

DATA EXCHANGE

USER GUIDE

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

StiboSystems

STEP Trailblazer 8.3

Table of Contents

Table of Contents	2
Data Exchange	21
Inbound Tools	21
Data Format	21
Receiver Method	21
Outbound Tools	21
Data Format	21
Delivery Method	22
Gateway Tool	22
Data Formats	23
Advanced STEPXML Format	26
Schema	26
Format Availability	26
Mapping	26
Inbound Data	27
Outbound Data	28
Alphabetical Index - XML Format	30
Format Availability	30
Mapping	30
Outbound Data	30
Ariba CIF Format	32
Format Availability	32
Mapping	32
Outbound Data	32
Ariba CIF 3.0 Format	34

Format Availability	34
Mapping	34
Outbound Data	34
BMEcat Format	36
Format Availability	36
Mapping	36
Inbound Data	36
Outbound Data	38
BMEcat 2005 Format	41
Format Availability	41
Mapping	41
Inbound Data	41
Outbound Data	43
CSV Format	46
Format Availability	46
Mapping	46
Inbound Data	46
Outbound Data	49
Exporting Data via JDBC with CSV Format	52
Install Required Drivers	52
Configure Properties	52
Select Format	53
Map Data	53
Select Delivery Method	55
Analytics using JDBC Example	57
Map Data	57
Select Delivery Method	59

cXML Format	61
Format Availability	61
Mapping	61
Outbound Data	61
eCl@ss Format	63
Format Availability	63
Mapping	63
Multiple Versions	63
Inbound Data	65
eCl@ss Classification Import	66
Map eCl@ss Attributes to STEP Attributes	69
ETIM Format	71
Format Availability	71
Mapping	71
ETIM Configuration	71
ETIM Versions	71
ETIM Classification	72
ETIM Feature Types	74
Valid Values are Determined by ETIM Version	75
Valid Units are Determined by Classification to Attribute Link	75
Handling Import Errors	76
ETIM and ETIM v2 Format	77
Format Availability	77
Mapping	77
Inbound Data	77
Inbound Parameters	77

Import Manager	78
IIEP	79
ETIM IXF Format	80
Format Availability	80
Mapping	80
Inbound Data	80
Create Product to ETIM Classification Link	83
Set Validity on ETIM Attributes for Products	86
Manually Import ETIM Attribute Values	91
Excel Format	104
Format Availability	104
Mapping	105
Inbound Data	105
Outbound Data	107
Importing Multi-Valued References or Links with Excel	109
Excel Custom Template	119
Format Availability	119
Mapping	119
Output	119
Excel Custom Template Configuration	121
Configuration	121
Excel Smartsheet Format	132
Format Availability	132
Setup Requirements	132
Additional Information	133
Initiating Products Using a Smartsheet	134

Prerequisites	134	System Configuration	162
Product Initiation	134	Smartsheet Cover Tab	162
Exporting a Template with a Supplier Selector	138	Smartsheets Special Attribute Handling	163
Maintaining Products Using a Smartsheet 140		Conditional Attributes	163
Prerequisites	140	Workflow State Mandatory Attributes	164
Product Maintenance	140	Smartsheet Background Processes and Queues	165
Smartsheet Data and Template Configurations	144	Using Smartsheet Import Background Process	165
Export Manager Wizard	144	Using Smartsheet Export Background Process	166
Select Objects Step	144	Smartsheet System Properties	168
Select Format Step	145	Limit supplier browsing	168
Map Data Step	150	Service URL override	168
Advanced Step	152	Trusted headers authentication	168
Select Delivery Method Step	152	FixedWidth Format	170
Saving and Modifying the Configuration ...	152	Format Availability	170
Additional Configuration Options	152	Mapping	170
Smartsheet Import Configurations	154	Inbound Data	170
Creating an Import Configuration	154	FixedWidth Format Configuration Examples	173
Select Format Step	154	Example Data	173
Advanced Settings Step	154	End Positions (Zero Based) Example	173
Saving and Modifying the Configuration ...	156	End Positions (One Based) Example	175
Using a Smartsheet	157	Width Example	177
Copying Data in a Smartsheet Template ..	159	Flatplan Excel Format	179
Smartsheet Types	160	Format Availability	179
Multi Level Smartsheets	160	Mapping	179
Multiple Product Object Type Smartsheets	161	Inbound Data	179
Additional Information About Smartsheets	162		

Outbound Data	180
Generic XML Format	182
Limitations	182
Format Availability	182
Mapping	182
Inbound Data	182
Outbound Data	184
Generic XML Inbound Processing Instructions	187
DimensionPointID Instruction in Generic XML	188
MultiSource Instruction in Generic XML	190
Record Instruction in Generic XML	191
Repeated Instruction in Generic XML	192
Source Instruction in Generic XML	195
Preventing Duplicate Field Names	195
SourceID Instruction in Generic XML	196
Generic XML Import Simple Example	199
Input Document	199
Template	200
Conversion Preview	201
Generic XML Import Advanced Example	202
Input Document	202
Creating a Template	203
Remove Redundant Elements	203
Add Record and Source Instructions	204
Add Repeated Instruction	205
Add Repeated Instruction With a Filter	207

Add MultiSource Instruction	208
Conversion Preview	209
Generic XML Outbound Processing Instructions	210
Allow Empty Tags Parameter in Generic XML	211
Attributes Instruction in Generic XML	214
Attribute Aspects in Generic XML	218
Attribute Links and LOVs Instruction in Generic XML	221
Mapping Attribute Links, LOVs, and LOV Values in Generic XML	225
Attribute Links and Units Instruction in Generic XML	232
Mapping Attribute Links, Units, and Unit Base in Generic XML	236
DoNotEmit Instruction in Generic XML	240
List of Values Object Type in Generic XML	243
Multi-Valued Objects in Generic XML	247
Multi-Cardinality Tags in Generic XML	249
Repeated Targets	250
MultiTarget	251
Composite Mapping Targets	252
Nestable Data Paths in Generic XML	254
Data Path Concept	254
Nesting Concept	254
Valid Path Step Types	255
Data Sources Available for Data Paths	255
Nestable Data Paths in Generic XML Example	257

Data Path Configuration	258
Mapping for Nestable Data Paths in Generic XML Example	261
RemovelfEmpty Instruction in Generic XML	270
Reporting Deleted Assets with a Generic XML OIEP	273
Object Type and Event Type Selection	273
Template	274
Mapping	275
Results	275
Reporting Deleted Classifications with a Generic XML OIEP	276
Object Type and Event Type Selection	276
Template	277
Mapping	278
Result	278
Reporting Deleted Entities with a Generic XML OIEP	279
Object Type and Event Type Selection	279
Template	280
Mapping	281
Results	281
Reporting Deleted Product References with a Generic XML OIEP	282
Product	282
Object Type and Event Type Selection	282
Template	283
Mapping	284
Results	284

Mapping to Report Deleted Product References with a Generic XML OIEP	286
Single-Cardinality Tags in Generic XML ...	295
Record	296
Target Default	297
Target Override Default	297
Units with a Generic XML OIEP	298
Object Type and Event Type Selection	298
Template	299
Mapping	299
Results	300
Mapping Unit Aspects with a Generic XML OIEP	301
Generic XML Sample Export	303
Global Product Classification Format	305
Format Availability	305
Mapping	305
Limitations	305
Setup Requirements	305
Inbound Data	306
Global Product Classification Import	308
Create a GPC Product to Classification Link Type	310
IDoc MATMAS 05 Format	314
Format Availability	314
Mapping	314
Inbound Data	314
Outbound Data	316
Publication Excel Format	319

Format Availability	319
Mapping	319
Inbound Data	319
Outbound Data	320
SmartLabel Format	322
Format Availability	322
Mapping	322
Outbound Data	322
Configuring the SmartLabel Template	325
SmartLabel Attributes	325
Modifying the SmartLabel Template	326
STEPXML Format	329
Limitations	329
Format Availability	329
Mapping	330
Inbound Data	330
Outbound Data	331
STEPXML Outbound Parameters	334
Global Settings	334
Data Objects	335
Configuration	339
Publishing	344
STEPXML Tags and Examples	346
AssetContent Tag in STEPXML	347
Asset Content Input	347
Asset Content Output	348
Assets Tag in STEPXML	351

Name	351
Values	351
AttributeLink Tag in STEPXML	354
IncludeInherited	354
AutoApprove in STEPXML	356
Backward Compatibility in STEPXML	357
Classifications Tag in STEPXML	359
IncludeParent	359
Additional STEPXML Tags	359
Name	359
Values	359
ClassificationReference	359
MetaData	359
Product to Classification Links Owned by the Classification	360
Create Objects in STEPXML	361
Data Containers in STEPXML	363
Setting Up Data Containers	363
Adding and Maintaining Data Containers	364
Delete Objects in STEPXML	367
Event-Based OIEP Triggered by Deleting Products, Classifications, and Assets	367
Delete Product Reference in STEPXML	369
Delete Values in STEPXML	372
Export Inheritance in STEPXML Example	373
Filter Products in STEPXML	375
Filter References in STEPXML	379
Asset References	379

Classification References	379	Values Attributes Example	
Product References	379	Product to Classification Links Owned by the Classification	403
Results	380	Referenced and Embedded XML	
Filter Values in STEPXML	382	Attributes in STEPXML	405
IncludeKey Tag in STEPXML	383	ReplacementRules Tag in STEPXML	411
Inherit References in STEPXML	385	STEP-ProductInformation Tag	411
Inherit Values in STEPXML	388	ReplacementRules Tag	412
IncludeInherited	388	SequenceProduct Tag in STEPXML	413
IncludeInheritedAttributes	388	SingleUpdateMode in STEPXML	417
Minimum, Referenced, and Selected in STEPXML	392	STEP-ProductInformation Tag in STEPXML	418
Hierarchy Export Including Objects Referenced Example	393	DefinitionsAsComments	418
Product Export Including Definitions, LOVs, and Units for Attributes Example	395	EventsAsComments	418
Multi-Context Data in STEPXML	398	ExportDeletedData	418
Products Tag in STEPXML	400	ExportDerivedAttrs	419
ExportSize	400	FollowOverrideSubProducts	419
IncludeParent	400	ResolveInlineRefs	420
Additional STEPXML Tags	400	Validation	420
Name	400	UNSPSC Format	421
Values	401	Format Availability	421
AssetCrossReference	401	Mapping	421
ProductCrossReference	401	Inbound Data	421
ClassificationReference	401	xCBL Format	425
Product Nesting Example	401	Format Availability	425
Nested structure	401	Mapping	425
Non-nested structure	402	Outbound Data	425
Product with Units, References, and Multi-	402	Data Mapping	428
		Inbound Data Mapping	428

Outbound Data Mapping	428
Inbound Map Data Options	429
Formats	429
Map To Object Type	430
Mapping Data	430
Modifying Data	430
Profiling Data	431
Tips for Mapping Data	431
Inbound Map Data - Auto Map	432
Predefined Header Text	432
Auto Map Example	433
Inbound Map Data - Constant	435
Mapping a Constant	435
Inbound Map Data - Generate Profile	439
Profiling Setup	439
Profiling Data	439
On the Fly Data Validation	444
Inbound Map Data - Map	445
Mapping the Same Column Multiple Times	445
Mapping Options	446
Add Child to Override - Map Inbound	447
Asset Reference - Map Inbound	450
Attribute - Map Inbound	457
Attribute Info - Map Inbound	462
Attribute Info File Structure Requirements	464
Attribute Validation Info - Map Inbound	467
Attribute Validation Info File Structure Requirements	469

Classification Reference - Map Inbound	472
Data Container Attribute - Map Inbound	475
Delete Values - Map Inbound	479
Entity Reference - Map Inbound	481
Entity Reference via Source Record ID - Map Inbound	486
Prerequisites	486
Map Entity References	486
ID or Key - Map Inbound	490
Mapping ID	490
Mapping Key	493
Multivalued Variable - Map Inbound	495
Name - Map Inbound	499
Object Type - Map Inbound	501
Map to Create New Items with Object Type	502
Map to Change the Existing Object Type	503
Overrides Product - Map Inbound	506
Parent - Map Inbound	509
Product Classification Links - Map Inbound	511
Product Reference - Map Inbound	514
Reference Meta-Data - Map Inbound	517
Variable - Map Inbound	522
Inbound Map Data - Map To	526
Mapping Based on Object Type	526
Asset Object Type - Map To Inbound	527
Attribute Object Type - Map To Inbound	528
Classification Object Type - Map To Inbound	529

Entity Object Type - Map To Inbound	531
Product Object Type - Map To Inbound	533
Inbound Map Data - Remove	535
Remove a Mapped Column	535
Inbound Map Data - Transform	537
Transforming Data	537
Dimension Dependent Data	538
Inbound Transformation Options	539
Outbound Map Data Options	547
Formats	547
Mapping in Outbound Tools	547
Mapping	549
Transformations	549
Outbound Map Data - Data Source	550
Available Data Source Options	550
Asset References - Data Source Outbound	552
Attributes (and Data Containers) - Data Source Outbound	556
Classification References – Data Source Outbound	560
Classifications - Data Source Outbound	562
Constant Value - Data Source Outbound	564
Custom Attributes - Data Source Outbound	568
Data Path - Data Source Outbound	569
Entity References – Data Source Outbound	570
ID or Key - Data Source Outbound	572
Index Words – Data Source Outbound	575

Mapping Index Words Linked to Products Using Word	576
Mapping Index Words Linked to Products Using Sort Word	577
Mapping Index Words from Classification	578
Inherit Data and References - Data Source Outbound	580
Insert Referenced Objects - Data Source Outbound	583
Is Deleted - Data Source Outbound	586
Multi Level Parent Attributes - Data Source Outbound	587
Multi Level References - Data Source Outbound	591
Name - Data Source Outbound	600
Object Type Name - Data Source Outbound	602
Page Number – Data Source Outbound	604
Parent ID - Data Source Outbound	605
Product Classification Links – Data Source Outbound	607
Product-Override Child ID – Data Source Outbound	610
Product References - Data Source Outbound	612
STEP Workflow Task Info - Data Source Outbound	615
System Setup - Data Source Outbound	622
Outbound Map Data - Mapping Targets	623
Tabular Format Targets	623
XML Format Targets	624
Outbound Map Data - Transform	626

Transforming Data	626
Outbound Add Transformation Options	627
Data Container Separator - Transform Outbound	632
Multi-Value Separator - Transform Outbound	635
Aspect - Transform Outbound	638
Attribute Metadata Aspect	644
Attribute Type Aspect	646
Base Unit Aspects	648
Closest Attribute Link Meta Data Aspect	652
Closest Attribute Link Meta Data Aspect and Display Sequence	654
Dimension Dependencies Aspects	658
LOV Value-ID or Value Aspect	661
Unit Aspects	662
Transformation Examples	664
Concatenate Column with Unit Data Example	665
Concatenate Data Using Variables Example	669
Concatenate into Single-Valued Data Example	674
Dimension Point for Single Dimension Data Example	677
LOV Value-ID or Value Aspect	681
Extract Characters at Position Example	682
Extract Characters Between and Extract Characters After Example	685
Format Number Example	690
Header and Value Aspects Example	693

If Equals Source Example	699
Insert Text Example	701
Math Operation Example	706
Merge Attributes and Constant Example	710
Merge Data for Multi-Valued Data Example	713
Replace Substrings of the Value Example	724
Split and Extract Data Example	727
Replace the Whole Value Example	734
Transform Date by Locale or by Pattern	740
Export Transformations	741
Import Transformations	744
Transform Using Attribute Transformation Example	748
Transformation Lookup Table Example	750
Export Manager	755
Setup Requirements	755
Additional Information	755
Creating a Data Export	756
Considerations for Exporting Products	760
Export Manager - Select Configuration	761
Export Manager - Select Objects	763
Export Manager - Select Format	766
Export Manager - Map Data	767
Export Manager - Advanced	768
Export Manager - Select Delivery Method	770
Email Delivery Method	771
File Delivery Method	773

FTP Delivery Method	774
JDBC Delivery Method	776
Server Side Delivery Method	778
SFTP Delivery Method	780
Websphere Commerce Import Delivery Method	782
Running a Data Export	783
Maintaining a Saved Export Configuration	785
View a Saved Export Configuration	785
Edit a Saved Export Configuration	785
Monitoring a Data Export	788
Monitoring the Export Process	788
Execution Report	789
Result	791
Scheduling a Data Export	792
Scheduled Data Export - Select Export Configuration	794
Scheduled Data Export - Schedule Process	795
Scheduled Data Export - Name Process	796
Scheduled Process Properties	797
Deleting a Scheduled Data Export	799
Gateway Integration Endpoints	802
Setup Requirements	802
Additional Information	802
Initial Setup for a Gateway Integration Endpoint	803
Create the Gateway Setup Group	803
Link the Gateway Object Type to Setup Group	804

Create an Instance of the Gateway Endpoint Object	805
Creating a Gateway Integration Endpoint	807
Configuring a Gateway Integration Endpoint	808
Proxy Configuration Setup	810
Gateway Integration Endpoint Bind	811
Running a Gateway Integration Endpoint	814
Maintaining a Gateway Integration Endpoint	816
Monitoring a Gateway Integration Endpoint	817
Monitor from STEP	817
Monitor Outside of STEP	817
View all endpoints	817
View a specific gateway endpoint	817
Gateway Integration Endpoint Statistics	819
Exporting Gateway Integration Endpoint Definitions as Comments	820
Import Manager	821
Setup Requirements	821
Additional Information	821
Creating a Data Import	822
Import Manager - Select Configuration	824
Import Manager - Select Data Source	826
Using FTP	826
Using File	827
Import Manager - Select Format	828
Import Manager - Map Data	830
Import Manager - Identify Objects	831

Importing With Object IDs	831
Importing with Unique Key	832
Importing Without Object IDs	832
Import Manager - Identify Destination	835
Import Manager - Select Business Rules	837
Business Rule Limitations	838
Import Manager - Advanced Settings	840
General Advanced Settings	841
Remove Options	842
Adding and Removing Un-Mapped References	846
Running a Data Import	848
Maintaining a Saved Import Configuration	850
View a Saved Import Configuration	850
Edit a Saved Import Configuration	850
Monitoring a Data Import	852
Monitoring the Import Process	852
Properties Flipper	853
Execution Report	854
Reimporting a Corrected File	855
Import Error Messages	856
Import Error Message Examples	860
Illegal Unit for Attribute	860
Unknown Parent Product	860
Maximum Length, Illegal Value	861
Out of Range, Illegal Value	862
Not a Number, Illegal Value	862
Object Existed with Another Object Type	863

Unknown Classification	864
Not Privileged to Create Object	864
Unknown Asset ID	865
Array Operation Failed	865
Optimistic Locking Errors	865
The Transaction has been Rolled Back: Unique Constraint	865
Inbound Integration Endpoints	870
Setup Requirements	870
Additional Information	870
Initial Setup for an Inbound Integration Endpoint	872
Create the Inbound Integration Setup Group	872
Link the IIEP Object Type to Setup Group	874
Create an Instance of the IIEP Object	875
Creating an Inbound Integration Endpoint	877
IIEP - Identify Endpoint	879
IIEP - Choose Receiver	880
IIEP Receiver Methods	881
Hotfolder Receiver	883
Hotfolder Receiver Using File Sequence	885
Hotfolder Receiver Using Meta Files	887
IBM Websphere MQ SSL Receiver	889
JMS Receiver	891
Configuration Examples	892
FileInitialContextFactory JMS Receiver	893
JMS Websphere Delivery Using SSL	893
Apache ActiveMQ JMS Receiver	894

Oracle AQ Receiver	895
REST Receiver	897
IIEP - Configure Endpoint	899
IIEP - Configure PreProcessor	901
IIEP - Configure Transformation by XSLT PreProcessor	902
Initial Setup for XSLT Stylesheets	902
Create an XSLT Stylesheet Object	906
Configure the Transformation by XSLT Preprocessor	908
IIEP - Configure XML Normalizer PreProcessor	910
IIEP - Configure Processing Engine	916
IIEP - Configure Asset Importer Processing Engine	917
Prerequisites	917
Configuration	918
Metadata File Format	921
IIEP - Configure STEP Importer Processing Engine	923
IIEP - Configure Match and Merge Importer	926
Match and Merge Importing Process and Logic	926
Configuration Wizard Steps	927
Source Record ID in STEPXML	930
IIEP - Configure PostProcessor	932
IIEP - Schedule Endpoint	933
IIEP - Configure Error Reporter	936
Running an Inbound Integration Endpoint	937

Enable and Invoke an Inbound Integration Endpoint	937
Disable an Inbound Integration Endpoint	940
Maintaining an Inbound Integration Endpoint	941
Inbound Integration Endpoint Tab	941
Description Flipper	941
Configuration Flipper	942
Receiver Configuration Flipper	943
Integration Endpoint Log Flipper	944
Background Processes Tab	944
Statistics Tab	945
Error Log Excerpts Tab	946
Log Tab	946
Status Tab	946
Monitoring an IIEP via Background Process	947
Selecting the Background Process Queue ..	948
Background Process Queue Size Property ..	948
Monitoring an IIEP via External System ...	949
Monitoring Sensors	949
REST API Monitoring	949
Import Error Messages	950
Import Error Message Examples	954
Illegal Unit for Attribute	954
Unknown Parent Product	954
Maximum Length, Illegal Value	955
Out of Range, Illegal Value	956
Not a Number, Illegal Value	956

Object Existed with Another Object Type	957
Unknown Classification	958
Not Privileged to Create Object	958
Unknown Asset ID	959
Array Operation Failed	959
Optimistic Locking Errors	959
The Transaction has been Rolled Back: Unique Constraint	959
Additional Information About Inbound Integration Endpoints	964
Exporting Inbound Integration Endpoint Definitions as Comments	965
Handling Failed IIEP Background Processes	967
Disable and Enable a Background Process	967
Resume a Background Process	967
Resolving a Failed Background Process	968
Inbound Integration Endpoint Structure ..	970
Receiver	970
Preprocessor	970
Processing Engine / Converter	970
Error Reporter	971
Postprocessor	971
Inbound Integration Endpoint Transactional Settings	972
Strict	972
Chain	972
None	973
Reference Target Lock Policy on Object Types	974

Outbound Integration Endpoints	975
Event-Based Outbound Integration Endpoint	975
Setup Requirements for Event-Based OIEPs	975
Additional Information for Event-Based OIEPs	976
Select Objects Outbound Integration Endpoint	976
Setup Requirements for Select Objects OIEPs	976
Additional Information for Select Objects OIEPs	977
Initial Setup for an Outbound Integration Endpoint	978
Create the Outbound Integration Setup Group	978
Link the OIEP Object Type to Setup Group ..	980
Create an Instance of the OIEP Object	981
Creating an Outbound Integration Endpoint	983
Creating an Event-Based Outbound Integration Endpoint	984
OIEP - Event Based - Identify Endpoint	986
OIEP - Event-Based - Choose Data Source	987
OIEP - Event Based - Configure Endpoint	988
OIEP - Event-Based - Manual Configuration	990
OIEP - Configuration Flipper	994
Process Engine	994
Error Reporter	994
Schedule	995

Queue for Endpoint	998	Copy to Directory	1016
Queue for Endpoint Processes	998	Deploy	1016
Transactional Settings	998	Email	1017
Number of Threads	998	FTP	1017
Maximum Number of Old Processes	998	IBM Websphere MQ SSL Delivery	1017
Maximum Age of Old Processes	998	JDBC	1018
Contexts	999	JMS	1018
Workspace	999	Mongo	1019
OIEP - Event-Based - Output Templates		Oracle AQ	1019
Flipper	1000	REST	1019
Configure the Object Types and Event		REST Direct	1020
Types	1000	SFTP	1020
Using Multiple OIEPs or Multiple Output		Wiki	1020
Templates	1002	OIEP Delivery Methods	1022
Multiple OIEPs	1002	Amazon SQS Delivery Method	1025
Multiple Output Templates	1002	Copy to Directory Delivery Method	1028
Limitations of Multiple Output Templates	1002	Deploy Delivery Method	1030
Configure the Format	1003	Email Delivery Method	1032
Configure the Pre-processor and Post-		FTP Delivery Method	1033
processor	1006	IBM Websphere MQ SSL Delivery	
Pre-processor	1006	Method	1035
Post-processor	1007	JDBC Delivery Method	1037
OIEP - Pre-Processor - Business Action	1009	JMS Delivery Method	1039
Business Action Configuration	1009	Sharedconfig.properties File Examples	1040
OIEP Configuration	1010	Mongo Delivery Method	1043
OIEP - Post-Processor - Context Splitter	1011	Mongo Delivery Method Elements	1047
OIEP - Post-Processor - Copy Context		Default Databases and Collections	1049
Dependent Values and References	1013	STEP JSON and Mongo JSON	1050
OIEP - Delivery Method Flipper	1016		
Amazon SQS	1016		

JavaScript Triggers	1050
Bound Variables	1051
Example JavaScript to Maintain a Collection in a Classification	1051
Prerequisites for Configuring the MongoDB Adapter	1053
Configure the OIEP	1053
Configuring the JavaScript Triggers	1055
MongoDB Adapter Setup Quick Guides ..	1056
Configuring Mongo Authentication Quick Guide	1056
SSL Configuration Quick Guide	1057
Mongo Delivery Method Conversion Example	1059
Values	1059
References and Links	1061
Attribute Types	1064
Reference Types	1064
Asset Push Locations	1064
Tables	1065
Table Formatting	1065
Attribute Links	1067
Unit	1067
List Of Values	1068
Oracle AQ Delivery Method	1069
REST Delivery Method	1070
REST Direct Delivery Method	1072
Additional Request Information	1075
SFTP Delivery Method	1077

Wiki Delivery Method	1079
OIEP - Event-Based - Event Triggering Definitions Tab	1081
Triggering Object Types Flipper	1082
Triggering Attributes Flipper	1087
Triggering Table Types Flipper	1088
Reference Type Triggers Flipper	1088
Triggering Data Container Types	1089
Miscellaneous Triggers Flipper	1090
OIEP - Event-Based - Event Queue Configuration Flipper	1092
Creating a Select Objects Outbound Integration Endpoint	1094
OIEP - Select Objects - Identify Endpoint	1096
OIEP - Select Objects - Choose Data Source	1097
OIEP - Select Objects - Configure Endpoint	1099
OIEP - Select Objects - Manual Configuration	1101
OIEP - Configuration Flipper	1104
Process Engine	1104
Error Reporter	1104
Schedule	1105
Queue for Endpoint	1108
Queue for Endpoint Processes	1108
Transactional Settings	1108
Number of Threads	1108
Maximum Number of Old Processes	1108
Maximum Age of Old Processes	1108

Contexts	1109
Workspace	1109
OIEP - Select Objects - Object Selection Configuration Flipper	1110
OIEP - Select Objects - Output Templates Flipper	1113
Configure the Object Types	1113
Using Multiple OIEPs or Multiple Output Templates	1115
Limitations of Multiple Output Templates	1115
Configure the Format	1115
Configure the Pre-processor and Post-processor	1119
Pre-processor	1119
Post-processor	1119
OIEP - Pre-Processor - Business Action	1121
Business Action Configuration	1121
OIEP Configuration	1122
OIEP - Post-Processor - Context Splitter	1123
OIEP - Post-Processor - Copy Context Dependent Values and References	1125
OIEP - Delivery Method Flipper	1128
Amazon SQS	1128
Copy to Directory	1128
Deploy	1128
Email	1129
FTP	1129
IBM Websphere MQ SSL Delivery	1129
JDBC	1130
JMS	1130

Mongo	1131
Oracle AQ	1131
REST	1131
REST Direct	1132
SFTP	1132
Wiki	1132
OIEP Delivery Methods	1134
Amazon SQS Delivery Method	1137
Copy to Directory Delivery Method	1140
Deploy Delivery Method	1142
Email Delivery Method	1144
FTP Delivery Method	1145
IBM Websphere MQ SSL Delivery Method	1147
JDBC Delivery Method	1149
JMS Delivery Method	1151
Sharedconfig.properties File Examples	1152
Mongo Delivery Method	1155
Mongo Delivery Method Elements	1159
Default Databases and Collections	1161
STEP JSON and Mongo JSON	1162
JavaScript Triggers	1162
Bound Variables	1163
Example JavaScript to Maintain a Collection in a Classification	1163
Prerequisites for Configuring the MongoDB Adapter	1165
Configure the OIEP	1165
Configuring the JavaScript Triggers	1167

MongoDB Adapter Setup Quick Guides ..1168

Configuring Mongo Authentication Quick Guide 1168

SSL Configuration Quick Guide 1169

Mongo Delivery Method Conversion Example 1171

Values 1171

References and Links 1173

Attribute Types 1176

Reference Types 1176

Asset Push Locations 1176

Tables 1177

Table Formatting 1177

Attribute Links 1179

Unit 1179

List Of Values 1180

Oracle AQ Delivery Method 1181

REST Delivery Method 1182

REST Direct Delivery Method 1184

Additional Request Information 1187

SFTP Delivery Method 1189

Wiki Delivery Method 1191

Running an Outbound Integration Endpoint 1193

Enable and Invoke an Outbound Integration Endpoint 1193

Disable an Outbound Integration Endpoint 1196

Maintaining an Outbound Integration Endpoint 1198

Outbound Integration Endpoint Tab 1198

Description Flipper 1198

Integration Endpoint Log Flipper 1198

Configuration Tab 1200

Configuration Flipper 1200

Event Queue Configuration Flipper 1202

Object Selection Configuration Flipper ...1203

Output Templates Flipper 1203

Delivery Method Flipper 1204

Background Processes Tab 1204

Statistics Tab 1205

Error Log Excerpts Tab 1206

Log Tab 1206

Status Tab 1207

Monitoring an OIEP via Background Process 1209

Selecting the Background Process Queue 1210

Background Process Queue Size Property 1210

Monitoring an OIEP via External Systems 1211

Monitoring Sensors 1211

REST API Monitoring 1211

Additional Information About All Outbound Integration Endpoints 1213

Exporting Outbound Integration Endpoint Definitions as Comments 1214

Handling Failed OIEP Background Processes 1216

Disable and Enable a Background Process 1216

Resume a Background Process 1216

Resolving a Failed Background Process ...1216

Outbound Integration Endpoint Structure	1219
Business Rules Event Processing	1219
Processing Engine / Converter	1219
Error Reporter	1220
Post-Processor	1220
Delivery	1220
Additional Information About Event-Based OIEPs	1221
Event-Based OIEP Event Batching	1222
Event-Based OIEP Examples	1223
Event-Based Example Business Rules for Derived Events	1224
Event-Based Example Leaf Products with Inherited Data	1229
Event-Based Example Products Linked to Classification	1232
Event-Based Example Upward Inheritance	1236
Event-Based OIEP Forward, Rewind, Purge, and Republish	1239
Forward Events	1240
Rewind Events	1241
Purge Events	1241
Republish Events	1242
Event-Based OIEP Multithreading Support	1244
Multithreading Setup	1244
Additional Considerations	1245
Increasing the Batch Size	1245
Serialized and Parallelized	1245

Event-Based OIEP Queued Events	1246
Estimate Unread Events	1246
View First Event Batch	1246
Event-Based OIEP Status and Queue Status	1249
Set the Queue Status	1249

Data Exchange

Data stored in STEP can be exchanged and integrated with external systems. For example, this allows STEP to receive data from an upstream system such as an ERP, and then send data downstream to a web server.

Although this section deals with configuration of imports and exports in workbench, the formats, receiver methods, and delivery methods are the same regardless of the interface used. For information about imports and exports in Web UI, see **Using Web UI**.

In addition to exchanging data, images and documents can also be exchanged. Assets and data can be sent downstream together using STEPXML. For details, see **Assets and Content with STEPXML**. For exchanging assets independent of data, see the **Digital Asset Exchange** documentation.

Inbound Tools

There are two primary methods for importing data to STEP:

1. **Import Manager** uses a selected format to import objects, classifications, and entities on demand.
2. **Inbound Integration Endpoints** (IIEPs) monitor a data source for receipt of new messages of a specified data format to import at scheduled intervals.

Data Format

The data format defines the inbound file type that will be processed. The same formats are available for both inbound tools (see **Data Formats**).

Receiver Method

For an IIEP, the selected receiver method determines how STEP accesses the inbound message (see **IIEP Receiver Methods**). The Import Manager does not use a receiver method.

Outbound Tools

The two primary methods for exporting data from STEP are:

1. **Export Manager** is used to export fixed data sets of objects, classifications, and entities on demand or at scheduled intervals using a selected format and delivery method.
2. **Outbound Integration Endpoints** (OIEPs) are primarily used to monitor STEP and automatically export relevant data via a selected data format and delivery method. OIEPs can also be used to export fixed data sets, and can be invoked on demand or at scheduled intervals.

Data Format

The data format determines the output file type. The same formats are available for both outbound tools (see **Data Formats**).

Delivery Method

The delivery method determines how the output file arrives at its destination. The available delivery methods are determined by the outbound tool used (see **OIEP Delivery Methods** and the **Export Manager - Select Delivery Method**).

Gateway Tool

Gateway Integration Endpoints enable STEP to access external systems through business rules that make calls to fetch data or update status. The business rule uses synchronous REST calls from JavaScript.

For example, to create objects in STEP using an ID from an external ERP system, STEP can use a gateway endpoint to access and retrieve the external system ID.

Data Formats

For inbound data, the same formats are generally available using Import Manager or an IIEP, with the exceptions noted in the table below.

For outbound data, the same formats are available using Export Manager or an OIEP.

Format	Description	Inbound	Outbound
Advanced STEPXML	XML that uses a template based on the STEPXML XSD. See Advanced STEPXML Format documentation. *Although Advanced STEPXML is not available for selection in inbound tools, Advanced STEPXML files can be imported successfully using the STEPXML Format option.	Yes*	Yes
Alphabetical Index - XML	XML that used with the InDesign solution to create indexes. See Alphabetical Index - XML Format documentation.	No	Yes
Ariba CIF	CIF is a simple, comma-delimited list of catalog items and their attributes. See Ariba CIF Format documentation.	No	Yes
Ariba CIF 3.0	CIF is a simple, comma-delimited list of catalog items and their attributes. See Ariba CIF 3.0 Format documentation.	No	Yes
BMEcat	XML-based standard for electronic data transfer by electronic catalogs. See BMEcat Format documentation.	Yes	Yes
BMEcat 2005	XML-based standard for electronic data transfer by electronic catalogs. See BMEcat 2005 Format documentation.	Yes	Yes
CSV	Comma-separated values (CSV) file. See CSV Format documentation.	Yes	Yes
cXML	XML 'commerce eXtensible Markup Language' (cXML). See cXML Format documentation.	No	Yes
eCI@ss	eCI@ss® is a hierarchical classification and product description system for grouping materials, products, and services. See eCI@ss Format documentation. *This import is only available via the eCI@ss Classification Import wizard.	Yes*	No

Format	Description	Inbound	Outbound
ETIM	CSV Electro-Technical Information Model (ETIM) classification structure where each file contains a single language. See ETIM and ETIM v2 Format documentation.	Yes	No
ETIM IXF	XML Electro-Technical Information Model (ETIM) classification structure that includes multiple languages. See ETIM IXF Format documentation.	Yes	No
ETIM v2	CSV Electro-Technical Information Model (ETIM) classification structure where each file contains a single language. See ETIM and ETIM v2 Format documentation.	Yes	No
Excel	Microsoft Excel spreadsheet file. See Excel Format documentation.	Yes	Yes
Excel Custom Template	Microsoft Excel spreadsheet file. See Excel Custom Template documentation.	No	Yes
Excel Smartsheet	Smartsheet spreadsheet file. See Excel Smartsheet Format documentation.	Yes	Yes
FixedWidth	Text file with rows and columns of data, where each column has a fixed width. See FixedWidth Format documentation.	Yes	No
Flatplan Excel	Excel file for publications. See Flatplan Excel Format . *This export is only available via Export Manager.	Yes	Yes*
Generic XML	XML with a variety of accessible formats. See Generic XML Format documentation.	Yes	Yes
Global Product Classification	GPC is a five (5) level classification which is modeled as a classification hierarchy in STEP. See Global Product Classification Format documentation. *This import is only available via the Global Product Classification Import wizard.	Yes*	No
IDoc MATMAS 05	XML format used when exchanging information with SAP. See IDoc MATMAS 05 Format documentation.	Yes	Yes

Format	Description	Inbound	Outbound
Publication Excel	Excel file for publications. See Publication Excel Format . *This export is only available via Export Manager.	Yes	Yes*
SmartLabel	Generic XML-based including a pre-defined template for integrators to provide a head start in configuring a SmartLabel™ export solution for both food and non-food product data. See SmartLabel Format documentation.	No	Yes
STEPXML	XML generated based on parameters displayed in a dialog. Unlike Advanced STEPXML format, no template is required. See STEPXML Format documentation.	Yes	Yes
UNSPSC	United Nations Standard Products and Services Code® (UNSPSC) multi-level classification folders. See UNSPSC Format documentation.	Yes	No
xCBL	XML Common Business Library (xCBL) for business-to-business e-commerce. See xCBL Format documentation.	No	Yes

Advanced STEPXML Format

The Advanced STEPXML format and the STEPXML format provide access to the same functionality but each has its own interface. The STEPXML format is easier for beginners to use since its user interface includes parameters with valid options. Using the Advanced STEPXML format requires knowledge of the STEPXML tags, XML attributes, and valid settings for the schema that must be created manually in the Export Manager or OIEP template parameter (see the **Outbound Data** section below).

As defined in the tags and examples documentation, not all data that can be exported using Advanced STEPXML can be imported again. Available tags and examples using STEPXML are available in the **STEPXML Tags and Examples** documentation.

Note: For assistance in creating a valid Advanced STEPXML template, use the STEPXML user interface to output a template and then modify as needed for inbound or outbound STEPXML data. For more information, see **STEPXML Format**.

Schema

The schema is not publicly available, which means it is not possible to validate STEPXML without access to the XSD file.

The XSD file for STEPXML is available at:

```
http://[application server]/files
```

A web view of the schema is available at:

```
http://[application server]/xsd/index.html
```

The STEPXML schema is extended continuously but should always be backwards compatible. For more information on legacy naming conventions, see the **Backward Compatibility in STEPXML** documentation.

Format Availability

Advanced STEPXML is available for selection in:

- IIEP - Although Advanced STEPXML is not available for selection in inbound tools, Advanced STEPXML files can be imported successfully using the **STEPXML Format** option.
- Import Manager - Although Advanced STEPXML is not available for selection in inbound tools, Advanced STEPXML files can be imported successfully using the **STEPXML Format** option.
- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

Mapping

Since STEPXML is the native format for STEP, mapping data is not required and the Map Data step is disabled.

Inbound Data

Although Advanced STEPXML is not available for selection in inbound tools, Advanced STEPXML files can be imported successfully using the **STEPXML Format** option.

Import Manager

Import Manager

Steps

1. Select Configuration

2. Select Data Source

3. Select Format

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Select Format

Format

STEPXML

Converter for the STEP Product Information XML format.

Validate

no

Conversion Preview:

<ID>	>	<Parent ID>	>	<Type ID>	>	<Name>	>	ChildCount	>	<
21926		21924		Level3		Refrigeration		0		
21929				ItemFolder		Refrigeratio...		0		
21933				Item		21933		4		1
21931				SalesItemFol...		Refrigeratio...		0		
21934				SalesItem		321934-24		0		
110701				SalesItem		Coil		0		

<

>

Back

Next

Finish

Cancel

IIEP

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint

2. Choose Receiver

3. Configure Endpoint

4. Configure PreProcessor

5. Configure Processing Engine

5.1. Select Sample File

5.2. Select Data Source

5.3. Select Format

5.4. Map Data

5.5. Identify Objects

5.6. Identify Destination

5.7. Select Business Rules

5.8. Advanced Settings

6. Configure PostProcessor

7. Schedule Endpoint

8. Configure Error Reporter

Configure Processing Engine : Select Format

FormatSTEPXML

Converter for the STEP Product Information XML format.

Validateno

Conversion Preview:

<ID>	>	<Parent ID>	>	<Type ID>	>	<Name>	>
21926		21924		Level3		Refrigeration	
21929				ItemFolder		Refrigeratio...	
21933				Item		21933	
21931				SalesItemFol...		Refrigeratio...	
21934				SalesItem		321934-24	
110701				SalesItem		Coil	

Back

Next

Finish

Cancel

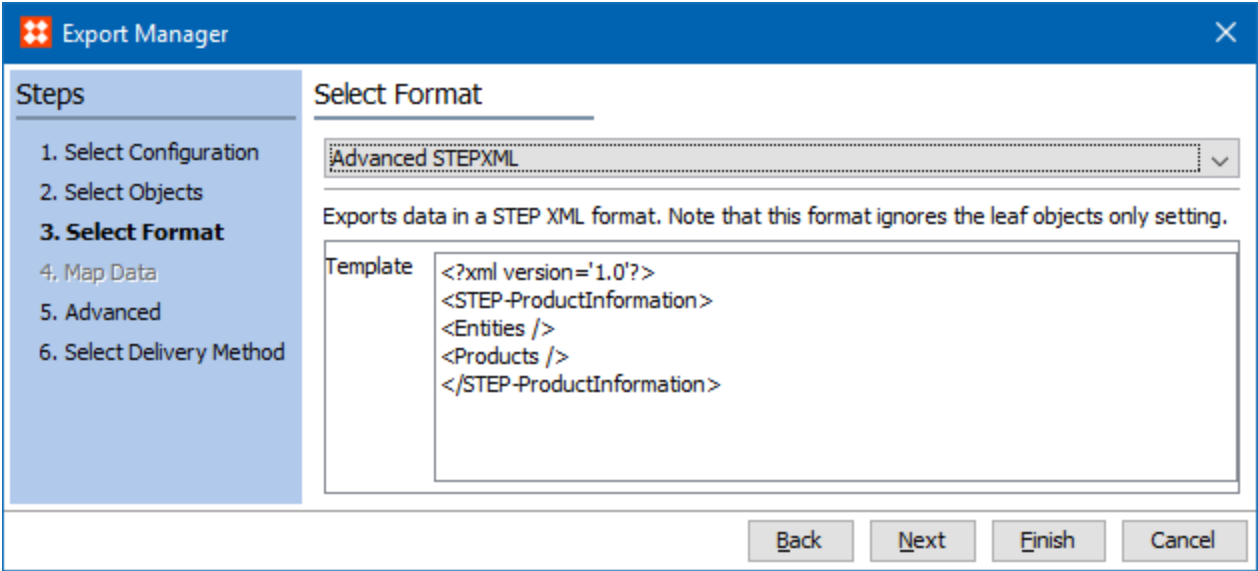
Outbound Data

The default template loaded for export will output all entities and products. Update it with your own template to configure the data, control the attribute groups and reference types, and define the layout of the export.

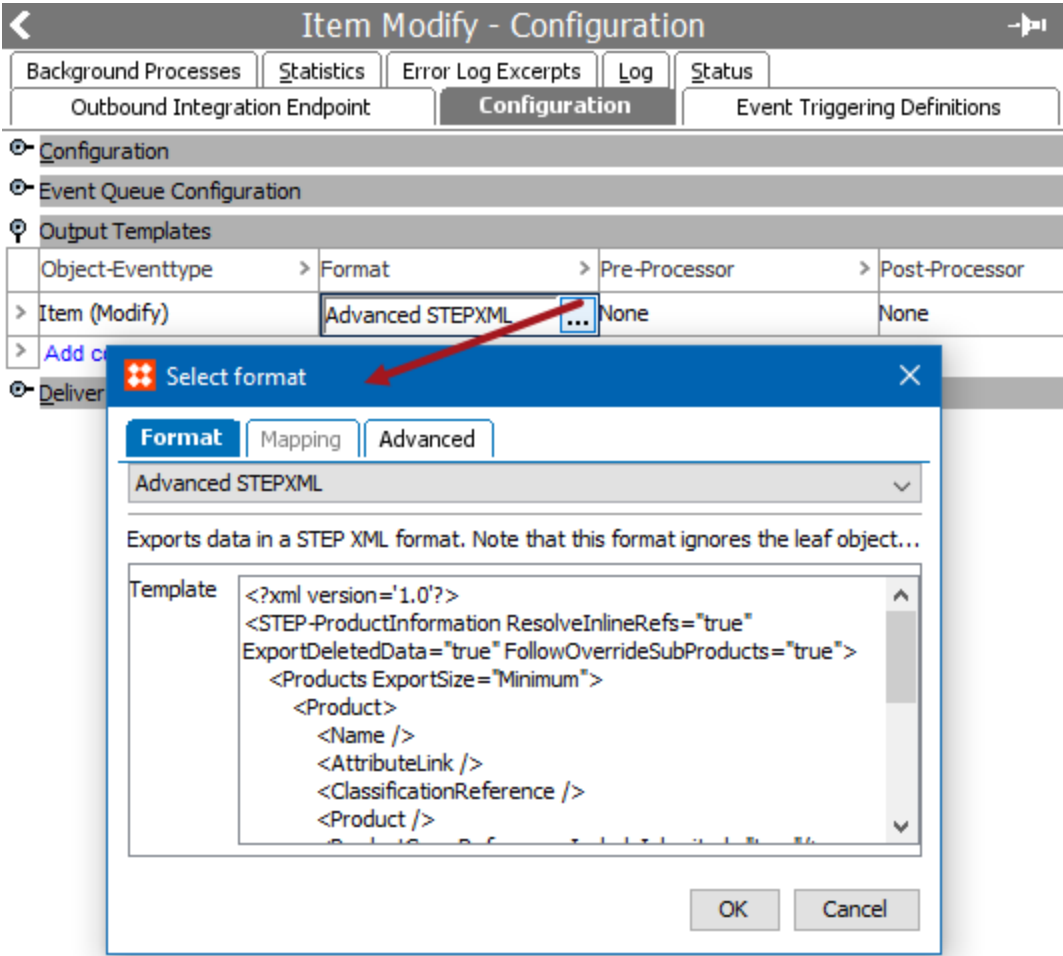
Since no undo functionality is available in the Template field, it is good practice to use an external XML editor program for creating and editing a template. To test a template's validity for output, paste the XML into STEP.

Available tags and example templates using STEPXML are available in the **STEPXML Tags and Examples** documentation. Ultimately, the template must be based on the STEPXML XSD. Click the **STEP API Documentation** button on the STEP WebStart page for a link to the XSD file.

Export Manager



OIEP



Alphabetical Index - XML Format

The Alphabetical Index - XML format exports data from an index word hierarchy for a STEP publication. 'Stock number' or 'article number' can also be exported for creating indexes from a STEP publication. For more information, see the **Creating Document Indexes** topic or the **Exporting Index Data from STEP** topic in the **STEP'n'design** documentation.

Format Availability

Alphabetical Index - XML is available for selection in:

- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

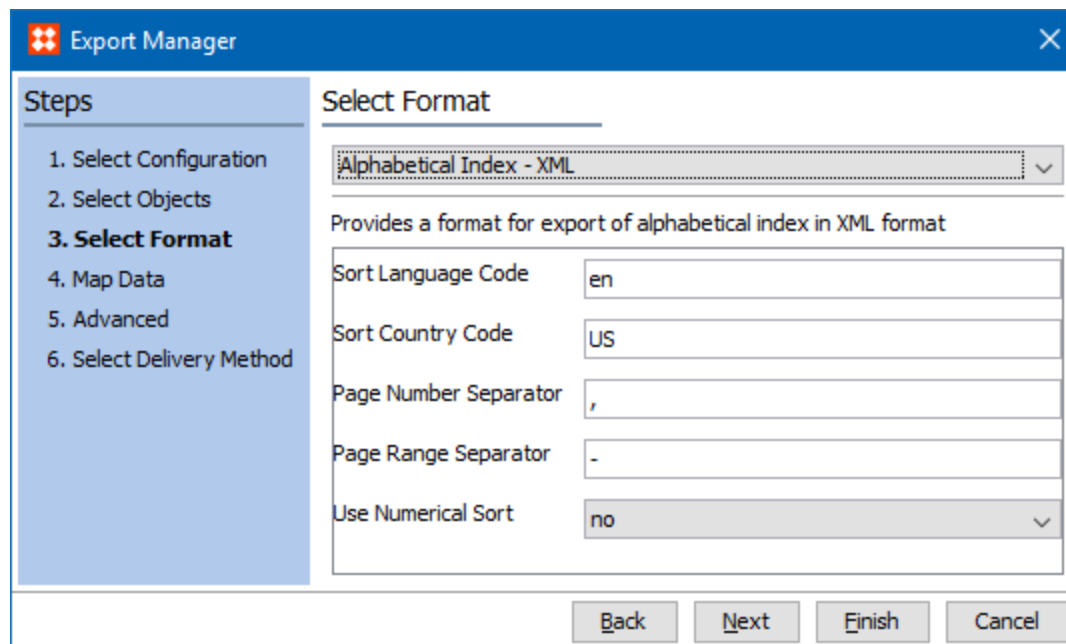
Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

Outbound Data

The same parameters are available in both Export Manager and OIEP.

Export Manager



The screenshot shows the 'Export Manager' dialog box with a blue header bar. On the left, a 'Steps' sidebar lists six steps: 1. Select Configuration, 2. Select Objects, 3. Select Format (highlighted), 4. Map Data, 5. Advanced, and 6. Select Delivery Method. The main area is titled 'Select Format' and features a dropdown menu with 'Alphabetical Index - XML' selected. Below the dropdown, a description reads: 'Provides a format for export of alphabetical index in XML format'. A form contains five fields: 'Sort Language Code' (en), 'Sort Country Code' (US), 'Page Number Separator' (,), 'Page Range Separator' (-), and 'Use Numerical Sort' (no). At the bottom, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

OIEP

Background Processes		Statistics		Error Log Excerpts		Log		Status	
Outbound Integration Endpoint				Configuration		Event Triggering Definitions			
Configuration									
Event Queue Configuration									
Output Templates									
Object-Eventtype		Format		Pre-Processor		Post-Processor			
> Item (Modify)		Alphabetical Index - XML (2 m. ...)		None		None			
> Add configuration									
Delivery Method									

Select format

Format

Mapping

Advanced

Alphabetical Index - XML

Provides a format for export of alphabetical index in XML format

Sort Language Code

en

Sort Country Code

US

Page Number Separator

,

Page Range Separator

-

Use Numerical Sort

no

OK

Cancel

Ariba CIF Format

The Ariba CIF format is a simple comma-delimited list of catalog items and their attributes. For more information, search the web.

Format Availability

Ariba CIF is available for selection in:

- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

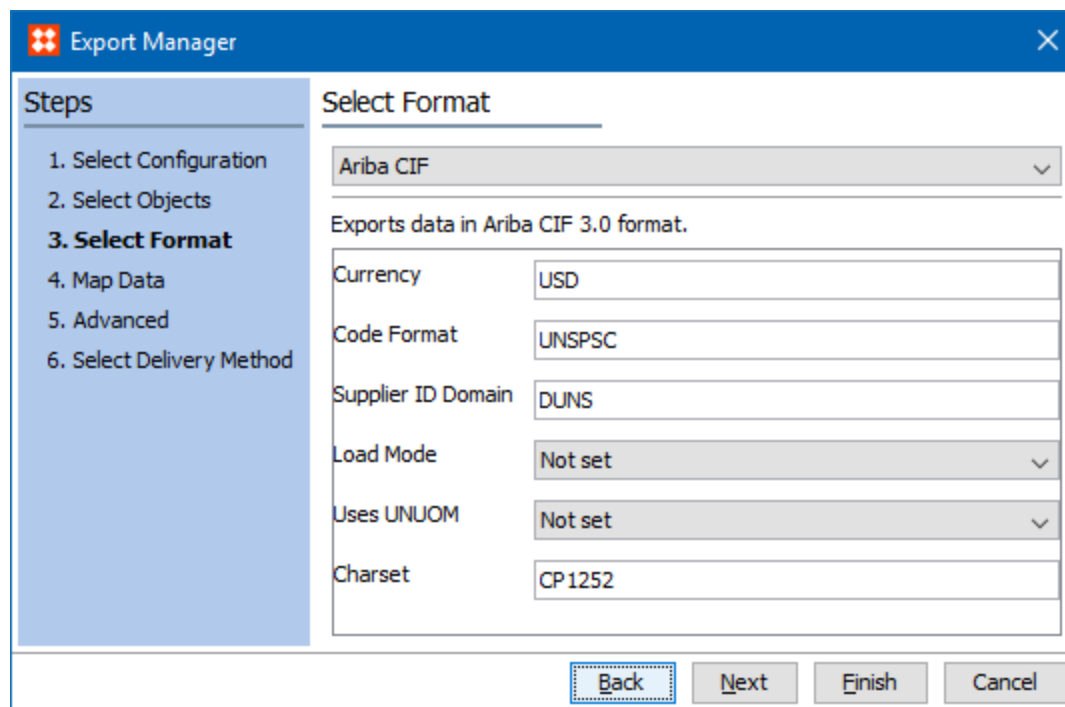
Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

Outbound Data

The same parameters are available in both Export Manager and OIEP.

Export Manager



The screenshot shows the 'Export Manager' dialog box with a blue header bar. On the left is a 'Steps' sidebar with a list of steps: 1. Select Configuration, 2. Select Objects, 3. Select Format (highlighted), 4. Map Data, 5. Advanced, and 6. Select Delivery Method. The main area is titled 'Select Format' and features a dropdown menu currently set to 'Ariba CIF'. Below this, a text label states 'Exports data in Ariba CIF 3.0 format.' followed by a series of input fields: 'Currency' (USD), 'Code Format' (UNSPSC), 'Supplier ID Domain' (DUNS), 'Load Mode' (Not set), 'Uses UNUOM' (Not set), and 'Charset' (CP1252). At the bottom right, there are four buttons: 'Back' (highlighted with a dashed border), 'Next', 'Finish', and 'Cancel'.

OIEP

Background Processes

Statistics

Error Log Excerpts

Log

Status

Outbound Integration Endpoint

Configuration

Event Triggering Definitions

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor	Post-Processor
Item (Modify)	Ariba CIF (4 mappings)	None	None
Add configuration			

Delivery Method

Select format

Format

Mapping

Advanced

Ariba CIF

Exports data in Ariba CIF 3.0 format.

Currency

USD

Code Format

UNSPSC

Supplier ID Domain

DUNS

Load Mode

Incremental

Uses UNUOM

Yes

Charset

CP1252

OK

Cancel

Ariba CIF 3.0 Format

The Ariba CIF 3.0 format is a simple comma-delimited list of catalog items and their attributes. For more information, search the web.

Format Availability

Ariba CIF 3.0 is available for selection in:

- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

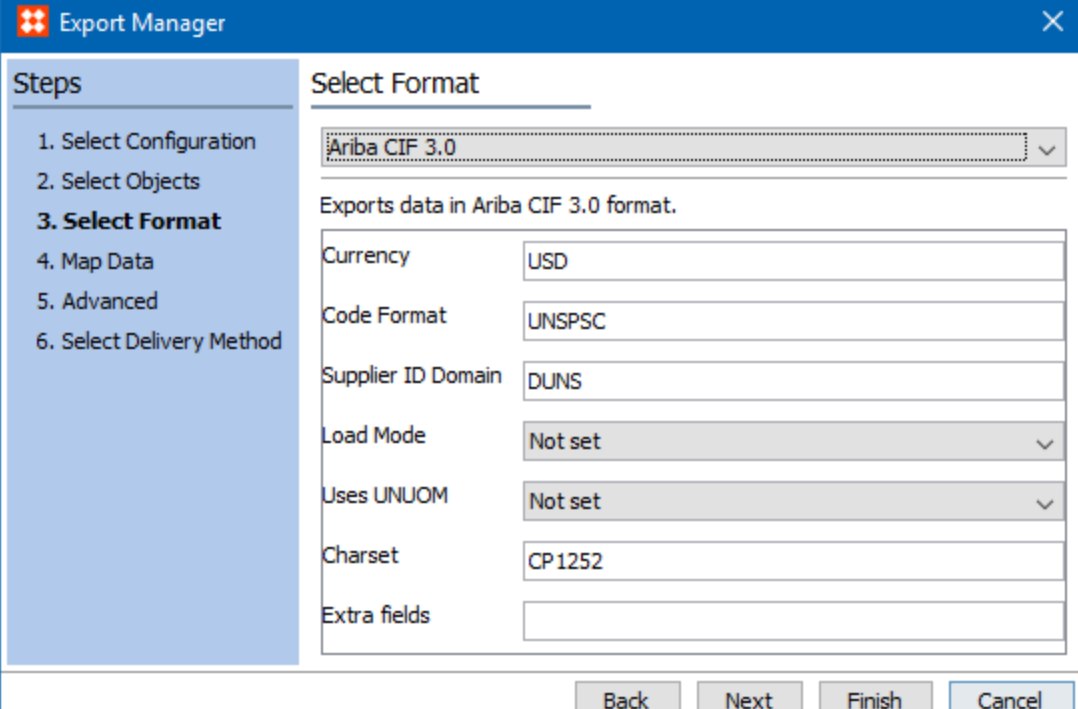
Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

Outbound Data

The same parameters are available in both Export Manager and OIEP.

Export Manager



The screenshot shows the 'Export Manager' dialog box with the 'Steps' pane on the left and the 'Select Format' pane on the right. The 'Steps' pane lists six steps: 1. Select Configuration, 2. Select Objects, 3. Select Format (highlighted), 4. Map Data, 5. Advanced, and 6. Select Delivery Method. The 'Select Format' pane shows a dropdown menu with 'Ariba CIF 3.0' selected. Below the dropdown, it says 'Exports data in Ariba CIF 3.0 format.' and lists several configuration options: Currency (USD), Code Format (UNSPSC), Supplier ID Domain (DUNS), Load Mode (Not set), Uses UNUOM (Not set), Charset (CP1252), and Extra fields (empty). At the bottom of the dialog are four buttons: Back, Next, Finish, and Cancel.

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

Parameter	Value
Currency	USD
Code Format	UNSPSC
Supplier ID Domain	DUNS
Load Mode	Not set
Uses UNUOM	Not set
Charset	CP1252
Extra fields	

OIEP

Background Processes

Statistics

Error Log Excerpts

Log

Status

Outbound Integration Endpoint

Configuration

Event Triggering Definitions

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor	Post-Processor
> Item (Modify)	Ariba CIF 3.0 (5 mappings)...	None	None
> Add configuration			

Select format

Format

Mapping

Advanced

Ariba CIF 3.0

Exports data in Ariba CIF 3.0 format.

Currency

USD

Code Format

UNSPSC

Supplier ID Domain

DUNS

Load Mode

Not set

Uses UNUOM

Not set

Charset

CP1252

Extra fields

OK

Cancel

BMEcat Format

Both BMEcat (version 1.2) and BMEcat 2005 are supported in STEP and use an XML-based standard for electronic data transfer by electronic catalogs. For more information about the standard, search the web.

For more information on BMEcat 2005, see the **BMEcat 2005 Format** documentation.

Format Availability

BMEcat is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**
- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

BMEcat 1.2 and BMEcat 2005 are the only formats that allow access to Product Attribute Link metadata attributes, via the transformation aspect 'Closest Attribute Link Meta Data.' For details, see **Aspect - Transform Outbound**.

If mapping an ETIM file to the BMEcat2005 format, see the **LOV Value-ID or Value Aspect** topic.

Inbound Data

Inbound Parameters

- **Validate** determines if STEP checks the validity of the BMEcat file against a BMEcat DTD file before loading any data.
- **Conversion Preview** displays data being imported when the BMEcat DTD file is in the system., otherwise the field is empty.

Import Manager

Import Manager

×

Steps

1. Select Configuration

2. Select Data Source

3. Select Format

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Select Format

Format

BMEcat

Converter for the BMEcat 1.2 format (see www.bmecat.org).

Validate

yes

Conversion Preview:

SUPPLIER_AID	>	DESCRIPTION_SHORT	>	ORDER_UNIT	>	PRICE_AMOU...	>	^
Products Galore, Inc.		Baseball style cap, adjustable...		20		3.33		v
< >								

Back

Next

Finish

Cancel

IIEP

Inbound Integration Endpoint Wizard

×

Steps

1. Identify Endpoint

2. Choose Receiver

3. Configure Endpoint

4. Configure PreProcessor

5. Configure Processing Engine

5.1. Select Sample File

5.2. Select Data Source

5.3. Select Format

5.4. Map Data

5.5. Identify Objects

5.6. Identify Destination

5.7. Select Business Rules

5.8. Advanced Settings

6. Configure PostProcessor

7. Schedule Endpoint

8. Configure Error Reporter

Configure Processing Engine : Select Format

Format

BMEcat

Converter for the BMEcat 1.2 format (see www.bmecat.org).

Validate

no

Conversion Preview:

SUPPLIER_AID	>	DESCRIPTION_SHORT	>	ORDER_UNIT	>	PRIC...	>	^
Products Galore, Inc.		Baseball style cap, adjustabl...		20		3.33		v
< >								

Back

Next

Finish

Cancel

Outbound Data

Exporting the BMEcat format involves supplying header data and catalog / product data via the parameters displayed on the Select Format step and during the Map Data step. Some information is required, as is indicated on the Map Data step. Although the export process can be run while mandatory fields are missing, the background process (BGP) will fail.

Outbound Parameters

1. For the **Export Validate** parameter, select an option from the dropdown to determine how mandatory fields (marked in red on the Map Data step) are handled during export:
 - **Warn and skip missing mandatory data** - When a value is missing, the product is not exported and the execution report includes the warning 'Skipping product ([ID] / [Name]) missing mandatory data: xxxx.'
 - **Warn and allow missing mandatory data** - When a value is missing, the product is exported minus the missing mandatory data, and the execution report includes the warning 'Found product ([ID] / [Name]) missing mandatory data: xxxx.'
 - **Abort on missing mandatory data** - When a value is missing, the export fails and the execution report includes 'Aborting export due to product ([ID] / [Name]) missing mandatory data: xxxx' and the time of the failure.
4. For the **Validate XML** parameter, **Yes** indicates that the BMEcat 1.2 export file is validated against the BMEcat DTD included in STEP. When validation fails, the BGP also fails, and the problem is reported in the BGP execution report. If set to **No**, validation is skipped, and the BGP does not fail due to differences found compared to the DTD.
5. For the **Price update only?** parameter, **No** indicates that the catalog section of the file is included. If set to **No**, the catalog section is excluded and only fields related to prices are exported.
6. For the **Update count** parameter, allows you to track the version number displayed in the tag with the 'prev_version' XML attribute within the file. This parameter must be manually incremented as needed.
7. For an explanation of the remaining parameters, search the web to find and refer to the 'Specification BMEcat® Version 1.2' document. No validation is performed on the text entered or the selections made, but if Validate XML = Yes, errors will be reported in the execution report, as defined above.

Export Manager

Export Manager

×

Steps

1. Select Configuration

2. Select Objects

3. **Select Format**

4. Map Data

5. Advanced

6. Select Delivery Method

Select Format

BMEcat

▼

Converts data to a BMEcat 1.2 format (see www.bmecat.org).

Export Validation

Warn and skip missing mandatory data

▼

Validate XML

Yes

▼

Price update only?

No

▼

Update count

0

Generator Information

Generated by STEP BMEcat2 plugin.

Language

Catalog ID

Catalog Version

Back

Next

Finish

Cancel

OIEP

Website - Configuration

Outbound Integration Endpoint

Configuration

Event Triggering Definitions

Background Processes

Status

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor
> Level 1 (Create)	BMEcat (2 mappings)	None

Select format

Format

Mapping

Advanced

BMEcat

Converts data to a BMEcat 1.2 format (see www.bmecat.org).

Export Validation

Warn and skip missing mandatory data

Validate XML

No

Price update only?

No

Update count

0

Generator Information

Generated by STEP BMEcat2 plugin.

Language

eng

Catalog ID

Catalog Version

OK

Cancel

BMEcat 2005 Format

Both BMEcat® (version 1.2) and BMEcat® 2005 are supported in STEP and use an XML-based standard for electronic data transfer by electronic catalogs. For more information about the standard, search the web.

For more information on BMEcat (version 1.2), see the **BMEcat Format** documentation.

Format Availability

BMEcat 2005 is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**
- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

BMEcat 2005 and BMEcat 1.2 are the only formats that allow access to Product Attribute Link metadata attributes, via the transformation aspect 'Closest Attribute Link Meta Data.' For details, see **Aspect - Transform Outbound**.

Inbound Data

Inbound Parameters

The following parameters are displayed on the inbound wizards:

- **Sample** is updated automatically when the format is selected.
- **Refresh** button updates the Conversion Preview pane.
- **Conversion Preview** displays data being imported.

Import Manager

Import Manager

Steps

1. Select Configuration

2. Select Data Source

3. Select Format

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Select Format

Format

BMEcat 2005

Converter for an BMEcat 2005 format described by a template

Sample

<BMECAT version="2005">
<T_NEW_CATALOG>
 <PRODUCT>
 <?Record?>

Conversion Preview:

Supplier P... >	Short Des... >	EAN code >	Manufact... >	Manufact... >	M
1000.06.025	Kabelverschr...	7611614104...	1000.06.025	Kaiser	10
1000.06.030	Kabelverschr...	7611614104...	1000.06.030	Kaiser	10
1000.06.035	Kabelverschr...	7611614104...	1000.06.035	Kaiser	10

Back

Next

Finish

Cancel

IIEP

The screenshot shows the 'Inbound Integration Endpoint Wizard' window, specifically the 'Configure Processing Engine : Select Format' step. The left sidebar lists the steps, with '5.3. Select Format' highlighted. The main area shows the 'Format' dropdown set to 'BMEcat 2005'. Below it, a 'Sample' XML template is displayed, and a 'Conversion Preview' table shows the resulting data structure.

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
- 5. Configure Processing Engine**
 - 5.1. Select Sample File
 - 5.2. Select Data Source
 - 5.3. Select Format**
 - 5.4. Map Data
 - 5.5. Identify Objects
 - 5.6. Identify Destination
 - 5.7. Select Business Rules
 - 5.8. Advanced Settings
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Processing Engine : Select Format

Format: **BMEcat 2005**

Converter for an BMEcat 2005 format described by a template

Sample

```
<BMECAT version="2005">
  <T_NEW_CATALOG>
    <PRODUCT>
      <?Record?>

      <SUPPLIER_PID><?Source Supplier Product ID?></SUPPLIER_PID>
      <PRODUCT_DETAILS>
        <DESCRIPTION_SHORT><?Source Short Description?></DESCRIPTION_SHORT>
      </PRODUCT_DETAILS>
    </PRODUCT>
  </T_NEW_CATALOG>
</BMECAT>
```

Conversion Preview:

Supplier P...	Short Des...	EAN code	Manufact...	Manufact...	Manufact...
1000.06.025	Kabelverschr...	7611614104...	1000.06.025	Kaiser	1000.06.025
1000.06.030	Kabelverschr...	7611614104...	1000.06.030	Kaiser	1000.06.030
1000.06.035	Kabelverschr...	7611614104...	1000.06.035	Kaiser	1000.06.035

Buttons: Back, Next, Finish, Cancel

Outbound Data

Exporting the BMEcat format involves supplying header data and catalog / product data via the parameters displayed on the Select Format step and during the Map Data step. Some information is required, as is indicated on the Map Data step. Although the export process can be run while mandatory fields are missing, the background process (BGP) will fail.

Outbound Parameters

1. In the **Sample** parameter, an XML template file is loaded automatically but can be modified as required. This defines the format of the XML file to be exported.
2. For the **Allow empty tags** parameter, **Yes** indicates that export tags with empty values are included in the output. If set to **No**, tags containing empty values are not included in the export.

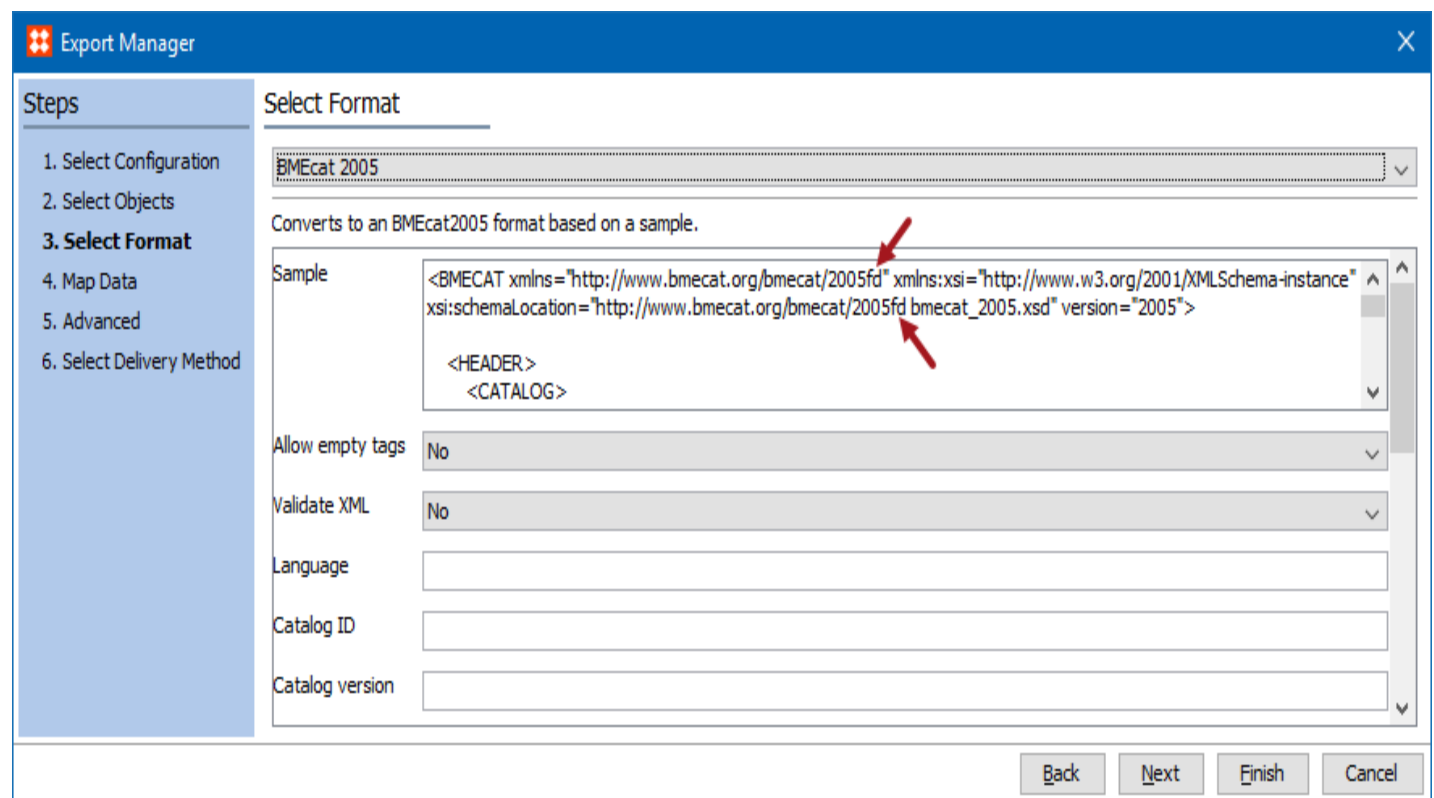
For example, setting this parameter to Yes enables all ETIM attributes, even those without a value in STEP, to be exported. Additionally, when ETIM values exported via BMEcat are empty, they should be represented by a dash (-). For information on transforming an empty attribute value to a text string, see the Outbound Add Transformation Option **Replace the whole value** section in the **Outbound Map Data - Transform** topic of the **Mapping** documentation.

- For the **Validate XML** parameter, **Yes** indicates that the BMEcat 2005 export file is validated against an XSD included in STEP. When validation fails, the background process also fails, and the problem is reported in the BGP execution report. If set to **No**, the validation is skipped and the BGP does not fail due to differences found when comparing to the XSD.

Note: To successfully validate against the XSD, the sample file must include 'fd' option (as shown in two places in the images below).

- For an explanation of the remaining parameters, search the web to find and refer to the 'Specification BMEcat® 2005' document. No validation is performed on the text entered or the selections made, but if Validate XML = Yes, errors will be reported in the execution report, as defined above.

Export Manager



Export Manager

Steps

- Select Configuration
- Select Objects
- Select Format**
- Map Data
- Advanced
- Select Delivery Method

Select Format

BMEcat 2005

Converts to an BMEcat2005 format based on a sample.

Sample

```
<BMECAT xmlns="http://www.bmecat.org/bmecat/2005fd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.bmecat.org/bmecat/2005fd bmecat_2005.xsd" version="2005">
  <HEADER>
  <CATALOG>
```

Allow empty tags: No

Validate XML: No

Language:

Catalog ID:

Catalog version:

Back Next Finish Cancel

OIEP

Website - Configuration

Outbound Integration Endpoint **Configuration** Event Triggering Definitions Background Processes Statistics Error Log Excerpts Log Status

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor	Post-Processor
> Level 3 (Create, Modify, Delete)	BMEcat 2005 (2 mappings) ...	None	None

Select format

Format Mapping Advanced

BMEcat 2005

Converts to an BMEcat2005 format based on a sample.

Sample

```
<BMECAT xmlns="http://www.bmecat.org/bmecat/2005fd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.bmecat.org/bmecat/2005fd bmecat_2005.xsd" version="2005">

  <HEADER>
    <CATALOG>
```

Allow empty tags No

Validate XML No

Language

Catalog ID

Catalog version

OK Cancel

CSV Format

The comma-separated values (CSV) file format is available for inbound and outbound data exchanges. Using CSV requires creating a data map between STEP and the data being processed, and may also include data transformations.

The following sample CSV data import file shows that the first row is a header, and the remaining data is delimited by a comma. Exported CSV files always include a header row, though this is optional for imports.

```
<Name>,<Parent ID>,Primary Color,Secondary Color
Mens T PBO,18209,Black,Orange
Mens T PBG,18209,Blue,Green
Mens T PGS,18209,Green,Silver
Mens T PGW,18209,Gray,white
Mens T POY,18209,Orange,Yellow
```

Keep the following points in mind when working with STEP data using CSV format:

- The following node types / super types can be imported and exported via CSV format: products, classifications, entities, assets (objects, not content), and attributes (objects and definitions).
- A CSV import or export file will include data in the same arrangement as relational database tables. This means that each object is displayed as a single row in the file and each object property item is displayed as a single column.
- Imports and exports are context and workspace specific. By default, data is imported to or extracted from the context and workspace in use when the process is started.
- When planning to import data back into STEP, include STEP ID in the export.

Format Availability

CSV is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**
- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

Inbound Data

CSV import allows creation of and updates to products, classifications, entities, attribute values and references. However, system setup objects (for example, attributes, LOVs, users, reference types, and so on), cannot be created or modified via import.

Because the Map Data process allows selection of only a single node type, only one node type / super type (products, entities, etc.) can be imported at a time. When multiple super types exist in the same import file, a separate import is required to successfully import each type of object, starting with classification data, then product data, and finally, entity data. When the inbound file includes data for node types other than the one selected, two things may happen: 1) assuming none of the data prevents the import, new objects are created using the supplied information and the selected super type, 2) the execution report details the skipped records when included data, like parent ID, is not found in the selected super type hierarchy. Alternately, split the inbound data file by super type and process accordingly.

For information on parallel imports involving multiple references on object types, see **Reference Target Lock Policy on Object Types**.

Inbound Parameters

The following parameters are available in both Import Manager and IIEP.

1. **Delimiter** - determines the character used to delimit the exported data. Options are semi-colon (;), comma (,), the tab character, the pipe symbol (|), or colon (:).
 - **Tab** is recommended to avoid splitting the data on printable characters that may occur within the values.
 - Only the delimiters in the dropdown are allowed. If the file uses a different delimiter, before importing, first update the file outside of STEP to use a delimiter allowed by STEP.
2. **Character Set** - determines the characters that can be successfully imported. Options include Windows-1252, ISO-8859-1 (also known as the Latin-1 character set), UTF-8, or UTF-16.
 - **UTF-8** is recommended unless you have a reason to do otherwise.
 - **UTF-8** or **UTF-16** (a Unicode character set) is required to correctly import special symbols, like a trademark symbol.
 - **Shift-JIS** character set used for the Japanese language.
3. **Has Header** - indicates if the file has a header row.
 - **Yes** indicates that the first line (row) of the CSV file has header information, such as attribute names that match STEP, the Auto Map feature is available to map the columns of data to the appropriate object in STEP.
 - **No** indicates that the first line (row) of the CSV file has actual data and no labels are included to identify data. Although a header row is not required, without it the user must be able to identify the data for manual mapping.
4. **Trim whitespace** - determines how blank spaces before or after the value are handled.
 - **Yes** removes leading and trailing spaces in imported values.
 - **No** leaves leading and trailing spaces in imported values.
5. **Allow Multi Line Values** - determines how new lines (return codes) are handled in the file, indicated by data with double quoted values that split over several lines. Typically, the newline character is interpreted as the end of data, delimiting a data record. However, sometimes data spans more than one line—that is, includes newline characters. In such cases, values must be quoted with the double quote character (") to be imported correctly.

- **Yes** values with new lines are quoted with the double quote character (").
 - **No** new line characters are interpreted as the end of data, delimiting a data record.
6. **Conversion Preview** - displays a sample of the first few lines of the file to allow verification that the selected options are correct.

Import Manager


Import Manager
✕

Steps

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

Select Format

Format

CSV

Converter for character separated formats where each line contains data about one object in fields separated by a delimiter character

Delimter

/

Character Set

windows-1252

Has Header

yes

Trim whitespace

no

Allow Multi Line Values

no

Conversion Preview:

<Name>	>	<Parent ID>	>	Primary Color	>	Secondary Color	>
Mens T PBO		18209		Black		Orange	
Mens T PBG		18209		Blue		Green	
Mens T PGS		18209		Green		Silver	
Mens T PGW		18209		Gray		White	
Mens T POY		18209		Orange		Yellow	

Back

Next

Finish

Cancel

IIEP

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
- 5. Configure Processing Engine**
 - 5.1. Select Sample File
 - 5.2. Select Data Source
 - 5.3. Select Format**
 - 5.4. Map Data
 - 5.5. Identify Objects
 - 5.6. Identify Destination
 - 5.7. Select Business Rules
 - 5.8. Advanced Settings
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Processing Engine : Select Format

Format: **CSV**

Converter for character separated formats where each line contains data about one object in fields separated by a delimit...

Delimiter: **,**

Character Set: **windows-1252**

Has Header: **yes**

Trim whitespace: **no**

Allow Multi Line Values: **no**

Conversion Preview:

<Name>	<Parent ID>	Primary Color	Secondary Color
Mens T PBO	18209	Black	Orange
Mens T PBG	18209	Blue	Green
Mens T PGS	18209	Green	Silver
Mens T PGW	18209	Gray	White
Mens T POY	18209	Orange	Yellow

Back Next Finish Cancel

Outbound Data

When data leaves STEP via CSV format, several options are available to customize the output.

Note: When using the JDBC delivery method users must select 'CSV' as the format. For more information on how to configure the CSV format for the JDBC delivery method, see the **Exporting Data via JDBC with CSV Format** topic in this guide.

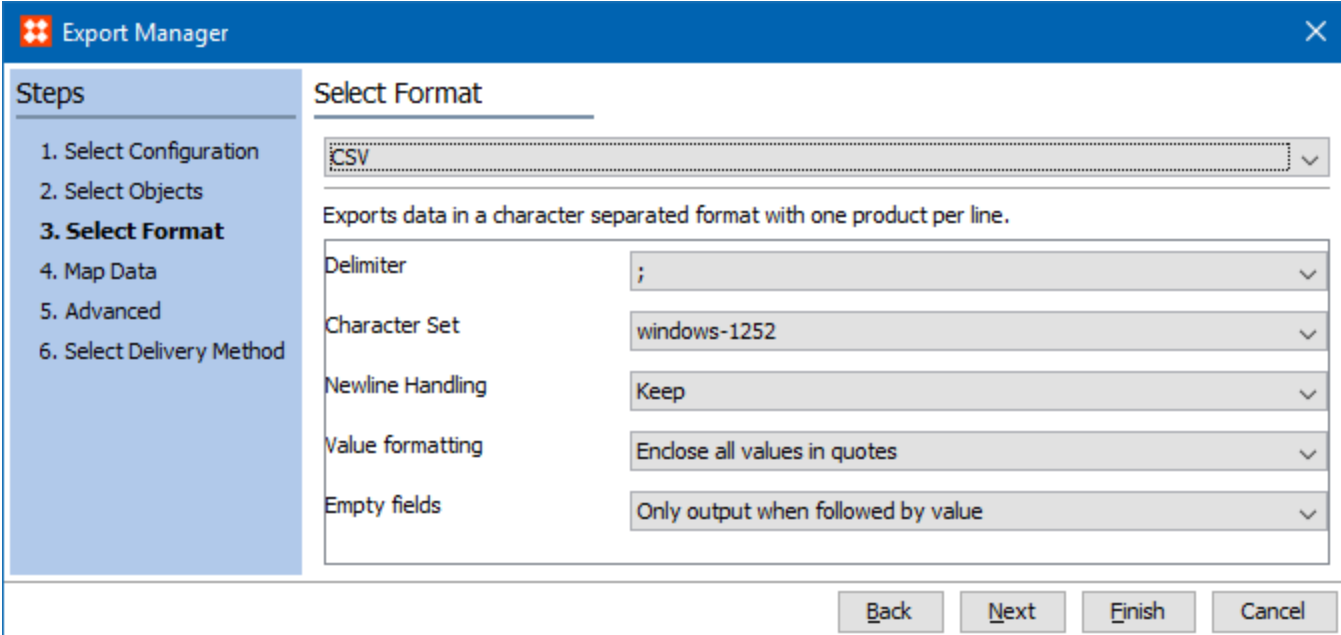
Outbound Parameters

The following parameters are available in both Export Manager and OIEP.

1. **Delimiter** - determines the character used to delimit the exported data. Options are semi-colon (;), comma (,), the tab character, the pipe symbol (|), or colon (:).
 - **Tab** is recommended to avoid splitting the data on printable characters that may occur within the values.
 - Only the delimiters in the dropdown are allowed. If a different delimiter is required, after exporting, update the file outside of STEP.

2. **Character Set** - determines the characters that can be exported. Options include Windows-1252, ISO-8859-1 (also known as the Latin-1 character set), UTF-8, or UTF-16.
 - **UTF-8** is recommended unless you have a reason to do otherwise.
 - **UTF-8** or **UTF-16** (a Unicode character set) is required to correctly export special symbols, like a trademark symbol.
 - **Shift-JIS** character set used for the Japanese language.
3. **Newline Handling** - determines how new lines (return codes) are handled when they occur in the data being exported.
 - **Convert to Space** (recommended) when new lines exist in the data being exported, to prevent data integrity issues caused by character-separated files having new lines within a record.
 - **Keep** allows new lines to be included and can be used when the file recipient can handle them within a record.
4. **Value formatting** - determines how values are displayed in the exported file.
 - **Enclose all values in quotes** takes STEP values and adds quotes during export.
 - **Export without quotes unless value contains delimiter character(s)** exports STEP values without modification unless a delimiter is found.
5. **Empty fields** - determines how fields without values are exported.
 - **Only output when followed by value** exports only fields that include a value.
 - **Always output** exports all fields, regardless of the presence of a value, and is required when splitting multivalued fields.

Export Manager



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

Select Format

CSV

Exports data in a character separated format with one product per line.

Delimiter	;
Character Set	windows-1252
Newline Handling	Keep
Value formatting	Enclose all values in quotes
Empty fields	Only output when followed by value

Back

Next

Finish

Cancel

OIEP

System Setup

Inbound Integration Endpoints

Match Codes and Matching Algorithms

Outbound Integration Endpoints

- 1WS 7.0 Provider Sender
- Asset Caching
- Change Attribute Link
- CSV_OIEP
- GREExport
- Item Modify
- Item Modify Non STEPXML
- MultiValue MultiContext
- Static Data Set
- Website

- Web UIs
- Workflow Profiles
- Workflows
- Derived Events
- Object Types & Structures
- Tags
- Units
- Users & Groups
- Reference Types
- Workspaces
- Table
- Keys
- Event Queues

CSV_OIEP - Configuration

Outbound Integration Endpoint

Configuration

Event Triggering Definitions

Background Processes

Statistics

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor	Post-Processor
> Item (Modify)	CSV (3 mappings)	None	None
> Add configuration			

Delivery Methods

Select format

Format

Mapping

Advanced

CSV

Exports data in a character separated format with one product per line.

Delimiter

;

Character Set

windows-1252

Newline Handling

Keep

Value formatting

Enclose all values in quotes

Empty fields

Only output when followed by value

OK

Cancel

© Stibo Systems - Confidential - STEP Trailblazer 8.3

51

Exporting Data via JDBC with CSV Format

Exporting data with the JDBC delivery method requires the use of the CSV format. Setup for this involves the following steps in the Export Manager or in an OIEP:

- **Add required drivers** for JDBC
- **Configure properties** in the sharedconfig.properties file
- **Select format** of CSV, using the required settings
- **Map Data**, making sure to include the required action field
- **Select Delivery Method** of JDBC, using the necessary settings

Each of these steps is described below.

For details on a use case in which JDBC is used to update analytics data, see the **Analytics using JDBC Example** topic.

Install Required Drivers

JDBC specification 4.1-compliant drivers should be placed in a directory accessible from all application servers. These drivers can then be made available for the delivery plugin via the dynamic properties JDBCDeliveryPlugin.DriverPath.[n] and JDBCDeliveryPlugin.DriverClass.[n]. For more information regarding applicable Java drivers, review the RDBMS vendor's homepage on the web.

Configure Properties

The options available in the following dropdown parameters on the JDBC delivery method are supplied by the sharedconfig.properties file on the STEP application server. Multiple entries can be added using the dynamic properties. The first configuration entry must use number 1, and each additional entry must be numbered sequentially (JDBCDeliveryPlugin.DriverPath.1, JDBCDeliveryPlugin.DriverPath.2, JDBCDeliveryPlugin.DriverPath.3, etc.).

Driver Location	L:/jdbc-drivers/mysql-connector-java-5.0.8-bin.jar
Driver Class	com.mysql.jdbc.Driver
Database URL	jdbc:mysql://localhost:3306/mydb

Use the following steps to supply data for the parameters displayed above:

1. Prior to configuration, clicking the **Driver Location** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JDBCDeliveryPlugin.DriverPath.[n]** property. As an example:

```
JDBCDeliveryPlugin.DriverPath.1 = L:/shared/mysql-connector-java-5.1.42-bin.jar
```

In this example, the drivers are stored on the application server's L:/shared drive.

2. Prior to configuration, clicking the **Driver Class** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JDBCDeliveryPlugin.DriverClass.[n]** property. As an example:

```
JDBCDeliveryPlugin.DriverClass.1 = com.mysql.jdbc.Driver
```

In this example, the drivers class used is 'com.mysql.jdbc.Driver.'

3. Prior to configuration, clicking the **Database URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JDBCDeliveryPlugin.URL.[n]** property.

```
JDBCDeliveryPlugin.URL.1 = jdbc:mysql://localhost:3306/mydb
```

In this example, the database URL used is 'jdbc:mysql://localhost:3306/mydb.'

Select Format

When exporting data using the JDBC delivery option it is important that, in the 'Select Format' step, users select 'CSV'. JDBC does not support any other formats in STEP. In order to enable proper functioning of the JDBC delivery option, the configuration of the 'CSV' format option should look like this:

Select Format

CSV

Exports data in a character separated format with one product per line.

Delimiter

;

Character Set

UTF-8

Newline Handling

Convert to space

Value formatting

Enclose all values in quotes

Empty fields

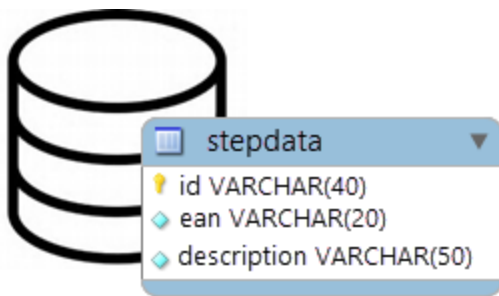
Always output

Map Data

The JDBC delivery option uses export mapping to allow STEP data to be inserted or updated ("upsert") into, or deleted from ("delete") a destination database table. When a configured object in the exported CSV file finds a match in the destination database, the interaction can be configured to either / both:

- 'upsert' the object, which means the object, if found based on the JDBC Key Columns, will be updated with the information in the exported CSV file, or if not found based on the JDBC Key Columns, the information in the exported CSV file will be inserted into the table in the destination database
- 'delete' the object, which means that if an object indicated in the exported CSV file is found in the destination database based on the JDBC Delete Key Columns, it will be deleted

For example, consider the following external database sample table:



In the 'Map Data' step, map the required objects considering the following points to ensure the CSV file can be successfully processed:

- The column headers in the exported CSV file must match the table column headers in the destination database. Map a constant value on the Header row for each mapped object to supply the external database table column header. For example, notice that the external table has a row named 'description' but is mapped to the STEP attribute 'Consumer Short Description.' The header parameter is used to make the exported data match the external table. For details about using the constant value data source, see the **Constant Value - Data Source Outbound** topic.
- Create an additional mapped column with the header of 'action' and the appropriate value of either delete or upsert. Use the transformation button to change the text displayed for both the Header and the Value.

Below is an example of an 'upsert' action in Map Data for the external database table shown above. This setup will direct the process to look for three configured attributes (ID, EAN, and Consumer Short Description) in the table in the destination database, and either insert the value (if not found), or update the value (if found):

Map Data

- <ID>
- <Name>
- <Parent ID>
- <Object Type Name>
- <Product-Override Child ID>
- <Is deleted>
- Constant Value"
- <Page Number>
- + All Attributes
- Select Attribute
- + Classifications
- + Index Words
- + Product Classification Links
- + Product References
- + Asset References
- + Classification References
- + Entity References
- + STEP Workflow Task Info
- Multi level References
- Multi level Parent attributes
- Insert Referenced Objects
- + Custom Attributes
- + System Setup

Exports data in a character separated format with one product per line.

Column (4 mapped)

upsert

Header "action"

✕

Value "upsert"

✕

<ID> ID

Header "id"

✕

Value <ID> ID

✕

EAN Value and unit

Header "ean"

✕

Value EAN Value and unit

✕

Consumer Short Description Value and unit

Header "description"

✕

Value Consumer Short Description Value and unit

✕

Important: For the external destination database, the JDBC delivery method assumes that the destination data table has a column used as a key, and a number of columns for values (for example, `key value1 value2 [value (n)]`.) This means a destination data table having only a single column (e.g., ID) will cause the JDBC export to fail, however, a table having at least two (2) columns (e.g., ID and Name) will work properly.

To create mapping for the delete action, the action column would have a value of "delete."

If using an OIEP, one output template would be created to handle Create and Modify events, and a second output template would handle Delete events. In this way, separate mapping is available for each, allowing for one to include the upsert action and the other to include the delete action.

Select Delivery Method

When setting the delivery method for the export, use the following parameters to configure it.

Select Delivery Method

JDBC

Delivers exported data in CSV format into a database table via JDBC.

Driver Location

L:/jdbc-drivers/mysql-connector-java-5.0.8-bin.jar

Driver Class

com.mysql.jdbc.Driver

Database URL

jdbc:mysql://localhost:3306/mydb

Username

user1

Password

••••••••

Table Name

stepdata

Key Columns

id

Delete Key Columns

id

1. In **Select Delivery Method**, choose **JDBC** from the dropdown.
2. In **Driver Location**, select from the dropdown one of the paths to the relevant .jar file.
3. In **Driver Class**, select from the dropdown one of the pre-configured driver class.
4. In **Database URL**, select from the dropdown one of the pre-configured URLs to the destination database.
5. In **Username**, enter the username required to access the destination database.
6. In **Password**, enter the password required to access the destination database.
7. In **Table Name**, enter the name of the table in the destination database to which STEP will be publishing data
8. In **Key Columns**, list the names of the columns appearing on both the exported CSV file and the table in the destination database, separated by commas (and no spaces), into which STEP will publish data
9. In **Delete Key Columns**, list the names of the columns appearing on both the exported CSV file and the table in the destination database, separated by commas (and no spaces), from which STEP will delete data. The headers contained in this field can differ from the headers in the 'Key Columns' field, but they must also be part of the upsert key definition.

Analytics using JDBC Example

One of the prime uses cases for deploying JDBC as a method of delivering data from STEP to a database is to populate dashboards for data analytics tools like Tableau and Qlik. To aid proper setup of a STEP integration with data analytics that makes use of the JDBC method, a use case specific to data analytics integrations is described below. For more information on setting up an analytics integration in the STEP Web UI, see the **Analytics Configuration** topic located in the Web User Interfaces guide.

One common approach to configuring data analytics dashboards is to enable display of historical data. As an example, a user has configured a Tableau dashboard to display regional sales data for a specific product. For this user, seeing sales figures for that same product a year ago, or two years ago, would greatly enhance the utility of the dashboard. The way to enable display of historical data using the JDBC method of delivering data is done in the mapping step of the Export Manager and Outbound Integration Endpoint (OIEP).

To implement this users will need to:

- **Install required drivers** for JDBC
- **Configure properties** in the sharedconfig.properties file
- **Select format** of CSV, using the required settings

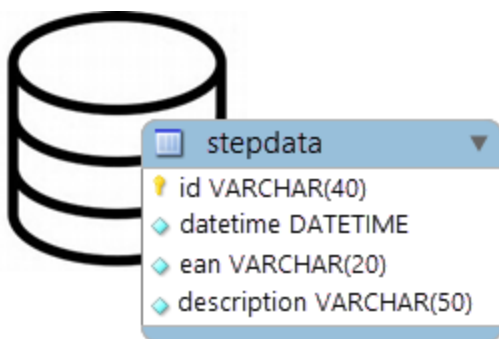
These steps are described in detail in the **Exporting Data via JDBC with CSV Format** topic.

The following steps, which are described in detail in the text below, are specific to this use case:

- **Map Data** to include the required action field and calculated attribute for date / time
- **Select Delivery Method** of JDBC, using the necessary settings

Map Data

With this example, data will be published to a table ('stepdata') that has the following layout:



The table has the potential to contain multiple rows representing the same object, and the 'datetime' column will hold information about the latest STEP revision date. This can be achieved with a calculated attribute, mapped in the export configuration as described below. For more information on calculated attributes, see the **Calculated Attributes** topic in the **Getting Started** guide.

Below is an example of an 'upsert' action in the Map Data step for the external database table shown above. This setup will direct the process to look for four configured attributes (ID, DATETIME, EAN, and Consumer Short Description) in the destination database table, and either insert the object (if not found), or update the object (if found):

Map Data

- <ID>
- <Name>
- <Parent ID>
- <Object Type Name>
- <Product-Override Child ID>
- <Is deleted>
- "Constant Value"
- <Page Number>
- + All Attributes
- Select Attribute
- + Classifications
- + Index Words
- + Product Classification Links
- + Product References
- + Asset References
- + Classification References
- + Entity References
- + STEP Workflow Task Info
- Multi level References
- Multi level Parent attributes
- Insert Referenced Objects
- + Custom Attributes
- + System Setup

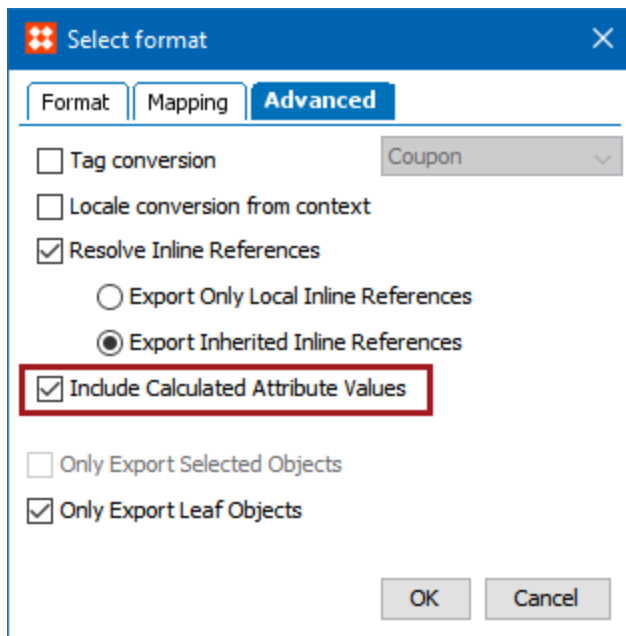
Exports data in a character separated format with one product per line.

Column (5 mapped)	
"upsert"	
Header	"action"
Value	"upsert"
<ID> ID	
Header	"id"
Value	<ID> ID
Last Edited Value and unit	
Header	"datetime"
Value	Last Edited Value and unit
EAN Value and unit	
Header	"ean"
Value	EAN Value and unit
Consumer Short Description Value and unit	
Header	"description"
Value	Consumer Short Description Value and unit

- The column headers in the exported CSV file must match the table column headers in the destination database. Map a constant value on the Header row for each mapped object to supply the external database table column header. For example, notice that the external table has a row named 'description' but is mapped to the STEP attribute 'Consumer Short Description.' The header parameter is used to make the exported data match the external table. For details about using the constant value data source, see the **Constant Value - Data Source Outbound** topic.
- Create an additional mapped column with the header of 'action' and the appropriate value of either delete or upsert. Use the transformation button to change the text displayed for both the Header and the Value.
- Create a calculated attribute using the function 'revisioneditdate()' and make it valid for your exported products. Then map the calculated attribute for export and update the header to match the column in the external table.

Details on this function are included in the **Other Functions** topic of the **Calculated Attributes** section of the **System Setup / Super User Guide**.

The example above uses the 'datetime' header, which is mapped to the 'Last Edited' calculated attribute. In addition to the other steps required to enact the JDBC method, the 'Include Calculated Attribute Values' parameter on the 'Advanced' step (shown below), must be checked.



If using an OIEP, one output template would be created to handle Create and Modify events, and a second output template would handle Delete events. In this way, separate mapping is available for each, allowing for one to include the upsert action and the other to include the delete action.

Select Delivery Method

The appropriate delivery configuration for this scenario can be seen below. Notice that both the 'id' and the 'datetime' column values are used as keys for the upsert action, meaning both values must match to determine whether to insert or update the object. When an exact match for both keys is found, the object is updated; when no match is found, the object is inserted.


Edit Delivery Configuration

Select Delivery Method

JDBC

Driver Location

L:/jdbc-drivers/mysql-connector-java-5.0.8-bin.jar

Driver Class

com.mysql.jdbc.Driver

Database URL

jdbc:mysql://localhost:3306/mydb

Username

root

Password

••••

Table Name

stepdata

Key Columns

id,datetime

Delete Key Columns

id

OK

Cancel

The value of this configuration is that if multiple instances of an object with the same ID but different datetime values (representing, for instance, different revisions of the object) are exported, the upsert action will insert into the table all instances of the single object. The effect of this setup is that various revisions of the same STEP data will be published to the table, giving users the ability to view historical data for objects in a data analytics dashboard. Further, if the Web UI has been configured to display data analytics dashboards, then Web UI users can view historical data on STEP objects.

cXML Format

The cXML format exports data using Ariba cXML 1.2 and is an XML-based language designed specifically for communication of e-commerce business documents. For more information, search the web.

Format Availability

cXML is available for selection in:

- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

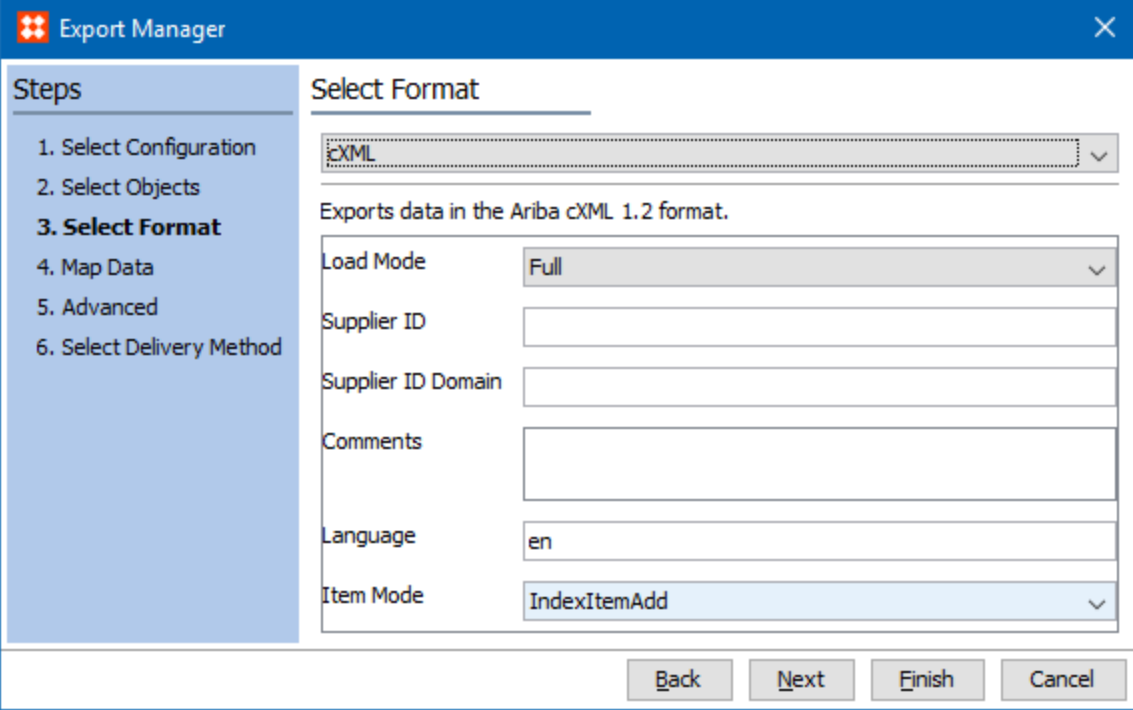
Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

Outbound Data

The same parameters are available in both Export Manager and OIEP.

Export Manager



The screenshot shows the 'Export Manager' dialog box with a blue header bar. On the left, a 'Steps' sidebar lists six steps: 1. Select Configuration, 2. Select Objects, 3. Select Format (highlighted), 4. Map Data, 5. Advanced, and 6. Select Delivery Method. The main area is titled 'Select Format' and features a dropdown menu with 'cXML' selected. Below this, a description states 'Exports data in the Ariba cXML 1.2 format.' A form contains several fields: 'Load Mode' (dropdown with 'Full' selected), 'Supplier ID' (text box), 'Supplier ID Domain' (text box), 'Comments' (text box), 'Language' (text box with 'en' entered), and 'Item Mode' (dropdown with 'IndexItemAdd' selected). At the bottom, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

OIEP

Background ProcessesStatisticsError Log ExcerptsLogStatus

Outbound Integration EndpointConfigurationEvent Triggering Definitions

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor	Post-Processor
Item (Modify)	cXML (6 mappings)	None	None
Add configuration			

Delivery Methods

Select format

FormatMappingAdvanced

cXML

Exports data in the Ariba cXML 1.2 format.

Load Mode

Full

Supplier ID

Supplier ID Domain

Comments

Language

en

Item Mode

IndexItemAdd

OK

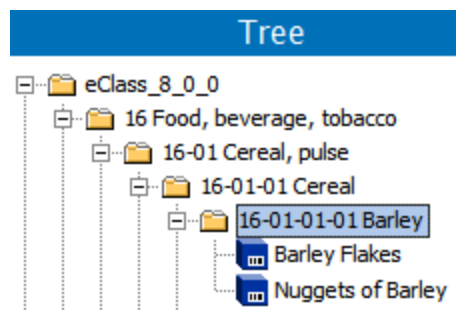
Cancel

eCl@ss Format

eCl@ss® is a hierarchical classification and product description system for grouping materials, products, and services. This includes a logical structure with a level of detail that corresponds to product-specific attributes. These classifications and standardized attributes are an international standard and can be used for reliable information exchange between suppliers and their customers. For detailed information about eCl@ss, search the web.

When your STEP system includes the eCl@ss license, you can import the Basic CSV and XML formats for any language using versions 7.0, 7.1, 8.0, 8.1, 9.0, and 9.1 and 10.0.1. For more information on the eCl@ss license, contact your Stiibo Systems account manager.

When an eCl@ss classification exists in STEP, you can make identification of your objects easier by using Bulk Update to add a 'product-to-classification link type' reference between your objects and the eCl@ss classification system. This displays your materials, products, and/or services within the eCl@ss folder as shown in the image below. For more information, see **Bulk Updates** documentation.



Format Availability

eCl@ss is available for selection only in the eCl@ss Classification Import wizard. For details, see **eCl@ss Classification Import**.

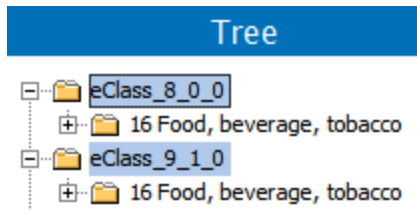
Mapping

There is no need to recreate attributes you already have in STEP to gain the benefits of using standard eCl@ss attributes. After importing an eCl@ss classification hierarchy, you can map each eCl@ss attribute to a STEP attribute to merge the attributes and delete the eCl@ss attribute. For details, see **Map eCl@ss Attributes to STEP Attributes**.

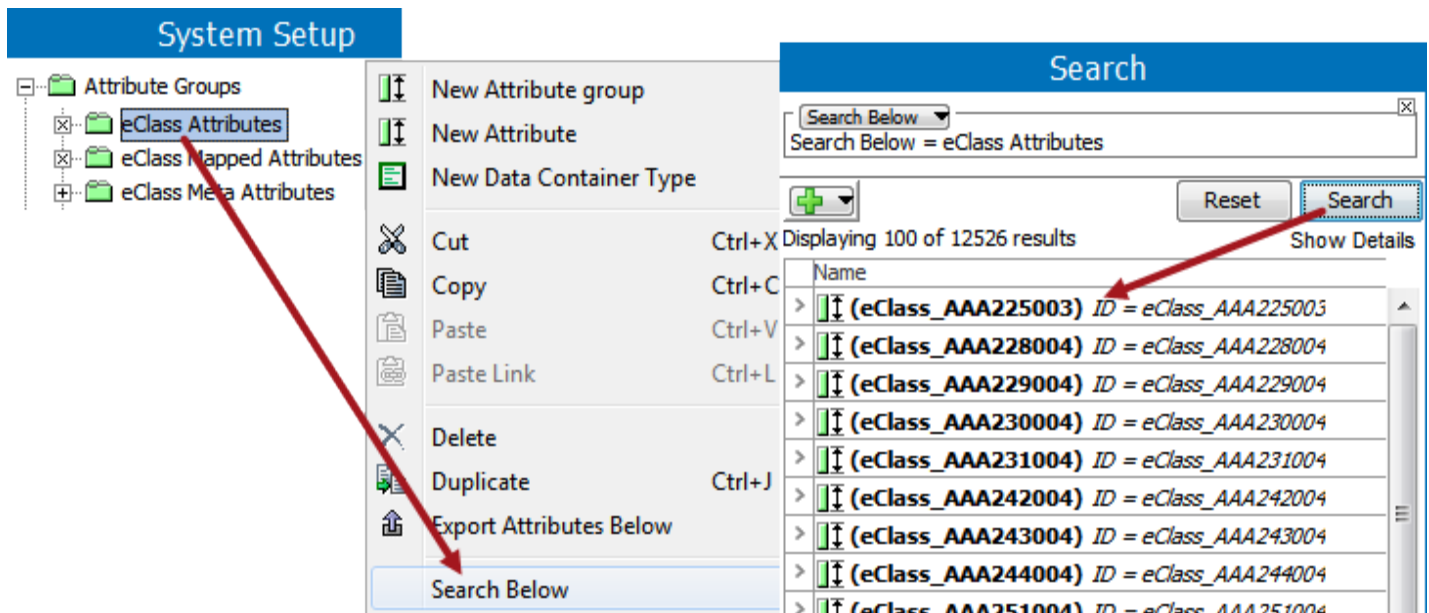
Important: Typically, industry standard attributes should not be altered, so limiting access to eCl@ss attributes via user privileges is recommended.

Multiple Versions

The eCl@ss classification data is delivered in a .ZIP file, where the file name indicates the version. The import wizard uses a background process to create classification folders that include the version number. This allows you to access multiple versions of eCl@ss simultaneously.



The import wizard also creates eCI@ss attributes, and saves all versions to the same attribute group. Depending on the number of classifications imported, System Setup may not allow browsing through the attributes created. However, you can view a portion of them, or find a particular one, using the right-click 'Search Below' functionality.



Any units or LOVs created upon import of an eCI@ss XML file will also be stored in their own unit / LOV groups labeled 'eCI@ss', with the version specified as a metadata attribute on the object.

Important: Units and LOVs can only be imported via eCI@ss XML files, *not* CSV.

If you want to update existing products to a newer eCI@ss classification version, you can re-classify them using the Bulk Update operation 'Industry Classification Synchronization'. For more information, see the **References and Links Operations for Bulk Update** section of the **Getting Started** documentation.

It is also possible to create version specific classification object types. To create a version specific object type structure, select 'Version specific object types' on the **Advanced** step of the eCI@ss Classification Importer. Selecting this option during import will automatically create a new version specific object type structure for eCI@ss classifications if one does not already exist for that version.

Tree

+

 eCI@ss 9.1

-

 eClass 8.1

-

 23 Machine element, fixing, mounting

-

 23-01 Control element

-

 23-01-01 Handle (control elem.)

-

 23-01-01-01 Button (control elem.)

-

 23-01-01-02 Palm grip

-

 23-01-01-03 Bow-type handle

-

 23-01-01-90 Handle (control elem.)

+

 23-01-02 Crank (control elem.)

<

Classification Sub Products References Referenced By Images & Documents

Description

Name	>	>	Value
ID			eClass_8.1_AAW867011
Name			23-01-01-01 Button (control elem.)
Object Type			eClass 8.1 CommodityClass
Revision			0.1 Last edited by USER3 on Thu Jun 08 09:16:44 EDT 2017
Approved			✖ Never Been Approved

For more information, see the **eCI@ss Classification Importer** section of the documentation.

Inbound Data

eCI@ss Classification Import

eCI@ss Classification Import

×

Steps

1. Select File

2. Select Classifications

3. Set Attribute Validity

4. Advanced

Select File

Select an eCI@ss classification archive.

fit\OneDrive - Stibo\8.2 Projects\173090100enUSbasicXML01.zip...

eClass®
CLASSIFICATION AND PRODUCT DESCRIPTION

Back

Next

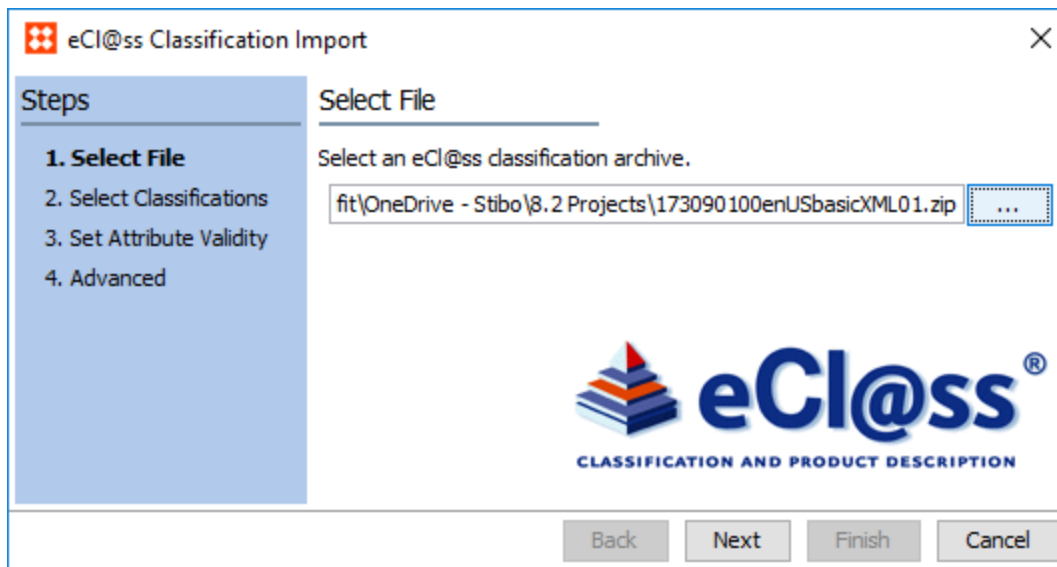
Finish

Cancel

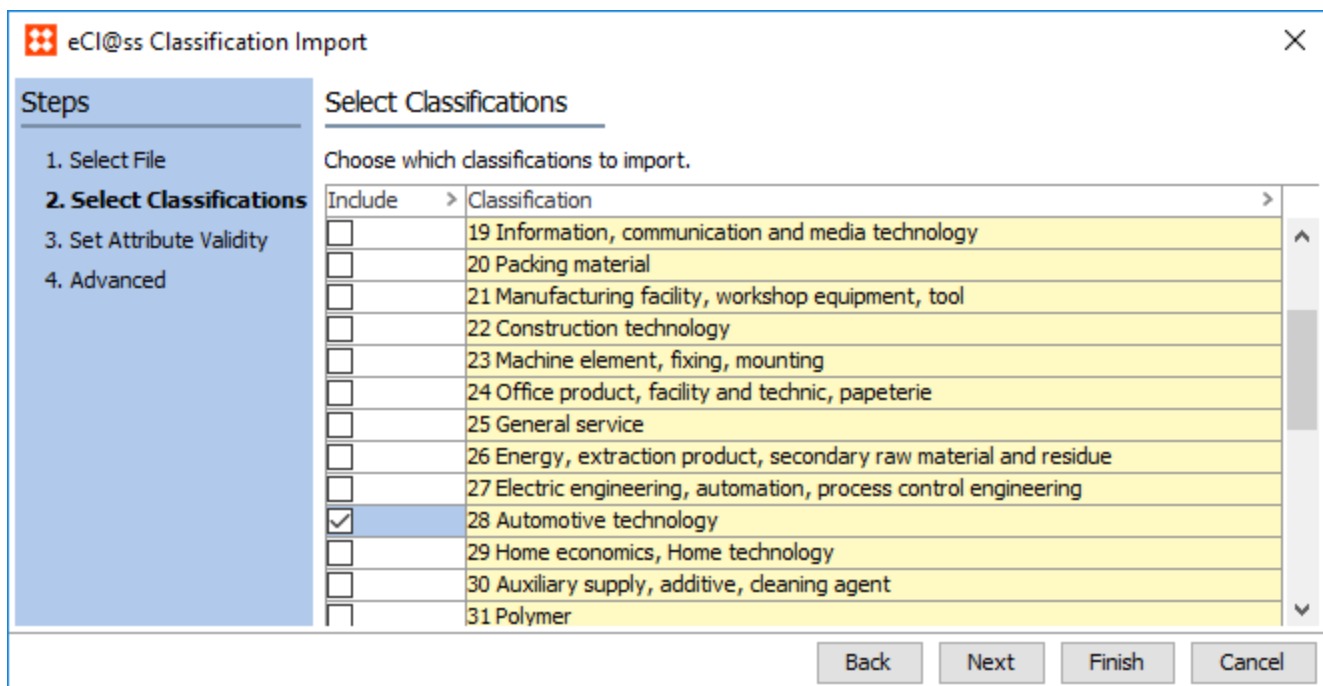
eCl@ss Classification Import

1. On File menu > Import > click eCl@ss... to open the **eCl@ss Classification Import** wizard.
2. For **Select File**, click the ellipsis button (...), and search or browse for the relevant eCl@ss .ZIP file from your local machine, click the Select File button, and click **Next**.

Note: The language of the file is indicated within the file name. For example, in the image below, '...enUS...' indicates that the file contains English / US.



3. For **Select Classifications**, select all classifications to import, and click **Next**.



- For **Set Attribute Validity**, select which product object types the imported eCI@ss attributes should be valid for. This step is optional, and only available for eCI@ss Basic XML imports.

Note: If an imported attribute already exists in the system, any selection made on this step will be added to the list of valid object types for the existing attribute.

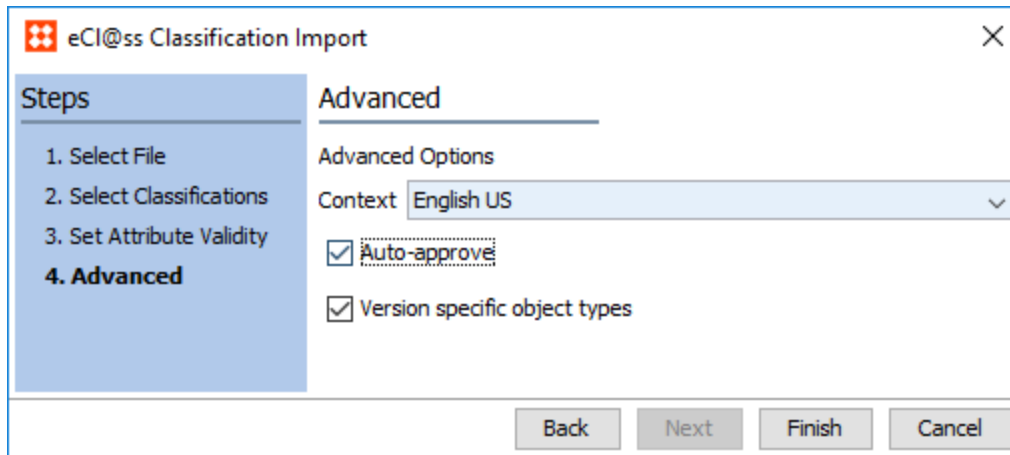
Important: If attribute validity is not set during the import it will have to be set manually later, otherwise you will not be able to maintain these attributes on linked eCI@ss products.

The screenshot shows the 'eCI@ss Classification Import' window. On the left, a 'Steps' pane lists: 1. Select File, 2. Select Classifications, 3. Set Attribute Validity (highlighted), and 4. Advanced. The main area is titled 'Set Attribute Validity' and contains a list of checkboxes for product object types: Pallet, Primary Product Classification, Product-override, Product Overrides, Products, Sales Item (checked), SalesItemFamily (checked), Sales Item Folder (checked), Sell Side, and SupplierCategory. At the bottom are buttons for Back, Next, Finish, and Cancel.

- For **Advanced**, select the context for the language of the eCI@ss file you are importing. Check the 'Auto-approve' checkbox if imported classifications should be approved automatically.

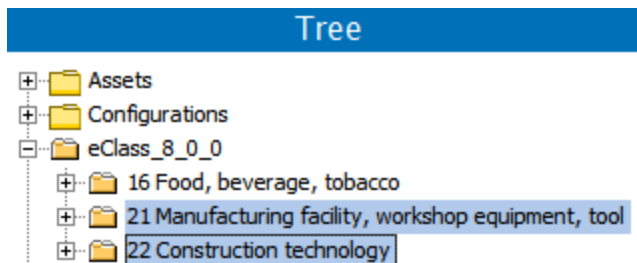
Click 'Version specific object types' if the importer should specify the eCI@ss version in the object types of the imported classifications. This will automatically create a new version specific object type structure for eCI@ss classifications if one does not already exist for that version.

Click **Finish** to begin the import.



Important: It is not possible to import a classification structure with the 'Version specific object types' option selected if classification object types had previously been created for classifications of the same version (but without this option selected). Attempting to do so will display an error in the background process and the import will fail.

6. Start the import. If required, refer to the information in **Running a Data Import** section. When the Import Manager Pipeline background process is complete, the newly created classification folders are displayed in the Tree.



7. Only complete this last step if you will import multiple eCI@ss languages for different contexts:
 - **Configure the language displayed in the Tree for the eCI@ss classifications:** In System Setup > Object Types & Structures > Alternate Classifications > eCI@ss Folder, for the top-level folder, set the Dimension Dependencies parameter to Language.
 - **Configure the language displayed for the eCI@ss attributes on the attribute editor:** In System Setup > Attribute Groups > eClass Meta Attributes, for every attribute in this group, set the Dimension Dependencies parameter to Language.
 - **Import additional eCI@ss data:** Run the import wizard again for each eCI@ss language required.

Map eCl@ss Attributes to STEP Attributes

eCl@ss classifications include standard eCl@ss attributes that are inherited downwards in the Primary Product Hierarchy.

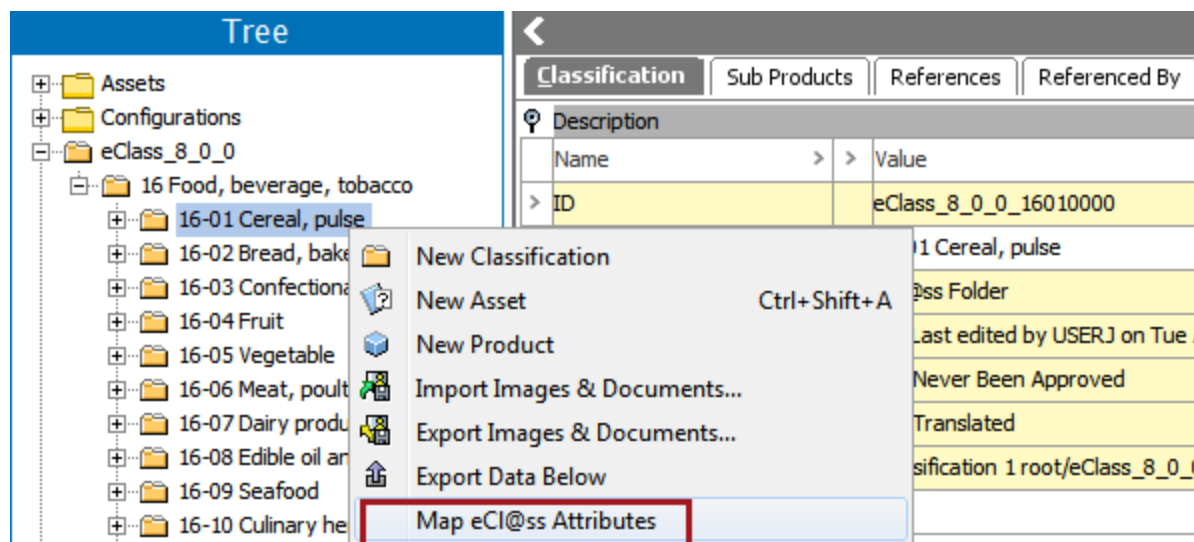
There is no need to recreate attributes you already have in STEP to gain the benefits of using standard eCl@ss attributes. After importing an eCl@ss classification hierarchy, you can map each eCl@ss attribute to a STEP attribute to merge the attributes and delete the eCl@ss attribute.

Note: The 'Map eCl@ss Attributes' option is not available for eCl@ss attributes imported via XML.

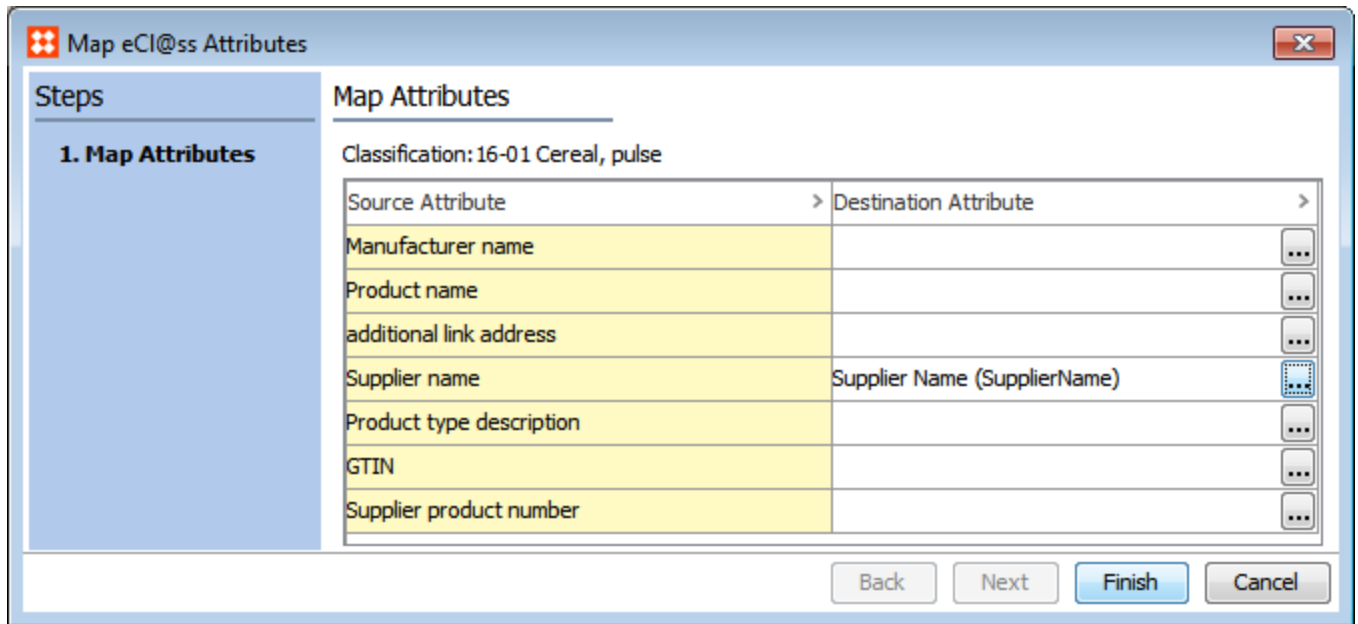
Important: Typically, industry standard attributes should not be altered, so limiting access to eCl@ss attributes via user privileges is recommended.

Mapping eCl@ss Attributes to STEP Attributes

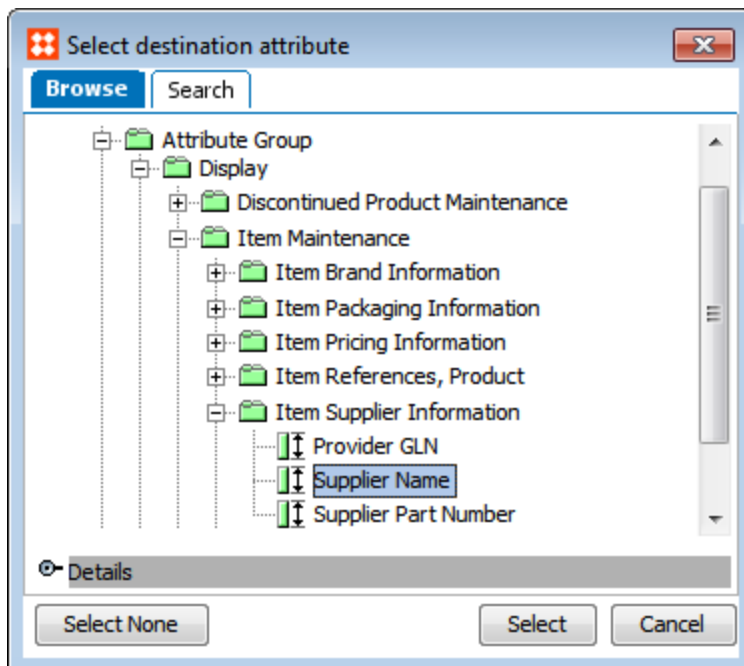
1. In the Tree, right-click the eCl@ss classification to map attributes *from* (the Source Attribute), and select the **Map eCl@ss Attributes** option.



2. In the wizard, click the ellipsis button (...) for the relevant eCl@ss attribute (the Source Attribute). The Source Attributes listed are those that exist for the selected classification.



3. Search or browse for the STEP attribute to map to (the Destination Attribute), select it and click the **Select** button. The eCI@ss attribute will be merged into this STEP attribute.



4. Click **Finish** on the Map eCI@ss Attributes wizard. Once merged, the eCI@ss attribute is deleted from both the available attributes in the wizard Source Attribute column and from the System Setup > eClass Attributes group folder.

ETIM Format

Electro-Technical Information Model (ETIM) is a classification structure used to standardize the electronic exchange of product data for electrical and electronic products and to enable the electronic trading of these products. The standard is designed to meet the requirements of the electrical industry and allows for a uniform, technical description of electrical goods and their assignment to a class of products. In order to support customers who need to work with data using the ETIM standard, the inbound data exchange functionality includes ETIM formats to allow for easy processing and setup of ETIM classifications and their related data elements. For more information on ETIM, search the web.

Format Availability

ETIM format is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**

Mapping

This format does not require a data map.

ETIM Configuration

Although the import tools do most of the configuration for ETIM classifications, some tasks must be performed manually.

1. Use Import Manager or IIEP and select the appropriate ETIM file version (as defined below) to perform the initial ETIM import. For more information, see **Creating a Data Import** or see **Creating an Inbound Integration Endpoint**.
2. Manually create the classification link from product to ETIM classification as defined in the **Create Product to ETIM Classification Link** documentation.
3. Manually make the ETIM attributes valid for the product object types as defined in the **Set Validity on ETIM Attributes for Products** documentation.

ETIM Versions

ETIM 5 and ETIM IXF classifications can be imported into STEP. The import tools cannot delete ETIM data or mark ETIM data as obsolete. However, ETIM IXF formats import the change code values and store them as metadata so it is possible to see if an attribute or classification has been marked for deletion or changed.

Multiple versions can exist in STEP simultaneously using multiple individual hierarchy structures. Attributes, LOVs, Units, Object Types, and Classifications are not shared between ETIM versions.

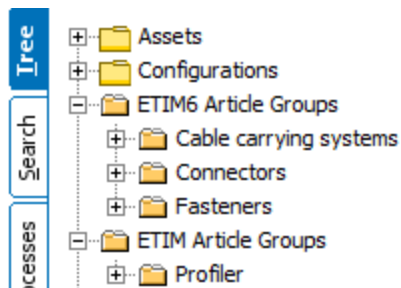
- ETIM 5 uses CSV and each file contains a single language. For more information about the ETIM 5 formats, see **ETIM and ETIM v2Format**.

- ETIM 6 and ETIM 7 use the ETIM IXF format. ETIM IXF is the ETIM XML standard and includes multiple languages (English, German, Dutch, and a number of other European languages). For more information about the ETIM IXF format, see **ETIM IXF Format**.

Important: To avoid a corrupt import, do not manipulate the ETIM files prior to import. Filtering of classifications should be handled by the import tools.

ETIM Classification

ETIM classifications are imported into STEP as an alternate classification, below the ETIM Article Groups node in the Tree.



Each imported ETIM article group will include the following Description attributes:

ETIM 5		ETIM IXF	
Attribute Label	Description	Attribute Label	Description
ID	Value displays ETIM preceding the ETIM ID	ID	Value displays the ETIM version preceding the ETIM ID
Name	The ETIM Description of the article group	Name	The ETIM Description of the article group
N/A	N/A	ETIM* Abbreviation	Abbreviation from ETIM
N/A	N/A	ETIM* Change Code	Change Code values from the ETIM file.
ETIM Date	The date of the last change to the classification in ETIM	ETIM* Date	The date of the last change to the classification in ETIM

ETIM 5		ETIM IXF	
Attribute Label	Description	Attribute Label	Description
ETIM Description	Description from ETIM, matches Name attribute	ETIM* Description	Description from ETIM, matches Name attribute
ETIM ID	ID from ETIM, matches ID attribute	ETIM* ID	ID from ETIM, matches ID attribute
ETIM Synonyms	Multivalued and contains 0 or more ETIM synonyms for the article classification	ETIM* Synonyms	Multivalued and contains 0 or more ETIM synonyms for the article classification
ETIM Version	The Article Classification version from the ETIM system	ETIM* Version	The Article Classification version from the ETIM system

* The ETIM version number will follow the ETIM naming convention for these description attributes. For instance, if the file is ETIM version 6, the attributes would be labeled ETIM6 followed by the actual label of the attribute.

Tree

- Assets
- Configurations
- ETIM6 Article Groups**
 - Cable carrying systems
 - Connectors
 - Accessories for industrial connectors
 - Adapter plate industrial connectors
 - Cable connector
 - Cap for industrial connectors
 - Chipcard connector
 - Circular connector (industrial connect
 - Coding for industrial connectors
 - Contact for industrial connectors
 - Contact insert for industrial connector
 - Contact insert holder for industrial cor
 - Fixing frame industrial connectors
 - Housing for industrial connectors
 - I/O connector
 - Modular connector (industrial connect
 - Printed circuit board connector
 - Printed circuit board terminal
 - Rectangular connector (industrial conr
 - RF connector
 - Fasteners
 - ETIM Article Groups**
 - Profiler
 - GPC

Accessories for industrial connectors rev.0.2 - Classification

Classification	Sub Products	References	Referenced By	Images & Documents	I
Description					
Name	>	>	Value		
ID			ETIM6 EC002943		
Name			Accessories for industrial connectors		
Object Type			ETIM6 Article Classification Type		
Revision			0.2 Last edited by USER on Tue Sep 29 13:18:55 PDT 2015		
Approved			✗ Never Been Approved		
Translation			Not Translated		
Path			Classification 1 root/ETIM6 Article Groups/Connectors/Accesso		
Visibility					
ETIM6 Abbreviation	abc				
ETIM6 Date	31				
ETIM6 Description	abc		Accessories for industrial connectors		
ETIM6 ID	abc		EC002943		
ETIM6 Synonyms	abc		Accessories for industrial connectors Screen connector De-mounting tool Bracket Handgrip Accessories for heavy duty connectors Strain relief Mounting tool		
ETIM6 Version	abc		1		

Attributes are linked to the relevant ETIM classification and inherited only in that classification. For example, when an item is classified in ETIM classification A and is then reclassified to ETIM classification B, the attributes from ETIM classification A become orphaned, although the values remain on the item. At the same time, the item will inherit the valid ETIM attributes from classification B.

ETIM Feature Types

ETIM operates with 'features' that roughly translate to attributes and LOVs. An ETIM feature can be of one of the following types.

Type	Type Description	Result
Logic	The logical data type holds two values: true / false or yes / no.	An LOV with the ID 'ETIM Logical' is created in STEP. It contains the values true and false. Features are imported as attributes using the LOV.

Type	Type Description	Result
Alphanumeric	The alphanumeric data type contains a set of predefined values.	An LOV contains the predefined set of values. Each value is provided with a value ID. Features are imported as attributes using the LOV.
Numeric	The numeric data type is a number that allows a unit.	Features are imported as Number validation type.
Range	The numeric range that allows a valid unit.	Features are imported as Number Range validation type.

Valid Values are Determined by ETIM Version

In ETIM, the set of valid values for a feature depends on the classification of the linked article and the ETIM version. In STEP, this is achieved by creating a new attribute for each classification the feature is linked into. Each of the attributes (for the same ETIM feature) use the same LOV but with an individual filter of the valid values.

Note: The core STEP solution does not offer any tools to validate that the user does not use an invalid unit when adding a value for an ETIM Classification.

ETIM 5

Typically, only one of the following ETIM 5 formats will be available in your import tools format list:

- ETIM - this import converter creates a new attribute for each classification the feature is linked into. Each of the attributes (for the same ETIM feature) uses the same LOV but with an individual LOV filter of the valid values.
- ETIM v2 - this import converter avoids having multiple attributes for the same feature by moving the LOV filter specifying the valid values from the attribute to the attribute link. This should make it far easier to import, export, and maintain the attributes.

For more information about the ETIM 5 formats, see **ETIM and ETIM v2 Format**.

ETIM IXF

- ETIM IXF, versions 6 and 7- this import converter creates a new attribute for each classification the feature is linked into. Each of the attributes (for the same ETIM feature) uses the same LOV but with an individual LOV filter of the valid values.

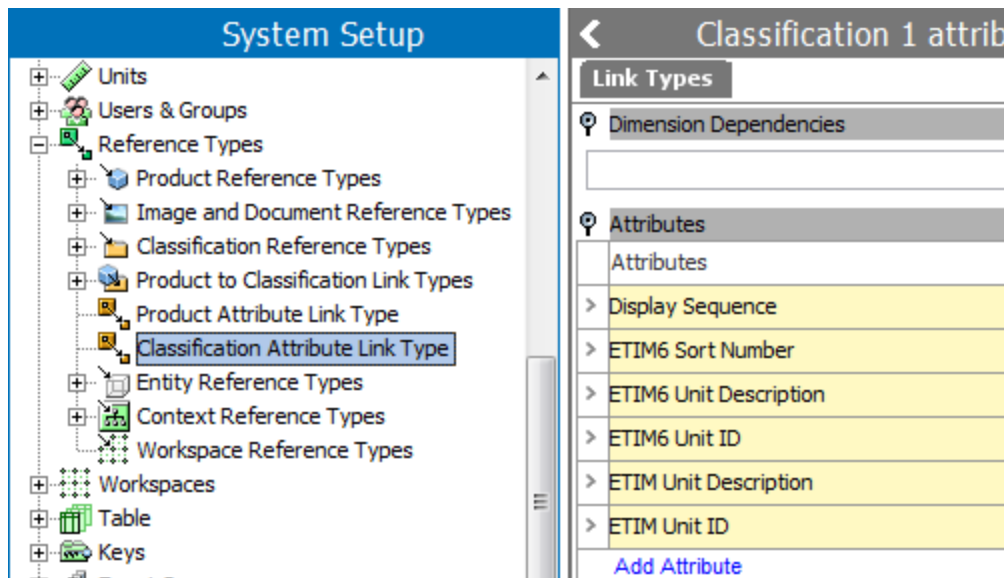
For more information about the ETIM IXF format, see **ETIM IXF Format**.

Valid Units are Determined by Classification to Attribute Link

An ETIM feature does not have a valid unit in ETIM 5 or ETIM IXF. However, when it is linked into an article classification, one (and only one) valid unit can be specified. This is different from STEP where an attribute has the same valid units no matter where the attribute is used.

Note: STEP does not include functionality to ensure only valid units are used when adding a value for an ETIM classification.

On the **Classification to Attribute Link**, the **ETIM Unit ID** and **Description** are stored as metadata to indicate that when used in this particular classification, only the unit specified is valid. Additional metadata includes 'ETIM Sort Number', 'ETIM Unit Description', and 'ETIM Unit ID'. The 'ETIM Sort Number' specifies the order in which ETIM determines the attribute is most relevant. If System Settings > 'Default Attribute to use as Display Sequence Attribute' is specified, the sort number will be put as metadata in this attribute, instead of using the 'ETIM Sort Number'. Notice however, that STEP does not support display sequence settings on classifications.



A number of the ETIM standard units correspond to the units already available on STEP Workbench. When the ETIM classification system is imported, existing STEP units are reused if they correspond to the ETIM units. All of the ETIM unit values (new and existing) are populated for the metadata attributes ETIM Description and ETIM ID.

Note: It is possible to load various localizations of the ETIM classification system. For example, you can load an English version into the English context and a German version into the German context. While the IDs are the same across the various localizations, the descriptions of article groups, classifications, attributes, units, and lists of values are localized.

Handling Import Errors

Depending on the locale configuration and the format of the import files, the import tool may report one of the following errors. Although the report can be long, the error is likely caused by the format of two attributes:

- **ETIM Version** - In the ETIM configuration file the values of this attribute contain 4 characters (5,00) while STEP only expects 3 characters.
- **ETIM Date** - The date format in the ETIM configuration file is dd.mm.yyyy hh:mm:ss while the STEP attribute is defined as ISO Date and Time (yyyy-mm-dd hh:mm:ss).

These errors can be resolved by manually importing the problem attributes as defined in the **Manually Import ETIM Attribute Values** documentation.

ETIM and ETIM v2 Format

ETIM 5 uses CSV and each file contains a single language. Typically, only one of the following ETIM 5 formats will be available in your STEP Import Manager:

- **ETIM** - this import converter creates a new attribute for each classification the feature is linked into. Each of the attributes (for the same ETIM feature) uses the same LOV but with an individual LOV filter of the valid values. This results in duplicate attributes and multiple LOVs when more than one classification has multiple sets of valid values.
- **ETIM v2** - this import converter avoids having multiple attributes for the same feature by moving the LOV filter specifying the valid values from the attribute to the attribute link. This should make it far easier to import, export, and maintain the attributes.

For details about the ETIM versions available in STEP, see the **ETIM Format** documentation.

Format Availability

ETIM format is available for selection in:

- **IIEP** - see **Creating an Inbound Integration Endpoint**
- **Import Manager** - see **Creating a Data Import**

Mapping

This format does not require a data map.

Inbound Data

Inbound Parameters

- **Article Group ID(s)** allows you to list one or more classifications as follows:
 - A single Article Group ID to import one classification.
 - A comma-separated list of Article Group IDs to import several classifications.
 - Blank to import the full ETIM classification system.
- **Character Set** allows you to select the option that includes any special characters required by the data being imported.
- **Conversion Preview** displays the data selected in the Article Group ID(s) parameter in the selected language.

Note: Once the ETIM import completes successfully, additional manual configuration is required. For details see the **ETIM Configuration** section of the **ETIM Format** documentation.

Import Manager

Import Manager

×

Steps

1. Select Configuration

2. Select Data Source

3. **Select Format**

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Select Format

Format

ETIM v2

▼

Imports an ZIP archive containing an ETIM database in CSV format. Will set up classification and attributes.

Article Group ID(s)

EG000047, EG000033

Character Set

UTF-8

▼

Conversion Preview:

Description	Article Group ID
	EG000059
(Avloppsvatten) mottagningsenheter	EG017110
Anslutnings- och förbindningsteknik/Isolertekni	EG000047
Anslutningsdon	EG000058
Antenn- och satellitteknik	EG000033
Apparatlådor och skåp	EG000011

Back

Next

Finish

Cancel

IIEP

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
- 5. Configure Processing Engine**
 - 5.1. Select Sample File
 - 5.2. Select Data Source
 - 5.3. Select Format**
 - 5.4. Map Data
 - 5.5. Identify Objects
 - 5.6. Identify Destination
 - 5.7. Select Business Rules
 - 5.8. Advanced Settings
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Processing Engine : Select Format

Format

ETIM v2

Imports an ZIP archive containing an ETIM database in CSV format. Will set up classification and attributes.

Article Group ID(s)

EG000059

Character Set

UTF-8

Conversion Preview:

Description	Article Group ID
	EG000059
(Avloppsvatten) mottagningsenheter	EG017110
Anslutnings- och förbindningsteknik/Isolertekni	EG000047
Anslutningsdon	EG000058
Antenn- och satellittekni	EG000033
Apparatlådor och skåp	EG000011
Arbetsutrustning	EG010330
Batterier och laddningsaggregat	EG000053

Back

Next

Finish

Cancel

ETIM IXF Format

ETIM IXF is the ETIM XML format and includes multiple languages (English, German, Dutch, and a number of other European languages). The import converter creates a new attribute for each classification the feature is linked into. Each of the attributes (for the same ETIM feature) uses the same LOV but with an individual LOV filter for the valid values. STEP supports ETIM version 6 and 7 and STEP also supports the dynamic, or pre-release versions in between major releases. For more information about ETIM functionality in STEP, see **ETIM Format**.

Format Availability

ETIM IXF format is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**

Mapping

This format does not require a data map.

Inbound Data

Inbound Parameters

- **Language** allows selection of a single ISO Language Code to import. Multiple languages must be imported separately. In the Advanced Settings step you will select a context for the import.
- **Article Group ID(s)** allows the user to list one or more classifications as follows:
 - A single Article Group ID to import one classification.
 - A comma-separated list of Article Group IDs to import several classifications.
 - Blank to import the full ETIM classification system.
- **Prefix** allows the user to manage the taxonomies of the ETIM structures within STEP by prepending text for the prefix parameter value. By default, this will be read from the file being imported into STEP but it can be changed depending on how it should be displayed in STEP. If it is an ETIM version 6 file it will read ETIM6 and if it is ETIM version 7 it will read ETIM7. However, if the file is an ETIM pre-release file, sometimes known as a dynamic file, it will load with the ETIM version number followed by a '-D'. So an ETIM version 6 dynamic file will automatically be populated as ETIM6-D.

Note: This ETIM-D version number should only be changed if a user wants to see multiple taxonomies for each dynamic version of the ETIM 6 imports. Otherwise each dynamic ETIM file will continue to load into the ETIM6-D file structure.

- **Set validity for ETIM attributes** allows the user to select the object types that are valid for the ETIM attributes. ellipses Click the ellipsis button (...) to display a list of object types that can be checked for validity in both the Import Manager and IIEP.
- **Conversion Preview** displays article groups that were read from the ETIM import file in the selected language.

Note: This should complete the ETIM IXF configuration. If changes need to be made to the configuration, see the **ETIM Configuration** section of the **ETIM Format** documentation.

Import Manager

Import Manager

Steps

1. Select Configuration

2. Select Data Source

3. Select Format

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Select Format

Format

ETIM IXF

Imports an XML based file containing an ETIM database. Will set up Classifications, Attributes, LOVs, Units and Object Types.

Language

EN

Article Group ID(s)

EG000030

Prefix

ETIM7

Select validity for ETIM attributes

Pack, Pallet

Conversion Preview:

Description	Article Group ID
(Wastewater) reception devices	EG017110
Accessories for flue gases and air	EG015560
Accessories for lighting	EG000030
Accessories for measurement and control devices	EG015870

Back

Next

Finish

Cancel

IIEP

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
- 5. Configure Processing Engine**
 - 5.1. Select Sample File
 - 5.2. Select Data Source
 - 5.3. Select Format**
 - 5.4. Map Data
 - 5.5. Identify Objects
 - 5.6. Identify Destination
 - 5.7. Select Business Rules
 - 5.8. Advanced Settings
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Processing Engine : Select Format

Format

ETIM IXF

Imports an XML based file containing an ETIM database. Will set up Classifications, Attributes, LOVs, Units ...

Language

de-DE

Article Group ID(s)

EG000047

Prefix

ETIM7

Select validity for ETIM attributes

Pack, Pallet

Conversion Preview:

Description	Article Group ID
Alarmanlagen, Notruf- und Meldesysteme	EG000054
Anschluss- und Verbindungstechnik/Isoliermaterial	EG000047
Antennen und Satellitentechnik	EG000033
Batterie, Akku, Ladegerät, Netzkabel	EG000053

Back

Next

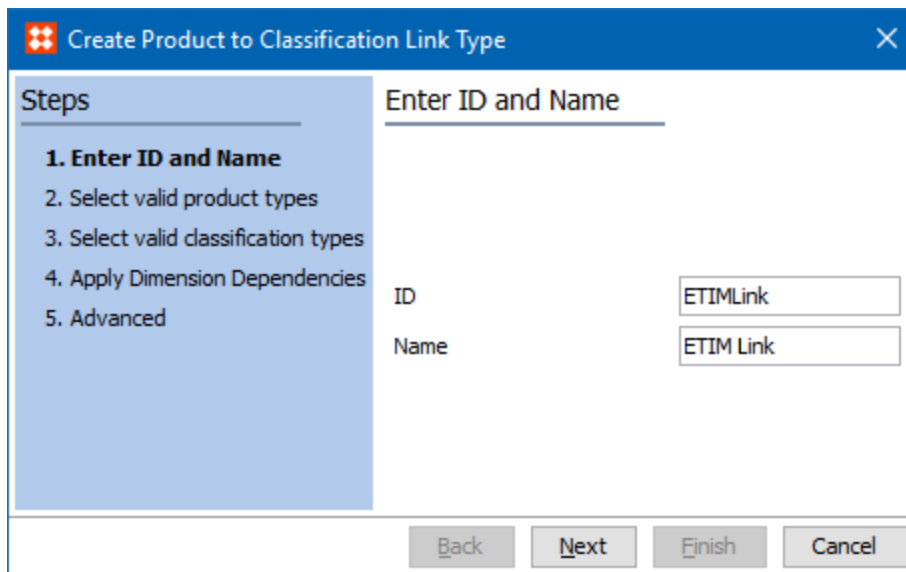
Finish

Cancel

Create Product to ETIM Classification Link

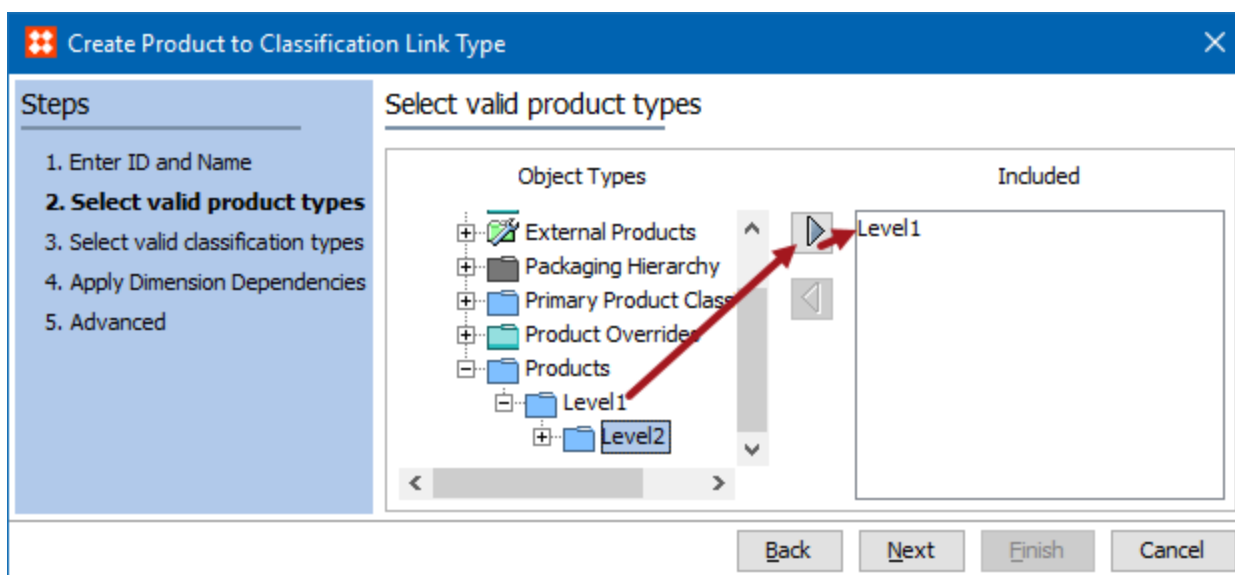
For details about other setup required for using ETIM classifications, see the **ETIM Configuration** section of the **ETIM Format** documentation.

1. System Setup > Reference Types > **Product to Classification Link Types**, right-click and create a new Product to Classification Link Type.
2. Type an ID and Name for the new link type and click the **Next** button.



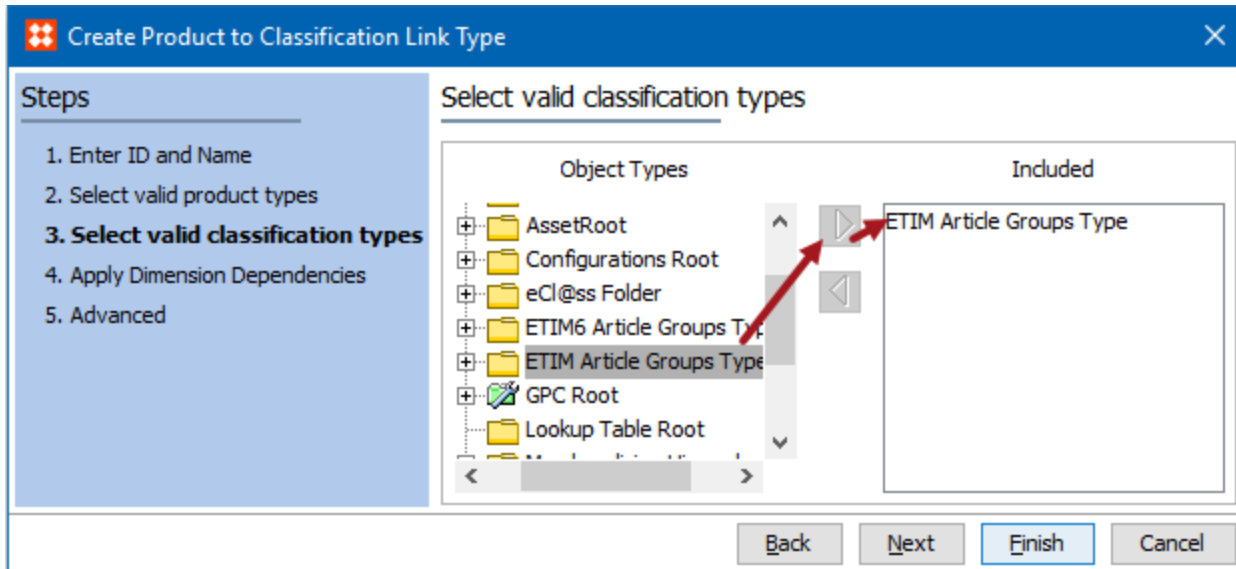
The screenshot shows the 'Create Product to Classification Link Type' dialog box. The title bar is blue with a Stibo logo and a close button. The main area is divided into two panes. The left pane, titled 'Steps', lists five steps: 1. Enter ID and Name (highlighted), 2. Select valid product types, 3. Select valid classification types, 4. Apply Dimension Dependencies, and 5. Advanced. The right pane, titled 'Enter ID and Name', contains two text input fields: 'ID' with the value 'ETIMLink' and 'Name' with the value 'ETIM Link'. At the bottom, there are four buttons: 'Back', 'Next' (highlighted), 'Finish', and 'Cancel'.

