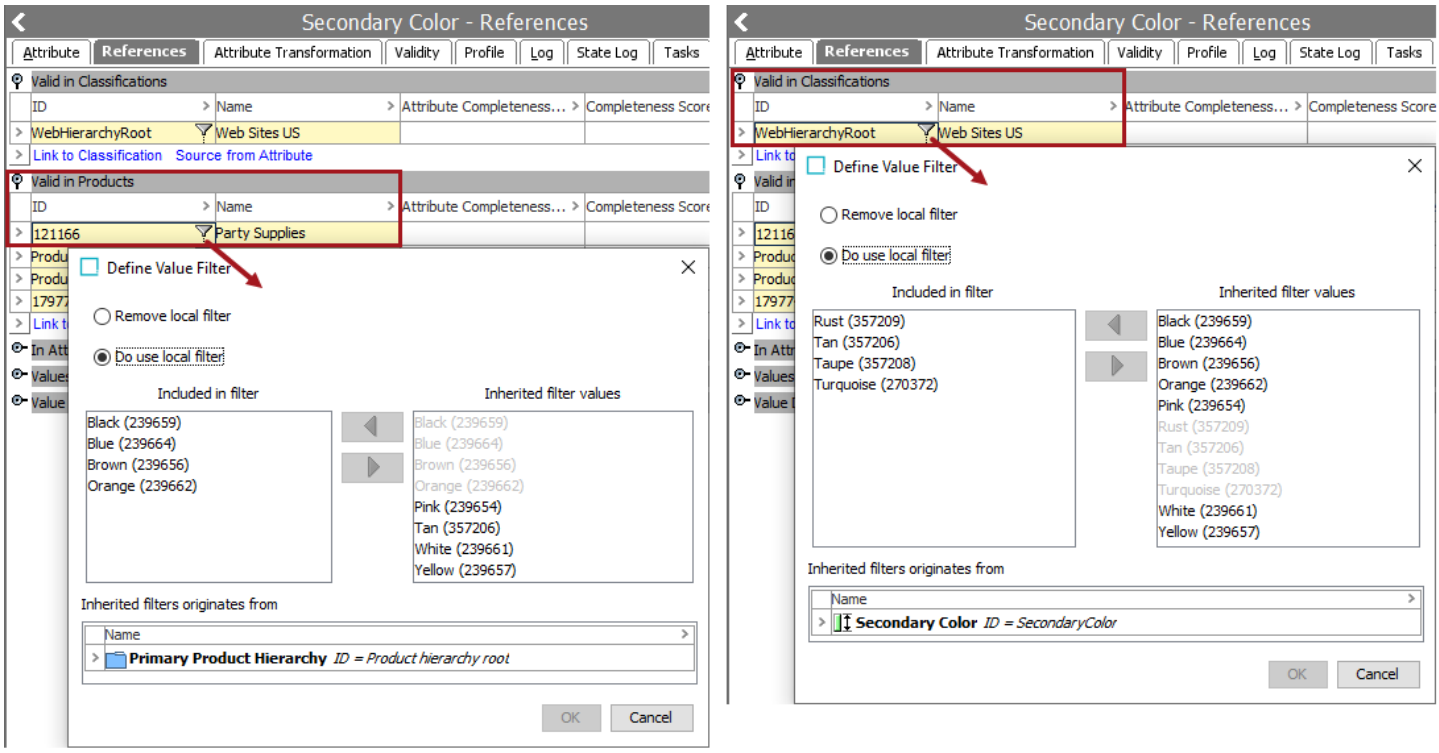
For more information on these parameters, see the **Product to Classification Link Type - Advanced** topic in the **Reference and Link Types** section of this guide.

Combining PPH and Classification Hierarchy LOV Filtering

Although combining LOV filtering on the PPH and on the classification hierarchy is possible, it is not recommended because it is difficult to maintain.

Important: Avoid setting a filter on both the PPH and also on a Classification since invalid LOV values can be added. Filtering that is applied on the PPH is not displayed and cannot be disabled when applying filters in the classification hierarchy.

To illustrate this issue, in the image below, the only valid values on the PPH (left image) are Black, Blue, Brown, and Orange. However, when setting the classification filter (right image), all of the LOV values are available to set as a local filter. In this example, four invalid values have been selected on the Classification filter. For the user enriching data in Tree, the PPH filter overrides the Classification filter settings.



Important: When LOV filtering is desired for an attribute, recommended practice is to choose to filter on either the PPH or on the Classification, but do not filter on both.

Filtering LOV Values

When setting a List of Values filter, it is possible to define which values are valid for a certain attribute.

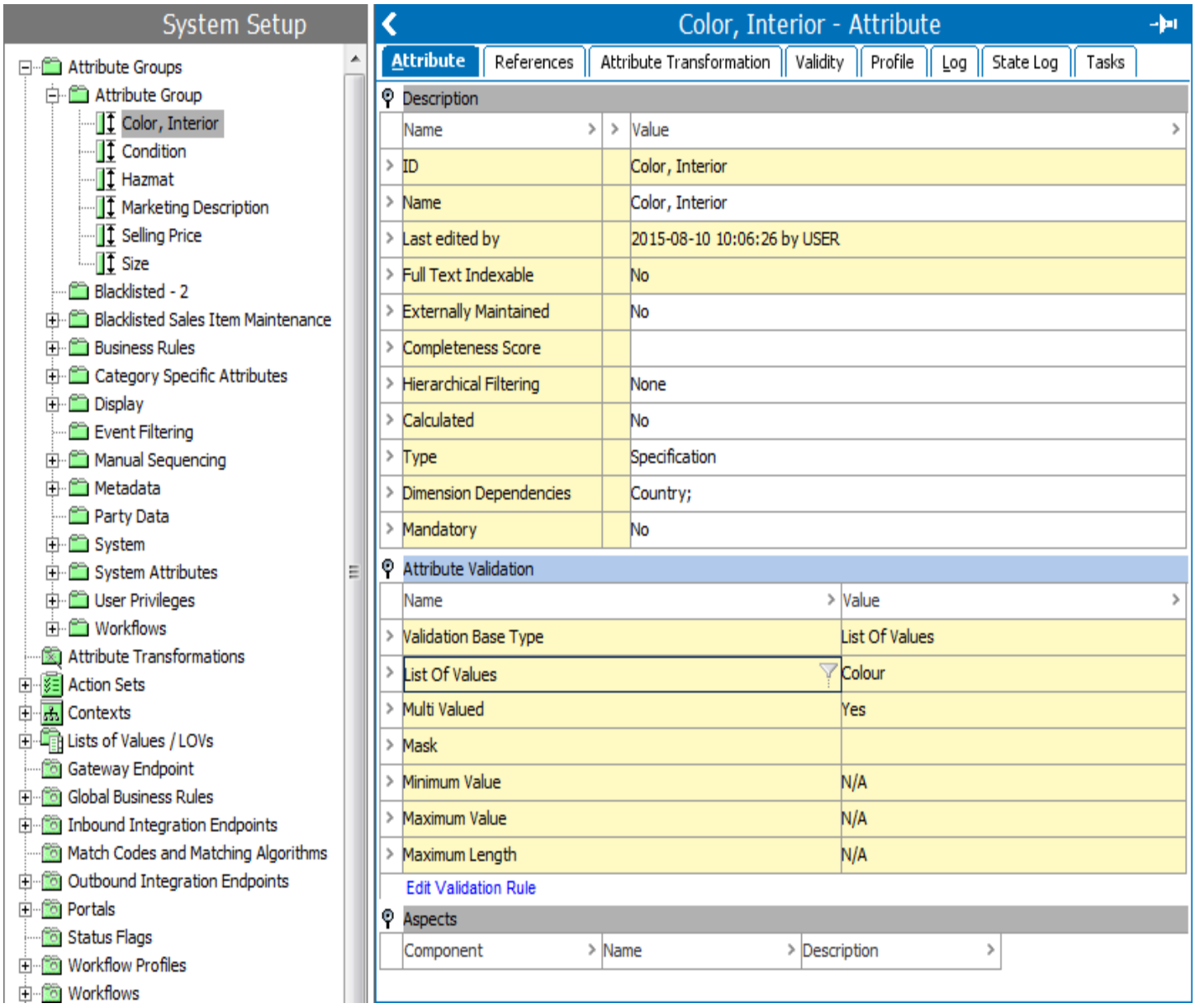
This is useful when you want to gather the information in one place and use only one LOV containing all the values, and at the same time be able to differentiate the values available for different attributes. As shown in the following table, a single LOV named 'Color' includes five values, but only three are allowed for each of the two attributes.

'Color' LOV Values	Legal Values for Attribute 1	Legal Values for Attribute 2
• Green • Red • Yellow • Blue • White	Filter Includes • Green • Yellow • White	Filter Includes • Green • Red • Blue

Important: Before filtering an LOV, review the **Considerations** section of the **Filtering LOVs** topic.

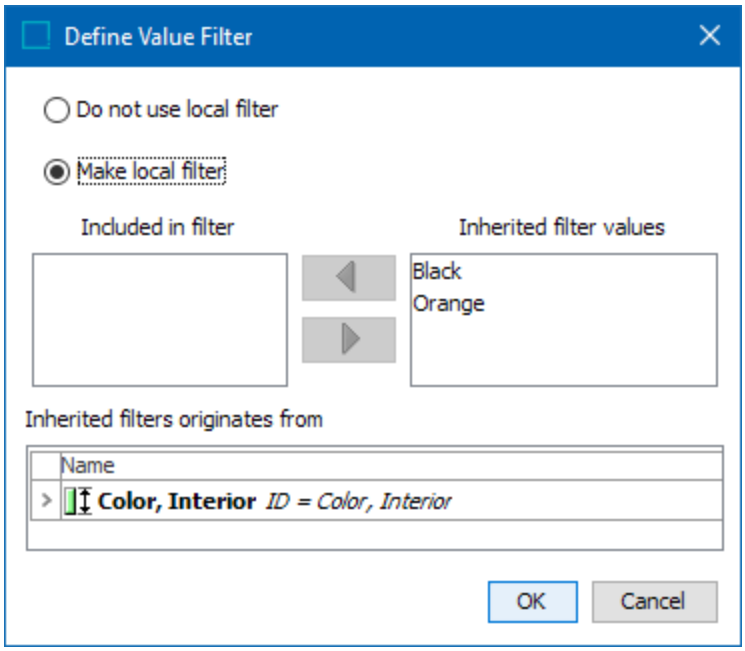
To filter LOV values based on an attribute:

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



2. Click the **Attribute** tab.
3. Under the **Attribute Validation** flipper, click the filter icon ( ) in the **List of Values** field to display the 'Define Value Filter' dialog.

> **Note:** To modify filters you must have privileges to maintain attributes.



4. Click the **Make local filter** radio button.
5. Move the values to be included in the filter to the 'Included in filter' list, as follows:
   - Double-click a value in the 'Inherited filter values' list.
   - Select a value in the 'Inherited filter values' list and click the left arrow button ( ).
6. Click **OK**. Values in the 'Included in filter' list are available in the relevant editor (product and/or classification) displaying the attribute (both specification and description). For more information, see the **Attributes** topic.

Filtering by Hierarchy on LOV Attributes

Hierarchical filtering is a way of setting a filter on an LOV attribute that is used on a specific product, product family, or classification and defines which values are valid for a certain area within a hierarchy, based on references.

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

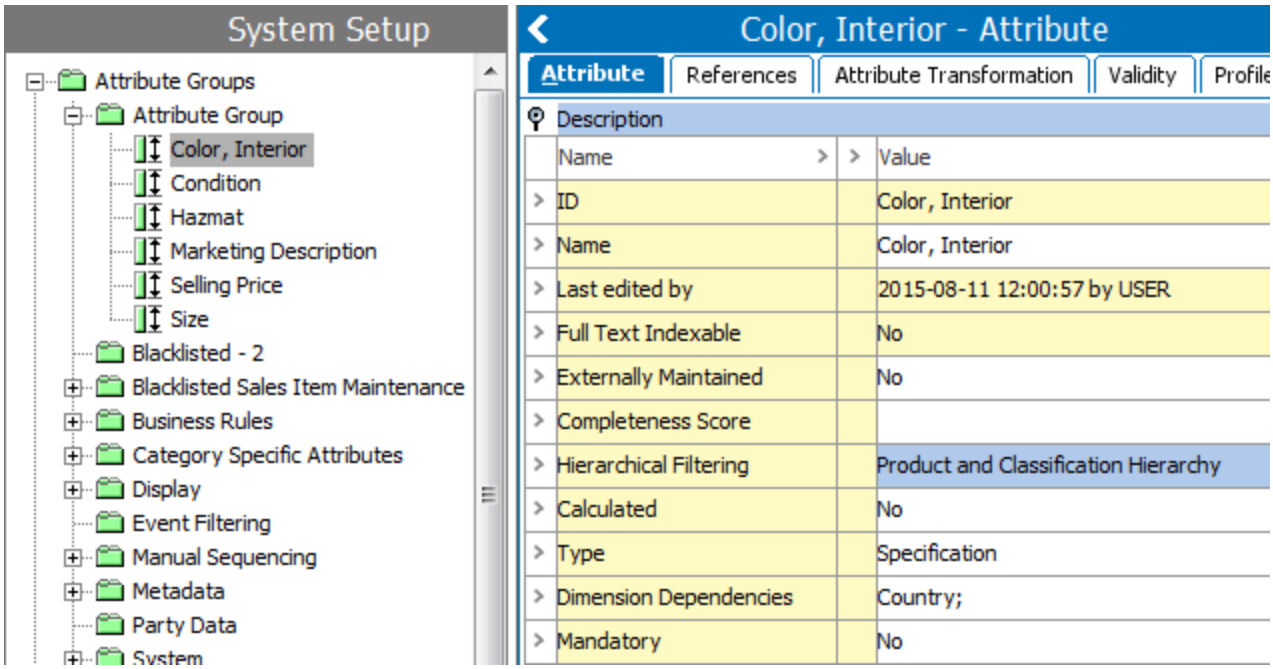
As shown in the table below, the 'Color' attribute uses the 'ColorLOV' but is filtered differently based on the location in the hierarchy. For the plastic product node of the product hierarchy, the ColorLOV allows three values, while the wood products node allows two different values.

'ColorLOV' Values	Legal Values for the 'Color' attribute on the Plastic Products Node	Legal Values for the 'Color' attribute on the Wood Products Node
• Green • Red • Yellow • Oak • Mahogany	Filter Includes • Green • Red • Yellow	Filter Includes • Oak • Mahogany

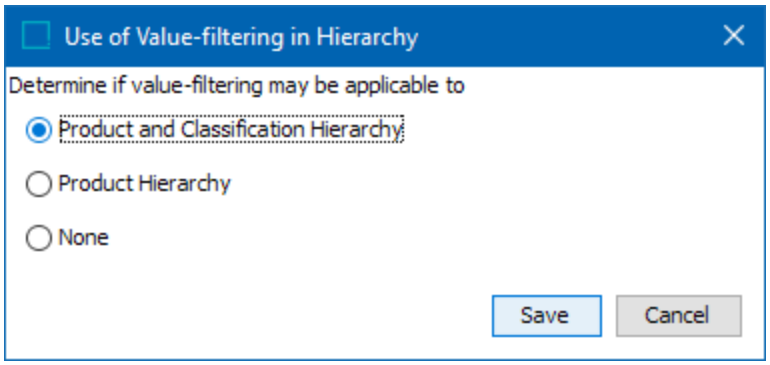
Important: Before filtering an LOV, review the **Considerations** section of the **Filtering LOVs** topic.

Filter an LOV Based on Hierarchy

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



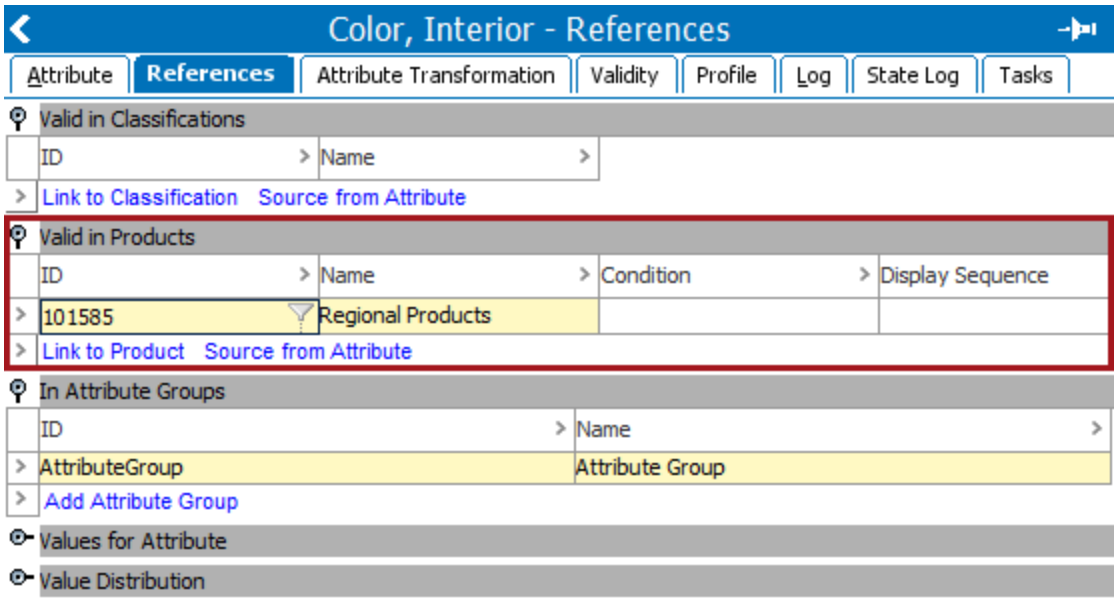
2. Click the **Attribute** tab.
3. Under the Description flipper, double-click the **Hierarchical Filtering** parameter to display the 'Use of Value-filtering in Hierarchy' dialog.



4. Select a radio button as follow:
   - **Product and Classification Hierarchy** - to display a filter option on the References tab for the 'Valid in Classifications' and 'Valid in Products ' flippers.

     Avoid hierarchical filtering on product and classification hierarchy where possible since it can negatively affect performance. For more information, see the **LOV Filtering Recommendations** topic in the **System Administration** online help.
   - **Product Hierarchy** - to display a filter option on the References tab for only the 'Valid in Products ' flipper.

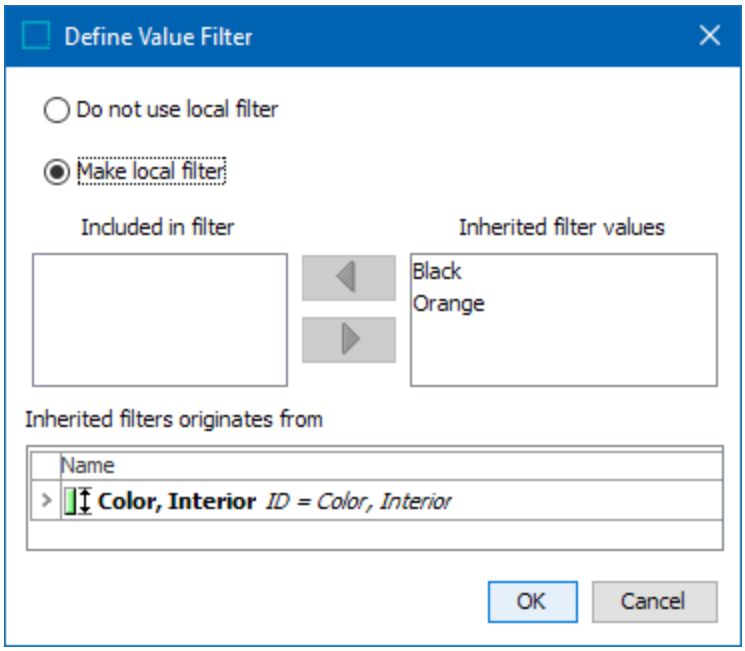
5. Click **Save**.
6. Click the **References** tab.



7. Under the **Valid in Products** flipper (shown above) and/or the **Valid in Classifications** flipper (based on your selection in the 'Use of Value-filtering in Hierarchy' dialog above), click the filtering icon ( ) in the field containing the relevant product and/or classification.

   Note: To modify filters you must have privileges to maintain attributes.

   The **Define Value Filter** dialog displays showing the values from the LOV under the 'Inherited filter values' list.

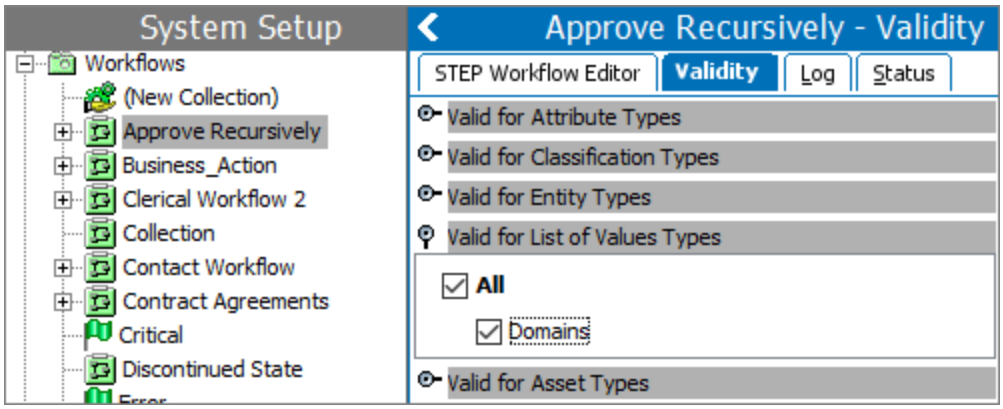


8. Click **Make local filter** radio button.
9. Move the values to be included in the filter to the 'Included in filter' list, as follows:
   - Select a value in the 'Inherited filter values' list and click the left arrow button ( ).
   - Double-click a value in the 'Inherited filter values' list.
10. Click **OK**. Values in the 'Included in filter' list are available in the relevant editor (product and/or classification) displaying the attribute (both specification and description). For more information, see the **Attributes** topic.

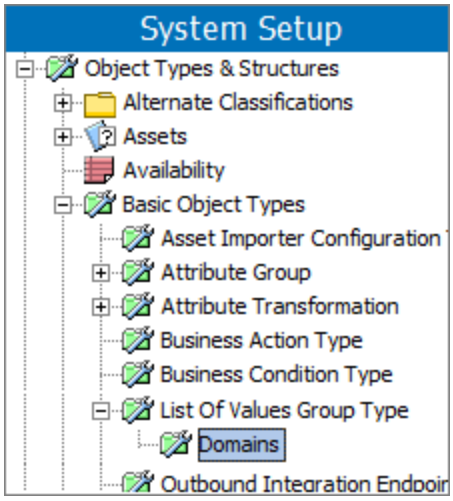
Workflows and LOVs

An LOV can be initiated into and/or removed from a workflow the same way as products, classification, entities, and assets. 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 must be valid for the LOV object type 'Domains' as shown in the image below.



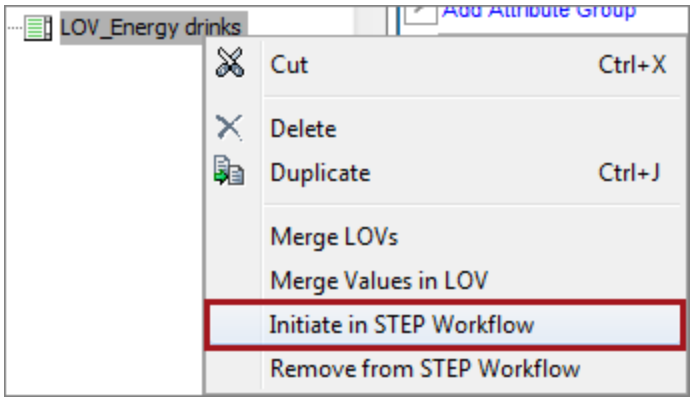
Domains are the only object type for LOVs and are found in workbench in System Setup > Object Types & Structures > Basic Object Types > List Of Values Group Type (as shown in the image below).



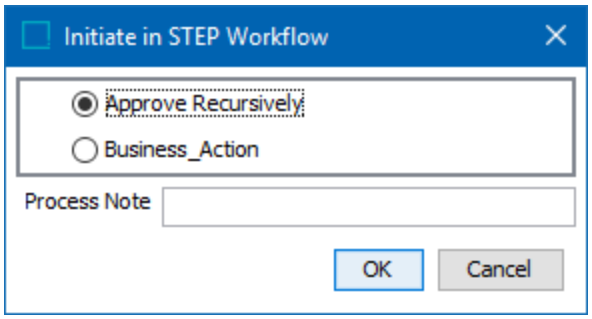
Initiate an LOV in a Workflow

Below are steps to initiate one LOV to a Workflow, however more than one LOV can be initiated to the same workflow.

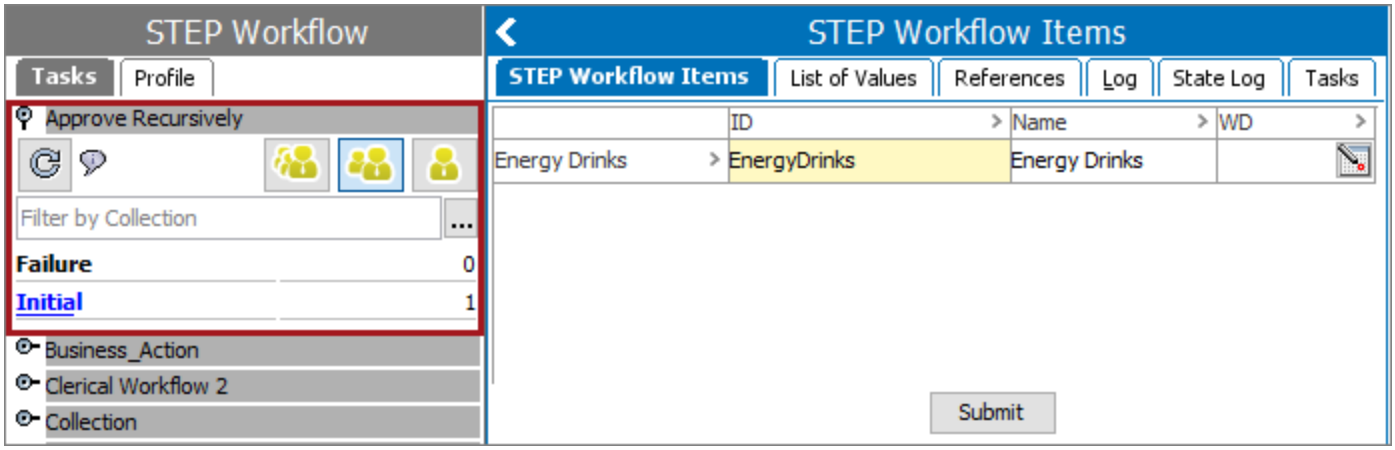
1. Right-click on an LOV, and select 'Initiate in STEP Workflow.'



2. The Initiate in STEP Workflow dialog will display listing only those workflows configured with validity for Domains (LOVs). 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 and selecting the workflow state. The workflow editor will display. Click the STEP Workflow Items tab and the LOV will display.

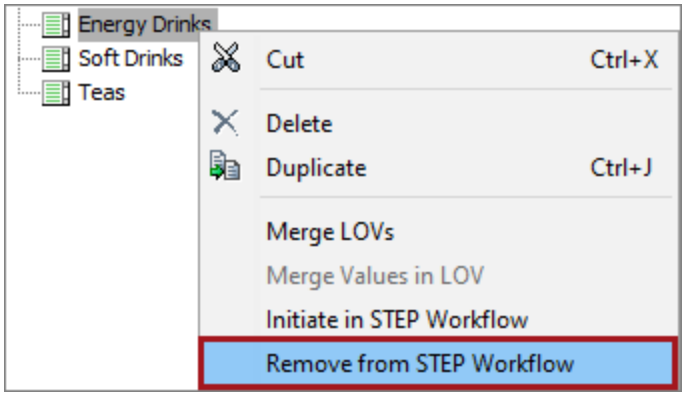


For more information, see the **Working with Tasks in Workflows** section of the **Workflows** documentation.

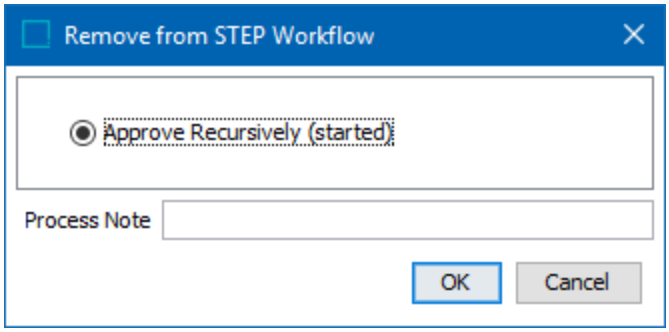
Remove an LOV from a Workflow

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.

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.

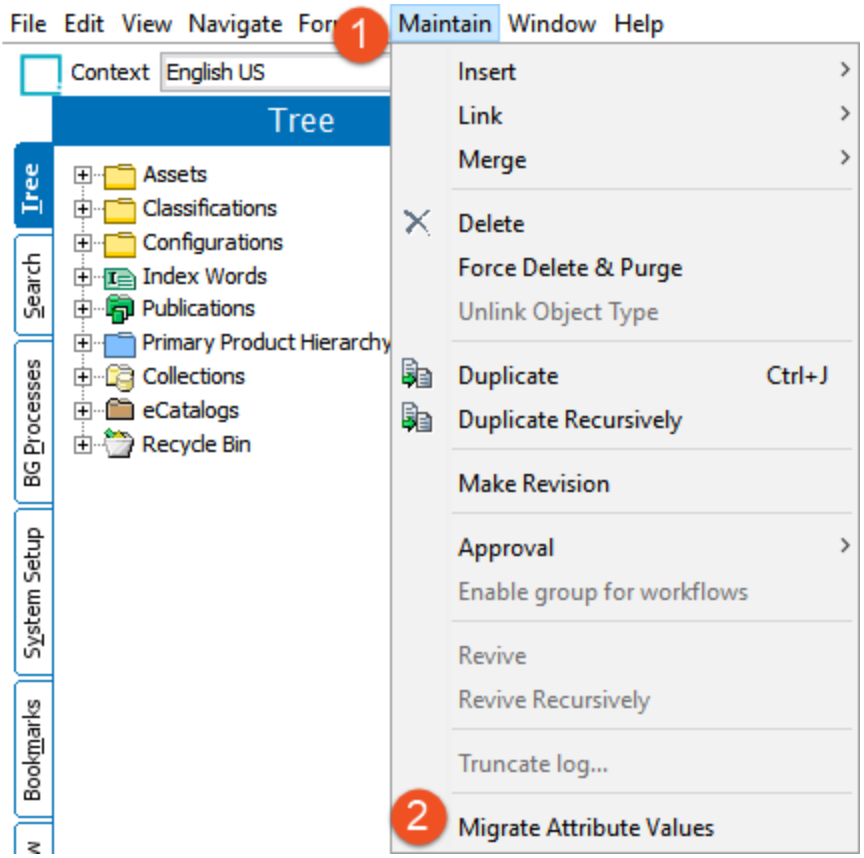
This section discusses how to create an attribute and then modify it based on your requirements. You will find information including setting validity, using inline references for values, and converting for certain scenarios.

For information on maintaining LOV attributes, see the **List of Value (LOVs)** section.

Attribute Value Migration

With STEP 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 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.

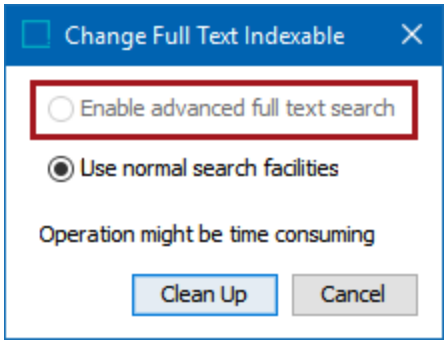


Prerequisites

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. If you go to the 'Full text indexable field' and it is set to Yes, then full text index is enabled. If it is set to No, double-click the 'Full text indexable' field. If the 'Enable advanced full text search' option is disabled in the dialog that appears, the index is not enabled for your system. For information on enabling, refer to the 'Important' information in the **Full Text Indexable Attributes** topic.



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 / Super User Guide** documentation.

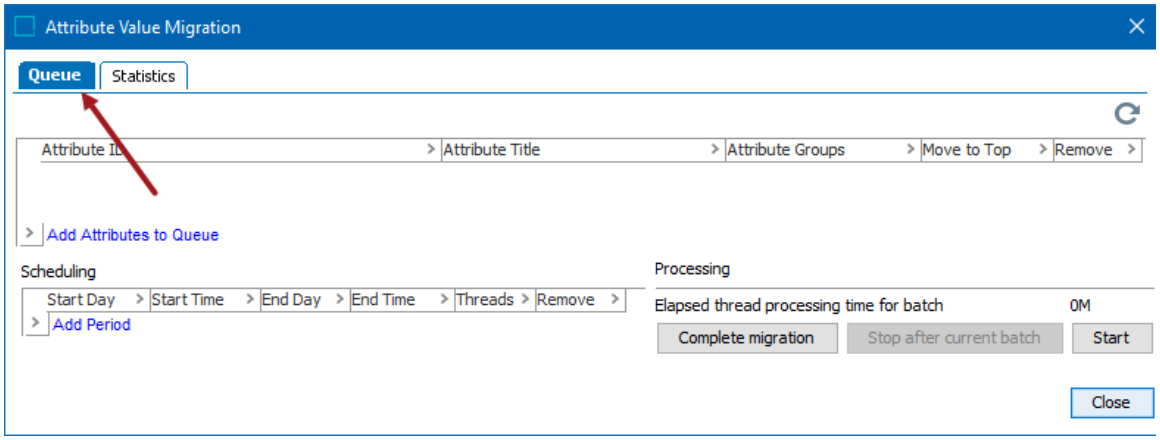
4. Shrinking 'value7' – Due to the 'value7' table having a function-based index, the shrink space approach used for 'valuemap' and 'valuemodel' cannot be used while STEP is running. Thus, when all values have been migrated, it is possible to schedule the required 'value7' operations for next system restart. What will happen 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.

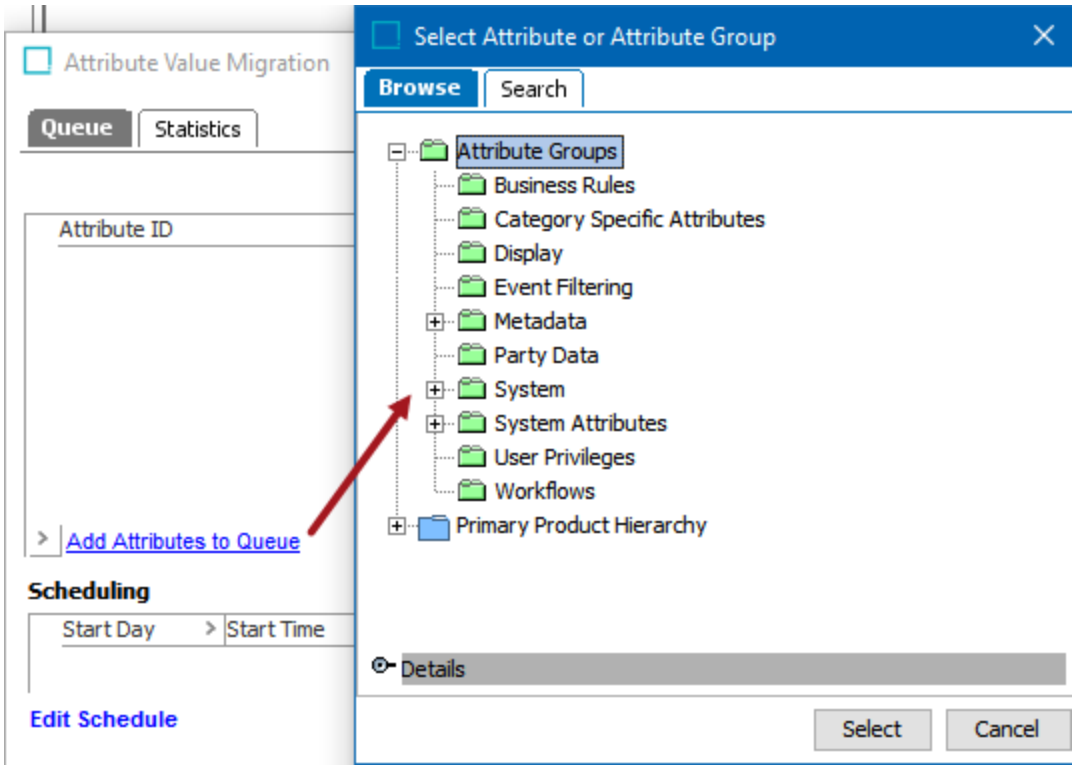
Queuing Attribute Values for Migration

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.

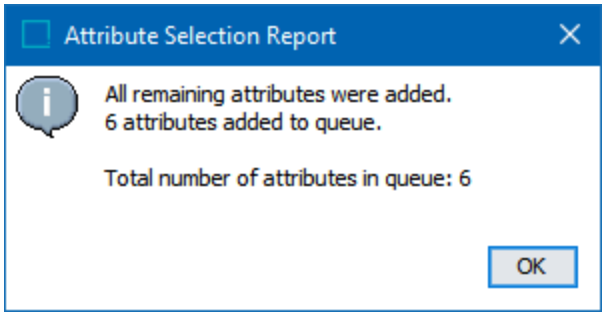


2. Click the **Add Attributes to Queue** link. A list of the migratable attributes will appear. Select **OK** after choosing the desired attributes to add to the queue.

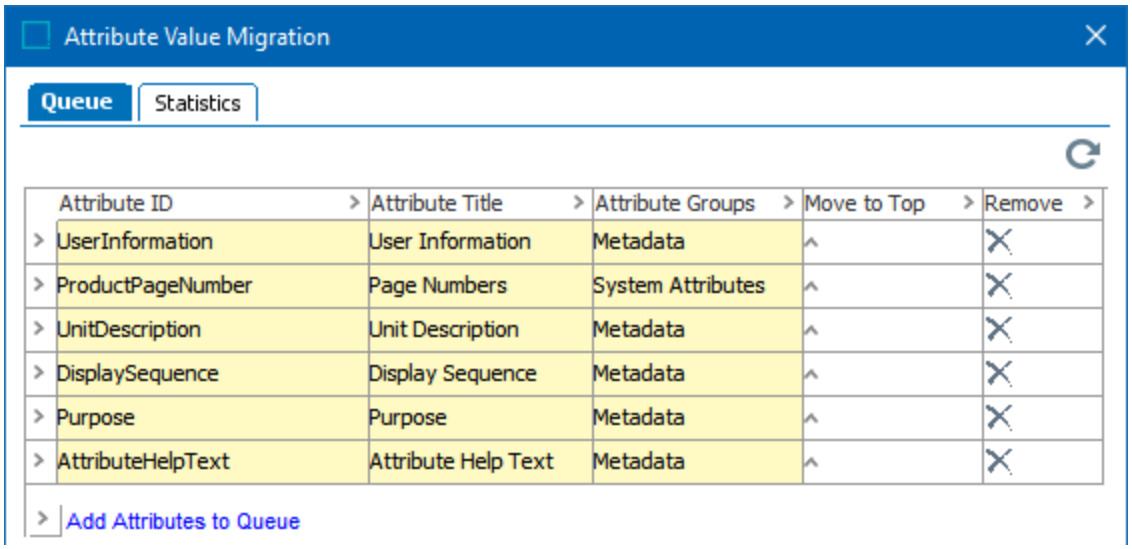
> **Note:** If an attribute group is selected, all attributes belonging to that group will be added to the queue. In this example, 'Attribute Groups' is selected, therefore, all the groups below it, as well as all the attributes under each attribute group, will be included.



A report of the number of added attributes will appear.



The attributes will be added to the queue.



As described above, it is possible to only queue a subset of the attributes that can be migrated. It is also possible to remove attributes from the queue. The reasons for having these options are:

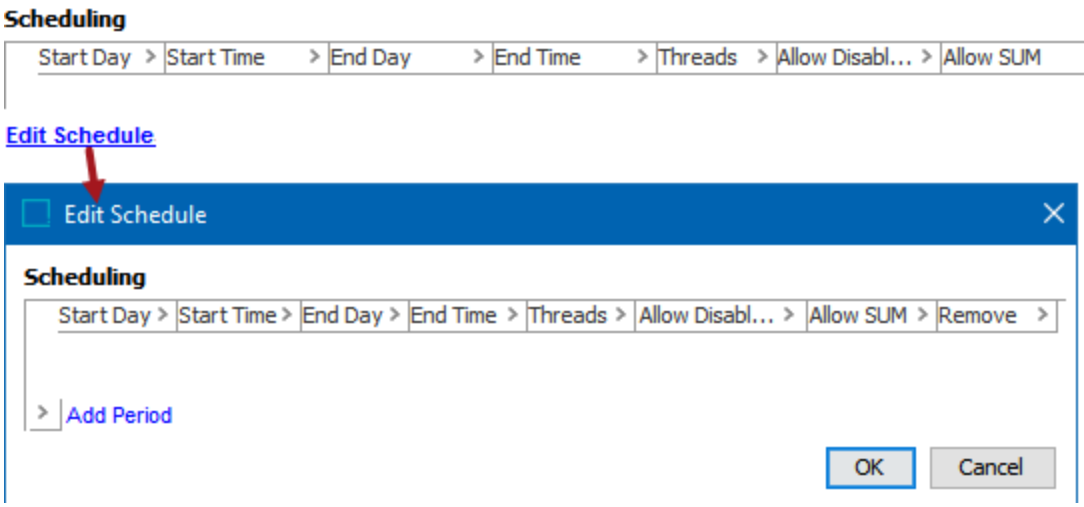
1. The CSV model requires different SQL API views to be used for querying values meaning that for customers using the SQL API it must be possible to postpone the migration of values for certain attributes until the SQL has been updated. For details, see the SQL API documentation by clicking the **STEP API Documentation** button on the Start 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

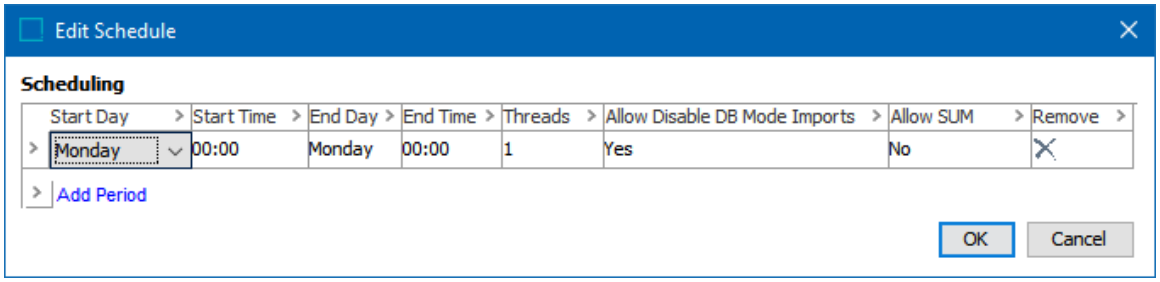
Before scheduling the migration and following the next steps, review the **Scheduling Considerations** section below.

Important: Consider the time zone of the application server compared to that of the workbench (the client) where the schedule is created or viewed. When scheduling a job, the local time zone is displayed in the workbench, but the time zone of the server is used to run the background process. Although displayed, the time zone of the client is not included in the instruction to the server to run the job. This can cause confusion about when the job will run since the scheduled time is not automatically converted to accommodate potential differences in time zones.

1. From the scheduling section of the 'Attribute Value Migration' dialog, select **Edit Schedule**. The 'Edit Schedule' dialog will appear.



2. Select **Add Period**, and a default time will be added. Configure by clicking on each option and selecting from the dropdown menu.

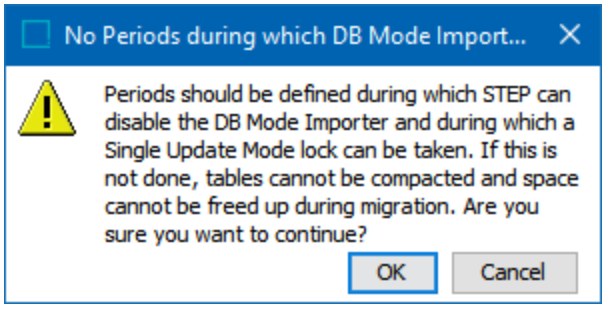


- **Start Day** - The day the scheduling period starts
- **Start Time** - The time on the starting day that the period starts
- **End Day** - The day the scheduling period ends
- **End Time** - The time on the ending day that the period ends
- **Threads** - How many threads the migration process may 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.

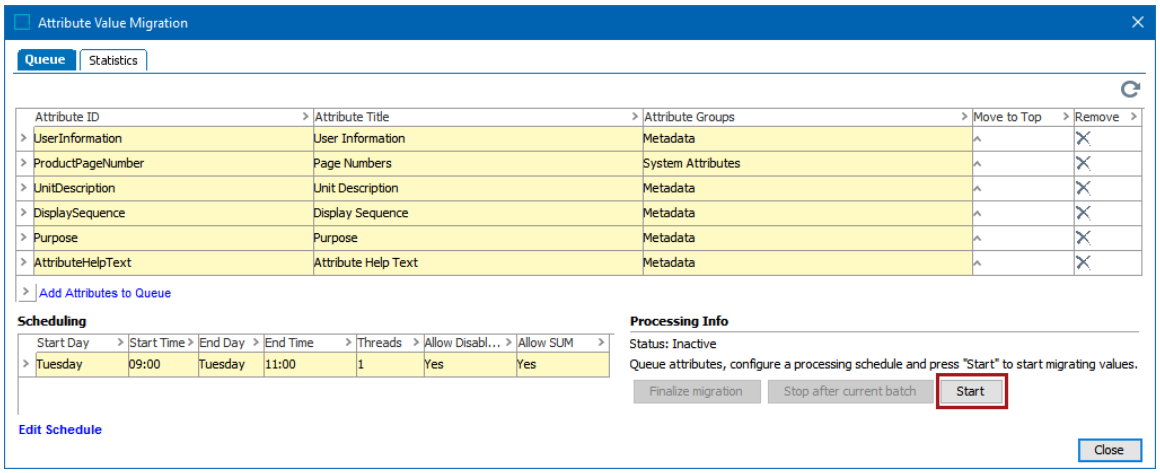
- A warning will be shown if no periods are defined where the process is allowed to disable the database mode importer and/or take a SUM lock. It is not possible to complete the migration without having defined these schedule periods.



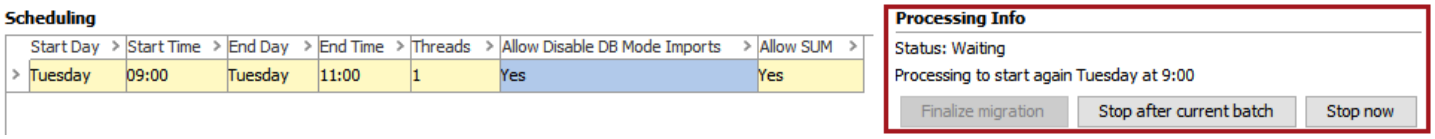
- The migration process will only run on a single application server in a clustered setup and this may be taken into account when specifying the number of threads for a processing period. How many threads to use depends on the system hardware and the system utilization. Typically one or two threads will 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**.

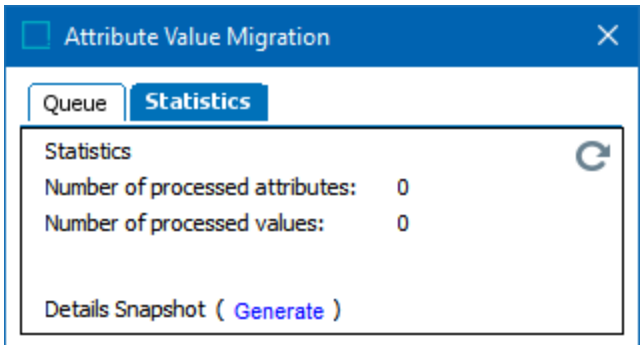


Your migration is now scheduled, and the process will begin at the earliest set period.

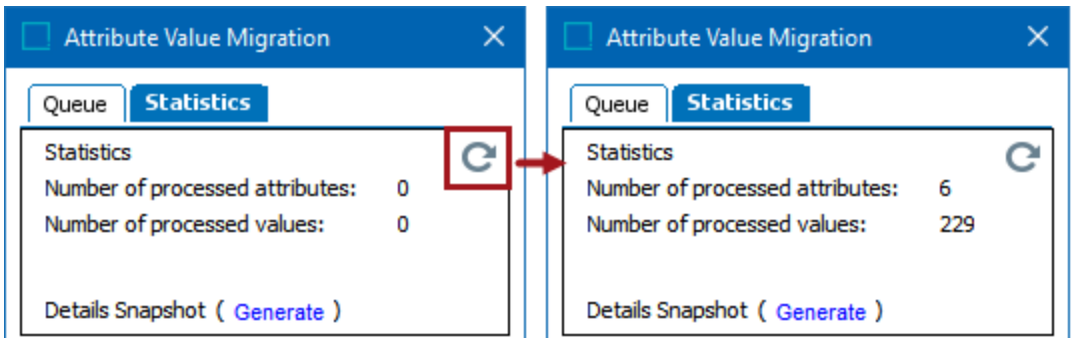


Statistics

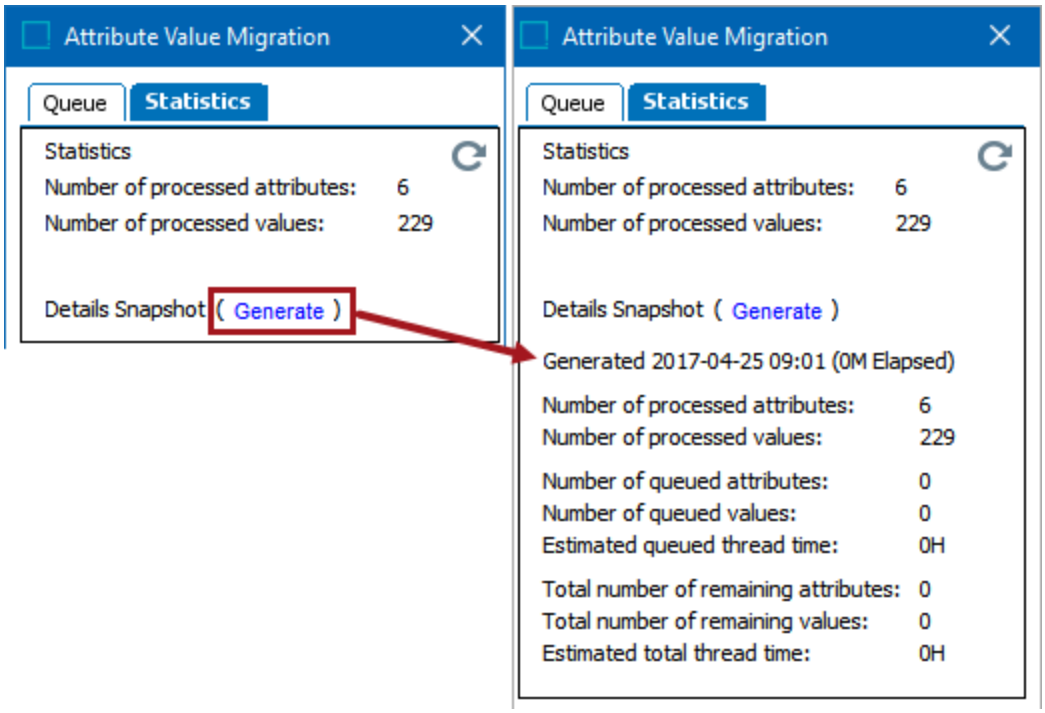
The Statistics tab shows details about the migration. The following image shows how the Statistics tab looks before migration has begun.



- **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

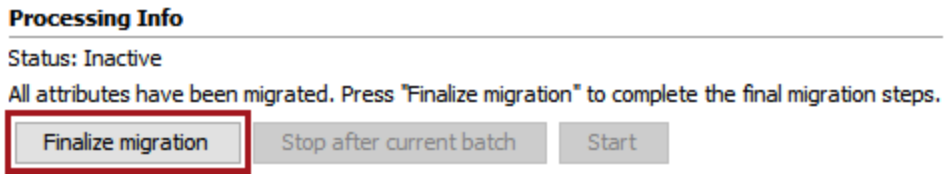
After the values for all migratable attributes on the system have been migrated, the final steps for the migration are ready to be initiated. This status can be noted on the 'Processing Info' section of the 'Attribute Value Migration' dialog.

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.

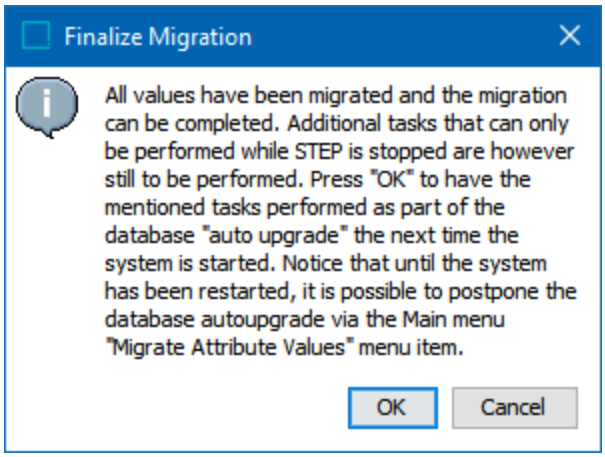
Important: The final processes of the migration require that the STEP system is restarted.

1. Click **Finalize migration** to generate a prompt noting that the migration is ready to be completed.

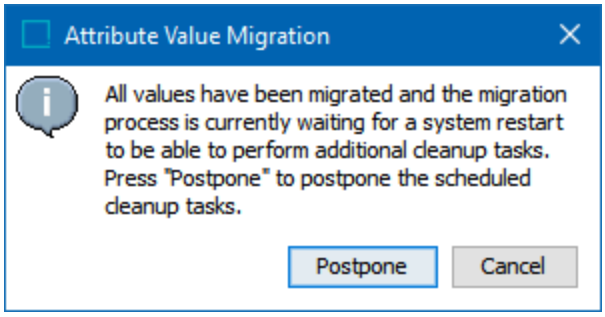


2. Press OK to schedule the final operations of the migration for the next system restart.

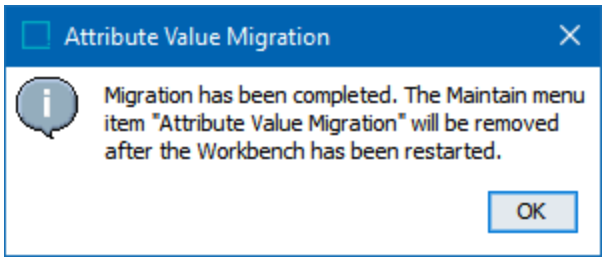
Important: The system may be offline for an extended period of time while the processes finish migrating. Plan for a period of downtime before restarting STEP.



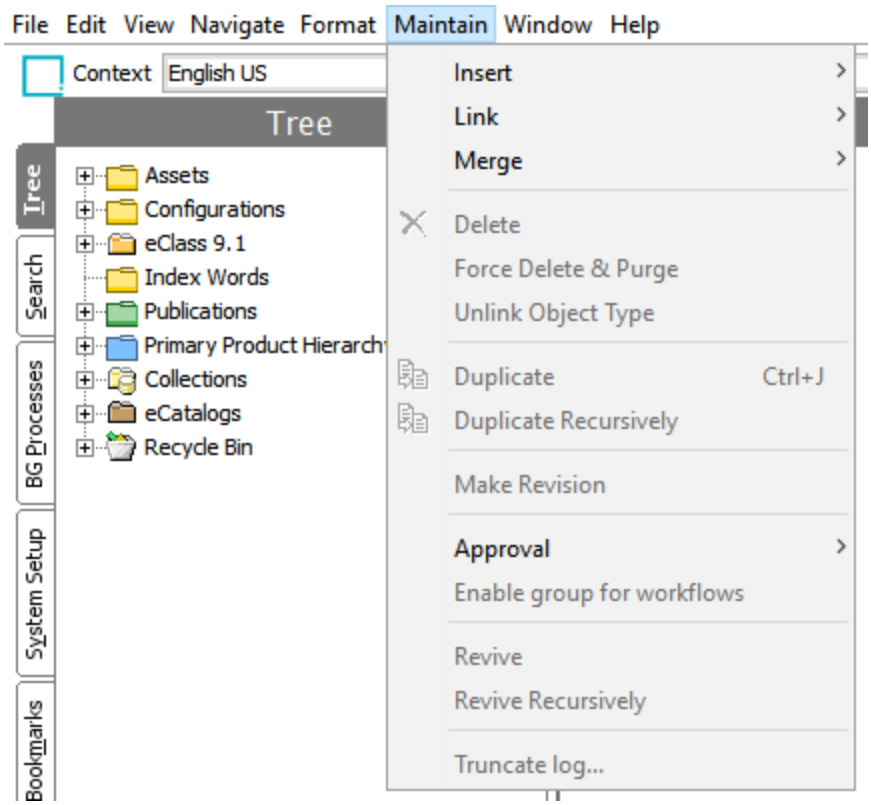
Should you need to postpone completing the migration for any reason, navigate to the **Migrate Attribute Values** option under the **Maintain** menu item. A dialog box will offer the option to postpone the final steps of this migration. Selecting this option will require you to finalize the migration 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.



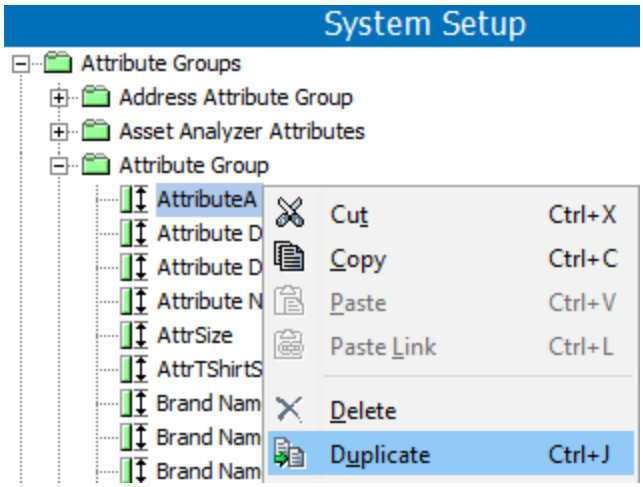
Changing an Attribute ID

The 'Switch Attribute IDs' option allows the user to switch the ID of two attributes. This functionality avoids entering single update mode for changes to attribute definitions (such as externally maintained, full-text indexable, multi-valued, data type, or LOV). Be aware that this method results in the loss of revision history on values of the attribute being changed.

Note: Removing a dimension dependency requires the use of a bulk update operation in a relevant context to repetitively move attribute values from relevant dimension points to the target attribute. This will then implement the prioritization of values. For more information, see the **Attribute Values: Merge Attribute Values Operation** topic in the **Bulk Updates** documentation.

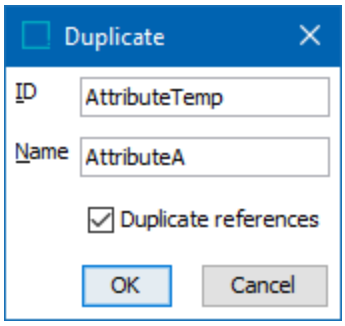
The following example shows switching the ID of an original non-externally maintained attribute (ID = AttributeA) that contains values, with a new externally maintained version of the same attribute via the Switch Attribute IDs option.

1. In System Setup, select the original attribute, right-click and choose the **Duplicate** option to make a copy of the attribute (i.e., same object validity and attribute links).

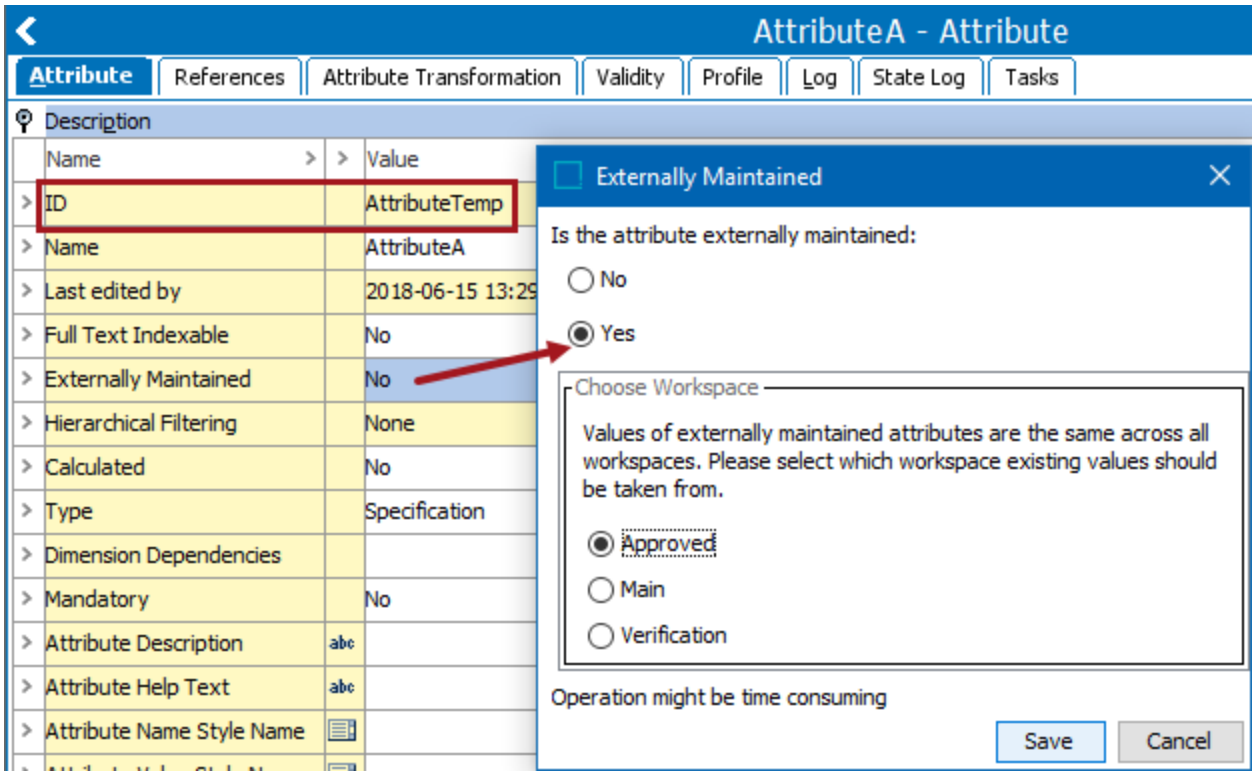


2. On the Duplicate dialog, check the '**Duplicate references**' checkbox, add a unique **ID**, set the **Name** to be the same as the original attribute, and click **OK**. This example shows 'AttributeTemp' as the ID.

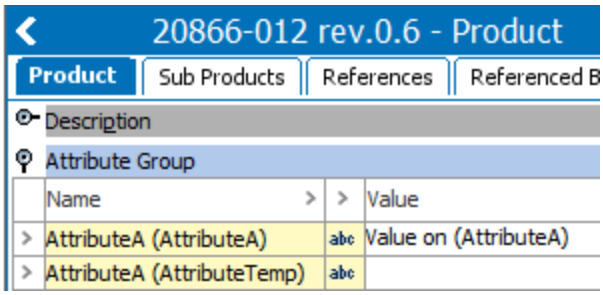
Note: Attribute IDs are case sensitive.



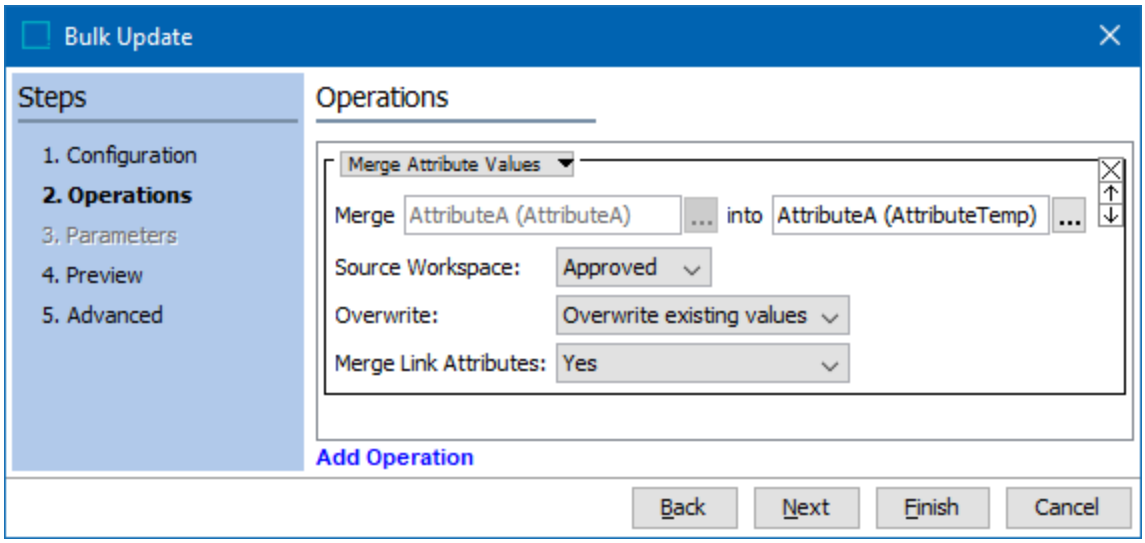
3. Select the new attribute (AttributeTemp), double click the **Externally Maintained** parameter, select the **Yes** radio button and the **Approved** workspace (no values exist but a selection is required), and click **Save**.



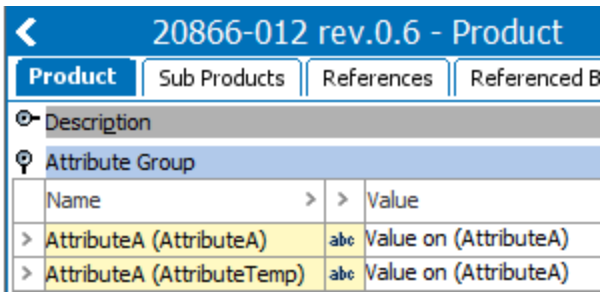
4. In Tree, verify both attributes are displayed on an object for which they are valid. Only the original attribute has a value. In this image, the ID was added into the Name field to distinguish the two attributes with the same name.



5. In System Setup, select the original attribute (AttributeA), right-click, and select the **Merge Attribute Values** option. The Bulk Update wizard is displayed. To complete the merge, follow the steps defined in the **Merge Attribute Values Operation** section of the **Attribute Values: Merge Attribute Values Operation** topic in the **Bulk Updates** documentation.

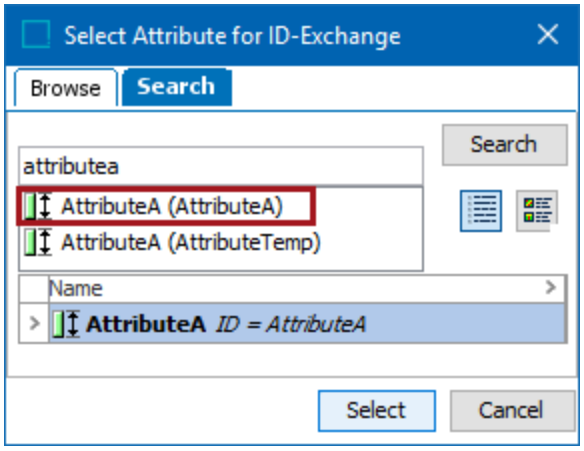


6. In Tree, on an object for which the attributes are valid, verify that value from ID AttributeA was copied to ID AttributeTemp. In this image, the ID was added into the Name field to distinguish the two attributes with the same name.



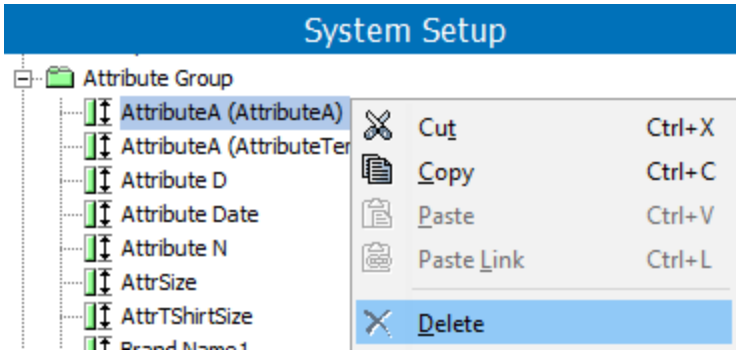
7. In System Setup, select ID AttributeTemp, right-click and choose the **Switch Attribute IDs** option to display the Select Attribute for ID-Exchange dialog.

Because the attributes have the same name, use the **Search** tab to select the original attribute, click **Search**, and click **Select**. In this example, the ID = AttributeA.

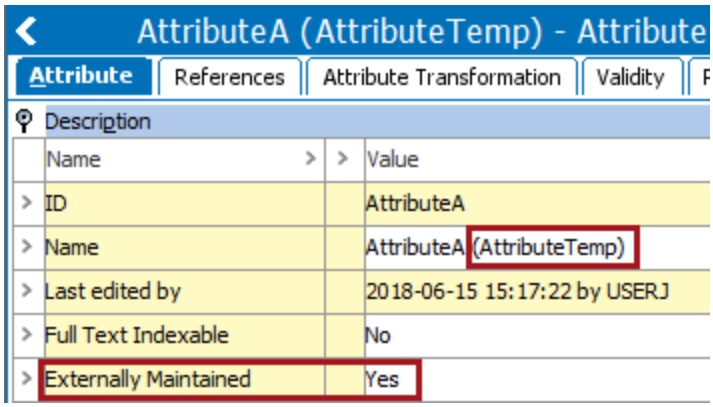


The IDs are exchanged between the two attributes.

8. Select the original AttributeA, which is not externally maintained and that now has the ID AttributeTemp, right-click, choose the **Delete** option. Use the **Force Delete** option to remove the attribute, values, and links, and bypass the Recycle Bin.



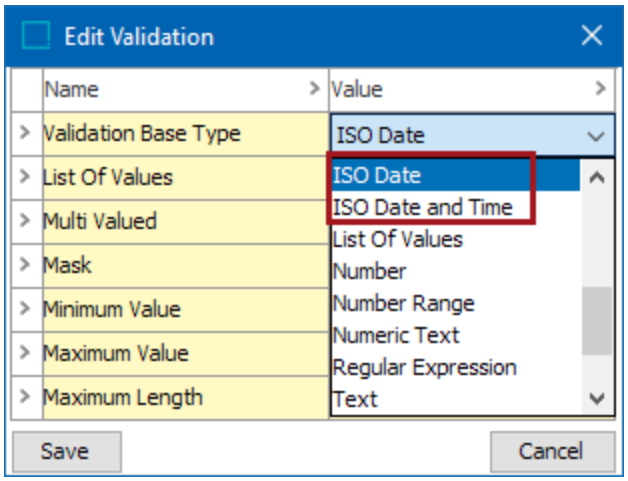
The remaining attribute with the name 'AttributeA', which started as AttributeTemp (as indicated by the original ID in the Name field), now has the values merged from AttributeA, has the ID AttributeA, and has the Externally Maintained parameter = Yes.


AttributeA (AttributeTemp) - Attribute
Attribute
References
Attribute Transformation
Validity
Description
Name
Value
ID
AttributeA
Name
AttributeA (AttributeTemp)
Last edited by
2018-06-15 15:17:22 by USERJ
Full Text Indexable
No
Externally Maintained
Yes

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. In simple terms, 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 convert 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 two 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.

For more information about switching attribute IDs, see **Changing an Attribute ID** in the **Attributes** section of the **System Setup / Super User Guide** documentation.

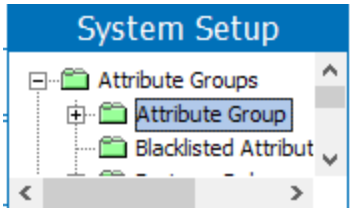
Creating Attributes

Attributes are maintained in System Setup, and are characteristics or 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.

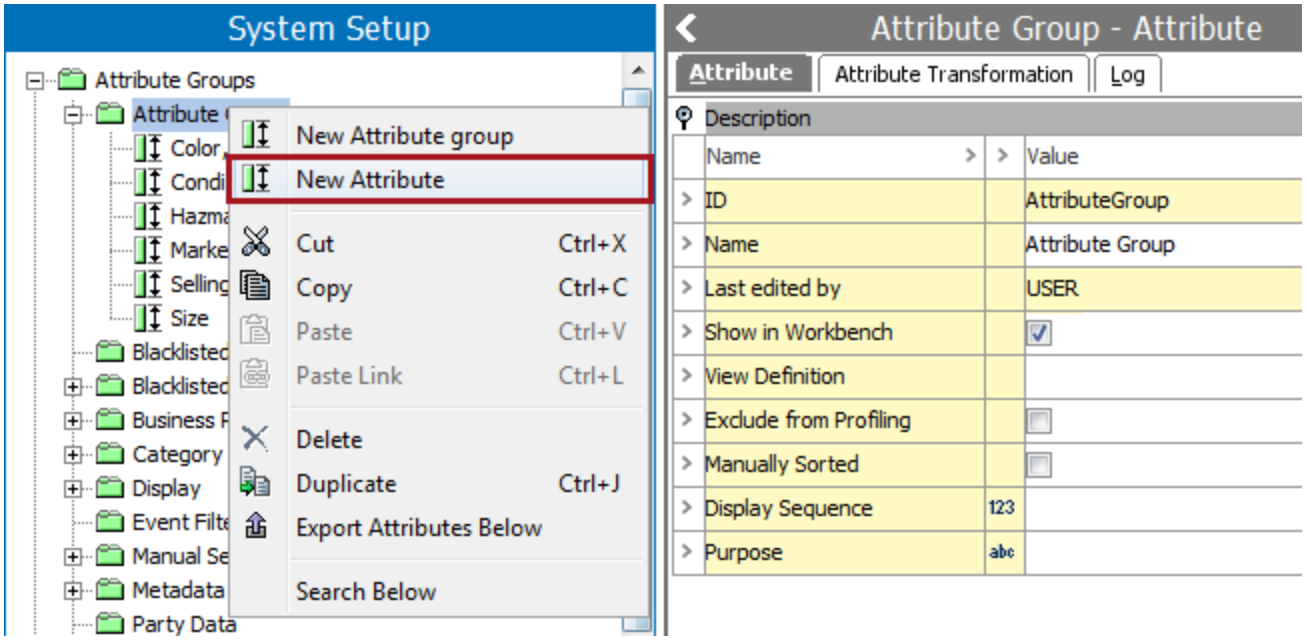
Attributes can also be created in bulk, using the export and import functionality and the Excel format. For details, see the **Managing Attribute Parameters with Excel** topic.

Create an Attribute

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.

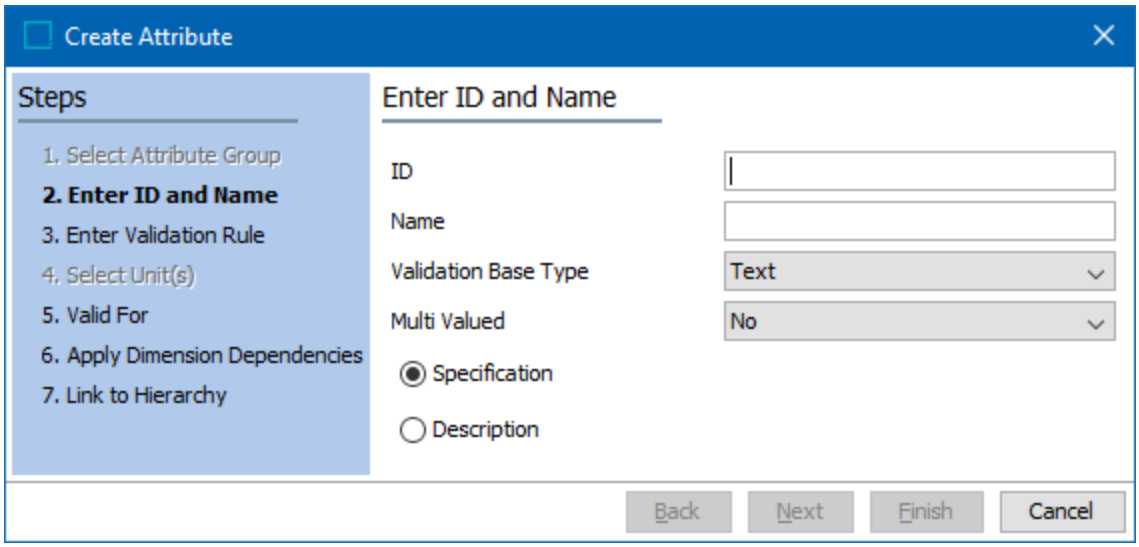
3. Type an ID and Name. See the **Attribute ID Considerations** section in this topic for more information.

Note: Attribute IDs are case sensitive.

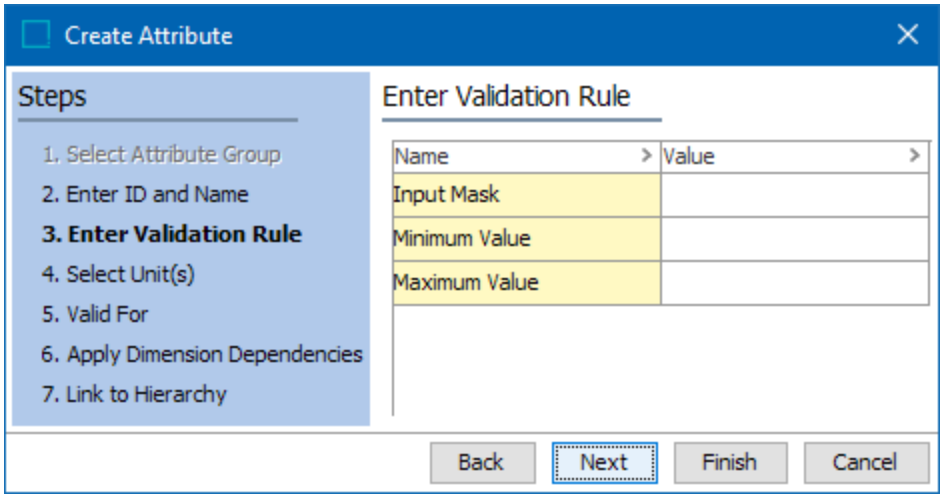
Select the sort of validation that the attribute should be based on from the **Validation Base Type** dropdown list. For a list of available validation base types, see the **Validation Rules** topic in this guide.

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

Determine whether it is a Specification or Description attribute. Click **Next**.



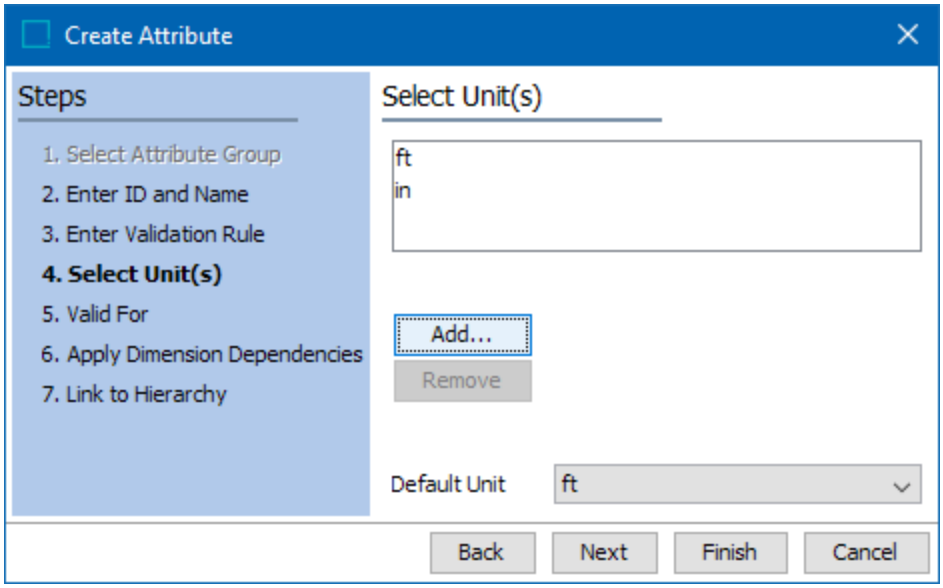
4. Type an Input Mask, Minimum Value and/or Maximum Value. Click **Next**.



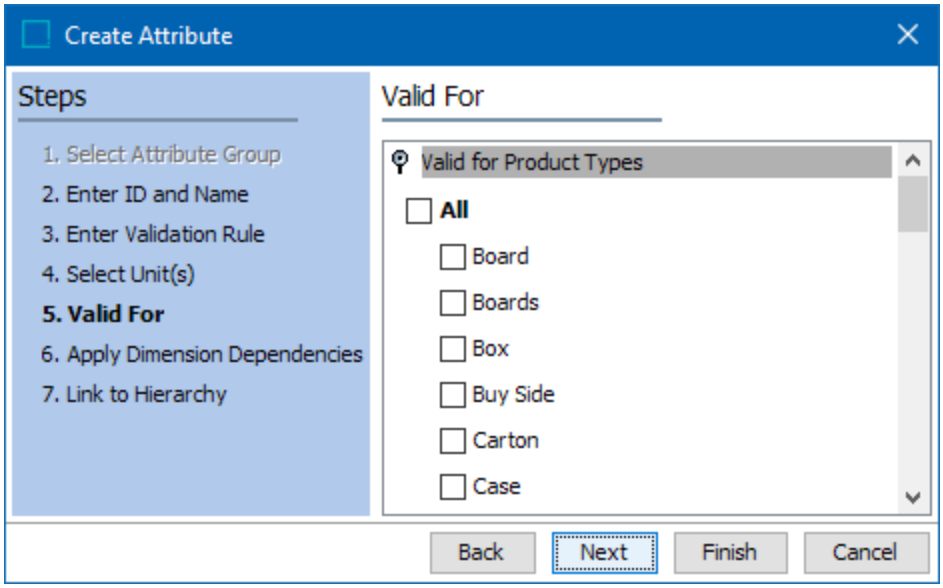
5. The **Select Units** step will only be visible if a validation base type of Embedded Number, Fraction, Integer, Number, Numeric Text, or Numeric Text (exclude tags) numeric validation base type has been selected.

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

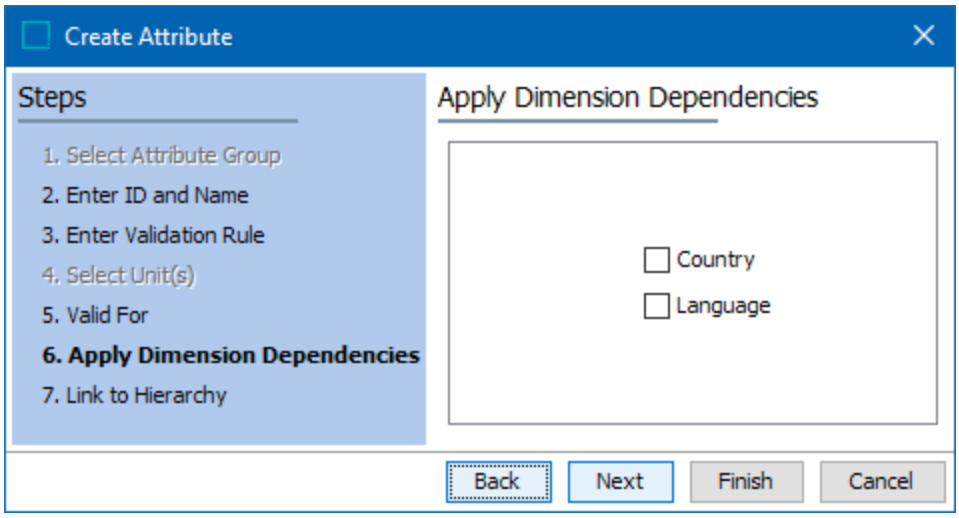
Select a unit to be selected as default unit. Click **Next**.



6. Select the product types the attribute is associated with. Click **Next**.

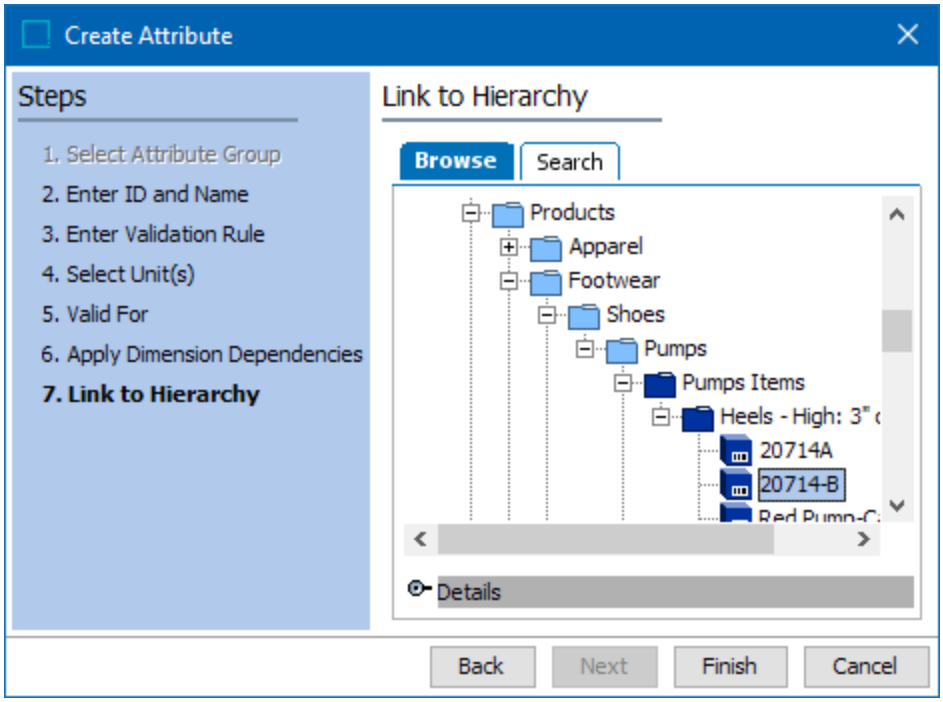


7. Apply Dimension Dependencies. Click **Next**.

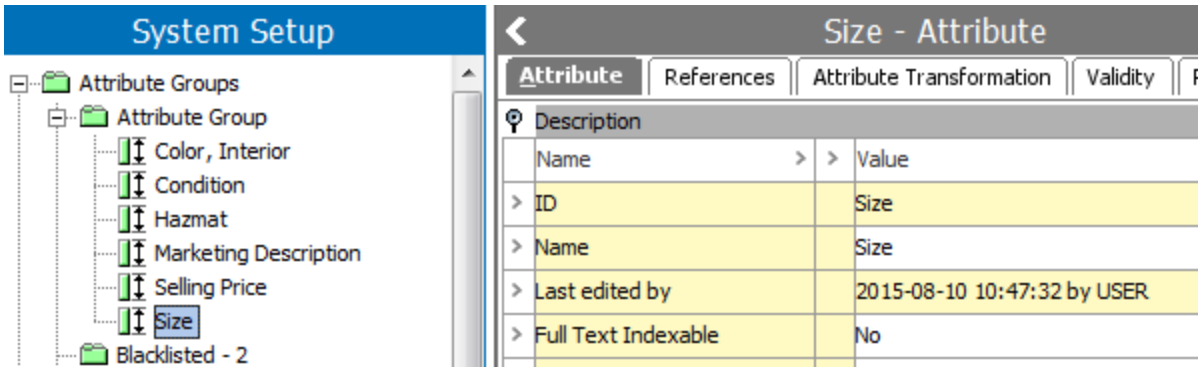


8. Select a tree node where the attribute should be linked.

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



9. Click **Finish** to create the attribute.



Note: Attributes are created as specification attributes by default. Specification attributes are only valid for Product Object Types. If an attribute should be valid for Classification Object Types, for example, it must be changed into a description attribute.

The next steps in attribute creation, as required, are to specify externally maintained, calculated, and/or specify the validity.

Attribute ID Considerations

Take the following information into consideration when creating attribute IDs for attributes that are used for Elasticsearch searches. See the **Search Screen** section for information about this functionality.

IDs cannot start and/or end with a period / dot (i.e., '.').

The following words are currently forbidden to be used as part of the ID:

- simpleValue
- lowerCaseValue
- unitId
- numeric
- dateTime

Examples of forbidden attribute IDs:

- [forbidden word], e.g., 'numeric'
- abc.[forbidden word], e.g., 'abc.numeric'
- abc.[forbidden word].abc, e.g., 'abc.numeric.abc'
- end with a dot, e.g., 'abc.'
- start with a dot, e.g., '.abc'
- dot or dots by itself / themselves, e.g., '.', '....'
- etc.

For faceted search, mappings cannot be made for these in Elasticsearch, and the attributes cannot be shown as facets.

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.

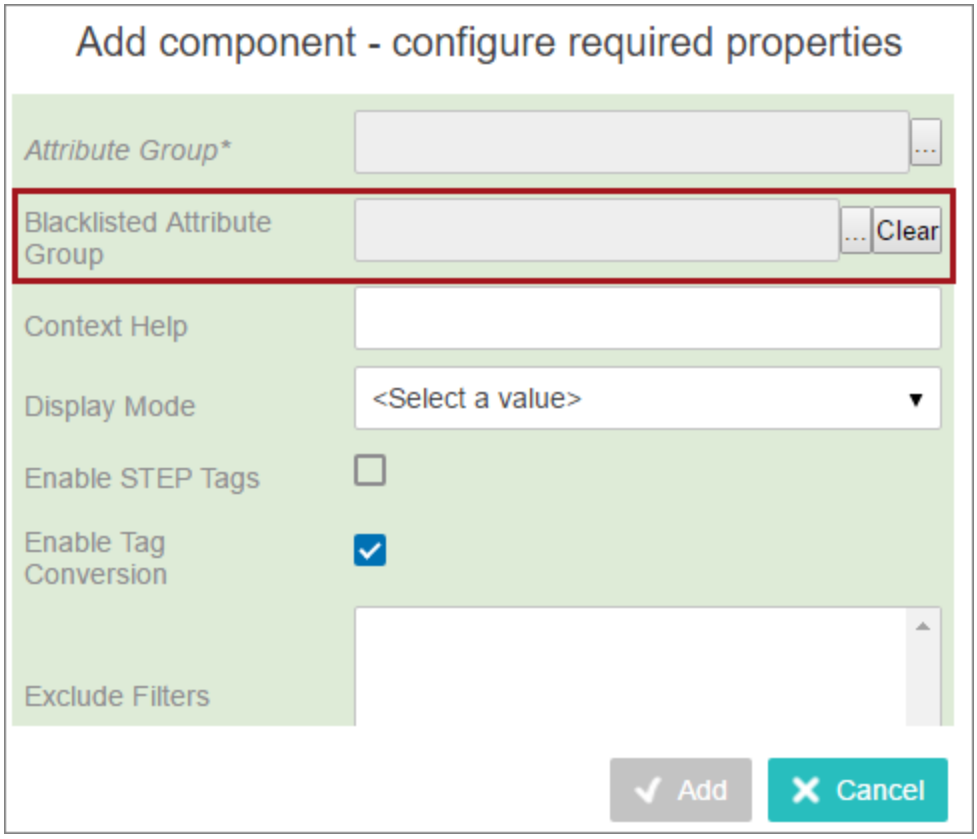
More information about attributes and creating attribute groups can be found in the **Attributes** section of the **System Setup / Super User Guide** documentation.

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.' Form more information the Attribute Value Group Component Properties topic in the Web UI Setup and User Guide / Web User Interface documentation.



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.

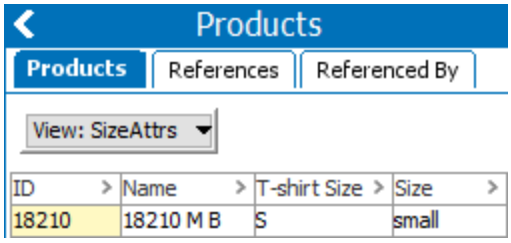
Merging Attribute Values

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.



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.

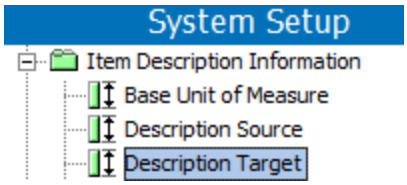
Verify Attributes to be Merged

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.

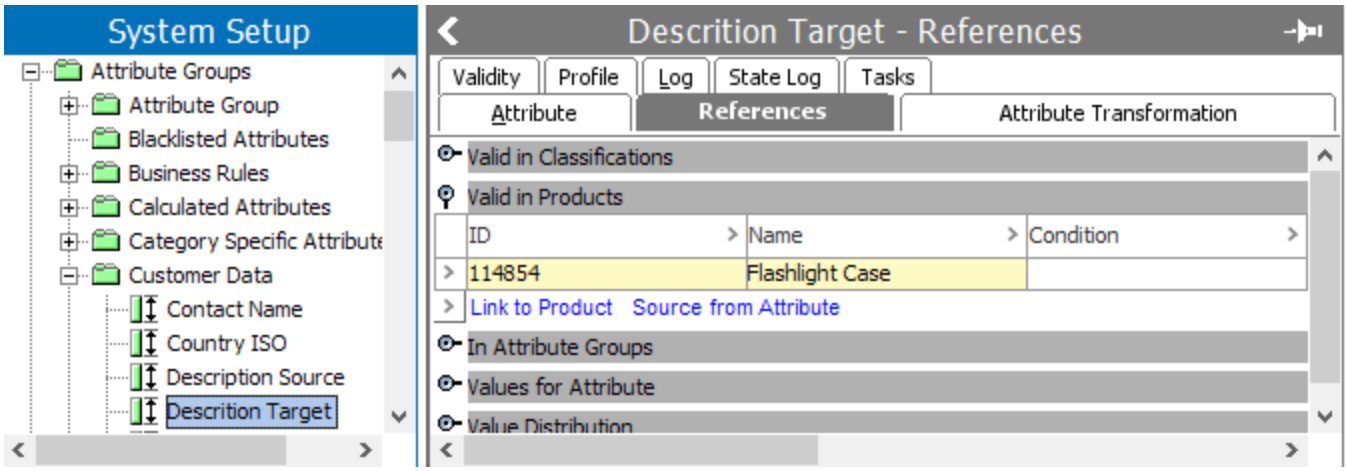
- Ensure that the upstream systems that feed the data to STEP are configured to send the data to the target attribute as the source attribute will be deleted.
- Ensure that if there are any existing import or export configurations using the source attribute, that the configuration modified. This is to avoid errors when trying to run the configuration after the source attribute is deleted form STEP.

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

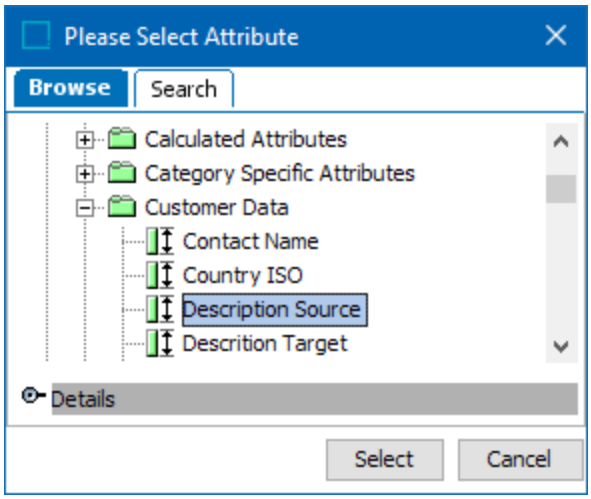


2. On the References tab, if it exists, open the 'Valid in Classifications' flipper, and click the **Source from Attribute** link.

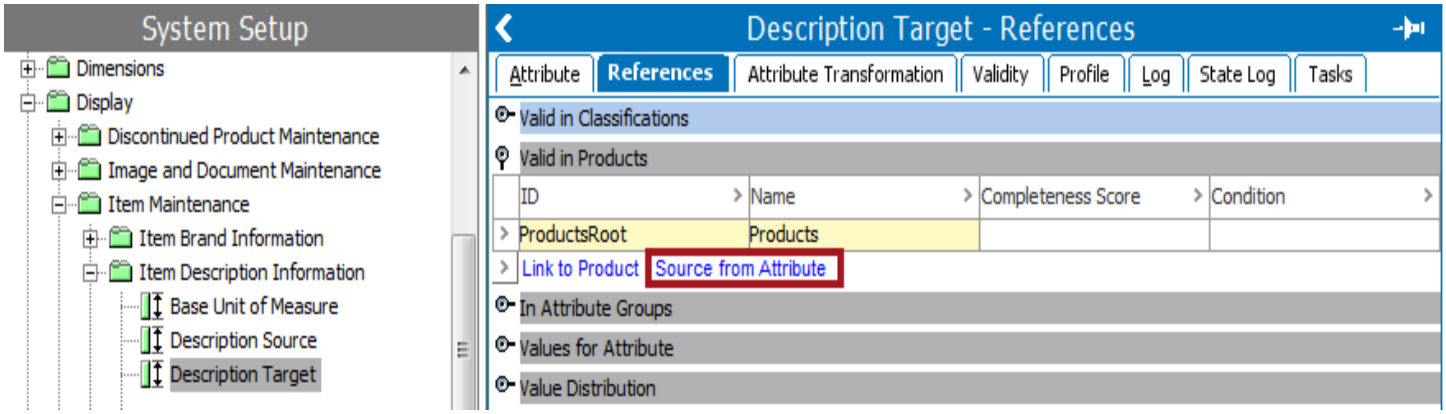
> **Note:** If the attribute is not valid for any classifications, verify that the source attribute is also not valid for classifications.



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.

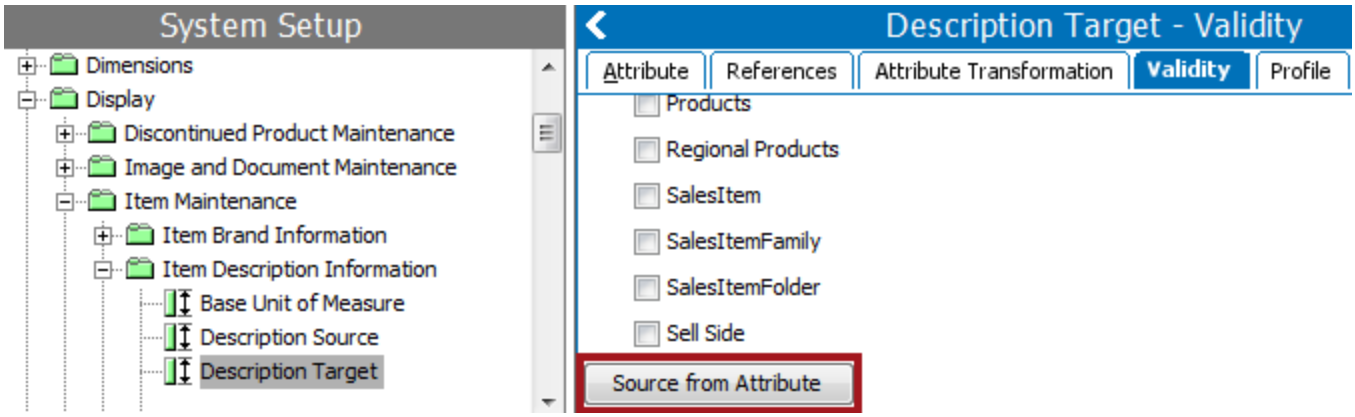


4. On the References tab open the 'Valid in Products' flipper (if it exists), and click the **Source from Attribute** link.



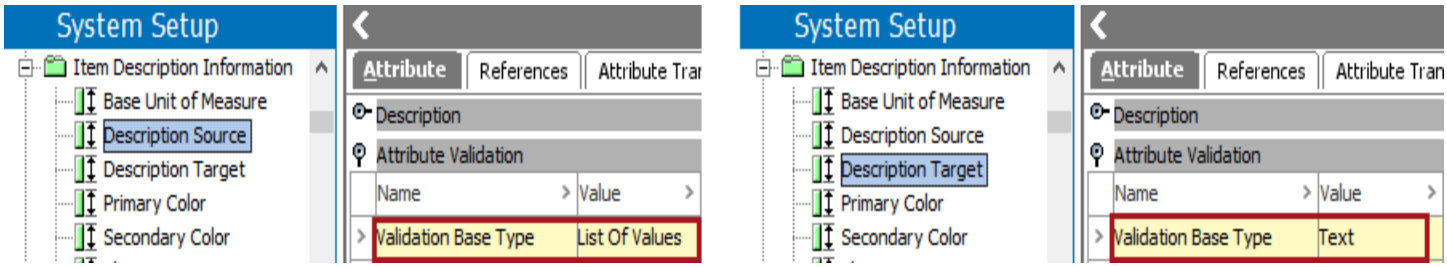
5. Use Browse or Search to find the **Source Attribute** and click **Select**. The product links of the source attribute are copied to the target attribute.

6. On the Validity tab, under the Valid for Product Types flipper, click the **Source from Attribute** button.



7. Use Browse or Search to find the **Source Attribute** and click Select. The validity of the source attribute is copied to the target attribute.

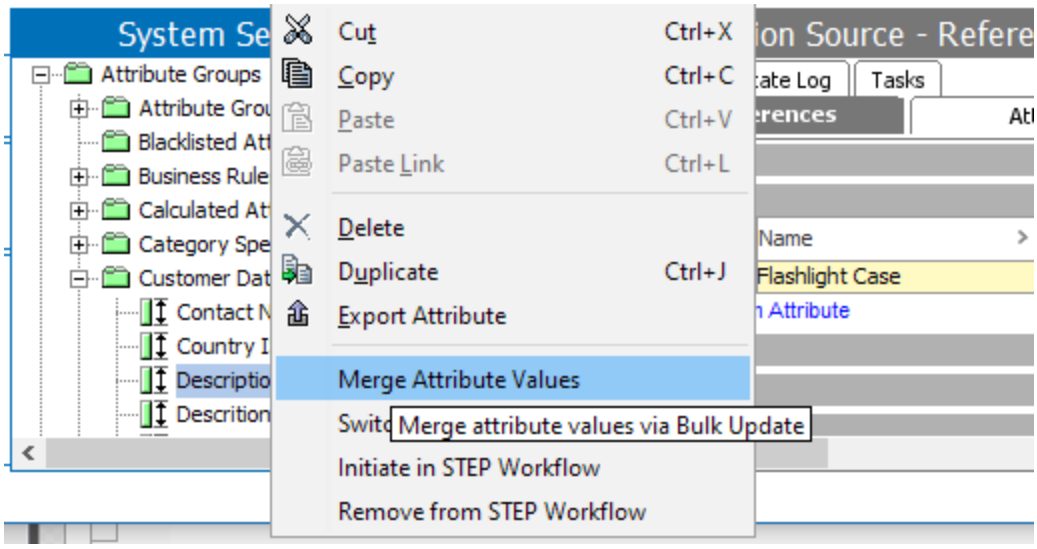
8. In System Setup, on the Attribute tab under the Attribute Validation flipper, compare the Validation Base Type for the **Target Attribute** and the **Source Attribute**.



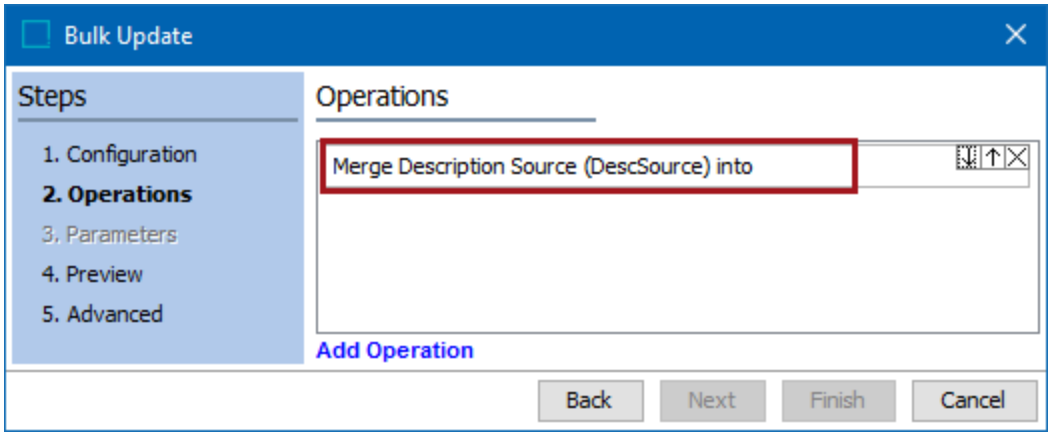
- If the Validation Base Types are the same and are not LOVs, continue to **Merge Values Using Bulk Update**.
- If the Validation Base Types are not the same, determine if they are compatible by manually adding an existing source attribute value to the target attribute successfully. If the manual test works, continue to **Merge Values Using Bulk Update**.
- If the Validation Base Types are not the same and they are not compatible, in the case of an target LOV, setting the 'Allow Users to Add Values' option to 'Yes' allows the new values to be written to the target LOV. Otherwise, see **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: Merge Attribute Values Operation** topic of the **Getting Started** / **User Guide** documentation.

Reparenting Attribute Groups

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

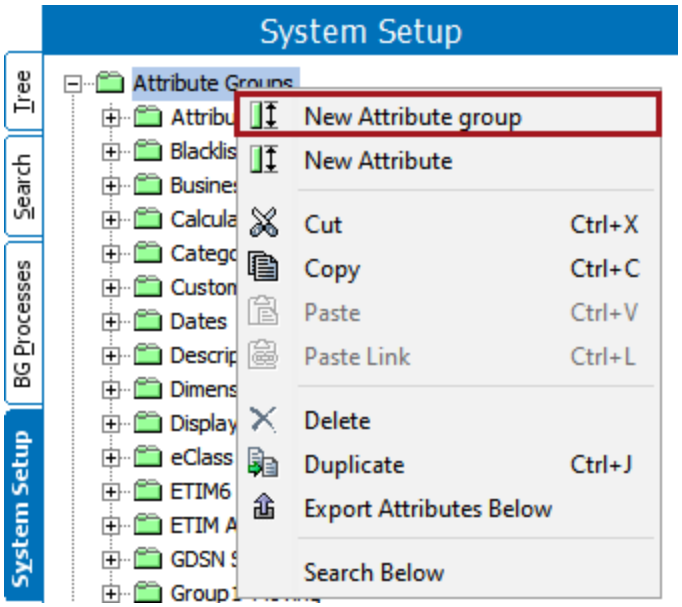
- Manual Reparenting - For smaller attribute groups with only a couple of attributes, users are able to easily drag and drop the attributes under a desired parent.
- STEPXML Reparenting - Using STEPXML, users can reparent an attribute group with numerous attributes inside.

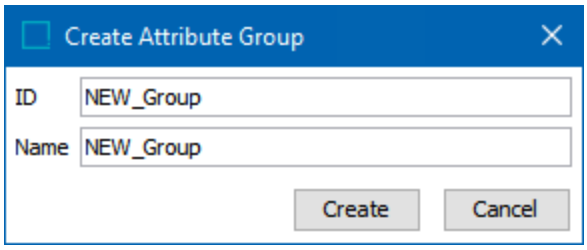
Manual Reparenting

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.

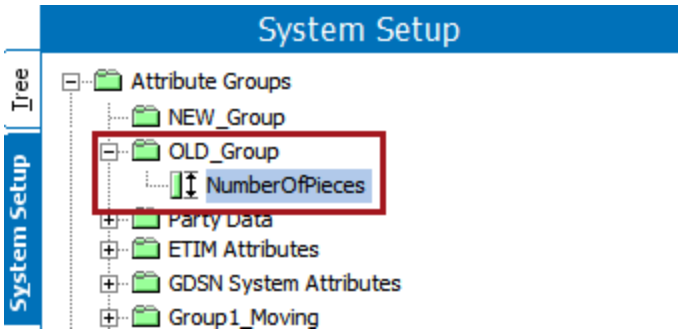
Note: To reuse the name of the existing attribute group inside of the new parent attribute, then that existing group must be deleted first. This is required because STEP auto-generates a name based on the STEP ID, and only one instance of a STEP ID may exist.

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

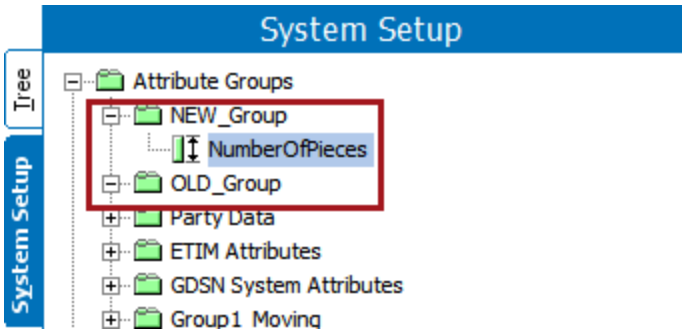




2. Open the attribute group containing the attribute to reparent by pressing the plus (+) icon next to the attribute group name.

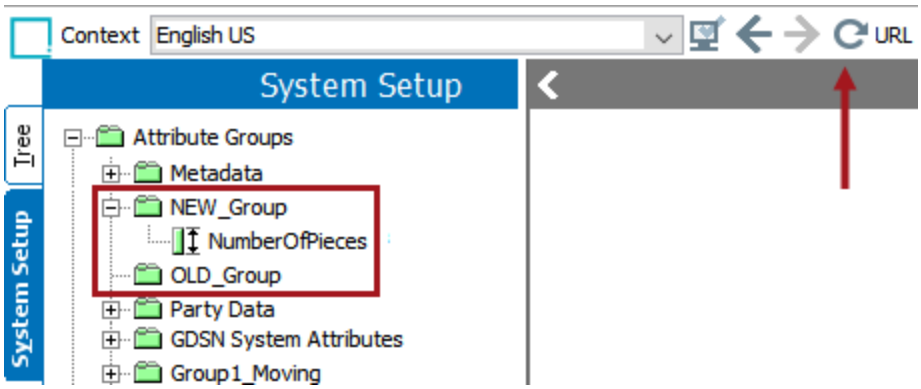


3. Drag the desired attribute(s) into the new attribute group.

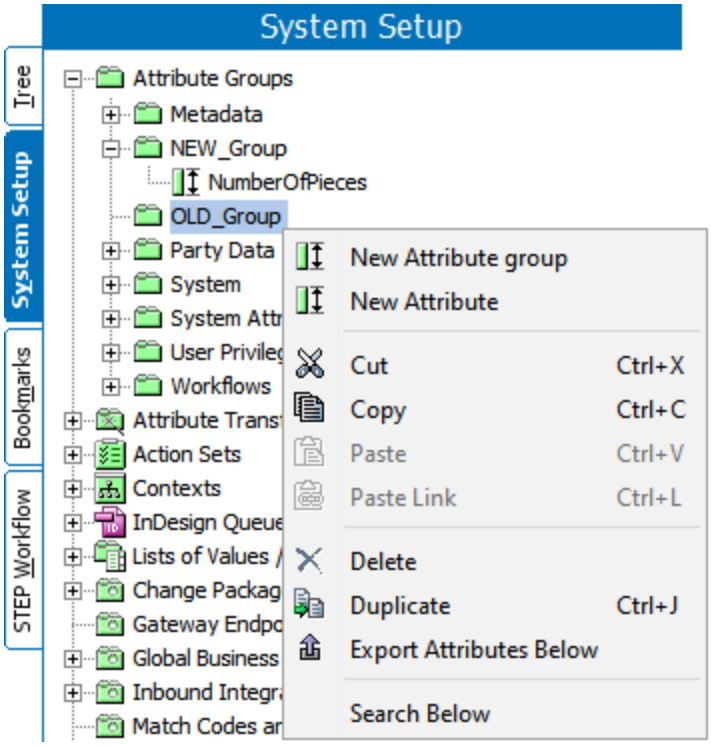


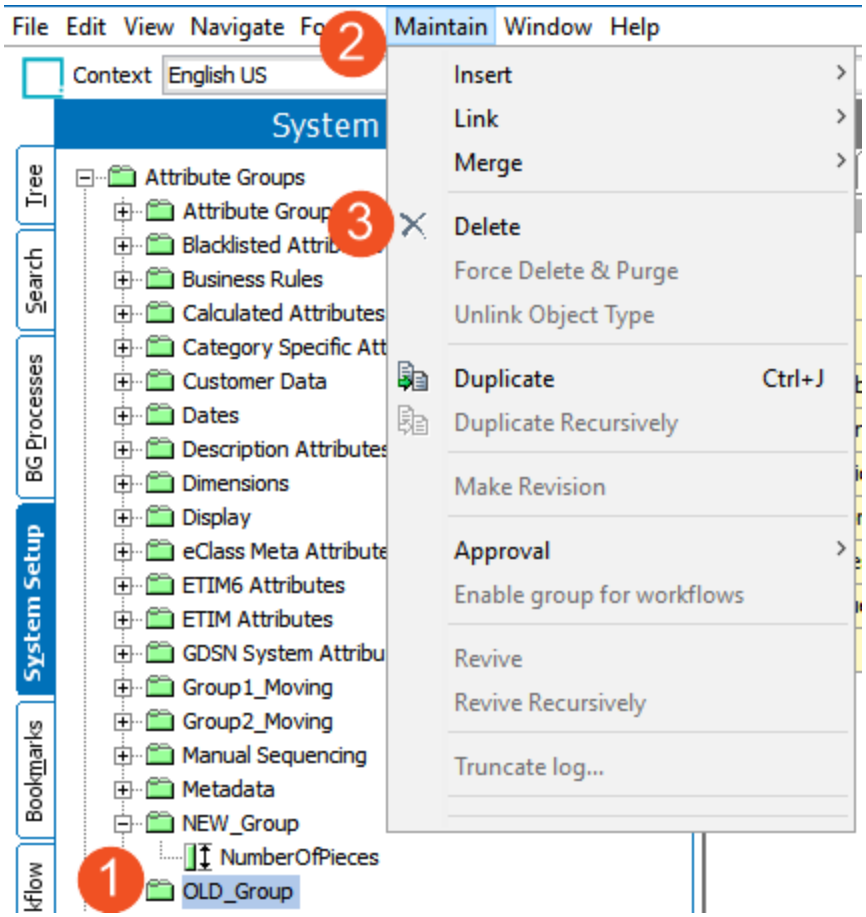
Note: When dragging the attribute, the old attribute group will retain the open icon, which is a minus (-) sign. This is due to the fact that the system has not been updated.

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

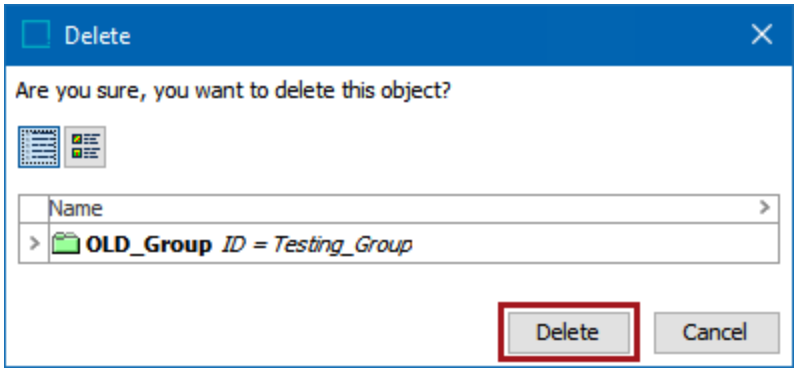


5. Delete your old group from the Tree tab by right-clicking on the attribute group and selecting **Delete**, or by going to the Maintain menu > **Delete**.

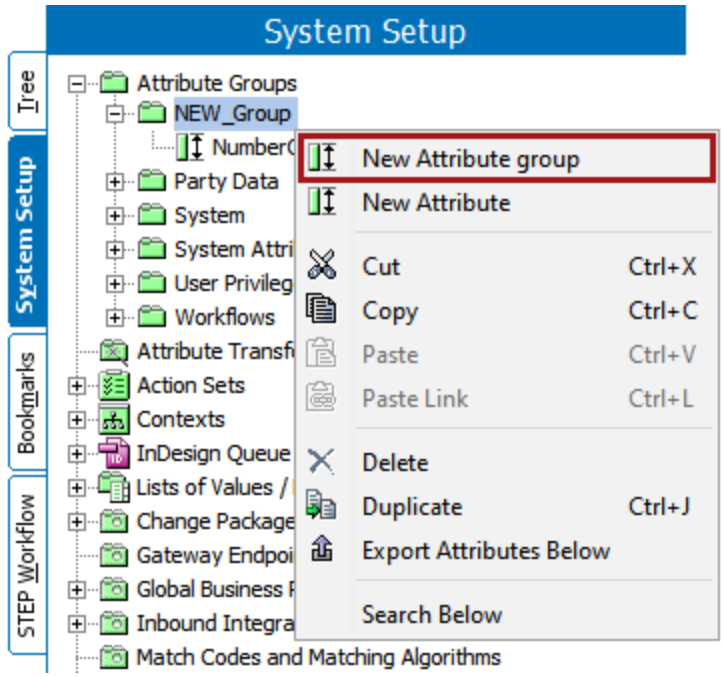


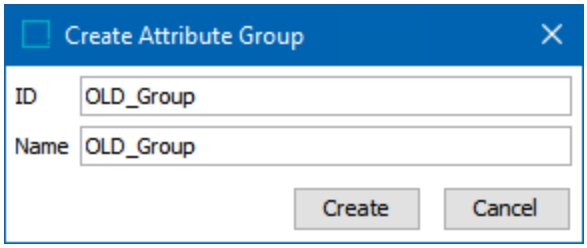


6. Click **Delete** to confirm the deletion on the pop-up prompt.

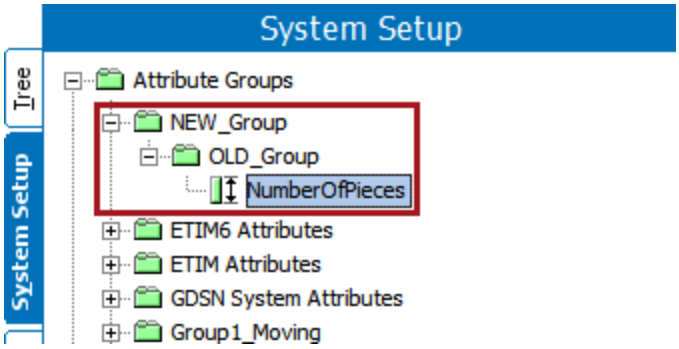


7. Create the old group under the new attribute group that was created in step 1.





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

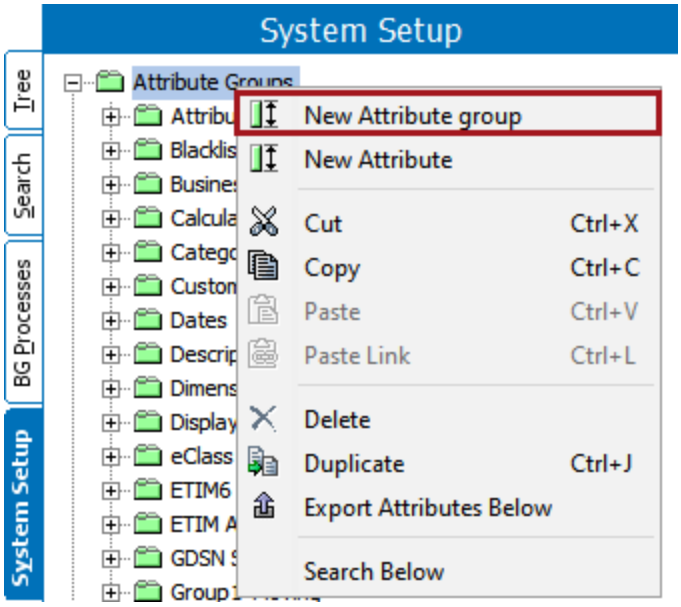


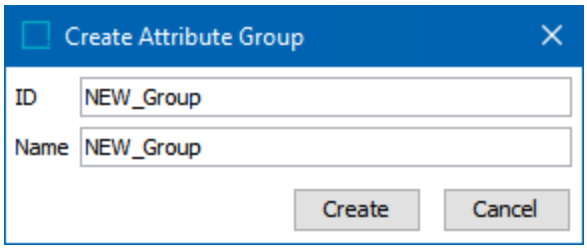
STEPXML Reparenting

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

Note: Using the STEPXML 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.

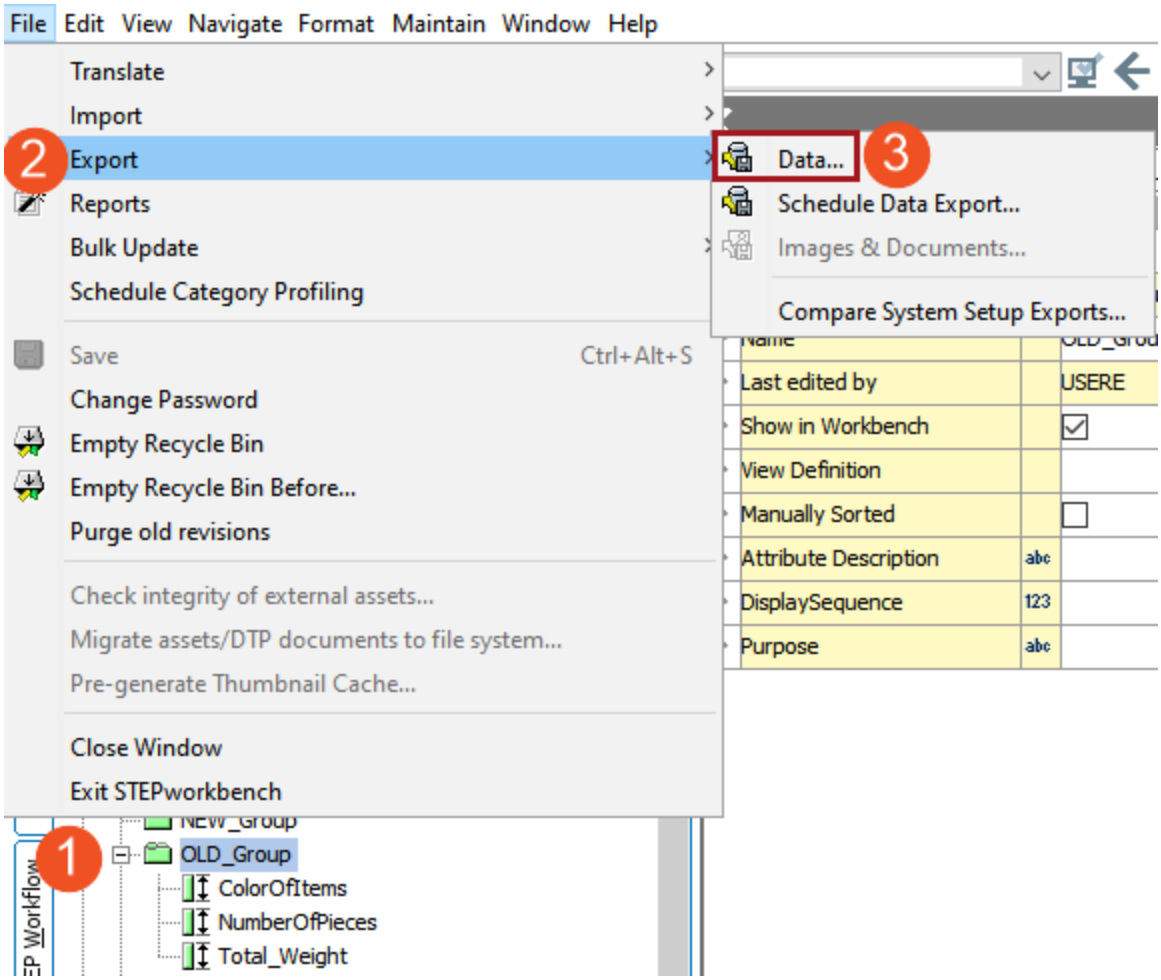




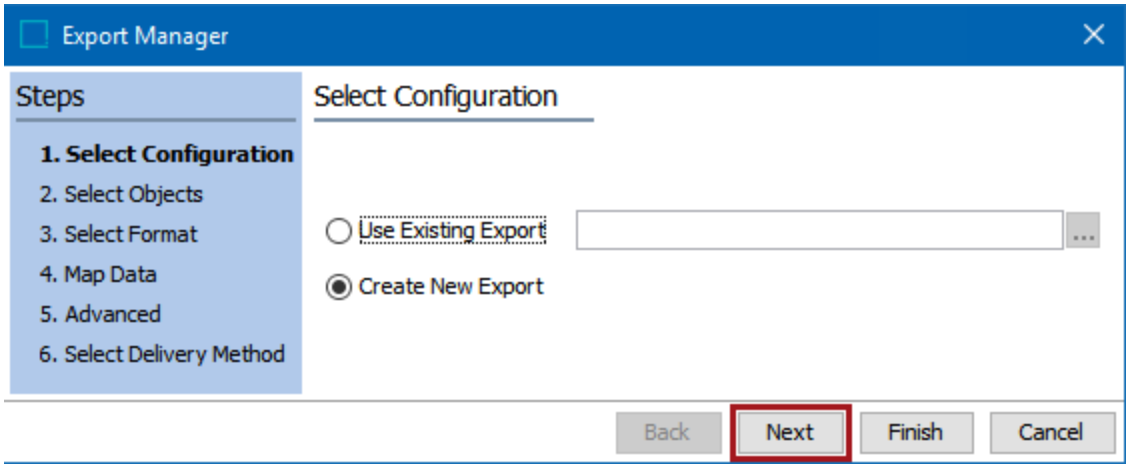
Export the Attribute Group

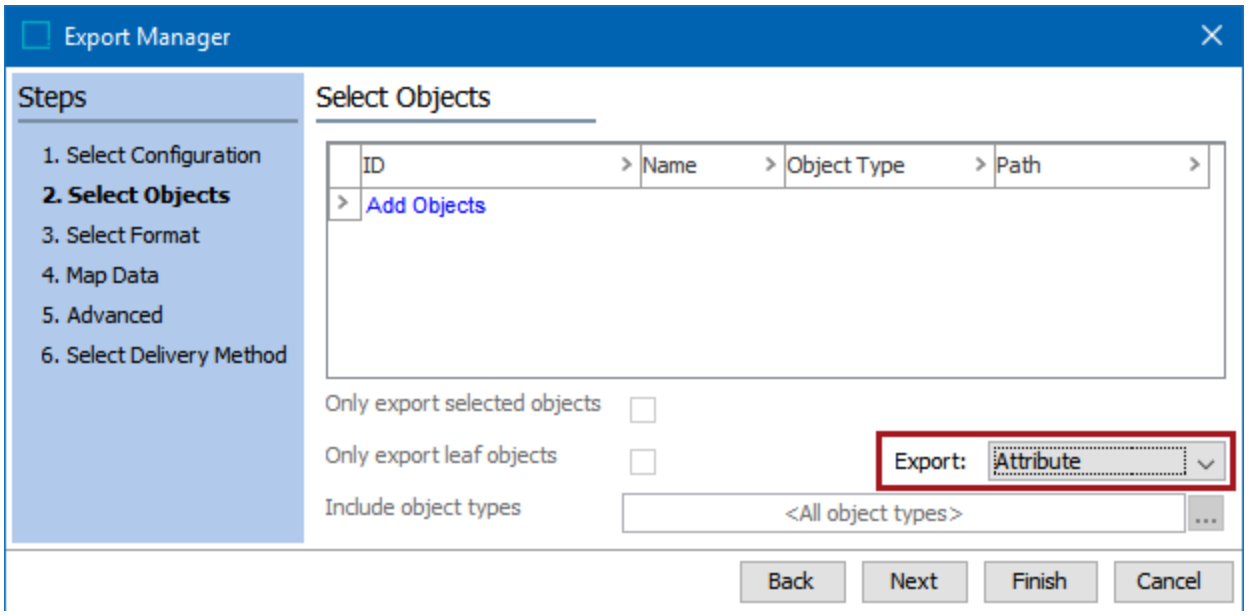
Next, export the attribute group that needs reparenting. To properly export an attribute group for reparenting:

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

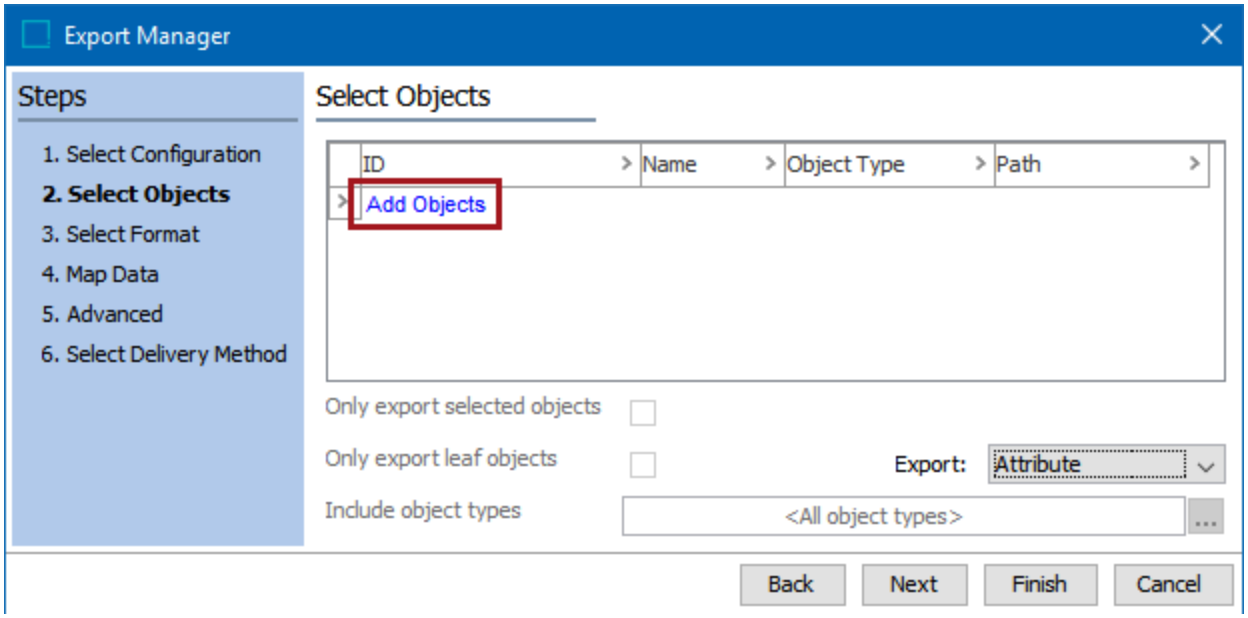


2. The Export Manager will display. Select **Next**, and on step 2 of the Export Manager, Select Objects, select **Attribute** from the Export dropdown menu.

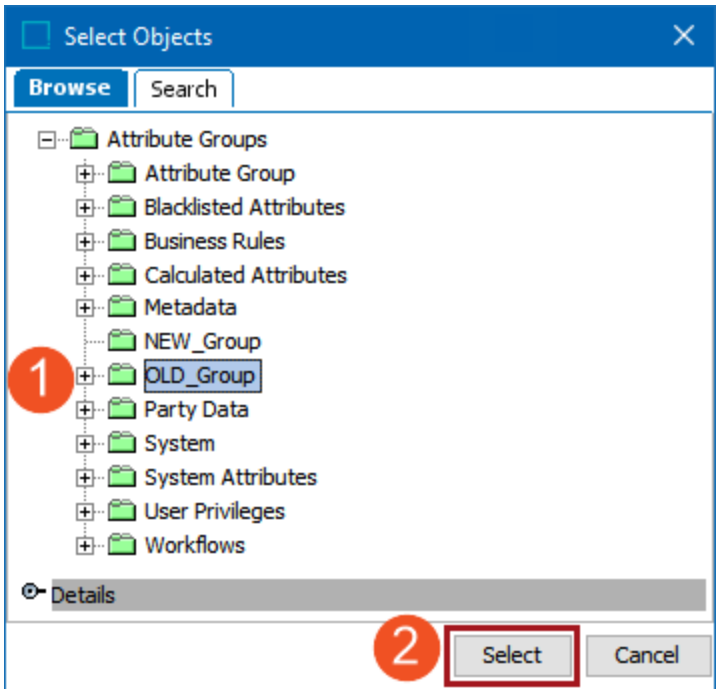




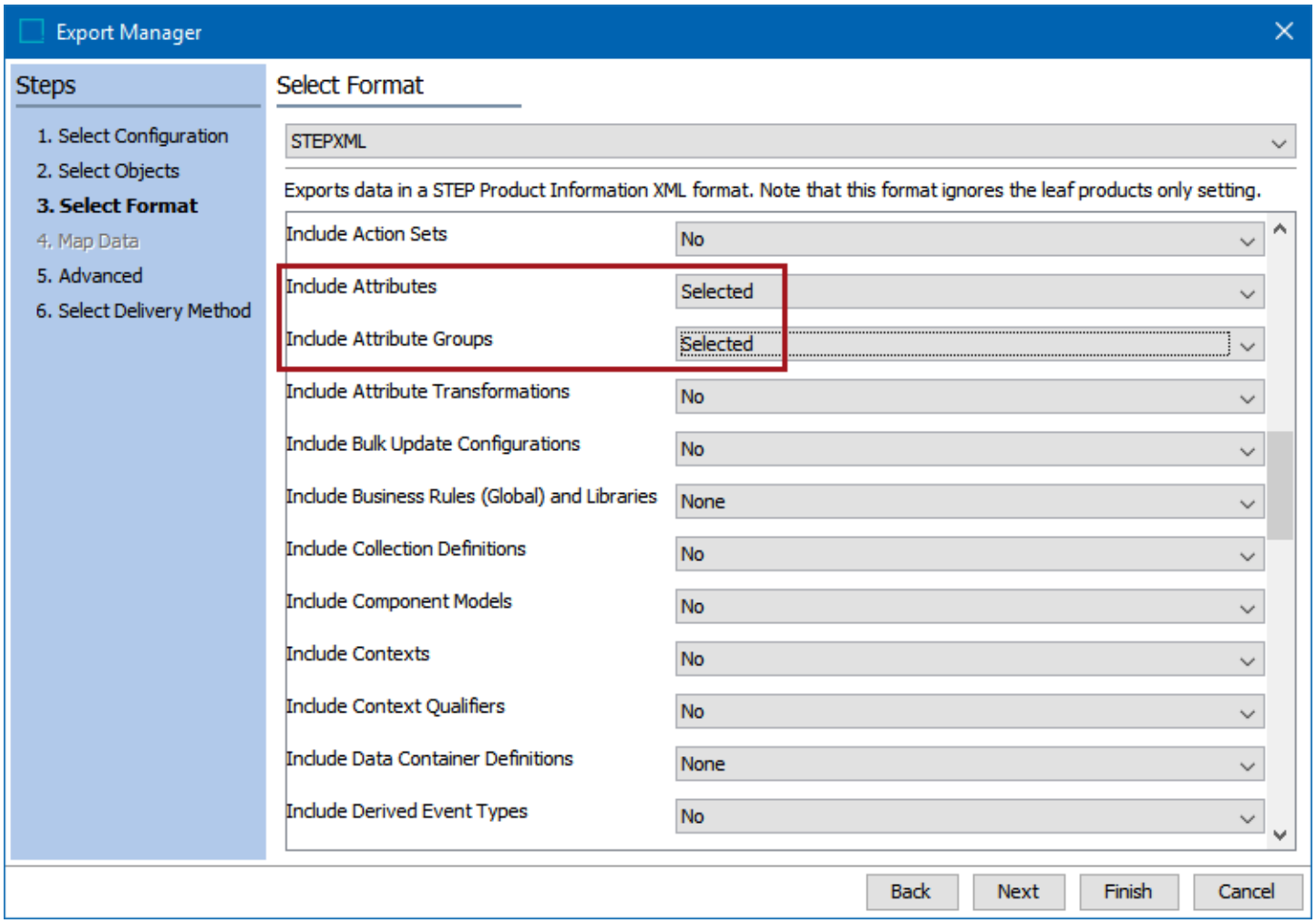
3. Finally, select **Add Objects** from the data field.



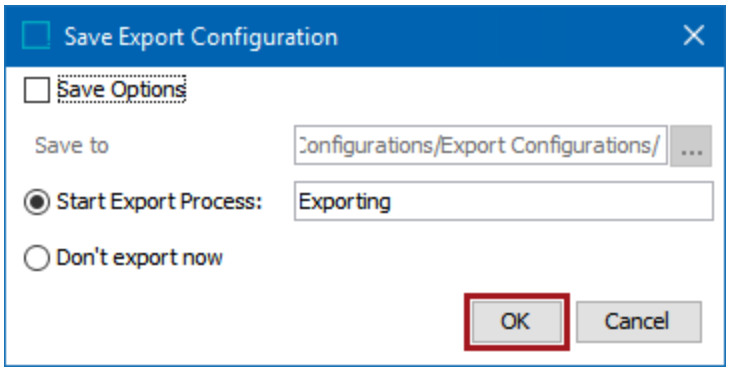
4. Select the desired attribute group to reparent by clicking the group then pressing **Select**. Select **Next** once the attribute group is added to the Select Objects list.



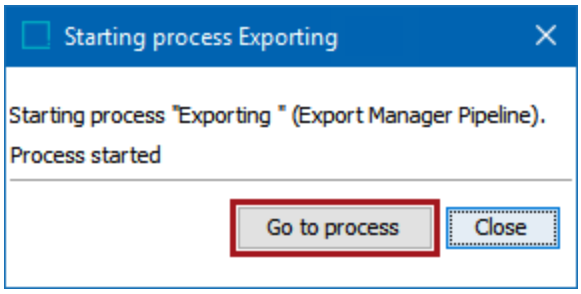
5. On step 3, Select Format, first select STEPXML from the dropdown menu. Then, navigate to **Include Attribute Groups** and **Include Attributes** and set both options to **Selected**. Select **Finish** to export the file.

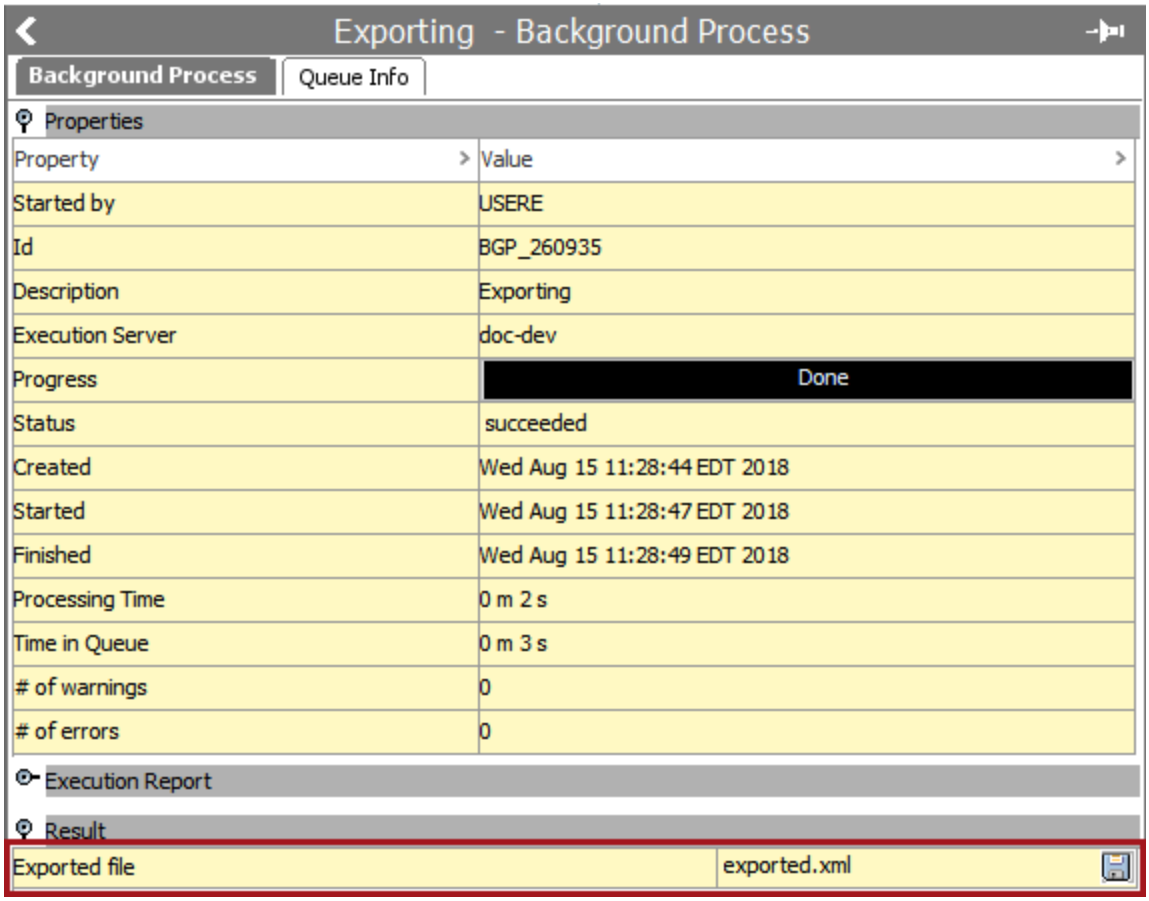


6. Press **OK** on the export configuration pop-up. To save this configuration for future use, select the Save Options box and select a location. Otherwise, the export will begin and give a status update.



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





Adding STEPXML

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

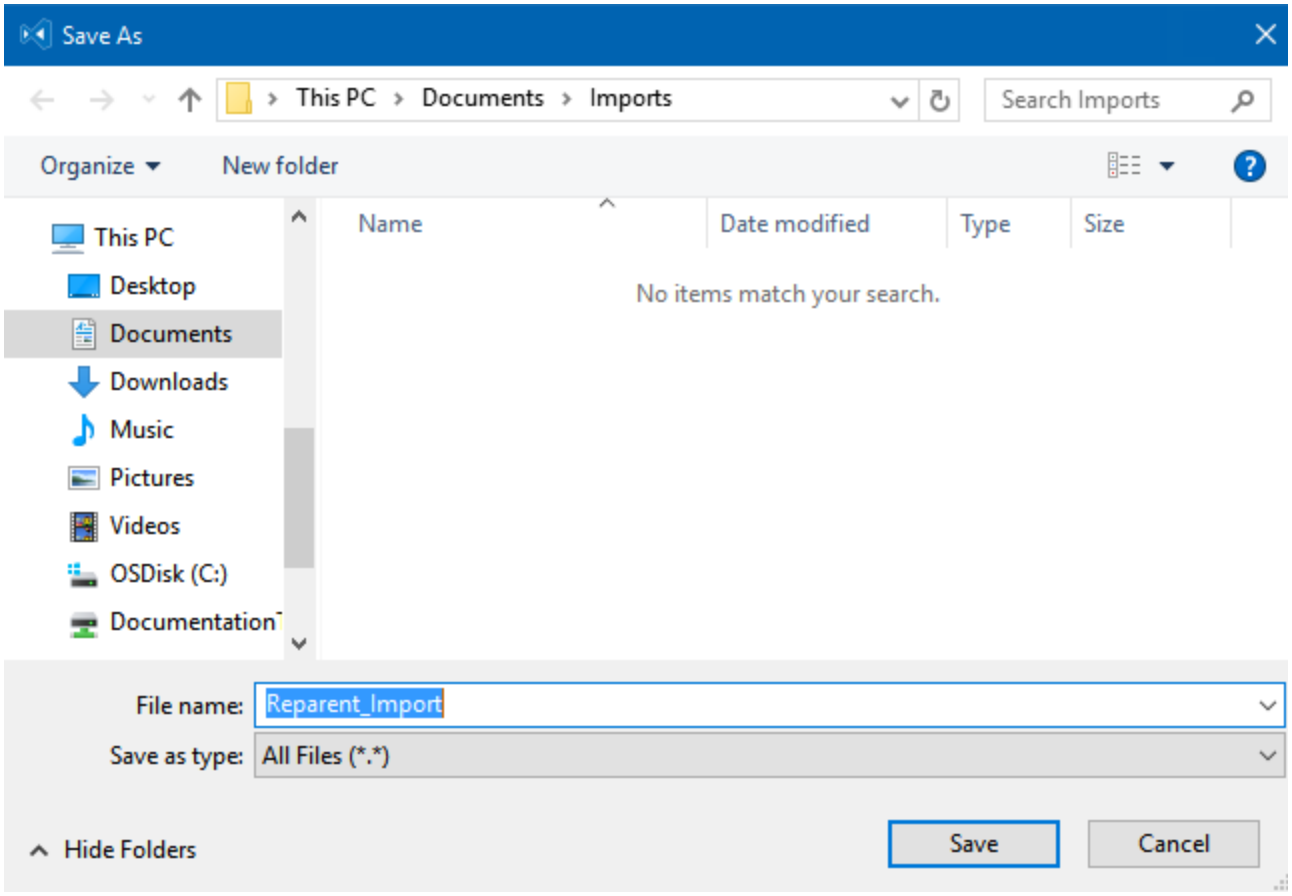
1. After double-clicking the exported file, it will open in the default XML browser. Locate the AttributeGroup ID="[Your old group]" tag.

```
15  <STEP-ProductInformation ExportTime="2018-08-15 11:28:48" ExportContext="C
16
17   <AttributeGroupList>
18
19   <AttributeGroup ID="OLD_Group" ShowInWorkbench="true" Referenced="true">
20     <Name>OLD_Group</Name>
21   </AttributeGroup>
22   </AttributeGroupList>
23   </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.

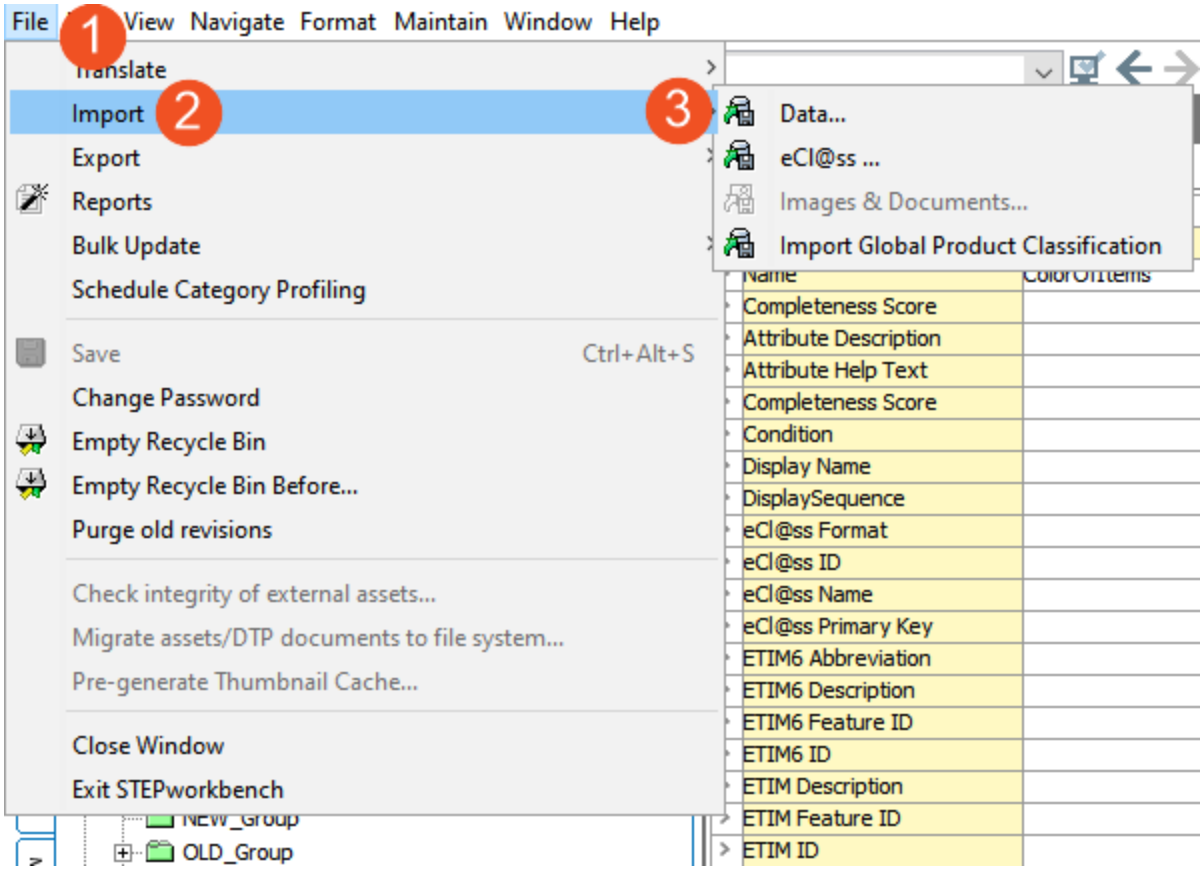
```
15  <STEP-ProductInformation ExportTime="2018-08-15 11:28:48" ExportContext="Context1" ContextID="C
16
17   <AttributeGroupList>
18
19   <AttributeGroup ID="OLD_Group" ShowInWorkbench="true" Referenced="true" ParentID="NEW_Group">
20     <Name>OLD_Group</Name>
21   </AttributeGroup>
22   </AttributeGroupList>
23   </STEP-ProductInformation>
```

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

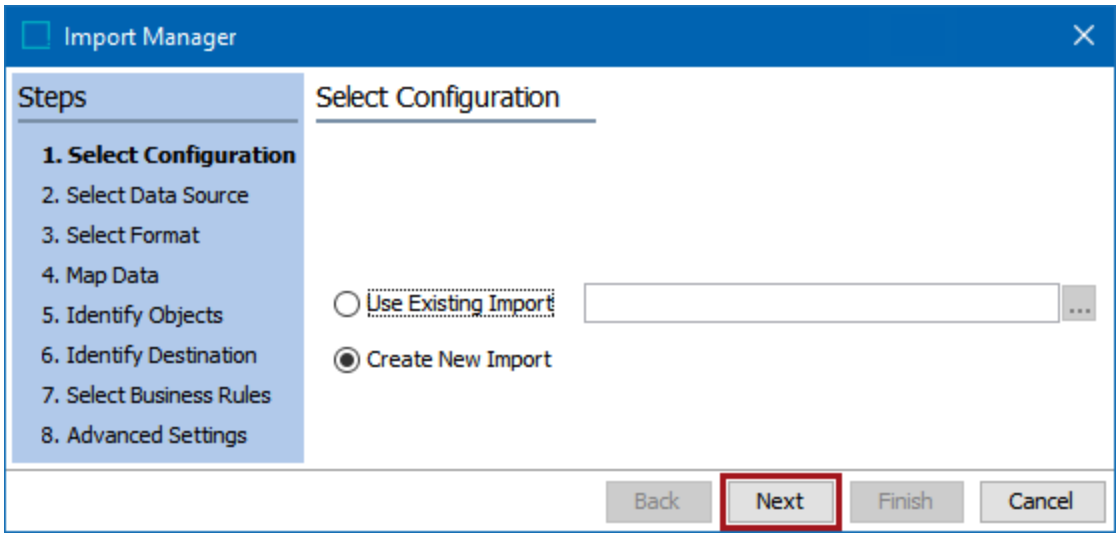


Upload the Modified File

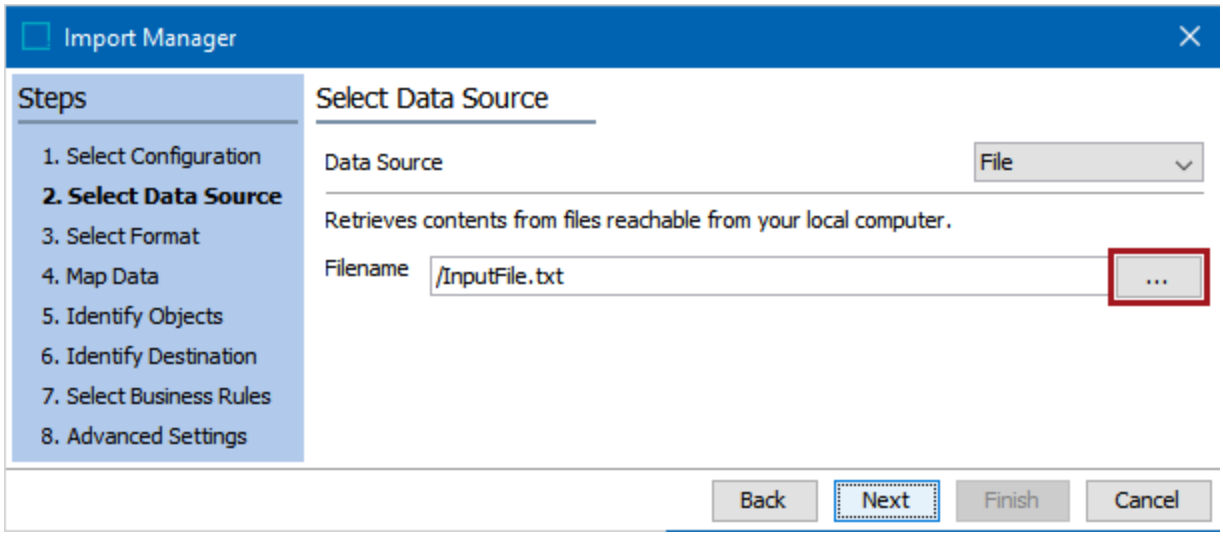
1. In the STEP Workbench, select File > Import > **Data...**.

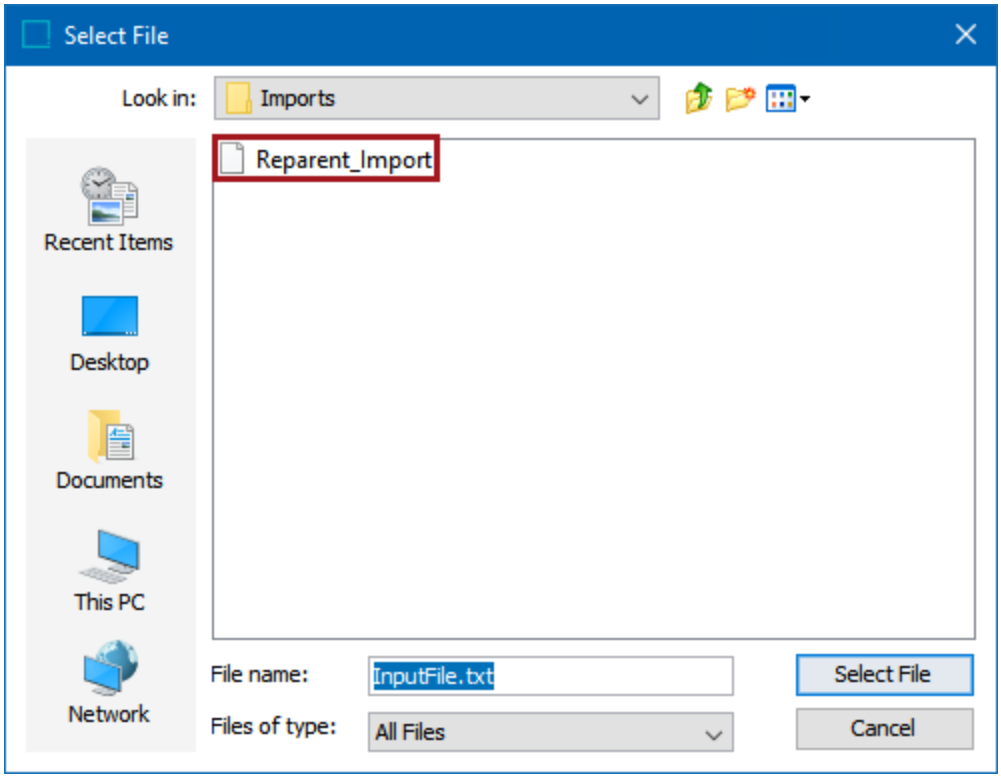


2. The Import Manager will load. Select **Next**.

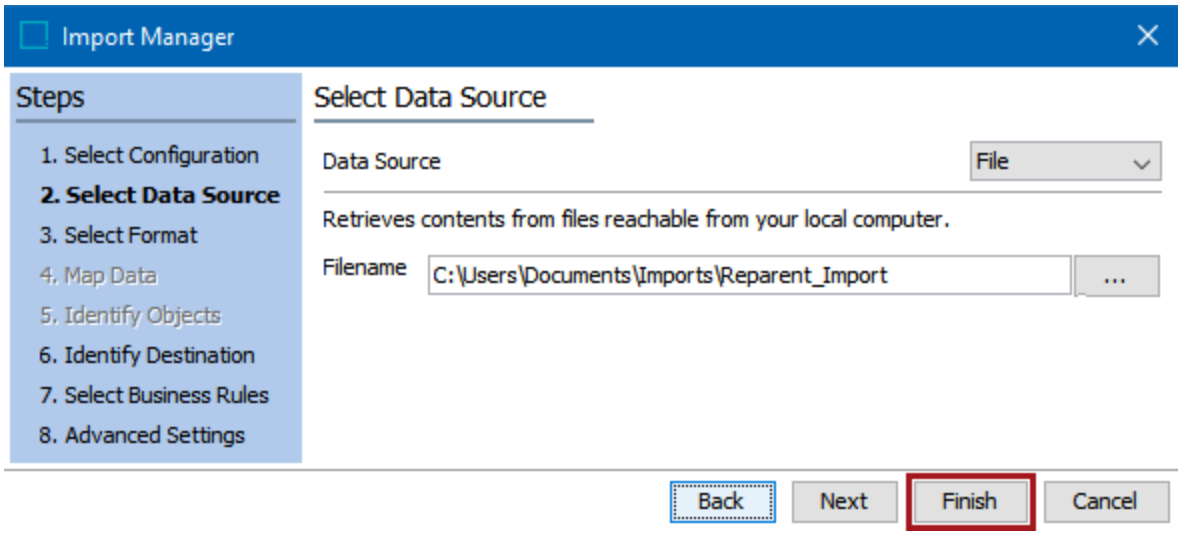


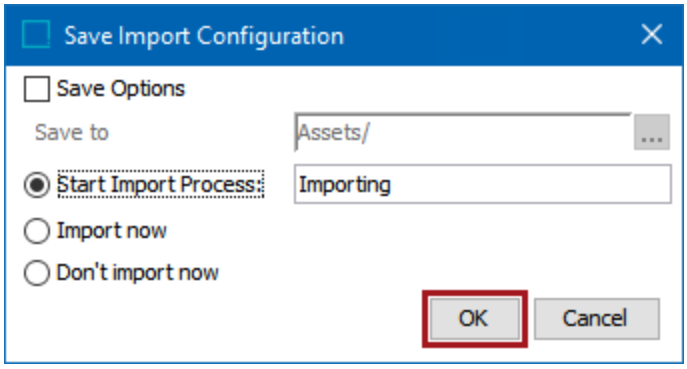
3. On **Step 2. Select Data Source**, select the previously modified STEPXML file from the last step.



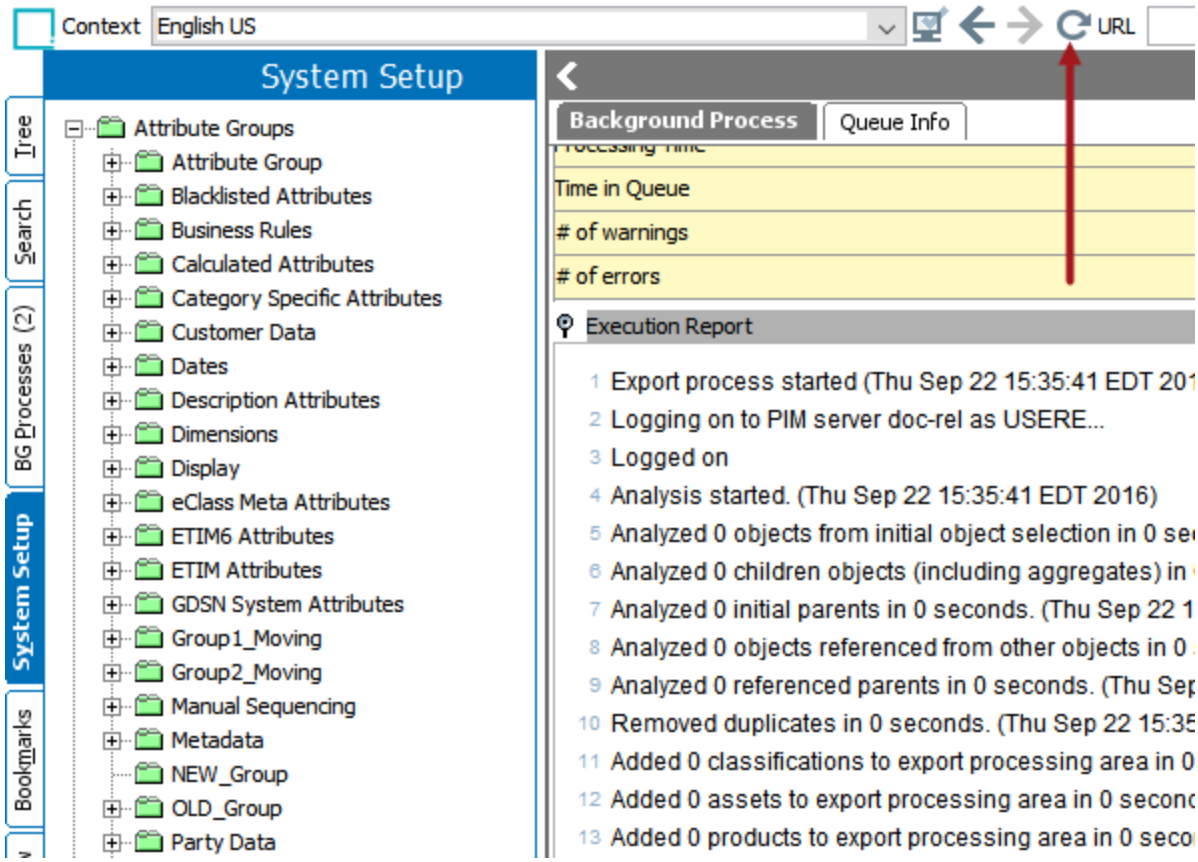


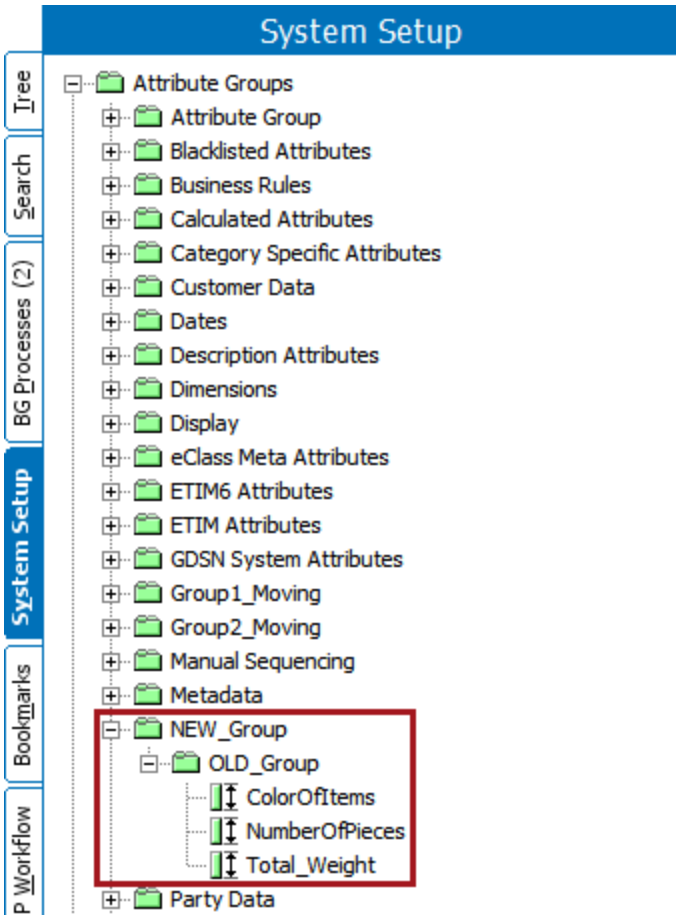
4. Select **Finish**. A Save Import Configuration dialog will pop-up. Select **OK** to import the STEPXML script.





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

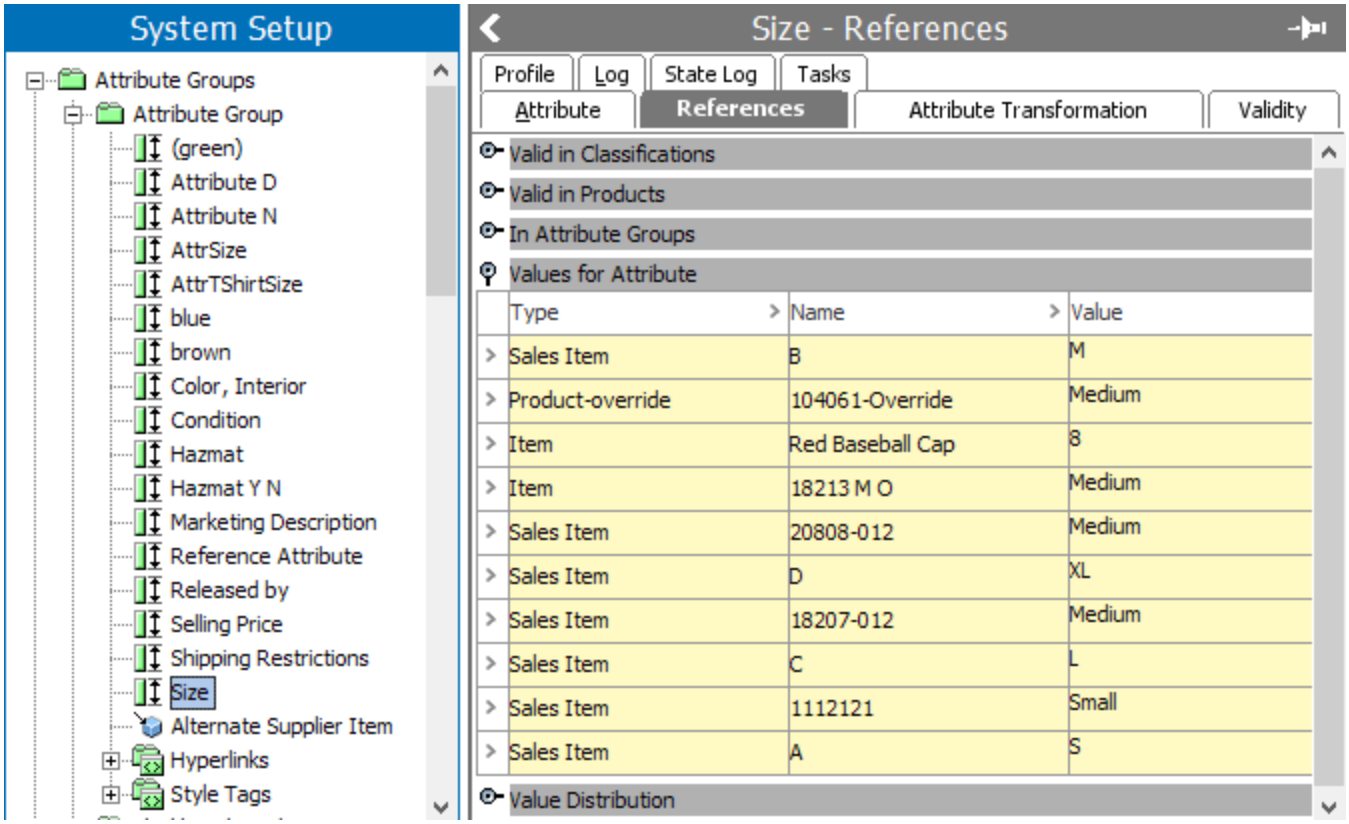

Context
English US
URL
System Setup
Tree
Search
BG Processes (2)
System Setup
Bookmarks
Attribute Groups
Attribute Group
Blacklisted Attributes
Business Rules
Calculated Attributes
Category Specific Attributes
Customer Data
Dates
Description Attributes
Dimensions
Display
eClass Meta Attributes
ETIM6 Attributes
ETIM Attributes
GDSN System Attributes
Group1_Moving
Group2_Moving
Manual Sequencing
Metadata
NEW_Group
OLD_Group
Party Data
Background Process
Queue Info
Time in Queue
# of warnings
# of errors
Execution Report
1 Export process started (Thu Sep 22 15:35:41 EDT 201
2 Logging on to PIM server doc-rel as USERE...
3 Logged on
4 Analysis started. (Thu Sep 22 15:35:41 EDT 2016)
5 Analyzed 0 objects from initial object selection in 0 se
6 Analyzed 0 children objects (including aggregates) in
7 Analyzed 0 initial parents in 0 seconds. (Thu Sep 22 1
8 Analyzed 0 objects referenced from other objects in 0
9 Analyzed 0 referenced parents in 0 seconds. (Thu Sep
10 Removed duplicates in 0 seconds. (Thu Sep 22 15:35
11 Added 0 classifications to export processing area in 0
12 Added 0 assets to export processing area in 0 second
13 Added 0 products to export processing area in 0 seco


System Setup
Tree
Search
BG Processes (2)
System Setup
Bookmarks
P 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

Viewing Attribute Values

Attribute values for all products can be viewed in System Setup using the attribute editor on the References tab.

Note: If the attribute is valid for a lot of products across hierarchies, it may take some time to populate the values.

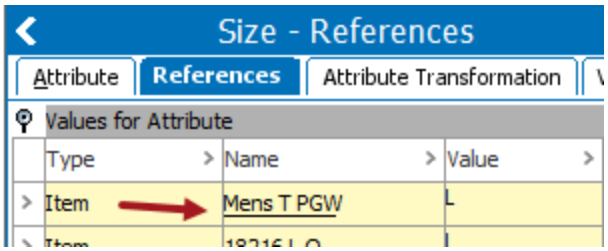


Viewing the values of an attribute means getting a list of the objects' names in which the current attribute has values, and what the value is.

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

View Values and Edit Values

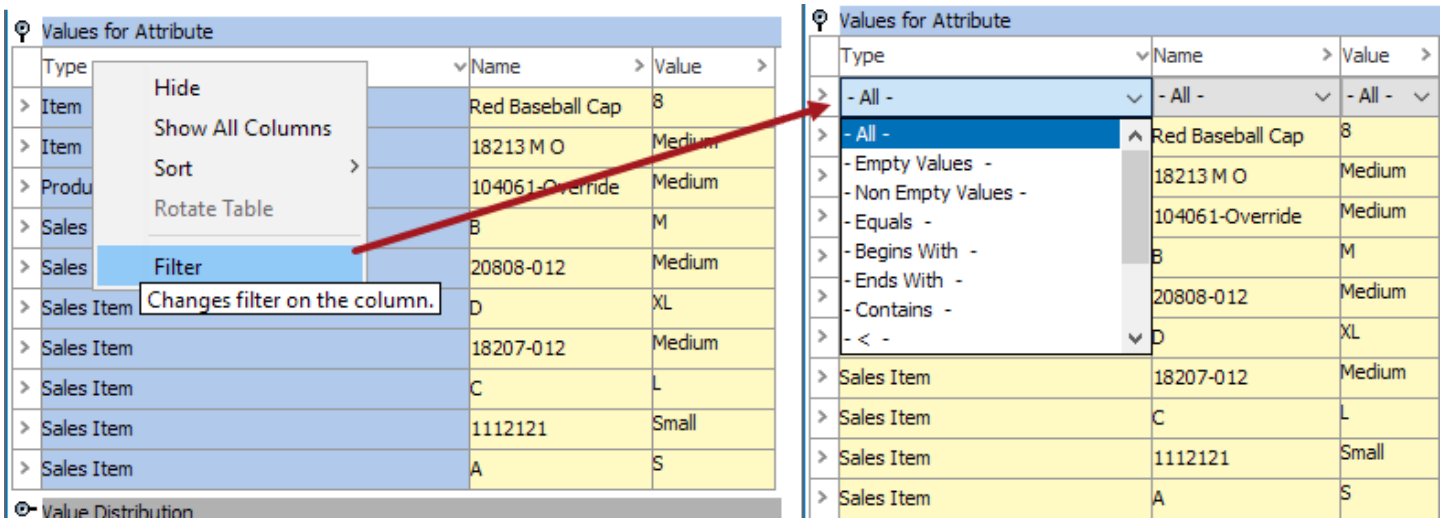
1. In System Setup, select an attribute and click the **References** tab the attribute editor.
2. Open the **Values for Attribute** flipper to display objects and their corresponding values for the attribute.
3. To edit the value, hover over the name of an object in the list to display a link.



4. Click the link to open the corresponding editor. For example, the product editor is displayed if the value exists on a product.

Filtering Values for Attribute

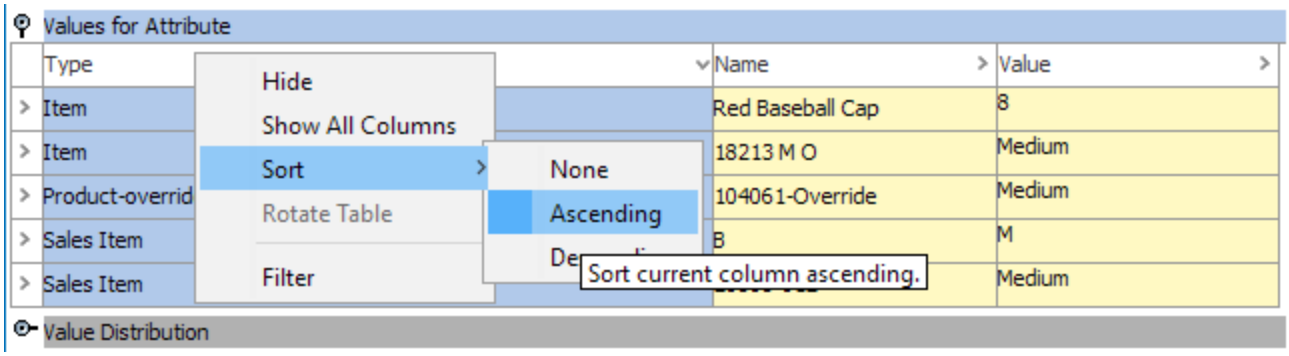
Filters can be added to the **Value of Attributes** flipper on any column. To filter, right-click on any column header and select **Filter**. Filter will then appear in all column headers.



Sorting Values for Attribute

The values can be sorted to 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.


Values for Attribute
Type
Name
Value
Item
Red Baseball Cap
8
Item
18213 M O
Medium
Product-overrid
104061-Override
Medium
Sales Item
B
M
Sales Item
Medium
Hide
Show All Columns
Sort
Rotate Table
Filter
None
Ascending
Sort current column ascending.
Value Distribution

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 a value, or going so far as to prevent approval and/or processing of an object until the required values are populated.

This topic describes:

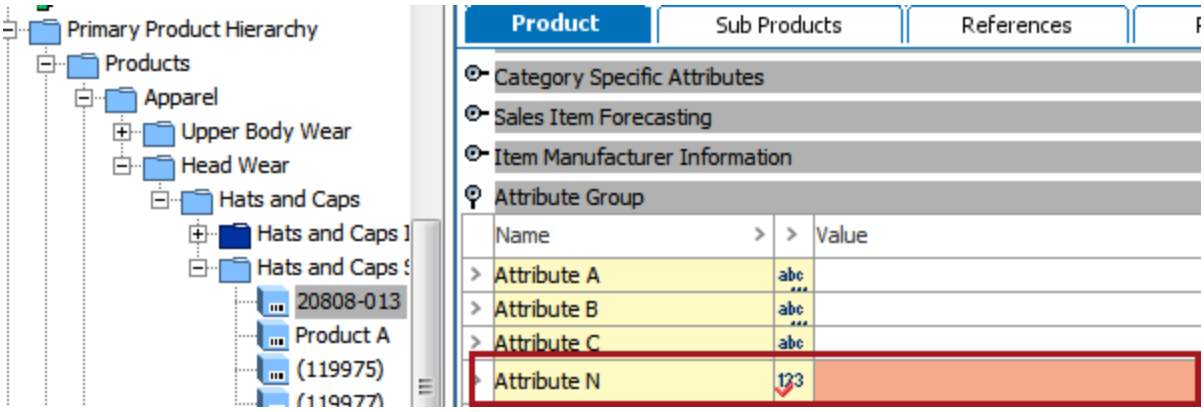
- The standard options for making attributes mandatory
- Detailed configuration and end user information for the various settings, or links to other topics that provide this information
- Considerations for how the different mandatory options behave across the varying interfaces (workbench, Web UI, and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

Standard Mandatory Attributes / 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.

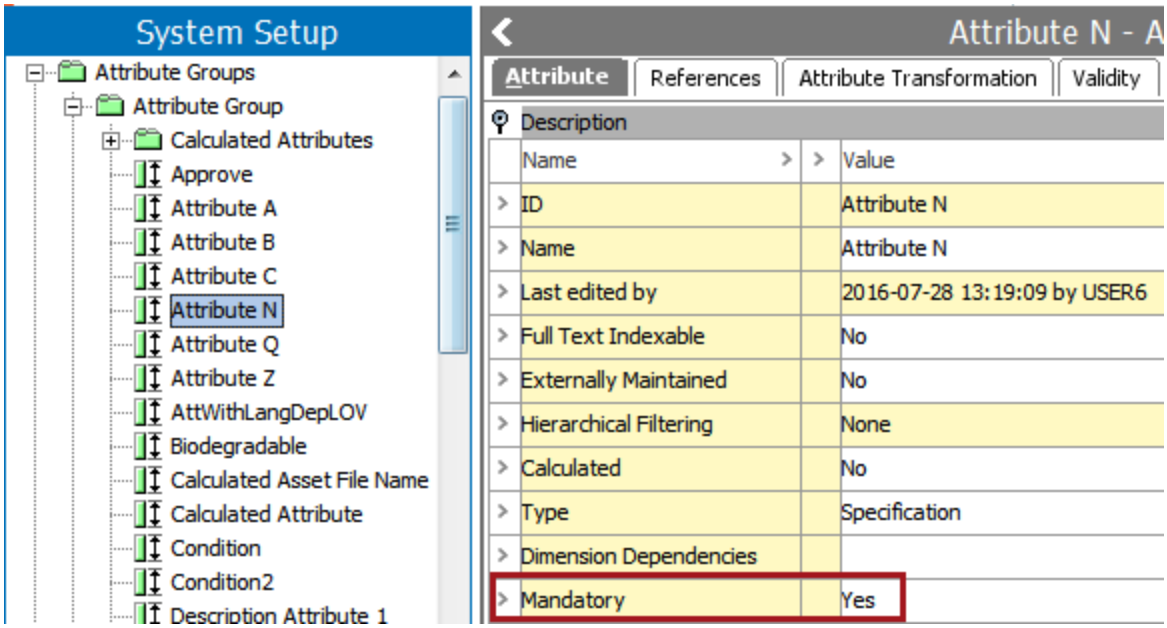
Workbench Visual Indicators

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

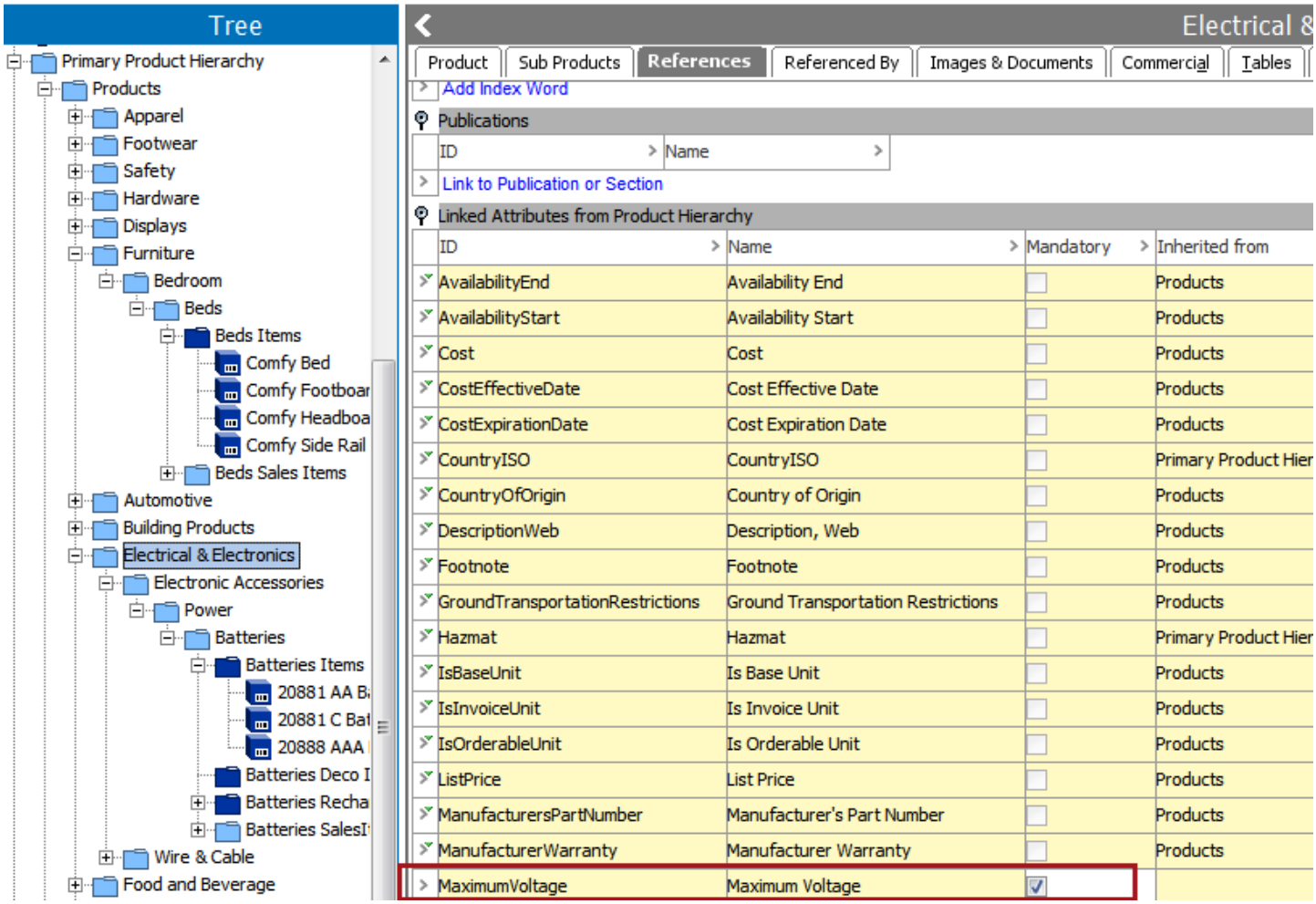


The attribute indicator has a red check mark and the value field is highlighted red for attributes marked as mandatory. Note that the red highlighting is present regardless of the workbench interface (e.g., the standard object editor in Tree navigator, the Tasks tab on an object in Tree, or the STEP 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.

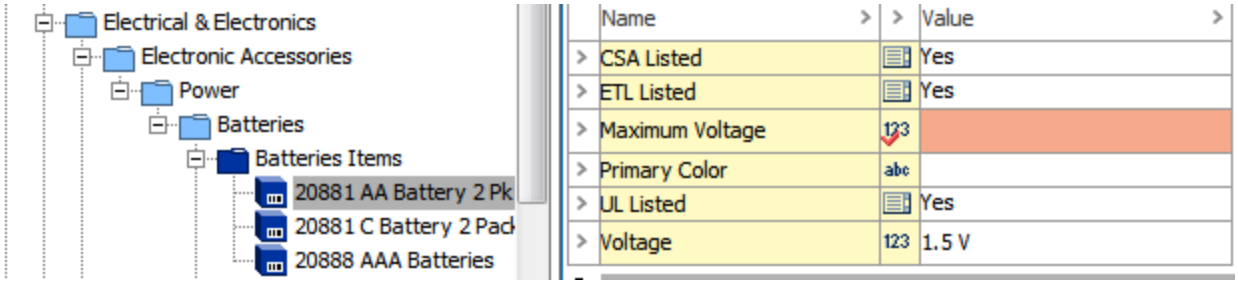
To configure this, update the standard **Mandatory** setting on the attribute to **Yes**.

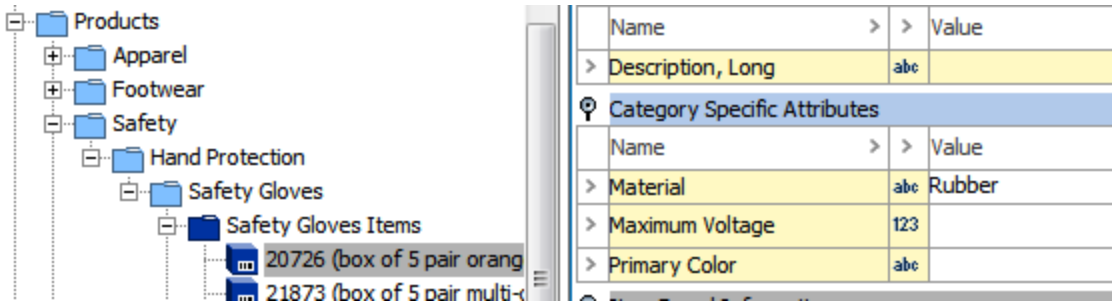


Alternatively, the attribute can be set as mandatory on an attribute link (available on product and classification super types) by clicking the standard **Mandatory** checkbox. Attributes set as mandatory in this way behave in the same manner as described above, though at a more granular level as the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



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





It is important to note that attributes set to mandatory using either of the standard methods described above will cause those attributes to also display as mandatory in the Web UI, regardless of how those attributes might be configured for display in the Web UI.

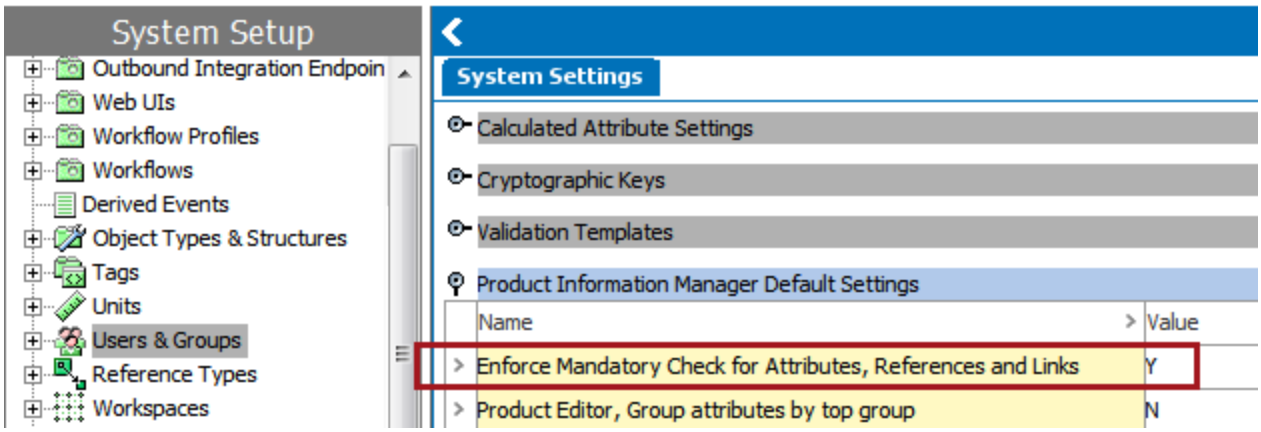
Verify Mandatory Attribute Values Upon Approval

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 earlier in the process, such as when they are actively populated (e.g., within workflows or imports).

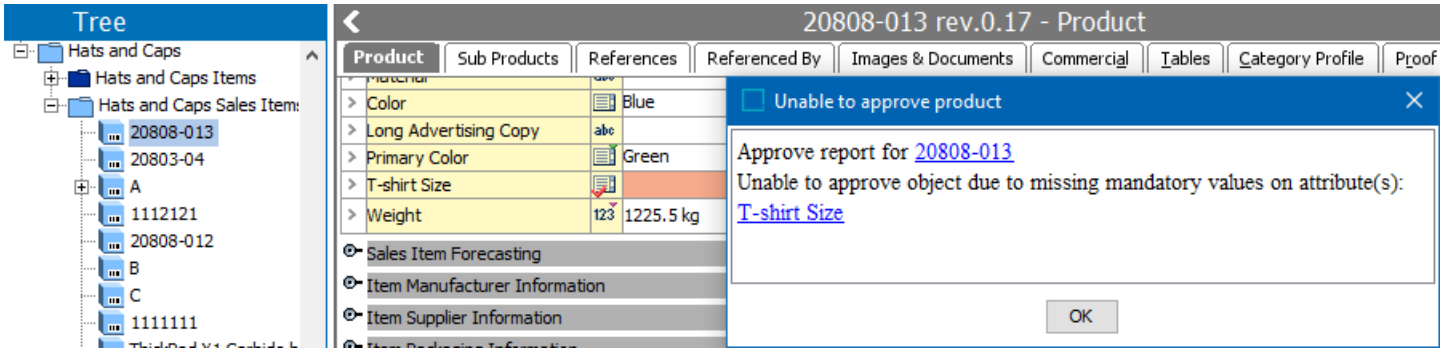
Note: Validity influences the standard mandatory settings, so the mandatory requirement is only enforced for the object types on which the attribute is valid.

For example, if all 'Item' object types require a UPC or GTIN, attributes for those could be made mandatory. 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.

In this case, 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 the 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, since 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). When standard 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. The 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. Attributes marked as mandatory for Smartsheets have their value cells highlighted blue to prompt the user to populate the field. If the user validates the sheet and has not populated the mandatory values, an error is reported.

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

For more information, see the **Mandatory Attributes and References in Smartsheets** topic in the **Excel Smartsheet Format** section of the **Data Exchange** / **Data Formats** documentation.

Web UI Mandatory Attributes

Mandatory attributes can be designated as mandatory either from the standard mandatory settings in workbench or through configuration solely in Web UI. For more information, see the **Mandatory Attributes in Web UI** topic in **Web User Interfaces / Web UI Setup and User Guide** documentation.

Workflow Mandatory Attributes

It is often necessary that certain attributes or attribute groups be set as mandatory for a state or transition in a 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 attribute 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.

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

Single and Multi-Valued Attributes

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

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

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

The following describes the single and multi-valued attributes:

Value Type	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.

Icons

In the editors, icons are used to indicate the validation rule together with the optionality and the link status for the attribute. For more information, see the **Validation Rules** topic.

An attribute with the validation base type number is used as an example below:

Single Valued Attribute	Description
	A single valued integer can be entered.
	A value is inherited from a higher level node. The inherited value can be overwritten.
	Attribute is set as mandatory. An integer must be entered.
	Attribute is set as mandatory. A value is inherited from a higher level in Tree. The inherited integer can be overwritten.

Multi Valued Attribute	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 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 description attributes can be found under the 'Description' flipper at the top of the value editor.

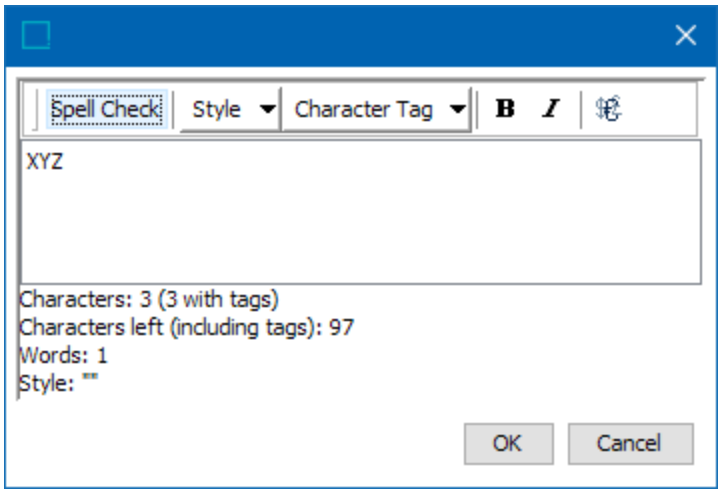
Attribute values can be edited either by selecting from a list, typing a value, or overwriting an existing value.

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 rule of the selected attribute. The value editor, shown below for a text attribute, will assist you in typing a valid value.



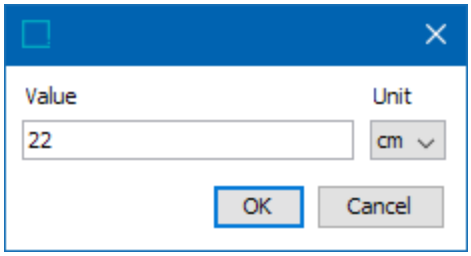
   In addition to adding values to the attribute in the editor, the below operations can also be performed:

   - Perform Spell Check on the values 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**

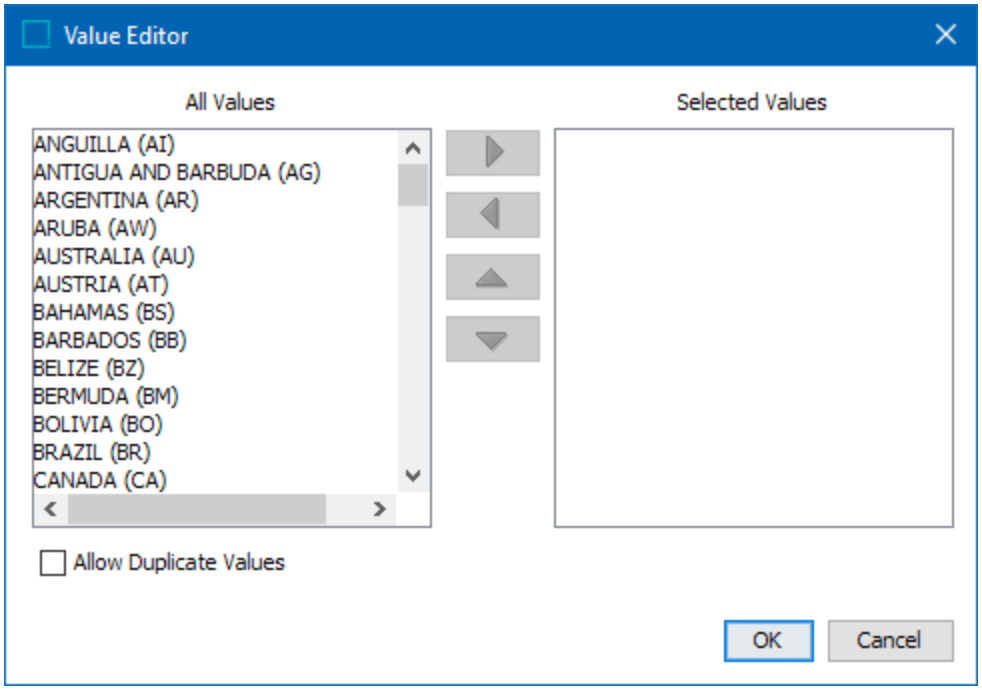
User documentation.

- Apply basic formatting to the text.

A 'Number' validation with units attribute editor is displayed below:



This is a multi-valued attribute editor based on an LOV:



Note: Editing large LOVs, those with more than 5,000 values, includes a Filter Values button(Filter Values). For more information, see the **Working with Large LOVs** section of the **List of Values (LOVs)** topic.

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, click on 'Allow Duplicate Values' checkbox in the Value Editor.

The attribute value has now been edited.

Inherited and Local Values

Values can be:

- Shared by inheritance from a higher product level (e.g., a product family) as defined in the **Inherited Attributes** topic.
- 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

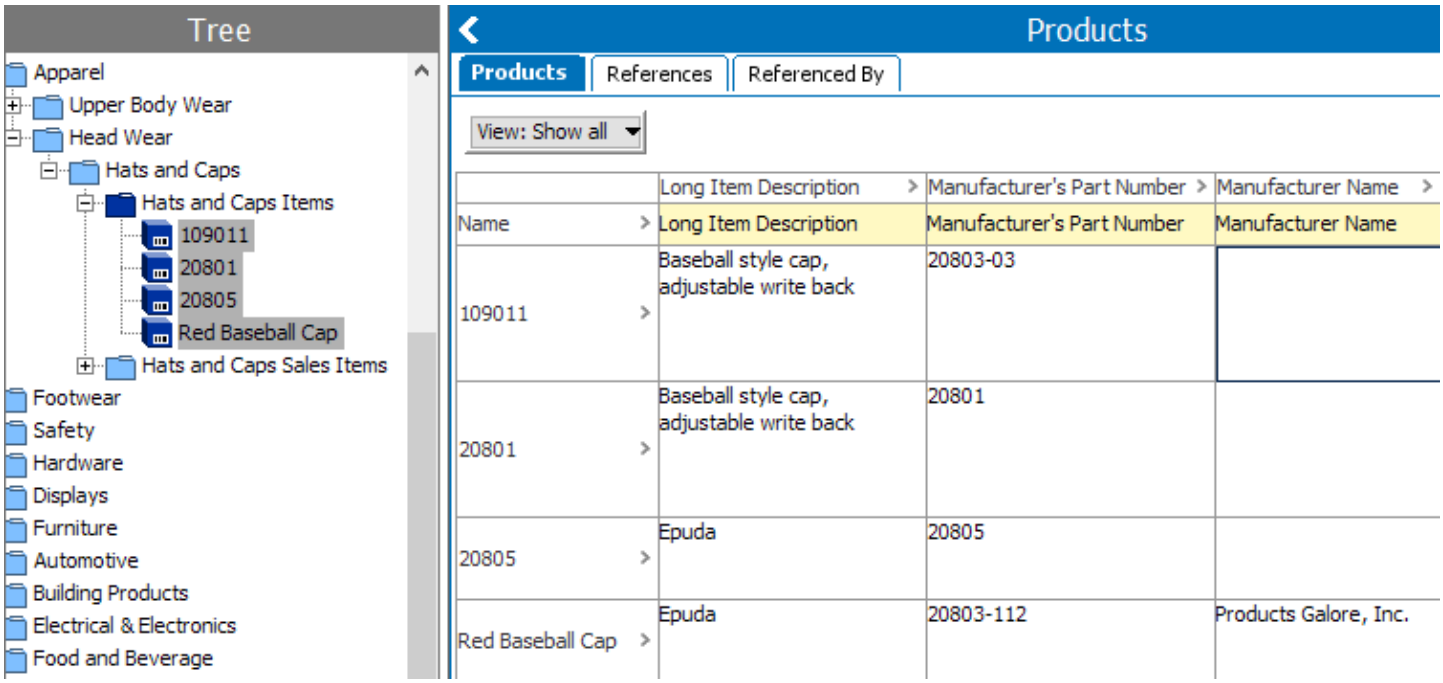
You can set various viewing options such as hiding attributes, rotating tables, and hiding attributes with empty or non-mandatory values. To learn more on how to set filters, see the **Editing Objects in the Tree Tab** topic in the **Getting Started / User Guide** documentation.

Attributes with Units

Below is a description of attributes with units:

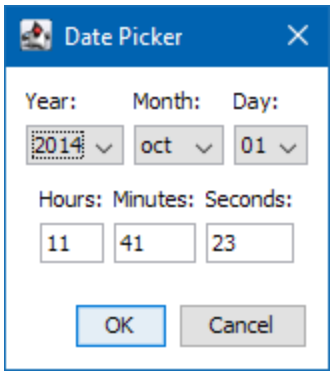
Attribute	Description
Attribute with default Unit	If the attribute has a default unit, it will appear automatically in the field after pressing <F2>.
Attribute with Units	If the attribute has units applied, you can either key in the relevant unit, or press <Alt> + <F2> to select a unit from the list.
Attribute is Numeric Text or Numeric Text (exclude tags) validated	If the attribute has 'Numeric Text' or 'Numeric Text (exclude tags)' 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 objects using the Ctrl key on the keyboard and then add values to the objects in the editor.



Editing 'ISO Date' 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.



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

   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:**

  Description attribute values do not inherit down to subordinate nodes. However, a calculated attribute can be used to display a value of a description attribute from node(s) above the currently selected node.

See the **Validation Rules** in the **Attributes** documentation for more information.

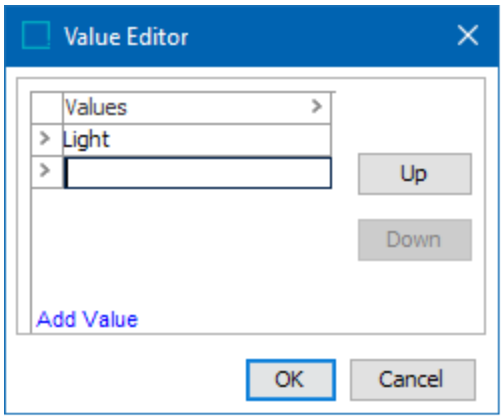
Editing Multi-Valued Attributes

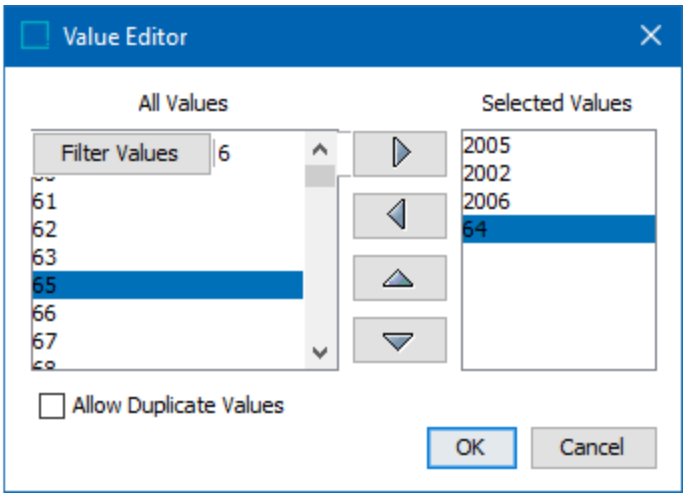
To edit the value of an attribute on an object, navigate to the desired object in **Tree** and select the value field of the relevant attribute.

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: 

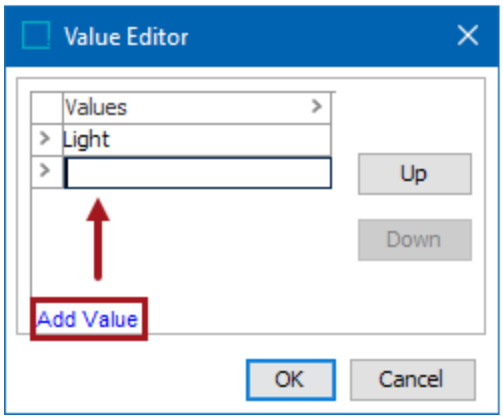
1. In **Tree**, select the relevant node to open an editor.
2. Depending on the editor, click the **Product** tab, **Classification** tab or **Images & Documents** tab.
3. Double-click the value field of the relevant attribute. The **Value Editor** dialog box displays, as shown below for a multi-valued text attribute and a multi-valued large LOV attribute.

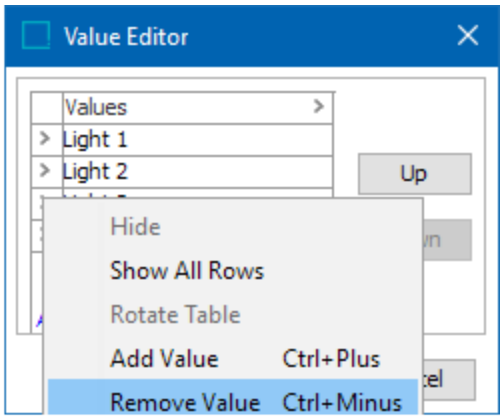




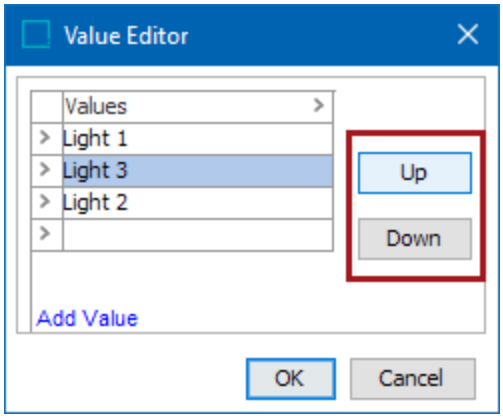
Note: Editing large LOVs, those with more than 5,000 values, includes a Filter Values button(Filter Values). For more information, see the **Working with Large LOVs** section of the **List of Values (LOVs)** topic.

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.'

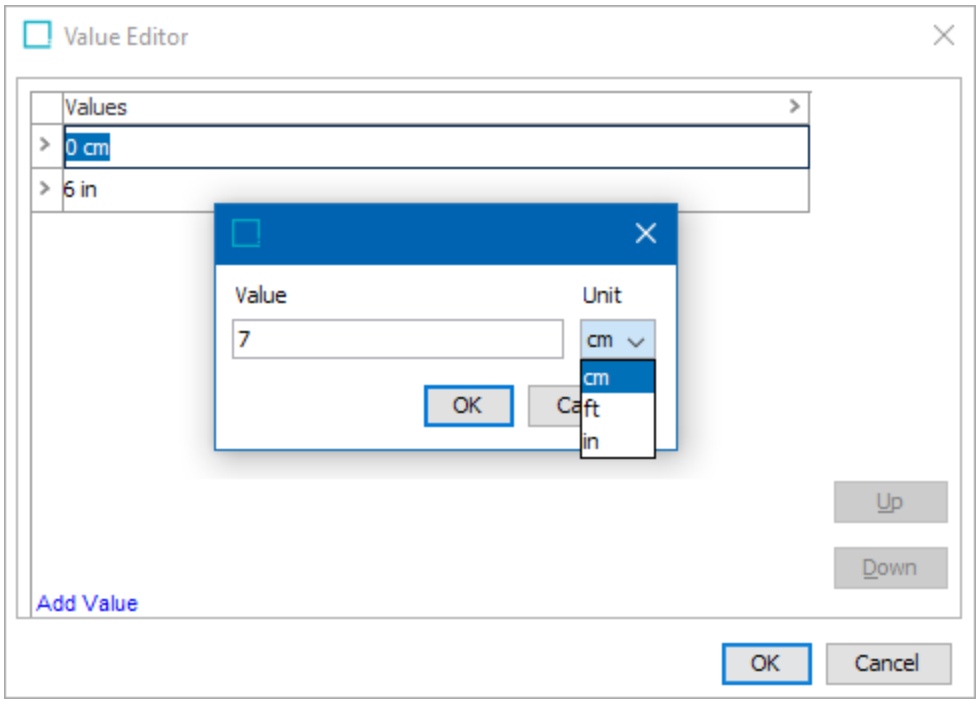




5. It is possible to select in which order the values of the multi-valued attribute should appear. the order can be changed by using the buttons Up and Down.



6. Click **OK** 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.


Value Editor
Values
0 cm
6 in
Value
Unit
7
cm
cm
ft
in
OK
Ca
Up
Down
Add Value
OK
Cancel

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.

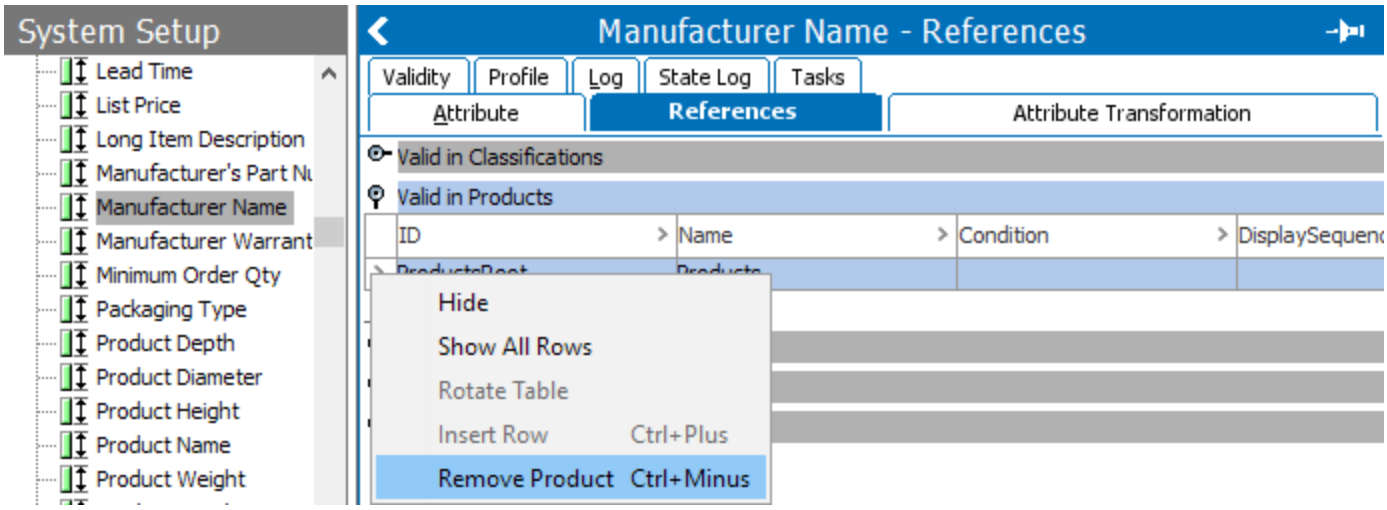
If an attribute with a value is deleted, or unlinked, from a product node, the attribute becomes an 'Orphan Attribute', and will keep values for the attribute. If the attribute is linked to the product again, the value will also appear again.

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

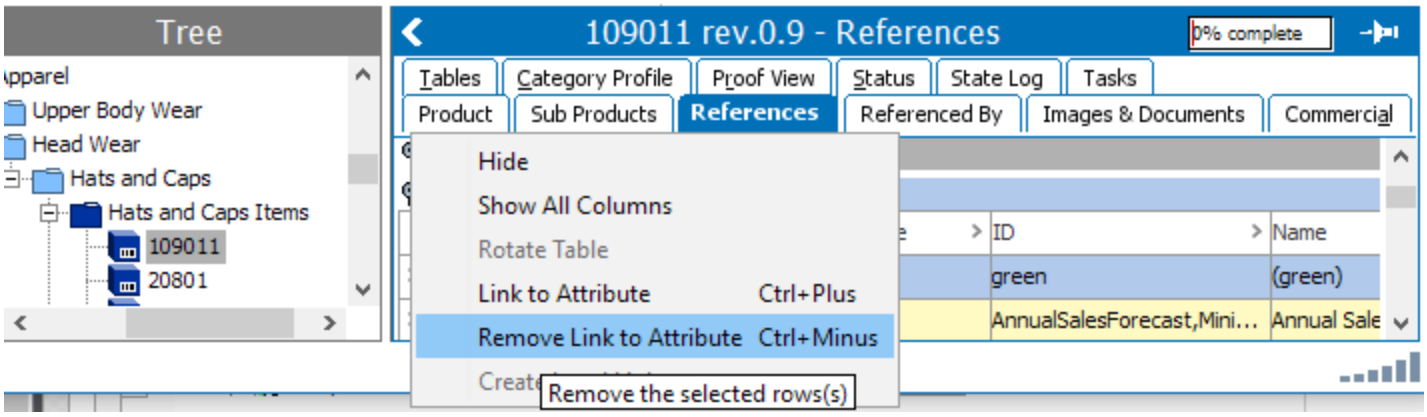
1. In **System Setup**, select the relevant attribute to open the editor.
2. Open the **References** tab where the linked objects are listed.
3. Click the **[>]** next to the desired object, and then click **Remove**.



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.

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.

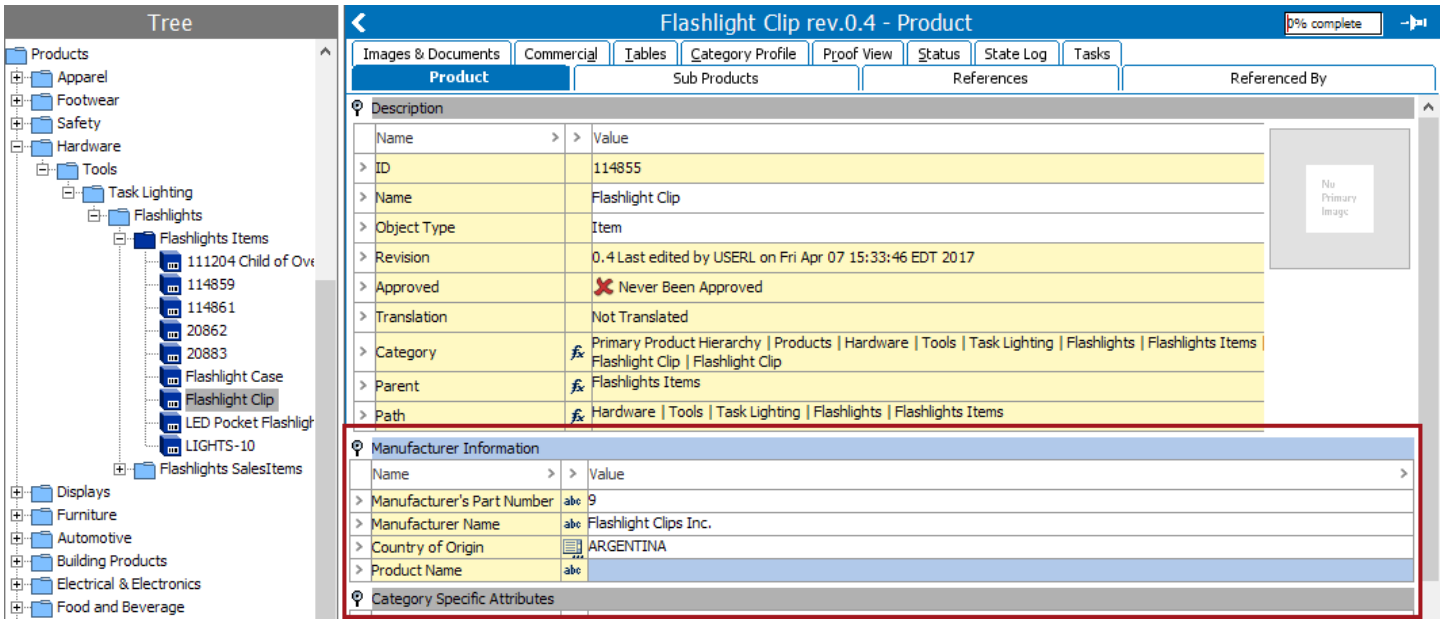
Specification	Example
Physical specification	• Color • Length • Weight • Type of material • Capacity
Feature	• Price • Country of manufacturer

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 **Linking 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 Ove
114859
114861
20862
20883
Flashlight Case
Flashlight Clip
LED Pocket Flashligh
LIGHTS-10
Flashlights SalesItems
Displays
Furniture
Automotive
Building Products
Electrical & Electronics
Food and Beverage
Flashlight Clip rev.0.4 - Product
0% complete
Images & Documents
Commercial
Tables
Category Profile
Proof View
Status
State Log
Tasks
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
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.

Validity on Specification Attributes

Specification attributes can only be made valid for product object types.

Setting validity for a specification attribute identifies the product object types that can use the attribute.

Note: The STEPXML format identifies selected Validity object types within the UserTypeLink tag using the UserTypeID element.

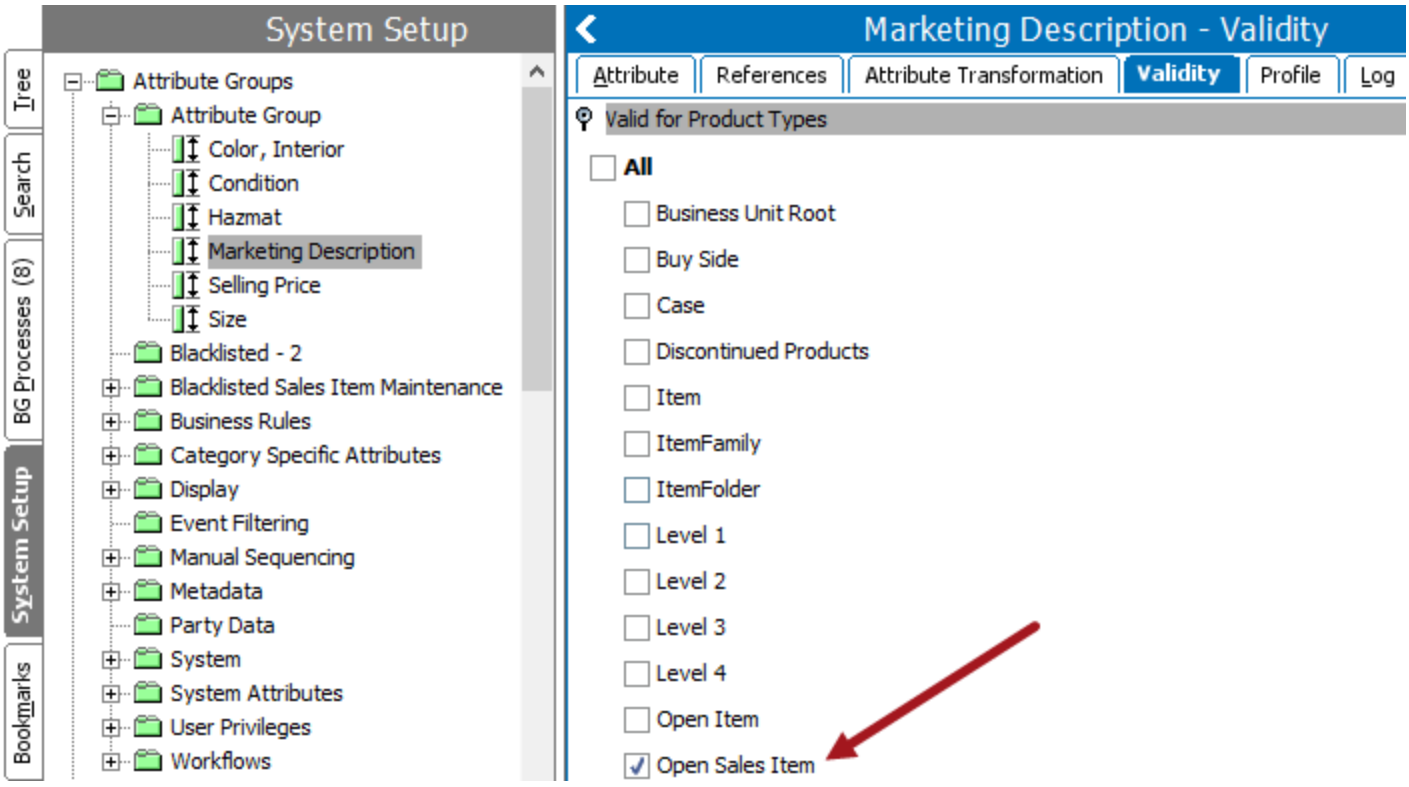
1. In System Setup, expand **Attribute Groups**, expand the relevant group, and select the relevant attribute.

   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 resides.

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).

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 '<.' (Values for numeric searches with no units are assumed supplied in the default unit for the attribute. For example, if kg is the default unit for the Weight attribute, 'Weight = 1' will match '1 kg' and '1000 g'.)
- 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.

A unit is housed in a unit group and can include a unit conversion. Both of these are discussed below, as well as information for maintaining each of these elements.

Unit Groups

A unit group is a grouping of compatible units. It contains units that apply to the same measurement. This makes it 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 are included in the following table.

Unit Group	Units
Temperature	• Celsius • Fahrenheit • Kelvin
Dimension	• mm • cm • meters • inches

Unit Group	Units
	• feet • yards
Volume	• cm3 • m3 • inch3

For 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

A unit conversion specifies the rules for conversion from one unit to another. One of the units must be a base unit. It is possible to define a conversion rule between two compatible units. Unit conversions are used when, for example, exporting data and when searching for Attribute Values. Examples of unit conversions are included in the following table.

Time(s)	ID	Name	Conversion
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, inspect the available units carefully before doing so. The pre-defined units are also provided with standard unit conversions

Conversion Factor and Conversion Offset

A unit conversion is specified in terms of a conversion factor and conversion offset.

Note: A conversion factor and conversion offset must be typed with minimum of one decimal and must never be empty. Type '0.0' to indicate a factor or offset of '0' (zero).

The following example shows the conversion rule for temperature conversion from Celsius into Fahrenheit using the formula: Temperature Fahrenheit = (1.8 * Temperature Celsius) + 32

Unit	Unit Conversion Factor	Conversion Offset
Fahrenheit (base unit)		
Celsius	1.8	32.0

As an explanation of this example, 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.

Additional Information

For information on creating or deleting the elements included in a unit, see the following topics:

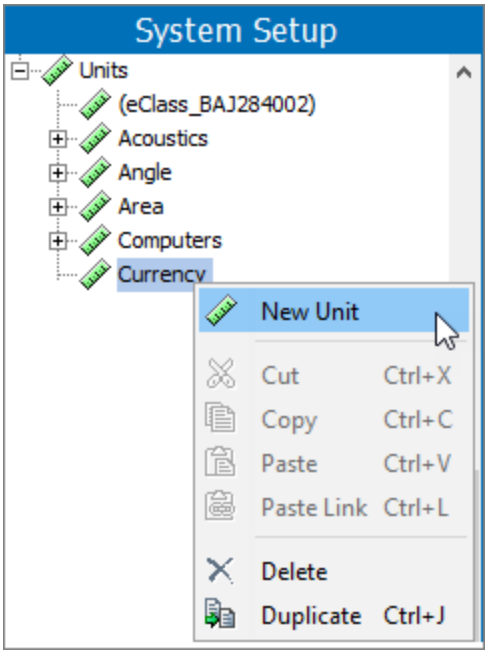
- Maintaining a Unit
- Maintaining a Unit Conversion
- Maintaining a Unit Group

Maintaining a Unit

Units can be created and deleted as necessary. Each maintenance option is defined below.

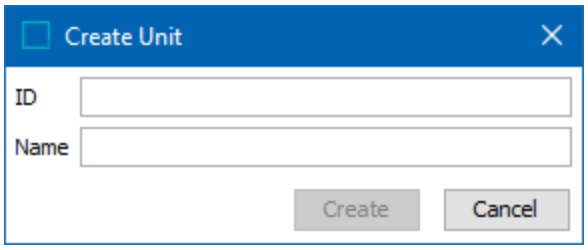
Create a Unit

1. Go to System Setup, open the Units node and click the relevant Unit Group. Right-click the relevant Unit Group and click the **New Unit** option.



Note: To create a new unit group, see the **Maintaining a Unit Group** topic in this guide.

2. In the Create Unit dialog:
   - Enter an **ID** for the new unit.
   - Enter a **Name** for the new unit.
   - Click the **Create** button.

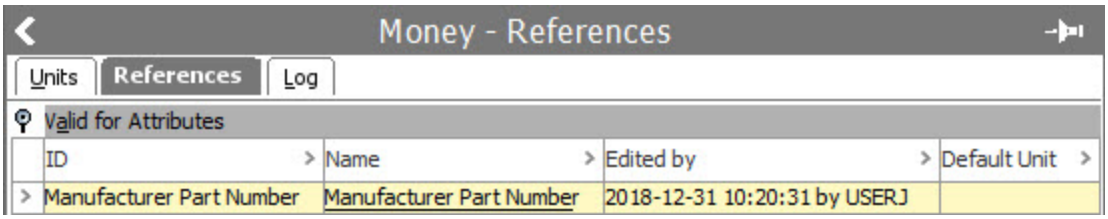


The newly created unit displays within a unit group. The unit can now be selected for attributes.

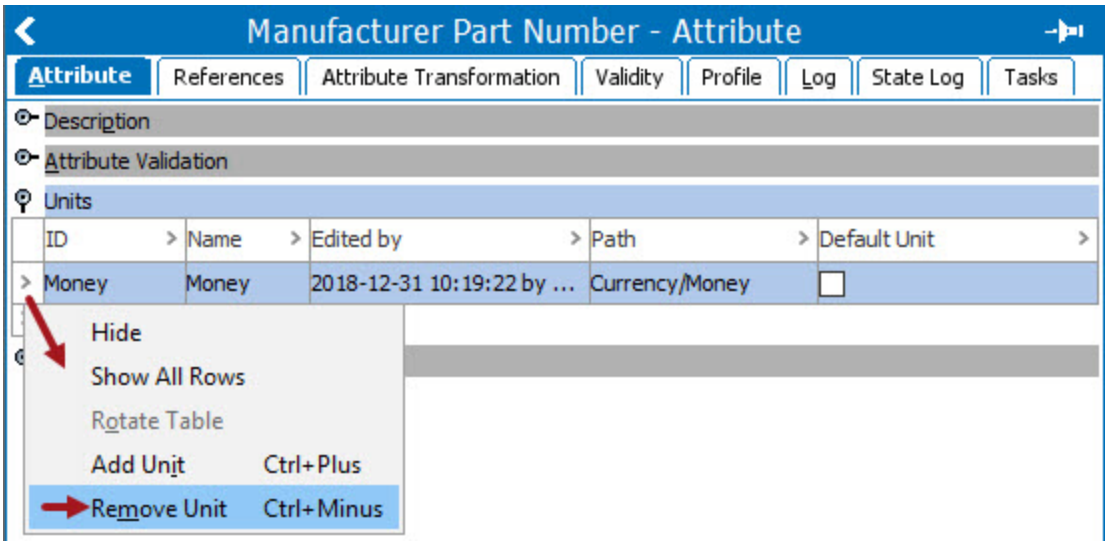
Delete a Unit

A unit group that contains 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. In System Setup, click the Units node, click the units group, and then click the unit to be deleted.
2. On the units editor, click the References tab to display the 'Valid for Attributes' flipper.

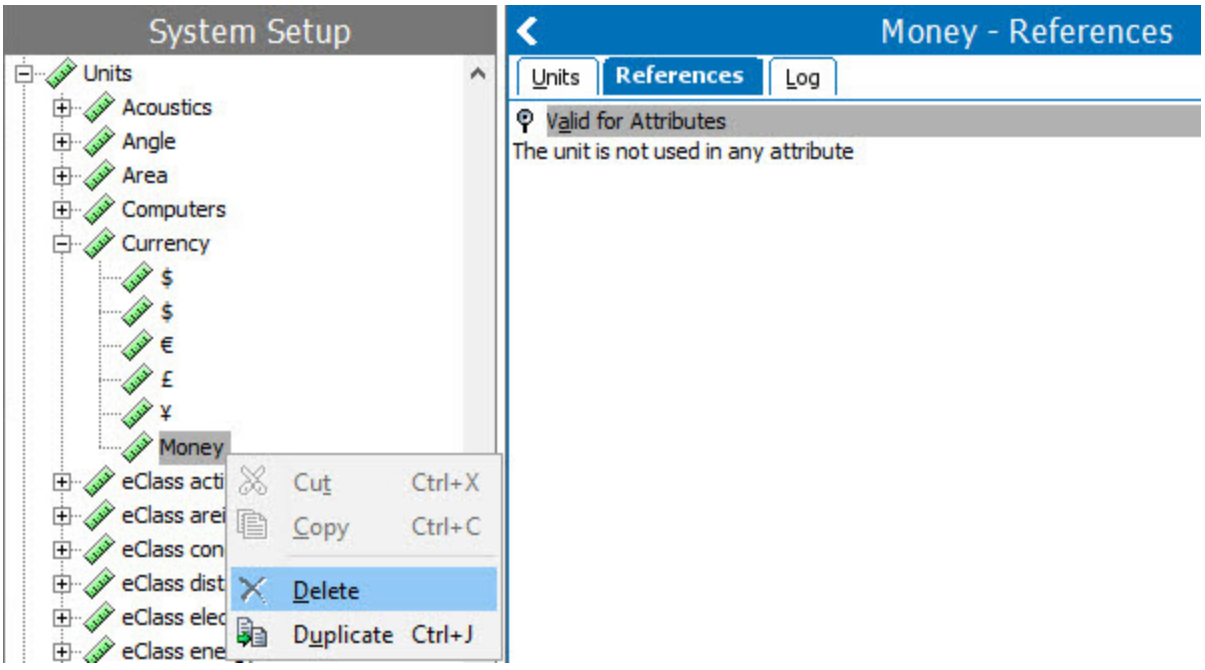


3. For each attribute displayed, use the following steps to remove the unit from the assigned attributes:
   - In the unit editor, click the attribute link in the Name column (as shown above) to display the attribute editor.



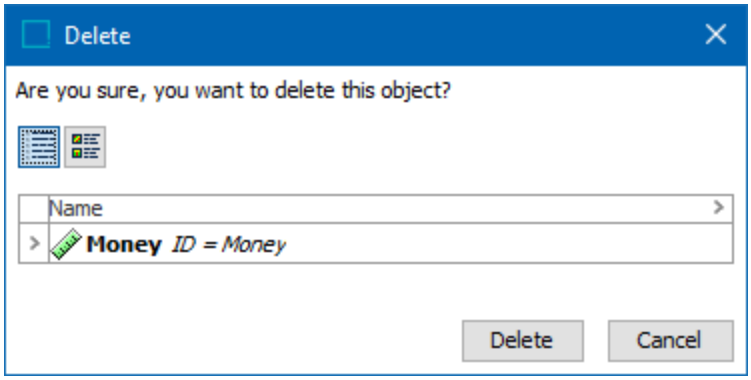
- On the Attribute tab, open the Units flipper.
- Right-click the row of the unit to be deleted and click the 'Remove Unit' option.

4. In the unit editor, when no attributes are displayed on the References tab for the 'Valid for Attributes' flipper, right-click the unit node and click the **Delete** option.

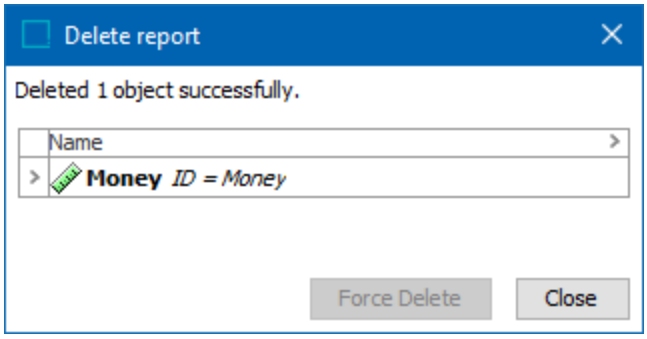


5. On the Delete warning dialog, choose an action:

- Click the Delete button to delete the unit.
- Click the Cancel button to return to the unit editor.



If the unit was deleted, the confirmation dialog is displayed.



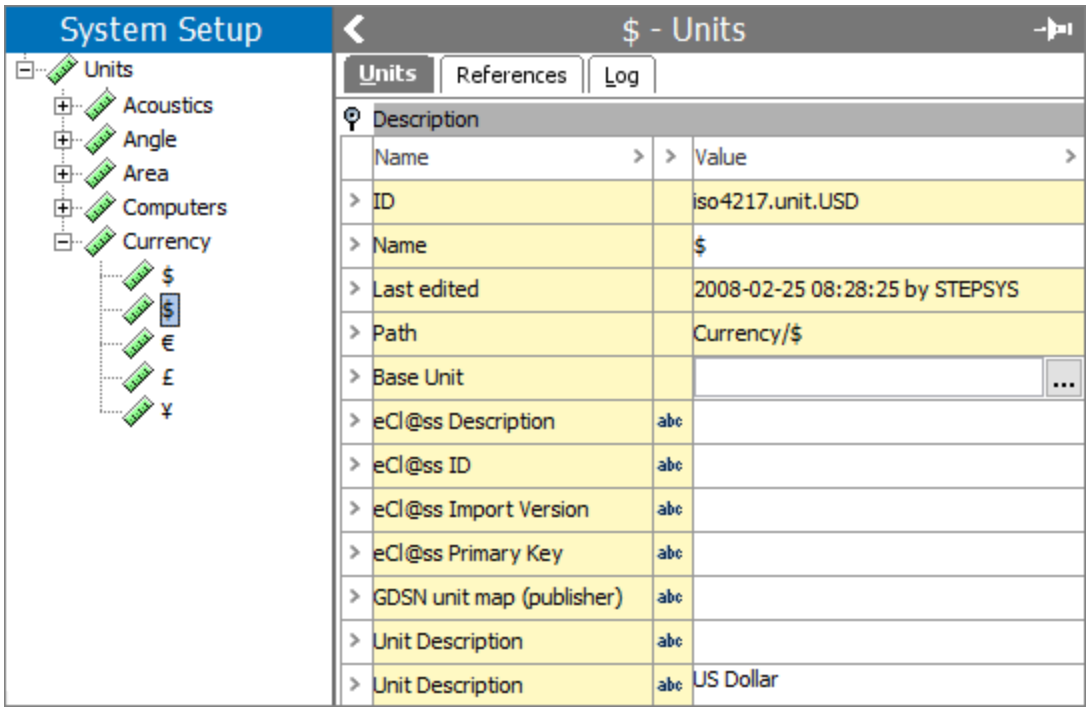
Maintaining a Unit Conversion

A unit conversion provides a way to 'translate' one unit to another. Standard units are defined with standard conversions.

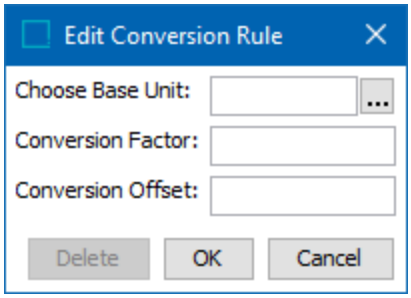
Unit conversions can be created and deleted as necessary. Each maintenance option is defined below.

Creating a Unit Conversion

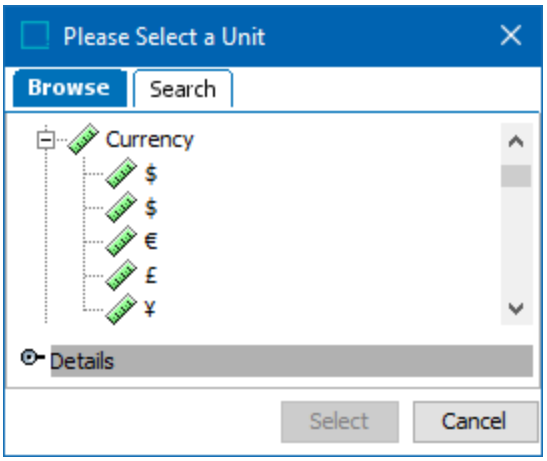
1. Go to System Setup, click the Units node and expand the relevant unit group.
2. Click a unit to display the units editor as shown below.



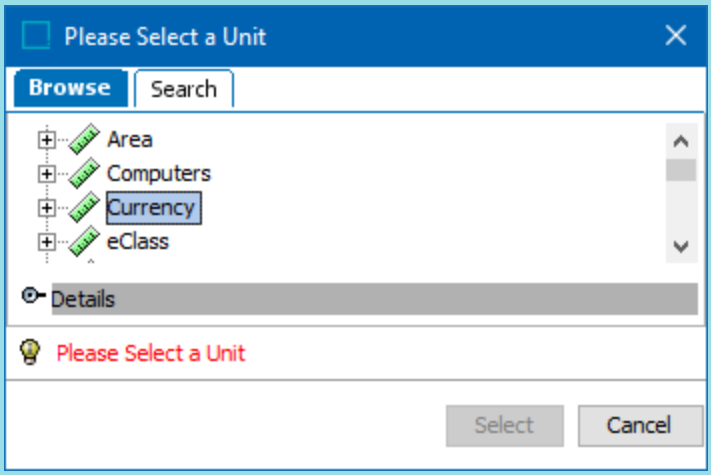
3. On the Units tab in the Base Unit parameter, click the ellipsis button (...), to display the **Edit Conversion Rule** dialog.



- For the **Choose Base Unit** parameter, click the ellipsis button (...), to display the **Please Select a Unit** dialog. Use the Browse or Search tabs to find and select the relevant unit, and then click the **Select** button.



Note: If a unit group is selected, a waring is displayed to select a unit (as shown below).



- In the **Conversion Factor** parameter, type the factor.
- In the **Conversion Offset** parameter, type the offset.

For example: 1 Unit = Conversion Factor * Base Unit + Offset

A formula might look like one of the following:

1 week = 7.0 days + 0.0

1 cm = 0.01 m + 0.0

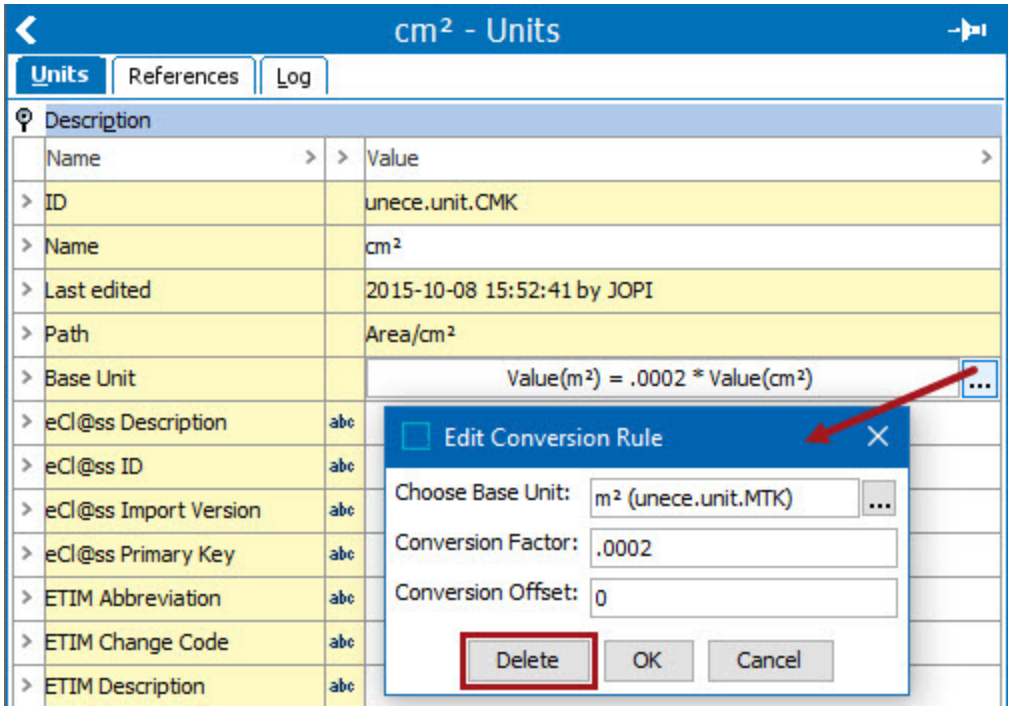
For more information, see the **Conversion Factor and Conversion Offset** section of the **Units** topic.

4. Click the **OK** button on the Edit Conversion Rule dialog to create the conversion rule between the two units displayed in the Base Unit field.

> **Important:** 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.

Deleting a Unit Conversion

1. Go to System Setup and click the Units node, expand the relevant unit group and click the relevant Base Unit where the conversion rule is to be removed.
2. On the Units tab of the units editor, click the ellipsis button (...) on the Base Unit parameter to display the Edit Conversion Rule dialog.



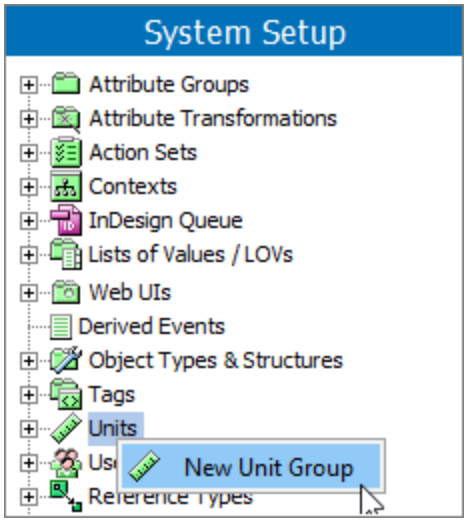
3. Click the **Delete** button. The conversion rule is removed.

Maintaining a Unit Group

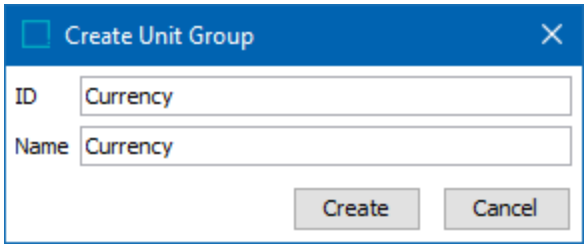
Unit groups can be created and deleted as necessary. Each maintenance option is defined below.

Creating a Unit Group

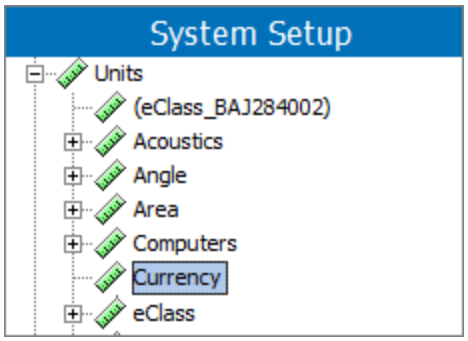
1. In System Setup, click the Units node, right-click the Units node, and then click the **New Unit Group** option.



2. In the Create Unit Group dialog:
   - Enter an **ID** for the new unit group.
   - Enter a **Name** for the new unit group.
   - Click the **Create** button.

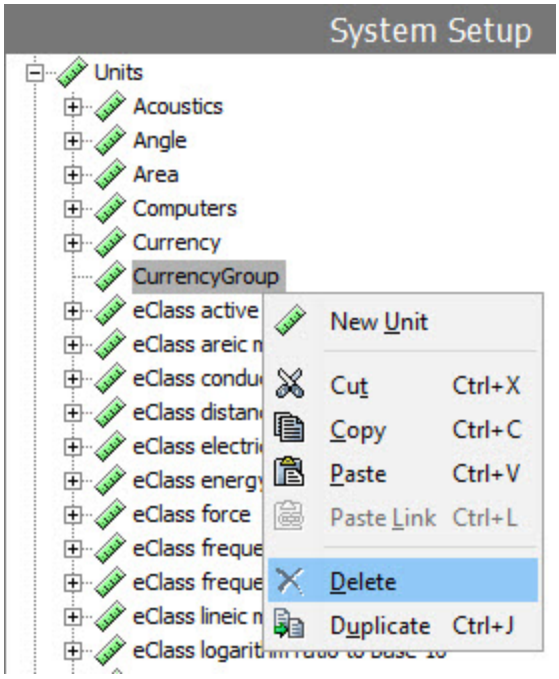


The newly created unit group displays in System Setup as shown below.

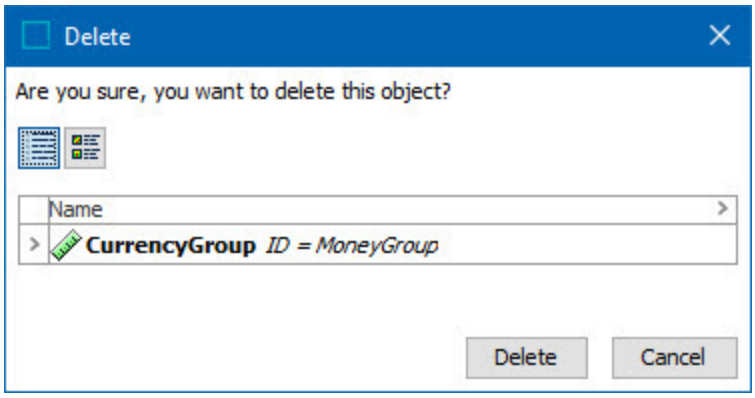


Deleting a Unit Group

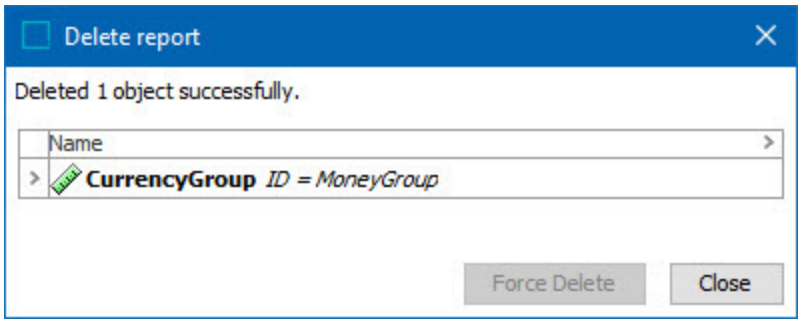
1. In System Setup, click the Units node, and click the units group to be deleted.
2. Verify no units exist within the group. To delete the exiting units, see the **Deleting a Unit** section of the **Maintaining a Unit** topic.
3. When no units are displayed for the group, right-click the unit node and click the **Delete** option.



4. On the Delete warning dialog, choose an action:
   - Click the Delete button to delete the unit group.
   - Click the Cancel button to return to the unit group editor.



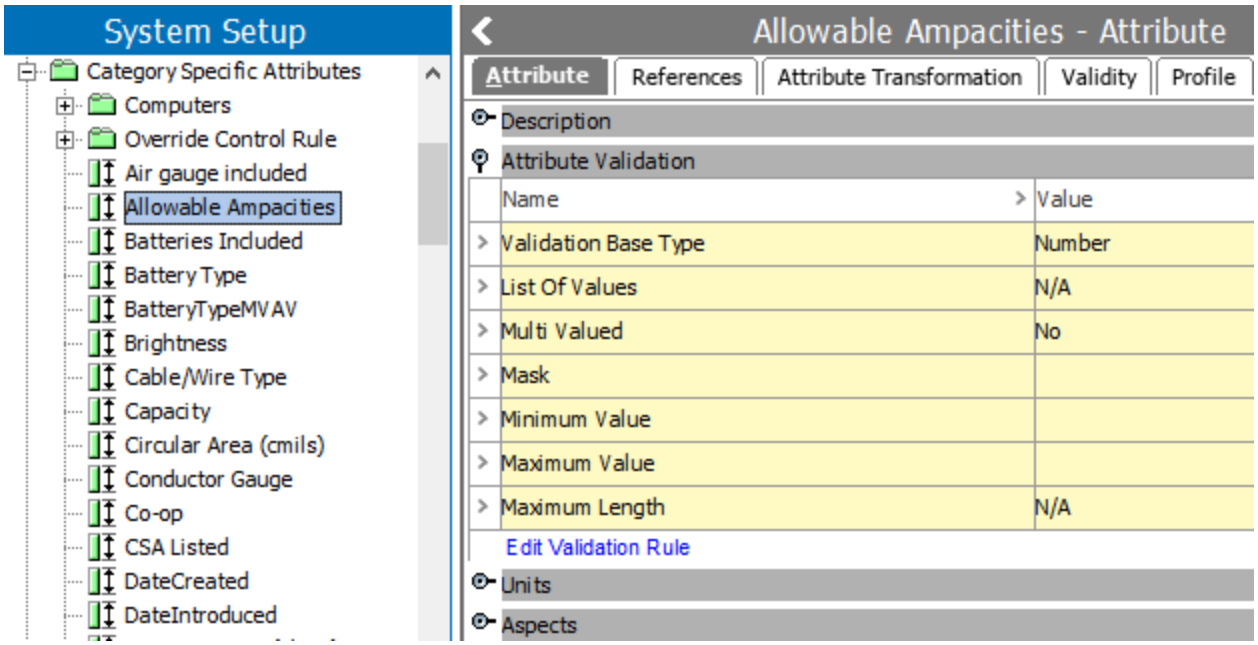
If the unit group was deleted, the confirmation dialog is displayed.



Validation Rules

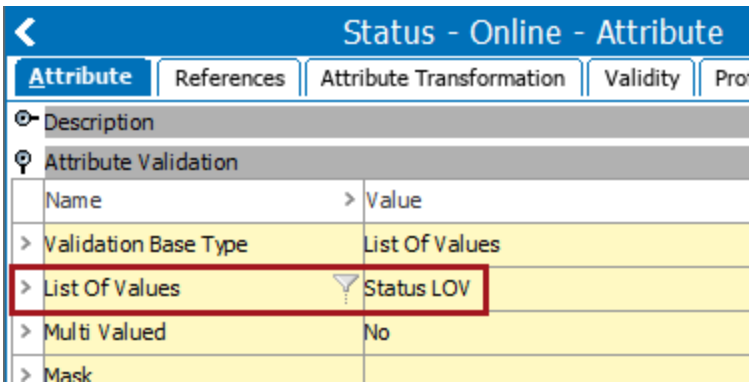
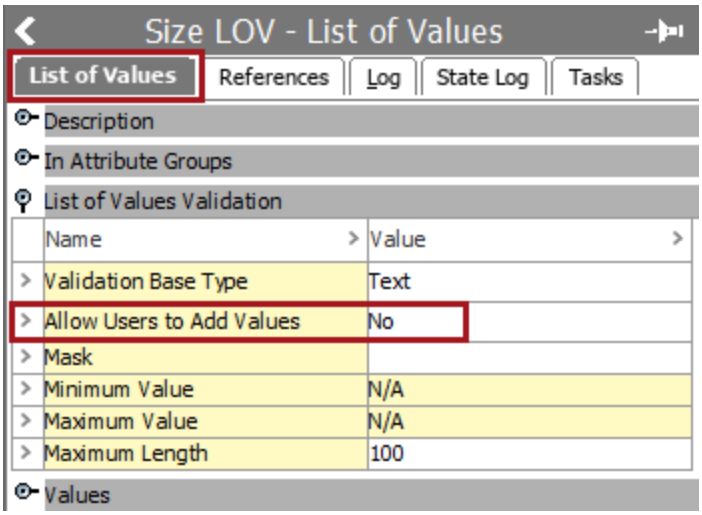
A validation rule defines the allowed content of a specific attribute or list of values and can include the following rule options.

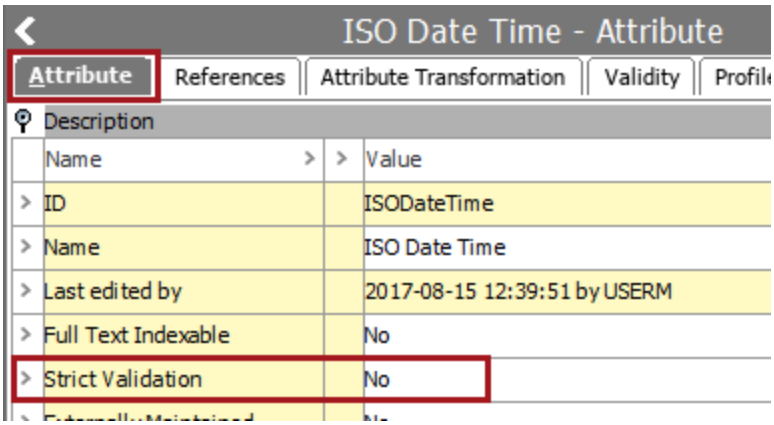
In System Setup, most validation rules are set for attributes under the Attribute Validation flipper and the available is based on the Validation Base Type selected.



Other rules are found under the Description flipper.

Validation Rule	Description
Validation Base Type	The basic data type, such as text, integer, number, date or List of Values (LOV). For more information, see the **Validation Base Type** topic. **Note:** Validation rule options change according to the selected Validation Base Type.
List of Values	If Validation Base Type is LOV, a reference to the actual LOV is shown. For more information, see the **List of Value (LOVs)** topic.

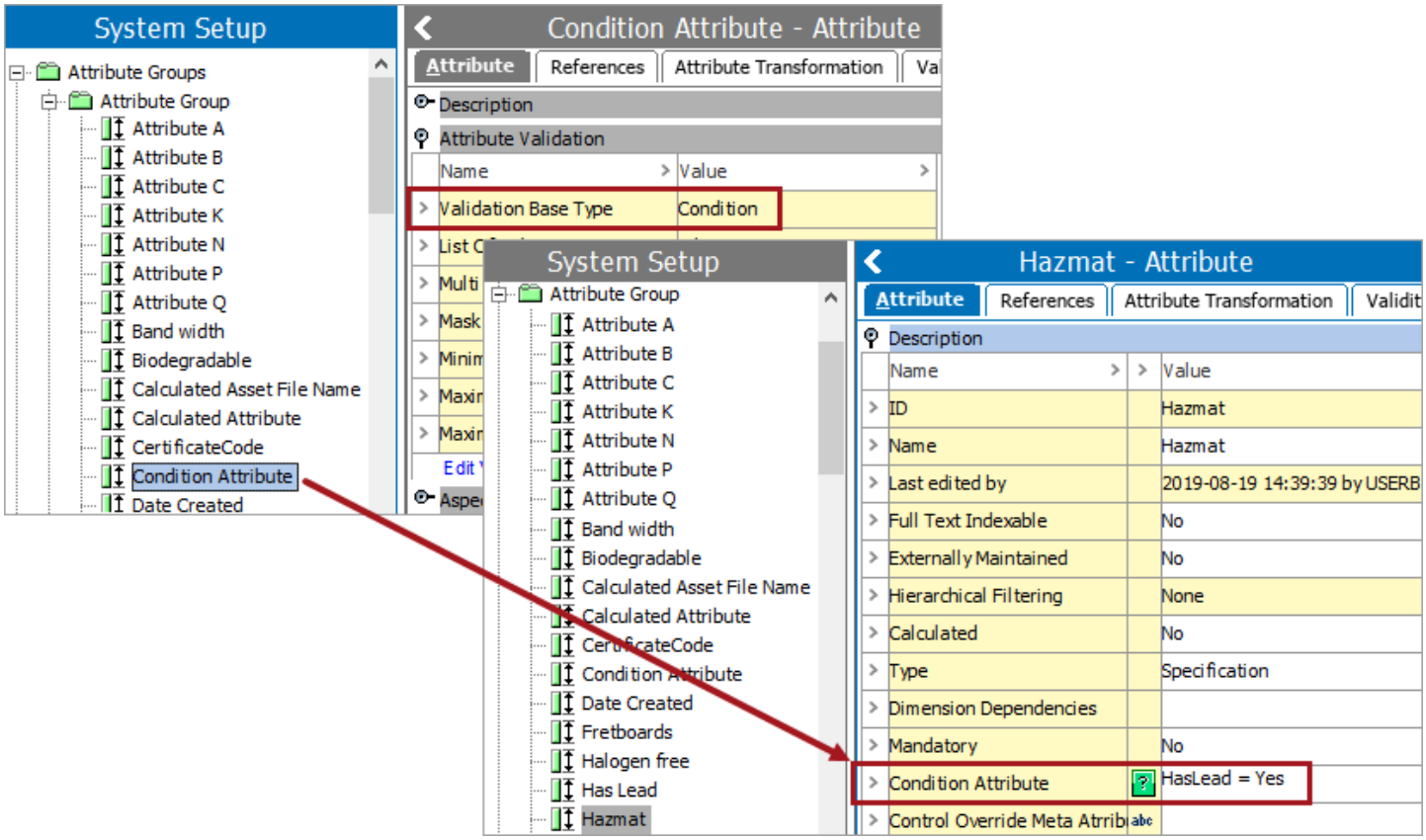
Validation Rule	Description
	
Multi Valued	For attributes only, multiple values on a single attribute are allowed / not allowed. For more information, see the **Single and Multi-valued Attributes** topic.
Allow Users to Add Values	For LOV object types only, values can be added by users outside of System Setup. 
Mask	An input mask defines the pattern of characters allowed when entering a value. **Note:** Do not make any entry for the Mask for attributes with the Base Validity of ISODATE or ISODATETIME. Those attributes have their own predefined internal Mask. Any Mask entry that you make will not only be redundant it will cause an error to be displayed when attempting to enter a date.
Minimum	E.g.; '5' means that the minimum numeric value is 5.

Validation Rule	Description
Value	
Maximum Value	E.g.; '7' means that the maximum numeric value is 7. Used with the Minimum example above, the value must be between 5 and 7.
Maximum Length	Only for text-based 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. For example, if you style a text bold, then the characters in the style tags <bold> and </bold> are also counted toward the maximum length. For more information, see the **Tags** topic. **Note:** In Web UI, the text box and rich text editor on a Node Editor, as well as the inline editor and multi-value text attribute on a Node List, will display with a text editor count display to reflect the number of characters added to the field.
Strict Validation	For 'ISO Date' or 'ISO Date and Time' attributes and LOVs only, the Strict Validation setting is located on the Attribute tab under the Description flipper.  Setting this field to **Yes** ensures that only the industry standard formats of 'ISO Date' and 'ISO Date and Time' are accepted: YYYY-MM-DD for 'ISO Date' and YYYY-MM-DD HH:MM:SS for 'ISO Date and Time.' These are predefined internal Masks and do NOT require any entry in the Mask field. In fact it is important that the Mask entry is left blank. If Strict Validation is set to **No**, then the system will accept dates and times in non-standard ISO formats. This was only implemented for backwards compatibility in older STEP systems and is not recommended for systems using STEP 6.0 and later. For these non-standard formats,

Validation Rule	Description
	searching is problematic and not consistent. Therefore, recommended practice is to *always* set Strict Validation to **Yes**. **Note:** In the workbench, if the user has a locale applied and the 'Localize dates' option (located on the User object in System Setup under System Settings > **PIM GUI Localization**) is set to 'Y,' then that user may *also* enter 'ISO Date' and 'ISO Date and Time' date values using the locale's format. Example: if an attribute has a validation of 'ISO Date' and the user's locale is set to 'English (United States) - en_US,' then a date may be entered either as '2025-06-13' or 'Jun 13, 2025', and an attribute with a validation of 'ISO Date and Time' may have an entry of '2025-03-27 18:30:30' or 'Mar 27, 2025 6:30:30 PM.' Both types of entry will be accepted in the workbench, and for either type of entry, the workbench display will show the value in the locale's format. But, the value stored in STEP will always be the ISO format. In addition, searching on dates using > or < or = will be consistent and accurate. If you have been using the Strict setting of **No** and you have entered dates that are not standard ISO Dates or Times, you can correct your entries to now conform to the ISO standard. By selecting 'Strict = **Yes**' the system checks to see if you have any non standard date / time entries and allows them to be auto-corrected. If you answer **Yes** the system launches a Background Process to remedy any and all non-ISO standard dates and times.

Validation Base Type

In the object editors, the validation base types are indicated by icons as shown below. For example, the description attribute 'Condition Attribute' has the 'Condition' validation base type. 'Condition Attribute' is valid on the 'Hazmat' attribute, and displaying that editor shows the condition icon ( ).



Validation Base Type	Icon	Description	Example
Condition		Allows applying conditions.	[AttributeID] = [Value1];[Value2]
Date		Allows a date to be entered according to a predefined input mask,	The mask is:

Validation Base Type	Icon	Description	Example
		indicating the valid pattern of the value. Since the mask is predefined, a Mask value is *not* entered within the Attribute Validation settings. Units cannot be applied to 'Date' attributes. **Note:** This validation type has been superseded by the two validations of 'ISO Date' and 'ISO Date and Time'. Recommended practice is to use one of these newer validations and to *not* use the 'Date' validation. If you *do* use this validation, you cannot apply a locale to the value on data exports, and, in many cases, searching for objects by date values will be inconsistent and inaccurate.	DD-MON-YYYY. A valid date is e.g., 15-JAN-2025
Embedded Number		Prefix, number, unit, and suffix are stored as separate entities. • **Required:** number Numbers can be searched on without regard for the surrounding text. • **Optional:** prefix, unit, and suffix (shown as '*none*' in the following list) In the examples: • Prefixes are '±', '≈', and '≤'. • Numbers are '2', '35', and '0.2'. • Units are 'g', '°C', *none*. • Suffixes are 'of protein', *none*, and '% THD'. The following calculated attribute functions can extract different pieces of the Embedded Number entry: • GETEMBEDDEDPREFIX(value) • GETEMBEDDEDVALUE(value) • GETEMBEDDEDSUFFIX(value) For more information, see the **Calculated Attributes** topic.	±2g of protein ≈35°C ≤0.2% THD

Validation Base Type	Icon	Description	Example
Fraction	3/4	Allows numbers including dash (-) and slash (/).	Valid values: 1/2, 1-1/2 Invalid value: 1 1/2
GLN	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 see the **Input Masks** section.	22222220 or 4444444444444
GTIN-8	G8	Validates the length and check digit of the eight digit GTIN.	22222220
GTIN-12	G12	Validates the length and check digit of the twelve digit GTIN.	333333333331
GTIN-13	G13	Validates the length and check digit of the thirteen digit GTIN.	4444444444444
GTIN-14	G14	Validates the length and check digit of the fourteen digit GTIN.	55555555555555
Integer	123	Allows only whole numbers.	Valid values: 5, 26, 199, and -23. Invalid values: .250, 67.34, and 50 1/2
ISO Date	31	Allows a date to be entered according to an internally predefined input mask, indicating the valid pattern of the value. Since the mask is predefined, a Mask value must *not* be entered within the Attribute Validation settings.	The mask is: YYYY-MM-DD A valid date is e.g.,

Validation Base Type	Icon	Description	Example
		Units cannot be applied to 'ISO Date' attributes. **Note:** It is strongly recommended to set Strict Validation to **Y** on any attribute that is 'ISO Date' validated. For more information, see the explanation of 'Strict Setting' in the **Validation Rules** topic. For workbench users with an assigned locale and a setting of 'Y' for 'Localize dates,' values may be entered not only in the ISO format but also by the locale's format. Both types of entry will be accepted, and for either type of entry, the workbench display will show the value in the locale's format, but the value stored in STEP will always be the ISO format.	2025-01-15
ISO Date and Time		Allows a date and time to be entered according to the internally predefined input mask, indicating the valid pattern of the value. Since the mask is predefined, a Mask value must not entered within the Attribute Validation settings. Units cannot be applied to 'ISO Date and Time' attributes. **Note:** It is strongly recommended to set Strict Validation to **Y** on any attribute that is 'ISO Date and Time' validated. For more information, see the explanation of 'Strict Setting' in the **Validation Rules** topic.	The mask is: YYYY-MM-DD HH24:MI:SS A valid date is e.g., 2025-01-15 12:15:40
LOV		LOV must be selected containing the valid values. Units cannot be applied to 'LOV' attributes.	An LOV (List Of Values) will contain the valid values of an attribute, e.g., • Red • Green • Blue
Number		Allows any numeric characters.	Valid values: -15, -24, 78, 1, .34, and 384.5.

Validation Base Type	Icon	Description	Example
			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	12a	All characters (numeric and alpha) are valid. It is also possible to link a Unit. The numeric text validation base type should be used for numerical data that may at times have non-numerical values. See the cell to the right for examples. **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') • 'cm' becomes 'cm cm' (value: 'cm', unit: 'cm')	Valid values: • 14 mm • 269 • N/A
Numeric Text (exclude	12a	Functions like the Numeric Text validation base type except style tags are excluded from character counts.	

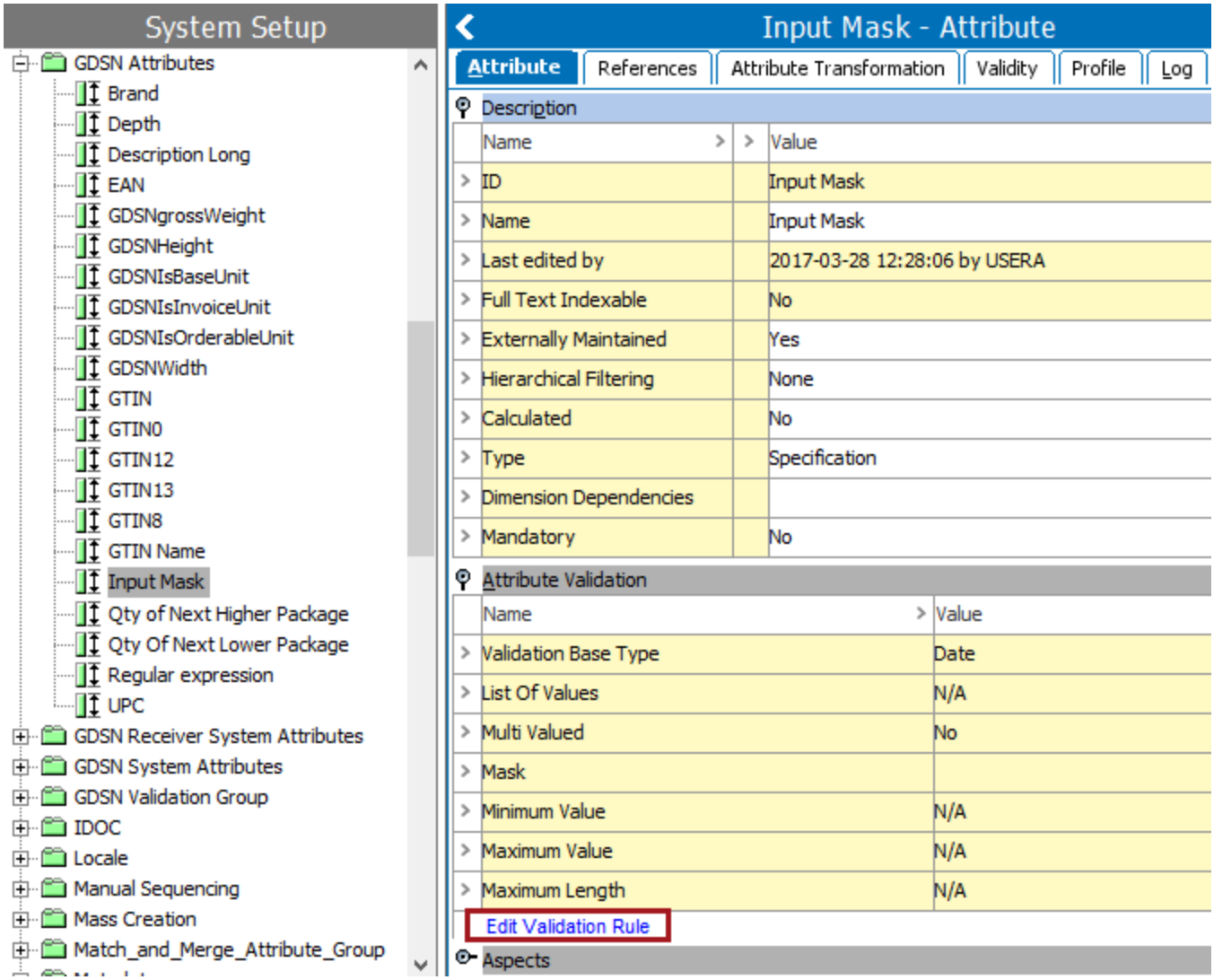
Validation Base Type	Icon	Description	Example
tags)		Additional tag exclusions are as follows: • Character tags, such as less than (<lt/>) or greater than (<gt/>), are counted as 1 character. • Footnote tags and footnote content are counted as 1 character. • Inline references are counted as 0. • Hyperlink tags are counted as 0, but hyperlink text is included in the character count. For example, the phrase 'This is **important** information' contains 29 characters on output, which is acceptable for an attribute with a maximum length of 30 that uses the Numeric Text (exclude tags) validation base type. However, if using the Numeric Text validation base type, the character count would be 42, since the value is stored in STEP as 'This is <bold>important</bold> information', with the bold style tags expanding the character count to 42.	
Regular Expression	RE	Allows for the applying of regular expressions as a validation base type. For more information, see the **Regular Expression** topic in **Resource Materials** online help. 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	abc	All characters (numeric and alpha) are valid. Units cannot be applied to 'Text' attributes.	
Text (exclude tags)	abc	Functions like the Text validation base type except style tags are excluded from character counts. Additional tag exclusions are as follows:	

Validation Base Type	Icon	Description	Example
		• Character tags, such as less than (<lt/>) or greater than (<gt/>), are counted as 1 character. • Footnote tags and footnote content are counted as 1 character. • Inline references are counted as 0. • Hyperlink tags are counted as 0, but hyperlink text is included in the character count. For example, the phrase 'This is **important** information' contains 29 characters on output, which is acceptable for an attribute with a maximum length of 30 that uses the Text (exclude tags) validation base type. However, if using the Text validation base type, the character count would be 42, since the value is stored in STEP as 'This is <bold>important</bold> information', with the bold style tags expanding the character count to 42.	
URL	URL	Allows applying a URL.	**Valid** http://www.stibo.com https://www.stibo.com http://stibo.com **Invalid** www.stibo.com stibo.com
Validation Templates	RE	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**.	

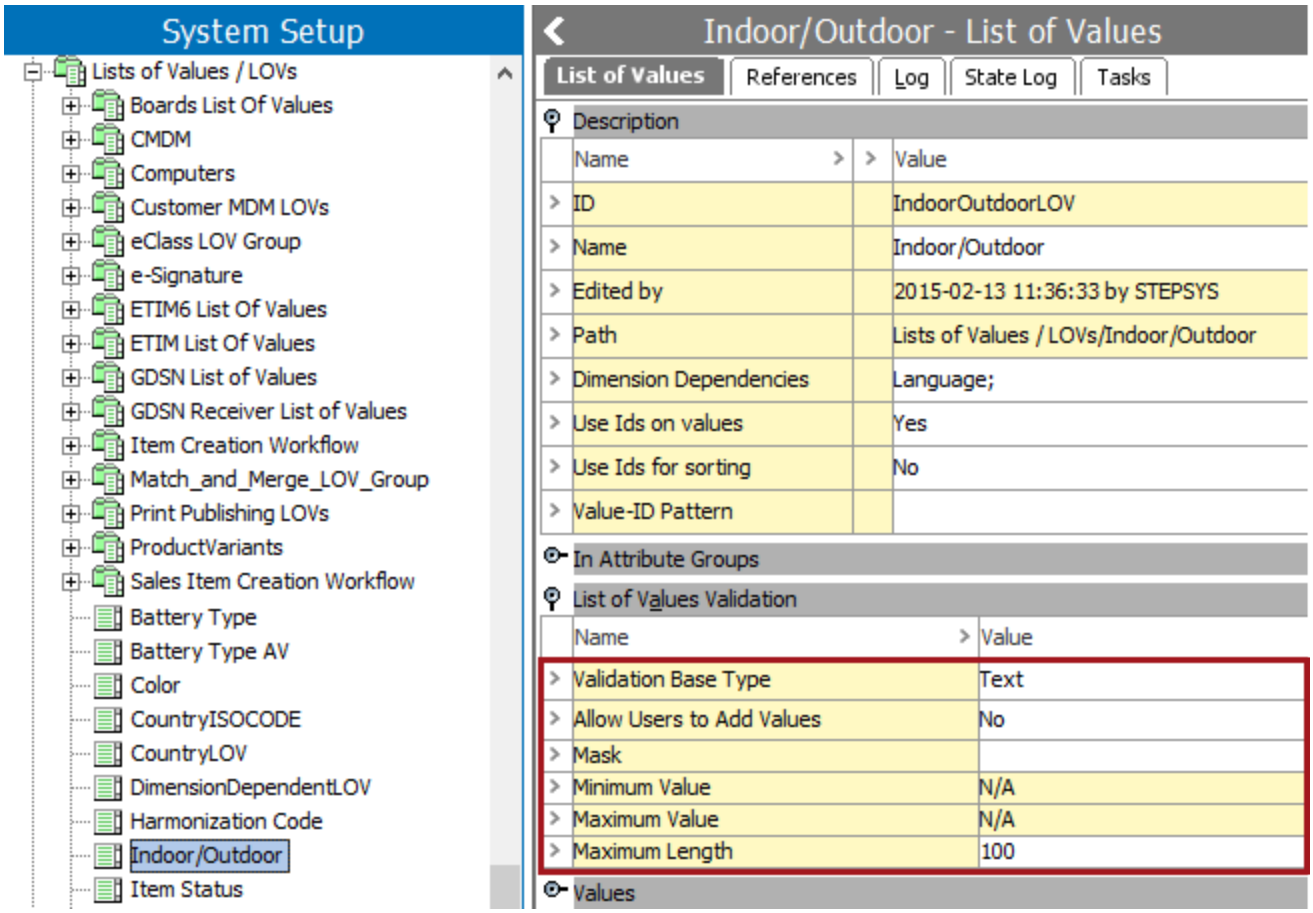
Editing Validation Rules

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.



For a List of Values / LOVs, edit directly on the List of Values tab.



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 or Numeric Text (exclude tags)
- Text (exclude tags) can be changed to Numeric Text or Numeric Text (exclude tags)
- Text and Text (exclude tags) 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 or Text (exclude tags)
- Number or Integer can be changed to Text or Text (exclude tags) only if no units are involved

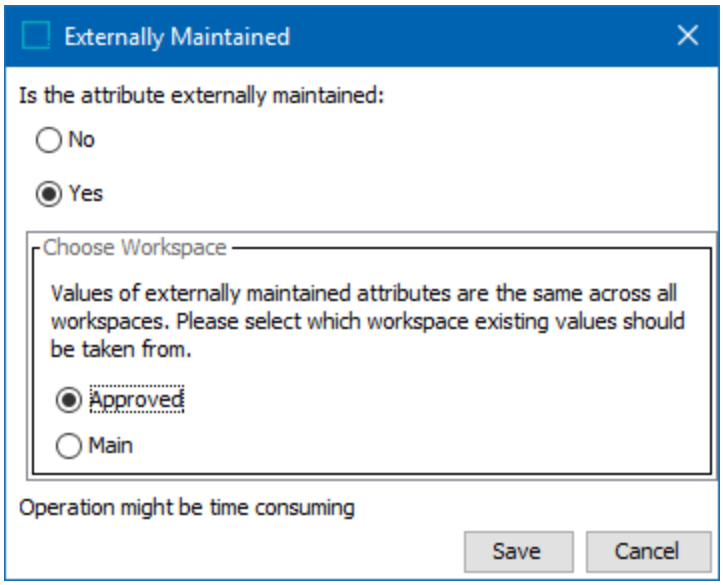
- Number or Integer can be changed to Numeric Text or Numeric Text (exclude tags)
- Fraction can be changed to Text or Text (exclude tags) only if no units are involved
- Fraction can be changed to Numeric Text or Numeric Text (exclude tags)
- Integer can be changed to Number

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 or Text (exclude tags) 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.



Custom Validation Error Messages

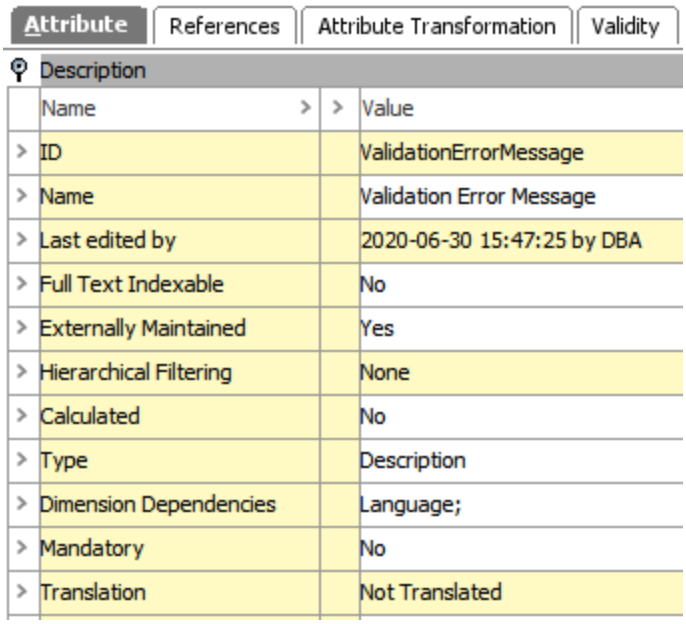
You can write your own error messaging for attribute validation errors. Creating clear error messaging provides a way for data stewards and various managers (e.g., brand, product, channel) to understand their errors and correct them immediately.

For example, writing a custom message for a Regular Expression (RegEx) validation that says 'Values must be a capital letter followed by a number, e.g., A7.,' is clear and concise compared to a system-generated message that says '[Blue] does not fit the regular expression "[A-Z]\d".'

Note: User-defined messages are optional. If a custom error message is not supplied, the existing system messages will still be shown. These error messages will display if there are errors when validating attributes in Web UI (shown above), workbench, Smartsheets, and upon import.

An attribute named 'Validation Error Message' exists on your system upon installation and is housed within your systems' Metadata attribute group in System Setup and is added to all attributes by default.

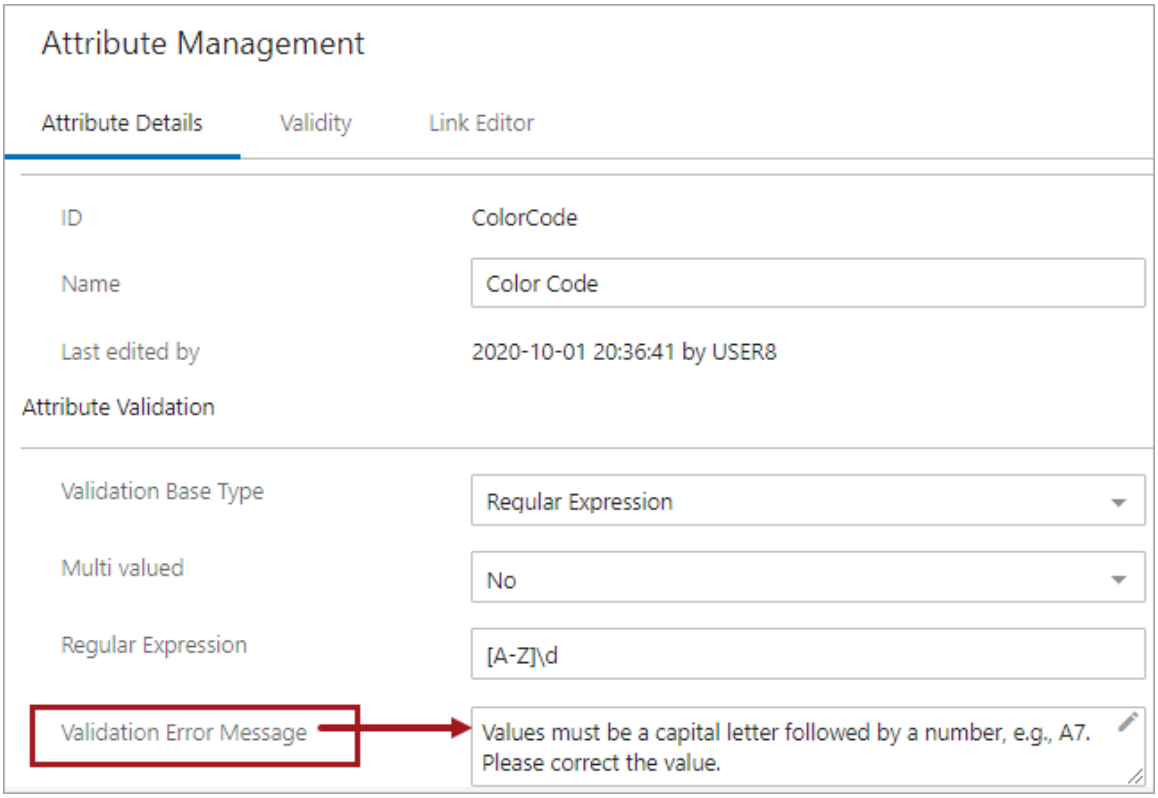
- If an attribute with the same ID (ValidationErrorMessage) already exists on a system before upgrading to 10.1 (and higher) versions, the existing attribute will not be changed nor will further configuration take place.
- If a Metadata attribute group already exists within another group, it will appear in that group even if within another higher-level group.
- If the Validation Error Message attribute is ever removed, it can be added back and set up as a metadata attribute again by following the directions that are found in the **Attribute Metadata on Attributes** topic.



Setting Up a Custom Validation Error Message

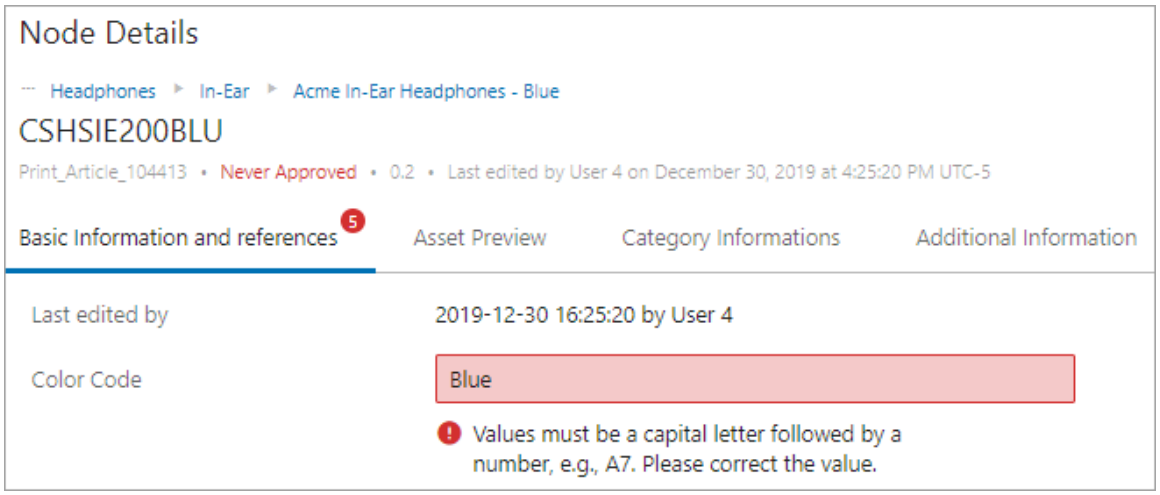
Define your custom error message within the metadata attribute field for the designated attribute.

- If using workbench, use the Attribute Editor.
- If using Web UI, define the message within the Attribute Management screen for the specific attribute (shown below). Save changes.



Using the Custom Validation Error Message

When you enter an attribute value and click out of the value field, if the attribute validation fails, then the custom error will display. Continuing with the example from above, see the error for an invalid Color Code below.



Considerations and Limitations

- As described above in the 'Setting Up the Validation Error Message' section, the validation errors are context specific and have nothing to do with locale.
- User-defined error messages are not present in the Map Data step of the Import Manager. Validation is done upon import and reported in the execution report.
- An execution report will always present the validation error messages from an XML-pointed import context and in cases where importing multi-context values.

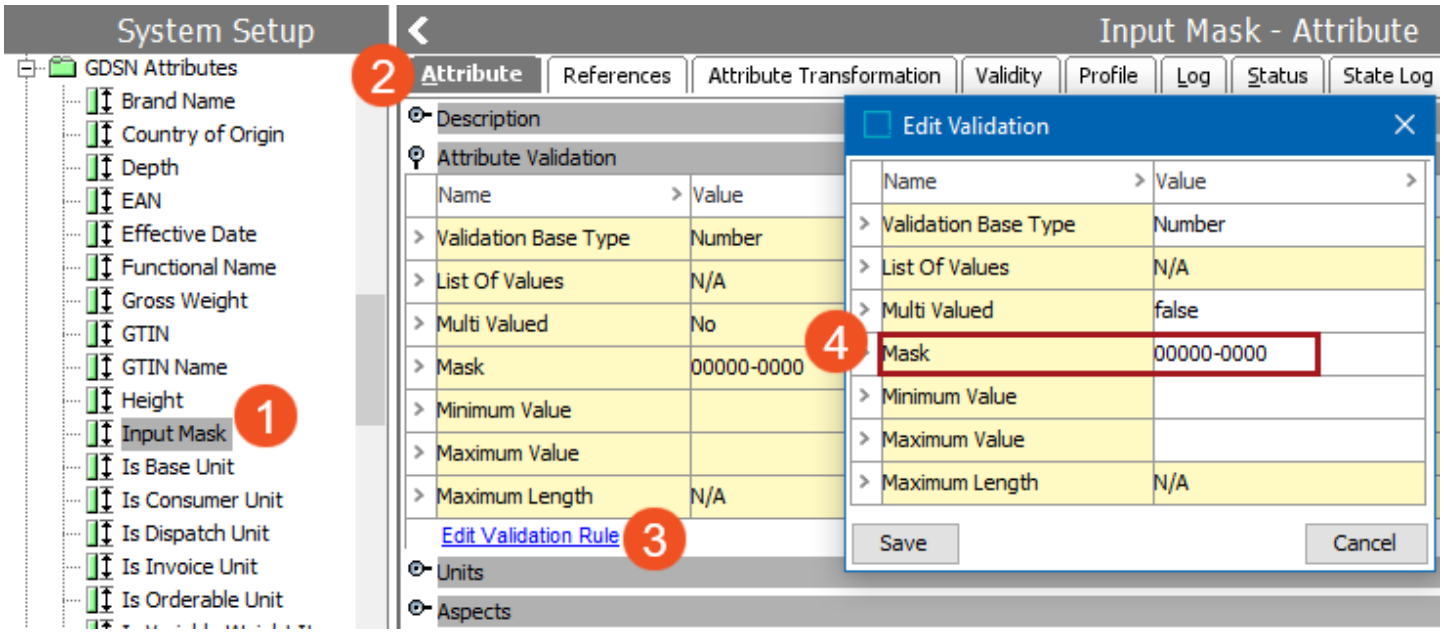
Input Masks

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.

Note that for attributes or LOVs with the Base Validity of ISO date and ISO Date and Time you should not enter any Input Mask at all. In fact you will encounter errors if you do enter a Mask and subsequently attempt to enter a Date or Date / Time value for an attribute.

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



Example Input Mask Use Cases

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

Example	Description
	The valid value for this attribute would have to be patterned as 12345-1234. Any other pattern, such as 123-1234 or 12345 would result in an error message appearing and the entry would not be accepted.
A LOV for an attribute contains values for a unique color code. Examples: BK (01) RD (02) Y (03) GR (04)	The requirement for this LOV input mask are: • The first character must contain an alpha character. • There may or may not be a second alpha character. • A space must exist between the letters and the numbers. • There must be two corresponding numerical characters and they must be enclosed in parenthesis. Thus the input mask would look something like this: L? (00)
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: 00000000000000000000 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 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

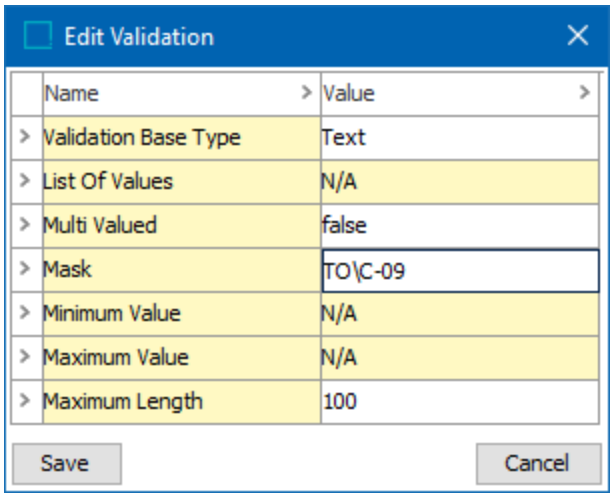
Mask	Description	Input Mask Example
?	The corresponding character must be an alphabetic letter up to the length of the mask. Empty values are OK at the end of the entry but no spaces otherwise.	Input Mask: ????? Valid values: ABCDE, AbCd, a Invalid values: AB12, A+, AB DE
a	The corresponding character 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%
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.

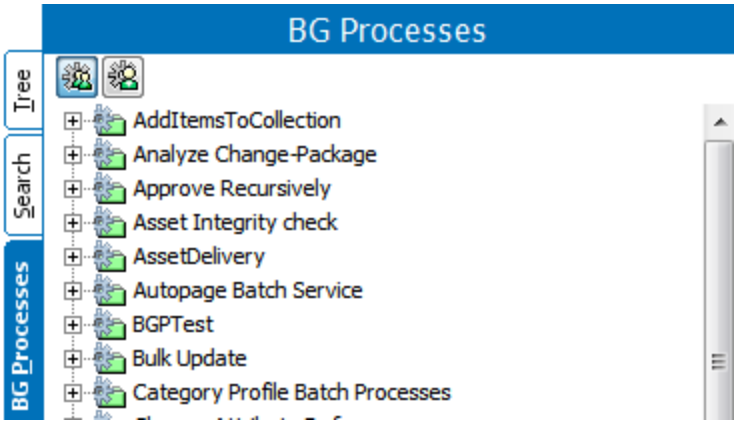
For example, in reference to the mask 'T-99', the 'T' and the '-' are always required to be entered but the '99' are masks for optional number per the definition in the table above. This means that the values T-1 and T-12 would be valid values but the values R-12 or T=1 would not be valid values.

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.



Background Processes and Queues

Background processes (BGPs) allow system actions to run in the ‘background’ while giving users the ability to continue working in the user interface. Most standard background process queues display on the 'BG Processes' tab in workbench and can be accessed there.



A background process queue is a way of prioritizing system processes that ensures high-level system performance. BGP queues can be named and configured to control which processes run on which server and how many may run at the same time on each server. Background processes that share a queue also share the number of simultaneous processes that can be run for a given application server.

Information related to these aspects of background process functionality can be found in this topic:

- Evaluating BGPs
- Parallel and Multi-Threading Properties
- Understanding config.properties Settings

Prerequisites

It is expected that anyone dealing with background process queues is familiar with the **BG Processes Tab** topic in the **Getting Started / User Guide** documentation.

Additional Information

For more information regarding Background Processes and Background Process Queues, see the following topics:

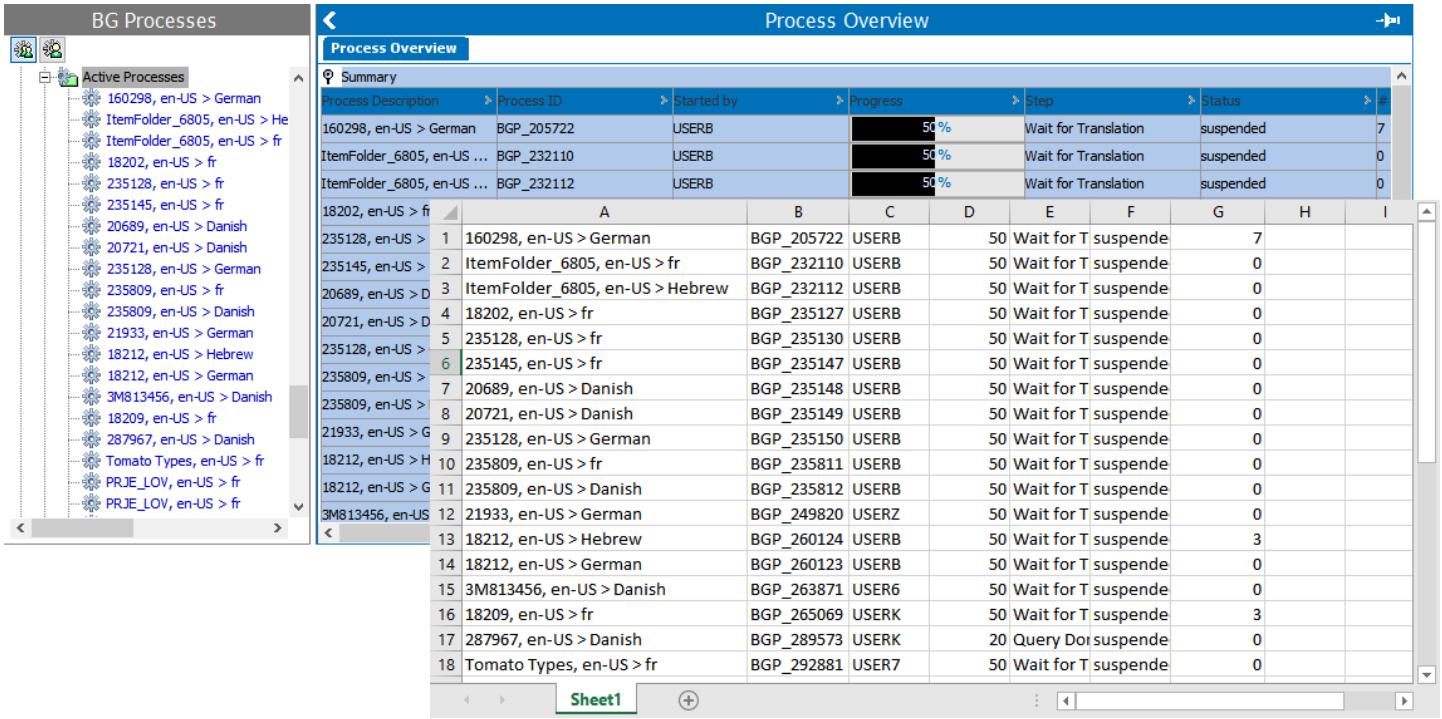
- BG Processes States
- Default Configuration for Background Process Queues
- Modifying Background Process Queue Configuration
- Managing Background Processes
- Scheduled Process Properties

- Deleting a Scheduled Background Process
- BG Processes Execution Management

Note: Deleting a user who is responsible for a BGP will result in errors. For more information, see the **Working with Users** topic in the **System Setup / Super User Guide** documentation.

Evaluating BGPs

Select a BGP folder (rather than an individual BGP) to display all BGPs currently in that state. Select the columns in a flipper, right-click and click Copy (or use Ctrl+C) to copy the data. Only the data visible on the screen is copied. For further evaluation, in Excel, paste the copied data using Ctrl+V and add the header row information.



Parallel and Multi-Threading Properties

In addition to integration endpoint transactional settings (see the **Integration Endpoint Transactional Settings** topic in the **Data Exchange** documentation), parallel and multi-threading properties can be used manage efficient processing.

Configuration properties allow admin users to manage system behavior for a wide range of functionality, for example, the management of background processes and web services.

Important: Ensure changes in the settings are applied with care and properly tested before promoting to production. Parallel setting and multi-threading can easily result in optimistic locking issues, which can have a negative impact on performance, rather than the desired performance gain.

The following config.properties file settings define how a background process is handled:

- **Size**: The number of BGPs that can run at the same time in the same queue. As an example, for data profiling setting the size = 0 on one server and setting size = 1 on another server forces all data profiling to be handled by a specific server. Recommended practice is to use one server to handle background processes and another to handle ad hoc user requests, which can give users improved system responsiveness.

  Use the **BackgroundProcess.Queue.*[name of queue]*.Size=*[number of allowed concurrent processes]*** property where the *[name of queue]* designation is replaced with the name of the queue for which the setting applies and the *[number of allowed concurrent processes]* designation determines how many concurrent background processes can run.

  A queue size of 1 means only one process can run at a time, while a queue size of 2 means two processes can run concurrently.

  For more information, see the **Default Configuration for Background Process Queues** section referenced below, the **Monitoring an IIEP via Background Process** topic, and the **Monitoring an OIEP via Background Process** topic in the **Data Exchange** documentation.

- **Parallel**: The following BGP types support the parallel option, which makes the same BGP use multiple CPUs on the server (multi-threading) in order to finish faster:
  - Data Profiling
  - Match and Merge Import
  - Match and Merge Web Service Endpoint
  - Exports
  - Bulk Updates
  - Imports
  - Policy evaluation
  - Matching and Linking
  - Experian Email Validation

  Use the **BackgroundProcess.Queue.*[name of queue]*.Parallel=*[number of CPUs]*** property where the *[name of queue]* designation is replaced with the name of the queue for which the setting applies and the *[number of CPUs]* designation determines how many CPUs are used.

  A queue parallel of 1 means only one CPU is used for the BGP, while a queue parallel of 2 means two CPUs are used for the BGP.

  Cores are the CPUs available to handle parallel processing. The CPU setting must not exceed the number of available cores. To view the number of cores on the application server, see the 'No of cores' parameter in the Activity tab Details section of the Admin Portal, as defined in the **Activity** topic in the **Administration Portal** documentation.

  Note: The parallel queue property's underlying implementation varies significantly based on the background process type. When starting 'X' number of jobs, you use at least 'X' times as many resources, which come from other processes and could impact performance. As such, thorough performance testing with production-level hardware and peak traffic conditions is highly recommended.

- **ProcessType**: This property is used to rename the default queue for a specific type of background process which allows the administrator to control the processes that run on each server.

  Use the **BackgroundProcess.ProcessType.*[process type ID]*.Queue=*[name of queue]*** property where the *[process type ID]* and the *[name of queue]* designations are replaced with the actual type ID and the actual name of the queue for which the setting applies.

For steps to change your configuration, see the **Modifying Background Process Queue Configuration** topic.

Single Application Server and Single Queue Example

This example includes inbound integration endpoints (IIEPs) but can also be applied to any feature that uses background processes.

Consider the following config.properties settings for a single server called appserver1:

```
BackgroundProcess.Queue.DataProfilerParallel.Parallel=4
BackgroundProcess.Queue.DataProfilerParallel.Size=4
BackgroundProcess.Queue.In.Parallel=8
BackgroundProcess.Queue.In.Size=1
```

Combined with these workbench parameter settings:

- The match and merge IIEP has 'Queue for endpoint processes=IN'
- The bulk update IIEP has 'Queue for endpoint processes=IN'

Result

A configuration with these settings results in the following activity on the single server:

- A 'Match and Merge IIEP' background process will not run simultaneously with a 'Bulk Update' background process. Both use the IN queue, which has size 1, limiting it to only one (1) background process at a time.
- A 'Match and Merge IIEP' background process will use up to 8 cores due to the parallel setting of 8 for the IN queue.

Two Application Servers Example

This example includes inbound integration endpoints (IIEPs) but can also be applied to any feature that uses background processes.

In addition to appserver1 described above, add a second application server (called appserver2) with the following config.properties settings:

```
BackgroundProcess.Queue.DataProfilerParallel.Size=0
BackgroundProcess.Queue.In.Size=0
```

Result

A configuration with these settings results in the following activity on the two servers:

- appserver1 runs up to four (4) data profiling processes in parallel. Data profiling process number five (5) remains in queue until one of the first four is complete.
- appserver2 never runs 'Match and Merge IIEP' background processes nor 'Bulk Update IIEP' background processes because this server has a size of 0 for IN queue, and both of these background processes use the IN queue.
- Since all BGPs run on appserver1, high BGP load does not adversely impact users because appserver2 is reserved for actions in the user interface.

Single Application Server and Two Queues Example

This example includes inbound integration endpoints (IIEPs) but can also be applied to any feature that uses background processes.

Consider the following config.properties settings for a single server called appserver3:

```
BackgroundProcess.Queue.InIIEP1.Size=10

BackgroundProcess.Queue.InIIEP2.Size=20
```

Combined with these workbench parameter settings:

- The match and merge IIEP has 'Queue for endpoint processes=InIIEP1' with 'Transactional settings = None'
- The bulk update IIEP has 'Queue for endpoint processes=InIIEP2' with 'Transactional settings = None'

Result

A configuration with these settings results in the following activity on the server:

- appserver3 runs up to 10 match and merge processes simultaneously. Match and merge process number 11 remains in queue until one of the first 10 is complete. These are the only processes that run on the InIIEP1 queue.
- appserver3 runs up to 20 bulk update background processes simultaneously. Bulk Update process number 21 remains in queue until one of the first 20 is complete. These are the only processes that run on the InIIEP2 queue.

Understanding config.properties Settings

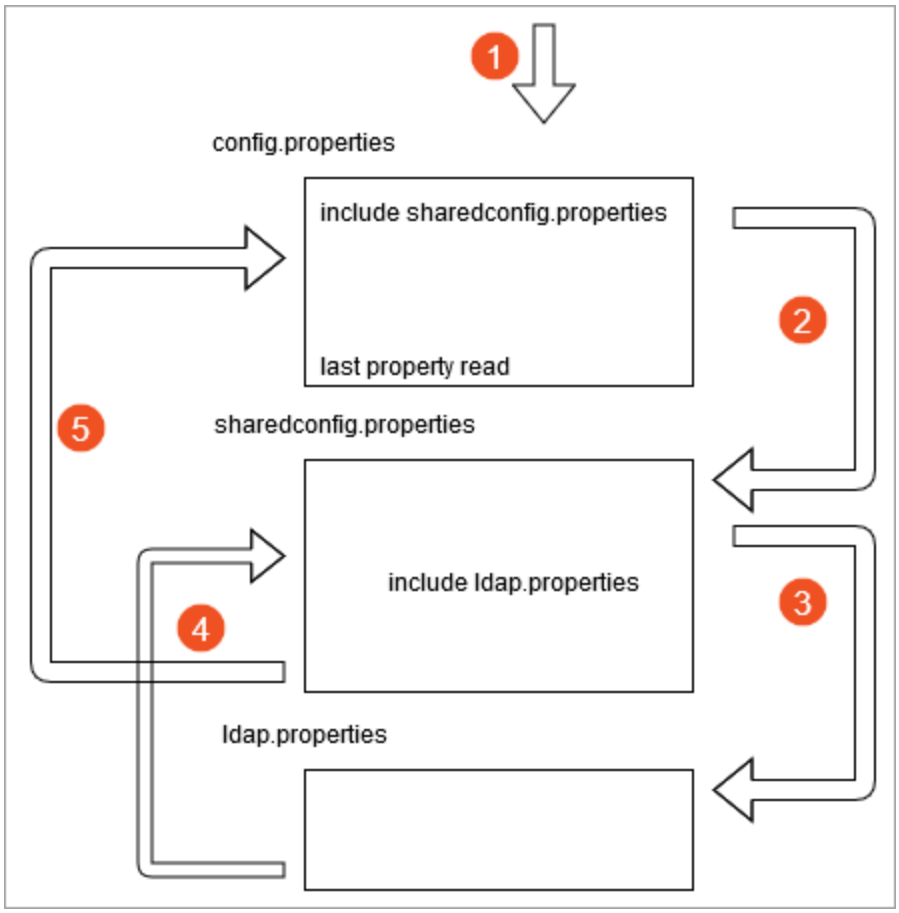
Configuration properties provide system administrators with options for controlling system behavior, logging, processing, storage of information on the application server(s), and many other things. Configuration properties are specified in the config.properties or sharedconfig.properties files on the application server(s). There is one config.properties file per application server so a clustered environment may have multiple config.properties files, with each containing settings local to a particular application server. Conversely, the sharedconfig.properties file is global so there will be only one file per setup, with all settings being applicable to all application servers. Most properties are set in the sharedconfig.properties file, with the config.properties file being used only for those properties that expressly require it.

The config.properties file is read first, line-by-line, until the first 'include' statement—typically the sharedconfig.properties file as shown below.