3. Select the valid product type(s), click the arrow button to move to the 'Included' list, and click the **Next** button.

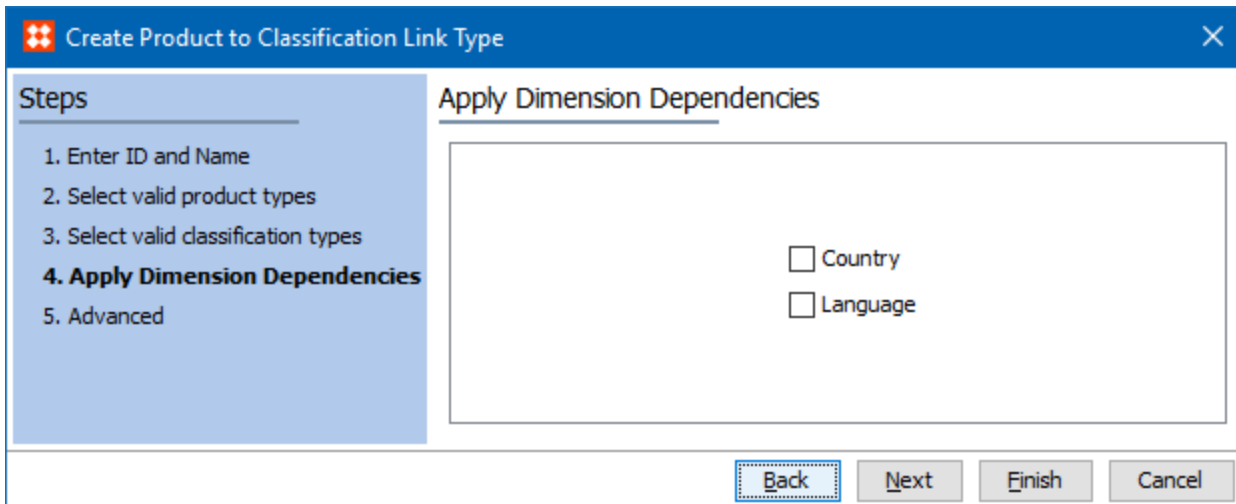


The screenshot shows the 'Create Product to Classification Link Type' dialog box, Step 2: Select valid product types. The left pane shows the 'Steps' list with '2. Select valid product types' highlighted. The right pane, titled 'Select valid product types', contains a tree view of 'Object Types' and an 'Included' list. The 'Object Types' tree includes 'External Products', 'Packaging Hierarchy', 'Primary Product Class', 'Product Overrides', 'Products', 'Level1', and 'Level2'. A red arrow points from 'Level1' in the tree to the 'Included' list, which currently contains 'Level1'. At the bottom, there are four buttons: 'Back', 'Next' (highlighted), 'Finish', and 'Cancel'.

4. Select the valid classification type(s), click the arrow button to move to the 'Included' list, and click the **Next** button.



5. Select any necessary dimension dependencies and click the **Next** button.



6. For the Advanced options, uncheck all check boxes and for Sub Products Inheritance Settings, select 'Inherit Links and Tables and Specification Attributes'.

Create Product to Classification Link Type
✕

Steps

1. Enter ID and Name
2. Select valid product types
3. Select valid classification types
4. Apply Dimension Dependencies
- 5. Advanced**

Advanced

☐ Allow multiple links

☐ Externally Maintained

☐ Mandatory

Sub Products Inheritance Settings

Inherit Links and Tables and Specification Attributes
 ▼

Back

Next

Finish

Cancel

7. Click the **Finish** button and verify the correct settings are displayed in the editor.

System Setup

- Reference Types
 - Product Reference Types
 - Image and Document Reference Types
 - Classification Reference Types
 - Product to Classification Link Types
 - Classifications
 - eCI@ss
 - ETIM Link**
 - GPC
 - Merchandising Link
 - Object Creation Templates
 - Supplier Link
 - UNSPSC Class
 - Website Link
 - Product Attribute Link Type
 - Classification Attribute Link Type
 - Entity Reference Types
 - Context Reference Types
 - Workspace Reference Types
 - Workspaces
 - Table

<
 ETIM Link - Product to Classification Link Type
→

Product to Classification Link Type

Validity

Log

Description

Name	Value
ID	ETIMLink
Name	ETIM Link
Last edited by	2016-12-19 15:19:11.362 by USERJ
Externally Maintained	No
Dimension Dependencies	
Allow multiple links	No
Mandatory	No
Inheritance of Links	Inherited
Inheritance of Specificatio...	Yes
Ignore LOV Filter definitio...	No
Completeness Score	123

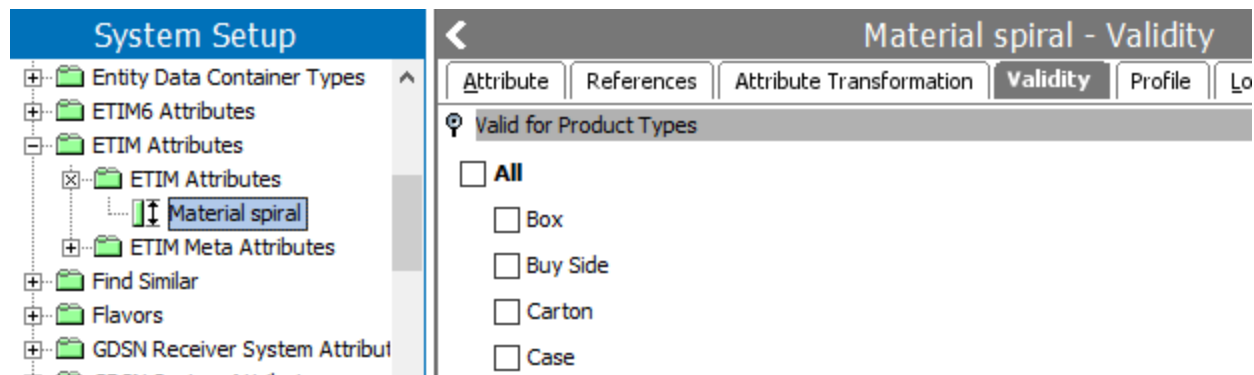
In Attribute Groups

Valid Attributes

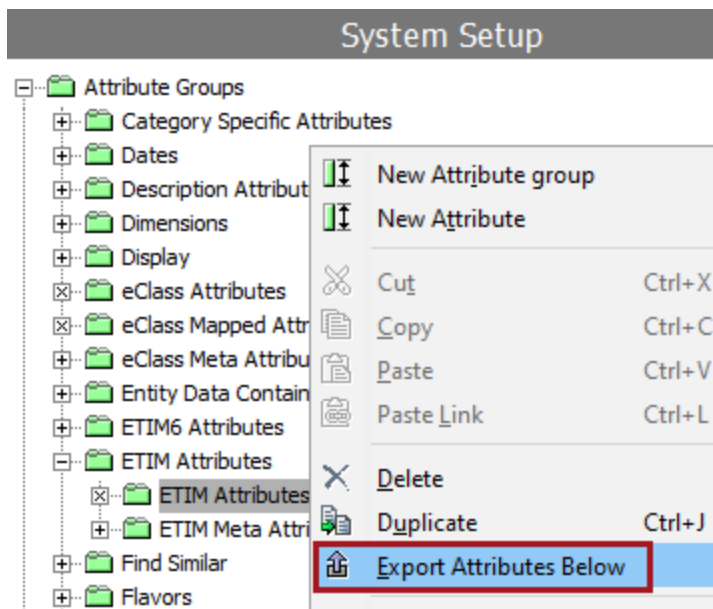
Set Validity on ETIM Attributes for Products

For details about other setup required for using ETIM classifications, see the **ETIM Configuration** section of the **ETIM Format** documentation.

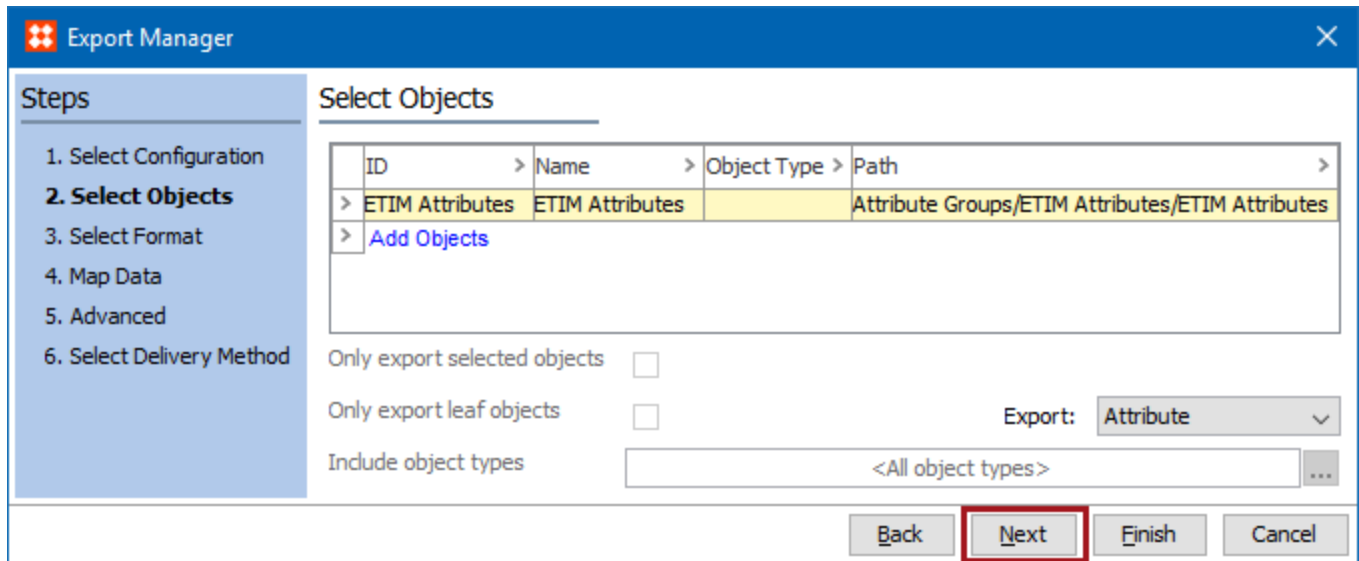
Use the following steps to set the valid product type on the Validity tab for the exported ETIM attributes.



1. Determine the product object type that should be selected on the ETIM Attributes Validity tab.
2. System Setup > Attribute Groups > ETIM Attributes > ETIM Attributes, right-click and select **Export Attributes Below**. This assumes that you want to set validity on all ETIM attributes. If needed, select only the subset that you want to modify and export those.



3. The Export Manager displays with the selected ETIM Attributes, click the **Next** button.



Export Manager

Steps

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

Select Objects

ID	Name	Object Type	Path
> ETIM Attributes	ETIM Attributes		Attribute Groups/ETIM Attributes/ETIM Attributes
Add Objects			

Only export selected objects ☐

Only export leaf objects ☐

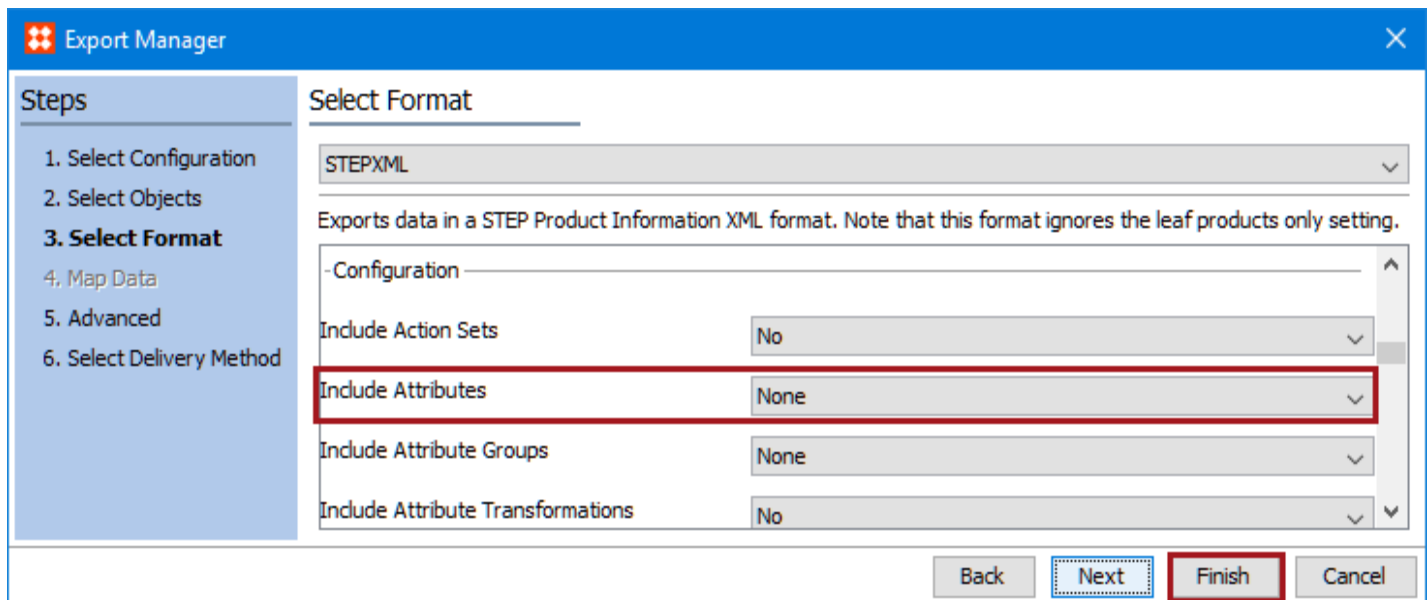
Export: **Attribute** ▼

Include object types ...

Back **Next** **Finish** **Cancel**

4. To ensure the smallest XML export file, on Select Format step:

- Set **STEPXML** as the format.
- Set the **Include Attributes** parameter to **Selected**.
- Set all other parameters to No / None.
- Click **Finish**.



Export Manager

Steps

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

Select Format

STEPXML ▼

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

- Configuration -

Include Action Sets ▼

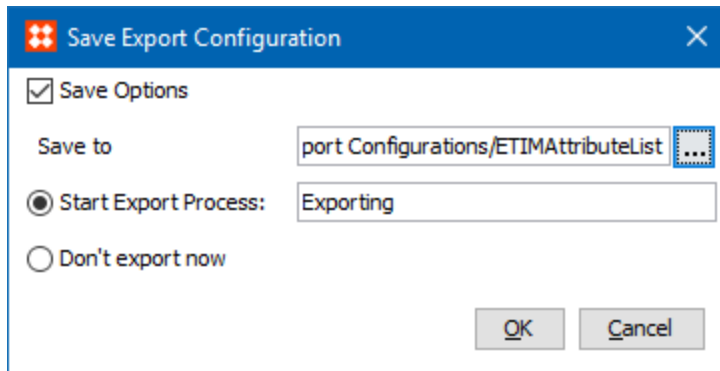
Include Attributes ▼

Include Attribute Groups ▼

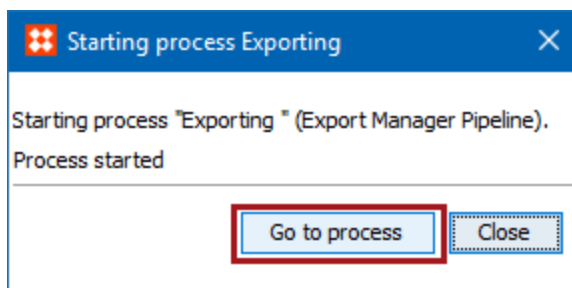
Include Attribute Transformations ▼

Back **Next** **Finish** **Cancel**

5. On the Save Export Configuration dialog, set the options as desired, and click **OK** to start the export.



- Click the **Go to process** button.



- Open the export file for editing and make the following updates as shown in the image below. In the example, 'Item' is the product object type that will be set as valid.

Below each instance of the `<AttributeGroupLink AttributeGroupID="ETIM Attributes"/>` tag, add the text `<UserTypeLink UserTypeID="Item"/>`, substituting the text 'Item' with the name of your own desired product object type determined in the first step. The double quote marks are required.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Configuration:
<STEP-ProductInformation ResolveInlineRefs="true">
<AttributeList ExportSize="Selected"/>
</STEP-ProductInformation>

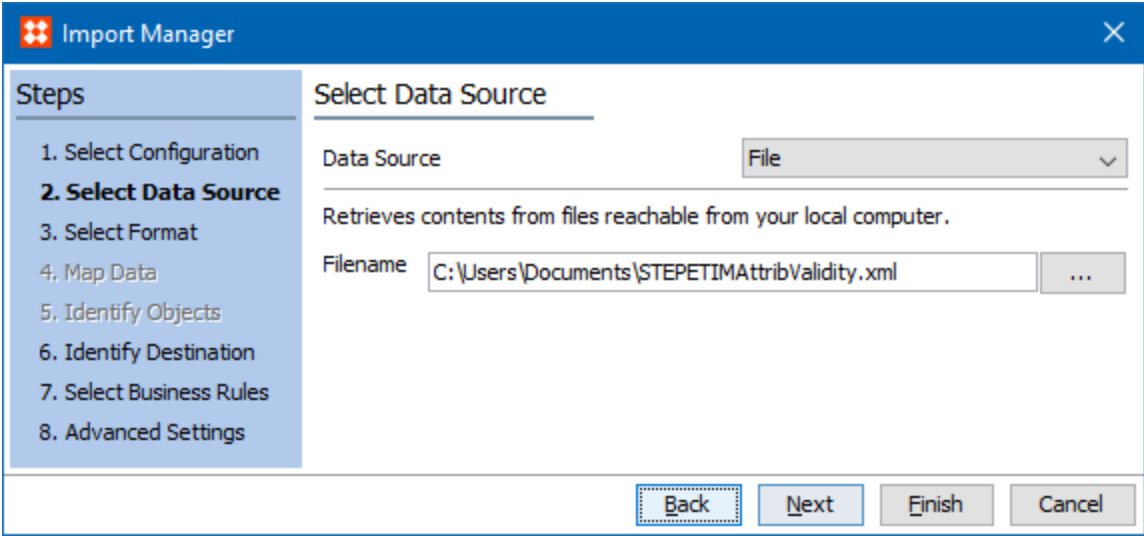
Export from Primary Product Hierarchy
Classifications All
Products All
Assets All
-->
<STEP-ProductInformation ExportTime="2017-01-05 09:35:33" ExportContext="Context1"
ContextID="Context1" WorkspaceID="Main" UseContextLocale="false">

  <AttributeList>

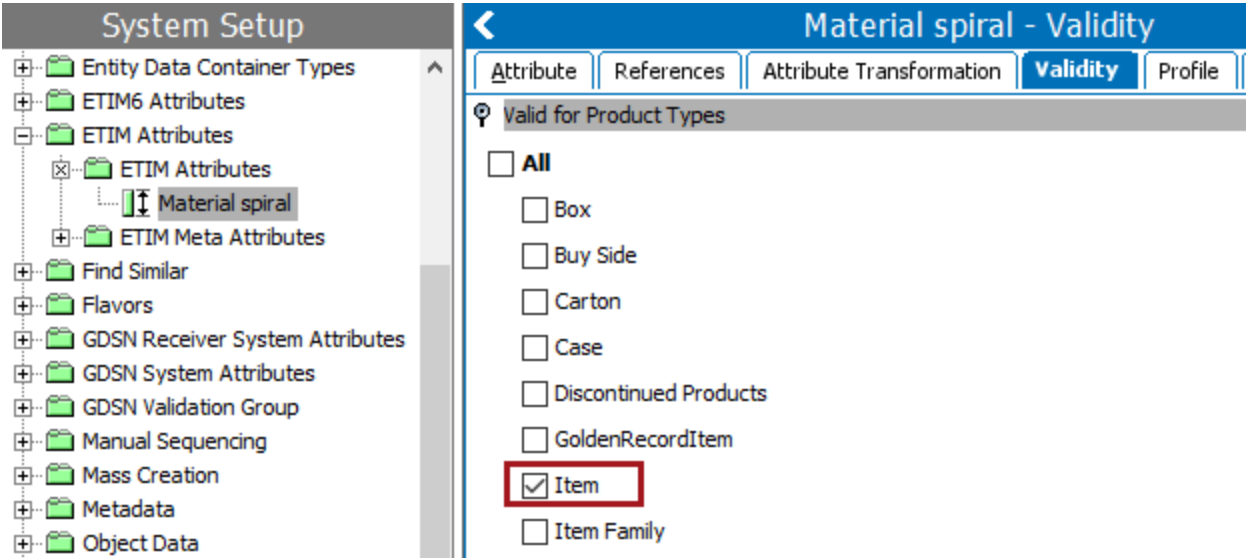
    <Attribute ID="ETIM EF021685 A" MultiValued="false" ProductMode="Normal" FullTextIndexed="false"
ExternallyMaintained="false" Derived="false" HierarchicalFiltering="true"
ClassificationHierarchicalFiltering="true" Selected="true" Referenced="true">
      <Name>Material spiral</Name>
      <ListOfValueLink ListOfValueID="ETIM EF021685"/>
      <AttributeGroupLink AttributeGroupID="ETIM Attributes"/>
      <UserTypeLink UserTypeID="Item"/>
    </Attribute>

    <Attribute ID="ETIM EF008218 A" MultiValued="false" ProductMode="Normal" FullTextIndexed="false"
ExternallyMaintained="false" Derived="false" HierarchicalFiltering="true"
ClassificationHierarchicalFiltering="true" Selected="true" Referenced="true">
      <Name>Position cable feed</Name>
      <ListOfValueLink ListOfValueID="ETIM EF008218"/>
      <AttributeGroupLink AttributeGroupID="ETIM Attributes"/>
      <UserTypeLink UserTypeID="Item"/>
    </Attribute>
```

8. Save the modified file with a new name.
9. Open Import Manager (File > Import > Data) and import the modified file. For details on the import process, see **Creating a Data Import**.



10. Verify that imported attributes now display the valid product type.



Manually Import ETIM Attribute Values

Occasionally, while using Import Manager to import a set of ETIM classifications, ETIM attribute values are not imported because the number of allowed characters / or the validation base type in STEP does not match the values included in the ETIM file. When this happens, use the following steps to manually import the ETIM attributes.

1. Start the Import Manager and complete the following steps:
 - **Import Manager - Select Configuration**
 - **Import Manager - Select Data Source**
2. For the Import Manager Select Format step, select the appropriate ETIM version.
 - ETIM version 5 using **ETIM and ETIM v2 Format**
 - ETIM versions 6 and 7 using **ETIM IXF Format**
3. On Import Manager Select Format, choose **CSV** and leave the other settings as default. The Conversion Preview is displayed.

Import Manager

Steps

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

Select Format

Format: **CSV**

Converter for character separated formats where each line contains data about one object in fields separated ...

Delimiter: **;**

Character Set: **windows-1252**

Has Header: **yes**

Trim whitespace: **no**

Allow Multi Line Values: **no**

Conversion Preview:

ARTCLASSID	ARTGROUPID	ARCLASSDESC	ARTCLASSVERSI...	ARTCLASSVERSI...
EC000001	EG000017	Busbar terminal	5,00	18.5.2011 00:00:00
EC000003	EG000020	Residual current cir...	5,00	18.5.2011 00:00:00
EC000005	EG000047	Cable end sleeve	5,00	18.5.2011 00:00:00

Back **Next** **Finish** **Cancel**

4. On Import Manager Map Data, set **Map to** as **Classification**.

Import Manager

Steps

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

Map Data

Source:

ARTCLASSID	ARTGROUPID	ARCLASSDESC	ARTCLASSVERSI...	ARTCLASSVERSI...
EC000001	EG000017	Busbar terminal	5,00	18.5.2011 00:00:00
EC000003	EG000020	Residual current cir...	5,00	18.5.2011 00:00:00

Result:

Map to: **Classification**

Auto Map Map Constant Remove Transform Generate Profile

Back Next Finish Cancel

5. In the Source table, select the **ARTCLASSID** column and click the **Map** button below the Result table.

Import Manager

Steps

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

Map Data

Source:

ARTCLASSID	ARTGROUPID	ARCLASSDESC	ARTCLASSVERSI...	ARTCLASSVERSI...
EC000001	EG000017	Busbar terminal	5,00	18.5.2011 00:00:00
EC000003	EG000020	Residual current cir...	5,00	18.5.2011 00:00:00

Result:

Map to: **Classification**

Auto Map **Map** Constant Remove Transform Generate Profile

Back Next Finish Cancel

6. In the Map To window, select the **ID** radio button, set the **ID Aspect** to **ID**, and click **OK**.

Map ARTCLASSID to

☒ ID

☐ Name

☐ Attribute

☐ Product Reference

☐ Asset Reference

☐ Classification Reference

☐ Entity Reference

☐ Reference Meta-Data

☐ Parent

☐ Object Type

☐ Variable

☐ Multivalued Variable

ID Aspect: ID

☐ Mandatory

Cancel OK

7. In the Result table, select the **ID=ARTCLASSID** column and click the **Transform** button.

Import Manager

Steps

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

Map Data

Source:

ARTCLASSID	ARTGROUPID	ARCLASSDESC	ARTCLASSVERSI...	ARTCLASSVERSI...
EC000001	EG000017	Busbar terminal	5,00	18.5.2011 00:00:00
EC000003	EG000020	Residual current cir...	5,00	18.5.2011 00:00:00

Result:

ID=ARTCLASSID ✓
EC000001
EC000003
EC000005

Map to: Classification

Buttons: Auto Map, Map, Constant, Remove, **Transform**, Generate Profile, Back, Next, Finish, Cancel

8. In the Transformation dialog, click the **Add Transformation** link, select the **Insert text before** option, and click **OK**.

Specify the sequence of transformations

Target: ID

☐ Mandatory

Value - Source: ARTCLASSID

[Add Transformation](#)

Select Transformation

- Generate soundex code
- If Equals, Source
- Insert text at a given position
- Insert text before**
- Math - Add source
- Math - Divide by source
- Math - Multiply by source
- Math - Subtract source

Buttons: OK, Cancel

Preview

ID=ARTCLASSID ✓
EC000001
EC000003
EC000005
EC000006
EC000007
EC000008
EC000009
EC000010
EC000011
EC000012

Buttons: Save, Reset, Cancel

9. Enter the ETIM version you want to import (including a trailing space) as the value to insert in order to match the ID in STEP and click the **Save** button. This text entered is based on the suffix added when the ETIM Article Groups were imported. The results are displayed in the Preview pane.

Specify the sequence of transformations in the table below.

Target:ID

☐ Mandatory

Value - Source:ARTCLASSID

Insert ETIM before

Add Transformation

Save Reset Cancel

Preview

ID=ARTCLASSID ✓
ETIM EC000001
ETIM EC000003
ETIM EC000005
ETIM EC000006
ETIM EC000007
ETIM EC000008
ETIM EC000009
ETIM EC000010
ETIM EC000011
ETIM EC000012

10. The Result table ID=ARTCLASSID column now displays the selected text preceding the ID originally displayed in the Source table.

Import Manager

Steps

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

Map Data

Source:

ARTCLASSID	ARTGROUPID	ARCLASSDESC	ARTCLASSVERSI...	ARTCLASSVERSI...
EC000001	EG000017	Busbar terminal	5,00	18.5.2011 00:00:00
EC000003	EG000020	Residual current cir...	5,00	18.5.2011 00:00:00

Result:

Map to: Classification

ID=(ARTCLASSID) ✓

ETIM EC000001
ETIM EC000003
ETIM EC000005

Auto Map Map Constant Remove Transform Generate Profile

Back Next Finish Cancel

11. In the Source table, select the **ARTCLASSVERSION** column and click the **Map** button below the Result table.

Import Manager

Steps

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

Map Data

Source:

ARTCLASSID	ARTGROUPID	ARCLASSDE...	ARTCLASSVERSION	ARTCLASSVERSIONDATE
EC000001	EG000017	Busbar terminal	5,00	18.5.2011 00:00:00
EC000003	EG000020	Residual curre...	5,00	18.5.2011 00:00:00
EC000005	EG000047	Cable end sleeve	5,00	18.5.2011 00:00:00

Result:

Map to: **Classification**

ID=(ARTCLASSID) ✓

ETIM EC000001

ETIM EC000003

ETIM EC000005

Auto Map **Map** Constant Remove Transform Generate Profile

Back Next Finish Cancel

12. In the Map To window, select the **Attribute** radio button, search to find and select the **ETIM Version** attribute, and click **OK**.

Map ARTCLASSVERSIONDATE to

☐ ID

☐ Name

☒ **Attribute**

☐ Product Reference

☐ Asset Reference

Browse **Search**

ETIM Version (ETIM Version)

Search

Advanced

Name

> **ETIM Version ID = ETIM Version**

13. The ETIM Version attribute has a maximum length of three (3) characters, so it displays as an error in the Result table. Select the **ETIM Version=ARTCLASSVERSION** column and click the **Transform** button.

Import Manager

Steps

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

Map Data

Source:

ARTCLASSID	ARTGROUPID	ARCLASSDESC	ARTCLASSVERSI...	ARTCLASSVERSI...	
EC000001	EG000017	Busbar terminal	5,00	18.5.2011 00:00:00	^
EC000003	EG000020	Residual current cir...	5,00	18.5.2011 00:00:00	v

Result:

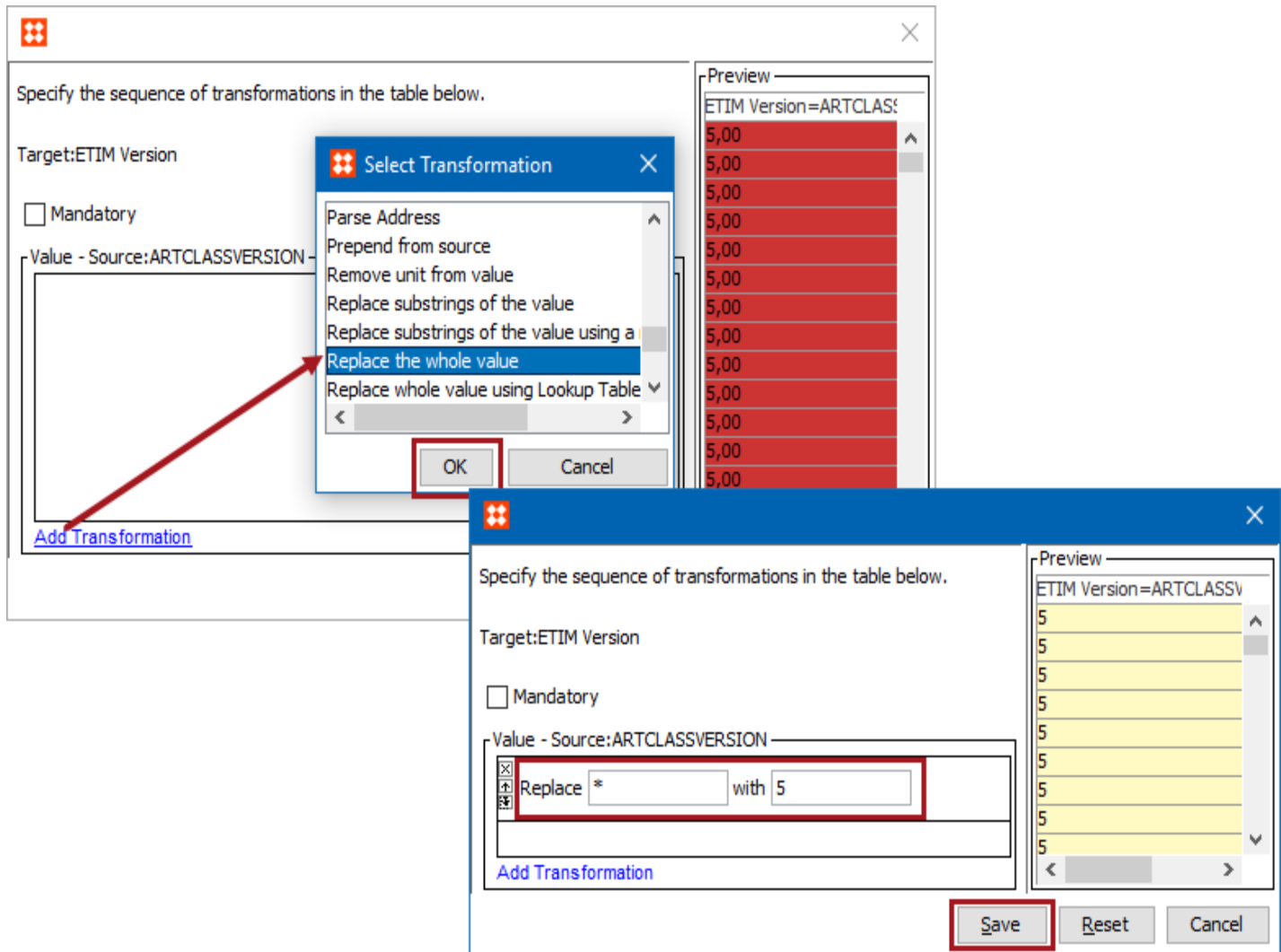
Map to: Classification v

ID=(ARTCLASSID) ✓	ETIM Version=ARTCLASSVERSION ⚠
ETIM EC000001	5,00
ETIM EC000003	5,00
ETIM EC000005	5,00

Auto Map
Map
Constant
Remove
Transform
Generate Profile

Back
Next
Finish
Cancel

- In the Transformation dialog, click the **Add Transformation** link, select the **Replace the whole value** option. For the values, add **replace * with** and enter a single digit based on the version being imported, and click the **Save** button.



15. The Result table ID=ARTCLASSVERSION column now displays the selected single digit for all rows. In the Source table, select the **ARTCLASSVERSIONDATE** column and click the **Map** button below the Result table.

Import Manager

Steps

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

Map Data

Source:

ARTCLASSID >	ARTGROUPID >	ARCLASSDESC >	ARTCLASSVERSI... >	ARTCLASSVERSIONDATE >
EC000001	EG000017	Busbar terminal	5,00	18.5.2011 00:00:00
EC000003	EG000020	Residual current cir...	5,00	18.5.2011 00:00:00
EC000005	EG000047	Cable end sleeve	5,00	18.5.2011 00:00:00

Result:

ID=(ARTCLASSID) ✓	ETIM Version=(ARTCLASSVERSION) ✓
ETIM EC000001	5
ETIM EC000003	5
ETIM EC000005	5

Map to: Classification

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

Buttons: Back, Next, Finish, Cancel

16. In the Map To window, select the **Attribute** radio button, search for and select the **ETIM Date** attribute, and click **OK**.

Map ARTCLASSVERSIONDATE to

☐ ID
☐ Name
☒ **Attribute**
☐ Product Reference
☐ Asset Reference

Browse Search

ETIM Date (ETIM Date)

Advanced

Name

> **ETIM Date** ID = ETIM Date

17. The ETIM Date attribute has a validation type of ISO Date and Time format, it does not match the format in the ETIM file, so it displays as an error in the Result table. Select the **ETIM Date=ARTCLASSVERSIONDATE** column and click the **Transform** button.

Import Manager

Steps

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

Map Data

Source:

ARTCLASSID	ARTGROUPID	ARCLASSDESC	ARTCLASSVERSI...	ARTCLASSVERSI...
EC000001	EG000017	Busbar terminal	5,00	18.5.2011 00:00:00
EC000003	EG000020	Residual current cir...	5,00	18.5.2011 00:00:00

Result:

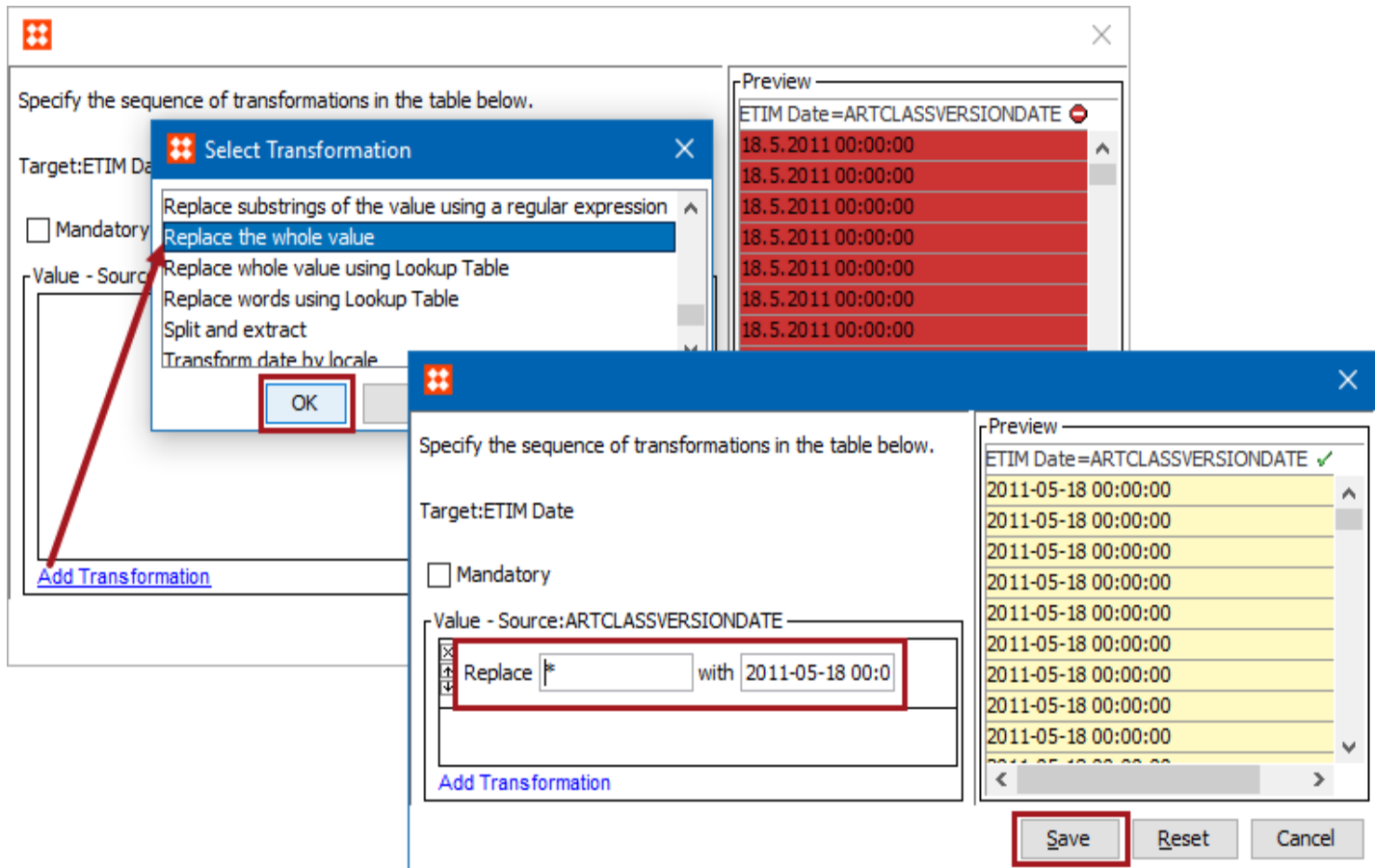
Map to: Classification

ID=(ARTCLASSID) ✓	ETIM Version=(ARTCLASSVERSION) ✓	ETIM Date=ARTCLASSVERSIONDATE ✗
ETIM EC000001	5	18.5.2011 00:00:00
ETIM EC000003	5	18.5.2011 00:00:00
ETIM EC000005	5	18.5.2011 00:00:00

Auto Map
Map
Constant
Remove
Transform
Generate Profile

Back
Next
Finish
Cancel

- In the Transformation dialog, click the **Add Transformation** link, select the **Replace the whole value** option, **replace * with 2011-05-18 00:00:00**, and click the **Save** button. This date is used since the date is the same for all ETIM Classifications.



19. All three (3) mapped columns now show valid values in the Result table. Click **Next** (selecting defaults) until the Import Manager Identify Destination step displays.

Import Manager

Steps

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

Map Data

Source:

ARTCLASSID	ARTGROUPID	ARCLASSDESC	ARTCLASSVERSI...	ARTCLASSVERSI...
EC000001	EG000017	Busbar terminal	5,00	18.5.2011 00:00:00
EC000003	EG000020	Residual current cir...	5,00	18.5.2011 00:00:00

Result:

ID=(ARTCLASSID) ✓	ETIM Version=(ARTCLASSVERSION) ✓	ETIM Date=(ARTCLASSVERSIONDATE) ✓
ETIM EC000001	5	2011-05-18 00:00:00
ETIM EC000003	5	2011-05-18 00:00:00
ETIM EC000005	5	2011-05-18 00:00:00

Map to: Classification

Auto Map Map Constant Remove Transform Generate Profile

Back **Next** Finish Cancel

20. Determine how to set the **Reject New** checkbox and click the **Finish** button:

- **Check** the checkbox if you imported only some ETIM classifications. This prevents the import of classifications which do not already exist in STEP. The execution report shows all objects rejected as a result of this option
- **Uncheck** the checkbox if you imported all ETIM classifications.

Import Manager

Steps

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

Identify Destination

Approver: User J (USERJ) ...

Import Workspace: Main

Default Parent: ...

Default Object Type: All Suppliers

Batch Directory: (None) ... Reset

Test Only Import ☐

Reject New ☐

Reject Updates ☐

Back **Next** Finish Cancel

21. Follow the steps to start the import as described in **Running a Data Import**.
22. When the import completes successfully, the transformed values are displayed for all ETIM Article Classification object types for the attributes ETIM Date and ETIM Version.

Tree

- ETIM 5 Article Groups
 - Distribution boards
 - Accessories for small distributio
 - Built-in field voor telecommunic
 - Cable connection box for entry
 - Cable entry cabinet
 - Cable tree
 - Connection cable tree for distr
 - Cover for distribution board
 - Cover strip for meter-/distribut
 - Empty cabinet
 - Energy distributor for construc
 - Gland plate for small distributio
 - Panel for distribution board
 - Separation plate for meter-/dis
 - Small distribution board
 - Small distribution board equip
 - Unequipped meter cabinet
 - Wiring set for small distribution
 - Sensors
 - Terminal blocks
 - Tools (Press, cut and isolate)

Cable tree rev.0.2 - Classific

Classification	Sub Products	References	Referenced By	Images
Description				
Name	>	>	Value	
ID			ETIM EC000273	
Name			Cable tree	
Object Type			ETIM Article Classification Type	
Revision			0.2 Last edited by USERJ on Tue Jan 03 12	
Approved			✗ Never Been Approved	
Translation			Not Translated	
Path			Classification 1 root/ETIM 5 Article Groups/	
Visibility				
ETIM Date			2011-05-18 00:00:00	
ETIM Description	abc		Cable tree	
ETIM ID	abc		EC000273	
ETIM Synonyms	abc		Wiring set Cable tree Cable harness	
ETIM Version	abc		5	

Excel Format

The following sample Excel data import file shows that the first row is a header, and the remaining rows are import data. Exported Excel files always include a header row, though this is optional for imports.

	A	B	C	D	E
1	<Name>	<Parent ID>	Primary Color	ClassificationID	DescriptionL
2	Mens T PBO w Class	18209	Black	22586	Comfortable fitting 100% cotton tee.
3	Mens T PBG w Class	18209	Blue	22586	Comfortable fitting 100% cotton tee.
4	Mens T PGS w Class	18209	Green	22586	Comfortable fitting tee of cotton/poly blend.
5	Womens T PGW w Class	18209	Gray	22584	Comfortable fitting tee of cotton/poly blend.
6	Womens T POY w Class	18209	Orange	22584	Comfortable fitting 100% cotton tee.

The available versions for Excel within STEP are:

- Excel 2003 - limit 65,536 rows
- Excel 2007 - limit 1,048,576 rows

It is recommended that users use the newest supportable version of Excel. File errors may occur during different STEP processes if using a template set to an older Excel version. For more information on supported versions of Excel, see the current **Platform and Software Support** documentation in the **System Release and Patch Notes**.

Keep the following points in mind when working with STEP data using the Excel format:

- The following node types / super types can be imported and exported via Excel format: products, classifications, entities, assets (objects, not content), and attributes (objects and definitions).
- A Excel import or export file will include data in the same arrangement as relational database tables. This means that each object is displayed as a single row in the file and each object property item is displayed as a single column.
- Imports and exports are context and workspace specific. By default, data is imported to or extracted from the context and workspace in use when the process is started.
- When planning to import data back into STEP, include STEP ID in the export.

Format Availability

Excel is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**
- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

Inbound Data

Excel import allows creation of and updates to products, classifications, entities, attribute values and references. However, system setup objects (for example, attributes, LOVs, users, reference types, and so on), cannot be created via import.

Because the Map Data process allows selection of only a single node type, only one node type / super type (products, entities, etc.) can be imported at a time. When multiple super types exist in the same import file, a separate import is required to successfully import each type of object, starting with classification data, then product data, and finally, entity data. When the inbound file includes data for node types other than the one selected, two things may happen: 1) assuming none of the data prevents the import, new objects are created using the supplied information and the selected super type, 2) the execution report details the skipped records when included data, like parent ID, is not found in the selected super type hierarchy. Alternately, split the inbound data file by super type and process accordingly.

Note the following specifications about importing Excel data:

- Unicode is used as the character set for importing Excel data files.
- Formulas are not allowed in the XLS or XLSX file. Data columns that include calculations or formulas must be loaded as fixed values.
- When loading data that include special characters (such as trademark [™], registered trademark ®, or mathematical symbols), ensure that the load file conforms to the Unicode character set.
- Ensure that data originally entered as a fraction (e.g., 3/16) was not converted to a date by Excel.
- All Excel cells must be set to 'Text' format.
- Remove any formatting applied to numeric values in Excel. This allows the Excel value to be entered into STEP, not the value that Excel transforms it into for display purposes.
- To remove an attribute value, use the entry **[delete]** in your load file. Empty cells do not cause values to be deleted. If the value being deleted was inherited, the result is not a blank field, but the inherited value is restored.
- Enter attribute values and units in the same cell, with the units following the values. STEP separates the values from the units and validates them individually. If your attribute values and units are in separate columns, you can either use the concatenation method in Excel or use a database program to combine them.

Also note when using Excel to import a reference type that allows multiple references, several elements affect the data added to STEP, as specified in the **Importing Multi-Valued References or Links with Excel** topic within this guide.

Inbound Parameters

The following parameters are available in both Import Manager and IIEP.

- **Conversion Preview** - a sample of the first few lines of the file is displayed to allow verification that the selected options are correct.
- **Has Header** - select if the file has a header line. If the first line (row) of the Excel file has header information, such as attribute names that match STEP, the Auto Map feature is available to map the columns of data to the appropriate object in STEP. Although a header row is not required, without it the user must be able to identify the data for manual mapping.
- **Trim whitespace** - select 'yes' to remove leading and trailing spaces in values, or select 'no' to leave them.

Import Manager

Import Manager

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

Select Format

Format
Excel

Converter for files in Microsoft Excel format that contain one object per row.

Has Header
yes

Trim whitespace
yes

Conversion Preview:

<Name>	<Parent ID>	Primary Color	ClassificationID
Mens T PBO w Class	18209	Black	22586
Mens T PBG w Class	18209	Blue	22586
Mens T PGS w Class	18209	Green	22586
Womens T PGW w Class	18209	Gray	22584
Womens T POY w Class	18209	Orange	22584

Back
Next
Finish
Cancel

IIEP

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
- 5. Configure Processing Engine**
 - 5.1. Select Sample File
 - 5.2. Select Data Source
 - 5.3. Select Format**
 - 5.4. Map Data
 - 5.5. Identify Objects
 - 5.6. Identify Destination
 - 5.7. Select Business Rules
 - 5.8. Advanced Settings
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Processing Engine : Select Format

Format

Excel

Converter for files in Microsoft Excel format that contain one object per row.

Has Header

yes

Trim whitespace

yes

Conversion Preview:

<Name>	<Parent ID>	Primary Color	ClassificationID	DescriptionL
Mens T PBO w Class	18209	Black	22586	Comfortable
Mens T PBG w Class	18209	Blue	22586	Comfortable
Mens T PGS w Class	18209	Green	22586	Comfortable
Womens T PGW w Class	18209	Gray	22584	Comfortable
Womens T POY w Class	18209	Orange	22584	Comfortable

Back

Next

Finish

Cancel

Outbound Data

When data leaves STEP via Excel format, the UTF-8 character set is used to export the data.

Note: When exporting object type, the default selection on the mapping step is 'Object Type Name'. However, when importing the same file, the auto-mapping feature expects the 'Object Type ID'. To generate an export file that is prepared for reimport, use the ID Aspect to map object type. For more information, see the **Aspect - Transform Outbound** section of the **Data Mapping** documentation.

Export Manager

Export Manager

Steps

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

Select Format

Excel

Exports data in Excel format.

Excel version: Excel 97/2003, Excel 97/2003, Excel 2007

Back Next Finish Cancel

OIEP

System Setup

- Inbound Integration Endpoints
- Match Codes and Matching Algorithms
- Outbound Integration Endpoints
 - Buy-Side Sell-Side, Advanced STEP
 - Buy-Side Sell-Side, Generic XML (fi
 - Buy-Side Sell-Side and Packaging H
 - Buy-side Sell-side and Packaging H
 - CrossContextAdvancedSTEPXML
 - DimDepLOV
 - Downstream Data_Event-based
 - Downstream Data_Object-Based
 - Excel_OIEP**
 - GDSN Provider Sender
 - MongoDB
 - Outbound Endpoint
 - Packaging Hierarchy, Adv STEPXM
 - Packaging Hierarchy, Event Based
 - Packaging Hierarchy, Generic XML
 - Padkaging Hierarchy, STEPXML
 - Select Objects Endpoint
 - TemplateOutboundEvent

Excel_OIEP - Configuration

Event Triggering Definitions Background Processes Statistics Error Log Excerpts

Outbound Integration Endpoint

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor
Item (Create, Modify, Delete)	Excel (0 mappings)	None

Add configuration

Delivery Method

Select format

Format Mapping Advanced

Excel

Exports data in Excel format.

Excel version: Excel 2007, Excel 97/2003, Excel 2007

OK Cancel

Importing Multi-Valued References or Links with Excel

When using Excel to import a reference type that allows multiple references, it is important to keep in mind the following points to ensure the intended references are applied in STEP.

- As an Excel file is processed by the importer, each row is read with no awareness of the previous or following rows. This means that if the same object is listed in the import file multiple times, updates made by the first occurrence of the object may be overwritten by subsequent occurrences.
- The Import Manager 'Advanced Settings' Remove Options functionality allows you to control when data in STEP is replaced by the inbound data, or instead, is appended by the inbound data.
- The combination of the inbound file structure, the import data mapping, and the Remove Unmapped options determine the data that will be available in STEP after a successful import.

Possible options for the Excel file layout are: 1) listing the object to be imported multiple times within the file, 2) listing each object once, with multiple references in the same cell, or 3) listing each object once, with multiple references in individual cells on the same row. Each setup is described below.

Object listed multiple times, each with a single link

Consider the scenario shown below where in STEP the product with ID 121933 is already linked to three classifications (ID BAProducts, ID SAProducts, and ID SASProducts) using the product to classification link type Supplier Link. This link type has the 'Allow multiple links' option enabled.

The screenshot shows the STEP interface. On the left is a 'Tree' view showing a hierarchy of products and suppliers. On the right is a 'References' table for product 121933. The table has columns for 'Reference Type' and 'Target'. Three rows are listed under 'Supplier Link', each pointing to a different classification: 'Suppliers/Supplies All/BA Products', 'Suppliers/Supplies All/SAProducts', and 'Suppliers/Supplies All/SASProducts'. A red box highlights the 'Supplier Link' column and the three rows.

An Excel import file has been prepared and includes two different valid classifications (ID OMProducts and ID PGProducts) for the product to classification link type with the ID Supplier Link.

	A	B	C
1	<ID>	<Name>	Supplier Link Classification Reference
2	121933	121933	OMProducts
3	121933	121933	PGProducts

Using the Import Manager's Map Data functionality, automatically map all possible columns, and then manually map the Product Classification Link type by ID.

The screenshot shows the 'Import Manager' window with the 'Map Data' step selected. The 'Source' table contains two rows of data. The 'Result' table shows the first two columns mapped. A sub-dialog titled 'Map Supplier Link Classification Reference to' is open, showing options to map the 'Supplier Link Classification Reference' column to either 'ID', 'Name', 'Attribute', or 'Product Classification Link'. The 'Product Classification Link' option is selected, and the 'Classification ID Aspect' is set to 'ID'.

<ID>	<Name>	Supplier Link Classification Reference
121933	121933	OMProducts
121933	121933	PGProducts

ID = <ID> ✓	Name = <Name> ✓	Classification = Supplier Link Classification Reference ✓
121933	121933	OM Products
121933	121933	PG Products

The multi-value classification references are displayed and ready to import.

The screenshot shows the 'Import Manager' window with the 'Map Data' step selected. The 'Source' table contains two rows of data. The 'Result' table shows the first two columns mapped. The 'Map' button is highlighted, and the 'Classification ID Aspect' is set to 'ID'.

<ID>	<Name>	Supplier Link Classification Reference
121933	121933	OMProducts
121933	121933	PGProducts

ID = <ID> ✓	Name = <Name> ✓	Classification = Supplier Link Classification Reference ✓
121933	121933	OM Products
121933	121933	PG Products

Complete the import wizard steps, noting that the selection on the Advanced tab determines how existing data in STEP is affected.

Remove Un-Mapped Classification References Not Set (Default)

Import Manager

Steps

1. Select Configuration

2. Select Data Source

3. Select Format

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Advanced Settings

Remove Options

Remove Un-Mapped References

☐ Classification References

Product Reference Type

> Add Product Reference Type

Asset Reference Type

> Add Asset Reference Type

Entity Reference Type

> Add Entity Reference Type

Classification Product Link Type

> Add Classification Product Link Type

The imported classifications are added to the existing ones for the mapped product to classification link type.

121933 rev.0.2 - Referen

Product

References

Referenced By

Images & Documents

Commercial

Tables

Discontinued Product Maintenance

Display

Reference Type

>

Target

>

>

>

Supplier Link

+

Suppliers/Supplies All/BA Products

Suppliers/Supplies All/OM Products

Suppliers/Supplies All/PG Products

Suppliers/Supplies All/SAPProducts

Suppliers/Supplies All/SASProducts

Website Link

+

Web Sites/Acme Retail Web Site/Apparel/Women's Apparel

Setting the Remove Un-Mapped Classification References option indicates that after the import, STEP will include only the product to classification link types (based on ID) contained in the import file.

© Stibo Systems - Confidential - STEP Trailblazer 8.3

111

Remove Un-Mapped Classification References Set

Import Manager

Steps

1. Select Configuration

2. Select Data Source

3. Select Format

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Advanced Settings

Remove Options

Remove Un-Mapped References

☐ Classification References

Product Reference Type

> Add Product Reference Type

Asset Reference Type

> Add Asset Reference Type

Entity Reference Type

> Add Entity Reference Type

Classification Product Link Type

> **Supplier Link**

> Add Classification Product Link Type

The imported classifications replace the existing ones for the mapped product to classification link type. Note that since each row in the Excel file is processed individually, only the final link for the object remains after the import is completed.

121933 rev.0.2 - Referen					
Product	References	Referenced By	Images & Documents	Commercial	Tables
Display					
Reference Type	>	Target	>	>	
> Supplier Link	+	Suppliers/Supplies All/PG Products			X
> Website Link		Web Sites/Acme Retail Web Site/Apparel/Women's Apparel			X

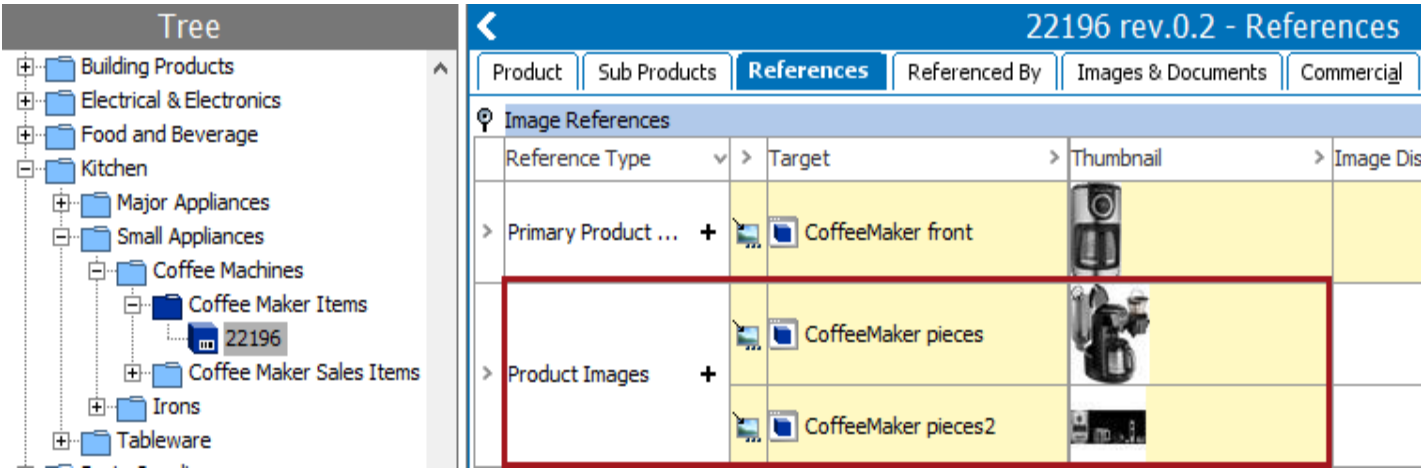
This result can be avoided by listing each object only once in the file, and either including all links of the same type within the same cell (as defined below) or by including each link in its own cell on the same row.

Object listed once, multiple references in the same cell

Consider the scenario shown below where in STEP a product already references two images (IDs CoffeeMaker pieces and CoffeeMaker pieces2) using the Product Images reference type. This reference type has the 'Allow multiple references' option enabled.

© Stibo Systems - Confidential - STEP Trailblazer 8.3

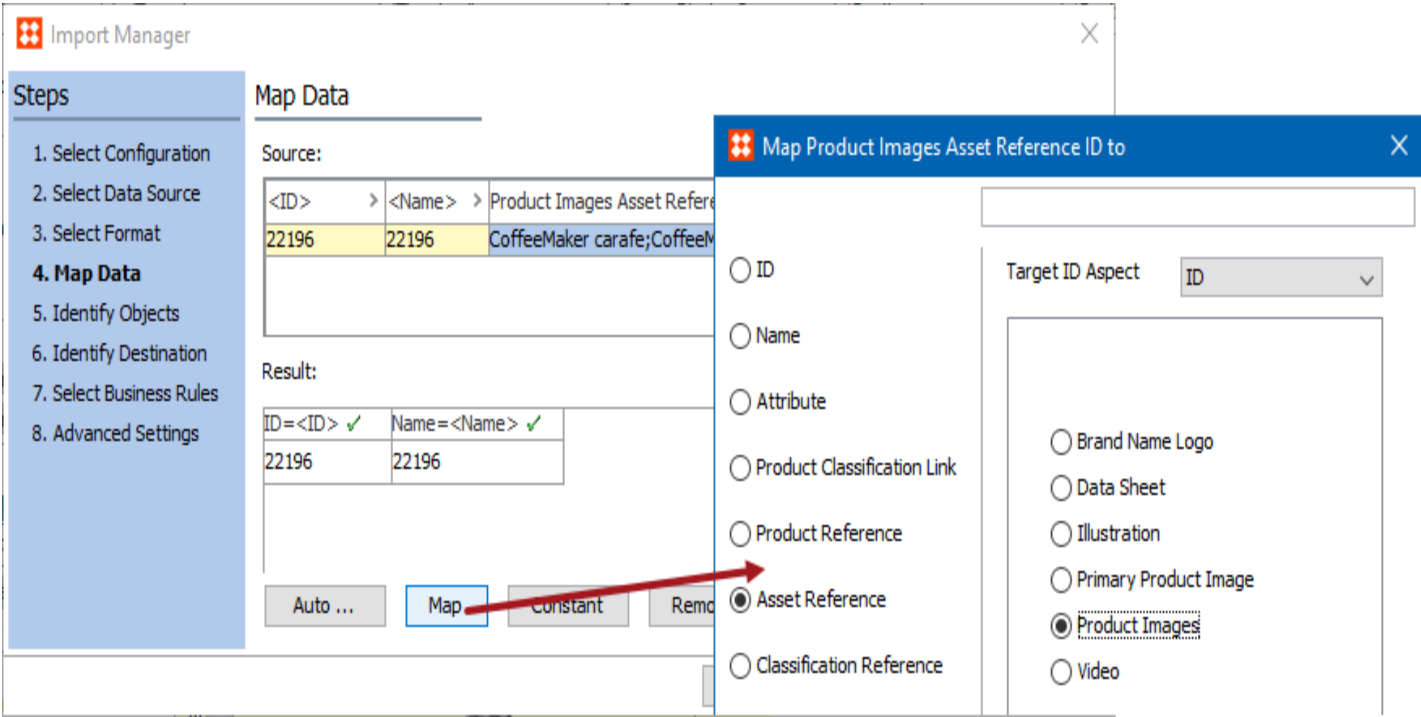
112



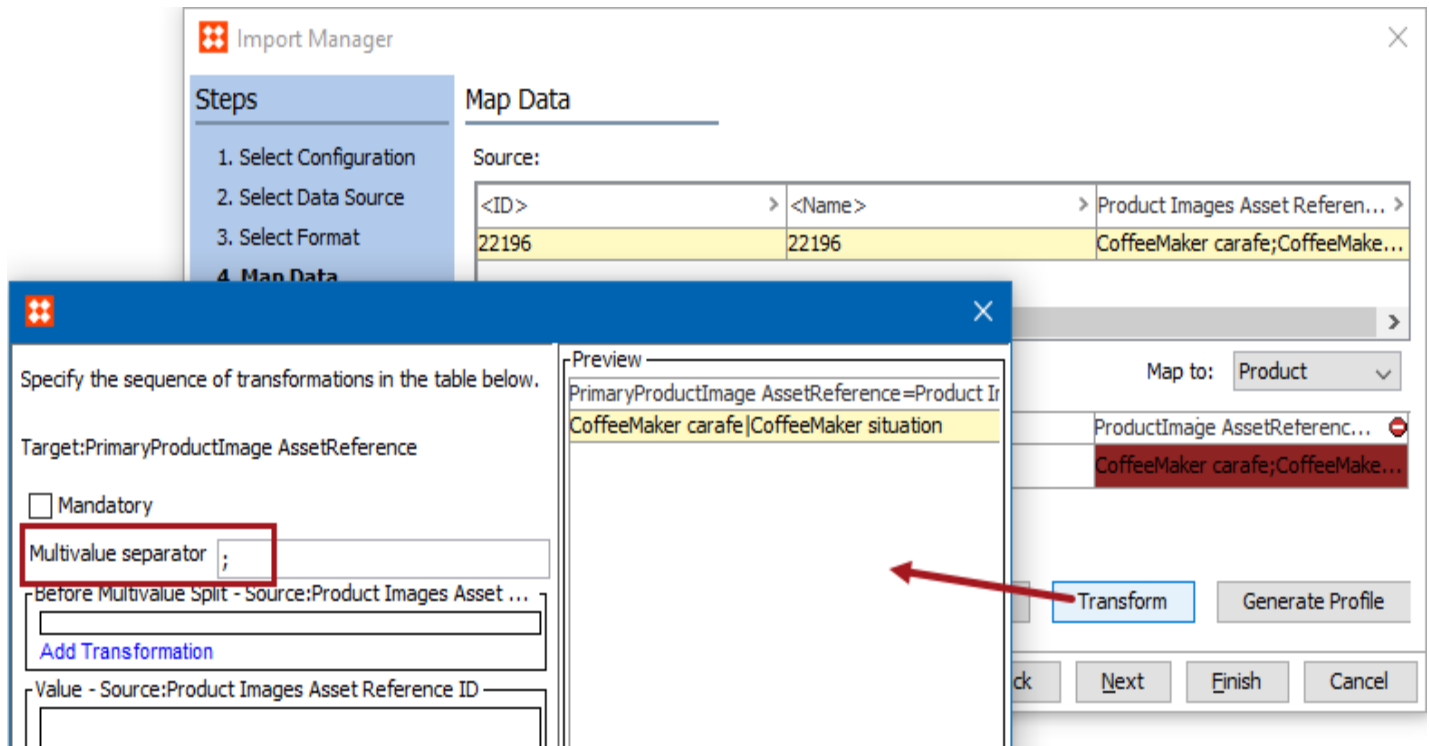
An Excel import file has been prepared and includes two different images (IDs CoffeeMaker carafe and CoffeeMaker situation) for the Product Images reference type.

	A	B	C
1	<ID>	<Name>	Product Images Asset Reference ID
2	22196	22196	CoffeeMaker carafe;CoffeeMaker situation

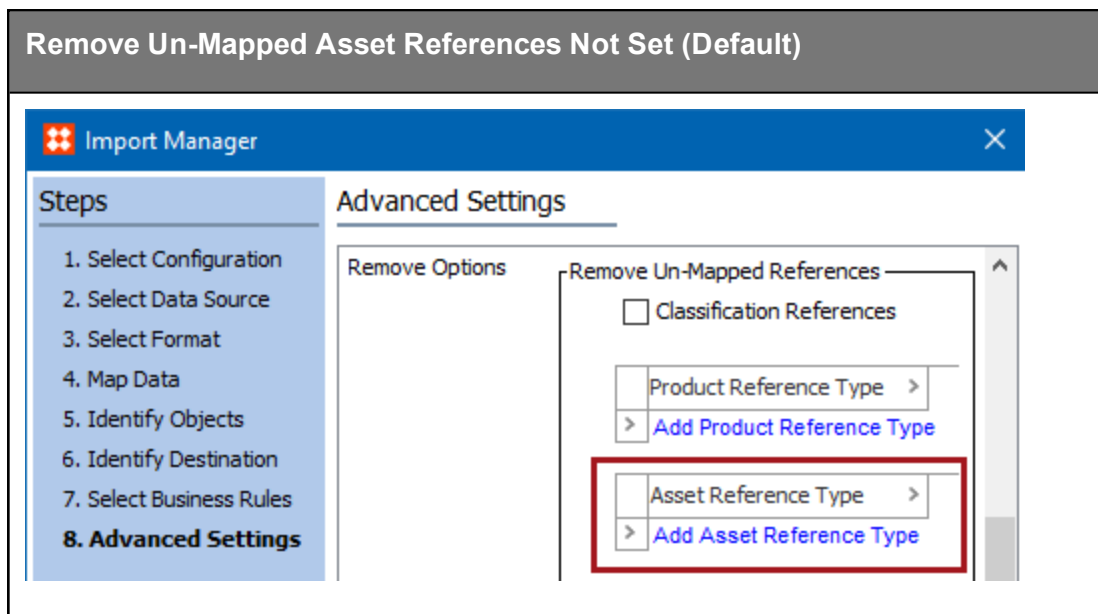
Using the Import Manager's Map Data functionality, automatically map all possible columns, and then manually map the new images to the Product Images reference type.



Finally, transform the multi-value references to an input format that STEP can read [replace each semicolon (;) with a pipe (|)], and verify that the Preview pane no longer shows a red background (which indicates an error).



Complete the import wizard steps, noting that the selection on the Advanced tab determines how existing data in STEP is affected.



Remove Un-Mapped Asset References Not Set (Default)

The imported images are added to the existing images for the selected reference type.

Reference Type	Target	Thumbnail
> Primary Product ...	CoffeeMaker front	
	CoffeeMaker carafe	
> Product Images	CoffeeMaker pieces	
	CoffeeMaker pieces2	
	CoffeeMaker situation	

Setting the Remove Un-Mapped References options indicates that after the import, STEP will include only the references contained in the import file.

Remove Un-Mapped Asset References Set to Asset Reference Type

Import Manager

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

Advanced Settings

Remove Options

Remove Un-Mapped References
☐ Classification References

Product Reference Type
Add Product Reference Type

Asset Reference Type
Product Images
Add Asset Reference Type

Remove Un-Mapped Asset References Set to Asset Reference Type

The imported images replace the existing images for the selected reference type.

<

22196

Product

Sub Products

References

Referenced By

Ima

Image References

Reference Type

Target

Thumbnail

> Primary Product ... +

CoffeeMaker front



> Product Images +

CoffeeMaker carafe



CoffeeMaker situation



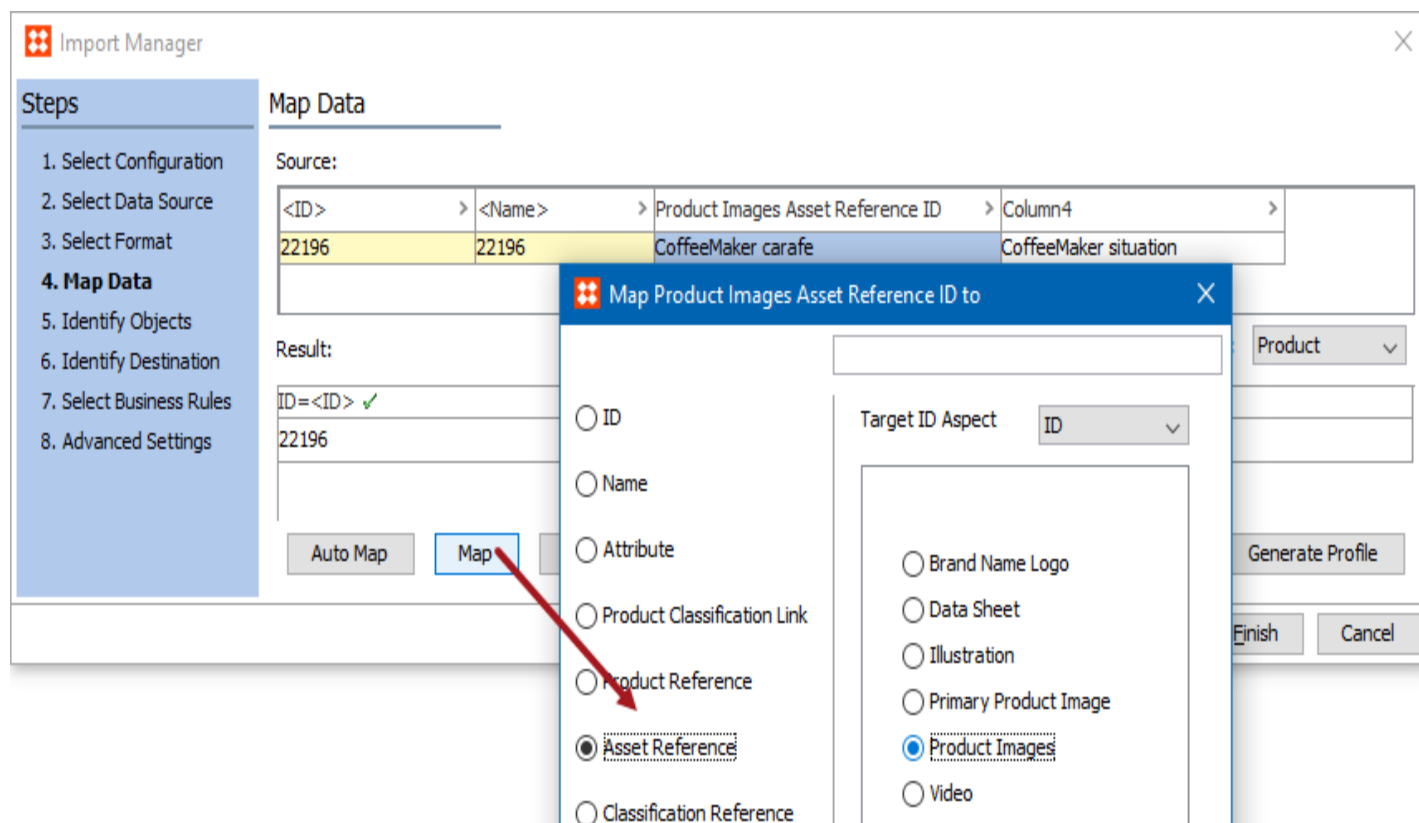
Item References, Classification

Object listed once, multiple references in individual cells

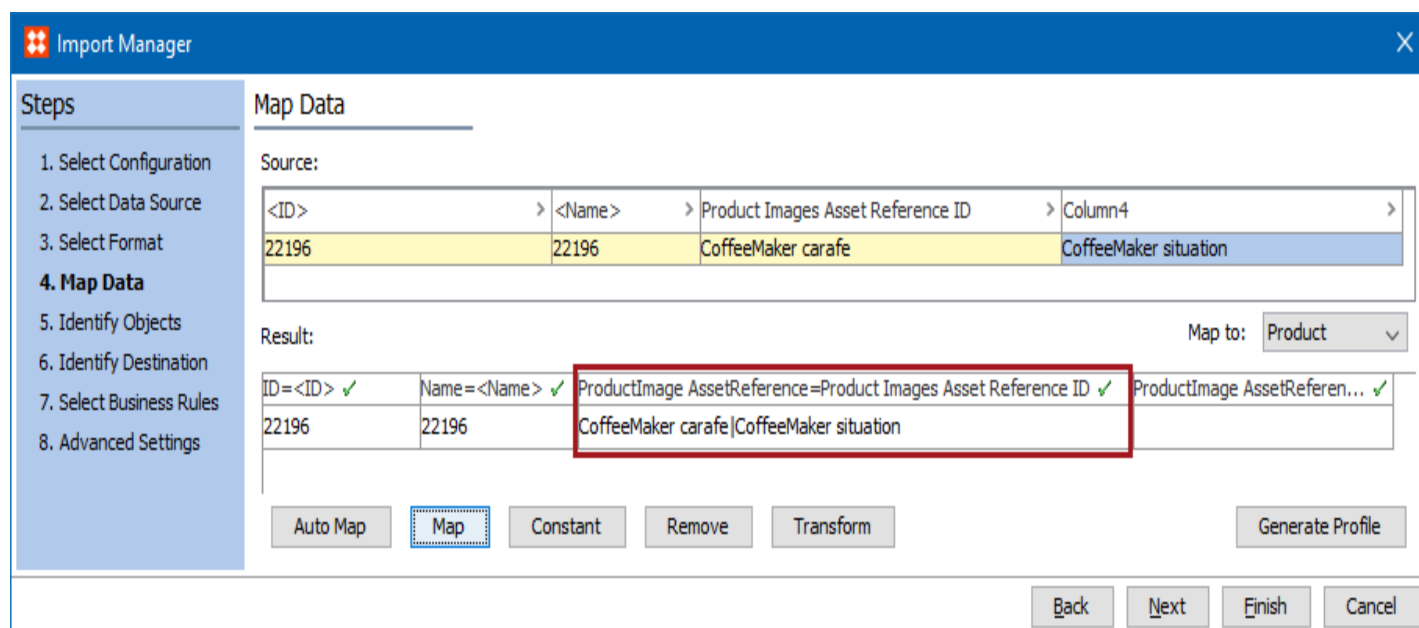
Consider the same scenario shown above where in STEP a product already references two images (IDs CoffeeMaker pieces and CoffeeMaker pieces2) using the Product Images reference type. This reference type has the 'Allow multiple references' option enabled. An Excel import file has been prepared and includes two different images (IDs CoffeeMaker carafe and CoffeeMaker situation) for the Product Images reference type.

	A	B	C	D
1	<ID>	<Name>	Product Images Asset Reference ID	
2	22196	22196	CoffeeMaker carafe	CoffeeMaker situation

Using the Import Manager's Map Data functionality, automatically map all possible columns, and then manually map each column of the new images to the Product Images reference type.



Note that as each individual cell is mapped to the same asset reference type, the values are added to the first cell in an input format that STEP can read [multi-values separated by a pipe (|)] in the same manner as the data was handled by the transformation in the previous example.



Complete the import wizard steps, noting that the selection on the Advanced tab determines how existing data in STEP is affected. The results of this import are identical as those in the previous example.

Excel Custom Template

The Excel Custom Template is an export component that allows users to quickly export one or multiple objects into a polished Excel output. This is done by adding placeholder tags into an Excel template for the attribute values that should be exported. The placeholder tags are extracted from the template and STEP data is mapped to those placeholder tags in the template. In this way, presentation-ready Excel files can be exported. The example below depicts a product fact sheet, one possible output of the Excel custom template.

The versions of Excel that are available in STEP are:

- Excel 2003
- Excel 2007

It is recommended that the newest supportable version of Excel is used. File errors may occur during different STEP processes if using a template set to an older Excel version. For more information on supported versions of Excel, see the current **Platform and Software Support** documentation in the **System Release and Patch Notes**.

Format Availability

Excel Custom Template is available for selection in:

- Export Manager - see **Creating a Data Export**

Mapping

This format requires creating a data map between STEP and the data being exported, and may also include data transformations. For details, see **Data Mapping**.

Output

The Excel Custom Template component is intended to produce Excel documents that are ready to present. Below is one example of how an Excel output might look.



Flusdale FLU12 105 2017 Road Bike

1216 €

This is for the core, the local heroes, the crit assassins, the working-man racers with chiseled legs and dreams of glory, because they know that when it comes to pure performance for the buck, nothing can touch FLU12. Lighter, stiffer and smoother than most carbon frames, it's not just the finest alloy racing bike ever made, it's the only choice for those who know.

Lorem ipsum dolor sit amet, proin potenti in a pellentesque at eu, eget elit amet ut turpis excepturi risus, porta sollicitudin felis phasellus wisi purus. Ut wisi tellus bibendum quis in, urna praesent ipsum eu placerat, pulvinar minima vestibulum leo augue elementum, nisl odio ad at condimentum aspernatur arcu, ut fermentum eget. Integer adipiscing fusce, suscipit in, nam pellentesque risus, nulla dictum fermentum auctor, nulla nulla est in. Nulla a morbi, urna purus etiam mauris dolor. Adipiscing nunc labore eros elit donec, lorem varius amet nisl magna rutrum odio,

Product Details

Frame:	FLU12, SmartForm C16069 Alloy, SPEED SAVE, BB30a, Di2 Ready	Brakeset:	Shimano 105 5800 caliper
Fork:	FLU12, SPEED SAVE, BallisTec full carbon, 1-1/8" to 1-1/4" steerer, integrated crown race	Handlebar:	Flusdale C3, butted 6061 Alloy Butted 6061 Alloy, Compact
Front Derailleur:	Shimano 105, braze-on	Stem:	Flusdale C3, 6061 Alloy, 31.8, 6 deg
Rear Derailleur:	Shimano 105	Bar Tape:	Flusdale Bar Tape w/Gel 2.5mm
Number of Gears	22	Wheelset:	Mavic Aksium WTS

Excel Custom Template Configuration

The Excel Custom Template is an export component that allows users to quickly export one or multiple objects into a polished Excel output.

Configuration

To configure an export of objects using the Excel Custom Template, follow these steps:

1. **Create Excel template with placeholder tags** – Users may set up a custom Excel template to fit a wide array of requirements. A template can include static elements, like headers or non-STEP images, that will display unchanged in the final export. To export attribute values from STEP into the export file, users must add placeholder tags to the template. A placeholder tag consists of any text appearing between a greater than (<) and less than (>) symbol. As long as the attribute is properly mapped in the Export Manager (addressed later in this same topic) and is both valid and instantiated for that object, the attribute value will be pulled into the exported Excel file.

STEP mages may also be brought in, but the placeholder tag for images must be formatted differently.

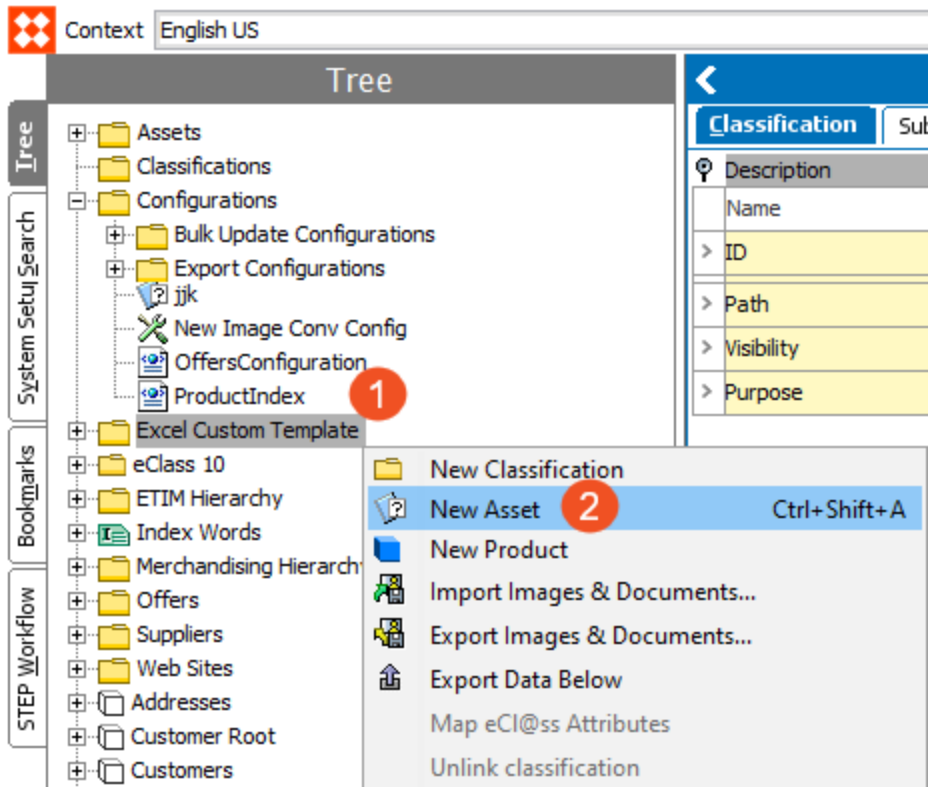
What follows is an example of a correctly formatted image tag: **<img:ProductImage_10x16>**.

Just as with attribute placeholder tags, image placeholder tags must begin and end with the greater than (<) and less than (>) symbols. Inside of these tags, the image placeholder tags must begin with 'img:'. Any text can be used following the colon, but best practice is to insert text relevant to the kind of image that should be exported. If the image will always be, for example, the primary product image, then the image tag text might be 'PrimaryProductImage.' It is important to note that only PNG, JPG, and JPEG image file types are allowed using the Excel Custom Template export format. Following the text is an underscore (_) followed by two numbers on either side of an 'x.' These numbers are dimensions, and dictate how many Excel cells the exported image will take up, the first determining the width (columns) of the image, and the second the height (rows). Below is an example of a simple Excel Custom template that will produce a simple Excel export file:

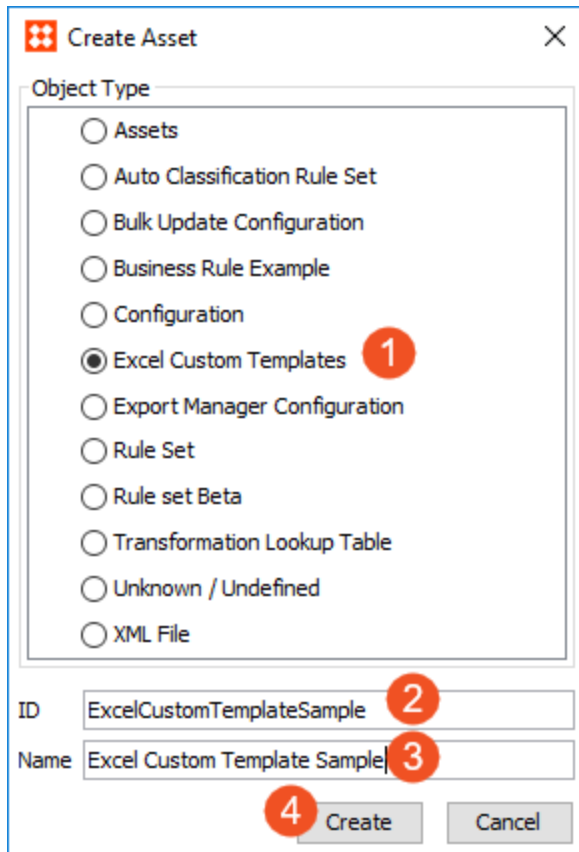
	A	B	C	D	E	F	G	H	I
1	Stibo Tshirt Shop								
2									
3	<Name>				<Primary Color>				
4	<img:ProductImage_3x8>								
5									
6									
7									
8									
9									
10									
11									
12									
13	Product Details								
14	Sell Pack Quantity:		<Sell Pack Qty>						
15	Weight:		<Weight>						
16									

Sheet1

2. **Create a Custom Excel Template asset type** – In the System Setup tab in workbench, expand the Object Types & Structures node and select the 'Assets' node. Right-click 'Assets' and select 'New Object Type.' Assign an ID and name to a new object type ('Custom Excel Templates' would be appropriate), and click 'Create.'
3. **Save the template as an asset in STEP** – The Excel custom templates are stored in STEP as assets. During export of objects using the Excel Custom Template export format, users will navigate to the appropriate template and select it for the export. To save the template as an asset, users may elect to create an asset root node and title it 'Excel Custom Templates,' as shown in the screenshot below. To save the template as an asset, select the root asset in which the template will be saved, right-click, and select 'New Asset.'



In the 'Create Asset' dialog, select the asset object type created in step two, in this case it is 'Excel Custom Templates,' and assign an ID and name to the template being saved.



Create Asset

Object Type

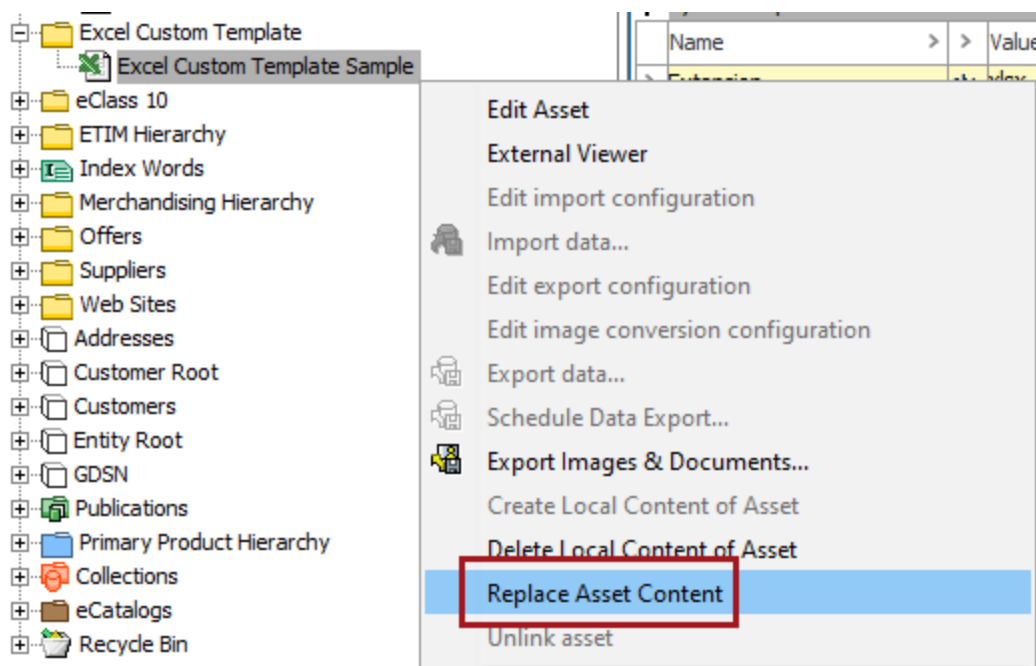
- ☐ Assets
- ☐ Auto Classification Rule Set
- ☐ Bulk Update Configuration
- ☐ Business Rule Example
- ☐ Configuration
- ☒ Excel Custom Templates **1**
- ☐ Export Manager Configuration
- ☐ Rule Set
- ☐ Rule set Beta
- ☐ Transformation Lookup Table
- ☐ Unknown / Undefined
- ☐ XML File

ID: ExcelCustomTemplateSample **2**

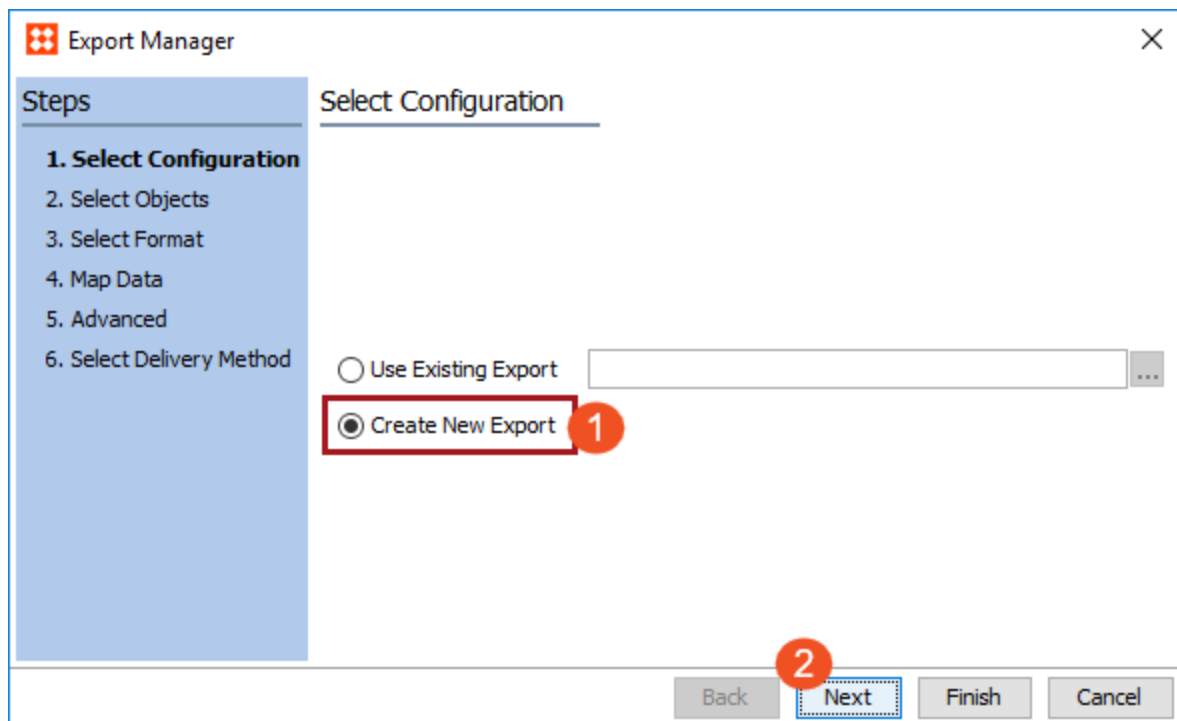
Name: Excel Custom Template Sample **3**

4 Create Cancel

Be sure to also upload the template to the asset by right-clicking the asset and selecting 'Replace Asset Content.' STEP will prompt the user to select a file from the user's system to upload to the asset object.



4. **Create an export configuration asset** – In order to retain the mappings that users establish in the Export Manager during configuration of their custom Excel export, it is useful to create and save an export configuration asset into which the export configuration can be saved. To create an export configuration asset, navigate to the 'Asset' root node, select it, right-click and select 'Add Asset' from the menu. Select the 'Export Manager Configuration' and then assign an ID and name to the asset.
5. **Export object(s)** – To access the Export Manager and generate an Excel export file from the custom template, select 'File' from the toolbar, then 'Export,' and then 'Data...'. The user will map the placeholder tags to STEP data, and then execute the export. For more information on the Export Manager functionality, see the **Export Manager** section of the Data Exchange documentation.
 - **Select Configuration** – Select 'Create New Export,' then 'Next.'



- **Select Objects** – Select one or more object(s). You may select multiple objects to export using a custom Excel template, but each object and its data will go into its own Excel file and then gathered into a .zip file, rather than all objects and their data going into a single Excel file.


Export Manager

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

Select Objects

ID	Name	Object Type	Path
> 124148	18213 M O	Item	Primary Product Hierarchy/Product...
> Add Objects			

Only export selected objects ☐
Only export leaf objects ☒
Export: Product

Include object types <All object types>

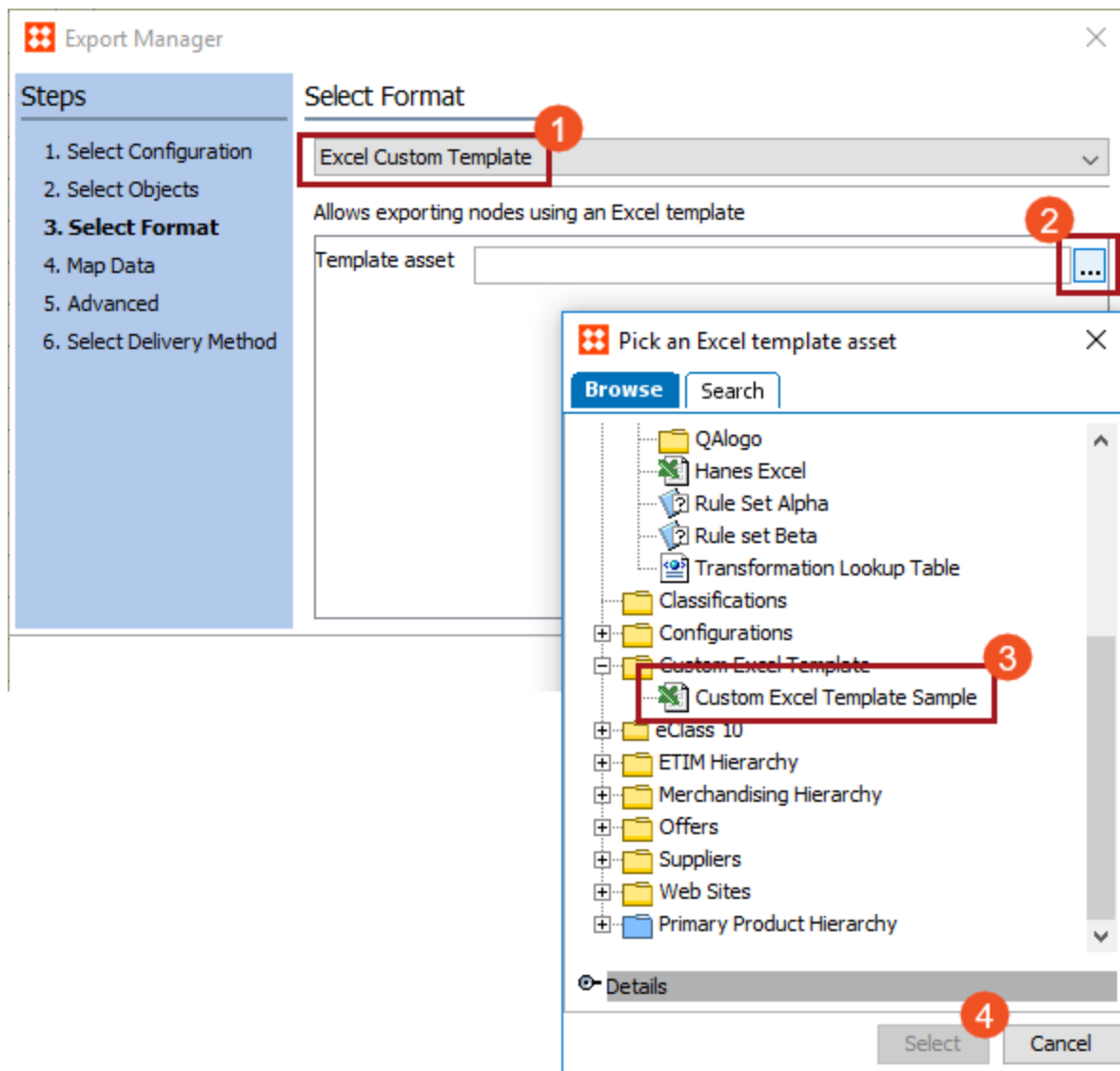
Back

Next

Finish

Cancel

- **Select Format** – Select the 'Excel Custom Template' export format from the dropdown, and then, in the 'Pick an Excel template asset' dialog, select the Excel Custom template asset that was saved in step three.



- **Map Data** – In the 'Map Data' screen, users will map the placeholder tags inserted into the custom Excel template to STEP attribute values. Listed in the right panel (mapping targets) are the placeholder tags that STEP read from the Excel template. In the left panel are the data sources. The placeholder tags must be mapped to a data source from the left panel for the data to export correctly into an Excel file. Map all targets in the right panel. When that is complete, click 'Next.'

For more information on mapping data, see the **Outbound Map Data Options** topics (Data Source and Mapping Targets).

Map Data

<ID>
<Name>
<Parent ID>
<Object Type Name>
<Product-Override Child ID>
<Is deleted>
"Constant Value"
<Page Number>
+ All Attributes
Select Attribute
+ Classifications
+ Index Words
+ Product Classification Links
+ Product References
+ Asset References
+ Classification References
+ Entity References
+ STEP Workflow Task Info
Multi level References
Multi level Parent attributes
Insert Referenced Objects
+ Custom Attributes
+ System Setup

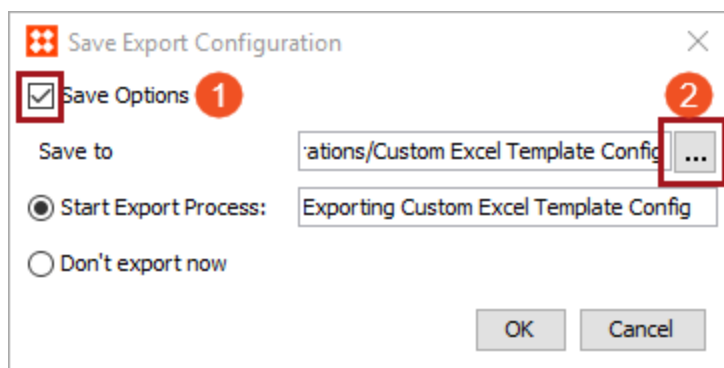
Allows exporting nodes using an Excel template

Name * Nothing mapped
Primary Color * Nothing mapped
ProductImage * Nothing mapped
Sell Pack Qty * Nothing mapped
Weight * Nothing mapped

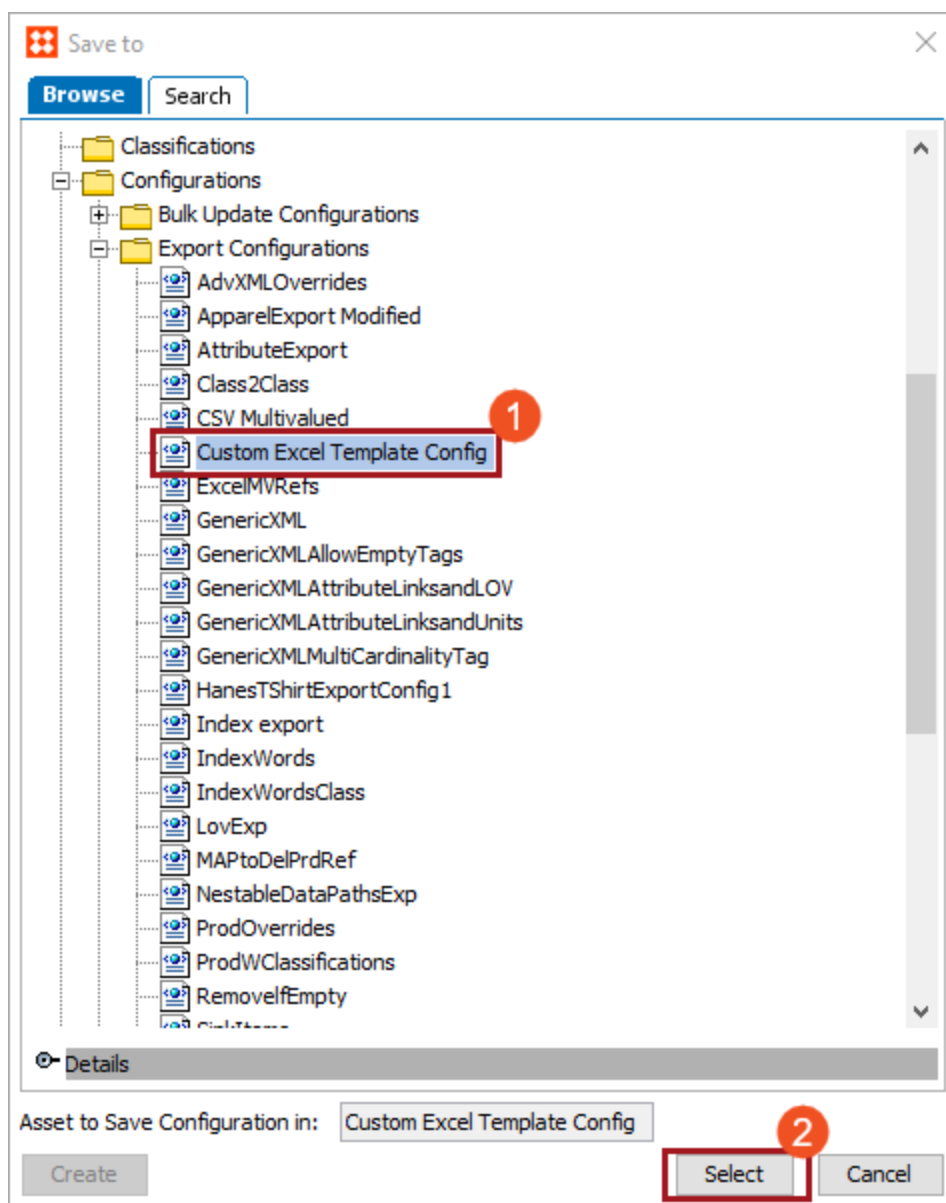
Inherit Data and References ☒
* = Mandatory field Red = Unmapped mandatory field

Back Next Finish Cancel

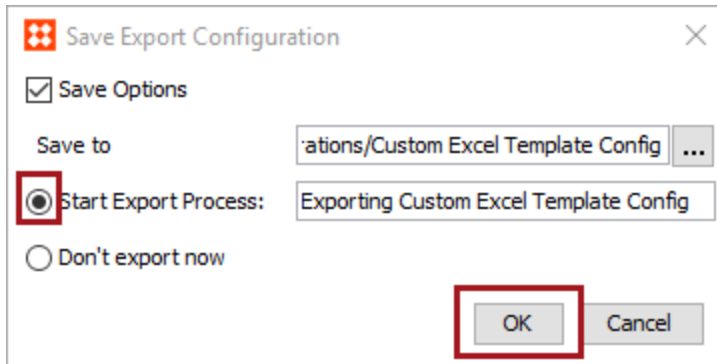
- **Advanced** – Make default selections, if applicable.
- **Select delivery method** – Select delivery method for the Excel document. For this example, 'File' has been chosen. Then click 'Finish.' Single-object exports will export to a single Excel file; multiple-object exports will export to a zip file containing an Excel file for each exported object.
- **Generate and save the export configuration** – Before the export is executed, a 'Save Export Configuration' window displays. Users may leave the 'Save Options' box blank and execute the search as it has been configured. Doing this means the export can only be run once as configured. If an export of the object or objects using the custom template must be run again, the mappings established between the placeholder tags and the attributes and references must be re-established. This is why, if exports are to be run on a custom Excel template more than once, it is best practice to save the configuration as an export configuration asset. To do this, check the box for 'Save Options,' and then click the ellipses button for the 'Save to' parameter.



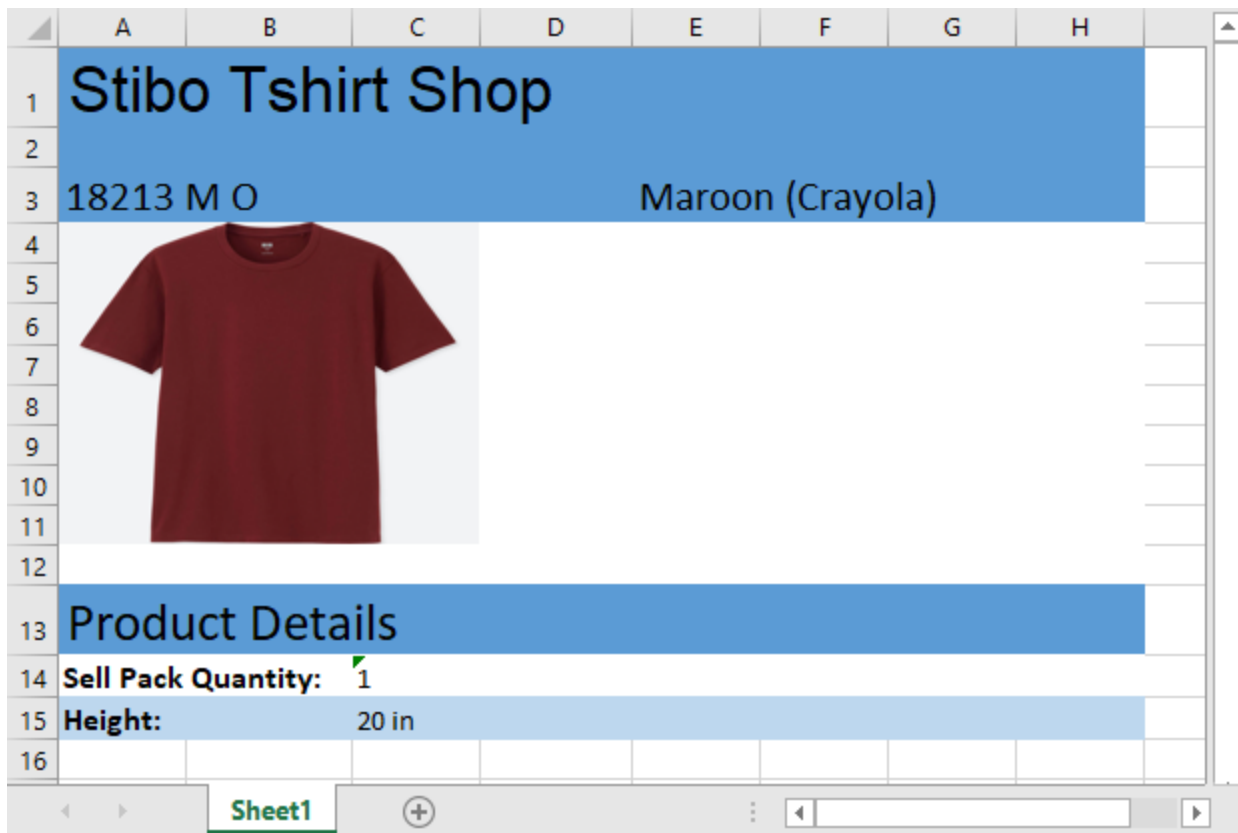
In the 'Save to' window that displays, navigate to the export configuration asset created in step four. Select the asset and click 'Select.'



Then, back in the 'Save Export Configuration' window, check 'Start Export Process' and then click 'OK.'



When the exported Excel file is opened (accessible in the completed background process), the result should look like this (this sample output comes from the template shown in the image accompanying step one in this topic):



It may be useful to keep the following points in mind when working with STEP data using the Excel Custom Template export format:

- The user can also access the Excel custom template export from a Task List or Node List screen in Web UI using the 'Export Action' toolbar action button.

- There is no import aspect to the Excel custom template. Custom templates can only be used to export STEP data.
- All values will be handled as text in Excel. (**Not** applicable to images.)
- Data can only be exported to standard Excel cells.

Excel Smartsheet Format

Smartsheets are a specialized Excel format with the ability to validate products before being imported into STEP. Typically, this format is used when maintaining product data or when initiating new products into STEP via the Supplier Web UI. If used to maintain product data, users can download a Smartsheet from the Supplier Web UI, and once the desired changes have been made, the data can be automatically validated in Excel. The Excel file can then be uploaded back into the Supplier Web UI. New products can also be initiated into STEP via predefined Smartsheet templates.

The data export and template configurations are created in the Export Manager, then added to the relevant template widgets and the export action in the Supplier Web UI. An import configuration should also be created and defined on the 'SmartSheetImportAction' in the Supplier Web UI (in case an import configuration is not specified in the export configuration).

Using Smartsheet format and Supplier Web UI may require special licenses. Ask your Stiibo account manager for more information.

Note: Though still available, the Quicksheet concept has been deprecated. The use of Smartsheets is recommended.

Format Availability

Smartsheets are designed to be used from the Supplier Web UI **only** and using them outside of this context will diminish their capabilities. Furthermore, future Smartsheet enhancements will most likely also depend on Web UI.

Important: Objects should **not** be exported to Excel Smartsheets from the STEP Workbench. Smartsheet templates exported from STEP Workbench will not be able to correct supplier linking of products created during import and will also provide different browsing capabilities. A maintenance Smartsheet exported from the STEP Workbench will not contain any information about workflow and state, so it will not be able to supply information about workflow mandatory columns. In addition, any business rules regarding the current workflow will be unavailable during import. It will also supply a different set of possible targets when maintaining product cross-references.

Setup Requirements

Before Smartsheets can be downloaded from the Supplier Web UI and/or imported into STEP, some initial configuration steps are required.

1. In Export Manager, create a Smartsheet data export and/or template configuration as described in **Smartsheet Data and Template Configurations**.
2. In Import Manager, create an import configuration as described in **Smartsheet Import Configurations**.
3. In the Supplier Web UI, configure the Smartsheet Import Action as described in the **Smartsheet Import Action** section of the **Web User Interfaces** documentation.
4. In the Supplier Web UI, configure the Smartsheet Export Action as described in the **Smartsheet Export Action** section of the **Web User Interfaces** documentation.

5. In the Supplier Web UI, configure the Smartsheet Export Widget as described in the **Smartsheet Export Widget** section of the **Web User Interfaces** documentation.

Additional Information

1. Review the available system configuration properties, the Smartsheet cover sheet tab, and Smartsheet special attribute handling, as described in the **Additional Information About Smartsheets** topic in this guide.
2. Review the available Web UI components for use with Smartsheets as described in the **Smartsheets in Web UI** topic of the **Web User Interfaces** documentation.
3. Understand the background processes and queues used with Smartsheets, see the **Smartsheet Background Processes and Queues** topic in this guide.
4. Alter the behavior of Smartsheets using configuration properties as described in the **Smartsheet System Properties** topic of this guide.
5. Review the available types of Smartsheet and the required handling as defined in **Using a Smartsheet** topic of this guide.
6. Understand how to initiate a product with a Smartsheet as defined in the **Initiating Products Using a Smartsheet** topic of this guide.
7. Understand how to maintain products with a Smartsheet as defined in the **Maintaining Products Using a Smartsheet** topic of this guide.

Initiating Products Using a Smartsheet

Smartsheets offer suppliers a simple and efficient means for initiating new products into STEP. The guidelines laid out below cover the general process.

Prerequisites

Several setup steps are required before Smartsheets can be used to initiate new products into STEP:

1. In Export Manager, create a Smartsheet data template configuration as described in **Smartsheet Data and Template Configurations**. This export configuration should be configured on the Smartsheet Export Widget so that the end user can select it when exporting a Smartsheet template from the Supplier Web UI.
2. In Import Manager, create an import configuration as described in **Smartsheet Import Configurations**. This import configuration should be specified in the accompanying template configuration and/or the Supplier Web UI Smartsheet Import Action. Without a configuration, a Smartsheet cannot be imported into the Supplier Web UI.
3. In the Supplier Web UI, configure the Smartsheet Import Action as described in the **Smartsheet Import Action** section of the **Web User Interfaces** documentation. This action is used to import the new product data.
4. In the Supplier Web UI, configure the Smartsheet Export Widget as described in the **Smartsheet Export Widget** section of the **Web User Interfaces** documentation. This widget is used to export Smartsheet templates from the Supplier Web UI.

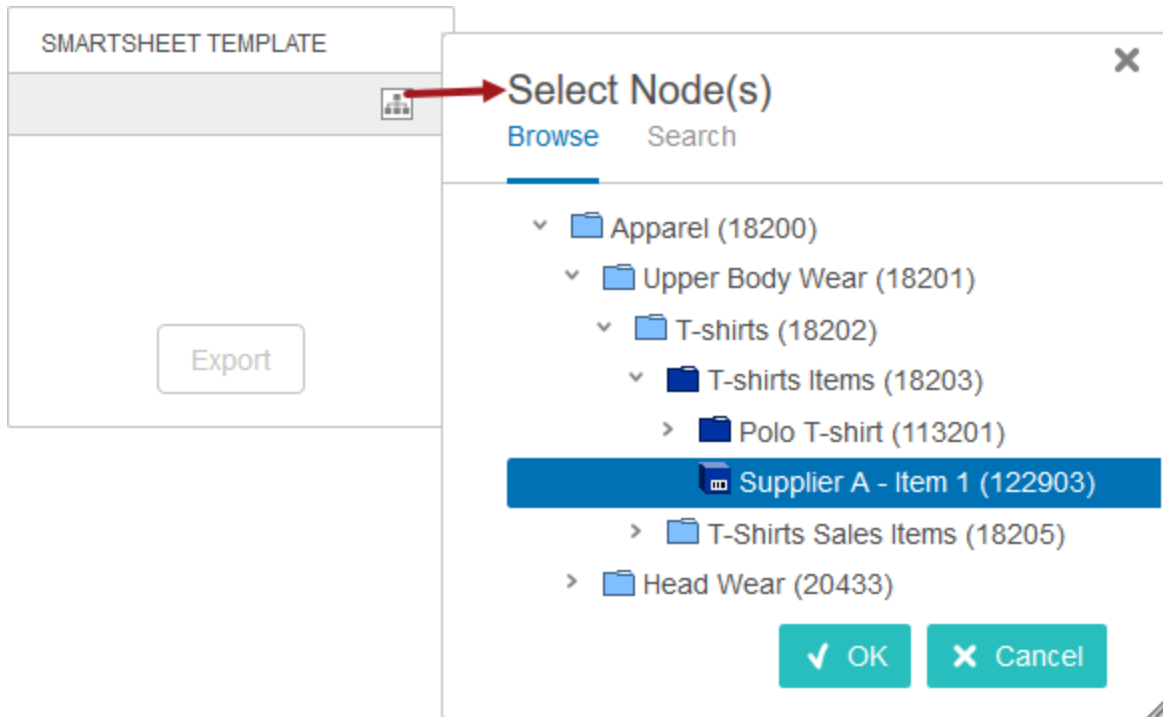
Product Initiation

New products can be initiated into STEP by: exporting a Smartsheet template from the Supplier Web UI, adding product data to the exported Excel file, and ultimately re-importing the edited Smartsheet back into the Supplier Web UI.

The Smartsheet Export widget allows export of a Smartsheet template, i.e. an empty Smartsheet that can be used to on-board new products. It allows you to start the download of the Multi-Level Smartsheet template, using an export configuration. Access is based on the user's supplier privileges, meaning that the component is only available if the user is a member of a supplier group.

Detailed steps for export are:

1. In the **Smartsheet Export** widget, click the browse hierarchy button to display the Select Node(s) dialog.



2. Select a valid node based on the object type(s) specified in the Smartsheet template and enable the OK button. An 'Invalid Selection' error is displayed and the OK button is not enabled when an invalid node is selected.
The category specific attributes that appear in the exported Smartsheet are dependent on the selected node and all applicable child nodes. No actual product data is exported.
3. Click the **Export** button to start the background process and export the template.
4. On the status dialog, click the link to the background process.



5. the Background Process Details page displays. Upon completion of the background process, click the link (for the file extension .xlsm) and choose to open the template.

Background Process Details	
ID	BGP_122923
Started By	SUPPLIERA
Description	Smartsheet template export
Template ID	Export Manager Pipeline
Status	✓ Succeeded Supplier_A__Item_1--2017-07-27--11-16.xlsm
Started	7/27/17 11:16:33 AM
Finished	7/27/17 11:16:49 AM
Elapsed	17 s

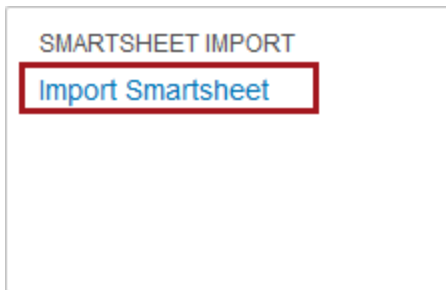
6. Enter product data into the exported Smartsheet.

	B	D	H	I
	Validate sheet	Next error		
	Duplicate row	Delete row		
9				
10	* <Name>	Brand Name	Country of Origin	
11	Red V-Neck	ACME	United States	...
12	Pink V-Neck	ACME	United States	...
13	Blue V-Neck	ACME	United States	...

For more information on navigating and editing a Smartsheet, see the **Using a Smartsheet** section of the documentation.

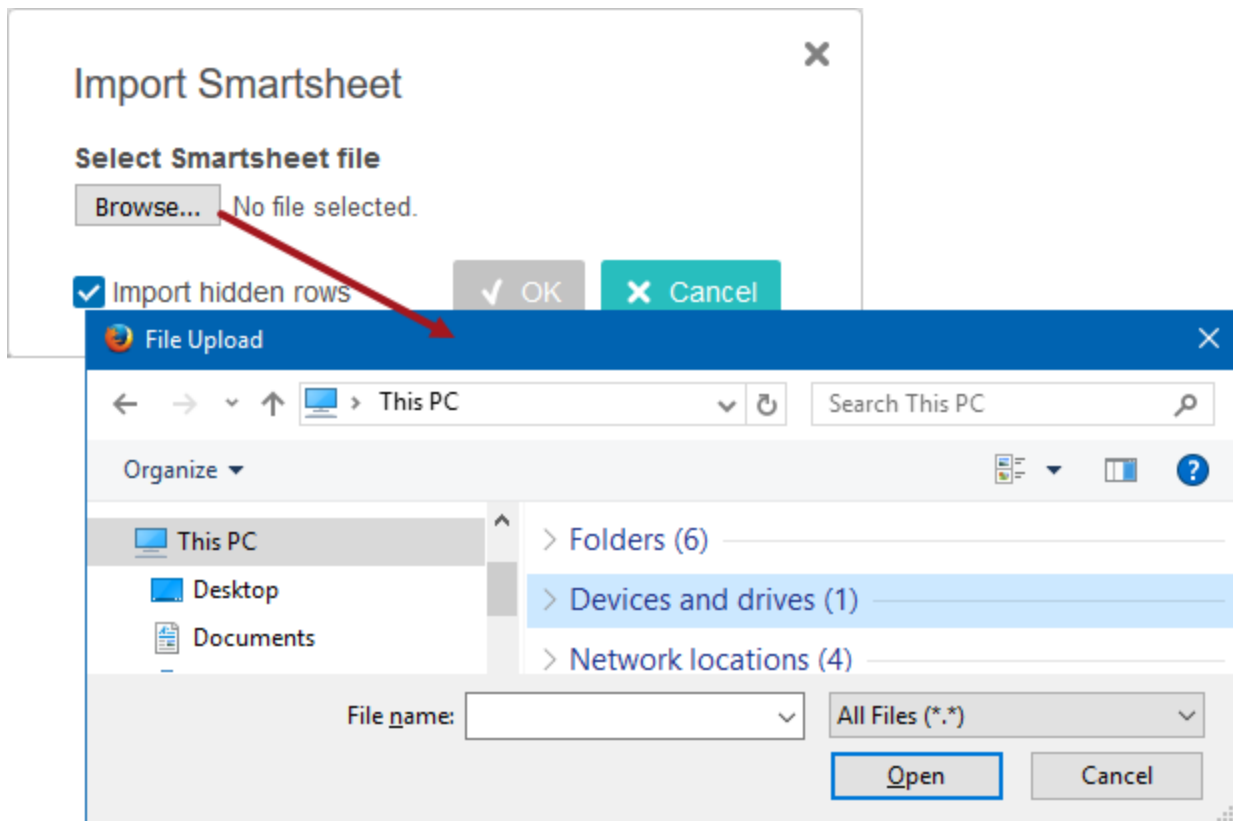
For information about the color coding of the cells, see the Cover tab, which is described in the **Additional Information About Smartsheets** topic.

- If required, validate the sheet by clicking the **Validate sheet** button and following the sheet validation process.
For more information on validating a Smartsheet, see the **Using a Smartsheet** section of the documentation.
- Once the sheet has been validated, save the document.
- Navigate back to the Supplier Web UI and click on the Smartsheet Import Action link.



10. In the Import Smartsheet dialog, click the **Browse** button to display a File Upload dialog. Navigate to the Smartsheet, select it, and then click **Open**.

Checking the 'Import hidden rows' checkbox will ensure that any hidden rows will be included in the import. If you opt not to import hidden rows, the background process will indicate how many hidden rows were skipped.



11. Click the newly enabled **OK** button to import the data via a background process.

Note: If errors occur during import, a Smartsheet file with the products that failed can be downloaded from the import process page. The error messages are displayed as screen tips in the Smartsheet file when hovering over an error cell. **Next Error** can be used to troubleshoot a set of errors. When the errors have been corrected, the file can be imported again. However, there is limited support for this type of re-import. Some cells will be locked and consequently skipped during import.

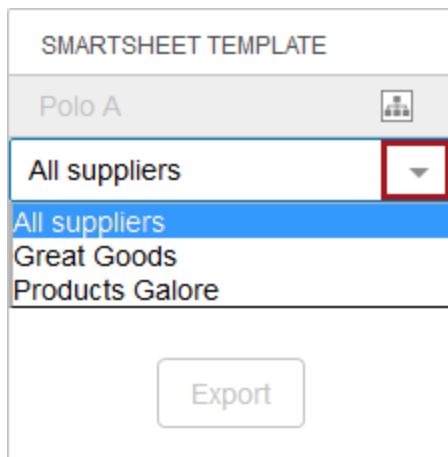
Exporting a Template with a Supplier Selector

Users that belong to multiple supplier groups can, within a single Smartsheet, load data for all suppliers to which they have access. Once configured, a supplier selector column will appear on the Smartsheet template. Users can select suppliers on a row-by-row basis, with each object being linked to the selected supplier upon import.

For information on how to configure a Smartsheet template with a supplier selector column, see the **Smartsheet Data and Template Configurations** section of the documentation.

Important: Before a Smartsheet can be exported with a supplier selector column, ensure that the 'Enable all-view for users that are a member of multiple suppliers' field is set to 'Y' in system properties.

1. In the **Smartsheet template** widget, click the browse hierarchy icon  and select a valid node.
2. If the user is a member of multiple supplier groups the supplier dropdown will appear underneath the node picker. If the user is only a member of a single supplier user group, the supplier dropdown will not be present.



The screenshot shows a 'SMARTSHEET TEMPLATE' widget. At the top, it says 'Polo A' next to a small grid icon. Below this is a dropdown menu currently showing 'All suppliers'. The dropdown is open, showing three options: 'All suppliers' (highlighted in blue), 'Great Goods', and 'Products Galore'. At the bottom of the widget is an 'Export' button.

Users can only make a selection from the supplier groups they belong to. Additionally, when selecting 'All suppliers', the supplier selector column will only include the supplier groups that the user belonged to at the time of the export.

If the user is not a member of multiple supplier groups or the user specifies a supplier using the dropdown, the supplier selector column will not appear in the exported sheet. If 'All suppliers' is selected, the column will appear in the export.

3. Click **Export** and download the Smartsheet.

	B	E	F
	Validate sheet		
9	Next error		
10	* <Name>	Supplier	<Object Type Name>
11	Product A	Great Goods	SalesItem (SalesItem)
12	Product B	Products Galore	SalesItem (SalesItem)
13	Product C	Great Goods	SalesItem (SalesItem)
14		Great Goods	
15		Products Galore	

Users who do not belong to the same supplier group as a product they are working with will be unable to validate the product. Additionally, if the user does not belong to the applicable supplier groups when importing the data, the newly created products will fail to link to their designated supplier groups.

Note: To avoid this, set up a business condition to run on import, checking for the supplier link. If the condition fails, then the creation of the product in question is rolled back.

Important: When using a Multi-Level Smartsheet, the supplier selector column can only be included on a top level.

Maintaining Products Using a Smartsheet

Smartsheets offer suppliers an efficient means for maintaining product data in STEP. The guidelines laid out below cover the general process.

Prerequisites

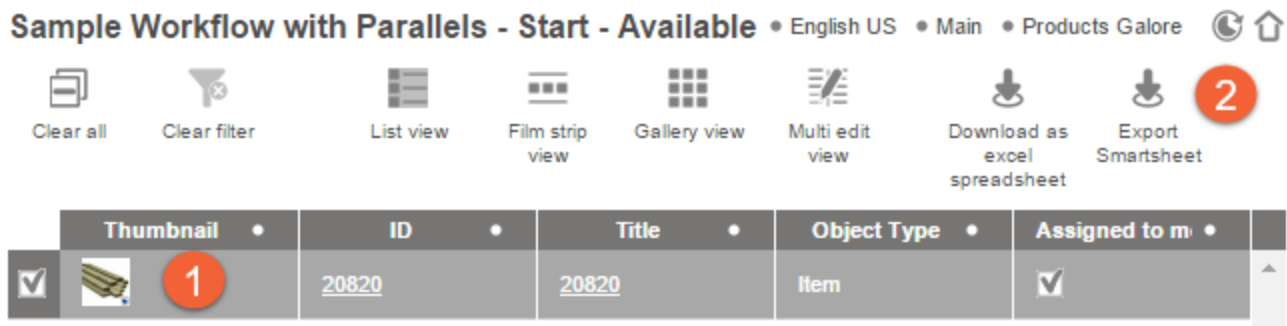
Several setup steps are required before Smartsheets can be used to maintain product data in STEP:

1. In Export Manager, create a Smartsheet data template configuration as described in **Smartsheet Data and Template Configurations**. This export configuration should be configured on the 'Smartsheet Export Action' so that the end user can select it when exporting product data from a workflow state in the Supplier Web UI.
2. In Import Manager, create an import configuration as described in **Smartsheet Import Configurations**. This import configuration should be specified in the accompanying data export configuration and/or the Supplier Web UI Smartsheet Import Action. Without a configuration, a Smartsheet cannot be imported into the Supplier Web UI.
3. In the Supplier Web UI, configure the Smartsheet Import Action as described in the **Smartsheet Import Action** section of the **Web User Interfaces** documentation. This action is used to import the modified product data.
4. In the Supplier Web UI, configure the Smartsheet Export Action as described in the **Smartsheet Export Action** section of the **Web User Interfaces** documentation. This widget is used to export Smartsheet maintenance sheets from the Supplier Web UI.

Product Maintenance

Product data can be maintained via a Smartsheet by: downloading product data from the Supplier Web UI, editing the product data in the exported excel document, and ultimately re-importing the edited Smartsheet back into the Supplier Web UI. Detailed steps are listed below:

1. In the Supplier Web UI, navigate to the workflow widget and click the relevant workflow state. Then, on the Tasklist screen, individually select the products you wish to export, or use multi-select, or the Select All button.
2. Click the Smartsheet Export Action button. In the sample image, this button is labeled 'Export Smartsheet'.



3. In the export dialog, select an export configuration from the dropdown and click **OK**.

Export Smartsheet

Choose Smartsheet Export Configuration

SmartSheetExport_Enrich ▾

✓ OK

✕ Cancel

4. Review the results of the successful background process, and click the file name link to open the Smartsheet for editing.

Background Process Details

IDBGP_122928

Started BySUPPLIERA

DescriptionExcel Smartsheet export from Web UI of any numbers of nodes

Template IDExport Manager Pipeline

Status

✓ Succeeded

Excel--2017-07-27--14-55.xlsm

Started7/27/17 2:55:20 PM

Finished7/27/17 2:55:28 PM

Elapsed8 s

5. Modify the product data in the exported Smartsheet.

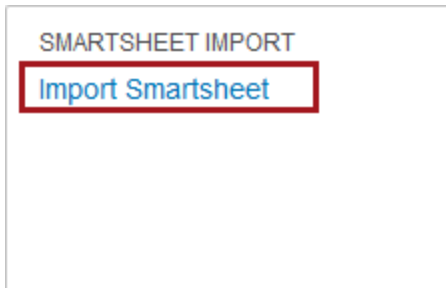
	B	C	D	E
	Validate sheet			
	Next error			
9				
10	* <Name> ▾	<Object Type Name> ▾	UPC ▾	EAN ▾
	20820	Item (Item)	014145204581	0014145204581
11				

For more information on navigating and editing a Smartsheet, see the **Using a Smartsheet** section of the documentation.

6. If required, validate the sheet by clicking the **Validate sheet** button and following the sheet validation process.

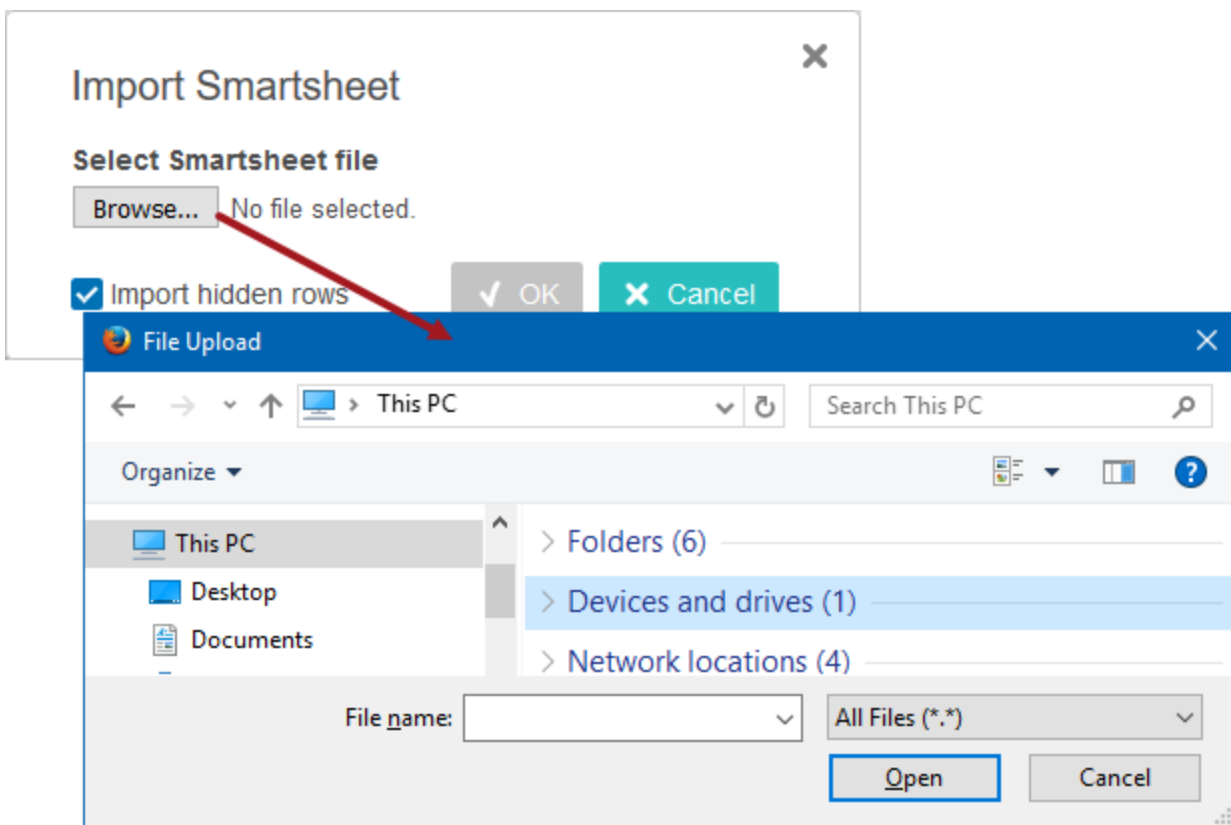
For more information on validating a Smartsheet, see the **Using a Smartsheet** section of the documentation.

7. Once the sheet has been validated, save the document.
8. Navigate back to the Supplier Web UI and click on the Smartsheet Import Action link.



9. In the Import Smartsheet dialog, click **Choose File** and select the Smartsheet, then click **OK**.

Checking the 'Import hidden rows' checkbox will ensure that any hidden rows will be included in the import. If you opt not to import hidden rows, the background process will indicate how many hidden rows were skipped.



Note: If errors occur during import, a Smartsheet file with the products that failed can be downloaded from the import process page. The error messages are displayed as screen tips in the Smartsheet file when hovering over an error cell. **Next Error** can be used to troubleshoot a set of errors. When the errors have been corrected, the

file can be imported again. However, there is limited support for this type of re-import. Some cells will be locked and consequently skipped during import.

Smartsheet Data and Template Configurations

Before a Smartsheet can be downloaded and edited, a Smartsheet data export or template configuration must be created in Export Manager. Once created, the configuration(s) can be assigned to the relevant template widgets and export actions in the Supplier Web UI, or it can be selected when exporting a Smartsheet.

To create a configuration, go through all the steps of the Export Manager wizard. The following section only describes the settings that are specific to Smartsheets.

For detailed information about the Export Manager, see the **Export Manager** section of the documentation.

Important: Smartsheets are designed to be used from the Supplier Web UI **only** and using them out of this context will diminish their capabilities. Furthermore, future Smartsheet enhancements will most likely also depend on Web UI. Specifically, objects should **not** be exported to Excel Smartsheets from the STEP Workbench. Smartsheet templates exported from STEP Workbench will not be able to correct supplier linking of products created during import and will also provide different browsing capabilities. A maintenance Smartsheet exported from the STEP Workbench will not contain any information about workflow and state, so it will not be able to supply information about workflow mandatory columns. In addition, any business rules regarding the current workflow will be unavailable during import. It will also supply a different set of possible targets when maintaining product cross-references.

Export Manager Wizard

Select Objects Step

On this step, select the objects for export.

Note: If the export configuration is saved and used from the Supplier Web UI, the actual products selected in this step are irrelevant. The products will be replaced with the products / categories selected in the Web UI.

Export Manager

Steps

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

Select Objects

ID	Name	Object Type	Path
> Add Objects			

Only export selected objects ☐

Only export leaf objects ☐

Export: Product

Include object types <All object types>

Back Next Finish Cancel

For both data export and template configurations, apply the following settings:

- Do not change the **Export** and **Include object types** default settings.
- For Smartsheet multilevel data exports, check **Only export selected products** and leave **Only export leaf objects** unchecked.
- For Smartsheet template export specifically, note the following:
 - It is recommended to enable **Only export selected products** so that the Export Manager does not have to examine the child products of the selected template product.
 - The selected objects are not exported in Smartsheet templates. However, the first product in the list or selection will be the parent of the top-level products created when the edited template is imported.
 - Enabling **Only export leaf objects** is not recommend since the import may fail, or new products may be created in the wrong place when the resulting template is later used via the saved configuration.

Select Format Step

On this step, select **Excel Smartsheet** as the file export format.


Export Manager

Steps

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

Select Format

Excel Smartsheet

Exports hierarchical data in interactive Excel format

Excel version

Excel 97/2003

Smartsheet type

Multiple level. Hierarchical structure flattened to sheet

Smartsheet usage

Template export (for onboarding new products)

Object Types

Select a product object type for each desired level.

1: Sales Item (SalesItem)

2:

Use Cross-Context Export

No

Smartsheet import configuration

Mandatory metadata attribute

Sort LOV by ID metadata attribute

Hide LOV-ID metadata attribute

Placeholder asset object type

[do not create asset placeholders]

Placeholder asset id prefix

Placeholder asset id separator

Placeholder asset parent classification

Smartsheet auto-size mode

Do not resize

Supplier selector column header

Supplier selector help text

Supplier selector column index

Allow Auto-Filter in Workbook

Yes

Allow Duplicate/Delete row in a Workbook

Yes

1. In the **Excel version** parameter, use the dropdown to select Excel 2007. This parameter is mandatory. Using an older version of Excel could cause problems with the files, including errors upon import. It is best to always choose the newest, supported version of Excel.
2. In the **Smartsheet type** parameter, select one of the following options from the dropdown:

- Select **Multiple level. Hierarchical structure flattened to sheet** if you want to work with multi-level Smartsheets. This is used to initiate and maintain products of different object types and at different hierarchy locations in STEP.
- Select **Multiple object types. Different product types intermixed** if you want to work with multi-object type Smartsheets. This handles multiple object types simultaneously, without them needing to be hierarchically related. This functionality works best when there is a large overlap between the set of attributes that are valid for each of the object types.

This parameter is mandatory.

For more information on Smartsheet types, see the **Using a Smartsheet** section of the documentation.

3. In the **Smartsheet usage** parameter, select one of the following options from the dropdown:
 - Select **Data export (for maintaining product data)** to export data for products already in the STEP database.
 - Select **Template export (for onboarding new products)** to export an empty Smartsheet with formatted columns that can be used to initiate new products.

This parameter is mandatory.

Note: Template exports cannot export reference values but reference types can be exported. It will not contain any data for the object type.

4. In the **Object Types** parameter, select one or more object types to be included in the export from the dropdown.
 - For multi-level Smartsheets, specify the object type of the products for each level in the multi-level export. The object type must be valid for each level. The top level must be allowed under the root product selected in the Select Object step of the wizard. A maximum of five (5) entries are allowed.
 - For multi-object type Smartsheets, specify the object type(s) to export. There is no limit on how many entries are allowed.

This parameter is mandatory.

5. In the **Use Cross-Context Export** parameter, use the dropdown to determine if language dependent attributes should be exported in the language of the selected context.

Set to **Yes** to specify which contexts should be included in the export. A column is created in the Smartsheet for each selected context on all language-dependent attributes. Remember to select the export context specified in the Advanced step of the wizard in the cross-context list. If the export context is not selected, there is a risk of exporting data in an incorrect context.

To add a context, click **Select Context** under the dropdown and choose the context(s) from the window that appears.

This parameter is mandatory.

6. In the **Smartsheet import configuration**, select an existing import configuration to apply to the Smartsheet. The import configuration may run business rules and other actions on the products upon import. This field is only mandatory if an import configuration has not been set on the 'SmartSheetImportAction' in the Supplier Web UI.

For more information of setting up the import configuration, see the **Smartsheet Import Configurations** section of the documentation.

7. In the **Mandatory metadata attribute** parameter, click the ellipsis button (...) and select an attribute to mark columns as mandatory in the Smartsheet. This attribute must be a Description attribute and have the validation base type Integer or Numeric. Once created, the attribute must be added as metadata to all relevant attributes or attribute links. For any attribute that *should* be made mandatory in the Smartsheet, a non-zero integer must be populated in the metadata attribute.

This parameter is optional.

For more information, see the **Mandatory Attributes** section of the **System Setup** documentation.

8. In the **Sort LOV by ID metadata attribute** parameter, click the ellipsis button (...) and select a metadata attribute to sort attribute columns that use LOVs by value ID. This attribute must be a Description attribute and have the validation base type Integer or Numeric. Once created, this attribute must be added as metadata to all relevant attributes. For any LOV that *should* have its values sorted by ID, a non-zero integer must be populated in the metadata attribute.

This configuration changes the order in dropdown lists and pop-up LOV selectors in the Smartsheet column(s).

The metadata value is ignored if the attribute is not LOV based or if the LOV does not use value ID. This makes it possible to have the same values for two different columns sorted both by value and value ID.

This parameter is optional.

For more information, see the **Adding IDs to Values in LOV** section of the **System Setup** documentation.

9. In the **Hide LOV-ID metadata attribute** parameter, click the ellipsis button (...) and select a metadata attribute to exclude external IDs from individual LOV-based Smartsheet columns. This is useful if you only want to display the value of the LOVs rather than their value IDs. This attribute must be a Description attribute and have the validation base type Integer or Numeric. Once created, this attribute must be added as metadata to all relevant attributes. For any LOV that *should not* display value IDs, a non-zero integer must be populated in the metadata attribute.

This parameter is optional.

10. In the **Placeholder asset object type** parameter, use the dropdown and select the object type of the assets that are created and referenced during import. This must be specified to enable asset placeholders. The available options are system dependent.

Note: Placeholder assets make it possible for the created assets to have a determinable ID so other processes are able to import and maintain actual asset content.

If **[do not create asset placeholders]** is selected, the asset placeholder system is disabled and normal asset references are maintained in the Smartsheet.

This parameter is mandatory.

11. In the **Placeholder asset ID prefix** parameter, enter a prefix for the ID placeholder assets created upon import. This ensures that asset IDs are unique across all suppliers during export from Workbench. The prefix will be changed during Web UI export. This field is optional.
12. In the **Placeholder asset ID separator** parameter, enter a separator to be inserted in the final ID between the prefix and the file name.

For example, if a member of the user group 'Vendor Ltd.' enters the asset file name 'Blue Running Shoe.png', the ID of the final placeholder asset is 'Vendor Ltd.:Blue Running Shoe.png', with ':' being the separator. This ID is, with high probability, unique between vendors. If the ID contains invalid characters or exceeds the 40 characters limit, the name will fail validation.

This field is optional.

13. In the **Placeholder asset parent classification** parameter, click the ellipsis button (...) and specify where placeholder assets are created when the Smartsheet is re-imported. This parameter may be useful if the imported Smartsheet was exported from STEP Workbench (and is therefore not associated with a Supplier group). If this parameter is filled in and the smart sheet is exported / imported from the Web UI, then this parameter is overwritten and the asset placeholder will go below the correct Supplier Asset folder in the Supplier Hierarchy.

This field is optional.

14. In the **Smartsheet auto-size mode** parameter, use the dropdown to determine whether to resize the exported Smartsheets by column-header width or by cell-content width using word wrap, or not to resize at all. This field is mandatory.

15. In the **Supplier selector column header** parameter, enter a name for the supplier selector column.

Note: This parameter, as well as the 'Supplier select help text' and 'Supplier select column index' below, are only available when creating a Template export and when 'Enable all-view for users that are a member of multiple suppliers' is set to 'Y' in system properties (Users & Groups node).

Unless this parameter is set, the supplier selector column is called 'Supplier' by default.

This field is optional.

16. In the **Supplier selector help text** parameter, enter what the tooltip says when the cursor is placed over the supplier selector column header in the exported sheet. This field is optional.
17. In the **Supplier selector column index** parameter, determine which column the supplier selector occupies. If the parameter is not set, then the column will be placed as far to the left as possible. If this parameter is set, all columns mapped in the Map Data step will be bumped to the right of whichever column is specified. Valid values are between 2 and 999. This field is optional.

18. For the **Allow Auto-Filter in Workbook** parameter, determine whether or not rows can be filtered in the Smartsheet via the dropdown. This field is mandatory.
19. For the **Allow Duplicate / Delete row in a Workbook** parameter, determine whether or not entire rows can be duplicated and deleted in the Smartsheet via the dropdown. This field is mandatory.
20. Click **Next** to move on to the next screen.

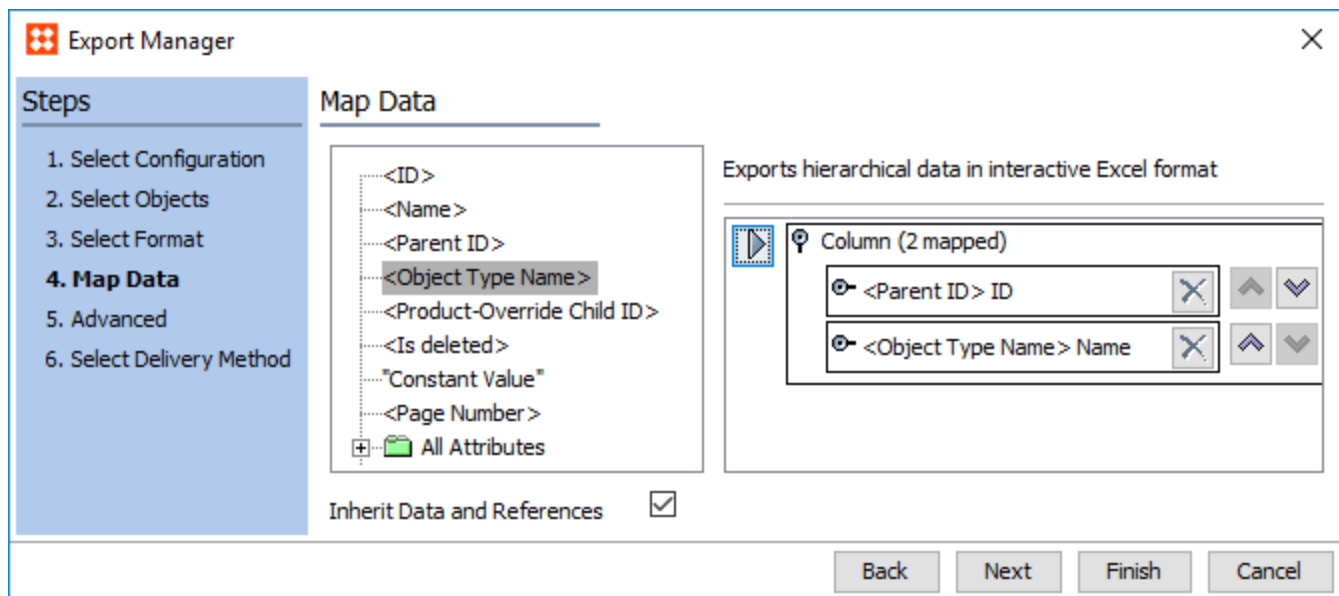
Map Data Step

Different mapping options are available depending on whether you are making a multi-object type or multi-level Smartsheet.

Special options for multi-object type Smartsheets:

- **Parent ID** is used as a Parent ID selector in the excel sheet so users can indicate the correct parent IDs for the products they are working with.
- **Object Type Name** Smartsheet column is used as an Object Type selector in the excel sheet so users can indicate the object types for the products they are working with.

These two columns can be mapped in multi-level Smartsheet as well, but cannot be manually edited.



Special options for multi-level Smartsheets:

- **Start level** specifies that the selected column starts a new object level. The top level can be left empty because it is implicit, however a value must be provided for all other Start levels below the first. The actual value is not important. To add a value, click the transformation button () and enter a value in the window that appears.

The Start level columns are also used for grouping products at each level. For example, all rows with 'T-Shirt' in the **Name** column (which is designated as a Start level column) become one product. All rows that have both

'Pink' in the **Color** column and 'T-Shirt' in the **Name** column become one SKU Level product under the new T-Shirt product.

It is not recommended to group by multivalued attributes and reference columns.

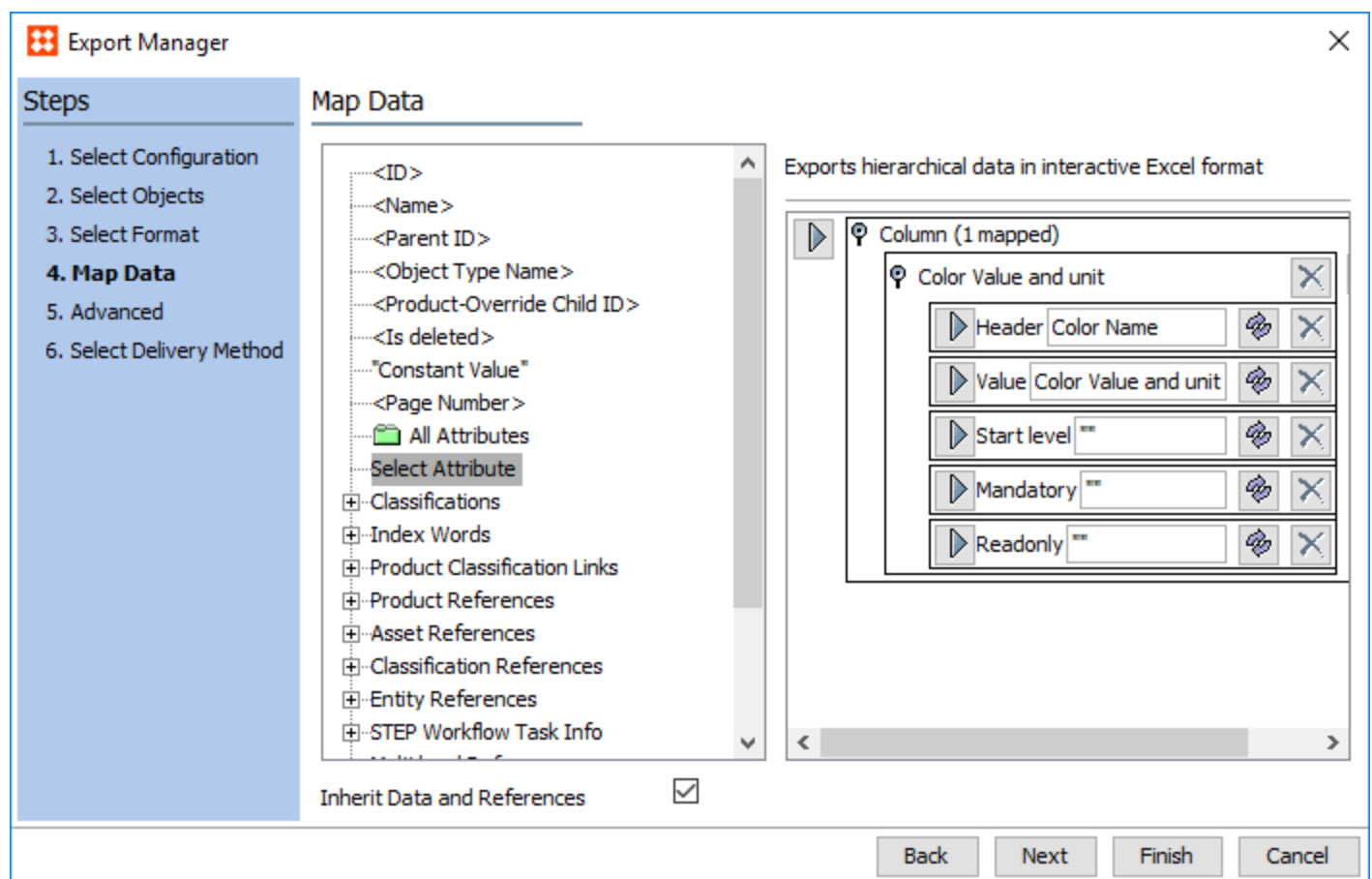
- **Mandatory** makes the column appear mandatory in the downloaded Smartsheet. If the Smartsheet is validated in Excel, cells that are empty but marked as mandatory will appear as invalid. However, no errors are generated when the Smartsheet is imported. To mark a column as mandatory, click the transformation button () and enter any value in the window that appears.

For more information, see the **Mandatory Attributes** section of the **System Setup** documentation.

- **Read only** makes the column read-only and locks all cells in the column in Excel. This setting is only relevant for data export. In template export, mappings marked as read-only are omitted.

To mark a column as read only, click the transformation button () and enter any value in the Transformation dialog. Click the Save button on the Transformation dialog and the Readonly mapped value is updated.

The following image shows that none of the special options have a transformation.



Important: Do not add any kind of transformation to the value of a mapping unless it is marked as read-only. Transformations are applied during Smartsheet data export. However, during Smartsheet import, the column will be handled as if it contains Value and Unit. This can lead to import errors or unexpected imported values.

Additional Information About Mapping

If attribute groups are mapped at this step, the individual attributes are ordered using the **Display Sequence** metadata for the attributes. If an attribute defines or starts a level in the Smartsheet, the first column in the group is used for sorting at this level.

Note: Attribute groups are not mapped recursively.

Sub-level description attributes cannot be exported to a Smartsheet unless a product of the sub-level object type is included in the export. This is usually not the case in template exports where it is recommended to select the **Export only selected products** option.

When Smartsheets are exported, LOV values are embedded in the Excel workbook so that a list of valid values is available when values are specified in an LOV-based column. The exported values are filtered according to the filters specified on the attribute and on the attribute link in the Product hierarchy.

Note: Smartsheets only support LOV filters on attributes and attribute links.

Advanced Step

Do not select **Locale conversion from context**. It can break the automatic conversion to the client's native decimal separator and can cause the import of the Smartsheet to fail.

Select Delivery Method Step

If configured, a Smartsheet background process can be specified on this step. For more information, see the **Smartsheet Background Processes and Queues** section of the documentation.

Saving and Modifying the Configuration

Saving an export configuration is described in the **Running a Data Export** topic of the **Export Manager** documentation.

Once saved, an export configuration can be edited, as described in the **Maintaining a Saved Export Configuration** topic of the **Export Manager** documentation.

Additional Configuration Options

Smartsheet column headers are capable of displaying tooltips when mousing over them. The text provided in the tooltips is controlled by a metadata attribute that must be configured in System Setup.

To configure the metadata attribute, navigate to **System Setup > User & Groups > Context Help Metadata attribute** and enter the name of the metadata attribute.

System Setup

- Test
- Event Processors
- Gateway Endpoints
- GDSN
- Global Business Rules
- Inbound Integration Endpoints
- Match Codes and Matching Algorithms
- Outbound Integration Endpoints
- Web UIs
- Workflow Profiles
- Workflows
- Derived Events
- Object Types & Structures
- Tags
- Units
- Users & Groups**
- Reference Types
- Workspaces
- Table
- Keys
- Event Queues
- Component Models
- Recycle Bin

System Settings

System Settings
Log

> Tag format for Web Services and ReST APIs
HTML3

Web UI Settings

Name	Value
> Control override meta attribute	
> Default context	Context1
> Default workspace	Main
> Web UI supplier classification object type	SuppliersRoot
> Web UI supplier products classification object type	SuppliersProducts
> Assets classification object type	SuppliersAssets
> Batches classification object type	SuppliersBatches
> Batch classification object type	SuppliersBatch
> Excel-template asset	QuickSheetTemplate
> Proof view stylesheet attribute	
> Step-users Web UI batches folder	SuppliersBatches
> Step-users Web UI assets folder	SuppliersAssets
> Context Help metadata attribute	AttributeHelpText
> Link type for vendor classification to product link	SupplierLink

Smartsheet Import Configurations

In order to import Smartsheets into STEP an import configuration must be available. This import configuration can be specified upon export of the product data / template (via the Smartsheet export configuration) or upon import (via the 'SmartsheetImportAction' configuration in Web UI).

For more information on configuring the Web UI import action, see the **Smartsheet Import Action** section of the **Web User Interfaces** documentation.

Smartsheets can be imported via the STEP Workbench Import Manager, however it is recommended to use the Smartsheet Import Action in the Supplier Web UI.

Note: While an import configuration is required, the vast majority of a Smartsheet's functionality is determined by the product data / template export configuration. For more information, see the **Smartsheet Data and Template Configurations** section of the documentation.

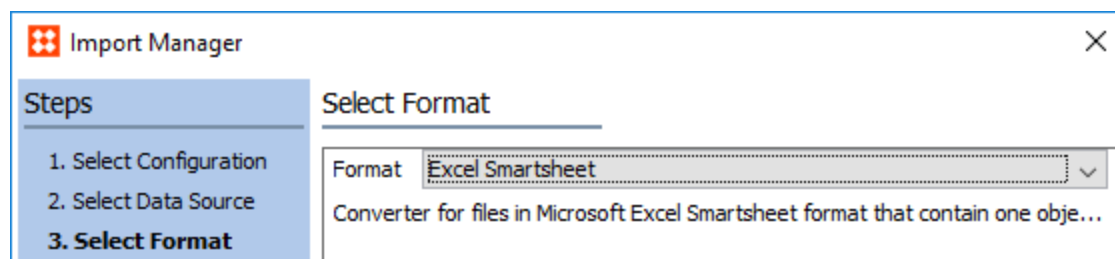
Creating an Import Configuration

To create a configuration, go through all the steps of the Import Manager wizard. The following section only describes the settings that are specific to Smartsheets.

For more information on the Import Manager, see the **Import Manager** section of the documentation.

Select Format Step

In the **Format** parameter, ensure 'Excel Smartsheet' is selected.



Important: If users encounter an error during the import process, it could be due to the template being used is set to an older version of Excel. It is recommended that users use the newest, supportable version of Excel. For more information on supported versions of Excel, see the current **Platform and Software Support** documentation in the **System Release and Patch Notes**.

Advanced Settings Step

If desired, it is possible to delete / replace references when importing a Smartsheet.

References can be cleared or replaced by selecting them during the final step in Import Manager:

1. In the **Remove Un-Mapped References** parameter, click the relevant 'Add Reference Type' option.

Remove Un-Mapped References

☐ Classification References

Product Reference Type >

> AccessoryOptional

→ Add Product Reference Type

Asset Reference Type >

> Brand Name Logo

→ Add Asset Reference Type

Entity Reference Type >

→ Add Entity Reference Type

Classification Product Li... >

→ Add Classification Product Link Type

2. A warning message will appear. Click **OK**.

Confirm Removal of Asset Reference

?

Please confirm that you wish to remove unmapped asset references for the imported objects

OK Cancel

3. Click to open the dropdown list and select the appropriate reference type.

Remove Un-Mapped References

☐ Classification References

Product Reference Type >

> AccessoryOptional

> Add Product Reference Type

Asset Reference Type >

> Brand Name Logo

> Brand Name Logo

Illustration

Installation Manual

> MSDS

Owners Manual

Primary Product Image

Product Image

> Video

In the above example, the asset reference 'Primary Product Image' will be replaced by any new references included in the Smartsheet import. If the reference field is blank, the import will clear the reference from the corresponding product. Only Primary Product Image references that are included in the input file will be present on the products included in the file.

If configured, a Smartsheet background process can also be specified on this step. For more information, see the **Smartsheet Background Processes and Queues** section of the documentation.

Saving and Modifying the Configuration

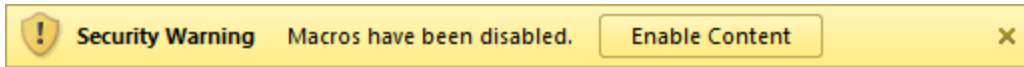
Saving an import configuration is described in the **Running a Data Import** topic of the **Import Manager** documentation.

Once saved, an import configuration can be edited, as described in the **Maintaining a Saved Import Configuration** topic of the **Import Manager** documentation.

Using a Smartsheet

Unlike typical Excel sheets, Smartsheets are very restrictive and must be handled in a very specific way.

1. In **Excel**, open the applicable Smartsheet. If the Smartsheet has never been opened before it will open on the **Cover** tab, which contains information about the download date and the color codes used in the Smartsheet. Additionally, a security warning will appear at the top of the cover page.



Click **Enable Content** to enable macros, otherwise, the Smartsheet will not work correctly.

2. In the workbook, one or more additional tabs are available, each corresponding to the categories selected during the export process. Click the desired tab to open the sheet.

The screenshot shows an Excel interface with a Smartsheet. The active sheet is 'Games', highlighted in red in the bottom tab bar. The table has the following data:

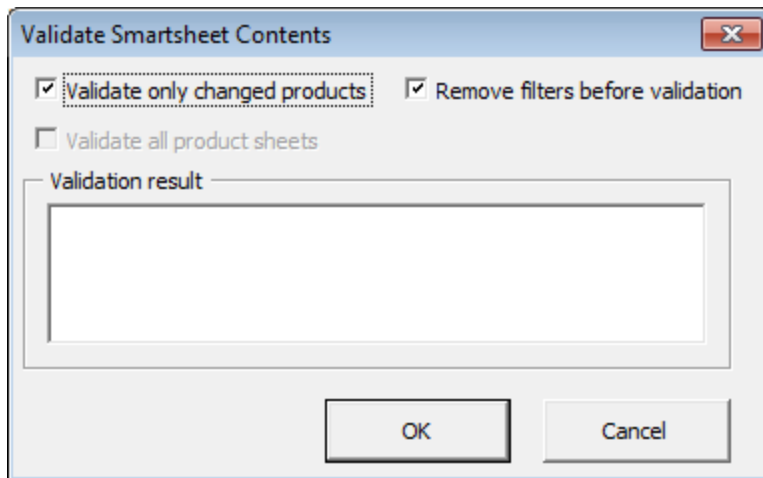
	B	C	F	G	H
9	Validate sheet				
	Next error				
10	* <Name>	Supplier	Co-op	PvP	Multiplayer
11	Game A	Products Galore	Yes	Yes	Yes
12	Game B	Products Galore	No	No	No
13	Game C	Products Galore	Yes	No	Yes
14	Game D	Products Galore			

3. Enter the data for the objects to be imported.

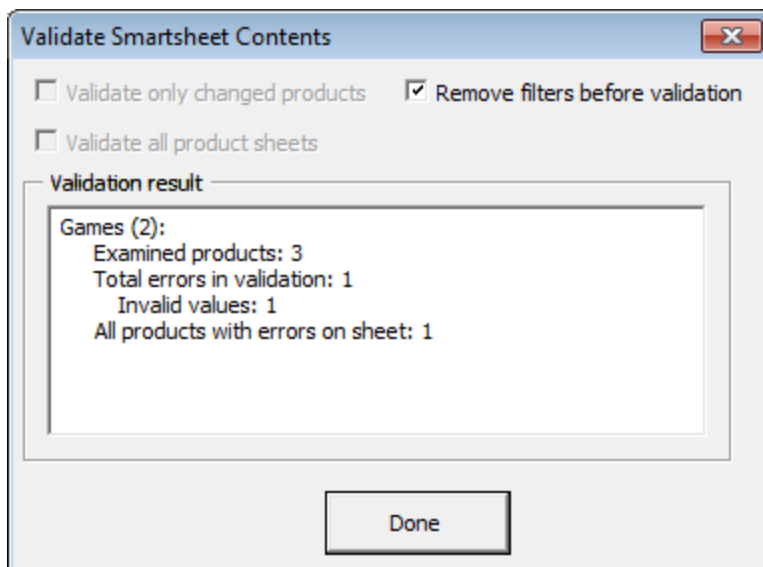
Note: In template sheets, all rows are editable, but any unmapped columns are locked to the user. In maintenance sheets, all unmapped rows and columns are locked.

4. Click **Validate sheet** in the top left corner. In the 'Validate Smartsheet Contents' dialog, three check boxes are available:
 - **Validate only changed products** - The Smartsheet will only validate products that have been changed. Leaving this box unchecked means all products are validated.
 - **Validate all product sheets** - The Smartsheet will validate all sheets in the workbook. Leaving this box unchecked means only products that appear on the currently selected sheet will be validated. This checkbox is only available if multiple sheets exist in the workbook.

- **Remove filters before validation** - The Smartsheet will remove all filters from the sheet(s) during validation. Leaving this box unchecked means that any rows currently being filtered will not be validated. Additionally, if the user hides any rows they will not be considered when troubleshooting errors via the **Next Error** button.



5. Click **OK**.
6. In the **Login** dialog, enter the relevant STEP user name and password and click **Login**.
7. After the validation is run the **Validate Smartsheet Contents** window will display if the validation succeeded or if there are errors that need to be corrected.



Errors can be located and manually corrected via the **Next Error** button.

- Errors for mandatory fields are shown in the cell that is missing data.
- Errors from the business condition are shown in the first column.

3. After resolving the errors, save the file.

Note: Some fields, such as node pickers, will prompt the user for a STEP user name and password. The system will only prompt for these credentials once.

Copying Data in a Smartsheet Template

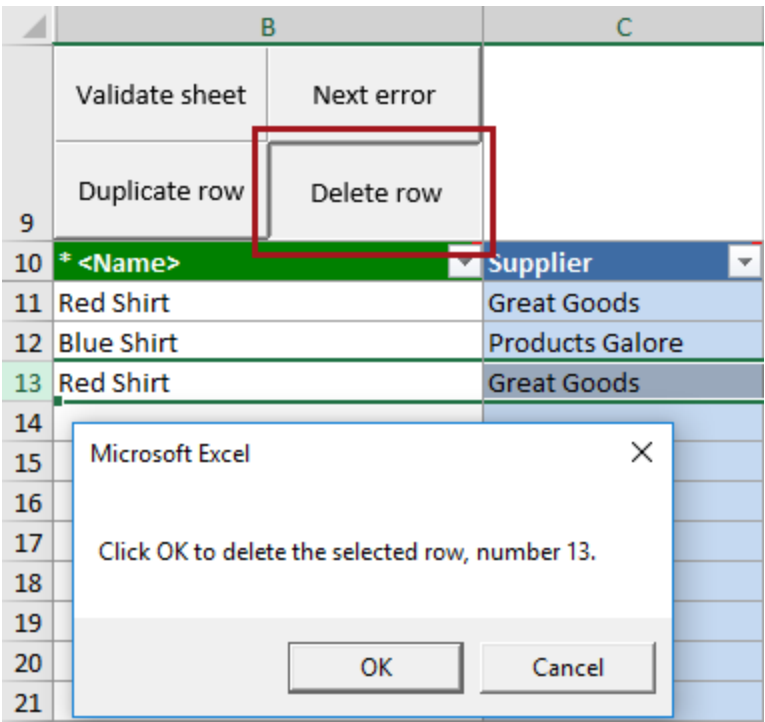
When initiating products using a Smartsheet template, users can duplicate data on a row-by-row basis, potentially saving them a lot of time. Note, however, that rows can only be copied in Smartsheet templates, and may need to be configured to do so.

1. To copy a row, select it and click the **Duplicate row** button.

	B	C
	Validate sheet	Next error
	Duplicate row	Delete row
9		
10	* <Name>	Supplier
11	Red Shirt	Great Goods
12	Blue Shirt	Products Galore
13	Red Shirt	Great Goods

2. A copy of the row will appear in the next open row.

This action cannot be reversed using the 'Undo' function built into Excel. Instead, to remove an unneeded row, select the row and click **Delete** row. When asked if you want to delete the selected row, click **OK** to delete it.



For more information on configuring these functions, see the **Smartsheet Data and Template Configurations** section of the documentation.

Smartsheet Types

Multi Level Smartsheets

In some cases, it is necessary to use Excel Smartsheets to initiate and maintain products of different object types and at different hierarchy locations in STEP. This is useful for maintaining / initiating product hierarchies, and displays the hierarchy in a flattened structure for easy editing.

For each product category selected for download, a sheet will be added to the Excel workbook.

	B	C	D
	Validate sheet	Next error	
	Duplicate row	Delete row	
9			
10	* <Name>	* SKU	COLOR
11	T-Shirts	Black Shirt	Black (179464)
12	T-Shirts	Blue Shirt	Blue (138809)
13	V-Necks	Black V-Neck	Black (179464)
14	V-Necks	Blue V-Neck	Blue (138809)

Multiple Product Object Type Smartsheets

In some cases it is necessary to use Excel Smartsheets to initiate / maintain products of different object types intermixed in one sheet. The multi-object Smartsheet allows users to easily upload multiple object types simultaneously, and without them needing to be hierarchically related. This functionality works best when there is a large overlap between the set of attributes that are valid for each of the object types.

Multi-object Smartsheets handle multiple mutually independent object types in the same Excel workbook. For each product category selected for download, a sheet will be added to the Excel workbook. Each sheet can contain all configured object types alongside each other.

	B	C	D	E
	Validate sheet	Next error		
	Duplicate row	Delete row		
9				
10	* <Name>	<Parent ID>	<Object Type Name>	
11	Baseball Caps	Hats and Caps Sa ...	SalesItemFamily (SalesItemFamily)	
12	Red Cap	Hats and Caps Sa ...	Sales Item (SalesItem)	
13			...	

Note: In order to support this, the object type must be mapped in the export configuration. If importing new products, this column must be populated with the desired object type for all products.

Additional Information About Smartsheets

System Configuration

Smartsheets have some additional configuration options that can impact the entire functionality and the rest of the system. These options include:

- Using special background processes and queues for Smartsheets specifically. For more information, see the **Smartsheet Background Processes and Queues** section of the documentation.
- Limit supplier browsing based on user privileges. For more information, see the **Smartsheet System Properties** section of the documentation.
- Override the URL of the Web UI server when exporting a Smartsheet. For more information, see the **Smartsheet System Properties** section of the documentation.
- Add Smartsheet validation to Excel's Trusted Header authentication. For more information, see the **Smartsheet System Properties** section of the documentation.

Smartsheet Cover Tab

In addition to the tab containing data, all Smartsheets exported contain an cover sheet tab. This tab includes details about the file, information about macros, and an explanation of color coding of various fields.

	B	C	D	E	F	G	H	I
2	Vendor Item Enrichment Spreadsheet							
3	User Name:	User 3						
4	Vendor Name:	Great Goods						
5	Download Date:	1/2/17 9:44 AM						
6								
7	This spreadsheet contains one or more sheets with item data that can be altered, validated and subsequently uploaded to the Web UI. Also, you can add new Items to the sheets. It is strongly recommended to validate the data (by clicking the Validate button) before upload to the Web UI.							
8								
9	The functionality of this workbook may be reduced if Macros are disabled. To enable Macros you will need to select Enable Macros in the the security dialog that appears when opening the spreadsheet. Please be advised this workbook contains no harmful macros							
10								
11	Color Codes:							
12	Blue = Mandatory. Information must be added.							
13	Yellow = Read Only. Information can not be altered.							
14	Yellow with grey text = Conditionally Read Only. Information can not be altered.							
15	Red = The cell holds data that failed the validation.							

Smartsheets Special Attribute Handling

Smartsheets support and enforce conditional attributes and workflow state mandatory attributes.

Conditional Attributes

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.

	B	C	D	E
	Validate sheet			
9	Next error			
10	* <ID>	<Name>	Fit	Flex Fit
11	101567	20808-013		
12	110190	Hat123		Yes
13	12345-001	A		
14	17739	1112121		Yes
15	20808	20808-012		

For more information on how conditional attributes behave in Smartsheet exports, see the **Conditional Attribute Display** section of the **System Setup** documentation.

Workflow State Mandatory Attributes

When maintaining objects in a Smartsheet that have been exported from a workflow state (i.e., from the Web UI Tasklist), any cells containing attributes that are mandatory for that particular state are highlighted blue. Additionally, unless the condition driving the mandatory status of the attributes returns 'False', the user must enter values for all applicable attributes before the sheet can be validated.

	B	C	F
	Validate sheet		
9	Next error		
10	* <Name>	Supplier	Co-op
11	Game A	Products Galore	Yes
12	Game B	Products Galore	No
13	Game C	Products Galore	Yes

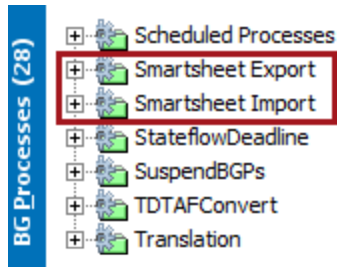
Important: Attributes mandatory for transitions are not enforced in Smartsheets as the selected transition is not known.

For more information, see the **Conditionally Mandatory Workflow Attributes** section of the **STEP Workflows** documentation.

Smartsheet Background Processes and Queues

When the default configurations are used, Smartsheet imports and exports are placed in the queue along with other import and export processes. However, Smartsheet imports and exports can be set up to run on unique queues so that they are not competing for resources with other import and export processes.

To accomplish this, the user must populate the appropriate process template when executing an import or export (or have it set in the configuration). Once executed, the import or export is displayed in a separate area in the **Background Processes** tab in workbench. Smartsheet export processes will appear in the **Smartsheet Export** folder while import processes will appear in the **Smartsheet Import** folder.



To view the background processes initiated by another user, click the 'All Users' button  at the top of the BG Processes tab.

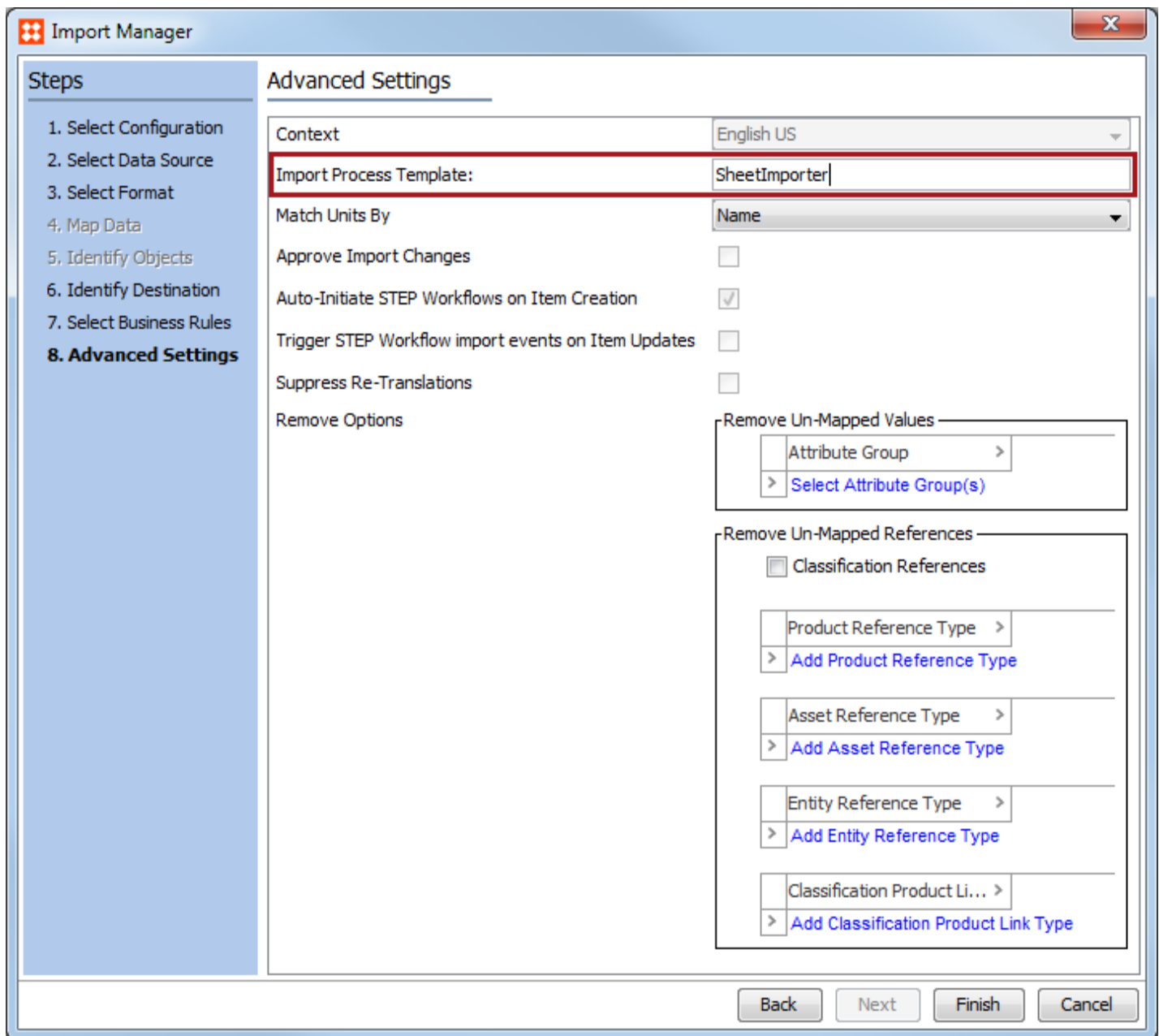
Important: 'BackgroundProcess.ProcessType.SheetExporter.Queue' and 'BackgroundProcess.ProcessType.SheetImporter.Queue' default to the same queue as Export Manager and Import Manager, and must be changed in sharedconfig.properties before they can run in their own queues. This step must be completed before the corresponding process templates can be used.

Using Smartsheet Import Background Process

On the Advanced Setting step in the Import Manager, navigate to the **Import Process Template** field and type 'SheetImporter'.

This will tell the system to display the import in the **Smartsheet Import** folder on the **Background Processes** tab.

See the **Import Manager** section of the documentation for more information.



Import Manager

Steps

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

Advanced Settings

Context: English US

Import Process Template: SheetImporter

Match Units By: Name

Approve Import Changes: ☐

Auto-Initiate STEP Workflows on Item Creation: ☒

Trigger STEP Workflow import events on Item Updates: ☐

Suppress Re-Translations: ☐

Remove Options

Remove Un-Mapped Values

Attribute Group: [Select Attribute Group\(s\)](#)

Remove Un-Mapped References

☐ Classification References

Product Reference Type: [Add Product Reference Type](#)

Asset Reference Type: [Add Asset Reference Type](#)

Entity Reference Type: [Add Entity Reference Type](#)

Classification Product Li...: [Add Classification Product Link Type](#)

Back Next Finish Cancel

Using Smartsheet Export Background Process

On the Select Delivery step in Export Manager, navigate to the **Export Process Template** field and type 'SheetExporter'.

This will tell the system to display the export in the **Smartsheet Export** folder on the **Background Processes** tab.

See the **Export Manager** section of the documentation for more information.


Export Manager

Steps

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

Select Delivery Method

File

Delivers exported data in a file. You will be notified when the file is ready for download.

Export Process Template:
SheetExporter

Back
Next
Finish
Cancel

Smartsheet System Properties

Configuration properties, set in the sharedconfig.properties file, are available for Smartsheets and can alter the system behavior.

Limit supplier browsing

It is possible to allow the Smartsheet user to browse and search the full data set in STEP in accordance with the user's privileges. This is controlled using the 'Smartsheet.LimitSupplierBrowsing' configuration property.

If this property is true, users of a Smartsheet that has been exported for a specific supplier will only be able to browse / search for products and assets linked into appropriate supplier classifications. This means that while product references and asset references remain intact, the Smartsheet displays only the objects the user has privileges to access.

If this property is false, users will be able to browse / search for all products and assets using the same privileges as in STEP Workbench.

Important: Regardless as to how this property is set, the browse / query service disregards whether or not the current user is a supplier user. If the property is true, even super users are limited to supplier classifications. If false, supplier users will be able to access the full hierarchy using normal privileges.

The setting defaults to true, as these limitations have always been applied in Smartsheets.

Service URL override

In order for the online data validation of Smartsheet data to work, the address (URL) of the service must be embedded in the exported sheet. When exported through Web UI, this setting is assigned the address of the Web UI server. It is possible to override this with a configuration setting (for STEP Workbench export or load balancing). The STEP attributes and references used in the Smartsheet can only be validated if this URL ultimately (through whichever Web UI server) points back to the STEP server from which the Smartsheet is exported. For example, it is possible to export a Smartsheet from WebUIServerA and have it validated at WebUIServerB as long as they share the same STEP backbone.

This setting is controlled via the 'Smartsheet.ValidationService' configuration property.

Note: Changing this setting may corrupt existing Smartsheet configurations. After a change, test the existing configurations and modify or recreate as necessary.

Trusted headers authentication

One can make use of Trusted Headers authentication in conjunction with Excel Smartsheets. This enables end users to, for example, make use of their intranet login and password when validating Excel Smartsheets. To make use of Trusted Headers in Excel Smartsheets, use the following instructions:

1. An authentication proxy needs to be in place that can:
 - Make basic authentication from the calls from Excel
 - Extract the User ID and password from the basic authentication token

2. Forward all requests, including all payloads, from Excel to STEP if the request is authenticated
3. Add a header to all requests to STEP containing the User ID
4. The following configuration parameters need to be set in the STEP sharedconfig.properties file:
 - Excel.UseTrustedHeaders=true
 - Excel.TrustedHeaders.Username=[Name of the header that contains the User ID]

FixedWidth Format

A FixedWidth format is a text file with rows and columns, and each column has a fixed width.

Format Availability

FixedWidth is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**

Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

Inbound Data

For examples of ways to modify fixed width data for import and further explanations on the inbound parameters, see **FixedWidth Format Configuration Examples**.

Inbound Parameters

- **Columns are specified by** allows you to determine how the initial element is identified via the following options:
 - **end positions (zero-based)** - the last character counted is counted with a 'zero' value.
 - **end positions (1 based)** - every character is counted and indicates the last character in a value set.
 - **width**- the numeric values of how long each section should be from the last character grouping is counted, and each new batch starts with 'one' as the value of the character.
- **Columns specification** allows you to further specify columns via the following options:
 - For **end positions (zero-based)**, specify the number (position) of the before last (n-1) character of each column.
 - For **end positions (1 based)**, type the number (position) of the last character of each column.
 - For **width**, specify the width / length in characters of each column.
- **Has Header** allows you to indicate if the file has a header line. If the first line (row) of the FixedWidth file has header information such as attribute names, you can use the Auto Map feature mapping the columns of data to STEP objects.
- **Filler Character**, if necessary, specify the filler character that you want to remove from the fields.
- **Are records separated by new line**, when set to 'yes', records in the inbound file are separated by new lines.
- **Character Set** select the desired character set to determine the characters that can be successfully imported. Options include Windows-1252, ISO-8859-1 (also known as the Latin-1 character set), UTF-8, or UTF-16. The common setting is **UTF-8**, unless you have a reason to do otherwise.

Import Manager

Import Manager

×

Steps

1. Select Configuration

2. Select Data Source

3. **Select Format**

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Select Format

Format

FixedWidth

▼

Converter for length separated formats where each line contains data about one object in fields separated by a fixed length

Columns are specified by

end positions (1 based)

▼

Columns specification

8,18,26,34

Has Header

yes

▼

Filler character

*

Are records separated by new line

yes

▼

Character Set

UTF-8

▼

Conversion Preview:

ID Name	>	Column 1Co	>	lumn 2Co	>	Column4	>	Column5	>
XYZ12		Pants		V2c		V2d		12345678	

< >

Back

Next

Finish

Cancel

IIEP

Inbound Integration Endpoint Wizard

×

Steps

1. Identify Endpoint

2. Choose Receiver

3. Configure Endpoint

4. Configure PreProcessor

5. Configure Processing Engine

5.1. Select Sample File

5.2. Select Data Source

5.3. Select Format

5.4. Map Data

5.5. Identify Objects

5.6. Identify Destination

5.7. Select Business Rules

5.8. Advanced Settings

6. Configure PostProcessor

7. Schedule Endpoint

8. Configure Error Reporter

Configure Processing Engine : Select Format

Format

FixedWidth

Converter for length separated formats where each line contains data about one object in fields separated by a fixed length

Columns are specified by

end positions (zero based)

Columns specification

7,17,25,33

Has Header

yes

Filler character

*

Are records separated by new line

yes

Character Set

UTF-8

Conversion Preview:

ID Name	>	Column 1Co	>	lumn 2Co	>	Column4	>	Column5	>
XYZ12		Pants		V2c		V2d		12345678	

Back

Next

Finish

Cancel

FixedWidth Format Configuration Examples

The 'Columns are specified by' parameter determines how the initial element in the import file is identified and therefore affects how the full data file is imported. Each option enables you to modify the separation of the data fields differently, as demonstrated in the following examples. Choose the option that causes the Conversion Preview panel to display the data attribute values as they should exist in STEP after the import.

For more details, see the **FixedWidth Format** documentation.

Select Format

Format

FixedWidth

Converter for length separated formats where each line contains data about one object in fields separated by a fixed length

Columns are specified by

end positions (1 based)

Columns specification

8,18,26,34

Has Header

yes

Filler character

*

Are records separated by new line

yes

Character Set

UTF-8

Conversion Preview:

ID Name	>	Column 1Co	>	lumn 2Co	>	Column4	>	Column5	>
XYZ12		Pants		V2c		V2d		12345678	

Example Data

Each of the examples below use the following input file.

```
ID      Name      Column 1Column 2Column 3
XYZ12***Pants*****V2c*****V2d*****12345678
```

End Positions (Zero Based) Example

The 'end positions (zero-based)' option means the last character counted is counted with a 'zero' value. In other words if the input values was 'trv2**tgh4****j5r' then the numeric value to put 'trv2**' in its own column and 'tgh4****' in it's own column would be 5,13. This is because in the first batch of characters, 'trv2**', the last asterisk (*) counts as 'zero'. The last asterisk (*) in the 'tgh4****' is technically number 14, but because it counts as 'zero', you put the number 13 next to signal that the character after it is considered the end of the batch.

1. **Columns are Specified by** is set to **end positions (zero based)**.
2. **Columns Specification** is set to **7,17,25,33**.
3. **Has Header** is set to **Yes**.
4. **Are records separated by new line** is set to **Yes**.
5. **Character set** is set to **UTF-8**.

The Conversion Preview displays the filler characters.

Select Format

Format FixedWidth

Converter for length separated formats where each line contains data about one object in fields separated by a fixed length

Columns are specified by end positions (zero based)

Columns specification 7,17,25,33

Has Header yes

Filler character

Are records separated by new line yes

Character Set UTF-8

Conversion Preview:

ID Name	Column 1Co	lumn 2Co	Column4	Column5
XYZ12***	Pants*****	V2c*****	V2d*****	12345678

6. **Filler character** is set to * (which is being used to take up unneeded space between values)

The Conversion Preview no longer displays the filler characters.

Select Format

Format FixedWidth

Converter for length separated formats where each line contains data about one object in fields separated by a fixed length

Columns are specified by end positions (zero based)

Columns specification 7,17,25,33

Has Header yes

Filler character *

Are records separated by new line yes

Character Set UTF-8

Conversion Preview:

ID Name	>	Column 1Co	>	lumn 2Co	>	Column4	>	Column5	>
XYZ12		Pants		V2c		V2d		12345678	

End Positions (One Based) Example

The 'end positions (1 based)' option indicates then the user needs to put in the numeric values of the last digit that should be considered part of each 'value'. This includes any symbols that are used as spaces. For example, if the value was 'trv2**tgh4****j5r' then the numeric value to put 'trv2**' in its own column would be 6 because each character counts individually.

1. **Columns are Specified by** is set to **end positions (1 based)**.
2. **Column specification** is set to **8,18,26,34**.
3. **Has Header** is set to **Yes**.
4. **Are records separated by new lines** is set to **Yes**.
5. **Character set** is set to **UTF-8**.

The Conversion Preview displays the filler characters.

Select Format

Format	FixedWidth
Converter for length separated formats where each line contains data about one object in fields separated by a fixed length	
Columns are specified by	end positions (1 based)
Columns specification	8,18,26,34
Has Header	yes
Filler character	
Are records separated by new line	yes
Character Set	windows-1252

Conversion Preview:

ID Name	>	Column 1Co	>	lumn 2Co	>	Column4	>	Column5	>
XYZ12***		Pants*****		V2c*****		V2d*****		12345678	
< >									

6. **Filler character** is set to * (which is being used to take up unneeded space between values)

The Conversion Preview no longer displays the filler characters.

Select Format

Format	FixedWidth
Converter for length separated formats where each line contains data about one object in fields separated by a fixed length	
Columns are specified by	end positions (1 based)
Columns specification	8,18,26,34
Has Header	yes
Filler character	*
Are records separated by new line	yes
Character Set	UTF-8

Conversion Preview:

ID Name	>	Column 1Co	>	lumn 2Co	>	Column4	>	Column5	>
XYZ12		Pants		V2c		V2d		12345678	
< >									

Width Example

The 'width' option indicates the number of characters in each section, excluding the last section. Each new section starts with 'one' as the value of the character.

For example, for 'trv2**tgh4****j5r' because there are 6 characters in the first section, '6' is put first. There are 8 in the next section, and the remaining numbers are in the third section, which is not counted. Thus the numbers entered would be 6,8.

1. **Columns are Specified by** is set to **width**.
2. **Column specification** is set to **8,10,8,8**.
3. **Has Header** is set to **Yes**.
4. **Are records separated by new lines** is set to **Yes**.
5. **Character set** is set to **UTF-8**.

The Conversion Preview displays the filler characters.

Select Format

Format FixedWidth

Converter for length separated formats where each line contains data about one object in fields separated by a fixed length

Columns are specified by width

Columns specification 8,10,8,8

Has Header yes

Filler character

Are records separated by new line yes

Character Set windows-1252

Conversion Preview:

ID Name	>	Column 1Co	>	lumn 2Co	>	Column4	>	Column5	>
XYZ12***		Pants*****		V2c*****		V2d*****		12345678	

<
>

6. **Filler character** is set to * (which is being used to take up unneeded space between values)

The Conversion Preview no longer displays the filler characters.

Select Format

Format FixedWidth

Converter for length separated formats where each line contains data about one object in fields separated by a fixed length

Columns are specified by width

Columns specification8,10,8,8

Has Headeryes

Filler character*

Are records separated by new lineyes

Character Setwindows-1252

Conversion Preview:

ID Name	>	Column 1Co	>	lumn 2Co	>	Column4	>	Column5	>
XYZ12		Pants		V2c		V2d		12345678	
< >									

Flatplan Excel Format

Two Excel formats are available for the export and import of Flatplans and Flatplanner publications: **Flatplan Excel** and **Publication Excel**.

Format Availability

Flatplan Excel is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**
- Export Manager - see **Creating a Data Export**

Mapping

Mapping data is not required and the Map Data step is disabled.

Inbound Data

The Flatplan Excel format is displayed as 'Flatplanner' format for inbound data.

Import Manager

Import Manager

×

Steps

1. Select Configuration

2. Select Data Source

3. Select Format

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Select Format

Format Flatplanner

Imports an Flatplanner Excel files

Conversion Preview:

Publicatio... >	Publicatio... >	Publicatio... >	RuleSettings >	VersionName >	VersionCo... >	▲
Acme Pet Su...	Default publi...	107822				
				English US	Context1	▼
						◀ ▶

IIEP

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint

2. Choose Receiver

3. Configure Endpoint

4. Configure PreProcessor

5. Configure Processing Engine

5.1. Select Sample File

5.2. Select Data Source

5.3. Select Format

5.4. Map Data

5.5. Identify Objects

5.6. Identify Destination

5.7. Select Business Rules

5.8. Advanced Settings

6. Configure PostProcessor

7. Schedule Endpoint

8. Configure Error Reporter

Configure Processing Engine : Select Format

Format Flatplanner

Imports an Flatplanner Excel files

Conversion Preview:

Publicatio... >	Publicatio... >	Publicatio... >	RuleSettings >	VersionName >	VersionCo... >
Acme Pet Su...	Default publi...	107822		English US	Context1

Back

Next

Finish

Cancel

Outbound Data

For step-by-step instructions on how to export a Publication Excel or Flatplan Excel spreadsheet from a Flatplanner publication in STEP, see **Exporting a Publication Excel Sheet** in the **STEP'n'design** documentation.

Export Manager

After selecting a publication on Tree, right-click and choose **Export Data Below** to see the Flatplan Excel format option in Export Manager.

Export Manager

Steps

1. Select Configuration

2. Select Objects

3. Select Format

4. Map Data

5. Advanced

6. Select Delivery Method

Select Objects

ID	Name	Object Type	Version	Path
205913	Acme Tools	Publication	English US	Publications/Standard Publications/Acme Tools

Add Objects

Export Manager

Steps

1. Select Configuration

2. Select Objects

3. Select Format

4. Map Data

5. Advanced

6. Select Delivery Method

Select Format

Flatplan Excel

Exports Publishing hierarchy to excel format.

Back

Next

Finish

Cancel

Export: Publication Objects

Next

Finish

Cancel

Generic XML Format

Generic XML is an XML-based language used to specify how data is extracted from or written to an XML document during import or export. In STEP, the Generic XML format allows you to import data from and export data to a variety of different XML formats without the need for extensions or customizations. STEP uses a template and different processing instructions for inbound and outbound Generic XML data, as defined below.

Limitations

When using the Generic XML functionality, be mindful of these limitations:

- Only products, entities, classifications, assets, attribute definitions, and LOV definitions can be imported and exported with Generic XML.
- The Generic XML import functionality cannot handle nested records.
- Neither data nor objects can be deleted when importing with Generic XML.

Format Availability

Generic XML is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**
- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

Inbound Data

The XML file is parsed via a template that extracts data and represents it in a tabular format. This requires that you have a representative source file, namely, one where all the different elements that can occur, and that you want to handle, are represented.

The following is an example of a Generic XML template:

```
<Products>
  <Product>
    <Name/>
  </Product>
</Products>
```

In the example, nodes are matched with the tag <Products> on the first level of the document. Below <Products>, the template matches nodes with the tag name <Product>.

For details about the available processing instructions, see the **Generic XML Inbound Processing Instructions** documentation.

For details about mapping the processing instructions to STEP data, see **Inbound Map Data Options** documentation.

Import Manager

Select Format

Format

Generic XML

Converter for a generic XML format described by a template

Sample



<Item>
<?Record?>
<EAN><?Source?></EAN>
<PrimarySpecs>
<ProductInformation>
<General>

Conversion Preview:

EAN	>	ProductNa...	>	Consumer...	>	AvailableF...	>	Protocol	>	Color.Value	>	Batte
2700524977...		AC-UZ444		Active 3D Gl...		2015-01-01 ...		Acme;Sony;...		Gray		C2
< >												

IIEP

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
5. **Configure Processing Engine**
 - 5.1. Select Sample File
 - 5.2. Select Data Source
 - 5.3. **Select Format**
 - 5.4. Map Data
 - 5.5. Identify Objects
 - 5.6. Identify Destination
 - 5.7. Select Business Rules
 - 5.8. Advanced Settings
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Processing Engine : Select Format

Format

Generic XML

Converter for a generic XML format described by a template

Sample

```

<Products>
  <Product>
    <?Record?>
    <Name><?Source?></Name>
    <Market>
      <?Repeated?>
      <Priority><?Source?></Priority>
      <Attributes>
        <Attribute ID="[?SourceID?]" UnitID="[?Source?]">
        <?Repeated?>

```

Conversion Preview:

Name	Colors	USA.Priority	USA.42.V...	USA.42.U...	USA.84.V...	USA.84.U...
Pilot Pen	Red;Green;B...	Low	3.068	Pounds	12.499	Dollars
Parker Pen	Gold;Silver	Low	3.728	Pounds	24.199	Dollars

Back

Next

Finish

Cancel

Outbound Data

For a Generic XML export, a template including instructions and placeholders determines how data should be handled, then STEP data is mapped into the XML template. For details, see **Outbound Map Data Options** documentation.

Since no undo functionality is available in the Template field, it is good practice to use an external XML editor program for creating and editing a template. To test a template's validity for output, paste the XML into STEP.

For details about the available processing instructions, some of which are only available in an OIEP, see the **Generic XML Outbound Processing Instructions** documentation.

Export Manager

Export Manager

Steps

1. Select Configuration

2. Select Objects

3. **Select Format**

4. Map Data

5. Advanced

6. Select Delivery Method

Select Format

Generic XML

Converts to a generic XML format based on a sample.

Sample

<attribute>
<?Record?>
<ID><?Target?></ID>
<Info>
<Type><?Target?></Type>
<MultiValue><?Target?></MultiValue>
<FullText><?Target?></FullText>

DocType

Allow empty tags

No

Back

Next

Finish

Cancel

OIEP

DimDepLOV - Configuration

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

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor
Item, Sales Item (Create, Modify, Del...	Generic XML (2 mappings)	None

Add configuration

Delivery Method

Select format

Format | Mapping | Advanced

Generic XML

Converts to a generic XML format based on a sample.

Sample

```
<attribute>
  <?Record?>
  <ID><?Target?></ID>
  <Info>
    <Type><?Target?></Type>
    <MultiValue><?Target?></MultiValue>
```

DocType

Allow empty tags No

OK Cancel

Generic XML Inbound Processing Instructions

Descriptions and examples of the available processing instructions used by STEP within Generic XML are covered in the following sections:

- Record Instruction in Generic XML
- Source Instruction in Generic XML
- MultiSource Instruction in Generic XML
- SourceID Instruction in Generic XML
- Repeated Instruction in Generic XML
- DimensionPointID Instruction

For examples of an input document, template and the conversion preview, see the following sections:

- Generic XML Import Simple Example
- Generic XML Import Advanced Example

Note: Only instructions included in both the document and the generic template are processed. Instructions not defined in the template are ignored.

DimensionPointID Instruction in Generic XML

Values in generic XML for the same attribute in different languages could be represented using the same ID with an additional attribute stating the language. This normally causes a name collision in generic XML. However, generic XML can import such dimension dependent data using the `<?DimensionPointID?>` instruction. This instruction can be used in a `?Repeated?` block in combination with `?SourceID?` or alone. When the `?DimensionPointID?` instruction is used, the name of the field extracted will be appended with the value of the `?DimensionPointID?` in square brackets, thus avoiding the name collision.

It is important to note that the use of the `?DimensionPointID?` instruction, does not actually import into the specified dimension points - this has to be set up using transformations in the Map Data step. Please see the **Inbound Map Data - Map** topic in the **Data Format** documentation. For this reason, the values appended to the column names using the `?DimensionPointID?` do not have to match the Dimension Point IDs, as this can be handled in the Map Data step of the import.

The following example shows how language dependent data can be extracted using generic XML. Note that the ID of the attribute is the same value, 52.

```
<Products>
  <Product ID="EXA-5002-1004">
    <Attributes>
      <Attribute>
        <ID>52</ID>
        <Lang>Eng</Lang>
        <Value>Some description text</Value>
      </Attribute>
      <Attribute>
        <ID>52</ID>
        <Lang>De</Lang>
        <Value>Einige beschreibenden Text</Value>
      </Attribute>
    </Attributes>
  </Product>
</Products>
```

The following Generic XML template extracts the sourceID from a node value, and appends the value inside the `<Lang>` tags in square brackets, e.g. '52[Eng]' and '52[De]' so they can be mapped to the respective languages.

```
<Products>
  <Product ID="[/Source?]">
    <?Record?>
      <Attributes>
```



```

<Attribute>
  <?Repeated?>
  <ID><?SourceID?></ID>
  <Lang><?DimensionPointID?></Lang>
  <Value><?Source?></Value>
</Attribute>
</Attributes>
</Product>
</Products>

```

MultiSource Instruction in Generic XML

To extract the value of a repeating XML element as a multi-value field that can be mapped to a multi-valued attribute / reference in STEP, place a `<?MultiSource [OptionalIdentifier]?>` instruction in either the node or the attribute of the `<?Record?>` scope:

```
<Products>
<Product ID=" [?Source?] ">
<?Record?>
<KeyWords>
<Keyword><?MultiSource?></Keyword>
</KeyWords>
</Product>
</Products>
```

Within the same scope, multiple nodes / attributes matching a `<?MultiSource [OptionalIdentifier]?>` declaration will contribute to the same multi-value field. The input document below, uses the template above.

```
<Products>
<Product ID="42">
<KeyWords>
<Keyword>Red</Keyword>
<Keyword>Green</Keyword>
<Keyword>Blue</Keyword>
</KeyWords>
</Product>
</Products>
```

It produces the fields 'ID' and 'Keyword' with values ID=42 and Keyword=Red|Green|Blue. Note that container-node, 'KeyWords' in this case, is not a requirement to extract a multi-value field. A document with the 'Keyword' nodes directly under the 'Product' node could be processed by using the following template:

```
<Products>
<Product ID=" [?Source?] ">
<?Record?>
<Keyword><?MultiSource Keywords?></Keyword>
</Product>
</Products>
```

Like the `<?Source?>` instruction, the `<?MultiSource?>` instruction also supports explicitly specifying a field name using an optional identifier. For more information, see the **Preventing Duplicate Field Names** section of the **Source Instruction in Generic XML** documentation.

Record Instruction in Generic XML

The template must declare which element corresponds to a single record (product, entity, etc.). This is done using the `<?Record?>` processing instruction, which applies to the XML node that surrounds it. A template must contain exactly one `<?Record?>` processing instruction.

```
<Products>
  <Product>
    <?Record?>
      <Name/>
    </Product>
  </Products>
```

In this example, STEP will extract data for each product object inside the `<Product>` and `<Products>` elements, regardless of the number that exist. The product node is the record node of the template. The record node must not be declared as a repeated node. Although this is a valid template, no data is being supplied by this template.

Repeated Instruction in Generic XML

Additional declarations are required when extracting data from repeated structures below the record node. Declaration of a repeated node happens with the `<?Repeated [OptionalSpaceSeparatedFilters]?>` processing instruction. The `<?Repeated?>` instruction must be placed inside `<?Record?>` scope. For example:

```
<Products>
  <Product ID="EXA-5002-1004">
    <Values>
      <Value AttributeID="52">Some description text</Value>
      <Value AttributeID="77">53 kg</Value>
    </Values>
  </Product>
</Products>
```

Note that the **Value** node is repeated. To extract data from both occurrences, the template must declare that a node is a repeated node, which means that a set of fields has to be generated for each node encountered. The template also has to declare an ID source on or below the repeated node.

The following template could be used to extract the data from the above Generic XML:

```
<Products>
  <Product ID="[?Source ID?]">
    <?Record?>
      <Values>
        <Value AttributeID="[?SourceID?]"><?Repeated?><?Source?></Value>
      </Values>
    </Product>
  </Products>
```

The ID source is used to map each occurrence of a repeated node to a specific set of fields of the form `[Identifier].[SourceTagName]`. This is required to ensure that related repeated nodes map to the same fields across different records.

By default, a repeated scope produces a set of fields for each ID Source value encountered in the input document. Using the 'optional space separated filters' allows you to select values from specific ID keys or to guard against additional columns being generated as input documents evolve in the future.

```
<ARTICLE>
  <ARTICLE_PRICE_DETAILS>
    <DATETIME type="valid_start_date">
      <DATE>2010-01-01</DATE>
    </DATETIME>
```

```
<PRODUCT_PRICE price_type="net_list">
  <PRICE_AMOUNT>11.08</PRICE_AMOUNT>
  <PRICE_CURRENCY>EUR</PRICE_CURRENCY>
  <TAX>0.19</TAX>
  <LOWER_BOUND>1</LOWER_BOUND>
</PRODUCT_PRICE>
<PRODUCT_PRICE price_type="nrp">
  <PRICE_AMOUNT>23.05</PRICE_AMOUNT>
  <PRICE_CURRENCY>EUR</PRICE_CURRENCY>
  <TAX>0.19</TAX>
  <LOWER_BOUND>1</LOWER_BOUND>
</PRODUCT_PRICE>
<PRODUCT_PRICE price_type="employee">
  <PRICE_AMOUNT>8.52</PRICE_AMOUNT>
  <PRICE_CURRENCY>EUR</PRICE_CURRENCY>
  <TAX>0.19</TAX>
  <LOWER_BOUND>1</LOWER_BOUND>
</PRODUCT_PRICE>
</ARTICLE_PRICE_DETAILS>
</ARTICLE>
```

The price and currency information can be extracted for the net_list and nrp price types only by following the proceeding steps and using the following template:

1. Copy the above Generic XML to a new file, and save as an XML file
2. Open Import Manager, select the file as the Data Source Filename, and click 'Next'
3. The user can enter/ paste the template below in the Import Manager, and a conversion preview can be generated by clicking the refresh button, .

```
<ARTICLE>
  <?Record?>
  <ARTICLE_PRICE_DETAILS>
    <PRODUCT_PRICE price_type="[?SourceID?]">
      <?Repeated net_list nrp?>
      <PRICE_AMOUNT><?Source Amount?></PRICE_AMOUNT>
      <PRICE_CURRENCY><?Source Currency?></PRICE_CURRENCY>
    </PRODUCT_PRICE>
  </ARTICLE_PRICE_DETAILS>
```

</ARTICLE>

Select Format

Format

Generic XML

▼

Converter for a generic XML format described by a template

Sample

```

<ARTICLE>
  <?Record?>
  <ARTICLE_PRICE_DETAILS>
    <PRODUCT_PRICE price_type="[?SourceID?]">
      <?Repeated net_list nrp?>
        <PRICE_AMOUNT><?Source Amount?></PRICE_AMOUNT>
        <PRICE_CURRENCY><?Source Currency?></PRICE_CURRENCY>
      </PRODUCT_PRICE>
    </ARTICLE_PRICE_DETAILS>
  </ARTICLE>

```

Conversion Preview:

net_list.Amount	>	net_list.Currency	>	nrp.Amount	>	nrp.Currency	>
11.08		EUR		23.05		EUR	

This produces the fields:

- net_list.Amount = 11.08
- net.list.Currency = EUR
- nrp.Amount = 23.05
- nrp.Currency = EUR

The <?Repeated?> instruction can take an arbitrary number of filter terms separated by white-space characters. When no terms are defined, the filtering functionality is disabled.

Source Instruction in Generic XML

To extract the content of an XML element as a field, place a `<?Source [OptionalIdentifier]?>` processing instruction within the `<?Record?>` scope of the template. By default, the heading (column) name of a field is the name of the surrounding parent node.

```
<Products>
  <Product>
    <?Record?>
      <Name><?Source?></Name>
      <Text><?Source Description?></Text>
    </Product>
  </Products>
```

In the above example, each product hierarchy encountered exports a row with data for the columns Name and Description.

To make the mapping step easier, use your own identifiers by defining an optional identifier. As shown in the `<Text>` instruction in the above example, the Text field will display with a heading of 'Description' since the optional identifier was used within the source tag. The optional identifier displays in the Map Data step.

You can also extract the value of an attribute by using the `[?Source [Optional Identifier]?]` instruction as follows:

```
<Products>
  <Product Name="[?Source?]" Text="[?Source Description?]">
    <?Record?>
  </Product>
</Products>
```

Preventing Duplicate Field Names

Fields extracted using the `<?Source?>` and `<?MultiSource?>` instructions can sometimes have the same names, either because of identical tag or attribute names, or because of duplication of names explicitly chosen by the template creator. In this situation, the Import Manager ensures that all field names are unique by adding a post-fix of the form (n), where n is the first integer ≥ 2 for which the combined field name is not already in use.

If a repeated scope contains multiple `<?Source?>` or `<?MultiSource?>` instructions, and one of them results in a name duplication, only that field will be renamed.

Note: Resolve name collisions by providing explicit, non-duplicating names.

SourceID Instruction in Generic XML

?SourceID? instructions must be placed inside repeated scope. Only one ?SourceID? is allowed for each scope.

- <?SourceID?> is used to extract the identifier for a repeated element when the identifier is a nested element.
- [?SourceID?] is used to extract the identifier for a repeated element when the identifier is an attribute.

The following is an example of an input document that requires the repeated node and ID source declarations to happen on different nodes:

```
<Products>
  <Product ID="EXA-5002-1004">
    <Attributes>
      <Attribute>
        <ID>52</ID>
        <Value>Some description text</Value>
      </Attribute>
      <Attribute>
        <ID>77</ID>
        <Value>53 kg</Value>
      </Attribute>
    </Attributes>
  </Product>
</Products>
```

The following is a Generic XML template for processing the input document:

```
<Products>
  <Product ID="[?Source?]">
    <?Record?>
      <Values>
        <Value AttributeID="[?SourceID?]">
          <?Repeated?>
            <?Source?>
          </Value>
        </Values>
      </Product>
    </Products>
```


When used on the input document, this template outputs the fields 'ID', 'Value-52', 'Value-77'. In this case, the subtree from 'Value' and down is said to be in repeated scope.

The Generic XML template this time extracts the sourceID from a node value. Follow the proceeding steps and use the following template to extract the input document:

1. Copy the above Generic XML to a new file, and save as an XML file.
2. Open Import Manager, select the file as the Data Source Filename, and click 'Next'.
3. Enter or paste the template below in the Import Manager, and generate a conversion preview by clicking the refresh button, .

```
<Products>
  <Product ID="[?Source?]">
    <?Record?>
      <Attributes>
        <Attribute>
          <?Repeated?>
            <ID><?SourceID?></ID>
            <Value><?Source?></Value>
          </Attribute>
        </Attributes>
      </Product>
    </Products>
```

For the input document, this template generates the fields **ID**, **52.Value** and **77.Value**.

Select Format

Format Generic XML

Converter for a generic XML format described by a template

Sample



```
<?Record?>
<Attributes>
  <Attribute>
    <?Repeated?>
    <ID><?SourceID?></ID>
    <Value><?Source?></Value>
  </Attribute>
</Attributes>
</Product>
</Products>
```

Conversion Preview:

ID	52.Value	77.Value
EXA-5002-1004	Some description text	53 kg

Generic XML Import Simple Example

The following is an example of an import document and template that uses most of the features present in a Generic XML template.

Input Document

Copy the following text to a new file and save as an XML file. Open Import Manager and select the file as the Data Source Filename and click 'Next'.

```
<Products>
  <Product>
    <Name>Pilot Pen</Name>
    <Market>
      <Priority>Low</Priority>
      <Attributes>
        <Attribute ID="42" UnitID="Pounds">3.068</Attribute>
        <Attribute ID="84" UnitID="Dollars">12.499</Attribute>
      </Attributes>
      <MarketID>USA</MarketID>
    </Market>
    <Market>
      <Priority>High</Priority>
      <Attributes>
        <Attribute ID="42" UnitID="Kg">1534</Attribute>
        <Attribute ID="84" UnitID="Euro">22.995</Attribute>
      </Attributes>
      <MarketID>EU</MarketID>
    </Market>
    <Colors>
      <Color>Red</Color>
      <Color>Green</Color>
      <Color>Blue</Color>
    </Colors>
  </Product>
  <Product>
    <Name>Parker Pen</Name>
    <Market>
```

```

        <Priority>Low</Priority>
    <Attributes>
        <Attribute ID="42" UnitID="Pounds">3.728</Attribute>
        <Attribute ID="84" UnitID="Dollars">24.199</Attribute>
    </Attributes>
    <MarketID>USA</MarketID>
</Market>
<Market>
    <Priority>Extreme</Priority>
    <Attributes>
        <Attribute ID="42" UnitID="Kg">1864</Attribute>
        <Attribute ID="84" UnitID="Euro">29.999</Attribute>
    </Attributes>
    <MarketID>EU</MarketID>
</Market>
<Colors>
    <Color>Gold</Color>
    <Color>Silver</Color>
</Colors>
</Product>
</Products>

```

Template

Copy the following text and paste it in the Import Manager Select Format step, Sample field.

```

<Products>
    <Product>
        <?Record?>
            <Name><?Source?></Name>
            <Market>
                <?Repeated?>
                    <Priority><?Source?></Priority>
                    <Attributes>
                        <Attribute ID="[?SourceID?]" UnitID="[?Source?]">
                            <?Repeated?>
                                <?Source Value?>

```

```

    </Attribute>
  </Attributes>
<MarketID><?SourceID?></MarketID>
</Market>
<Colors>
  <Color><?MultiSource Colors?></Color>
</Colors>
</Product>
</Products>
```

Conversion Preview

Click the Sample reload button . The Conversion Preview area displays the results of the input document against the provided template. The following Map Data step presents data in the same format that is used for and CSV / Excel import. Additionally, Generic XML data must be mapped and can be transformed. To learn more about inbound transformations, see the **Inbound Map Data - Transform** documentation.

Select Format

FormatGeneric XML

Converter for a generic XML format described by a template

Sample



<Item>
<?Record?>
<EAN><?Source?></EAN>
<PrimarySpecs>
<ProductInformation>
<General>

Conversion Preview:

EAN	>	ProductNa...	>	Consumer...	>	AvailableF...	>	Protocol	>	Color.Value	>	Batte
2700524977...		AC-UZ444		Active 3D Gl...		2015-01-01 ...		Acme;Sony;...		Gray		C2
<div>												

© Stibo Systems - Confidential - STEP Trailblazer 8.3

201

Generic XML Import Advanced Example

The following is an example of a more complex import document and template that uses most of the features present in a Generic XML template.

Input Document

Copy the following text to a new file and save as an XML file. Open Import Manager and select the file as the Data Source Filename and click 'Next'. Note that any number of additional items could be included by repeating the data within the Item tag.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemLoad>
  <Items>
    <Item>
      <EAN>2700524977488</EAN>
      <PrimarySpecs>
        <ProductInformation>
          <General>
            <ProductName>AC-UZ444</ProductName>
            <ConsumerShortDescription>Active 3D
Glasses</ConsumerShortDescription>
            <AvailableFrom>2015-01-01 00:00:00</AvailableFrom>
          </General>
          <Variants>
            <Name>Color</Name>
            <Value>Gray</Value>
          </Variants>
          <Variants>
            <Name>Battery</Name>
            <Value>C2</Value>
          </Variants>
          <Manuals>
            <Manual type="ManualDE">Manual-9</Manual>
            <Manual type="ManualEN">Manual-10</Manual>
          </Manuals>
        </ProductInformation>
      </PrimarySpecs>
    </Item>
  </Items>
</ItemLoad>
```

```

    <SyncProtocols>
      <Protocol>Acme</Protocol>
      <Protocol>Sony</Protocol>
      <Protocol>Sharp</Protocol>
    </SyncProtocols>
  </Item>
</Items>
</ItemLoad>

```

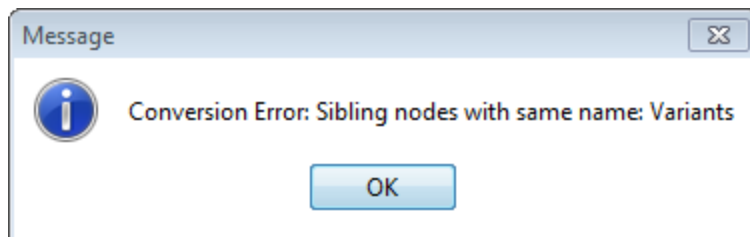
Creating a Template

To create a template from an import file, copy the file without the XML declaration (the first line that indicates the XML version), and paste into the Sample field on the Import Manager Select Format step. Then, remove any duplicate elements. Since there is only a single Item element in the example above, it can be pasted as is.

Note: When creating a template from a source file, ensure all of the elements that can occur, and that must be handled, are represented. Only instructions included in both the document and the generic template are processed. Instructions not defined in the template are ignored.

Remove Redundant Elements

Click the Sample reload button  and a message is displayed about sibling nodes with the same name:



Remove of repetitions of the 'Variants', 'Manual', and 'Protocol' elements, until only one of each remains. The Sample field now displays the following:

```

<ItemLoad>
  <Items>
    <Item>
      <EAN>2700524977488</EAN>
      <PrimarySpecs>
        <ProductInformation>
          <General>
            <ProductName>AC-UZ444</ProductName>

```

```

        <ConsumerShortDescription>Active 3D
Glasses</ConsumerShortDescription>

        <AvailableFrom>2015-01-01 00:00:00</AvailableFrom>

    </General>

    <Variants>

        <Name>Color</Name>

        <Value>Gray</Value>

    </Variants>

    <Manuals>

        <Manual type="ManualDE">Manual-9</Manual>

    </Manuals>

</ProductInformation>

</PrimarySpecs>

<SyncProtocols>

    <Protocol>Acme</Protocol>

</SyncProtocols>

</Item>

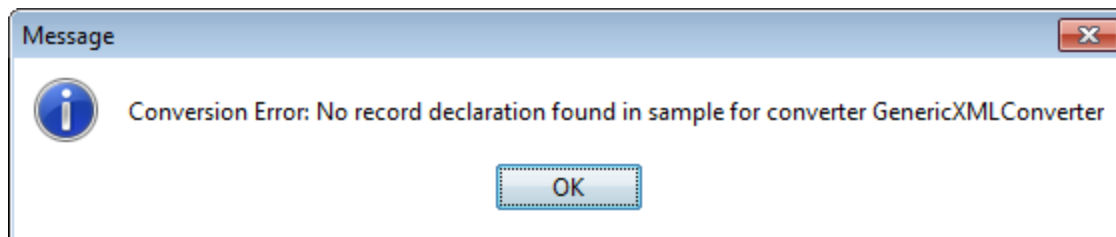
</Items>

</ItemLoad>

```

Add Record and Source Instructions

Click the Sample reload button  and a message is displayed about a missing record declaration:



Use the `<?Record?>` and `<?Source?>` instructions to handle all the cases where you simply need to extract one value between a start and end tag.

```

<ItemLoad>

    <Items>

        <Item>

            <?Record?>

            <EAN><?Source?></EAN>

```



```

    <PrimarySpecs>
      <ProductInformation>
        <General>
          <ProductName><?Source?></ProductName>

<ConsumerShortDescription><?Source?></ConsumerShortDescription>
          <AvailableFrom><?Source?></AvailableFrom>
        </General>
        <Variants>
          <Name>Color</Name>
          <Value>Gray</Value>
        </Variants>
        <Manuals>
          <Manual type="ManualDE">Manual-9</Manual>
        </Manuals>
      </ProductInformation>
    </PrimarySpecs>
    <SyncProtocols>
      <Protocol>Acme</Protocol>
    </SyncProtocols>
  </Item>
</Items>
</ItemLoad>

```

Add Repeated Instruction

To see how each option affects the outcome, make the following updates in the Sample template and then click the Sample reload button  to update the Conversion Preview.

For the 'Variants' element, if we insert the <?Source?> instruction in the 'Value' element as shown below, the result will be the value for the first repetition, 'Gray'.

```

<Variants>
  <Name>Color</Name>
  <Value><?Source?></Value>
</Variants>

```

Conversion Preview:									
EAN	>	ProductName	>	ConsumerShort...	>	AvailableFrom	>	Value	>
2700524977488		AC-UZ444		Active 3D Glasses		2015-01-01 00:00:00		Gray	

To get values from both 'Variants' elements, we have two options: MultiSource and Source / Repeated / SourceID.

To get the Value from each repetition and be able to map them all to a multi valued attribute / multiple reference / links targets, we can use the `<?MultiSource?>` instruction.

```
<Variants>
  <Name>Color</Name>
  <Value><?MultiSource?></Value>
</Variants>
```

Conversion Preview:									
EAN	>	ProductName	>	ConsumerShort...	>	AvailableFrom	>	Value	>
2700524977488		AC-UZ444		Active 3D Glasses		2015-01-01 00:00:00		Gray;C2	

If the data should be mapped to different attributes, as is the case in our example, where the Name value of each repetition identifies the attribute, use these three instructions:

- `<?Source?>`
- one `<?Repeated?>` to indicate that Variants is a repeated element
- one `<?SourceID?>` to indicate that the Name value should be used as an identifier for each repetition

```
<Variants>
  <?Repeated?>
  <Name><?SourceID?></Name>
  <Value><?Source?></Value>
</Variants>
```

Conversion Preview:											
EAN	>	ProductName	>	ConsumerSh...	>	AvailableFrom	>	Color.Value	>	Battery.Value	>
2700524977488		AC-UZ444		Active 3D Glasses		2015-01-01 00...		Gray		C2	

The Import Manager displays the column header for each element using the pattern [Identifier].[SourceTagName].

Add Repeated Instruction With a Filter

To see how each option affects the outcome, make the following updates in the Sample template and then click the Sample reload button  to update the Conversion Preview.

The repeated Manual element is handled differently since the type of identifier is an attribute.

Again, start by placing the `<?Source?>` instruction to the value:

```
<Manuals>
  <Manual type="ManualDE"><?Source?></Manual>
</Manuals>
```

Conversion Preview:

EAN	>	ProductName	>	ConsumerShortDes...	>	AvailableFrom	>	ManualDE	>	Color.Value	>	Battery.Value	>
2700524977488		AC-UZ444		Active 3D Glasses		2015-01-01 00:00:00		Manual-9		Gray		C2	

Again, to get values for all repetitions, use the `<?Repeated?>` instruction and also add an identifier instruction. This time, since the identifier is an attribute value, use the square bracket version `[?SourceID?]`.

```
<Manuals>
  <Manual type="[?SourceID?]"><?Repeated?><?Source?></Manual>
</Manuals>
```

Conversion Preview:

EAN	>	ProductName	>	ConsumerShort...	>	AvailableFrom	>	Color.Value	>	Battery.Value	>	ManualDE.Manual	>	ManualEN.Manual	>
2700524977488		AC-UZ444		Active 3D Glasses		2015-01-01 00:00:00		Gray		C2		Manual-9		Manual-10	

To filter repeated elements so that only elements with specific identifiers are considered, add the identifier to the `<?Repeated?>` instruction. To get only ManualEN data, update the template as follows.

```
<Manuals>
  <Manual type="[?SourceID?]"><?Repeated
ManualEN?><?Source?></Manual>
</Manuals>
```

Conversion Preview:

EAN	>	ProductName	>	ConsumerShortDescription	>	AvailableFrom	>	Color.Value	>	Battery.Value	>	ManualEN.Manual	>
2700524977488		AC-UZ444		Active 3D Glasses		2015-01-01 00:00:00		Gray		C2		Manual-10	

Add MultiSource Instruction

For the repeated Protocol element there is no identifier, so it makes sense to use the `<?MultiSource?>` processing instruction. When this instruction is used, the `<?Repeated?>` instruction is not required.

```
<SyncProtocols>
  <Protocol><?MultiSource?></Protocol>
</SyncProtocols>
```

Conversion Preview:							
EAN	ProductName	ConsumerShortDescription	AvailableFrom	Protocol	Color.Value	Battery.Value	ManualEN.Manual
2700524977488	AC-UZ444	Active 3D Glasses	2015-01-01 00:00:00	Acme;Sony;Sharp	Gray	C2	Manual-10

Now that all instructions have been added, the template is below.

```
<ItemLoad>
  <Items>
    <Item>
      <?Record?>
      <EAN><?Source?></EAN>
      <PrimarySpecs>
        <ProductInformation>
          <General>
            <ProductName><?Source?></ProductName>

            <ConsumerShortDescription><?Source?></ConsumerShortDescription>

            <AvailableFrom><?Source?></AvailableFrom>
          </General>
          <Variants>
            <?Repeated?>
            <Name><?SourceID?></Name>
            <Value><?Source?></Value>
          </Variants>
          <Manuals>
            <Manual type="[?SourceID?]"><?Repeated?><?Source?></Manual>
          </Manuals>
        </ProductInformation>
      </PrimarySpecs>
    </Item>
  </Items>
</ItemLoad>
```

```
<Protocol><?MultiSource?></Protocol>

</SyncProtocols>

</Item>

</Items>

</ItemLoad>
```

Conversion Preview

Click the Sample reload button . The Conversion Preview area displays the results of the input document against the provided template.

Select Format

Format Generic XML

Converter for a generic XML format described by a template

Sample<ItemLoad><Items><Item><?Record?><EAN><?Source?></EAN><PrimarySpecs><ProductInformation><General><ProductName><?Source?></ProductName><ConsumerShortDescription><?Source?></ConsumerShortDescription><AvailableFrom><?Source?></AvailableFrom></General><Variants><?Repeated?><Name><?SourceID?></Name><Value><?Source?></Value></Variants><Manuals><Manual type="[?SourceID?]"><?Repeated?><?Source?></Manual></Manuals></ProductInformation></PrimarySpecs><SyncProtocols><Protocol><?MultiSource?></Protocol></SyncProtocols></Item></Items></ItemLoad>

Conversion Preview:

EAN	>	ProductName	>	ConsumerShortDescription	>	AvailableFrom	>	Protocol	>	Color.Value	>	Battery.Value	>	ManualDE.Manual	>	ManualEN.Manual	>
2700524977488		AC-U2444		Active 3D Glasses		2015-01-01 00:00:00		Acme;Sony;Sharp		Gray		C2		Manual-9		Manual-10	

Generic XML Outbound Processing Instructions

Descriptions and examples of the available processing instructions used by STEP within Generic XML are covered in the following sections:

- Allow Empty Tags Parameter in Generic XML
- Attributes Instruction in Generic XML
- Attribute Links and LOVs Instruction in Generic XML
- Attribute Links and Units Instruction in Generic XML
- DoNotEmit Instruction in Generic XML
- List of Values Object Type in Generic XML
- Multi-Valued Objects in Generic XML
- Multi-Cardinality Tags in Generic XML
- Nestable Data Paths in Generic XML
- RemoveIfEmpty Instruction in Generic XML
- Reporting Deleted Assets with a Generic XML OIEP
- Reporting Deleted Classifications with a Generic XML OIEP
- Reporting Deleted Entities with a Generic XML OIEP
- Reporting Deleted Product References with a Generic XML OIEP
- Single-Cardinality Tags in Generic XML
- Units with a Generic XML OIEP

For an example of selected objects, template, mapping, and the result, see the **Generic XML Sample Export** section.

Allow Empty Tags Parameter in Generic XML

The 'Allow empty tags' parameter determines if a mapped attribute tag is exported when no value is present. For example, consider an object with an attribute value that is then removed. Using this parameter in an event-based OIEP, the value deletion would cause an update event, and the output would include this attribute with an empty value.

If tags for parent attributes with children that do not contain values should be removed completely, see the **RemovelfEmpty Instruction in Generic XML** section.

Objects

The object selected for export includes attributes without values.

Product	Sub Products	References	Referenced By	Images & Documents	Commerci
Description					
Name	>	>	Value		
ID			20803		
Name			20803		
Object Type			Item		
Revision			0.21 Last edited by USER on Mon Oct 26 13:56:00 EDT 2015		
Approved			✓ Approved in Current Context on Mon Oct 26 13:56:09 EDT		
Path			Primary Product Hierarchy/Products/Apparel/Head Wear/Hats a		
UPC	abc		002567954136		
AllFields	123				
Category	fx		Classification 1 root Suppliers SuppliesAll Products 20803		
City	abc				
Status					

Template

Choose **Generic XML** format and provide a template in the Sample field.

```
<Products>
  <Product>
    <?Record?>
    <ProductID><?Target?></ProductID>
    <ProductUPC><?MultiTarget?></ProductUPC>
    <ProductAllFields><?MultiTarget?></ProductAllFields>
    <ProductCategory><?MultiTarget?></ProductCategory>
    <ProductCity><?MultiTarget?></ProductCity>
    <ProductStatus><?MultiTarget?></ProductStatus>
```

```
</Product>
```

```
</Products>
```

Set **Allow empty tags** parameter to **Yes** if tags without values should be output or set to **No** if empty tags should be eliminated.

Select Format

Generic XML

Converts to a generic XML format based on a sample.

Sample

```

<Products>
  <Product>
    <?Record?>
    <ProductID><?Target?></ProductID>
    <ProductUPC><?MultiTarget?></ProductUPC>
    <ProductAllFields><?MultiTarget?></ProductAllFields>
    <ProductCategory><?MultiTarget?></ProductCategory>
    <ProductCity><?MultiTarget?></ProductCity>
    <ProductStatus><?MultiTarget?></ProductStatus>
  </Product>
</Products>

```

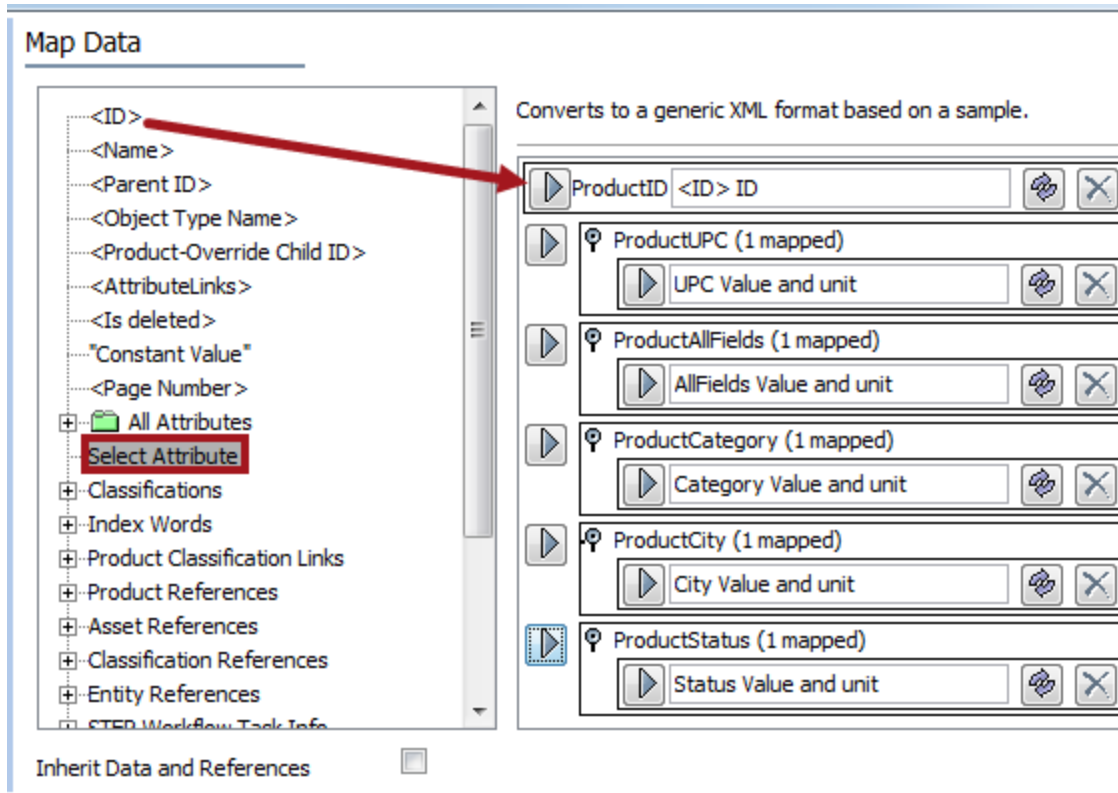
DocType

Allow empty tags

Yes

Mapping

Use the **Select Attribute** aspect for the attribute data targets after ID.



Results

With 'Allow empty tags' parameter = Yes, tags are exported for attributes with empty values.

```
<Products>
  <Product>
    <ProductID>20803</ProductID>
    <ProductUPC>002567954136</ProductUPC>
    <ProductAllFields></ProductAllFields>
    <ProductCategory>Classification 1 root | Suppliers | SuppliesAll | Products | 20803</ProductCategory>
    <ProductCity></ProductCity>
    <ProductStatus></ProductStatus>
  </Product>
</Products>
```

With 'Allow empty tags' parameter = No, only tags with values are exported.

```
<Products>
  <Product>
    <ProductID>20803</ProductID>
    <ProductUPC>002567954136</ProductUPC>
    <ProductCategory>Classification 1 root | Suppliers | SuppliesAll | Products | 20803</ProductCategory>
  </Product>
</Products>
```

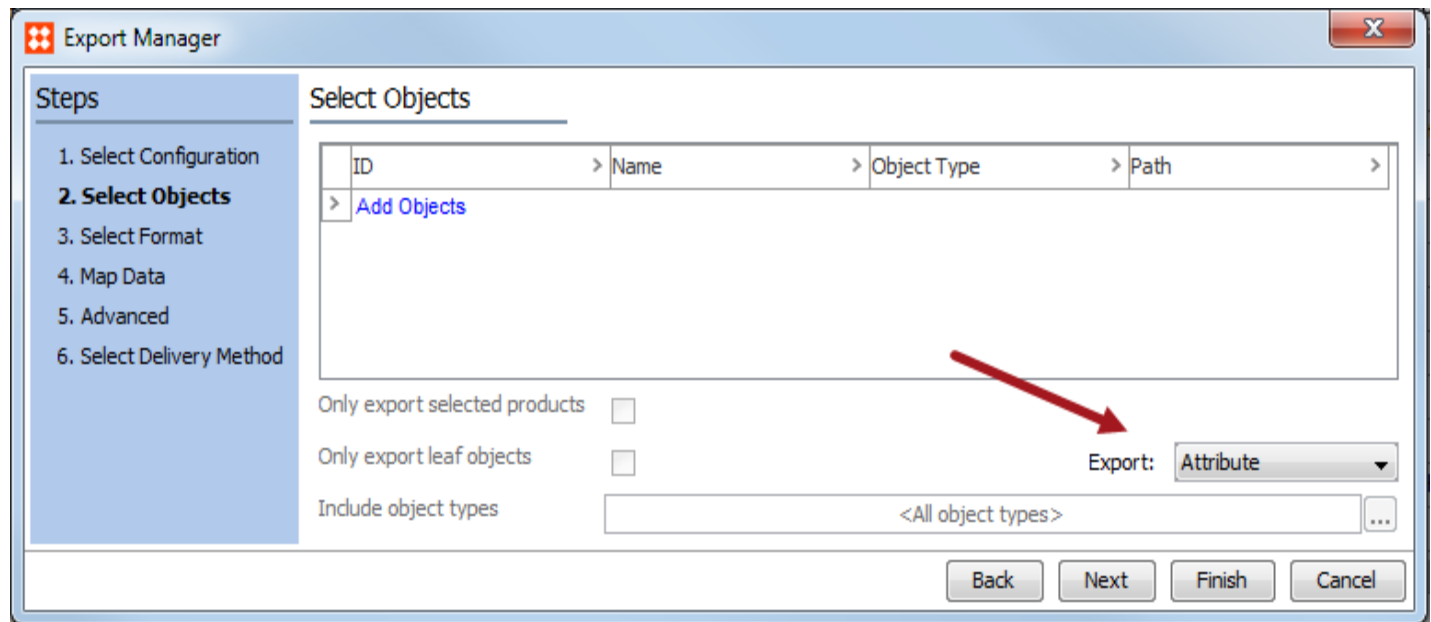
Attributes Instruction in Generic XML

The Attribute template and mapping make it possible to specify which attribute setup information is included when an attribute or attribute group is exported. This type of mapping and export is used when the intent is to export data for attributes themselves (e.g., ID, Name, Type, Validation, etc.), not attribute values.

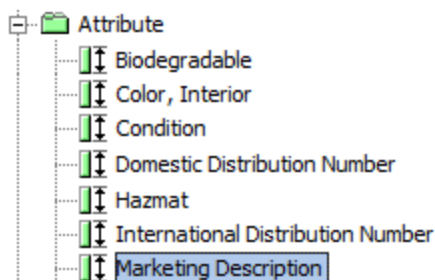
Follow the steps below to extract all available attribute data. If some data points on attributes are not required, they can be removed from the template and the mapping.

Objects

Choose 'Attribute' for the export.



Use the **Add Objects** link to select a single attribute, multiple attributes, or an attribute group to export.



Note: If exporting a group, data for each field in the template is evaluated for each child in the group.

Template

Choose the **Generic XML** format option. The following is an example of what could be provided in the Sample field. This template corresponds to the mapping options below. However, any element can be removed and/or renamed as needed to meet the export requirements.

```
<root>
  <attribute>
    <?Record?>
    <ID><?Target?></ID>
  <Info>
    <Type><?Target?></Type>
    <MultiValue><?Target?></MultiValue>
    <FullText><?Target?></FullText>
    <Calculated><?Target?></Calculated>
    <ExternalMaintained><?Target?></ExternalMaintained>
    <DimDepend><?Target?></DimDepend>
  </Info>
  <ValidationInfo>
    <BaseType><?Target?></BaseType>
    <MinVal><?Target?></MinVal>
    <MaxVal><?Target?></MaxVal>
    <MaxLength><?Target?></MaxLength>
    <InputMask><?Target?></InputMask>
    <LOVID><?Target?></LOVID>
  </ValidationInfo>
  <AttrGroupReference>
    <RefID><?Target?></RefID>
    <RefName><?Target?></RefName>
  </AttrGroupReference>
</attribute>
</root>
```

Mapping

The mapping screen auto-populates based on the provided template, but will display each field in the right panel as 'Nothing mapped'. Select the appropriate aspect in the left panel and use the arrow buttons to map it to a template field on the right. If all available fields are mapped, the end result will look like this:

Select format

Format Mapping Advanced

Attribute

<ID>
<Name>
Attribute Info
 <Attribute Type>
 <Attribute Multi Valued>
 <Attribute FullText>
 <Attribute Calculated>
 <Attribute Externally Maintained>
 <Attribute Dimension dependencies>
Attribute Validation Info
 <Attribute Validation Base Type>
 <Attribute Validation Minimum value>
 <Attribute Validation Maximum value>
 <Attribute Validation Maximum length>
 <Attribute Validation Input mask>
 <Attribute Validation LOV ID>
Attribute Group References
 <AttributeGroup Ref ID>
 <AttributeGroup Ref Name>
"Constant Value"
All Attributes
Select Attribute
Custom Attributes

Converts to a generic XML format based on a sample.

ID	<ID> ID		
Type	<Attribute Type> Type		
MultiValue	<Attribute Multi Valued> Multi Valued		
FullText	<Attribute FullText> FullText		
Calculated	<Attribute Calculated> Calculated		
ExternalMaintained	<Attribute Externally Maintained> Externally Maintained		
DimDepend	<Attribute Dimension dependencies> Dimension dependencies		
BaseType	<Attribute Validation Base Type> Base Type		
MinVal	<Attribute Validation Minimum value> Minimum value		
MaxVal	<Attribute Validation Maximum value> Maximum value		
MaxLength	<Attribute Validation Maximum length> Maximum length		
InputMask	<Attribute Validation Input mask> Input mask		
LOVID	<Attribute Validation LOV ID> LOV ID		
RefID	<AttributeGroup Ref ID> ID		
RefName	<AttributeGroup Ref Name> Name		

Inherit Data and References ☐

OK Cancel

Note: The Inherit Data and References checkbox option does not affect the Attributes instruction since it handles attribute definitions.

See the **Attribute Aspects in Generic XML** documentation for a description of each available data point.

Results

A sample output of the above mapping is shown below.

```

<root>
  <attribute>
    <ID>Marketing Description</ID>
    <Info>
      <Type>Property</Type>
      <MultiValue>>false</MultiValue>
      <FullText>>false</FullText>
      <Calculated>>true</Calculated>
      <ExternalMaintained>>false</ExternalMaintained>
      <DimDepend>Language; Country</DimDepend>
    </Info>
    <ValidationInfo>
      <BaseType>text</BaseType>
      <MinVal/>
      <MaxVal/>
      <MaxLength>100</MaxLength>
      <InputMask/>
      <LOVID/>
    </ValidationInfo>
    <AttrGroupReference>
      <RefID>AttributeGroup</RefID>
      <RefName>Attribute Group</RefName>
    </AttrGroupReference>
  </attribute>
</root>

```

Attribute Aspects in Generic XML

Aspects, also called data sources, are displayed when mapping Attributes for export. The description of each aspect is included below.

Select format

Format

Mapping

Advanced

Attribute

<ID>

<Name>

Attribute Info

Attribute Type>

Attribute Multi Valued>

Attribute FullText>

Attribute Calculated>

Attribute Externally Maintained>

Attribute Dimension dependencies>

Attribute Validation Info

Attribute Validation Base Type>

Attribute Validation Minimum value>

Attribute Validation Maximum value>

Attribute Validation Maximum length>

Attribute Validation Input mask>

Attribute Validation LOV ID>

Attribute Group References

AttributeGroup Ref ID>

AttributeGroup Ref Name>

"Constant Value"

All Attributes

Select Attribute

Custom Attributes

Converts to a generic XML format based on a sample.

ID <ID> ID

Type <Attribute Type> Type

MultiValue <Attribute Multi Valued> Multi Valued

FullText <Attribute FullText> FullText

Calculated <Attribute Calculated> Calculated

ExternalMaintained <Attribute Externally Maintained> Externally Maintained

DimDepend <Attribute Dimension dependencies> Dimension dependencies

BaseType <Attribute Validation Base Type> Base Type

MinVal <Attribute Validation Minimum value> Minimum value

MaxVal <Attribute Validation Maximum value> Maximum value

MaxLength <Attribute Validation Maximum length> Maximum length

InputMask <Attribute Validation Input mask> Input mask

LOVID <Attribute Validation LOV ID> LOV ID

RefID <AttributeGroup Ref ID> ID

RefName <AttributeGroup Ref Name> Name

Inherit Data and References

OK

Cancel

Aspect	Description
ID	Extracts only the ID of the attribute not the attribute value. For example, if an attribute has ID 1234 and name 'Weight,' then '1234' is extracted.
Name	Extracts only the name of the attribute and not the attribute value. For example, if an attribute has ID 1234 and name 'Weight,' then the word 'Weight' is extracted.
Attribute Type	Extracts the type of the attribute, i.e., either Description or Specification. For example, for a description attribute the word 'Description' is extracted.
Attribute Multi Valued	Extracts if the attribute is multi-valued. If the attribute is a multi-valued attribute the word 'true' is extracted, otherwise the word 'false' is extracted.
Attribute FullText	Extracts if the attribute is full text indexable. If the attribute is a full text indexable attribute the word 'true' is extracted, otherwise the word 'false' is extracted.
Attribute Calculated	Extracts if the attribute is a calculated attribute. If the attribute is a calculated attribute the word 'true' is extracted, otherwise the word 'false' is extracted.
Attribute Externally Maintained	Extracts if the attribute is externally maintained. If the attribute is externally maintained the word 'true' is extracted, otherwise the word 'false' is extracted.
Attribute Dimension Dependencies	Extracts if the attribute has dimension dependencies. If the attribute has dimension dependencies they are listed and separated by a semicolon (;).
Attribute Validation Base Type	Extracts the validation base type of the attribute. For example, if the attribute has Text as validation base type, the word 'text' is extracted.
Attribute	Extracts the minimum value for the attribute, if available.

Aspect	Description
Validation Minimum value	For example, for a number attribute with a minimum value of 100, the number '100' is extracted. For a number attribute without a minimum value, a single target tag is extracted with a trailing slash (i.e., <MinVal/>).
Attribute Validation Maximum value	Extracts the maximum value for the attribute, if available. For example, for a number attribute with a maximum value of 500, the number '500' is extracted. For a number attribute without a maximum value, a single target tag is extracted with a trailing slash (i.e., <MaxVal/>).
Attribute Validation Maximum length	Extracts the maximum length of the attribute, if available. For example, for an attribute with a maximum length of 20, the number '20' is extracted. For a number attribute without a maximum length, a single target tag is extracted with a trailing slash (i.e., <MaxLen/>).
Attribute Validation Input mask	Extracts the attribute validation input mask, if available. For example, if an attribute has an attribute validation mask of 'F0,' then 'F0' is extracted. For an attribute without an input mask, a single target tag is extracted with a trailing slash (i.e., <InputMask/>).
Attribute Validation LOV ID	Extracts the ID of the List Of Values, if available. For example, for an attribute configured to use the List Of Value with ID Colors, the word 'Colors' is extracted. For an attribute without an LOV, a single target tag is extracted, with a trailing slash (i.e., <LOVID/>).
AttributeGroup Ref ID	Extracts a list of IDs of the attribute groups that include the attribute. Each group ID is separated by a semicolon (;). For example, if the attribute is in attribute groups with ID Group1 and ID Group2 then 'Group1;Group2' is extracted.
AttributeGroup Ref Name	Extracts a list of names of the attribute groups that include the attribute. Each group name is separated by a semicolon (;). For example, if the attribute is in attribute groups named GroupName1 and GroupName2, then 'GroupName1;GroupName2' is extracted.

For a description of the Constant Value, All Attributes, Select Attributes, and Custom Attributes aspects, see **Outbound Map Data - Data Source** section.

Attribute Links and LOVs Instruction in Generic XML

Using the following Generic XML template and mapping allows the export of System Setup information along with Product information. In this scenario, the following is exported:

- the ID and Name of the Product
 - the ID and Name of the linked attributes on the Product
 - the ID, Name, and Values of the LOV on any of the linked attributes

Units for linked attributes can also be mapped and exported as discussed in **Attribute Links and Units Instruction in Generic XML**.

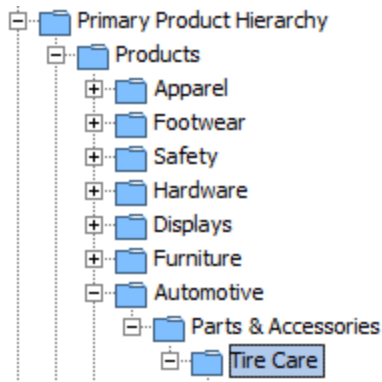
The data path data source is also available using the IDoc MATMAS 05 format.

Objects

Choose **Product** as object type to export.

The screenshot shows the 'Export Manager' dialog box with the 'Select Objects' step selected in the left sidebar. The main area displays a table with columns: ID, Name, Object Type, and Path. Below the table, there are checkboxes for 'Only export selected objects' and 'Only export leaf objects', both of which are unchecked. The 'Include object types' field is set to '<All object types>'. The 'Export:' dropdown menu is set to 'Product', with a red arrow pointing to it. At the bottom, there are buttons for 'Back', 'Next', 'Finish', and 'Cancel'.

Use the **Add Objects** link to select the data to export.



Template

Choose **Generic XML** for format and provide the following template text in the Sample field.

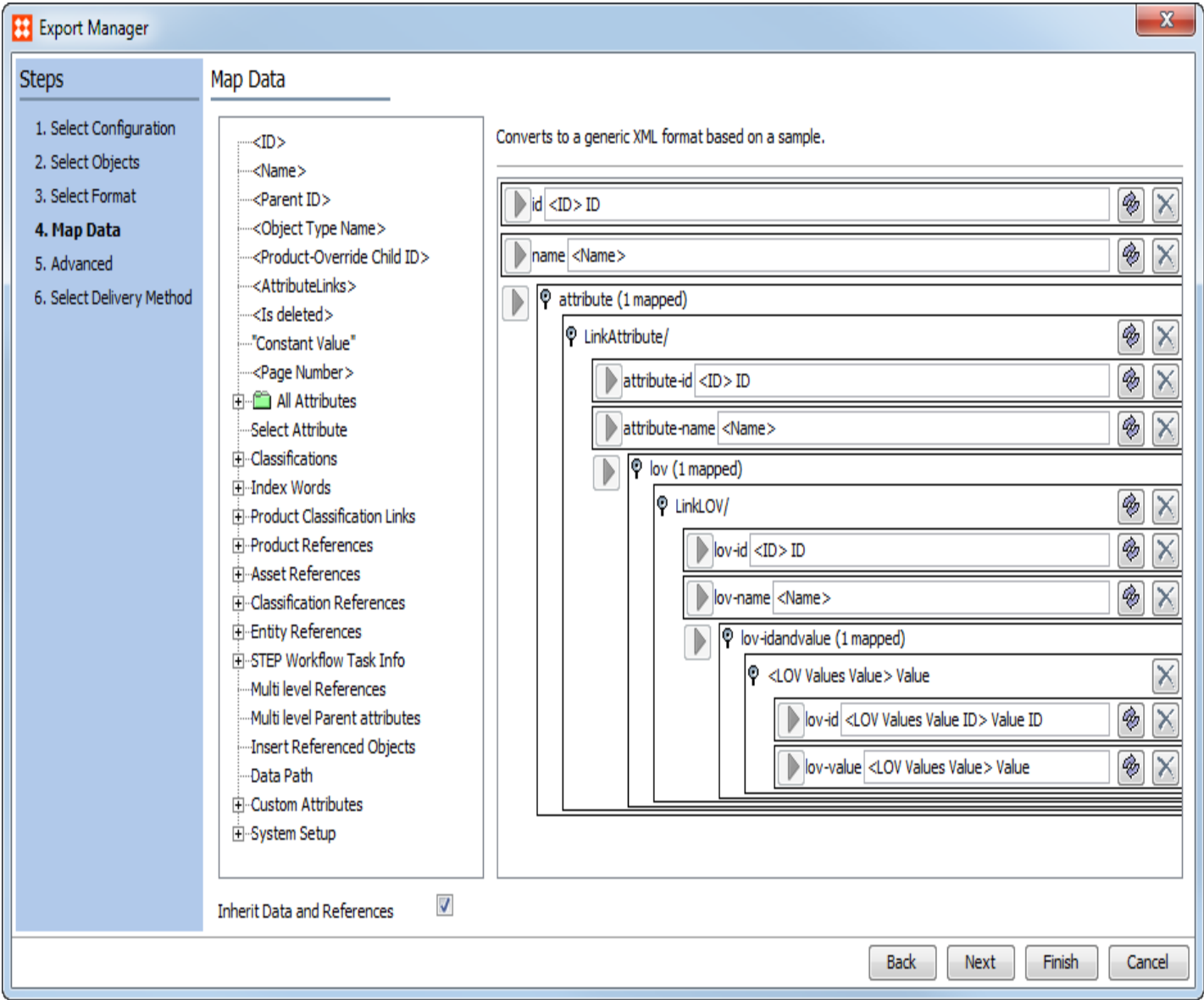
```

<products>
  <product>
    <?Product?>
    <id><?Target?></id>
    <name><?Target?></name>
    <attribute>
      <?MultiTarget?>
      <id><?Target attribute-id?></id>
      <name><?Target attribute-name?></name>
    <lov>
      <?MultiTarget?>
      <id><?Target lov-id?></id>
      <name><?Target lov-name?></name>
    <lov-idandvalue>
      <?MultiTarget?>
      <value><?Target lov-value?></value>
      <id><?Target lov-id?></id>
    </lov-idandvalue>
  </lov>
</attribute>
</product>
</products>

```

Mapping

The Data Path aspect enables the complete set of mapping rules displayed below:



For details on mapping, see **Mapping Attribute Links, LOVs, and LOV Values in Generic XML**.

Results

The Product, its linked Attributes, and any LOVs (and the values) used by the linked attributes are exported.

Note: Only local attribute linds are exported.

```

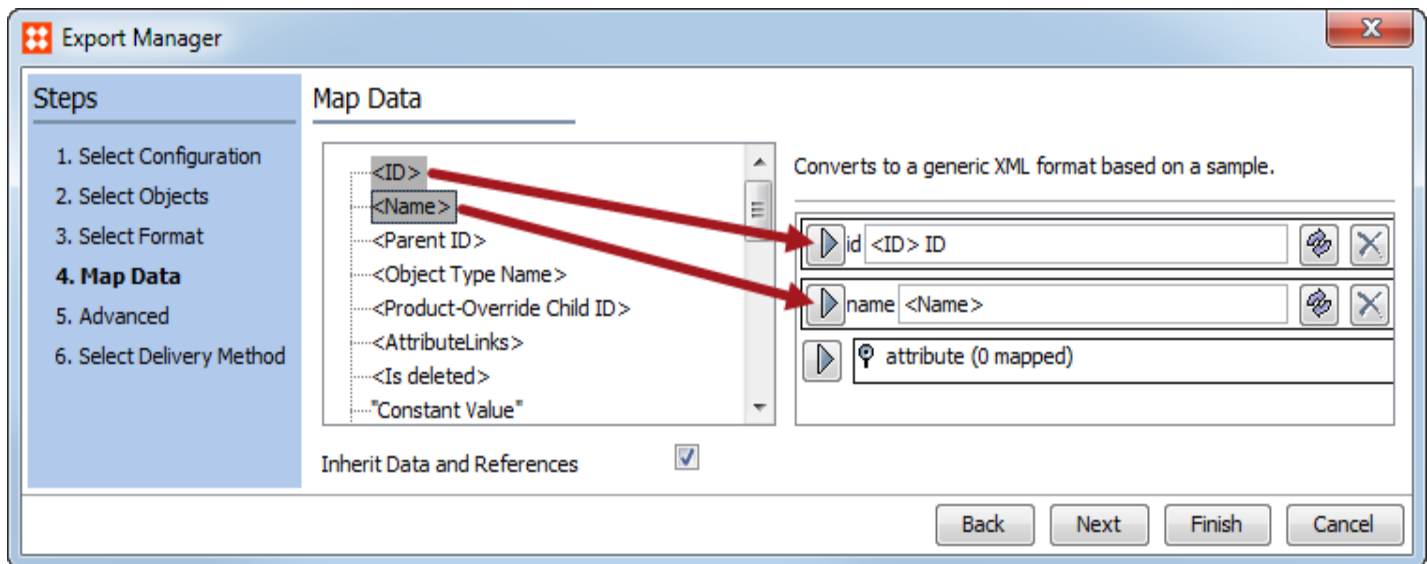
<product>
  <id>8312</id>
  <name>Tire Care</name>
  <attribute>
    <id>AirGaugeIncluded</id>
    <name>Air gauge included</name>
    <lov>
      <id>YesNo</id>
      <name>Yes/No (translatable)</name>
      <lov-idandvalue>
        <id/>
        <value>No</value>
      </lov-idandvalue>
      <lov-idandvalue>
        <id/>
        <value>Yes</value>
      </lov-idandvalue>
    </lov>
  </attribute>
  <attribute>
    <id>PressureReliefValve</id>
    <name>Pressure Relief Valve</name>
    <lov>
      <id>YesNo</id>
      <name>Yes/No (translatable)</name>
      <lov-idandvalue>
        <id/>
        <value>No</value>
      </lov-idandvalue>
      <lov-idandvalue>
        <id/>
        <value>Yes</value>
      </lov-idandvalue>
    </lov>
  </attribute>
  <attribute>
    <id>HoseLength</id>
    <name>Hose length</name>
  </attribute>
</product>

```

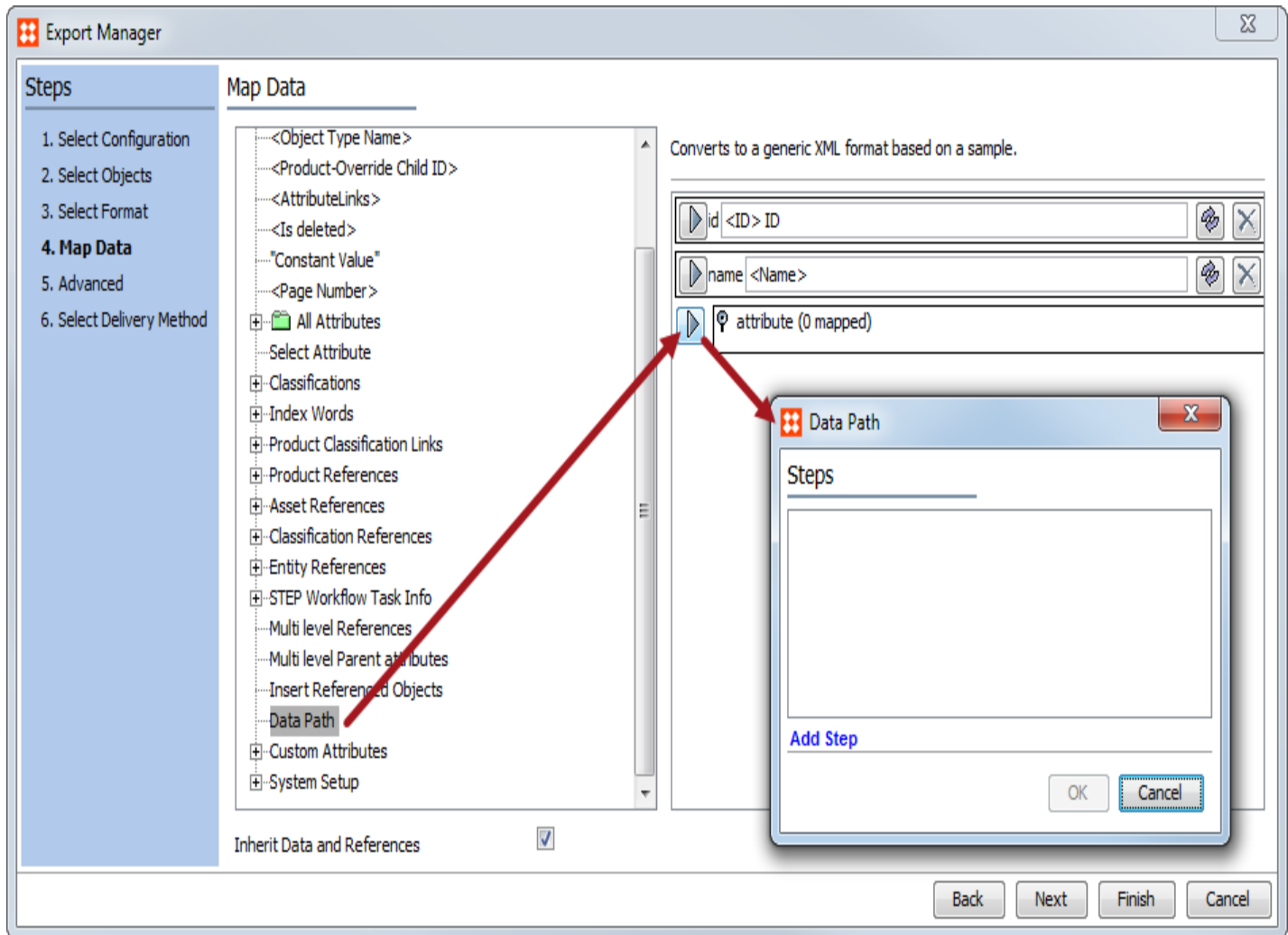
Mapping Attribute Links, LOVs, and LOV Values in Generic XML

The **Attribute Links and LOVs Instruction in Generic XML** section discusses additional setup for a Generic XML or IDoc MATMAS 05 export. Use the following steps to map the Product, linked attributes, and their LOV information.

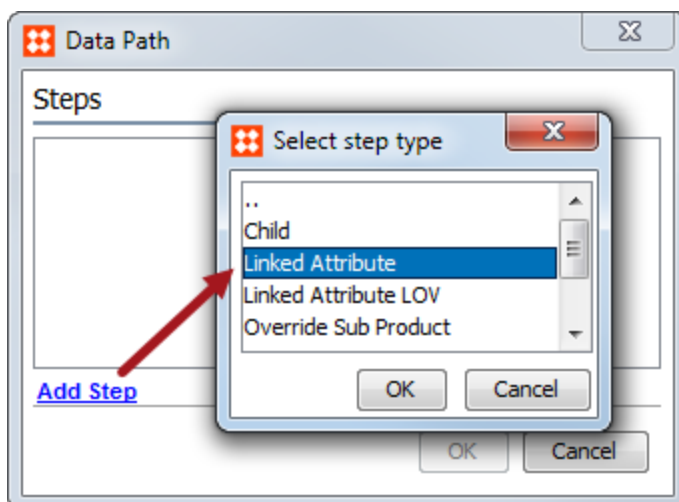
1. Map **ID** and **Name** for the Product.



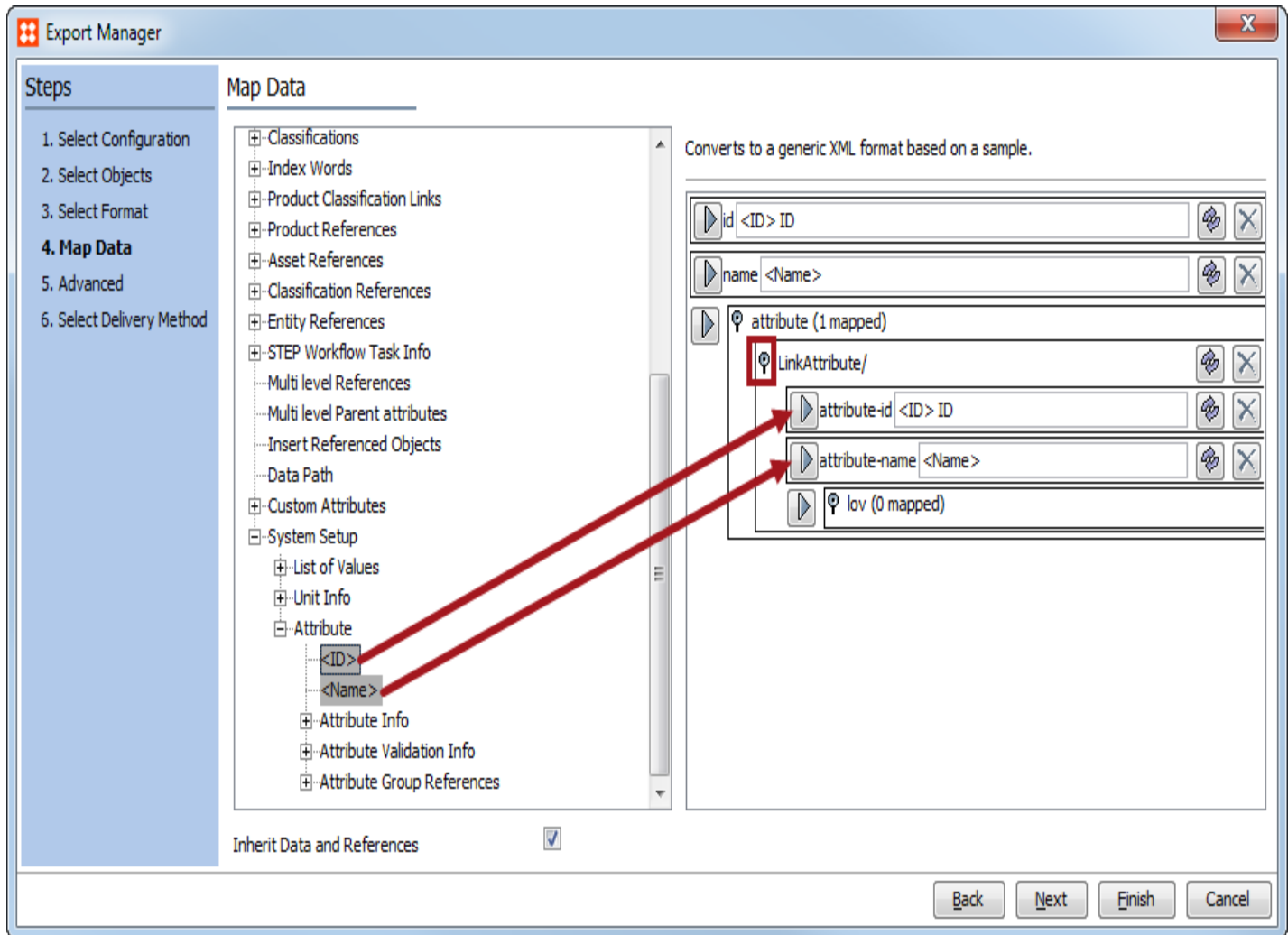
2. Map **Data Path** to attribute.



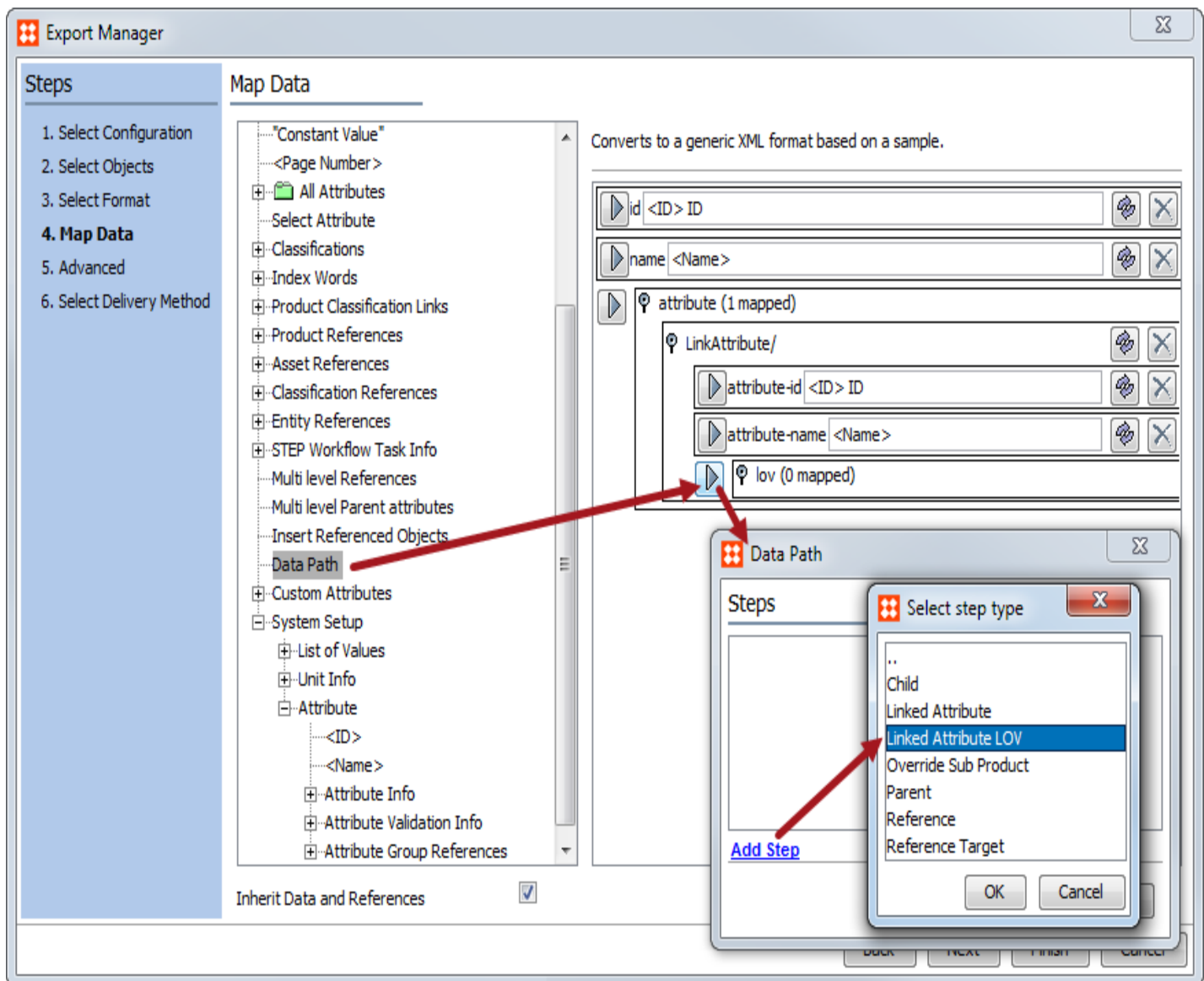
3. Click the **Add Step** link and select **Linked Attribute**.



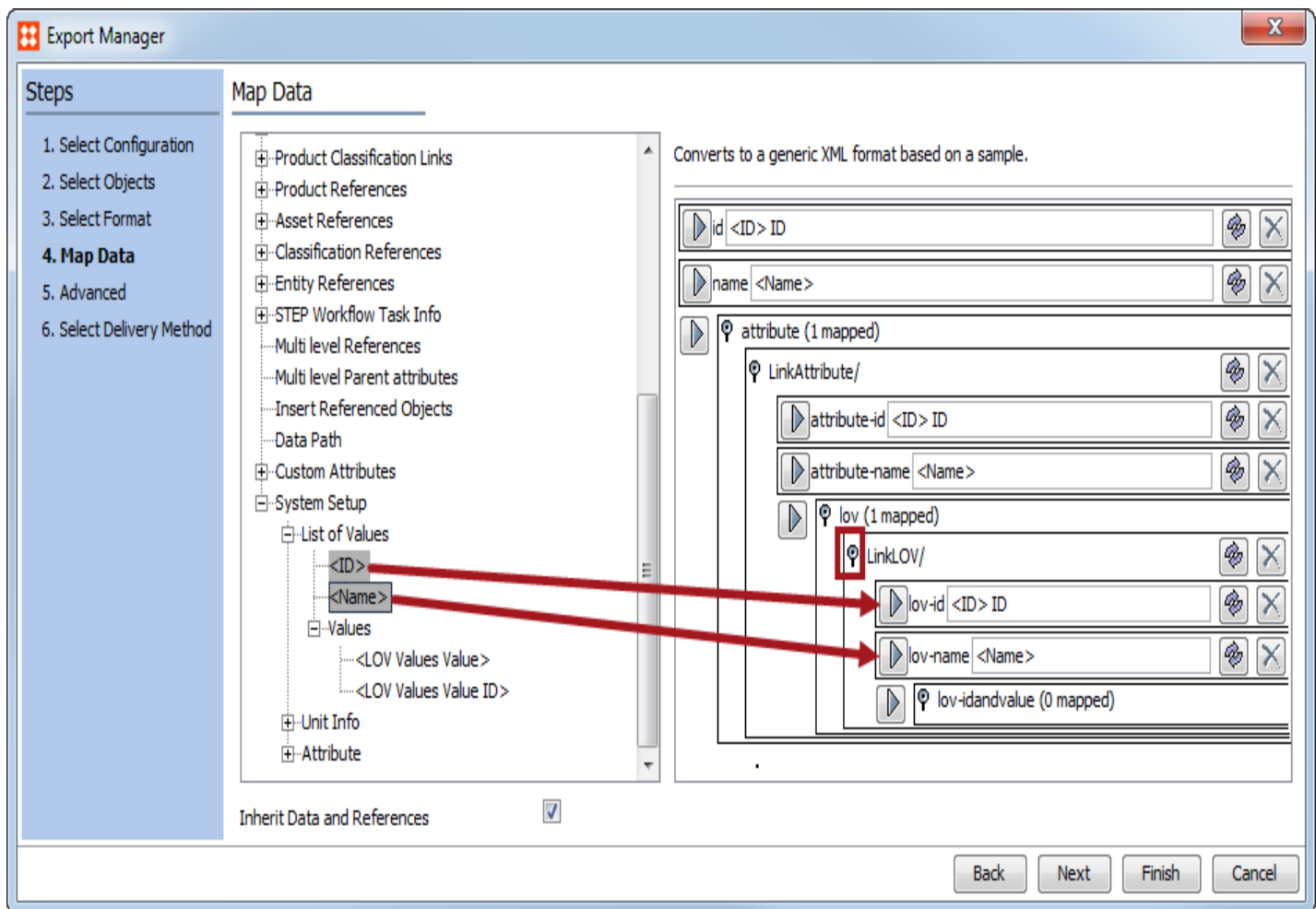
4. Click OK to close the **Select step type** and **Data Path** dialogs.
5. Open the LinkAttribute/ flipper and map Attribute **ID** and **Name**.



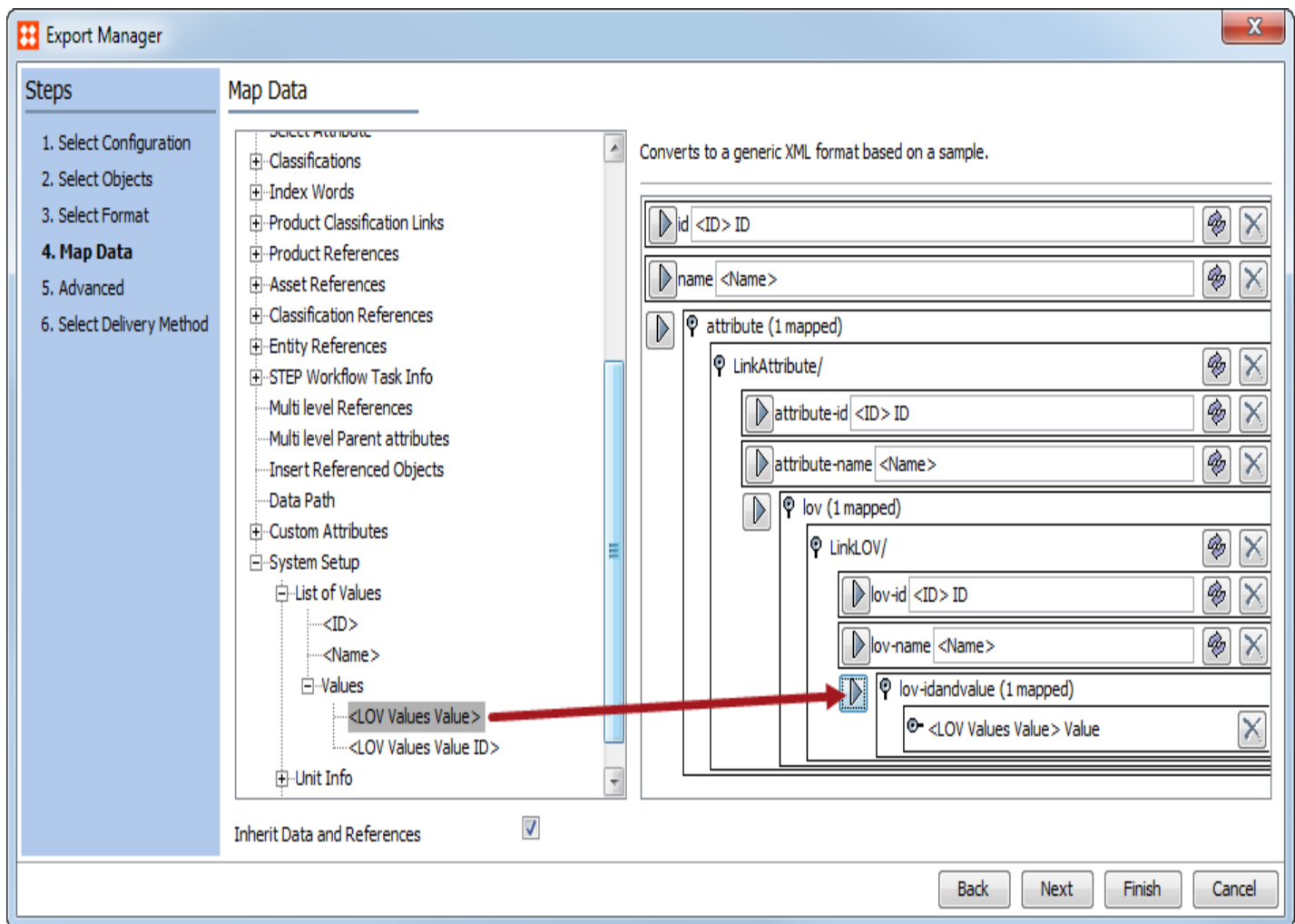
6. Map **Data Path** to lov, click **Add Step** and select **Linked Attribute LOV**.



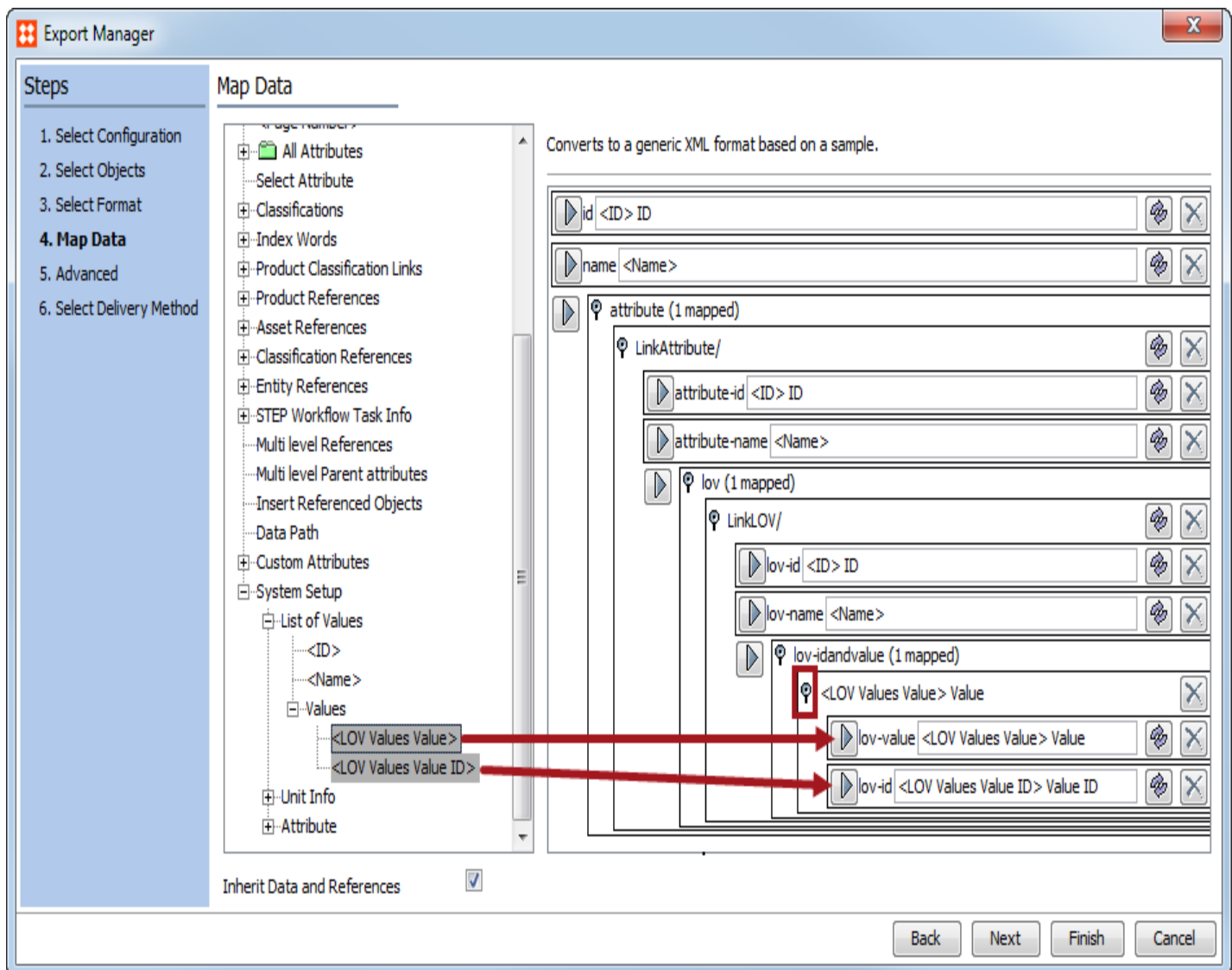
7. Click OK to close the **Select step type** and **Data Path** dialogs.
8. Open the LinkLOV/ flipper and map Attribute **ID** and **Name**.



9. Map **LOV Values Value** to lov-idandvalue.



10. Open the <LOV Values Value> Value flipper and map **LOV Values Value** and **LOV Values ID**.



Attribute Links and Units Instruction in Generic XML

Using the following Generic XML template and mapping allows the export of System Setup information along with Product information. In this scenario, the following is exported:

- the ID and Name of the Product
 - the ID and Name of the linked attributes on the Product
 - the ID, Name, and Base of the Unit on any of the linked attributes

LOVs for linked attributes can also be mapped and exported as described in **Attribute Links and LOVs Instruction in Generic XML**.

The data path data source is also available using the IDoc MATMAS 05 format.

Objects

Choose **Product** as object type to export.

Export Manager

Steps

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

Select Objects

ID	Name	Object Type	Path
> Add Objects			

Only export selected objects ☐

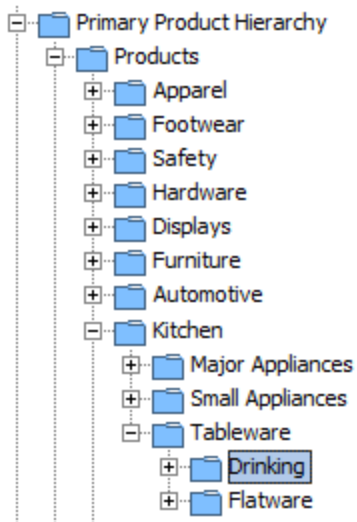
Only export leaf objects ☐

Include object types

Export: Product

Back Next Finish Cancel

Use the **Add Objects** link to select the data to export.



Template

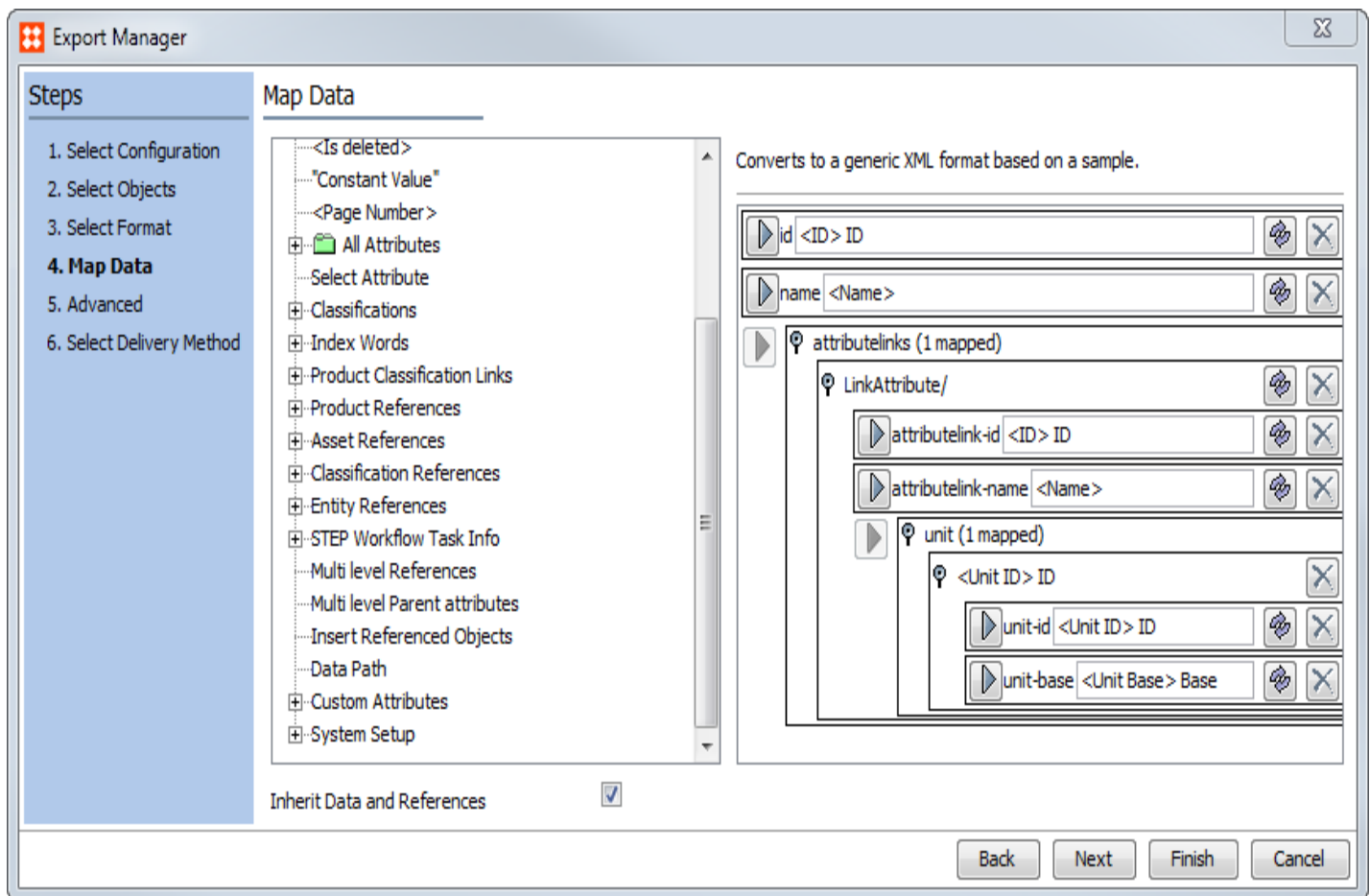
Choose **Generic XML** for format and provide the following template text in the Sample field.

```

<products>
  <product>
    <?Product?>
    <id><?Target?></id>
    <name><?Target?></name>
    <attributelinks>
      <?MultiTarget?>
      <id><?Target attributelink-id?></id>
      <name><?Target attributelink-name?></name>
    </unit>
    <?MultiTarget?>
    <attributelink-unit-id><?Target unit-id?></attributelink-unit-id>
    <attributelink-unit-base><?Target unit-base?></attributelink-unit-base>
  </unit>
</attributelinks>
</product>
</products>
  
```

Mapping

The Data Path aspect enables the complete set of mapping rules displayed below:



For details on mapping, see the **Mapping Attribute Links, Units, and Unit Base in Generic XML** section.

Results

The Product, its linked Attributes, and any Units (and the Base) used by the linked attributes are exported.

Note: Only local attribute links are exported.

```

<products>
  <product>
    <id>8318</id>
    <name>Drinking</name>
    <attributelinks>
      <id>Capacity</id>
      <name>Capacity</name>
      <unit>
        <attributelink-unit-id>unece.unit.FTK</attributelink-unit-id>
        <attributelink-unit-base>unece.unit.MTK</attributelink-unit-base>
      </unit>
      <unit>
        <attributelink-unit-id>unece.unit.MTK</attributelink-unit-id>
        <attributelink-unit-base/>
      </unit>
      <unit>
        <attributelink-unit-id>unece.unit.OZA</attributelink-unit-id>
        <attributelink-unit-base>unece.unit.MTQ</attributelink-unit-base>
      </unit>
    </attributelinks>
  </product>
</products>

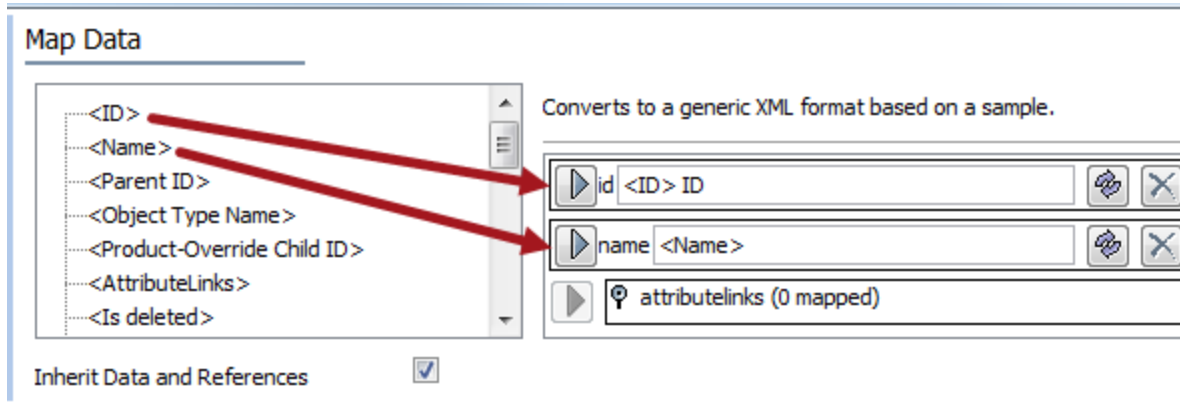
```

Mapping Attribute Links, Units, and Unit Base in Generic XML

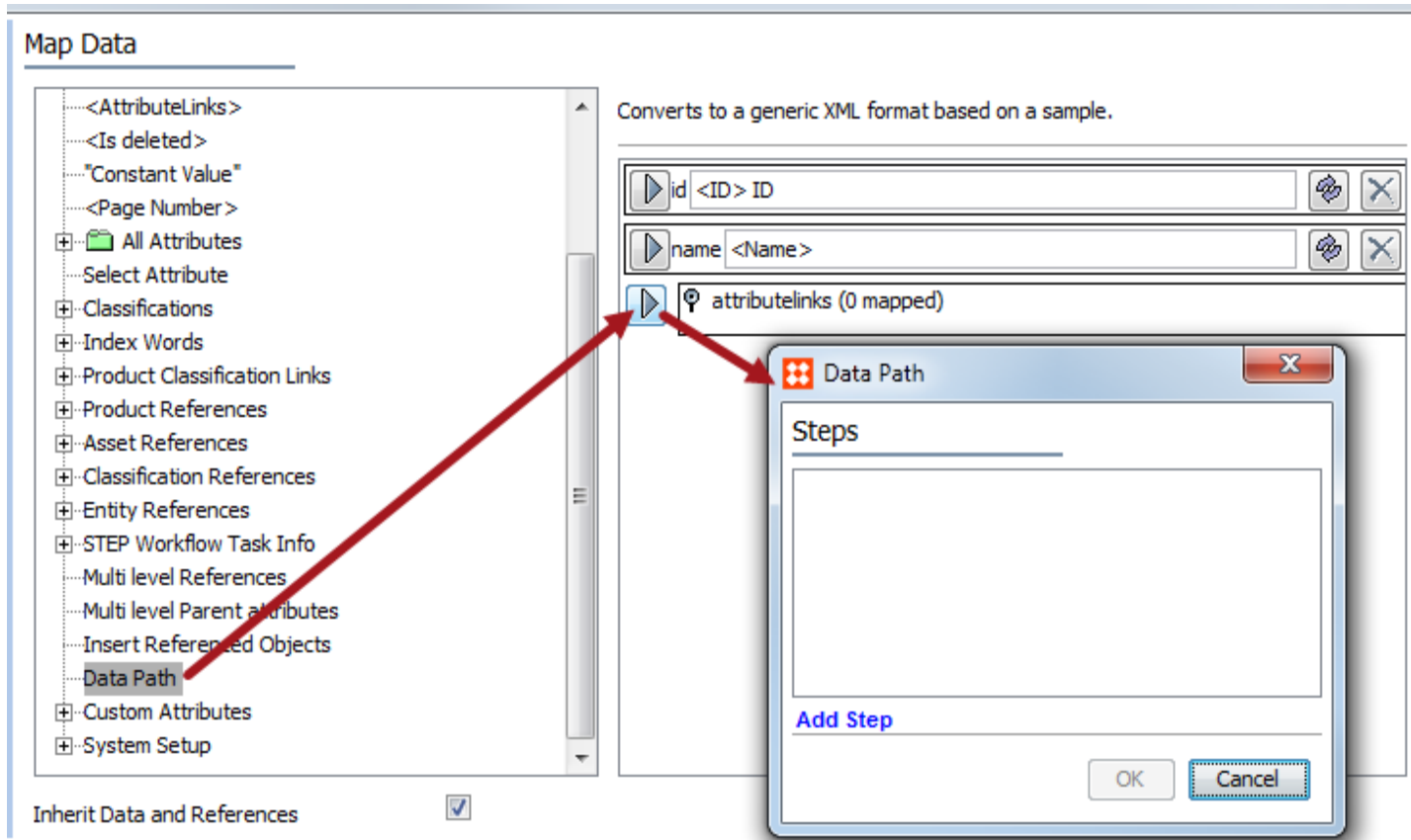
The **Attribute Links and Units Instruction in Generic XML** section discusses additional setup for a Generic XML export. Use the following steps to map the Product, linked attributes, and their Unit information.

The data path data source is also available using the IDoc MATMAS 05 format.

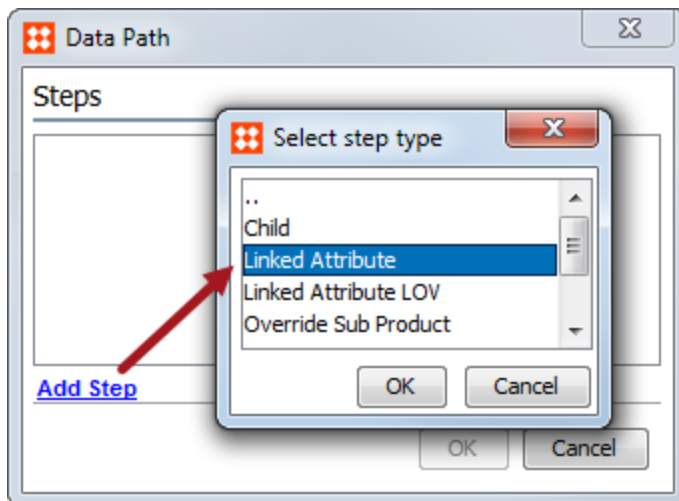
1. Map **ID** and **Name** for the Product.



2. Map **Data Path** to attributelinks.



- Click the **Add Step** link and select **Linked Attribute**.



- Click OK to close the **Select step type** and **Data Path** dialogs.
- Open the LinkAttribute/ flipper and map Attribute **ID** and **Name**.

Map Data

Entity References

STEP Workflow Task Info

Multi level References

Multi level Parent attributes

Insert Referenced Objects

Data Path

Custom Attributes

System Setup

List of Values

Unit Info

Attribute

<ID>

<Name>

Attribute Info

Attribute Validation Info

Attribute Group References

Converts to a generic XML format based on a sample.

id <ID> ID

name <Name>

attributelinks (1 mapped)

LinkAttribute/

attributelink-id <ID> ID

attributelink-name <Name>

unit (0 mapped)

Inherit Data and References ☒

6. Map **Unit ID** to unit.

Map Data

Insert Referenced Objects

Data Path

Custom Attributes

System Setup

List of Values

Unit Info

<Unit ID>

<Unit Base>

<Unit Factor>

<Unit Offset>

Attribute

<ID>

<Name>

Attribute Info

Attribute Validation Info

Attribute Group References

Converts to a generic XML format based on a sample.

id <ID> ID

name <Name>

attributelinks (1 mapped)

LinkAttribute/

attributelink-id <ID> ID

attributelink-name <Name>

unit (1 mapped)

<Unit ID> ID

Inherit Data and References ☒

7. Open the <Unit ID> flipper and map Unit **ID** and **Base**.

Map Data

STEP Workflow Task Info

Multi level References

Multi level Parent attributes

Insert Referenced Objects

Data Path

Custom Attributes

System Setup

List of Values

Unit Info

<Unit ID>

<Unit Base>

<Unit Factor>

<Unit Offset>

Attribute

<ID>

<Name>

Attribute Info

Attribute Validation Info

Attribute Group References

Converts to a generic XML format based on a sample.

id <ID> ID

name <Name>

attributelinks (1 mapped)

LinkAttribute/

attributelink-id <ID> ID

attributelink-name <Name>

unit (1 mapped)

<Unit ID> ID

unit-id <Unit ID> ID

unit-base <Unit Base> Base

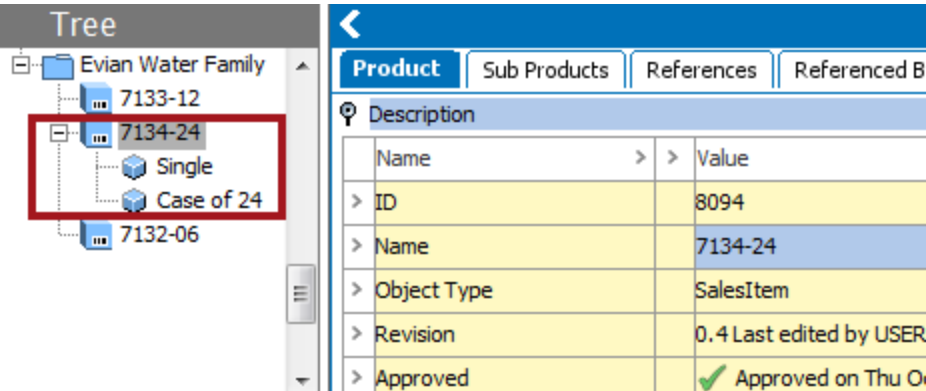
Inherit Data and References ☒

DoNotEmit Instruction in Generic XML

The Generic XML <?DoNotEmit?> tag allows you to hide levels that exist for the exported values or references from different products, by displaying data on the same level. The tag surrounding the DoNotEmit instruction is excluded from the output.

Objects

The following SalesItem and children are selected for export.



Template

These templates allow you to compare the difference that using DoNotEmit has on the output.

Without DoNotEmit	With DoNotEmit
<pre><Families> <Family ID="[/Target ID?]"> <?Record?> <Name><?Target Name?></Name> <Children> <Child> <?MultiTarget?> <ID><?Target ID?></ID> <Name><?Target Name?></Name> </Child> </Children> </Family> </Families></pre>	<pre><Families> <Family ID="[/Target ID?]"> <?Record?> <Name><?Target Name?></Name> <Children> <Child> <?DoNotEmit?> <?MultiTarget?> <ID><?Target ID?></ID> <Name><?Target Name?></Name> </Child> </Children> </Family> </Families></pre>

Mapping

Mapping for export is the same for both templates.

1. After mapping the SalesItem ID and Name, use the Data Path - Child option to map the Child target.

The screenshot shows the 'Map Data' window in Stibo Systems. On the left is a tree view of data elements. The 'Data Path' element is selected, and a red arrow points from it to the 'Child (0 mapped)' entry in the mapping table. The mapping table has three rows: 'ID' mapped to '<ID> ID', 'Name' mapped to '<Name>', and 'Child (0 mapped)'. Below the table is a 'Data Path' dialog box with a 'Steps' list and an 'Add Step' button. A red arrow points from 'Add Step' to a 'Select step type' dialog box. This dialog box has a list of step types, with 'Child' selected. At the bottom of the 'Map Data' window, there is a checkbox labeled 'Inherit Data and References' which is checked.

2. Map the Child's ID and Name.

Map Data

<ID>

<Name>

<Parent ID>

<Object Type Name>

<Product-Override Child ID>

<AttributeLinks>

<Is deleted>

"Constant Value"

<Page Number>

All Attributes

Select Attribute

Classifications

Converts to a generic XML format based on a sample.

ID <ID> ID

Name <Name>

Child (1 mapped)

Child/

ID <ID> ID

Name <Name>

Inherit Data and References

Results

The following results are achieved when exporting using the sample templates and mapping above.

Without DoNotEmit	With DoNotEmit
<pre><Families> <Family ID="8094"> <Name>7134-24</Name> <Children> <Child> <ID>8082</ID> <Name>Single</Name> </Child> <Child> <ID>107807</ID> <Name>Case of 24</Name> </Child> </Children> </Family> </Families></pre>	<pre><Families> <Family ID="8094"> <Name>7134-24</Name> <Children> <ID>8082</ID> <Name>Single</Name> <ID>107807</ID> <Name>Case of 24</Name> </Children> </Family> </Families></pre>

© Stibo Systems - Confidential - STEP Trailblazer 8.3

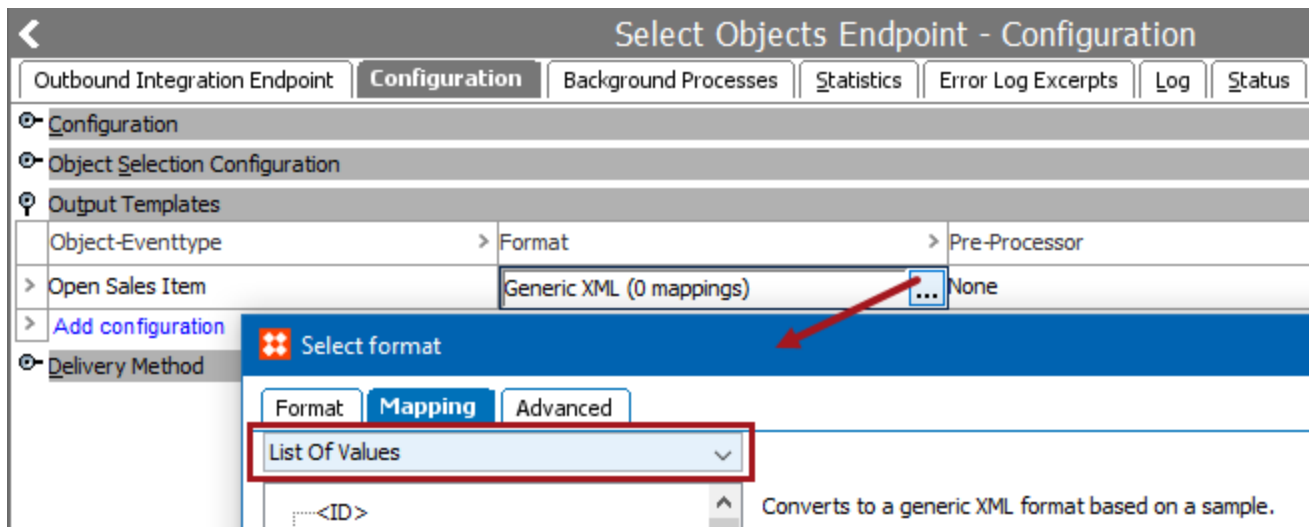
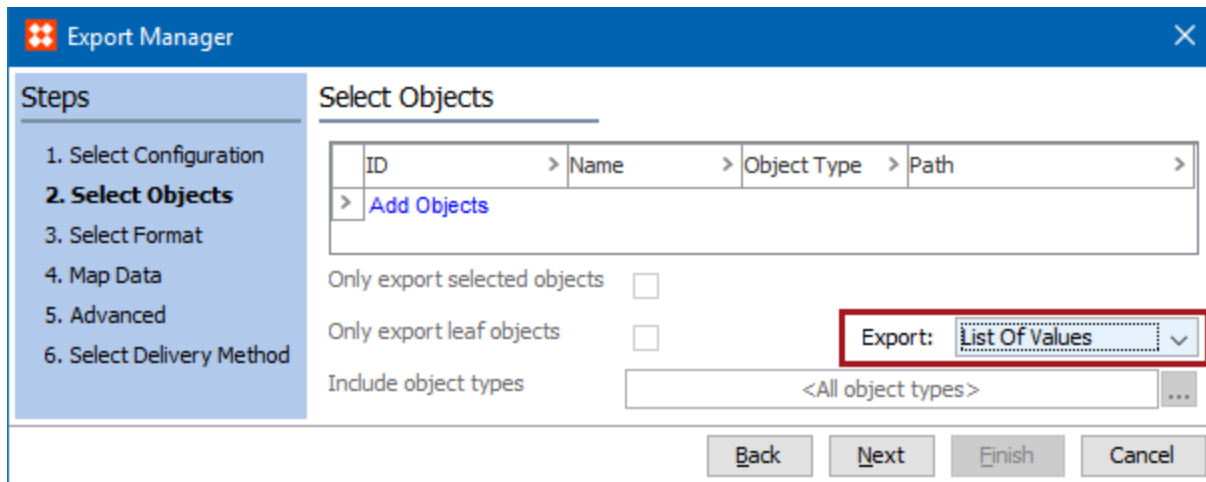
242

List of Values Object Type in Generic XML

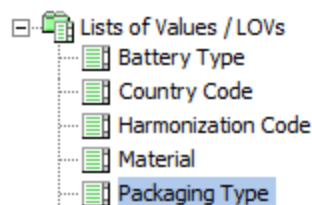
Export tools require that you select objects and provide both a sample template and mapping when exporting Generic XML.

Objects

1. Select **List of Values** as object type to export, based on the export tool being used.



2. Select the LOV to export, based on your export tool.



For Export Manager, see the **Export Manager - Select Objects** topic.

For OIEP, see the **Object Types** section of the **OIEP - Event-Based - Event Triggering Definitions Tab** topic or the **OIEP - Select Objects - Object Selection Configuration Flipper** topic.

Template

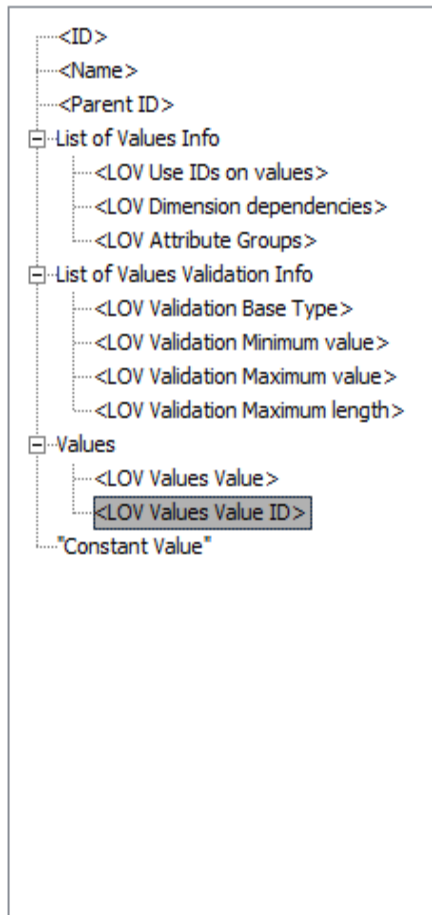
Choose **Generic XML** as the export format and provide the following template text in the Sample field. The template must define all data / fields of the LOV that should be output. All options are included in the following sample template.

```
<root>
  <lov>
    <?Record?>
    <id><?Target?></id>
    <name><?Target?></name>
    <parent><?Target?></parent>
    <baseType><?Target?></baseType>
    <minValue><?Target?></minValue>
    <maxValue><?Target?></maxValue>
    <maxLength><?Target?></maxLength>
    <useIDs><?Target?></useIDs>
    <dimensionDependencies><?Target?></dimensionDependencies>
    <attributeGroups><?Target?></attributeGroups>
    <values id="[/Target id?]"><?MultiTarget?><?Target?></values>
  </lov>
</root>
```

Mapping

Below, the <LOV Values Value> node is mapped twice to allow the Value ID and content to be displayed on the same line.

Map Data



Converts to a generic XML format based on a sample.

▶	id	<ID> ID	⚙	✕
▶	name	<Name>	⚙	✕
▶	parent	<Parent ID> ID	⚙	✕
▶	baseType	<LOV Validation Base Type> Base Type	⚙	✕
▶	minValue	<LOV Validation Minimum value> Minimum value	⚙	✕
▶	maxValue	<LOV Validation Maximum value> Maximum value	⚙	✕
▶	maxLength	<LOV Validation Maximum length> Maximum length	⚙	✕
▶	useIDs	<LOV Use IDs on values> Use IDs on values	⚙	✕
▶	dimensionDependencies	<LOV Dimension dependencies> Dimension dependencies	⚙	✕
▶	attributeGroups	<LOV Attribute Groups> Attribute Groups	⚙	✕
▶	values (1 mapped)			
	⊕	<LOV Values Value> Value		✕
	▶	values	<LOV Values Value> Value	⚙ ✕
	▶	id	<LOV Values Value ID> Value ID	⚙ ✕

Inherit Data and References



Results

The output includes information for the LOV selected for export. If the LOV parameter 'Use IDs on Values' is set to Yes, the export will include both the Value and Value ID.

```

<root>
  <lov>
    <id>PackagingTypeLOV</id>
    <name>Packaging Type</name>
    <parent>List Of Values group root</parent>
    <baseType>text</baseType>
    <minValue></minValue>
    <maxValue></maxValue>
    <maxLength>100</maxLength>
    <useIDs>true</useIDs>
    <dimensionDependencies></dimensionDependencies>
    <attributeGroups></attributeGroups>
    <values id="PX">Pallet</values>
    <values id="SW">Shrinkwrapped</values>
    <values id="SPL">Spool</values>
    <values id="BP">Banded Package</values>
    <values id="TU">Tube</values>
    <values id="CG">Card</values>
    <values id="BX">Box</values>
    <values id="LSE">Loose</values>
    <values id="CT">Carton</values>
    <values id="CY">Cylinder</values>
    <values id="BG">Bag</values>
    <values id="CS">Case</values>
  </lov>
</root>

```

Multi-Valued Objects in Generic XML

There are two Generic XML options for exporting fields that allow multiple values.

- Repeated Target exports a delimited list of the values in a single tag
- MultiTarget exports each value individually, in separate tags

MultiTarget is usually the preferred option.