```
#=============================================================================#
# Include other config files
#=============================================================================#
@include /workarea/sharedconfig.properties

#=============================================================================#
# StandAlone Settings (Application Server)
# NOTE: Only used if ApplicationServer.Type = standalone
#=============================================================================#
ApplicationServer.Type = standalone
```

When an 'include' statement is found, the included file is read line-by-line (and any 'include' statements in those files are also read) before reading is continued in the config.properties file. A property can exist more than once in the properties files and the last value read is the only active value. This allows properties in the config.properties that come after an 'include' statement to overwrite the same properties in the sharedconfig.properties file.



For more information on config.properties settings, see the **Configuration** topic in the **Administration Portal** documentation.

BG Processes Execution Management

Users have a simplified background process (BGP) execution mechanism to prioritize the order in which background processes are handled.

Waiting BGPs can be prioritized for execution based on creation time and a priority. Priority can be set for event processors, OIEPs, and IIEPs (for the latter two, the configurable priority is for the 'worker' BGPs started by the endpoints). The application server capacity determines how many BGPs can run in parallel.

The background process execution mechanism selects BGPs to start across all BGPs waiting to be executed. By default, this mechanism is not enabled. This can be enabled via the configuration property BackgroundProcess.Scheduler.PluginID=OneQueueBGPScheduler.

BackgroundProcess.Scheduler.PluginID=OneQueueBGPScheduler

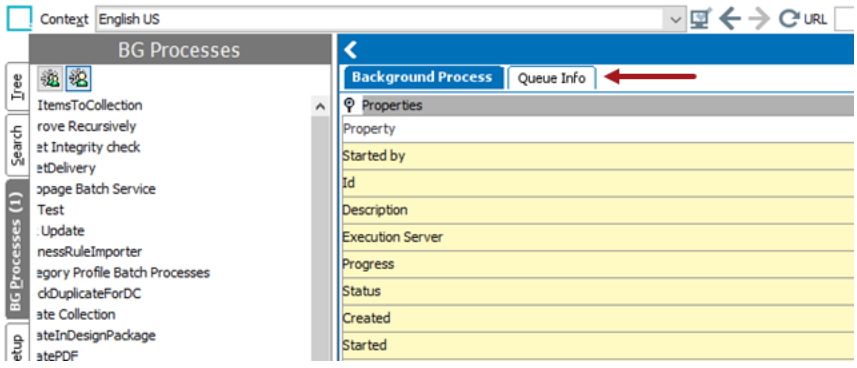
- Set in: /workarea/sharedconfig.properties
- Overrides default: "MultipleQueuesBGPScheduler"

Selects the BGP scheduler plugin to use. Will default to use old default scheduler - using queues as usual. Can be changed to enable new one-queue task scheduler. This scheduler has no concept of queues (it has only one). And BGPs are sorted by priority and time waiting on queue. BGP will always be started in this order. A BGP will only be started if load on server is determined to hold this BGP.

Important: Before applying to a production setting, the background process execution mechanism should be applied and thoroughly tested on lower-level environments.

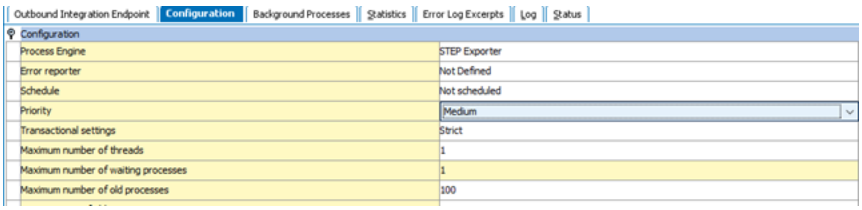
When the mechanism is enabled:

- Existing configurations that assign BGPs to queues and set the degree of parallelism for the queues is ignored.
- The 'Queue Info' BGP editor tab is no longer displayed.



- You cannot specify queues for IEPs and event processors.
- You can specify a priority for the event processor BGP and BGPs started by IEPs from these configuration

types.

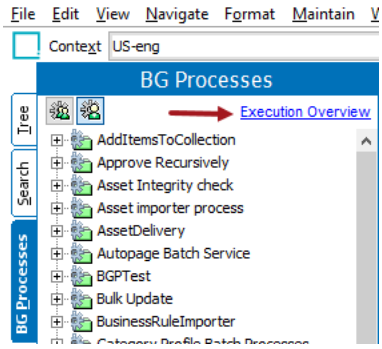


Each application server has a fixed number of 'BGP Workers' available for processing BGPs. The number of available BGP Workers is set automatically, depending on the application server hardware configuration. Individual application servers polls the list of waiting BGPs with regular intervals and starts as many processes as it has capacity for (via available BGP Workers). The waiting BGPs are prioritized for execution first based on priority, and the on creation time ('Start at' for scheduled event processor BGPs).

Poller processes for IIEP, OIEPs, and BGP scheduler processes (BGP type 'Schedule') bypass the prioritization and are always allowed to run.

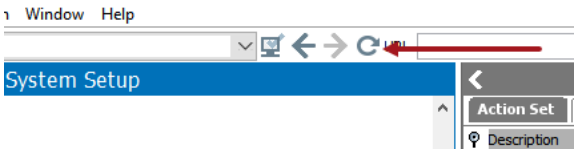
Background Process Execution Overview

Admin users with the 'View Background Processes of Other Users' setup action can monitor the execution of background processes via the Background Process Execution Overview link available on the BG Processes tab in workbench.



The Background Process Execution Overview is divided into these sections: Servers, Running Background Processes, and Waiting Background Processes.

Note: The overview is not automatically updated. Use the general workbench Force Reload option or click the Execution Overview link again to update the data.



Servers

The Servers section provides an overview of the application servers in the STEP cluster with details about the number of server CPUs, the total number of BGP Workers ('Max Workers'), the number of active BGP Workers, and the CPU load. Application server details are not refreshed every time the execution overview is refreshed, therefore, there is a timestamp in the Latest Snapshot column for when the data was collected.

Running Background Processes

The Running Background Processes section shows the BGPs currently being executed by STEP. The following is a list of the columns associated with Running Background Processes:

- ID – The ID of the BGP (link).
- Description – The BGP Description (link).
- Started By – ID of the STEP user who started the BGP.
- Progress – BGP Progress.
- Created – Time that the BGP was created. For scheduled event processor BGPs, this shows the 'Start at' time.
- Started – Time that the BGP was started.
- Workers – The number of BGP Workers used for executing the BGP. This can be more than one for BGPs using the parallel execution / multi-threading framework. See the **Parallel Processes** section below for more details.
- Running on – The application server executing the BGP.
- Type – The background process type.
- Owner – The IIEP, OIEP, event processor, or setup entity that created the BGP.
- Priority – The priority of the BGP.

Waiting Background Processes

The Waiting Background Processes section shows the BGPs waiting to be executed. The columns shown are:

- ID – The ID of the BGP (link).
- Description – The BGP Description (link).
- Started By – ID of the STEP user who started the BGP.
- Created – Time that the BGP was created. For scheduled event processor BGPs, this shows the 'Start at' time.
- Priority – The priority of the BGP.

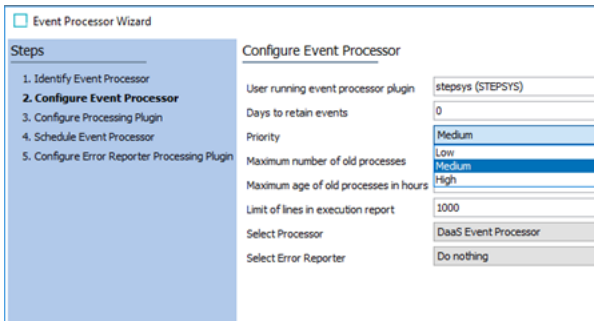
- Type – The background process type.
- Owner – The IIEP, OIEP, event processor, or setup entity that created the BGP.
- Depends Upon – IDs of BGPs potentially blocking the execution when the BGP depends on other BGPs to complete before they can start.

Priority

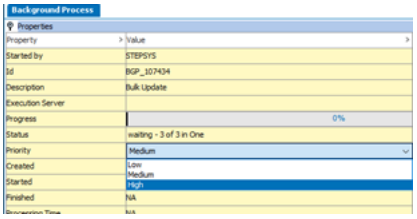
Waiting BGPs are prioritized for execution based on the priority of the BGP and the created time.

Choose a method to set BGP priorities:

1. When configuring event processors, IIEPs, and OIEP, users with maintain permissions can set a Low / Medium / High priority. The default value is 'Medium.'
   - For IIEPs and OIEPs, this is the priority of BGPs Workers created by the endpoints.
   - For event processors, this is the priority of the BGP associated with the event processor.



2. Users with the 'Manage Background Process Execution' setup action can set the priority of waiting BGPs directly in the workbench Background Process editor. This allows admin users to make ad hoc changes to how waiting processes are prioritized. By default, BGPs other than event processors and IEPs have priority level 'Medium.'



Important: It is recommended to initially leave all event processor and IEP priority settings at their default 'Medium' value. This means that BGPs created / managed by these configuration types have the same priority as user-initiated BGPs (and BGPs created / managed by other event processors / IEPs). Changing the priority to 'Low or 'High' should only be done if it is verified that a specific integration / background process should always have lower / higher priority than other BGPs being executed on the system.

Parallel Processes

With the new mechanism enabled, the degree of parallelism / number of threads per queue set via the configuration properties for processes supporting parallel execution (multi-threading) are no longer honored. Instead, the number of threads / BGP Workers to use for a BGP are automatically determined based on the application server hardware.

Multi-threaded BGPs can initially be inactive, even though they have been claimed by a server and appear in the 'Running Background Processes' section (with a status of 'running'). This occurs when the server does not have the desired number of BGP workers available. In this case, no new BGPs on the server start until the desired number of workers is available and the multi-threaded process can start.

For OIEPs only, the user can specify the maximum number of threads / workers to use for processes started by the endpoint. This option limits the number of threads / workers only and cannot be used to increase the amount beyond what the server hardware configuration allows.

BG Processes Failsafe

STEP has a mechanism to prevent repeated restarts of background processes (BGPs) that were running on an application server that crashed. This mechanism prevents continuous loops of server crashes caused by BGPs.

If a BGP is believed to be a potential cause of an application server crash, STEP will put the background process into a 'failed' state, where it will not be restarted automatically. Additionally, the BGP is put into a 'quarantined' state which requires user intervention.

Note: When the system administrator is shutting down an application server, it is important to do so properly with the official tools, as simply killing the application server JVM for STEP or cutting power will be interpreted as a crash and may cause this mechanism to be triggered. Shutting down a STEP application server with the official tools will not have this effect.

An analysis of why the process was quarantined will need to be done in order to determine what action to take. Not all background processes being quarantined are the cause of a crash. In some cases, it is a matter of being in the wrong place at the wrong time, which is unfortunately difficult to detect automatically.

Sometimes, the crash is not caused by the specific background process that was quarantined. If the issue can be found and fixed, the quarantined background process can be restarted. A background process can be restarted from the workbench. Alternatively, if it is determined that the quarantined background process should not be started again, it can be deleted.

Integration Endpoints

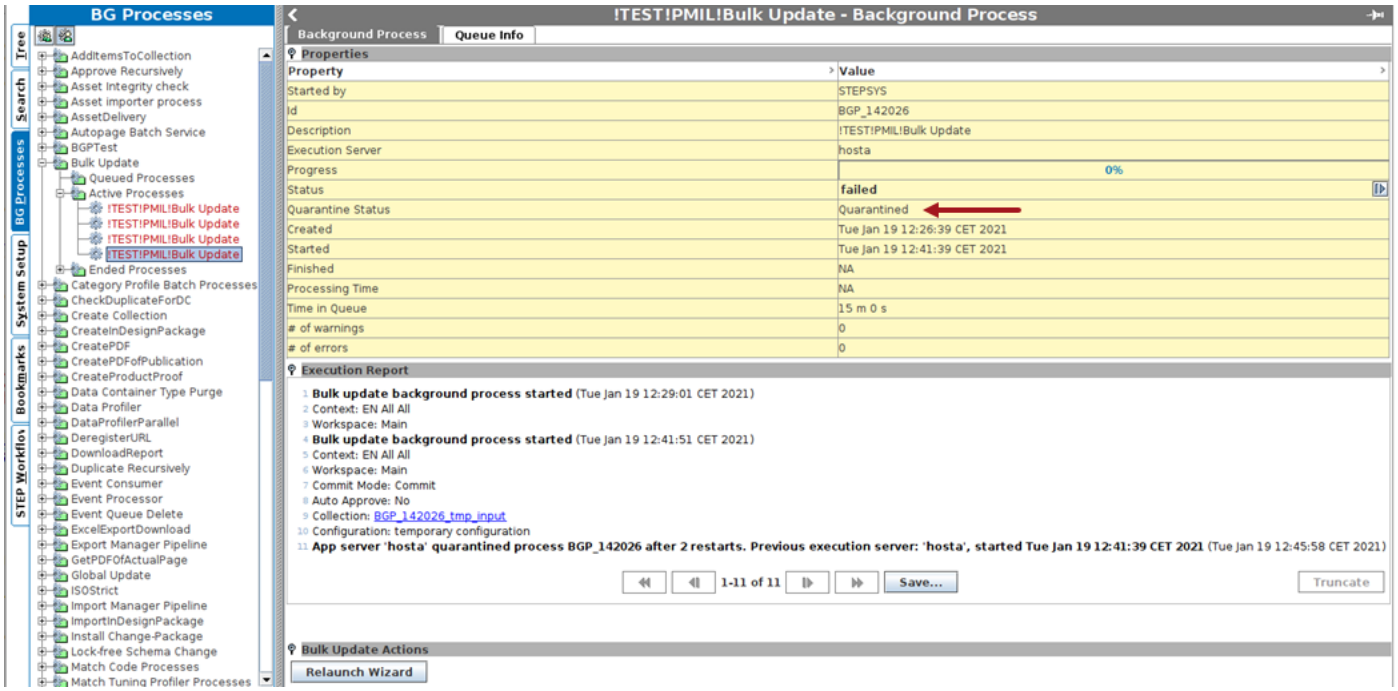
Integration endpoints will halt if any of their associated background processes are quarantined. This is done to avoid starting background processes, which could cause the same problem to repeat. The execution reports of both the integration endpoint and the background process will indicate the connection between the two processes. The integration endpoint will need to be manually restarted in the workbench. It is strongly recommended to first deal with any other quarantined background processes of the integration endpoint. Resuming an integration endpoint will not resume any associated quarantined background processes.

Important: It is vital that integration endpoints are monitored so they can be restarted after such an incident. Sensors for integration endpoints are already available for this purpose.

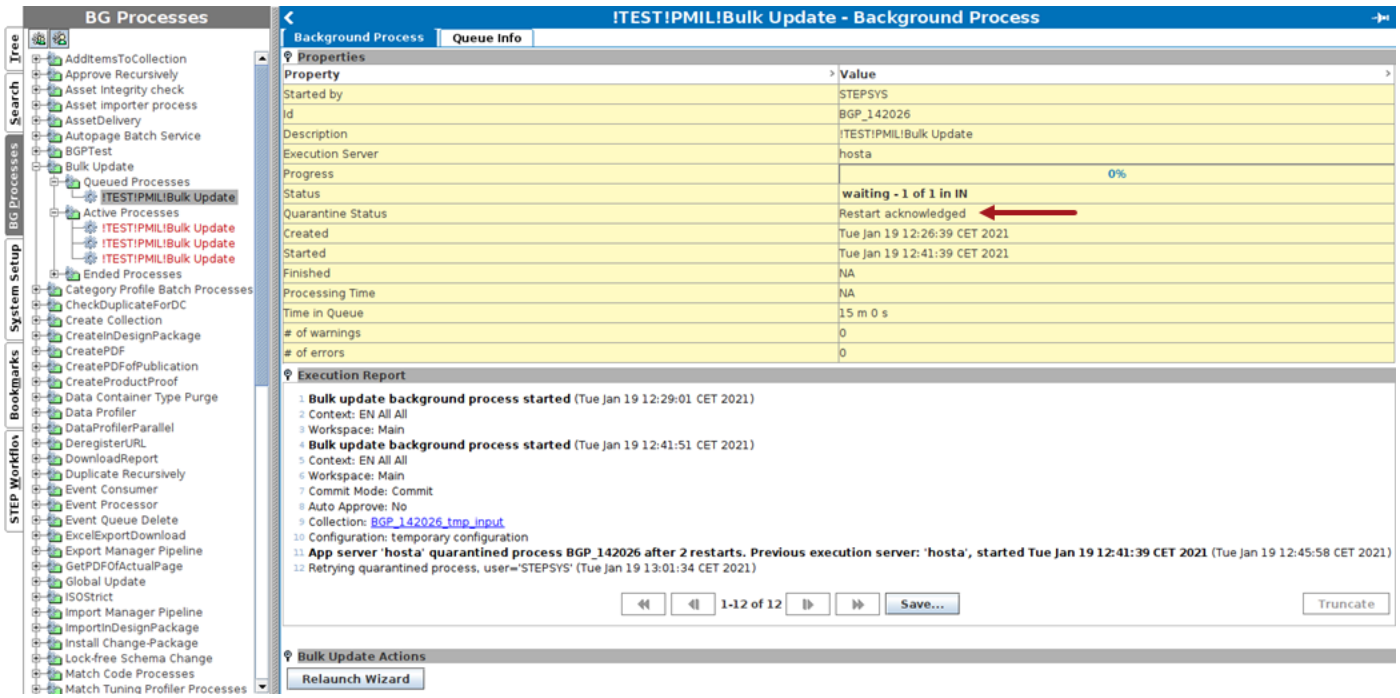
Possible Values for Quarantined State

The Quarantine Status property will only be displayed if a background process is or has ever been in quarantine. Otherwise, the property line will not show on the BGP tab. If a background process has been quarantined, there are three possible values that will be displayed:

Quarantined: When the background process is currently in quarantine. In normal cases, the background process is in a 'failed' state when this happens.



Restart acknowledged: A quarantined background process has restarted, but has not yet being executed, e.g. because it just happened or it is waiting on its execution queue. Normally, the process is in 'waiting' state.



Previously Quarantined: A background process was in quarantine, but is no longer quarantined. The background process can now be in any of the possible BGP states.

Searching for Quarantined Background Processes

Quarantined background processes can be found using drill down search in the workbench:

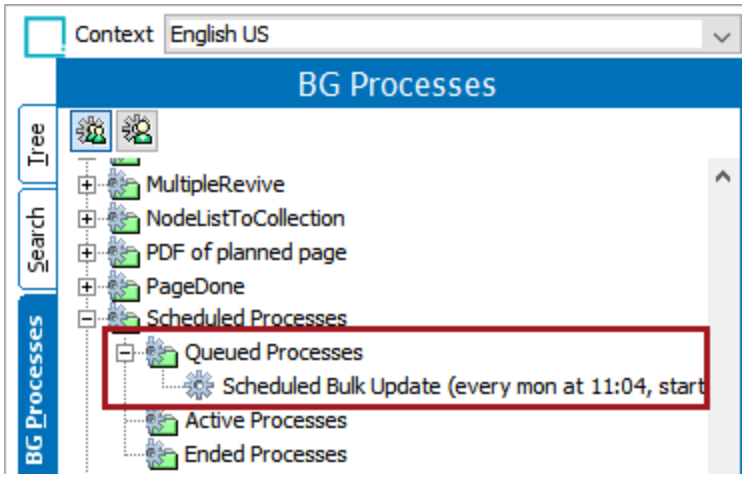
1. Select the **Search** tab on the left.
2. In the criterion drop down list, change the criterion to **Background Process Search**.
3. For the **Quarantined parameter**, select Quarantined.
4. Press the **Search** button to start the search.

BG Processes States

The following states are possible for a background process that has started.

Queued Processes

This shows the list of processes which are in queue but have not yet been started. When the queued process is started it moves to the active process node.



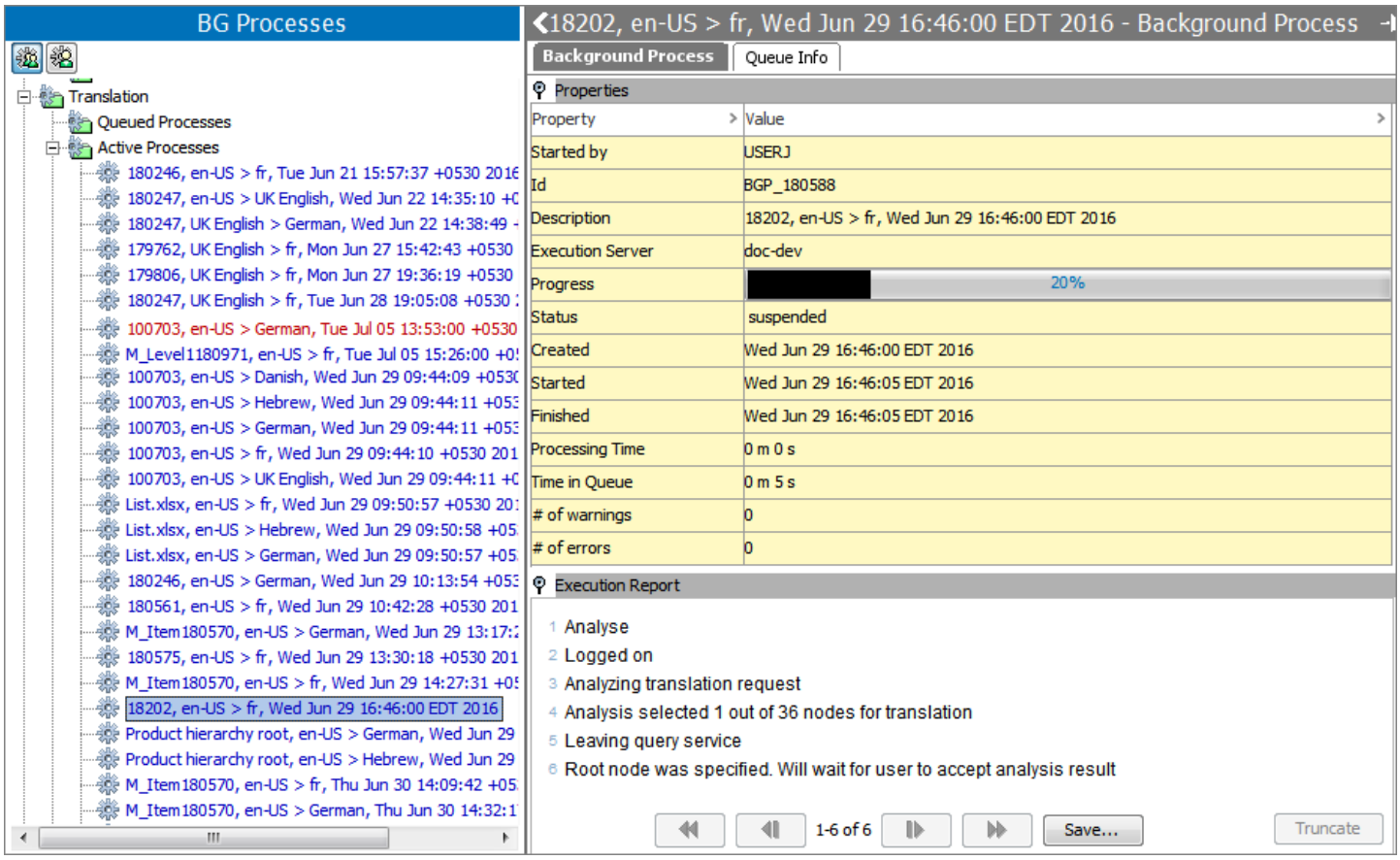
Active Processes

This shows the list of processes which are currently being executed.

Active processes that have failed remain in the active process node but are highlighted with red colored text.

- For failed IIEP processes, see the **Handling Failed IIEP Background Processes** topic in the **Data Exchange** documentation.
- For failed OIEP processes, see the **Handling Failed OIEP Background Processes** topic in the **Data Exchange** documentation.

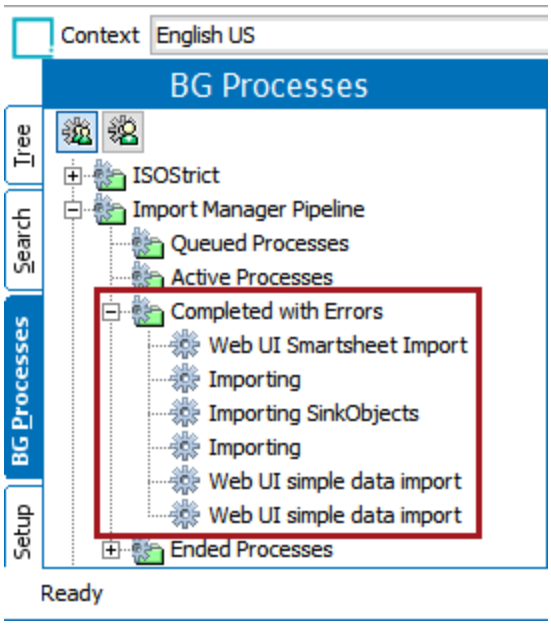
To abort an active BGP, select it, open the Maintain menu, and click Delete. The process is moved to the Ended Processes folder.



Completed with Errors Processes

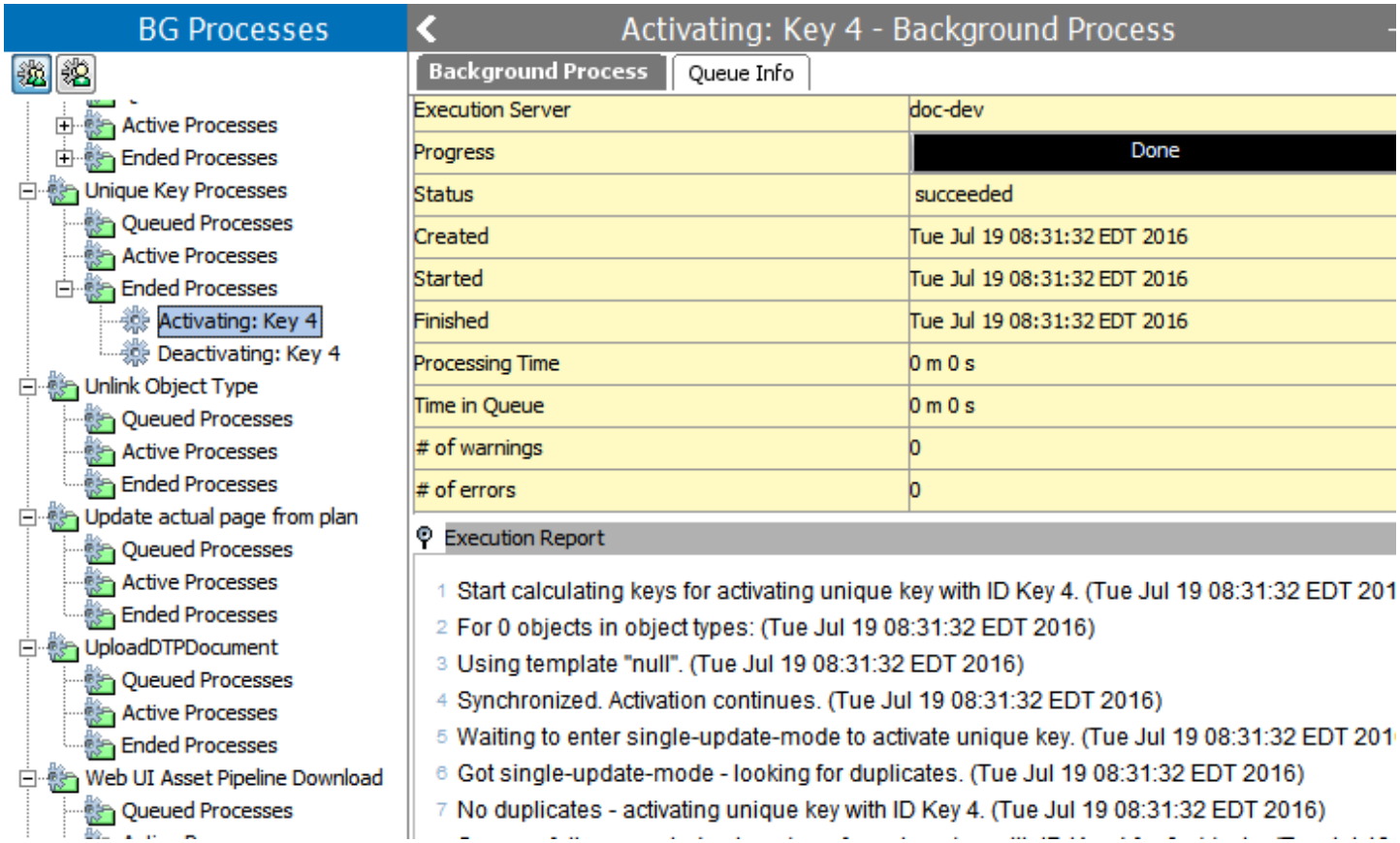
Processes that partially completed but also had errors. The minor errors are displayed with the details, but the events that can be processed will eventually complete.

Note: 'Completed with Errors' option is only available for a limited number of processes including Asset importer process, Import Manager pipeline, Match Code processes, Smartsheet Import, and StateflowDeadline.



Ended Processes

This displays a full list of all the processes which have been processed without any errors.



Default Configuration for Background Process Queues

In the default configuration, the following queues are used:

- Import processes have their own queue named IN.
- Export and web publisher processes share a queue named OUT.
- Scheduled processes have their own queue named SCHEDULE.
- All other background processes share a queue named MISC.

The IN, OUT, and SCHEDULE queues are set up to execute one process at a time, while the MISC queue is set up to execute two processes simultaneously. For more information on these settings, see the **Background Process Queue Multi-Threading Properties** section of the **Background Processes and Queues** topic.

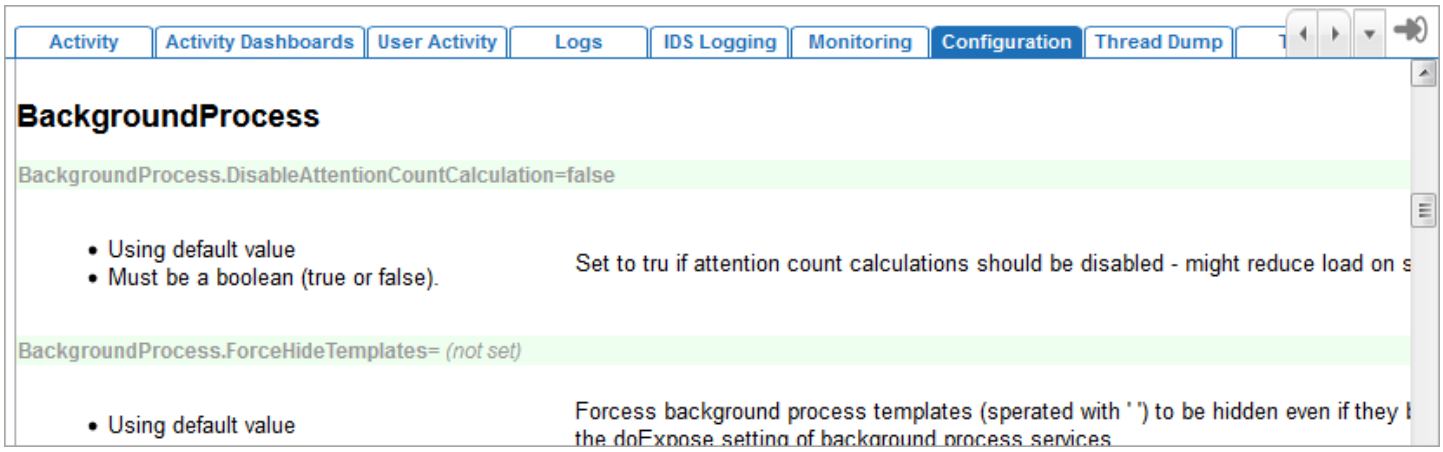
The following background process properties define the type of data being processed:

- BackgroundProcess.ProcessType.**Exporter**.Queue=OUT
- BackgroundProcess.ProcessType.**Importer**.Queue=IN
- BackgroundProcess.ProcessType.**Schedule**.Queue=SCHEDULE
- BackgroundProcess.ProcessType.**WebPublisher**.Queue=OUT

The following background process properties define the default queue sizes:

- BackgroundProcess.Queue.IN.Size=1
- BackgroundProcess.Queue.MISC.Size=2
- BackgroundProcess.Queue.OUT.Size=1
- BackgroundProcess.Queue.SCHEDULE.Size=1

All of the background process settings can be viewed in the STEP System Administration portal, on the Configuration tab under the Background Process section.



For more information, see the **Configuration** topic in the **Administration Portal** documentation.

Modifying Background Process Queue Configuration

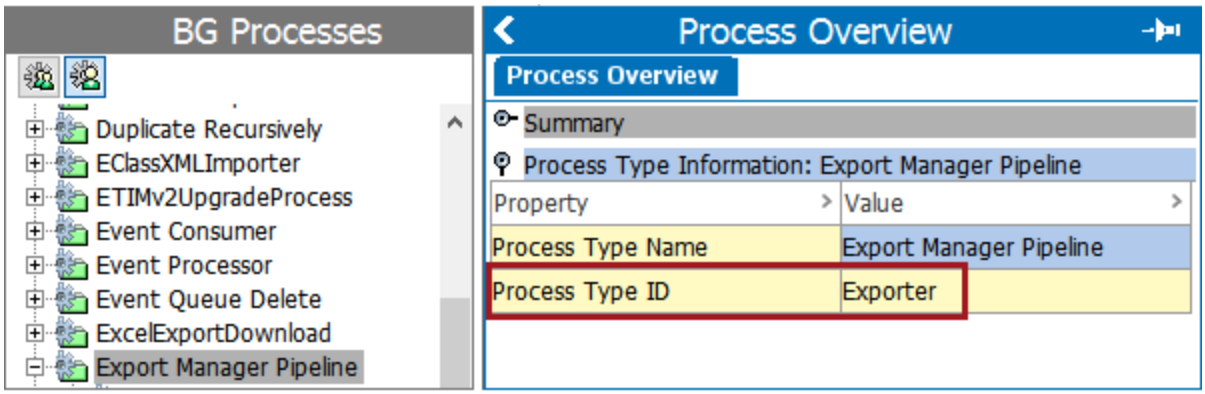
In the steps below, the default configuration is changed to allow for the export and web publisher processes to be run on separate queues.

Note: If you run a clustered setup and you make the following changes in the shared file, the settings regarding the size will be applicable for each application server. So, if you edit the shared configuration file as described in the example 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.

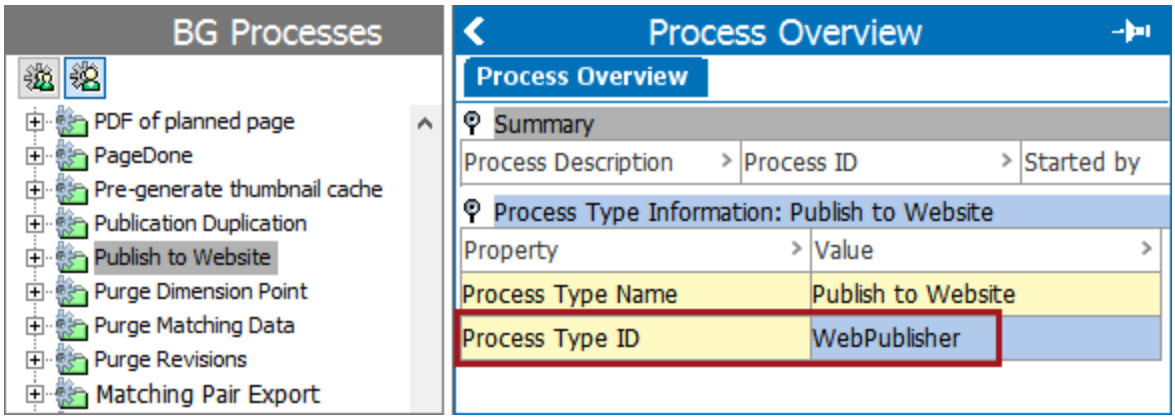
Update Queue Configuration

Use the following steps to modify the export and web publisher processes to run two processes simultaneously.

1. In the workbench, click the **BG Processes** tab.
2. Click **Export Manager Pipeline** node to display the Process Overview.



3. Expand the **Process Type Information: Export Manager Pipeline** flipper.
4. Right-click the value of the **Process Type ID**, copy and paste the value in a temporary text file. In this example 'Exporter' is copied and pasted. This value will be used to set the configuration property below.
5. Click the **BG Processes** tab then click **Publish to Website**. A Process Overview dialog displays.



6. Expand the **Process Type Information: Publish to Website** flipper.
7. Right-click the value of the **Process Type ID**, copy and paste the value to a temporary text file. In this example 'WebPublisher' is copied and pasted. This value will be used to set the configuration property below.
8. Open the sharedconfig.properties file on the STEP application server.
9. Add the case-sensitive **BackgroundProcess.ProcessType.*[Process Type ID]*.Queue=*[Queue]*** property to create the new queues. Replace the text in square brackets with the required data and use the following required format of the property:

   ```
   BackgroundProcess.ProcessType.[Process Type ID].Queue=[Queue]
   ```

   Continuing with the above example, the entry in the properties file would include the values pasted to the temporary file above:

   ```
   BackgroundProcess.ProcessType.Exporter.Queue=EXPO
   ```

   ```
   BackgroundProcess.ProcessType.WebPublisher.Queue=WEBP
   ```

   In this example, [Process Type ID] is replaced with 'Exporter' and 'WebPublisher' to identify the new types of queues.

   [Queue] is replaced with the new Export process queue named EXPO, and the new Web Publisher queue named WEBP.

10. Add the case-sensitive **BackgroundProcess.Queue.*[Name]*.Size** property to set the size on the new queues. Replace the text in square brackets with the required data and use the following required format of the property:

    ```
    BackgroundProcess.Queue.[Name].Size=[Size]
    ```

    Continuing with the above example, the entry in the properties file would be:

    ```
    BackgroundProcess.Queue.EXPO.Size=2
    ```

    ```
    BackgroundProcess.Queue.WEBP.Size=2
    ```

    In this example, the new queue EXPO and the new queue WEBP can each run two concurrent processes.

11. Restart the application server to apply the changes to the properties file.

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 state:

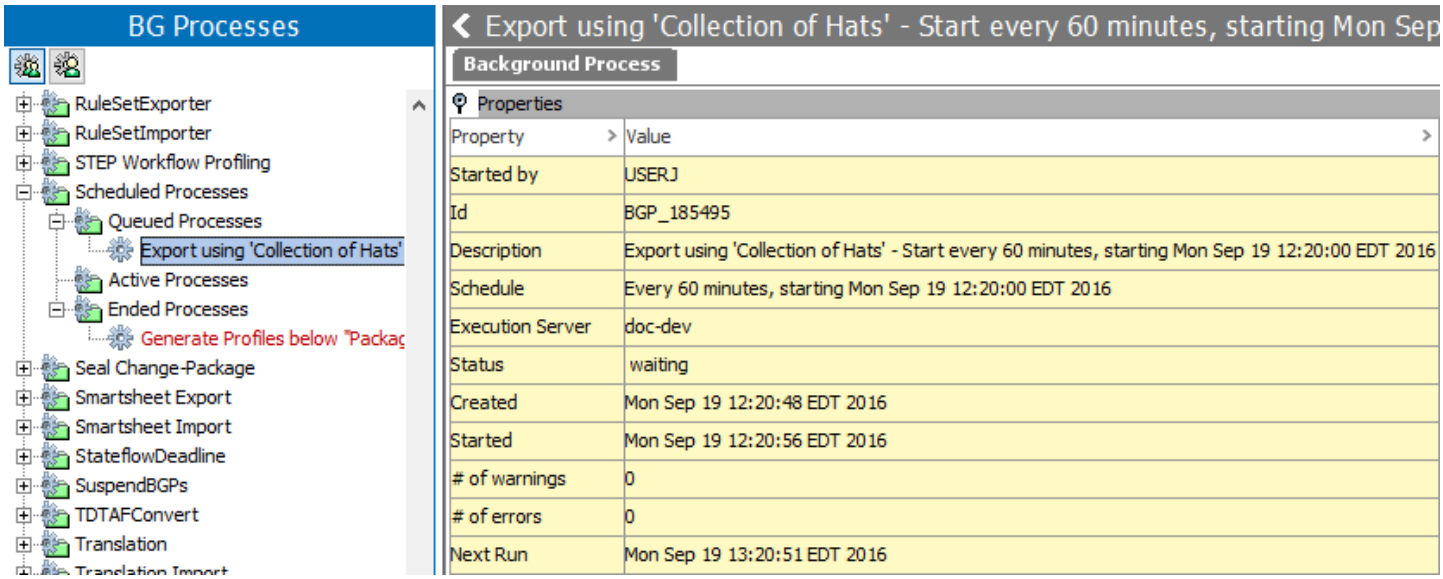
- **Queued Processes** are waiting for the next scheduled run.
- **Active Processes** are currently running.
- **Completed with Errors Processes** are available for a limited number of processes and at least some of the process succeeded.
- **Ended Processes** have been aborted or the schedule is completed.

For details on the states of a background process, see the **BG Processes States** topic.

For details on scheduling a background process, see the relevant section of online help, such as the **Scheduling a Data Export** topic in the **Data Exchange** documentation or the **Scheduling Bulk Updates for Collections** topic in the **System Setup / Super User Guide** documentation.

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 the 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.

Important: Consider the time zone of the application server compared to that of the workbench (the client) where the schedule is created or viewed. When scheduling a job, the local time zone is displayed in the workbench, but the time zone of the server is used to run the background process. Although displayed, the time zone of the client is not included in the instruction to the server to run the job. This can cause confusion about when the job will run since the scheduled time is not automatically converted to accommodate potential differences in time zones.



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.

Creating a Scheduled Background Process

The steps to create a schedule for a BGP vary based on the process. See the following topics for details:

- Generating, Updating, and Scheduling a Data Profile
- Scheduling Bulk Updates for Collections
- Scheduling a Data Export
- Scheduling a Collection of Objects for Translation
- Scheduling a Data Translation

Alternatively, inbound and outbound integration endpoints and event processors can be scheduled to run background processes that are not available on the BG Processes tab. Instead, in workbench on the System Setup tab, view these related BGPs on the Background Processes tab of the integration or event processor editor. Some endpoints, such as gateway and web service, do not run on a schedule.

- IIEP - Schedule Endpoint
- OIEP - Configuration Flipper
- EPW - Schedule Event Processor

Deleting a Scheduled Background Process

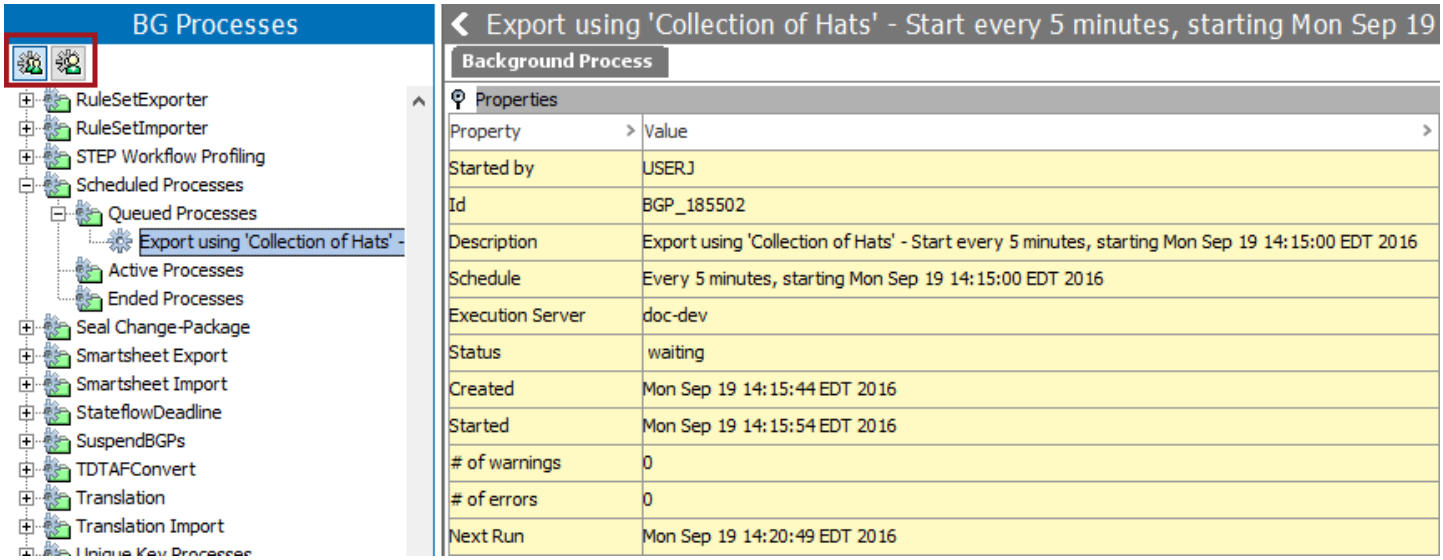
A scheduled process continues to use the configuration it was started on. Therefore, if you change the parameters of the configuration, you should delete the relevant scheduled processes.

Once a process is shown in the Queued Processes node, the schedule details cannot be changed. Instead, it must be deleted and rescheduled as needed.

Delete a Scheduled Process

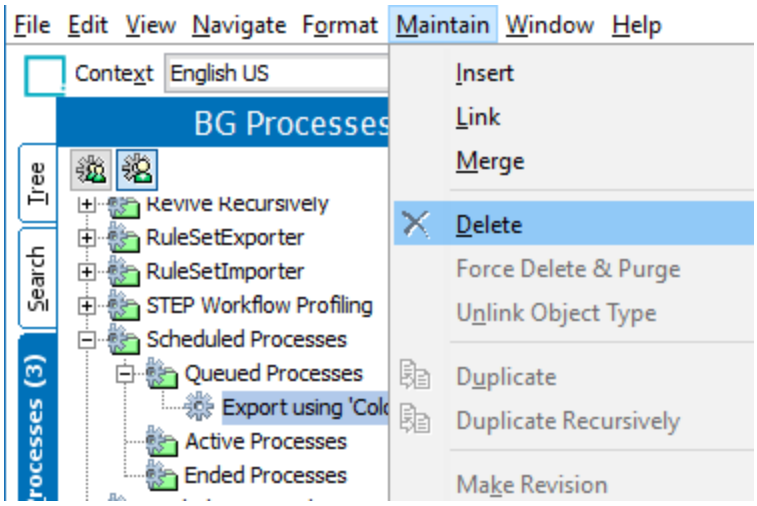
The images below show an export process, but the steps are the same for all scheduled processes.

1. On the BG Processes tab, expand Scheduled Processes node and open **Queued Processes**. If needed, use the filter buttons at the top of the BG Processes tab to display processes started by other users.



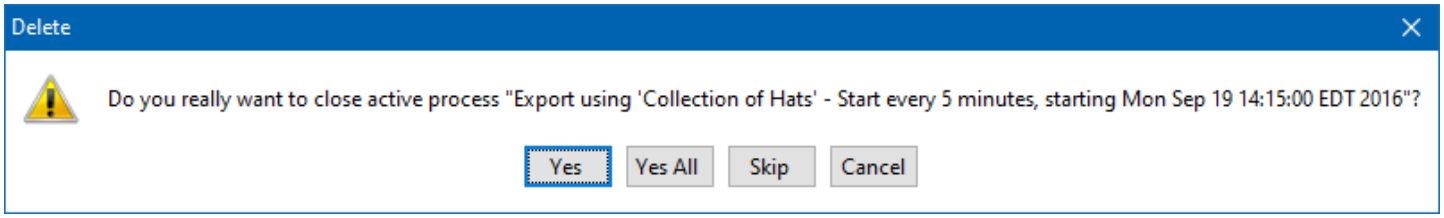
   For details on the states of a background process, see the **BG Processes States** topic in the **System Setup / Super User Guide** documentation.

2. Select the one or more processes to delete, and from the **Maintain** menu, click **Delete**.



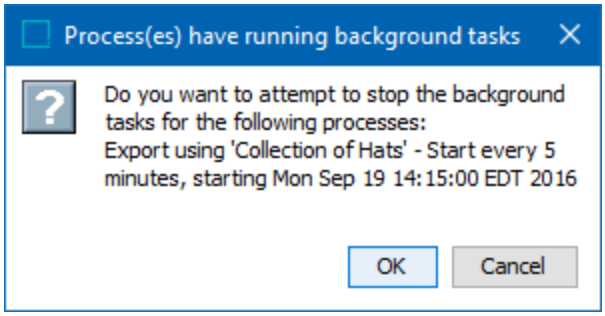
Important: This step moves the scheduled process to an aborted status and displays it in red text as an Ended Process. The details of the process continue to display, however, an aborted scheduled process cannot be started again. To run it again, use the details to recreate the scheduled process.

3. On the Delete dialog confirm or deny deletion using the following options:



- **Yes** deletes only the single selected process shown in the message.
- **Yes All** deletes all selected processes.
- **Skip** does not delete the single selected process shown in the message.
- **Cancel** does not delete any of the selected processes.

Since deleting a queued process attempts to move it to Ended Processes, you must confirm that you want to stop the background tasks.

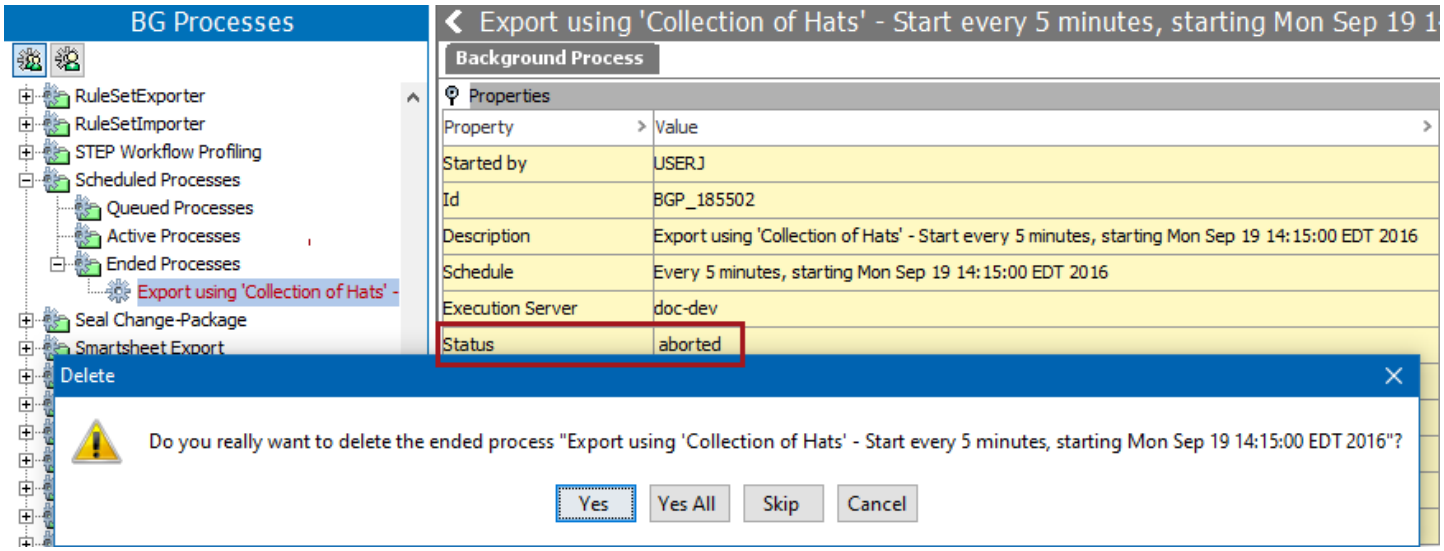


4. Select an aborted process in Ended Processes.

Important: Once deleted, an aborted scheduled process is removed from STEP and cannot be restored.

5. From the Maintain menu, click **Delete** to display the Delete dialog again. Confirm or deny deletion using the following options:

- **Yes** deletes only the single selected process shown in the message. The scheduled process is removed from STEP.
- **Yes All** deletes all selected processes. The scheduled processes are removed from STEP.
- **Skip** does not delete the single selected process shown in the message. The scheduled process remains in aborted status.
- **Cancel** does not delete any of the selected processes. The scheduled process remains in aborted status.



Managing Background Processes

The longer you use STEP, the more background processes are generated. Over time, this creates a large amount of data that has no long-term value. Managing this data involves deleting completed processes to reduce the amount of unnecessary data.

Automatic Deletions

By default, all ended background processes auto-delete after 168 hours (one week). When a process is older than the number of hours specified, it is deleted.

Important: Auto-delete removes the process from the application server and also from the file server.

The AutoDeleteBackgroundProcesses control when processes are deleted. Properties are case-sensitive and can be written to the sharedconfig.properties file on the application server.

The admin portal includes information on reviewing system properties. See the **Accessing the Admin Portal** documentation and the **Configuration** documentation, both of which can be found in the **Admin Portal** guide.

AutoDeleteBackgroundProcesses.AgeInHours.[Process Type ID]

This property determines how old the ended background processes must be before being deleted. This applies to processes that have ended in the 'Aborted', 'CompletedWithErrors', 'Succeeded', or 'Failed' states.

The default number of hours can be modified for the export, import, and web publishing processes using these properties:

```
AutoDeleteBackgroundProcesses.AgeInHours.Exporter = 168
AutoDeleteBackgroundProcesses.AgeInHours.Importer = 168
AutoDeleteBackgroundProcesses.AgeInHours.WebPublisher = 168
```

AutoDeleteBackgroundProcesses.DelayInHours.[Process Type ID]

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].

```
AutoDeleteBackgroundProcesses.DelayInHours=1
```

AutoDeleteBackgroundProcesses.OperatingHours.[Process Type ID]

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.

```
AutoDeleteBackgroundProcesses.OperatingHours
```

AutoDeleteBackgroundProcesses.OperatingHoursTimeZone.[Process Type ID]

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.

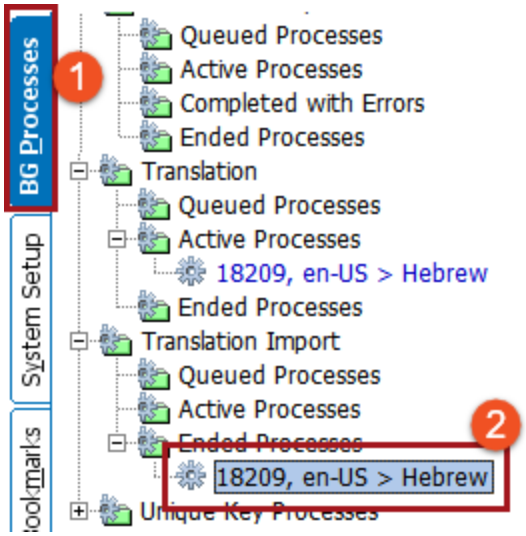
```
AutoDeleteBackgroundProcesses.OperatingHoursTimeZone
```

Note: It is important to note that this property should not be set unless the AutoDeleteBackgroundProcesses.OperatingHours property has also been set.

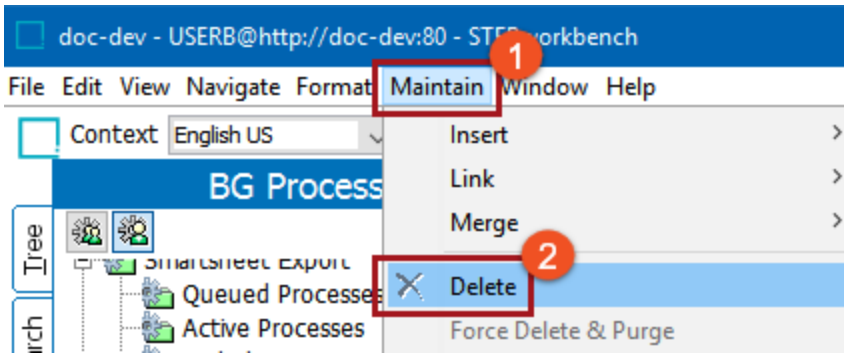
Manual Deletions

Background processes may be deleted manually as follows.

1. In workbench, click the **BG Processes** tab
2. Select the background process to be deleted.



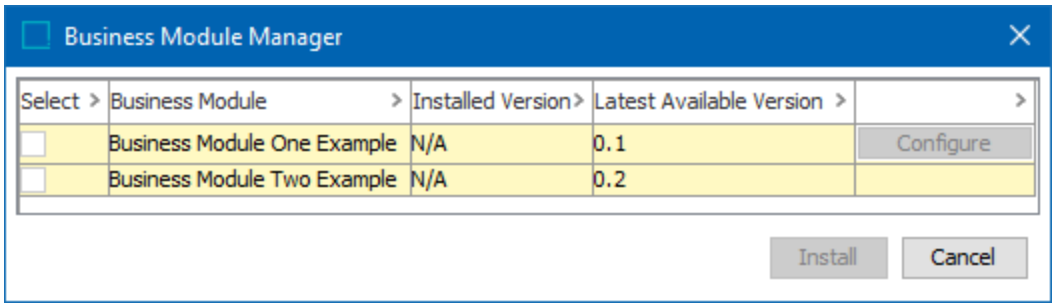
3. From the toolbar, click the **Maintain** menu.
4. Select **Delete** to remove the background process.


doc-dev - USERB@http://doc-dev:80 - STE orkbench
File Edit View Navigate Format Maintain Window Help
1
Context English US
BG Process
Tree
Queued Processes
Active Processes
Insert
Link
Merge
2
Delete
Force Delete & Purge

Business Module Manager

A business module is a functional configuration which handles use cases based on common practices observed in the industry. The Business Module Manager allows an administrator to install a pre-configured data model for a business purpose, such as the 'sell side' of the product enrichment process, or the 'buy side' for the vendor data onboarding process. Business module installation can include the creation of product objects types, classification objects, lists of values types, etc.

The 'Business Module Manager' option displays on the File menu for all users. The install functionality is available for users who are assigned the 'Install Business Modules' setup action privilege. For more information, see the **Setup Actions** topic.



When a business module is activated by Stibo Systems, it displays in the Business Module Manager dialog. The dialog includes:

- Select - checkbox to indicate a module should be installed.
- Business Module - names of the activated modules. Hovering over the business module name displays a tooltip with a description of the module.
- Installed Version - the version of the module currently on your system.
- Latest Available Version - the current available activated version.
- Configuration - when additional parameters required to install the module, the 'Configure' button is enabled.

Installing a Business Module

1. Activate the 'Install' button by meeting the requirements for the selected modules, as follows:
   - If multiple modules are selected for install, safeguards ensure any required order is followed.
   - If the 'Configure' button is enabled, click to display an additional dialog with required parameters.
2. Click the Install button to start the installation background process.
3. Review the BG Processes progress under the 'Install Business Modules' node.

   If an error occurs during install, the BGP Execution Report shows the problem and actions are rolled back. The Installed Version column is not modified.

Available Business Modules

- **PIM for Retail**: see the **Solution Enablement** documentation.

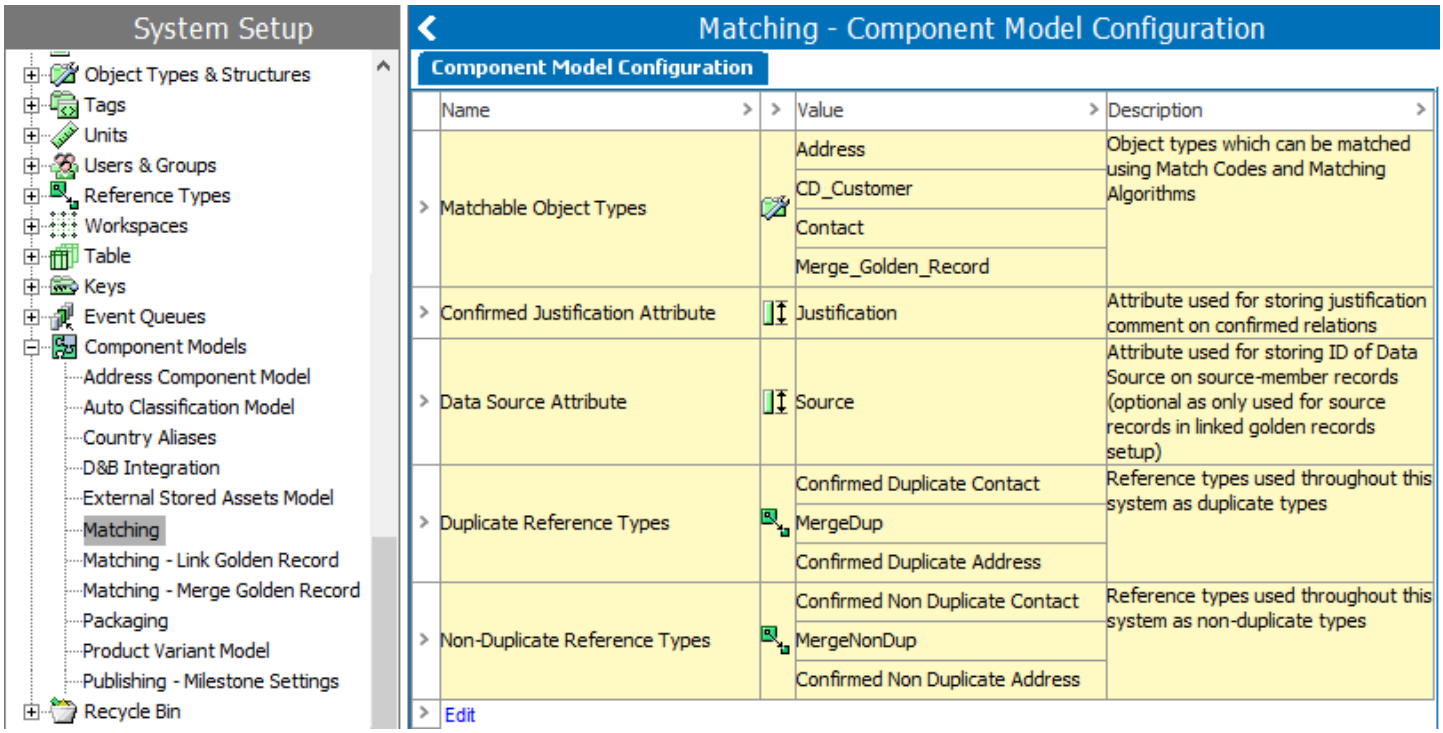
Component Models

Component Models provide configuration instructions for functionalities that require multiple objects, references, or other elements to perform a specified operation. For example, to use Matching functionality (as shown below) the component model identifies the object types used for matching, the attributes used to hold justification and source information, as well as the reference types used to identify duplicates and non-duplicates.

Exporting component models with STEPXML contains references to the applicable reference types, object types, and attributes objects. Alternatively, exporting the definition of referenced objects is possible with the 'V1' setting on the 'Export.ComponentModel.Version' property. To export definitions, in the STEP application server sharedconfig.properties file, add the case-sensitive property as shown below:

```
Export.ComponentModel.Version=V1
```

By default, the value for this property is 'V2' and indicates that actual references are included in the export.



Generally, the available component models are explained within the applicable section of documentation. For information on the component models, see the following topics.

Automatic Classifications

- Auto Classification Model - in the Automatic Classification documentation

Automotive

The following component models are defined in the Automotive Quick Start Guide.

- Automotive – Application Model
- Automotive – AutoCare Model
- Automotive – Import Flow Process
- Automotive – NAPA Model
- Automotive – TecDoc Model

The following component model is defined in the Automotive Reference Guide.

- Automotive – Country Model

Data Integrations

- Address Component Model - in the Loqate Integration section of the Data Integration documentation
- CASS Address Component Model - in the Loqate Integration section of Data Integration documentation
- D&B Integration - in the Dun & Bradstreet Integration section of the Data Integration documentation
- Email Component Model - in the Experian Email Validation Integration section of the Data Integration documentation

Digital Assets

- Asset Analyzer Component Model - in the Asset Analyzer section of the Digital Assets documentation
- Asset Download Component Model - in the Asset Download section of the Digital Assets documentation
- External Stored Assets Model - in the Storing Assets Externally documentation

Matching, Linking, and Merging

The following component models are defined in the Matching, Linking, and Merging documentation.

- Matching
- Matching - Link Golden Record
- Matching - Merge Golden Record

Packaging

- Packaging - in the Packaging Hierarchy Editor topic within the Web User Interfaces / Web UI Setup and User Guide

Product Variants

- Product Variant Model - in the Product Variants section of the System Setup / Super User Guide documentation

Publishing

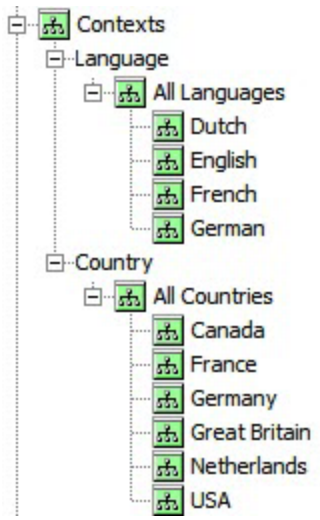
- Publishing – Milestone Settings - in the Flatplanner section of the STEP Publisher (InDesign) documentation

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, recommended 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 Recommended Practices
- Adding a Dimension Dependency After Loading Data
- Contexts
- Maintaining Contexts
- Context Recommended Practices
- Context Locales

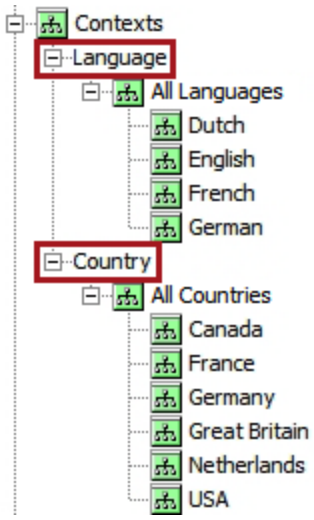
Dimensions and Dimension Points

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

Dimensions

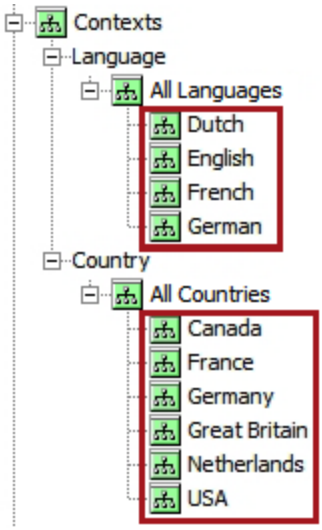
A **dimension** is a definition of a context-sensitive domain in STEP, such as language or country, which allows for different layers of information in which content can vary. A dimension is presented in STEP as a hierarchy of dimension points. Each system must have at least one dimension.

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 Recommended 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** / **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 Points Recommended 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.

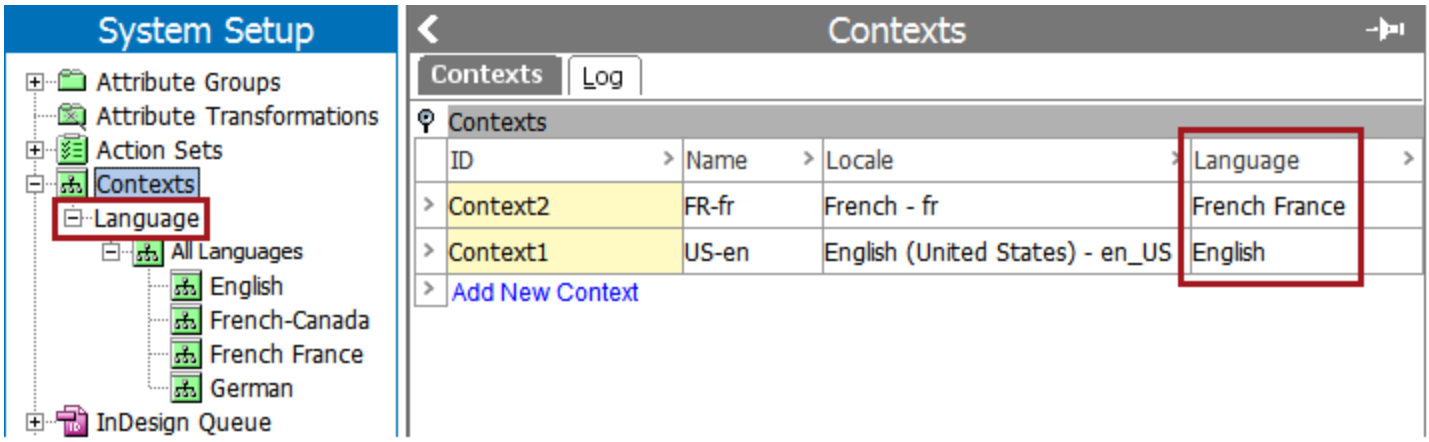
Arguments for Additional Attributes	Arguments for Additional Dimensions
Users can compare values for all versions side-by-side in an Excel file.	Users can compare values for a single attribute across dimensions using the STEP context mode. Select this icon from the workbench to switch to STEP context 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.

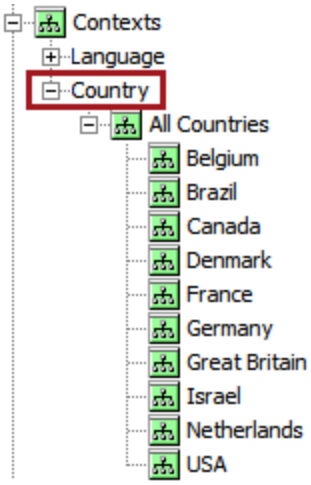
Even if you do not use multiple languages, you must have a language dimension for the language in which you are maintaining your data. Choose a real language so that it is very clear what belongs there. The language dimension is special in STEP because it tracks changes between a source language and 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 screenshot below shows a dimension hierarchy in System Setup and two contexts that only use the Language dimension.



Country Dimension

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



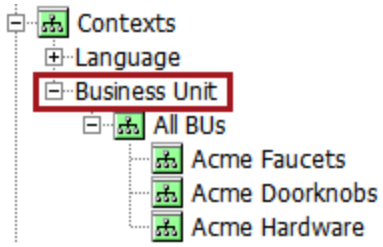
You should only use a country dimension if the following conditions apply:

- 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 Systems GDSN solution, which requires that each target market references a context containing a language and country dimension.

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

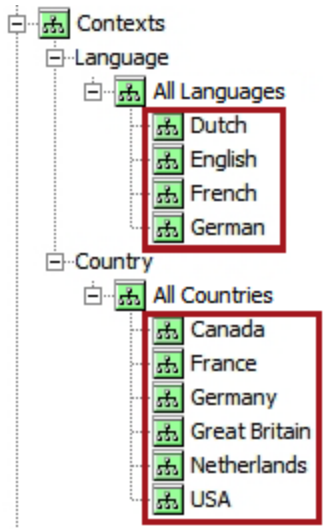
Alternatives to market dimensions are:

- 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

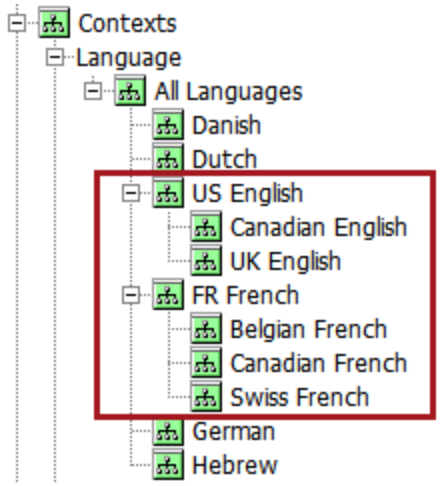
Dimension Point Do's and Don'ts

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.

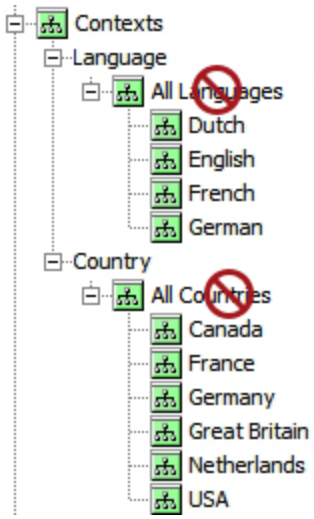


For more information on use cases and the value inheritance of parent / child dimensions and attribute values, see the **Dimension Dependent Attributes** topic.

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

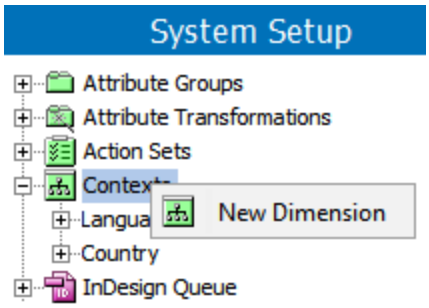
This topic describes how to:

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

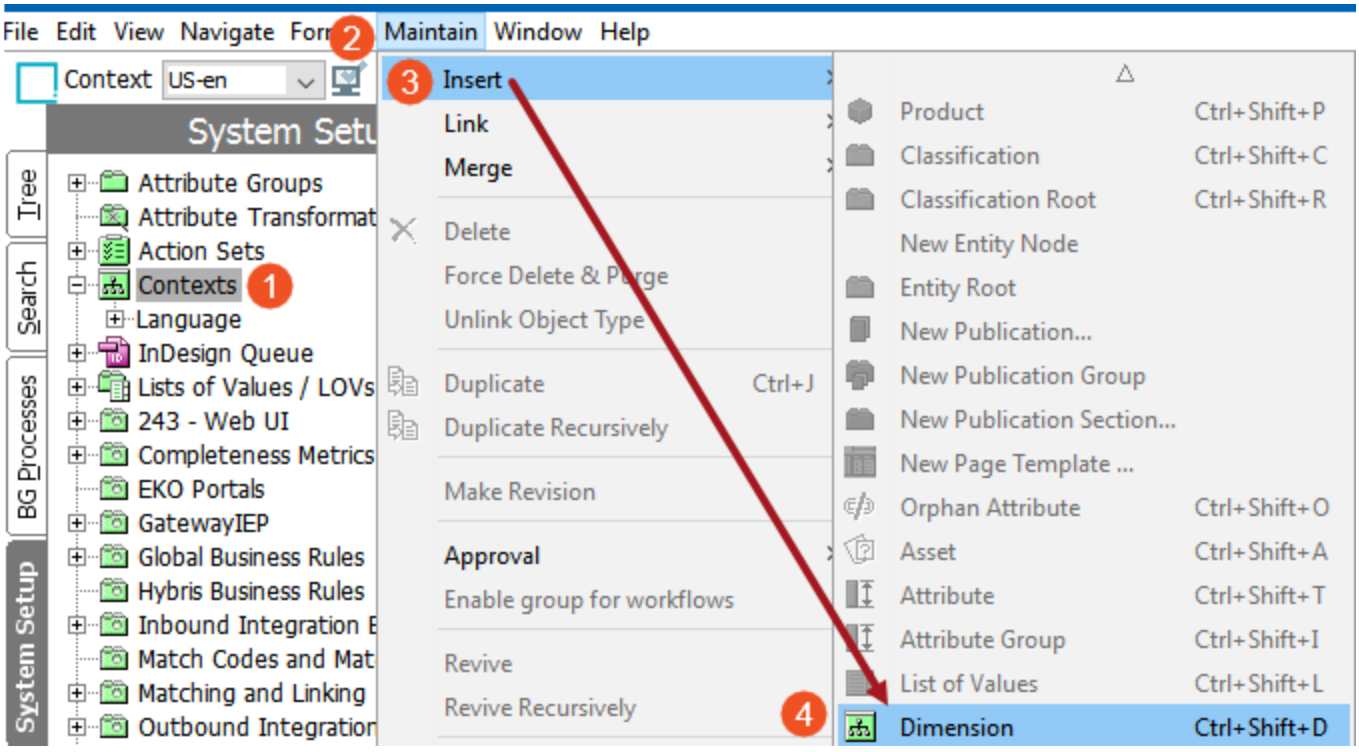
Creating a Dimension

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.

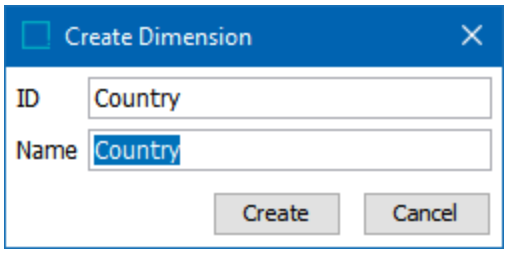
1. Navigate to System Setup > **Contexts**.
2. With Contexts selected, right-click and select **New Dimension**.



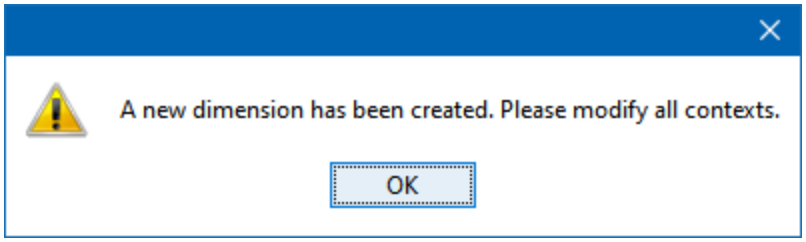
- Alternatively, select Contexts and go to Maintain > Insert > **Dimension**.



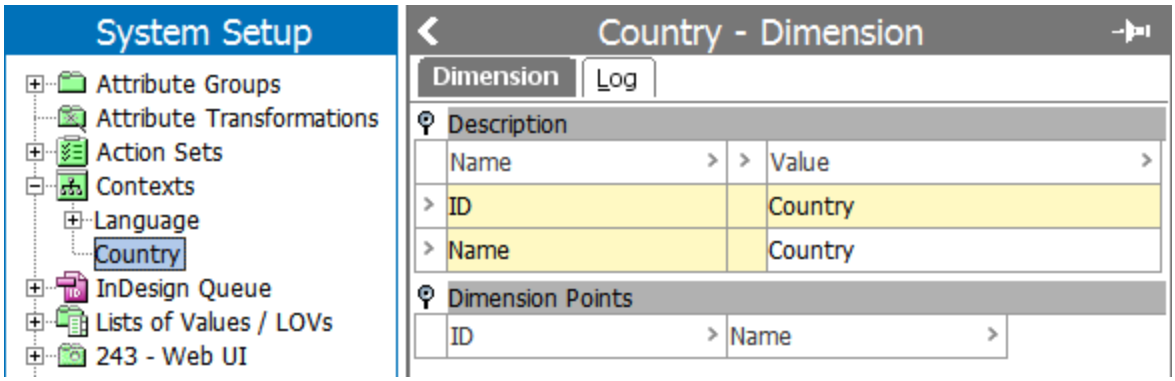
3. In the **Create Dimension** dialog that displays, enter an ID and Name for the new dimension, then click **Create**. If the ID and Name should be identical, press 'Enter' on the keyboard after typing the ID and it will repeat in the 'Name' field.



4. Click **OK** on the confirmation dialog that displays.



5. 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.

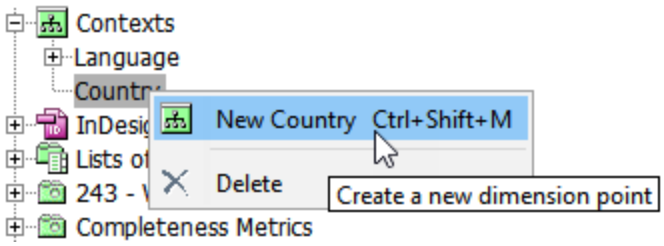


Creating a Dimension Point

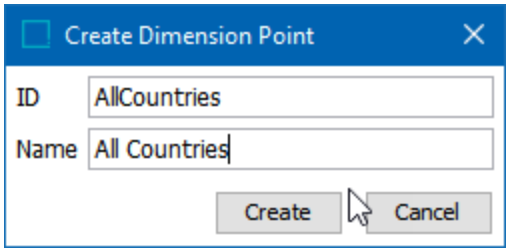
After a new dimension has been created, the next step is to create at least one dimension point below it.

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).

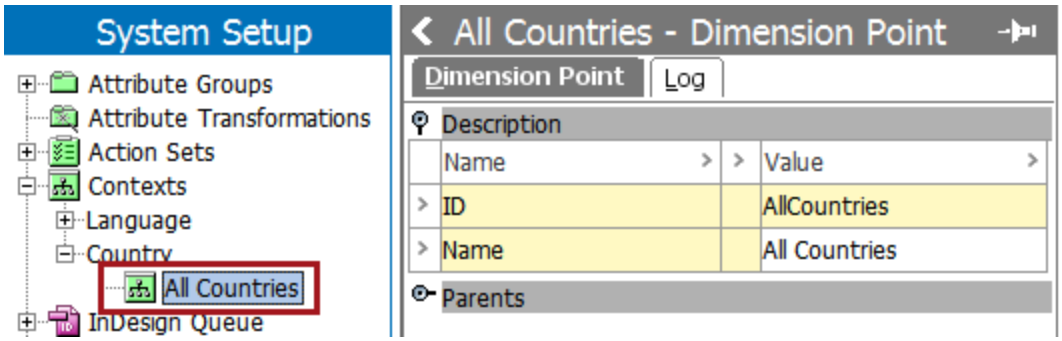
> **Note:** The first dimension point created beneath a dimension must be a top-level **All** level, which is where *non*–dimension-dependent values are stored. Specific dimension points are created below the All level.



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.

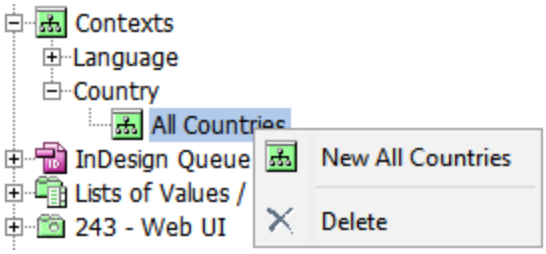


4. The new dimension point is created.

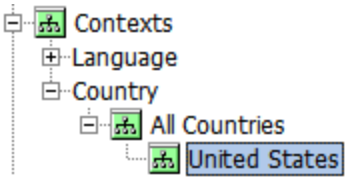


5. To create additional dimension points below the All level, select the All level, right-click, and click **New** (for example, New All Countries).

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



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

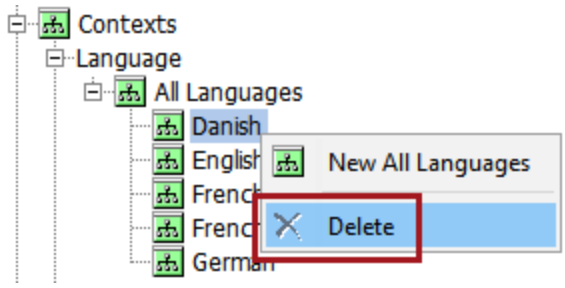


Deleting a Dimension Point

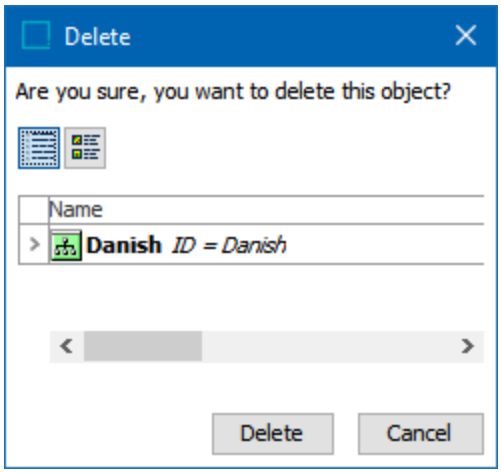
If you need to delete a dimension or dimension point, you must first delete all dimension points that exist below it.

> **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.

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



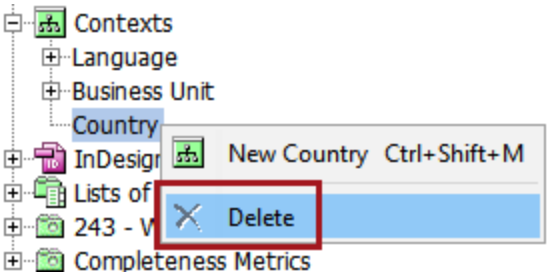
3. In the Delete dialog that displays, click **Delete** to remove the dimension point.



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**.



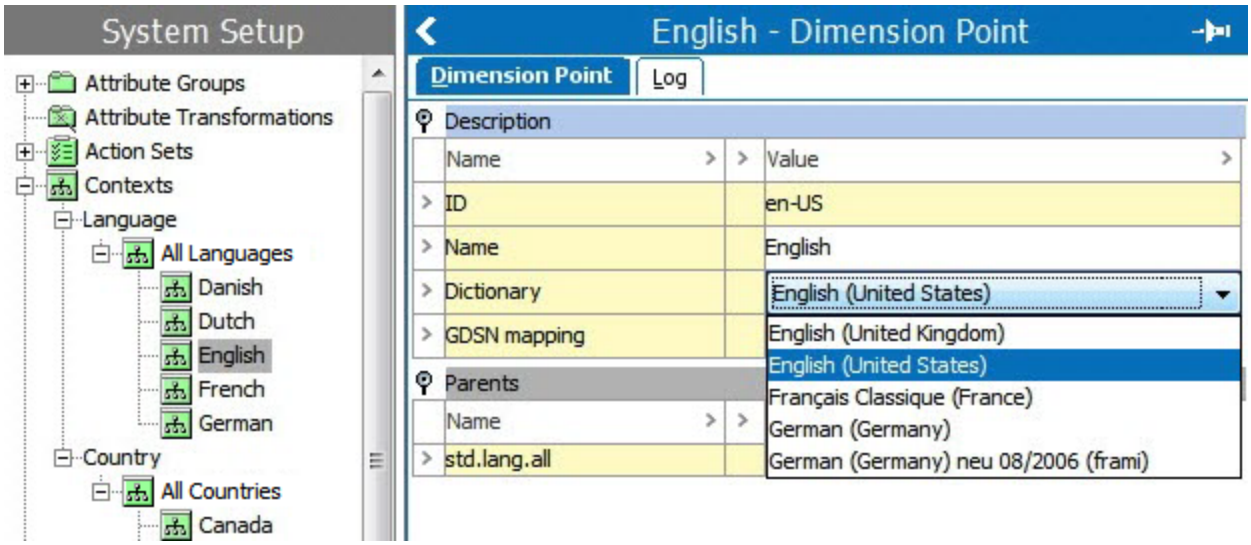
3. In the Delete dialog that displays, click **Delete** to remove the dimension.

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.

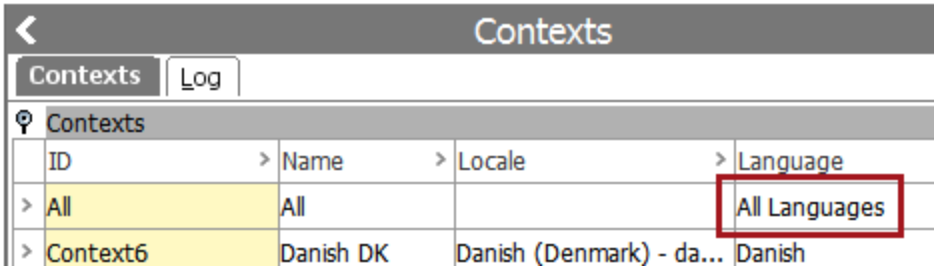


Adding a Dimension Dependency After Loading Data

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

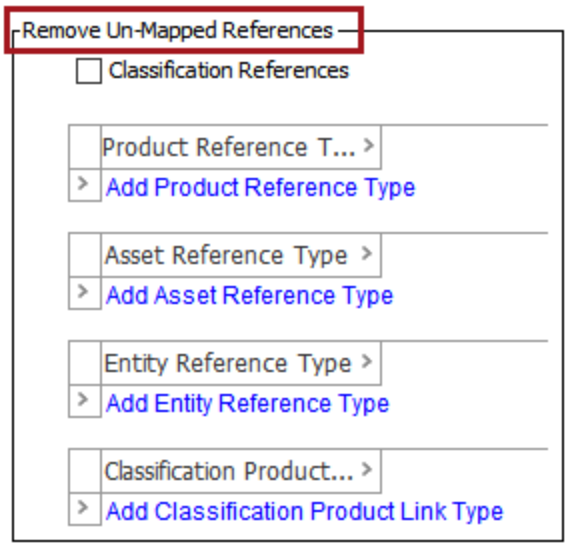
1. Temporarily create a context that has the 'All' level of the dimension you want to apply to the attribute.



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

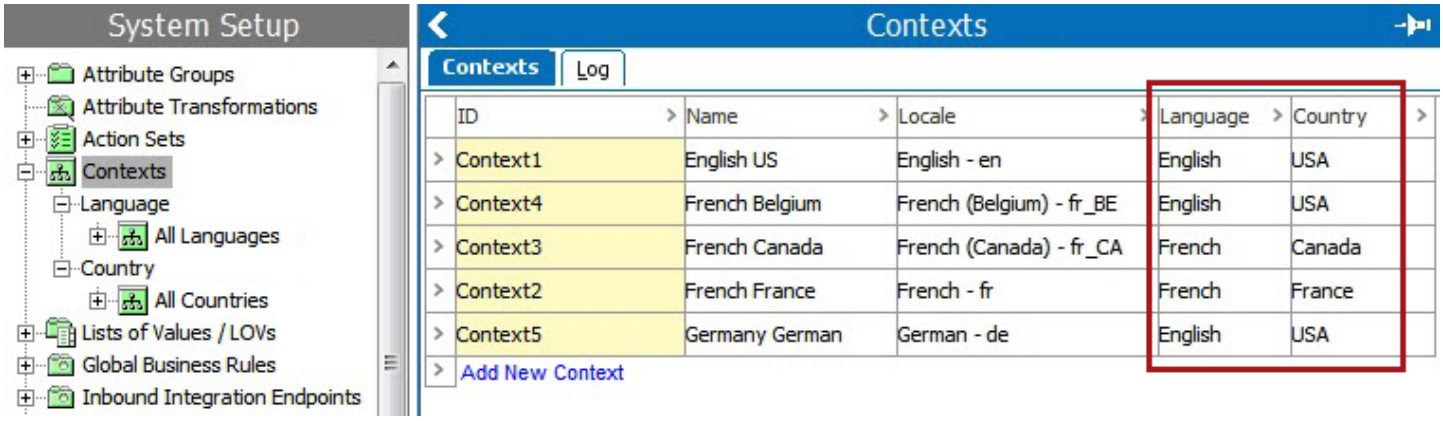
1. Temporarily create a context that has the 'All' level of the dimension you want to apply to the reference type.
2. In the Export Manager, select all products in the database and map the reference types that you want to apply the dimension to.
3. Save the export configuration.
4. Save two copies of 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.



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.

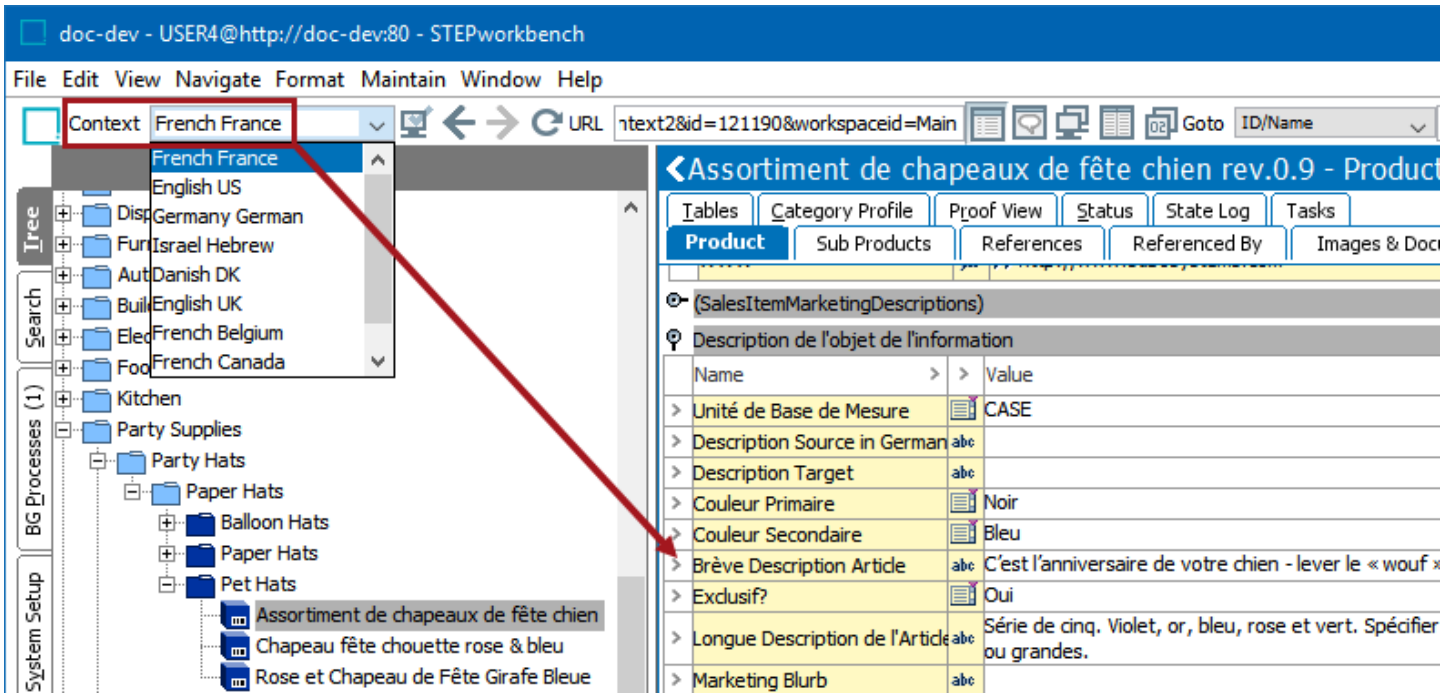
Contexts

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.



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.

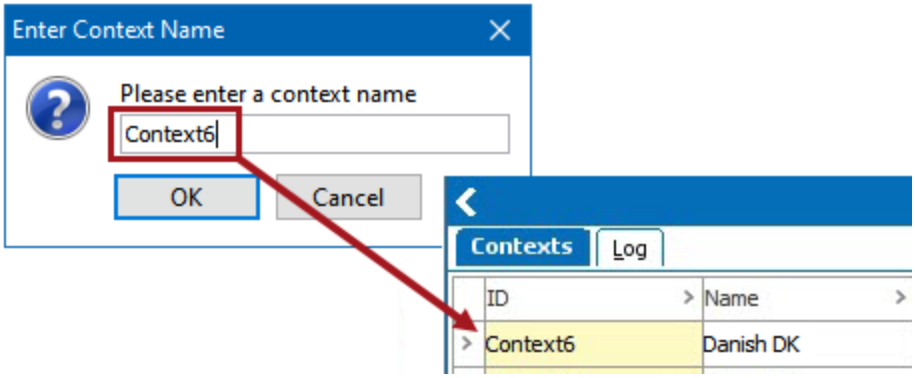
You can also find information about contexts and Web UI functionality in the **Multi Context Screen** topic and the **Corner Bar Context Selector Component** topic in the **Web User Interfaces** documentation.

Context Recommended 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 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.

Global Context

- Do not use a **global** context. 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.

Deleting the Global Context

If a global context already exists on your system—and **there is no content** whatsoever in it—then *delete it*.

Important: If a global context already exists on your system—and **it contains content**—*do not delete it* unless you plan to first move the data into a different context. Deleting a global context that contains data will affect ALL data in the context; not just attribute values, but LOV values, metadata on links, references, assets, etc.

Removing a global context from an existing system requires more than a small amount of analysis and possibly significant cleanup / rework (repushing assets, for example). Additional reasons to not to delete a global context with data include:

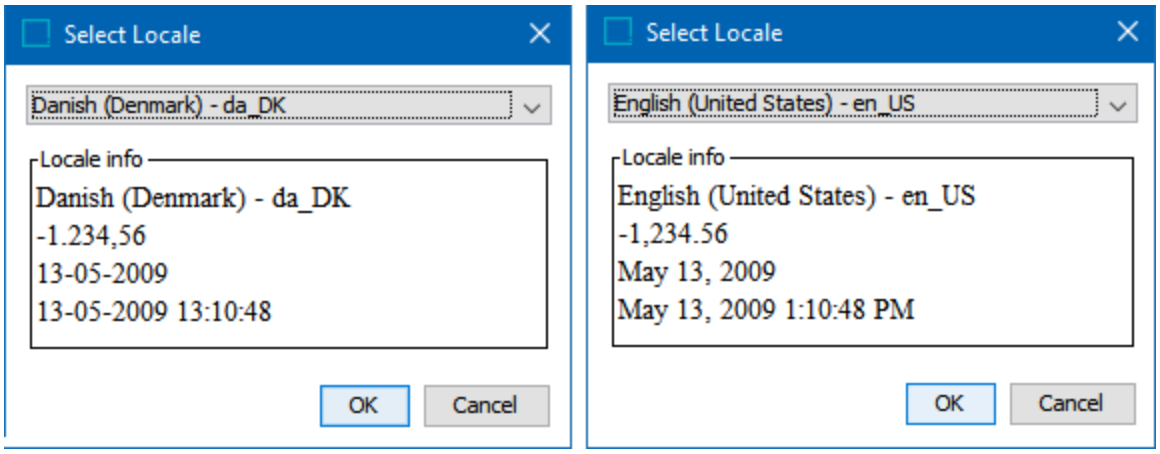
- If the data is inherited from the global context by other contexts, though the data will still exist, it will likely no longer be editable.
- If the context is assigned to any configurations (integration end points, export, import, etc.), then these configurations will no longer be valid.
- Assets, if dimension dependent, are also affected, especially in a print initiative where images are pushed to folder structures corresponding to the dimension. This means that functions such as 'Replace Asset Content' would be impossible.

Context Locales

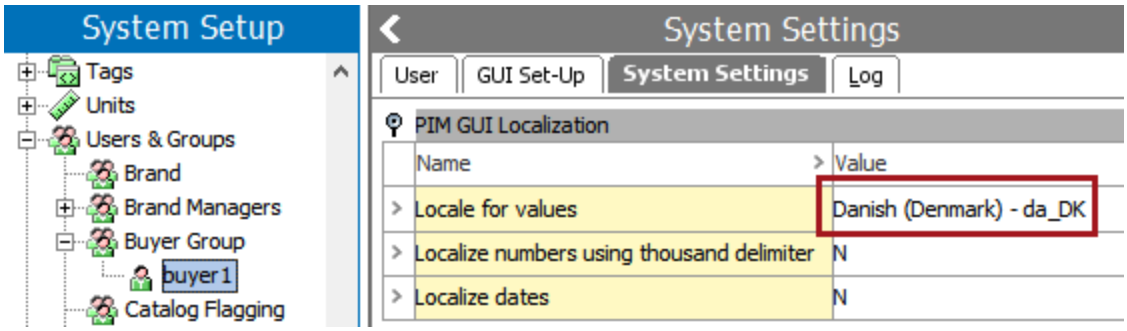
A locale is a part of a context. Locales control the display for users in the workbench, as well as Web UI buttons, tabs, and so forth.

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 and Numeric Text (exclude tags) are really text attributes and cannot be accommodated with the locale functionality.



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

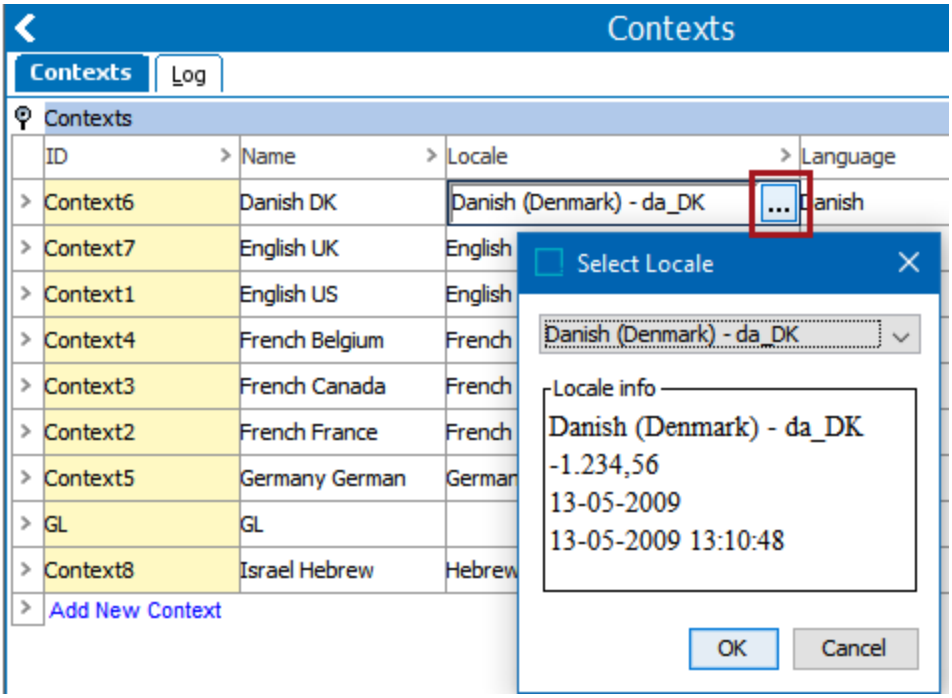


For more information on using locale with users in System Setup > System Settings, see the **GUI Localization** topic in the **System Setup / Super User Guide** documentation.

Locales are also used in attribute transformations, STEP table transformations, and in STEP Publisher.

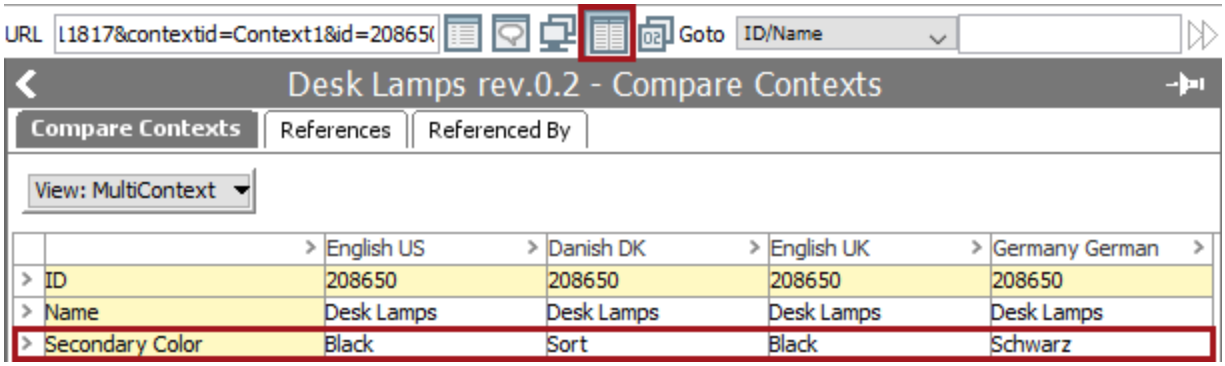
Locale Do's

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



Context Mode View

The context mode view allows users to compare values in multiple contexts. For example, selecting contexts that include values in different languages displays the language-dependent attribute values for each context.



Using Context Mode View

1. Click the context mode button ( ) in the toolbar to display the Compare Contexts view.
2. Modify the contexts displayed using the View menu > **Target** option, as described in the **View Menu** section of the **Menu Items** documentation.
3. Limit the number of attributes displayed using a **View**. For more information, see the **Multi-Editors** section of the **Editing Objects in the Tree Tab** topic of the **Getting Started / User Guide** documentation.
4. Right-click a column header or data row to show filter and sort, rotate, and hide options.

Maintaining Contexts

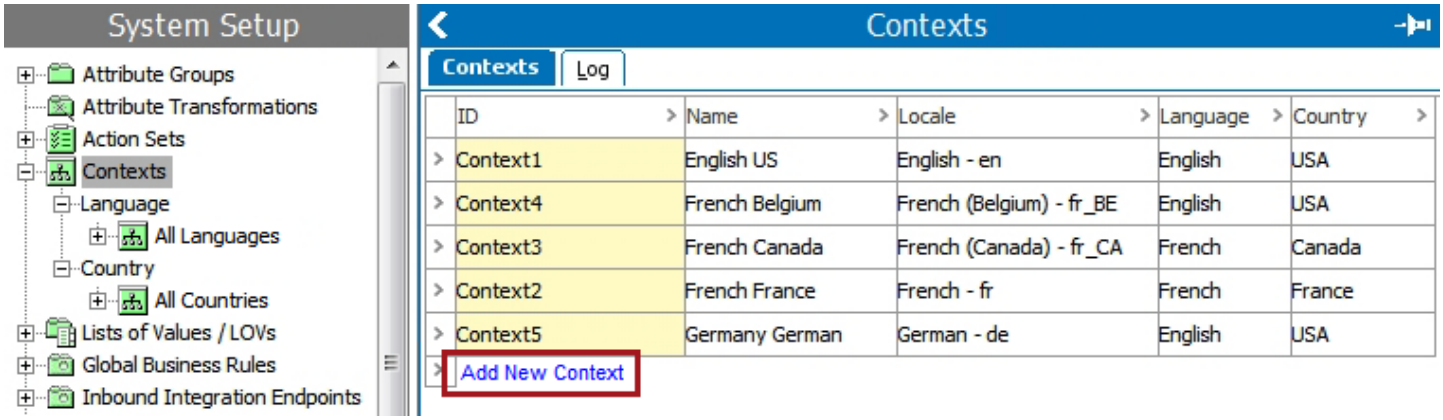
This section describes how to create, edit, and delete contexts. Also explained is the concept of hidden values, which can be caused when a context is deleted, and how to specify a default context.

Creating a Context

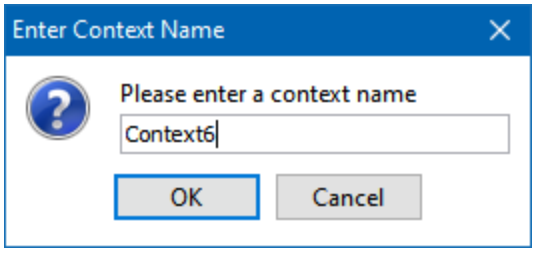
Contexts are maintained in System Setup > **Contexts**. A context must contain at minimum one language dimension point; for example, English.

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

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

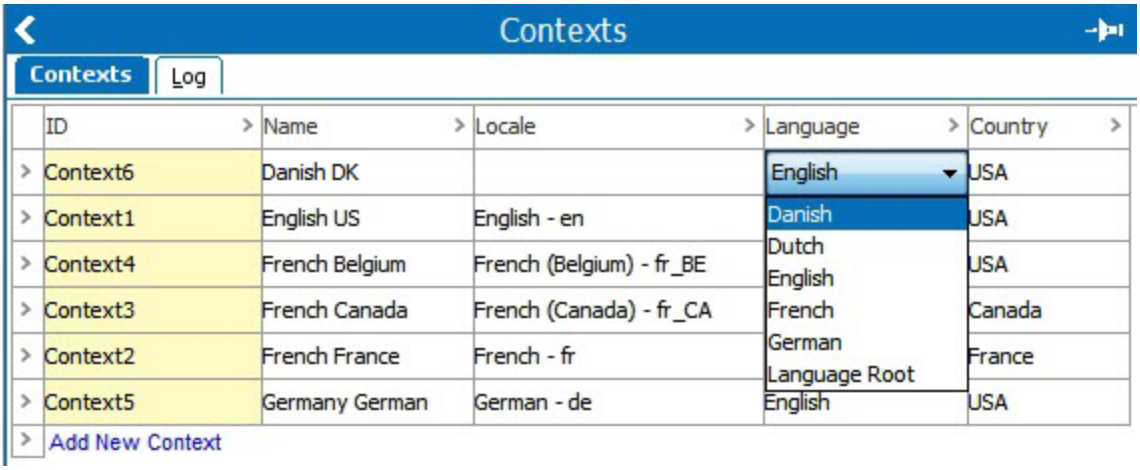


2. In the **Enter Context Name** dialog that displays, enter a name for the context, then click **OK**. By default, the name entered in this field also becomes the context **ID**. See the **Context Recommended Practices** topic in this documentation for suggestions on what ID format to use when naming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



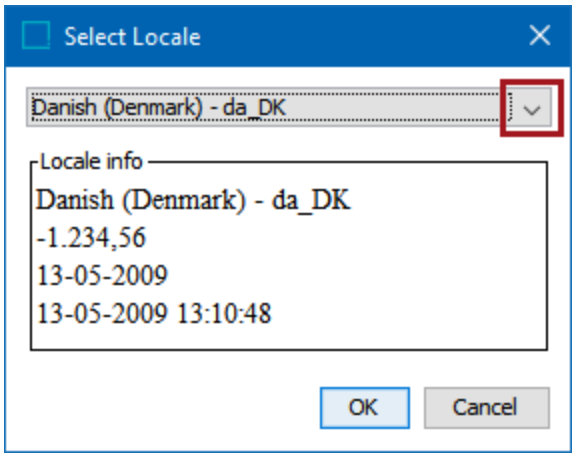
3. A new 'default' context appears with a blank locale and English as the language. If other dimensions exist, such as Country, a 'default' value will also appear in that column, such as 'All Countries' or 'USA.'
4. In the **Name** column, replace the auto-created name with a new name for the context,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

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

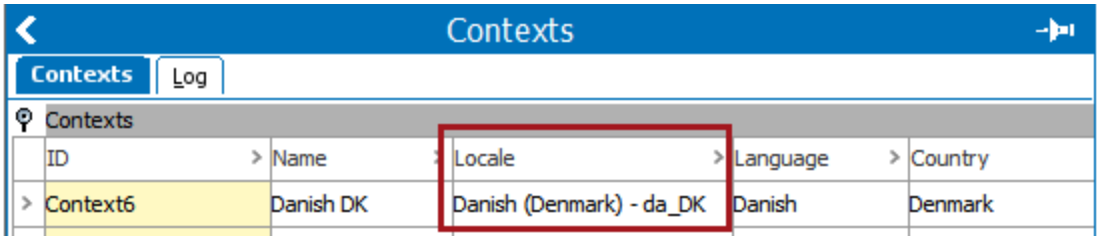


6. Click the blank field in the **Locale** column, then click the ellipsis button (...) to select the locale that will be used for data in the context.

7. In the **Select Locale** dialog that displays, choose the relevant locale from the dropdown list, then click **OK**.



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



9. 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:

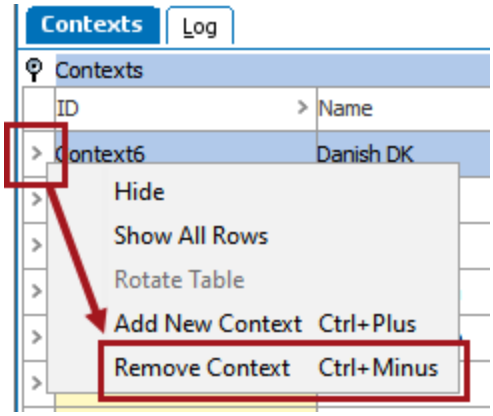
- Enter a new name in the Name field
- Choose a new locale
- Select a new dimension point from the dropdown menu in one of the dimension columns (such as Language or Country)

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

Deleting a Context

To delete a context:

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

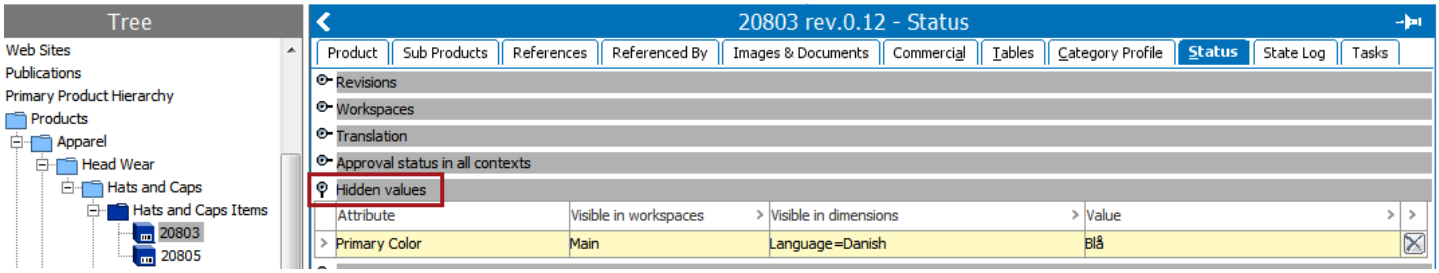


2. Click **Remove Context**, or press Ctrl + Minus on your keyboard
3. The context is deleted.

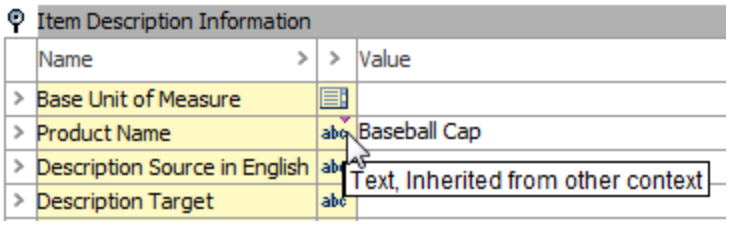
Hidden Values

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:

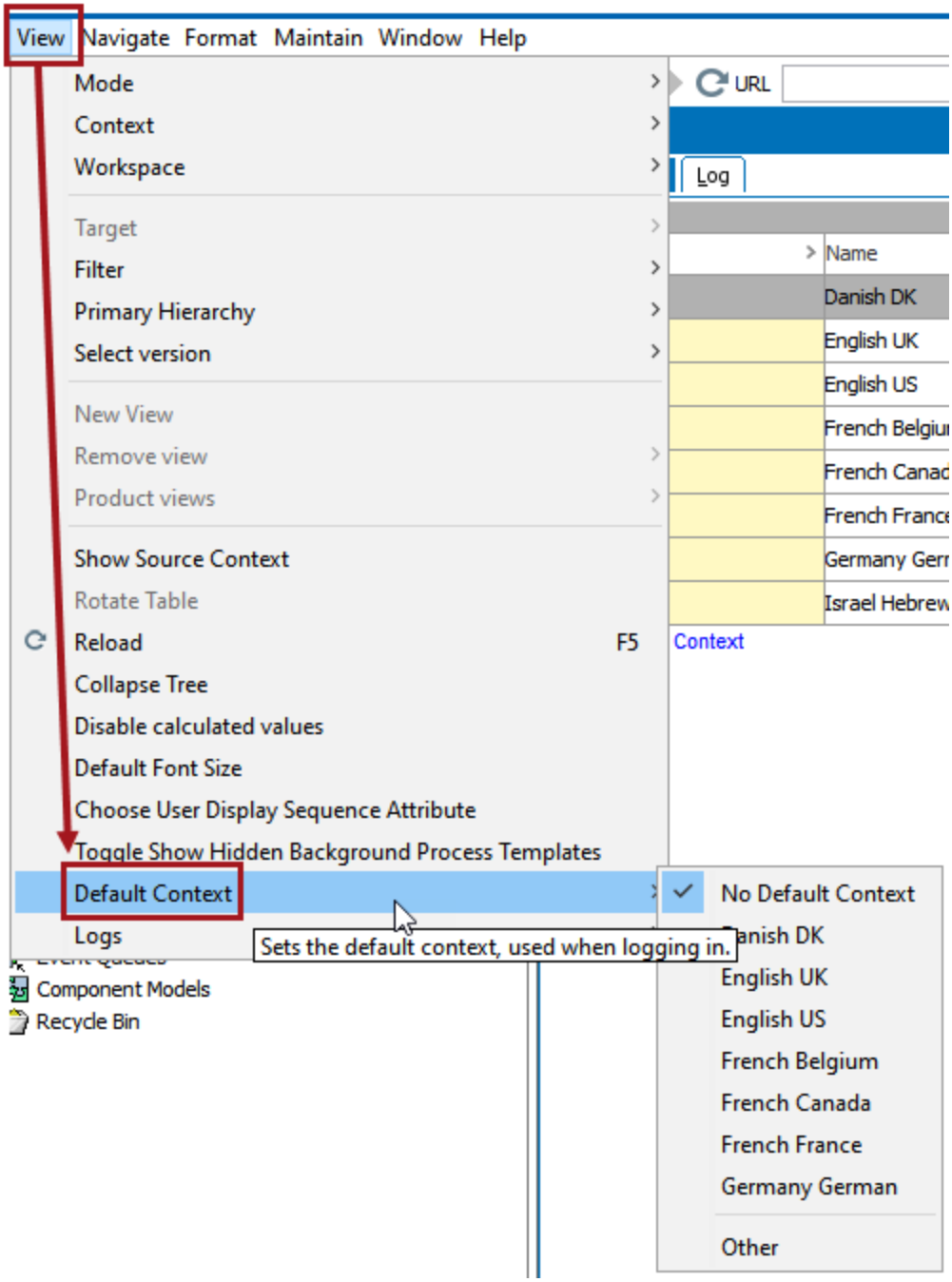


Note: Deleting a context can also cause a product to not be fully approved even if it is approved in all contexts. This is because there are still existing values in a deleted context.

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**.

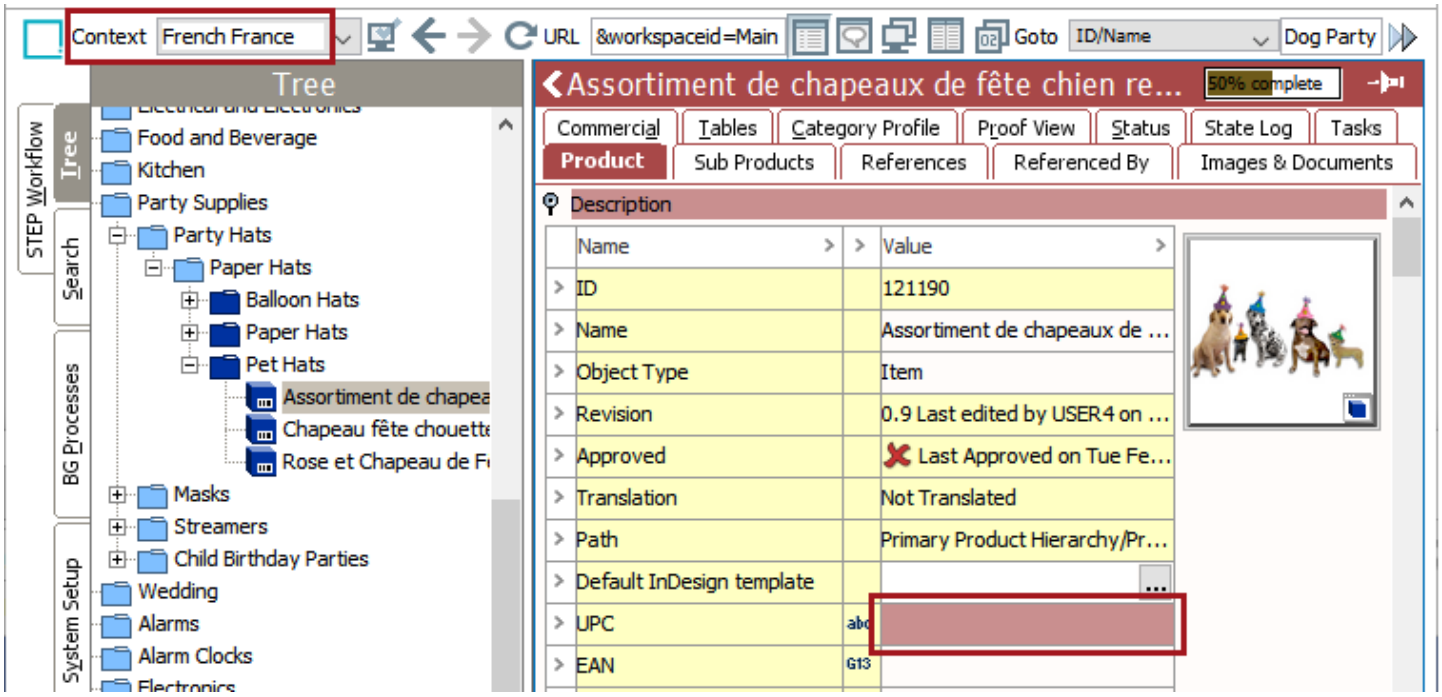


2. Select the preferred context from the list of contexts that displays. If you do not see your context, click **Other**.
3. Choose your preferred context from the list in the 'Other Contexts' dialog, then click **OK**.



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.



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 **Revisions** 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.

Core Event	Event Type	Action Performed to Generate the Core Event
GLOBALEVENT	Create	One of these System Setup objects is created: attribute, attribute group, list of values group, list of values, unit group, unit, product to classification link type, and reference type objects.
GLOBALEVENT	Delete	One of these System Setup objects is deleted: attribute, attribute group, list of values group, list of values, unit group, unit, product to classification link type, and reference type objects.
GLOBALEVENT	Modify	One of these System Setup objects is modified: attribute, attribute group, list of values group, list of values, unit group, unit, product to classification link type, and reference type objects.
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.

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, see the:

- **OIEP - Event-Based - Event Triggering Definitions Tab** section of the **Outbound Integration Endpoints** documentation.
- **Event Triggering Definitions Tab** section of the **Event Processors** documentation.

Important: When exporting delete events, generally events are processed in the same order as they occurred. However, in some circumstances, the order is not reliable. For more information, see the **Event-Based OIEP Order of Delete Events** topic.

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 and leave the field (save is automatic)	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 and leave the field (save is automatic)	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 start 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 **Revisions**.

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 **Revisions**.

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**.
- No change flags are produced for calculated attributes since the values are not stored in the database. For more information, see the **Calculated Attributes** topic.
- 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:
		Event 1:
		
		`<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>`
		
		Event 2:
		
		`<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>`

Txn	Action	Event XML Processed
	• Prior revision values: **Y in 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.	<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>
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** • Approved values: **Y in USA** and **Y in Canada**	Event 4: <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>

Txn	Action	Event XML Processed
	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:
		
		`<Products>`
		`  <Product ID="181120" UserTypeID="Item" ParentID="181119">`
		`  <Name QualifierID="USA">Acme Anvil</Name>`
		
		`  <Values>`
		`  </Values>`
		`  </Product>`
		`</Products>`
		
		Event 6:
		
		`<Products>`
		`  <Product ID="181120" UserTypeID="Item" ParentID="181119">`
		`  <Name QualifierID="USA">Acme Anvil</Name>`
		
		`  <Values>`
		`  </Values>`
		`  </Product>`
		`</Products>`

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 trigger 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.

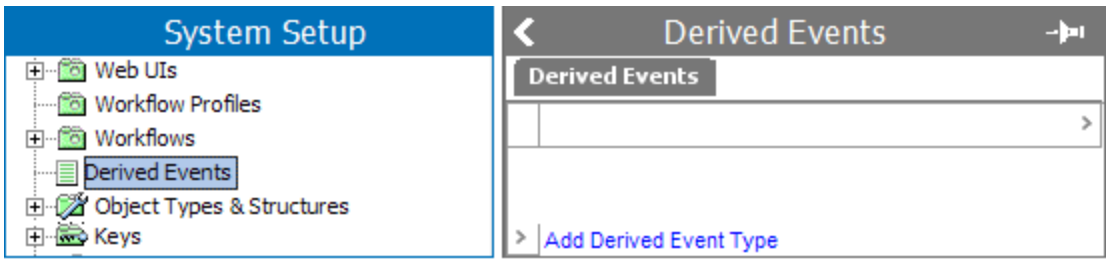
Also, a thorough understanding of business rules is needed to create successful derived events. For more information, see the **Business Rules** section, and see the **Event-Based Example Business Rules for Derived Events** section of the **Data Exchange** documentation.

Add a Derived Event Type

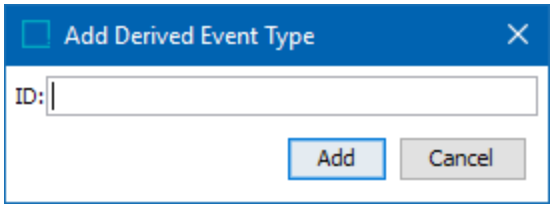
A derived event type must exist before it can be configured for an OIEP.

To create a derived event type:

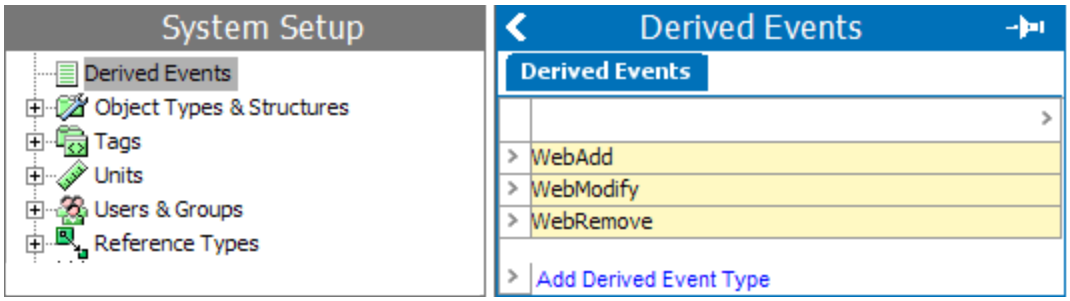
1. Go to System Setup.
2. Click **Derived Events**.



3. On the Derived Events tab, click the **Add Derived Event Type** link, and the Add Derived Event Type dialog will display.



4. 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.



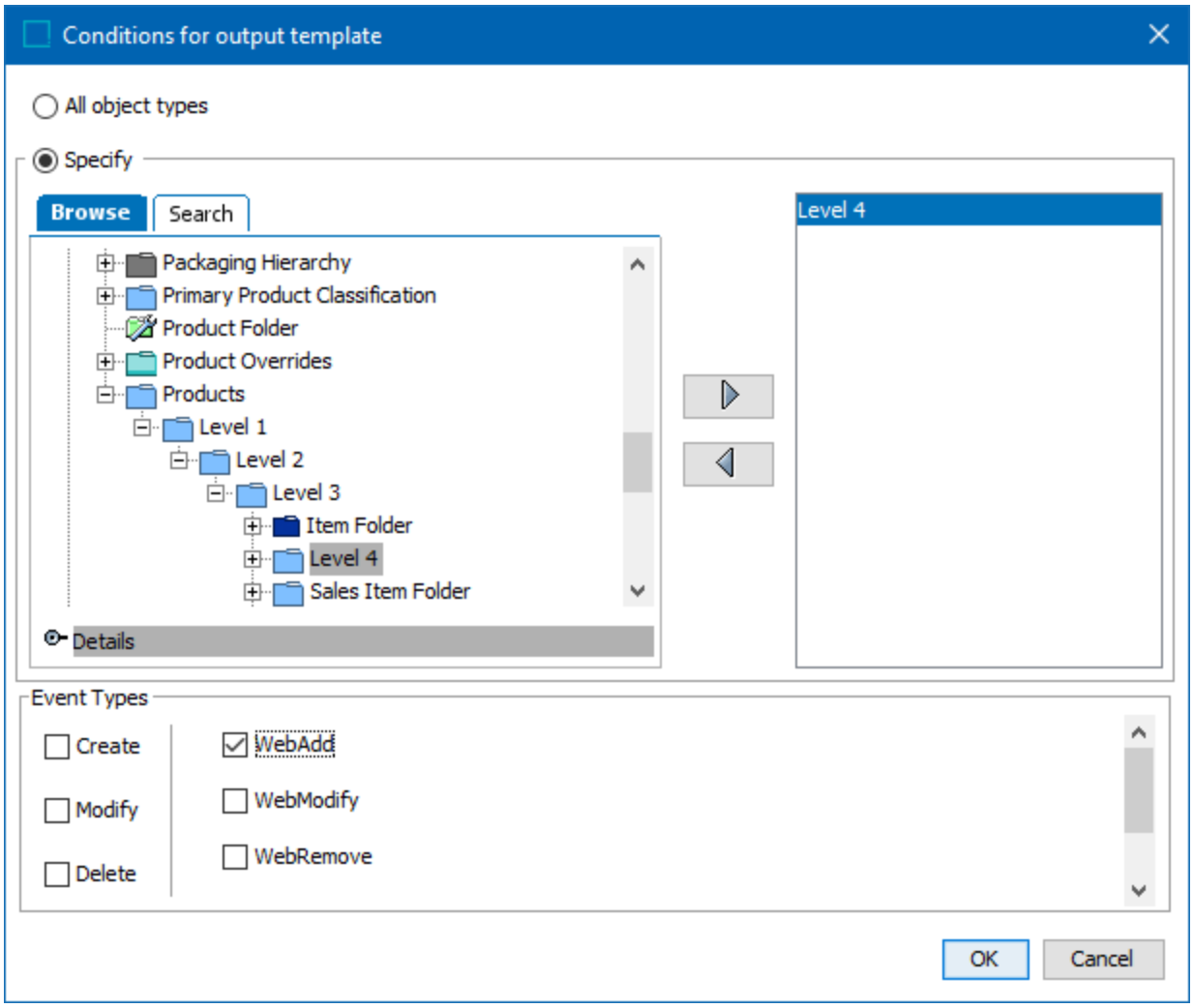
Use a Derived Event Type in an OIEP

When configuring an event-based OIEP output template, derived event types can be selected to enable sending files formatted in different ways or to include different data based on the action that triggers the event. For example, when pushing data to a website, consider the following derived event types:

- WebAdd - when an object is first linked into a classification.
- WebModify - when an object link to a classification is modified.
- WebRemove - when an object is unlinked from a classification.

To use a derived event type in an OIEP:

1. In an event-based OIEP, Configuration > Output Templates flipper > click the **Add configuration** link, or edit an output template that is already displayed.
2. On the **Conditions for Output Template** dialog, select the object type and the derived event type. In this example the derived event type is 'WebAdd'.



3. Repeat the previous steps to create additional output templates for other derived events.

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

Output Templates

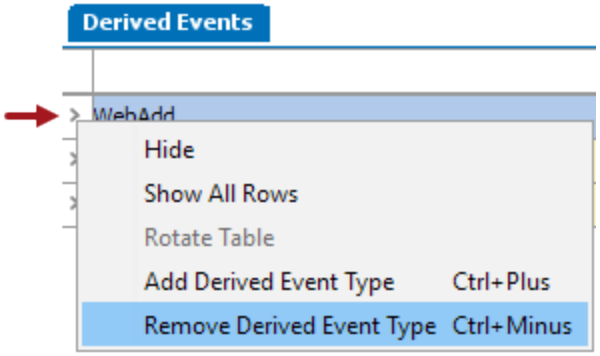
Object-Eventtype	Format	Pre-Processor	Post-Processor
Level4 (WebAdd)	Excel (3 mappings)	None	None
Level4 (WebModify)	Excel (3 mappings)	None	None
Level4 (WebRemove)	Excel (3 mappings)	None	None
Add configuration			

Delete a Derived Event Type

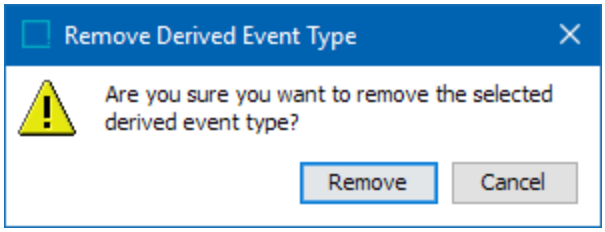
It is possible to delete a derived event type when it is no longer necessary. This also removes it from the available Event Types displayed when configuring an OIEP Output Template.

To delete a derived event type:

1. Go to System Setup.
2. Click **Derived Events**.
3. Find the derived event type to be removed, select the row icon, and the available options will display.



4. 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.

Events Generated on Main Workspace

By default, approval triggers an event for workspace revisable objects, meaning that from the Main workspace saving an edit (but not approving it) does not generate an event. However, in the Web UI Product Editor screen, events are needed for unapproved Main workspace edits so that the data is available for searching, sufficiencies, and PDX. To achieve this, the following additional manual configuration is required.

Important: The configuration below is only valid for unapproved updates saved on the Web UI Product Editor screen. The events generated are consumed by Product Data Exchange (PDX), the Search Screen, and the Sufficiency Panel.

Note: If you are using Search Screen / Elasticsearch functionality (with the X.Elasticsearch license enabled), as an alternative to the configuration below, you will have an OIEP / EP Triggering Workspace option when configuring Event Triggering Definitions and can opt to trigger events from the Main workspace. Refer to the **OIEP - Event-Based - Event Triggering Definitions Tab** topic in the **Data Exchange** documentation.

Saving edits on the Product Editor screen in Web UI generates a 'postAction' for each of the following components:

- **PDX** - defined in the **Product Data Exchange** topic of the **Data Integration** documentation.
- **Sufficiencies** - defined in the **Sufficiency Panel** topic of the **System Setup / Super User** documentation.

The postAction executes a specified business action which creates a derived event. The derived event is then placed in the specified event processor queue. When the event processor is invoked, the unapproved data is made available to the specified component.

Configure Events for Unapproved Updates

Follow this configuration for each of the active components on your system.

1. Create derived events to capture the applicable postActions, as defined in the **Derived Events** topic of the **System Setup** documentation.

   For example, the following derived event names are shown in the following images and would be required by a system with PDX and Sufficiency Panel components activated:

   - PDXDerivedEvent
   - SufficiencyDerivedEvent

2. Create a JavaScript business action for each of the applicable activated components, as defined in the **Creating a Business Rule or Library** topic of the **Business Rules** documentation.

Important: While not advised, the following recommended IDs can be changed, but they must be defined in the sharedconfig.properties file as shown in the **Overwriting postAction Business Action IDs** section below.

- PDX - use the ID **pdsPostActionID**
- Sufficiency - use the ID **sufficiencyPostActionID**

3. Edit the business actions created above with the following information, as defined in the **Editing a Business Rule** topic of the **Business Rules** documentation.

   Note: See the **Sample Configurations** section below for relationships between the derived events, event processors, and business actions for each of the activated components.

   - On the Business Rule Editor dialog, if necessary, set **Valid Object Types** to restrict the object types that will trigger the derived event.
   - Click the **Add new Business Action** link and select the **Execute JavaScript** operation from the dropdown.
   - On the Edit Operation dialog, add the **Event Queue** bind to identify the event processor and its queue.
   - Set the **Derived Event Type** bind to identify the derived event created in the first step.
   - Set the **Current Object** bind to take action on the object being handled by the business action.
   - Add the **JavaScript** code to send the updated unapproved data, for example:

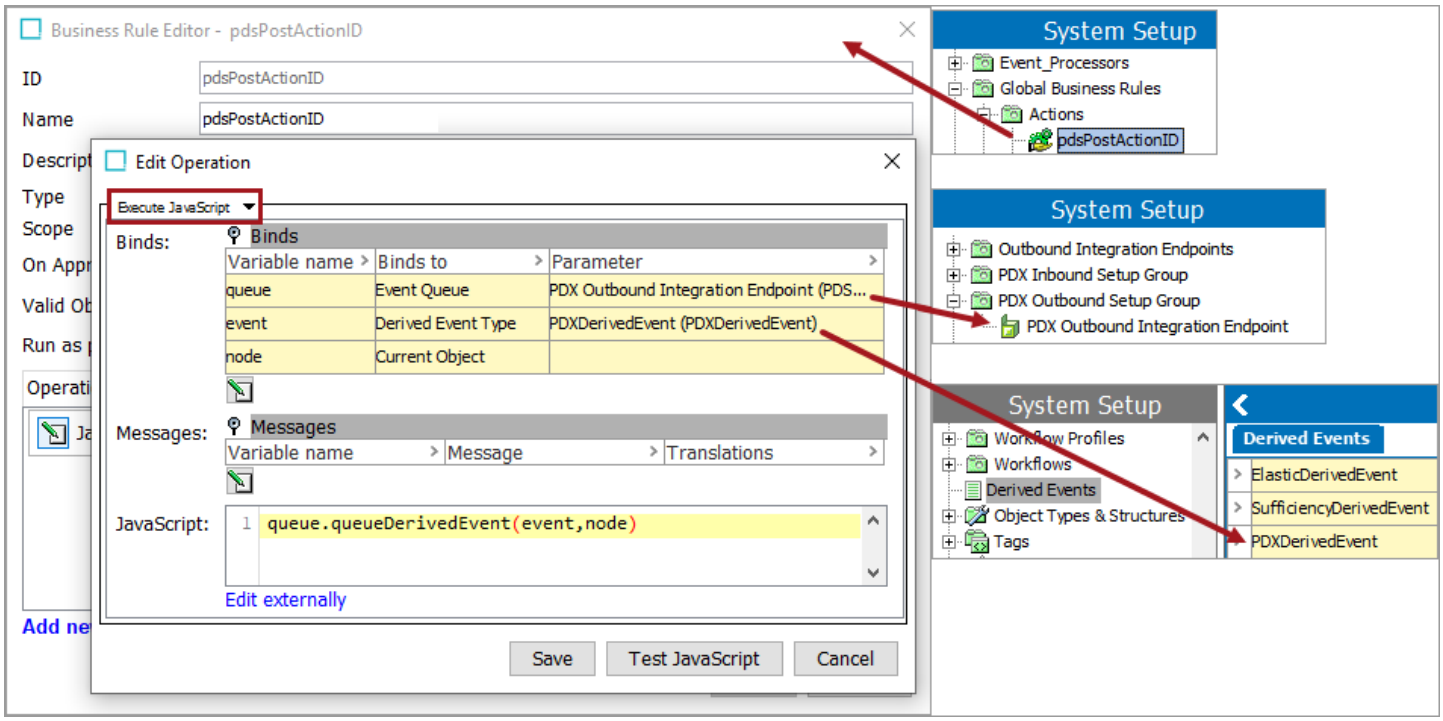
     ```
     queue.queueDerivedEvent(event,node)
     ```

Once this configuration is complete and the defined event processor is enabled and running, clicking the **Save** button in the Product Editor Web UI screen triggers the derived events, which are handled by the defined event processor.

Sample Configurations

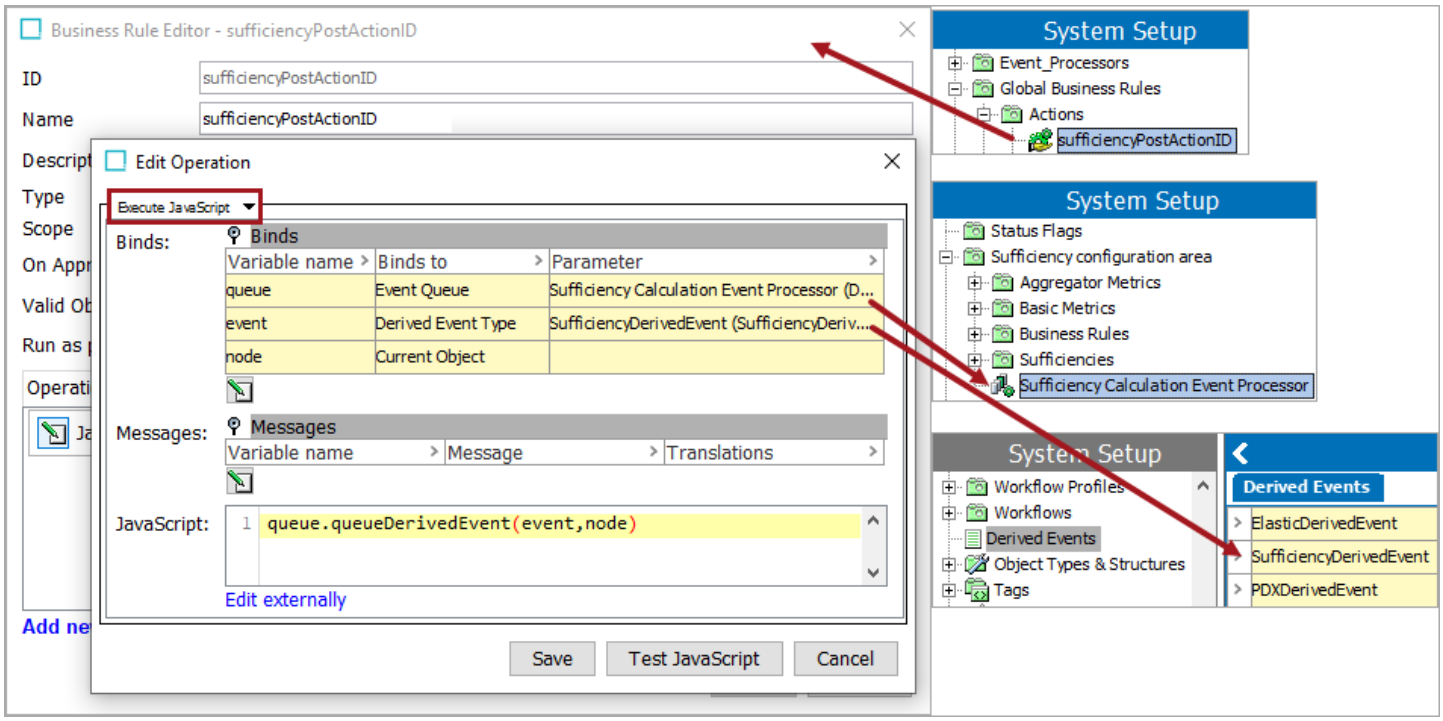
For PDX, the sample below:

- shows the business action with ID 'pdsPostActionID'
- identifies the OIEP with ID 'PDSOutboundIntegrationEndpoint' (including selection of the PDXDerivedEvent in the Output Template Object-EventTypes parameter, as well as the Main workspace (Approved is selected by default))
- uses the derived event with ID 'PDXDerivedEvent'



For Sufficiency Panel, the sample below:

- shows the business action with ID 'sufficiencyPostActionID'
- identifies the event processor with ID 'SufficiencyCalculationEventProcessor' (running the Data Sufficiency Calculator processor)
- uses the derived event with ID 'SufficiencyDerivedEvent'



Overwriting postAction Business Action IDs

By default, postActions are identified using the IDs detailed in the configuration steps above. Overwriting these IDs is not necessary, nor is it recommended, but it is possible using the following properties.

Important: Choosing to overwrite the postAction IDs requires that you use the override ID for the business actions created to generate the derived events.

In the sharedconfig.properties file on the STEP application server, add the following the case-sensitive entries:

- PDS.UpdatePostActionID

```
PDS.UpdatePostActionID=modifiedPDSpostActionID
```

- Sufficiency.UpdatePostActionID

```
Sufficiency.UpdatePostActionID=modifiedSUFFpostActionID
```

Changes to the properties file, outlined above, are implemented when the server is restarted.

Events Not Triggered

While many actions trigger an event as described in the **Events** topics, there are some exceptions.

For Object Types, creation does not trigger an event.

For the following objects, while creation does trigger an event, the changes listed do not trigger an event.

Object	Changes to these items do not trigger an event
Attribute	• Description flipper > Calculated parameter • Description flipper > Type parameter • Description flipper > Dimension Dependencies parameter • Description flipper > metadata parameters defined by system users • Attribute Validation flipper > Validation Base Type parameter • Attribute Validation flipper > List of Values (LOVs) parameter
Business Rules	• Business Rule flipper > Valid Object Types parameter • Business Rule flipper > On Approve parameter • Business Rule flipper > Run as Privileged parameter
LOVs (Lists of Values)	• Description flipper > Use IDs for Sorting parameter
Reference Types	• Type of Reference • Description flipper > Inheritance parameter • Description flipper > Mandatory parameter
Web UIs	• Description flipper > ID parameter • Description flipper > Name parameter • Description flipper > Default parameter

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** topic.
2. Launch the event processor wizard as described in **Creating an Event Processor** topic.
3. In the wizard, add an ID, name, and object type as described in **EPW - Identify Event Processor** topic.
4. In the wizard, specify the processing and error handling as described in **EPW - Configure Event Processor** topic.
5. In the wizard, specify the information specific to the processing plugin selected in the previous step as described in **EPW - Configure Processing Plugin** topic.
6. In the wizard, specify how often the event processor should start as described in **EPW - Schedule Event Processor** topic.
7. In the wizard, populate the send report to address parameter as described in **EPW - Configure Error Reporter Processing Plugin** topic.
8. Confirm / Update triggering definitions as described in **Event Triggering Definitions** topic.
9. Set Queue Status Events to 'Read Events' as described in the **Queue Status** section of the **EP - Event Processor Tab** topic.
10. Enable the event processor as described in the **Enable Event Processor** section of the **Running an Event Processor** topic.

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. For information on the recommended practices for Error Handling, see the **Recommended Error Handling Practices** topic in **Resource Materials** 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.

Note: Some processing plugins may not be available without enablement of a corresponding add-on component and/or license. Contact your account manager or partner manager to begin the process of enabling a license or licenses for your system. Instructions for installing components can be found in the **SPOT Program** topic in the **System Administration** documentation.

Processing Plugin Name	Description
Asset Publisher	Offers many of the same options as traditional Asset Push functionality and can be configured to publish asset content directly to Azure Blob Storage / S3. See the **Asset Publisher Processing Plugin Parameters and Triggers** documentation. Also, refer to the Asset Push documentation within the Digital Assets documentation.
Asynchronous Translation Message Processor	Allows users to send scheduled translation jobs to asynchronous translation services based on a derived event, and depending on user preferences, batch translation events into as little as one translation job, reducing translation costs. See the **Asynchronous Translation Message Processor Processing Plugin Parameters and Triggers** documentation.
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. See the **CASS Certification Report Processing Plugin Parameters and Triggers** documentation.
Data Sufficiency Calculator	Works in tandem with the sufficiency configuration type to calculate sufficiency

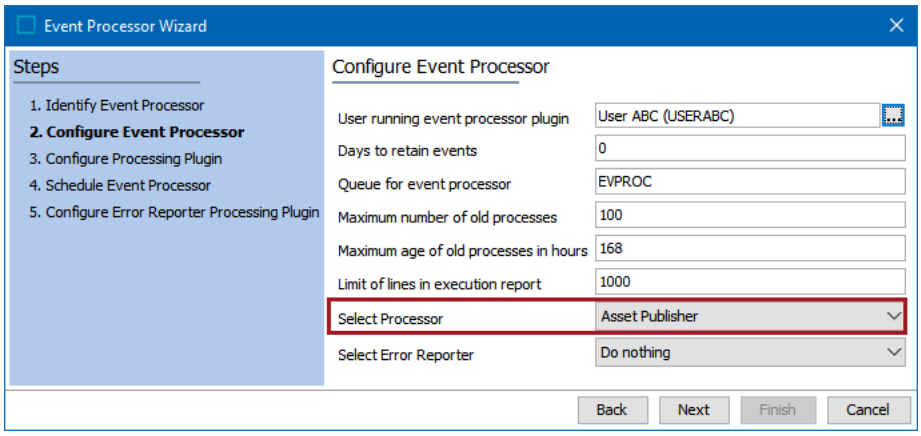
Processing Plugin Name	Description
	scores for a given product. Assesses all the sufficiency configuration types within a system, and calculates the sufficiency scores for the products included in the 'Triggering Object Types' parameter in the 'Event Triggering Definitions' tab of the event processor. See the **Data Sufficiency Calculator Processing Plugin Parameters and Triggers** documentation.
Elasticsearch Indexer	Allows STEP to publish data to Elasticsearch for a faceted search experience. See the **Elasticsearch Indexer 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.
Execute Business Action for Event Batch	Allows a business action to be run from an event processor, so that when changes occur, a custom business action can be asynchronously triggered. The configured business action executes once per event batch (instead of once per event like the Execute Business Action). See **Execute Business Action for Event Batch 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**.
Historic Data Analyzer	Allows for the analysis for data for historic reporting. The event processor (IdOfReport_RA) is created automatically when starting the report. See the **Historic Reporting** documentation in **Resource Materials**.
Historic Data Generator	Allows for the generation of historic reports. The event processor (IdOfReport_RG) is created automatically upon generation of a historic report. See the **Historic Reporting** documentation in **Resource Materials**.

Processing Plugin Name	Description
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.
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.
Source Record Data Management – Historical Values Cleanup	Allows incremental purging of the oldest historical source data and optimizing storage to keep disk usage as low as possible. Storage of source data is only supported on object types identified by the 'Matching - Merge Golden Records' component model. See **Source Record Data Management Processing Plugin Parameters and Triggers** documentation.
Splitting Processor	Designed for historical reporting. The functionality is in ramp-up status, and while accessible, may not work for you as expected since it is still under development. Historical reporting is traditionally extremely slow, and there is a need to distribute events in a controlled manner. The Splitting Processor takes a number of queues from other event processors or outbound integration endpoints, and either copies the entire batch of events to all of them or distributes the events evenly between the queues. See **Splitting Processor Processing Plugin Parameters and Triggers** documentation.

Asset Publisher Processing Plugin Parameters and Triggers

The Asset Publisher event processor plugin can be configured to publish asset content to a specific location.

To access the 'Configure Processing Plugin' parameters as shown below, the **Asset Publisher** processor must be selected within the Select Processor parameter during the wizard step 'Configure Event Processor.'



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.

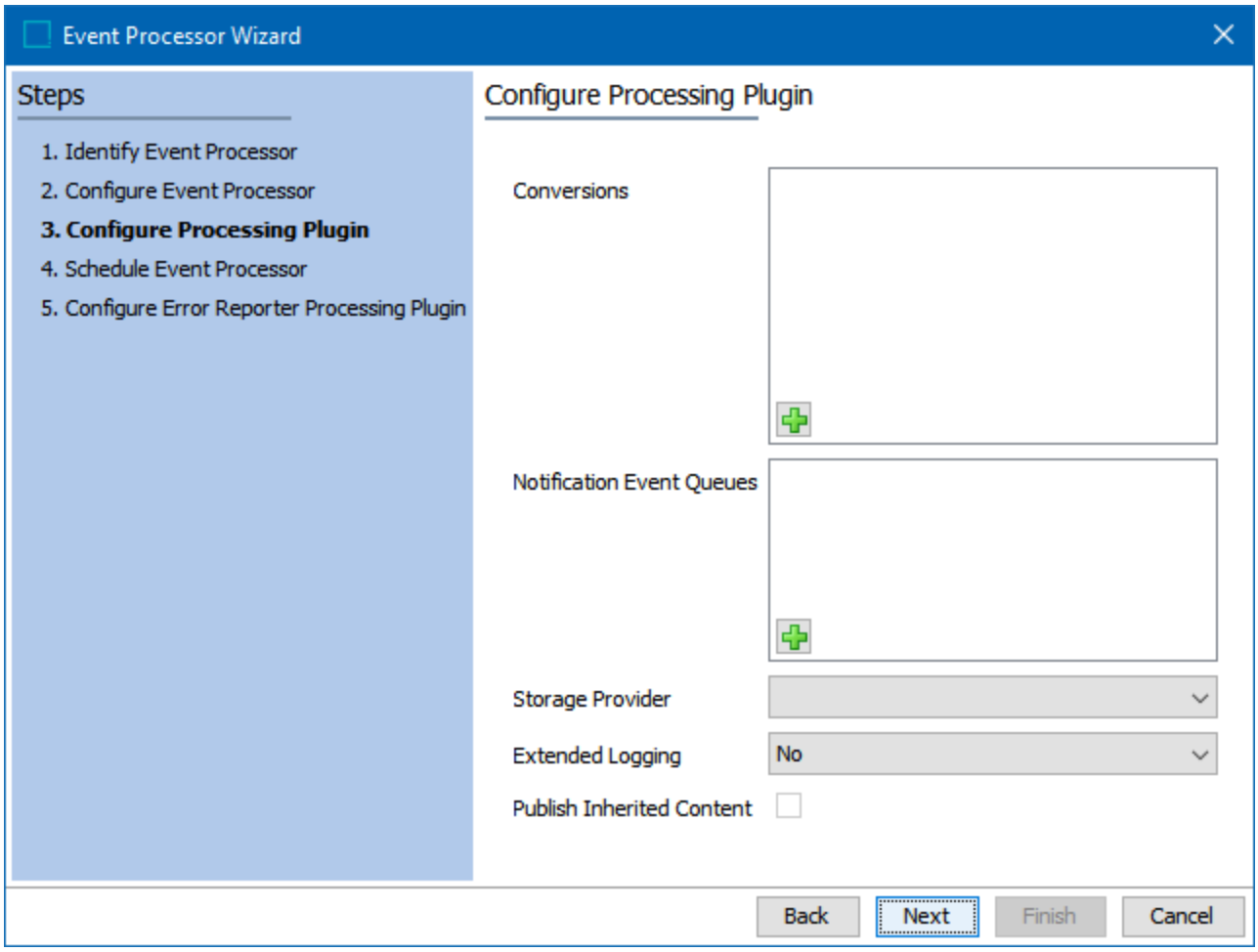
Prerequisites

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.

Additionally, if image conversion is required as part of the asset push, the image conversion configuration must be completed prior to the asset push configuration, so that the conversion can be selected as part of the configuration. See the **Image Conversion Configuration** topic for more information.

Parameters

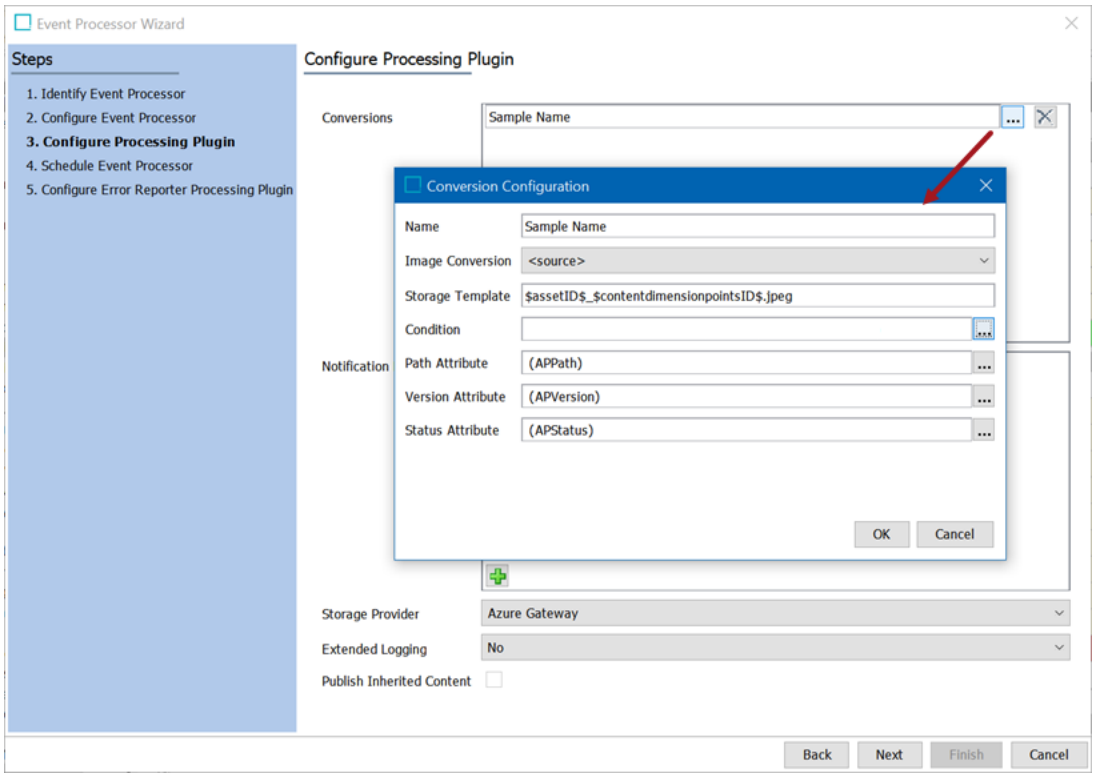
Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.



Note: You can hover over the parameter value fields within the wizard to access descriptive tool tip text.

Conversions: These are the image conversions to publish. Click the plus button in the left corner of the parameter field. A row will be added within the field; click the ellipsis button (...) to display the Conversion Configuration dialog.

The Conversion Configuration dialog has the following fields with the properties listed below:



- **Name**: This is the name of the conversion.
  - Mandatory field
  - Must be unique within an Asset Publisher configuration
- **Image Conversion**: Select the image conversion to be used.
  - Mandatory field
  - If the assets need their format, size, or image color settings converted when pushed to the target system, select the relevant conversion configuration from the list. The <source> option is default and will produce no conversions. Using the dropdown, you will see a list that contains all of the conversions that are available in the system. Configurations that have (conversion) appended to their names are custom conversions and can be changed and/or edited. Those without (conversion) appended are standard conversions and cannot be changed and/or edited.
- **Storage Template**: Specify the path and file name to store the conversion on blob storage. This is a mandatory field. The following variables / macros are available:
  - **$assetID$** - ID of the asset being published
  - **$IDpath$** - Two path levels constructed from the last four (4) characters of the asset ID; use when folder structure for assets is needed
  - **$contentdimensionpointsID$** - ID(s) of the dimension point(s) for which the asset is being published
  - **$extension$** - The original extension for the asset. Will not include leading dot (.)
  - **$autodetected-extension$** - The extension produced by the asset conversion. Should only be used for the rate case where the conversion can produce files with different extensions. If a conversion

always produces the same extension, the information should be hardcoded instead. Will not include leading dot (.)

> **Note:** For the storage file path template, similar to Asset Push, the system lowercases and replaces characters that are illegal in a Windows file system with '_'). As an example: If assetid is 'ABC<> and storage template is '$IDpath$/$assetID$.jpg', then it will resolve to the storage path 'bc/__/ABC__.jpg'. Illegal Windows characters are: \ / : * ? < > |

Validation is done when saving the configuration. You can choose to fix the issue at that point (if one exists) or save the configuration as-is and update at a later time.

- **Condition**: If specified, then this condition has to be evaluated to be true for the given asset to be published by this conversion.
  - Optional
  - You have the ability to bind asset content to restrict that asset conversion only be done on, for example, specific MIME types by using the 'Current Asset Content Object' bind as described in the **Other Binds** topic in the **Resource Materials** documentation. If you want to restrict conversion only on assets in specific classifications, then this should be a condition defined in the triggering condition on the queue (where it will be the asset itself that can be bound in the condition as opposed to asset-content).
- **Path Attribute**: The Path Attribute is used to store the actual published path of this asset (from the storage template).
  - Optional
  - Single-valued
  - Externally maintained
  - Must have same dimension-dependencies as asset content
  - Must be valid for an object type used in triggering-object-types of event processor and the path attribute must not be used in any other enabled Asset Publisher event processor
- **Version Attribute**: This attribute is used to store the version of the published asset. The value is stored on the asset.
  - Mandatory field
  - Single-valued
  - Externally maintained
  - Must have same dimension-dependencies as asset content
  - Must be valid for an object type used in triggering-object-types of event processor and the path attribute must not be used in any other enabled Asset Publisher event processor
- **Status Attribute**: If specified, then this attribute is used to store the status of the publish operation. The value is stored on the asset.
  - Optional
  - Single-valued
  - Externally maintained
  - Must have same dimension-dependencies as asset content

- Must be valid for an object type used in triggering-object-types of event processor and the path attribute must not be used in any other enabled Asset Publisher event processor

The constraints detailed above are checked when the event processor is processing events. If the constraints are not fulfilled, the event processor will stop and fail.

Notification Event Queues: Click the plus button in the left corner of the parameter field. A row will be added within the field; click the ellipsis button (...) to display the Select Processor for Events dialog. Select the processor using standard Browse / Search Node Picker functionality.

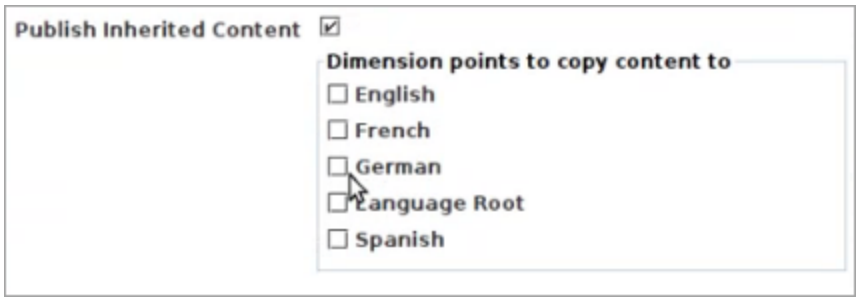
Storage Provider: Storage provider to push converted images to. If there are no dropdown options, then the Gateway Integration Endpoints either do not exist or have not been configured.

For specific information on how to set up Gateway Integration Endpoints for the Asset Publisher functionality, see the following topics located in the **Data Exchange** documentation:

- **Configuring a Gateway Integration Endpoint - Amazon S3 Blob Storage**
- **Configuring a Gateway Integration Endpoint - Microsoft Azure Blob Storage**

Extended Logging: Change to 'Yes' to enable extended logging in the execution report of the event processor. This will log information for each upload taking places. If you do not want this logging, set to 'No' (the default setting).

Publish Inherited Content: This option is available on systems where asset content is dimension dependent and the content depends on a single dimension. By default, asset content is only published for the dimension points it is local to. Using this option, it is possible to have content also exported for specific dimension points that the content can be inherited to.



Event Triggers

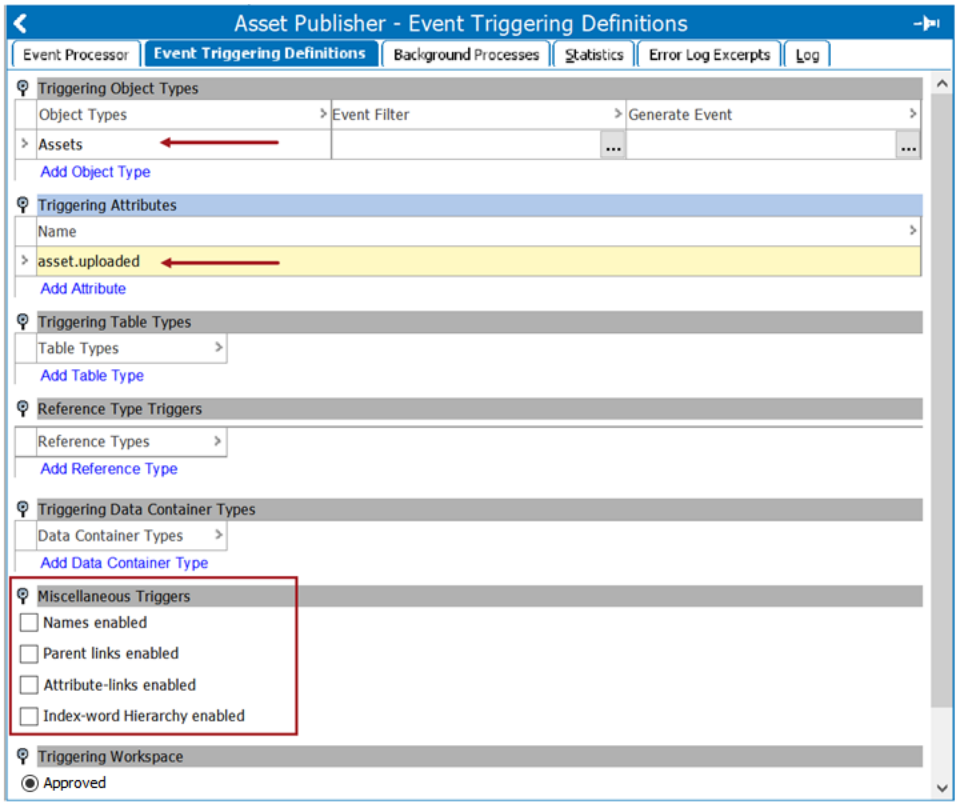
For the event processor to take effect, it must be configured to only listen for changes on asset type objects housed within the same previously-selected classification folder.

The recommended setup of the triggering definition of the Asset Publisher event processor is to:

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.

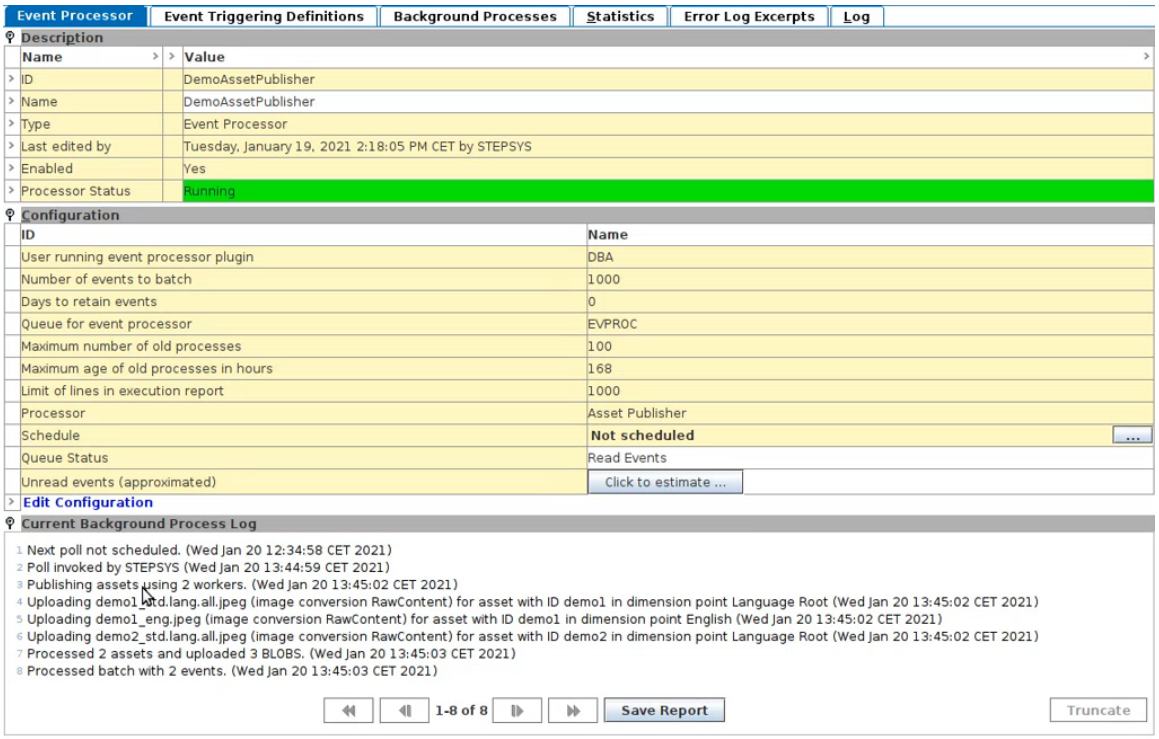
Important: You must select at least one asset type for Triggering Object Types or nothing will be processed by the event processor.

3. Click the **OK** button to add the selection to the Triggering Object Types table.
4. Trigger on attribute 'asset.uploaded.'
5. Uncheck all the triggers at the end, e.g., name, parent links, etc.
6. Verify that Approved is enabled under Triggering Workspace. Publishing can only be done from the Approved workspace.

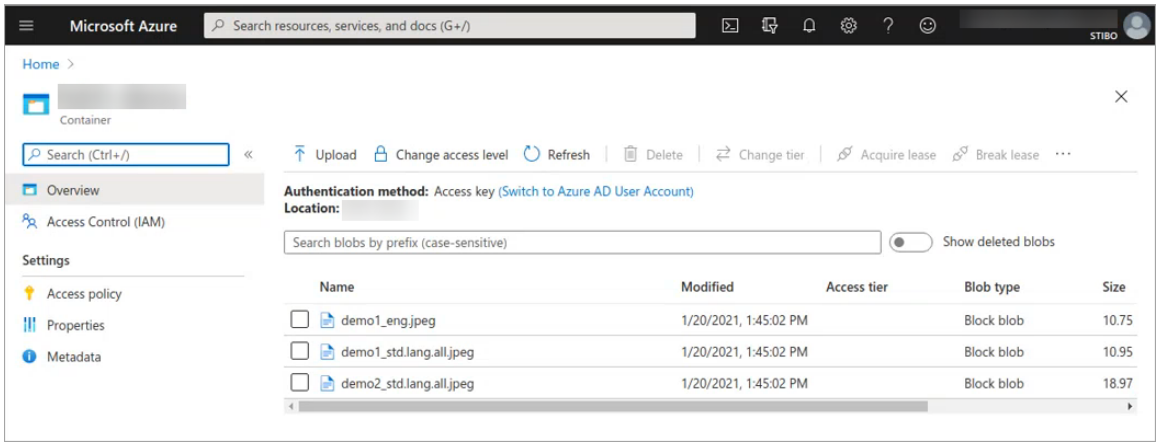


When the event processor runs, a configuration check is done. Also, the attributes that you set for path and version should have the same dimension dependencies as asset content. Upon validation, if anything is wrong, the event processor will stop. You will then need to reconfigure and run again.

Below is an example of an event processor that processed successfully. It processed two (2) assets and uploaded three (3) blobs.



Continuing the example above, you will see the three blobs in blob storage:



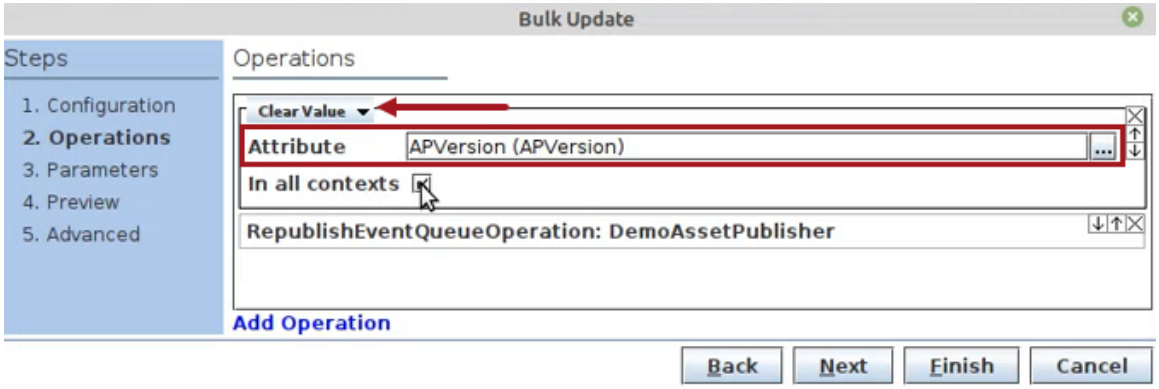
You can return to the asset(s) published and see the updates for the path, status, and version. Your results will depend on your own setup and upload time.

APPath	abc	demo1_std.lang.all.jpeg
APStatus	abc	published
APVersion	abc	2021-01-19 13:30:28

Republishing Assets

If you want to republish these assets (e.g., they were deleted from blob storage, they were resized, the path changed, etc.), then you will run a bulk update using the Clear Value operation to clear the value for a specific attribute. Or, you can run more than one Clear Value operation to clear the value for multiple attributes. See the **Attribute Values: Clear Value Operations** topic within the **Bulk Updates** section for how to use this operation.

You will then need to send a republish event to the asset(s). You can then reinvoke the event processor and republish the assets to blob storage.



Asynchronous Translation Message Processor Processing Plugin Parameters and Triggers

The Asynchronous Translation Message Processor plugin allows users to send scheduled translation jobs to asynchronous translation services based on a derived event, and depending on user preferences, batch translation events into as little as one translation job, reducing translation costs.

For information regarding sending a scheduled translation job to an asynchronous translation service, see the **Scheduling a Collection of Objects for Translation** topic in the **Translation** documentation.

Note: The Asynchronous Translation Message Processor processing plugin is only available to users that have the external-async-kernel component installed in their system. For more information, contact your Stibo Systems representative.

Prerequisites

This document 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.

In addition to the configuration of the event processor, the following items need to be configured:

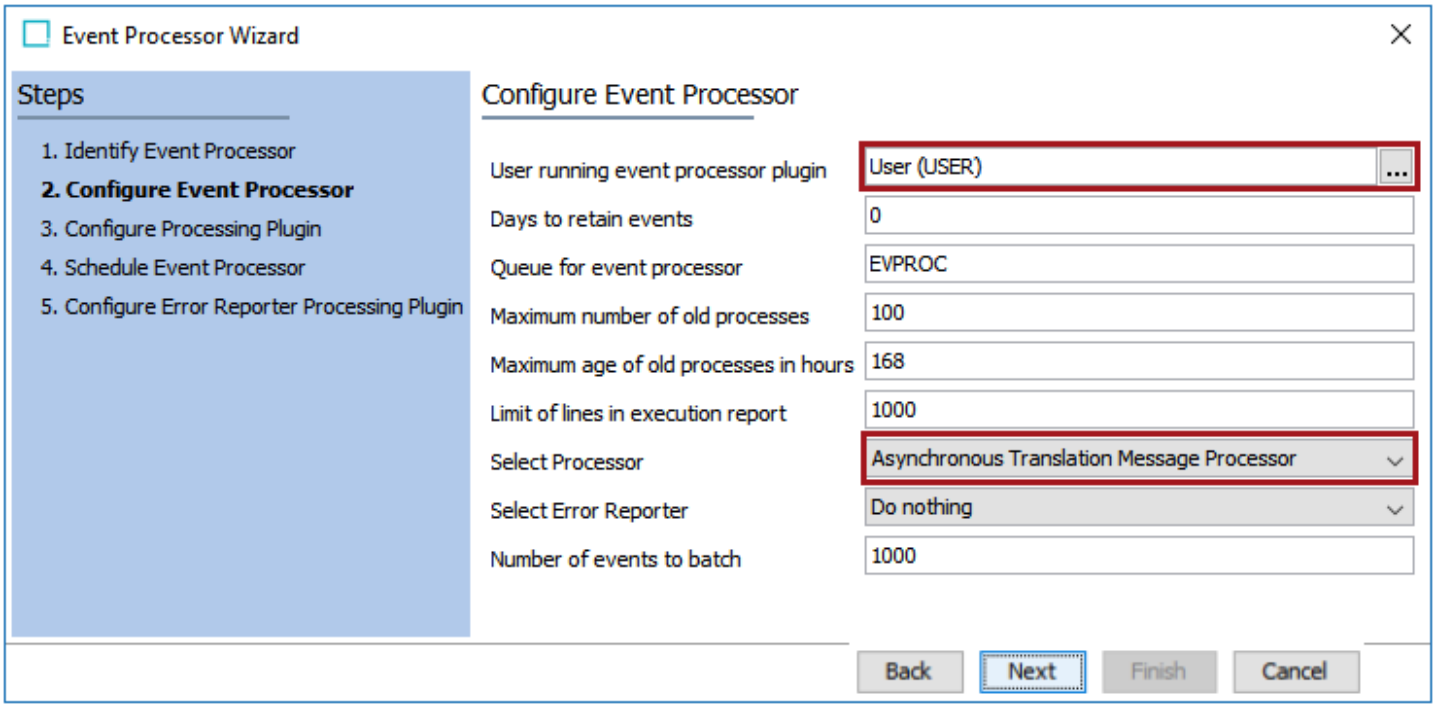
- At least one asynchronous translation service, with at least one translation configuration as part of that service
- A derived event

For more information regarding asynchronous translation services, see the **Asynchronous Translations** topic in the **Data Integration** documentation.

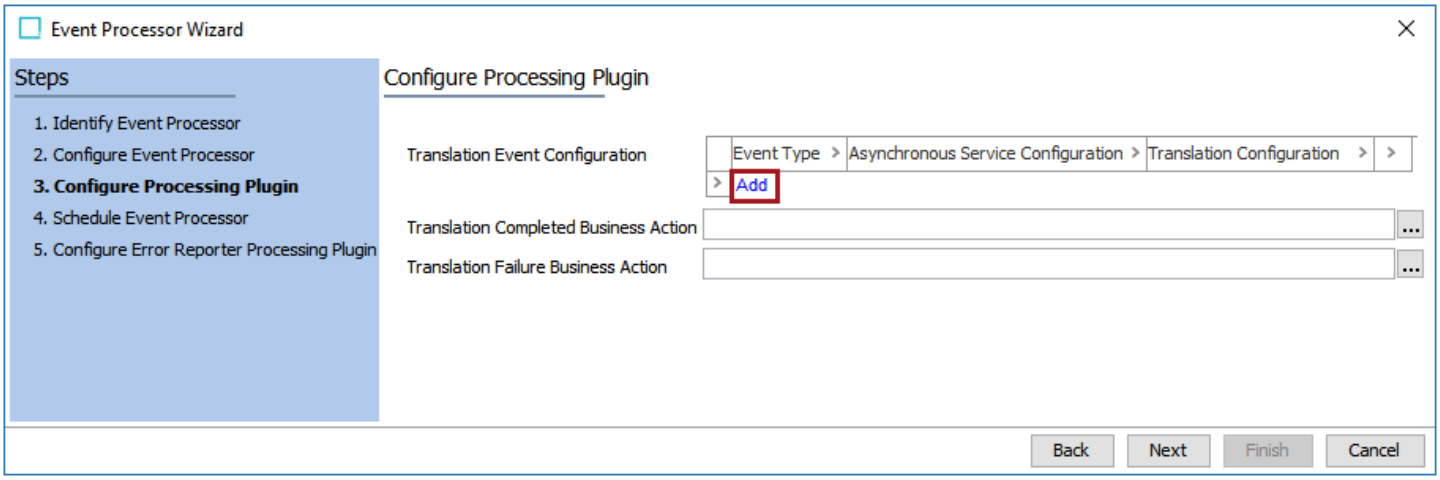
For more information regarding derived events, see the **Derived Events** topic in the **Events** documentation.

Parameters

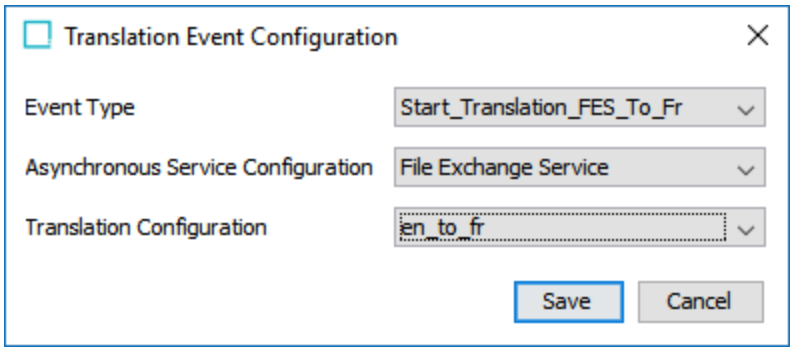
Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.



1. For the **User running event processor plugin** parameter, recommended practice is to create a user specifically for this event processor with adequate privileges to access the data that will be published.
2. For the **Select Processor** parameter, choose the **Asynchronous Translation Message Processor** processing plugin to display the expected parameters on the Configure Processing Plugin step.
3. Click the **Next** button to display the wizard step 'Configure Processing Plugin'.



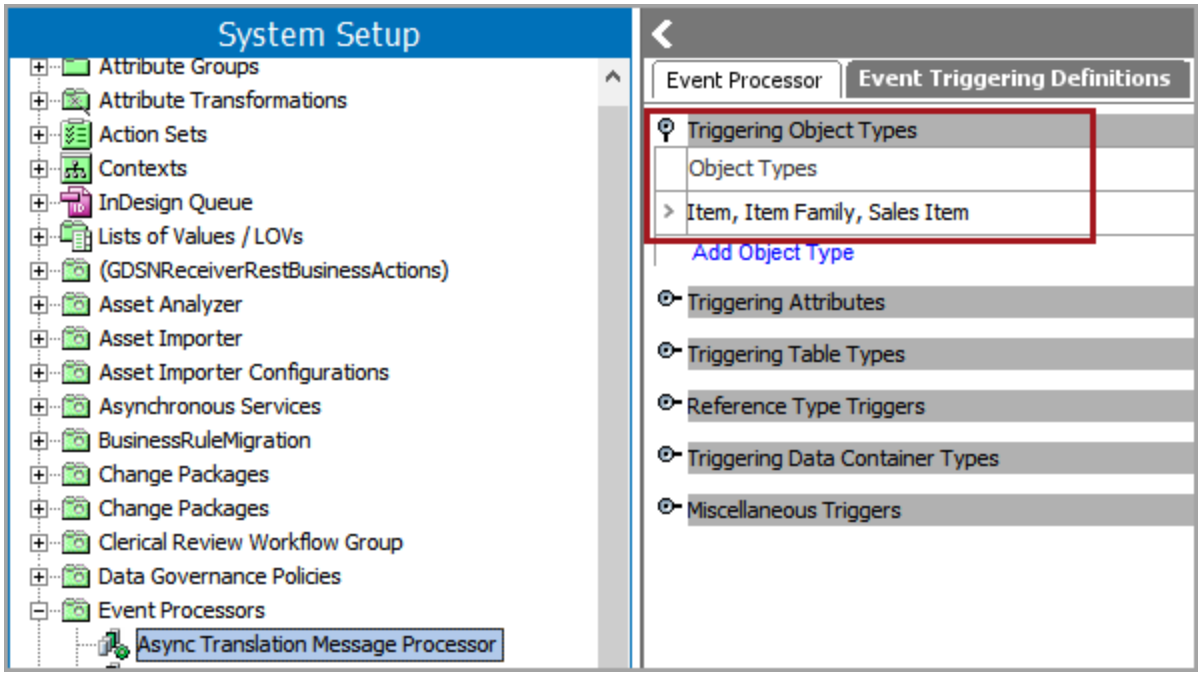
4. Click **Add**. The Translation Event Configuration window opens.



5. For the **Event Type** parameter, select the derived event you would like to use for the scheduled translation. In this example, the user has selected 'Start_Translation_FES_To_Fr.'
6. For the **Asynchronous Service Configuration** parameter, select the asynchronous service configuration that you would like to use for the scheduled translation. In this example, the user has selected 'File Exchange Service.'
7. The **Translation Configuration** parameter contains selection options based on the asynchronous service configuration that is selected for the Asynchronous Service Configuration parameter. In this example, the user has selected 'en_to_fr.'

Event Triggers

When configuring the Asynchronous Translation Message event processor, all item object types that are collected for a translation must be included in the 'Triggering Object Types' parameter on the 'Event Triggering Definitions' tab within the event processor. The image below demonstrates an example of triggering object types found in a collection created for translation purposes.



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.

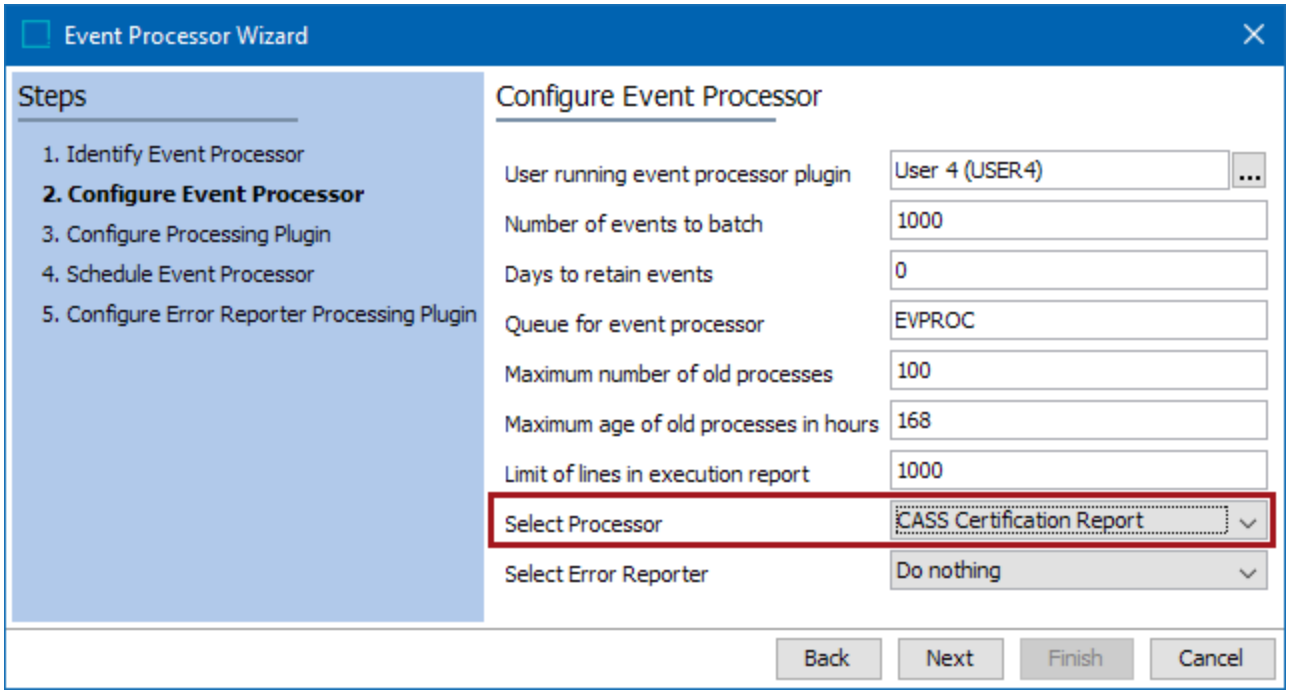
- Users must be based in the United States, as CASS is not valid outside of the US.
- Users must be connected to a Loqate Local server installation and have purchased a CASS license. The CASS components will not work with a Loqate Cloud API installation. For more information, see the **Loqate Integration** 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 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.

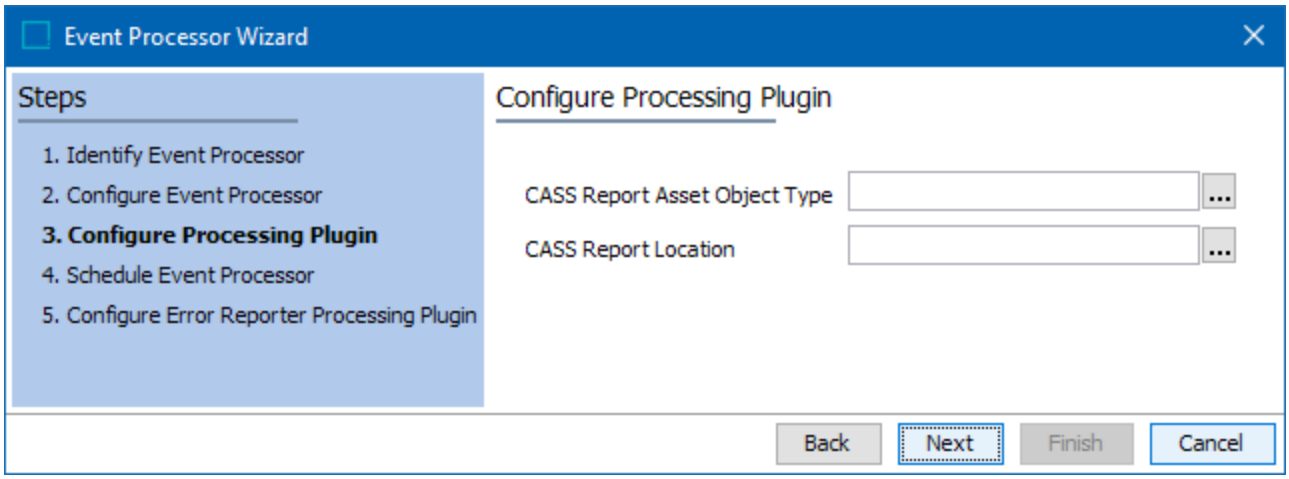
Parameters

Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.

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.'



Once the CASS Certification Report processing plugin has been selected, click the **Next** button to display the 'Configure Processing Plugin' wizard screen.



- **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.

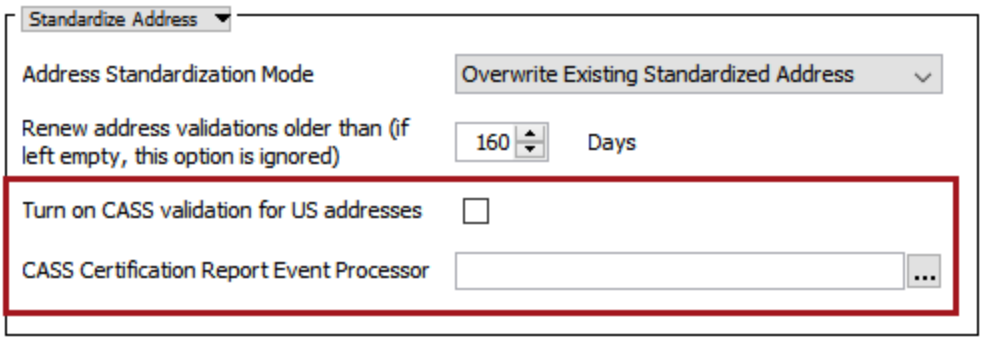
Once any required configurations are made, click the **Next** button to display the wizard step 'Schedule Event Processor.'

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: Standardize Address Operation** 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** section of the **EP - Event Processor Tab** topic.

Data Sufficiency Calculator Processing Plugin Parameters and Triggers

The Data Sufficiency Calculator processing plugin works in tandem with the Sufficiency Configuration Type to calculate sufficiency scores for a given product. When run, the Data Sufficiency Calculator Processing plugin assesses all the sufficiency configuration types within a system, and calculates the sufficiency scores for the products included in the 'Triggering Object Types' parameter in the 'Event Triggering Definitions' tab of the event processor. These calculated scores are then output into data containers on the evaluated products. The sufficiency scores can be viewed in the Sufficiency Inspector panel in the Web UI and data containers within the Web UI for applicable products.

Note: In order to access the Data Sufficiency Calculator Processor plugin, a 'data-sufficiency' add-on component must be activated on your system. See your Stibo Systems representative for more information.

For more information regarding the sufficiency configuration type, see the **Sufficiency Configuration Type** topic in the **Data Governance** documentation.

For more information on the sufficiency inspector in the Web UI, see the **Sufficiency Panel** topic in the **Data Governance** documentation.

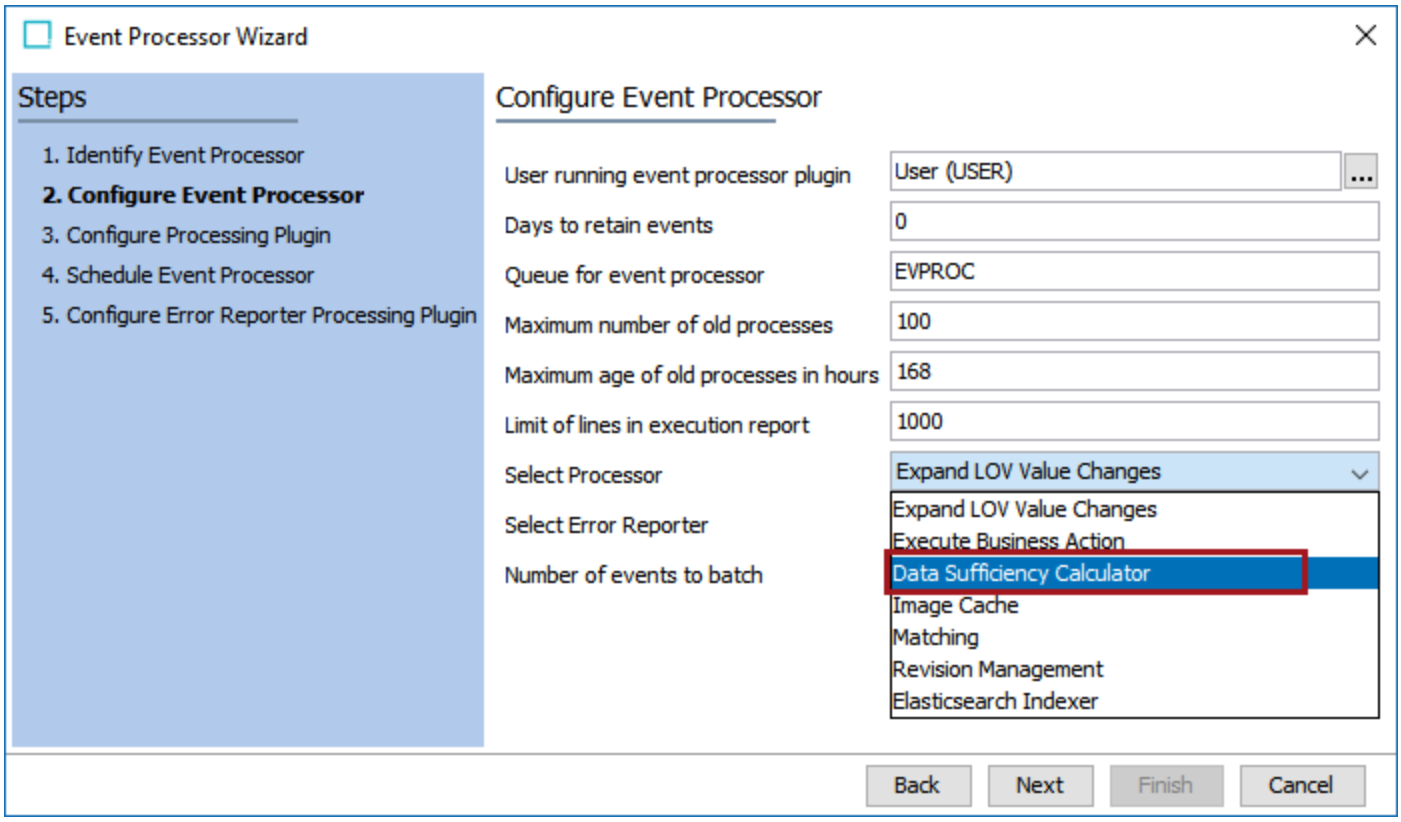
Prerequisites

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, including event triggering definitions, see the **Event Processors** documentation.

If you want the Data Sufficiency Calculator event processor plugin to utilize multiple threads for calculating data sufficiency metrics, the number of threads to use can be configured via the configuration property `BackgroundProcess.Queue.<QUEUE_NAME>.Parallel` (defaults to 2).

Parameters

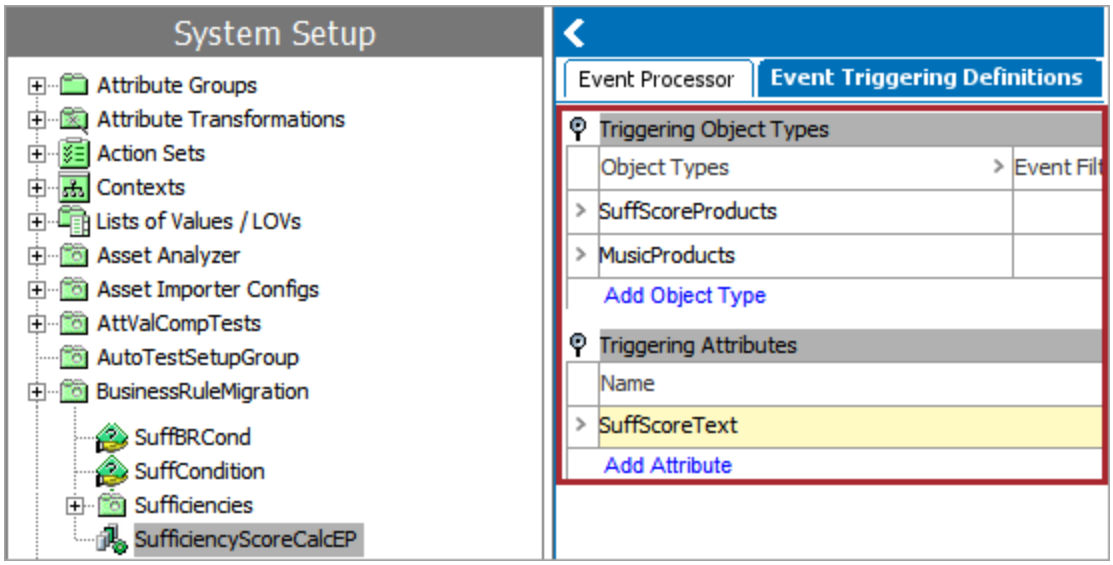
Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.



1. For the **User running event processor plugin** parameter, recommended practice is to create a user specifically for this event processor with adequate privileges to access the data that will be published.
2. For the **Select Processor** parameter, choose the **Data Sufficiency Calculator** processing plugin to display the expected parameters on the Configure Processing Plugin step.

Event Triggers

The 'Triggering Object Types' parameter, located within the Event Triggering Definitions tab, must include the object types and the attributes included in the metrics that are part of the sufficiencies being run to evaluate data quality and completeness. In the example below, the object types 'SuffScoreProducts' and 'MusicProducts' will be evaluated for quality and completeness when the event processor is run. 'SuffScoreText' is the triggering attribute for the event processor.



Note: When configuring the Data Sufficiency Calculator event processor, attributes included in the 'Data Sufficiency Attribute Group' attribute group (addressed later in this topic) should not be added as triggering attributes; additionally, data containers within the 'Data Sufficiency Attribute Group' attribute group should not be added as triggering data container types. Adding either of these as triggers for the event processor can produce unintended events. For more information regarding the Data Sufficiency Attribute Group, see the **Sufficiency Scores for Product Data Quality and Completeness** topic.

Elasticsearch Indexer Processing Plugin Parameters and Triggers

Elasticsearch Indexer processes core and derived events to trigger the publication of STEP data to the Elasticsearch search engine for display on the faceted Search Screen in Web UI.

Important: To access and use faceted search powered by Elasticsearch, the ‘elasticsearch-integration’ add-on component must be activated on your system. To access and use the faceted Search screen, the 'ui-search' add-on component must be activated on your system. See your Stibo Systems representative for more information.

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.

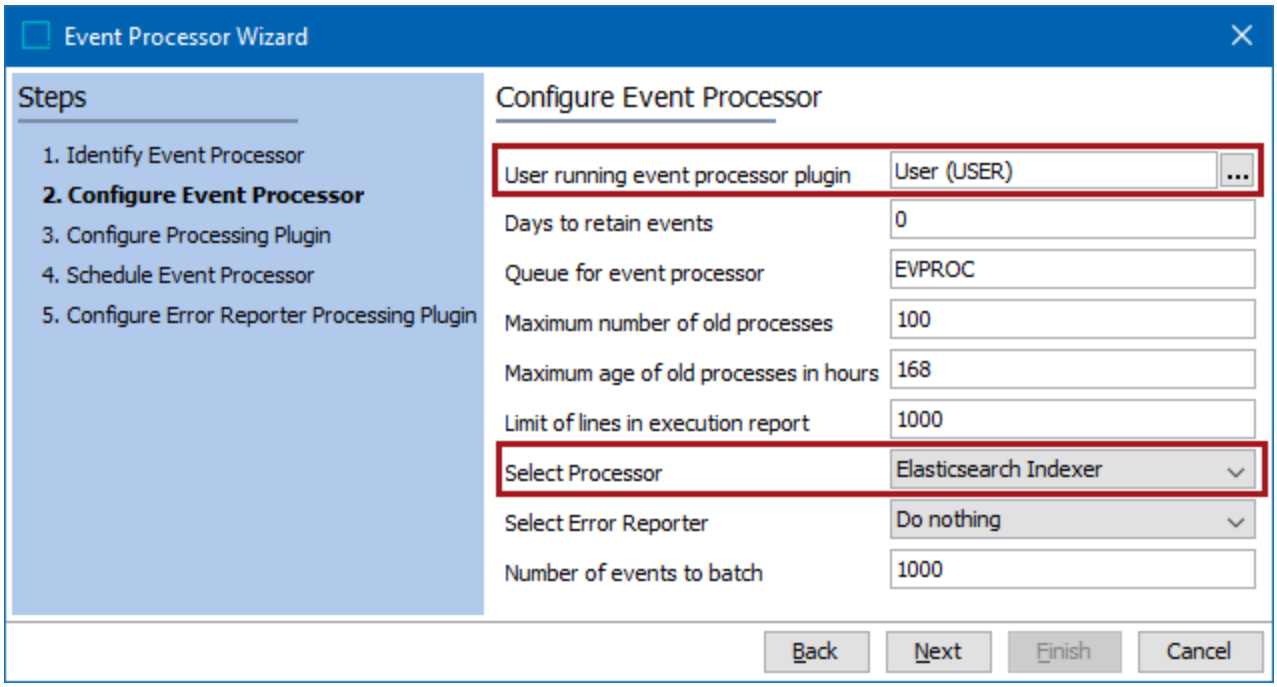
Prerequisites

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 Elasticsearch Indexer requires an Elasticsearch configuration created via a wizard (not manually). For details, see the **Creating an Elasticsearch Configuration** topic in the **Web User Interfaces / Web UI Setup and User Guide** documentation.

Parameters

Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.



While the event processor is created automatically by clicking the Finish button on the Elasticsearch Configuration wizard, the following parameters can be edited directly on the event processor.

1. On the 'Configure Event Processor' step, for the **User running event processor plugin** parameter, recommended practice is to create a user specifically for this event processor with adequate privileges to access the data that will be published.
2. For the **Select Processor** parameter, choose the **Elasticsearch Indexer** processing plugin to display the expected parameters on the 'Configure Processing Plugin' step.
3. Click the **Next** button twice to display the wizard step 'Schedule Event Processor' and schedule as required.

Other changes to an existing Elasticsearch event processor must be made from the Elasticsearch configuration, as defined in the **Maintaining an Elasticsearch Configuration** topic.

Event Triggers

Basic event triggers are set using the Elasticsearch Configuration wizard. Set additional triggers based on your data model and as identified by the numbered items 1-3 in the image below.

- **Triggering Object Types** - add object type(s) for which instances contain data to be published to Elasticsearch. Additionally, include business rules and/or derived events as necessary to provide inherited or referenced data, as follows:

  Triggering for Inherited Data: When triggering on specific object types, such as 'Product' (item 1), data inherited to the Product object could be changed on the parent. To handle this case, the Product object parents (item 2) are triggered and then processed via a business condition in the Event Filter parameter and a business action in the Generate Event parameter. The condition always returns 'False' since you do not

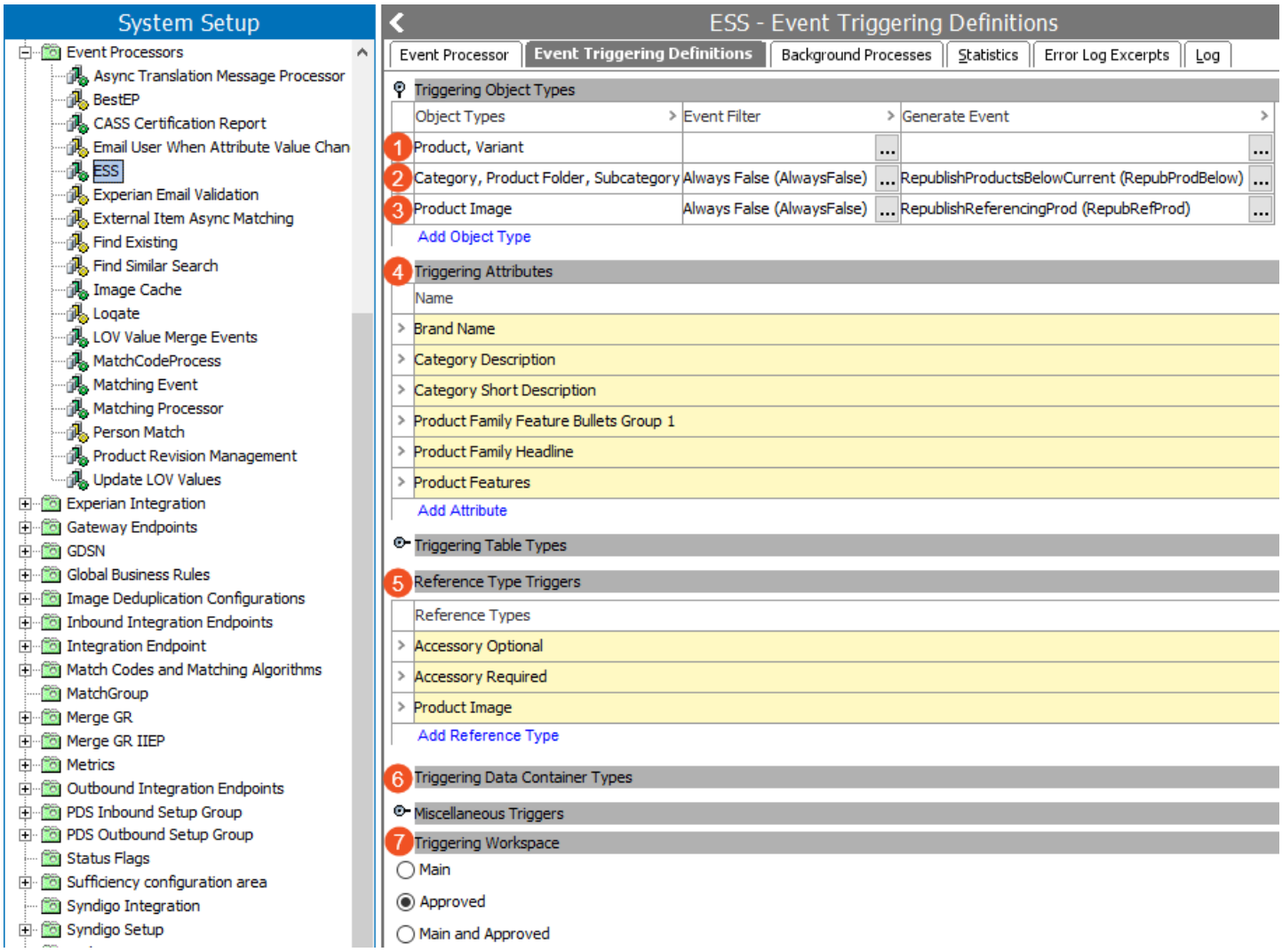
want the parent object types to be published. The action generates a derived event to recursively republish only Product objects for the parent object types that triggered the event processor.

Triggering for Referenced Data: When triggering on specific object types, such as 'Product' (item 2), you can republish the Product object when data changes on an object referencing the Product. To handle this case, the Product object referenced image (item 3) is triggered and then processed via a business condition in the Event Filter parameter and a business action in the Generate Event parameter. The condition always returns 'False' since you do not want the referenced object types to be published. The action generates a derived event to evaluate the incoming references from products array, and if the source of a reference is a Product, creates an event for it. Use this only when the Elasticsearch configuration includes data from referenced objects, for example, when those objects change, you want to republish the 'root node' to publish the data to Elasticsearch.

For similar use cases and samples of the JavaScript business rules, see the **Event-Based OIEP Examples** topic within the **Data Exchange** documentation.

- **Triggering Attributes** - add all attributes (item 4) that should be monitored.
- **Reference Type Triggers** - add reference types (item 5) that should be monitored.
- **Triggering Data Container Types** - data containers (item 6) are not allowed with Elasticsearch. Sufficiency score data container information is automatically included for publication and is not required to be defined as an event trigger.
- **Triggering Workspace** - (item 7) select the workspace(s) to update the Elasticsearch database. Processing each and every event from the maintenance workspace can lead to bottlenecks, and in worst case, to overall performance decrease of the entire system. This option is unique to systems running the 'elasticsearch-integration' add-on component, as defined in the **Data Enrichment Components** topic.

Event Triggering Definitions are the same in Event Processors as they are in an OIEP. For details about the triggers, see the **OIEP - Event-Based - Event Triggering Definitions Tab** in the **Data Exchange** documentation.



Next, publish data to the Event Processor running the Elasticsearch Indexer as defined in the **Elasticsearch Publishing** topic.

Execute Business Action for Event Batch Processing Plugin Parameters and Triggers

The Execute Business Action for Event Batch processing plugin allows the configured business action to be executed once per event batch instead of once per event like when using the Execute Business Action processing plugin. The ability to execute one action for an event batch is especially relevant when communicating with external services that accept batch requests.

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.

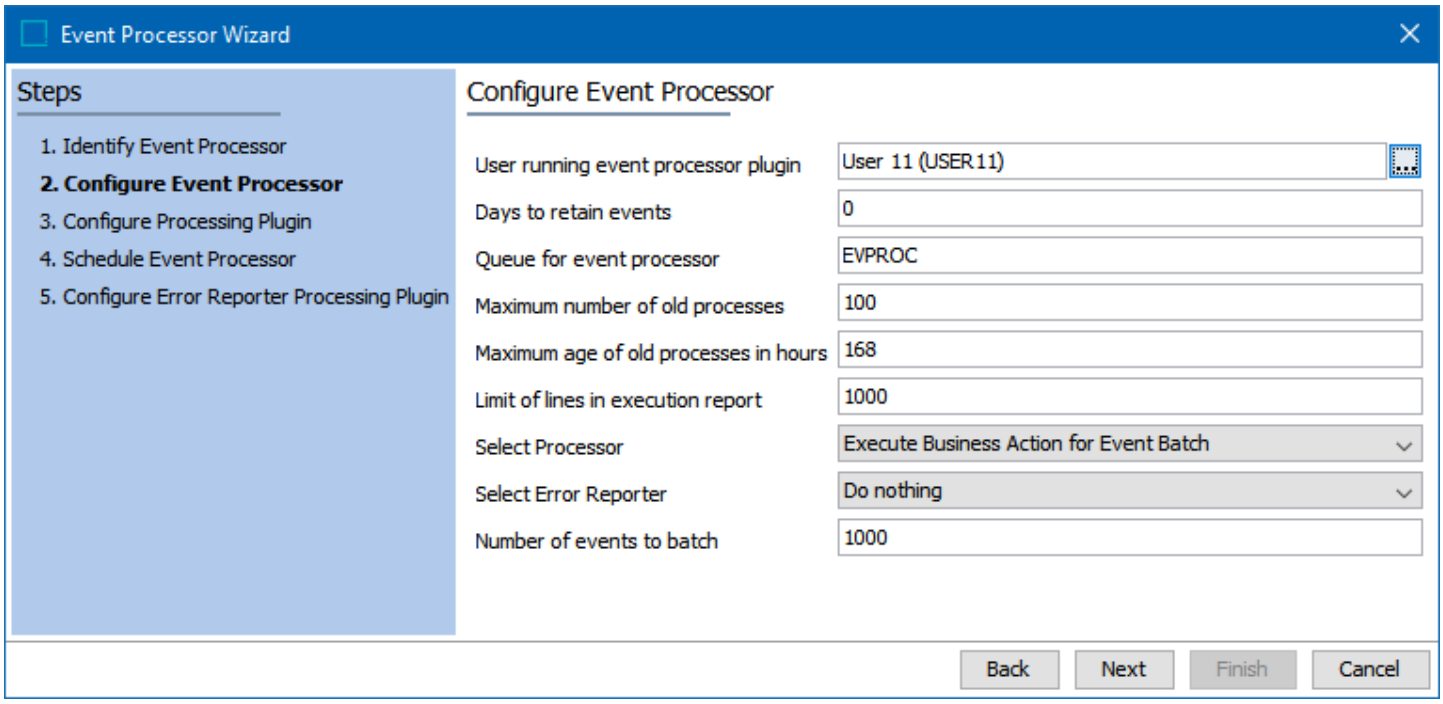
Before configuration of an event processor using the Execute Business Action for Event Batch processor can occur, a business action must be created. For more information, see the **Business Actions** section within the **Business Rules** documentation.

The business action referenced from the Execute Business Action for Event Batch plugin has no concept of 'current object.' Therefore, the JavaScript Current Object bind and most non-JavaScript business action plugins cannot be used. Business actions referenced from the plugin must be made applicable for all object types.

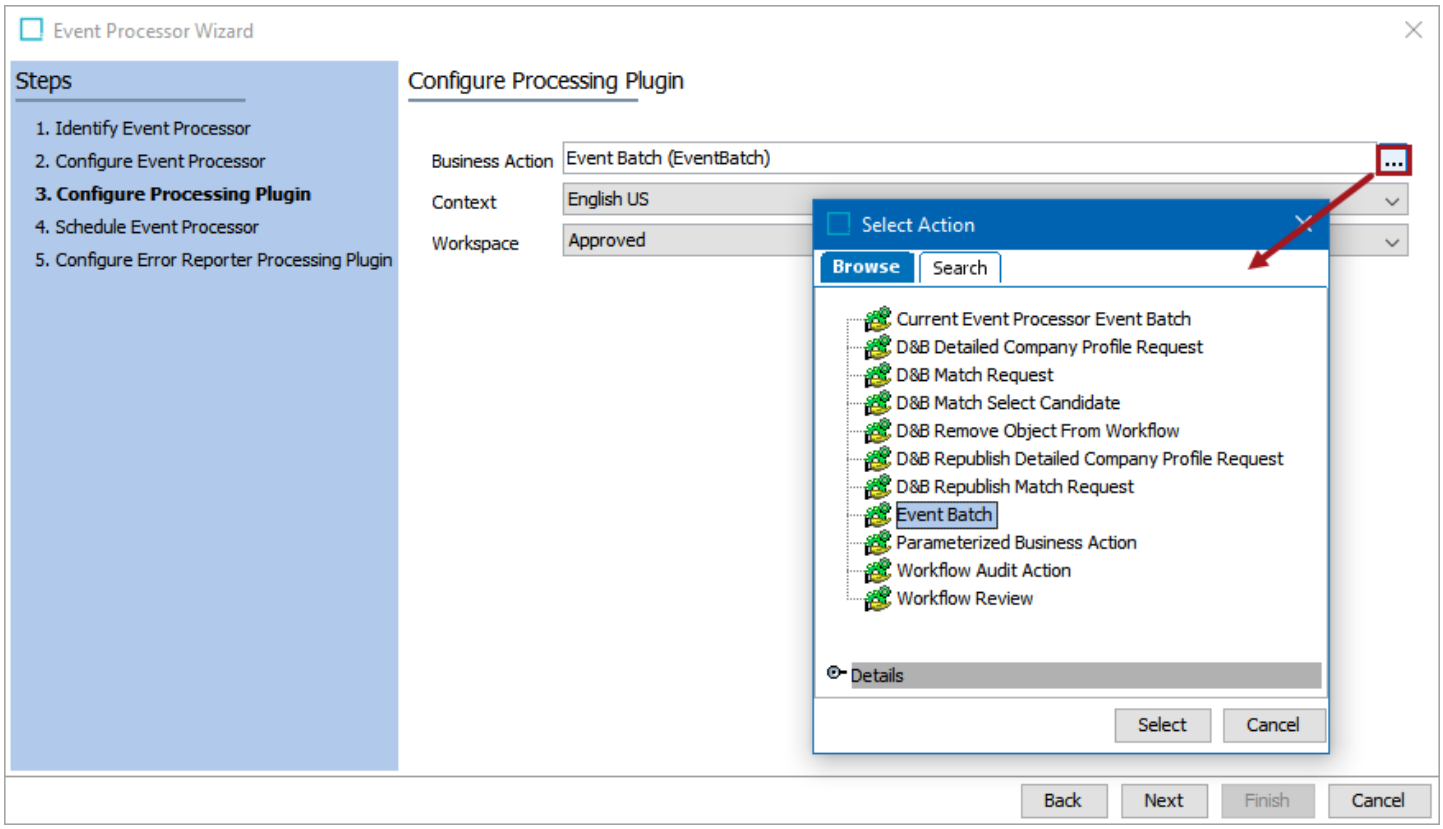
Parameters

Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.

To access the 'Configure Processing Plugin' parameters as shown below, the **Execute Business Action for Event Batch** processing plugin must be selected within the Select Processor parameter during the wizard step 'Configure Event Processor.'



Once the Execute Business Action for Event Batch processing plugin has been selected, click the **Next** button, and the wizard step 'Configure Processing Plugin' will display.



- **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.

As mentioned above, the JavaScript Current Object bind and most non-JavaScript business action plugins cannot be used. The following JavaScript example iterates the events in a batch and logs the ID of each associated Node:

```
// logger bound to Logger
// batch bound to Current Event Processor Event Batch
var it = batch.getEvents().iterator();

while (it.hasNext()) {
                var event = it.next();
                var node = event.getNode();
                if (node) {
                                logger.info("Handling " + event.getNode().getID());