Objects

The Country of Origin multi valued attribute is exported for each of these selected products:

Description
Name >> Value
> ID 8083
> Name EVN-06
> Object Type Item
Item Import/Export Information
Name >> Value
> Country of Origin  FRANCE CANADA CHILE

Description
Name >> Value
> ID 8108
> Name EVN-12
> Object Type Item
Item Import/Export Information
Name >> Value
> Country of Origin  AUSTRALIA CHILE

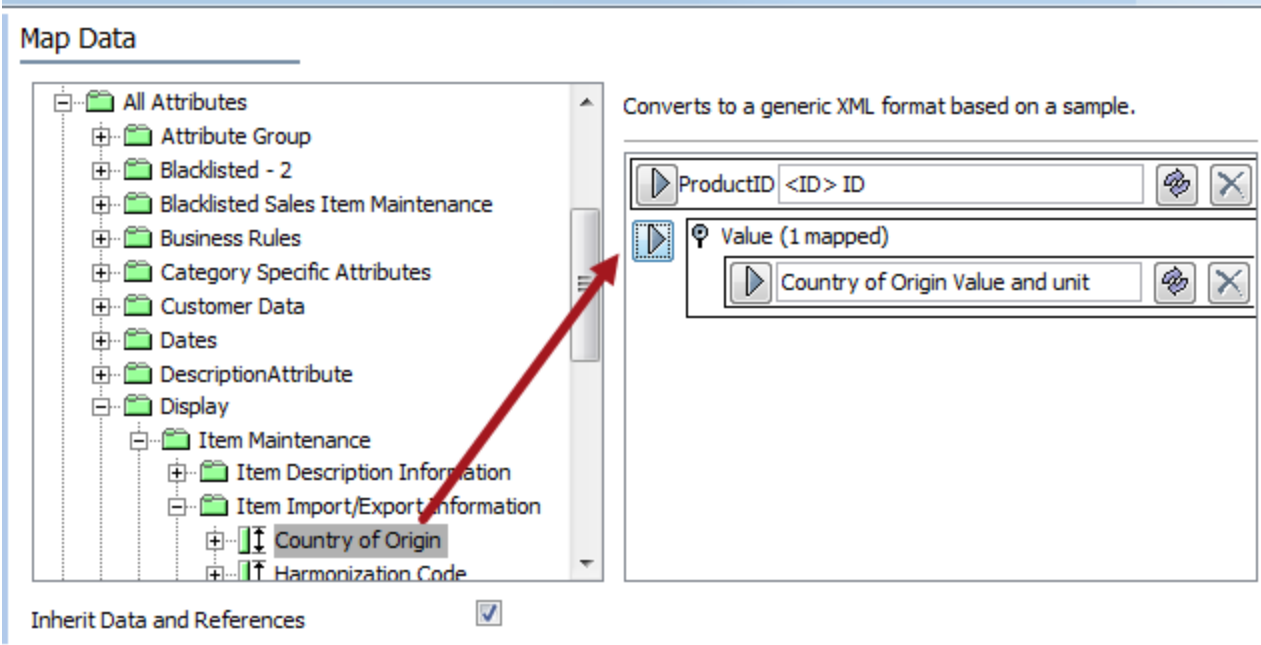
Description
Name >> Value
> ID 20674
> Name EVN-24
> Object Type Item
Item Import/Export Information
Name >> Value
> Country of Origin  NEW ZEALAND

Template

Repeated Target	MultiTarget
<pre><Products> <Product> <?Record?> <ProductID><?Target?></ProductID> <Value><?Target?></Value> <Value><?Target?></Value> </Product> </Products></pre>	<pre><Products> <Product> <?Record?> <ProductID><?Target?></ProductID> <Value><?MultiTarget?></Value> </Product> </Products></pre>

Mapping

The mapping steps are the same regardless of the Multi Valued option used.



Results

Repeated Target	MultiTarget
<pre><Products> <Product> <ProductID>8083</ProductID> <Value>FRANCE; CANADA; CHILE</Value> </Product> <Product> <ProductID>8108</ProductID> <Value>AUSTRALIA; CHILE</Value> </Product> <Product> <ProductID>20674</ProductID> <Value>NEW ZEALAND</Value> </Product> </Products></pre>	<pre><Products> <Product> <ProductID>8083</ProductID> <Value>FRANCE</Value> <Value>CANADA</Value> <Value>CHILE</Value> </Product> <Product> <ProductID>8108</ProductID> <Value>AUSTRALIA</Value> <Value>CHILE</Value> </Product> <Product> <ProductID>20674</ProductID> <Value>NEW ZEALAND</Value> </Product> </Products></pre>

Multi-Cardinality Tags in Generic XML

Single cardinality tags are used when the data produced by the associated data source is considered a 'single piece of information' and will be output inside a single occurrence of the enclosing XML element. When more than one piece of data exists for the same data source (for example, multiple images on the same product), the output requires multiple-cardinality tags.

The following multi-cardinality mapping options are discussed below: Repeated Target, MultiTarget, and Composite Mapping Targets.

Objects

The following objects are used for each of the examples in this section.

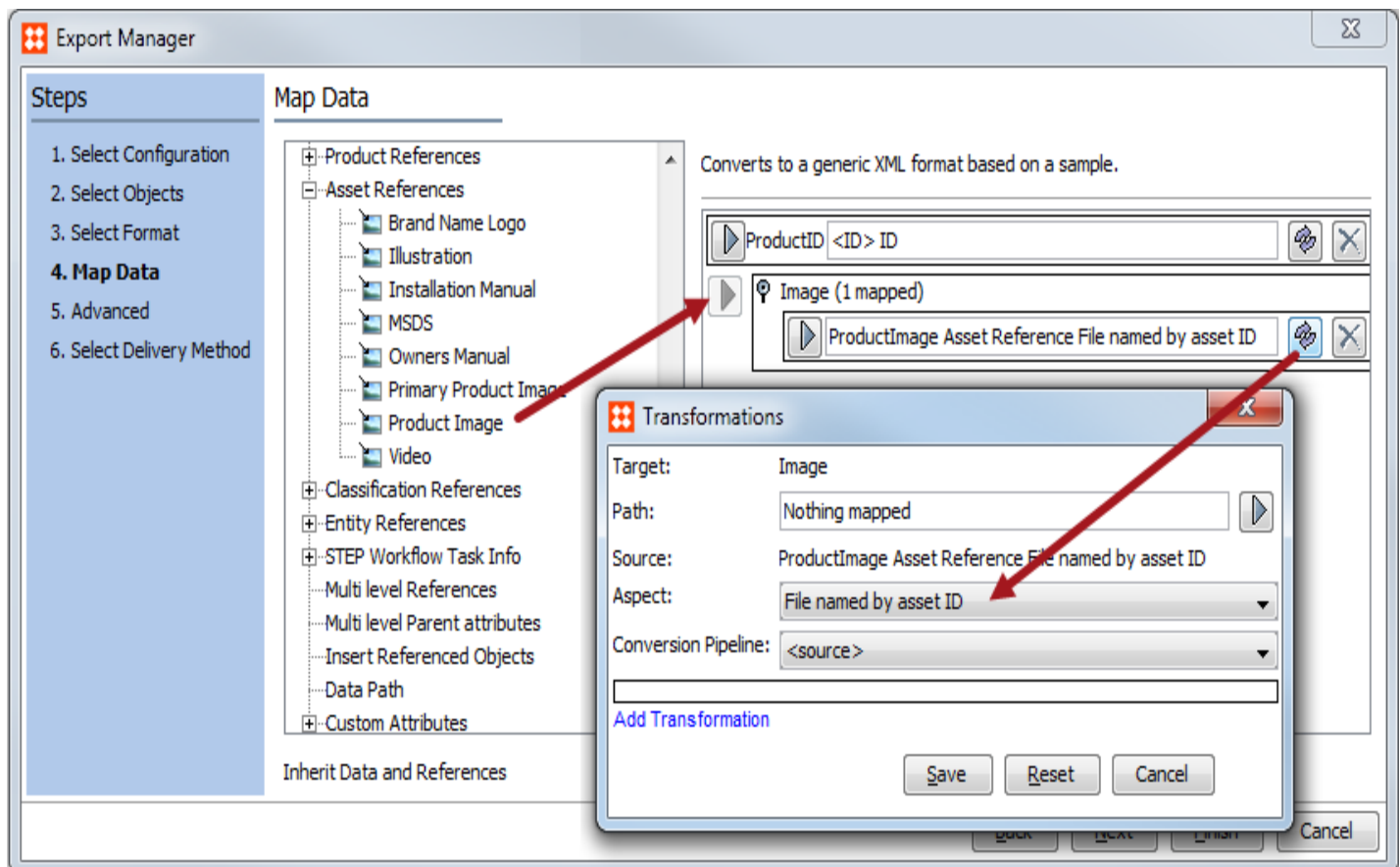
 Image References

Reference Type

Source	Reference Type	Target
> 20803	Product Image	 107625
	Product Image	 107628
> 20805	Product Image	 107625
	Product Image	 107628
	Product Image	 107629
> 109308	Product Image	 107628

Mapping

The following mapping is used for each of the examples in this section.



Repeated Targets

Repeating the same mapping target allows multiple instances of the same element to be exported. In the following sample template, note that the image target tag appears two times as `<Image><?Target?></Image>`.

When using Repeated Targets, mapping target declarations must be repeated immediately and the tag name and mapping target ID must be identical. This technique **cannot** be used for mapping targets declared below composite mapping targets.

Template

```
<Products>
  <Product>
    <?Record?>
    <ProductID><?Target?></ProductID>
    <Image><?Target?></Image>
    <Image><?Target?></Image>
  </Product>
</Products>
```

Results

The results include all instances of the images exported, even when more than two are referenced by the product. This is the same result as with MultiTarget if the mapping targets are not composite.

```
<Products>
  <Product>
    <ProductID>20803</ProductID>
    <Image>107628.png</Image>
    <Image>107625.tif</Image>
  </Product>
  <Product>
    <ProductID>20805</ProductID>
    <Image>107628.png</Image>
    <Image>107629.jpg</Image>
    <Image>107625.tif</Image>
  </Product>
  <Product>
    <ProductID>109308</ProductID>
    <Image>107628.png</Image>
  </Product>
</Products>
```

MultiTarget

The `<?MultiTarget [Optional Identifier]?>` processing instruction allows for multiple objects, multiple attributes, or multiple values for a single attribute to be represented in the same way.

In the following sample template, the image multitarget tag appears only one time as `<Image><?MultiTarget?></Image>`. However, when more than one image exists in the data, all of them are exported, as shown in the results section.

This technique **can** be used for mapping targets declared below composite mapping targets.

Template

```
<Products>
  <Product>
    <?Record?>
      <ProductID><?Target?></ProductID>
      <Image><?MultiTarget?></Image>
    </Product>
  </Products>
```

Results

The results include all instances of the images exported, even when more than two are referenced by the product. This is the same result as with the Repeated Target if the mapping targets are not composite.

```
<Products>
  <Product>
    <ProductID>20803</ProductID>
    <Image>107628.png</Image>
    <Image>107625.tif</Image>
  </Product>
  <Product>
    <ProductID>20805</ProductID>
    <Image>107628.png</Image>
    <Image>107629.jpg</Image>
    <Image>107625.tif</Image>
  </Product>
  <Product>
    <ProductID>109308</ProductID>
    <Image>107628.png</Image>
  </Product>
</Products>
```

Composite Mapping Targets

The previous multi-cardinality examples show exporting simple values, also called flat values. Often there is a need for repeating a more complex structure, which is possible in Generic XML with composite mapping targets.

For example, when exporting product data where each product has a number of asset references and the output needs to contain a small XML structure (substructure) for each asset. Each substructure will contain the required values of the asset, which in some cases, will include the same value as other substructures.

Mapping a data source to a composite mapping target will cause the composite substructure to be evaluated and written to the output once for each value produced by the data source.

Template

```
<products>
  <product>
    <?Record?>
      <ProductID><?Target?></ProductID>
      <image>
        <?MultiTarget?>
          <id><?Target?></id>
          <filename><?Target?></filename>
          <description><?Target?></description>
        </image>
      </product>
    </products>
```


Results

- The image tag is a multi-cardinality mapping target
- The additional mapping targets **inside** the image tag make it a **composite** mapping target
- The remaining mapping targets (id, filename and description) become ordinary mapping targets placed **below** the composite mapping target

```

<products>
  <product>
    <ProductID>20803</ProductID>
    <image>
      <id>107628</id>
      <filename>107628.png</filename>
      <description>20801</description>
    </image>
    <image>
      <id>107625</id>
      <filename>107625.tif</filename>
      <description>20805</description>
    </image>
  </product>
  <product>
    <ProductID>20805</ProductID>
    <image>
      <id>107628</id>
      <filename>107628.png</filename>
      <description>20801</description>
    </image>
    <image>
      <id>107629</id>
      <filename>107629.jpg</filename>
      <description>20803</description>
    </image>
    <image>
      <id>107625</id>
      <filename>107625.tif</filename>
      <description>20805</description>
    </image>
  </product>
  <product>
    <ProductID>109308</ProductID>
    <image>
      <id>107628</id>
      <filename>107628.png</filename>
      <description>20801</description>
    </image>
  </product>
</products>

```

Nestable Data Paths in Generic XML

The Nestable Data Path feature is only available during export of nested repeated structures using Generic XML.

The **Nestable Data Paths in Generic XML Example** illustrates the concepts and capabilities provided by this feature.

Data Path Concept

The mapping step of the export wizard is used to bind Data Sources to Mapping Targets. Data sources evaluate and produce values for the export result relative to the object selected for export (for example product, entity, classification). When bound correctly, an ID data source will produce the ID of the object currently being processed for export, while an Attribute data source will produce the value of some specific attribute on the object being processed.

The Data Path data source makes it possible to change the object that other data sources, nested below the Data Path source, use when they are asked to evaluate. Instead of evaluating against the object being exported, they will be evaluated relative to some other object, or reference, reached by following a path from the exported object. For example, the following mapping target structure is produced by the Nestable Data Path Example:

- Product-ID
- Product-Name
- Accessory (composite, multiple-cardinality)
 - Accessory-ID
 - Accessory-Description
- Image (composite, multiple-cardinality)
 - Image-ID
 - Image-Name

Mapping the existing ID data source to the Accessory-ID mapping target would result in exporting the ID value from the product.

Instead, mapping a Data Path data source to the Accessory composite target, configured with the path (Reference Accessory, Reference Target), two things will happen:

- The Accessory sub-tree will be evaluated and output once for each reference of type Accessory found from the object being processed for export.
- Data sources mapped in the sub-tree will be evaluated relative to the referenced products(s) found by following the data path.

The ID source mapped to Accessory-ID mapping target will result in the IDs of the accessory products, and an Attribute source mapped to Accessory-Description will produce an attribute value from the accessory object.

Nesting Concept

A Data Path can be inserted below another Data Path, creating a nesting effect.

By mapping a Data Path data source to the composite Image target, configured with the path (Reference Image, Reference Target), two things will happen:

- The Image sub-tree will be evaluated and output once for each reference of type Image found from the accessory product found by the parent Data Path source.
- Data sources mapped in the sub-tree will be evaluated relative to the referenced asset(s) found by following the Data Path.

An ID source mapped to Image-ID will produce the IDs of the asset associated with the accessory product, and a Name source mapped to Image-Name will produce the name of the asset.

Valid Path Step Types

Configuring the path of a Data Path source is made up of a number of path steps. Each step describes how to reach a new set of references or objects by moving from the references / objects reached by the previous step.

The first step in a top-level Data Path source is taken relative to the object being processed for export. Following steps, either in the same path or the path of a nested Data Source, are taken relative to the set of object or references produced by the preceding step.

The Data Path data source supports different types of path steps. Starting from the objects selected for export, it is valid to configure Data Path sources with initial steps of type Parent, Reference and Override Sub Product. Following steps, including steps configured for nested Data Path sources, are only valid according to the following rules:

Step	Reach	Valid followed by
Parent	Parent node from node	Parent
Reference	Reference of a specific type from node	Reference Target
Reference Target	Target of a reference from a reference	Reference
Override Sub Product	Adopted children of a product-override node	Reference

Note: Although the mapping step of the Data Export wizard does not prevent the user from configuring invalid paths, paths that are not valid will result in an undefined exported.

Data Sources Available for Data Paths

The data sources that are to be mapped below a Data Path source require special support for evaluating relative to the objects / references identified by the paths. The following standard data sources support such nesting:

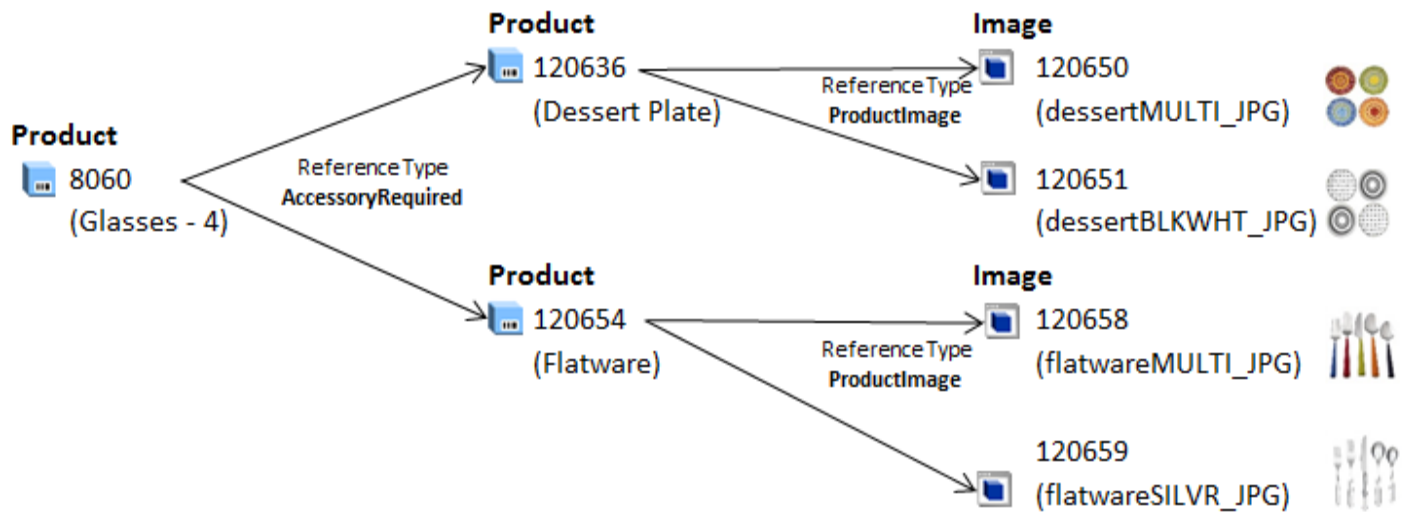
- ID
- Name
- Parent ID
- Object-Type

- Constant Value
- Attribute
- Attribute Group

Note: Data sources without special support evaluate relative to the object being processed for export even when placed below a Data Path source.

Nestable Data Paths in Generic XML Example

In the following example, the data set contains a product which has multiple references of type **AccessoryRequired**, to accessory products. Each accessory product has multiple references of type **ProductImage**, to assets.



Using Nested Data Paths when mapping will produce an export that includes all of the IDs shown above, in the following levels:

- Level 1 Product tag including the ID and Name of the product selected for export
- Level 2 Accessory tag including the ID, Name and Description for each product with an **AccessoryRequired** reference type
- Level 3 Image tag including the ID and Name for each image with a **ProductImage** reference type

Template

Choose **Generic XML** for **Format** and provide the following template text in the Sample field.

```

<Products>
  <Product>
    <?Product?>
    <ID><?Target?></ID>
    <Name><?Target?></Name>
    <Accessory>
      <?MultiTarget?>
      <ID><?Target AccessoryReqID?></ID>
      <Name><?Target AccessoryReqName?></Name>
      <Description><?Target AccessoryReqDescription?></Description>
    </Accessory>
  </Product>
</Products>
  
```

```

<Image>
  <?MultiTarget?>
  <ID><?Target ProductImageID?></ID>
  <Name><?Target ProductImageName?></Name>
</Image>
</Accessory>
</Product>
</Products>

```

Data Path Configuration

The mapping target structure resulting from the above template is:

Target/Structure	Data Source	Configuration
ID	ID	
Name	Name	
Accessory	Data Path	Data Path=Reference 'AccessoryRequired', Reference Target (composite, multiple-cardinality)
AccessoryReqID	ID	
AccessoryReqName	Name	
AccessoryReqDescription	Attribute	Select Attribute='Description, Web' Value and unit
Image	Data Path	Data Path=Reference 'ProductImage', Reference Target (composite, multiple-cardinality)
ProductImageID	ID	
ProductImageName	Name	

Mapping

Map Data

<ID>

<Name>

<Parent ID>

<Object Type Name>

<Product-Override Child ID>

<AttributeLinks>

<Is deleted>

"Constant Value"

<Page Number>

+ All Attributes

Select Attribute

+ Classifications

+ Index Words

+ Product Classification Links

+ Product References

+ Asset References

+ Classification References

+ Entity References

+ STEP Workflow Task Info

Multi level References

Inherit Data and References

☒

Converts to a generic XML format based on a sample.

▶ ID <ID> ID

▶ Name <Name>

▶ Accessory (1 mapped)

'AccessoryRequired'/Target/

▶ AccessoryReqID <ID> ID

▶ AccessoryReqName <Name>

▶ AccessoryReqDescription Description, Web Value and unit

▶ Image (1 mapped)

'Product Image'/Target/

▶ ProductImageID <ID> ID

▶ ProductImageName <Name>

For step-by-step mapping instructions, see the **Mapping for Nestable Data Paths in Generic XML Example** documentation.

© Stibo Systems - Confidential - STEP Trailblazer 8.3

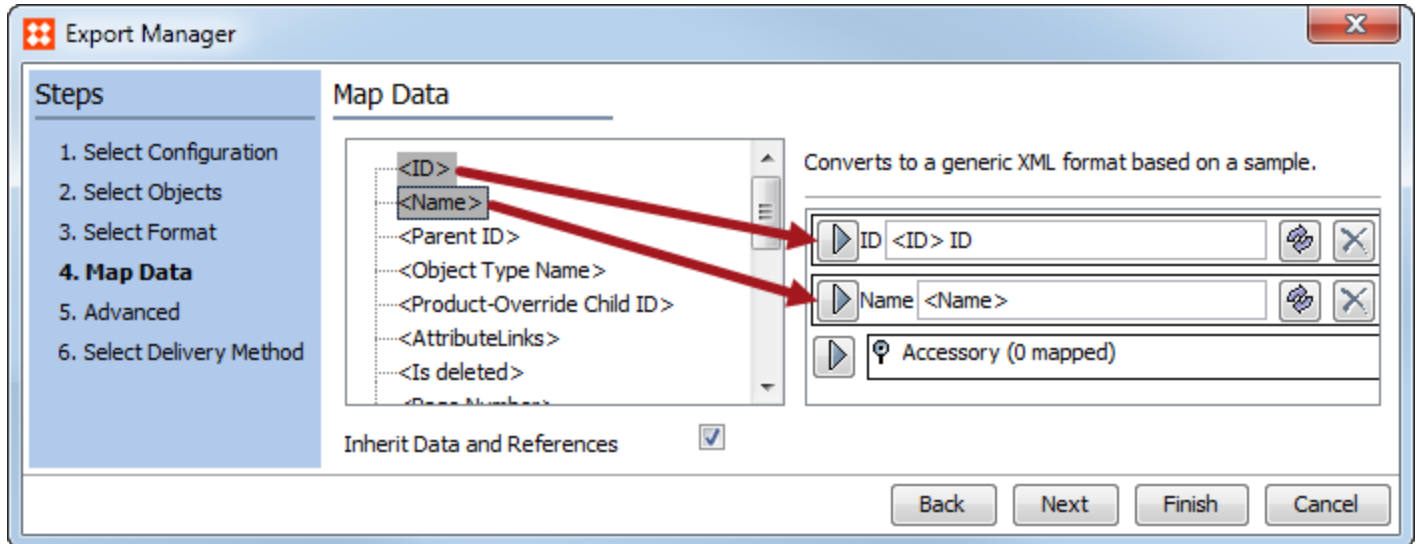
259

Results

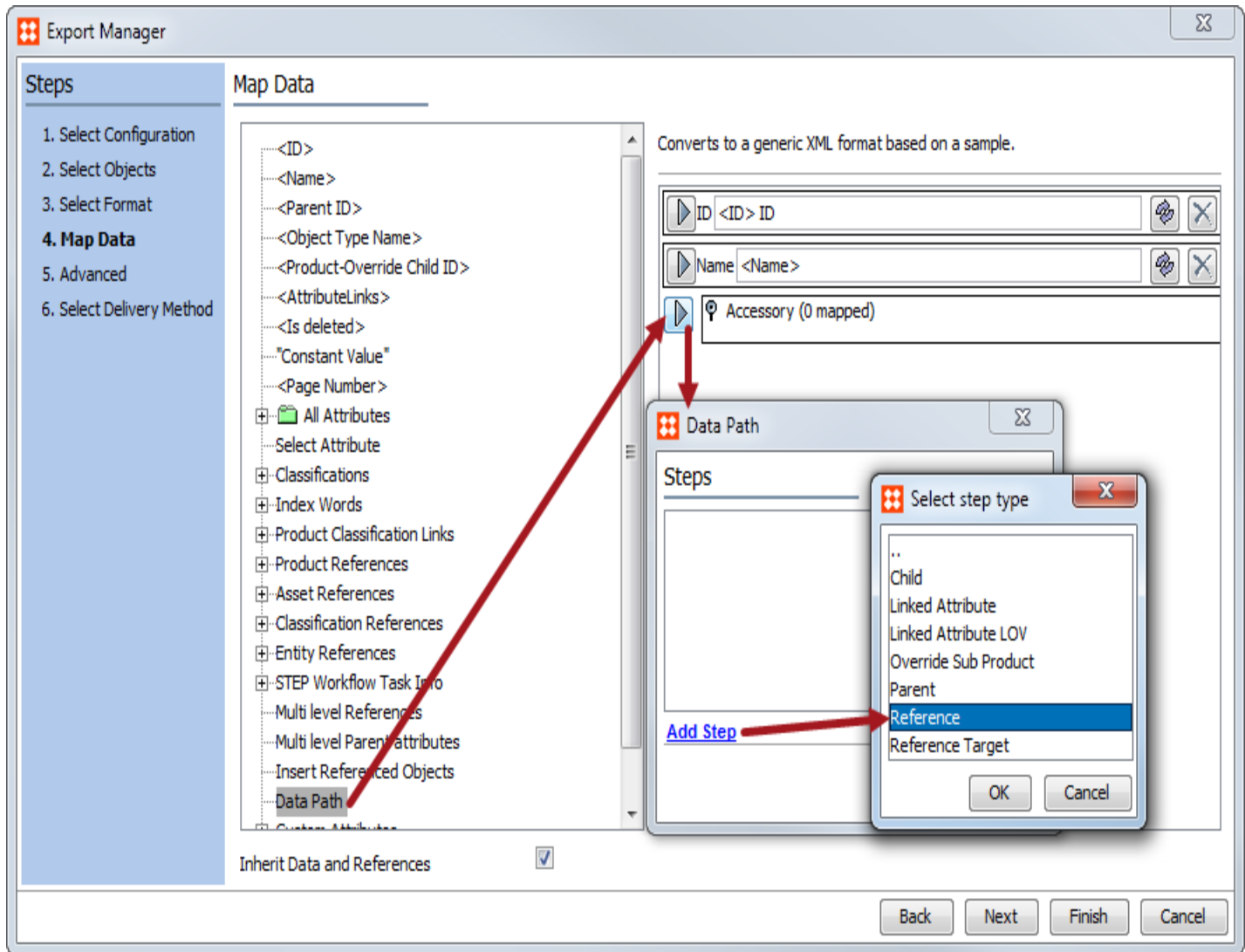
```
<Products>
  <Product>
    <ID>8060</ID>
    <Name>Glasses - 4</Name>
    <Accessory>
      <ID>120636</ID>
      <Name>Dessert Plate</Name>
      <Description>Set of 4 festive dessert plates.</Description>
      <Image>
        <ID>120650</ID>
        <Name>dessertMULTI_JPG</Name>
      </Image>
      <Image>
        <ID>120651</ID>
        <Name>dessertBLKWHT_JPG</Name>
      </Image>
    </Accessory>
    <Accessory>
      <ID>120654</ID>
      <Name>Flatware</Name>
      <Description>Five-piece, single place setting.</Description>
      <Image>
        <ID>120658</ID>
        <Name>flatwareMULTI_JPG</Name>
      </Image>
      <Image>
        <ID>120659</ID>
        <Name>flatwareSILVR_JPG</Name>
      </Image>
    </Accessory>
  </Product>
</Products>
```


Mapping for Nestable Data Paths in Generic XML Example

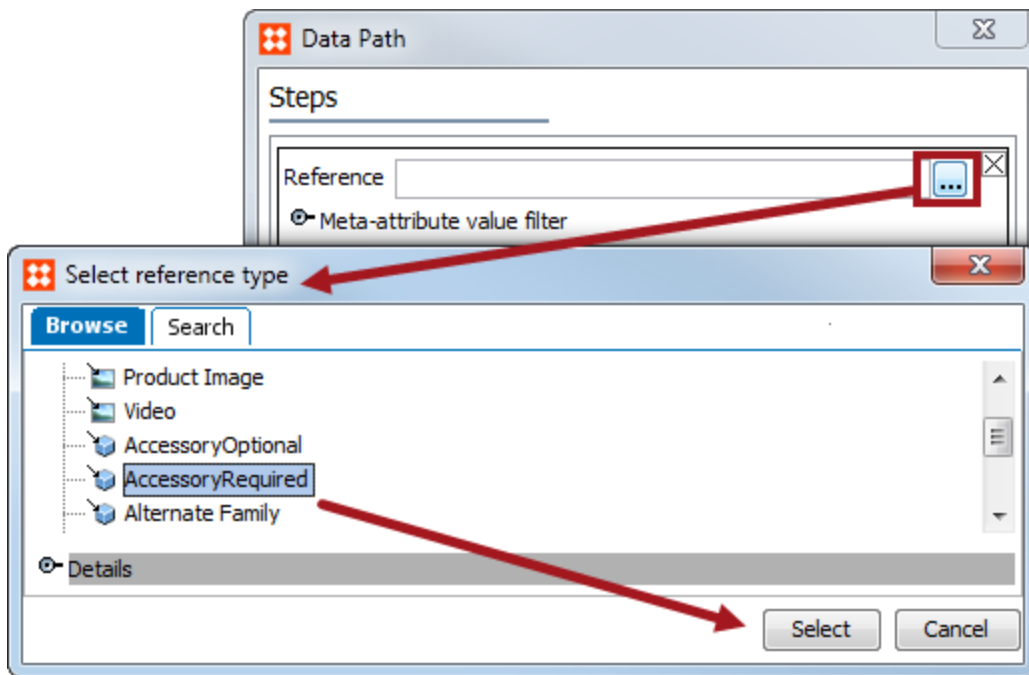
1. Map **ID** and **Name** for Level 1.



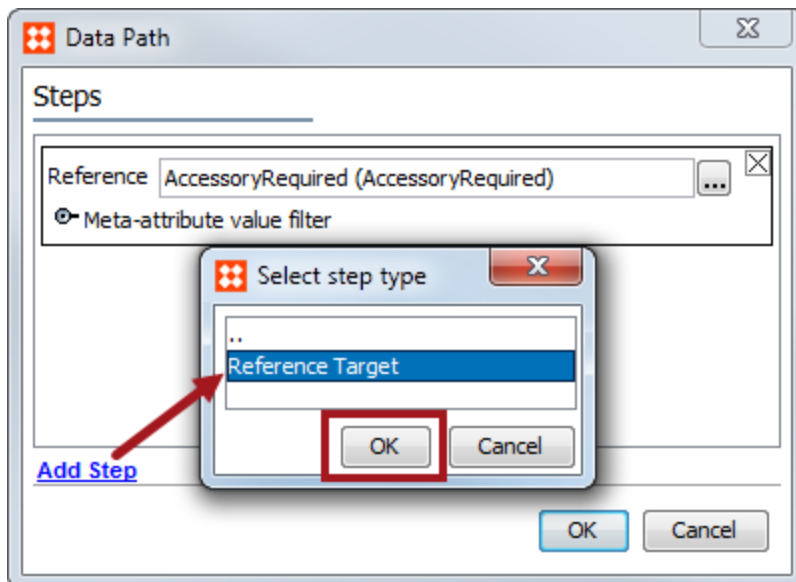
2. Map **Data Path** to Accessory, click the **Add Step** link, select the **Reference** step type, and click **OK**.



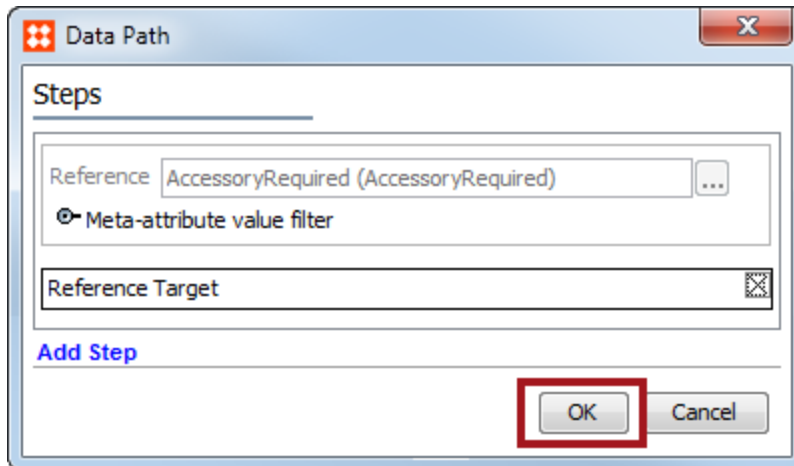
3. On the Data Path dialog, click the ellipsis button (...), select AccessoryRequired reference type, and click the **Select** button.



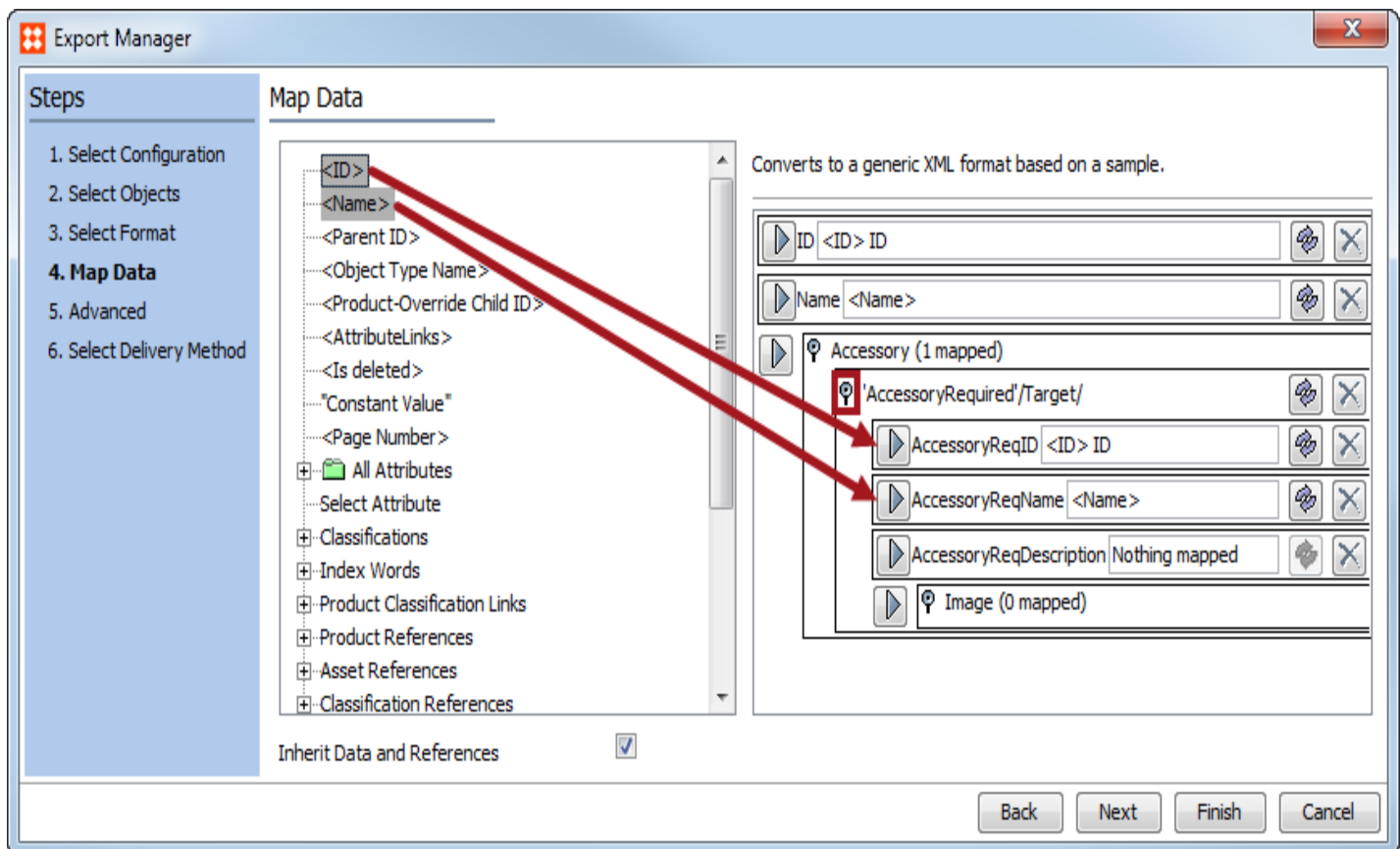
4. Click the **Add Step** link, select **Reference Target** for Accessory and click **OK**.



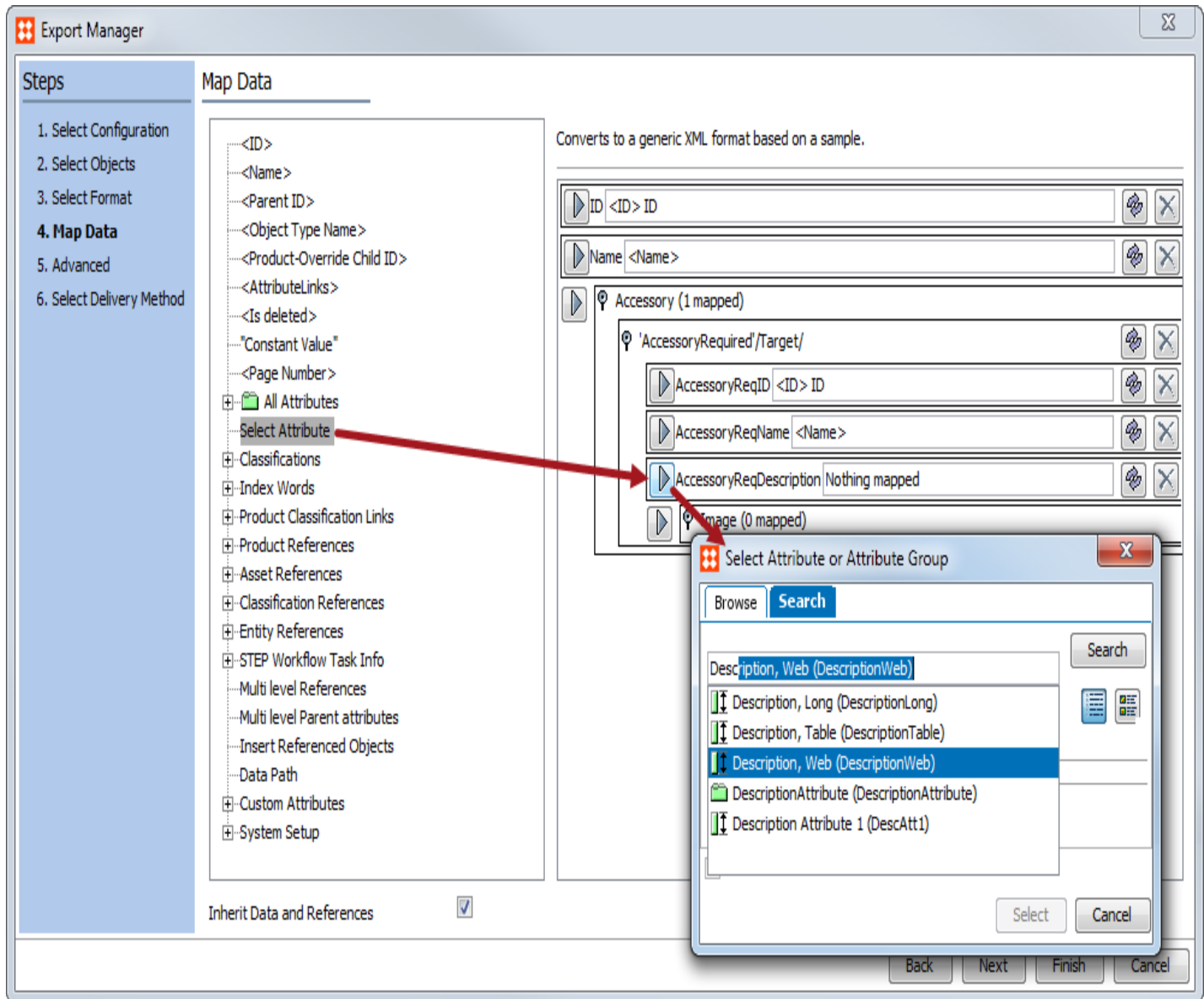
5. Click **OK** to confirm the Reference and Reference Target Data Path fields for the Accessory.



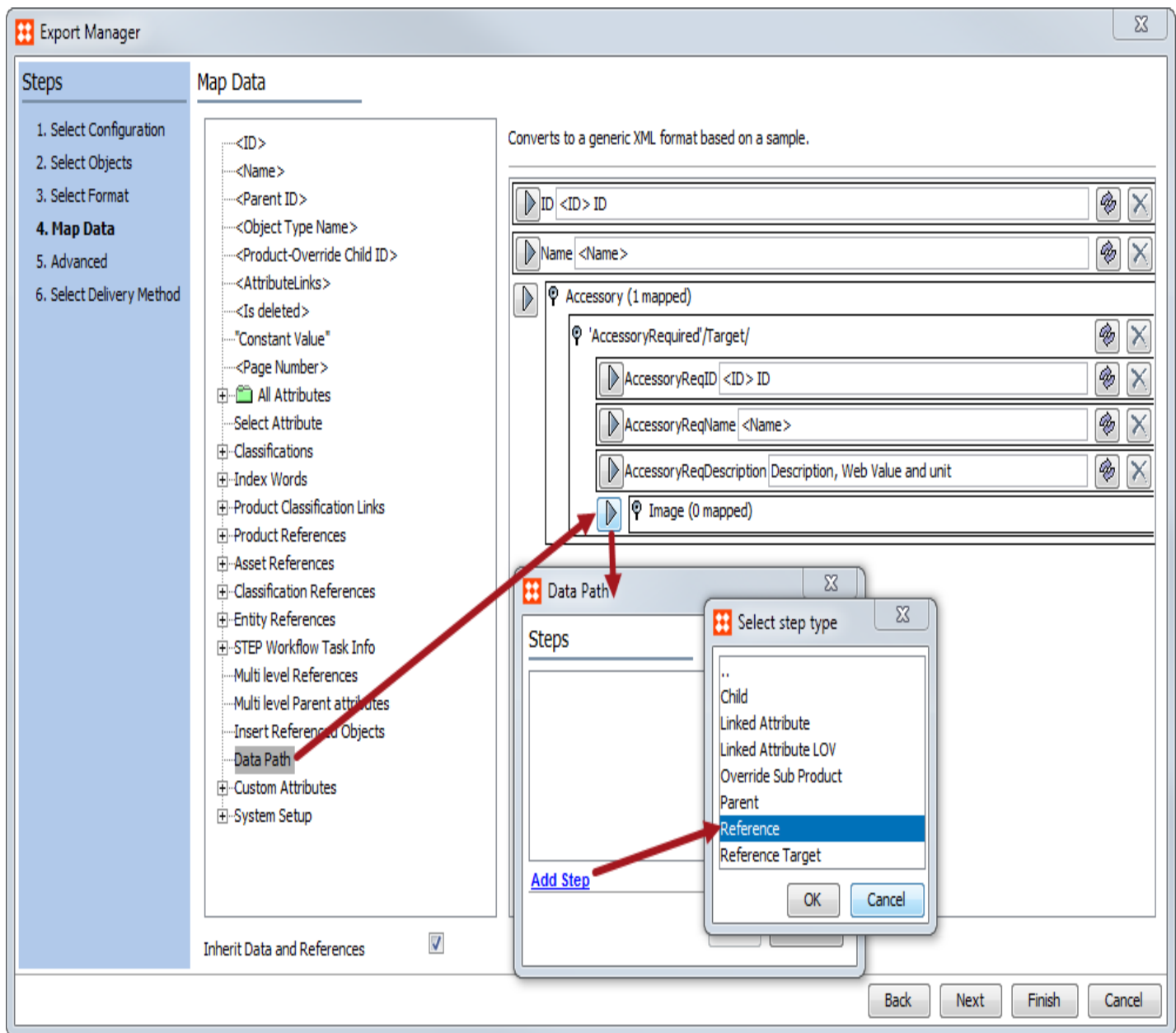
6. Open the 'AccessoryRequired'/Target/ flipper to display the nested mapping options. Map **ID** and **Name** for the Level 2 products being exported.



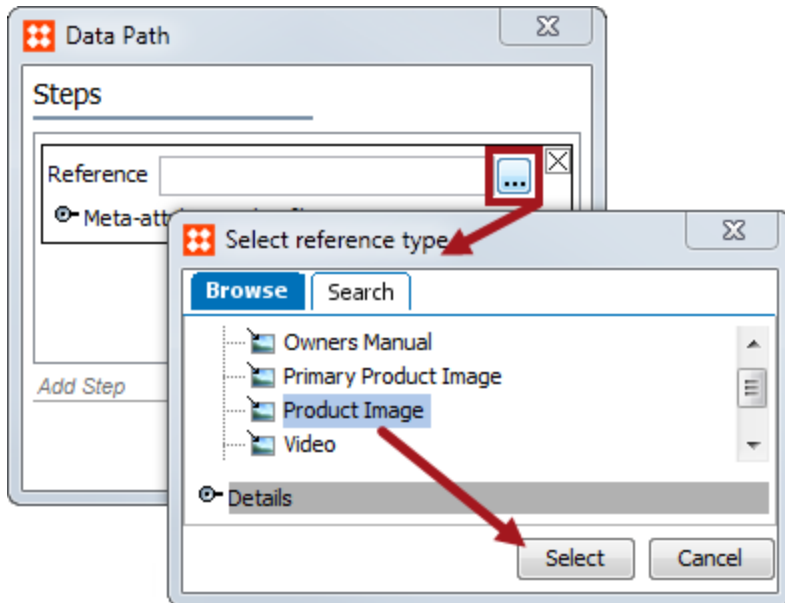
7. Use **Select Attribute** to map 'Description, Web' for Level 2 (products) and click the **Select** button.



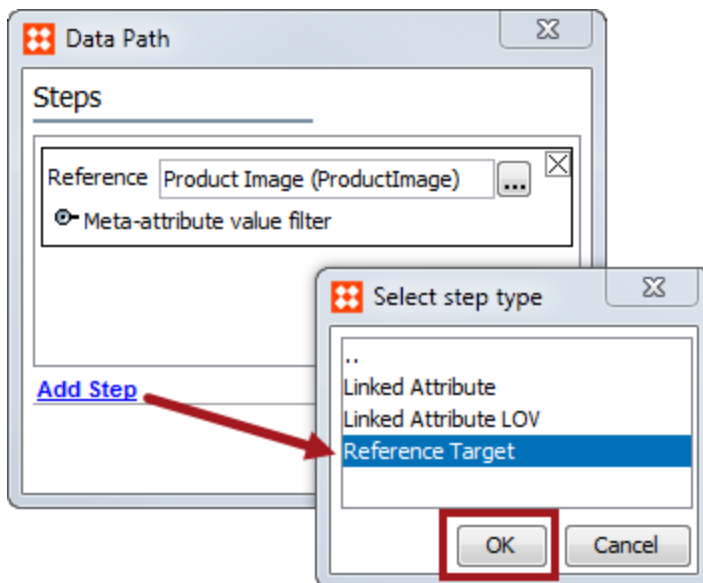
8. Map **Data Path** to Image, click the **Add Step** link, select the **Reference** step type, and click **OK**.



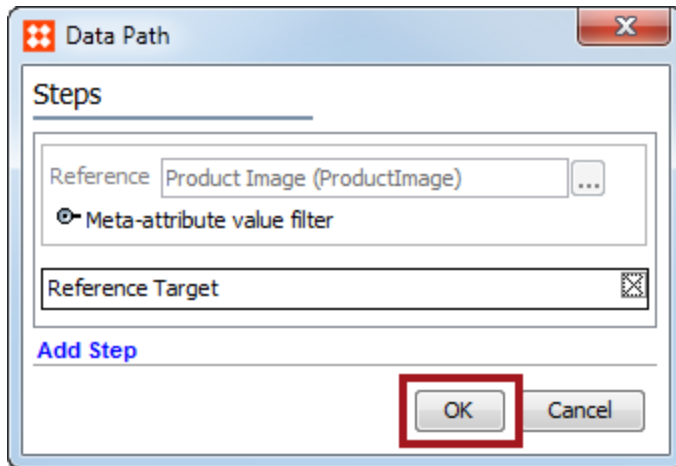
- On the Data Path dialog, click the ellipsis button (...), select ProductImage reference type, and click the **Select** button.



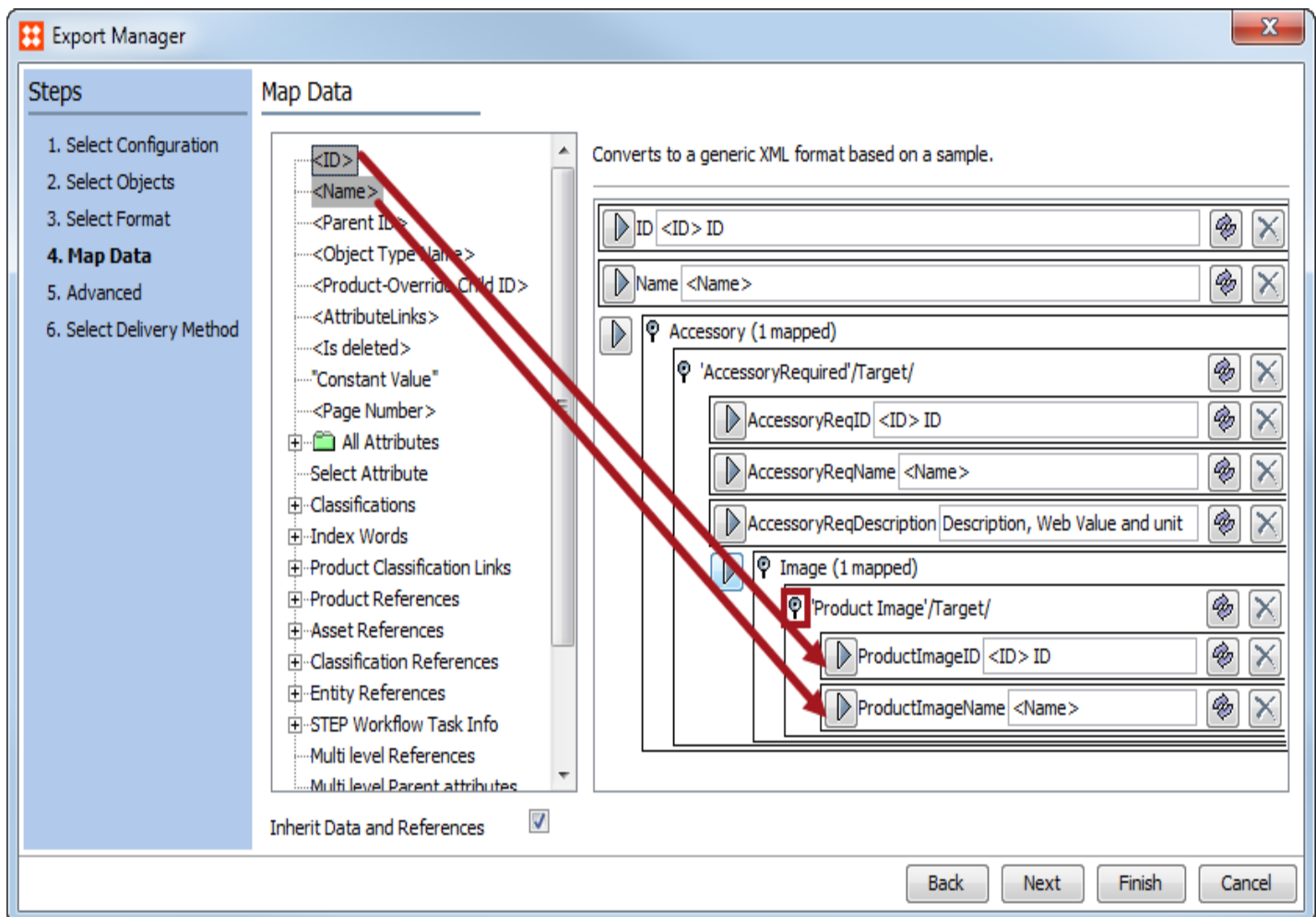
10. Click the **Add Step** link, select **Reference Target** for Image and click **OK**.



11. Click **OK** to confirm the Reference and Reference Target for the Image.



12. Open the 'ProductImage'/Target/ flipper to display the nested mapping options. Map **ID** and **Name** for Level 3 (images).



All mapping is complete.

Map Data

<ID>

<Name>

<Parent ID>

<Object Type Name>

<Product-Override Child ID>

<AttributeLinks>

<Is deleted>

Constant Value

<Page Number>

+ All Attributes

Select Attribute

+ Classifications

+ Index Words

+ Product Classification Links

+ Product References

+ Asset References

+ Classification References

+ Entity References

+ STEP Workflow Task Info

Multi level References

Converts to a generic XML format based on a sample.

ID <ID> ID

Name <Name>

Accessory (1 mapped)

'AccessoryRequired'/Target/

AccessoryReqID <ID> ID

AccessoryReqName <Name>

AccessoryReqDescription Description, Web Value and unit

Image (1 mapped)

'Product Image'/Target/

ProductImageID <ID> ID

ProductImageName <Name>

Inherit Data and References

RemoveIfEmpty Instruction in Generic XML

In an outbound Generic XML template, the Generic XML instruction **RemoveIfEmpty** makes it possible to remove the open and close tags surrounding a value if there is no value for any one of the attributes in the section.

The **RemoveIfEmpty recursive** instruction allows you to remove an entire section of the output. This means that if a section has no values for any of the attributes, the section is eliminated from the file. The 'RemoveIfEmpty recursive' tag is useful when a whole section should be excluded when no values are found. It should not be used if the section includes only mandatory attributes since the output would be incomplete.

If the open and close tags of an individual line should remain even when no value is supplied, use the 'Allow empty tags' option, described in the **Allow Empty Tags Parameter in Generic XML** section.

Objects

The objects selected for export includes two products, one without either of the optional values (Manufacturer's part Number and T-shirt Size) and the other with only one of the optional attributes.

Tree

Primary Product Hierarchy

Products

Apparel

Upper Body Wear

T-shirts

T-shirts Items

Cotton T-Shirts

18216 L O

18210 M B

Products

Products

References

Referenced By

View: RemoveIfEmpty

ID	Name	Provider GLN	Parent	Brand Name	Manufacturer's Part Number	T-shirt Size
181951	18216 L O	0823335000008	Cotton T-Shirts	Hanes		
18210	18210 M B		Cotton T-Shirts	Hanes	H5250	

Template

Choose **Generic XML** format and provide the following template text in the Sample field.

With RemoveIfEmpty	With 'RemoveIfEmpty recursive'
<pre><Products> <Product> <?Record?> <ID><?Target?></ID> <NAME><?Target?></NAME> <mandatoryAttributes> <Parent><?Target?></Parent> <BrandName><?Target?></BrandName></pre>	<pre><Products> <Product> <?Record?> <ID><?Target?></ID> <NAME><?Target?></NAME> <mandatoryAttributes> <Parent><?Target?></Parent> <BrandName><?Target?></BrandName></pre>

With RemoveIfEmpty	With 'RemoveIfEmpty recursive'
<pre> <ProviderGLN><?Target?></ProviderGLN> </mandatoryAttributes> <optionalAttributes> <?RemoveIfEmpty?> <PartNo><?Target?></PartNo> <Size><?Target?></Size> </optionalAttributes> </Product> </Products> </pre>	<pre> <ProviderGLN><?Target?></ProviderGLN> </mandatoryAttributes> <optionalAttributes> <?RemoveIfEmpty recursive?> <PartNo><?Target?></PartNo> <Size><?Target?></Size> </optionalAttributes> </Product> </Products> </pre>

- The RemoveIfEmpty and/or 'RemoveIfEmpty recursive' tags are not used below the 'mandatoryAttributes' section since all attributes should be output, even if they are empty.
- The RemoveIfEmpty and/or 'RemoveIfEmpty recursive' tags are included below the 'optionalAttributes' section since that information is only required if one or more of the targets have values.

Set **Allow empty tags** to **No**. If set to Yes, the 'optionalAttributes' section that is using 'RemoveIfEmpty recursive' instruction will output even when all values are blank.

Select Format

Generic XML

Converts to a generic XML format based on a sample.

Sample

```

<Products>
<Product>
<?Record?>
<ID><?Target?></ID>
<NAME><?Target?></NAME>
<mandatoryAttributes>
    <Parent><?Target?></Parent>
    <BrandName><?Target?></BrandName>
    <ProviderGLN><?Target?></ProviderGLN>
</mandatoryAttributes>

```

DocType

Allow empty tags

No

Mapping

Use the **Select Attribute** aspect for the attribute data targets after ID and Name.

Map Data

Converts to a generic XML format based on a sample.

☒ Inherit Data and References

<ID>
 <Name>
 <Parent ID>
 <Object Type Name>
 <Product-Override Child ID>
 <AttributeLinks>
 <Is deleted>
 "Constant Value"
 <Page Number>
 + All Attributes
 Select Attribute
 + Classifications
 + Index Words
 + Product Classification Links
 + Product References

ID <ID> ID
 NAME <Name>
 Parent Parent Value and unit
 BrandName Brand Name Value and unit
 ProviderGLN Provider GLN Value and unit
 PartNo Manufacturer's Part Number Value and unit
 Size T-shirt Size Value and unit

Results

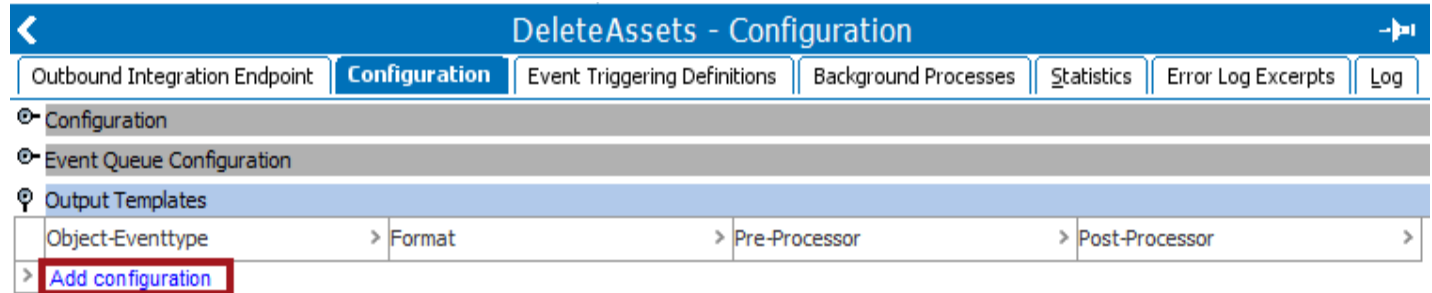
With RemoveIfEmpty	With 'RemoveIfEmpty recursive'
<pre> <Products> <Product> <ID>181951</ID> <NAME>18216 L O</NAME> <mandatoryAttributes> <Parent>Cotton T-Shirts</Parent> <BrandName>Hanes</BrandName> <ProviderGLN>0823335000008</ProviderGLN> </mandatoryAttributes> </Product> <Product> <ID>18210</ID> <NAME>18210 M B</NAME> <mandatoryAttributes> <Parent>Cotton T-Shirts</Parent> <BrandName>Hanes</BrandName> <ProviderGLN/> </mandatoryAttributes> </Product> </Products> </pre>	<pre> <Products> <Product> <ID>181951</ID> <NAME>18216 L O</NAME> <mandatoryAttributes> <Parent>Cotton T-Shirts</Parent> <BrandName>Hanes</BrandName> <ProviderGLN>0823335000008</ProviderGLN> </mandatoryAttributes> </Product> <Product> <ID>18210</ID> <NAME>18210 M B</NAME> <mandatoryAttributes> <Parent>Cotton T-Shirts</Parent> <BrandName>Hanes</BrandName> <ProviderGLN/> </mandatoryAttributes> <optionalAttributes> <PartNo>H5250</PartNo> </optionalAttributes> </Product> </Products> </pre>

Reporting Deleted Assets with a Generic XML OIEP

When assets are deleted and approved in STEP, Outbound Integration Endpoints (OIEPs) can pass deletion events when the OIEP is configured to listen for Delete events on the relevant objects. As a result, the downstream system is alerted to the deleted asset.

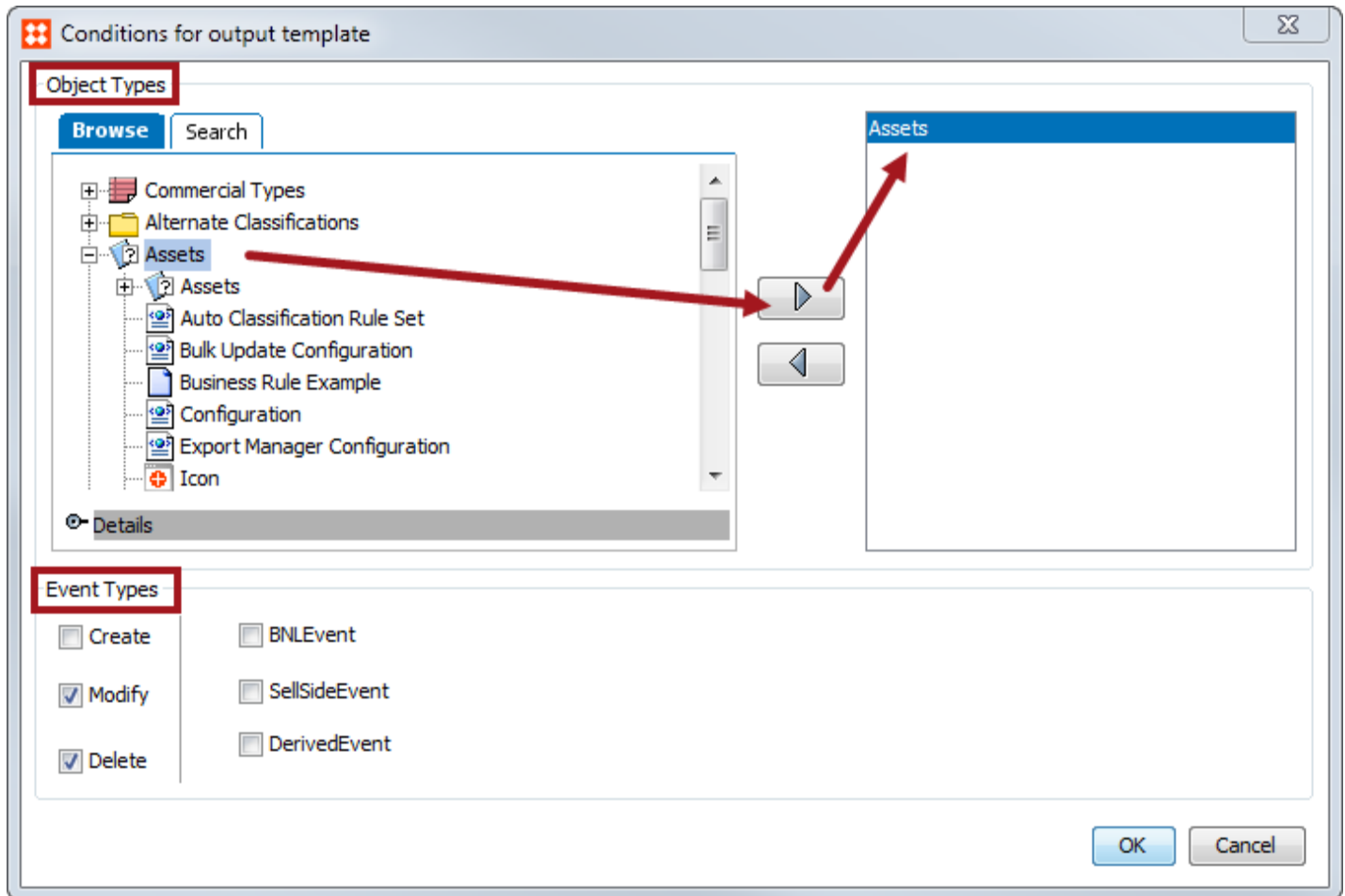
Object Type and Event Type Selection

On the OIEP, open the Configuration tab. Under the Output Templates flipper, click the **Add configuration** link.



Set the Object Types and Event Types.

For this example, Modify events are also included to demonstrate the available functionality.



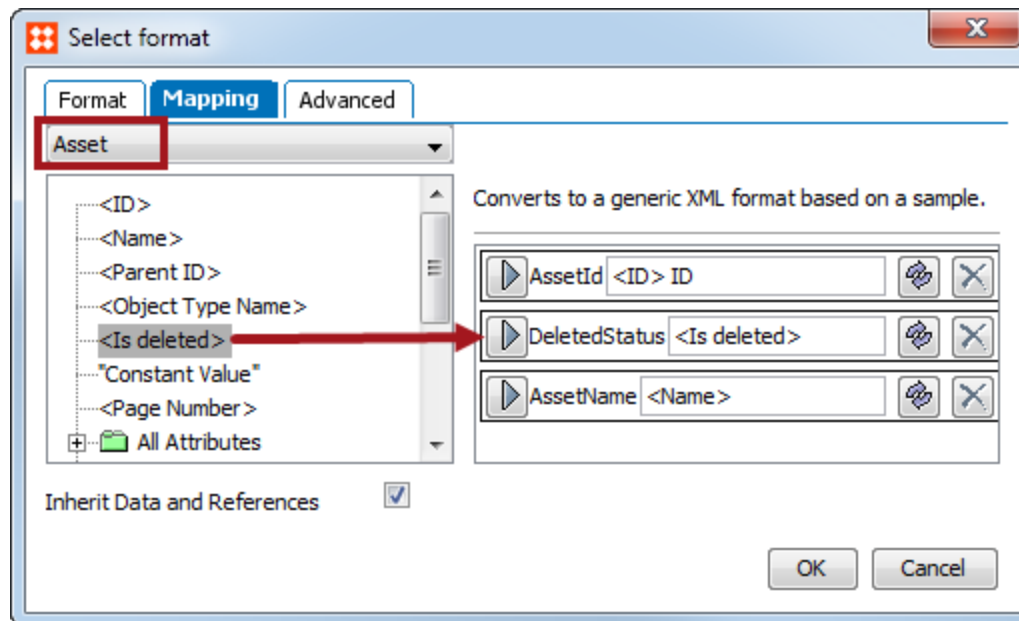
Template

In the Output Template, click the ellipsis button (...), and set the format to **Generic XML**.

Provide the following text in the Sample field.

```
<root>
  <Assets>
    <?Record?>
      <Asset AssetID="[/Target AssetId]" DeletedStatus="[/Target DeletedStatus?]">
        <AssetName> <?Target?> </AssetName>
      </Asset>
    </Assets>
  </root>
```

Mapping



Results

Selecting **Approve Deletion** for an Asset in the Recycle Bin and/or **Approve Object** for modification of an Asset on the Tree triggers the OIEP.

Invoking the OIEP generates the following output:

```
<root>
  <Assets>
    <Asset AssetID="MSDS_00020" DeletedStatus="IsDeleted">
      <AssetName/>
    </Asset>
  </Assets>
  <Assets>
    <Asset AssetID="MSDS_00019" DeletedStatus="">
      <AssetName>MSDS_00019 modified</AssetName>
    </Asset>
  </Assets>
</root>
```

For deleted Assets, the Deleted Status displays **"IsDeleted"**. Only ID is retained for export on Delete events.

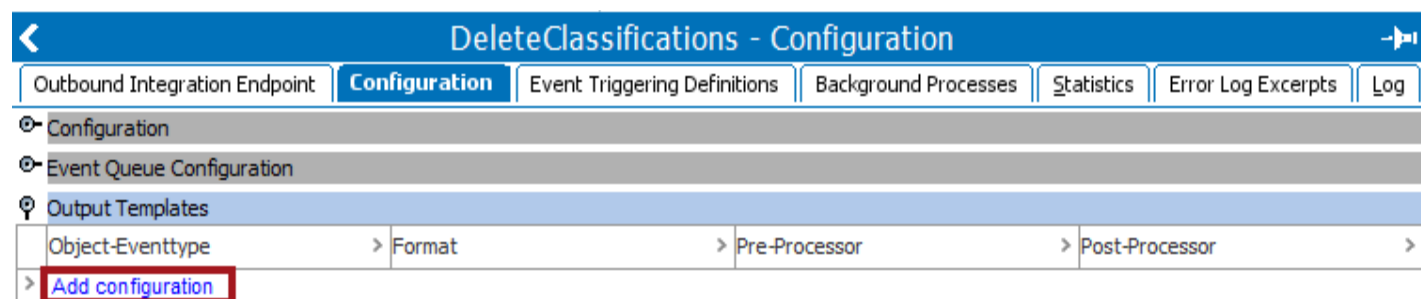
For modified Assets, the Deleted Status displays "", which indicates the asset is not deleted. All mapped fields are exported for Create and Modify events.

Reporting Deleted Classifications with a Generic XML OIEP

When Classifications are deleted and approved in STEP, Outbound Integration Endpoints (OIEPs) can pass deletion events when the OIEP is configured to listen for Delete events on the relevant objects. As a result, the downstream system is alerted to the deleted classification.

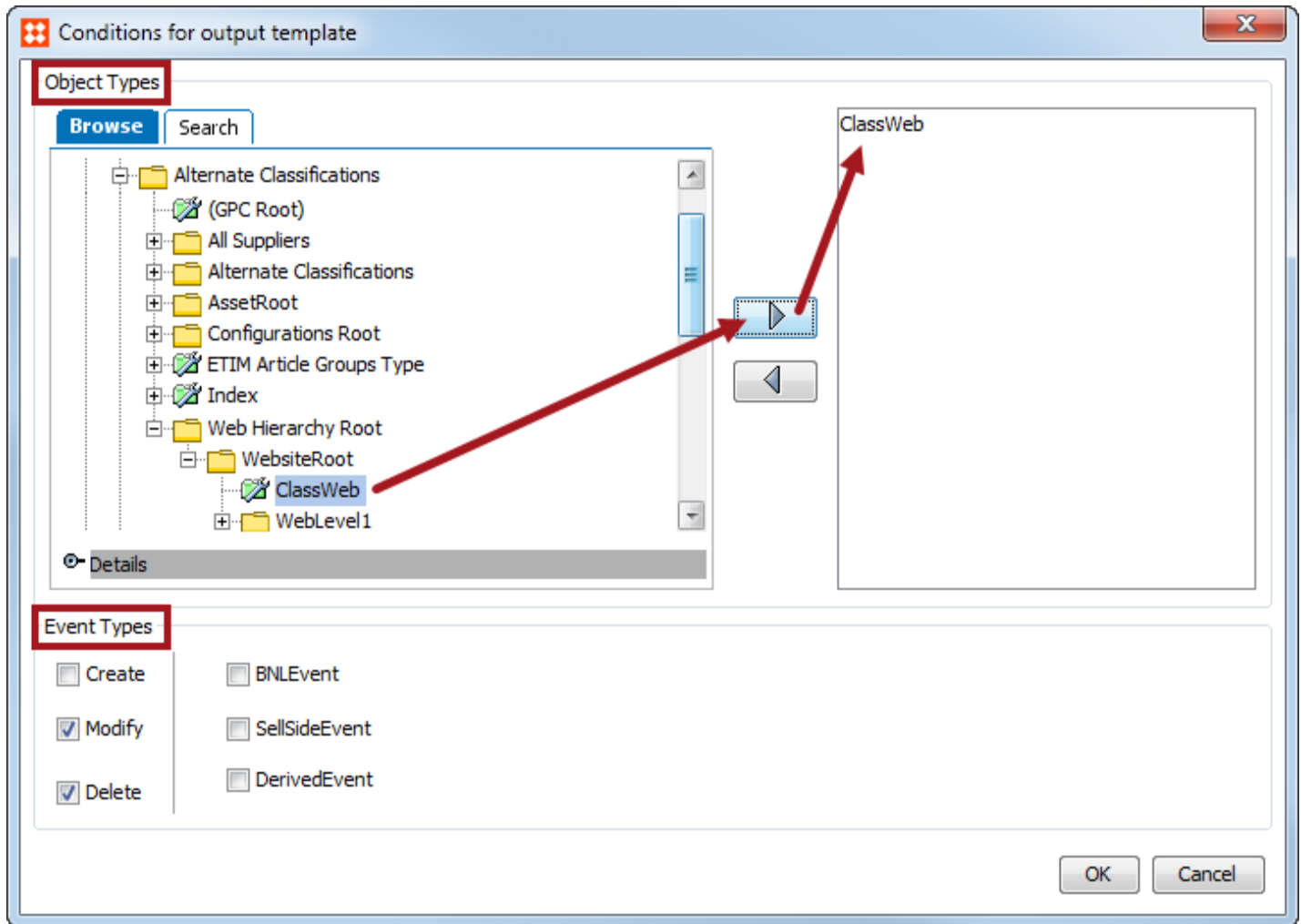
Object Type and Event Type Selection

On the OIEP, open the Configuration tab. Under the Output Templates flipper, click the **Add configuration** link.



Set the Object Types and Event Types.

For this example, Modify events are also included to demonstrate the available functionality.



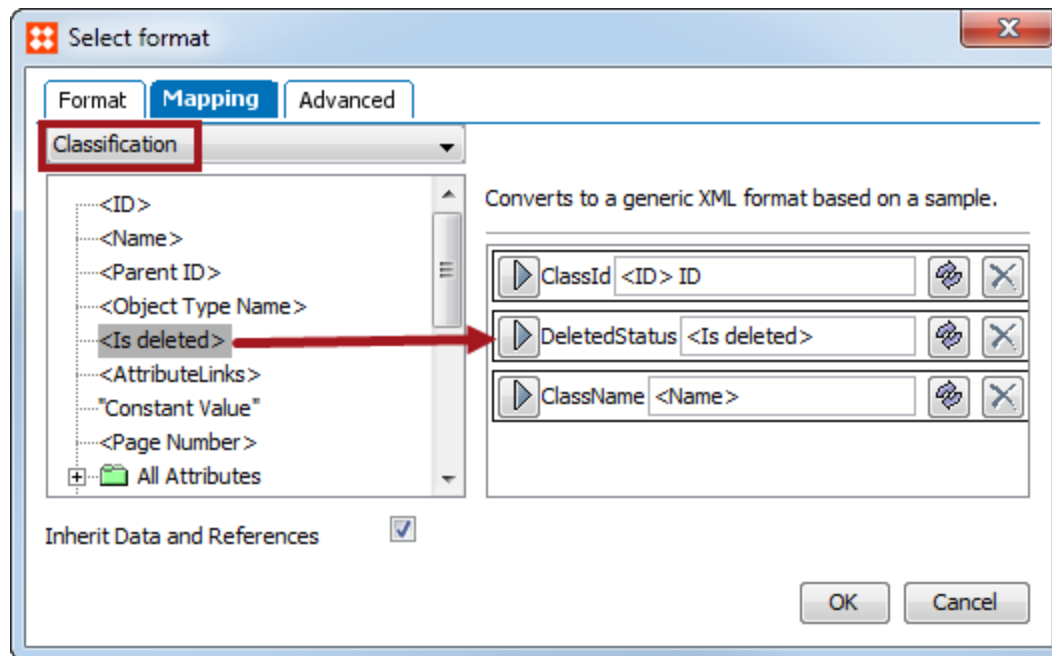
Template

In the Format field, click the ellipsis button (...), and set the format to **Generic XML**.

Provide the following text in the Sample field.

```
<root>
  <Classifications>
    <?Record?>
    <Class ClassID="[/Target ClassId?]" DeletedStatus="[/Target DeletedStatus?]">
      <ClassName> <?Target?> </ClassName>
    </Class>
  </Classifications>
</root>
```

Mapping



Result

Approve Deletion for a Classification in the Recycle Bin and/or **Approve Object** for modification of a Classification on the Tree triggers the OIEP.

Invoking the OIEP generates the following output:

```
<root>
  <Classifications>
    <Class ClassID="SofasAndChairs" DeletedStatus="IsDeleted">
      <ClassName/>
    </Class>
  </Classifications>
  <Classifications>
    <Class ClassID="TablesAndChairs" DeletedStatus="">
      <ClassName>Tables And Chairs modified</ClassName>
    </Class>
  </Classifications>
</root>
```

For deleted Classifications, the Deleted Status displays **"IsDeleted"**. Only ID is retained for export on Delete events.

For modified Classifications, the Deleted Status displays "", which indicates the classification is not deleted. All mapped fields are exported for Create and Modify events.

Reporting Deleted Entities with a Generic XML OIEP

When Entities are deleted and approved in STEP, Outbound Integration Endpoints (OIEPs) can pass deletion events when the OIEP is configured to listen for Delete events on the relevant objects. As a result, the downstream system is alerted to the deleted asset. Entities must have the Revisability field set to Workspace Revisable, not Global Revisable.

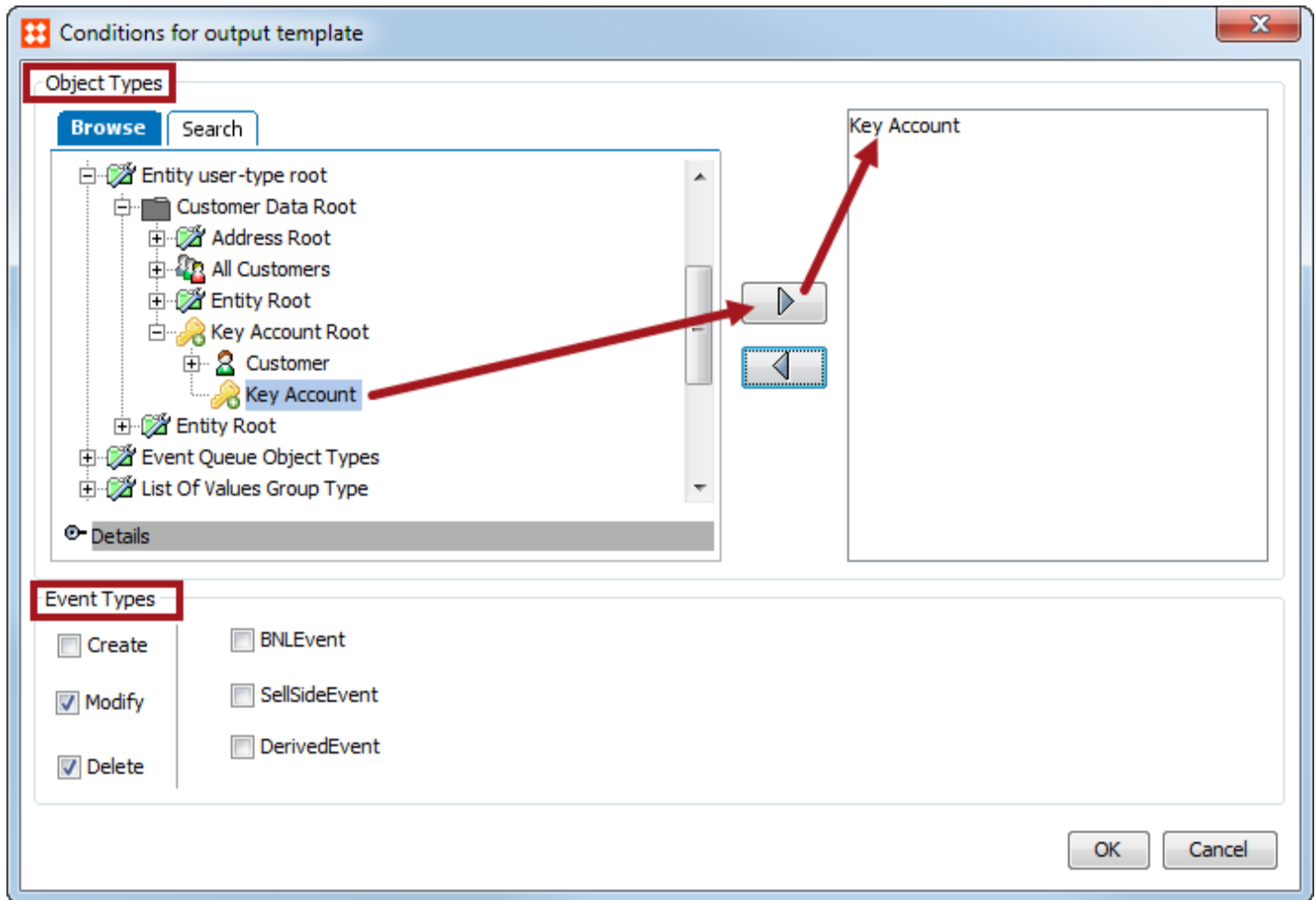
Object Type and Event Type Selection

On the OIEP, open the Configuration tab. Under the Output Templates flipper, click the **Add configuration** link.

Outbound Integration Endpoint	Configuration	Event Triggering Definitions	Background Processes	Statistics	Error Log Excerpts	Log
Configuration Event Queue Configuration Output Templates						
Object-Eventtype	>	Format	>	Pre-Processor	>	Post-Processor
> Add configuration						

Set the Object Types and Event Types.

For this example, Modify events are also included to demonstrate the available functionality.



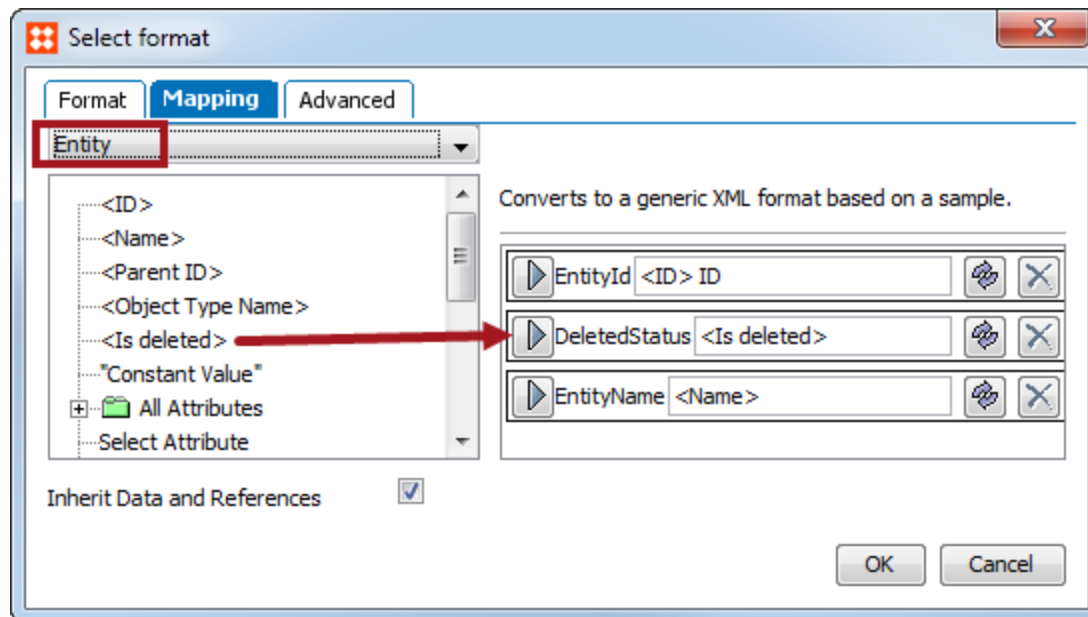
Template

In the Output Template, click the ellipsis button (...), and set the format to **Generic XML**.

Provide the following text in the Sample field.

```
<root>
  <Entities>
    <?Record?>
      <Entity EntityID="[/Target EntityId?]" DeletedStatus="[/Target DeletedStatus?]">
        <EntityName> <?Target?> </EntityName>
      </Entity>
    </Entities>
  </root>
```

Mapping



Results

Approve Deletion for an Entity in the Recycle Bin and/or **Approve Object** for modification of a Entity that is Workspace Revisable on the Tree triggers the OIEP.

Invoking the OIEP generates the following output:

```
<root>
  <Entities>
    <Entity EntityID="KeyB" DeletedStatus="IsDeleted">
      <EntityName/>
    </Entity>
  </Entities>
  <Entities>
    <Entity EntityID="KeyA" DeletedStatus="">
      <EntityName>Key A modified</EntityName>
    </Entity>
  </Entities>
</root>
```

For deleted Entities, the Deleted Status displays **"IsDeleted"**. Only ID is retained for export on Delete events.

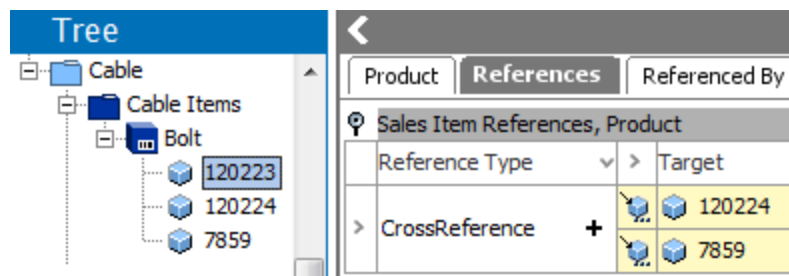
For modified Entities, the Deleted Status displays **" "**, which indicates the entity is not deleted. All mapped fields are exported for Create and Modify events.

Reporting Deleted Product References with a Generic XML OIEP

Outbound Integration Endpoints (OIEPs) can pass deletion events for Product References when the OIEP is configured to listen for Modify events on the relevant objects. The Modify Event Type is necessary since the object itself is not being deleted, but rather a reference on the object which continues to exist.

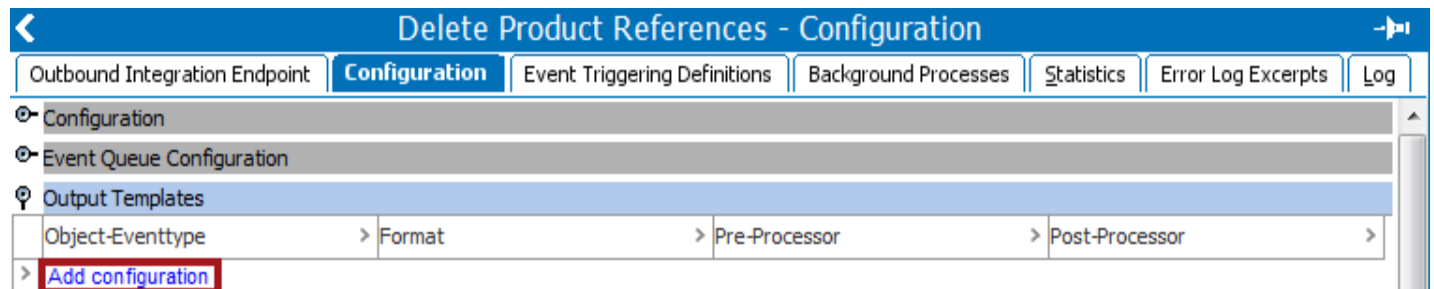
Product

On product 120223, the Reference Type of CrossReference to Target 7859 must be deleted in order to trigger the OIEP.

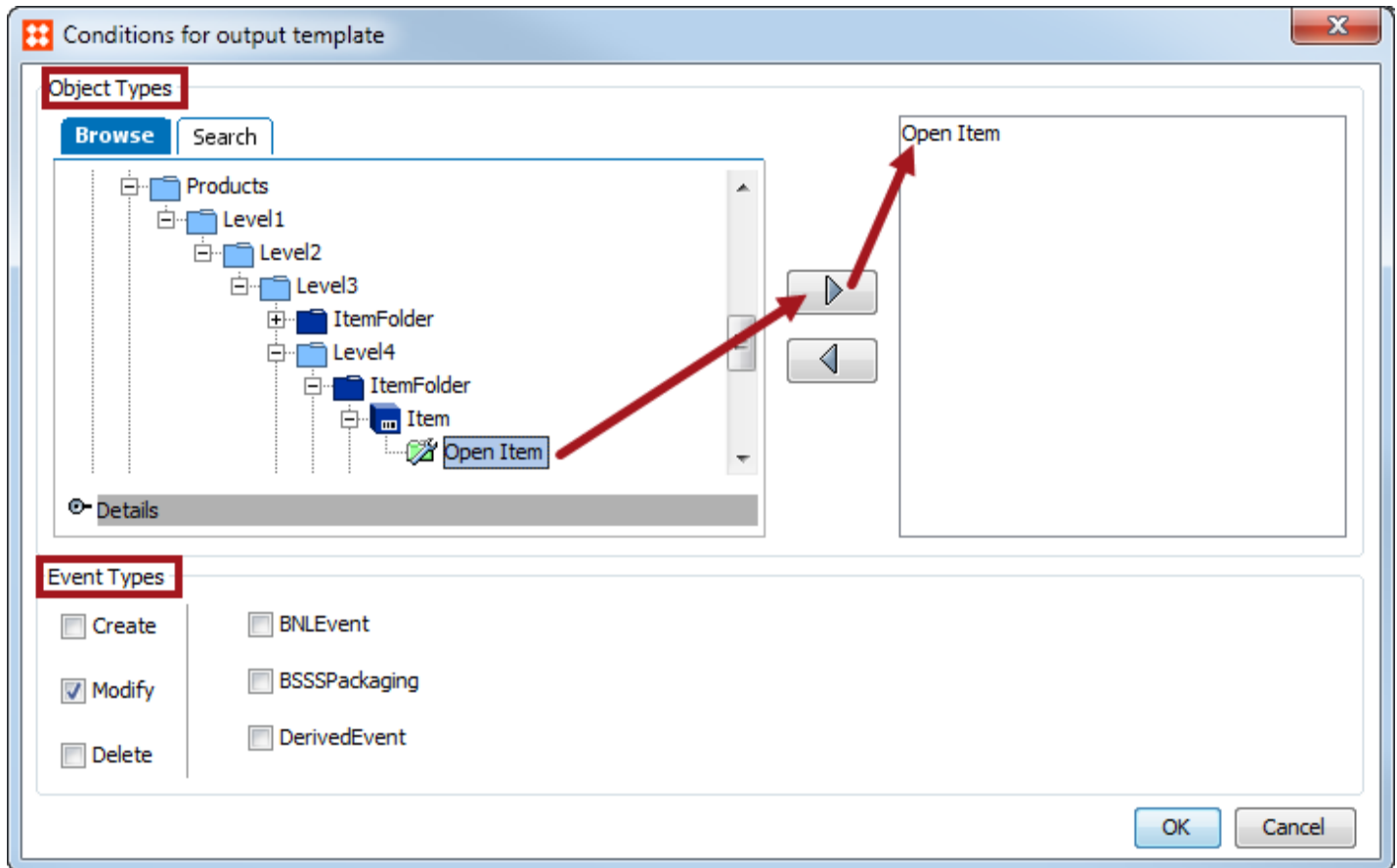


Object Type and Event Type Selection

On the OIEP, open the Configuration tab. Under the Output Templates flipper, click the **Add configuration** link.



Set the Object Types and Event Types.



Template

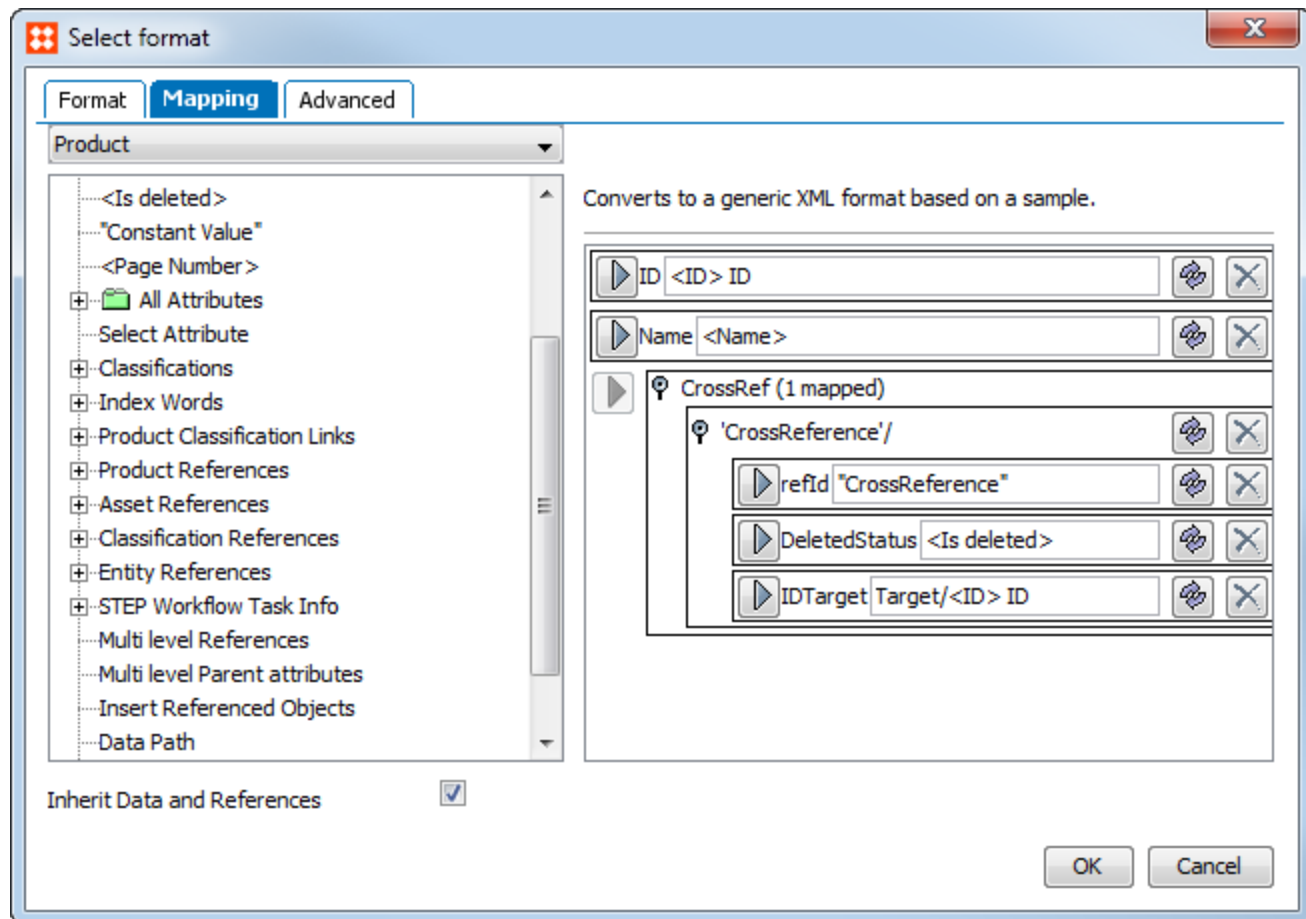
In the Output Template, click the ellipsis button (...), and set the format to **Generic XML**.

Provide the following text in the Sample field.

```
<Products>
  <Product ID="[/Target ID?]">
    <?Record?>
      <Name><?Target?></Name>
      <CrossRefs>
        <CrossRef refTypeID="[/Target refId?]" DeletedStatus="[/Target DeletedStatus?]">
          <?MultiTarget?>
            <IDTarget><?Target?></IDTarget>
          </CrossRef>
        </CrossRefs>
      </Product>
    </Products>
```

Mapping

The Constant Value 'CrossReference' is used to identify the type of reference being monitored. The <Is deleted> aspect monitors for deletions. Data Path mapping to the Target reference provides the deleted ID.



For detailed mapping steps, see the **Mapping to Report Deleted Product References with Generic XML OIEP** documentation.

Results

When the Product Reference for item 7859 is deleted and the product 120223 is approved, the OIEP is triggered. Invoking the OIEP generates the following file:

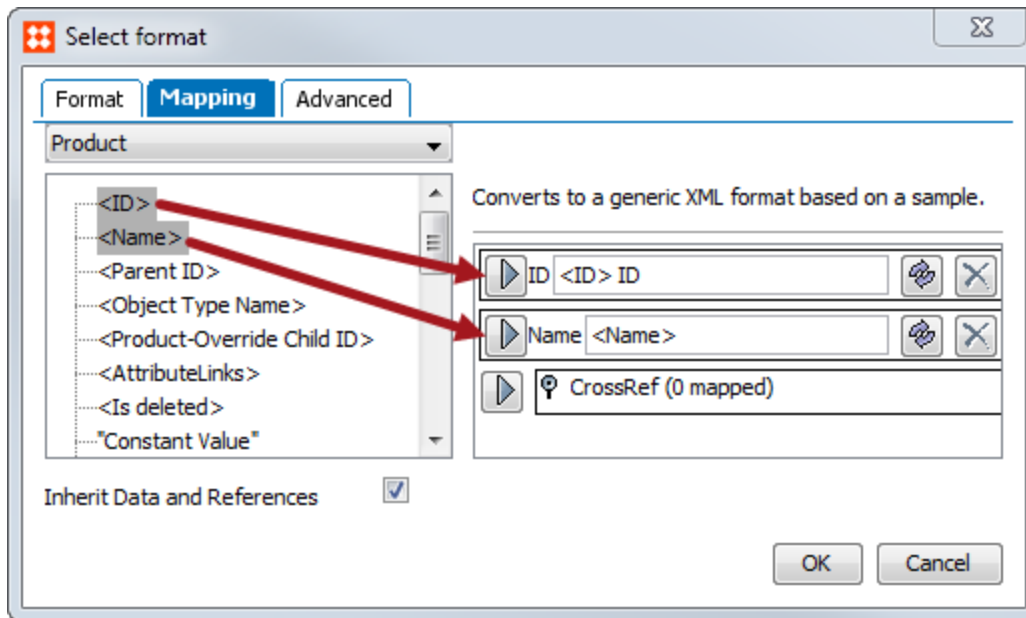

```
<Products>
  <Product ID="120223">
    <Name>120223</Name>
    <CrossRefs>
      <CrossRef refTypeID="CrossReference" DeletedStatus="">
        <IDTarget>120224</IDTarget>
      </CrossRef>
      <CrossRef refTypeID="CrossReference" DeletedStatus="IsDeleted">
        <IDTarget>7859</IDTarget>
      </CrossRef>
    </CrossRefs>
  </Product>
</Products>
```

Note: Both of the product references originally on the product are exported. The DeletedStatus target shows blank for the reference that still exists, and shows "IsDeleted" for the removed reference.

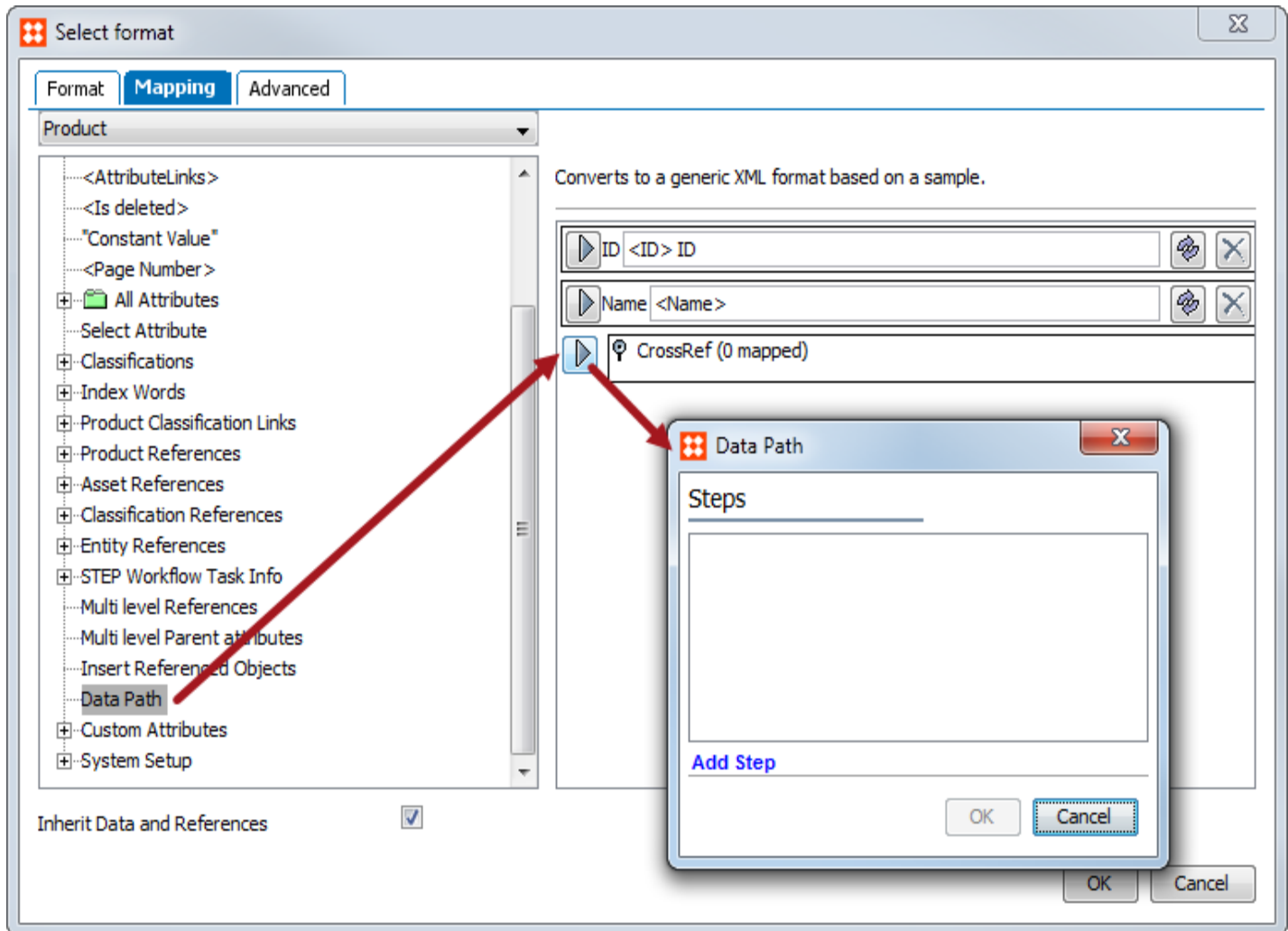
Mapping to Report Deleted Product References with a Generic XML OIEP

The **Reporting Deleted Product References with Generic XML OIEP** section discusses additional setup for a Generic XML export. Use the following steps to map the Product and monitor for deletion of product references.

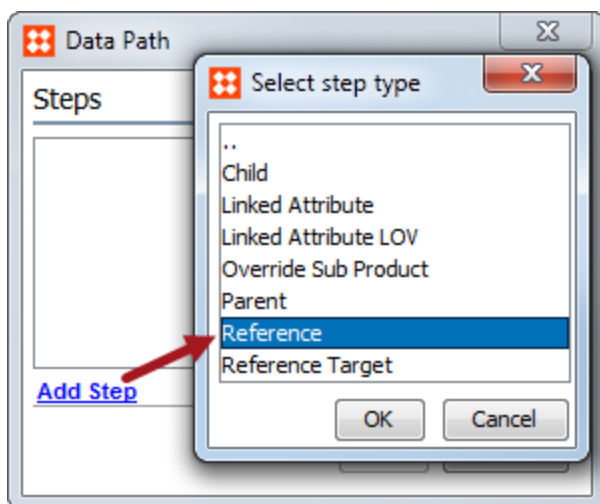
1. Map **ID** and **Name** for the Product.



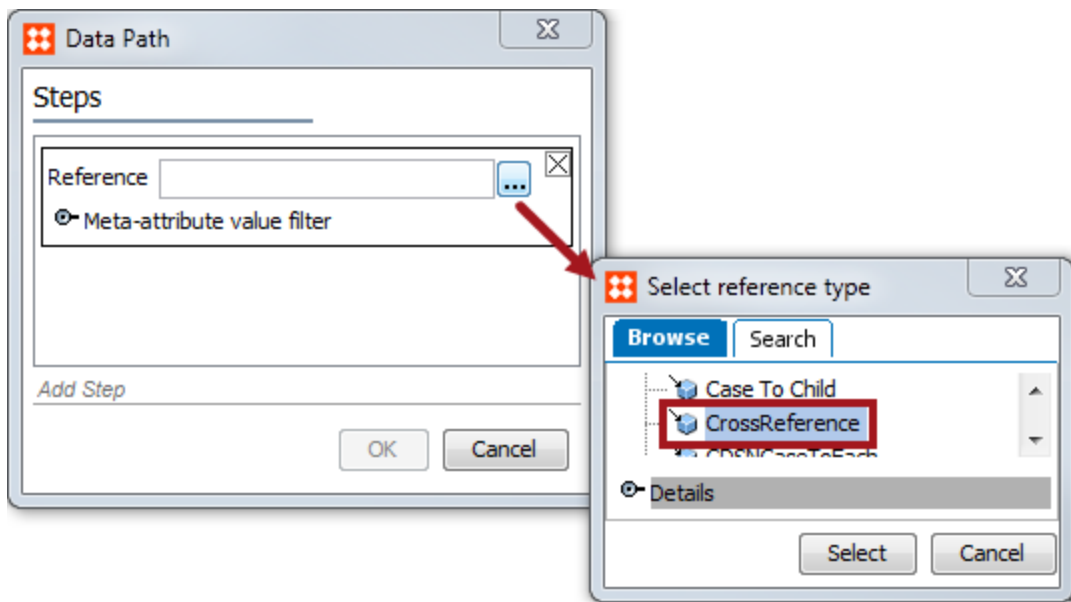
2. Map **Data Path** to CrossRef.



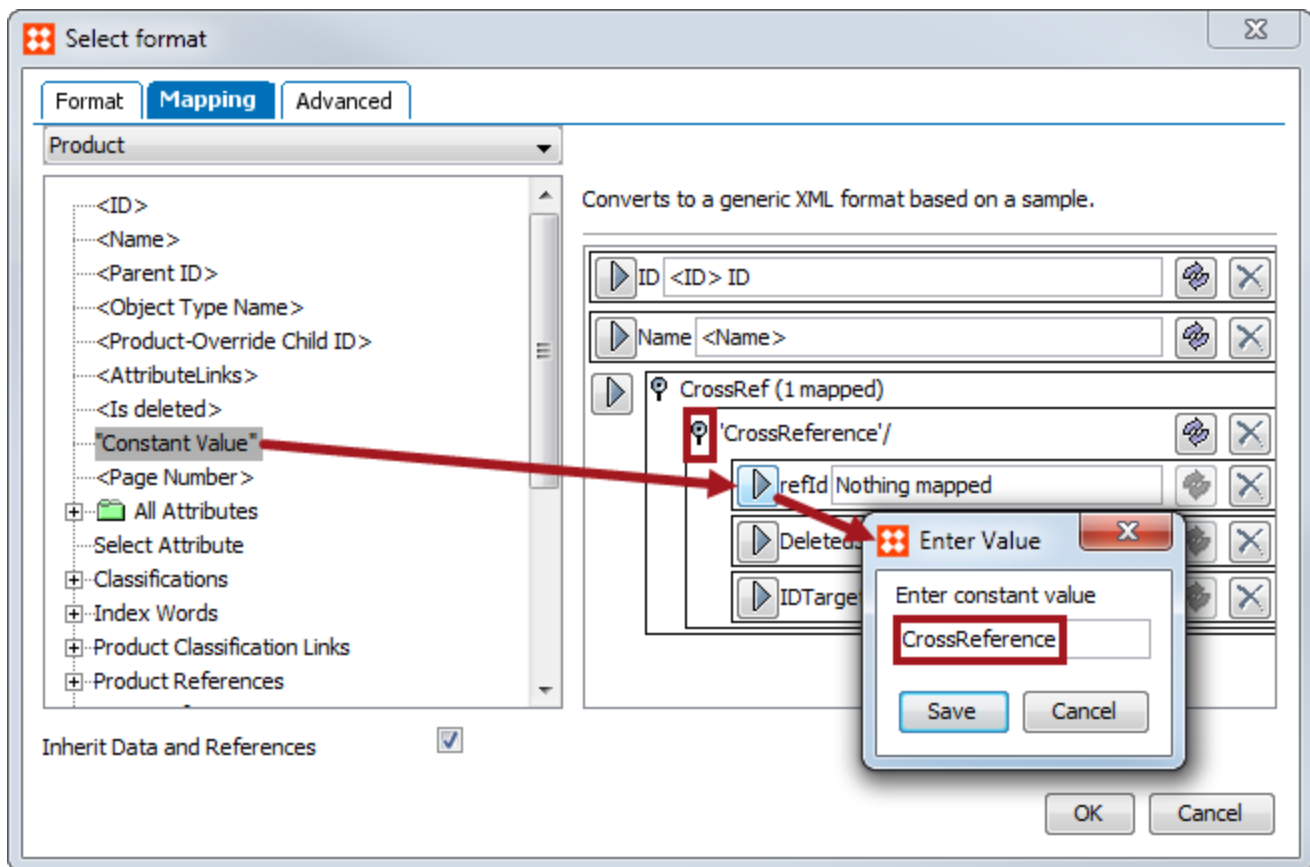
3. Click the **Add Step** link and select **Reference**.



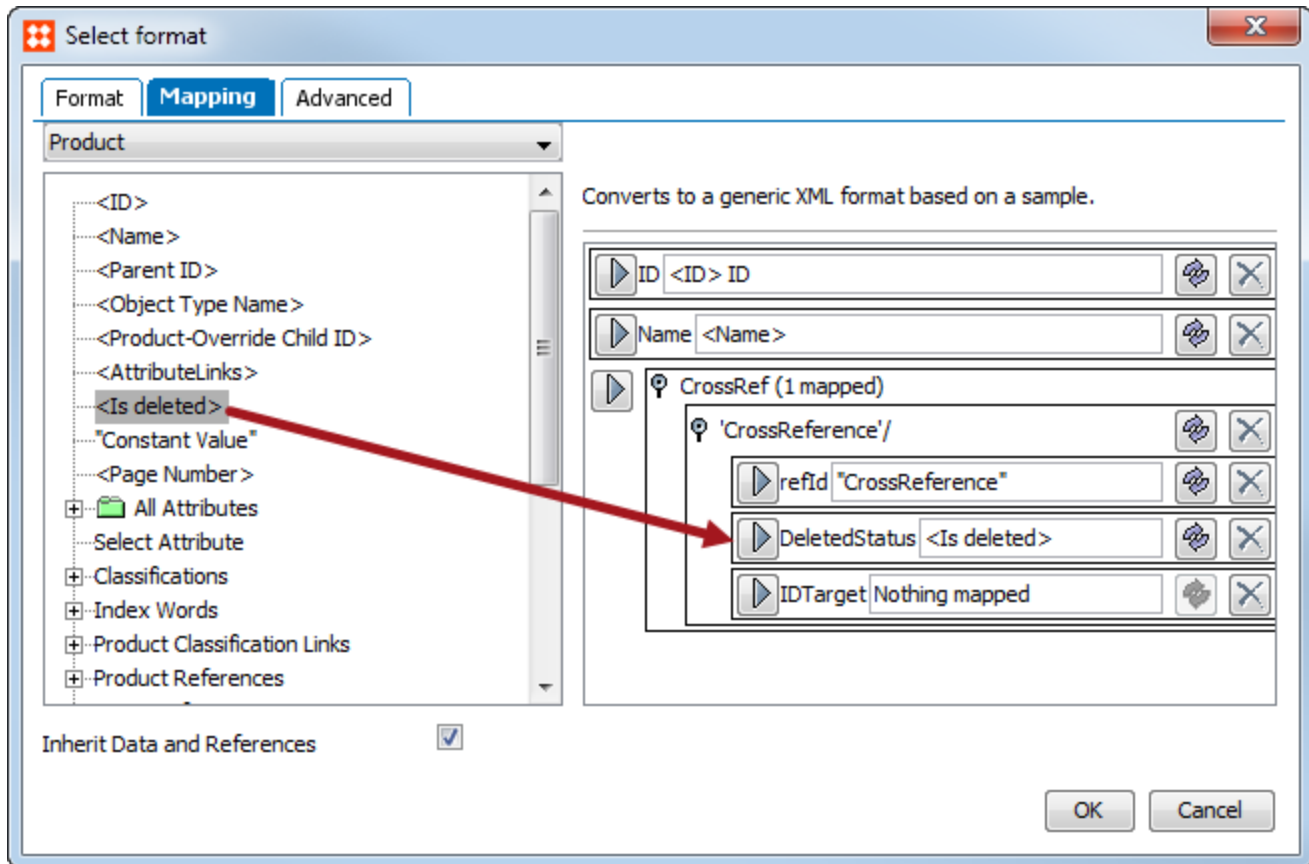
4. Click **OK** to close the **Select step type** dialog.
5. On the Data Path dialog, click the ellipsis button (...) and select the **Reference Type** to monitor, then click the **Select** button.



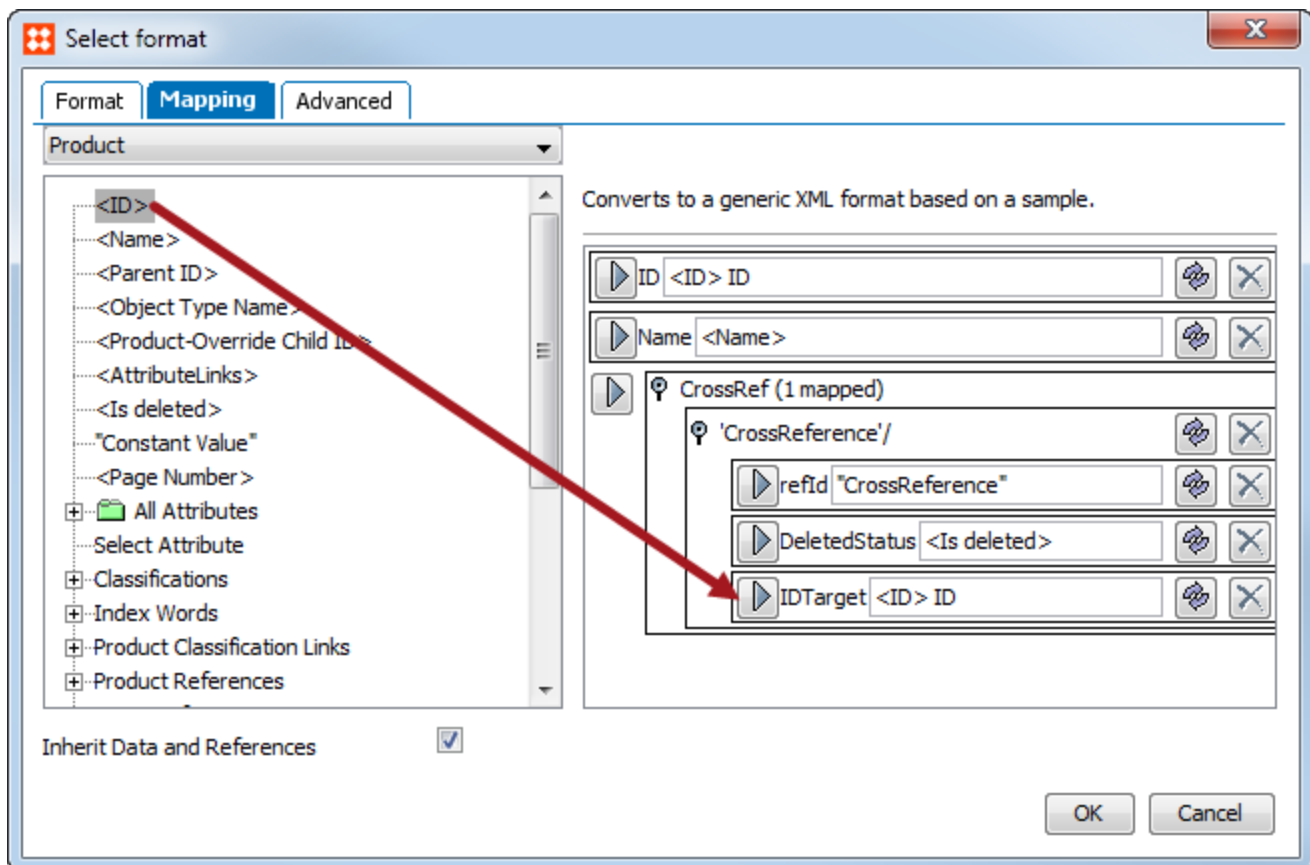
6. Click **OK** to close the **Data Path** dialog.
7. Open the 'CrossReference' flipper, map **Constant Value** to refID, add the text, and click the **Save** button.



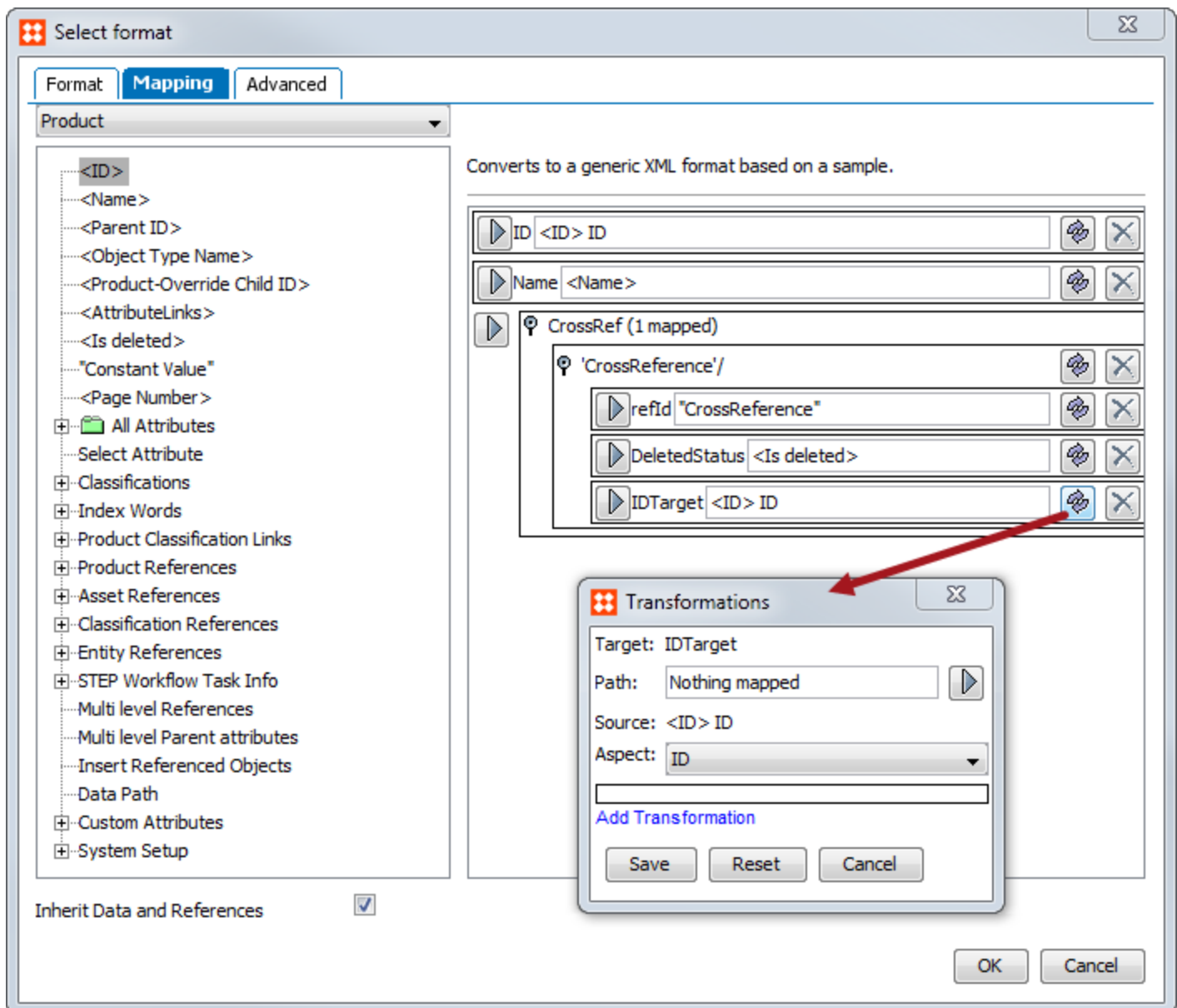
8. Map **<Is deleted>** to DeletedStatus.



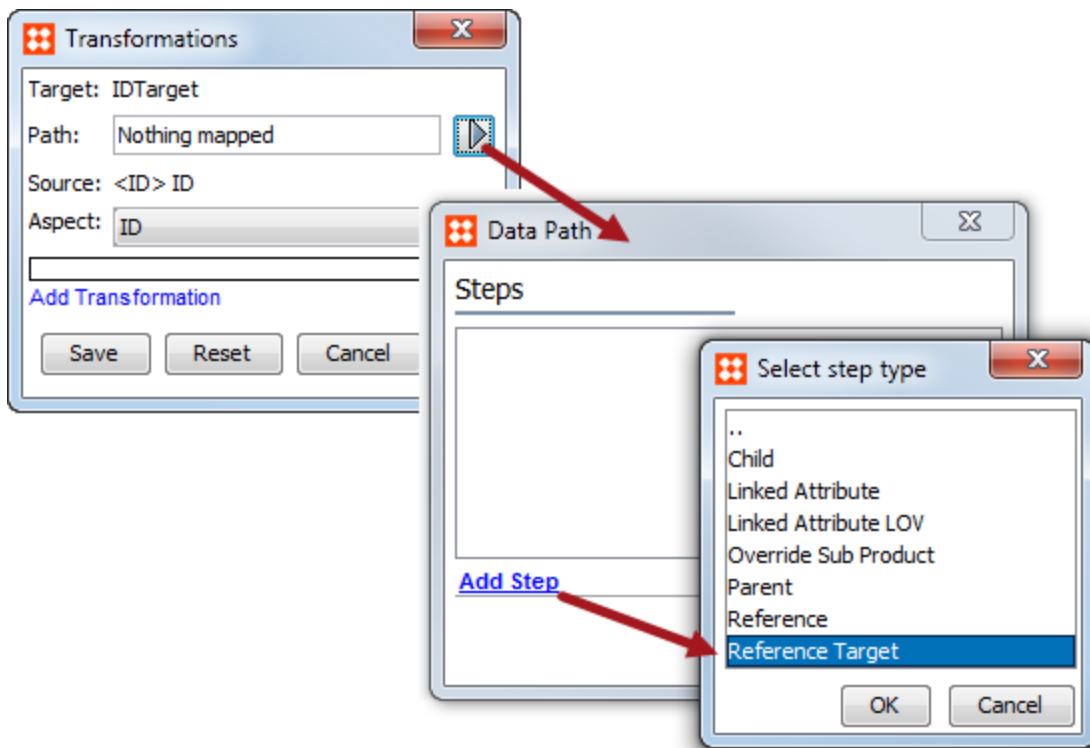
9. Map **ID** to IDTarget.



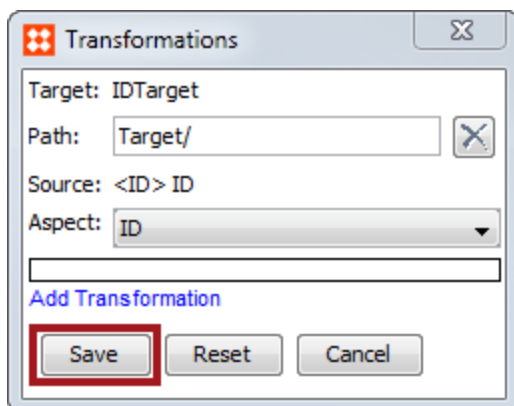
10. Click the **Transformations** button.



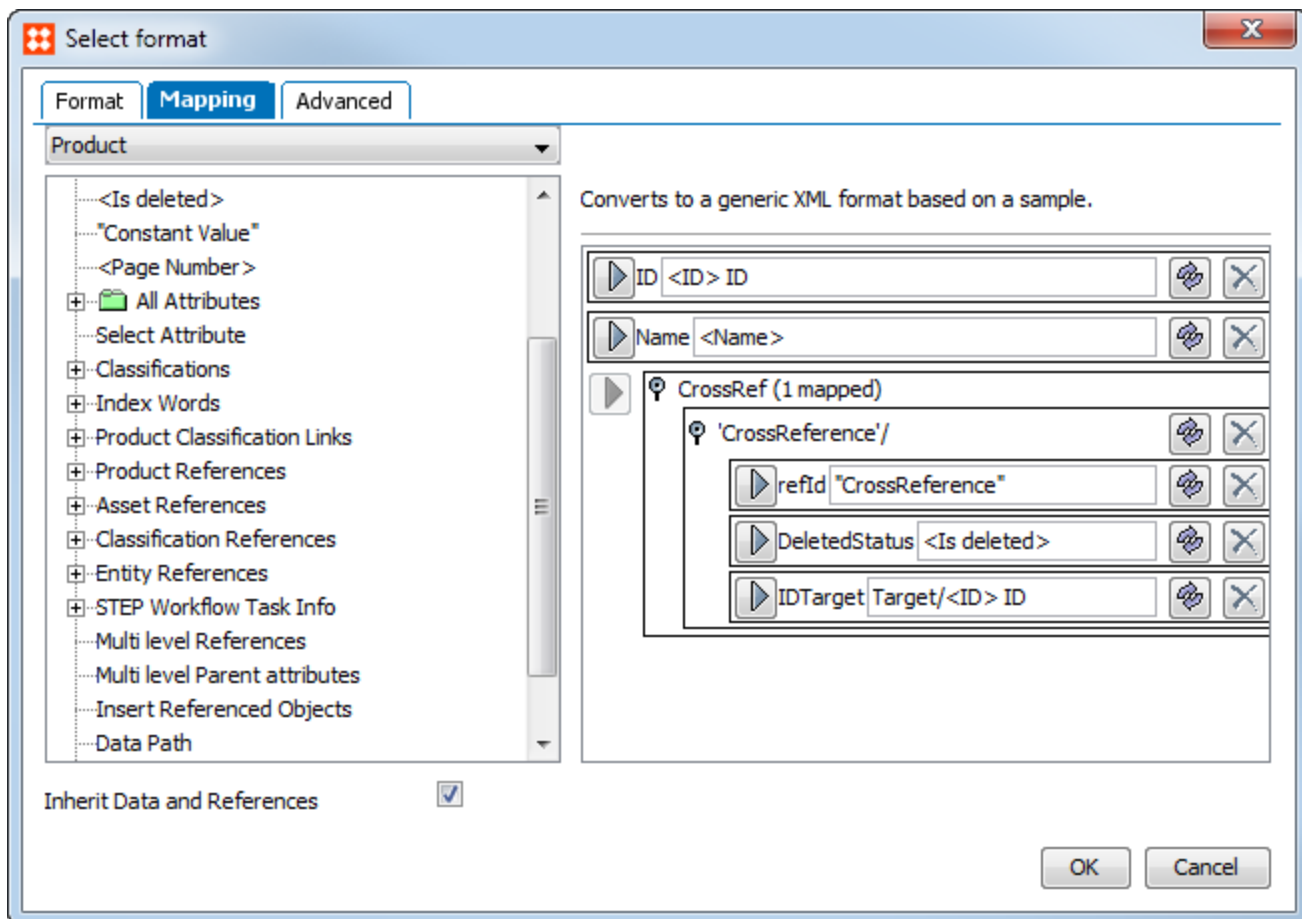
11. Click the **Path** button, click the **Add Step** link, and select **Reference Target**.



12. Click **OK** to close the **Select step type** and **Data Path** dialogs.
13. Click **Save** to close the **Transformations** dialog.



All mapping is complete.



Single-Cardinality Tags in Generic XML

A Generic XML template can include a variety of tags to affect the generated XML document. The following tags are single-cardinality, which means that the values exported by the associated data source is considered a 'single piece of information' and it will be exported inside a single occurrence of the enclosing XML element.

The following single cardinality mapping target tags are discussed below: Record, Target Default, and Target Override Default.

Objects

Export Manager

Steps

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

Select Objects

ID	Name	Object Type	Path
> 22155	Comfy Bed	Item	Primary Product Hiera
> 22168	Comfy Footboard	Item	Primary Product Hiera
> 22167	Comfy Headboard	Item	Primary Product Hiera
> 22165	Comfy Side Rail	Item	Primary Product Hiera
> Add Objects			

Only export selected products ☐

Only export leaf objects ☒

Export: Product

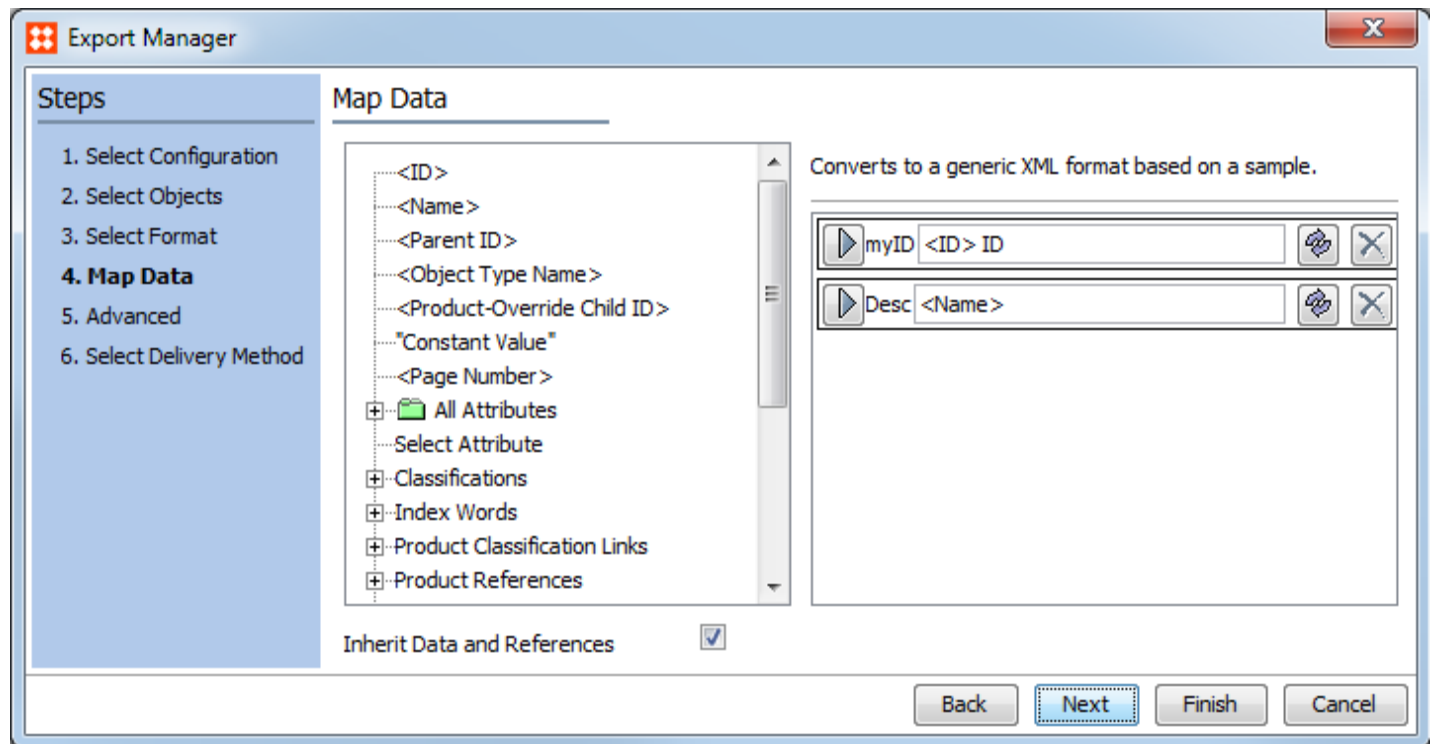
Include object types <All object types>

Back **Next** Finish Cancel

Template

```
<Products>
  <Product>
    <?Record?>
    <ID><?Target myID?></ID>
    <Desc><?Target?></Desc>
  </Product>
</Products>
```

Mapping



Results

```
<Products>
  <Product>
    <ID>22155</ID>
    <Desc>Comfy Bed</Desc>
  </Product>
  <Product>
    <ID>22165</ID>
    <Desc>Comfy Side Rail</Desc>
  </Product>
  <Product>
    <ID>22167</ID>
    <Desc>Comfy Headboard</Desc>
  </Product>
  <Product>
    <ID>22168</ID>
    <Desc>Comfy Footboard</Desc>
  </Product>
</Products>
```

Record

The `<?Record?>` tag is a processing instruction that tells Export Manager which part of the template to repeat for each selected object. There must be exactly one `<?Record?>` in the template.

In the example, the Product tag is the Record node of the template. Therefore everything between the <Product> and </Product> tags will be repeated for each object exported from STEP and will be included in the Generic XML output document.

Target Default

The <?Target?> tag defines a mapping target to which a data source can be bound. Each mapping target must have a unique identifier so it can be correctly used in the mapping step. By default, the name of the surrounding tag is used. Our example template includes two mapping targets: myID and Desc.

Target Override Default

When the same default identifier is used by more than one mapping target, you must override the default. For example, if you are exporting the ID of a product and the ID of an asset, <ID> would default for both objects. To create a unique identifier, specify a custom identifier within the <?Target [Optional Identifier]?> tag as indicated by 'myID' in the example above. Use any unique text in place of 'myID' to accurately label the required data.

Note: To avoid duplicate ID labels, use custom IDs for attribute-based mapping targets.

Rules

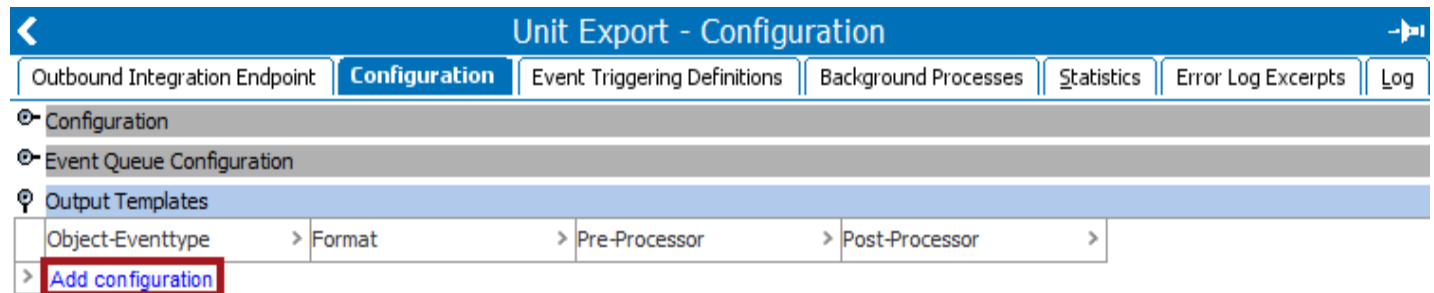
- There must be exactly one <?Record?> in the template.
- Mapping targets may only be used inside <?Record?> scope.
- Each mapping target must have a unique label.

Units with a Generic XML OIEP

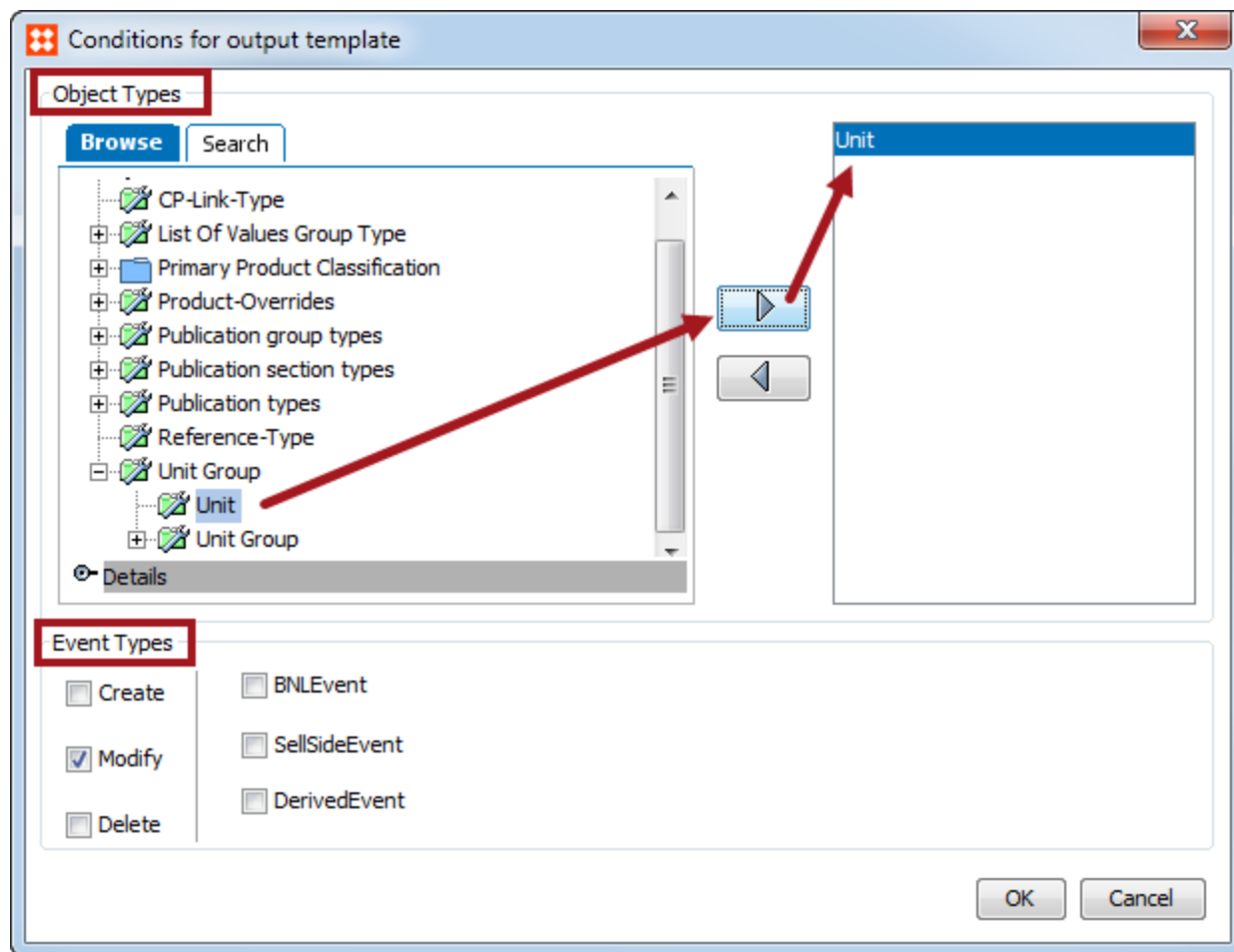
Outbound Integration Endpoints (OIEPs) allow the export of unit information.

Object Type and Event Type Selection

On the OIEP, open the Configuration tab. Under the Output Templates flipper, click the **Add configuration** link.



Set the Object Types and Event Types.



Template

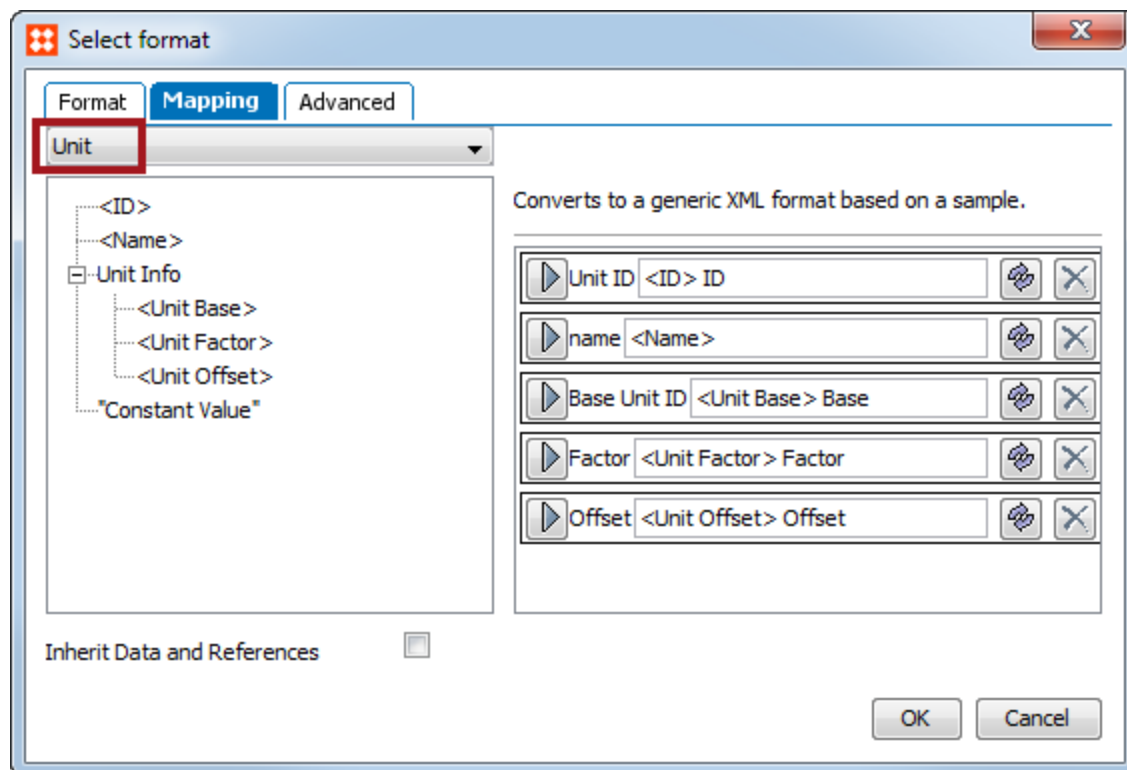
In the Output Template, click the ellipsis button (...), and set the format to **Generic XML**.

Provide the following text in the Sample field.

```
<ExportUnits>
  <Units>
    <Unit ID="[?Target Unit ID?]">
      <?Record?>
        <Name>
          <?Target name?>
        </Name>
        <UnitConversion BaseUnitID="[?Target Base Unit ID?]" Factor="[?Target
Factor?]" Offset="[?Target Offset?]" />
      </Unit>
    </Units>
  </ExportUnits>
```

Mapping

Select **Unit** from the dropdown and then map the data sources to the targets provided by the template.



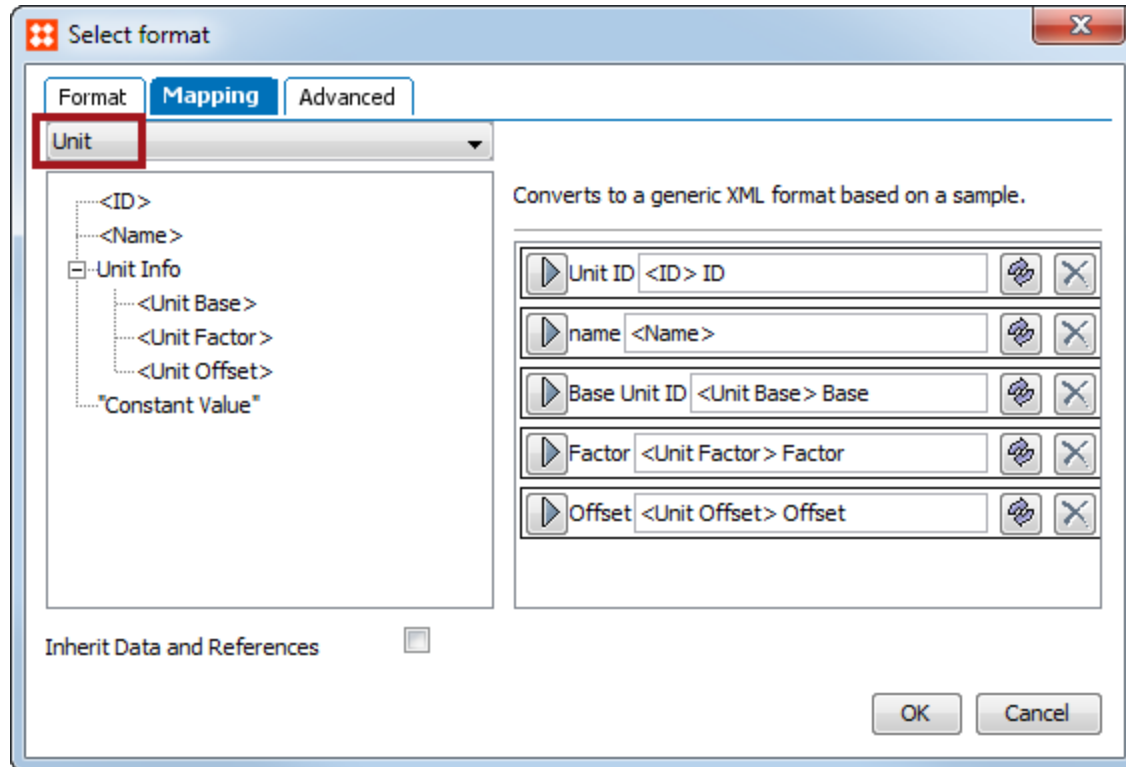
See the **Mapping Units Aspects with Generic XML OIEP** section for more information.

Results

```
<ExportUnits>
  <Units>
    <Unit ID="unece.unit.INK">
      <Name>in²</Name>
      <UnitConversion BaseUnitID="unece.unit.MTK" Factor="6.4515E-4" Offset="0.0"/>
    </Unit>
    <Unit ID="unece.unit.ACR">
      <Name>acre</Name>
      <UnitConversion BaseUnitID="unece.unit.MIK" Factor="3.0" Offset="1.0"/>
    </Unit>
    <Unit ID="unece.unit.FTK">
      <Name>ft²</Name>
      <UnitConversion BaseUnitID="unece.unit.MTK" Factor="0.09290305" Offset="0.0"/>
    </Unit>
    <Unit ID="unece.unit.CMK">
      <Name>cm²</Name>
      <UnitConversion BaseUnitID="unece.unit.MTK" Factor="2.0E-4" Offset="0.0"/>
    </Unit>
  </Units>
</ExportUnits>
```


Mapping Unit Aspects with a Generic XML OIEP

Aspects, also called data sources, are displayed when mapping Units for export. The description of each Aspect is included below.



Aspect	Description
ID	Extracts the ID of the unit. For example, if a unit has the ID 1234 and the name 'Pound,' then '1234' is extracted.
Name	Extracts the name of the unit. For example, if a unit has the ID 1234 and the name 'Pound,' then the word 'Pound' is extracted.
Unit Base	Extracts the ID of the base unit for the unit. For example, if a unit has base unit 'm,' then the 'm' is extracted.
Unit Factor	Extracts the conversion factor that should be multiplied to a value with the unit to get the value in the base unit.

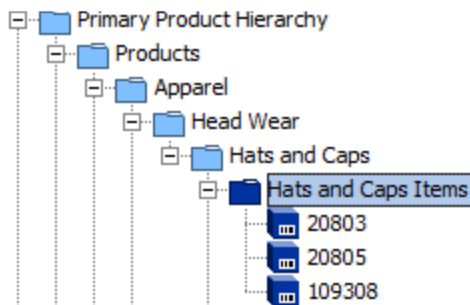
Aspect	Description
	For example, if a unit has a base unit conversion of 'value(g) = .001 * value(kg),' then '.001' is extracted.
Unit Offset	Extracts the conversion factor that should be added to a value with the unit to get the value in the base unit. For example, if a unit has a base unit conversion of 'Value(K) = .556 * Value(°F) + 255.3722,' then '255.3722' is extracted.

Generic XML Sample Export

Export Manager requires that you select objects and provide both a sample template and mapping when exporting Generic XML.

Objects

Select the Objects to export.



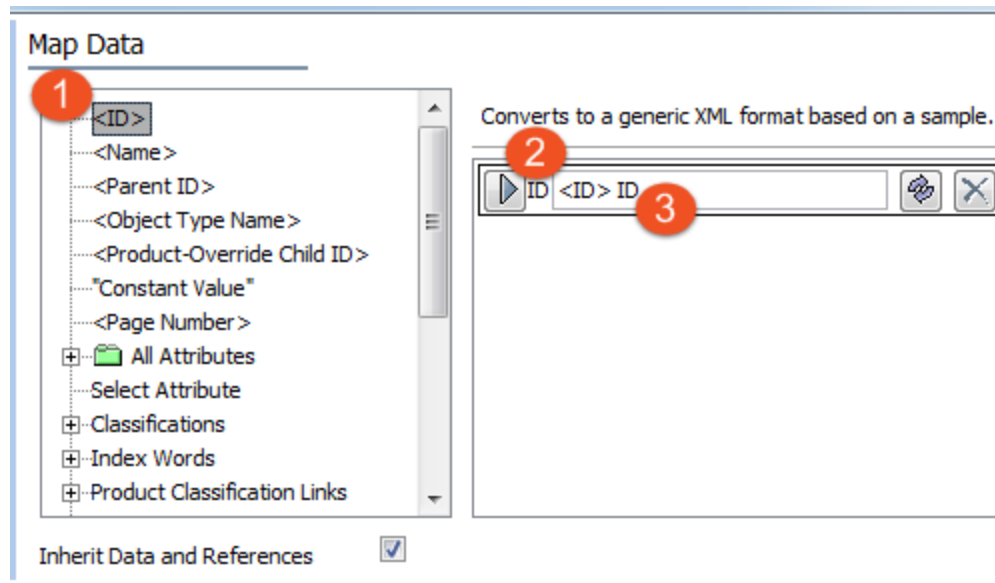
Template

Choose **Generic XML** for **Format** and provide the following template text in the Sample field.

```

<Products>
  <Product>
    <?Record?>
    <ID><?Target?></ID>
  </Product>
</Products>
  
```

Mapping



1. Available data source options are listed based on the object(s) selected.
2. Mapping targets are listed based on the template you provided.
3. Select a data source and click the mapping target arrow button to complete the mapping rule.

Results

After completing additional Export Manager fields, the previous settings will generate the following XML output document:

```
<Products>
  <Product>
    <ID>20803</ID>
  </Product>
  <Product>
    <ID>20805</ID>
  </Product>
  <Product>
    <ID>109308</ID>
  </Product>
</Products>
```

- the <Product>...</Product> tag section is repeated for each product exported from STEP
- the content of the <ID>...</ID> tag is based on the mapping rule

Global Product Classification Format

The standards for the Global Product Classification (GPC) are maintained by GS1, and information relating to the GPC can be found on the web.

GPC is a five (5) level classification which is modeled as a classification hierarchy in STEP, using the objects defined in the table below.

Tree

GPC

54000000-Baby Care

54100000-Baby Feeding/Hygiene

54101500-Baby Feeding

10000490-Baby Feeding - Tableware

10000491-Baby Feeding - Bottles

10000492-Baby Feeding Accessories

10000493-Baby Feeding Aids (Non Powered)

54101500-Baby Feeding re

Classification

Sub Products

References

Referenced By

Description

Name	Value
ID	GPCBabyCare54101500
Name	54101500-Baby Feeding
Object Type	Class

Format Availability

GPC is available only via the File menu > Import > click **Import Global Product Classification** to open the Global Product Classification Import wizard. See additional information on prerequisites below.

Mapping

No data mapping is required for GPC.

Limitations

Before starting the Global Product Classification importer, please note the following known issues:

- Segment Object type ID is incorrect.
- The import creates multiple duplicate attributes
- The import creates unique LOVs, one for each attribute. Instead of applying
- GPC LOVs cannot be moved (re-parented)
- Once the first GPC segment has been imported, the advanced import configurations are locked and cannot be edited.

Setup Requirements

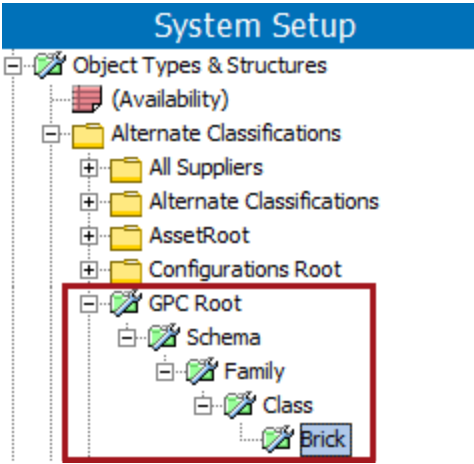
Setting up and using GPC Classifications involves the following steps:

- Obtain the GPC download files and extract zipped data. Each segment of the current production release GPC can be downloaded from the GS1 website.
- Run the Global Product Classification Import wizard, as defined in **Global Product Classification Import**.

3. Create a product to classification link type, as defined in **Create a GPC Product to Classification Link Type**.

Inbound Data

Only the GPC Root is a standard STEP default object type. The others are created during the GPC import. Following a successful import, the object types are displayed in System Setup as shown below.



Importing a GPC Segment creates the following required elements:

- Alternate Classification Object Types (Segment, Family, Class, and Brick)
- Alternate Classification hierarchy objects for the imported segment
- Attributes for the imported segment. Attributes will be created under the STEP standard attribute group 'GPC'.
- LOVs for the imported segment attributes. LOVs are created directly under the root node 'List of Values / LOVs'.

GPC Object Types

Use the following table to match GPC elements with STEP object types:

Level	GPC Name	STEP ID	STEP Name	STEP Parent Name
1	GPC (GS1 - Global Product Classification)	GPC Root	GPC Root	Alternate Classifications
2	Segment	GPC Schema	Schema	GPC Root
3	Family	GPC Family	Family	Schema

Level	GPC Name	STEP ID	STEP Name	STEP Parent Name
4	Class	GPC Classification	Class	Family
5	Brick	GPC Brick	Brick	Class

Global Product Classification Import


Global Product Classification Import

Steps

1. Select GPC File
2. Advanced

Select GPC File

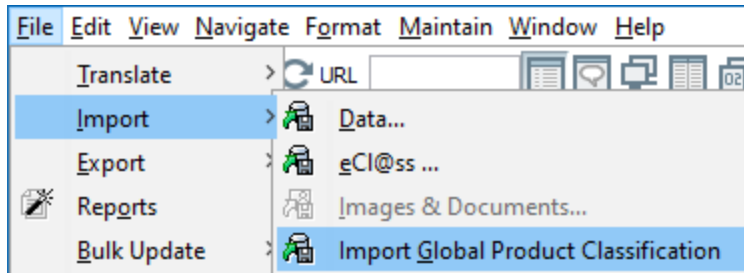
Select GPC File

:s\GPC XML\Baby Care - 54\GS1 Baby Care as at 01062014.xml
...

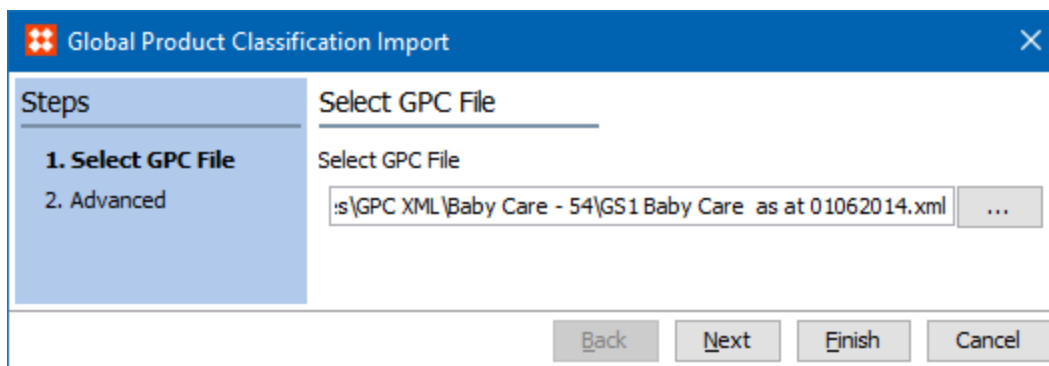
Back
Next
Finish
Cancel

Global Product Classification Import

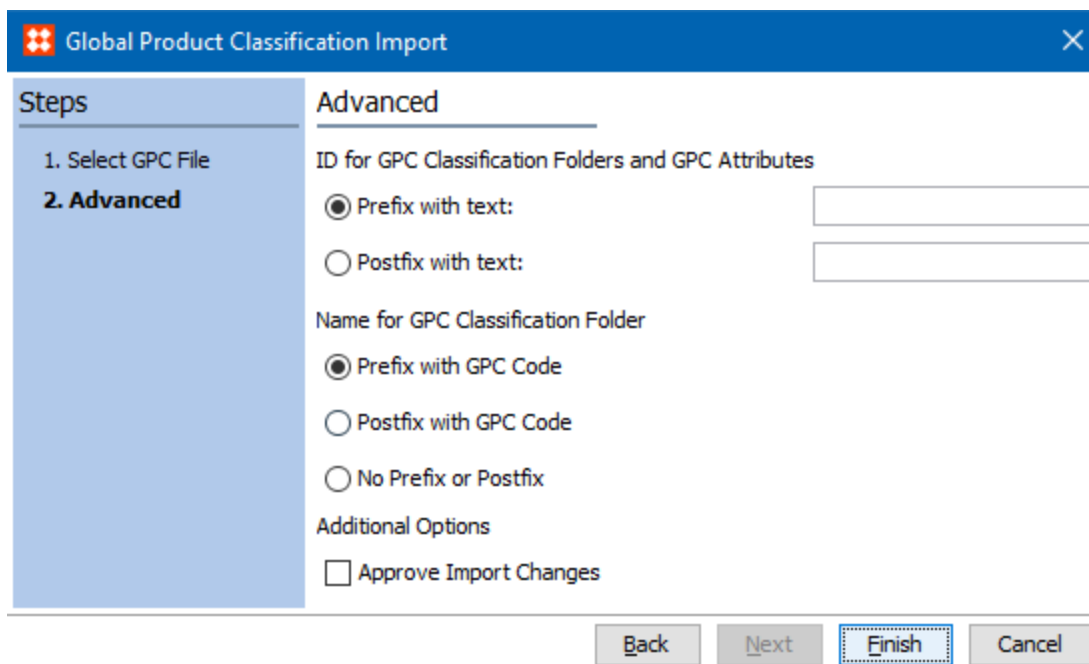
1. Download the desired GPC segments from the web and extracting the data.
2. Open the Global Product Classification Import wizard from File > Import.



3. For **Select GPC File**, click the ellipsis button (...) to display the selection dialog and search or browse for the downloaded XML file and select it.



4. For **Advanced**, select the desired options as defined below:



- **ID for GPC Classification Folders and GPC Attributes:**

- **Prefix with text** allows you to add data (in the text field) that should precede the ID of the folders and attributes.
- **Postfix with text** allows you to add data (in the text field) that should follow the ID of the folders and attributes.

- **Name for GPC Classification Folder:**

- **Prefix with GPC Code** causes the GPC code to precede the GPC Classification folder name.
- **Postfix with GPC Code** causes the GPC code to follow the GPC Classification folder name.
- **No Prefix or Postfix** indicates the GPC Classification folder name needs no additional text.

- **Additional Options:**

- **Approve Import Changes** sets the classification data to approved as it is imported, without need for review.

5. Click **Finish**.
6. Start the import. If required, refer to the information in **Running a Data Import** section. When the Import Manager Pipeline background process is complete, the newly created classification folders are displayed in the Tree.
7. If needed, monitor the import process using the **Monitoring a Data Import** documentation.
8. Complete the GPC setup following the steps defined in **Create a GPC Product to Classification Link Type**.

Create a GPC Product to Classification Link Type

After completing the GPC Import, create a product-to-classification link type to be used from the Packaging Object Types to the 'GPC Brick' object types.

For more information on links, see the **Creating a Product to Classification Link Type** section of the **System Setup / Super User Guide** documentation.

Creating Links

1. On System Setup > Reference Types > Product to Classification Link Types > right-click and select **New Product to Classification Link Type**.
2. For **Enter ID and Name**, type 'GPC' for the **ID** (it is also added as the **Name**) and click **Next**.

Create Product to Classification Link Type

Steps

1. Enter ID and Name
2. Select valid product types
3. Select valid classification types
4. Apply Dimension Dependencies
5. Advanced

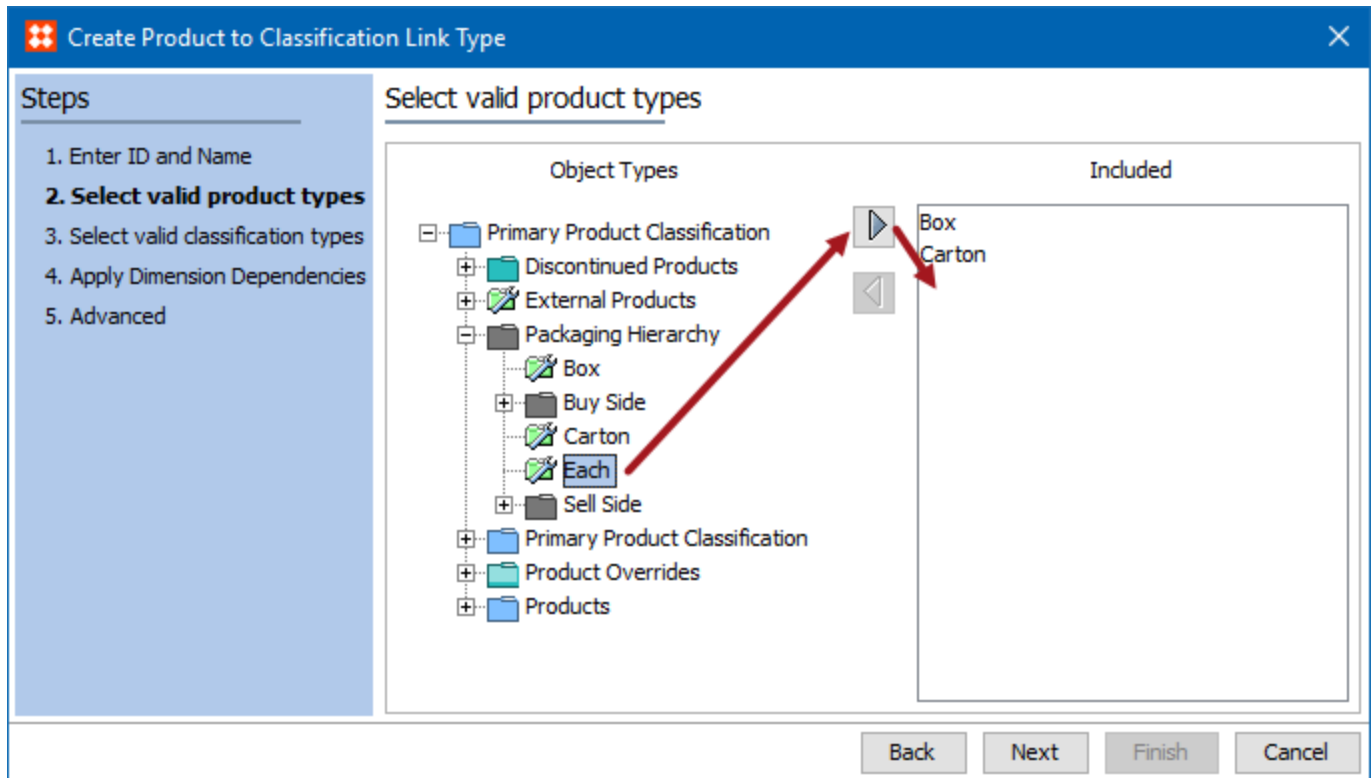
Enter ID and Name

ID: GPC

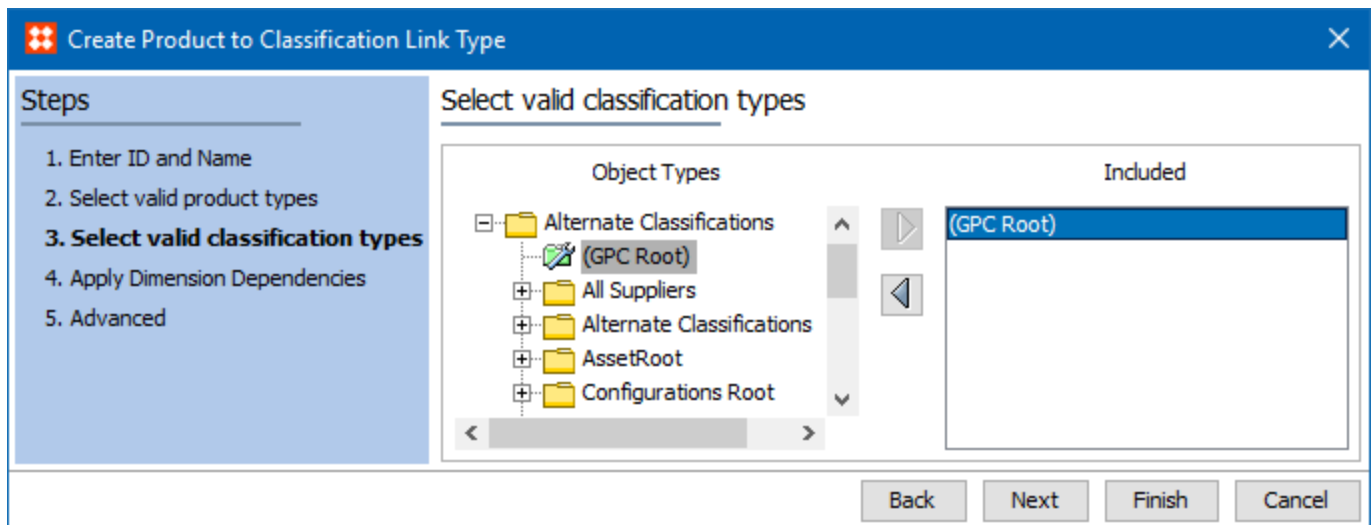
Name: GPC

Back Next Finish Cancel

3. For **Select valid project types**, highlight an object type and click the arrow button to move it to the 'Included' list. Repeat for each valid product type.



4. For **Select valid classification types**, highlight a classification object type and click the arrow button to move it to the 'Included' list.



5. For **Apply Dimension Dependencies**, select the appropriate dependencies from those available in your system if needed.

Create Product to Classification Link Type

Steps

1. Enter ID and Name
2. Select valid product types
3. Select valid classification types
- 4. Apply Dimension Dependencies**
5. Advanced

Apply Dimension Dependencies

☒ Country

☐ Language

Back Next Finish Cancel

6. For **Advanced**, select the appropriate options.

- **Allow multiple links** When checked, this option enables more than one link of this type to be added.
- **Externally Maintained** When checked, this option indicates that this data will be updated in an external system, outside of STEP.
- **Mandatory** When checked, this option indicates that the object cannot be approved without this link.
- **Sub Products Inheritance Settings** In order for GPC attribute to display, you must select 'Inherit Links and Tables and Specification Attributes'.

Create Product to Classification Link Type

Steps

1. Enter ID and Name
2. Select valid product types
3. Select valid classification types
4. Apply Dimension Dependencies
- 5. Advanced**

Advanced

☒ Allow multiple links

☐ Externally Maintained

☐ Mandatory

Sub Products Inheritance Settings

Inherit Links and Tables and Specification Attributes

Back Next Finish Cancel

7. Click **Finish** to create the link type and verify settings in the editor.

System Setup

- Reference Types
 - Product Reference Types
 - Image and Document Reference Types
 - Classification Reference Types
 - Product to Classification Link Types
 - Classifications
 - ecBss
 - URS
 - Merchandising Link
 - Object Creation Templates
 - Supplier Link
 - UNSPSC Class
 - Website Link
 - Product Attribute Link Type
 - Classification Attribute Link Type
 - Entity Reference Types
 - Context Reference Types
 - Workspace Reference Types

GPC - Product to Classification Link Type

Product to Classification Link Type	Validity	Log
Description		
Name	>	Value
ID	>	GPC
Last edited by	>	2018-11-15 09:22:10.541 by USER
Externally Maintained	>	No
Dimension Dependencies	>	No
Allow multiple links	>	No
Mandatory	>	No
Inheritance of Links	>	Inherited
Inheritance of Specification Attributes	>	Yes
Ignore LOV Filter definitions on inherited Attributes for links	>	No
Completeness Score	>	0

IDoc MATMAS 05 Format

The IDoc MATMAS 05 format is XML-based and is used when exchanging information with SAP. For more information, search the web.

Format Availability

IDoc MATMAS 05 is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**
- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

Inbound Data

After importing the required SAP attributes into STEP, then use a source sample file from the SAP system to import values for the products. Mapping is required to identify where the values will live in STEP, and may also include data transformations.

Inbound Parameters

The following parameters are available in both Import Manager and IIEP.

- **Sample** - displays an import template based on the file selected on the Data Source step of the import wizard.
- **Conversion Preview** - displays values from the sample file in columns.

Import Manager

Import Manager

×

Steps

1. Select Configuration

2. Select Data Source

3. **Select Format**

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Select Format

Format

IDoc MATMAS 05

▼

Converter for an IDoc MATMAS XML format described by a template

Sample

<MATMAS05>
 <IDOC>
 <E1MARAM SEGMENT = "[?Source E1MARAM-Segment?]">
 <?Record?>
 <MTART><?Source?></MTART>
 <MBRSH><?Source?></MBRSH>
 <MATKL><?Source?></MATKL>
 <BISMT><?Source?></BISMT>
 <MEINS><?Source?></MEINS>
 <BRGEW><?Source?></BRGEW>

Conversion Preview:

E1MARAM...	MTART	MBRSH	MATKL	BISMT	MEINS	BRGEW
3240070	ZHLB	M	100	9912001	PCE	0,01
3240173	ZHLB	M	100	9912002	PCE	0,01
3240175	ZHLB	M	100	9912003	PCE	0,01

◀

▶

Back

Next

Finish

Cancel

IIEP

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
- 5. Configure Processing Engine**
 - 5.1. Select Sample File
 - 5.2. Select Data Source
 - 5.3. Select Format**
 - 5.4. Map Data
 - 5.5. Identify Objects
 - 5.6. Identify Destination
 - 5.7. Select Business Rules
 - 5.8. Advanced Settings
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Processing Engine : Select Format

Format

IDoc MATMAS 05

Converter for an IDoc MATMAS XML format described by a template

Sample

<MATMAS05>
<IDOC>
<E1MARAM SEGMENT="[?Source E1MARAM-Segment?]">
<?Record?>
<MTART><?Source?></MTART>
<MBRSH><?Source?></MBRSH>
<MATKL><?Source?></MATKL>
<BISMT><?Source?></BISMT>
<MEINS><?Source?></MEINS>
<BRGEW><?Source?></BRGEW>

Conversion Preview:

E1MARAM...	MTART	MBRSH	MATKL	BISMT	MEINS	BRGEW
3240070	ZHLB	M	100	9912001	PCE	0,01
3240173	ZHLB	M	100	9912002	PCE	0,01
3240175	ZHLB	M	100	9912003	PCE	0,01
3240195	ZHLB	M	100	9912004	PCE	0,01
3240196	ZHLB	M	100	9912005	PCE	0,01

Back

Next

Finish

Cancel

Outbound Data

Mapping is required to identify where the values live in STEP, and may also include data transformations.

Outbound Parameters

The following parameters are available in both Export Manager and OIEP.

- **Sample** - displays the standard template for export. This template may be modified to exclude attributes not required by your system. Since no undo functionality is available in the Sample field, it is good practice to use an external XML editor program for creating and editing a template. To test a template's validity for output, paste the XML into STEP.
- **DocType** - not valid for IDoc MATMAS 05 format

Export Manager

Export Manager

Steps

1. Select Configuration

2. Select Objects

3. Select Format

4. Map Data

5. Advanced

6. Select Delivery Method

Select Format

IDoc MATMAS 05

Converts to an IDoc MATMAS XML format based on a sample.

Sample

```

<MATMAS05>
  <IDOC BEGIN = "1" >
    <E1MARAM SEGMENT="[?Target E1MARAM-SEGMENT?]">
      <?Product?>
      <MTART><?Target?></MTART>
      <MBRSH><?Target?></MBRSH>
      <MATKL><?Target?></MATKL>
      <BISMT><?Target?></BISMT>
      <MEINS><?Target?></MEINS>
      <BRGEW><?Target?></BRGEW>
      <NTGEW><?Target?></NTGEW>
      <GEWEI><?Target?></GEWEI>
      <VOLEH><?Target?></VOLEH>

```

DocType

Back

Next

Finish

Cancel

OIEP

Background Processes

Statistics

Error Log Excerpts

Log

Status

Outbound Integration Endpoint

Configuration

Event Triggering Definitions

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor	Post-Processor
Item (Modify)	IDoc MATMAS 05 (0 m...)	None	None
Add configuration			

Delivery Method

Select format

Format

Mapping

Advanced

IDoc MATMAS 05

Converts to an IDoc MATMAS XML format based on a sample.

Sample

```
<MATMAS05>
  <IDOC BEGIN = "1" >
    <E1MARAM SEGMENT = "[?Target E1MARAM-SEGMENT?]">
      <?Product?>
      <MTART><?Target?></MTART>
      <MBRSH><?Target?></MBRSH>
      <MATKL><?Target?></MATKL>
      <BISMT><?Target?></BISMT>
      <MEINS><?Target?></MEINS>
      <BRGEW><?Target?></BRGEW>
    </E1MARAM SEGMENT>
  </IDOC BEGIN>
</MATMAS05>
```

DocType

OK

Cancel

Publication Excel Format

Publications can be exported and imported in an Excel format called Publication Excel. The Publication Excel format supports all publication types: standard STEP'n'design ('drag and drop'), Flatplanner, and AutoPage.

Format Availability

Publication Excel is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**
- Export Manager - see **Creating a Data Export**

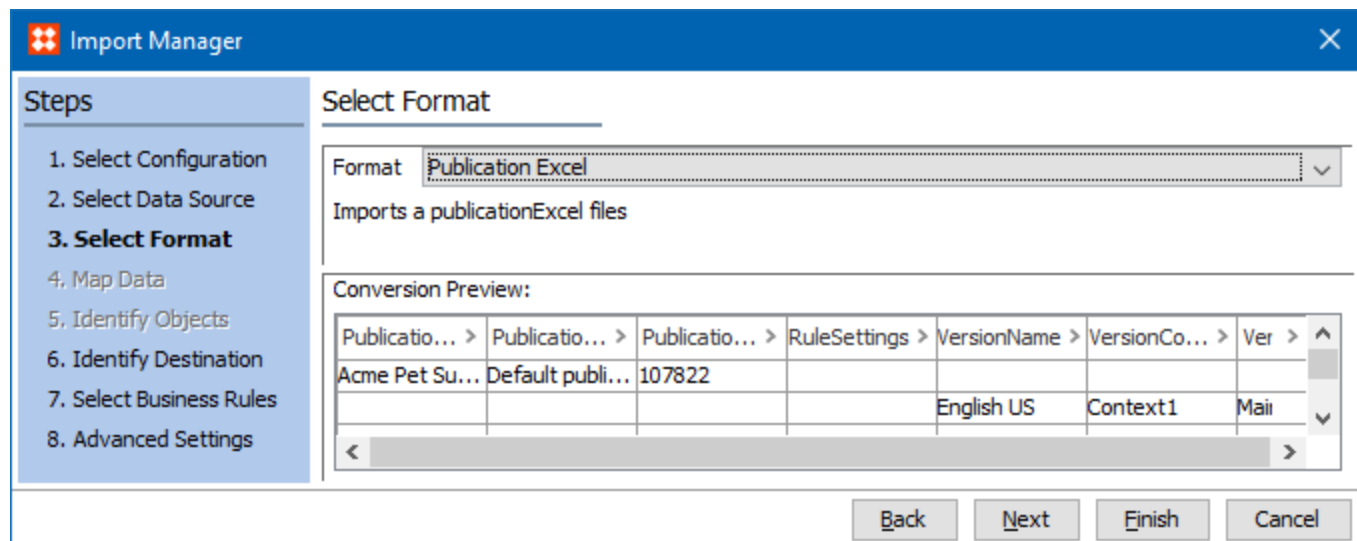
Mapping

Mapping data is not required and the Map Data step is disabled.

Inbound Data

For more information on the Publication Excel import process, see **Importing Publications in Excel**.

Import Manager

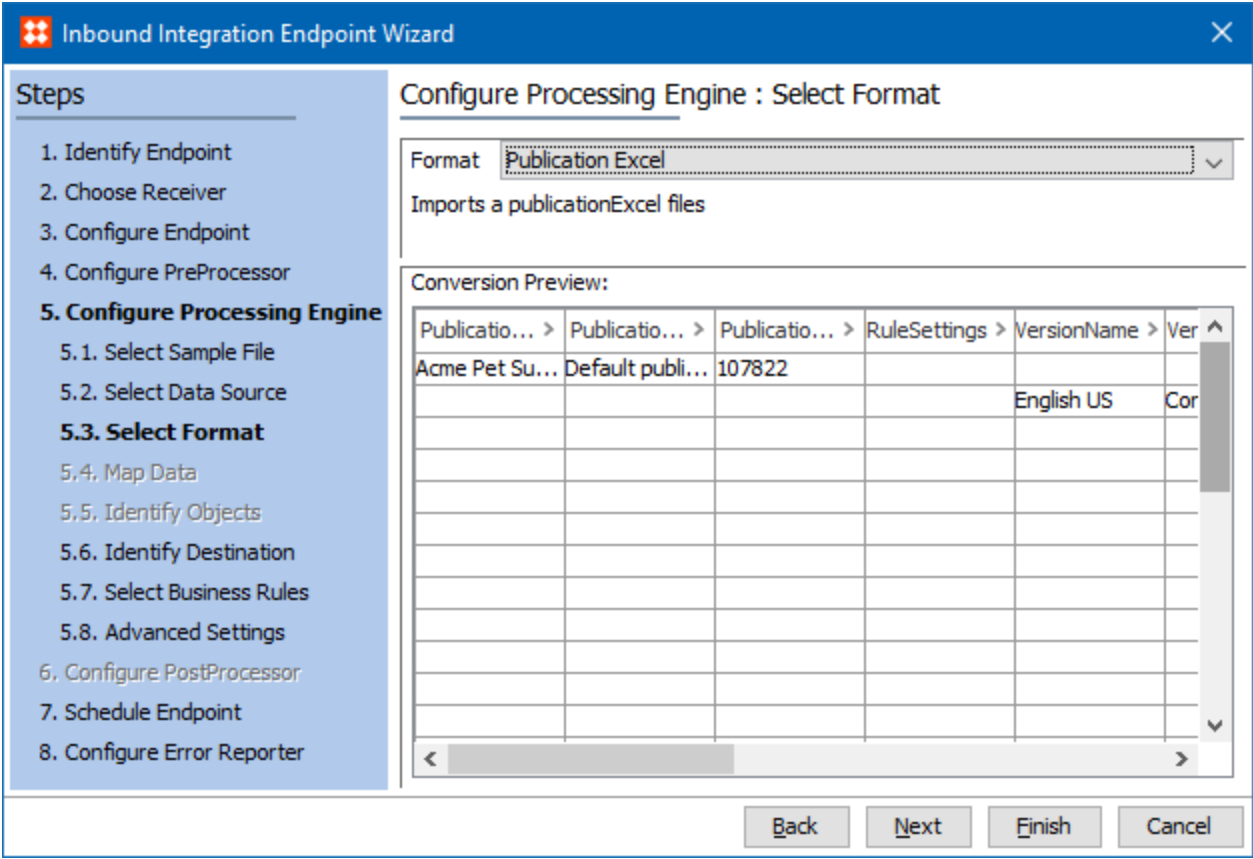


The screenshot shows the 'Import Manager' dialog box with the 'Select Format' step selected. The 'Format' dropdown is set to 'Publication Excel', and the description below it says 'Imports a publicationExcel files'. The 'Conversion Preview' table shows the following data:

Publicatio...	Publicatio...	Publicatio...	RuleSettings	VersionName	VersionCo...	Ver
Acme Pet Su...	Default publi...	107822		English US	Context1	Mai

At the bottom of the dialog box, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

IIEP



Outbound Data

For more information on the Publication Excel export process, see **Exporting Publications in Excel**.

For step-by-step instructions on how to export a Publication Excel or Flatplan Excel spreadsheet from a Flatplanner publication in STEP, see **Exporting a Publication Excel Sheet** in the **STEP'n'design** documentation.

Export Manager

After selecting a publication on Tree, right-click and choose **Export Data Below** to see the Publication Excel format option in Export Manager.

Export Manager

Steps

1. Select Configuration

2. Select Objects

3. Select Format

4. Map Data

5. Advanced

6. Select Delivery Method

Select Objects

ID	Name	Object Type	Version	Path
205913	Acme Tools	Publication	English US	Publications/Standard Publications/Acme Tools
Add Objects				

Export Manager

Steps

1. Select Configuration

2. Select Objects

3. Select Format

4. Map Data

5. Advanced

6. Select Delivery Method

Select Format

Publication Excel

Exports Publication to excel format.

Include Unique Keys as IDsNo

Back

Next

Finish

Cancel

Export:Publication Objects

Next

Finish

Cancel

SmartLabel Format

The SmartLabel format is based on Generic XML and includes a pre-defined template for integrators to provide a head start in configuring a SmartLabel™ export solution for both food and non-food product data. For more information, search the web.

The default template includes required SmartLabel attributes, plus category labels and comments to identify a repeating value group (RVG) section and a flattened style section. The default template should be modified to include only the data you require for a SmartLabel. The mapping process provides assistance in identifying and linking existing STEP attributes to the SmartLabel fields. Ultimately, the SmartLabel output is an XML document of the selected sections and mapped attributes.

Format Availability

SmartLabel is available for selection in:

- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

Outbound Data

Mapping is required to identify where the values live in STEP, and may also include data transformations.

Outbound Parameters

The following parameters are available in both Export Manager and OIEP.

- **Template** - displays the standard template for export each time a new export is created. The default template includes both food and non-food attributes, and should be modified to exclude attributes not required in your export. The modified template is saved with the export configuration or OIEP output template. For details on modifying the template to suit your data, see the **Configuring the SmartLabel Template** documentation.

'RemovelfEmpty recursive' is applied to all sections to prevent unnecessary tags from being exported. For more information, see **RemovelfEmpty Instruction in Generic XML** in the **Generic XML Format** documentation.
- **Allow empty tags** - for details about this parameter, see the **Allow Empty Tags Parameter in Generic XML** documentation.

Export Manager

Export Manager

×

Steps

1. Select Configuration

2. Select Objects

3. Select Format

4. Map Data

5. Advanced

6. Select Delivery Method

Select Format

SmartLabel

Converts to a SmartLabel XML format based on a template.

Template

1 <!--

2 Element definitions:

3 RequiredAttributes - Defined by SmartLabel as 'Required' for both Food and Non-Food products

4 FoodRequiredAttributes - Defined by SmartLabel as 'Required' for only Food products

5 NonFoodRequiredAttributes - Defined by SmartLabel as 'Required' for only Non-Food products

6 VoluntaryAttributes - Defined by SmartLabel as 'Voluntary' for both Food and Non-Food products

7 FoodVoluntaryAttributes - Defined by SmartLabel as 'Voluntary' for Food products

8 NonFoodVoluntaryAttributes - Defined by SmartLabel as 'Voluntary' for Non-Food products

9 -->

10

11 <SmartLabel version="1.5" templateVersion="1.1">

12

13 <Product id="[?Target Product ID?]">

14 <?Record?>

15

16 <Header>

17

18 <RequiredAttributes>

19 <?RemoveIfEmpty recursive?>

20 <!--1.1--><Attribute id="productName"><?Target Product Name?></Attribute>

Allow empty tags No

Back

Next

Finish

Cancel

OIEP

Outbound Integration Endpoint
Configuration
Event Triggering Definitions
Background Processes
Statistics
Error Log Excerpts
Log
Status

Configuration
Event Queue Configuration
Output Templates

Object-Eventtype	Format	Pre-Processor	Post-Processor
> Item (Modify)	SmartLabel (3 mappings)	None	None

Select format

Format
Mapping
Advanced

SmartLabel

Converts to a SmartLabel XML format based on a template.

Template

```

1 <!--
2 Element definitions:
3     RequiredAttributes - Defined by SmartLabel as 'Required' for both Food and Non-Food produ
4     FoodRequiredAttributes - Defined by SmartLabel as 'Required' for only Food products
5     NonFoodRequiredAttributes - Defined by SmartLabel as 'Required' for only Non-Food product
6     VoluntaryAttributes - Defined by SmartLabel as 'Voluntary' for both Food and Non-Food pro
7     FoodVoluntaryAttributes - Defined by SmartLabel as 'Voluntary' for Food products
8     NonFoodVoluntaryAttributes - Defined by SmartLabel as 'Voluntary' for Non-Food products
9 -->
10
11 <SmartLabel version="1.5" templateVersion="1.1">
12
13     <Product id="[?Target Product ID?]">
14         <?Record?>
15
16         <Header>
17
18             <RequiredAttributes>
19                 <?RemoveIfEmpty recursive?>
20                     <!--1.1--><Attribute id="productName"><?Target Product Name?></Attribute>
21                     <!--1.1--><Attribute id="brandName"><?Target Brand Name?></Attribute>

```

Allow empty tags
No

OK
Cancel

Configuring the SmartLabel Template

Selecting the SmartLabel format in Export Manager or on an OIEP displays the default template, which includes required and voluntary attributes for both food and non-food products. This template is intended to be modified to include only the attributes you require and any additional comments to best meet your needs.

Additional comments can be added to your modified template as needed. The following text identifies comments:

<!-- shows where a comment begins

--> indicates the end of comment

The modified template is saved with the export configuration or in an OIEP output template. For information about saving a configuration for Export Manager, see **Running a Data Export**. For information about an OIEP output template, see **OIEP - Event-Based - Output Templates Flipper** or **OIEP - Select Objects - Output Templates Flipper**.

Note: To restore the default template once it has been modified, create a new export using Export Manager or a new output template using OIEP, and select the SmartLabel format.

SmartLabel Attributes

SmartLabel landing pages split the product attribution into the following sections: header, footer, attribute tabs, and sub-sections within attribute tabs. The export template uses these same groupings.

To make analysis and mapping easier, all attributes have been grouped into six possible categories. These indicate whether an attribute is required or voluntary for food and non-food products. Attribute groupings are identified within the XML by the following six elements:

- RequiredAttributes - Defined by SmartLabel as 'Required' for both Food and Non-Food products
- FoodRequiredAttributes - Defined by SmartLabel as 'Required' for only Food products
- NonFoodRequiredAttributes - Defined by SmartLabel as 'Required' for only Non-Food products
- VoluntaryAttributes - Defined by SmartLabel as 'Voluntary' for both Food and Non-Food products
- FoodVoluntaryAttributes - Defined by SmartLabel as 'Voluntary' for Food products
- NonFoodVoluntaryAttributes - Defined by SmartLabel as 'Voluntary' for Non-Food products

The default template values are meant to assist during the mapping process in STEP and the client SmartLabel solution. These values should be changed based on business requirements. It is not required that the values used exist as attribute names or IDs within STEP.

Required Attributes

For all required attributes, the default template includes the following predefined details:

1. SmartLabel Attribute Number - Defined in FOOD_ATTRIBUTES-Smartlabel_V_1-5 and NON-FOOD_ATTRIBUTES-Smartlabel_V_1-5.

2. Attribute id - Value that will be included in the export.
3. Attribute description - Value will be seen during mapping process.

For example, the following image shows the required attribute 3.1 is included in the default template as a comment, along with the appropriate attribute ID and attribute description. The numbers correspond to the list above.

```
<FoodRequiredAttributes>
  <1 moveIfE2y recursive?>
  <!--3.1--><Attribute id="servingSize"><?Target Serving Size?></Attribute>
```

Voluntary Attributes

Although voluntary attributes have not been defined in the default template, a placeholder is included as well as the SmartLabel attribute numbers.

For example, a placeholder for voluntary attribute number 1.7 is included as a comment in the default template, but the ID and description are not provided.

```
<VoluntaryAttributes>
  <?RemoveIfEmpty recursive?>
  <!--1.7-->
</VoluntaryAttributes>
```

Modifying the SmartLabel Template

Modification to the template should be expected so that it will fit each unique data model. Attributes may need to be added, removed, or renamed. In addition, template changes may be required for attributes that have multiple values or exist on a referenced object.

Note: To quickly find the examples shown in this topic, it is helpful to display the unmodified default template in STEP, use Ctrl+A to select all text, and then paste it into an XML editor tool that includes a search feature.

MultiValues

The default template has been predefined with MultiTarget values based on GS1 attribute standards.

If an attribute has multiple values that will get sent to SmartLabel, MultiTarget must be used:

```
<Attribute id="ingredientPurpose"><?MultiTarget Ingredient
Purpose?></Attribute>
```

If an attribute has a single value, Target would be used instead:

```
<Attribute id="ingredientPurpose"><?Target Ingredient Purpose?></Attribute>
```

Attributes on Repeating Referenced Objects

If attributes are to be captured from repeating referenced objects, a MultiTarget to the reference must be included in the template.

For example, the template includes this syntax on the ingredients sections, which can be found for other attribution as well.

```
<Ingredients>
  <?MultiTarget Ingredient Reference?>
  <!--Alternate 4.1--><Attribute id="ingredientSequence"><?Target Ingredient Sequence?></Attribute>
  <!--Alternate 4.1--><Attribute id="ingredientName"><?Target Ingredient Name?></Attribute>
  <!--4.8/4.9--><Attribute id="ingredientPurpose"><?MultiTarget FDA categories in 31 CFR - Ingredient Purpose?></Attribute>
</Ingredients>
```

In this example, if ingredient information was not stored on a referenced object then the <?MultiTarget Ingredient Reference?> section (indicated by the arrow above) would be removed and additional attribution added within the <Ingredients> tag as necessary.

Attributes on Nested Repeating Referenced Objects

If attributes are to be captured from nested repeating referenced objects, a MultiTarget to each reference must be added to the template.

An example of this is included in the template and can be used for other attribution.

```
<!-- Format used if multiple sets of serving/preparation information and/or nutrient details are stored on repeating referenced objects
<Nutrient>
  <?RemoveIfEmpty recursive?>
  <?MultiTarget Nutrient Info Reference?>
  <!--3.1--><Attribute id="servingSize"><?Target Serving Size?></Attribute>
  <!--3.2--><Attribute id="numberOfServingsPerPackage"><?Target Number of Servings Per Package?></Attribute>
  <!--3.9--><Attribute id="nutritionalReferralOrDisclosure"><?Target Nutrition Referral or Disclosure Statement?></Attribute>
  <!--3.10--><Attribute id="preparationType"><?Target Preparation Type?></Attribute>

  <NutrientDetail>
    <?RemoveIfEmpty recursive?>
    <?MultiTarget Nutrient Detail Reference?>
    <!--3.3/3.4--><Attribute id="measurementPrecision"><?Target Measurement Precision?></Attribute>
    <!--3.3/3.5--><Attribute id="nutrientType"><?Target Nutrient Type?></Attribute>
    <!--3.3/3.6--><Attribute id="quantityContained"><?Target Quantity Contained?></Attribute>
    <!--3.3/3.7--><Attribute id="dailyValuePercent"><?Target Daily Value Percent?></Attribute>
  </NutrientDetail>
</Nutrient>
-->
```

In this example, data is stored on a repeating 'Nutrient Info' reference and additional data nested below this on a repeating 'Nutrient Detail' reference. To use this example section, the closing comment tag around the nutrient section (indicated by the arrow on the last row) should be moved to the end of first line of the section.

Additionally, the comment tags surrounding the SmartLabel attribute numbers are incomplete (two dashes are required) and must be modified as required.

More nested levels can be added following the same formatting.

Empty Tags

The default template includes `<?RemoveIfEmpty recursive?>` defined above all attribute groups. This will ensure no tags for empty attributes or attribute groups are included in the export. If empty tags are required, then this should be placed within specific attribute groups or removed all together from the template.

STEPXML Format

STEPXML is the native XML format for STEP. Where other supported import / export formats are limited to specific super types and node types, STEPXML makes it possible to import / export the majority of the STEP configuration and data objects. STEPXML is especially suited for transferring data and configurations between systems in a DTAP (Development, Testing, Acceptance, and Production) environment and for inbound integrations with systems / middleware capable of producing STEPXML.

As illustrated below, all data that enters and exits STEP via any inbound / outbound functionality uses the STEPXML format. For example, when a CSV file is selected for import, STEP converts the data to STEPXML and then imports the data.



For users familiar with XML file structure and creation, the Advanced STEPXML format provides a template parameter where an XML can be supplied manually. For more information, see the **Advanced STEPXML Format** topic.

Limitations

The following limitations exist with STEPXML:

- Historical data (data in old revisions) cannot be exported or imported.
- Asset content (the binary files associated with Asset objects) can be exported but not imported.
- Table data can be exported but not imported.
- Individual values for existing multivalued attributes cannot be exported or imported.
- Legal source object types and legal target object types for existing reference / link type definitions cannot be removed or added.
- Legal units for existing attributes cannot be removed.

Format Availability

STEPXML is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**
- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

Mapping

Since STEPXML is the native format for STEP, mapping data is not required and the Map Data step is disabled.

Inbound Data

STEPXML can contain a number of different processing instructions for import. Some of these instructions are similar to the options available in the import wizard for the tabular formats (non-XML formats like Excel, CSV, and FixedWidth).

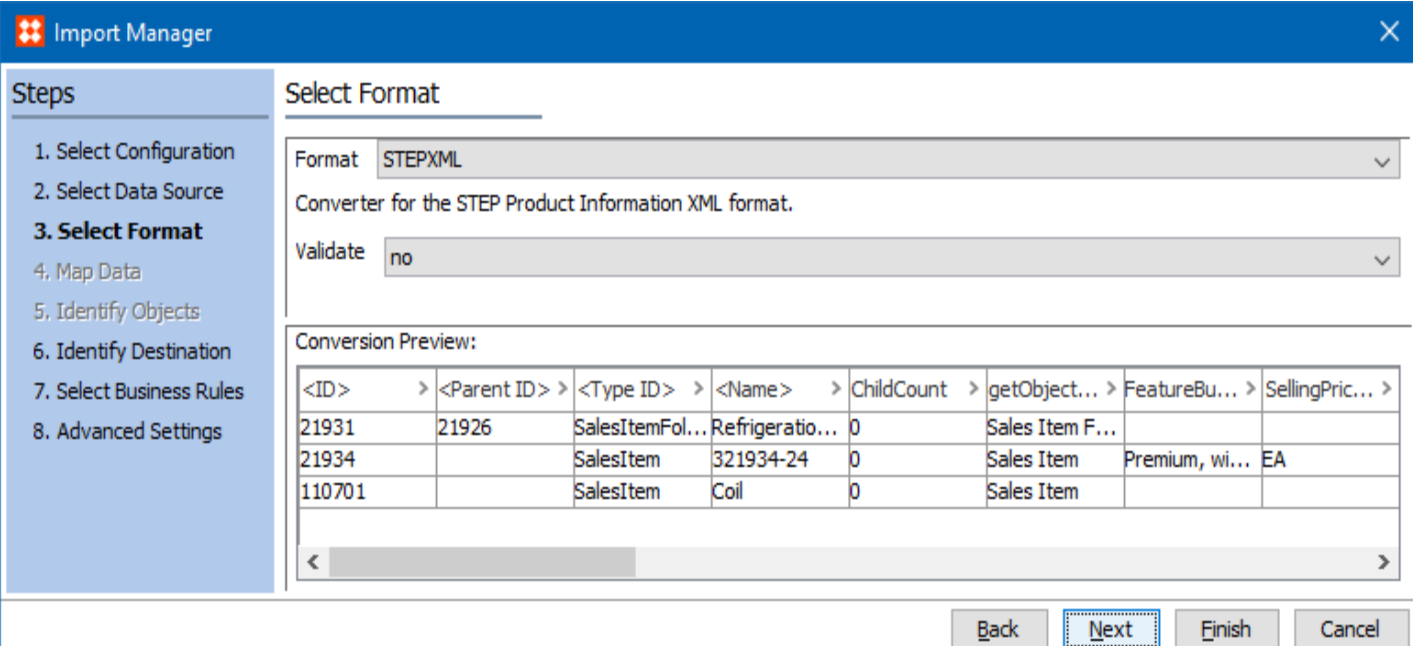
When updating most objects via import, the STEP ID is required. However, for products, entities, classifications, and assets, a key can also be used for identification. For more information about keys, see the **Unique Keys** section of the **System Setup / Super User Guide** documentation.

It is only necessary to include the data needing to be updated. By default, existing data that is not included in the import file will not be deleted or modified.

Inbound Parameters

An inbound file can create new objects, update existing objects, and delete existing values and links, among other options. For more information, see the **STEPXML Tags and Examples** documentation.

Import Manager



Import Manager

Steps

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

Select Format

Format: **STEPXML**

Converter for the STEP Product Information XML format.

Validate: **no**

Conversion Preview:

<ID>	<Parent ID>	<Type ID>	<Name>	ChildCount	getObject...	FeatureBu...	SellingPric...
21931	21926	SalesItemFol...	Refrigeratio...	0	Sales Item F...		
21934		SalesItem	321934-24	0	Sales Item	Premium, wi...	EA
110701		SalesItem	Coil	0	Sales Item		

Back Next Finish Cancel

IIEP

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
5. Configure Processing Engine
5.1. Select Sample File
5.2. Select Data Source
5.3. Select Format
5.4. Map Data
5.5. Identify Objects
5.6. Identify Destination
5.7. Select Business Rules
5.8. Advanced Settings
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Processing Engine : Select Format

Format STEPXML
Converter for the STEP Product Information XML format.
Validate no

Conversion Preview:

<ID>	>	<Parent ID>	>	<Type ID>	>	<Name>	>	ChildCount	>	getObject...	>	FeatureBu...	>	SellingPric...	>
21931		21926		SalesItemFol...		Refrigeratio...		0		Sales Item F...					
21934				SalesItem		321934-24		0		Sales Item		Premium, wi...		EA	
110701				SalesItem		Coil		0		Sales Item					

Back Next Finish Cancel

Outbound Data

The STEPXML format uses a number of parameters to determine the data included or excluded for the export file. While the output format is the same as using Advanced STEPXML, these parameters eliminate the need for creating an XML template.

Outbound Parameters

All parameters available via STEPXML are also available in Advanced STEPXML, which includes additional options. An export using the STEPXML format generates a file that can be used as a template for additional imports or exports using Advanced STEPXML. For details about each STEPXML parameter, see the **STEPXML Outbound Parameters** documentation.

Export Manager

Export Manager

Steps

1. Select Configuration

2. Select Objects

3. Select Format

4. Map Data

5. Advanced

6. Select Delivery Method

Select Format

STEPXML

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

-Global Settings-

Export Data for Selected Contexts

No

Include Schema Reference

No

-Data Objects-

Include Inherited Data

No

Flatten Hierarchies

No

Back

Next

Finish

Cancel

OIEP

System Setup

Inbound Integration Endpoints

Match Codes and Matching Algorithm

Outbound Integration Endpoints

- 1WS 7.0 Provider Sender
- Asset Caching
- Change Attribute Link
- CSV_OIEP
- GREExport
- Item Modify
- Item Modify Non STEPXML
- MultiValue MultiContext
- Static Data Set
- Website

Web UIs

Workflow Profiles

Workflows

- Derived Events

Object Types & Structures

Tags

Units

Users & Groups

Reference Types

Workspaces

GREExport - Configuration

Outbound Integration Endpoint Configuration Event Triggering Definitions Background Processes Sta

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor
Contact (Create, Modify, Delete)	STEPXML	None

Select format

Format Mapping Advanced

STEPXML

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

Global Settings

Export Data for Selected ContextsNo

Include Schema ReferenceNo

Data Objects

Include Inherited DataNo

OK

Cancel

© Stibo Systems - Confidential - STEP Trailblazer 8.3

333

STEPXML Outbound Parameters

The following parameters are available when exporting using the STEPXML format and are separated into the following categories:

- **Global Settings** are applied to both data objects and configuration objects.
- **Data Objects** are applied to objects that hold data, with the most commonly used options at the top, and the others listed in groups of similar options. For example, the classification-related options are listed together.
- **Configuration** options are listed in alphabetical order, and are applied to the item described in the label.
- **Publishing** options are typically used for a print application.

Select Format

STEPXML

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

-Global Settings-

Export Data for Selected ContextsNo

Include Schema ReferenceNo

-Data Objects-

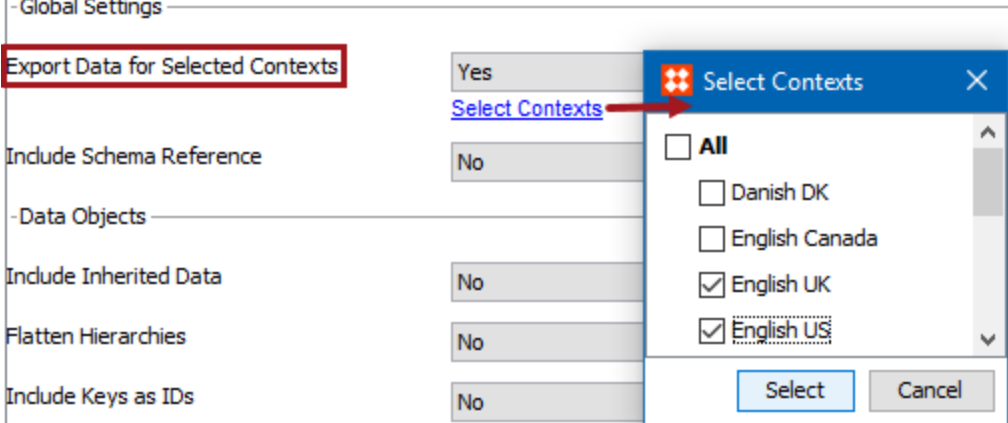
Include Inherited DataNo

Flatten HierarchiesNo

Include Keys as IDsNo

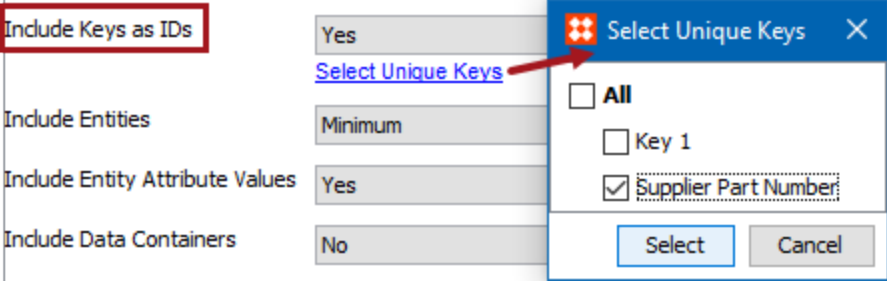
Global Settings

Global Settings Parameter	Description
Export Data for Selected Contexts	Select No to export only for the current context. Select Yes to export multiple contexts and display the Select Contexts link. Click the link to display the Select Contexts dialog. Select the necessary contexts and click the Select button.

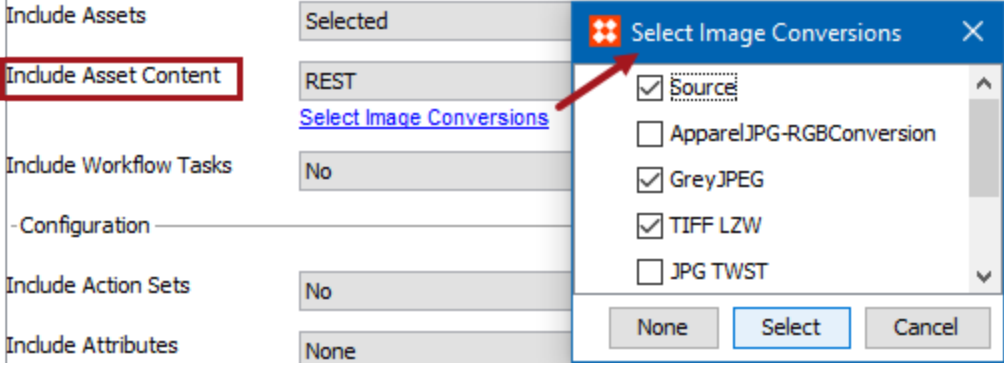
Global Settings Parameter	Description
	 For more information, see the Contexts section of the System Setup documentation or the Multi-Context Data in STEPXML topic.
Include Schema Reference	Select No or Yes to output a reference to the schema file. For more information about schema, click the STEP API Documentation button on the STEP WebStart page.

Data Objects

Data Objects Parameter	Description
Include Inherited Data	Select No or Yes to control the output of inherited values and references. For more information, see Export Inheritance in STEPXML Example.
Flatten Hierarchies	Select No to output STEPXML in a nested layout. Select Yes to output a non-nested layout, where values and references localized on parent objects are included on each object below the parent. This option does not analyze the validity of the attributes. Values localized on parent objects will be exported and included on each exported leaf object.
Include Keys	Select No to ignore keys.

Data Objects Parameter	Description
as IDs	Select Yes to use keys instead of STEP IDs and display the Select Unique Keys link. Click the link to display the Select Unique Keys dialog. Select the necessary keys, and click the Select button.  For more on keys, see the Unique Keys section of the System Setup documentation. For an Advanced STEPXML template, this option is included in the IncludeKey Tag. For more information, see the IncludeKey Tag in STEPXML topic.
Include Entities	For an entity hierarchy, meta attributes applied, and entity references are exported as follows: • Select All to output all entities and entity references. • Select Minimum or Selected to output entities based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select Referenced to output all entities in the hierarchy, and entities referenced from entities in the hierarchy. • Select None to output no entities or entity references.
Include Entity Attribute Values	Select No or Yes to control output of entity values.
Include Data Containers	Select No or Yes to control output of data containers when exporting entities.
Include Products	For a product hierarchy, the meta attributes applied, attribute links, attribute values, and references from products are exported as follows: • Select All to output all products and product references. • Select Minimum or Selected to output products based on the explanation in the

Data Objects Parameter	Description
	Minimum, Referenced, and Selected in STEPXML documentation. • Select Referenced to output all products in the hierarchy, and products referenced from products in the hierarchy. • Select None to output no products or product references.
Include Product Attribute Values	Select No or Yes to control output of product values.
Include Overridden Products	Select No or Yes to control output of product overrides.
Include Classifications	For classifications, the meta attributes applied, attribute links and references from classifications are exported as follows: • Select All to output all classifications and classification references. • Select Minimum or Selected to output classifications based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select Referenced to output all classifications in the hierarchy. • Select None to output no classifications or classification references.
Include Tables	Select No or Yes to control output of STEP tables as they appear in the preview tab in STEP. Table transformations and commercial data are not included in the exported file. Note that Tables cannot be imported.
Include Table Definitions	For tables, row type definitions, column type definitions, and table type definitions, including applied transformations are exported as follows: • Select All to output all classifications and classification references. • Select Minimum to output classifications based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select None to output no classifications or classification references. The following options are available: None, Minimum, and All.
Include	For assets, the description attributes and meta attributes applied are exported as follows:

Data Objects Parameter	Description
Assets	- Select All to output all assets and activate the 'Include Asset Content' parameter. - Select Selected or Minimum to output assets based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. - Select None to output no asset text, references, nor digital contents. To export asset content, use the 'Include Asset Content' parameter below. For an Advanced STEPXML template, this option is included in the Assets Tag. For more information, see Assets Tag in STEPXML.
Include Asset Content	Requires that the 'Include Assets' parameter is set to All, Selected, or Minimum. Select None to exclude asset contents. Select Binary or REST to export digital asset contents along with the asset text and references, and display the Select Image Conversions link. Both options require a selection of at least one Image Conversion. Click the link to display the Select Image Conversions dialog. Select the necessary conversions and click the Select button.  For details about setting up export of asset content, see Assets and Content with STEPXML in the Digital Asset Exchange documentation
Include Workflow Tasks	Select No or Yes to control exporting STEP workflow task information for objects with active tasks.

Configuration

Configuration Parameter	Description
Include Action Sets	Select No or Yes to control output of a list of all actions applied to each action set.
Include Attributes	For attributes, the validation base types, dimension dependencies applied units, applied LOV hierarchy filters, calculated templates, and so on, are exported as follows: • Select All to output all attributes, including fundamental system specific attributes (having an ID that starts with stibo. or asset.). • Select Selected or Minimum to output attributes based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select None to output no attributes.
Include Attribute Groups	For attribute groups, the view definitions and settings applied to attribute groups are exported as follows: • Select All to output all attribute groups. • Select Selected or Minimum to output attribute groups based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select None to output no attribute groups.
Include Attribute Transformations	Select No or Yes to control output of attribute transformations and their configurations.
Include Bulk Update Configurations	Assets saved as bulk update configurations are exported as follows: • Select All to output all bulk update configurations. • Select Minimum to output bulk update configurations based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select None to output no bulk update configurations.
Include Business Rules (Global) and Libraries	For global business rules and libraries, the ID, name, links, on approve setting, dependencies, valid object types, and templates are exported as follows: • Select All to output all global business rules and libraries. • Select Selected or Referenced to output attributes based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select None to output no global business rules or libraries.

Configuration Parameter	Description
	Templates are exported in Base64 and can only be imported without modifications into another STEP system.
Include Collection Definitions	Select No or Yes to control output of collections groups and collections, including search URLs. The exported file will not contain products, classifications, or assets included in a collection.
Include Component Models	Select No or Yes to control output of details of component models, including ID, name, object types and links. The Component Model node in System Setup displays all available component models on your system.
Include Contexts	Select No or Yes to control output of dimensions, dimension points, contexts, applied dimension points, and locale settings.
Include Context Qualifiers	Select No or Yes to control output of a list of qualifiers used in exported data. When used in combination with export of product, classification, or asset data, controls qualifier IDs and dimension point IDs.
Include Data Container Definitions	Data container types are exported as follows: • Select All to output all data container types. • Select Selected or Minimum to output data container types based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select None to output no data container types.
Include Derived Event Types	Select No or Yes to control output of objects created in System Setup as Derived Event Type. For more information, see Derived Events in the System Setup .
Include eCatalogs	Select No or Yes to control output of eCatalog configurations and price lists. Does not include product selection lists.
Include Event Processors	Select No or Yes to control output of event processor information, including ID, name, links, and configuration. For more information, see Event Processors in the System Setup / Super User Guide documentation. The configurations are exported in Base64 and can only be imported without modifications into another STEP system.
Include Event	Select No or Yes to control output of event queues and applied message templates, output

Configuration Parameter	Description
Queues	formatting, and triggering definitions.
Include Export Configurations	Assets saved as export configurations are exported as follows: • Select All to output all export configurations. • Select Minimum to output export configurations based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select None to output no export configurations.
Include Import Configurations	Assets saved as import configurations are exported as follows: • Select All to output all import configurations. • Select Minimum to output import configurations based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select None to output no import configurations.
Include Integration Endpoints	Select No or Yes to control output of ID, name, and configurations. The configurations are exported in Base64 and can only be imported without modifications into another STEP system.
Include Key Definitions	Select No or Yes to control output of definitions of objects created as keys. For more on keys, see the Unique Keys section of the System Setup documentation.
Include Link, Reference and Object Types	Important: Handle reference types and object types with an ID that start with stibo. carefully since they are fundamental objects. System specific object types, edge types, and system specific reference types are exported as follows: • Select All to output all link, reference, and object types. • Select Selected or Minimum to output only link, reference, and object types in the selected hierarchy based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select None to output no attributes.
Include List of Values	LOV definitions and values in LOVs, including applied meta attributes, validation base types, and applied units are exported as follows: • Select All to output all LOV definitions and values.

Configuration Parameter	Description
	- Select Selected or Minimum to output LOV definitions and values based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. - Select None to output no LOV definitions or values.
Include Match Codes	Select No or Yes to control output of match code information, including ID, name, links, setup group, valid entity hierarchies, and valid object types. For more information, see the Matching, Linking, and Merging documentation.
Include Matching Algorithms	Select No or Yes to control output of matching algorithms, including ID, name, links, and configuration. For more information, see the Matching, Linking, and Merging documentation. The configurations are exported in Base64 and can only be imported without modifications into another STEP system.
Include Setup Entities	Select No or Yes to control output of setup entities including ID, name, links, and configuration. The configurations are exported in Base64 and can only be imported without modifications into another STEP system.
Include Setup Groups	Select No or Yes to control output of setup groups that are defined in system setup to hold integration endpoints, Web UI configurations, STEP workflows, and business rules. For more information, see Setup Groups in the System Setup documentation.
Include System Settings	Select No or Yes to control output of the following default settings defined on the Users and Groups root node: - Image & Document Settings flipper > Dimension Dependencies - Calculated Attribute Settings flipper > Dimension Dependencies - Product Information Manager Default Settings flipper - Flatplanner Default Settings flipper - DTP Default Settings flipper - GDSN Default Settings flipper - Terms List Settings flipper - WebServices Default Settings flipper - Web UI Default Settings flipper - Default Reference Type of Primary Image flipper

Configuration Parameter	Description
	Table Default width and height flipper
Include Table Types	Row type definitions, column type definitions, and table type definitions, including applied transformations are exported as follows: - Select All to output all table types. - Select Minimum to output table types based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. - Select None to output no table types.
Include Tags	Select No or Yes to control output of style tags, character tags, special characters, footnotes, and hyperlinks, including information about rendering, short cuts, and applied output formatting.
Include Transformation Lookup Tables	Assets saved as transformation lookup table configurations are exported as follows: - Select All to output all transformation lookup table configurations. - Select Minimum to output transformation lookup table configurations based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. - Select None to output no transformation lookup table configurations. For information on lookup tables, see the Transformation Lookup Tables topic in System Setup / Super User Guide documentation.
Include Units	Unit groups and units, including applied meta attributes, values, and base unit conversions are exported as follows: - Select All to output all unit groups and units. - Select Selected or Minimum to output unit groups and units based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. - Select None to output no unit groups or units.
Include Users and User Groups	Select No or Yes to control output of all user groups and users, including information about applied privileges, meta attributes, restricted GUI setup, etc. User passwords are not included in the output.
Include Web UI Configurations	Select No or Yes to control output of Web UI configurations in the exported file.

Configuration Parameter	Description
Include Websites	Select No or Yes to control output of websites, with and without workspace.
Include Workflows	Workflow information, including ID, name, links, valid object types, and configuration are exported as follows. The configurations are exported in Base64 and can only be imported without modifications into another STEP system. • Select All to output all workflows. • Select Selected to output workflows based on the explanation in the Minimum, Referenced, and Selected in STEPXML documentation. • Select None to output no workflows.

Publishing

Publishing Parameter	Description
Include Publication Groups	Select No or Yes to control output of publication groups.
Include Publications	Select No or Yes to control output of publications.
Include Publication Sections	Select No or Yes to control output of publication sections.
Include Planned Pages	Select No or Yes to control output of planned pages.
Include Autopage References	Select No or Yes to control output of AutoPage references. For more information, see Exporting and Importing AutoPage Publications in STEPXML in the AutoPage documentation.
Include Baskets	Select No or Yes to control output of details of Flatplanner baskets.

Publishing Parameter	Description
Include Pagination Rules	Select No or Yes to control output of details of pagination rules. For more information, see Exporting and Importing AutoPage Publications in STEPXML in the AutoPage documentation.
Include Actual Page Data	Select No or Yes to control output of details of actual page data.

STEPXML Tags and Examples

Advanced STEPXML tags determine how data is imported and exported from STEP. Update or replace the default template with Advanced STEPXML tags to generate the desired output. Although this is not an all-inclusive list, the most commonly required data can be handled using the following tags, XML attributes, and setup:

- AssetContent Tag in STEPXML
- Assets Tag in STEPXML
- AttributeLink Tag in STEPXML
- Auto Approve in STEPXML
- Backward Compatibility in STEPXML
- Classifications Tag in STEPXML
- Create Objects in STEPXML
- Data Containers in STEPXML
- Delete Objects in STEPXML
- Delete Product Reference in STEPXML
- Delete Values in STEPXML
- Export Inheritance in STEPXML Example
- Filter Products in STEPXML
- Filter References in STEPXML
- Filter Values in STEPXML
- IncludeKey Tag in STEPXML
- Inherit References in STEPXML
- Inherit Values in STEPXML
- Minimum, Referenced, and Selected in STEPXML
- Multi-Context Data in STEPXML
- Products Tag in STEPXML
- Referenced and Embedded XML Attributes in STEPXML
- ReplacementRules Tag in STEPXML
- SequenceProduct Tag in STEPXML
- SingleUpdateMode in STEPXML
- STEP-ProductInformation Tag in STEPXML

AssetContent Tag in STEPXML

The AssetContent tag is used to export digital asset content for both image assets and non-image assets via STEPXML. Asset content can be exported using either Base64 encoded binary, which encodes the asset content into a series of characters, or a relative REST resource URL, which exports a URL that points to where that specific asset content is stored. This tag cannot be used for import.

The AssetContent tag is valid in an outbound Advanced STEPXML format template, which requires an entry-level knowledge of XML structure. Alternatively, the STEPXML format provides the same functionality via a set of parameters, and automatically writes the tag and XML attributes in a template in the background, based on the parameter settings.

For more information on the export process, see **Assets and Content with STEPXML** in the **Digital Asset Exchange** documentation.

Asset Content Input

When exporting asset content via Advanced STEPXML, an input template is required and must include the **AssetContent** tag with the XML attribute **ExportType**. The valid XML attribute values for 'ExportType' are REST and Binary. These values indicate the method used to output the asset content.

As an example, when the value for the 'ExportType' XML attribute in the AssetContent tag is 'REST', the image content output is a URL. When the 'ExportType' value is 'Binary', the output is the image content encoded using Base64.

The AssetContent tag is further defined using the required **ImageConversionConfiguration** tag with the XML attribute **ID** to indicate the desired image configuration(s). In addition to exporting image asset content using one or many saved image conversions, the source image asset is also available for export, meaning no conversion is performed.

Note: At least one image conversion configuration must be included in order to export image asset content, otherwise the AssetContent tag is not output.

When exporting non-image assets, only the ImageConversionConfiguration ID="Source" is valid. Documents, for example, are not affected by image conversion configurations.

For information on creating image conversion configurations, see **Image Conversion Configurations** topic in the **Digital Asset Exchange** documentation.

Relative REST URL Template Example

The AssetContent tag appearing in the following sample STEPXML template outputs a relative REST URL for both the 'source' asset and a conversion with the ID '184968'.

```
<?xml version="1.0" encoding="UTF-8"?>
<STEP-ProductInformation ResolveInlineRefs="true"
FollowOverrideSubProducts="true">
  <Assets ExportSize="Minimum">
    <Asset>
```

```

<Name/>
<AssetPushLocation/>
<ClassificationReference/>
<Values/>
<AssetContent ExportType="REST">
  <ImageConversionConfiguration ID="Source"/>
  <ImageConversionConfiguration ID="184968"/>
</AssetContent>
</Asset>
</Assets>
</STEP-ProductInformation>

```

Binary Content Template Example

The AssetContent tag appearing in the following sample STEPXML template outputs binary content for both the 'source' asset and a conversion with the ID '124011'.

```

<?xml version="1.0" encoding="UTF-8"?>
<STEP-ProductInformation ResolveInlineRefs="true"
FollowOverrideSubProducts="true">
  <Assets ExportSize="Minimum">
    <Asset>
      <Name/>
      <AssetPushLocation/>
      <ClassificationReference/>
      <Values/>
      <AssetContent ExportType="Binary">
        <ImageConversionConfiguration ID="Source"/>
        <ImageConversionConfiguration ID="124011"/>
      </AssetContent>
    </Asset>
  </Assets>
</STEP-ProductInformation>

```

Asset Content Output

Based on the output method selected, several tags and XML attributes are used to identify the asset content.

- **AssetContent** is used for both REST and Binary, holds the required AssetContentSpecification tags, and includes no XML attributes.
- **AssetContentSpecification** is used for both REST and Binary, and holds the image configuration IDs exported using the following XML attributes:
 - **ImageConversionConfigurationID** indicates the image conversion used for the exported asset content. As shown in the examples below, an empty (blank) ID indicates the original, unconverted image is exported. This results from selecting 'Source' when using the STEPXML export option for asset content or when including 'Source' in the Advanced STEPXML template.
 - **RelativeURL** is used only for the REST option and includes the relative URL that can be used to access the asset content, based on the STEP Core REST API.
 - **IncludesBinaryContent** is used only for the Binary option and indicates whether or not the BinaryContent tag is included in the output.
- **AssetBinaryContent** is used only for the Binary option. One of these tags is output for each of the AssetContentSpecification tags in the file and includes the following XML attributes:
 - **ImageConversionConfigurationID** indicates which of the converted images is represented by the binary content that follows. As shown in the examples below, an empty (blank) ID indicates the original, unconverted image is exported. This results from selecting 'Source' when using the STEPXML export option for asset content or when including 'Source' in the Advanced STEPXML template.
 - **MimeType** is used only for the Binary option, and allows the receiving system to decode the binary data and convert it back into an asset file.
 - **Checksum** is used only for the Binary option, allows the receiving system to verify that the binary data has been correctly received and decoded, and enables that system to determine if the data has changed since it was last received.
- **BinaryContent** is used only for the Binary option, holds the BASE64 encoded binary data, and includes no XML attributes.

Note: If no content is available for the selected image(s), the execution report generated by the export contains an entry for the image but the output will not include the XML tag.

Relative REST URL Output Example

When REST is the selected method for export, each configuration ID is accompanied by a relative URL that can be used to access the asset content.

```
<AssetContent>
  <AssetContentSpecification ImageConversionConfigurationID="" RelativeURL="/restapi/assets/21869/content"/>
  <AssetContentSpecification ImageConversionConfigurationID="185016" RelativeURL="/restapi/assets/21869/conversionconfigurations/185016"/>
</AssetContent>
```

The path that displays as the relative URL varies based on the conversion selected:

Assets Tag in STEPXML

The assets tag is used to export the metadata on the asset and can include multiple additional tags, each is described below.

A single <Assets/> tag implies several other default tags. Consider this structure:

```
<?xml version='1.0'?>
<STEP-ProductInformation>
  <Assets>
    <Asset>
      <Name/>
      <Values/>
    </Asset>
  </Assets>
</STEP-ProductInformation>
```

For example, to only export the asset name, use the following structure:

```
<Assets>
  <Asset>
    <Name/>
  </Asset>
</Assets>
```

Name

The <Name/> tag returns the object name as output.

Values

The <Values/> tag returns the attribute values as output.

Note: Since only description attributes can be added to an asset, value inheritance is not applicable.

For example, the following asset object is exported using Advanced STEPXML using the first template displayed previously in this topic. The results are shown below.

Object

Tree

- [-] Assets
 - [+] Icons
 - [+] Illustrations
 - [+] Installation Manuals
 - [+] Logos
 - [+] MSDS Sheets
 - [+] Owners Manual
 - [-] Product Images
 - [-] 1
 - [-] 12
 - Chihuahuaa
 - [+] 2
 - [+] 4
 - [+] 6
 - [+] 9
 - [+] A
 - [+] B
 - [+] C
 - [+] D
 - [+] E
 - [+] F
 - [+] G
 - [+] H
 - [+] I
 - [+] L
 - [+] M
 - [+] O
 - [+] P
 - [+] R
 - [+] S
 - [+] W
 - [+] Y
 - [+] Product Videos
 - [+] Classifications
 - [+] Configurations
 - [+] Custom Excel Template
 - [+] eClass 10
 - [+] ETIM Hierarchy
 - [+] Index Words
 - [+] Merchandising Hierarchy
 - [+] Offers
 - [+] Suppliers

Chihuahuaa rev.2.5 - Images & Documents

Images & Documents

References

Referenced By

Status

State Log

Tasks

Description

Name	Value
ID	115308
Name	Chihuahuaa
Object Type	Product Image
Revision	2.5 Last edited by USERY on Fri Dec 01 05:14:59 EST 2017
Approved	✖ Last Approved on Tue Nov 21 04:05:42 EST 2017
Translation	Not Translated
Path	Classification 1 root/Assets/Product Images/1/12/Chihuahuaa
Content In	Language=English
Catalog Group Theme	abc
Asset Keywords	abc
GetClassificationsOfAnAsset	fx 20 12

System Properties:

Name	Value
Class	abc True color
Colorspace	abc RGB
Compression	abc JPEG
Depth	123 8 (bits/sample)
Extension	abc jpg
Filename	abc 136.JPG
Format	abc JPEG (Joint Photographic Experts Group JFIF image)
Height	123 1151.46 (mm)
MIME Type	abc image/jpeg
Pixel Height	123 3264 (pixels)
Pixel Width	123 2448 (pixels)
Profile	abc Custom EXIF, Custom XMP
Samples	123 3 (samples/pixel)
Size	abc 2,270,288
Upload Time	abc 2017-01-29 11:29:37
Width	123 863.59 (mm)
Horizontal DPI	123 72 (dpi)
Vertical DPI	123 72 (dpi)

Results

The following excerpt of the output shows the values from the Description flipper and the System Properties flipper of the asset.

```

17 <STEP-ProductInformation ExportTime="2018-03-02 16:11:18" ExportContext="Context1" ContextID="Context
18
19 <Assets>
20 <Asset ID="115308" UserTypeID="ProductImage">
21 <Name>Chihuahua</Name>
22 <Values>
23 <Value AttributeID="asset.pixel-height" UnitID="pixels">3264</Value>
24 <Value AttributeID="asset.size">2270288</Value>
25 <Value AttributeID="asset.compression">JPEG</Value>
26 <Value AttributeID="asset.ydpi" UnitID="dpi">72</Value>
27 <Value AttributeID="asset.format">JPEG (Joint Photographic Experts Group JFIF image)</Value>
28 <Value AttributeID="asset.xdpi" UnitID="dpi">72</Value>
29 <Value AttributeID="asset.width" UnitID="mm">863.5999999999999</Value>
30 <Value AttributeID="asset.filename">136.JPG</Value>
31 <Value AttributeID="asset.uploaded">2017-01-29 11:29:37</Value>
32 <Value AttributeID="asset.class">True color</Value>
33 <Value AttributeID="asset.extension">jpg</Value>
34 <Value AttributeID="asset.pixel-width" UnitID="pixels">2448</Value>
35 <Value AttributeID="asset.height" UnitID="mm">1151.4666666666665</Value>
36 <Value AttributeID="asset.colorsapce">RGB</Value>
37 <Value AttributeID="asset.depth" UnitID="bits/sample">8</Value>
38 <Value AttributeID="asset.mime-type">image/jpeg</Value>
39 <Value AttributeID="asset.profile">Custom EXIF, Custom XMP</Value>
40 <Value AttributeID="asset.samples" UnitID="samples/pixel">3</Value>
41 <Value AttributeID="GetClassificationsOfAnAsset" Derived="true">20
42 12</Value>
43 </Values>
44 </Asset>
45 </Assets>
46 </STEP-ProductInformation>

```

AttributeLink Tag in STEPXML

The **AttributeLink** tag outputs local attribute links for the products, product-overrides, or classifications in the export.

```
<AttributeLink/>
```

IncludeInherited

When modified with the **IncludeInherited** option, the **AttributeLink** tag also exports inherited attribute links.

```
<AttributeLink IncludeInherited="true">
```

The **AttributeLink** tag is allowed within the following tags:

- Product - Attribute links are inherited from parent products to top-level products.
- Product-Overrides - Attribute links are inherited from overridden product to product-override. Attribute links inherited from Classifications are not included.
- Classification - Attribute links are inherited from parent classifications to top-level classifications.

Template

No Inheritance	IncludeInherited
<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <Product> <Name/> <AttributeLink> <MetaData/> </AttributeLink> </Product> </Products> </STEP-ProductInformation></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <Product> <Name/> <AttributeLink IncludeInherited="true"> <MetaData/> </AttributeLink> </Product> </Products> </STEP-ProductInformation></pre>

Results

Only local attribute links for the exported product are output.

No Inheritance

```
<STEP-ProductInformation ExportTime="2015-12-23 14:58:34" ExportC
  <Products>
    <Product ID="20878" UserTypeID="Level4" ParentID="20876">
      <Name Changed="true">Batteries Level4</Name>
      <AttributeLink AttributeID="Voltage"/>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Local and inherited attribute links for the exported product are output.

IncludeInherited

```
<STEP-ProductInformation ExportTime="2015-12-23 15:02:53" ExportCon
  <Products>
    <Product ID="20878" UserTypeID="Level4" ParentID="20876">
      <Name Changed="true">Batteries Level4</Name>
      <AttributeLink AttributeID="Voltage"/>
      <AttributeLink AttributeID="MaximumVoltage" Inherited="3"/>
      <AttributeLink AttributeID="ETLListed" Inherited="3"/>
      <AttributeLink AttributeID="CSAListed" Inherited="3"/>
      <AttributeLink AttributeID="ULListed" Inherited="3"/>
      <MetaData>
        <Value AttributeID="DisplaySequence">2</Value>
      </MetaData>
    </Product>
  </Products>
</STEP-ProductInformation>
```

AutoApprove in STEPXML

The data structure between the Approved workspace and the Main workspace must be consistent, and is automatically enforced within STEP. This required consistency causes AutoApprove to fail under the following conditions:

1. An object cannot be approved until its parent is in the approved workspace.
2. A reference or link cannot be approved until the target is in the approved workspace.
3. A mandatory attribute or a mandatory reference / link type cannot be approved until the required data exists.
4. An object that causes any business rule that is configured to run “on Approval” or run during import cannot be approved until the condition is satisfied.

In each of these cases, an error message is logged in the execution report when the approval fails.

Configuration

Insert **AutoApprove="Y"** in the **STEP-ProductInformation** tag to approve objects during import, as illustrated in the following example:

```
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main"
AutoApprove="Y">
    ...
</STEP-ProductInformation>
```

For more information, see **STEP-ProductInformation Tag in STEPXML**.

Backward Compatibility in STEPXML

The STEPXML schema is extended continuously but should always be backwards compatible. Due to this fact, you may encounter legacy naming in the STEPXML format.

The legacy names that most frequently cause problems are:

- **UserType** – generally this means object type. Nested UserType elements inside the UserTypes element represent object type definitions. UserTypeLink elements inside the Attributes element indicates that the attribute is valid for specific object type. UserTypeID attributes (for example in the Product element) specifies that the product is of a specific object type.

```
<Product ID="I-SalesItem-1111" UserTypeID="SalesItem" ParentID="I-Level3-111">
```

- **ProductMode** – attribute elements have a ProductMode attribute to indicate a description or specification attribute.

This example shows the attribute is a description attribute (value: Property).

```
<Attribute ID="EAN" Multivalued="false" ProductMode="property" FullTextIndexed=false"
  ExternallyMaintained="true" Deriver=false" Selected="true" Referenced="true"> <Name>EAN</Name>
</Attribute>
```

This example shows the attribute is a specification attribute (value: Normal).

```
<Attribute ID= "Depth" Multivalued="false" DefaultUnitID="unece.unit.CMT" ProductMode="Normal" FullTEXTIndex="false"
  ExternallyMaintained="false" Derived="false" Selected="true" Referenced="true"></Attribute>
</Attribute>
```

- **Derived** – attribute element that specifies if the attribute is calculated.

```
<Value AttributeID="CountryOfOrigin" Derived="true">GB</Value>
<Value AttributeID="ManufacturerEAN" Derived="true">9513864492123</Value>
```

- **ClassificationReference** – used inside both the Product and Asset elements. For assets, this is the parent link (assets can have multiple parents).

```
<ClassificationReference ClassificationID="SuppliesAllAssets" />
<ClassificationReference ClassificationID="22806" />
```

Tree

MA

Manual EN

Manual DE

Manual ES

Manual FR

Manual EN rev.0.3 - References

Images & DocumentsReferencesReferenced ByStatusState LogTasks

Classifications

ID	Name	Object Type
SuppliesAllAssets	Assets	Suppliers Assets
22806	MA	Asset Level 2

Link to Classification

For products, the element represents product-to-classification links.

```
<Product ID="flashlight case" UserTypeID="Item" ParentID="Flashlights Items">
  <Name>flashlight case</Name>
  <ClassificationReference ClassificationID="ProductsGaloreProducts" Type="SupplierLink">
```

Tree

Tools

Task Lighting

Flashlights

Flashlights Items

flashlight case

ProductSub ProductsReferencesReferenced ByImages & Documents

(ItemReferencesClassification)

Reference Type	Target
Supplier Link	Suppliers/Products Galore/Products

Suppliers/Products Galore/Products
ID = ProductsGaloreProducts

Classifications Tag in STEPXML

The Classification tag can include an XML attribute.

```
<?xml version="1.0" encoding="UTF-8"?>
<STEP-ProductInformation >
  <Classifications ExportSize="Minimum">
    <Classification IncludeParent="true"/>
  </Classifications>
</STEP-ProductInformation>
```

When an XML attribute is absent, the default setting is used. For example:

```
<Classifications>
  <Classification IncludeParent="true"/>
</Classifications>
```

The following XML attribute is available:

IncludeParent

- False - default setting
- True - when a classification is exported from STEP, the file includes all parent levels.

Additional STEPXML Tags

The Classification tag can include multiple additional internal tags, each is described below the following sample STEPXML. If no elements are explicitly specified, a default configuration will be used. For an example of the XML, see the online version of this topic.

Name

The <Name/> tag returns the object name as output.

Values

The <Values/> tag returns the attribute values as output. For more information about values output, see the **Inherit Values in STEPXML** section.

ClassificationReference

The <ClassificationReference/> tag returns all classification cross references as output.

MetaData

The <MetaData/> tag returns metadata as output.

Product to Classification Links Owned by the Classification

By default the product to classification links owned by the classification are exported in two places:

1. Under the Product as a reference to the classification.
2. Under the Classification as a reference to the product. For an XML example of the classification links, see the online version of this topic.

The reason the links are exported under the products is historical, and it has not been changed due to backwards compatibility. However, logically they should be exported under the classification, as they are owned by the classification. And when the links are modified, they affect the revision of the classification - not the product. It is possible to stop the classification owned links from being exported under the products by using the recorder option 'IncludeClassificationOwned' on the 'Products' element. By setting this to "false" the classification links will not be exported under the products:

```
<STEP-ProductInformation>

    <Products IncludeClassificationOwned="false">

        <Product>

        </Product>

    </Products>

</STEP-ProductInformation>
```

Create Objects in STEPXML

New objects can be created in the Tree and in System Setup using the STEPXML import.

Note: To create an STEPXML import template, first export relevant data, and use that as base for the import file.

Minimum Data Required

Minimum requirements for information vary based on the object type. For example, to create a new product, the file must include ID, parent ID, and object type ID, as shown in the following sample STEPXML:

```
<?xml version="1.0" encoding="UTF-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main">
  <Products>
    <Product ID="181952LB" UserTypeID="Item" ParentID="18209">
      <Name>181952 LB</Name>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Additionally, a default parent and object type can be specified using the products tag attributes. For more information, see the **Examples** section below and the **Products Tag in STEPXML** topic.

Using Autogenerated IDs

If an ID pattern is specified for the object type being created, the ID can be excluded, as shown in the following sample STEPXML:

```
<?xml version="1.0" encoding="UTF-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main">
  <Products>
    <Product UserTypeID="Item" ParentID="18209">
      <Name>181952 LB</Name>
    </Product>
  </Products>
</STEP-ProductInformation>
```

For more information, see **Autogenerate Using Name Pattern and ID Pattern**.

Using Key Instead of ID

If a STEP Key (unique key) is specified for the object being created, the ID can be excluded. For examples and more information, see the **Creating Objects with Active Keys** topic of the **Keys** documentation within the **System Setup / Super User Guide**.

Examples

As an example of minimum required data, the following STEPXML would create a new SalesItem object under SalesItemFamily-12456 with a default ID and default name, but nothing else:

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main" >
  <Products DefaultParentID="SalesItemFamily-12456"
DefaultUserID="SalesItem">
    <Product/>
  </Products>
</STEP-ProductInformation>
```

However, using the following STEPXML, the new object would include a specified ID and name, as well as several attribute values:

```
<?xml version="1.0" encoding="UTF-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main">
  <Products>
    <Product ID="181951LB" UserID="Item" ParentID="18209">
      <Name>18216 L B</Name>
      <Values>
        <Value AttributeID="TshirtSize">L</Value>
        <MultiValue AttributeID="CountryOfOrigin">
          <Value ID="CN">CHINA</Value>
        </MultiValue>
        <Value AttributeID="Material">100% Cotton</Value>
        <Value AttributeID="ShortItemDescription">T-shirt, short sleeve, Beefy-
T, Mens L, Blue</Value>
        <Value AttributeID="ProductWeight" UnitID="unece.unit.ONZ">2</Value>
      </Values>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Data Containers in STEPXML

By using data container types, users are able to model a composite entity (e.g., customer record) containing data containers (addresses, contact names, contact information, email addresses, phone numbers, etc.) with relevant attributes.

When importing or exporting entity objects, a number of tags are used to identify data containers upon export, and to build out the data container structure for import. Additionally, data container types and data containers can be set up using STEPXML tags.

Setting Up Data Containers

The **DataContainerTypes** tag houses all of the data container details, and the **DataContainerType** tag builds out the structure of the data container.

The below STEPXML shows an example setup that includes three data container types: Shipping Address, Email, and Main Address.

- The `<AttributeGroupLink AttributeGroupID="EntityDataContainers"/>` tag indicates the ID of the attribute group in which the data containers reside. In this case, it is the EntityDataContainers attribute group.
- The `<AttributeLink AttributeID="Country"/>` tag indicates the ID of the attributes are associated with each data container type. Country is just one example.
- `<UserTypeLink UserTypeID="CD_Customer"/>` indicates which entity object type is valid for the data container type. The ID is shown (e.g., CD_Customer). If this tag combo does not appear in the STEPXML, then validity has not been set on that data container type (shown in the Email data container type example STEPXML).

```
<DataContainerTypes>
  <DataContainerType ID="ShippingAddress" MultiValued="true" IDPattern="[id]"
Referenced="true">
    <Name>Shipping Address</Name>
    <AttributeGroupLink AttributeGroupID="EntityDataContainers"/>
    <AttributeLink AttributeID="Street"/>
    <AttributeLink AttributeID="Country"/>
    <AttributeLink AttributeID="State"/>
    <AttributeLink AttributeID="Zip"/>
    <AttributeLink AttributeID="CountryISOCode"/>
    <AttributeLink AttributeID="City"/>
    <UserTypeLink UserTypeID="CD_Customer"/>
  </DataContainerType>
  <DataContainerType ID="Email" MultiValued="false">
    <Name>Email</Name>
    <AttributeGroupLink AttributeGroupID="CustomerData"/>
```

```

    <AttributeGroupLink AttributeGroupID="EntityDataContainers"/>
    <AttributeLink AttributeID="S-Email"/>
  </DataContainerType>
  <DataContainerType ID="MainAddress" MultiValued="false"
IDPattern="MainAddress-[id]" Referenced="true">
    <Name>Main Address</Name>
    <AttributeGroupLink AttributeGroupID="EntityDataContainers"/>
    <AttributeLink AttributeID="Street"/>
    <AttributeLink AttributeID="Country"/>
    <AttributeLink AttributeID="State"/>
    <AttributeLink AttributeID="Zip"/>
    <AttributeLink AttributeID="CountryISOCode"/>
    <AttributeLink AttributeID="City"/>
    <UserTypeLink UserTypeID="CD_Customer"/>
  </DataContainerType>
</DataContainerTypes>

```

The **DataContainerType** tag can include the following valid XML attributes:

MultiValued	- False - Indicates that the data container type only allows for a single data container - True - Indicates that the data container type allows for multiple data containers
ID	Displays the ID for the data container type or data container. When importing, the ID attribute can be omitted if autogenerated IDs are enabled.
IDPattern	Displays the ID pattern set up for the data container type, if applicable. This pattern will apply to all data containers of that type.
Referenced	- When exporting data and choosing to include Data Container Type Definitions, Referenced="true" displays within the DataContainerType tag to indicate that a data container of that type is referenced in other data within the same export. - If there is no Referenced info shown, then a data container of that type is not being exported in the same data set.

Adding and Maintaining Data Containers

The **DataContainers** tag indicates where, within the entity object tags, the data container information starts.

If a data container type allows for multiple containers, then it is indicated by the **MultiDataContainer** tag. Individual data containers are identifiable by the **DataContainer** tag. The Value tag, with the AttributeID XML attribute, is used to show / set up the attributes and attribute values.

Example STEPXML code is shown below:

```
<Entity ID="CUS_114563" UserTypeID="CD_Customer">
  <Name>Customer D</Name>
  <EntityCrossReference EntityID="ADD_120042" Type="CustomerToAddress"/>
  <Values>
    <Value AttributeID="S-Email">customer@company.com</Value>
  </Values>
  <DataContainers>

  <MultiDataContainer Type="ShippingAddress">
    <DataContainer ID="ShippingAddress_Kennesaw">
      <Values>
        <Value AttributeID="Street">3550 George Busbee Pkwy NW</Value>
        <Value AttributeID="Country">United States</Value>
        <Value AttributeID="Zip">30144</Value>
        <Value AttributeID="State">GA</Value>
        <Value AttributeID="City">Kennesaw</Value>
        <Value AttributeID="CountryISOCODE">US</Value>
      </Values>
    </DataContainer>
    <DataContainer ID="ShippingAddress_Canada">
      <Values>
        <Value AttributeID="Street">116 Spadina Ave</Value>
        <Value AttributeID="Country">Canada</Value>
        <Value AttributeID="Zip">M5V 2K6</Value>
        <Value AttributeID="State">ON</Value>
        <Value AttributeID="City">Toronto</Value>
        <Value AttributeID="CountryISOCODE">CA</Value>
      </Values>
    </DataContainer>
  </MultiDataContainer>
  <DataContainer ID="MainAddress_Denmark" Type="MainAddress">
```

```

<Values>
  <Value AttributeID="Street">Axel Kiers Vej 11</Value>
  <Value AttributeID="Country">Denmark</Value>
  <Value AttributeID="Zip">8270</Value>
  <Value AttributeID="City">Højbjerg</Value>
  <Value AttributeID="CountryISOCode">DK</Value>
</Values>
</DataContainer>
</DataContainers>
</Entity>

```

This is how the above code renders in STEP Workbench:

Customer	Data Containers	References	Referenced By	Matching	Status	State Log	Tasks
Description							
Name	>	>	Value				
ID	CUS_114563						
Name	Customer D						
Object Type	Customer						
Revision	0.9 Last edited by USER8 on Thu Jan 26 00:32:10 EST 2017						
Path	Entity hierarchy root/Entity Root/Customer Hierarchy/Customer D						
Email	abc	customer@company.com					

Customer

Data Containers

References

Referenced By

Matching

Status

State Log

Tasks

Contacts

Main Address

	ID	>	Attribute Name	>	>	Value
>	MainAddress_Denmark		City	abc		Højbjerg
			Country	abc		Denmark
			Country ISO Code	abc		DK
			State	abc		
			Street	abc		Axel Kiers Vej 11
			Zip	12a		8270

Secondary Address

Shipping Address

	ID	>	City	>	Country	>	Country ISO...	>	State	>	Street	>	Zip	>
>	ShippingAddress_Canada		Toronto		Canada		CA		ON		116 Spadina Ave		M5V 2K6	
>	ShippingAddress_Kennesaw		Kennesaw		United States		US		GA		3550 George Busbee Pkwy NW		30144	
>	Add Data Container													

Delete Objects in STEPXML

Products, entities, classifications, assets, attributes, LOVs, and units can be deleted via STEPXML. The delete action varies based on the object type and the revisability setting, as follows:

- For workspace revisable object types (all object types except possibly entities), delete moves the object to the Recycle Bin.
- For attributes, LOVs, units, and global revisable entities, delete purges the object.

For more information on revisability, see **Managing Revisions in STEP** or **Revisability on Entity Object Type**.

For example, the following STEPXML moves four product objects (which are always workspace revisable) to the Recycle Bin.

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main">
  <DeleteProducts>
    <DeleteProduct ID="L7629"/>
    <DeleteProduct ID="L7321"/>
    <DeleteProduct ID="L7847"/>
    <DeleteProduct ID="L7623"/>
  </DeleteProducts>
</STEP-ProductInformation>
```

Event-Based OIEP Triggered by Deleting Products, Classifications, and Assets

When an Outbound Integration Endpoint (OIEP) is configured to listen for Delete events, use the **DeleteProducts**, **DeleteClassifications**, or **DeleteAssets** tags to output information about deleted objects. After deleting the object, the OIEP is triggered by the Approve Deletion action within the Recycle Bin.

```
<DeleteProducts/>
<DeleteClassifications/>
<DeleteAssets/>
```

Note: The Force Delete option is not available for products, classifications, or assets when an OIEP is configured to deliver deletion information.

Template

```
<STEP-ProductInformation ExportDeletedData="true">
  <DeleteProducts/>
</STEP-ProductInformation>
```

Results

Once the deleted Product (Classification or Asset) displays in the Recycle Bin, use Approve Deletion to trigger the OIEP. The ID of the deleted object is included in the OIEP output.

```
<STEP-ProductInformation ExportTime="2015-12-31 15:24:43"  
  <DeleteProducts>  
    <DeleteProduct ID="124311"/>  
  </DeleteProducts>  
</STEP-ProductInformation>
```

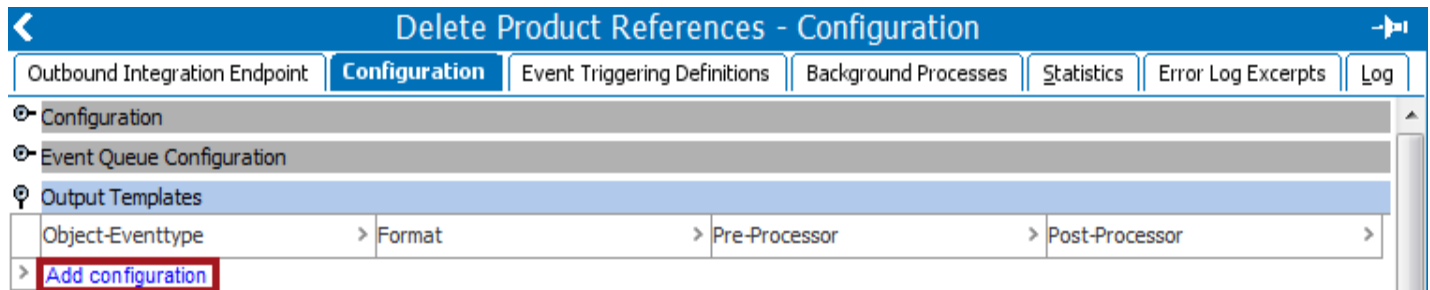
For more information on event-based OIEPs, see the **Outbound Integration Endpoints** documentation.

Delete Product Reference in STEPXML

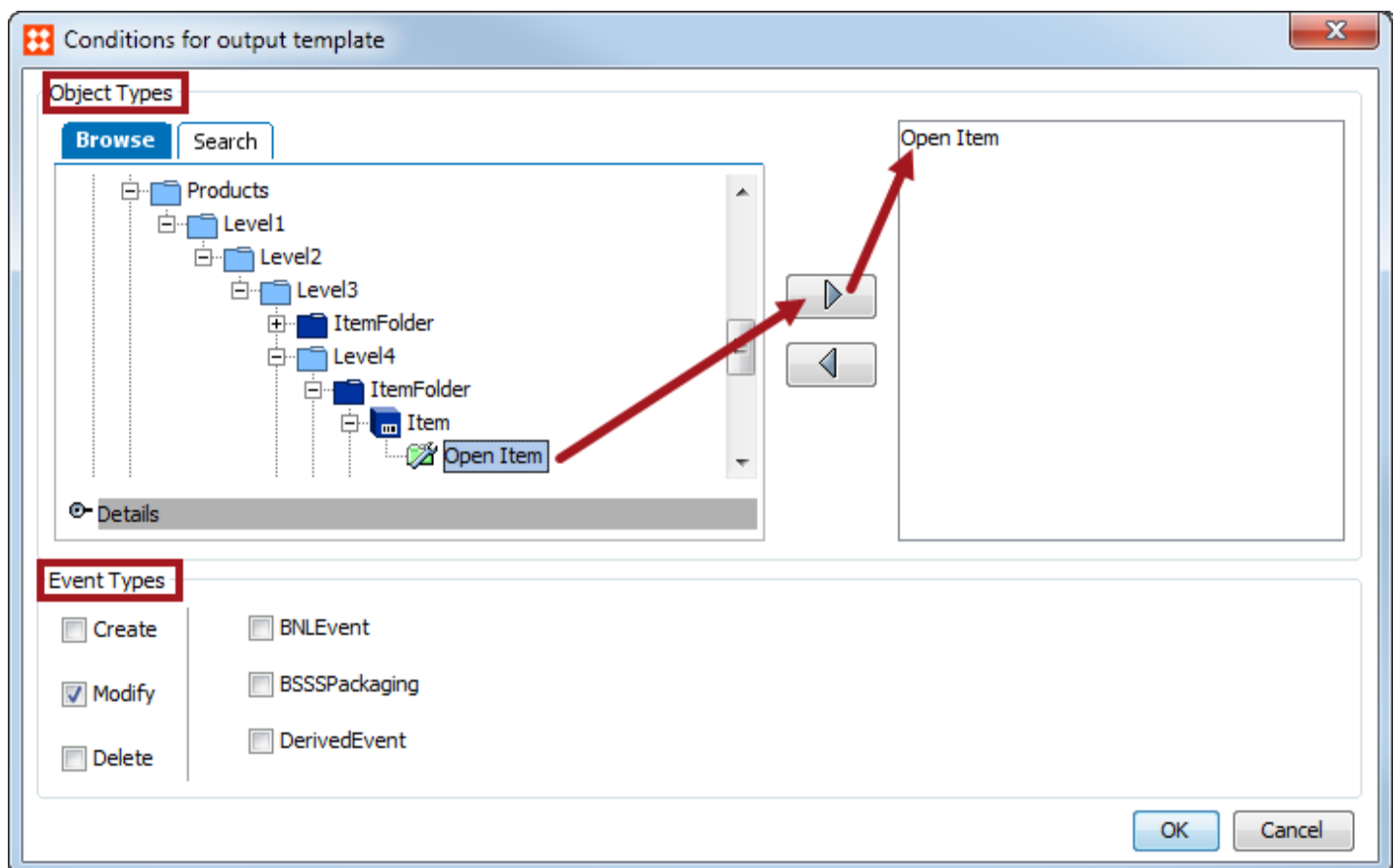
When an Outbound Integration Endpoint (OIEP) is configured to listen for Modify events, use the **STEP-ProductInformation** tag with the **ExportDeletedData** STEPXML attribute to output information about deleted product references. After deleting the reference, approving the product triggers the OIEP.

Object Type and Event Type Selection

View the OIEP and open the Configuration tab. Under the Output Templates flipper, click the **Add configuration** link.



Set the Object Types and Event Types.

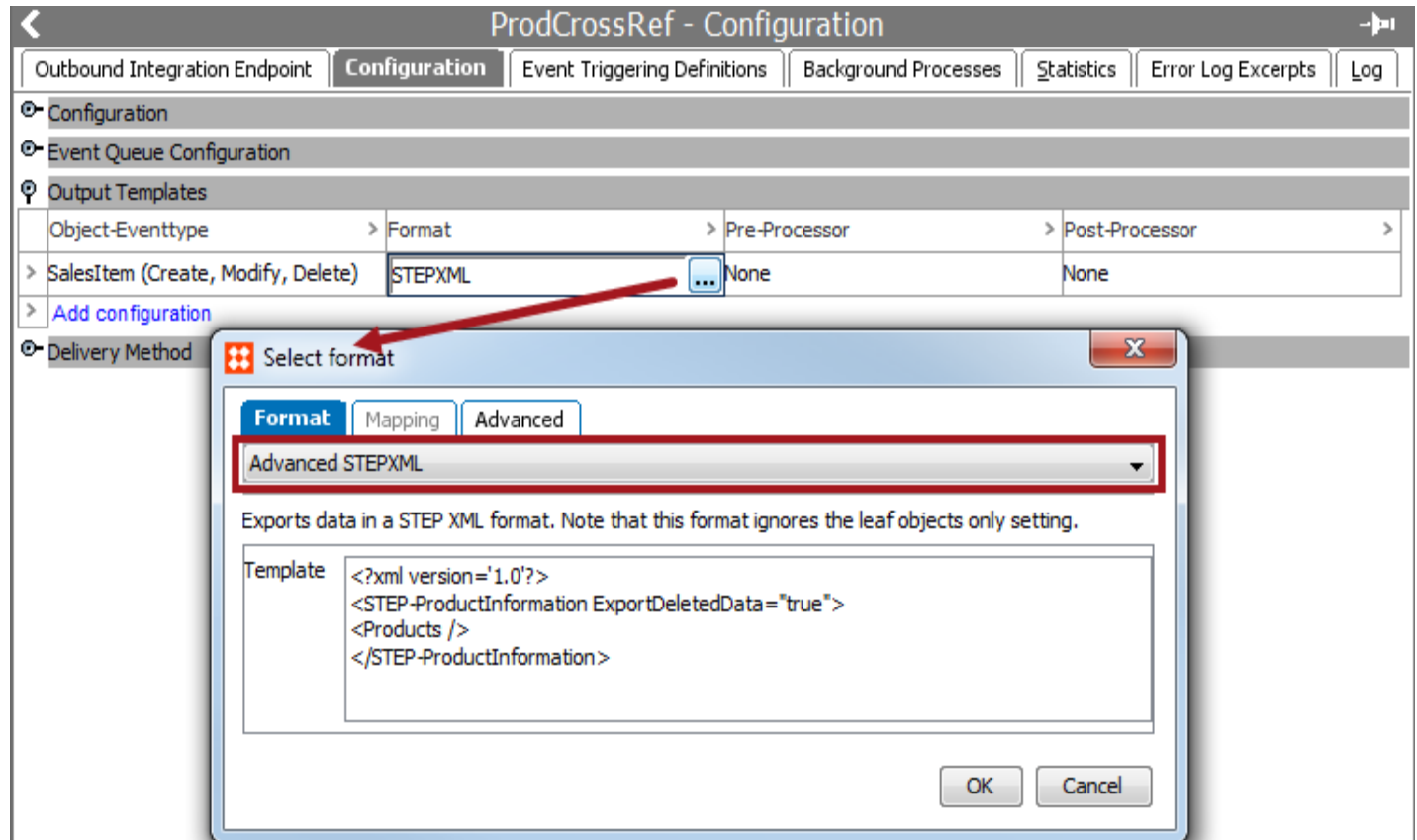


Template

In the Output Template, click the ellipsis button (...), and set the format to **Advanced STEPXML**.

Provide the following text in the Template field.

```
<?xml version='1.0'?>
<STEP-ProductInformation ExportDeletedData="true">
<Products/>
</STEP-ProductInformation>
```



Result

When a Product Reference is deleted and the product is approved, the **DeleteProductCrossReference** tag indicates that the Product Reference is intended to be removed in the downstream system.

The previous settings generated the following Advanced STEPXML output document.

```

<STEP-ProductInformation ExportTime="2015-09-30 11:12:22" ExportContext="Context1" ContextID="Cont
  <Products>
    <Product ID="22163" UserTypeID="SalesItem" ParentID="22157">
      <Name>22163-01</Name>
      <ProductCrossReference ProductID="22155" Type="PrimaryDataSource"/>

      <AssetCrossReference AssetID="22215" Type="PrimaryProductImage"/>

      <Values>
        <Value AttributeID="AnnualSalesForecastMaximum">300</Value>
        <Value AttributeID="AnnualSalesForecast,Minimum">100</Value>
        <Value AttributeID="FeatureBullet3">26 slats of layer-glued birch adjust to your body wei
        <Value AttributeID="SellingPrice" UnitID="iso4217.unit.USD">1299</Value>
        <Value AttributeID="FeatureBullet1">Made of solid wood, which is a durable and warm natur
        <Value AttributeID="DescriptionTable">Comfy bed, espresso, Queen</Value>
        <Value AttributeID="FeatureBullet2">Adjustable bed sides allow you to use mattresses of d
        <Value AttributeID="DescriptionLong">Made of solid wood, which is a durable and warm natu
        <Value AttributeID="DescriptionWeb">Made of solid wood, which is a durable and warm natur
        <Value AttributeID="SalesItemShortDescription">Bed, Espresso, Comfy, Queen</Value>
        <Value AttributeID="SellingPriceUOM">EA</Value>
        <Value AttributeID="Parent" Derived="true">Beds Sales Items</Value>
        <Value AttributeID="Path" Derived="true">Products | Furniture | Bedroom | Beds | Beds Sale
        <Value AttributeID="Category" Derived="true">Primary Product Hierarchy | Products | Furni
      </Values>
      <DeleteProductCrossReference ProductID="22168" Type="PrimarySupplierItem"/>
    </Product>
  </Products>
</STEP-ProductInformation>

```

Delete Values in STEPXML

Upon import, to delete an existing value, include the appropriate value element and leave the value part blank.

The example below updates the product with ID 'L6576' by setting the value for 'Horizontal Pixels' to 1920 and removing any existing value for 'Manufacturer's Description'.

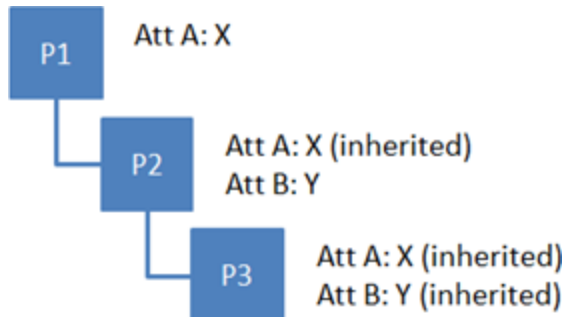
```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main">
  <Products>
    <Product ID="L6576">
      <Name>Panasonic Viera TC-P65VT50</Name>
      <Values>
        <Value AttributeID="Manufacturer's Description"/>
        <Value AttributeID="Horizontal Pixels">1920</Value>
      </Values>
    </Product>
  </Products>
</STEP-ProductInformation>
```


Export Inheritance in STEPXML Example

By default, inherited values and references / links are not exported in STEPXML.

In the sample hierarchy below:

- Product 1 (P1) includes the local value 'X' for attribute A (Att A).
- Product 2 (P2) inherits the value 'X' from attribute A (Att A), and has the local value 'Y' for attribute B (Att B).
- Product 3 (P3) inherits the value 'X' from attribute A (Att A) and inherits the value 'Y' from attribute B (Att B).



Using default settings to export the sample hierarchy, only local values are output and results in the following:

- If only P3 is exported, no values are exported since no attribute values are local (both are inherited).
- If P2 and P3 are exported, the only value is Att B from P2. No values for P3 are exported since no attribute values are local (both are inherited).
- If P1, P2, and P3 are exported, the only values are Att A from P1, Att B from P2. No values for P3 are exported since no attribute values are local (both are inherited).

Override Default Behavior

Overriding the default behavior means that an inherited attribute value is exported once when the object that holds the local value is not exported.

Using the override and continuing with the above example data, the values are output as follows:

- If only P3 is exported, the value for Att A (inherited) and Att B (inherited) are exported.
- If P2 and P3 are exported, for P2, the value for AttA (inherited) and the value for Att B (local) are exported. For P3, no values are exported since both values already appear on P2.
- If only P2 is exported, the value for AttA (inherited) and the value for Att B (local) are exported.
- If P1, P2, and P3 are exported, for P1, the value for Att A (local) is exported. For P2, the value for Att B (local) is exported. For P3, no values are exported since both values already appear on other objects.

For more information on setting the override, see the following topics:

- For STEPXML format, use the Data Objects parameter **Include Inherited Data** = 'Yes' as defined in the **STEPXML Outbound Parameters** topic.
- For Advanced STEPXML format, use the **IncludedInherited** tag options as defined in the **Inherit Values in STEPXML** topic.

Filter Products in STEPXML

When the output from STEP should include only specific objects, use the **FilterUserType** tag with the **ID** XML attribute. Only objects identified by the ID are exported.

```
<FilterUserType ID="ItemFolder"/>
```

Consider the scenario where a template delivers the parent structure for the approved product. Although six (6) product levels exist, only three (3) of the levels and their parents are required.

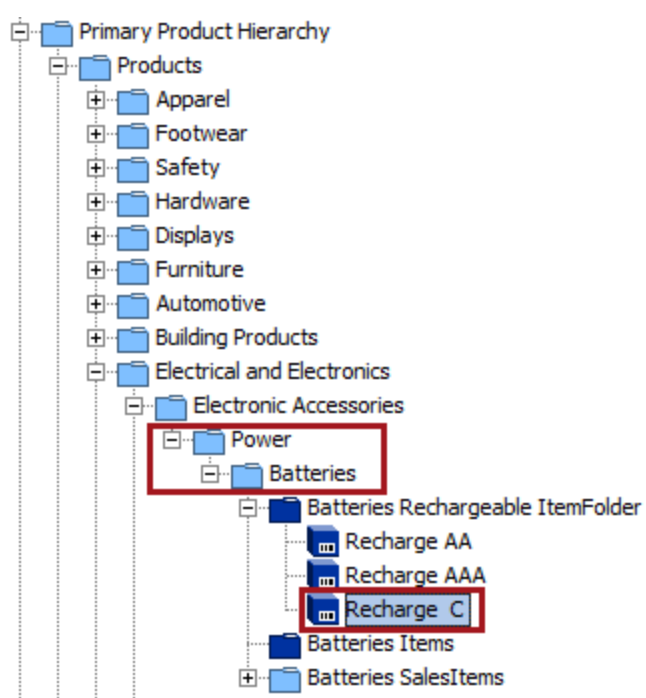
Use multiple instances of the **FilterUserType** tag to include all of the required objects and to filter out the objects that are not needed.

Template

Without Filtering	With Filtering
<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <Product IncludeParent="true"> <Name/> <ProductCrossReference/> <Product/> </Product> </Products> </STEP-ProductInformation></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <FilterUserType ID="Level3"/> <FilterUserType ID="Level4"/> <FilterUserType ID="Level5"/> <Product IncludeParent="true"> <Name/> <ProductCrossReference/> <Product/> </Product> </Products> </STEP-ProductInformation></pre>

Modified Object

For each template, the same single item (Recharge C) is modified and approved, which could trigger an OIEP.



Results

Without Filtering

```
<STEP-ProductInformation ExportTime="2015-12-17 15:13:24" ExportContext="Context1" ContextID="C

<Products>
  <Product ID="Product hierarchy root" UserTypeID="Product user-type root" Selected="false">
    <Name>Primary Product Hierarchy</Name>
    <Product ID="ProductsRoot" UserTypeID="Products" Selected="false">
      <Name>Products</Name>
      <Product ID="8302" UserTypeID="Level1" Selected="false">
        <Name>Electrical and Electronics</Name>
        <Product ID="20875" UserTypeID="Level2" Selected="false">
          <Name>Electronic Accessories</Name>
          <Product ID="20876" UserTypeID="Level3" Selected="false">
            <Name>Power</Name>
            <Product ID="20878" UserTypeID="Level4" Selected="false">
              <Name>Batteries</Name>
              <Product ID="123942" UserTypeID="ItemFolder" Selected="false">
                <Name>Batteries Rechargeable Items</Name>
                <Product ID="123963" UserTypeID="Level5">
                  <Name Changed="true">Recharge C</Name>
                  <ProductCrossReference ProductID="20882" Type="PrimaryDataSource"/>
                  <ProductCrossReference ProductID="123943" Type="CrossReference"/>
                  <ProductCrossReference ProductID="123944" Type="CrossReference"/>
                </Product>
              </Product>
            </Product>
          </Product>
        </Product>
      </Product>
    </Product>
  </Products>
</STEP-ProductInformation>
```

With Filtering

```
<STEP-ProductInformation ExportTime="2015-12-17 15:23:43" ExportContext="Context"

  <Products>
    <Product ID="20876" UserTypeID="Level3" ParentID="20875" Selected="false">
      <Name>Power</Name>
      <Product ID="20878" UserTypeID="Level4" Selected="false">
        <Name>Batteries</Name>
        <Product ID="123963" UserTypeID="Level5" ParentID="123942">
          <Name Changed="true">Recharge C</Name>
          <ProductCrossReference ProductID="20882" Type="PrimaryDataSource"/>
          <ProductCrossReference ProductID="123943" Type="CrossReference"/>
          <ProductCrossReference ProductID="123944" Type="CrossReference"/>
        </Product>
      </Product>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Filter References in STEPXML

When a downstream system requires a limited set of STEP references, use a reference tag with the **Type** XML attribute to indicate the required asset, classification, or product reference.

Use multiple instances of each reference tag to include all of the required objects.

Asset References

For example, to output asset references with an ID of 'PrimaryProductImage' and 'ProductImage', multiple **AssetCrossReference** tags are used with the **Type** XML attribute as follows:

```
<AssetCrossReference Type="PrimaryProductImage"/>
<AssetCrossReference Type="ProductImage"/>
```

Classification References

For example, to output only the classification reference with an ID of 'Supplier', the **ClassificationReference** tag is used with the **Type** XML attribute as follows:

```
<ClassificationReference Type="Supplier"/>
```

Product References

For example, to output only the product reference with an ID of 'CrossReference', the **ProductCrossReference** tag is used with the **Type** XML attribute as follows:

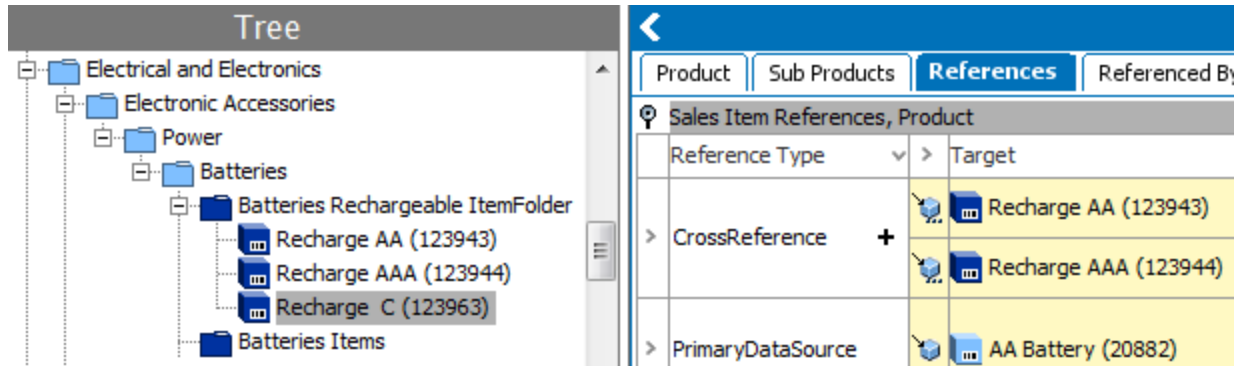
```
<ProductCrossReference Type="CrossReference"/>
```

Template

Without Filtering	With Filtering
<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <Product> <Name/> <ClassificationReference/> <ProductCrossReference/> </Product> </Products> </STEP-ProductInformation></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <Product> <Name/> <ClassificationReference/> <ProductCrossReference Type="CrossReference"/> </pre>

Modified Object

For each template, the same single item (Recharge C) is modified and approved, which can trigger the OIEP.



The screenshot shows the Stibo Systems interface. On the left, a 'Tree' view displays a hierarchy: 'Electrical and Electronics' > 'Electronic Accessories' > 'Power' > 'Batteries' > 'Batteries Rechargeable ItemFolder'. Under this folder, three items are listed: 'Recharge AA (123943)', 'Recharge AAA (123944)', and 'Recharge C (123963)'. The 'Recharge C (123963)' item is selected. On the right, the 'References' tab is active, showing a table titled 'Sales Item References, Product'. The table has two columns: 'Reference Type' and 'Target'. It lists three references: 'CrossReference' pointing to 'Recharge AA (123943)', 'CrossReference' pointing to 'Recharge AAA (123944)', and 'PrimaryDataSource' pointing to 'AA Battery (20882)'.

Results

Without filtering, all references for the exported product are output.

Without Filtering

```

<STEP-ProductInformation ExportTime="2015-12-16 16:23:02" ExportContext="Cxt"

  <Products>
    <Product ID="123963" UserTypeID="Item" ParentID="123942">
      <Name Changed="true">Recharge C</Name>
      <ClassificationReference ClassificationID="SupplyAll" Type="Supplier">
        <MetaData>
          <Value AttributeID="QtyOfItems" Derived="true">1</Value>
        </MetaData>
      </ClassificationReference>

      <ProductCrossReference ProductID="20882" Type="PrimaryDataSource"/>
      <ProductCrossReference ProductID="123943" Type="CrossReference"/>
      <ProductCrossReference ProductID="123944" Type="CrossReference"/>

    </Product>
  </Products>
</STEP-ProductInformation>
```

With filtering, only references of the specified type for the exported product are output.

With Filtering

```
<STEP-ProductInformation ExportTime="2015-12-16 16:23:02" ExportContext="Cxt"

  <Products>
    <Product ID="123963" UserTypeID="Item" ParentID="123942">
      <Name Changed="true">Recharge C</Name>
      <ClassificationReference ClassificationID="SupplyAll" Type="Supplier">
        <MetaData>
          <Value AttributeID="QtyOfItems" Derived="true">1</Value>
        </MetaData>
      </ClassificationReference>

      <ProductCrossReference ProductID="123943" Type="CrossReference"/>
      <ProductCrossReference ProductID="123944" Type="CrossReference"/>

    </Product>
  </Products>
</STEP-ProductInformation>
```

Filter Values in STEPXML

Use the following values tag to output all the attribute values:

```
<Values/>
```

Or specify them individually or by group using:

```
<Values>
  <Value AttributeID="Colour"/>
  <Value AttributeID="Size"/>
  <Value AttributeID="AllWebAttributes"/>
</Values>
```

To filter the attribute values, use the **AttributeID** XML attribute.

- When the output should include a specific attribute's value, include the attribute ID.
- When the output should include attributes within a particular attribute group, include the ID of the attribute group.

For an example of this messaging template, see the online version of this topic.

IncludeKey Tag in STEPXML

A unique key is a database item that represents a specific product in the STEP system. The item is composed of (transformed) product data. A key is a definition of how unique keys are composed. It specifies which object data is used and how it should be combined and transformed. The combination and transformation is done using the function framework (key formula). There can be exactly one unique key for each object for each defined key. No two objects can have the same unique key for an active key. For more information, see the **Unique Keys** section of the **System Setup / Super User Guide** documentation.

Unique keys are especially used when STEPXML is delivered to external systems like ERP systems, websites, and so on. These systems do not have access to the STEP ID of the exported products and therefore need a specific unique ID. For example, a SAP ID must be a 7-digit number.

In the following example, an advanced STEPXML template is defined to export the key ID based on the unique key 'SAP_ID'.

To export a key (with ID SAP_ID) through the Export Manager or the Event Messenger, you must include the XML tag `<IncludeKey KeyID="SAP_ID"/>` after the XML tag `<STEP-ProductInformation ....>`.

Template

```
<STEP-ProductInformation ExportDerivedAttrs="false"
FollowOverrideSubProducts="true">

<IncludeKey KeyID="SAP_ID"/>

<Products ExportSize="Minimum">
<Product>
    <Name/>
    <AttributeLink/>
    <ClassificationReference/>
    <ProductCrossReference/>
    <AssetCrossReference/>
    <Values/>
</Product>
</Products>
</STEP-ProductInformation>
```

Result

The template above instructs the exporter to also export keys of the corresponding IDs:

```
<STEP-ProductInformation
...
<DeleteProduct ID="77">
    <KeyValue KeyID="SAP_ID">SAP_77</KeyValue>
```

```

...
<Product ID="77" ...>
  <KeyValue KeyID="SAP_ID">SAP_77</KeyValue>
  ...
  <ProductCrossReference ProductID="77" ...>
    <KeyValue KeyID="SAP_ID">SAP_77</KeyValue>
    ...
  <Entity ID="77" ...>
    <KeyValue KeyID="SAP_ID">SAP_77</KeyValue>
    ...
  ...

```

Inherit References in STEPXML

When exporting, references within STEP are output against the object type where the references are defined. However, it is possible to inherit references to a specific product type when data is exported from STEP.

Inheriting references to all product types

To inherit references to all object types, use IncludeInherited.

For example:

```
<ClassificationReference IncludeInherited="true"/>
```

Inheriting references to a particular product type

To inherit references to a particular object type, the following tag must include the product object type that references should be inherited to. If the type is not added, then all reference types will be exported.

For example:

```
<AssetCrossReference Type="ProductImage" IncludeInheritedReferences="Level 3"/>
```

Input Data Example

The following sample data will be exported using the template below. The parent object includes the image references below:

Tree

Upper Body Wear

T-shirts

T-shirts Items

RoundNeck - T-shirts

Item 2

Item 3

Item 4

Item 5

Item 6

RoundNeck - T-shirts rev.0.3 - References

Product

Product Variants

Sub Products

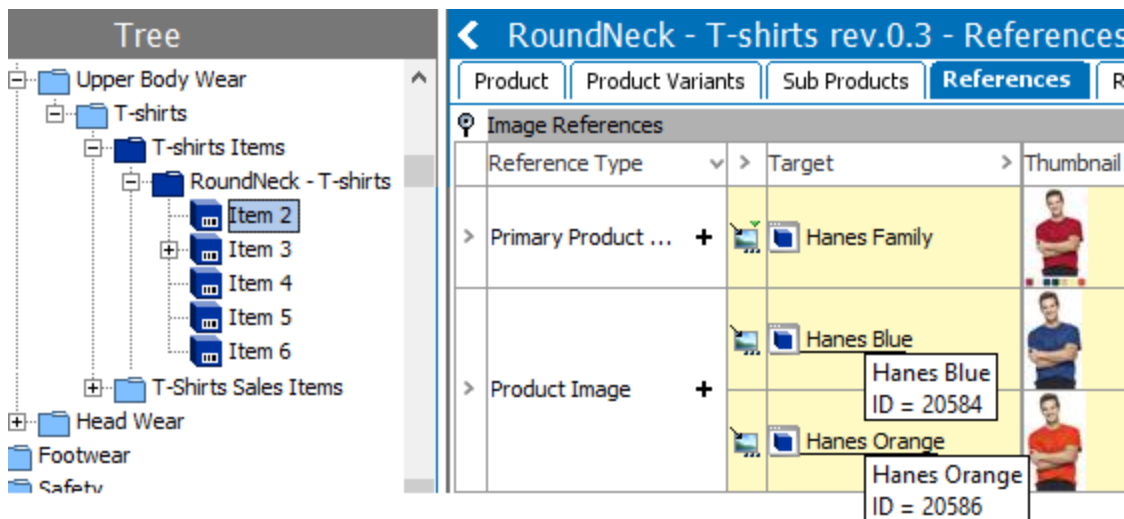
References

Reference

Image References

Reference Type	Target	Thumbnail
Primary Product ...	Hanes Family	
Product Image	Hanes Family	

The child object includes the following two local references (which override the inherited values from the parent):



Template

The boldface text includes two tags. One should be removed before mapping.

```
<?xml version="1.0" encoding="UTF-8"?>

<STEP-ProductInformation ExportDeletedData="true" ResolveInlineRefs="true"
ExportDerivedAttrs="true">

<Products>

    <Product IncludeParent="true">

        <Name/>

        <Values/>

        <Values IncludeInheritedAttributes="Level 3"/>

        <AssetCrossReference Type="ProductImage" IncludeInheritedReferences="Level 3"/>
or
        <AssetCrossReference IncludeInherited="true"/>
        <ProductCrossReference Type="Accessory"/>
        <ClassificationReference Type="Website"/>

    </Product>

</Products>

</STEP-ProductInformation>
```

Result

The following image shows an excerpt of the export using inheritance for only the ProductImage asset reference type using the template line: <AssetCrossReference Type="ProductImage" IncludeInheritedReferences="Level 3"/>.

```

42 ...
43 <Product ID="18203" UserTypeID="ItemFolder" Selected="false">
44   <Name>T-shirts Items</Name>
45   ...
46 <Product ID="124625" UserTypeID="ItemFamily">
47   <Name>RoundNeck - T-shirts</Name>
48   <AssetCrossReference AssetID="20585" Type="ProductImage"/>
49
50   ...
51 <Product ID="124626" UserTypeID="Item">
52   <Name>Item 2</Name>
53   <AssetCrossReference AssetID="20586" Type="ProductImage"/>
54   <AssetCrossReference AssetID="20584" Type="ProductImage"/>
55
56   ...

```

The following image shows an excerpt of the export using inheritance for the all reference types using the template line: <AssetCrossReference IncludeInherited="true"/>.

```

44 ...
45 <Product ID="18203" UserTypeID="ItemFolder" Selected="false">
46   <Name>T-shirts Items</Name>
47   ...
48 <Product ID="124625" UserTypeID="ItemFamily">
49   <Name>RoundNeck - T-shirts</Name>
50   <AssetCrossReference AssetID="20585" Type="ProductImage"/>
51   <AssetCrossReference AssetID="20585" Type="PrimaryProductImage"/>
52
53   ...
54 <Product ID="124626" UserTypeID="Item">
55   <Name>Item 2</Name>
56   <AssetCrossReference AssetID="20586" Type="ProductImage"/>
57   <AssetCrossReference AssetID="20584" Type="ProductImage"/>
58   <AssetCrossReference AssetID="6900" Type="OwnersManual"/>
59   <AssetCrossReference AssetID="20700" Type="ADA"/>
60
61   ...

```

Inherit Values in STEPXML

The Values tag is valid within the Assets, Classifications, and Products tags and by default outputs only local values. When no local value is available, even when an inherited value exists, no value is output.

The IncludeInherited and IncludeInheritedAttributes XML attributes allow inherited values to output to top-level objects or to specified object types.

IncludeInherited

Local values are included for all objects in the output. To avoid duplicate values throughout the file, inherited values are only included once, and display for the top-level object type being exported.

```
<Values IncludeInherited="true"/>
```

IncludeInheritedAttributes

Inherits attribute values down to objects of the specified object type. Use multiple instances of this tag when duplicate values are required for inherited attributes.

```
<Values IncludeInheritedAttributes="ItemFolder"/>
```

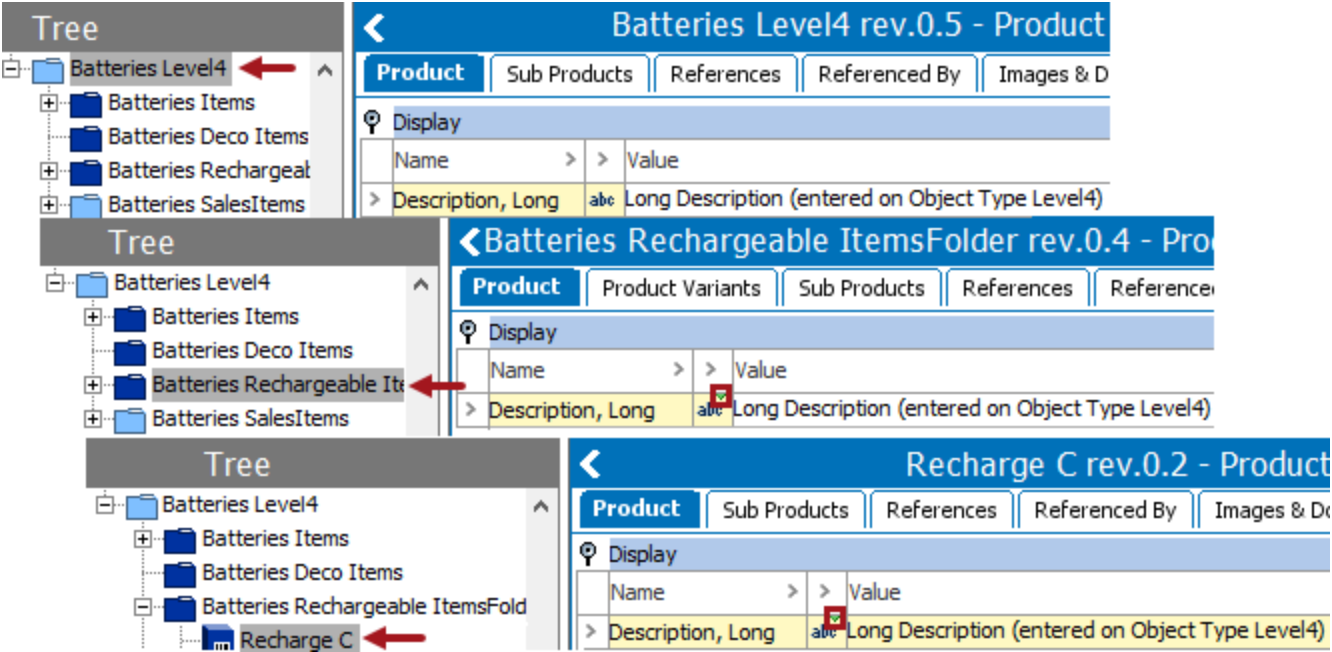
```
<Values IncludeInheritedAttributes="Item"/>
```

Note: IncludeInheritedAttributes will show attribute values on object types where they are not valid. For this reason, they cannot be imported back into STEP. The import process fails with errors saying each attribute is not valid for the object.

Input Data Example

With the IncludeInherited (set to true) or the IncludeInheritedAttributes XML attributes in the Advanced STEPXML Template parameter, the following data is exported:

- The 'Batteries Level4' object has a type of 'Level 4' and has a local value on the 'Description, Long' attribute.
- The 'Batteries Rechargeable ItemsFolder' object has a type of 'ItemFolder' and an inherited value (as indicated by the green down arrow) on the 'Description, Long' attribute.
- The 'Recharge C' object has a type of 'Item' and an inherited value (as indicated by the green down arrow) on the 'Description, Long' attribute.



Template

With the IncludeInherited XML attribute (set to true) or the IncludeInheritedAttributes XML attribute in the Advanced STEPXML Template parameter, the above data is exported:

IncludeInherited	IncludeInheritedAttributes
<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products ExportSize="Minimum"> <Product> <Name/> <Values IncludeInherited="true"/> </Product> </Products> </STEP-ProductInformation></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products ExportSize="Minimum"> <Product> <Name/> <Values IncludeInheritedAttributes="ItemFolder"/> <Values IncludeInheritedAttributes="Item"/> </Product> </Products> </STEP-ProductInformation></pre>

Note: The XML attribute ExportSize="Minimum" is required on the Products tag for IncludeInheritedAttributes to get attributes to be inherited to the particular object type.

Results

Two (2) levels are exported from the product hierarchy: ItemFolder and Item.

- Local and inherited values are output for the top level (ItemFolder)
- Only local values are displayed for the lower level (Item)

IncludeInherited

```

<STEP-ProductInformation ExportTime="2015-12-23 13:52:56" ExportContext="Context1" ContextID="Context1" W
  <Products>
    <Product ID="123942" UserTypeID="ItemFolder" ParentID="20878">
      <Name Changed="true">Batteries Rechargeable ItemFolder</Name>
      <Values>
        <Value AttributeID="PrimaryColor">Green (entered on Object Type ItemFolder)</Value>
        <Value AttributeID="DescriptionLong">Long Description (entered on Object Type Level4)</Value>
      </Values>
    </Product>
    <Product ID="123963" UserTypeID="Item">
      <Name Changed="true">Recharge C</Name>
      <Values>
        <Value AttributeID="ProductGroupDescription">Description (entered on Object Type Item)</Value>
      </Values>
    </Product>
  </Products>
</STEP-ProductInformation>

```

Three (3) levels are exported from the product hierarchy: Level4, ItemFolder, and Item.

- Only local values are output for the top level (Level4)
- Inherited values are displayed for both of the lower levels (ItemFolder and Item)

IncludeInheritedAttributes

```
<STEP-ProductInformation ExportTime="2015-12-23 13:24:52" ExportContext="Context1" ContextID="Context1" Wo

<Products>
  <Product ID="20878" UserTypeID="Level4" ParentID="20876">
    <Name Changed="true">Batteries Level4</Name>
    <Values>
      <Value AttributeID="DescriptionLong">Long Description (entered on Object Type Level4)</Value>
    </Values>
  </Product>
  <Product ID="123942" UserTypeID="ItemFolder">
    <Name Changed="true">Batteries Rechargeable ItemFolder</Name>
    <Values>
      <Value AttributeID="PrimaryColor">Green (entered on Object Type ItemFolder)</Value>
      <Value AttributeID="DescriptionLong">Long Description (entered on Object Type Level4)</Value>
    </Values>
  </Product>
  <Product ID="123963" UserTypeID="Item">
    <Name Changed="true">Recharge C</Name>
    <Values>
      <Value AttributeID="PrimaryColor">Green (entered on Object Type ItemFolder)</Value>
      <Value AttributeID="DescriptionLong">Long Description (entered on Object Type Level4)</Value>
      <Value AttributeID="ProductGroupDescription">Description (entered on Object Type Item)</Value>
    </Values>
  </Product>
</Products>
</STEP-ProductInformation>
```

Minimum, Referenced, and Selected in STEPXML

Many of the parameters for STEPXML format on the Select Format step of the wizard include the options Minimum, Referenced, and/or Selected. The following illustrates the difference in the output of these options.

Select Format

STEPXML

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

Include Attributes

Minimum

Include Attribute Groups

Selected

Include Attribute Transformations

No

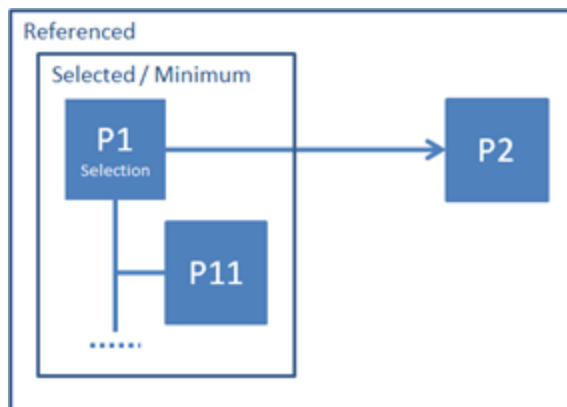
Include Bulk Update Configurations

No

Include Business Rules (Global) and Libraries

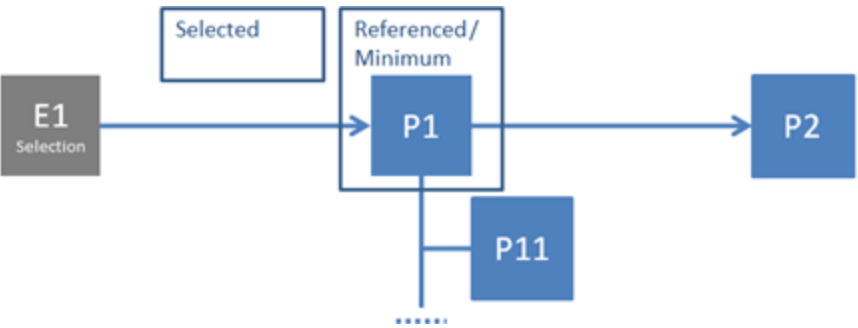
Referenced

In the example below, product P1 has been selected as root node for the export. The Selected and Minimum options cause P1 and its descendants to be exported. The Referenced setting will also include P2 since it is referenced from P1.

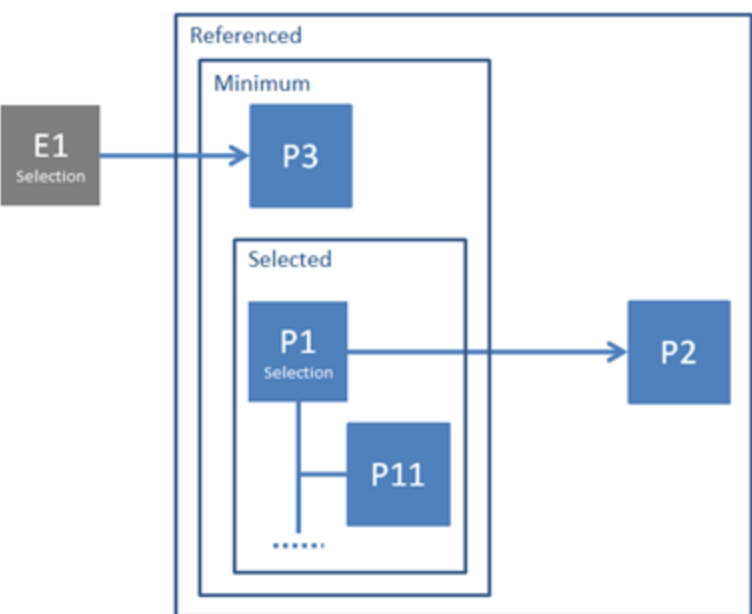


When exporting a product hierarchy, selecting Minimum outputs only the objects used in the selected product hierarchy.

In the following example, the root node selected for export is a non-product (E1) that references a product (P1). The Selected setting (for 'Include Products') will not cause any products to be exported. The Referenced and Minimum will instead have the same meaning, and both cause P1 to be exported, as shown below.

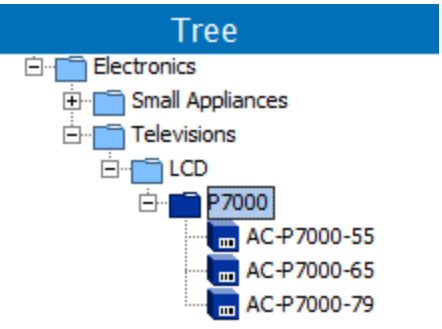


In the following example, the reason for having three options is demonstrated. Both a non-product (E1) and a product (P1) are selected as root nodes for the export. With this selection, Referenced, Minimum, and Selected will have three different meanings.



Hierarchy Export Including Objects Referenced Example

The following example exports the hierarchy shown below. In addition to data for these objects, the export also includes the product, classification, and asset objects referenced from these objects.



Product P7000 is selected as the root and the export uses the following settings:

- **Include Assets: Minimum** – The minimum set of asset objects related to the initial selection (the referenced assets).
- **Include Classifications: Minimum** – The minimum set of classification objects related to the initial selection (the linked classifications and, as a special feature for classifications, all ancestors up to the top level classification node).
- **Include Products: Referenced** – The selected product, its descendants, and referenced products.
- **Include Product Attribute Values: All** – Local values (inherited values are not exported by default).

By default, products and classifications will be exported in two nested structures. To get a flat structure (no product elements inside product elements and no classification elements inside classification elements), set the Data Objects > **Flatten Hierarchies** parameter to yes.

The image below shows the exported XML, stripped of values, names, and references. Notice the following elements of the export file:

1. Linked classifications
2. Referenced assets
3. Referenced products
4. Export selection

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main" UseContextLocale="false">
  <Classifications>
    <Classification ID="Classification 1 root" UserTypeID="Classification 1 user-type root" Selected="false">
      <Classification ID="Websites" UserTypeID="WebHierarchyRoot" Selected="false">
        <Classification ID="I-WebsiteRoot" UserTypeID="WebsiteRoot" Selected="false">
          <Classification ID="I-WebLevel1-2" UserTypeID="WebLevel1" Selected="false">
            <Classification ID="I-WebLevel2-21" UserTypeID="WebLevel2" Selected="false" Referenced="true"/>
            <Classification ID="I-WebLevel2-22" UserTypeID="WebLevel2" Selected="false" Referenced="true"/>
            <Classification ID="I-WebLevel2-24" UserTypeID="WebLevel2" Selected="false" Referenced="true"/>
          </Classification>
        </Classification>
      </Classification>
    </Classification>
  </Classifications>
  <Assets>
    <Asset ID="ProductImage-5" UserTypeID="ProductImage" Selected="false" Referenced="true"/>
    <Asset ID="Manual-7" UserTypeID="Manual" Selected="false" Referenced="true"/>
    <Asset ID="Manual-8" UserTypeID="Manual" Selected="false" Referenced="true"/>
    <Asset ID="ProductImage-6" UserTypeID="ProductImage" Selected="false" Referenced="true"/>
  </Assets>
  <Products>
    <Product ID="I-SalesItem-1111" UserTypeID="SalesItem" ParentID="I-Level3-111" Selected="false" Referenced="true"/>
    <Product ID="I-SalesItem-1121" UserTypeID="SalesItem" ParentID="I-Level3-112" Selected="false" Referenced="true"/>
    <Product ID="I-SalesItemFamily-1311" UserTypeID="SalesItemFamily" ParentID="I-Level3-131"/>
      <Name>P7000</Name>
      <ProductCrossReference ProductID="I-SalesItem-1111" Type="Accessory"/>
      <ProductCrossReference ProductID="I-SalesItem-1121" Type="Accessory"/>
      <AssetCrossReference AssetID="Manual-8" Type="ManualEN"/>
      <AssetCrossReference AssetID="Manual-7" Type="ManualDE"/>
      <AssetCrossReference AssetID="ProductImage-6" Type="SituationImage"/>
      <AssetCrossReference AssetID="ProductImage-5" Type="PrimaryProductImage"/>
      <Product ID="I-SalesItem-13113" UserTypeID="SalesItem"/>
      <Product ID="I-SalesItem-13112" UserTypeID="SalesItem"/>
      <Product ID="I-SalesItem-13111" UserTypeID="SalesItem"/>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Product Export Including Definitions, LOVs, and Units for Attributes Example

As another example, to export a single product from the same structure above, and also export the definitions for all attributes used by the product and all LOVs, and units used by these attributes.

A single product is selected as the root, and the export uses the following settings:

- **Configuration > Include Units: Minimum** – The Units used by the Attributes being exported.
- **Configuration > Include List of Values: Minimum** – The LOVs used by the exported Attributes.

- **Configuration > Include Attributes: Minimum** – The set of Attributes used by the objects being exported.
- **Data Objects > Include Products: Selected** – Only the selected product.
- **Data Objects > Include Product Attribute Values: Yes**

Below is a modified version of the results. All but one unit, one list of values group, one list of values and one attribute has been removed. Also, the exported product has again been stripped of name, references / links, and values.

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main"
UseContextLocale="false">
<UnitList>
  <Unit ID="unece.unit.CMT" Referenced="true">
    <Name>cm</Name>
    <UnitConversion BaseUnitID="unece.unit.MTR" Factor="0.01" Offset="0"/>
  </Unit>
</UnitList>
<ListsOfValues>
  <ListOfValue ID="ScreenSizes" ParentID="List Of Values group root"
    AllowUserValueAddition="false" UseValueID="true" Referenced="true">
    <Name>ScreenSizes</Name>
    <Validation BaseType="text" MinValue="" MaxValue="" MaxLength=""
    InputMask=""/>
    <Value ID="55">55'' (139 cm)</Value>
    <Value ID="65">65'' (164 cm)</Value>
    <Value ID="79">79'' (200 cm)</Value>
  </ListOfValue>
</ListsOfValues>
<AttributeList>
  <Attribute ID="ScreenSize" MultiValued="false" ProductMode="Normal"
    FullTextIndexed="false" ExternallyMaintained="true" Derived="false"
    HierarchicalFiltering="false" ClassificationHierarchicalFiltering="false"
    Referenced="true">
    <Name>Screen Size</Name>
    <ListOfValueLink ListOfValueID="ScreenSizes"/>
    <AttributeGroupLink AttributeGroupID="CategorySpecificAttributes"/>
    <UserTypeLink UserTypeID="SalesItem"/>
  </Attribute>
</AttributeList>
```



```
</Attribute>
</AttributeList>
<Products>
  <Product ID="I-SalesItem-13111" UserTypeID="SalesItem" ParentID="I-
SalesItemFamily-1311"/>
</Products>
</STEP-ProductInformation>
```

Multi-Context Data in STEPXML

A single STEPXML file can include data from multiple contexts, for example, data tied to specific dimensions points or combinations of dimension points from different dimensions. This 'cross-context' STEPXML will include a 'Qualifiers' section that maps the 'QualifierID' XML attribute value for dimension dependent elements to dimension points or combinations of dimension points. Cross-context STEPXML can be imported and exported.

Input Data Example

The following example shows data for a single product being exported and includes:

- Two contexts: 'US-eng' (Context1) and 'DE-de' (Context3)
- Two country dimension points: 'US' and 'DE'
- Two language dimension points: 'eng' and 'de'
- Three language-dependent attributes: 'LongAdvertisingCopy', 'ConsumerShortDescription', and 'Color'
- One country-dependent attribute: 'Warranty'
- Two attributes that are not dimension dependent: 'Weight' and 'EAN'

Within the output tool, set the **Export Data for Selected Contexts** option as discussed in the **STEPXML Outbound Parameters** topic.

Result

For the STEPXML output, see the online version of this topic.

Considerations for Qualifier IDs

Notice that the context element nested in the qualifier element is not required for imports. It is present to let downstream systems map values to specific contexts. The logic for this mapping is that a context uses the value for the qualifier where it has the lowest 'priority' number.

The qualifiers might not always be as straight-forward as in the example above, where you have one qualifier per dimension point. Although it is not recommended, dimension dependent data can depend on more than one dimension at a time. In that case, a 'pseudo qualifier' is displayed and references one dimension point per dependent dimension.

Several factors may complicate the reading of qualifier IDs:

- Dimensional values are inherited from more general dimension points.
- When adding multiple dimension dependencies, you get values for specific combinations of contexts, and the multi-context export result gets complicated to read.
- XML elements are repeated in their nested elements, and new XML elements are introduced. This means that while still adhering to the STEPXML XSD, the result is a superset of the typical single-context export format.
- A single qualifier will map to multiple contexts with different priorities, and as discussed above you may have several possible values with different priorities.

Since it can be very difficult to interpret multi-context export files, the recommended method for exporting multiple contexts is to use the Context Splitter post processor available in an OIEP. For more information, see the **OIEP - Post-Processor - Context Splitter** topic in the **Outbound Integration Endpoints** documentation.

Products Tag in STEPXML

The Products tag can include multiple options. When an option is absent, the default setting is used.

```
<Products ExportSize="Minimum">
  <Product IncludeParent="true">
    </Product>
  </Products>
```

The following XML attributes are available:

ExportSize

- Minimum - default setting. Other options are Selected and All.

For details about export size, see the **Minimum, Referenced, and Selected in STEPXML** section.

IncludeParent

- False - default setting
- True - when a product is exported from STEP, the file includes the approved product as well as its parents.

The Product tag can include multiple additional tags, each is described below the following sample STEPXML. When an option is absent, the data is excluded.

```
<?xml version="1.0" encoding="UTF-8"?>
<STEP-ProductInformation>
  <Products>
    <Product>
      <Name/>
      <Values IncludeInherited="true"/>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Additional STEPXML Tags

The Product tag can include multiple additional internal tags, each is described below the following sample STEPXML. When an option is absent, the default setting is used.

Name

The <Name/> tag returns the object name as output.

Values

The <Values/> tag returns the attribute values as output. For more information about values output, see the **Inherit Values in STEPXML** section, the **Delete Values in STEPXML** and the **Filter Values in STEPXML** section.

AssetCrossReference

The <AssetCrossReference/> tag returns all asset cross references as output. For more information about cross reference output, see the **Filter References in STEPXML** section.

ProductCrossReference

The <ProductCrossReference/> tag returns all product cross references as output. For more information about cross reference output, see the **Filter References in STEPXML** section.

ClassificationReference

The <ClassificationReference/> tag returns all classification cross references as output. For more information about cross reference output, see the **Filter References in STEPXML** section.

Additionally, see the following topics for more information on modifying product data:

- Auto Approve in STEPXML
- Filter Products in STEPXML
- Referenced and Embedded XML Attributes in STEPXML
- SequenceProduct Tag in STEPXML

Product Nesting Example

In the following STEPXML file, product data reflects the Tree hierarchy structure. SalesItem-13112 and SalesItem-13113 are children of SalesItemFamily-1311. It can be represented in both a nested and non-nested structure.

Nested structure

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="GL" WorkspaceID="Main">
  <Products>
    <Product ID="SalesItemFamily-1311" UserTypeID="SalesItemFamily"
ParentID="Level3-131">
      <Product ID="SalesItem-13112" UserTypeID="SalesItem"/>
      <Product ID="SalesItem-13113" UserTypeID="SalesItem"/>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Non-nested structure

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="GL" WorkspaceID="Main">
  <Products>
    <Product ID="SalesItemFamily-1311" UserTypeID="SalesItemFamily"
ParentID="Level3-131"/>
    <Product ID="SalesItem-13112" UserTypeID="SalesItem"
ParentID="SalesItemFamily-1311"/>
    <Product ID="SalesItem-13113" UserTypeID="SalesItem"
ParentID="SalesItemFamily-1311"/>
  </Products>
</STEP-ProductInformation>
```

Product with Units, References, and Multi-Values Attributes Example

The following STEPXML file being imported shows a single product of the object type SalesItem (UserTypeID in the XML). The XML mirrors the STEP model so that all data owned by the product is represented in the product element.

Notice that units, references, and multi-valued attributes are represented. Also, the data in the product element does not include information about attribute, unit, and reference target names.

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main">
  <Products>
    <Product ID="I-SalesItem-1111" UserTypeID="SalesItem" ParentID="I-Level3-
111">
      <Name>AC-YU500A</Name>
      <ClassificationReference ClassificationID="I-WebLevel2-23"
Type="WebsiteLink"/>
      <AssetCrossReference AssetID="ProductImage-2" Type="PrimaryProductImage"/>
      <AssetCrossReference AssetID="Manual-6" Type="ManualEN"/>
      <Values>
        <MultiValue AttributeID="DisplayLanguage">
          <Value>English</Value>
          <Value>German</Value>
          <Value>Spanish</Value>
        </MultiValue>
        <Value AttributeID="LongAdvertisingCopy">Active 3D Glasses</Value>
        <Value AttributeID="Color" ID="BK">Black</Value>
```

```

<Value AttributeID="Weight" UnitID="unece.unit.KGM">0.036</Value>
<Value AttributeID="EAN">4905524977623</Value>
</Values>
</Product>
</Products>
</STEP-ProductInformation>

```

Import Manager

Steps

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

Select Format

Format: **STEPXML**

Converter for the STEP Product Information XML format.

Validate: **no**

Conversion Preview:

<ID>	<Parent ID>	<Type ID>	<Name>	DisplayLanguage
I-SalesItem-1111	I-Level3-111	SalesItem	AC-YU500A	English German Spanish

Back Next Finish Cancel

Product to Classification Links Owned by the Classification

By default the product to classification links owned by the classification are exported in two places:

1. Under the Product as a reference to the classification.
2. Under the Classification as a reference to the product. For an XML example of the classification links, see the online version of this topic.

The reason the links are exported under the products is historical, and it has not been changed due to backwards compatibility. However, logically they should be exported under the classification, as they are owned by the classification. And when the links are modified, they affect the revision of the classification - not the product. It is possible to stop the classification owned links from being exported under the products by using the recorder option 'IncludeClassificationOwned' on the 'Products' element. By setting this to "false" the classification links will not be exported under the products:

```

<STEP-ProductInformation>
  <Products IncludeClassificationOwned="false">
    <Product>
      </Product>
    </Products>
  </STEP-ProductInformation>

```

</STEP-ProductInformation>

For information on IncludeClassificationOwned, see the **Owns Product Links on Alternate Classifications Object Type** topic in the **System Setup / Super User Guide** documentation.

Referenced and Embedded XML Attributes in STEPXML

The ID and Type of an object's references can be exported using the **Referenced** XML attribute.

```
<Product Referenced="true">
```

In order to export details about the reference itself, use the **Embedded** XML attribute.

```
<Entity Referenced="true" Embedded="true">
```

Note: Embedded referenced objects cannot be imported. Any changes to the embedded object, like its name, will not update the data in STEP if loaded for import.

When required by a downstream system, the **Referenced** and **Embedded** XML attributes are used together to provide additional details about the reference in line with the product, classification, or asset that owns the reference.

For example, if STEP allows a product-to-product cross reference between products A and B, then the data for both products A and B can be delivered via the product reference. The cross referenced product details are exported within the product that references it.

Template

The `MetaData` tag is used to output attribute values maintained on the product reference itself.

Referenced Without Embedded	Referenced With Embedded
<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation ResolveInlineRefs="true"> <Products> <FilterUserType ID="Item"/> <Product IncludeParent="true"> <Name/> <ProductCrossReference Type="CrossReference"> <MetaData/> <Product Referenced="true"> <Name/> <Values/> </Product></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation ResolveInlineRefs="true"> <Products> <FilterUserType ID="Item"/> <Product IncludeParent="true"> <Name/> <ProductCrossReference Type="CrossReference"> <MetaData/> <Product Referenced="true" Embedded="true"> <Name/> <Values/></pre>

Referenced Without Embedded	Referenced With Embedded
</ProductCrossReference> </Product> </Products> </STEP-ProductInformation>	</Product> </ProductCrossReference> </Product> </Products> </STEP-ProductInformation>

Modified Object

For each template, the same single item (Recharge C) is modified and approved, which triggers the OIEP.