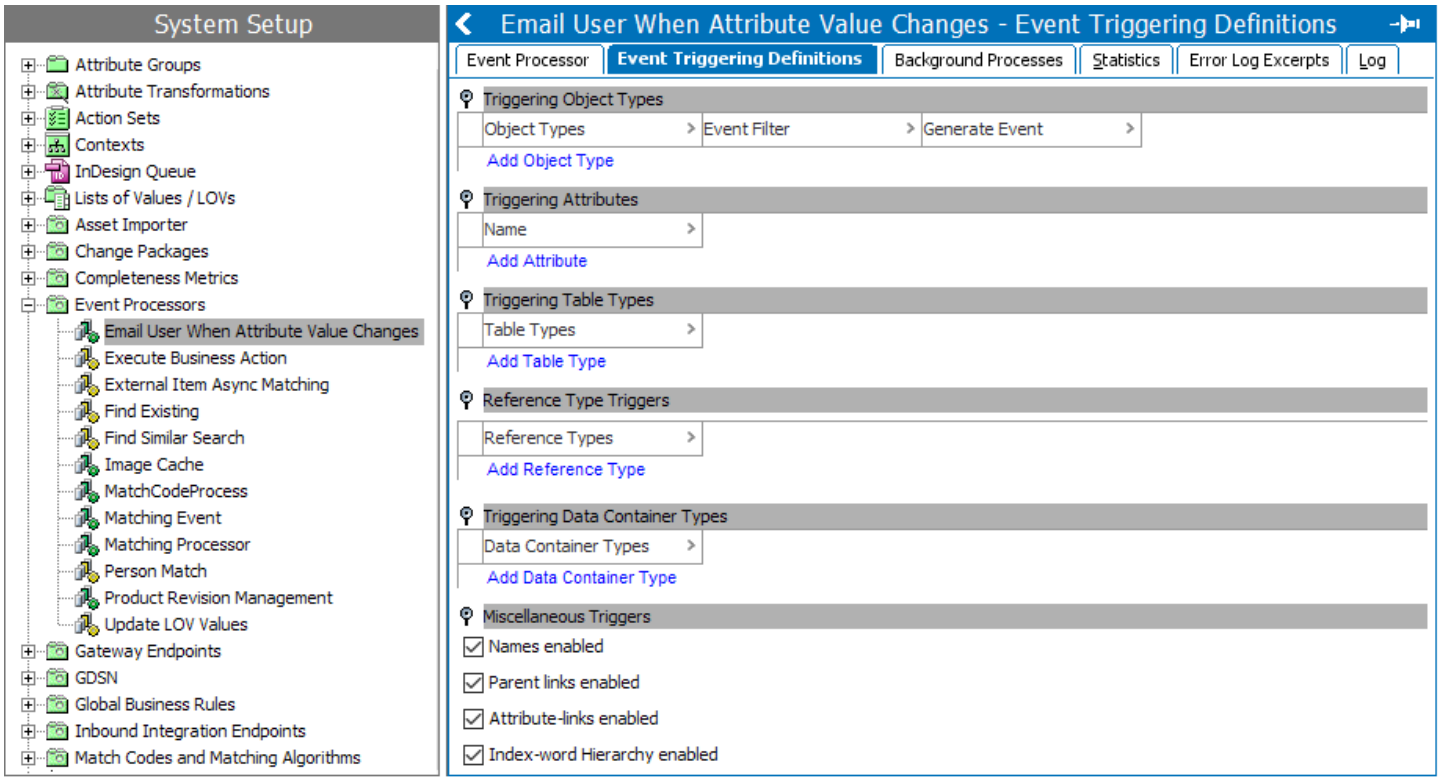
                }
}
```

It is not recommended to have long running business actions. This plugin should only be used for cases where it is critical that the business action work on a batch of events.

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 for Event Batch 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. At a minimum, an object type must be selected.



- 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).
- 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 affected.

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

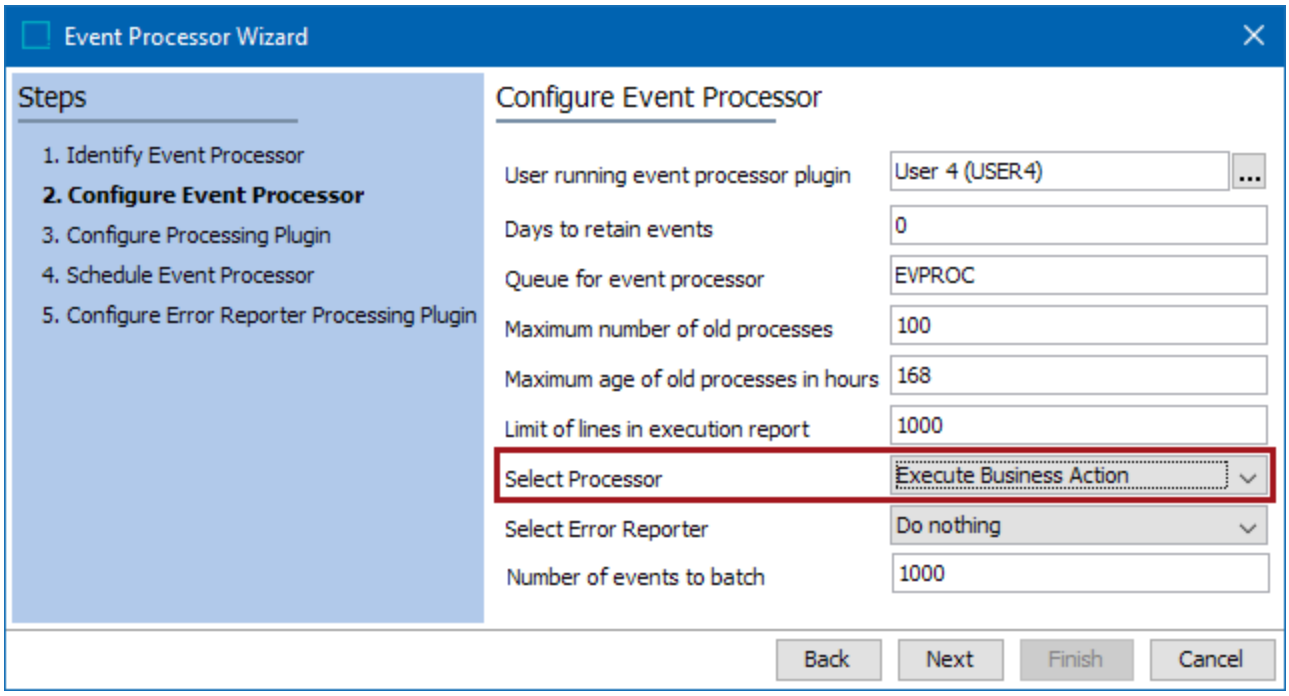
This section of documentation describes configuration steps for this specific processor, but that is only one part of configuring an event processor. For the full set of instructions on configuring an event processor, see the **Event Processors** documentation.

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.

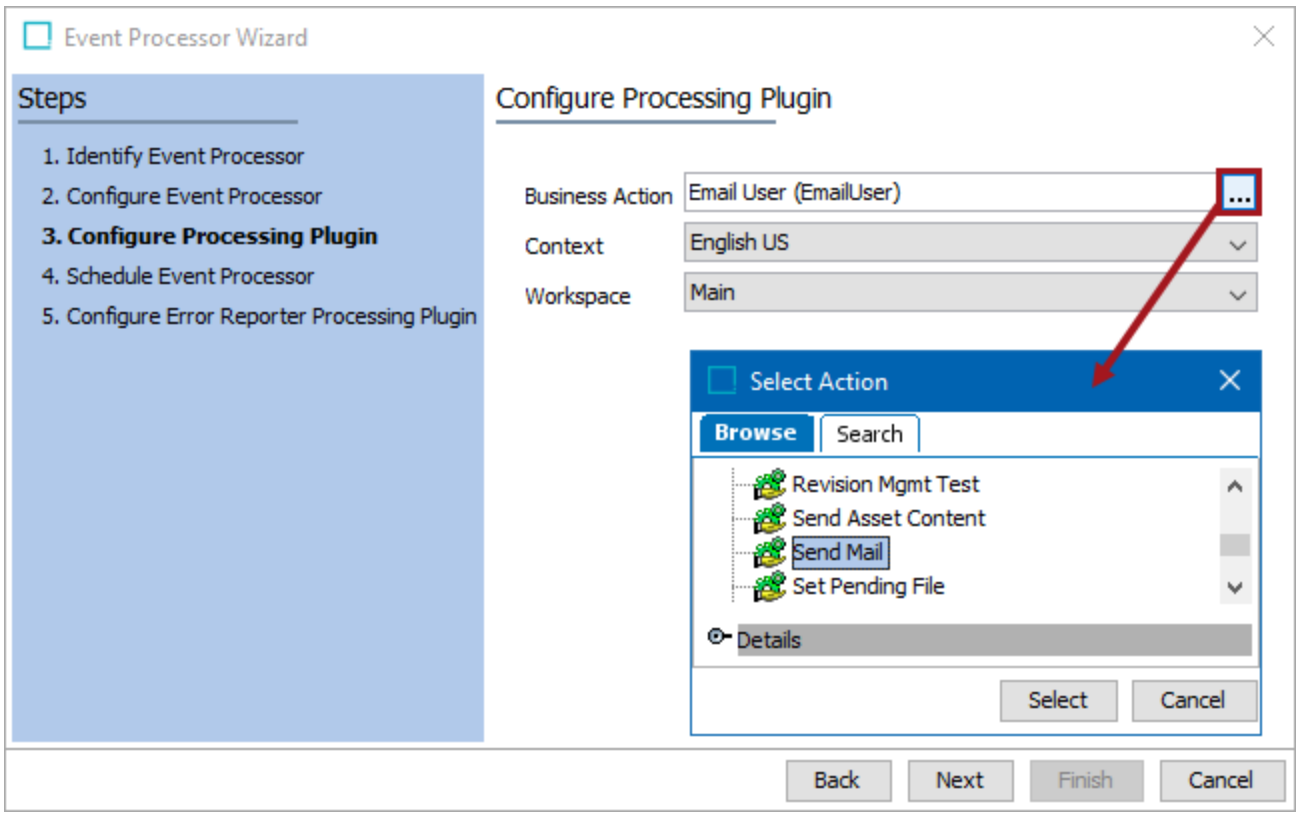
Parameters

Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.

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.'



Once the Execute Business Action processing plugin has been selected, click the **Next** button, and the wizard step 'Configure Processing Plugin' will display.



- **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.

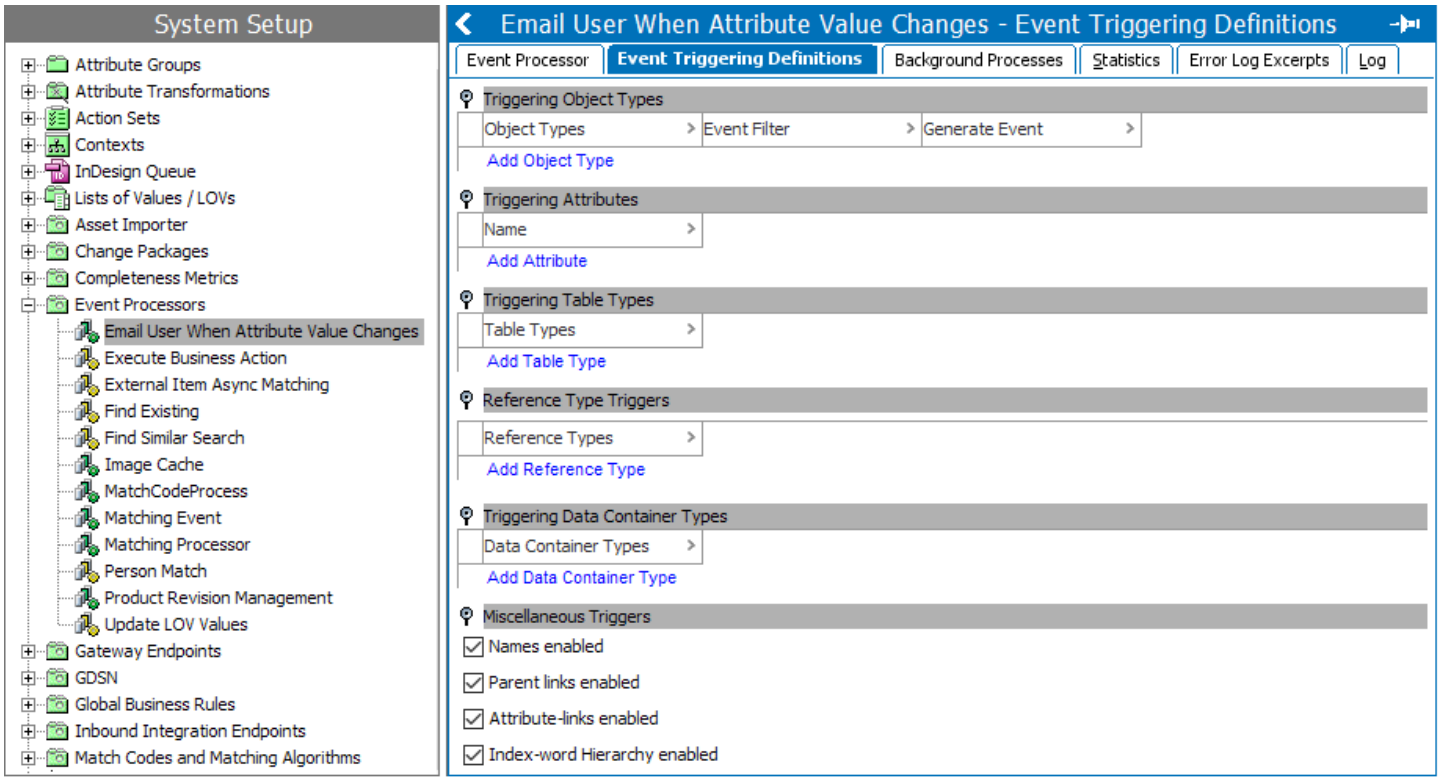
Once any required configurations are made, click the **Next** button to display the wizard step 'Schedule Event Processor.'

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. At a minimum, an object type must be selected.



- 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 affected.

Important: By default, events are discarded. For more information about the optional configuration to enable the Queue Status, see the **Queue Status** section of the **EP - Event Processor Tab** topic.

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.

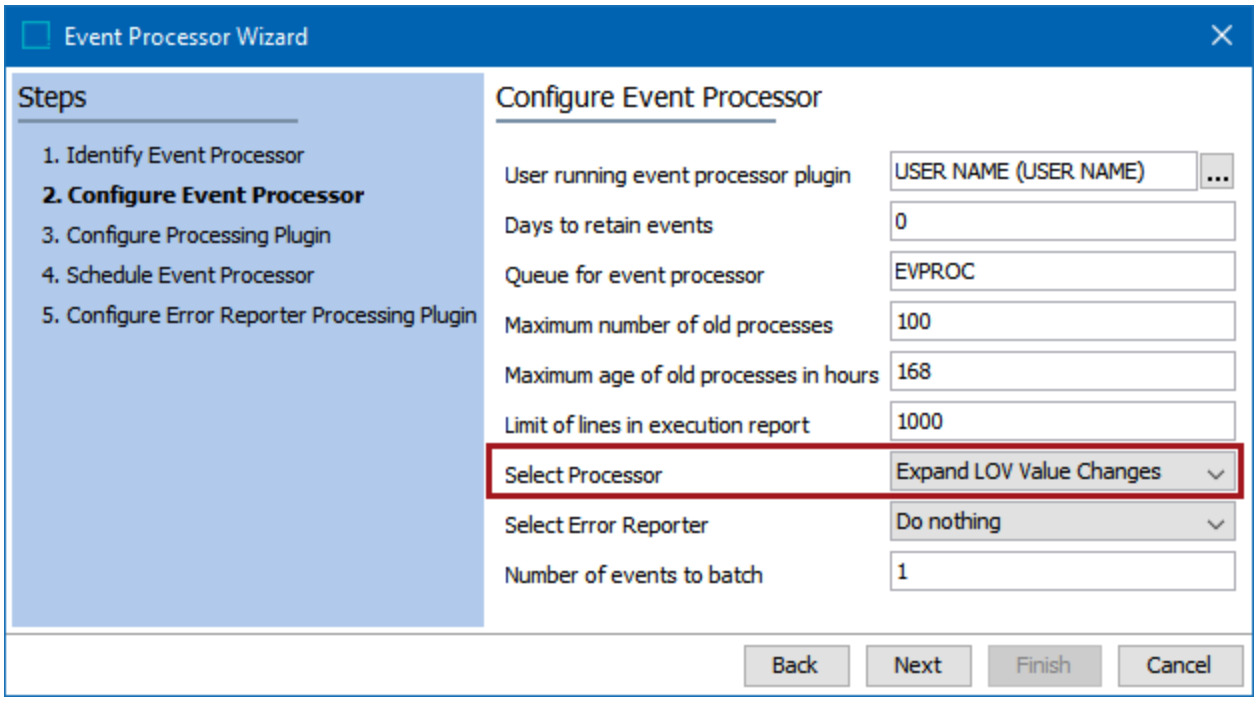
Prerequisites

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.

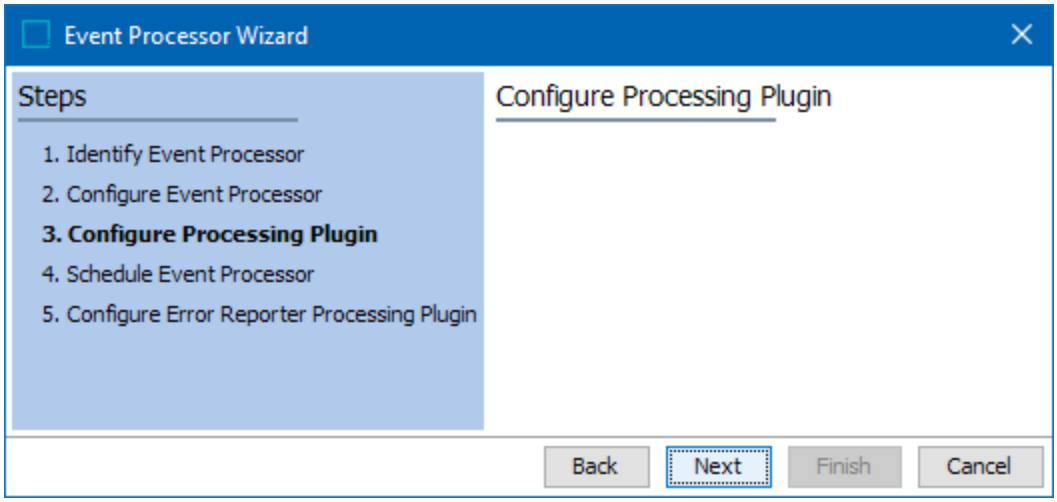
It is expected that anyone configuring the Expand LOV Value Changes processing plugin is familiar with the **List of Values** documentation within the **System Setup** guide.

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.'



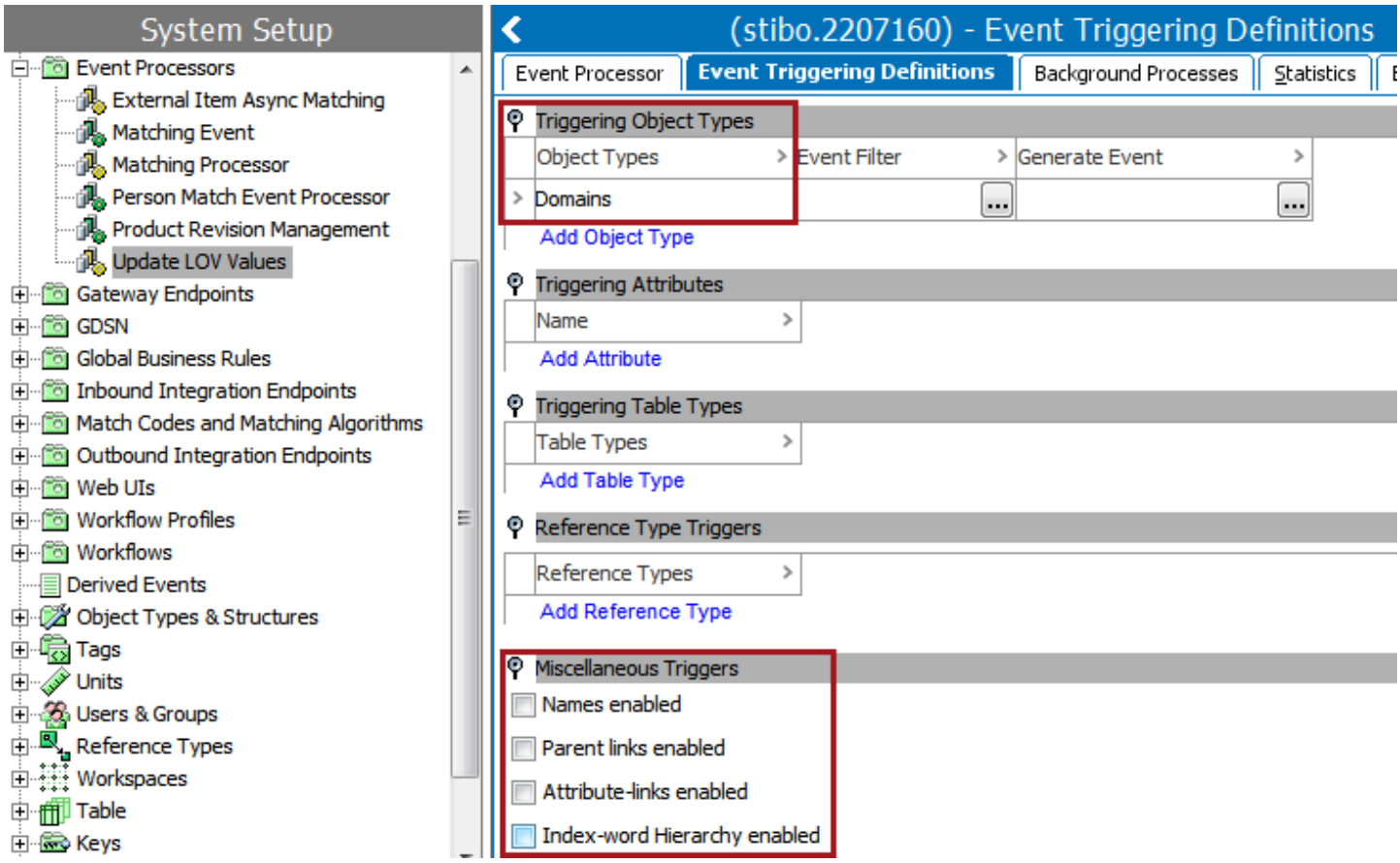
Once the Expand LOV Value Changes processing plugin has been selected, click the **Next** button, and the wizard step 'Configure Processing Plugin' will display.



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.



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

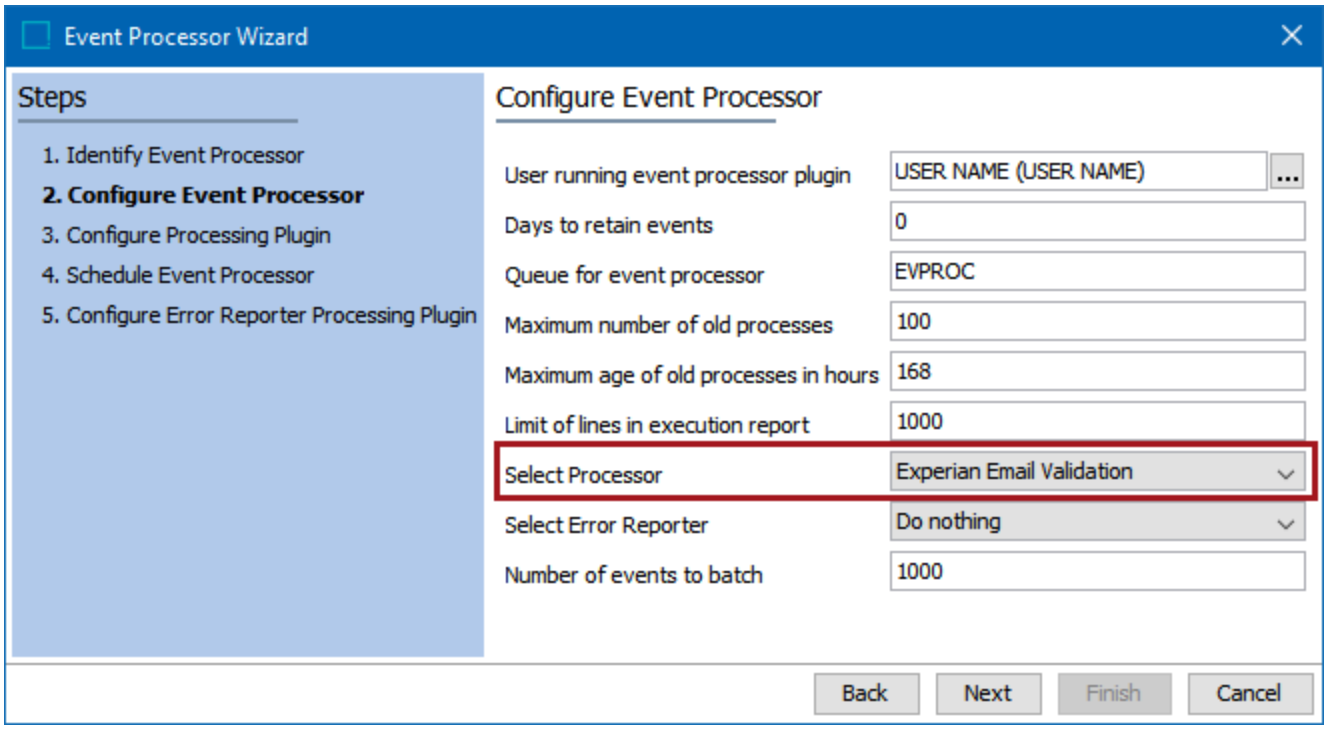
This section of documentation describes configuration steps for this specific processor, but that is only one part of configuring an event processor. For the full set of instructions on configuring an event processor, see the **Event Processors** documentation.

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.

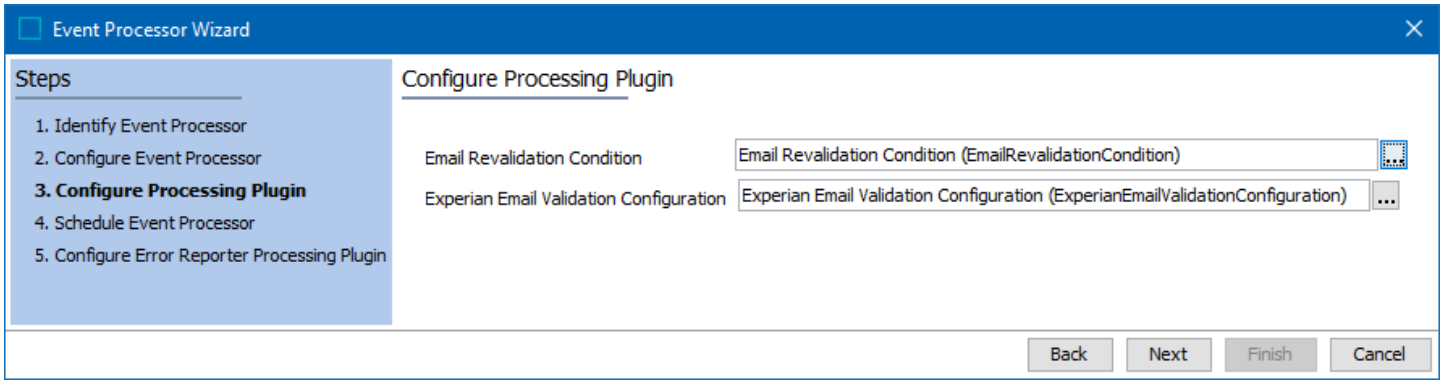
Parameters

Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.

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.'



Once the Experian Email Validation processing plugin has been selected, click the **Next** button, and the wizard step 'Configure Processing Plugin' will display.



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.

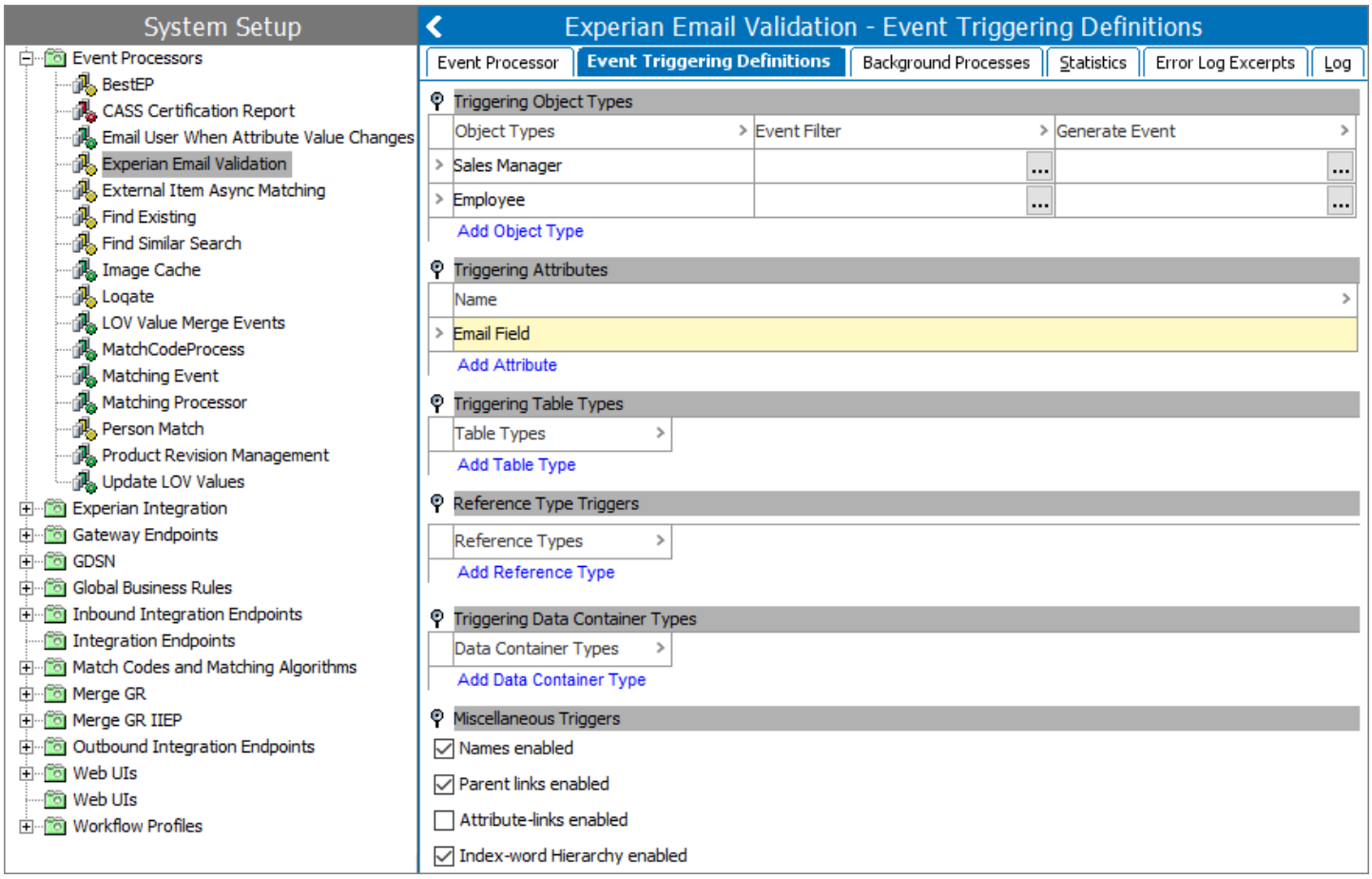
Event Triggers

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.



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.

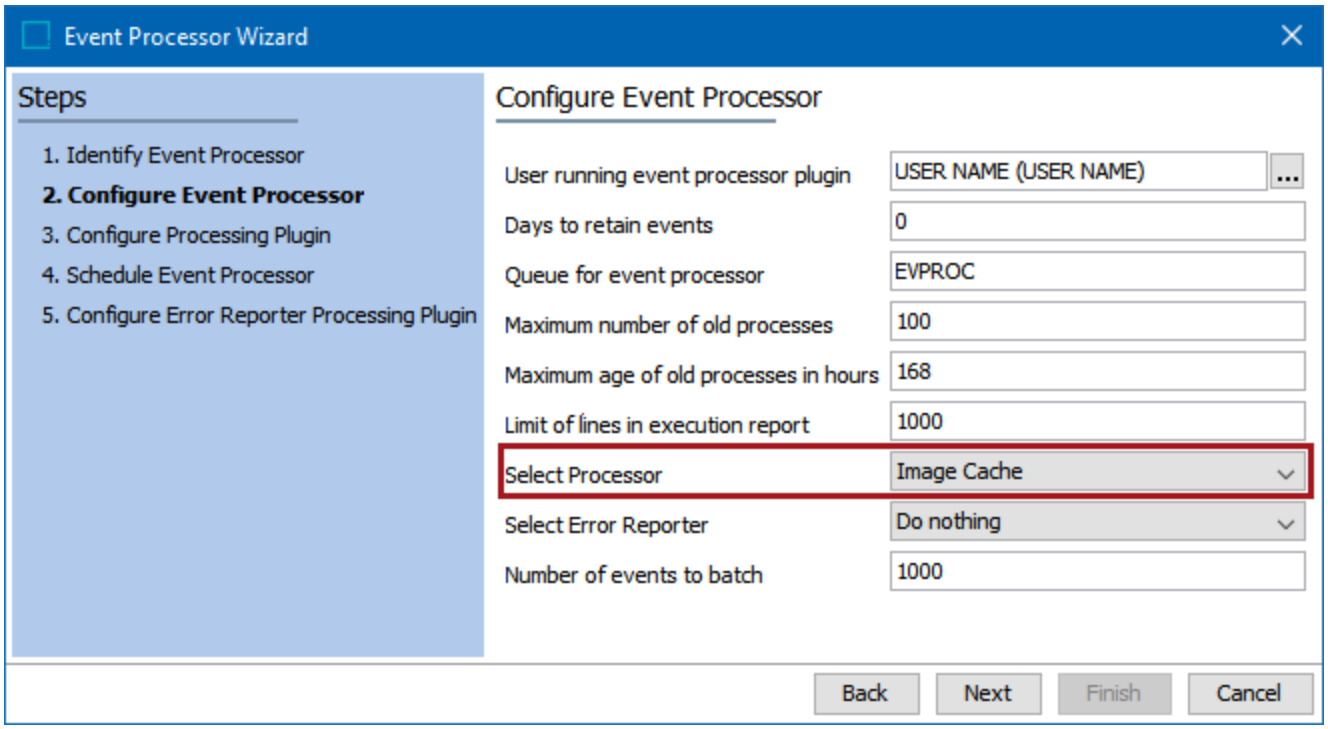
Note: In the case of a STEP system app server restart, the event processor restarts in the same queue which can potentially cause the same emails to be resent for validation.

Important: By default, events are discarded. For more information about the optional configuration to enable the Queue Status, see the **Queue Status** section of the **EP - Event Processor Tab** topic.

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.'



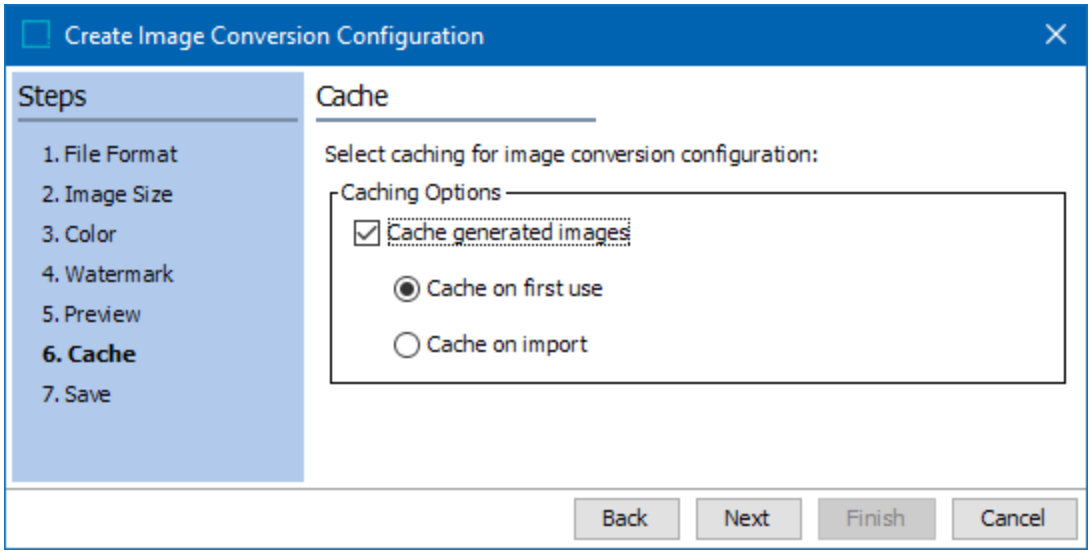
The **Parameters** and **Event Triggers** sections below contain important information on settings that should be considered when creating an event processor using this processing plugin.

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

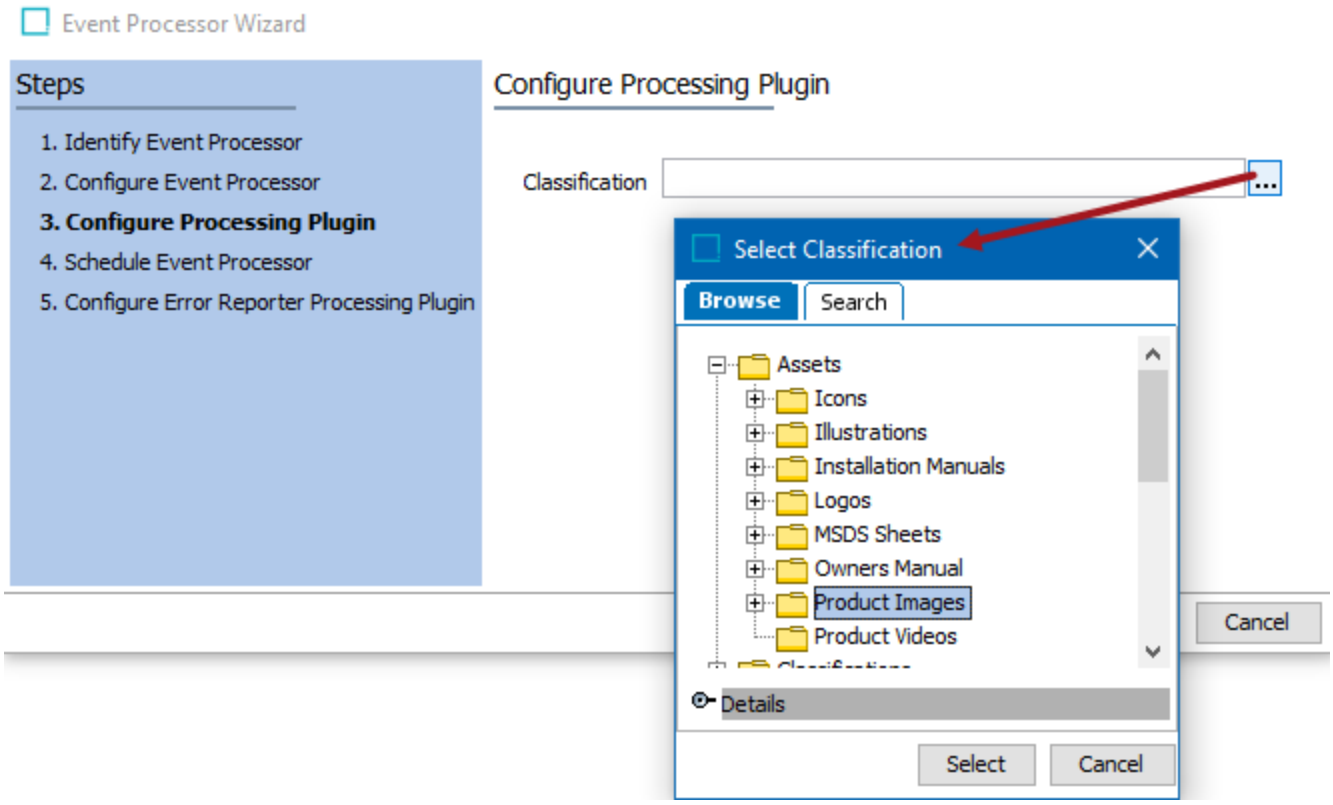
information on creating these conversions, see the **Image Conversion Configuration** topic of the **Digital Assets** documentation.



Parameters

Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.

- **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.

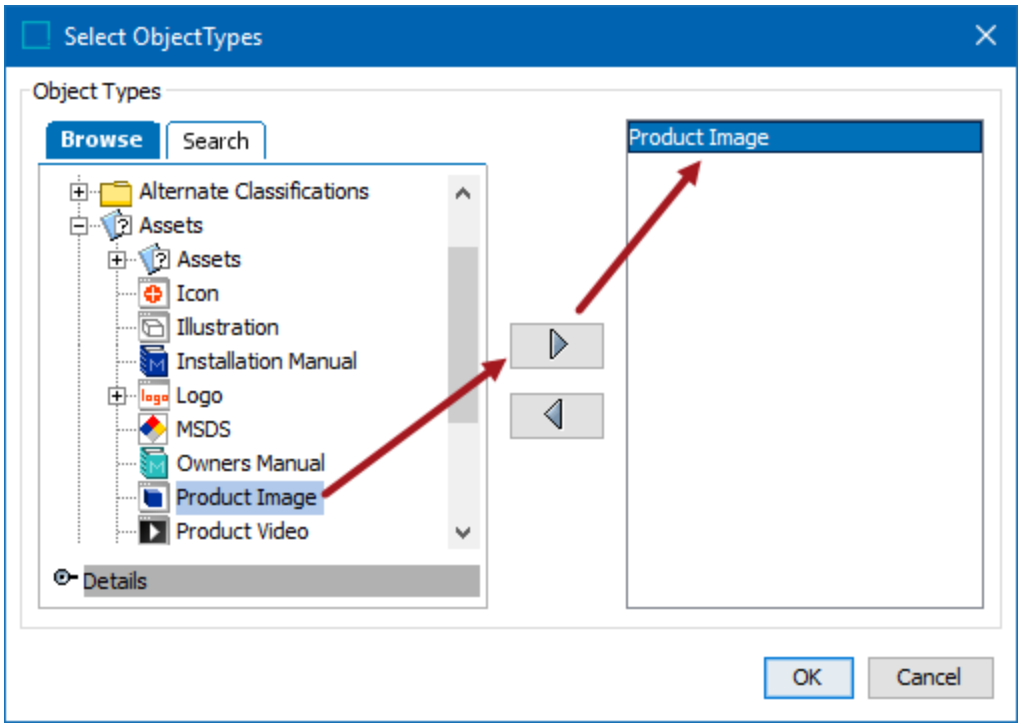


Important: Although caching images is optional, a cache improves export performance, especially for those assets that are exported frequently. Image caching is automatically defined for STEP with no setup required. All image caching is controlled by the ImageCache.* properties in the sharedconfig.properties 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

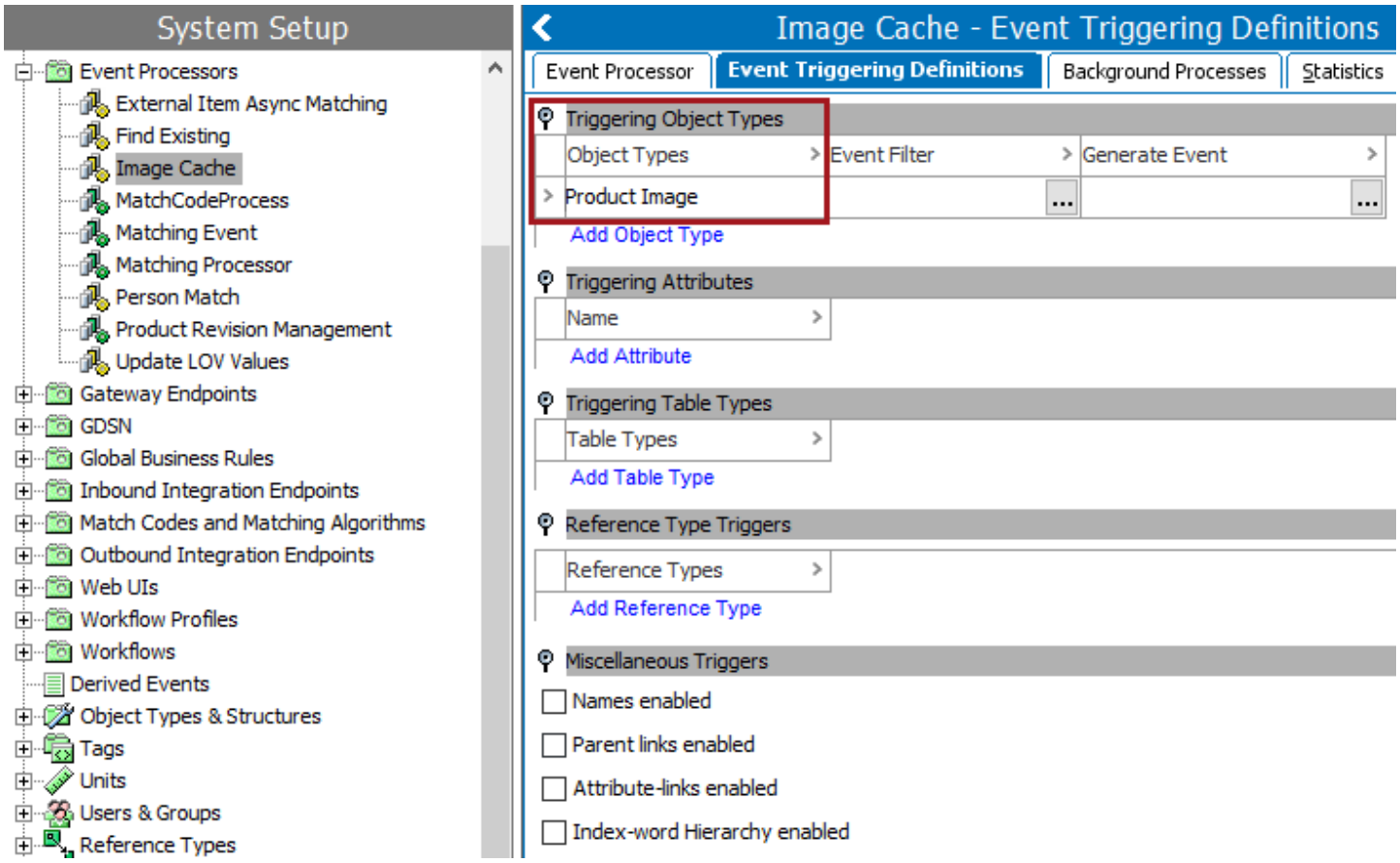
For the event processor to take effect, it must be configured to only listen for changes on asset type objects housed within the same previously selected classification folder.

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.



4. 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

The result of matching may change when data changes. Updates result in events and 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.

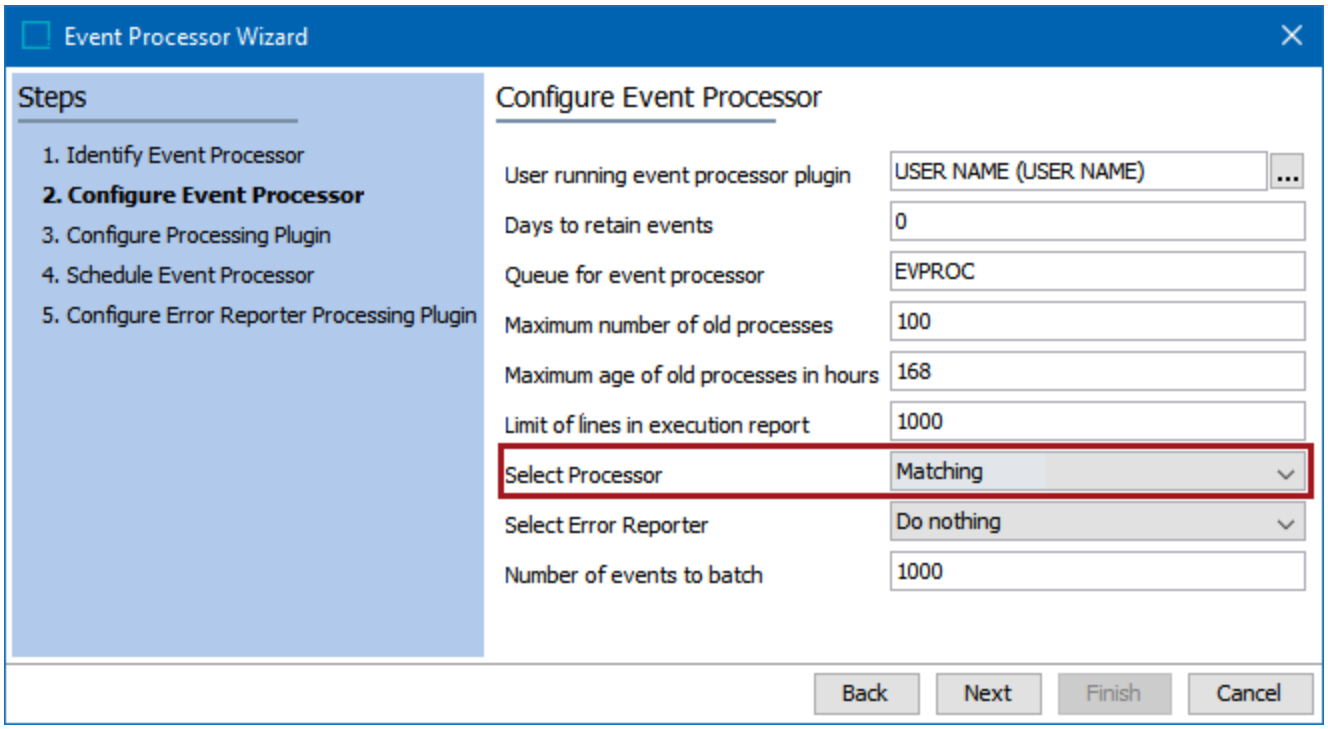
The configuration requires selection of a matching algorithm, and that match codes are only updated for objects acted on by the algorithm and only when events are passed for those objects. For details about match codes and matching algorithms, see **Matching, Linking, and Merging Elements** in the **Matching, Linking, and Merging** documentation.

Important: The 'Run Matching Algorithm' functionality is controlled by System Setup > Action Sets > Setup Actions > Maintain Deduplication Configurations.

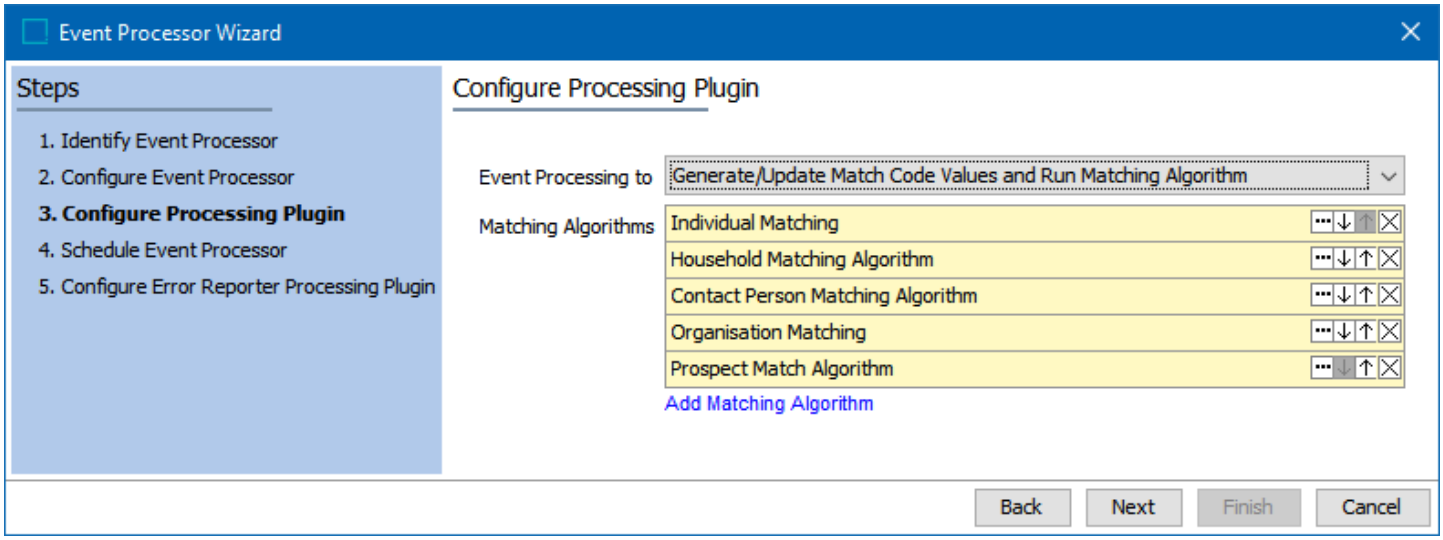
Parameters

Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.

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.'



Once the Matching processing plugin has been selected, click the **Next** button, and the wizard step 'Configure Processing Plugin' displays.

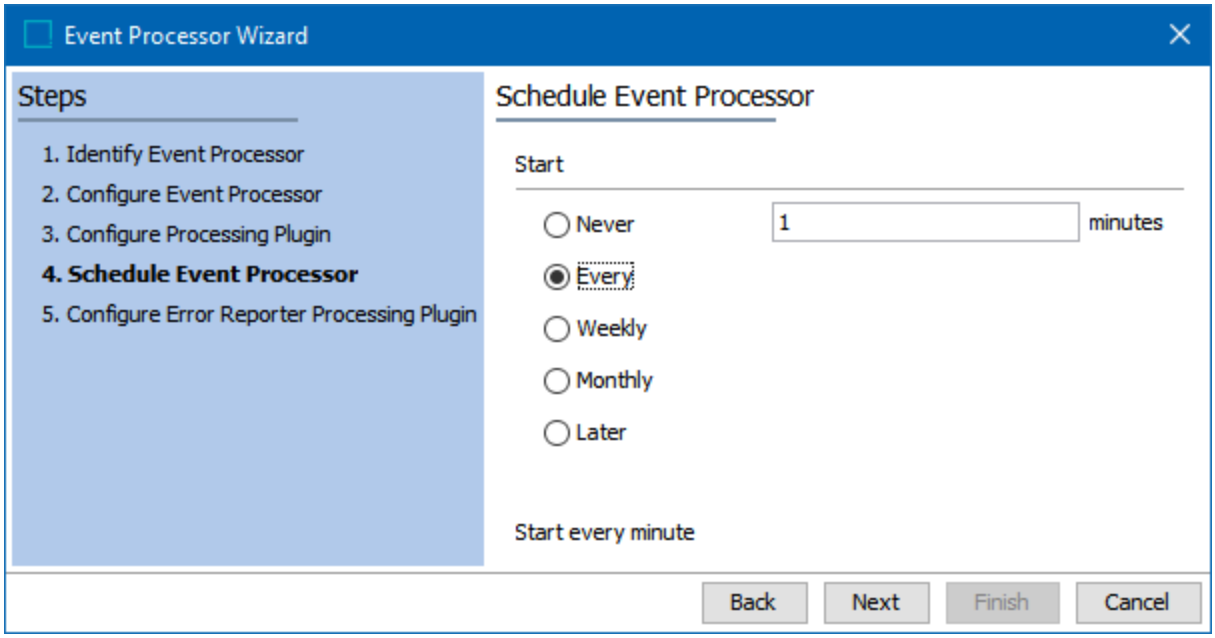


- **Event Processing to:** Select an option from the dropdown.
  - **Run Matching Algorithm Only:** runs the selected algorithm(s) using the existing match codes.
  - **Generate / Update Match Code Values Only:** creates missing match codes, deletes invalid match codes, but does not run the algorithm(s).

  - **Generate / Update Match Code Values and Run Matching Algorithm:** this is the recommended option and 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.

Important: While it is possible to use the same Matching Algorithm across Event Processors, this scenario will usually result in an optimistic locking and/or unique constraint violation when the two processors conflict with one another. Optimistic locking is the implementation where objects are modified concurrently. The result of this is that the system cannot know which value should be represented. To avoid these issues, ensure that each algorithm on the system is run by a single Event Processor.

On the Schedule Event Processor step, select the desired schedule. The recommendation is every 1 minute.



Note: The comparison run by a Matching event processor happens asynchronous to the actual import, leaving up to a minute between importing a record that should go into the clerical review workflow and making the record visible in the clerical review workflow.

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.

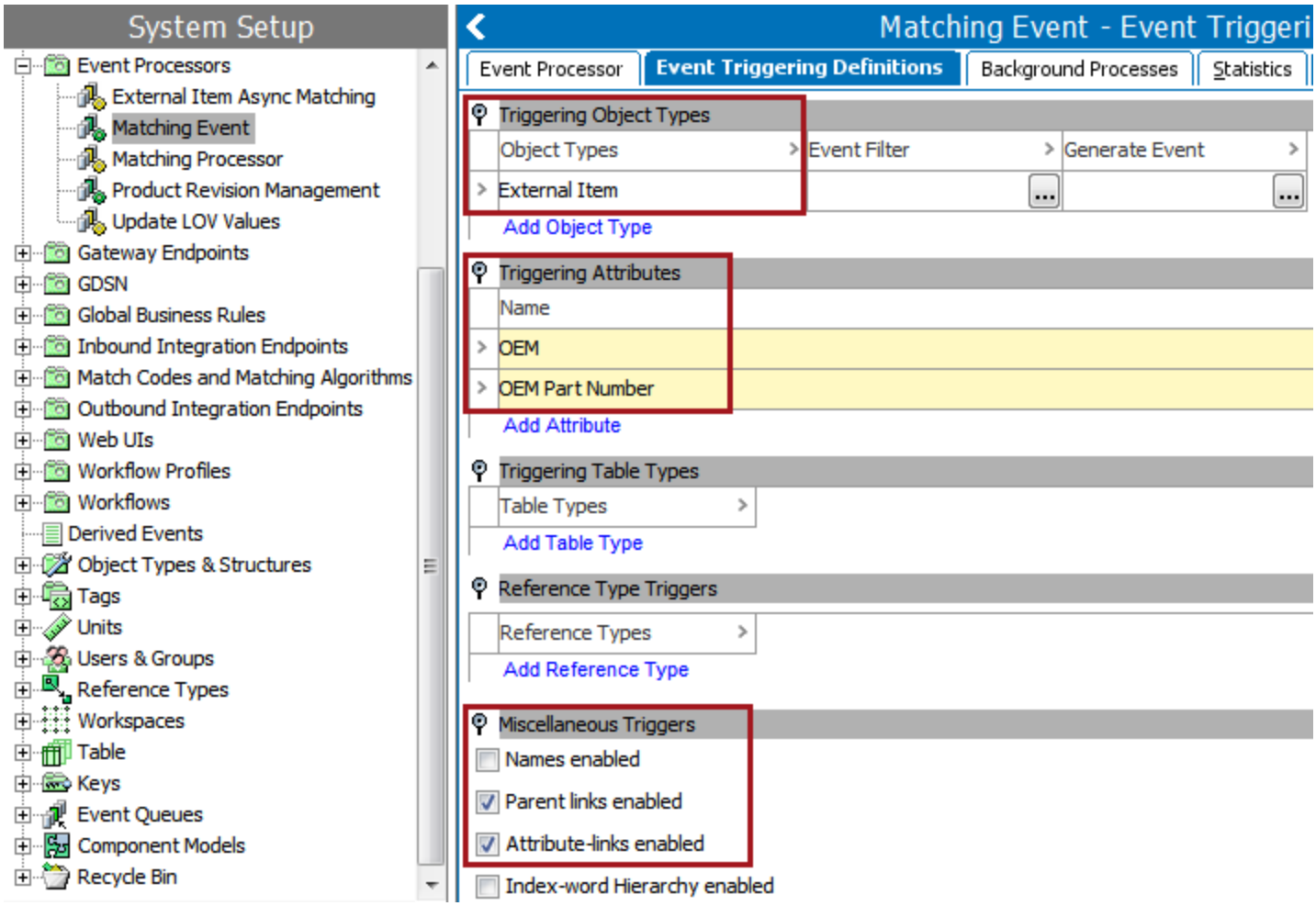
For a **match and merge** scenario, based on the selected algorithm, for existing golden records, the event processor performs a merge or initiates a clerical review. Add triggers for the following:

- references defined by your **Matching component model**: Non-Duplicate Reference Types
- references defined by your **Matching - Merge Golden Record References component model**: Unmerged-From Relation Reference Types
- attributes, references, and data containers included in your **Match Criteria**

> **Important:** For accurate match and merge functionality, the event processor must trigger on updates that can change the outcome of the record comparisons. To accomplish this, the recommendation is to trigger on any attribute, reference, or data container that is used in the match criteria.

For a match and merge scenario, avoid triggers on the following attributes and reference types as defined by your component models:

- Potential Duplicate Reference Type
- Merged-Into Relation Reference Types
- Source Relation Reference Type
- Potential Duplicate Match Algorithm ID Attribute
- Source Record ID Attribute
- Deactivated Attribute



When configuration is complete, enable the matching event processor. For detailed steps, see the **Enabling Event Processors** section of the **Running an Event Processors** topic in the **System Setup / Super User Guide** documentation.

Troubleshooting

The following logs and statistics provide information for troubleshooting the event process.

- Review the event processor background process log to determine when events were processed and what actions were taken in response.
- Review event processor performance measurements on the Statistics tab for both matching algorithms and match code configurations.

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.

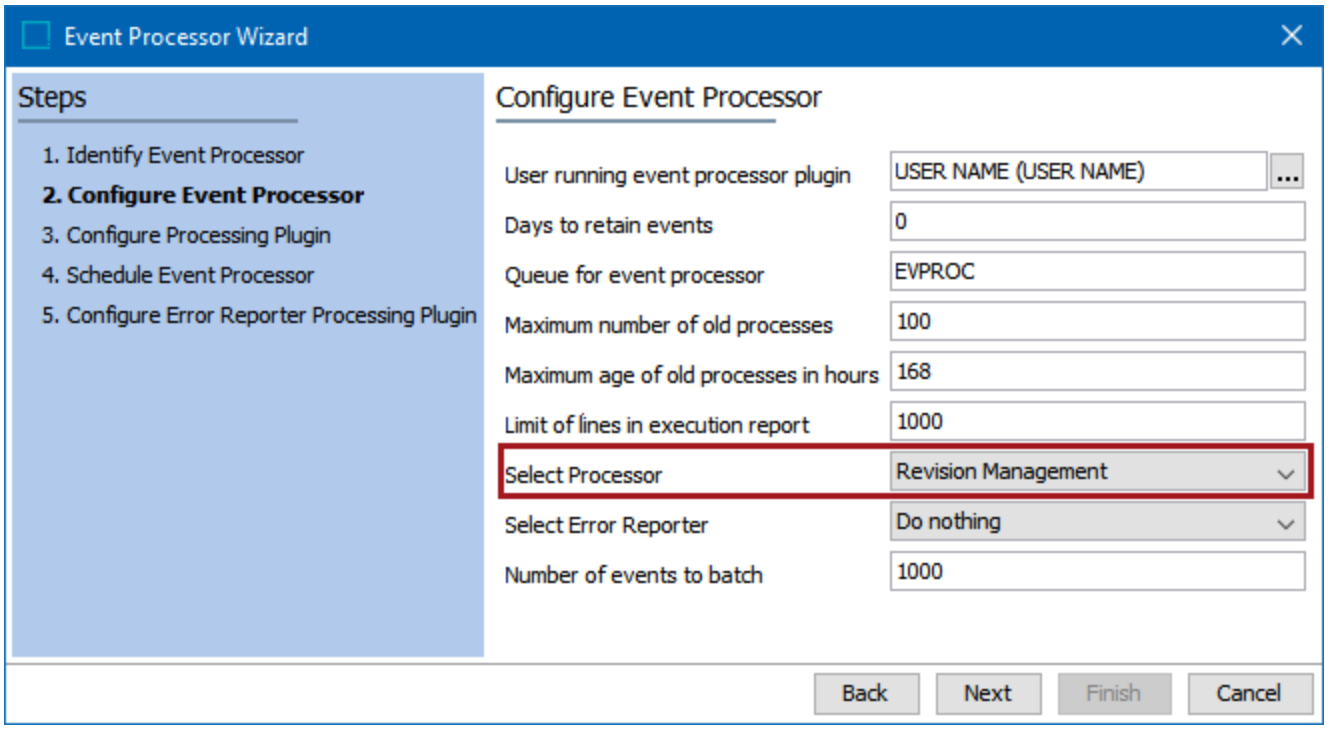
Prerequisites

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.

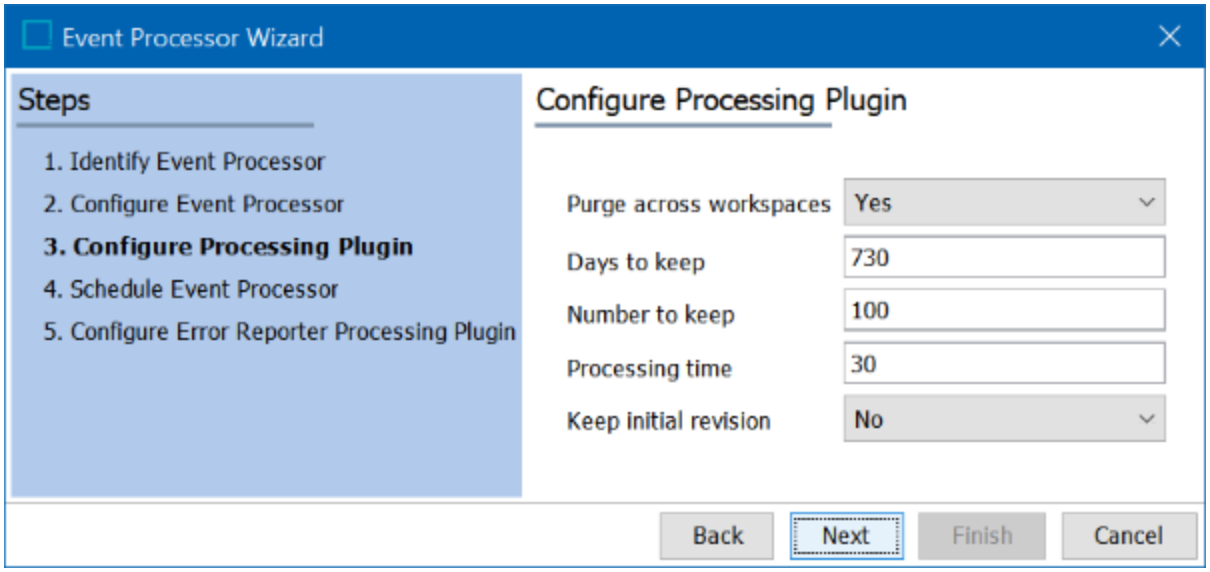
Parameters

Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.

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.'



Once the Revision Management processing plugin has been selected, click the **Next** button, and the wizard step 'Configure Processing Plugin' will display.



- **Purge across workspaces:** Select 'Yes' to purge object revisions across all workspaces or 'No' (the default) for only the execution workspace. More revisions can typically be removed if the analysis and removal is done across workspaces, but this method takes more time. When this option is enabled, the 'Processing time' and 'Keep initial revision' settings will have no effect.

Important: If both the 'Purge across workspaces' and the 'Keep initial revision' are set to 'Yes,' the event processor will fail.

- **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. Consider setting this to 1 for most implementations.
- **Number to keep:** maximum number of revisions remaining after processing. Consider setting this to 1 for most implementations.
- **Processing time:** determines the maximum number of seconds that the processor will run on each object. **Must be set to 30.**

  Note: If 'Processing time' is set to any number other than 30, you will encounter a RuntimeException when events are processed.

- **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.

Important: A revision is deleted when it meets at least one of the conditions set in the Days to keep, Number to keep, and Processing time parameters. These parameters work as an 'OR' statement to determine the revisions to permanently remove.

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, **or**
2. More than 100 revisions will remain after deletion of the revision being processed, **or**
3. The processor has not run for 30 seconds on the revision being processed, **or**
4. The revision being processed is NOT the initial revision.

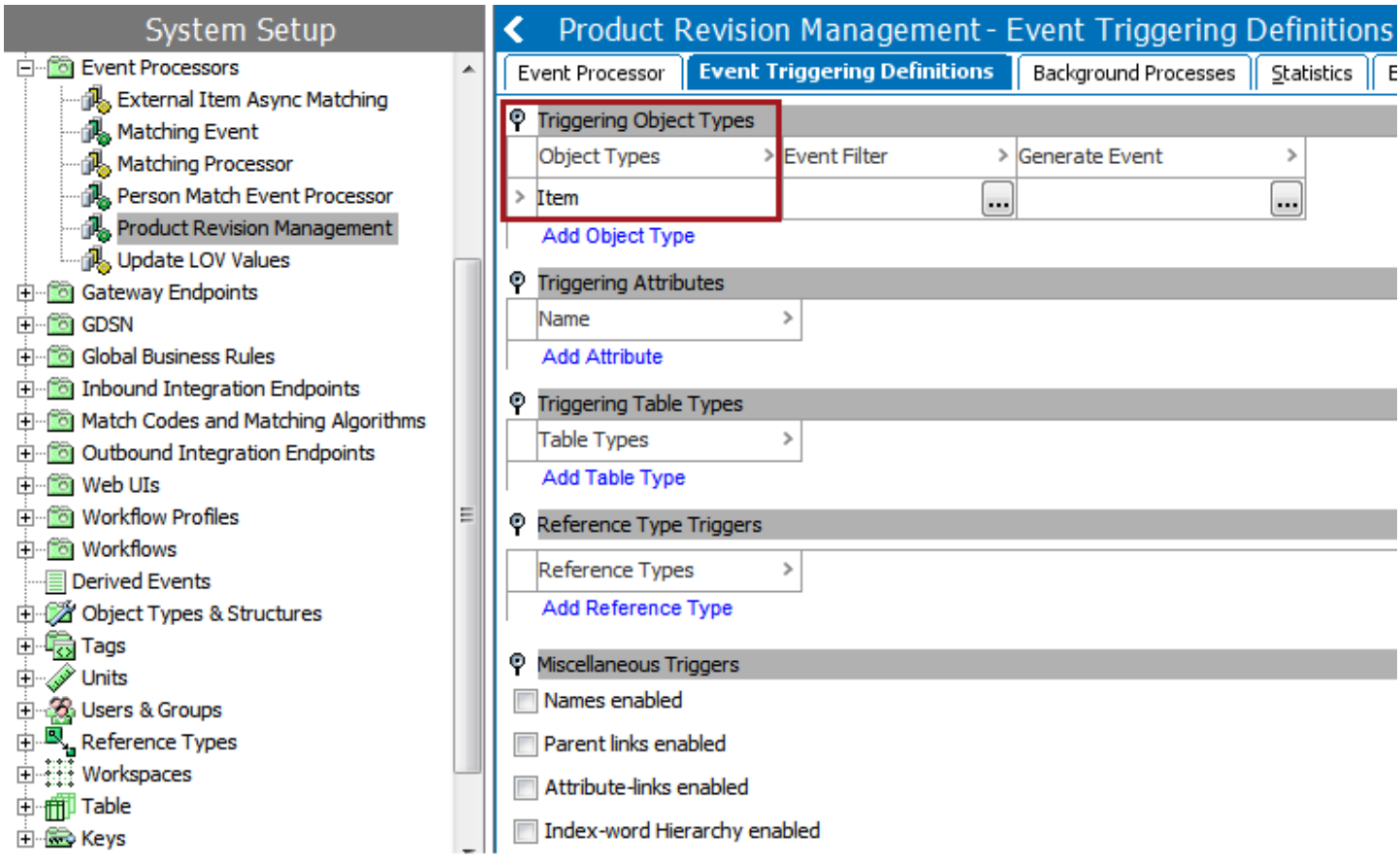
Event Triggers

Select the object type(s) that should have revisions deleted according to the configured limits.

Note: The revision management event processor **does not** search the database and remove the revisions of all objects when the revision management event processor criteria are met. Instead, the revision management event processor **does** removes revisions only according to the criteria set of a single object when an event is triggered for the object.

If a different number of revisions is needed based on the object type, create an additional event processor with the necessary configuration limits.

Additionally, set Triggering Attributes, Reference Triggering Types, and/or Miscellaneous Triggering Types as necessary. Event Triggering Definitions are the same in Event Processors as they are in an OIEP. For details about the triggers, see the **OIEP - Event-Based - Event Triggering Definitions Tab** in the **Data Exchange** documentation.


System Setup
Event Processors
External Item Async Matching
Matching Event
Matching Processor
Person Match Event Processor
Product Revision Management
Update LOV Values
Gateway Endpoints
GDSN
Global Business Rules
Inbound Integration Endpoints
Match Codes and Matching Algorithms
Outbound Integration Endpoints
Web UIs
Workflow Profiles
Workflows
Derived Events
Object Types & Structures
Tags
Units
Users & Groups
Reference Types
Workspaces
Table
Keys
Product Revision Management - Event Triggering Definitions
Event Processor
Event Triggering Definitions
Background Processes
Statistics
Triggering Object Types
Object Types
Event Filter
Generate Event
Item
Add Object Type
Triggering Attributes
Name
Add Attribute
Triggering Table Types
Table Types
Add Table Type
Reference Type Triggers
Reference Types
Add Reference Type
Miscellaneous Triggers
Names enabled
Parent links enabled
Attribute-links enabled
Index-word Hierarchy enabled

Source Record Data Management Processing Plugin Parameters and Triggers

Storing source data takes up disk space in the database. To keep the extra storage impact stable over time, the historical source data values should eventually be purged. The source data does not use the In-Memory functionality and so does not require additional memory. When importing new records, individual records can require an additional 20-30 percent of space to accommodate for the source records. However, with minor updates to existing golden records, source data can require 250-300 percent of additional disk space for the individual records.

The 'Source Record Data Management – Historical Values Cleanup' event processor incrementally purges the oldest historical source data. This includes simultaneously cleaning historical values and optimizing storage to keep disk usage as low as possible. Depending on the average amount of changing values at each import, the extra storage impact for updates could be reduced to 50-80 percent.

The purging logic is individual per source record and per attribute / reference / data container, so frequently changing values does not cause other historical values to be purged prematurely.

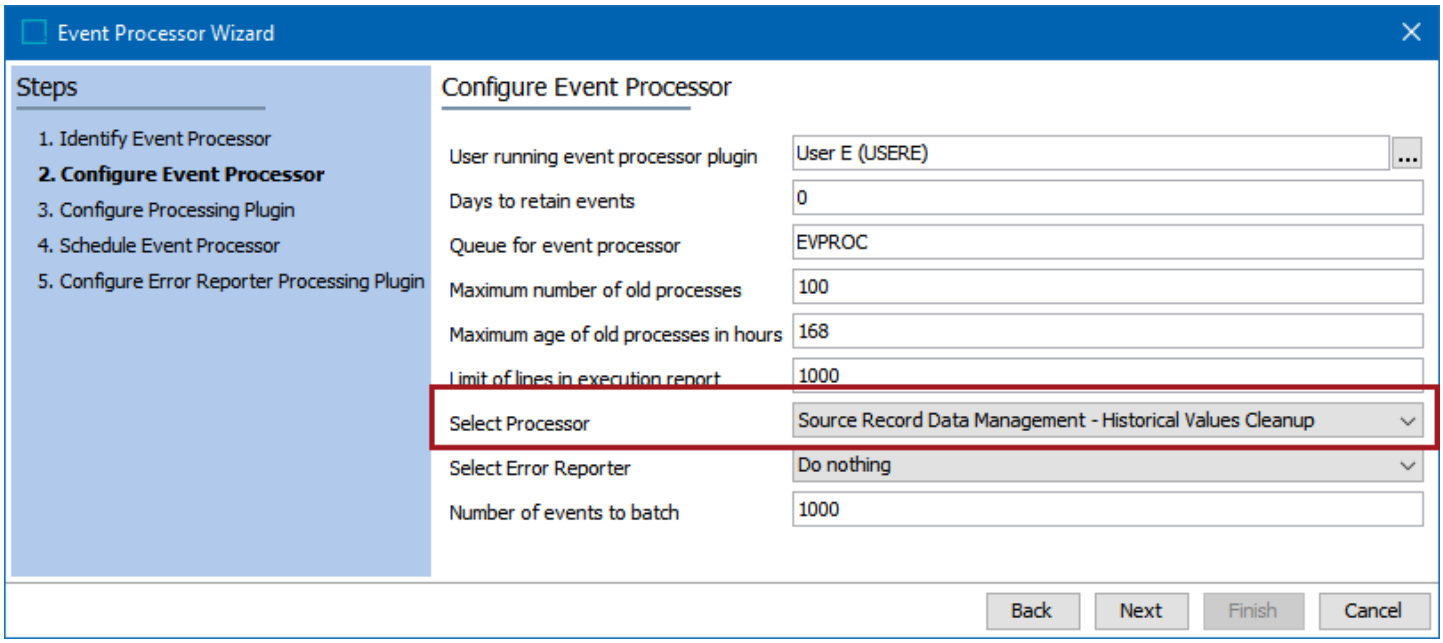
The **Parameters** and **Event Triggers** sections below contain important information on settings that should be considered when creating an event processor using this processing plugin.

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.

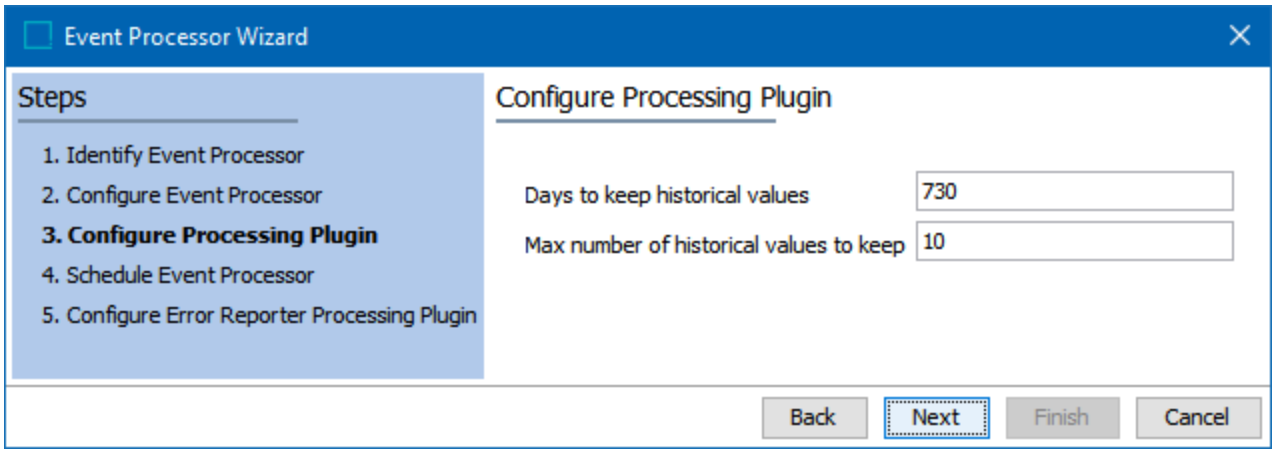
Parameters

Each of the relevant parameters for the Event Processor Wizard 'Configure Processing Plugin' step are described below. Any additional wizard parameters with importance for this plugin are also included in this topic.

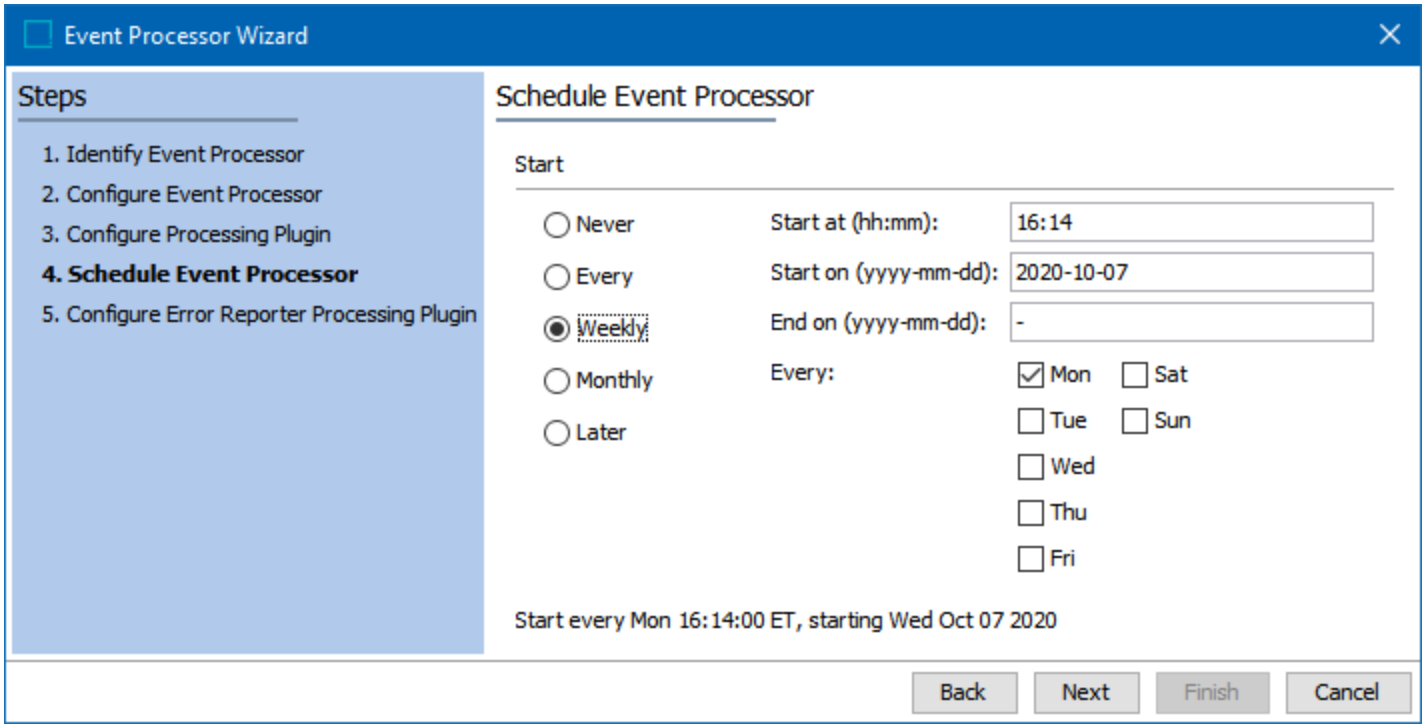


1. On the 'Configure Event Processor' step, for the **Select Processor** parameter, choose the **Source Record Data Management – Historical Values Cleanup** processing plugin to display the expected parameters on the 'Configure Processing Plugin' step.
2. Click the **Next** button to display the wizard step 'Configure Processing Plugin' and set the following parameters:
   - Days to keep historical values – The number of days historical values are kept before purging
   - Max number of historical values to keep – The number of values, changing over time, to keep

Note: The data purge occurs if just one of the two configuration criteria are met. To disable the purging of historical values and keep the disk usage optimization functionality, set both configuration values to a very high number.



3. Click the **Next** button to display the wizard step 'Schedule Event Processor' and schedule as required.



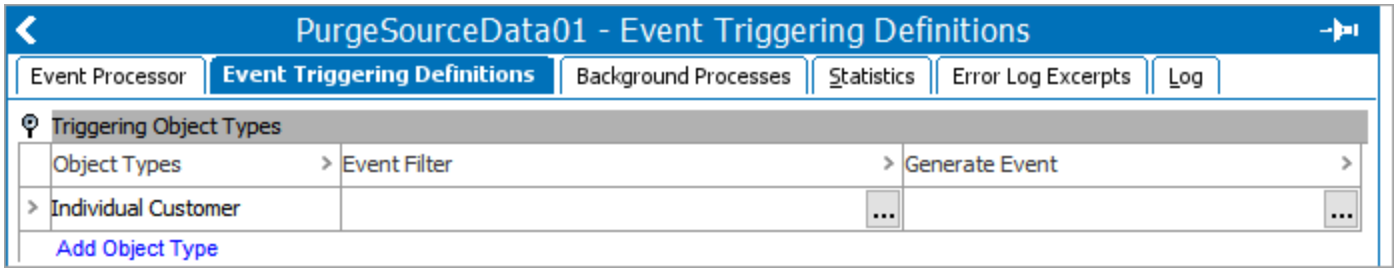
4. Click the **Next** button to display the wizard step 'Configure Error Reporter Processing Plugin' and configure as required.

5. Click the **Finish** button.

Event Triggers

For the event processor to function as scheduled, the Event Triggering Definitions tab must be configured correctly.

- **Triggering Object Types** - set at least one valid Golden Record object types.
- **Triggering Workspace** - if using workspaces, typically this should be the Main workspace.



Event Triggering Definitions are the same in Event Processors as they are in an OIEP. For details about the triggers, see the **OIEP - Event-Based - Event Triggering Definitions Tab** in the **Data Exchange** documentation.

Initial Setup for Event Processors

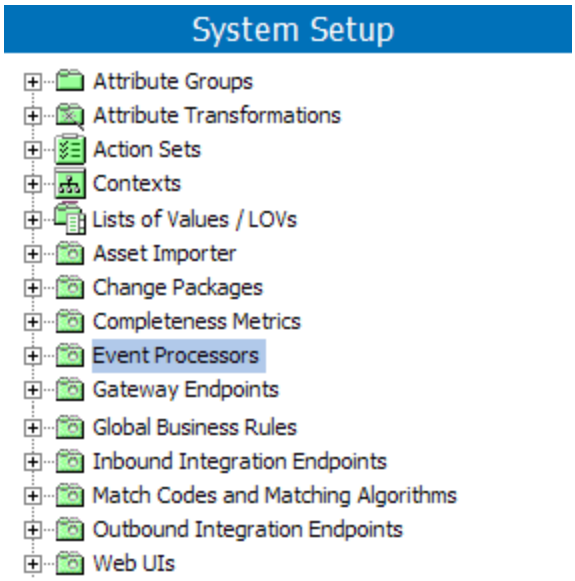
Before you can create an event processor, the following one-time setup steps must be completed. Details for each of these are provided below.

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 an event processor node already exists. The name of the node on your system is not required to match the one in the image below.

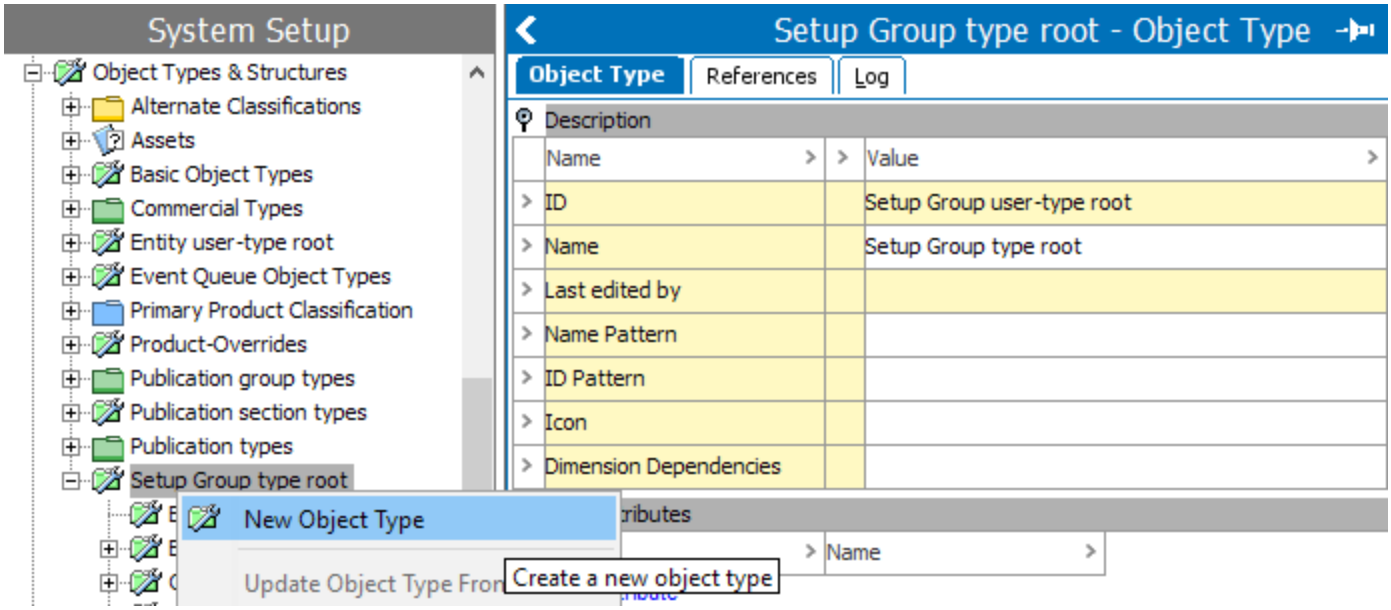


Once the setup has been completed, the steps in this section are only needed if additional levels of organization are desired.

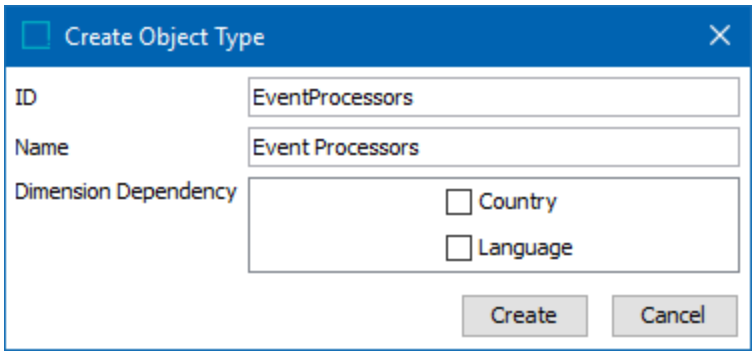
Create Setup Group Type for Event Processors

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.

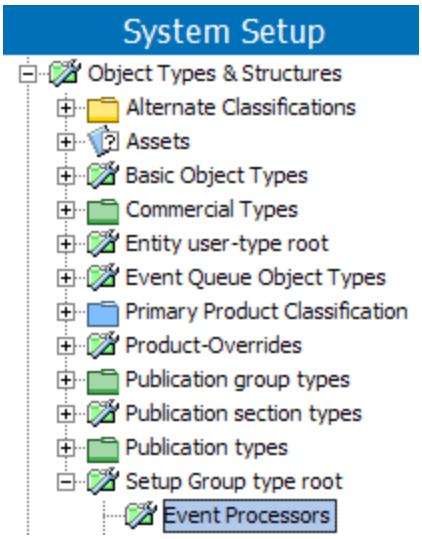


3. Click **New Object Type**, and the Create Object Type dialog will display.

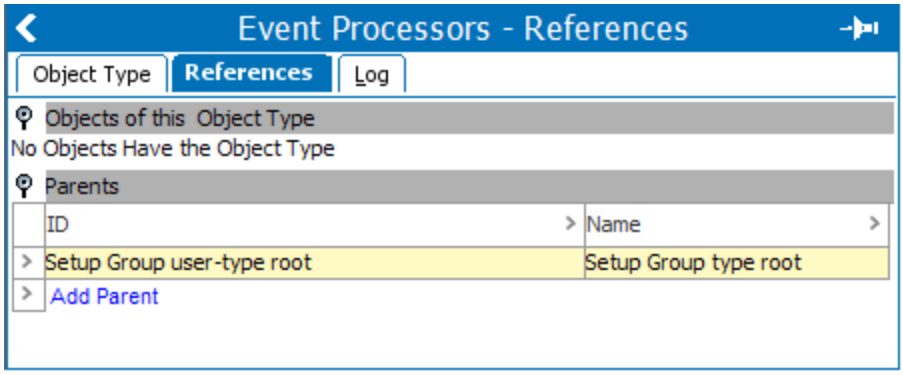


4. Enter an **ID**.
5. Enter a **Name**.
6. Select any required **Dimension Dependency**.
7. Click **Create.**

   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.



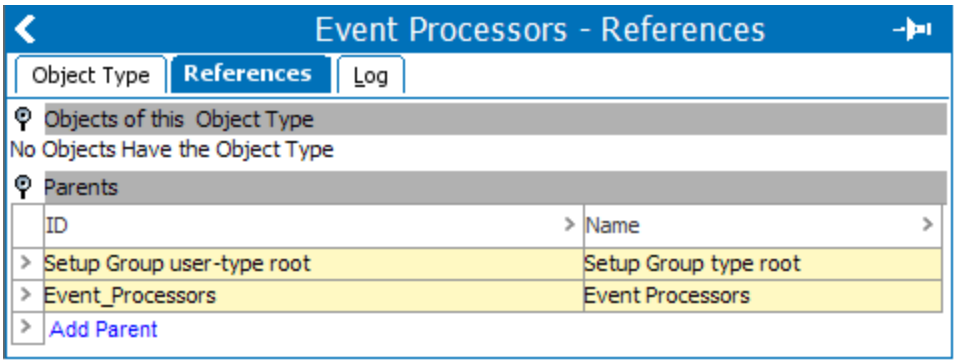
8. Select the newly added Setup Group type > References tab > open the Parents flipper.



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.

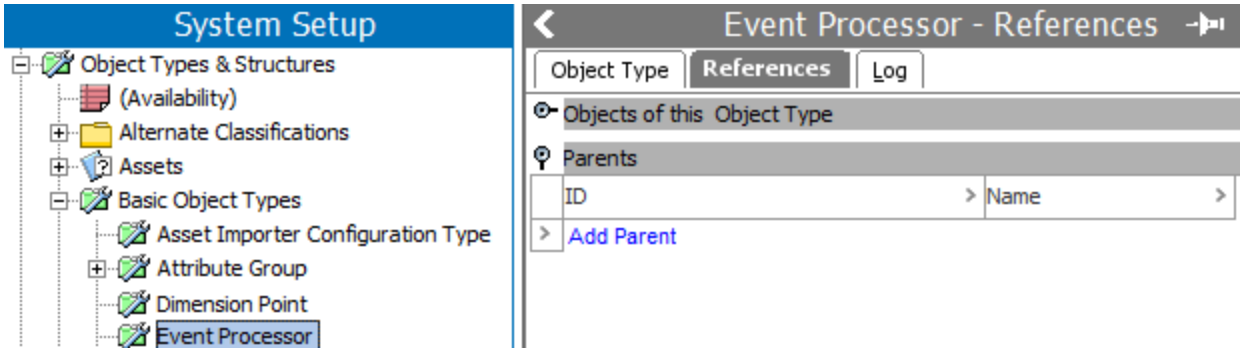
    The dialog will close, and the newly created setup group type (i.e., Event Processors) will be listed as a parent along with the Setup group user-type root.



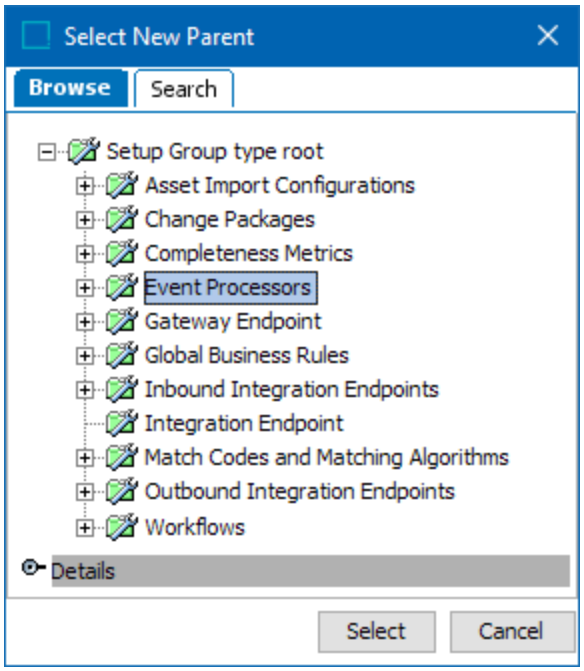
Link Event Processor Object Types to Setup Group Types

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.



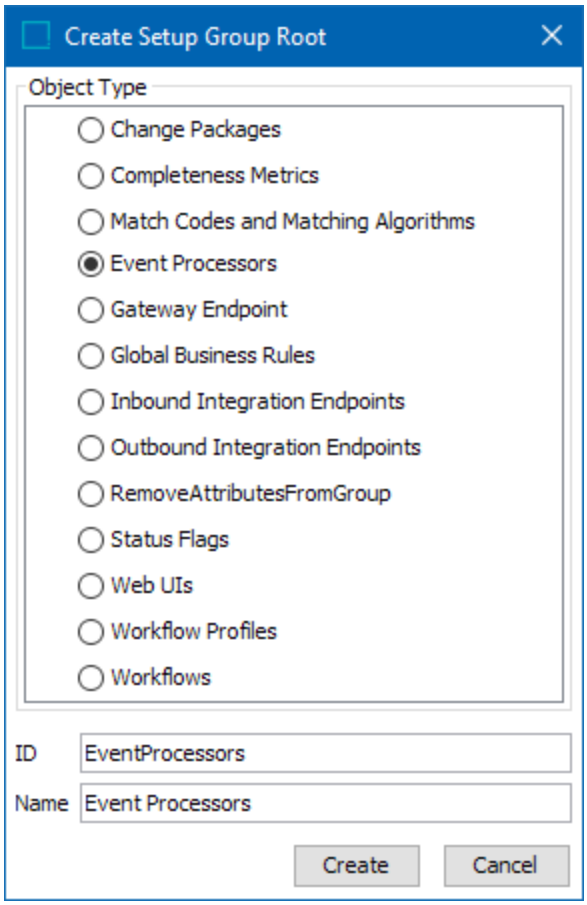
3. Click the **References** tab.
4. Open the **Parents** flipper.
5. Click the **Add Parent** link, and the Select New Parent dialog will display.
6. Browse or search to select **the relevant setup group type**.
7. Click the **Select** button.



Create an Event Processor Setup Group

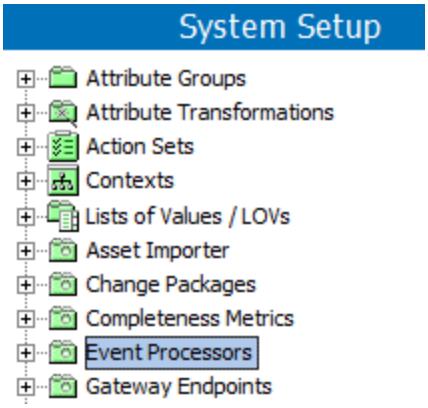
Creating a setup group allows your event processor setup group type to appear as a node in the System Setup hierarchy.

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.



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.



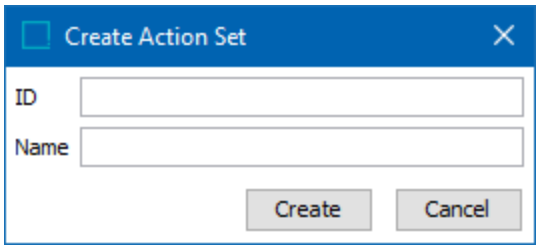
Configure Action Sets and Privileges

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**.



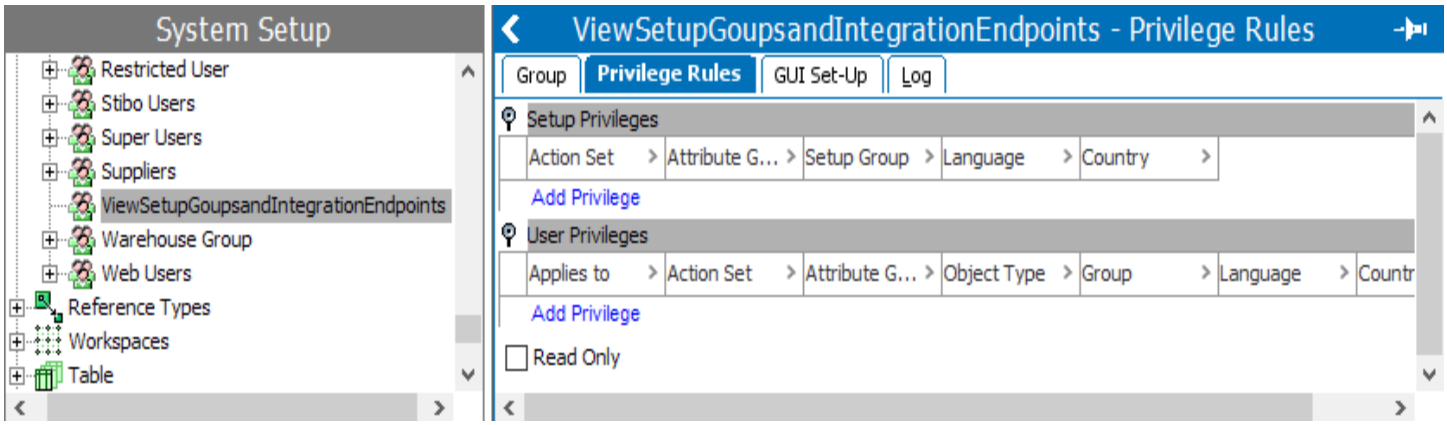
7. Go to the new Action Set > Action Set tab > click **Add Action**, and the Select Action dialog will display.
8. Select **one or more of the listed actions** that are relevant for event processors that you want to be part of this action set.
9. Click the **Select** button.

Some commonly used Actions are described below:

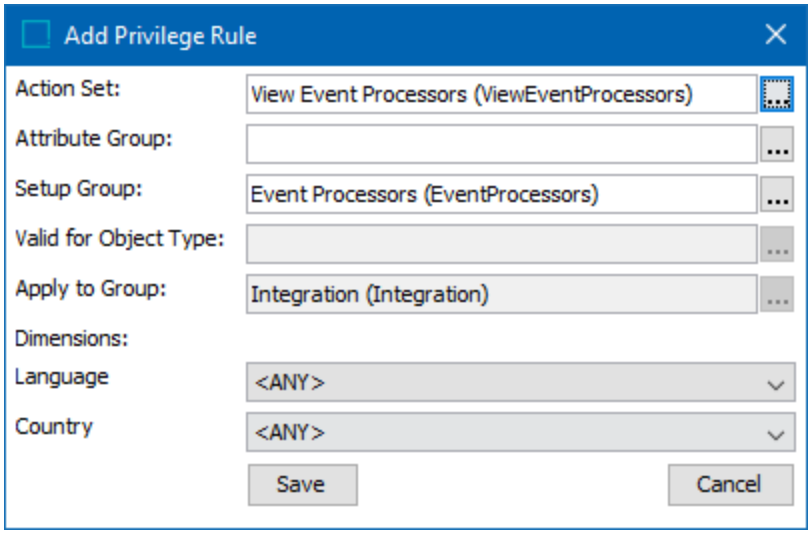
- **Maintain Event Processor** - Users can create, delete, and configure event processors. Users can also use the cut, copy, and paste actions.
- **Maintain Setup Group** - Users can create and delete setup groups. User can also use the cut, copy, and paste actions.
- **View Event Processor** - Users 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).
2. Click the **Privilege Rules** tab.



3. Under the Setup Privileges flipper, click the **Add Privilege** link to display the Add Privilege Rule dialog.



4. In Action Set, click the ellipsis button (...), and then select **the relevant event processor action set**.
5. In Setup Group, click the ellipsis button (...), and then select **the relevant event processor setup group**.
6. Click **Save**.

   For more information on privileges, see **Adding User Privileges for a Group** and for action sets see **Action Sets**.

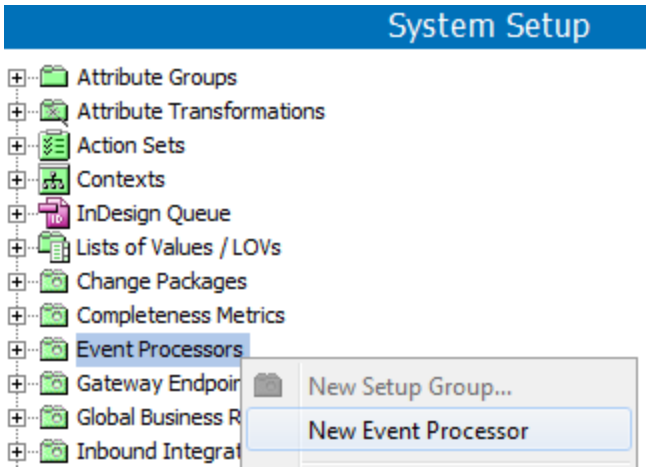
Creating an Event Processor

Prior to creating an event processor, the one-time setup tasks (as described in **Initial Setup for Event Processors** documentation) must be performed 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.

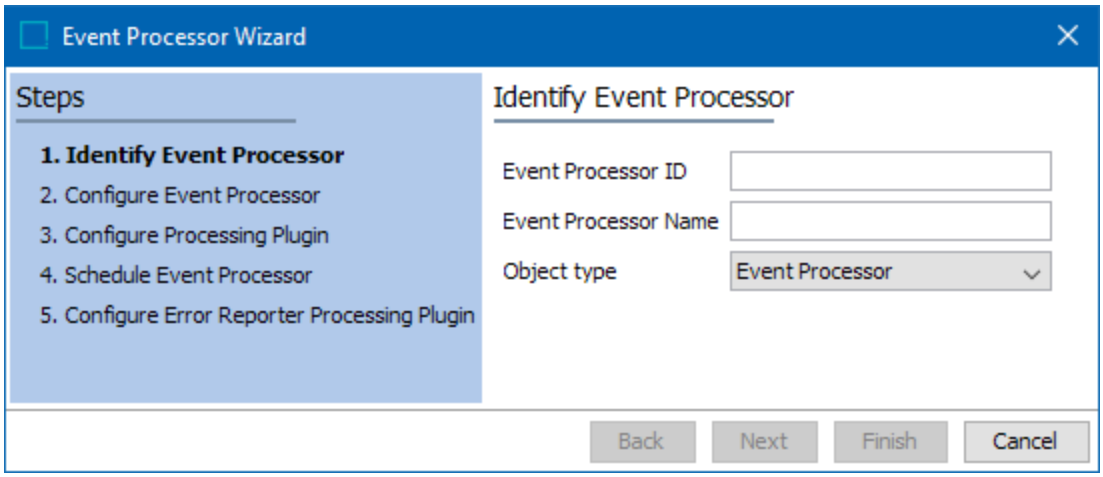
An event processor is created in the workbench System Setup using an event processor wizard. Below are general steps for creating a new event processor. However, the event processor wizard 'Configure Processing Plugin' dialog will vary based upon the processing plugin selected. Information for configuring 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:

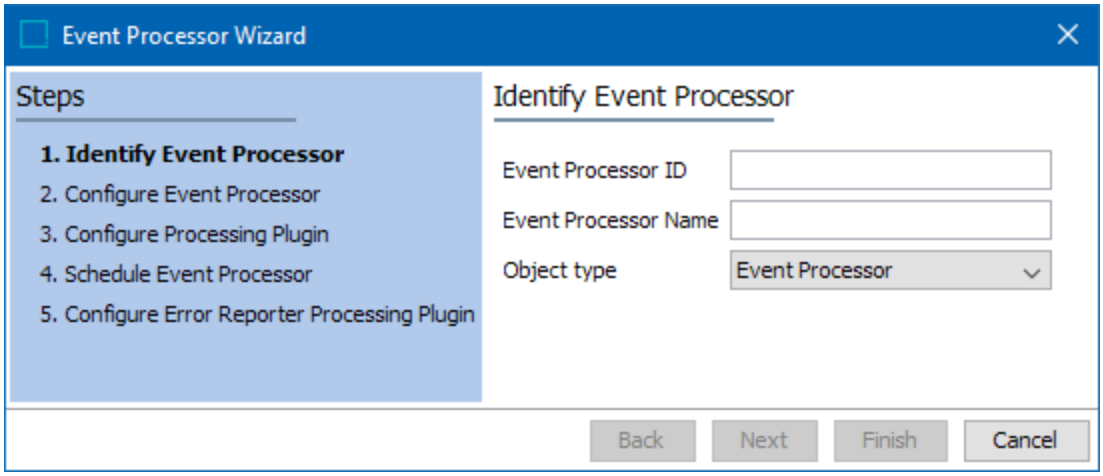


Complete setup requires providing data in the wizard as well as some manual configuration. All setup includes the following sections:

1. In the wizard, specify an ID, name, and object type as described in **EPW - Identify Event Processor** topic.
2. In the wizard, specify the processing and error handling as described in **EPW - Configure Event Processor** topic.
3. In the wizard, specify the information specific to the event processor selected in the previous step as described in **EPW - Configure Processing Plugin** topic.
4. In the wizard, specify how often the event processor should start as described in **EPW - Schedule Event Processor** topic.
5. In the wizard, populate the send report to address parameter as described in **EPW - Configure Error Reporter Processing Plugin** topic.
6. Confirm / Update triggering definitions as described in **Event Triggering Definitions Tab** topic.
7. Set Queue Status Events to 'Read Events' as described in the **Queue Status** section of the **Event Processor Tab** topic.
8. Enable the event processor as described in the **Enable Event Processor** section of the **Running an Event Processor** topic.

EPW - Identify Event Processor

To complete the Event Processor Wizard Identify Event Processor steps:

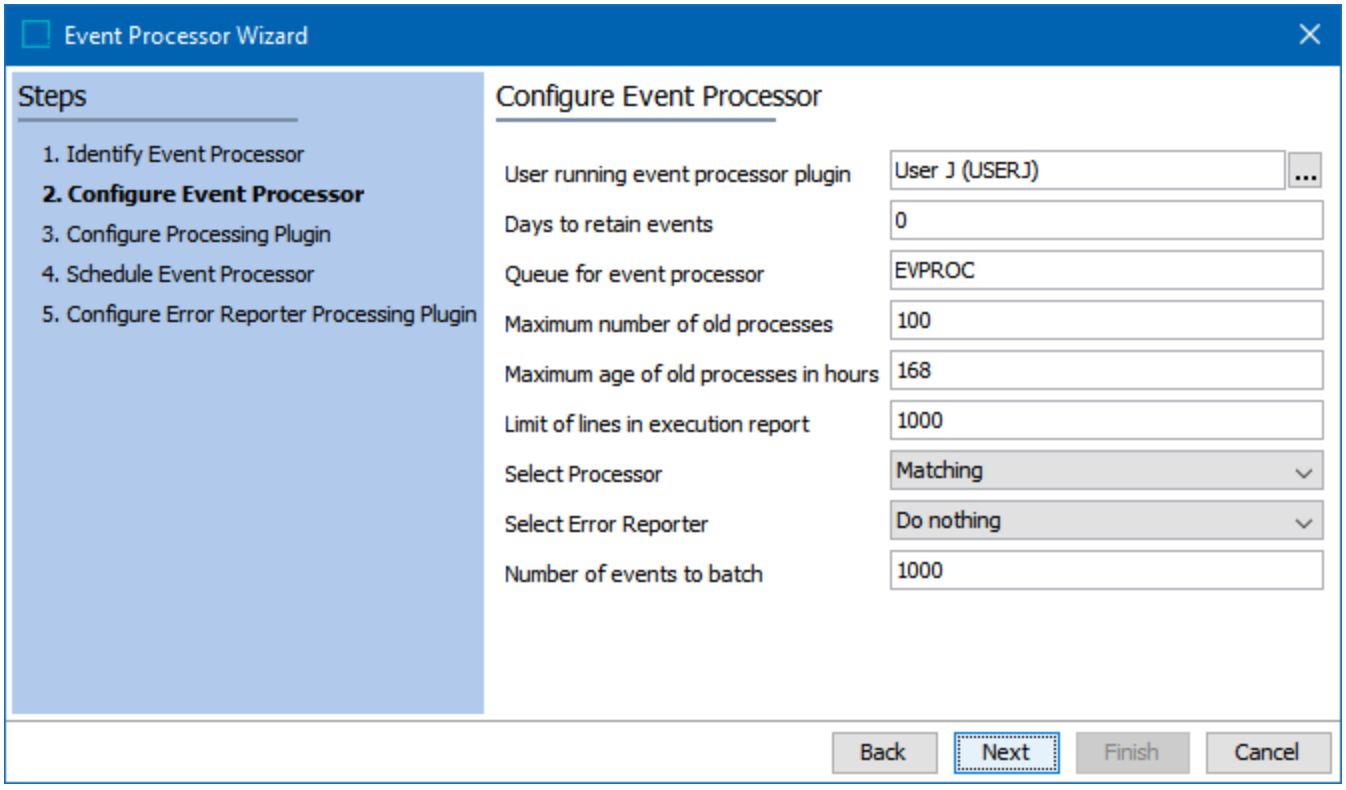


1. Enter an ID without spaces. (IDs should never contain spaces, especially leading or trailing spaces)
2. Enter a Name for the event processor. The name will be displayed as the name of the background process.
3. Confirm the object type parameter is set to the default event processor object type.
4. 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.



1. **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.

2. **Maximum age of old processes in hours:** Specify the maximum age of ended processes that the system will keep.
3. **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.

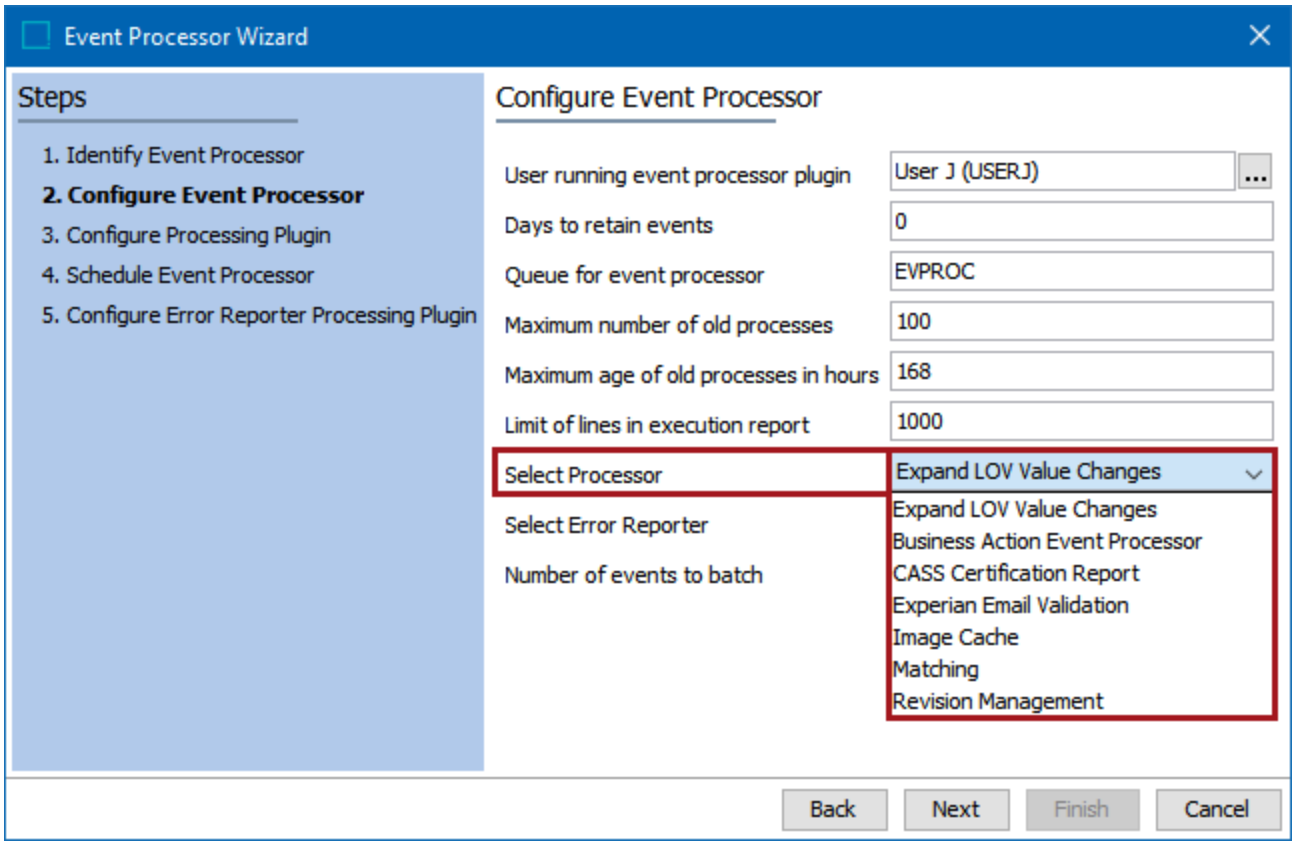
4. **Queue for event processor:** Enter the name of a queue to process data from the event processor. 'EVPROC' is the default, however entering the name of a new queue will result in the new queue automatically being created upon completion of the wizard.
5. **Maximum number of old processes:** Specify the number of ended processes the system will keep.
6. **Limit of lines in execution report:** Specify the maximum number of lines to store from the execution report in the log.
7. **Select Processor:** Choose from the available options, for more information, see the **Processing Plugins** documentation.
8. **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.
9. **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.
10. 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.

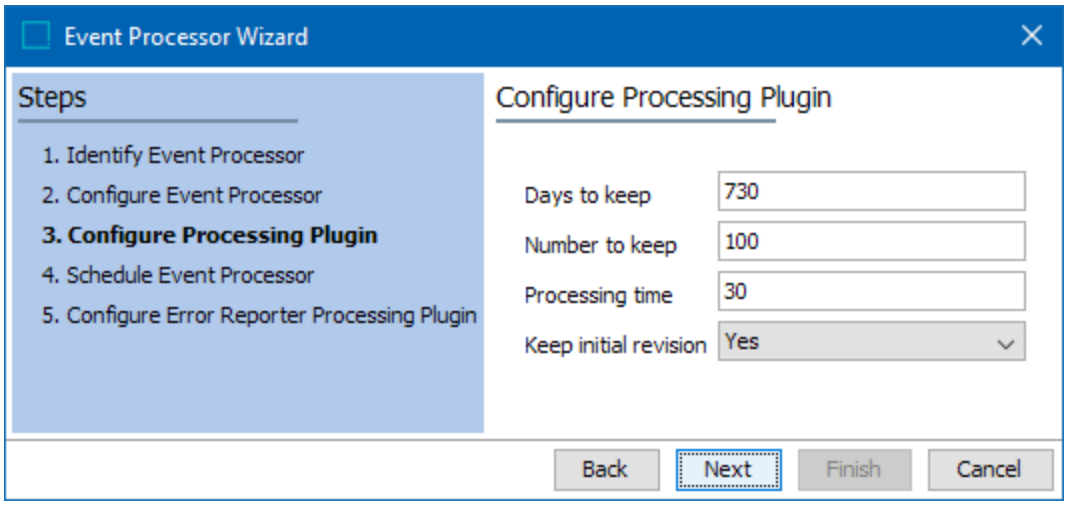
The options / parameters displayed within the Event Processor Wizard step 'Configure Processing Plugin' will vary depending on the processor selected during the previous step, 'Configure Event Processor.'

The screenshot below displays an example of the Event Processor Wizard step where the Select Processor parameter displays a dropdown list of available processors.



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.



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.

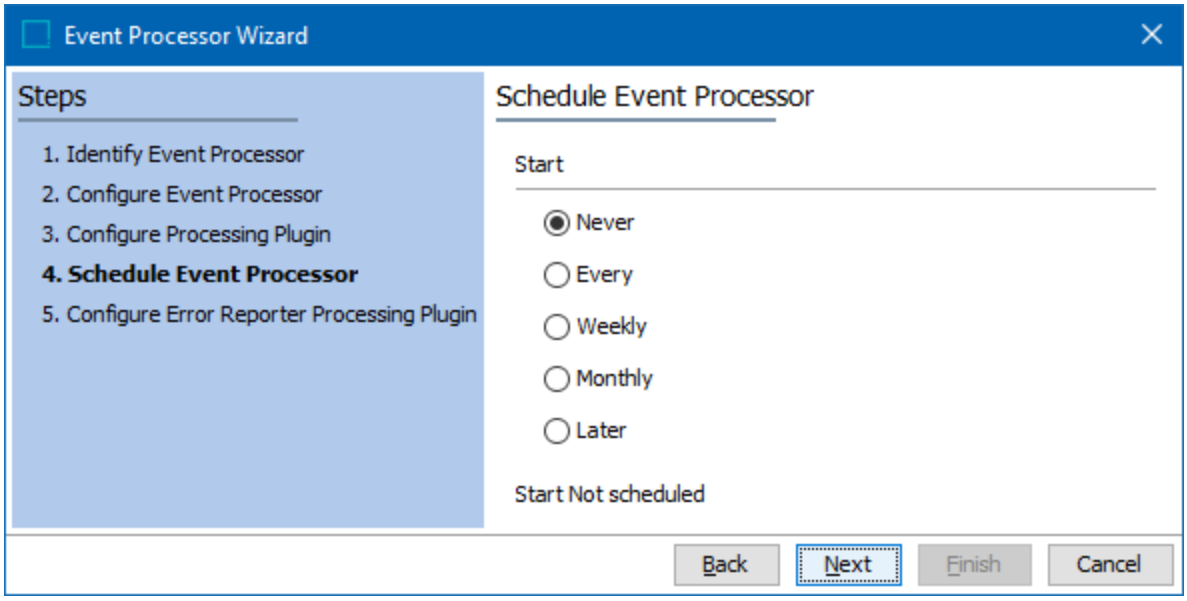
- **Asset Publisher Plugin Parameters and Triggers**
- **Asynchronous Translation Message Processor Processing Plugin Parameters and Triggers**
- **CASS Certification Report Processing Plugin Parameters and Triggers**
- **Data Sufficiency Calculator Processing Plugin Parameters and Triggers**
- **Elasticsearch Indexer Processing Plugin Parameters and Triggers**
- **Execute Business Action Processing Plugin Parameters and Triggers**
- **Execute Business Action for Event Batch Processing Plugin Parameters and Triggers**
- **Expand LOV Value Changes Processing Plugin Parameters and Triggers**
- **Experian Email Validation Processing Plugin Parameters and Triggers**
- **Historic Data Analyzer (within Historic Reporting)**
- **Historic Data Generator (within Historic Reporting)**
- **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.

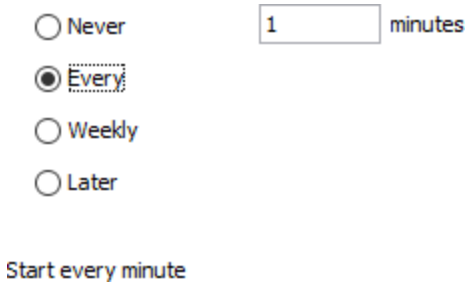
EPW - Schedule Event Processor

The wizard step 'Schedule Event Processor' will display with the following Start options.

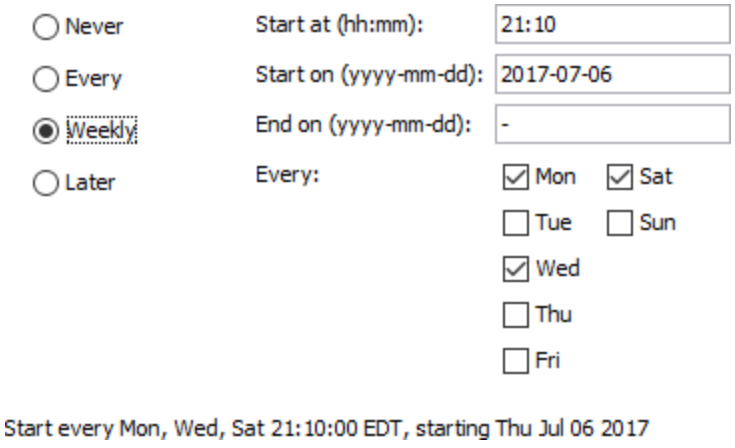


Important: Consider the time zone of the application server compared to that of the workbench (the client) where the schedule is created or viewed. When scheduling a job, the local time zone is displayed in the workbench, but the time zone of the server is used to run the background process. Although displayed, the time zone of the client is not included in the instruction to the server to run the job. This can cause confusion about when the job will run since the scheduled time is not automatically converted to accommodate potential differences in time zones.

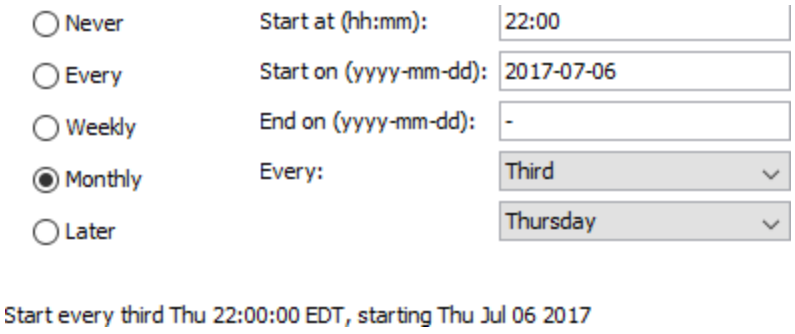
- **Never** - invoke the event processor manually, no additional parameters are required, and no schedule is applied. This is the default setting and should be used while testing your endpoint.
- **Every** - automatically run the event processor repeatedly, every selected number of minutes. One (1) minute 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.



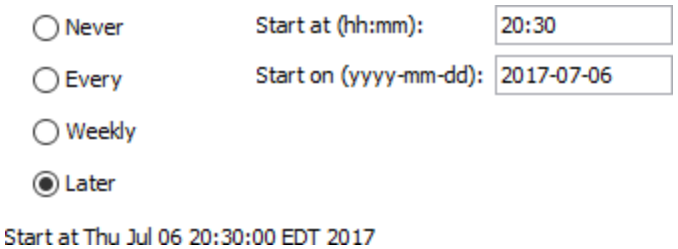
- **Weekly** - automatically run the endpoint repeatedly, based on the selected time, start and end dates, and days of the week. Use this option if a daily schedule is needed. The 'Start at' parameter determines the time of day that the endpoint will run. The 'Start on' parameter determines the date 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.



- **Monthly** - automatically run the event processor repeatedly, once a month, based on the selected time, start and end dates, week of the month, and day of the week. The 'Start at' parameter determines the time of day that the event processor will run. The 'Start on' parameter determines the date the event processor will first run, while the 'End on' parameter determines the date of the event processor's final run. The 'Every' dropdown parameter selections for the week of the month and the day of the week determine when the event processor will run. The selections are summarized at the bottom of the dialog.



- **Later** - automatically run the event processor only once, at the time and date specified. The selections are summarized at the bottom of the dialog.



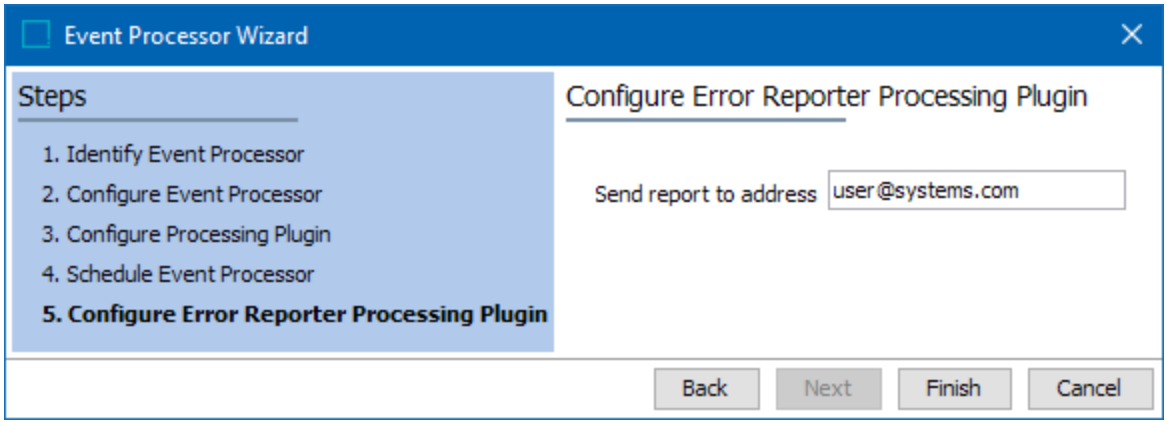
Important: Selecting a 'Schedule Event Processor' setting will **not** enable / start the event processor. Additional steps are required after completing the wizard to enable / start the event processor. The selection within this step can be edited later. Choosing an option will set the schedule for 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.



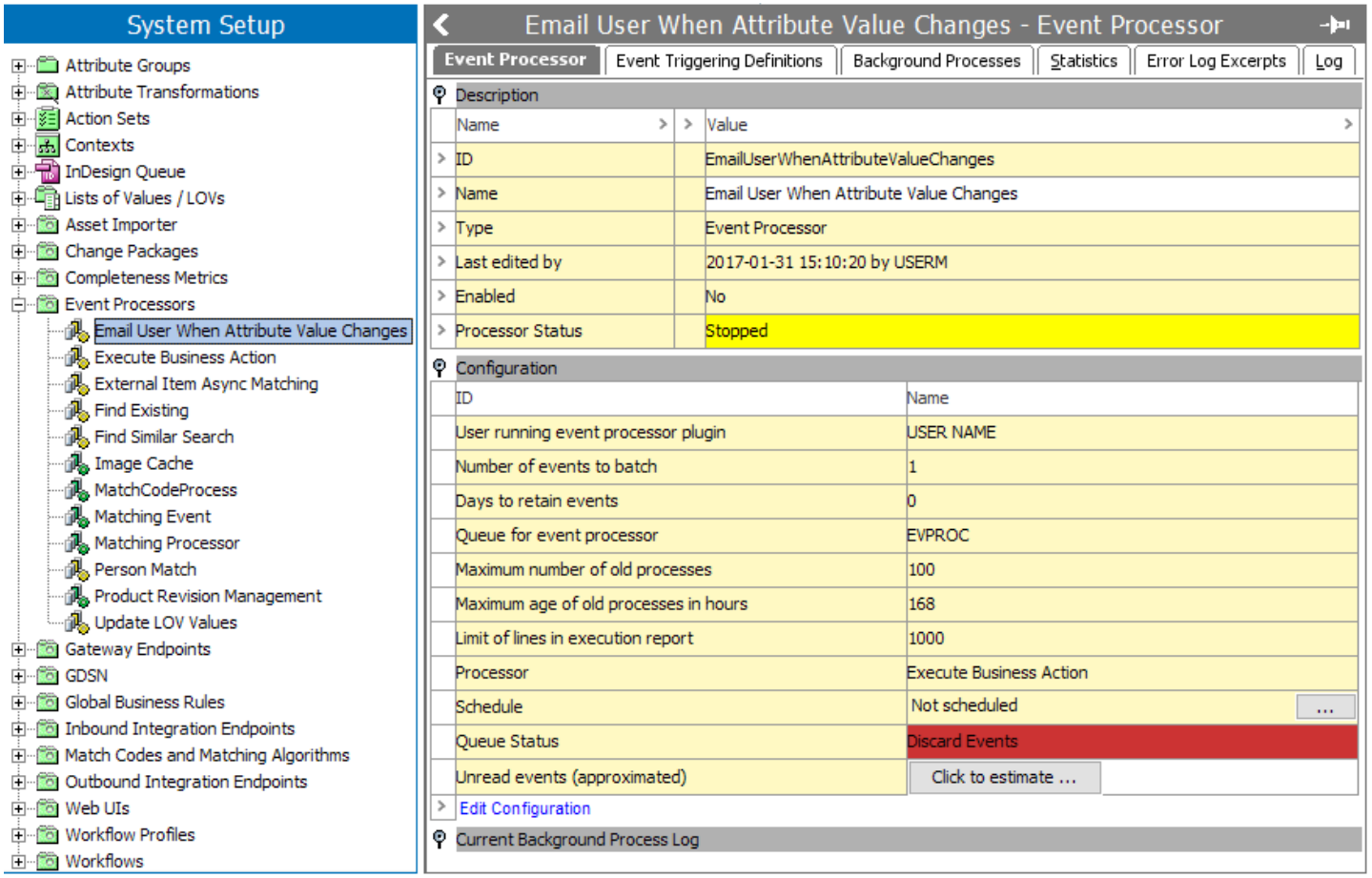
Note: If the 'Select Error Reporter' option was not set to 'Mail Error Report,' then the 'Send report to address' option will not display. Use the Back button to return to the 'Configure Event Processor' step, and edit the selection.

- **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.

  For the Error reporter plugin to send email, the SMTP server must be configured on the application server.

  For information on configuring email from STEP, see the **Email from STEP** topic in the **Resource Materials** of online help.

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.

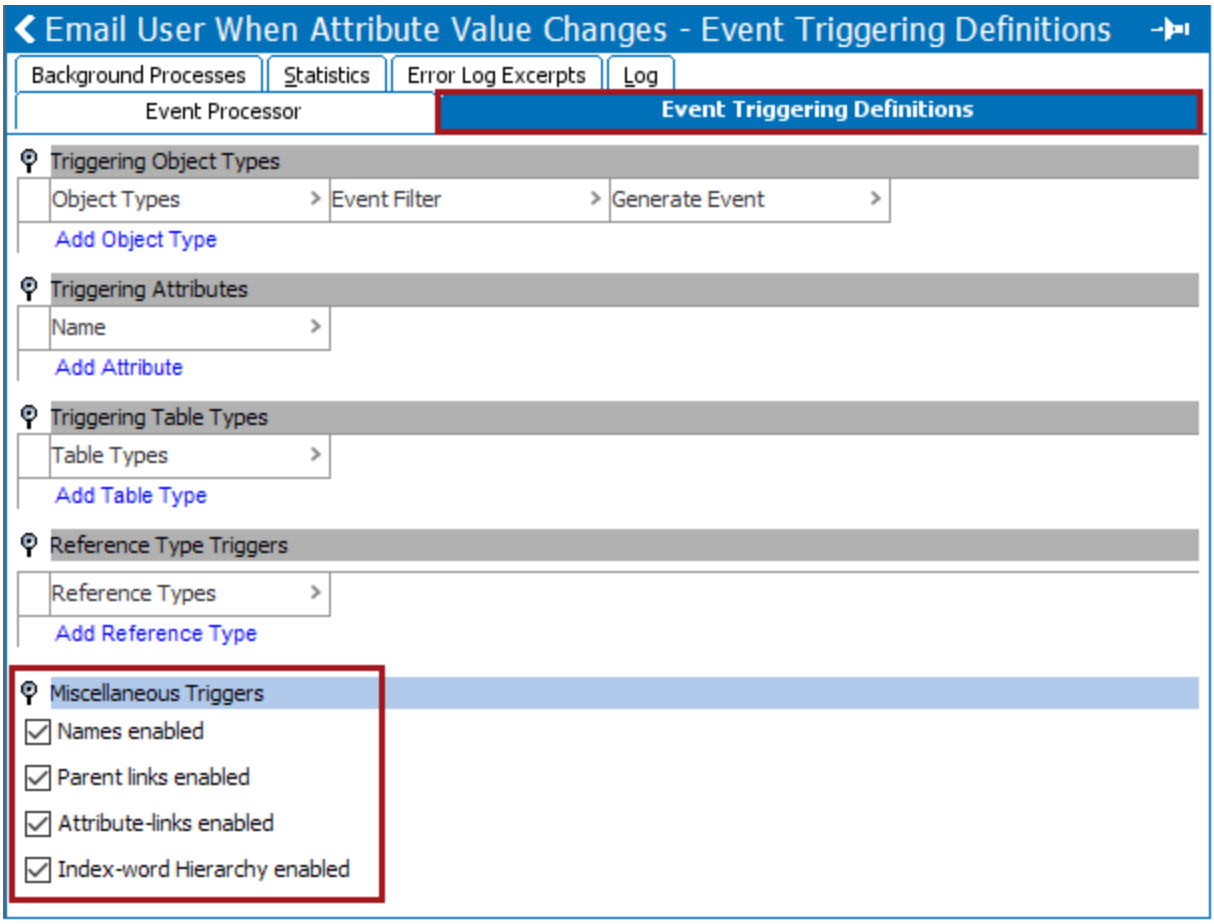


No matter the selections made during the wizard, new event processors are created with the following 'Event Processor' tab parameter settings:

- **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



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. Event processors run in optimistic mode, meaning that savepoints will not be created, thus decreasing execution time and increasing performance.

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

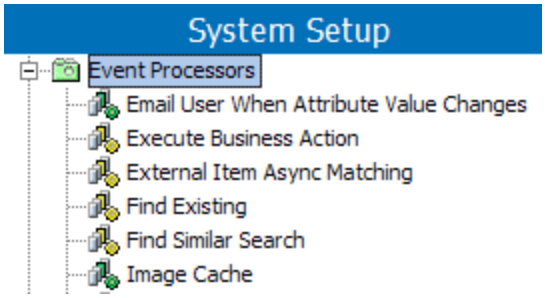
An event processor can be in one of several statuses: Enabled, Disabled, and Stopped.

The table below displays each icon and a description of each event processor status.

Icon	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 **EP - Event Processor Tab** documentation.

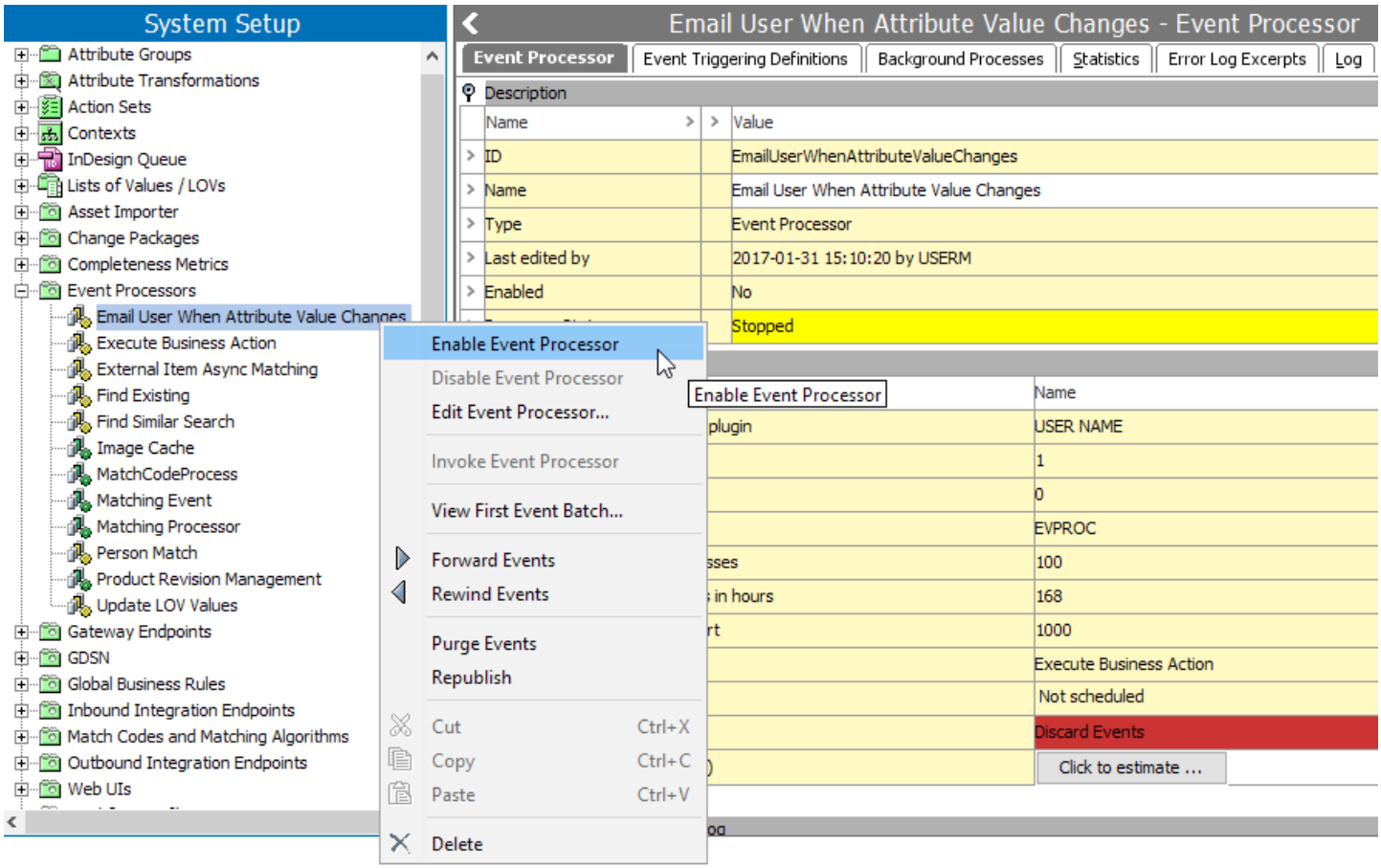
When configuring and managing event processors, it is important to monitor their status, because an event processor only functions when it is enabled.

Enable Event Processor

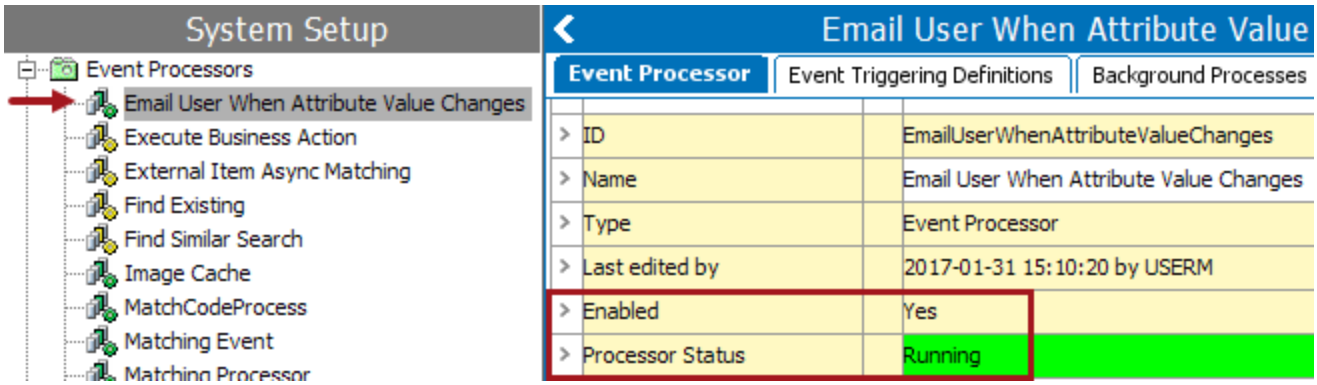
When an event processor is created, it is in a stopped or disabled status, and must be enabled to be of use.

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**.



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.



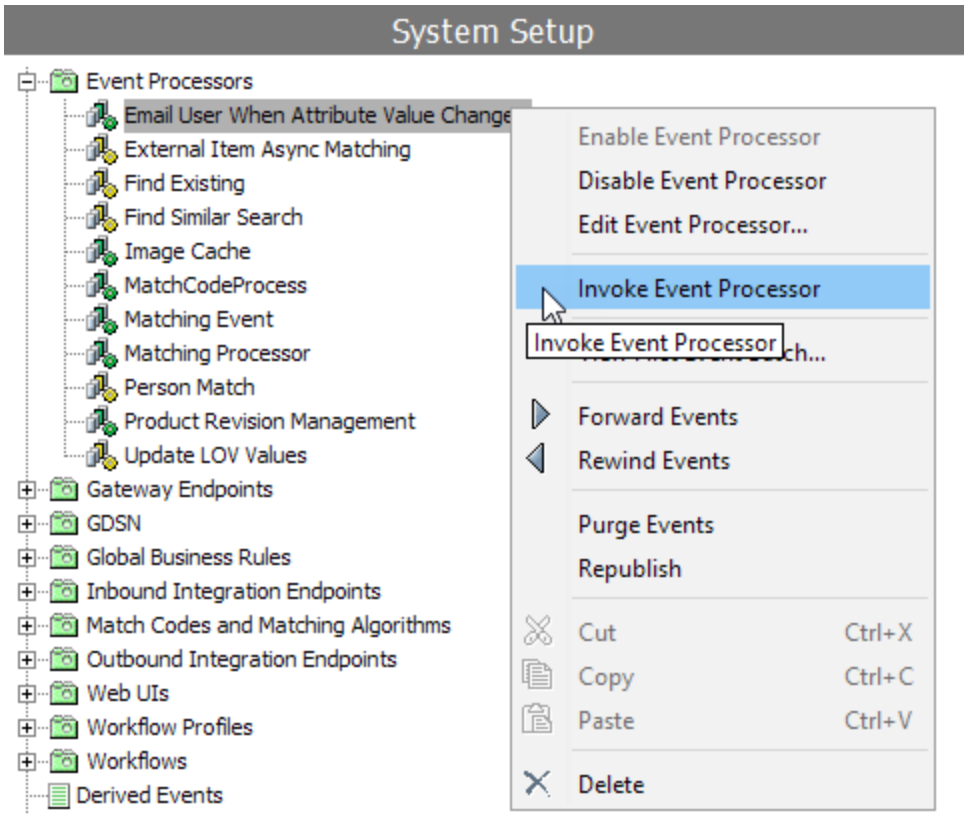
Important: If you are ready for the processor to apply the actions based on the events processed, make sure the Queue Status parameter is also set to Read Events, otherwise the enabled event processor will not function. For more information see the **Queue Status** section of the **EP - Event Processor Tab** topic.

Invoke Event Processor

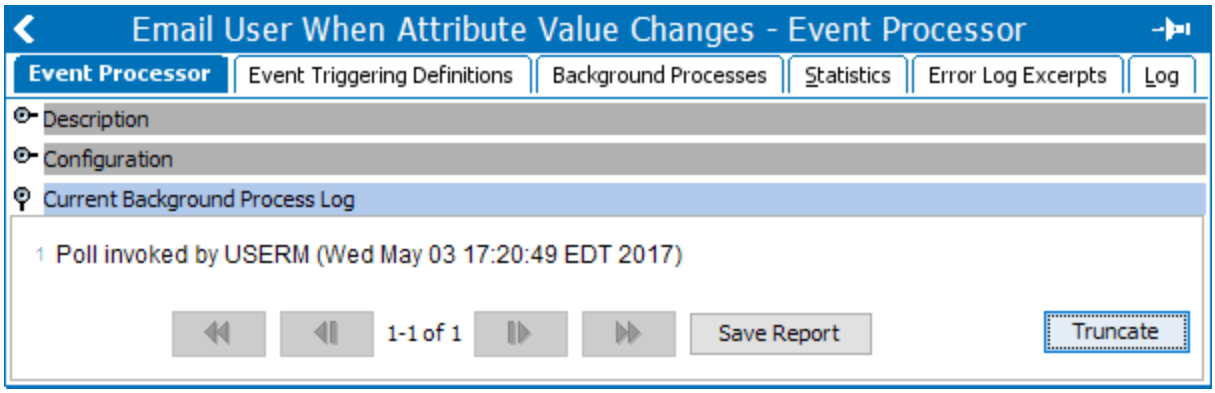
An event processor can only be invoked if its status is Enabled.

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)'.

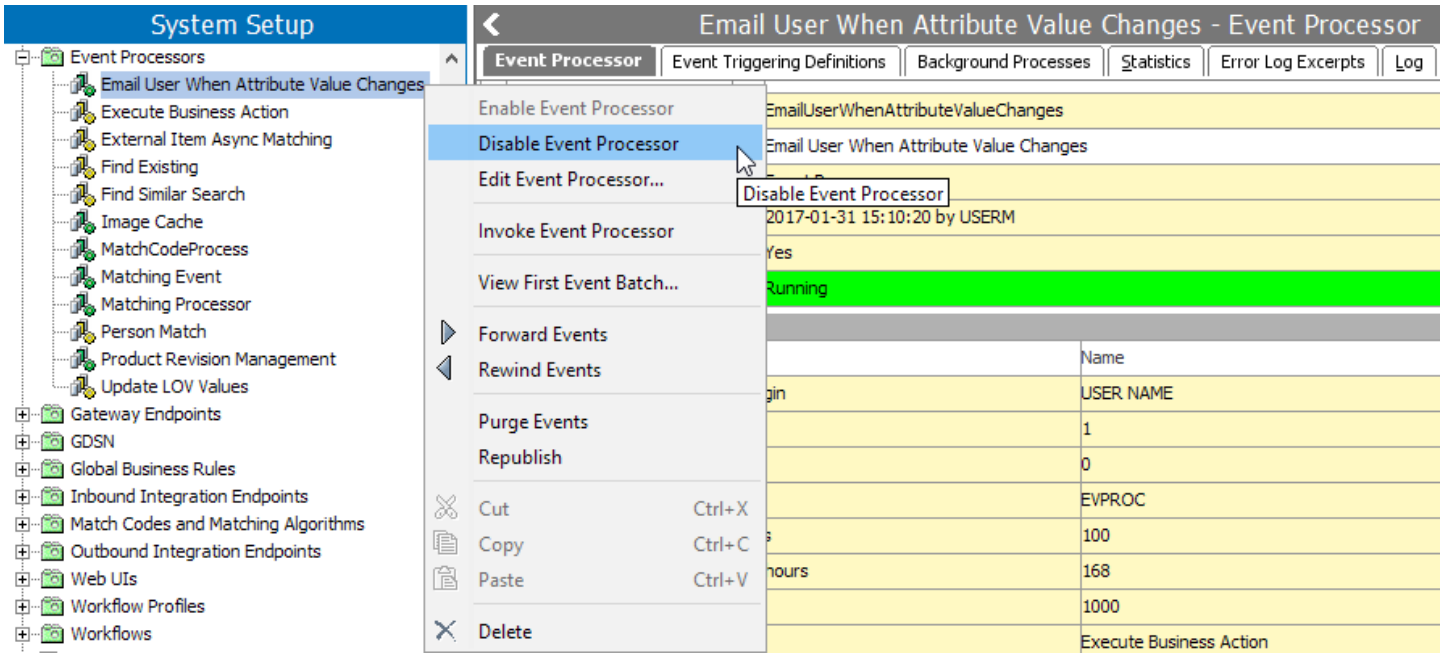


Disable Event Processor

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.

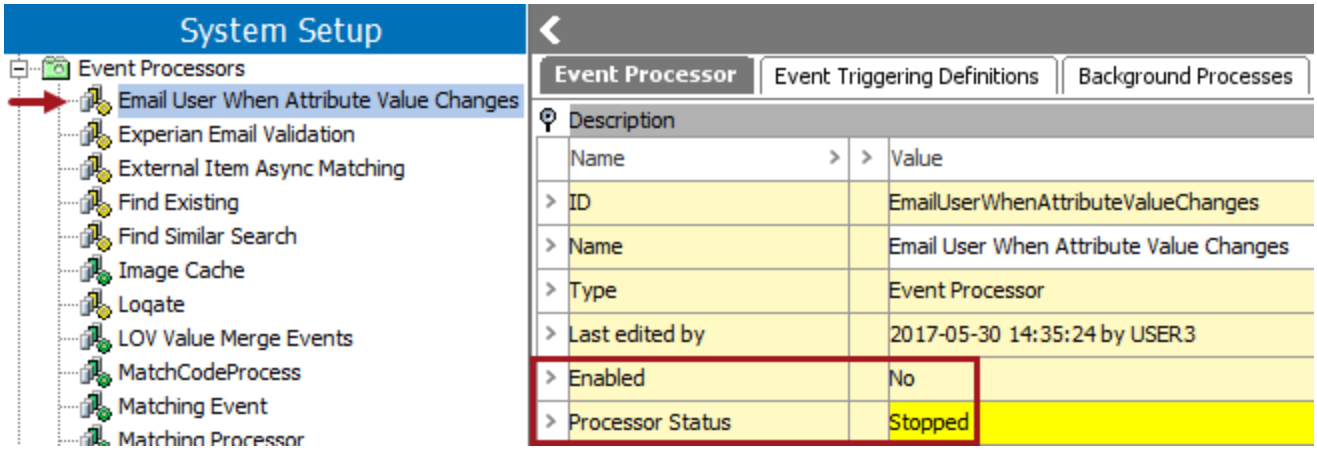
The steps below describe how to disable an event processor in an enabled status.

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**.



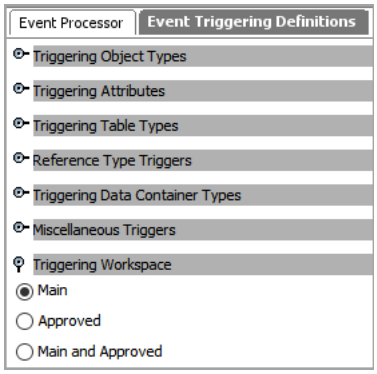
3. Confirm the event processor is now disabled by viewing the:

   - 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.



Generating Events from Main

By default, approval triggers an event for workspace revisable objects, meaning that from the Main workspace saving an edit (but not approving it) does not generate an event. To change the triggering workspace, go to the Event Trigger Definitions tab and change your selection under the Triggering Workspace flipper.



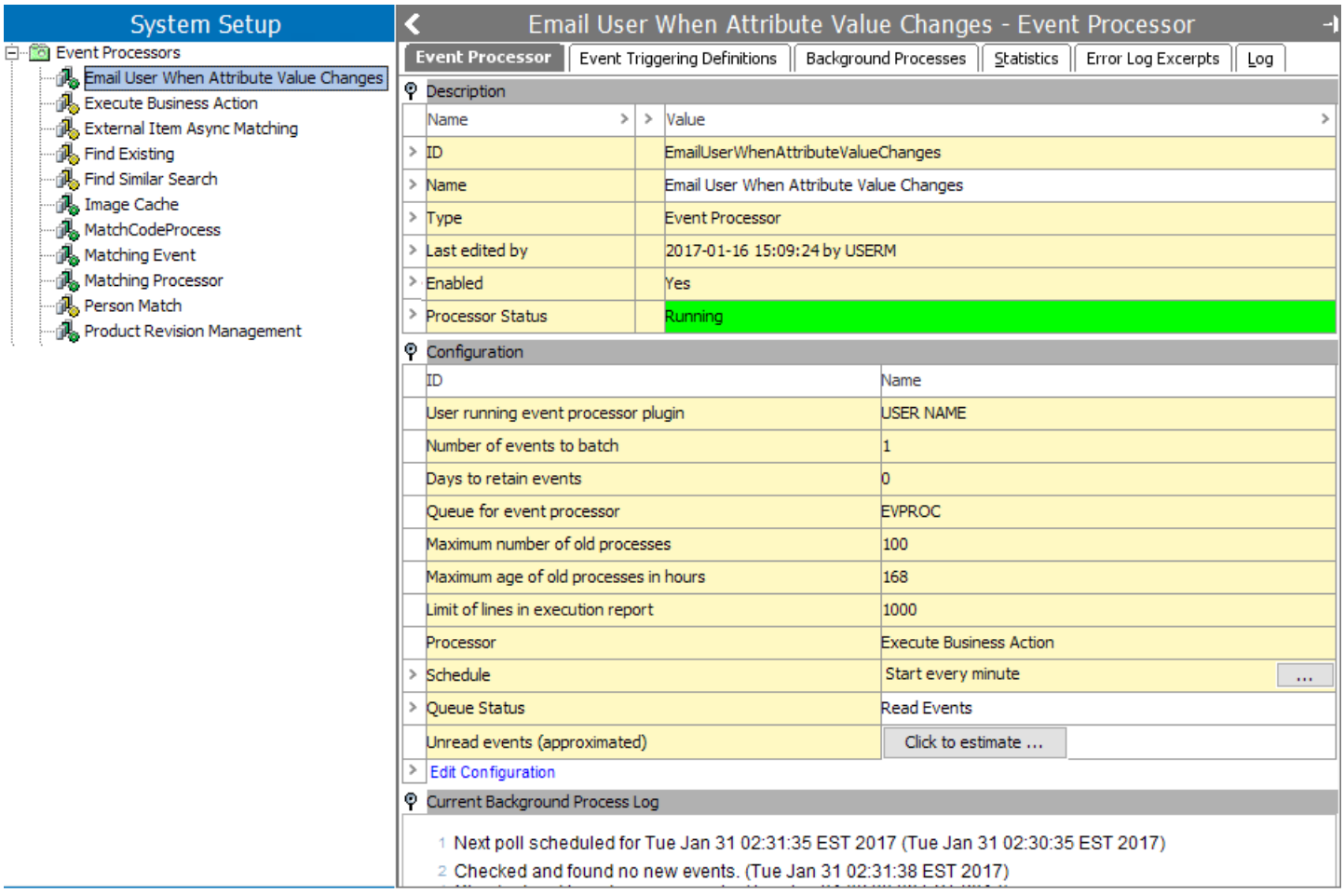
Maintaining an Event Processor

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



Edit Event Processor

There are many reasons and methods to edit an event processor.

Steps to edit an event processor are below.

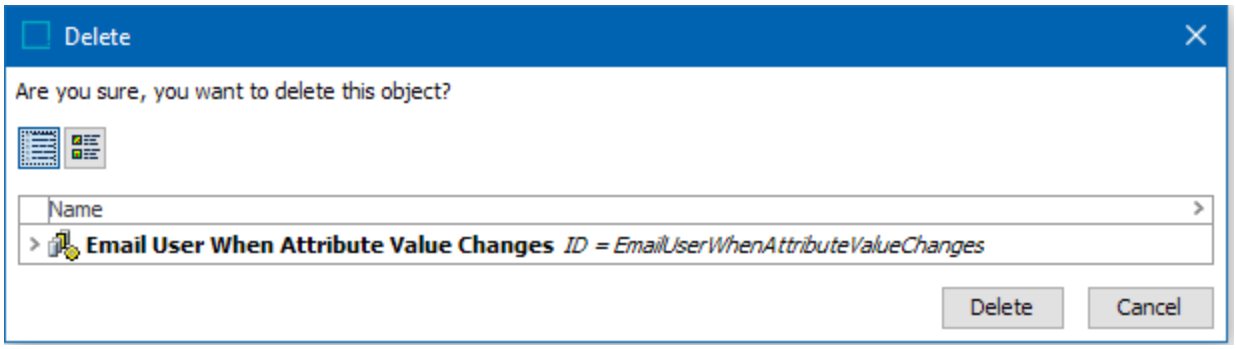
1. On the System Setup tab, select **the event processor to be edited**.
2. Choose a method to edit the wizard settings displayed in the Event Processor editor > Event Processor tab > Configuration flipper:
   - Right-click the event processor and select **Edit Event Processor**.
   - Expand the Configuration 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.

EP - Event Processor Tab

The Event Processor tab on an event processor contains information about the description, configuration, and current background process log.

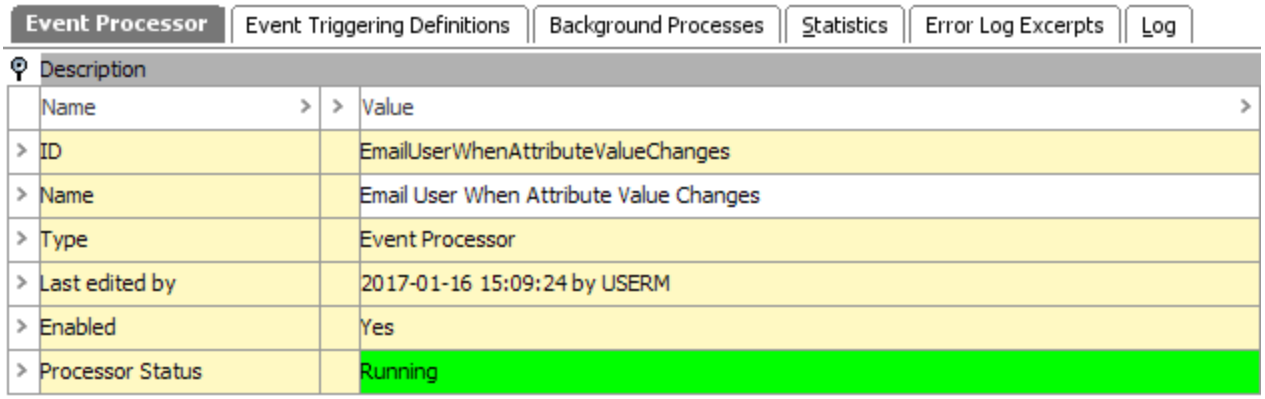
Prerequisites

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.

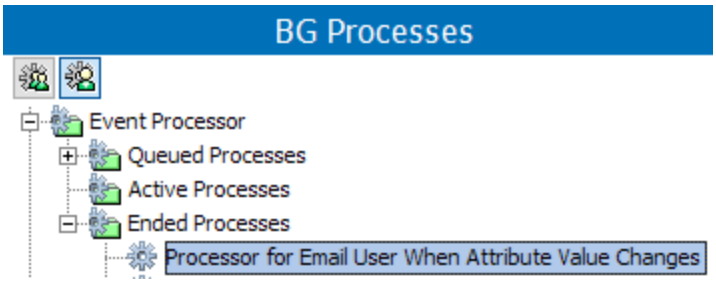


ID

- 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

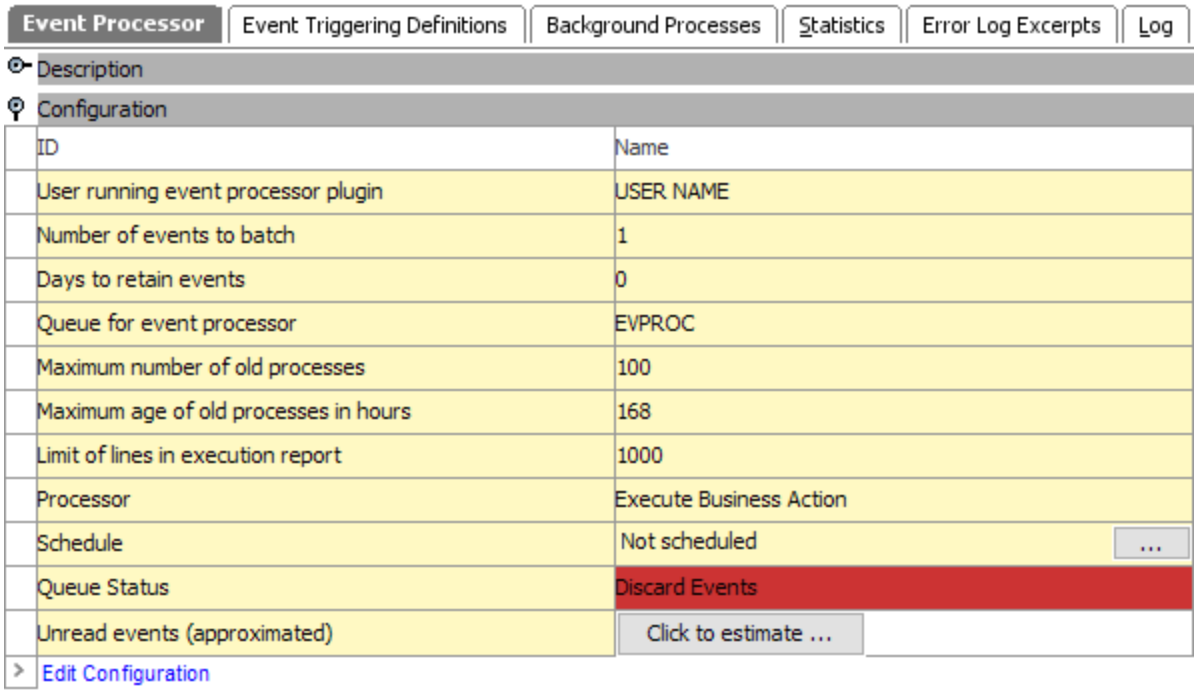
- Displays the current event processor state.
- When enabled the parameter displays 'Yes', the Processor Status parameter automatically displays 'Running'.
- When disabled the parameter displays 'No', the Processor Status parameter automatically displays 'Stopped'.
- A processor must be enabled for it 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.



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

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.

Use the event processor editor to update the **Queue Status** as follows:

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.

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.

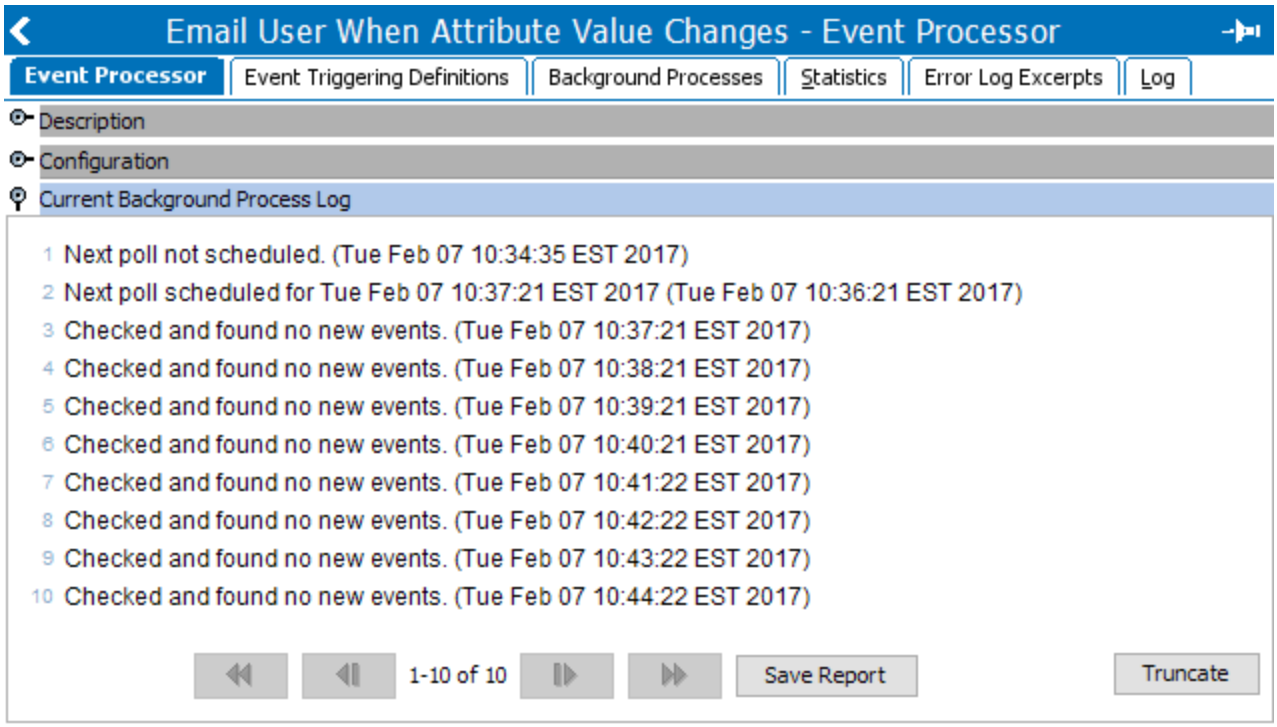
Unread events (approximated)

- 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

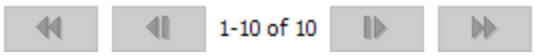
The Current Background Process Log flipper is blank until the event processor state is enabled. Once enabled, information about when the event processor is expected to run and/or what occurred when the processor ran will display.

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'.



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

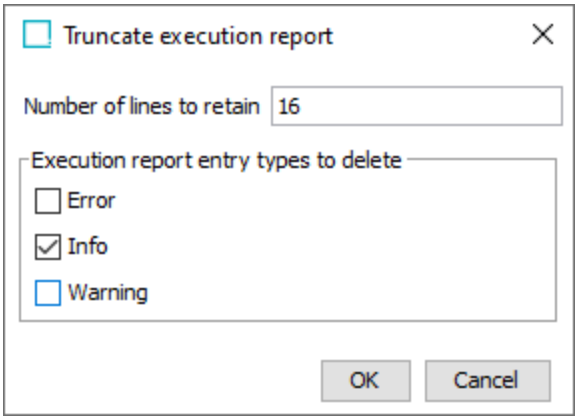
The Current Background Process Log can be saved as an 'HTML Only' file type.

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:



- Number of lines to retain
  - By default, the current number of log entries is displayed, meaning all will be kept.
  - Enter the number of most recent lines to keep from the end of the log, in order to only save the last 'X' number of lines. This works in conjunction with the 'Execution report entry types to delete' option below.
- Execution report entry types to delete
  - The entry types that can be deleted are Error, Info, and Warning.
  - By default 'Info' is selected, meaning only Info lines will be removed based on the lines to retain.
  - 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.

EP - Event Triggering Definitions Tab

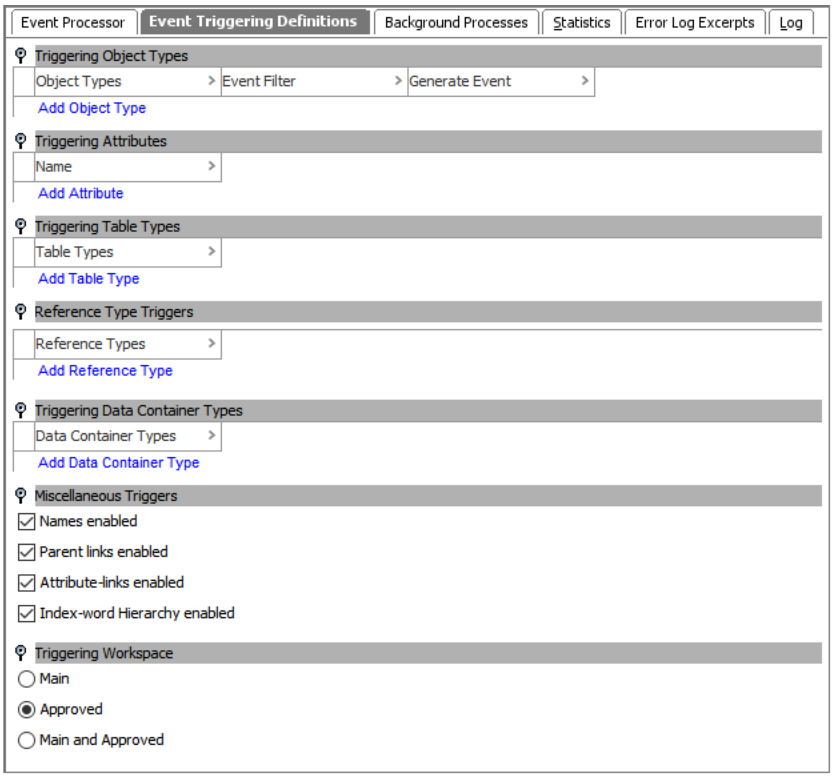
The Event Triggering Definitions Tab on an event processor contains information about the configuration to be set to trigger events as per the requirement.

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.

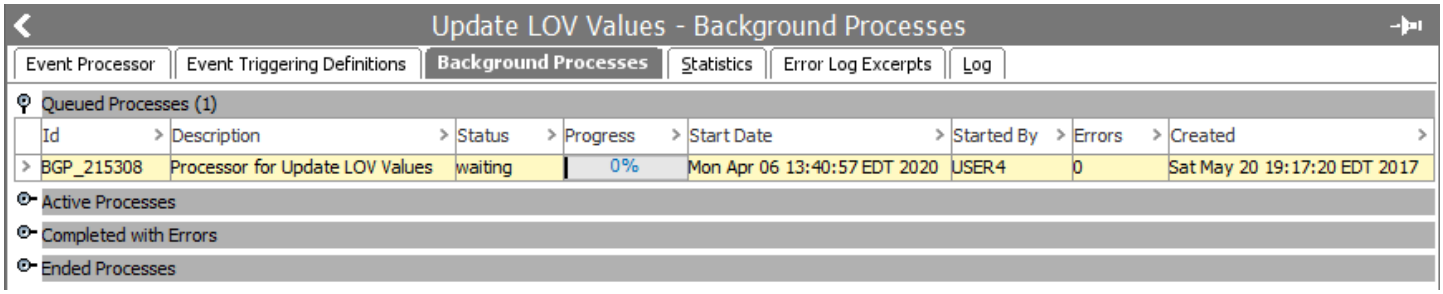
To help streamline setup for event processors using processing plugins, the information for common parameters and triggering definition setups for each of the processing plugins is covered in the following sections:

- **Asset Publisher Plugin Parameters and Triggers**
- **Asynchronous Translation Message Processor Processing Plugin Parameters and Triggers**
- **CASS Certification Report Processing Plugin Parameters and Triggers**
- **Data Sufficiency Calculator Processing Plugin Parameters and Triggers**
- **Elasticsearch Indexer Processing Plugin Parameters and Triggers**
- **Execute Business Action Processing Plugin Parameters and Triggers**
- **Execute Business Action for Event Batch Processing Plugin Parameters and Triggers**
- **Expand LOV Value Changes Processing Plugin Parameters and Triggers**
- **Experian Email Validation Processing Plugin Parameters and Triggers**
- **Historic Data Analyzer (within Historic Reporting)**
- **Historic Data Generator (within Historic Reporting)**
- **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 **Queue Status** section of the **EP - Event Processor Tab** topic.

EP - Background Processes Tab

An active event processor uses an associated background process to handle the scheduled invocation of the processor. The event processor's Background Processes tab contains information about the available processes. Hover over a BGP ID to display the background process link, click it to open the individual background process.

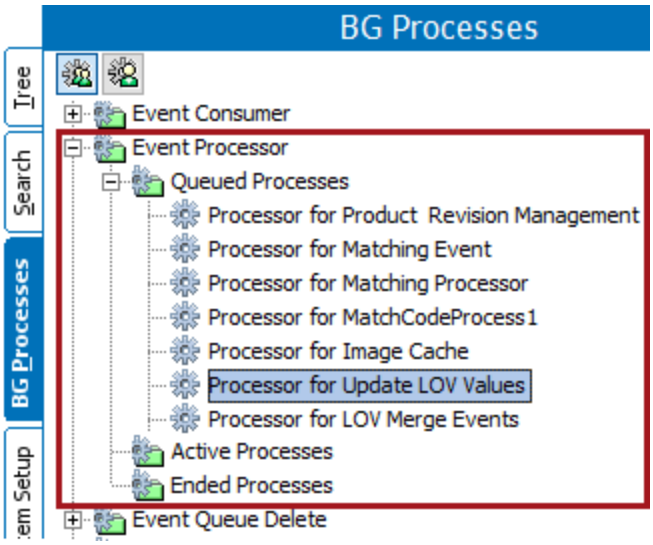


Note: The functionality of the event processor Background Processes tab is the same for integration endpoints.

The BGP information related to the selected event processor includes:

- **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.

This information is also available from the BG Processes tab under the Event Processor node when selecting a processor.



For more information, see the following topics:

- The event processor editor is defined in the **Maintaining an Event Processor** topic.
- Background process states are defined in the **BG Processes States** topic in the **System Setup / Super User Guide** documentation.
- Background processes are defined in the **Background Processes and Queues** topic in the **System Setup / Super User Guide** documentation.

EP - Statistics Tab

The Statistics tab on an event processor 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.

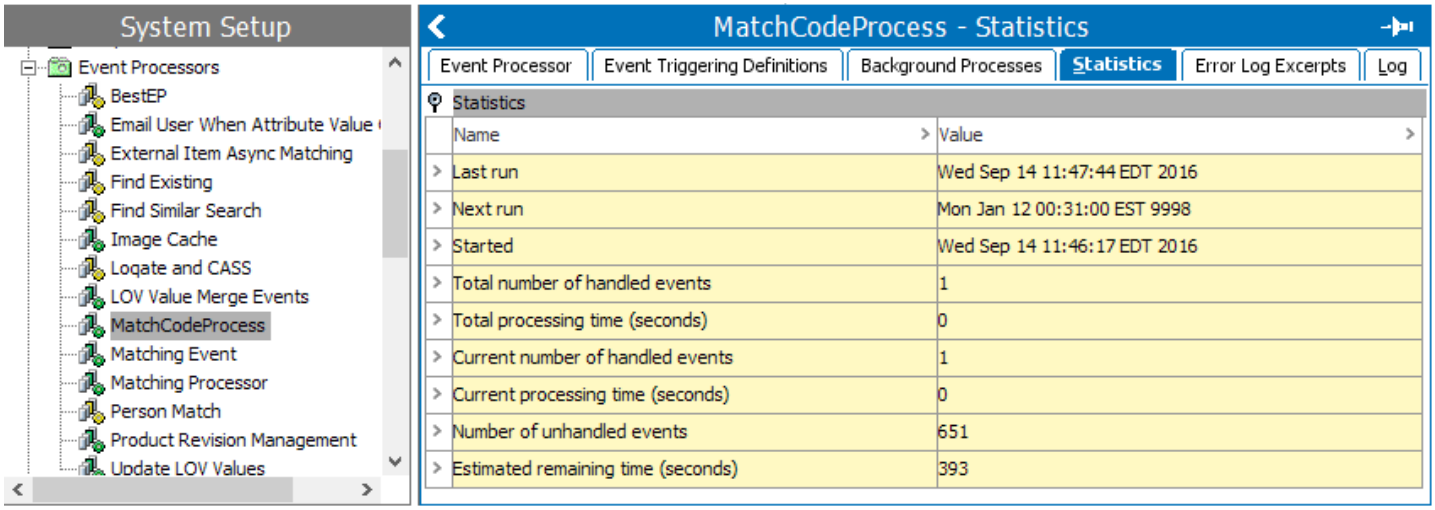
Prerequisites

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)



EP - Error Log Excerpts Tab

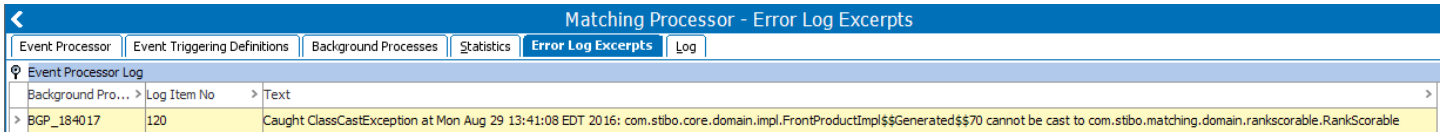
The Error Log Excerpts tab on an event processor shows data from the main Java log file related to failed background processes with Log Level > 'Info'.

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.



- **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.

EP - Log Tab

The Log tab on an event processor provides information about changes to the Event Processor configuration.

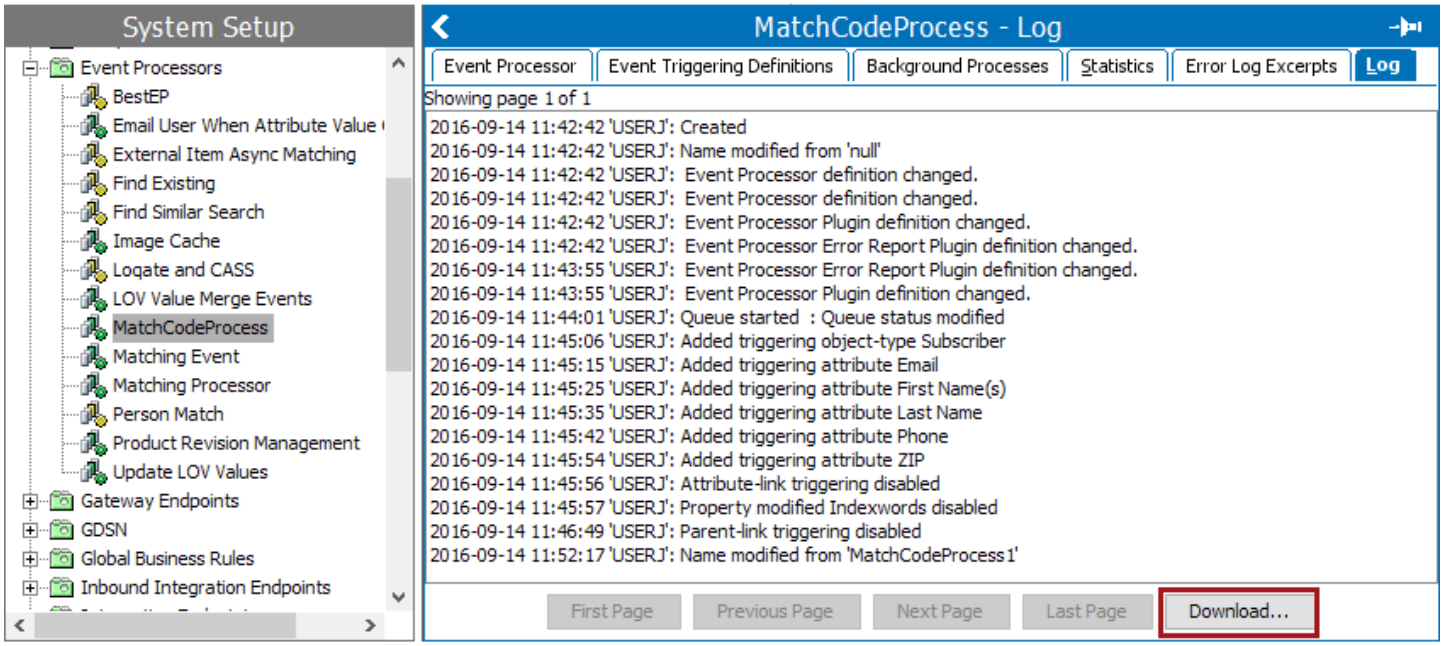
Prerequisites

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.

Click the Download button to download the log information.



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, an estimate and details are available for review in System Setup.

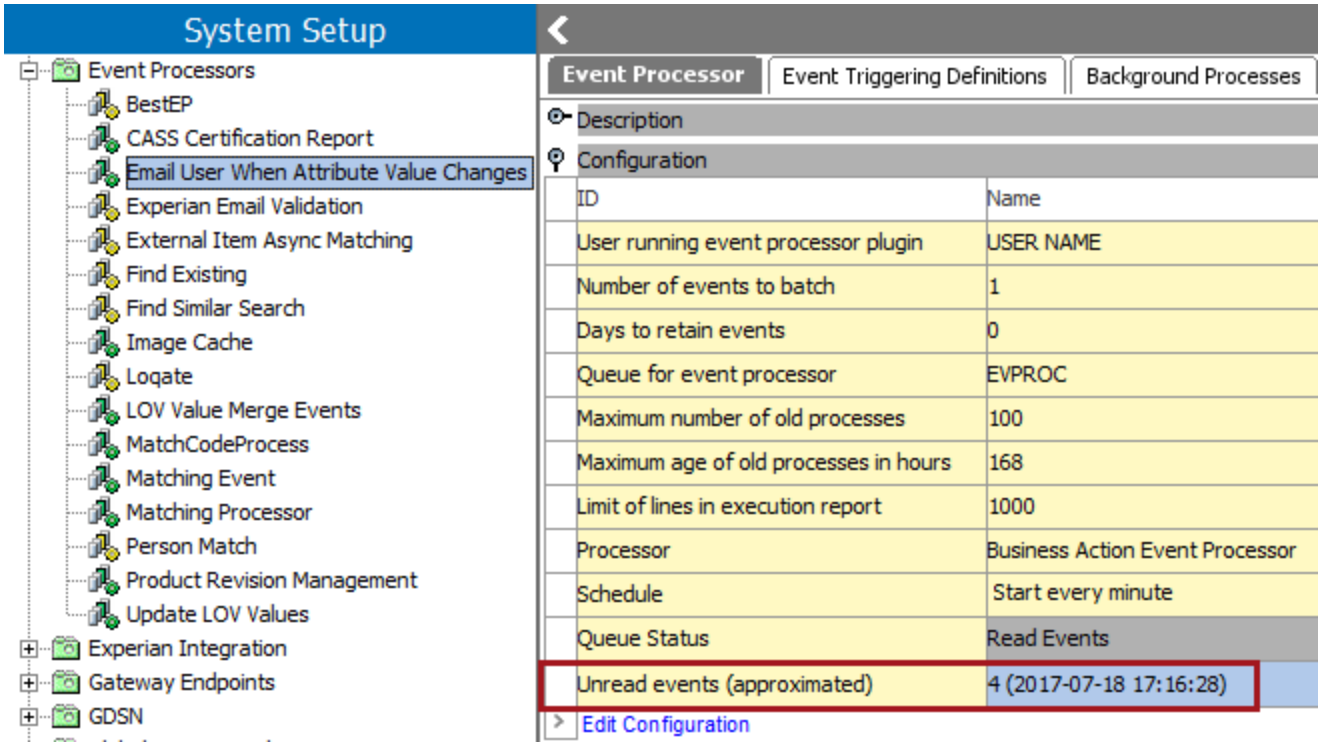
Estimate Unread Events

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.

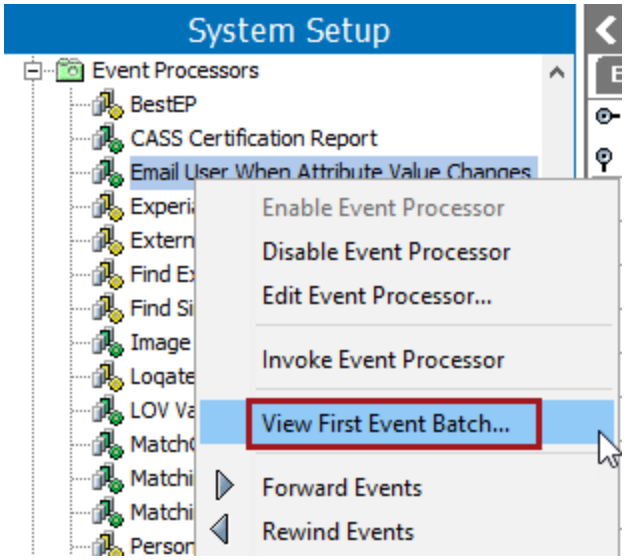


Using the forward, rewind, purge, or republish events options causes the number of unread events on the queue to change. For more information, see **Event-Based OIEP Forward, Rewind, Purge, and Republish**.

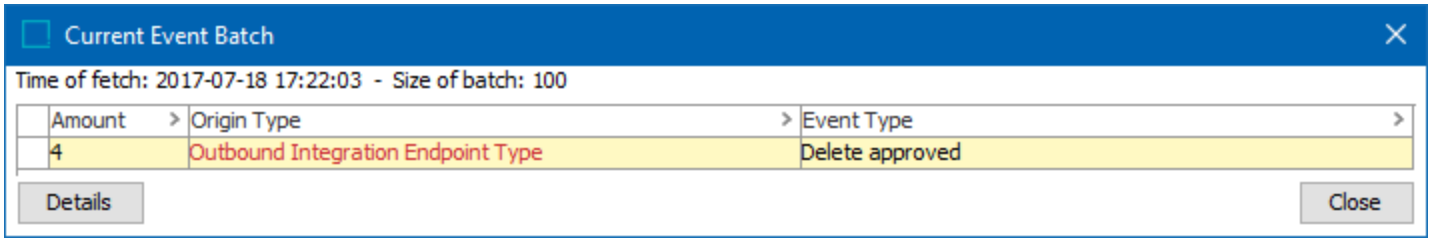
View First Event Batch

This option allows you to view more than just a count of the events.

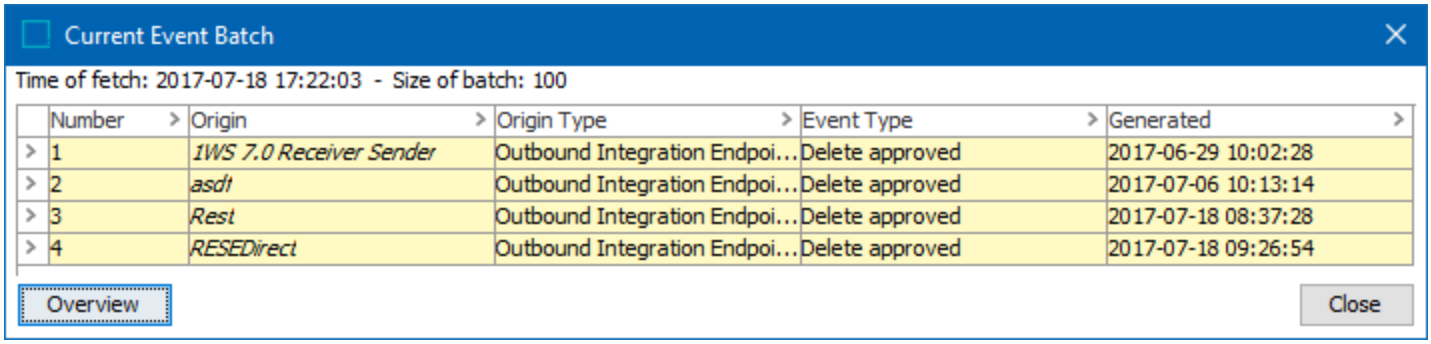
1. Right-click the event processor and select **View First Event Batch**.



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.

Event Queues

The Event Queues node on the System Setup tab can be used to create, modify, and schedule export events from STEP to a local file system. An event queue extracts the data messages and a stylesheet is used to render the messages as static HTML that can be viewed in a browser. The configuration of an event queue can allow for data or assets to be exported, and actions to be automatically performed.

Two types of queues are available for outbound information:

- **Event Queue** legacy functionality for data messages

  While manually creating a data event queue is allowed, improved functionality for exporting data messages is available in an Outbound Integration Endpoint (OIEP). The event queue for an Event-Based OIEP is created automatically when creating the OIEP. For more information, see the **Queue for Endpoint Processes** section of the **OIEP - Configuration Flipper** topic in the **Data Exchange** documentation.

  For information on a manually created event queue, see the **Creating a New Event Queue** topic and the **Maintaining Event Queues** topic.

- **AssetPush Event Queue** and sidecar for assets

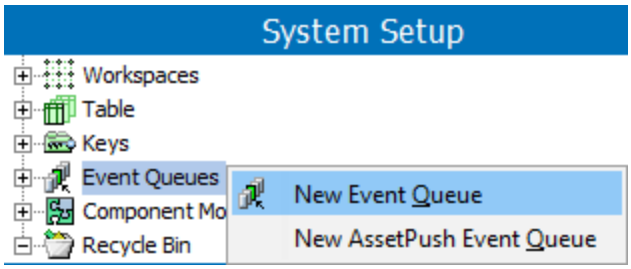
  An event queue for Asset Push must be created manually as defined in the **Creating and Maintaining Asset Push Event Queues** topic of the **Digital Assets** documentation.

Creating a New Event Queue

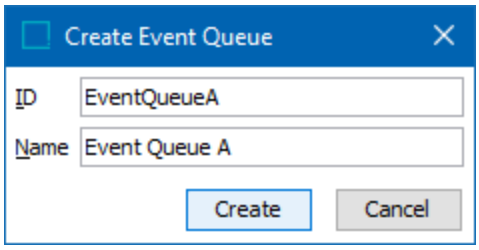
A manually created event queue includes templates for a variety of object types.

Important: The functionality of this legacy feature is limited. Using an OIEP is preferred, as defined in the **Creating an Outbound Integration Endpoint** topic in the **Data Exchange** documentation.

1. In System Setup, right-click the Event Queues node and select the New Event Queue option.

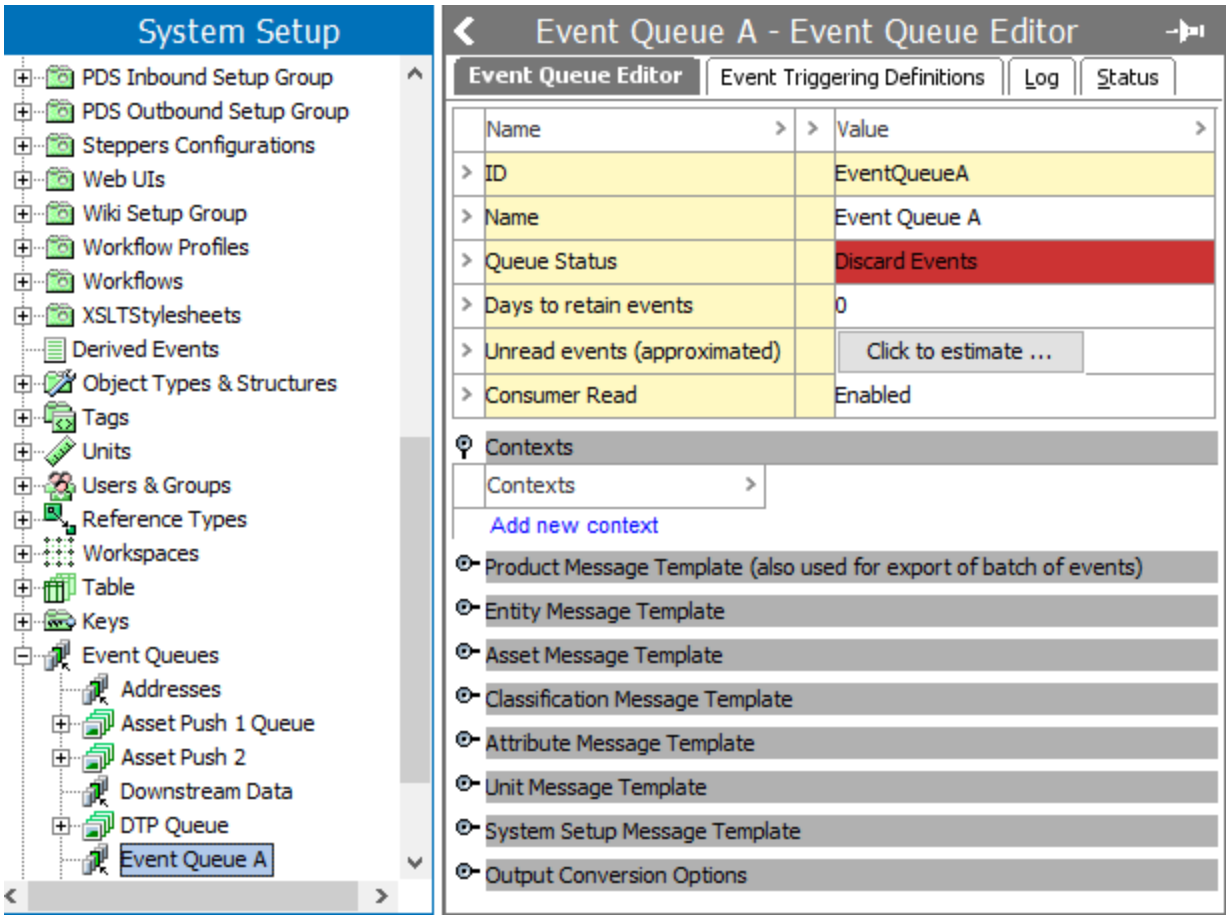


2. Add an ID and a name and click the **Create** button to save the event queue.



   The queue is displayed below the Event Queues node.

3. Select the new queue and on the Event Queue Editor tab, edit the parameters as defined below.

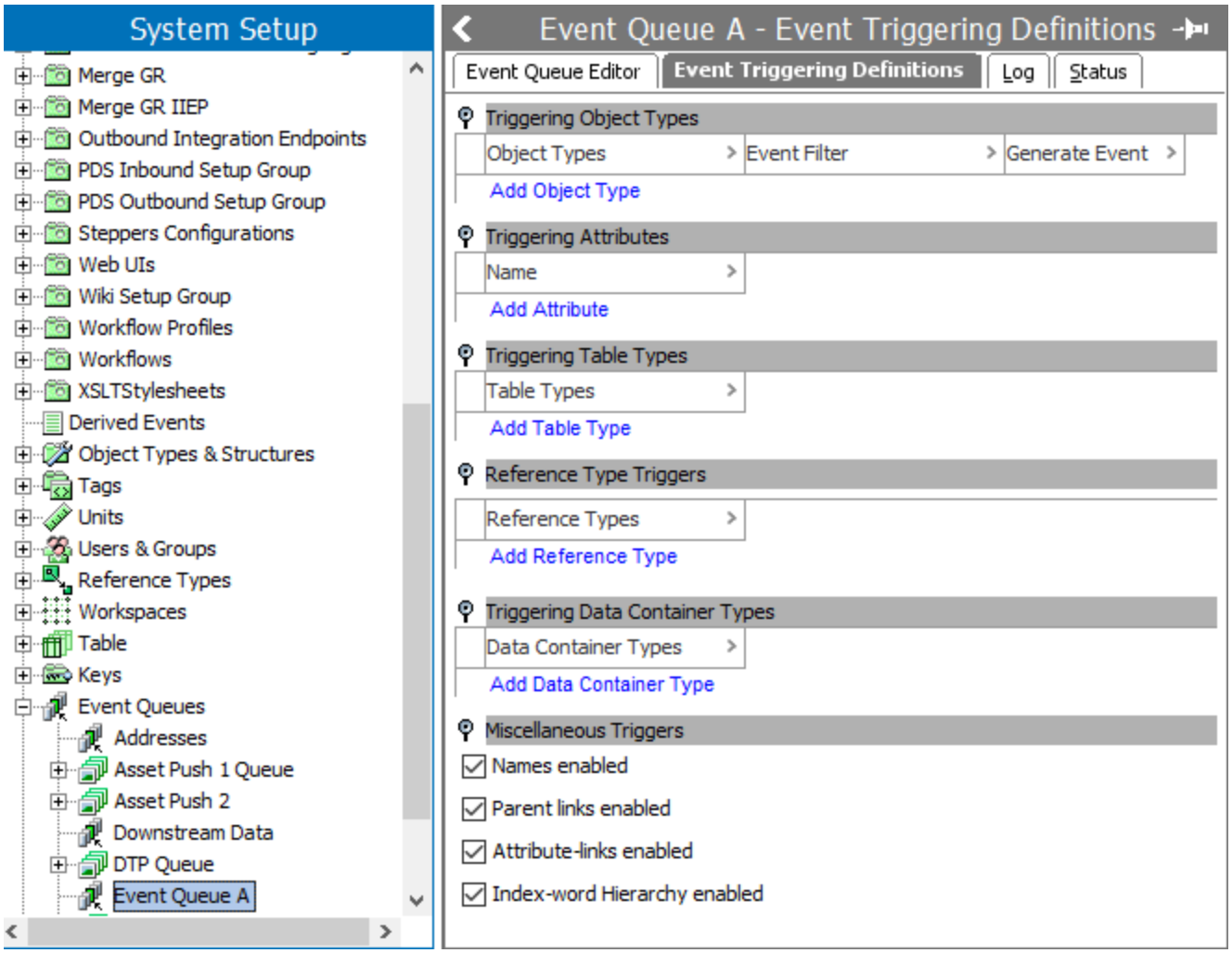


- For details on the **Queue Status** parameter, see the **Event-Based OIEP Status and Queue Status** topic in the **Data Exchange** documentation.
- For details on the **Days to retain events** parameter, see the **Event-Based OIEP Forward, Rewind, Purge, and Republish** topic in the **Data Exchange** documentation.
- For details on the **Unread event (approximated)** parameters, see the **Event-Based OIEP Queued Events** topic in the **Data Exchange** documentation.
- For the **Consumer Read** parameter, **Disabled** (default) means processed events are held and are not delivered to their final destination. This setting stops events from queuing, for example, when the location that receives the deliveries is experiencing problems such as a full disk. **Enabled** means processed events can be delivered to their final destination.

Note: The Event Consumer can be created and edited via the right-click menu when selecting the event queue as defined in the **Create or Edit Event Consumer** topic.

- Make additional changes as necessary for the **Contexts** flipper, the various **Template** flippers, and the **Output Conversion Options** flipper.

4. On the Event Triggering Definitions tab, edit the parameters as defined in the **OIEP - Event-Based - Event Triggering Definitions Tab** topic of the **Data Exchange** documentation.



5. To set the desired queue action, complete queue configuration as defined in the **Maintaining Event Queues** topic.

Maintaining Event Queues

Maintaining an event queue includes defining the actions it should perform and then running the event queue on a schedule.

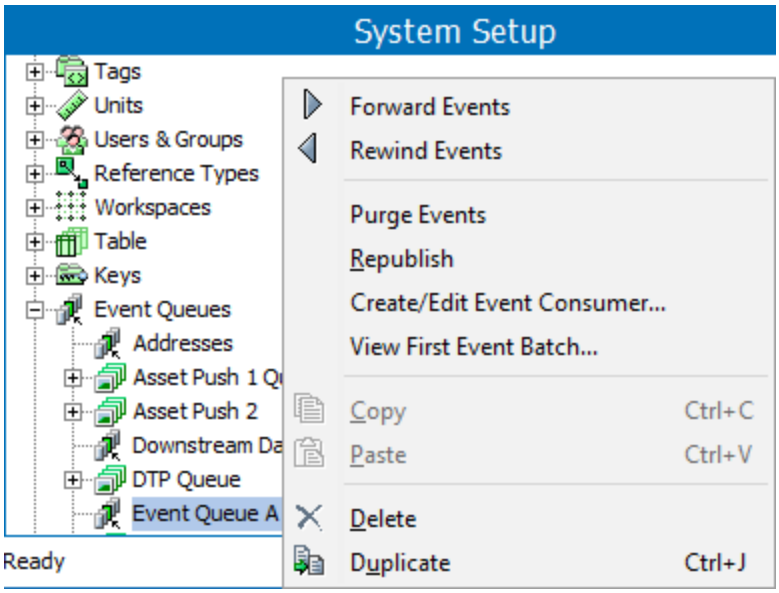
Queue Status and Consumer Read Settings

The action of the event queue is determined by a combination of the 'Queue Status' and 'Consumer Read' parameters, as defined in the table below.

Queue Status	Consumer Read	Description
Read Events	Enabled	Active - Data is delivered to downstream systems.
Read Events	Disabled	Paused - The feed is temporarily disabled. Generated events continue to be read and queued, and are retained within STEP. Messages are not delivered to the downstream system until the Consumer Read setting is changed to ‘Enabled’.
Discard Events	Disabled	Inactive - No new events are processed and no data is delivered downstream.
Discard Events	Enabled	Transition - Allows the queue to process queued events, but does not generate any new events since new events are being handled by a different queue. Not commonly used but can be employed when one queue will take over from another, or prior to running a bulk update process when data should not be delivered downstream.

Running an Event Queue

The right-click menu of an event queue includes the following options.



For information on the Forward Events, Rewind Events, Purge Events, and Republish options, see the topic **Event-Based OIEP Forward, Rewind, Purge, and Republish** in the **Data Exchange** documentation.

For information on the Create/Edit Event Consumer option, see the **Create or Edit Event Consumer** topic.

For information on the View First Event Batch option, which displays events from approved objects that were triggered, see the topic **Event-Based OIEP Queued Events** in the **Data Exchange** documentation.

Create or Edit Event Consumer

Once an event queue exists, an event consumer is required to define where the data will be delivered via a background process.

Important: The functionality of this legacy feature is limited. Using an OIEP is preferred, as defined in the **Creating an Outbound Integration Endpoint** topic in the **Data Exchange** documentation.

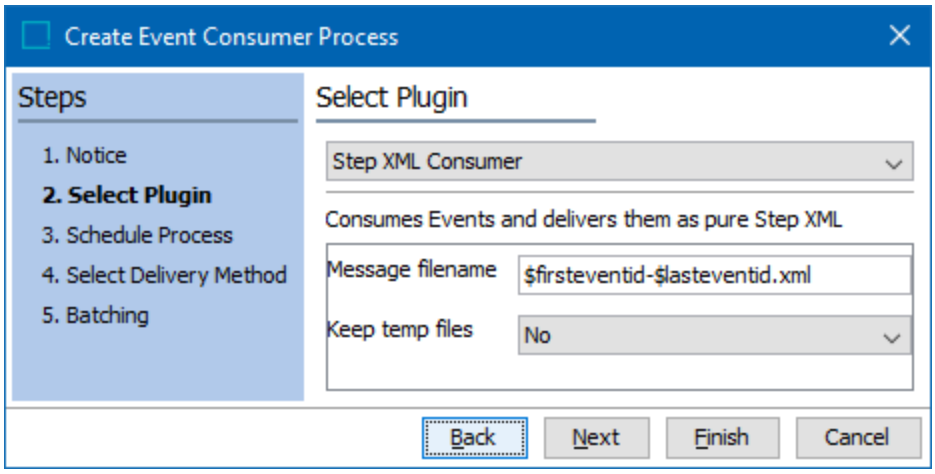
Prerequisites

Complete the steps defined in the following topics before creating an event consumer:

1. Creating a New Event Queue
2. Maintaining Event Queues

Creating or Editing an Event Consumer

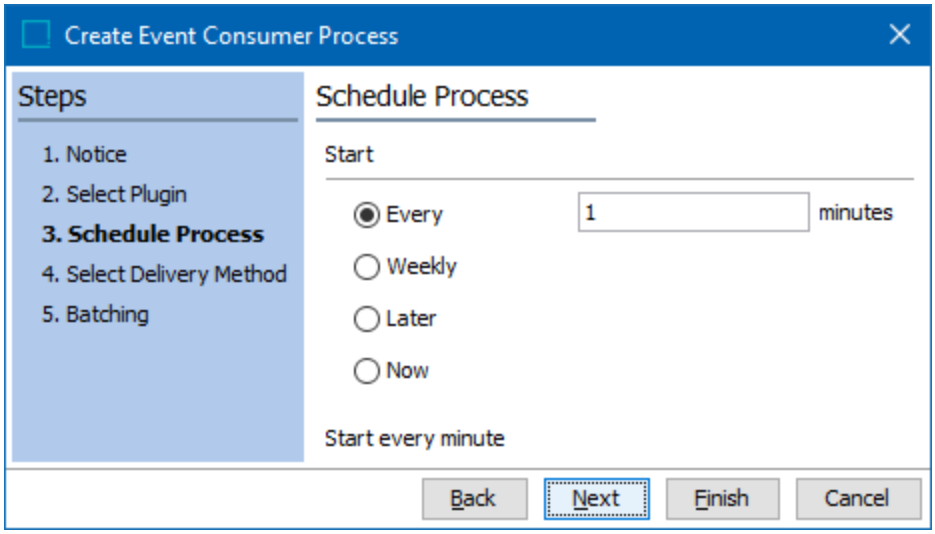
1. In System Setup, right-click an event queue and select the 'Create/Edit Event Consumer...' option to display the Create Event Consumer Process wizard.
2. On the **Notice** step, the message indicates if you are creating a new event consumer or editing an existing one, click the **Next** button.
3. On the **Select Plugin** step, choose a plugin from the dropdown and complete the additionally required parameters, then click the **Next** button.



Note: Hover over a parameter label to display information on the required data.

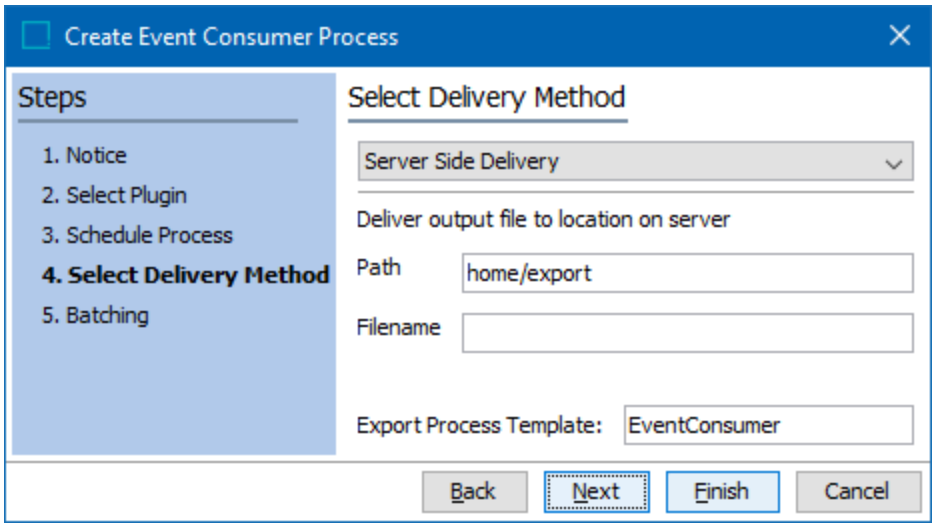
- **Step XML Consumer** (default) delivers events as STEPXML. For more information on this format, see the **STEPXML Format** topic in the **Data Exchange** documentation.
- **Demo Consumer (Firefox)** uses a consumed event queue with a specific Firefox demo plugin to deliver XML that can be rendered by the Firefox browser.

4. On the **Schedule Process** step, set the schedule as defined in **Scheduled Data Export - Schedule Process** topic in the **Data Exchange** documentation, then click the **Next** button.

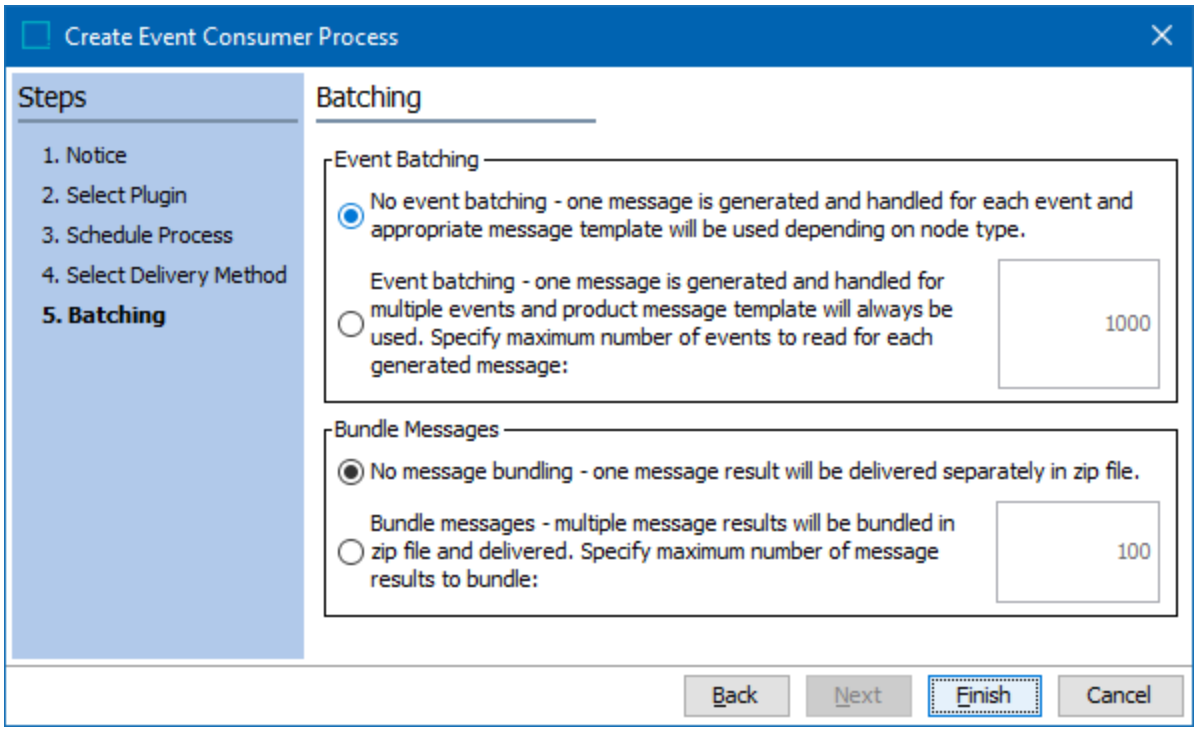


Important: Consider the time zone of the application server compared to that of the workbench (the client) where the schedule is created or viewed. When scheduling a job, the local time zone is displayed in the workbench, but the time zone of the server is used to run the background process. Although displayed, the time zone of the client is not included in the instruction to the server to run the job. This can cause confusion about when the job will run since the scheduled time is not automatically converted to accommodate potential differences in time zones.

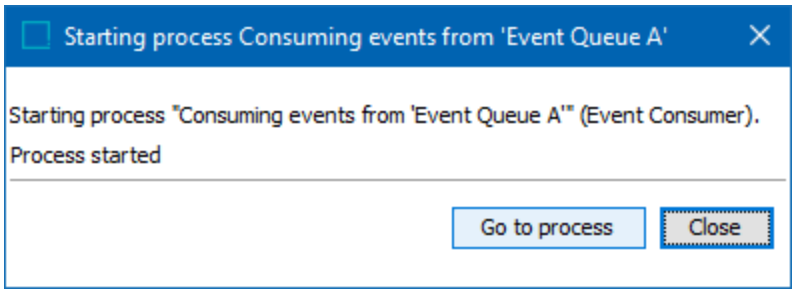
5. On the **Select Delivery Method** step, choose a method from the dropdown and complete the additionally required parameters as defined in the **Export Manager - Select Delivery Method** topic in the **Data Exchange** documentation, then click the **Next** button.



6. On the **Batching** step, select an option as defined in **OIEP - Event-Based - Choose Data Source** topic in the **Data Exchange** documentation, then click the **Finish** button to start the background process.



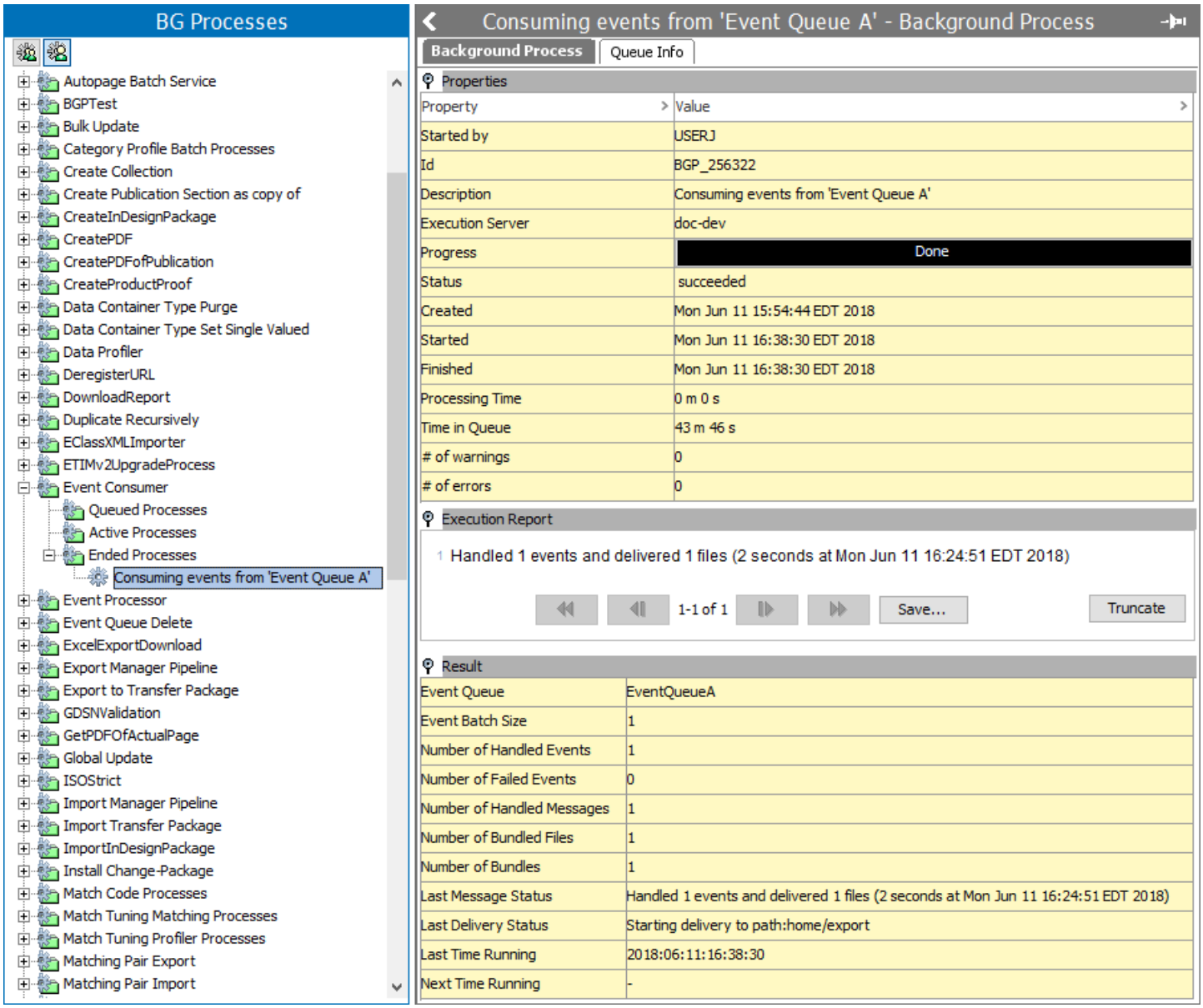
7. On the 'Starting process Consuming events from '[eventqueue]' dialog, make a selection.



- **Go to process** displays the background process on the BG Processes tab under the 'Event Consumer' node.
- **Close** starts the background and closes the 'Starting process...' dialog.

Output

The background process editor as shown below, displays details about the progress, success, or failure of the BGP. The Result flipper includes the location of the output based on the Delivery Method step selection.


BG Processes
Autopage Batch Service
BGPTest
Bulk Update
Category Profile Batch Processes
Create Collection
Create Publication Section as copy of
CreateInDesignPackage
CreatePDF
CreatePDFofPublication
CreateProductProof
Data Container Type Purge
Data Container Type Set Single Valued
Data Profiler
DeregisterURL
DownloadReport
Duplicate Recursively
EClassXMLImporter
ETIMv2UpgradeProcess
Event Consumer
Queued Processes
Active Processes
Ended Processes
Consuming events from 'Event Queue A'
Event Processor
Event Queue Delete
ExcelExportDownload
Export Manager Pipeline
Export to Transfer Package
GDSNValidation
GetPDFOfActualPage
Global Update
ISOStrict
Import Manager Pipeline
Import Transfer Package
ImportInDesignPackage
Install Change-Package
Match Code Processes
Match Tuning Matching Processes
Match Tuning Profiler Processes
Matching Pair Export
Matching Pair Import
Consuming events from 'Event Queue A' - Background Process
Background Process
Queue Info
Properties
Property
Value
Started by
USERJ
Id
BGP_256322
Description
Consuming events from 'Event Queue A'
Execution Server
doc-dev
Progress
Done
Status
succeeded
Created
Mon Jun 11 15:54:44 EDT 2018
Started
Mon Jun 11 16:38:30 EDT 2018
Finished
Mon Jun 11 16:38:30 EDT 2018
Processing Time
0 m 0 s
Time in Queue
43 m 46 s
# of warnings
0
# of errors
0
Execution Report
1 Handled 1 events and delivered 1 files (2 seconds at Mon Jun 11 16:24:51 EDT 2018)
1-1 of 1
Save...
Truncate
Result
Event Queue
EventQueueA
Event Batch Size
1
Number of Handled Events
1
Number of Failed Events
0
Number of Handled Messages
1
Number of Bundled Files
1
Number of Bundles
1
Last Message Status
Handled 1 events and delivered 1 files (2 seconds at Mon Jun 11 16:24:51 EDT 2018)
Last Delivery Status
Starting delivery to path:home/export
Last Time Running
2018:06:11:16:38:30
Next Time Running
-

Metrics

Metrics evaluate data and return an integer between 1 and 100. The returned integer, known as a 'metric score', can be used in the profiling of objects and is vital for generating data policies.

Several different metric types are available:

- Completeness Metrics
- Value Metrics
- Business Function Metrics
- Aggregator Metrics
- PDX Channel Metric
- Attribute Comparison Metric
- Attribute Value Analysis Metric
- Conditional Attribute Value Metric
- Number of Referenced Target Object Metrics

In addition to information about the different metric types, this section also covers initial setup of metrics in STEP, creating and editing metrics, and information about viewing the metrics in Web UI and workbench.

Aggregator Metrics

An aggregator metric allows users to combine scores from multiple metrics. By default, the aggregator metric returns an average score based on these combined scores; however, by adding a business function to an aggregator metric, users are able to obtain scores based on other variables, e.g., if one metric is weighted heavier than another metric within the aggregator metric, the score would reflect the difference based on the weight of each metric.

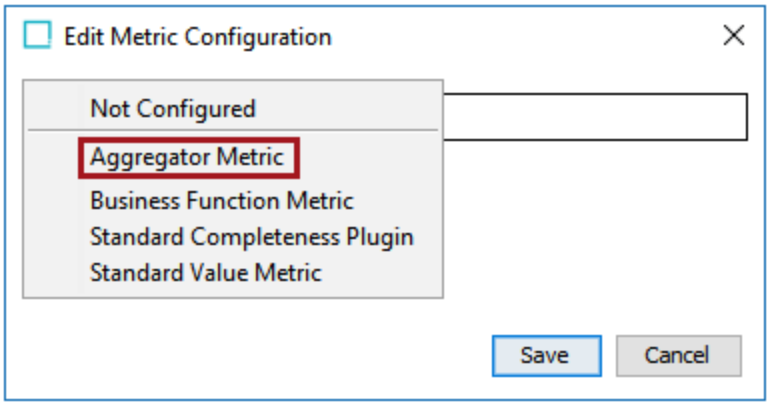
Aggregator Metric Configuration

Before metrics can be created, a Setup Group must be configured to hold them. For information on configuring a setup group for metrics, see the **Initial Setup for Metrics** topic in the **Metrics** documentation.

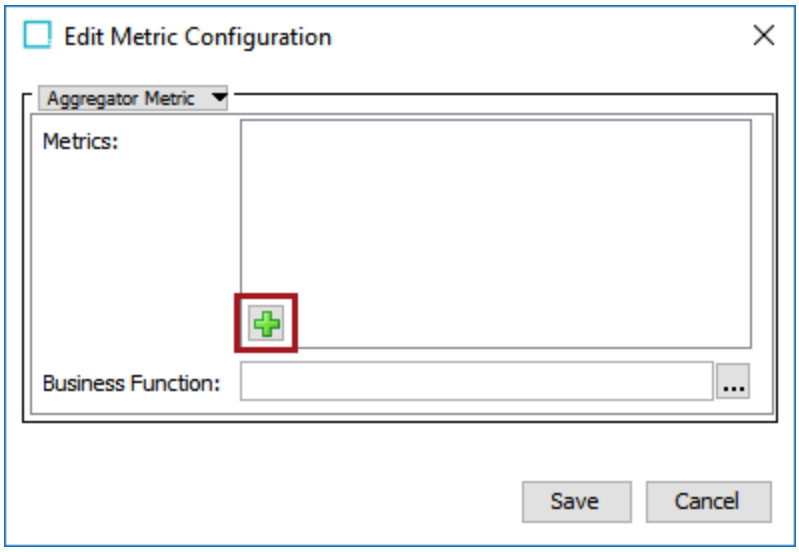
For information on creating a metric, see the **Creating and Editing Metrics** topic in the **Metrics** documentation.

To configure an aggregator metric:

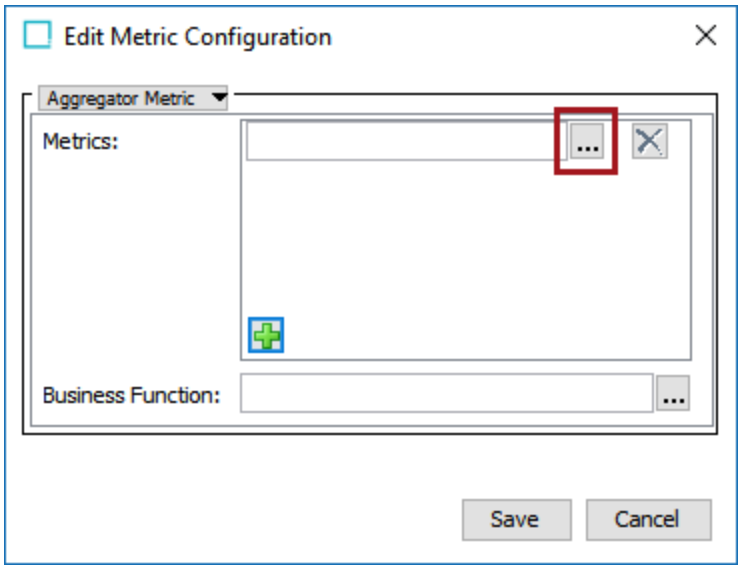
1. From the Edit Metric Configuration wizard, select 'Aggregator Metric.'



2. Click the green plus sign.

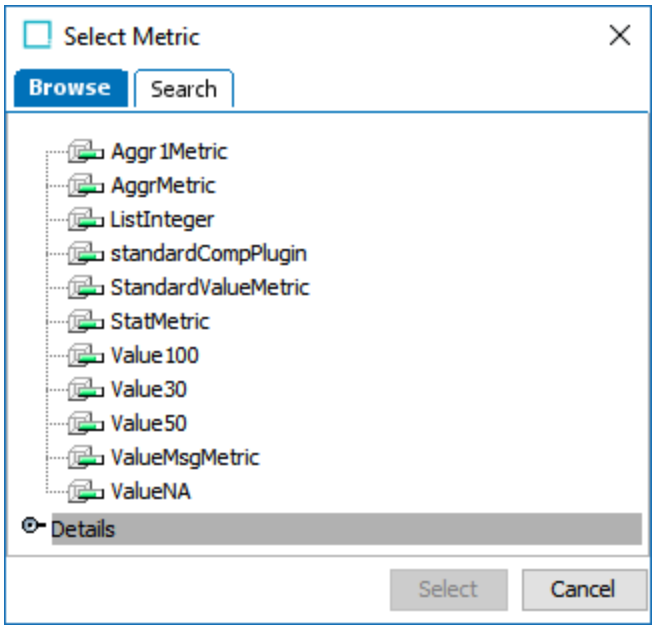


3. Click the ellipsis button.

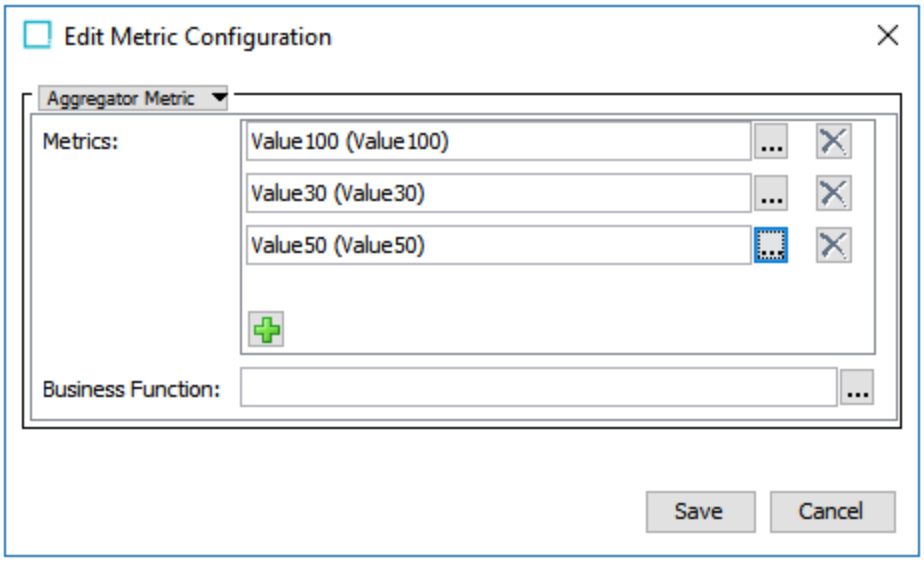


4. Select the metric you want to include in the aggregator metric and click 'Select.'

Note: Only one metric can be selected at a time. After a metric is selected, the green plus button must be clicked on again to add another metric.



In the image below, three metrics have been selected from the available metric list.

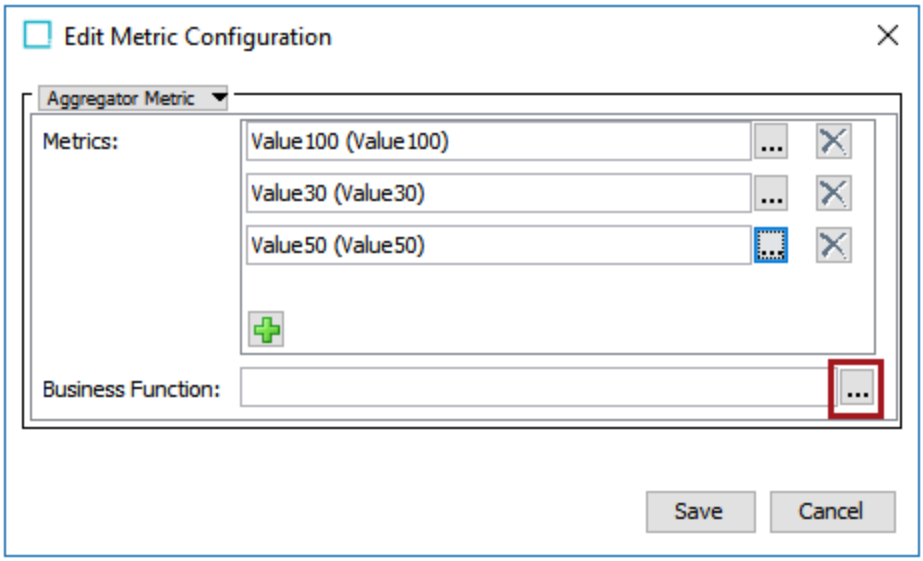


Adding a business function to an aggregator metric

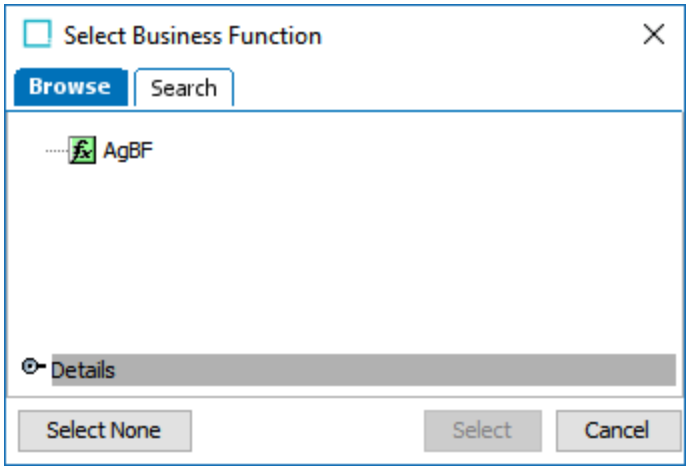
Although optional, the purpose of adding a business function to an aggregator metric is to return a score that is not the average score of the configured metrics. If no business function is selected, the returned score will be the average score of the configured metrics included in the aggregator metric.

To add a business function to an aggregator metric:

1. Click on the business function ellipsis.



2. Select a business function and click the 'Select' button.

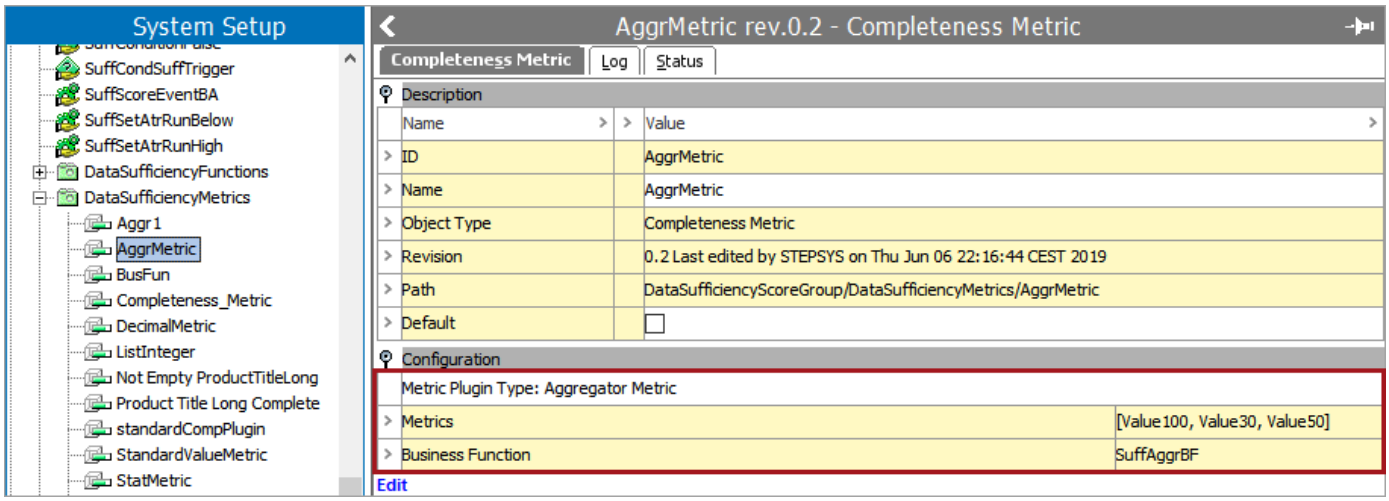


3. Click 'Save' to close the 'Edit Metric Configuration' wizard.

The aggregator metric is configured and ready for use.

Note: To use a business function with an aggregator metric, the input parameter type must be set to 'List<MetricResult>' and the return type must be an integer. Business functions without these configuration settings will not be available as a valid option within the Select Business Function window. For more information on business functions, see the **Business Functions** topic in the **Functions** documentation.

With the configured metric selected, the name of the metric plugin type (in this case, 'Aggregator Metric'), the metrics included in the aggregator metric, and the name of the selected business function will be listed under the Configuration flipper.


System Setup
SuffCondSuffTrigger
SuffScoreEventBA
SuffSetAtrRunBelow
SuffSetAtrRunHigh
DataSufficiencyFunctions
DataSufficiencyMetrics
Aggr1
AggrMetric
BusFun
Completeness_Metric
DecimalMetric
ListInteger
Not Empty ProductTitleLong
Product Title Long Complete
standardCompPlugin
StandardValueMetric
StatMetric
AggrMetric rev.0.2 - Completeness Metric
Completeness Metric
Log
Status
Description
Name
Value
ID
AggrMetric
Name
AggrMetric
Object Type
Completeness Metric
Revision
0.2 Last edited by STEPSYS on Thu Jun 06 22:16:44 CEST 2019
Path
DataSufficiencyScoreGroup/DataSufficiencyMetrics/AggrMetric
Default
Configuration
Metric Plugin Type: Aggregator Metric
Metrics
[Value100, Value30, Value50]
Business Function
SuffAggrBF
Edit

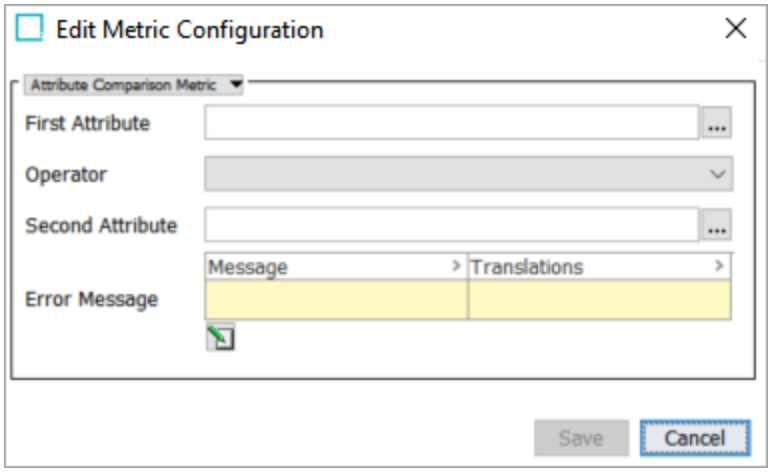
Attribute Comparison Metric

The Attribute Comparison Metric compares the values of two selected attributes based on an operator selected by the user, and returns a score of '100' (true) or '0' (false). Users are able to create an error message for instances when the metric returns a score of '0'; this message is meant to detail the reason for the '0' score and can have multiple translations.

Important: Metrics are used in conjunction with sufficiencies to determine the quality and completeness of a product's data. While this topic covers the functionality and configuration of the Attribute Comparison Metric, it is important that users are also familiar with sufficiencies and how they are combined with metrics. Information regarding sufficiencies can be found in the **Sufficiency Configuration Type** topic.

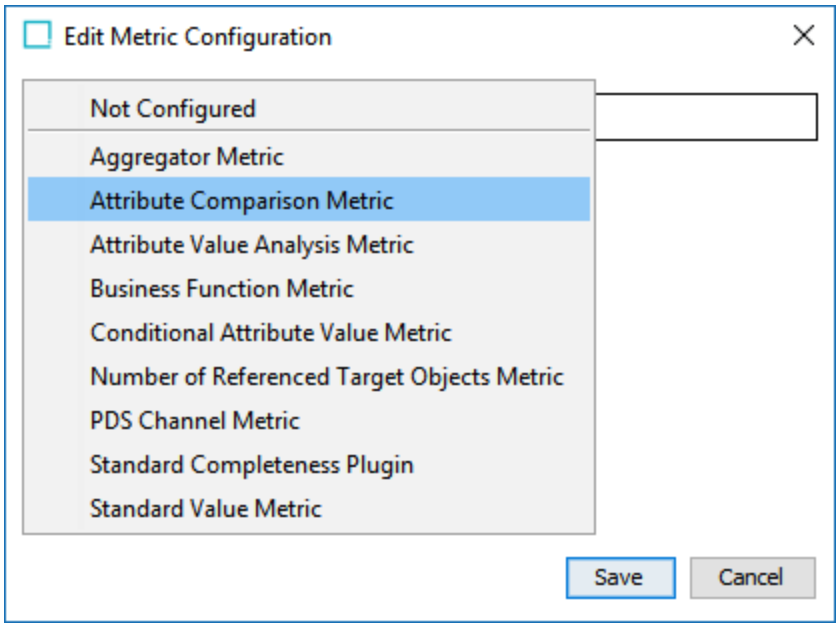
Prerequisites

Before metrics can be created, a Setup Group must be configured. For information on configuring a setup group for metrics, see the **Initial Setup for Metrics** topic in the **Metrics** documentation. For information on creating a metric, see the **Creating and Editing Metrics** topic in the **Metrics** documentation.



Attribute Comparison Metric configuration

1. From the Edit Metric Configuration wizard, select 'Attribute Comparison Metric.'



2. With the Attribute Comparison Metric selected, click the ellipsis buttons (...) to select the attributes that you want to compare and select an operator from the 'Operator' dropdown list.

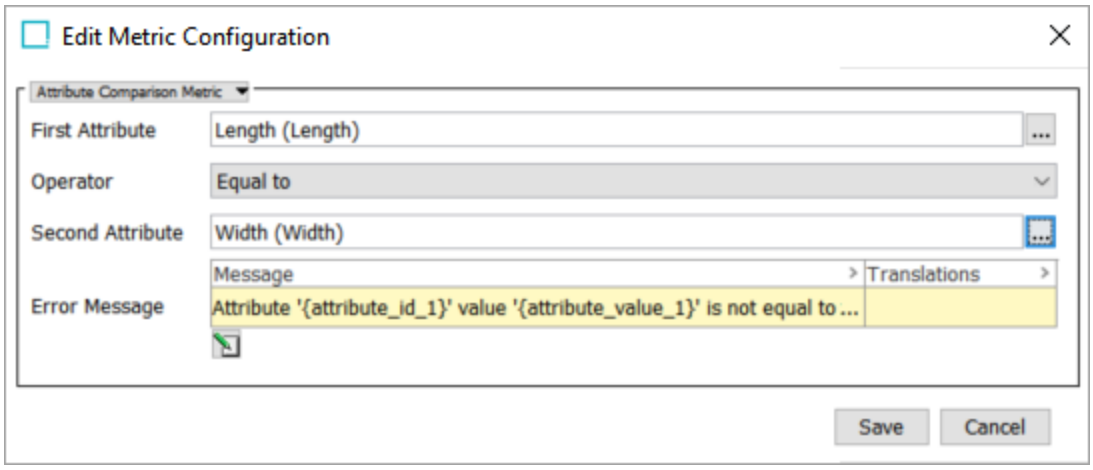
   Once an operator is selected from the 'Operator' dropdown list, a default error message will appear in the 'Error Message' text field, based on the selected operator.

   Operator options include:

   - Equal to
   - Not Equal to
   - Contained in
   - Greater than
   - Less than

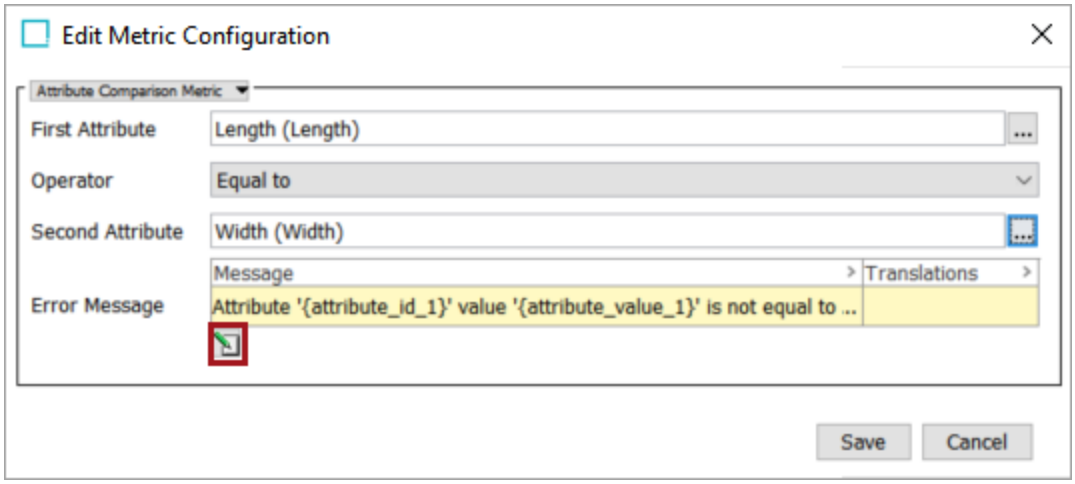
   Note: If the first attribute and/or second attribute has a value that is not a number, the 'Greater than' and 'Less than' operators will not be available.

   In the example below, the user has selected 'Length' for the first attribute, 'Width' for the second attribute, and 'Equal to' for the operator. If the value for the first attribute is equal to the value for the second attribute, then the returned score will be '100' (true). If the value for the first attribute is not equal to the value of the second attribute, then the returned score will be '0' (false), and will be accompanied with a message, as displayed in the 'Error Message' text field.



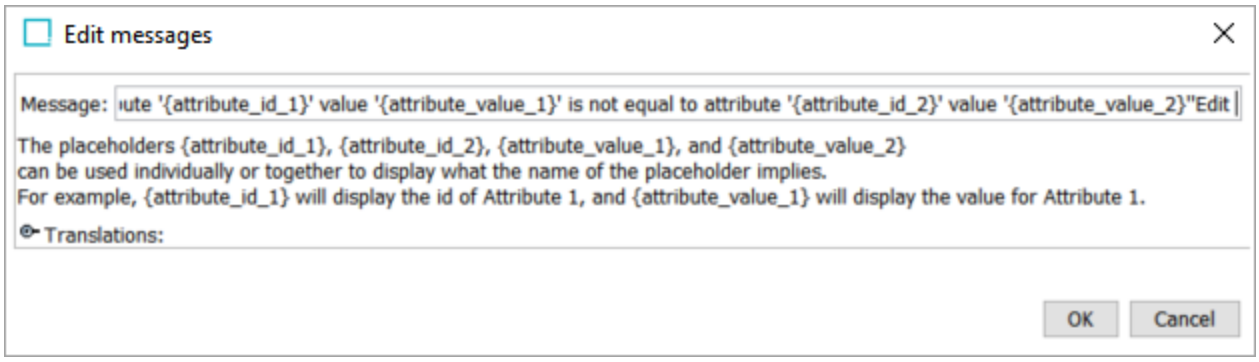
Note: If the user selects an attribute that is not valid for a product, the metric will be considered non-applicable for that product and no data quality information will be produced when the Sufficiency Data Calculator event processor is run.

3. To edit the error message and/or add additional translations, click on the 'Error Message' editor button.

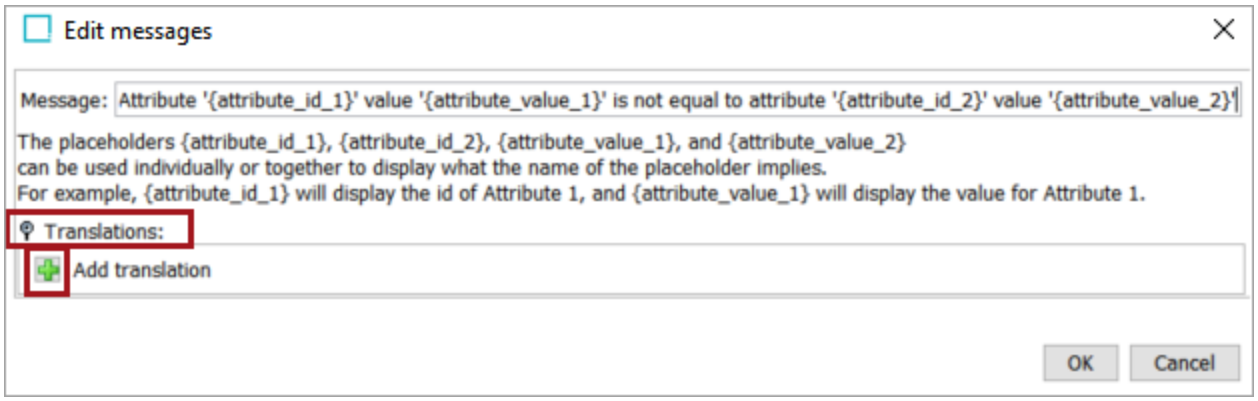


The 'Edit messages' window appears. Note that the 'Message' text field is editable. Descriptive text below the text field describes in detail how the placeholders included in the error message work in order to create an error message. Although the default error message (or something similar) is suggested, users are able to edit the error message, depending on their individual needs.

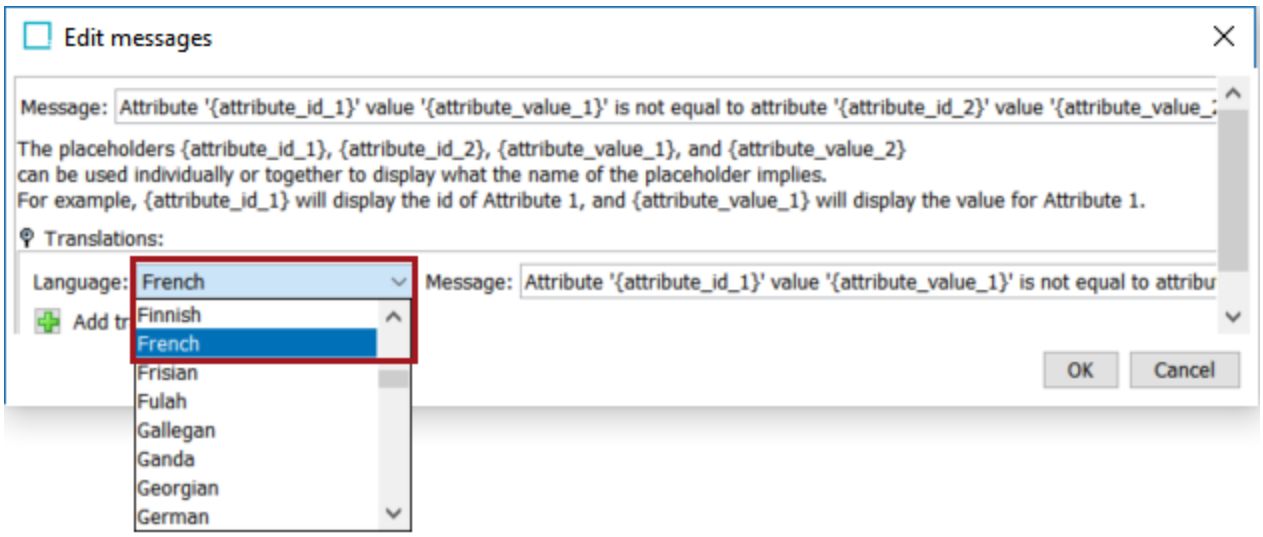
Note: Although the error message can be edited by the user, the placeholders themselves (attribute_id and attribute_value) cannot be edited. If the placeholders are edited, the user will receive a validation error message when attempting to save the edited message. When the metric is run, placeholders are replaced by the 'real values,' e.g., 'attribute_id' would be replaced with the name of the attribute that the metric is evaluating.



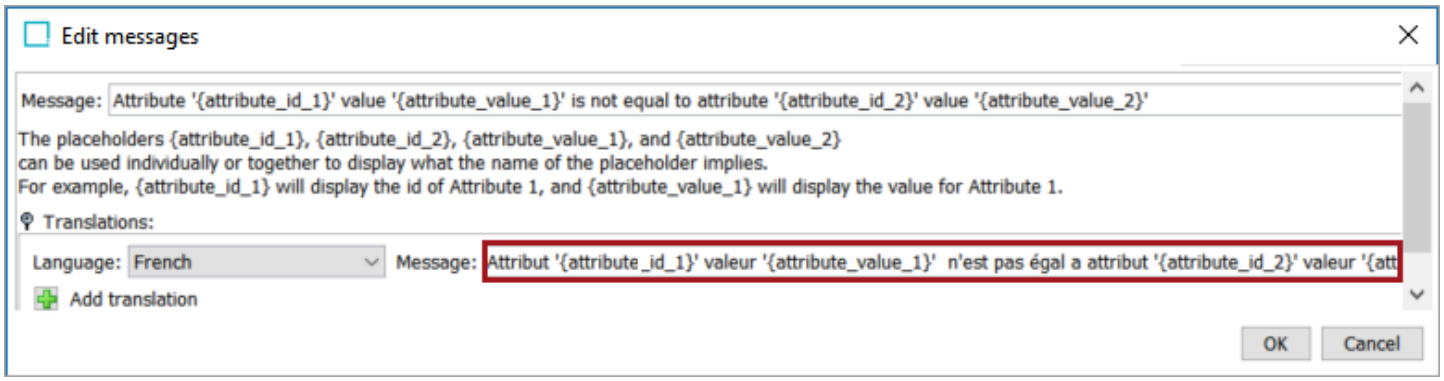
4. Select the 'Translations' flipper and click the green plus sign located next to 'Add translation.'



5. Click on the arrow located in the 'Language' dropdown menu; in the example below, the user has opened the dropdown menu and selected 'French.'



6. Once a language is selected, edit the message to reflect the language specified for translation.

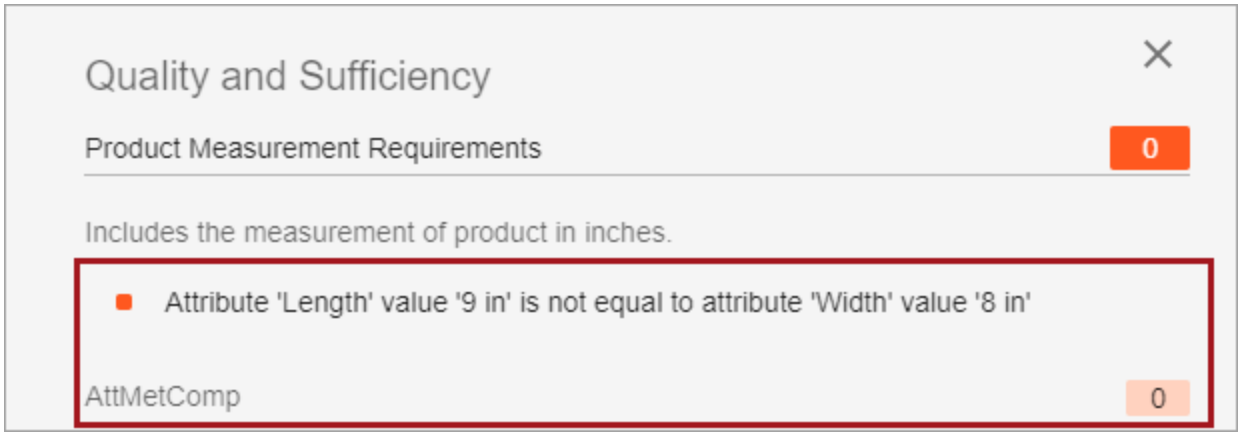


7. Click the green plus sign to add another translation or 'OK' to close the 'Edit messages' window.
8. Click 'Save' to close the Edit Metric Configuration wizard.

The metric is saved.

Note: Translation error messages are displayed by using locales; they are not context-dependant. For more information about locales, see the **Localization** topic in the **Administrative Portal** documentation. For more information about localized messages, see the **Localized Messages for JavaScript Business Rules** topic in the **Business Actions** documentation.

Once the metric is added to a sufficiency and the sufficiency is run, the results can be seen in the sufficiency panel within the Web UI. In the example below, the Attribute Comparison Metric (for this example, titled 'AttMetComp') has been added to the sufficiency 'Product Measurement Requirements,' and returned a score of '0' because the attribute 'Length' value (9 in) is not equal to the attribute "Width' value (8 in), as detailed in the returned message.

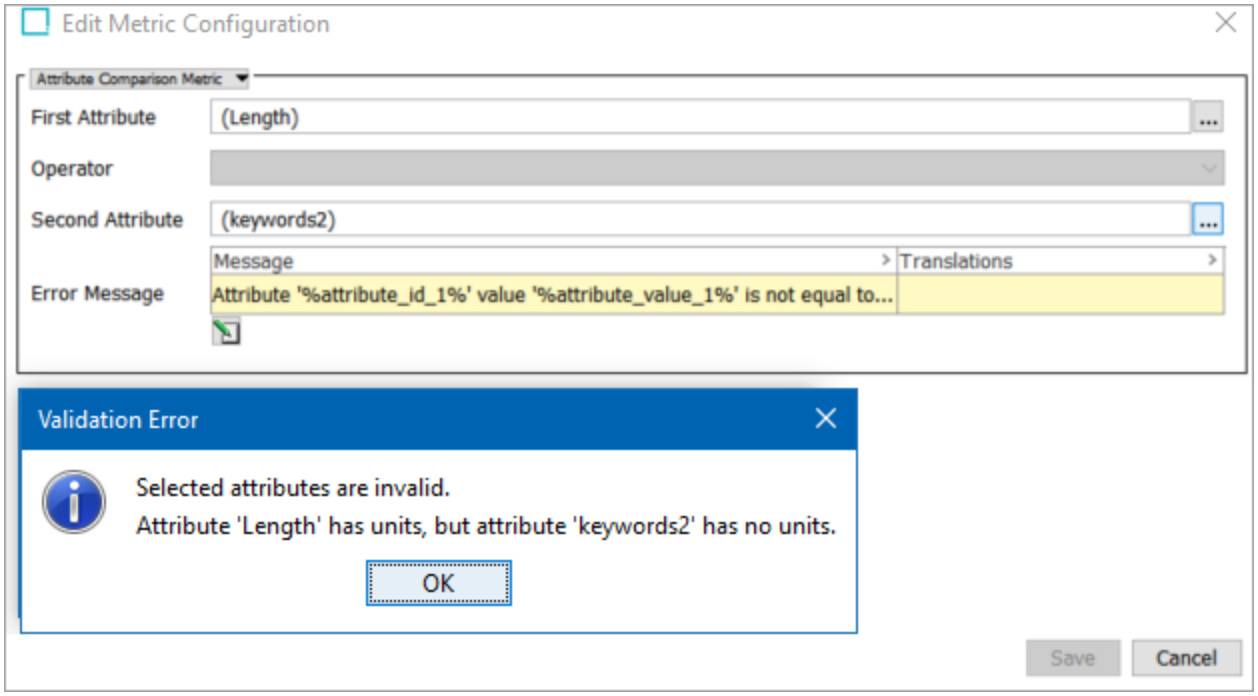


Validation Errors

In certain instances, validation errors can occur, depending on the attributes selected for comparison. A validation error (accompanied by an error message) will occur if:

- The selected attributes have the same ID.
- One selected attribute is multivalued and the other selected attribute is not multivalued.
- One selected attribute includes a unit and the other selected does not include a unit of measure.

In the example below, the first attribute 'Length' has units, whereas the second attribute 'keywords2' does not have units. Because of the incompatibility, a validation error occurs, with an error message that describes the reason for the error.



The following instances will also produce validation errors. However, these errors will not appear until the metric is run. The error message will appear in the Background Process Log of the Data Sufficiency Calculator event processor.

- Both selected attributes have units, but one of the units does not have a set default unit, and no unit information is provided for the attribute value by the user (in the product).
- Both selected attributes have units, but the units do not have the same base unit.
- Both selected attributes have units, but one or both of the selected attributes do not have a base unit conversion rule.

For more information regarding units, including base units and conversion rules, see the **Units** topic in the **System Setup** documentation.

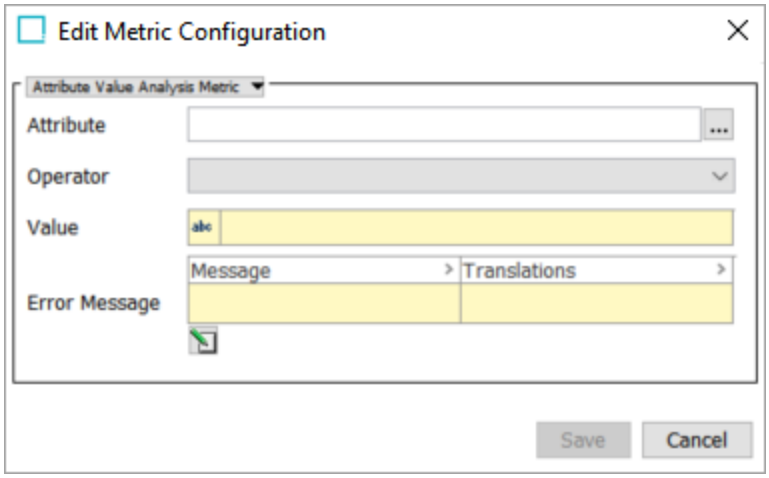
Attribute Value Analysis Metric

The Attribute Value Analysis Metric analyzes the value of a selected attribute based on an operator selected by the user, and returns a score of '100' (true) or '0' (false). Users are able to create an error message for instances when the metric returns a score of '0'; this message is meant to detail the reason for the '0' score and can have multiple translations.

Important: Metrics are used in conjunction with sufficiencies to determine the quality and completeness of a product's data. While this topic covers the functionality and configuration of the Attribute Comparison Metric, it is important that users are also familiar with sufficiencies and how they are combined with metrics; information regarding sufficiencies can be found in the **Sufficiency Configuration Type** documentation.

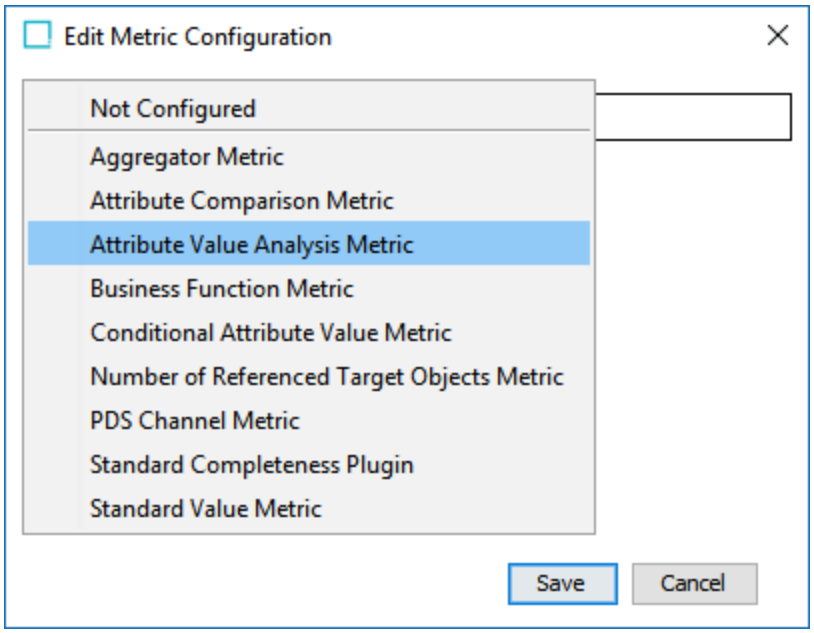
Prerequisites

Before metrics can be created, a Setup Group must be configured to hold them. For information on configuring a setup group for metrics, see the **Initial Setup for Metrics** topic in the **Metrics** documentation. For information on creating a metric, see the **Creating and Editing Metrics** topic in the **Metrics** documentation.



Attribute Value Analysis Metric configuration

1. From the Edit Metric Configuration wizard, select 'Attribute Value Analysis Metric.'



2. With the Attribute Analysis Metric selected, click the ellipsis button (...) to select the attribute that you want to analyze, select an operator from the 'Operator' dropdown list, and input a value into the 'Value' text field.

   Once an operator is selected from the 'Operator' dropdown list, a default error message will appear in the 'Error Message' text field, based on the selected operator.

   Operator options include:

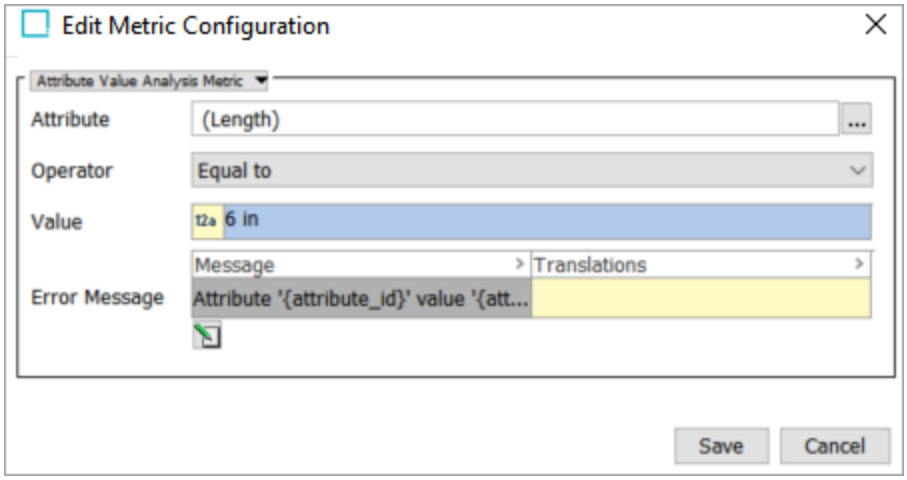
   - Is empty
   - Is not empty
   - Equal to
   - Not Equal to
   - Contains
   - Regular expression
   - Greater than
   - Less than
   - Length is equal to
   - Length is greater than
   - Length is less than

   Note: If the attribute selected has a validation base type that does not allow for numbers, the 'Greater than' and 'Less than' operator options will not be available. If 'Is empty' or 'Is not empty' is selected as an operator, the 'Value' text field will be disabled.

   In the example below, the user has selected 'Length' for the attribute and 'Equal to' for the operator. The user wants to know if the value of the 'Length' attribute is equal to '6 in.' If it is, then the returned score will be '100'

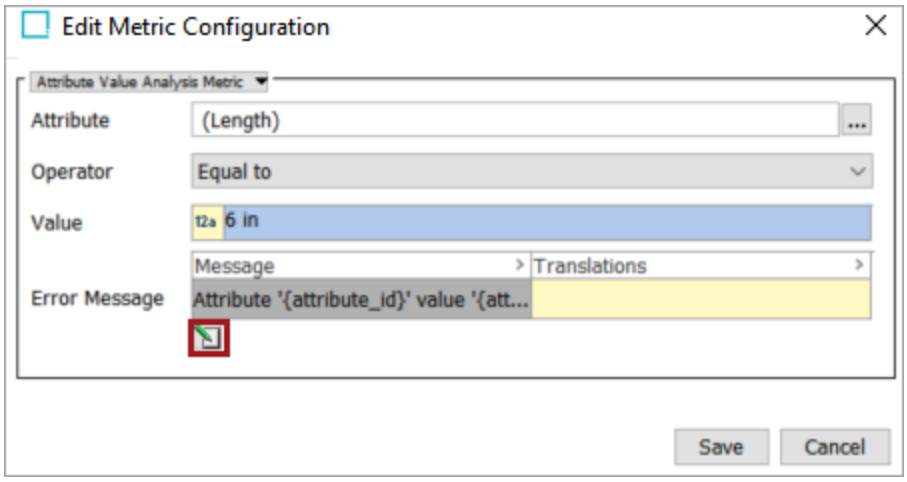
(true). If it is not, then the returned score will be '0' (false), and this score will be accompanied with a message, as displayed in the 'Error Message' text field.

Note: In this example, the Value text field includes a validation base type of 'Numeric Text' (displayed as '12a'). The validation base type displayed in the Value text field is dependent on the validation base type of the attribute selected by the user for analysis. For information regarding validation base types, see the **Base Unit ID, Name, and Value Aspects** topic in the **Outbound Map Data** documentation.



Note: If the user selects an attribute that is not valid for a product, the metric will be considered non-applicable for that product and no data quality information will be produced when the Sufficiency Data Calculator event processor is run.

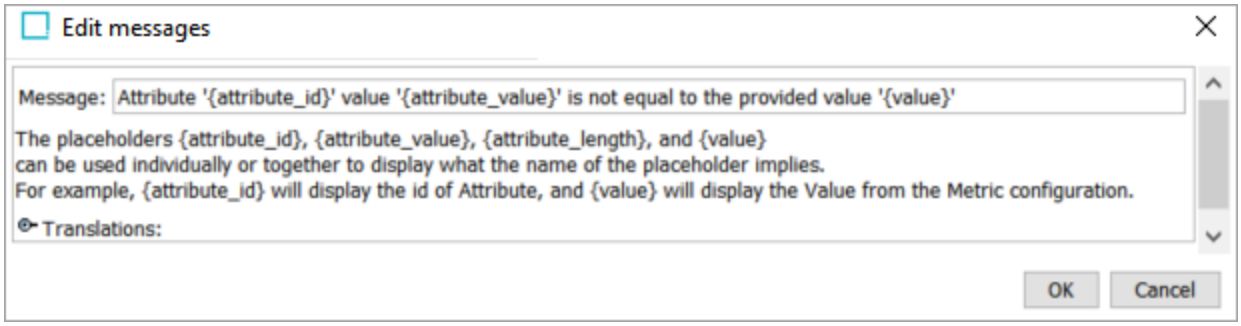
3. To edit the error message and/or add additional translations, click on the 'Error Message' editor button.



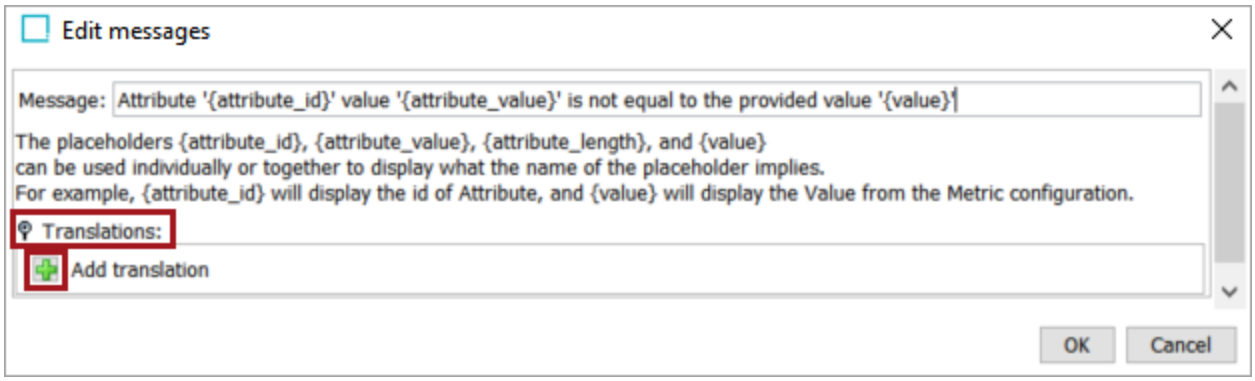
The 'Edit messages' window appears. Note that the 'Message' text field is editable. Descriptive text below the text field describes in detail how the placeholders included in the error message work in order to create an

error message. Although the default error message (or something similar) is suggested, users are able to edit the error message, depending on their individual needs.

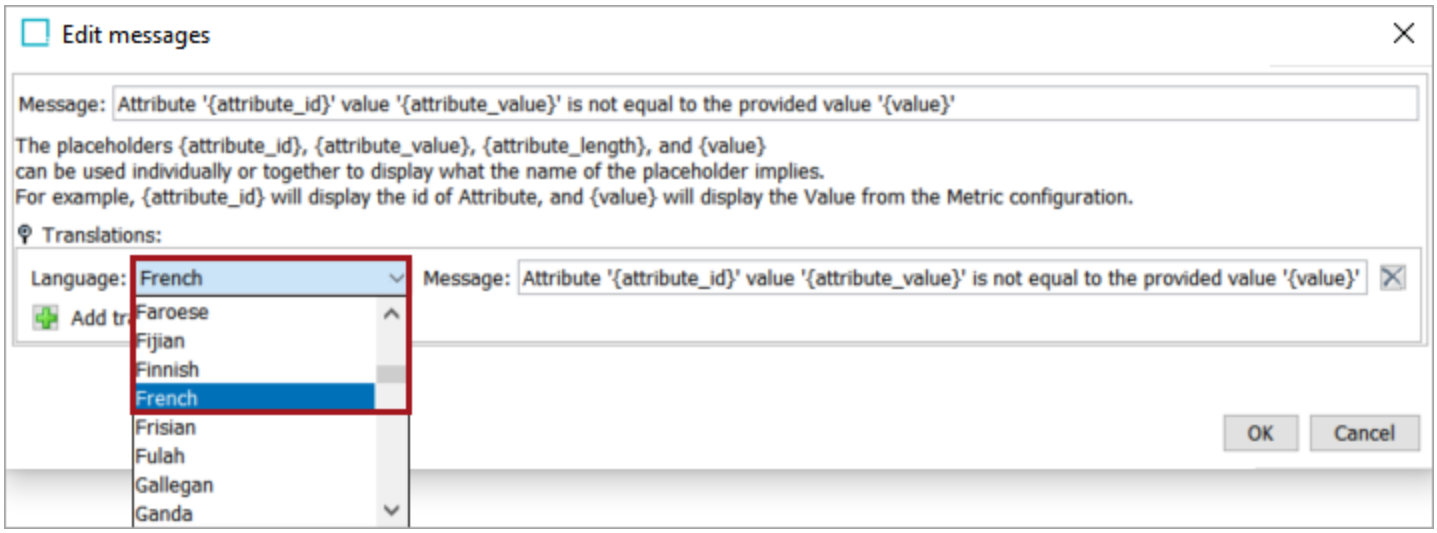
Note: Although the error message can be edited by the user, the placeholders themselves (attribute_id, attribute_value) cannot be edited. If the placeholders are edited, the user will receive a validation error message when attempting to save the edited message. When the metric is run, placeholders are replaced by the ' real values,' e.g., 'attribute_id' would be replaced with the name of the attribute that the metric is evaluating.



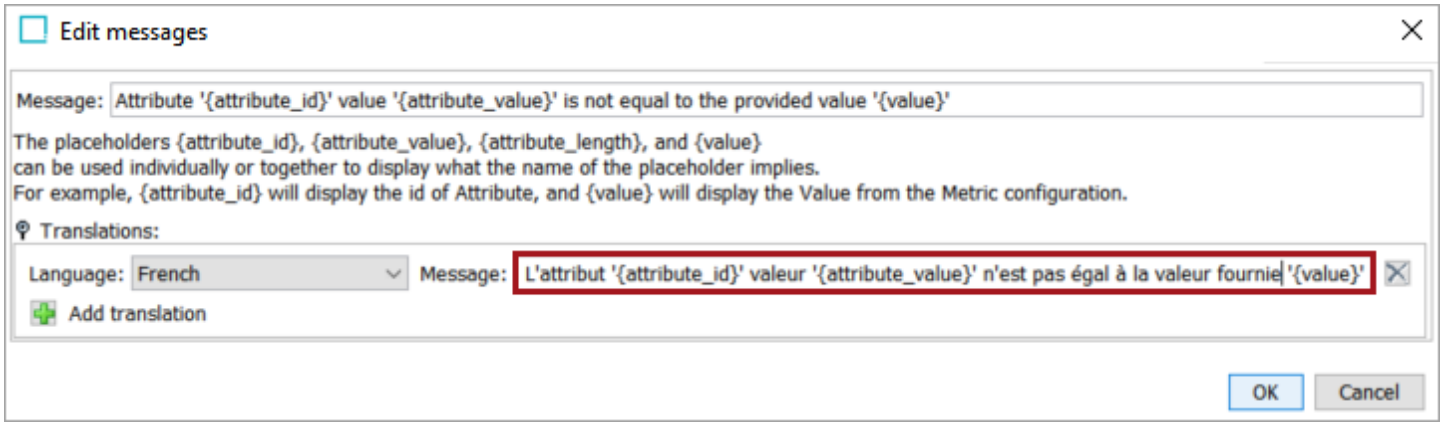
4. Select the 'Translations' flipper and click the green plus sign located next to 'Add translation.'



5. Click on the arrow located in the 'Language' dropdown menu; in the example below, the user has opened the dropdown menu and selected 'French.'



6. Once a language is selected, edit the message to reflect the language specified for translation.

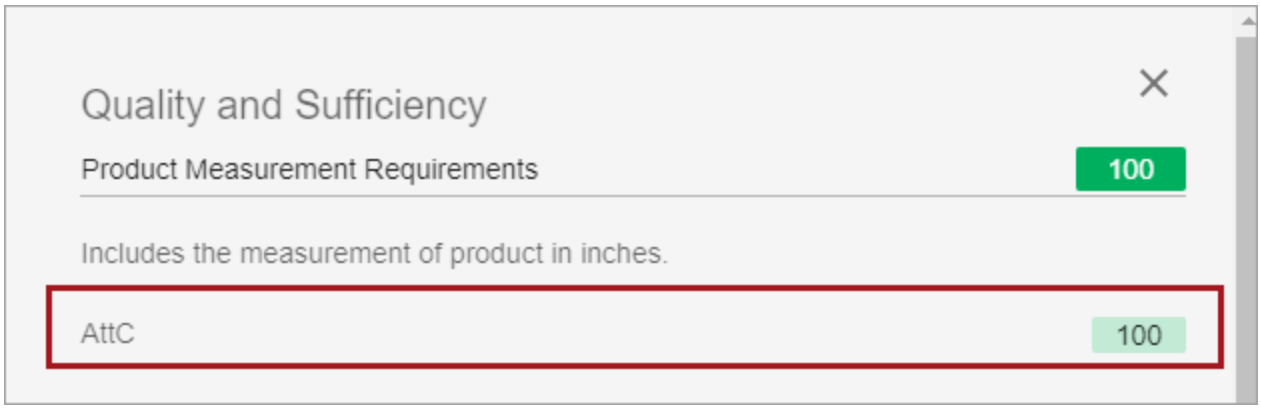


7. Click the green plus sign to add another translation or 'OK' to close the 'Edit messages' window.
8. Click 'Save' to close the Edit Metric Configuration wizard.

The metric is saved.

Note: The localized error messages are displayed by using locales; they are not context-dependent. For more information about locales, see the **Localization** topic in the **Administrative Portal** documentation. For more information about localized messages, see the **Localized Messages for JavaScript Business Rules** topic in the **Business Actions** documentation.

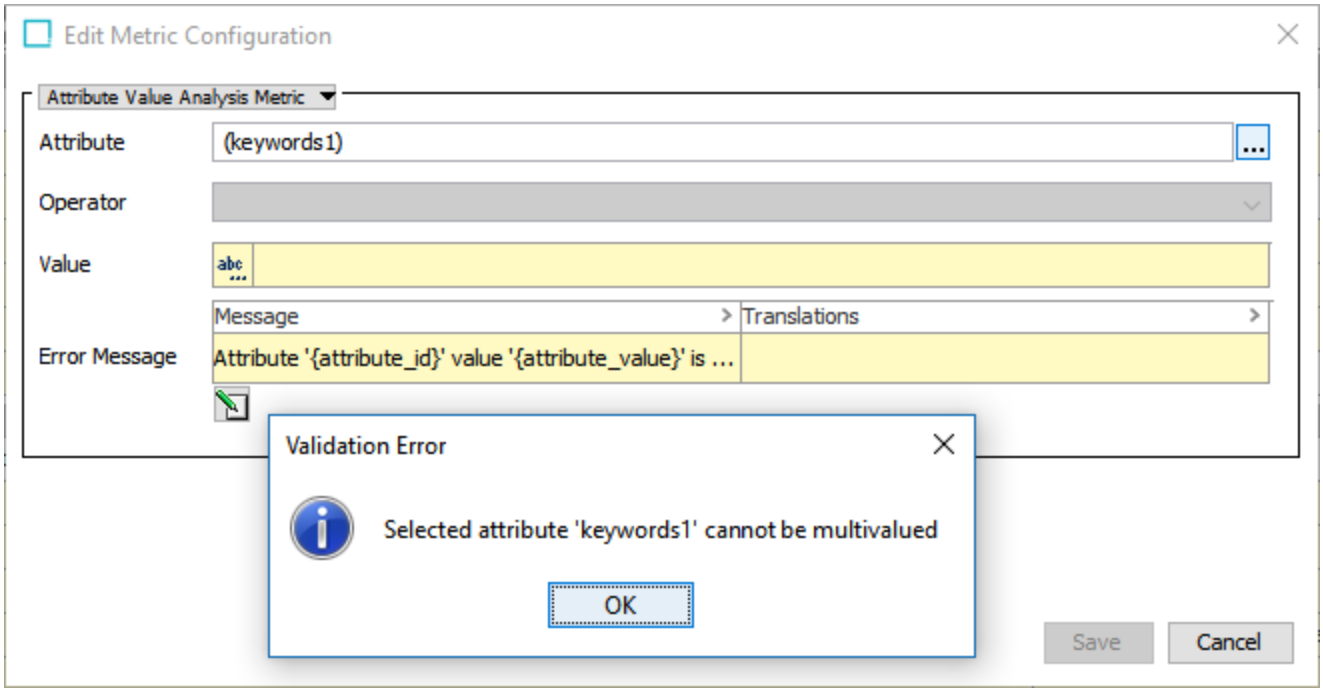
Once the metric is added to a sufficiency and the sufficiency is run, the results can be seen in the sufficiency panel within the Web UI. In the example below, the Attribute Value Analysis Metric (titled 'AttC') has been added to the sufficiency 'Product Measurement Requirements,' and returned a score of '100' because the attribute 'Length' value is equal to '6 in,' which was the value input into the metric. Because the returned score is '100' and not '0,' there is no accompanying error message.



Validation Errors

In certain instances validation errors can occur, depending on the attribute selected for analysis. A validation error (accompanied by an error message) will occur if:

- The provided value is not valid based on the selected attribute validation base type.
- The selected attribute is multivalued.



The following instances will also produce validation errors. However, these errors will not appear until the metric is run. The error message will appear in the Background Process Log of the Data Sufficiency Calculator event processor.

- The selected attribute has a unit but the value provided by the user in the metric configuration has no unit.
- The selected attribute has no unit but the value provided by the user in the metric configuration has a unit.
- The selected attribute and the value provided by the user in the metric configuration have different base unit conversion rules.
- The selected attribute's unit and/or the value's unit provided by the user in the metric configuration has no base unit conversion rule.

For more information regarding units, including base units and conversion rules, see the **Units** topic in the **System Setup** documentation.

Business Function Metrics

Business Function metrics use JavaScript business functions to evaluate data and return metric scores, and depending on the JavaScript, messages that provide more detailed information about the returned scores.

For information on business functions, see the **Business Functions** topic of the **Business Rules** documentation.

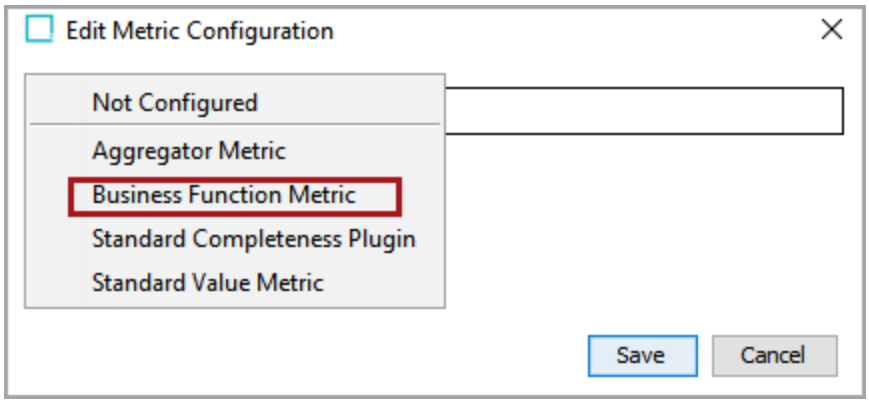
Business Function Metric Configuration

Before metrics can be created, a Setup Group must be configured to hold the metrics. For information on configuring a setup group for metrics, see the **Initial Setup for Metrics** topic in the **Metrics** documentation.

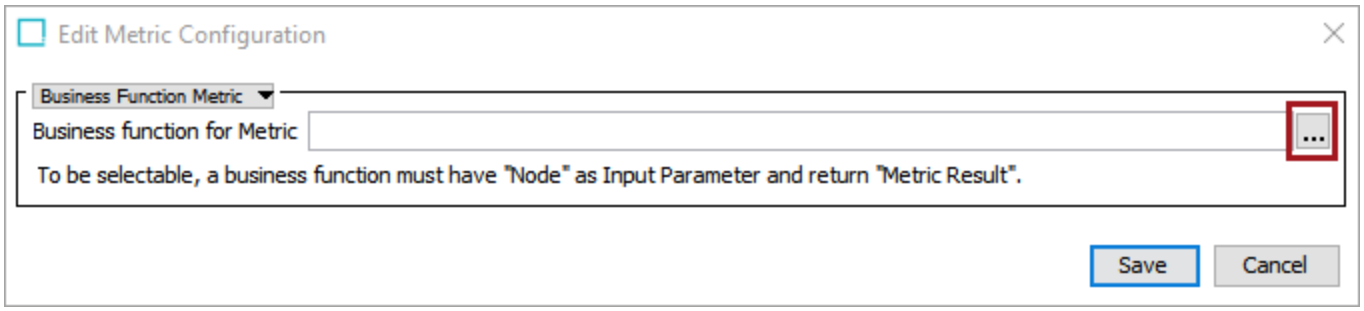
For information on creating a metric, see the **Creating and Editing Metrics** topic in the **Metrics** documentation.

To configure a business function metric:

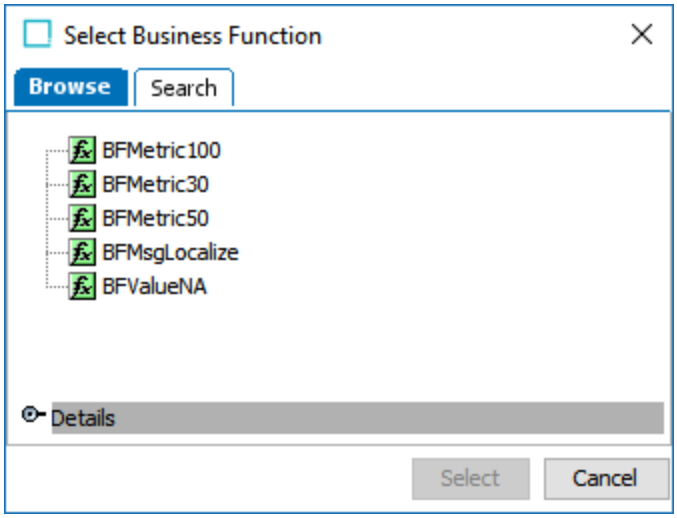
1. From the Edit Metric Configuration wizard, select 'Business Function Metric.'



2. Click the ellipsis button (...) by the **Business function for Metric** parameter.



3. Select the business function to include as part of the business function metric. To search for a specific business function, click the 'Search' tab and enter the search criteria.



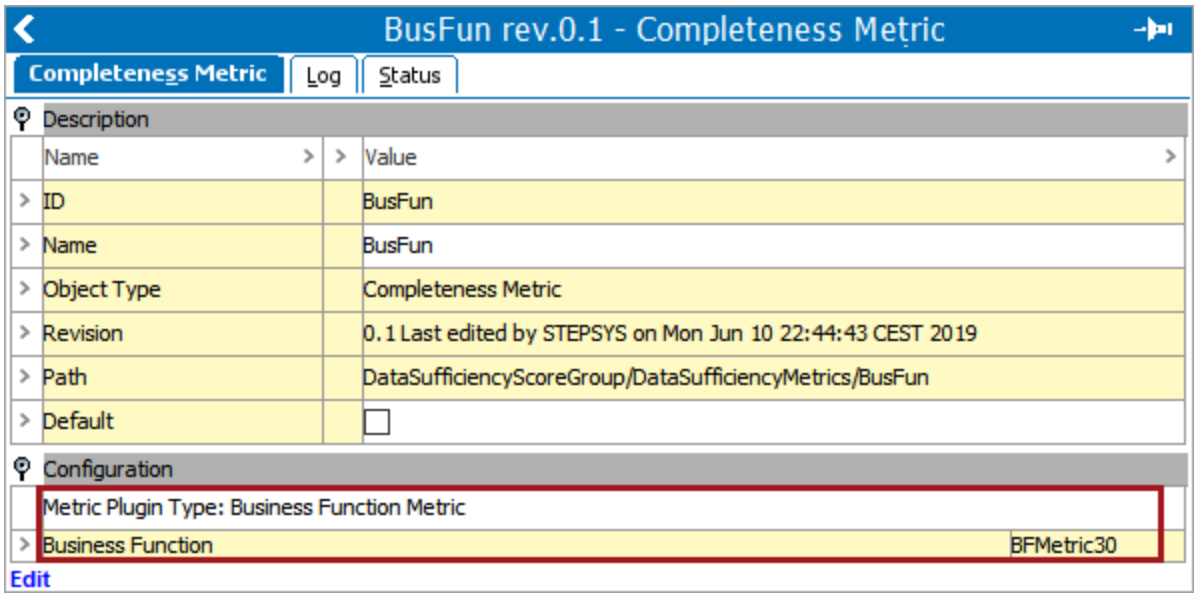
Note: For a business function to be selectable for a business function metric, it must have 'Node' as the input and 'MetricBusinessFunctionResult' as the return type.

4. Click 'Save' to close the 'Edit Metric Configuration' wizard.

The business function metric is configured and ready for use.

Important: The MetricBusinessFunctionResult of a JavaScript business function must be instantiated directly in the business function. It cannot be instantiated by a JavaScript library.

With the configured metric selected, the name of the metric plugin type (in this case, 'Business Function Metric') and the name of the selected business function will be listed under the Configuration flipper.


BusFun rev.0.1 - Completeness Metric
Completeness Metric
Log
Status
Description
Name
Value
ID
BusFun
Name
BusFun
Object Type
Completeness Metric
Revision
0.1 Last edited by STEPSYS on Mon Jun 10 22:44:43 CEST 2019
Path
DataSufficiencyScoreGroup/DataSufficiencyMetrics/BusFun
Default
Configuration
Metric Plugin Type: Business Function Metric
Business Function
BFMetric30
Edit

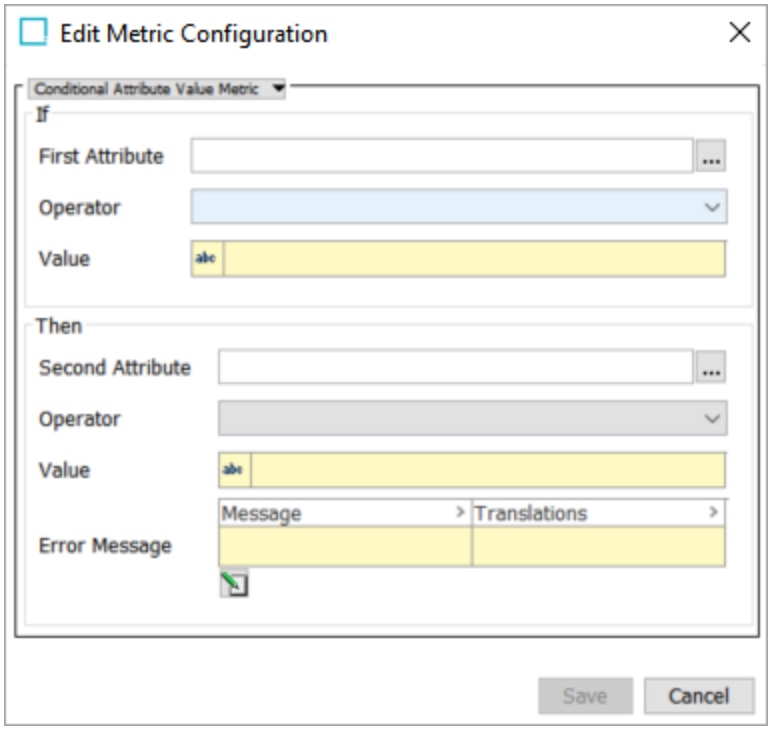
Conditional Attribute Value Metric

The Conditional Attribute Value Metric evaluates the value of a selected attribute based on an operator, and if that evaluation returns as true, then the metric will proceed and evaluate the value of another selected attribute based on an operator. The results of the evaluation (based on the selected operator for the second attribute) are either a '100' (true) or '0' (false).

Important: Metrics are used in conjunction with sufficiencies to determine the quality and completeness of a product's data. While this topic covers the functionality and configuration of the Conditional Attribute Value Metric, it is important that users are also familiar with sufficiencies and how they are combined with metrics. Information regarding sufficiencies can be found in the **Sufficiency Configuration Type** topic.

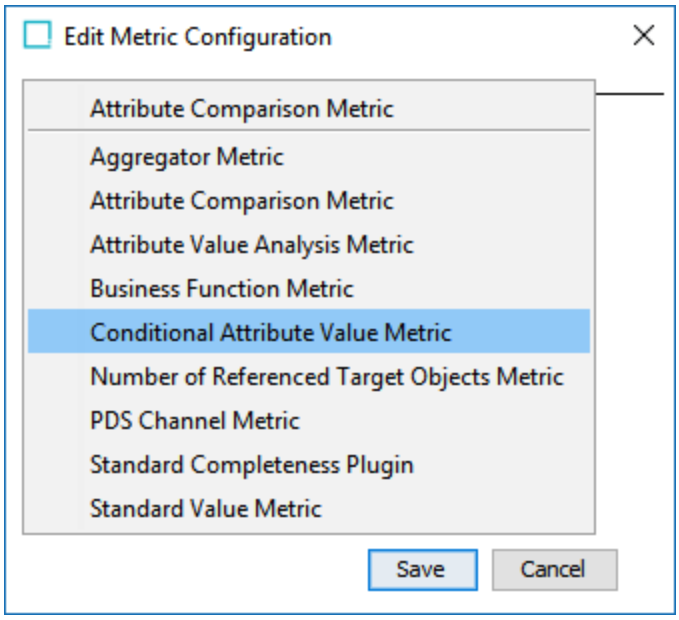
Prerequisites

Before metrics can be created, a Setup Group must be configured to hold them. For information on configuring a Setup Group for metrics, see the **Initial Setup for Metrics** topic in the **Metrics** documentation. For information on creating a metric, see the **Creating and Editing Metrics** topic in the **Metrics** documentation.



Conditional Attribute Value Metric configuration

1. From the Edit Metric Configuration wizard, select 'Conditional Attribute Value Metric.'



2. With the Conditional Attribute Value Metric selected, in the 'If' section of the metric, click the ellipsis button to select the first attribute, select an operator from the operator dropdown menu, and input a value into the 'Value' text field. Repeat these actions for the parameters within the 'Then' section of the metric.

   Once an operator is selected from the 'Operator' dropdown list, a default error message will appear in the 'Error Message' text field, based on the selected operator.

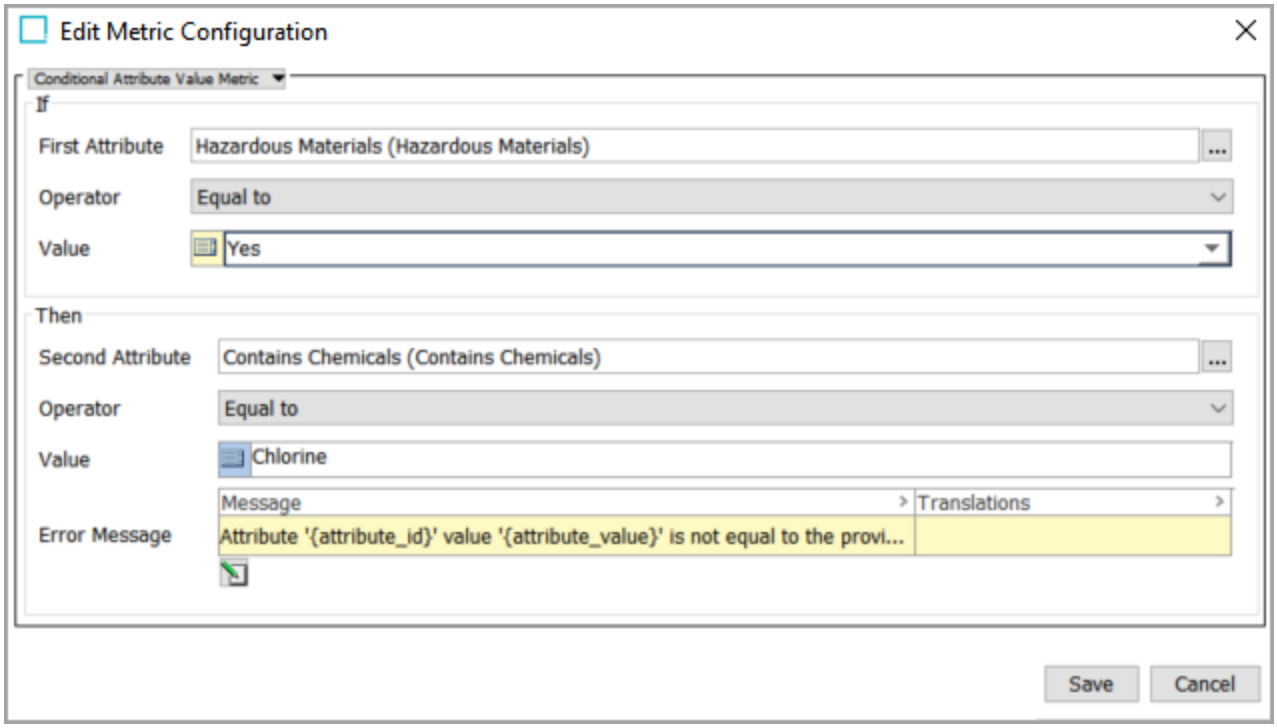
   Operator options include:

   - Is empty
   - Is not empty
   - Equal to
   - Not equal to
   - Contains
   - Regular expression
   - Greater than
   - Less than
   - Length is equal to
   - Length is greater than
   - Length is less than

   Note: If the attribute selected has a validation base type that does not allow for numbers, the 'Greater than' and 'Less than' operator options will not be available. If 'Is empty' or 'Is not empty' is selected as an operator, the Value text field will be disabled.

   In the example below, in the 'If' section of the metric, the user has selected the attribute 'Hazardous Materials' for the first attribute, 'Equal to' for the operator, and input 'Yes' in the 'Value' text field. In the 'Then' section of

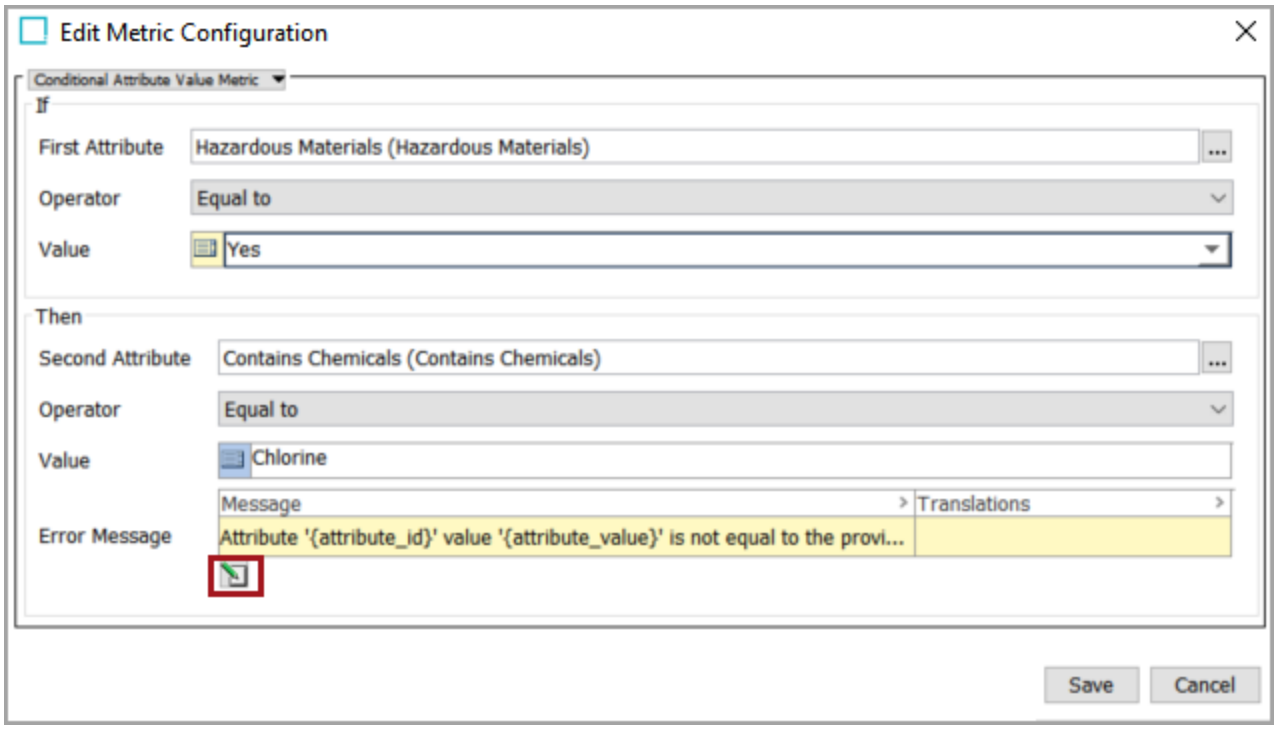
the metric, the user has selected 'Contains Chemicals' for the second attribute, 'Contains' for the operator, and input 'Chlorine' in the 'Value' text field.