Tree

Electrical and Electronics

Electronic Accessories

Power

Batteries

Batteries Rechargeable ItemFolder

Recharge AA (123943)

Recharge AAA (123944)

Recharge C (123963)

Batteries Items

<

Product

Sub Products

References

Referenced By

Sales Item References, Product

Reference Type		Target
> CrossReference	+	Recharge AA (123943)
		Recharge AAA (123944)
> PrimaryDataSource		AA Battery (20882)

Results

Referenced Without Embedded

```
<STEP-ProductInformation ExportTime="2015-12-21 15:14:13" ExportContext="Context1" ContextID="Co:

  <Products>
    <Product ID="123944" UserTypeID="Item" ParentID="123942" Selected="false" Referenced="true">
      <Name>Recharge AAA</Name>
      <Values>
        <Value AttributeID="SupplierPartNumber">66165165</Value>
        <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accesso:
        <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
      </Values>
    </Product>
    <Product ID="123943" UserTypeID="Item" ParentID="123942" Selected="false" Referenced="true">
      <Name>Recharge AA</Name>
      <Values>
        <Value AttributeID="SupplierPartNumber">32153563</Value>
        <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accesso:
        <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
      </Values>
    </Product>
    <Product ID="123963" UserTypeID="Item" ParentID="123942">
      <Name Changed="true">Recharge C</Name>
      <ProductCrossReference ProductID="123943" Type="CrossReference"/>
      <ProductCrossReference ProductID="123944" Type="CrossReference"/>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Referenced With Embedded

```
<STEP-ProductInformation ExportTime="2015-12-21 15:11:51" ExportContext="Context1" ContextID="Co:

<Products>
  <Product ID="123944" UserTypeID="Item" ParentID="123942" Selected="false" Referenced="true">
    <Name>Recharge AAA</Name>
    <Values>
      <Value AttributeID="SupplierPartNumber">65165165</Value>
      <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accesso
      <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
    </Values>
  </Product>
  <Product ID="123943" UserTypeID="Item" ParentID="123942" Selected="false" Referenced="true">
    <Name>Recharge AA</Name>
    <Values>
      <Value AttributeID="SupplierPartNumber">32153563</Value>
      <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accesso
      <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
    </Values>
  </Product>
  <Product ID="123963" UserTypeID="Item" ParentID="123942">
    <Name Changed="true">Recharge C</Name>
    <ProductCrossReference ProductID="123943" Type="CrossReference">
<Product ID="123943" UserTypeID="Item" ParentID="123942">
  <Name>Recharge AA</Name>
  <Values>
    <Value AttributeID="SupplierPartNumber">32153563</Value>
    <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accesso
    <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
  </Values>
</Product></ProductCrossReference>
  <ProductCrossReference ProductID="123944" Type="CrossReference">
<Product ID="123944" UserTypeID="Item" ParentID="123942">
  <Name>Recharge AAA</Name>
  <Values>
    <Value AttributeID="SupplierPartNumber">65165165</Value>
    <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accesso
    <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
  </Values>
</Product></ProductCrossReference>

</Product>
</Products>
</STEP-ProductInformation>
```

As another example, the following Advanced STEPXML template is defined to export a customer entity hierarchy. Each customer has references to contacts, and each contact has one or more references to addresses. All objects are in this example are modeled with entity objects.

 Customer X
  Customer to Contact
 John
  Contact to Address
 Jane's Work Address

 ~~Customer X~~
 ~~Customer to Contact~~
 Peter
  Contact to Address
 John's Work Address

The following is a general Advanced STEPXML template that exports all used entity objects below the customer hierarchy.

```
<?xml version='1.0'?>
<STEP-ProductInformation>
<Entities>
  <Entity>
    <EntityCrossReference Type="Customer to Contact">
      <Entity Referenced="true">
        <Name />
      <EntityCrossReference Type="Contact to Address">
        <Entity Referenced="true">
          <Name />
        </Entity>
      </EntityCrossReference>
    </Entity>
  </EntityCrossReference>
</Entities>
</STEP-ProductInformation>
```

The template exports customer entity objects and their references of the type **Customer to Contact** and **Contact to Address**. However, the referenced objects are not exported as embedded in the customer hierarchy. To export the referenced contacts and their address objects embedded in the customer hierarchy, you have to insert `<Entity Referenced="true" Embedded="true">` in the advanced template as illustrated in the following.

```
<?xml version='1.0'?>
<STEP-ProductInformation>
<Entities>
  <Entity>
    <EntityCrossReference Type="Customer to Contact">
      <Entity Referenced="true" Embedded="true">
        <Name />
      <EntityCrossReference Type="Contact to Address">
        <Entity Referenced="true" Embedded="true">
          <Name />
        </Entity>
      </EntityCrossReference>
    </Entity>
  </EntityCrossReference>
</Entities>
```

```
</EntityCrossReference>  
</Entity>  
</Entities>  
</STEP-ProductInformation>
```

ReplacementRules Tag in STEPXML

Removing references and links via inbound STEPXML involves two steps:

1. In the **STEP-ProductInformation** tag, use the applicable entry setting (listed below in the 'STEP-ProductInformation Tag Replace Instructions' section) to indicate that existing references or links should be replaced.
2. Specify the reference or link type using the **ReplacementRules** section.

STEP-ProductInformation Tag

The replace instruction indicates the type of data to be replaced.

The following STEPXML removes all existing product-to-attribute links (using ReplaceAttributeLinks="Y" in the STEP-ProductInformation tag) from the product with ID 'UNSPSC000.52161505', so only the links in the file are present in STEP after the import.

Additionally, for the same product, the following STEPXML removes all existing asset references of the type 'Guides' via the <ReplaceProductToAssetCrossReference ReferenceTypeID="Guides"/> in the ReplacementRules section. After the import, only the references in the file remain in STEP.

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main"
  ReplaceAttributeLinks="Y">
  <ReplacementRules>
    <ReplaceProductToAssetCrossReference ReferenceTypeID="Guides"/>
  </ReplacementRules>
  <Products>
    <Product ID="UNSPSC000.52161505">
      <Name>Televisions</Name>
      <AssetCrossReference AssetID="Mounting Guide" Type="Guides"/>
      <AttributeLink AttributeID="Aspect Ratio"/>
      <AttributeLink AttributeID="Diagonal Screen Size"/>
    </Product>
  </Products>
</STEP-ProductInformation>
```

The following instructions can be used to remove existing values that are not in the import file:

- ReplaceAttributeLinks
- ReplaceClassificationToClassificationCrossReferences
- ReplaceClassificationValues

- ReplaceOverrideToSubProductReferences
- ReplaceProductToClassificationReferences
- ReplaceProductToProductCrossReferences
- ReplaceProductValues
- ReplaceTerms
- ReplaceValidUserTypesForAttributes

For more information, see **STEP-ProductInformation Tag in STEPXML**.

ReplacementRules Tag

While you can delete references using the ReplacementRules tag, use caution as this removes **all** references of the indicated type for **all** loaded objects.

To limit the deletions performed, insert a **ReplacementRules** tag within the STEPXML file to delete only the references of a specified type.

For example, the following STEPXML deletes only the 'ProductToSupersededBy' references:

```
<ReplacementRules>
  <ReplaceProductToProductCrossReference
    ReferenceTypeID="ProductToSupersededBy"/>
</ReplacementRules>
```

The following instructions are available:

- ReplaceClassificationReferenceType
- ReplaceCrossReferenceType
- ReplaceProductToAssetCrossReferenceType
- ReplaceProductToProductCrossReferenceType
- ReplaceProductValuesForAttributeGroupType

Note: The DeleteProductCrossReference tag is only available for outbound XML when using event-based integration endpoints as discussed in **Delete Product Reference in STEPXML**.

SequenceProduct Tag in STEPXML

The sequence number is the order of the products in STEP. When used with the **IncludeParent** tag, the **SequenceProduct** tag exports the sequence numbers of all exported products.

```
<SequenceProduct/>
```

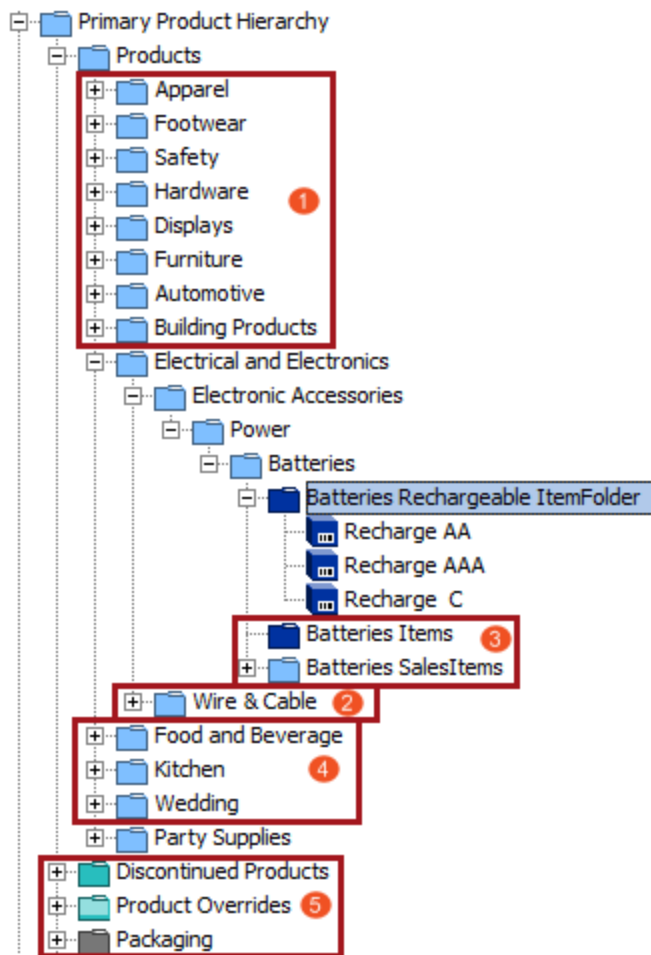
Classifications are always exported in the sequence they are displayed in the workbench.

Template

With SequenceProduct	Without SequenceProduct
<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <Product IncludeParent="true"> <SequenceProduct/> <Name/> </Product> </Products> </STEP-ProductInformation></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <Product IncludeParent="true"> <Name/> </Product> </Products> </STEP-ProductInformation></pre>

Modified Object

For each template, the same single item (Batteries Rechargeable ItemFolder) is modified and approved, which triggers the OIEP.



Results

All parents for the exported product are output.

Sequence numbers for all nodes under each of the parents are output.

Sequence numbers for nodes equal to the changed object are output.

With SequenceProduct

```
<STEP-ProductInformation ExportTime="2016-01-04 10:04:58" ExportContext="Context1" ContextID="
<Products>
  <Product ID="Product hierarchy root" UserTypeID="Product user-type root" Selected="false">
    <Name>Primary Product Hierarchy</Name>
    <Product ID="ProductsRoot" UserTypeID="Products" Selected="false">
      <Name>Products</Name>
      <SequenceProduct ID="18200"/>
      <SequenceProduct ID="20688"/>
      <SequenceProduct ID="20718"/>
      <SequenceProduct ID="20864"/>
      <SequenceProduct ID="22005"/> ①
      <SequenceProduct ID="22150"/>
      <SequenceProduct ID="8298"/>
      <SequenceProduct ID="8299"/>
      <Product ID="8302" UserTypeID="Level1" Selected="false">
        <Name>Electrical and Electronics</Name>
        <SequenceProduct ID="8313"/> ②
        <Product ID="20875" UserTypeID="Level2" Selected="false">
          <Name>Electronic Accessories</Name>
          <Product ID="20876" UserTypeID="Level3" Selected="false">
            <Name>Power</Name>
            <Product ID="20878" UserTypeID="Level4" Selected="false">
              <Name>Batteries</Name>
              <SequenceProduct ID="20879"/>
              <SequenceProduct ID="20880"/> ③
              <Product ID="123942" UserTypeID="ItemFolder">
                <Name Changed="true">Batteries Rechargeable ItemFolder</Name>
              </Product>
            </Product>
          </Product>
        </Product>
      </Product>
      <SequenceProduct ID="8303"/>
      <SequenceProduct ID="8304"/> ④
      <SequenceProduct ID="109946"/>
    </Product>
    <SequenceProduct ID="DiscontinuedProductsRoot"/>
    <SequenceProduct ID="ProductOverridesRoot"/> ⑤
    <SequenceProduct ID="PackagingRoot"/>
  </Product>
</Products>
</STEP-ProductInformation>
```

All parents for the exported product are output.

Without SequenceProduct

```
<STEP-ProductInformation ExportTime="2016-01-04 10:13:31" ExportContext="Context1" ContextID="(
  <Products>
    <Product ID="Product hierarchy root" UserTypeID="Product user-type root" Selected="false">
      <Name>Primary Product Hierarchy</Name>
      <Product ID="ProductsRoot" UserTypeID="Products" Selected="false">
        <Name>Products</Name>
        <Product ID="8302" UserTypeID="Level1" Selected="false">
          <Name>Electrical and Electronics</Name>
          <Product ID="20875" UserTypeID="Level2" Selected="false">
            <Name>Electronic Accessories</Name>
            <Product ID="20876" UserTypeID="Level3" Selected="false">
              <Name>Power</Name>
              <Product ID="20878" UserTypeID="Level4" Selected="false">
                <Name>Batteries</Name>
                <Product ID="123942" UserTypeID="ItemFolder">
                  <Name Changed="true">Batteries Rechargeable ItemFolder</Name>
                </Product>
              </Product>
            </Product>
          </Product>
        </Product>
      </Product>
    </Products>
  </STEP-ProductInformation>
```

SingleUpdateMode in STEPXML

Importing certain changes requires that the STEP system enters single-update mode. This ensures that during import, only the import process is allowed to modify data in the database. When the import process enters single-update mode, no user can write any data; although viewing data is still allowed. For more information, see **Single-Update Mode**.

The following import changes require single-update mode:

- Adding a dimension to attributes, object types, reference types, units, LOVs, etc.
- Changing attributes from being internally maintained to being externally maintained
- Changing validation base type settings on attributes or term object types.

Note: If the import process cannot enter single-update mode because users are working in STEP, the import process goes into a wait state until single-update mode is possible.

Insert **SingleUpdateMode="Y"** in the **STEP-ProductInformation** tag to cause the import process to enter single-update mode, as illustrated in the following example:

```
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main"  
SingleUpdateMode="Y">
```

For more information, see **STEP-ProductInformation Tag in STEPXML**.

STEP-ProductInformation Tag in STEPXML

The **STEP-ProductInformation** tag determines the data exported for Products. It allows multiple XML attributes (described below) to be listed on a single line, each separated by a single space. XML attributes that are absent use the default setting.

```
<STEP-ProductInformation ExportDeletedData="true" ResolveInlineRefs="true"
FollowOverrideSubProducts="true" ExportDerivedAttrs="true" Validation="XSD"
DefinitionsAsComments="true" EventsAsComments="false">
```

Note: The true and false settings for all XML attributes are case-sensitive and must be lower case as shown in the above example.

DefinitionsAsComments

- False - default setting
- True - configuration definitions for Business Rules, Web UIs, IEPs, or Workflows are exported as comments in the output

For more details, see the **Export Configuration Definitions as Comments** section of the **Configuration Management** documentation.

EventsAsComments

- True - default setting
- False - events are not exported as comments in the output

The following image shows the comments section of a STEPXML export that included events:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
<properties>
<entry key="eventID">1795966</entry>
<entry key="eventType">1</entry>
<entry key="nodeID">124629</entry>
<entry key="nodeType">product</entry>
<entry key="eventCreationTime">2017-12-27 01:49:50.0</entry>
<entry key="eventRevision">0.5</entry>
<entry key="messageCreationTime">2017-12-27 01:50:40.194</entry>
</properties>
-->
```

ExportDeletedData

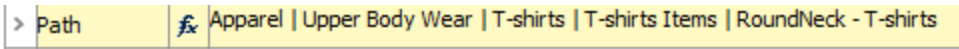
- False - default setting
- True - change markers indicate if data has been added to or removed from a product. This includes attribute values, product references and classification references.

For more information, see the **Event-Based OIEP Triggered by Deleting Products, Classifications, and Assets** section of the **Delete Objects in STEPXML** topic.

ExportDerivedAttrs

- False - default setting
- True - calculated attribute values are derived when an object is output from STEP

With the ExportDerivedAttrs XML attribute set to true in the Advanced STEPXML Template parameter, the following calculated attribute is exported:



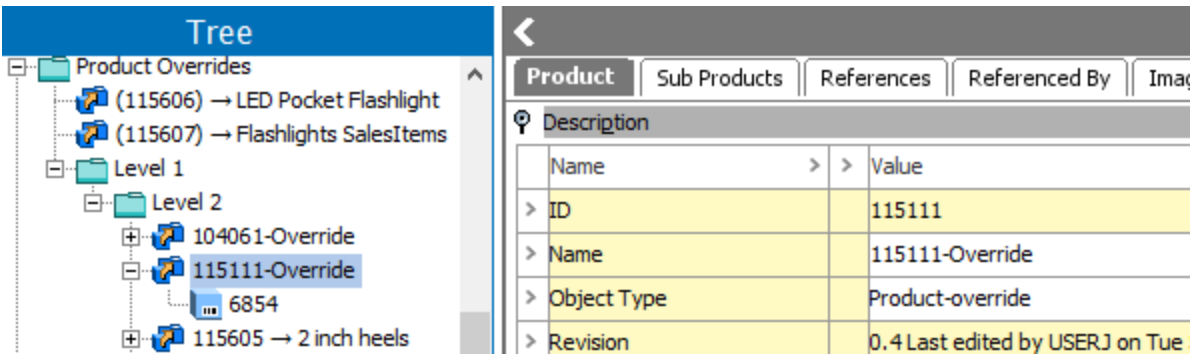
An excerpt from the output generated includes this data:

```
106 <Value AttributeID="Path" Derived="true">Apparel | Upper Body Wear | T-shirts | T-shirts Items | RoundNeck - T-shirts</Value>
```

FollowOverrideSubProducts

- False - default setting
- True - all Sub-Products of Product Overrides are included in the output XML

With the FollowOverrideSubProducts XML attribute set to true in the Advanced STEPXML Template parameter, the following product override is exported:



An excerpt from the output generated includes this data:

```
52 ...
53 <Product ID="115111" UserTypeID="std.AdaptorType" OverridesProductID="" ParentID="101608">
54   <Name>115111-Override</Name>
55   <Values>
58   </Values>
59   <OverrideSubProduct ProductID="6854"/>
60 ...
```

ResolveInlineRefs

- False - default setting
- True - all inline references are resolved within the attributes values that are exported

With the ResolveInlineRefs XML attribute set to true in the Advanced STEPXML Template parameter, the following attribute with inline references is exported:

Name	>	>	Value
> Long Item Description	abc		One of our best-selling T-shirts, available in Red or Black.

An excerpt from the output generated includes this data:

```

25 <Values>
26   ...
27   <Value AttributeID="LongItemDescription">One of our best-selling T-shirts, available in Red or Black.</Value>
28   ...
29 </Values>

```

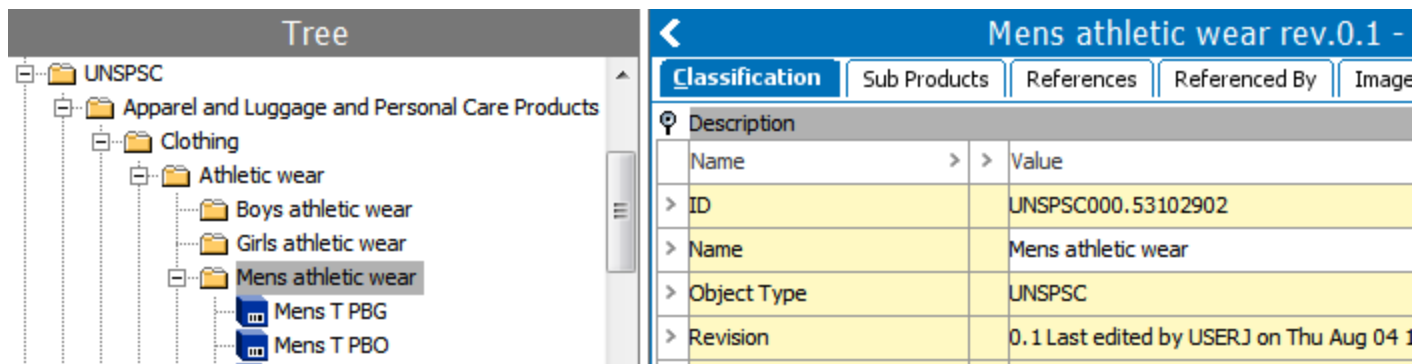
Validation

- None - default. Other option is are Schema and XSD

UNSPSC Format

United Nations Standard Products and Services Code® (UNSPSC) is a global classification framework for products and services in all industry sectors. Importing a UNSPSC file into STEP creates a set of multi-level classification folders, depending on the contents of the import file. For detailed information about UNSPSC, search the web.

When a UNSPSC classification exists in STEP, you can make product / service identification easier by using Bulk Update to add a 'product-to-classification link type' reference between your products / services and the UNSPSC classification system. This displays your products / services within the UNSPSC folder as shown in the image below. For more information, see the **Bulk Updates** documentation.



Format Availability

UNSPSC is available for selection in:

- IIEP - see **Creating an Inbound Integration Endpoint**
- Import Manager - see **Creating a Data Import**

Mapping

This format does not require a data map.

Inbound Data

All data included in the file selected for import will be imported; selection of particular classifications is not available.

The content and required preparation of a UNSPSC file depends on the UNSPSC version, as described below:

Version	File Types Included in ZIP	File Prep Required	File to Import
13	One (1) Excel file with six (6) tabs	Extract before import	Excel file
17, 18, 19, 20	Six (6) individual .TXT files	None	.ZIP file

Multiple versions of UNSPSC classifications can exist simultaneously in STEP, provided different UNSPSC classification ID prefixes are used.

Prerequisite

To accommodate importing multiple levels within a single classification, the UNSPSC classification object type must exist and have itself as a parent, as defined in the following steps.

1. In System Setup, create the UNSPSC alternate classification object type.
2. On the alternate classification editor Reference tab, click the **Add Parent** link and select the UNSPSC object. The hierarchy becomes an endless chain where UNSPSC is repeated for every level.

The screenshot shows the 'System Setup' interface on the left and the 'UNSPSC - References' tab on the right. In the 'System Setup' tree, 'UNSPSC' is listed under 'Alternate Classifications' and is highlighted with a red box. In the 'UNSPSC - References' tab, the 'References' sub-tab is active, showing a table of parents. The table has columns 'ID' and 'Name'. A row shows 'UNSPSC' as the parent of 'UNSPSC', which is also highlighted with a red box. Below the table is a blue 'Add Parent' link.

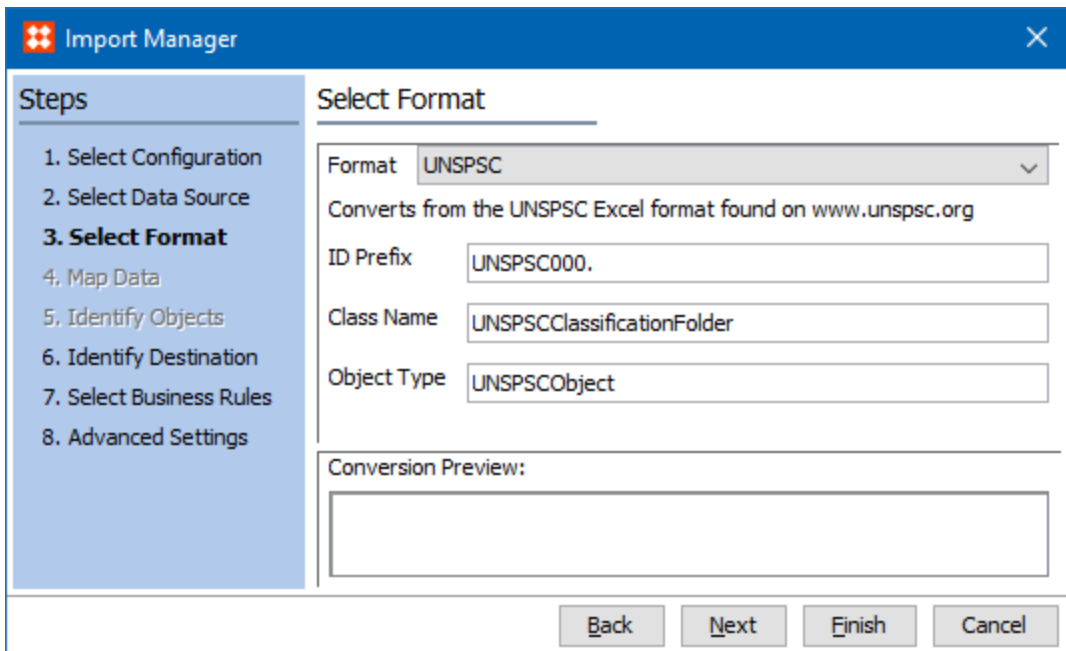
To import multiple UNSPSC languages for different contexts, on the top-level UNSPSC folder, set the Dimension Dependencies parameter to Language.

The screenshot shows the 'System Setup' interface on the left and the 'UNSPSC - Object Type' tab on the right. In the 'System Setup' tree, 'UNSPSC' is listed under 'Alternate Classifications' and is highlighted with a red box. In the 'UNSPSC - Object Type' tab, the 'Object Type' sub-tab is active, showing a table of properties. The table has columns 'Name' and 'Value'. A row shows 'Dimension Dependencies' with the value 'Language;', which is highlighted with a red box. Below the table is a blue 'Add Parent' link.

Inbound Parameters

- **ID prefix** is the text you want to appear preceding the classification 'code' from the UNSPSC file. The 'code' is imported as a STEP classification ID (see the image above). 'UNSPSC000.' is the default text prefixed to the ID.
- **Class Name** field, enter a name for the root node of the classification. A new classification folder with this name will be created on the tree. The default setting is UNSPSC.
- **Object Type** field, enter the object type ID of your UNSPSC object. This object type must already exist in System Setup. The default setting is UNSPSC. Update this field to match your object type ID if it is not 'UNSPSC.'
- **Conversion Preview** information is not available for UNSPSC format.

Import Manager



The screenshot shows the 'Import Manager' dialog box with a blue header and a close button. On the left is a 'Steps' sidebar with eight items: '1. Select Configuration', '2. Select Data Source', '3. Select Format' (highlighted in blue), '4. Map Data', '5. Identify Objects', '6. Identify Destination', '7. Select Business Rules', and '8. Advanced Settings'. The main area is titled 'Select Format' and contains a 'Format' dropdown menu set to 'UNSPSC'. Below this is a note: 'Converts from the UNSPSC Excel format found on www.unspsc.org'. There are three text input fields: 'ID Prefix' with the value 'UNSPSC000.', 'Class Name' with the value 'UNSPSCClassificationFolder', and 'Object Type' with the value 'UNSPSCObject'. Below these is a 'Conversion Preview' section with an empty text area. At the bottom are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

IIEP

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint

2. Choose Receiver

3. Configure Endpoint

4. Configure PreProcessor

5. Configure Processing Engine

5.1. Select Sample File

5.2. Select Data Source

5.3. Select Format

5.4. Map Data

5.5. Identify Objects

5.6. Identify Destination

5.7. Select Business Rules

5.8. Advanced Settings

6. Configure PostProcessor

7. Schedule Endpoint

8. Configure Error Reporter

Configure Processing Engine : Select Format

Format

UNSPSC

Converts from the UNSPSC Excel format found on www.unspsc.org

ID Prefix

UNSPSC000.

Class Name

UNSPSCClassificationFolder

Object Type

UNSPSCObject

Conversion Preview:

Back

Next

Finish

Cancel

xCBL Format

The xCBL format exports data using the CommerceOne XML-based language. For more information, search the web.

Format Availability

xCBL is available for selection in:

- Export Manager - see **Creating a Data Export**
- OIEP - see **Creating an Outbound Integration Endpoint**

Mapping

This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For details, see **Data Mapping**.

Outbound Data

The same parameters are available in both Export Manager and OIEP.

Export Manager

Export Manager

×

Steps

1. Select Configuration

2. Select Objects

3. **Select Format**

4. Map Data

5. Advanced

6. Select Delivery Method

Select Format

xCBL

▼

Exports data in a CommerceOne xCBL 4.0 format.

Catalog ID

Catalog Pretty Name

Catalog Date

Catalog Version

Short description

Valid From

Valid Until

Default Language

en

Default Currency Code

Schema name

Schema description

Schema version

Schema standard

Back

Next

Finish

Cancel

OIEP

Background Processes

Statistics

Error Log Excerpts

Log

Status

Outbound Integration Endpoint

Configuration

Event Triggering Definitions

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor	Post-Processor
> Item (Modify)	xCBL (4 mappings) ...	None	None
> Add configuration			

Select format

Format

Mapping

Advanced

xCBL

Exports data in a CommerceOne xCBL 4.0 format.

Catalog ID

Catalog Pretty Name

Catalog Date

Catalog Version

Short description

Valid From

Valid Until

Default Language

Default Currency Code

Schema name

Schema description

Schema version

Schema standard

en

OK

Cancel

Data Mapping

The data format selected determines if data mapping is required to sync data from external systems with STEP data. The mapping functionality also includes transformations, which allow the data to be modified.

Mapping functions are comparable for inbound (Import Manager and IIEPs) and outbound (Export Manager and OIEPs) tools, however there are two user interfaces.

Inbound Data Mapping

Data mapping for import allows you to determine where the incoming data belongs in STEP. For information about the available inbound options, see **Inbound Map Data Options**.

Outbound Data Mapping

For export, data mapping identifies the location of STEP data being output. For information about the available outbound options, see **Outbound Map Data Options**.

Inbound Map Data Options

The mapping step for the Import Manager and IIEP wizards include the same user interface.

For more information about the additional import steps, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Formats

Mapping differs slightly based on the type of data being imported.

Tabular Formats

The following tabular formats require mapping for inbound data:

- CSV Format
- Excel Format
- FixedWidth Format

XML Formats

The following XML formats require mapping for inbound data:

- BMEcat Format
- BMEcat2005 Format
- Generic XML Format
- IDoc MATMAS 05 Format

As defined below, in addition to mapping data, you can also select the object type being imported, modify data before import, and verify the quality of the inbound data.

Once mapped, source columns are shown with an yellow background. The result section displays a green check for a data column that has been validated for the object type.

Map Data

Source:

<Name>	>	<Parent ID>	>	Primary Color	>	Secondary Color	>
Mens T PBO		18209		Black		Orange	
Mens T PBG		18209		Blue		Green	

Result:

Map to: Product

Name = <Name> ✓	Parent = <Parent ID> ✓	PrimaryColor = Primar... ✓	SecondaryColor = Se... ✓
Mens T PBO	Cotton T-Shirts	Black	Orange
Mens T PBG	Cotton T-Shirts	Blue	Green

Auto Map

Map

Constant

Remove

Transform

Generate Profile

For mapping examples using common column data, see the **Inbound Map Data Examples**.

Map To Object Type

The 'Map to' field determines the object type of the data being imported. For more information, see the **Inbound Map Data - Map To** section.

Mapping Data

Mapping can be accomplished using one of these three ways:

1. Auto Map - click the **Auto Map** button, STEP reads the header row of the import file, and relates all matches to corresponding objects in STEP. For more information, see the **Inbound Map Data - Auto Map** section.
2. Map - manually select an import column and use the **Map** button to relate a STEP object. For more information, see the **Inbound Map Data - Map** section.
3. Auto Map and Map manually - after auto mapping an import file with a header row, manually update or correct the results.

If the Result panel includes a column if you do not want to import, click the **Remove** button to eliminate the mapping. For more information, see the **Inbound Map Data - Remove** section.

For additional guidance, see the **Tips for Mapping Data** below.

Modifying Data

The following options are available to modify data being imported into STEP:

- **Constant** allows you to add an attribute value to all records. Common setup uses this feature for filling some object attributes. For more information, see the **Inbound Map Data - Constant** section.

- **Transform** provides numerous ways to modify data before it is loaded into STEP, such as split one data column into multiple attribute values, append and prepend data to the values, perform search and replace, and so on. For more information, see the **Inbound Map Data - Transform** section.

Profiling Data

The **Generate Profile** data quality option enables you to locate errors in the data you want to import. The functionality applies to row and column based import formats such as Excel, CSV, and FixedWidth. For more information, see the **Inbound Map Data - Generate Profile** section.

Tips for Mapping Data

Keep the following information in mind when mapping import files:

- Not every column of the input file has to be mapped. For example, if there are five (5) pieces of data to load, but the import file has 20 columns, only map the five (5) relevant columns.
- Mapped columns must correspond to a selected STEP object. Columns that are not mapped are not loaded into STEP.
- Each row (item) of an input file results in one object created or updated. STEP attempts to import every row of the file. Ensure your file contains only the objects that should be created or updated in STEP and remove any unwanted rows before importing.
- Every node and level within the Primary Product Hierarchy is regarded as being a STEP product. The term 'product' includes product families, subcategories, minor categories, major categories, and so on.
- Import file columns can be in any order. Auto Map allows you to automatically match data based on the name or ID of an existing attribute, or you can Map to map columns manually.
- Units are not required in the import file for values for STEP attributes with a default unit. For example, when loading values for the attribute 'Maximum Operating Temperature' with the default unit of °F, there is no need to include the °F in the load file if they are all specified in Fahrenheit. The system imports the value and assigns the °F automatically.
- All mapped columns must be mapped against only one object type (product, asset, classification, entity, or attribute). If the file contains a more than one object type (for example products and entities), you must import the same file several times, each time choosing a single object type. This procedure is often used when building a new product hierarchy.

Inbound Map Data - Auto Map

Auto Map compares data in the input file's header row with the IDs / names of the attributes in STEP. When the inbound tool finds a match, the column is automatically mapped to the corresponding object attributes. The Auto map feature only works when the input file has a header row. You can add a header row if one does not exist, otherwise, you can manually map the data.

Map Data

Source:

<Name>	>	<Parent ID>	>	Primary Color	>	Secondary Color	>
Mens T PBO		18209		Black		Orange	
Mens T PBG		18209		Blue		Green	

Result:

Map to: Product

Name = <Name> ✓	Parent = <Parent ID> ✓	PrimaryColor = Primar... ✓	SecondaryColor = Se... ✓
Mens T PBO	Cotton T-Shirts	Black	Orange
Mens T PBG	Cotton T-Shirts	Blue	Green

Auto Map

Map

Constant

Remove

Transform

Generate Profile

For more information about the additional import steps, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Predefined Header Text

Auto Map recognizes the following **case-sensitive** header text to indicate these special columns:

Header Text	Description
<ID>	STEP ID of object
<ID KeyID=...>	ID of unique key for object For example, <ID KeyID=SupplierPartNumber> will automap to the unique key with the ID 'SupplierPartNumber'.
<Name>	STEP name of object
<Object Type ID>	STEP object type ID Note: The default selection for exporting object type is 'Object Type Name', however the

Header Text	Description
	importer expects the 'Object Type ID' . To generate an export file that is prepared for reimport, either manually map the object type, or use the Transformation Aspect to map object type ID in the exported file. For more information, see the Aspect - Transform Outbound section of the Data Mapping documentation.
<Parent ID>	STEP ID of object's parent
<Parent KeyID=..>	ID of unique key for object's parent For example, <Parent KeyID=Key3> will automap to the unique key with the ID 'Key3'.

The import process attempts to auto map all other columns in the input file using Name or ID of existing attributes. Both successful and unsuccessful mappings are shown in the message dialog. If an auto mapping is unsuccessful, you can map that column manually as defined in the **Inbound Map Data - Map** topic.

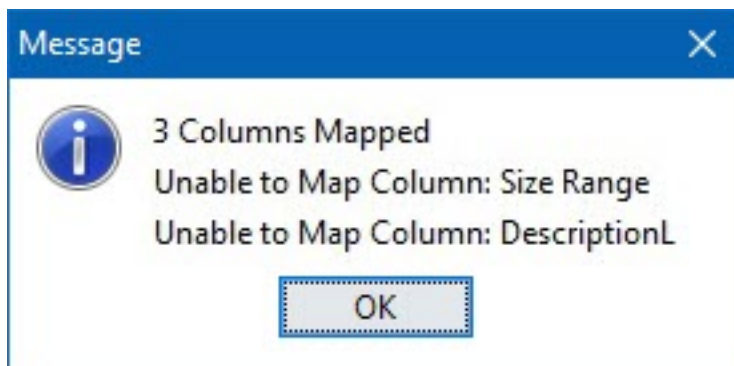
Auto Map Example

The file being imported includes following columns:

Source:

<Name>	>	<Parent ID>	>	Primary Color	>	Size Range	>	DescriptionL	>
Mens T PBO		18209		Black		XS - XXXL		Comfortable fitting 100% cotton tee.	
Mens T PBG		18209		Blue		S - XL		Comfortable fitting 100% cotton tee.	
Mens T PGS		18209		Green		XS - XXXL		Comfortable fitting tee of cotton/poly blend.	
Mens T PGW		18209		Gray		S - XL		Comfortable fitting tee of cotton/poly blend.	
Mens T POY		18209		Orange		S - XL		Comfortable fitting 100% cotton tee.	

- In the inbound Map Data step, click the **Auto Map** button. After matching header text to the attributes in STEP, a message displays with the results. The 'Unable to Map Column' error indicates that no attribute exists in STEP with the name. Click **OK**.



Important: If your system has multiple attributes with the same name, use caution when auto mapping. STEP will find a match, but it may not be the attribute you expected. STEP will match exactly if the input file contains the Attribute ID in the header.

2. The Map Data page is updated to show:

- Successfully mapped columns are displayed with a yellow background in the Source panel and a green check mark in the Result panel.
- Columns that are not mapped have a white background in the Source panel.

Source:

<Name>	>	<Parent ID>	>	Primary Color	>	Size Range	>	DescriptionL	>
Mens T PBO		18209		Black		XS - XXXL		Comfortable fitting 100% cotton tee.	
Mens T PBG		18209		Blue		S - XL		Comfortable fitting 100% cotton tee.	
Mens T PGS		18209		Green		XS - XXXL		Comfortable fitting tee of cotton/poly blend.	
Mens T PGW		18209		Gray		S - XL		Comfortable fitting tee of cotton/poly blend.	
Mens T POY		18209		Orange		S - XL		Comfortable fitting 100% cotton tee.	

Result:

Map to: Product

Name=<Name> ✓	Parent=<Parent ID> ✓	PrimaryColor=Primary Color ✓
Mens T PBO	Cotton T-Shirts	Black
Mens T PBG	Cotton T-Shirts	Blue
Mens T PGS	Cotton T-Shirts	Green
Mens T PGW	Cotton T-Shirts	Gray
Mens T POY	Cotton T-Shirts	Orange

3. Review the **Result** panel. Make any adjustments or transformations if needed.

Note: If your system has multiple attributes with the same name, use caution when auto mapping. STEP will find a match, but it may not be the attribute you expected.

4. When the mapping is correct, click **Next**.

Inbound Map Data - Constant

You can add an attribute value to all records by mapping a constant value. Common setup uses this feature for filling some object attributes.

Map Data

Source:

<Name>	>	<Parent ID>	>	Primary Color	>	Secondary Color	>
Mens T PBO		18209		Black		Orange	
Mens T PBG		18209		Blue		Green	

Result:

Map to: Product

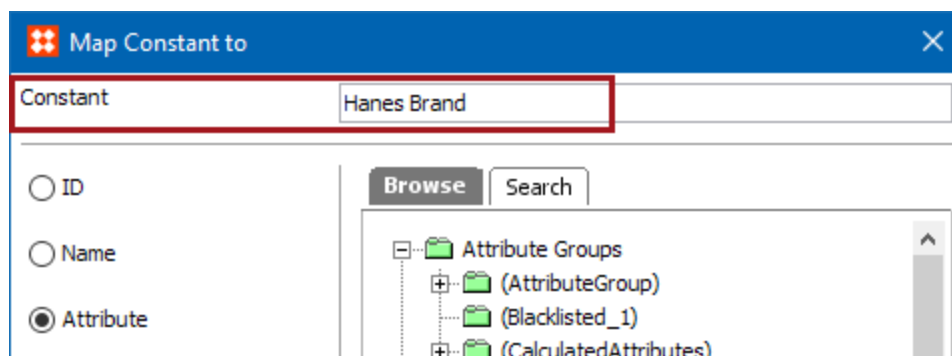
Name=<Name> ✓	Parent=<Parent ID> ✓	PrimaryColor=Primar... ✓	SecondaryColor=Se... ✓
Mens T PBO	Cotton T-Shirts	Black	Orange
Mens T PBG	Cotton T-Shirts	Blue	Green

Auto Map Map **Constant** Remove Transform Generate Profile

For more information about the additional import steps, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Mapping a Constant

- Below the Result panel, click the **Constant** button.
- On the **Map Constant To** dialog, in the **Constant** field, enter text that should be assigned to all records.



The dialog box titled "Map Constant to" has a close button (X) in the top right corner. It features a "Constant" field with a red border containing the text "Hanes Brand". Below this field are three radio buttons: "ID", "Name", and "Attribute", with "Attribute" selected. To the right of the radio buttons are "Browse" and "Search" buttons. Below these buttons is a tree view showing a hierarchy of folders: "Attribute Groups", "(AttributeGroup)", "(Blacklisted_1)", and "(CalculatedAttributes)".

- Select the **Attribute** radio button and use **Browse** or **Search** to select the attribute to be modified. Click **OK**.

Map Constant to

Constant

Hanes Brand

☐ ID

☐ Name

☒ Attribute

☐ Product Classification Link

☐ Product Reference

Browse

Search

Manufacturer Name (ManufacturerName)

Search

Advanced

Name

> **Manufacturer Name** ID = ManufacturerName

4. The **Result** panel displays the constant value as if it exists in the input file.

Result:

Map to: Product

ID=<ID> ✓	Name=<Name> ✓	Parent=<Parent ID> ✓	PrimaryColor=(PrimaryColor) ✓	ManufacturerName=Hanes Brand ✓
18210	18210 M B	Cotton T-shirts	Blue	Hanes Brand
18212	18212 L B	Cotton T-shirts	Red	Hanes Brand
18213	18213 M O	Cotton T-shirts	Black	Hanes Brand
18216	18216 L O	Cotton T-shirts	Green	Hanes Brand
100703	12-GGK79	T-shirts Items	Red	Hanes Brand
112456		T-shirts Items	Orange	Hanes Brand
113202		Polo T-shirt	Green	Hanes Brand

Once a constant value has been mapped to an attribute, for the first 200 rows of the import file, the data is validated against the data model. If the data is not applicable for an attribute due to mismatch with the configured validation will be highlighted as below.

Result:

Map to: Product

Date=Target
20695
20695
20695
20695

Value "20695" not valid for attribute "Date"

A user can also add the constant value to the LOV attribute, they must enter the value of the LOV, which is case sensitive.

If the LOV attribute holds an LOV with value IDs, then the user can enter the value ID. If the option 'Match value ID in LOV' is checked, the IDs are matched against the IDs in the LOV, and values are applied based on the match, as shown below.

The screenshot shows the 'Map Constant to' dialog box. The 'Constant' field at the top contains the text 'red'. On the left, a list of radio buttons includes 'ID', 'Name', 'Attribute' (which is selected), 'Product Classification Link', 'Product Reference', 'Asset Reference', 'Classification Reference', 'Entity Reference', 'Reference Meta-Data', 'Object Type', 'Variable', 'Multivalued Variable', 'Overrides product', and 'Add child to override'. To the right of the 'Attribute' selection is a search panel with 'Browse' and 'Search' buttons. Below these is a search input field containing 'Color (Color)' and a 'Search' button. Underneath is an 'Advanced' section with a tree view showing 'Name' and a selected item '> Color ID = Color'. At the bottom of the dialog, there is a 'Create' button, a 'Mandatory' checkbox (unchecked), and a 'Match value ID in LOV' checkbox (checked). 'Cancel' and 'OK' buttons are at the bottom right.

The Result panel displays the mapped values below, and matches the LOV ID.

Import Manager

X

Steps

1. Select Configuration

2. Select Data Source

3. Select Format

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Map Data

Source:

<ID>	>	<Name>	>	<Parent ID>	>	Primary Color	>
181951		18216 L O		18209		Blue	

< >

Result:

Map to: Product

ID = <ID> ✓	Name = <Name> ✓	Parent = <Parent ID> ✓	PrimaryColor = blue (match value id)
181951	18216 L O	Cotton T-Shirts	blue

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Back

Next

Finish

Cancel

Note: The LOV ID is case sensitive, if there is a mismatch with the LOV ID that exists in STEP, validation will return an error as shown below.

PrimaryColor=pink (match value id)

pink

Value "pink" not found in LOV "(Color)" of attribute "Primary Color"

Inbound Map Data - Generate Profile

The import functionality provides a data quality option that enables you to locate errors in the data you want to import. The functionality applies to row and column based import formats such as Excel, CSV, and FixedWidth.

Map Data

Source:

<Name>	>	<Parent ID>	>	Primary Color	>	Secondary Color	>
Mens T PBO		18209		Black		Orange	
Mens T PBG		18209		Blue		Green	

Result:

Map to: Product

Name=<Name> ✓	Parent=<Parent ID> ✓	PrimaryColor=Primar... ✓	SecondaryColor=Se... ✓
Mens T PBO	Cotton T-Shirts	Black	Orange
Mens T PBG	Cotton T-Shirts	Blue	Green

For more information about the additional import steps, see [Creating a Data Import](#) or [Creating an Inbound Integration Endpoint](#).

Profiling Setup

To limit the number of profiled rows and prevent out of memory exceptions, use the property **Import.Profile.Max.Products** in the sharedconfig.properties file. The default value for the property is 10,000. If the number of rows in the import file exceeds the limit, the remaining rows are not profiled.

Profiling Data

1. Map all required data on the Map Data page and click the Generate Profile button to display the Import Data profile.

Map Data

Source:

<ID>	> <Name>	> <ShortItemDescription[DK]>	> <ShortItemDescription[UK]>	> <Unique Key>
114852	Blue Light	Lightweight and durable	Durable, lightweight flashlight	AJJKF22
114854	Green Flashlight	Carbon Case	Durable Carbon case	AJJKF23
111204	LED Flashlight	200 voltz	120 voltz	668998643
114859	Pink light	20 volt light pink flashlight	Light pink flashlight	AAJJKF24

Result:

Map to: Product

ID=<ID> ✓	Name=<Name> ✓	ShortItemDescription=<ShortItemDescription[DK]> ✓	ShortItemDescription=<ShortItemDescription[UK]> ✓	ID KeyID=Supplier
114852	Blue Light	Lightweight and durable	Durable, lightweight flashlight	AJJKF22
114854	Green Flashlight	Carbon Case	Durable Carbon case	AJJKF23
111204	LED Flashlight	200 voltz	120 voltz	668998643

Auto Map

Map

Constant

Remove

Transform

Generate Profile

2. The Import Data Profile page includes an Overview and a Details view page.

3. Select the **Overview** radio button to display the following three widgets:

- **Object Count:** displays the number of profiled objects
- **Most Complete:** displays the 10 most complete columns
- **Least Complete:** displays the 10 least complete columns

Import Data Profile

☒ Overview
☐ Details

Object Count

6

Objects

Most Complete Column

Column	Completeness
> <ID>	100%
> <Name>	83.33%
> <ShortItemDescription[DK]>	83.33%
> <ShortItemDescription[UK]>	83.33%

Least Complete Column

Column	Completeness
> <Name>	83.33%
> <ShortItemDescription[DK]>	83.33%
> <ShortItemDescription[UK]>	83.33%
> <Unique Key>	83.33%

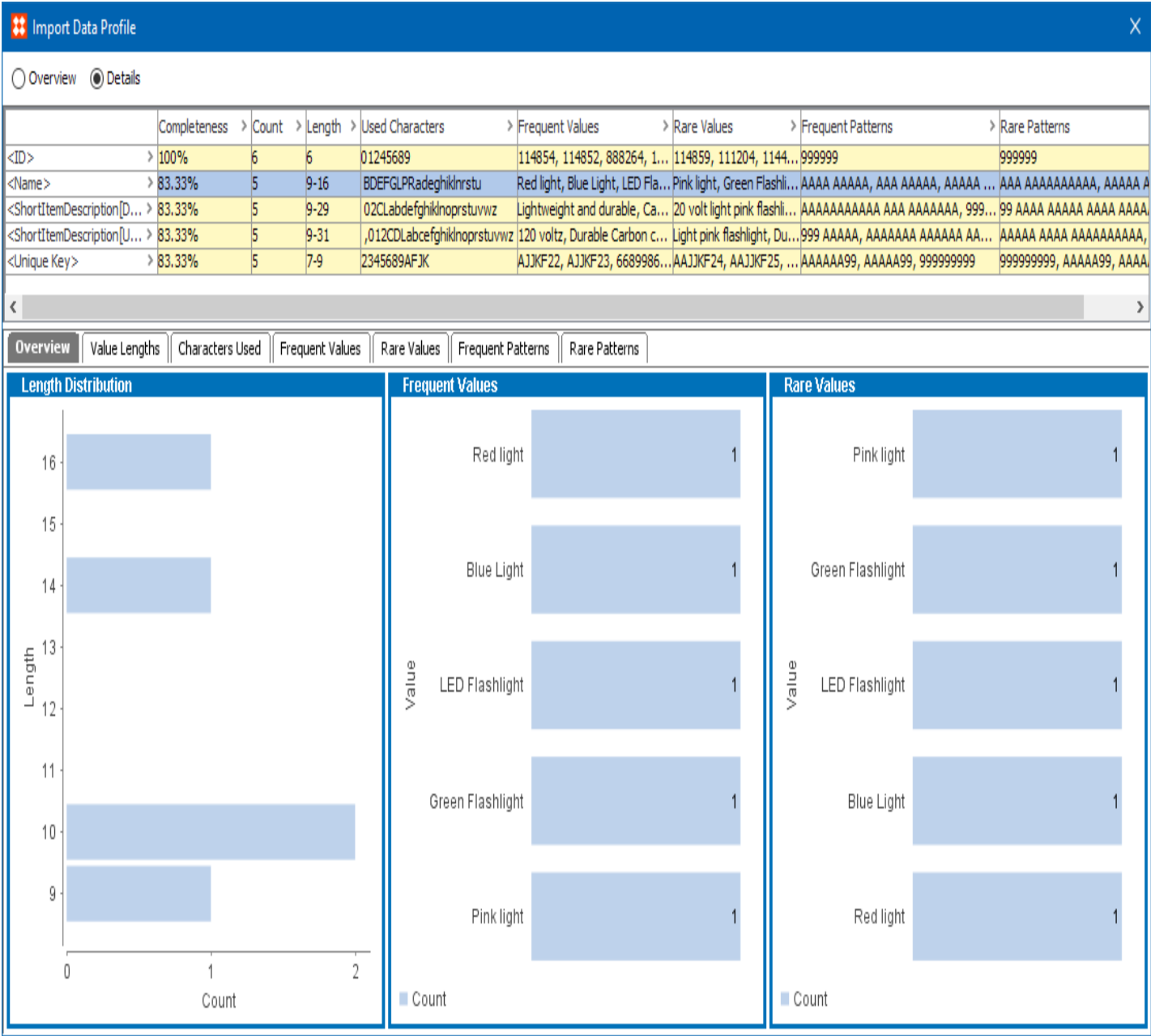
Not that when you select and right click on either the Most Complete or the Least Complete Column you will find 2 options which are shown below:

Most Complete Column	
Column	Completeness
> <ID>	100%
> <Name>	
> <ShortItemDescription[DK]	
> <ShortItemDescription[UK]	
> <Unique Key>	83.33%

When **Filtering Enabled** is selected, you will get a dropdown list which allows you to filter the based on: Empty Values, Non Empty Values, Equals, Begins with, Ends with, Contains, '<' (lesser than), and '>' (Greater than) options which is nothing but a criteria. Once used, based on the criteria selected, the results will be displayed.

Similarly, when the option Copy table for Excel is selected, both the columns are selected and can be pasted in an excel sheet.

- 4. Select the **Details** radio button to display a master section at the top and a details section at the bottom.
In the master section, every column of the import file is represented as a row.



The columns in the master section contain the following information.

Column	Description
Completeness	Shows the degree to which the attribute is populated in the category rounded to the nearest whole percent. Inherited values are included in the calculation.

Column	Description
Count	Lists the number of products below the category where the attribute has a value (including inherited values) and the number of products below the category where the attribute could possibly have a value.
Used Characters	List of the characters used for values in the column
Length	Displays value lengths, for example, 3-99 or 10.
Frequent Values	Displays a comma-separated list of the most frequently appearing values. The list is sorted ascending on frequency.
Rare Values	Comma-separated list of the least frequently appearing values. Sorted descending on frequency.
Frequent Patterns	This tab lists the most frequent patterns of the values of a given attribute together with the number values that match the pattern. The pattern describes the structure of a value.
Rare Patterns	This tab lists the most frequent patterns of the values of a given attribute together with the number values that match the pattern. The pattern describes the structure of a value.

5. Select a row in the master section to populate the details section at the bottom. The five tabs present a different view of the information that is displayed in the master section.

- **Overview** - Displays widgets for length distribution, frequent values, and rare values bar charts. If all values are of the same length, the length distribution bar chart is not displayed. Likewise, if only a single unique value exists in the import column, the frequent and rare values widgets are not displayed.
- **Value Lengths** - Displays the number of occurrences of each value length.
- **Characters Used** - Displays the number of occurrences of each character.
- **Frequent Values** - Lists the most frequently appearing values.
- **Rare Values** - Lists the least frequently appearing values.
- **Frequent Patterns** - Lists the frequent pattern in which the values are entered
- **Rare Patterns** - List the most rare pattern in which the values are entered

For more information on Data Profiling, see the **Data Profiling** documentation.

On the Fly Data Validation

When you map columns in the import file to data in STEP with the Data Quality functionality, the mapped data is validated on the fly. The data validation status is displayed with an icon on the header of each column in the Result area of the **Map Data** step. Cells with validation errors are highlighted with red background color.

Result:

Map to: Product

ID = <ID> ✓	Name = <Name> ✓	ShortItemDescription = <ShortItemDescription[UK]> ✓	ID KeyID = SupplierPartNumber = <Unique Key> ⛔
114852	Blue Light	Durable, lightweight flashlight	AJJKF22
114854	Green Flashlight	Durable Carbon case	AJJKF23
111204	LED Flashlight	120 voltz	668998643
114859	Pink light	Light pink flashlight	AAJJKF24
114440	Red light		AAJJKF25
888264		120 voltz	

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Mouse over a cell with validation errors and the hover text provides information about the problem. If you apply transformations to a result column, the validation feature is also available in the Transformation Preview. Data is re-validated with every change. See the **Inbound Map Data - Transform** topic in the **Data Exchange** documentation for more details.

Important: Only data for the first 200 rows in the import file is validated and displayed in the result panel.

It is possible to start an import process even if validation errors have been discovered and the same errors are displayed in the background process in the execution report.

Inbound Map Data - Map

When an inbound file does not contain a header row, you must use the Map button to map the data manually. Manual mapping can also be used to modify the mappings generated with the Auto Map feature. For information about automatic mapping, see the **Inbound Map Data - Auto Map** documentation.

Map Data

Source:

<Name>	>	<Parent ID>	>	Primary Color	>	Secondary Color	>
Mens T PBO		18209		Black		Orange	
Mens T PBG		18209		Blue		Green	

Result:

Map to: Product

Name=<Name> ✓	Parent=<Parent ID> ✓	PrimaryColor=Primar... ✓	SecondaryColor=Se... ✓
Mens T PBO	Cotton T-Shirts	Black	Orange
Mens T PBG	Cotton T-Shirts	Blue	Green

Auto Map Map Constant Remove Transform Generate Profile

Source columns must be mapped individually. If multiple source columns are selected prior to clicking the Map button, a message is displayed saying the source selection is not valid.

Map Data

Source:

<ID>	>	<Name>	>	<Parent ID>
20862		20862		20859
20883		20883		20859

Message



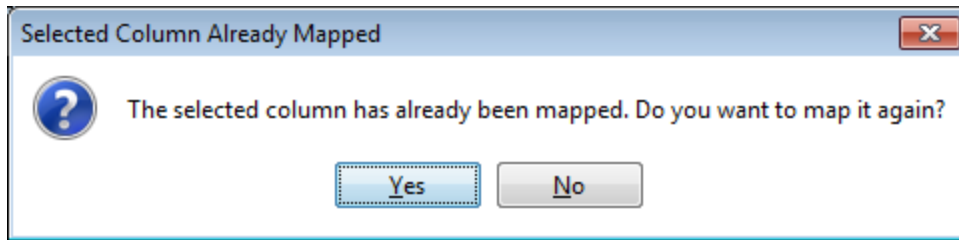
Illegal source selection: Please select cells in one column only.

OK

Mapping the Same Column Multiple Times

A single source column can be mapped as many times as needed. For example, if an import column contains three dimensions for height, length, and width, you might want to map it three times so that the elements can be saved to three separate STEP attributes. For an example, see **Split and Extract Data Example**.

When you attempt to map a column that is already mapped, although a warning displays, you can click **Yes** to map it again, or click **No** and select a different column.



The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Mapping Options

These mapping options are available both in IIEP and Import Manager.

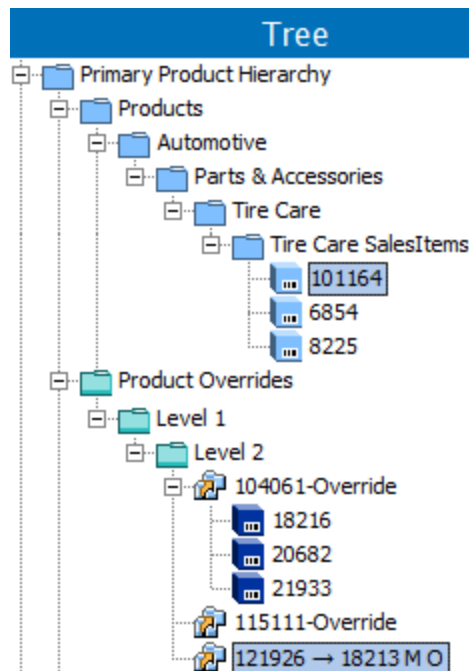
See the following topics for the specific options available for manually mapping, based on the object type selected in the **Map to** dropdown parameter.

- Add Child to Override - Map Inbound
- Asset Reference - Map Inbound
- Attribute - Map Inbound
- Attribute Info - Map Inbound
- Attribute Validation Info - Map Inbound
- Classification Reference - Map Inbound
- Data Container Attribute - Map Inbound
- Delete Values - Map Inbound
- Entity Reference - Map Inbound
- ID or Key - Map Inbound
- Multivalued Variable - Map Inbound
- Name - Map Inbound
- Object Type - Map Inbound
- Overrides Product - Map Inbound
- Parent - Map Inbound
- Product Classification Links - Map Inbound
- Reference Meta-Data - Map Inbound
- Variable - Map Inbound

Add Child to Override - Map Inbound

A product override folder can include products from various 'standard' (non-override) product families. Similarly, a product override leaf object (child to override) can live under a different product folder (override or standard) than the folder in which its corresponding standard object resides.

A single import file can include multiple products added as a child to a single Product Override or to multiple Product Overrides. For this example, two 'salesitem' product objects are added as children to two separate Product Override objects.



The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Prerequisites

The required steps include:

1. Setting up the Product Override object type in System Setup. See the **Product-Override Object Types and Commercial Object Types** topic in the **System Setup / Super User Guide** documentation.
2. Verifying the import file before starting the import as defined in the **Verify Import File** section below.
3. Mapping 'Add child to override' as defined in the **Map Add Child to Override** section below. To add a child during creation of an initial product override, see the **Product Overrides** topic in the **Getting Started / User Guide** documentation.

Verify Import File

To ensure that the object is imported successfully and correctly, before starting the import, open the import file and ensure that the following fields exist:

- **ID:** The ID of the parent that the child product override will live under.
- **Add Child to Override:** The ID of the product in STEP that will get an override version of itself as a child to a parent override.

This example shows an Excel import file (although any inbound format can be used). The file adds two objects as children for one product override (ID 104061), and one of those objects is also added as a child on another product override (ID 115111). Any additional data on the final row in this file will be applied to the product override (ID 121926) based on the setting of the Mandatory parameter as defined below.

A	B
<ID>	Add Child Override
104061	101164
104061	6854
115111	6854
121926	

Map Add Child to Override

Once System Setup is configured, an object can be imported as a child to a product override using Import Manager or Inbound Integration Endpoint.

1. Use Auto Map to map the ID, or map it manually. For more information, see **ID or Key - Map Inbound**.
2. In the Source section, first select the Add Child Override column and then click the **Map** button to display the 'Map ...to' dialog. The label of the field is used as the title of this dialog.

The screenshot shows the 'Map Data' dialog with a 'Source' table and a 'Result' table. The 'Source' table has columns '<ID>' and 'Add Child Override'. The 'Result' table shows the mapping result: 'ID = <ID>' with a green checkmark. A red arrow points from the 'Map' button in the 'Source' table to the 'Map Add Child Override to' dialog. The dialog has a title bar with a close button and a list of radio buttons: ID, Name (selected), Attribute, Product Classification Link, Product Reference, and Asset Reference.

3. On the Map ... to dialog, select the **Add child to override** radio button.

☐ Overrides product

☒ Add child to override

☒ Mandatory

Cancel

OK

4. Check the **Mandatory** option appropriately:
- If **checked**, imported objects must have an Add Child Override value in the file. If an Add Child Override value does not exist for an object, the object is skipped and not imported.

If **unchecked**, all objects are imported regardless if an Add Child Override value is specified or not.
5. Click **OK** and the **Result** panel displays your Add Child Override values. A validated data column is marked with a green check mark as shown below.

Result:

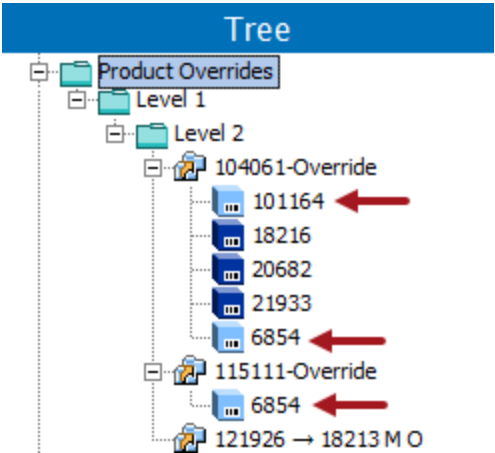
Map to: Product

ID=<ID> ✓	ProductOverrideChild=Add Child Override ✓
104061	101164
104061	6854
115111	6854
121926	

6. Complete the mapping and initiate the import.

Result

When import is successful, the objects display under the designated Product Override parent in Tree, as indicated by the arrows in the following image. The last object was not modified since mandatory was checked and no ID was included for the Add Child Override field in the import file.



Asset Reference - Map Inbound

Asset references are available for product, classification, and entity objects. The reference types available for mapping during import are based on the asset reference types that exist in STEP.

Before you create mappings, note the following:

- The existing reference types for the super type selected in the wizard are available for mapping. New reference types cannot be specified or created during import. The reference type is not read from the input file.
- The same column can be mapped multiple times when a different asset reference type is needed for the same target.

Below are the steps required to map the following asset references:

- Product to Asset References
- Classification to Asset References
- Entity to Asset References

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Prerequisites

To ensure that the object is imported successfully and correctly, before starting the import, open the import file and ensure that the following data exists:

- The inbound file must contain the ID / key (not the name) of the source object for the reference (product, classification, or entity).
- The inbound file must contain the ID / key (not the name) of the target object for the reference (asset).
- Objects in the inbound file must already exist in STEP.
- The desired reference type must be valid for the object types being referenced.

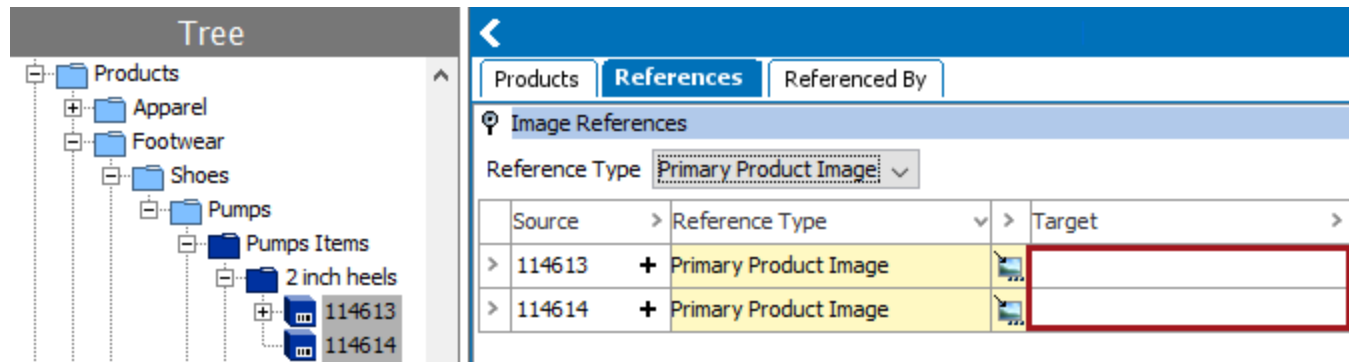
If the objects to be imported or the Target to be mapped is identified via a unique key, you must verify that the Key required is available and activated. For more information, see the **Activating and Deactivating Keys** topic of the **System Setup / Super User Guide** documentation.

Map Product to Asset Reference

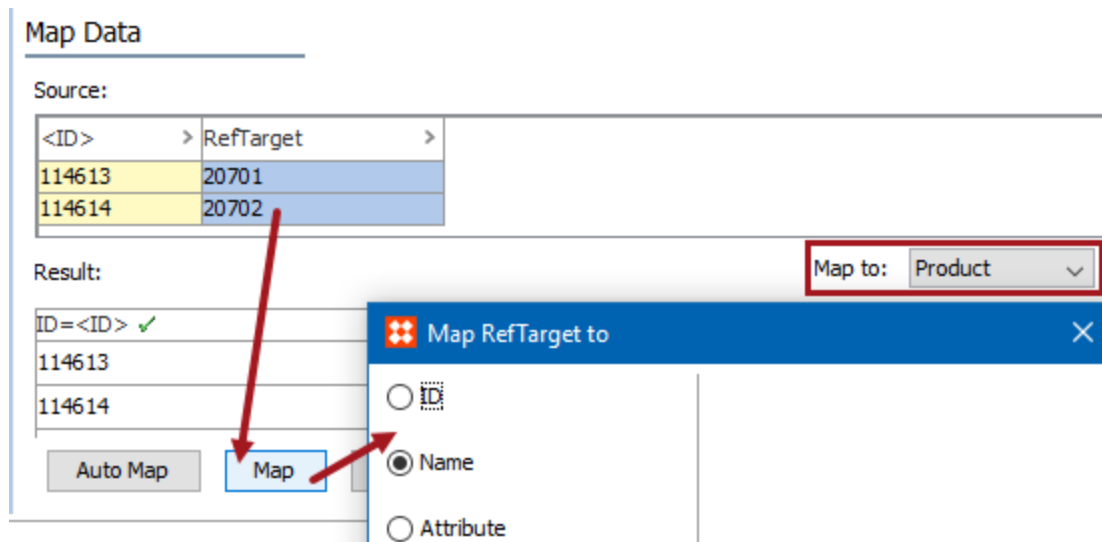
This example shows an Excel import file (although any inbound format can be used). The file includes target asset objects to be used as the primary product image for two products (indicated by the ID column).

A	B
<ID>	RefTarget
114613	20701
114614	20702

The import will specify an asset that should reference a product using the reference type 'Primary Product Image.' Before the import, there are no image references for either source product, as shown simultaneously in the image below.



1. In the Source window, select the ID or Key column and map it. For more information see the **ID or Key - Map Inbound** documentation.
2. Below the Source section, in the **Map to** parameter, select the **Product** super type.
3. In the Source section, select the reference target column and then click the **Map** button to display the Map ... to dialog. The label of the field is used within the title of this dialog.



4. On the 'Map ...to' dialog, select the **Asset Reference** radio button to display the list of available reference types on your system. Select the radio button of the desired reference type. Because the Product super type was selected previously, only product-to-asset references are displayed.

Map RefTarget to

☐ ID

☐ Name

☐ Attribute

☐ Product Classification Link

☐ Product Reference

☒ Asset Reference

☐ Classification Reference

Target ID Aspect

ID

☐ Icons

☐ MSDS

☐ Owners Manual

☐ PDF

☒ Primary Product Image

☐ Product Images

☐ Showroom Preferred

5. Check the **Mandatory** option appropriately:
- If **checked**, imported objects must have an ID for the selected reference in the file. If an ID for the reference does not exist for an object, the object is skipped and not imported.
 - If **unchecked**, all objects are imported regardless if an ID for the reference is specified or not.

☐ Add child to override

☒ Mandatory

Cancel

OK

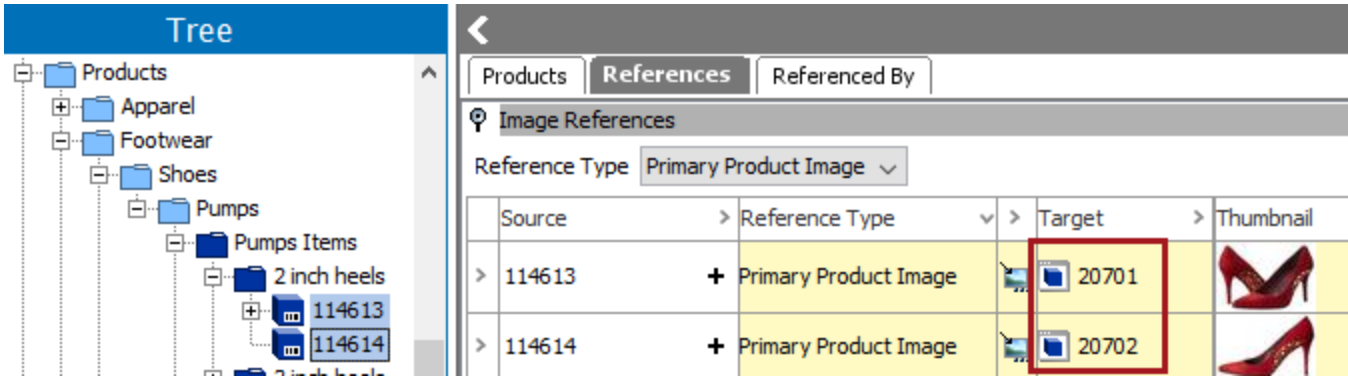
6. Click **OK** and the **Result** panel displays your asset reference values. The Result section shows the selected product-to-asset reference type. A validated data column is marked with a green check mark as shown below.

Result:

Map to: Product

ID=<ID> ✓	PrimaryProductImage AssetReference=RefTarget ✓
114613	20701
114614	20702

7. Complete the mapping and initiate the import. When the import is successful, the imported references display on the References tab of the product within the appropriate flipper.



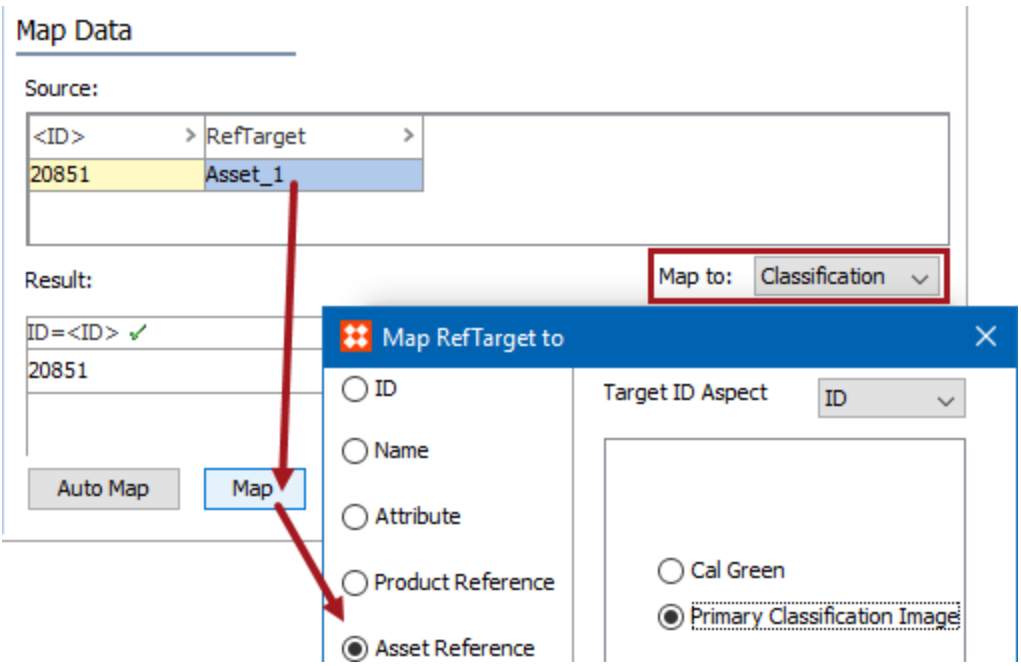
Map Classification to Asset Reference

Importing a classification-to-asset reference is similar to the previous steps for a product-to-asset reference, with the following differences:

- If you want to make references from the classification objects to an Entity / Asset / classification, verify that the object and Entity / Asset / classification exists in STEP and also the reference type is valid for the object types in the input file. Lastly, verify that the classification ID is specified in the input file, not the name.
- The import file must include the ID of the classification and the ID of the asset.

A	B
<ID>	RefTarget
20851	Asset_1

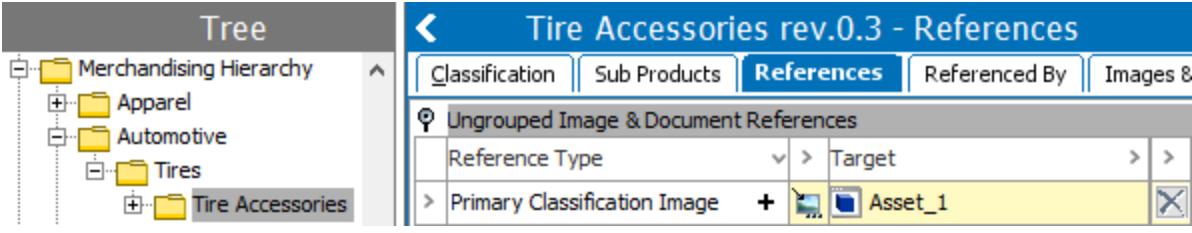
- The **Map to** parameter must be set to the Classification super type prior to selecting the reference target and clicking the Map button.



- Only classification-to-asset references are displayed after selecting the Asset Reference radio button on the **Map ... to** dialog. The Result section shows the selected classification-to-asset reference type.



- When the import is successful, the imported references display on the References tab of the classification within the appropriate flipper.



Map Entity to Asset Reference

Importing an entity-to-asset reference is similar to the previous steps for a product-to-asset reference, with the following differences:

- The import file must include the ID of the entity and the ID of the asset.

A	B
<ID>	RefTarget
CUS_107835	Asset_3

- The **Map to** parameter must be set to the Entity super type prior to selecting the reference target and clicking the Map button.

Map Data

Source:

<ID>	> RefTarget	>
CUS_107835	Asset_3	

Result:

ID=<ID> ✓
CUS_107835

Map to: Entity

Auto Map Map

Map RefTarget to

☐ ID
☐ Name
☐ Attribute
☐ Product Reference
☒ Asset Reference

Target ID Aspect ID

☐ GDSNFormat
☐ Primary Entity Image
☒ Primary Product Image

- Only entity-to-asset references are displayed after selecting the Asset Reference radio button on the **Map ... to** dialog. The Result section shows the selected entity-to-asset reference type.

Result:

Map to: Entity

ID=<ID> ✓	Primary Entity Image AssetReference=RefTarget ✓
CUS_107835	Asset_3

- On the Identify Destination step of the import, on the **Default Parent** parameter, click the ellipsis button (...) to display the valid entity parent nodes. Select a node and click the **Select** button.

Identify Destination

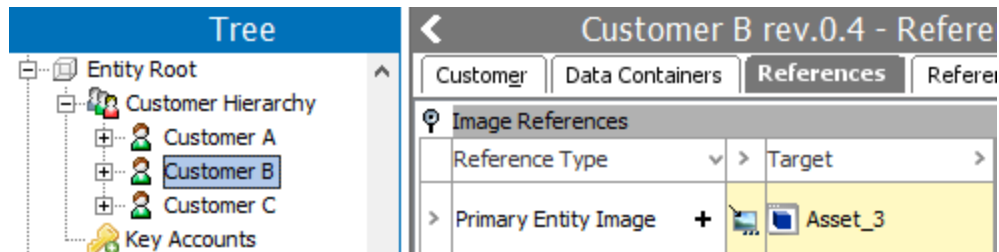
Approver User J (USERJ) ...

Import Workspace Main

Default Parent Entity Root (Customer_Data_Root) ...

Default Object Type: All Customers

- When the import is successful, the imported references display on the References tab of the entity within the appropriate flipper.



Attribute - Map Inbound

Mapping for attributes is available for both attributes that already exist in STEP and those that need to be created, as defined below.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Map to an Existing Attribute

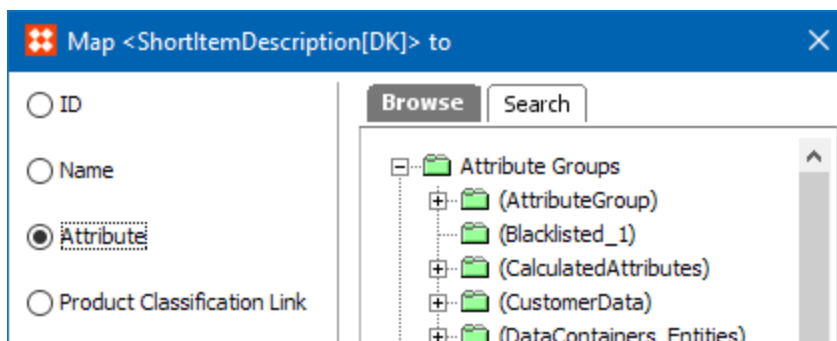
You can map a data column in your input file to an attribute that already exists in STEP. For Classification(s) you can map the data to a description attribute only.

1. In the Source section, select the column of data that you want to map to an attribute and click the **Map** button.

Source:

<ID>	>	<Name>	>	<Parent ID>	>	Primary Color	>	Secondary Color	>
MT18400		Mens T PBO		18209		Black		Orange	
MT18401		Mens T PBG		18209		Blue		Green	
MT18402		Mens T PGS		18209		Green		Silver	
MT18403		Mens T PGW		18209		Gray		White	
MT18404		Mens T POY		18209		Orange		Yellow	

2. In the 'Map ... to' dialog, select the **Attribute** radio button.



3. In the displayed attribute selector, use Browse or Search to select the STEP attribute to be mapped.

Search is available on all or part of the attribute's name or ID. Click the advanced search flipper to use other options, such as searching for attributes with a specific value.

Note: Only description attribute(s) can be mapped to a classification. Selecting a specification attribute displays an error.

4. If the attribute selected is an LOV, set the **Match value ID in LOV** appropriately:

- If **checked**, the ID of the LOV must be included in the inbound file.
- If **unchecked**, the LOV ID is not required.

5. Check the **Mandatory** option appropriately:

- If **checked**, imported objects must have an attribute value in the file. If an attribute value does not exist for an object, the object is skipped and not imported.
- If **unchecked**, all objects are imported regardless if an attribute value is specified or not.

- Click **OK** and the **Result** panel displays your selection of column and attribute. A validated data column is marked with a green check mark as shown below.

Result: Map to: Product

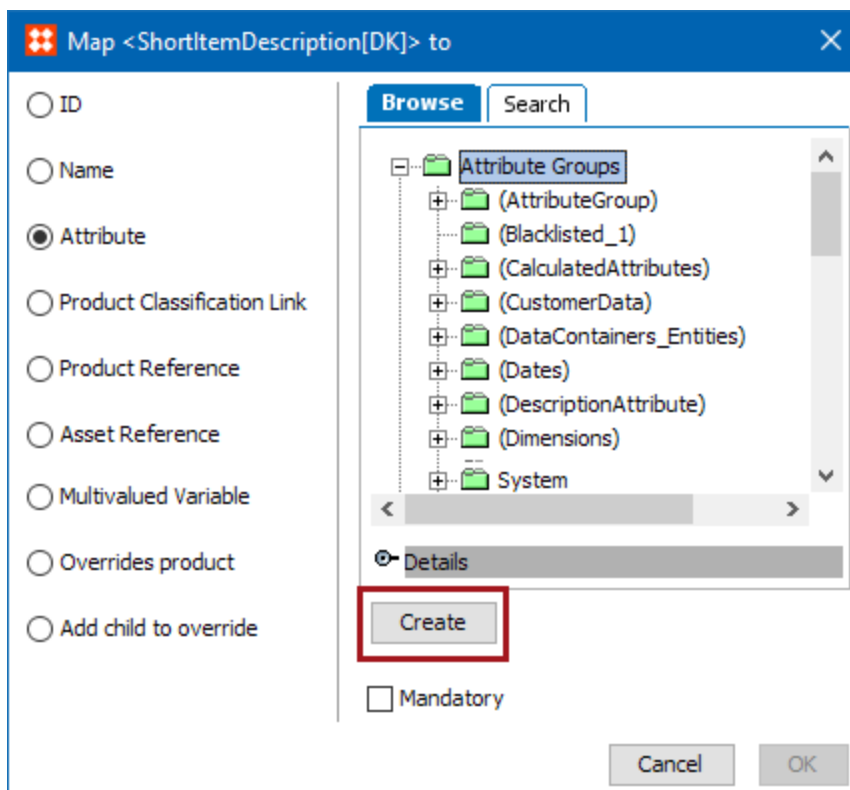
ID=<ID> ✓	Name=<Name> ✓	Parent=<Parent ID> ✓	PrimaryColor=Primary Color ✓
MT 18400	Mens T PBO	Cotton T-Shirts	Black
MT 18401	Mens T PBG	Cotton T-Shirts	Blue
MT 18402	Mens T PGS	Cotton T-Shirts	Green
MT 18403	Mens T PGW	Cotton T-Shirts	Gray
MT 18404	Mens T POY	Cotton T-Shirts	Orange

- Complete the mapping and initiate the import.

Create a New Attribute and Map

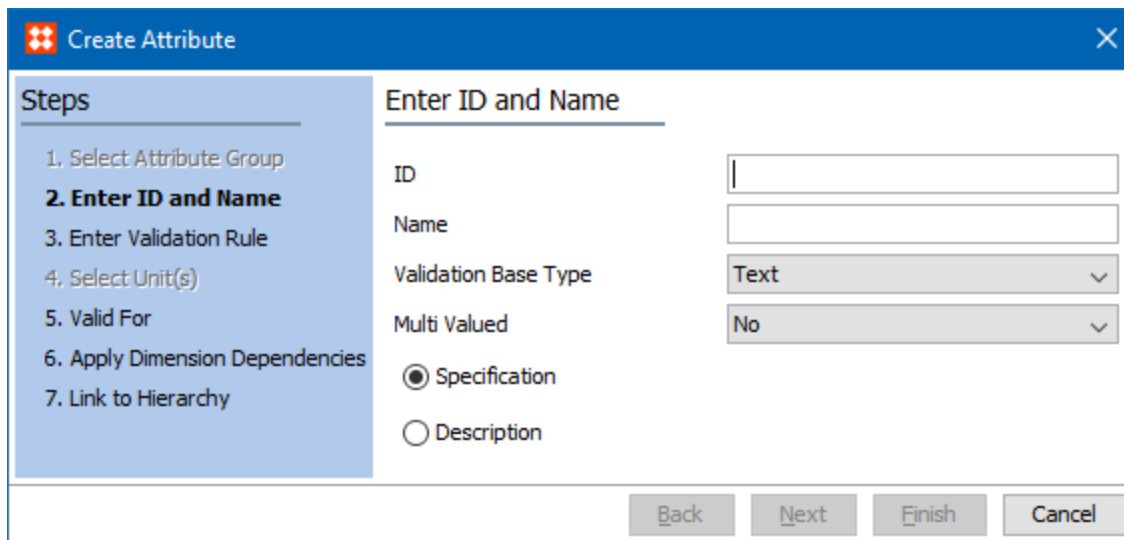
When a data column in your input file is an attribute that does not yet exist in STEP, you can create the new STEP attribute during the import process, and then map the data column to it.

- In the 'Map to' window, select the attribute radio button and navigate to the appropriate attribute group to house the new attribute. Click the **Create** button.



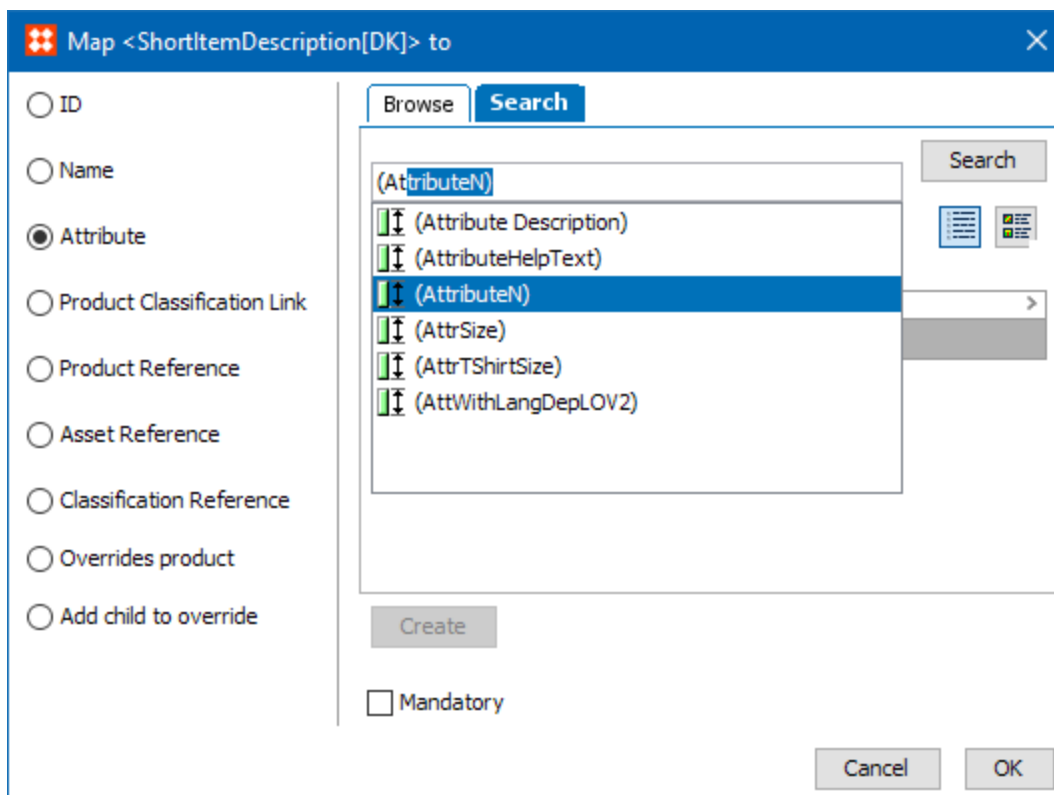
2. In the Create Attribute window, provide required information for the new attribute. For details on how to create an attribute, see the **Creating Attributes** topic in the **System Setup** documentation.

Note: Only description attributes are valid for classifications



The 'Create Attribute' dialog box is shown with the 'Enter ID and Name' step selected. The 'Steps' pane on the left lists seven steps: 1. Select Attribute Group, 2. Enter ID and Name (highlighted), 3. Enter Validation Rule, 4. Select Unit(s), 5. Valid For, 6. Apply Dimension Dependencies, and 7. Link to Hierarchy. The main area contains fields for 'ID' and 'Name', a 'Validation Base Type' dropdown set to 'Text', and a 'Multi Valued' dropdown set to 'No'. Below these are two radio buttons: 'Specification' (selected) and 'Description'. At the bottom are 'Back', 'Next', 'Finish', and 'Cancel' buttons.

3. Click **Finish** on the Create Attribute window to close it and return to the Map To dialog.
4. In the attribute selector window, use Browse or Search to select the STEP attribute just created.



The 'Map <ShortItemDescription[DK]> to' dialog box is shown. On the left is a list of attribute types with radio buttons: ID, Name, Attribute (selected), Product Classification Link, Product Reference, Asset Reference, Classification Reference, Overrides product, and Add child to override. The main area has 'Browse' and 'Search' tabs. The 'Search' tab is active, showing a search bar with '(AttributeN)' and a list of attributes: (Attribute Description), (AttributeHelpText), (AttributeN) (highlighted), (AttrSize), (AttrTShirtSize), and (AttrWithLangDepLOV2). There is a 'Search' button and a 'Create' button below the list. At the bottom are 'Cancel' and 'OK' buttons, and a 'Mandatory' checkbox.

5. Check the **Mandatory** option appropriately:

- If **checked**, imported objects must have an attribute value in the file. If an attribute value does not exist for an object, the object is skipped and not imported.
- If **unchecked**, all objects are imported regardless if an attribute value is specified or not.



The screenshot shows a dialog box with a light gray background. On the left, there is a radio button next to the text "Add child to override". To the right of this, there is a vertical line. To the right of the line, there is a checked checkbox next to the text "Mandatory". At the bottom right of the dialog box, there are two buttons: "Cancel" and "OK".

6. Click **OK** to map this new attribute to the previously selected data column.

7. Complete the mapping and initiate the import.

Attribute Info - Map Inbound

The Attribute Info mapping options are used to create new attributes or update existing attribute definitions. This option is available only when Attribute is selected from the object super type 'Map to' dropdown. The attribute definitions available for import can be found in the attribute editor as defined in the **Attribute Info File Structure Requirements** section below.

This example shows an Excel import file (although any inbound format can be used). The sample file updates the attribute's Type and Calculated parameters for three attributes. Additional attribute info columns can be added to the import file as needed.

A	B	C	D
<ID>	<Name>	<Attribute Type>	<Attribute Calculated>
BrandName	Brand Name	Normal	false
Fit	Fit	Normal	false
ShippingRestrictions	Shipping Restrictions	Property	true

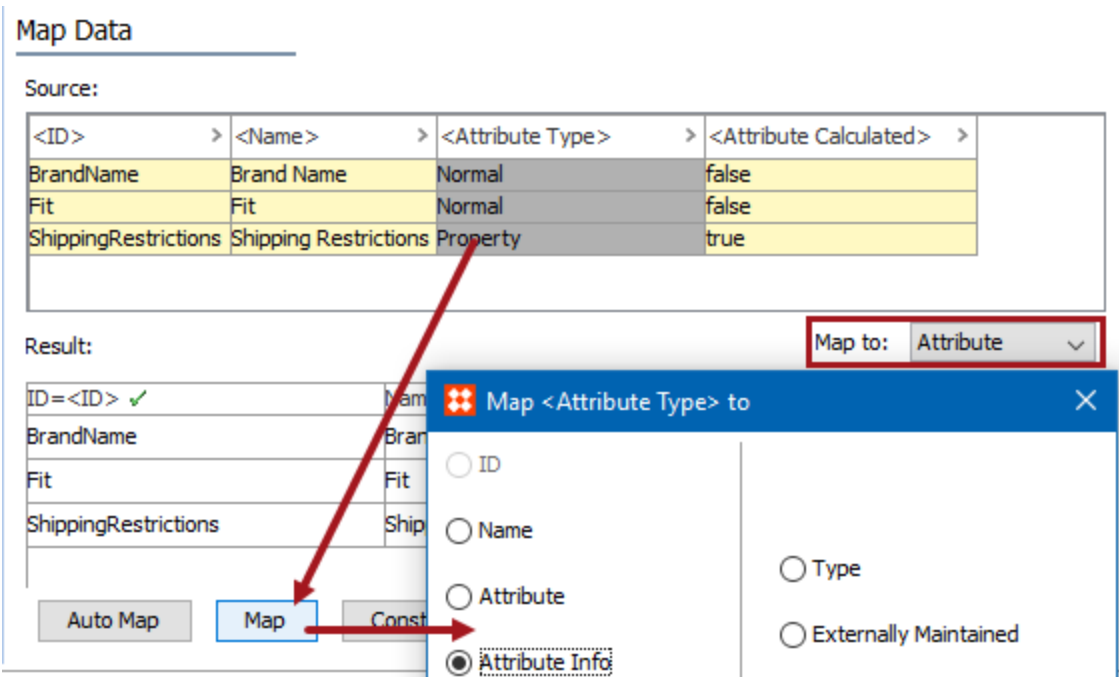
The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Verify Import File

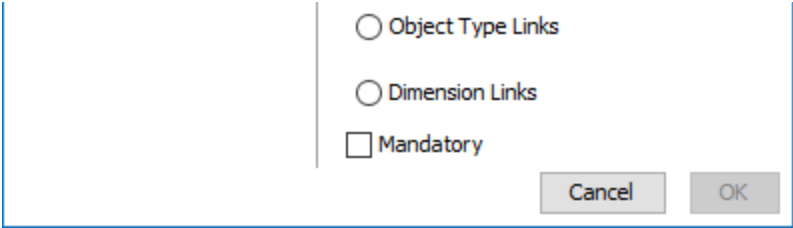
The import file must include the attribute ID or key, and must follow the format defined in the **Attribute Info File Structure Requirements** section below.

Map Attribute Info

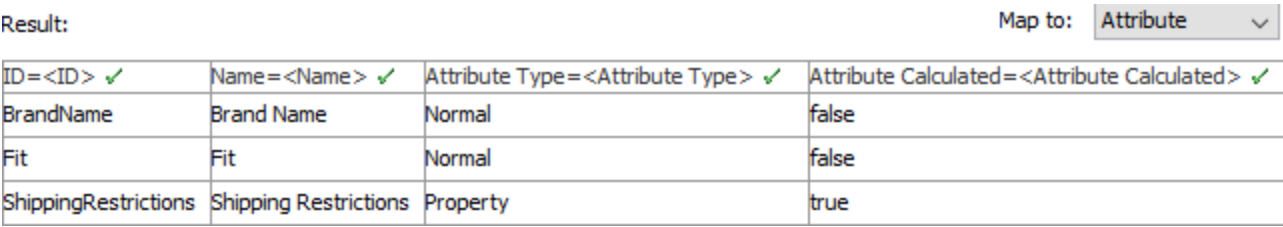
1. Below the Source section, in the **Map to** parameter, select the **Attribute** super type.
2. Use Auto Map to map the ID, or map it manually. For more information, see **ID or Key - Map Inbound**.
3. In the Source section, first select the attribute option column and then click the **Map** button to display the 'Map ... to' dialog. The label of the field is used as the title of this dialog.



- 4. Select a specific Attribute Info option from the radio buttons on the right panel of the 'Map ... to' dialog. This selection enables the OK button. Each of the available options are defined in the **Attribute Info File Structure Requirements** section below.
- 5. Check the **Mandatory** option appropriately:
 - If **checked**, imported objects must have a value for the selected Attribute Info in the file. If a value does not exist for an object, the object is skipped and not imported.
 - If **unchecked**, all objects are imported regardless if the selected Attribute Info is specified or not.



- 6. Click **OK** and the **Result** panel displays your Add Child Override values. A validated data column is marked with a green check mark as shown below.



7. Complete the mapping and initiate the import.

Attribute Info File Structure Requirements

On the Map ... to dialog, after selecting the Attribute Info radio button, the following options are displayed on the right side. These options correspond to parameters available in the attribute editor under the identified flippers.



Important: The predefined values shown below are case-sensitive. An error is displayed during mapping if the case is not as defined. For example, a value of 'FALSE' is not allowed when a value of 'false' is expected.

Data on the attribute editor can be set for parameters in the following tabs and flippers as defined below:

1. Attribute Tab, Description flipper
2. Attribute Tab, Attribute Validation flipper
3. Attribute Tab, Units flipper
4. References Tab, Attribute Groups flipper
5. Validity Tab

Note: No updates are made in STEP to existing values when an attribute definition is imported with an empty value.

Attribute Tab Description Flipper Parameters

- **Type** sets an attribute as description when the value is **Property**, or specification when the value is **Normal**.
- **Externally maintained** sets an attribute as externally maintained when the value is **true**, or internally maintained when value is **false**. When updating an existing attribute that has a value for this parameter, single update mode is required.
- **Full Text Indexed** sets an attribute as full text indexed when the value is **true**, or not when the value is **false**. When updating an existing attribute that has a value for this parameter, single update mode is required.
- **Mandatory** sets an attribute as mandatory when the value is **true** or not when the value is **false**. This field is never output in any export file, and must be added manually to an input file.
- **Completeness** previously used to set the completeness of an attribute but is a legacy option. This field is no longer used by STEP. To learn about attribute and data completeness, and how to calculate them now, see the **Data Profiling** documentation.

- **Calculated** sets an attribute as calculated when the value is **true**, or not when the value is **false**. For more information on calculated attributes, see the **Calculated Attributes** section of the **System Setup / Super User Guide** documentation.
- **Calculated Value Template** can only be used when the Calculated field is set to 'yes' in the workbench (see previous bullet) and sets the value template. This field is never output in any export file, and must be added manually to an input file. For more information on calculated attributes, see the **Calculated Attributes** section of the **System Setup / Super User Guide** documentation.
- **Calculated Unit Template** can only be used when the Calculated field is set to 'yes' in the workbench. This is only available for attributes with a validation base type of Number, Embedded Number, Fraction, Integer, Number Range, Numeric Text, or Regular Expression. This field is never output in any export file, and must be added manually to an input file. For more information on calculated attributes, see the **Calculated Attributes** section of the **System Setup / Super User Guide** documentation.
- **Dimension links** sets the Dimension Dependencies of an attribute. For example, language, country, or market for an attribute. Updating a language dimension to a country dimension is not allowed and displays an error for incompatible dimensions. When adding country dimension to a language dimension attribute, single update mode is required.

Attribute Tab Attribute Validation Flipper Parameters

- **Multi Valued** sets an attribute as multivalued when the value is **true**, or single valued when the value is **false**.

Attribute Tab Units Flipper Parameters

- **Default Unit ID** sets a default unit and is only available for attributes with a validation base type of Number, Embedded Number, Fraction, Integer, Number Range, Numeric Text, or Regular Expression. This determines which unit displays with a check mark in the Default Unit column of the Units flipper. This field is never output in any export file, and must be added manually to an input file.

When creating an attribute with a default unit via import, the unit must already exist in STEP.

For more information on validation base type, see the **Validation Rules** section of the **System Setup / Super User Guide** documentation.

For more information on units, see the **Units** section of the **System Setup / Super User Guide** documentation.

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

References Tab Attribute Groups Flipper Parameters

- **Attribute group links** specifies under which attribute group this new attribute should be created or linked. The attribute group ID must be included in the input file. When linking to more than one attribute group, separate ID with a semicolon delimiter. This data is added to the attribute editor's References tab under the In Attribute Groups flipper.

Validity Tab Parameters

- **Object type links** sets the validity of an attribute which defines in which object type level this attribute should be valid. This field is never output in any export file, and must be added manually to an input file. This data is

Attribute Validation Info - Map Inbound

The Attribute Validation mapping options are used to set parameters for attributes. This option is available only when Attribute is selected from the object super type 'Map to' dropdown. The attribute definitions available for import can be found in the attribute editor as defined in the **Attribute Validation Info File Structure Requirements** section below.

This example shows an Excel import file (although any inbound format can be used). The sample file creates multiple attributes and sets the Validation Base Type and ID for an LOV attribute. Additional attribute validation information columns can be added to the import file as needed.

A	B	C	D	E
<ID>	<Name>	<AttributeGroup Ref ID>	<Attribute Validation Base Type>	<LOV ID>
TextAttrb	Text Attribute	ItemBrandInformation	Text	
LOVAttrb	LOV Attribute	ItemBrandInformation		Colour
EmbNumAttrb	Embedded Number At	ItemBrandInformation	Embedded_Number	

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Verify Import File

The import file must include the attribute ID, and must follow the format defined in the **Attribute Validation Info File Structure Requirements** section below.

Map Attribute Validation Info

1. Below the Source section, in the **Map to** parameter, select the **Attribute** super type.
2. Use Auto Map to map the ID, or map it manually. For more information, see **ID or Key - Map Inbound**.
3. If mapping manually, in the Source section, first select the attribute option column and then click the **Map** button to display the 'Map ... to' dialog. The label of the column is used within the title of the dialog.

Map Data

Source:

<ID>	<Name>	<AttributeGroup ... >	<Attribute Validati... >	<LOV ID>
TextAttrb	Text Attribute	ItemBrandInformation	Text	
LOVAttrb	LOV Attribute	ItemBrandInformation		Colour
EmbNumAttrb	Embedded Number A...	ItemBrandInformation	Embedded_Number	

Result:

ID=<ID> ✓	Name=<Name> ✓
TextAttrb	Text Attribute
LOVAttrb	LOV Attribute
EmbNumAttrb	Embedded Number At

Map to: Attribute

Map <LOV ID> to

☐ ID
☐ Name
☐ Attribute
☐ Attribute Info
☒ Attribute Validation Info

☐ Validation Base Type
☐ Validation Max Length
☐ Validation Input Mask
☐ Validation Min Value
☐ Validation Max Value

Auto Map Map Constant

- Select the Attribute Validation Info option, then select a specific type of validation information from the radio buttons on the right panel of the 'Map ... to' dialog to enable the OK button. Each of the available options are defined in the **Attribute Validation Info File Structure Requirements** section below.
- Check the **Mandatory** option appropriately:
 - If **checked**, imported objects must have a value for the selected attribute information in the file. If a value does not exist for an object, the object is skipped and not imported.
 - If **unchecked**, all objects are imported regardless if the selected attribute information is specified or not.

☐ LOV ID
☐ Legal Units
☐ Mandatory

Cancel OK

- Click **OK** and the **Result** panel displays your attribute information values. A validated data column is marked with a green check mark as shown below.

Result:

Map to: Attribute

ID=<ID> ✓	Name=<Name> ✓	AttributeGroup Ref ID=<...> ✓	Attribute Validation Base Type=<Attribute Validation Base Type> ✓	Attribute Validation LOV ID=<LOV ID> ✓
TextAttrb	Text Attribute	Item Brand Information	Text	
LOVAttrb	LOV Attribute	Item Brand Information		Color
EmbNumAttrb	Embedded Number ...	Item Brand Information	Embedded_Number	

7. Complete the mapping and initiate the import.

Attribute Validation Info File Structure Requirements

On the Map ... to dialog, after selecting the Attribute Validation Info radio button, the following options are displayed.

Map <Attribute Validation Base Type> to

☐ ID

☐ Name

☐ Attribute

☐ Attribute Info

☒ Attribute Validation Info

☐ Variable

☐ Multivalued Variable

☐ Validation Base Type

☐ Validation Max Length

☐ Validation Input Mask

☐ Validation Min Value

☐ Validation Max Value

☐ LOV ID

☐ Legal Units

☐ Mandatory

Cancel

OK

These options correspond to parameters available in the attribute editor Attribute tab, which can be set in the Attribute Validation flipper and the Units flipper.

System Setup

Reference Attribute

Released by

SellingPrice

Shipping Restrictions

Size

TshirtQuantity

SellingPrice - Attribute

Attribute

References

Attribute Transformation

Validity

Profile

Log

State Log

Tasks

Description

Attribute Validation

Units

Aspects

Each available parameter for this mapping option is defined below.

1. **Validation Base Type** sets the Validation Base Type parameter on an attribute. The following text is not case sensitive, but is required to set the validation base type. For LOVs, see the **LOV ID** section below.

Note: No spaces are included in types with multiple words. Instead, use the following list to include an underscore (_) or simply exclude the spaces.

- Condition
- Date
- Embedded_Number
- Fraction
- Integer
- ISODate
- ISODatetime
- Number
- NumberRange
- Numeric_Text
- RegExp
- Text
- Url

For more information on validation base type, see the **Validation Rules** section of the **System Setup / Super User Guide** documentation.

2. **Validation Max Length** sets the Maximum Length parameter which determines the number of characters allowed on an attribute. Primarily, this is used for text and numeric text validation base type attributes.
3. **Validation Input Mask** sets the Mask parameter on an attribute. For more on masks, see the **Input Masks** topic in the **System Setup / Super User Guide** documentation.
4. **Validation Min Value** sets the Minimum Value parameter for an attribute. Primarily, this is used for number and integer validation base type attributes.
5. **Validation Max Value** sets the Maximum Value parameter for an attribute. Primarily, this is used for number and integer validation base type attributes.
6. **LOV ID** can only be used when the Validation Base Type is List Of Values (LOV). Specify the ID of a list of values that already exists in System Setup. For more on LOVs, see the **List of Value (LOVs)** topic in the **System Setup / Super User Guide** documentation.

Note: The Validation Base Type column will be empty for an LOV since it is identified by the ID.

7. **Legal Units** sets the Units parameter options and is only available attribute with Validation Base Types of number, embedded number, fraction, integer, number range, numeric text, or regular expression. For more information, see the **Units** topic in **System Setup / Super User Guide** documentation.

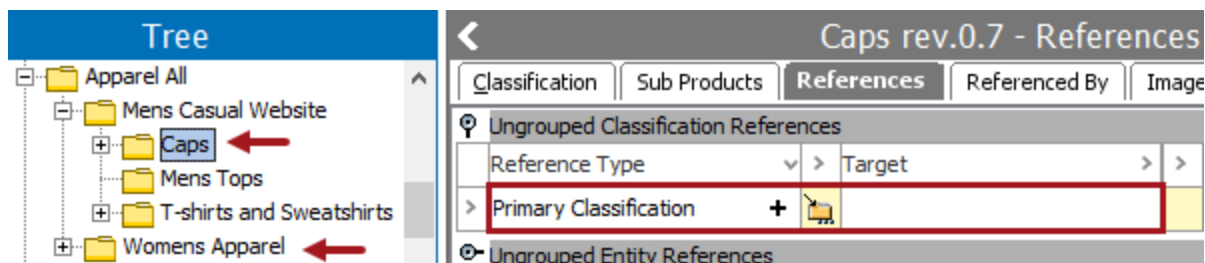
This field is never output in any export file, and must be added manually to an input file. If more than one unit type is being imported for an attribute, then IDs should be separated with a delimiter, for example a semicolon.

Classification Reference - Map Inbound

The classification reference option allows creating references in STEP from classifications to products, classifications, or entities. Mapping a classification reference includes specifying the reference type from those exist in STEP.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

The steps below will create a 'Primary Classification' reference between the 'Caps' source classification folder (ID 22585) and the 'Womens Apparel' target classification folder (ID 22584).



Prerequisites

Prior to import:

1. The object types and classification types must exist in STEP.
2. The reference type must be valid in STEP for the object types in the import file.
3. The import file must include at least two columns: one with the ID / unique key of the source object, and the other with the ID / unique key of the target object. (The reference type will be specified in the wizard.)
4. The import file must include the classification ID, not the classification name.

Map Classification Reference

1. Below the Source section, use the **Map to** dropdown to select the **Classification** object super type.
2. Map the source ID or key column as the ID using the Auto Map button, or manually. For more information, see the **ID or Key - Map Inbound** topic.
3. In the Source section, select the target column of data and click the **Map** button.

Map Data

Source:

Source Object ID	Target Object ID
22585	22584

Result:

ID=Source Object ID ✓
22585

Map to: Classification

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

4. In the **Map to** window, select the **Classification Reference** radio button, and then select the reference type. Only existing reference types are listed. You cannot specify or create new reference types during the import.
5. For the **Target ID Aspect** parameter:
 - Select **ID** when the inbound file includes the ID for the source.
 - Select the **Key** from the dropdown when the inbound file includes a key for the target.

Map Target Object ID to

Target ID Aspect: ID

☐ ID
☐ Name
☐ Attribute
☐ Product Reference
☐ Asset Reference
☒ Classification Reference

Reference Types:

☐ Class2Class
☐ Class2SecondaryClass
☒ Primary Classification

Note: Selecting a reference type that is not valid for the objects being referenced results in a background process error.

6. Check the **Mandatory** option appropriately:
 - If **checked**, imported objects must have an attribute value in the file. If an attribute value does not exist for an object, the object is skipped and not imported.
 - If **unchecked**, all objects are imported regardless if an attribute value is specified or not.

☐ Add child to override

☒ Mandatory

Cancel

OK

7. Click **OK** and the **Result** panel displays the name of the selected reference target object. A validated data column is marked with a green check mark as shown below.

Map Data

Source:

Source Object ID	>	Target Object ID	>
22585		22584	

Result:

Map to: Classification

ID=Source Object ID ✓	Primary Classification	ClassificatonCrossReference=Target Object ID ✓
22585	Womens Apparel	

Auto Map

Map

Constant

Remove

Transform

Generate Profile

8. Complete any additional mapping.
9. Initiate the import.

Results

The result of the import is the new 'Primary Classification' reference on the source classification.

Tree

Apparel All

Mens Casual Website

Caps

Mens Tops

T-shirts and Sweatshirts

Womens Apparel

Caps rev.0.7 - References

Classification

Sub Products

References

Referenced By

Im.

Ungrouped Classification References

Reference Type	>	Target	>	>
> Primary Classification	+	Womens Apparel		

Ungrouped Entity References

Data Container Attribute - Map Inbound

The Data Container Attribute option allows you to map data container attributes to specific (new or existing) data containers when mapping to entity objects.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Map Data Container Attribute

1. Below the Source section, in the **Map to** parameter, select the **Entity** super type.
2. In the Source section, select the column of data containing the first attribute of the data container, and click the **Map** button.

In the example below, the 'Shipping Address.Input City' column will be mapped to the appropriate data container type and attribute.

Source:

<Name>	> Shipping Address.<ID>	Shipping Address.Input City >	Shipping Address.I... >
GR 1	ShippingAddress8876	Roswell	USA
GR 2	ShippingAddress8877	Marietta	USA

Unless configured with auto-ID, a data container ID column (or variable) must also be specified in the attribute mappings. In this example, the 'Shipping Address.<ID>' column represents the data container IDs being imported.

Source:

<Name>	Shipping Address.<ID>	Shipping Address.I... >	Shipping Address.I... >
GR 1	ShippingAddress8876	Roswell	USA
GR 2	ShippingAddress8877	Marietta	USA

Note that the data container type ID column should not be mapped directly, and the ID should instead be mapped to each individual attribute. Explicitly setting an ID for each data container in this way allows for subsequent imports to update the data container instances. If data container IDs are not provided as a source column in the imported file, a variable may be created to serve in its place. This is especially helpful when importing and updating single instance data containers.

Instructions on mapping the ID to each attribute is described below.

For more information on variables, see the **Variable - Map Inbound** section of the **Data Exchange** documentation.

3. In the 'Map ... to' dialog, select the **Data Container Attribute** radio button. This option is only available when the Entity super type is specified in the 'Map To' dropdown.

4. For **Data Container type**, Click the ellipsis button (...) and select the data container type associated with the attribute column being mapped.
5. Select the **Attribute** using the dropdown. Only attributes valid for the data container type mapped in the previous step will display.
6. The **Match value ID in LOV** parameter (shown in the image below), is *only* displayed if the data container attribute holds an LOV where:
 - Users are not allowed to add values (a hard domain LOV) **and**
 - The LOV uses Value IDs

In this case, the attribute column in the import file can include the value IDs. Check the box if the IDs in the Excel sheet should be matched against the IDs in the LOV, and values applied based on the match. For more information on hard domain LOV, see the **List of Values (LOVs)** section of the **System Setup** documentation.

7. For **Data Container ID column**, use the dropdown to select the header of the data container ID column applicable to the data container attribute being mapped.

Note: All source column headers and configured variables are listed in the dropdown.

8. For the **Use Auto-ID** parameter, check the box to assign automatic IDs to data containers if you do not want to manually map an ID column. If a data container ID column is selected this box will become unchecked.
9. Add a separator symbol in the **Data Container separator** field, if applicable. This only impacts data container types configured to allow multiple data containers. The separator signals to the STEP system that there are

multiple data containers for a single entity so that attributes are created / updated accurately.

Note: All attributes being mapped to a specific data container type must use the same ID column or an error will display. Using the example below, Shipping Address2.Country Code (shown) and Shipping Address2.State (not shown) will share the same Data Container ID.

Map Shipping Address 2.Country Code to

☐ ID
☐ Name
☐ Attribute
☐ Product Reference
☐ Asset Reference
☐ Classification Reference
☐ Entity Reference
☐ Reference Meta-Data
☐ Parent
☐ Object Type
☐ Variable
☐ Multivalued Variable
☒ **Data Container Attribute**

Data Container type: Shipping Address 2 (211422)
 Attribute: Country Code
☒ Match value ID in LOV
Data Container ID column: Shipping Address 2. <ID>
☐ Use Auto-ID
 Data Container separator: #
☐ Mandatory

Cancel OK

10. Check the **Mandatory** option appropriately:

- If **checked**, imported objects must have a value in the file. If a value does not exist for an object, the object is skipped and not imported.
- If **unchecked**, all objects are imported regardless if a value is specified or not.

11. Click **OK** and the **Result** panel displays your selection.

Result: Map to: Entity

ID = <ID> ✓	Name = <Name> ✓	ShippingAddress.City = Shipping Address.City ✓
CUS_114563	Customer D	Kennesaw Toronto Paris

Delete Values - Map Inbound

When updating existing objects, use the following delete command to remove values for attributes or the name of the object that exists in STEP. Keep in mind the following points:

- The delete command is case-sensitive and must be included in the import file as: **[delete]**
- The object name can be deleted for attributes, products, assets, classifications, and entities.
- The attribute value for most Validation Base Types can be deleted, excluding 'Condition' and 'Calculated Attribute.' Only the value of an attribute is deleted, and not the object itself.
- If the value being deleted was inherited, the result is not a blank field, but restoration of the inherited value.
- Selected multivalued attributes can be deleted by mapping to 'Multivalued Variable.' For more information, see the **Multivalued Variable - Map Inbound** topic.

The delete function works for both tabular and XML formats as shown below.

Importing the following tabular format import file, the primary color value would be removed if the column was mapped to the appropriate STEP object.

	A	B
1	<ID>	Primary Color
2	179915	[delete]

Importing the following XML format file, the existing value of the attribute 'ManufacturersPartNumber' would be removed, for the mapped column.

```

12 <Products>
13   <Product ID="20883" UserTypeID="Item" ParentID="20859">
14     <Name>20883</Name>
15     <Value AttributeID="ManufacturersPartNumber">[delete]</Value>
16   </Product>
17 </Products>

```

Map to Delete a Value

1. In the inbound file, add the **[delete]** command for the cells where existing data should be deleted.
2. Map the inbound data following the steps required for the type of data. Details for mapping each object type are included in the Inbound **Map Data - Map** topic.
3. Initiate the import.

Example

Consider the 'Name' attribute value on a product.

Product			Sub Products	References	Refer
Description					
	Name	>	>	Value	
>	ID			114861	
>	Name			Product Name	
>	Object Type			Item	

The following import file that is configured to delete the 'Name' attribute value on the product.

	A	B
1	<ID>	<Object Type Name>
2	114861	[delete]

After the successful import, the 'Name ' attribute value is blank.

Product			Sub Products	References	Referenc
Description					
	Name	>	>	Value	
>	ID			114861	
>	Name				
>	Object Type			Item	

Entity Reference - Map Inbound

Entity references can be created for products, classifications, or entities. The available reference types are based on the entity reference types that exist in STEP. The mapping steps are the same, regardless of the reference type needed. Some examples of an input file and results for entity reference types are included below.

The import creates a reference on the source object's editor References tab. It can also be viewed from the target object's editor References By tab.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Prerequisites

The inbound file must include:

1. The ID / key (not the name) of the reference source: product, classification, or entity.
2. The ID / key (not the name) of the reference target: entity.
3. Objects that already exist in STEP.

Map Entity References

1. Below the Source section, use the **Map to** dropdown to select the object super type defined by the source of the reference type. For example, on a product-to-entity, the source is product, and on an classification-to-entity reference, the source is classification.
2. Map the source ID or key column using the Auto Map button, or manually. For more information, see the **ID or Key - Map Inbound** topic.

Columns that include Predefined Header Text can be automatically mapped as defined in the **Inbound Map Data - Auto Map** topic. For the sample below, this includes the <ID> column.

3. In the Source section, select the target column of data and click the **Map** button.

Map Data

Source:

<ID>	Target
20862	121608

Result:

ID=<ID> ✓

20862

Map to: Product

Auto Map Map Constant Remove Transform Generate Profile

4. In the 'Map ... to' dialog, select the **Entity Reference** radio button, and then select the reference type.

Only existing reference types are listed. You cannot specify or create new reference types during the import.

5. For the **Target ID Aspect** parameter:

- Select **ID** when the inbound file includes the ID for the target.
- Select the **Key** from the dropdown when the inbound file includes a key for the target.

The screenshot shows the 'Map Target to' dialog box. On the left, there is a list of radio buttons: Name, Attribute, Product Classification Link, Product Reference, Asset Reference, Classification Reference, and Entity Reference. The 'Entity Reference' option is selected. On the right, there is a 'Target ID Aspect' dropdown menu set to 'ID'. Below this, there is a list of radio buttons for reference types: EntityRef, ProdToEntity, and ShippingAddress. The 'ProdToEntity' option is selected. A red arrow points from the 'Entity Reference' option on the left to the 'ProdToEntity' option on the right.

Note: Selecting a reference type that is not valid for the objects being referenced results in a background process error.

6. Check the **Mandatory** option appropriately:

- If **checked**, imported objects must have an attribute value in the file. If an attribute value does not exist for an object, the object is skipped and not imported.
- If **unchecked**, all objects are imported regardless if an attribute value is specified or not.

The screenshot shows a dialog box with a checkbox labeled 'Mandatory' which is checked. There is also an 'Add child to override' radio button on the left. At the bottom right, there are 'Cancel' and 'OK' buttons.

7. Click **OK** and the **Result** panel displays your selection of reference target. A validated data column is marked with a green check mark as shown below.

Result: Map to: Product

ID=<ID> ✓	ProdToEntity EntityReference=Target ✓
20862	78 South Hampton Rd

- 8. Complete any additional mapping.
- 9. If the Parent ID is not included in the inbound file, set the **Default Parent** parameter on the **Identify Destination** step. The parameter is the same for IIEP and Import Manager as defined in the **Import Manager - Identify Destination** topic.
- 10. Initiate the import.

Entity-to-Entity Reference Example

This sample file will be used to create an Affiliate Of reference between two entity objects holding customer data.

	A	B
1	Entity Source	Entity Target
2	CUS_121603	CUS_121604

Before the import, the source entity has no Affiliate Of references.

Tree

Customers

Alfred Anderson

Benjamin Billingsley

Entity Root

GDSN

Alfred Anderson rev.0.1 - Referen

CustomerData ContainersReferencesReferenced By

Ungrouped Entity References

Reference TypeTarget

Affiliate Of

Tree

Customers

Alfred Anderson

Benjamin Billingsley

Entity Root

GDSN

Benjamin Billingsley rev.0.1 - Referen

CustomerData ContainersReferencesReferenced By

Referenced by Objects

Reference TypeSource

Affiliate Of

Select the Entity object super type before mapping the source and target entities.

Map Data

Source:

Entity Source

Entity Target

CUS_121603

CUS_121604

Result:

Map to: Entity

ID=Entity Source ✓

AffiliateOf EntityReference=Entity Target ✓

CUS_121603

Benjamin Billingsley

Since the Parent ID was not included in the import file, the Default Parent is selected on the Identify Destination step.

Identify Destination

Approver

User J (USERJ)

...

Import Workspace

Main

▼

Default Parent

Customers (CustomerList)

Default Object Type:

Account

▼

Batch Directory

(None)

...

Reset

Test Only Import

☐

Reject New

☐

Reject Updates

☐

Once the import background process completes without error, the new reference is displayed on both the source and the target objects.

Tree

Customers

Alfred Anderson

Benjamin Billingsley

Entity Root

GDSN

Alfred Anderson rev.0.2 - References

Customer

Data Containers

References

Referenced By

Status

State

Ungrouped Entity References

Reference Type	Target	Completeness Score
Affiliate Of	Benjamin Billingsley	

Tree

Customers

Alfred Anderson

Benjamin Billingsley

Entity Root

GDSN

Benjamin Billingsley rev.0.1 - Referenced By

Customer

Data Containers

References

Referenced By

Status

State

Referenced by Objects

Reference Type	Source	Completeness Score
Affiliate Of	Alfred Anderson	

Entity-to-entity references can also be defined by the target entity's Source Record ID. For more information on mapping entity-to-entity references in this way, see the **Entity Reference via Source Record ID - Map Inbound** section of the **Data Exchange** documentation.

Product-to-Entity Reference Example

In this sample file, a product ID will be mapped as the source, and an entity ID is the target of the imported reference.

	A	B
1	<ID>	Target
2	20862	121608

The mapped source and target are displayed in the Result panel.

Map Data

Source:

<ID>	> Target	>
20862	121608	

Result:

Map to: Product

ID = <ID> ✓	ProdToEntity EntityReference = Target ✓
20862	78 South Hampton Rd

Entity Reference via Source Record ID - Map Inbound

In the event that a STEP ID is not available, it is possible to import entities with cross references to other entities via the target entity's Source Record ID. This allows incoming objects to reference other objects that either already exist in STEP or are available in the same import.

This feature is intended for use in business-to-business customer data solutions that utilize a Match and Merge strategy, and can be included in an Inbound Integration Endpoint used by a Match and Merge configuration. For more information on configuring an IIEP for Match and Merge, see the **IIEP - Configure Match and Merge Importer** section of the **Data Exchange** documentation.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Prerequisites

In addition to the steps outlined in the **Map Entity References** section below, the following setup is required to display the Match & Merge Importer option on the Target ID Aspect parameter:

1. The 'Matching - Merge Golden Record' component model must be valid.
2. At least one matching algorithm must be configured.
3. At least one entity-to-entity reference type must have a target of a 'Golden Record Object Type' as defined by the component model, but NOT be used in the component model as the 'Source Relation Reference Type' or the 'Merged-Into Relation Reference Type'.

The inbound file must include:

1. The ID / key (not the name) of the reference source: entity.
2. The Source Record ID key (not the name) of the reference target: entity.

Map Entity References

1. Below the Source section, use the **Map to** dropdown to select the object super type defined by the source of the reference type. For example, on an entity-to-entity, the source is entity.
2. Map the source ID or key column using the **Auto Map** button, or do so manually. For more information, see the **ID or Key - Map Inbound** topic.

Columns that include Predefined Header Text can be automatically mapped as defined in the **Inbound Map Data - Auto Map** topic.

3. In the Source section, select the target column of data and click the **Map** button.

Map Data

Source:

SourceSystem	SourceRecordID	SourceRefReco...	Name	CD_FirstName	CD_MiddleName	CD_L
SystemA	SSA_00011	SSA_00010;SSA_...	Beverly Hofstad	Beverly		Hofst
SystemA	SSA_00012	SSA_00007	Priya Rappali	Priya		Rappi
SystemA	SSA_00013	SSA_00007;SSA_...	V.M. Rappali	V	M	Rappi
SystemA	SSA_00014	SSA_00003	Amy Fowler	Amy		Fowle
SystemA	SSA_00016	SSA_00003;SSA_...	Mary Cooper	Mary		Coop
SystemA	SSA_00006	SSA_99999	Bernadette Rosten	Bernadette	Maryann	Roste
SystemA	SSA_00006	SSA_00006	Bernadette Rosten	Bernadette	Maryann	Roste

Result:

SourceRecordID...	CD_FirstName...	CD_LastName...	CD_EmailAddre...	CD_PhoneNum...	CD_Country=...	CD_St
SSA_00011	Beverly	Hofstad	Beverly.Hofstadt...	8544956433	US	NJ
SSA_00012	Priya	Rappali	Priya.Koothrappal...	6958193869	CA	ON
SSA_00013	V	Rappali	v.m.koothrappali...	6748756336	IN	DL
SSA_00014	Amy	Fowler	Amy.Fowler@tbbt...	4258232055	US	CA
SSA_00016	Mary	Cooper	Mary.Cooper@tb...	8525376759	US	TX
SSA_00006	Bernadette	Rosten	Bernadette Roste	7404849544	US	CA

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Back

Next

Finish

Cancel

4. In the **Map to** window, select the **Entity Reference** radio button, and then select the **Target ID Aspect** parameter:

- Select **Match & Merge Importer: Source Record ID** when the inbound file includes the Source Record ID for the target.

Map SourceRefRecordId to

☐ ID
☐ Name
☐ Attribute
☐ Product Reference
☐ Asset Reference
☐ Classification Reference
☒ Entity Reference

Target ID Aspect

☒ Address
☐ Affiliate Of

☐ CIC to CIC Information
☐ CMDM Company Hierarchy Link
☐ CMDM External Organization Customer Link
☐ CMDM Franchise Hierarchy Link
☐ CMDM Region Link
☐ CMDM Sales Manager Link

ID
 Key: Key 1
 Match & Merge Importer: Source Record ID

5. For the **Reference Type** parameter, choose the relevant Reference Type.
6. For the **Source System** parameter, choose the relevant Source System:
 - 'Constant' provides a constant value that can be chosen. The values include all the source system found in the system.
 - 'Column' lists the set of columns in the Excel / CSV file in question. Choose the column that contains the source system values.
 - 'Variable' lists the variables that could have been generated in a previous step.

Target ID Aspect Match & Merge Importer: Source Record ID

Reference Type MergeSourceRelation

Source System

☐ Constant (MergeGR32395)

☒ Column SourceSystem

☐ Variable

This Target ID Aspect is only relevant for usage with the Merge Golden record matching strategy.

7. Check the **Mandatory** option appropriately:
 - If **checked**, imported objects must have an attribute value in the file. If an attribute value does not exist for an object, the object is skipped and not imported.
 - If **unchecked**, all objects are imported regardless if an attribute value is specified or not.

☐ Add child to override

☒ Mandatory

Cancel OK

8. Click **OK** and the **Result** panel displays your selection of column and attribute. A validated data column is marked with a green check mark as shown below.

Result: Map to: Entity 

D_Street=CD... ✓	CD_ZipCode=... ✓	MergeSourceR... ✓
78 Communipaw ...	7304	SAP_001
11 Richmond St ...	M5H 2G4	SAP_002
lock H	110015	SAP_003

9. Complete any additional mapping.
10. Save the import configuration and add it to the relevant IIEP used for the Match & Merge use case in question.

ID or Key - Map Inbound

The STEP ID is the most common way to uniquely identify an object and it is required for updating values for existing objects. This type of mapping is described below in the **Mapping ID** section.

However, when creating new objects, and a STEP ID is not included in the input file, you can match an existing object using its key instead. This type of mapping is described below in the **Mapping Key** section.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Mapping ID

IDs must be unique and can be up to 40 characters long. Once saved in STEP, an ID can never be changed. Common setup is to avoid using special characters in the ID. When the object is a family, subcategory, or major category, etc., a common setup is to use a number for the ID and use the actual category name as the name.

When updating existing products, you must specify the object ID as it exists in STEP.

When STEP is configured to generate IDs automatically, the import can generate object IDs when no match in STEP is found to exist in the import file. In this case, a blank column in the import file will be mapped. When your object ID is an actual part number, common setup is to not configure automatically generated IDs similar to the actual part number, since it can cause confusion when identifying data.

- 1. In the Source section, select the column that contains the object ID.

Map Data

Source:

<ID>	<Name>	<Parent ID>	<Object Type Name>
18210	18210 M B	18209	Item
18212	18212 L B	18209	Item
18213	18213 MO	18209	Item
18216	18216 L O	18209	Item

Result:

Map to: Product

Auto Map

Map

Constant

Remove

Transform

Generate Profile

- 2. From the **Map to** dropdown, choose the option for the type of data being imported, and then click the **Map** button. In this example, products are being imported.

Map Data

Source:

<ID>	<Name>	<Parent ID>	<Object Type Name>
18210	18210 M B	18209	Item
18212	18212 L B	18209	Item
18213	18213 M O	18209	Item
18216	18216 L O	18209	Item

Result:

Map to: Product

- Product
- Asset
- Classification
- Entity
- Attribute

Aut... **Map** Constant Remove Transform Generate Profile

3. In the 'Map ... to' dialog, select **ID** radio button, and select the **ID** in the 'ID Aspect' dropdown list.

Map <ID> to

☒ ID

☐ Name

☐ Attribute

☐ Entity Reference

ID Aspect ID

4. Determine if the 'Mandatory' option needs to be checked and do so accordingly:
- **Check** the 'Mandatory' option only if each object being imported has an ID. If they do not all have IDs, then the object will be skipped, will not display in the Results field, and will not be imported.
 - **Unchecked** the 'Mandatory' option if any objects are missing the ID but still have other values. Doing this allows all objects to be imported, provided their object type is configured to generate IDs automatically. If there is no configuration in the 'ID Pattern', the object missing the ID will be skipped and not import.

☐ Multivalued Variable
☐ Overrides product
☐ Add child to override
☒ **Mandatory**

Cancel OK

Execution Report

- 1 Retrieval started (Mon Jan 02 14:14:43 EST 2017)
- 2 Retrieved 25088 bytes (Mon Jan 02 14:14:43 EST 2017)
- 3 Conversion started (Mon Jan 02 14:14:43 EST 2017)
- 4 Converted 4 objects (Mon Jan 02 14:14:43 EST 2017)
- 5 Logged on
- 6 Mapping started (Mon Jan 02 14:14:43 EST 2017)
- 7 Mapping completed (Mon Jan 02 14:14:43 EST 2017)
- 8 Import Started (Mon Jan 02 14:14:43 EST 2017)
- 9 Logged On
- 10 Using import mode "domain"
- 11 Starting first import pass (creating system setup objects)
- 12 Starting second import pass (importing data)
- ! 13 Row 2: The objecttype with ID 'Item' does not have auto names specified
- 14 Row 2: The product with ID ' ' was skipped

5. Click **OK** and the **Result** section shows the mapped column with the ID of the objects to be created and/or have their attributes modified in STEP.

Result: Map to: Product

ID=<ID> ✓
18210
18212
18217

6. Complete the mapping and initiate the import.

Mapping Key

Objects, parent objects, and referenced objects can also be identified using keys. Keys are automatically generated when the underlying attributes are mapped during import.

If an existing object has a key that is active and has a specified object type, it can be mapped to identify an object during import. For more information, see the **Unique Keys** section of the **System Setup / Super User Guide** documentation.

- 1. Prior to importing, verify unique keys are activated in STEP so that they are available for mapping.
- 2. In the Source table, select a column which contains unique key information, and click the **Map** button.

Map Data

Source:

<ID>	>	<Name>	>	<Parent ID>	>	<Object Type Name>	>	(SupplierPartNumber)	>
20862		20862		20859		Item		20862	
20883		20883		20859		Item		20899	
110306		abcdefghijklmnopq...		20859		Item			
111204		LED Flashlight		20859		Item			

<

>

Result:

Map to: Product

Auto Map

Map

Constant

Remove

Transform

Generate Profile

- 3. In the 'Map ... to' dialog, select the ID radio button option, and then select the respective key in the ID Aspect dropdown list. This indicates the ID of the current object is represented by the specified key.

Map <ID> to

☒ ID

☐ Name

☐ Attribute

☐ Product Classification Link

ID Aspect

Key: Supplier Part Number

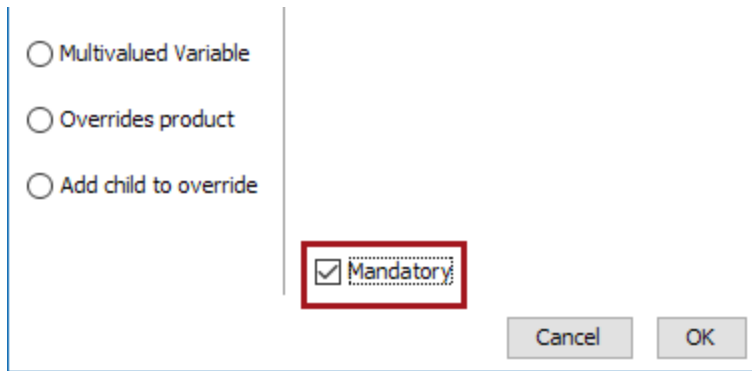
Key: Supplier Part Number

Key: Computer Supplier Part Number

Note: Keys will only be available for selection if they are active and have an object type specified.

- 4. Determine if the 'Mandatory' option needs to be checked, and do so accordingly:
 - **Check** the 'Mandatory' option only if each object being imported has an unique key. If they do not all have unique keys, then the object will be skipped, will not display in the Results field, and will not be imported.

- **Unchecked** the 'Mandatory' option if any objects are missing the unique key, but still have other values that need to be imported.



A screenshot of a dialog box with a light blue border. On the left side, there are three radio buttons stacked vertically: 'Multivalued Variable', 'Overrides product', and 'Add child to override'. To the right of these, there is a checkbox labeled 'Mandatory' which is checked. The 'Mandatory' checkbox and its label are enclosed in a red rectangular box. At the bottom right of the dialog box, there are two buttons: 'Cancel' and 'OK'.

Note: If no matching object is found (based on keys) and the import configuration allows for the creation of new objects, a new object will be created. To accomplish this, STEP IDs must be omitted from the import, and all attributes that make up the key definition must be mapped and populated on the current object. Additionally, the object type being created must use autogenerated IDs. Failing to provide the underlying key attributes will result in an error.

5. Complete the mapping and initiate the import.

Multivalued Variable - Map Inbound

When mapping to multivalued variables, a user is able to perform the same transformation on each value within a multivalued attribute, asset, classification, entity, or product. Mapping to this option enables the user to map without having to split each part of the multivalued attribute, asset, classification, entity, or product out, perform multiple mappings and transformations, and then map all of them back into one variable again.

Note that it is only possible to map to multivalued variables during the Map Data process when importing, and multivalued variables are not accessible elsewhere in STEP. It does not have a permanent place / usage outside of being a way to transform a multivalued attribute, asset, classification, entity, or product during import.

To map using a multivalued variable, follow the steps below. Note that the below example shows an Excel file being imported, though all standard inbound formats are accepted:

	A	B	C	D
1	ID	Name	Size	Color
2	26658	box 1	10;20;30	brown

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Map Multivalued Variable

1. Use Auto Map to map the ID, or map it manually. For more information, see **ID or Key - Map Inbound**.
2. Map all other data required for import. For more information, see **Inbound Map Data - Map**.
3. Map the column that holds data that needs transformations on each value to the Multivalued Variable option, and type the name in the Variable Name field.

The screenshot shows a dialog box titled "Map Size to". On the left, there is a list of radio button options: "Entity Reference", "Reference Meta-Data", "Parent", "Object Type", "Variable", and "Multivalued Variable". The "Multivalued Variable" option is selected and highlighted with a red rectangular box. A red arrow points from this selected option to the "Variable Name" field on the right, which contains the text "Multivalued Var Size".

4. Check the **Mandatory** option appropriately:

- If **checked**, imported objects must have an attribute value in the file. If an attribute value does not exist for an object, the object is skipped and not imported.
- If **unchecked**, all objects are imported regardless if an attribute value is specified or not.

☐ Add child to override

☒ Mandatory

5. Click **OK** and the **Intermediate Variables** section displays the multivalued variable to temporarily hold modified data for additional transformations.

Map Data

Source:

ID	Name	Size	Color
26658	box 1	10;20;30	brown

Intermediate Variables:

Multivalued Var Size=Size
 10|20|30

Result:

Map to: Product

ID=ID ✓	Name=Name ✓	PrimaryColor=Color ✓
26658	box 1	brown

6. Click on the multivalued variable in the Intermediate Variables section, and click the **Transform** button.
7. Select any needed transformations. The Preview pane shows how the transformations modify each part of the value in the multivalued variable, as 'multivalued parts' are designated by the **Multivalued separator** field. In this example the semi-colon (;) is used as the separator and each portion of multiplied by 2.

Specify the sequence of transformations in the table below.

Target: Multivalued Var Size=Size

☐ Mandatory

Multivalue separator: ;

Before Multivalue Split - Source: Size

Add Transformation

Value - Source: Size

Value = Value * 2

Add Transformation

Preview

Multivalued Var Size=Size ✓

20|40|60

Save Reset Cancel

- Once transformed, select the multivalued variable in the Intermediate Variables section and click on Map. Map appropriately.

Map Data

Source:

ID	Name	Size	Color
26658	box 1	10;20;30	brown

Intermediate Variables:

Multivalued Var Size=Size

20|40|60

Result:

Map to: Product

ID=ID ✓	Name=Name ✓	PrimaryColor=Color ✓	Size=Multivalued Var Size ✓
26658	box 1	brown	20 40 60

Auto Map Map Constant Remove Transform Generate Profile

Once the import is completed successfully, the attribute displays the transformed values.

> Size	123 ...	20 40 60

Mappings, multivalued variables, and transformations are saved in an import configuration for use each time the same import is run.

The transformations that can use the variable during import are the following:

- Append from multivalue source
- Append from source
- If equals source
- Math - Add source
- Math – Divided by source
- Math – Multiply by source
- Math – Subtract source
- Prepend from Source

For more on the above transformations see the **Inbound Transformation Options** topic in the **Data Exchange** documentation.

For an example of the 'Append from source' transformation being used, see the **Concatenate Data Using Variables Example** topic in the **Data Exchange** documentation.

Name - Map Inbound

Names should be used to properly to identify the object (not as a description). Avoid using special characters in the name. Unique names are not required, but do aid in identifying data. When the object is a family, subcategory, or major category, etc., a common setup is to use a number for the ID and use the actual category name as the name. When updating attribute values on existing objects in STEP, mapping the object name is not required, even if it exists in the input file.

Common setup involves mapping the object name when one of the following conditions is true:

- New objects are being imported into STEP.
- Existing names of all objects in the import file should be changed.
- The import file includes object names for a specific context and the objects are dimension dependent. For example, you sell the same product in three different markets, but identify them by a different part number in each market. Although you might typically hold a single product in STEP with a unique ID, in this case, for each market's context, you could enter the appropriate 'part number' into the Name. For more information on how to import content in different context from the same import file, see the **Dimension Point for Single Dimension Data Example** topic.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Map a Name

1. In the Source section, select the data column to map to the object name, and then click **Map**.

Source:

<ID>	<Name>	<Parent ID>	Primary Color	Secondary Color
MT 18400	Mens T PBO	18209	Black	Orange
MT 18401	Mens T PBG	18209	Blue	Green
MT 18402	Mens T PGS	18209	Green	Silver
MT 18403	Mens T PGW	18209	Gray	White
MT 18404	Mens T POY	18209	Orange	Yellow

2. In the 'Map ... to' dialog (the label of the field is used as the title of this dialog.), select the **Name** radio button.


Map <Name> to

☐ ID
☒ Name
☐ Attribute

3. Check the **Mandatory** option appropriately:
- If **checked**, imported objects must have a name in the file. If an object does not have a name, it is skipped and not imported.
 - If **unchecked**, all objects are imported, even those without a name.

☐ Add child to override

☒ Mandatory

Cancel

OK

4. Click **OK** and the **Result** panel displays that the mapped column will specify the Name of the objects in the input file.

Result:

Map to: Product

ID=<ID> ✓	Name=<Name> ✓
MT18400	Mens T PBO
MT18401	Mens T PBG
MT18402	Mens T PGS
MT18403	Mens T PGW
MT18404	Mens T POY

5. Complete the mapping and initiate the import.

Object Type - Map Inbound

This option is used whenever a new object is created or to change the existing object type of an object being imported. Object type can be changed for products, classifications, assets, and entities (with the limitations defined below).

Although these examples use Excel, any format that is valid for both inbound and outbound exchange is allowed.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Limitations for Entities

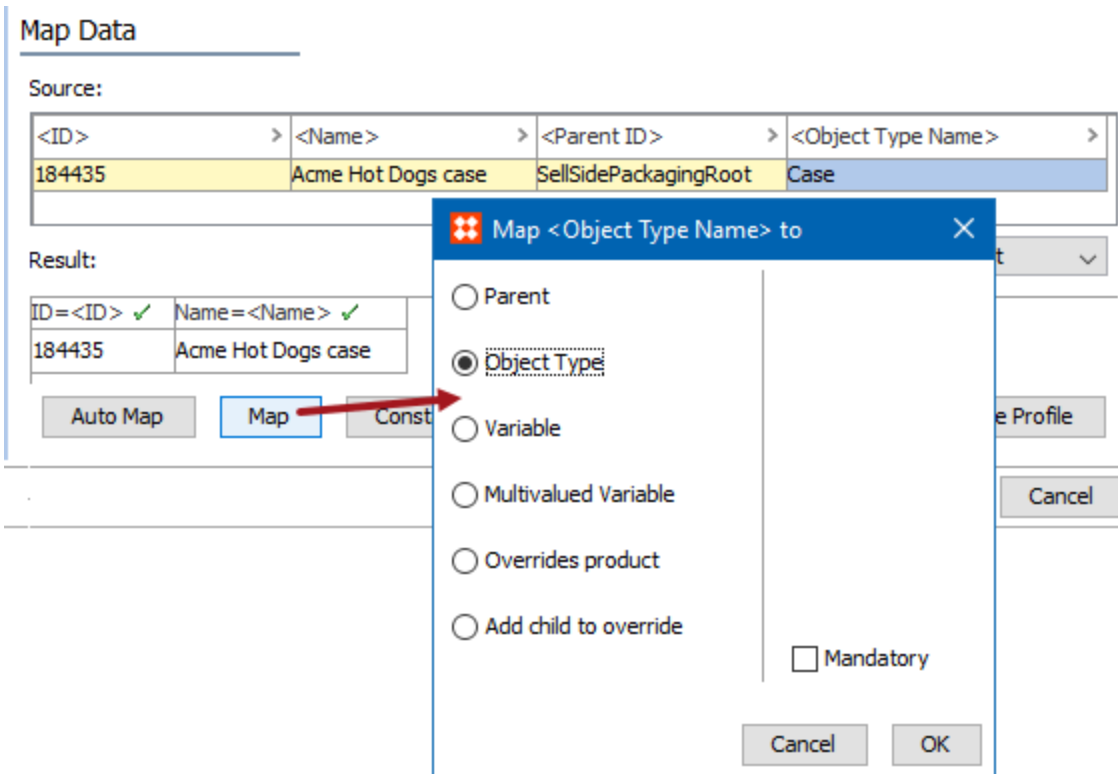
The following limitations exist for changing the object type on an entity:

- Changing a globally revisable entity object type requires single update mode. See the **Single-Update Mode** topic in **System Setup / Super User Guide** documentation.
- Changing the object type of a child entity if the parent is globally revisable is not allowed.
- Changing the object type of the entity when the Term List is different is not allowed.

Mapping

The steps for mapping to create new objects or to update existing objects are the same as follow. For prerequisites and examples of creating or updating, see the following sections below:

- **Map to Create New Items with Object Type**
 - **Map to Change the Existing Object Type**
1. Import the file to make the update in STEP and map the **ID (or key)** and **object type**. For more information on mapping ID or Key, see the **ID or Key - Map Inbound** topic.
 2. In the Source section, select the object type column, click the **Map** button.
 3. In the 'Map ... to' dialog, select the **Object Type** radio button. No additional configuration is necessary.



4. Check the **Mandatory** option appropriately:

- If **checked**, imported objects must have an attribute value in the file. If an attribute value does not exist for an object, the object is skipped and not imported.
- If **unchecked**, all objects are imported regardless if an attribute value is specified or not.

5. Click **OK** and the **Result** panel displays your selection of column and attribute. A validated data column is marked with a green check mark as shown below.

Result:

Map to: Product

ID=<ID> ✓	Name=<Name> ✓	Parent=<Parent ID> ✓	Object Type=<Object Type Name> ✓
184435	Acme Hot Dogs case	Sell Side Packaging	Case

6. Complete the mapping and initiate the import.

Map to Create New Items with Object Type

The inbound field mapped to object type defines the object type being imported and also defines what parent type is allowed.

Prerequisites

The following conditions must be met in order to set an object type during import:

1. The inbound file must include: object ID (or key), parent ID, and object type.
2. The object type in the inbound file must already exist in STEP.
3. When the parent ID is not in the inbound file, and the object type is not valid, an error is displayed in the background process.

Inbound Data

An import file includes multiple new objects, the parent ID, and the desired object type.

	A	B	C	D
1	<ID>	<Name>	<Parent ID>	<Object Type Name>
2	Create1ID	Create1	20861	Item
3	Create2ID	Create2	20861	Item
4	Create3ID	Create3	20861	Item
5	Create4ID	Create4	20861	Item

Once all columns have been mapped, the Result section shows the objects that will be created.

Map Data

Source:

<ID>	>	<Name>	>	<Parent ID>	>	<Object Type Name>	>
Create1ID		Create1		20861		Item	
Create2ID		Create2		20861		Item	
Create3ID		Create3		20861		Item	
Create4ID		Create4		20861		Item	

Result:

Map to: Product

ID=<ID> ✓	Name=<Name> ✓	Parent=<Parent ID> ✓	Object Type=<Object Type Name> ✓
Create1ID	Create1	Flashlights SalesItems	Item
Create2ID	Create2	Flashlights SalesItems	Item
Create3ID	Create3	Flashlights SalesItems	Item
Create4ID	Create4	Flashlights SalesItems	Item

Auto Map
Map
Constant
Remove
Transform
Generate Profile

Map to Change the Existing Object Type

During import, it is possible to change the object type of one or more existing objects. This is useful during data migration, data cleanup activities, for example when the wrong object type was originally assigned.

Important: When changing object types, common setup is to keep the import simple and update changes to attributes separately. Since not all attributes are valid for all object types, import errors will occur if attempting to update invalid attributes.

Prerequisites

The following conditions must be met in order to change an object type during import:

- 1. The inbound file must include the object's ID (or unique key) and the ID of the new object type.
- 2. The existing objects being modified must be updated to one of the valid options available on the Object Type parameter dropdown (as shown in the image below).
- 3. The parent node must allow children of the new object type being selected or an error is displayed in the background process.
- 4. The attribute validations must be the same between the original object type and the new object type or an error is displayed in the background process.
- 5. The dimension dependencies must be the same between the original object type and the new object type or an error is displayed in the background process.

Existing Data

Consider an object which has the valid object type options of Case, Pack, and Pallet (as shown in the image below). In our example steps below, assume that the current selection of 'pack' should be changed to 'case.'

Acme Hot Dogs pk rev.0.4 - Prod

Images & DocumentsCommercialTablesProof ViewStatusState Log

ProductReferences

Description

Name	>>	Value
ID		184435
Name		Acme Hot Dogs pk
Object Type		Pack
Revision		Case
Approved		Pack
Translation		Pallet
		Not translated

The import file is created by first exporting the objects that need to have a new object type.

	A	B	C	D
1	<ID>	<Name>	<Parent ID>	<Object Type Name>
2	184435	Acme Hot Dogs pk	SellSidePackagingRoot	Pack

Inbound Data

Update the export file to include the data needed for import:

- Change the name to reflect the object type change
- Change the object type
- If necessary, include a new Parent ID in the file to also reparent the object (change the location in the hierarchy). For this example, we left the object in its current hierarchy location.

	A	B	C	D
1	<ID>	<Name>	<Parent ID>	<Object Type Name>
2	184435	Acme Hot Dogs case	SellSidePackagingRoot	Case

After a successful import, refresh the object and review it in the Tree. The original object type of 'Pack' has been changed to 'Case,' and the name has been updated.

Acme Hot Dogs case rev.0.5 - Product		
Images & Documents	Commercial	Tables
Proof View	Status	State Log
Tasks		
Product	References	R
Description		
Name	>	Value
ID	>	184435
Name	>	Acme Hot Dogs case
Object Type	>	Case
Revision	>	0.5 Last edited by USERJ on Mon Feb 13 16:55:51 EST 2017
Approved	>	 Last Approved on Wed Jan 04 17:40:29 EST 2017

Overrides Product - Map Inbound

Product overrides are alternate versions of products and product families that may have differing values, references, links, and structures. Attributes and values applied to the product family are inherited to the product override and can be replaced with local values and references on the product override.

The Overrides Product option is only available for the Product object super type.

For more information on Product Overrides, see the **Product Override** topic in the Getting Started / User Guide documentation or the **Override Scenario** use case topic in the **Calculated Attributes** section of the **System Setup / Super User Guide** documentation.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Prerequisites

- Before importing a product override, it must first be setup in System Setup. See the **Product-Override Object Types and Commercial Object Types** topic in the **System Setup / Super User Guide** documentation.
- Verify the import file includes the following fields:
 - ID:** The ID of the new override product to be imported
 - Overrides:** The ID of the product in STEP that will get an override version of itself
 - Object Type ID:** The ID of the object type in System Setup that the override product will be
 - Parent ID:** The ID of the parent the override product will live under

While the example shows an Excel import file, all inbound formats are allowed.

	A	B	C	D
1	<ID>	Overrides	<Object Type ID>	<Parent ID>
2	115606	114852	std.AdaptorType	ProductOverridesRoot

Map Overrides Product

- Below the Source section, use the **Map to** dropdown to select the **Product** object super type.
- Map the ID or key column using the Auto Map button, or manually. For more information, see the **ID or Key - Map Inbound** topic.

Columns that include Predefined Header Text can be automatically mapped as defined in the **Inbound Map Data - Auto Map** topic. For this sample file, this includes the <ID>, <Object Type ID>, and <Parent ID> columns.

- In the Source section, select the Overrides column of data and click the **Map** button.

Map Data

Source:

<ID>	Overrides	<Object Type ID>	<Parent ID>
115606	114852	std.AdaptorType	ProductOverridesRoot

Result:

ID=<ID> ✓	Parent=<Parent ID> ✓	Object Type=<Object Type ID> ✓
115606	Product Overrides	Product-override

Map to: Product ▼

Auto Map Map Constant Remove Transform Generate Profile

Note: The Object Type ID in System Setup is standard in STEP for Product Override and thus displays as 'std.AdaptorType.' However, once mapped, STEP automatically changes to the object type that will be displayed in Tree after import.

4. In the 'Map ... to' dialog, select the **Overrides product** radio button.

☐ Variable
☐ Multivalued Variable
☒ Overrides product
☐ Add child to override

☐ Mandatory

Cancel OK

5. Check the **Mandatory** option appropriately:

- If **checked**, imported objects must have an attribute value in the file. If an attribute value does not exist for an object, the object is skipped and not imported.
- If **unchecked**, all objects are imported regardless if an attribute value is specified or not.

6. Click **OK** and the **Result** panel displays your selection of column and attribute. A validated data column is marked with a green check mark as shown below.

Result:

Map to: Product ▼

ID=<ID> ✓	Parent=<Parent ID> ✓	Object Type=<Object Type ID> ✓	OverridesProductID=Overrides ✓
115606	Product Overrides	Product-override	114852

7. Complete the mapping and initiate the import.

Example

Prior to import, the Product Overrides node in Tree displays a single override.

Tree

Primary Product Hierarchy

Products

Discontinued Products

Product Overrides

(115607) → Flashlights SalesItems

Level 1

Packaging

GDSN Products

Product Overrides rev.0.1

Product

Sub Products

References

Referenced By

Description

Name	Value
ID	ProductOverridesRoot
Name	Product Overrides
Object Type	Product Overrides

After a successful import, a new product override object is displayed and the overridden product is shown.

Tree

Primary Product Hierarchy

Products

Discontinued Products

Product Overrides

(115606) → LED Pocket Flashlight

(115607) → Flashlights SalesItems

Level 1

(115606) → LED Pocket Flashlight rev.0

Product

Sub Products

References

Referenced By

Images & Documents

C

Default InDesign template	
Overridden Product	LED Pocket Flashlight (114852)
Completeness Score	0.23
GetOverriddenByProducts	This product is overridden by 0 product override(s):

Parent - Map Inbound

The parent column indicates where to locate, or relocate, the imported object(s). When updating attribute values for existing objects, mapping the parent column is not required, even if it exists in the input file. Common setup involves mapping the parent column when one of the following conditions is true:

- new objects are being imported into STEP
- existing objects within the hierarchy need to be relocated

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Map Parent

1. In the Source section, select the column of data that includes valid STEP IDs for parents, and click the **Map** button.

Source:

<ID>	>	<Name>	>	<Parent ID>	>	Primary Color	>	Secondary Color	>
MT18400		Mens T PBO		18209		Black		Orange	
MT18401		Mens T PBG		18209		Blue		Green	
MT18402		Mens T PGS		18209		Green		Silver	
MT18403		Mens T PGW		18209		Gray		White	
MT18404		Mens T POY		18209		Orange		Yellow	

2. In the 'Map ... to' dialog (the label of the field is used as the title of this dialog.), select the **Parent** radio button. Select the 'Parent ID' option from the **Parent ID Aspect** dropdown list. To use a Key, instead of the Parent ID, select the appropriate option from the Parent ID Aspect dropdown.

☐ Entity Reference
☐ Reference Meta-Data
☒ Parent
☐ Object Type

Parent ID Aspect

Parent ID
Parent ID
Key: Key 1
Key: Supplier Part Number
Key: Computer Supplier Part Number

3. Check the **Mandatory** option appropriately:
 - If **checked**, imported objects must have a parent ID in the file. If a parent ID does not exist for an object, the object is skipped and not imported.
 - If **unchecked**, all objects are imported regardless if a parent ID is specified or not.

☐ Add child to override

☒ Mandatory

Cancel

OK

4. Click **OK** and the **Result** panel displays your selection.

Result:

Map to: Product

ID = <ID> ✓	Name = <Name> ✓	Parent = <Parent ID> ✓
MT 18400	Mens T PBO	Cotton T-Shirts
MT 18401	Mens T PBG	Cotton T-Shirts
MT 18402	Mens T PGS	Cotton T-Shirts
MT 18403	Mens T PGW	Cotton T-Shirts
MT 18404	Mens T POY	Cotton T-Shirts

Note that if valid STEP IDs were not found in the first 200 rows of the file, the Result panel will highlight in red color, and a tool tip will show when hovered over the row, that the object was not found in STEP.

Parent = <Parent ID> ⛔
111204
20899
668998643
3585275

Product with ID "668998643" not found

5. Complete the mapping and initiate the import.

Product Classification Links - Map Inbound

A product in the inbound file can be linked to one or more classifications in STEP using the valid product-to-classification link types.

Prerequisites

The inbound file must include:

1. The ID / Key of the product.
2. The ID / Key of the classification.
3. A separate field for each classification ID when linking the same product to multiple classifications.

Although an Excel file is shown in the examples, any inbound format is allowed.

	A	B	C	D	E	F
1	<ID>	<Name>	<Parent ID>	<Object Type ID>	Link 1	Link 2
2	22624	L23-RP41000	ItemFolder_6805	Item	SuppliesAllProducts	
3	8046	RP4100	ItemFolder_6805	Item	SuppliesAllProducts	SAPProducts

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Map Product Classification

1. Below the Source section, use the **Map to** dropdown to select the **Product** object super type.
2. Map the ID or key column using the Auto Map button, or manually. For more information, see the **ID or Key - Map Inbound** topic.

Columns that include Predefined Header Text can be automatically mapped as defined in the **Inbound Map Data - Auto Map** topic. For this sample file, this includes the <ID>, <Name>, <Object Type ID>, and <Parent ID> columns.

3. In the Source section, select the classification ID column of data and click the **Map** button.

Map Data

Source:

<ID>	<Name>	<Parent ID>	<Object Type ID>	Link 1	Link 2
22624	L23-RP41000	ItemFolder_6805	Item	SuppliesAllProducts	
8046	RP4100	ItemFolder_6805	Item	SuppliesAllProducts	SAPProducts

Result:

Map to: Product

ID = <ID> ✓	Name = <Name> ✓	Parent = <Parent ID> ✓	Object Type = <Object Type ID> ✓
22624	L23-RP41000	Tire Care Items	Item
8046	RP4100	Tire Care Items	Item

4. In the 'Map ... to' dialog, select the **Product Classification Link** radio button.

For the **Classification ID Aspect** dropdown, select:

- **ID** if the file includes the object ID
- **Key** if the file includes a unique key. Only activated keys are displayed.

Map Supplier Link to

☐ Name
☐ Attribute
☒ **Product Classification Link**
☐ Product Reference

Classification ID Aspect

ID
 ID
 Key: Part_Number

Note: Each classification object type can have only one product-to-classification link type assigned. STEP identifies the classification by the ID / key and applies the single valid link type available for the mapped product.

5. Check the **Mandatory** option appropriately:

- If **checked**, imported objects must have an attribute value in the file. If an attribute value does not exist for an object, the object is skipped and not imported.
- If **unchecked**, all objects are imported regardless if an attribute value is specified or not.

☐ Add child to override

☒ **Mandatory**

- Map additional product-to-classification links if necessary.
- Click **OK** and the **Result** panel displays your selection of column and attribute. A validated data column is marked with a green check mark as shown below.

Result:

Map to: Product

ID=<ID> ✓	Name=<Name> ✓	Parent=<Parent ID> ✓	Object Type=<Object Type ID> ✓	Classification=Link 1 ✓	Classification=Link 2 ✓
22624	L23-RP41000	Tire Care Items	Item	SASProducts	
8046	RP4100	Tire Care Items	Item	SASProducts SASProducts	

Note: Although the data is not displayed in the Results panel for links after the first one, it is imported as demonstrated in the example below.

- Complete the mapping and initiate the import.

Example

Prior to import, the products show no product-to-classification links in the editor.

Tree

- Automotive
 - Parts & Accessories
 - Tire Care
 - Tire Care Items
 - L23-RP4100
 - RP4100
 - Tire Care SalesItems
- Building Products

No Title

Products References Referenced By

Item References, Classification

Reference Type All

Source	Reference Type	Target
L23-RP4100	Supplier Link	
RP4100	Supplier Link	

After a successful import, new links are displayed on the product's References tab.

Tree

- Automotive
 - Parts & Accessories
 - Tire Care
 - Tire Care Items
 - L23-RP41000
 - RP4100
 - Tire Care SalesItems
- Building Products
- Electrical & Electronics

No Title

Products References Referenced By

Item References, Classification

Reference Type: Supplier Link

Source	Reference Type	Target
L23-RP41000	Supplier Link	Suppliers/Supplies All/SASProducts
RP4100	Supplier Link	Suppliers/Supplies All/SASProducts

Product Reference - Map Inbound

You can reference products, classifications, or entities IDs / Unique keys listed in one column of the input file to another column of products IDs / Unique keys in that file, and make a product reference between them in STEP. When you make this selection, you must specify which reference type to use and the reference type should be valid for the object types. The available reference types are based on the product reference types that exist in STEP.

When using the Product Reference option, consider the following information:

- When mapping references, you must have at least two columns in your input file. One with the ID / unique key of the source object, and the other with the ID / unique key of the target object. When you map these columns, you can specify the reference type that should be used to link the objects.
- The reference types listed in the wizard are the list of existing reference types. You cannot specify or create new reference types at this point (during the import).
- When making product-to-product references, you must have at least two columns in your input file. One with the ID of the source product, and the other with the ID of the target product. When you map these columns, you can specify the reference type that should be used to link the products.
- When making product-to-product references, you cannot specify the product reference type in the input file. You must select the product reference type in the wizard screen. The product reference type should be valid for the object types in the input file. In a single load operation, you can only make one type of product-to-product reference. You can, however, load the same file again and specify a different product reference type.
- For product-to-product and product-to-asset references, the list of reference types listed in the wizard is the list of existing reference types. You cannot specify or create new reference types at this point.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Prerequisites

If the objects to be imported or the Target to be mapped is identified via a unique key, you must verify that the Key required is available and activated. For more information, see the **Activating and Deactivating Keys** topic of the **System Setup / Super User Guide** documentation.

The following example assumes that the 'Allow multiple references parameter' on the Reference Type is set to Yes, which means that multiple products can be linked to the same product reference. For more information, see the **Multiple References for the Same Objects** topic within the **System Setup / Super User Guide** documentation.

Map Product Reference

1. In the Source window, select the ID or Key column and map it. For more information see the **ID or Key - Map Inbound** documentation.
2. Select the reference target column in the Source section and then click the 'Map' button.

Map Data

Source:

<ID>	>Target
134854	20695
134887	20695
179682	20695
129666	20695
191337	20695

Result:

Map to: Product

3. In the 'Map ... to' dialog, the label of the import field is used as the title. Make the following selections:

- Select the **Product Reference** radio button option.
- In the 'Target ID Aspect' dropdown list, select either ID or the appropriate key. See the **Prerequisites** section above if mapping by key.
- Select the desired Reference type.

Map Target to

☐ ID
☐ Name
☐ Attribute
☐ Product Classification Link
☒ **Product Reference**

Target ID Aspect

☐ (AccessoryOn
☐ (AccessoryKe
☐ (AlternateFamily)
☐ (AlternateSupplierItem)
☒ (BillOfMaterials)
☐ (CaseToChild)

ID
 ID
 Key: Simple1
 Key: Supplier Part Number
 Key: Key 3

4. Check the **Mandatory** option appropriately:

- If **checked**, imported objects must have a Target ID in the file. If a Target ID does not exist for an object, the object is skipped and not imported.
- If **unchecked**, all objects are imported regardless if a Target ID is specified or not.

☐ Add child to override

☒ Mandatory

Cancel

OK

5. Click the **OK** button to display the mapping in the Result section.

Map Data

Source:

<ID>	> (BillOfMaterials) Product Reference >
134854	20695
134887	20695
179682	20695
129666	20695
191337	

<

>

Result:

Map to: Product

ID=<ID> ✓	BillOfMaterials ProductReference=(BillOfMateri... ✓
134854	Red Pumps
134887	Red Pumps
179682	Red Pumps
129666	Red Pumps
191337	

Auto...

Map

Constant

Remove

Transform

Generate Profile

6. If needed, map the same Target column again to a different product reference type.

Result:

Map to: Product

ID=<ID> ✓	BillOfMaterials ProductRefer... ✓	AlternateSupplierItem ProductReference=(Bill... ✓
134854	Red Pumps	Red Pumps
134887	Red Pumps	Red Pumps
179682	Red Pumps	Red Pumps
129666	Red Pumps	Red Pumps
191337		

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Reference Meta-Data - Map Inbound

Attribute values can be imported as metadata for all reference types or link types.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

For example, viewing the References tab on the selected products shows the 'Image Display Sequence' metadata attribute on the 'Product Images' reference type.

Tree

T-Shirts Family

123854 B

123855 O

124460 B

T-Shirts Sales Items

Head Wear

Footwear

Safety

Hardware

No Title

ProductsReferencesReferenced By

Image References

Reference TypeProduct Images

Source	Reference Type	Target	Image Display Sequence
> 123854 B	+ Product Images		
> 123855 O	+ Product Images		
> 124460 B	+ Product Images		

To import metadata attribute values into the 'Image Display Sequence' metadata attribute, an input file must contain columns holding the metadata attribute values, the Product IDs, and the referenced target asset IDs.

Map Reference Metadata

- Map the source ID or key column using the Auto Map button, or manually. For more information, see the **ID or Key - Map Inbound** topic.

Columns that include Predefined Header Text can be automatically mapped as defined in the **Inbound Map Data - Auto Map** topic. For the sample below, this includes the <ID> column.
- In the Source section, select the target column of data and click the **Map** button.

Map Data

Source:

<ID>	Asset Target	ImageDisplaySequence
123854	20584	3
123855	112806	1
124460	20586	2

Result:

Map to: Product

ID = <ID> ✓

123854

123855

124460

Auto Map Map Constant Remove Transform Generate Profile

- In the 'Map ... to' dialog (below), select the **Asset Reference** radio button, and then select the reference type. Only existing reference types are listed. You cannot specify or create new reference types during the import.
- For the **Target ID Aspect** parameter:
 - Select **ID** when the inbound file includes the ID for the target.
 - Select the **Key** from the dropdown when the inbound file includes a key for the target.

Map Asset Target to

☐ ID

☐ Name

☐ Attribute

☐ Product Classification Link

☐ Product Reference

☒ Asset Reference

☐ Classification Reference

Target ID Aspect ID

☐ ADA

☐ Data Sheet

☐ Installation Manual

☐ MSDS

☐ Owners Manual

☐ PDF

☒ Product Images

- Click **OK** and the **Result** panel displays the selection of reference asset target. A validated data column is marked with a green check mark as shown below.
- In the Source section, select the metadata attribute column of data and click the **Map** button.

Map Data

Source:

<ID>	Asset Target	ImageDisplaySequence
123854	20584	3
123855	112806	1
124460	20586	2

Result:

Map to: Product

ID = <ID> ✓	ProductImage AssetReference=Asset Target ✓
123854	Hanes Blue
123855	HanesGreenT
124460	Hanes Orange

Auto Map

Map

Constant

Remove

Transform

Generate Profile

7. In the 'Map ... to' window, select the **Reference Meta-Data** radio button to display the additional parameters.

Map ImageDisplaySequence to

☐ ID

☐ Name

☐ Attribute

☐ Product Classification Link

☐ Product Reference

☐ Asset Reference

☐ Classification Reference

☐ Entity Reference

☒ Reference Meta-Data

Reference Type

Product Images (ProductImage)

...

Attribute

Image Display Sequence (ImageDisplaySequence)

...

Target ID Column

Asset Target

▼

Target ID Aspect

ID

▼

Match value ID in LOV

☐

- Complete the additional parameters as follows:
 - For **Reference Type**, select the type of reference that should receive the metadata values. In this example, the asset reference type 'Product Images' is selected.
 - For **Attribute**, select the attribute that should receive the metadata values. In this example, the attribute 'Image Display Sequence' is selected.
 - For **Target ID Column**, from the dropdown, select the column in the import file that includes the ID of the reference target. In this example, the 'Asset Target' column in the input file is selected.

- For **Target ID Aspect**, from the dropdown, select **ID** when the inbound file includes the ID for the target; select the **Key** when the inbound file includes a key for the target.
- For **Match value ID in LOV**, if the metadata attribute holds an LOV with value IDs, the import file can also hold the value IDs. If checked, the ID of the LOV must be included in the inbound file; if unchecked, the LOV ID is not required.

9. Check the **Mandatory** option appropriately:

- If **checked**, imported objects must have an attribute value in the file. If an attribute value does not exist for an object, the object is skipped and not imported.
- If **unchecked**, all objects are imported regardless if an attribute value is specified or not.

A dialog box with a blue border. On the left, there is a radio button labeled "Add child to override". On the right, there is a checked checkbox labeled "Mandatory". At the bottom right, there are two buttons: "Cancel" and "OK".

10. Click **OK** and the **Result** panel displays your selection of reference target. A validated data column is marked with a green check mark as shown below.

The "Map Data" panel shows a "Source" table and a "Result" table. The "Source" table has columns for ID, Asset Target, and ImageDisplaySequence. The "Result" table shows the mapped data with a "Map to:" dropdown set to "Product". Below the tables are buttons for "Auto Map", "Map", "Constant", "Remove", "Transform", and "Generate Profile".

Source:		
<ID>	Asset Target	ImageDisplaySequence
123854	20584	3
123855	112806	1
124460	20586	2

Result:			Map to: Product
ID = <ID>	ProductImage AssetReference = Asset Target	ProductImage.ImageDisplaySequence (1) = ImageDisplaySequence	
123854	Hanes Blue Hanes Blue	3	
123855	HanesGreenT HanesGreenT	1	
124460	Hanes Orange Hanes Orange	2	

11. Complete any additional mapping.

12. Initiate the import.

Results

In the Tree, selecting the updated products and displaying the 'Product Images' reference type shows the new referenced objects as well as the metadata on the reference.

Tree

T-Shirts Family

123854 B

123855 O

124460 B

T-Shirts Sales Items

Head Wear

Footwear

Safety

Hardware

No Title

ProductsReferencesReferenced By

Image References

Reference TypeProduct Images

Source	Reference Type	Target	Image Display Sequence
123854 B	Product Images	Hanes Blue	3
123855 O	Product Images	HanesGreenT	1
124460 B	Product Images	Hanes Orange	2

Variable - Map Inbound

A variable temporarily holds modified data for additional transformation. Variables are only available when importing a format that requires mapping, and are not accessible or stored elsewhere in STEP. They are used to include specified math operations, or to append values to the input data before importing the new value into STEP.

For example, consider a file that includes values for a multivalued attribute, but each value is in a column of its own. A variable can hold the values, and a separator character, so that upon import, the attribute is updated with a true multivalued value.

The Variable option is available for use with assets, attributes, classifications, entities, and products.

Variables, transformations, and mappings are included when saving an import configuration. This makes it possible to use them again without having to recreate them. For more information about saving an import configuration, see the **Running a Data Import** topic in the **Import Manager** documentation. or the **IIEP - Configure STEP Importer Processing Engine** topic in the **IIEP** documentation.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Map a Variable

1. Below the Source section, use the **Map to** dropdown to select the object super type required by the import file.
2. In the Source section, select the column of data that you want to map to the variable and click the **Map** button.

Map Data

Source:

<ID>	Color 1	Color 2
20861	Black	White
20862	Red	Yellow

Result:

ID = <ID> ✓

20861

20862

Map to: Product ▼

Auto Map Map Constant Remove Transform Generate Profile

3. In the 'Map ... to' dialog, select the **Variable** radio button and provide text to describe the purpose of the variable.

☐ Entity Reference
☐ Reference Meta-Data
☐ Parent
☐ Object Type
☒ Variable

Variable Name

- Click **OK** and the **Intermediate Variables** section is displayed on the Map Data dialog. The originally selected Source column is shown as mapped to the variable text provided. The data is now available for multiple transformations.

Map Data

Source:

<ID>	Color 1	Color 2
20861	Black	White
20862	Red	Yellow

Intermediate Variables:

Multivalued Color = Color 1
Black
Red

Result:

Map to: Product

ID = <ID> ✓
20861
20862

Auto Map

Map

Constant

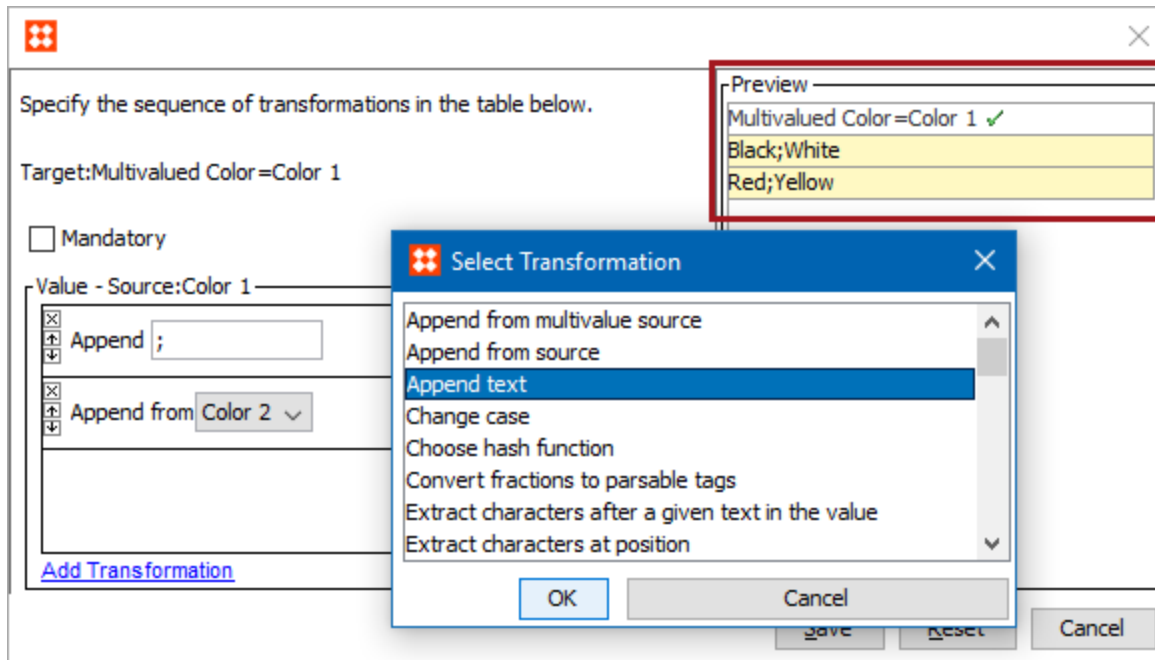
Remove

Transform

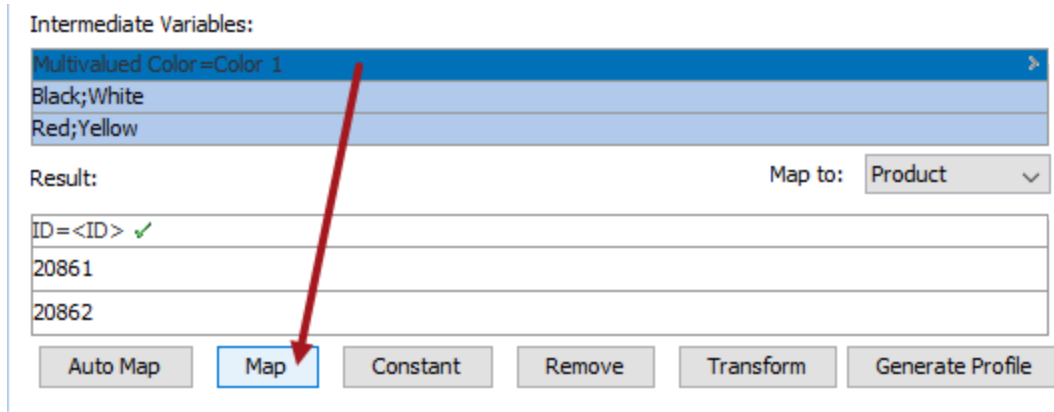
Generate Profile

- Select the intermediate variable column and click the **Transform** button. The transformations dialog is displayed. For more on the transformations, see the **Inbound Map Data - Transform** topic in the **Data Exchange** documentation.
- Click the **Add Transformation** link to display the Select Transformation dialog.
- Select the necessary transformation and provide any additional required information. For this exercise, we use **Append Text** and **Append from Source** as shown below.

Continue to add all required transformations until the **Preview** panel shows the desired results.



8. Click **Save** on the transformations button to display the Preview results in the Intermediate Variables section.
9. Select the intermediate variable column and click the **Map** button.



10. In the 'Map ... to' dialog, select the **Attribute** radio button and select the attribute that will hold the transformed values, setting the Mandatory option as necessary. In this exercise, we select the Color attribute. For more information on mapping to attributes, see the **Attribute - Map Inbound** topic.
11. Click **OK** and the **Result** panel displays your selection of column (the intermediate variable Multivalued Color) and attribute (Color). A validated data column is marked with a green check mark as shown below.

Intermediate Variables:

Multivalued Color=Color 1 >

Black;White

Red;Yellow

Result: Map to: Product

ID=<ID> ✓	Color=Multivalued Color ✓
20861	Black White
20862	Red Yellow

12. Complete the mapping and initiate the import.

Example

Consider two objects with a multivalued attribute named Color.

Tree

Hardware

Tools

Task Lighting

Flashlights

Flashlights Items

20861

20862

Products

References

Referenced By

View: Color

ID	Color
> 20861	
> 20862	

The import file includes two colors for each object, and the variable is used to hold the transformation that adds both colors to the same attribute.

Tree

Hardware

Tools

Task Lighting

Flashlights

Flashlights Items

20861

20862

20883

Products

References

Referenced By

View: Color

ID	Color
> 20861	Black White
> 20862	Red Yellow

For an example of the 'Append from source' transformation being used, see the **Concatenate Data Using Variables Example** topic in the **Data Exchange** documentation.

Inbound Map Data - Map To

The 'Map to' field allows you to import objects of the types product, asset, classification, entity, and attribute. Setting this field determines the options available when you click the Map button.

Map Data

Source:

<Name>	>	<Parent ID>	>	Primary Color	>	Secondary Color	>
Mens T PBO		18209		Black		Orange	
Mens T PBG		18209		Blue		Green	

Result:

Map to: Product

Name = <Name> ✓	Parent = <Parent ID> ✓	PrimaryColor = Primar... ✓	SecondaryColor = Se... ✓
Mens T PBO	Cotton T-Shirts	Black	Orange
Mens T PBG	Cotton T-Shirts	Blue	Green

Auto Map

Map

Constant

Remove

Transform

Generate Profile

For more information about the additional import steps, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Mapping Based on Object Type

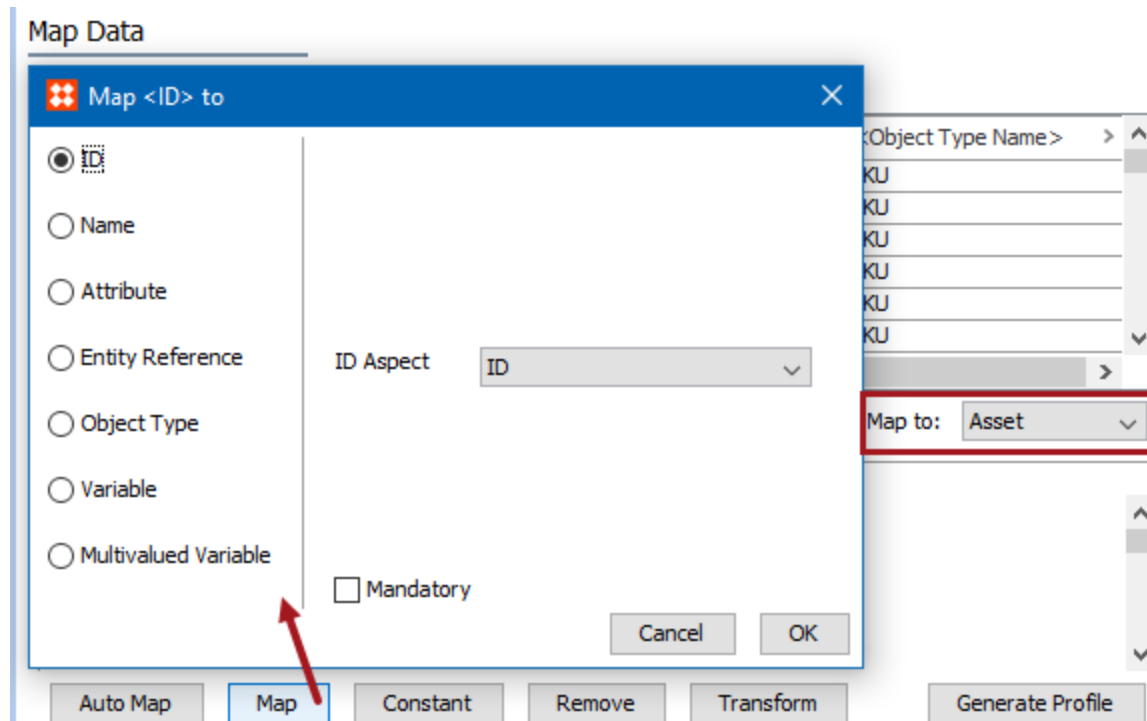
The following object types can be mapped:

- Asset Object Type - Map To Inbound
- Attribute Object Type - Map To Inbound
- Classification Object Type - Map To Inbound
- Entity Object Type - Map To Inbound
- Product Object Type - Map To Inbound

Asset Object Type - Map To Inbound

For inbound data, after selecting the Asset super type in the **Map to** dropdown, click the **Map** button to import information for assets. Asset content is not included in this data import, but can be added separately using the **Digital Asset Exchange** documentation.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.



For more information, see the **Map Inbound** topics on the available options:

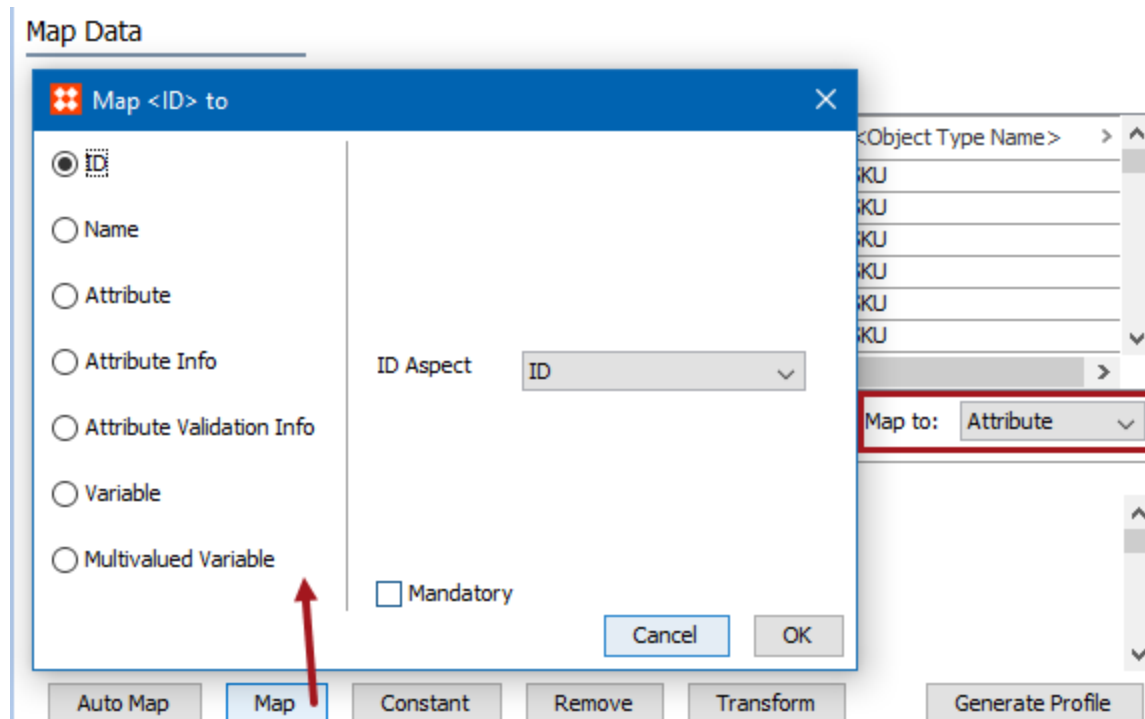
- ID or Key
- Name
- Attribute
- Entity Reference
- Object Type
- Variable
- Multivalued Variable

Attribute Object Type - Map To Inbound

For inbound data, after selecting the Attribute super type in the **Map to** dropdown, click the **Map** button to import information for attributes.

Note: When mapping to an attribute, it is common setup to only change the object type of a list of objects when those objects are all at the same level in the STEP hierarchy.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.



For more information, see the **Map Inbound** topics on the available options:

- ID or Key
- Name
- Attribute
- Attribute Info
- Attribute Validation Info
- Variable
- Multivalued Variable

Classification Object Type - Map To Inbound

For inbound data, after selecting the Classification super type in the **Map to** dropdown, click the **Map** button to import information for classifications.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Map Data

Map <ID> to

☒ ID
☐ Name
☐ Attribute
☐ Product Reference
☐ Asset Reference
☐ Classification Reference
☐ Entity Reference
☐ Reference Meta-Data
☐ Parent
☐ Object Type
☐ Variable
☐ Multivalued Variable

ID Aspect: ID

☐ Mandatory

Cancel OK

Map to: Classification

Auto Map Map Constant Remove Transform Generate Profile

For more information, see the **Map Inbound** topics on the available options:

- ID or Key
- Name
- Attribute
- Product Reference
- Asset Reference

- Classification Reference
- Entity Reference
- Reference Meta-Data
- Parent
- Object Type
- Variable
- Multivalued Variable

Entity Object Type - Map To Inbound

For inbound data, after selecting the Entity super type in the **Map to** dropdown, click the **Map** button to import information for entities.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

For more information, see the **Map Inbound** topics on the available options:

- ID or Key
- Name
- Attribute
- Product Reference

- Asset Reference
- Classification Reference
- Entity Reference
- Reference Meta-Data
- Parent
- Object Type
- Variable
- Multivalued Variable
- Data Container Attribute

Product Object Type - Map To Inbound

For inbound data, after selecting the Product super type in the **Map to** dropdown, click the **Map** button to import information for products.

The mapping instructions are the same for both the Import Manager tool and the IIEP tool. For details about starting an import, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Map Data

Map <ID> to

- ☒ ID
- ☐ Name
- ☐ Attribute
- ☐ Product Classification Link
- ☐ Product Reference
- ☐ Asset Reference
- ☐ Classification Reference
- ☐ Entity Reference
- ☐ Reference Meta-Data
- ☐ Parent
- ☐ Object Type
- ☐ Variable
- ☐ Multivalued Variable
- ☐ Overrides product
- ☐ Add child to override

ID Aspect: ID

☐ Mandatory

Cancel OK

Map

For more information, see the **Map Inbound** topics on the available options:

- ID or Key
- Name
- Attribute
- Product Classification Links
- Product Reference
- Asset Reference
- Classification Reference
- Entity Reference
- Reference Meta-Data
- Parent
- Object Type
- Variable
- Multivalued Variable
- Overrides Product
- Add child to override

Inbound Map Data - Remove

If the Result panel displays a column that you do not want to import, you can remove it.

Map Data

Source:

<Name>	>	<Parent ID>	>	Primary Color	>	Secondary Color	>
Mens T PBO		18209		Black		Orange	
Mens T PBG		18209		Blue		Green	

Result:

Map to: Product

Name = <Name> ✓	Parent = <Parent ID> ✓	PrimaryColor = Primar... ✓	SecondaryColor = Se... ✓
Mens T PBO	Cotton T-Shirts	Black	Orange
Mens T PBG	Cotton T-Shirts	Blue	Green

For more information about the additional import steps, see [Creating a Data Import](#) or [Creating an Inbound Integration Endpoint](#).

Remove a Mapped Column

1. In the **Result** panel, select the mapped column to remove.

Result:

Map to: Product

ID = <ID> ✓	Name = <Name> ✓	Parent = <Parent ID> ✓	PrimaryColor = Primary Color ✓	ManufacturerName = Acme Clot... ✓	SecondaryColor = Primary Color ✓
MT18400	Mens T PBO	Cotton T-Shirts	Black	Acme Clothing	Black
MT18401	Mens T PBG	Cotton T-Shirts	Blue	Acme Clothing	Blue
MT18402	Mens T PGS	Cotton T-Shirts	Green	Acme Clothing	Green
MT18403	Mens T PGW	Cotton T-Shirts	Gray	Acme Clothing	Gray
MT18404	Mens T POY	Cotton T-Shirts	Orange	Acme Clothing	Orange

2. Click the **Remove** button.

The **Result** panel no longer displays the column, and the removed data will not be imported.

Result:

Map to: Product ▼

ID=<ID> ✓	Name=<Name> ✓	Parent=<Parent ID> ✓	PrimaryColor=Primary Color ✓	ManufacturerName=Acme ... ✓
MT18400	Mens T PBO	Cotton T-Shirts	Black	Acme Clothing
MT18401	Mens T PBG	Cotton T-Shirts	Blue	Acme Clothing
MT18402	Mens T PGS	Cotton T-Shirts	Green	Acme Clothing
MT18403	Mens T PGW	Cotton T-Shirts	Gray	Acme Clothing
MT18404	Mens T POY	Cotton T-Shirts	Orange	Acme Clothing

Attempting to remove more than one column at a time is not allowed and displays the following message.

Message

 Illegal source selection: Please select cells in one column only.

OK

Inbound Map Data - Transform

There are numerous ways to use the transformation options to modify the data before it is loaded into STEP. With transformations, you can split one data column into multiple attribute values, append and prepend data to the values, perform search and replace, and so on.

Transformations can also be used to clean data prior to importing. For example, for an attribute named 'Color', the values for Black are represented as 'Black', 'black', 'B', and 'BLK'. You can use a transformation to ensure a single value is added for all of the original values.

Keep the following in mind when using transformations:

- You can apply multiple transformations to the columns of data, but be aware that they are processed sequentially, from top to bottom.
- When setting up transformations that are implemented with a saved configuration, it is recommended that you work with smaller load files first. You can then build up the complexity of your transformations in a step-by-step mode.
- Several of the transformations allow you to manipulate units mm, kg. etc., including manipulations such as removing the unit from the value. Other transformations allow you to extract only the unit from a value. In these cases, the attribute that you are mapping to must be one that is allowed to have units and further, must be one that already has those units assigned to it.

For examples of the most common transformations, see **Transformation Examples**.

To use transformations on attributes in STEP, or on data in tables and publications, see **Attribute Transformations** in the **System Setup / Super User Guide** documentation and **Table Formatting Transformations** in the **Tables** documentation.

Transforming Data

The mapping step for Import Manager and IIEPs include the same options after clicking the **Transform** button.

Configure Processing Engine : Map Data

Source:

<ID>
MT 1840
MT 1840
MT 1840
MT 1840
MT 1840

Result:

ID=<ID>
MT 1840
MT 1840
MT 1840
MT 1840
MT 1840

Specify the sequence of transformations in the table below.

Target:Name

☐ Mandatory

Value - Source: <Name>

Add Transformation

Preview

Name = <Name> ✓

Mens T PBO

Mens T PBG

Mens T PGS

Mens T PGW N

Mens T POY N

Save Reset Cancel

Auto Map Map Constant Remove Transform Generate Profile

Back Next Finish Cancel

For details on each of the available transformation options, see the **Inbound Transformation Options** topic.

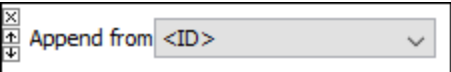
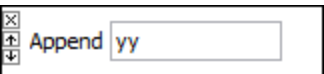
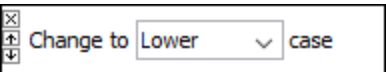
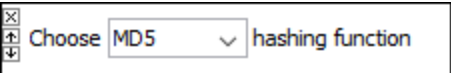
For more information about the additional import steps, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

Dimension Dependent Data

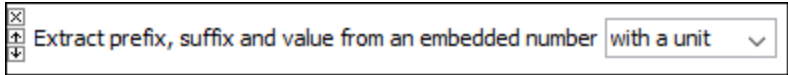
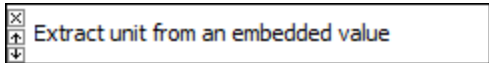
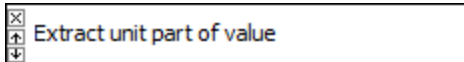
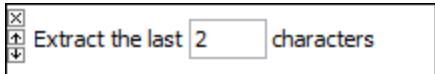
When mapping data from a file to an attribute that is dependent on a single dimension (for example, language or country), transformations make it possible to specify which Dimension Point is required. This allows values for multiple languages or countries to be imported from a single file. For more information, see **Dimension Point for Single Dimension Data Example** documentation.

Inbound Transformation Options

The following transformations are available for the inbound tools Import Manager and IIEP. For details about adding a transformation option, see the **Inbound Map Data - Transform** topic.

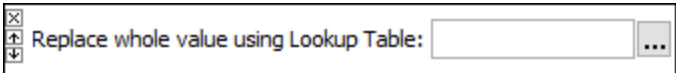
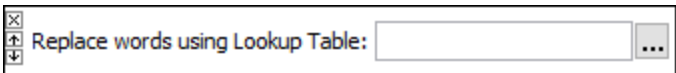
Transformation	Description
Append from multivalue source	 Adds the multi-value text from the source (ID or Name or other attribute value), such as a suffix, to the end of the value. For an example using this transformation, see Merge Data for Multi-Valued Data Example.
Append from source	 Adds the text from the source (ID or Name or other attribute value), such as a suffix, to the end of the value. For an example using this transformation, see Concatenate into Single-Valued Data Example.
Append text	 Adds the specified text as a suffix to the end of the value. For an example using this transformation, see Concatenate Data Using Variables Example.
Change case	 Sets values to all upper case, all lower case, or mixed case. Mixed case sets the first letter of the value to upper case and the remainder to lower case. For an example using this transformation, see Replace the Whole Value Example.
Change hash function	 Converts the attribute value to a cryptographic text using the selected cryptographic function: MD5 or SHA-1. This option is displayed when the STEPMatching licensing is applied. The function reads an attribute value, encrypts it, and saves the encrypted value as a standard attribute value in STEP. It could be used to encrypt (or 'hash') address field(s) so that by evaluating a single field, a changed address could trigger a revalidation process. While available for Attribute Transformation within System Setup, it is intended for use only when importing data using Import Manager or IIEP. No corresponding export function exists, nor is there any

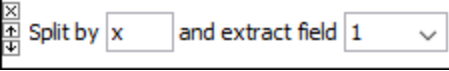
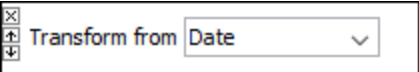
Transformation	Description
	decryption functionality within STEP
Convert fractions to parsable tags	☒ ☐ ☐ Convert fractions to parsable tags using mode Normal Scans a text string for fractions. If a fraction is found, then the fraction is converted into parsable tags. A text string that contains the fraction 1/2 cm is transformed to: <code><StartNumerator/><bflldin/>1<FractionBar/><StartDenominator/>2<EndFraction/>cm</code> You have to set up an output formatting to Indesign for <code><StartNumerator/></code>, <code><bflldin/></code>, <code><FractionBar/></code>, <code><StartDenominator/></code> and <code><EndFraction/></code> to mount the fractions.
Extract characters after a given text in the value	☒ ☐ ☐ Extract characters after end Extracts a part of the value, starting after the occurrence of the specified text string. For an example of this transformation, see Extract Characters Between and Extract Characters After Example.
Extract characters at position	☒ ☐ ☐ Starting from position 1 extract 3 characters Extracts a specified number of characters after a specified position in a text string. For an example using this transformation, see Extract Characters at Position Example.
Extract characters before a given text in the value	☒ ☐ ☐ Extract characters before end Extracts a part of the value, starting from the beginning of the text string and ending at the occurrence of the specified text string. For an example using this transformation, see Replace the Whole Value Example.
Extract characters between two pieces of text in	☒ ☐ ☐ Extract characters between ab and cd Extracts the texts between the specified starting text string and the specified ending text string. The start and end text string cannot be the same. For an example of this transformation, see

Transformation	Description
the value	Extract Characters Between and Extract Characters After Example.
Extract prefix, suffix and value from an embedded number	 Shows the prefix, suffix, and value of the embedded number with / without the unit. This is only relevant if the attribute you are mapping to is of the type 'embedded_number', otherwise, the transformation does not function.
Extract unit from an embedded value	 Extracts only the unit from an embedded value. The prefix, suffix, and value are not extracted. This is only relevant if the attribute you are mapping to is of the type 'embedded_number', otherwise, the transformation does not function.
Extract unit part of value	 Extracts only a unit used in a value of a numeric text, integer, fraction or number attribute.
Extracts the last N characters of the value	 Extracts a specified number of characters of a value, counting from the end.
Format number	 Formats the number to only have a specified number of decimals, and enables you to localize the number. For an example using this transformation, see Format Number Example.
Generate Soundex code	Soundex code for the value. Soundex is a general algorithm which creates the same type of 4 alphanumeric code for the words with similar pronunciation.

Transformation	Description
If equals, source	✕ ↑ ↓ If value equals match string Source <ID> Else source <ID> Checks if the value is equal to the source value. For an example using this transformation, see If Equals Source Example.
Insert text at a given position	✕ ↑ ↓ Insert abc at position 1 Inserts a series of characters into the attribute value at the specified position. For an example using this transformation, see Insert Text Example.
Insert text before	✕ ↑ ↓ Insert xx before Adds the specified text as a prefix before the value. For an example using this transformation, see Insert Text Example.
Math - Add Source	✕ ↑ ↓ Add <ID> Performs the addition operation on the value with the source. Only works for numeric values.
Math - Divide by source	✕ ↑ ↓ Divide by <ID> Maximum number of decimals (Leave blank to use number of decimals from source) Remove trailing zeroes ☐ Performs the division operation on the value with the source. Only works for numeric values.
Math - Multiply by source	✕ ↑ ↓ Multiply by <ID> Performs the multiplication operation on the value with the source. Only works for numeric values.
Math - Subtract	✕ ↑ ↓ Subtract <ID>

Transformation	Description
source	Performs the subtract operation on the value with the source. Only works for numeric values.
Math Operation (+, -, *, /)	☐ Value = Value + 0 Performs the standard math operations on the value: addition, subtraction, multiplication, and division. Only valid for integer and number attributes. For an example of this transformation, see Math Operation Example.
Parse address	☐ AddressParser Transformation Splits an address contained in a single field into multiple fields using <multisep/>. This allows addresses to be normalized and validated. For example, the import of "1200 Mail Rd Hays KS 11111" should be transformed to "KS<multisep/>HAYS<multisep/>11111<multisep/>MAIL<multisep/>RD<multisep/>1200".
Prepend from source	☐ Prepend from <ID> Adds the text from the source, such as a prefix, at the beginning of the value.
Remove unit from value	☐ Remove unit from value Extracts the value of an attribute. If the value includes a unit, then the unit will be ignored. For an example using this transformation, see Split and Extract Data Example.
Replace substrings of the value	☐ Replace sub-strings x with y Changes the specified text string or empty value with the replacement text string or empty value. All matches on the value in the 'Replace sub-strings' text box are replaced. For an example of this transformation, see Replace Substrings of the Value Example.
Replace substrings of the value	☐ Replace if matches x* with Uses regular expression to change the specified find value (in the first text box) with the

Transformation	Description				
using a regular expression	replacement value (in the second text box). From the dropdown, select a replace option: 'if matches', 'first match of', or 'every match'. The if matches option must be set so that the find field defines the <u>entire</u> string to be replaced because it operates like the 'Replace the whole value' transformation. The first match of and every match options operate like the 'Replace substrings of the value' transformation and they replace only the text specified in the find field. For more information on regular expression and practical examples, see the Regular Expression topic in System Setup / Super User Guide documentation.				
Replace the whole value	 Replaces the specified text or empty value with a different value. You must enter the entire value, or leave the Replace field blank. Only works on text-based attributes. For an example using this transformation, see Replace the Whole Value Example.				
Replace the whole value using Lookup Table	 Replaces the specified value with another value from the selected Lookup Table. For more information, see the Transformation Lookup Tables topic in the System Setup / Super User Guide.				
Replace words using Lookup Table	 Replaces the specified words with another value from the selected Lookup Table. 'Words' are defined as characters or sequences of characters where the character / sequence: <table border="1"> <thead> <tr> <th>Follows either</th><th>AND Precedes either</th></tr> </thead> <tbody> <tr> <td> - The start of the string / value - A space - Punctuation </td><td> - The end of the string / value - A space - Punctuation </td></tr> </tbody> </table> Note: It is not possible to match on From entries with punctuation. This means the From value 'Apt.' will not match anything in the string 'Main Street 100, Apt. 3'. For more information, see the Transformation Lookup Tables topic in the System Setup /	Follows either	AND Precedes either	- The start of the string / value - A space - Punctuation	- The end of the string / value - A space - Punctuation
Follows either	AND Precedes either				
- The start of the string / value - A space - Punctuation	- The end of the string / value - A space - Punctuation				

Transformation	Description
	Super User Guide.
Split and extract	 Split the value where a given string is found and returns one of the created fields. You can, for example extract data from a value after the 4th occurrence of a hyphen, and before the 5th occurrence. So if the value is 179-245-Y-MYGR-TG105P-FDSJ-198, the result is TG105P. For an example using this transformation, see Split and Extract Data Example.
Transform date by locale	 Localized dates should follow the DateFormat.SHORT JavaScript formatting style. Times should be in the DateFormat.MEDIUM formatting style. Transforms Date, ISO Date, or ISO Date and Time attributes to localized formats. Inbound transforms locale-formatted Date, ISO Date, or ISO Date and Time attributes. The locale format must match the context being used in workbench at the time of import. If not, the 'Transform date by pattern' transformation should be used. Outbound transforms Date, ISO Date, or ISO Date and Time attributes to localized formats. Locale is determined by the context being used in workbench at the time of export. For step-by-step instructions using this transformation, see the Transform Date by Locale or by Pattern topic of the Transformation Examples section of the Data Exchange documentation.
Transform date by pattern	 Transforms Date, ISO Date, or ISO Date and Time attributes to a patterned format (e.g., yyyy.MM.dd G 'at' HH:mm:ss z, EE, MMM d, "yy). You can use the pre-filled pattern provided or manually enter (or cut and paste) a different pattern format. Validation Base Type for the attribute being exported must match the 'Transform from' dropdown choice. If exporting data to a pattern that includes a time zone, the time zone is determined by the STEP server location. Inbound transforms patterned dates (e.g., yyyy.MM.dd G 'at' HH:mm:ss z, EE, MMM d, "yy) to Date, ISO Date, or ISO Date and Time. Validation Base Type for the mapped attribute must match the 'Transform to' dropdown choice. Outbound transforms Date, ISO Date, or ISO Date and Time attributes to a patterned format

Transformation	Description
	(e.g., yyyy.MM.dd G 'at' HH:mm:ss z, EE, MMM d, "yy). You can use the prefilled pattern provided or manually enter (or cut and paste) a different pattern format. Validation Base Type for the attribute being exported must match the 'Transform from' dropdown choice. If exporting data to a pattern that includes a time zone, the time zone is determined by the STEP server location. For step-by-step instructions using this transformation, see the Transform Date by Locale or by Pattern topic of the Transformation Examples section of the Data Exchange documentation.
[Transform using attribute transformation]	Attribute Transformation Applies an attribute transformation that is already defined in System Setup under Attribute Transformations. Makes it possible to apply a set of transformations to a value. The transformation takes a predefined attribute transformation as argument. For an example using this transformation, see Transform Using Attribute Transformation Example.

Outbound Map Data Options

Mapping allows you to specify data sources that you want to extract for each record. When mapping is active, only data that is mapped is exported. For Excel / CSV, each STEP data source mapped is displayed as a column in the output.

Mapping is not required for the STEPXML or Advanced STEPXML formats.

Formats

Mapping differs slightly based on the type of data being output.

Tabular Formats

The following tabular formats require mapping for outbound data:

- CSV Format
- Excel Format
- Excel Smartsheet Format

XML Formats

The following XML formats require mapping for outbound data:

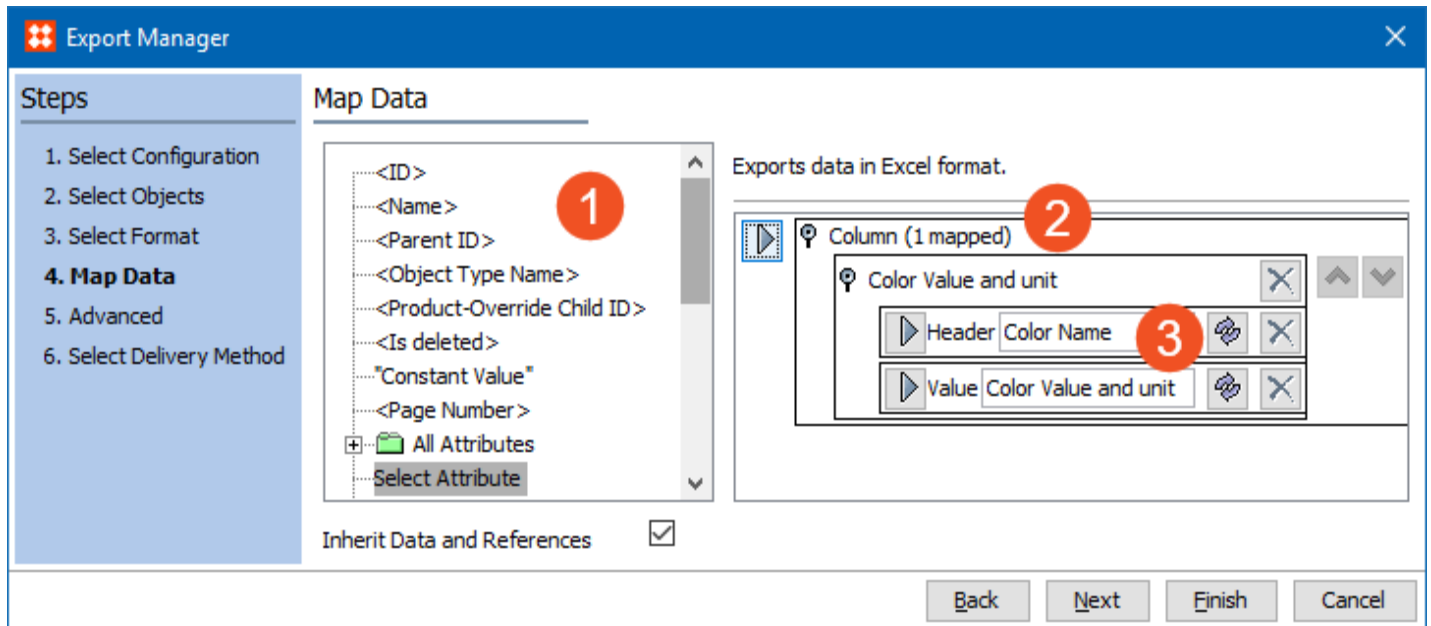
- BMEcat Format
- BMEcat 2005 Format
- Generic XML Format
- IDoc MATMAS 05 Format

Mapping in Outbound Tools

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

The Map Data step is displayed in the Export Manager wizard based on the selected format.



Export Manager

Steps

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

Map Data

Exports data in Excel format.

1. List of attributes: <ID>, <Name>, <Parent ID>, <Object Type Name>, <Product-Override Child ID>, <Is deleted>, "Constant Value", <Page Number>, All Attributes, Select Attribute.

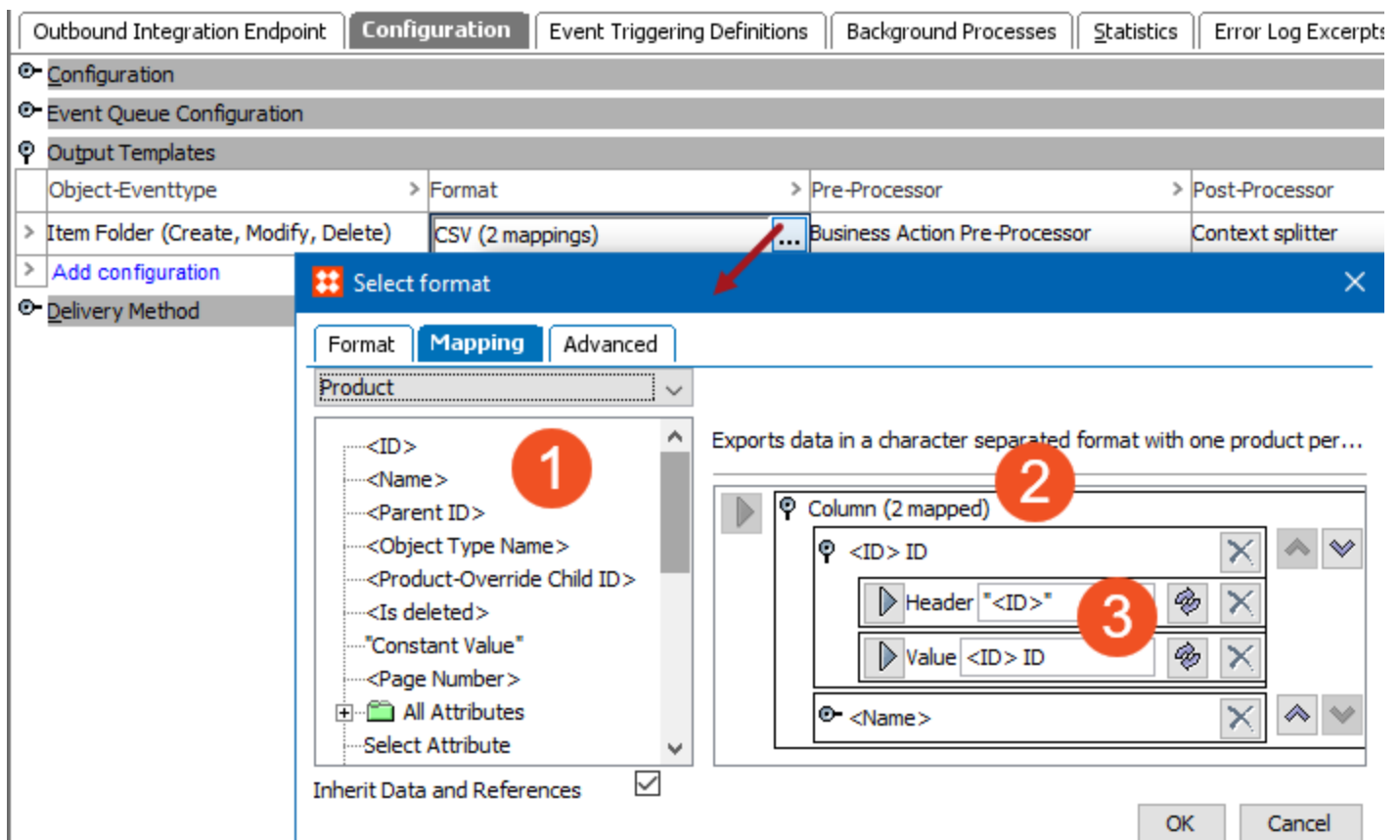
2. Column (1 mapped): Color Value and unit

3. Header: Color Name, Value: Color Value and unit

Inherit Data and References ☒

Back Next Finish Cancel

For outbound integration endpoints, based on the format selected in the Format parameter, mapping is available on a Mapping tab.



Configuration

Outbound Integration Endpoint Configuration Event Triggering Definitions Background Processes Statistics Error Log Excerpts

Configuration

Event Queue Configuration

Output Templates

Object-Eventtype	Format	Pre-Processor	Post-Processor
Item Folder (Create, Modify, Delete)	CSV (2 mappings)	Business Action Pre-Processor	Context splitter

Add configuration

Delivery Method

Select format

Format Mapping Advanced

Product

1. List of attributes: <ID>, <Name>, <Parent ID>, <Object Type Name>, <Product-Override Child ID>, <Is deleted>, "Constant Value", <Page Number>, All Attributes, Select Attribute.

2. Column (2 mapped): <ID> ID

3. Header: "<ID>", Value: "<ID> ID", <Name>

Exports data in a character separated format with one product per...

Inherit Data and References ☒

OK Cancel

Mapping

The numbered elements in the images above include:

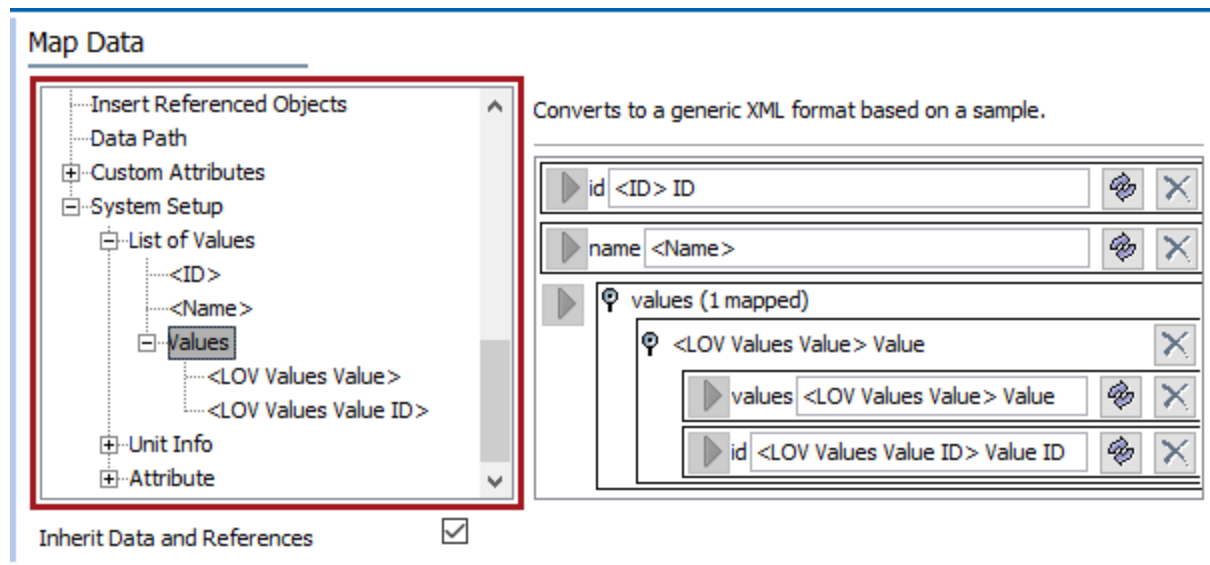
1. **Data Sources** indicate the original location of the information to be exported. The mapping page provides a way to associate the mapping targets with the STEP information available for the selected object types being exported. For more information on each source option, see the **Outbound Map Data - Data Source** documentation.
2. **Mapping Targets** are placeholders for the actual data being exported. The selected format determines which targets are available. See the mapping section of your format documentation (listed above) and the **Outbound Map Data - Mapping Targets** documentation.
3. **Mapping Rules** represent the relationship, also called a binding, between a data source and a mapping target. Bindings can be configured in a number of ways, adding a constant value, applying a transformation (including choosing a different aspect), or mapping additional STEP data.

Transformations

The transform button allows you to modify the mapped data before export. For details about the available options, see the **Outbound Map Data - Transform** documentation.

Outbound Map Data - Data Source

Data Sources (outlined in the Map Data image below) indicate the original location of the information being exported. A data source provides a way to associate the mapping targets (defined by the template and displayed on the right side) with the STEP information available for the selected object types being exported.



The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

The process to map data differs only slightly based on the data source selected. Generally, mapping data involves selecting one or more data sources from the left panel and then clicking the mapping target arrow in the right panel. The selected data appears in the list of columns to export.

Available Data Source Options

The following options are available for all formats that require outbound mapping, with the exceptions noted:

- All Attributes - Data Source Outbound
- Asset References - Data Source Outbound
- Attribute Links - Data Source Outbound or - for Generic XML and IDoc MATMAS 05 only
- Classification References - Data Source Outbound
- Classifications - Data Source Outbound
- Constant Value - Data Source Outbound

- Custom Attributes - Data Source Outbound
- Data Path - Data Source Outbound - for Generic XML and IDoc MATMAS 05 only
- Entity References - Data Source Outbound
- ID or Key - Data Source Outbound
- Index Words - Data Source Outbound - for STEP'n'design only
- Inherit Data and References - Data Source Outbound
- Insert Referenced Objects - Data Source Outbound
- Is Deleted - Data Source Outbound - for Generic XML only
- Multi Level Parent Attributes - Data Source Outbound
- Multi Level References - Data Source Outbound
- Name - Data Source Outbound
- Object Type Name - Data Source Outbound
- Page Number - Data Source Outbound - for STEP'n'design only
- Parent ID - Data Source Outbound
- Product Classification Links - Data Source Outbound
- Product-Override Child ID - Data Source Outbound
- Product References - Data Source Outbound
- Select Attribute - Data Source Outbound
- STEP Workflow Task Info - Data Source Outbound
- System Setup - Data Source Outbound - for Generic XML only

Asset References - Data Source Outbound

Asset references can be exported when they reference products, classifications, or entities. Asset references can be mapped via their ID or a unique key. Each selected asset reference is extracted into a separate field in the output.

The Aspect option is available for Asset References and allows a variety of data on the asset reference to be exported.

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

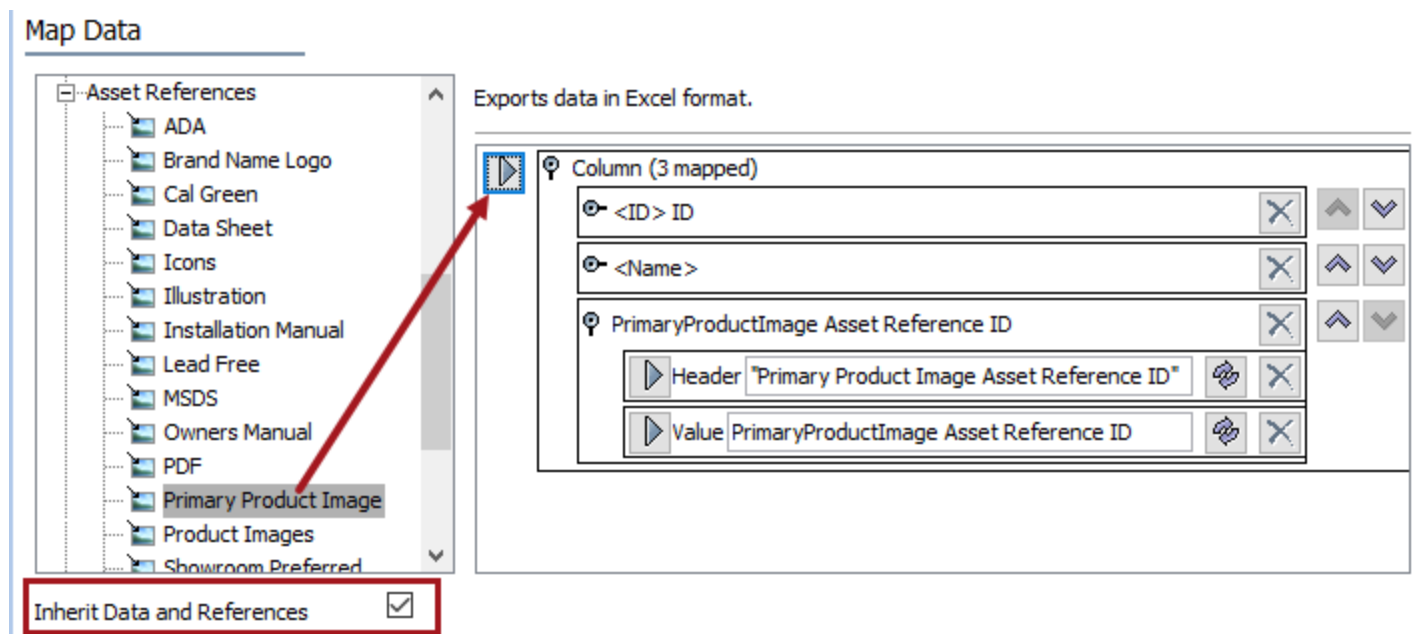
After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Asset References

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, expand **Asset References**, and select the desired asset reference type.

Note: For an OIEP, you must first select Product from the dropdown at the top of the Mapping tab.

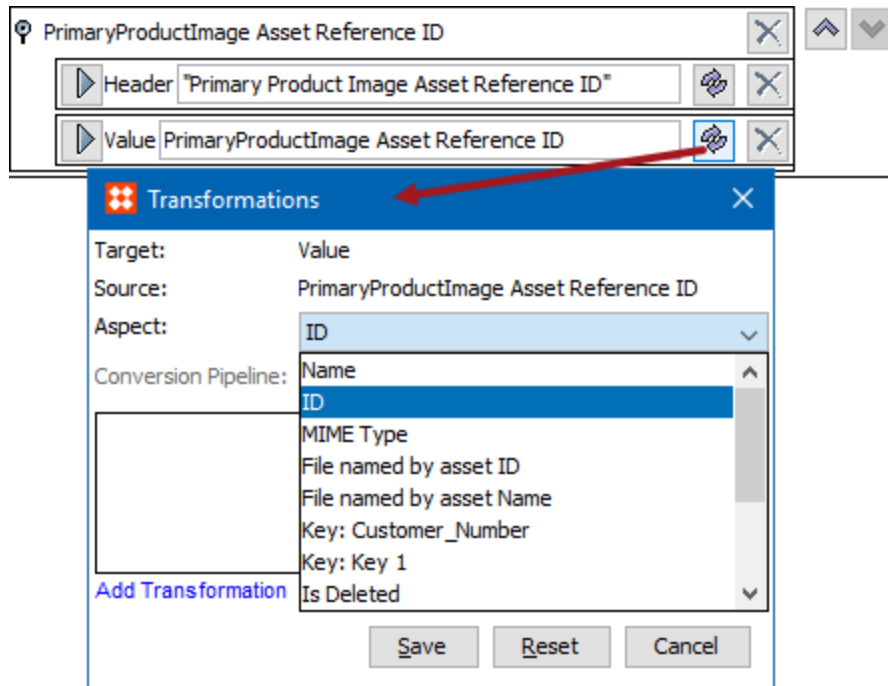
3. Click the right arrow button (▶) in the right panel to add the selected asset reference type.



4. Check the **Inherit Data and References** option appropriately:
 - If **checked**, inherited data and references are included in the export.
 - If **unchecked**, inherited data and references is not included in the export.

For information on inheritance, see the Inheritance in the **Inherit Data and References - Data Source Outbound** topic.

5. If needed, change the reference Value Aspect to output data other than the reference ID. For examples, see the **Results** section below. For details on the options, see the **Aspect - Transform Outbound** topic.



- Open the asset reference flipper to display the Header and Value elements.
 - Click the transformation button () for the Value element to display the Transformations dialog and the Aspects parameter.
 - Click the **Aspect** dropdown to display and select an option.
 - Click the **Save** button to apply any aspect changes.
5. Apply any transformations necessary to change the output without changing the original data. See the **Outbound Map Data - Transform** topic for details.
 6. Complete any additional mappings and initiate the export.

Results

The exported file includes Product ID, Name, Asset Reference Type ID as header and referenced asset ID as value.

<ID>	<Name>	Primary Product Image Asset Reference ID ()
123853	T-Shirts Family	20585

When exporting multiple products that are linked to assets with different asset reference types, each asset type is displayed in its own column. When multiple references for the same reference type exist, they are separated by a semicolon.

<ID>	<Name>	Primary Product Image Asset Reference ID	Product Images Asset Reference ID
123853	T-Shirts Family	20585	
123854	123854 B		20584;910836
123855	123855 O		20586
123857	123857 G		112806

When the Name aspect is used, the output includes product ID, name, and mapped Asset Reference Type Name as header and referenced asset's name as value for all exported rows.

<ID>	<Name>	Product Images Asset Reference Name ()
123854	123854 B	Hanes Blue;Blue Shirt

When the MIME Type aspect is used, the output includes asset reference type MIME Type as the header and referenced asset's MIME Type as its value in the exported file. For more information, see the **MIME Types** section of the **System Setup / Super User Guide** documentation.

<ID>	<Name>	Primary Product Image Asset Reference MIME Type ()
123854	123854 B	image/tiff
123855	123855 O	image/tiff
123857	123857 G	image/tiff

When the 'File named by asset ID' aspect is selected, multiple files are output in a .ZIP file:

- The digital media files associated with the referenced assets. If the referenced asset is an image, the image can be converted using the conversion pipeline.
- The selected export format file. This file includes product ID, name and 'Asset reference file named by Asset ID' as header and asset ID with corresponding asset extension as value for all the exported rows.

<ID>	<Name>	Product Images Asset Reference File named by asset ID ()
123854	123854 B	20584.tif
123855	123855 O	20586.tif
123857	123857 G	112806.jpg

When the 'File named by asset Name' aspect is selected, multiple files are output in a .ZIP file:

- The digital media files associated with the referenced assets. If the referenced asset is an image, the image can be converted using the conversion pipeline.
- The selected export format file. This file includes product ID, name and 'Asset reference file named by Asset Name' as header and asset name with corresponding asset extension as value for all the exported rows.

<ID>	<Name>	Product Images Asset Reference File named by asset Name ()	Primary Product Image Asset Reference File named by asset Name ()
123853	T-Shirts Family		Hanes Family.tif
123854	123854 B	Hanes Blue.tif	
123855	123855 O	Hanes Orange.tif	
123857	123857 G	HanesGreenT.jpg	

Attributes (and Data Containers) - Data Source Outbound

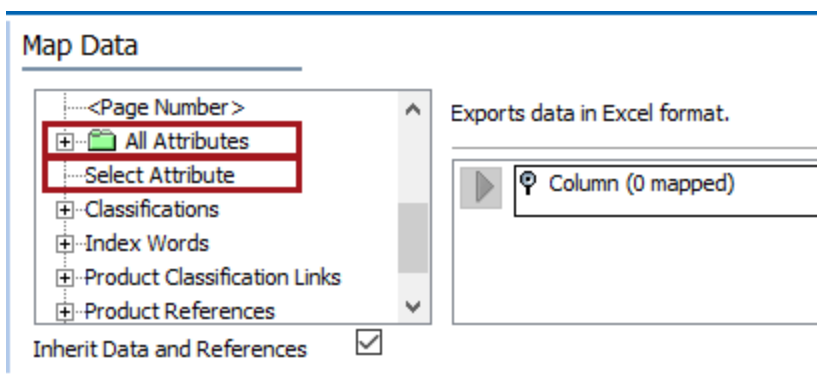
Attributes and Data Containers can be selected and exported for the objects being exported.

The following options are available to export attributes:

- **All Attributes** shows attributes that are relevant for the objects that are selected for export. This includes specification attributes linked to the selected object(s) and description attributes that are valid for the selected object(s). You can select and map attribute groups in the hierarchy, allowing you to map all attributes in the group. The top level node All Attributes can also be mapped, allowing you to map all attributes relevant for the selected object(s).

The All Attributes option is also used to map **data containers**. For details, see the **Mapping Data Containers via All Attributes** section below.

- **Select Attribute** allows you to browse or search for an attribute, and also Force Calculation when the attribute is calculated.



The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

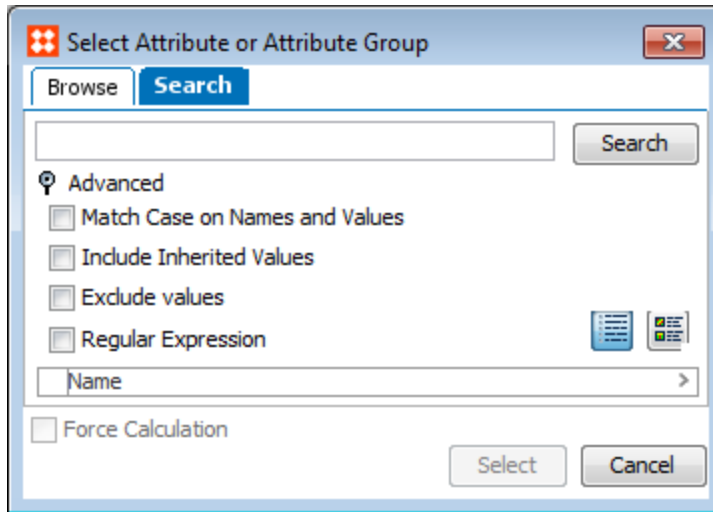
After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Attributes via Select Attribute

Browse or Search can be used to find any attribute.

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.

2. On the Map Data step, in the left panel, click **Select Attribute**.
3. In the right panel, click the right arrow (▶) to display the **Select Attribute or Attribute Group** dialog.



4. Find the attribute being mapped using one of the following methods:
 - The Browse tab allows finding an item via the hierarchy.
 - The Search tab allows entering part of the name of the attribute to to display matches. Click the **Search** button to display all matches and choose an attribute. Open the Advanced flipper to use the advanced search options. For more information, see the **Advanced Options** topic in the **Getting Started / User Guide** documentation.

When selecting a calculated attribute, the Force Calculation checkbox is enabled. Checking this box ensures the value of the calculation will be resolved upon export. When unchecked, calculated values can still be generated via the **Include Calculated Attribute Values** option on the Advanced step.

Note: Calculated attributes can be mapped and resolved even when they are not valid for the objects being exported. However, the elements of the calculation (for example, the value of an attribute) must be valid and have a value in order to display as expected in the final output.

3. Select the relevant attribute, and then click **Select**. If Force Calculation was checked, the attribute transformation dialog is displayed with the Force Calculation option checked.
4. Repeat these steps to add all other required attributes.
5. Apply any transformations, which can change the output without changing the original data. See the **Aspect - Transform Outbound** topic.
6. Complete any additional mappings and initiate the export.

Mapping Attributes or Attribute Groups via All Attributes

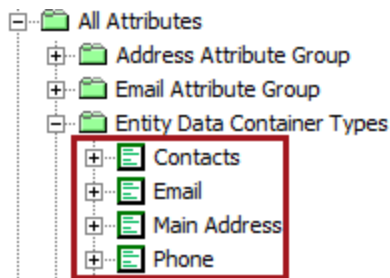
Only attributes that are relevant for the objects that are selected for export are available.

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.

2. On the Map Data step, in the left panel, expand the **All Attributes** node and select one or more attributes or attribute groups.
3. In the right panel, click the right arrow button () to add the selections.
4. Apply any transformations, which can change the output without changing the original data. See the **Aspect - Transform Outbound** topic.
5. Complete any additional mappings and initiate the export.

Mapping Data Containers via All Attributes

For exporting data containers into CSV or Excel formats (which require mapping), only **All Attributes** can be used. Using this option, data containers can be mapped directly to the export or included in a higher level selection.

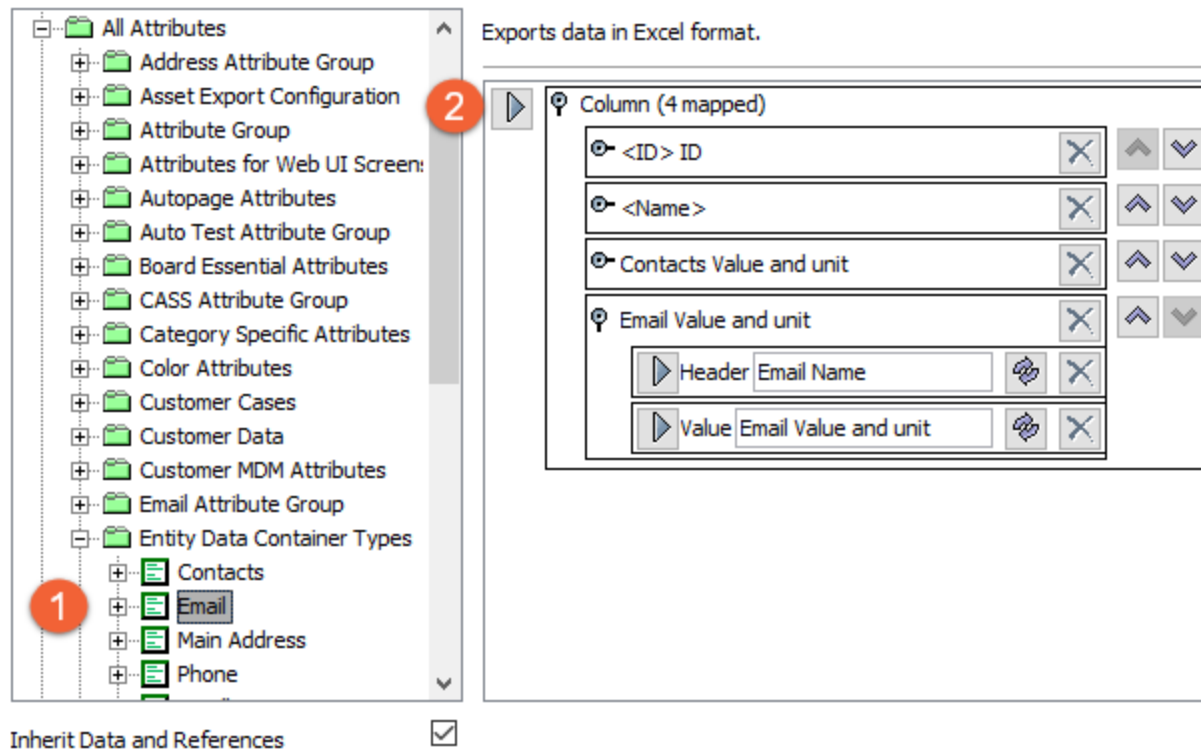


For information on exporting data containers without mapping, see the **Data Containers in STEPXML** section of the **Data Exchange** documentation.

The following steps define mapping for data containers.

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, click **All Attributes** and select the desired data container type(s).

Map Data



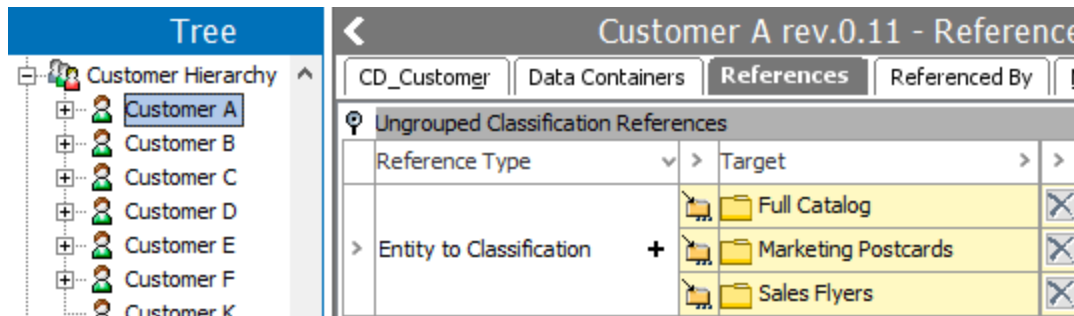
3. Click the right arrow button (▶) in the right panel to add the selected data container type as a group of columns.
4. Add additional data container types, if needed.
5. Apply any transformations, which can change the output without changing the original data. See the **Outbound Map Data - Transform** topic.
6. Complete any additional mappings and initiate the export.

For information on handling data container transformations during export, see the **Data Container Separator - Transform Outbound** section of the **Data Exchange** documentation.

Classification References – Data Source Outbound

The option to map a classification is only available when the object super type selected for the export is Product or Entity. Only the classification reference types displayed are those that match the selected object type. Classification-to-classification references are not available for output.

When mapping a classification reference, users can include or exclude inherited references.



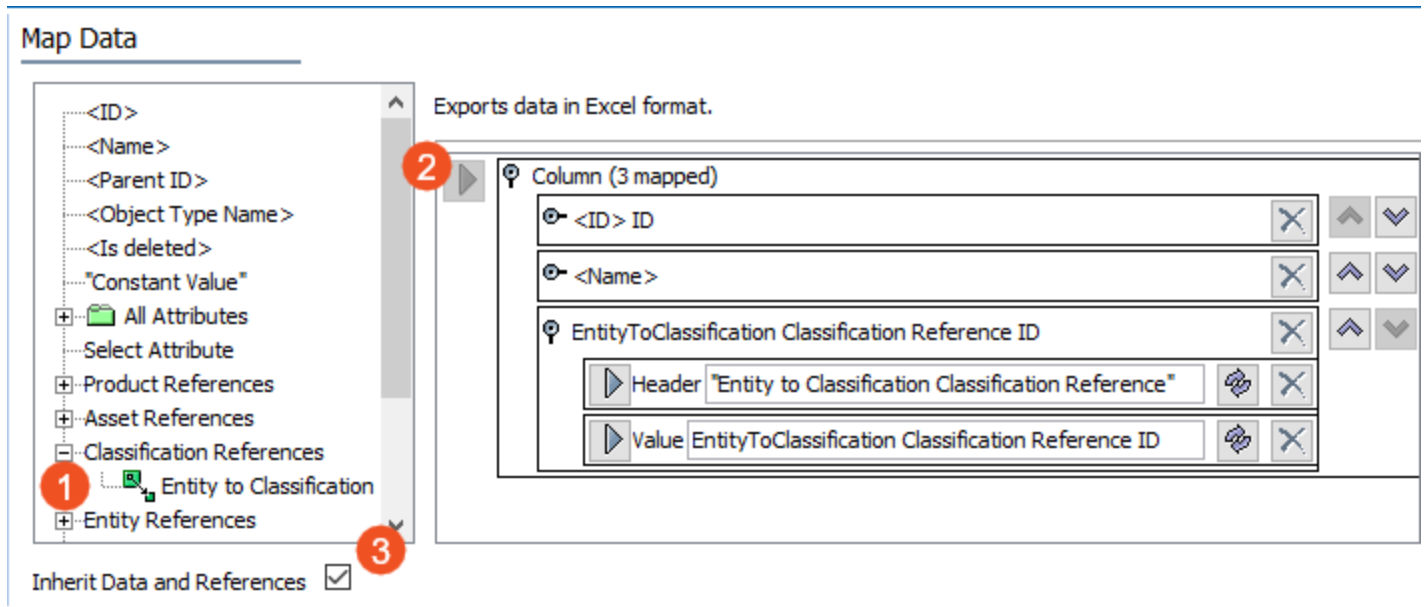
The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Classification References

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, expand Classification References, and select the desired Classification Reference type.



3. Click the right arrow button in the right panel to add the selected classification reference type as a column. Add additional classification reference types, if needed.
4. Check the **Inherit Data and References** option appropriately:

- If **checked**, inherited data and references are included in the export.
- If **unchecked**, inherited data and references is not included in the export.

For information on inheritance, see the Inheritance in the **Inherit Data and References - Data Source Outbound** topic.

5. Apply any transformations, which can change the output without changing the original data. See the **Outbound Map Data - Transform** topic.
6. Complete any additional mappings and initiate the export.

Results

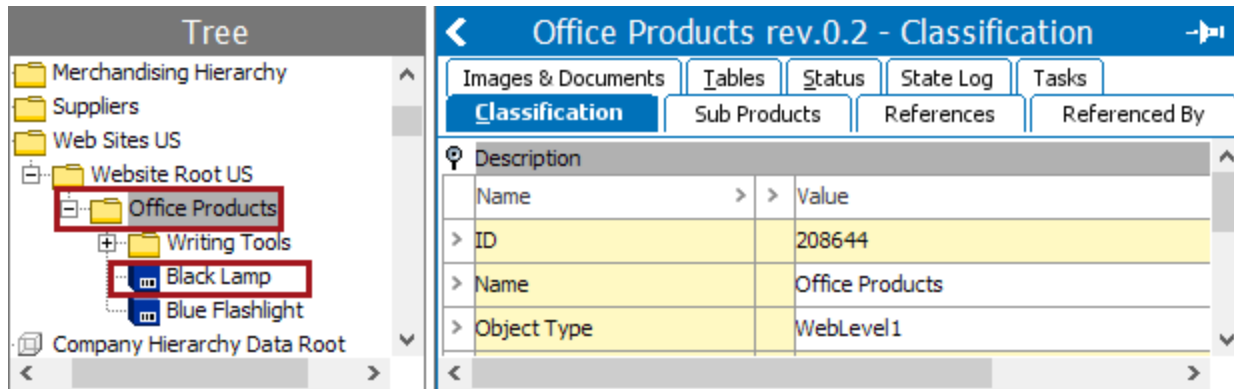
The output includes the classification reference type name as a header and the ID for each row.

When the product references multiple classifications, the classification reference IDs are displayed in a semi-colon separated list.

	A	B	C
1	<ID>	<Name>	Entity to Classification Classification Reference
2	CUS_101545	Customer A	FullCatalog;MarketingPostcards;SalesFlyers
3	CUS_107835	Customer B	SalesFlyers

Classifications - Data Source Outbound

The option to map a classification is only available when the object type selected for the export is Product. Additionally, the ID of a classification is only output when the product being exported has a reference to the mapped classification or to one of the descendant classifications. For example, by mapping the top level classification, details of all the classifications referenced by product are exported.



The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

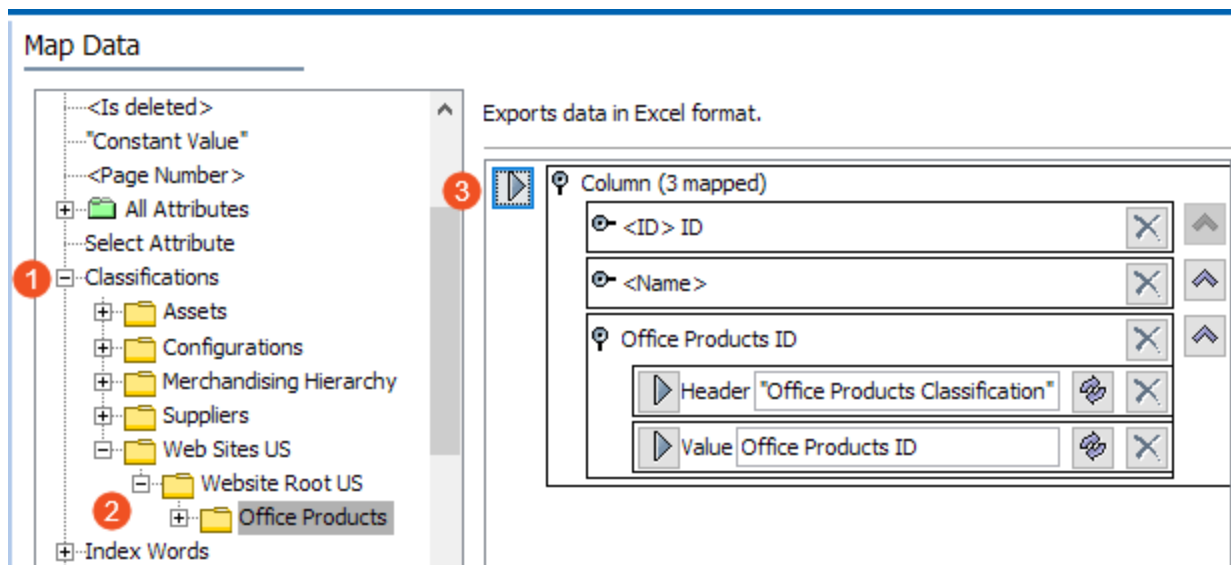
For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Classifications

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, expand the Classification node to show all the classifications.

3. Select the desired classification and click the right arrow button.



4. Apply any transformations, which can change the output without changing the original data. See the **Outbound Map Data - Transform** topic.
5. Complete any additional mappings and initiate the export.

Results

The output includes the classification reference type name as a header and the ID for each row.

	A	B	C
1	<ID>	<Name>	Office Products Classification
2	208653	Black Lamp	208644

When the product references multiple classifications, the classification IDs are displayed in a semi-colon separated list.

	A	B	C
1	<ID>	<Name>	Suppliers Classification
2	134887	Blue Flashlight	20851;208645

Constant Value - Data Source Outbound

Mapping a constant value allows you to replace the default data with your own text. For example, use a constant value to change a header from the default of <Name> to 'Product Name', or to add static text to the value.

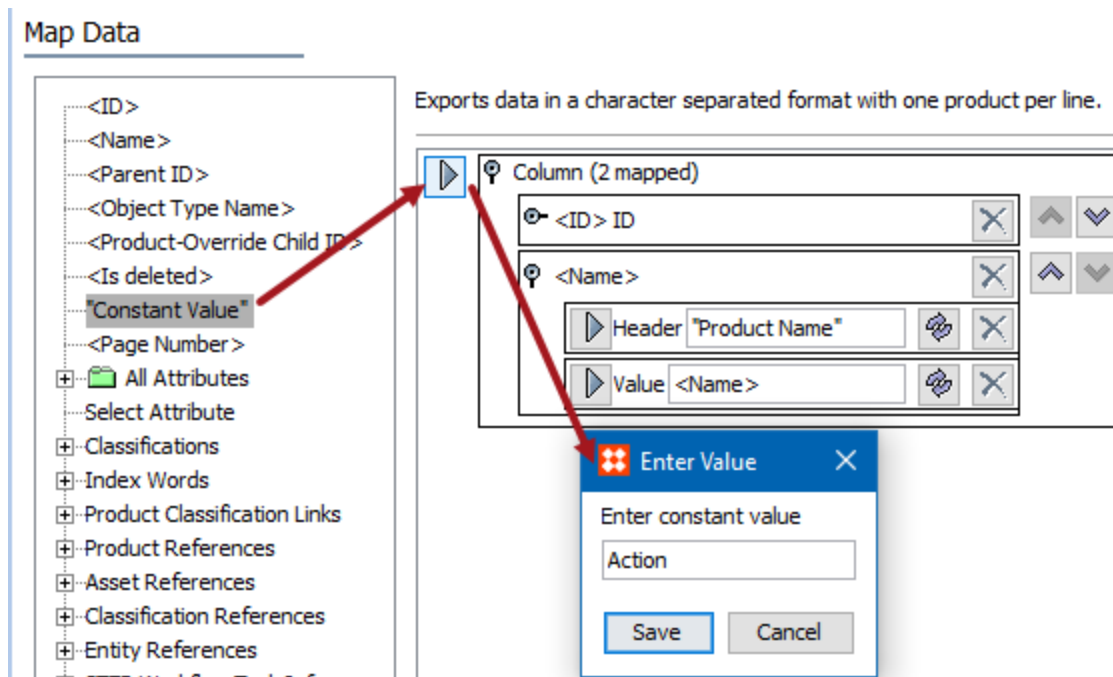
The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

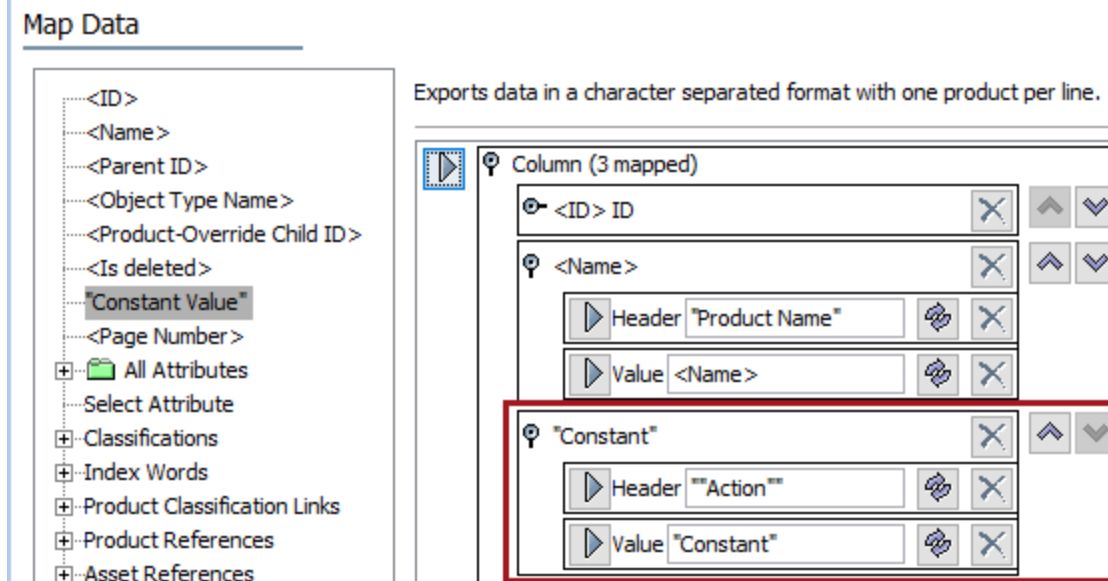
For an expanded example of mapping aspects and constants, see the **Header and Value Aspects Example** documentation.

Mapping Constant Value as a New Field

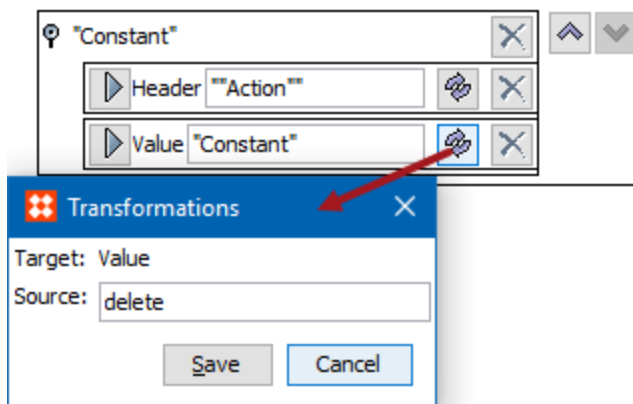


1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, select **Constant Value**.
3. In the right panel, click the top-most right arrow () to add a new field to the export file.

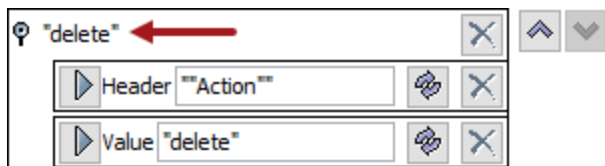
4. In the **Enter Value** dialog, enter the constant value to assign to all records in the export file, and click **Save**.
5. The new "Constant" field is displayed using the text you entered as the Header.



6. Use the transformation button to update the Value field, if necessary. You can also use the transformation to remove the extra set of double quotes displayed by default.



7. Click **Save** and the value is then displayed as the field name.



8. Complete the mapping and initiate the export.

Results

The exported file includes ID, product name and an action column with constant value “delete”.

<ID>	Product Name	"Action"
123853	T-Shirts Family	delete
123854	123854 B	delete
123855	123855 O	delete
123857	123857 G	delete

Mapping Constant Value Within a Field

Map Data

The screenshot shows the 'Map Data' interface. On the left, a list of fields includes '<ID>', '<Name>', '<Parent ID>', '<Object Type Name>', '<Product-Override Child ID>', '<Is deleted>', 'Constant Value', '<Page Number>', 'All Attributes', 'Select Attribute', 'Classifications', 'Index Words', 'Product Classification Links', 'Product References', and 'Asset References'. The 'Constant Value' field is highlighted. A red arrow points from 'Constant Value' to the 'Header' field in the 'Column (2 mapped)' panel on the right. The 'Header' field is currently set to '<Name>'. Another red arrow points from the 'Header' field to the 'Enter Value' dialog box. The dialog box has a title bar 'Enter Value' and a text input field containing 'Product Name'. Below the input field are 'Save' and 'Cancel' buttons. The 'Column (2 mapped)' panel also shows a 'Value' field set to '<Name>'.

1. In the left panel, select **Constant Value**.
2. In the right panel, click the right arrow (▶) for the desired field.
3. In the **Enter Value** dialog, enter the constant value to assign to all records in the export file, and click **Save**.
4. The constant value is displayed as a new element in the right panel. If necessary, remove any duplicate data by clicking the X button

Map Data

<ID>
<Name>
<Parent ID>
<Object Type Name>
<Product-Override Child ID>
<Is deleted>
Constant Value"
<Page Number>
+ All Attributes
Select Attribute
+ Classifications

Exports data in a character separated format with one product per line.

Column (2 mapped)

<ID> ID
<Name>
Header "<Name>"
Product Name"
Value <Name>

5. Complete any additional mappings and initiate the export.

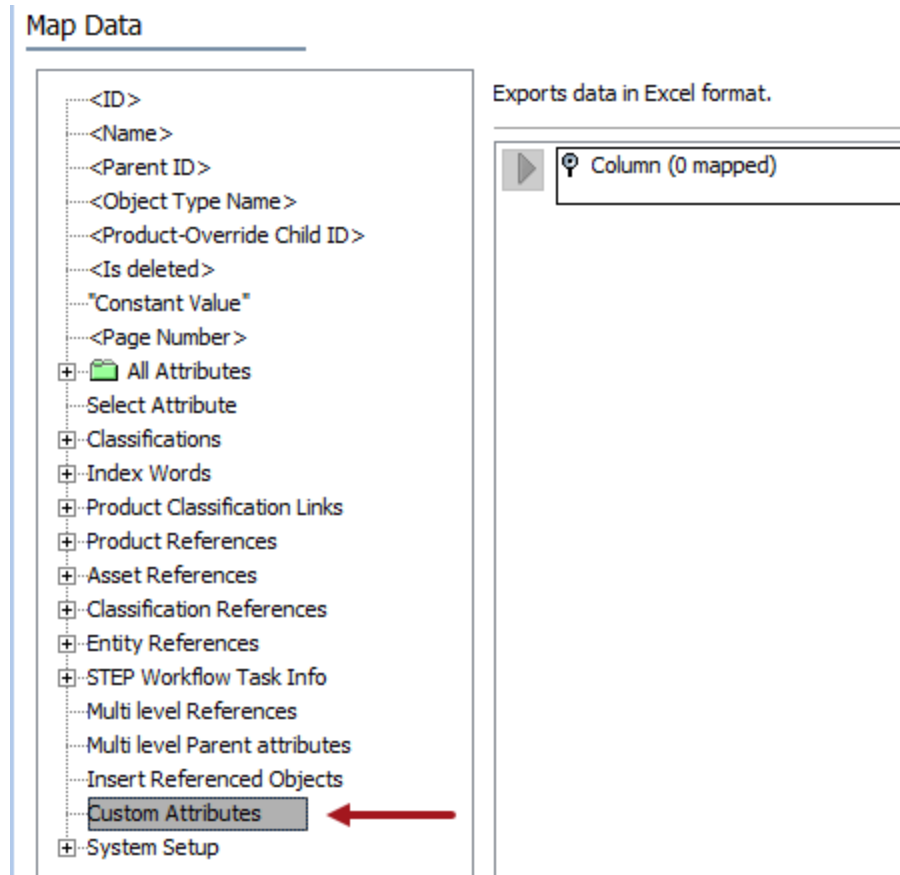
Results

The exported file includes existing name column header to be modified with a constant value as 'Product Name'.

<ID>	Product Name
123853	T-Shirts Family
123854	123854 B
123855	123855 O
123857	123857 G

Custom Attributes - Data Source Outbound

Custom attributes are legacy functionality only available via custom code written by a STEP developer. When custom attributes exist, the Custom Attributes data source is used to export the data.



The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Data Path - Data Source Outbound

The Data Path data source is only available with the Generic XML and IDoc MATMAS 05 format. The following mapping examples use the 'Data Path' data source:

- Nestable Data Paths in Generic XML
- Nestable Data Paths in Generic XML Example
- Mapping for Nestable Data Paths in Generic XML Example

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

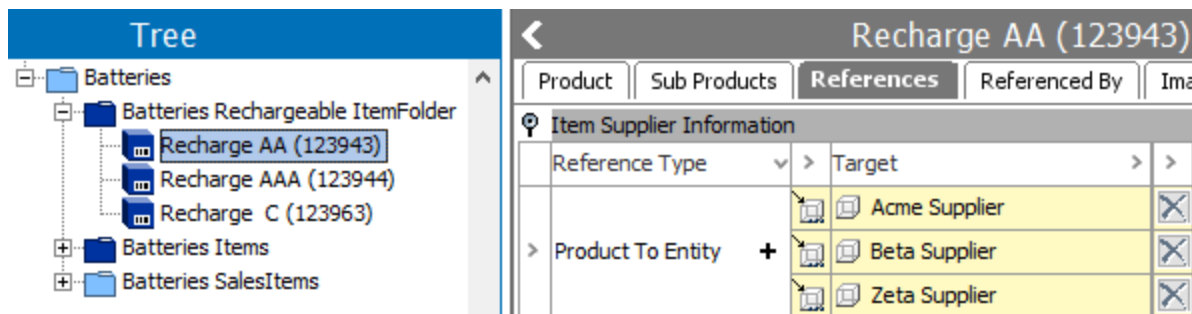
Entity References – Data Source Outbound

The option to map an entity reference is available when the object super type selected for the export is Asset, Classification, Entity, or Product. The only displayed entity reference types for mapping are those that match the selected super type. Publication-to-entity references are not available for output.

For information on selecting an object super type, see the following topics:

- For Export Manager, use the Export dropdown as described in the **Export Manager - Select Objects** topic.
- For an event-based OIEP, use the object super type dropdown as described in the **OIEP - Event-Based - Output Templates Flipper** topic.
- For a select objects OIEP, use the object super type dropdown as described in the **OIEP - Select Objects - Output Templates Flipper** topic.

For example, consider details of various suppliers of a product, where suppliers are the entity object within STEP. The entity reference mapping would allow you to extract the list of suppliers for a product.



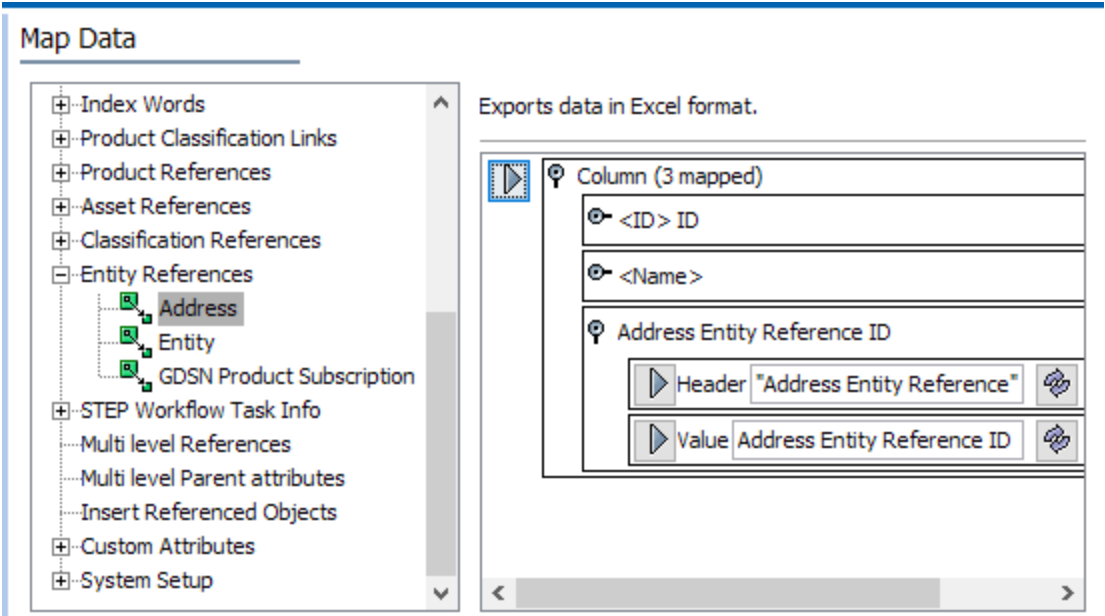
The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Entity References

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, select **Entity References** to expand the node.
3. Select the desired entity reference type.



- 4. In the right panel, click the top-most right arrow to add the selected entity reference type to the export file.
- 5. Apply any transformations, which can change the output without changing the original data. See the **Outbound Map Data - Transform** topic.
- 6. Complete any additional mappings and initiate the export.

Results

The output includes the entity reference type name as a header and the ID for each row.

When the object references multiple entities, the entity reference IDs are displayed in a semi-colon separated list.

	A	B	C
1	<ID>	<Name>	Product To Entity Entity Reference
2	123943	Recharge AA (123943)	AcmeSupplier;BetaSupplier;ZetaSupplier
3	123944	Recharge AAA (123944)	ZetaSupplier

ID or Key - Data Source Outbound

When the STEP ID is used to identify an object, the ID data source is used to identify the mapping target intended to hold ID. All objects in STEP can be exported using the ID data source.

When a key is used in place of the STEP ID by external systems, prior to exporting, keys must be activated in STEP so they can be displayed in the Map Data step of an export tool. For more on keys, see the **Unique Keys** section of the **System Setup / Super User Guide** documentation.

While one image below shows Generic XML, and the steps below show Excel, this topic applies to any of the formats that require mapping as defined in the **Outbound Map Data Options** topic.

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

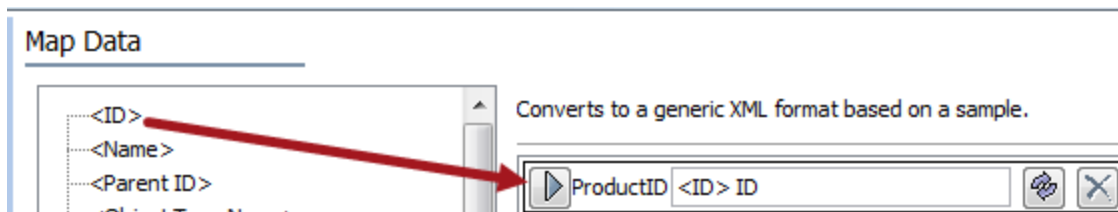
For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping ID or Key

The steps for mapping IDs are very similar to those for mapping a key.

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, select **<ID>**.



3. Click the right arrow button (▶) in the right panel to add the ID flipper as a column. If mapping the ID, skip to the last step.
4. If mapping a key, open the <ID> flipper, and under **Value**, click the **Transformation** icon (⚙️). In the **Transformations** dialog, select the relevant key from the **Aspect** list and click **Save**.

Map Data

<ID>

<Name>

<Parent ID>

<Object Type Name>

<Product-Override Child ID>

<AttributLinks>

<Is deleted>

"Constant Value"

<Page Number>

+ All Attributes

Select Attribute

+ Classifications

+ Index Words

+ Product Classification Links

+ Product References

+ Asset References

+ Classification References

+ Entity References

+ STEP Workflow Task Info

Exports data in Excel format.

Column (1 mapped)

<ID> ID

Header "<ID>"

Value <ID> ID

Transformations

Target: Value

Source: <ID> ID

Aspect: ID

ID

Add Transformation

Key: Key 1

Key: Key2

Key: Key 3

Key: Supplier Part Number

Save

5. For both ID and Key mappings, finally, complete any additional mappings and initiate the export.

Results

The following data is exported with the results displayed below, based on mapping the ID or 'Supplier Part Number' key.

<

Products

Products

References

Referenced By

View: ManfName

	ID >	Supplier Part Number >
Name >	ID	Supplier Part Number
T-Shirts Family >	123853	678-551346
123857 G >	123857	678-21336
123854 B >	123854	678-21334
123855 O >	123855	678-21335

If only the ID is mapped, the exported file includes the IDs of the objects.

<ID>
123853
123854
123855
123857

If only the 'Supplier Part Number' active key is mapped, the exported file includes the ID
KeyID=SupplierPartNumber as header and its corresponding key values for each of the exported objects.

<ID KeyID=SupplierPartNumber>
678-551346
678-21334
678-21335
678-21336

Index Words – Data Source Outbound

The option to map index words is available for an index hierarchy. For more information on indexes, see **Creating Document Indexes** in the **STEP'n'design** documentation.

Index words that are linked to products display on the product's References Tab under the Index Words flipper.

Tree

Apparel

Upper Body Wear

Head Wear

Hats and Caps

Hats and Caps Item

(123118)

(123119)

109011

123117

20801

20805

20805 rev.0.17 - References

ProductSub ProductsReferencesReferenced ByImages & Document

Index Words

Inherited From	Index	Level 1
> Locally defined	Indexdoc	Index Words/Indexdoc/Epuda quam
> Locally defined	Indexdoc	Index Words/Indexdoc/Hanes Baseball Cap
> Locally defined	Indexdoc	Index Words/Indexdoc/harion
> Add Index Word		

Publications

Linked Attributes from Product Hierarchy

Index words modeled as classification objects are displayed in the classifications section of the Tree. The Sort Word attribute can be used to order an index non-alphabetically.

Tree

Assets

Configurations

Index Words

Indexdoc

Acme Baseball Cap

Epuda quam

Hanes Baseball Cap

harion

Acme Baseball Cap rev.0.1 - Index Word

Index WordSub ProductsReferencesReferenced ByStatusState Log

Description

Name	Value
> ID	Acme Baseball Cap
> Name	Acme Baseball Cap
> Sort Word	abc10

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

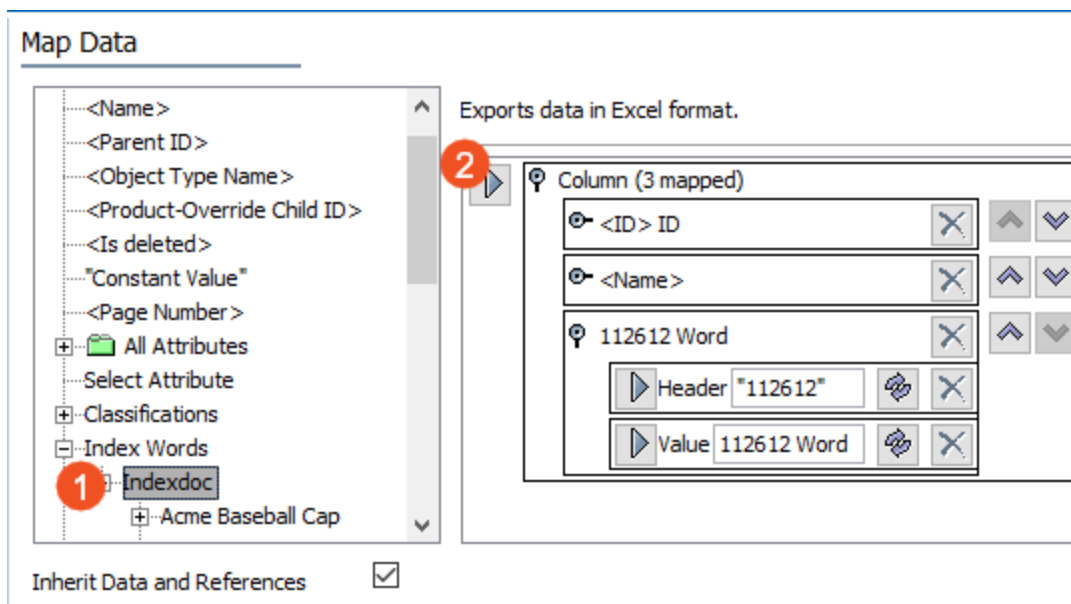
The index words of an index hierarchy can be exported as follows:

- Index Words Linked to Products Using Word - exports the index word
- Index Words Linked to Products Using Sort Word - exports the sort word attribute value
- Index Words from Classification - exports all index words in the selected hierarchy

Each of these options is defined below.

Mapping Index Words Linked to Products Using Word

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. Select the Product super type. For more information, see **Export Manager - Select Objects**, the **Configure the Format** section of the **OIEP - Event-Based - Output Templates Flipper**, or the **Configure the Format** section of the **OIEP - Select Objects - Output Templates Flipper**.
3. Ensure products with index word references are selected for output.
4. On the Map Data step, in the left panel, expand Index Words, and select the desired index words.



5. Click the right arrow button (▶) in the right panel to add the selected index words type as a column. The added column displays that Word is being used for export. Add additional index words, if needed.
6. Apply any transformations, which can change the output without changing the original data. See the **Outbound Map Data - Transform** topic.
7. Complete any additional mappings and initiate the export.

Results of Index Words Linked to Products Using Word

The output includes the ID and name (of product), and index word ID as headers.

When the multiple index words are linked to a product, the index words names are displayed in a semi-colon separated list.

	A	B	C
1	<ID>	<Name>	112612
2	121933	121933	Acme Baseball Cap
3	20805	20805	Hanes Baseball Cap;Epuda quam;harion

Mapping Index Words Linked to Products Using Sort Word

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. Select the Product super type. For more information, see **Export Manager - Select Objects**, the **Configure the Format** section of the **OIEP - Event-Based - Output Templates Flipper**, or the **Configure the Format** section of the **OIEP - Select Objects - Output Templates Flipper**.
3. Ensure products with index word references are selected for output.
4. On the Map Data step, in the left panel, expand Index Words, and select the desired index words.

Map Data

Exports data in Excel format.

Column (3 mapped)

- <ID> ID
- <Name>
- 112612 Sort Word
 - Header "112612"
 - Value 112612 Sort Word

Transformations

Target: Value

Source: 112612 Sort Word

Aspect: Sort Word

Multi-value separator: ;

Add Transformation

Save Reset Cancel

5. Click the right arrow button in the right panel to add the selected index words type as a column. Add additional index words, if needed.
6. Open the flipper for the index words that should use the sort word parameter. On the Value element, click the **Transformation** button ().

- On the Transformations dialog, for the Aspect parameter, select **Sort Word** from the dropdown. Click the **Save** button. The mapped column now displays Sort Word.
- Complete the mapping and initiate the export.

Results of Index Words Linked to Products Using Sort Word

The output includes the ID and name of product, and index word ID as headers. Data includes the sort word attribute value defined on the index word that is linked to the product.

When the multiple index words are linked to a product, the sort words values are displayed in a semi-colon separated list.

	A	B	C
1	<ID>	<Name>	112612
2	121933	121933	10
3	20805	20805	15;80;50

Mapping Index Words from Classification

- Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
- Select the Classification super type. For more information, see **Export Manager - Select Objects**, the **Configure the Format** section of the **OIEP - Event-Based - Output Templates Flipper**, or the **Configure the Format** section of the **OIEP - Select Objects - Output Templates Flipper**.
- On the Map Data step, in the left panel, select any information necessary for the classification

Map Data

<ID>

<Name>

<Parent ID>

<Object Type Name>

<Is deleted>

"Constant Value"

<Page Number>

+ All Attributes

Inherit Data and References ☒

Exports data in Excel format.

Column (2 mapped)

<ID> ID

<Name>

- Complete the mapping and initiate the export.

Results of Index Words from Classification

The output includes the mapped information of all index words in the selected hierarchy.

	A	B
1	<ID>	<Name>
2	Acme Baseball Cap	Acme Baseball Cap
3	Hanes Baseball Cap	Hanes Baseball Cap
4	Epuda quam	Epuda quam
5	harion	harion
6	polyester	polyester

Inherit Data and References - Data Source Outbound

The 'Inherit Data and References' option shown below is the default setting and is global, in that it applies to all attribute values or references selected in the Map Data window. When checked, the output includes values that are inherited from objects at a higher level in the hierarchy.

For general information on inheritance, see the Inheritance in the **Product Hierarchy** topic in the **Getting Started / User Guide** documentation.

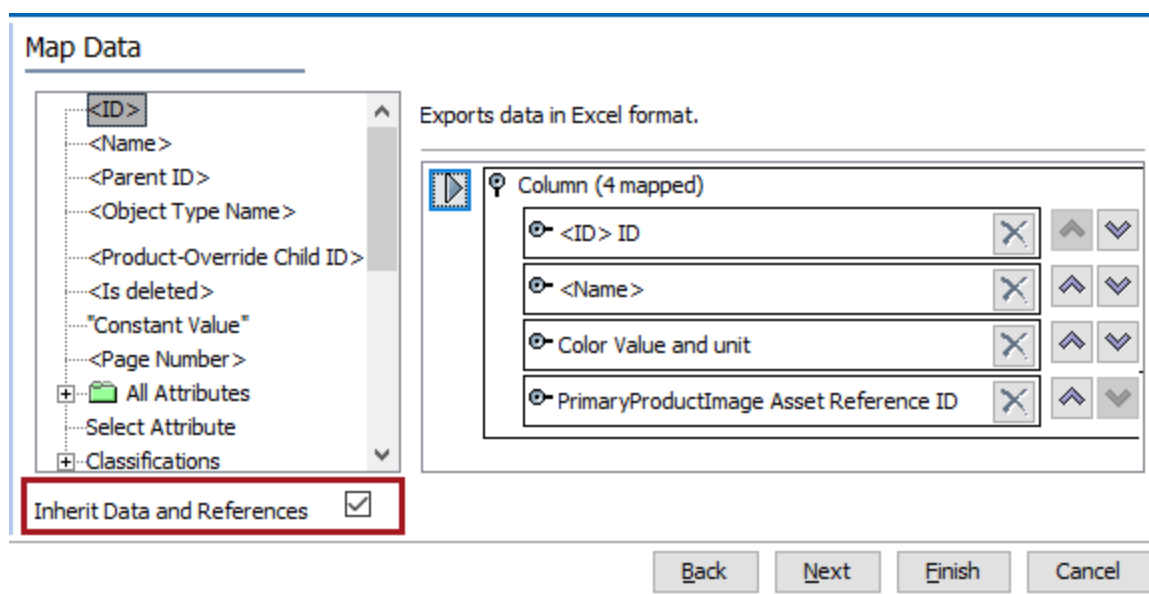
The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Inherited Data and References

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, below the left panel, check the **Inherit Data and References** option appropriately to affect all attribute values or references mapped:
 - If **checked**, inherited data and references are included in the export.
 - If **unchecked**, inherited data and references is not included in the export.



The screenshot shows the 'Map Data' window. On the left, a tree view lists various attributes: <ID>, <Name>, <Parent ID>, <Object Type Name>, <Product-Override Child ID>, <Is deleted>, "Constant Value", <Page Number>, All Attributes, Select Attribute, and Classifications. At the bottom left, the 'Inherit Data and References' checkbox is checked and highlighted with a red rectangle. On the right, a section titled 'Exports data in Excel format.' contains a table with 4 mapped columns: <ID> ID, <Name>, Color Value and unit, and PrimaryProductImage Asset Reference ID. Each row has a delete button (X) and up/down arrows. At the bottom right are buttons for Back, Next, Finish, and Cancel.

- 3. Use the necessary data sources to map the required data. For information, see **Outbound Map Data - Data Source**.
- 4. Initiate the export.

Results

The following data is exported with the results displayed below, based on the 'Inherit Data and References' parameter.

The Color attribute has a local value on the family object, and the values are inherited to the same attribute of the child object. The following shows a view created to show the values available for both the parent and the child.

Tree

T-Shirts Family

123855 O

123854 B

T-Shirts Sales Items

Head Wear

Footwear

Safety

Hardware

Displays

Furniture

Automotive

Products

ProductsReferencesReferenced By

View: Color

	ID	Color
Name	ID	Color
T-Shirts Family	123853	Blue (RYB) Orange (RYB) Green (RYB)
123855 O	123855	Blue (RYB) Orange (RYB) Green (RYB)

Inheritance can be verified by the viewing the attribute on the editor for the child, and hovering over the icon on the attribute.

Blue (RYB)

Orange (RYB)

Green (RYB)

Text, MultiValued, Inherited, Inherited from other context

The 'Primary Product Image' reference value is also being inherited to the child level in the hierarchy.

ProductsReferencesReferenced By

Image References

Reference TypePrimary Product Image

Source	Reference Type	Target	Thumbnail
T-Shirts Family	Primary Product Image	Hanes Family	
123855 O	Primary Product Image	Hanes Family	

Asset Reference, MultiValued, Inherited

When the 'Inherit Data and References' parameter is not checked, and the child object is exported, only the values and the references local on the child object are exported. In this case, all values were inherited, so no values are exported.

<ID>	<Name>	Color	Primary Product Image Asset Reference ID
123855	123855 O		

When the 'Inherit Data and References' parameter is checked, and the child object is exported, the values and the references are inherited to the child object.

<ID>	<Name>	Color	Primary Product Image Asset Reference ID
123855	123855 O	Blue (RYB);Orange (RYB);Green (RYB)	20585

Insert Referenced Objects - Data Source Outbound

The 'Insert Referenced Objects' data source exports all objects referenced from the selected reference type before the object that is selected to be exported. The column added by this mapping contains the ID of the owner of the reference, in order to see that the objects belong together.

This option can be used to export objects referenced from asset references, product references, classification references, and entity references. All the reference types from the System Setup > Reference Types except Link Types are available for export with this option.

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

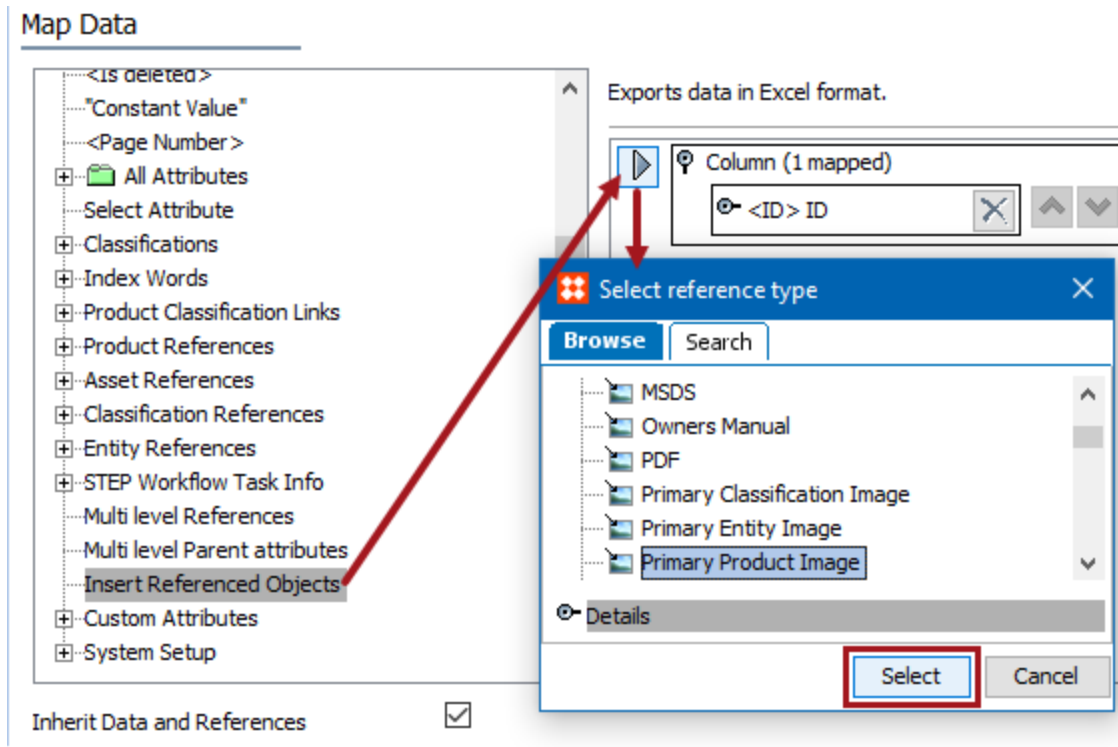
For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Insert Referenced Object

Examples of a variety of exports are included in the **Results** section below.

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, select **Insert Referenced Object**, click the right arrow button () in the right panel, select the desired reference type, and click the **Select** button.



3. Check the **Inherit Data and References** option appropriately:

- If **checked**, inherited data and references are included in the export.
- If **unchecked**, inherited data and references is not included in the export.

For information on inheritance, see the Inheritance in the **Inherit Data and References - Data Source Outbound** topic.

4. Apply any transformations, which can change the output without changing the original data. For example, exporting the asset reference type name as a header and asset name as the value. See the **Aspect - Transform Outbound** topic. Example outputs are included in the **Results** section below.
5. Complete any additional mappings and initiate the export.

Results

In this example of Asset References, the product being exported has three references to images:

Tree					
Electronics		Product			
Televitions		Sub Products			
LCD		References			
P7000		Referenced By			
AC-P7000-55		Image			
AC-P7000-65					
AC-P7000-79					
AC-P7000-83					

Image References					
>	Primary Product Image	+		107629	
>	Product Image	+		107625	
				107628	

Using the 'Insert Referenced Objects' option, the export shows the product ID once under the <ID> column and again for each of the two Ref Owner ID columns.

	A	B	C	D
1	<ID>	<Name>	Ref Owner ID of PrimaryProductImage	Ref Owner ID of ProductImage
2	107628	107628	179626	179626
3	107625	107625	179626	179626
4	107629	107629	179626	179626
5	179626	AC-P7000-79	179626	179626

In this example, the product being exported has a Classification Reference of type 'Class2Class', a Product Reference of type 'Bill of Materials', and an Entity Reference of type 'EntityRef'.

Tree

- Polo T-shirt
 - Polo A
- RoundNeck - T-shirts
 - T-Shirts Sales Items
- Head Wear
- Footwear
- Safety
- Hardware
- Displays
- Furniture
- Automotive
- Building Products
- Electrical & Electronics

Polo A rev.0.7 - References

- Product
- Sub Products
- References**
- Referenced By
- Ima

Display

Reference Type	Target
Class2Class	T-shirts and Sweatshirts

Item References, Product

Reference Type	Target	Qty
Bill of Materials	Item 1	

Ungrouped Entity References

Reference Type	Target	Con
EntityRef	12 North Franklin St	

Using the 'Insert Referenced Objects' option, the export shows the product ID once under the <ID> column and again for each of the three Ref Owner ID columns.

	A	B	C	D	E
1	<ID>	<Name>	Ref Owner ID of ClassificationReference	Ref Owner ID of BillOfMaterials	Ref Owner ID of EntityRef
2	18212	Item 1	113202	113202	113202
3	121605	12 North Franklin St	113202	113202	113202
4	22586	T-shirts and Sweatshirts	113202	113202	113202
5	113202	Polo A	113202	113202	113202

Is Deleted - Data Source Outbound

Although displayed for all formats, the '<Is deleted>' data source can be used by Generic XML in OIEPs as described in the following mapping examples:

- Reporting Deleted Assets with a Generic XML OIEP
- Reporting Deleted Classifications with a Generic XML OIEP
- Reporting Deleted Entities with a Generic XML OIEP
- Reporting Deleted Product References with a Generic XML OIEP

For additional 'is deleted' functionality, see the 'Is Deleted' option available in the **Aspect - Transform Outbound** topic.

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

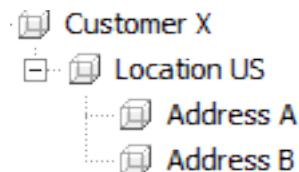
For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Multi Level Parent Attributes - Data Source Outbound

The Multi Level Parent Attributes option enables the exporting of data on parent objects as a separate column for each object being exported. This option is available for exporting parent attribute values for product, entity, and classification objects.

For example, a customer hierarchy is created in STEP using entity nodes. Addresses are grouped into locations.



When this customer hierarchy is exported, you can export attribute values on parent nodes. This means that the row that holds Address A and Address B could also include values that only exist on the Location US or Customer X entity objects.

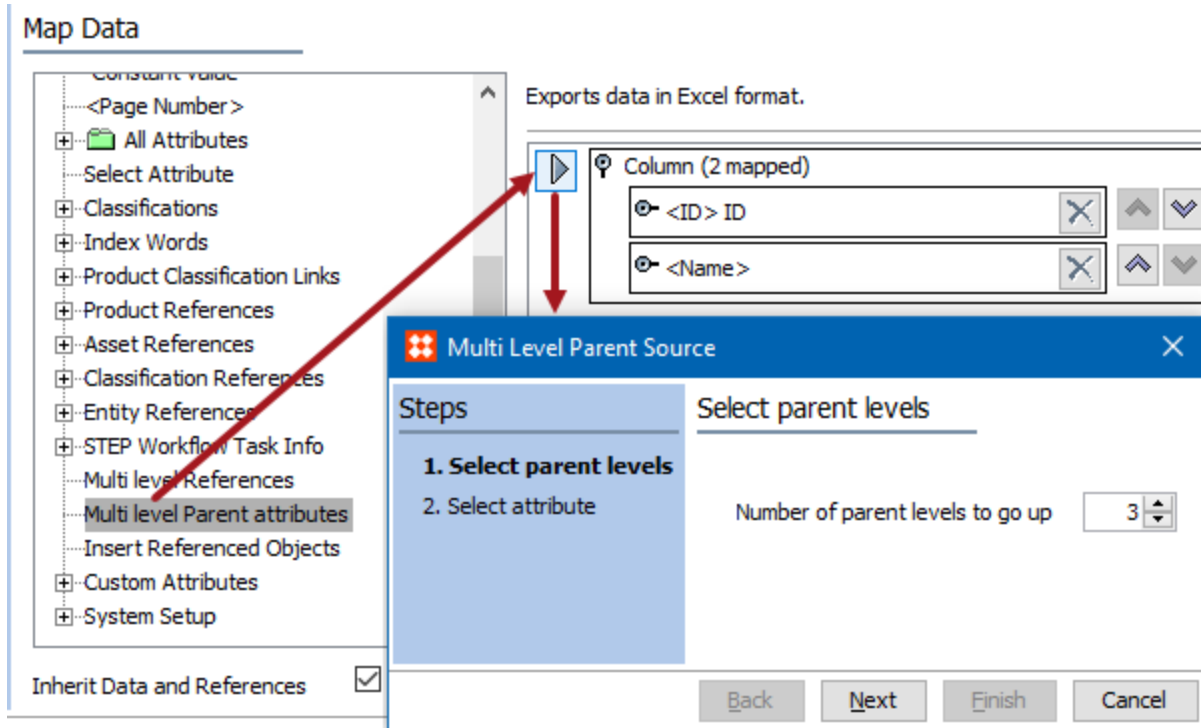
The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

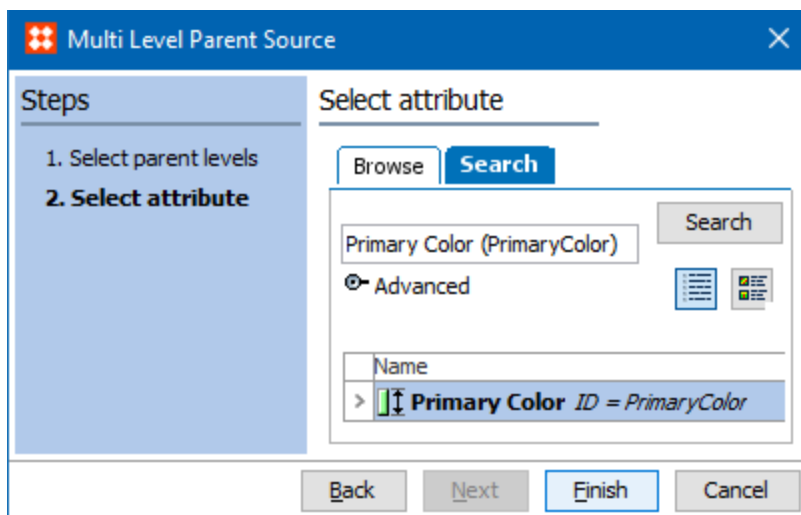
After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Multi Level Parent Attributes

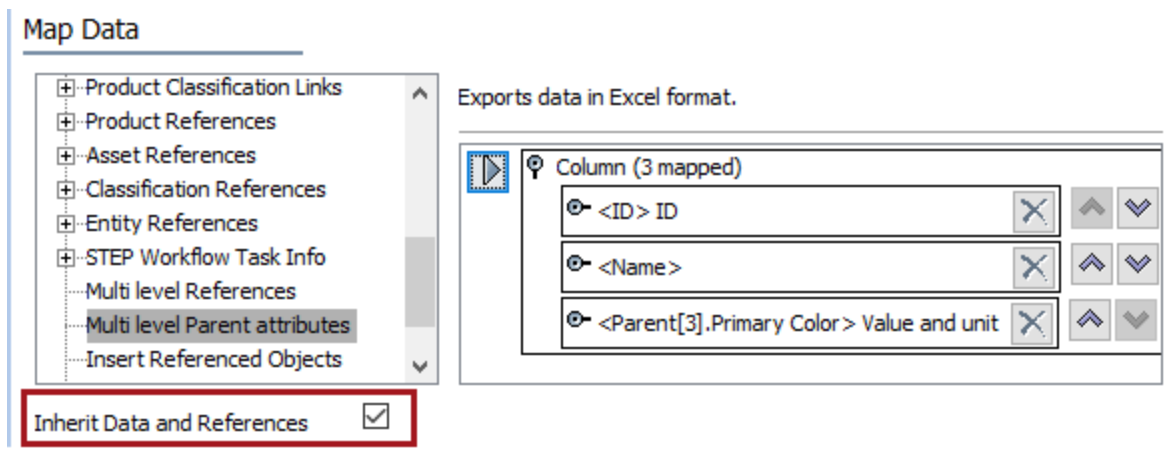
1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, select the **Multi level Parent attributes** option, and in the right panel, click the arrow mapping button () to display the Multi Level Parent Source dialog.



3. For the **Select parent levels** step, specify the number of parent levels to go up within the hierarchy, and click **Next**. The default of '1' indicates that only the value from the immediate parent is exported. Increasing this setting exports the value for the parent at the specified level, as shown above with the setting of '3'.
4. For the **Select attribute** step, use **Browse** or **Search** to locate the attribute to be exported, and click **Finish**.

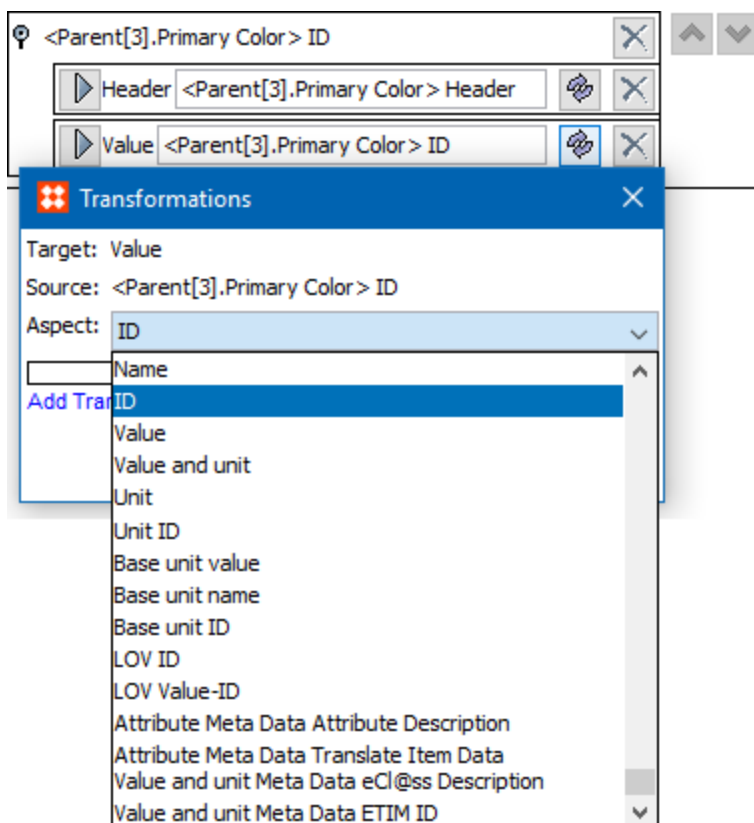


5. Back on the Map Data dialog, check the **Inherit Data and References** option appropriately:
 - If **checked**, inherited data and references are included in the export.
 - If **unchecked**, inherited data and references is not included in the export.



For information on inheritance, see the Inheritance in the **Inherit Data and References - Data Source Outbound** topic.

6. If needed, change the asset reference Value Aspect to output data other than the asset reference ID. For details on the options, see the **Aspect - Transform Outbound** topic.



- Open the asset reference flipper to display the Header and Value elements.
 - Click the transformation button () for the Header or for the Value element to display the Transformations dialog and the Aspects parameter.
 - Click the **Aspect** dropdown to display and select an option.
 - Click the **Save** button to apply any aspect changes.
7. Apply any transformations necessary to change the output without changing the original data. See the **Outbound Map Data - Transform** topic for details.
8. Complete any additional mappings and initiate the export.

Results

The 'Primary Color' attribute holds different data for each of the selected levels in the hierarchy, as displayed in the view named 'Color'. The three parent levels are indicated by the white numbers in the orange circles.

Tree

Primary Product Hierarchy

Product Root

Apparel

Upper Body Wear

Head Wear

Hats and Caps

Hats and Caps Items

109011

123117

20801

Products

References

Referenced By

View: Color

	ID	Primary Color	
Name	ID	Primary Color	
Head Wear	20433	Yellow	3
Hats and Caps	20435	Green	2
Hats and Caps Items	20436	Blue	1
109011	109011	White	

When exported for a multi-level parent setting of '3', the following is output, which includes the value of the exported object, as well as the third parent level up the hierarchy.

<ID>	<Name>	<Parent[3].Primary Color>
109011	109011	Yellow

© Stibo Systems - Confidential - STEP Trailblazer 8.3

590

Multi Level References - Data Source Outbound

The multi-level references data source option is available for all formats that require mapping. This allows you to map data from referenced / ancestor objects. You can also map so that one or more additional rows are generated for referenced objects. This option is available and mapped the same for product, classification, asset, and entity objects.

For example, a customer hierarchy is created in STEP using entity nodes. Each customer has a reference of the type Region to a Location. Each Location has a reference of type Address to specific addresses. When the customer hierarchy is exported, it is possible to follow the references and export data from the address objects.



Using the Multi Level References option allows the following information from the object to be exported:

- Meta data attributes on references with values.
- Names of referenced objects.
- Attribute values on referenced objects.

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

To demonstrate this data source option, this topic includes the following sections:

- Handling Multiple References
- Mapping Multi Level References
- Results

Handling Multiple References

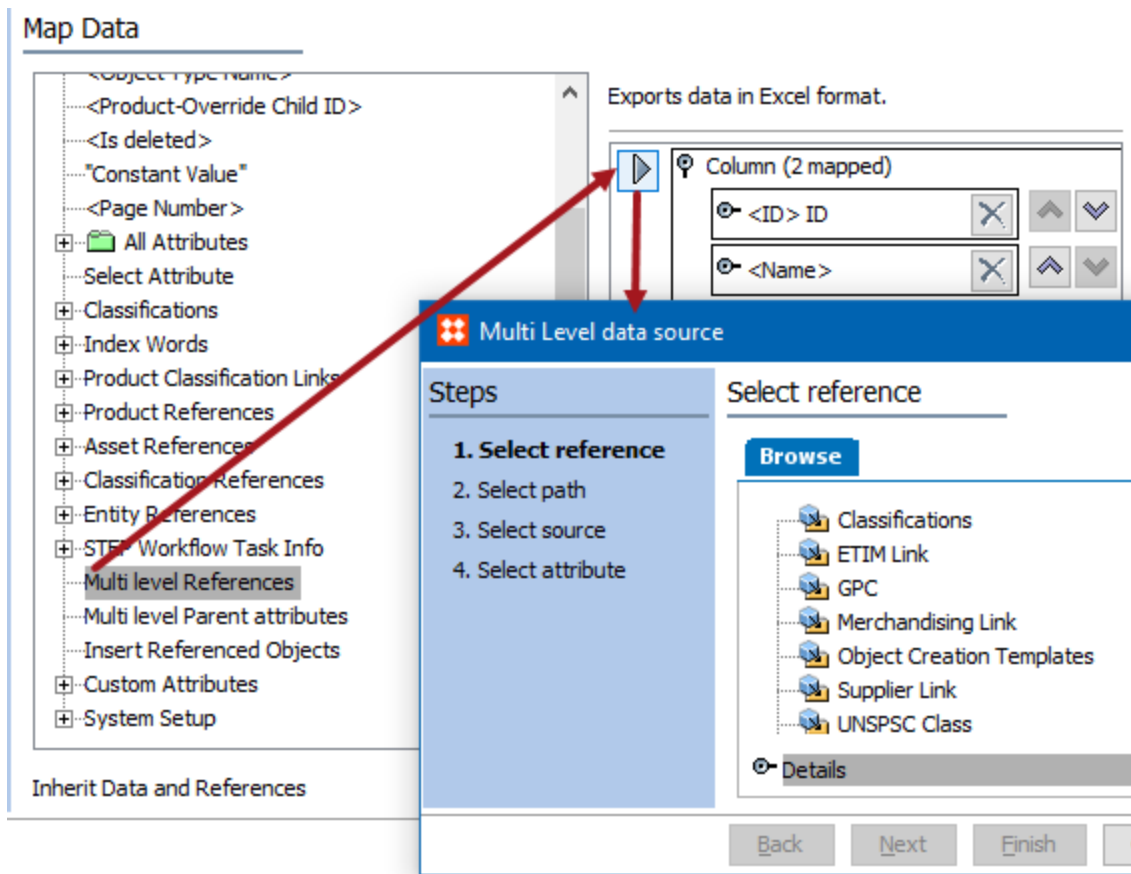
When multiple references exist, the options for the output are:

- All references in a single field, separated - This is the default setting and is achieved using the Multi Valued option.
- One reference per field - Using the Item option repeatedly, each reference can be output in individual fields.

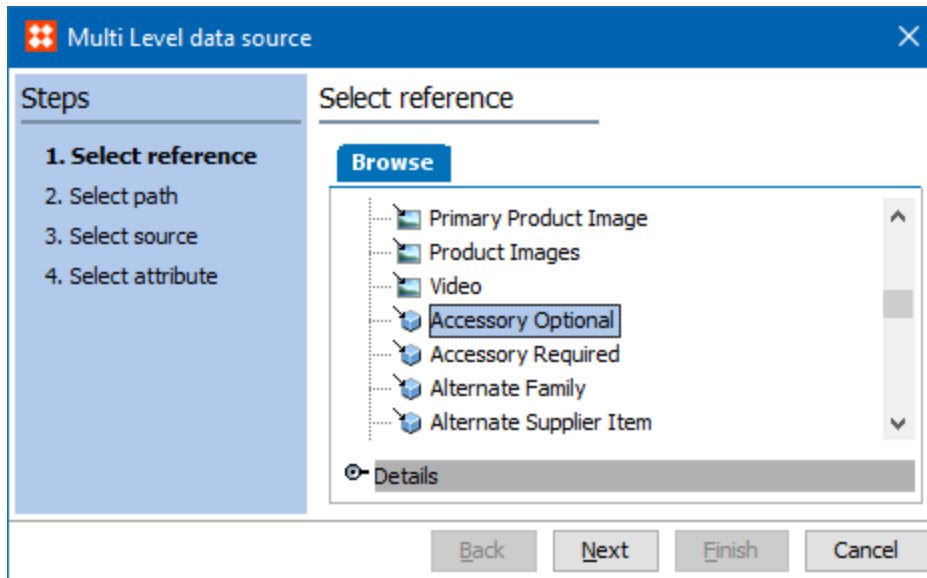
Details for using each option is described below.

Mapping Multi Level References

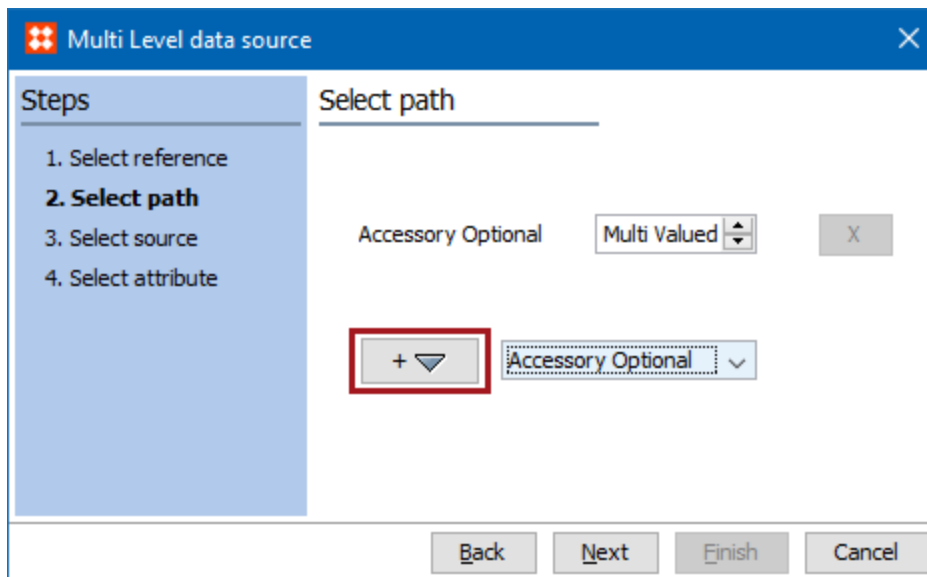
1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, select **Multi level References** data source.
3. In the right panel, click the right arrow icon to display the **Multi Level data source** wizard.



4. In **Select reference**, highlight the relevant source reference type and click **Next**.



5. In **Select path**, the legal reference types available are displayed in the dropdown. Select the required reference type. To add an additional reference type, click the plus / down arrow button, .



6. Select the necessary reference levels and click **Next**.

Based on selected reference, the following options may be available:

- **Single valued** - exports data from the referenced object of the selected type where only one is allowed.

Open Item Single valued 

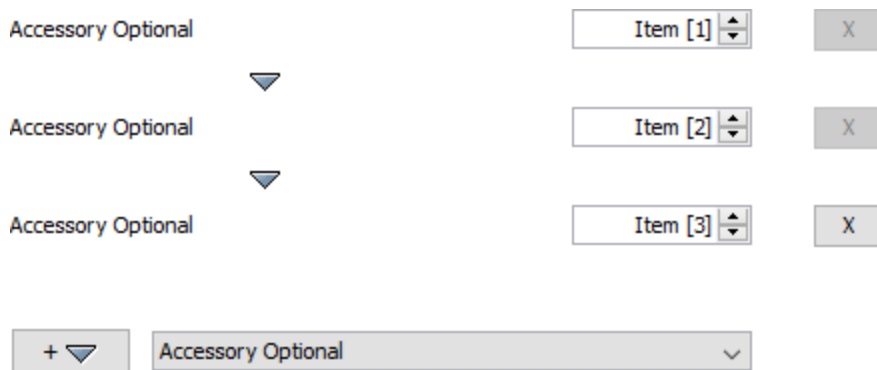
- **Multivalued** - exports data from all referenced objects of the selected type where multiple references are allowed. All reference values are output in a single field, with a separator character.



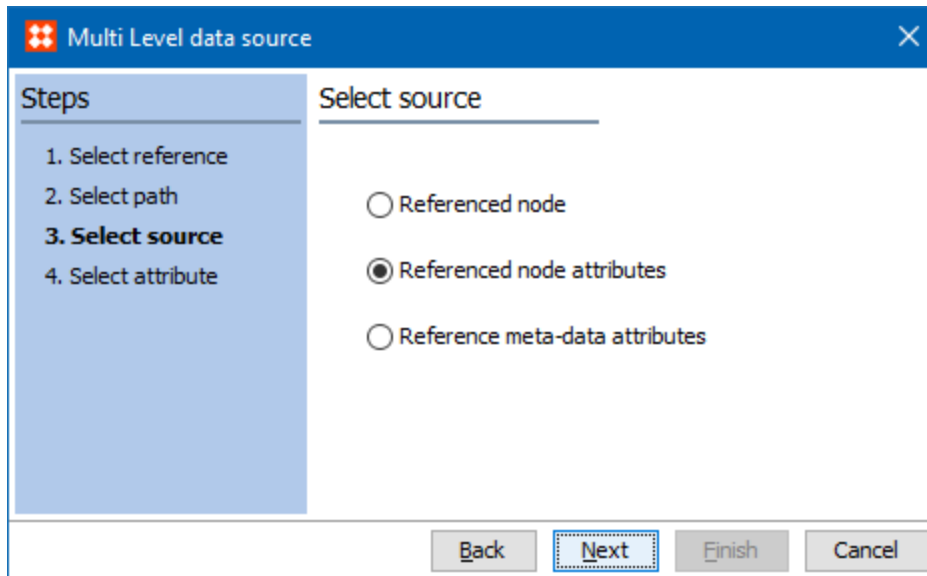
- **Item []** - exports data from only a single referenced object, determined by the selected Item number. For example, Item [1] only exports data from the first referenced object.



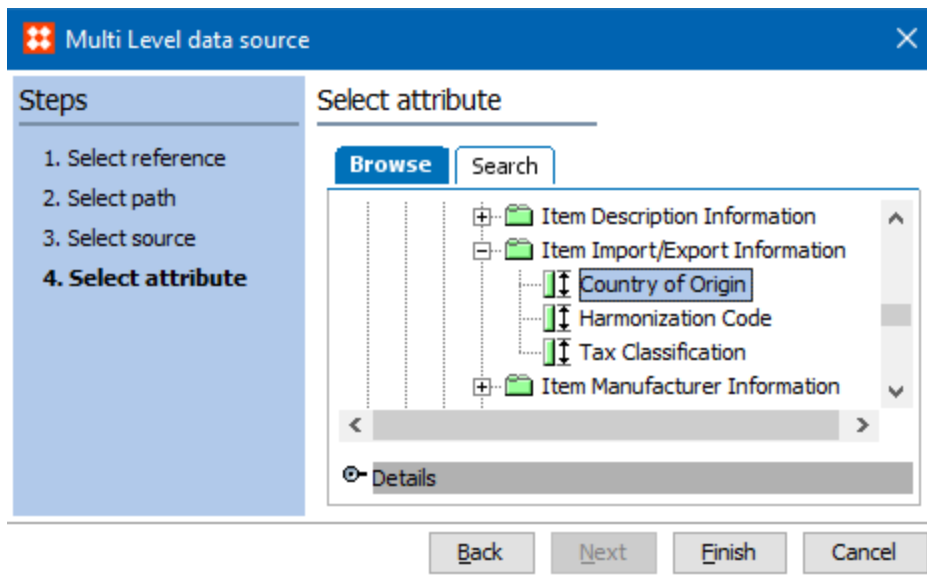
- **Item [] with multiple references** - exports multiple references in individual fields (not a single, separated field as with the Multivalued option), first determine the maximum number of possible instances for the selected reference type per referencing object. Then change Multi Valued to Item [] option and use the plus / down arrow button, , until the maximum number is displayed.



- **X** - Use the delete button to remove any unnecessary levels.
7. In **Select Source**, choose the type of data to be exported:
- **Referenced node** exports the names of the objects. The Select Attribute screen is disabled.
 - **Referenced node attributes** exports attribute values on the referenced objects for the selected attribute. Click **Next** to display the **Select Attribute** screen.
 - **Reference meta-data attributes** exports attribute values applied to the reference. Click **Finish**.

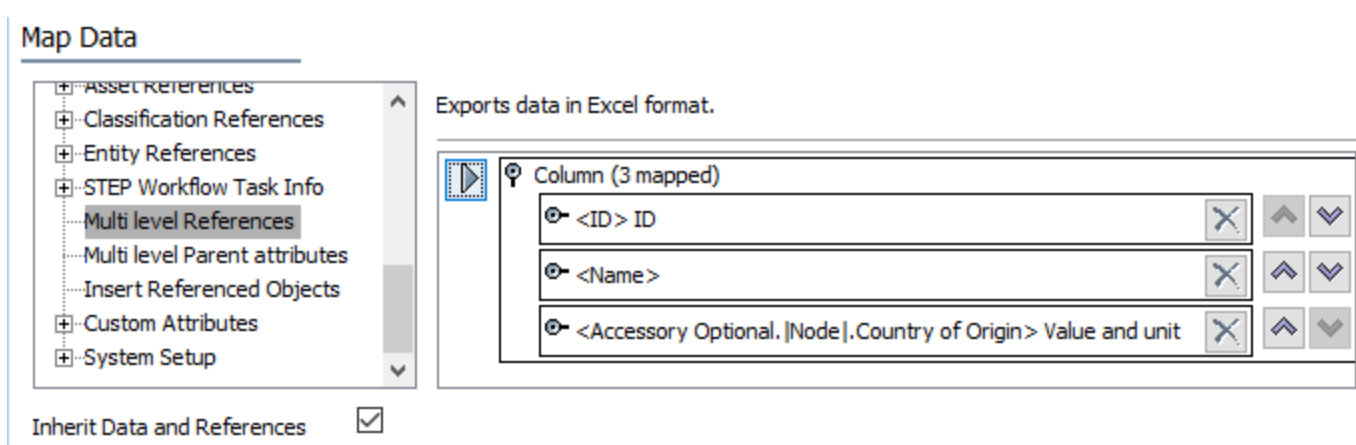


8. If needed, in **Select Attribute**, use **Browse** or **Search** to locate the attribute that holds the values to be exported and click **Finish**.



The multi-level reference mapping is complete and displayed on the Map Data dialog.

9. Complete any additional mappings and initiate the export.



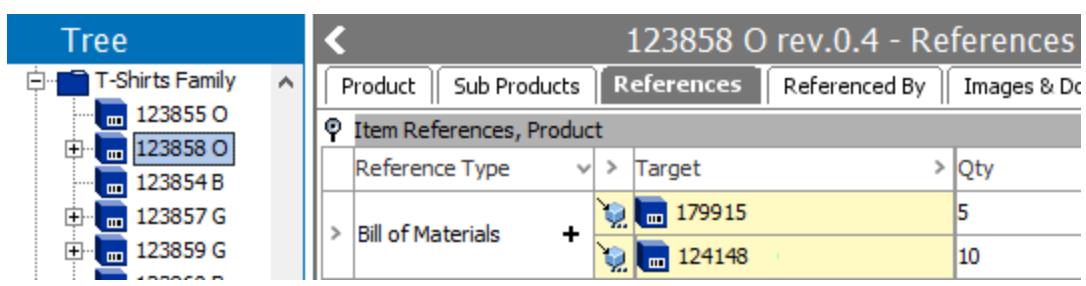
The single object output includes the values for both of the Accessory Optional references. Although this example used the Excel format, any other format would produce the same output. In the following image, '|Node|.' is the referenced product of the exported product. This is included since on the **Select source** step, we chose 'Referenced node attributes' and then for the **Select attribute** step, we selected 'Country of Origin'.

	A	B	C
1	<ID>	<Name>	<Accessory Optional. Node . Country of Origin>
2	20805	Baseball Cap	SPAIN;CHINA

Results

The following section includes examples of exporting for multi- and single-valued reference types.

Export a product with references to a multi-valued reference type. For example, the product below has multiple references via the 'Bill of Materials' reference type, and includes metadata attribute 'Qty'.



When mapping the **Multi Level References** data source:

- On the **Select reference** step, choose the 'Bill of Materials' reference type.
- On the **Select path** step, set 'Bill of Materials' as 'Multi Valued'.
- On the **Select source** step, choose 'Referenced node'.

The final mapping is displayed as follows:

Map Data

+

Index Words

+

Product Classification Links

+

Product References

+

Asset References

+

+

+

Multi level References

Multi level Parent attributes

Insert Referenced Objects

+

+

Exports data in Excel format.

▶

Column (3 mapped)

⊖

<ID> ID

×

⬆

⬇

⊖

<Name>

×

⬆

⬇

⊖

<Bill of Materials> ID

×

⬆

⬇

▶

Header <Bill of Materials> Header

⊗

×

▶

Value <Bill of Materials> ID

⊗

×

Once the export is initiated, the output includes the originally selected object ID and name, as well as the ID for each of the multiple references, separated by a semicolon (;).

	A	B	C
1	<ID>	<Name>	<Bill of Materials>
2	124458	123858 O	179915;124148

Export the same product again, but this time when mapping the **Multi Level References** data source:

- On the **Select reference** step, choose the 'Bill of Materials' reference type, as we did previously.
- On the **Select path** step, set 'Bill of Materials' as 'Multi Valued', as we did previously.
- On the **Select source** step, now choose 'Reference meta-data attributes'.
- On the **Select attribute** step, now choose the 'Qty' attribute.

The final mapping is displayed as follows:

Map Data

+

Index Words

+

Product Classification Links

+

Product References

+

Asset References

+

+

+

Multi level References

Multi level Parent attributes

Insert Referenced Objects

+

+

Exports data in Excel format.

▶

Column (3 mapped)

⊖

<ID> ID

×

⬆

⬇

⊖

<Name>

×

⬆

⬇

⊖

<Bill of Materials.Qty> Value and unit

×

⬆

⬇

▶

Header <Bill of Materials.Qty> Header

⊗

×

▶

Value <Bill of Materials.Qty> Value and unit

⊗

×

Export the same product again, but this time when mapping the **Multi Level References** data source:

- On the **Select reference** step, choose the 'Bill of Materials' reference type.
- On the **Select path** step, set 'Bill of Materials' as 'Item [1]'.
- On the **Select source** step, choose 'Reference meta-data attributes'.
- On the **Select attribute** step, choose the 'Qty' attribute.

The final mapping is displayed as follows:

Map Data

Classifications

Index Words

Product Classification Links

Product References

Asset References

Classification References

Entity References

STEP Workflow Task Info

Multi level References

Multi level Parent attributes

Insert Referenced Objects

Custom Attributes

System Setup

Exports data in Excel format.

Column (3 mapped)

<ID> ID

<Name>

<Bill of Materials[1].Qty> Value and unit

Header <Bill of Materials[1].Qty> Header

Value <Bill of Materials[1].Qty> Value and unit

Once the export is initiated, the output includes the originally selected object ID and name, as well as the value for the first 'Qty' attribute metadata.

	A	B	C
1	<ID>	<Name>	<Bill of Materials[1].Qty>
2	124458	123858 O	5

The same functionality is available for single-valued reference types, where only one item can be referenced. The following product includes a reference to another product with ID 100703 via the 'Supplier Replacement Item' reference type. No additional references of this type are allowed, as indicated by the absence of the + sign following the reference type.

Tree

T-Shirts Family

123855 O

123858 O

123854 B

123857 G

123859 G

123857 G rev.0.8 - References

Product

Sub Products

References

Referenced By

Item References, Product

Reference Type

Target

Qty

> Bill of Materials

+

> Supplier Replacement Item

100703

10

When mapping the **Multi Level References** data source:

- On the **Select reference** step, choose the 'Supplier Replacement Item' reference type.
- On the **Select path** step, leave 'Supplier Replacement Item' as 'Single Valued'.
- On the **Select source** step, choose 'Referenced node'.

The final mapping is displayed as follows:

Map Data

Exports data in Excel format.

Column (3 mapped)

<ID> ID	X	↑	↓
<Name>	X	↑	↓
<Supplier Replacement Item> ID	X	↑	↓
Header <Supplier Replacement Item> Header	X		
Value <Supplier Replacement Item> ID	X		

Once the export is initiated, the output includes the originally selected object ID and name, as well as the ID for the 'Supplier Replacement Item' referenced object.

	A	B	C
1	<ID>	<Name>	<Supplier Replacement Item>
2	123857	123857 G	100703

Exporting the same object again, but on the **Select source** step, we chose 'Referenced node attributes' and then for the **Select attribute** step, we selected 'Country of Origin' results in the following. As before, '.|Node|.' indicates the selected source was 'Referenced node attributes'.

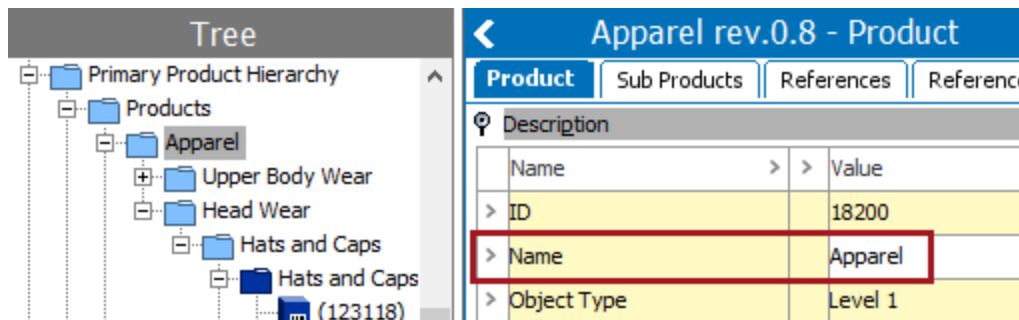
	A	B	C
1	<ID>	<Name>	<Supplier Replacement Item. Node .Country of Origin>
2	123857	123857 G	ANTIGUA AND BARBUDA

Exporting the same object again, but on the **Select source** step, we chose 'Reference meta-data attributes' and then for the **Select attribute** step, we selected 'Qty' results in the following.

	A	B	C
1	<ID>	<Name>	<Supplier Replacement Item.Qty>
2	123857	123857 G	10

Name - Data Source Outbound

The option to map a name allows output of the value of the Name attribute for the selected objects.



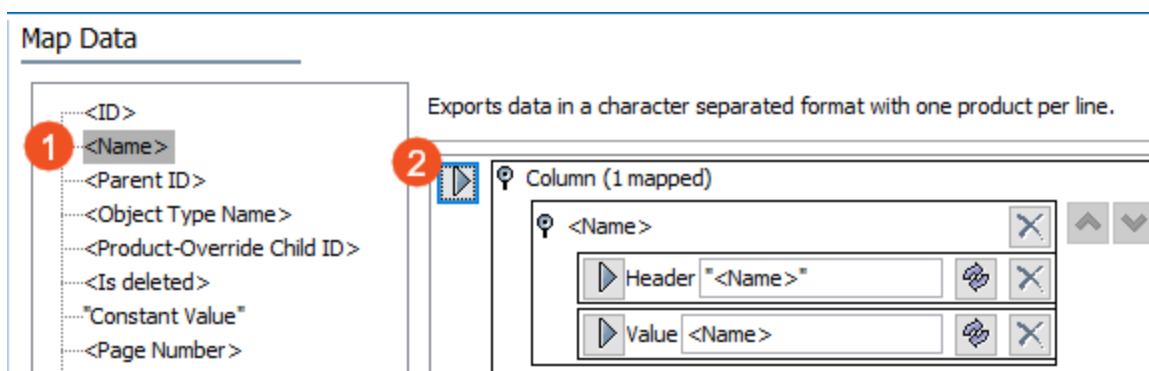
The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Name

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, select the <Name> node and click the mapping button (▶).



3. Apply any transformations, which can change the output without changing the original data. See the **Outbound Map Data - Transform** topic.
4. Complete any additional mappings and initiate the export.

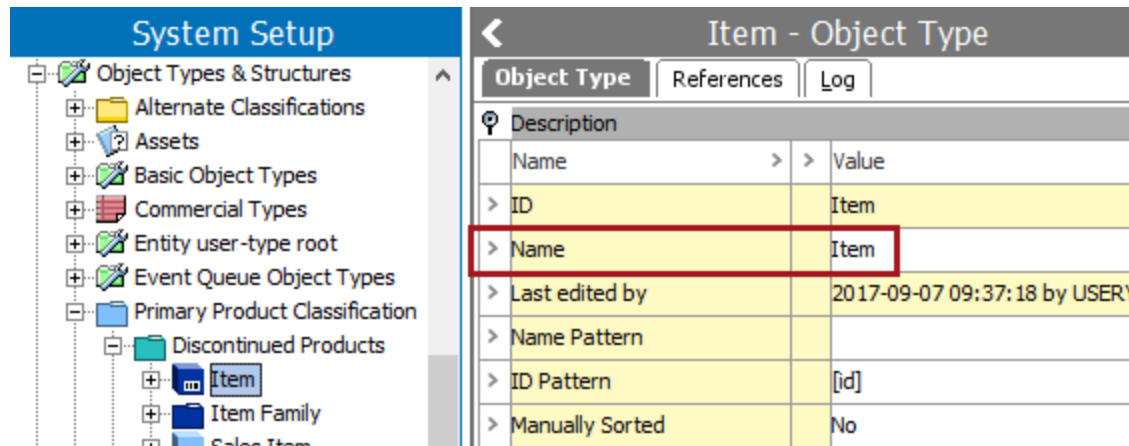
Results

The output includes <Name> as a header and the value for each row.

	A
1	<Name>
2	18210 M B_EN
3	18212 L B
4	18213 M O
5	18216 L O

Object Type Name - Data Source Outbound

The option to map the name of a STEP object type can allow a downstream system to identify the type of data being received.



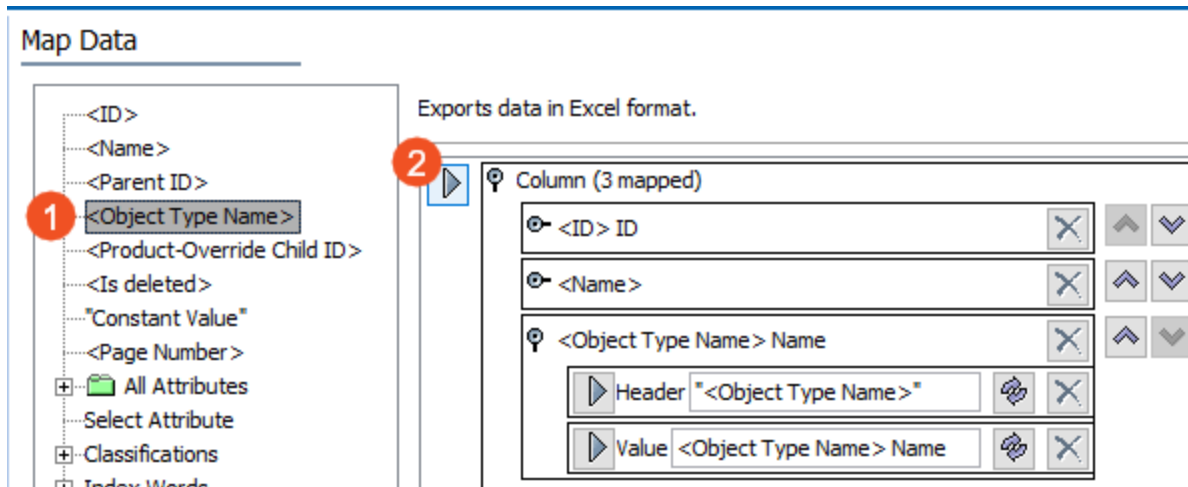
The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Object Type Name

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. Select the super type for the desired object types. For more information, see **Export Manager - Select Objects**, the **Configure the Format** section of the **OIEP - Event-Based - Output Templates Flipper**, or the **Configure the Format** section of the **OIEP - Select Objects - Output Templates Flipper**.
3. On the Map Data step, in the left panel, select the <Object Type Name> node and click the right arrow button.



4. Click the Transformation button () to apply any transformations, which can change the output without changing the original data.

Note: While the default selection is 'Object Type Name' for export, when importing the same file, the auto-mapping feature expects the 'Object Type ID'. To generate an export file that is prepared for reimport, use the Transformation Aspect to map object type ID. For more information, see the **Aspect - Transform Outbound** section of the **Data Mapping** documentation.

5. Complete any additional mappings and initiate the export.

Results

The output includes <Object Type Name> as a header (or Object Type ID if transformed) and the value for each row.

	A	B	C
1	<ID>	<Name>	<Object Type Name>
2	121933	121933	Open Item
3	20803	Red Baseball Cap	Item
4	20805	20805	Item
5	101567	20808-013	Sales Item
6	109013	20803-04	Sales Item
7	110604	20808-08	Sales Item
8	110606	20808-09	Sales Item
9	121932	SalesItem 121932	Open Sales Item

Page Number – Data Source Outbound

After all InDesign documents for a publication have been finalized and saved back to STEP, data about the mounted products and the pages on which they appear can be exported from the publication in order to generate an **Alphabetical Index - XML** file. This XML file format is used to create both types of indexes—the basic 'stock number' index and the more advanced 'alphabetical index.'

Details about this option can be found in the **Exporting Index Data from STEP** section of the **STEP'n'design** documentation.

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Parent ID - Data Source Outbound

The option to map the parent ID of an object identifies the object's place in the hierarchy.

When a key is used in place of the STEP parent ID by external systems, the key(s) must be mapped instead. For more on keys, see the **Unique Keys** section of the **System Setup / Super User Guide** documentation.

The parent is displayed as part of the Path parameter on an object editor, and indicates the node directly above the selected object.

Tree

Primary Product Hierarchy

Products

Apparel

Upper Body Wear

Head Wear

Hats and Caps

Footwear

Safety

Hardware

Displays

Furniture

Automotive

Building Products

Head Wear rev.0.4 - Product

Product

Sub Products

References

Referenced By

Images & Documents

Commerce

Description

Name	Value
ID	20433
Name	Head Wear
Object Type	Level 2
Revision	0.4 Last edited by USERJ on Mon Aug 28 12:22:01 EDT 2017
Approved	Last Approved on Thu May 21 14:58:40 EDT 2015
Translation	Not Translated
Path	Primary Product Hierarchy/Products/Apparel/Head Wear

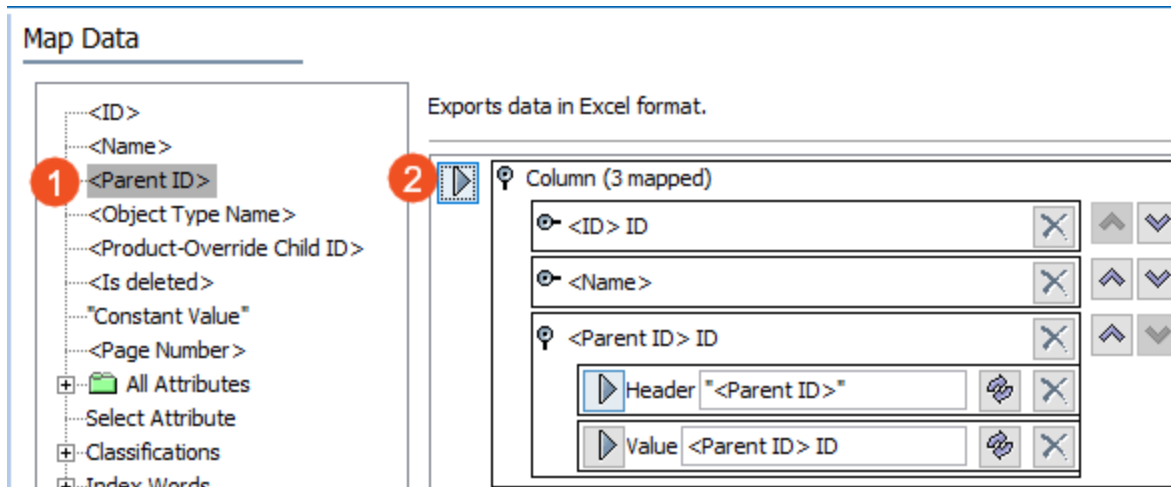
The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Parent ID or Key

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, select the <Parent ID> node and click the right arrow button.



3. Click the Transformation button () to apply any transformations, which can change the output without changing the original data. For example, change the output from Name to ID using the Aspect parameter. See the **Outbound Map Data - Transform** topic.
4. Complete any additional mappings and initiate the export.

Results

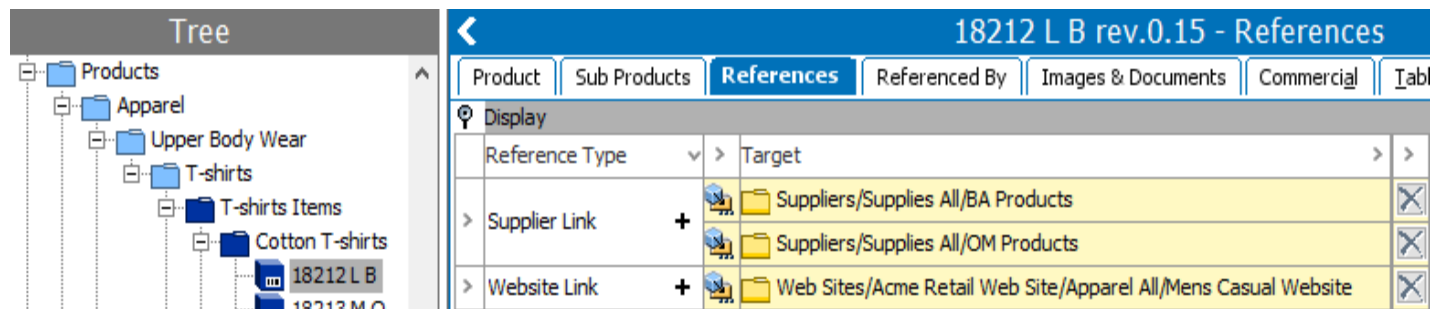
The output includes <Parent ID> as a header (or Object Type ID if transformed) and the value for each row.

	A	B	C
1	<ID>	<Name>	<Parent ID>
2	18210	18210 M B_EN	18209
3	18212	18212 L B	18209
4	18213	18213 M O	18209
5	18216	18216 L O	18209
6	100703	12-GGK799	113201

Product Classification Links – Data Source Outbound

The option to map a Product Classification Link is only available when the object super type selected for the export is Product.

When mapping a Product Classification Link, users can include or exclude inherited references.



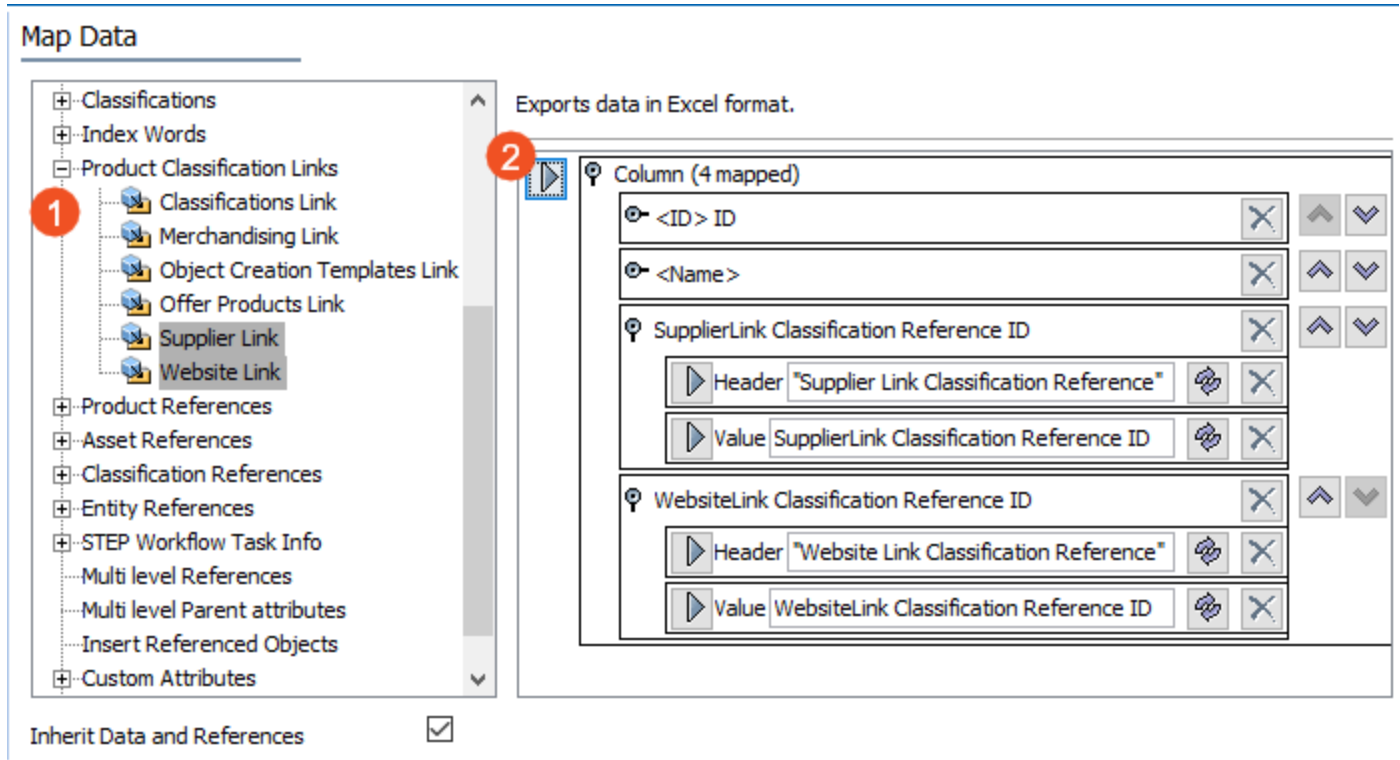
The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Product Classification Links

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, expand Product Classification Links, and select one or more product classification link types.



- Click the right arrow button in the right panel to add the selected product classification link type as a column.
- Check the **Inherit Data and References** option appropriately:
 - If **checked**, inherited data and references are included in the export.
 - If **unchecked**, inherited data and references is not included in the export.

For information on inheritance, see the Inheritance in the **Inherit Data and References - Data Source Outbound** topic.

- Click the Transformation button () to apply any transformations, which can change the output without changing the original data. See the **Outbound Map Data - Transform** topic.
- Complete any additional mappings and initiate the export.

Results

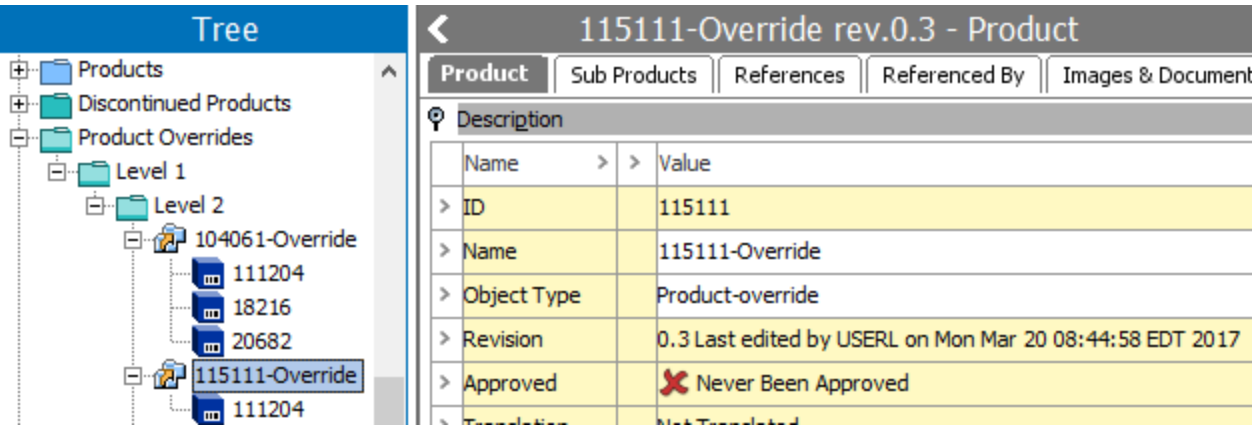
The output includes the product classification link type name as a header and the ID for each row.

When the product references multiple product classification links, the product classification link IDs are displayed in a semi-colon separated list.

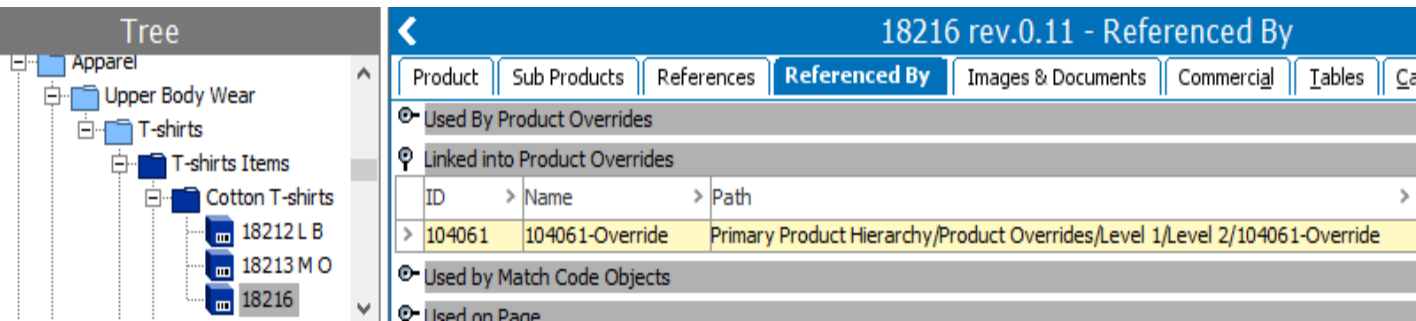
	A	B	C	D
1	<ID>	<Name>	Supplier Link Classification Reference	Website Link Classification Reference
2	18212	18212 L B	BAProducts;OMProducts	22583

Product-Override Child ID – Data Source Outbound

The option to map a Product-Override Child ID is only available when the object super type selected for the export is Product. This allows you to get child products associated to product-overrides, as shown in the following image.



When viewing a product that has been overridden, the Referenced By tab displays the overrides in the Linked into Product Overrides flipper.



When mapping a product-override child ID, users can include or exclude inherited references.

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

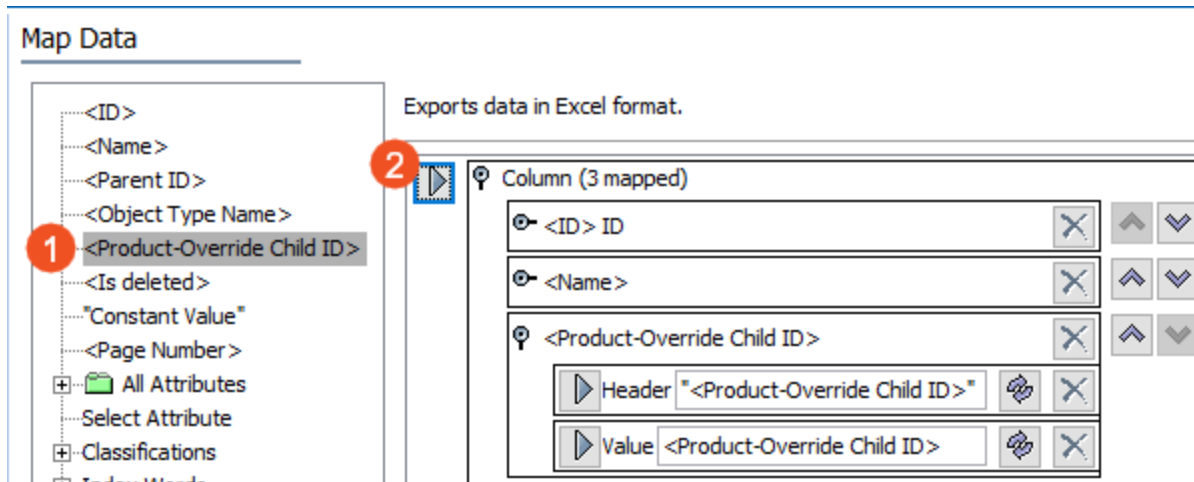
For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Product-Override Child ID

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.

2. Select the product super type and the product override objects for export. For more information, see **Export Manager - Select Objects**, the **Configure the Format** section of the **OIEP - Event-Based - Output Templates Flipper**, or the **Configure the Format** section of the **OIEP - Select Objects - Output Templates Flipper**.
3. On the Map Data step, in the left panel, select the Product-Override Child ID option.



4. Click the right arrow button (▶) in the right panel to add the product-override child ID as a column.
5. Check the **Inherit Data and References** option appropriately:
 - If **checked**, inherited data and references are included in the export.
 - If **unchecked**, inherited data and references is not included in the export.

For information on inheritance, see the Inheritance in the **Inherit Data and References - Data Source Outbound** topic.
6. Click the Transformation button (⚙️) to apply any transformations, which can change the output without changing the original data. For example, change the output from Name to ID using the Aspect parameter. See the **Outbound Map Data - Transform** topic.
7. Complete any additional mappings and initiate the export.

Results

The output includes the product IDs selected for export and the product-override child ID as a header.

When the product-override has multiple children, the child IDs are displayed in a semi-colon separated list.

	A	B	C
1	<ID>	<Name>	<Product-Override Child ID>
2	104061	104061-Override	20682;18216;111204
3	115111	115111-Override	111204

Product References - Data Source Outbound

Product references can be exported when they reference products, classifications, or entities. Product references can be mapped via their ID or a unique key. Each selected product reference is extracted into a separate field in the output.

The Aspect option is available for Product References and allows a variety of data on the product reference to be exported.

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

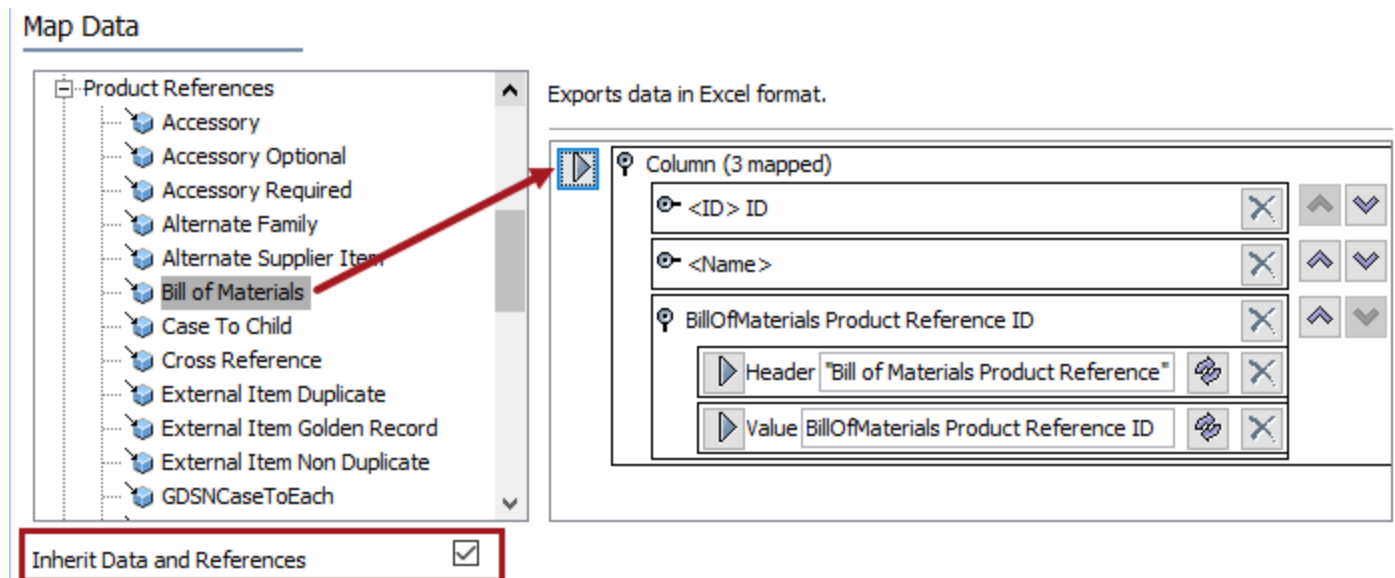
For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, expand **Product References**, and then select the preferred reference type from the list of those available in your system.

Note: For an OIEP, you must first select 'Product' from the dropdown at the top of the Mapping tab.

3. In the right panel, click the right arrow button (▶) to add the selected reference type.

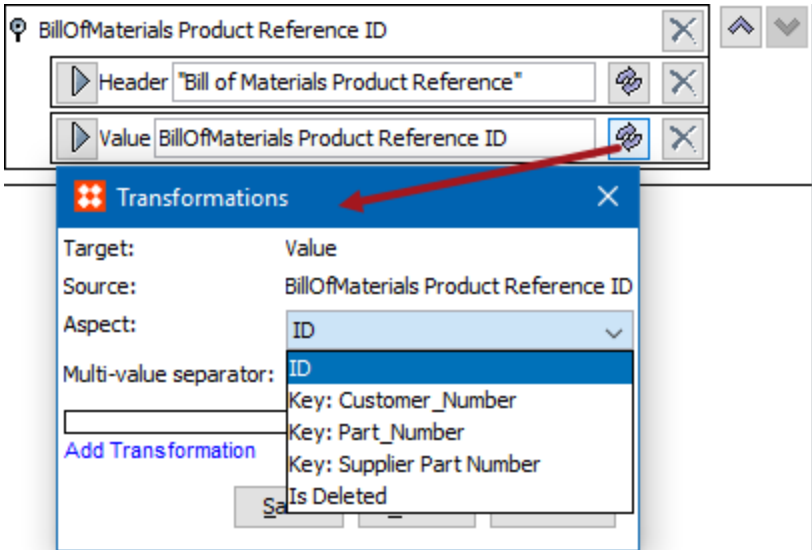


4. Check the **Inherit Data and References** option appropriately:

- If **checked**, inherited data and references are included in the export.
- If **unchecked**, inherited data and references is not included in the export.

For information on inheritance, see the Inheritance in the **Inherit Data and References - Data Source Outbound** topic.

5. If needed, change the reference Value Aspect to output data other than the reference ID. For details on the options, see the **Aspect - Transform Outbound** topic.



- Open the asset reference flipper to display the Header and Value elements.
 - Click the transformation button () for the Header or for the Value element to display the Transformations dialog and the Aspects parameter.
 - Click the **Aspect** dropdown to display and select an option.
 - Click the **Save** button to apply any aspect changes.
6. Click the transformation button () to apply any transformations necessary to change the output without changing the original data. See the **Outbound Map Data - Transform** topic for details.
7. Complete any additional mappings and initiate the export.

Results

The following output is generated using the mapping shown above. The file includes the ID of the object selected for export, the name of that object, and the selected product reference type ID, as well as values for the object.

	A	B	C
1	<ID>	<Name>	Supplier Replacement Item Product Reference
2	18212	Item 1	100703

When product-to-product references exist for the exported object, a column is output for each product reference type ID, as well as the value for the reference type, as shown below. Multiple references for the same reference type are output in a semi-colon-separated list, as shown below.

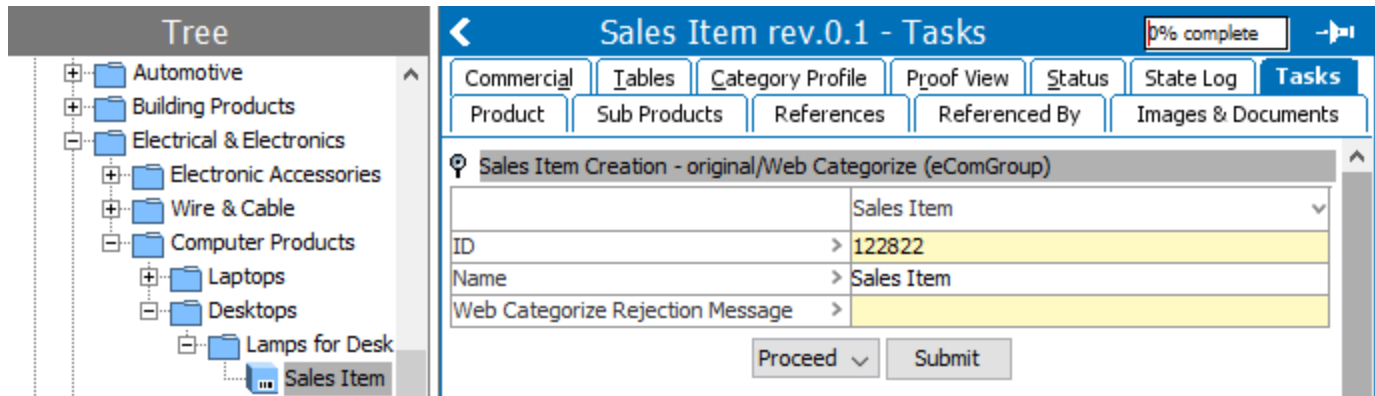
	A	B	C	D
1	<ID>	<Name>	Bill of Materials Product Reference	Supplier Replacement Item Product Reference
2	18212	Item 1		100703
3	18213	18213 M O		124148
4	18216	18216	123857;123854	123855

When the transformation Aspect parameter is used and a Key is selected to identify a reference type, the output includes the key in the reference type column, as shown below.

	A	B	C
1	<ID>	<Name>	Bill of Materials Product Reference KeyID=SupplierPartNumber
2	18212	Item 1	179911

STEP Workflow Task Info - Data Source Outbound

The option to map STEP Workflow Task Info is only available when the object type selected for the export is Product. This option reports on a product's existence and status in the selected workflow(s) at the time of the export. A number of aspects are available to give access to different workflow information. Each aspect must be mapped individually.



Note: This same information can be exported via STEPXML which requires no mapping and includes all aspects and all workflows. STEPXML output can be used to further explain the output generated with a format that requires mapping. For more on STEPXML output, see the **Exporting Workflow Statuses** topic in the **Workflows** documentation.

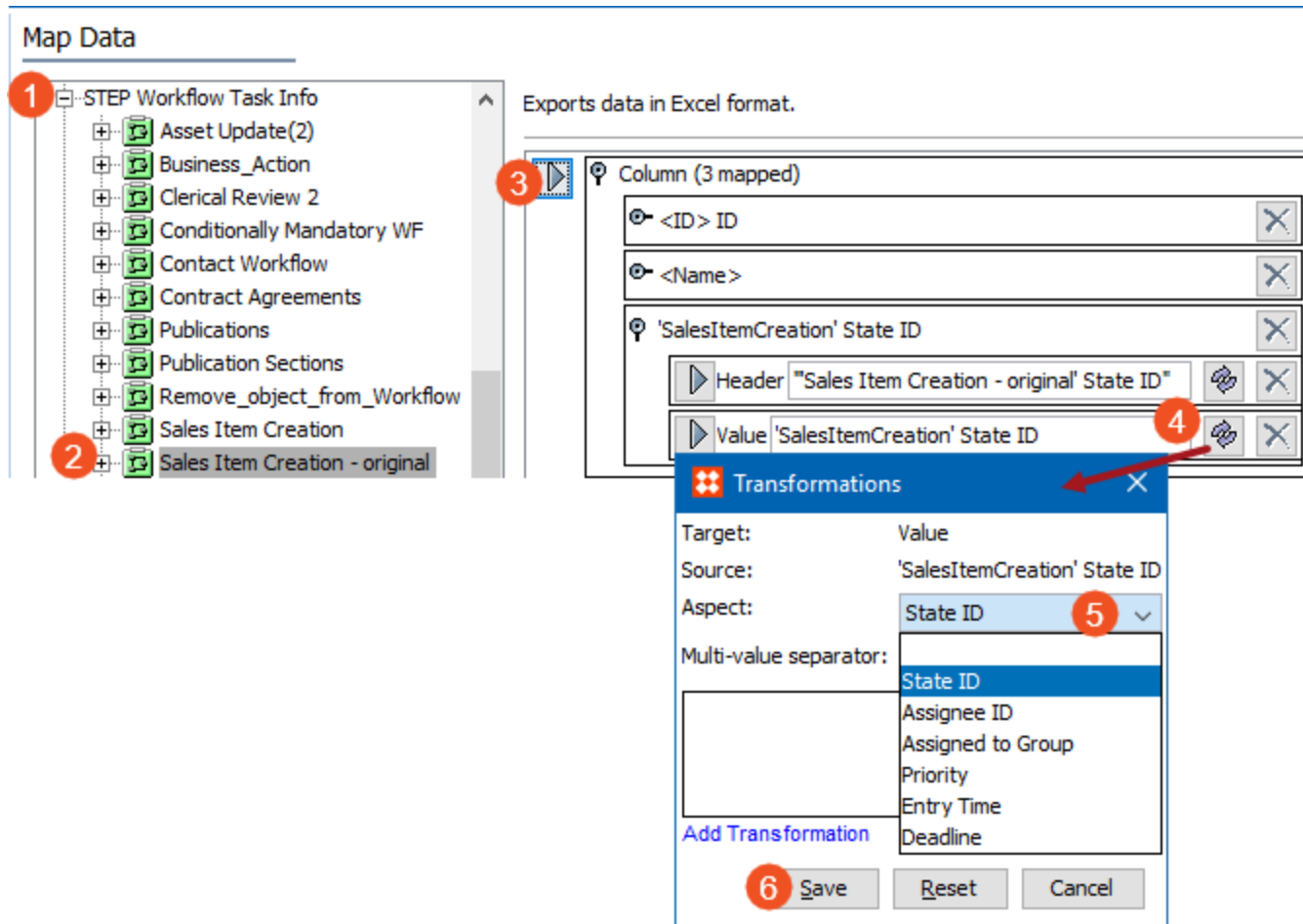
The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping STEP Workflow Task Info

1. Select the data to be exported for the output tool. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, expand the STEP Workflow Task Info node to show all the workflows.
3. Select one or more workflows and click the right arrow button (▶).



4. The default aspect is State ID at the time of the export. To output a different aspect for the time of the export, click the Transformation button (), and choose one of the following aspect options on the Transformations dialog:

- **State ID** - outputs the IDs of the current states of the product.
- **Assignee ID** - outputs current assignees.
- **Assigned to Group** - outputs 'true' or 'false' based on the selection of a group as assignee.
- **Priority** - is a legacy feature that has been replaced with Status Flag functionality.
- **Entry Time** - outputs entry time of all current states, and records only one time for a parallel or cluster.
- **Deadline** - when applicable, outputs deadline specified for the state.

For more information, see the **Outbound Map Data - Transform** topic. Each aspect output is illustrated in the **Results** section below.

5. Complete any additional mappings and initiate the export.

Results

The images below include the product exported and the Excel result based on the selected aspect. Columns A and B show the ID and Name for the product. Column C shows the workflow task information. When the product has multiple values for the aspect, the data is displayed in a semi-colon separated list.

When multiple workflows are mapped, each is displayed in a single column as defined by the header.

• State ID Data and Output

When a product exists in a single workflow state for the workflow as shown below, the workflow state is displayed.

Orange Lamp rev.0.1 - State Log

0% complete

Images & DocumentsCommercialTablesCategory ProfileProof ViewStatusState LogTasks

ProductSub ProductsReferencesReferenced By

Sample Workflow (3 states)

☒ Show transitions☐ Show assignments☐ Show notes☐ Show status flag changes☐ Hide parallel and cluster states

Time	User	Event	From State	To State	Note	Assignee	Status Flag	Log Type
> 2017-09-12 11:15:08	USERJ	Proceed	Start	Do Work				transition
> 2017-07-14 12:29:06	USERM			Start	Auto-initiated when object was created by import process			transition

A	B	C
<ID>	<Name>	'Sample Workflow (3 states)' State ID
208655	Orange Lamp	DoWork

When a product exists in multiple workflow states for the same workflow as shown below, all the relevant workflow states are displayed along with the Cluster and Parallel State IDs that the product exists within.

(115603) rev.0.4 - State Log

Images & Documents Commercial Tables Category Profile Proof View Status **State Log** Tasks

Product Sub Products References Referenced By

Sample Workflow with Parallels

☒ Show transitions ☐ Show assignments ☐ Show notes ☐ Show status flag changes ☐ Hide parallel and cluster states

	Time	User	Event	From State	To State	Note	Assignee	Status Flag	Log Type
>	2017-09-25 13:01:28	USERJ	Proceed	Start	Gather Images				transition
>	2017-09-25 13:01:28	USERJ	Proceed	Start	Gather Data				transition
>	2017-09-25 13:01:28	USERJ	Proceed	Start	Product Assets				transition
>	2017-09-25 13:01:28	USERJ	Proceed	Start	Enrich Data				transition
>	2017-09-25 13:01:28	USERJ	Proceed	Start	Parallel				transition
>	2017-09-25 13:01:07	USERJ			Start				transition

A	B	C
<ID>	<Name>	'Sample Workflow with Parallels' State ID
115603		Enrich_Data;Gather_Data;Gather_Images;Parallel;Product_Assets

- **Assignee ID Data and Output**

When a workflow task has an assignee (set in the workflow State Editor on the Assignee tab), for the current states, the assignees are output. The following images show results for a workflow with a parallel that includes multiple clusters, so the product is assigned to multiple users at the same time.

<

Sales Item rev.0.1 - State Log

0% comp

Images & Documents

Commercial

Tables

Category Profile

Proof View

Status

State Log

Tasks

Product

Sub Products

References

Referenced By

🔍 Sales Item Creation - original

☐ Show transitions

☒ Show assignments

☐ Show notes

☐ Show status flag changes

☐ Hide parallel and du

	Time	>	User	>	Event	>	From State	>	To State	>	Note	>	Assignee	>	Status Flag	>	Log
>	2017-09-28 16:01:06		USERJ				Digital Asset Review						ImageGroup				assign
>	2017-09-28 16:01:06		USERJ				Copy Writing						MarketingCopyGroup				assign
>	2017-09-28 16:01:06		USERJ				Web Categorize						eComGroup				assign
>	2017-09-28 16:01:06		USERJ				Auto 3						USERJ				assign
>	2017-09-28 16:01:06		USERJ				Auto 2						USERJ				assign
>	2017-09-28 16:01:06		USERJ				Auto 4						USERJ				assign
>	2017-09-28 16:01:06		USERJ				Digital Asset Cluster						USERJ				assign
>	2017-09-28 16:01:06		USERJ				Copy Writing Cluster						USERJ				assign
>	2017-09-28 16:01:06		USERJ				Web Categorization Cluster						USERJ				assign
>	2017-09-28 16:01:06		USERJ				Sales Item Enrichment Parallel						USERJ				assign

A	B	C
<ID>	<Name>	'Sales Item Creation - original' Assignee ID
122822	Sales Item	MarketingCopyGroup;USERJ;USERJ;ImageGroup;USERJ;USERJ;eComGroup

• Assigned to Group Data and Output

This allows you to verify if any groups are involved in the workflow. When a workflow task has a group as an assignee (set in the workflow State Editor on the Assignee tab), the output displays 'true'. If only single users are assigned, the output is 'false'.

State Editor

Edit

Comments

Web UI Screen Mapping

Mandatory Attributes

State

On Entry

On Exit

Assignee

Deadline/Escalation

☐ Workflow Default

Assign Task to the default Assignee specified for the Workflow

☒ Select User/Group

Marketing Copy Group (MarketingCopyGroup)

...

Assign Task to selected User or Group

☐ Fetch from Variable

Assign Task to User or Group read from Workflow Variable.

☐ Executing User

Assign Task to the User performing the Transition into the current State

Advanced

Show Edited State

A	B	C
		'Sales Item Creation - original'
<ID>	<Name>	Assigned to Group
122822	Sales Item	false;false;false>true;false;false>true

• Priority Data and Output

The output includes only -1, which indicates the lowest priority. This legacy feature has been replaced with status flag functionality. For more information, see the **Status Flags** topics in the **Workflow** documentation.

A	B	C
		'Sales Item Creation - original' Priority
<ID>	<Name>	
122822	Sales Item	-1;-1;-1;-1;-1;-1;-1

• Entry Time Data and Output

When an object enters a workflow task state, the server time is recorded and can be output using this aspect. In the output, a single time is logged upon entry into a parallel or cluster, so the entry times do not correspond one-to-one with the times displayed on the State Log tab of the object for workflows with parallels and clusters.

Tree

Task Lighting

Flashlights

Desk Lamps

Desk Lamp Items

Black Lamp

Orange Lamp

Green Lamp

Blue Lamp

Pink Lamp

Orange Lamp rev.0.1 - State Log

Commercial

Tables

Category Profile

Proof View

Status

State Log

Tasks

Product

Sub Products

References

Referenced By

In

Sample Workflow (3 states)

☒ Show transitions

☐ Show assignments

☐ Show notes

☐ Show

Time	User	Event	From St
2017-09-12 11:15:08	USERJ	Proceed	Start
2017-07-14 12:29:06	USERM		

A	B	C
<ID>	<Name>	'Sample Workflow (3 states)'
208655	Orange Lamp	Entry Time
		2017-09-12 11:15:08

• **Deadline Data and Output**

When a deadline is set on a workflow using the Deadlines/Escalations tab in the Workflow Editor or the WorkflowDeadline component available in Views and Mappings, the date can be output using this aspect. For information on setting deadlines, see the **Deadlines and Escalations in Workflows** topic and **Views and Mappings for Workflows** in the **Workflows** documentation.

18213 M O rev.0.10 - Tasks

Tables

Category Profile

Proof View

Status

State Log

Tasks

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Workflow with Deadlines/State 2 (Super user)

DeadlineNOTReadOnly

Submit

The deadline: 2017-09-12 07:40:27 reached

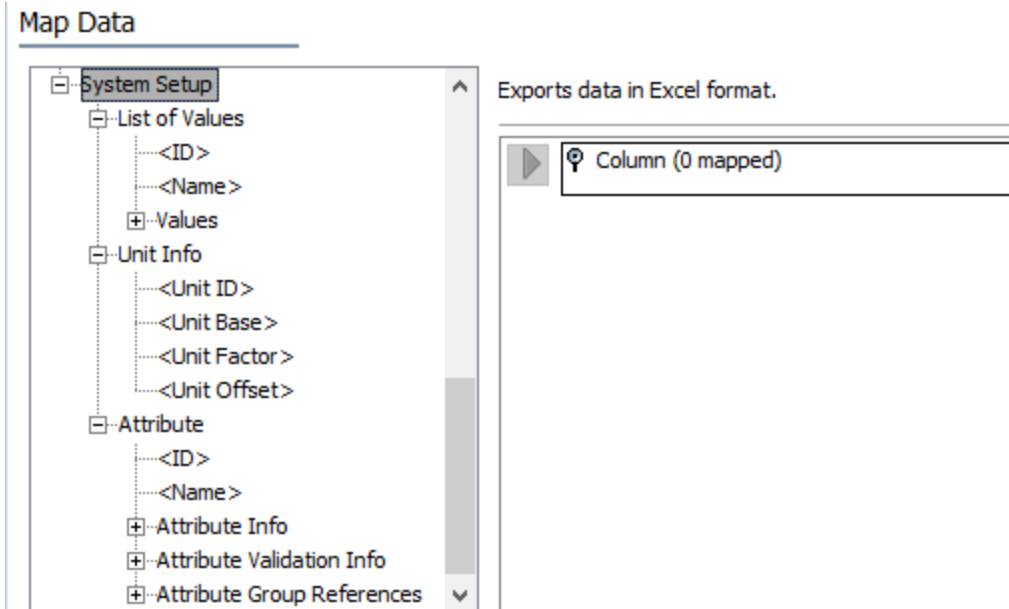
A	B	C
<ID>	<Name>	'Workflow with Deadlines' Deadline
18213	18213 M O	2017-09-12 07:40:27

System Setup - Data Source Outbound

Although displayed for all formats, the System Setup data source gives access to details on List of Values, Units, and Attributes only when the Generic XML format is used.

For examples using this data source, see:

- Attribute Links and LOVs Instruction in Generic XML
- Attribute Links and Units Instruction in Generic XML



The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Outbound Map Data - Mapping Targets

Mapping targets, displayed in the right panel, indicate the order and details of the data being exported. The process to map data differs only slightly based on the data source selected.

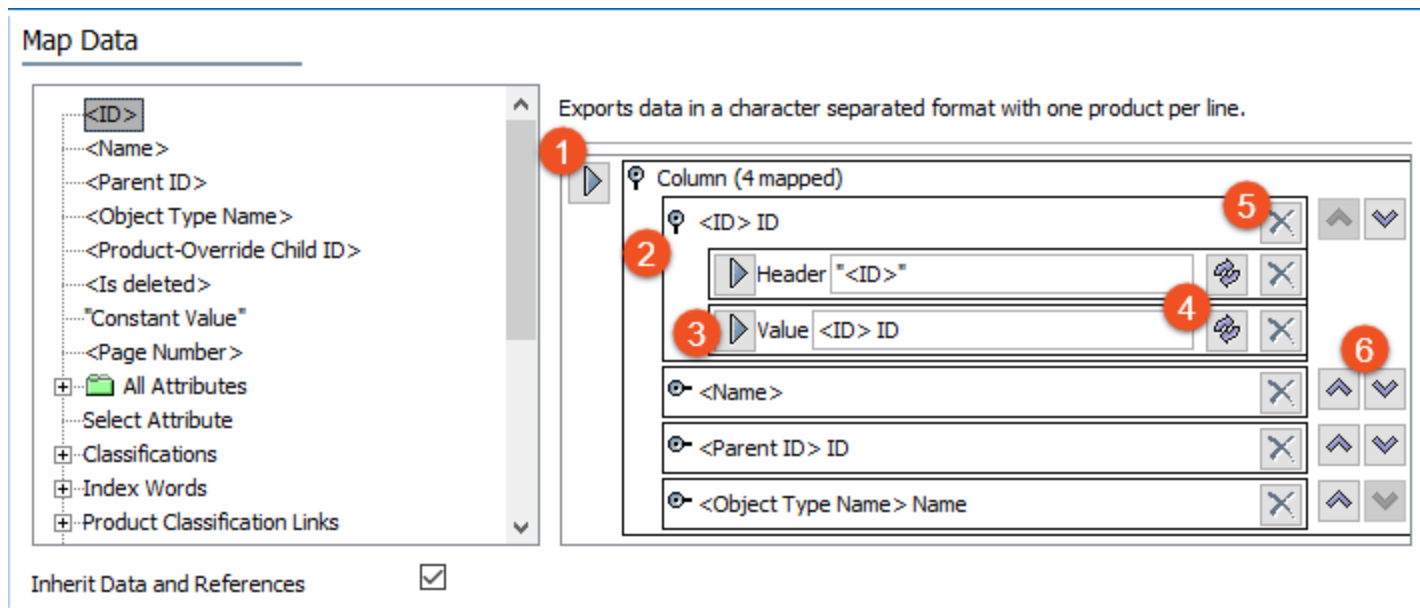
Generally, mapping data involves selecting a data source from the left panel and clicking the mapping button (▶) in the right panel. Based on the selected data source, the mapping rule appears immediately, or additional configuration is required. Ultimately, all data being exported is displayed as a mapping rule.

The user interface for the mapping step of the Export Manager tool and the Outbound Integration Endpoint tool are not exactly the same. For Export Manager, mapping is described in the **Export Manager - Map Data** topic. For OIEPs, mapping is handled in the Output Templates flipper under the Format parameter as described in the **OIEP - Event-Based - Output Templates Flipper** topic or the **OIEP - Select Objects - Output Templates Flipper** topic.

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

Tabular Format Targets

For tabular formats (as identified in the **Outbound Map Data Options** topic), the numbered options in the following image are available for creating mapping targets for outbound data.

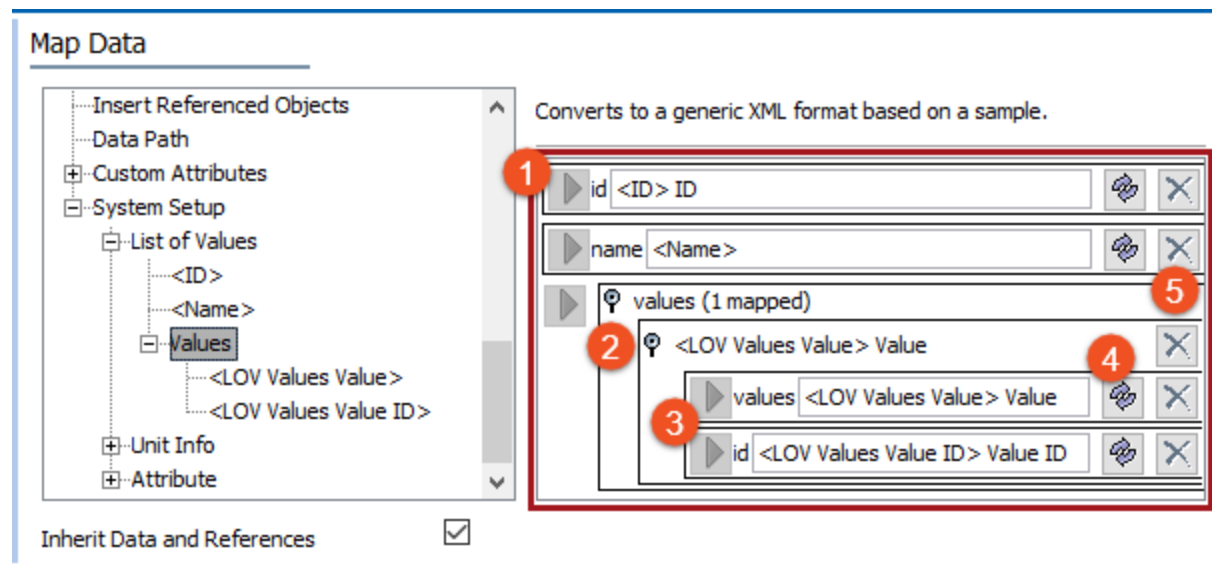


1. Mapping button assigns the selected data source to the selected mapping target.
2. A flipper is available when multiple mapping options are available for a single mapping target.
3. Mapping button assigns the selected data source to the selected mapping target within a flipper.
4. The transform button allows you to modify the STEP data prior to export.

5. The delete button removes a mapping target, eliminating the data from the only export only. This does not affect the actual data in STEP.
6. Up and down buttons allow you to resort the mapping targets.

XML Format Targets

For XML-based formats (as identified in the **Outbound Map Data Options** topic), the numbered options in the following image are available for creating mapping targets for outbound data.



1. Mapping button assigns the selected data source to the selected mapping target.
2. A flipper is available when multiple mapping options are available for a single mapping target.
3. Mapping button assigns the selected data source to the selected mapping target within a flipper.
4. The transform button allows you to modify the STEP data prior to export.
5. The delete button removes a mapping target, eliminating the data from the only export only. This does not affect the actual data in STEP.

Additionally, for XML, the sample file displayed on the Select Format step determines the fields available for export in the Map Data step. To modify the order of the mapping targets, edit the sample file.

Select Format

Generic XML

Converts to a generic XML format based on a sample.

Sample