When run, this metric will first check the attribute value for 'Hazardous Materials;' if the value is equal to 'Yes,' it will then check the value for the attribute 'Contains Chemicals.' If the attribute value contains the word 'Chlorine,' a score of '100' will be returned. If the word 'Chlorine' is not present in the attribute value, it will return a score of '0' and display a message in the 'Message Preview' text field based on the 'Equal to' operator. If the first attribute 'Hazardous Materials' is not equal to 'Yes,' then the metric will not check the value for the second attribute.

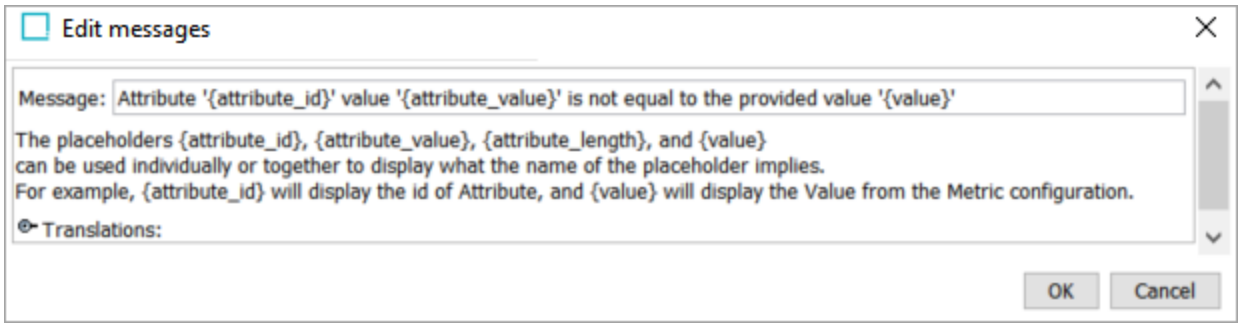
Note: If the user selects an attribute that is not valid for a product, the metric will be considered non-applicable for that product and no data quality information will be produced when the Sufficiency Data Calculator event processor is run.

3. To edit the error message and/or add additional translations, click on the Error Message editor button.

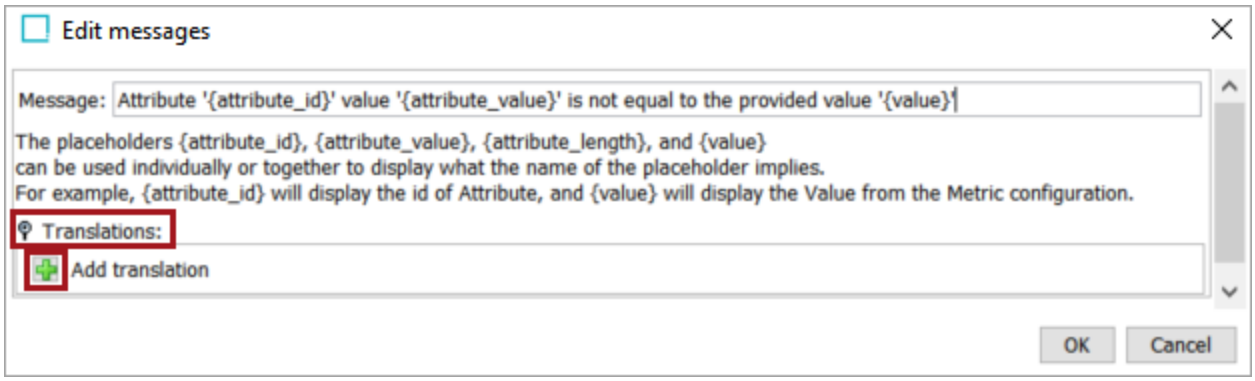


The 'Edit messages' window appears. Note that the 'Message' text field is editable. Descriptive text below the text field details how the placeholders included in the error message work in order to create an error message. Although the default error message (or something similar) is suggested, users are able to edit all or part of the error message, depending on their individual needs.

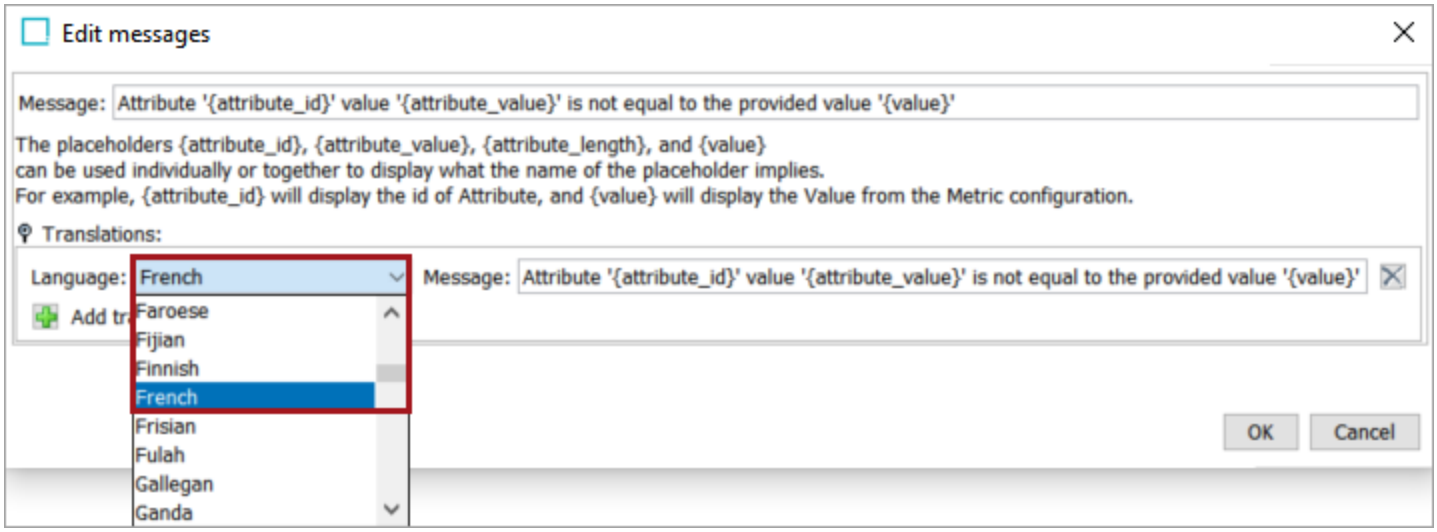
Note: Although the error message can be edited by the user, the placeholders themselves (e.g., attribute_id, attribute_value) cannot be edited. If the placeholders are edited, the user will receive a validation error message when attempting to save the edited message. When the metric is run, placeholders are replaced by the ' real values,' e.g., 'attribute_id' would be replaced with the name of the attribute that the metric is evaluating.



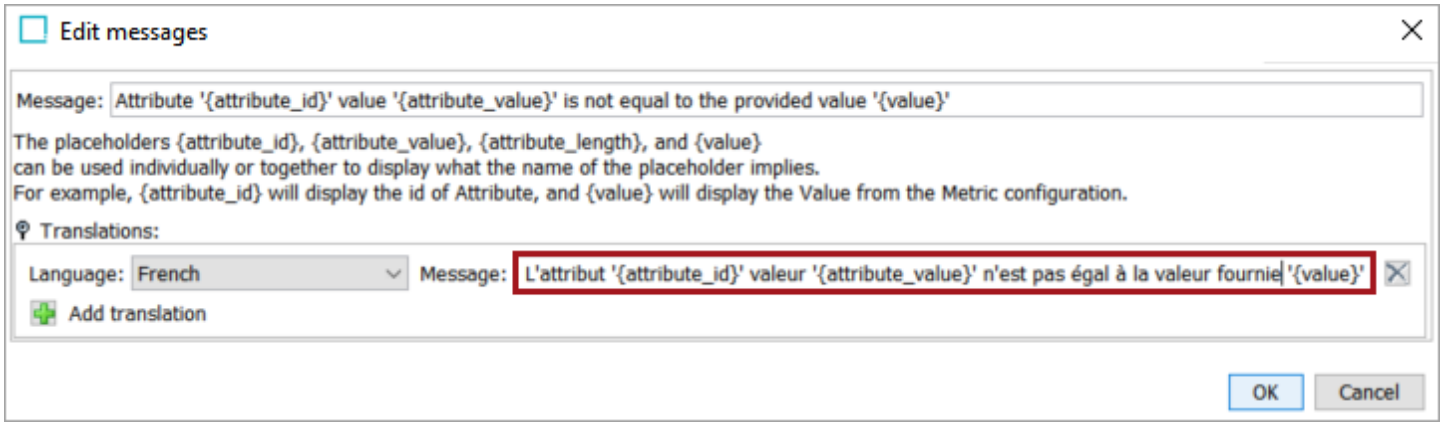
4. Select the 'Translations' flipper and click the green plus sign located next to 'Add translation.'



5. Click on the arrow located in the 'Language' dropdown menu; in the example below, the user has opened the dropdown menu and selected 'French.'



6. Once a language is selected, edit the message to reflect the language specified for translation.



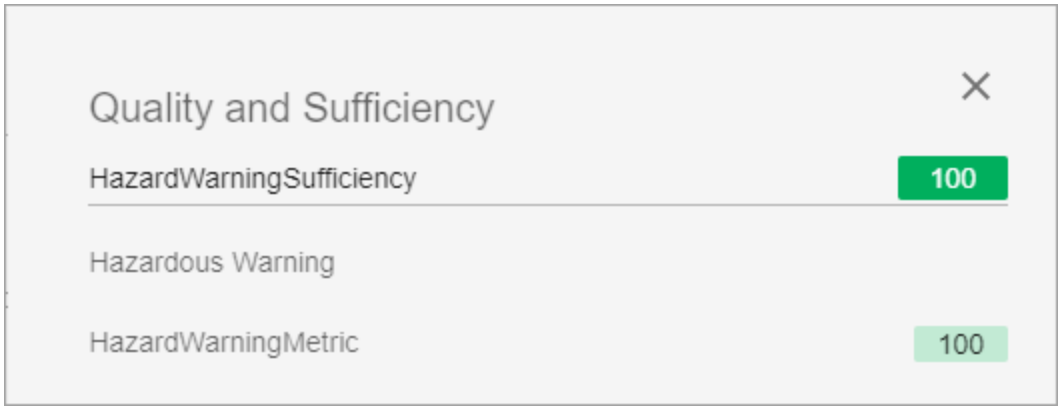
7. Click the green plus sign to add another translation or 'OK' to close the 'Edit messages' window.

8. Click 'Save' to close the Edit Metric Configuration wizard.

The metric is saved.

Note: The localized error messages are displayed by using locales; they are not context-dependent. For more information about locales, see the **Localization** topic in the **Administrative Portal** documentation. For more information about localized messages, see the **Localized Messages for JavaScript Business Rules** topic in the **Business Actions** documentation.

Once the metric is added to a sufficiency and the sufficiency is run, the results can be seen in the sufficiency panel within the Web UI. In the example below, the Conditional Attribute Value Metric (titled 'HazardWarningMetric') has been added to the sufficiency 'HazardWarningSufficiency,' and returned a score of '100' because the attribute 'Hazardous Materials' value is equal to 'Yes' and the attribute 'Contains Chemical' value is equal to 'Chlorine.' Because the returned score is '100' and not '0,' there is no accompanying error message.

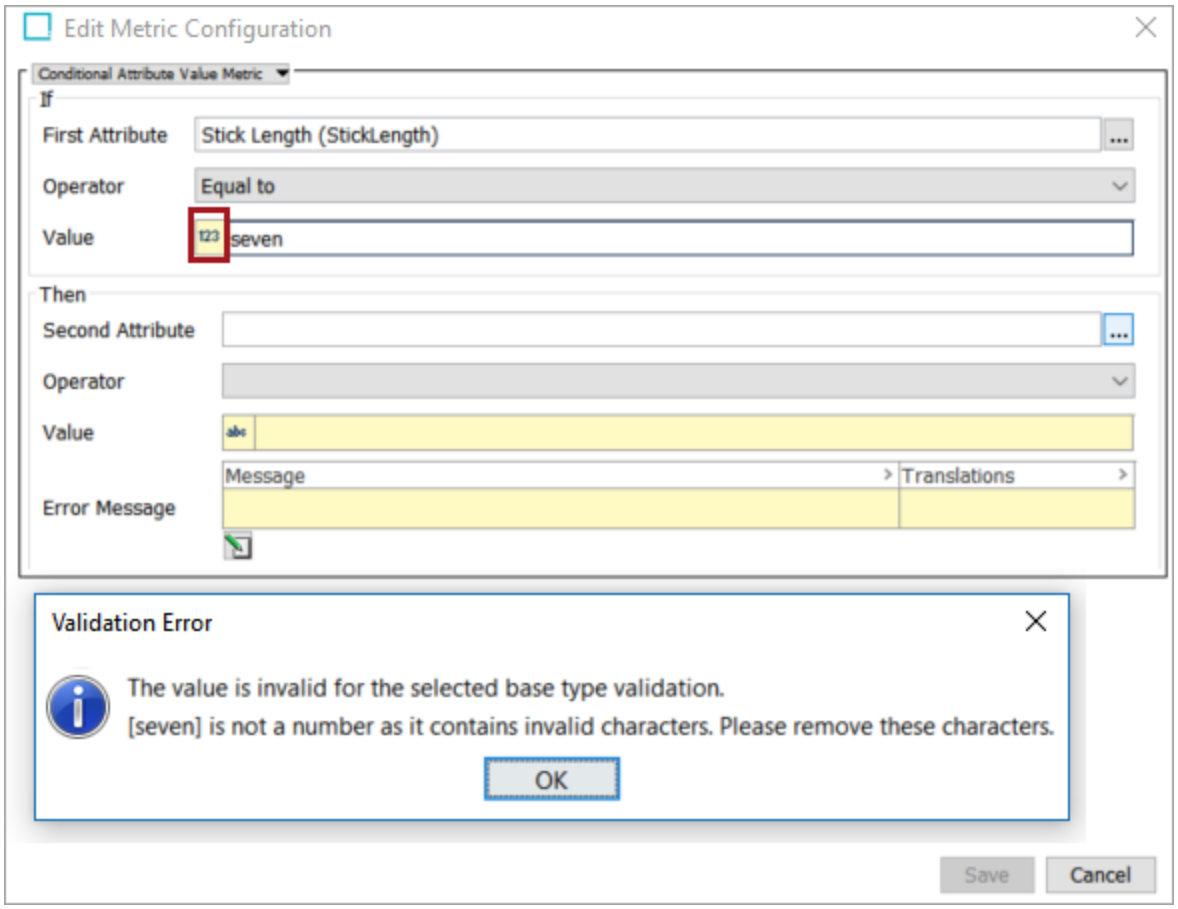


Validation Errors

In certain instances, validation errors can occur as a result of the attributes selected for analysis. A validation error (accompanied by an error message) will occur if:

- The provided value is not valid based on the selected attribute validation base type.
- The selected attribute is multivalued.

In the example below, the user has input the word 'seven' into the 'Value' text field, but because the attribute selected for analysis has a number validation base type (identified by the highlighted '123'), the provided value must contain numbers only. Because of the incompatibility, a validation error occurs, with an error message that describes the reason for the error.



The following instances will also produce validation errors. However, these errors will not appear until the metric is run. The error message will appear in the Background Process Log of the Data Sufficiency Calculator event processor.

- A selected attribute has a unit but the value provided by the user in the metric configuration has no unit.
- A selected attribute has no unit but the value provided by the user in the metric configuration has a unit.
- A selected attribute and the value provided by the user in the metric configuration have different base unit conversion rules.
- A selected attribute's unit and/or the value's unit provided by the user in the metric configuration has no base unit conversion rule.

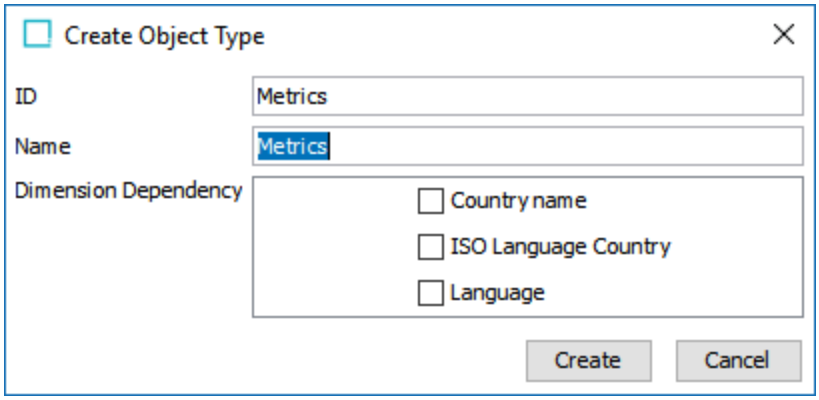
For more information regarding units, including base units and conversion rules, see the **Units** topic in the **System Setup** documentation.

Initial Setup for Metrics

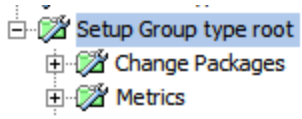
Before metrics can be created, a Setup Group must be configured to hold them.

Note: A Setup Group is required if the user wants to migrate completeness scores to a Description attribute. For more information, see the **Migrating Completeness Scores to Description Attributes** section of the **Data Profiling** documentation.

1. In **System Setup**, expand **Object Types & Structures**.
2. Right-click **Setup Group type root**, and then choose **New Object Type**.
3. Enter an **ID**, and a **Name** such as 'Metrics,' and then click **Create**.

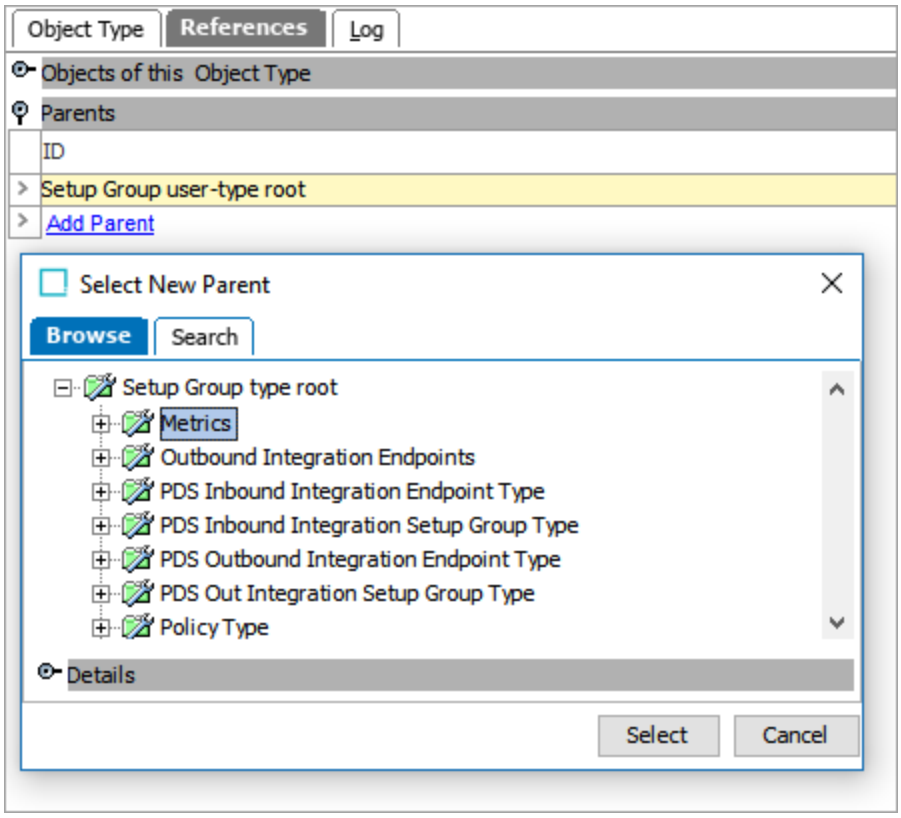


The **Setup Group** appears as a child below **Setup Group type root**.



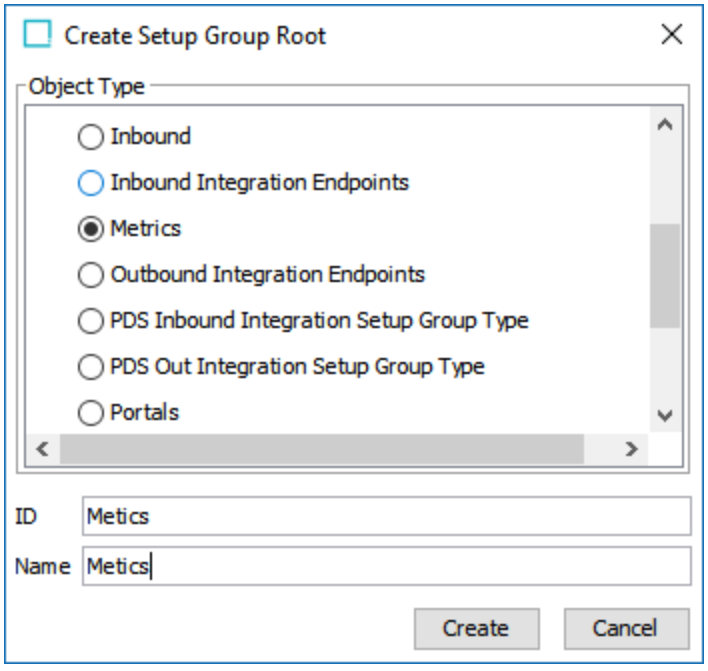
Next, you have to link the Metric object type to a the Setup Group.

4. In **Object Types & Structures**, expand **Basic Object Types**, then select **Completeness Metric**.
5. On the **References** tab, click **Add Parent**.
6. In the **Select New Parent** dialog, select the Setup Group just created, then click **Select** to make it a valid parent.



Next, create an instance of the Metric object.

7. On the **System Setup** tab, click **Maintain**, mouse over **Insert**, and then select **Setup Group Root**.
8. In the **Create Setup Group Root** dialog, select the new object type.



9. Enter an **ID** and a **Name**, then click **Create**.

A Setup Group is created as a node in the System Setup hierarchy, and you can start creating metrics.

For more information about creating metrics, see the **Creating and Editing Metrics** section of the documentation.

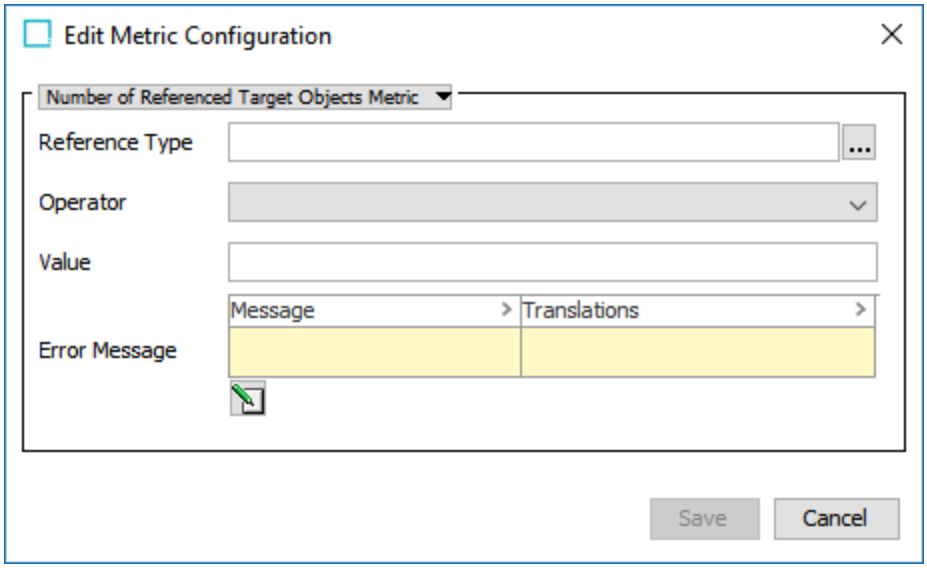
Number of Referenced Target Objects Metric

The Number of Referenced Target Objects Metric detects the number of target objects of a selected reference type for a given product, and based on an operator selected by the user, returns either a score of '100' (true) or '0' (false). Users are able to create an error message for instances when the metric returns a score of '0;' this message is meant to detail the reason for the '0' score and can have multiple translations. For more information on reference types, see the **Reference Types** topic in the **Reference and Link Types** documentation.

Important: Metrics are used in conjunction with sufficiencies to determine the quality and completeness of a product's data. While this topic covers the functionality and configuration of the Number of Referenced Target Object Metrics, it is important that users are also familiar with sufficiencies and how they are combined with metrics; information regarding sufficiencies can be found in the **Sufficiency Configuration Type** documentation.

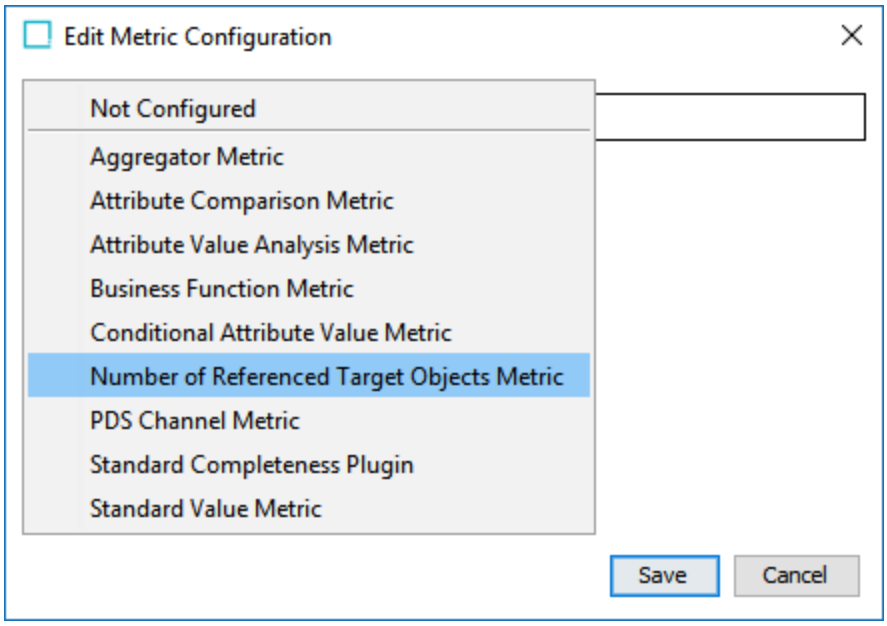
Prerequisites

Before metrics can be created, a Setup Group must be configured to hold them. For information on configuring a setup group for metrics, see the **Initial Setup for Metrics** topic in the **Metrics** documentation. For information on creating a metric, see the **Creating and Editing Metrics** topic in the **Metrics** documentation.



Number of Referenced Target Objects Metric Configuration

1. From the Edit Metric Configuration wizard, select 'Number of Referenced Target Objects Metric.'

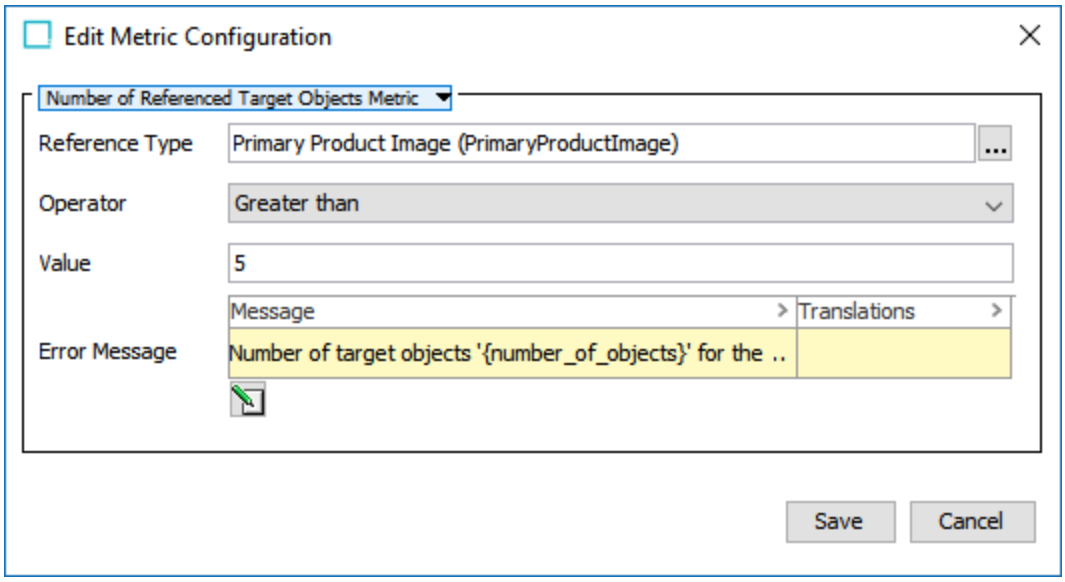


2. With the Number of Referenced Target Objects Metric selected, click the ellipsis button (...) to select the reference type that you want to evaluate, select an operator from the 'Operator' dropdown list, and input a numerical value into the 'Value' field. Once an operator is selected from the 'Operator' dropdown list, a default error message will appear in the 'Error Message' text field, based on the selected operator.

   Operator options include:

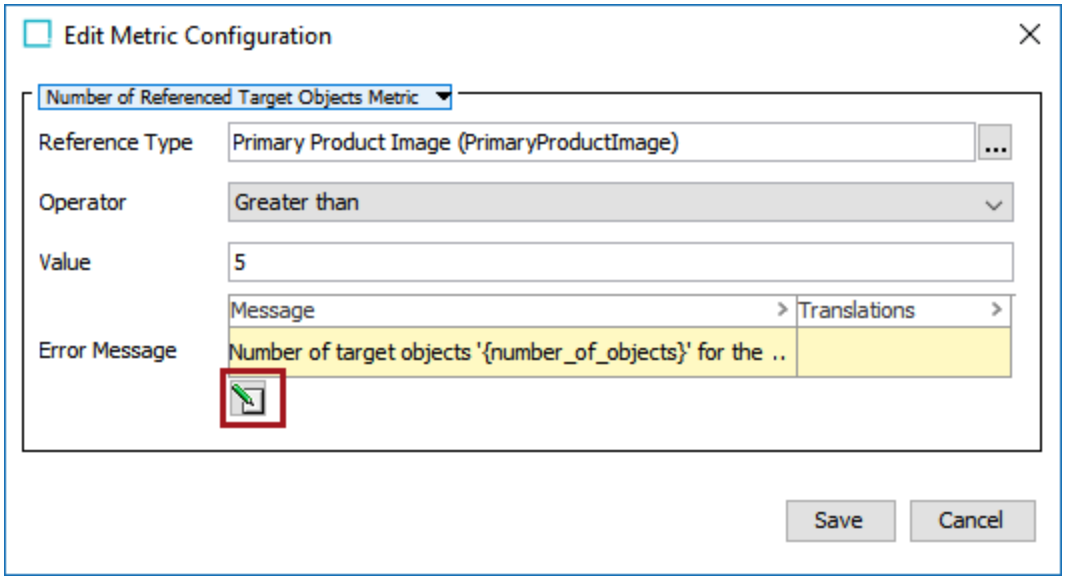
   - Equal to
   - Not Equal to
   - Greater than
   - Less than

   In the example below, the user has configured the metric to determine whether a product has more than '5' Primary Product Image references. If the product does have more than five product image references, the metric will return a score of '100' (true). If there are five or less primary product image references, the metric will return a score of '0' (false), and will be accompanied with a message, as displayed in the 'Error Message' text field.



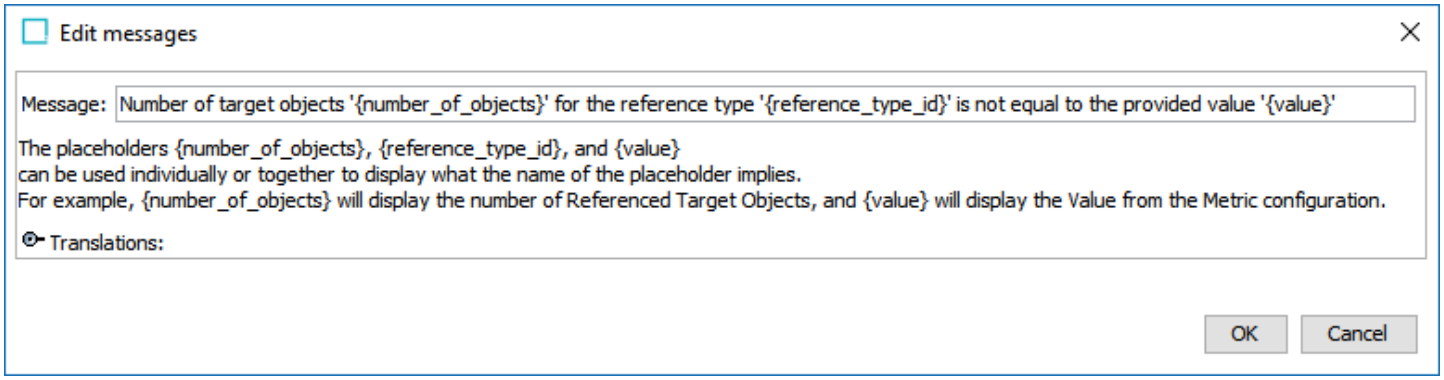
Note: If the user selects a reference type that is not valid for a product, the metric will be considered non-applicable for that product and no data quality information will be produced when the Sufficiency Data Calculator event processor is run. Additionally, only integers are valid for the 'Value' field; non-integer characters input into the 'Value' field will result in a validation error when the user attempts to save the metric.

3. To edit the error message and/or add additional translations, click on the 'Error Message' editor button.



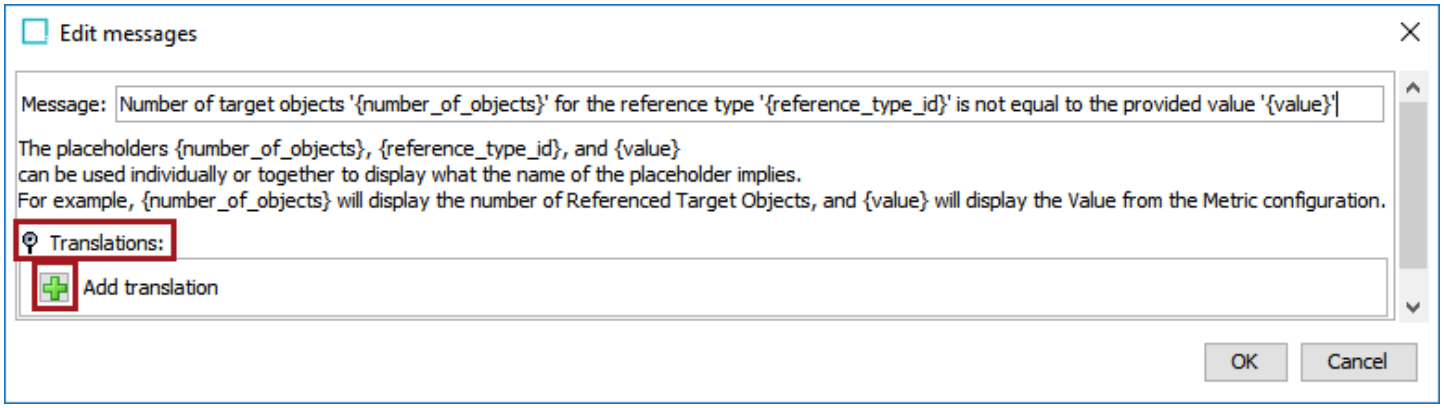
The 'Edit messages' window appears. Note that the 'Message' text field is editable. Descriptive text below the text field details how the placeholders included in the error message work in order to create an error

message. Although the default error message (or something similar) is suggested, users are able to edit all or part of the error message, depending on their individual needs.

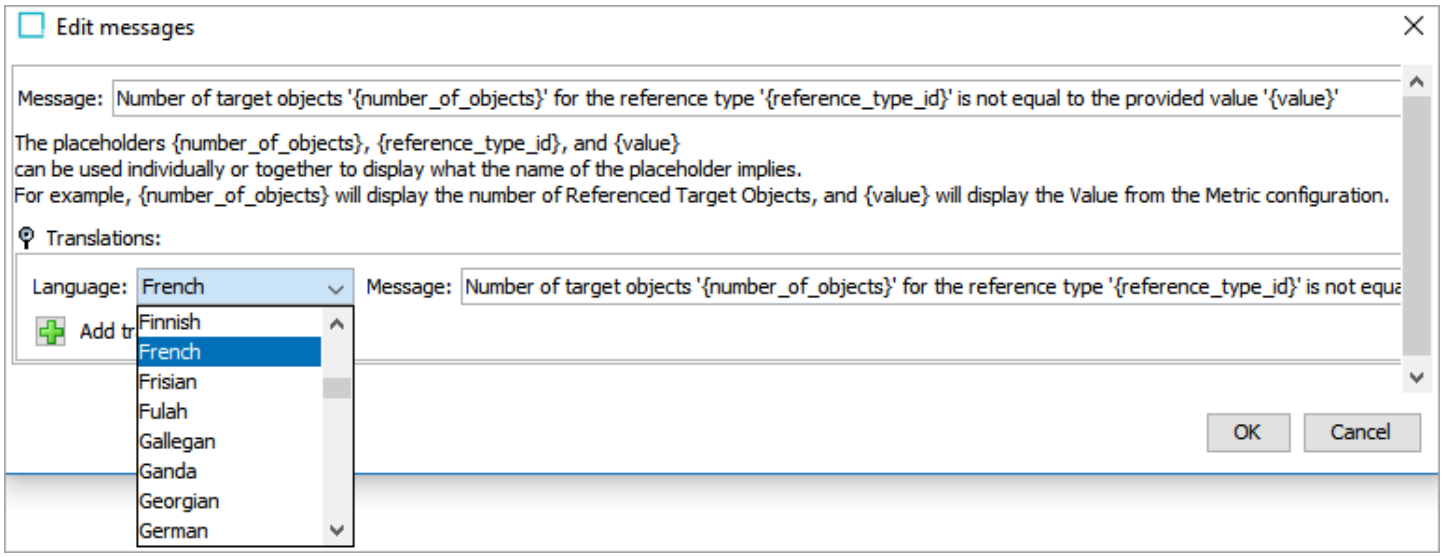


Note: Although the error message can be edited by the user, the placeholders themselves (e.g., number_of_objects) cannot be edited. If the placeholders are edited, the user will receive a validation error message when attempting to save the edited message. When the metric is run, placeholders are replaced by the 'real values,' e.g., 'number_of_objects' would be replaced with the number input into the Value field (in the example above, '5').

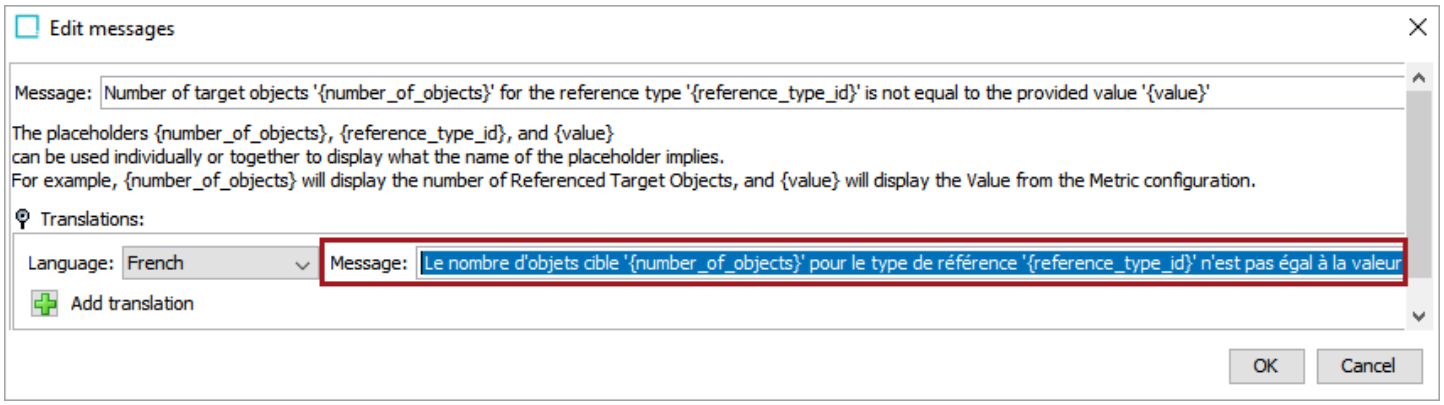
4. Select the 'Translations' flipper and click the green plus sign located next to 'Add translation.'



5. Click on the arrow located in the 'Language' dropdown menu; in the example below, the user has opened the dropdown menu and selected 'French.'



6. Once a language is selected, edit the message to reflect the language specified for translation.

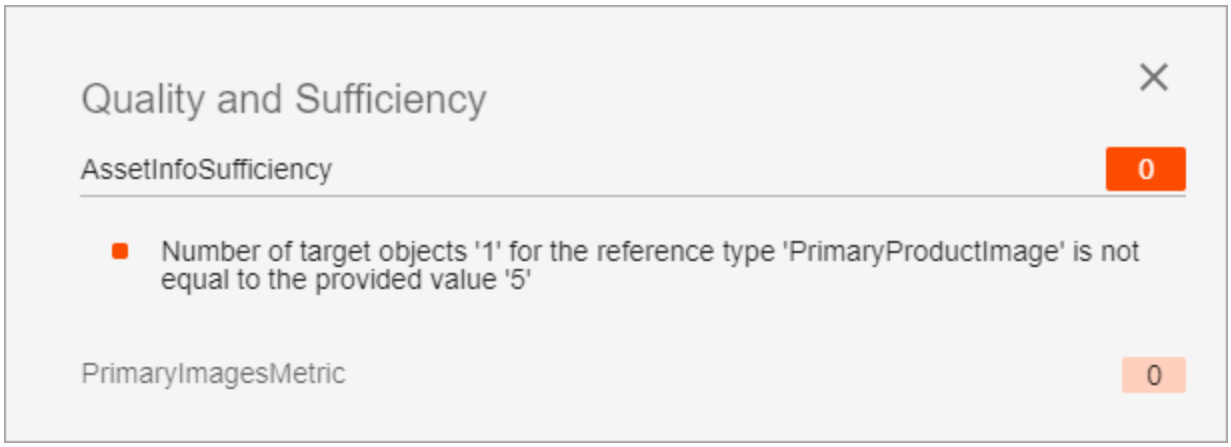


7. Click the green plus sign to add another translation or 'OK' to close the 'Edit messages' window.
8. Click 'Save' to close the Edit Metric Configuration wizard.

The metric is saved.

Note: Translation error messages are displayed by using locales; they are not context-dependant. For more information about locales, see the **Localization** topic in the **Administrative Portal** documentation. For more information about localized messages, see the **Localized Messages for JavaScript Business Rules** topic in the **Business Actions** documentation.

Once the metric is added to a sufficiency and the sufficiency is run, the results can be seen in the sufficiency panel within the Web UI. In the example below, the Referenced Target Object Metric (for this example, titled 'PrimaryImagesMetric') has been added to the sufficiency 'AssetInfoSufficiency,' and returned a score of '0' along with an error message that details the reason for the returned score.


Quality and Sufficiency
AssetInfoSufficiency
0
Number of target objects '1' for the reference type 'PrimaryProductImage' is not equal to the provided value '5'
PrimaryImagesMetric
0

PDX Channel Metric

The PDX (Product Data Exchange) Channel Metric gives users the ability to measure the quality and completeness of a product's data compared to the data standard of a retail channel in PDX. By obtaining this information, users are able to determine whether the product is ready to be syndicated to the retail channel before actually submitting the product for syndication.

To view the data produced by the PDX Channel Metric in the sufficiency panel in the Web UI, the add-on component 'PDX-sufficiency-link' must be installed on your system. Instructions for installing components can be found in the SPOT Program topic in the System Administration documentation.

To use the PDX Channel Metric, the following PDX license must also be enabled.

- X.Adapter.ProductDataSyndication

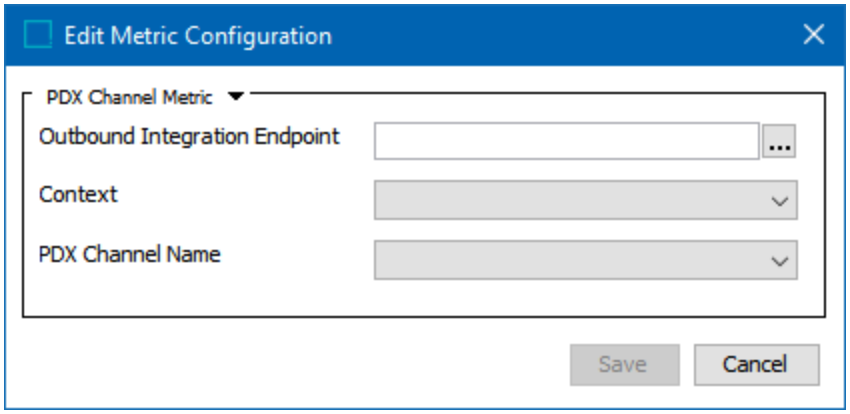
Contact your account manager to enable the license for your system.

Important: Metrics are used in conjunction with sufficiencies to determine the quality and completeness of a product's data. While this topic covers the functionality and configuration of the PDX Channel Metric, it is important that users are also familiar with sufficiencies and how they are combined with metrics; information regarding sufficiencies can be found in the **Sufficiency Configuration Type** topic.

Prerequisites

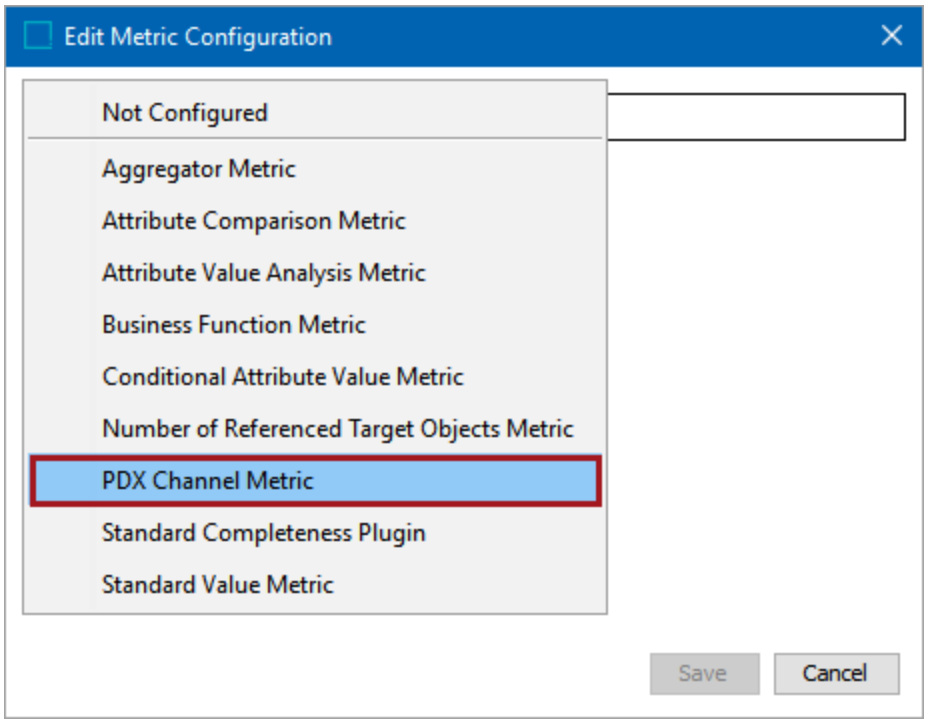
Before metrics can be created, a Setup Group must be configured. For information on configuring a Setup Group for metrics, see the **Initial Setup for Metrics** topic in the **Metrics** documentation. For information on creating a metric, see the **Creating and Editing Metrics** topic in the **Metrics** documentation.

Prior to configuring a PDX Channel Metric, a PDX Outbound Integration Endpoint (OIEP) must be configured. The relation between the PDX Channel Metric and the PDX OIEP is explained in further detail in step 5 of this topic. For information on configuring a PDX OIEP, see the **PDX Outbound Integration Endpoint Configuration** topic in the **Product Data Exchange** documentation.

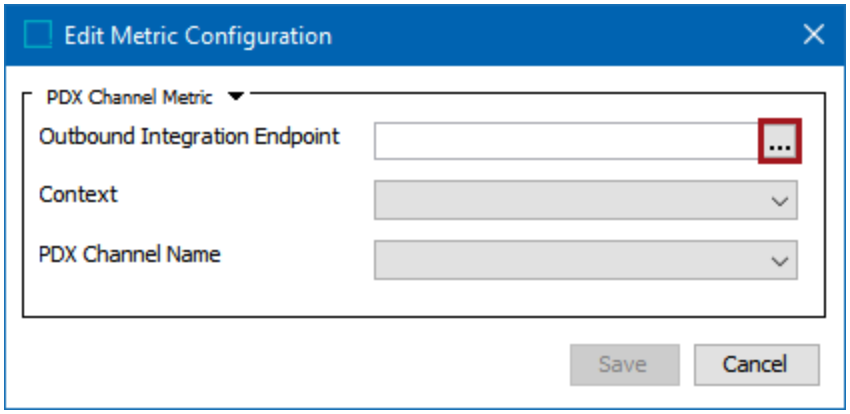


PDX Channel Metric configuration

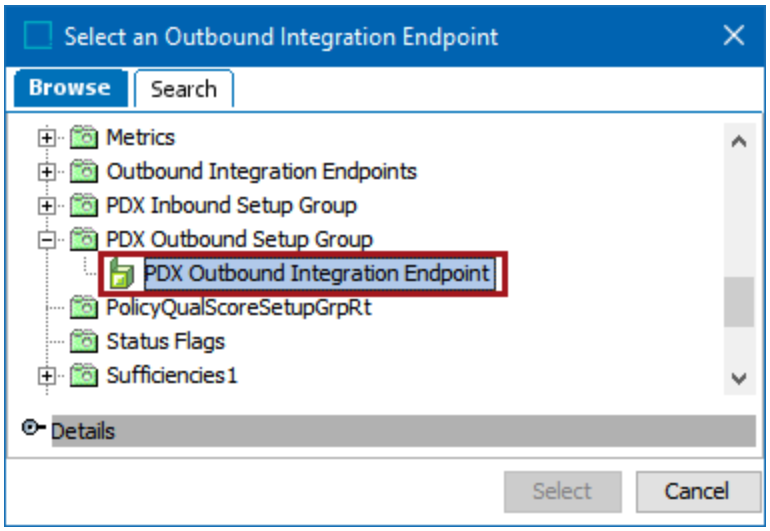
1. From the Edit Metric Configuration wizard, select 'PDX Channel Metric.'



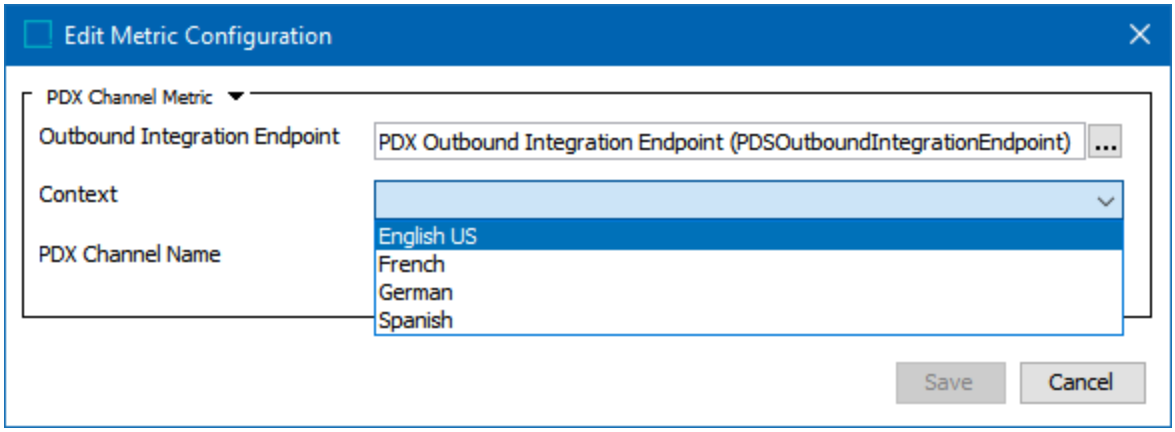
2. With the PDX Channel Metric selected, click the ellipsis button (...) located next to the Outbound Integration Endpoint.



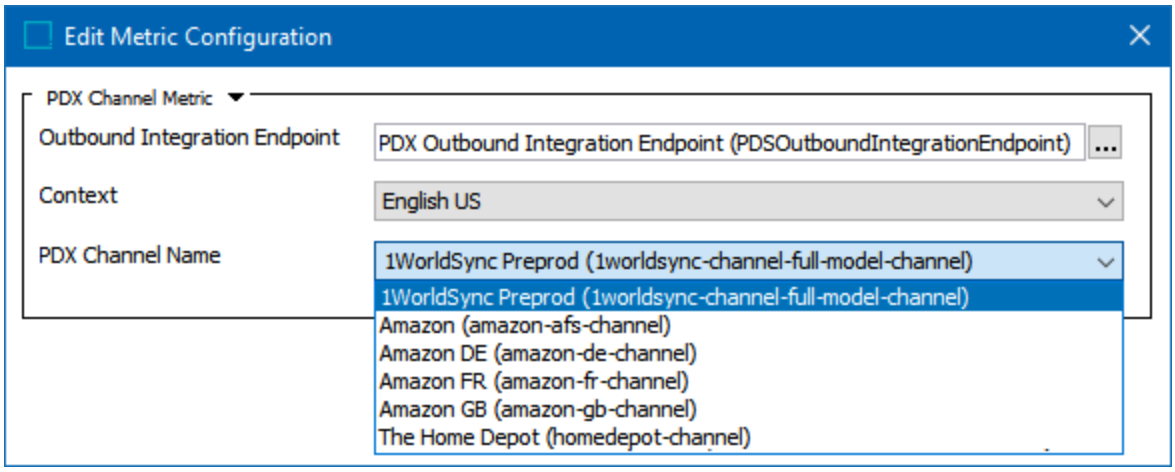
3. Select the PDX Outbound Integration Endpoint you would like to use for this metric and click 'Select.' In this example, 'PDX Outbound Integration Endpoint' has been selected.



4. From the Context dropdown list, select the context that you would like to use for the metric. In the example below, the user has selected 'English US.'



5. From the PDX Channel Name dropdown list, select the PDX channel name that you would like to use for the metric. In the example below, the user has selected the '1WorldSync PreProd' channel.

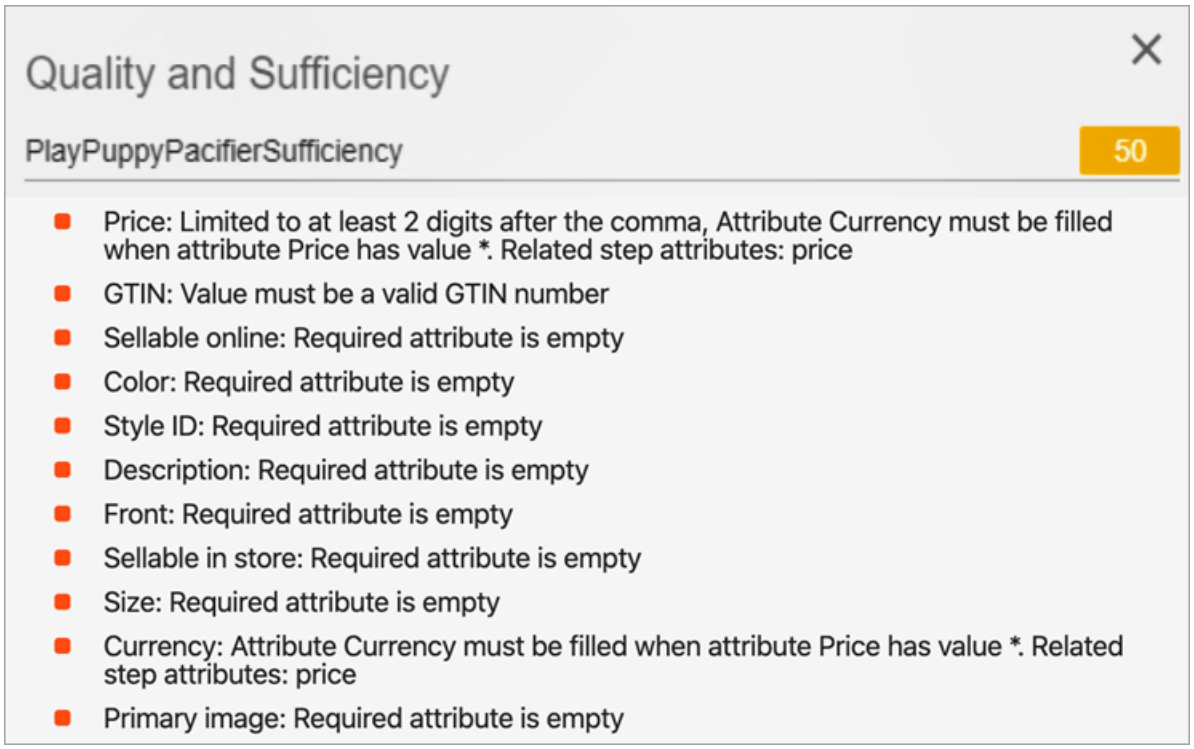


Note: Available context options are dependent on the 'Contexts' setting in the PDX OIEP configuration. Available PDX channels are determined within the PDX system, e.g., any channels available in the PDX system will be included in the 'PDX Channel Name' dropdown menu as a selectable option. For more information on configuring the PDX Outbound Integration Endpoint, see the **PDX Outbound Integration Endpoint Configuration** topic in the **Product Data Exchange** documentation. For more information regarding channel options within PDX, contact your PDX representative.

6. Click 'Save' to close the Edit Metric Configuration wizard.

The PDX Channel Metric is saved.

Once the metric is run within a sufficiency, the results can be seen in the sufficiency panel within the Web UI. In the example below, the PDX Channel Metric has returned details regarding several attributes that are missing mandatory values, have incorrect formatting of values, etc., based on the requirements of the selected channel. With this information, users can provide values for the mandatory attributes and run the sufficiency again to re-assess the metric results.


Quality and Sufficiency
PlayPuppyPacifierSufficiency
50
Price: Limited to at least 2 digits after the comma, Attribute Currency must be filled when attribute Price has value *. Related step attributes: price
GTIN: Value must be a valid GTIN number
Sellable online: Required attribute is empty
Color: Required attribute is empty
Style ID: Required attribute is empty
Description: Required attribute is empty
Front: Required attribute is empty
Sellable in store: Required attribute is empty
Size: Required attribute is empty
Currency: Attribute Currency must be filled when attribute Price has value *. Related step attributes: price
Primary image: Required attribute is empty

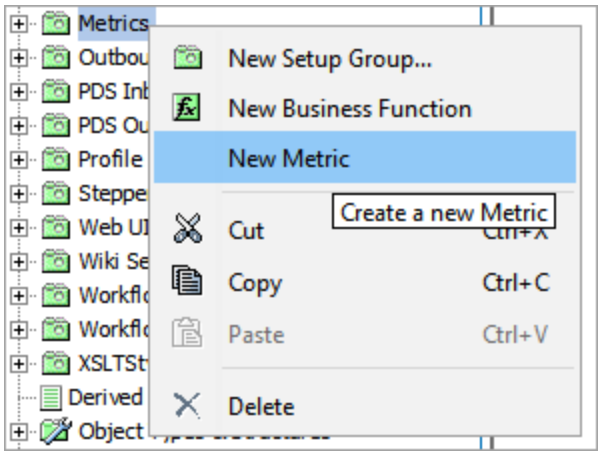
Creating and Editing Metrics

Once initial setup is complete, metrics can be created.

For more information on initial setup steps, see the **Initial Setup for Metrics** section of this documentation.

Creating a Metric Object

1. In **System Setup**, right-click the **Setup Group Root** that holds metrics, and then click **New Metric**.

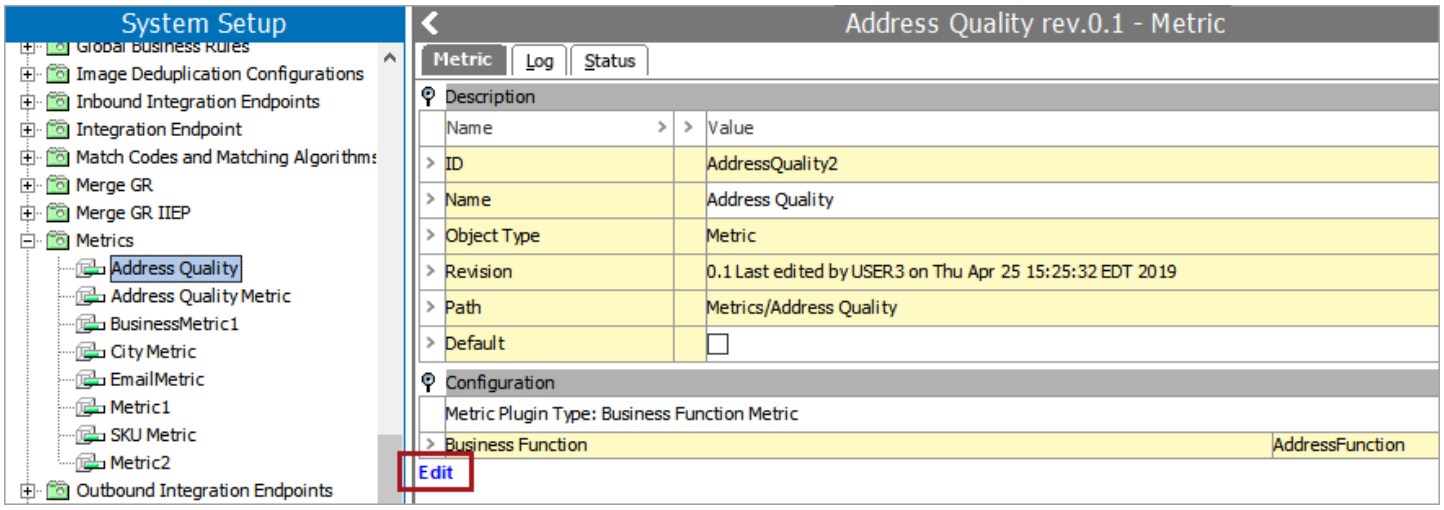


2. Enter an **ID** and a **Name** for the metric, and then click create. The metric is added as a child below the selected **Setup Group Root**.

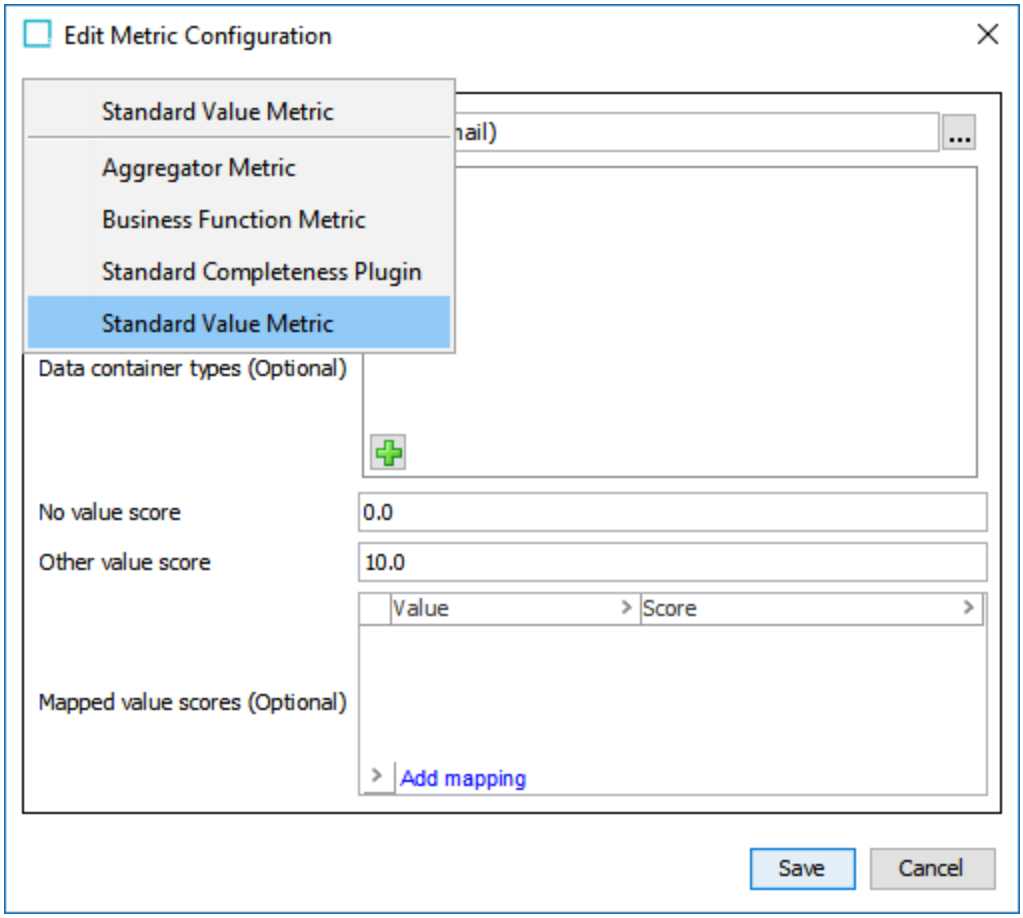
Configuring a Metric

Once the metric object has been created, a metric type must be specified and configured.

1. Select the metric that you just created, and then, on the **Metric** tab, click **Edit**.



2. In the **Edit Metric Configuration** editor, select the relevant metric type from the dropdown. Options include: Aggregator, Business Function, Completeness, and Value. Each metric type requires different configuration steps.



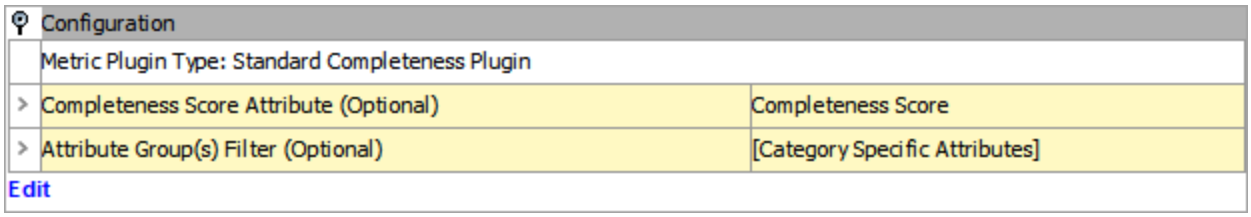
For more information on configuring an Aggregator Metric, see the **Aggregator Metrics** section of this documentation.

For more information on configuring a Business Function Metric, see the **Business Function Metrics** section of this documentation.

For more information on configuring a Completeness Metric, see the **Completeness Metrics** section of this documentation.

For more information on configuring a Value Metric, see the **Value Metrics** section of this documentation.

3. Once the metric type has been specified and configured, click **Save**. You can view the configuration settings in the **Configuration** area of the **Metric** tab.



Set a Default Metric

When a metric is set as the default metric, the completeness meter shown on an object's editor tab displays the value from the default metric.

1. In **System Setup**, expand the **Setup Group Root** that holds metrics, and then select the metric you want to apply as default.
2. On the **Metric** tab, in the **Description** area, select **Default**.

Completeness Metrics

Completeness metrics define what data is considered when evaluating the completeness of an object. STEP allows the use of multiple completeness metrics simultaneously. This is useful, for example, if multiple channel specific completeness measurements are required. An attribute can be created to hold completeness scores for a specific completeness metric, thus offering a means to display different completeness measurements for the same object. Additionally, multiple metrics can use the same completeness score attribute.

Note: A completeness metric does not have to be an attribute with associated completeness scores. In these cases, all data objects (attributes and references) are part of the completeness calculation and have an equal score.

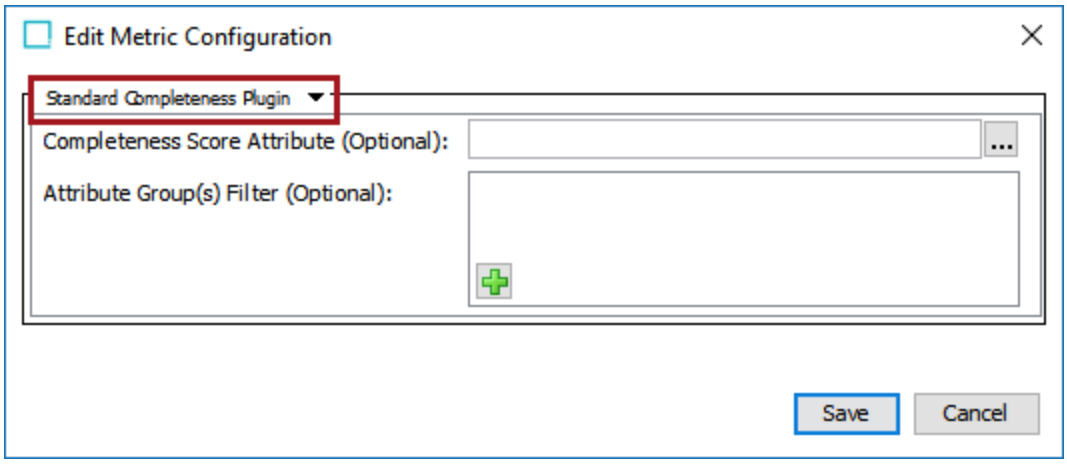
Configuration

1. Optionally, when creating a completeness metric, create an attribute that can hold the completeness score values.

   Note that if you want to migrate legacy completeness values stored in a special system attribute to a Description attribute, the attribute must meet a series of requirements. For more information, see the **Migrating Completeness Scores to Description Attributes** section of the **Data Profiling** documentation.

   If an attribute is not going to be used for migrating completeness scores, it does not have to meet these requirements.

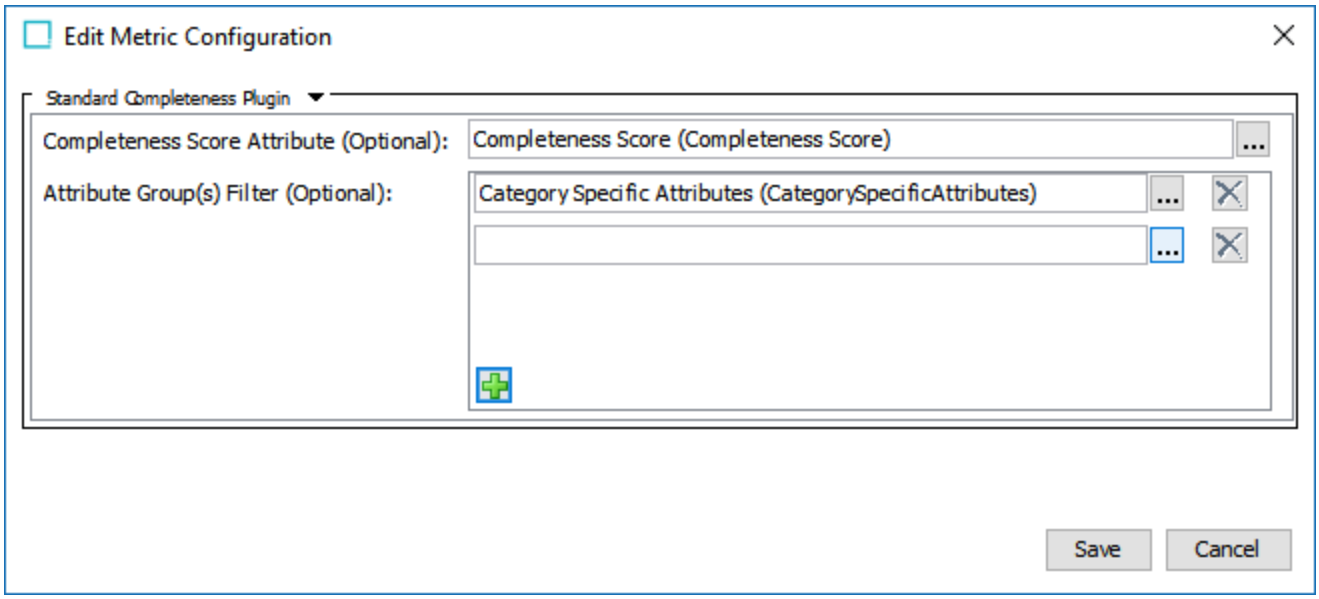
2. To configure a completeness metric, open the editor for the desired metric object, and from the dropdown select **Standard Completeness Plugin**.



3. The **Completeness Score Attribute** field allows you to select which attribute will hold the completeness scores for the completeness metric, if applicable. If nothing is specified, all elements of the object that could

contribute to the overall object completeness score will, and they are all weighed evenly.

The Attribute Group(s) Filter specifies which attributes and references / links should be considered in the completeness calculation.



The Completeness Plugin provides the following options:

Selection	Description
No attribute or attribute groups are selected	All data objects (attributes and references) are part of the completeness calculation, and they have an equal score.
An attribute is selected	Only completeness scores from the selected attribute are used for the completeness calculation.
One or more attribute groups are selected	Only completeness scores from data objects in the selected attribute group(s) are used for the completeness calculation, and they have an equal score.
An attribute and one or more attribute groups are selected	Only the completeness score from the selected attribute is used for the completeness calculation.

4. Select 'Save,' and the metric is ready to be used in a data profile.

On the **Completeness Metric** tab, in the **Configuration** area, you can view the configuration settings.

Configuration	
Completeness Metric Plugin Type: Standard Completeness Plugin	
> Completeness Score Attribute (Optional)	Completeness Score
> Attribute Group(s) Filter (Optional)	[Category Specific Attributes]

Edit

Note: If data containers should be considered for an entity's completeness score, the completeness metric must use default settings.

Migrating Completeness Scores to Description Attributes

Completeness scores are stored in a standard Description attribute. This means that all values can be stored as STEPXML and that completeness score setups can be moved easily between systems.

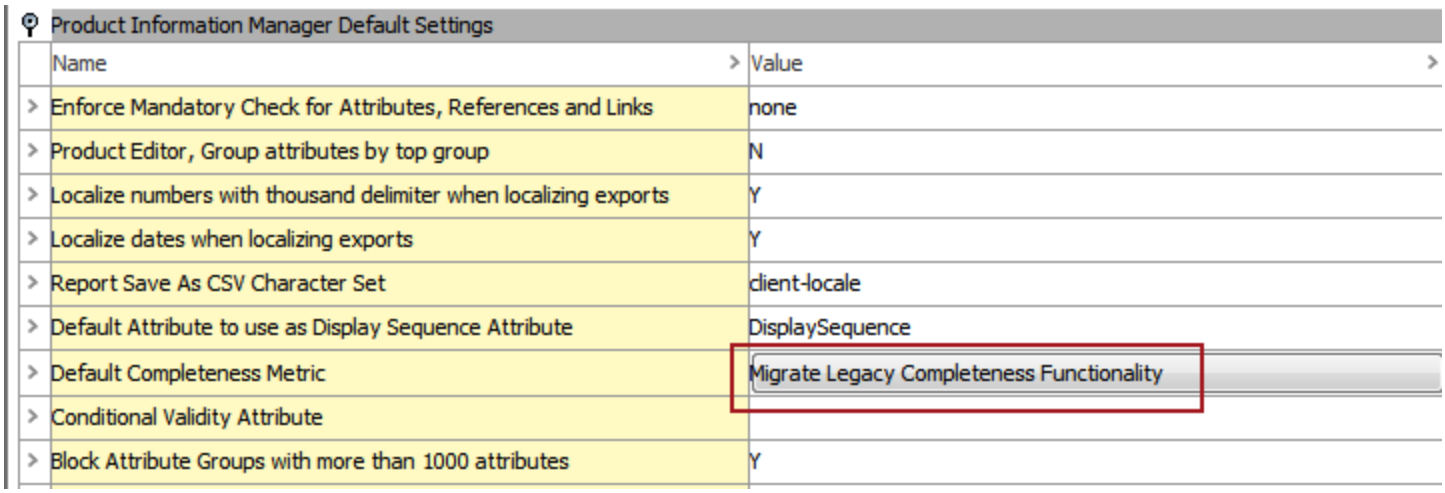
Before STEP 7.4, completeness score values were stored in a special system attribute. To migrate values from the system attribute to a Description attribute, use the **Migrate legacy completeness functionality** in **Users & Groups**.

Migrate Completeness Scores

The following describes the steps involved in migrating completeness scores to a Description attribute:

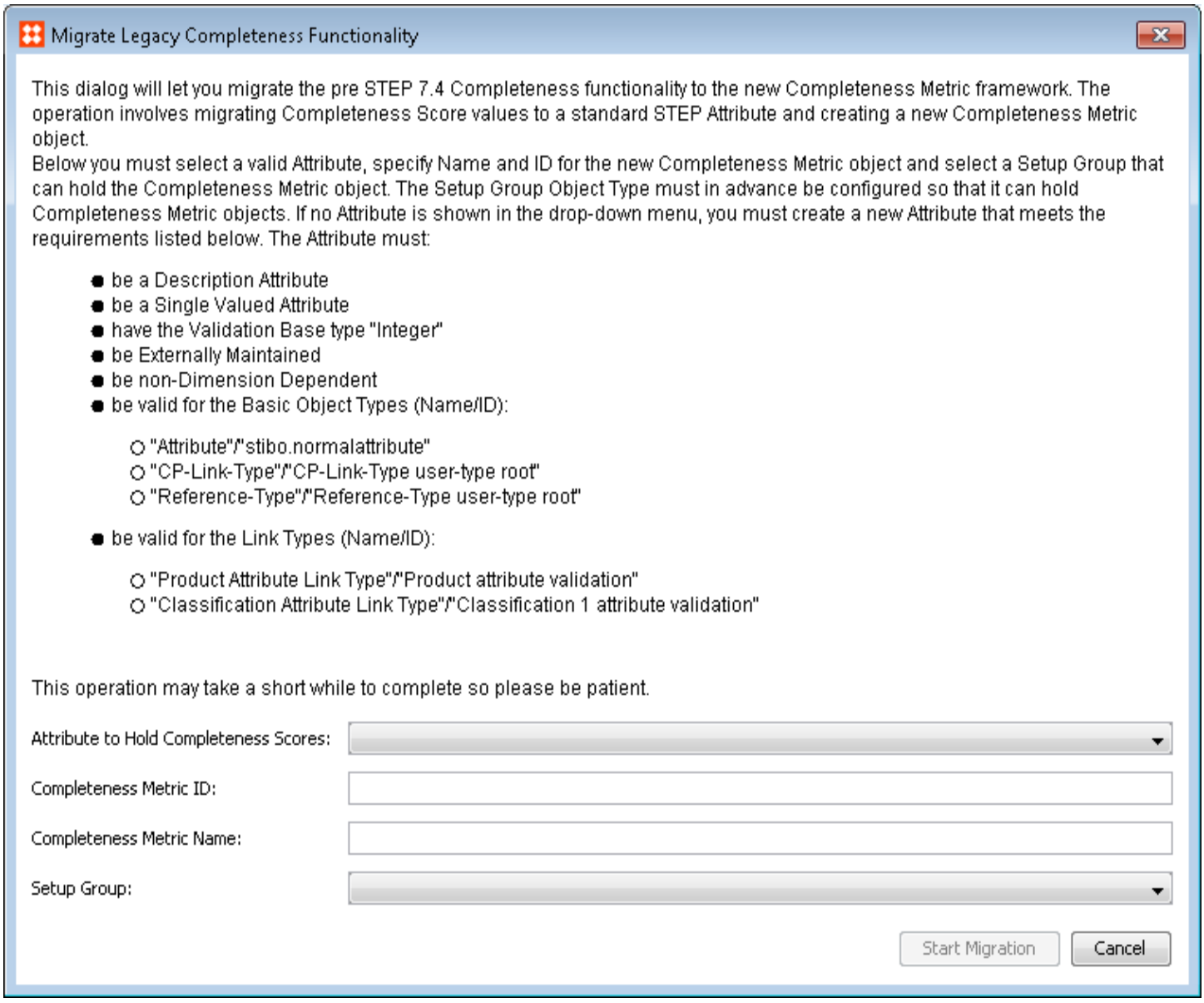
1. Create an attribute that can hold the completeness score values. The attribute must meet the following requirements:
    - It must be a Description attribute.
    - It must be a Single valued attribute.
    - It must have the Validation base type Integer.
    - It must be externally maintained.
    - It must not be dimension dependent.
    - It must be valid for the basic object type Attribute/stibo.normalattribute.
    - It must be valid for the basic object type CP-Link-Type/CP-Link-Type user-type root.
    - It must be valid for the basic object type Reference-Type/Reference-Type user-type root.
    - It must be valid for the link type Product Attribute Link Type/Product attribute validation.
    - It must be valid for the link type Classification Attribute Link Type/Classification 1 attribute validation.

    For information on how to set validity on an attribute, see the **Setting Validity of Description Attributes** section of the **System Setup / Super User Guide** documentation.
2. In **System Setup**, click **Users & Groups**.
3. In the **Product Information Manager Default Settings** area, click the **Migrate Legacy Completeness Functionality** button.



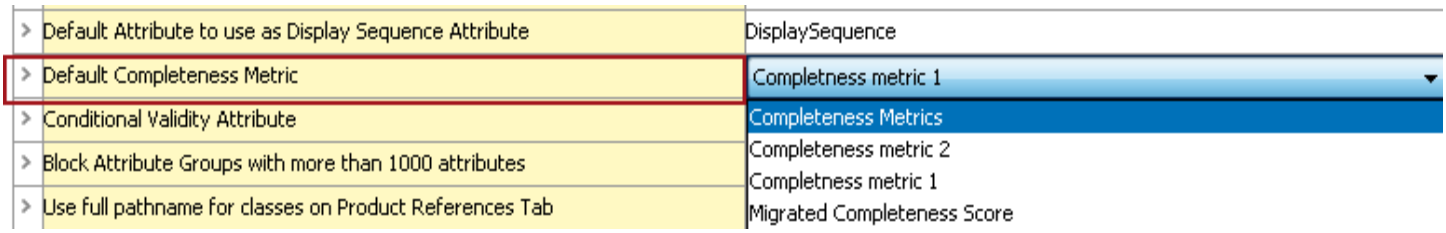
4. In the **Migrate Legacy Completeness Functionality** dialog, select the attribute created in Step 1 to hold the completeness scores. In order to progress with migration, an attribute must be created first.

   During the migration process, a Completeness Metric object is created, and an **ID** and a **Name** must be specified for the object. Select a **Setup Group** to hold the Completeness Metric.



The migration is not performed as a background process, so the dialog stays open until the migration has completed.

The migration is a one-time process and the **Migrate Legacy Completeness Score** button is not available after the migration has been completed. Instead, select the default completeness metric from a drop-down list.

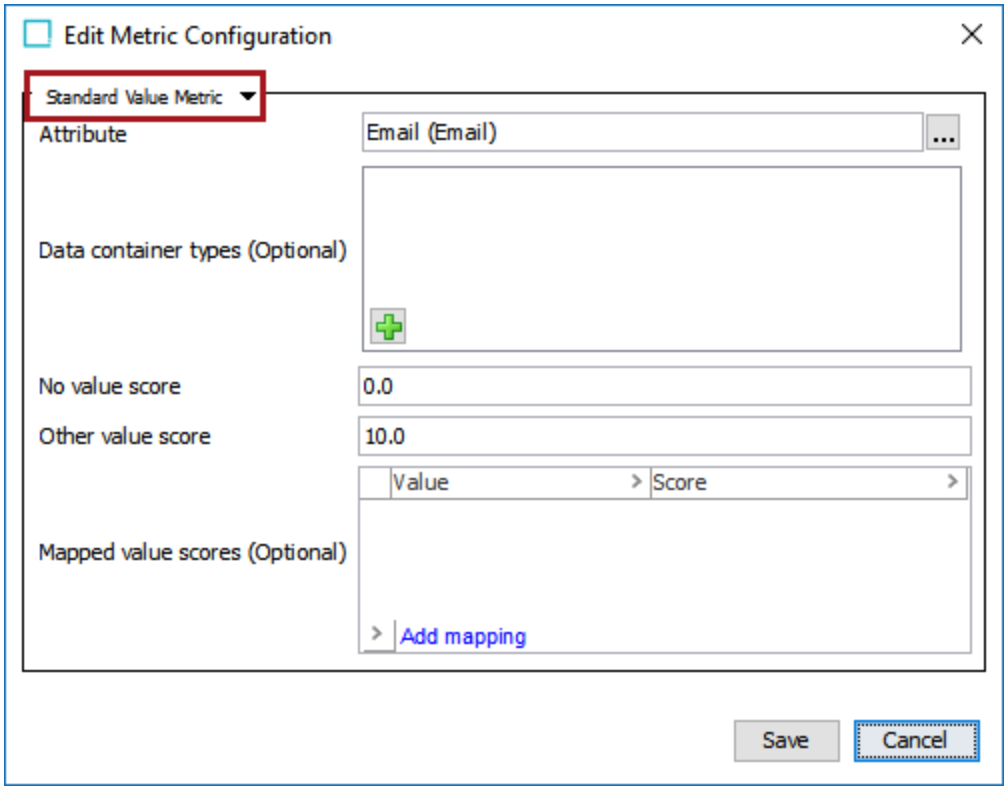


Value Metrics

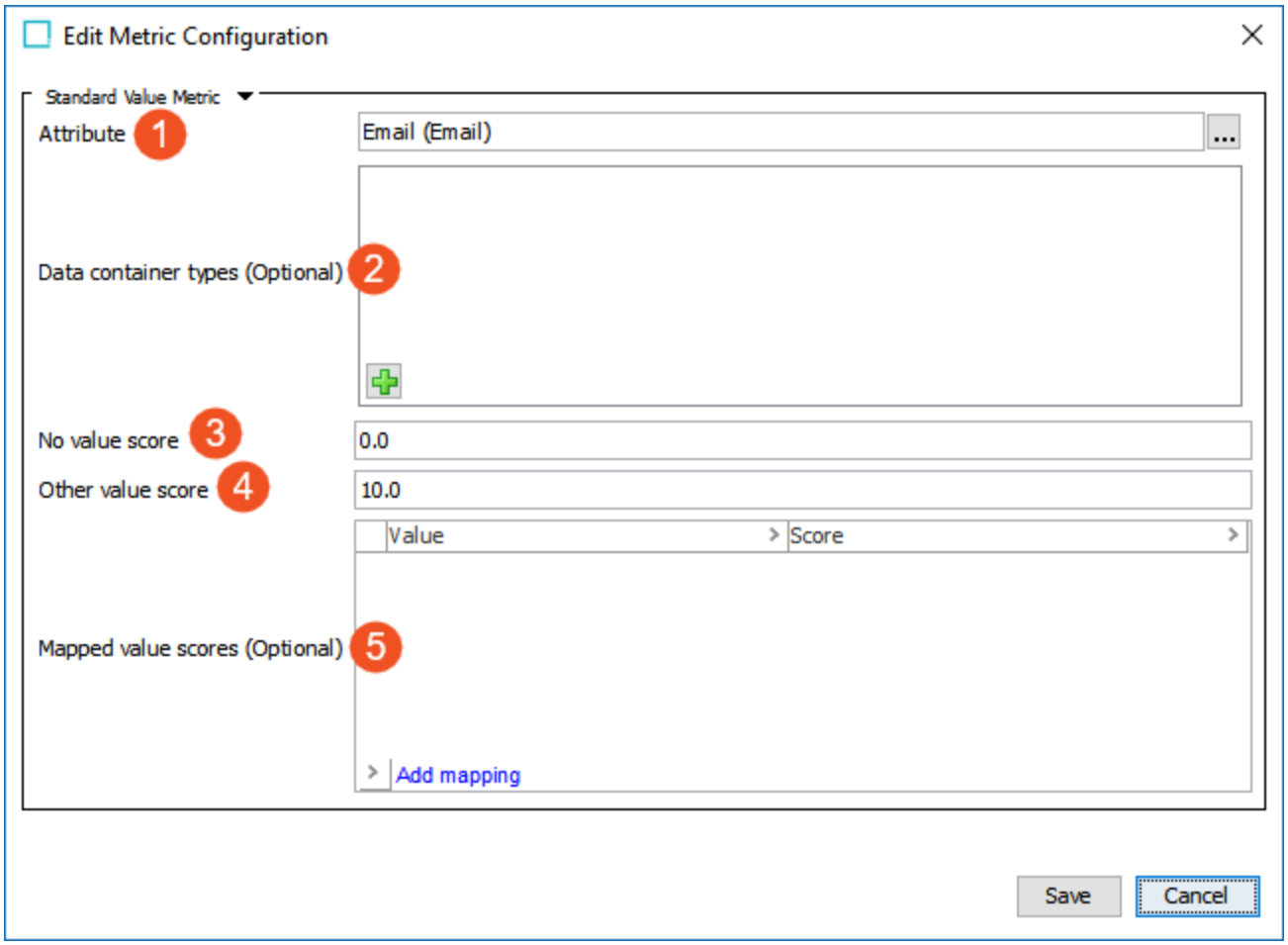
A value metric evaluates an attribute value and generates a metric score based on the result.

Configuration

To configure a value metric, open the editor for the desired metric object, and from the dropdown select **Standard Value Metric**.



A number of configuration options are available for value metrics.



1. **Attribute** - In this field, select the attribute to evaluate.

Note: The chosen attribute must be single value only, and must either be a simple numeric value, a text value, an LOV value, or a calculated value. Attempting to configure an attribute that does not comply with these requirements will result in an error.

2. **Data container types (Optional)** - If desired, a data container type can be included in the evaluation.
3. **No value score** - If there is no value for this attribute, then it returns the specified point value.
4. **Other value score** - If there are any values besides those listed in the **Mapped value scores** parameter, it will return this score.
5. **Mapped value scores (Optional)** - This field allows for mapping specific values to specific scores.

Select 'Save,' and the metric is ready to be used in a data profile or with data policies. To create a policy, start with the configuration of a dataset definition, as described in the **Creating a Dataset Definition** topic of this documentation.

On the **Completeness Metric** tab, in the **Configuration** area, you can view the configuration settings.

Configuration	
Metric Plugin Type: Standard Value Metric	
Data quality Score Attribute (Required)	Email
Data container types (Optional)	[]
No value score	0.0
Other value score	10.0
Value scores	[]

Edit

Business Function Metrics

Business Function metrics use JavaScript business functions to evaluate data and return metric scores, and depending on the JavaScript, messages that provide more detailed information about the returned scores.

For information on business functions, see the **Business Functions** topic of the **Business Rules** documentation.

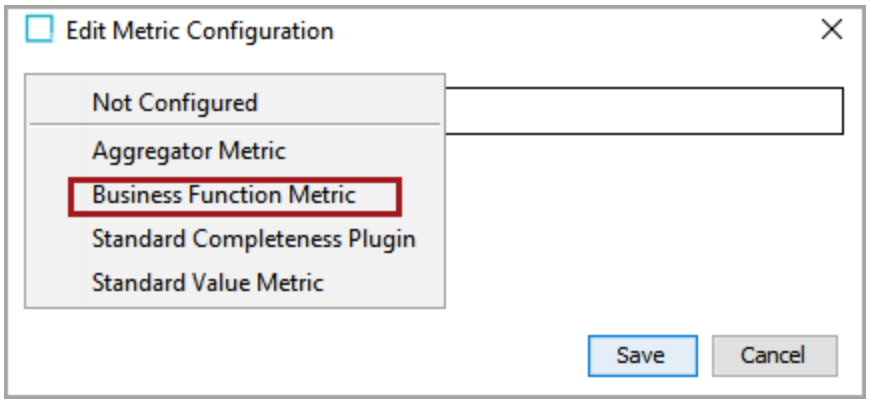
Business Function Metric Configuration

Before metrics can be created, a Setup Group must be configured to hold the metrics. For information on configuring a setup group for metrics, see the **Initial Setup for Metrics** topic in the **Metrics** documentation.

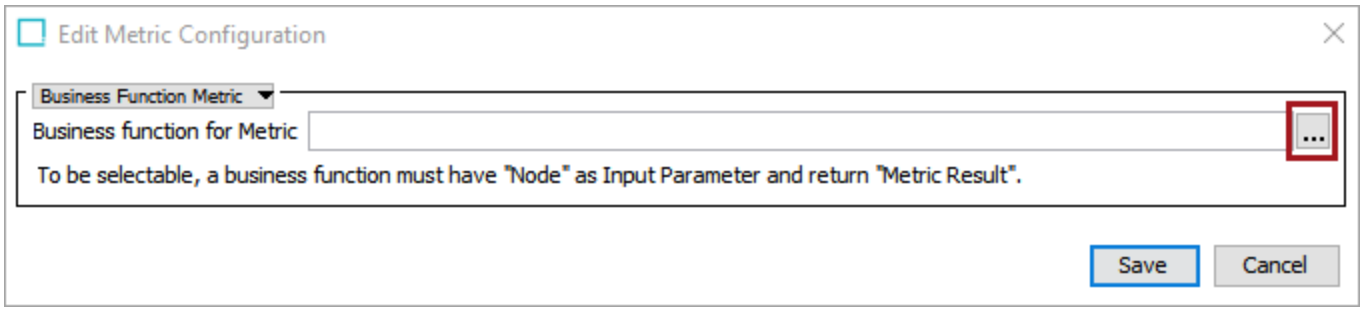
For information on creating a metric, see the **Creating and Editing Metrics** topic in the **Metrics** documentation.

To configure a business function metric:

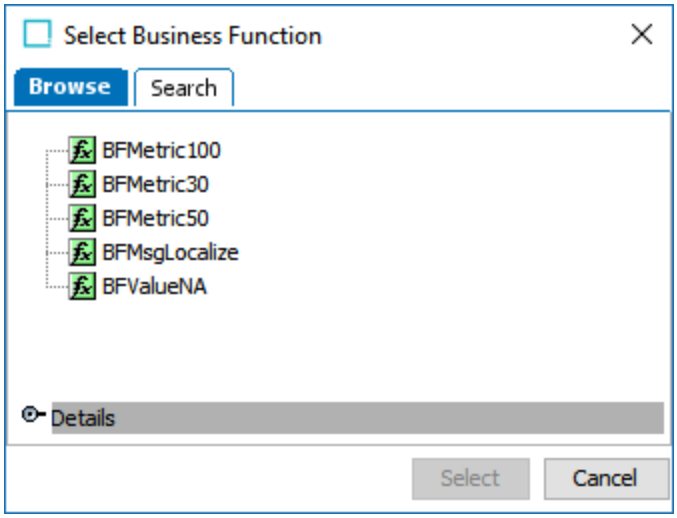
1. From the Edit Metric Configuration wizard, select 'Business Function Metric.'



2. Click the ellipsis button (...) by the **Business function for Metric** parameter.



3. Select the business function to include as part of the business function metric. To search for a specific business function, click the 'Search' tab and enter the search criteria.



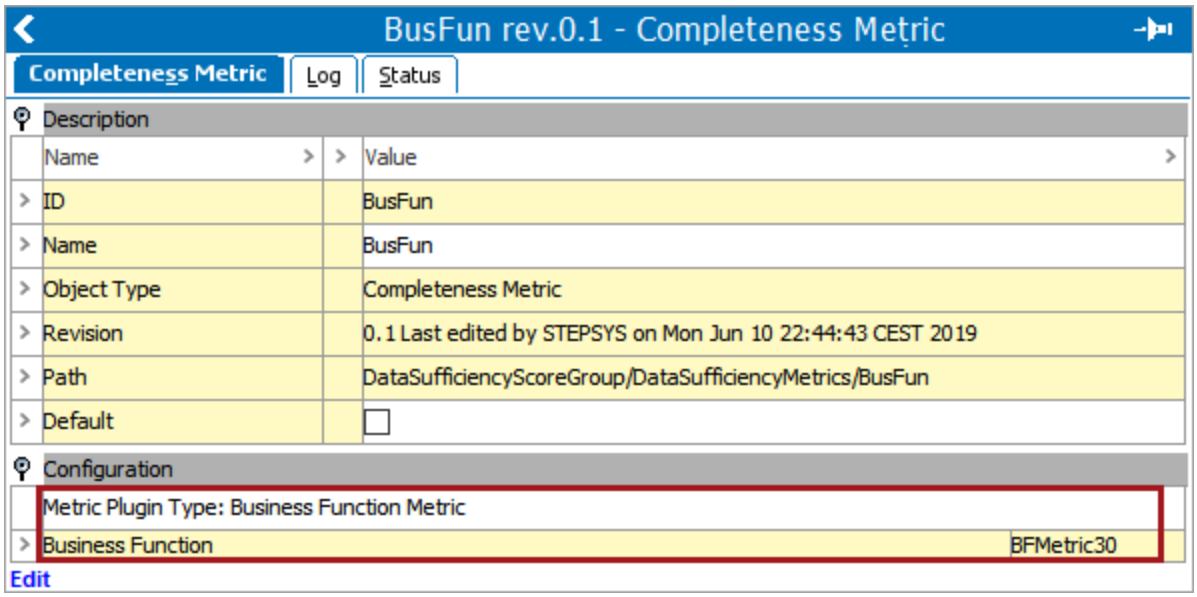
Note: For a business function to be selectable for a business function metric, it must have 'Node' as the input and 'MetricBusinessFunctionResult' as the return type.

4. Click 'Save' to close the 'Edit Metric Configuration' wizard.

The business function metric is configured and ready for use.

Important: The MetricBusinessFunctionResult of a JavaScript business function must be instantiated directly in the business function. It cannot be instantiated by a JavaScript library.

With the configured metric selected, the name of the metric plugin type (in this case, 'Business Function Metric') and the name of the selected business function will be listed under the Configuration flipper.


BusFun rev.0.1 - Completeness Metric
Completeness Metric
Log
Status
Description
Name
Value
ID
BusFun
Name
BusFun
Object Type
Completeness Metric
Revision
0.1 Last edited by STEPSYS on Mon Jun 10 22:44:43 CEST 2019
Path
DataSufficiencyScoreGroup/DataSufficiencyMetrics/BusFun
Default
Configuration
Metric Plugin Type: Business Function Metric
Business Function
BFMetric30
Edit

Aggregator Metrics

An aggregator metric allows users to combine scores from multiple metrics. By default, the aggregator metric returns an average score based on these combined scores; however, by adding a business function to an aggregator metric, users are able to obtain scores based on other variables, e.g., if one metric is weighted heavier than another metric within the aggregator metric, the score would reflect the difference based on the weight of each metric.

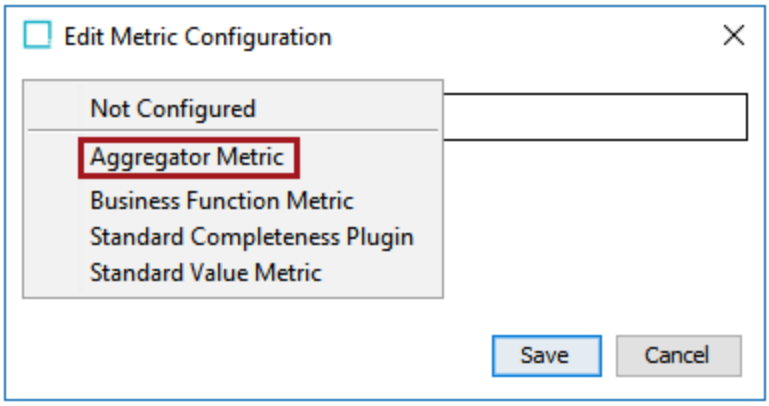
Aggregator Metric Configuration

Before metrics can be created, a Setup Group must be configured to hold them. For information on configuring a setup group for metrics, see the **Initial Setup for Metrics** topic in the **Metrics** documentation.

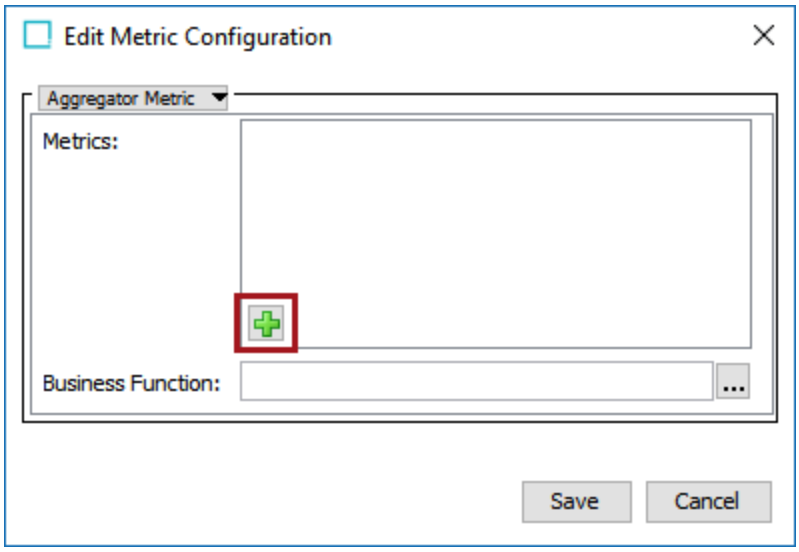
For information on creating a metric, see the **Creating and Editing Metrics** topic in the **Metrics** documentation.

To configure an aggregator metric:

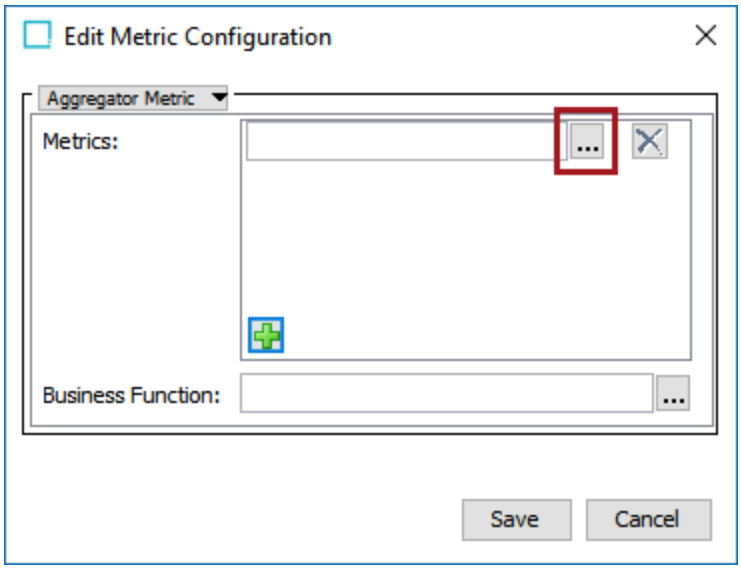
1. From the Edit Metric Configuration wizard, select 'Aggregator Metric.'



2. Click the green plus sign.

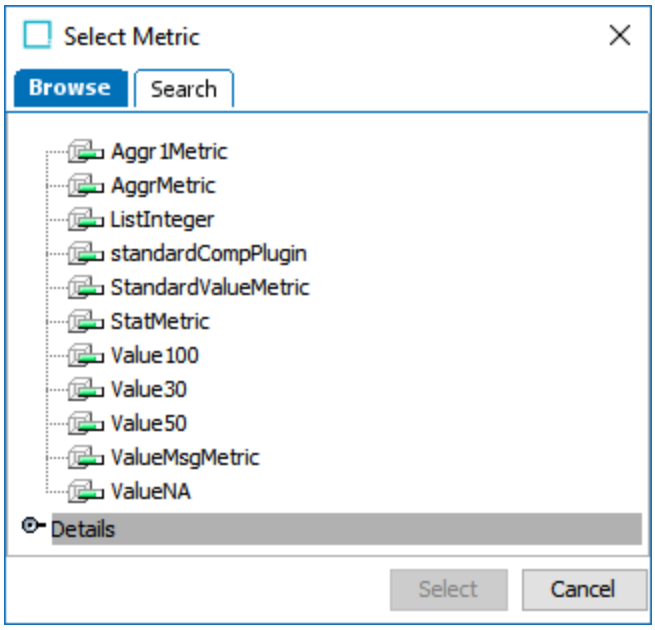


3. Click the ellipsis button.

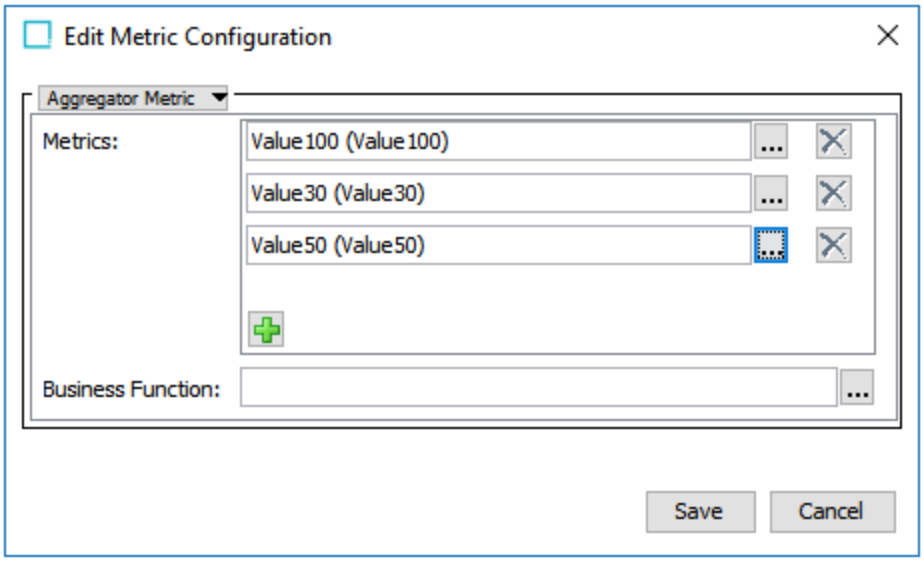


4. Select the metric you want to include in the aggregator metric and click 'Select.'

Note: Only one metric can be selected at a time. After a metric is selected, the green plus button must be clicked on again to add another metric.



In the image below, three metrics have been selected from the available metric list.

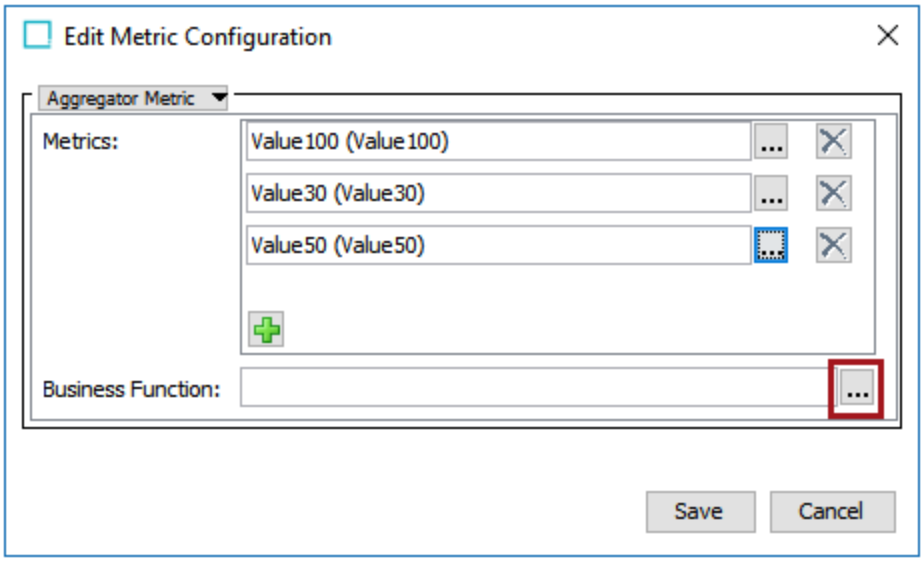


Adding a business function to an aggregator metric

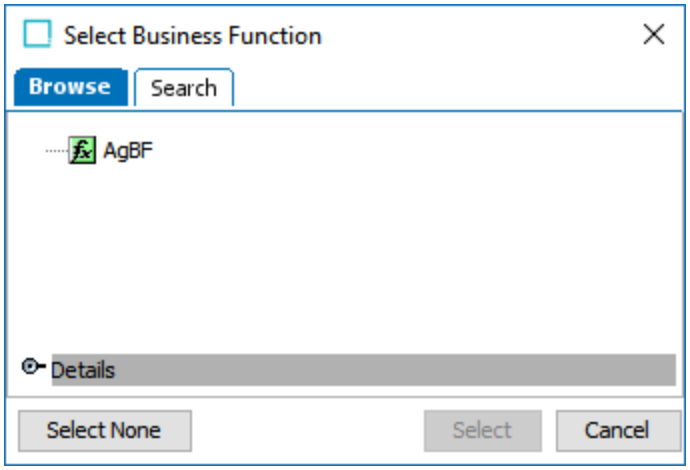
Although optional, the purpose of adding a business function to an aggregator metric is to return a score that is not the average score of the configured metrics. If no business function is selected, the returned score will be the average score of the configured metrics included in the aggregator metric.

To add a business function to an aggregator metric:

1. Click on the business function ellipsis.



2. Select a business function and click the 'Select' button.

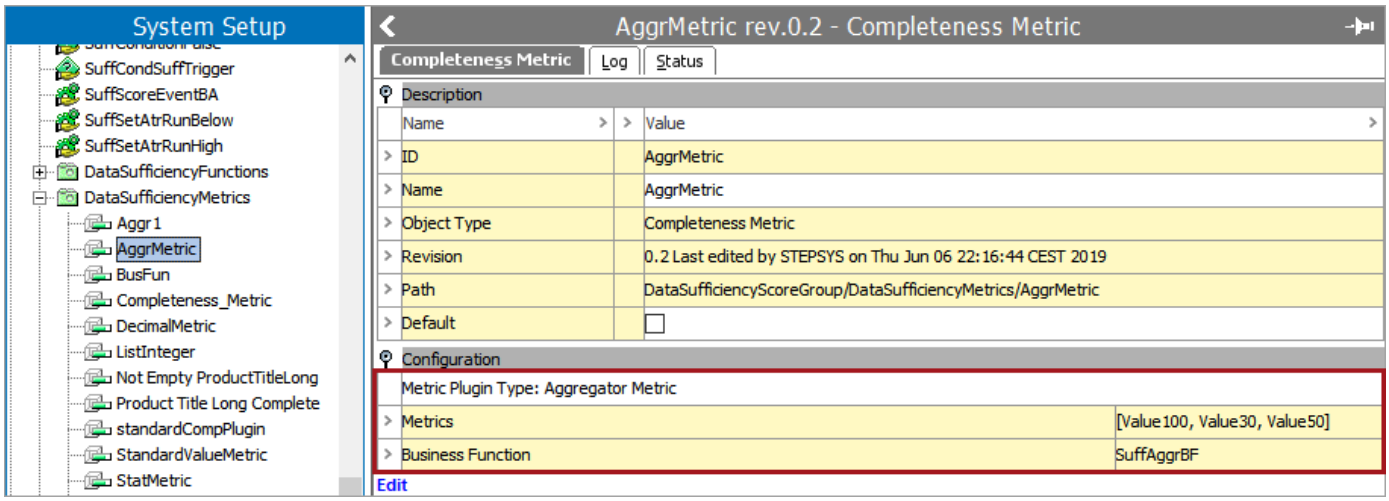


3. Click 'Save' to close the 'Edit Metric Configuration' wizard.

The aggregator metric is configured and ready for use.

Note: To use a business function with an aggregator metric, the input parameter type must be set to 'List<MetricResult>' and the return type must be an integer. Business functions without these configuration settings will not be available as a valid option within the Select Business Function window. For more information on business functions, see the **Business Functions** topic in the **Functions** documentation.

With the configured metric selected, the name of the metric plugin type (in this case, 'Aggregator Metric'), the metrics included in the aggregator metric, and the name of the selected business function will be listed under the Configuration flipper.


System Setup
SuffCondSuffTrigger
SuffScoreEventBA
SuffSetAtrRunBelow
SuffSetAtrRunHigh
DataSufficiencyFunctions
DataSufficiencyMetrics
Aggr1
AggrMetric
BusFun
Completeness_Metric
DecimalMetric
ListInteger
Not Empty ProductTitleLong
Product Title Long Complete
standardCompPlugin
StandardValueMetric
StatMetric
AggrMetric rev.0.2 - Completeness Metric
Completeness Metric
Log
Status
Description
Name
Value
ID
AggrMetric
Name
AggrMetric
Object Type
Completeness Metric
Revision
0.2 Last edited by STEPSYS on Thu Jun 06 22:16:44 CEST 2019
Path
DataSufficiencyScoreGroup/DataSufficiencyMetrics/AggrMetric
Default
Configuration
Metric Plugin Type: Aggregator Metric
Metrics
[Value100, Value30, Value50]
Business Function
SuffAggrBF
Edit

Metric Visualization

Metric score indicators provide information about the data quality of objects. For completeness metrics, this overall score is calculated based on completeness scores that you set for individual attributes, reference types, data containers, and product to classification link types. While it is possible to calculate a quality percentage without settings scores for these individual data objects, it is recommended to do so, as you can weigh them based on relevance or outright omit those that should not apply to the overall quality percentage.

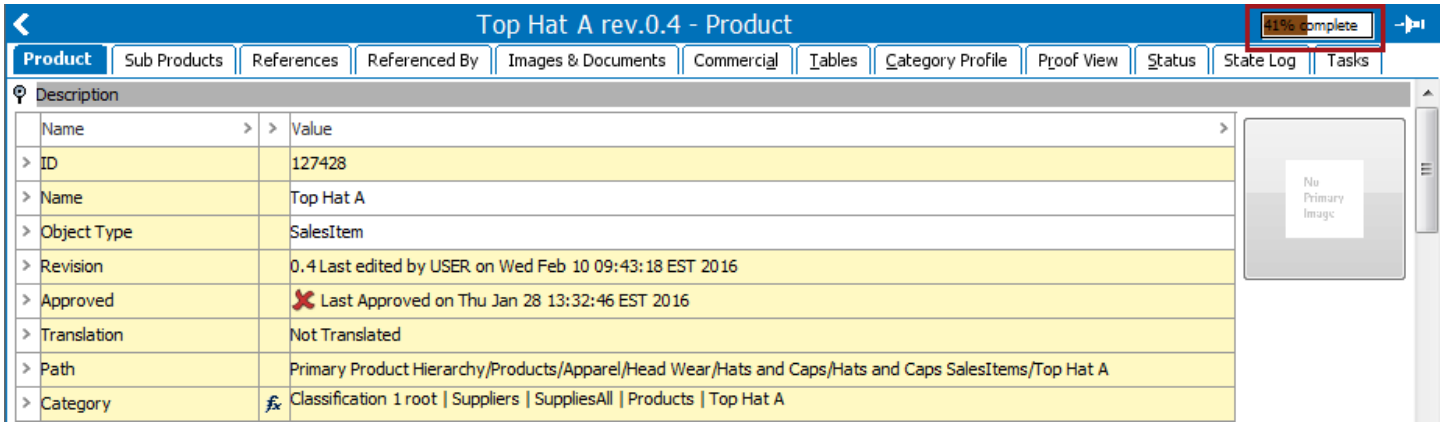
For value and function metrics, this score is a simple reflection of the resulting metric score.

Note: This functionality is not available on product overrides.

View Quality Scores

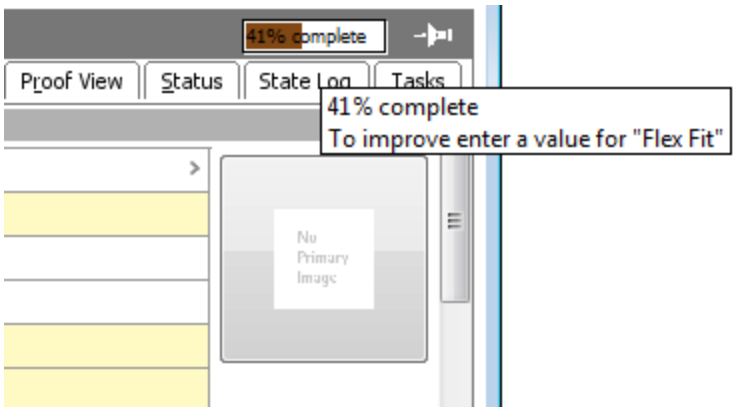
1. In the **Tree** tab, expand the applicable Hierarchy, and then locate the relevant object.
2. Click the main object tab or the **References** tab. The **Completeness Meter** in the upper right corner displays the current status of the object expressed as a percentage.

Note: If using a completeness metric and no indicator appears, it is likely that no completeness scores have been set for the data valid for the object.



Important: Steps 3 and 4 are only available for completeness metrics.

3. Hover the mouse over the **Completeness Meter** to view a text describing what you can do to improve the quality percentage.



4. To get a more detailed view of how the quality percentage for the selected object is calculated, double-click the **Completeness Meter**. A **Completeness Info** window appears, which displays all attributes, data containers, and referenced objects that contribute to the quality percentage and how they are weighted.

Completeness Info

Completeness Info

Completeness Metric: Completeness Metric B (Completeness Metric B)
Object 65.08% Complete

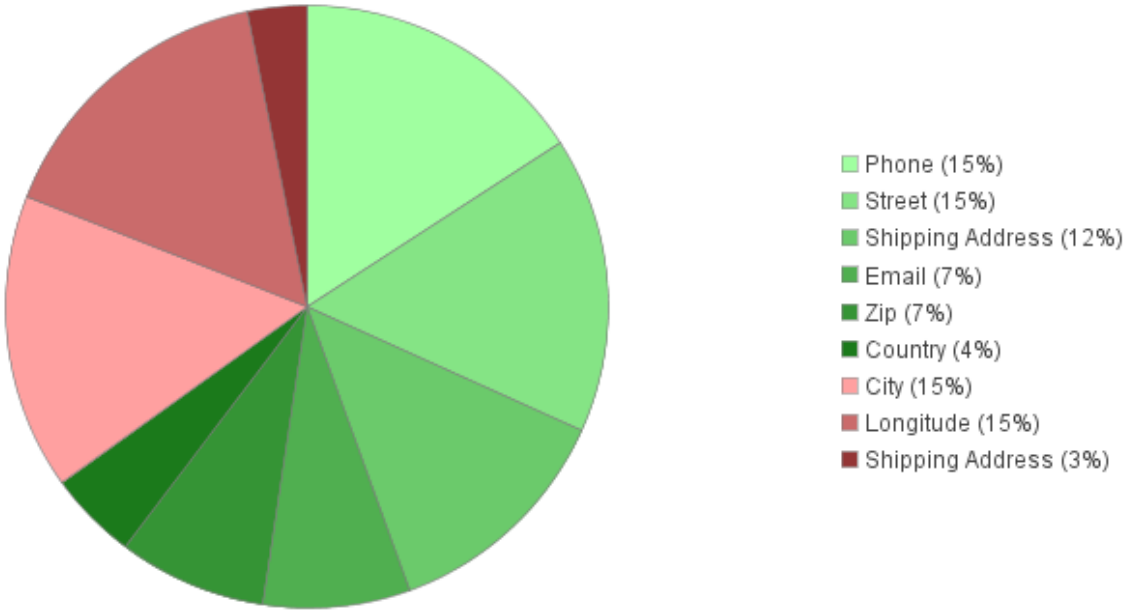
Weights - Unpopulated Data - 34.92% collapse

- 15.87% City
- 15.87% Longitude
- 3.17% Shipping Address (of possible 15.87%)

Weights - Populated Data - 65.08% collapse

- 15.87% Phone
- 15.87% Street
- 12.7% Shipping Address (of possible 15.87%)
- 7.94% Email
- 7.94% Zip
- 4.76% Country

Scored Attributes / References / Data Containers



OK

Metric score indicators are also available in Web UI. See the **Metric Visualization in Web UI** section of the **System Setup** documentation.

User Privileges

The quality of an object is always calculated and displayed as if the user had privileges to view and edit all relevant data. However, hints are only provided for data that the user is allowed to edit.

Additionally, this functionality is only accessible to users with the appropriate privileges. For more information, see the **Setup Actions** section of the **System Setup / Super User Guide** documentation.

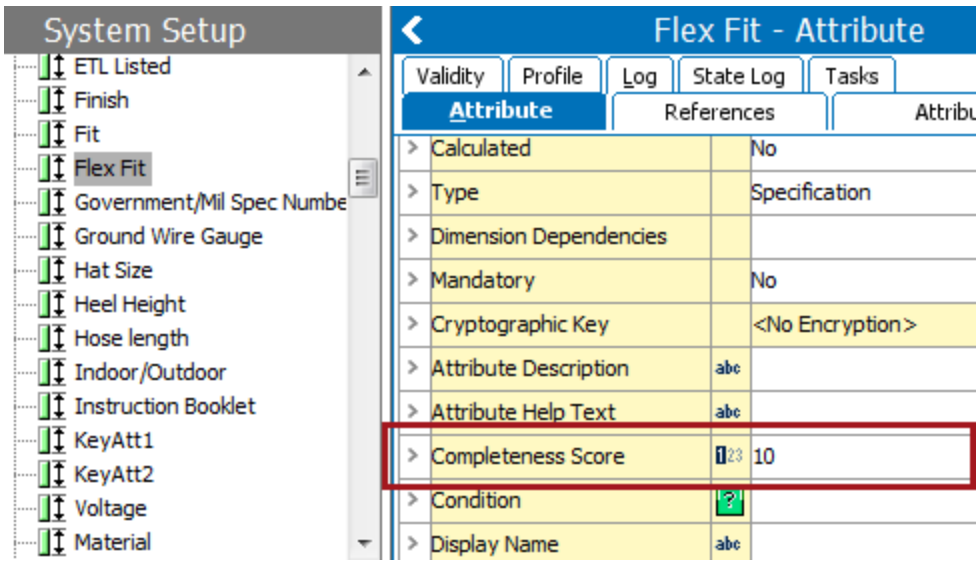
Define Completeness Scores

For a typical completeness metric setup, in order to view object quality information, completeness scores must be configured on all relevant attributes, references, data containers, and product to classification link types. This score determines how much to weigh an element of the object when factoring the object's quality percentage.

Note: For product reference types, images and document reference types, and product to classification link types, the scores are defined on the individual type's main editor.

Important: The below features and configuration steps are only relevant to completeness metrics.

1. In **System Setup** expand **Reference Types** or **Attribute Groups**, and then locate the relevant reference type / attribute.
2. In the **Completeness Score** field, under the **Description** section, enter the preferred score. Scores must be integers.



Note: You cannot specify scores for calculated attributes.

Alternatively, when setting the score on the link between a product or classification and a specification attribute, the completeness score of an attribute can be specified on the actual object in the **Tree**.

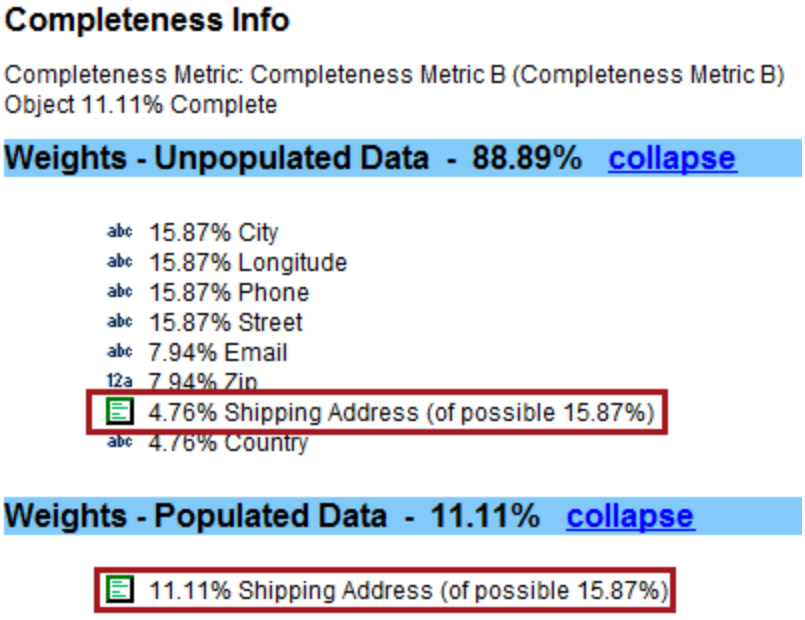
1. In the **Tree**, expand the applicable hierarchy, and then locate the desired object.
2. On the **References** tab, go to **Linked Attributes** and then click the flipper icon to expand the table.
3. In the **Completeness Score** column, enter the preferred score(s) for the relevant attribute links.

Linked Attributes from Product Hierarchy

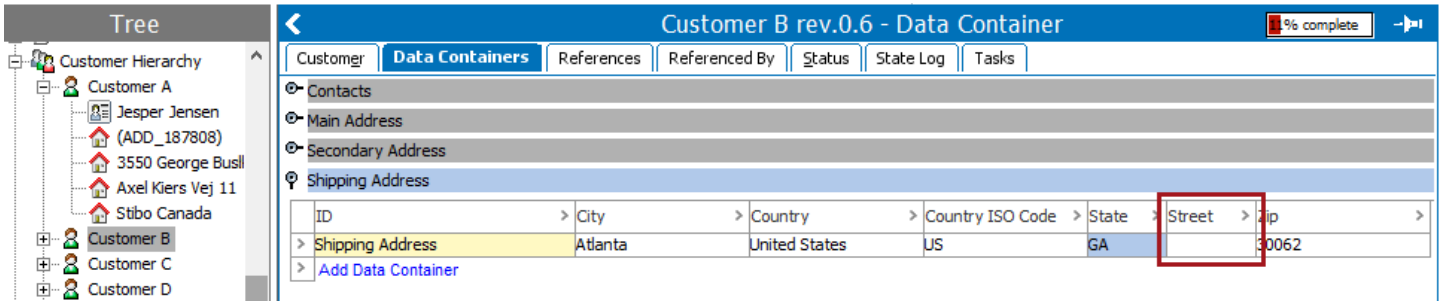
DisplaySequence	ID	Name	Completeness Score
	DescriptionWeb	Description, Web	
2	DescSource	Description Source	
2	DescTarg	Description Target	
	Disc Dimensions	Disc Dimensions	10

Setting Completeness Scores for Data Containers

Unlike other data objects that contribute to the object's quality percentage, a data container can be incomplete itself, making it possible to contribute only a fraction of its maximum score. The weight of the total possible score of the data container is set on the data container type (just like for attributes and reference types). Any values absent from the individual attributes that comprise the data container are reflected in its score. Additionally, the attributes within a data container can have their own varying scores. In a sense, a data container has its own quality percentage, and contributes either 100% of its score to the object's quality percentage, or, if one or more attributes within are missing values, it contributes a fraction of its total possible score.



In the above example, the data container 'Shipping Address' can contribute a total of 15.87% completeness to the object's overall quality percentage. However, because it has attributes with missing values, it is only contributing 11.11% completeness; roughly 70% of its maximum score.

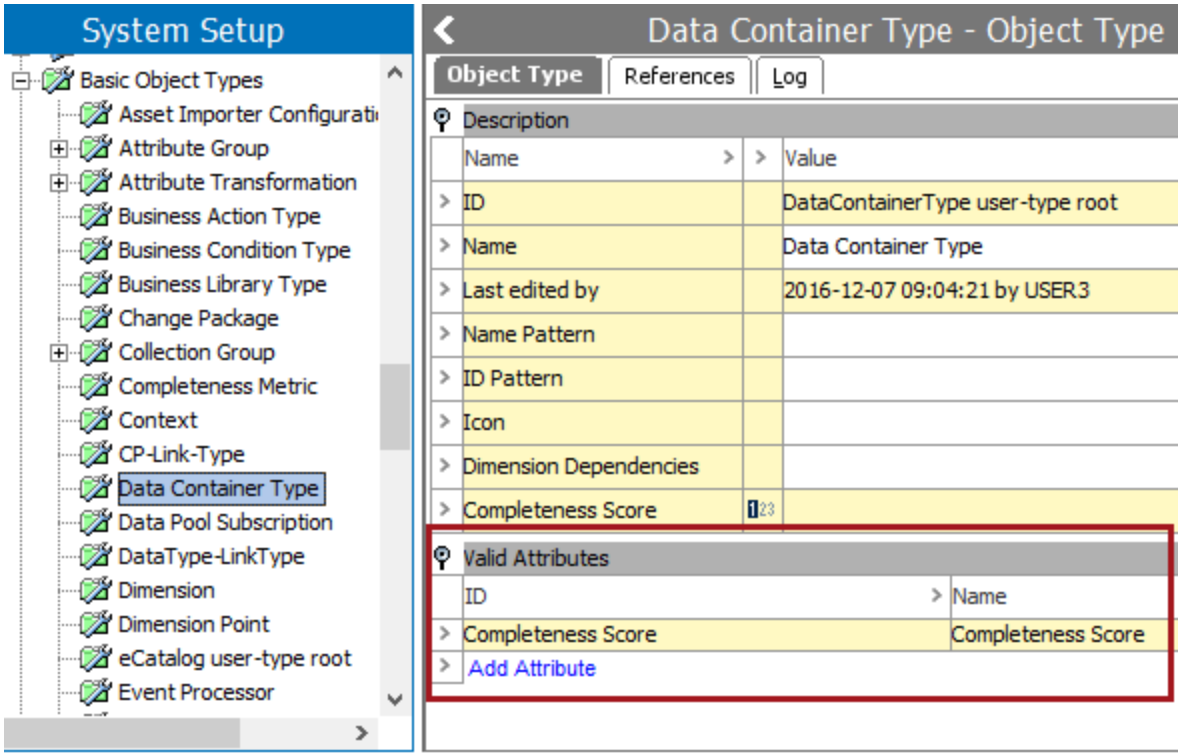


By viewing the data container itself (pictured above), you can see that the attribute 'Street' is missing a value, which explains why the data container is not contributing 100% of its score. Had all six attributes been weighted with a neutral score, each would contribute roughly 17% to the data container's score. However, based on the completeness breakdown, we know that the attribute 'State' is responsible for roughly 30% of the score. This is because 'State' has a higher weighted score than some of the other attributes.

Note: If a second 'Shipping Address' instance were added, its attributes would be factored in with those of the existing 'Shipping Address' when calculating the score, as they are both held within the same data container.

As with all attributes, the scores for attributes that comprise a data container are defined on the Completeness Score metadata field found on the attributes. The maximum score for the data container itself is defined by the same field on the data container type (in this example, 'Shipping Address').

Before completeness scores can be applied to data containers, however, ensure that the Completeness Score attribute is valid for the Data Container Type object type.



Export Completeness Values

Completeness values can be exported to Excel and CSV.

Important: The below features and configuration steps are only relevant to completeness metrics.

1. Create a calculated attribute and enter the function "completeness()" in the Value template.

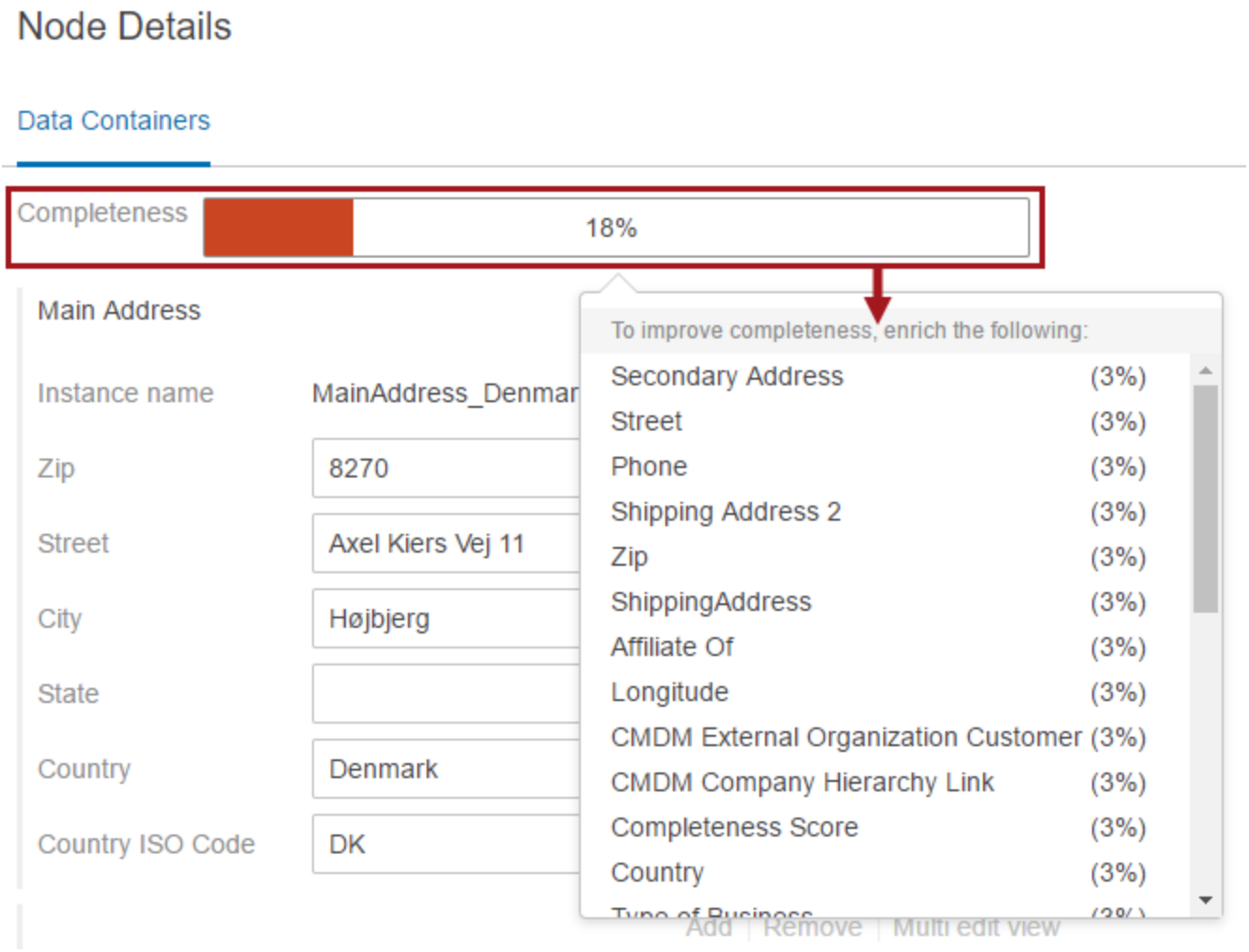
   For more information, see the **Calculated Attributes** section of the **System Setup / Super User Guide** documentation.

2. From the **File** menu, hover over **Export** and then click **Data...**.
3. In **Step 4 - Map Data** of the Export Manager wizard, add **Select Attribute** to columns. The **Select Attribute or Attribute Group** dialog is displayed. Search or browse for the desired calculated attribute.
4. Select the calculated attribute, check **Force calculation**, and then continue with the export.

   For more information, see the **Creating a Data Export** section of the **Export Manager** documentation.

Metric Visualization in Web UI

Metric score indicators are available on Node Details component pages, allowing users to view the quality of the data currently selected node in Web UI. Like in workbench, object quality is represented by a completeness meter. Additionally, mousing over the meter will provide text describing what you can do to improve the quality percentage.

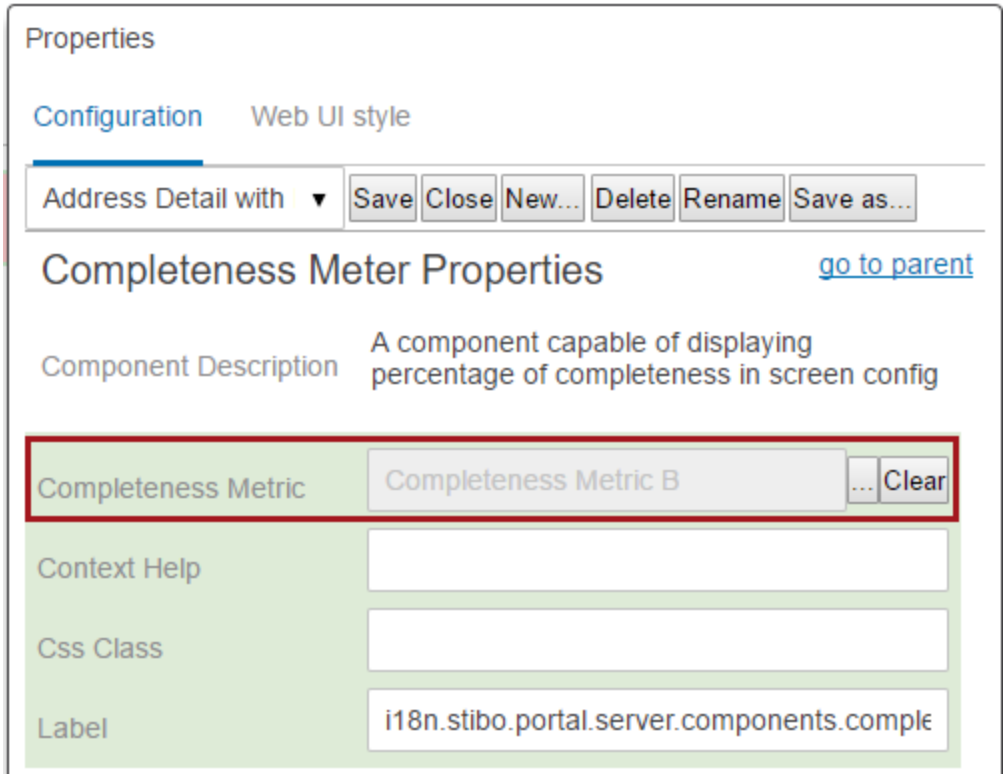


Adding the Completeness Meter Component

1. In design mode, navigate to Node Editor properties.
2. Under Child Components, click **Add**.
3. In the 'Add component' window that appears, select 'Completeness Meter,' and click **Add**.
4. Click **Save**.

Configuring the Completeness Meter Component

1. Navigate to the Completeness Meter properties.
2. In the Completeness Metric field, click the ellipsis button (...), and select the desired completeness metric from the node selector. This selection will determine which completeness metric to use when evaluating the data on the object.

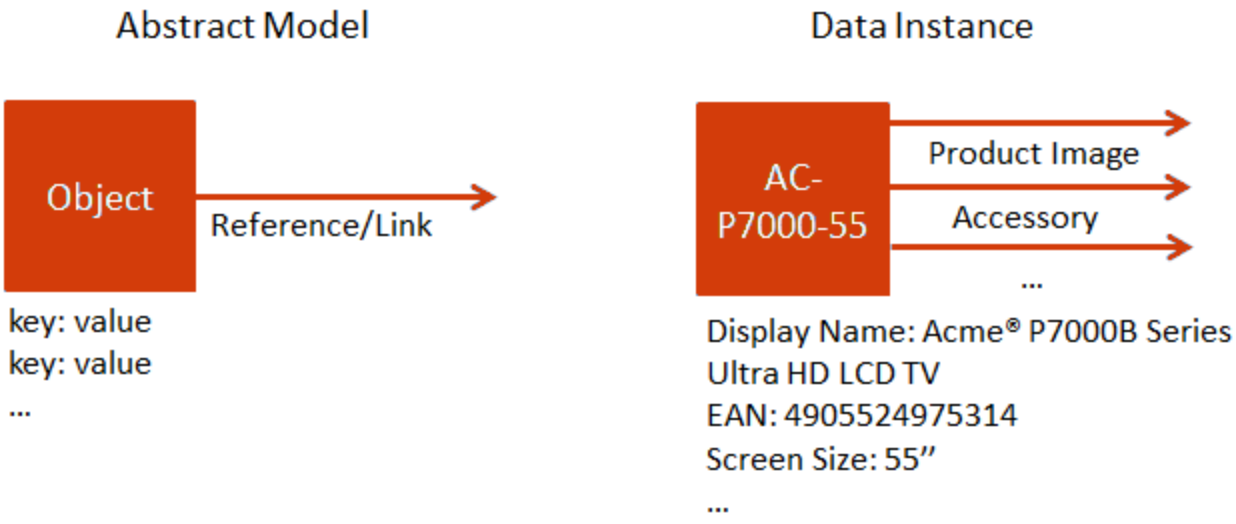


3. Click **Save**.

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.

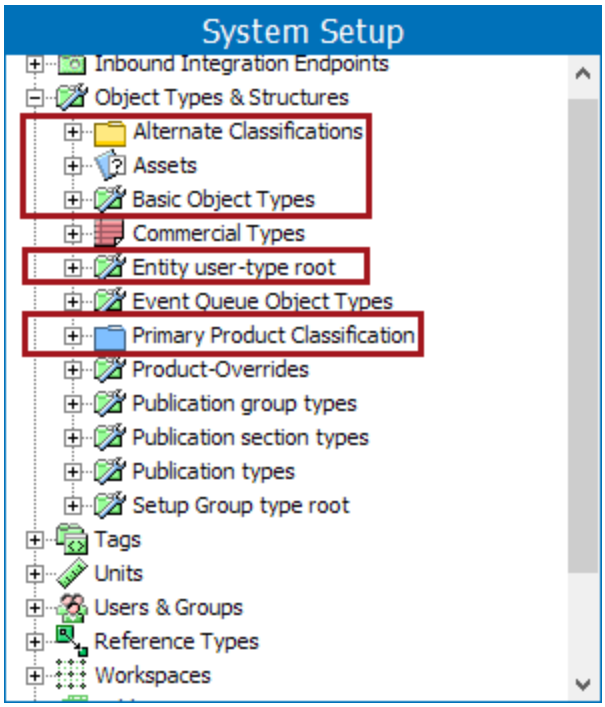
Object Type Facts in STEP

- 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
Classification	Super type for modeling product and asset classification hierarchies	• Web Category • Asset Folder	Televisions
Publications	Super type for modeling hierarchies and objects used for print publishing solutions.	• Flier • Magazine • Pamphlet	Winter Sale Catalog

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 *system data*. There are many Basic Object Types including: • Attribute Group • Context • Dimension • User When an object of any basic object type is created, the relevant object type is automatically applied. New basic object types cannot be created. For the complete list of basic object types and more information, see the **Basic Object Types** section.
Non-Basic Object Type / Tree Object Type	A non-basic object type applies to nodes that are used for structuring the product data hierarchies including: • Alternate Classifications

Object Type kind	Description
	• 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.

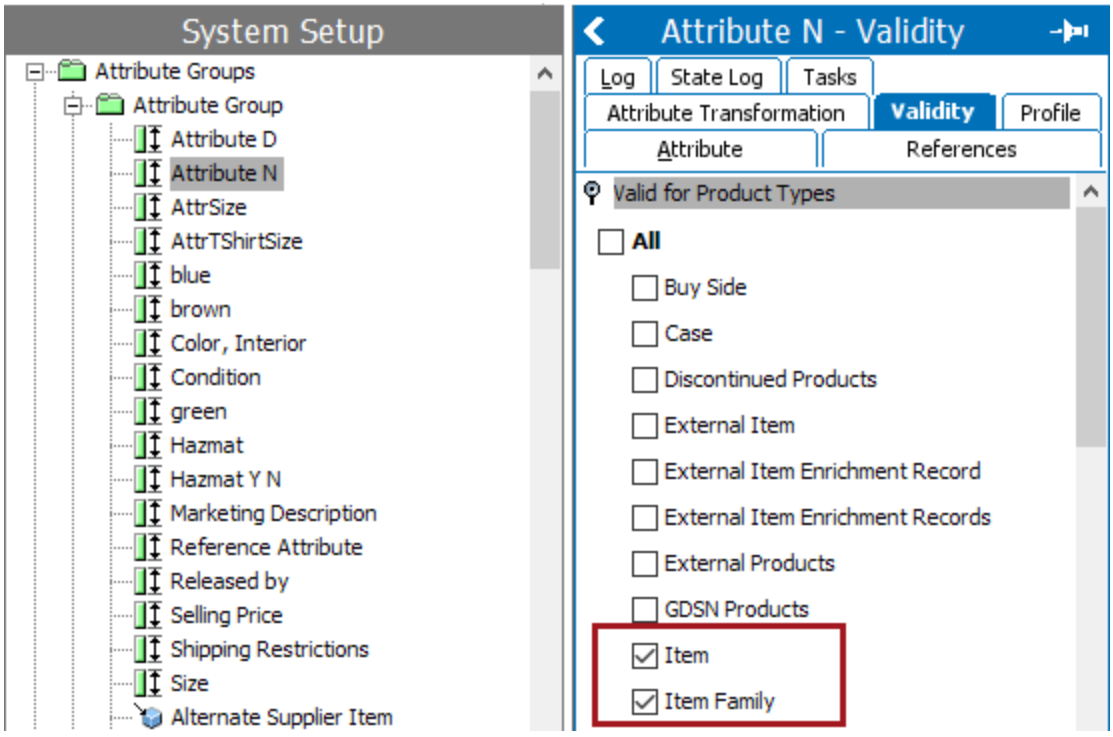
Attributes Valid for Object Types

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.

Autogenerate 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.

IDs alone, however, are not unique identifiers in STEP because 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, an asset, an attribute, and an LOV 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. It is also advisable to avoid any spaces in IDs.

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.

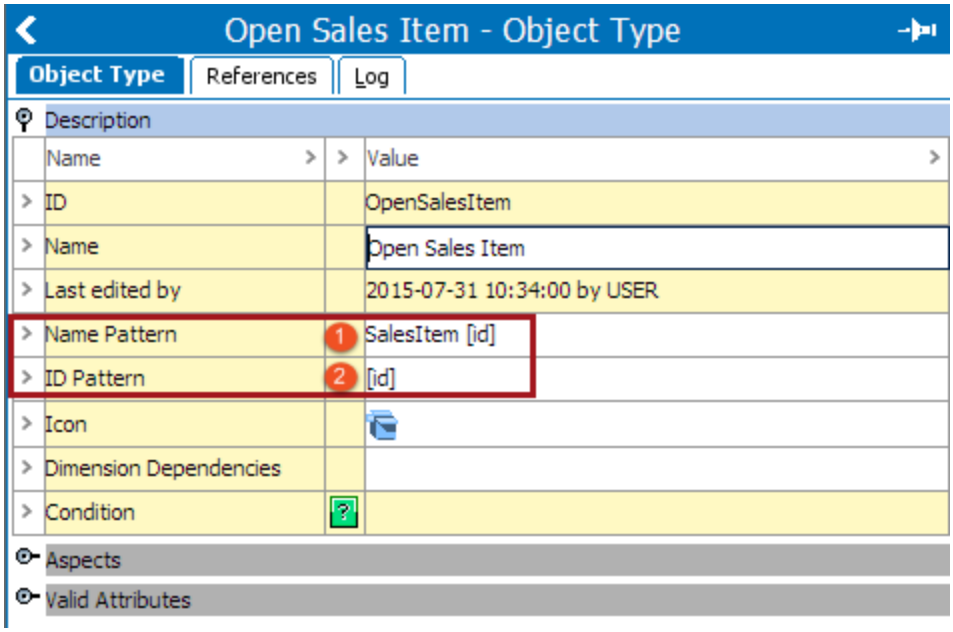
For example, an auto ID Pattern is generally used when the products are created automatically, either by import or from an upstream system. This could be the case if there is a legacy system where the products were originated and subsequently introduced to STEP. To create the products in STEP, an auto ID Pattern may be used.

Defining the Name and ID Pattern

The Name and ID of a node can be autogenerated by entering a template pattern. An Object Type with a Name Pattern and/or ID Pattern will cause the system to present to the user the resulting pattern when a new node of that Object Type is created manually.

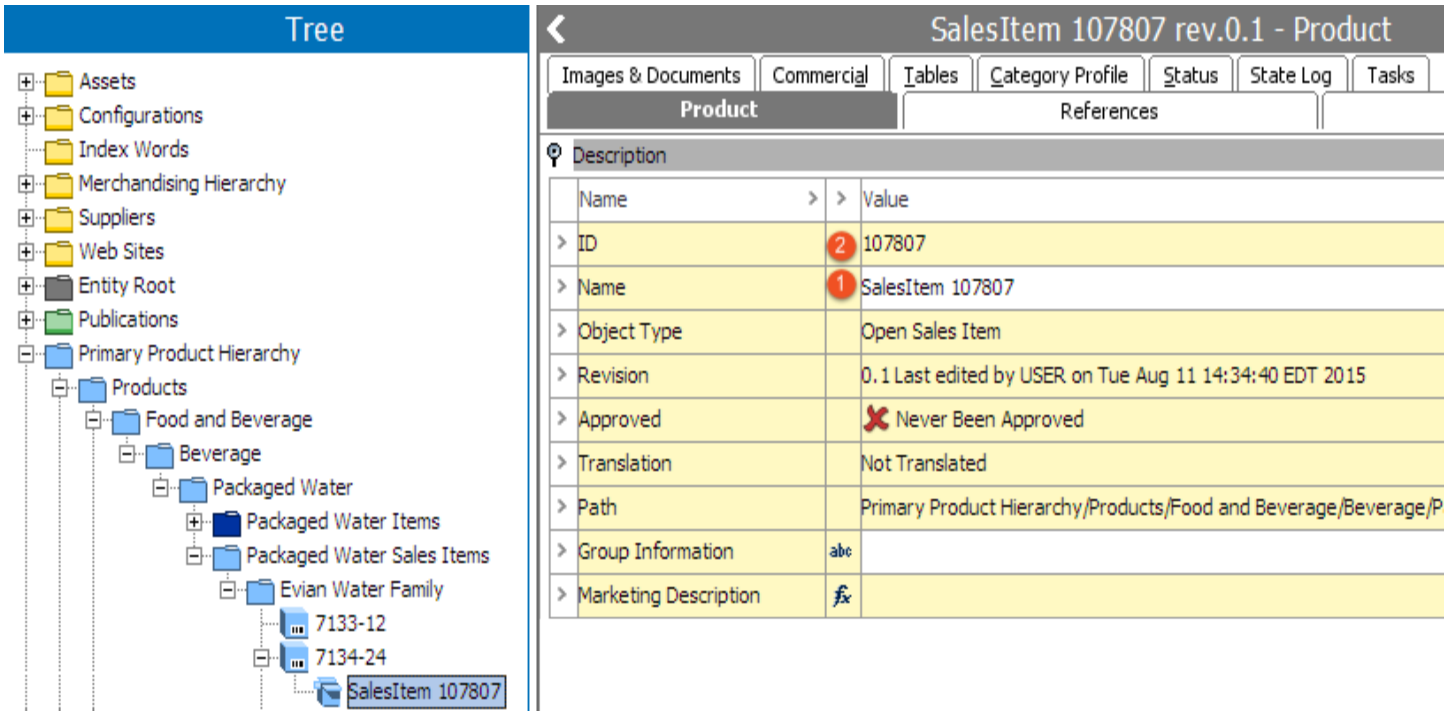
- Static text entered in the Object Type's Name Pattern or ID Pattern will be displayed in the Name or ID field when a new node is created.
- A unique incremental counter automatically displays the next available number in a sequence when the tag '[id]' is used. This counter is extremely advantageous when used in an ID Pattern, because it guarantees a unique ID every time a new node of that Object Type is created.
- Name Patterns and ID patterns typically consist of both an incremental counter and static text. Static text may be entered before or after the [id], or both. Examples: *ATTR_[id]*, *[id]_FAM*, and *PROD_[id]_LINE*.
- Similar to the [id] tag functionality, IDs can be autogenerated using a universally unique identifier (UUID) generator by using the tag '[uuid]'. Users should be aware that a UUID takes up 36 characters out of the maximum 40 characters that a STEP ID can contain, so it either needs to be used alone or with a four character prefix and/or suffix.
- ID Patterns are far more commonly used than Name Patterns.

To specify an ID / Name Pattern, select the Object Type in System Setup / Object Types and Structures, and enter the desired 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] or [uuid] for the ID field.

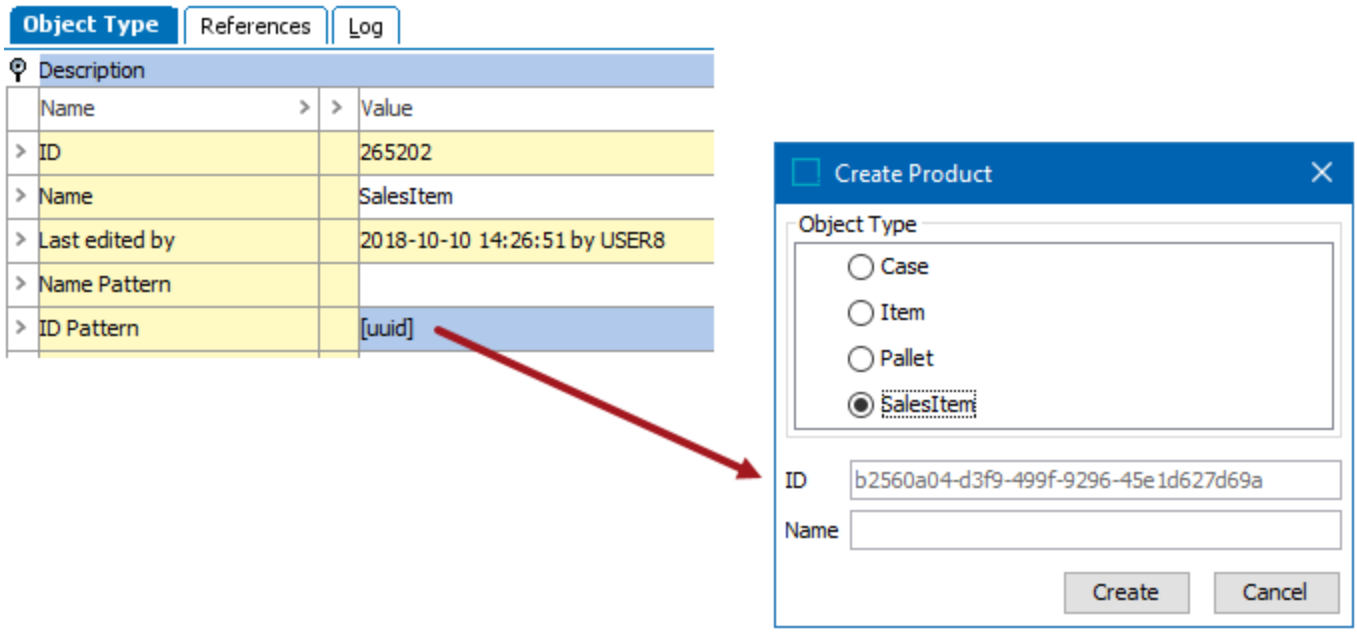


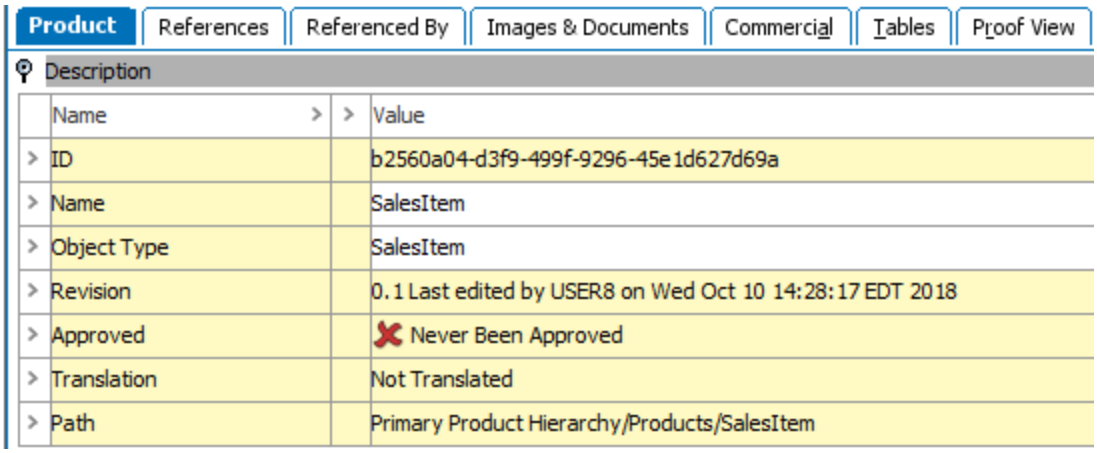
When a new node is created on the Tree for the Open Sales Item Object Type:

- The Name is 'SalesItem 107807' because in this case 107807 is the next unique number available to STEP Workbench.
- When both the Name Pattern and ID Pattern are specified, the ID number is the same as that for the Name. The counter does not increment just because the same tag [id] was used in both Pattern fields.



Below is an example using the [uuid] tag:





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 override the ID. Objects with user defined IDs can however still be created via data imports. This should generally be avoided, since mixing user defined and autogenerated IDs will almost certainly eventually lead to errors.
2. When manually creating a node where the Object Type has an entry for the Name Pattern, the autogenerated Name is presented as a suggestion and can be changed.
3. There is no guarantee that the generated ID is not already used by an existing object. This might happen if IDs are being generated both automatically and manually via an import.

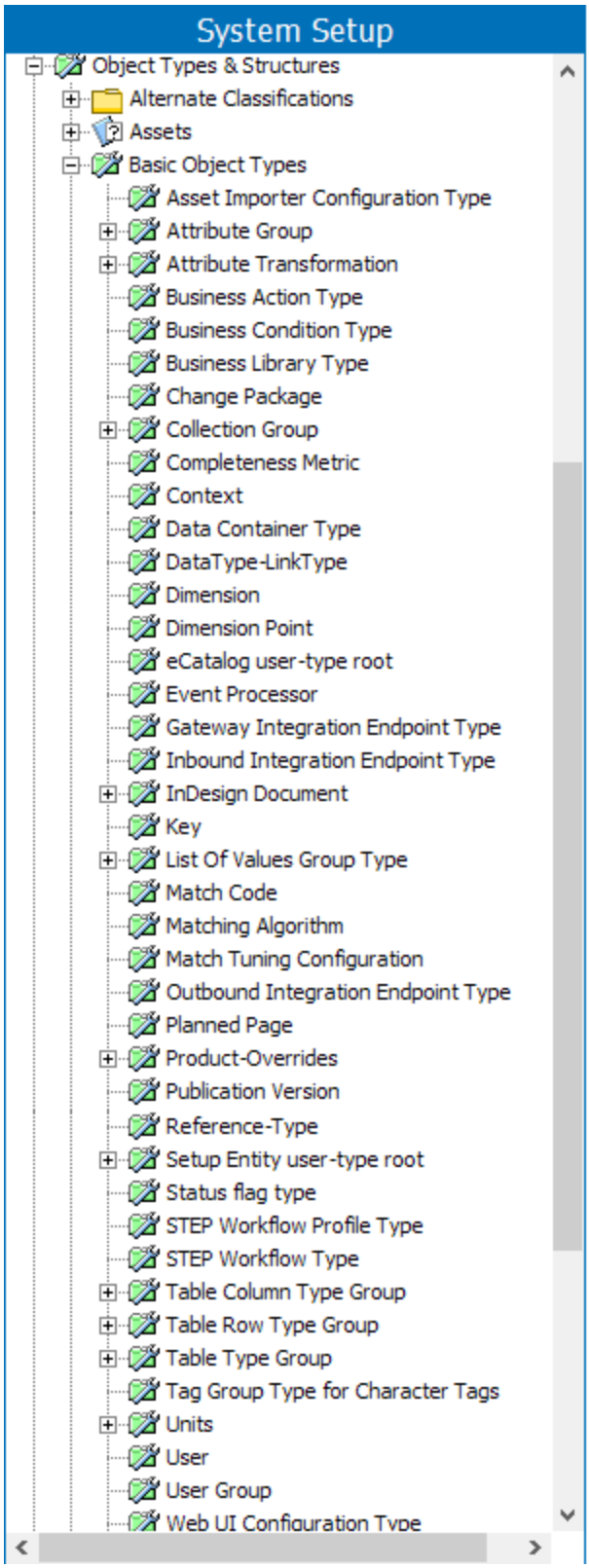
Basic Object Types

A basic object type is a node used for structuring the system data, for example, attributes, contexts, users, tables, etc. They are also called predefined object types and structures.

Whenever an object is created using a basic object type, the relevant object type is automatically applied. A user can define the automatic ID pattern and name pattern for the basic objects.

Basic object types cannot be manually created, however, when additional components are purchased and added to STEP, additional basic object types may display.

Basic objects are viewed under System Setup > Object Types and Structures > Basic Object Types. Below is one example of basic object types in a STEP instance.


System Setup
Object Types & Structures
Alternate Classifications
Assets
Basic Object Types
Asset Importer Configuration Type
Attribute Group
Attribute Transformation
Business Action Type
Business Condition Type
Business Library Type
Change Package
Collection Group
Completeness Metric
Context
Data Container Type
DataType-LinkType
Dimension
Dimension Point
eCatalog user-type root
Event Processor
Gateway Integration Endpoint Type
Inbound Integration Endpoint Type
InDesign Document
Key
List Of Values Group Type
Match Code
Matching Algorithm
Match Tuning Configuration
Outbound Integration Endpoint Type
Planned Page
Product-Overrides
Publication Version
Reference-Type
Setup Entity user-type root
Status flag type
STEP Workflow Profile Type
STEP Workflow Type
Table Column Type Group
Table Row Type Group
Table Type Group
Tag Group Type for Character Tags
Units
User
User Group
Web UI Configuration Type

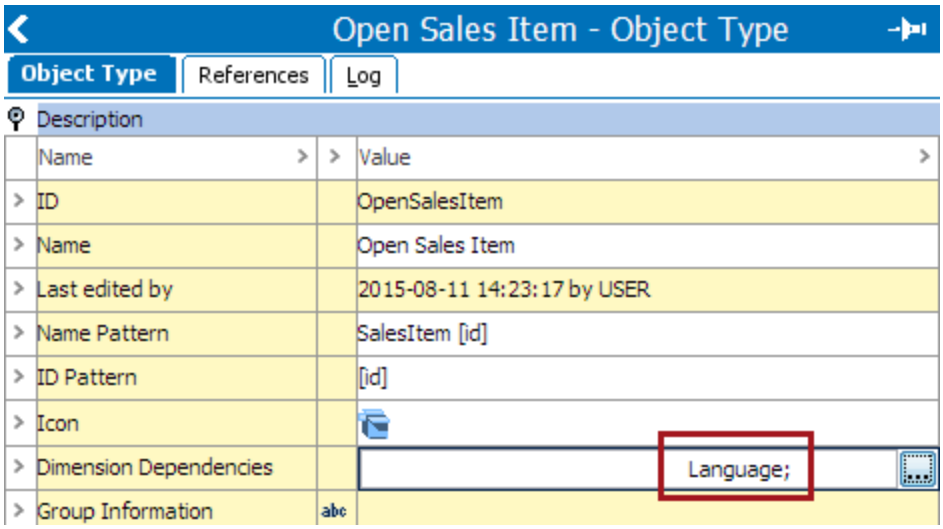
Note: Basic object types cannot be deleted.

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.



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.

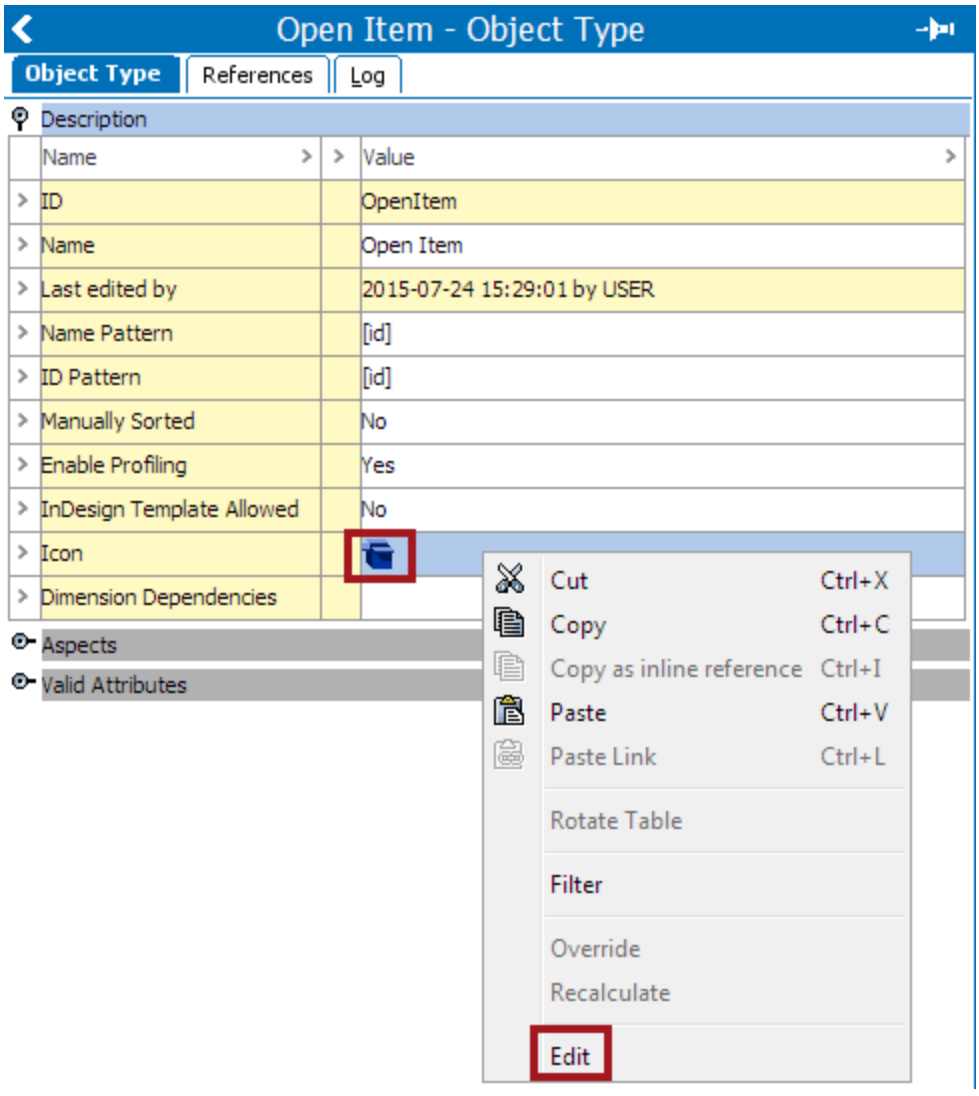
Object Type Icons

All objects in STEP, both system-specific and user-created, have an icon assigned to them. These icons allow users to easily differentiate between different object types when navigating in the Tree. STEP provides a series of default icons for system-specific object types (like Primary Product Hierarchy and the Classification root node, for example) and then a series of icons that can be assigned to user-created object types. STEP also supports creation and assignment of custom icons that users create outside of STEP.

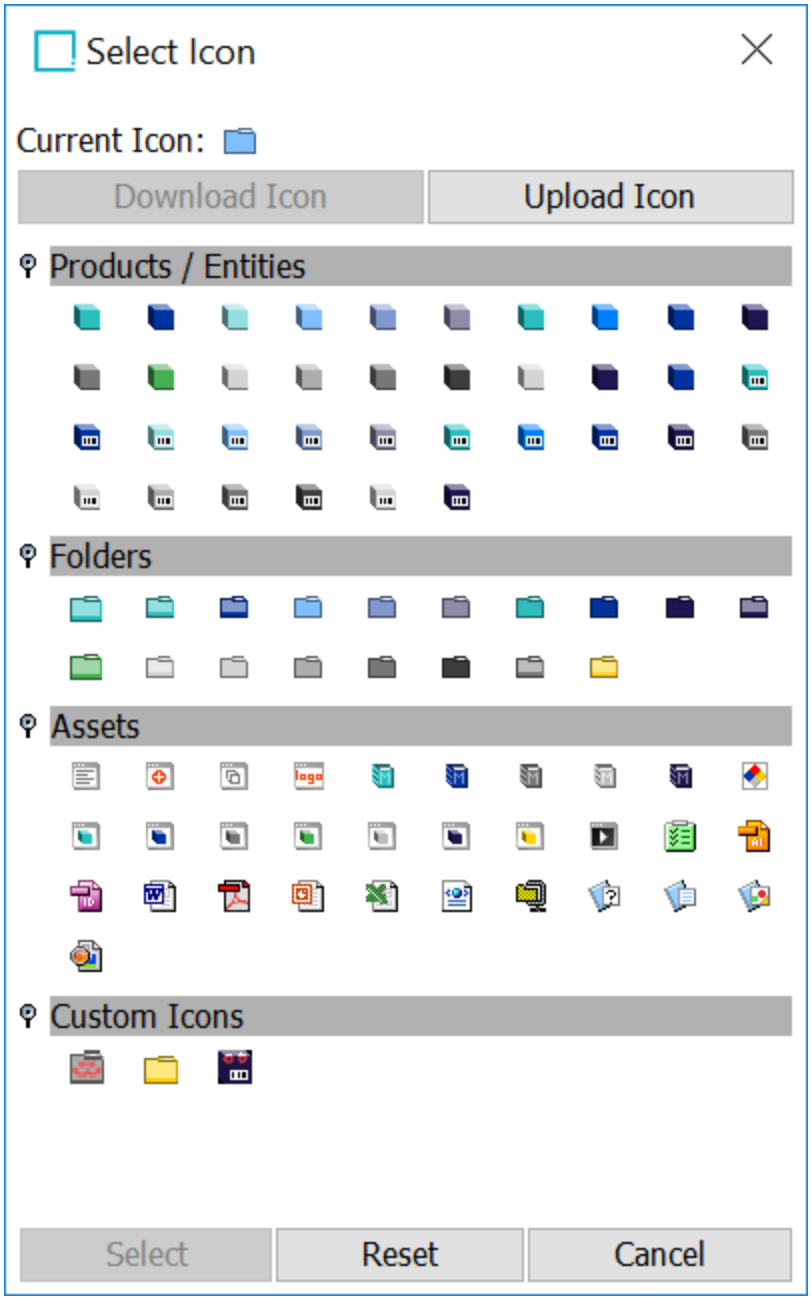
Editing an Object Type Icon

To make changes to an icon assigned to a given object type, take the following steps:

1. On the System Setup tab, under the 'Object Types & Structures' node, select the 'Icon' field of any listed object type, right-click, and then select the **Edit** option at the bottom of the dropdown that displays.



2. The 'Select Icon' dialog displays.



The 'Select Icon' dialog contains information for users to reference as well as actions users may take.

Current Icon: Displays the icon for the selected object type. To download the icon assigned to the current object type, users must first highlight it by clicking the icon that displays.

Download icon: Once an icon is selected, users click this button to bring the selected icon image file onto their own system in order to customize the icon. Users may download both the icon currently assigned to the selected object type, or any of the grouped icons displayed in the 'Select Icon' dialog.

Upload icon: Users click this button to load a custom icon from an external source directly into STEP. Once uploaded, the icon will appear in the list of grouped icons under 'Custom icons.'

Product / Entities: Icons displayed under this collapsible header are well-suited for assignment to user-created product and entity object types.

Folders: Icons displayed under this collapsible header are well-suited for assignment to parent object types.

Assets: Icons displayed under this collapsible header are well-suited for assignment to asset object types.

Custom icons: Icons displayed under this collapsible header are custom icons, or icons that were made outside of STEP and uploaded. Icons that display in this group may be custom icons that are already assigned to an object type, or custom icons that have not yet been assigned to any object type. Any icons uploaded to STEP using the 'Upload icon' button will display.

Select: Click this button to assign the selected icon to an object type. This button is, effectively, the 'Assign' button, and completes the task of assigning an icon to an object type.

Reset: Click this button to revert the icon assigned to the selected object type to the system-designated default icon.

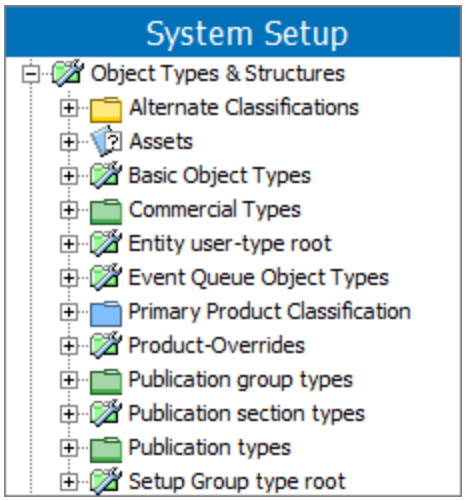
Cancel: Closes the 'Select Icon' dialog.

3. When the 'Select' button is clicked, the icon assignment is made. The 'Select Icon' dialog closes and the selected icon immediately displays on the object type in the Tree.

Though the core STEP-specific icons, like the blue folder icon for 'Primary Product Hierarchy' for example, can be changed, it is recommended that they be left either unaltered or modified only slightly. The reason for this is that changing them may cause conflicts or ambiguities with the descriptions and / or screenshots in the online help.

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**.



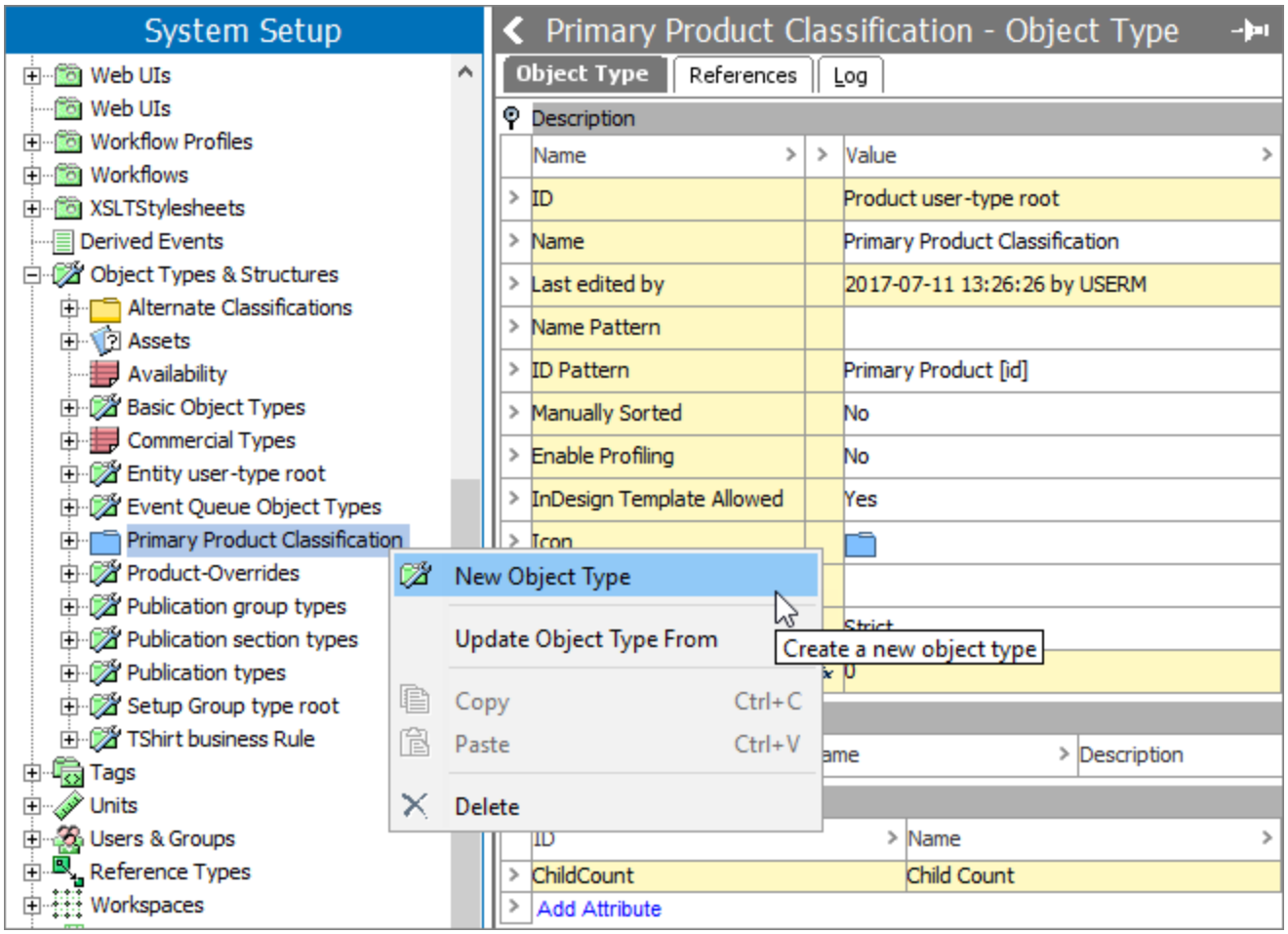
Creating an Object Type

The following standard root object types 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.

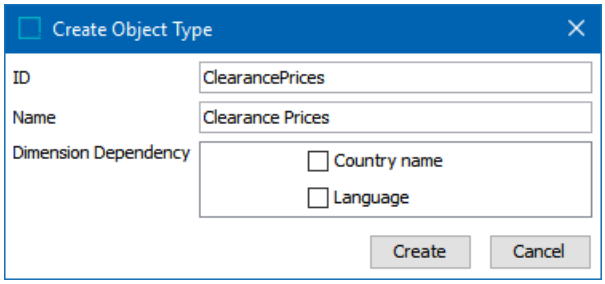
Create an Object Type

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.



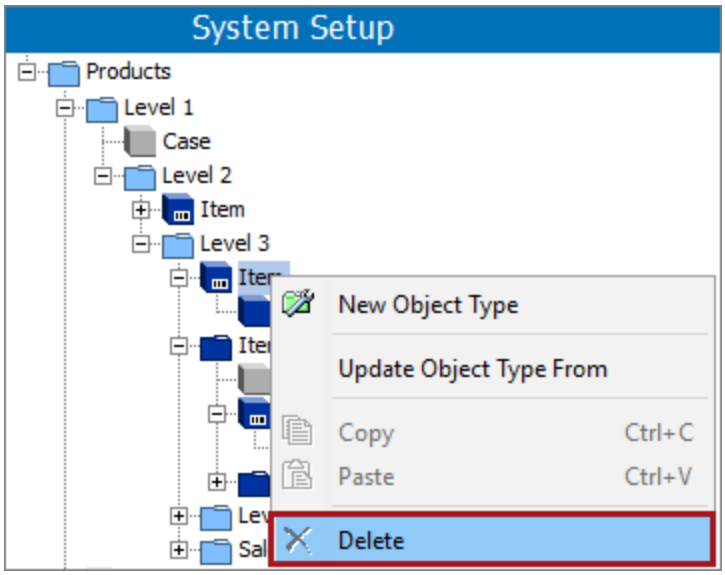
> **Note:** If the Product root object type has an ID Pattern enabled, then the ID will display auto populated, and not editable. For more information see the **Autogenerate Using Name Pattern and ID Pattern** topic within this guide.

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. On the Object Type editor, References tab, use the Parents flipper to define which object types can hold the newly created object type. For more information, see the **Object Type Hierarchy** topic.
7. 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.

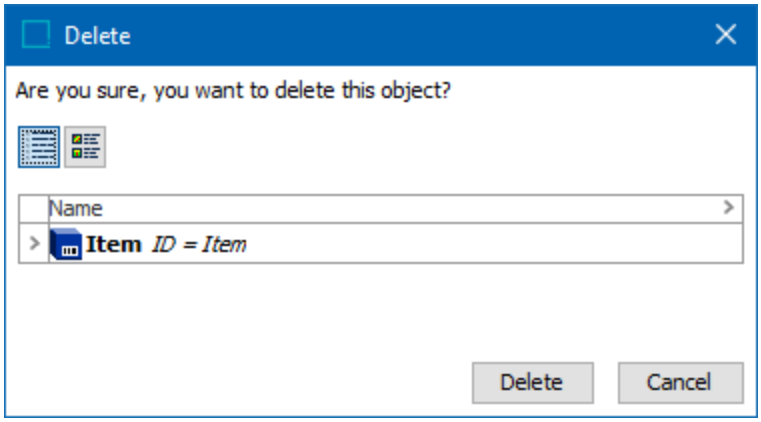
Deleting an Object Type

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.

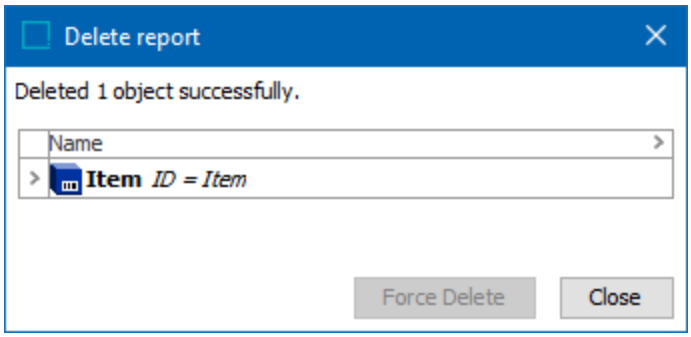


2. The Delete dialog will display asking you to confirm the deletion.

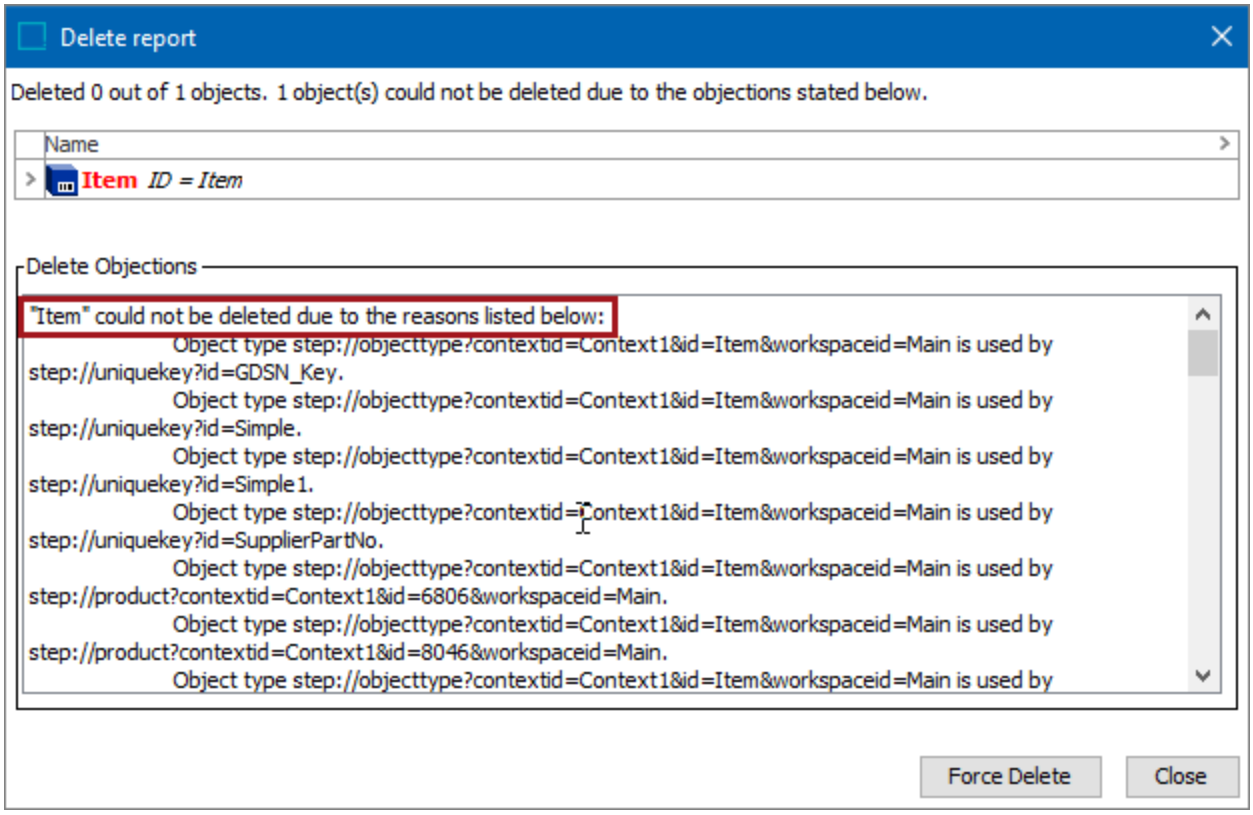


3. Click the **Delete** button, and the Delete report dialog will display.

   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.

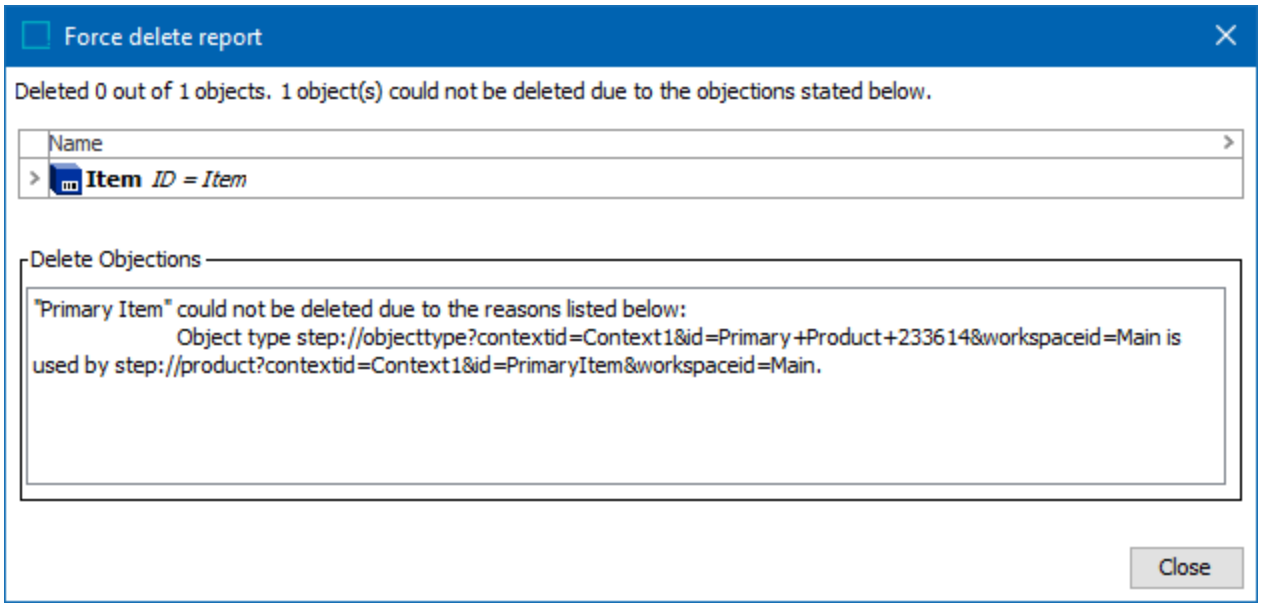


If the object is not successfully deleted, then the Delete report will display with Delete Objections as shown below.



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.

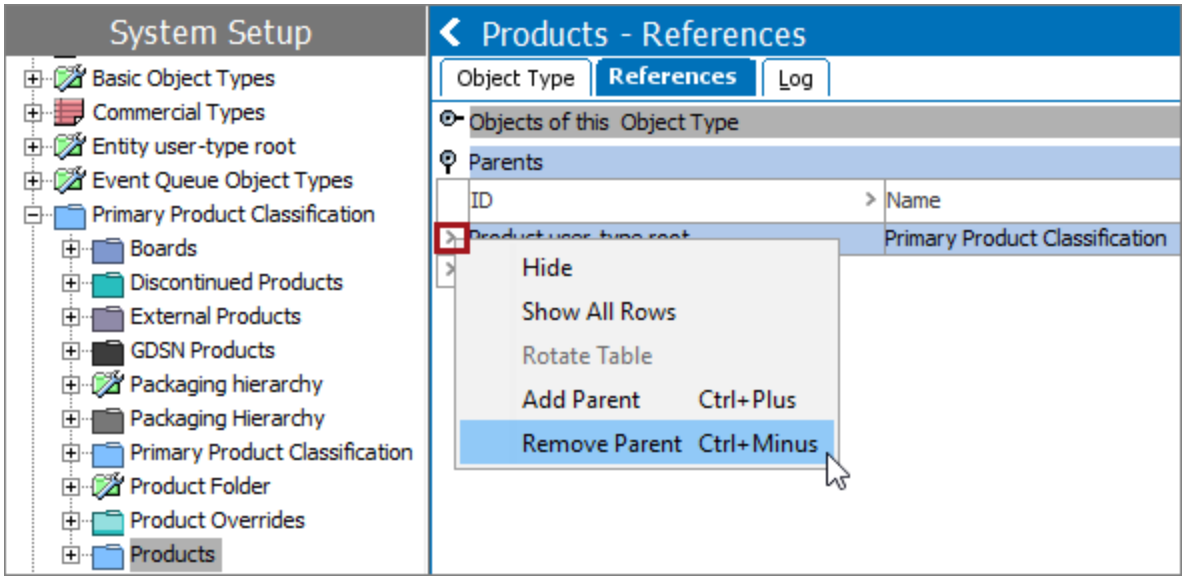

Force delete report
Deleted 0 out of 1 objects. 1 object(s) could not be deleted due to the objections stated below.
Name
Item ID = Item
Delete Objections
"Primary Item" could not be deleted due to the reasons listed below:
Object type step://objecttype?contextid=Context1&id=Primary+Product+233614&workspaceid=Main is used by step://product?contextid=Context1&id=PrimaryItem&workspaceid=Main.
Close

Unlinking Object Types

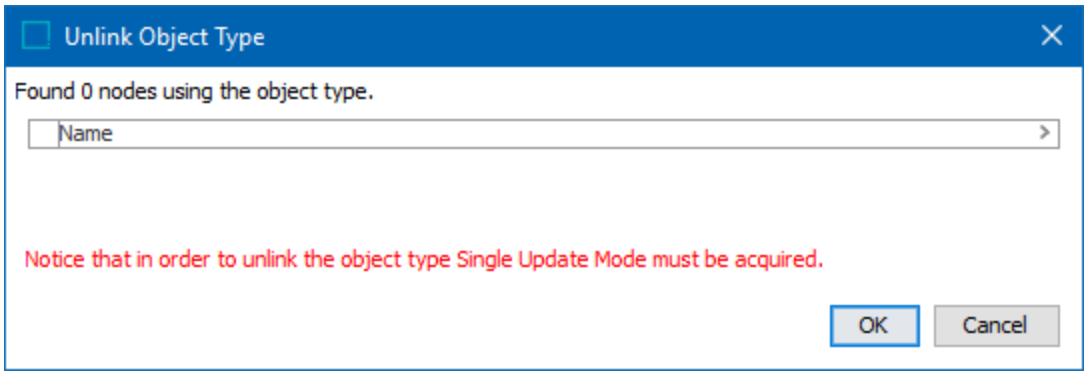
Unlinking an object type requires the object maintains at least one parent, all children must be removed from parents of this parent type in all Workspaces, and the system must enter Single-Update Mode. For more information, see the **Single-Update Mode** topic within the **System Setup / Super User Guide**.

Below are the steps to unlink an object type in System Setup.

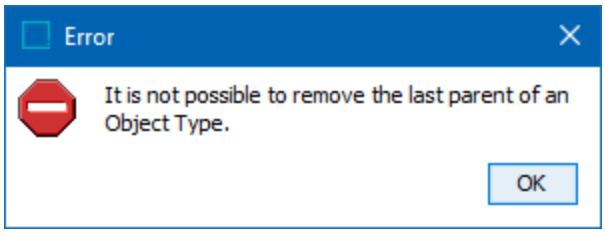
1. Go to **System Setup** > Expand **Object Types & Structures** > Select **a relevant object type** and the **Object Type Editor** will display.
2. Click the **References** tab.
3. Expand the **Parents** flipper, right-click the row '>,' and click the **Remove Parent** menu option (as shown below).



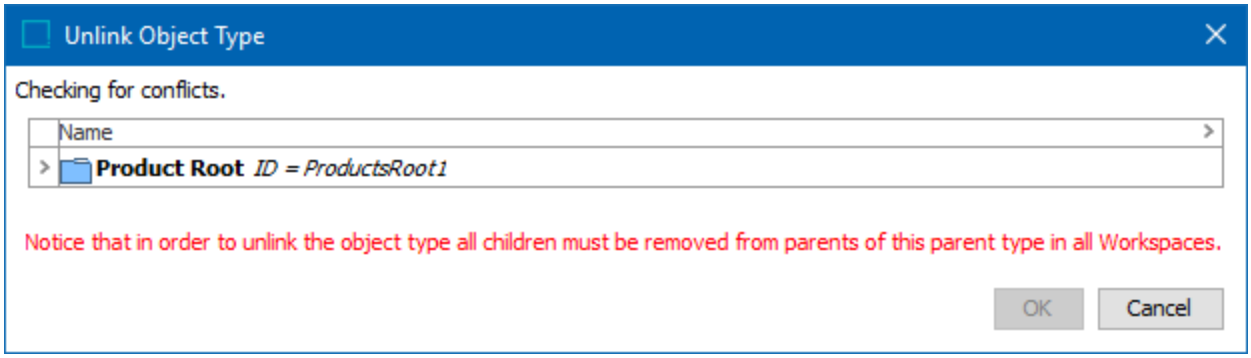
4. If more than one parent exists, and no children exist, then the Unlink Object Type dialog will display as shown below. Click the **OK** button and the Object Type Editor will display with the parent removed. Click the **Cancel** button to display the Object Type Editor with no changes.



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.

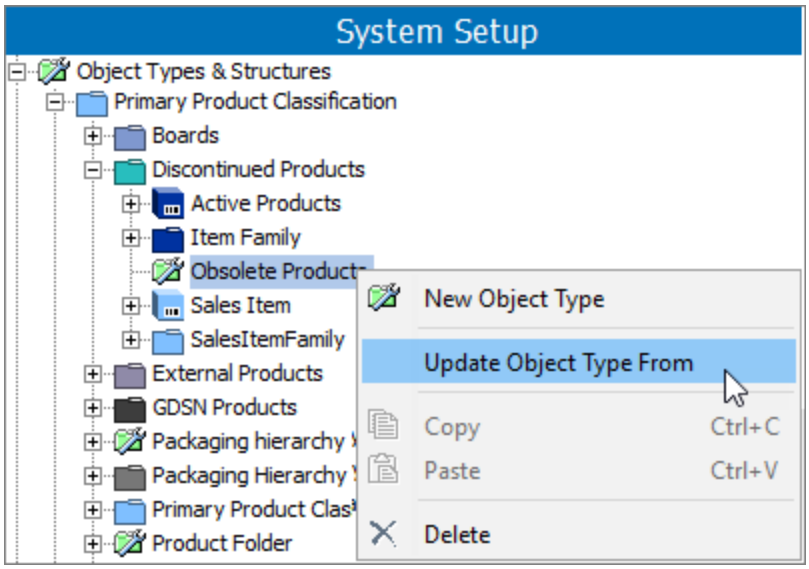
This concept could also be used for Onboarding products, so that there would be a temporary object type being assigned until the product is enriched and ready to be published, at which time its object type would be changed to the actual object type. This would defend against publishing incomplete data 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**.

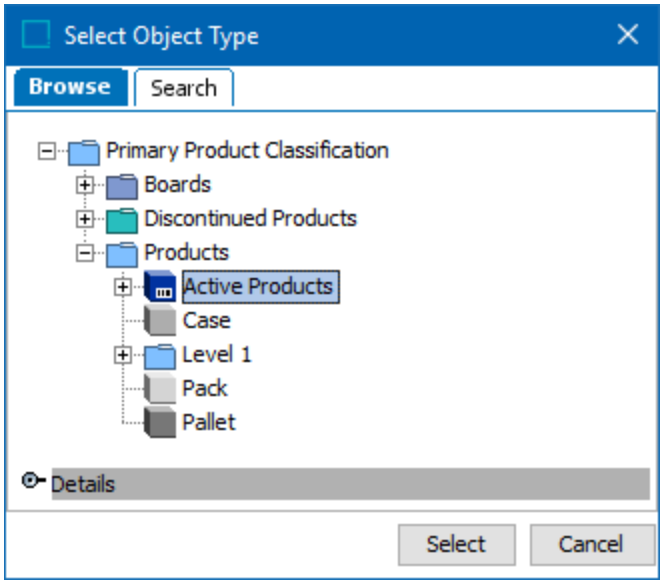
Steps to Update an Object Type from Another Object Type

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.

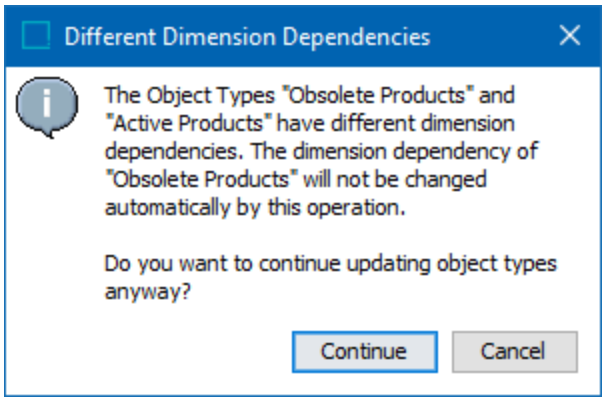


3. The Select Object Type dialog will display. Use the Browse or Search tabs to find the **object type with the validity settings that need to be applied**. For this example, the 'Active Products' object type is selected.



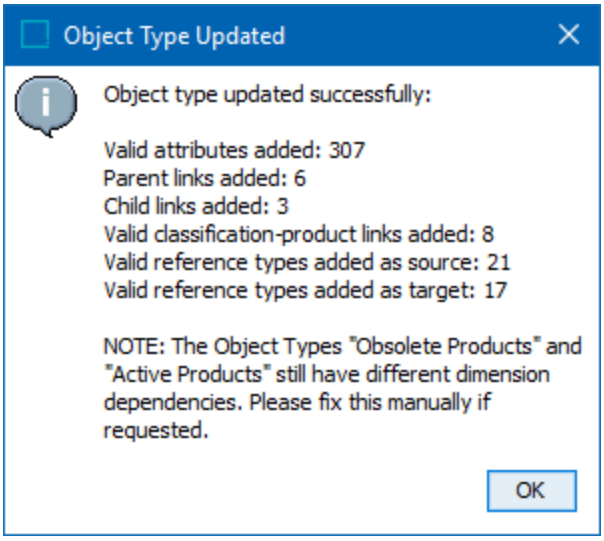
4. Click the **Select** button.

   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.

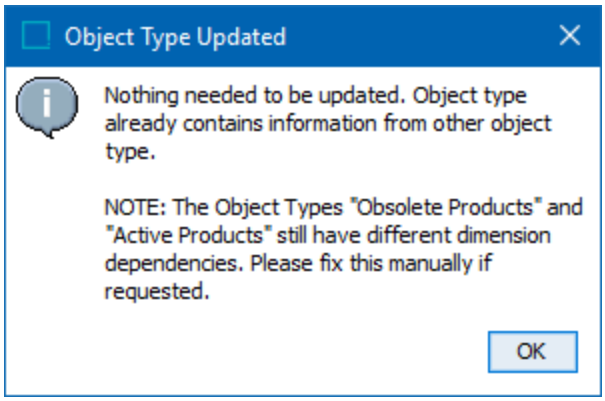


5. The Object Type Updated status dialog will display indicating how the selected object type was updated by listing the number of:

    - 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

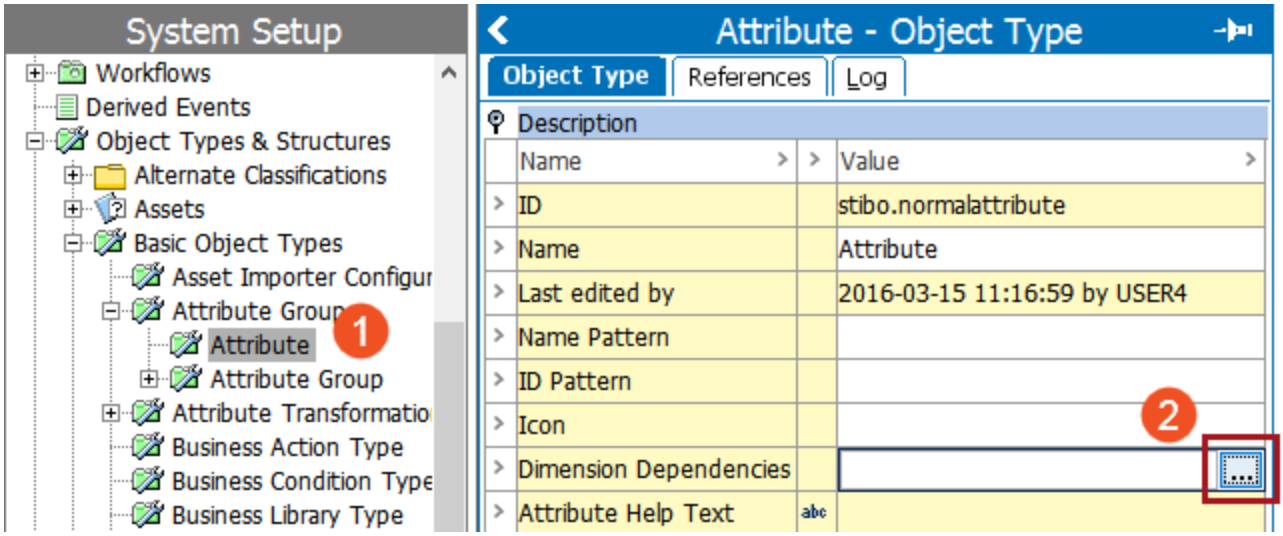
Dimension dependency is set for objects on their respective object types in System Setup. Almost all object types in STEP (both basic and non-basic) can be made dimension dependent so that their names will display in different languages in different contexts.

- **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

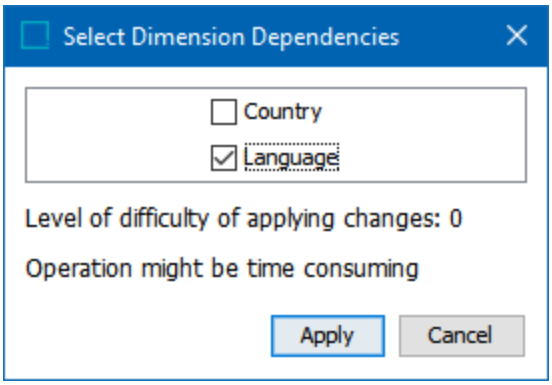
Making an Object Type Dimension Dependent

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.

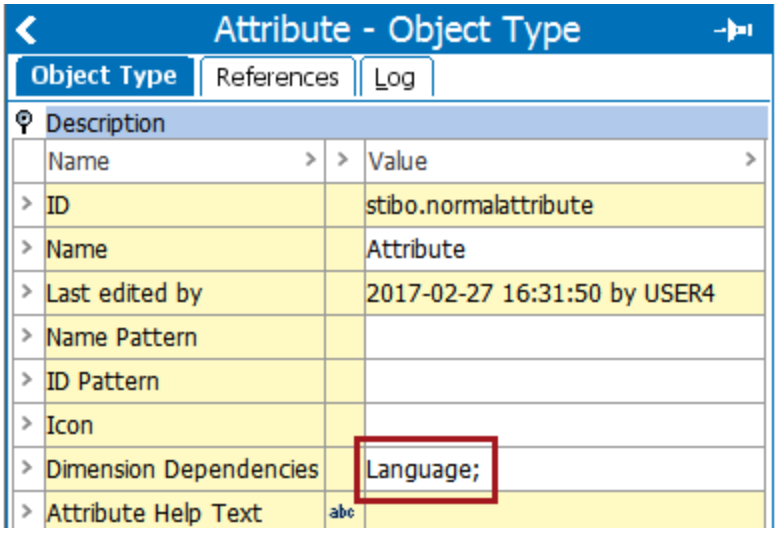
1. Go to System Setup, and navigate to object type that you would like to make dimension dependent.
2. On the **Object Type** tab, double-click in the **Dimension Dependencies** field, then click the ellipsis button (...).



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



4. The object type is now dimension dependent, and the STEP names of any object using this object type are now dimension dependent.

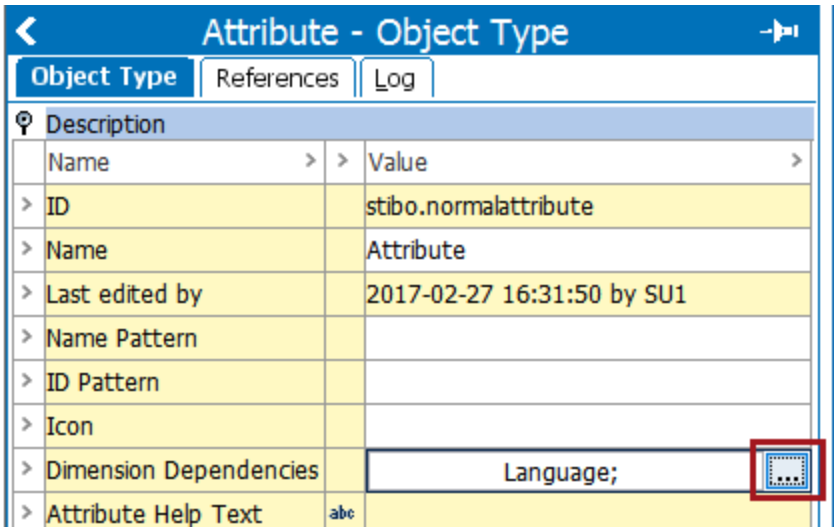


Removing Dimension Dependency from an Object Type

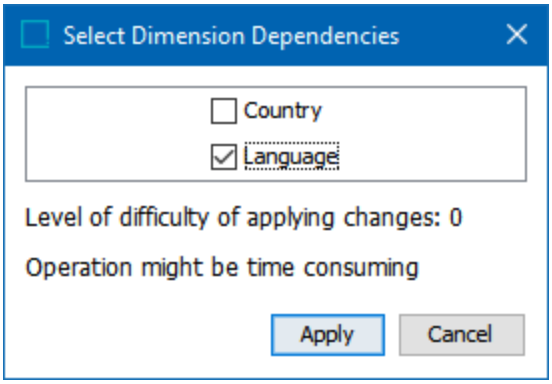
Dimension dependency is removed from an object in a near-identical fashion that it is applied.

Note: The dimension dependency should normally not be changed. Be aware of possible conflicts if a dimension dependency is removed.

1. Navigate to the relevant object type in System Setup.
2. Double-click in the **Dimension Dependencies** field, then click the ellipsis button (...).

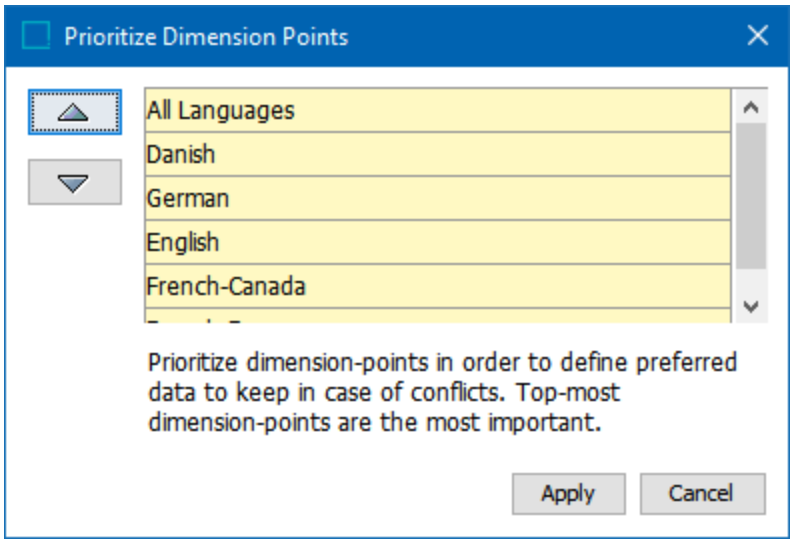


3. In the **Select Dimension Dependencies** dialog that displays, uncheck the box(es) for the relevant dimension(s), then click the **Apply** button.



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/***.

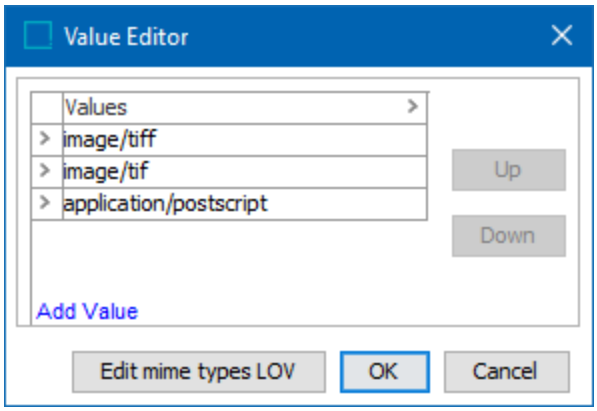
MIME Type	Definition
application/*	All
application/pdf	Acrobat PDF
application/illustrator	Adobe Illustrator
application/vnd.ms-excel	Microsoft Excel
application/vnd.ms-powerpoint	Microsoft PowerPoint
application/msword	Microsoft Word
application/postscript	PostScript (including .eps files)
application/x-indesign	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/svg+xml	SVG image*
image/tif	TIF image
image/tiff	TIF image
image/webp	WebP image; Google file format
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

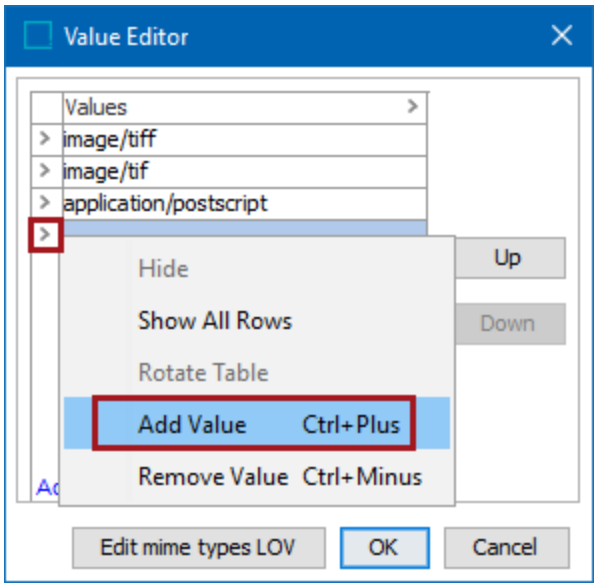
Important: *Caution should be used when importing **SVG Image** files since they can contain harmful scripts if images from unknown sources are loaded.

Setting MIME Types for Object Types

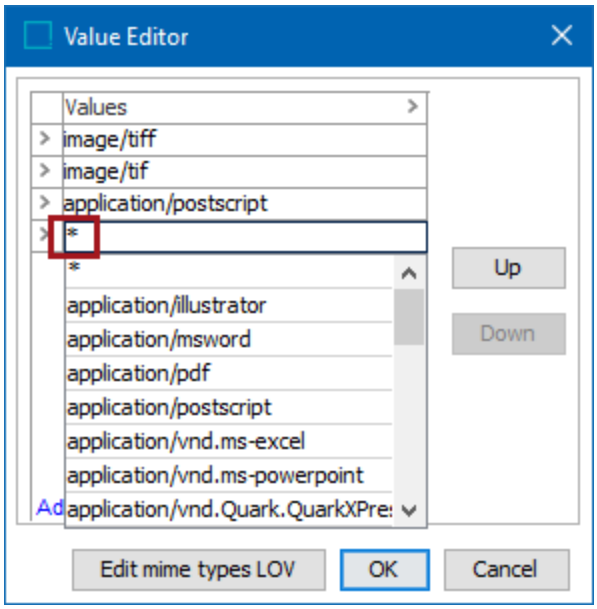
1. Go to **System Setup** > Select **Object Types & Structures** > **Assets** > select an **Image** or **Document Object Type**. (Mime Types are not valid for any other object types.) The corresponding editor will display.
2. Double-click in the **MIME Types** field to display the **Value Editor** dialog.



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**.

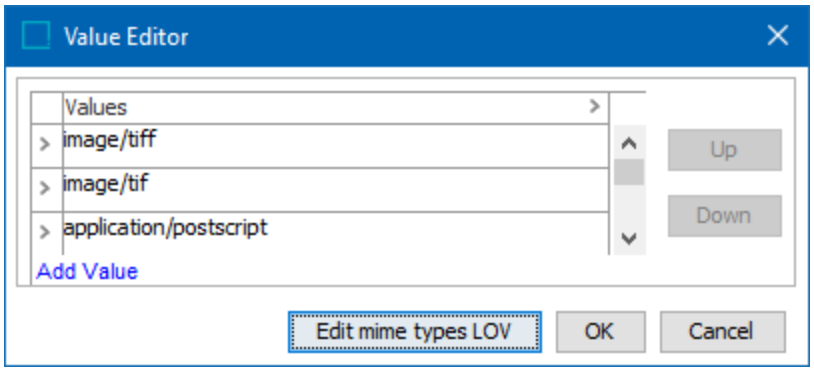


5. Repeat steps 3 and 4 to select more MIME Types.
6. Click **OK** to display the Object Type Editor assigned MIME Type(s).

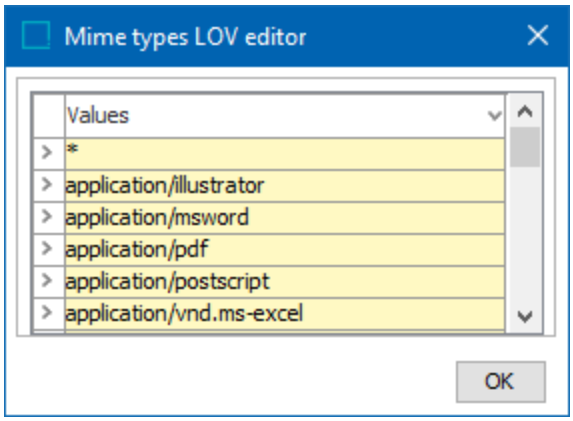
   When importing an image or a document, it is automatically assigned the Object Type with the specific MIME Type.

Editing MIME Types LOV

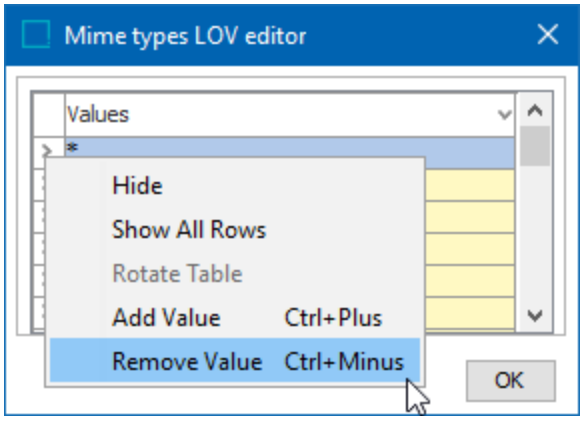
1. In Value Editor click on the 'Edit MIME types LOV' button.



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.

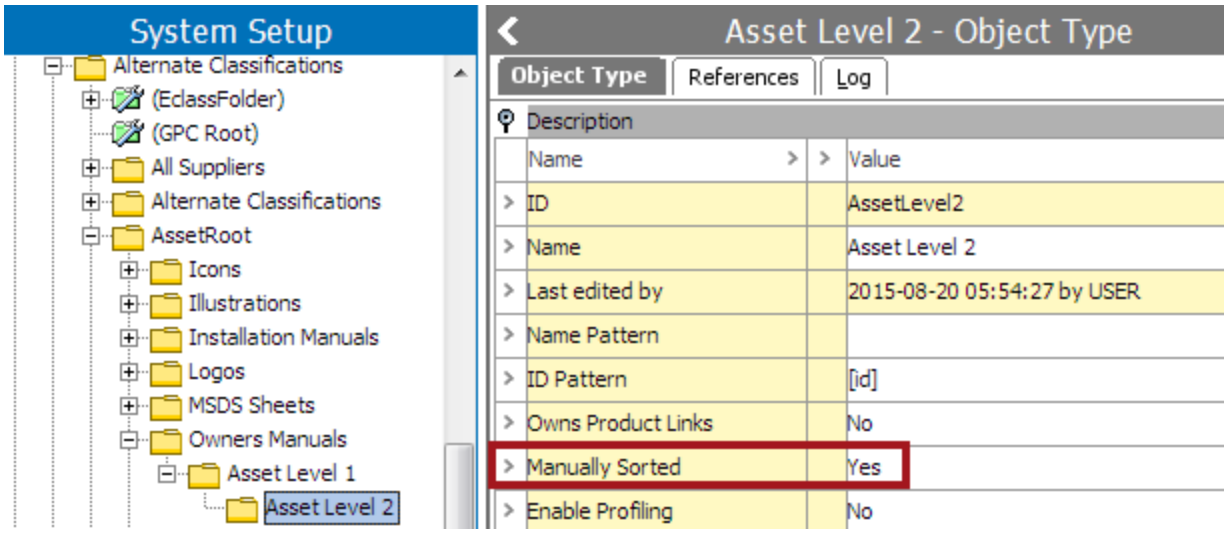


- Click Hide to no longer display a MIME type.
- Click Show All Rows to display all existing MIME types.
- Click Remove Value to delete the MIME type.

Modifying Manually Sorted

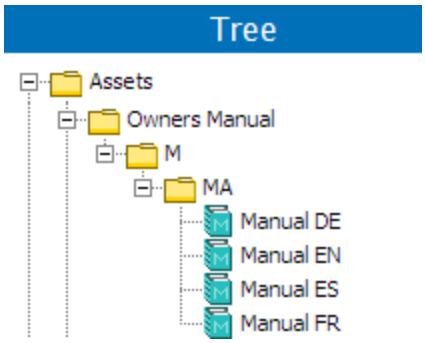
In the Tree, objects in the Alternate Classifications hierarchy, Primary Product Classification hierarchy, and Publication hierarchy sort alphabetically by default. This setting can be modified.

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.

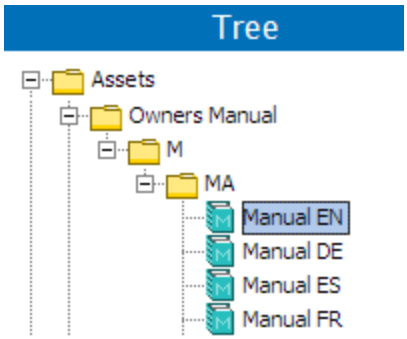


2. Set the **Manually Sorted** field:

   - **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.



   - **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.



Note: In Web UI, the Sequence view on a Node list allows for the manual sorting of objects and classifications. For more information see the Node List Component topic in the Web User Interfaces / Web UI Setup and User Guide documentation.

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
Entities	Defines the structure of the Entity hierarchy.	• Customers • Addresses

Object Type	Definition	Example
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:** Basic objects cannot be edited, nor 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.

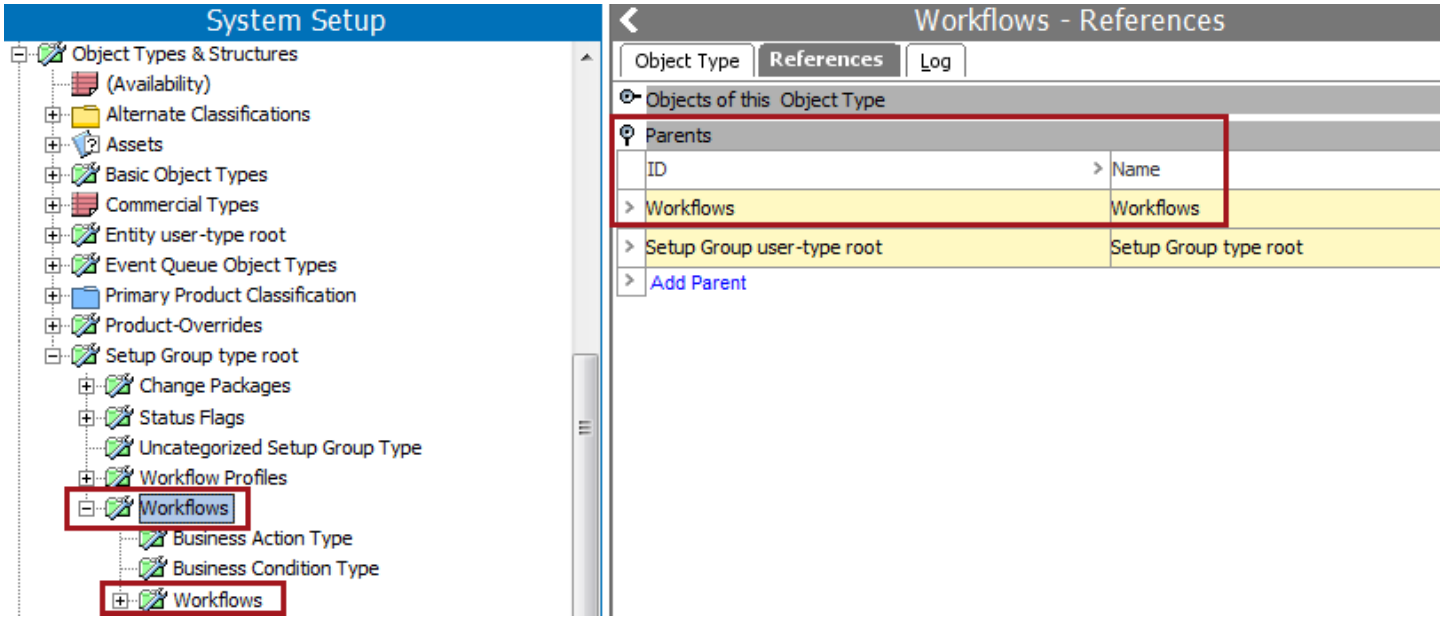
Child nodes of such Object Types are also referred to as 'leaf' nodes in the hierarchies.

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.



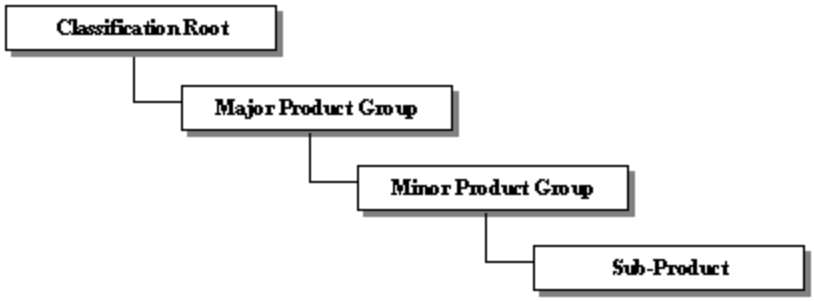
Classification Hierarchy Structures

The two hierarchy structures that may be defined are:

- Classification hierarchy
- Product hierarchy

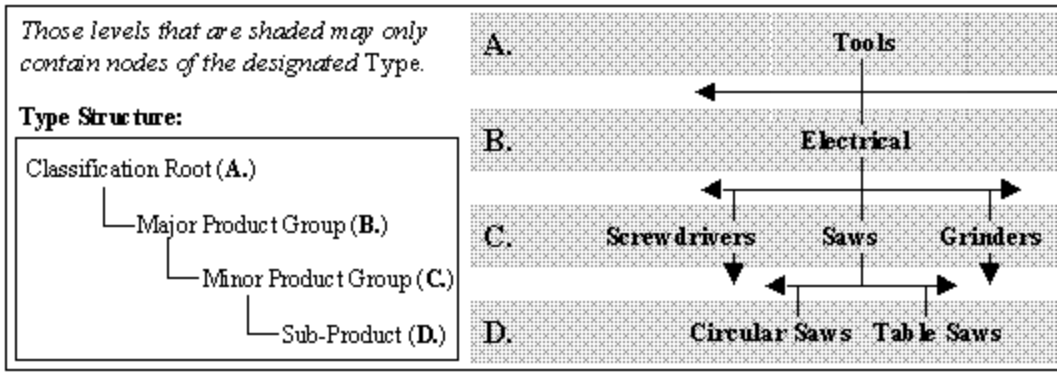
Note: There is no hierarchy that can be created for assets and entities.

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.

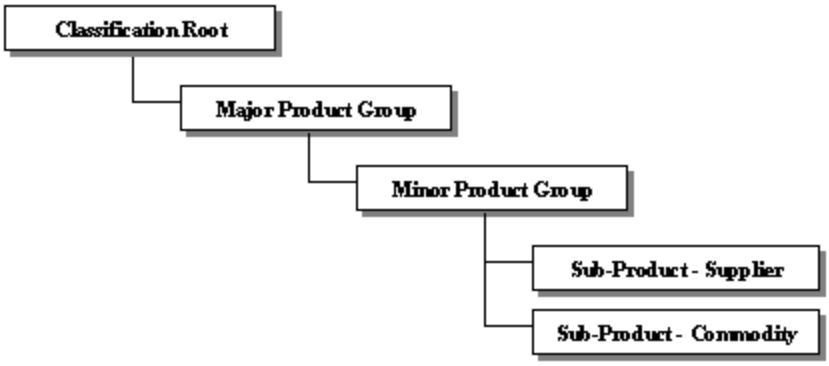


In this example, only a **Major Product Group** may be a child of a **Classification Root**, a **Minor Product Group** may only be a child of a **Major Product Group**, etc. As defined by this structure, it is not possible to create a **Sub-Product** under the **Classification Root**. Any nodes of the 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:



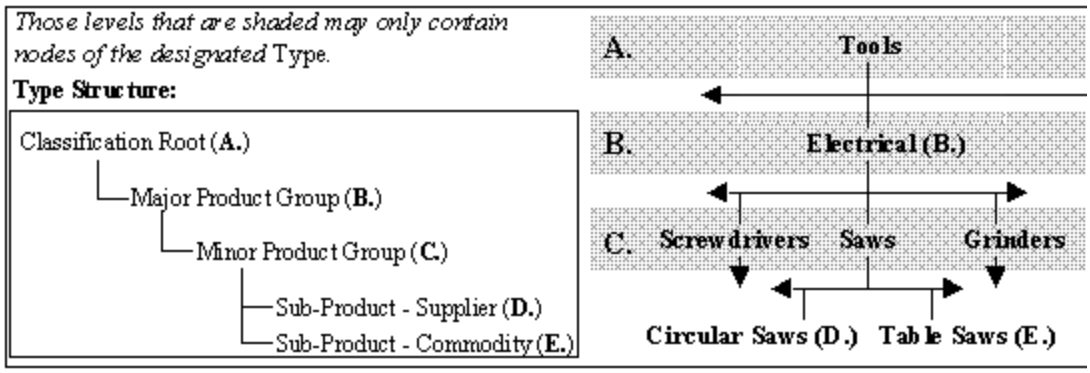
The first example illustrates a simple structure where each object type defines a level in the hierarchy. To build on this illustration, the second example shows it is also possible to create an object type structure in which a parent object type may have multiple children.



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**

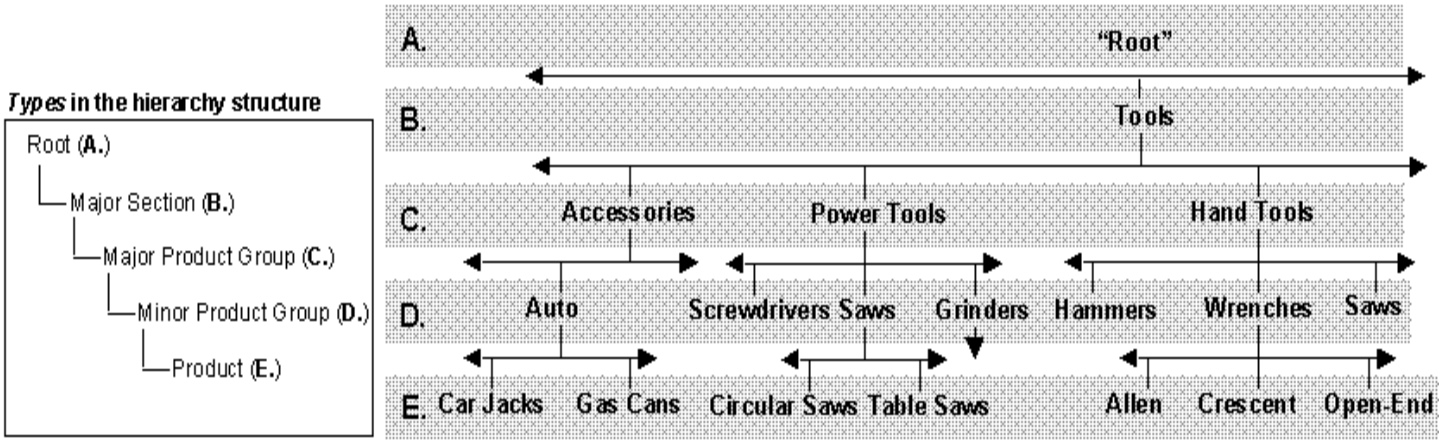
Fixed Hierarchy Structure

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'.

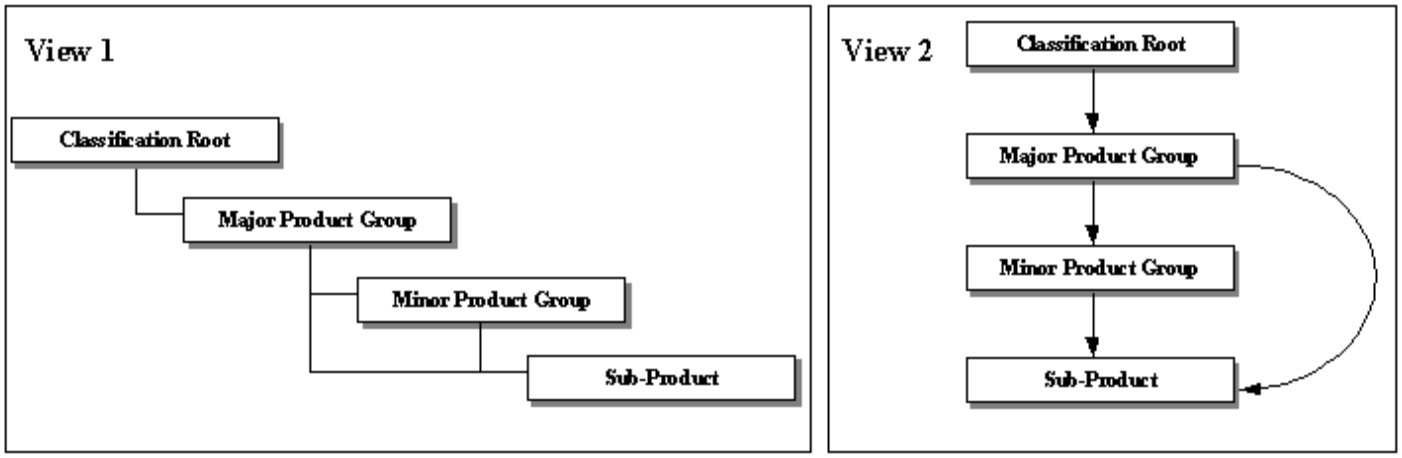


For information, see the Flexible Hierarchy Structure section.

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.

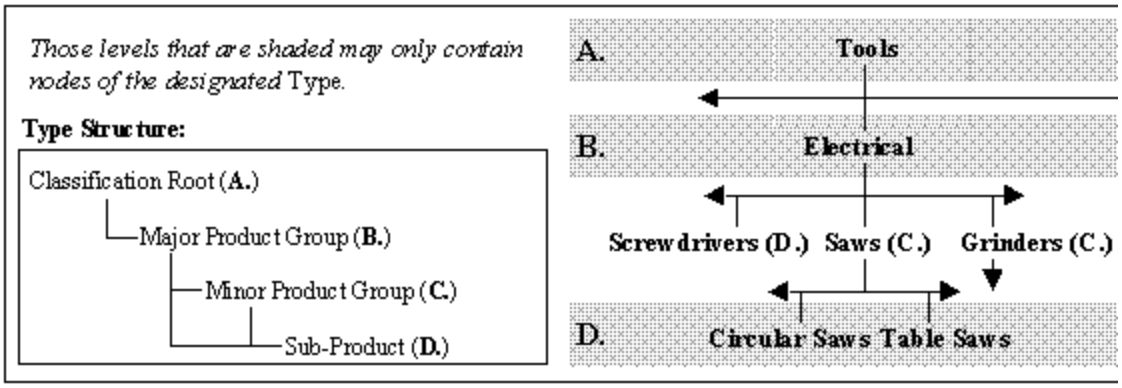
For example, the leaf nodes are of the same object type, but may be created in different levels. This is accomplished by allowing an object type to have multiple parents and displayed in the figure below.



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**.

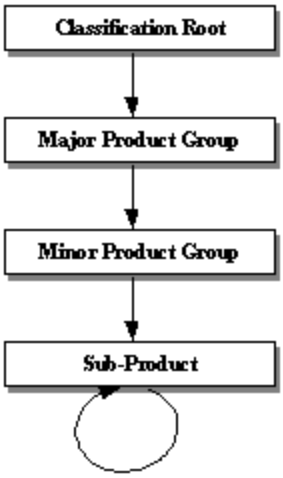
The following figure represents a hierarchy based on a varying object type structure.



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.

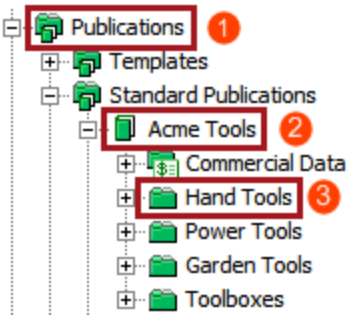


Creating a Standard Publication Object Type Hierarchy

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.

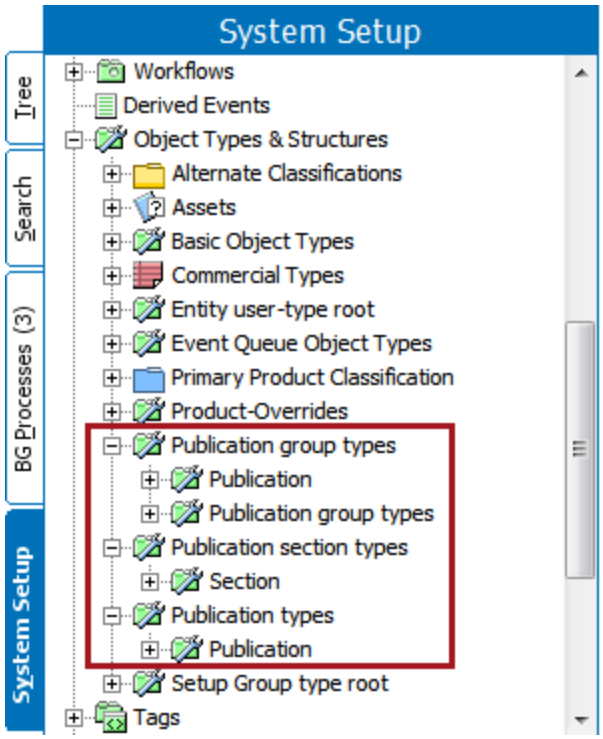
The following screenshot shows what a basic publication hierarchy looks like in the Tree. The publication object types addressed in this documentation are used to build the following objects:

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, fliers, 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

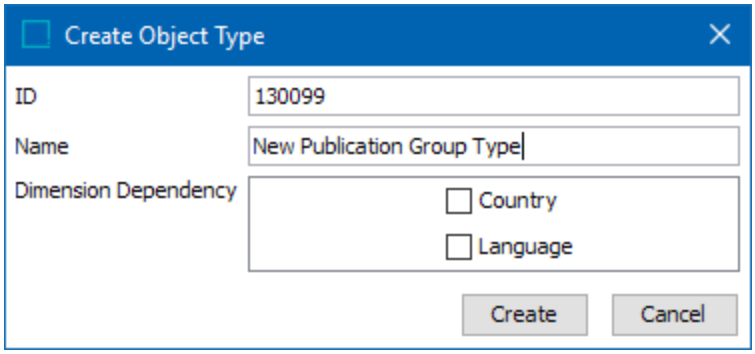
The following sections of this documentation will guide you through the process of creating new publication hierarchy object types as well as configuring valid parent / child relationships between these object types.

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.

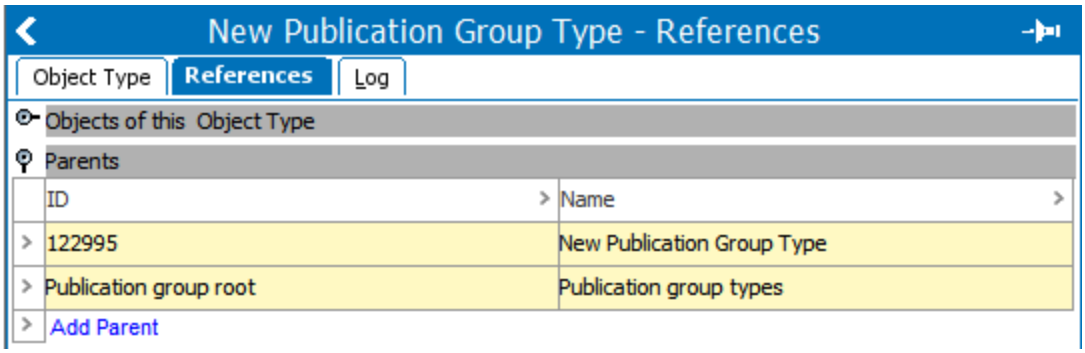
For more information on the creation and configuration of publication hierarchy objects in the Tree, see the **Publication Hierarchy** section of the **STEP Publisher** documentation.

Creating a Publication Group Object Type

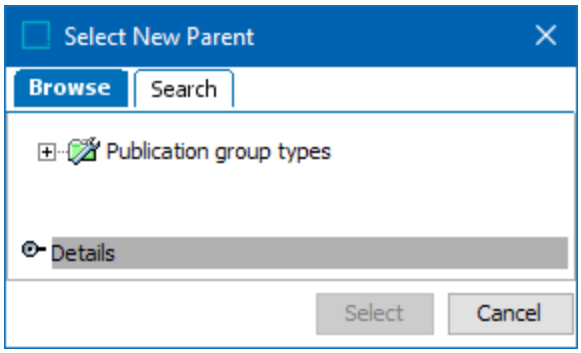
1. In System Setup, open the **Object Types & Structures** hierarchy.
2. Select **Publication group types**, right-click, and then select **New Object Type**. A **Create Object Type** dialog box appears.



3. If the STEP ID for the **Publication group root** object has not been set to autogenerate with an ID Pattern, type an ID in the **ID** field.
4. In the **Name** field, type a name.
5. Check the relevant **Dimension Dependency** check boxes, if applicable. (Dimension dependencies are optional, and are not typically used with publication group object types.)
6. Click **Create**. An Editor displays for the newly created publication group object type.
7. Click the **References** tab.



8. Click **Add Parent** to define which parent publication group object(s) should be valid for the new publication group object. The **Select New Parent** dialog box appears.

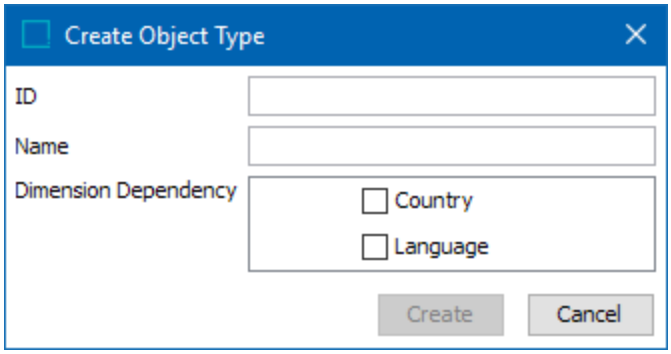


9. Search or browse for the relevant publication group type, then click **Select**.

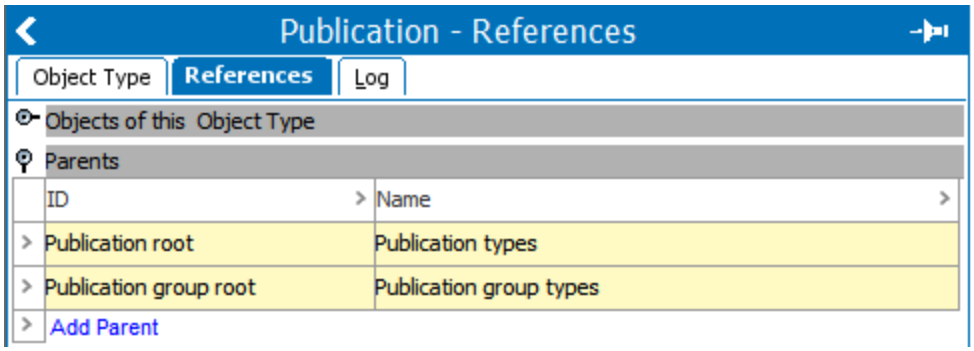
Note: Publication group objects may only be children of other publication group objects. They may not be children of publication objects or section objects.

Creating a Publication Object Type

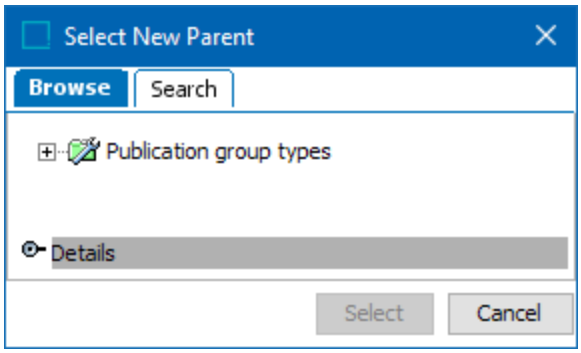
1. In System Setup, open the **Object Types & Structure** hierarchy.
2. Click **Publication types**, right-click, and then click **New Object Type**. A **Create Object Type** dialog box appears.



3. If the STEP ID for the 'Publication root' object has not been set to autogenerate with an ID Pattern, type an ID in the **ID** field.
4. In the **Name** field, type a name.
5. Check the relevant **Dimension Dependency** check boxes, if applicable. (Dimension dependencies are optional, and are not typically used with publication object types.)
6. Click **Create**. An editor displays for the newly created publication object type.
7. Click the **References** tab.



8. Click **Add Parent** to define which parent publication group object(s) should be valid for the publication object. The **Select New Parent** dialog box appears.

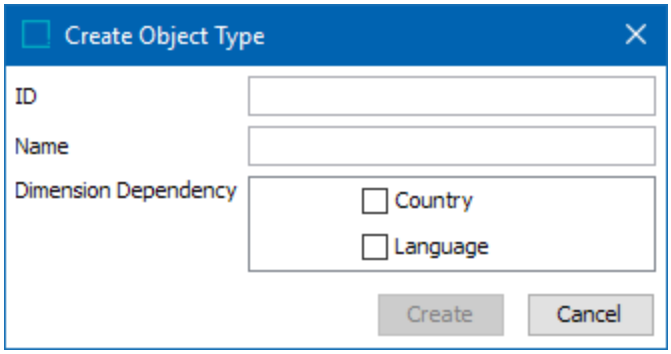


9. Search or browse for the relevant publication group type, then click **Select**.

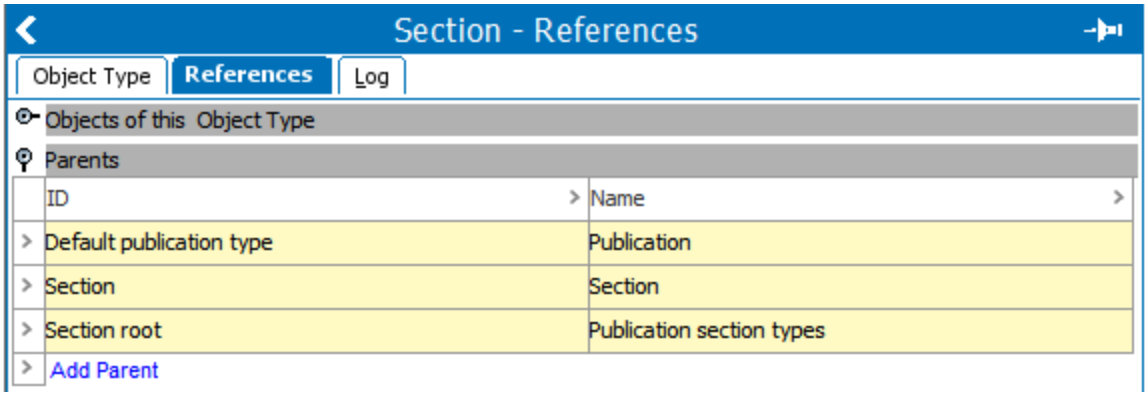
Note: Publication objects may only be children of publication group objects. They may not be children of other publication objects or section objects.

Creating a Publication Section Object Type

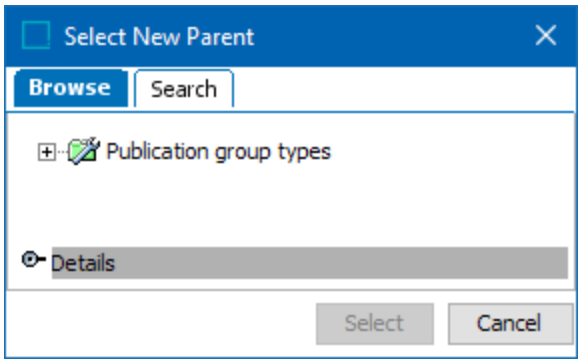
1. In System Setup, open **Object Types & Structures**.
2. Click **Publication section types**, right-click, and then select **New Object Type.** The **Create Object Type** dialog appears.



3. If the STEP ID for the 'Section root' object has not been set to autogenerate with an ID Pattern, type an ID in the **ID** field.
4. In the **Name** field, type a name.
5. Check the relevant **Dimension Dependency** check boxes, if applicable. (Dimension dependencies are optional, and are not typically used with publication section object types.)
6. Click **Create**. An Editor displays for the newly created Section object type.
7. Click the **References** tab.



8. Click **Add Parent** to define which parent publication and/or section object(s) should be valid for the section object. A **Select New Parent** dialog box appears.



9. Search or browse for the relevant publication and/or section type, and then click **Select**.

Note: Publication section objects may be children of publication objects and other section objects but they cannot be children of publication group objects.

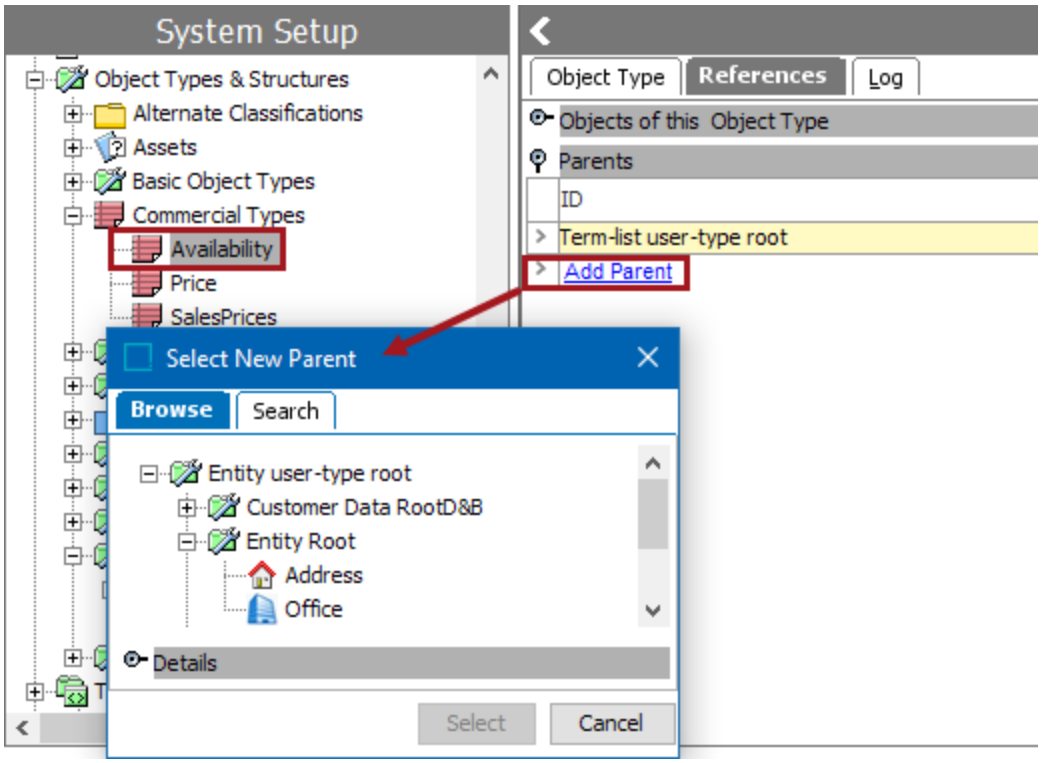
Product-Override Object Types and Commercial Object Types

Product-Override object types and Commercial object types appear in their own hierarchy under Object Types & Structures.

Before using Product-Overrides in the Product hierarchy or commercial data in the Entity hierarchy, the levels in which those objects can be legally created / maintained must be specified.

- Product-Override object types must be linked into the Product Object Type hierarchy.
- Commercial data types must be linked into Entity object types if they are needed within an Entity hierarchy. (Commercial Types can be used directly in publications and eCatalogs without additional linking.)

Link the Object Types from the References tab on either a Commercial Type or the References tab on a Product-Override by clicking the **Add Parent** link. A node selector displays a list of legal Object Types for linking.

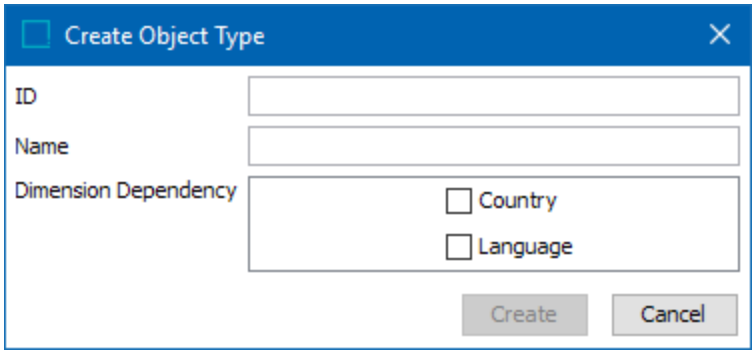


Note: When Commercial Types must be used in Entity Hierarchies, manually link the Commercial Type into the Entity Hierarchy.

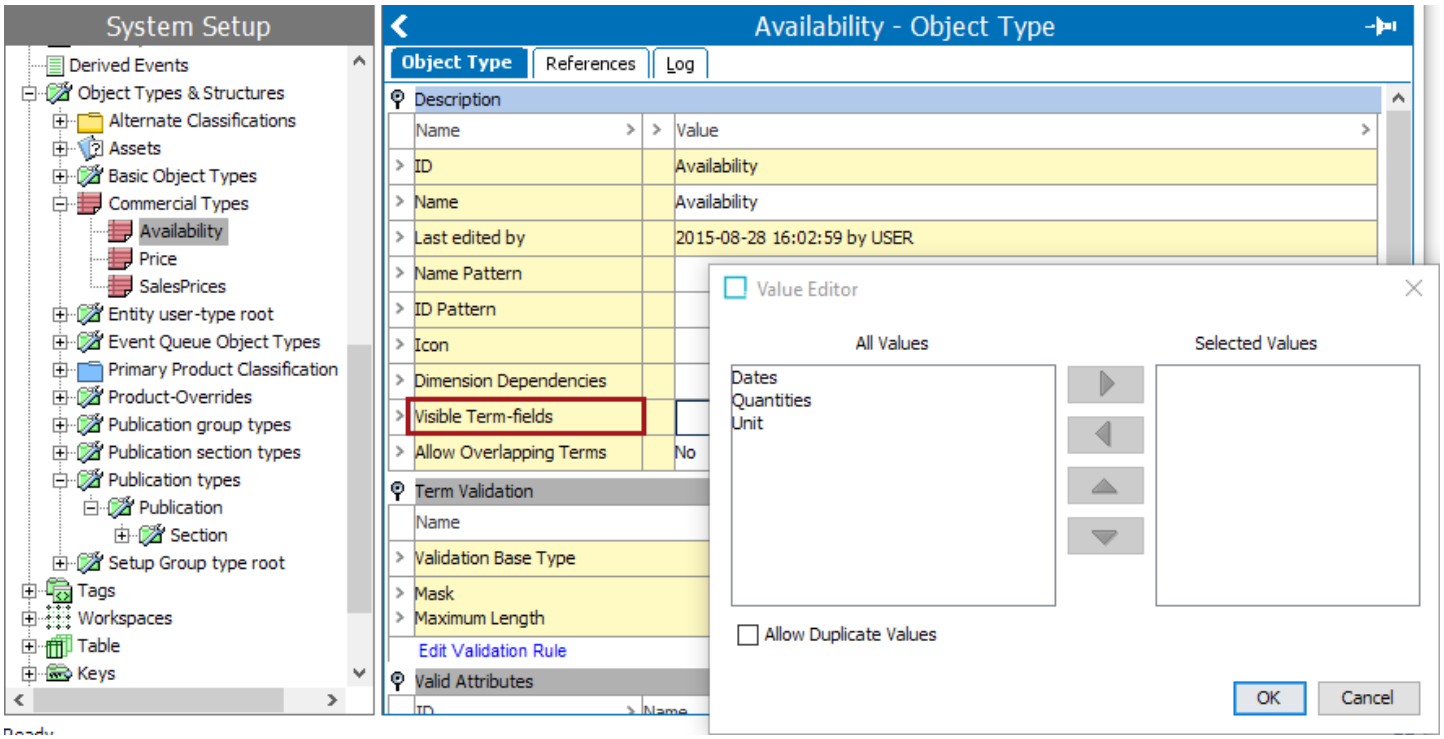
Creating a Commercial List Type

Commercial object types appear in their own hierarchy under Object Types & Structures.

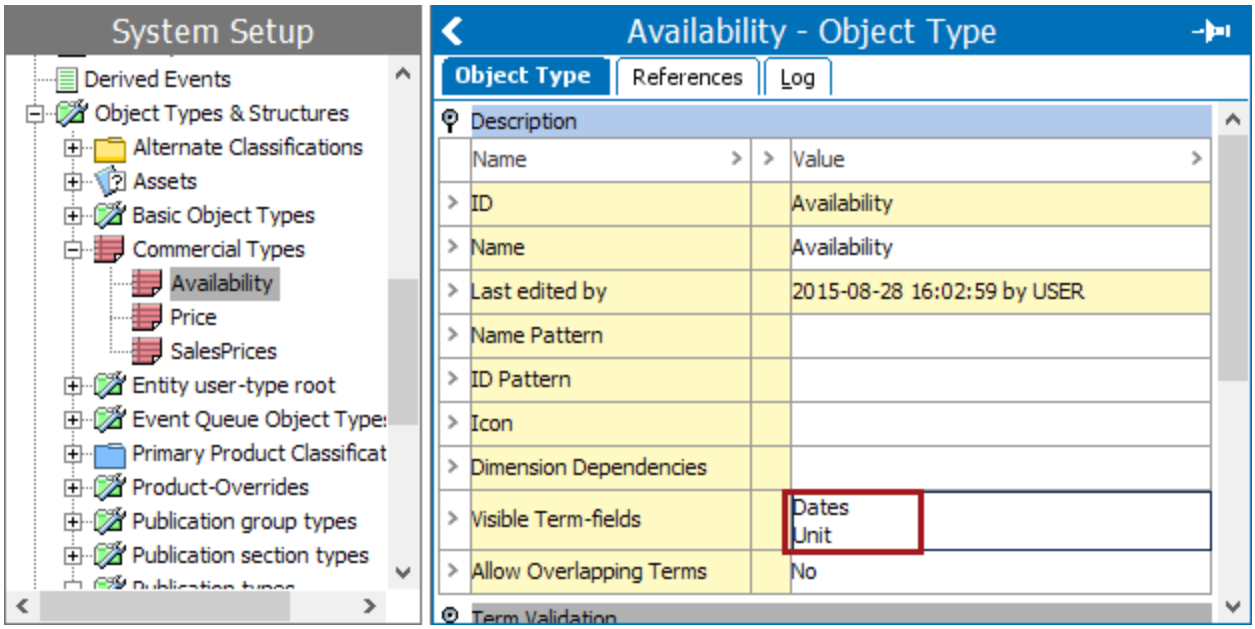
1. In System Setup, open **Object Types & Structures**.
2. Click **Term-list user-type root**, right-click, and select **New Object Type** to display the **Create Object Type** dialog.



3. Type an **ID** and **Name**. Dimension Dependencies are optional, and are not typically used with Commercial List Types.
4. Click **Create** to display the created Commercial List Type dialog.
5. In the **Object Type** tab, double-click in the **Value** column of the **Visible Term-fields** to display the **Value Editor** dialog.



6. 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.

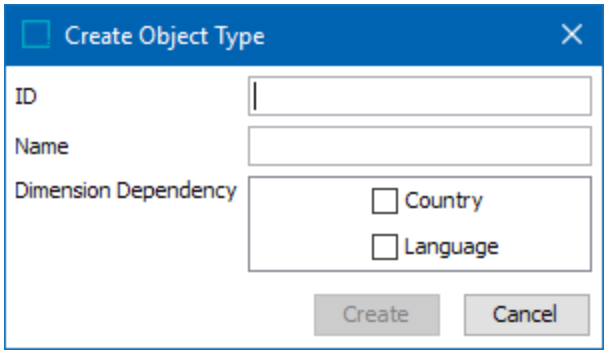


For more information, see the Commercial Data documentation.

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.



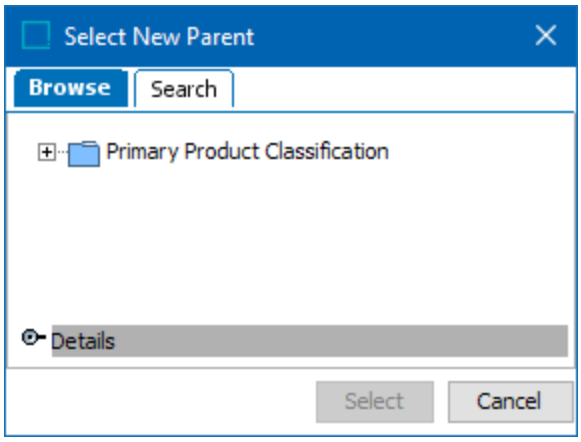
3. Type an **ID** and **Name** and optionally, check Dimension Dependencies.
4. Click **Create** and a new Product-Override Object Type is created.

> **Note:** To enable a Product-Override, additional linking is required. For more information, see the **Enabling a Product Override** section.

Enabling a Product-Override

A Product-Override Object Type must be linked to the folders in the Primary Product Hierarchy on which a Product-Override should be possible.

1. In System Setup, open **Object Types & Structures**.
2. Open **Product-Overrides**, and select the relevant Object Type to display an editor.
3. On the **References** tab click the **Add Parent** link to display the **Select New Parent** dialog.

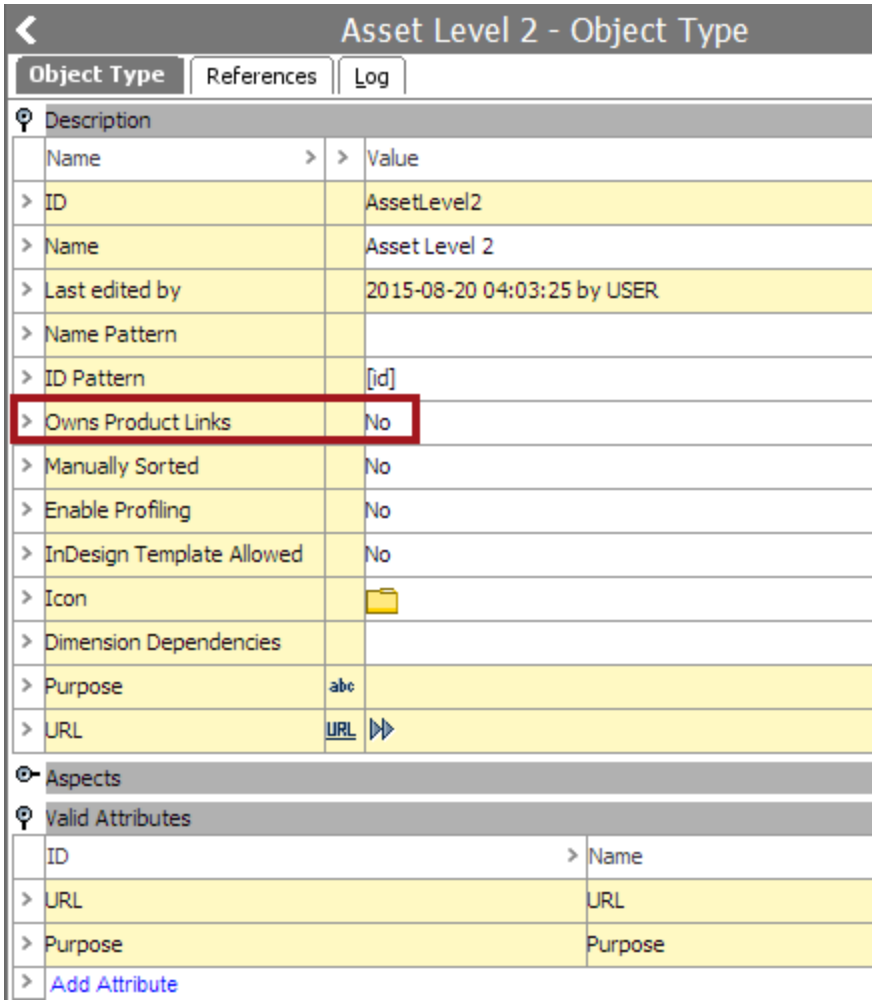


4. Use Search or Browse to select an Object Type under **Product user-type root**.
5. Click **Select** to apply a Product-Override to a Product Object Type.

> **Note:** It may be necessary to define specific settings for each Product-Override Type such as manual attribute sequencing and rules for template changes.

Owns Product Links on Alternate Classifications Object Type

Owns Product Links specifies the ownership of a link from a product to a classification. The Owns Product Links field determines if the link should be unapproved when products are linked to the classification.



The available options for Owns Product Links are:

- **No** means the product 'owns' the link to the classification. When an additional product is linked to the classification the approval status of the classification is not changed. It is not necessary to approve the classification in order to see linked products in the classification in the approved workspace.
- **Yes** means the classification reference 'owns' the link to the product. When an additional product is linked, the classification will become unapproved. It is necessary to approve the classification in order to see linked products in the classification in the 'Approved Workspace.'

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.

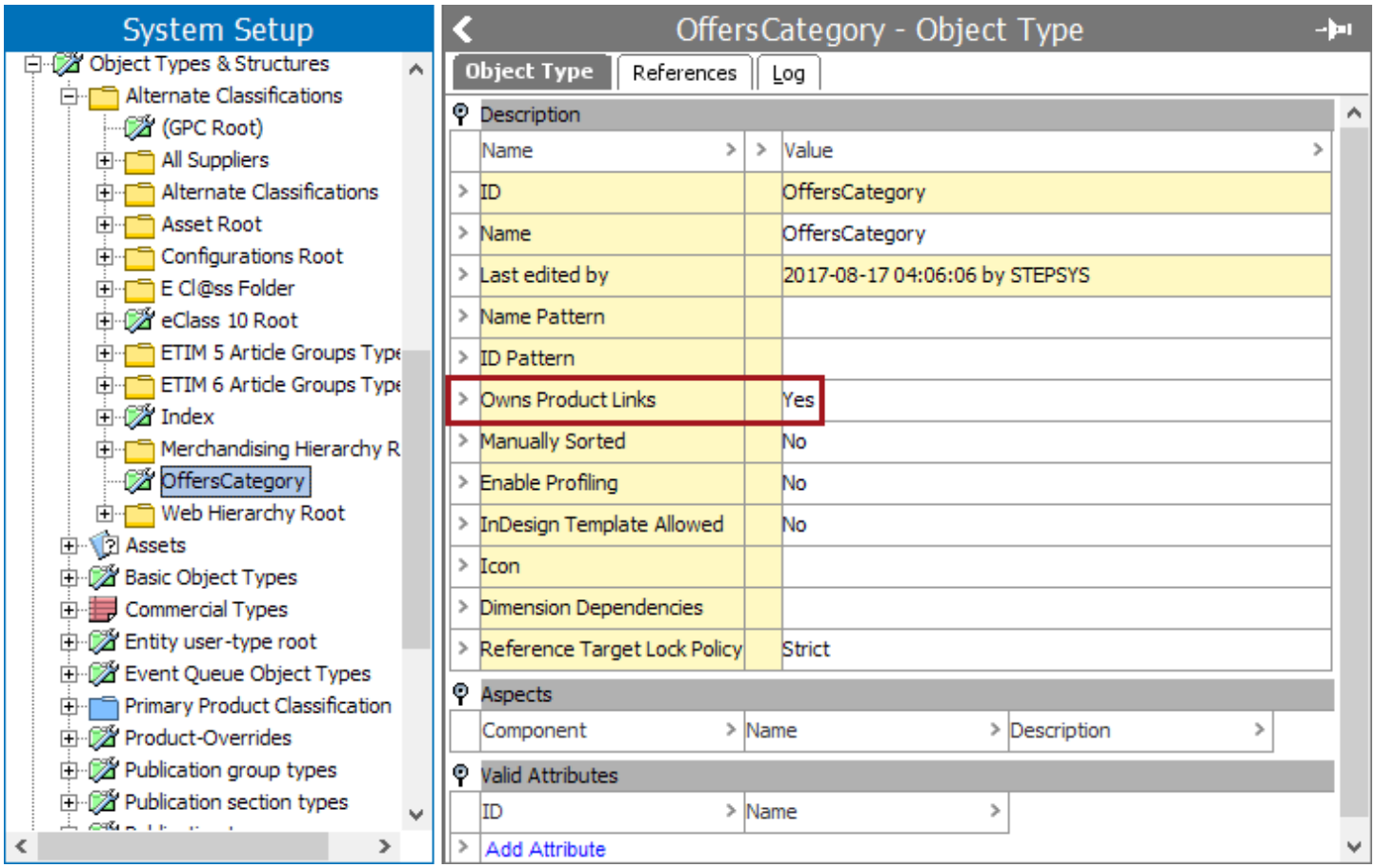
Owns Product Links Affects STEPXML

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.

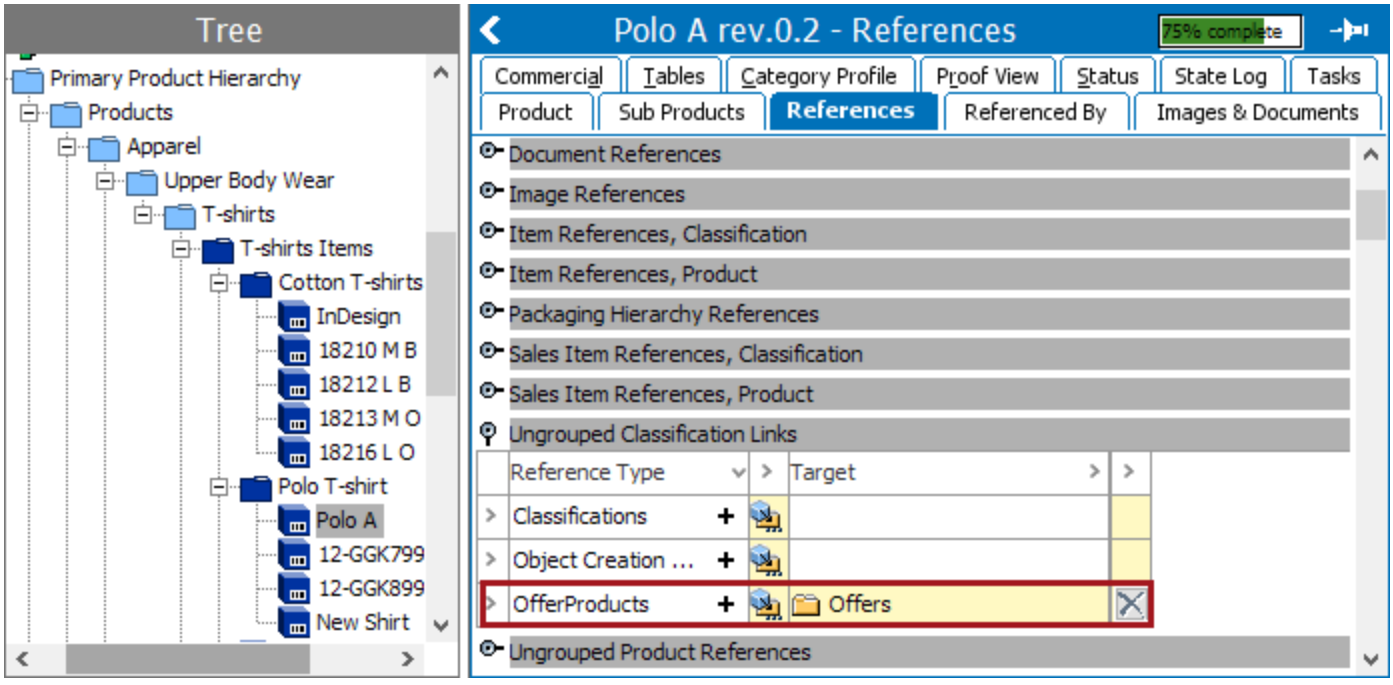
To stop the classification-owned links from being exported under the products, set the recorder option 'IncludeClassificationOwned' on the Products element to false as shown in the red text below:

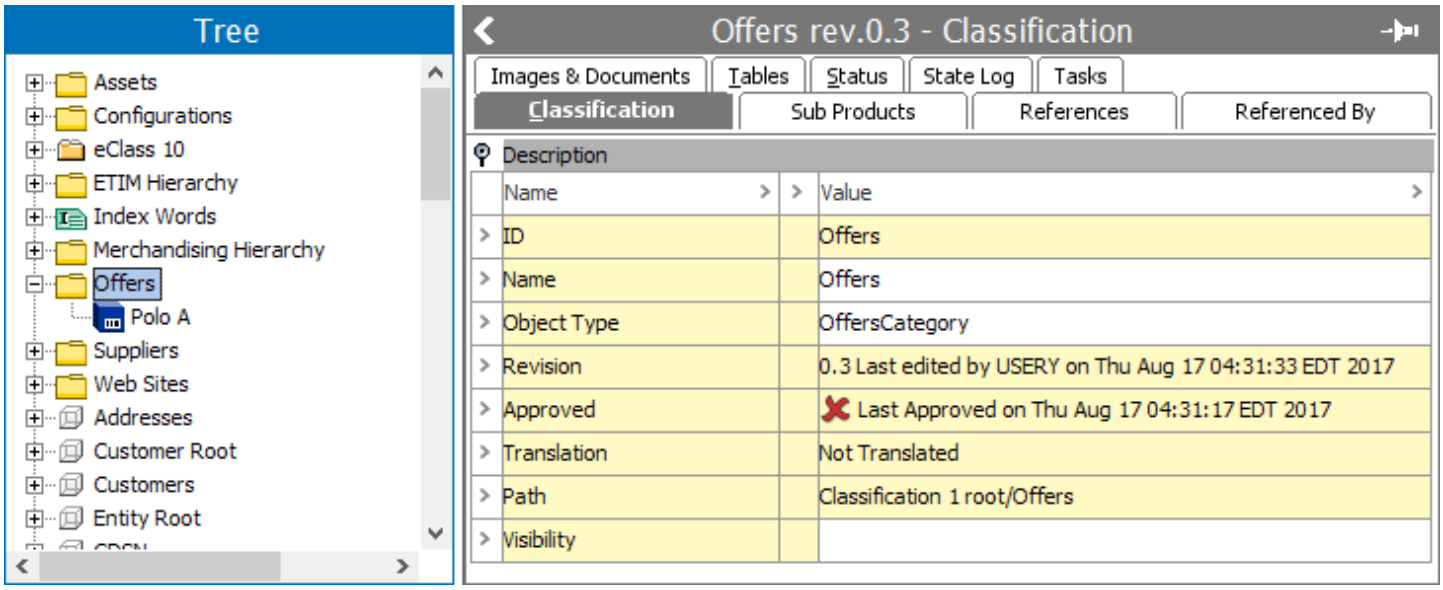
```
<STEP-ProductInformation>
<Products IncludeClassificationOwned="false">
<Product>
</Product>
</Products>
</STEP-ProductInformation>
```

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'.

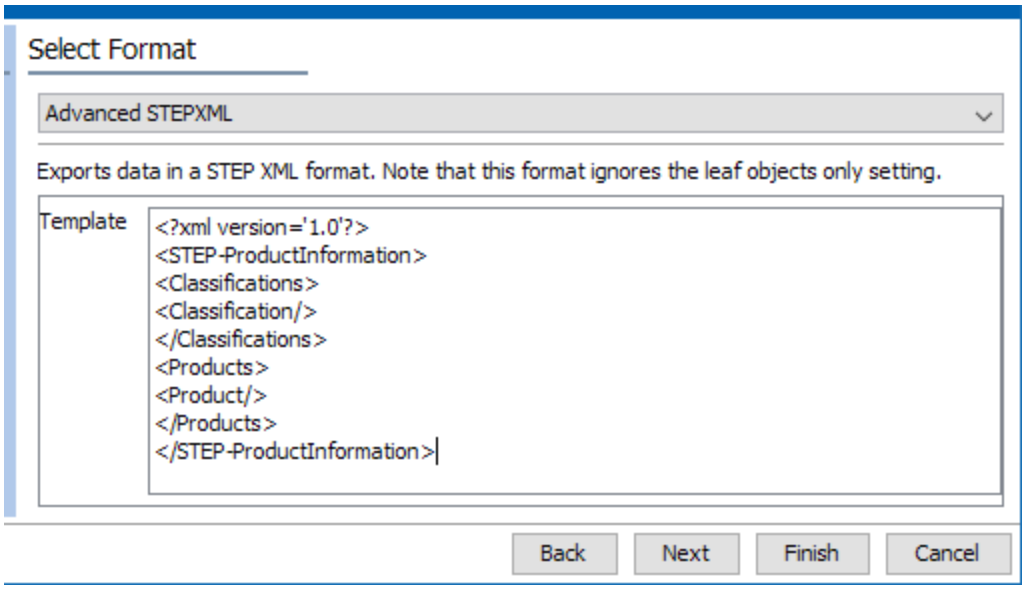


Product 'Polo A' is linked to classification 'Offers' through Classification Link 'OfferProducts.'





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:

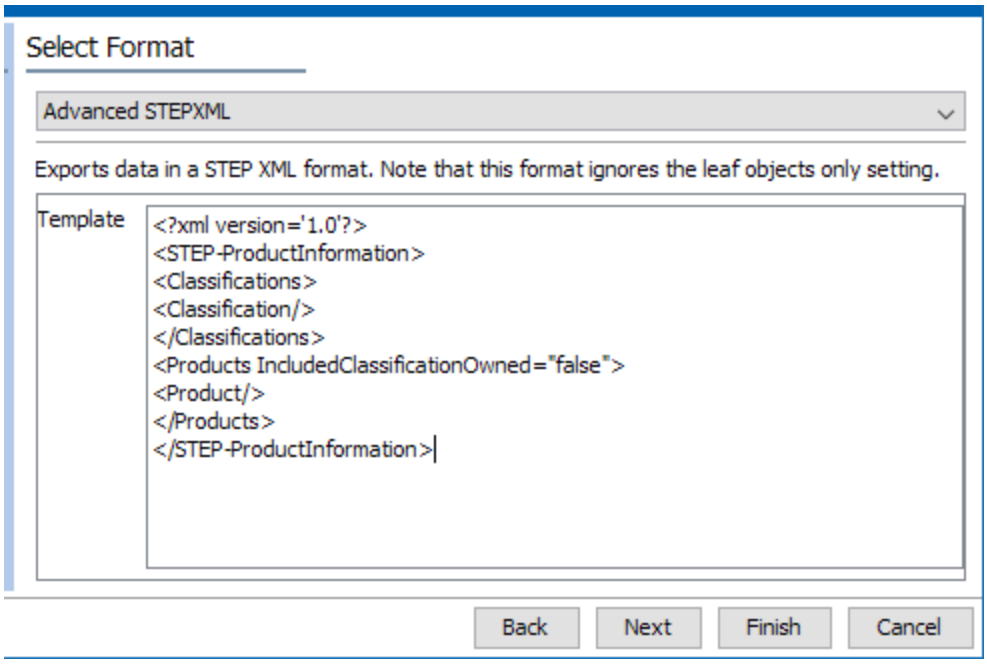


When the user has finished going through the wizard and pressed 'Finished,' part of the complete output will look like the following:

```
  <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"/>
```

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:



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.

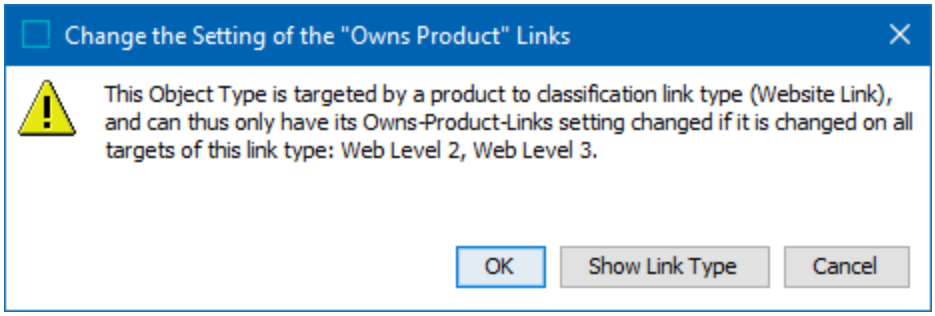
Note: Changing the setting of Owns Product Links can be a time-consuming operation.

Steps to Change the Owns Product Links

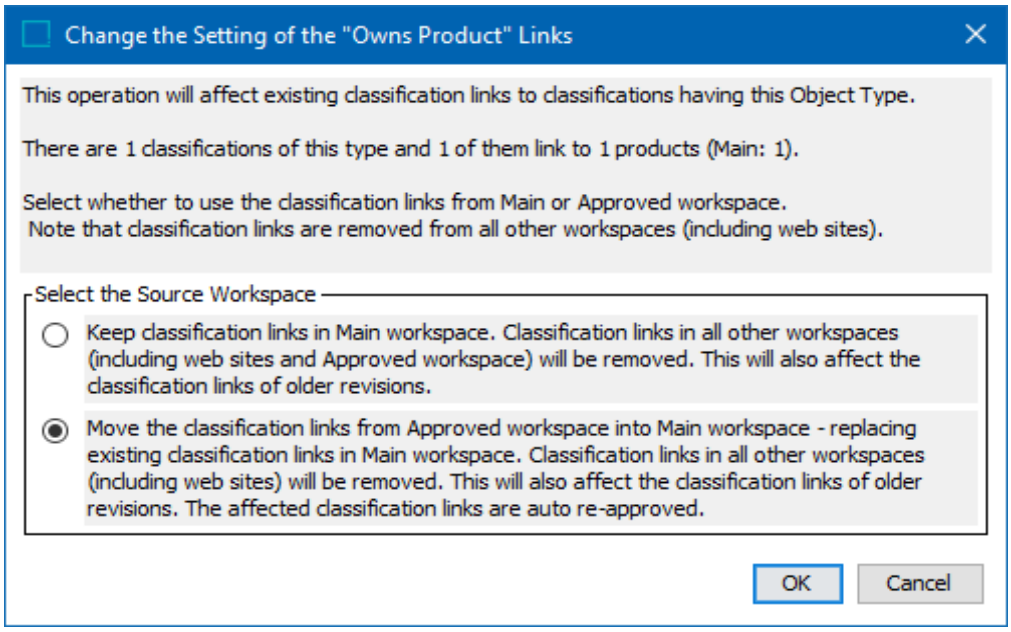
Changing the Ownership for an existing Classification Object Type which is being used as Target in various Classification Link Types can be done by following the steps below:

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.

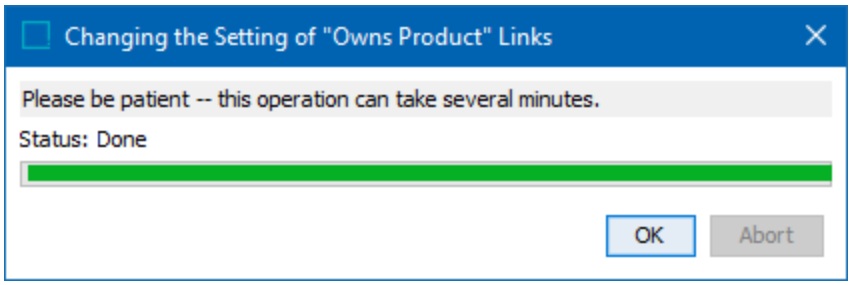


4. Select an option to handle the existing links across the Workspace:

> **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' parameter for an asset, entity, classification, and product object type manages how objects are locked while they are being referenced.

When long-running transactions cannot be simplified easily, consider setting the 'Reference Target Lock Policy' parameter to 'Relaxed' on the object types to which the long transaction applies.

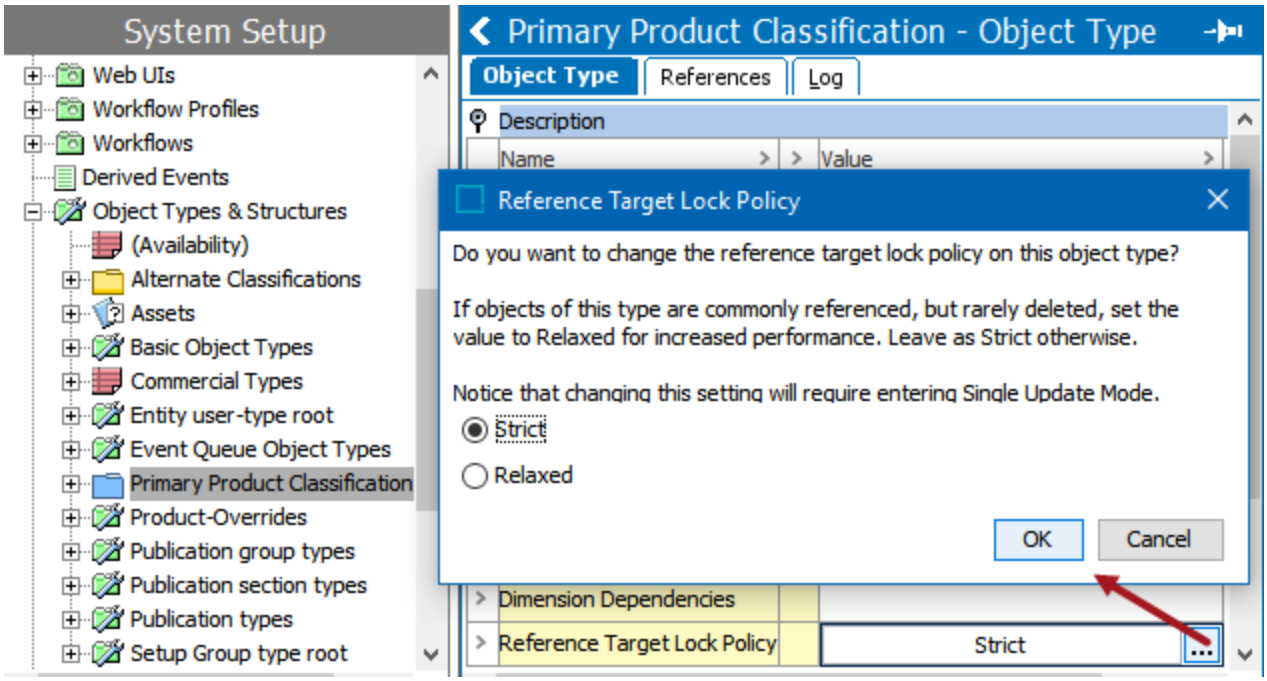
Strict or Relaxed

The 'Strict' setting is default for the 'Reference Target Lock Policy' parameter. When a reference between two objects is being created, the target object is locked to ensure that it is not being deleted while the reference is being created. This means that only one process or user can edit it at a time. Often, deletion of the object is not a deletion from the database, but a deletion due to revision control when the insert of a new history entry occurs.

When using the Optimistic Locking option, a 'Strict' setting can result in an issue 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. Since STEP continues to retry the import, this can negatively impact inbound feeds.

The 'Relaxed' setting uses a less restrictive lock (a shared 'user lock') on the reference target objects being edited so that concurrent updates by multiple processes and/or users are allowed. A full lock is used only when a deletion is attempted. The Relaxed setting allows faster parallel creation of references to the same target object. This setting, when used with object types that are frequently referenced but rarely deleted, improves the performance and stability of parallel inbound imports, bulk updates, and users concurrently creating references to the same objects.

Important: Do not use the 'Relaxed' setting on an object type for objects that are often deleted as this can result in poor performance, and risks locks and deadlocks.

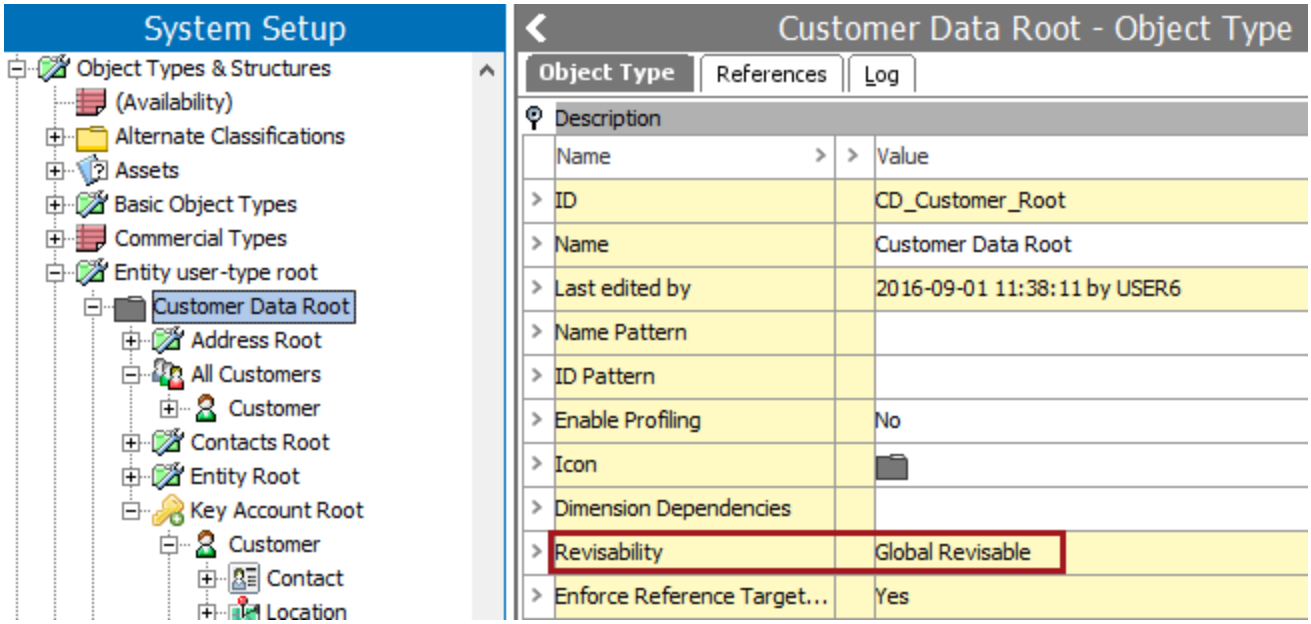


The Relaxed / Strict setting can be changed as needed, but requires the system to enter Single Update Mode for only as long as it takes to update the setting.

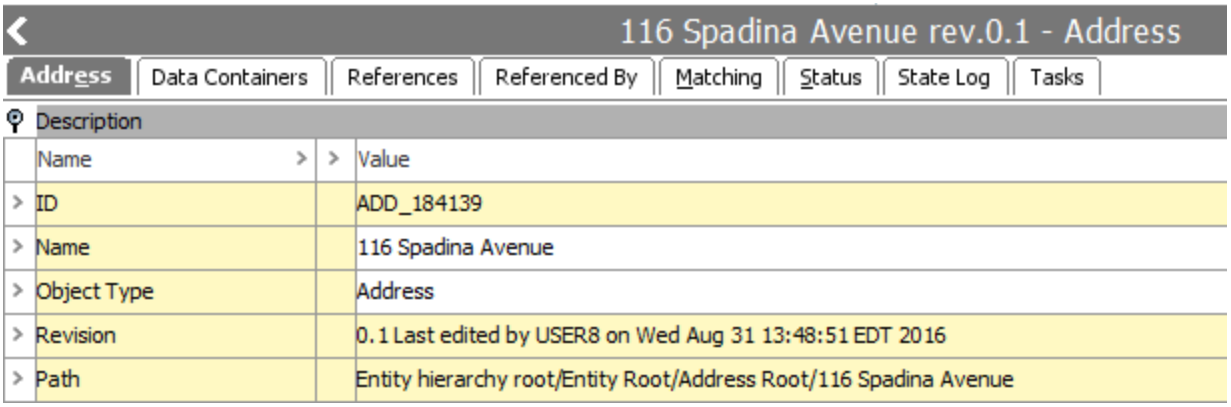
Note: Changing this setting during peak production hours can have a negative impact on performance.

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.

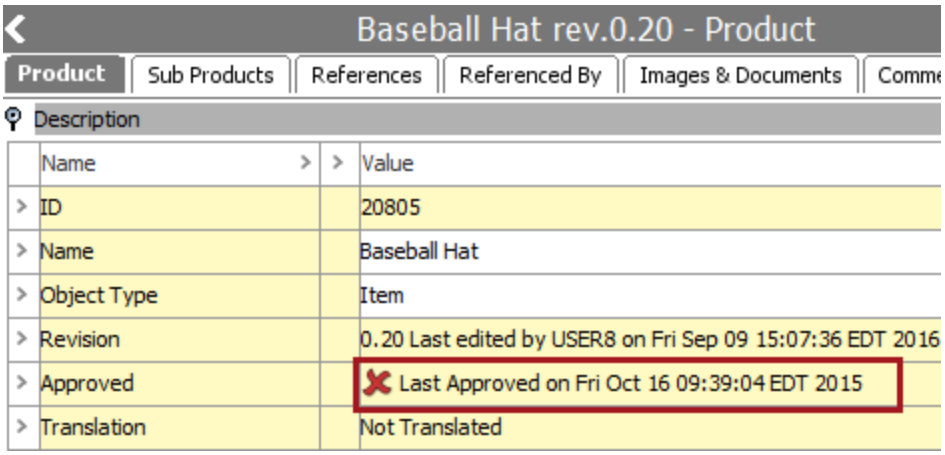


- Global revisable entity values display identically in all workspaces, are not under revision control, and therefore do not require approval.



- 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: Workspace revisable is typically selected when the party data is imported into STEP from an upstream system after being reviewed and approved and does not need to be revisited.



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.

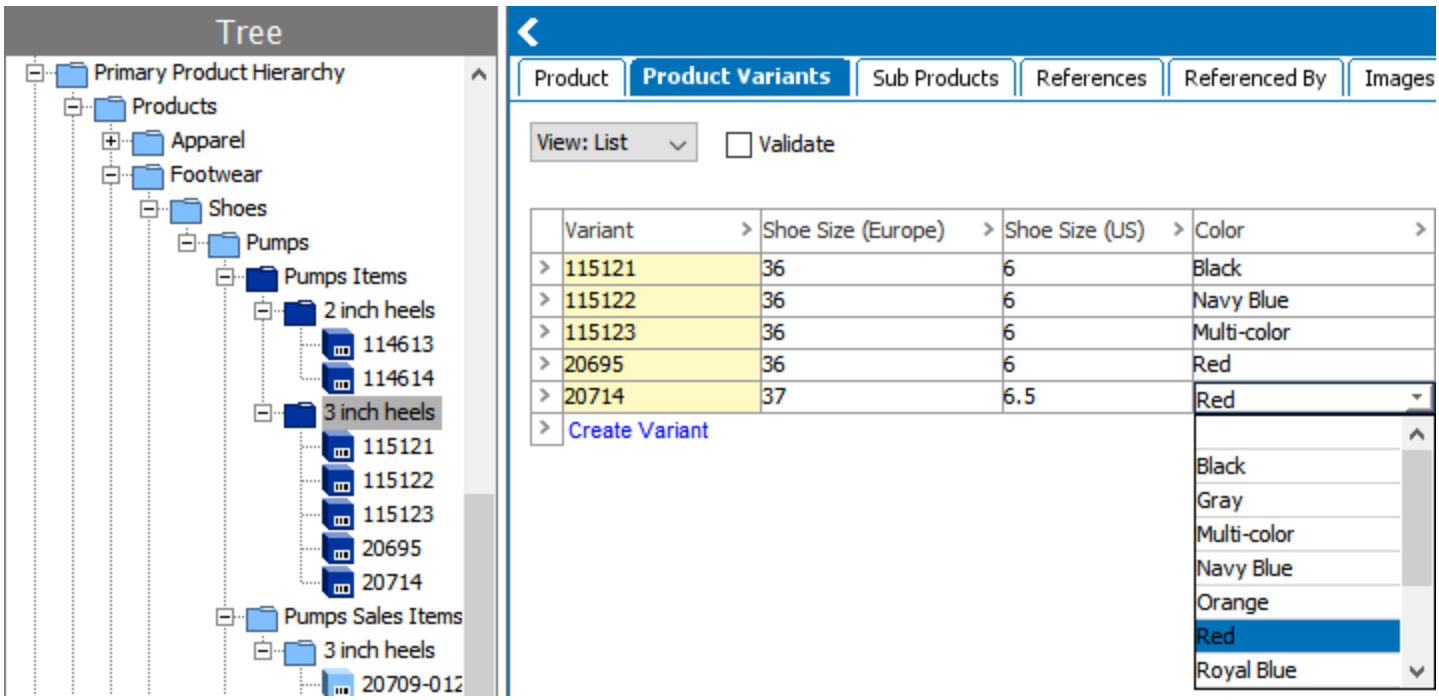
Note: When building an entity object structure, a top entity object type can be global revisable, while child entity object types are workspace revisable. However, a top entity object type that is workspace revisable can only have child entity object types that are also workspace revisable.

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 those 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 hold only the combinations of the color and size varieties (e.g., size 7 red, size 8 blue, etc).

The following image shows this concept.



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.

Important: Any attributes that are considered variant attributes cannot be multivalued.

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:

- Products Variants List in Workbench
- Products Variants List in Web UI
- Product Variants Matrix in Workbench
- Product Variants Matrix in Web UI
- Business Condition: Validate Product Variant in the Business Rules documentation

Setting Up Product Variants

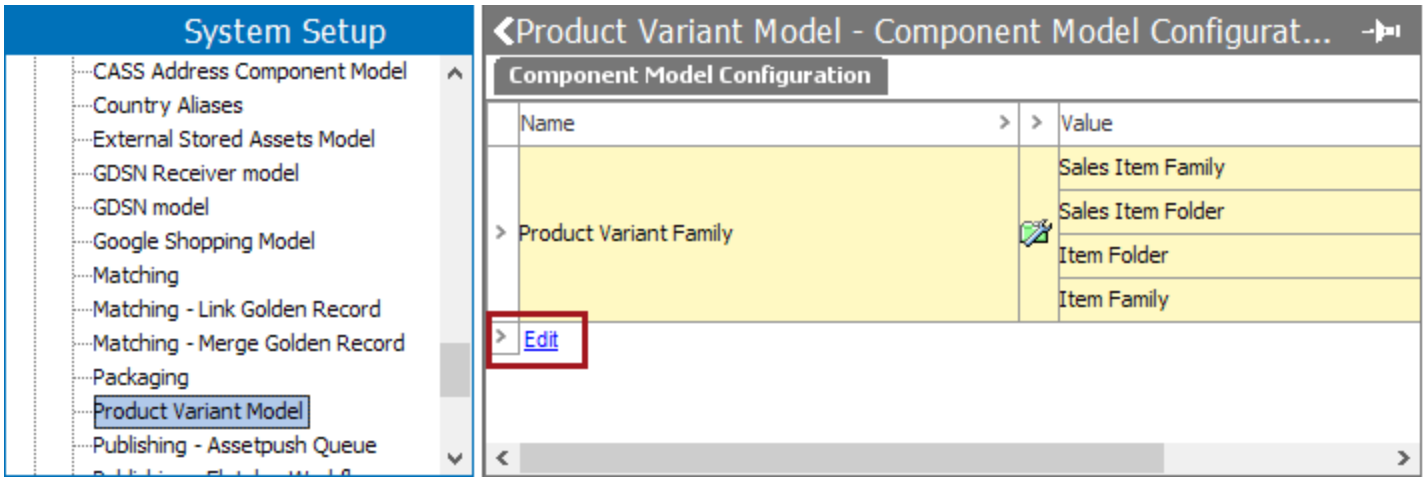
To set up product variants, complete the following steps as defined in the sections below:

- Add the object types to the component model.
- Create product variant priority attribute.
- Specify variant and non-variant attributes.

Add the Object Types to the Component Model

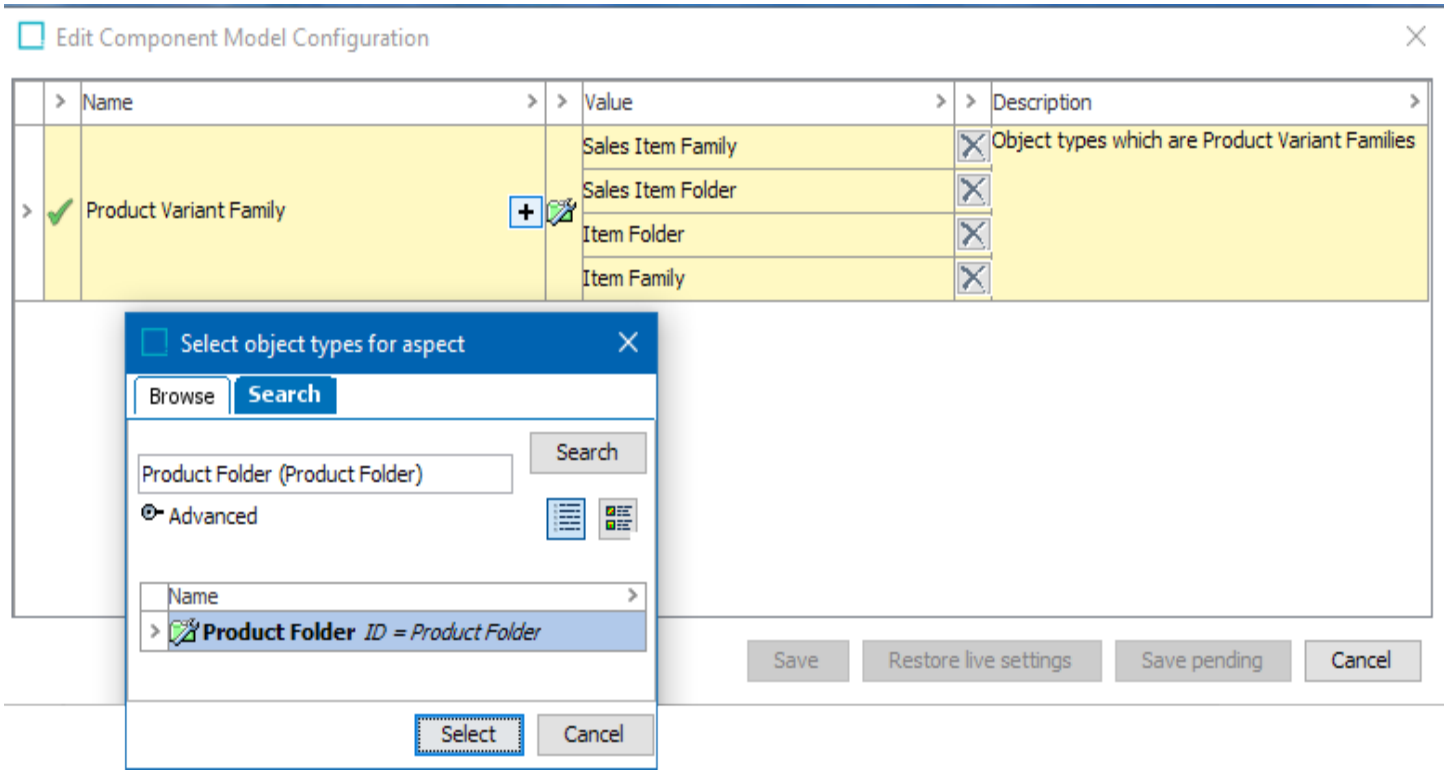
The object type of the parent folder must be added to the component model to enable the product variant capabilities and display the Product Variants tab in the product editor on Tree.

1. In System Setup, expand the Component Models node, and select **Product Variant Model**.
2. On the 'Component Model Configuration' tab, click the **Edit** link.

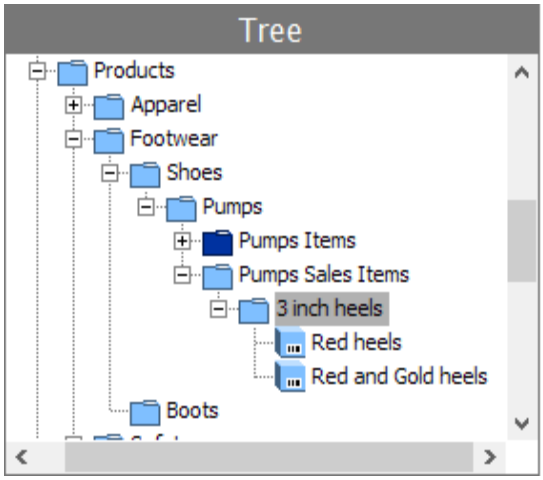


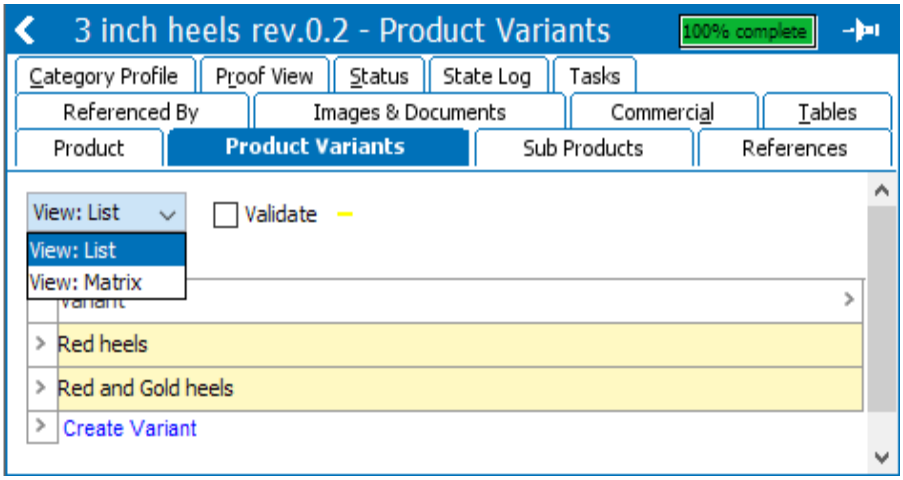
3. In the 'Edit Component Model Configuration' dialog, double-click the plus sign (), choose the object type(s) of the parent folder(s) and click the **Select** button. The parent object types selected are the levels that will house individual items and be used to display the actual item object types.

   When all object types have been added, click **Save** to create a 'Product Variants' tab on all products of the selected object types.



On Tree, select an object that has been added within the Product Variant Model component model and click the 'Product Variants' tab. As shown below, the **List** view and the **Matrix** view are available, but no attributes are displayed until a variant priority attribute is created as defined in the next section.

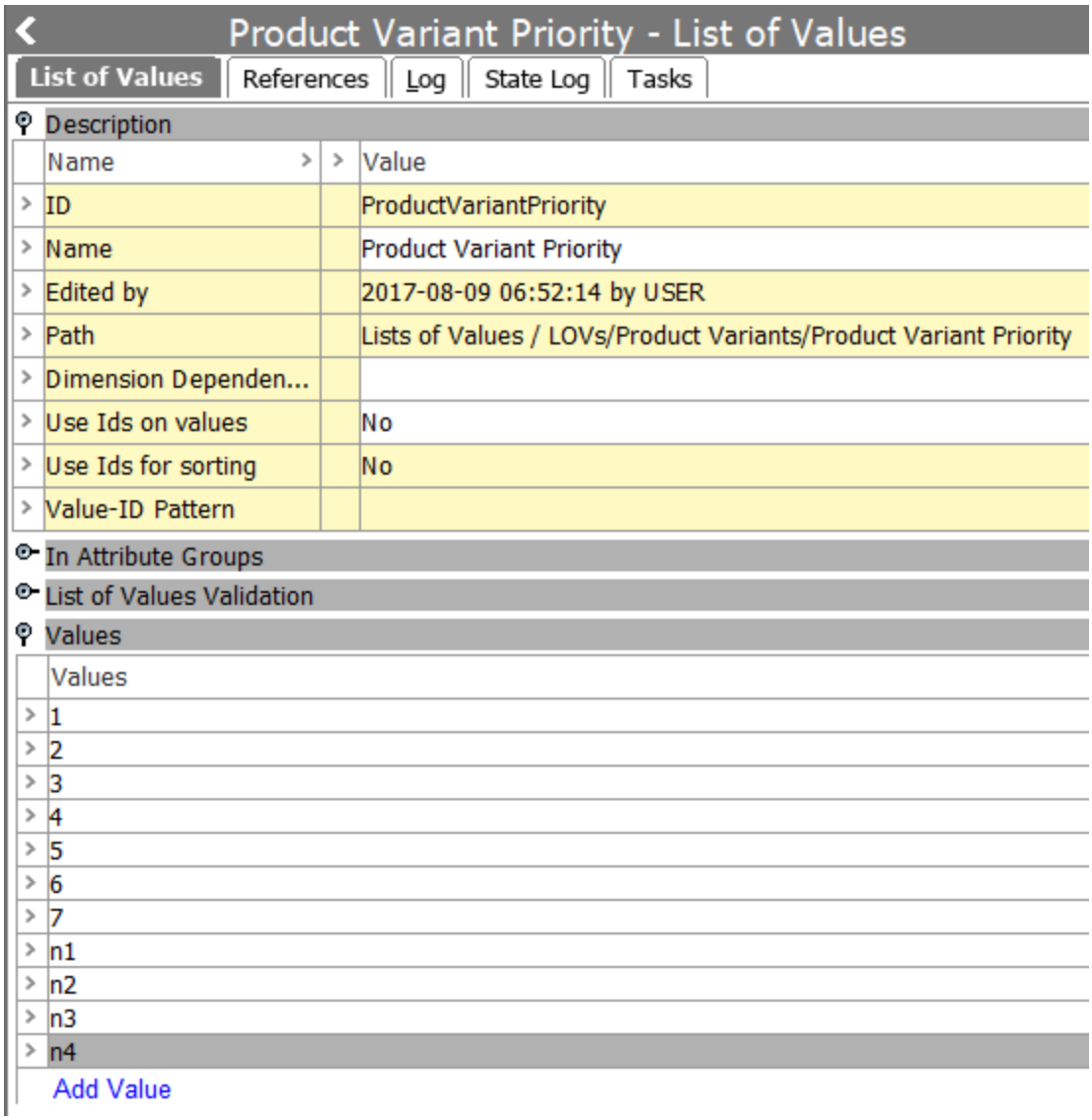




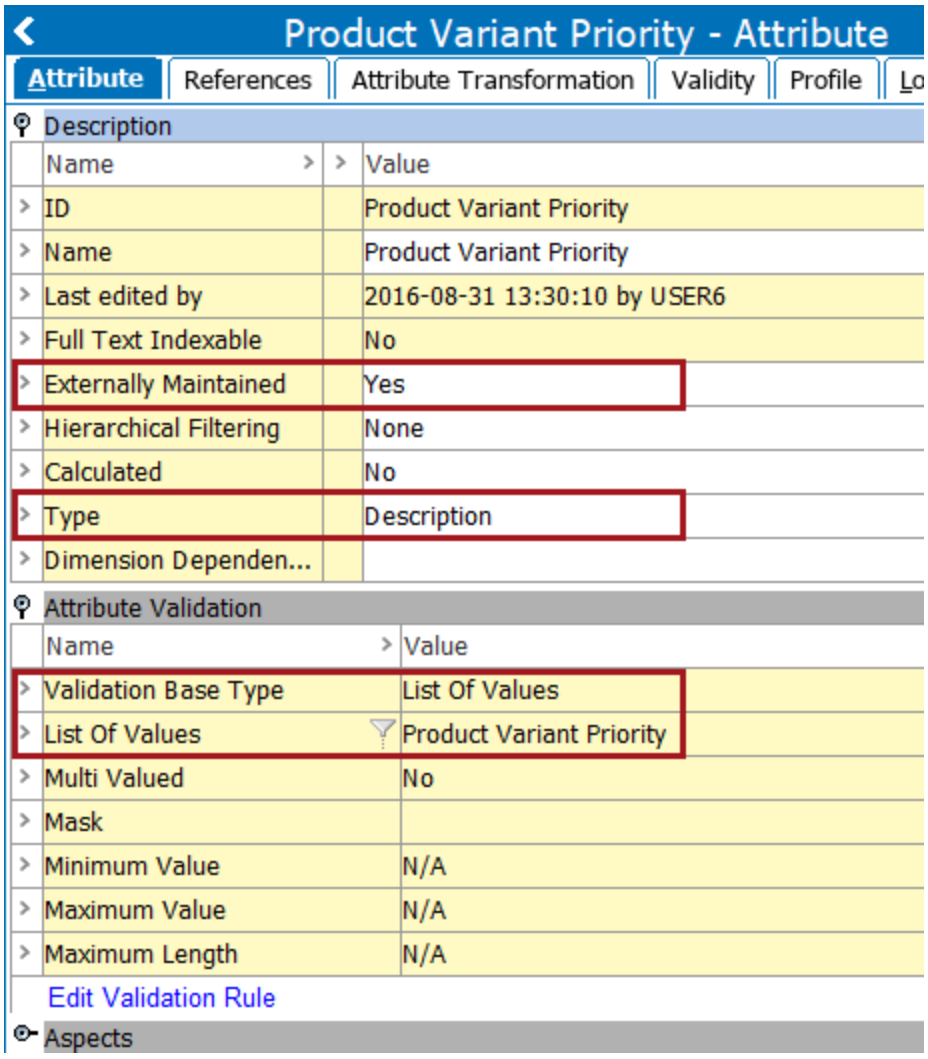
Create a Product Variant Priority Attribute

The priority metadata attribute lives on the link from the parent to the variant and non-variant attributes and allows you to define the order of the attributes.

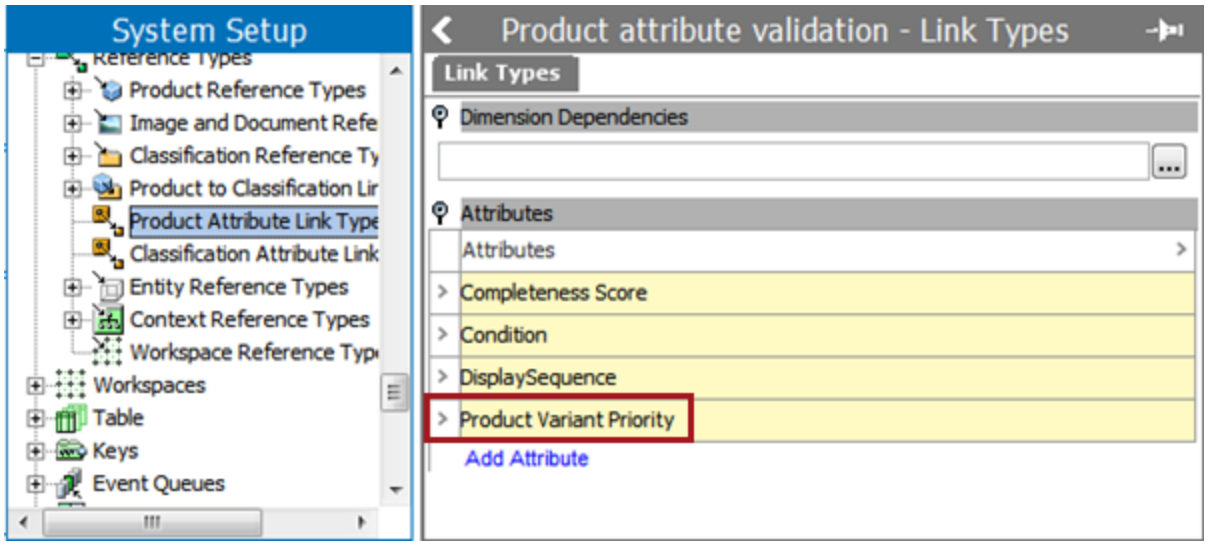
1. Create an LOV with numbers and 'n'-prefixed number values.



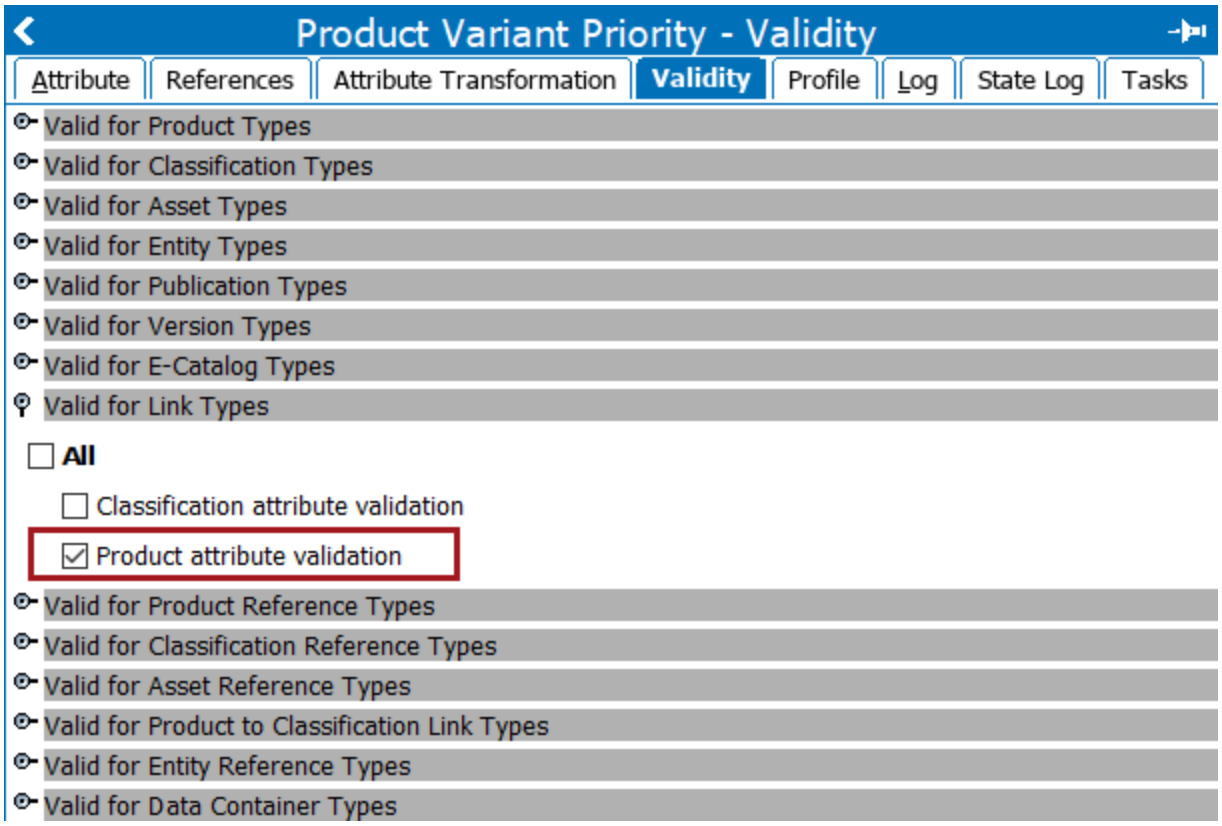
2. Create an externally maintained description attribute, choose List Of Values as the validation base type, and assign the LOV created above.



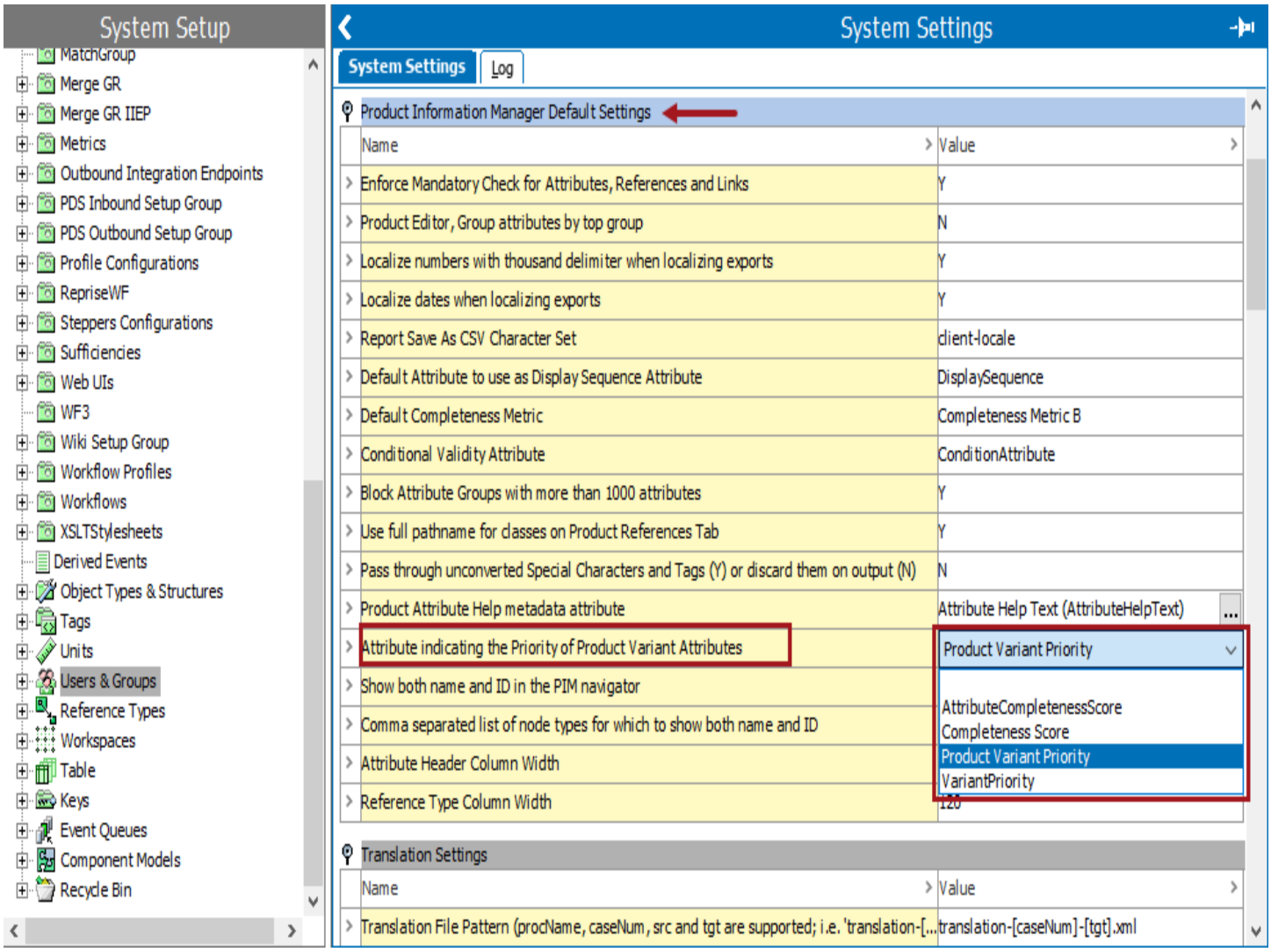
3. In System Setup, expand the Reference Types node, select **Product Attribute Link Type**, and then add the attribute created above. This makes the priority attribute available on the valid objects within the References tab under the Linked Attributes from Product Hierarchy flipper.



This can also be achieved from the attribute Validity tab under the 'Valid for Link Types' flipper by selecting the 'Product attribute validation' checkbox.



4. In System Setup, select the Users & Groups node, open the 'Product Information Manager Default Settings' flipper, and select the new attribute from the **Attribute indicating the Priority of Product Variant Attributes** dropdown.



Specify Variant and Non-variant Attributes

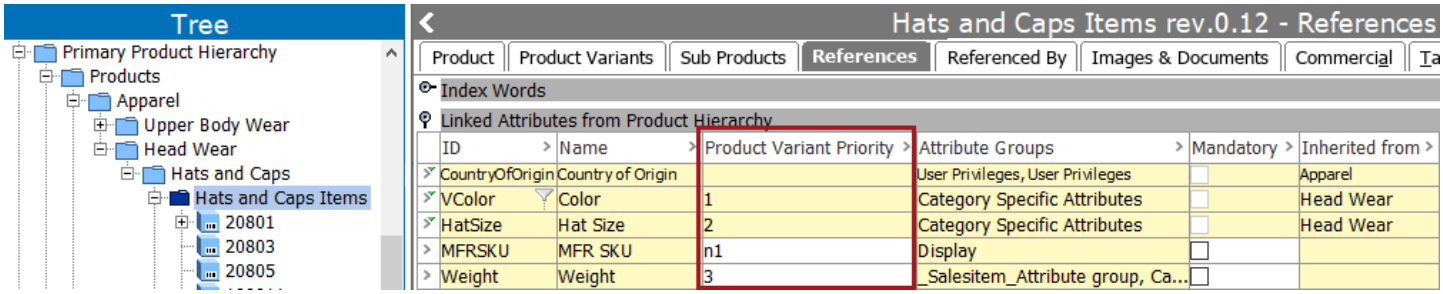
To specify which values are considered to determine variants, you use the variant priority metadata attribute created above. The metadata attribute must be valid on the link from the parent folder to the variant attributes. This way, each linked attribute can be attached with a number indicating that the attribute is a variant attribute and specifying where in the related views the attribute is shown.

Attributes linked from a product variant family are categorized into three groups:

- **Variant attributes:** A variant priority attribute value is a number: 1, 2, 3, ...
- **Non-variant attributes:** The variant priority attribute value is a number prefixed with the text "n": n1, n2, n3, ...
- **Regular attributes:** The variant priority attribute is empty.

Important: Product variant attributes cannot be multivalued.

In the example below, the variant priority attribute on the link from 'Hats and Caps Items' to Color is 1, from 'Hats and Caps Items' to Hat Size is 2, and from 'Hats and Caps Items' to Weight is 3. This indicates that Color, Hat Size, and Weight are **variant attributes** and that they will display in that order.



As created in the previous step, the variant priority attribute value on a linked attribute can also be a number prefixed with an 'n' (e.g., 'n1' or 'n2'). This specifies that the linked attribute is a **non-variant attribute** and is visible in the views related to variant attributes. However, for validation, deduplication, and variant creation, it is not considered a variant attribute. In the example above (which is also continued below), the link from 'Hats and Caps Items' to MFR SKU is 'n1', which indicates that the SKU is shown in the views, but it is not used to distinguish variants.

The **regular attribute** Country of Origin is linked from 'Hats and Caps Items' but has no value for the variant priority attribute, which indicates that is not used for variants.

Note: Set the variant priority as high in the hierarchy as possible to reduce the setup and maintenance effort.

Choose a method to identify the variant and non-variant attributes.

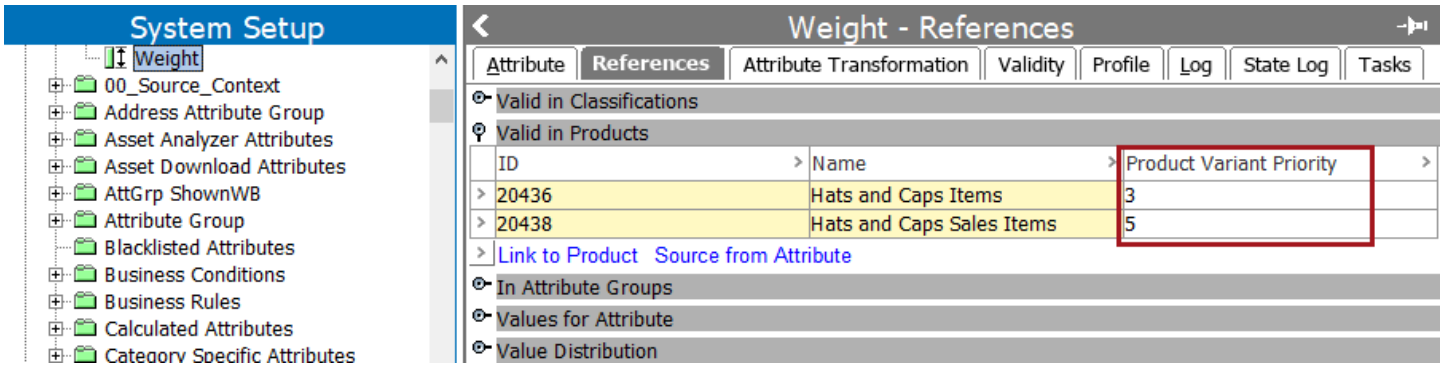
To see the priorities for all attributes that impact variants on the family (as shown in the image above):

1. In the Tree, expand the 'Primary Product Hierarchy' and select the relevant product folder configured for variants.
2. Click the **References** tab.
3. In the 'Linked Attributes from Product Hierarchy' flipper, specify the display priority of the linked attributes using the attribute created previously.

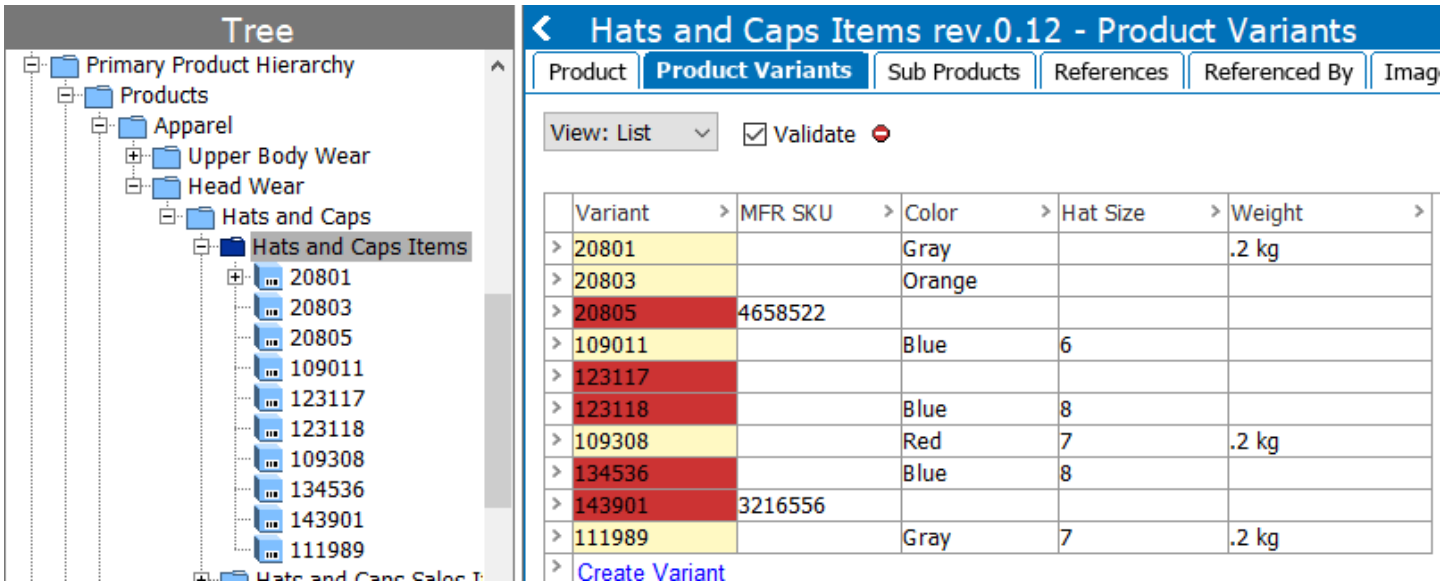
Or

To see the priorities based on attribute, which may include multiple variant families (as shown in the image below):

1. In System Setup, select an variant attribute.
2. Click the **References** tab and open the 'Valid in Products' flipper.
3. Assign the priority for each product family using the Product Variant Priority attribute.



In the following image, the results of the Product Variant Priority settings above are displayed on the Product Variant tab using the list view.



See the following topics for information on viewing variants as a list or a matrix:

- Products Variants List in Workbench
- Products Variants List in Web UI
- Product Variants Matrix in Workbench
- Product Variants Matrix in Web UI

Products Variants List in Workbench

The List view is an overview of all variant products and the variant attribute values. This view also allows you to create and delete variants and modify variant values.

In the list view, one product variant ID is displayed on a row, and the value for the variant attributes are shown in the columns. After validating the data, potential duplicate product IDs are displayed with a red background, and hover text shows all product variant IDs that match the selected attributes.

Note: Product variant attributes cannot be multivalued.

For a more advanced view, use the matrix view as defined in the **Product Variants Matrix in Workbench** topic.

To view variant information in the Web UI, see the **Products Variants List in Web UI** topic or the **Product Variants Matrix in Web UI** topic.

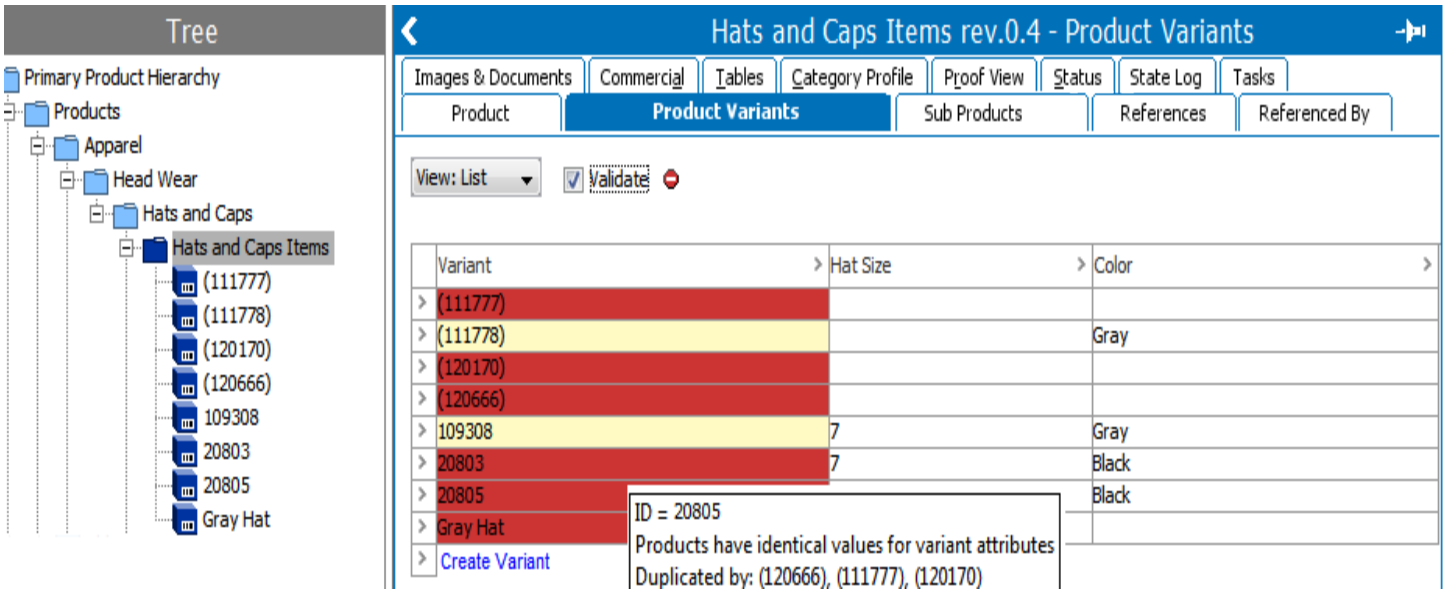
Prerequisite

Before using a product variant view, you must complete the steps in the **Setting Up Product Variants** topic.

Use the List View

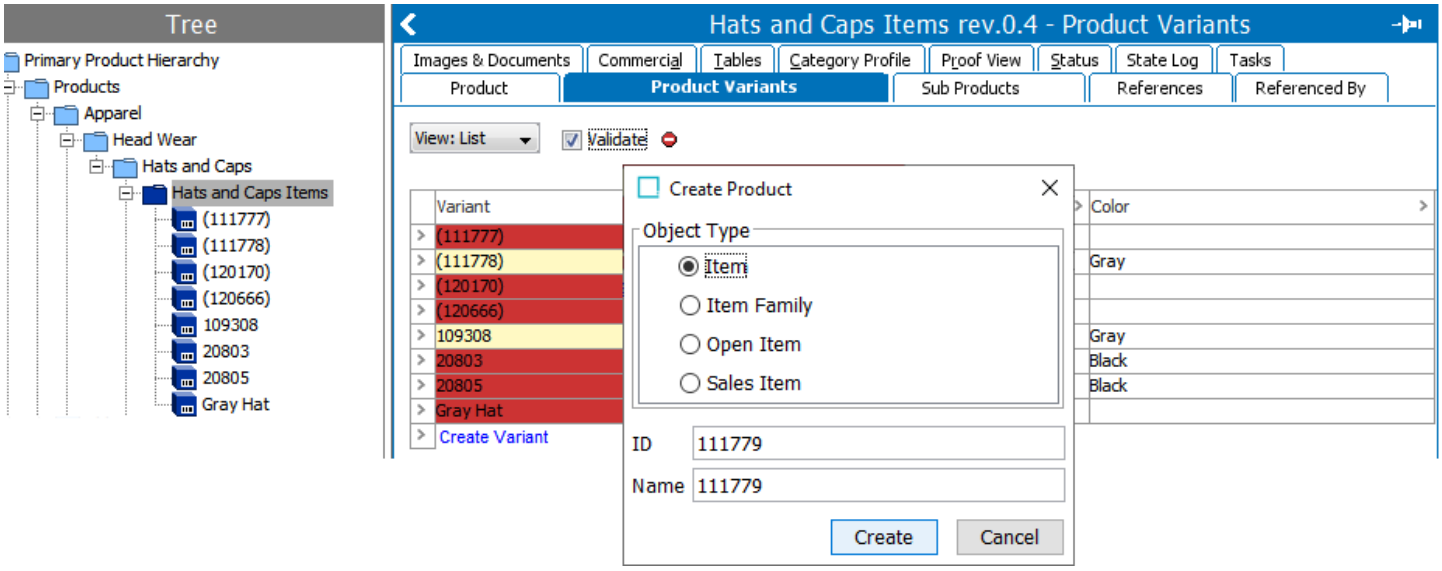
1. In the Tree, select a relevant product parent node that is configured for variants.
2. Click the Product Variants tab, and select **View: List** from the dropdown.
3. Check **Validate** checkbox to view the following information:
   - The possible duplicate variants highlighted in red indicate that at least one other ID has the same values for the variant attributes, including empty values.
   - Hover over a product variant ID to display the ID as a hyperlink. Click the link to display the editor for the product.
   - Hover over a product variant ID to display information about its variant values, non-variant values, and possible duplicates.

Important: While the Validate box is checked, updates are made to the display and hover text as data is changed. Once the checkbox is cleared, no additional updates are displayed.



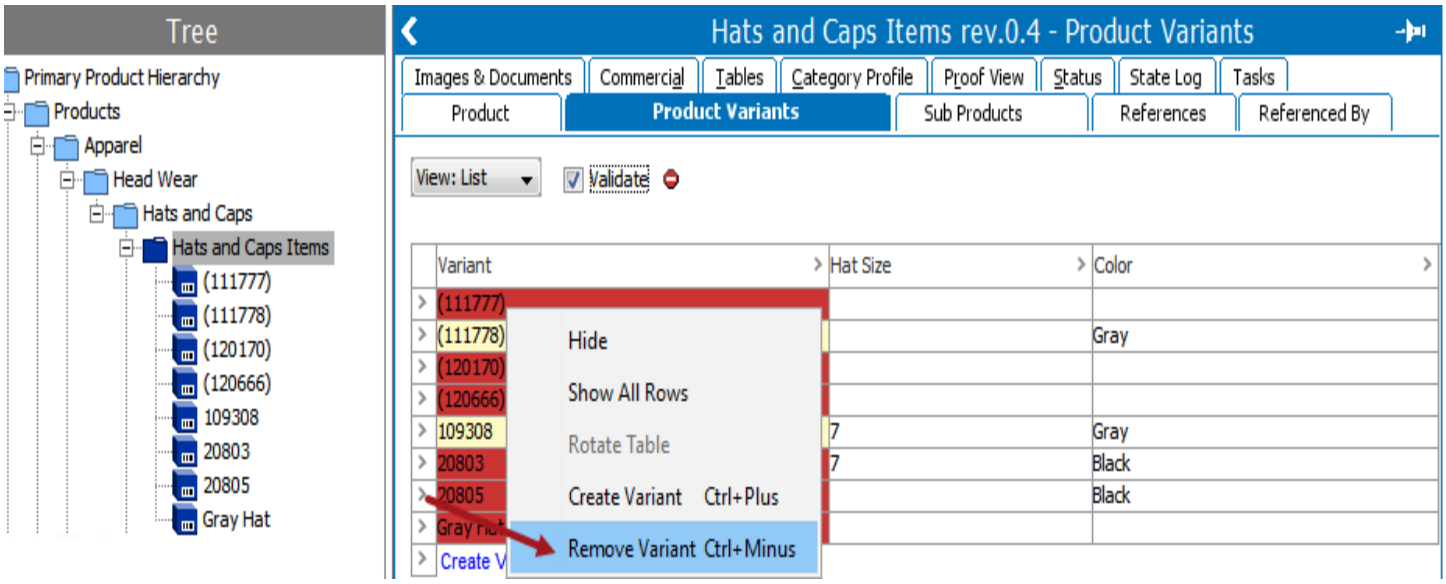
Create a Variant

To manually create a variant, click the Create Variant link, and then complete the parameters on the Create Product dialog. The typical way to create variants is to import them to the parent node.



Delete a Variant

To delete a variant, click the arrow to the left of the row, and then choose **Remove Variant**. A confirmation dialog is displayed and the user can delete the variant or cancel the process.


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 & Documents
Commercial
Tables
Category Profile
Proof View
Status
State Log
Tasks
Product
Product Variants
Sub Products
References
Referenced By
View: List
Validate
Variant
Hat Size
Color
(111777)
(111778)
(120170)
(120666)
109308
20803
20805
Gray
Black
Black
Hide
Show All Rows
Rotate Table
Create Variant Ctrl+Plus
Remove Variant Ctrl+Minus

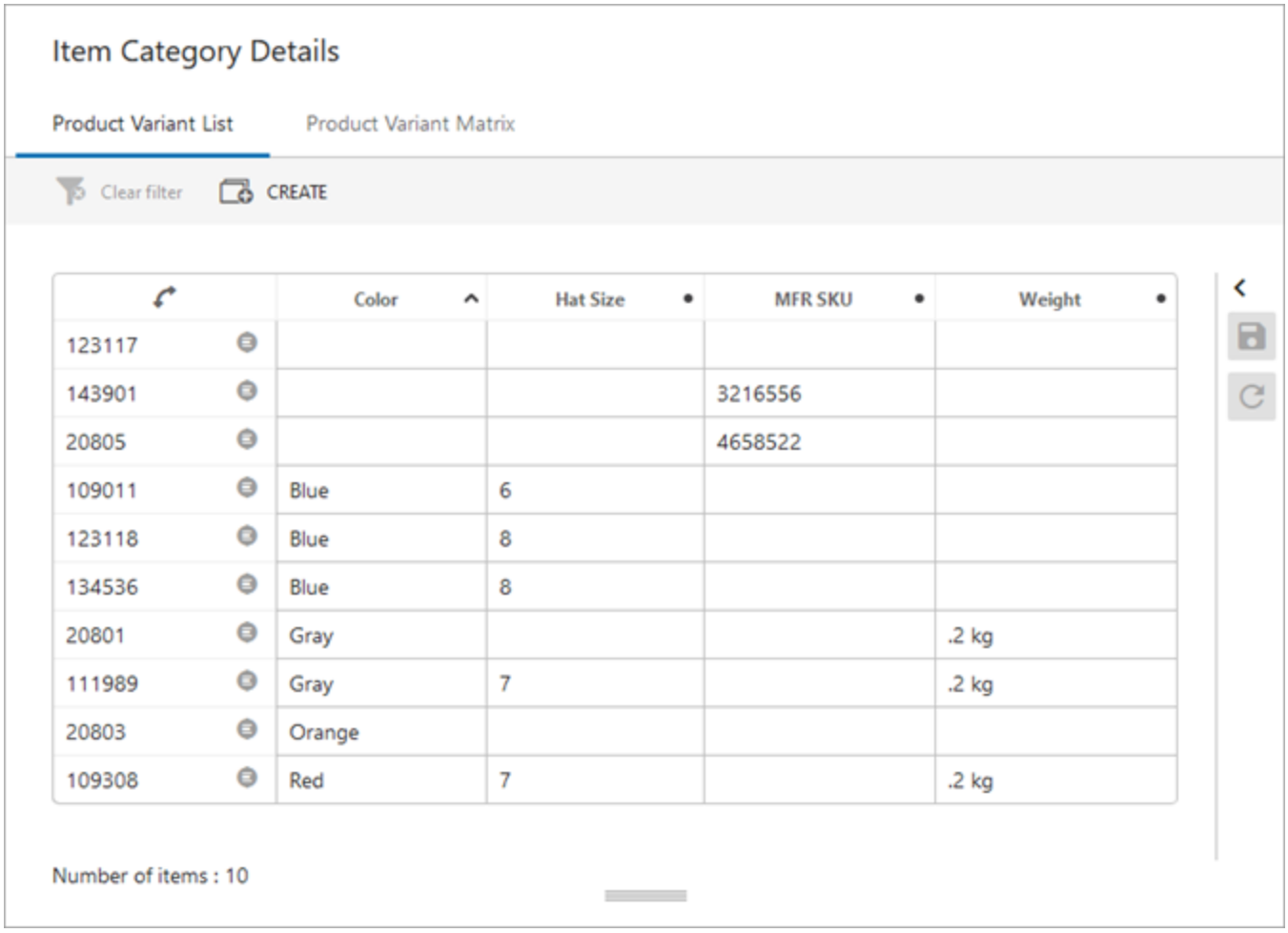
Products Variants List in Web UI

The List view is an overview of all variant products and the variant attribute values. This view also allows you to create and delete variants and modify variant values.

Note: Product variant attributes cannot be multivalued.

In the list view, one product variant ID is displayed on a row and the value for the variant attributes are shown in the columns.

The configured Product Variants List tab page is shown in the following image:



For a more advanced view including the ability to display the duplicates, use the matrix view as defined in the **Product Variants Matrix in Web UI** topic.

To view variant information in the workbench, see the **Products Variants List in Workbench** topic or the **Product Variants Matrix in Workbench** topic.

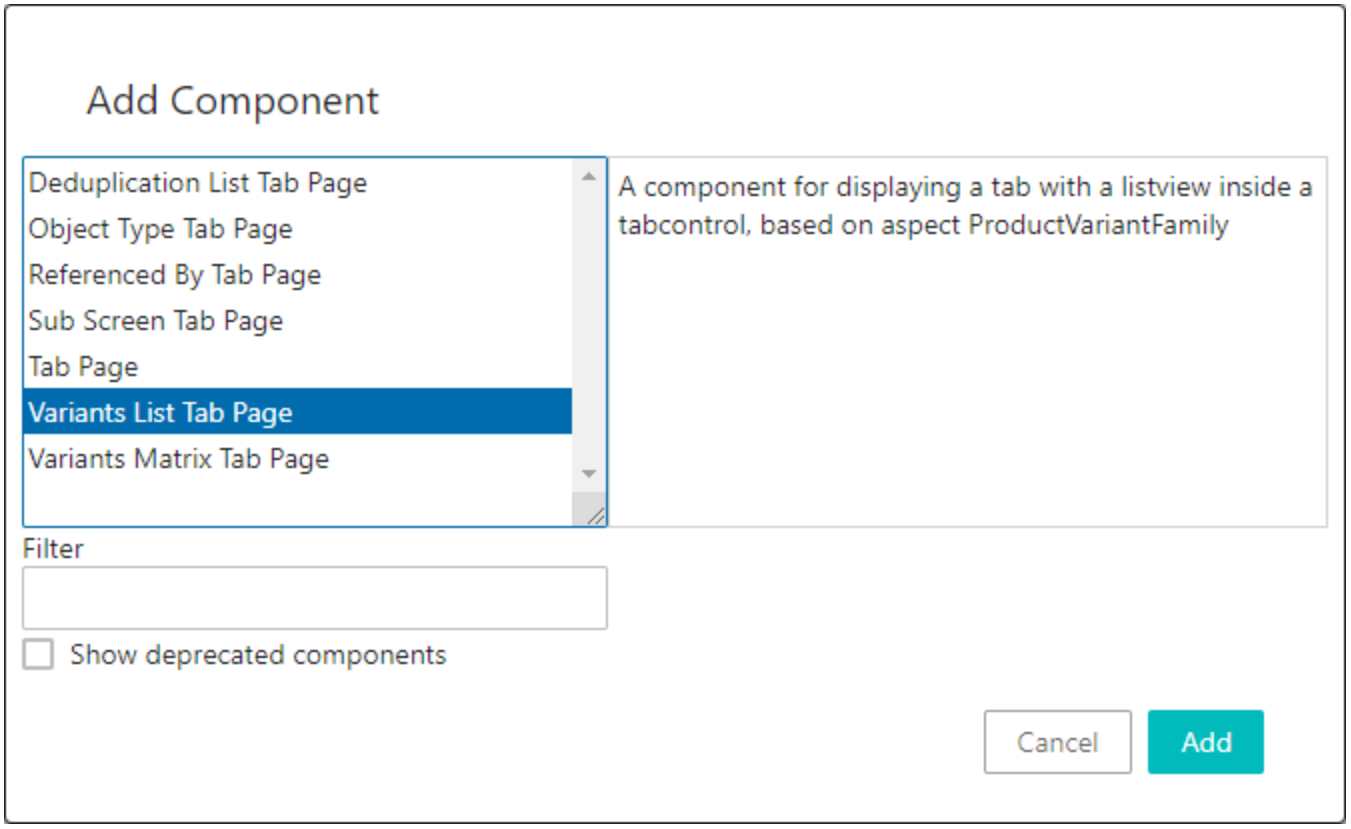
Prerequisite

Before using a product variant view, you must complete the steps in the **Setting Up Product Variants** topic.

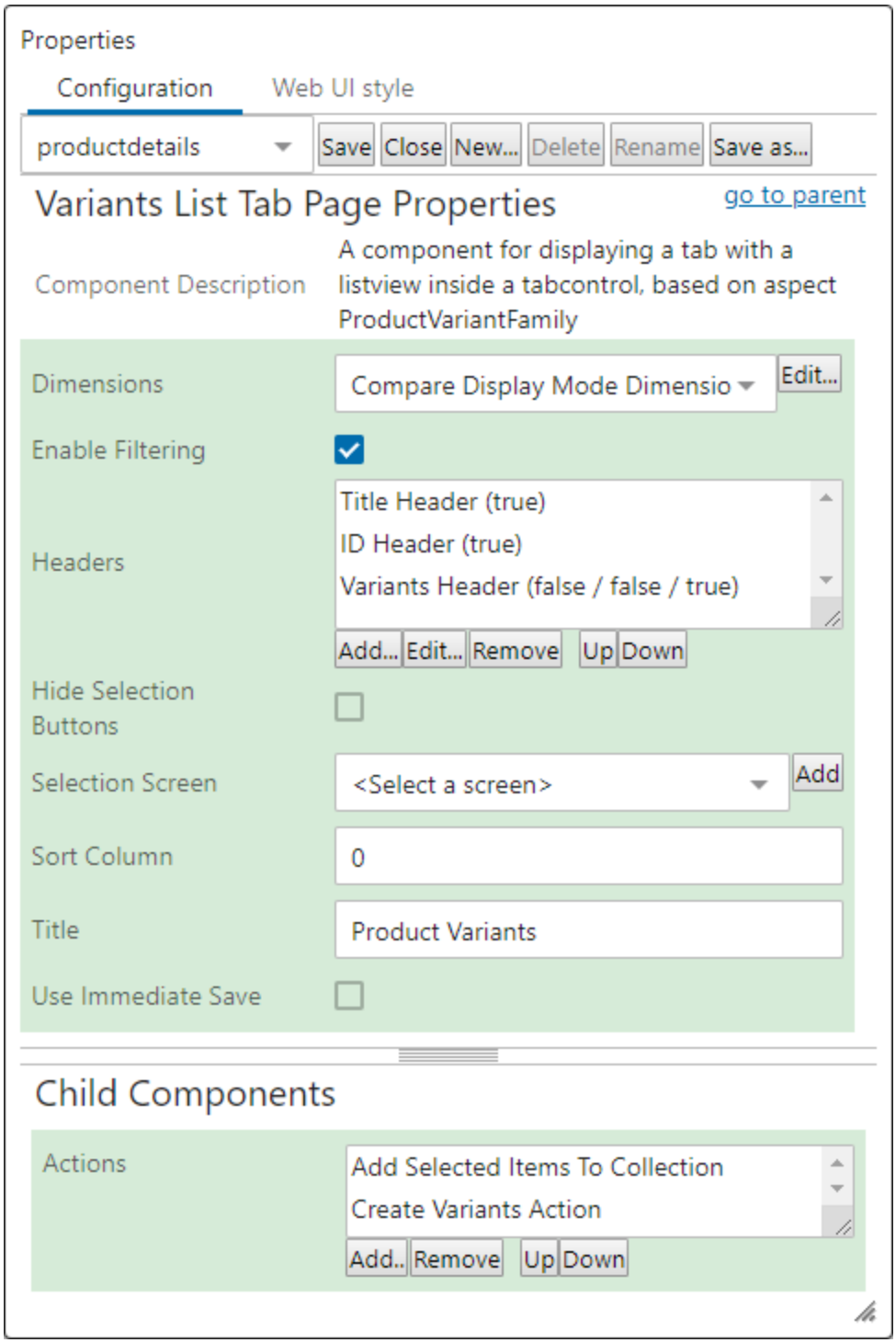
Configure Product Variants List

Use the following steps to display a product variants list view.

1. In Web UI, product variants can be displayed using the **Variants List Tab Page**. You can insert it into a **Tab Control** on a **Node Details** screen using the Child Components 'Main' parameter. For more information, see the **Design Mode Basics** topic and the **Node Details Screen** topic in the **Web User Interfaces / Web UI Setup and User Guide** documentation.



2. Once added to the Node Details screen, double-click the Variant List Tab Page component to set the configuration parameters.



Dimensions

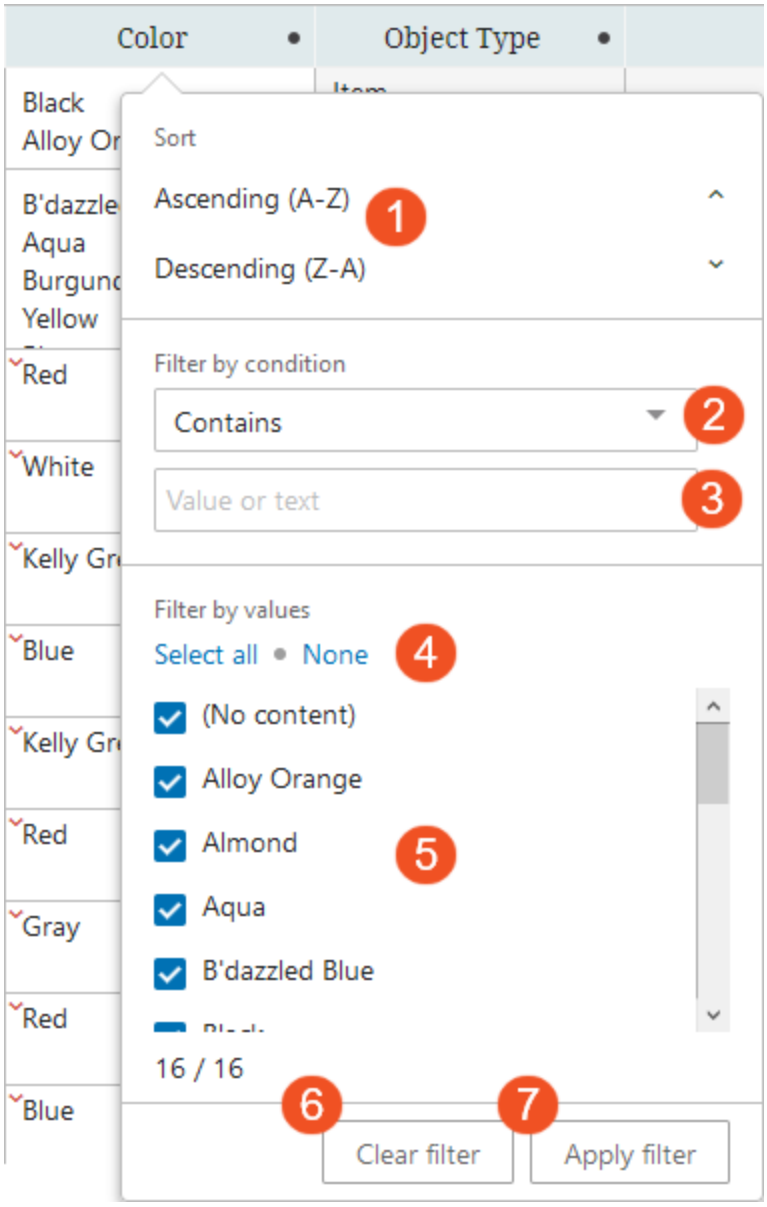
This option may change the standard sheet dimensions.

Enable Filtering

When the Enable Filtering option is selected on this component, the ability to filter based on attribute values is enabled.



To apply a filter, click anywhere in a column's header. This brings up the sorting and filtering window highlighted in the screenshot below:



Referencing the numbered elements in the above screenshot to the corresponding numbers in the list below, functionality in the filtering window enables users to:

1. Sort the attributes in either ascending (^) or descending (v) alphabetic order.
2. Filter the linked attributes by a condition to show only attributes that:
   - 'Begins with' - The filter value begins with one or a sequence of character.
   - 'Contains' - The filter value contains one or a specific sequence of characters. This value may appear anywhere within a value string.
   - 'Does not contain' - The filter value does not contain one or a specific sequence of characters, whether in a single-valued or multi-valued cell. This value may appear anywhere within a value string.
   - 'Equals' - The filter value matches a precise set of characters.
   - 'Include Only' - This filter option allows users to set a list of values separated by a delimiter, like commas, semicolons, pipes, tabs, or newlines, and if one or more of the items in the list matches, then those rows will be populated. If no items from the list are present in the column, then no results will display.
   - 'Wildcard' - This filter value option will display attribute values containing a repeated set of characters that includes known variable characters that can be substituted with an asterisk '*' symbol. Wildcard may be used singularly or multiple times in a filter setting. For more information, see the **Wildcard in Searches** section of the **Searching Functionality** topic of the **Getting Started / User Guide** documentation.
3. With any of the above filtering criteria in place, users may enter their search terms into the **Search** field. As characters are typed into this field, the blue-checked values that display in the filtering menu change to reflect a filtered list of attribute values culled from the column's full list of values. From these, users may select only those attributes they want to display by adding a check beside each.

   The default filter setting is 'Contains.' For the settings 'Begins with,' 'Equals,' and 'Wildcard,' if a user just deselects any value in the search list, then the filter will revert to a 'Contains' filter once it is applied. For the settings 'Begins with,' 'Contains,' 'Equals,' and 'Wildcard,' any values entered that do not meet the filter criteria are removed from the filtered selection. For the setting 'Does not contain,' any rows containing values that do meet the filter criteria are removed from the filtered selection.

   Note: If you are dealing with multi-valued attributes where more than one value is populated, then All Values for that attribute are considered when creating a filter. Excluding one value by a filter does not mean the attribute will necessarily be removed from the view.
4. Click the 'Select all' button to ensure all listed attribute values are checked. Click the 'None' button to deselect all attribute values. When users click 'None', the 'Apply filter' button is grayed out and cannot be selected until at least one of the attribute values is selected.
5. Users may also forgo the filtering and manually select from the full attribute list only those attributes they want to display by adding a check beside each. All values are checked by default.
6. Quickly clear any work you have done in the filtering window by clicking the 'Clear filter' button. The configuration you have applied remains in this filtering window, even if you clear the filter on the actual attribute link table. To clear all filtering, first clear the filter from the table, and then in the filtering window, click the 'Clear filter' button.
7. Apply the configured filter to the attribute link table by clicking the 'Apply filter' button.

Note: The sorting and filtering option applied in the attribute link editor screen of a product will be applicable only on the current selection. Once the user navigates to a different selection, sorting and filtering applied on the previous selection will be removed.

Note: When a Node List contains more than the configured number of objects (the default maximum being 5,000), sorting and filtering is disabled. This is because when Node Lists contain more than the configured maximum for display, all objects above that maximum will not display. Because sorting and filtering actions applied to the displayed objects will not also apply to non-displayed objects, sorting and filtering is disabled. For example, if an advanced search result contains 10,000 results, and the user clicks a header to sort and filter the Node List based on the values in the column, it will not be obvious that all objects after 5,000 are not included in the filtering, and so the resulting list will not be accurate. When users initiate sorting and filtering on a Node List with more than the configured maximum number of objects, an informational dialog displays that reads, 'Sorting and filtering on this subset of data would display inaccurate results, so they have been disabled.' This restriction is applied to all display modes that support sorting and filtering, i.e., Multi Edit, Table, and Compare views.

Headers

Use the 'Variant Header' option to display variant attributes on the table component.

Hide Selection Buttons

This option determines if the 'Select All' and 'Clear selections' options are displayed.

Selection Screen

This option determines the screen that will be displayed when one item is selected. If nothing is selected, the details panel will be hidden.

Sort Column

This field sets the starting column for sorting.

Title

This is the title of the configured component.

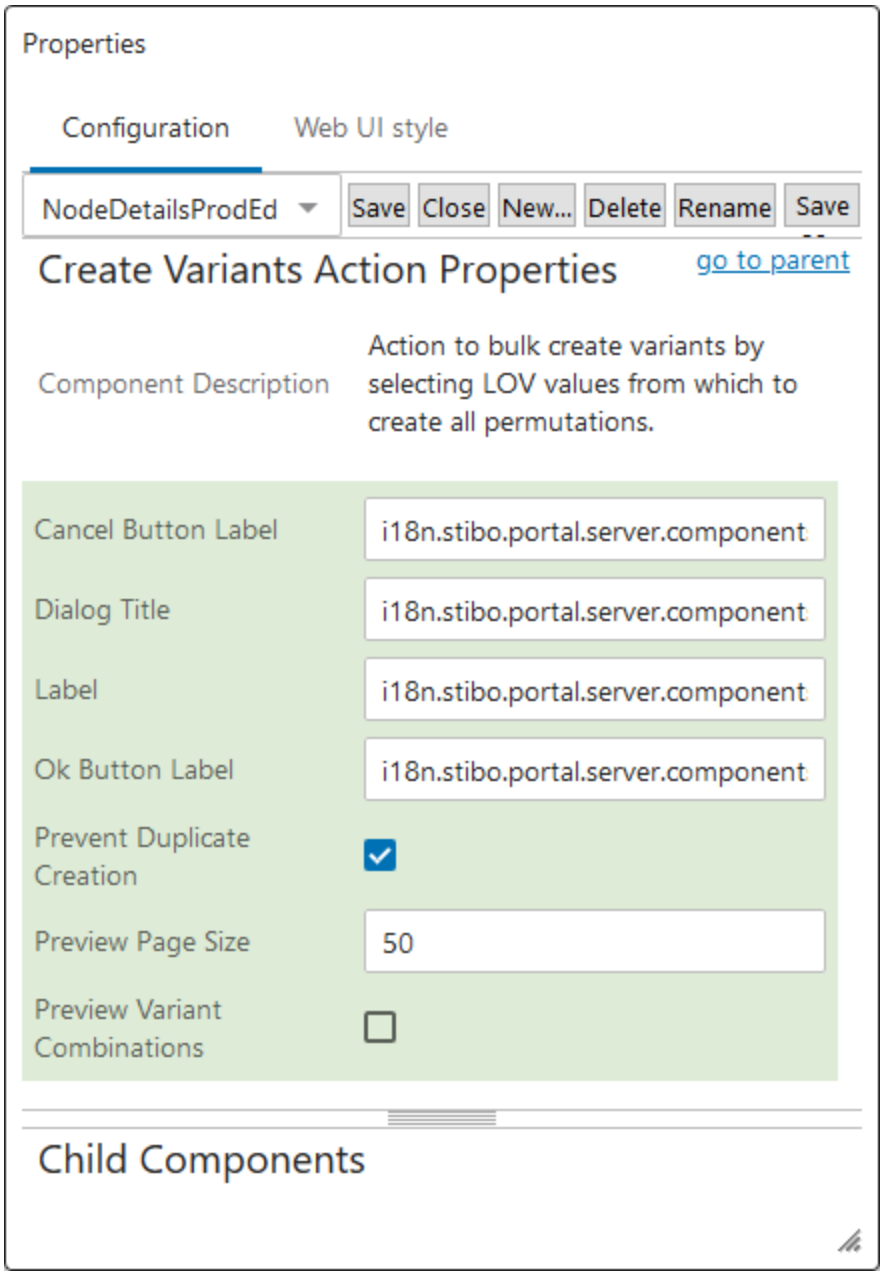
Use Immediate Save

If selected, then STEP will automatically save changes as soon as an edit is performed. Otherwise, the user will need to explicitly save their changes.

Actions

Add the 'Create Variants Action' button if desired. This child component allows for actions to be added for the user to interact with the data. For more information, see the **Action Buttons** topic in the **Web User Interfaces / Web UI Setup and User Guide** documentation.

If added, configure the 'Create Variants Action'



Prevent Duplicate Creation

When checked, clicking the 'Create Variants Action' button and selecting duplicate variant values displays an error and duplicates are not allowed.

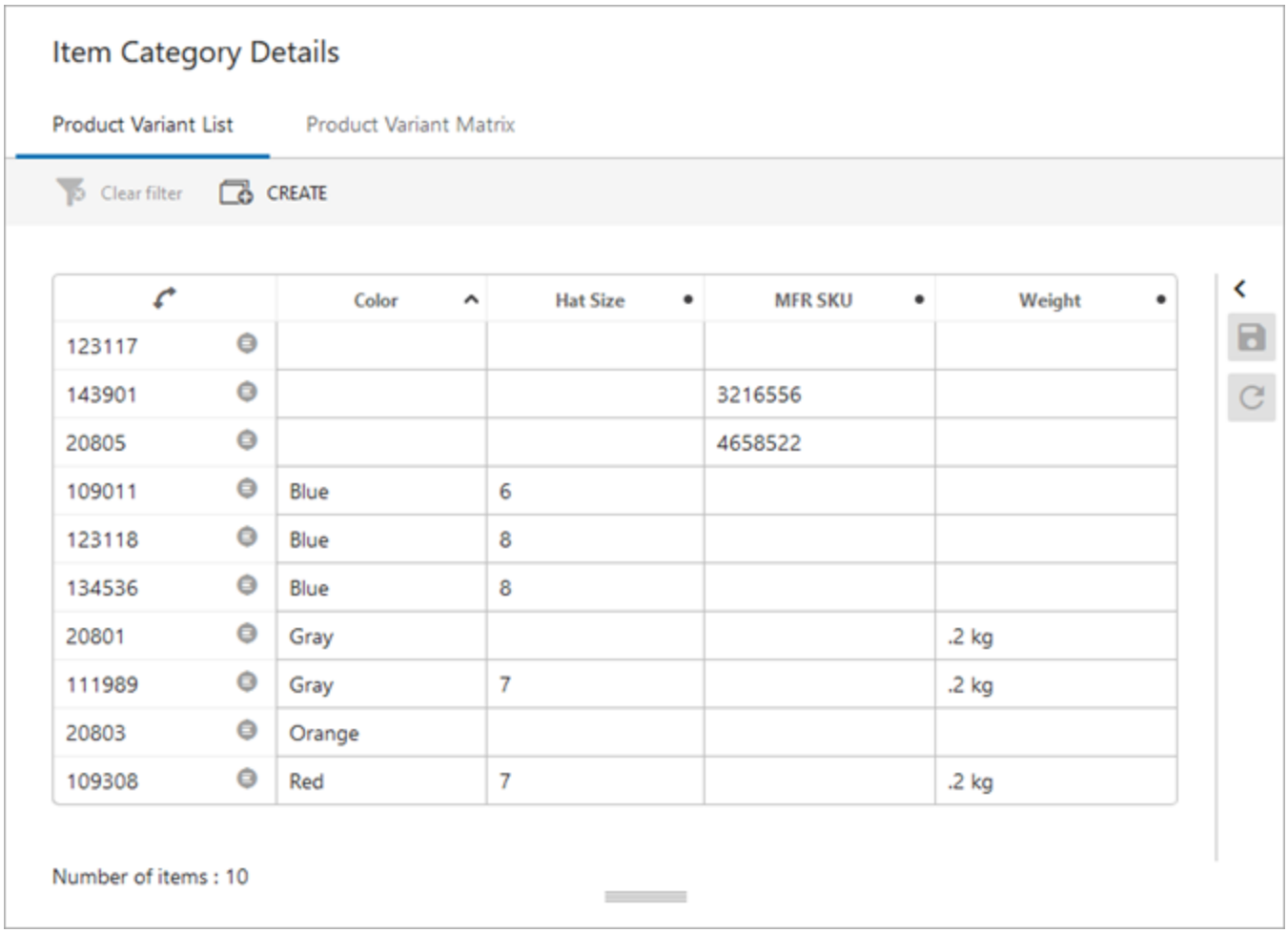
Preview Variant Combinations

When checked, clicking the 'Create Variants Action' button, selecting variant values, and clicking OK opens a Preview dialog with all new variants to be created. The user can click OK to create the items being previewed, or Cancel to return to the create variants dialog.

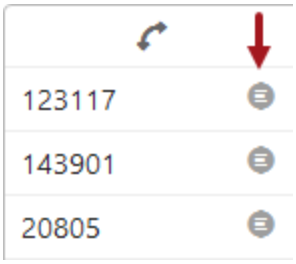
Note: As with any screen, the Node Details screen with Variants List Tab Page must be mapped appropriately in Main Properties > Mappings for end users to be able to access it.

Use the List View

1. In the Tree, expand the Primary Product Hierarchy, and select the relevant product parent node that is configured for variants to display the list.



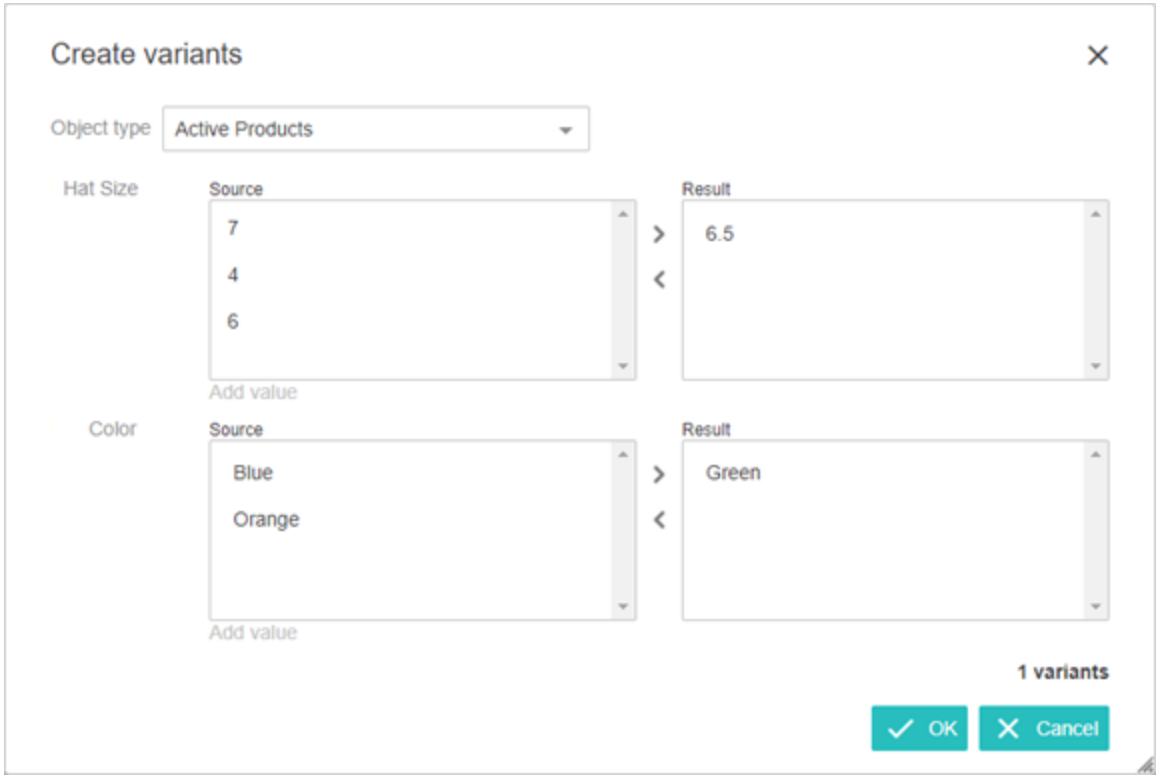
2. Click the Variants List tab to view the following information:
   - By default, product IDs are displayed in the first column and variant attributes and values are displayed to the right.
   - If a Selection Screen is configured (in the Configure Product Variants List section above), click the details button to the right of an ID to show details for the selected product.



- Double-click into a cell to add a value for the variant attribute. LOV attributes show a dropdown and non-LOV attributes allow you to type directly into the cell.

Create a Variant

If added as an Action during configuration, click the 'Create variants' button to display the 'Create variants' dialog. Variant attributes are displayed and values can be added for the selected object type. If the new values match an existing variant, the potential duplicates are shown above the Cancel button. In the following example, one other variant already exists with the selected 'Hat Size' and 'Color' combination selected.

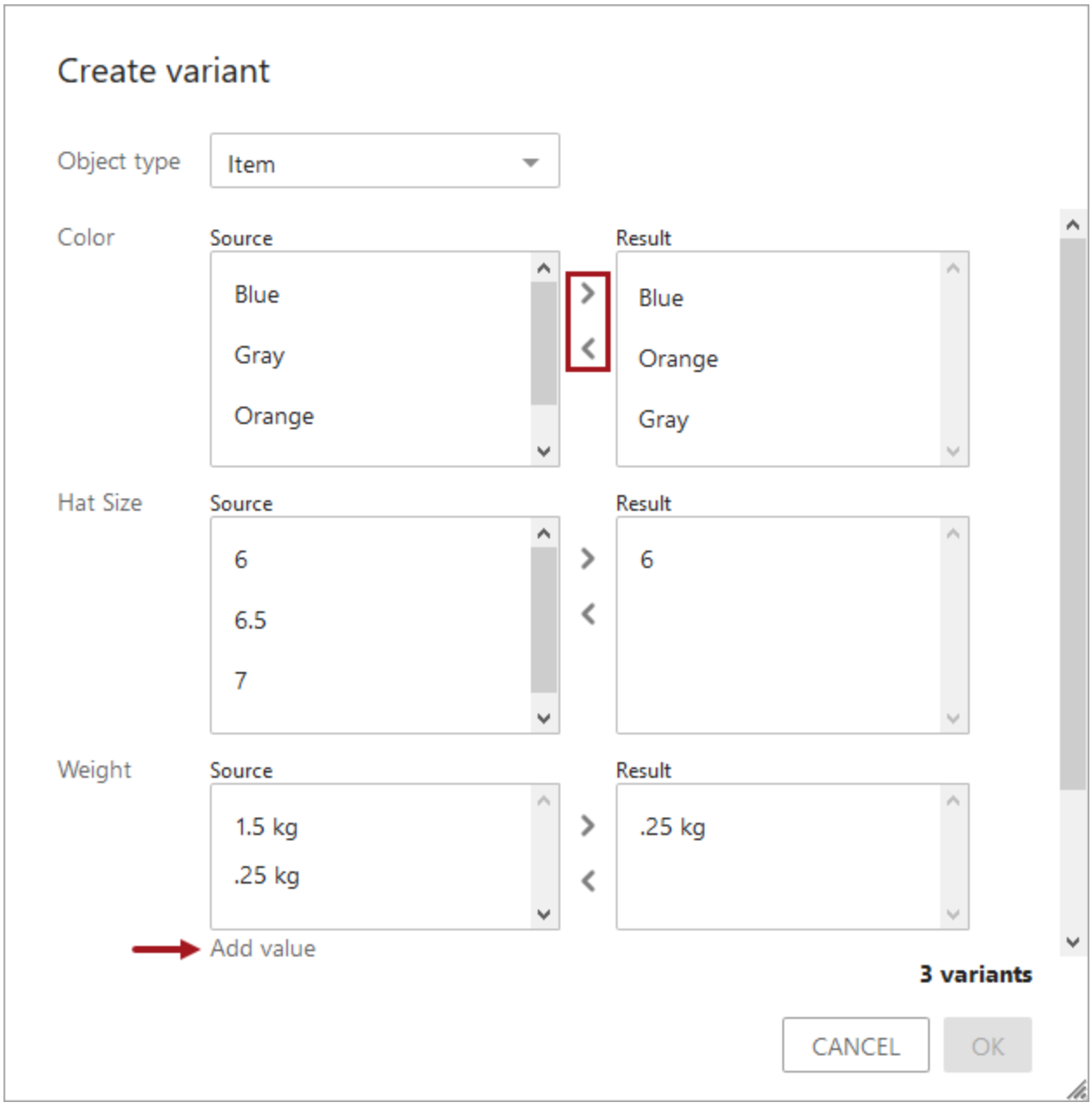


When creating a variant manually (rather than using an import file), note that only variant attributes values are added. Additional values can be added on a product editor screen.

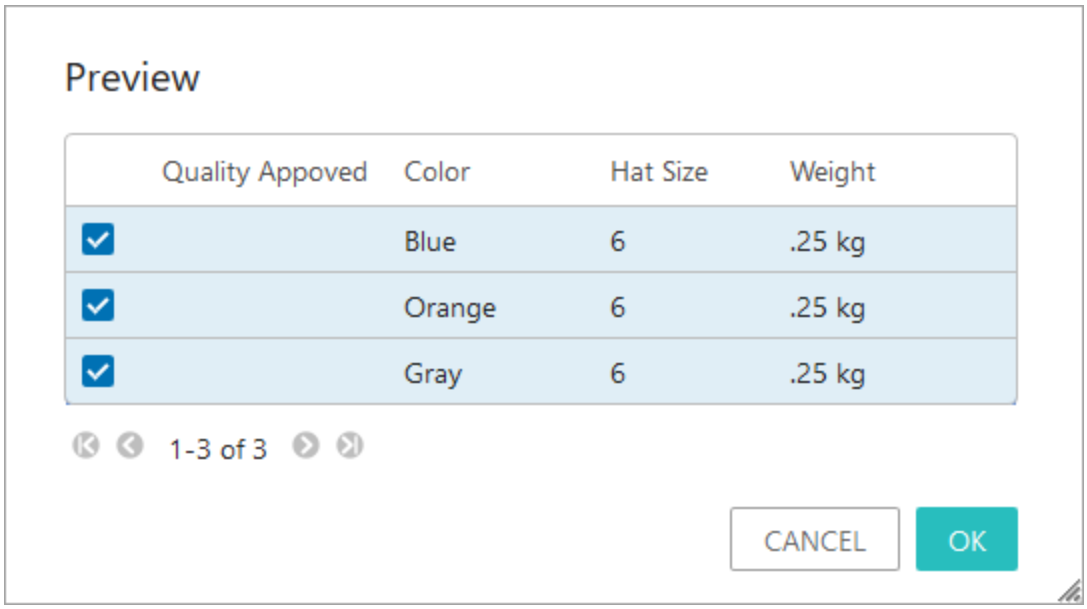
1. Verify an ID pattern for the variant object type is defined. This 'Create variants' button is disabled if no ID pattern is included. For more information, see the **Autogenerate using Name Pattern and ID Pattern**

topic in this guide.

2. Select a cell within the table to activate the 'Create Variant' button (configured in the Configure Product Variants Matrix section above).
3. Click the 'Create Variant' button to display the 'Create Variant' dialog. Variant attributes and the values are displayed and for the selected object type. Add the required data and click OK.



4. If the parameter **Preview Variants Combinations** is configured (in the Configure Product Variants Matrix section above), the preview dialog shows the chosen combinations to be created. Click OK to proceed with the process or click Cancel to return to the create variant dialog.



Note: If the **Prevent Duplicate Creation** parameter is checked (configured in the Configure Product Variants Matrix section above), the combinations that match an existing variant are not created when the action is applied.

Product Variants Matrix in Workbench

The Matrix view focuses on the product IDs and is organized using two variant attributes. In the matrix view, product variant IDs are placed in a cell with attributes that match the selected row and column attributes. After validating the data, potential duplicate products are displayed with a red background and show all product variant IDs that have matching selected attributes.

For a simpler view, use the list view as defined in the **Product Variants List in Workbench** topic.

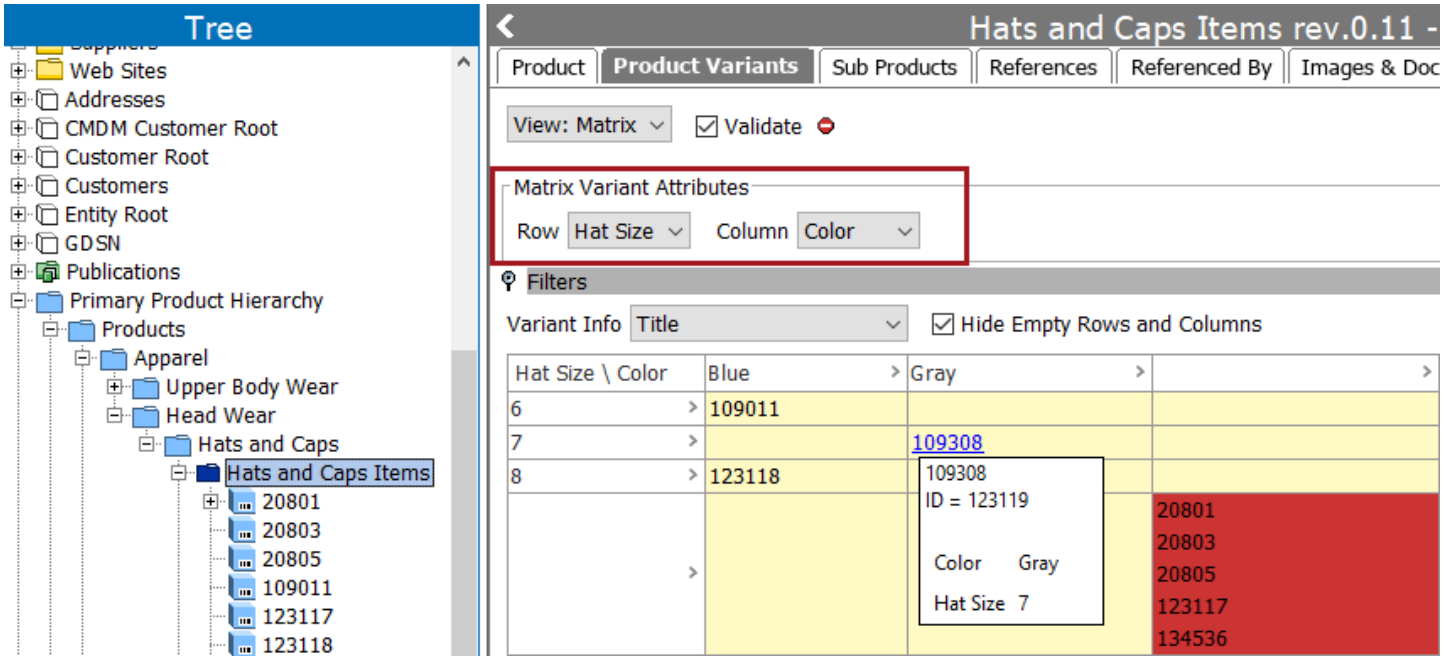
To view variant information in the Web UI, see the **Products Variants List in Web UI** topic or the **Product Variants Matrix in Web UI** topic.

Prerequisite

Before using the product variant view, you must complete the steps in the **Setting Up Product Variants** topic.

Use the Matrix View

1. In the Tree, select the relevant parent node that is configured for variants.
2. Click the Product Variants tab, and select **View: Matrix** from the dropdown.
3. In the **Matrix Variant Attributes** section, choose two variant attributes to set the data displayed as **Row** and **Column**.

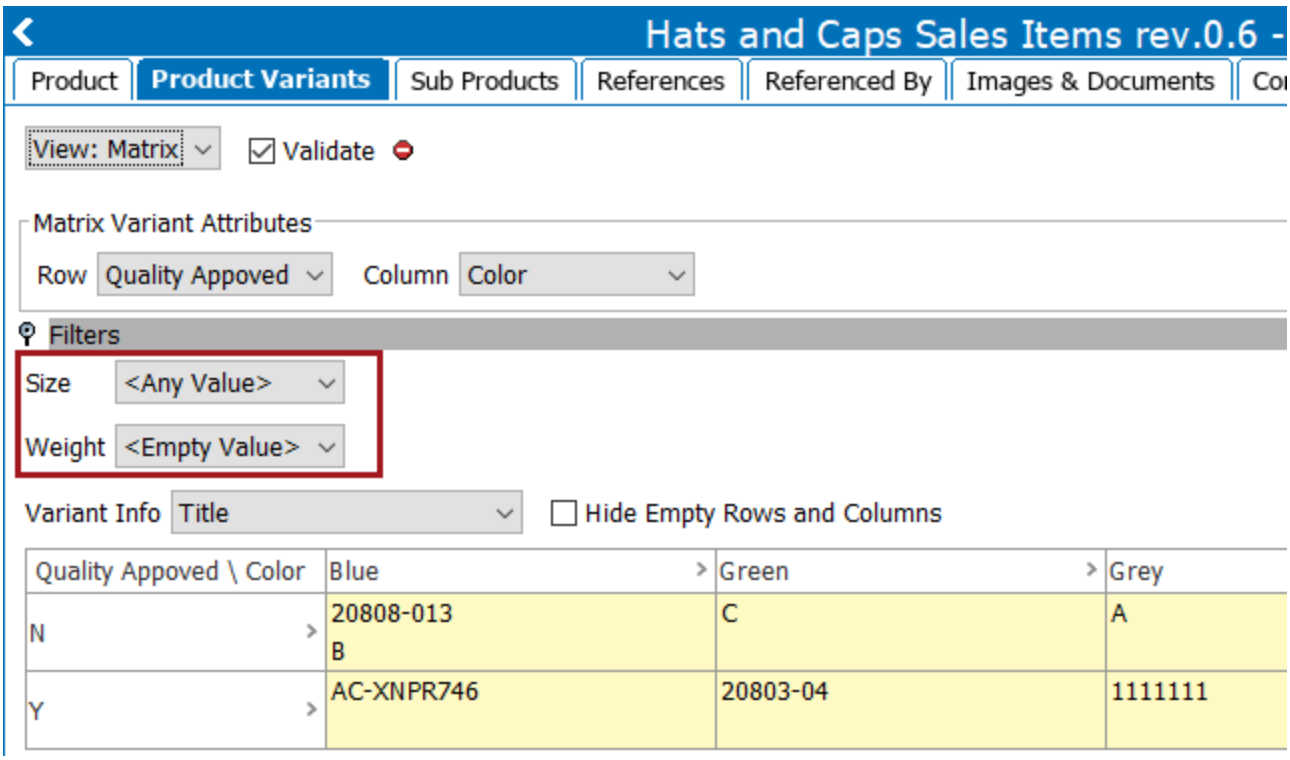


4. Check the **Validate** checkbox to view the following information:

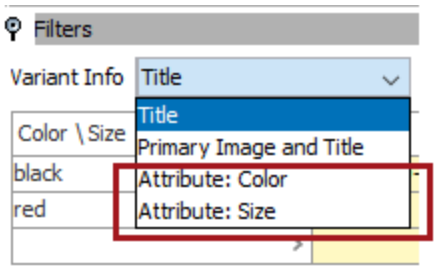
- The possible duplicate variants are highlighted in red to show IDs that have the same values for the variant attributes, including empty values.
- Hover over a product variant ID to display the ID as a hyperlink. Click the link to display the editor for the product.
- Hover over a product variant ID to display information about its variant values and possible duplicates.

5. In the **Filters** area, product variant attributes that are not selected as a row or column (shown below) are available, as defined below. The filter options are existing values, <Any value>, and <Empty value>.

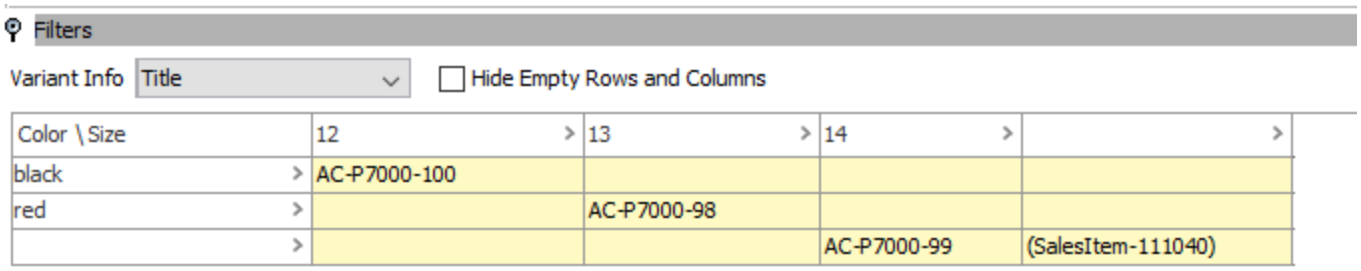
   - If more than two variant attributes are in use, each is displayed as a filter option between the Filters flipper and the Variant Info dropdown. In the image below, the variant attributes are Quality Approved, Color, Size, and Weight. Since Quality Approved and Color were selected for Row and Column, Size and Weight are available as filters.



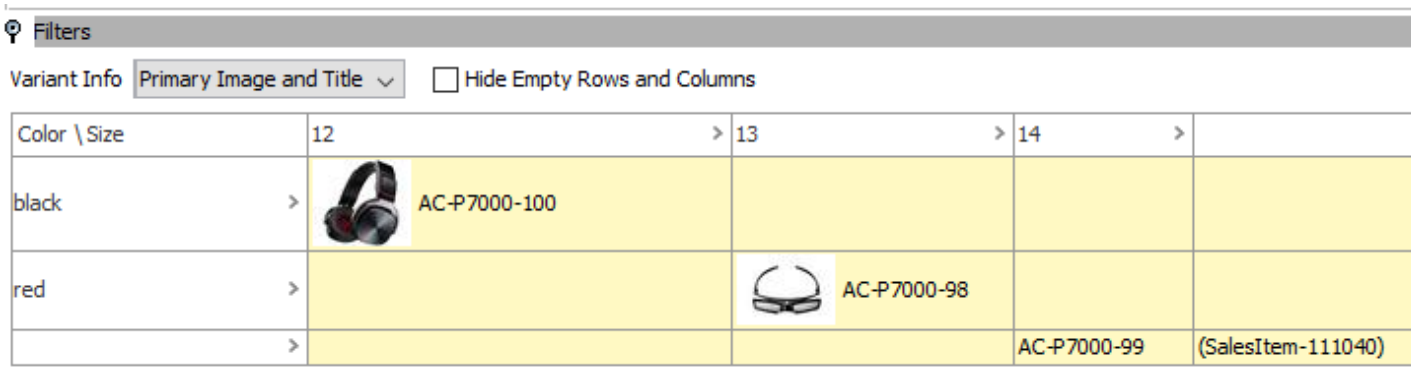
   - The **Variant Info** dropdown includes default options of 'Title' and 'Primary Image and Title' as well as variant attributes, such as 'Attribute: Color' shown below.



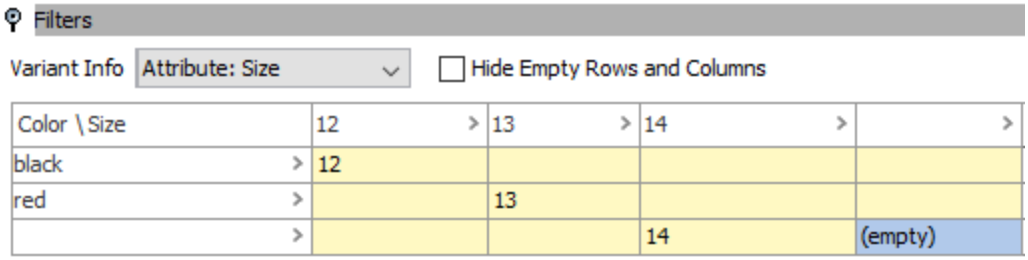
- **Title**: Displays the results only with the name of the product.



- **Primary Image and Title**: Displays the result with the name and primary image of the product. If there is no image, it displays only the name for the product.

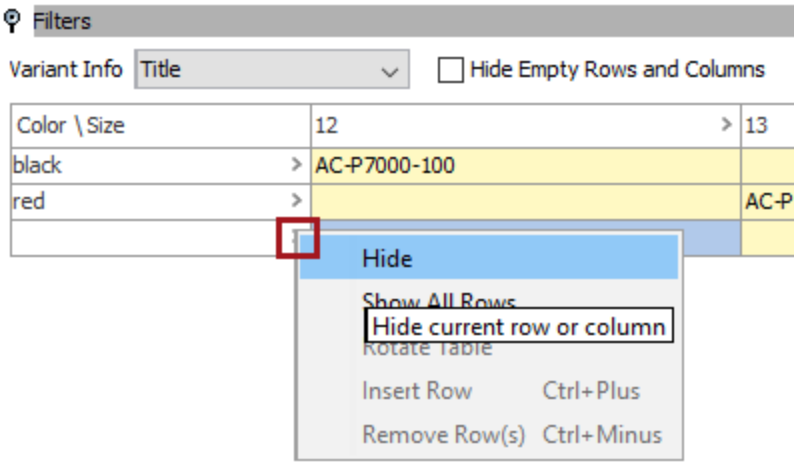


- **Variant Attributes**: If a variant attribute was selected in the drop-down list, it displays the values. If there are any blank values, it displays as '(empty).' In the example below the 'Size' attribute values are displayed across the Color attribute values.



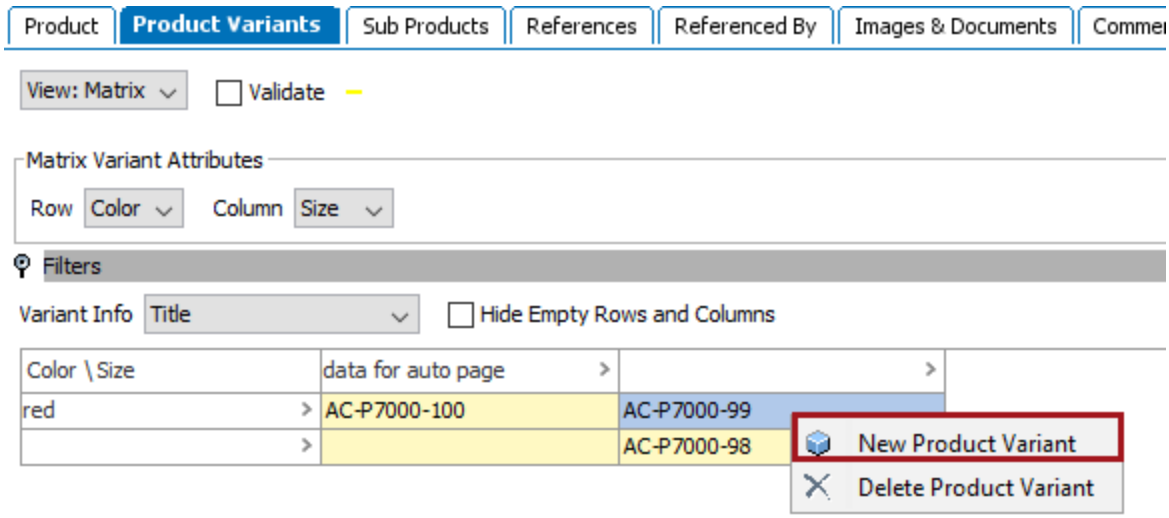
- The **Hide Empty Rows and Columns** checkbox allows empty rows / columns to be hidden in the view.

  Additionally, a user can hide rows and columns by clicking the arrow [>] for the row / column and choosing the Hide option to temporarily hide them from view.



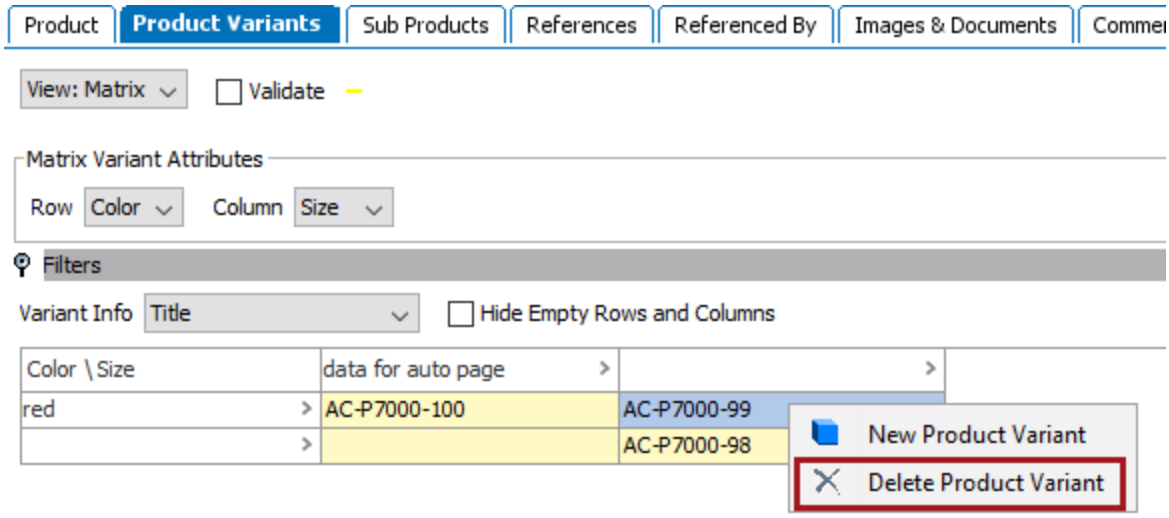
Create a Variant

To manually create a variant, right-click the relevant cell, and then click **New Product Variant**. A variant is created with attribute values that correspond to those of the selected cell. The typical way to create variants is to import them to the parent node.



Delete a Variant

To delete a variant, right-click the relevant cell, and then choose **Delete Product Variant**. A confirmation dialog is displayed and the user can delete the variant or cancel the process.


Product
Product Variants
Sub Products
References
Referenced By
Images & Documents
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

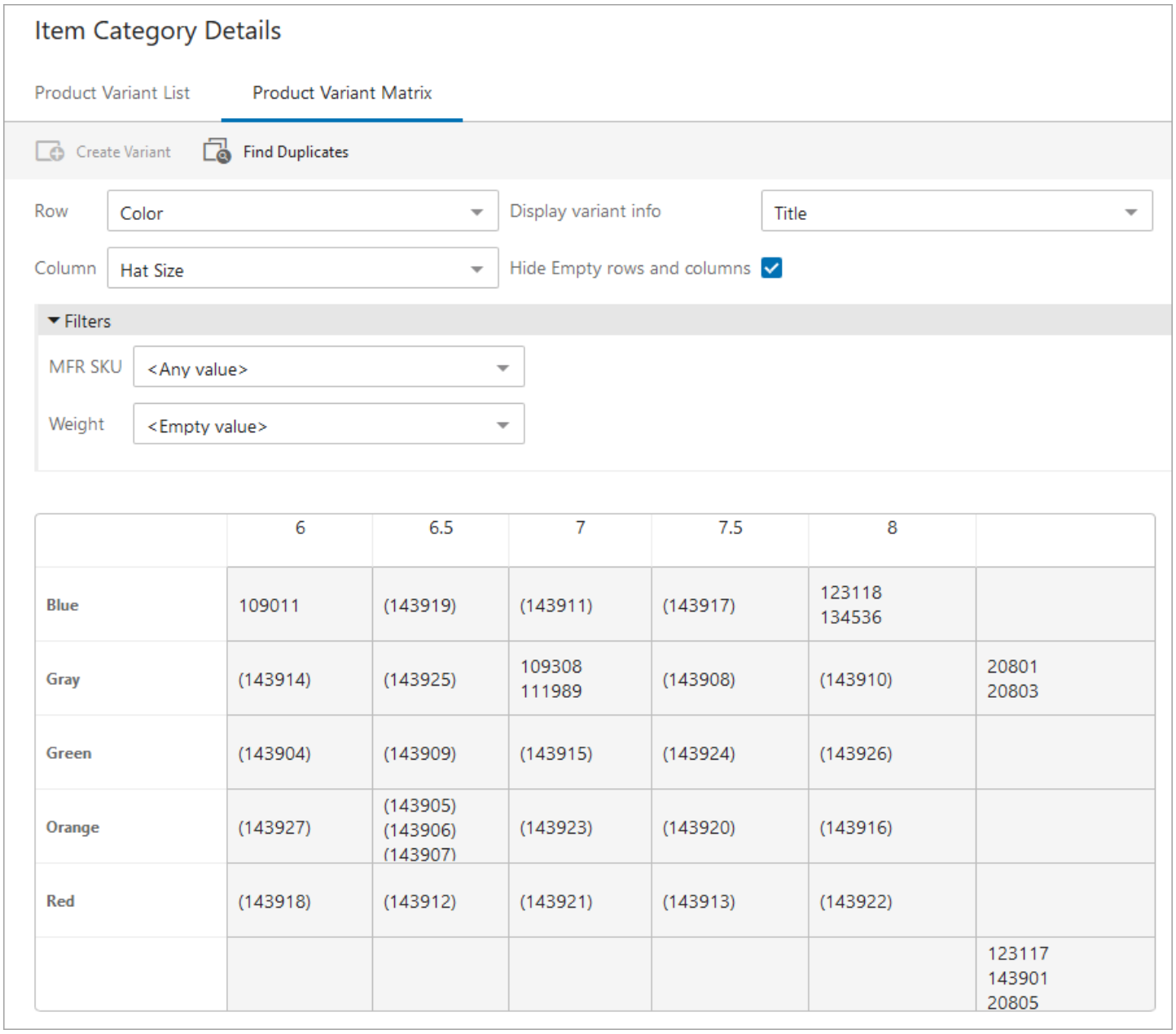
Product Variants Matrix in Web UI

The Matrix view focuses on the product IDs and is organized using two variant attributes. In the matrix view, product variant IDs are placed in a cell with attributes that match the selected row and column attributes. After validating the data, potential duplicate products are displayed with a red background and show all product variant IDs that match the selected attributes.

For a simpler view, use the list view as defined in the **Product Variants List in Web UI** topic.

To view variant information in the workbench, see the **Product Variants Matrix in Workbench** topic or the **Products Variants List in Workbench** topic.

The configured Product Variants Matrix tab page is shown in the following image:



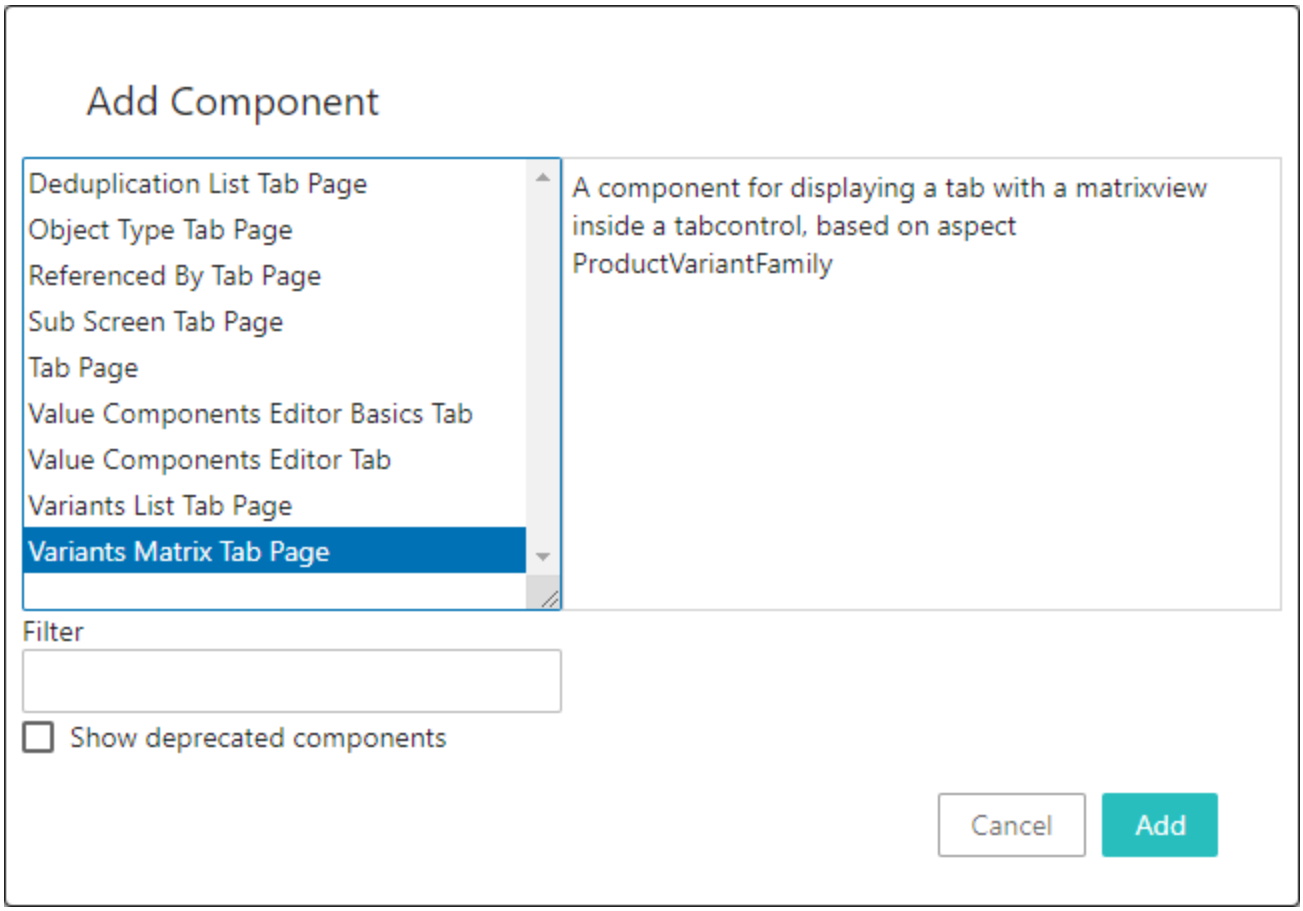
	6	6.5	7	7.5	8	
Blue	109011	(143919)	(143911)	(143917)	123118 134536	
Gray	(143914)	(143925)	109308 111989	(143908)	(143910)	20801 20803
Green	(143904)	(143909)	(143915)	(143924)	(143926)	
Orange	(143927)	(143905) (143906) (143907)	(143923)	(143920)	(143916)	
Red	(143918)	(143912)	(143921)	(143913)	(143922)	
						123117 143901 20805

Prerequisite

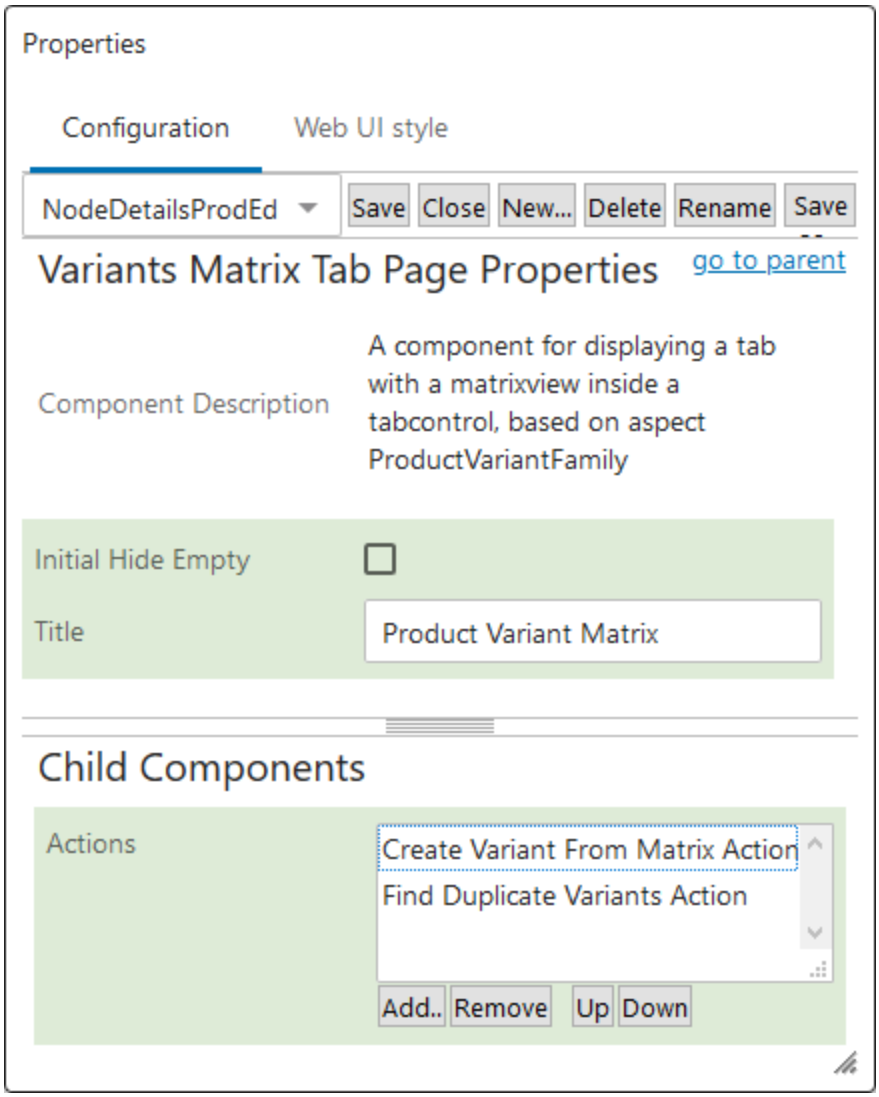
Before using the product variant view, you must complete the steps in the **Setting Up Product Variants** topic.

Configure Product Variants Matrix

1. In the Web UI, product variants can be displayed using **Variants Matrix Tab Page**. You can insert it into a **Tab Control** on a **Node Details** screen using the Child Components 'Main' parameter. For more information, see the **Design Mode Basics** topic and the **Node Details Screen** topic in the **Web User Interfaces / Web UI Setup and User Guide** documentation.



2. Once added to the Node Details screen, double-click the Variants Matrix Tab Page to set the configuration parameters.



Initial Hide Empty

When checked, the 'Hide Empty rows and columns' checkbox is checked by default and only rows and columns with values are displayed initially. The user can uncheck the 'Hide Empty rows and columns' checkbox to see empty variant attribute values.

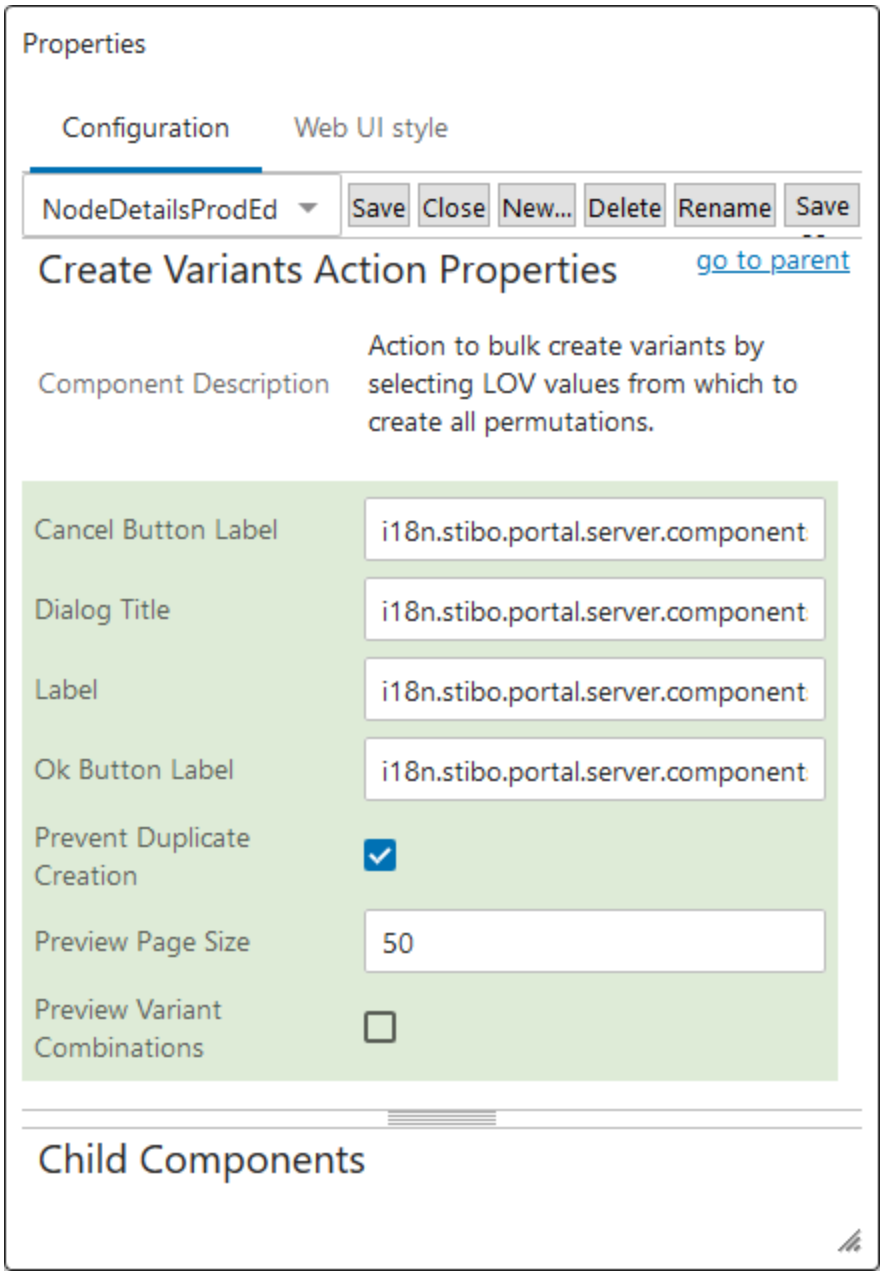
Title

This is the title of the configured component.

Actions

Add the 'Create Variant From Matrix Action' and 'Find Duplicate Variants Action' buttons if desired. This child component allows for actions to be added for the user to interact with the data. For more information, see the **Action Buttons** topic in the **Web User Interfaces / Web UI Setup and User Guide** documentation.

If added, configure the 'Create Variants Action.'



Prevent Duplicate Creation

When checked, clicking the 'Create Variants Action' button and selecting duplicate variant values displays an error and duplicates are not allowed.

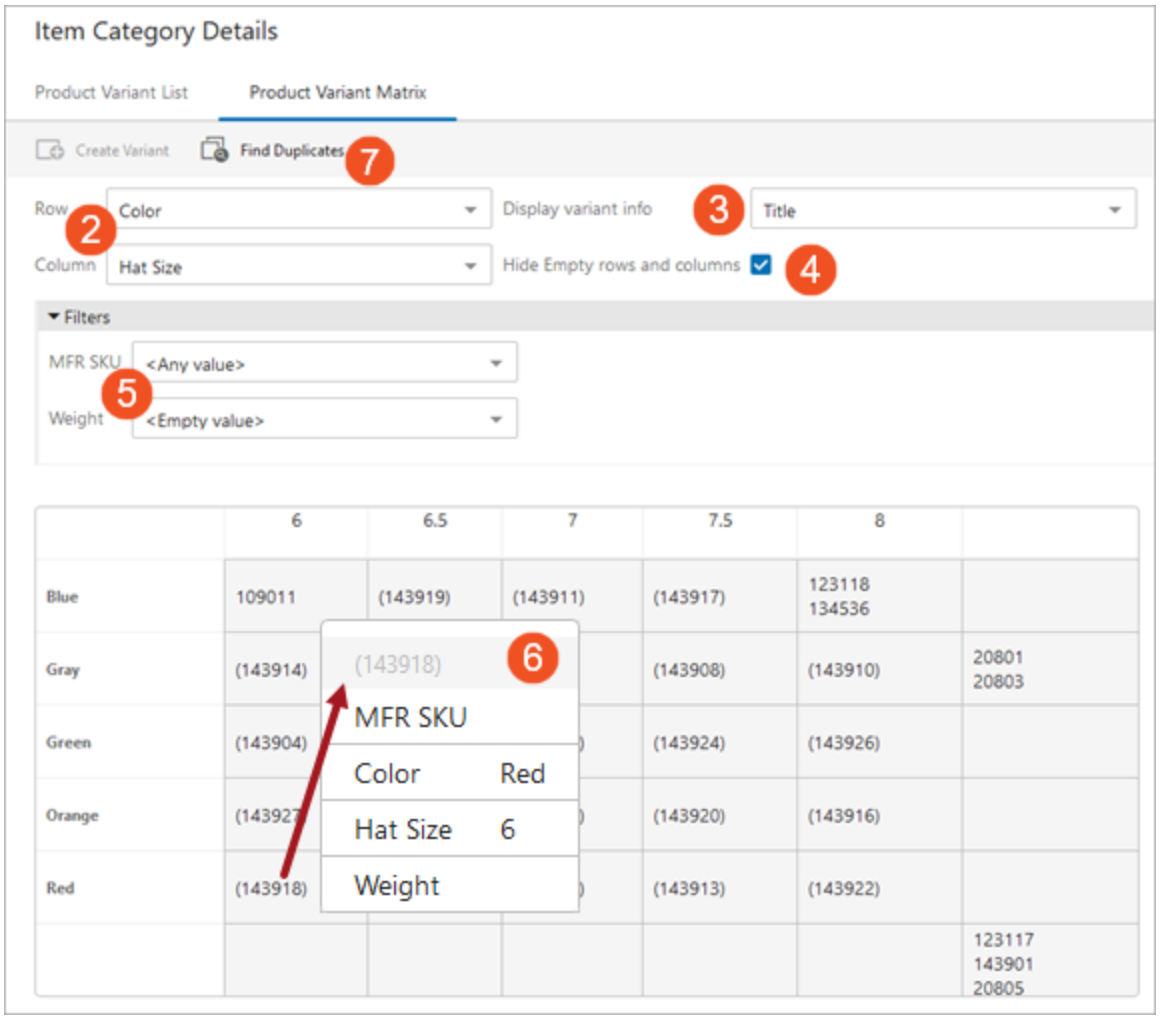
Preview Variant Combinations

When checked, clicking the 'Create Variants Action' button, selecting variant values, and clicking OK opens a Preview dialog with all new variants to be created. The user can click OK to create the items being previewed, or Cancel to return to the create variants dialog.

Note: As with any screen, the Node Details screen with Variants Matrix Tab Page must be mapped appropriately in Main Properties > Mappings for end users to be able to access it.

Use the Matrix View

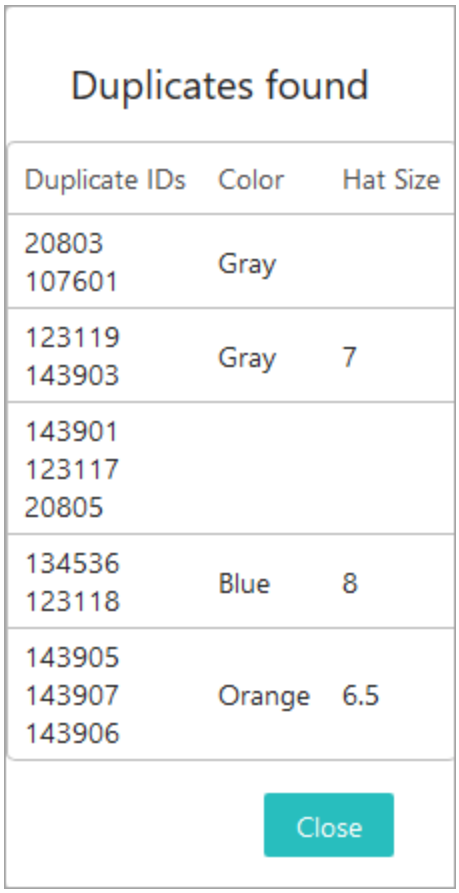
1. In the Tree, expand the Primary Product Hierarchy, and select the relevant product parent node that is configured for variants to display the matrix.



2. Set the **Row** and **Column** parameters to the variant attribute to be displayed the row and the column of the matrix. All variant attributes are available.
3. Set the **Display variant info** parameter to determine how the variant's details within the matrix are displayed. In addition to the variant attribute values, 'Title' and 'Primary Image and Title' are also available for selection.
4. Check the **Hide Empty rows and columns** parameter to exclude empty valued options from the matrix.
5. In the **Filters** section, matrix data can be displayed based on the variant attributes not selected as Row or Column. The filter options are existing values, <Any value>, and <Empty value>. In the image above, the

variant attributes are MFR SKU, Color, Hat Size, and Weight. Since Color and Hat Size were selected for Row and Column, MFR SKU and Weight are available as filters.

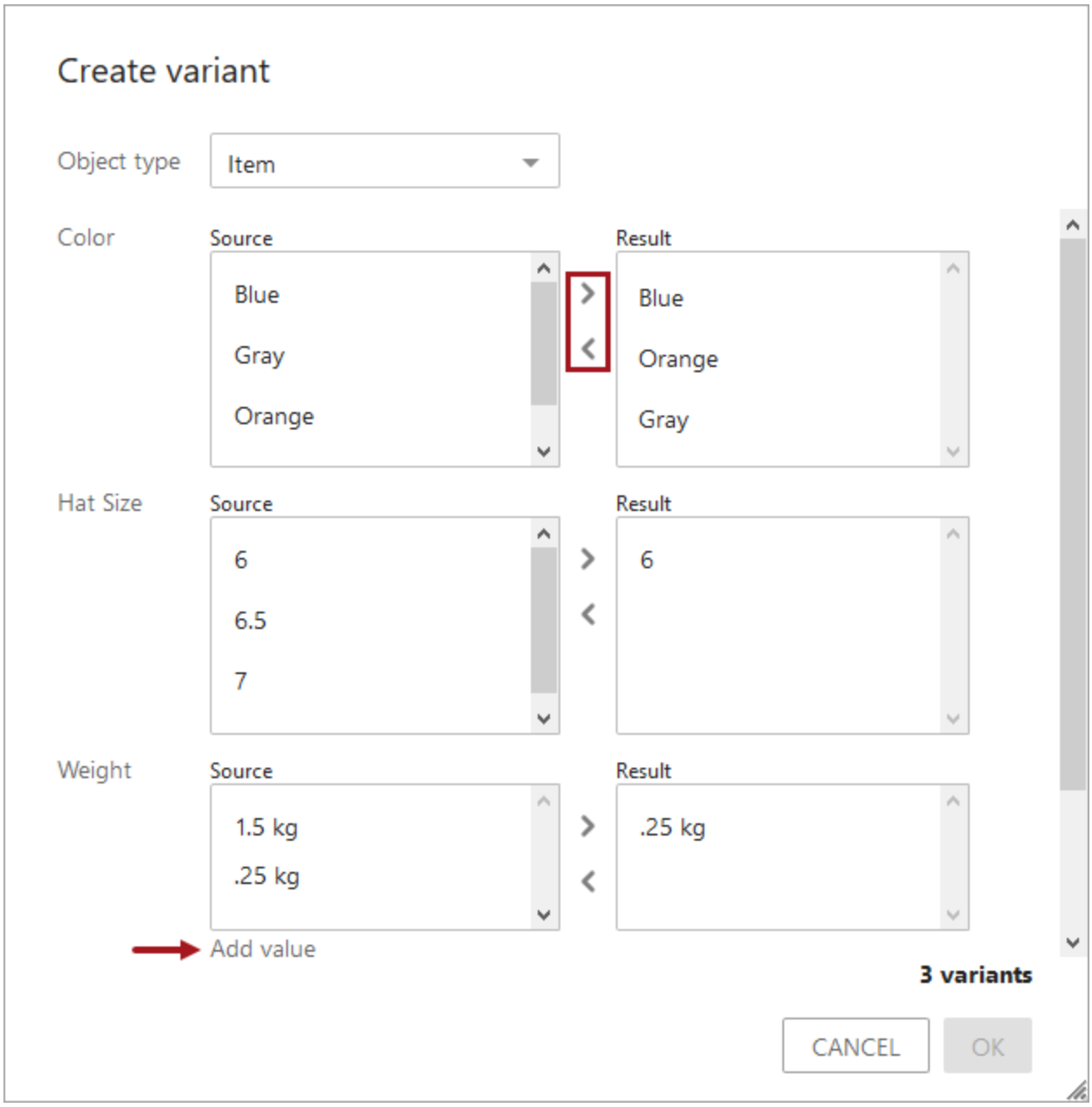
6. On the Variants Matrix, hover over a variant ID to display details about the variant attributes.
7. Click the 'Find Duplicates' button (if configured in the Configure Product Variants Matrix section above) to display the 'Duplicates found' dialog. Values that are used for more than one variant are displayed as potential duplicates. In the following example, two variant IDs have a value of 'Gray' for the 'Color' attribute and no value for 'Hat Size' attribute.



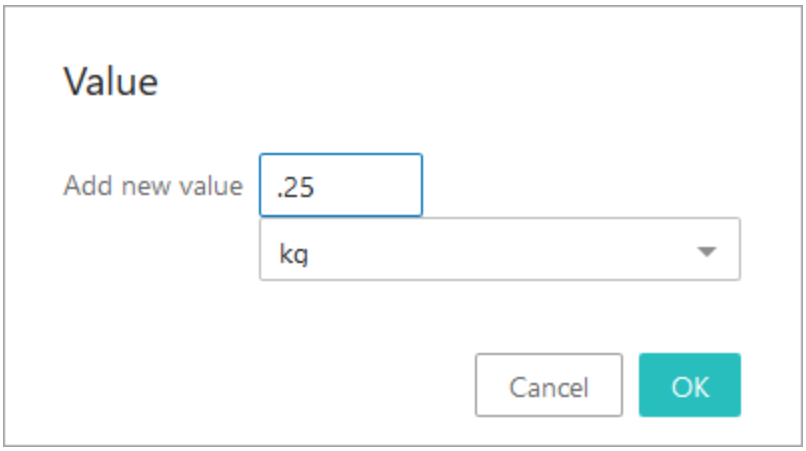
Create a Variant

When creating a variant manually (rather than using an import file), note that only variant attributes values are added. Additional values can be added on a product editor screen.

1. Verify an ID pattern for the variant object type is defined. This 'Create variants' button is disabled if no ID pattern is included. For more information, see the **Autogenerate using Name Pattern and ID Pattern** topic in this guide.
2. Verify the 'Create Variant' button was configured in the Configure Product Variants Matrix section above.
3. Click the 'Create Variant' button to display the 'Create Variant' dialog. Variant attributes and the existing values are displayed and for the selected object type.



4. For each variant attributes:
   - Click one or more of the existing values in the Source column and click the > button to move it to the Result column. You can also double-click existing values to move them, or multi-select values before using the > button.
   - The number of new variants to be created is displayed as the count above the OK button.
   - Click the 'Add value' link when available to supply a new value. Click OK to close the Value dialog.



5. Click OK.
6. If the parameter **Preview Variants Combinations** is configured in the Configure Product Variants Matrix section above, the preview dialog shows the chosen combinations to be created. Click OK to proceed with the process or click Cancel to return to the create variant dialog.



Note: If the **Prevent Duplicate Creation** parameter is checked (configured in the Configure Product Variants Matrix section above), the combinations that match an existing variant are not created when the action is applied.

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.

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.

This condition works on the Product Variant level and returns 'False' if the product variant is a duplicate.

Note: This condition can affect performance based on the number of siblings and variant attributes to be checked.

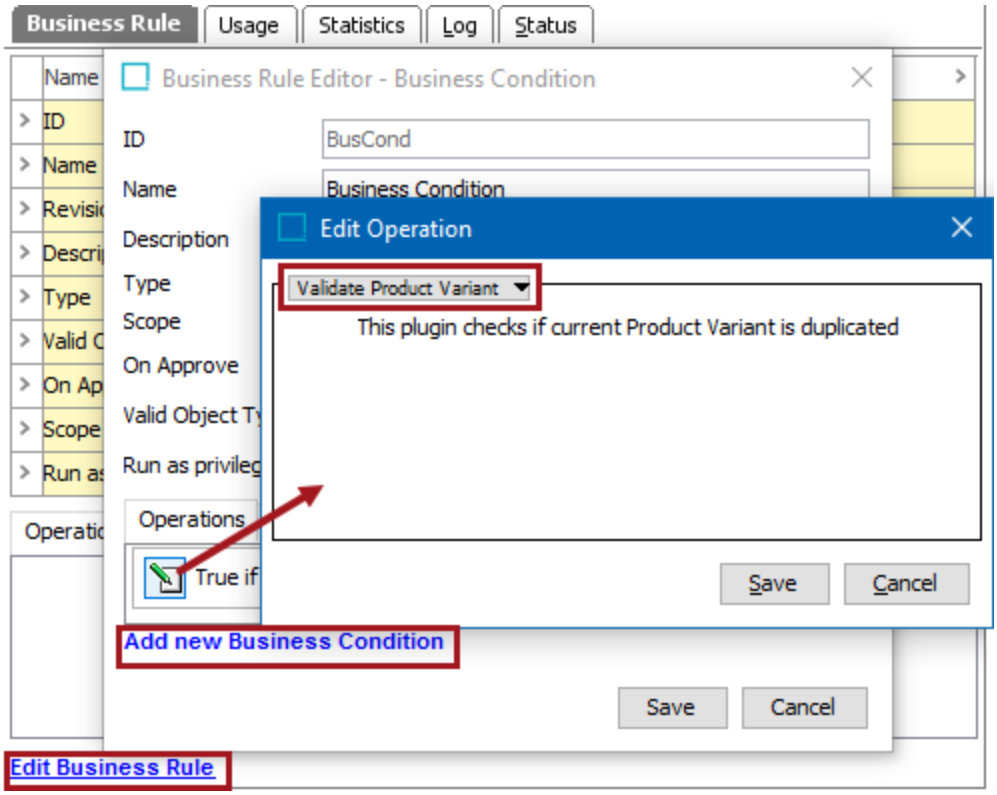
For more information and required configuration, see the **Product Variants** topic in the **System Setup / Super User Guide** documentation.

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 for System Setup

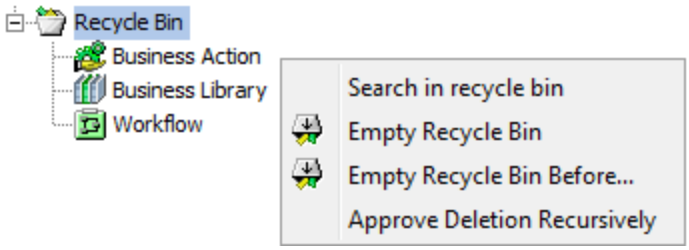
The deletion concept in STEP is quite similar to what you know from Windows and Mac OS. Thus, if you delete the objects in System Setup, they will be moved to the Recycle Bin in the System Setup tab. However, not all objects will display in the Recycle Bin.

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 for the Tree Tab** topic with the **Object Maintenance in the Tree Tab** 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.

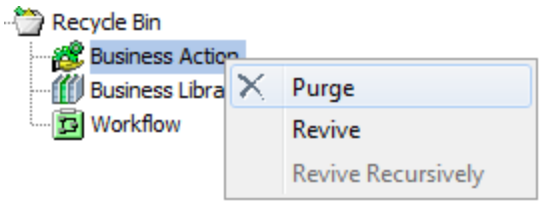
  Note: Items / objects in the Recycle Bin are not searched when doing STEP searches nor when using the workbench Goto functionality.

- **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.

Note: Items in the Tree Recycle Bin are subject to approval, which will be flagged with 'In Use' and 'Not In Use' indicators. The System Setup Recycle Bin will only show the 'Not In Use' indicator. For more

information, see the **In Use and Not In Use Objects** section in the **Approval of Objects** topic of the **Getting Started / User Guide** documentation.

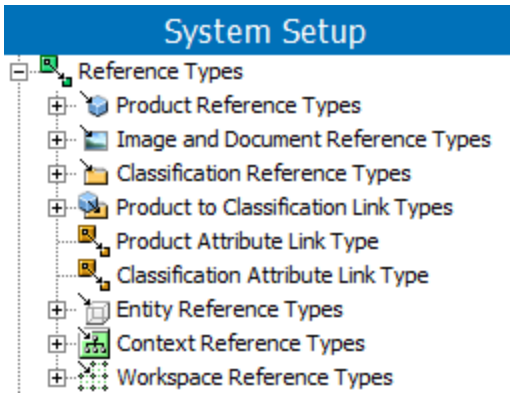
Deleted business rules, workflows, and integration endpoints can be seen in the Recycle Bin and can be revived from that location. Similarly they can be purged.



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 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.

Consideration

For each reference type or link type, the recommended limit is 1000 of the same type on an object. Exceeding this number of references can result in degraded performance across the workbench and Web UI when the data is accessed. For example, when importing objects with an excessive number of references / links, or in Web UI, when displaying the objects and the references / links.

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**
- Create references for use in data containers as described in **References on Data Containers**

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**.

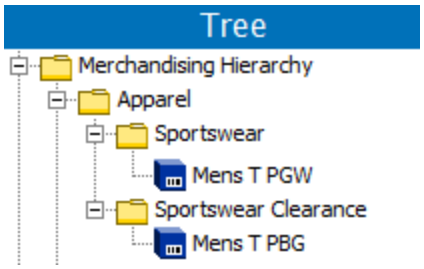
Additional Information About Classification Attribute Links

- 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.

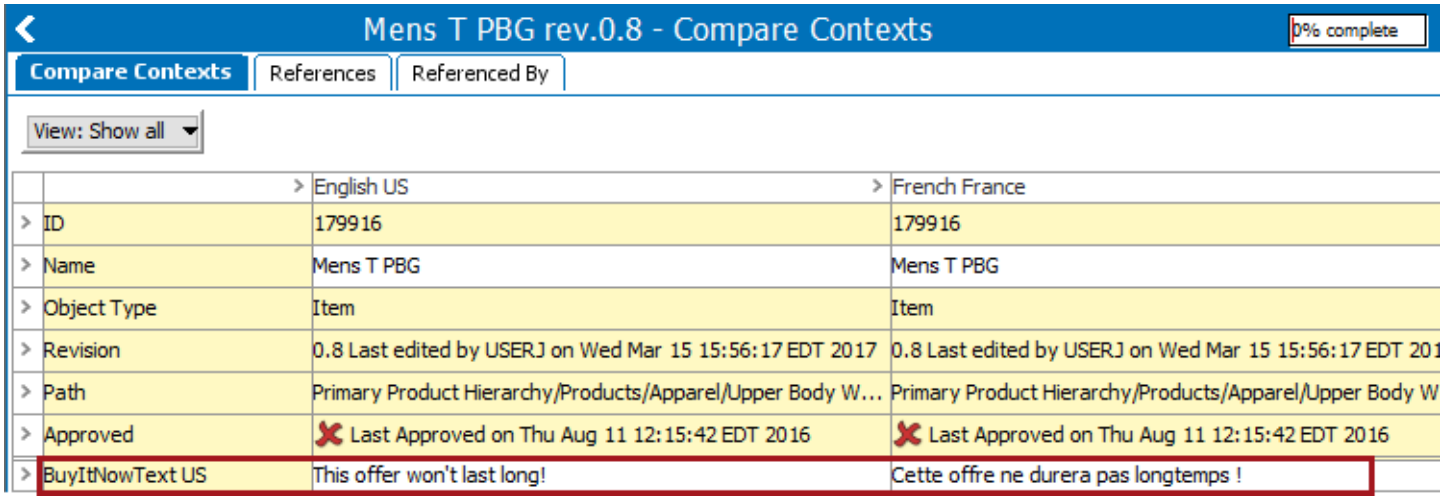


With additional setup, the Classification Attribute Link Type can also include dimension dependency and metadata attributes as described below.

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.



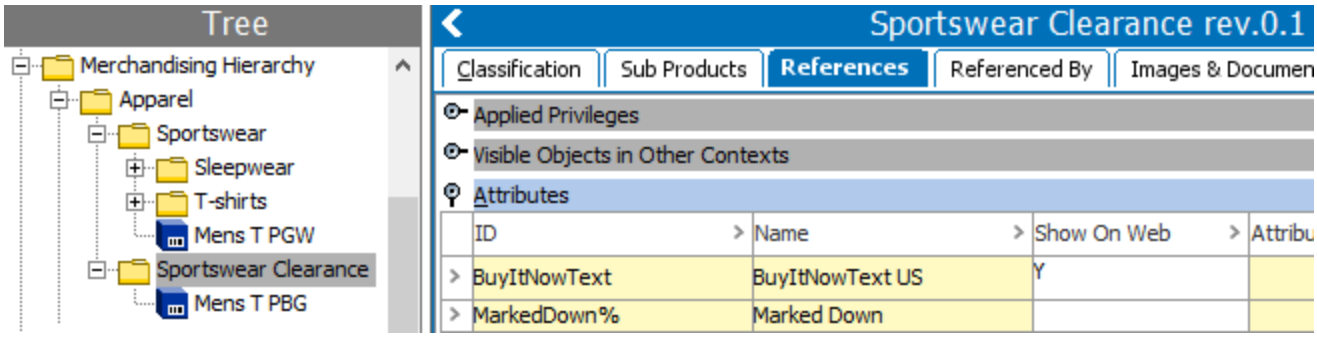
	English US	French France
ID	179916	179916
Name	Mens T PBG	Mens T PBG
Object Type	Item	Item
Revision	0.8 Last edited by USERJ on Wed Mar 15 15:56:17 EDT 2017	0.8 Last edited by USERJ on Wed Mar 15 15:56:17 EDT 201
Path	Primary Product Hierarchy/Products/Apparel/Upper Body W...	Primary Product Hierarchy/Products/Apparel/Upper 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.

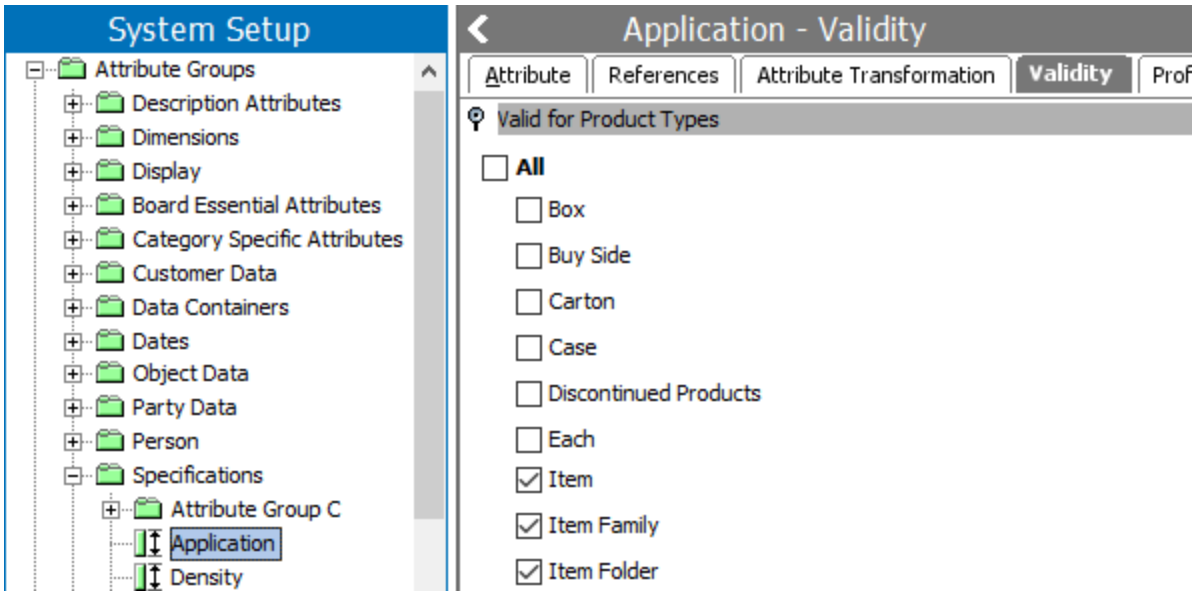


For more information, see **Description Attributes**.

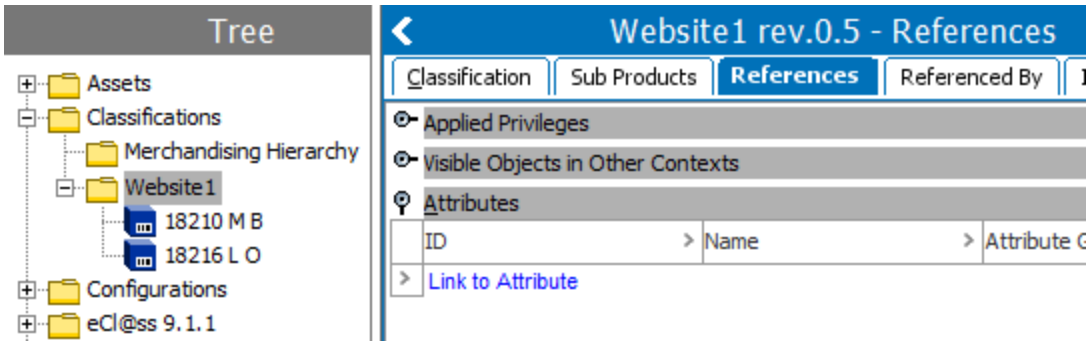
Creating a Classification Attribute Link

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.

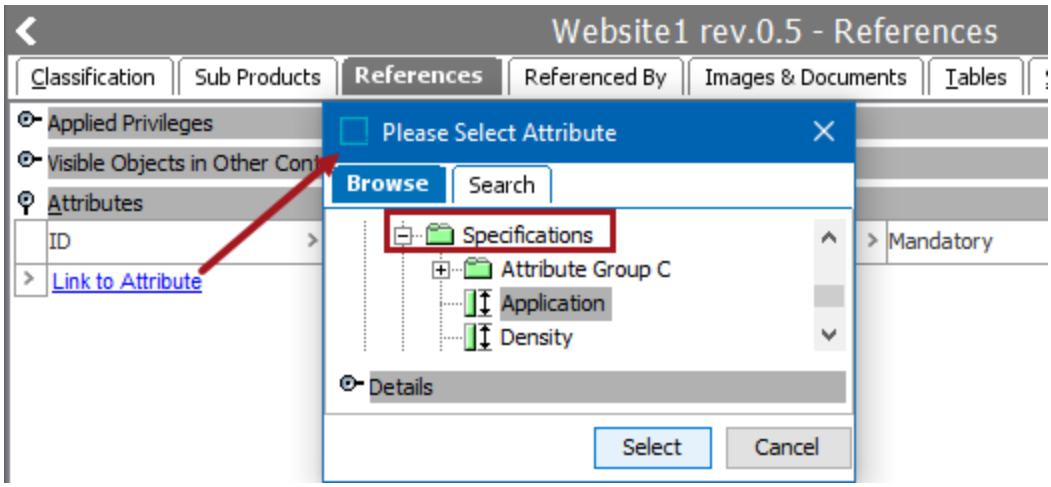
1. In System Setup, verify that the specification attribute to be linked has Validity that includes the products in the classification.



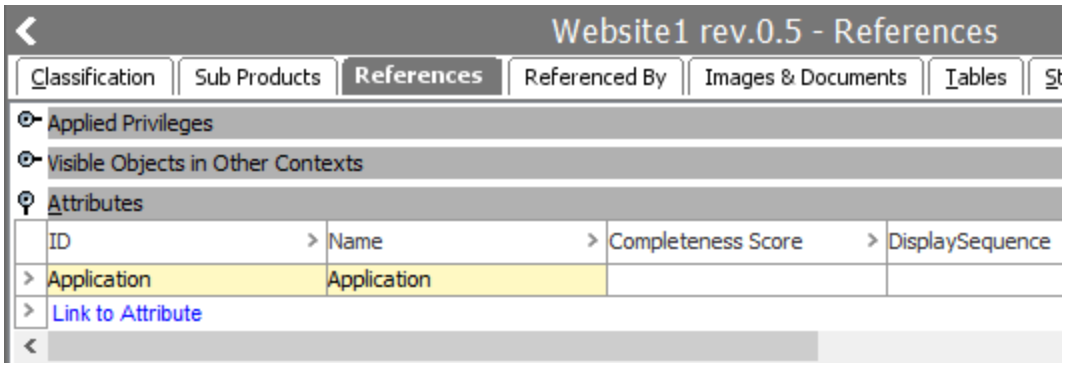
2. On Tree, select the classification that contains the products (via a Product to Classification Link Type) that should display the specification attribute.



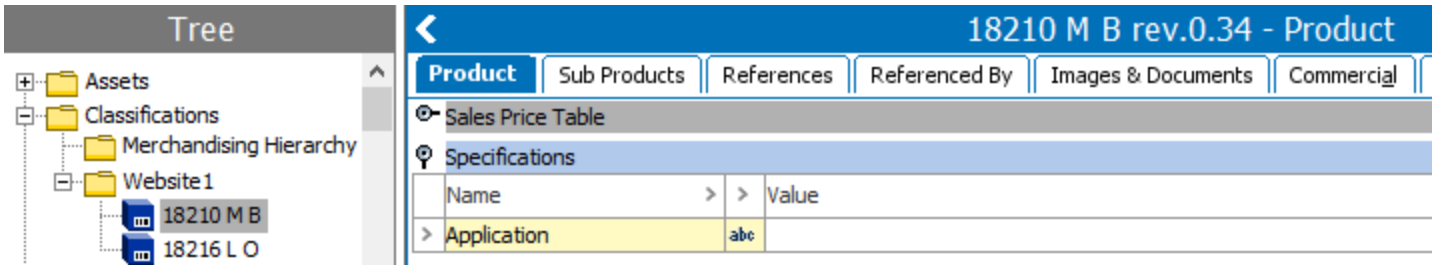
3. On the References tab > Attributes flipper> click the 'Link to Attribute' link to display the Select Attribute dialog.



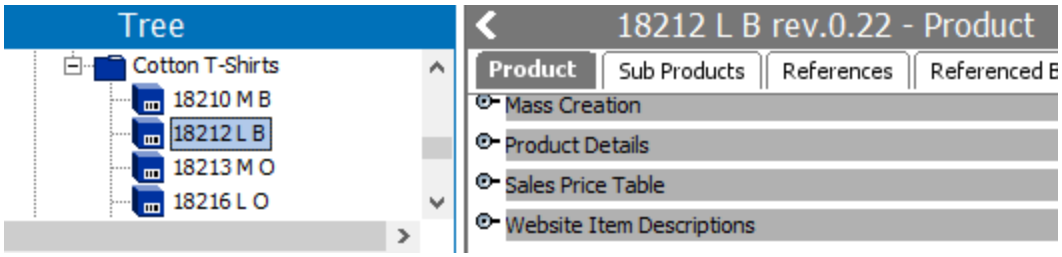
4. Note the attribute group of the desired specification attribute, choose the attribute, and click the Select button. The specification attribute is displayed under the Attributes flipper.



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

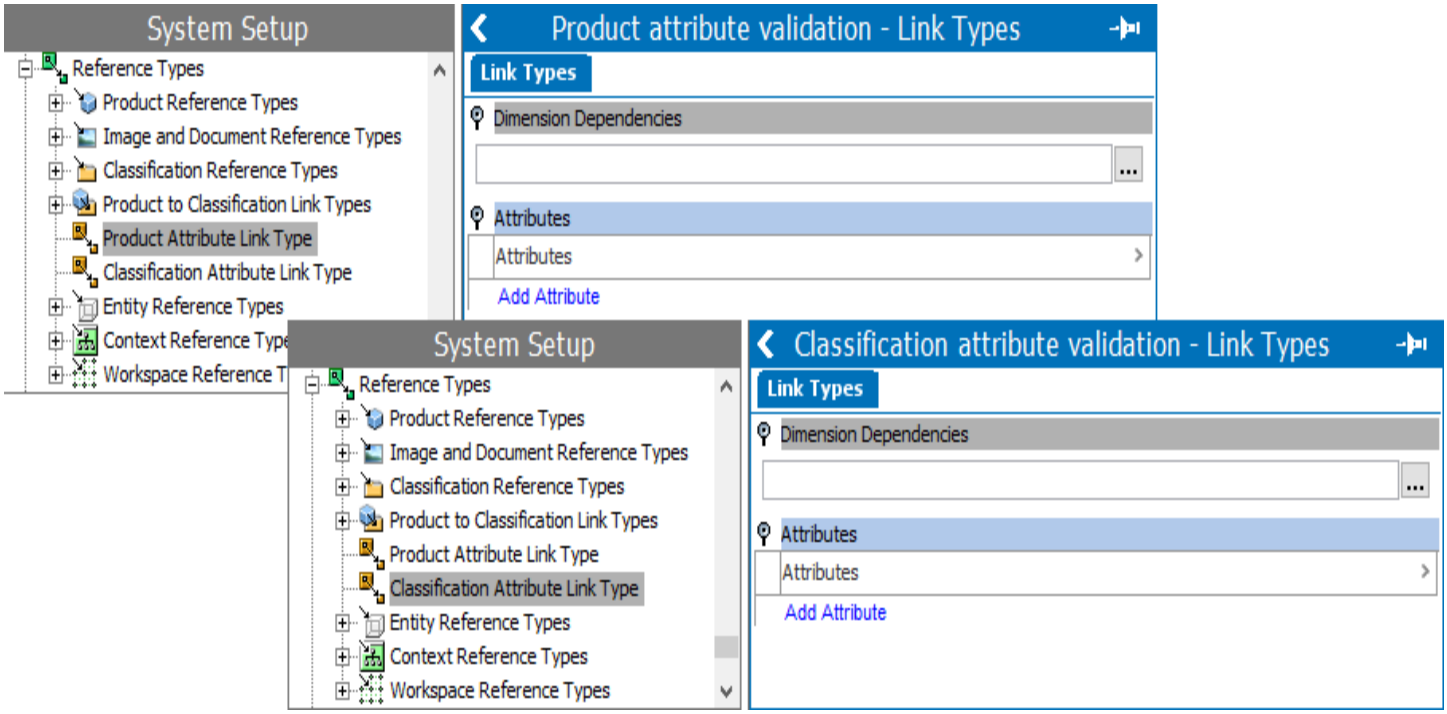
Setting Up an Attribute Link Type

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.

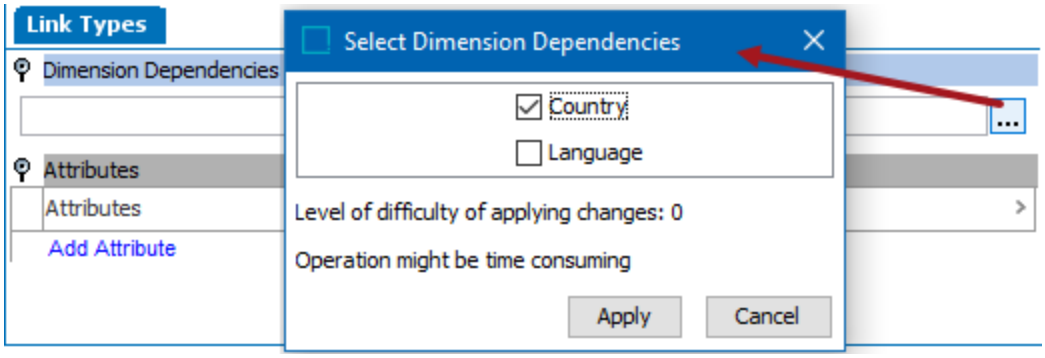
Only the default attribute link types can exist in STEP. Unlike references, user-created types are not allowed. The settings on the default attribute link types are applied to all 'Product Attribute Link Type' and 'Classification Attribute Link Type' used in the database.

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.

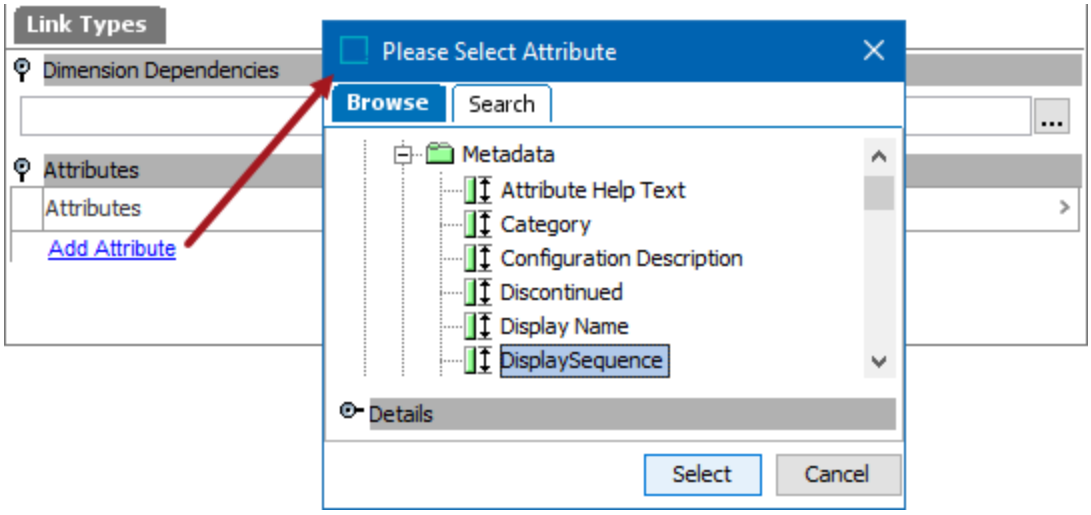


2. If required, set the dimension dependencies on the Link Types tab, under the Dimension Dependencies flipper. Only one dimension is recommended if any, due to the increased complexity of managing more than one. Setting an attribute link type to be dimension dependent will allow variations of values on the linked specification attribute across dimension points, and by extension, contexts.



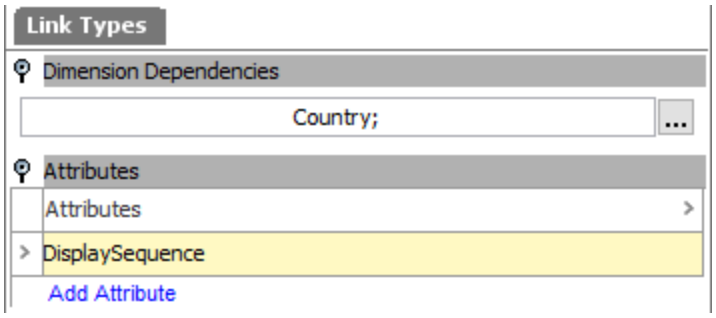
- Click the ellipsis button (...) to display the Select Dimension Dependencies dialog.
- Choose the required dimension(s).
- Click **Apply**.

3. If required, set metadata attributes on the Link Types tab, under the Attributes flipper. Specification attributes and groups cannot be added.

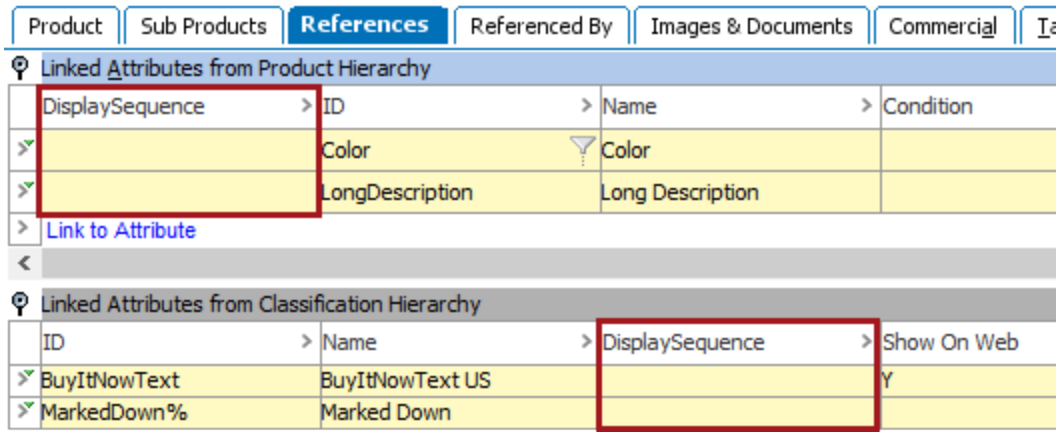


- Click the **Add Attribute** link to display the Select Attribute dialog.
- Browse or search for the required description attribute and choose it.
- Click **Select**.

4. The link type displays the selected dimension dependencies and metadata attributes.



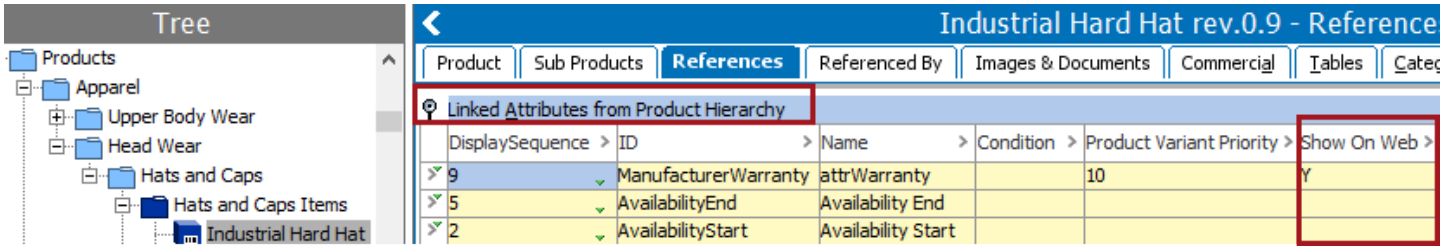
If the same metadata attribute is set for both attribute links, the product's References tab shows the attribute in both the 'Linked Attributes from Product Hierarchy' and the 'Linked Attributes from Classification Hierarchy' flippers.



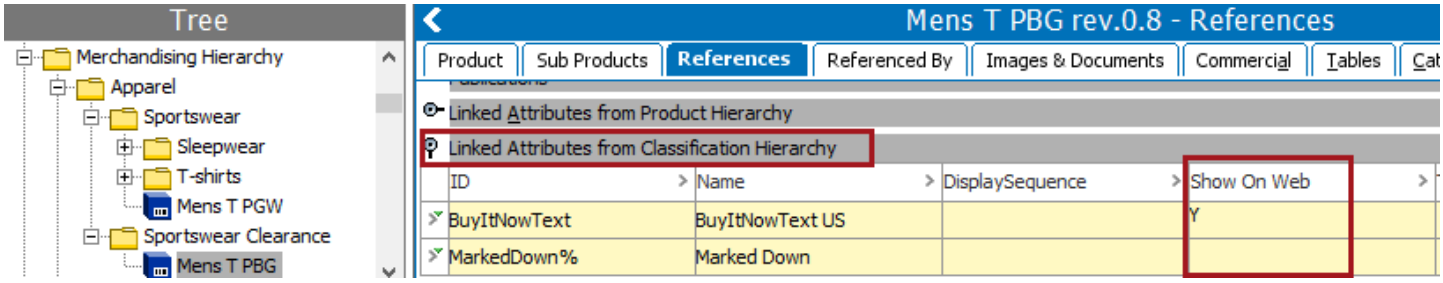
5. Add metadata values as described in **Maintaining Attribute Link Metadata Values**.

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.



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.



To add a new metadata attribute for an attribute link type, see the **Setting Up an Attribute Link Type**.

As defined in the sections below, values can be added for a metadata attribute on an attribute link type and can be:

- Inherited
- Written locally
- Made mandatory

Inherited Metadata Values

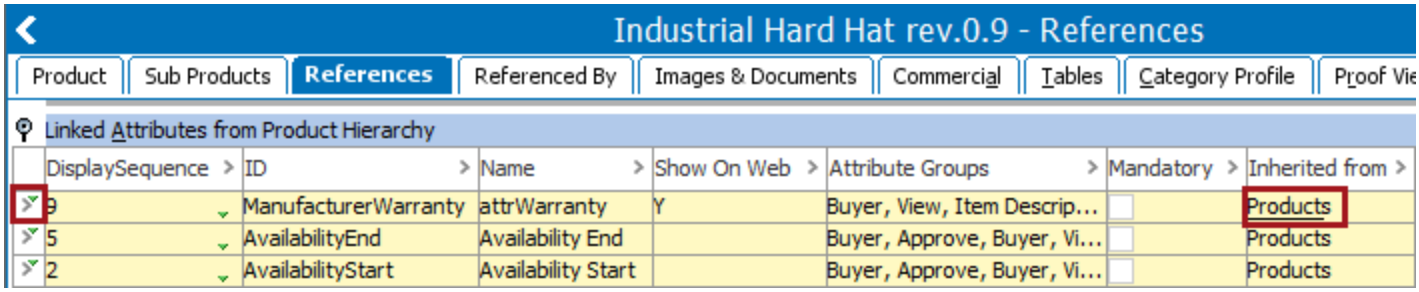
A value being inherited is indicated by the small green arrow ( ) in the first column of the list. Inherited values must be edited at the parent level, and are displayed as read-only for all children.

Use inheritance when the value is the same from all or most of the products from the 'Inherited from' level and below.

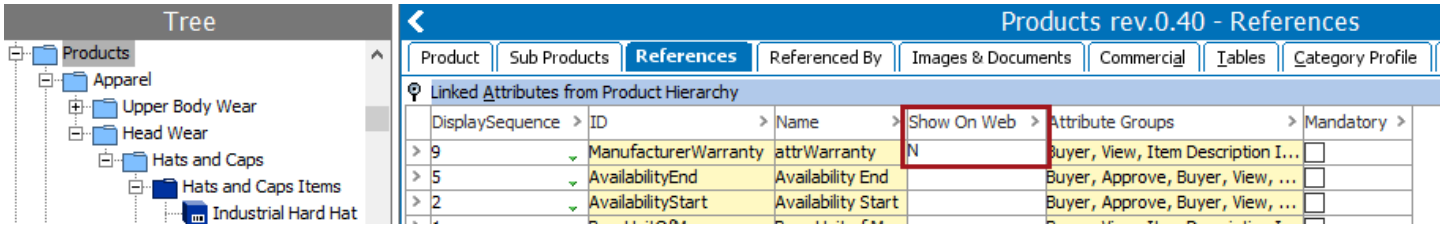
Edit an Inherited Value

Note: Although the following images show the Product Attribute Link, the process is the same for Classification Attribute Link maintenance.

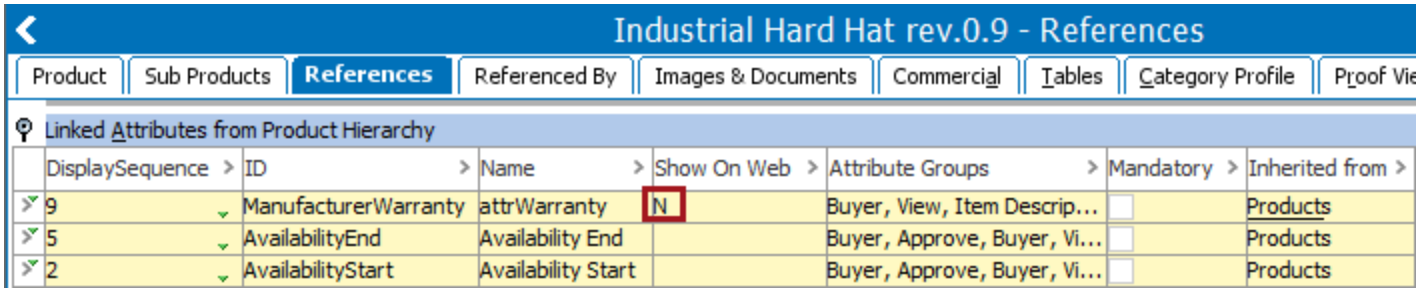
1. Click the link in the 'Inherited from' column to display the parent object where the value is being held.



2. Edit the value on the parent object. Also note that the 'Inherited from' column is not displayed for a parent object.



All children display the new value.



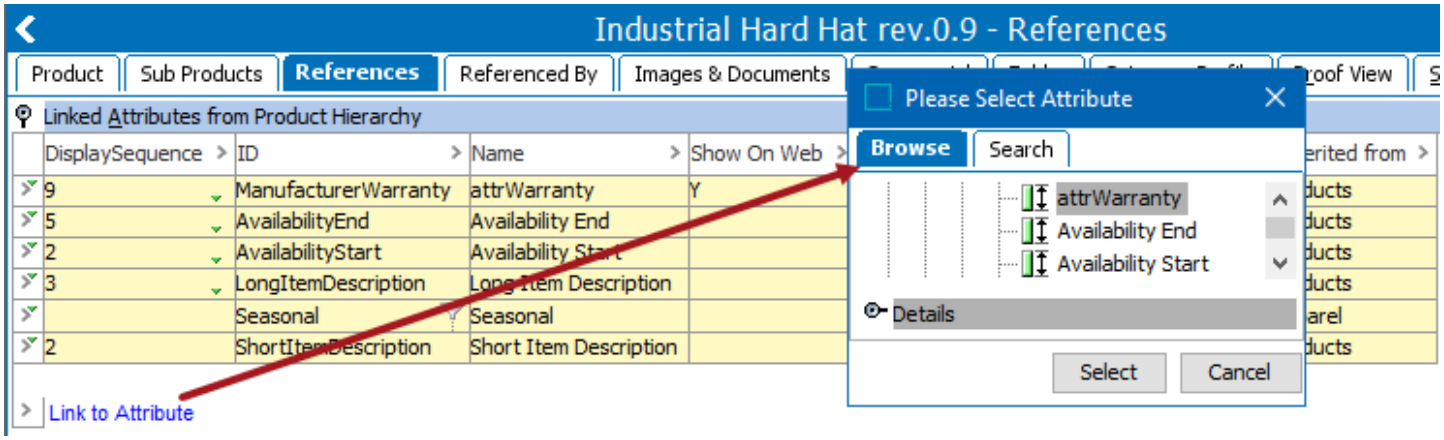
Local Metadata Values

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.

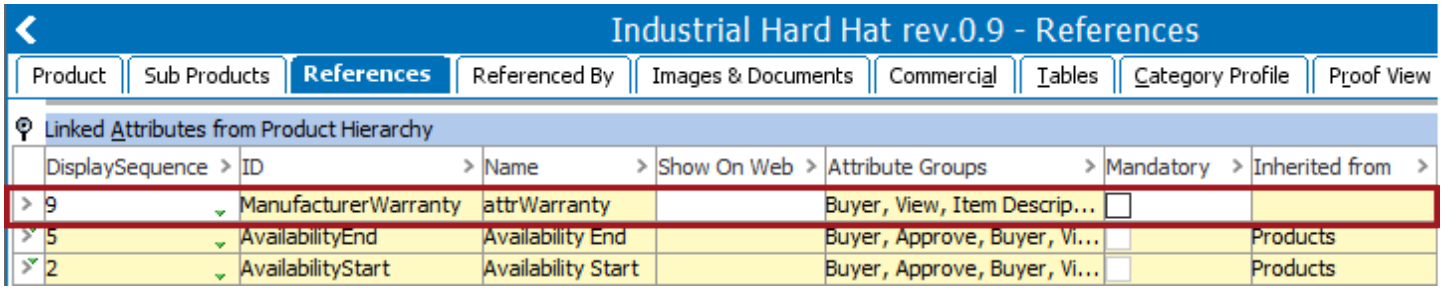
Add a Local Value

Note: Although the following images show the Product Attribute Link, the process is the same for Classification Attribute Link maintenance.

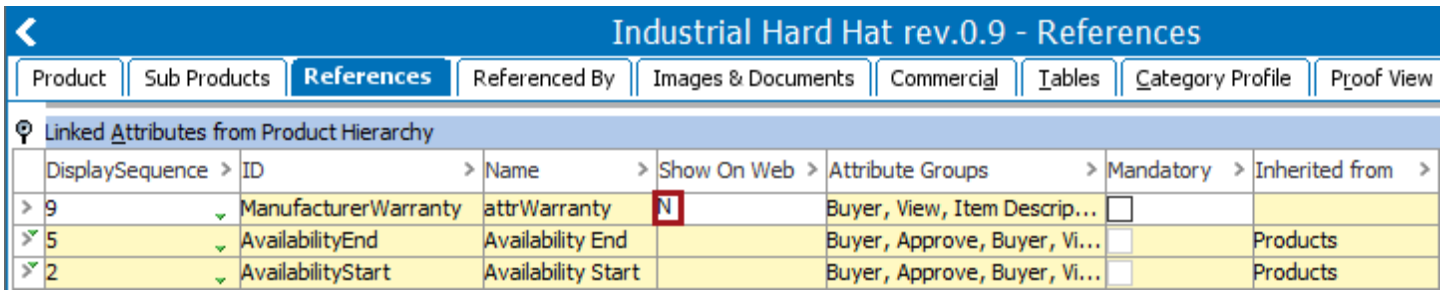
1. On the product References or Referenced By tabs, open the 'Linked Attributes from Product Hierarchy' (or 'Linked Attributes from Classification Hierarchy') flipper.
2. Click the **Link to Attribute** link to display the Select Attribute dialog.
3. Choose a specification attribute (description attributes are not allowed) and click the **Select** button.



4. Now that the attribute can be edited, add a local value.



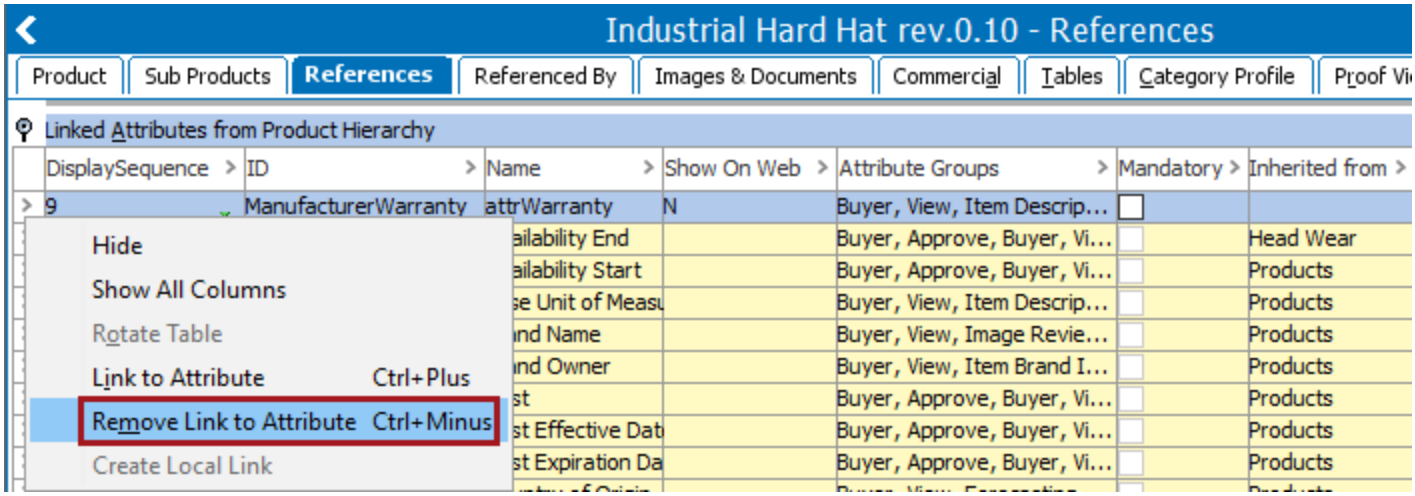
The local value shows at the level where it is added, and at any lower levels, via inheritance.



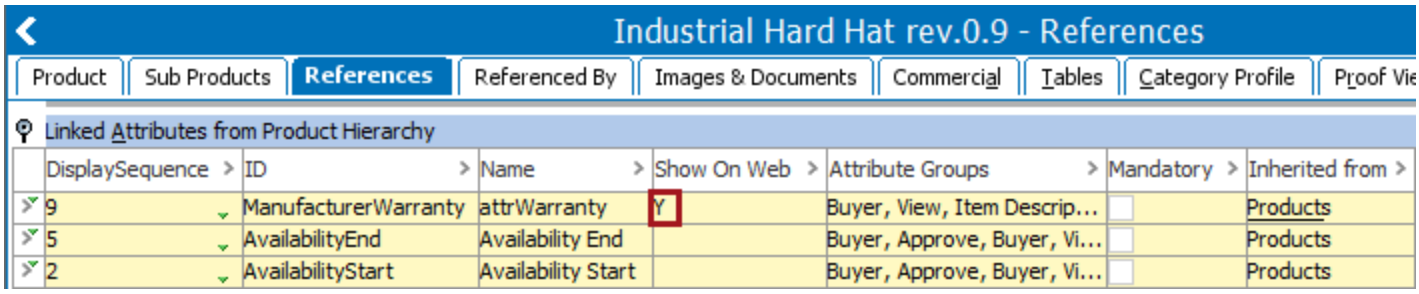
Restore an Inherited Value

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.



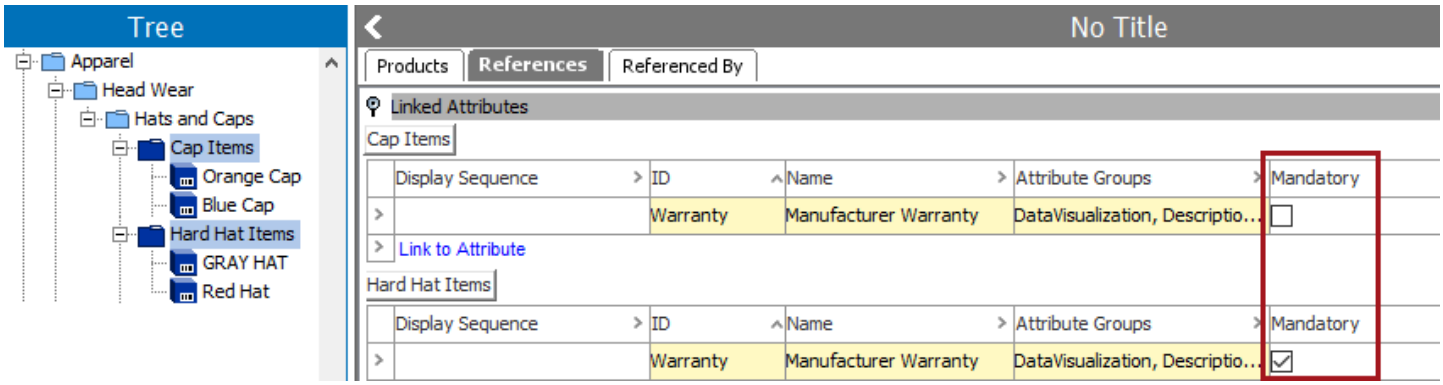
The inherited value is displayed again.



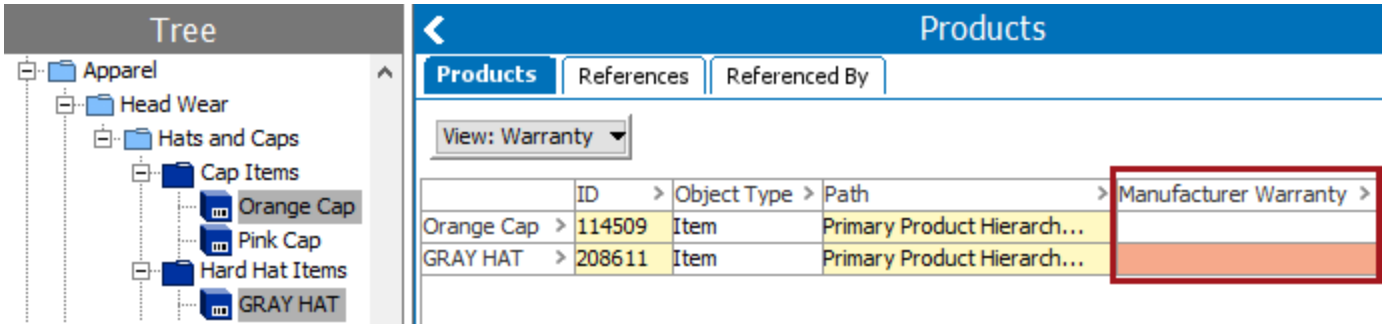
Mandatory Linked Attribute Values

The mandatory checkbox allows the attribute value to be required at the level in the hierarchy where the link is created. This provides a way to have an attribute that is available for all objects of the valid type, while only making it mandatory when the object types reside in a particular location.

For example, the 'Manufacturer Warranty' attribute is required for hard hats, but is not necessary on ball caps.



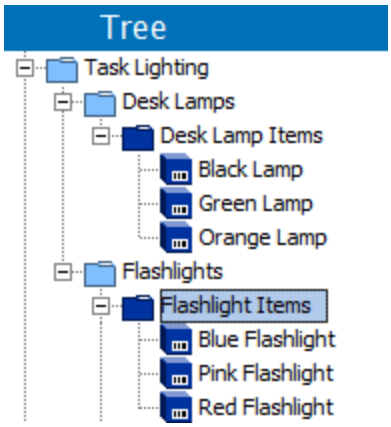
When viewing objects below the mandatory settings, one shows with a white background because it is not mandatory, and the other has a red background because it is mandatory.



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.

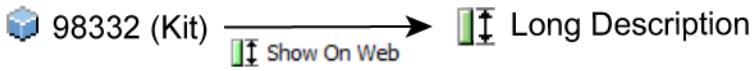
For example, if the language dependency is selected for the Product Attribute Link, the attribute 'Long Description' can be written in English for the English US context and written in French for the French FR context.

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**.

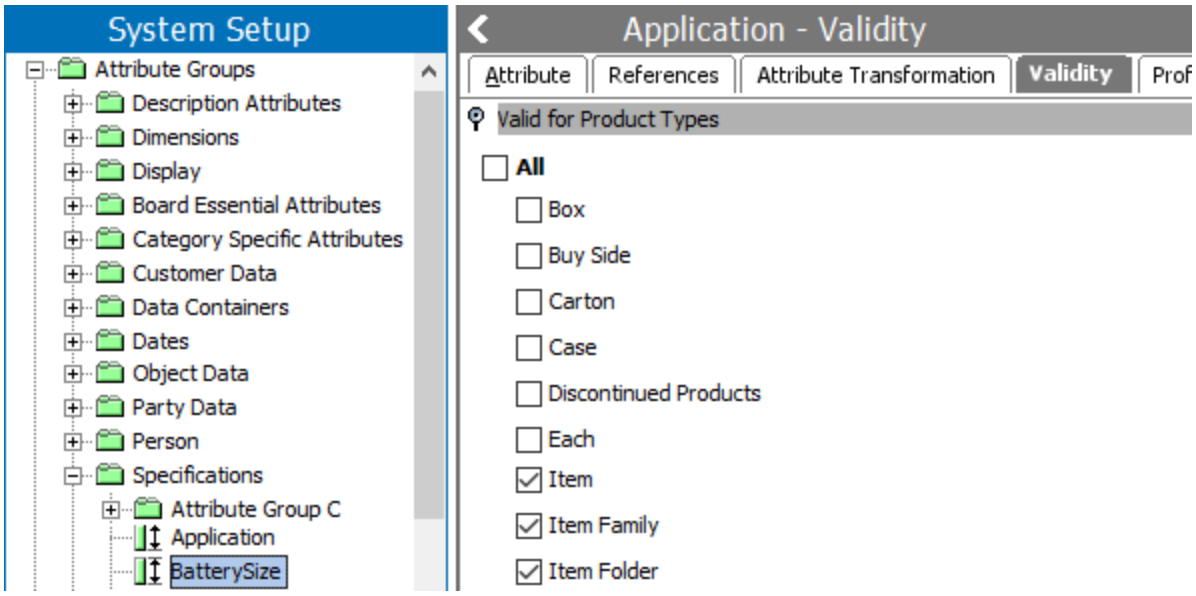
For information on creating a product attribute link type, see **Setting Up an Attribute Link Type**.

For information on creating a product attribute link, see **Creating a Product Attribute Link**.

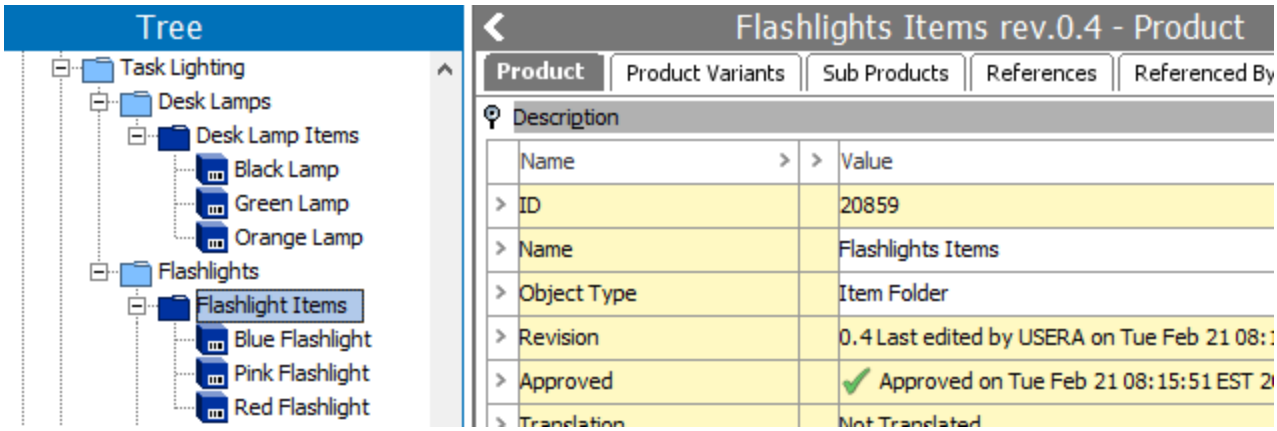
Creating a Product Attribute Link

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).

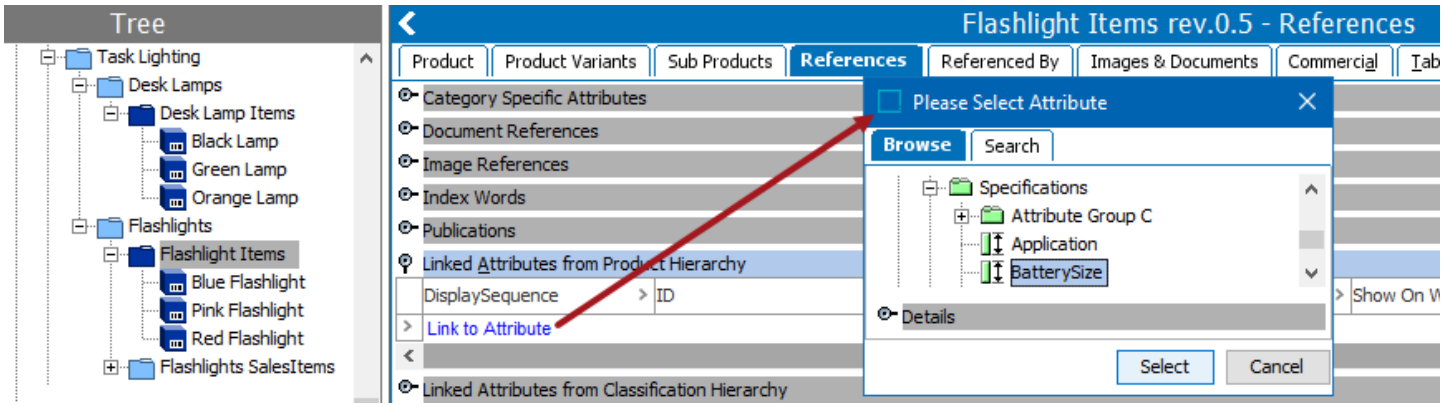
1. In System Setup, verify that the specification attribute to be linked has Validity that includes the objects in the intended levels of the PPH.



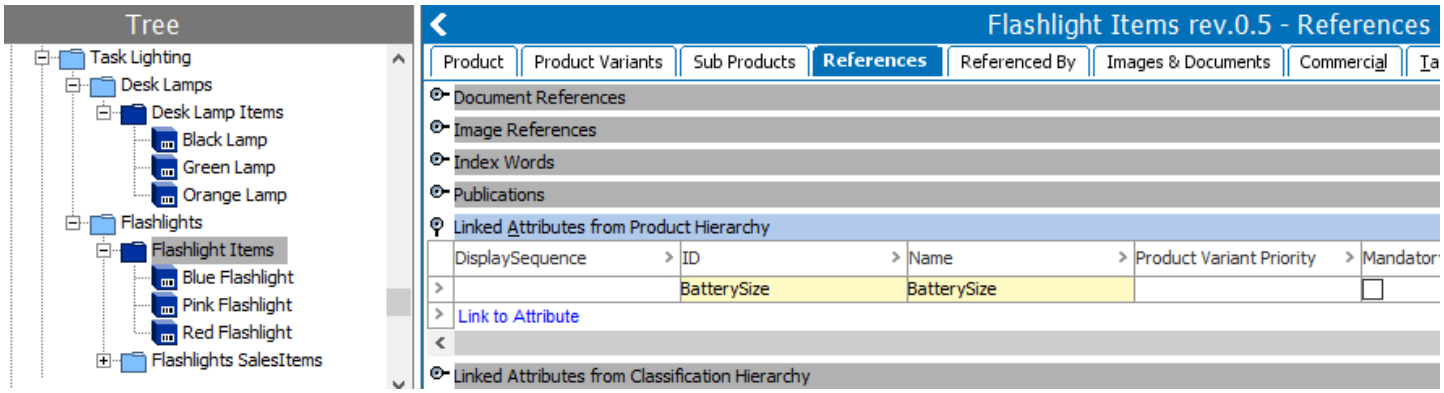
2. On Tree, select the PPH that should display the specification attribute.



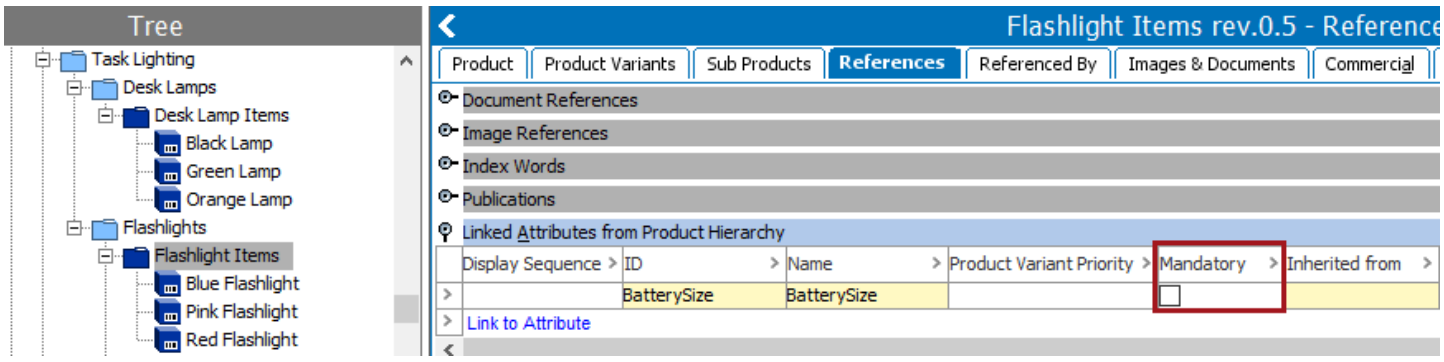
3. On the References or Referenced By tabs > 'Linked Attributes from Product Hierarchy' flipper> click the 'Link to Attribute' link to display the Select Attribute dialog.



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.



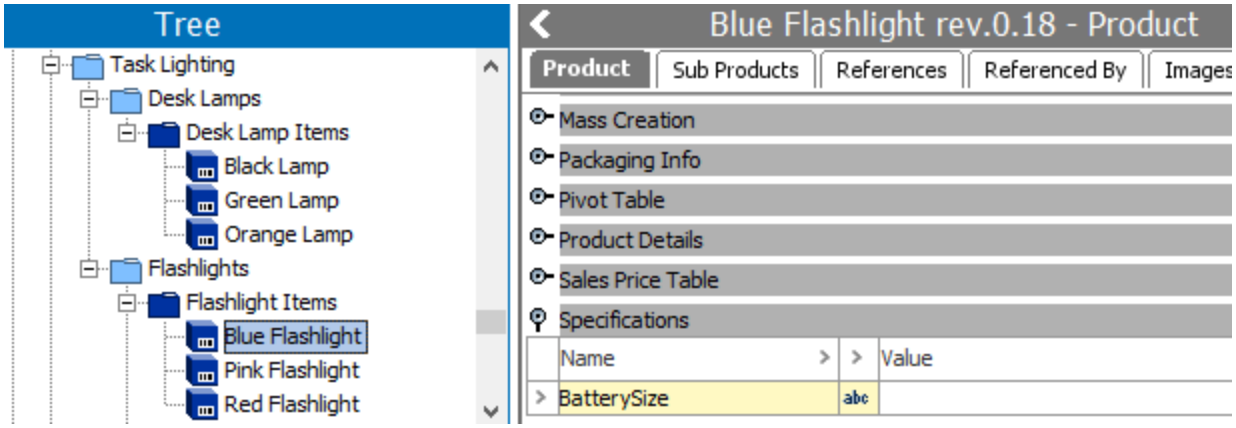
5. Optionally, scroll to the right to display the Mandatory column. If the attribute value is required at the linked location, check the checkbox.



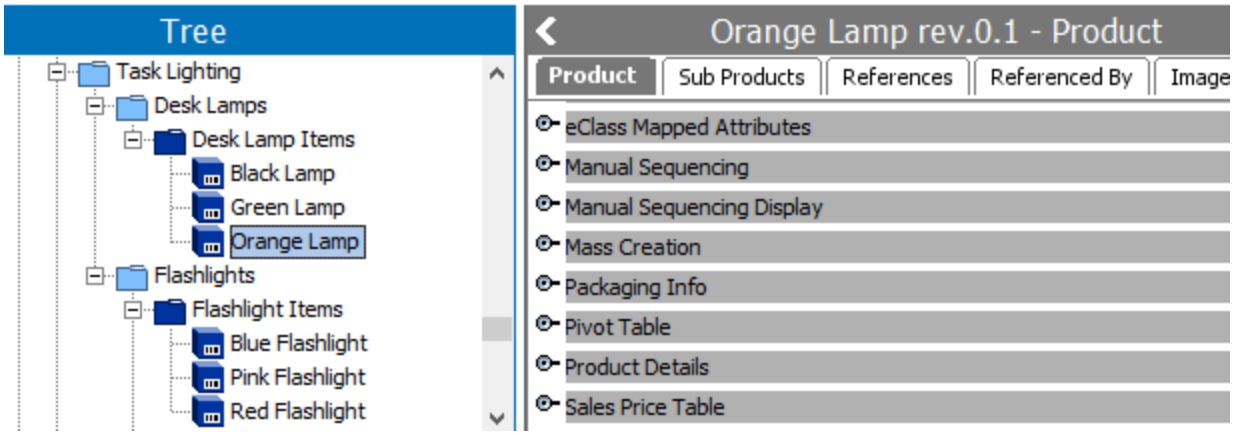
For more information on Mandatory settings, see the **Mandatory Linked Attribute Values** section of the **Maintaining Attribute Link Metadata Values** topic in this guide.

6. Select an object in the linked PPH level, open the attribute group flipper identified previously, and verify that the linked attribute is available. The mandatory visual indicator is also present if the Mandatory checkbox is

checked.



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).



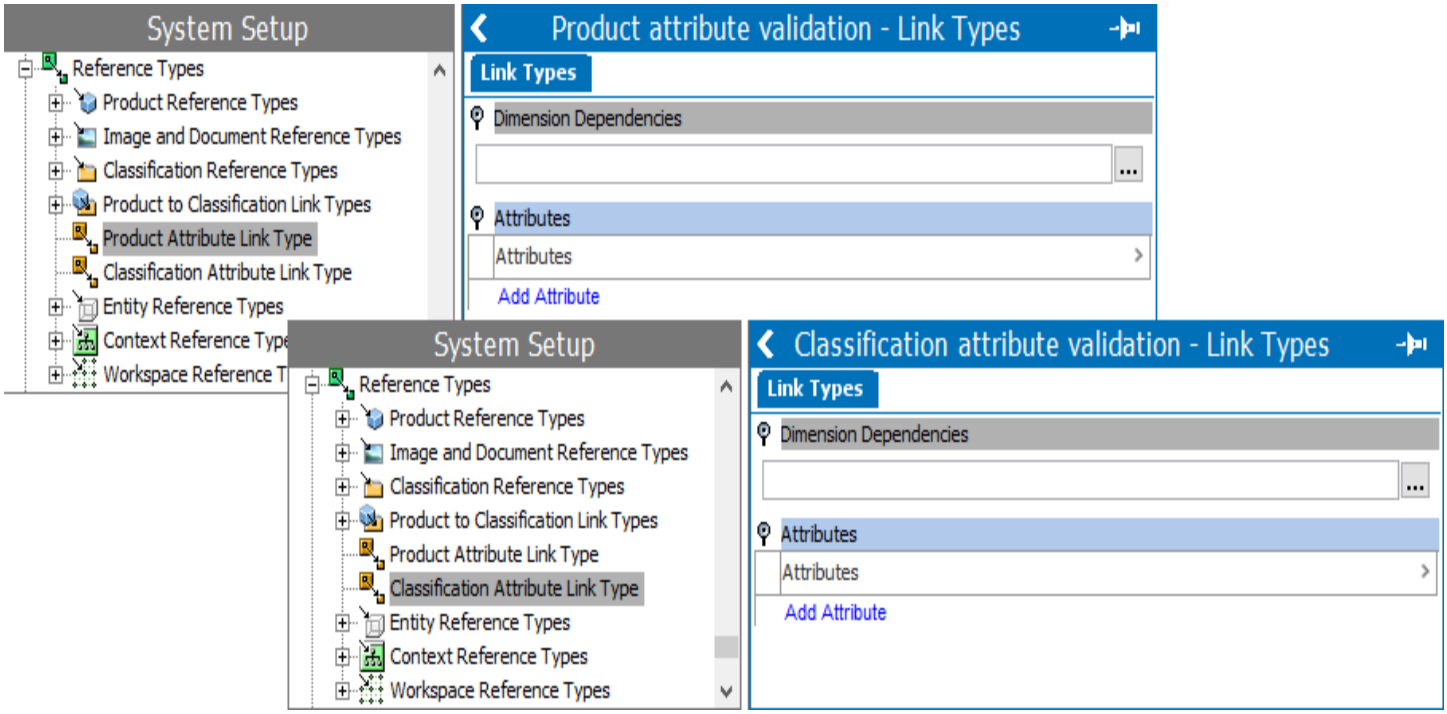
Setting Up an Attribute Link Type

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.

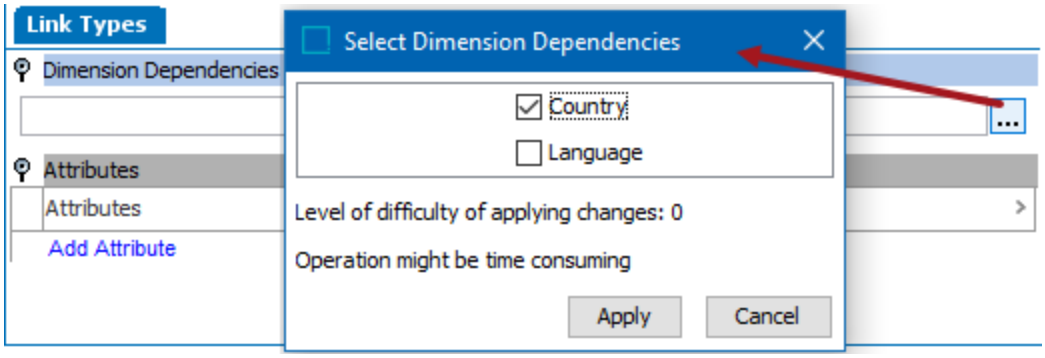
Only the default attribute link types can exist in STEP. Unlike references, user-created types are not allowed. The settings on the default attribute link types are applied to all 'Product Attribute Link Type' and 'Classification Attribute Link Type' used in the database.

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.

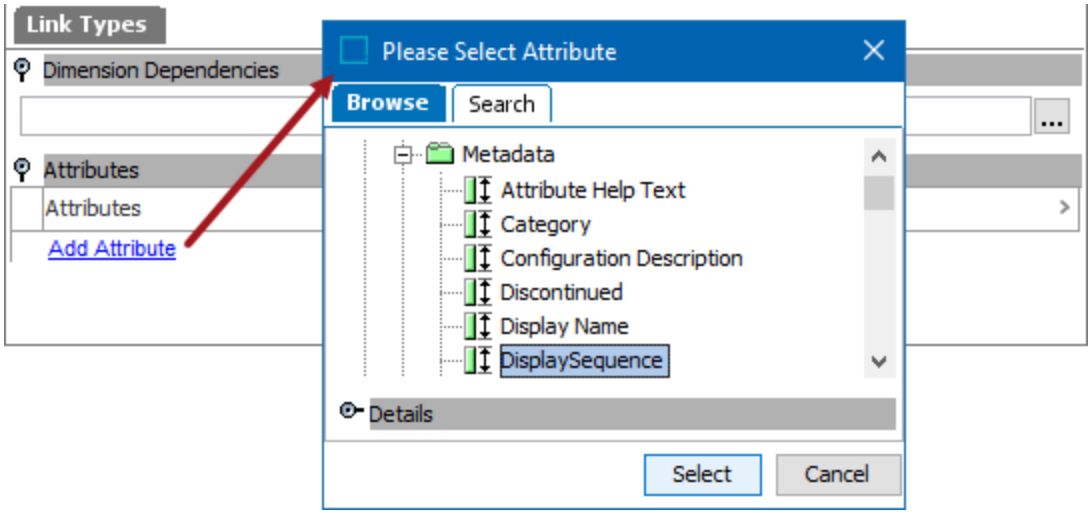


2. If required, set the dimension dependencies on the Link Types tab, under the Dimension Dependencies flipper. Only one dimension is recommended if any, due to the increased complexity of managing more than one. Setting an attribute link type to be dimension dependent will allow variations of values on the linked specification attribute across dimension points, and by extension, contexts.



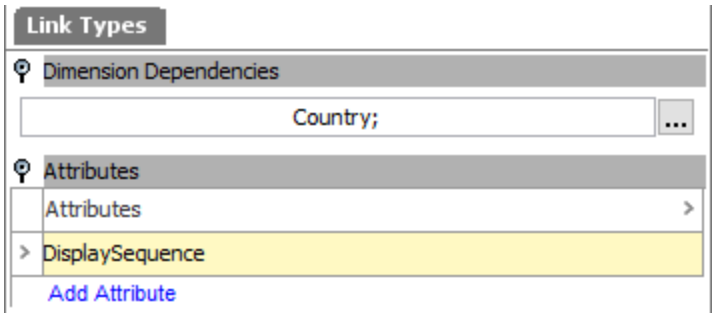
- Click the ellipsis button (...) to display the Select Dimension Dependencies dialog.
- Choose the required dimension(s).
- Click **Apply**.

3. If required, set metadata attributes on the Link Types tab, under the Attributes flipper. Specification attributes and groups cannot be added.

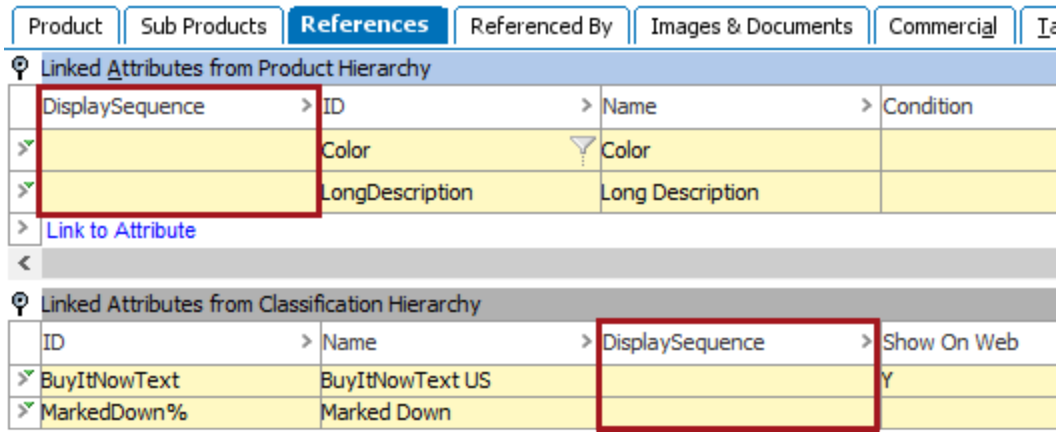


- Click the **Add Attribute** link to display the Select Attribute dialog.
- Browse or search for the required description attribute and choose it.
- Click **Select**.

4. The link type displays the selected dimension dependencies and metadata attributes.



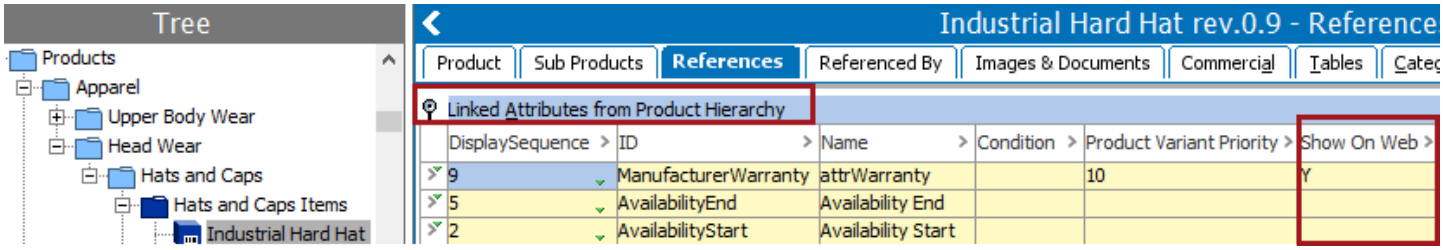
If the same metadata attribute is set for both attribute links, the product's References tab shows the attribute in both the 'Linked Attributes from Product Hierarchy' and the 'Linked Attributes from Classification Hierarchy' flippers.



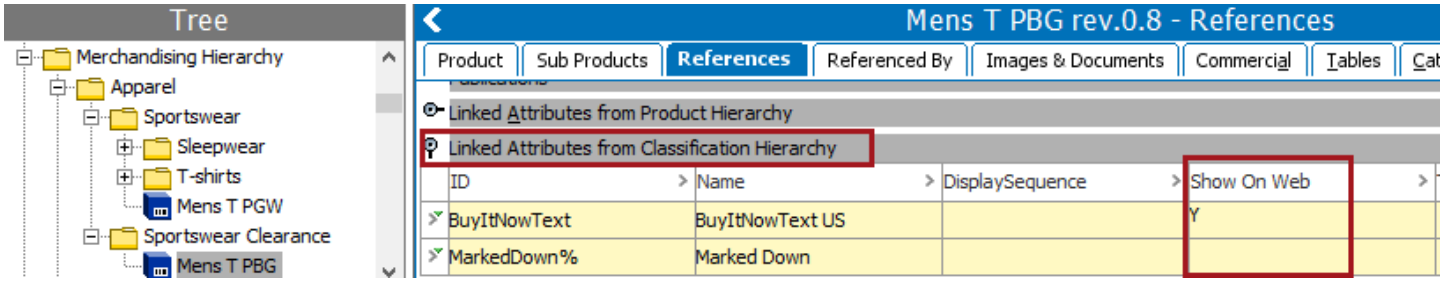
5. Add metadata values as described in **Maintaining Attribute Link Metadata Values**.

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.



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.



To add a new metadata attribute for an attribute link type, see the **Setting Up an Attribute Link Type**.

As defined in the sections below, values can be added for a metadata attribute on an attribute link type and can be:

- Inherited
- Written locally
- Made mandatory

Inherited Metadata Values

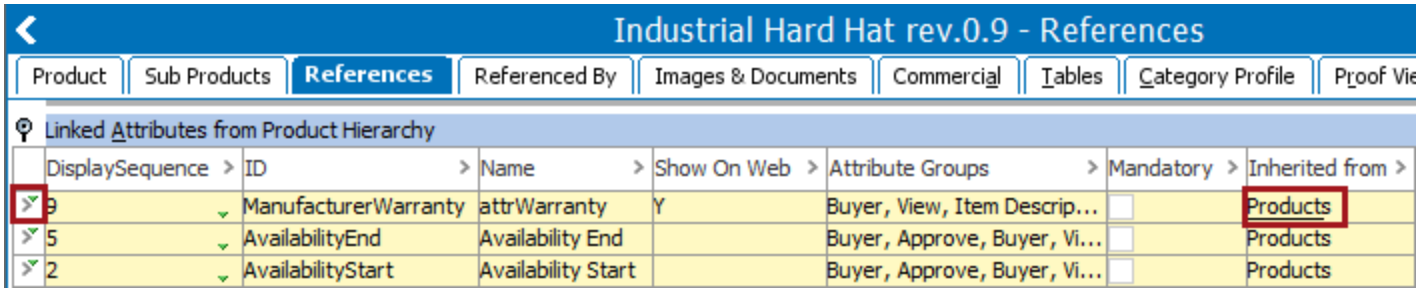
A value being inherited is indicated by the small green arrow ( ) in the first column of the list. Inherited values must be edited at the parent level, and are displayed as read-only for all children.

Use inheritance when the value is the same from all or most of the products from the 'Inherited from' level and below.

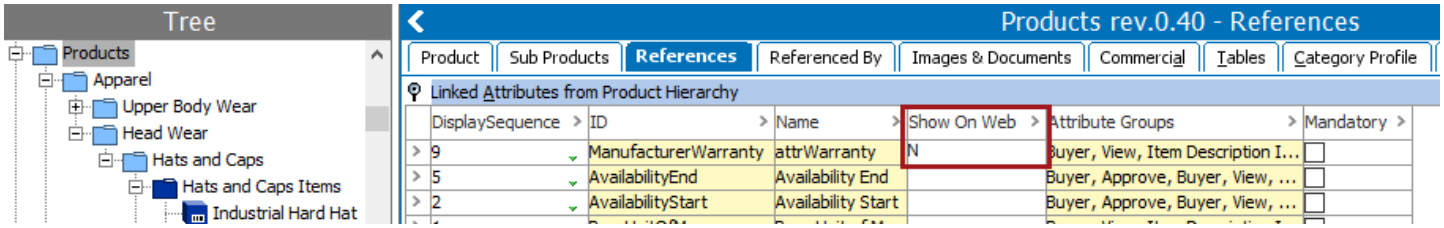
Edit an Inherited Value

Note: Although the following images show the Product Attribute Link, the process is the same for Classification Attribute Link maintenance.

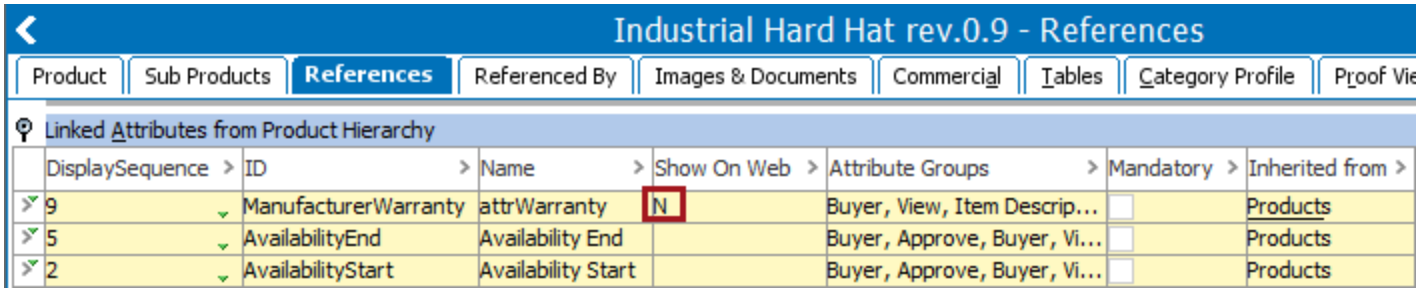
1. Click the link in the 'Inherited from' column to display the parent object where the value is being held.



2. Edit the value on the parent object. Also note that the 'Inherited from' column is not displayed for a parent object.



All children display the new value.



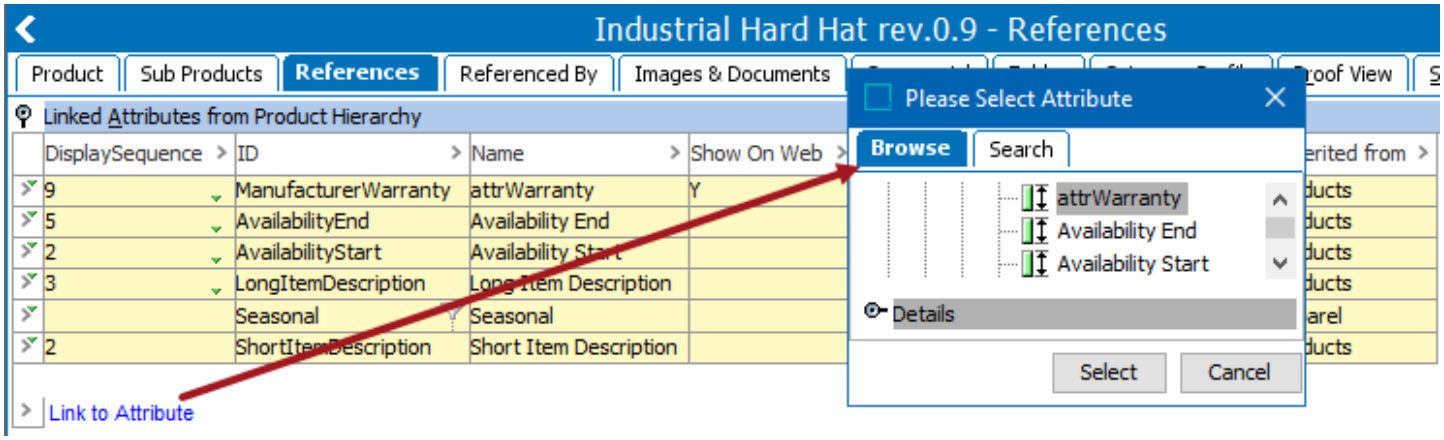
Local Metadata Values

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.

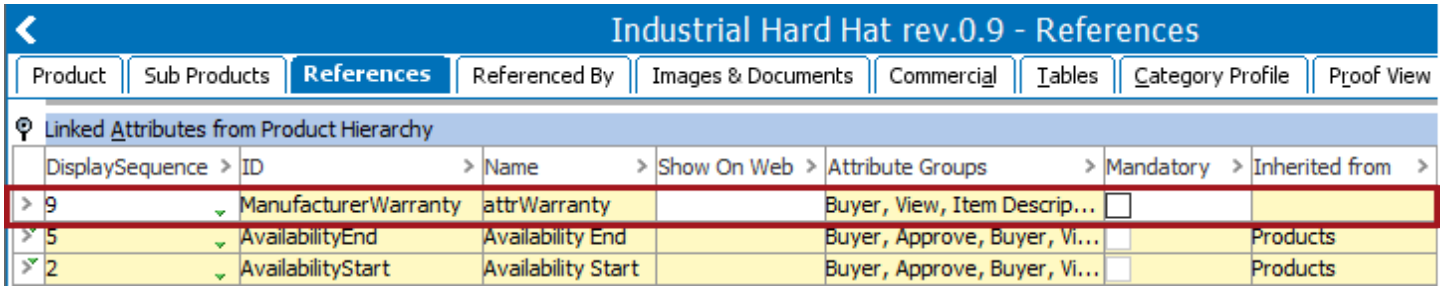
Add a Local Value

Note: Although the following images show the Product Attribute Link, the process is the same for Classification Attribute Link maintenance.

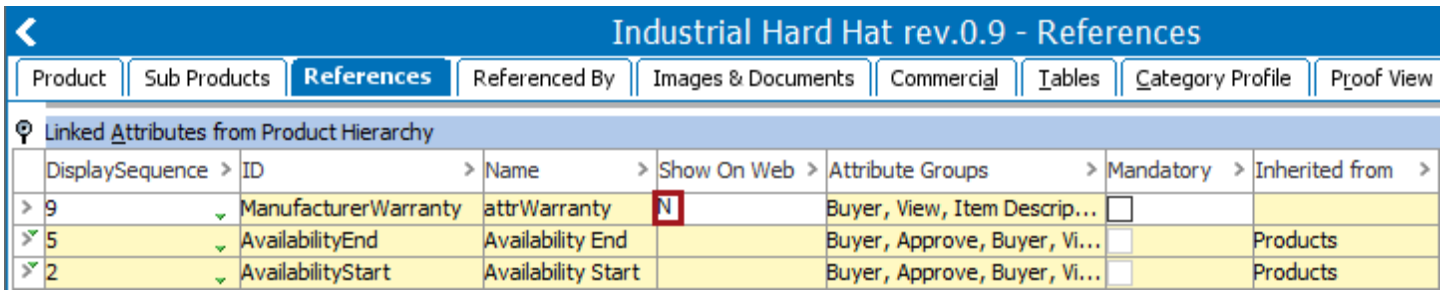
1. On the product References or Referenced By tabs, open the 'Linked Attributes from Product Hierarchy' (or 'Linked Attributes from Classification Hierarchy') flipper.
2. Click the **Link to Attribute** link to display the Select Attribute dialog.
3. Choose a specification attribute (description attributes are not allowed) and click the **Select** button.



4. Now that the attribute can be edited, add a local value.



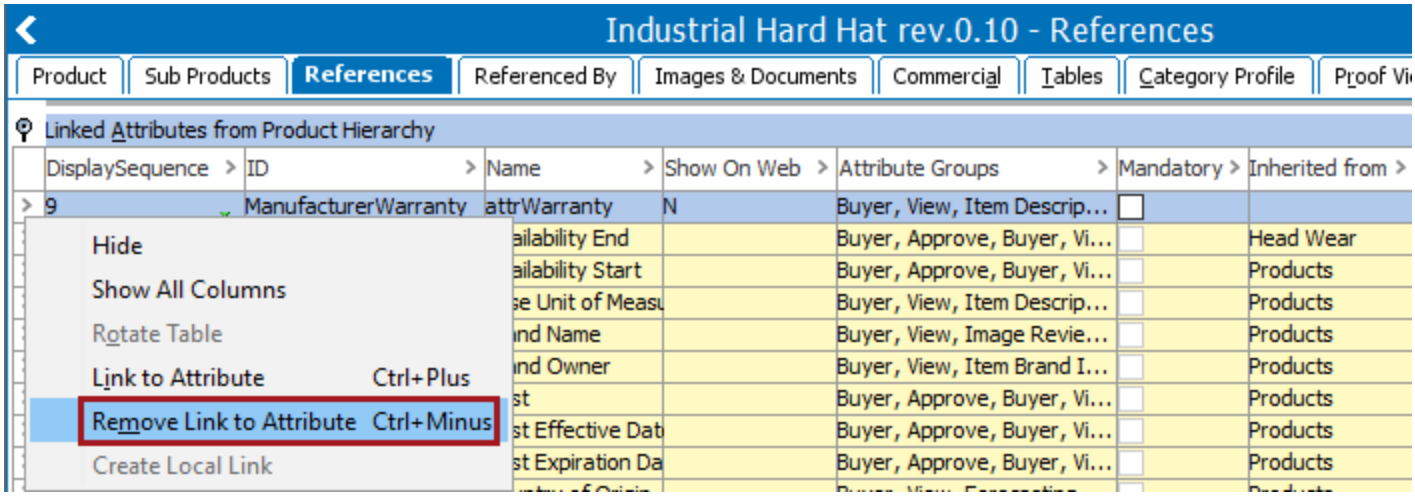
The local value shows at the level where it is added, and at any lower levels, via inheritance.



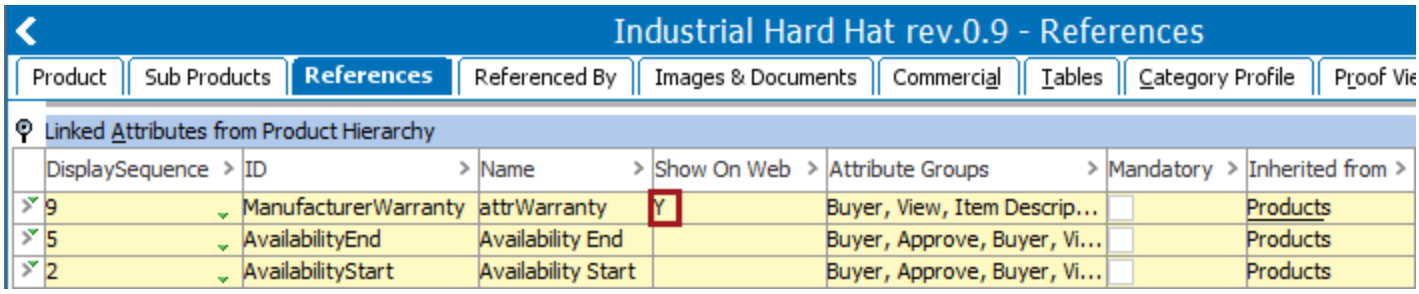
Restore an Inherited Value

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.



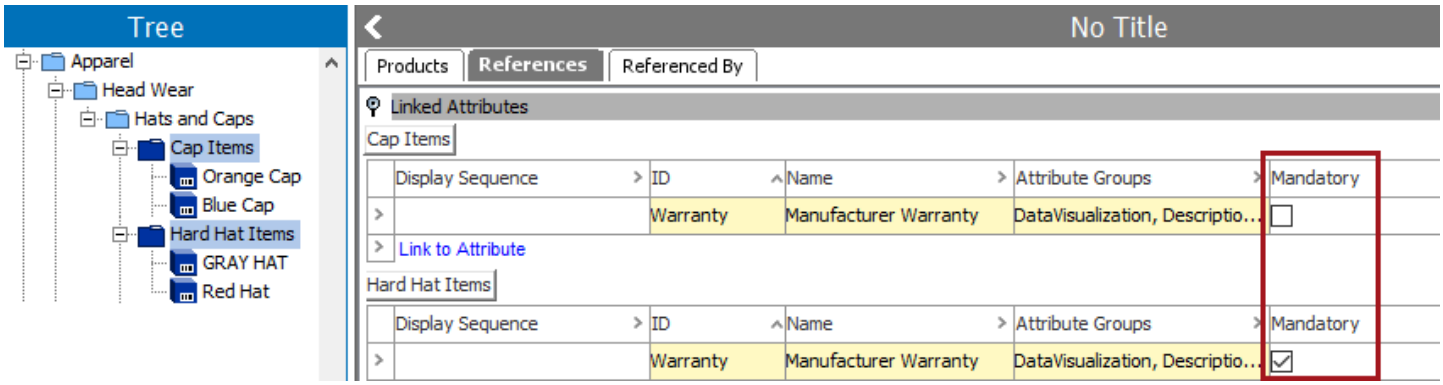
The inherited value is displayed again.



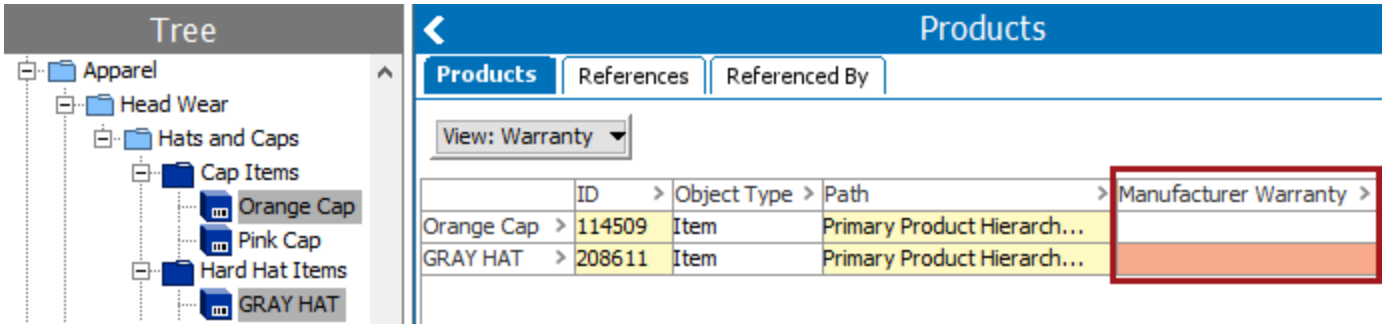
Mandatory Linked Attribute Values

The mandatory checkbox allows the attribute value to be required at the level in the hierarchy where the link is created. This provides a way to have an attribute that is available for all objects of the valid type, while only making it mandatory when the object types reside in a particular location.

For example, the 'Manufacturer Warranty' attribute is required for hard hats, but is not necessary on ball caps.



When viewing objects below the mandatory settings, one shows with a white background because it is not mandatory, and the other has a red background because it is mandatory.



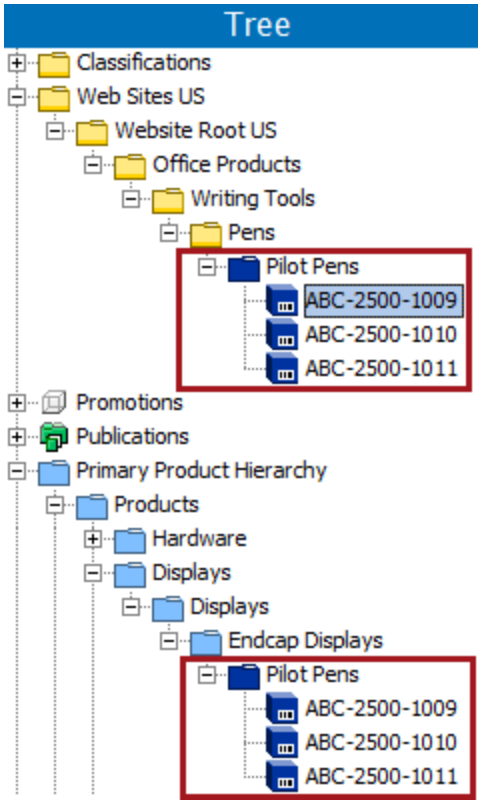
Product to Classification Link Types

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 tab.

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:

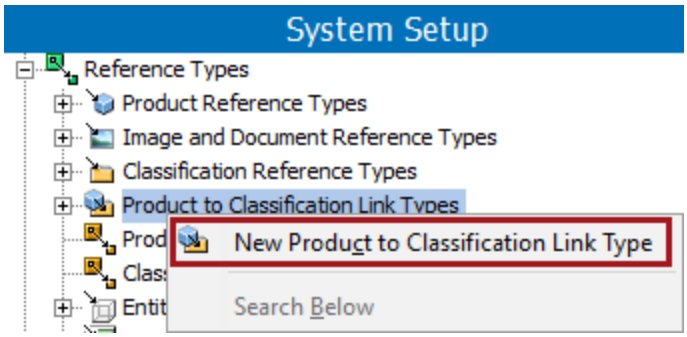
- Adding a link to an attribute group allows grouping of references and links, customized views, and added security via user group privileges. For more information, see **Attribute Groups for Reference and Link Types**.
- Dimension dependency determines if the link is displayed in a context, based on 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.

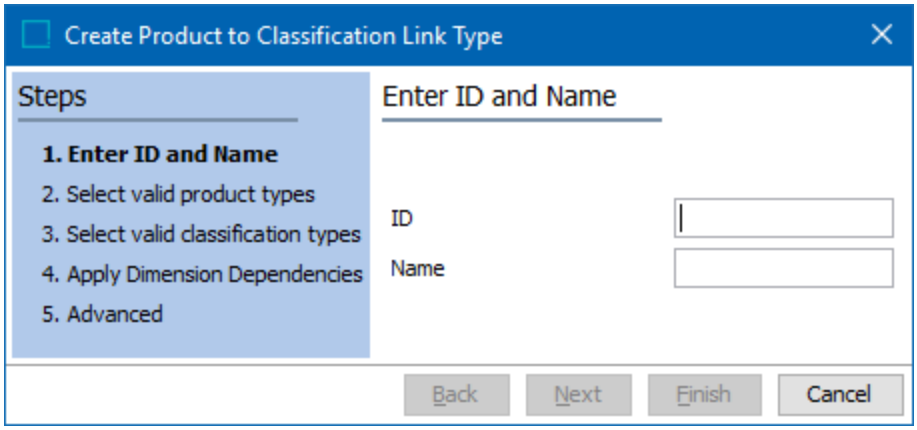
1. In System Setup > Reference Types > select **Product to Classification Link Type** > right-click, and choose the **New** option from the menu.



   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.

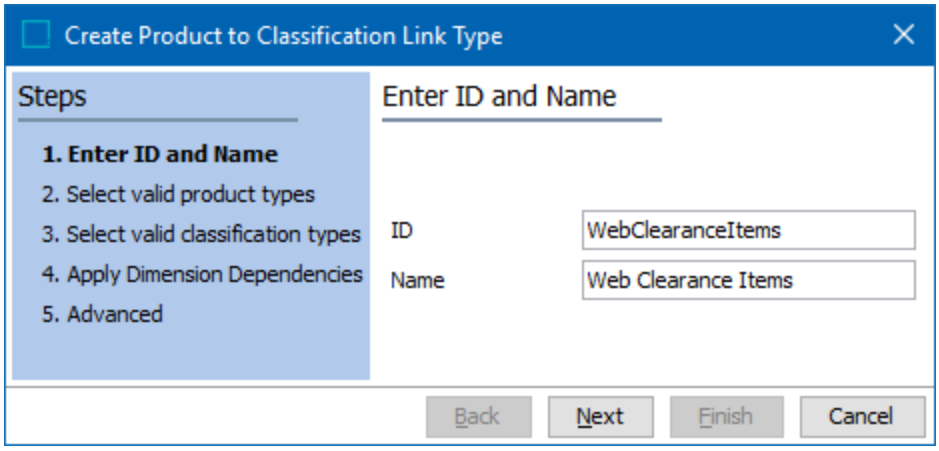
2. The Create Product to Classification Link Type wizard displays and includes the following steps:



- **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.

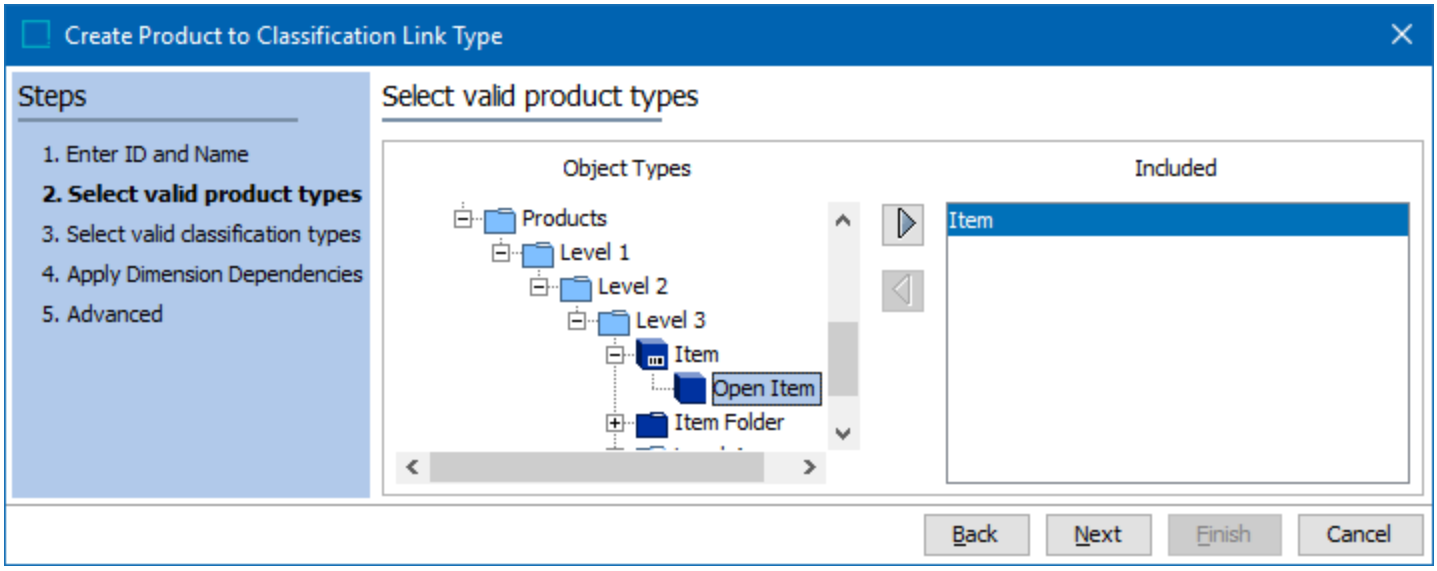
Once the reference type exists, create a reference between the two objects as outlined in **Creating a Product to Classification Link**.

Product to Classification Link Type - Enter ID and Name



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

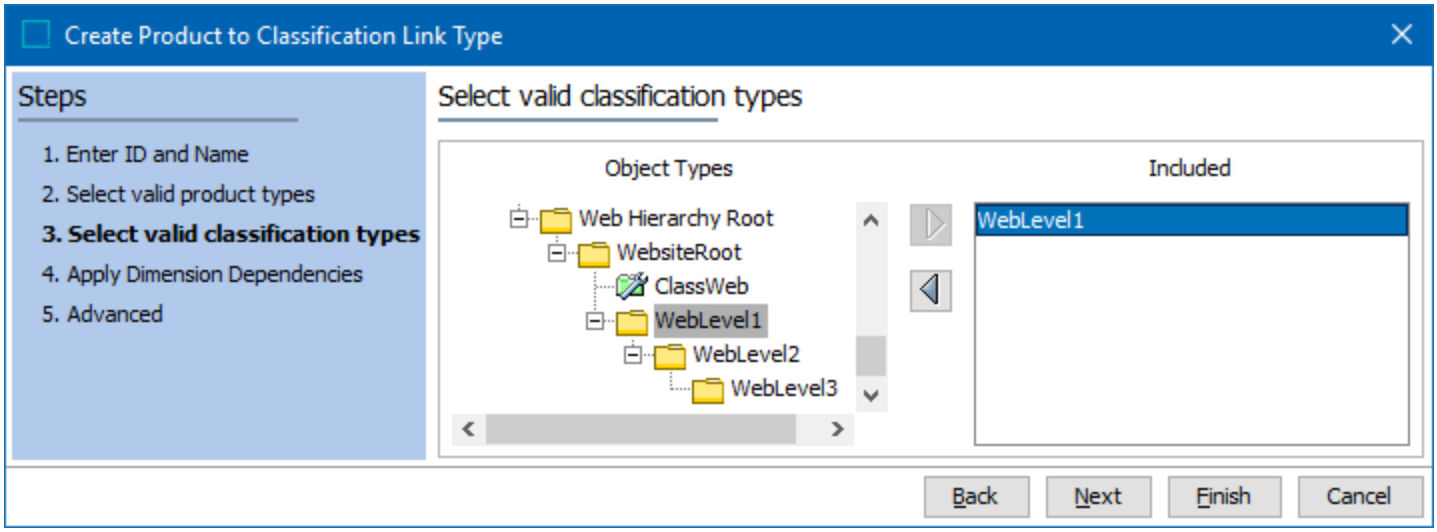


1. Use the displayed hierarchy to browse to and select one or more product object types of the reference. This is where the reference starts. For more information, see **Direction of a Reference**. Selecting an item from the Object Types pane enables the right arrow button.
2. Click the right arrow button ( ) 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 ( ).

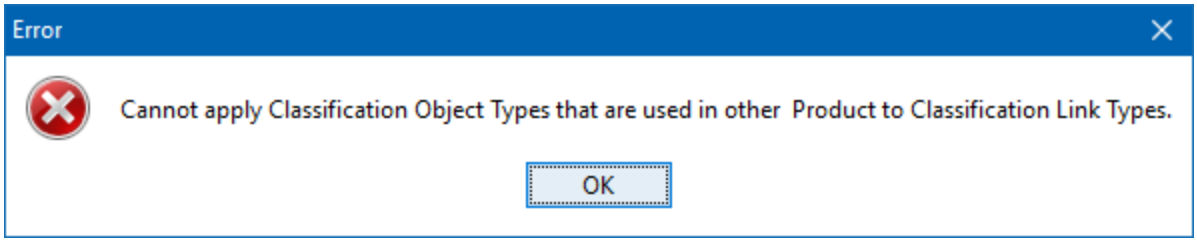
3. Click the **Next** button to display **Product to Classification Link Type - Select Valid Classification Types**.

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.

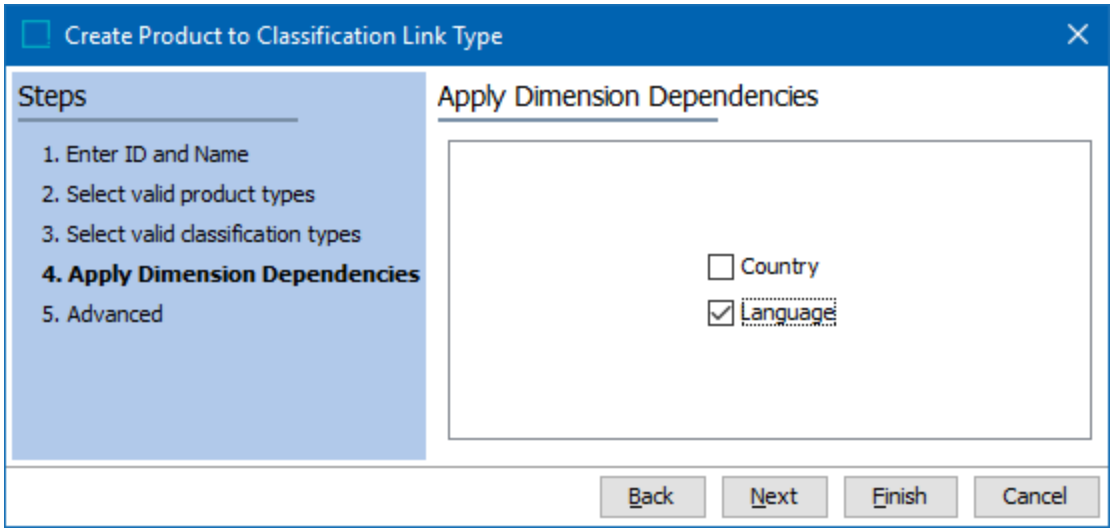


2. Click the right arrow button ( ) to move the selected object type to the Included pane. All objects in the Included pane will be valid for the Product to Classification Link type being created.

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

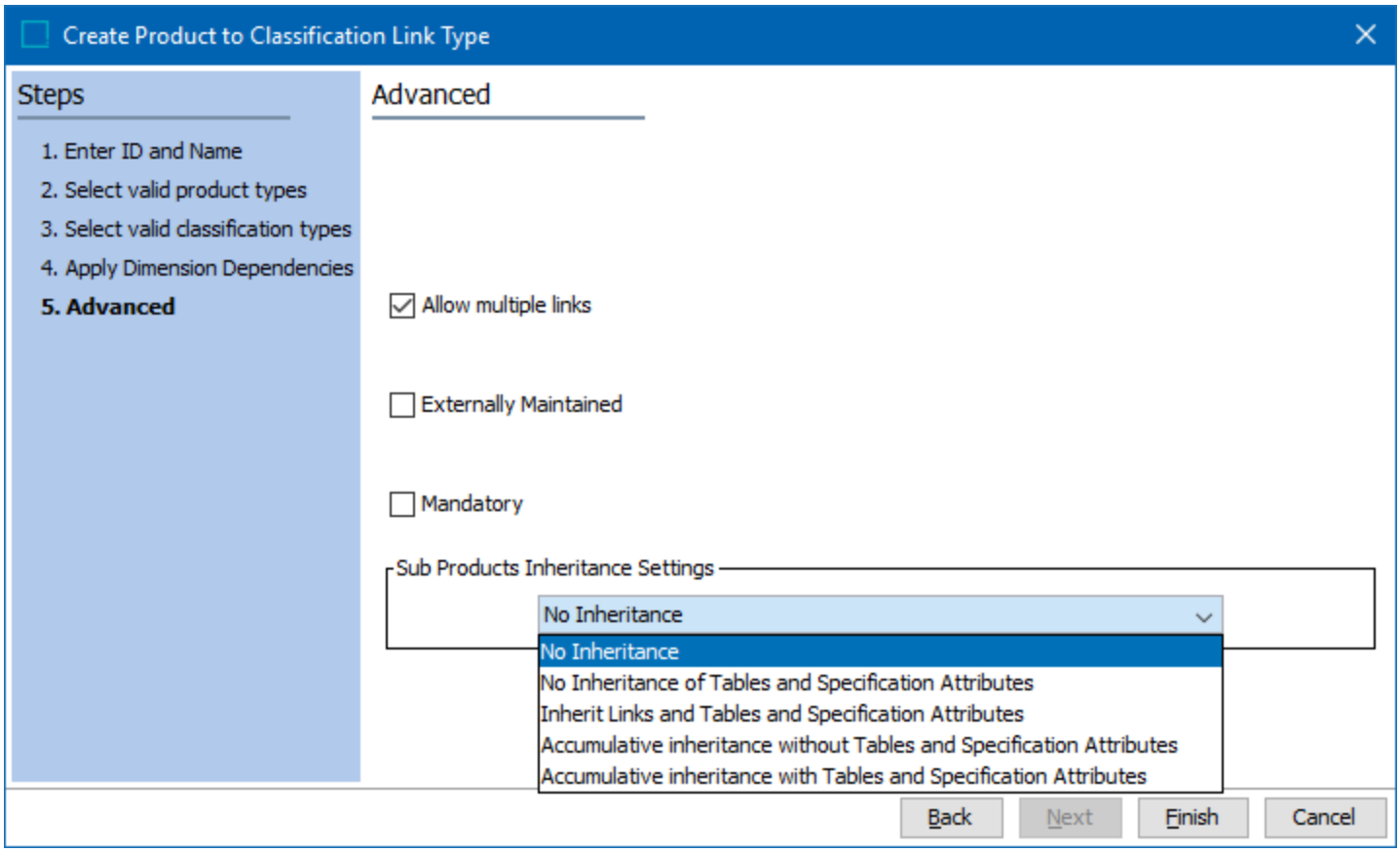


1. If the link type should be dimension dependent, select a dimension checkbox. Only one dimension is allowed for a link type. If multiple dimension dependencies are needed, create a separate link type.

   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



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.

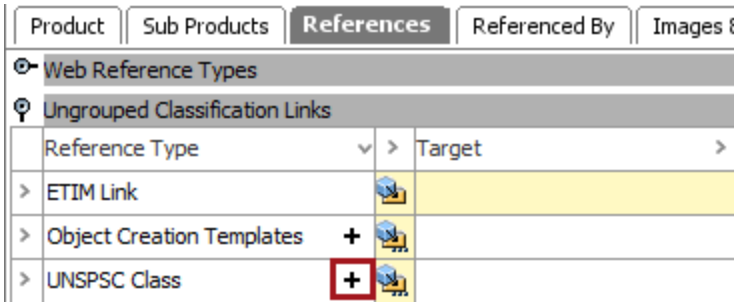
3. **Mandatory** determines if the link type is required for approval of the product and is indicated by the icon. The approval check for mandatory 'Product to Classification Link Type' is determined by the ownership of the link, which is set on the classification object types. For more information, 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

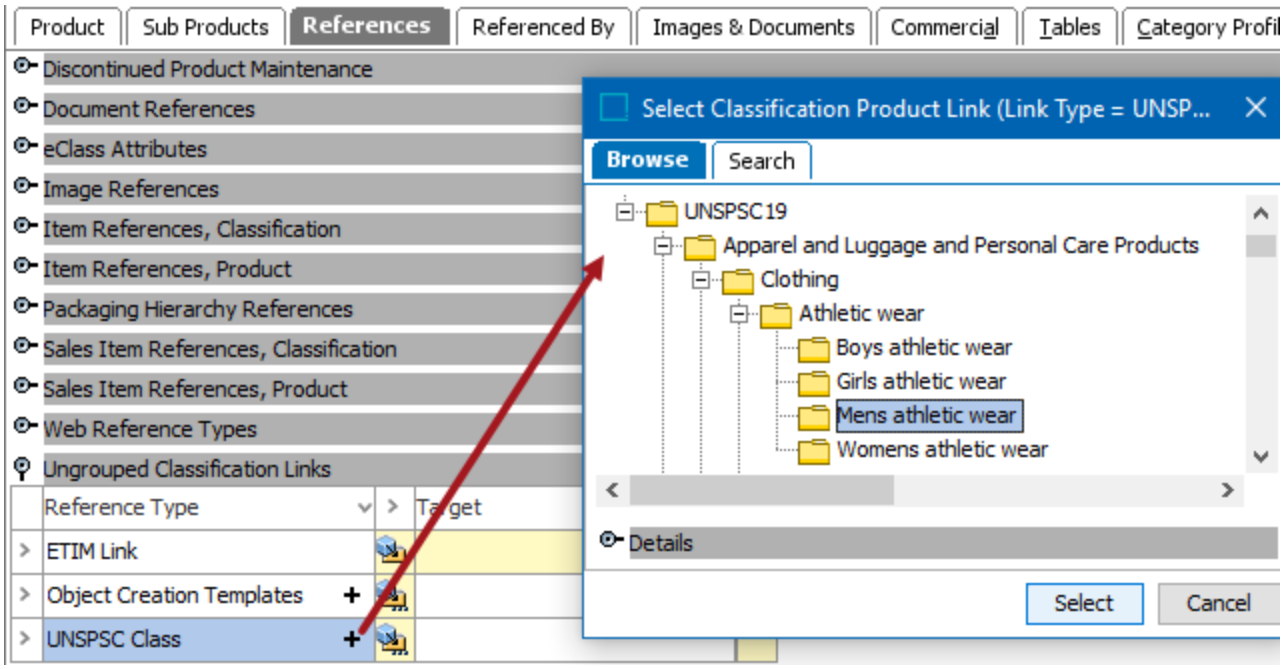
After creating a product to classification link type, you can add that type on an object's References or Referenced By tabs.

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**.

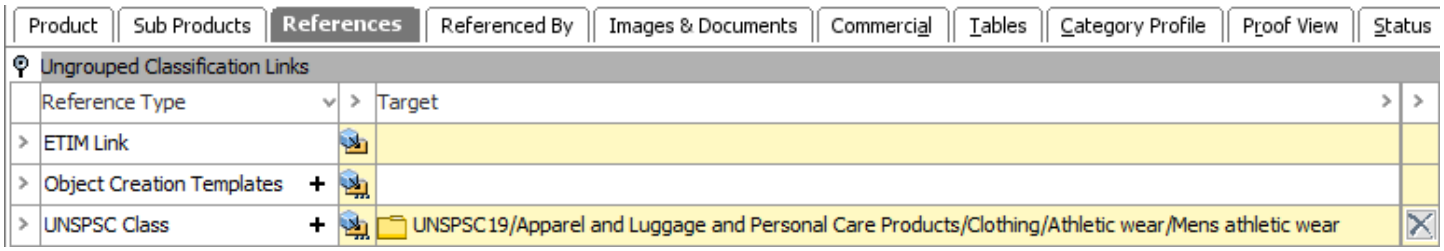
1. To add a product to classification link to either the **References** or **Referenced By** tab of an object, use one of the following ways to find the product to classification link type in a flipper:
   - If the link type has been added to an attribute group, it appears under a flipper that is named 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.



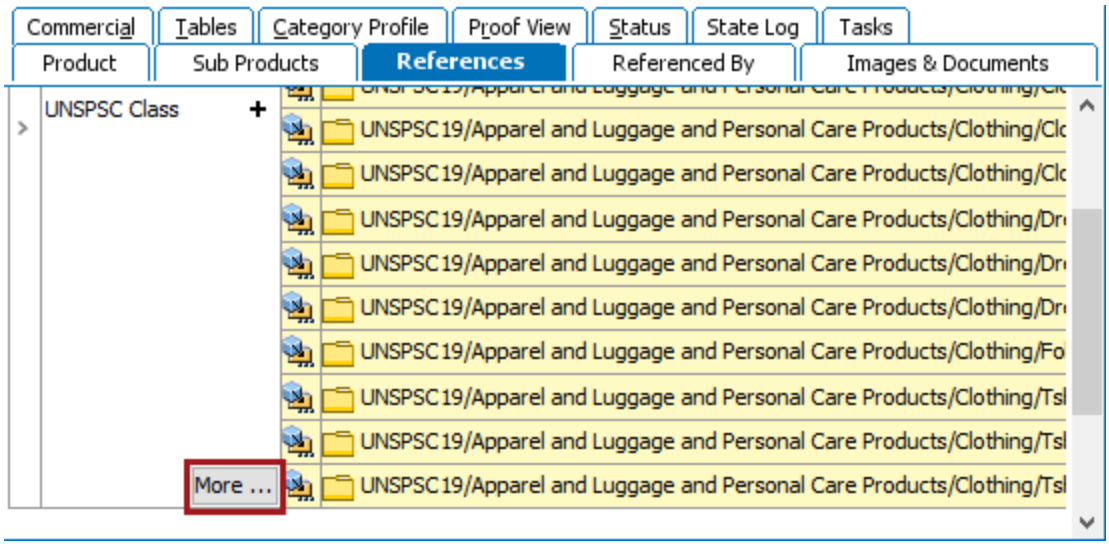
Once selected, the object displays as an added link.



- 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.

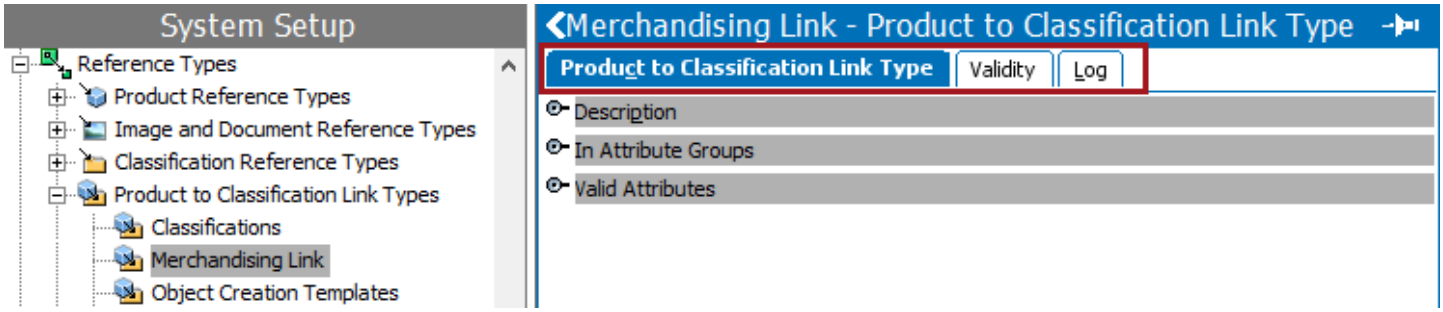


Repeatedly clicking the button doubles the number of links shown each time, until the full set is displayed. Note that if new links are added, they display on the list immediately indicating that they were added properly, though normal functionality will resume if the user presses the 'More...' button 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.

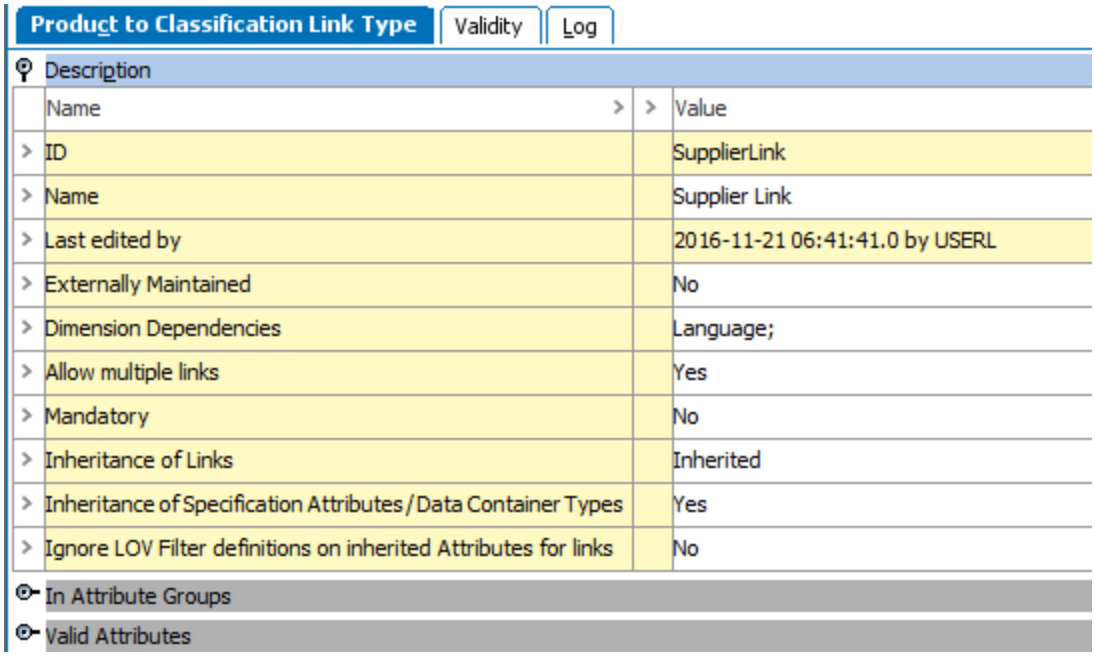


Product to Classification Link Type Tab

This tab holds basic information within the Description, In Attribute Groups, and Valid Attribute flippers.

Description Flipper

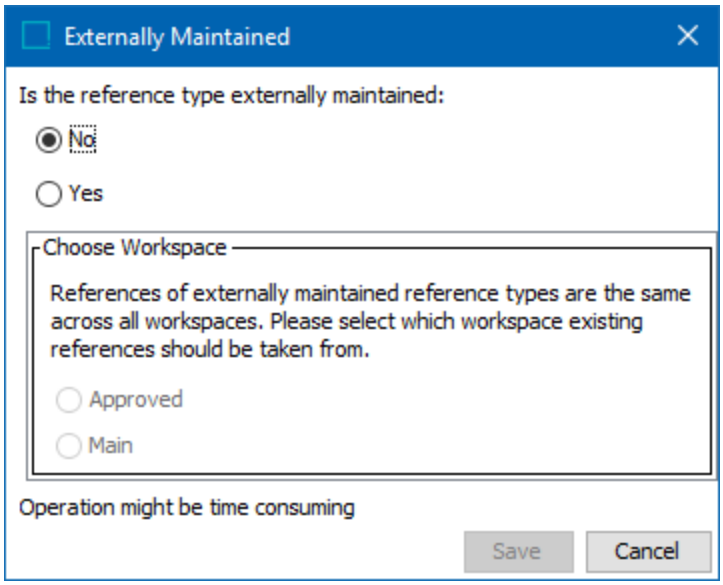
The Description flipper includes basic information to identify the link. This data is originally set up in the Product to Classification Link Type wizard, see **Creating a Product to Classification Link Type** documentation.



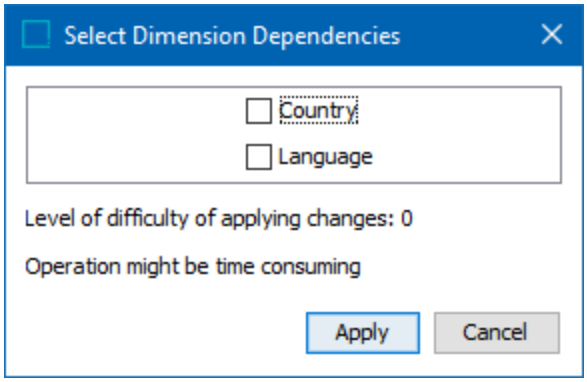
All fields except the ID and 'Last edited by' can be edited. The method required to edit a parameter in the link editor is determined by the parameter type:

- 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.



- 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.



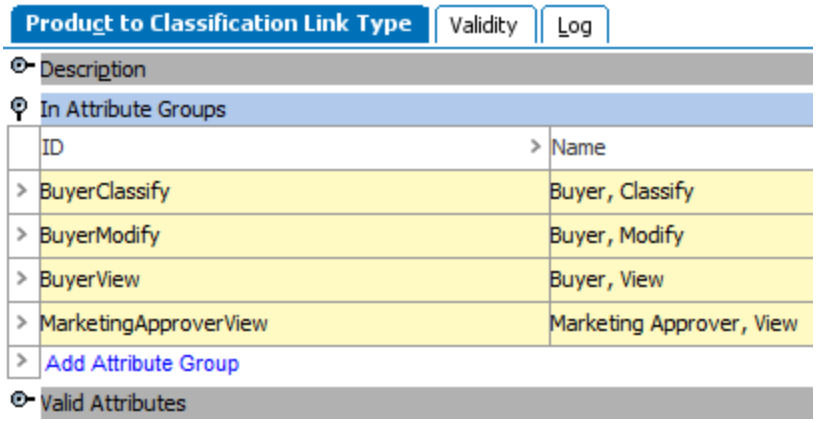
- For **Allow multiple links**, click the parameter to display the dropdown list of options. Three (3) small dots displayed below the icon indicates multiple links are allowed (). This parameter is discussed in the **Product to Classification Link Type - Advanced** documentation.
- For **Mandatory**, 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**.

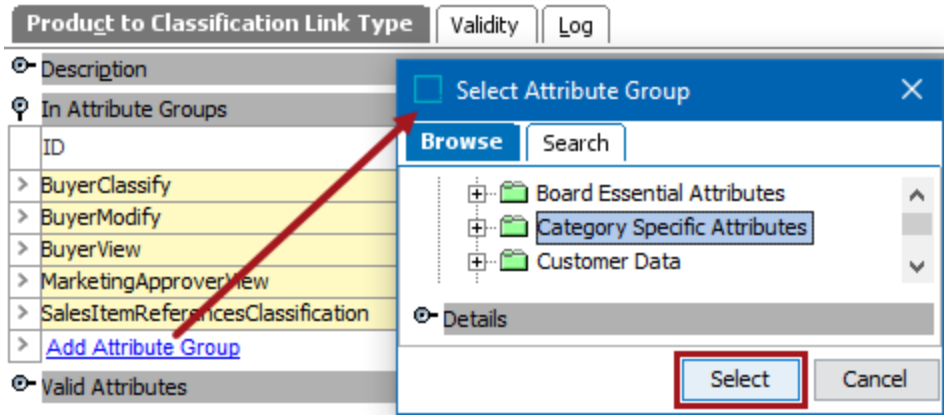
- For **Inheritance of Links** and **Inheritance of Specification Attributes**, click the parameter to display the dropdown list of options. The small green down arrow displayed above the icon indicates the value is inherited (). This parameter is discussed in the **Product to Classification Link Type - 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 / Data Container Types = 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 **Setting Hierarchical Filters on Attributes**.

In Attribute Groups Flipper

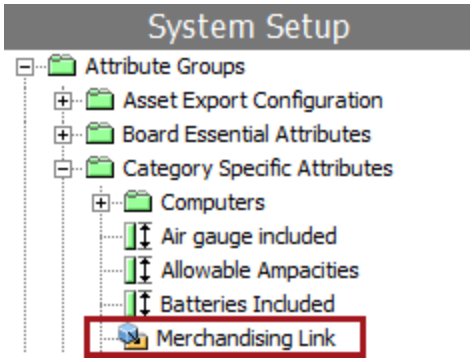
The 'In Attribute Groups' flipper includes all attribute groups that include the link. Adding a link type to one or more attribute groups determines the label of the flipper that displays the link in an object editor, enables you to set up user privileges to restrict approval of the link type, and to define a customized view of the link types in the group. For more information, see **Attribute Groups for Reference and Link Types** documentation.



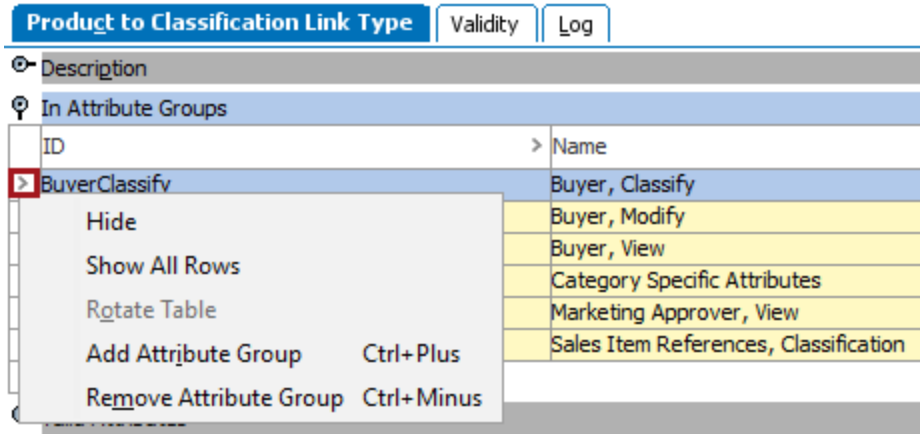
- To add an attribute group, click the **Add Attribute Group** link to display the Select Attribute Group dialog. Choose a group and click the Select button to add the link type to the attribute group.



Once added to an attribute group, the link type is also displayed within the System Setup > Attribute Groups node.

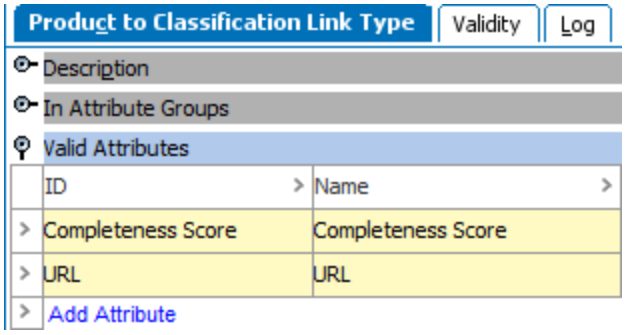


- To modify the attribute groups included or displayed, right-click the arrow to the left of the group ID and select an option from the menu.

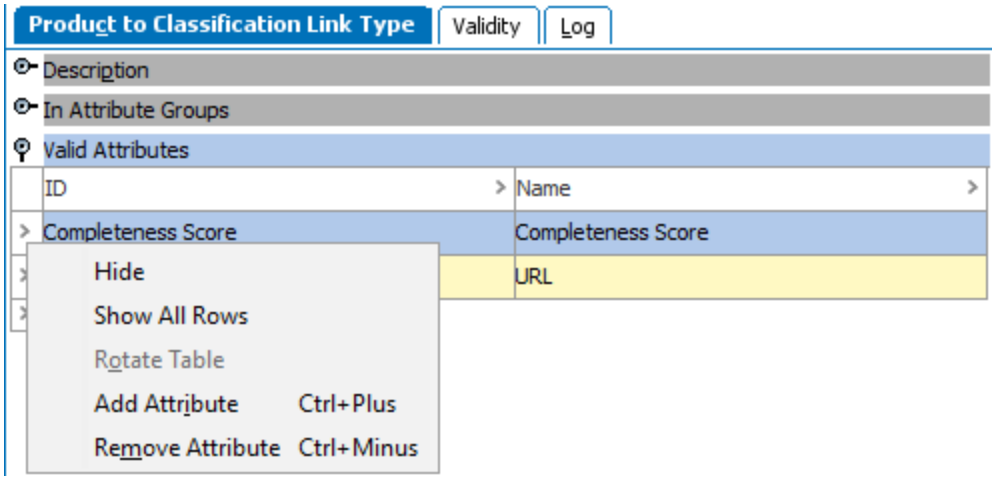


Valid Attributes Flipper

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.



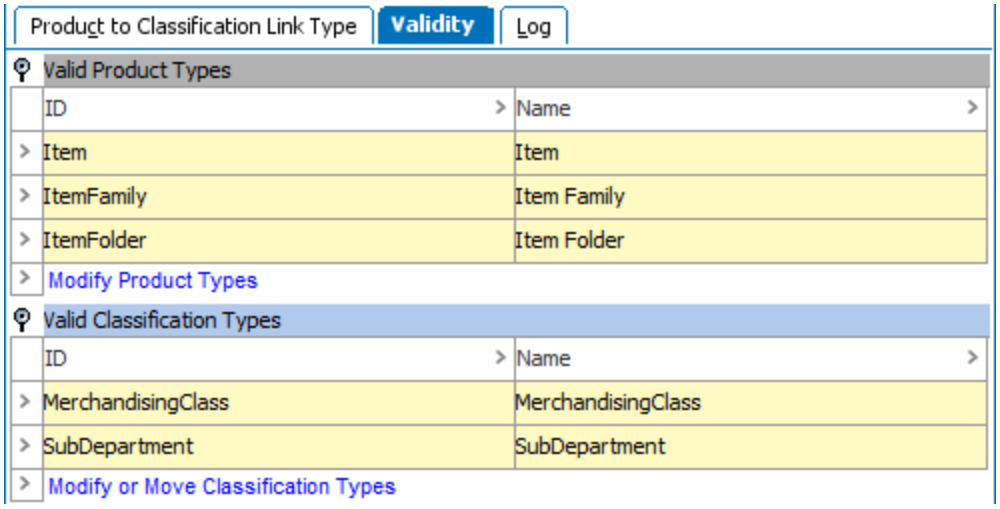
To modify the attributes included or displayed, right-click the arrow to the left of the ID and select an option from the menu.



For more information, see **Metadata Attributes on Reference and Link Types** documentation.

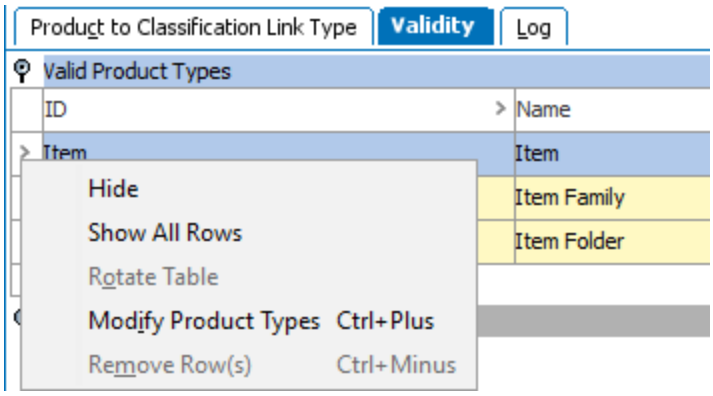
Validity Tab

The Validity tab stores the source product types and target classification types allowed on the product to classification link type.



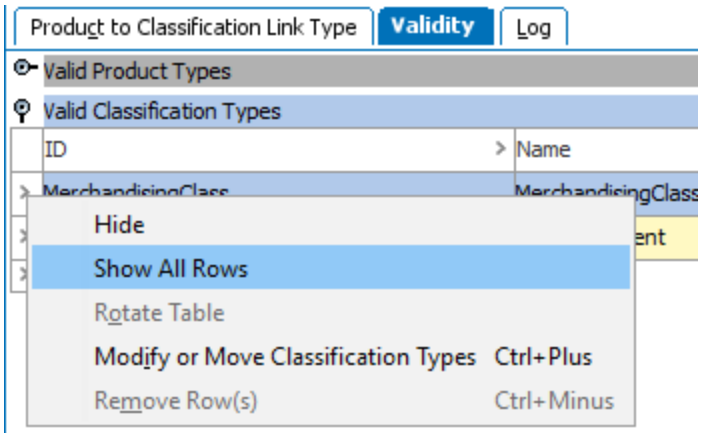
- For **Valid Product Types**, any number of product object types can be added. Click the 'Modify Product Types' link to add or remove a product object type. When an object type is removed from the list, if Product to Classification Link of this type already exists for the product object type, the links 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.



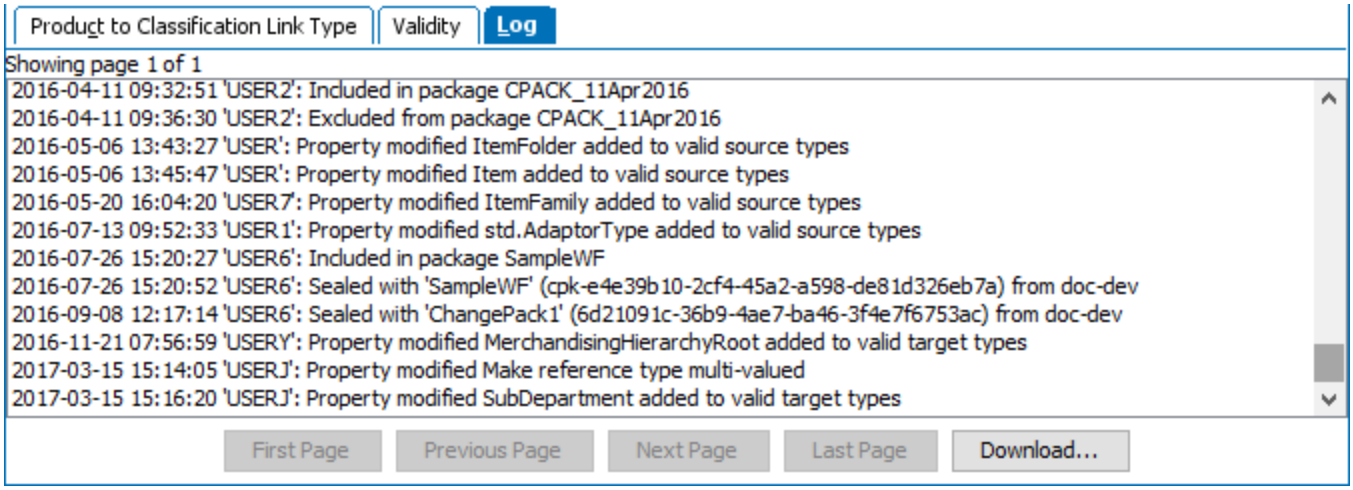
- For **Valid Classification Types**, any number of classification object types can be added, however, the same object type cannot be added to more than one product to classification link type. Click the 'Modify Product Types' link to add or remove a classification object type. This parameter is discussed 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.



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.



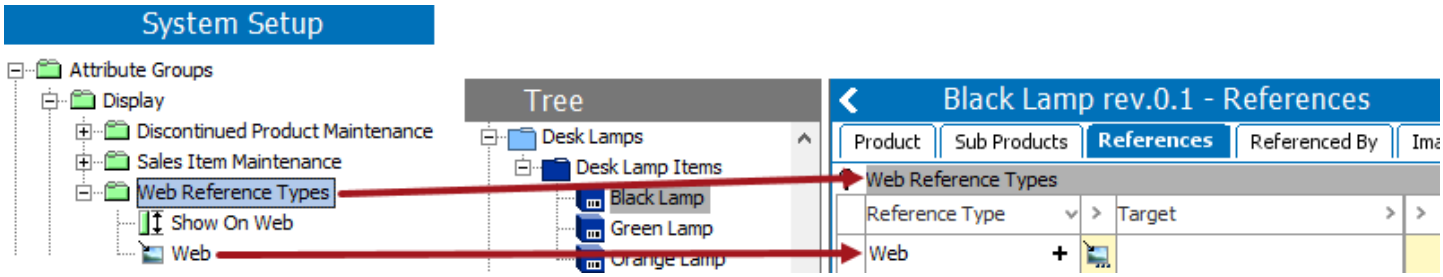
As additional log entries become available and the page is filled, the pagination buttons are enabled. Additionally, you can download the complete log as a zipped text file.

Attribute Groups for Reference and Link Types

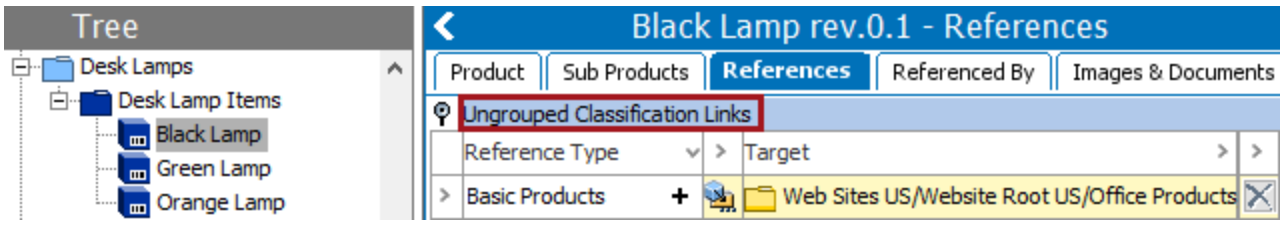
Adding a Reference or Product to Classification Link type to an attribute group provides easy identification, a customized view, and control of approvals via user privileges.

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.

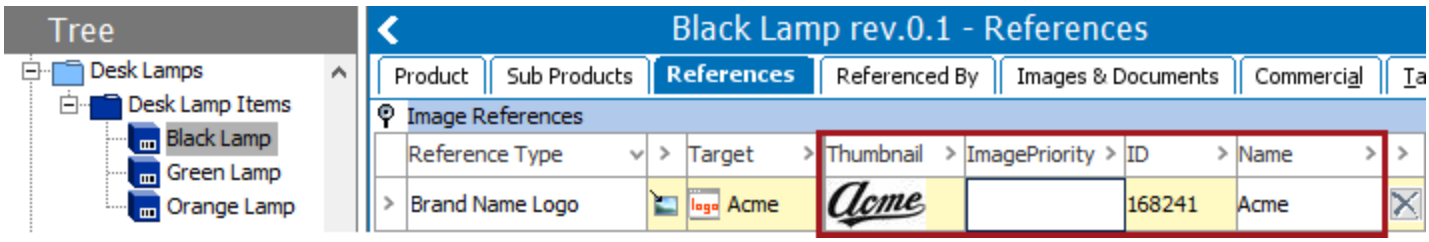


Otherwise, the reference or link displays in a flipper with the 'Ungrouped' label.



Customized View

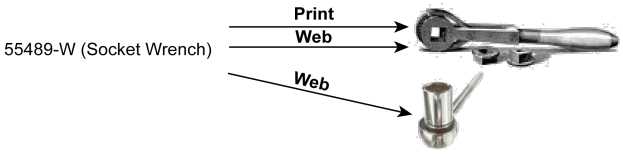
A customized view can be defined on the attribute group of all reference or link types in the group to display additional information on the reference or link. For more information, see the **Creating a Customized View** section in the **Attribute Groups** topic of the **System Setup / Super User Guide** documentation.



Control of Approvals via User Privileges

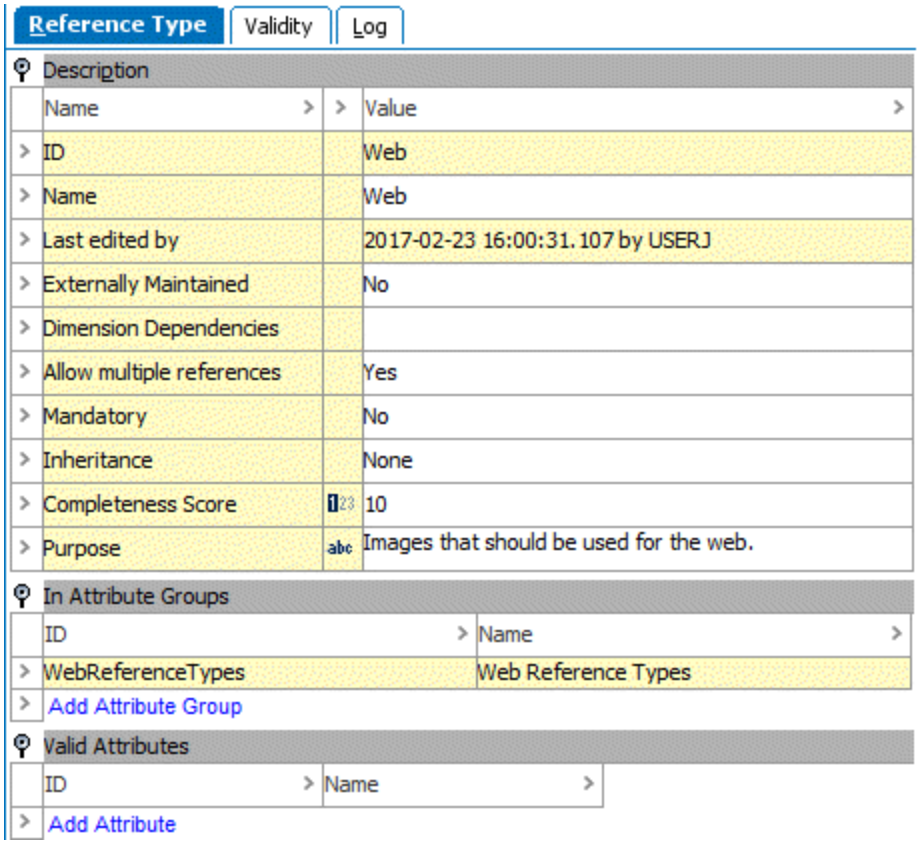
Perhaps most importantly, an attribute group allows you to control user privileges for approval of the reference or link. This means that approval of specific types of references and links is based on the user's access to the attribute group, so that users will only be allowed to approve references or 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.

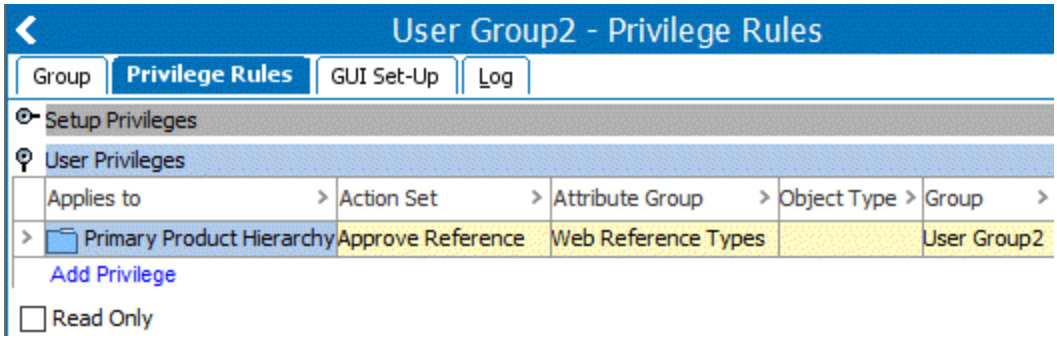


To implement this, the reference type named 'Web' is added to an attribute group named 'Web Reference Types.'

A reference or link type can be added to as many attribute groups as needed by different users with different privileges.



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.



A user with this privilege setup will only be allowed to approve references or links in the attribute group 'Web Reference Types.'

For more information on adding a Reference or Product to Classification Link type to an attribute group, see the **In Attribute Groups Flipper** section of:

- **Maintaining a Reference Type**
- **Maintaining a Product to Classification Link Type**

Dimension Dependent Reference and Link Types

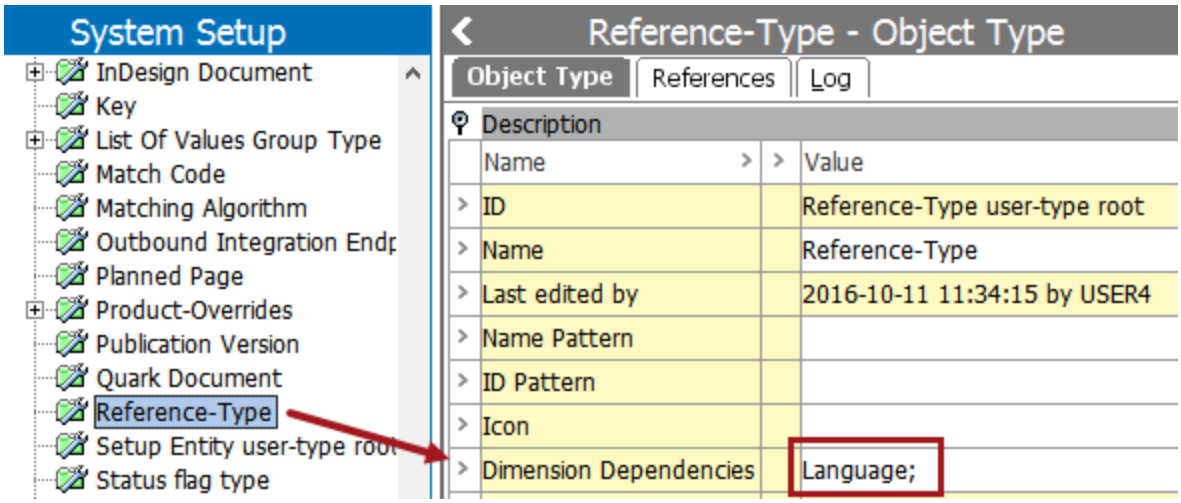
A Reference or Product to Classification Link type can be set up to be dependent on a dimension. Dimension dependency is used to specify the dimension points, and by extension, the contexts, where a reference or link should display or should be suppressed. A reference or link can only be dependent on 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.



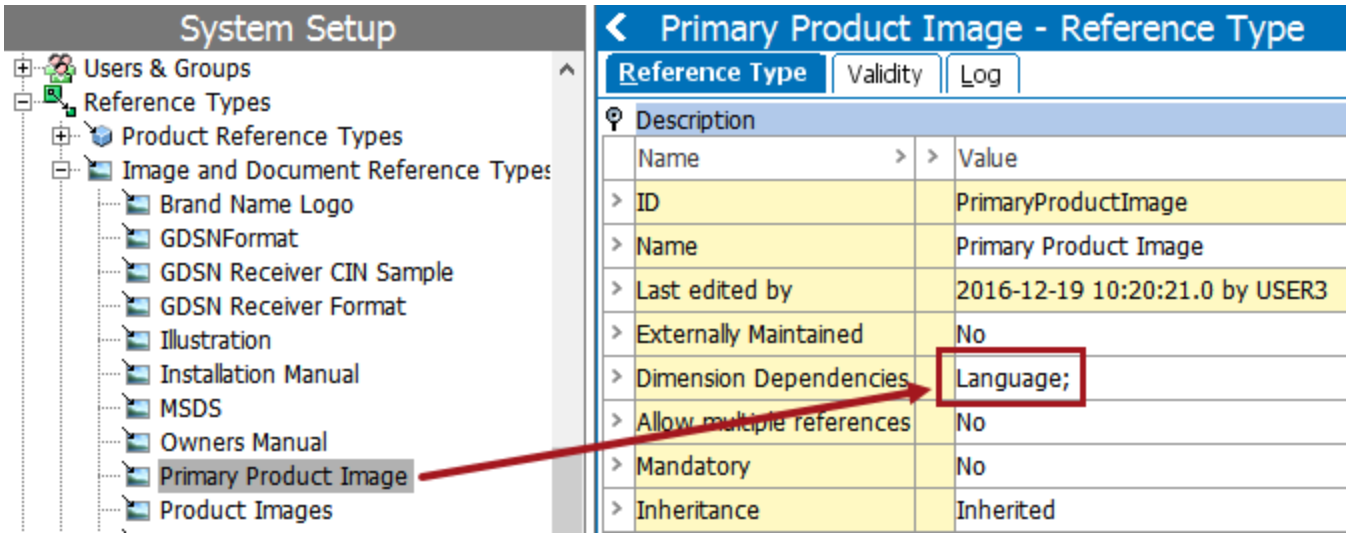
Using the Dimension Dependencies parameter on the Reference-Type object type is a global setting that enables translation of reference type names and requires that all reference type names be translated. Translated

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.



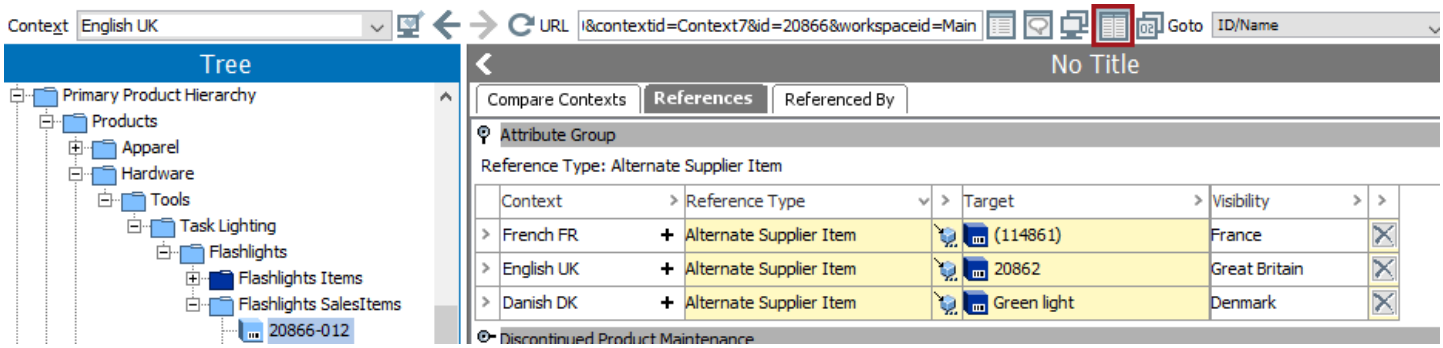
This is not a typical setup because the References / Referenced By tabs can become difficult to manage due to the layout required when viewing multiple contexts.

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

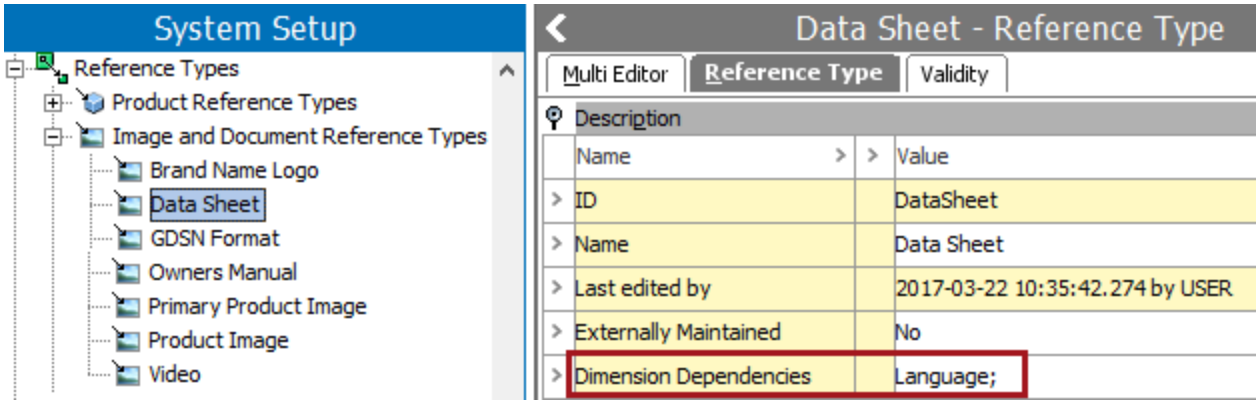
When dimension dependency is applied to a reference type, values can be applied based on the selected dimension. For example, the following reference is dependent on the Country dimension, so each country can have a different setting. The Context view displays all values for each country, based on the targets selected from the View > Target menu option.



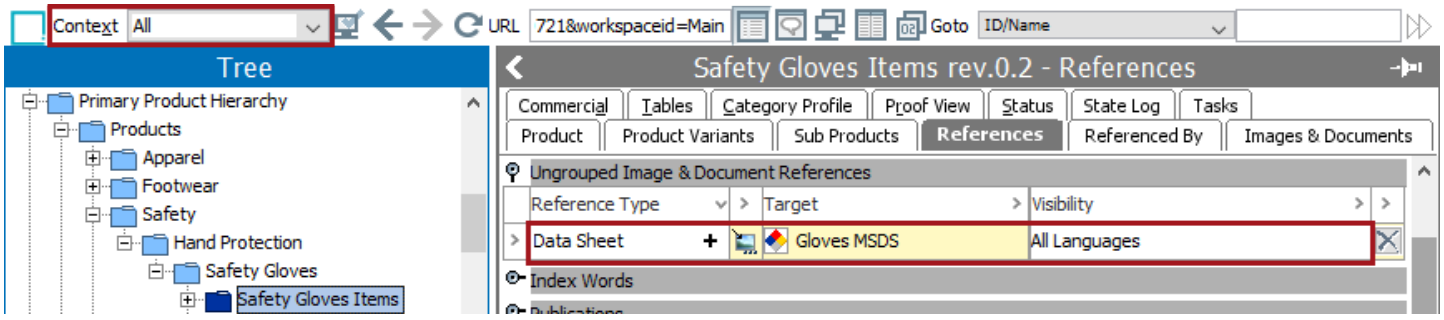
Dimension Dependent Reference Example

Consider a product that needs a 'Data Sheet' reference. The reference should be linked to the product for all languages except the English. Use the following steps to accomplish this:

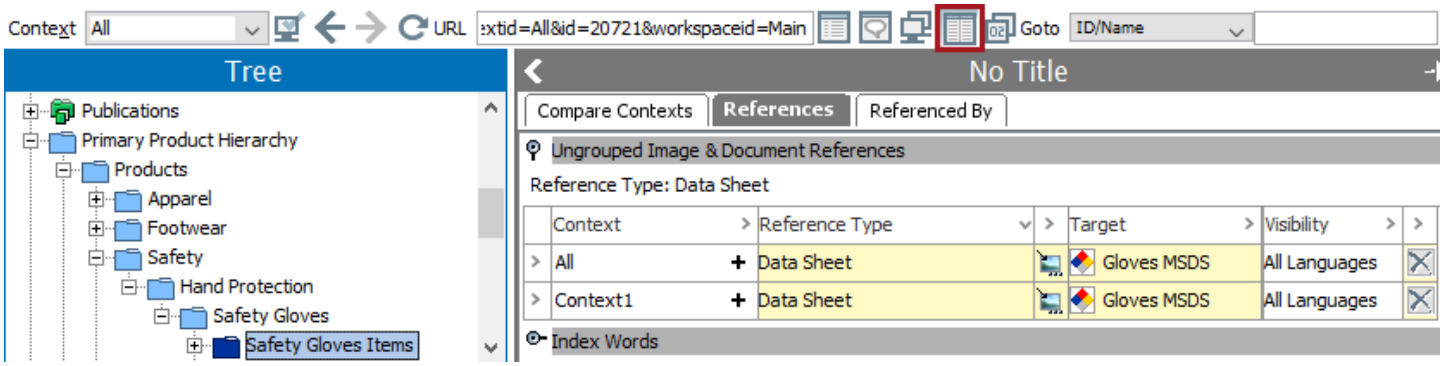
1. Create a reference type named 'Data Sheet' and set it to be language dependent.



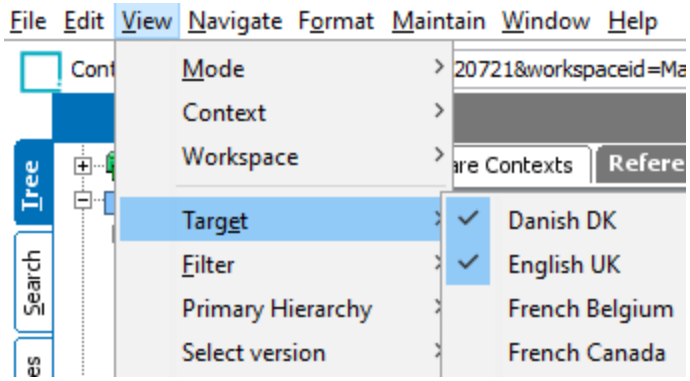
2. Select the 'All' context, add a Data Sheet reference. The reference is inherited to all languages.



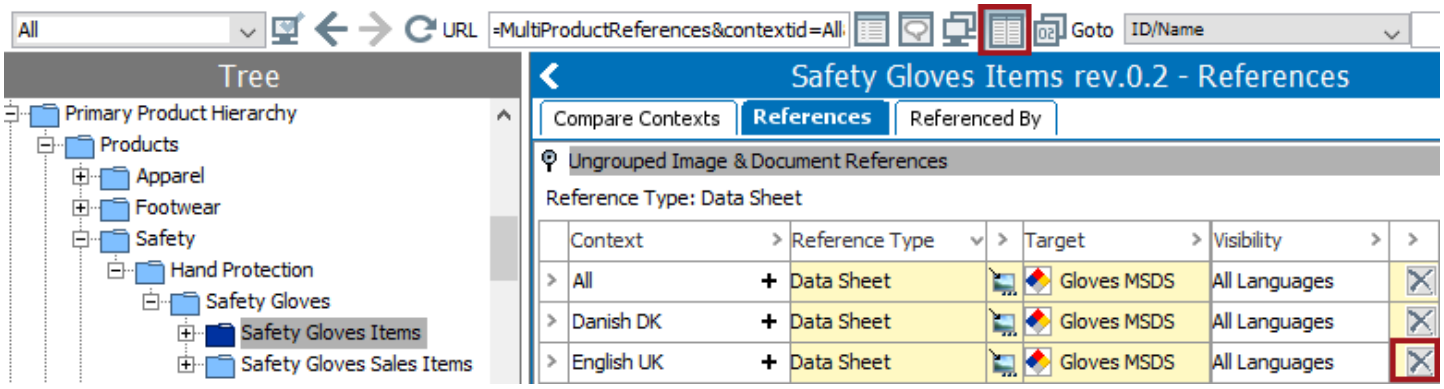
3. Display the References tab in Context Mode by clicking in the toolbar.



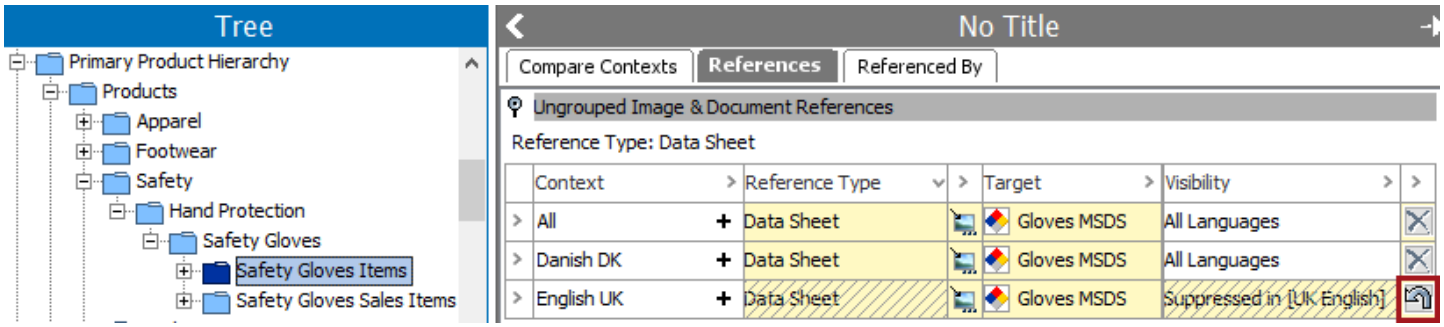
4. Navigate to View > Target and select the Contexts for the comparison view to display.



5. Click the X button to suppress the reference for the English language.



6. The suppressed reference type now appears shaded. If necessary, the reference can be restored by clicking the undo button on the shaded row.

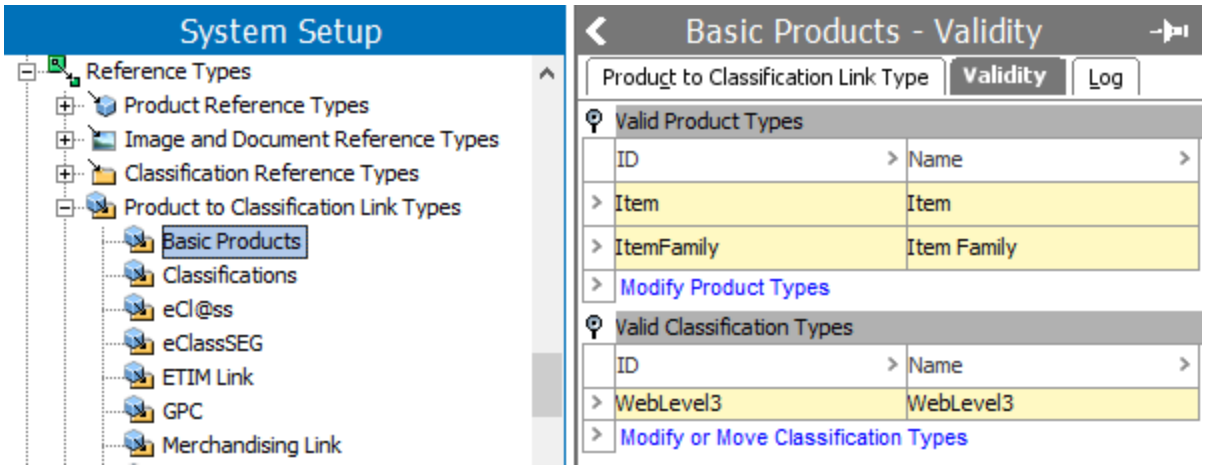

Tree
Primary Product Hierarchy
Products
Apparel
Footwear
Safety
Hand Protection
Safety Gloves
Safety Gloves Items
Safety Gloves Sales Items
No Title
Compare Contexts
References
Referenced By
Ungrouped Image & Document References
Reference Type: Data Sheet
Context
Reference Type
Target
Visibility
All
Data Sheet
Gloves MSDS
All Languages
Danish DK
Data Sheet
Gloves MSDS
All Languages
English UK
Data Sheet
Gloves MSDS
Suppressed in [UK English]

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.

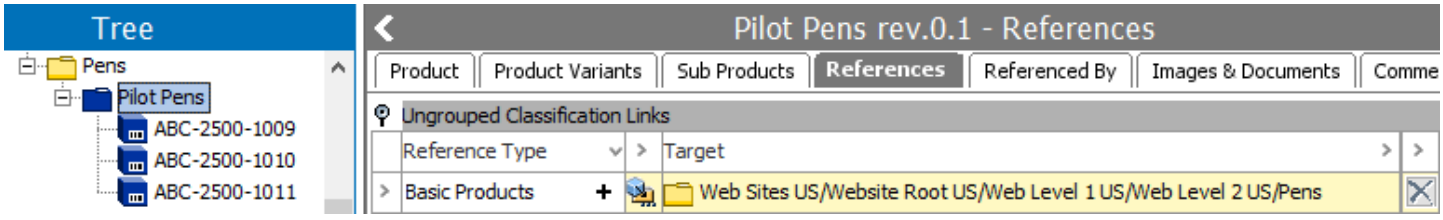
System Setup

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.



Tree

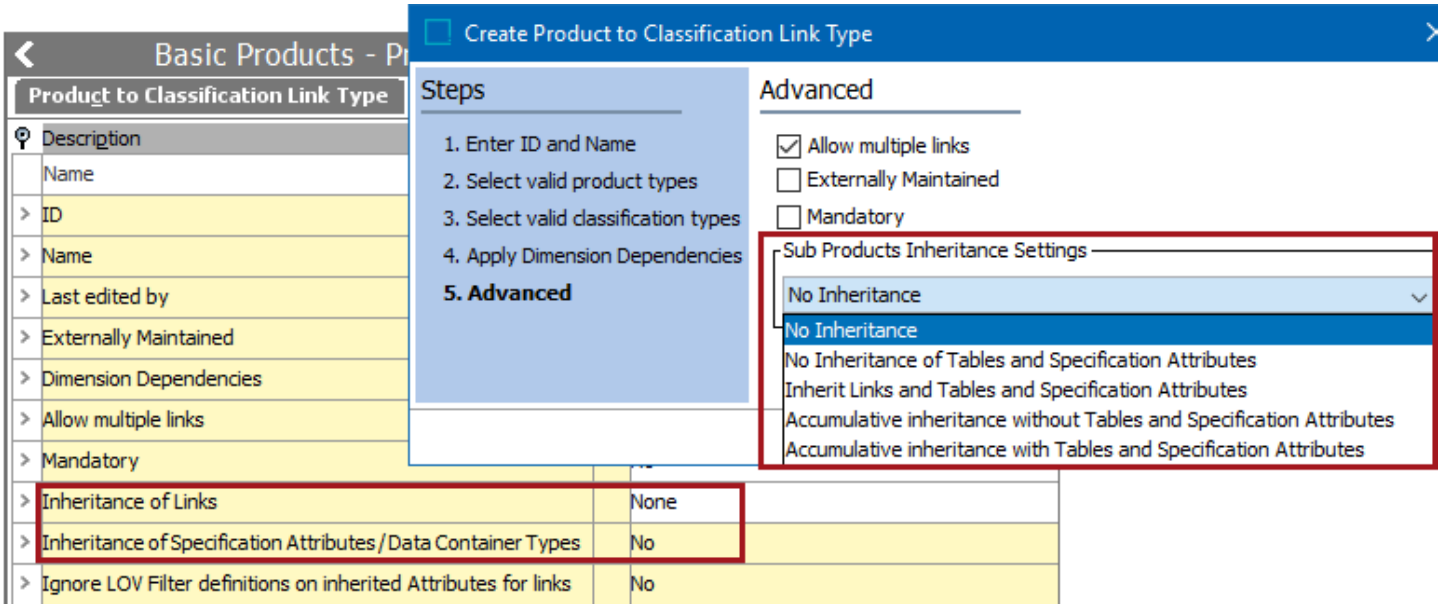
The 'Basic Products' product to classification link type is used to link a classification named 'Pens' (object type Weblevel3) to a product family named 'Pilot Pens' (object type ItemFamily).



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 / Data Container Types.' Note that the

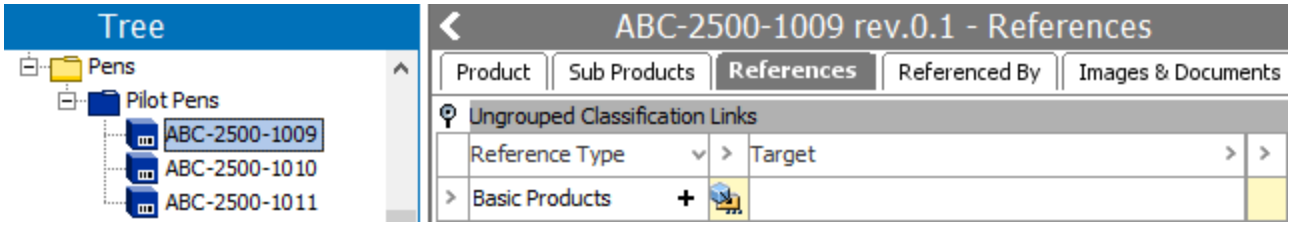
inheritance of tables is indicated by the 'Inheritance of Specification Attribute' parameter setting.



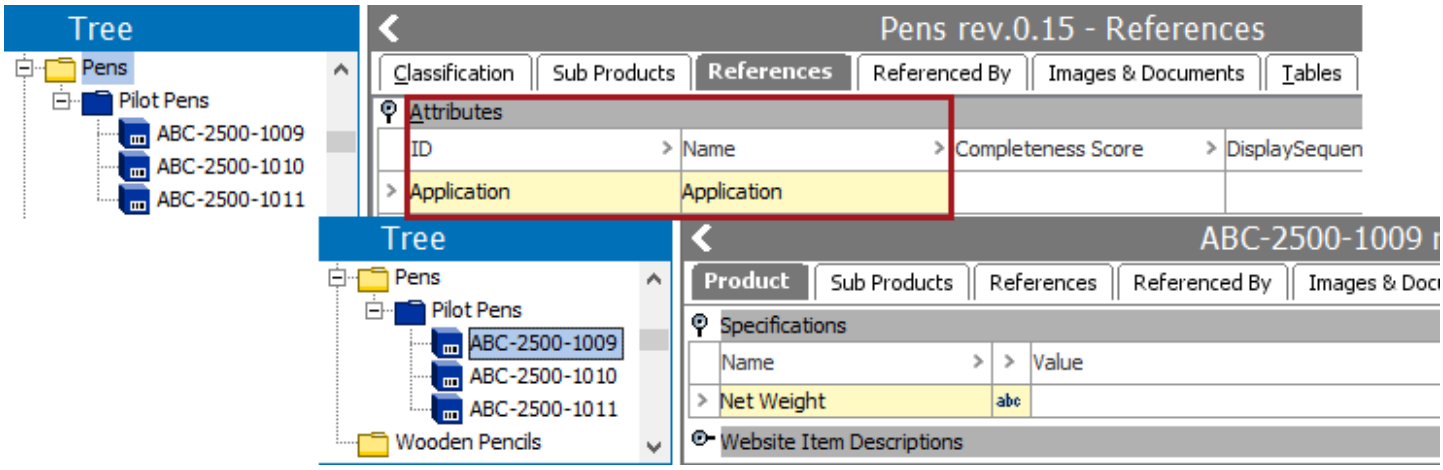
No Inheritance

This option in the wizard displays as 'Inheritance of Links = None' and 'Inheritance of Specification Attributes / Data Container Types = No' in the editor.

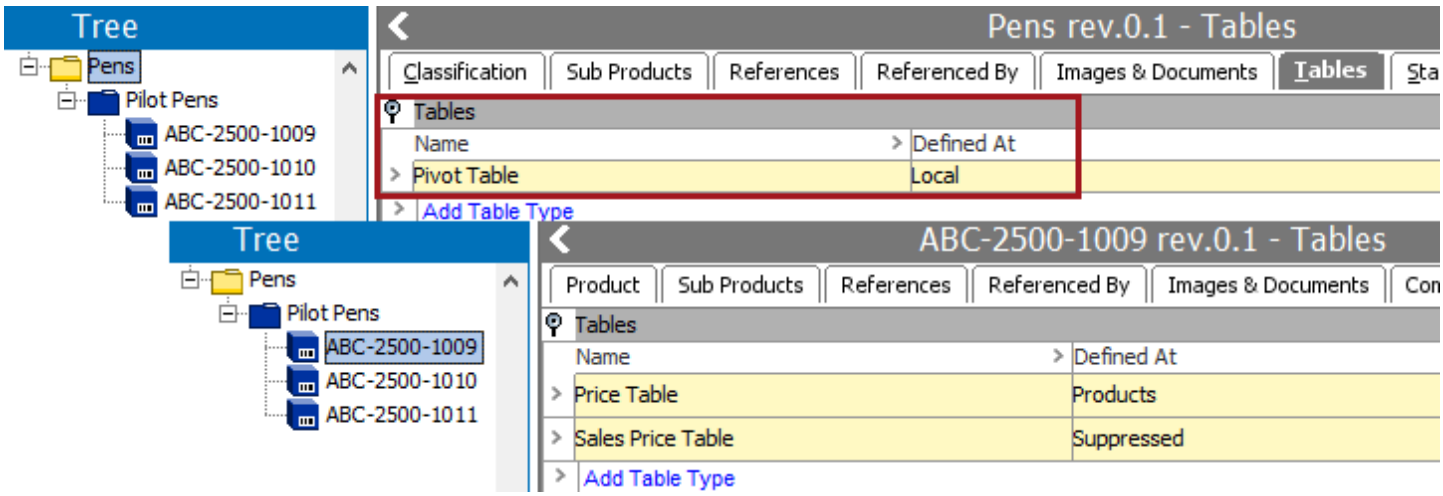
The value of the Basic Products link is not inherited from the product family (Pilot Pens) to the linked products (ABC-2500-1009, ABC-2500-1010, and ABC-2500-1011).



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.



Tables applied to the classification are not inherited to products below the classification.

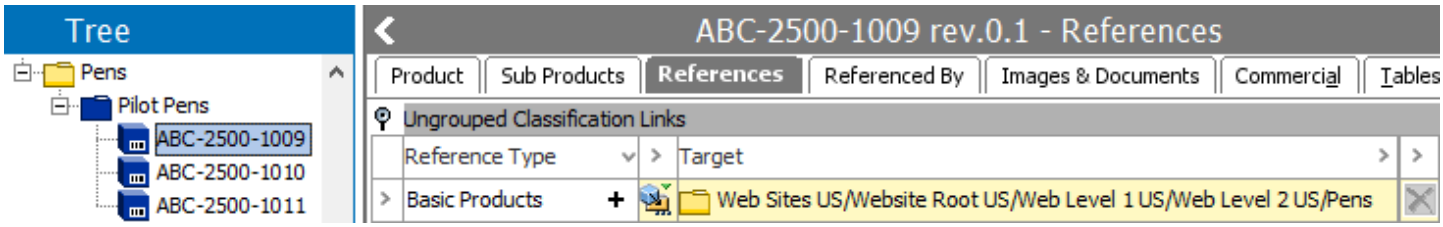


No Inheritance of Tables and Specification Attributes

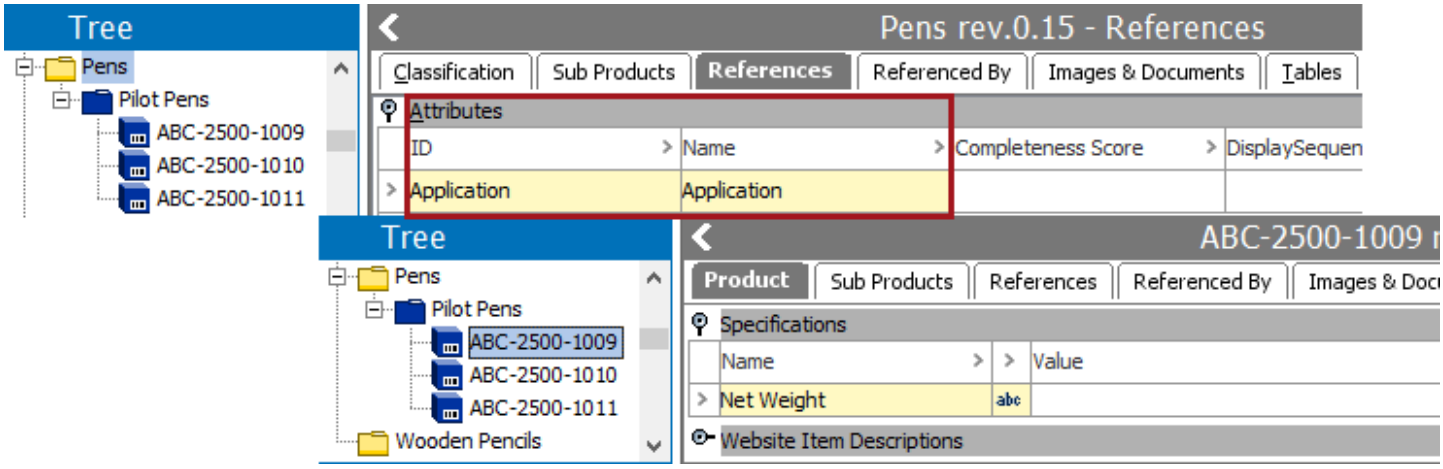
This option in the wizard displays as 'Inheritance of Links = Inherited' and 'Inheritance of Specification Attributes / Data Container Types = No' in the editor.

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).

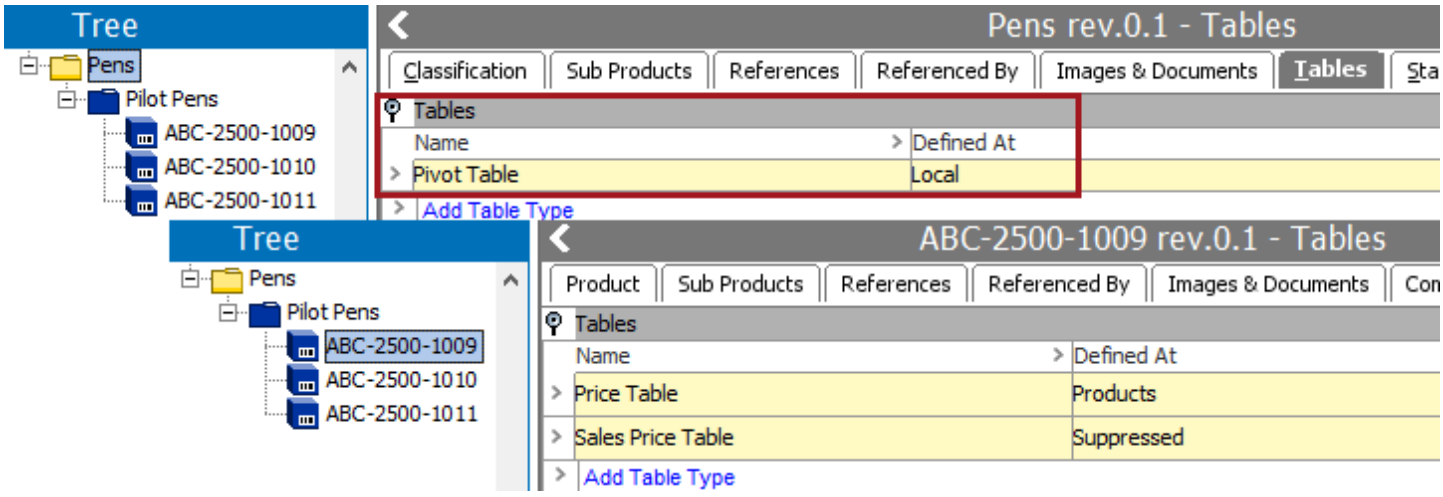
Note: If multiple links are allowed, and a new link of this type is added to a linked product (sub product), the inherited link is removed and only the local link remains.



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.



Tables applied to the classification are not inherited to products below the classification.

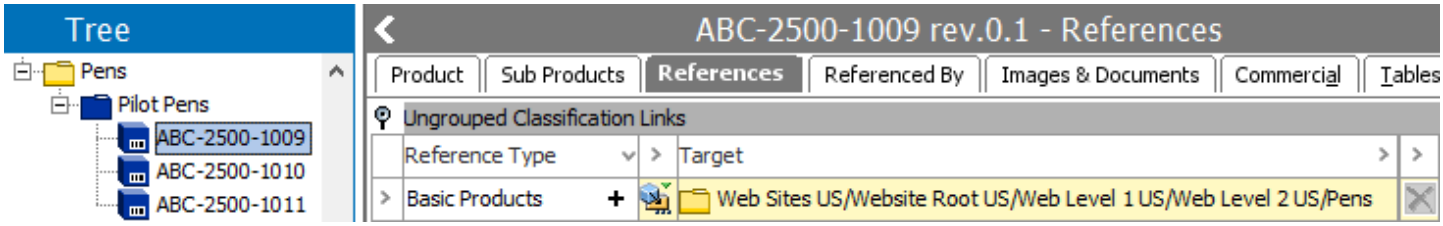


Inherit Links and Tables and Specification Attributes

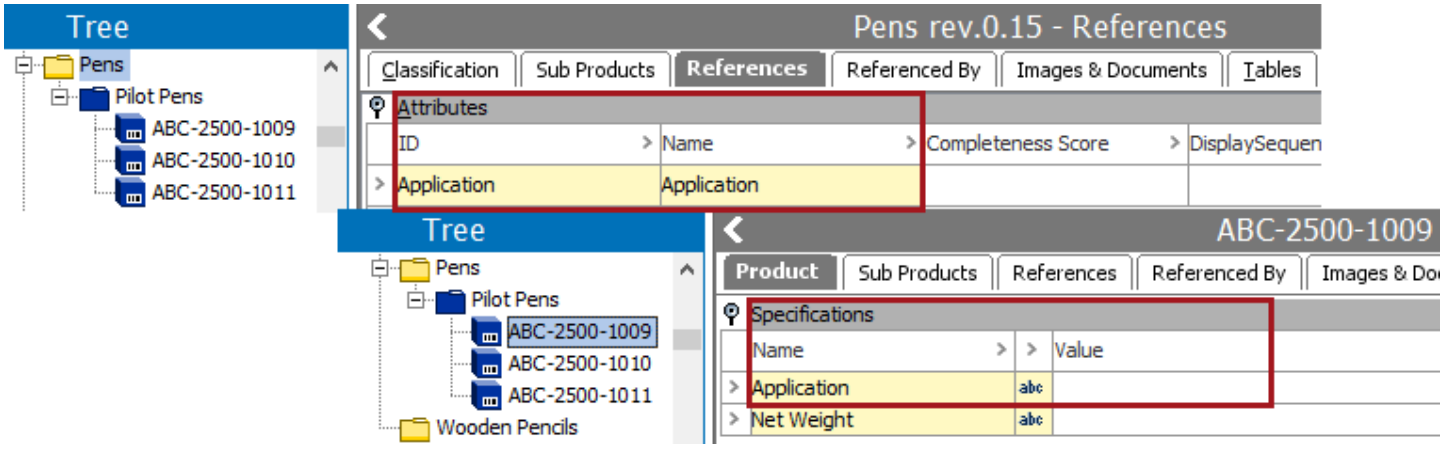
This option in the wizard displays as 'Inheritance of Links = Inherited' and 'Inheritance of Specification Attributes / Data Container Types = Yes' in the editor.

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).

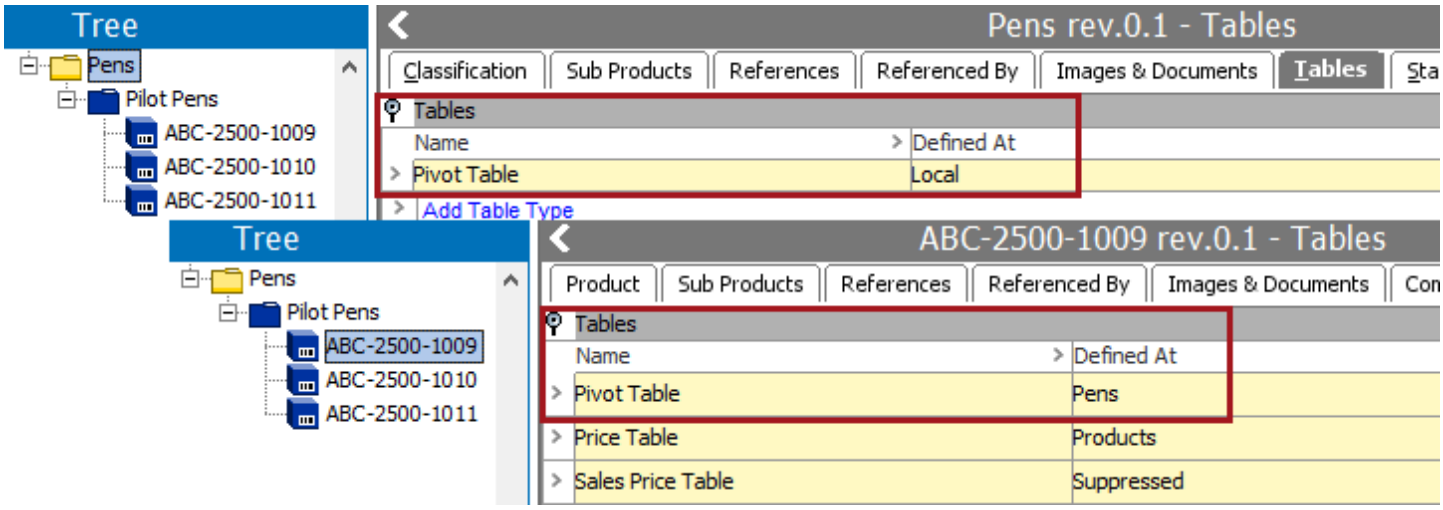
Note: If multiple links are allowed, and a new link of this type is added to a linked product (sub product), the inherited link is removed and only the local link remains.



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.



Tables applied to the classification are inherited to products below the classification.

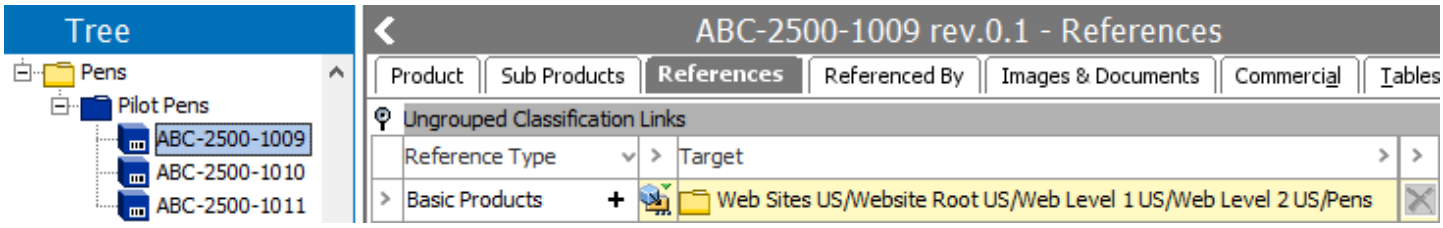


Accumulative Inheritance without Tables and Specification Attributes

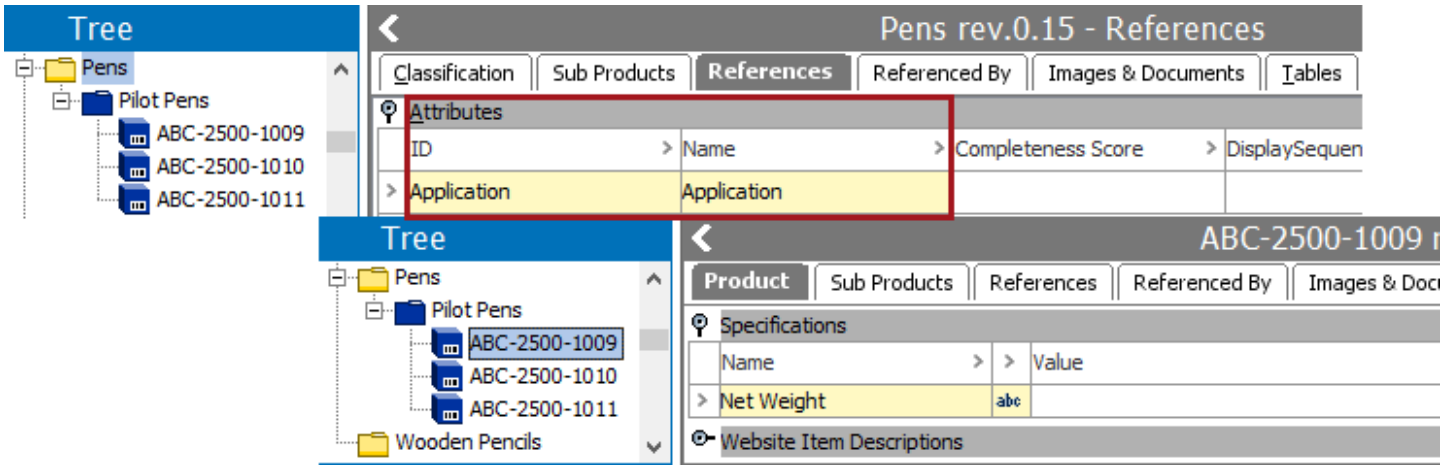
This option in the wizard displays as 'Inheritance of Links = Accumulative' and 'Inheritance of Specification Attributes / Data Container Types = No' in the editor.

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).

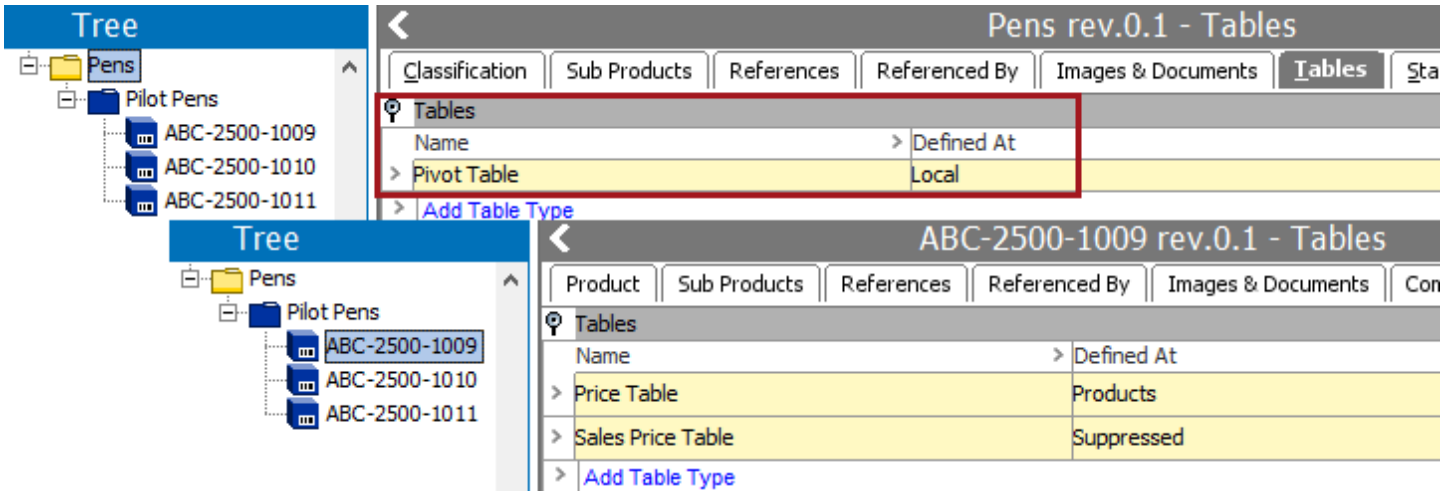
Note: If multiple links are allowed, and a new link of this type is added to a linked product (sub product), the inherited link is retained and the local link is added.



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.



Tables applied to the classification are not inherited to products below the classification.

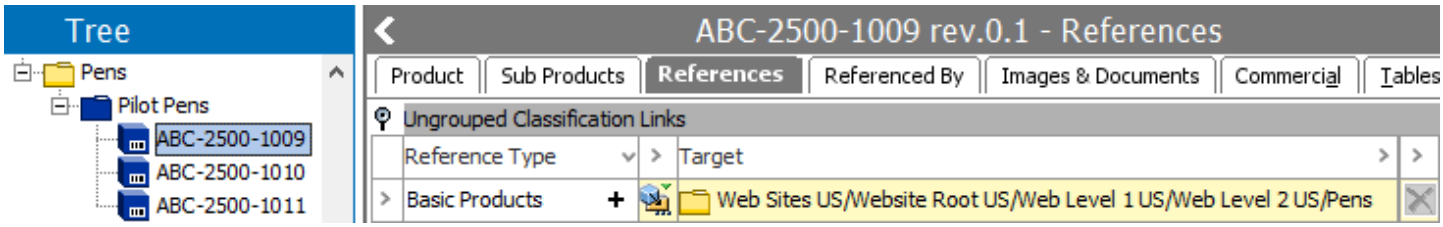


Accumulative Inheritance with Tables and Specification Attributes

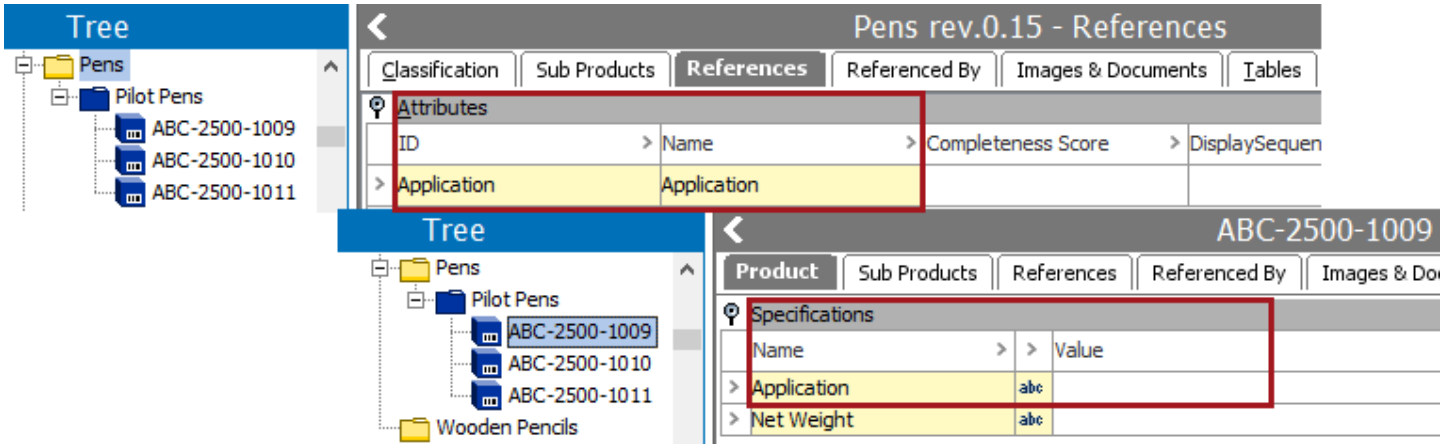
This option in the wizard displays as 'Inheritance of Links = Accumulative' and 'Inheritance of Specification Attributes / Data Container Types = Yes' in the editor.

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).

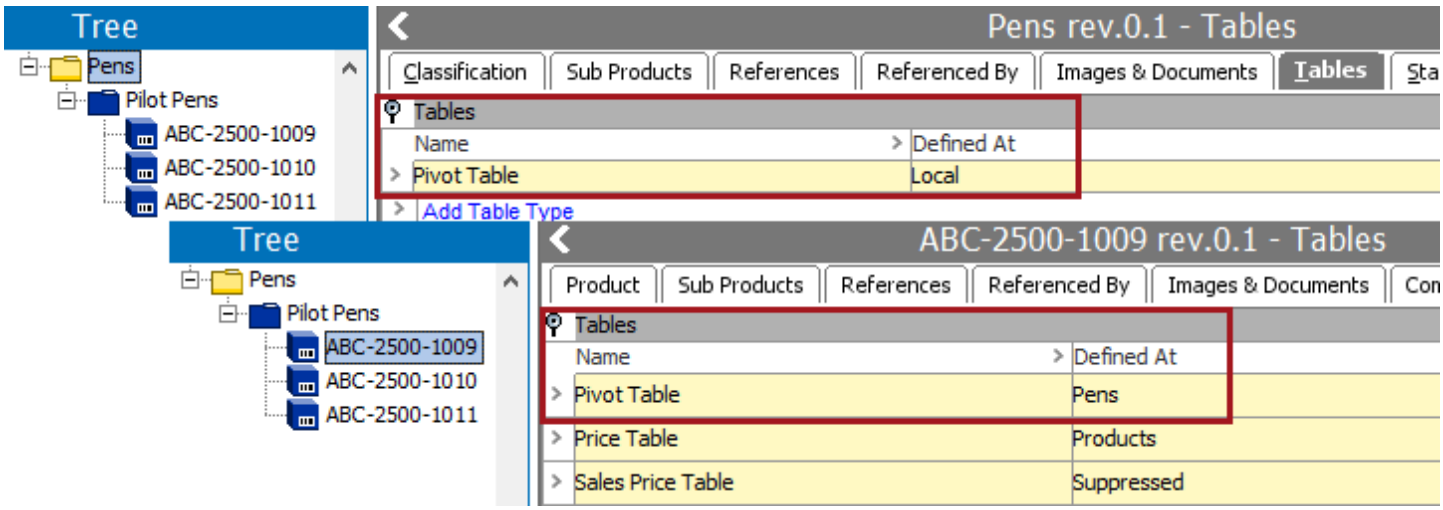
Note: If multiple links are allowed, and a new link of this type is added to a linked product (sub product), the inherited link is retained and the local link is added.



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.



Tables applied to the classification are inherited to products below the classification.



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 / Data Container Types' parameters as defined in **Maintaining a Product to Classification Link Type** documentation.

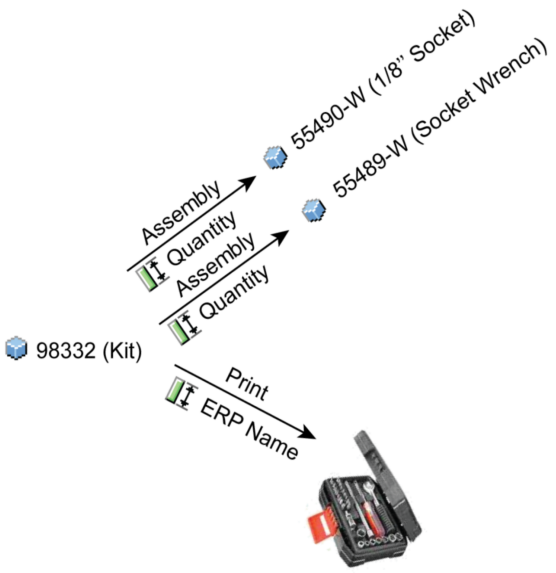
Metadata Attributes on Reference and Link Types

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.



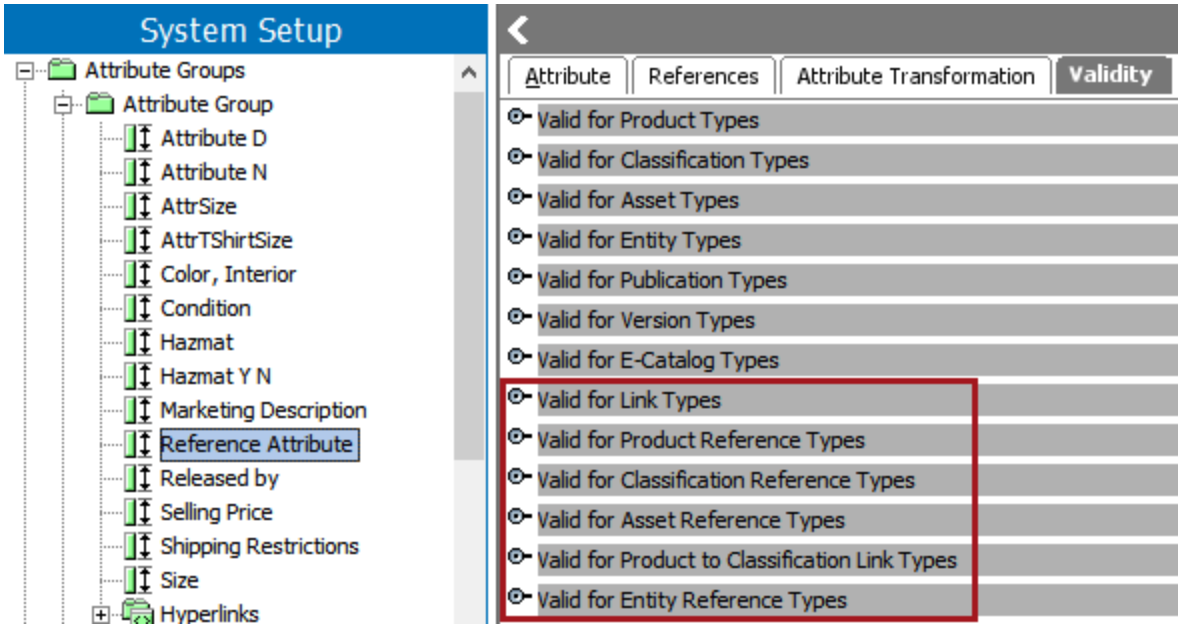
- A description attribute named Quantity has been applied the reference type Assembly. This attribute holds number of sockets to be included in the kit.
- A description attribute named 'ERP Name' has been applied the reference type Print. This attribute holds the name of the reference used in an ERP System.

For information on adding metadata to the attribute link types, see **Product Attribute Link Type** or **Classification Attribute Link Type**.

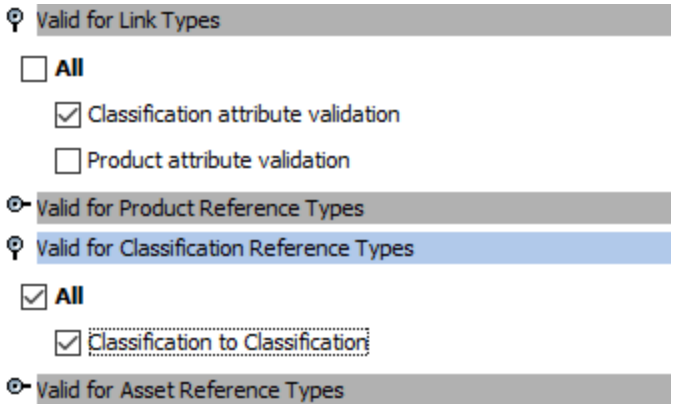
Add a Metadata Attribute

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.

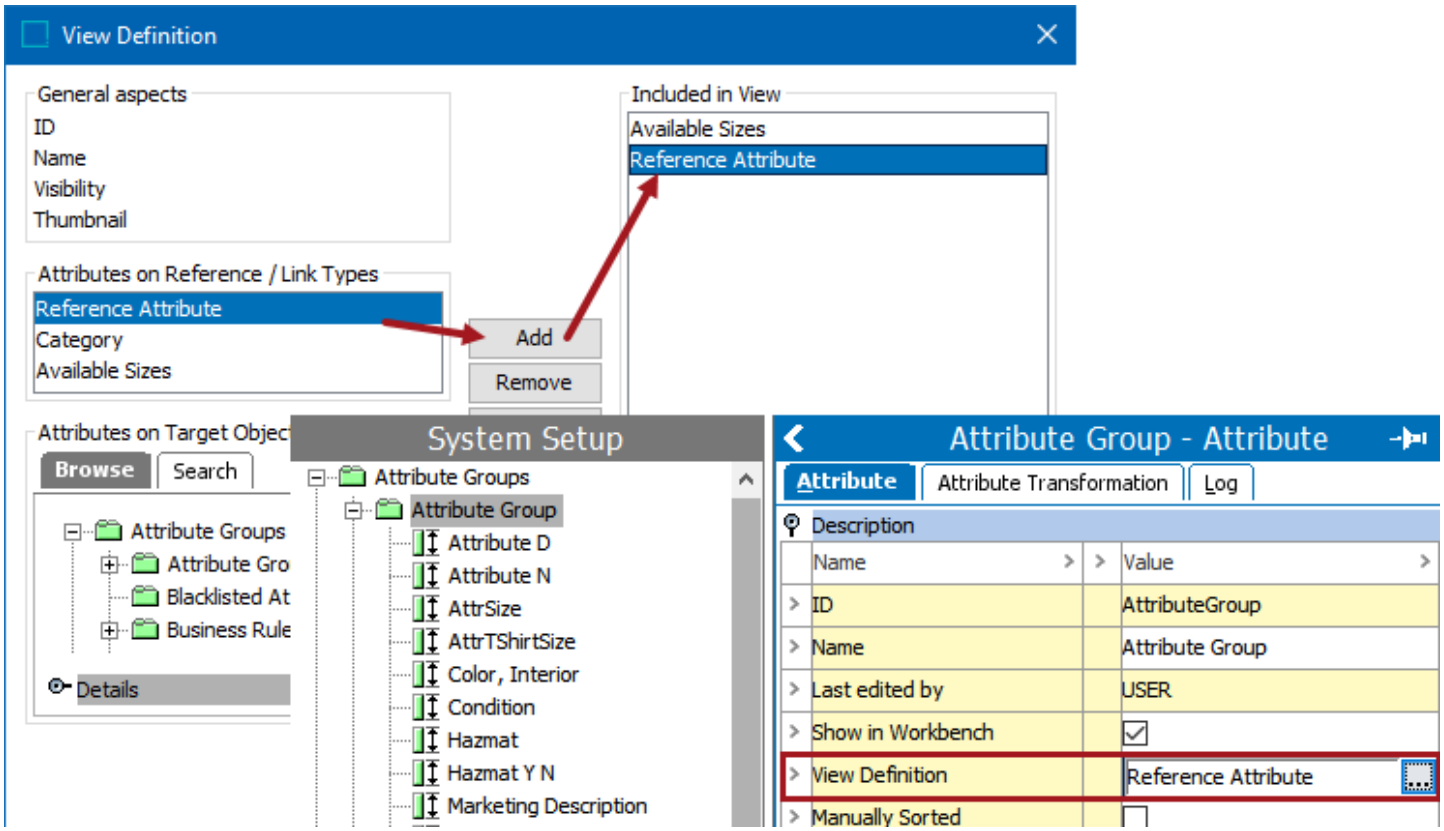


3. Select all reference types and link types that should display the metadata attribute. Checking an option also updates the Attributes flipper of the selected reference or link type.

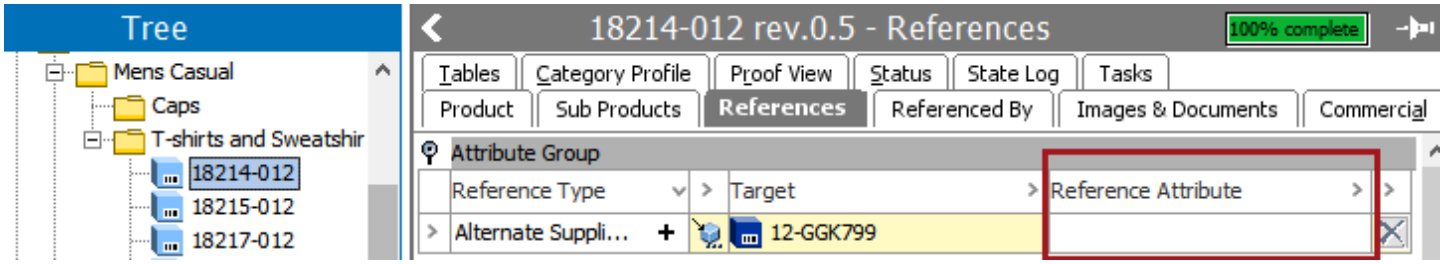


For more information, see **Validity on Description Attributes** in the **System Setup / Super User Guide**.

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.



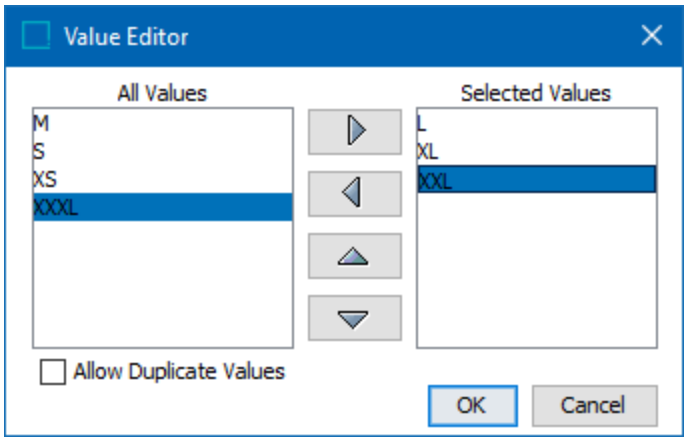
Edit a Metadata Attribute

Although the images below show a reference type, the same setup is used for editing metadata on a Product to Classification link type.

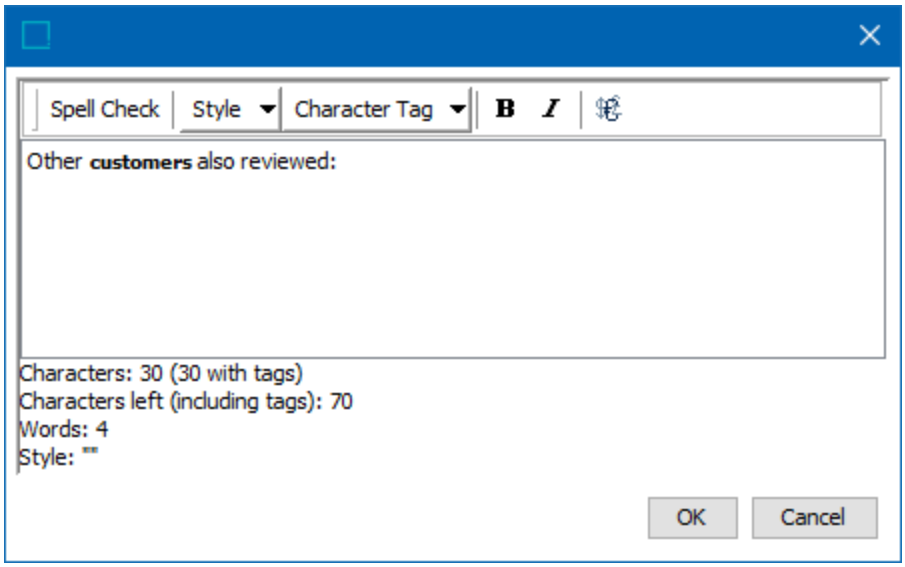
On the **References** tab, double-click an editable field (any white field as shown above) and edit using the available functionality based on the type of the metadata attribute.

- 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.

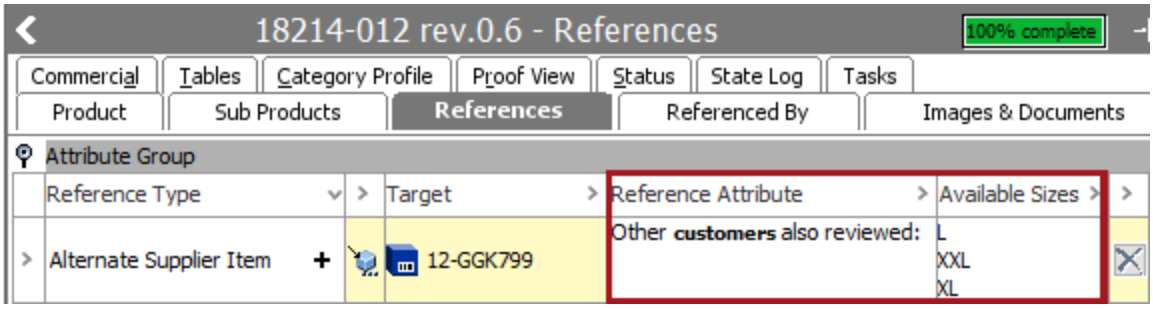
  Choose a value and use the necessary arrow button to include it in the Selected Values list, or to rearrange the order of the values.



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.

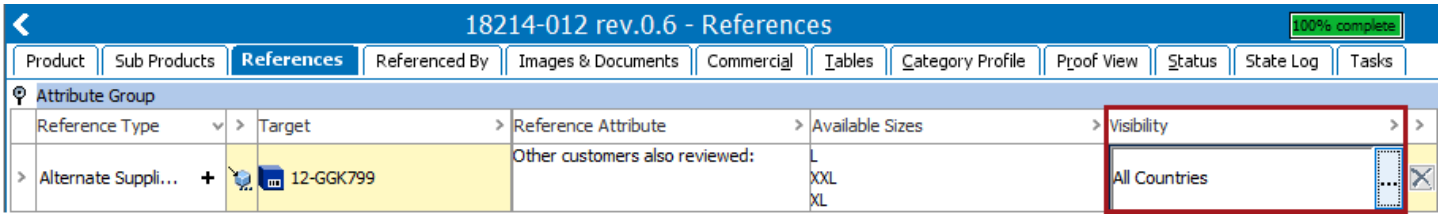
- Making the visibility global means the reference settings are inherited from the global context. For example, when all contexts require the same setting.
- Making the visibility local means the reference does not inherit its settings from the global context. For example, if you want to define attributes 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.

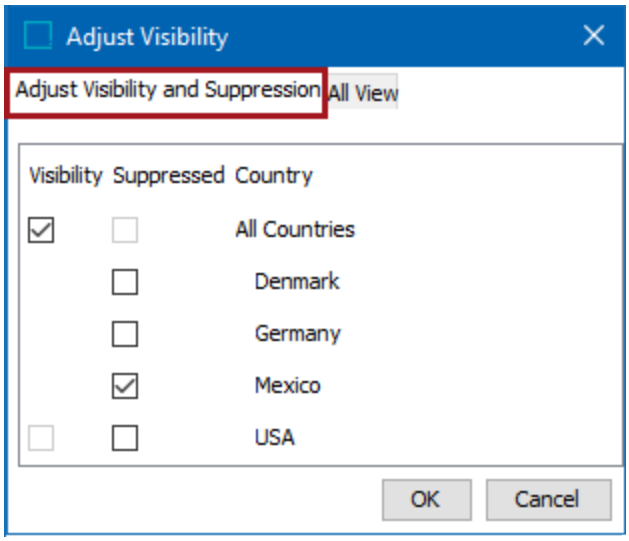
Note: The Visibility column is displayed on a reference when the General Aspect option 'Visibility' is added to the 'Included in View' panel on the View Definition of the attribute group. For more information, see the **Add a Metadata Attribute** section of the **Metadata Attributes on Reference and Link Types** documentation.

Setting Reference and Link Type Visibility

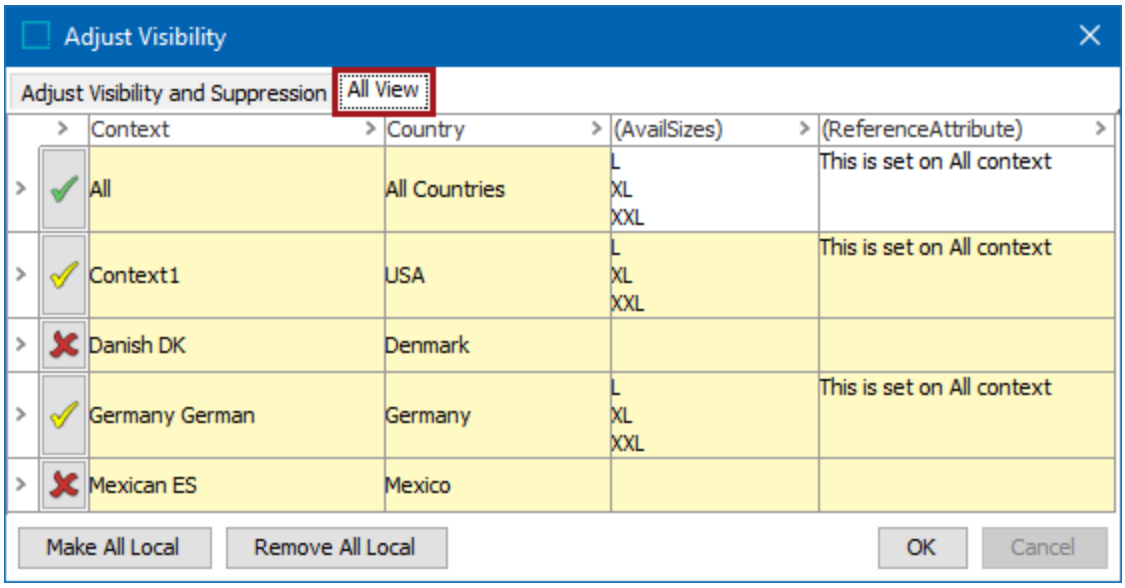
1. In the Tree, select the relevant node, click the **References** tab, and open the flipper that contains the dimension-dependent reference to be edited.



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.



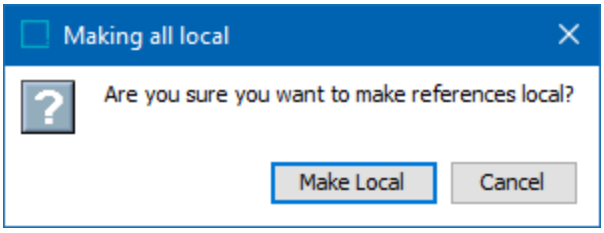
3. In the Suppressed column, check any contexts that should not display the reference.
4. Click the **All View** tab and note the status of the visibility of each context:
   - ✓ indicates that the metadata attribute is local and not being inherited (All). The setting is inherited by the contexts that 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).



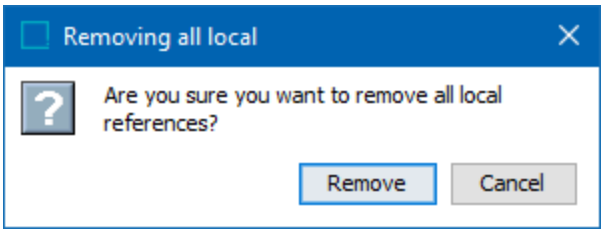
5. Modify the visibility of each required context as needed:
   - Click ✓ or ✗ to make the visibility of a reference local for only the selected context. Assuming the user has privileges to create the reference, the indicator changes to ✓, the reference value field changes to a white

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.

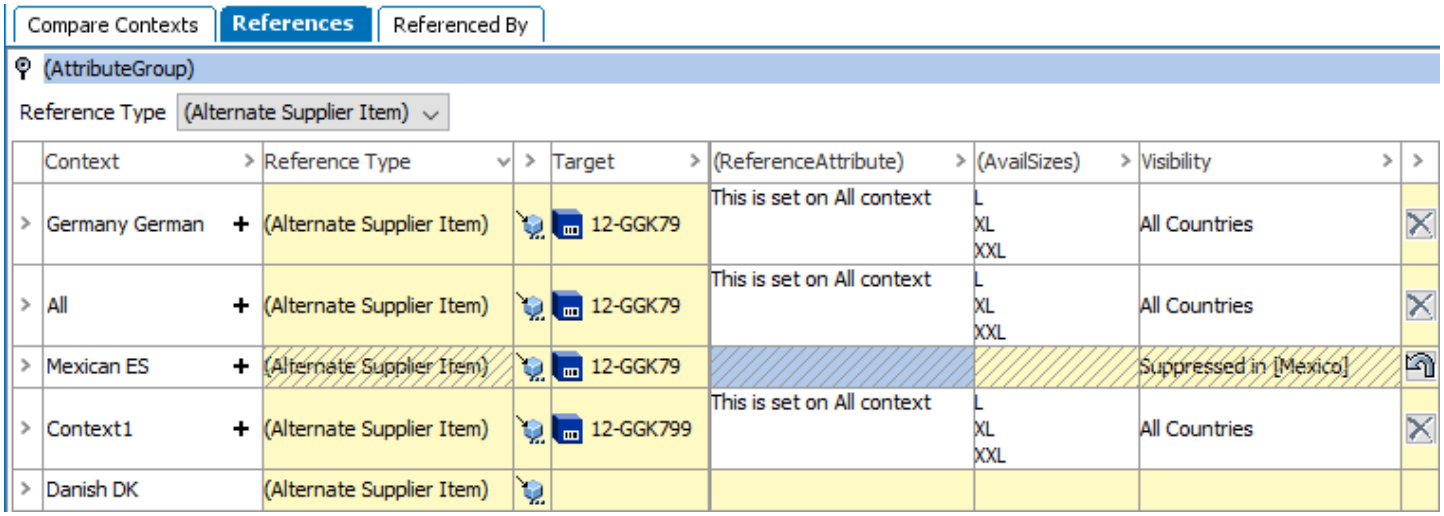


- Click Remove All Local to restore inheritance for all contexts currently local. In the 'Removing all local' dialog, confirm that you want to restore inheritance by clicking the **Remove** button. If a global context is local, all check marks turn yellow except the icon for the contexts that were previously 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.

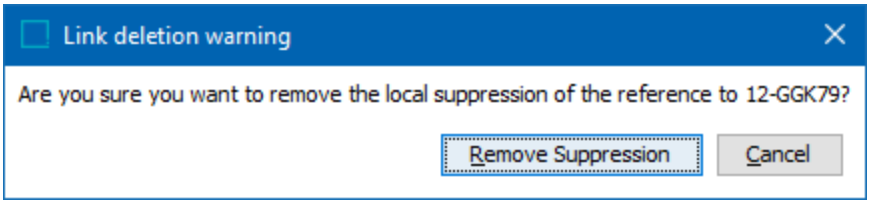


6. Verify the results of the visibility by checking the References page on the object and make any additional changes. Use the Context Mode by clicking in the toolbar.

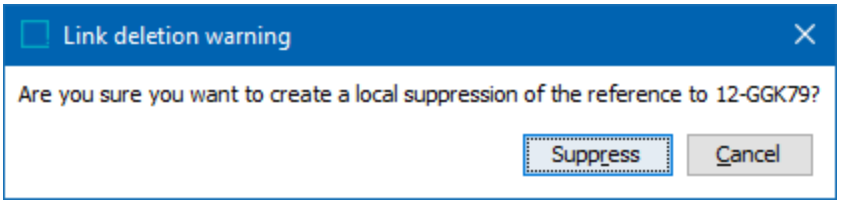
Note: Navigate to View > Target to select the Contexts for the comparison view to display.



- + 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 object 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 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 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**.

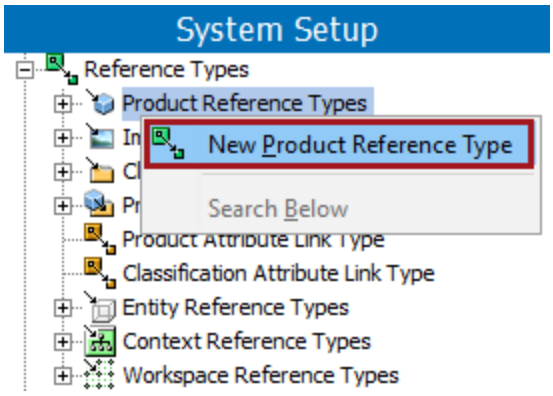
To create a reference type, follow the steps outlined in **Creating a Reference Type**.

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:

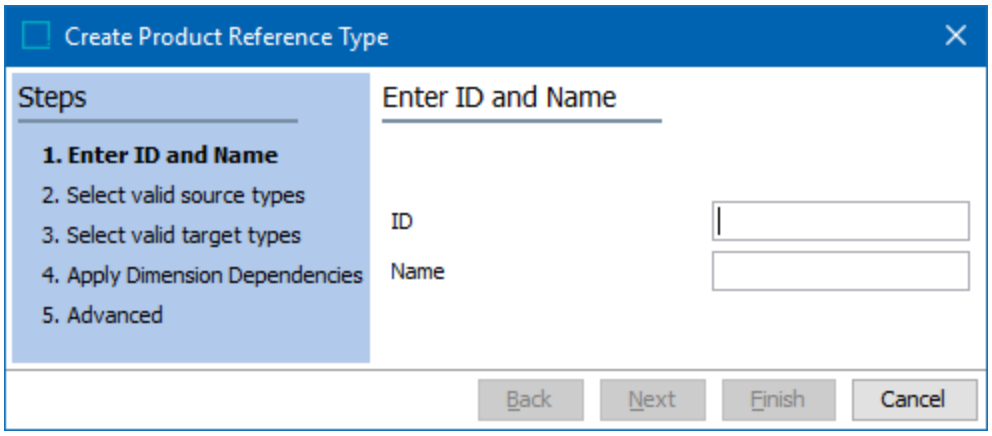
   - Product Reference Type > New Product Reference Type
   - Image and Document Reference Type > New Asset Reference Type
   - Classification Reference Type > New Classification Reference Type
   - Entity Reference Type > New Entity Reference Type
   - Context Reference Type > New Context Reference Type
   - 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.

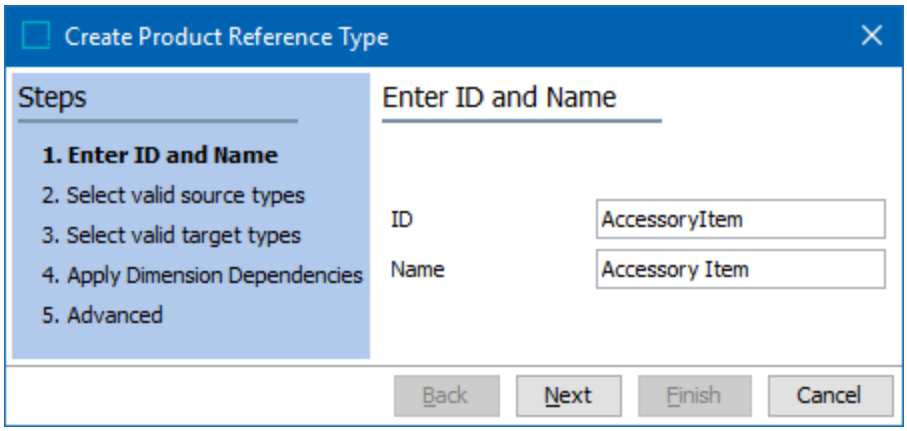


Each reference wizard includes the following steps:

- **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**.

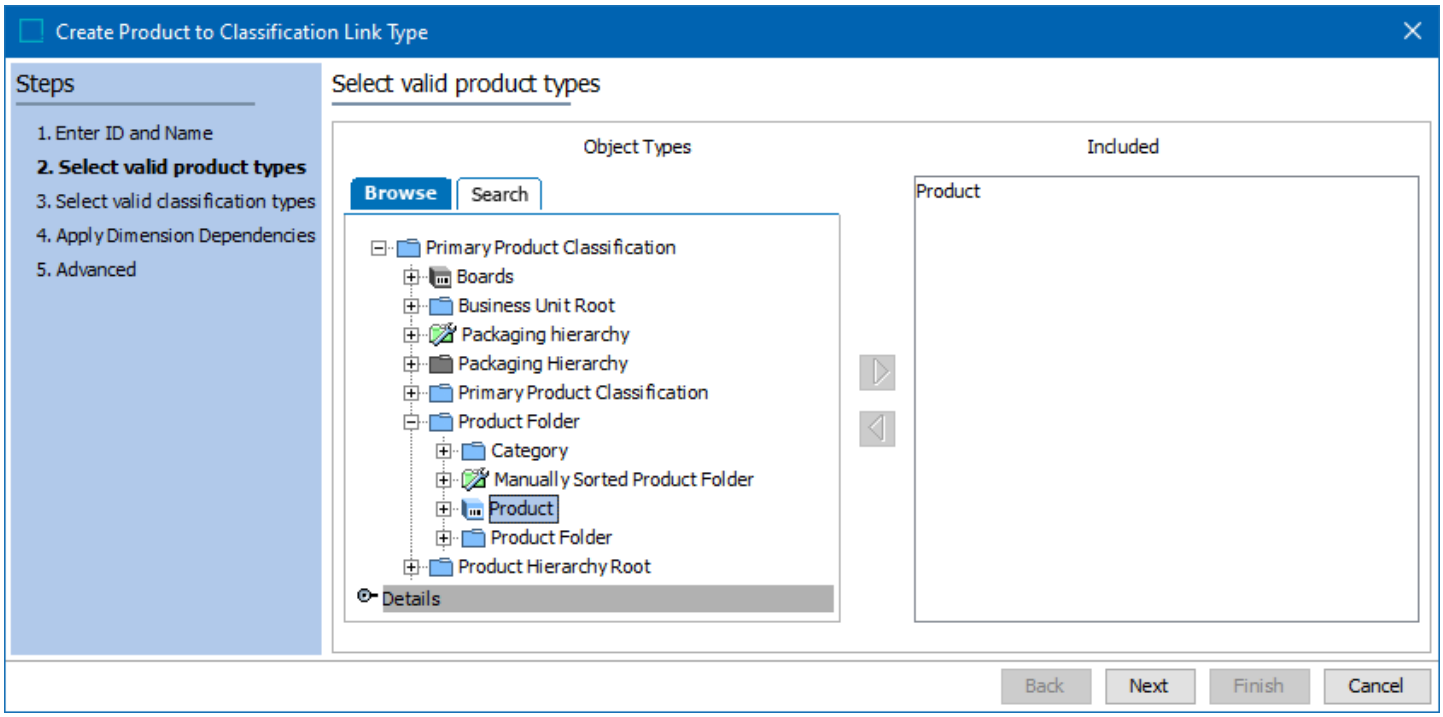
Reference Type - Enter ID and Name



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. If the ID is in use for any other configured reference type, an error will display notifying the user that the requested ID is already in use, and the wizard will not progress until a new ID is entered.
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



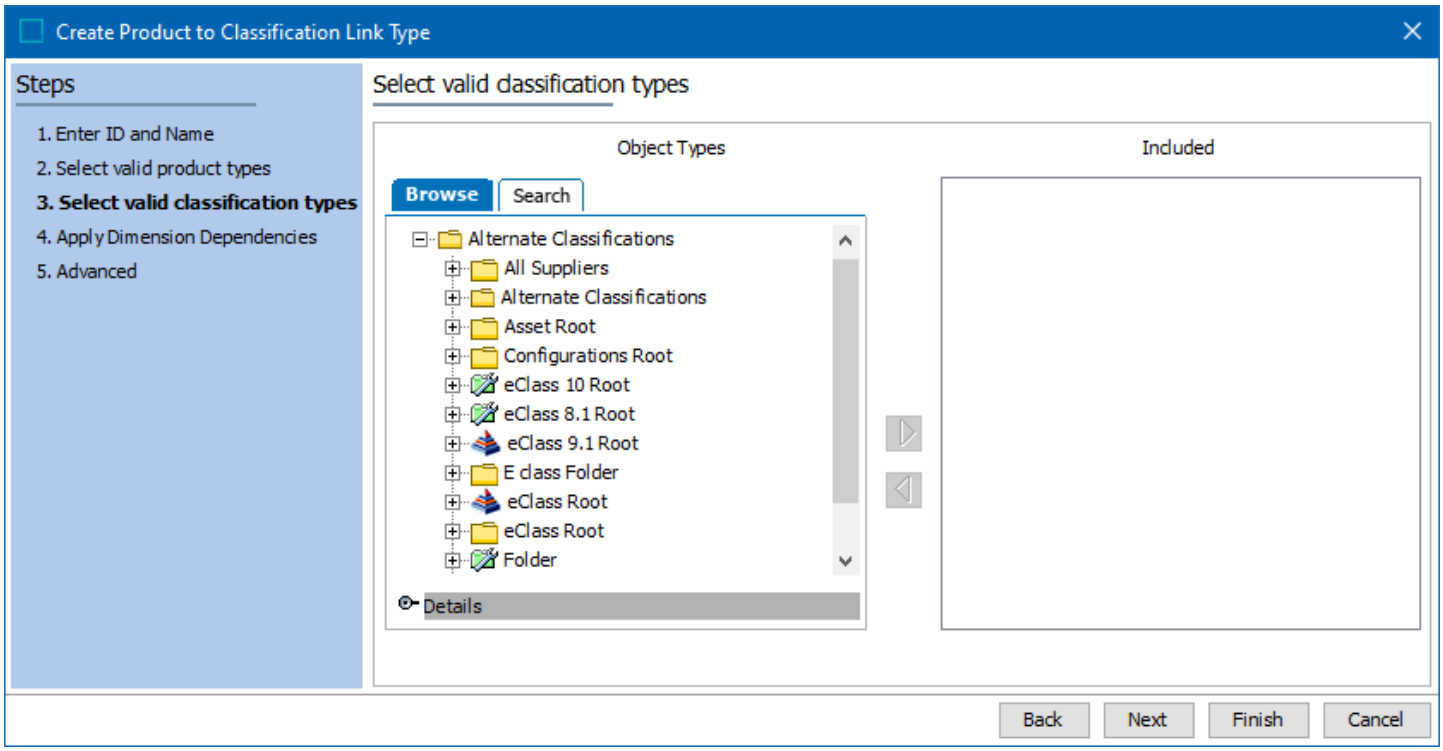
The contents of the Object Types pane displayed in this wizard step show only valid object types based on the selected reference type (Product, Image and Document, Classification, Entity, Context, and Workspace).

1. Browse through the hierarchy to select one or more source object types using the 'Browse' tab, or use the 'Search' tab to type the desired source object types and then select from a dropdown of typeahead-generated options. The 'source' object type selected defines which kinds of objects may reference other objects, i.e., which objects can start the reference. 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



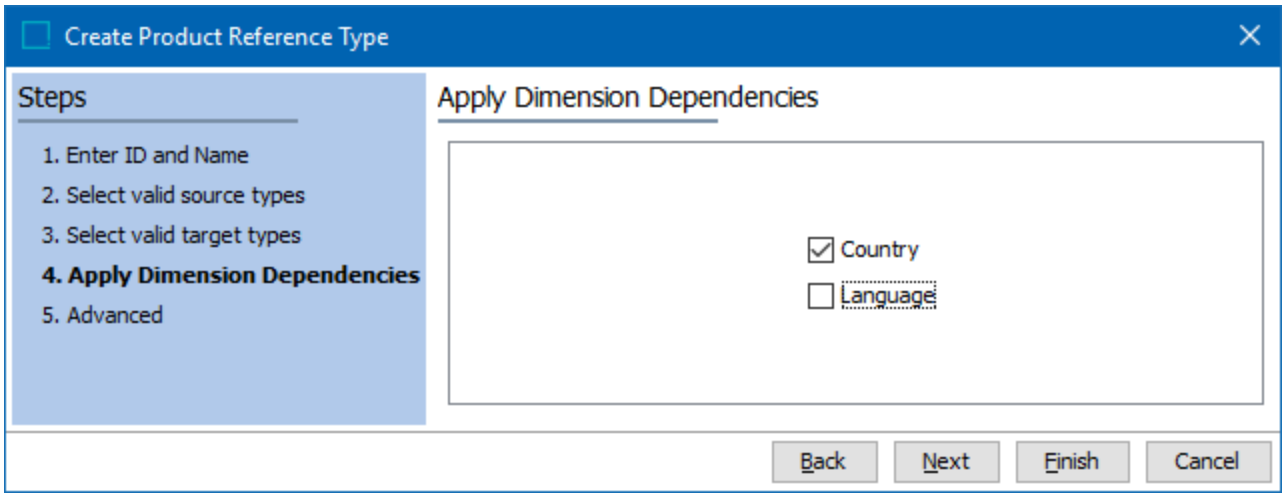
The contents of the Object Types pane in this wizard step displays only valid object types based on the selected reference type (Product, Image and Document, Classification, Entity, Context, and Workspace).

1. Browse through the hierarchy to select one or more target object types using the 'Browse' tab, or use the 'Search' tab to type out the desired target object type and then select from a dropdown of typeahead-generated options. The 'target' object type selected defines which kinds of objects may be referenced, i.e., which objects can end the reference. 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**.

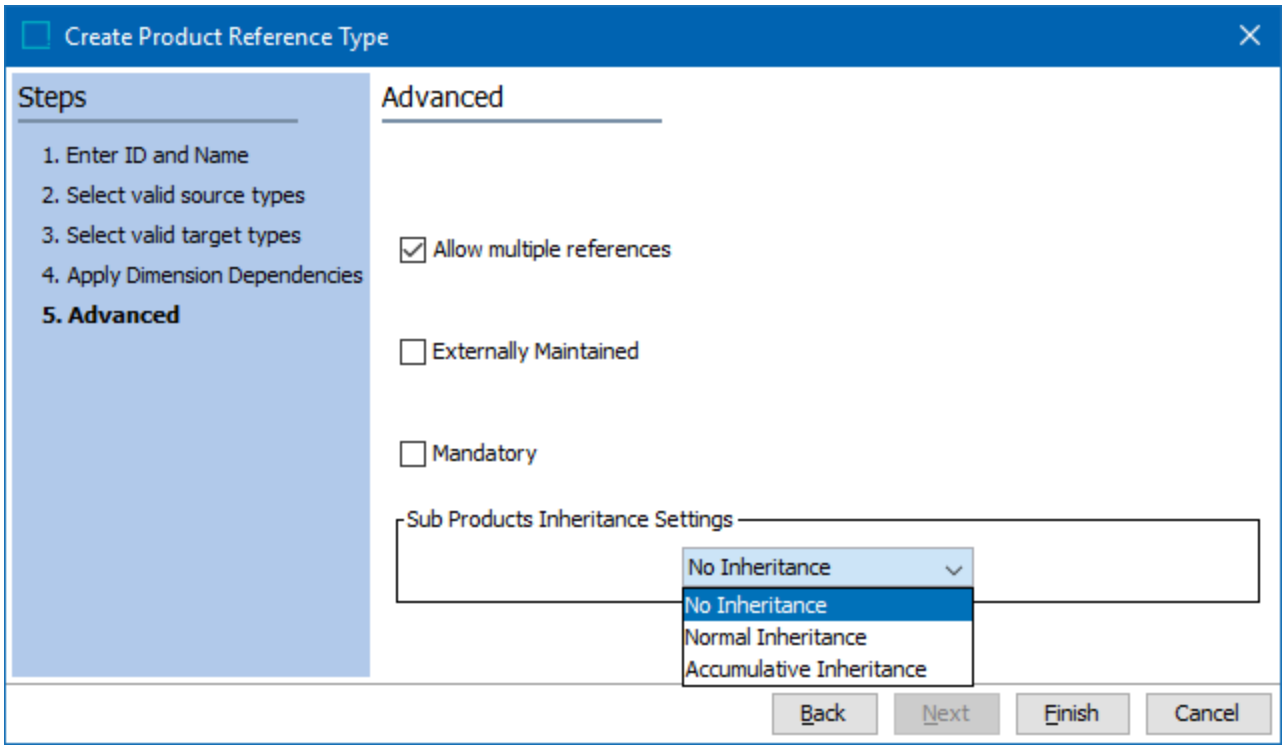
Reference Type - Apply Dimension Dependencies



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



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.

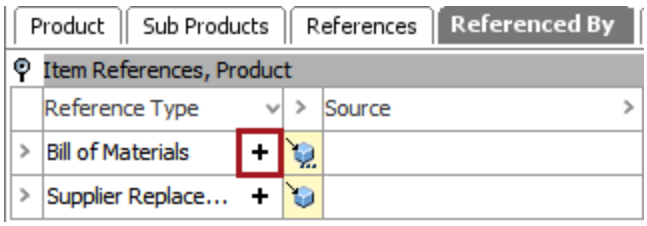
3. **Mandatory** determines if the reference type is required for approval of the source object. For more information, see the **Maintaining a Reference Type** documentation.
   - **Checked** means the reference type is required from a source object to a target object before the source object can be 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

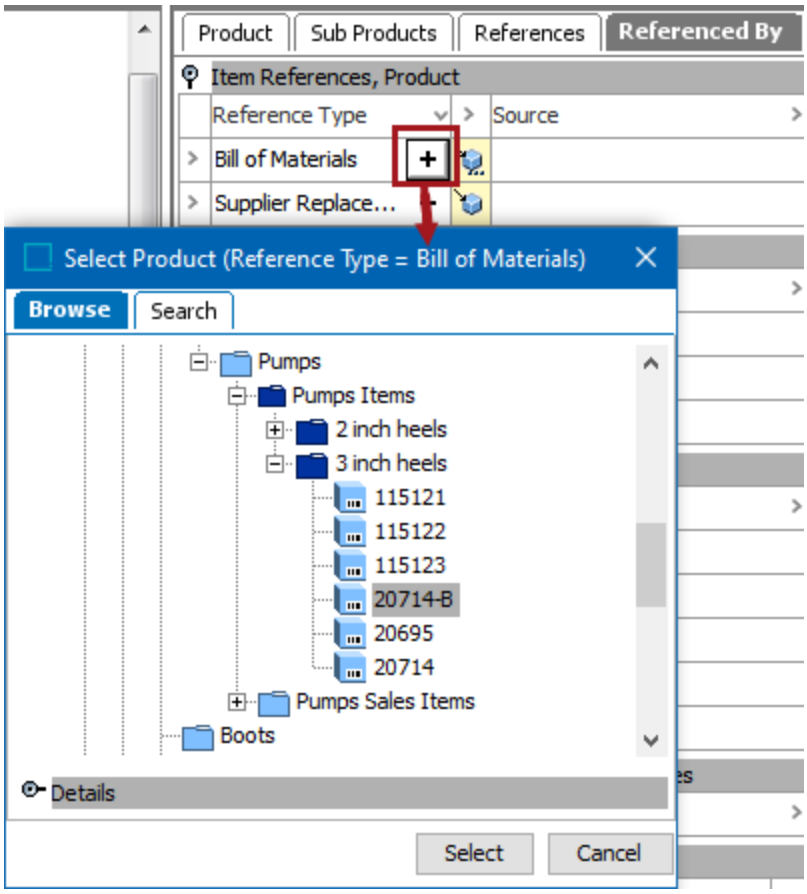
After creating a reference type, you can add that type on an object's References or Referenced By tabs.

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.

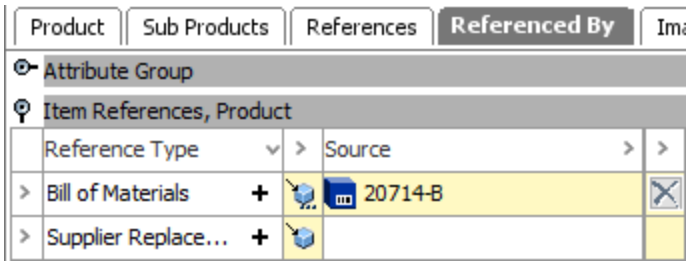
1. To add references to either the **References** or **Referenced By** tab of an object, use one of the following ways to find the reference type in a flipper:
   - If the reference type has been added to an attribute group, it appears under a flipper that is named the same as the attribute group, as 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.



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.



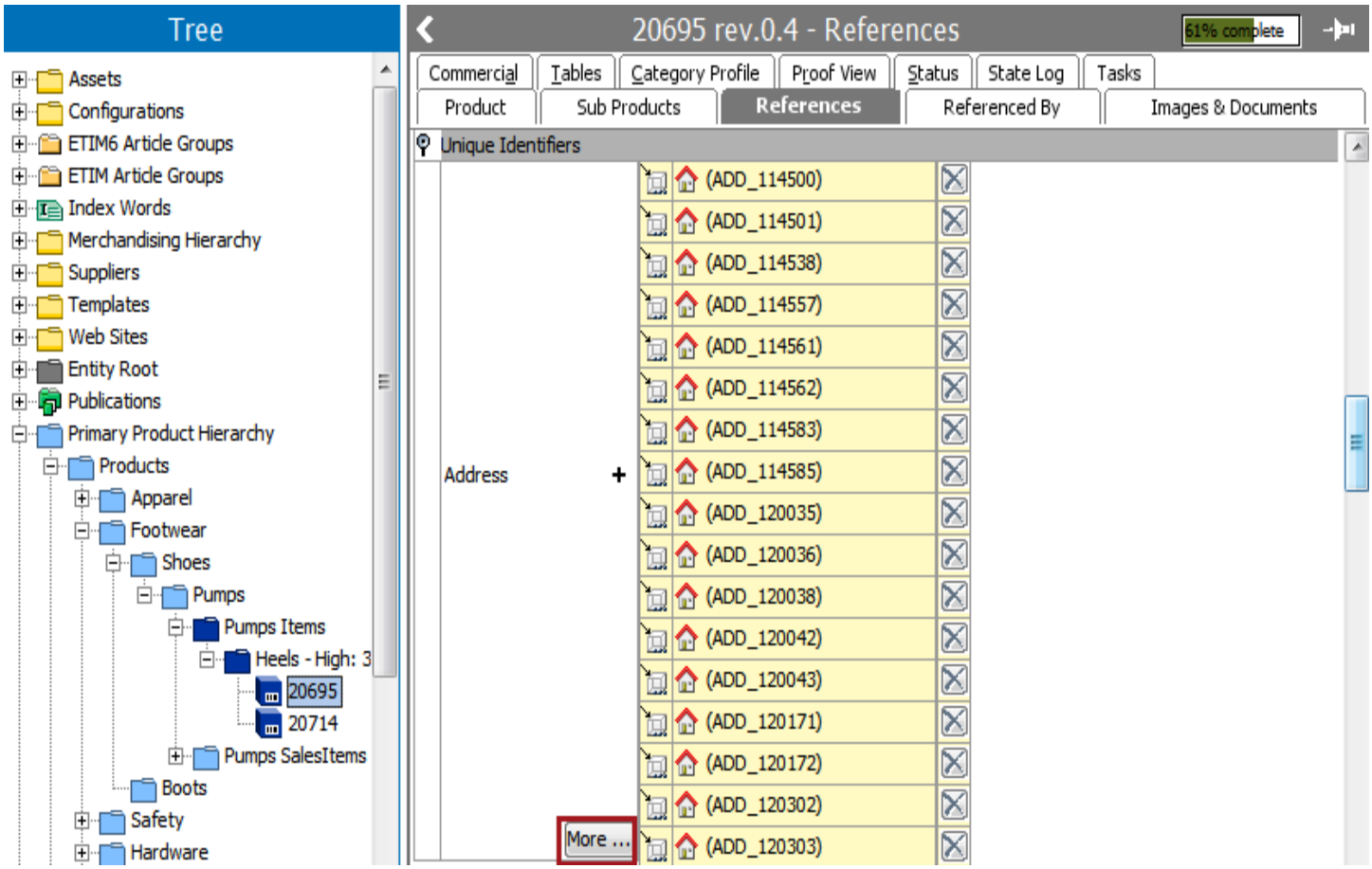
4. Once selected, the object displays as an added reference.



- 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.

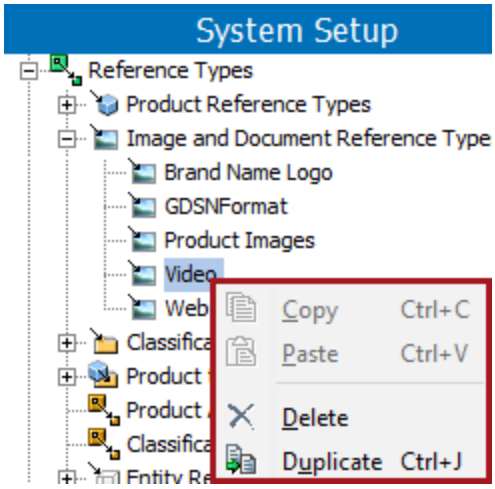


Repeatedly clicking the button doubles the number of references shown each time, until the full set is displayed. Note that if new references are added, they display on the list immediately indicating that they were added properly, though normal functionality will resume if the user presses the 'More...' button again. This means that items may become hidden until the list is fully expanded.

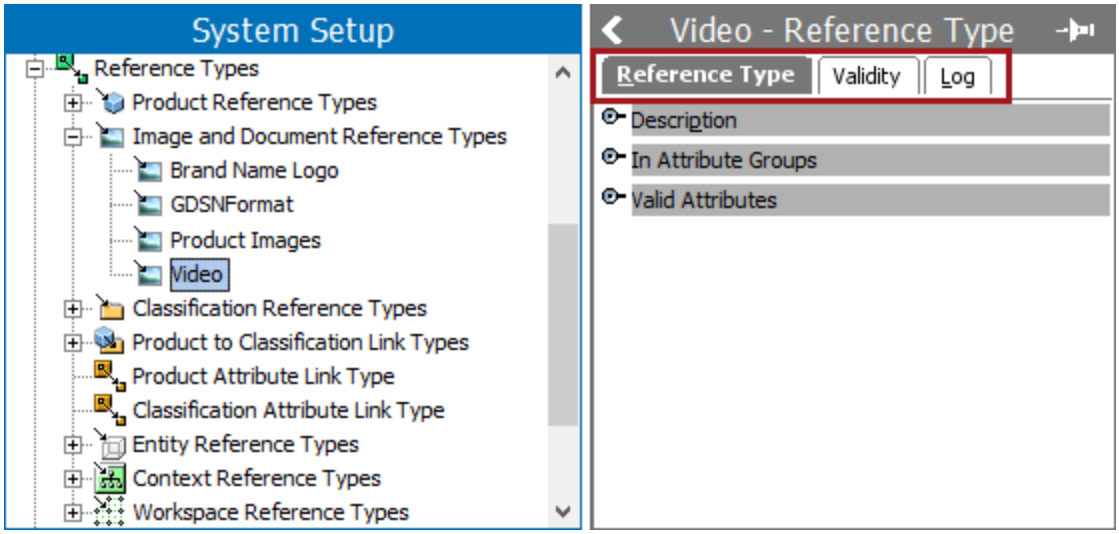
After the list has been fully expanded for the given reference type, it stays expanded until the user refreshes the page or clicks away to another object in workbench.

Maintaining a Reference Type

Select a reference to display the Reference Type editor and make updates to the reference type settings. The right-click menu provides access to copy, paste, delete, or duplicate a reference.



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.

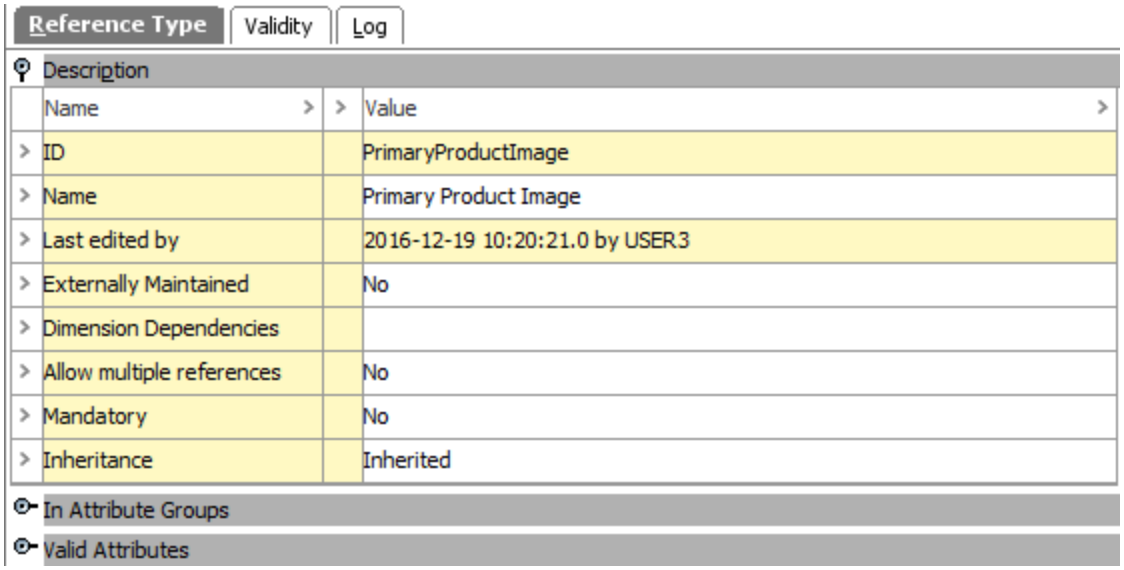


Reference Type Tab

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.



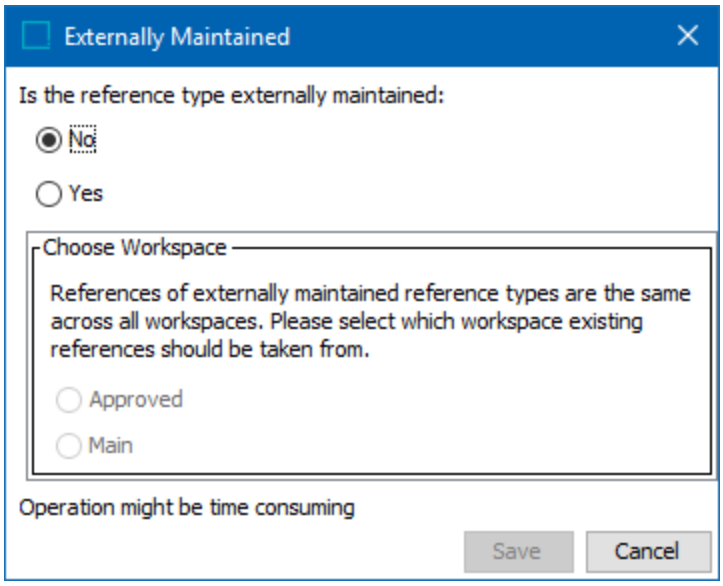
All fields except the ID and 'Last edited by' fields can be edited. The method required to edit a parameter in the references editor is determined by the parameter type:

- 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.

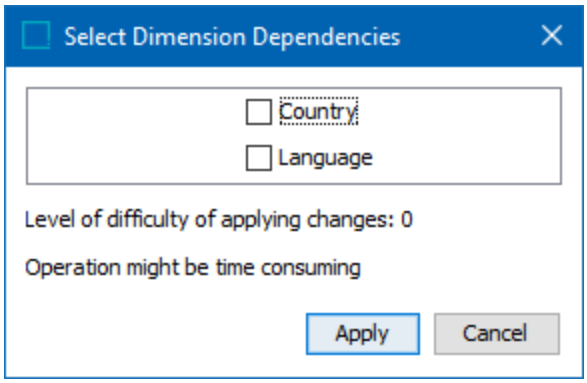
  For example, references to vendor logos are imported from an ERP system and are not under revision control. To address this case, an image reference type named 'Vendor Logo' is set to be externally maintained, which means that:

  - 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.

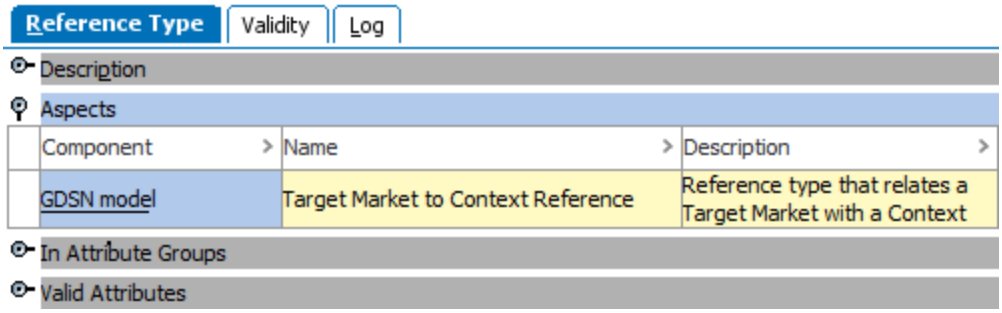
  For example, a product reference type named 'Part of Kit' is set to be mandatory and is valid for source object types 'Product Family' and target object types 'Articles,' which means that:

  - 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 under the Component Models node. For more information, see the **Component Models** topic.

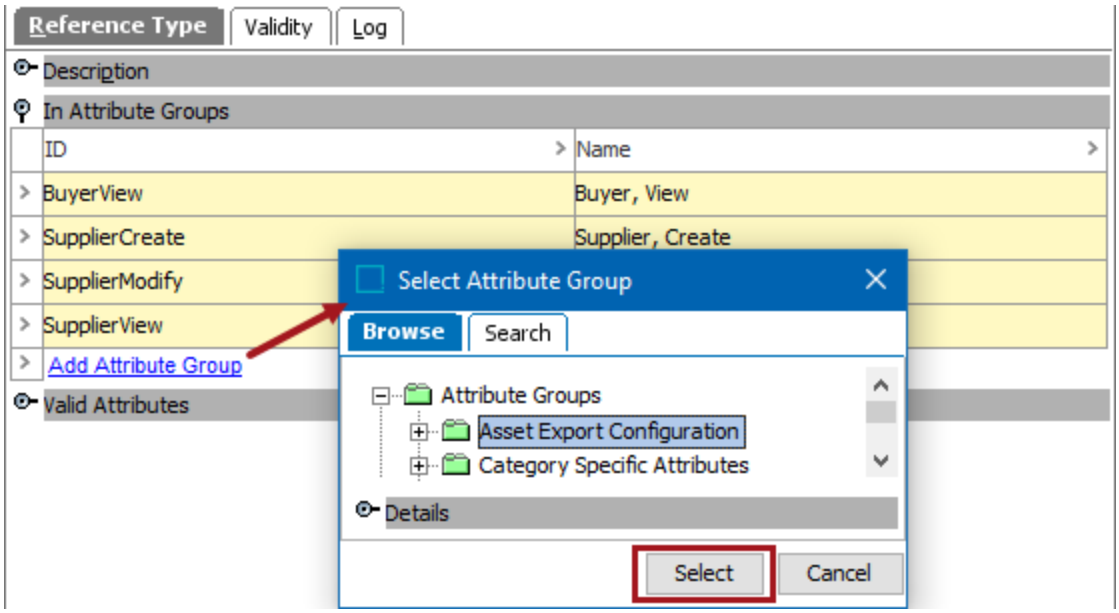
No edit option is available on the Aspects flipper. Changes to component model references must be made directly on the component model. Click the link in the Component column to display the component model using the reference.



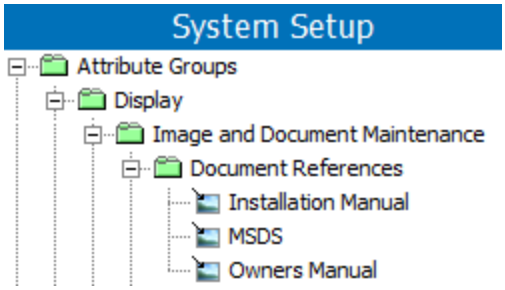
In Attribute Groups Flipper

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.

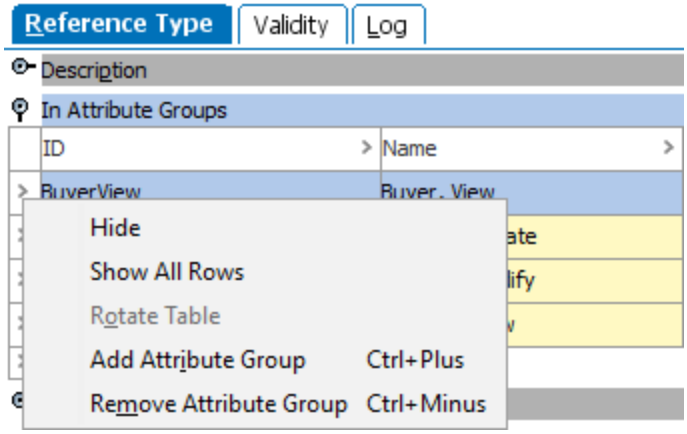
- To add an attribute group, click the **Add Attribute Group** link to display the Select Attribute Group dialog. Choose a group and click the Select button to add the reference to an attribute group.



The reference type is also displayed within the Attribute Groups node on System Setup.



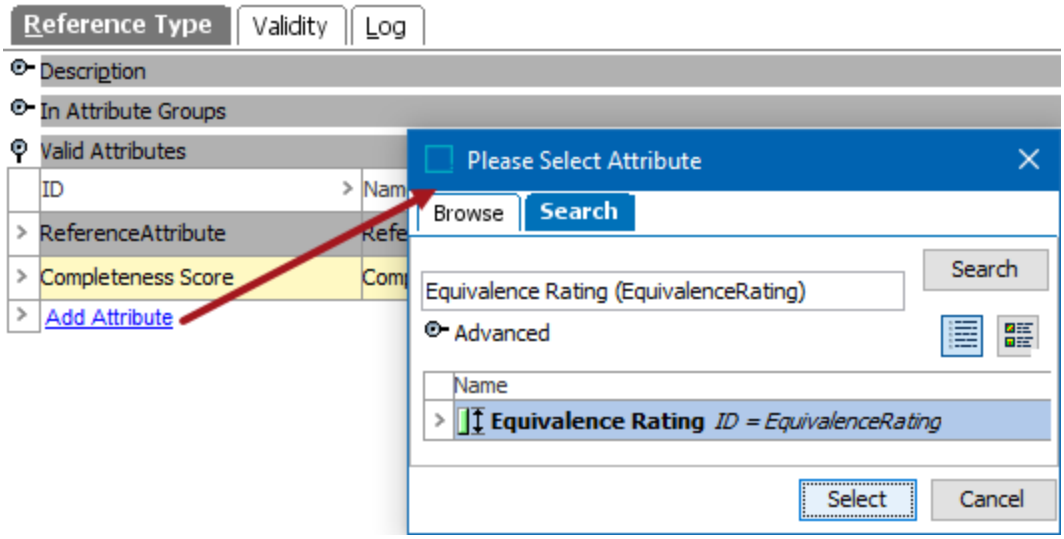
- To modify the attribute groups included or displayed, right-click the arrow to the left of the ID and select an option from the menu.



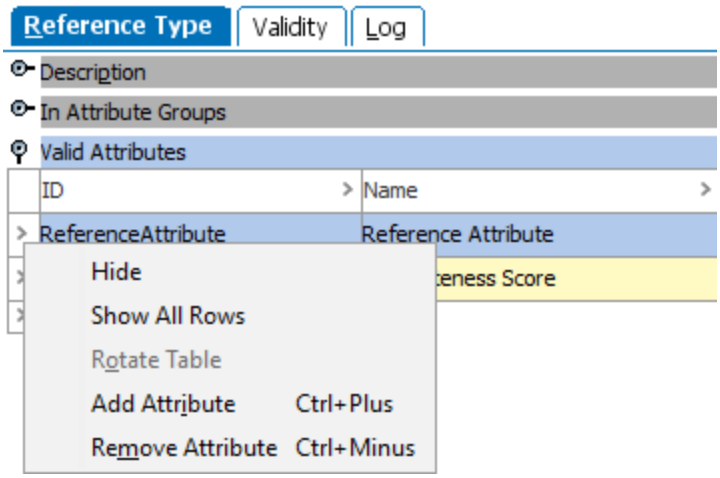
Valid Attributes Flipper

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.



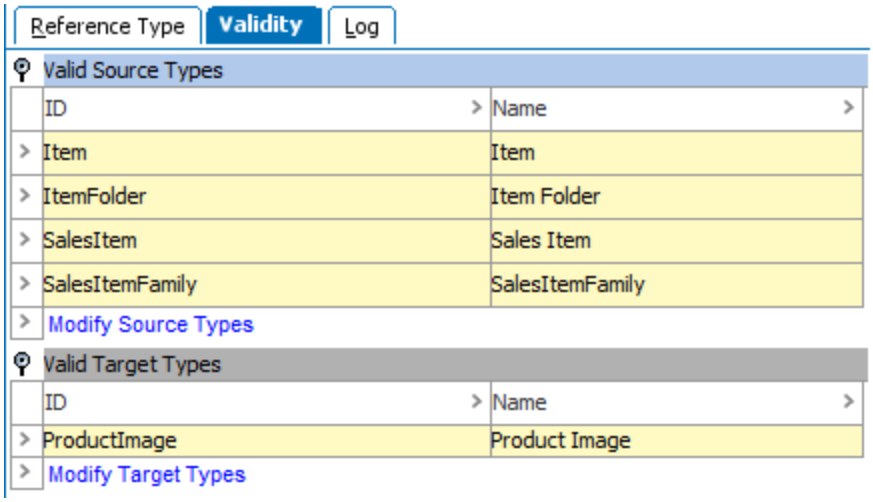
- To modify the metadata attributes included or displayed, right-click the arrow to the left of the ID and select an option from the menu.



For more information, see **Metadata Attributes on Reference and Link Types** documentation.

Validity Tab

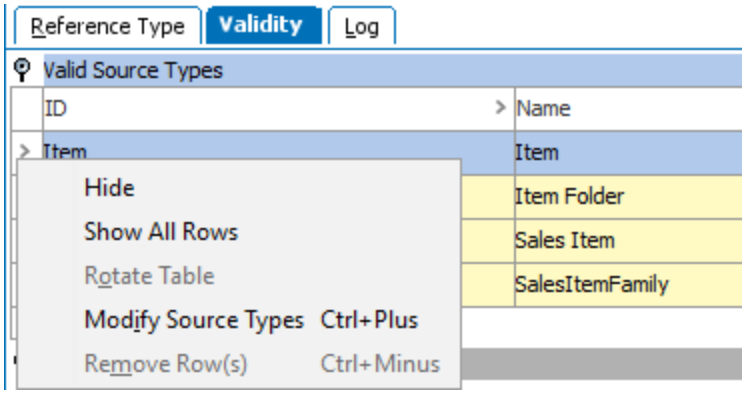
The Validity tab stores the source object types and the target object types allowed on the selected reference type.



- 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.

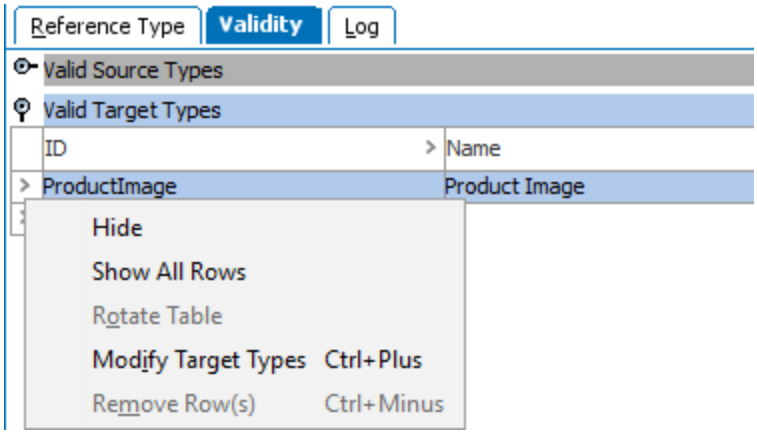


- For **Valid Target Types**, any number of target object types can be added, but the object types available are restricted based on the selected reference type. Click the 'Modify Target Types' link to add or remove an

object type. This parameter is discussed in the **Reference Type - Select Valid Target Types** documentation.

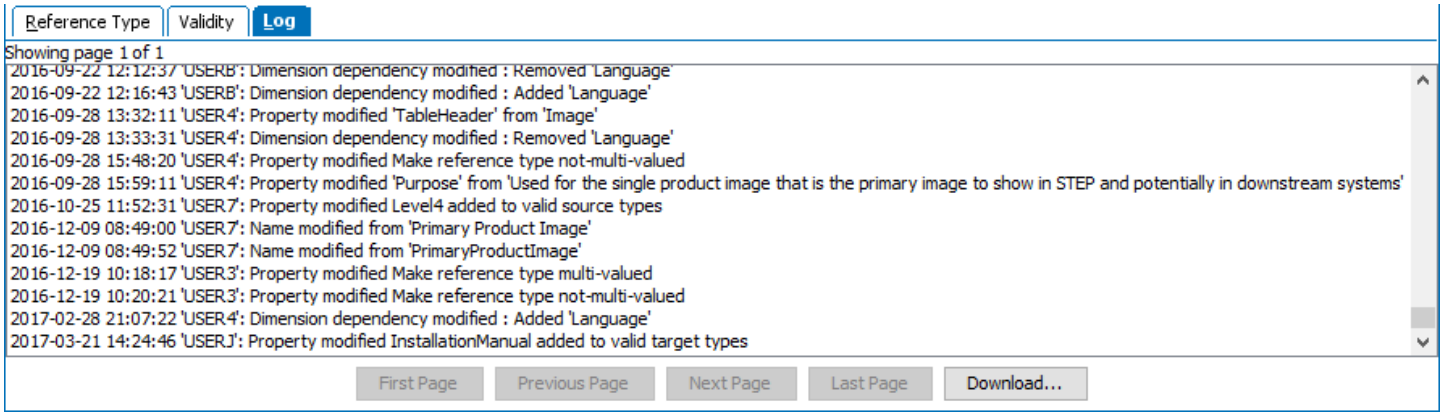
Note: When an object type is removed from the list, if a reference of this type already exists for the deleted object type, the links are deleted.

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.



Log Tab

The Log tab stores the information about the changes that have taken place on the reference type since it was created.



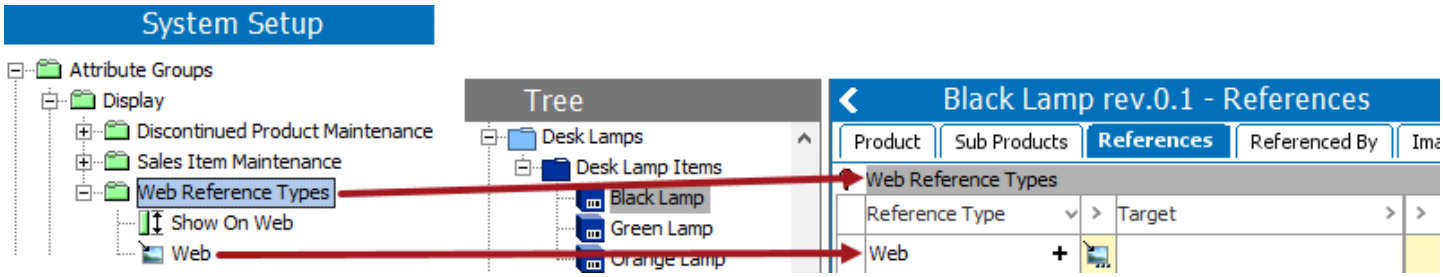
As additional log entries become available and the page is filled, the pagination buttons are enabled. Additionally, you can download the complete log as a zipped text file.

Attribute Groups for Reference and Link Types

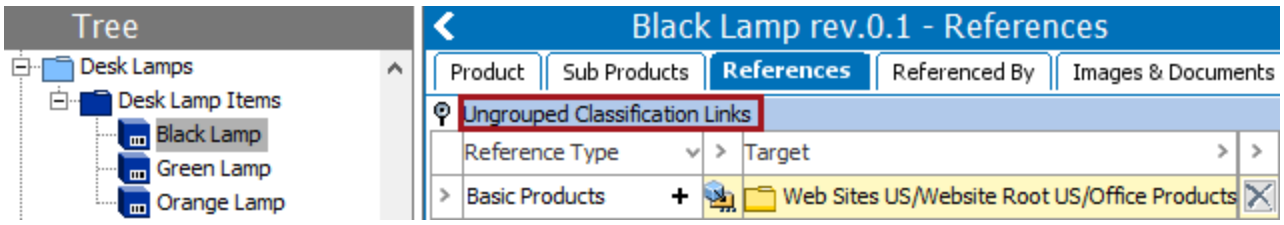
Adding a Reference or Product to Classification Link type to an attribute group provides easy identification, a customized view, and control of approvals via user privileges.

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.

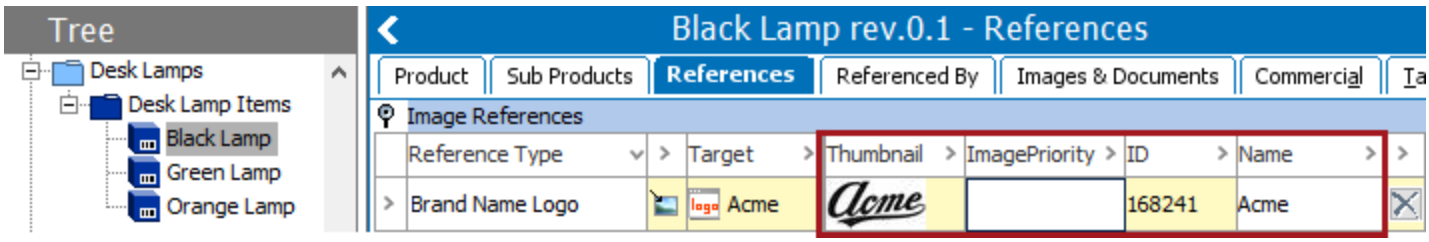


Otherwise, the reference or link displays in a flipper with the 'Ungrouped' label.



Customized View

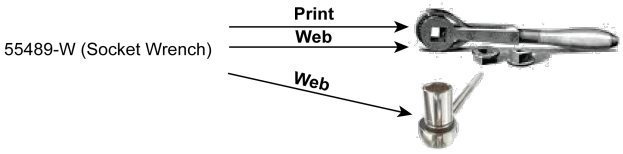
A customized view can be defined on the attribute group of all reference or link types in the group to display additional information on the reference or link. For more information, see the **Creating a Customized View** section in the **Attribute Groups** topic of the **System Setup / Super User Guide** documentation.



Control of Approvals via User Privileges

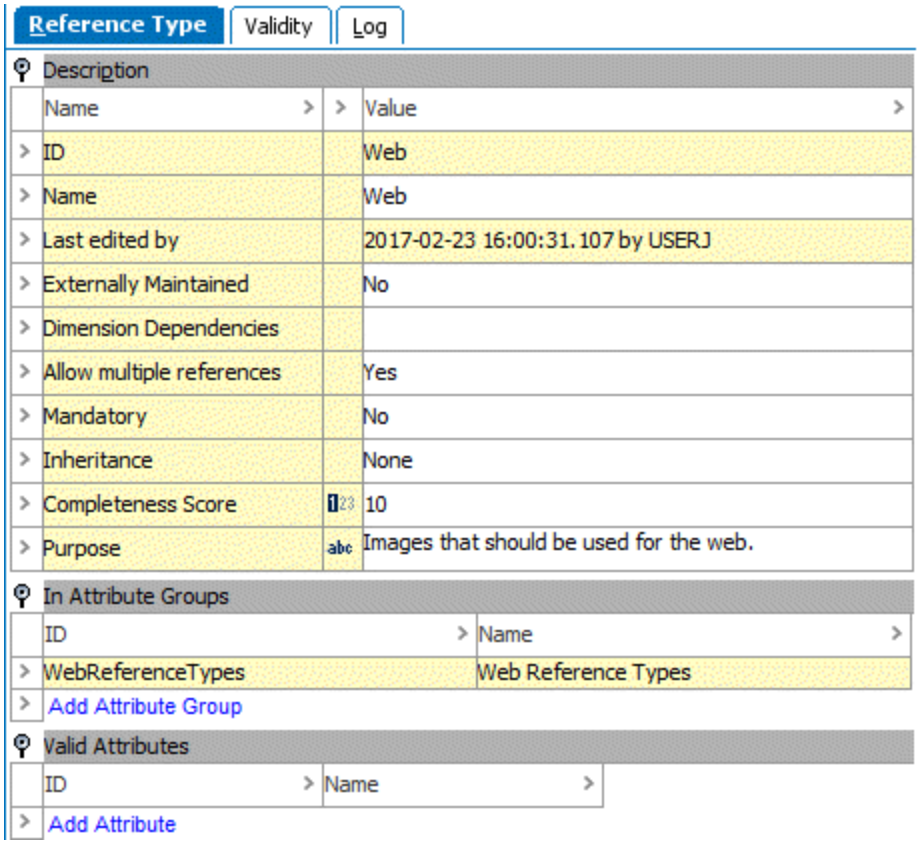
Perhaps most importantly, an attribute group allows you to control user privileges for approval of the reference or link. This means that approval of specific types of references and links is based on the user's access to the attribute group, so that users will only be allowed to approve references or 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.

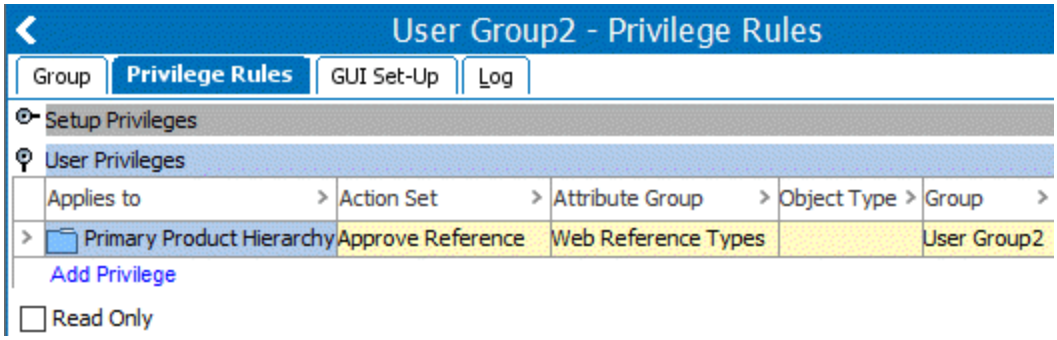


To implement this, the reference type named 'Web' is added to an attribute group named 'Web Reference Types.'

A reference or link type can be added to as many attribute groups as needed by different users with different privileges.



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.



A user with this privilege setup will only be allowed to approve references or links in the attribute group 'Web Reference Types.'

For more information on adding a Reference or Product to Classification Link type to an attribute group, see the **In Attribute Groups Flipper** section of:

- **Maintaining a Reference Type**
- **Maintaining a Product to Classification Link Type**

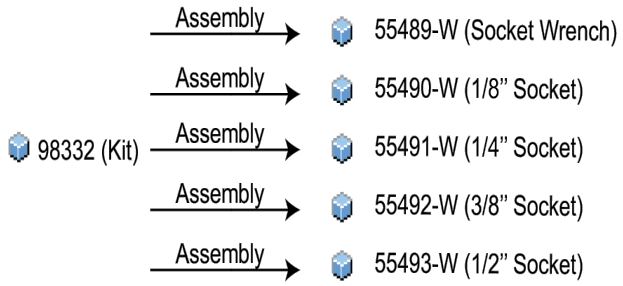
Direction of a Reference

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

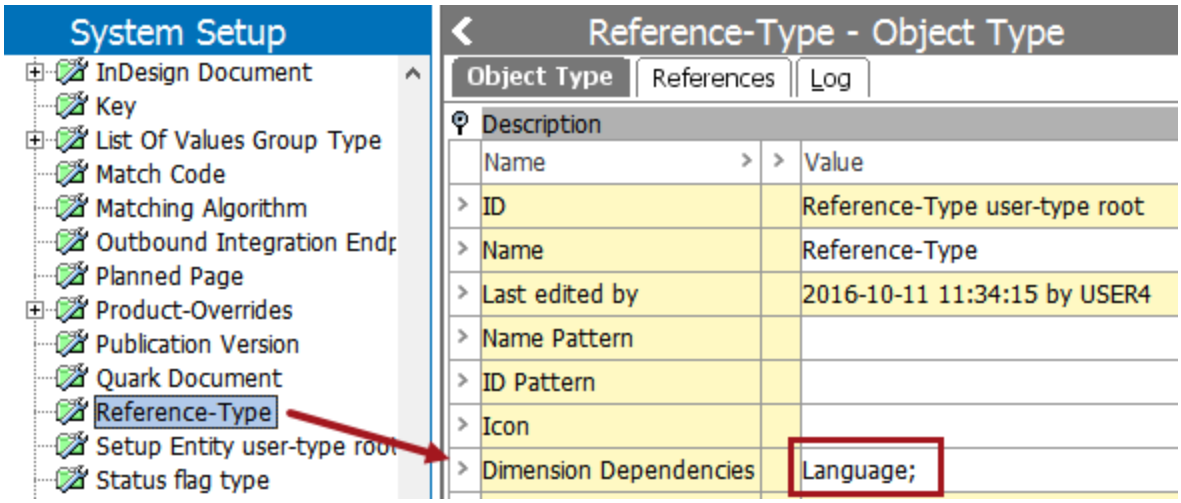
A Reference or Product to Classification Link type can be set up to be dependent on a dimension. Dimension dependency is used to specify the dimension points, and by extension, the contexts, where a reference or link should display or should be suppressed. A reference or link can only be dependent on 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.



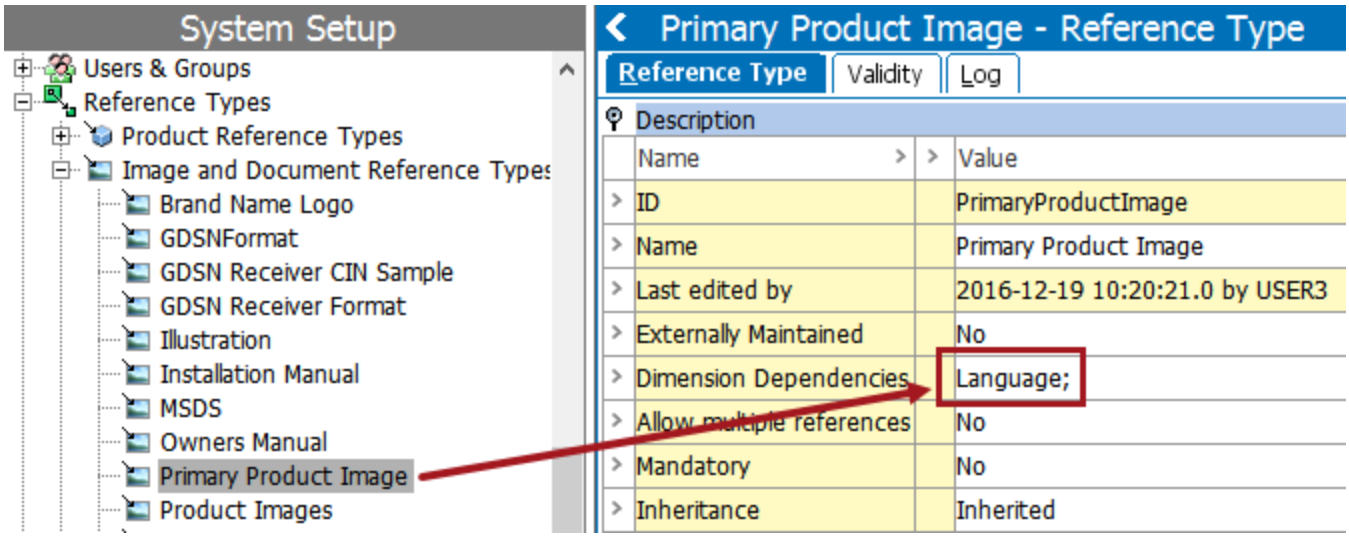
Using the Dimension Dependencies parameter on the Reference-Type object type is a global setting that enables translation of reference type names and requires that all reference type names be translated. Translated

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.



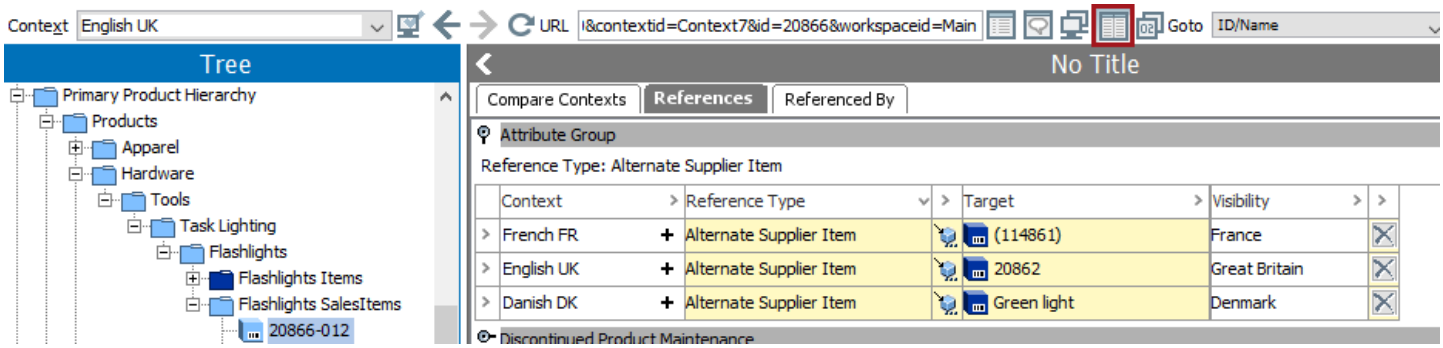
This is not a typical setup because the References / Referenced By tabs can become difficult to manage due to the layout required when viewing multiple contexts.

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

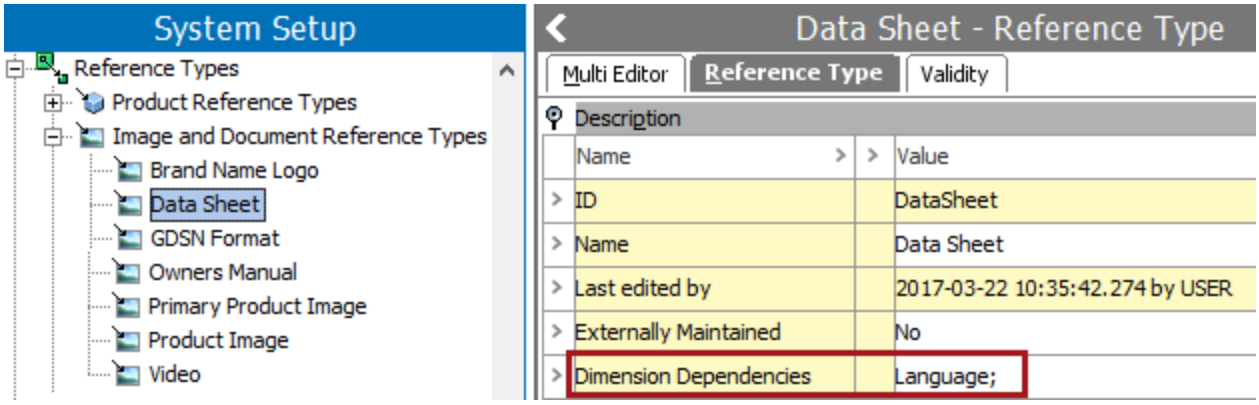
When dimension dependency is applied to a reference type, values can be applied based on the selected dimension. For example, the following reference is dependent on the Country dimension, so each country can have a different setting. The Context view displays all values for each country, based on the targets selected from the View > Target menu option.



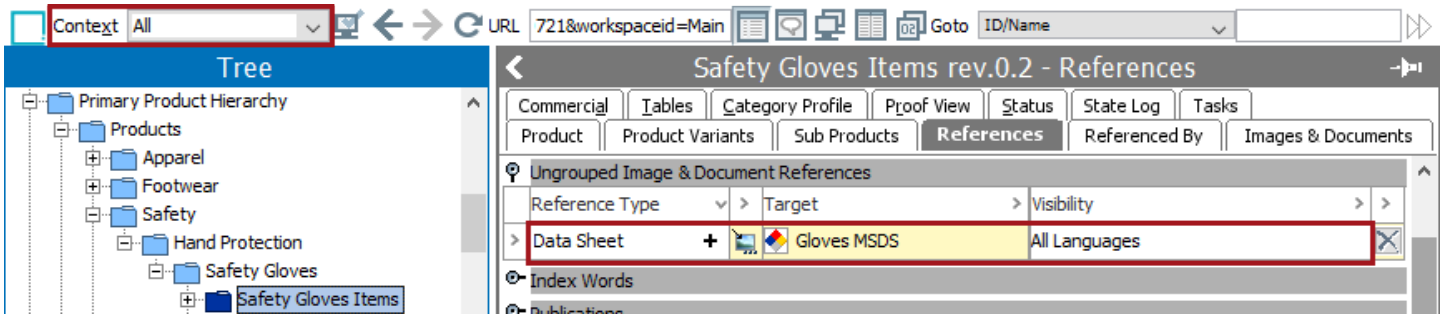
Dimension Dependent Reference Example

Consider a product that needs a 'Data Sheet' reference. The reference should be linked to the product for all languages except the English. Use the following steps to accomplish this:

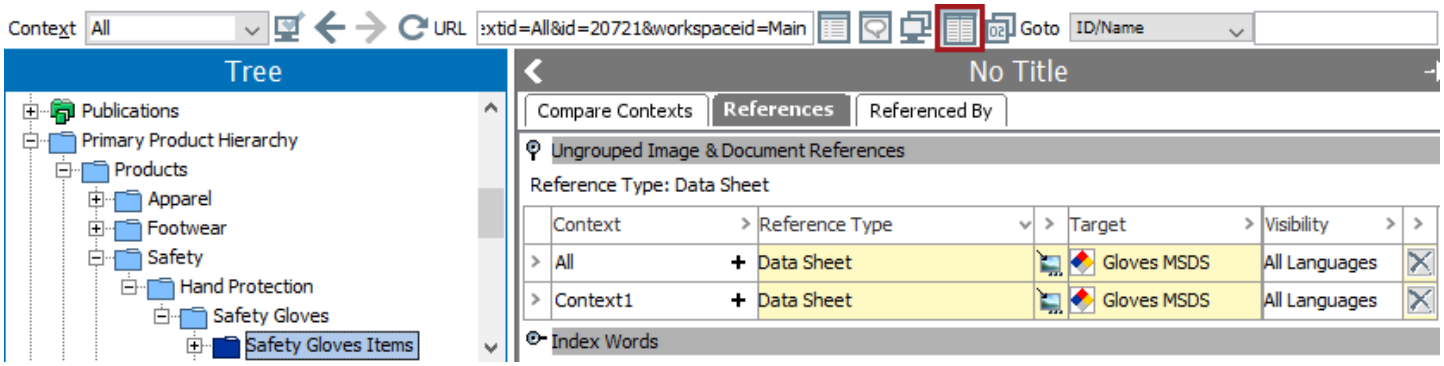
1. Create a reference type named 'Data Sheet' and set it to be language dependent.



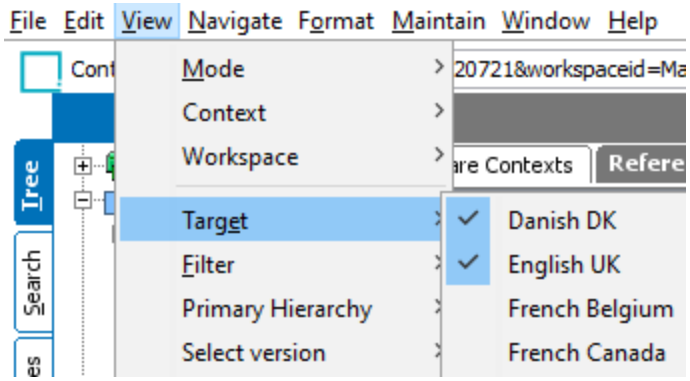
2. Select the 'All' context, add a Data Sheet reference. The reference is inherited to all languages.



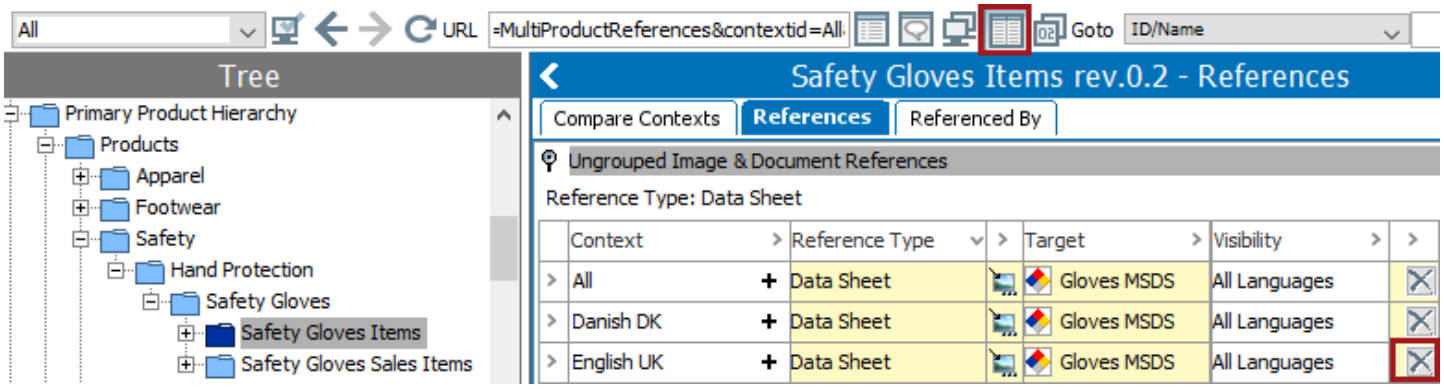
3. Display the References tab in Context Mode by clicking in the toolbar.



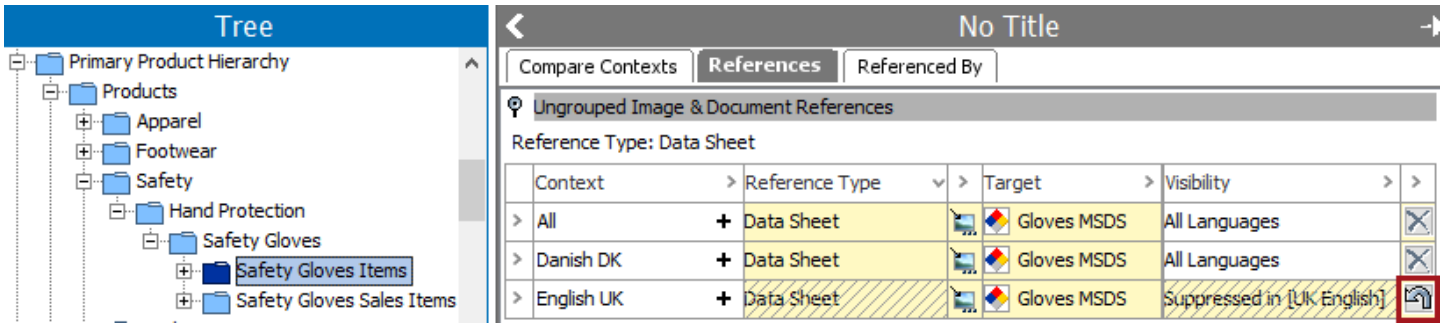
4. Navigate to View > Target and select the Contexts for the comparison view to display.



5. Click the X button to suppress the reference for the English language.



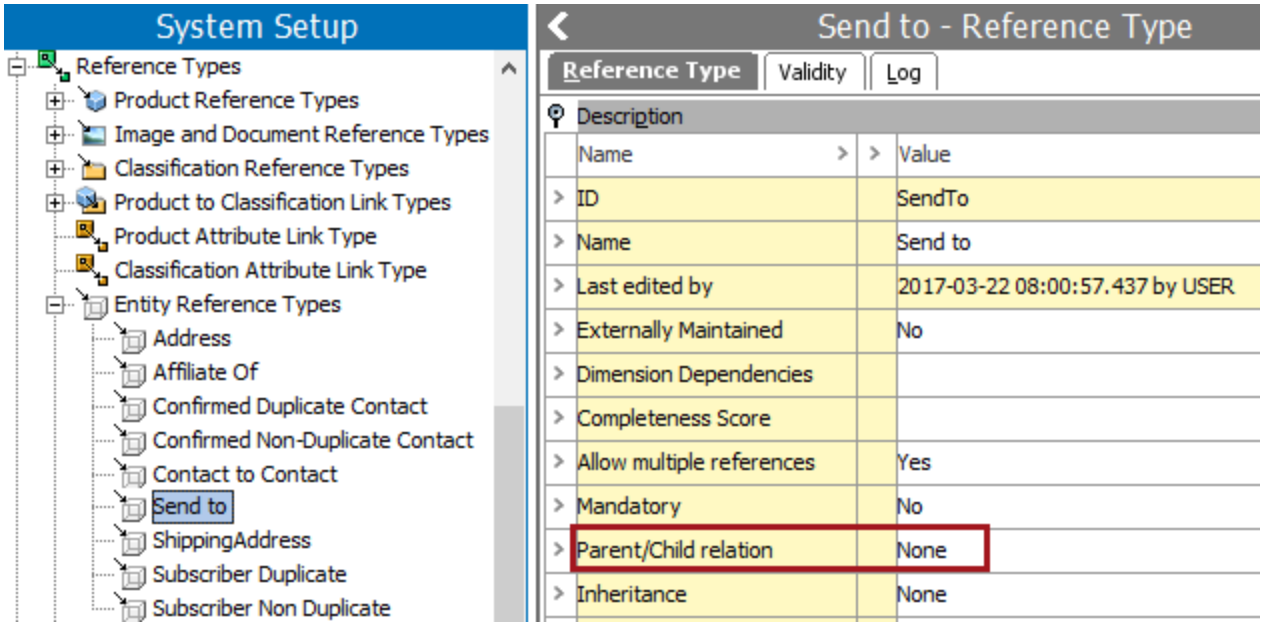
6. The suppressed reference type now appears shaded. If necessary, the reference can be restored by clicking the undo button on the shaded row.


Tree
Primary Product Hierarchy
Products
Apparel
Footwear
Safety
Hand Protection
Safety Gloves
Safety Gloves Items
Safety Gloves Sales Items
No Title
Compare Contexts
References
Referenced By
Ungrouped Image & Document References
Reference Type: Data Sheet
Context
Reference Type
Target
Visibility
All
Data Sheet
Gloves MSDS
All Languages
Danish DK
Data Sheet
Gloves MSDS
All Languages
English UK
Data Sheet
Gloves MSDS
Suppressed in [UK English]

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.



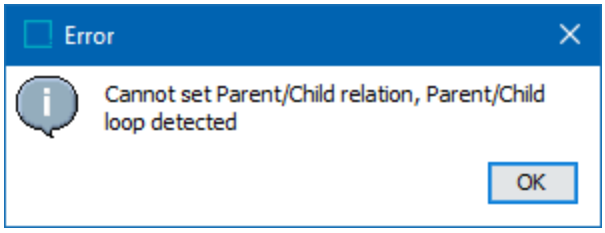
The **Parent/Child Relation** field includes these options, which are illustrated in the 'Parent/Child Relation Example' section below:

- **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** documentation.
- Aggregating references enables **Copy**, **Paste**, **Drag**, and **Drop** between entity hierarchies. STEP will automatically apply the reference at the new position.

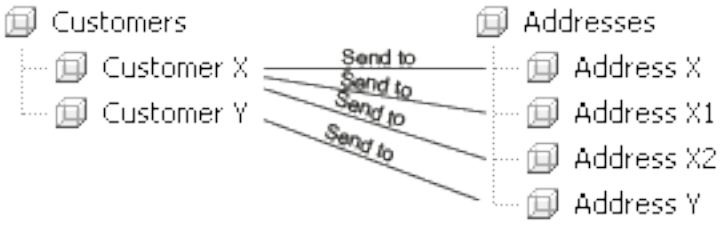
- Entity objects types can be set up to be globally revisable or internally revisable. For more information about this functionality, see the **Revisability on Entity Object Type** documentation.
- STEP prevents you from setting the Parent/Child Relation field if the objects that are parent and child 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).

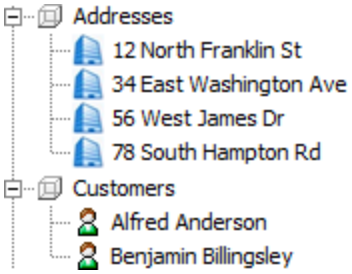
The image below shows the following relationships: Customer X has three Addresses, and Customer Y has one Address. Each of the Addresses (target) are referenced from Customer (source) using the 'Send to' reference type.



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.

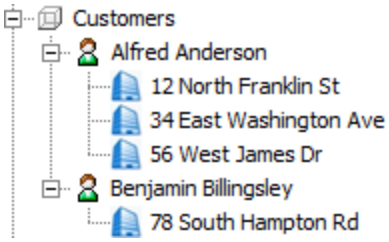
None

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:



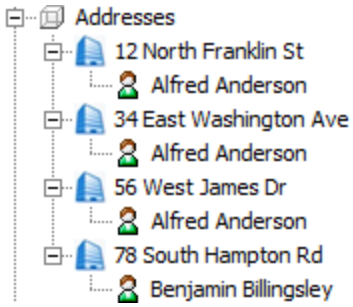
Source as parent, target as child

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

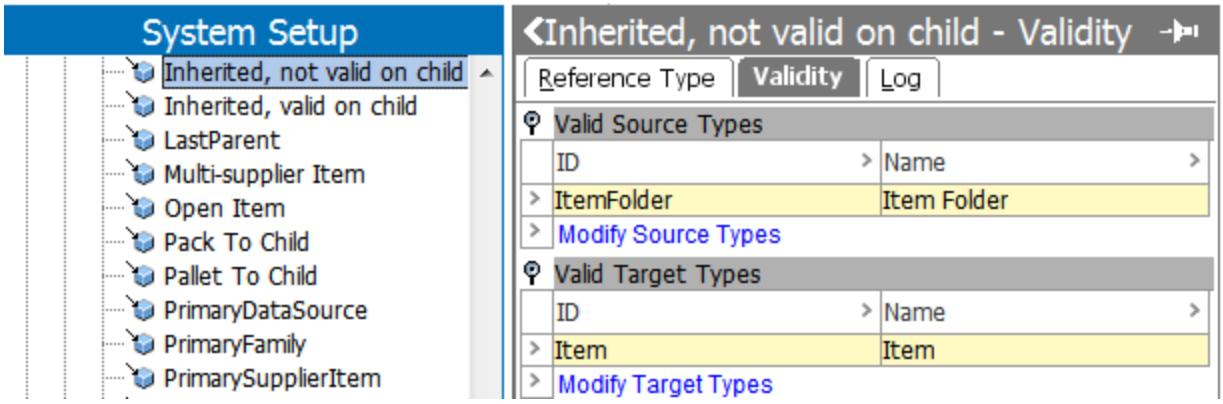
On the Tree, expanding the Customer entity hierarchy displays all referenced Address entities, as shown in the image below:



Inheritance Example for a Reference

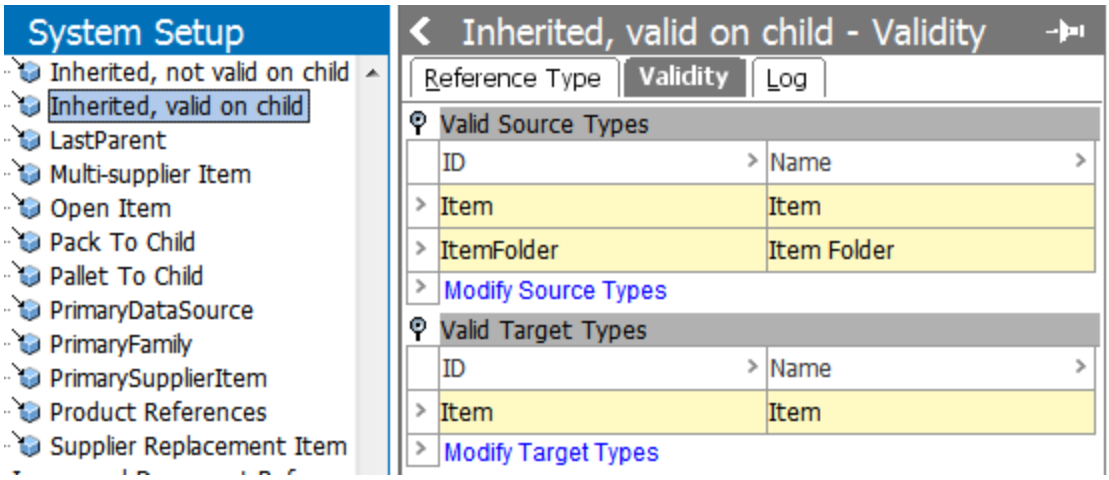
Before defining types of inheritance on references, it is important to keep in mind the valid sources and targets for a reference. If a reference is valid on an object, it can also be edited there. However, a reference that is not valid on an object can still be visible on the object, due to the reference 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:



The child object type is a valid target of the reference, but not a valid source object.

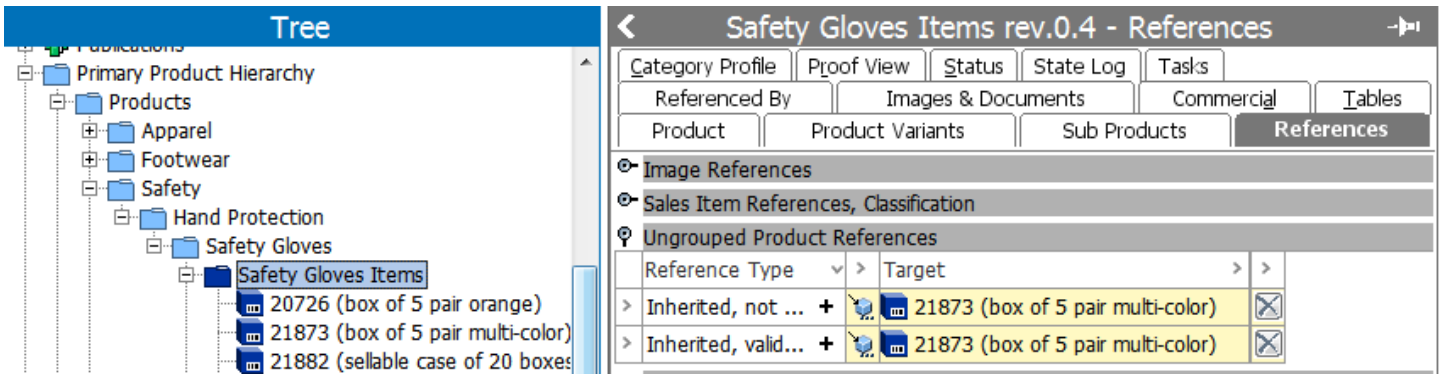
In contrast, a second reference is created called 'Inherited, valid on child' that has the child object valid as both a source and a target of the reference:



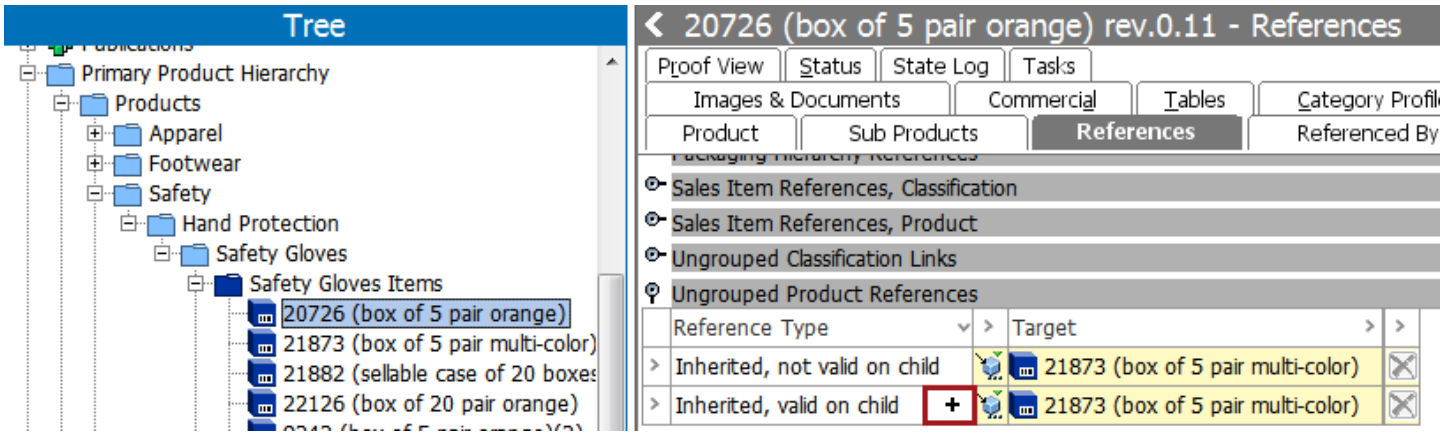
When both references are set with inheritance, any reference added to the parent (Item Folder object) will also be present on the child (Item object).

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:

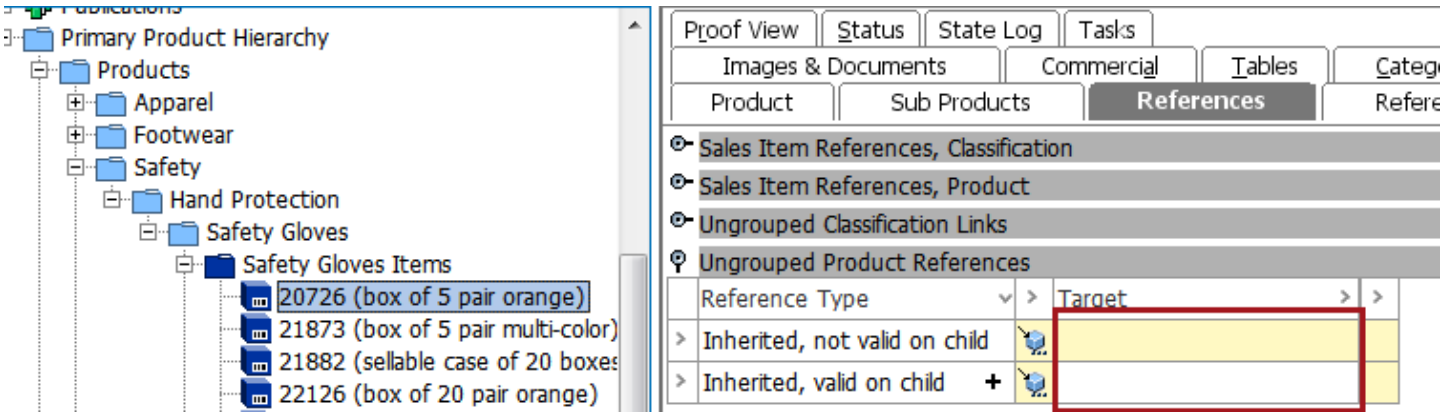


Child object:



Note that only the reference that is also valid on the child has a [+] available, indicating it is editable.

This is perhaps better illustrated when the references are removed:



Notice 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.

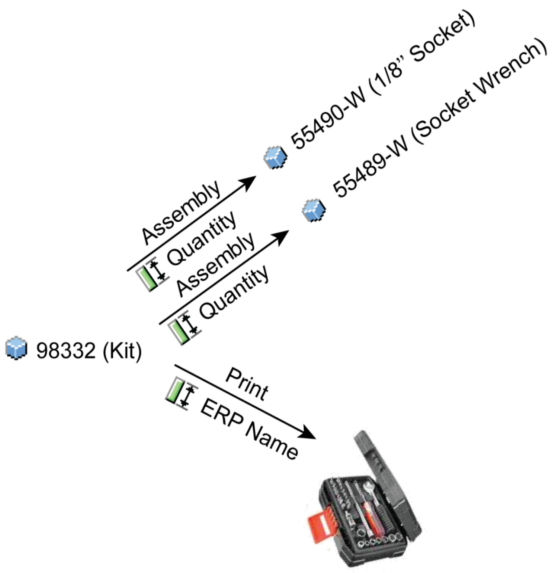
Metadata Attributes on Reference and Link Types

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.



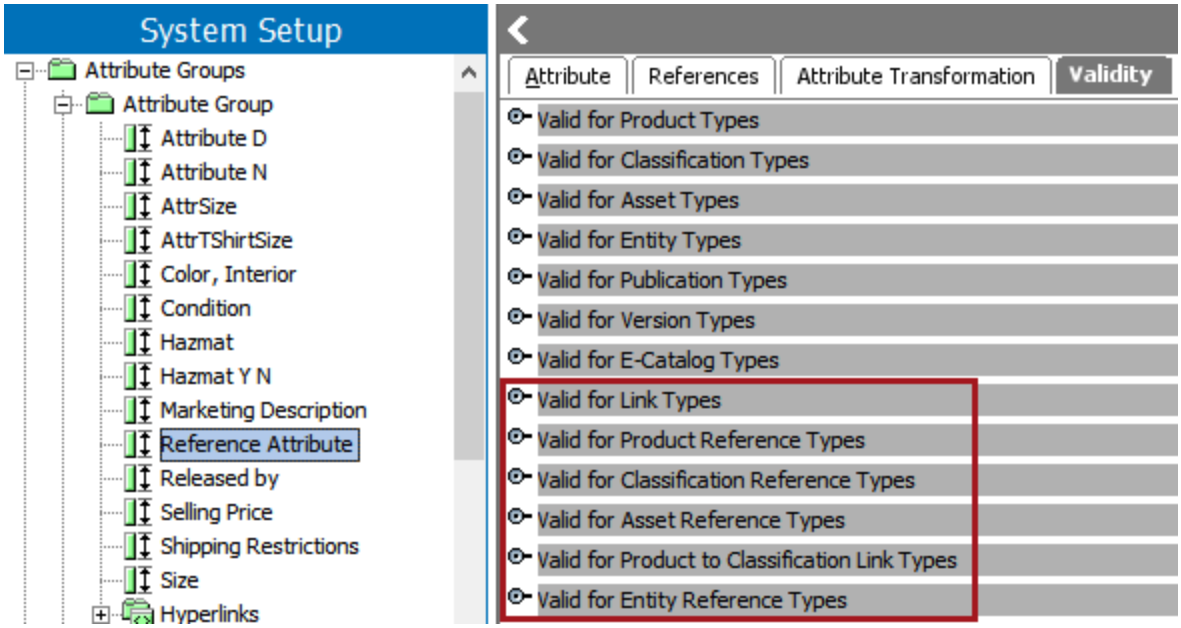
- A description attribute named Quantity has been applied the reference type Assembly. This attribute holds number of sockets to be included in the kit.
- A description attribute named 'ERP Name' has been applied the reference type Print. This attribute holds the name of the reference used in an ERP System.

For information on adding metadata to the attribute link types, see **Product Attribute Link Type** or **Classification Attribute Link Type**.

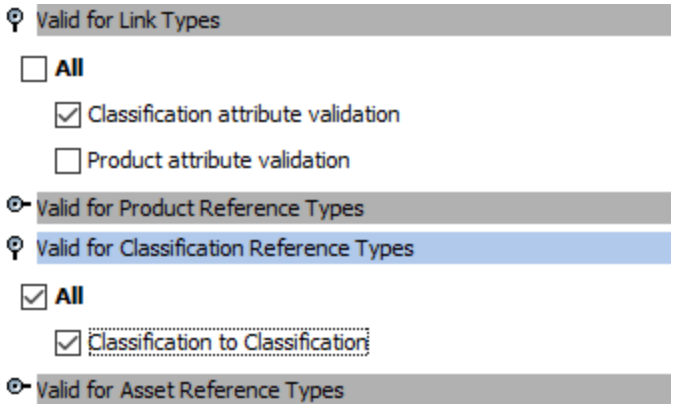
Add a Metadata Attribute

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.

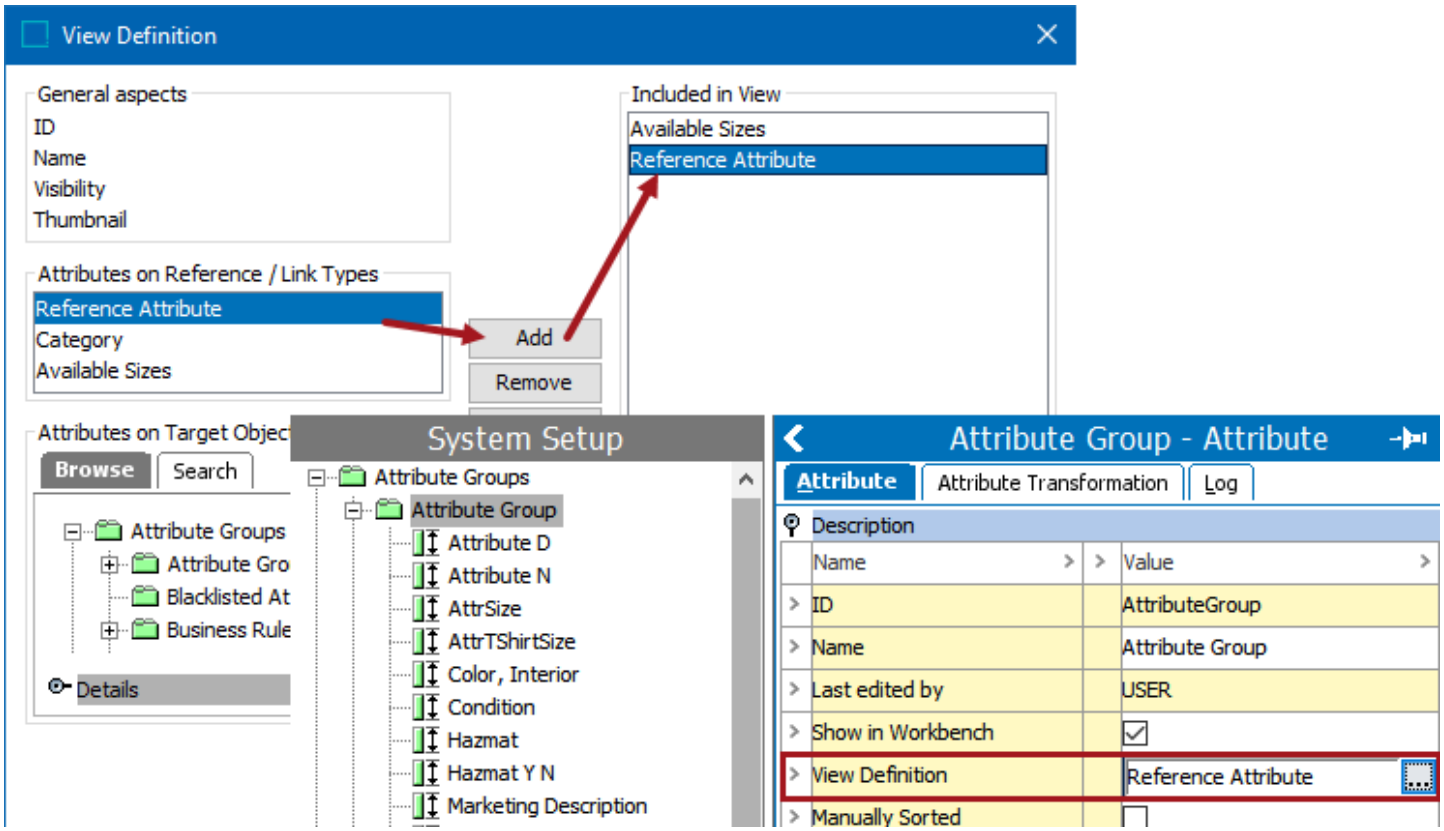


3. Select all reference types and link types that should display the metadata attribute. Checking an option also updates the Attributes flipper of the selected reference or link type.

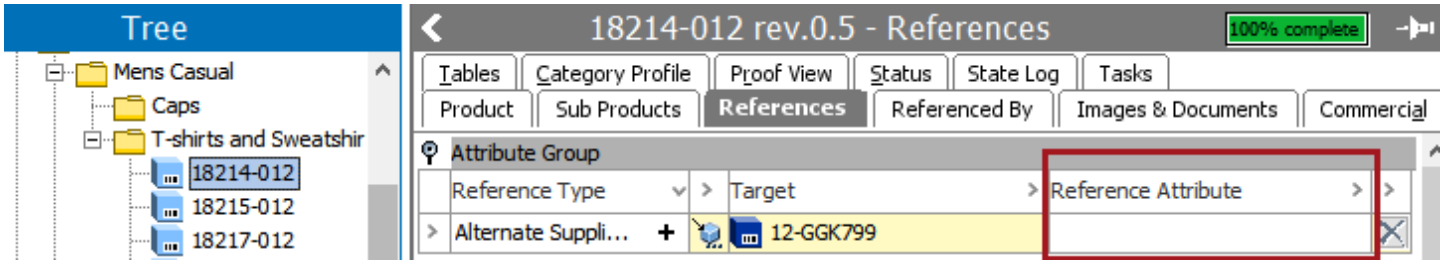


For more information, see **Validity on Description Attributes** in the **System Setup / Super User Guide**.

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.



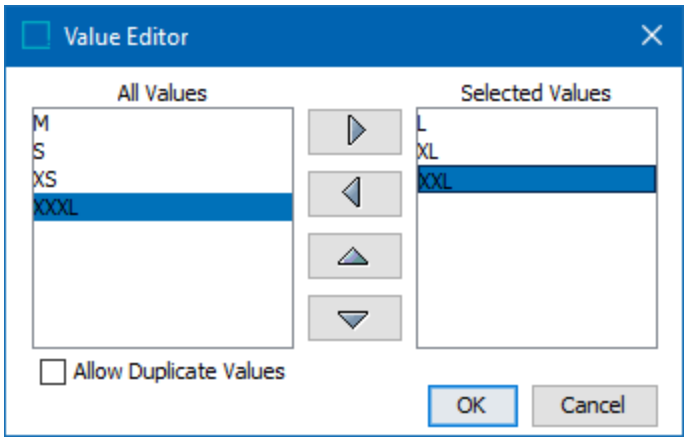
Edit a Metadata Attribute

Although the images below show a reference type, the same setup is used for editing metadata on a Product to Classification link type.

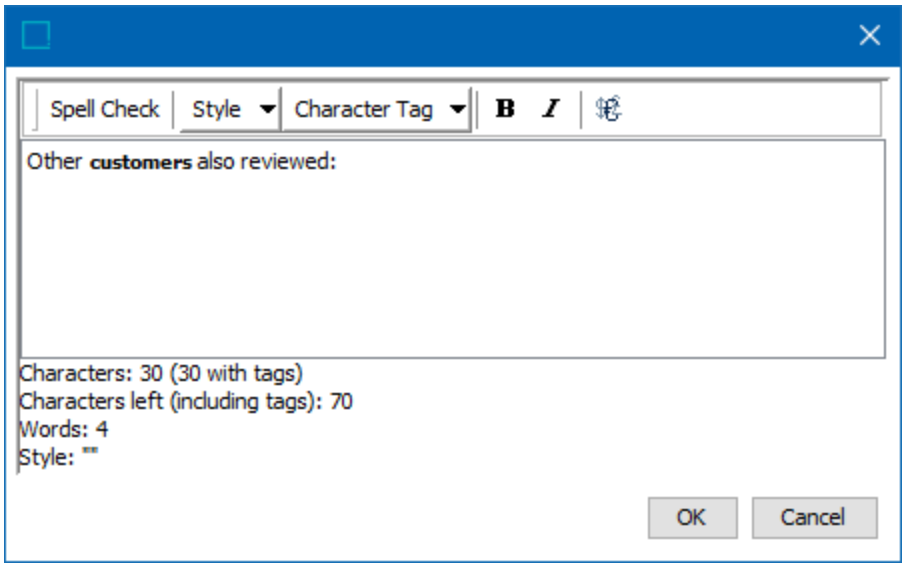
On the **References** tab, double-click an editable field (any white field as shown above) and edit using the available functionality based on the type of the metadata attribute.

- 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.

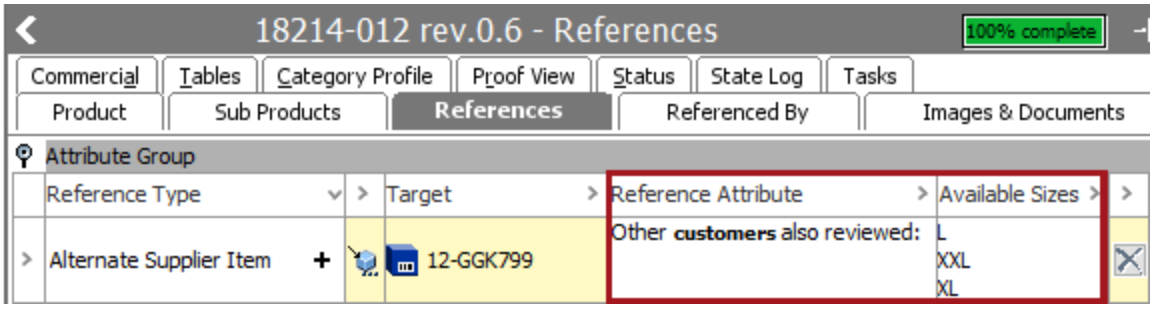
  Choose a value and use the necessary arrow button to include it in the Selected Values list, or to rearrange the order of the values.



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

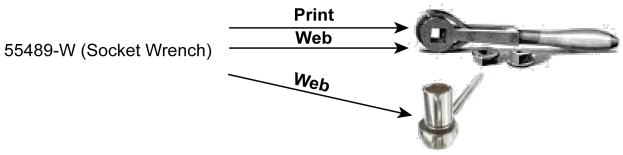
It is possible to have multiple references of the same type between the same two objects, by creating a separate reference type for each need. Typically, such references may also have different restrictions applied. In this way, the same product could be linked to the same image by multiple product-to-asset 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.

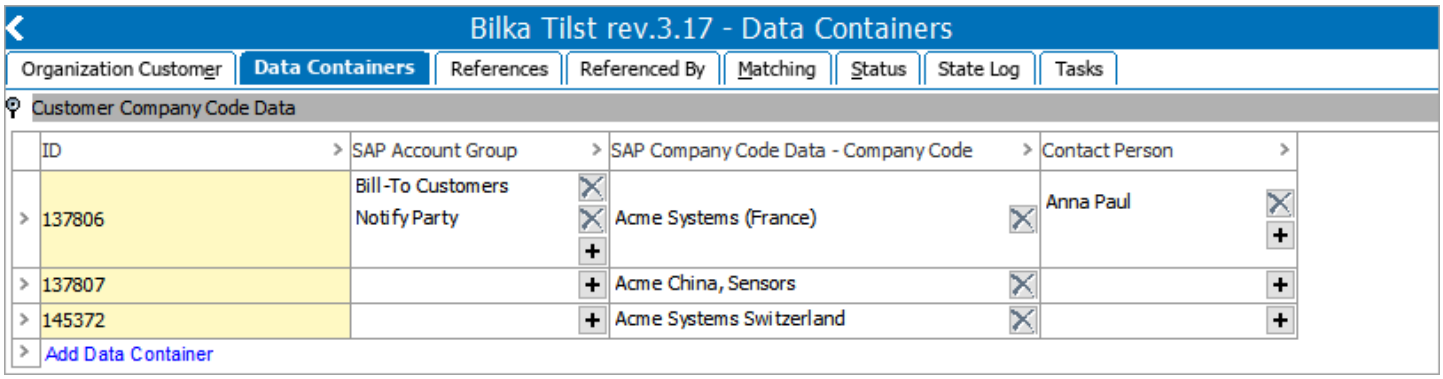


For more information, see the **Allow multiple references** parameter section of **Maintaining a Reference Type** documentation.

References on Data Containers

References to other objects can be stored on both single and multi-valued data containers. The supported reference types include: Entity Reference Type, Product Reference Type, Classification Reference Type, and Image and Document Reference Type.

Whenever a data container type is selected as the source object, any references of the relevant type are stored and displayed on the Data Containers tab of the object that holds the data container rather than that object's References tab.



For more information on data containers, see the **Data Containers** topic in the **System Setup** documentation.

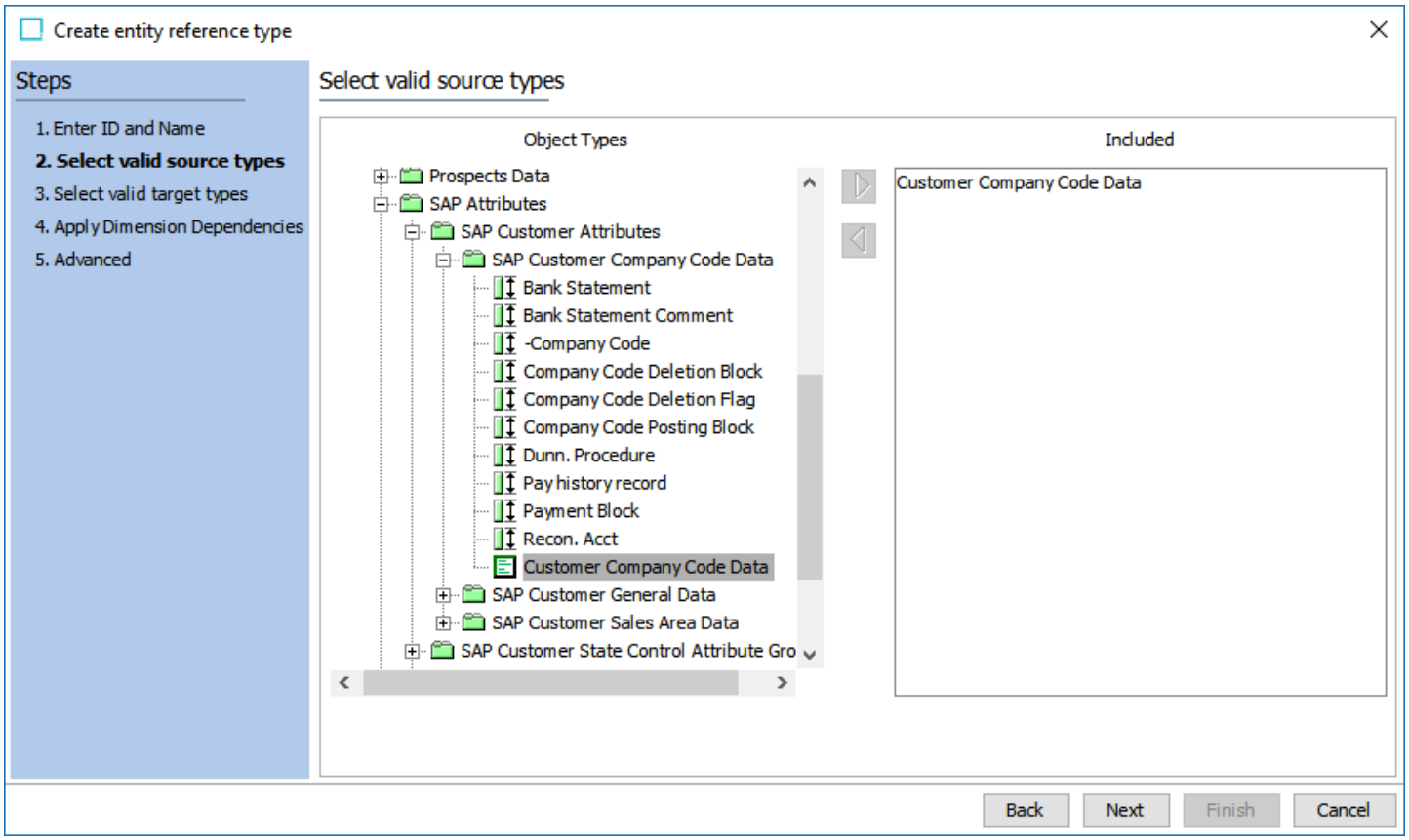
References on data containers can also be displayed in the Web UI via the Globally Configured Data Container component. For more information, see the **Data Containers in Web UI** section of the **Web User Interfaces / Web UI Setup and User Guide** documentation.

Configuring a Reference Type with a Data Container Type as Source Object

The process for creating a reference type for use in data containers differs slightly from the normal reference type configuration steps. For more general information on creating reference types, see the **Creating a Reference Type** section of this documentation.

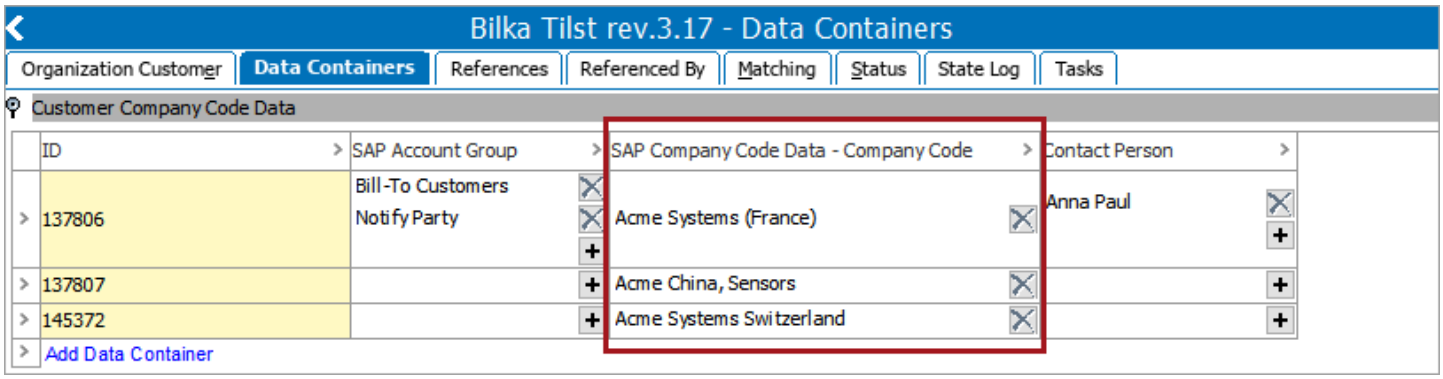
Caveats specific to this kind of reference type include:

- On the second step of the **Create entity reference type** wizard, select a data container type(s) as the object type.



- No configuration is available on the fourth step of the wizard, as this kind of reference type cannot be dimension dependent.
- On the fifth step of the wizard, inheritance settings are not available.

Once the reference type has been created, a column for the reference type will appear on all valid data containers in the workbench.

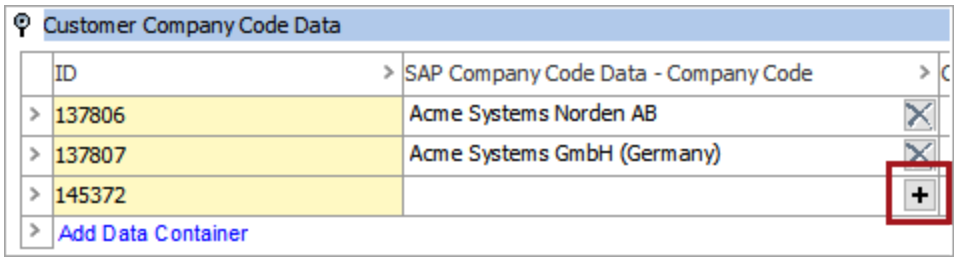


Important: Reference types that use a data container as a source cannot have metadata attributes on them. If a reference type already has metadata allowed on it, it cannot also have a data container configured as a source.

Editing References on Data Containers

References on data containers are maintained within the Data Containers workbench tab.

1. To add a reference, navigate to the cell of the relevant reference type and click the + button.



2. In the window that appears, browse or search for the desired object and click **Select**.

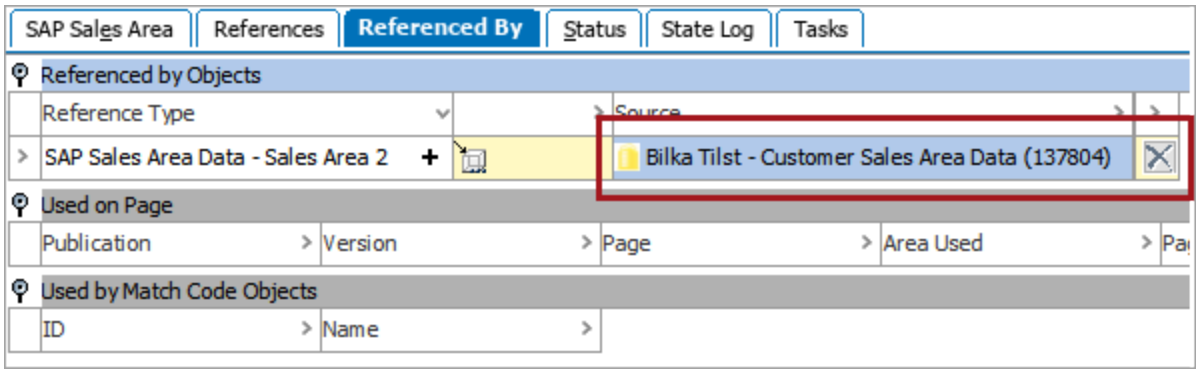
To delete a reference, click the button and confirm the deletion.

For more information on editing data containers, see the **Adding and Maintaining Data Container Instances** section of the **System Setup** documentation.

Viewing Data Container References via Referenced By Tab

These references may be viewed on the Referenced By tab of the target object, which includes the name of the data container and the object it belongs to.

Note: Using data containers to reference other objects is the only way a single object can reference another object multiple times.



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.

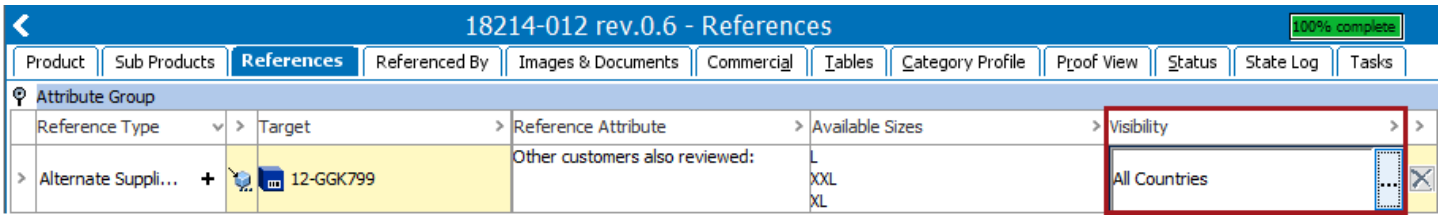
- Making the visibility global means the reference settings are inherited from the global context. For example, when all contexts require the same setting.
- Making the visibility local means the reference does not inherit its settings from the global context. For example, if you want to define attributes 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.

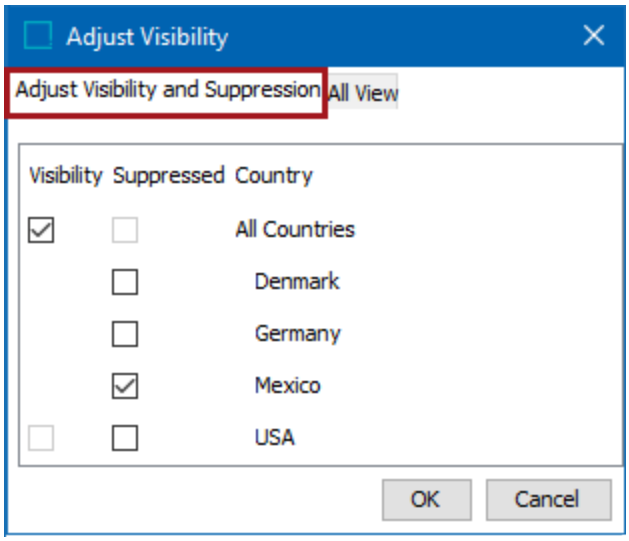
Note: The Visibility column is displayed on a reference when the General Aspect option 'Visibility' is added to the 'Included in View' panel on the View Definition of the attribute group. For more information, see the **Add a Metadata Attribute** section of the **Metadata Attributes on Reference and Link Types** documentation.

Setting Reference and Link Type Visibility

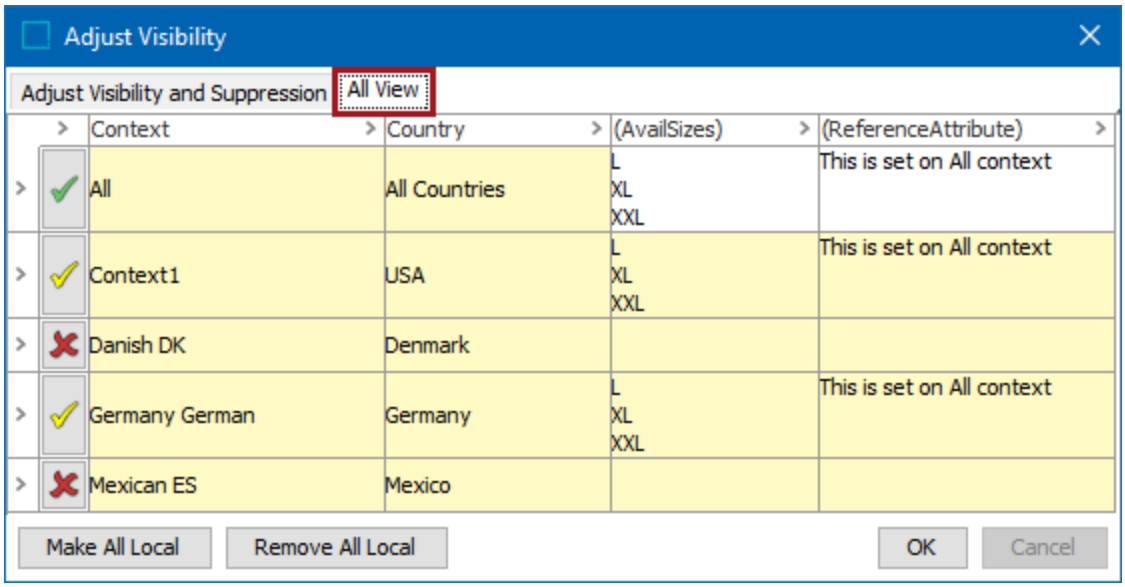
1. In the Tree, select the relevant node, click the **References** tab, and open the flipper that contains the dimension-dependent reference to be edited.



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.



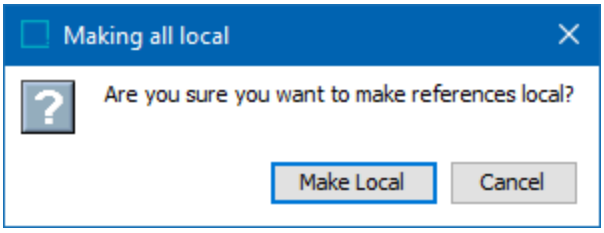
3. In the Suppressed column, check any contexts that should not display the reference.
4. Click the **All View** tab and note the status of the visibility of each context:
   - ✓ indicates that the metadata attribute is local and not being inherited (All). The setting is inherited by the contexts that 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).



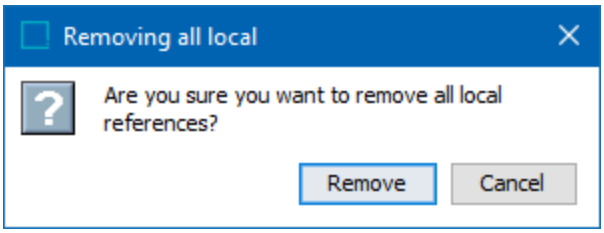
5. Modify the visibility of each required context as needed:
   - Click ✓ or ✗ to make the visibility of a reference local for only the selected context. Assuming the user has privileges to create the reference, the indicator changes to ✓, the reference value field changes to a white

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.

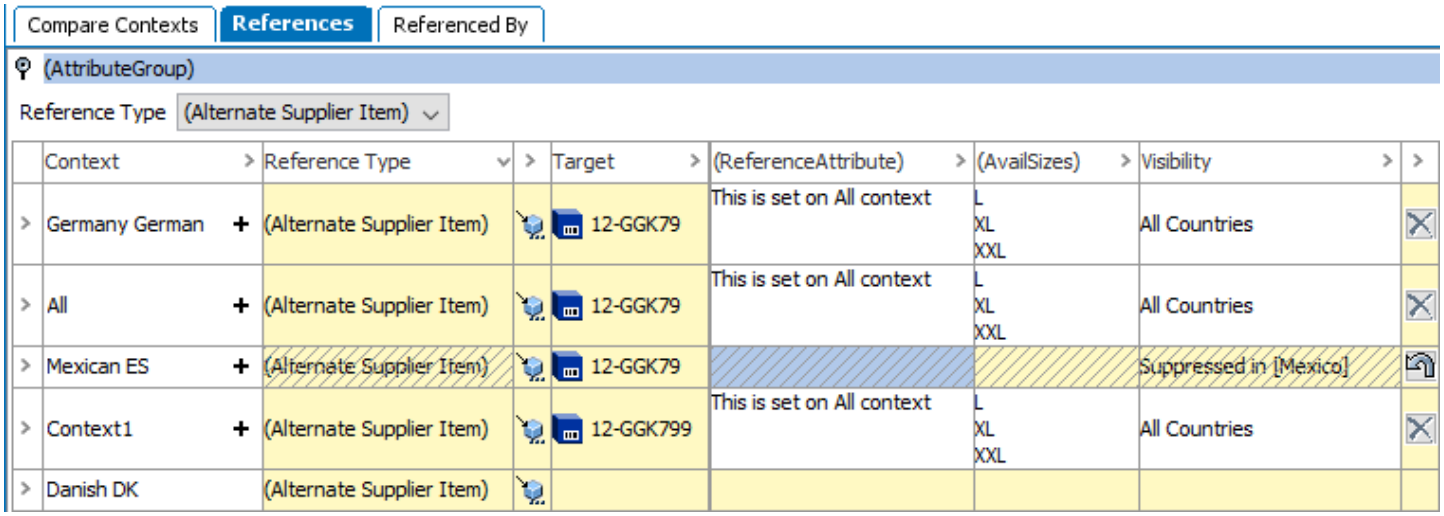


- Click Remove All Local to restore inheritance for all contexts currently local. In the 'Removing all local' dialog, confirm that you want to restore inheritance by clicking the **Remove** button. If a global context is local, all check marks turn yellow except the icon for the contexts that were previously 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.

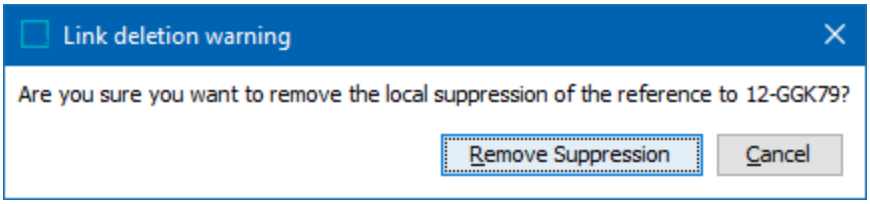


6. Verify the results of the visibility by checking the References page on the object and make any additional changes. Use the Context Mode by clicking in the toolbar.

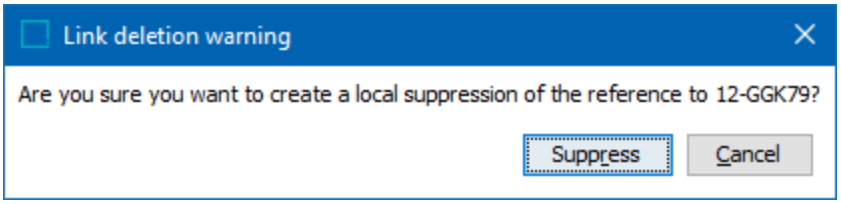
Note: Navigate to View > Target to select the Contexts for the comparison view to display.



- + 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.



Revisions

In STEP, historical versions of objects are stored as revisions, providing a historical 'snapshot' of an object. All versions of a revisable object are automatically preserved. Each time a change is made by a different user, a numbered revision is created and logged. Revisions allow 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 the system.

Revisions are maintained for the following object types: business rules, workflows, integration endpoints, products, entities, classifications, assets, standard standalone event queues, and Web UI configurations. Although each of these object types have their own unique features, the overall functionality of creating and maintaining revisions is universal across objects.

Revisions and Approvals

While *revisions* and *approvals* are related, they should not be confused.

- Objects that are *workspace revisable* are subject to approvals, and approving an object automatically generates a revision of it. For more information on approving objects, see the **Approval of Objects** section of the **Getting Started / User Guide** documentation.
- 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 in the **Generating Revisions** topic.
- 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, see the **Revisability on Entity Object Type** topic within this guide.

Important: It is critical to consider revision handling for each system. Implement appropriate processes to purge old revisions and ensure that the number of revisions does not exceed the expected capacity of the system, which can cause performance issues.

The following topics explain how to work with revisions:

- **Comparing Revisions**
- **Generating Revisions**
- **Maintaining Revisions**
- **Reverting to Past Revisions**
- **Timing for Automatic Revisions**

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.

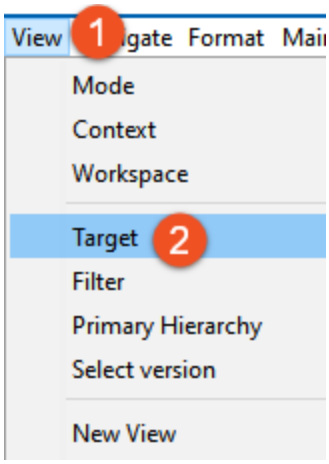
Revision history can also be displayed in Web UI. To learn more, see the **Multi Revision Screen** topic in the **Web User Interfaces** documentation.

To enable revision comparing in workbench:

1. With the object selected, click the Revision mode button from the workbench menu bar. The two most recent revisions for that object are displayed.

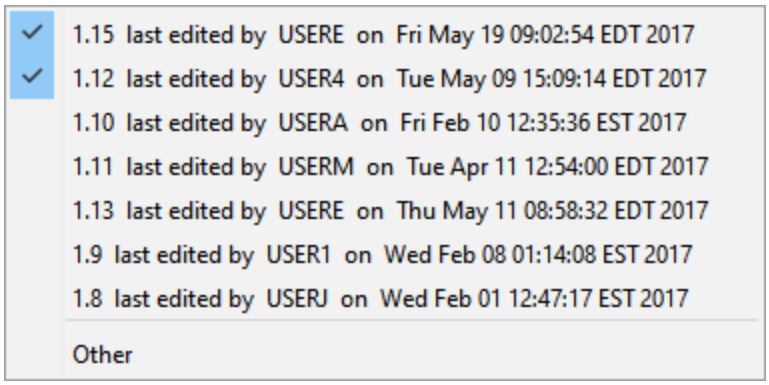


2. To view additional revisions, select 'View' from the toolbar, and then select 'Target'.

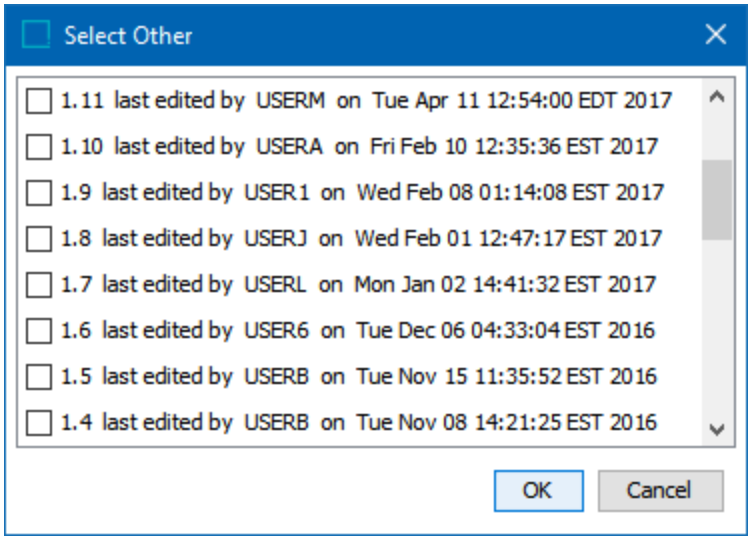


Note: Target cannot be selected unless the object is being viewed in an appropriate mode, like Revision mode.

3. In the dropdown, select a revision from the eight most recent revisions listed or the 'Other' option. A check marks indicates a revision being displayed.
   - To add a single additional revision, click any unchecked revision.
   - To remove a revision, uncheck the revision.

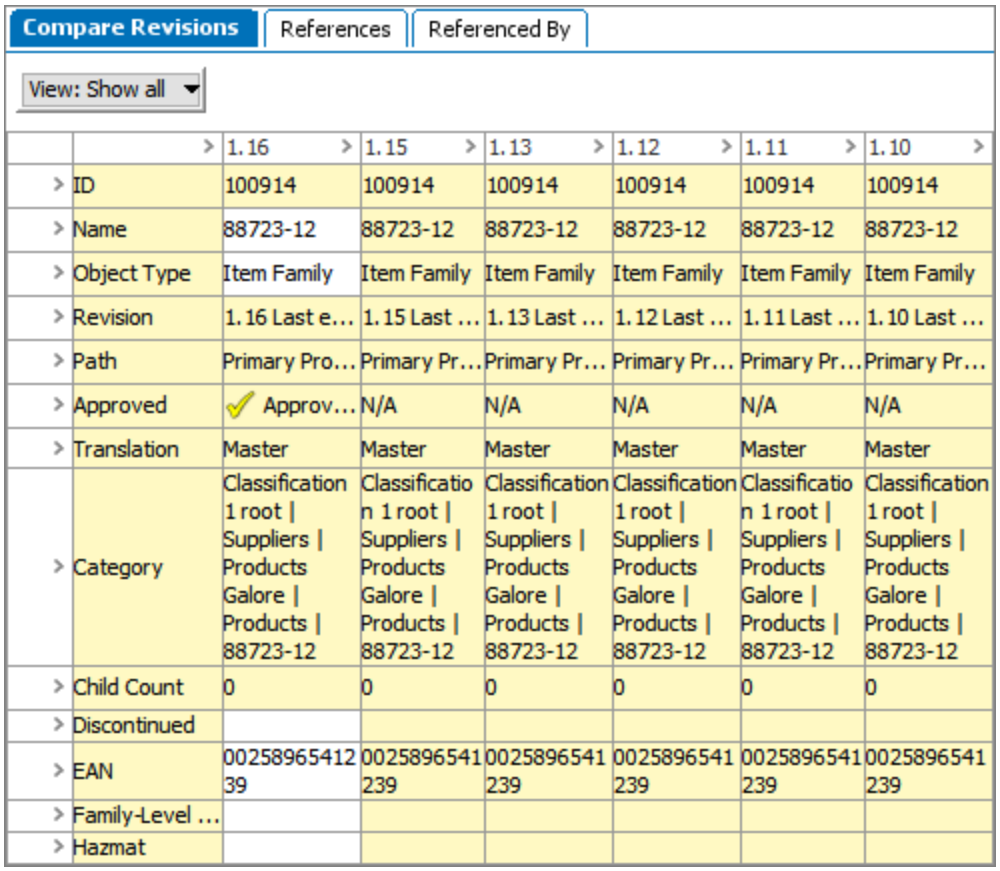


- To add or remove multiple revisions, click the 'Other' option to display a list of all revisions. Select all required revisions and click the **OK** button to display them in the comparison view.



Note: When more revisions are added than can be displayed on the screen, a scroll bar displays allowing access to all revisions.

4. Review the displayed revisions on the Compare Revisions tab.



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	Classification 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	Classification 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						

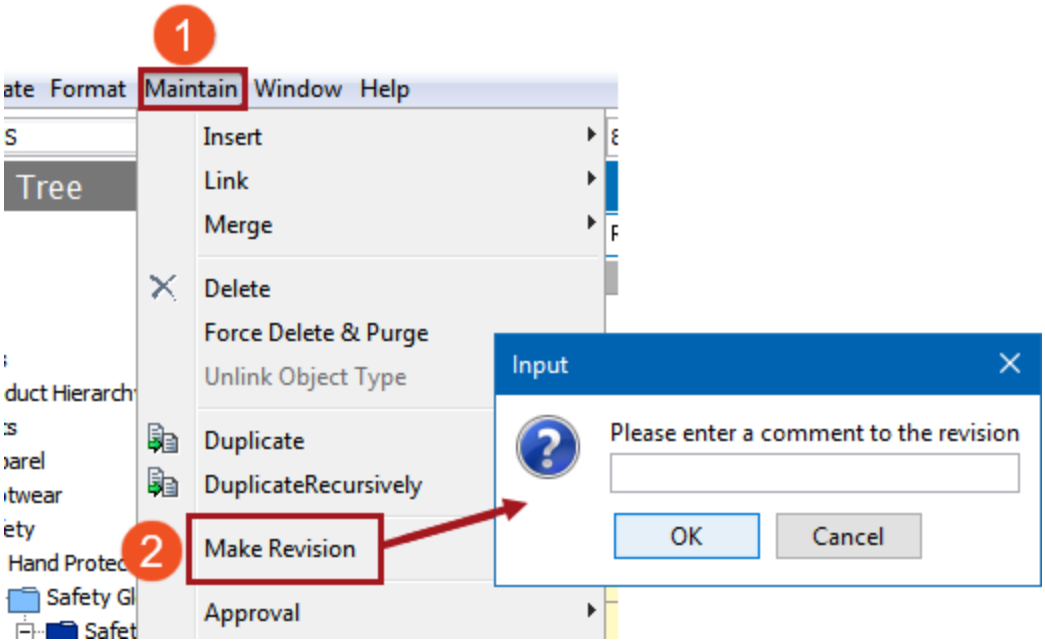
Generating Revisions

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. 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.

For more information, see the **Important considerations for revisions section** below.

A revision can be created in the following ways:

- Automatically upon creation of a new workspace revisable object.
- Automatically upon approval of a workspace revisable object.
- Automatically each time a change is made by a different user, or when any change is made following an approval. For assets, when the digital media file associated with an asset in STEP is changed, this generates a revision on the asset object.
- Automatically when 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. For more information, see the **Timing for Automatic Revisions** topic.
- Manually, from Maintain menu > Make Revision. Enter a comment and click **OK**.



Note: Revisions can be created when the data is perfect, in preparation for future unexpected changes. Creating a manual revision allows you to revert back to perfect data. Manual creation of revisions must be done for one node at a time; it is not possible to create revisions manually in bulk.

Important considerations for revisions

- Externally maintained data and calculated data are exceptions to the revision generation options mentioned. 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. Among other examples, the same applies for workflows that included business rules that are subsequently deleted, objects that referenced other objects or assets that were subsequently deleted, or reference and link types when deleted.

Maintaining Revisions

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.

Any revision can be permanently removed from STEP using a purge option. Once a revision is purged it cannot be recovered. The following methods are available for purging revisions and are defined below:

- Automatic purging of revisions
- Manually purging multiple revisions
- Manually purging a single revision

A revision cannot be purged if it is the latest revision or if it is being used by STEP, for example, in an event processor or OIEP.

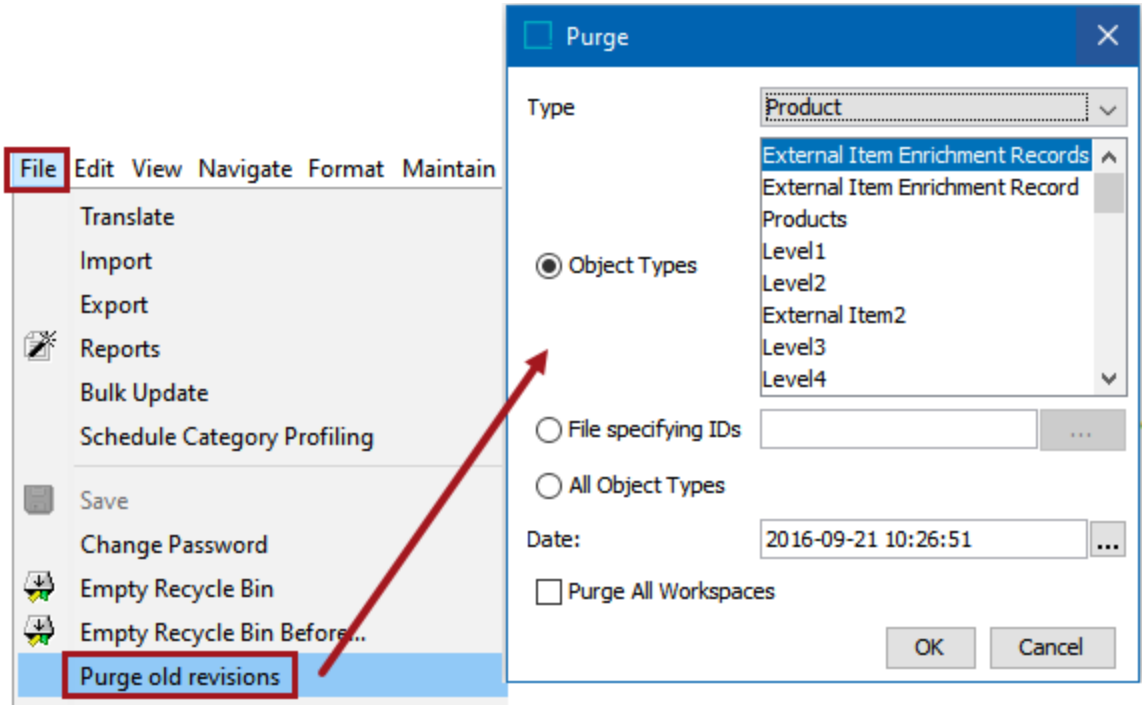
Automatic Purging of Revisions

The Revision Management event processor can be used to aid in cleanup and it is recommended that this be set up on all systems. For more information on this event processor, see the **Revision Management Processing Plugin Parameters and Triggers** topic in the **System Setup / Super User Guide** documentation.

Manually Purging Multiple Revisions

This manual revision cleanup option is recommended for ad hoc cleanup only. Additional automatic revision cleanup processes should also be in place.

1. On the File menu, select **Purge old revisions** to display the Purge dialog.



2. For the **Type** parameter, select the single type of object for which revisions should be purged. Selecting any type filters the subsequent Object Types selection accordingly.

   The available selections are Product, Classification, Asset, DTP Documents, and System Configuration.

3. Select one of the following options to specify the revisions to be purged:

   - **Object Types:** Select this radio button and select one or more object types in the list to have revisions purged for all objects of the selected type(s). Multiple object types can be selected using Ctrl + Click. The available selections are filtered using the Type dropdown above.
   - **File specifying IDs:** Select this radio button to load a .txt file (with one ID per row) 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.

     Take care using this option as STEP IDs are only unique within each object super type (e.g., Products, Classifications, etc.). For example, if an asset and a product can have the same STEP ID, but if the 'Product' object type is 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' selected, or the wrong revisions could be accidentally purged.

   - **All Object Types**: Select this radio button to have revisions purged across all objects in STEP within the 'Type' chosen from the dropdown menu.

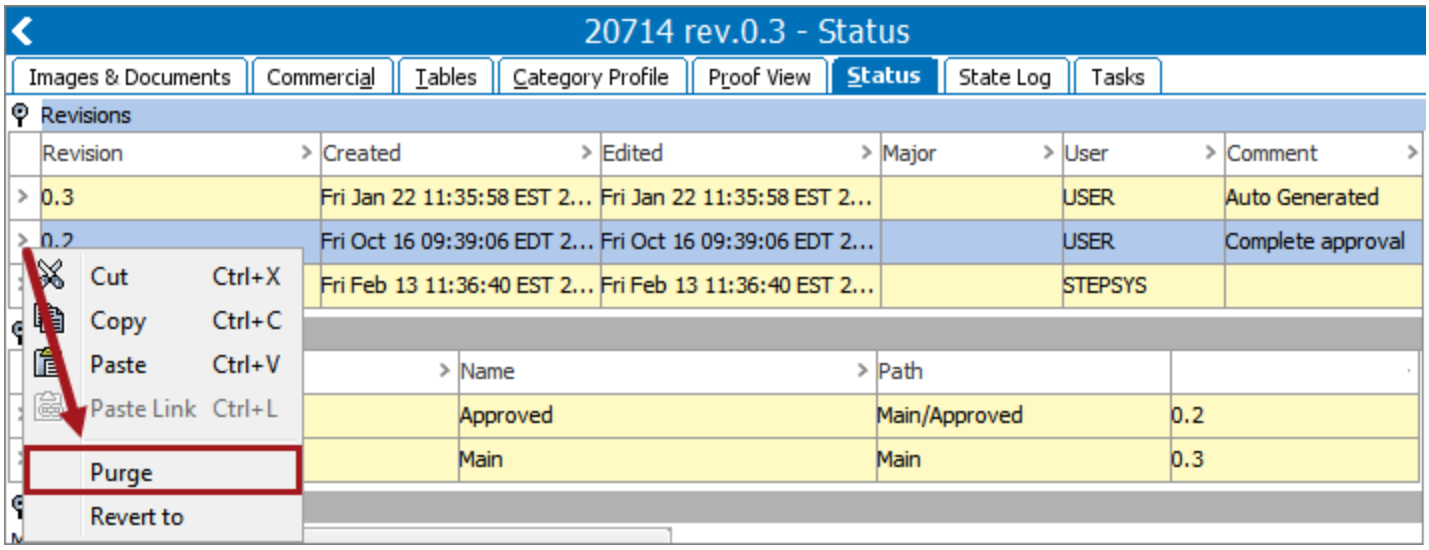
4. For the **Date** parameter, after selecting one of the three radio button options, specify up to what date / time the revisions should be purged. If no revision meets the date exactly, it will keep the most recent revision. For example, if the user selects today, but the last revision was yesterday. The system would keep yesterday's revision because it is most recent.

5. For the **Purge All Workspaces** parameter, if checked, revisions made in all workspaces are purged, for example in the Approved and Main workspaces. If unchecked, revisions are only deleted from the current workspace. For more on workspaces, see the **Workspaces** topic in the **System Setup / Super User Guide** documentation.
6. Click **OK** to perform the purge as configured, or click **Cancel** to close the dialog.

Manually Purging a Single Revision

This manual revision cleanup option allows a single revision to be purged. Additional automatic revision cleanup processes should also be in place.

1. Open the Status tab of the object.
2. Click the arrow on the revision to be removed.
3. From the menu, select **Purge**. The selected row is removed from the system.



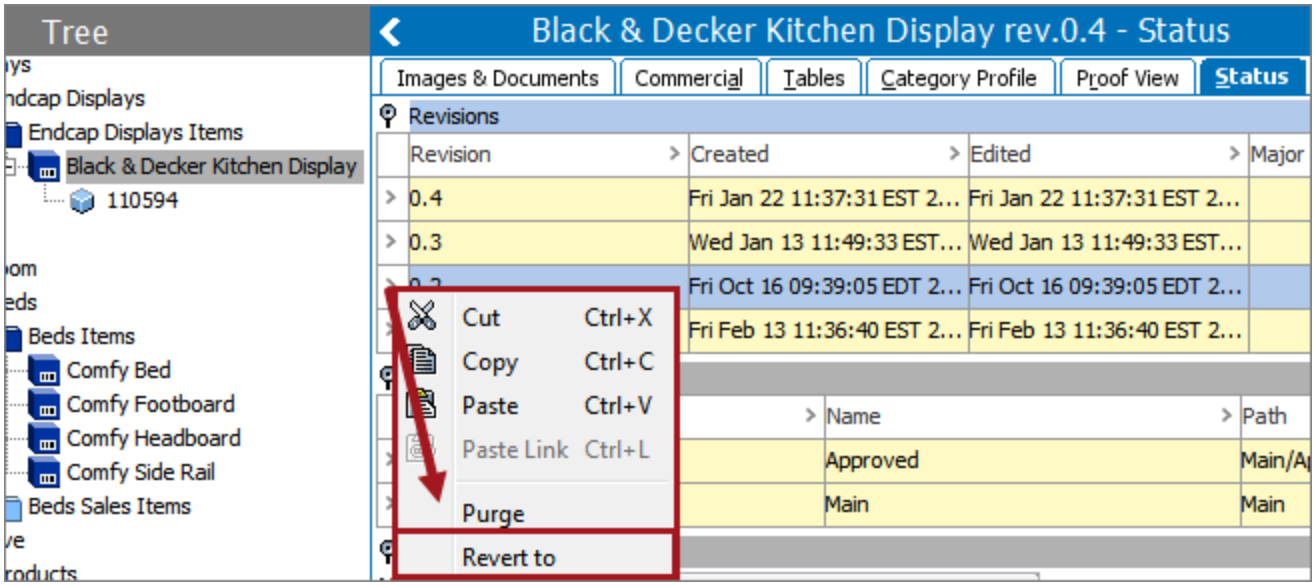
Reverting to Past Revisions

Reverting to a past revision generates a *new* revision that matches the one reverted to. It does not discard any intermediate revisions, nor is any audit trail lost.

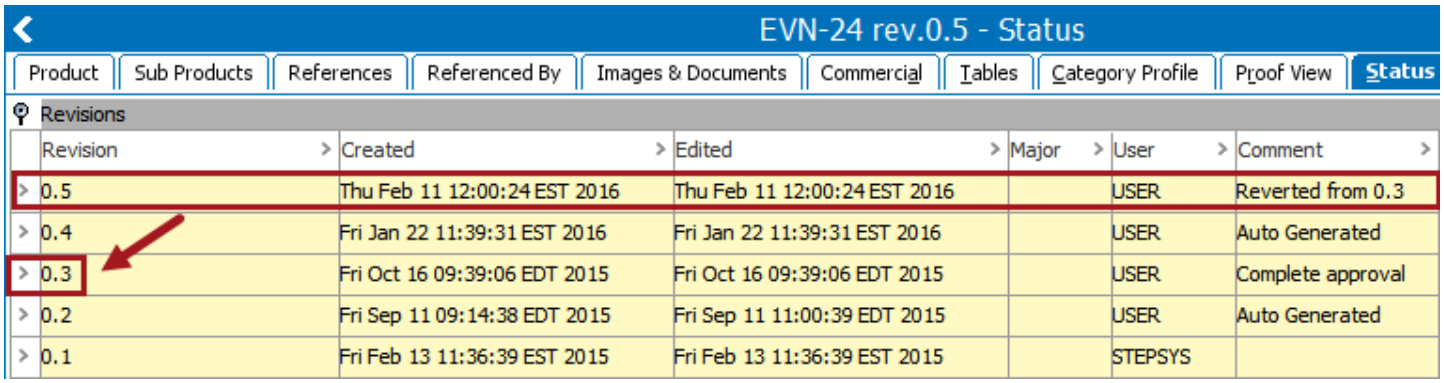
Important: To revert back to a revision successfully, all configurations applied to the object must still be in place. For example, when reverting to a previous version of an integration endpoint, all related attributes, references, and business rules must still exist, otherwise the revived endpoint may not work as intended.

To revert to a past revision:

1. In the object editor, open the Status tab.
2. Click the arrow next to the desired revision to display a menu.
3. Click the **Revert to** option.



To track the history of the object, an automatic comment includes the revision of the object that has been reverted.


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

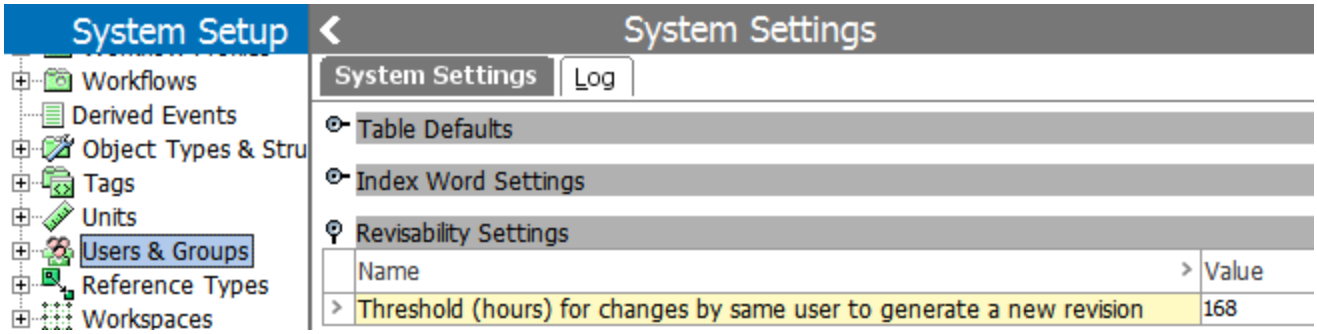
Timing for Automatic Revisions

By default, the system creates a revision on an object if a user makes a change to the object and the time from when the object was first touched after the last revision was made is greater than the timing threshold. This is particularly useful in cases where an object is primarily maintained by a single user and would not otherwise have a revision 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 for all revisable objects is 168, or one week, before a revision is created. If any user acts on the object during the time frame 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.

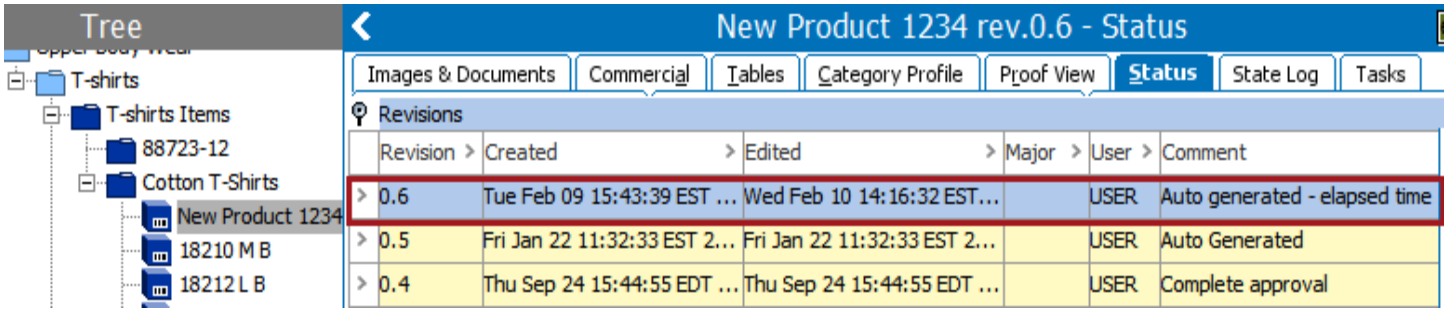
The number of hours can be changed in **System Setup** > **Users and Groups** > **Revisability Settings** flipper. Update the 'Threshold (hours) for changes by same user to generate a new revision' parameter.

- The minimum allowed hours is 24, and there is no maximum.
- If the parameter value is blank, the auto-revision functionality is disabled, and all revisions must be made manually or will be made due to different users modifying the revisable object.



Important: Set the parameter *at or above* the 168 (one week) default to prevent creation of an excessive number of revisions.

When an object gets a revision due to reaching the threshold parameter value, it is noted with 'elapsed time' on the object's Status tab > Comment column.

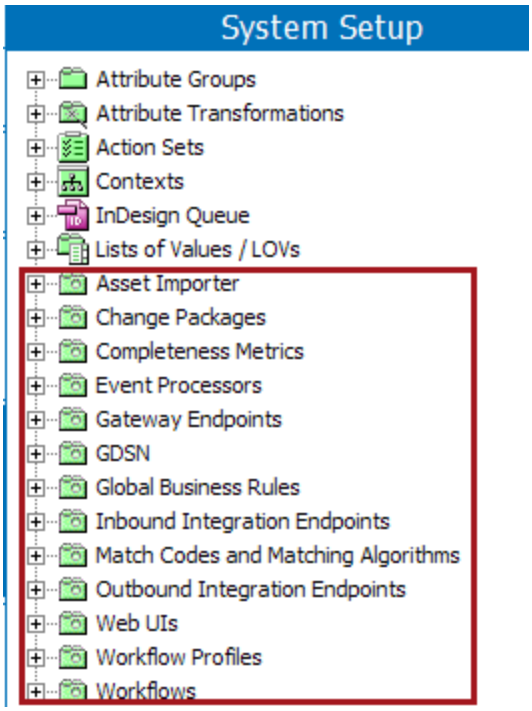


Revisions are created before the set number of hours expires using the manual method or other automatic methods, as described in the **Generating Revisions** topic.

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, there are predefined setup groups which can 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.

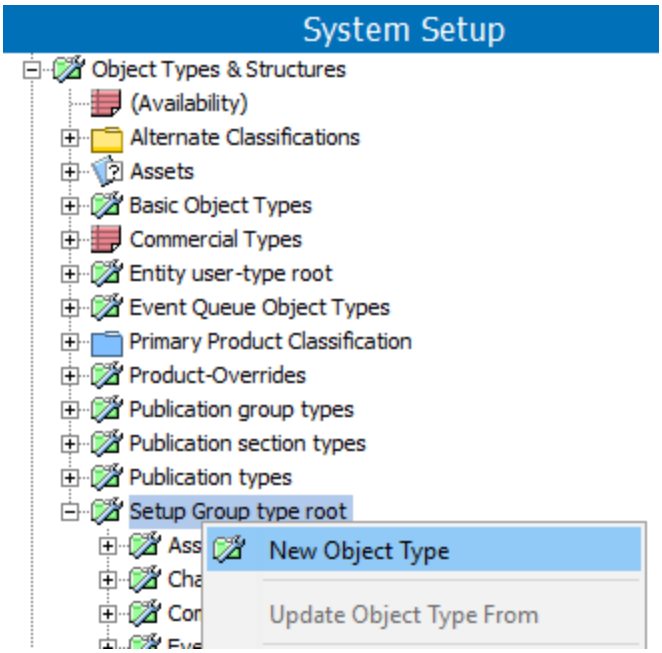
Setup Actions	Description
View Setup Groups	Users with this action applied will be able to view setup groups.
Maintain Setup Groups	Users with this action applied will be able to create and delete setup groups. They are also able to use the cut, copy and paste actions.

Creating Setup Group Object Types

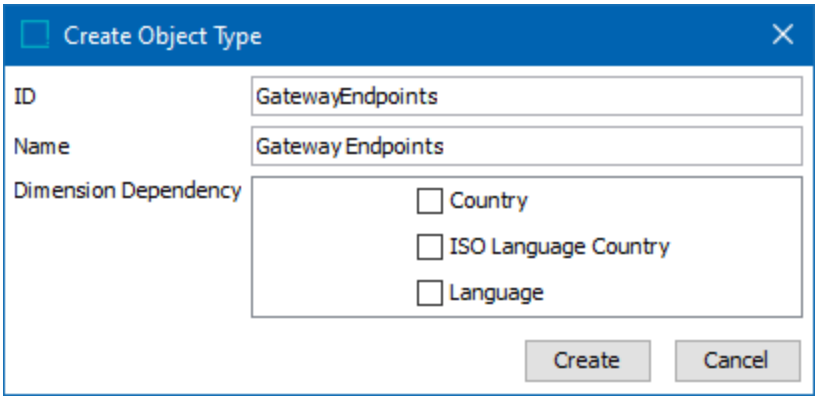
Before you can create any configuration in System Setup, you need to create one or more Setup Groups and define in which Setup Group a specific object type (e.g., integration endpoints, business rules, Web UI configurations, status flags, workflow configurations) is allowed to be created.

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. Add an **ID**, **Name** and click **Create**.

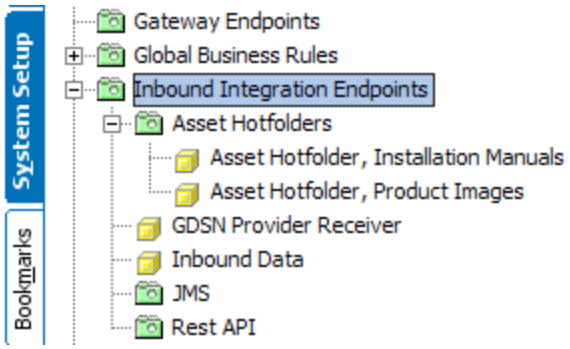


Note: The IDs can be auto generated for setup group creation. See the **Autogenerate Using Name Pattern and ID Pattern** topic in the **System Setup** / **Super User Guide** documentation.

The **Setup Group** displays as a child in the **Setup Group type root**.

3. Repeat these steps 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, you can maintain the respective integrations under respective setup groups.

Next, see the **Creating Setup Groups in System Setup** topic.

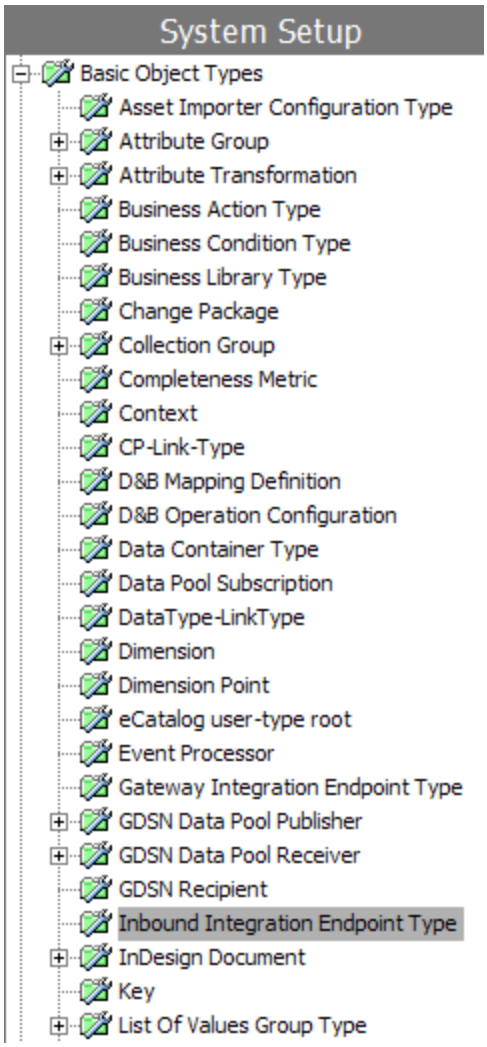
Defining Type of Objects to be Stored in Setup Groups

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 can be selected to make setup groups from are listed below:

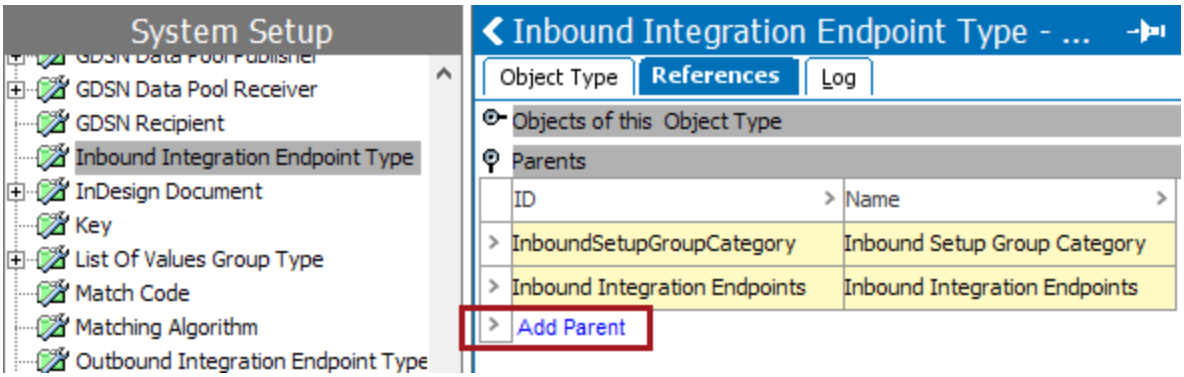
- Business Action Type
- Business Condition Type
- Business Library Type
- Inbound Integration Endpoint Type
- Outbound Integration Endpoint Type
- Web UI Configuration Type
- STEP Workflow Type

The example below illustrates how to 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 created as setup groups to hold other child objects.

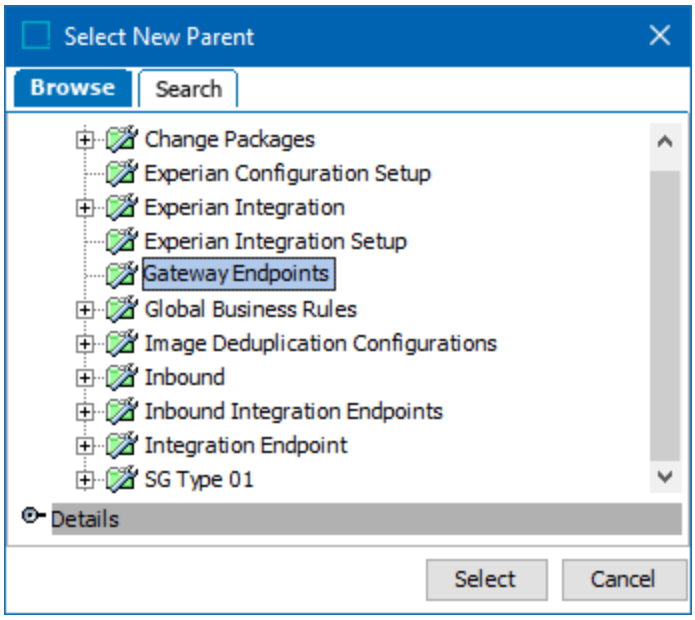
1. In **Object Types & Structures** > **Basic Object Types** > click **Inbound Integration Endpoints**.



2. In the **Inbound Integration Endpoint Type** editor, click the **References** tab and **Add Parent**.



3. In the 'Select New Parent' dialog 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

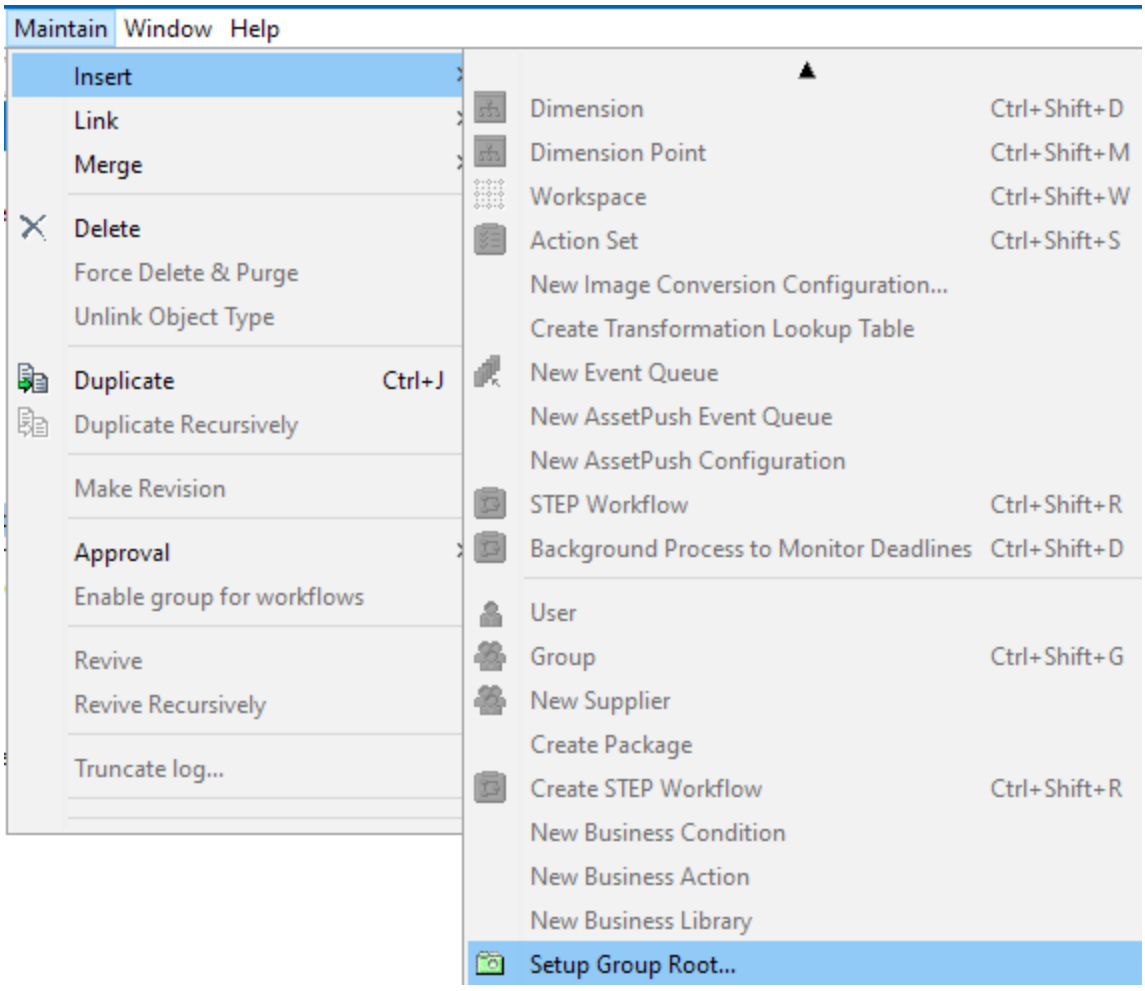
Creating Setup Groups in System Setup

After creating setup group object types, as defined in the **Creating Setup Group Object Types** topic, create the setup groups as defined below.

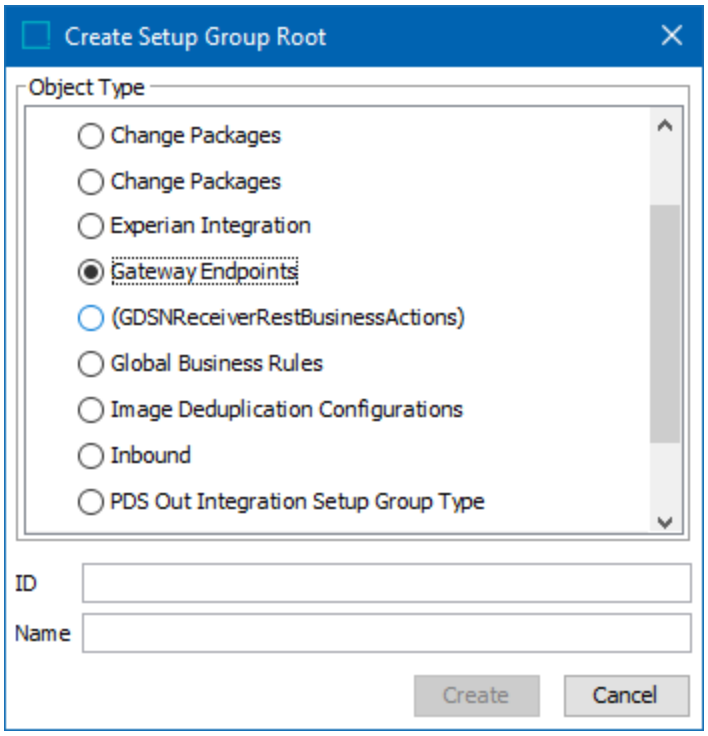
Setup Groups can be used to hold integration endpoints, business rules and/or Web UI configurations.

Note: You must have already created a Setup Group Object Type and have 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**.

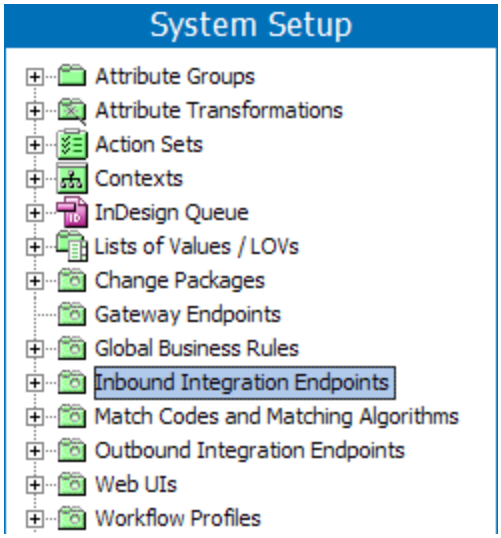


A **Create Setup Group Root** dialog appears.



3. Add an **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.

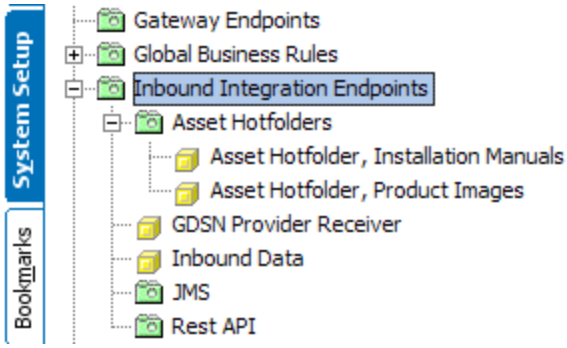
Setting Up Privileges Example

In this example, two user groups will be created.

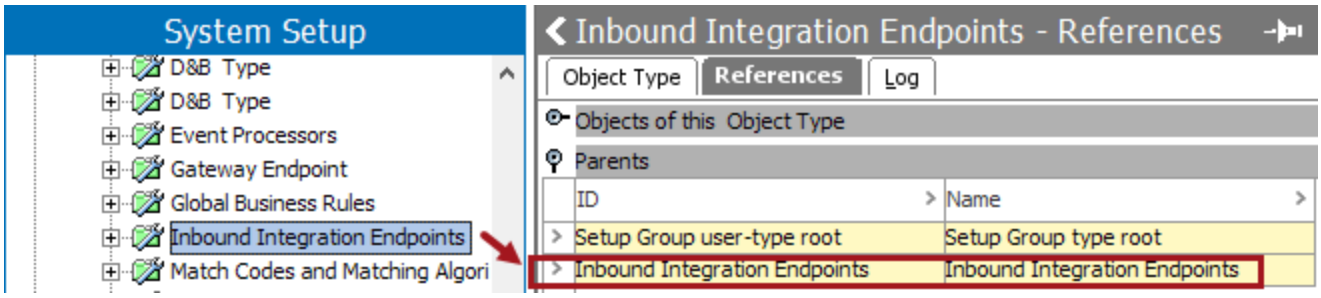
- One user group created will be named **View Setup Group and Integration Endpoints**. Users linked into this group will be granted the privilege to view Setup Groups and Integration Endpoints.
- The other user group created will be named **Maintain Setup Group and Integration Endpoints**. It will inherit 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, see **Action Sets** in the **Super User / System Setup**.

A structure of the setup groups could be as illustrated below.



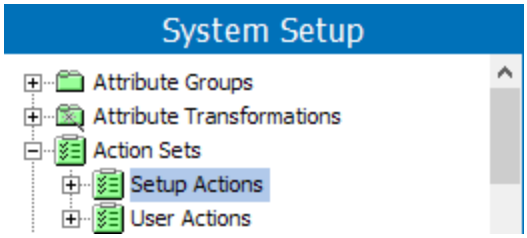
To be able to create additional setup groups that are able to hold Integration Endpoints below the **Integration Endpoint** setup group folder, the **Setup Group type root** 'Integration Endpoints' must have itself added as a parent. This same concept applies to all setup groups. See the screenshot below.



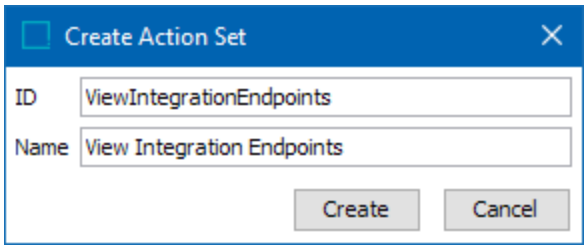
For more on how to maintain object types, see the **Maintaining Object Types** documentation in the **System Setup / Super User** documentation.

Privilege Setup Steps

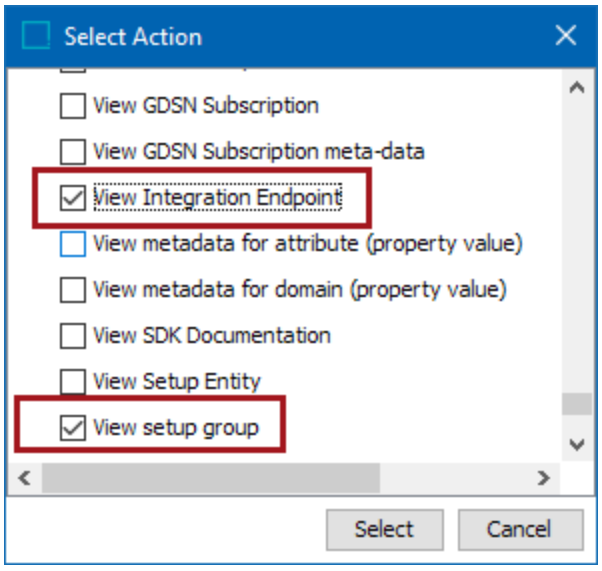
1. Go to System Setup > Action Sets > **Setup Actions**.

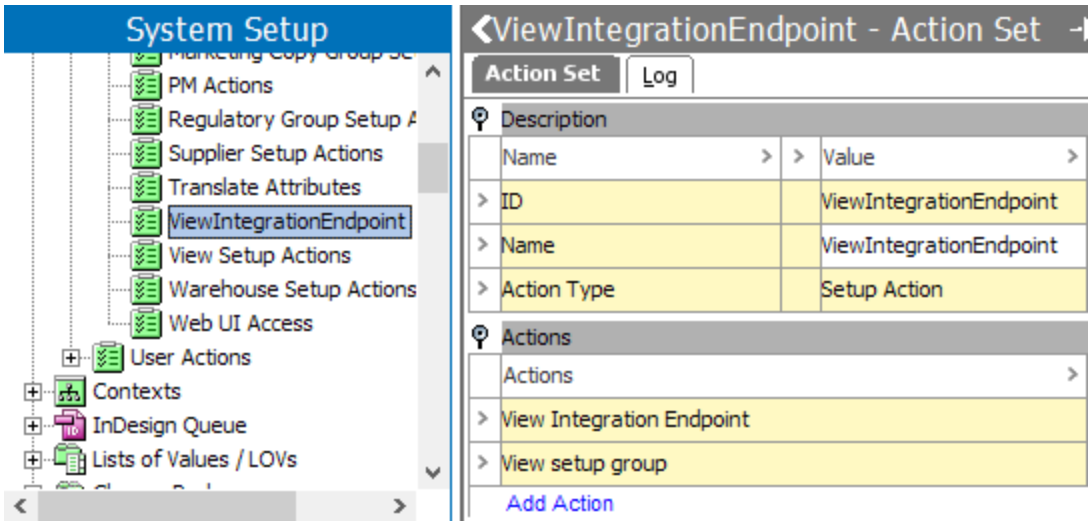


2. Right-Click and select **New Action Set**. Add an ID, Name, and click **Create**.

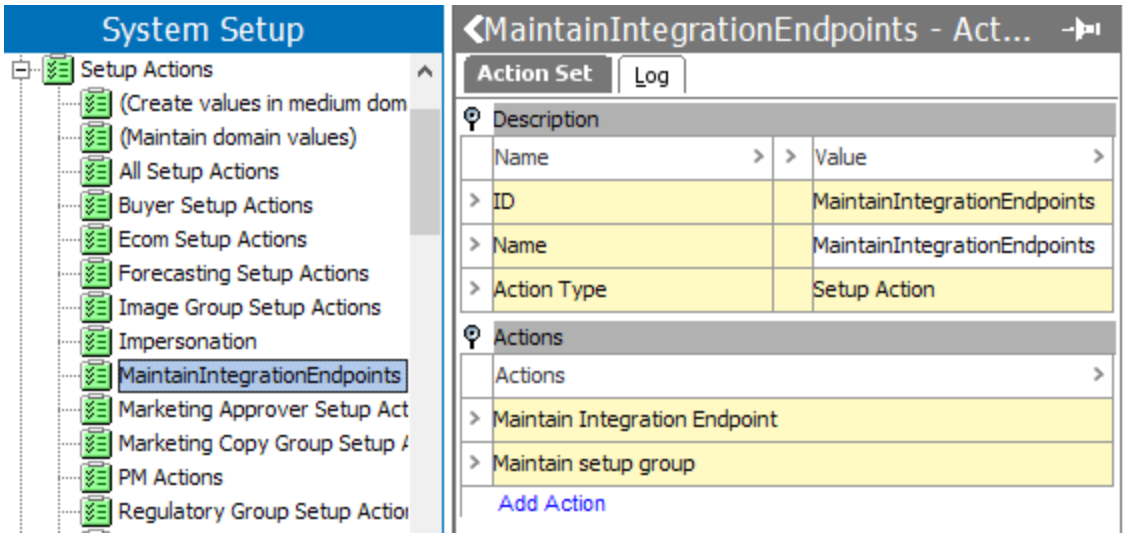


3. Select the newly created action and in the Action Set tab, click **Add Action**.
4. In the Select Action dialog, check **View Integration Endpoint** and **View Setup Group** and click **Select**.

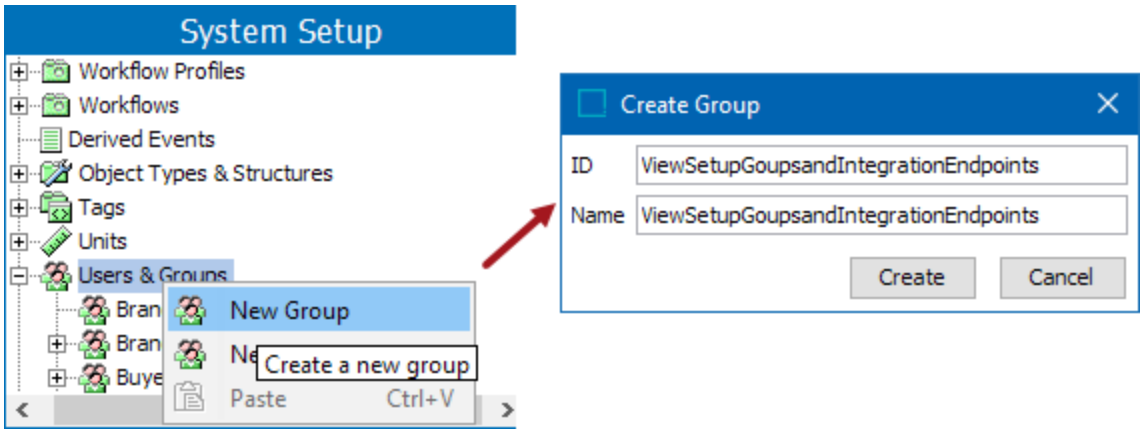




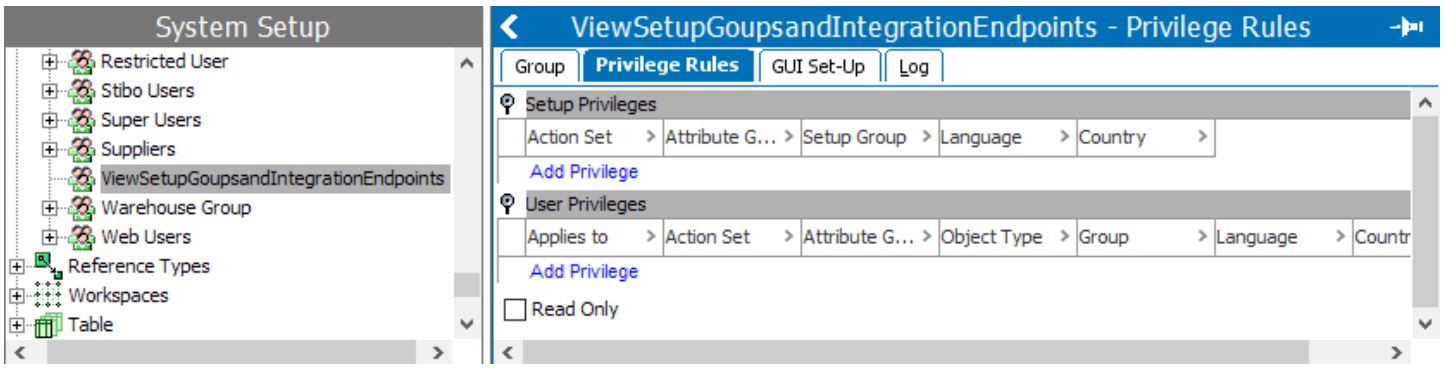
5. Repeat step 1-5 and create another Action Set containing the actions **maintain setup group** and **maintain integration endpoint**.



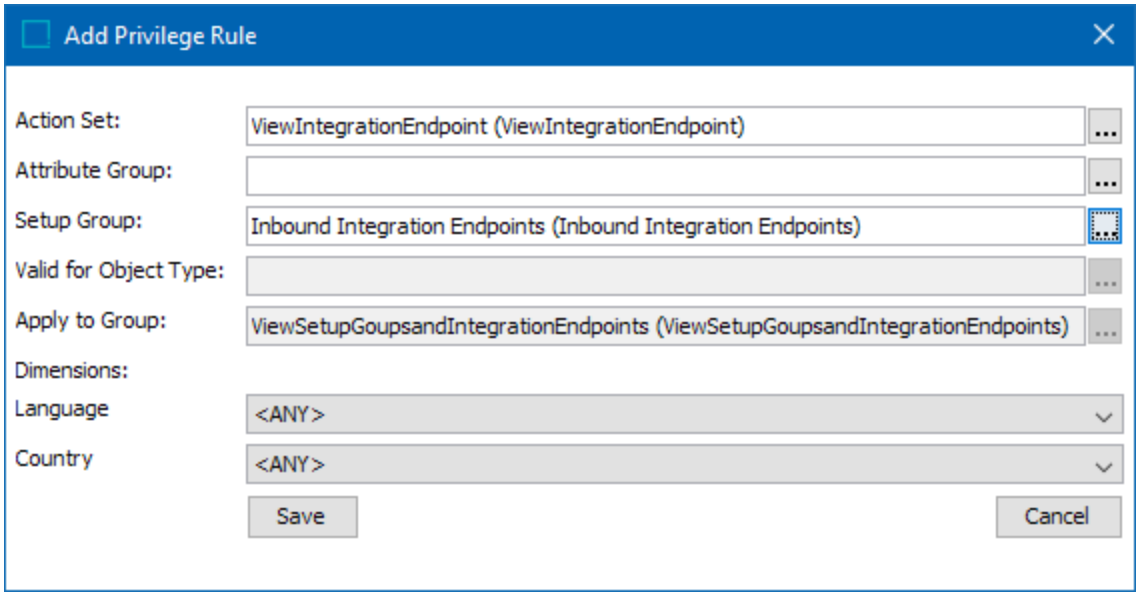
6. Go to the **Users & Groups** node, right-click and select **New Group**. Add an ID, Name, and click **Create**.



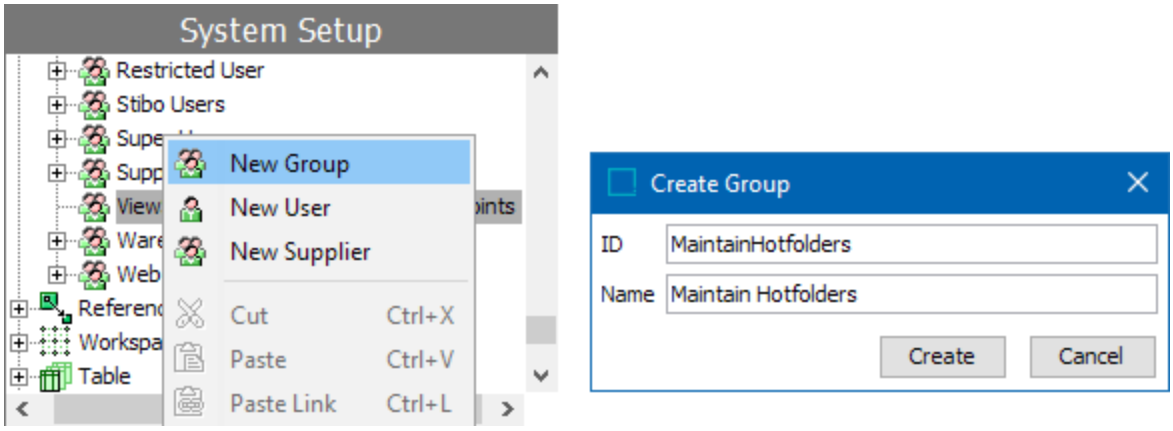
7. In **User Group editor**, click **Privilege Rules** tab.



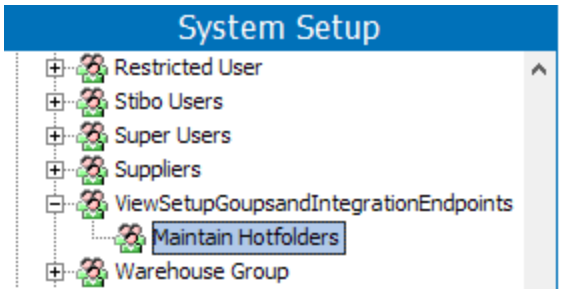
8. Under the **Setup Privileges** flipper, click **Add Privilege**. The Edit Privilege Rule dialog displays:



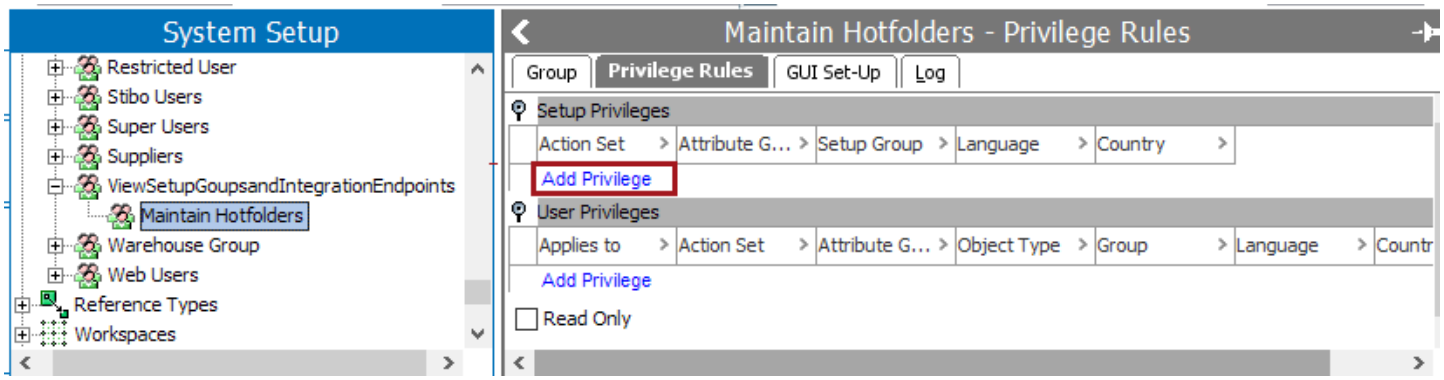
9. In the **Action Set** field, apply the **View Integration Endpoint** Action Set.
10. In the **Setup Group** field, apply the Setup Group named **Inbound Integration Endpoints**. Click Save when finished.
11. Next, in Users & Groups, select and right-click the group just created: **View Setup Groups and Integration Endpoints**. Select **New Group**. Add the ID, Name, and click **Create**.



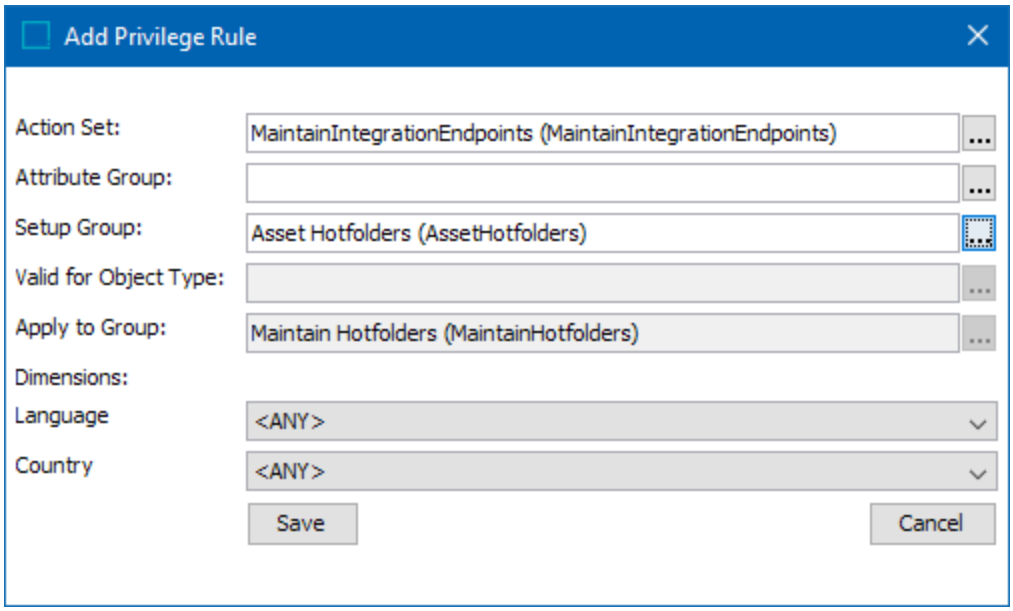
The new user group will appear below the user group **View Setup Groups and Integration Endpoints**.



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**.



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'.

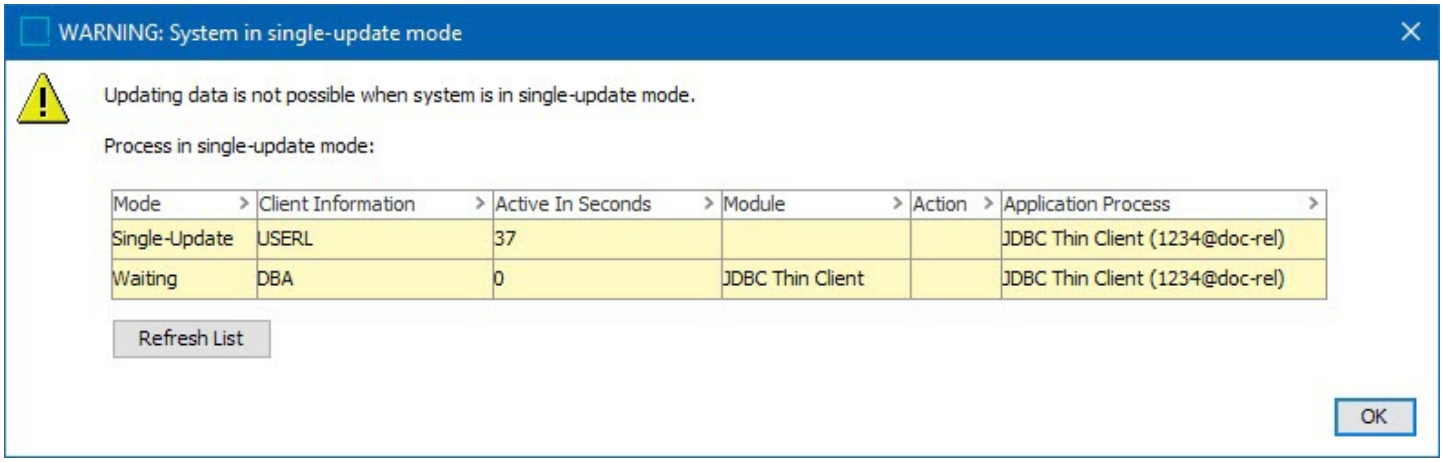
Note: The above is an example to set the User Privileges and necessarily doesn't mean that the only actions mentioned in the example are sufficient for a user to work on STEP. The user should also have other actions defined to work.

Single-Update Mode

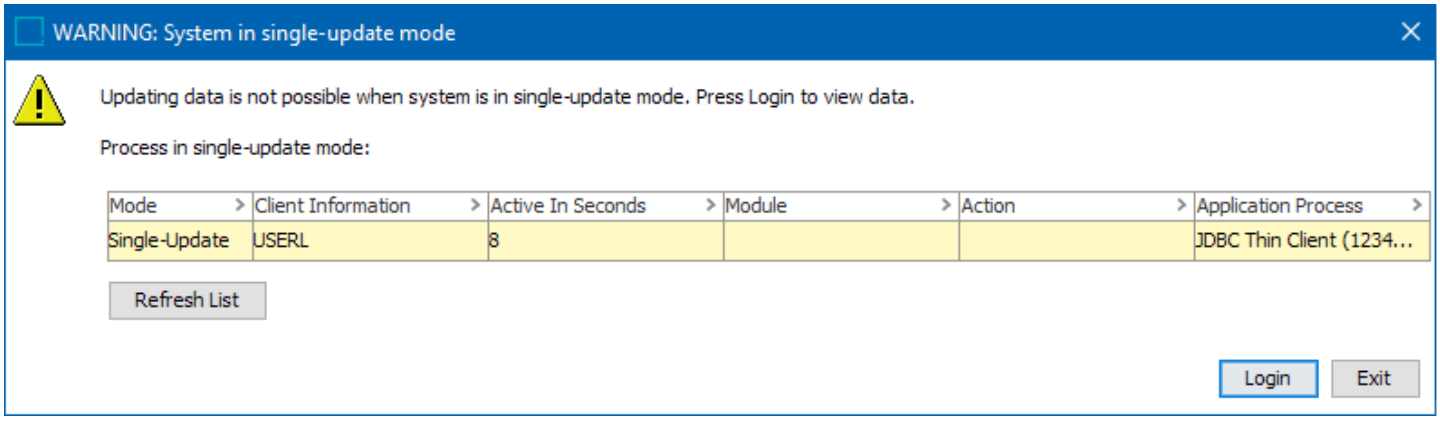
Certain system operations (i.e., GUI updates, data import) can corrupt data if they are run at the same time a different user is updating the database.

The system 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 displays.



Additionally, if you log on to the system while it is in single-update mode, a warning displays.



The warning displays information about which connection is currently running in single-update mode. While the user not in single-update mode cannot change data, they can still view data.

Note: The importer will only allow entry into 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 Guide** 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'.

Single-Update Mode Examples

The following examples are not intended to describe all scenarios in which the Single-Update Mode (SUM) is triggered, but are listed to illustrate some possible issues that SUM may introduce.

Example 1: Add / Remove a Dimension Dependency

If a user adds or removes a dimension dependency at the same time someone else is manipulating data that uses that is dimension dependency, then there is a risk of introducing inconsistent data in the database for the particular data that is being manipulated.

Note: Only the concurrently manipulated data is at risk.

For example, adding a dimension dependency to an attribute while data for this attribute is simultaneously being imported could result in inconsistent data. One way this might occur is if the importer caches the attribute's dimension dependency but does not notice the new dimension dependency has been applied until a number of seconds have elapsed. In that interim period, the values written for this attribute will have the wrong dimension dependencies.

Example 2: Merge Attributes

If one user is manipulating values on an attribute while a different user is merging that attribute, then there is a risk that inconsistent data may be introduced. This applies to all attributes participating in the merge.

Note: Only the concurrently manipulated values are at risk.

One example of how this might occur is if one user is importing values into the primary attribute at the same time another user is merging a different attribute into the same primary attribute. The merge operation might find a value of the secondary attribute to merge into the primary attribute, and at the same time, the other operation might create a new value for the same product for the primary attribute. These two values might be visible in same workspaces / contexts and be in conflict.

Another potential scenario where this could occur is when one user is importing values into an attribute that is about to be deleted by another user's operation. The import succeeds because the secondary attribute was still present as the second user's process was initiated.. However, once the merge is complete, the imported values are deleted with the attribute, thus the merge operation is unable to merge the values because the system did not detect the values in time.

Example 3: Changing the Property of an Attribute

This example covers the following four operations:

a. Change the 'Externally Maintained' flag of an attribute (i.e., store the historic values of an attribute or not)
b. Change an attribute to or from using LOV validation
c. Change an attribute to or from being a BLOB (Binary Large OBject). That is, making an attribute free text searchable or by changing its value length past the 100 bytes boundary
d. Change an attribute to or from being multi-valued

Changing one of these properties at the same time as another user is manipulating values of the attribute might result in inconsistent data.

Note: Only the concurrent manipulated values are at risk.

For example, changing one of these settings of an attribute while at the same time importing data for this attribute could lead to inconsistent data being imported for the attribute. The importer might have cached the original property of the attribute and not noticed the change until after a few seconds. In these few seconds, values written for this attribute will have an incorrect state, and these values will not be remedied by the batch operation since it has already moved past this point in the database table.

Guidance for Preventing SUM Activation

To avoid entering single-update mode when changing attribute definitions, the following procedure can be used:

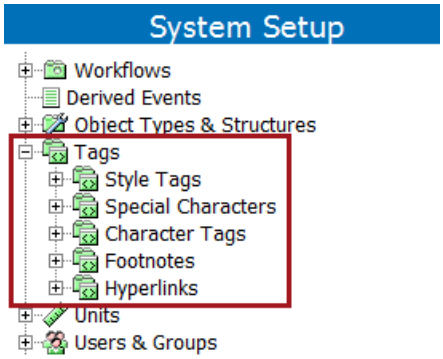
1. Create a new attribute (Attribute B) with the same desired attribute definition (including the changes from the origin attribute, Attribute A).
2. Use Bulk Update in the STEP Workbench to copy all attribute values from Attribute A to Attribute B. For more information on using bulk update functionality, see the **Bulk Updates** section of the **Getting Started / User Guide** 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' (such as style tags, hyperlinks, and footnotes) have a separate start and stop tag, e.g., <bold>xxxx</bold>.
- 'Unpaired tags' (such as special character and character tags) do not have a separate start and stop tag. Instead they are 'self-contained', e.g., <in/> or <ft/>.

For every STEP tag created the user can define what the appropriate corresponding command codes are for different media outputs. For example, the <bold> </bold> tags may be converted to and 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 STEP Tags

These are the main tag types:

Special Character

Special Characters are Unicode characters, which in some cases need to be formatted. 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.

Character Tags

Character Tags allow 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 (or any other character or characters that are not Unicode characters) such as a telephone ☎ or a special arrow ➔.

Style Tags

Style Tags apply changes to how text looks. For example, it can make plain black text bold, blue, or can format text as a list with bullets or numbers. Style Tags must have an opening and closing tag, for example, <bolditalic> </bolditalic>. Style Tags are used with STEP Tables in order to apply styling to the contents of table cells.

Footnotes

Footnotes are used in STEP Tables only to add an additional piece of information to attribute values when the attributes are used. 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. For more information, see the **Tables** documentation.

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.

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 text other than a URL in the value field.

Working With Tags

Follow these topics to use tags in STEP:

1. Create tags - see the **Creating Tag Groups and Tags** topic in this documentation.
2. Configure tags properly - see the **Configuring Tag Definitions** topic in this documentation.
3. Establish proper output formatting - see the **Configuring Tag Output Formatting** topic in this documentation.
4. Understand more about tags in attribute values - see

- the **Special Character Tags** topic.
- the **Character Tags** topic.
- the **Style Tags** topic.
- the **Hyperlink Tags** topic.

5. Make tags visible in Web UI - see the Enable Tag Conversion parameter of the **Attribute Value Component** topic in the **Web User Interfaces / Web UI Setup and User Guide** documentation.

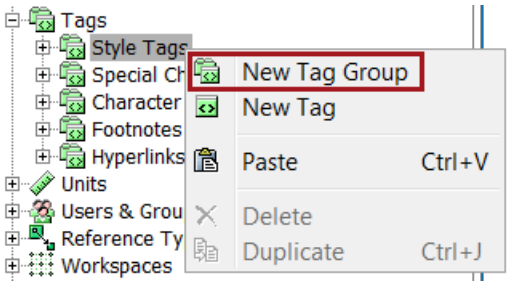
Creating Tag Groups and Tags

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

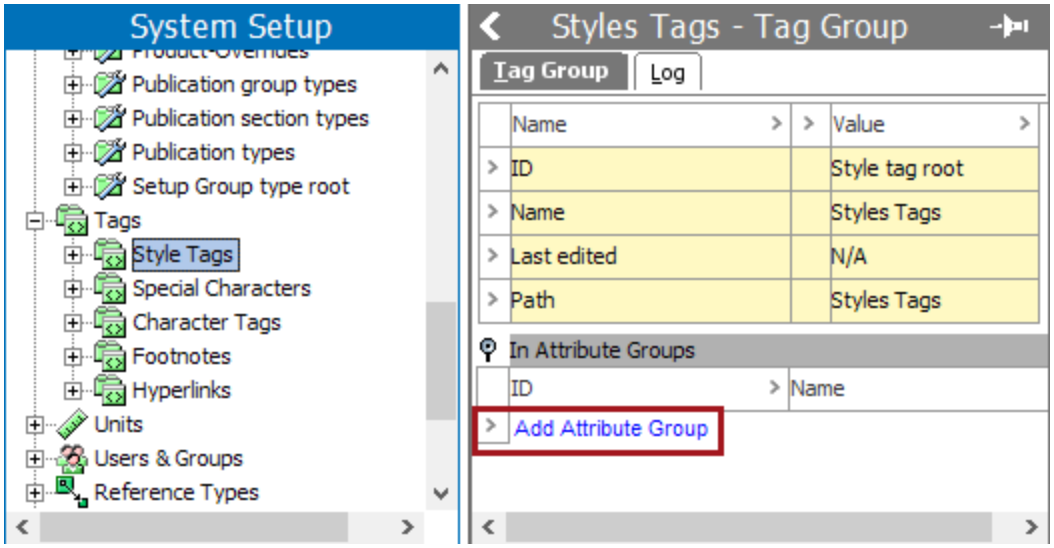
Follow the steps below to create a tag group under one of the types of tags:

1. In **System Setup**, right-click Special Characters, Style Tags, Character Tags, Footnotes, or Hyperlinks.
2. In the menu, select **New Tag Group**.



3. Add an **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 available for tighter control over tag usage are 'Use tag' and 'Maintain tags'. See the **Users and Groups** section of this documentation for more information on how to set up privilege groups.

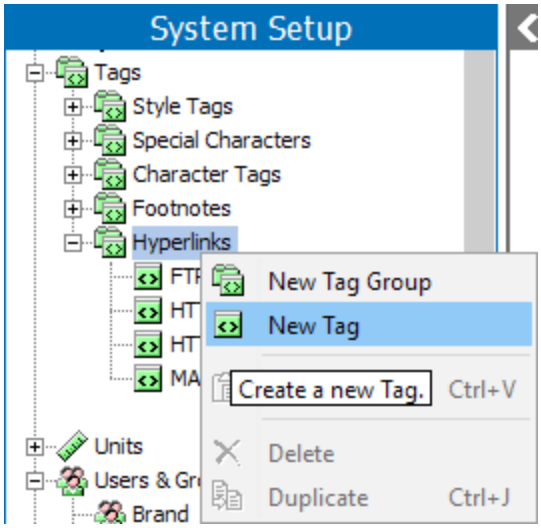
Tags

The method to create the various tags are generally the same, although there are some special considerations as defined below.

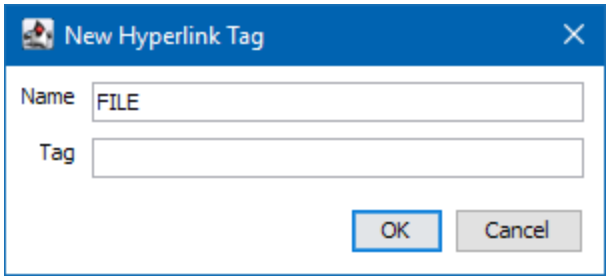
Important: *All* STEP tags are case-sensitive. When loading files with attribute values that include STEP tags, they must exactly match how the tag was set up.

Follow the steps below to create a tag:

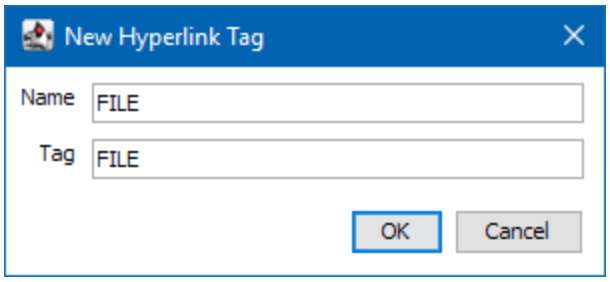
1. Go to the **System Setup** tab, expand **Tags**.
2. Right-click on the desired tag type, then select **New Tag**.



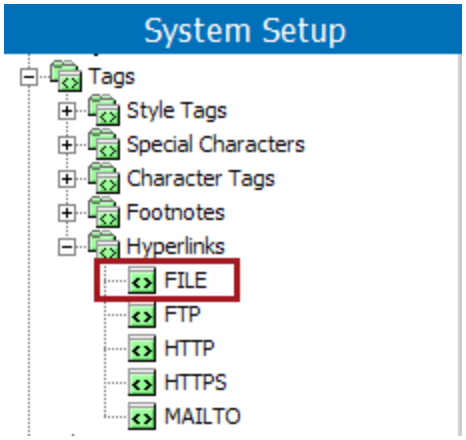
3. In the dialog that appears, enter a display **Name** for the tag. The name can be up to 80 characters long, and can include spaces.



4. In the **Tag** field, enter a tag. The tag will be used, for example, in a document when importing files using STEP tags. Spaces and the special characters < (less than), > (greater than), or / (forward slash) are not permitted in the tag.



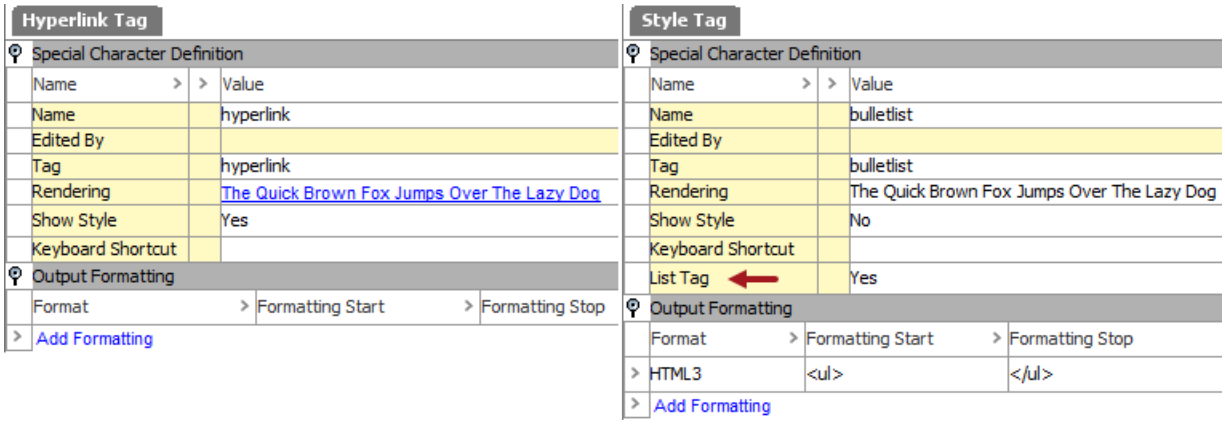
5. Click **OK**. The tag is created, and is available in the hierarchy.



6. Configure and define rendering of the tag as defined in the **Configuring Tag Definitions** topic in this documentation.

Configuring Tag Definitions

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. Generally, configuring tags is the same regardless of the type.



On the Tag tab (a Hyperlink and a Style tag are shown above), the Special Character Definition flipper includes the following editable parameters:

- **Name** - displays the name of the tag. The name can have a maximum of 80 characters. Spaces between words are allowed.
- **Tag** - 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** - for hyperlinks, character tags, and style tags, see the **Rendering** section below.
- **Show Style** - specify whether or not the tag is shown in the dropdown 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** - specify a keyboard shortcut to use to insert the tag.
- **List Tag** - for style tags only, 'Yes' defines the start and end of an entire bullet or numbered list for HTML output. For example, the style will indicate the use of the <ul> … HTML tags. For more information, see the **Configuring tags for a list** section below.

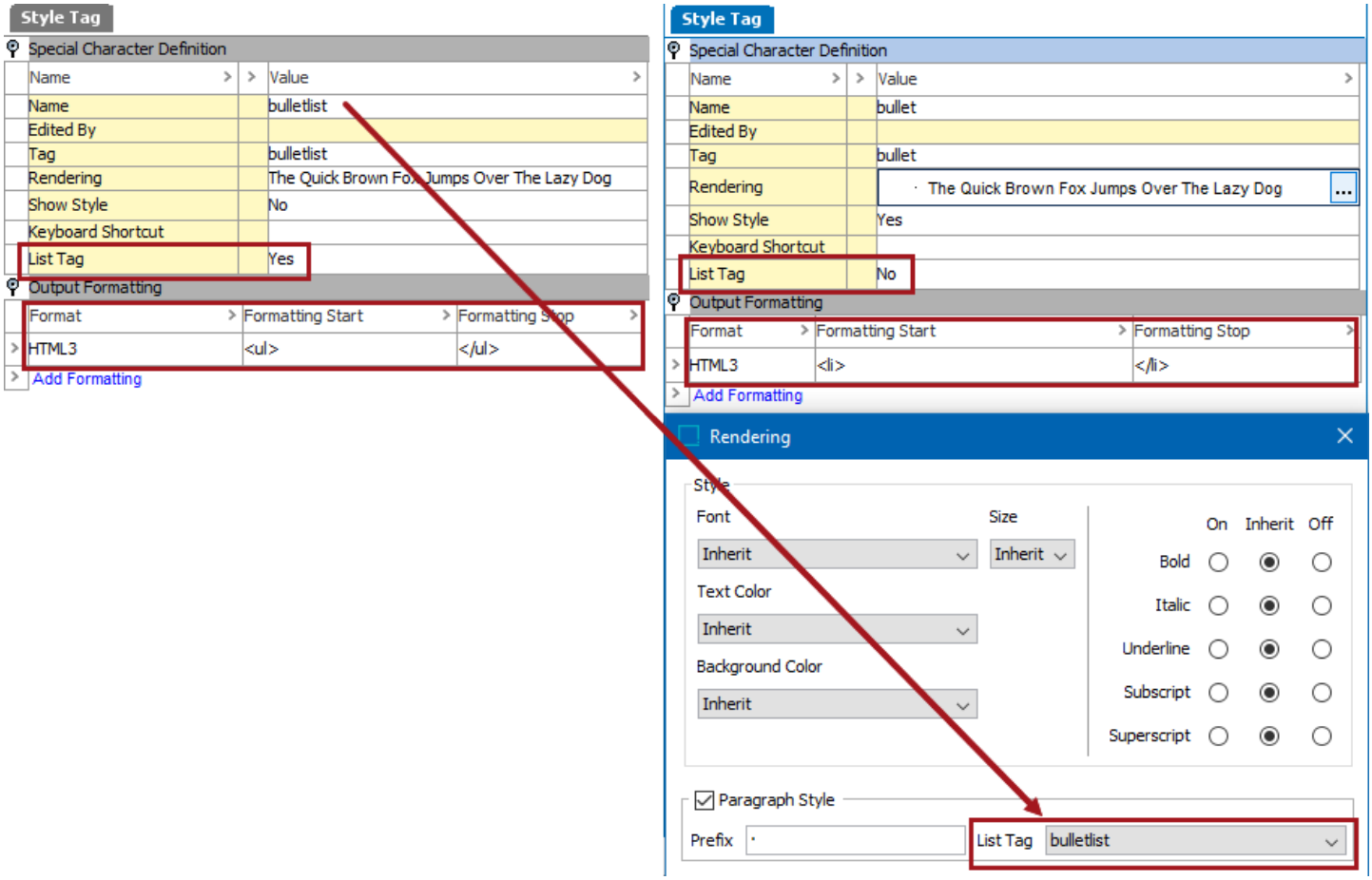
  Multiple level / nested level lists are not allowed using this method. However, a conversion to HTML can specify a numbered list (such as is done by an … command), and that will cause a numbered list to display on the web.

After completing the Special Character Definition flipper parameters, format the output as defined in the **Configuring Tag Output Formatting** topic in this documentation.

Configuring tags for a list

A list requires two tags:

- one tag defines the start and the end of the entire list; List Tag parameter = Yes; … tags are defined
- a second tag defines the individual items in the list; List Tag parameter = No; … tags are defined



For example, in the image above, the 'bulletlist' tag includes the tags that surround the entire list. And the 'bullet' tag defines how items in the list are formatted, and identifies the 'bulletlist' tag as the list wrapper.

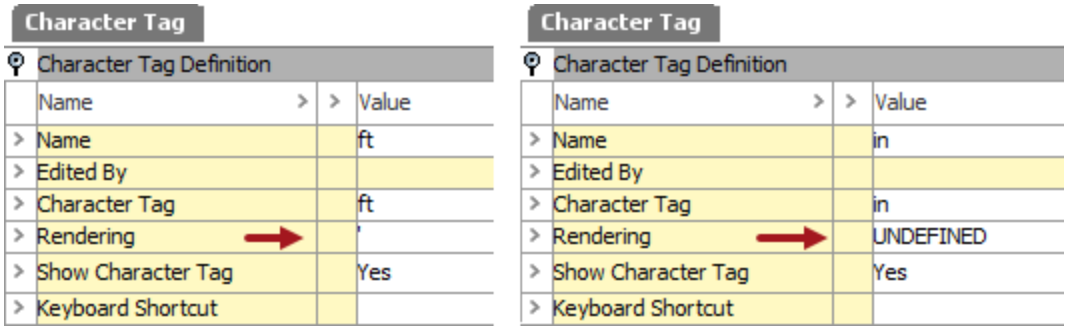
For details, see the **Rendering** section below.

Rendering

The Rendering parameter within a tag definition displays how the tag will be displayed within STEP and for downstream systems.

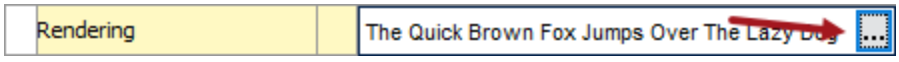
Character Tags

To determine how a Character Tag is rendered, enter character(s) directly into the value field, otherwise UNDEFINED is displayed. For example, the tag for 'foot' (ft) is defined, but the tag for 'inches' (in) is not defined.

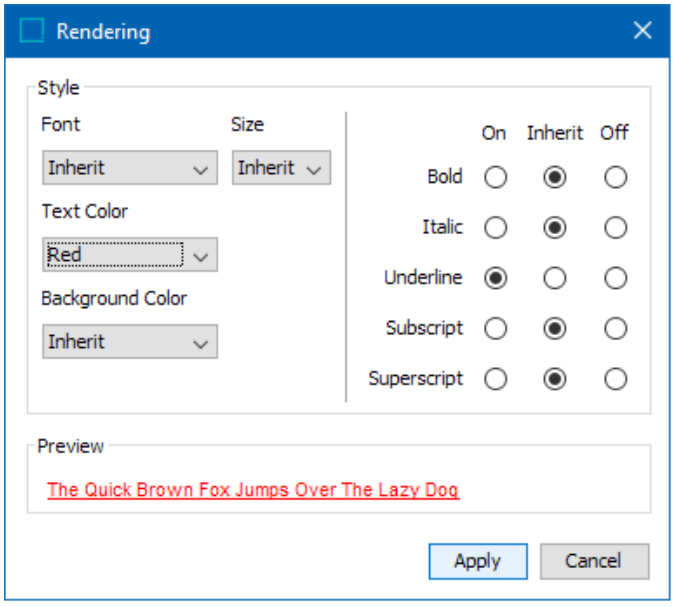


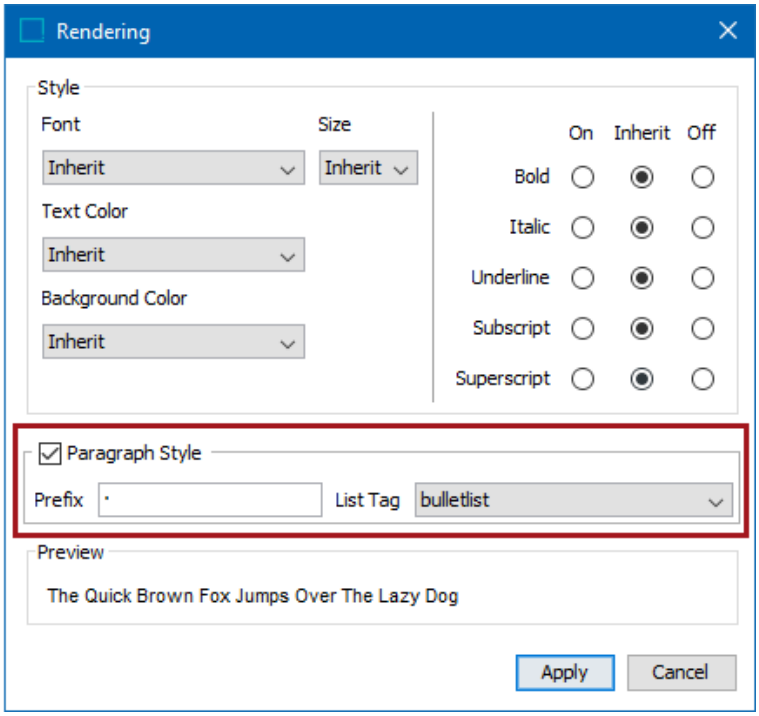
Hyperlinks and Style Tags

To define how a Hyperlink or a Style Tag is rendered, click the Rendering value field to display an ellipsis button.



Click the ellipsis button (...) to open the appropriate Rendering dialog. Below, a hyperlink tag is shown first, and a style tag rendering dialog (with the additional Paragraph Style group) is shown second.





Style Settings

By default, a tag inherits all settings, but a number of styling options can be set to modify the look of the text. When configuring the Rendering parameter for a hyperlink or style tag, the Preview section of the Rendering dialog shows how a tag is set to display in the user interface.

For both hyperlinks and style tags, use the Style area to define the following style settings:

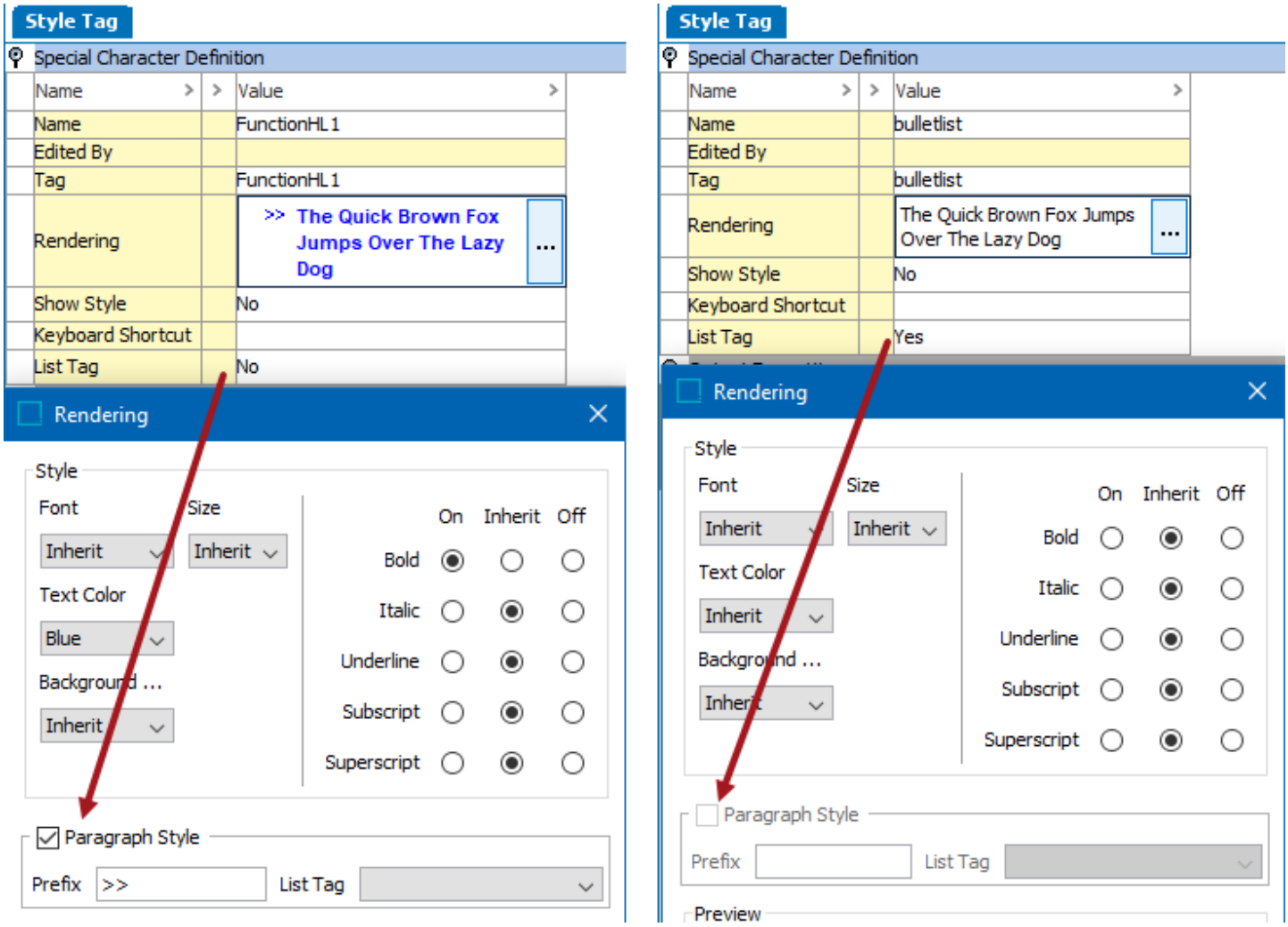
Style	Description
Font	Select a font from the fonts installed on the computer to be used in the rich text editors, or choose **Inherit** to use the font from the parent.
Size	Select a size from the list of available font sizes to be used in the rich text editors, or choose **Inherit** to use the size from the parent.
Text Color	Select a color from the list of standard text colors, or choose **Inherit** to use the color from the parent.
Background Color	Select a color from the list of standard background colors, or choose **Inherit** to use the color from the parent.
Bold, Italic, Underline, Subscript, Superscript	Click the **On** radio button to apply the selected style tag, the **Inherit** radio button it take the style from the parent, or the **Off** radio button to turn the styling off.

Paragraph Style Settings

The Paragraph Style parameter group determines styling for a paragraph. This allows adding introductory text or characters to every paragraph within an attribute value.

Follow these steps to configure the Paragraph Style section:

1. Set the 'Paragraph Style' checkbox as follows, based on the tag's Definition flipper List Tag setting:
   - If **No**, check the 'Paragraph Style' checkbox to add text that will be displayed on each paragraph.
   - If **Yes**, leave the 'Paragraph Style' checkbox unchecked.



2. When available, in the 'Prefix' parameter, enter a character or string of text to be used as prefix for a paragraph or for the bullet list, such as >> or •.
3. When required, define the style for the surrounding bullet list tags in the 'List Tag' parameter, select the style tag created to start and end an entire bullet list (the <ul> … tags).

   For details on implementing list styles, see the **Styling lists** section of the **Style Tags** topic.
4. Click the **Apply** button.

Configuring Tag Output Formatting

Because all STEP tags are media neutral, a conversion of the STEP tag can be set up to accommodate the requirements of downstream systems. The conversion is based on the Output Formatting flipper in the tag and the Tag Conversion parameter in the output tool. Both elements are defined below.

Output Formatting Flipper

Multiple output formats can be added to each tag to accommodate the requirements of the different systems receiving the 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.

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	Formatting Stop
HTML3		
Add Formatting		

Character Tag

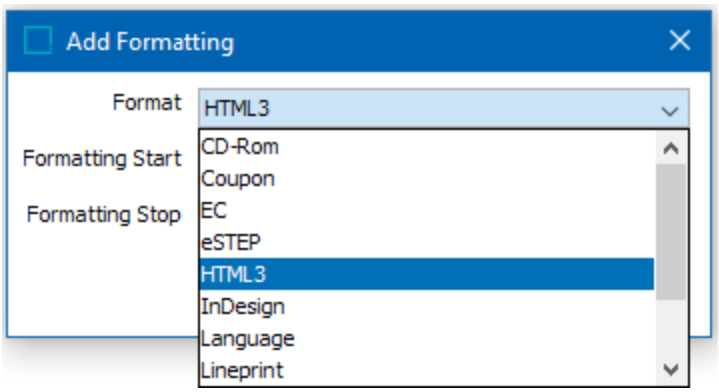
Character Tag Definition

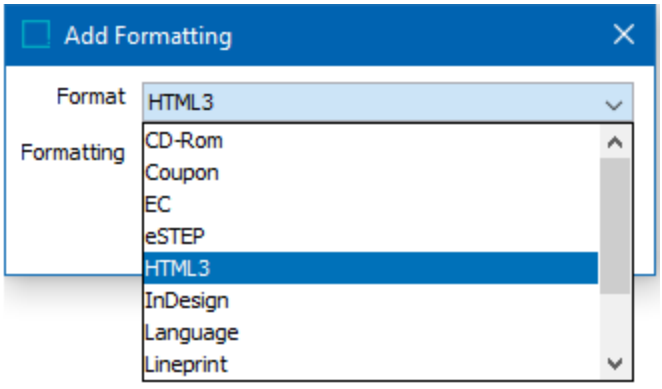
Name	Value
Name	ft
Edited By	
Character Tag	ft
Rendering	'
Show Character Tag	Yes
Keyboard Shortcut	

Output Formatting

Format	Formatting
InDesign	'
HTML3	&#162
Add Formatting	

1. On a tag editor, open the Output Formatting flipper and click the **Add Formatting** link.
2. For the **Format** parameter, choose an output format from the dropdown menu. This is just a name and no predefined set of codes is being chosen. The name allows the user to set up one or more different tag conversions to meet the requirements of different downstream systems or applications.



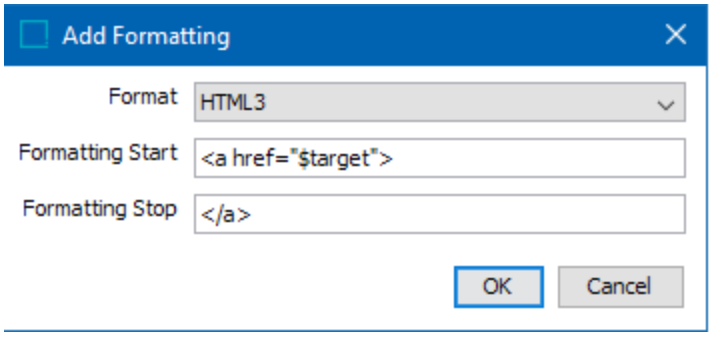


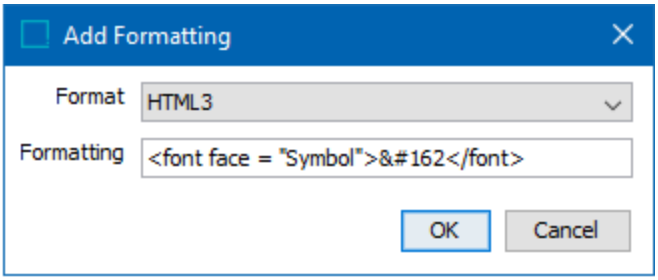
For a web application the output format name of 'HTML3' is usually selected, and output formatting is set up to convert all the STEP media neutral tags to appropriate HTML tags.

3. Enter the appropriate formatting information as follows:

   - the **Formatting Start** parameter start tag for the selected output format name in and the end tag in
   - the **Formatting Stop** parameter if adding formatting for Style Tags, Footnotes, and Hyperlinks.
   - the **Formatting** parameter, for Special Characters and Character Tags only, 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.





4. Click **OK** to add the formatting, or **Cancel** to return to the tag editor.

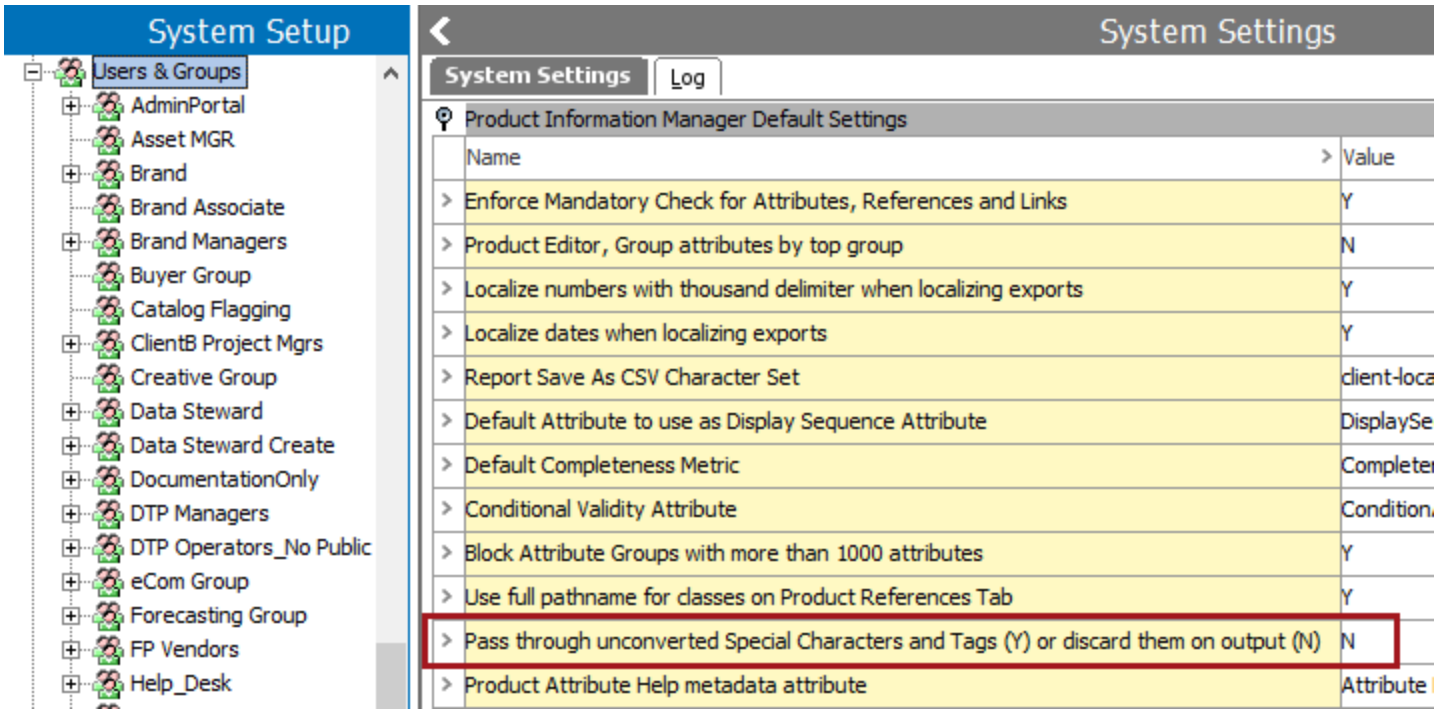
   The format and formatting / tags display in the table.

5. On System Setup, click the Users and Groups node, and open the Product Information Manager Default Settings flipper and find the **Pass through unconverted Special Characters and Tags (Y) or discard them on output (N)** parameter.

   Set the parameter as follows to handle formats when no Formatting entry exists:

   - **N** results in style tags, character tags, and special characters being stripped out upon output.
   - **Y** results in style tags, character tags, and special characters being output as STEP Tags.

   For more information, see the **Product Information Manager Default Settings** in the **System Setup / Super User Guide** documentation.

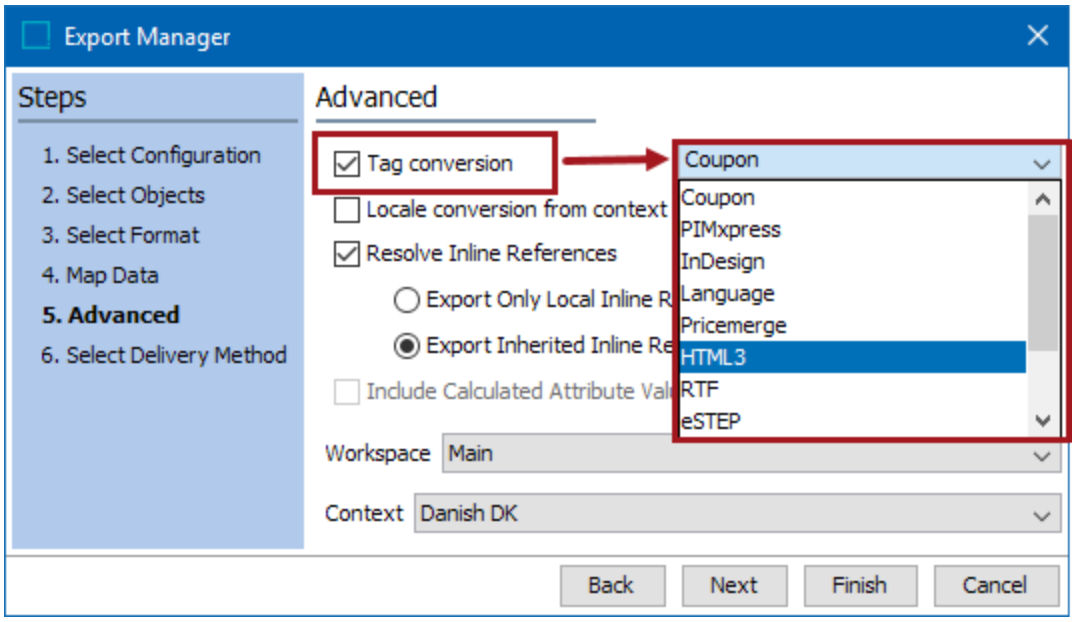


InDesign Output Formatting

For more information, see the **Publication Template Layout, Formatting, and Styles** topic in the **STEP Publisher (InDesign)** documentation.

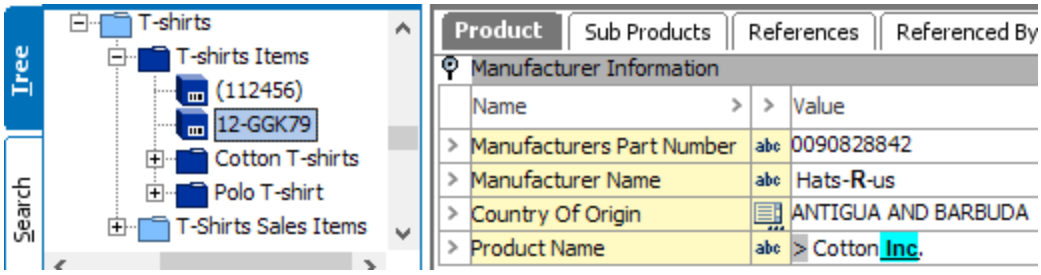
Tag Conversion Parameter

The tag conversion parameter in the Export Manager or the Outbound Integration Endpoint output tools determines how the exported tags are processed by downstream systems. The conversion options available are defined in the Output Formatting flipper within the tag editor.



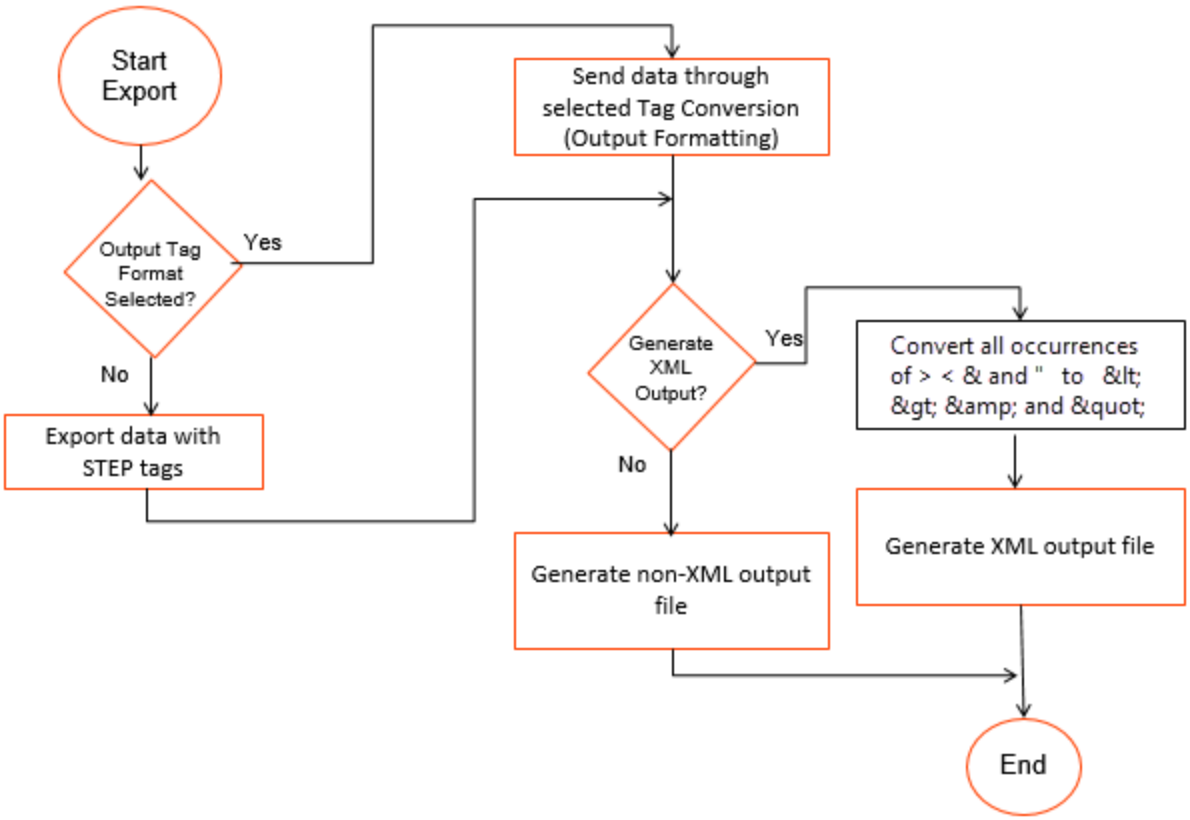
For more on output tools, see the **Export Manager** and the **Outbound Integration Endpoints** topics in the **Data Exchange** documentation.

The picture and table 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	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>.

When outputting XML files, the conversion of < and > characters, etc., comes after it has gone through tag conversion. So for XML output, you cannot force a < or a > unless a custom solution is in effect.



The same is true 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 generally not supported, with these exceptions shown in the table below.

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 within an XML file. They are as follows:

Name	Symbol	Predefined Entities / Character Entity Reference	Unicode number
double quotation mark	"	"	Unicode 0022

Name	Symbol	Predefined Entities / Character Entity Reference	Unicode number
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. Using characters in the ranges &#0; to &#31; and &#127; to Ÿ should be avoided since they are not Unicode characters.

Special Character Tags

Special Characters are not really 'tags' per se, but this feature provides the ability to convert a single Unicode character in STEP to one or more characters on output. 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.

Note: In the ASCII character set, only the characters from the space character through the tilde character are available for object IDs. These are Decimal 32 to Decimal 126, or Hexadecimal characters 20 to 7E.

The following standard special characters are predefined and 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.

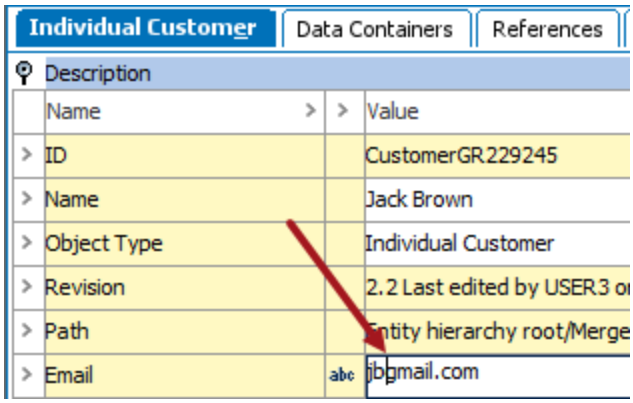
Standard Special Characters	Description
No-break Space (NBSP)	'No-break Space' character is used to prevent splitting a line at a space
Line Feed (LF)	Line Feed (LF) is a hard return

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.

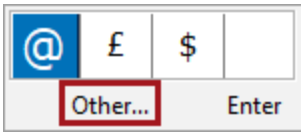
Insert a Special Character

Use the following steps to insert a special character.

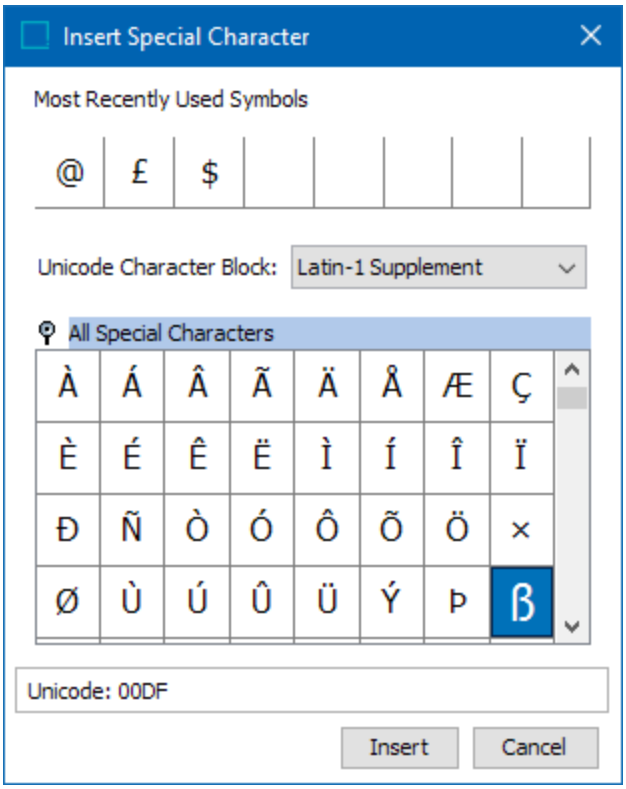
1. In an attribute value field, position the cursor where the special character should be.



2. Right-click and choose the 'Insert Special Character' option.
   - If special characters have been used previously, a dialog displays with a small number of the most recently used symbols. If the required character is displayed, click the character to add it to the value, or to display additional options, click the **Other...** option (or press the Enter keyboard key).



   - On the 'Insert Special Character' dialog, choose a Unicode Character Block from the dropdown, and select a special character.



3. Click the **Insert** button to insert the special character, or click the **Cancel** button to return to the object editor.

When inserted, the selected character is displayed in the value of the attribute where added.



Character Tags

Character Tags are not paired, meaning that they do not require a opening and closing tag. For all character tags, the character(s) entered in the Rendering parameter display in the workbench GUI with a gray background. For example, >.

Because there are several thousand special characters available on many different types of fonts, it may be necessary to create 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.

> and < Characters

Recommended practice is to set up tag entries for 'less than' <lt/> and 'greater than' <gt/> characters along with their rendering. This enables a user to enter the < or > directly on the keyboard. STEP will take this single keystroke and immediately converts that to the STEP tag <lt/> or <gt/>, and shows in the workbench as the 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

Although the < and > characters are allowed in an object ID, it is not recommended practice. If these character tags are entered within an object's name, STEP allows them and saves them as the STEP character tags <lt/> and <gt/>. However, this may cause 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 throws an error, though it will still load the data and will transform the < and > characters to an <lt/> or < gt/>.

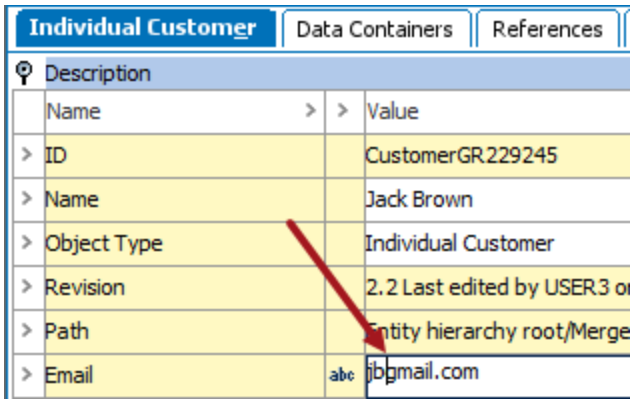
& and " Characters

The & (ampersand) and the " (double straight quote) characters are 'automatic' when converting to on output. An accepted industry standard is to use the *&* and *"* when these characters occur in a text stream in XML. However, do not enter *& and "* directly into STEP attribute values in the workbench or Web UI as it will not give the desired results. Instead, enter & and " directly from the keyboard.

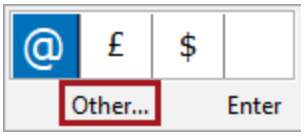
Insert a Character Tag

Use the following steps to implement a character tag.

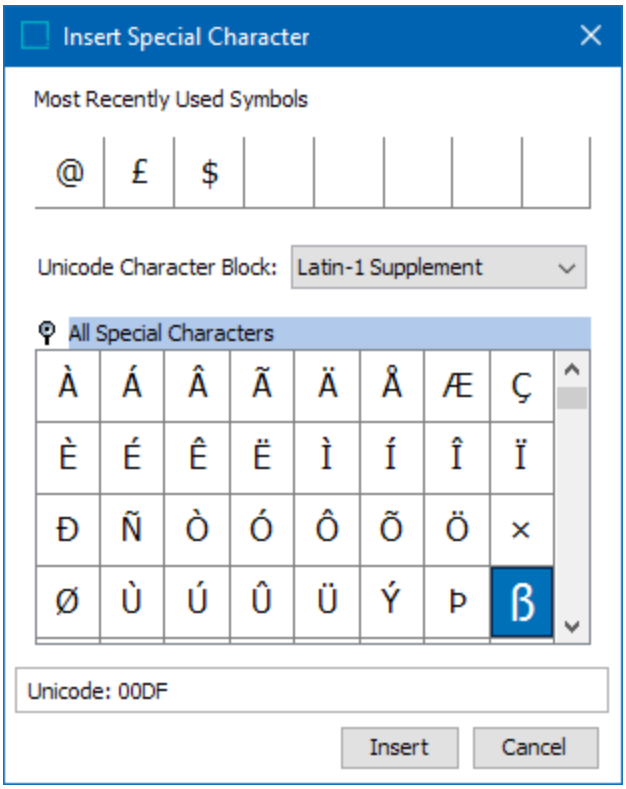
1. In an attribute value field, position the cursor where the special character should be.



2. Right-click and choose the 'Insert Special Character' option.
   - If special characters have been used previously, a dialog displays with a small number of the most recently used symbols. If the required character is displayed, click the character to add it to the value, or to display additional options, click the **Other...** option (or press the Enter keyboard key).



   - On the 'Insert Special Character' dialog, choose a Unicode Character Block from the dropdown, and select a special character. Click the **Insert** button or click the **Cancel** button to return to the object editor.



Style Tags

Style Tags use paired tags to change how text looks. The system automatically inserts the symbols < / >.

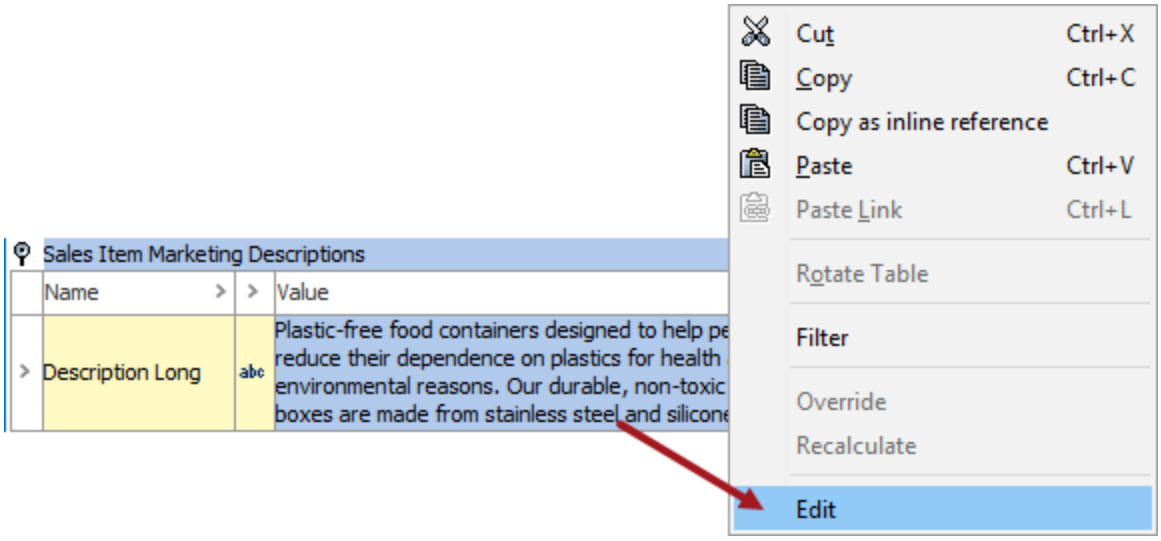
Style tags cannot be used within an object's name, but can be used to define how to display part of an attribute value (such as a word or two within a paragraph), or that the complete value should be displayed as a list. The functionality for creating bullets is an accommodation in STEP that allows lists to be produced correctly for HTML output. Since InDesign has no such stringent requirements, the special handling is just ignored.

Insert a Style Tag

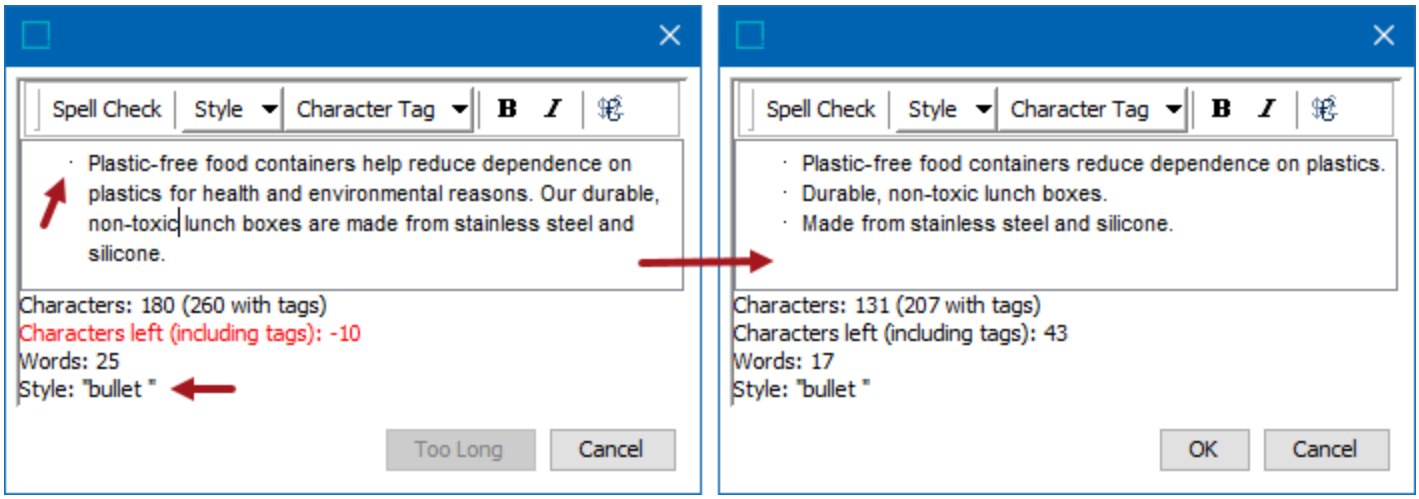
The following methods are available: styling lists, styling individual words, and styling multiple words.

Styling lists

1. Verify the two style tags required for a bullet list already exist, see the **Configuring tags for a list section** of the **Configuring Tag Definitions** topic.
2. Click the attribute value field that requires a list format, right-click, and choose **Edit** to display the large editor.



3. In the large editor, click into the text, click the Style and apply the individual bullet item style, then update the text by pressing the **Enter** for each bullet item.

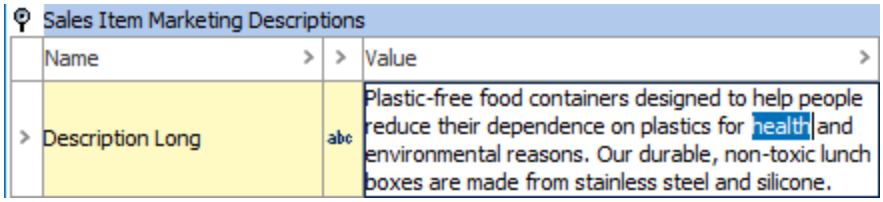


4. Click **OK** to apply the changes.

For details on options in the large editor, see the **Styling multiple words** section below.

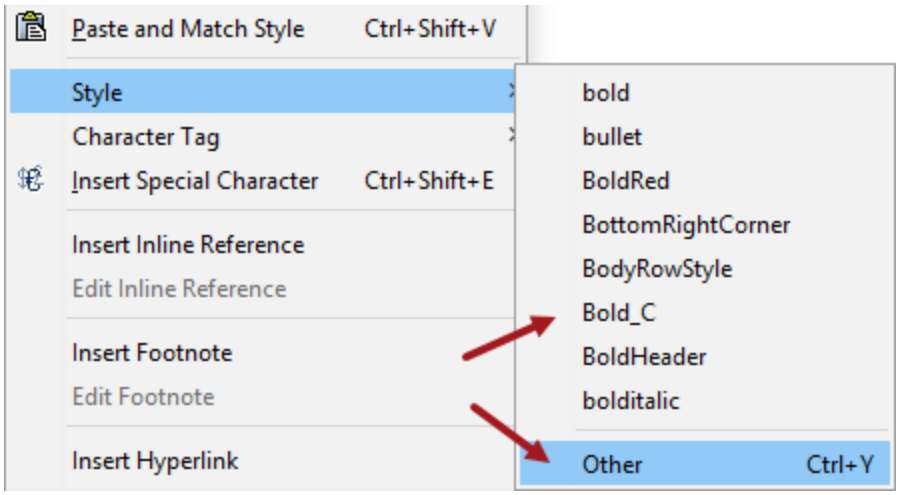
Styling individual words

1. In an attribute value field, select the text for the style tag.



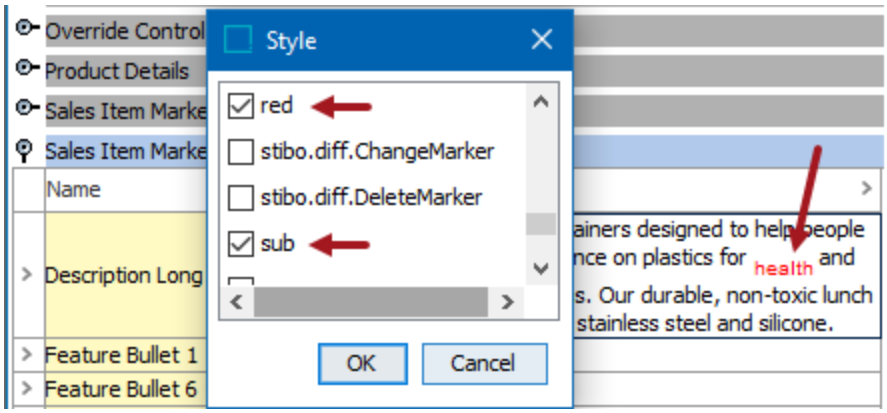
2. Right-click, highlight the 'Style' option, and choose a style as follows:

- Select from the list of recently used tags. Only one style can be applied at a time.
- Click the **Other** option to display a Style dialog with all styles, as shown in the following step.



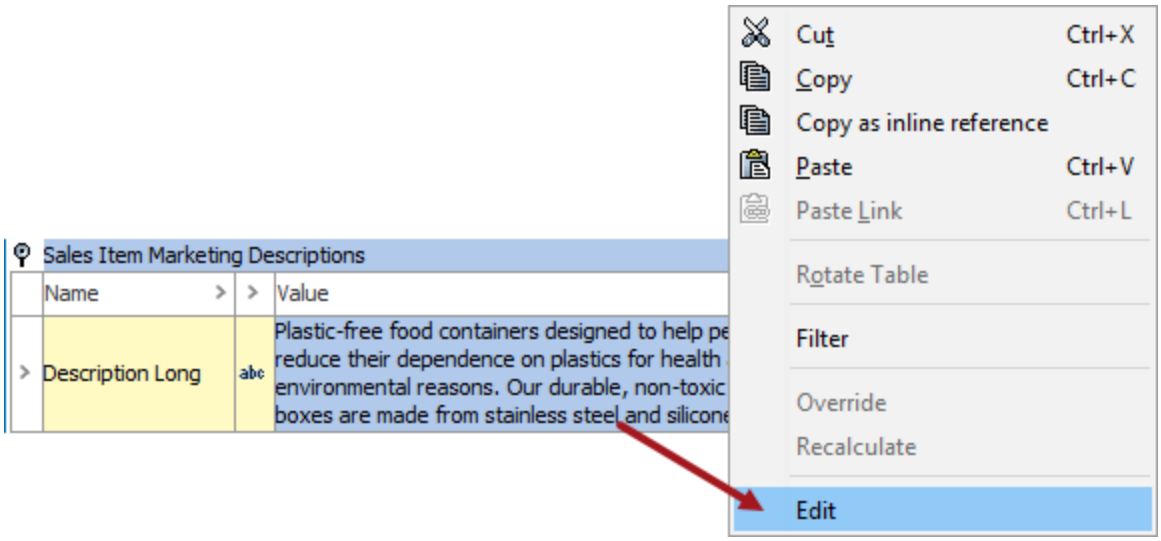
3. In the Style dialog, select one or more styles and click **OK**.

   In this example, the 'red' (color) and 'sub' (subscript) styles are selected, and the original selection is updated to show both styles.

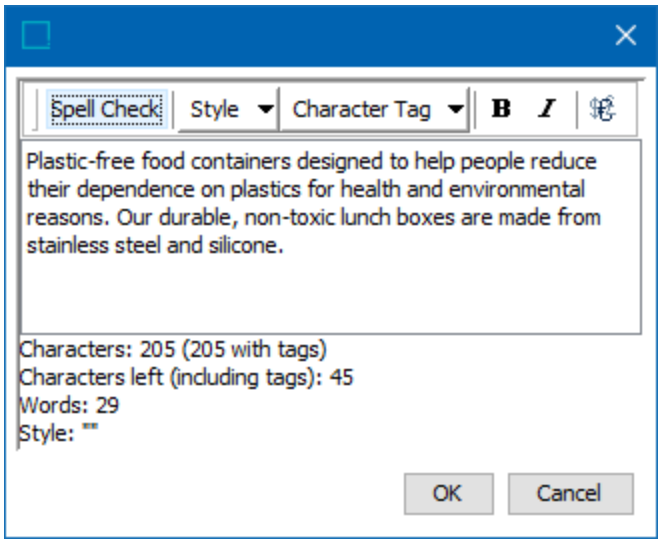


Styling multiple words

1. Click the value field, right-click and choose **Edit** to display the large editor.

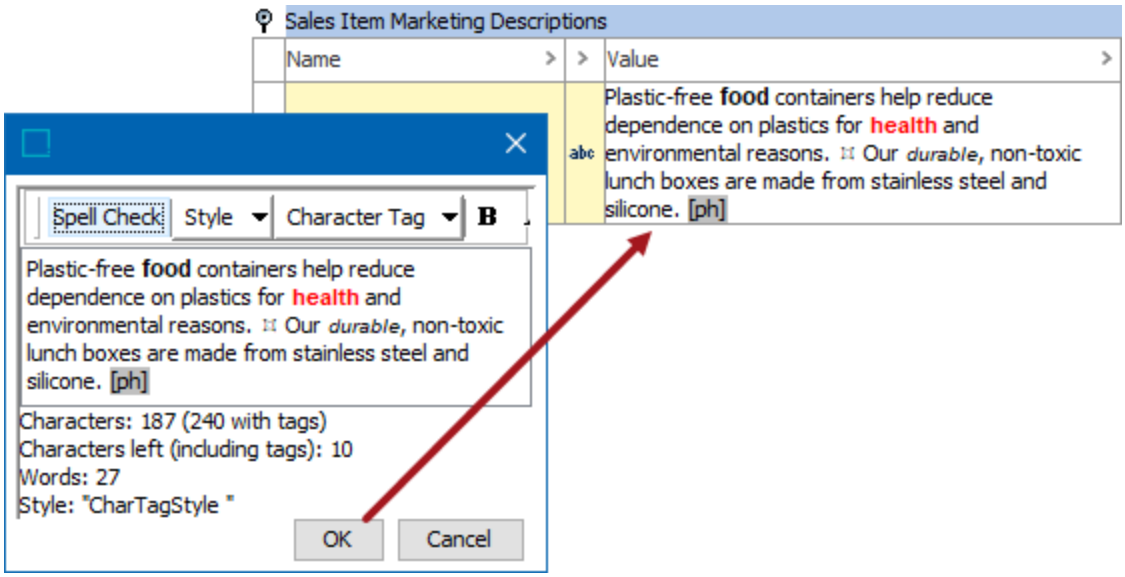


2. In the large editor, select the text to be styled and apply the appropriate tags as follows. Additional options are defined below:



- **Spell Check** - click to highlight incorrectly spelled words in red / italics.
- **Style** - select text and use the dropdown to apply one or more style from the available style tags.
- **Character Tag** - position cursor at the point where the character should be added and use the dropdown to insert the character.
- **B** - select text and click the button to bold the selection.
- *I* - select text and click the button to add italics to the selection.
- - position cursor at the point where the special character should be added and use the dropdown to insert the special character.
- To remove a style, select the styled text and either uncheck the applied style, or click the bold or italics button to remove it, or delete the text and retype it.
- The 'Characters left (including tags):' information warns the user when the text exceeds the limit. When the limit is exceeded, the OK button text is replaced with the text 'Too Long' and the changes cannot be saved.
- The 'Style:' information displays the tags applied at the point of the cursor (as shown in other images in this topic).

3. Click **OK** to apply all styling changes and see the effect in the object editor.


Sales Item Marketing Descriptions
Name
Value
Plastic-free food containers help reduce dependence on plastics for health and environmental reasons. ⌗ Our durable, non-toxic lunch boxes are made from stainless steel and silicone. [ph]
Spell Check
Style
Character Tag
B
Plastic-free food containers help reduce dependence on plastics for health and environmental reasons. ⌗ Our durable, non-toxic lunch boxes are made from stainless steel and silicone. [ph]
Characters: 187 (240 with tags)
Characters left (including tags): 10
Words: 27
Style: "CharTagStyle "
OK
Cancel

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 format types of hyperlinks: FILE, MAILTO, FTP, HTTP, and HTTPS. This way the hyperlinks can 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.

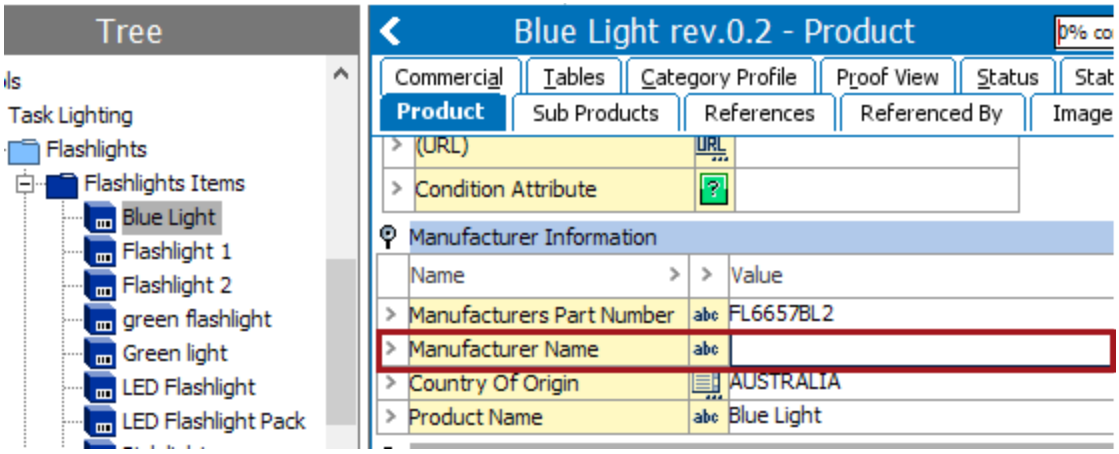
For print, these tags may be used on PDF documents created in InDesign. If the reader opens the PDF and clicks on a hyperlink in the PDF, they are then directed to the relevant external website.

For information on creating and configuring Hyperlink tags, see the **Creating Tag Groups and Tags** and **Configuring Tag Definitions** topics in this documentation. Once setup, the Hyperlink tag will also require an output format. For more information, see the **Configuring Tag Output Formatting** section of this documentation.

Insert a Hyperlink

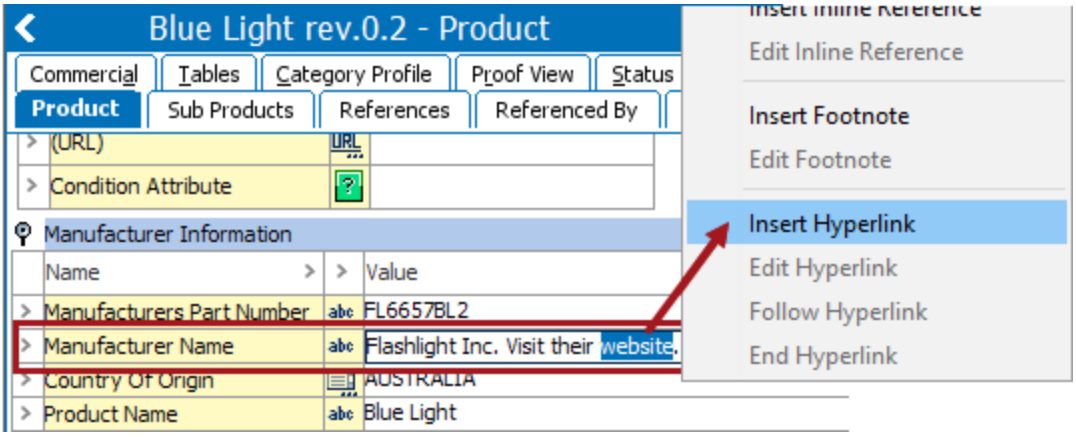
Use the following steps to implement a hyperlink.

1. Navigate to the desired object in STEP and click into the attribute with the validation base type of text.



2. In the attribute field, enter the required text, including the text to be used for the hyperlink.
3. Highlight the hyperlink text, right-click, and choose **Insert Hyperlink** from the menu.

   In this example, 'website' will be the hyperlink text.



4. On the 'Insert Hyperlink' dialog, leave the 'Hyperlink type' dropdown set to 'hyperlink.' In the URL parameter type the required URL and click **OK**.



Note: If the URL is for a public folder in the user's intranet, enter the folder path.

The rendering that was specified for the hyperlink is displayed, as shown below.

5. 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.



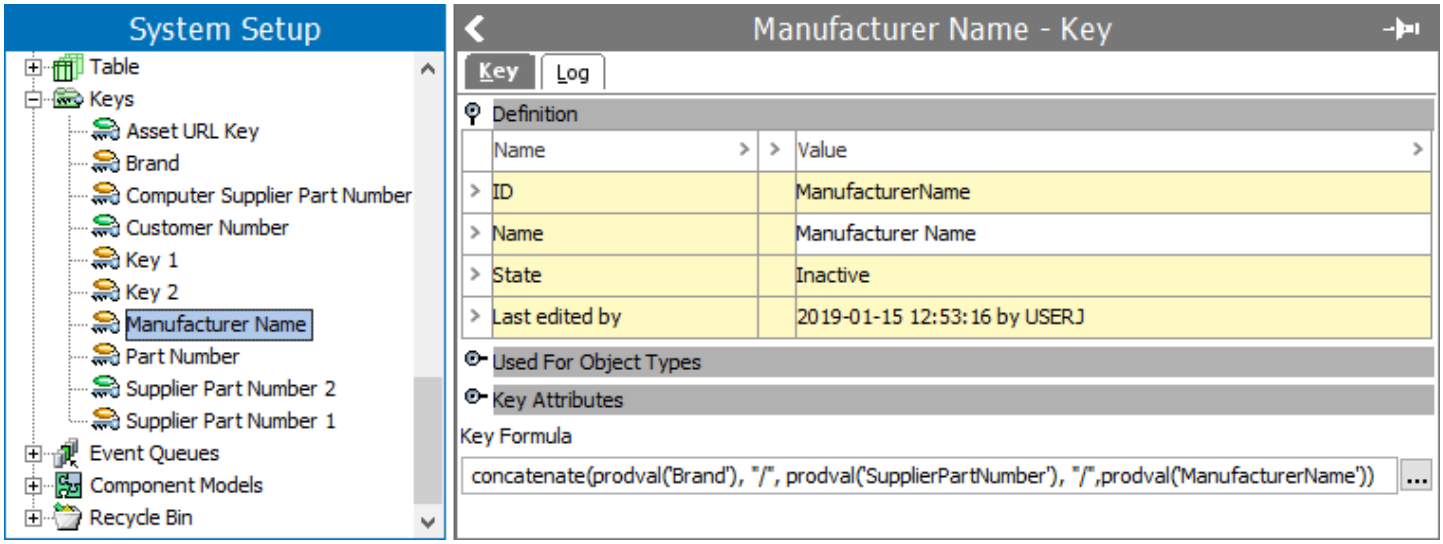
6. Right-click the hyperlink and select **Follow Hyperlink** to verify the hyperlink is configured correctly.



If the link is configured correctly, the URL is displayed. Otherwise, remove the hyperlink and repeat these steps.

Unique Keys

When a STEP ID is not available (such as with imported data) or when it is not an ideal unique identifier (i.e., external systems need to update data in 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. The terms 'key' and 'unique key' are used interchangeably and both indicate an identifier that cannot be duplicated.



The idea behind the STEP Key concept is that an external system can identify an object in STEP, not via its ID or URL, but by using one or more attribute values. Thus, a Key in STEP can be 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 have 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.

Note: To edit a key, it must be in the deactivated state. Keys that are in the activated state cannot be edited. For more information regarding activating or deactivating keys, see the topic **Activating and Deactivating Keys** in the **Unique Keys** 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 key.

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.
- must not be dimension dependent (values should be the same across all contexts).
- must 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 product object types have the following attributes values:

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.

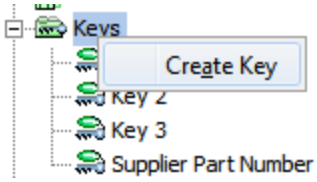
Prerequisites

It is expected that anyone creating a STEP Key is familiar with the **Defining Keys** section of the **Keys** topic within this guide.

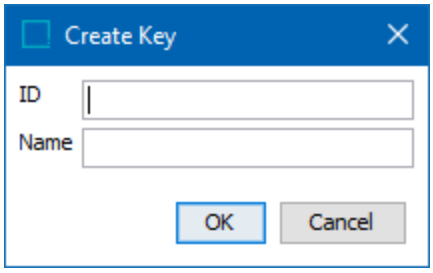
Create a Key

Below are steps to create a key.

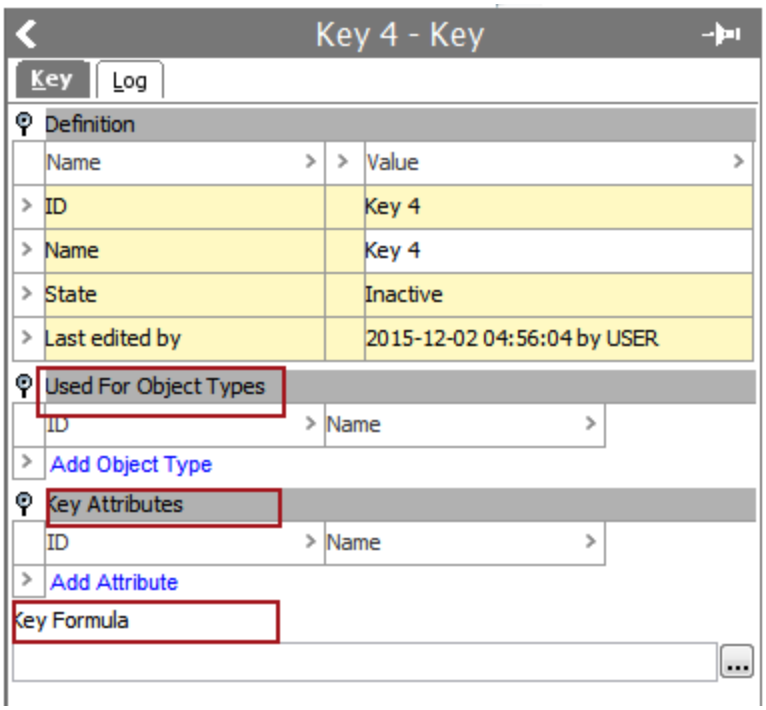
1. Go to **System Setup** > Right-click **Keys**, and then click **Create Key**.



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.



4. 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.

5. Expand the **Key Attributes flipper** > Click the **Add Attribute** link > Browse or search for the attributes to be used in the key formula.

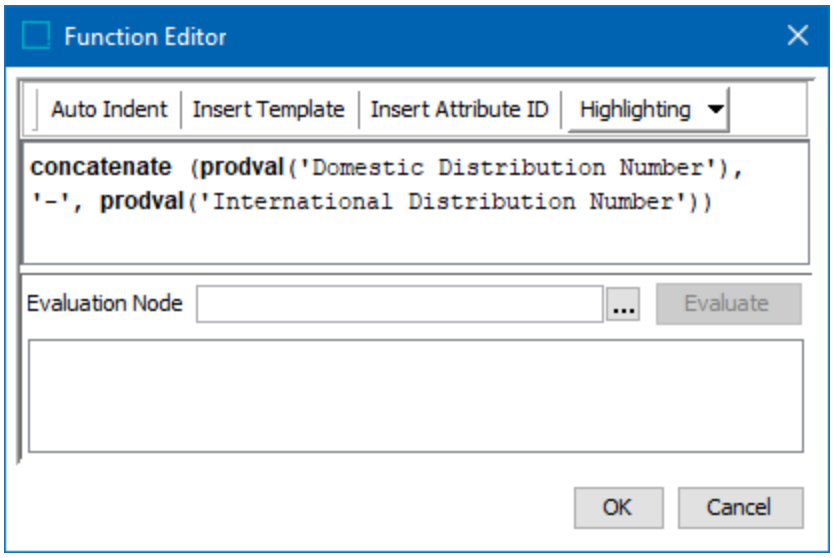
   Note: A number of restrictions apply to the attributes used for calculating the keys. For more information, see the **Defining Keys** section of the **Keys** topic within this guide.

6. 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 recommended 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 **Resource Materials** online help.

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.

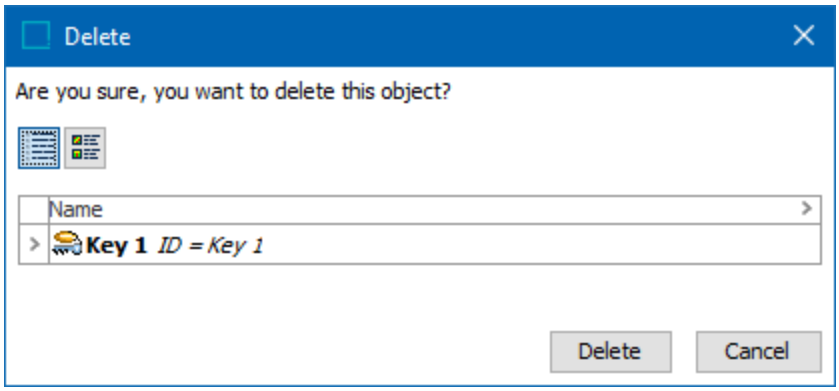
Optionally, before activating the key, use the Check Key function to confirm the setup of the STEP Key. For more information, see the **Checking Keys** topic within this guide.

Note: To edit a key, it must be in the deactivated state. Keys that are in the activated state cannot be edited. For more information regarding activating or deactivating keys, see the topic **Activating and Deactivating Keys** in the **Unique Keys** documentation.

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 APIs.

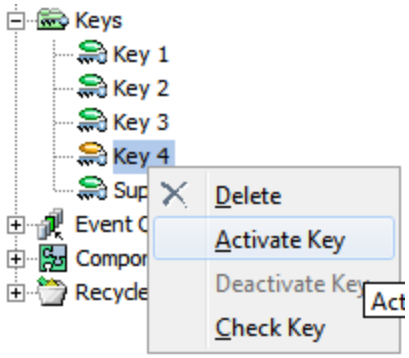
Prerequisites

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

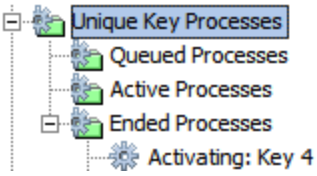
Below are steps to activate a key.

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.

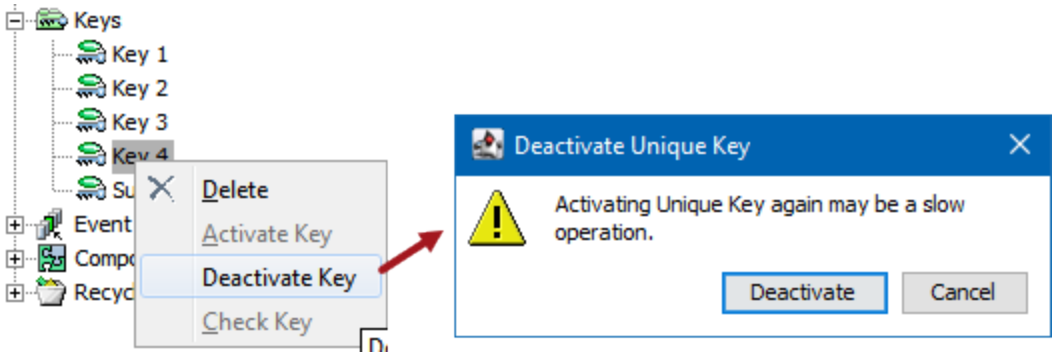


Keys are not generated for objects with missing key attribute values. Once values are supplied for all key attributes of an object, keys are generated on the fly.

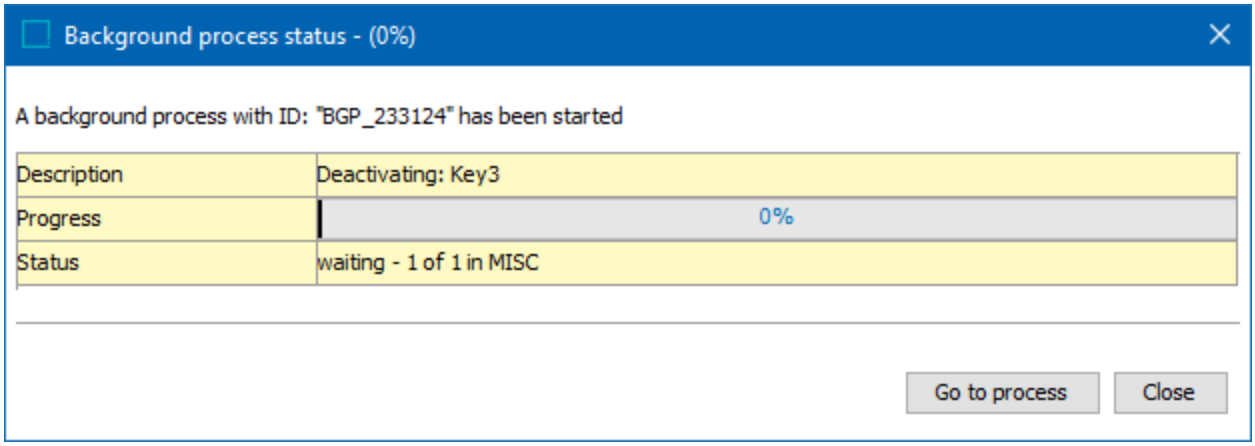
When a key has been activated, it is no longer possible to edit the values on the key attributes, unless you have set up special user privileges. However, you must deactivate the key first.

Deactivate a Key

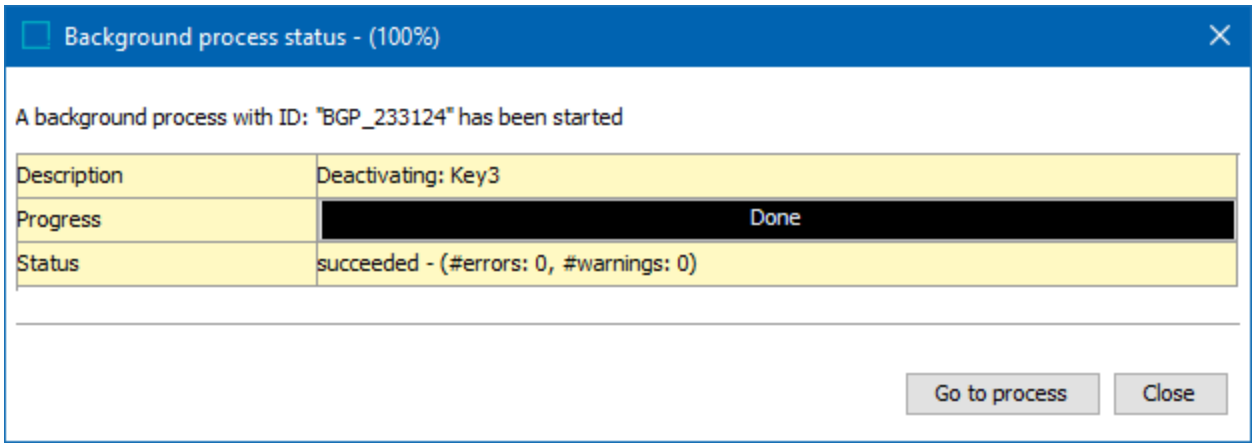
1. In **System Setup**, expand **Keys**, right-click **the relevant key**, select **Deactivate Key**, and the Deactivate Unique Key dialog will display.



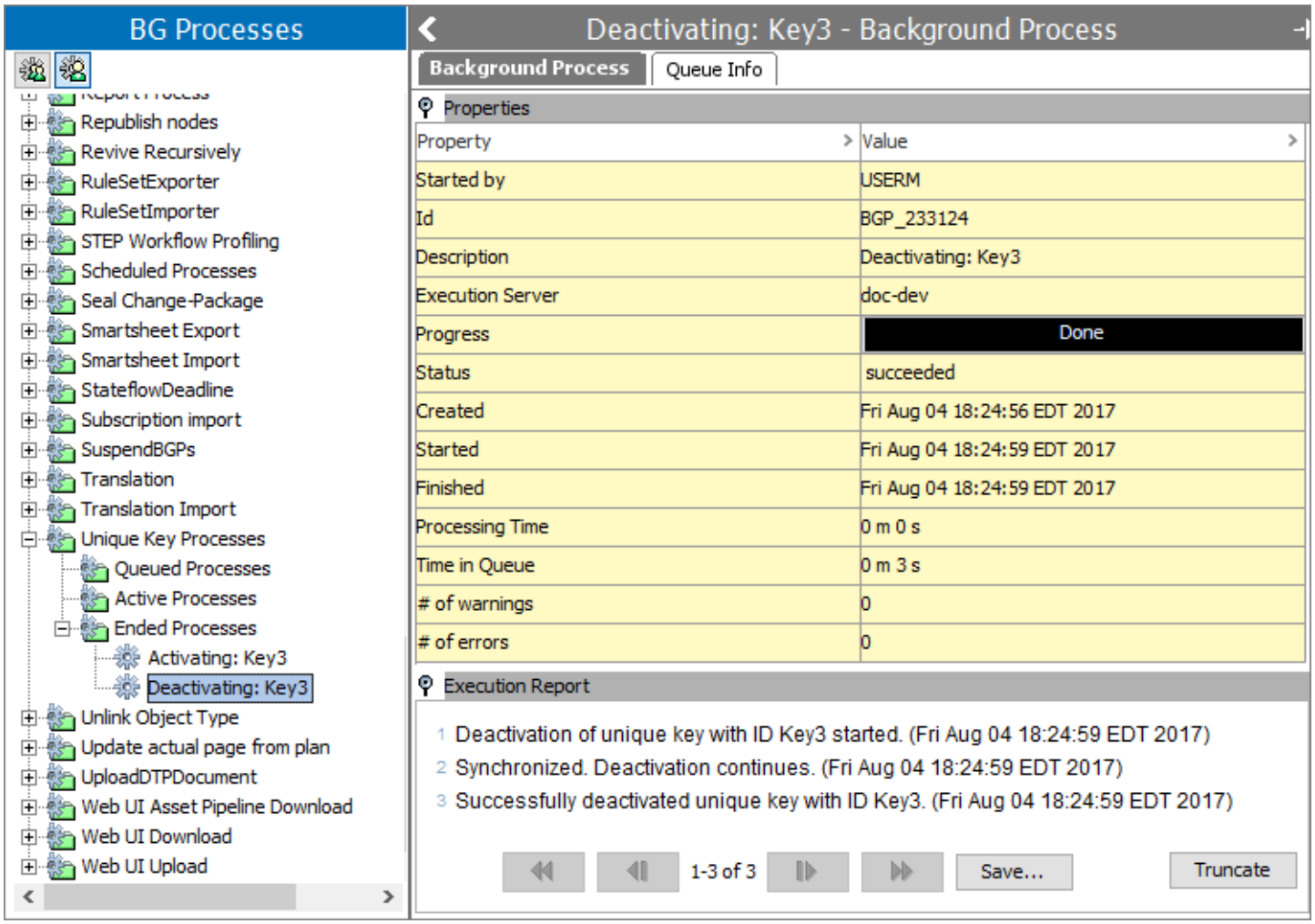
2. Click the **Deactivate** button, and the Background process status dialog will display. A BGP will start and attempt to deactivate the key.



3. Once the BGP has completed, the Background process status dialog will display as 'Done' (as shown below).



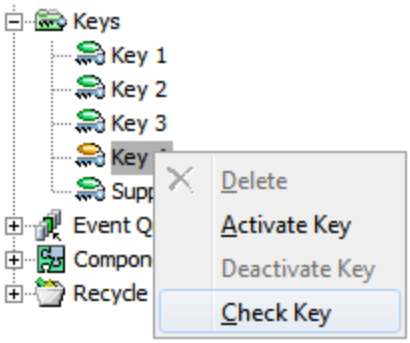
4. Optionally, click the **Go to process** button to view the Background Process Execution Report.



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:

- STEPXML
- Web Services APIs
- Web UI Simple Importer
- Workbench

STEPXML

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 STEPXML a special 'KeyValue' element is available but it is also possible to identify a Sales Item by providing the value for the EAN attribute. Both STEPXML snippets shown below would update the existing Sales Item with EAN 6725372937625.

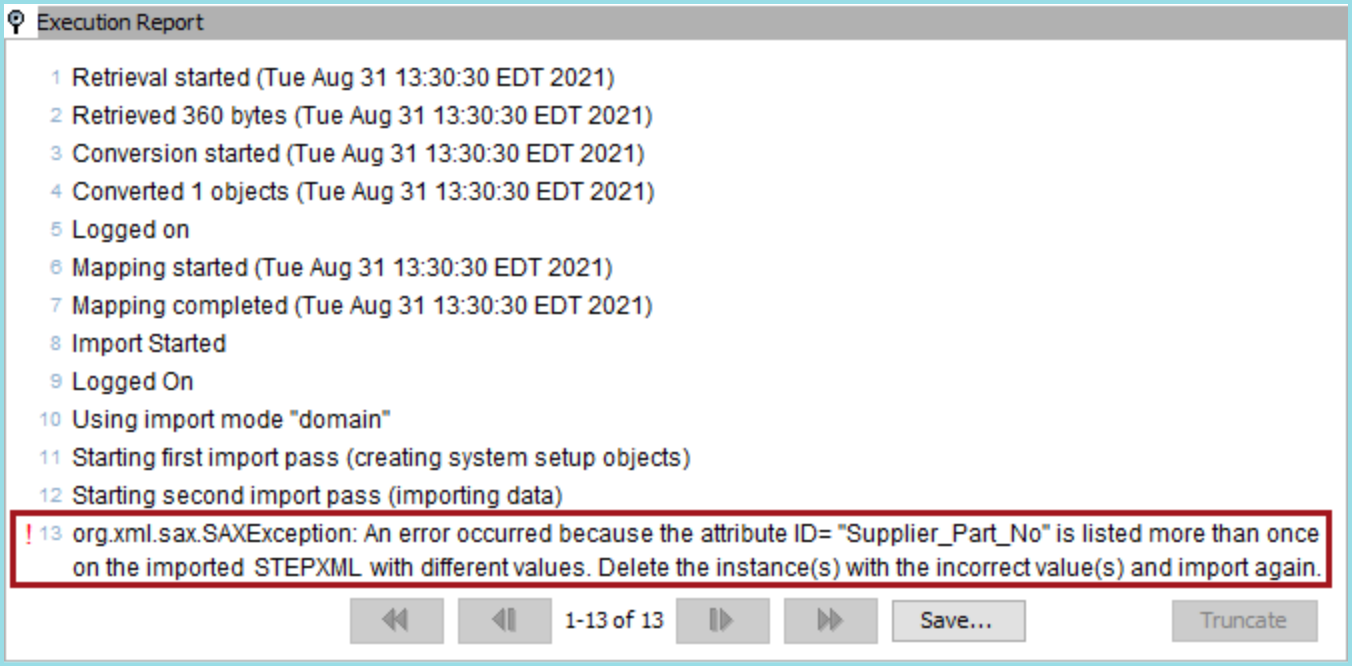
```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="GL" WorkspaceID="Main">
  <Products>
    <Product>
      <KeyValue KeyID="EAN">6725372937625</KeyValue>
      <Values>
        <Value AttributeID="HE Color">Black</Value>
      </Values>
    </Product>
  </Products>
</STEP-ProductInformation>
```

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="GL" WorkspaceID="Main">
  <Products>
    <Product>
      <Values>
```

```
6                <Value AttributeID="EAN">6725372937625</Value>
7                <Value AttributeID="HE Color">Black</Value>
8              </Values>
9            </Product>
10         </Products>
11       </STEP-ProductInformation>
```

Note: If no object with the specified key exists, 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 attribute? values take precedence.

When creating new objects via a STEPXML import file, if the same unique key value exists more than once, the system will ignore the duplicate unique key, skip object creation of the duplicate, and proceed with the import. Additionally, if different unique key values exist for the same object in a STEPXML file, no object is created and an informative error message displays letting the user know how to fix the problem before importing again.



For Generic XML and the tabular formats, STEP IDs can be left out and an object identified via values for key attributes. Because the 'Identify Destination' step of the Import Manager does not support Keys and objects, items 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:

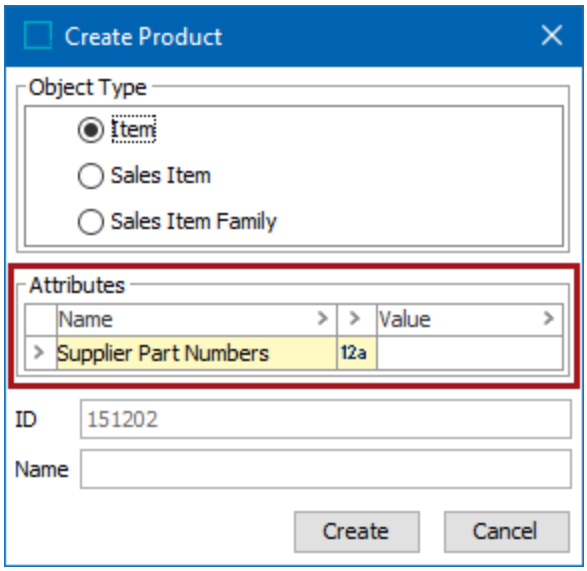
```
1        GET http://
         [host]/restapi/objectbykey/EAN/6725372937625?context=Context1&wor
         kspace=Main
```

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.



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.

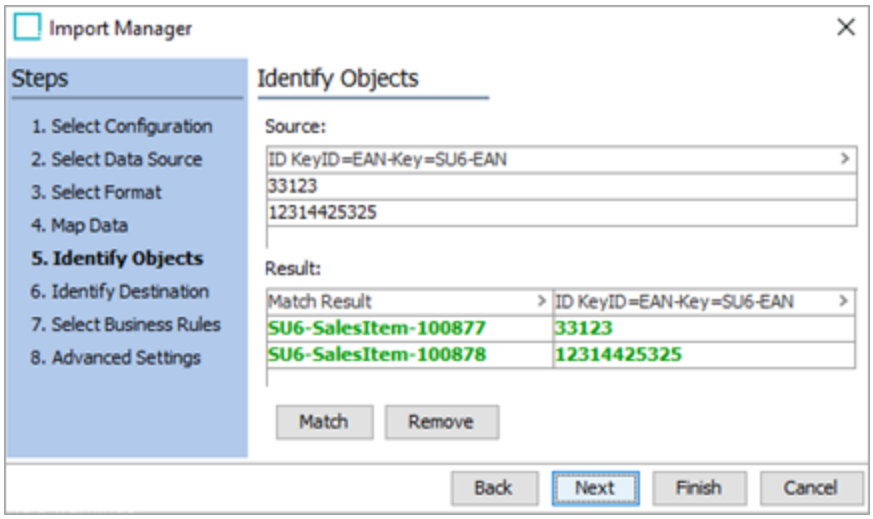
When a product of an object type for which an active keys exists is imported, a key is generated provided that all key attributes are populated. Any key conflicts are reported in the Execution report, and the new product will not be created.

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 in Excel, CSV, and XML files, provide the object type and values for all key attributes are included.

The Identify Objects step of the Import Manager wizard reports the matched objects with the respective STEP IDs in green (as shown below).



The following STEPXML shows the KeyValue tag as well as the individual values that are combined to build the key.

```
<STEP-ProductInformation ExportTime="2013-09-15 16:16:26"
ExportContext="GL" ContextID="GL" WorkspaceID="Main">
  <Products>
    <Product UserTypeID="KeyBProdType" ParentID="UniqueKeyImport">
      <KeyValue KeyID="KeyB">ValAvalBvalC</KeyValue>
      <Name>keyprodXMLImpChanged</Name>
      <Values>
        <Value AttributeID="UniqueKeyImportAttC">valC</Value>
        <Value AttributeID="ImportAtt">valA</Value>
        <Value AttributeID="UniqueKeyImportAttB">valB</Value>
        <Value AttributeID="UniqueKeyImportAttA">ValA</Value>
      </Values>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Deferred Cross References

It is possible to create objects during import using only a key value, and not using STEP IDs (e.g., product ID, classification ID, entity ID). It is also possible within the same STEPXML file to create a reference between new objects that do not yet exist in STEP. However, this scenario requires deferring cross reference creation.

Deferring cross reference creation allows the importer to create the new objects being referenced, while postponing creation of the references until the required objects exist. To avoid an error when importing a STEPXML file that contains new objects as well as new cross references, the following two criteria must be met:

- The data import must be done in Domain Mode, as defined in the **General Advanced Settings** section of the **Import Manager - Advanced Settings** topic within the **Data Exchange** documentation.
- The target object of the imported cross reference can be identified by only one active key, as defined in the **Unique Keys** topic.

Consider the following example XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<STEP-ProductInformation ContextID="GL" WorkspaceID="Main"
UseContextLocale="false">
  <Entities>
    <Entity UserTypeID="Company" ParentID="Division">
      <KeyValue KeyID="MainKey">1</KeyValue>
      <Name>Group 1</Name>
    </Entity>
    <Entity UserTypeID="Company" ParentID="Division">
      <KeyValue KeyID="MainKey">123456789</KeyValue>
      <Name>Group 2</Name>
      <EntityCrossReference Type="PrimaryCrossRef">
        <KeyValue KeyID="MainKey">93403</KeyValue>
      </EntityCrossReference>
    </Entity>
    <Entity UserTypeID="Company" ParentID="Division">
      <KeyValue KeyID="MainKey">93403</KeyValue>
      <Name>Group 3</Name>
      <EntityCrossReference Type="PrimaryCrossRef">
        <KeyValue KeyID="MainKey">1</KeyValue>
      </EntityCrossReference>
    </Entity>
  </Entities>
```

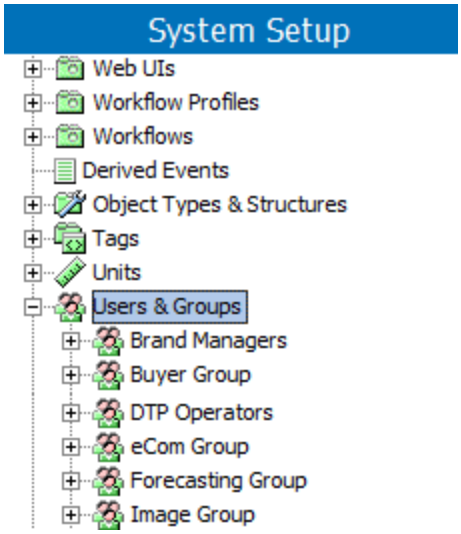
```
</STEP-ProductInformation>
```

In this example, the three entities are named Group 1, Group 2, and Group 3. Assume that none of these entities exist in STEP. Notice that Group 2 has a cross reference to Group 3 using a KeyValue, and that Group 3 has a cross reference to Group 1, also using a KeyValue.

Since none of the entities exist, the cross-references between these non-existing objects must be deferred.

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 recommended 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 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 controlled via 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

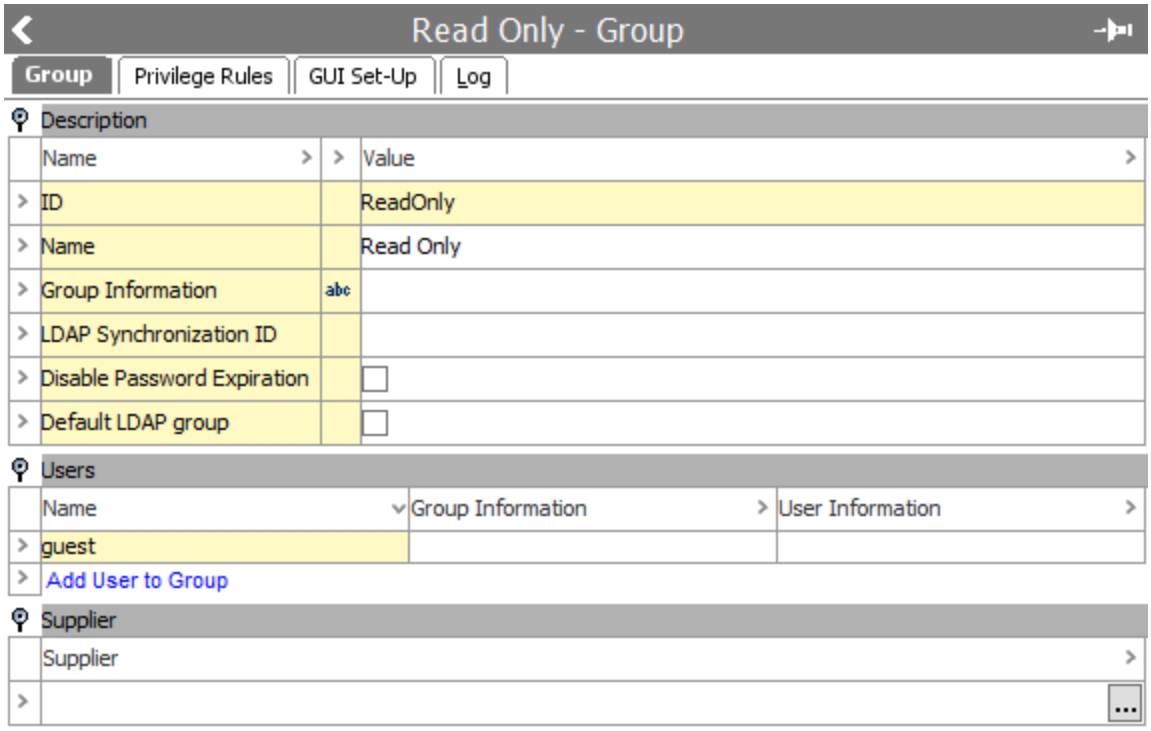
A member belonging to the **Super User** Group is typically an operator who in addition to data entry also is working with system setup and higher-level maintenance.

- 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

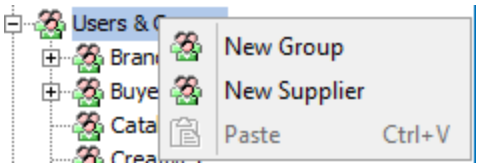
This section explains how to create and delete User Groups. While always containing the ID and Name attributes, the User Group tab will look different based upon the what system settings and attributes are configured and set up on your STEP system. For more information, see the **Privilege Rules** topic and the **GUI Setup** topic.



Creating a User Group

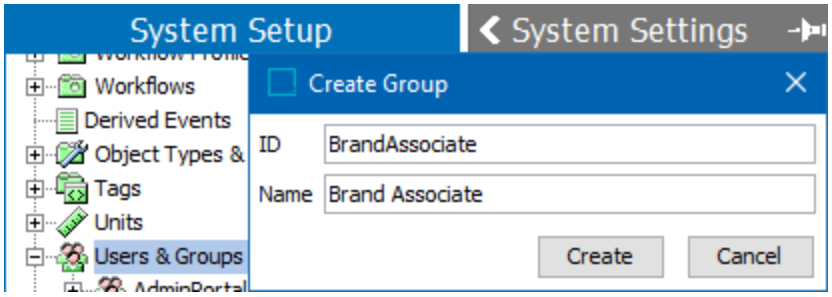
Groups are created in System Setup > **Users & Groups**.

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

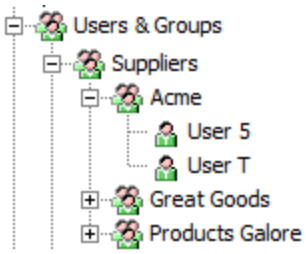
be defined for the group and Users added. For more information, see the **Privilege Rules** topic. For more information about creating a user, see the **Creating a User**.



Passwords can be configured to not expire for a User Group via the Group tab. If the Disable Password Expiration box is checked / enabled, the passwords for the users in that specific group will not expire. This setting overrides the password expiration on the global setting.

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.

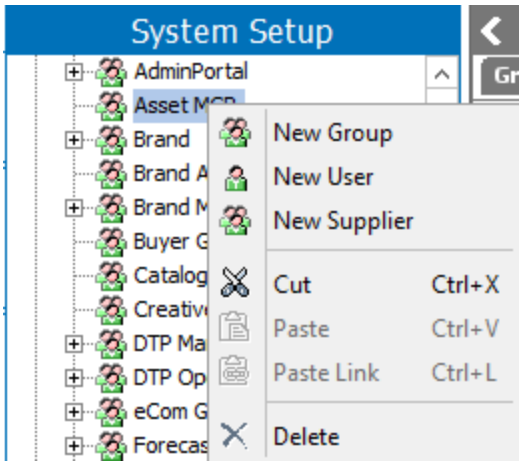


Note: A supplier user who is linked into multiple supplier groups can see all the supplier hierarchies they are a member of. However, supplier users cannot see any products or assets that are linked into multiple suppliers' hierarchies.

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.


System Setup
AdminPortal
Brand
New Group
New User
New Supplier
Cut Ctrl+X
Paste Ctrl+V
Paste Link Ctrl+L
Delete

Working with Users

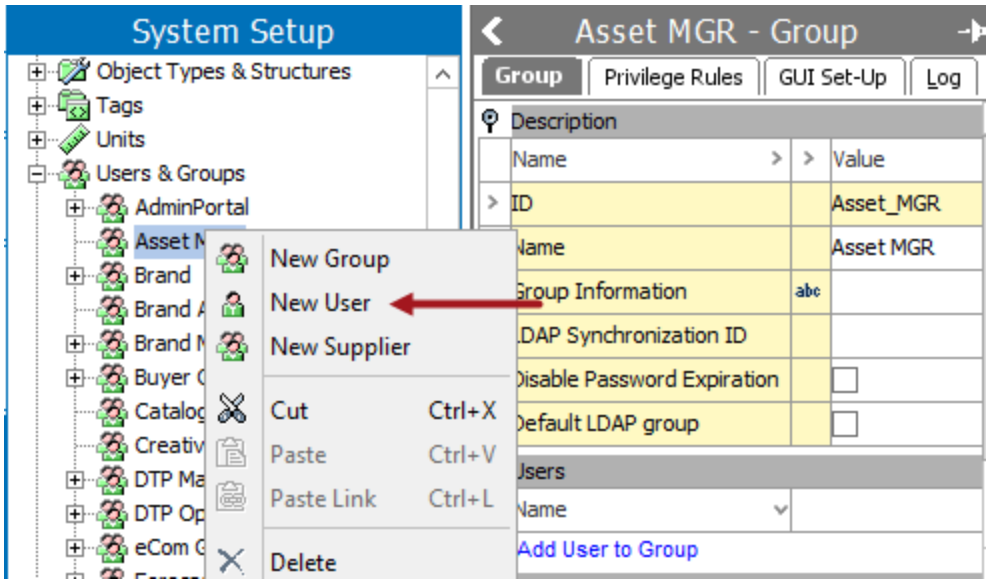
STEP can only be accessed by authenticated users. This section contains information needed to work with Users in Users and Groups from the STEP Workbench. Information detailed in this topic include:

- 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**.

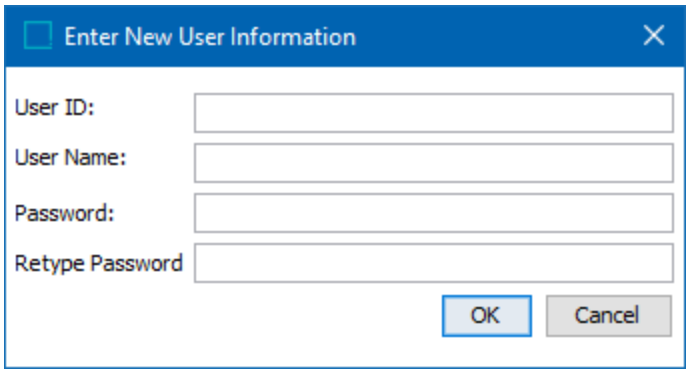
Creating a User

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**.

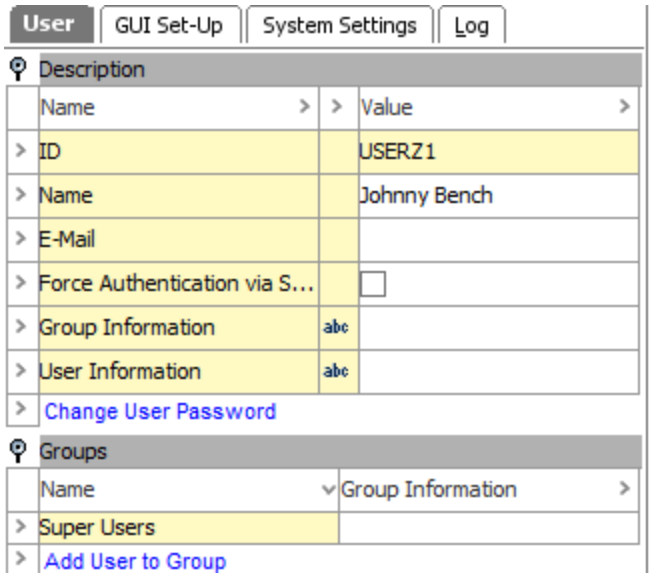


A dialog box displays and the following information must be completed for the user to be created:

- **User ID -** The ID that the user will use to sign into the STEP Workbench or Web UI. 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. Passwords may have up to 24 characters and are case sensitive.
- **Retype Password -** Retype the password just entered.



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.



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 Group Information and User Information shown in the example above.

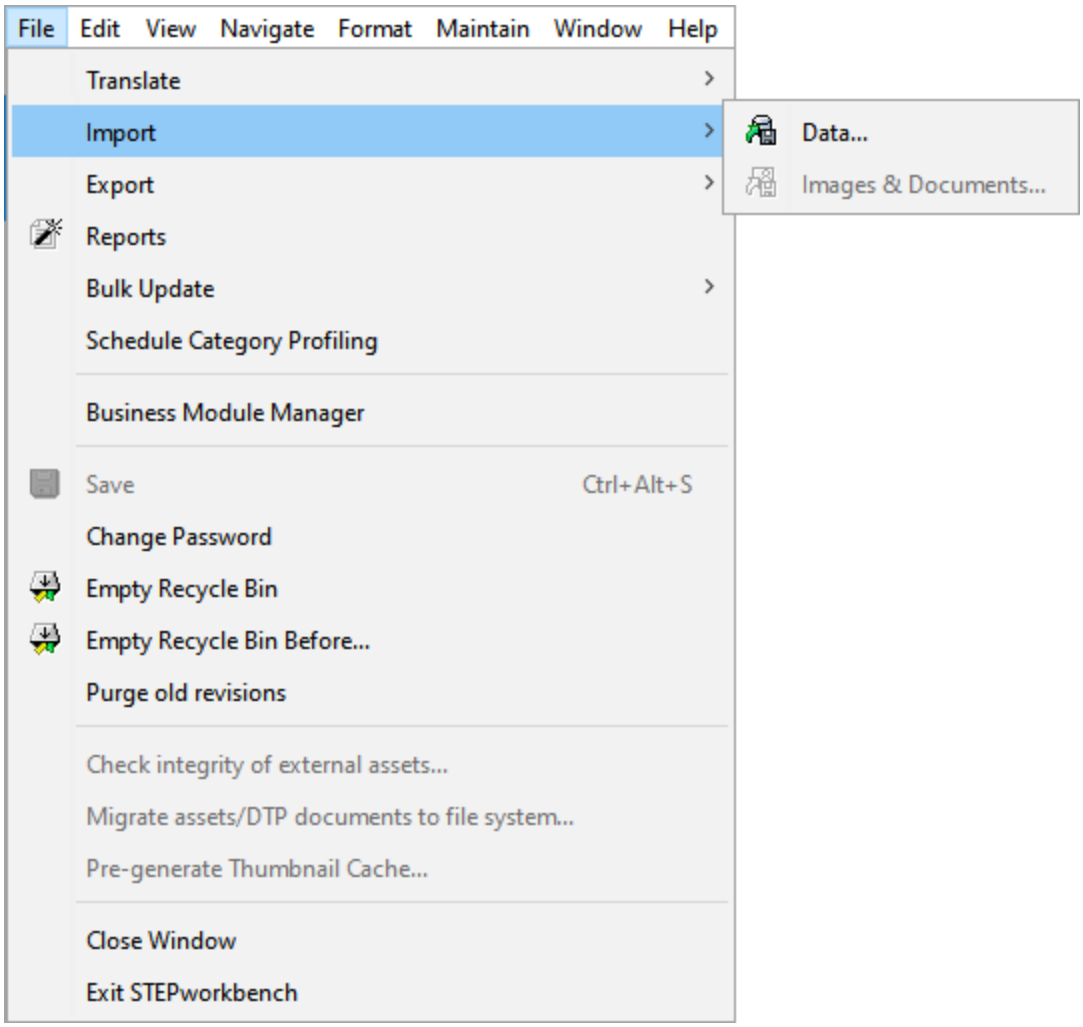
Bulk Adding New Users

With a STEPXML import, it is possible to create multiple new users at once in an instance of STEP. To create multiple users and the privilege groups required to enable those users, a STEPXML import file must be created that details this information. The STEPXML example below creates a user named 'AdminSteve' who is privileged with all setup and user actions for the Product hierarchy, Classification 1, and Entity hierarchy roots. This example can be used as a template for creating new users. Copy this template for each new user being created, replacing the existing values with the values relevant to the new user being created.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
  <STEP-ProductInformation ContextID="GL" WorkspaceID="Main">
    <UserGroupList>
      <UserGroup ID="AdminSteve" ReadOnly="false">
        <Name>AdminSteve</Name>
        <PrivilegeRule ActionSetID="All setup actions"/>
        <PrivilegeRule ActionSetID="All user actions" ProductID="Product hierarchy
root"/>
        <PrivilegeRule ActionSetID="All user actions"
ClassificationID="Classification 1 root"/>
        <PrivilegeRule ActionSetID="All user actions" EntityID="Entity hierarchy
root"/>
      </UserGroup>
    </UserGroupList>
    <UserList>
      <User ID="AdminSteve" ForceAuthentication="false" Password="Password"
EMail="AdminSteve@acme.com">
        <Name>AdminSteve</Name>
        <UserGroupLink UserGroupID="AdminSteve"/>
      </User>
    </UserList>
  </STEP-ProductInformation>
```

Once the STEPXMLhas been built with all required users, the file may be imported via the File > Import > **Data...** menu option.



For more information, see the **Creating a Data Import** topic in the **Data Exchange** documentation.

Adding a User to a Group

Users are added to groups in System Setup > **Users & Groups**.

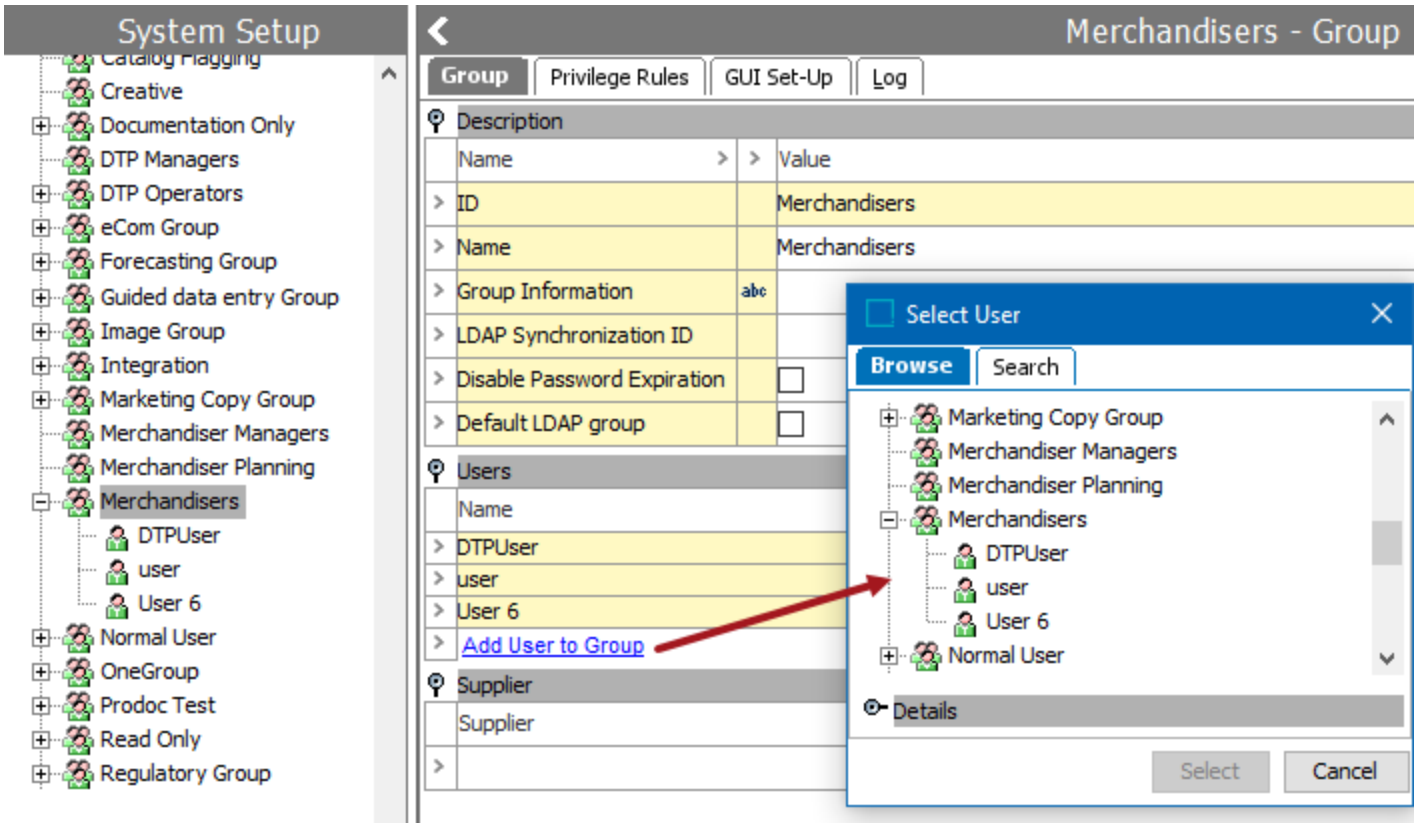
Adding a user to a group will provide the user with the privileges assigned to this group.

1. In System Setup, open **Users & Groups**, and then click the relevant group.

   The group appears in the group editor.

2. Under the Users flipper, click the **Add User to Group** link.

   A **Select User** dialog box displays.



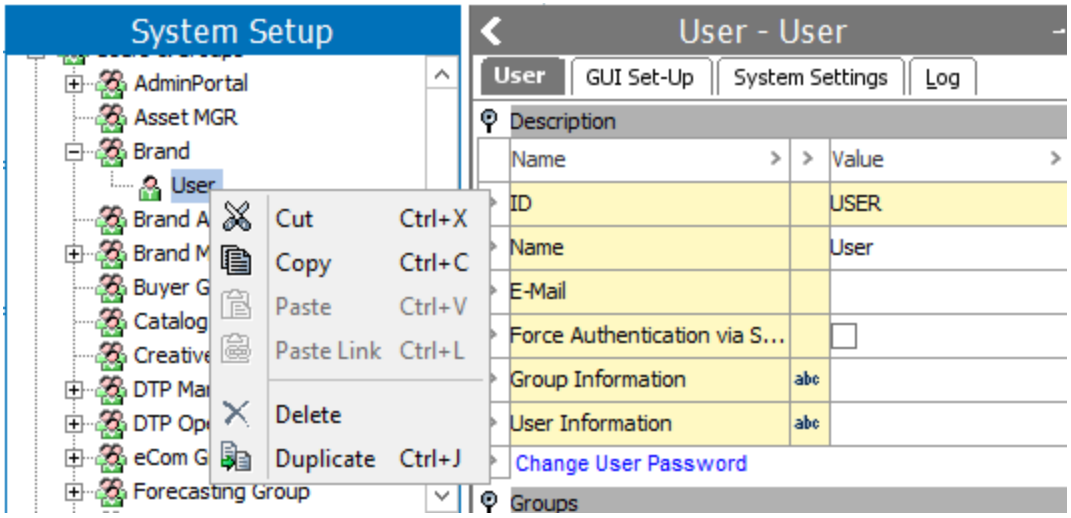
3. Select the relevant user, and then click **Select**.

   A user has now been added to the selected group, and the user appears under **Users**.

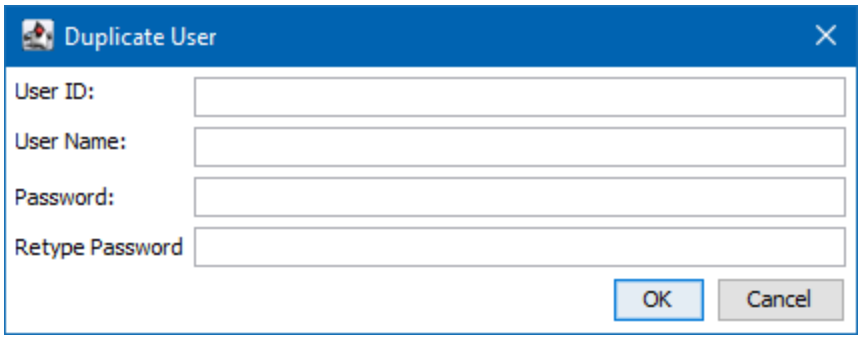
Note: A user can be a member of as many groups as needed, but this is not recommended practice as it makes it difficult to define what privileges a user actually has.

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, create the first user following the process in **Creating a User**, right-click the user you just created, and select 'Duplicate'.



A dialog displays as shown in the image below. Then 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.

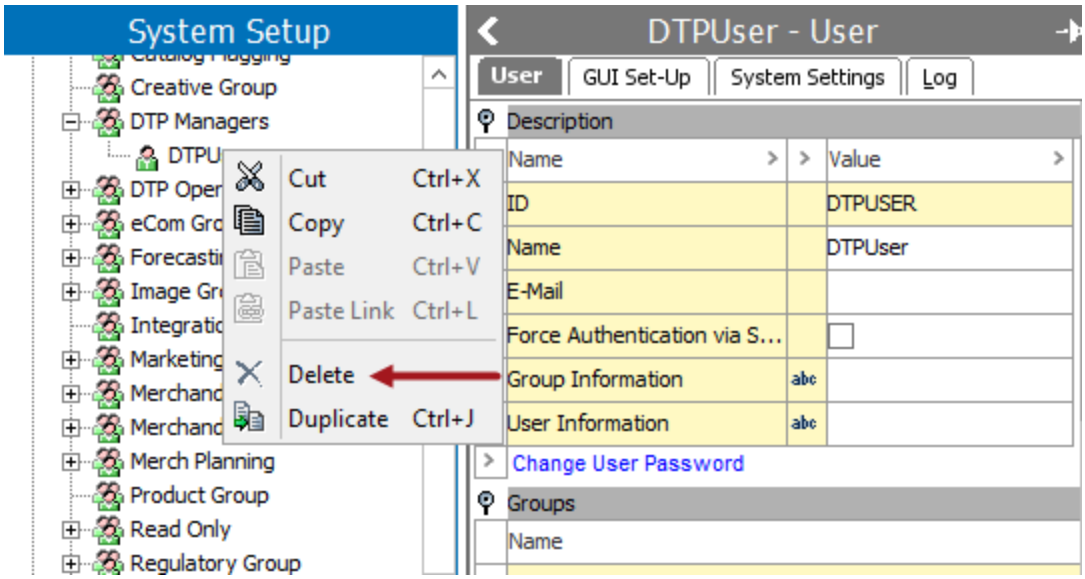


Important: Duplicating a user can be very effective, but if you are using metadata attributes on the duplicated user, the object type values on those will be duplicated as well. So for instance, if a user email address value is in the user being duplicated, that value will be carried over to the new user object and will need to be changed.

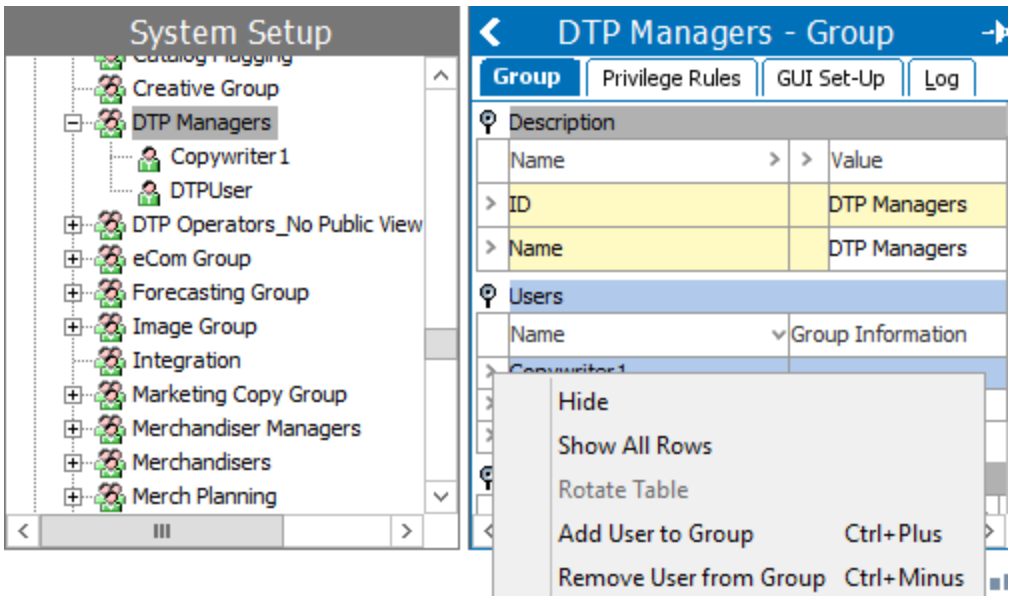
Removing a User from a Group

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 be deleted by finding the user in the group where the user resides, right-clicking on the user, and selecting **Delete** in the dialog. This will delete the user from all groups in the system.

There may be reasons a user cannot be deleted such as the user is an assignee within a workflow task or you are trying to delete your own user ID / name. If this is the case, you will be presented with additional information within the dialog that displays.

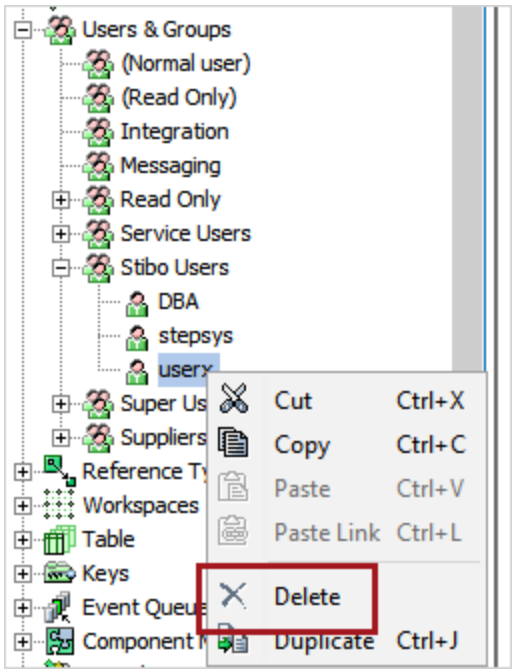


Sometimes there are circumstances where a user is a member of multiple groups but may need to be removed from one or more groups while remaining in another group or groups. In this case, as shown below, find the user group to remove the user from, under the Users flipper, right-click the arrow next to the User ID, and select **Remove User from Group**. If the user needs to be removed from multiple groups, repeat this process for each group.

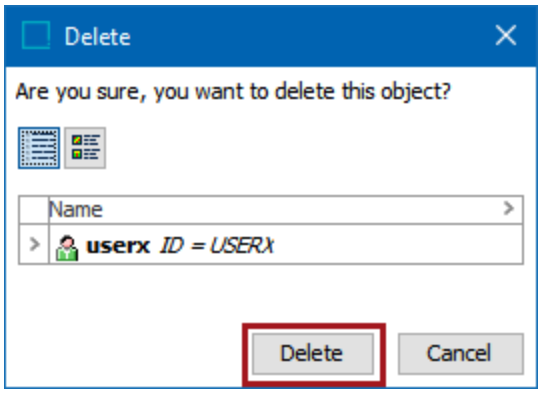


Deleting a User

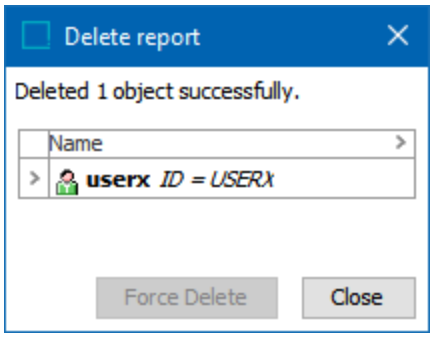
Occasionally, a user may need to be removed from accessing STEP. This action is done by way of right-clicking the desired user, and then selecting **Delete**.



A confirmation prompt will display. Select 'OK' to confirm deletion.



After a background process completes, the user will be deleted.



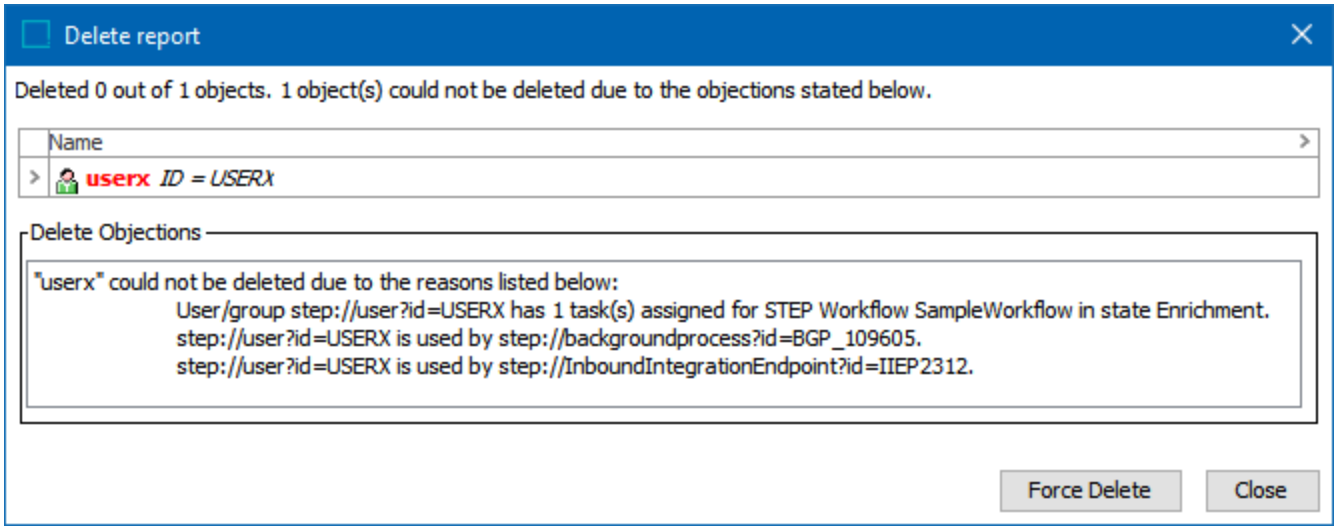
Considerations

If the user to be deleted has any outstanding actions, the user cannot be deleted.

Outstanding actions that will halt the deletion process include:

- scheduled background processes, either queued or running, that were started by the user
- inbound and outbound integration endpoints that are configured with the user in the Identify Endpoint step User parameter, or
- workflow states that have the user assigned to them.

Attempting to delete the user with responsibilities displays the following error showing where in STEP the user is assigned to tasks. Until the user is removed from these actions, they cannot be deleted, even with the 'Force Delete' button.



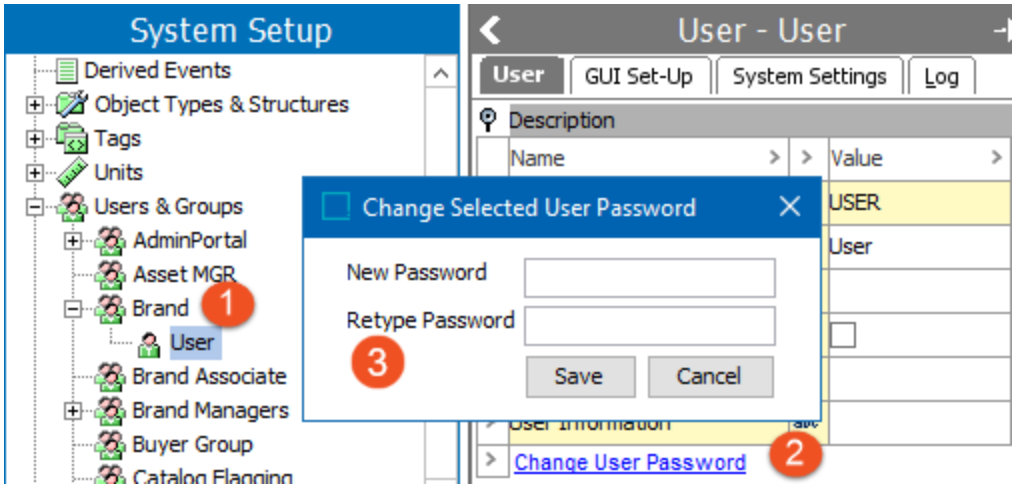
Changing a User Password

This section outlines the two most common scenarios in which user passwords are changed. In the first scenario, the user forgets their 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:

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, like asterisks (*) and percentages (%), are not allowed. Passwords are case sensitive. Only an administrative-level user should have the permissions to reset a user's password from the Users & Groups node. Options for users to change their own passwords are outlined below. For security reasons, the password is stored in the system as encrypted so that even an administrator cannot view it.

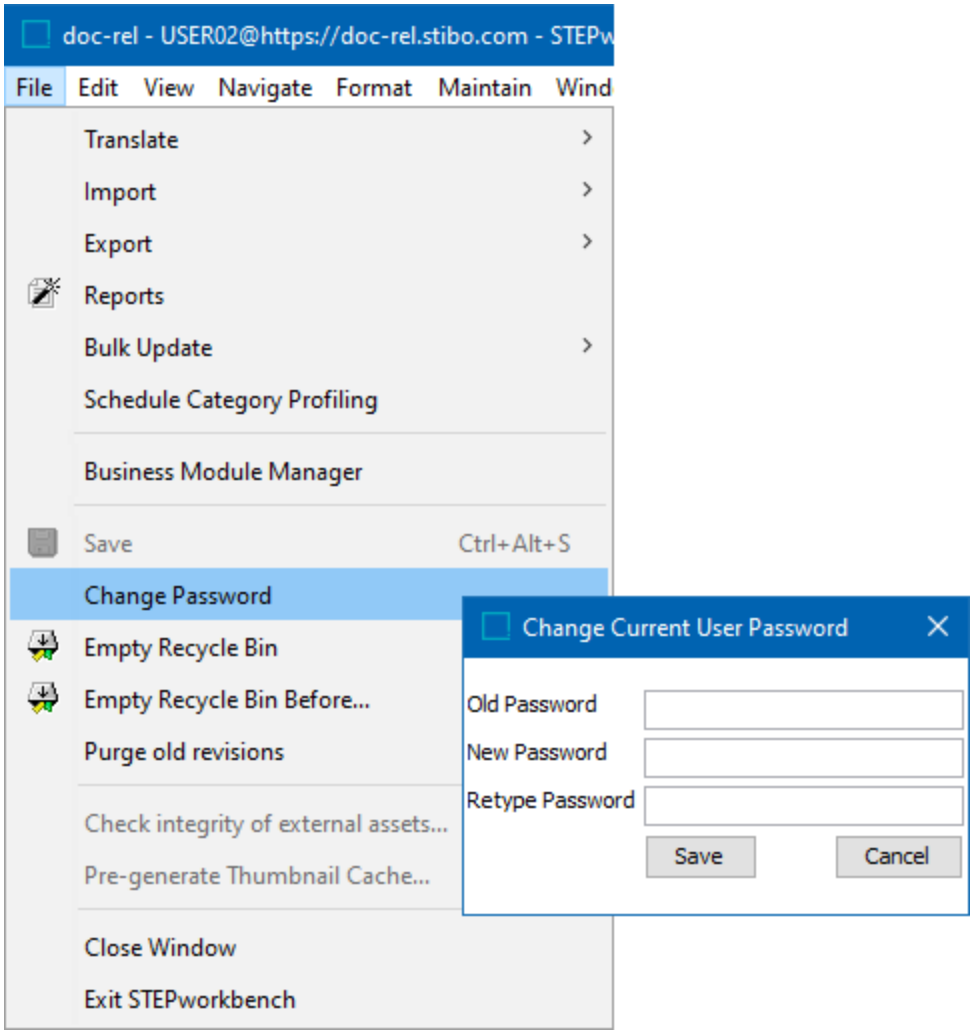
Note: On the Users & Groups root node, you can define a password security policy to be used. Information about setting security policies can be found in the **Security Policy** section of Users and Groups.

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.

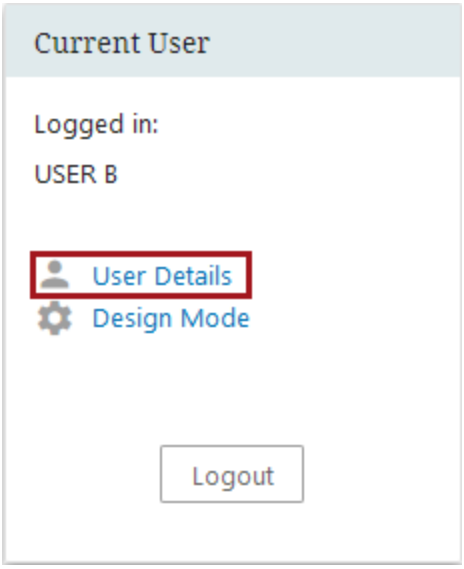
From the workbench

From the File menu, select **Change Password** and a **Change Current User Password** dialog box will display. 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.

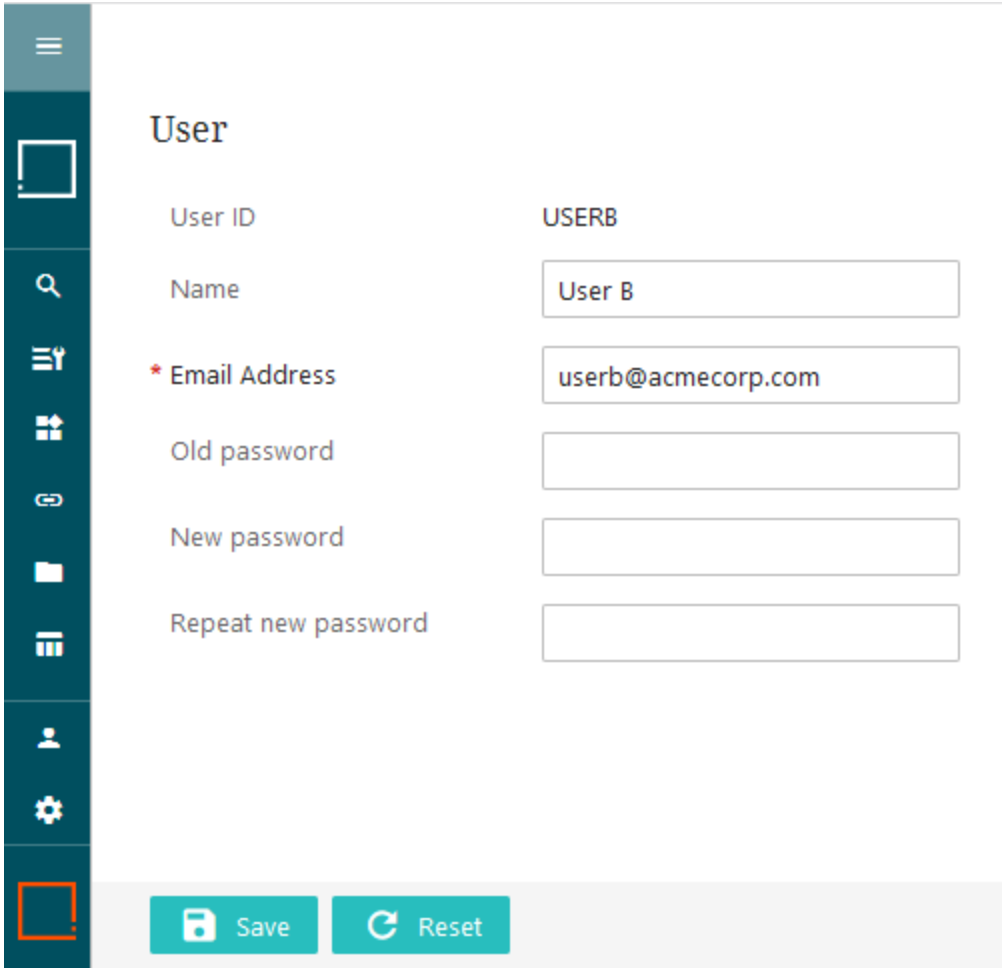


From the Web UI

Users can also change their passwords when logged into the Web UI via the User Screen. Access to a User Screen can be configured in different ways. For example, if a User Widget linked to a 'User Details' screen has been configured on the Homepage, then a user can click on the 'User Details' link (as shown in the screenshot below) to access the change password functionality.



If correctly configured, users will see the 'User Details' screen.



On the 'User Details' screen, users can take the following steps to change their password:

- In the 'Old password' field, enter the current password for the user.
- In the 'New password' field, enter the desired new password for the user.
- In the 'Repeat new password' field, enter the same password you entered in the 'New Password' field.
- Click 'Save' at the bottom of the screen. Your password has successfully been saved.

The new password will grant you access to Web UIs to which your user group has access, as well as the workbench.

To enable users to have the ability to change their passwords on the Web UI, the 'Force Authentication via STEP' setting must be checked for the specific user. This setting can be found on the user node in workbench. If this setting is left unchecked, the 'Old password,' 'New password,' and 'Repeat new password' fields will not display, and the user will be unable to change their passwords in the Web UI.



It is also important to note that if additional password security has been configured for passwords on your system, specifically 'Password Strength Validation,' any new passwords should conform to those standards.

For more information on establishing robust password security protocols, see the **Security Policy topic**, also on the **System Setup / Super User Guide** documentation.

For more information on accessing user information in the Web UI, see the **User Widget** and **User Details Screen** topics in the **Web User Interfaces** documentation. For more information on the User Details screen, see the **User Details Screen** topic, also in the **Web User Interfaces / Web UI Setup and User Guide** documentation.

User Anonymizer

User Anonymizer is intended to help companies meet their anonymization needs. This add-on component and its functionality are inspired by the 'right to be forgotten' (i.e., one aspect of the European General Data Protection Regulation (GDPR)). However, there may also be other reasons to anonymize a user as part of an established data clean-up routine / record management policy.

The User Anonymization process consists of replacing a deleted user's STEP ID with an anonymous ID. Once the User Anonymizer process has completed, you will see anonymization in three places in both workbench and Web UI:

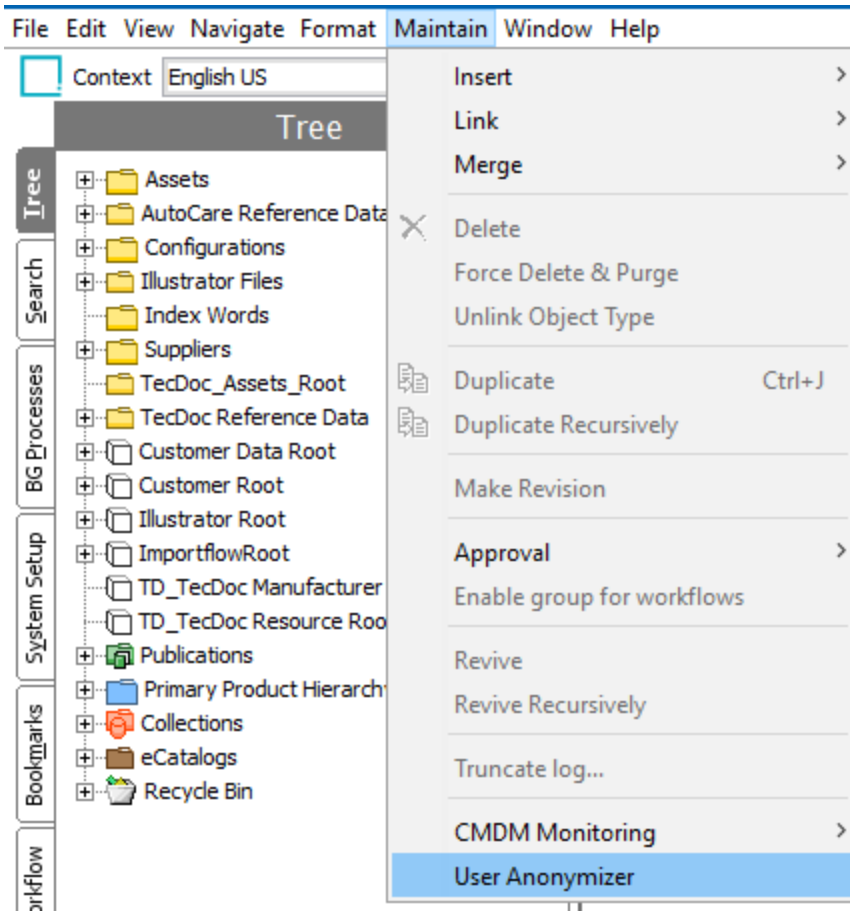
- Revision logs
- Workflow status logs
- Object change logs

Important: Before using this functionality, users should confirm that they are not breaking any retention policies or legal obligations regarding the maintenance of record history. Be aware that anonymizing log files goes against the recommendations of security standards like the OWASP Application Security Verification Standard. In general, Stibo Systems does not recommend using this functionality unless there are very vital business and/or legal reasons for doing so.

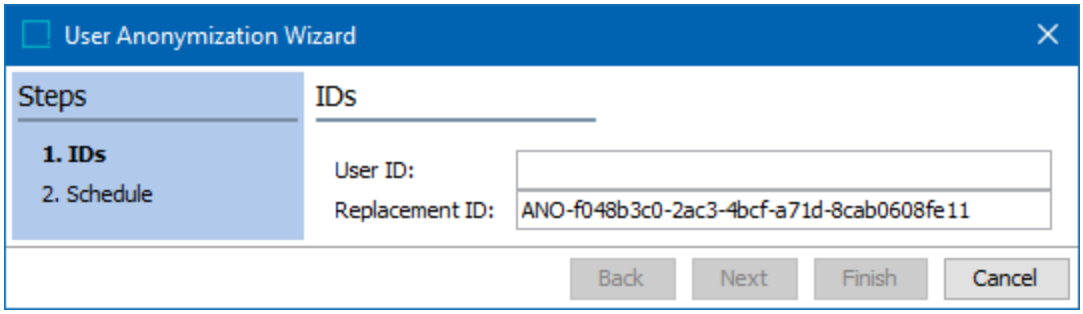
Starting the Anonymization Process

User Anonymizer is done via the workbench by admin users, and anonymizing can only be done to deleted users. This topic does not cover how to delete users. For more information on deleting a user, see the **Removing a User from a Group** section of the **Working with Users** topic.

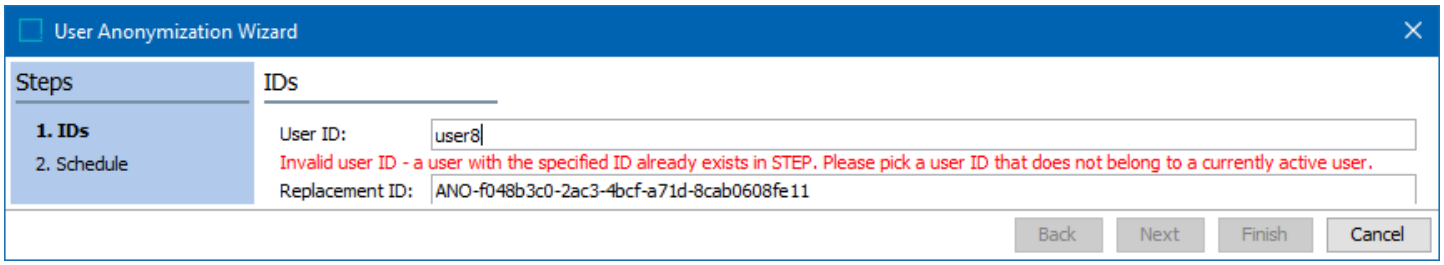
1. In workbench Tree, go to the Maintain menu and select User Anonymizer.



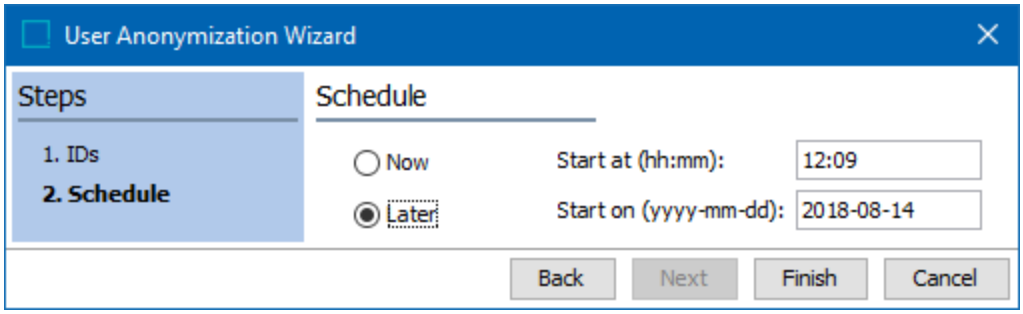
2. Enter the STEP ID of the deleted user to be anonymized. The ID is not case sensitive.



If you enter the ID of an active user, you will see the following message and cannot move ahead:



3. Click Next.
4. Select 'Later' to enter the date and time to run the process. It may be best for you to do the anonymization after regular business hours. Or, you may leave the default setting of 'Now' selected, and the process will run once you exit the User Anonymization Wizard.

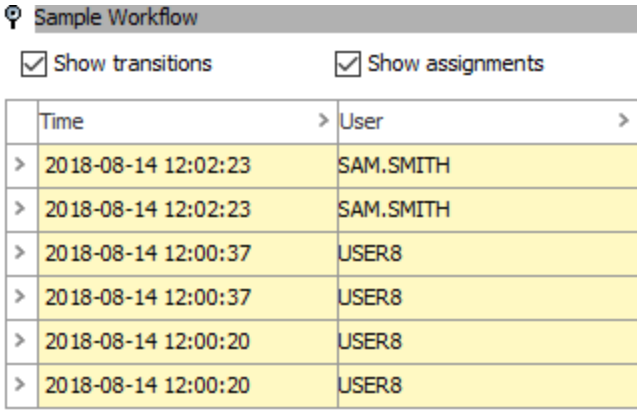


5. Click Finish.

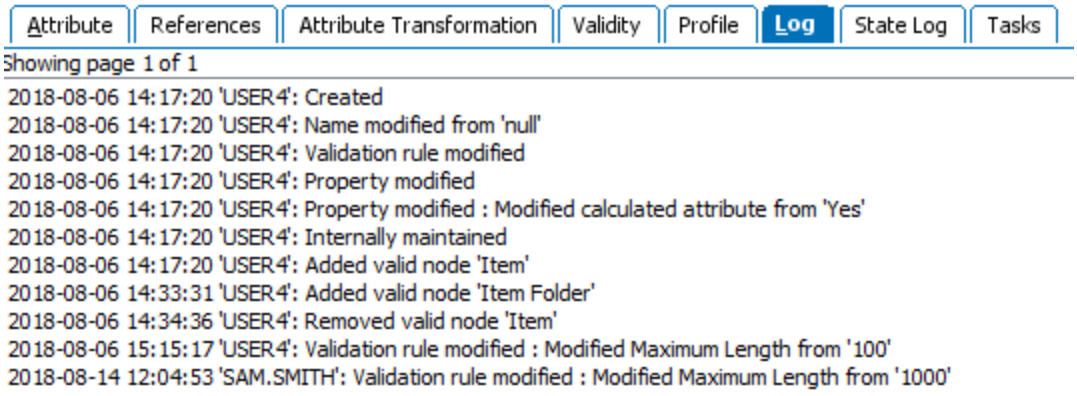
Example: Anonymizing a User

The following example follows user ‘Sam Smith’ as his ID is anonymized.

SAM.SMITH is the user identified in a number logs. For example:

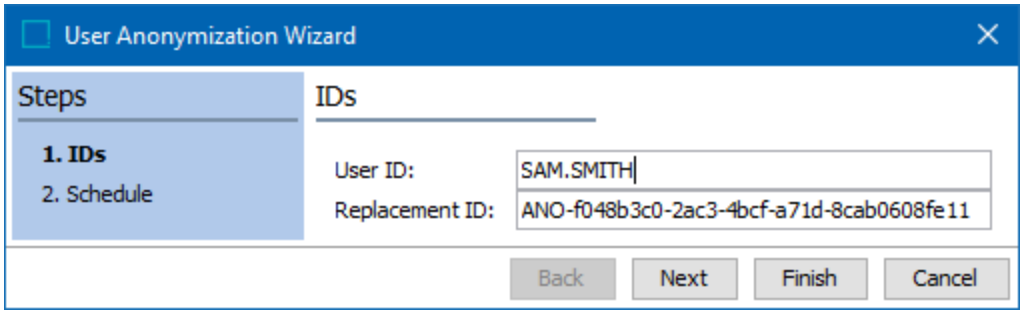


Time	User
2018-08-14 12:02:23	SAM.SMITH
2018-08-14 12:02:23	SAM.SMITH
2018-08-14 12:00:37	USER8
2018-08-14 12:00:37	USER8
2018-08-14 12:00:20	USER8
2018-08-14 12:00:20	USER8

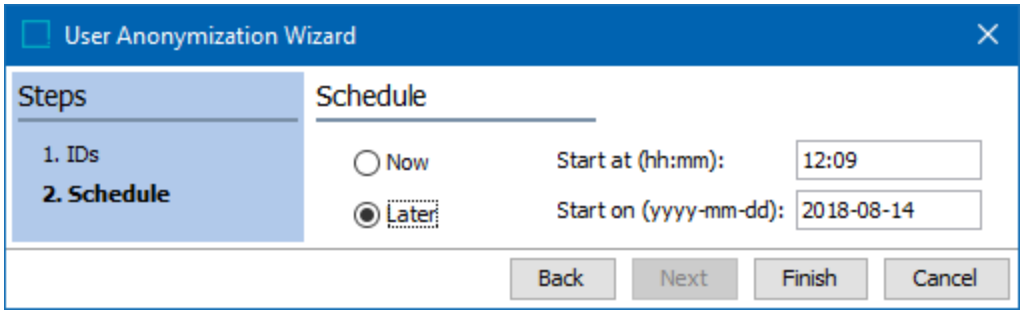


Sam Smith's User ID (SAM.SMITH) is deleted since Sam moved to a new position and no longer accesses a STEP system. It is not vital to the business, nor is there a legal reason, to keep these records.

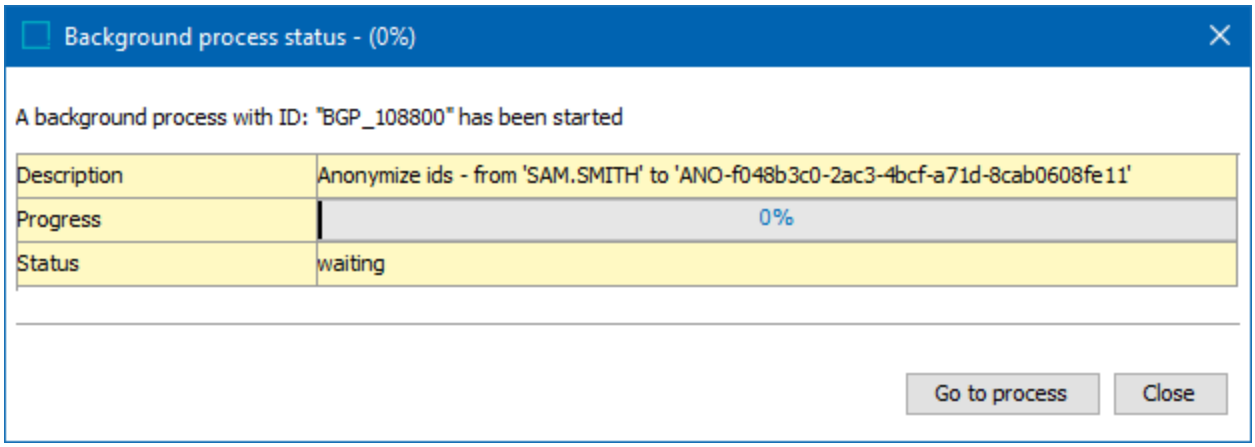
The admin user running the User Anonymizer process enters 'SAM.SMITH' in the ID field and clicks Next.

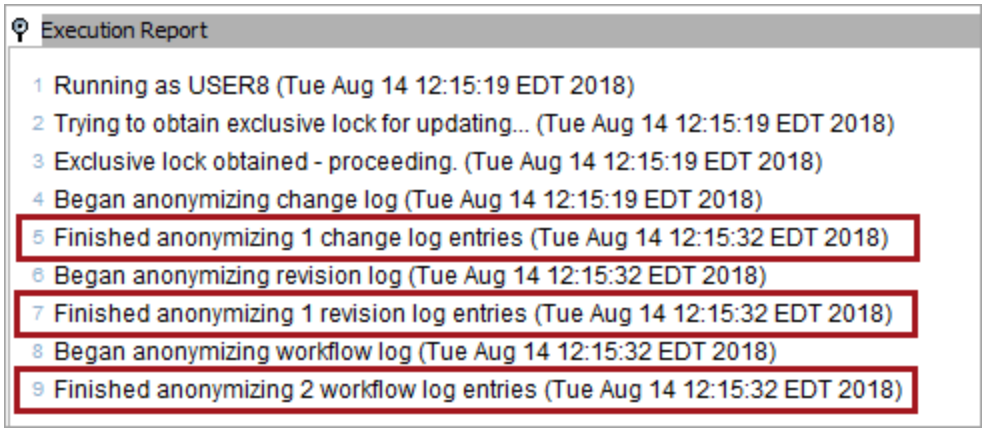


Sam has been at the company for years, Because there are extensive records to anonymize, the process is scheduled for an off time so that there is no business impact as the process runs.



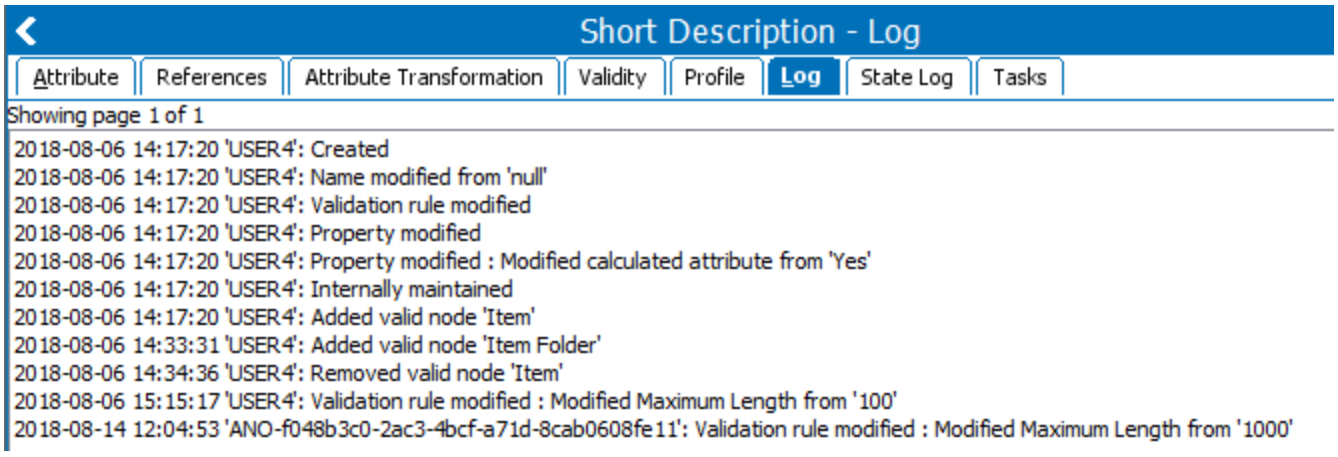
A background process begins, and an Execution Report shows when the anonymization is complete.



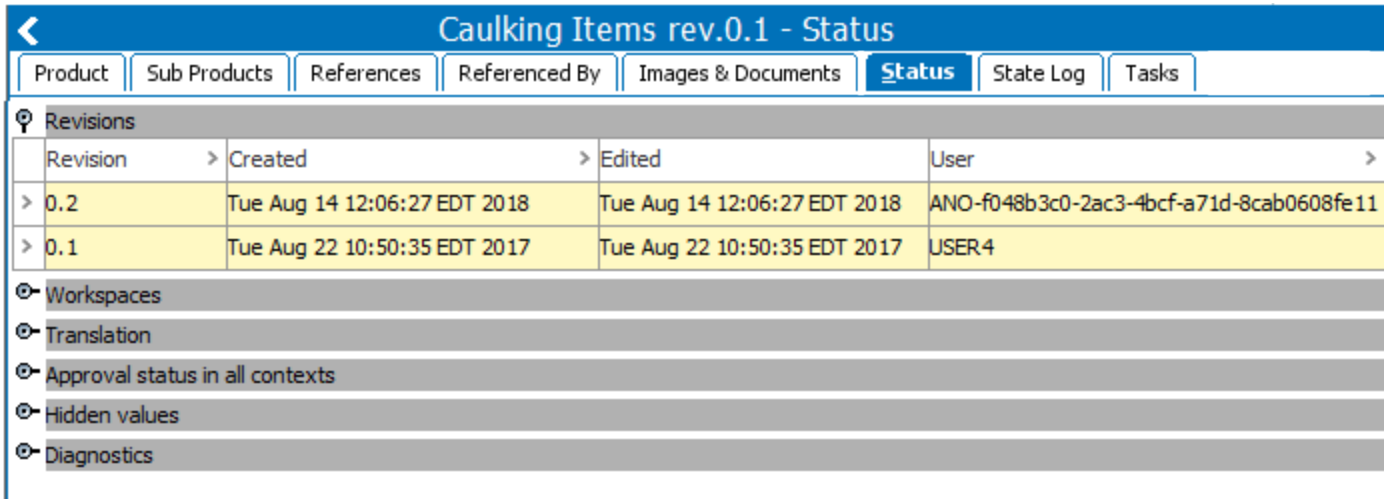


The new anonymized ID shows in place of Sam's ID in the following places:

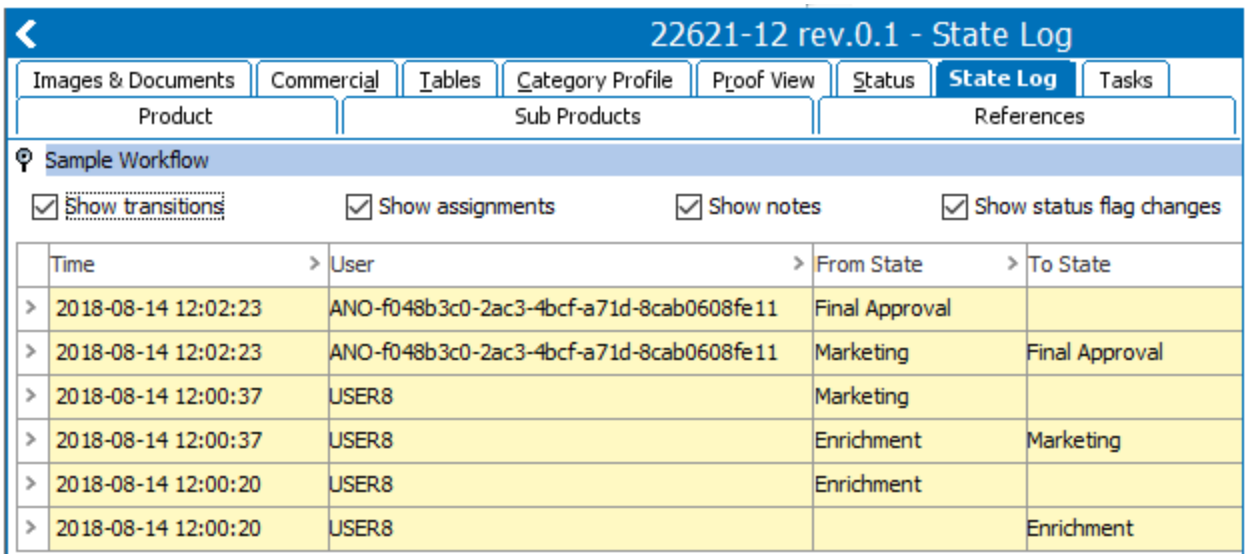
- **[Object] Change Log** (in this example, it is the change log for the 'Short Description' attribute)



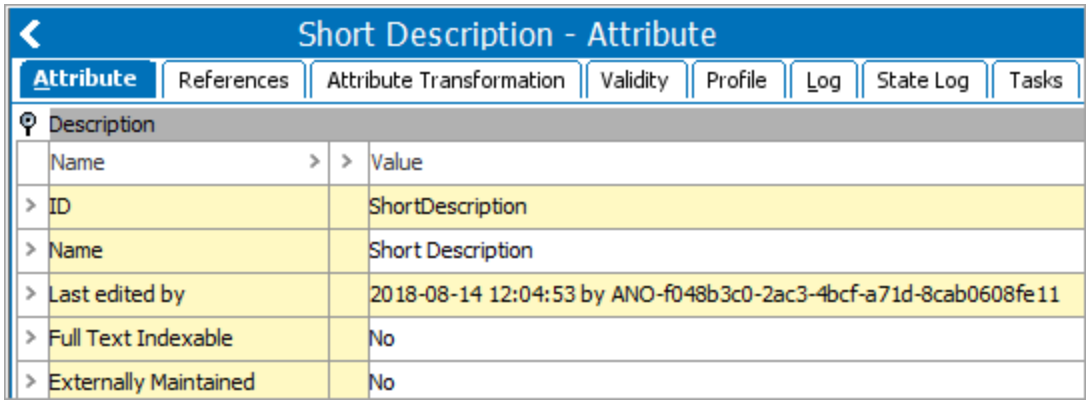
- **Revision Log**



- **Workflow [State] Log**



Additionally, the user ID is also anonymized within an object's 'Last edited by' aspect.



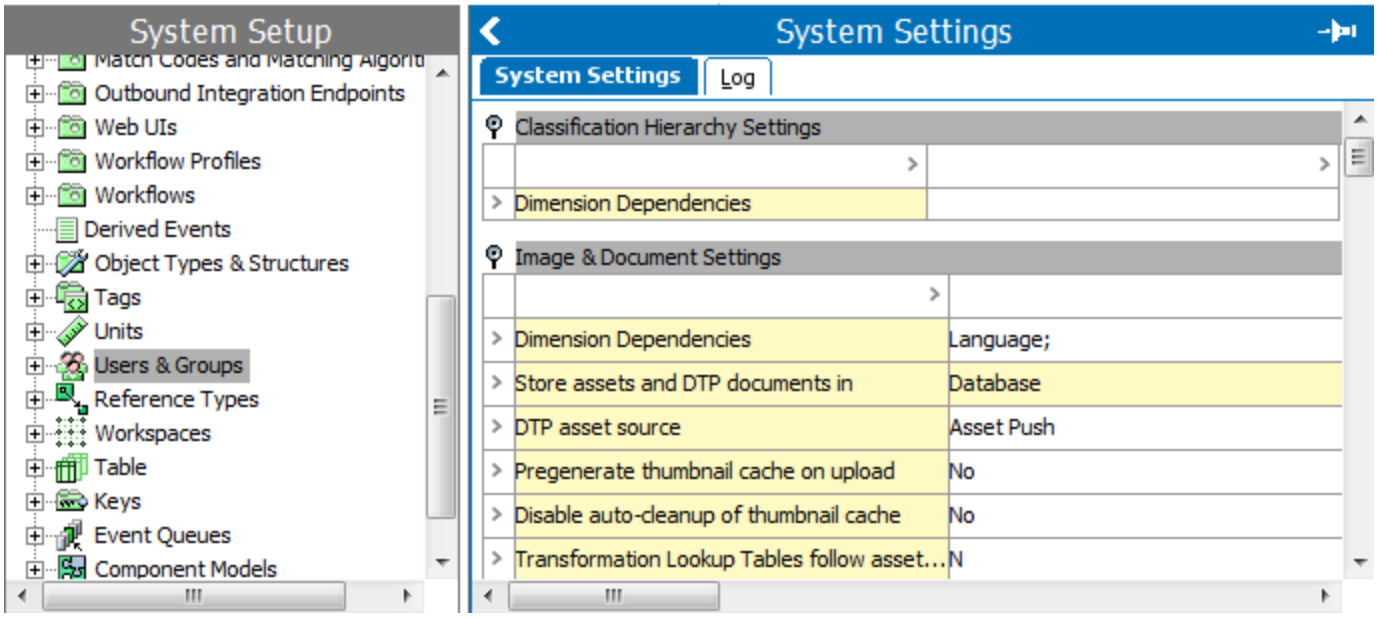
While the examples in this topic are all showing the User Anonymizer end results in workbench, the anonymization is visible in Web UI, where configured and applicable (as a starting point, see the **Web User Interfaces** documentation). For example, revision information on the Multi Revision screen:

	Name	ID	Brand	Revision
0.2 (CURRENT)	Caulking Items	8242	Products Galore	0.2 Created by ANO-f048b3c0-2ac3-4bcf-a71d-8cab0608fe11 on Tue Aug 14 12:06:27 EDT 2018
0.1	Caulking Items	8242		0.1 Created by USER4 on Tue Aug 22 10:50:35 EDT 2017

System Settings

Global system settings are maintained in the **System Setup** tab, under the **Users & Groups** root node.

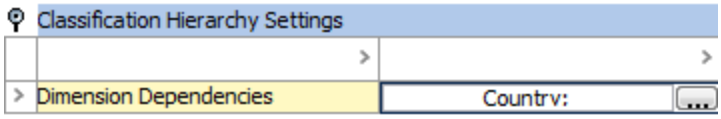
In System Setup, click **Users & Groups**. A **System Settings** editor appears. It lists a number of general and global settings on the system. Each of these settings will be described throughout this section.



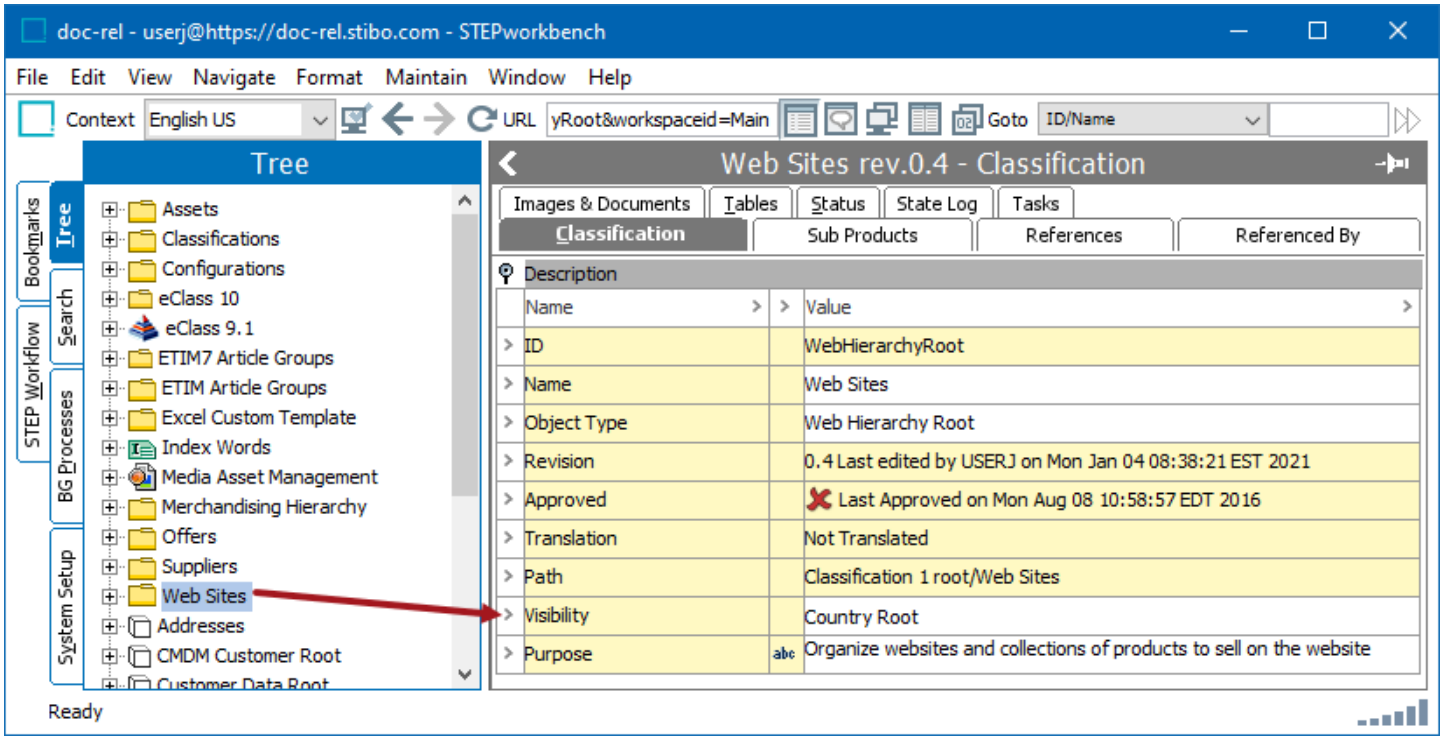
Classification Hierarchy Settings

Dimension Dependencies

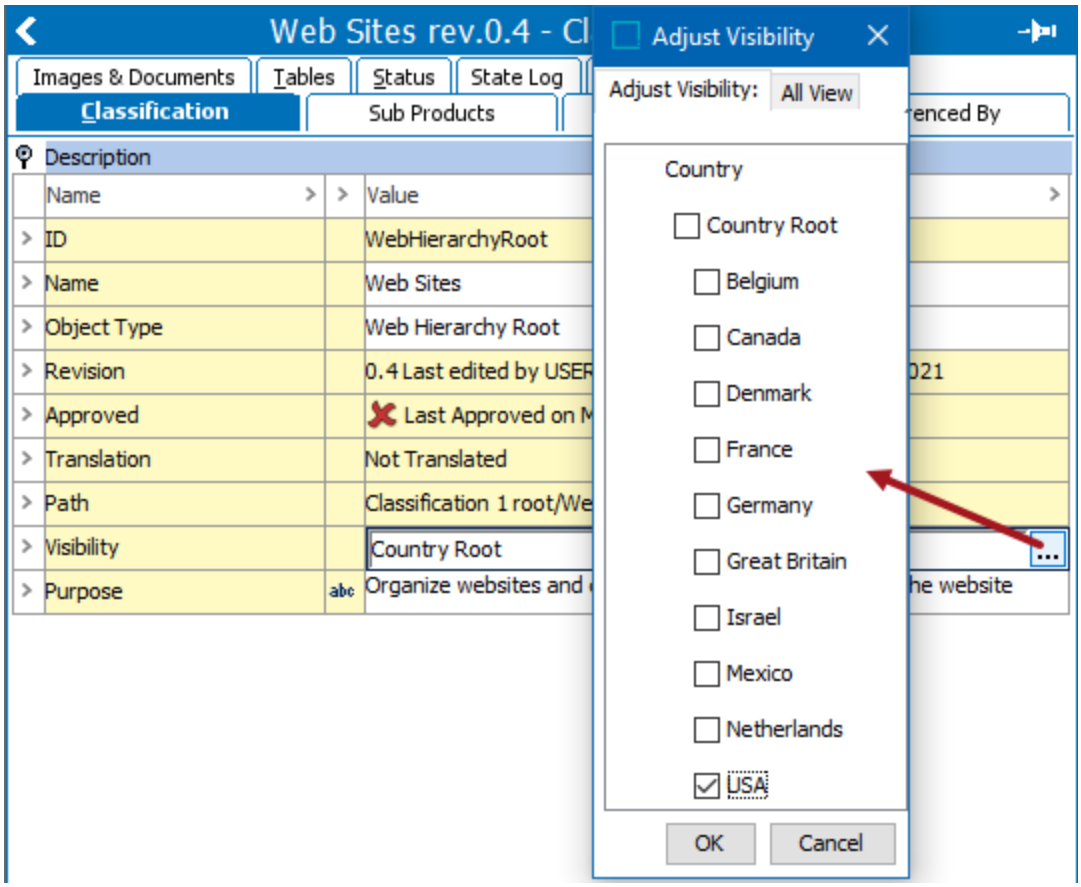
This option allows for **folders** in classification structures to be dimension-dependent. This is different than the dimension dependencies that are set on the object type. Selecting dimension dependencies will determine whether classification structure will or will not be shown in specific languages or countries. In the example below, the dimension dependency has been set for country by clicking on the ellipsis **[...]** button and selecting Country.



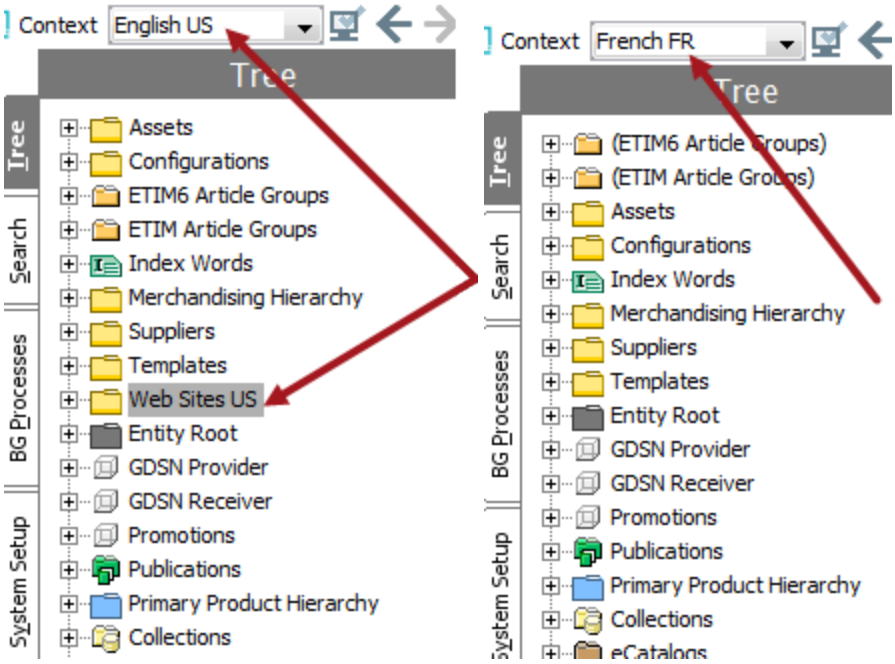
Selecting this dimension dependency added the Visibility metadata attribute to the classification objects in Tree.



Click within the Visibility row, and then click the ellipsis button (...) that is available 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.

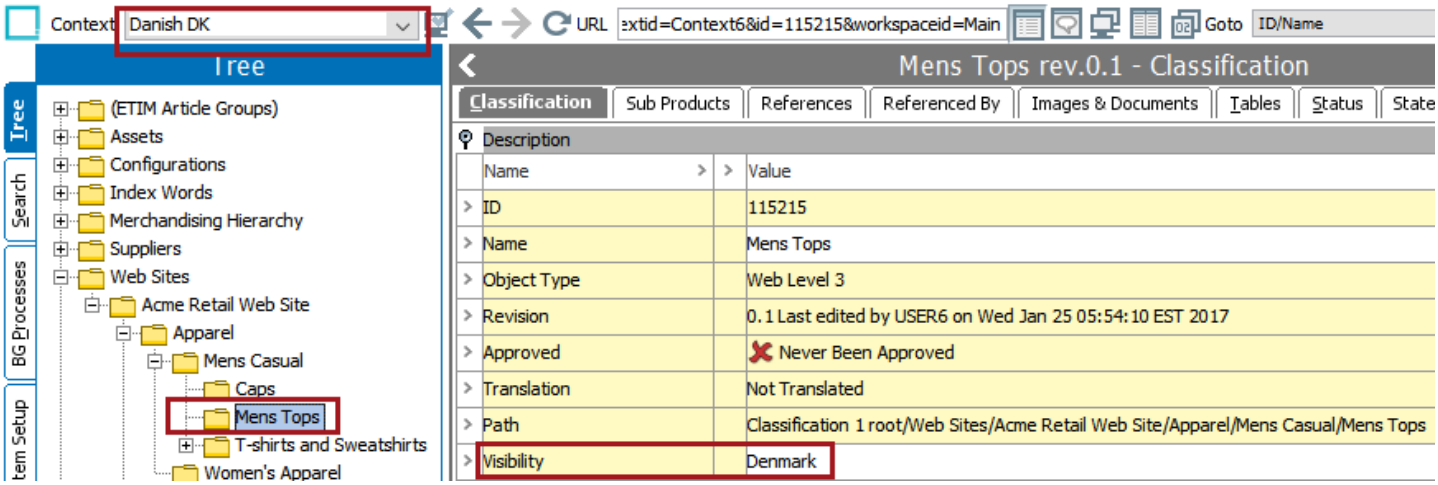


The example below shows an application where classifications have been made country-dependent, and an entire web structure has been created that is visible only in a context where English US is the selected country. The Web Sites Classification folder is not visible in the French FR context.



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.



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.

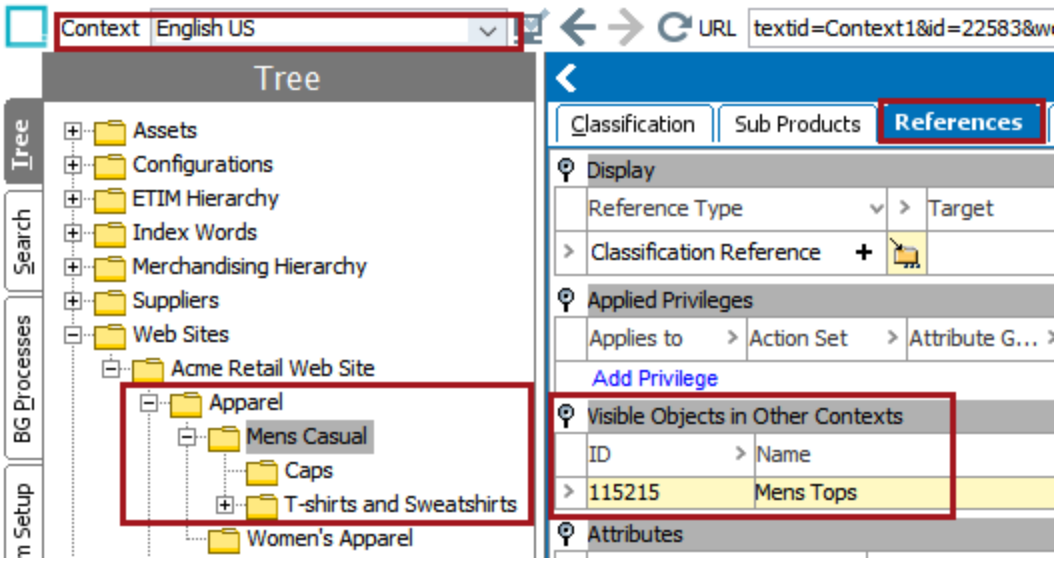
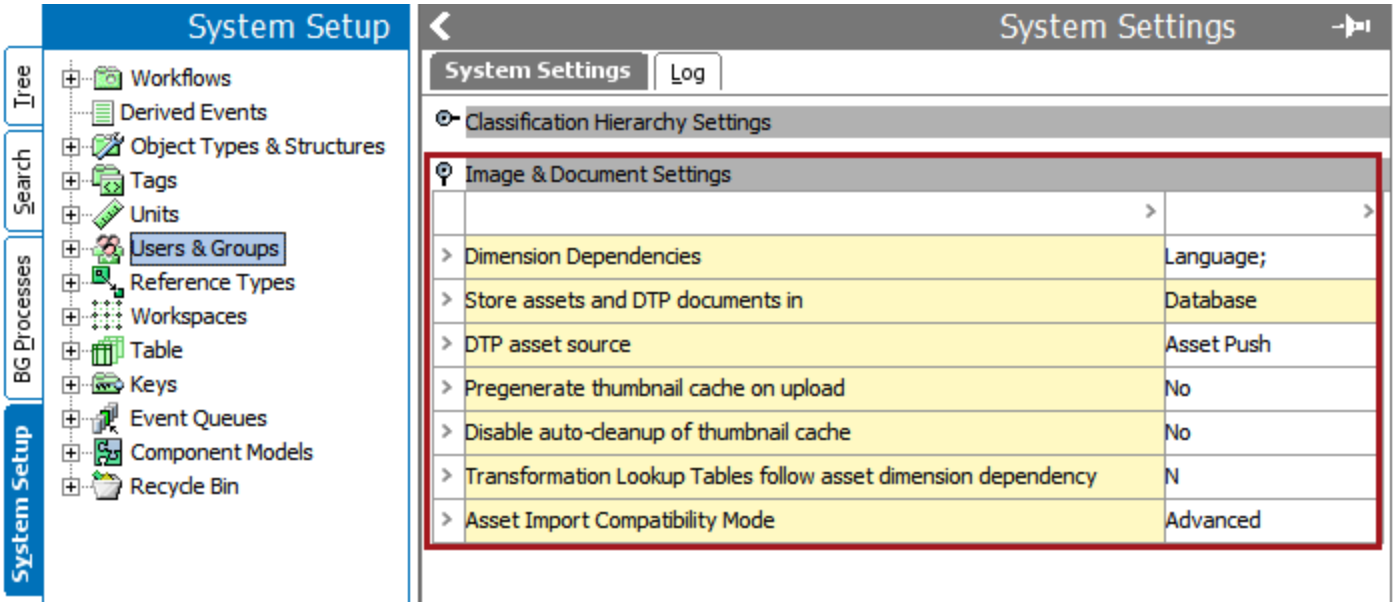


Image and Document Settings

This section is used to describe the various options available in the **System Setup** > **Users & Groups > Image & Document Settings**.



Dimension Dependencies

The Dimension Dependencies option 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 the ellipsis 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: Database and File System. Database is the default and stores the Assets and DTP documents to the Oracle DB. 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.

Important: This parameter should not be changed without first determining how assets are stored, and carefully considering the benefits of each option. If steps have already been taken to establish a File System solution, this parameter should not be switched back to Database unless the user is absolutely certain that Database is the desired setup.

Before this option can be modified, properties must be added in the sharedconfig.properties file, as defined in the **Initial Setup for an External File Structure** topic of the **Digital Assets** documentation.

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. A user can perform this action manually using the **Pre-generate Thumbnail Cache...** option on the File menu, as defined in the **File Menu** topic of the **Getting Started / User Guide**.

Disable auto-cleanup of thumbnail cache

By default, the thumbnail cache in STEP is cleared when it reaches 100MB.

- Set the option to 'Yes' to allow the cache to grow as large as needed to maintain the thumbnail images.
- Set the option to 'No' to improve performance and limit the cache to the default size.

Transformation Lookup Tables follow asset dimension dependency

A transformation that uses a lookup table can be 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 also converts the words 'semaine', 'mois', and 'année' to the corresponding abbreviations in the French language dimension.

While allowed, recommended practice is to create separate lookup tables for each language.

Note: This works only if the Dimension Dependency of the 'Lookup Tables' and the Dimension Dependency of 'Images and Documents' are set the same.

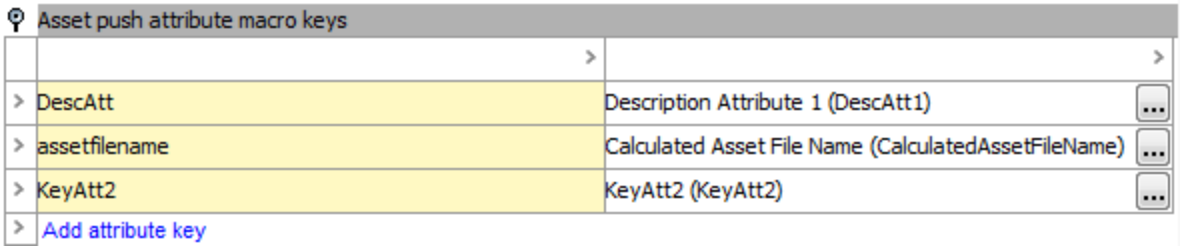
For more information, see the **Transformation Lookup Tables** topic in the **Resource Materials** online help.

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**, see the **Relative Path Template** section of the **Asset Push** documentation.



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.

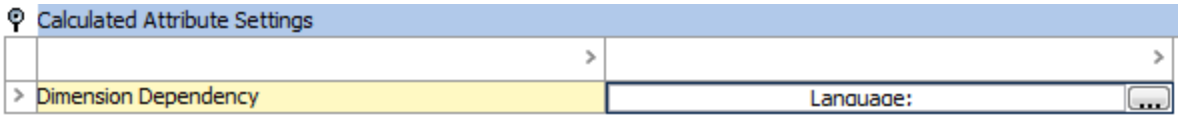
For example, when calculating 'width x depth x height' using the language dimension dependency,

- 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')
```



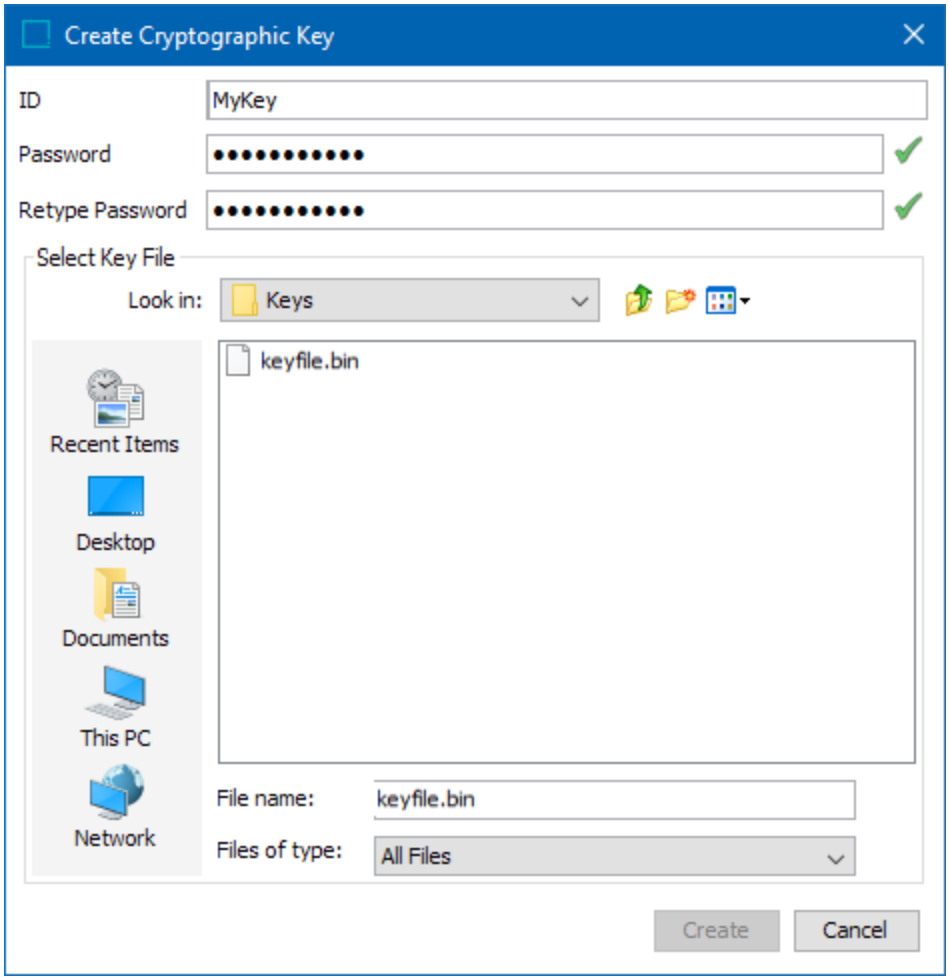
For information about calculated attributes, see the **Calculated Attribute** section of the **System Setup / Super User Guide** documentation.

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 HIPAA 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

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).

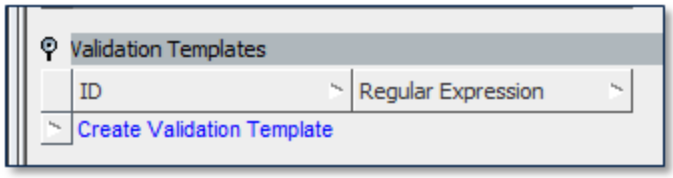
In STEP it is possible to make customized validation base types to be used on Attributes and LOVs. In this field you can create validation templates to be used when creating Attributes or LOVs.

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.

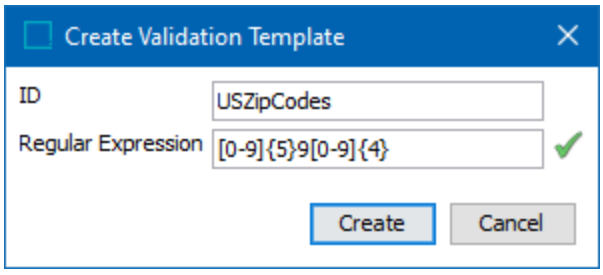
The example below shows how a new Validation Template may be created and how that new template subsequently displays in the drop-down list of validation base types when a new attribute is created (or when changing an existing attribute's validation).

To create a Validation Template:

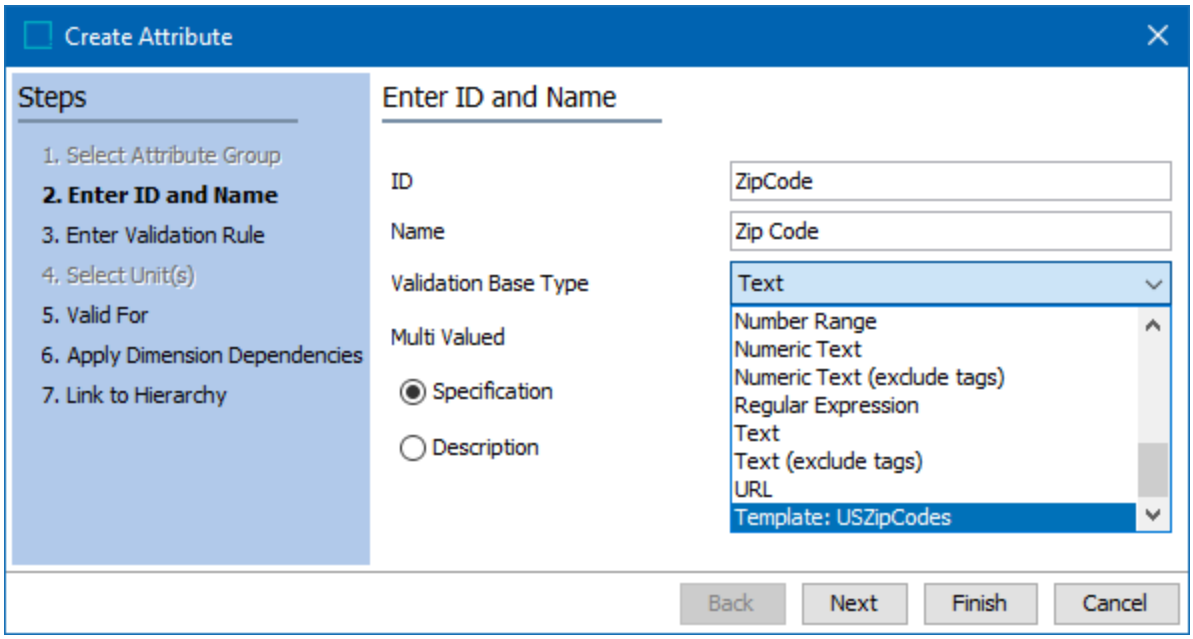
1. Click **Create Validation Template**.



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.



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 to edit the regular expression for that attribute, they must do it at the attribute level. Changing the Regular Expression Template in System settings has no effect.

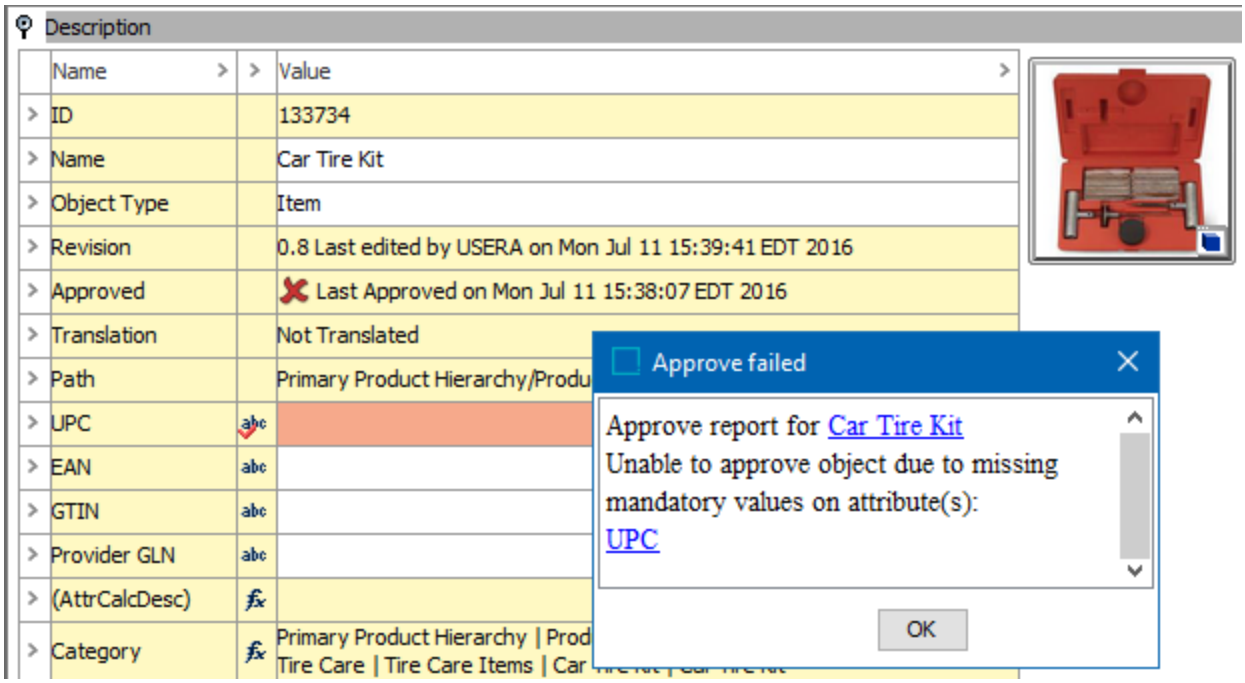
Product Information Manager Default Settings

This section describes the options available in System Setup > **Users & Groups > Product Information Manager Default Settings.**

Product Information Manager Default Settings	
Name	Value
Enforce Mandatory Check for Attributes, References and Links	none
Product Editor, Group attributes by top group	N
Localize numbers with thousand delimiter when localizing exports	Y
Localize dates when localizing exports	Y
Report Save As CSV Character Set	client-locale
Default Attribute to use as Display Sequence Attribute	DisplaySequence
Default Completeness Metric	Test
Conditional Validity Attribute	ConditionAttribute
Block Attribute Groups with more than 1000 attributes	Y
Use full pathname for classes on Product References Tab	N
Pass through unconverted Special Characters and Tags (Y) or discard them on output (N)	N
Product Attribute Help metadata attribute	Attribute Description (Attribute Description) ...
Attribute indicating the Priority of Product Variant Attributes	Product Variant Priority
Show both name and ID in the PIM navigator	Y
Comma separated list of node types for which to show both name and ID	
Attribute Header Column Width	250
Reference Type Column Width	120

Enforce Mandatory Check for Attributes, References and Links

This is the setting that controls whether or not objects can be 'Approved' if they are missing information that has been declared as mandatory. In the image below, the attribute UPC has been set up as mandatory. If the value is set to 'none', the object can still be approved, but the missing attribute value will give a visual indication of the missing data and shows in a background of red / brown. If the value for this setting is set to 'Y', the object cannot be approved.

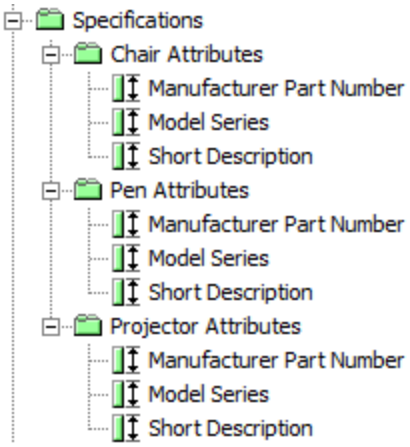


Note: This is a System-wide setting, thus it cannot be turned on and off at different places in the Tree hierarchies.

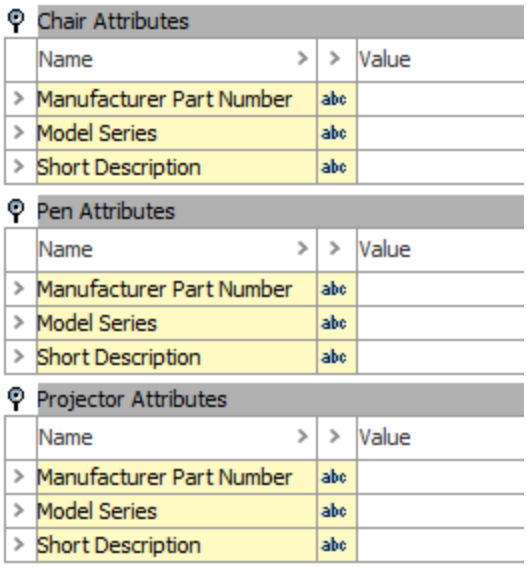
Product Editor, Group attributes by top group

There are cases where a user may create an attribute group that has multiple attribute sub-groups containing the same attributes. This can be the case when setting up User Rights and Privileges where a user can change an attribute value in one sub-group, but does not have the privilege to change the attribute value for the same attribute in the other sub-groups. This setting will control how these attributes are viewed in the workbench.

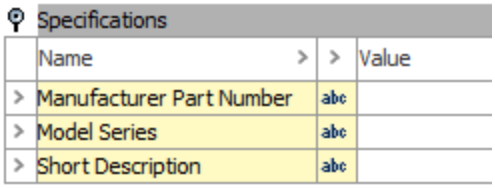
Fore example, you might have an attribute group called 'Specifications' as in the example below. Within that group, you may have several sub-groups containing the same attributes. Again, the use case would be that the user would be responsible for the 'Chair Attributes', but would not be granted privileges to change the same attributes within the 'Pen Attributes' sub-group or the 'Projector Attributes' sub-group.



When the value for this setting is set to 'N', the attribute sub-groups would show up in workbench as in the following image. In the use case example, the user would only have privileges to change "Chair Attributes' but 'Pen Attributes' and 'Projector Attributes' would still appear.



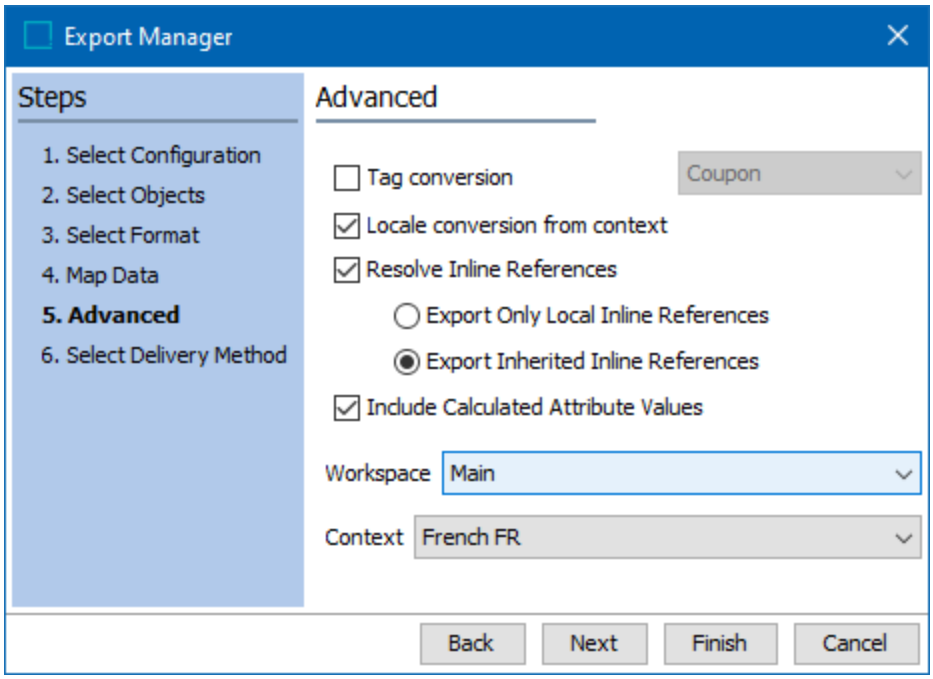
When the value for this setting is set to 'Y', the attributes would roll up into the top group. The user would still have privileges to update 'Chair Attributes' but 'Pen Attributes' and 'Projector Attributes' do not appear.



Localize numbers with thousand delimiter when localizing exports

This setting affects the way numeric values are shown for a selected context. For data exports, this setting causes the system to apply the appropriate thousand delimiter to attributes that have the validation of Number, Integer, Fraction, or Embedded Number. 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 Flatplanner and the STEP 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.

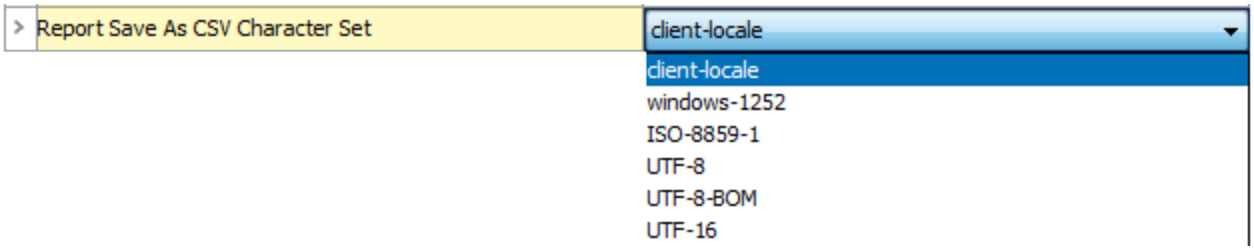
Localize dates when localizing exports

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 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.

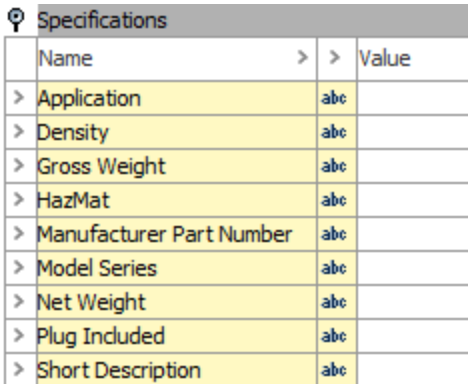
Report Save As CSV Character Set

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.

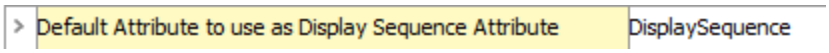


Default Attribute to use as Display Sequence Attribute

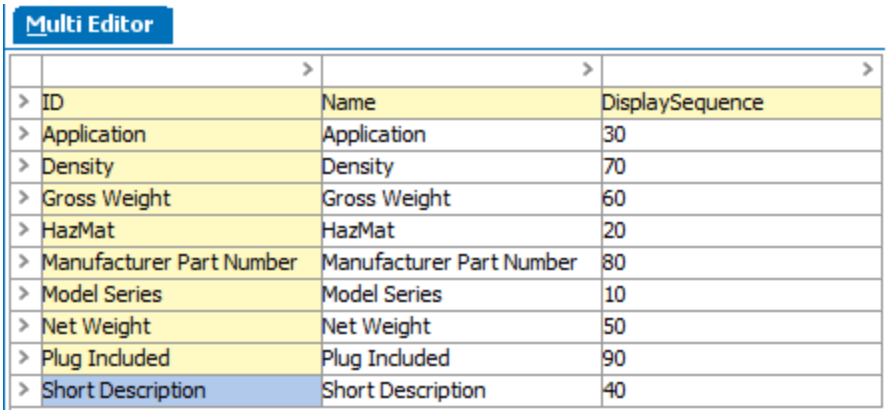
By default, attributes in attribute groups are shown in alphabetical order when viewed in the workbench or Web UI as shown in the example below. This setting allows the user to select the metadata attribute to use when displaying the attributes in the Product Editor to sequence the order of the attributes.



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.



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 Metric

STEP allows the use of multiple metrics. This setting allows for the selection of a default metric. When a metric is set as the default metric, the completeness meter shown on the product and entity editor tabs in the Workbench display the value from the default metric. For more information about Metrics, see the **Metrics** section of the **System Setup** documentation.

Conditional Validity Attribute

This setting is used to activate a conditional attribute. For more information about conditional attributes, see **Conditional Attribute Display** topic of the **Attributes** documentation.

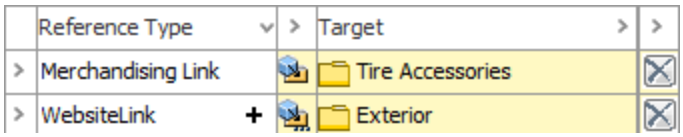
Block Attribute Groups with more than 1000 attributes

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

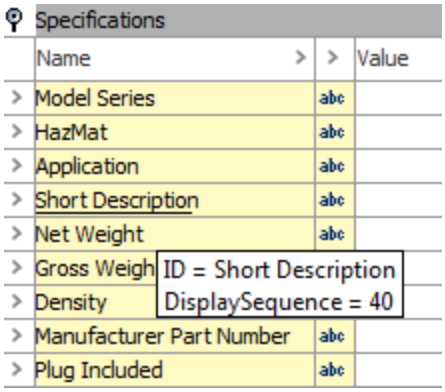


Pass through unconverted Special Characters and Tags (Y) or discard them on output (N)

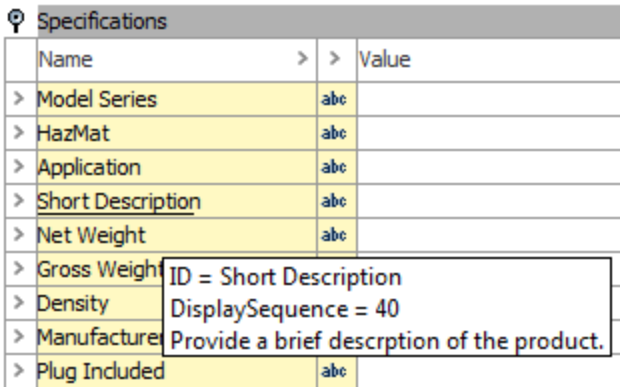
This setting is used to determine whether unconverted Tags and Style Tags should be outputted when there is not an actual conversion for that particular Tag. For example, if this option is set to 'Y', any tag that does not have a conversion in a certain media the actual 'internal' STEP tag will be output 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.



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.

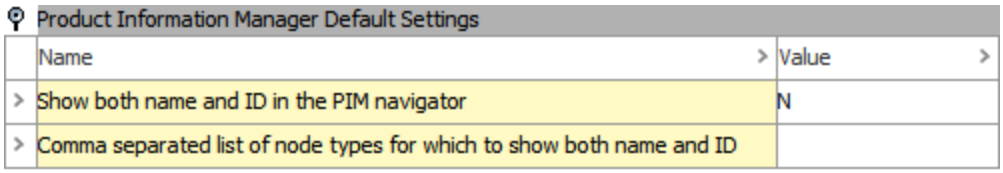


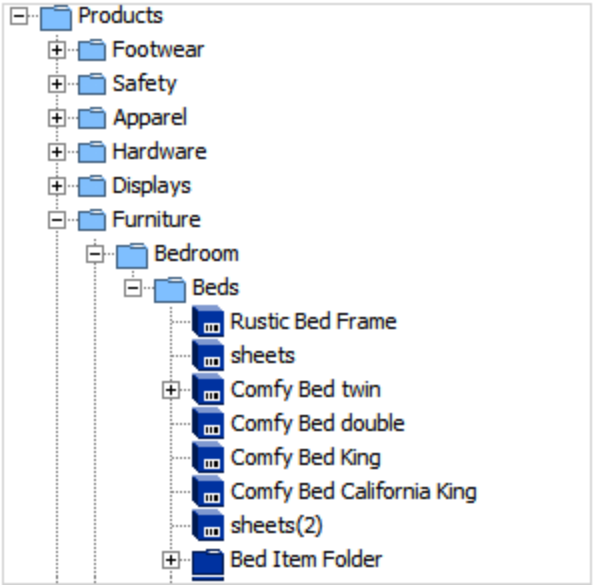
Attribute indicating the Priority of Product Variant Attributes

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, see the **Create a Product Variant Priority Attribute** section of **Product Variants**.

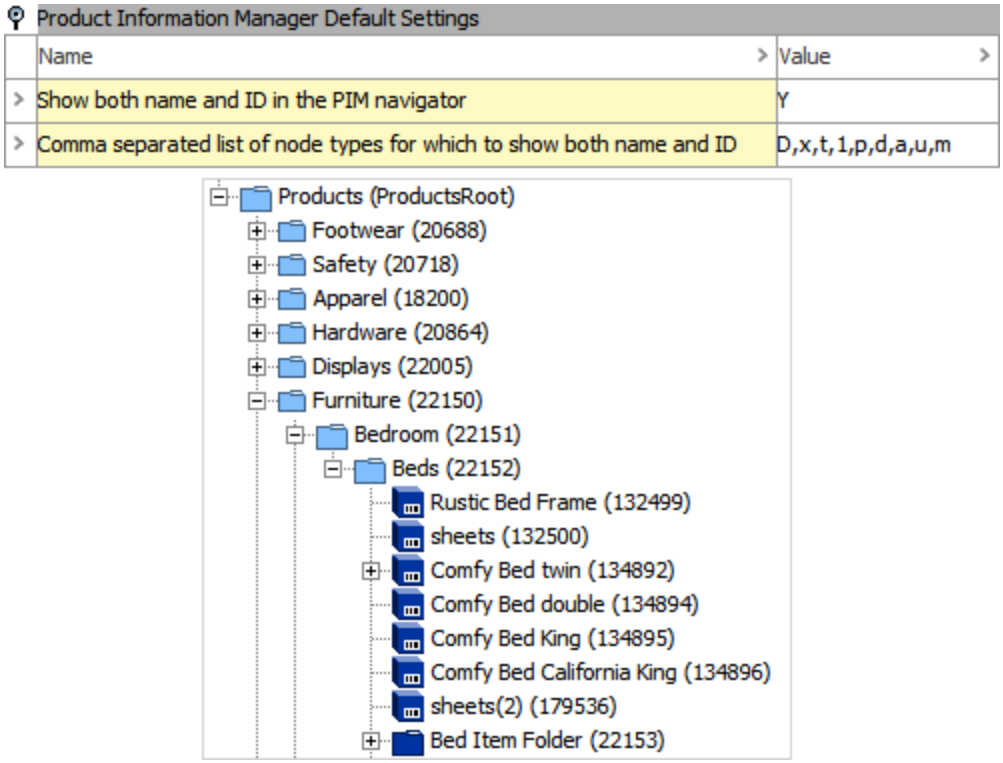
Show both name and ID in the PIM navigator

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. The default setting is set to 'N.'





When the setting the value is changed to 'Y,' users will need to change the **Comma separated list of node types for which to show both name and ID** setting.



For information on what the values in the **Comma separated list of node types for which to show both name and ID** setting relate to, see the next section.

Note: Once this setting is changed from 'N' to 'Y,' or vice versa, the user must log out of the workbench for the changes to apply. Also, if no values are populated in the **Comma separated list of node types for which to show both name and ID** setting, or there are values but the **Show both name and ID in the PIM navigator** setting is set to 'N', then no IDs will display.

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.

When selecting multiple node types, a comma must be used to separate the values.

Note: Entering values in this setting will not be effective 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

Note: These node type keys relate to the showing of the associated IDs in the Tree hierarchy not whether or not the associated object types are shown in STEP.

Attribute Header Column Width

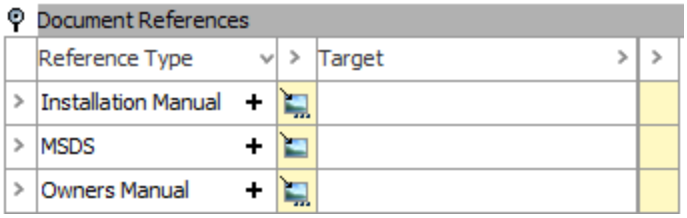
This setting affects the width of the Header Column when viewing products in the multi-products mode regardless of the rotation of the table. In the images below, the Header Column is the left most column in white. The first image has a column width of 250 and the second image has a column of 450.

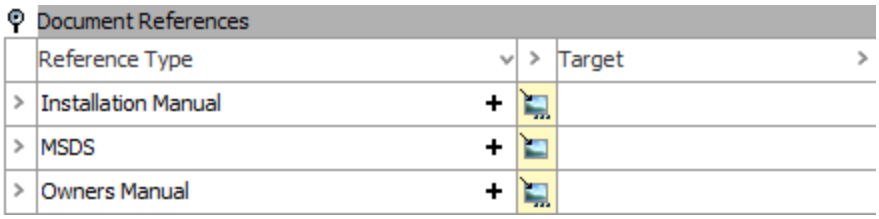
	ID	Name
Name	ID	Name
20004	133737	20004
20005	133738	20005
20006	133739	20006
2007	138845	2007
20-68204	20682	20-68204
555-22346	6806	555-22346

	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.

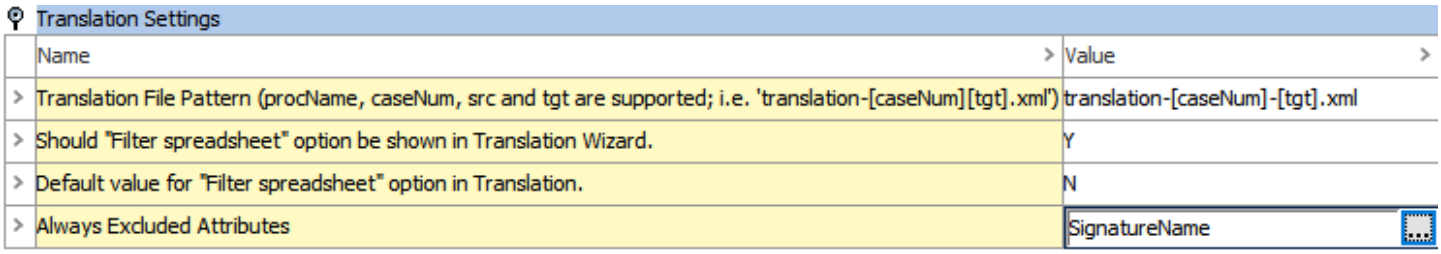




Translation Settings

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**.



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:

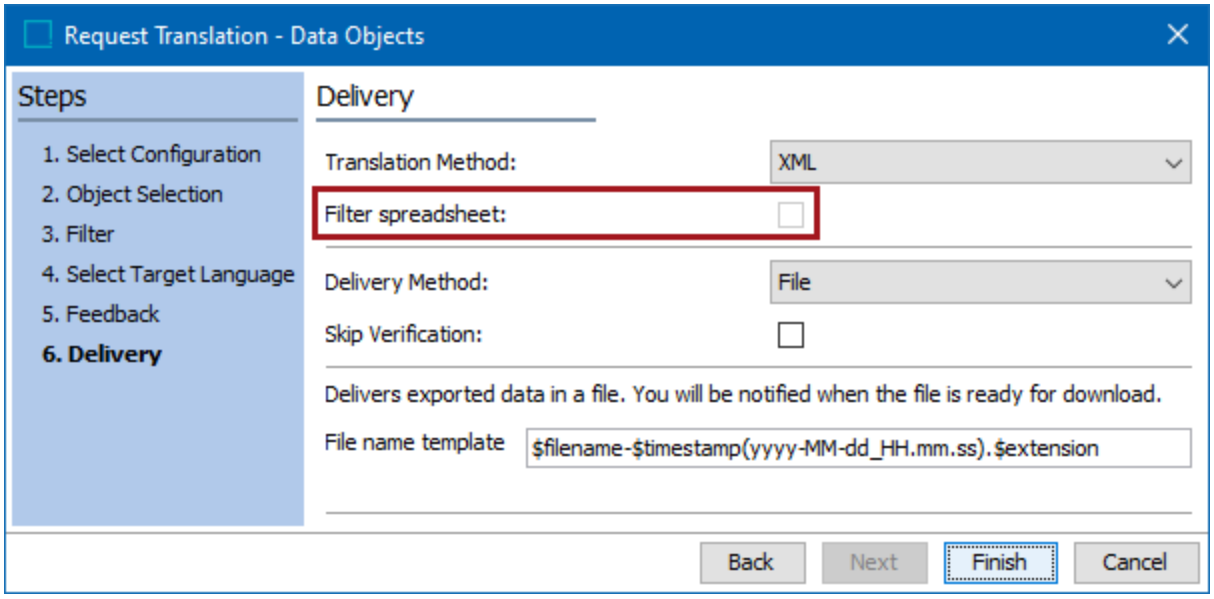
Translation-BGP_24304-US Eng-French Canadian.xml

Note: Users that currently incorporate or are planning to incorporate asynchronous services for translation purposes must include 'translation-' (and not 'procName') at the beginning of the file name pattern, as shown in the image above. For more information regarding asynchronous services for translation, see the **Asynchronous Translations** topic in the **Data Integration** documentation.

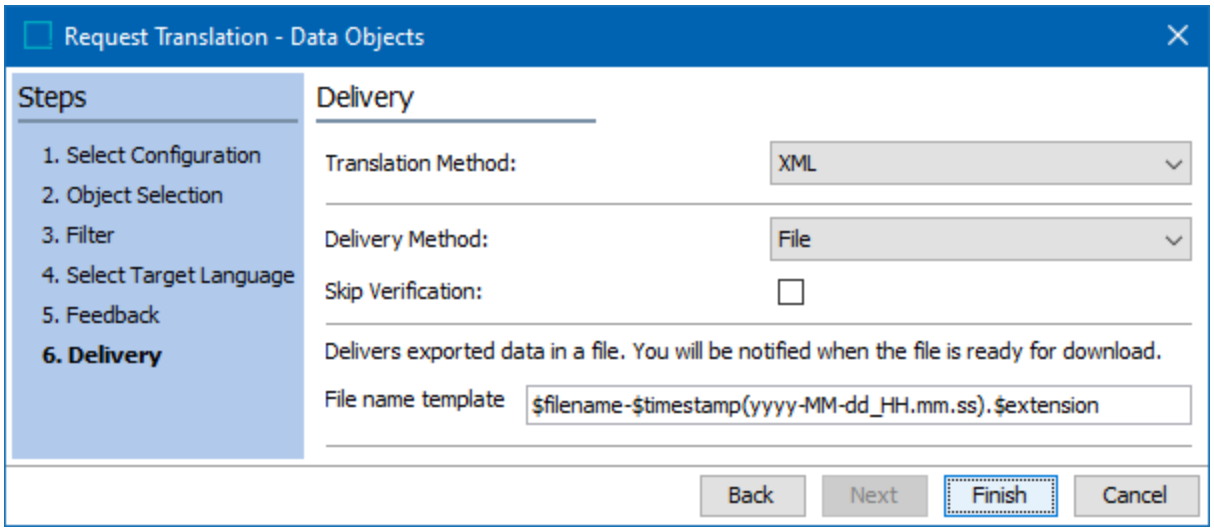
Should "Filter spreadsheet" option be shown in Translation Wizard

The 'Should "Filter spreadsheet" option be shown in Translation Wizard.' setting provides the ability to control whether or not an XLS translation file should contain both text attribute values that need to be translated as well as text attribute values that do not need to be translated. This is accomplished by giving users the ability to filter out all text entries from the XLS translation file that do not require translation by checking a 'Filter spreadsheet:' option.

If set to **Y**, the 'Filter spreadsheet:' option will display on the 'Delivery' screen (step 4) of the 'Request Translation' wizard, which is used when starting a translation workflow. If checked, the exported XLS file will only include text entries that require translations.



If set to **N**, the 'Filter spreadsheet:' option will not be available on this screen.



This option must be set to **Y** before the 'Default value for "Filter spreadsheet" option in Translation.' setting (described next) can be enabled.

Default value for "Filter spreadsheet" option in Translation

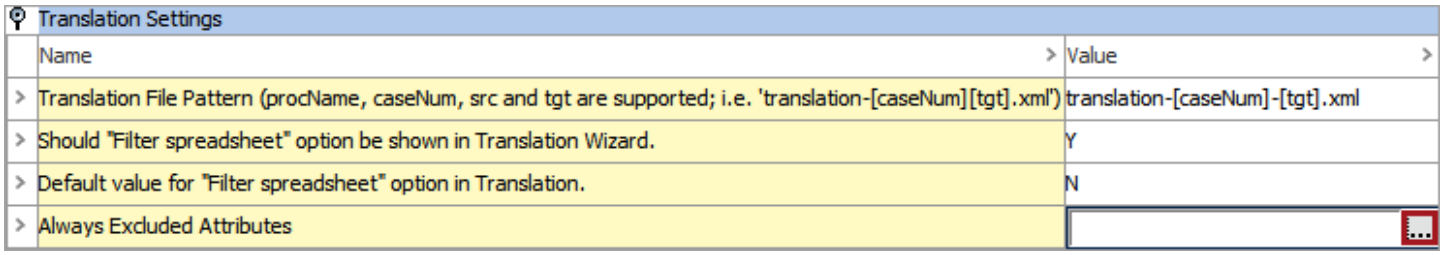
If set to **Y**, the 'Filter spreadsheet:' option will automatically be checked on the Delivery screen (step 4) of the 'Request Translation' wizard. However, this will only work if the 'Should "Filter spreadsheet" option be shown in Translation Wizard' setting is set to **Y**.

Always Excluded Attributes

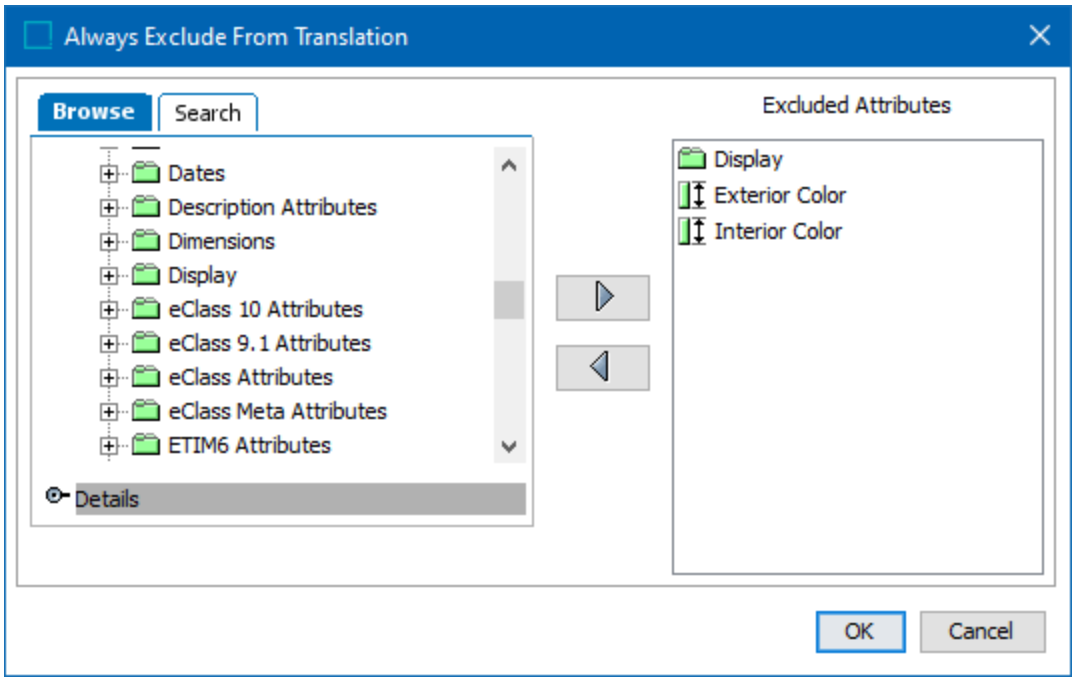
Language-dependent attributes and attribute groups selected in the 'Always Excluded Attributes' parameter will always be excluded from translation requests, i.e., the values set for these attributes and attribute groups will be ignored when creating a translation export, even if present on objects being translated.

To select attributes and attribute groups that should always be excluded for translations:

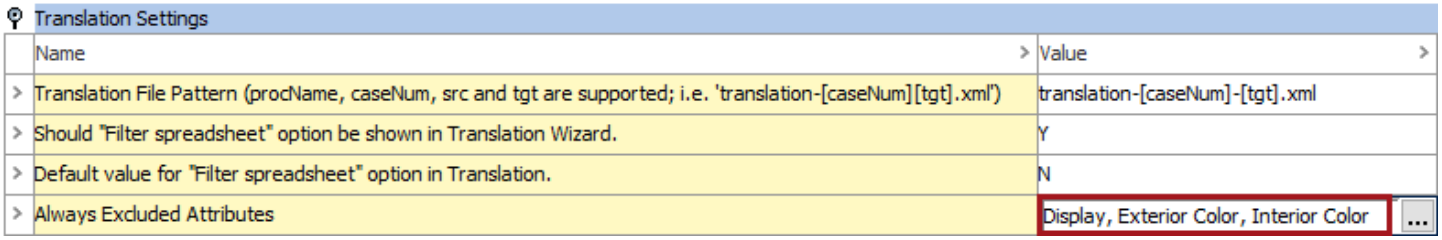
1. Click the ellipsis button (...) located next to the 'Always Excluded Attributes' parameter.



2. In the 'Always Exclude From Translation' dialog, select any language-dependent attribute or attribute group and click the right directional arrow to include in the 'Excluded Attributes' column. In the example shown in the screenshot below, the user has selected the attribute group 'Display' and the attributes 'Exterior' and 'Interior.' To remove any attribute or attribute group from the 'Excluded Attributes' column, select it and click the left directional arrow.



3. Click 'OK' to close the 'Always Exclude From Translation' dialog.



The selected attributes and attribute groups will now always be excluded from translations.

Terms List Settings

Add the products from the selection list import file directly instead of looking up leaf products

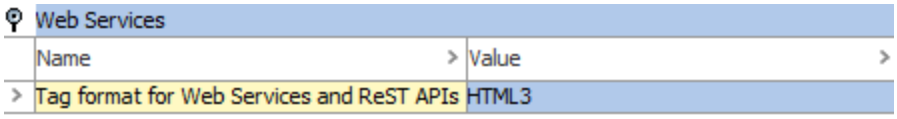
If set to 'Y' then commercial data can be stored against parent nodes. If set to 'N,' commercial data can only be stored against leaf products.

Note: This setting is only valid for eCatalogs and is now deprecated. It has no relationship to commercial terms lists used for print publishing.

Web Services

Tag format for Web Services and REST APIs

This option allows a user to state which output conversion format to use when accessing STEP via the API. The conversion format is not a dropdown list and the value must be entered manually, (e.g., HTML3), as shown in the image below.



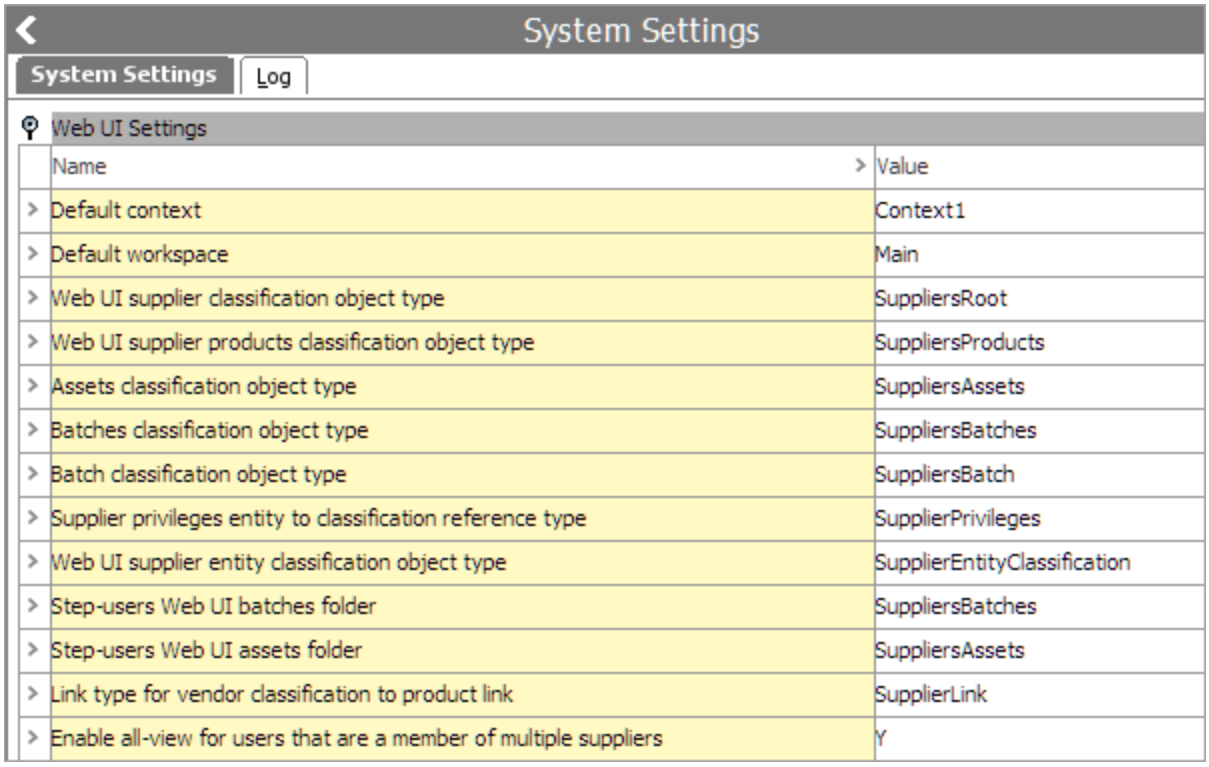
This option is only relevant for STEP systems licensed for the SOAP and REST APIs.

Note: If the purpose of using the API is to be able to edit and resubmit data to STEP, it is recommended that you do not use a conversion format, because the tag will not go through a reverse lookup on re-entry.

Web UI Settings

This section is used to describe the various options available in the **System Setup** > **Users & Groups > System Settings > Web UI Settings**.

Some settings may be controlled by licenses that may not be enabled for your system.



Note: Enter the ID—not the Name—of the Object Type.

Control Override Meta-Attribute

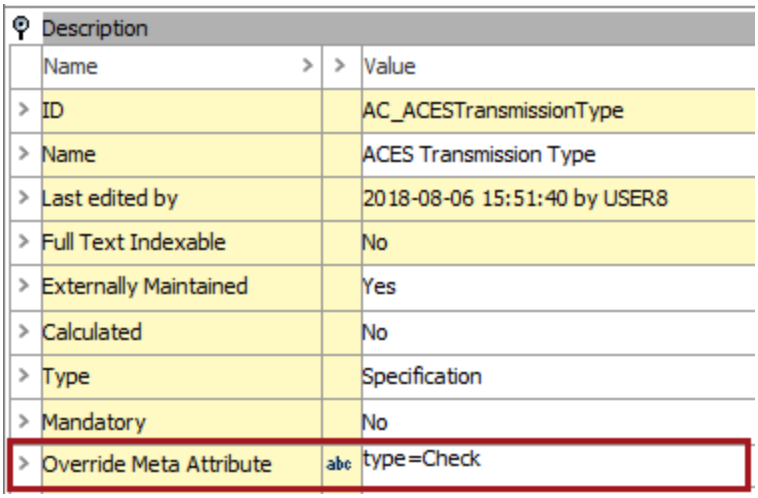
This setting is a way to define the control type for an attribute in the workbench on the attribute itself. This means that an admin user can define the control type without needing to open the Attribute Management screen in the Web UI or setting the override control type on the attribute itself when configured on a screen / list via the Web UI designer. Ultimately, this gives a user workbench control over everything about an attribute's configuration, even how it displays in the Web UI.

With this setting, the control type can be controlled in workbench and Web UI by following these steps:

1. Create a description attribute that is made valid / available for the Attribute object type.
2. In STEP Workbench, go to Users & Groups (as shown in the above image).

3. Under Web UI Settings > Control, override meta-attribute, populate the parameter with the description attribute created in step 1.

After set-up, users can also control the override setting from the description attribute (set up in step 1) for the specific attribute itself. See the example below:



The following is a list of the attribute values that can be used and is dependent on the validation base type.

Override Control Type	Validation Base Type	Control override meta-attribute values for workbench (case sensitive)
Checkbox	LOV	type=Check
Radio	LOV	type=Radio
Radio with clear option	LOV	type=Radiowithclearoption
Rich Text	Text, Text (exclude tags), Numeric Text, Numeric Text (exclude tags)	type=Richtext
Text Area	Text, Text (exclude tags), Numeric Text, Numeric Text (exclude tags)	Type=Textarea
Text Box	Text, Text (exclude tags), Numeric Text, Numeric Text (exclude tags)	type=Textbox
Typeahead	LOV	type=Typeahead

For override control in the Web UI, refer to the information in the **Override Control Type Rule** topic in the **Web User Interfaces** documentation and read more about (and see examples) of the control type overrides

available. Users should also be familiar with the **Attribute Value Component** and **Attribute Value Group Component** topics, also.

If override control is set up in Web UI also, when changes are made in workbench, it also changes (where applicable) in Web UI and vice versa. If a user types in an incorrect / invalid control type in workbench, then no changes are made in the Web UI.

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 a user signs into the Web UI for the first time. However, after a user signs into the Web UI, they can change the default workspace if desired.

When working in the Approved workspace, the 'Approved' workspace indicator is highlighted.



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). These object types connect the 'root node' for each of the created suppliers. Filling in this information is only required if you are setting up a Web UI that will include a supplier hierarchy. For more information regarding a Supplier Item Web UI setup, see the **Supplier Web UI Setup Guide**documentation.

Web UI Supplier Products Classification Object Type

Working in conjunction with the selected 'Web UI supplier classification object type,' this option identifies the root node where the supplier's products reside. For more information regarding a Supplier Item Web UI setup, see the **Supplier Web UI Setup Guide**documentation.

Assets Classification Object Type

This option identifies the object type that was created for storing the supplier's assets. For more information regarding a Supplier Item Web UI setup, see the **Supplier Web UI Setup Guide**documentation.

Batches Classification Object Type

This option identifies the root node where the batch classification objects reside. For more information regarding a Supplier Item Web UI setup, see the **Supplier Web UI Setup Guide**documentation.

Batch Classification Object Type

This option identifies the object type for the batch folders that reside under the batches classification object type. For more information regarding a Supplier Item Web UI setup, see the **Supplier Web UI Setup Guide**documentation.

Supplier Privileges Entity to Classification Reference Type

This option links supplier entities to the corresponding classification.

Important: When configuring the entity to classification reference type, you must set 'Externally Maintained' to 'Yes,' and set 'Allow Multiple References' to 'No.'

This setting is available when the Supplier Self-Service Onboarding commercial license is enabled on a system. Contact your account manager or partner manager to begin the process of enabling a license or licenses for your system. For more information regarding a Supplier Item Web UI setup, see the **Supplier Web UI Setup Guide**documentation.

Web UI Supplier Entity Classification Object Type

Working in conjunction with the selected 'Web UI supplier classification object type', this option identifies the root node where the supplier's entities reside.

This setting is available when the Supplier Self-Service Onboarding commercial license is enabled on a system. Contact your account manager or partner manager to begin the process of enabling a license or licenses for your system. For more information regarding a Supplier Item Web UI setup, see the **Supplier Web UI Setup Guide**documentation.

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 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 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 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.

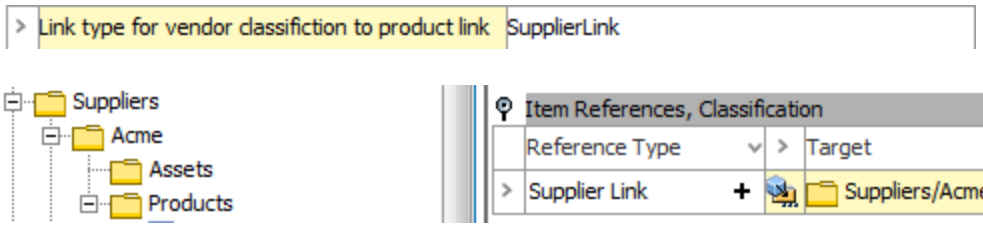
Context Help Metadata Attribute

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. For more information about using the Context Help metadata attribute, 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. Once configured, products created through the Supplier Item Web UI automatically link to the supplier's classification structure. As shown in the image below, the Reference Type ID is entered and will be made available for use in the Web UI.

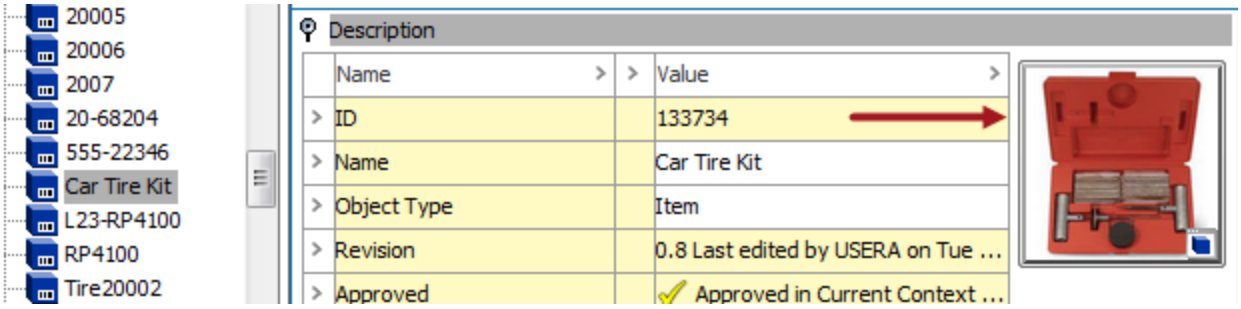


Enable All-view for Users that are a Member of Multiple Suppliers

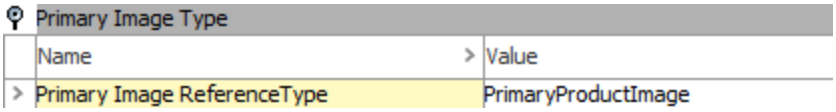
If set to 'Y,' this setting enables a supplier user that is a member of multiple supplier groups to have views across all the groups when looking at tasks. For more information about the multiple views for suppliers, see the **All-View Across Multiple Suppliers** section of **Supplier Web UI Setup Guide** topic.

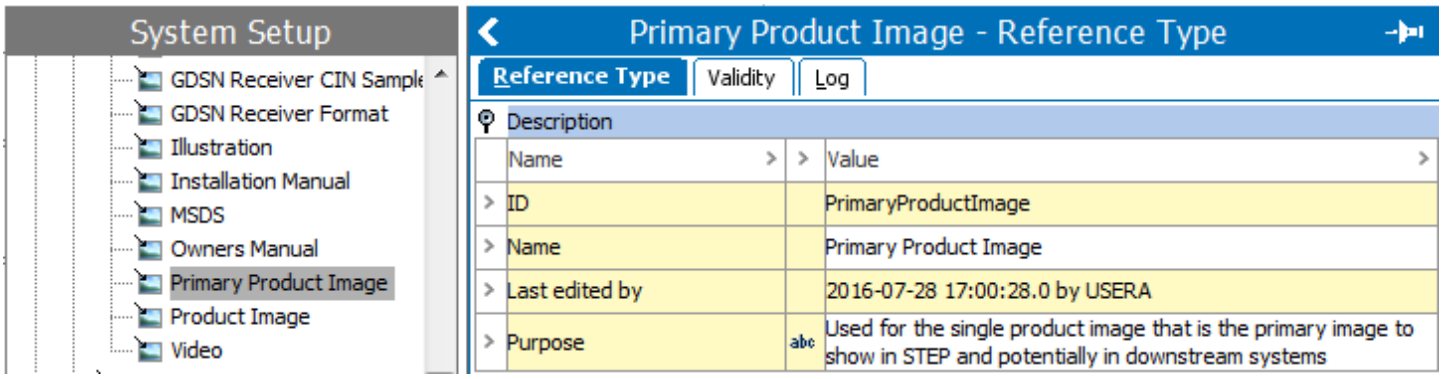
Primary Image Type

In the Primary Image Type flipper, the **Primary Image Reference Type** setting is used to display a thumbnail of the object image in the workbench or in the Web UI. An example of how this image looks in the workbench is shown in the image below.



By default, the **Primary Image Reference Type** value is the ID of the Reference Type, 'PrimaryProductImage.' However, if a user has created another reference type to be used to associate an image thumbnail to an object, the ID of that image type can be entered in this field and the new reference type will associate the images to the objects.





Flatplanner Default Settings

This section describes the available options under System Setup > Users & Groups > **Flatplanner Default Settings**.

Flatplanner Default Settings

Name	Value
Enable keep layouts for publications	Y
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.

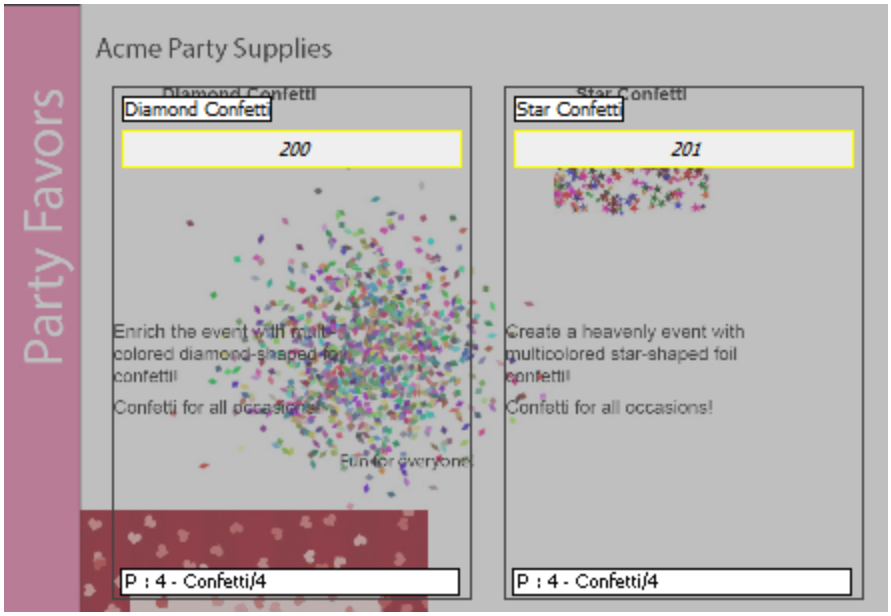
If 'Enable keep layouts for publications' is set to **N**, users are free to save manually adjusted pages back to STEP, but the next time a page is mounted from the corresponding Flatplan, it will mount with the layout from the original product template(s).

(Note that a 'kept' layout will be removed from the Flatplan if the corresponding mounted [actual] pages are deleted from STEP.)

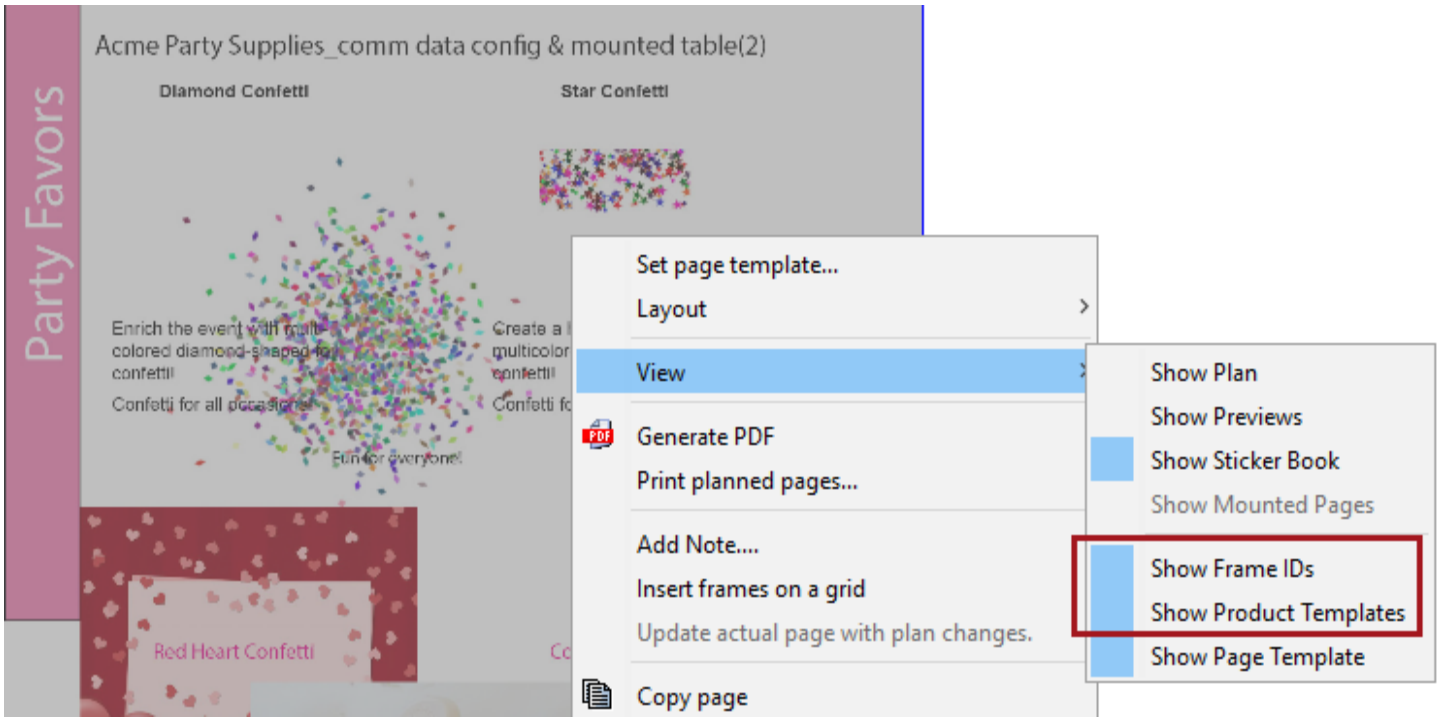
For more information, see **Reusing Saved Layouts** in the **Flatplanner** documentation.

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

Setting this option to **Y** allows for object layout pagination rules to be reapplied to non-empty planned page frames, i.e., frames that contain either a product, classification, or asset. If the product template applied to the frame has been changed to something other than the template specified in the pagination rule, it can be reverted to the template specified in the pagination rule by using the 'Reapply layout rules' feature in Flatplanner. If this option is set to **N**, then 'Reapply layout rules' will not work.

For more information on how to use the 'Reapply layout rules' feature, see the **Applying Product Templates to Page Frames** topic in the **Flatplanner** documentation.

For more information on Flatplanner pagination rules, see the **Flatplanner Pagination Rules** section of the **Flatplanner** documentation.

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	Print Display Sequence (PrintDisplaySequence) ...
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
Proof View Start Page Number	1

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 panel and choosing 'Create document from plan'), the default file name of the mounted document is the page number (for example, 1.indd). By setting 'Use Publication Name prefix when naming DTP documents' to 'Y', the publication name will be added as a prefix to that document file name. For example, if the publication name in STEP is 'Bath Catalog Fall Collection,' the mounted document will be called ''Bath Catalog Fall Collection_1.indd'

Note: If there are spaces or any invalid or 'questionable' characters in the publication name (for example, ™), they will be removed.

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 Publisher** documentation.

Calculated DTP Table width includes left and right rules

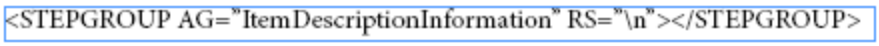
This option tells STEP to make an adjustment to correct an anomaly in InDesign tables when vertical rules are placed on the left and right side of a table. InDesign does not normally include the width of a left or right vertical rule in the overall width of a table. Instead, it effectively over-sets 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 Publisher 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.



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 Publisher** 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 Publisher** documentation.

Asset Order Meta-attribute

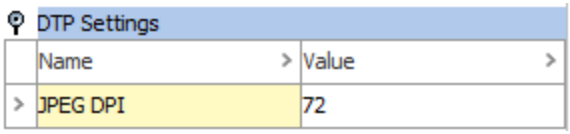
This feature enables a sequence order to be set on assets (usually images) when products are mounted to InDesign pages. This sequence is set by the use of a metadata attribute that is valid on the link between products and image reference types. If there are multiple references from a product to various 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 Publisher** 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



This setting affects the regular product proof view (as accessed from the Proof View tab in the workbench and through the Proof View component in the Web UI) 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.

For more information on Proof Views, see the **Proof View** section of the **STEP Publisher** documentation.

DTP Page - preview image

DTP Settings	
Name	Value
DTP Page - preview image	Low-resolution

Images that are mounted onto InDesign pages are represented on the page by low-resolution images. These low-resolution images are linked to the original high-resolution images, and InDesign will know where the original images are stored as long as the image links are not broken.

The 'DTP Page - preview image' setting is used to control whether the low-resolution or high-resolution images will be embedded in Flatplanner page previews, which are fetched from the STEP Workbench and generated by the InDesign Server (IDS). This setting is intended to improve the time taken to render page previews. If the setting is changed, the IDS must be restarted before the new setting takes effect.

The 'DTP Page - preview image' options and their behavior are as follows:

- **None:** No preview of DTP page. Not recommended.
- **Low-resolution**: Tells the InDesign server to build the page preview with the low-resolution images embedded in the InDesign document.
- **High-resolution:** Tells the InDesign server to build the page preview from the actual high-res images. This is the recommended setting.

If a Flatplanner page does not have an abnormally high number of images, e.g., less than 80, then there is not a noticeable difference in the time that it takes to render the preview, making the setting of high-resolution appropriate.

Note: The 'DTP Page - preview image' setting and the immediately preceding setting of 'JPEG DPI' are unrelated.

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.

Proof View Start Page Number

This setting allows for Proof Views and the PDFs created from Proof Views to begin on either page 1 (a right-hand page) or page 2 (a left-hand page). A starting page of '2' is the default.

A selection of '1' is useful, for example, when producing documents for PDF output, such as datasheets, that may not be produced in a 2-page spread format and should start page numbering at 1.

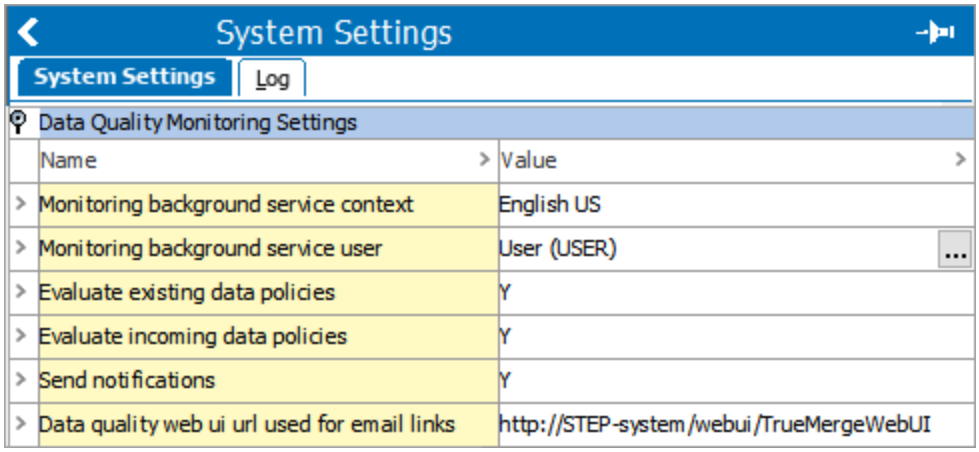
For more information on Proof Views, see the **Proof View** section of the **STEP Publisher** documentation.

Data Quality Monitoring Settings

This section is used to describe the various options available in the **System Setup** > **Users & Groups** > **System Settings** > **Data Quality Monitoring Settings**.

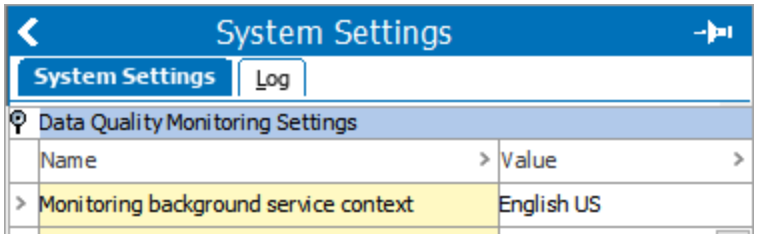
Important: This section only displays if the cmdm-monitoring component has been activated. Contact Stibo Systems with assistance, if needed.

The information within this section outlines the settings needed to properly monitor data policies.



Selecting a Data Policies Default Context

Data monitoring is performed in a single context. In this example, English US was selected.



With English US selected, all other context in the system will go unmonitored. For more information, see the **Web UI Configurations for Policies** topic in this documentation.

Monitoring background service user

This option sets the ID of the user that executes the 'MonitoringBackgroundServiceUserEntry' global setting. If left empty, the system will set this value to 'stepsys.' In this example, it is set to USER.

Note: Any attribute values need to be readable by this user for data monitoring. It is recommended that the user should have elevated permissions, specifically a system user with privileges to read anything in the system.

Evaluate existing data policies

If this is set to 'N,' then no evaluation will be done on existing data. This setting is for the policies running on data already in the system.

Evaluate incoming data policies

If this is set to 'N,' then no policy evaluation will be done on stream imported nodes like the Match and Merge importer.

Send notifications

If set to 'N,' then no notifications will be sent on policy breaches.

Data Quality web ui url used for email links

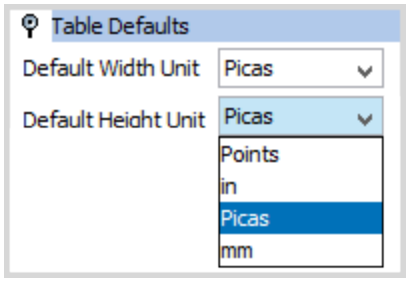
This value is where the policy list is located in the associated Web UI. To find this URL, navigate to the policy screen to be displayed, and copy everything before the first hash. In the following example, the text in bold is used as the Web UI URL.

```
http://STEP-system/webui/TrueMergeWebUI
#contextID=Context1&workspaceID=Main&selection=260789&nodeType=collection&displayMod
e=-1680893663.0
```

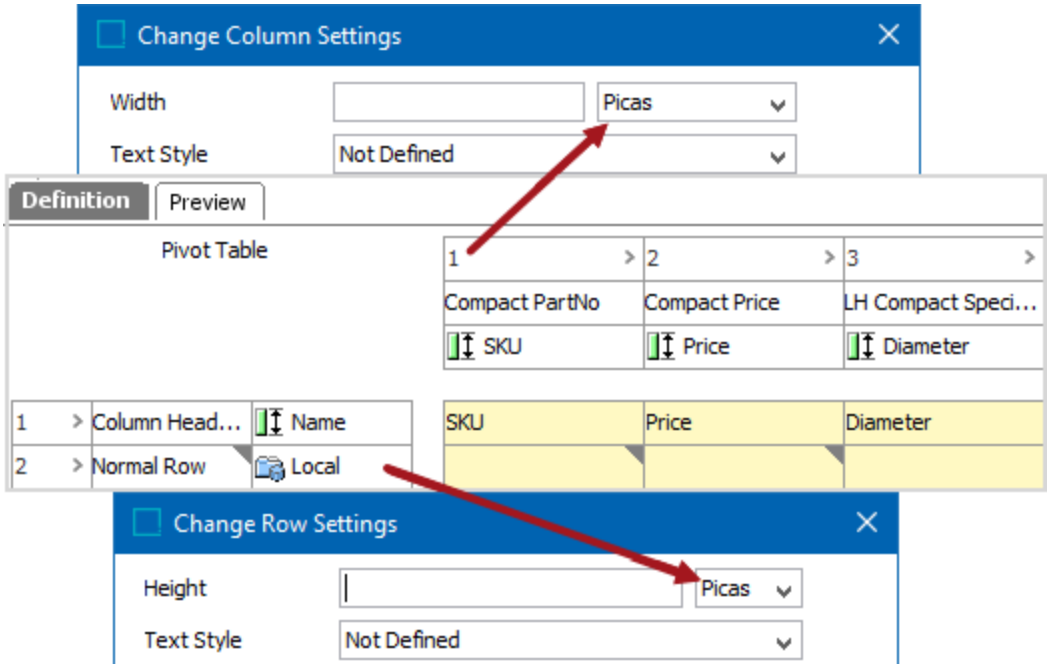
This allows email to include links to the list or to a specific policy. Without this option set, there will not be a 'View Policy List' button in notification emails.

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.



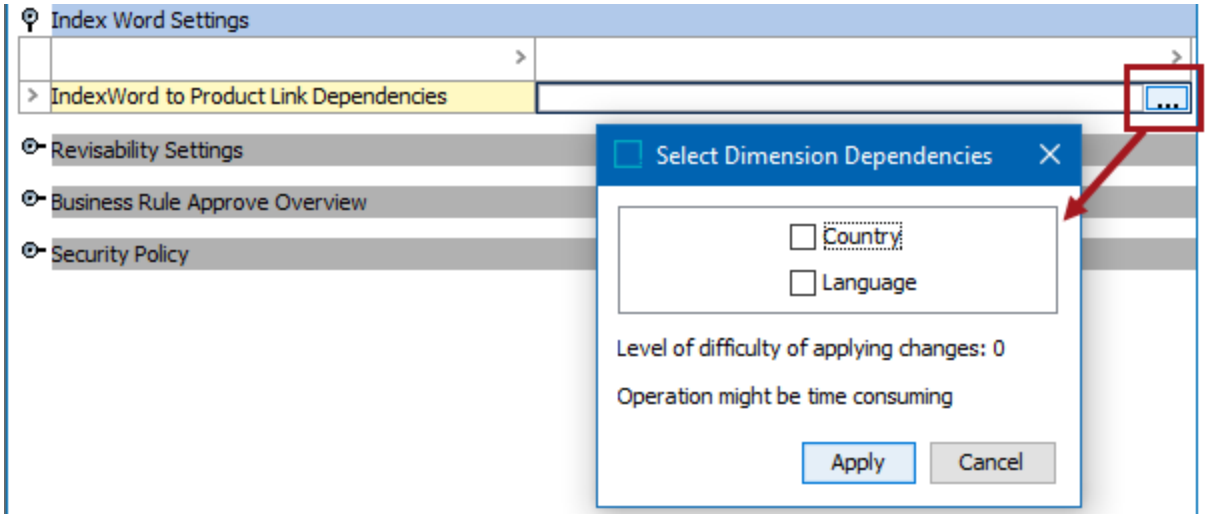
Once the default units have been set, the units will be the default units in the table editor as shown in the image below. If no default unit is selected, the default units will be blank and must be manually selected every time.



For more information on STEP Tables, see the **Tables** documentation.

Index Word Settings

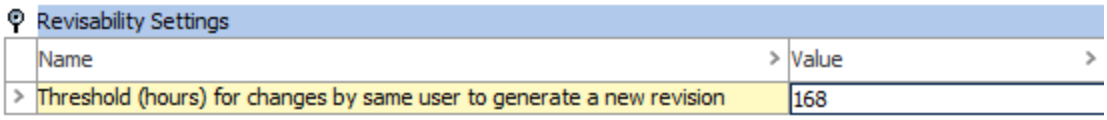
By default, the links between products and index words are not dimension independent. By selecting a dimension under 'IndexWord to Product Link Dependencies,' you can make these links dimension dependent. Making the links between products and index words dimension dependent can be useful for a situation where not all products are sold in all countries.



For more information on index words, see the **Creating Document Indexes** section of the **STEP Publisher** documentation.

Revisability Settings

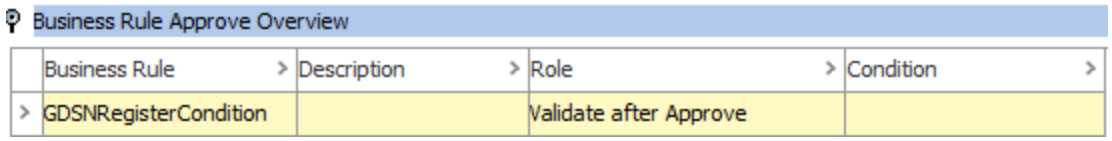
Threshold (hours) for changes by same user to generate a new revision



STEP automatically preserves all versions of a revisable object. Each time a change is made by a different user, a numbered revision is created and logged. This setting sets the threshold for the number of hours to occur before a new revision by the same user is reflected in STEP. The default setting is 168 hours or one week. 24 hours is the minimum number of hours that can be entered.

For more information about revisions, see the **Revisions** topic.

Business Rule Approve Overview


Business Rule Approve Overview

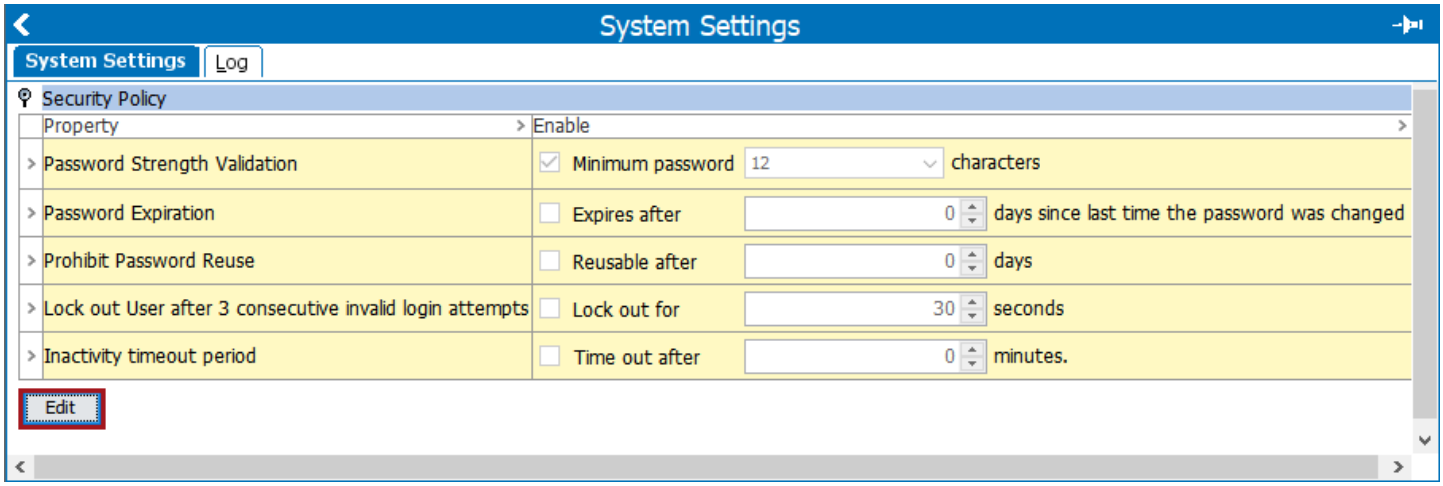
	Business Rule	Description	Role	Condition
>	GDSNRegisterCondition		Validate after Approve	

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, see the **Business Rules on Approval** section of the **Business Rules** documentation.

Security Policy

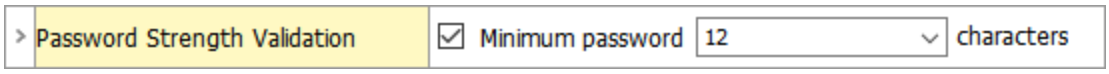
Admin users can configure security settings related to passwords to ensure that STEP users adhere to the required password security level. To access the password settings, on the **System Setup** tab, select the **Users & Groups** node. Scroll down to find the **Security Policy** flipper, which includes the following settings, shown below:



Important: Changes made to these parameters will be applied to **all** user groups located under the **Users & Groups** node. These settings cannot be customized for single users or user groups.

To change the Security Policy parameters, click the **Edit** button below the parameters to display the Security Policy edit dialog.

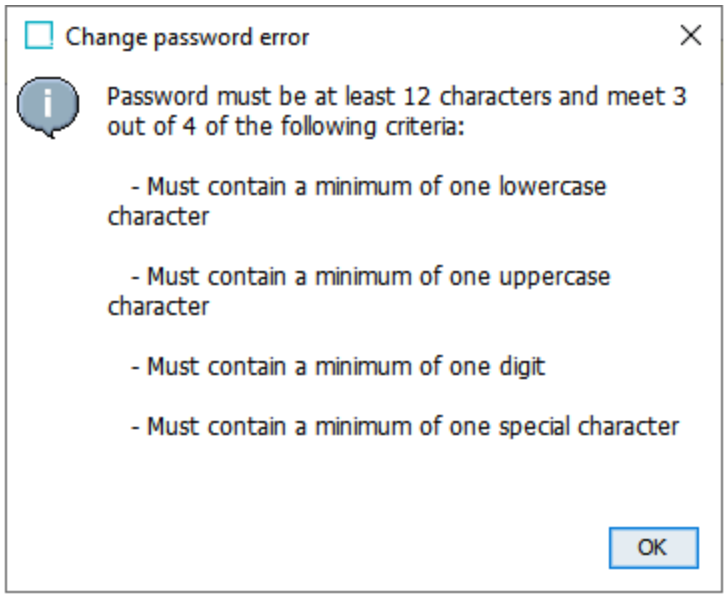
- **Password Strength Validation** – When checked, new passwords must conform to at least three of the four requirements intended to ensure a strong password.

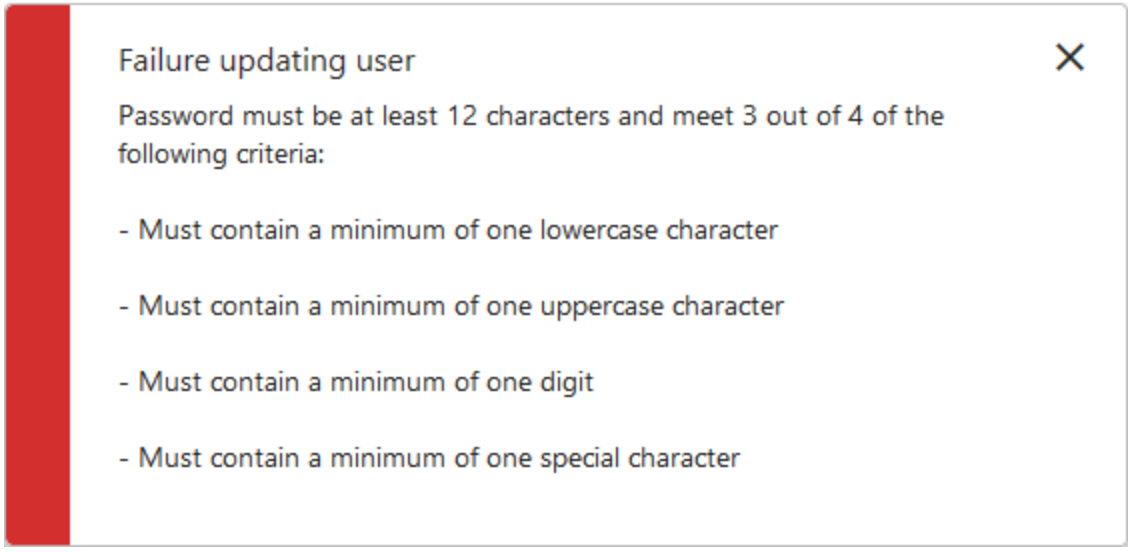


  - Must contain a minimum of one lower case character
  - Must contain a minimum of one upper case character
  - Must contain a minimum of one digit
  - Must contain a minimum of one special character

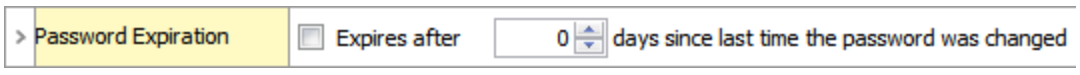
  Additionally, the password must meet the minimum password character requirement, which ranges from the default of eight (8) characters to the maximum of 20.

The following error displays in workbench and in Web UI if not all password requirements are met.



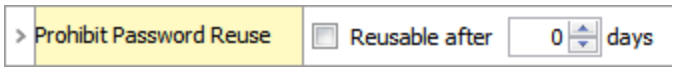


- **Password Expiration** – When checked, all new passwords will expire after the set number of days have passed. Once a password has expired, upon the next log in attempt, the user is notified and prompted to change their password.

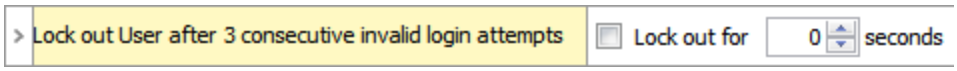


Note: Passwords can be set to not expire for a user group via the Group tab for a specific user group. See Working with User Groups documentation for more information.

- **Prohibit Password Reuse** – When box checked, the system prohibits a user from reusing old passwords. Old passwords are unavailable for reuse based on the set number of days.

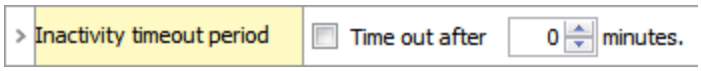


- **Lock out User after 3 consecutive invalid login attempts** – When checked, three (3) failed attempts to log in by entering a wrong or mistyped password will prohibit any new log in attempts for the set number of seconds.

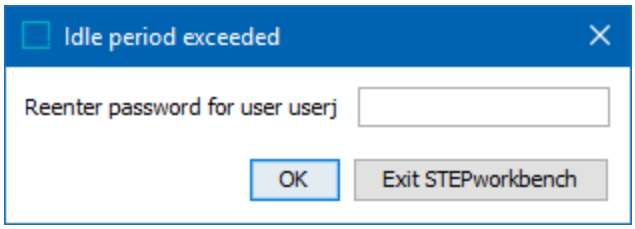


Note: Administrators who reset passwords for users locked out of the system also reset the time-out period for those users.

- **Inactivity timeout period** – When checked, the system can be configured to automatically sign out any logged in user once the set number of inactive minutes is reached. This setting is most appropriate for STEP systems with a large number of active users.

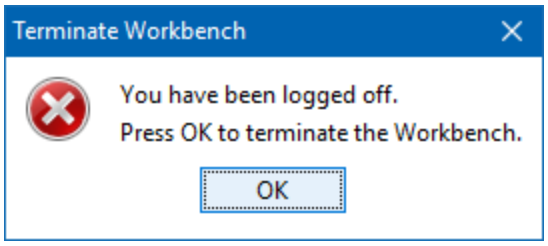


Once the inactivity timeout period is reached, the following dialog is displayed for the logged out user, and the password must be entered to resume work in workbench.



As an additional, but unrelated security measure, workbench token renewal time is required every four hours by default. While increasing that time also increases token-related security risks proportionally, modifications can be made via the sharedconfig.properties file. Include the case-sensitive property text 'Step.Token.ExpiryTimeInHours=[hours]' to set a time between five and 72 hours. For example, Step.Token.ExpiryTimeInHours=5.

When the expiry time is reached, the Terminate Workbench dialog shown below is displayed and the user is logged out.


Terminate Workbench
You have been logged off.
Press OK to terminate the Workbench.
OK

GUI Localization

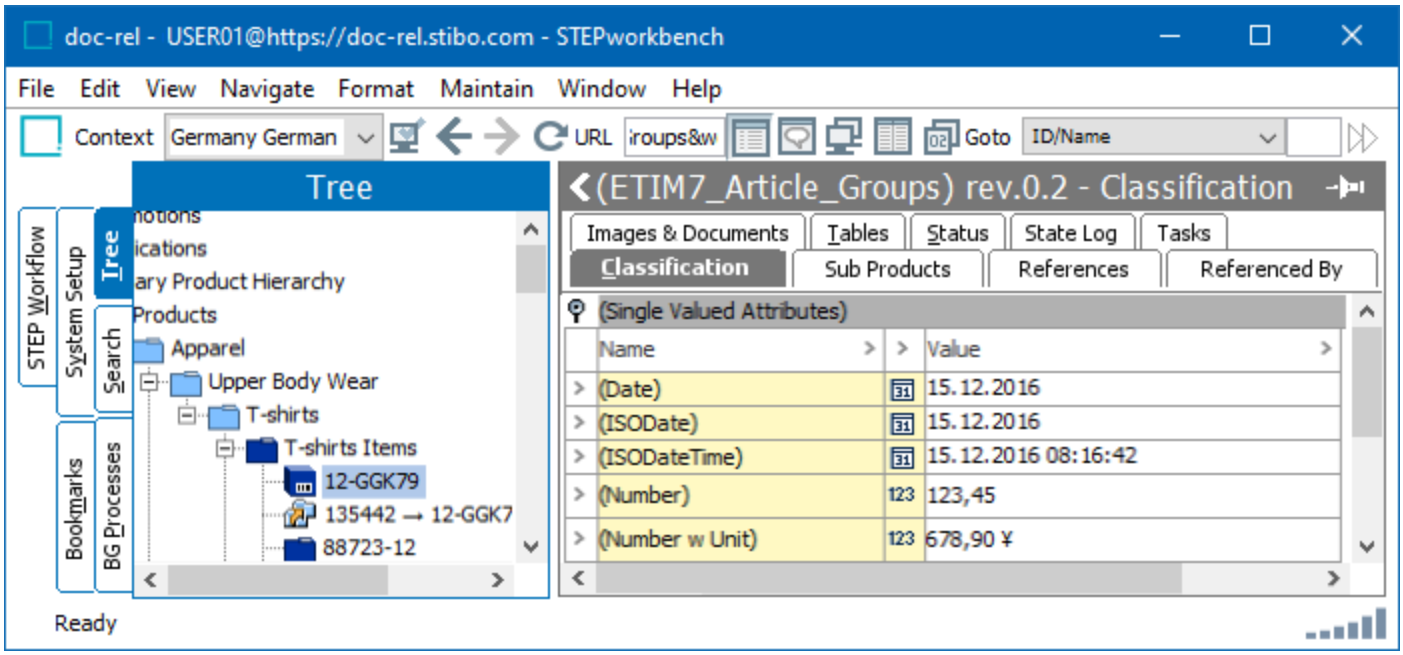
GUI Localization is maintained in System Setup > **Users & Groups** > **Users** on the **System Settings** tab.

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.

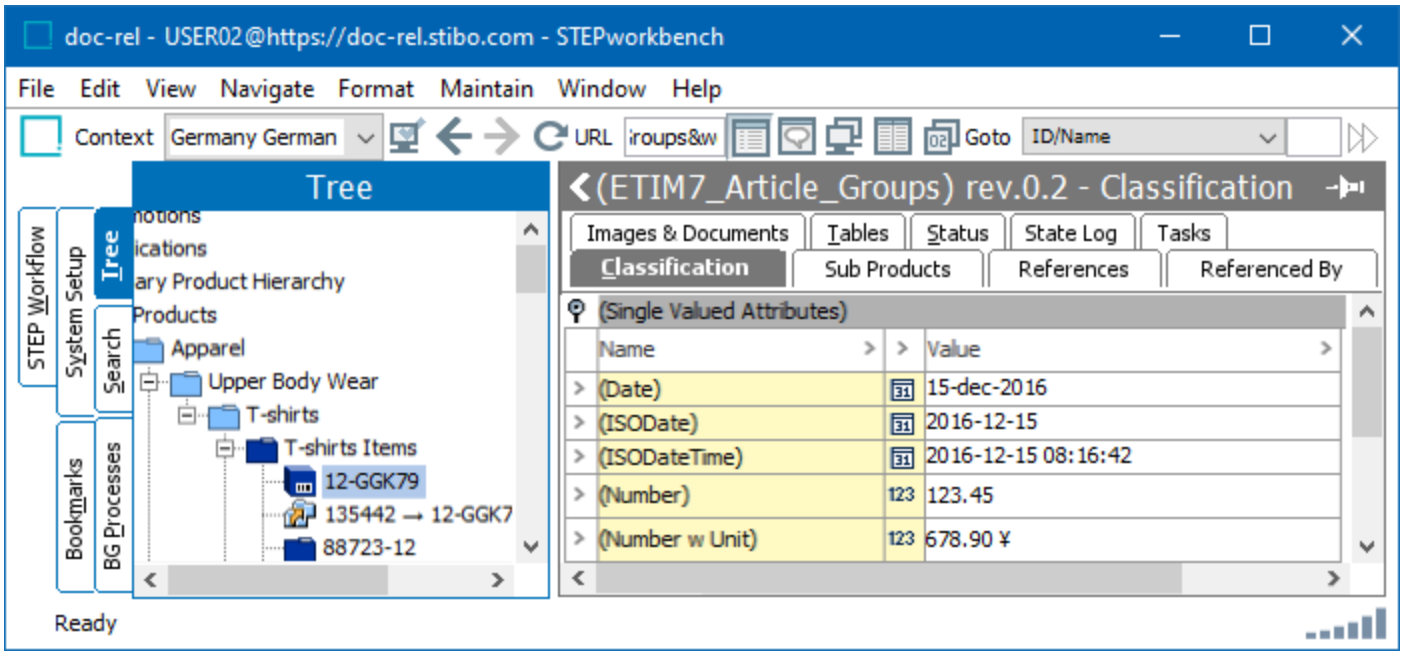
For example, User 01 is using German - de/de locale:



In this example User01 is logged on the Workbench. The User is setup to use a German - de/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.

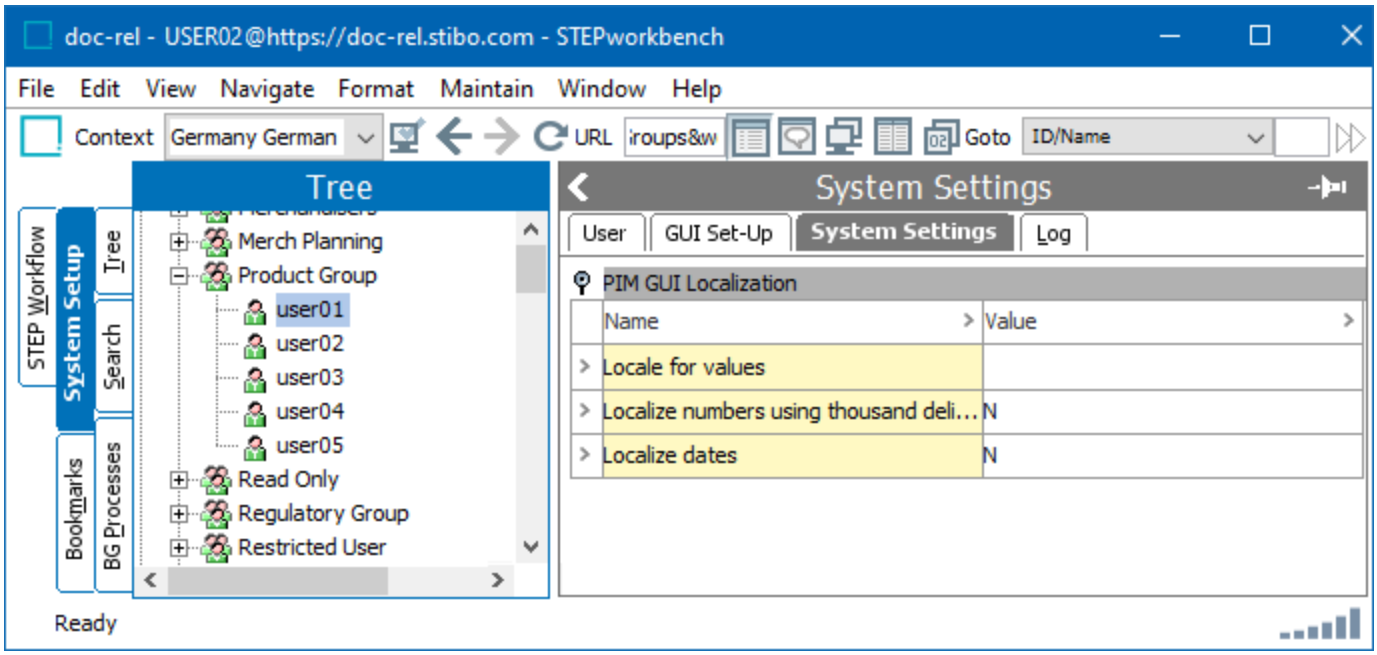
For example, User 02 is using English - us locale:



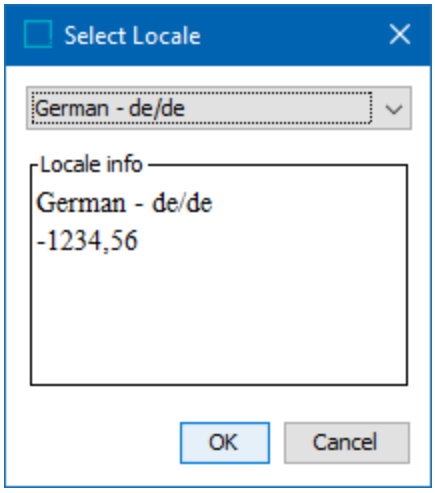
In this example User02 is logged on the Workbench. The User is setup to use a English-us locale for numbers. A dot will appear in single and multi valued attributes and LOVs. User 02 will only be allowed to use a dot as a decimal separator for numbers.

Setting up a GUI Localization for a User

GUI Localization is maintained in System Setup: **Users & Groups** > **Users** in the **System Settings** tab.



1. In **Locale for values**, click field to open a locale selector dialog. In this example a German - de/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/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/de locale is selected for User 01. For User 01, isodates will be shown as 13.05.2009 and 'ISO Date and Time' will be shown as 13.05.2009 19:10:48.

Privilege Rules

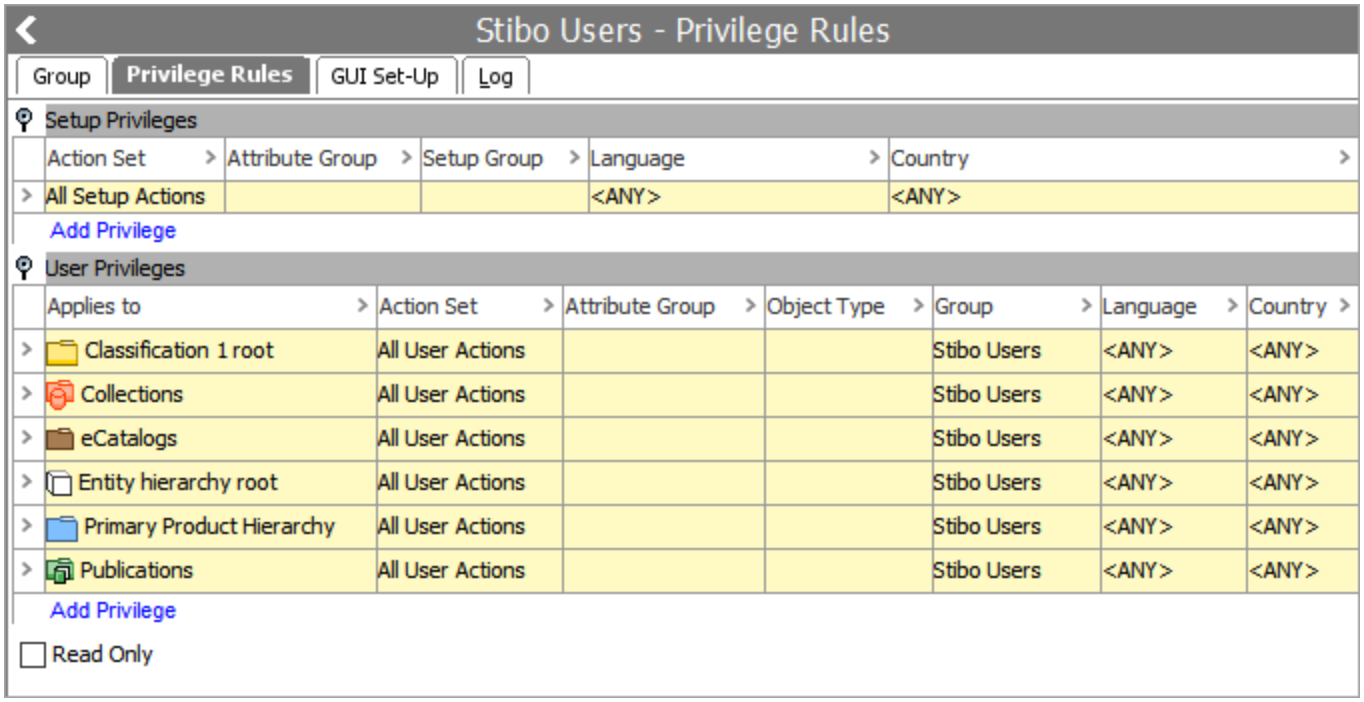
Privilege rules are maintained in System Setup within the **Users & Groups** node on individual **Groups**.

Privilege rules are rules that specify a set of permitted actions that members of a specific group can perform on a specified set of data. There is no limit to the number of privilege rules that can be applied to the same group.

Each privilege rule is defined by specifying:

- A group (of users)
- An action set
- A structure node (products, classifications, collections, publications, entities, and eCatalogs)
- A workflow state
- An attribute group (or one attribute).
- An object type
- Limitations to certain dimensions

Each group must be assigned privilege rules, defining which actions the members of that group are allowed to perform, and which data the members of the group is allowed to work with. Defining the privilege rules forms one part of the privilege control system settings. The second part, defining action sets, is performed in System Setup under Action Sets. For more information, see **Action Sets** topic. A privilege rule grants all of the users in a group the permission to perform any of the actions in the selected action set. This permission can be restricted to specific sets of data.



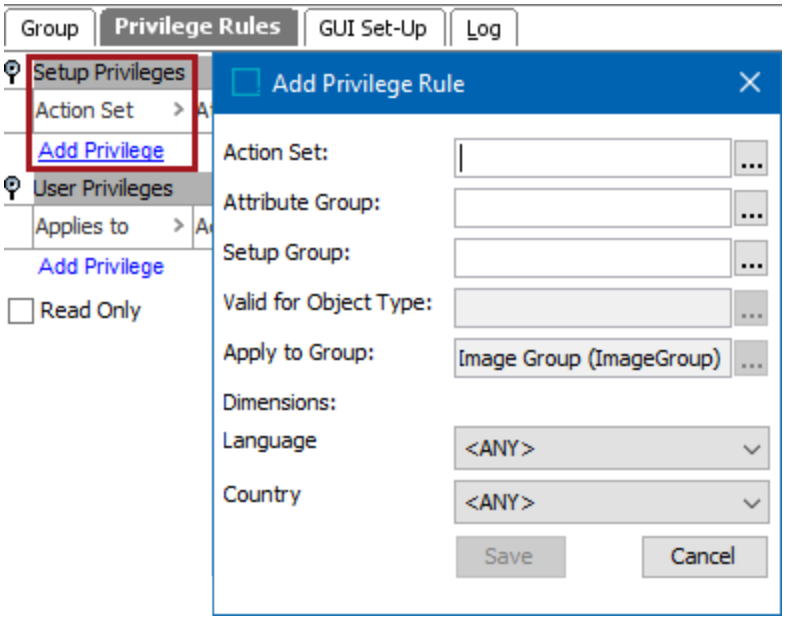
To prevent a group of users from making changes, see the Read-Only Access section below.

Important: Privilege Rules applied a User Group will be inherited to sub User Groups. This means it is possible to apply general Privilege Rules on a top level and only specify local Privilege Rules on sub User Groups. GUI setup applied a top User Group will not be inherited to any sub User Group. 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.

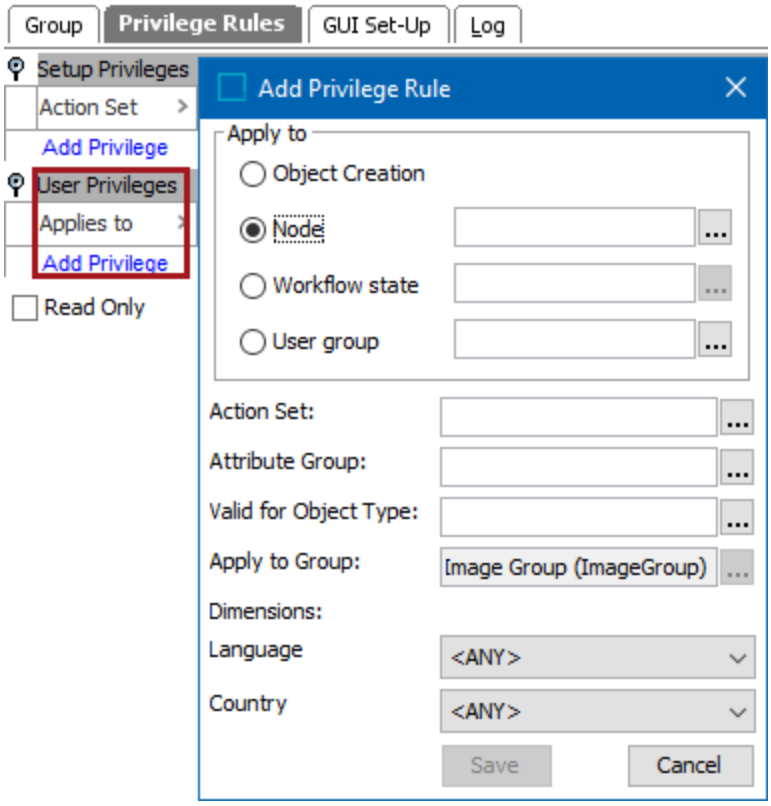
Privilege Rule Types

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, in a 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 multiple 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

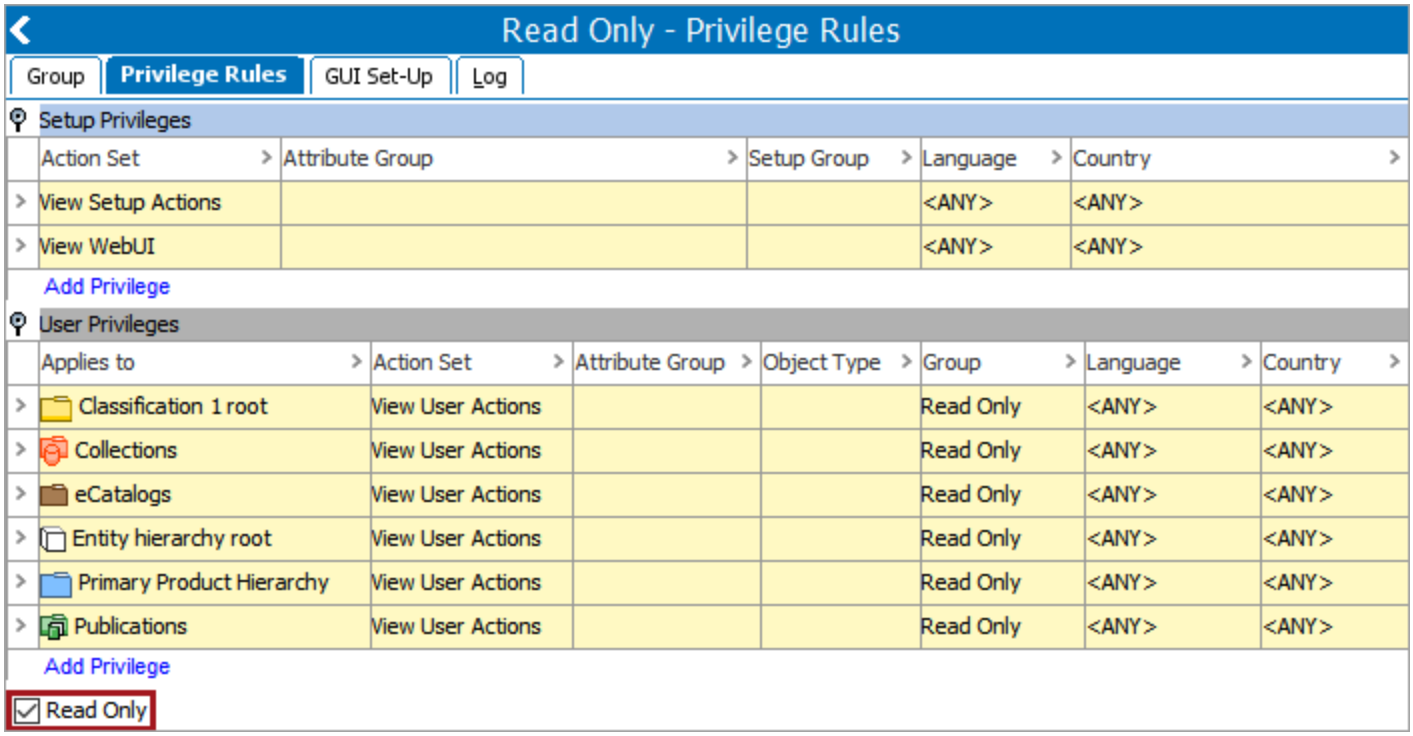
If a group has permission to work with a specific classification node, attribute values can be edited for all products linked to (or below) that node. However, access to the classification node does not grant the privilege to link new products in the product hierarchy. To do that, the group must have permission to work with the specific product node as well.

Read-Only Access

To prevent a group of users from making changes, create a group with the following privilege rules settings:

- Only 'View actions' rights are added
- The 'Read Only' checkbox is checked

For example:



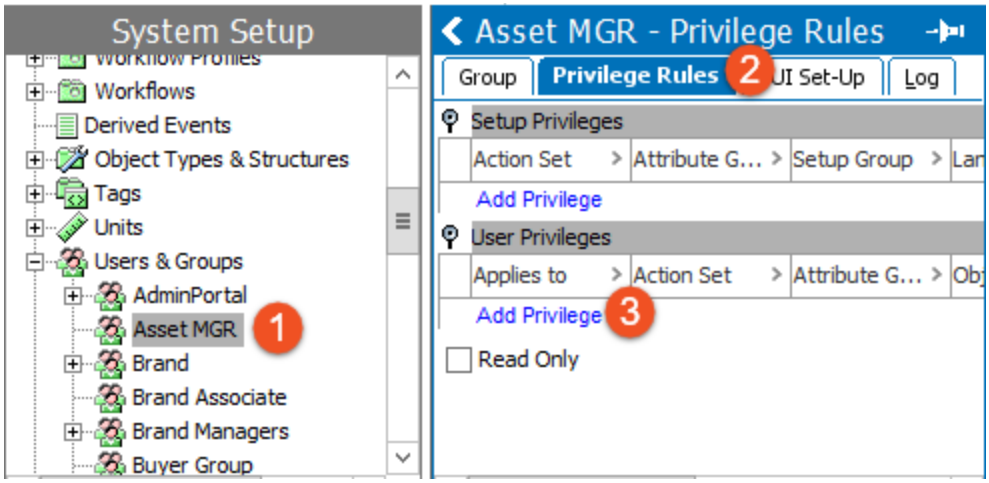
This example group provides the users with only view access; the group members are not able to make any changes while logged into the system.

Adding User Privileges for a Group

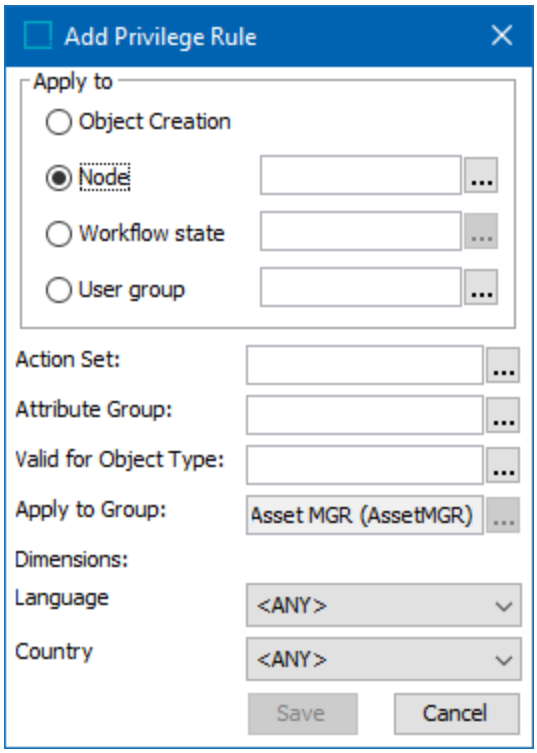
User Privileges are maintained in System Setup: **Users & Groups** > **Privilege Rules** tab.

A Privilege rule specifies a set of permitted Actions that members of a specific Group can perform on a specified set of data. There is no limit to the number of Privilege Rules that can by applied to the same Group 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, open 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**.

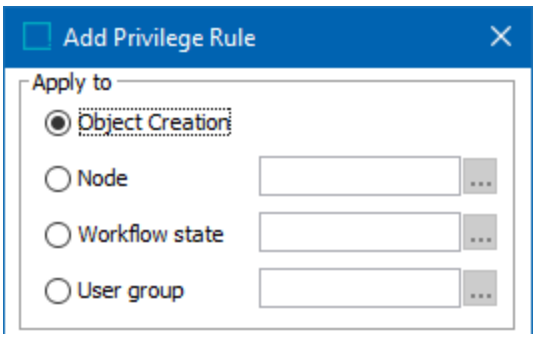


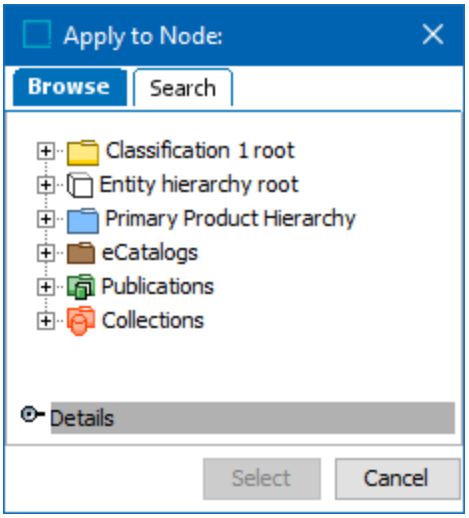
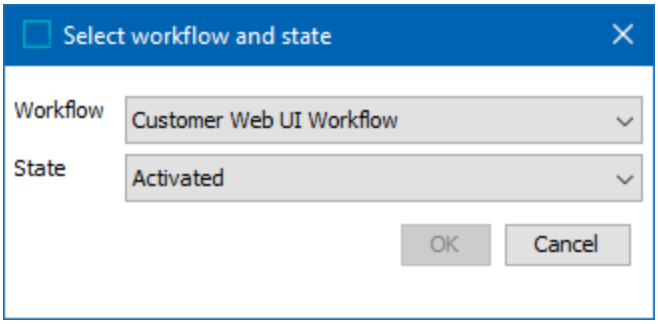
4. The **Add Privilege Rule** dialog box will appear as shown in the image below.

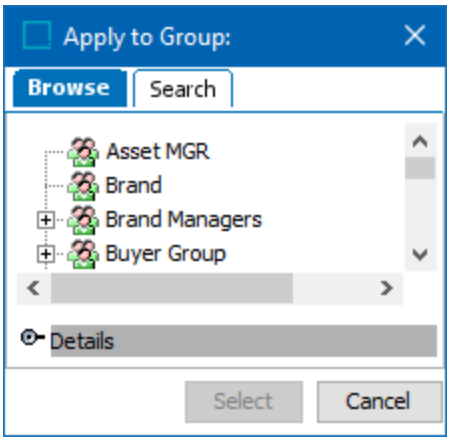
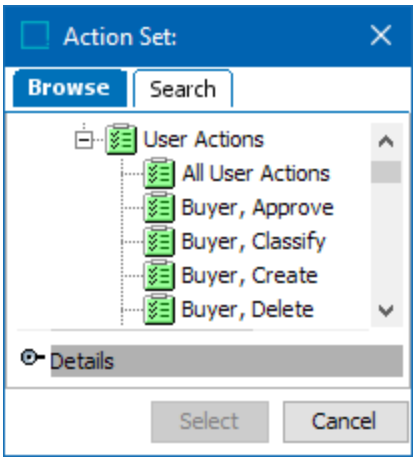


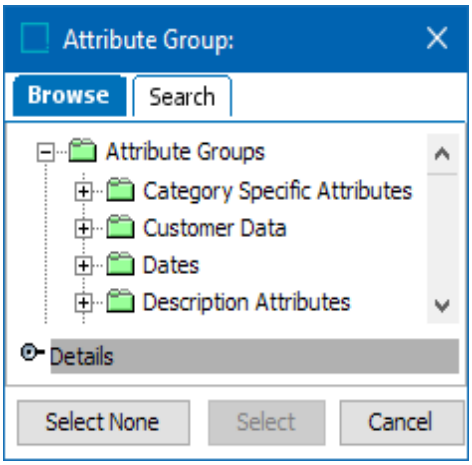
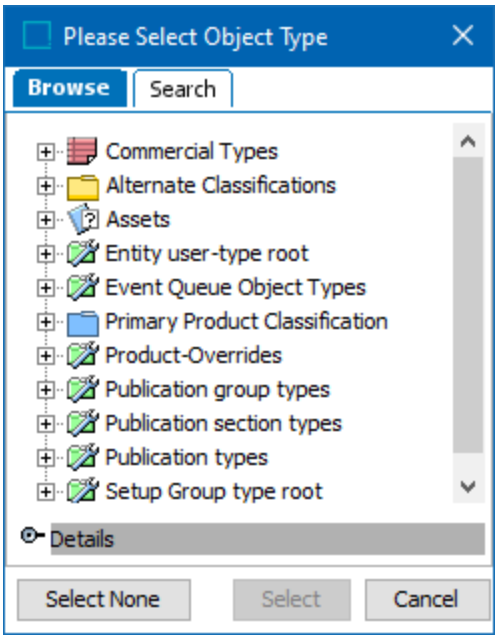
5. Make the relevant changes, and then click **Save**.

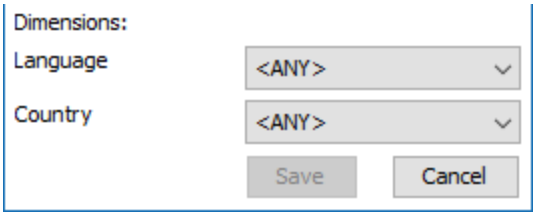
Add Privilege Rule Options

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. 

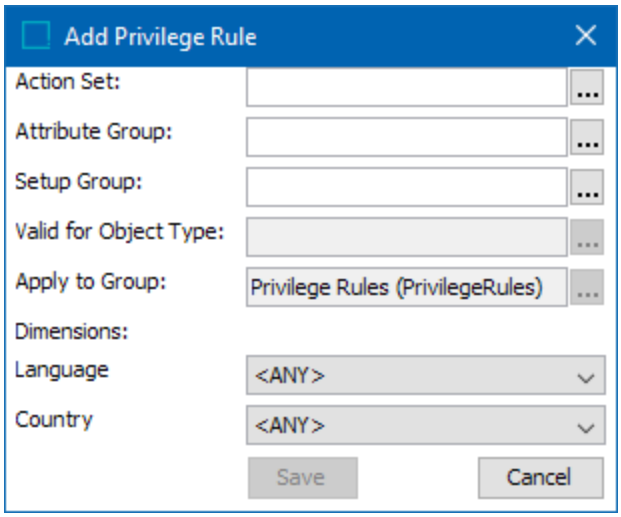
Adding Setup Privileges for a Group

Group Privileges are maintained in System Setup: **Users & Groups**: **Privilege Rules** tab.

Setup privileges cover the Actions that are performed by members of a specific group in System Setup.

1. From the System Setup menu, click the flipper to open 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**.

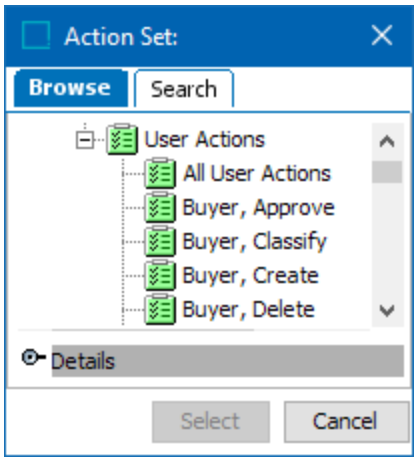
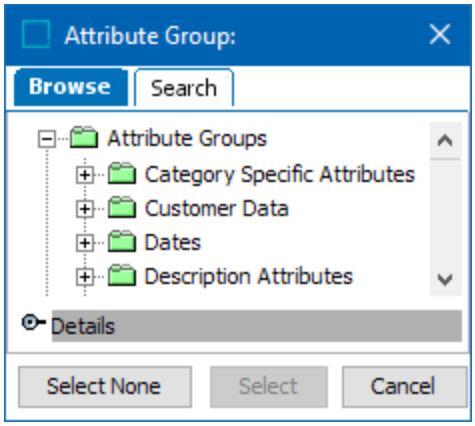
   An **Add Privilege Rule** dialog box appears.

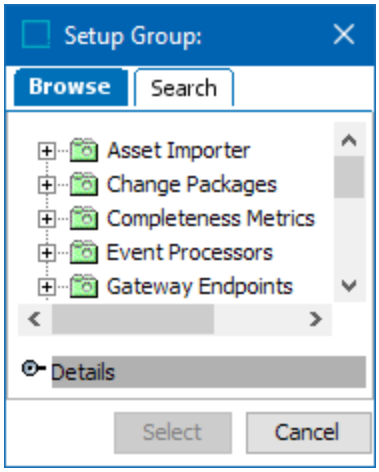


4. Make the relevant changes, and then click **Save**.

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.

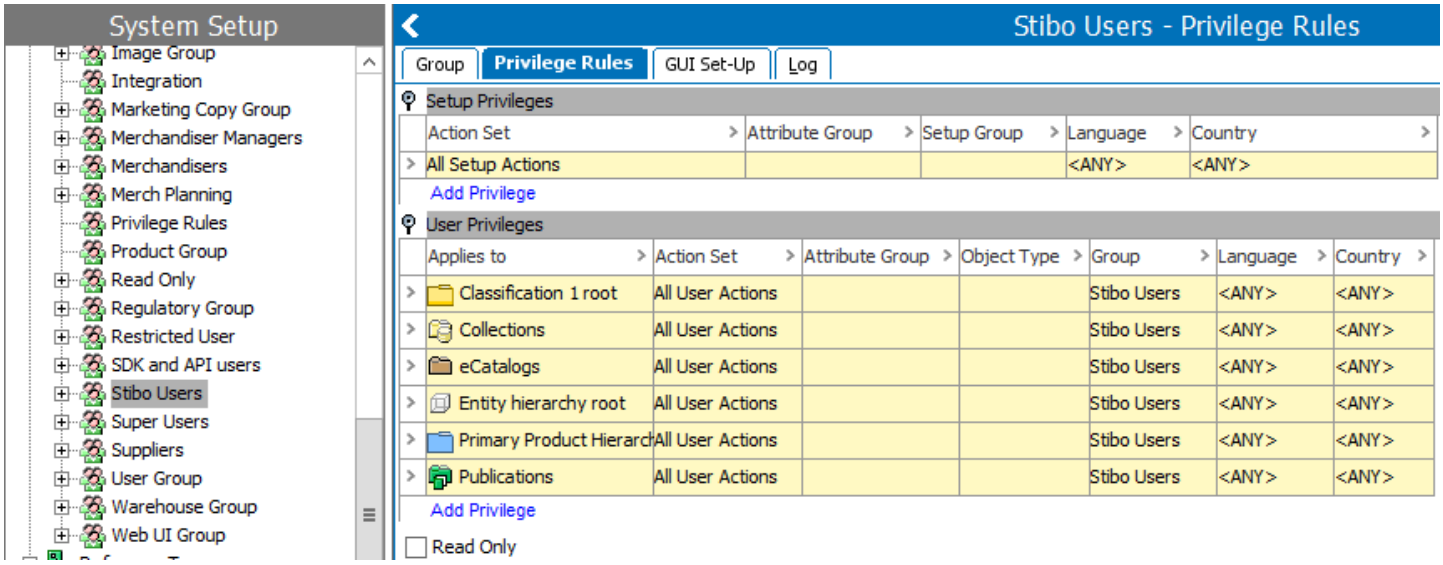
Setting	Description
	
Valid for Object type	Selecting an Object Type is not a part of a Setup privilege.
Language Country	Optional. Select the allowed Dimension to be accessed.

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**. Setting dimension points is a time consuming process. 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.

Once saved, the Setup Privilege rule is added to the User Group.

Editing Privilege Rules

Privileges are maintained in System Setup: **Privilege Rule** tab in the **User Group** Editor.



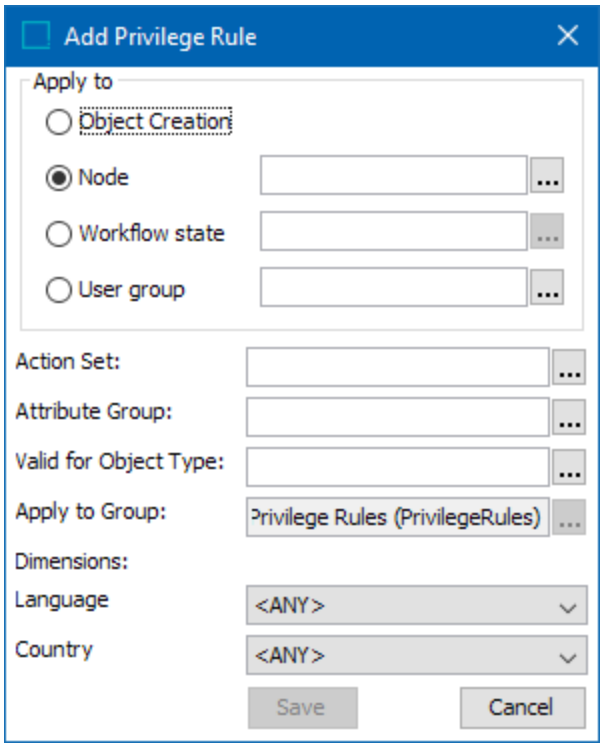
For more information about Privileges Rules and the Privileges Rule Types, see **Privilege Rules** in the **Users and Groups** documentation.

1. In System Setup, open **Users & Groups**, and then click the relevant Group.

   The Group appears in the **Group** Editor.

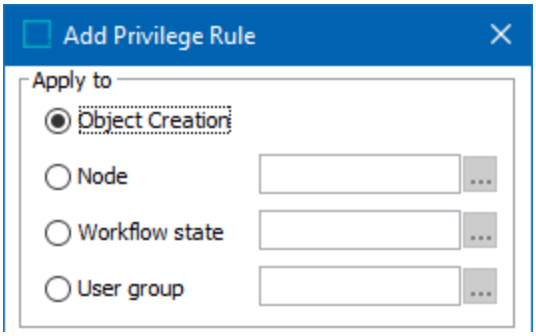
2. Click the **Privilege Rules** tab.
3. Click the relevant Privilege to be changed, right-click, and then click **Edit Privilege Rule**.

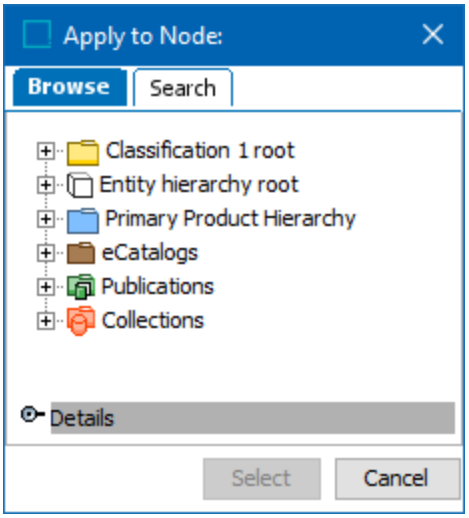
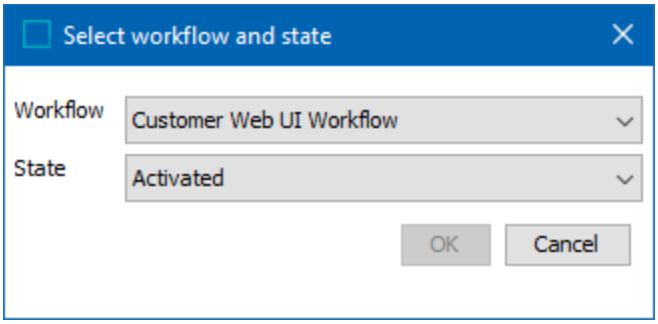
   An **Edit Privilege Rule** dialog box appears.

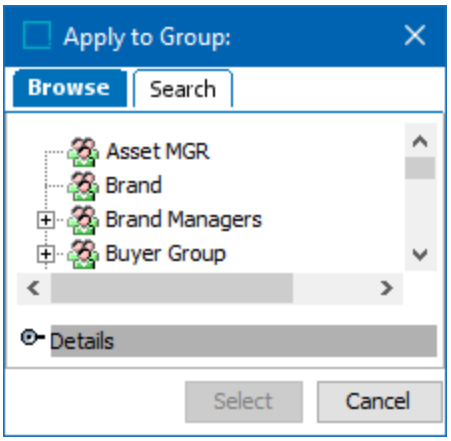
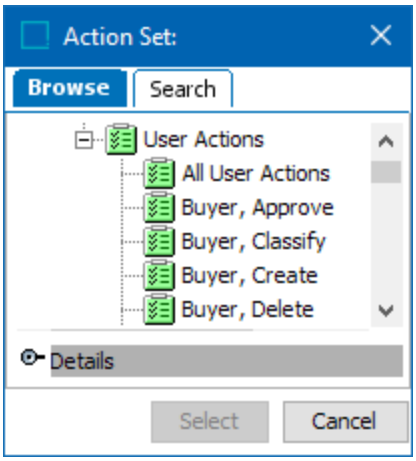


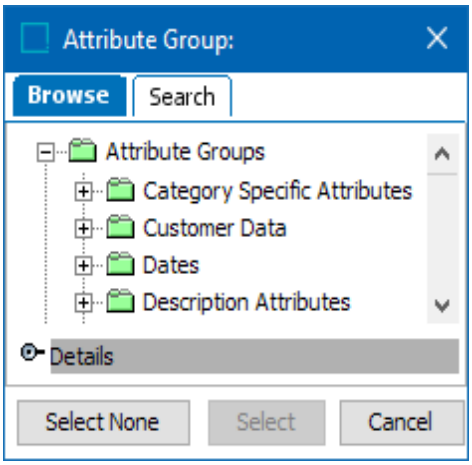
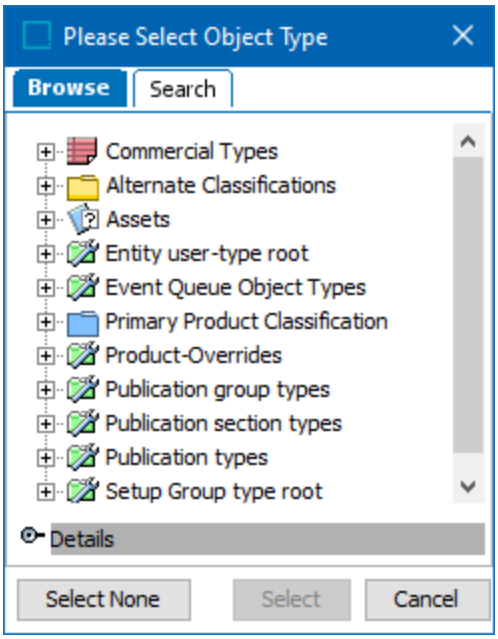
4. Make the relevant changes, and then click **Save**.

Edit Privilege Rule Options

Setting	Description
Object Creation	When configuring user privileges, the Object Creation option allows an administrator to give specified access to a user who also has privileges to create new objects in STEP. More information about this option can be found in **Object Creation Privileges**. 
Apply to Node	Select the allowed classification or product node for the group. This would include specific sub-nodes as well.

Setting	Description
	If a specific Object type and node is selected, the Users in the Group can only see Objects within that selection. e.g., a specific Classification or Product node, or Products of a certain Object type. However, if a Product not included in the selection is linked to a Classification node to which the User has access, the User can see the Product, but not edit it. 
Workflow state	When you create a user privilege rule for a STEP Workflow state, the group will have that privilege for nodes in that particular state in the STEP Workflow. More information about this parameter can be found in **Configuring Workflow Privileges**. 
User Group	Web UI user impersonation enables a privileged user to act as another user, but using their own password so that the impersonated user's password is not revealed. This is done through selecting a User Group. More information about this parameter can be found in **Web UI User Impersonation**.

Setting	Description
	
Action Set	Select the permitted Action Sets for the Group. Remember to create Action Sets that are very specific to the privileges that you want the User Group to have. For more information about Action Sets, see **About Action Sets** in System Setup. 
Attribute Group	Optional. Select the permitted Attribute Group. Select if the Group should only be allowed to work with attributes in a specific Attribute Group. Note that if are attributes within a group that should not be valid for a User to have access to, a separate group should be maintained containing the attributes the Users should have access to. 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. 

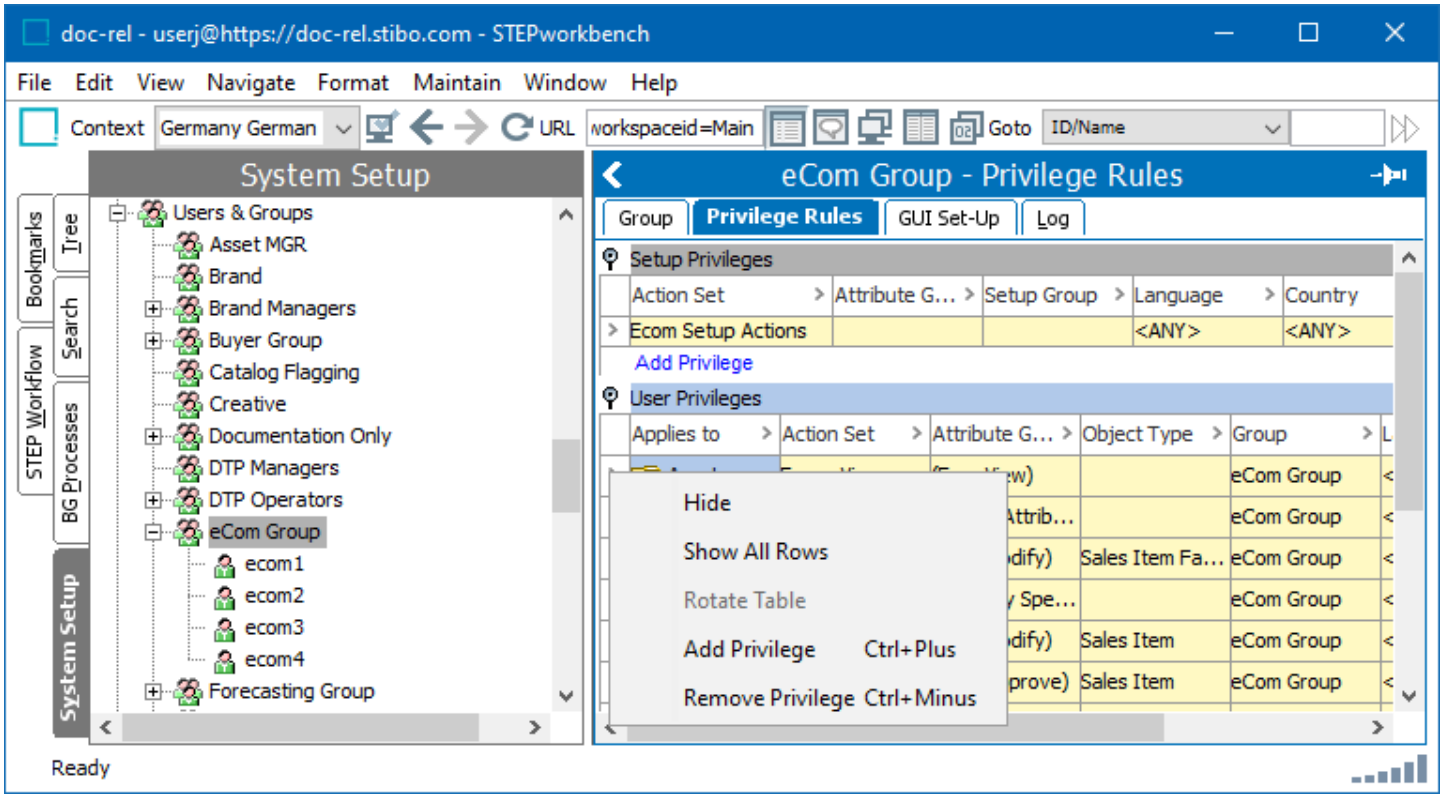
Removing a Privilege Rule from a Group

1. In System Setup, open **Users & Groups**, and then click the relevant Group.

   The Group appears in the **Group** Editor.

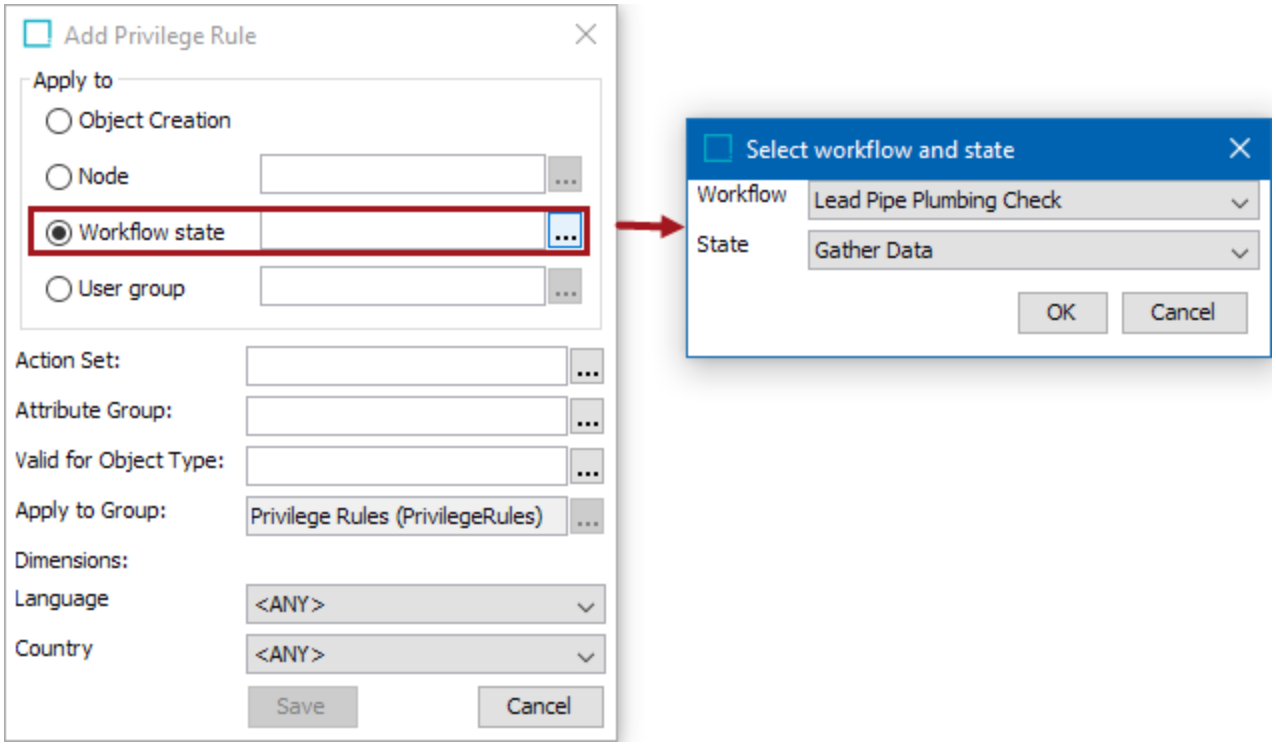
2. Click the **Privilege Rules** tab.
3. Click the row header of the Privilege rule to be removed, and then click **Remove Privilege**.

   The Privilege rule will now be removed from the list and will no longer apply to the members of the selected Group.



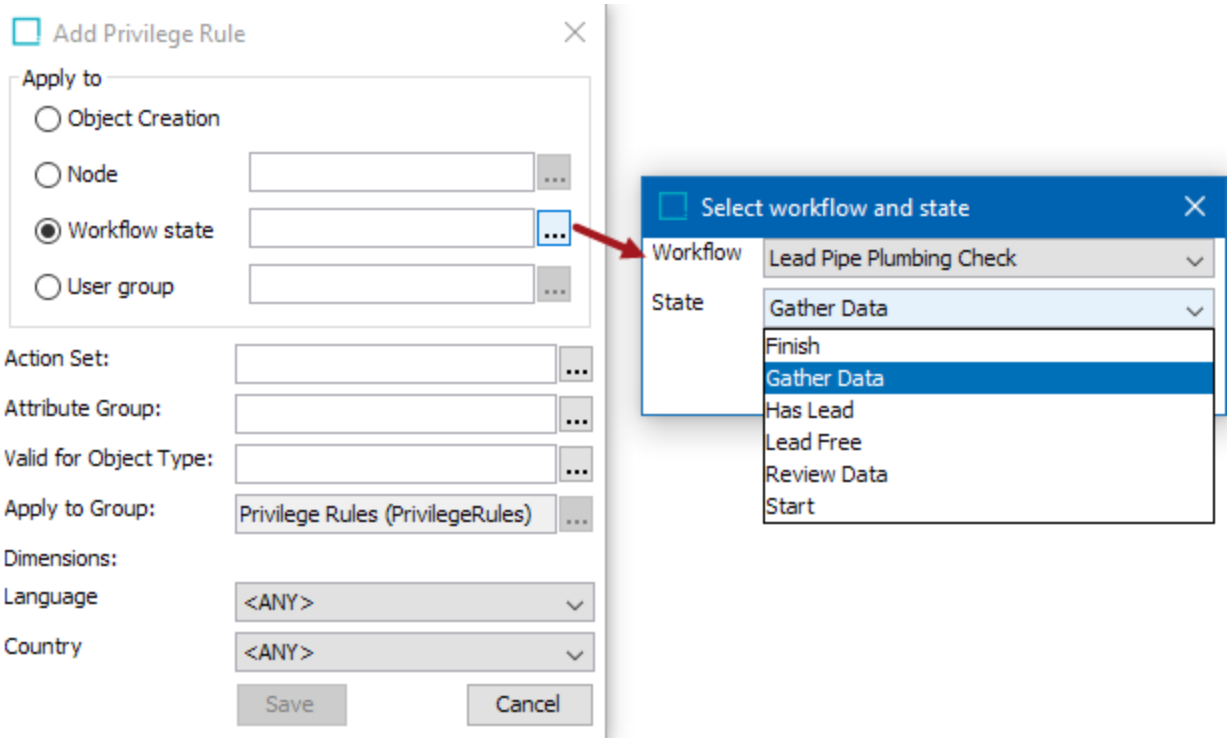
Configuring Workflow Privileges

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 click the ellipsis button (...), you will get the below STEP Workflow state selector:



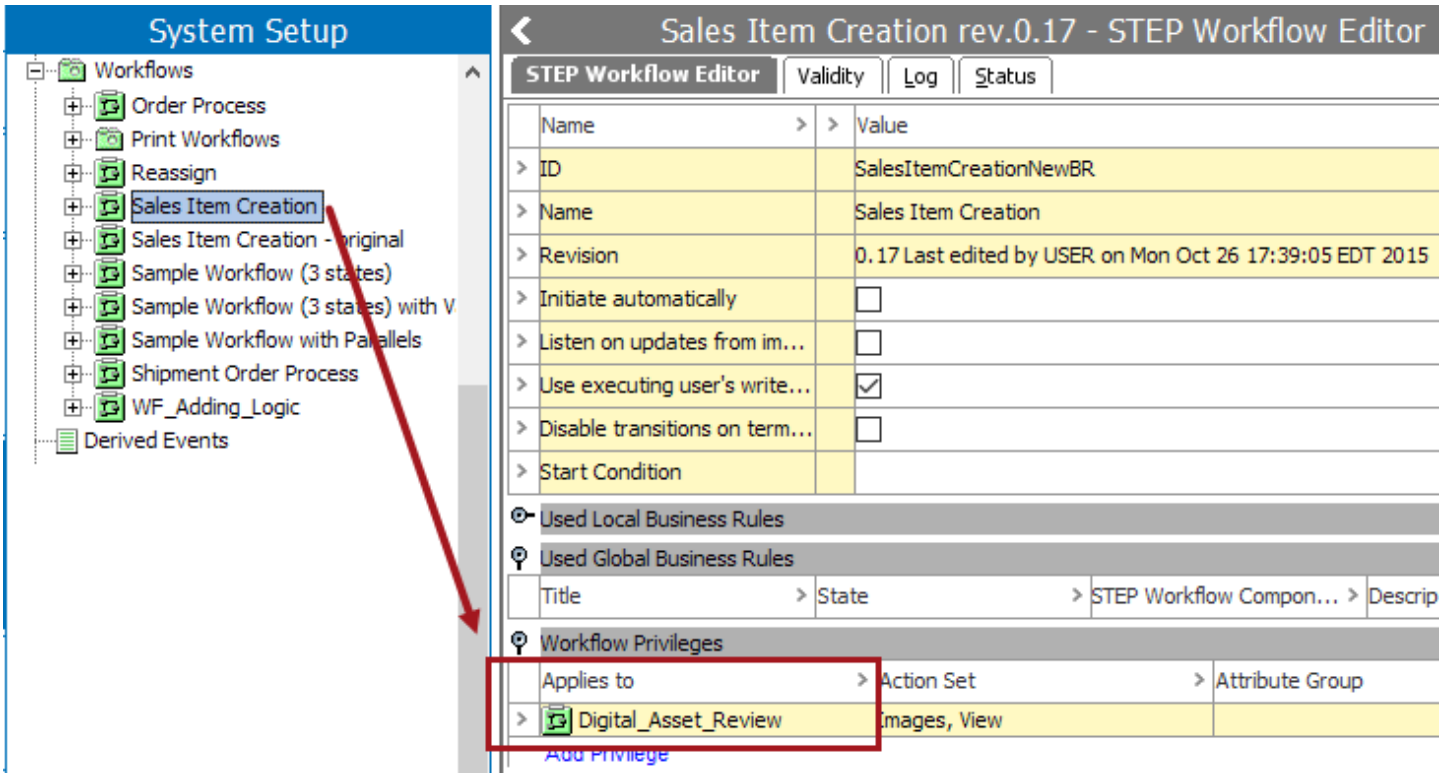
Workflow state selector

The dialog consist of two combo boxes. The top combo box allows you to select the STEP Workflow. Once the STEP Workflow has been selected the button combo box will be populated with all the states within the selected STEP Workflow. The states are arranged such that cluster and parallel states will have their child states shown below them indented. If you hover the arrow on a state, the hover text will give information about the type of the state; initial, cluster, parallel, or final.

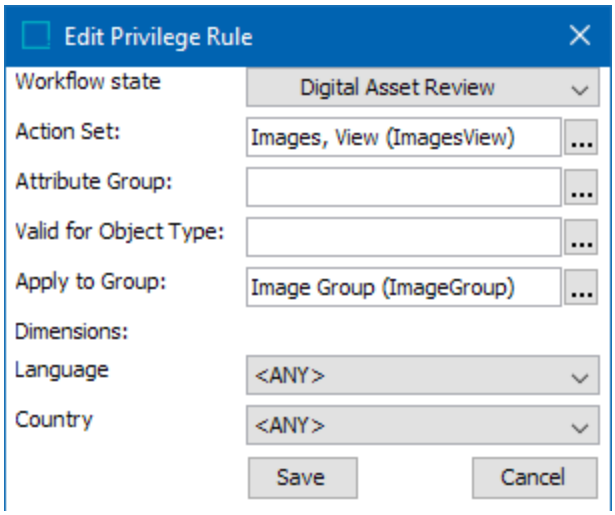


Editing Privilege Rules with focus on STEP Workflows

The above describes how to edit privilege rules with focus on a User Group. However, it is also possible to edit privileges with focus on either a product or classification hierarchy. Thus you can also find the privilege table on the "Referenced By" tab when having selected a product - or on the "References" tab when having select a classification in the Workbench. Additionally, you can set up privileges while focused on STEP Workflows by using the "Workflow Privileges" table on the STEP Workflow Editor. This shows all the privileges rules referring the selected Workflow. The first column in the table is slightly different from the privilege rules table when shown for a group, where it shows the name of the STEP Workflow followed by the name of the state the rule applies to. On the STEP Workflow Editor the "Applies to" column only displays the state - as the STEP Workflow is given from the selection.



Likewise, when editing a privilege rule or adding a new privilege rule via the STEP Workflow Editor, the dialog is slightly different. Instead of radio buttons to select either a node or a Workflow and State, a combo-box allows you to select a state of the given Workflow.



It is possible to see which states have privilege rules in the Workflow Editor. The states will be marked with an encircled "P".

Limitations on Workflow privileges

Not all actions can be granted as Workflow / State privileges. It is valid for a Workflow / State specific privilege rule to add an action set containing some of the actions that does not apply to Workflow / State specific rules. These actions will be ignored. A list of the valid actions for Workflow / State specific rules are:

- Approve Classification Product Links
- Approve Create
- Approve Data Containers
- Approve Modify
- Approve References
- Approve Values
- Classify asset
- Classify product
- Create asset reference
- Create classification reference
- Create collection
- Create data container
- Create entity
- Create entity reference
- Create product
- Create product reference
- Declassify asset
- Declassify product
- Delete asset
- Delete asset reference
- Delete classification reference
- Delete data container
- Delete entity
- Delete entity reference
- Delete product

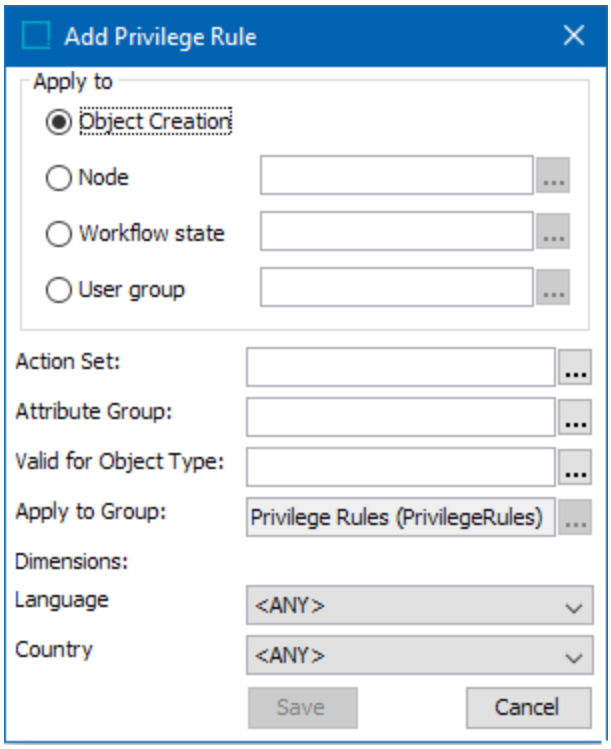
- Delete product reference
- Download asset
- Link (product) attribute into classification
- Modify metadata for asset
- Modify metadata for asset reference
- Modify metadata for attribute to classification link (also translate)
- Modify metadata for classification
- Modify metadata for classification reference (link value) (also translate)
- Modify metadata for data container (property 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
- Move classification from
- Move classification to
- Move entity from
- Move entity to
- Move product from
- Move product to
- View asset
- View asset reference
- View attribute to classification link
- View classification reference

- View Collection
- View Collection Group
- View data container
- View entity
- 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 data container (property 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
- View product reference
- Unlink (product) attribute from classification
- Upload to asset

Object Creation Privileges

In addition to configuring user privileges, the Object Creation option allows an administrator to further specify the access a user who also has privileges to create new objects in STEP. 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.

Note: The Object Creation option provides additional control for what a user can do when creating an object. Separate user action(s) that allow a user to create object(s) are also required, as defined in the **User Actions** topic.



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 cannot 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 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)

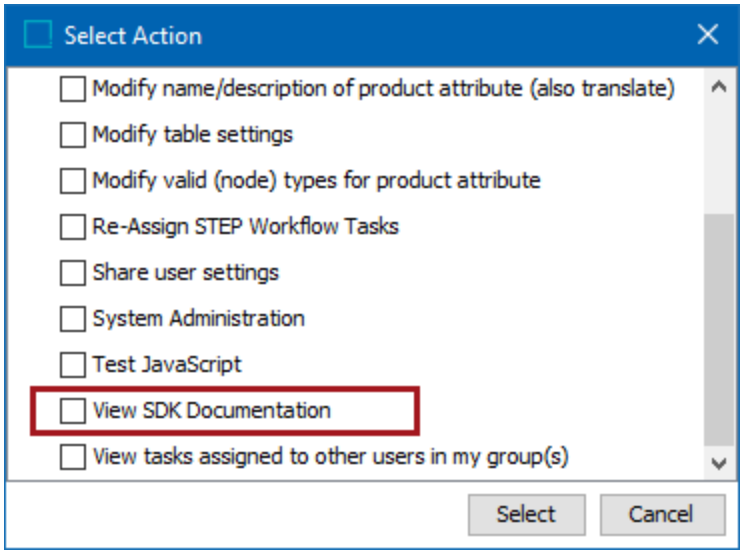
Granting Access to the SDK and API Documentation

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 Start 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.

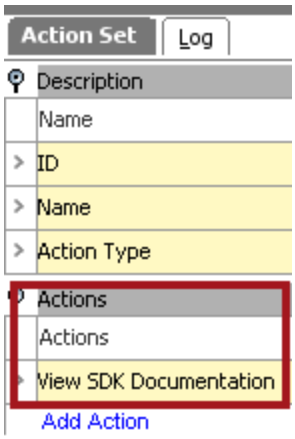
Assigning the privilege is a two-step process: First, you create an action set that includes the **View SDK Documentation** setup action privilege, and then you assign the action set to the relevant groups.

Grant Access to the SDK and API documentation

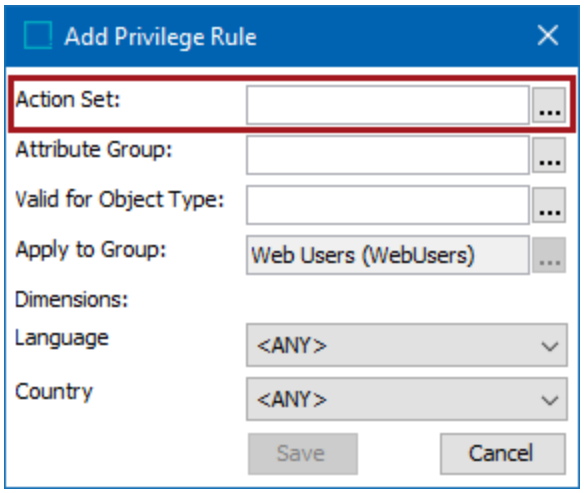
1. In **System Setup**, expand **Action Sets**, right-click **Setup Actions**, and then click **New Action Set**.
2. Enter an **ID** and a **Name** for the action set, and then click **OK**.
3. On the **Action Set** tab, click **Add Action**.
4. In the **Select Action** dialog, scroll down and select **View SDK Documentation**, and then click **Select**.



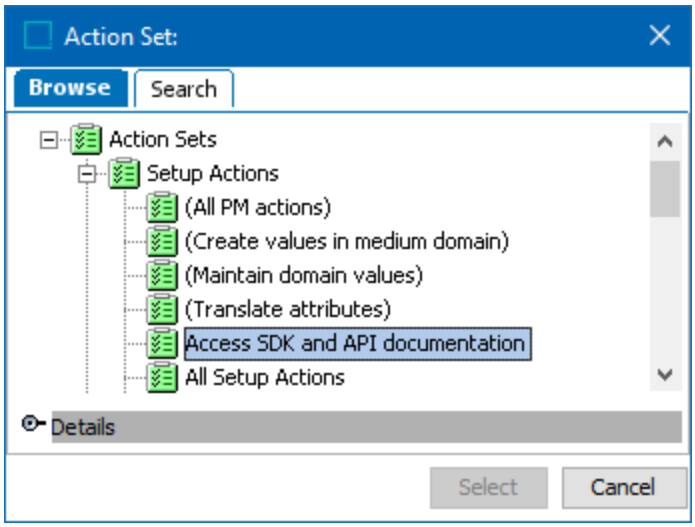
The action is now listed in the **Actions** area on the **Action Set** tab, and you can add the action set to the relevant user group.



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

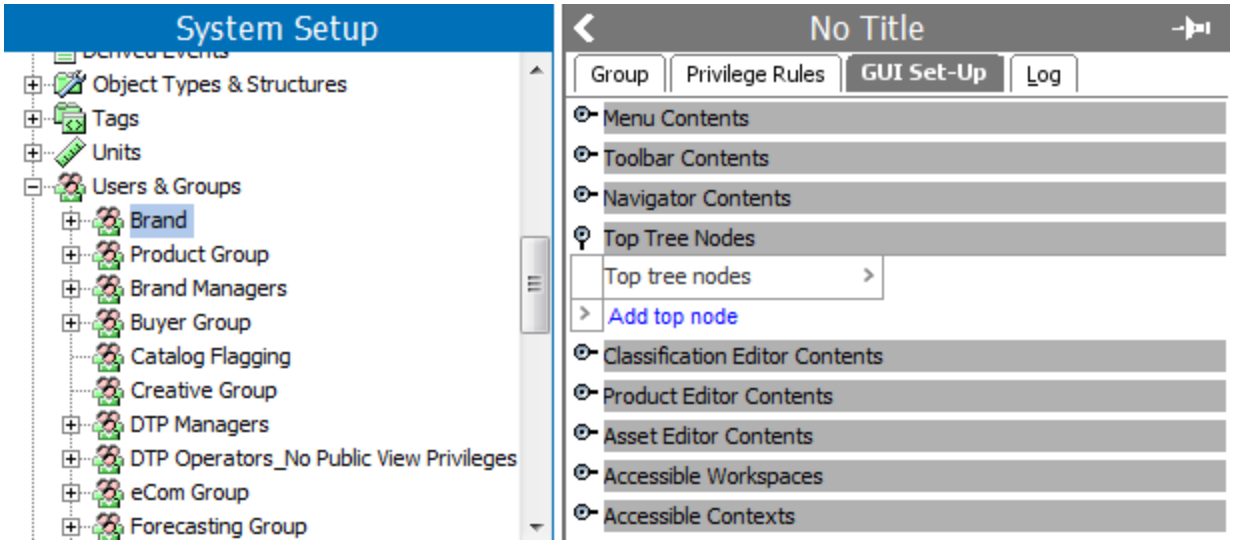
GUI Set-Up is maintained in System Setup: **Users & Groups** > **Groups** or **Users**

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**

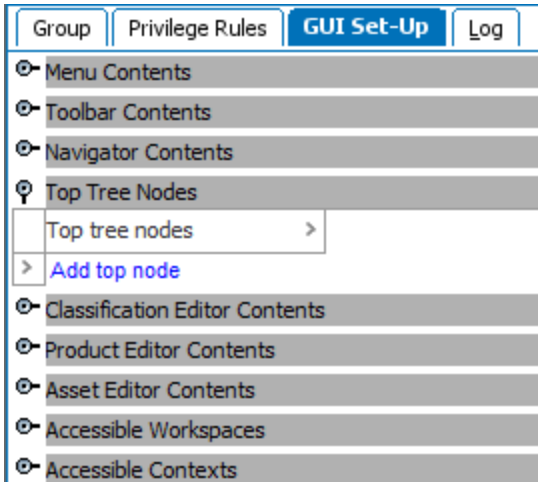
GUI Set-Up on User Groups makes it possible to customize menu contents, navigator contents, accessible workspaces and contexts and Classification, Product and Asset editor contents for Users in the Group.

1. In System Setup, open **Users & Groups**, and then click the relevant Group.

   A **Group** Editor appears.

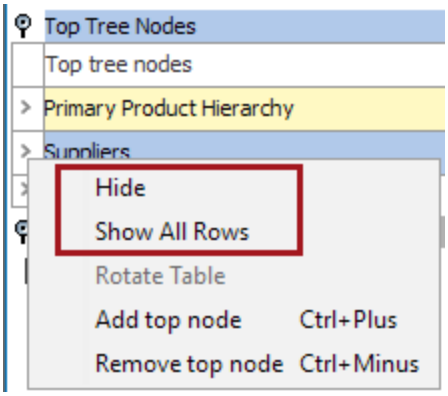
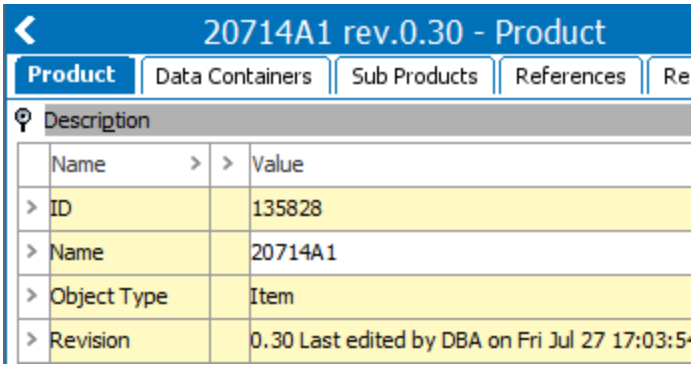
2. Click the **GUI Set-Up** tab.

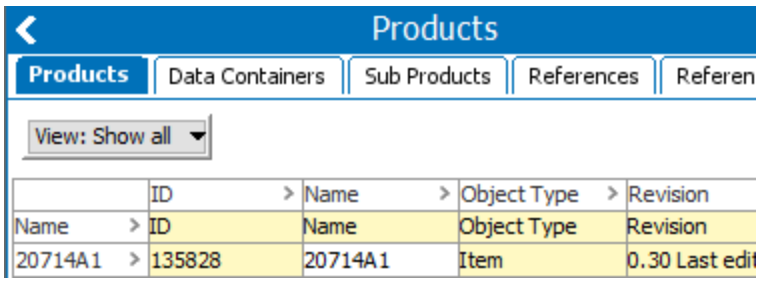
   An editor appears listing settings that can be applied on a Group.



GUI Set-Up tab flipper	Description
Menu Contents	Open this flipper to customize menu contents. Choose one of the following settings: • **'On'** means that the menu item will be shown • **'Off'** means that the menu item will not be shown • **'Advanced'** means that the menu item will be shown in a sub menu item named 'Advanced'. Advanced is typically selected for menu items that are rarely used.

GUI Set-Up tab flipper	Description
	For easy accessibility, 'Enable All' and 'Disable All' buttons are available.
Toolbar Contents	Open this flipper to customize the icons or menus to be shown in the tool panel. Choose one of the following settings: • **'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	Open this flipper 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 Choose one of the following settings: • **'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	Open this flipper to customize the tree nodes to be shown in the Tree Navigator. You can select any top node for Classification, Products, Collections, etc. Once a top node is selected, if you want to prevent other users from making changes to it, you can 'Hide' the selection. To make it visible again, select 'Show All Rows.'

GUI Set-Up tab flipper	Description
	
Classification Editor Contents	Open this flipper to customize the fields and tabs to be shown in the Classification Editor.
Product Editor Contents	Open this flipper to customize the fields and tabs to be shown in the Product Editor. After changing the following **Product** options, you must close STEP and log in again to see the updates. • **Use normal product editor** displays the standard editor for the selected object.  • **Use view** displays data using the selected multi-object view for a single object.

GUI Set-Up tab flipper	Description
	
Asset Editor Contents	Open this flipper to customize the fields and tabs to be shown in the Asset Editor.
Accessible Workspaces	Open this flipper to limit the number of Workspaces that can be viewed and accessed.
Accessible Contexts	Open this flipper 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 nor web services. If the user has privileges to access a product hierarchy which is restricted by the GUI Set-up, the user 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**.

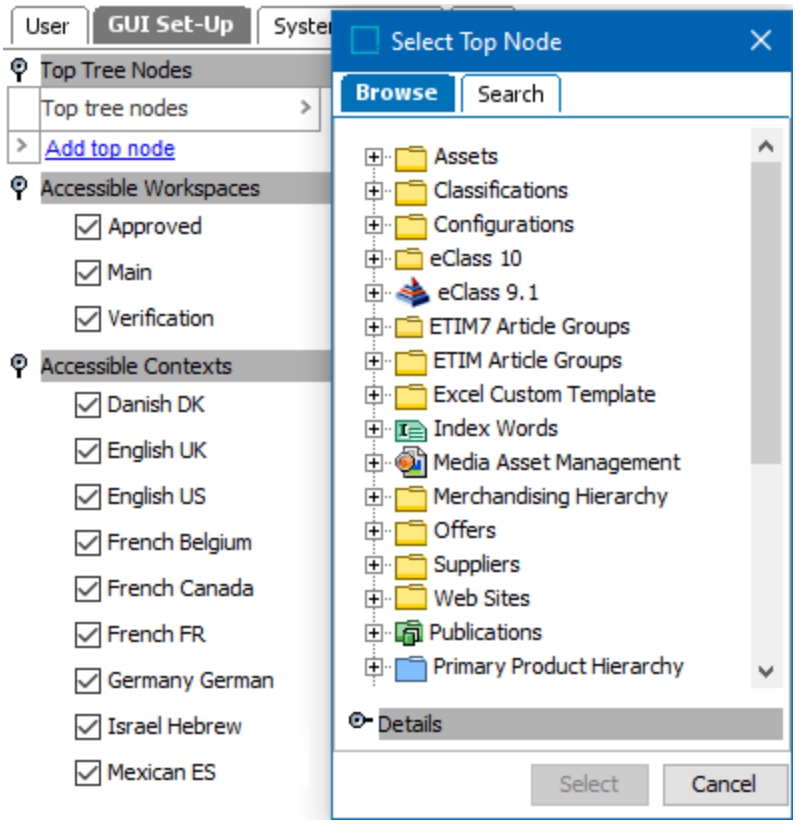
GUI Set-Up for a User makes it possible to customize top tree nodes for the user and provides an administrator with a way to make relevant workspaces and contexts accessible.

1. In System Setup, open **Users & Groups**, open 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.

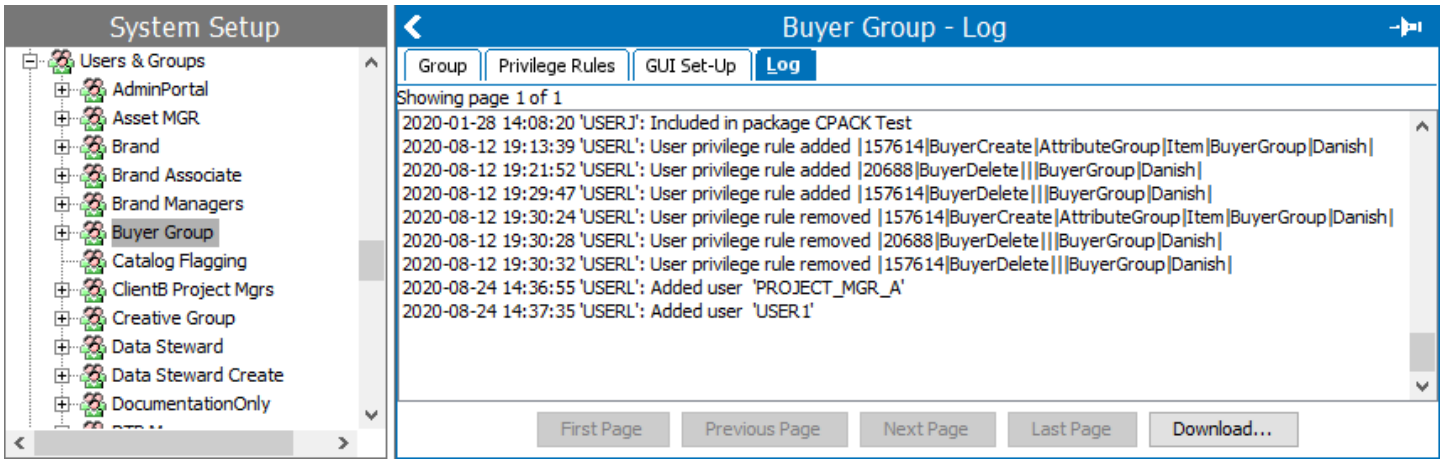
Log for Users and Groups

The Log tab on Users and Groups records any changes that occur to a particular user group or a specific user. This information is essential for audit purposes, as well as aiding data stewards and/or system architects in setting up and maintaining privileges. The information recorded uses IDs for all data recorded. The recorded information for each data point is the following:

- Date the changed occurred
- Time of change
- Who made the change
- What was updated or changed

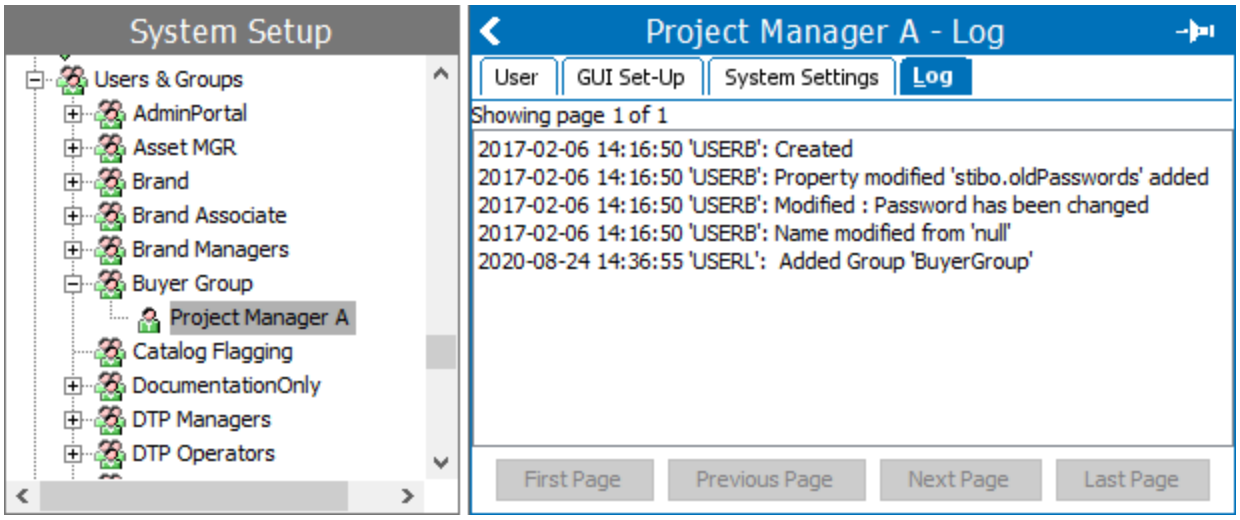
Some examples of information points that the user group Log tab holds are:

- Adding the group to a Change package
- Adding or removing privilege rules
- Adding or removing users to the group.



Some examples of information points that the individual user Log tab holds are:

- When the user was created
- Any property modifications
- Password changes
- User groups the user is added to or removed from


System Setup
Users & Groups
AdminPortal
Asset MGR
Brand
Brand Associate
Brand Managers
Buyer Group
Project Manager A
Catalog Flagging
DocumentationOnly
DTP Managers
DTP Operators
Project Manager A - Log
User
GUI Set-Up
System Settings
Log
Showing page 1 of 1
2017-02-06 14:16:50 'USERB': Created
2017-02-06 14:16:50 'USERB': Property modified 'stibo.oldPasswords' added
2017-02-06 14:16:50 'USERB': Modified : Password has been changed
2017-02-06 14:16:50 'USERB': Name modified from 'null'
2020-08-24 14:36:55 'USERL': Added Group 'BuyerGroup'
First Page
Previous Page
Next Page
Last Page

Web UI User Impersonation Setup and Reporting

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.

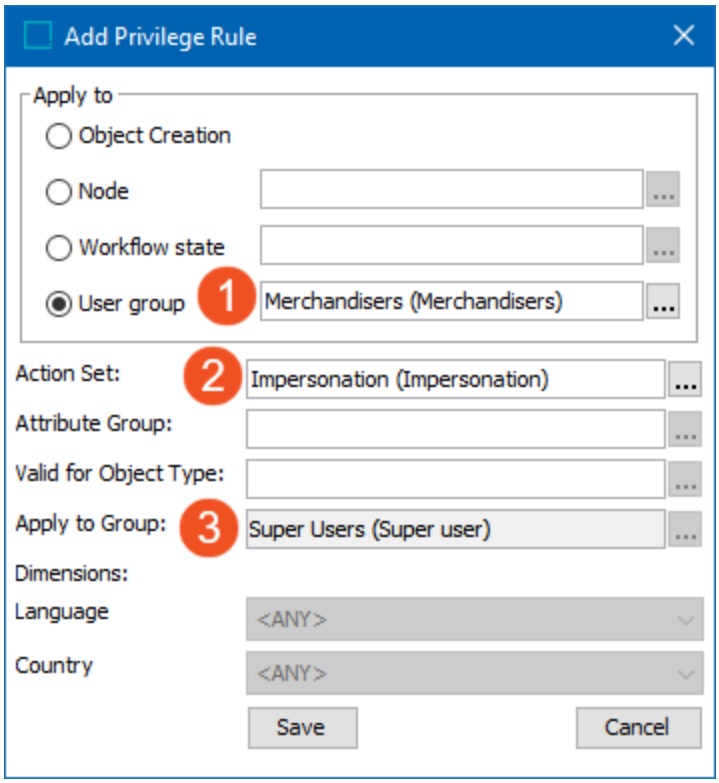
Specific privileges have been created to support impersonation:

- 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.

Once those actions are set up in an action set, they can be applied to designated user groups. Refer to the **Action Sets** section of the **System Setup / Super User Guide** documentation for setting up and maintaining action sets.

In System Setup, go to Users & Groups and click on the group that can impersonate. On the Privilege Rules tab in the editor, click 'Add Privilege" under User Privileges and fill in the dialog as follows. See the **Privilege Rules** topic in the **System Setup / Super User Guide** documentation for additional information.

1. Target (group being impersonated)
2. Action set with Impersonate User privilege
3. Source (group that can impersonate)



Note: Impersonation can only be used in Web UI.

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:

- Select User To Impersonate is an option within the Global Navigation Panel > User Settings
- Impersonate User Widget can be inserted in the Homepage
- Corner Bar Impersonate User component can be inserted in the Corner Bar, noting that the Corner Bar is superseded by the Global Navigation Panel.

The impersonation components are only visible if the user is a member of a user group with impersonation rights for another group. For demonstration of the first two methods, see the **User Impersonation** topic in the **Using Web UI** section of the **Web User Interfaces / Web UI Setup and User Guide** documentation.

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.

Impersonation edits are indicated as performed by the authenticated user (impersonator) instead of the previous way of doing it, which was to log transactions by the impersonated user. While Web UI impersonation is intended to be used for troubleshooting purposes, changes to existing data may be made in accordance with the impersonated user's privileges. Therefore, the new way of logging user actions presents an accurate audit

history, allowing for differentiation of user activity in log files and in other areas in Web UI (and workbench) that track revision history and user actions. 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: For pre-10.1 impersonation: the logs, revision histories, etc., will continue to reflect the previous way of tracking impersonation users, which is that all actions made by the impersonator are logged as actions taken by the user being impersonated. Hence, consistency between old and new should not be expected.

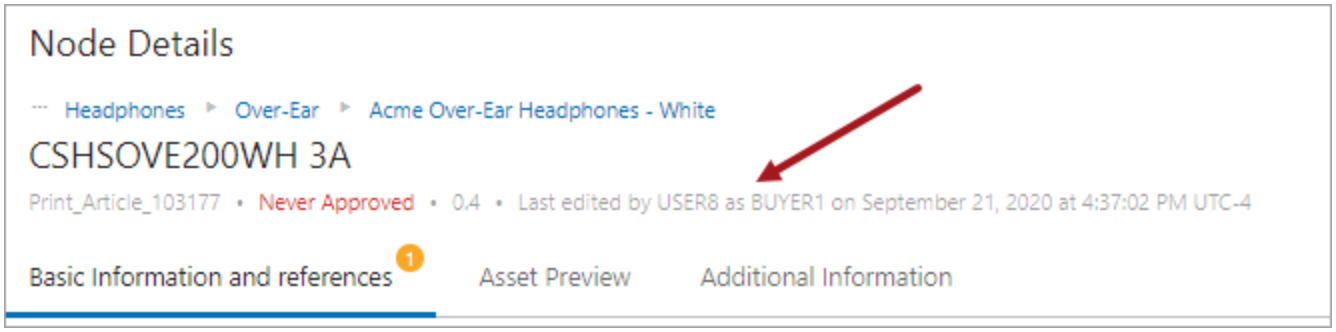
If you want to revert to the old way of tracking impersonation, the `Impersonation.RunAsImpersonatedUser=true` must be added to the sharedconfig.properties file. The impersonation functionality will switch to the impersonated user instead of a system-generated impersonation user. **Warning: All actions taken during impersonation will be performed and logged as the actual impersonated user, which will impact auditability when identifying data changes.**

Whenever a user starts impersonating another user, a user with the name <'Impersonator User ID> as <Target User ID>' and the ID, which will begin with the 'STIBO.I' prefix, will be created. The audit trail will then indicate that it is this new user revising objects or making other changes. These special user types are hidden from all browse, search, and assignment interfaces and will not count against the licensed number of users. The Admin Portal's user activity report fills this void in visibility and is shown in the first image below (as well as some other examples).

Admin Portal > User Activity Report:

A	B	C	D
User ID	User Name	Interface	Latest Activity
USER7	User 7	bgp	2020-09-21T11:05:33-04:00[America/New_York]
USERJ	User J	workbench	2020-09-21T11:05:33-04:00[America/New_York]
STIBO.I.74BE24A09843B2A48ACFAF82ED17E5CF	USER8 as BUYER1	web ui	2020-09-21T11:05:39-04:00[America/New_York]
USER8	User 8	workbench	2020-09-21T11:05:10-04:00[America/New_York]
STIBO.I.444E7C50B8AE6CEFE84A9261F1BBAB85	USER8 as SUPPLIER1	web ui	2020-09-21T10:38:41-04:00[America/New_York]

Node Details > Summary Cards:



Multi Revision Screen:

	ID	Name	Approval	Revision
0.3 (CURRENT)	Print_Article_103243	CSHSOVE200BLUE	×Never Been Approved	0.3 Created by USER8 as USER5 on Wed Oct 21 15:52:23 EDT 2020
0.2	Print_Article_103243	CSHSOVE200BLU	N/A	0.2 Created by User 4 on Mon Dec 30 16:25:20 EST 2019
0.1	Print_Article_103243	CSHSOVE200BLU	N/A	0.1 Created by User 4 on Wed Dec 18 20:36:45 EST 2019

Revision History:

Value history

Revision	Edited on	Edited by	Value
0.3	2020-10-21 16:22:49	USER8 as USER5	These headphones...
0.1-0.2	2019-12-18 20:36:45	User 4	

Important: Workbench audit history shows user IDs, so you will only see the STIBO.I user when looking at revision history in the workbench. The STIBO.I prefix indicates that the change was made under impersonation, and prompt you or other users to view the Admin Portal User Activity Report to uncover exactly who made the change.

Product | Data Containers | Sub Products | References | Referenced By | Images & Documents | Commercial | Tables | Proof View | Status | State Log | Tasks

Revisions

Revision	Created	Edited	User	Comment
0.5	Wed Oct 14 15:47:08 EDT 2020	Wed Oct 14 15:53:36 EDT 2020	USER8	Auto Generated
0.4	Mon Sep 21 16:37:02 EDT 2020	Mon Sep 21 16:37:02 EDT 2020	STIBO.I.74BE24A09843B2A48ACFAF82ED17E5CF	Auto Generated
0.3	Mon Apr 06 10:13:11 EDT 2020	Mon Apr 06 10:13:11 EDT 2020	USER8	Auto Generated
0.2	Mon Dec 30 16:25:20 EST 2019	Mon Dec 30 16:25:20 EST 2019	USER4	Auto generated - elapsed time
0.1	Wed Dec 18 20:36:45 EST 2019	Wed Dec 18 20:36:45 EST 2019	USER4	

For more information on setting up components to access impersonation in Web UI, see the following topics:

- **Homepage Widgets** (see the **Impersonate User Widget** section)
- **Global Navigation Panel** (see the **Main Properties** section)

For information on accessing and using impersonation, see the following:

- **User Impersonation** (see the **Using Web UI** documentation)
- **Corner Bar Child Components** (see the **Corner Bar Impersonate User** section)

Important: The Corner Bar Component and its child components have been superseded. The information linked above is strictly for legacy systems that have the functionality already configured.

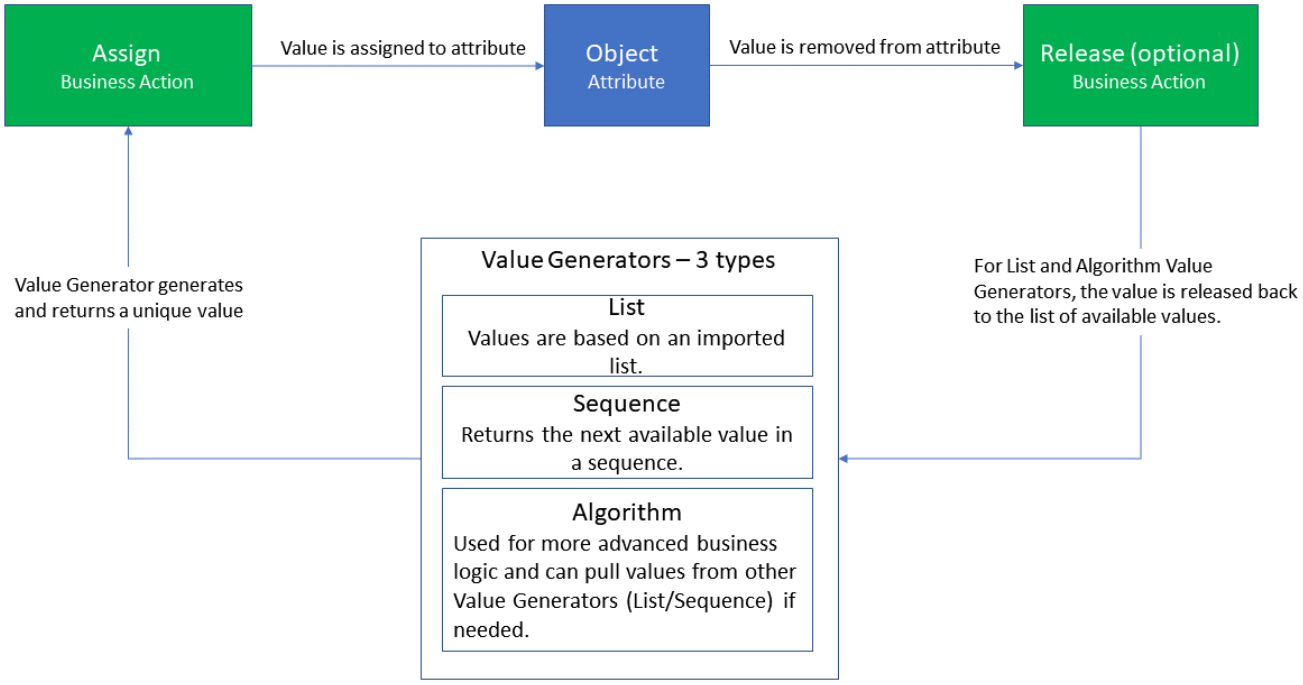
For more information on how impersonation works with the Analytics integration, see the **Embedded Analytics Platform** topic in the **Analytics** section.

Value Generators

Value Generators allow users to create and hold unique values (often in the form of numbers) for an object's attributes so that these values can be used as identifiers. The ability to generate these unique identifiers is especially helpful in ensuring ease of integration and data synchronization when communicating with external systems.

The Value Generator Configuration Type allows users with the appropriate privileges to create several different value generators:

- **Algorithm Value Generator**
- **List Value Generator**
- **Sequence Generator**

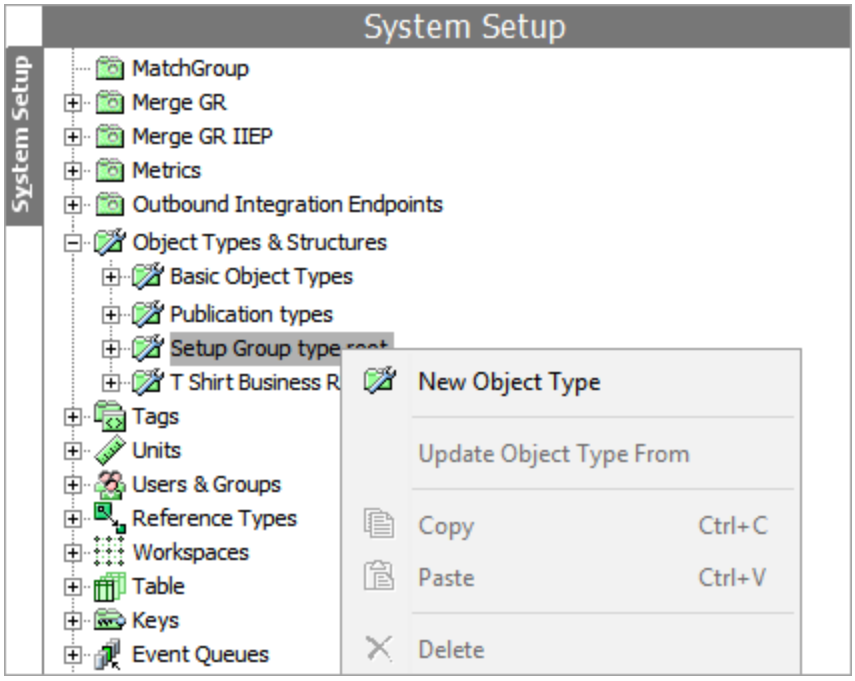


Note: To access Value Generators, the 'value-generator' add-on component must be added to your system. Additionally, users must have the 'Maintain Value Generator' user privilege in order to configure and maintain Value Generators. Instructions for installing components can be found in the **SPOT Program** topic in the **System Administration** documentation. For more information regarding user privileges, see the **Privilege Rules** topic in the **System Setup** documentation.

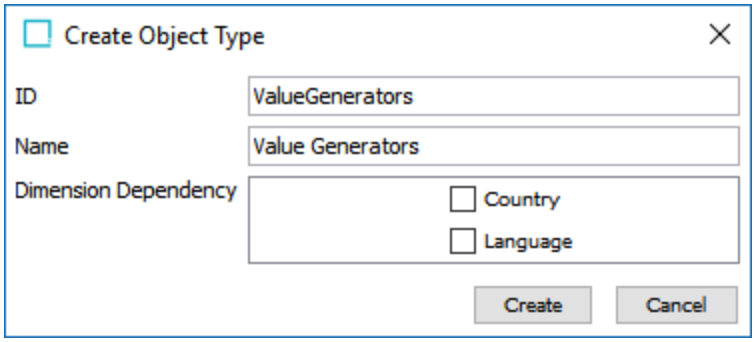
Creating a Value Generator Configuration Type

A Value Generator Configuration Type must be created before a specific value generator can be configured.

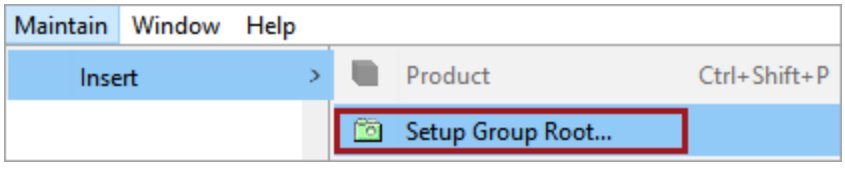
1. In the System Setup, right-click 'Setup Group Type Root' and select 'New Object Type.'



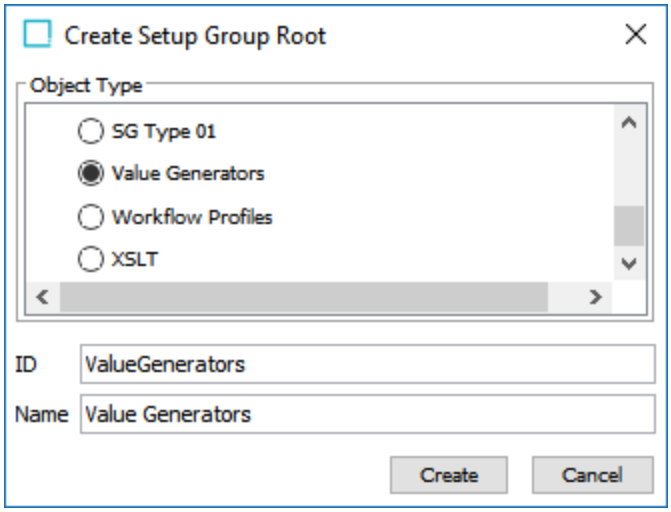
2. In the Create Object Type dialog box, enter a value for the ID and Name text field. In the example below, the user has created an object type with the ID 'ValueGenerators' and the Name 'Value Generators.'



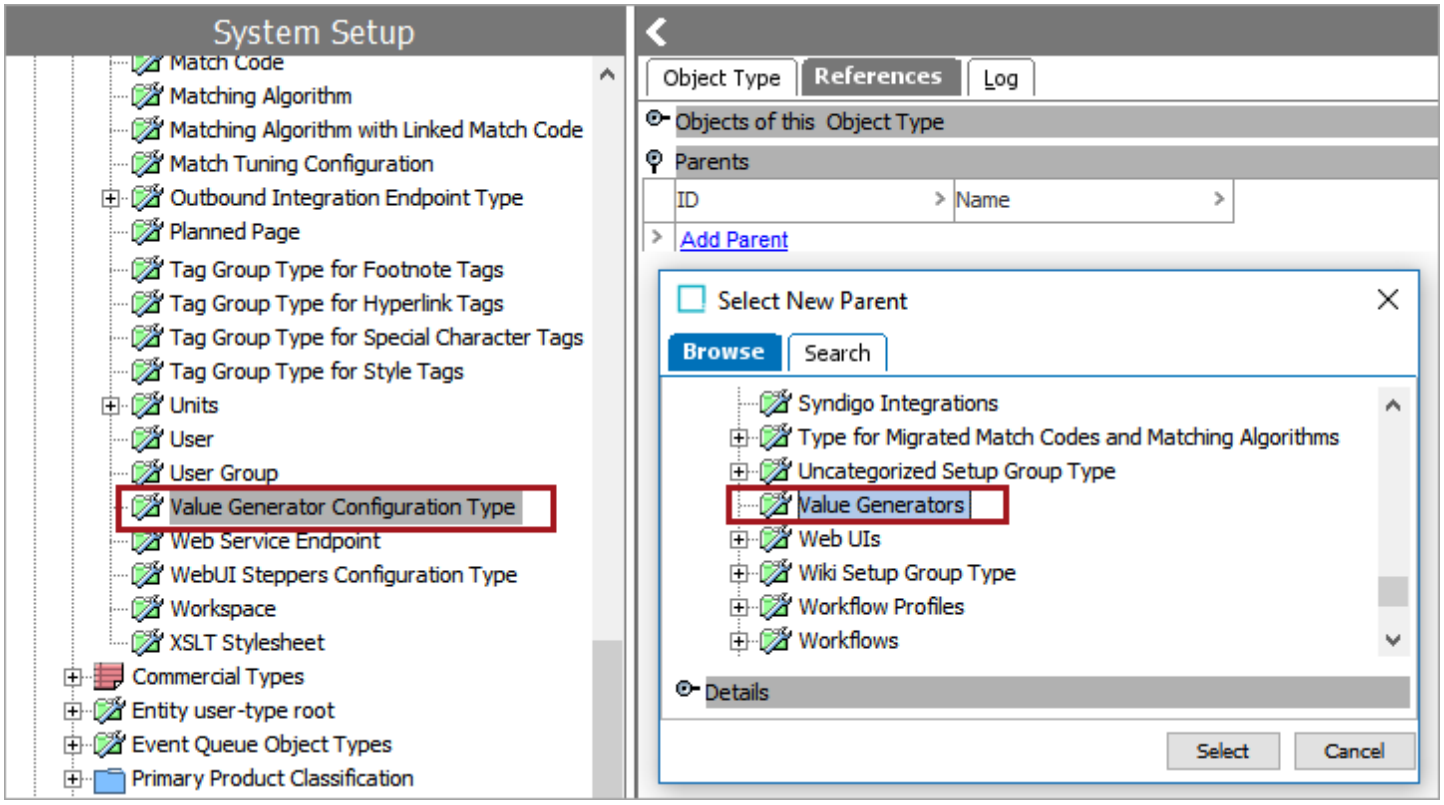
3. Click 'Create' to save the settings and close the dialog.
4. In the toolbar menu, click 'Maintain,' select 'Insert,' and click 'Setup Group Root....'



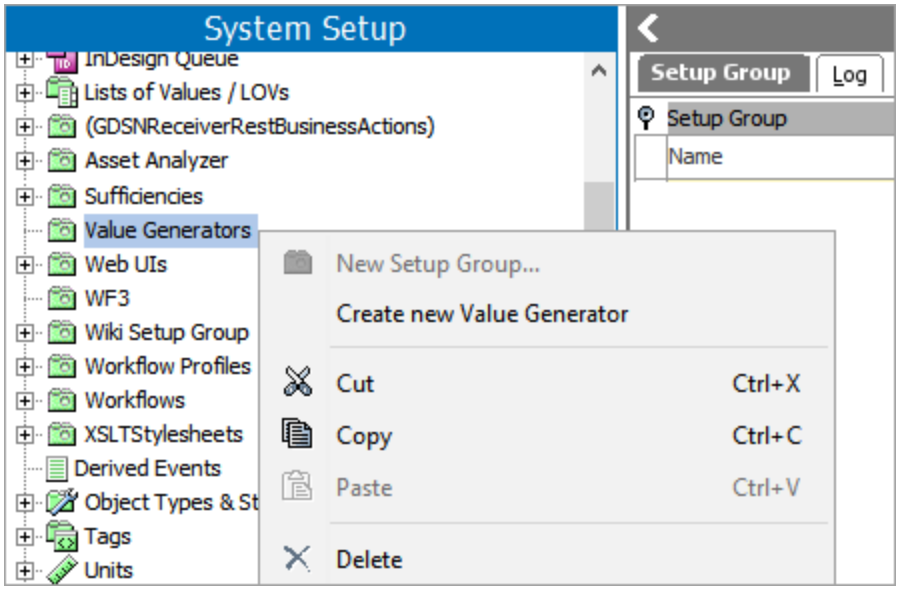
5. In the Create Setup Group Root dialog, select the object type created in the previous step (in this example, 'Value Generators') and enter a value for the ID and Name text field. In this example, the user created a setup group root with the ID 'ValueGenerators' and the Name 'Value Generators.'



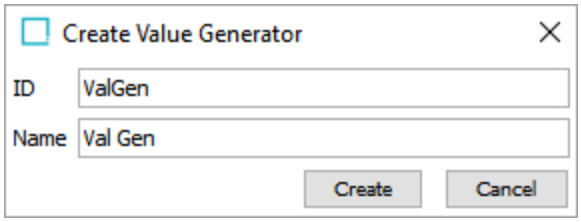
6. Click 'Create' to close the dialog.
7. Select 'Value Generator Configuration Type' from the Basic Objects menu in System Setup. In the References tab, select 'Add Parent,' and in the Select New Parent dialog, select the Setup Group (in this example, 'Value Generators') created in the previous step.



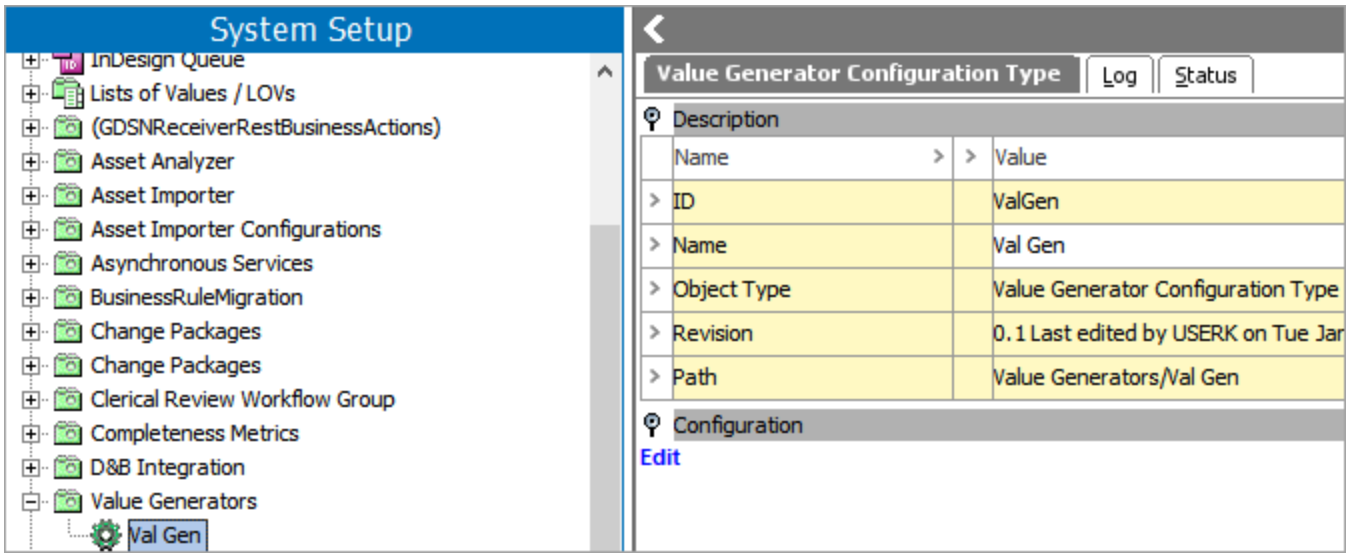
8. Click 'Select' to close the Select New Parent dialog.
9. Right - click the 'Value Generators' Setup Group and select 'Create new Value Generator.'



10. In the Create Value Generator dialog box, enter a value for the ID and Name text field. In this example, the user has created a value generator with the ID 'ValGen' and the Name 'Val Gen.'



11. Click 'Create.' The dialog closes and the Value Generator Configuration Type has been created.



Now that a Value Generator Configuration Type has been created, a specific value generator can be configured. For information regarding the configuration of Algorithm Value Generators, see the **Algorithm Value Generator** topic. For information regarding the configuration of List Value Generators, see the **List Value Generator** topic. For information regarding the configuration of Sequence Generators, see the **Sequence Generator** topic.

Algorithm Value Generator

The Algorithm Value Generator allows users to generate a value based on business logic (e.g., concatenating two or more attribute values to create a new value, adding a category name to a product line to create a new value, etc.). Additionally, the Algorithm Value Generator can generate values produced by List Value Generators and/or Sequence Value Generators.

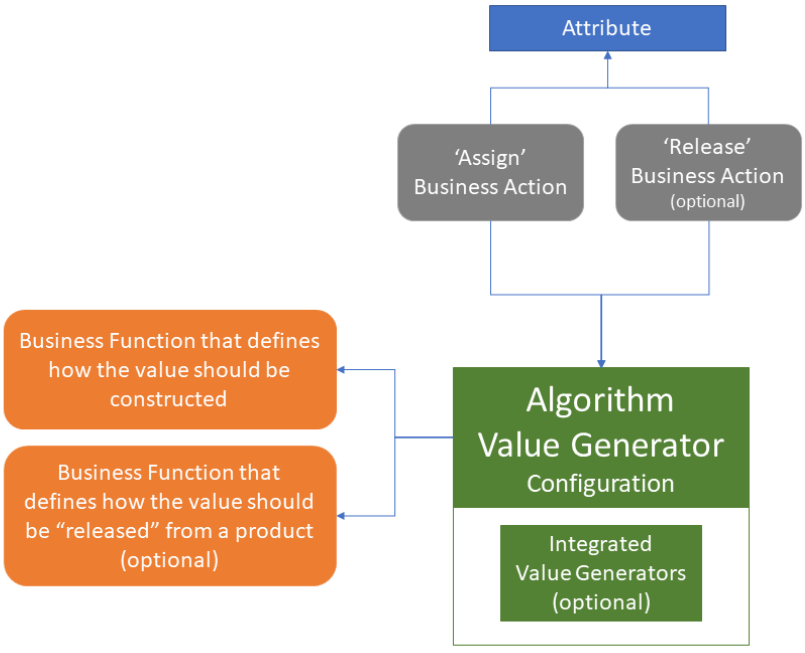
Prerequisites

Before an Algorithm Value Generator can be configured, a Value Generator Configuration Type must be created. For instructions on how to create a Value Generator Configuration Type, see the **Value Generators** topic.

Business functions and business actions must also be configured to generate and release values using the Algorithm Value Generator.

While this topic includes specific requirements necessary for business functions and business actions to be used in tandem with Algorithm Value Generators, users need a working knowledge of business functions (e.g., creating and inputting binds, return types, input parameters, etc.), and a working knowledge of configuring and implementing business actions. Additionally, users should have an understanding of JavaScript and how it works in conjunction with business functions. For more information, see the **Business Rules** documentation.

The diagram below visually displays how the Algorithm Value Generator works in relation with business actions and business functions to generate / release values.



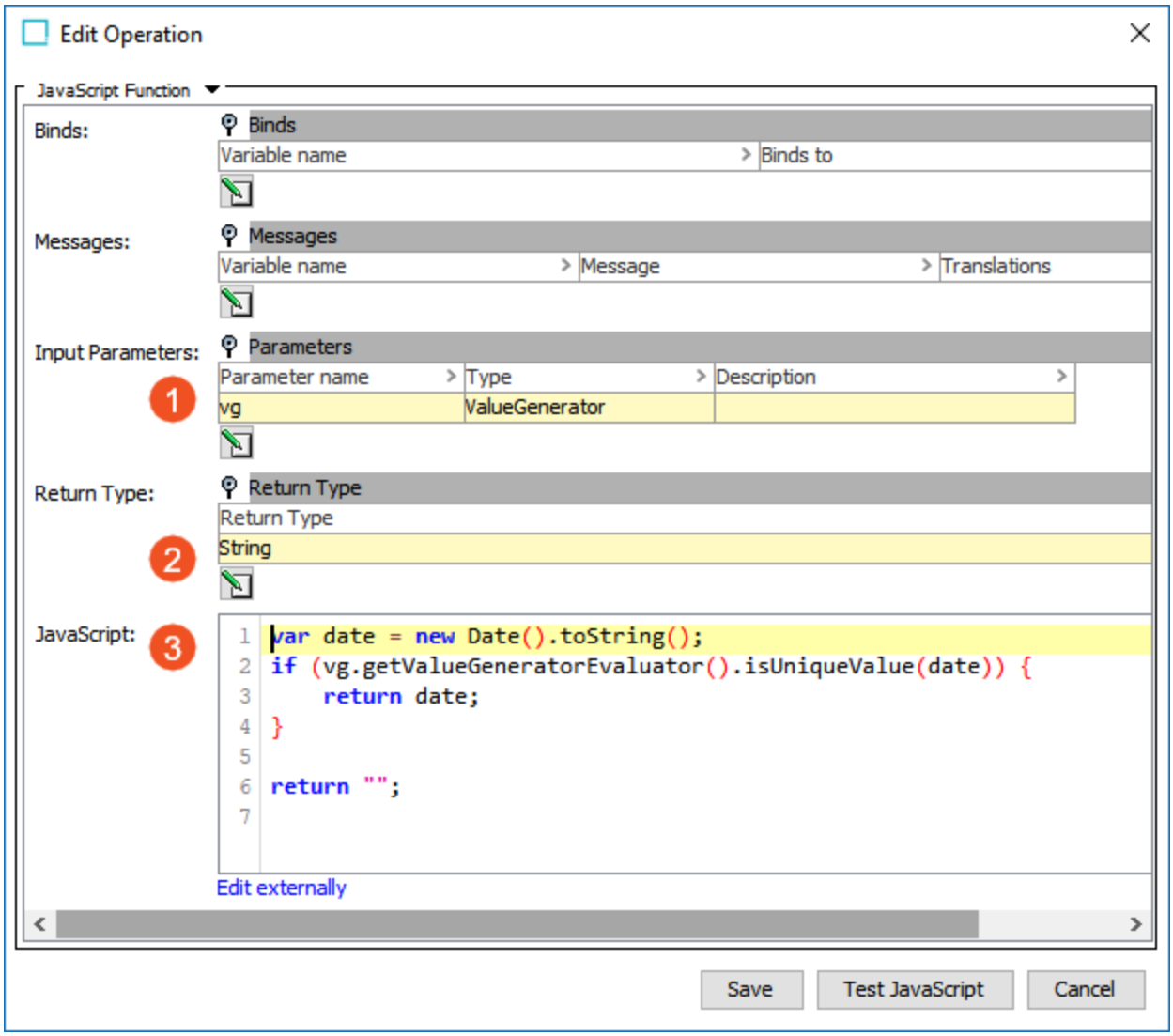
Configuring Business Functions to Generate a Value

There are several ways in which a business function can be configured to be used in tandem with an Algorithm Value Generator, depending on business needs. This topic includes examples of business functions that have been configured to generate values using the Algorithm Value Generator. One example shows a business function that does not use integrated value generators to generate a value. Another example show a business function that uses integrated value generators (specifically, a List Value Generator and a Sequence Value Generator).

Note: Although these examples demonstrate how business functions can be configured to work with an Algorithm Value Generator to generate a value, only the business function that uses integrated value generators is used in conjunction with the Algorithm Value Generator and business action examples addressed later in this topic.

Example business function for an Algorithm Value Generator that does not include integrated value generators

In this example, the business function has been configured in a way that does not use integrated value generators. This business function produces a value (in this case: a date) and checks for uniqueness of that value. If the date is not unique, it returns an empty string.

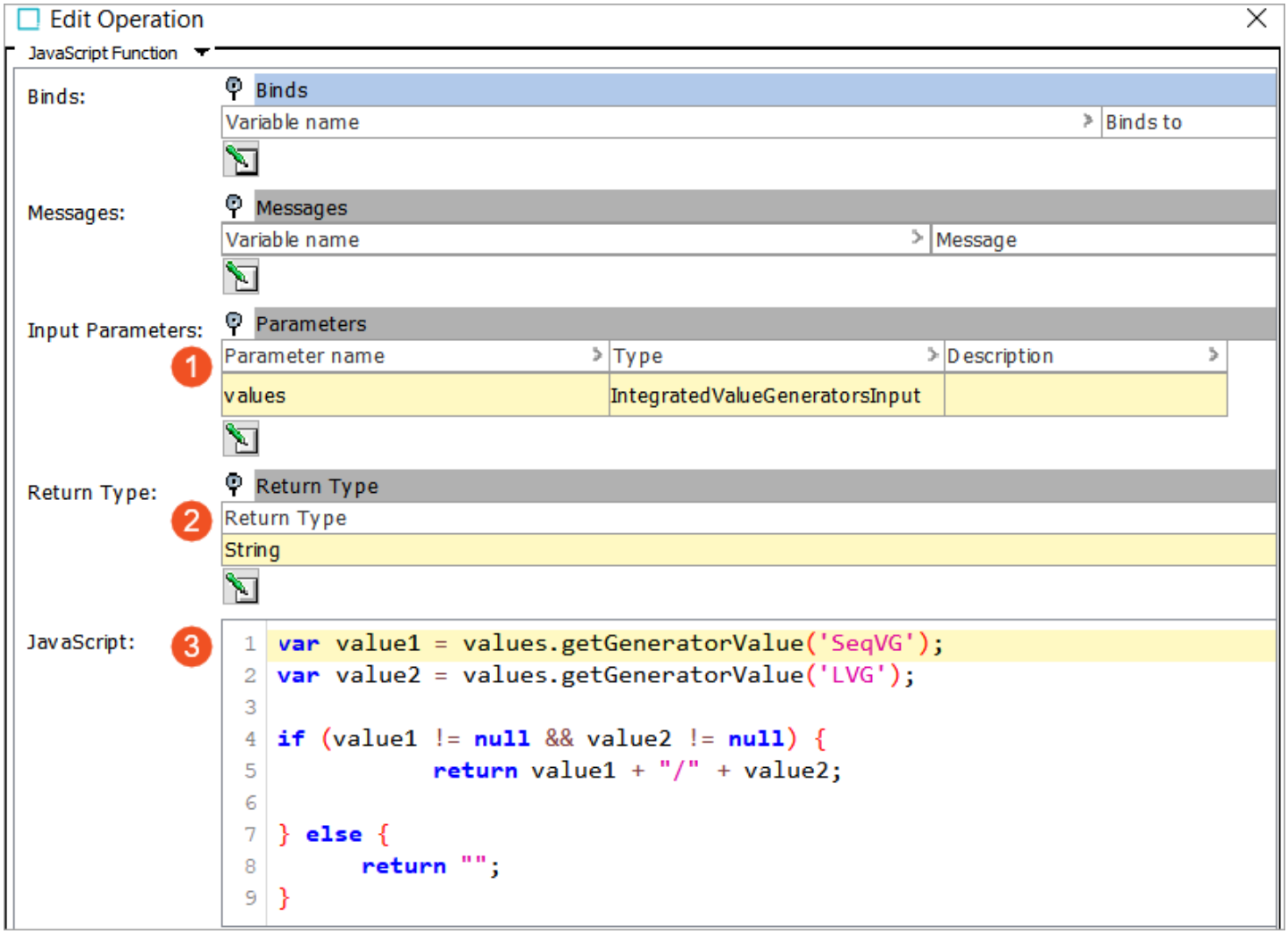


1. **Input Parameters**: Create a parameter name (in this example, 'vg'). Select Value Generator > 'ValueGenerator' as the type.
2. **Return Type**: Select 'String' as the return type.
3. **JavaScript**: A new date is returned as a value for the attribute if the value is unique. Otherwise, an empty string is returned as the value.

Note: Clicking the **Test JavaScript** button on the JavaScript Function for Value Generator can return a 'rollback performed' error. Instead, configure a business action (as defined in the sections below) to use the value generator and verify the result.

Example business function for an Algorithm Value Generator using integrated value generators

In this example, the business function has been configured to incorporate a List Value Generator and Sequence Value Generator to generate a value.



1. **Input Parameters**: Create a parameter name (in this example, 'values'). Select Value Generator > 'IntegratedValueGeneratorsInput' as the type.
2. **Return Type**: Select 'String' as the return type.
3. **JavaScript**: If the value of a Sequence Value Generator (in this example, 'SeqVG') and the value of a List Value Generator (in this example, 'LVG') are not equal to null, return a value that combines the values from 'SeqVG' and 'LVG.' These values are separated by '/.' If either of these value generators are equal to null (that is, there are no available values), then an empty string is returned as the value.

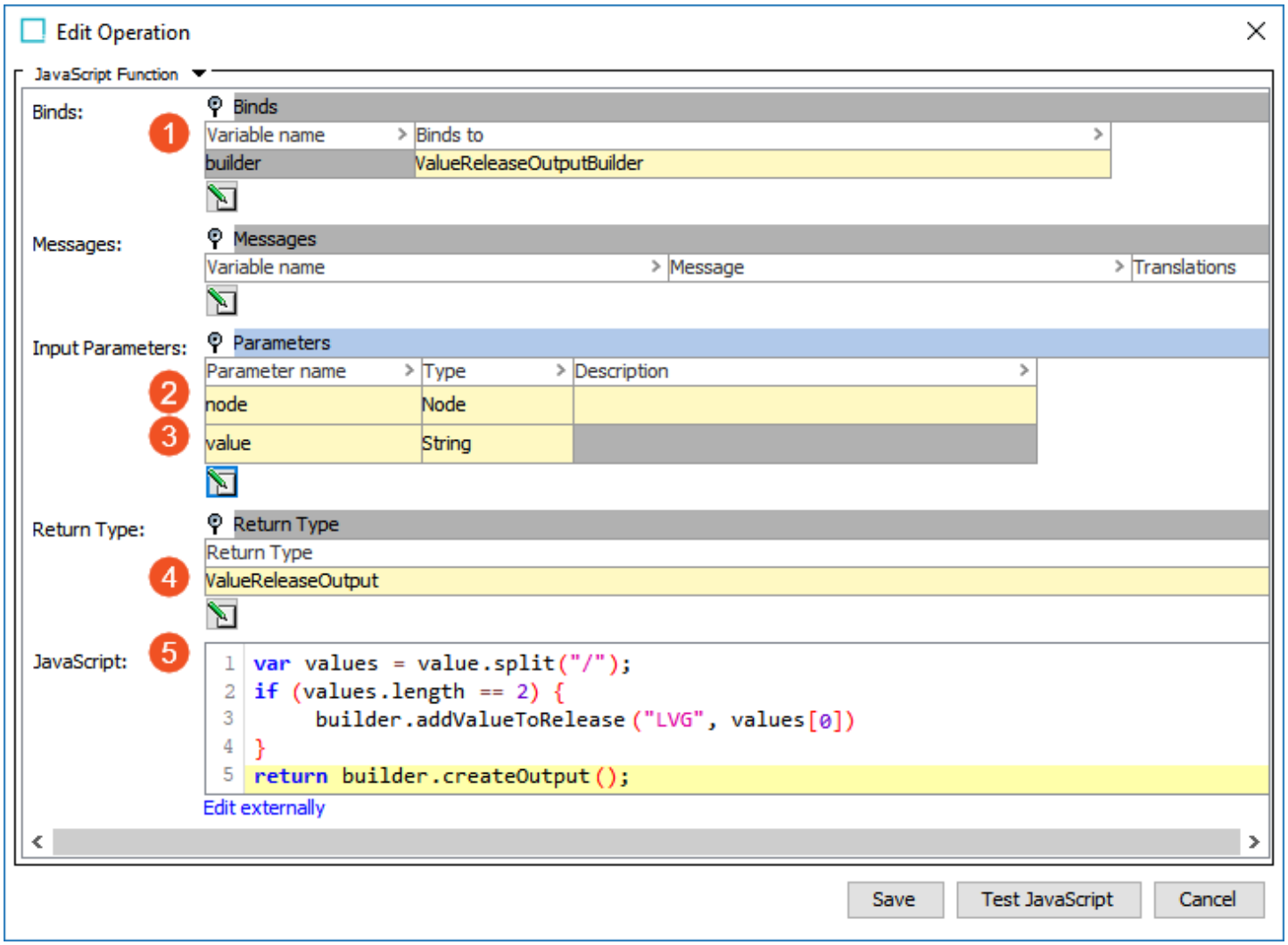
Note: Clicking the **Test JavaScript** button on the JavaScript Function for Value Generator can return a 'rollback performed' error. Instead, configure a business action (as defined in the sections below) to use the value generator and verify the result.

For information regarding List Value Generators, see the **List Value Generator** topic. For information regarding Sequence Value Generators, see the **Sequence Value Generator** topic.

Configuring a Business Function to Release a Value

Although similar to business functions that are configured to generate values, there are a number of specific differences when editing a business function whose purpose is to be used in tandem with an Algorithm Value Generator to release a value. These setting are described in the example below.

Important: Values that have been released can be used at a later time. However, released values for the Sequence Value Generator cannot be used at a later time as doing so would disrupt the sequence of numbers.

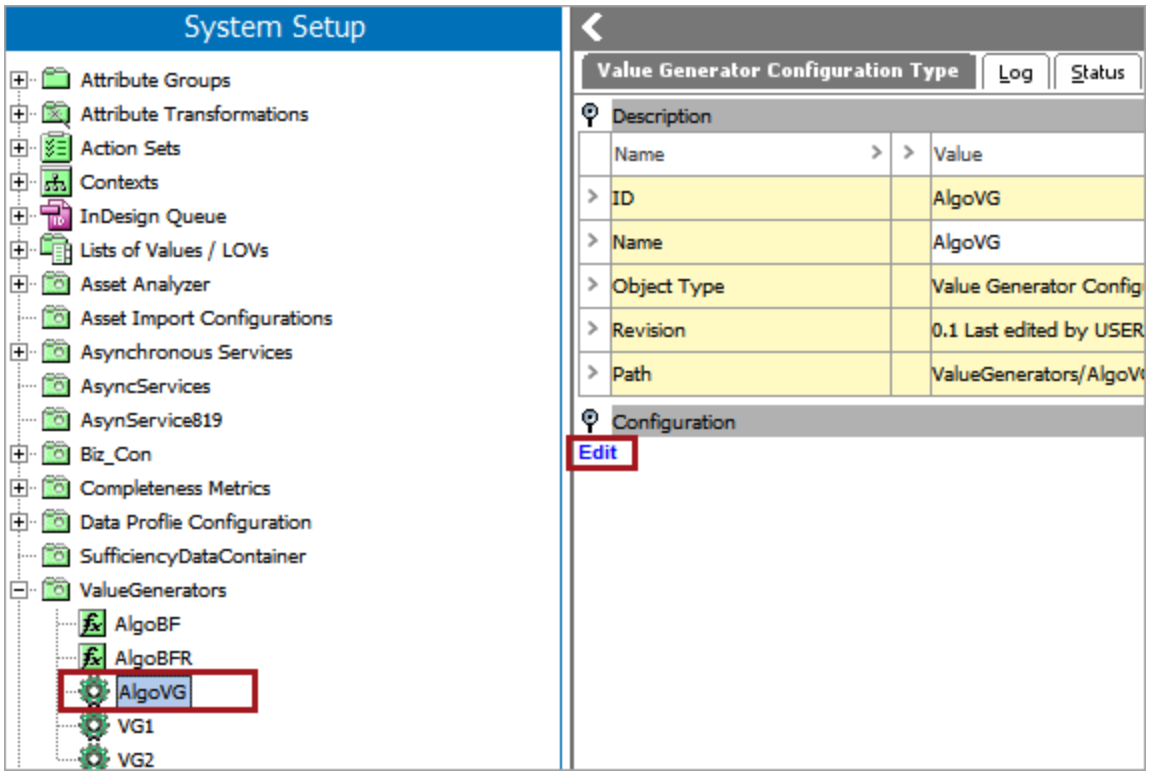


1. **Binds**: Create a variable name (in this example, 'builder'). Select Value Generator > 'ValueReleaseOutputBuilder' as the binds to.
2. **Input Parameters**: Create a parameter name (in this example, 'node'). Select 'Node' as the type.
3. **Input Parameters**: Create a parameter name (in this example, 'value'). Select 'String' as the type.
4. **Return Type**: Select Value Generator > 'ValueReleaseOutput' as the return type.
5. **JavaScript**: If the value is equal to a combination of two values and comprised of a value made up of two values separated by a forward slash, the value is released.

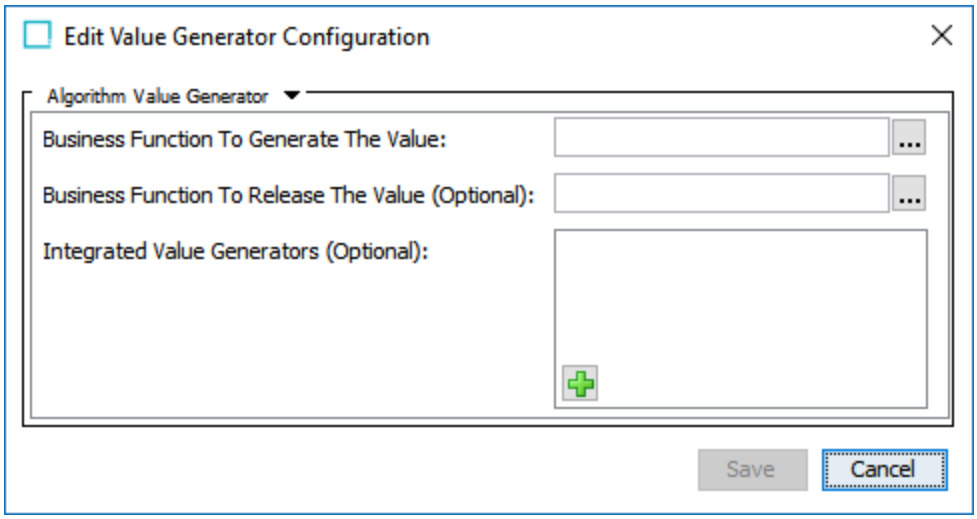
Configuring an Algorithm Generator

Once a business function is configured, an Algorithm Value Generator can be configured. Users should configure the business function prior to configuring the Algorithm Value Generator, as part of configuring the Algorithm Value Generator is selecting the business function to be used to generate / release a value.

1. With the desired Value Generator Configuration Type selected (in this case, AlgoVG), click the 'Edit' link, which is located under the Configuration flipper.



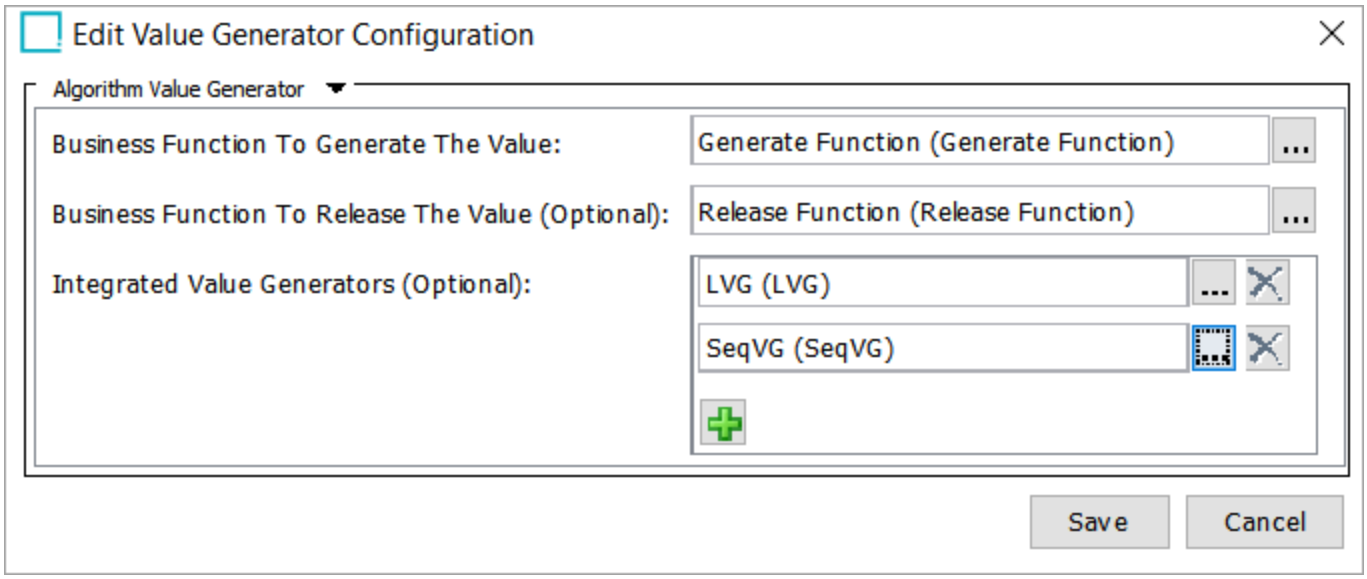
2. Select 'Algorithm Value Generator' from the dropdown menu. The parameters for configuring the Algorithm Value Generator are now visible and ready for editing.



3. Click the ellipsis button (...) next to the ' Business Function to Generate the Value' text field and select the business function to generate a value. In this example, the user has selected the second business function (the business function that includes two integrated value generators and is titled 'Generate Function') that was created earlier in this topic.

4. Click the ellipsis button (...) next to the Business Function to Release the Value (Optional) text field, and select the business function to release a value. In this example, the user has selected the business function (titled 'Release Function') that was created earlier in this topic.
5. In the 'Integrated Value Generators (Optional) text field, select the value generators you want to use within the Algorithm Value Generator by first clicking the green cross and then clicking the ellipsis button (...).

   In this example, the user has selected the List Value Generator ('LVG') and Sequence Value Generator ('SeqVG'), both of which are incorporated as part of the JavaScript within the first business function created earlier in this topic.



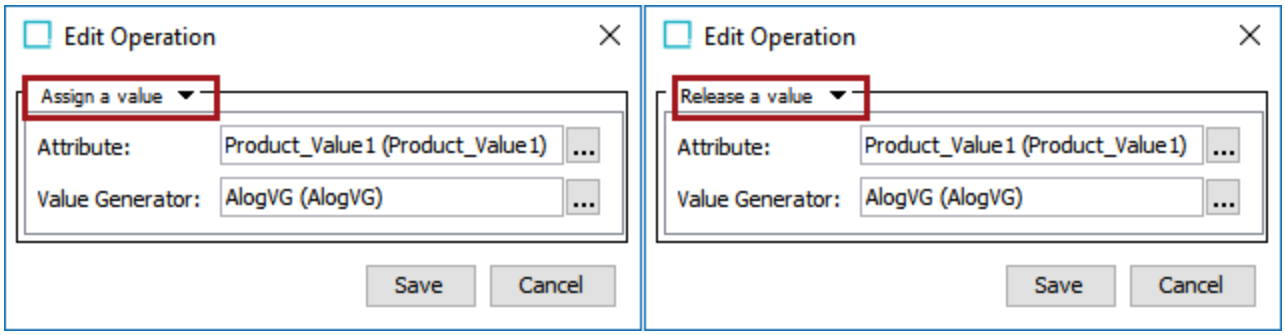
6. Click 'Save' to save your settings and close the dialog.

Configuring a Business Action to Generate / Release Values

Business actions must be configured to assign / release values using the Algorithm Value Generator.

While this section includes details to configure business actions used in conjunction with the Algorithm Value Generator, it does not include the steps necessary to create the business actions. For more information regarding the initial setup of business actions (which are considered to be part of business rules), see the **Business Rules** documentation.

1. Within the business action Edit Operation dialog, select 'Value Generator' > 'Assign a value' or 'Release a value' from the dropdown menu, depending on which action to perform.
2. Select an attribute and value generator. In this example the user has selected the attribute 'Product_Value1' and the value generator 'AlgoVG,' which was configured earlier in this topic.



'Assign a value' option

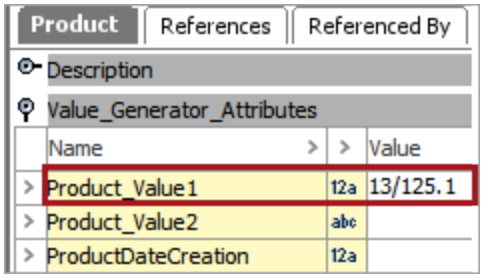
- The value is assigned to the selected attribute.
- The value is obtained from the selected value generator.

'Release a value' option

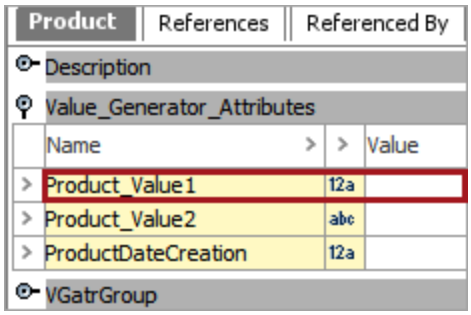
- The value id removed from the selected attribute.

Important: Values that have been released can be used at a later time. However, released values for the Sequence Value Generator cannot be used at a later time as doing so would disrupt the sequence of numbers.

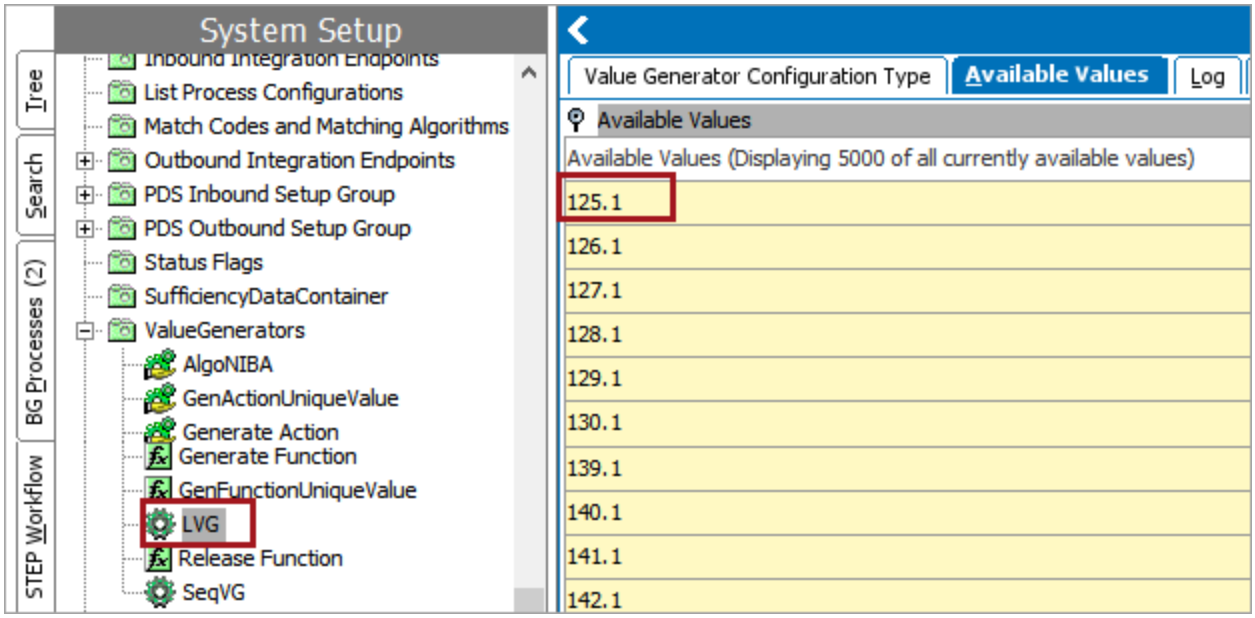
Once the business action configured to assign a value is run, the value for the attribute 'Product_Value1' is populated by a value as determined by the business function (in this case, a value taken from a Sequence Value Generator and List Value Generator, separated by a forward slash).



Once the business action configured to release a value is run, the value for the attribute 'Product_Value1' is removed from the attribute.



The value from the List Value Generator 'LVG' (125.1) has been placed back into the Available Values list and can be reused.



List Value Generator

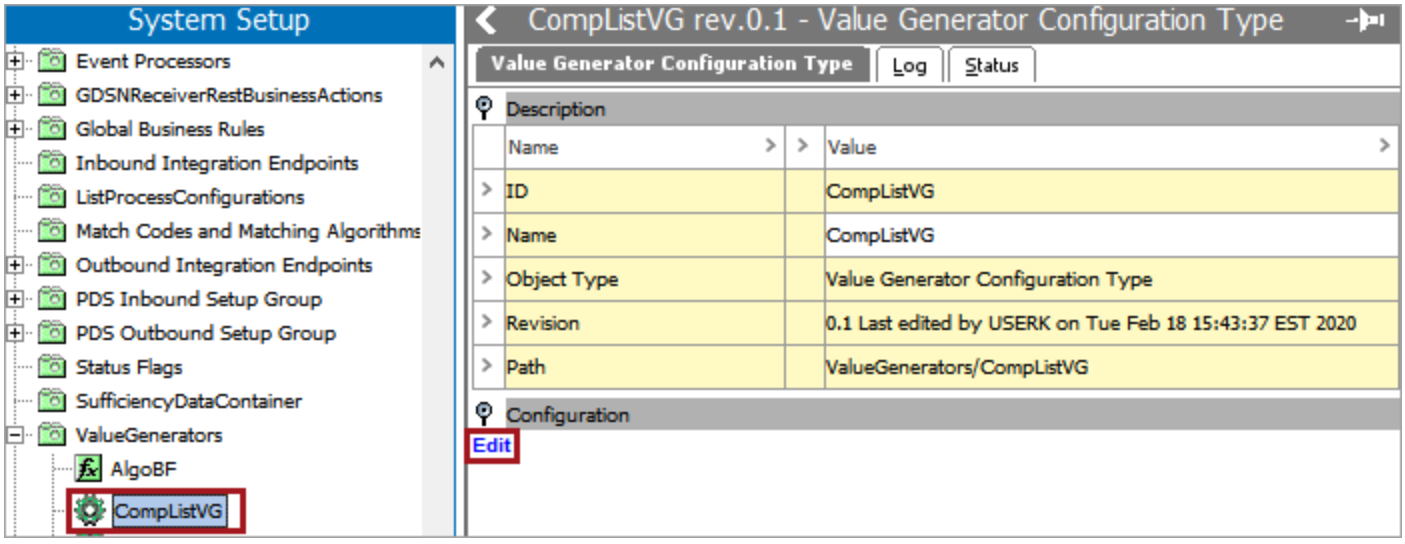
The List Value Generator allows users to import values from a specified list (e.g., a list of numbers) and assigns those values to an attribute of an object. Additionally, if a user decides that they do not want to retain a value that has been assigned to an attribute, the value can be returned to the list for later use.

Configuring a List Value Generator

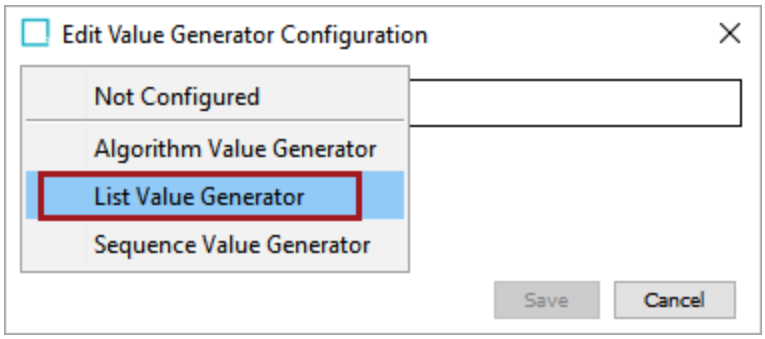
The following steps are required prior to using the List Value Generator to generate values for selected attributes.

> **Note:** Before a List Value Generator can be configured, a Value Generator object type must be created. For instructions on how to create a Value Generator object type, see the **Value Generators** topic.

1. With the desired Value Generator Configuration Type selected, click the 'Edit' link, which is located under the Configuration flipper in the Value Generator Configuration Type editor.



2. From the dropdown menu, select 'List Value Generator.'



Within the Edit Value Generator Configuration dialog, there are several editable parameters:

- **Prefix (Optional)**: Characters placed within this text field will precede values generated from a selected list.
- **Suffix (Optional)**: Characters placed within this text field will proceed values generated from a selected list.
- **Notification (Optional)**: As the descriptive text indicates, once the Warning Threshold number is met (i.e., the number of values left within the list selected by the value generator), an email will be sent to the email recipients included in the Email text field. A semicolon (;) is used to separate email addresses.

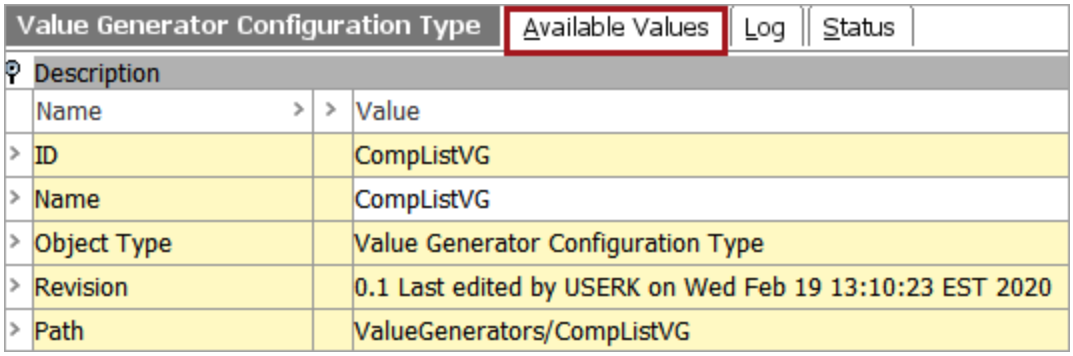
Important: The Warning Threshold and Email parameters are dependent on each other, e.g., if one is populated, the other must be populated as well.

3. Once the desired parameters are populated, Click 'Save' and close the Edit Value Generator Configuration dialog.

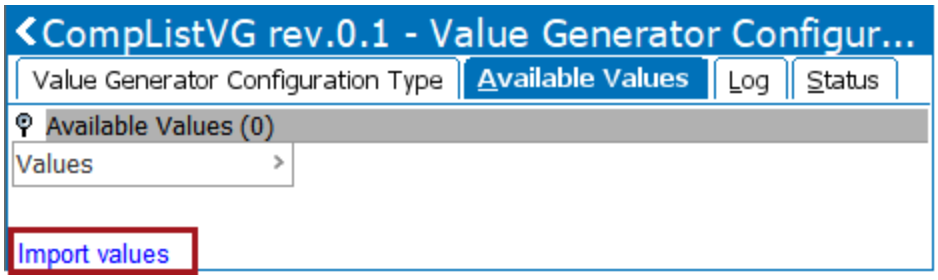
Importing Values into the List Value Generator

Once the List Value Generator is configured, values must be imported for use.

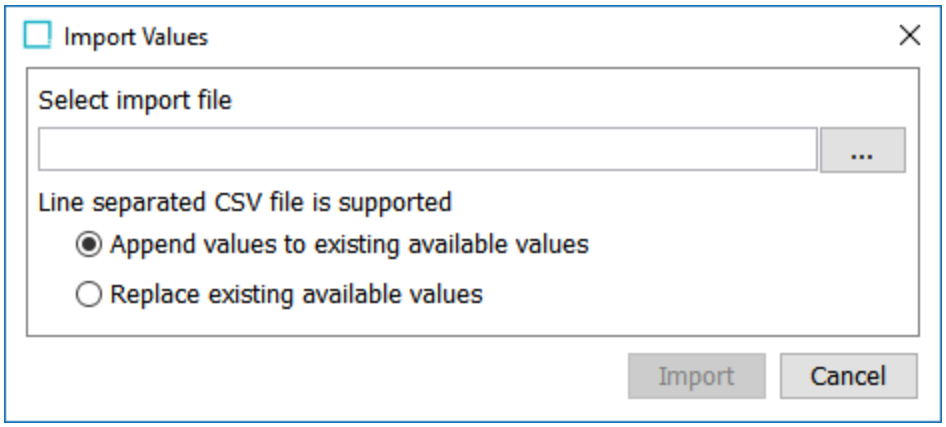
1. With the List Value Generator configured, click on the 'Available Values' tab.



2. With the Available Values tab open, select the 'Import values' link.

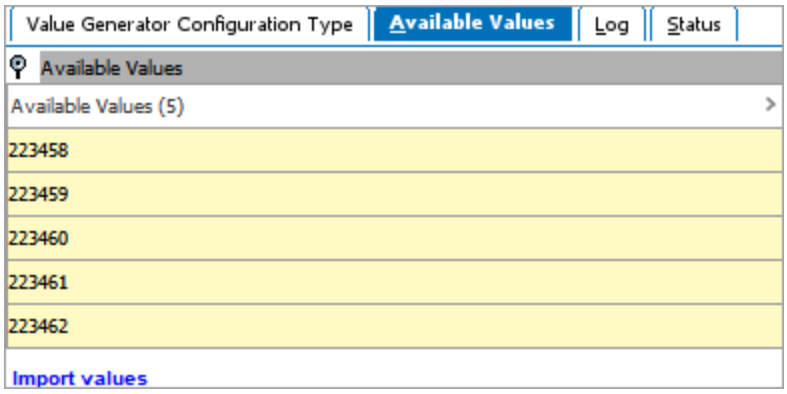


3. In the Import Values dialog, click the ellipsis button (...) located next to the 'Select import file' text field. Once the file is selected, click the 'Import' button to import the values and close the dialog.



Important: Only line-separated CSV files are supported, and values that are imported can be added to ('Append values...') or replace already-existing values (if applicable). Importing duplicate values is not permitted. If a user tries to import duplicate values, a message stating which values were omitted during import will appear. A duplicate value is considered any value that already exists in the Available Values list or has already been assigned to an attribute from the List Value Generator.

The available values are now ready to be used with the List Value Generator. By default, the available values are sorted by when they were imported and the order of values within the CSV file itself (e.g., the most recent imported values will be at the bottom of the list, and a value within a higher row will be above a value in a lower row). In this example, the first available value is '223458.' This value will be seen later in this topic when the business actions used in conjunction with the List Value Generator are run.



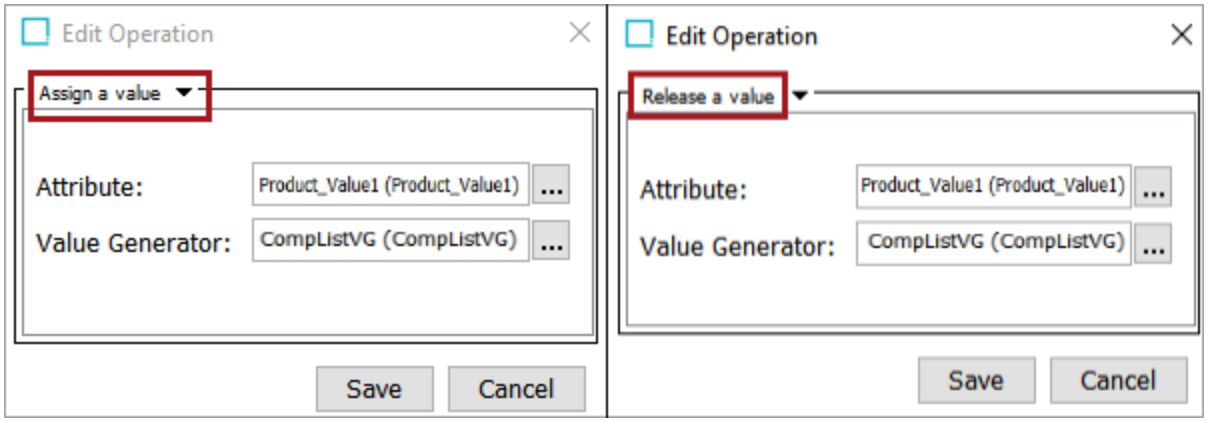
Note: The 'Available Values' table can display up to 5,000 values at one time. If there are more than 5,000 values in a list, the message 'Displaying 5000 of a currently available values' will be displayed next to 'Available Values.' Otherwise, the specific number of available values (in this example, '5') will be within the parenthesis next to 'Available Values.' Additionally, filtering will only apply to the values that are visible within the table. For more information regarding filtering options, see the **Sorting and Filtering Views** topic in the **Views** section of the **Publisher (Adobe InDesign Integration)** documentation.

Configuring Business Actions for a List Value Generator

Once a list of available values have been added to a List Value Generator, business actions must be configured to assign / release values.

While this section will cover the specifics regarding the configuration of business actions used in conjunction with the List Value Generator, it will not cover the steps necessary to create business actions themselves. For more information regarding the initial setup of business actions (which are considered to be part of business rules), see the **Business Rules** topic in the **Using Business Rules in STEP** documentation.

1. Within the business action Edit Operation dialog, select 'Value Generator' > 'Assign a value' or 'Release a value,' depending on which action to perform.
2. Select an attribute and value generator. In this example the user has selected the attribute Product_Value1 and the value generator CompListVG, which was configured earlier in this topic.



With the 'Assign a value' option selected:

- The value will be assigned to the selected attribute.
- The value will be obtained from the top of the selected value generator's value list.

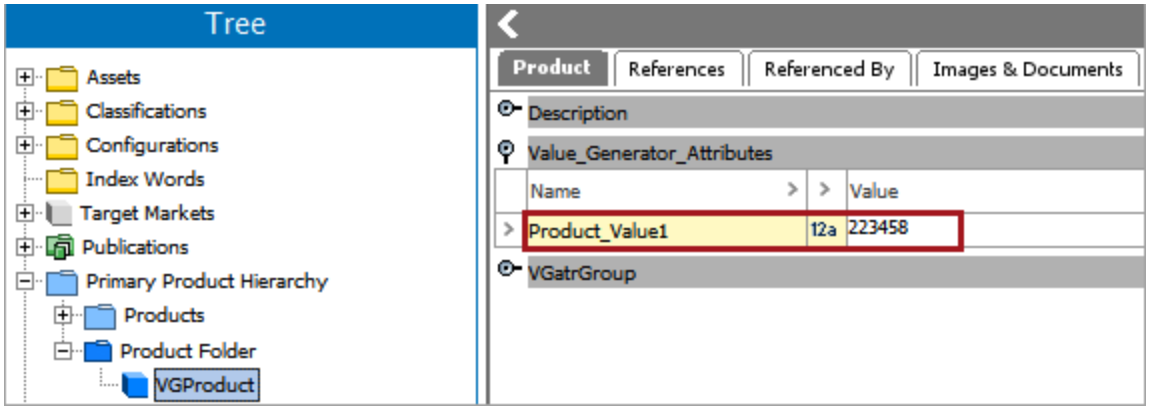
With the 'Release a value' option selected:

- The value will be removed from the selected attribute.
- The value will be put back at the top of the selected value generator's value list.

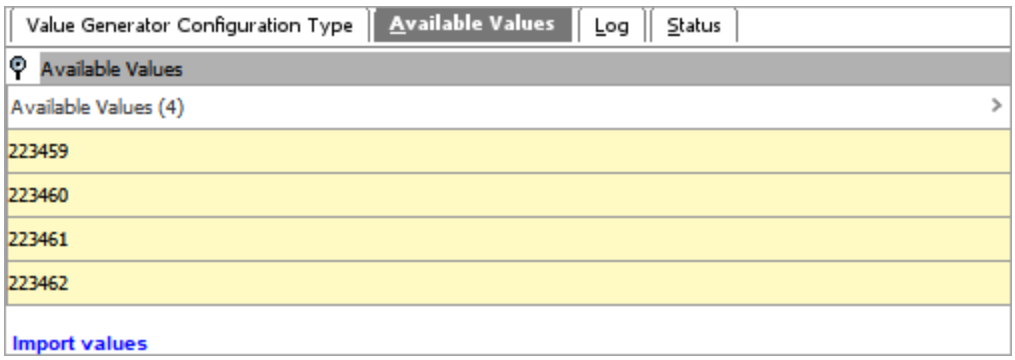
Important: When selecting an attribute that the value will be applied to, be aware of the attribute's validation base type, especially if the prefix or suffix text field was populated when the List Value Generator was configured. An error may occur if, for example, a prefix of 'GTIN' (or some other combination of letters) is included as a prefix or suffix for the list value generator, and then an attribute is selected, which does not allow for anything besides numbers. For more information on validation base types, see the **Validation Base Type** topic in the **Validation Rules** documentation.

3. Click 'Save' to save your settings and close the dialog.

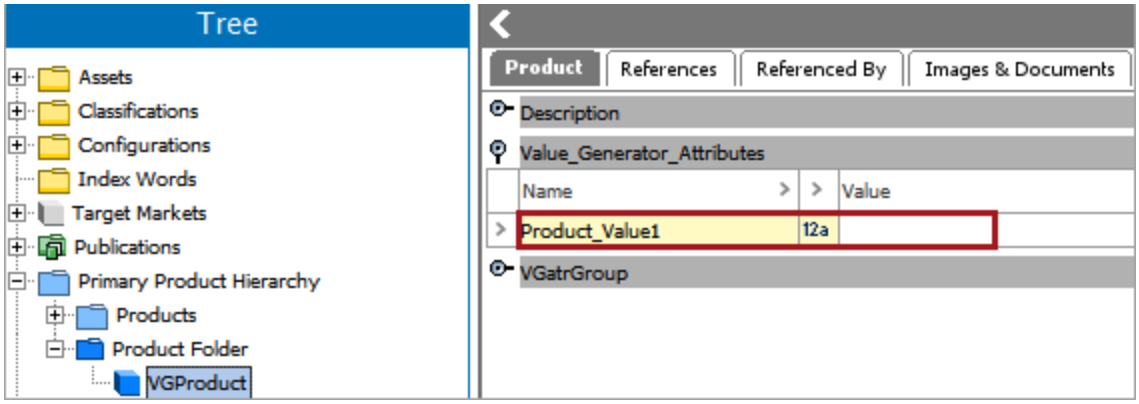
   When the business action with the 'Assign a value' setting selected is run, it will obtain the next value from the list in the List Value Generator and assign it to the attribute 'Product_Value1.'



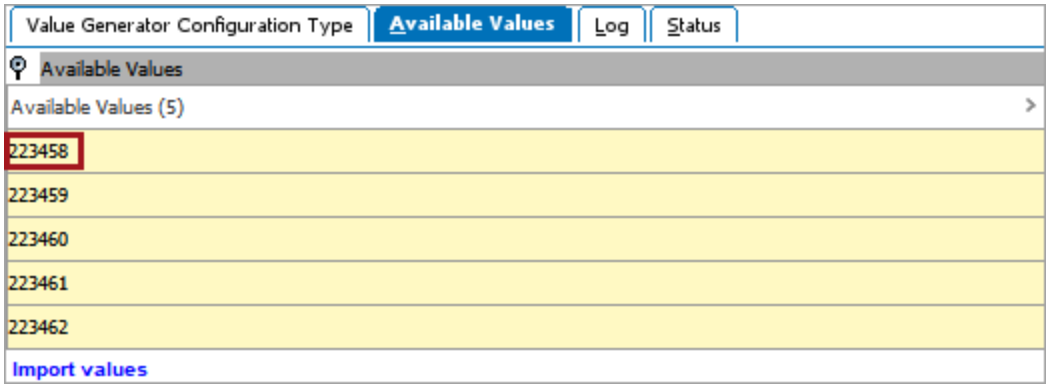
   The Available Values list now only contains four available values.



When the business action with the 'Release a value' setting selected is run, it will remove the value from the attribute.



The value is placed back into the Available Values list in the List Value Generator and is ready to be assigned to an attribute when the 'Assign a value' business action is run.



Sequence Value Generator

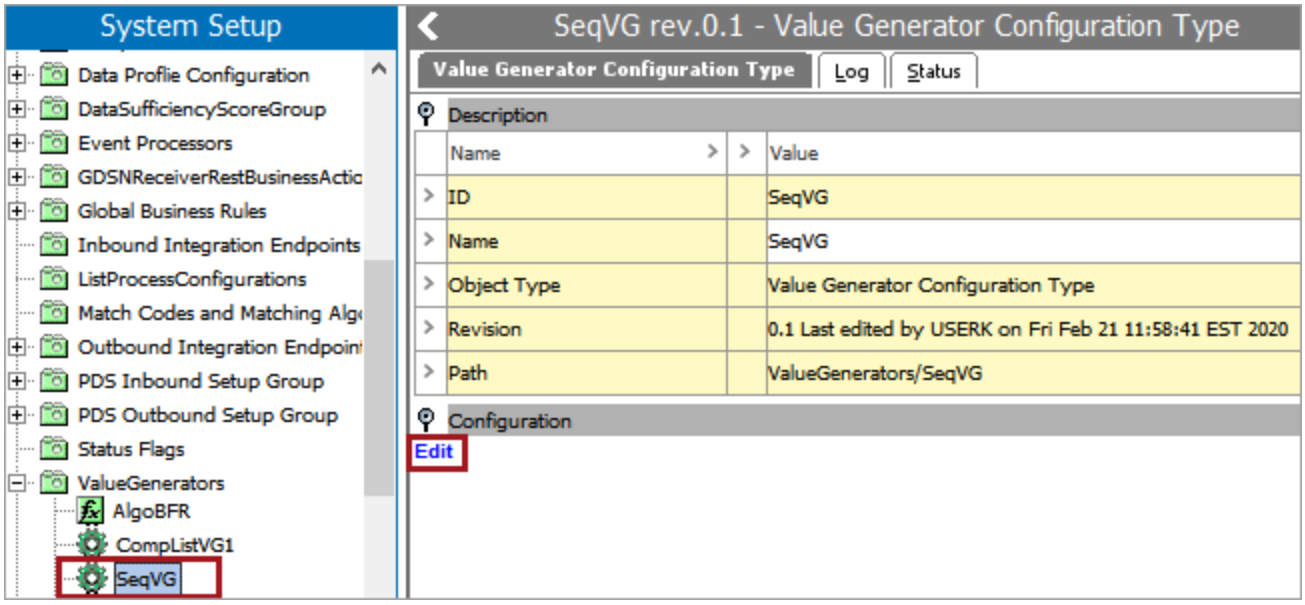
The Sequence Value Generator allows users to assign unique values to attributes, based on a user-determined sequence of numbers.

Configuring a Sequence Value Generator

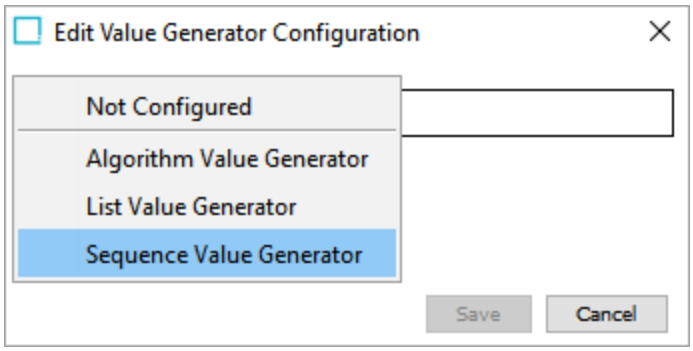
The following steps are required prior to using the Sequence Value Generator to generate values for selected attributes.

Note: Before a Sequence Generator can be configured, a Value Generator object type must be created. For instructions on how to create a Value Generator object type, see the **Value Generators** topic.

1. With the desired Value Generator Configuration Type selected, click the 'Edit' link, which is located under the Configuration flipper of the Value Generator Configuration Type editor.



2. From the dropdown menu, select 'Sequence Value Generator.'

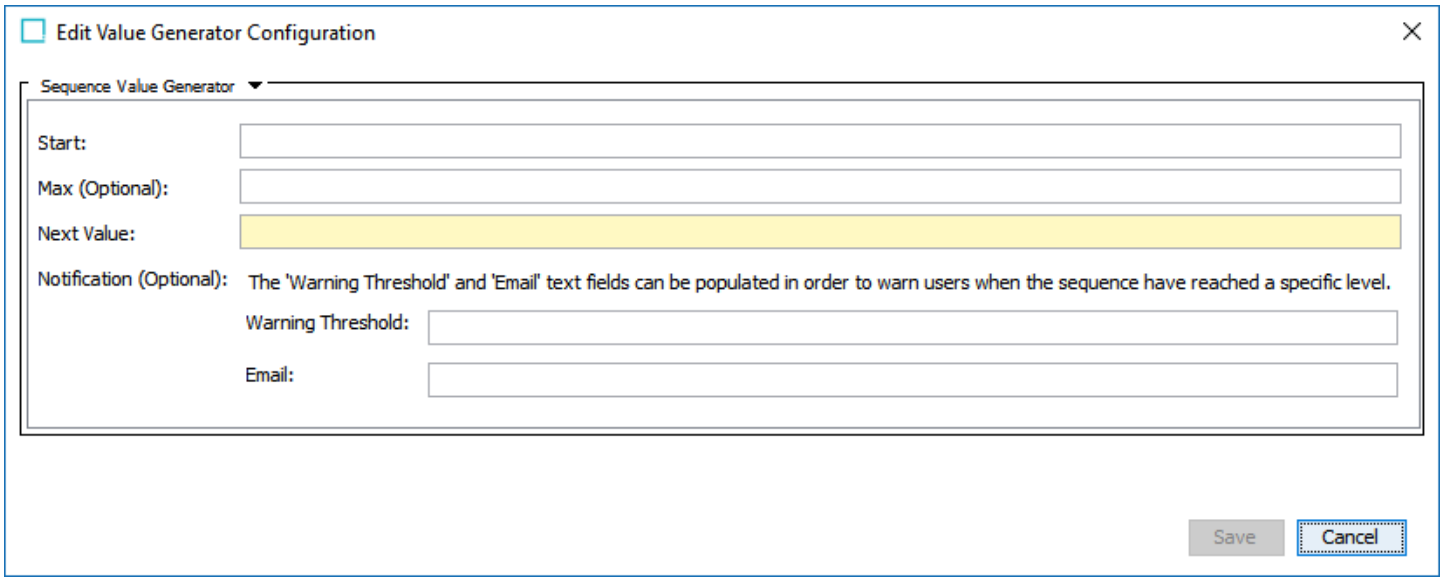


Within the Edit Value Generator Configuration dialog, there are several editable parameters:

- **Start**: The starting number for the number within the range.
- **Max (Optional)**: The maximum number within the range.
- **Notification (Optional)**: As the descriptive text indicates, once the Warning Threshold number is met (i.e., the sequence has reached a specific level), an email will be sent to the email recipients included in the Email text field. A semicolon (;) is used to separate email addresses.

Important: The Warning Threshold and Email parameters are dependent on each other, e.g., if one is populated, the other must be populated as well.

The parameter Next Value is a read-only text field. It displays the next available number in the sequence.



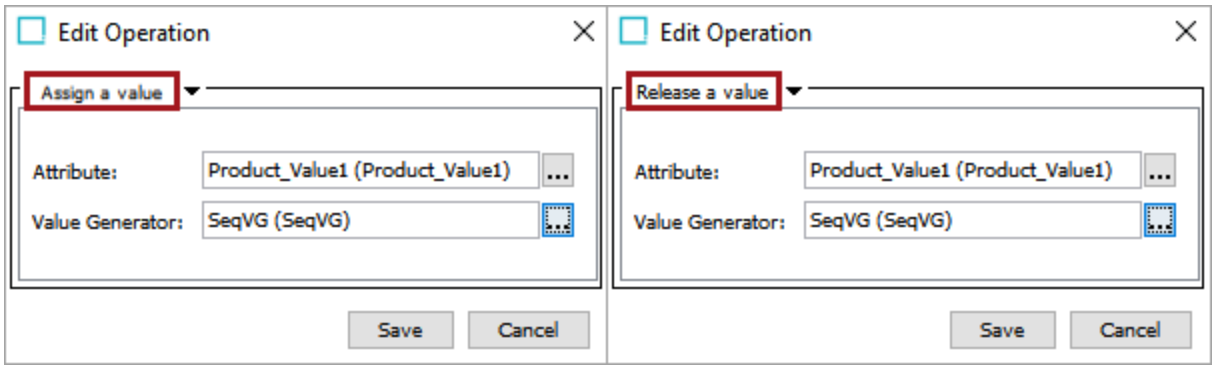
3. Once the desired parameters are populated, click 'Save' and close the Edit Value Generator Configuration dialog.

Configuring Business Actions for a Sequence Value Generator

Now that a Sequence Value Generator has been configured, business actions must be configured to assign / release values.

While this section will cover the specifics regarding the configuration of business actions used in conjunction with the Sequence Value Generator, it will not cover the steps necessary to create business actions themselves. For more information regarding the initial setup of business actions (which are considered to be part of business rules), see the **Business Rules** topic in the **Using Business Rules in STEP** documentation.

1. Within the business action Edit Operation dialog, select 'Value Generator' > 'Assign a value' or 'Release a value,' depending on the action to perform.
2. Select an attribute and value generator. In this example the user has selected the attribute Product_Value1 and the value generator SeqVG, which was configured earlier in this topic.



With the 'Assign a value' option selected:

- The value will be assigned to the selected attribute.
- The value will be obtained from the selected value generator.

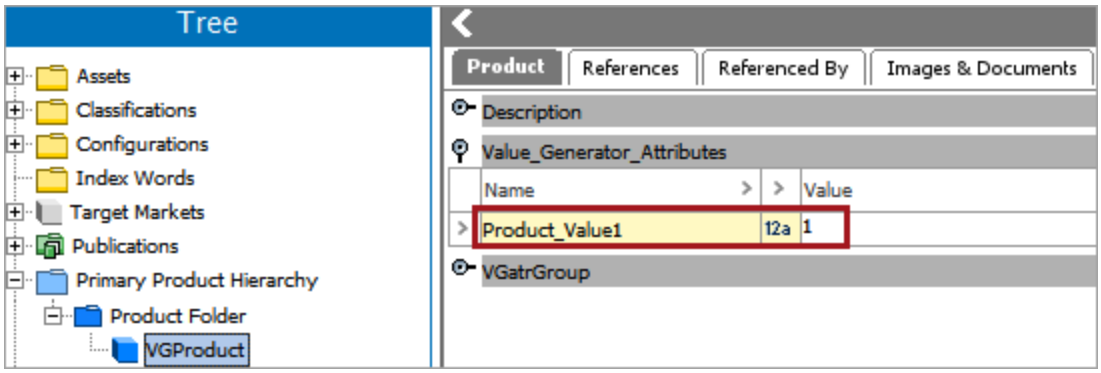
With the 'Release a value' option selected:

- The value will be removed from the selected attribute.

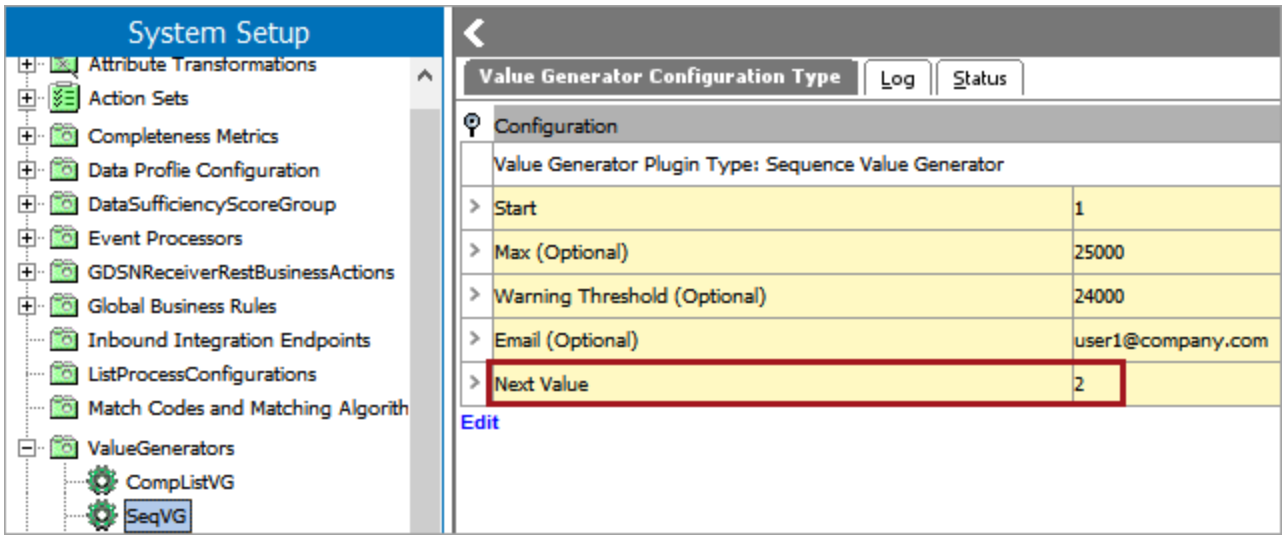
Note: Unlike the Algorithm Generator and List Value Generator, the value will not be returned to the list of available values when the 'Release a value' option is selected, as doing so would disrupt the number sequence.

3. Click 'Save' and close the dialog.

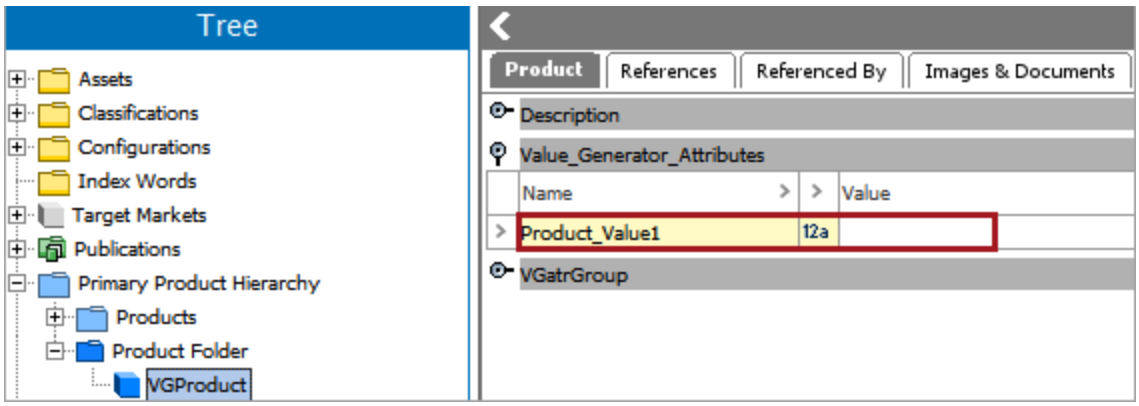
   When the business action with the 'Assign a value' setting selected is run, it will obtain the next available value from the Sequence Value Generator (in this case, '1') and assign it to the attribute 'Product_Value1.'



With the number obtained from the Sequence Value Generator, you can see the next available value (2), as seen in the image below.



When the business action with the 'Release a value' setting selected is run, it will remove the value from the attribute.



Wiki Metadata

Configuration Governance involves capturing data that explains how your system works and how to use it. This could include metadata that documents, for example, the purpose of an attribute or integration endpoint, the owner of an LOV, or the process to request a change to a workflow. For STEP, the wiki provides a way to capture configuration governance metadata on many system setup objects, and displays it via a 'Go to Wiki' link.

The Wiki Metadata functionality requires installation of the XWiki advanced open source enterprise wiki platform and database, which reside outside of STEP. However, the XWiki content pages can be launched from both workbench and Web UI, and the content can also be searched. For information about XWiki, review the content found on the web at: https://www.xwiki.org/xwiki/bin/view/Documentation/

Executing the Wiki Metadata STEP installation recipe includes creation of a pre-configured, event-based OIEP to handle creation of and updates to wiki pages. After a successful installation of both STEP and XWiki, links to the wiki are included in the workbench and in Web UI.

Setup Requirements

Setting up and using the wiki involves the following steps:

1. Install, configure, and test the Wiki Metadata as defined in **Initial Setup for Wiki Metadata**.
2. Generate new wiki pages as defined in **Creating Wiki Metadata Pages**.

Additional Information

The following information is useful once the wiki is set up:

1. Display a wiki page for a selected system setup object as defined in:
   - **Viewing Wiki Metadata Pages from Workbench**
   - **Viewing Wiki Metadata Pages from Web UI**
2. Modify the data on the wiki page as defined in **Editing Wiki Metadata Pages**.

Initial Setup for Wiki Metadata

Complete the following one-time steps to set up and use Wiki Metadata.

1. Install the XWiki platform and database on a system, noting the hardware and software requirements and following the instructions on the web at:
   http://www.xwiki.org/xwiki/bin/view/Documentation/AdminGuide/Installation/

   Note: Although HSQL database is allowed for XWiki installation, it is not recommended for a STEP production environment.

2. Go to the directory where you installed XWiki and execute the start file appropriate for your system (Windows or Unix).
3. Contact your Stibo Systems representative to activate the 'wikimetadata' component.
4. On the STEP application server, edit the sharedconfig.properties file, and add the following case-sensitive properties:
   - **Wiki.OIEPID=WikiMetadataEndpoint** - This property identifies the ID (WikiMetadataEndpoint) of the OIEP created upon restarting the server after applying the STEP Wiki Metadata installation license.
   - **WikiEndpointImportStartPlugin.ContextID=[context ID]** - Replace [context ID] with the ID of the context that should be set on the OIEP. If this property is not added, the default context is global with an ID of 'GL'. The OIEP creation requires that a context exists with either an ID of 'GL' or the context ID defined by this property. Multiple contexts can be added if needed, using a comma-delimited list of IDs.
   - **WikiEndpointImportStartPlugin.User=[user ID]** - Replace [user ID] with the ID of the user that should be set on the OIEP. If this property is not added, the default user is database administrator with an ID of 'DBA'. The OIEP creation requires that a user exists with either an ID of 'DBA' or the user ID defined by this property.
5. Stop and restart the application server to create a new setup group named 'Wiki Setup Group' in System Setup, as well as a new event-based OIEP named 'WikiMetadataEndpoint'. If either the setup group or OIEP already exists, even in the recycle bin, they cannot be created.

   Important: An OIEP with an ID that matches the ID given by the Wiki.OIEPID property is required to display the wiki links in Web UI.

6. In System Setup, manually configure the **OIEP Delivery Method** as required to work with your system. For details on configuration, see the **Wiki Delivery Method** topic in the **Data Exchange** documentation.
7. In System Setup, manually configure the WikiMetadataEndpoint event-based OIEP parameters:
   - On the Configuration tab > Configuration flipper, update the **Schedule** to run as required. By default, the OIEP is created to be invoked every minute. For details, see the **Schedule** section of the **OIEP - Configuration Flipper** topic in the **Data Exchange** documentation.

- On the Configuration tab > Configuration flipper, update the **Workspace** parameter if needed. The default workspace is Approved.
- On the Configuration tab > Event Queue Configuration flipper, update the **Queue Status** parameter to **Read Events**. By default, the OIEP is created to discard events, and no wiki pages are created with this setting. For details, see the **Event-Based OIEP Status and Queue Status** topic in the **Data Exchange** documentation.
- On the OIEP's Event Triggering Definitions tab, verify that the triggers are adequate to generate the required wiki pages. For more information, see the **Creating Wiki Metadata Pages** topic.

8. In System Setup, right-click the OIEP and click the **Enable Integration Endpoint** option to run the OIEP based on the schedule. For details, see the **Running an Outbound Integration Endpoint** topic in the **Data Exchange** documentation.

Setup is now complete. Continue by verifying that setup was successful by following the **Testing Wiki Metadata** steps below.

Testing Wiki Metadata

1. In the System Setup attribute node, create a new attribute. This action should trigger the creation of a new wiki page.
2. Right-click the WikiMetadataEndpoint OIEP and click the **View First Event Batch** option. The Current Event Batch dialog displays the attribute you created and a number of events triggered by the creation process.
   - If the attribute is not displayed with events, revisit the OIEP manual configuration steps.
3. On the Current Event Batch dialog click the **Close** button.
4. Right-click the WikiMetadataEndpoint OIEP and click the **Invoke Endpoint** option. A background process is started.
5. On the WikiMetadataEndpoint OIEP Background Processes tab, verify the process succeeds.
   - If the background process fails, revisit the configuration of the Delivery Method on the OIEP.
6. Right-click the new attribute and click the **Go to Wiki** option.
   - If the wiki page displays with the expected data, continue to the **Creating Wiki Metadata Pages** topic.
   - If the wiki page does not display as expected, revisit the setup steps and update as required.

Creating Wiki Metadata Pages

After the OIEP setup is complete, events are generated based on the settings of the OIEP's Event Triggering Definitions tab. For more information on how events are handled by an OIEP, see the **OIEP - Event-Based - Event Triggering Definitions Tab** topic in the **Data Exchange** documentation.

Events queued by the OIEP are processed by the queue to create and update wiki pages.

Events can be triggered for:

- Attribute groups
- Attributes
- Business rules (action, condition, function, and library)
- Integration endpoints (inbound, outbound, and gateway)
- LOVs
- Reference types
- User Groups
- Workflows

Some system setup object types include a 'Go to Wiki' link, but do not trigger an event. For details on objects or changes that do not trigger events, see the **Events Not Triggered** topic in the **System Setup / Super User Guide** documentation.

In the wiki, two pages are created for each system setup object triggered: [object ID]-STEP and [object ID]-CUSTOM. The STEP page includes metadata sent from the STEP system. The CUSTOM page includes a template for user defined content.

When the OIEP includes multiple contexts, multiple versions of the data will be included in the wiki STEP page. Data from the available contexts are displayed with the qualifier ID from the context.

Bulk Wiki Page Creation

The Republish command allows you to populate the wiki as a whole. For more details, see the **Event-Based OIEP Forward, Rewind, Purge, and Republish** topic in the **Data Exchange** documentation.

> **Note:** The time required to complete the republishing process is based on the number of objects being republished, the environment, and the server performance.

1. Select the Workspace that should be used for republishing events.
2. On System Setup, open the WikiMetadataEndpoint OIEP Configuration tab.
3. On the Event Queue Configuration flipper, click the **Republish** button.
4. Use the **Select Setup Nodes to Republish** checkboxes to select the objects that should be republished. For example, check the 'Republish all Attributes' checkbox and the 'Republish all setup nodes' checkbox.

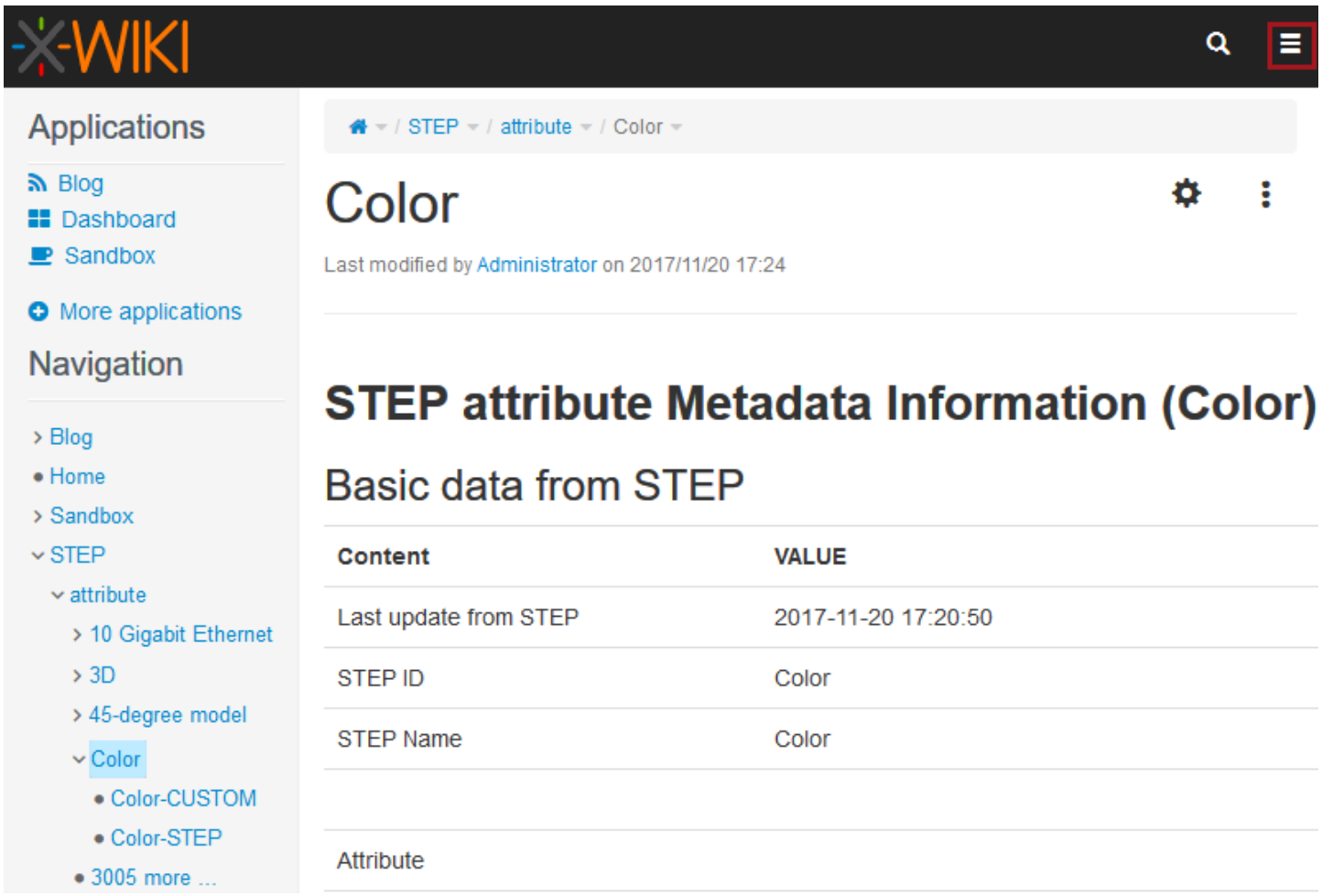
5. Add text to identify the **Process Description**. This is a required parameter.
6. Click the **Start Republish** button to start the republishing background process.

Editing Wiki Metadata Pages

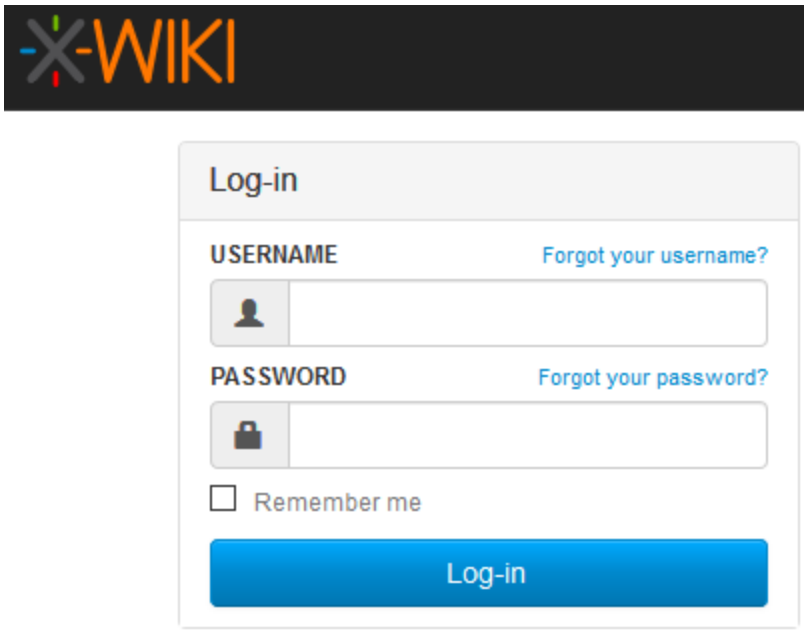
Once wiki pages exist, users can update them as new information becomes available.

1. For the desired object, display the wiki by clicking the **Go to Wiki** link. For details, see the **Viewing Wiki Metadata Pages from Web UI** topic or the **Viewing Wiki Metadata Pages from Workbench** topic.

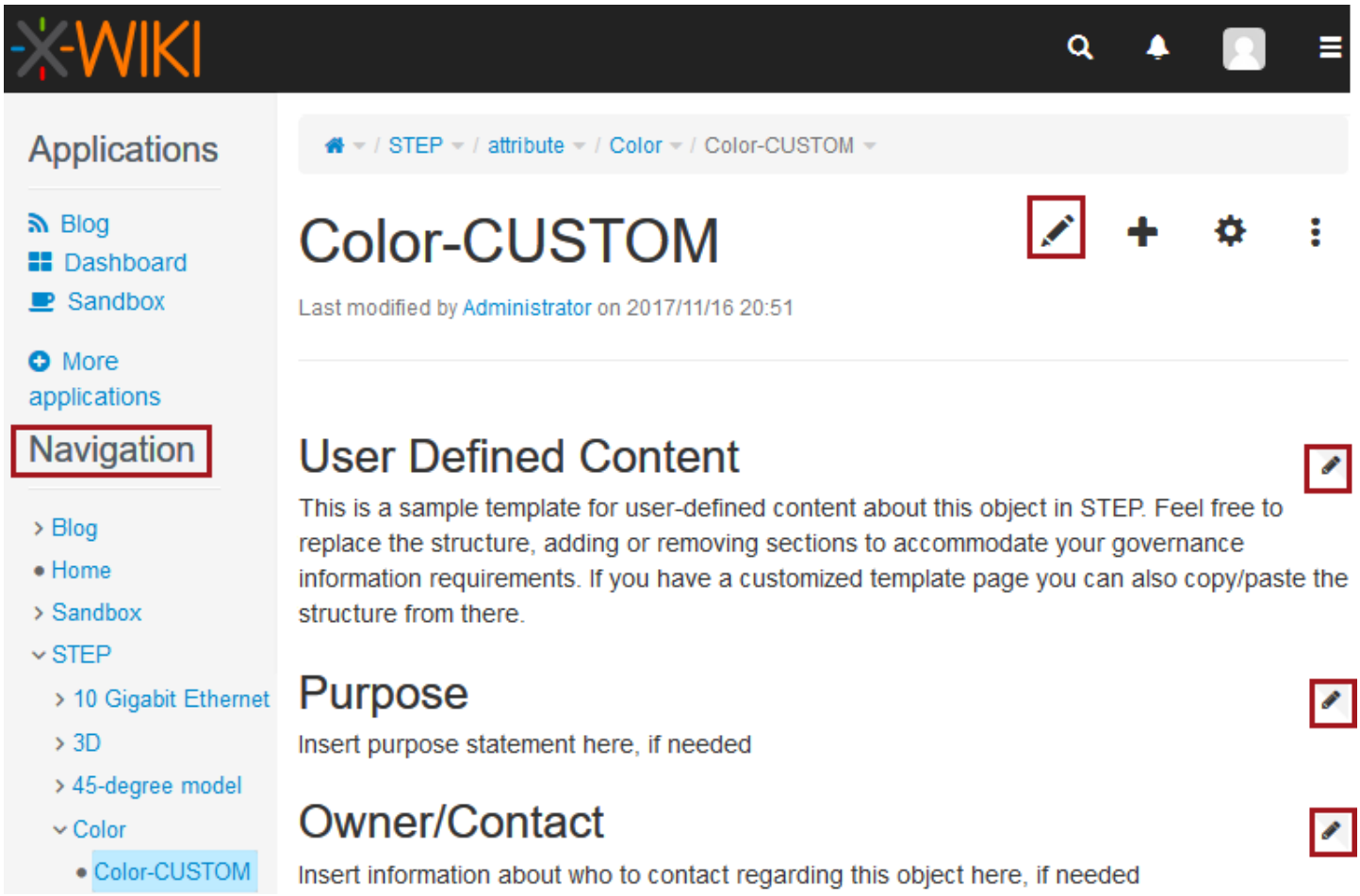
   The composite page is displayed as shown below and includes first the STEP metadata and then the CUSTOM data follows.



2. Log in to enable the editing functionality. Click the XWiki menu button ( ), then click the **Log-In** option to display the login dialog.



3. Type your username and password and click the **Log-in** button. Additional buttons are now available on the composite page.
4. Under the Navigation menu, select the **CUSTOM** version of the desired object. In the image below, the 'Color' attribute is being viewed, so the Color-CUSTOM page is selected.



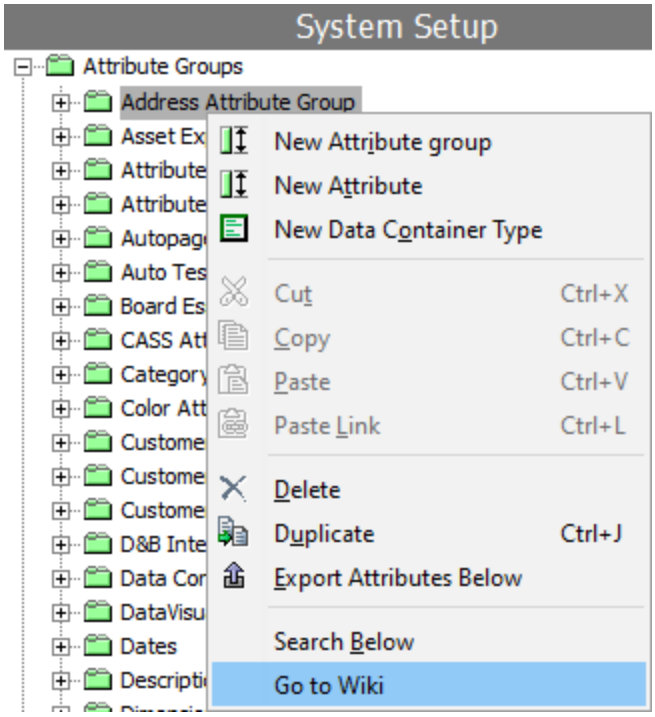
5. Click the edit button ( ) for the section that will be modified, or click the edit button in the wiki page header to modify all sections.
6. Make any updates.
7. Click the **Save & Continue** button to stay in edit mode, or click **Save & View** to return to the custom page.

Viewing Wiki Metadata Pages from Workbench

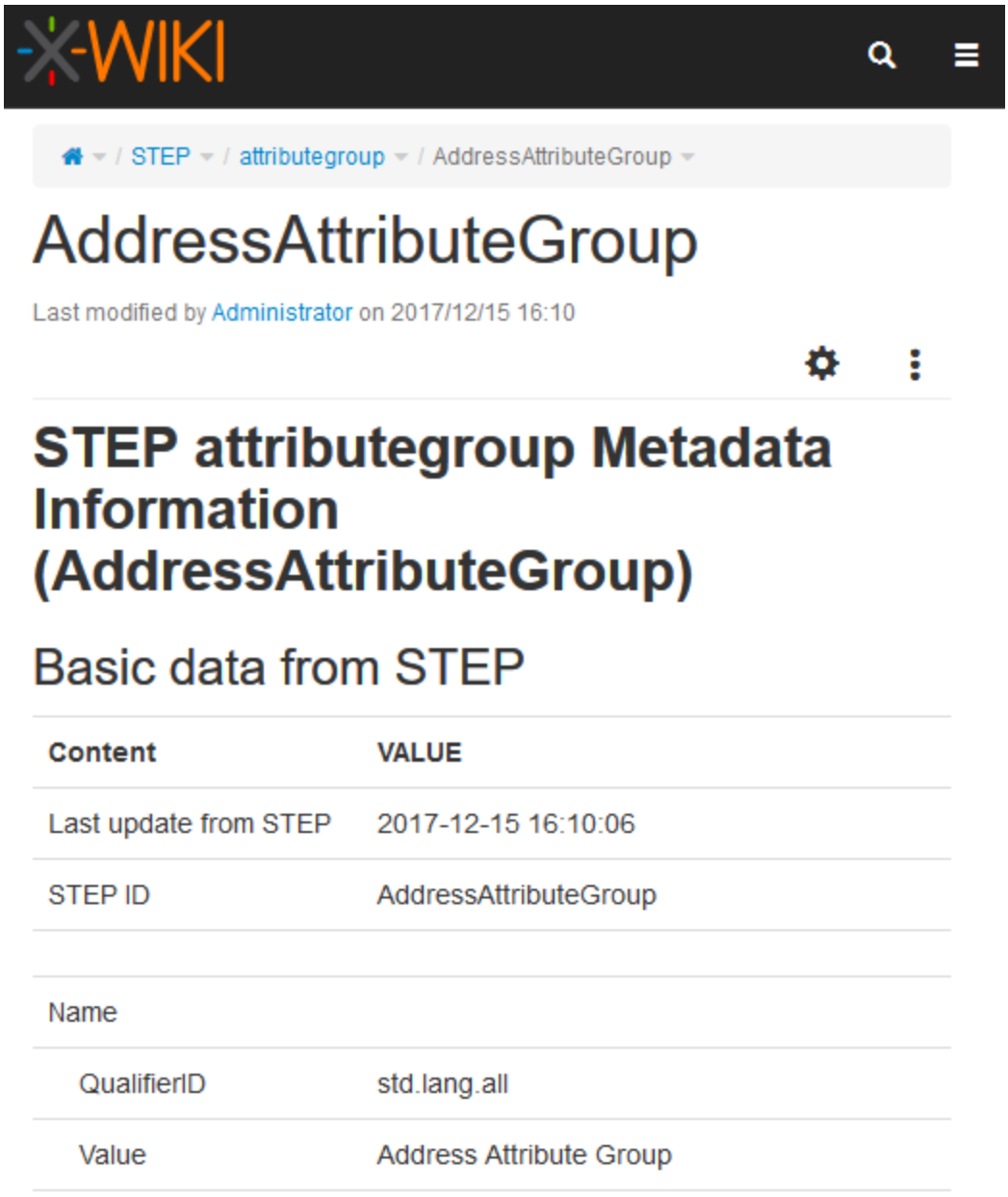
Once the setup is complete, the OIEP is running, and the wiki is running, no additional steps are required to access the wiki from workbench. Using the default setup, invoking the OIEP creates and modifies wiki pages.

The availability of a 'Go to Wiki' link is restricted only by the display of the system setup object and no additional privileges are required. Although many system setup objects include a wiki link, not all of them are supported in creating a wiki page. The objects that do trigger events for wiki are listed in the **Creating Wiki Metadata Pages** topic. For a list of objects that do not trigger events or wiki pages, see the **Events Not Triggered** topic in the **System Setup / Super User Guide** documentation.

In workbench, the 'Go to Wiki' link can be accessed from the right-click menu.

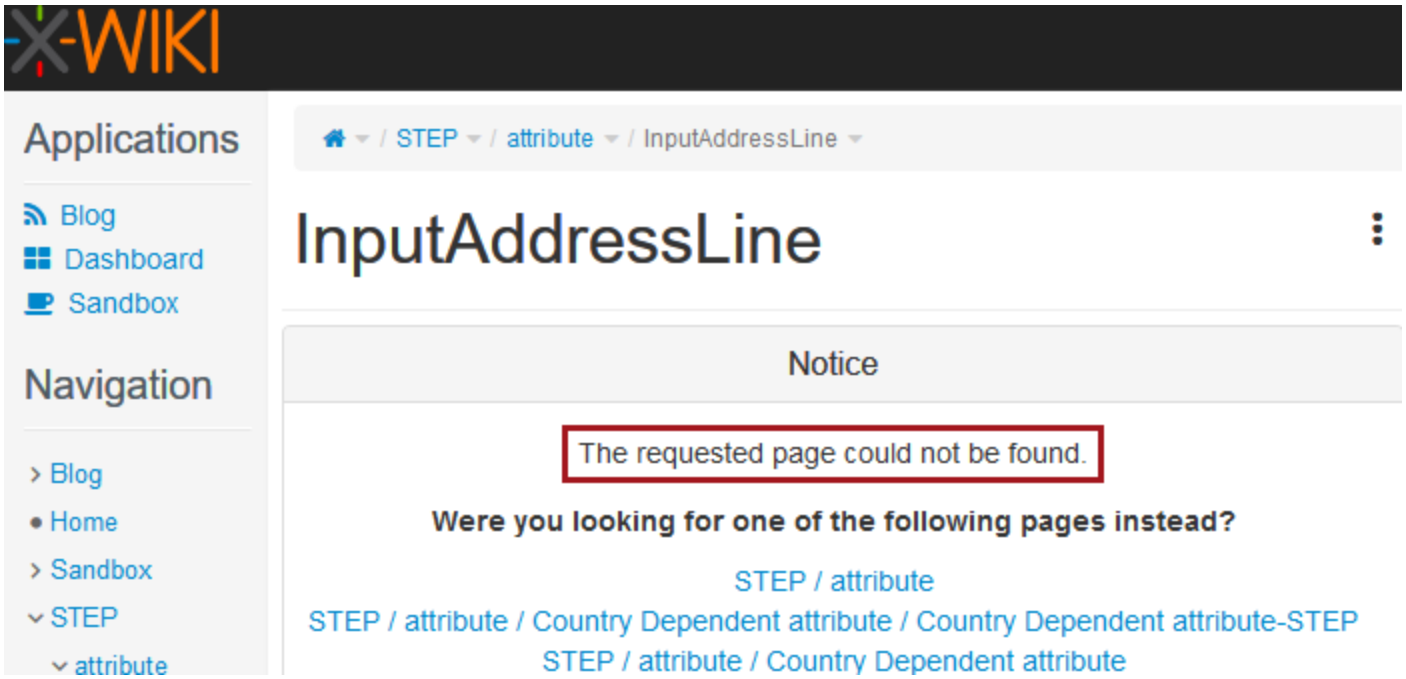


Click the link to display the wiki page in the user's default browser with the object's ID as the title.



Page Not Found

When a 'Go to Wiki' link is clicked but an event has not been processed by the OIEP to create a wiki page, as shown below, a message is displayed by the wiki that the requested page is not found. In this case, you can make a change on the object to trigger an event, or use the republish option to generate the missing page as described in the **Creating Wiki Metadata Pages** topic.


X-WIKI
Applications
Blog
Dashboard
Sandbox
Navigation
Blog
Home
Sandbox
STEP
attribute
STEP / attribute / InputAddressLine
InputAddressLine
Notice
The requested page could not be found.
Were you looking for one of the following pages instead?
STEP / attribute
STEP / attribute / Country Dependent attribute / Country Dependent attribute-STEP
STEP / attribute / Country Dependent attribute

Viewing Wiki Metadata Pages from Web UI

Once the setup is complete, the OIEP is running, and the wiki is running, additional steps may be required to access the wiki from Web UI as defined in the **Prerequisites** section below. Using the default setup, invoking the OIEP creates and modifies wiki pages.

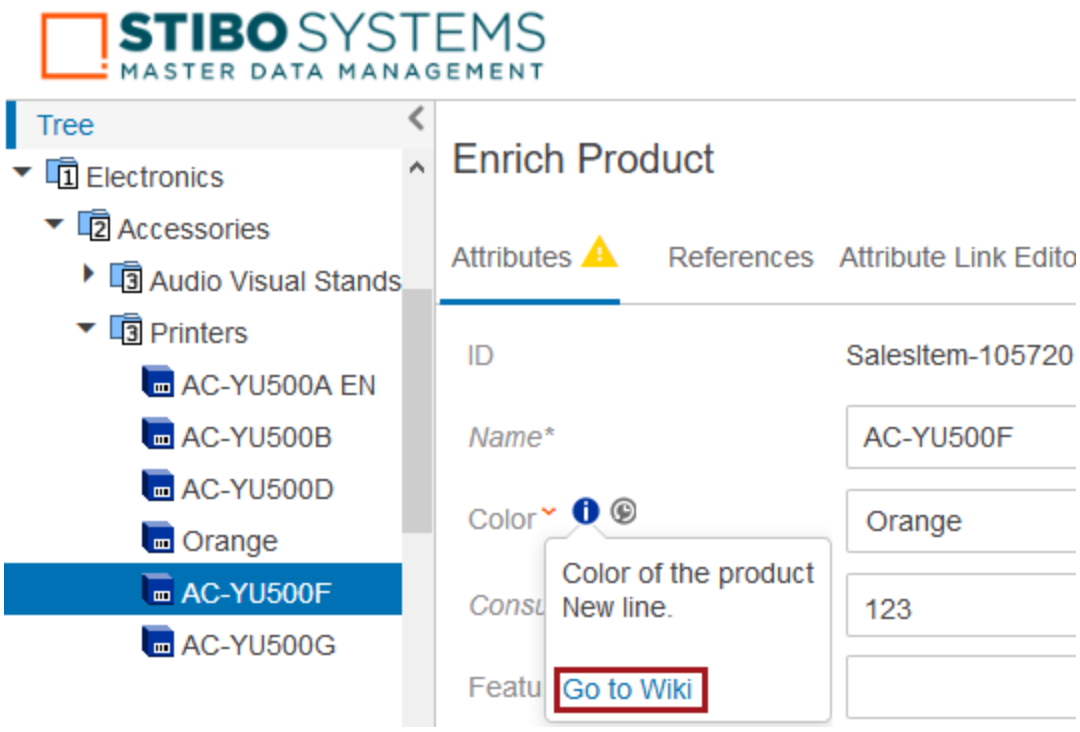
The availability of a 'Go to Wiki' link is restricted only by the display of the system setup object and no additional privileges are required. Although many system setup objects include a wiki link, not all of them are supported in creating a wiki page. The objects that do trigger events for wiki are listed in the **Creating Wiki Metadata Pages** topic. For a list of objects that do not trigger events or wiki pages, see the **Events Not Triggered** topic in the **System Setup / Super User Guide** documentation.

Prerequisites

1. For the 'Go to Wiki' link to be displayed in Web UI, the OIEP must exist in workbench. Creation of the OIEP is handled by the Wiki Metadata installation, as described in **Initial Setup for Wiki Metadata**.
2. Enabling the 'Go to Wiki' link involves the following setup, based on the screen or component, as defined in the following **Using a Web UI** topics:
   - For components that include the Display Context Help parameter, when that parameter is enabled, the link is displayed on the information popup. For configuration information, see the **Attribute Help Text in Web UI**.
   - For the **Attribute Link Editor Screen**, the Attribute Link Wiki Header can be added in the Headers parameter.
   - For the **Attribute Management Screen**, on the Tab Control child component > Tab Page (Attribute Details) > Node Editor > Row child component > add the Wiki Link Value component.
   - No additional configuration is needed to display the link on the Status Selector Homepage widget or the **Status Selector Homepage Widget**.

View Wiki Pages

In Web UI, the 'Go to Wiki' link can be accessed from the blue information icon which is displayed by hovering over a label.

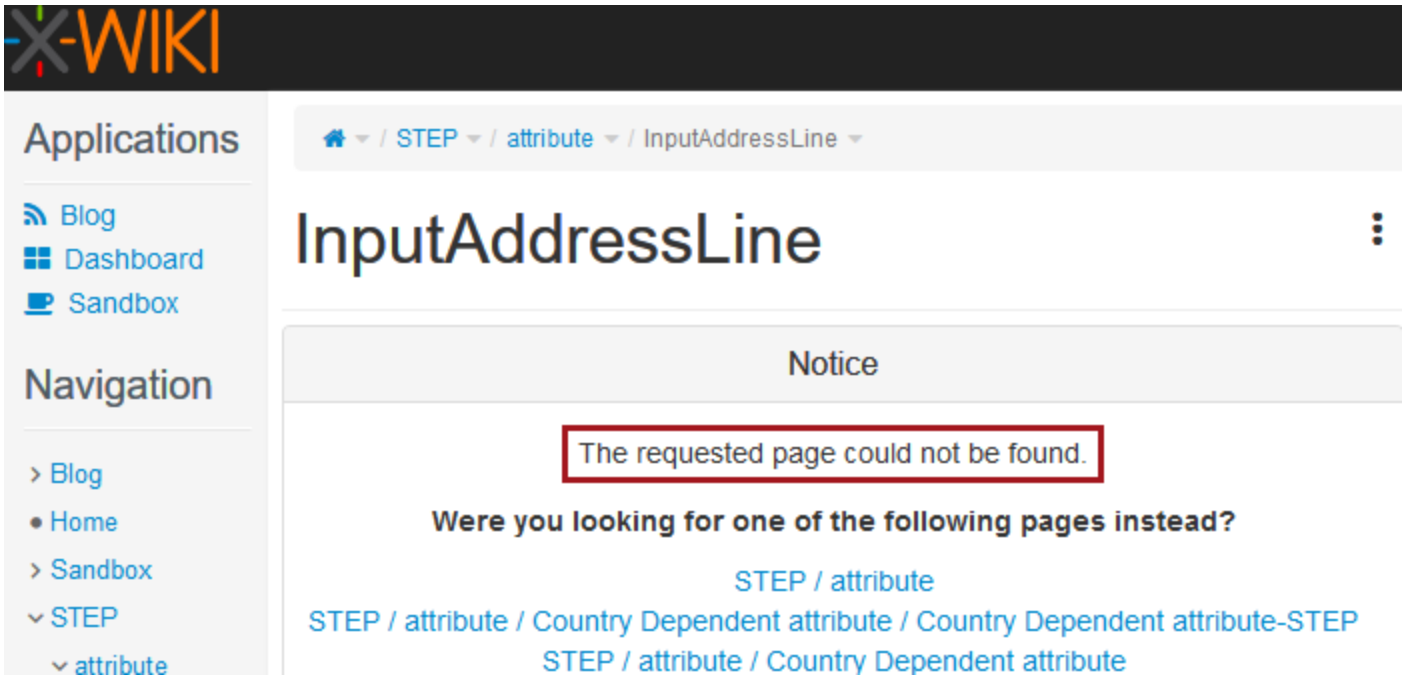


Click the link to display the wiki page in the user's default browser with the object's ID as the title.



Page Not Found

When a 'Go to Wiki' link is clicked but an event has not been processed by the OIEP to create a wiki page, as shown below, a message is displayed by the wiki that the requested page is not found. In this case, you can make a change on the object to trigger an event, or use the republish option to generate the missing page as described in the **Creating Wiki Metadata Pages** topic.


X-WIKI
Applications
Blog
Dashboard
Sandbox
Navigation
Blog
Home
Sandbox
STEP
attribute
STEP / attribute / InputAddressLine
InputAddressLine
Notice
The requested page could not be found.
Were you looking for one of the following pages instead?
STEP / attribute
STEP / attribute / Country Dependent attribute / Country Dependent attribute-STEP
STEP / attribute / Country Dependent attribute

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.

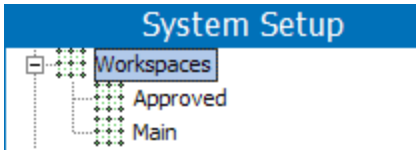
Hence, the Main workspace serves as an area for acquiring and maintaining data and the relation between data. Importing, enriching and categorizing data is done in the Main workspace. The Approved workspace stores data revisions that are deemed fit for production purposes - data that is ready to be published / 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.
- 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

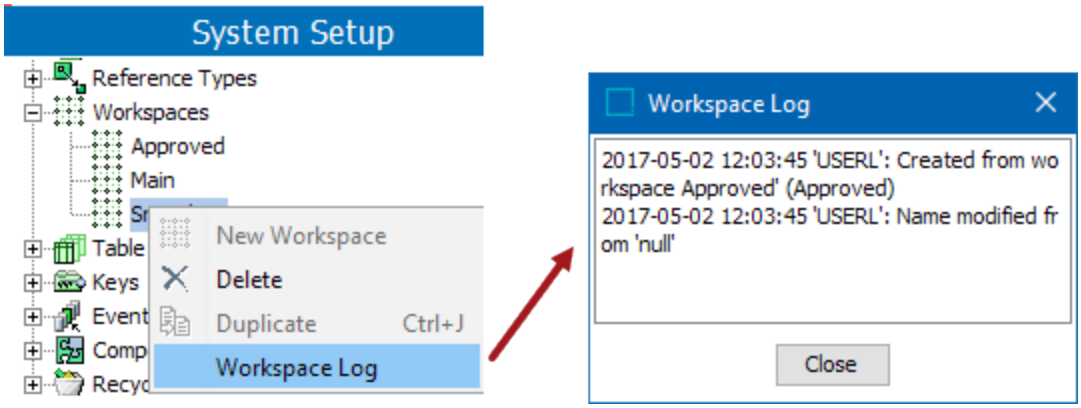
The following information is about how to properly maintain workspaces created outside of the Main and Approved workspaces. All workspaces are maintained in System Setup under the Workspaces node.

Creating Workspaces

As of the 9.2 release, it is no longer possible to create new workspaces. Workspaces created prior to 9.2 are still accessible, but cannot be duplicated.

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.

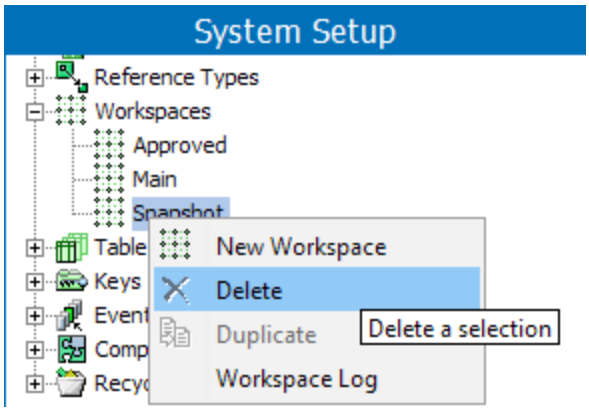


Deleting Workspaces

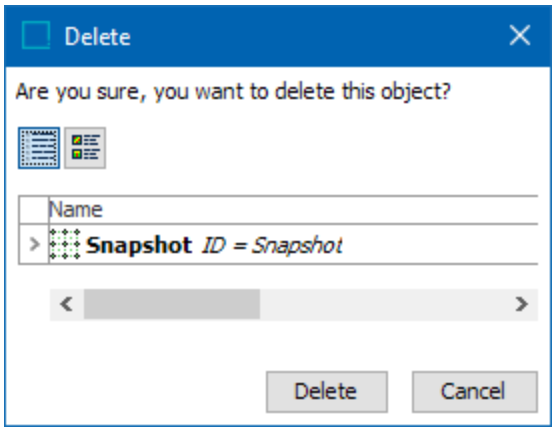
If a workspace is no longer needed, it can be deleted. Before deleting a workspace, make sure to transfer data that should be kept to another workspace before deleting. Deleting a workspace is very time consuming. The Main and Approved workspaces cannot be deleted.

Note: The workspace for a current Object is selected from the **Workspaces** list (upper right corner).

1. In System Setup, open **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.