```
<root>
<attribute>
  <?Record?>
  <ID><?Target?></ID>
  <Info>
    <Type><?Target?></Type>
    <MultiValue><?Target?>
    <FullText><?Target?>
    <Calculated><?Target?>
    <ExternalMaintained><?Target?>
    <DimDepend><?Target?>
  </Info>

```

Map Data

-<ID>
-<Name>
-<Parent ID>
-<Object Type Name>
-<Product-Override Child ID>
-<AttributeLinks>
-<Is deleted>
-"Constant Value"
-<Page Number>
- + All Attributes
- Select Attributes

Converts to a generic XML format based on a sample.

ID	Nothing mapped		X
Type	Nothing mapped		X
MultiValue	Nothing mapped		X
FullText	Nothing mapped		X
Calculated	Nothing mapped		X

Outbound Map Data - Transform

During the Map Data process, transformations modify how data is displayed when it is exported from STEP. For example, with transformations, you can split one STEP data column into multiple values, append and prepend data to values, and perform search and replace.

Transformations are cumulative and are carried out in strict sequence order. That is, if you specify many transformations, each one takes place on the result of the prior transformation, not on the original data. Since there is no preview of the effect of transformation instructions, it is a good idea to implement one or two transformations at a time, save the configuration, and gradually build the complete set of transformations.

Transformations that are textual in nature cannot be applied to attribute values that are not textual.

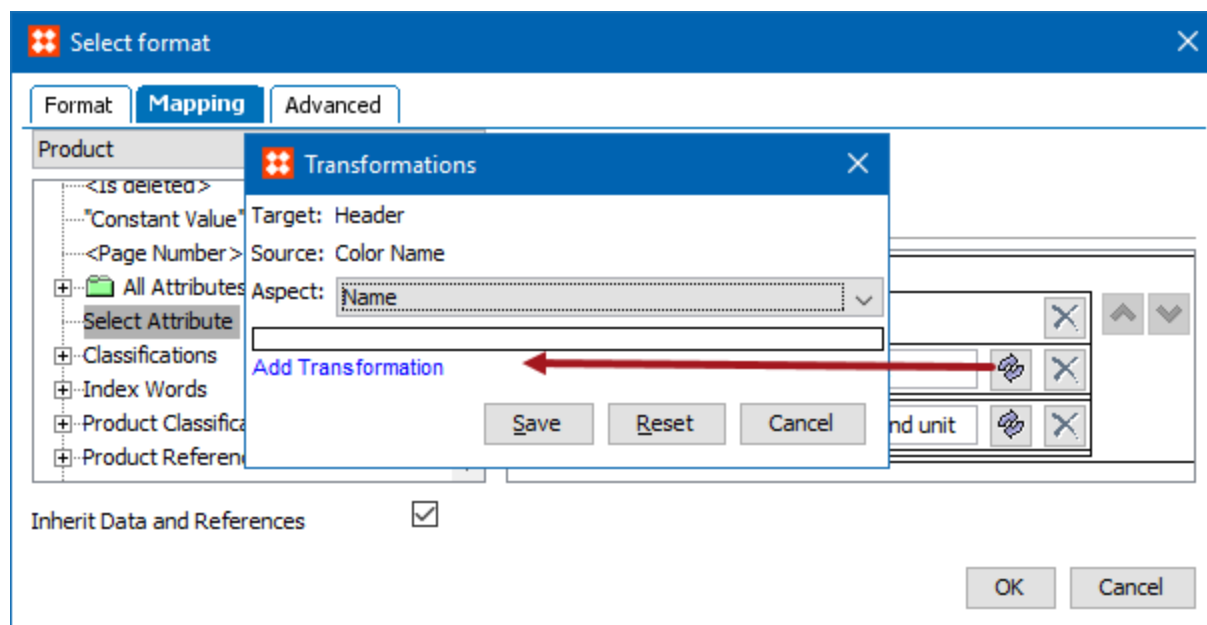
For examples of the most common transformations, see **Mapping Transformation Examples**.

To use transformations on attributes in STEP, or data in tables and publications, see **Attribute Transformations** in the **System Setup / Super User Guide** documentation.

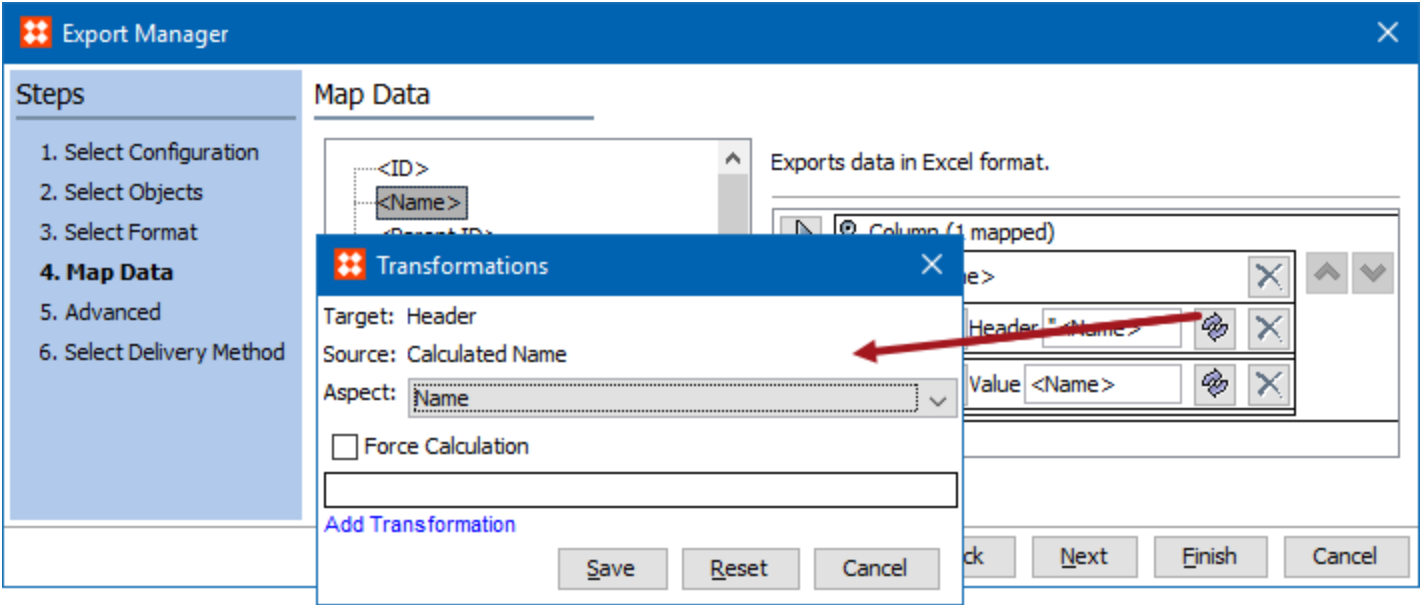
Transforming Data

The mapping step for Export Manager and OIEPs include the same options after clicking the **Transform** button.

In OIEPs, the transformations dialog is accessed from the OIEP editor Configuration tab > > Output Templates flipper > edit the Format column > select the Mapping tab.



In the Export Manager, the transformations dialog is accessed the Map Data step.



Based on the item being transformed, the following parameters are displayed on the Transformations dialog, and allow you to complete the transformation setup:

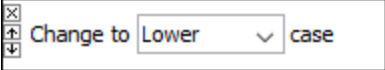
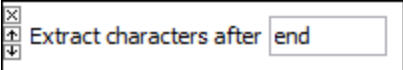
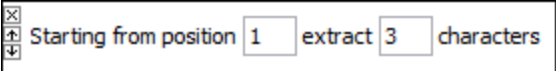
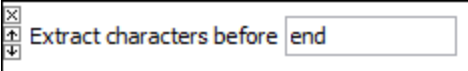
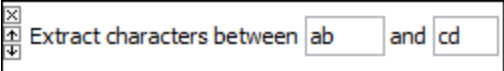
- **Aspect**
- **Data Container Separator**
- **Force Calculation** is only available when a calculated attribute has been selected. Checking this box ensures the value of the calculation will be resolved upon export. When unchecked, calculated values can still be generated via the **Include Calculated Attribute Values** option on the Advanced step.
- **Multi-Value Separator** text box
- **Add Transformation** link (options are shown below)

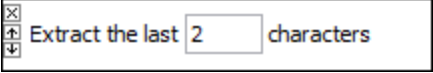
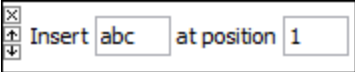
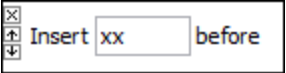
For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

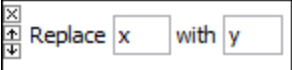
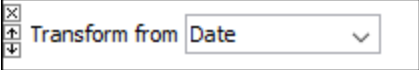
Outbound Add Transformation Options

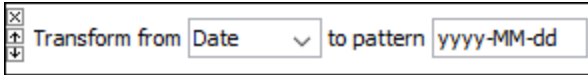
The following transformations are available in a dropdown displayed after clicking the Add Transformation link.

Transformation	Description
Append text	Append yy Adds the specified text as a suffix to the end of the value. For an example using this transformation, see Concatenate Data Using Variables Example.

Transformation	Description
Change case	 Sets values to all upper case, all lower case, or mixed case. Mixed case sets the first letter of the value to upper case and the remainder to lower case. For an example using this transformation, see Replace the Whole Value Example.
Convert fractions to parsable tags	 Scans a text string for fractions. If a fraction is found, then the fraction is converted into parsable tags. A text string that contains the fraction 1/2 cm is transformed to: <StartNumerator/><bfdin/> 1<FractionBar/><StartDenominator/>2<EndFraction/>cm You have to set up an output formatting to Indesign for <StartNumerator/>, <bfdin/>, <FractionBar/>, <StartDenominator/> and <EndFraction/> to mount the fractions.
Extract characters after a given text in the value	 Extracts a part of the value, starting after the occurrence of the specified text string. For an example of this transformation, see Extract Characters Between and Extract Characters After Example.
Extract characters at position	 Extracts a specified number of characters after a specified position in a text string. For an example using this transformation, see Extract Characters at Position Example.
Extract characters before a given text in the value	 Extracts a part of the value, starting from the beginning of the text string and ending at the occurrence of the specified text string. For an example using this transformation, see Replace the Whole Value Example.
Extract characters	

Transformation	Description
between two pieces of text in the value	Extracts the texts between the specified starting text string and the specified ending text string. The start and end text string cannot be the same. For an example of this transformation, see Extract Characters Between and Extract Characters After Example .
Extracts the last N characters of the value	 Extracts a specified number of characters of a value, counting from the end.
Format number	 Formats the number to only have a specified number of decimals, and enables you to localize the number. For an example using this transformation, see Format Number Example.
Format the number with unit	 Formats the number with a unit to only have a specified number of decimals, and enables you to localize the number.
Insert text at a given position	 Inserts a series of characters into the attribute value at the specified position. For an example using this transformation, see Insert Text Example.
Insert text before	 Adds the specified text as a prefix before the value. For an example using this transformation, see Insert Text Example.
Math Operation (+, -, *, /)	 Performs the standard math operations on the value: addition, subtraction, multiplication, and division. Only valid for integer and number attributes. For an example of this transformation, see Math Operation Example.

Transformation	Description
Replace substrings of the value	 Changes the specified text string or empty value with the replacement text string or empty value. All matches on the value in the 'Replace sub-strings' text box are replaced. For an example of this transformation, see Replace Substrings of the Value Example.
Replace substrings of the value using a regular expression	 Uses regular expression to change the specified find value (in the first text box) with the replacement value (in the second text box). From the dropdown, select a replace option: 'if matches', 'first match of', or 'every match'. The if matches option must be set so that the find field defines the <u>entire</u> string to be replaced because it operates like the 'Replace the whole value' transformation. The first match of and every match options operate like the 'Replace substrings of the value' transformation and they replace only the text specified in the find field. For more information on regular expression and practical examples, see the Regular Expression topic in System Setup / Super User Guide documentation.
Replace the whole value	 Replaces the specified text or empty value with a different value. You must enter the entire value, or leave the Replace field blank. Only works on text-based attributes. For an example using this transformation, see Replace the Whole Value Example.
Split and extract	 Split the value where a given string is found and returns one of the created fields. You can, for example extract data from a value after the 4th occurrence of a hyphen, and before the 5th occurrence. So if the value is 179-245-Y-MYGR-TG105P-FDSJ-198, the result is TG105P. For an example using this transformation, see Split and Extract Data Example.
Transform date by locale	 Localized dates should follow the DateFormat.SHORT JavaScript formatting style. Times should be in the DateFormat.MEDIUM formatting style. Transforms Date, ISO Date, or ISO Date and Time attributes to localized formats.

Transformation	Description
	Inbound transforms locale-formatted Date, ISO Date, or ISO Date and Time attributes. The locale format must match the context being used in workbench at the time of import. If not, the 'Transform date by pattern' transformation should be used. Outbound transforms Date, ISO Date, or ISO Date and Time attributes to localized formats. Locale is determined by the context being used in workbench at the time of export. For step-by-step instructions using this transformation, see the Transform Date by Locale or by Pattern topic of the Transformation Examples section of the Data Exchange documentation.
Transform date by pattern	 Transforms Date, ISO Date, or ISO Date and Time attributes to a patterned format (e.g., yyyy.MM.dd G 'at' HH:mm:ss z, EE, MMM d, "yy). You can use the pre-filled pattern provided or manually enter (or cut and paste) a different pattern format. Validation Base Type for the attribute being exported must match the 'Transform from' dropdown choice. If exporting data to a pattern that includes a time zone, the time zone is determined by the STEP server location. Inbound transforms patterned dates (e.g., yyyy.MM.dd G 'at' HH:mm:ss z, EE, MMM d, "yy) to Date, ISO Date, or ISO Date and Time. Validation Base Type for the mapped attribute must match the 'Transform to' dropdown choice. Outbound transforms Date, ISO Date, or ISO Date and Time attributes to a patterned format (e.g., yyyy.MM.dd G 'at' HH:mm:ss z, EE, MMM d, "yy). You can use the prefilled pattern provided or manually enter (or cut and paste) a different pattern format. Validation Base Type for the attribute being exported must match the 'Transform from' dropdown choice. If exporting data to a pattern that includes a time zone, the time zone is determined by the STEP server location. For step-by-step instructions using this transformation, see the Transform Date by Locale or by Pattern topic of the Transformation Examples section of the Data Exchange documentation.

Data Container Separator - Transform Outbound

If a data container attribute is being mapped, the Transformations dialog will show a field for adding in a separator between data container attribute values. Any character entered in this field will be added to the data set that is exported. Inheritance is not applicable for data containers. Transformations are available for any format that requires mapping.

Important: 'Data Container separator' is only visible in the Transformations dialog for Entity object types with data container types that allow for multiple data containers.

Transformations

Target: Value

Source: Shipping Address Value and unit

Aspect: Value and unit

Data Container separator: #

Multi-value separator: ;

Add Transformation

Save Reset Cancel

In this example, the entity 'Allen' has 'Shipping Address 2' as a data container with the 'Country Code' attribute, which includes values. The 'Data Container separator' parameter is used to determine the text that is exported between the values of the data container attribute.

Tree

- Addresses
- Customer Root
- Customers
- Entity Root
 - Account Group 1
 - Allen
 - Customer Hierarchy
 - Key Accounts
- GDSN

Allen rev.0.2 - Data Containers

Customer Data Containers References Reference

Shipping Address

Shipping Address 2

ID	Country Code
Kennesaw	GA
Paris	PR
Toronto	TR

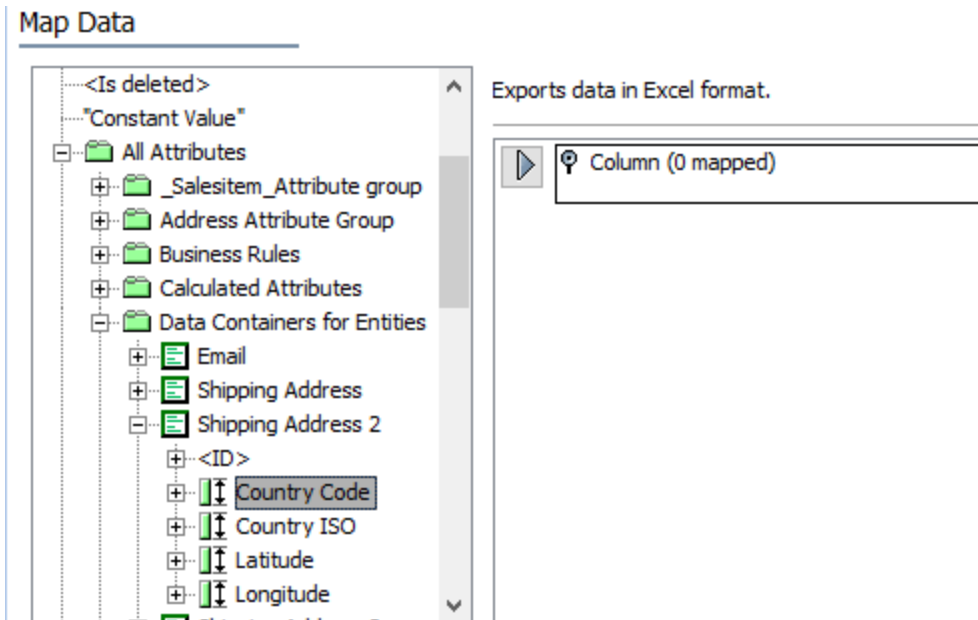
Add Data Container

Configuration

1. In the output tool, choose Entity from the object type dropdown, and select the entity to be exported. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, expand the **All Attributes** data source, then expand the **Data**

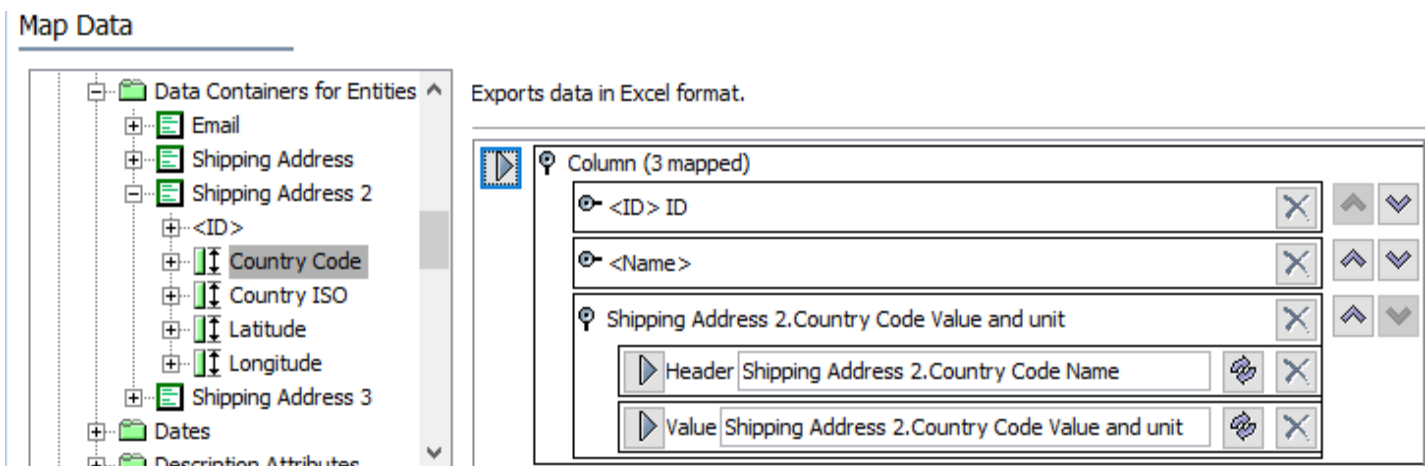
Containers for Entities group.

- Expand the required data container, and then select the required attribute.



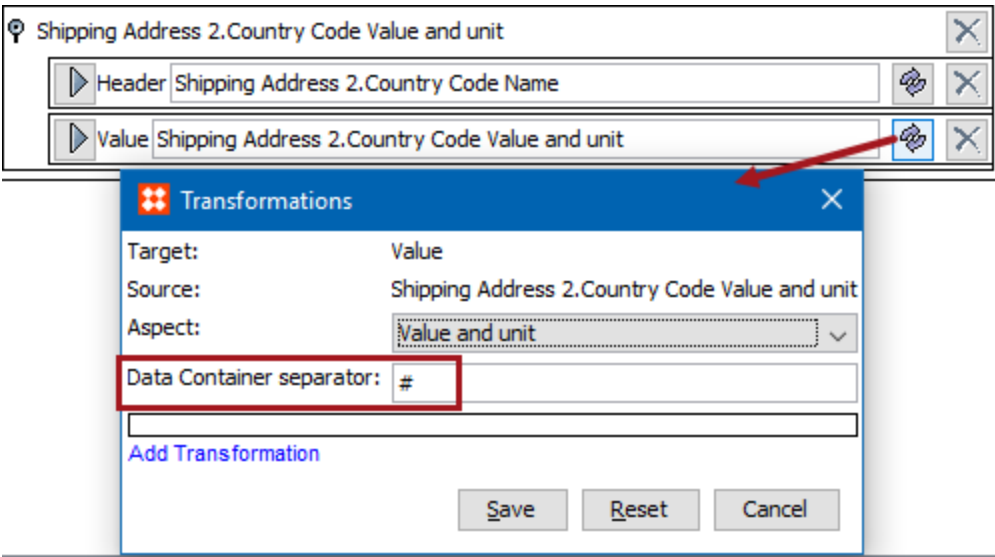
Important: The **All Attributes** data source can also be used to map the Data Container itself and get all of the attribute values associated with the data container. The **Select Attribute** data source does not give access to the data container type, nor the attributes.

- In the right panel, click the mapping arrow (▶) to display the attribute as an item for export and open the flipper.



- On the Value element, click the transformation button (⚙️) to display the Transformations dialog. By default, the '#' is displayed in the **Data Container separator** parameter. If required, specify a different separator.

Important: When exporting a tabular format, select a different character from the one used to delimit columns.



6. If necessary, click the **Reset** button to revert back to the dialog default settings.
7. Click the **Save** button.
8. If necessary, click the Transformation button () again to apply any additional transformations, which can change the output without changing the original data. See the **Outbound Map Data - Transform** topic.
9. Complete any additional mappings and initiate the export.

Results

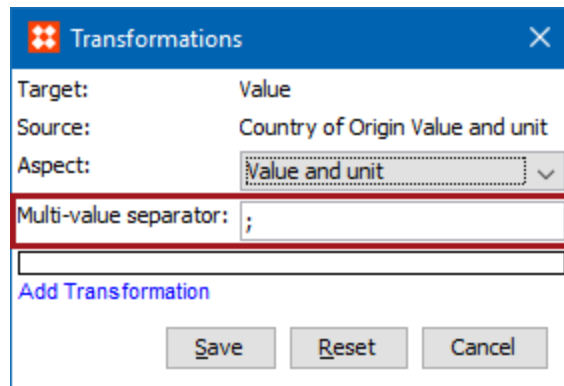
Using the data and configuration in the example above, the output of the entity object includes the entity ID and name, as well as the values in the 'Shipping Address 2' data container, separated by the default separator.

	A	B	C
1	<ID>	<Name>	Shipping Address 2.Country Code
2	CUS_114563	Allen	GA#TR#PR

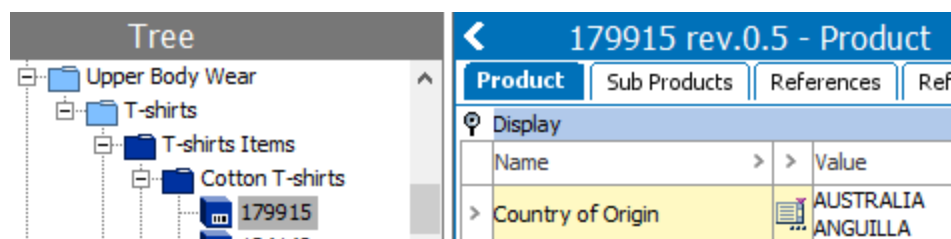
Multi-Value Separator - Transform Outbound

When multivalued attributes are mapped, use a transformation to specify a character to be used as value delimiter. Transformations are available for any format that requires mapping.

Important: 'Multi-value separator' is only visible in the Transformations dialog for attributes with the Multi Valued attribute validation set to Yes.



In this example, the product '179915' has a multivalued attribute 'Country Of Origin', which includes multiple values. The 'Multi-value separator' parameter is used to determine the text that is exported between the values.



Configuration

1. In the output tool select the data to be exported. For more information, see the **Data Exchange** topic.
2. On the Map Data step, in the left panel, click the **Select Attribute** data source.
3. In the right panel, click the mapping arrow (▶) to display the Select Attribute or Attribute Group dialog. Use the **Browse** or **Search** option to find and select the 'Country of Origin' attribute. Click the **Select** button.

Map Data

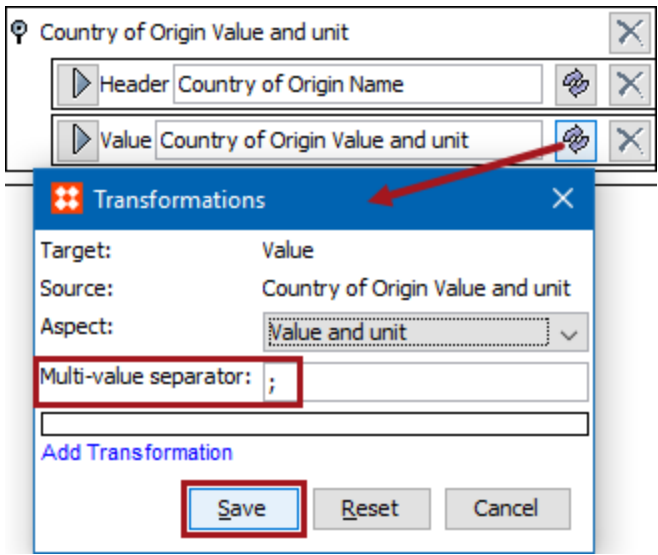
Exports data in Excel format.

The left pane shows a list of attributes under the 'All Attributes' folder, including 'Select Attribute', 'Classifications', 'Index Words', 'Product Classification Links', 'Product References', 'Asset References', 'Classification References', 'Entity References', 'STEP Workflow Task Info', 'Multi level References', 'Multi level Parent attributes', 'Insert Referenced Objects', 'Custom Attributes', and 'System Setup'.

The right pane shows a 'Column (2 mapped)' section with two mapped attributes: '<ID> ID' and '<Name>'. Below this is a 'Select Attribute or Attribute Group' dialog box. The dialog has a 'Browse' tab and a 'Search' tab. The 'Search' tab is active, showing a search result for 'Country of Origin (CountryOfOrigin)' with the value 'Country of Origin ID = CountryOfOrigin'. The 'Select' button is highlighted with a red box.

4. Open the flipper for the mapped attribute.
5. On the Value element, click the transformation button () to display the Transformations dialog. By default, the '#' is displayed in the **Multi-value separator** parameter. If required, specify a different separator.

Important: When exporting a tabular format, select a different character from the one used to delimit columns.



- 6. If necessary, click the **Reset** button to revert back to the dialog default settings.
- 7. Click the **Save** button.
- 8. If necessary, click the Transformation button () again to apply any additional transformations, which can change the output without changing the original data. See the **Outbound Map Data - Transform** topic.
- 9. Complete any additional mappings and initiate the export.

Results

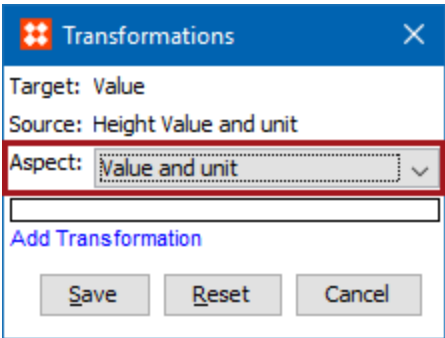
Using the data and configuration in the example above, the output of the product includes the ID and name, as well as the values in the 'Country of Origin' attribute, separated by the selected separator.

	A	B	C
1	<ID>	<Name>	Country of Origin
2	179915	179915	AUSTRALIA;ANGUILLA

Aspect - Transform Outbound

The Aspect determines the data element that is exported, for example, name, ID, value, value and unit, LOV ID, and metadata. (All data elements are listed in the table below). Using aspects allows you to generate multiple columns of different data elements for the same attribute. The options available for an aspect are based on the data source selected. Transformations are available for any format that requires mapping.

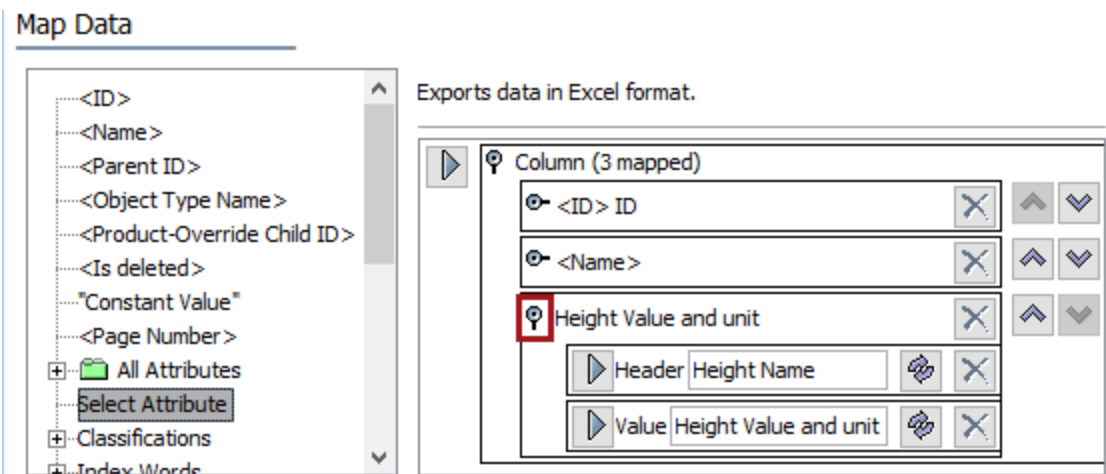
Important: 'Aspect' is only visible in the Transformations dialog for the data sources included in the **Data Sources with Aspect** below.



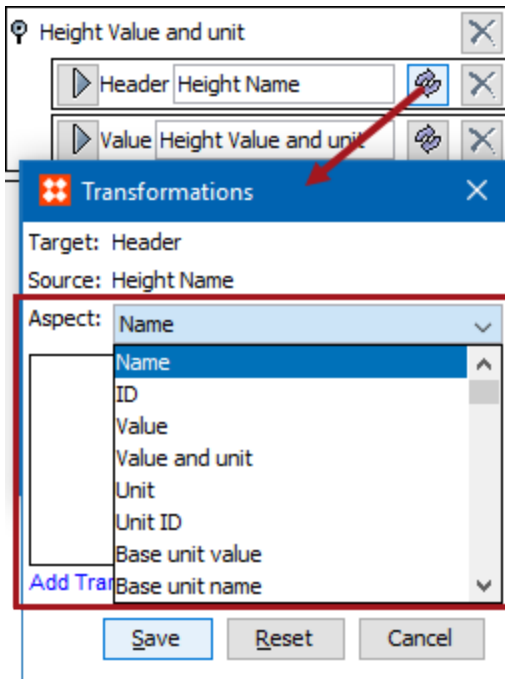
For a general example of mapping aspects, see the **Header and Value Aspects Example**. For specific examples of the available aspects, see the topics referenced in the **Aspect Data Elements Options** table below.

Configuration

- 1. In the output tool select the data to be exported. For more information, see the **Data Exchange** topic.
- 2. On the Map Data step, in the left panel, select the data source. For more information, see the **Outbound Map Data - Data Source** topic.
- 3. In the right panel, click the mapping arrow (▶) and provide any required selections to add the data source as a mapping target. For more information, see the **Outbound Map Data - Mapping Targets** topic.
- 4. Open the flipper for the mapping target.



- On the Header or Value element, click the transformation button () to display the Transformations dialog with the Aspect parameter.



- Select an Aspect from the dropdown and click the **Save** button. For details on the options, see the topics referenced in the **Aspect Data Elements Options** table below.
- If necessary, click the Transformation button () again to apply any additional transformations, which can change the output without changing the original data. See the **Outbound Map Data - Transform** topic.
- Complete any additional mappings and initiate the export.

Aspect Data Element Options

The following options are available in the Aspect dropdown, and will vary based on the selected data source.

Aspect Data Element	Description
Attribute Meta Data [attribute name]	Extracts metadata (Description) attributes that are available on the mapped attribute and are displayed with the prefix 'Attribute Meta Data', as described in Attribute Metadata Aspect .
Attribute Type	Extracts the Validation Base Type of the attribute, as described in Attribute Type Aspect .

Aspect Data Element	Description
Base Unit ID	Extracts the ID of the base unit that is associated with the unit of the product attribute value, as described in Base Unit Aspects .
Base Unit Name	Extracts the name of the base unit that is associated with the unit of the object attribute value, as described in Base Unit Aspects .
Base Unit Value	Extracts the attribute value according to the base unit. This aspect can be applied when established associations exist between units, and where one unit is the base unit, as described in Base Unit Aspects .
Closest Attribute Link Meta Data [attribute name]	For BMEcat 1.2 and BMEcat 2005 formats only, extracts Product Attribute Link metadata attribute values, as described in Closest Attribute Link Meta Data Aspect . The 'closest' value is defined as the first value encountered, either explicit or inherited, while moving up the hierarchy from the product being exported.
Dimension Dependencies	Extracts the names of the dimension dependencies on the selected attribute, as described in Dimension Dependencies Aspects .
[name] Dimension	Extracts the ID of the dimension point selected by name for the selected attribute based on the exported context, as described in Dimension Dependencies Aspects .
File name by asset ID	Extracts multiple files in a .ZIP file including the ID of the digital media files associated with the referenced assets (which can be converted using the conversion pipeline) and the selected export format file.
File name by asset Name	Extracts multiple files in a .ZIP file including the name of the digital media files associated with the referenced assets (which can be converted using the conversion pipeline) and the selected export format file.
ID	Extracts the ID of the object. For example, if an object has the ID '1234' and the name 'Weight', then '1234' is extracted for object.

Aspect Data Element	Description
Is Deleted	Valid only for an event-based OIEP, outputs a Boolean value—true or false—if the target for the reference has been deleted. When mapped on Generic XML format, the value "IsDeleted" is returned. When mapped in Export Manager, the value is always blank.
Key: [Key name]	Extracts the key specified for the object. For more information, see the Unique Keys topic of the System Setup / Super User Guide documentation.
LOV ID	Extracts the ID of the LOV used by the mapped attribute. For example, mapping the ItemColor attribute, which uses the ColorLOV, outputs the ID of ColorLOV.
LOV Value-ID	Extracts the Value ID of the LOV value used by the mapped attribute. For example, mapping the ItemColor attribute, which uses the ColorLOV, outputs the Value ID of value selected from ColorLOV on the attribute.
LOV Value-ID or Value	When mapping Attributes or an Attribute Group, this aspect extracts the LOV Value-ID when present, otherwise it will extract the values in the attributes, as described in LOV Value-ID or Value Aspect .
MIME Type	Extracts the MIME Type of an asset. For example, when an asset with the MIME Type of 'image/png' is referenced to an object. Mapping the Asset Reference data source and using the MIME Type aspect, extracts the MIME type of 'image/png' for the asset reference. For more information, see the MIME Types topic of the System Setup / Super User Guide documentation.
Name	Extracts the name of the object. For example, if an attribute has the ID '1234' and the name 'Weight', then the word 'Weight' is extracted per object.
Unit	Extracts the name of the unit associated with the object attribute value, but not the actual value, as described in Units Aspects .

Aspect Data Element	Description
Unit ID	Extracts the ID of the unit associated with the object attribute value, but not the actual value.
Value	Extracts the attribute value. If the attribute has units assigned to it, and the values have units assigned to them, units are not extracted.
Value And Unit	Extracts the object attribute value and the related unit (if any).
Value and unit Meta Data [name of attribute]	Extracts the value and unit (if any) on the selected metadata attribute that exists on the mapped attribute. For example, if the metadata attribute 'ETIM description' is valid on the ColorsListLOV object, the mapping a product attribute that uses ColorListLOV and using this aspect exports the value and unit on that metadata attribute.

Data Sources with Aspect

The following outbound data source options include the Aspect parameter on the Map Data step:

- Asset References - Data Source Outbound
- Attributes (and Data Containers) - Data Source Outbound
- Classification References - Data Source Outbound
- Classifications - Data Source Outbound
- Constant Value - Data Source Outbound
- Entity References - Data Source Outbound
- ID or Key - Data Source Outbound
- Index Words - Data Source Outbound
- Insert Referenced Objects - Data Source Outbound
- Multi Level Parent Attributes - Data Source Outbound
- Multi Level References - Data Source Outbound
- Object Type Name - Data Source Outbound
- Page Number - Data Source Outbound
- Parent ID - Data Source Outbound
- Product Classification Links - Data Source Outbound

- Product References - Data Source Outbound
- STEP Workflow Task Info - Data Source Outbound
- System Setup - Data Source Outbound

Attribute Metadata Aspect

Metadata (Description) attributes that are available on the mapped attribute are displayed with the prefix 'Attribute Meta Data'.

For general information on setting a Transformation aspect, see the **Aspect - Transform Outbound** topic.

For example, the description attribute 'Completeness Score' is valid on attributes and on products.

In the Tree, the product 179915 has the metadata Completeness Score attribute with a value of 3, which indicates that the product is 30 percent 'complete' when measured against the company's data standards. The Height attribute has a value of 15.

Tree

Primary Product Hierarchy

Product Root

Apparel

Upper Body Wear

T-shirts

T-shirts Items

Cotton T-shirts

179915

124148

18213 M O

18216

179915

< 179915 rev.0.5 - Product

ProductSub ProductsReference

Description

Name	>	>	Value
ID			179915
Name			179915
Completeness Score	123		3

Dimensions

Name	>	>	Value
Height	123		15 cm

In System Setup, the 'Height' attribute has the metadata Completeness Score attribute with a value of 7, which indicates that the attribute is 70 percent 'complete' when measured against the company's configuration standards.

System Setup

Dimensions

Height

Length

Width

Dimension Detail

Holes

Display

eClass 10 Attributes

< Height - Attribute

AttributeReferencesAttribu

Description

Name	>	>	Value
ID			Height
Name			Height
Completeness Score	123		7

Configuration

In addition to the ID and name of the product, the values identified above are mapped using the Select Attribute data source and the specified aspect.

Note: The name of the selected aspect is displayed after the name of the mapped attribute.

- Height attribute with the value aspect of 'Attribute Meta Data Completeness Score'.
- Completeness Score attribute with the value aspect of 'Value'.
- Height attribute with the value aspect of 'Value'.

Map Data

Exports data in Excel format.

- <ID>
- <Name>
- <Parent ID>
- <Object Type Name>
- <Product-Override Child ID>
- <Is deleted>
- "Constant Value"
- <Page Number>
- + All Attributes
- Select Attribute
- + Classifications
- + Index Words
- + Product Classification Links
- + Product References
- + Asset References
- + Classification References
- + Entity References
- + STEP Workflow Task Info
- Multi level References
- Multi level Parent attributes
- Insert Referenced Objects
- + Custom Attributes

Column (5 mapped)

- <ID> ID
- <Name>
- Height Attribute Meta Data Completeness Score
 - Header Height Name
 - Value Height Attribute Meta Data Completeness Score
- Completeness Score Value
 - Header Completeness Score Name
 - Value Completeness Score Value
- Height Value
 - Header Height Name
 - Value Height Value

Results

Product 179915 is exported with the following output:

- ID is 179915.
- Name is 179915.
- Height attribute with the value aspect of 'Attribute Meta Data Completeness Score' is 7.
- Completeness Score attribute with the value aspect of 'Value' is 3.
- Height attribute with the value aspect of 'Value' aspect is 15.

	A	B	C	D	E
1	<ID>	<Name>	Height	Completeness Score	Height
2	179915	179915	7	3	15

Attribute Type Aspect

The Validation Base Type of an attribute can be extracted using the 'Attribute Type' aspect. For more information on validation base types, see the **Validation Rules** topic in the **System Setup / Super User Guide**.

For general information on setting a Transformation aspect, see the **Aspect - Transform Outbound** topic.

For example, the description attribute 'Completeness Score' is valid on attributes and on products.

In the Tree, the product 179915 has the value of 15 for the Height attribute.

Tree

Primary Product Hierarchy

Product Root

Apparel

Upper Body Wear

T-shirts

T-shirts Items

Cotton T-shirts

179915

124148

18213 M O

18216

179915

< 179915 rev.0.5 - Product

Product

Sub Products

Reference

Description

Name	>	>	Value
ID	>	>	179915
Name	>	>	179915
Object Type	>	>	Item

Dimensions

Name	>	>	Value
Height	>	>	123 15 cm

In System Setup, the 'Height' attribute has a Validation Base Type of Number.

System Setup

Dimensions

Height

Length

Width

Dimension Details

Holes

Display

eClass 10 Attributes

eClass Attributes

eClass Meta Attribute

< Height - Attribute

Attribute

References

Attribute Transf

Description

Name	>	>	Value
ID	>	>	Height
Name	>	>	Height

Attribute Validation

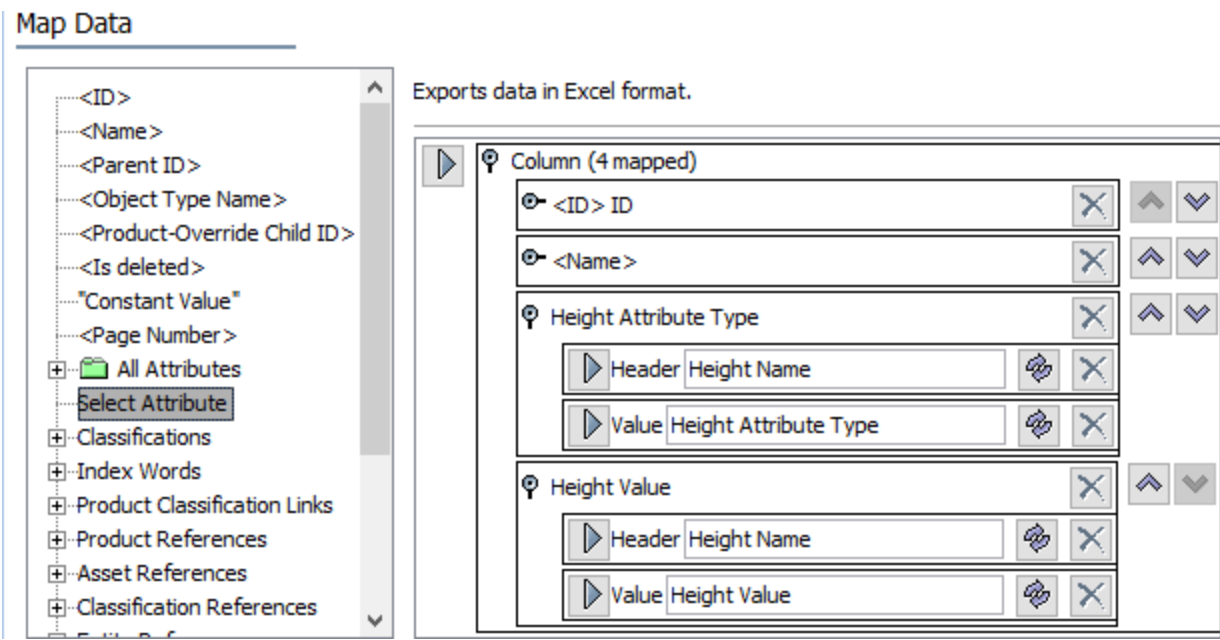
Name	>	Value
Validation Base Type	>	Number

Configuration

In addition to the ID and name of the product, the values identified above are mapped using the Select Attribute data source and the specified aspect.

Note: The name of the selected aspect is displayed after the name of the mapped attribute.

- Height attribute is mapped with the aspect of 'Attribute Type'.
- Height attribute with the value aspect of 'Value'.



Results

Product 179915 is exported with the following output:

- ID is 179915.
- Name is 179915.
- Height attribute with the value aspect of 'Attribute Type' is 'number'.
- Height attribute with the value aspect of 'Value' aspect is 15.

	A	B	C	D
1	<ID>	<Name>	Height	Height
2	179915	179915	number	15

Base Unit Aspects

Data for attributes that include a unit can be extracted using the following base unit aspects:

- **Base Unit ID** - Extracts the ID of the base unit that is associated with the unit of the attribute value.
- **Base Unit Name** - Extracts the name of the base unit that is associated with the unit of the attribute value.
- **Base Unit Value** - Extracts the attribute value according to the base unit. This aspect can be applied when established associations exist between units, and where one unit is the base unit.

For general information on setting a Transformation aspect, see the **Aspect - Transform Outbound** topic.

For example, for the Width attribute on a product, the value '3' is associated with the unit 'ft'.

Tree

Primary Product Hierarchy

Product Root

Apparel

Upper Body Wear

T-shirts

T-shirts Items

Cotton T-shirts

179915

124148

18213 M O

18216

179915

< 179915 rev.0.5 - Product

Product

Sub Products

Reference

Description

Name	>	>	Value
ID	>		179915
Name	>		179915
Object Type	>	Item	

Dimensions

Name	>	>	Value
Width	>	123	3 ft

The Width attribute has a Number validation base type, and allows units of cm (centimeters), mm (millimeters) , in (inches), and ft (feet).

System Setup

- Dimensions
 - Height
 - Length
 - Width
 - Dimension Details
 - Holes
- Display
- eClass 10 Attributes
- eClass Attributes
- eClass Meta Attribute
- ETIM6 Attributes
- ETIM Attributes
- GDSN Attributes
- GDSN System Attribute
- LOVs
- Manual Sequencing
- Manufacturer Information
- Metadata
- MovingAttribute 1

Height - Attribute

Attribute

References

Attribute Transf

Description

Name	Value
ID	Height
Name	Height

Attribute Validation

Name	Value
Validation Base Type	Number

Units

ID	Name	Edited by
unece.unit.CMT	cm	2016-07-26 ..
unece.unit.FOT	ft	2010-03-29 ..
unece.unit.INH	in	2017-01-10 ..
unece.unit.MMT	mm	2017-01-12 ..

[Add Unit](#)

Each of the units is associated with the same base unit of m (meters) and include a conversion factor. For example, feet are multiplied by .3048 to get meters. The base unit of meters includes all required conversion rules. For more information on base units, see the Units topic in the **System Setup / Super User Guide**.

System Setup

- Units
 - Acoustics
 - Angle
 - Area
 - Currency
 - Electricity and Magnetism
 - ETIM Unit Family
 - Frequency
 - Heat
 - Length
 - µm
 - Å
 - cm
 - ft
 - in
 - µm
 - km
 - m
 - mile
 - mm
 - nm
 - yd
 - Light and Related Electromagn

m - Units

Units

References

Log

Description

Name	Value
ID	unece.unit.MTR
Name	m
Last edited	2016-07-26 15:26:26 by USER
Path	Length/m
Base Unit	

Unit Conversions from Base

Rule
Value(m) = 1609.344 * Value(mile)
Value(m) = .0254 * Value(in)
Value(m) = 1000 * Value(km)
Value(m) = .001 * Value(mm)
Value(m) = 0.01 * Value(cm)
Value(m) = .9144 * Value(yd)
Value(m) = .3048 * Value(ft)
Value(m) = .000000001 * Value(nm)

Configuration

In addition to the ID and name of the product, the values identified above are mapped using the Select Attribute data source and the specified aspect.

Note: The name of the selected aspect is displayed after the name of the mapped attribute.

- Width attribute is mapped with the aspect of 'Value and unit'.
- Width attribute is mapped with the aspect of 'Base Unit ID'.
- Width attribute is mapped with the aspect of 'Base Unit Name'.
- Width attribute is mapped with the aspect of 'Base Unit Value'.

Map Data

- <ID>
- <Name>
- <Parent ID>
- <Object Type Name>
- <Product-Override Child ID>
- <Is deleted>
- "Constant Value"
- <Page Number>
- + All Attributes
- Select Attribute
- + Classifications
- + Index Words
- + Product Classification Links
- + Product References
- + Asset References
- + Classification References
- + Entity References
- + STEP Workflow Task Info
- Multi level References
- Multi level Parent attributes
- Insert Referenced Objects
- + Custom Attributes
- + System Setup

Exports data in Excel format.

Column (6 mapped)

☐	<ID> ID	X	↑	↓
☐	<Name>	X	↑	↓
☐	Width Value and unit	X	↑	↓
	Header Width Name	X		
	Value Width Value and unit	X		
☐	Width Base unit ID	X	↑	↓
	Header Width Name	X		
	Value Width Base unit ID	X		
☐	Width Base unit name	X	↑	↓
	Header Width Name	X		
	Value Width Base unit name	X		
☐	Width Base unit value	X	↑	↓
	Header Width Name	X		
	Value Width Base unit value	X		

Results

Product 179915 is exported with the following output:

- ID is 179915.
- Name is 179915.

- Width attribute with the value aspect of 'Value' is '3 ft'.
- Width attribute with the value aspect of 'Base Unit ID' aspect is 'unece.unit.MTR'. Since attribute value uses the 'ft' unit, and it is associated with the meters base unit, the meters ID is used.
- Width attribute with the value aspect of 'Base Unit Name' aspect is blank.
- Width attribute with the value aspect of 'Base Unit Value' aspect is '0.9144'. The attribute value is converted from 3 ft to the corresponding number in m, so the foot conversion of .3048 is multiplied by the attribute value of 3.

	A	B	C	D	E	F
1	<ID>	<Name>	Width	Width	Width	Width
2	179915	179915	3 ft	unece.unit.MTR		0.9144000000000001

Closest Attribute Link Meta Data Aspect

BMEcat 1.2 and BMEcat 2005 are the only formats that allow access to Product Attribute Link metadata attribute values, via this transformation aspect. The 'closest' value is defined as the first value encountered, either explicit or inherited, while moving up the hierarchy from the product being exported.

The options available for this aspect include the names of the Product Attribute Link type reference types.

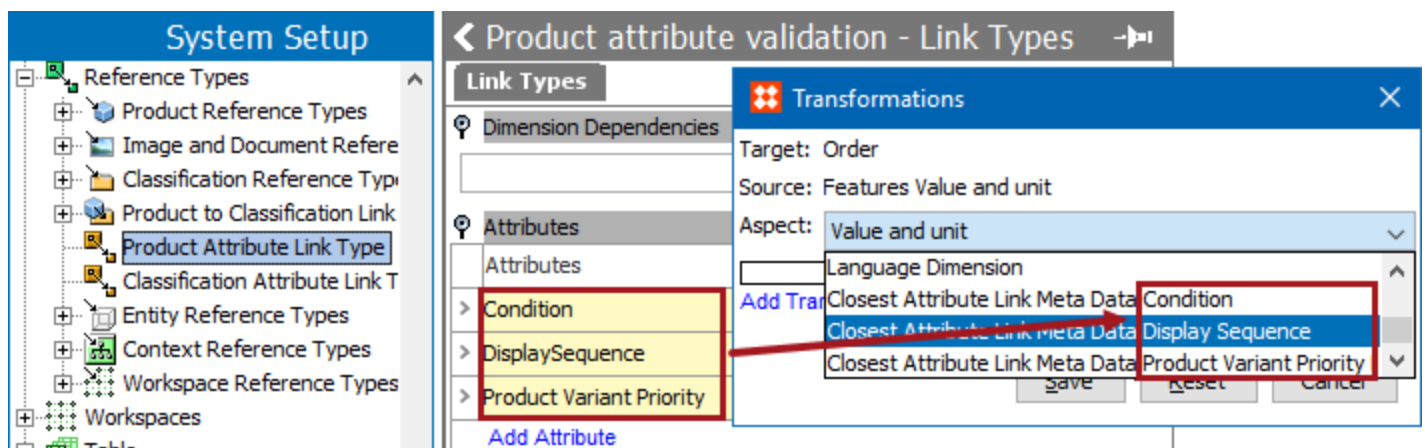
For more information on BMEcat 1.2, see the **BMEcat Format** topic. For more information on BMEcat 2005, see the **BMEcat 2005 Format** topic.

Limitations

- Product Attribute Link metadata attribute values cannot be imported.
- Classification Attribute Link Type metadata and attributes inherited from classifications are not accessible.
- For a specific configuration, there is a limitation on exporting inherited values for the Display Sequence attribute, as described in the **Closest Attribute Link Meta Data and Display Sequence** topic.

For general information on setting a Transformation aspect, see the **Aspect - Transform Outbound** topic.

For example, as shown in System Setup, the following product attribute link type includes the metadata attributes Condition, DisplaySequence, and Product Variant Priority. These same metadata attributes are available via the transformation Aspect parameter.



Note: The name of the selected aspect is displayed after the name of the mapped attribute.

In the tree, product 121191 has an inherited value of '4' for the Product Variant Product metadata attribute displayed on the References tab under the Linked Attributes from Product Hierarchy flipper.

Tree

- Party Supplies
 - Party Hats
 - Paper Hats
 - Balloon Hats
 - Yellow & Pink Party Hat
 - 121191
 - Christmas Party Hat

121191 rev.0.9 - References

Product	References	Referenced By	Images & Documents	Commercial
Linked Attributes from Product Hierarchy				
ID	Name	Product Variant Priority		
ShortItemDescription	Short Item Description	4		
Size	Size			
Status	Status			

Configuration

On the Select Format step of the outbound data tool, select BMEcat 1.2 or BMEcat 2005.

In addition to the mapping targets required by the BMEcat formats, the Select Attribute data source is used for the Short Description mapping target. The 'Short Item Description' attribute is mapped, and the transformation Aspect is set to Closest Attribute Link Meta Data Product Variant Priority.

Map Data

- "Constant Value"
- <Page Number>
- All Attributes
- Select Attribute
- Classifications
- Index Words

Converts to an BMEcat2005 format based on a sample.

Supplier Product ID	<ID> ID	
Short Description	Short Item Description Closest Attribute Link Meta Data Product Variant Priority	

Results

Product 121191 is exported with the following output:

- Supplier_PID is 121191 because it was mapped to the product ID.
- The 'Short Item Description' attribute with the value aspect of 'Closest Attribute Link Meta Data Product Variant Priority' aspect is 4, and is displayed in the preformatted DESCRIPTION_SHORT tag.

```

<PRODUCT>

<SUPPLIER_PID>121191</SUPPLIER_PID>

<PRODUCT_DETAILS>

<DESCRIPTION_SHORT>4</DESCRIPTION_SHORT>

```

Closest Attribute Link Meta Data Aspect and Display Sequence

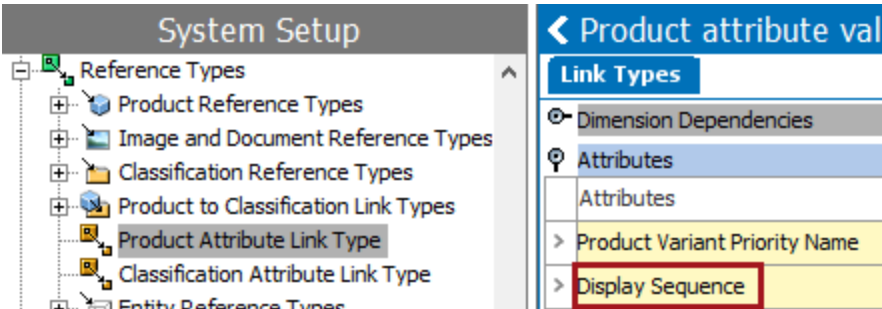
BMEcat 2005 is the only format that allows access to Product Attribute Link metadata attribute values, via the Closest Attribute Link Meta Data transformation aspect.

Inheritance of the Display Sequence attribute value used as metadata varies based on the configuration as follows:

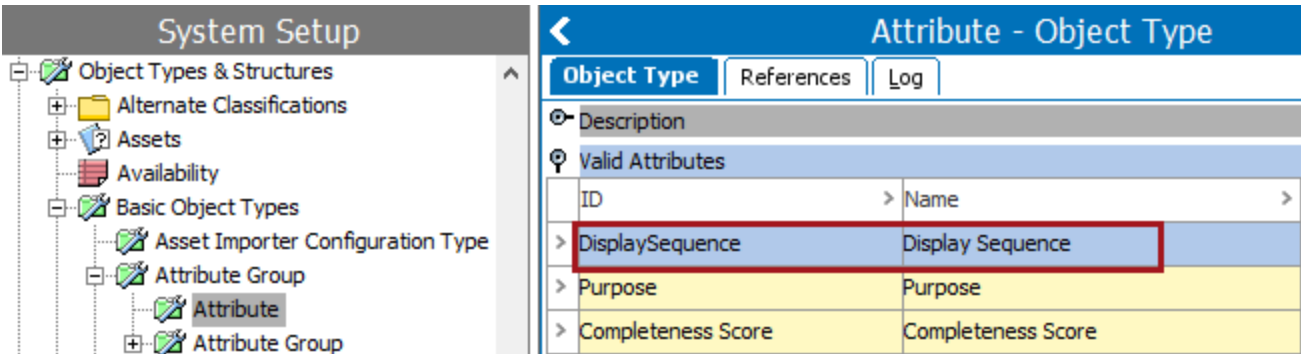
- Using the standard setup of a display sequence attribute, both explicit and inherited values via the Product Attribute Link are exported as expected. Standard setup is defined in the **Attribute Display Sequence** topic of the **Getting Started / User Guide** documentation.
- Configuring the Display Sequence attribute on an attribute node, and setting the value directly on the attribute in System Setup, results in both explicit and inherited values being displayed via the Product Attribute Link, but inherited values are not exported.

For an example of setting the Display Sequence on the attribute, consider the following:

The Product Attribute Link Type includes a metadata attribute with the name Display Sequence. This is the attribute identified in System Setup as **Default Attribute to use as Attribute Display Sequence Attribute**.



The same attribute is valid as metadata on the attribute object type (below the Object Types & Structures node in System Setup).



The value is then set on an attribute that is valid for a product.

System Setup

Attribute Groups

Category Specific Attributes

Color Attributes

Customer Data

Data Containers

Dates

Description Attributes

Dimensions

Display

Discontinued Product Maintenance

Image and Document Maintenance

Item Maintenance

Item Brand Information

Brand Owner

Brand Name

Brand Name - Attribute

AttributeReferencesAttribute TransformationValidityPro

Description

Name	>	>	Value
ID			BrandName
Name			Brand Name
Last edited by			2016-10-28 06:28:57 by DBA
Full Text Indexable			No
Externally Maintained			No
Hierarchical Filtering			None
Calculated			No
Type			Specification
Display Sequence	123	8	

The value from the attribute is displayed on a product on the References tab, under the Linked Attributes from Product Hierarchy flipper.

Notice the two small green down arrows:

- The arrow in the first column (to the left of the Display Sequence column) indicates that the value is inherited from the location higher up the Tree that is identified by the Inherited From column.
- The arrow within the attribute value field indicates that the value is inherited from the attribute.

Tree

Primary Product Hierarchy

Products

Apparel

Upper Body Wear

T-shirts

T-shirts Items

88723-12

88723-12 rev.1

ProductProduct VariantsSub ProductsReferencesReferenced ByImage

Linked Attributes from Product Hierarchy

Display Sequence	ID	Name	Attribute Groups
8	BrandName	Brand Name	Buyer, View, Image R.
14	BrandOwner	Brand Owner	Buyer, View, Item Bra.
	Color	Color	Color Attributes, Sup..

In this example, viewing the Products node (indicated as Inherited From) for the Brand Name attribute continues to show the arrow that indicates the value is inherited from the attribute. Notice that the Brand Owner linked attribute is not inheriting from the attribute, indicated by the missing green arrow.

Tree

Primary Product Hierarchy

Products

Apparel

Upper Body Wear

T-shirts

T-shirts Items

88723-12

Products rev.0.49 - References

Product

Sub Products

References

Referenced By

Linked Attributes from Product Hierarchy

Display Sequence	ID	Name
> 8	BrandName	Brand Name
> 14	BrandOwner	Brand Owner
>	DescriptionWeb	Description, Web
>	ExteriorColor	Exterior Color

With this configuration, the Brand Name and Brand Owner attributes are mapped. In this mapping, a constant value has been added to both Brand Name and Brand Owner to clearly identify it in the output file. The 'Closest Attribute Link Meta Data Display Sequence' transformation aspect is also selected for each mapped attribute.

Map Data

<ID>

<Name>

<Parent ID>

<Object Type Name>

<Product-Override Child ID>

<Is deleted>

"Constant Value"

<Page Number>

All Attributes

Select Attribute

Classifications

Index Words

Product Classification Links

Product References

Asset References

Classification References

Entity References

STEP Workflow Task Info

Multi level References

Multi level Parent attributes

Insert Referenced Objects

Custom Attributes

System Setup

Converts to an BMEcat2005 format based on a sample.

Supplier Product ID

<ID> ID

Short Description

"Brand Name Closest Value = "

Brand Name Closest Attribute Meta Data Display Sequence

Long Description

"Brand Name Closest Value = "

Brand Owner Closest Attribute Meta Data Display Sequence

EAN

Transformations

Target: Long Description

Source: Brand Owner Closest Attribute Meta Data Display Sequence

Aspect: Closest Attribute Link Meta Data Display Sequence

Add Transformation

Save

Reset

Cancel

Buyer ERP Product Group

Nothing mapped

Supplier ERP Product Group

Nothing mapped

Delivery Time

Nothing mapped

Inherit Data and References

☒

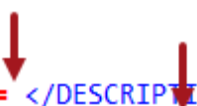
The output shows that the value inherited from the attribute (Brand Name) is not exported, while the value inherited from the Tree (Brand Owner) is exported.

The following portion of the Generic XML output file shows the constant value text and the inherited values:

```

31 <T_NEW_CATALOG>
32 <PRODUCT>
33 <SUPPLIER_PID>100914</SUPPLIER_PID>
34 <PRODUCT_DETAILS>
35 <DESCRIPTION_SHORT>Brand Name Closest Value = </DESCRIPTION_SHORT>
36 <DESCRIPTION_LONG lang="en">Brand Name Closest Value = 14</DESCRIPTION_LONG>
37 <EAN/>

```



Dimension Dependencies Aspects

Data for the dimension dependencies parameter on attributes can be extracted using the following dimension dependency aspects:

- **Dimension Dependencies** - Extracts the names of the dimension dependencies on the selected attribute.
- **[name] Dimension** - Extracts the ID of the dimension point selected by name, for the selected attribute, based on the exported context.

For general information on setting a Transformation aspect, see the **Aspect - Transform Outbound** topic.

For example, the Tree displays that the Short Item Description attribute is valid on a product with ID 121179 and name 'Yellow & Pink Party Hat' and has a value in the contexts being exported.

Tree

- Party Supplies
 - Party Hats
 - Paper Hats
 - Paper Hats
 - Yellow & Pink Party Hat
 - Medieval Party Hat

Yellow & Pink Party Hat rev.0.7 - Product

Product

References

Referenced By

Images & Documents

Commercial

Tables

Item Description Information

Name	>	Value
Short Item Description	abc	Celebrate any occasion with this yellow and pink party hat!

Item Manufacturer Information

In System Setup, the 'Short Item Description' attribute shows that two dimensions have been selected: 'Language' and 'Country name'. A mapping option is displayed for each of the Dimension Dependencies applied to the attribute.

System Setup

- Shipping Pack Depth
- Shipping Pack Diameter
- Shipping Pack Height
- Shipping Pack Weight
- Shipping Pack Width
- Short Item Description
- Stackable?
- Stacked Height

Short Item Description - Attribute

Attribute

References

Attribute Transformation

Validity

Description

Name	>	Value
ID		ShortItemDescription
Name		Short Item Description
Type		Specification
Dimension Dependencies		Language; Country

For the dimension points, for the context selected during the export, the ID of the dimension point is used. In this example, one export is from the 'Danish DK' context (with Language Danish and Country Denmark) and the other is from the 'French Canada' context (with Language French and Country Canada). For more information, see the **Dimensions, Dimension Points, and Contexts** topic in the **System Setup / Super User Guide**.

System Setup

Contexts

Language

All Languages

Danish

English

UK English

French

Country

All Countries

Canada

Denmark

France

USA

Multi Editor

Multi Editor

ID	Danish	fr	CA	DK
Name	Danish	French	Canada	Denmark
Object Type				
Path				

Contexts

ContextsLog

Contexts

ID	Name	Locale	Language	Country
Context3	French Canada	French (Canada) - fr_CA	Danish	Denmark
Context6	Danish DK	Danish (Denmark) - da_DK	UK English	Great Britain

Configuration

In addition to the ID and name of the product, the values identified above are mapped using the Select Attribute data source and the specified aspect.

Note: The name of the selected aspect is displayed after the name of the mapped attribute.

- Short Item Description attribute is mapped with the aspect of 'Dimension Dependencies'.
- Short Item Description attribute is mapped with the aspect of 'Country Dimension'.
- Short Item Description attribute is mapped with the aspect of 'Language Dimension'.

Map Data

<ID>

<Name>

<Parent ID>

<Object Type Name>

<Product-Override Child ID>

<Is deleted>

"Constant Value"

<Page Number>

+ All Attributes

Select Attribute

+ Classifications

+ Index Words

+ Product Classification Links

+ Product References

+ Asset References

+ Classification References

+ Entity References

+ STEP Workflow Task Info

Multi level References

Multi level Parent attributes

Insert Referenced Objects

Exports data in Excel format.

Column (5 mapped)

<ID> ID

<Name>

Short Item Description Dimension Dependencies

Header Short Item Description Name

Value Short Item Description Dimension Dependencies

Short Item Description Country Dimension

Header Short Item Description Name

Value Short Item Description Country Dimension

Short Item Description Language Dimension

Header Short Item Description Name

Value Short Item Description Language Dimension

Results

The product is exported two times, once for each context, with the following output:

- For both contexts: ID is 121179.
- For both contexts: Name is Yellow & Pink Party Hat.
- For both contexts: Short Item Description attribute with the value aspect of 'Dimension Dependencies' is Language and Country, separated by a semicolon.
- Short Item Description attribute with the value aspect of 'Language Dimension' is the ID of the Language object, namely 'Danish' for the Danish DK context and 'fr' for the French Canada context.
- Short Item Description attribute with the value aspect of 'Country Dimension' is the ID of the Country object, namely 'DK' for the Danish DK context and 'CA' for the French Canada context.

	A	B	C	D	E
1	<ID>	<Name>	(ShortItemDescription)	(ShortItemDescription)	(ShortItemDescription)
2	121179	Yellow & Pink Party Hat	Language;Country	Danish	DK

	A	B	C	D	E
1	<ID>	<Name>	Courte Description de l'Article	Courte Description de l'Article	Courte Description de l'Article
2	121179	Yellow & Pink Party Hat	Language;Country	fr	CA

LOV Value-ID or Value Aspect

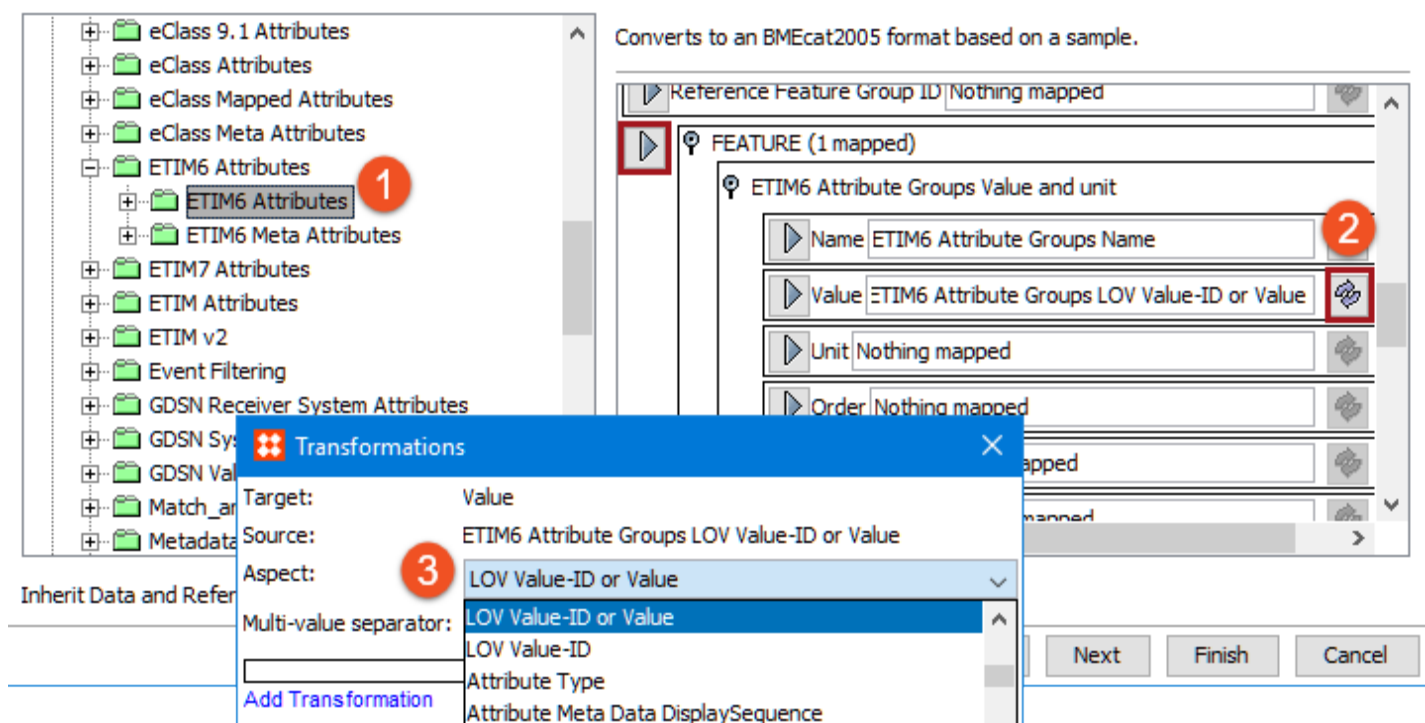
Sometimes it may be necessary to map LOV Value IDs instead of the LOV Values when mapping attribute groups. For instance, when exporting ETIM via BMEcat2005, you would map an attribute group to the feature field and the LOV Value IDs (versus the LOV Values) would be mapped to be exported.

The 'Value and Unit' aspect is the default aspect when mapping attribute groups and exports only the values. Using the 'LOV Value-ID or Value' aspect will allow for values to be exported for all attributes unless it is an LOV attribute with a Value ID.

Configuration

1. In the left panel, select the desired attribute or attribute group, and in the right panel, click the mapping arrow () where the values are to be mapped. In the following image, an attribute group called 'ETIM6 Attributes' is being mapped to the BMEcat2005 FEATURE row, which then creates the 'ETIM6 Attribute Groups' flipper in the right panel with its contents.
2. Click on the transformation button for the row where the LOV values will be mapped.
3. On the Transformations dialog, in the Aspect dropdown, select **LOV Value-ID or Value**. Selecting this means values will be exported for all attributes unless it is an LOV attribute with a Value ID. For attributes with a Value ID, the LOV Value ID will be exported.

Map Data

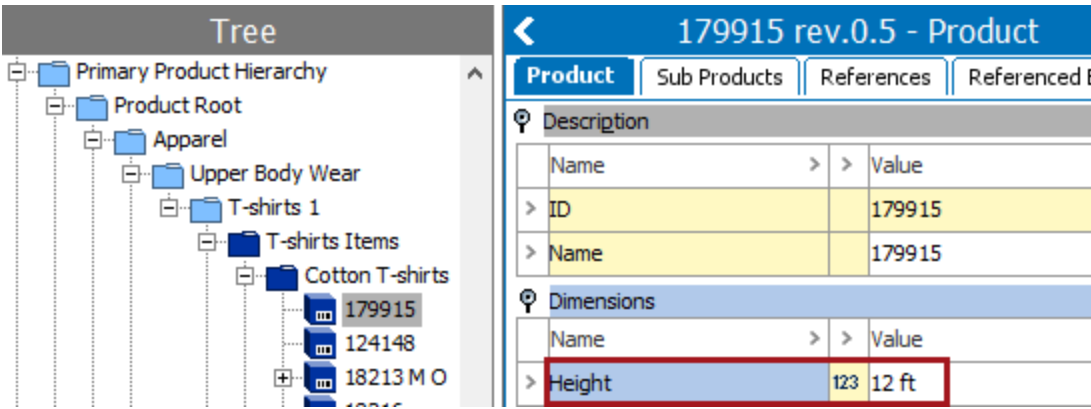


The screenshot shows the 'Map Data' interface with a tree view on the left and a mapping table on the right. The tree view lists various attribute groups, with 'ETIM6 Attributes' selected and highlighted with a red circle and the number 1. The mapping table on the right shows a row for 'FEATURE (1 mapped)' with a mapping arrow icon (1) next to it. Below this, the 'ETIM6 Attribute Groups Value and unit' section is visible, with a red circle and the number 2 next to the 'Value' field, which contains the text 'ETIM6 Attribute Groups LOV Value-ID or Value'. A 'Transformations' dialog box is open in the foreground, showing the 'Aspect' dropdown menu with 'LOV Value-ID or Value' selected, highlighted with a red circle and the number 3. The dialog also shows 'Target: Value', 'Source: ETIM6 Attribute Groups LOV Value-ID or Value', and 'Multi-value separator: LOV Value-ID or Value'. At the bottom of the dialog are buttons for 'Next', 'Finish', and 'Cancel'.

Unit Aspects

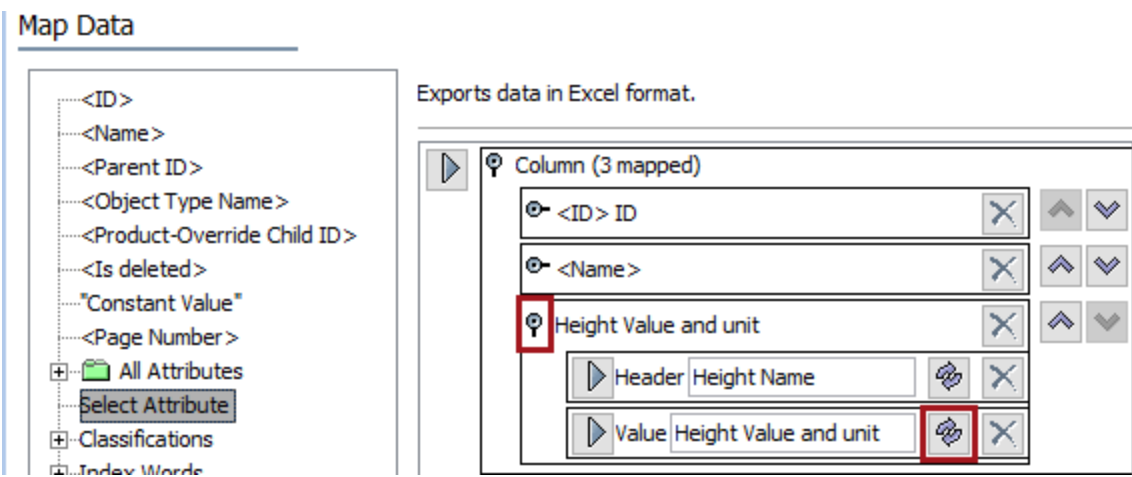
When working with values that may include units, use the Aspect option to control what data is exported, as defined below.

Consider an attribute with the value 12 ft for the Height attribute, as shown in the following image. Exporting that value using the transformation aspect 'Value' outputs the number 12 without the 'ft' units. Using the 'Value and unit' aspect outputs the number 12 and the 'ft' units.

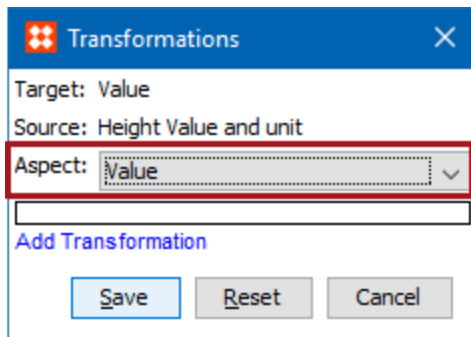


Configuration

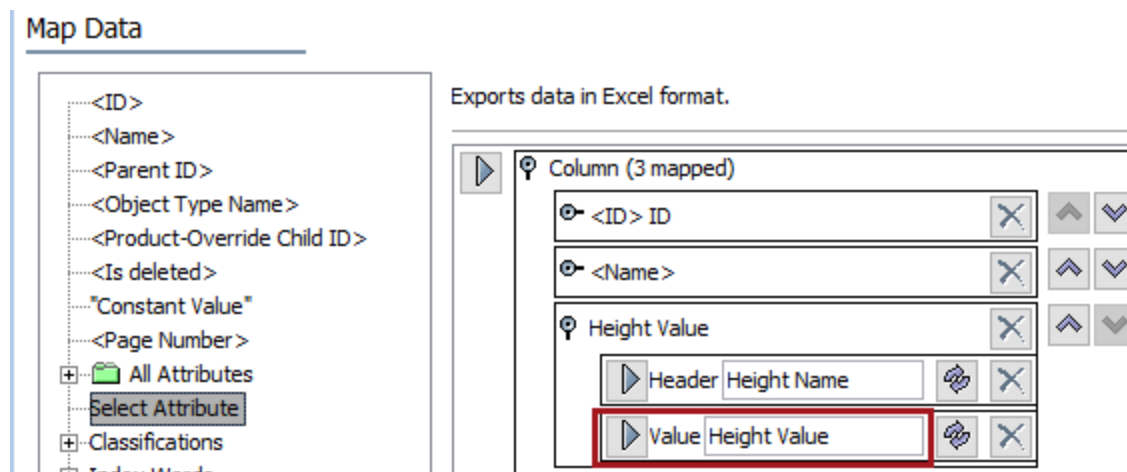
1. In the output tool select the data to be exported. For more information, see the **Data Exchange** topic.
2. On the Map Data step, map the required attribute using the **Select Attribute** data source. For more information, see the **Attributes and Data Containers - Data Source Outbound** topic.
3. In the right panel, click the flipper for the attribute value that should be extracted without unit to display the Header and Value elements.
4. Click the **Transformation** button to display the Transformations dialog.



- In the **Transformations** dialog, from the **Aspect** list, select an aspect and click the **Save** button. In our example, the **Value** aspect will export only the number part of '12', while the **Value and unit** aspect exports both number and unit of '12 ft'. For number attributes with units, 'Value and unit' is the default aspect.



The Value element is displayed with the Height attribute using the 'Value' aspect.



- Complete any additional mappings and initiate the export.

Results

The attribute ID and name are exported, along with only the value portion of the attribute value.

	A	B	C
1	<ID>	<Name>	Height
2	179915	179915	12

To export the value and the unit, repeat the previous steps 1-3. In step 4, set Aspect = 'Value and unit' in the Transformation dialog. Again, the attribute ID and name are exported, along with the value and unit for the attribute.

	A	B	C
1	<ID>	<Name>	Height
2	179915	179915	12

Transformation Examples

Regardless if the data is being imported or exported, or is being used within STEP, generally the transformation options have the same effect.

Transformations are available using the STEP Workbench import and export processes for formats that require **Data Mapping** as discussed in the **Data Exchange** documentation, and also using **Attribute Transformations** for attributes in the **System Setup / Super User Guide** documentation.

For additional help importing data, see the **Creating A Data Import** section of the **Data Exchange** documentation. And, for additional help exporting data, see the **Creating a Data Export** section of the **Data Exchange** documentation.

For examples of some of the most commonly used transformations, see the sections below:

- Concatenate Column with Unit Data Example
- Concatenate Data Using Variables Example
- Concatenate into Single-Valued Data Example
- Dimension Point for Single Dimension Data Example
- Extract Characters Between and Extract Characters After Example
- Extract Characters at Position Example
- Format Number Example
- Header and Value Aspects Example
- If Equals Source Example
- Insert Text Example
- Math Operation Example
- Merge Attributes and Constant Example
- Merge Data for Multi-Valued Data Example
- Replace Substrings of the Value Example
- Split and Extract Data Example
- Replace The Whole Value Example
- Transform Date by Locale or by Pattern
- Transform Using Attribute Transformation Example
- Transformation Lookup Table Example

Additional examples using the Transformation Aspects parameter, see the **Aspect - Transform Outbound** topic.

Concatenate Column with Unit Data Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

In this example, the source file is a CSV or XLS file with product data where one column has a value and the following column has the corresponding units for that value.

One column has numbers such as 12, 15, 22, and another column has units like mm, cm, and inches. The following options are available to properly load this into STEP.

- External update: Concatenate the value and unit columns in the source file, and save the concatenated column values as text. In the import, map the concatenated column to the attribute.
- During import: Concatenate the value and unit columns using the mapping procedure as illustrated below.

Concatenate value and unit during import

1. In the Import Manager 'Map Data' step, map the product ID by selecting the column that contains the product ID, in the Source window as usual.

Map Data

Source:

<ID>	>	<Name>	>	<Parent ID>	>	<Object Type Name>	>	Height	>	unit
114852		Blue Light		20859		Item		2		ft
114854		Green Flashlight		20859		Item		3		in
111204		LED Flashlight		20859		Item		4		cm
114859		Pink light		20859		Item		4		in

Result:
Map to: Product

ID=<ID> ✓
114852
114854
111204
114859

Auto Map Map Constant Remove Transform Generate Profile

2. Select the column that contains the attribute values, in this case 'Height', and map it to the appropriate attribute.

Map Data

Source:

<ID>	> <Name>	> <Parent ID>	> <Object Type ... >	Height	> unit
114852	Blue Light	20859	Item	2	ft
114854	Green Flashlight	20859	Item	3	in
111204	LED Flashlight	20859	Item	4	cm
114859	Pink light	20859	Item	4	in

Result:

Map to: Product

ID=<ID> ✓	Height=Height, unit ✓
114852	2 ft
114854	3 in
111204	4 cm
114859	4 in

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Note that the column has the corresponding units mapped automatically, and that the Result panel shows the value with the corresponding units. It can be observed in the header of the Result panel that **Height = Height, unit**. This means that it is concatenated with the 'unit' column.

In order for it to concatenate correctly, the attribute must have the corresponding units in STEP. If it does not, an error is displayed.

ID	Name	Edited by	Path	Default Unit
> unece.unit.CMT	cm	2016-07-26 15:26:26 by USER	Length/cm	☒
> unece.unit.FOT	ft	2010-03-29 10:41:48 by STEPSYS	Length/ft	☐
> unece.unit.INH	in	2016-09-14 14:26:17 by USER4	Length/in	☐

> Add Unit

Result:

Map to: Product

ID=<ID> ✓	Height=Height, unit ⚠
114852	2 ft
114854	3 in
111204	4 cm
114859	5 mm

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Note: For auto-mapping of the 'units' column to the attribute values, the imported file must include the units in the column following the column that holds the attribute values.

3. If you need to check the concatenation of the selected attribute column, in this case 'Height', select the attribute column in the Result filed, and click the **Transform** button. It will show the applied transformation for the mapped column.

Result: Map to: Product

ID=<ID> ✓	Height=Height, unit ✓
114852	2 ft
114854	3 in
111204	4 cm
114859	4 in

Auto Map Map Constant Remove **Transform** Generate Profile

Specify the sequence of transformations in the table below.

Target: Height

☐ Mandatory

Value - Source: Height

Add Transformation

Unit

☐ Constant: cm

☒ Source: unit

Add Transformation

Preview

Height=Height, unit ✓

2 ft

3 in

4 cm

4 in

Save Reset Cancel

It is worth noting that if the 'Mandatory' option is checked, it will skip the empty values and those objects will not be imported.

Specify the sequence of transformations in the table below.

Target: Height

☒ Mandatory

Preview

Height=Height, unit ✓
2 ft
3 in
4 cm

Map Data

Source:

<ID>	>	<Name>	>	<Parent ID>	>	<Object Type ...>	Height	>	unit
114852		Blue Light		20859		Item	2		ft
114854		Green Flashlight		20859		Item	3		in
111204		LED Flashlight		20859		Item	4		cm
114859		Pink light		20859		Item			

Result:

Map to: Product

ID = <ID> ✓	Height=Height, unit ✓
114852	2 ft
114854	3 in
111204	4 cm

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Concatenate Data Using Variables Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

In some cases, a user might need to transform both pieces of data that are to be concatenated before the concatenation. Variables make that type of transformation possible. For more information about variables, see the **Variable - Map Inbound** topic.

Important: Saving an import configuration saves variables and transformations, as well as the mappings, for use each time the same import is run.

This example includes the following transformations:

- Math operation (+, -, *, /)
- Append text
- Append from source

In this example, an import file has a 'Height (in)' and 'Width (in)' column. Both hold values that will be merged into a single multi-valued attribute named, 'ShippingRestrictions', which will be in centimeters instead of inches. The format should look like: [Width cm] x [Height cm].

1. Map 'Height (in)' to the 'ShippingRestrictions' attribute and apply needed transformations. In this case the 'Math operation (+, -, *, /)' and the 'Append text' transformation were used.

Specify the sequence of transformations in the table below.

Target: ShippingRestrictions

☐ Mandatory

Value - Source: Height (in)

Value = Value	*	2.54
Append	x	

Add Transformation

Preview

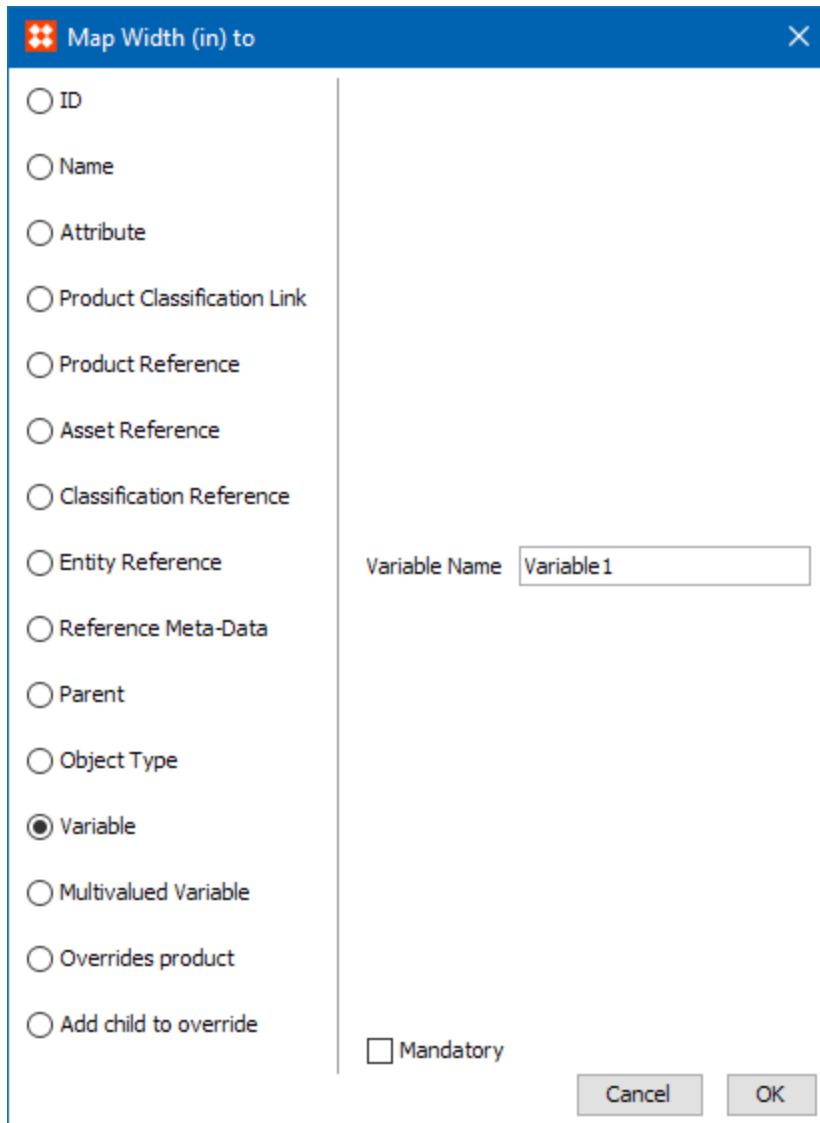
ShippingRestrictions=Height (in) ✓

30.48 x

101.6 x

Save Reset Cancel

2. Manipulate the data in the 'Width (in)' column before it is appended by saving the transformations made in the first mapping, selecting the 'Width (in)' column in the Source area, and clicking the **Map** button. In the 'Map [column name] to' dialog, select the **Variable** radio button and enter a variable name.



The dialog box titled "Map Width (in) to" contains a list of radio button options on the left and a text input field on the right. The options are: ID, Name, Attribute, Product Classification Link, Product Reference, Asset Reference, Classification Reference, Entity Reference, Reference Meta-Data, Parent, Object Type, Variable (selected), Multivalued Variable, Overrides product, and Add child to override. The text input field is labeled "Variable Name" and contains the text "Variable1". At the bottom, there is a checkbox labeled "Mandatory" which is unchecked, and two buttons labeled "Cancel" and "OK".

3. Click **OK** to create a new intermediate level, which is displayed between the Source and Result panels.

Map Data

Source:

<ID>	> Height (in)	> Width (in)	>
20859	12	4	
20859	40	7	

Intermediate Variables:

Variable 1=Width (in)	>
4	
7	

Result:

Map to: Product

ShippingRestrictions=(Height (in)) ✓
30.48 x
101.6 x

Auto Map
Map
Constant
Remove
Transform
Generate Profile

- You can now transform this data before it is appended to the data already mapped to the 'ShippingRestrictions' attribute. Select the variable column, click **Transform**, and then add the transformation that change inches to centimeters. Click **Save**.

Map Data

Source:

<ID>	Height (in)	Width (in)
20859	12	4
20859	40	7

Intermediate Variables:

Variable 1=Width (in)
10.16
17.78

Specify the sequence of transformations in the table below.

Target: Variable 1=Width (in)

☐ Mandatory

Value - Source: Width (in)

Value = Value * 2.54

Add Transformation

Save

Reset

Cancel

Preview

Variable 1=Width (in) ✓
10.16
17.78

Map to: Product

Transform

Generate Profile

- In the Result area, select the 'ShippingRestrictions=(Height (in))' column, click **Transform** and add the 'Append from source' transformation where the intermediate variable is now available. Select the variable and click **Save**.

Specify the sequence of transformations in the table below.

Target: ShippingRestrictions

☐ Mandatory

Value - Source: Height (in)

Value = Value	*	2.54
Append	x	
Append from	Variable1	

Add Transformations

Preview

ShippingRestrictions=(Height (in)) ✓

30.48 x 10.16

101.6 x 17.78

Save Reset Cancel

- The variable now displays the transformed value, which can be mapped in the same manner as source columns are mapped.

Map Data

Source:

<ID>	Height (in)	Width (in)
20859	12	4
20859	40	7

Intermediate Variables:

Variable1=Width (in)

10.16

17.78

Result:

Map to: Product

ShippingRestrictions=(Height (in)) ✓

30.48 x 10.16

101.6 x 17.78

Auto Map Map Constant Remove Transform Generate Profile

Concatenate into Single-Valued Data Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

In the Map Data step of the Import Manager and the IIEP configuration, you can concatenate data from different columns and map the result to a single entity in STEP. You can do this for row and column based import formats such as Excel, CSV, and FixedWidth.

This example includes the following transformations:

- Append text
- Append from source

In this example, you have an import file where you want to concatenate the data in the 'Brand Name' column with the data in the '<Name>' column and map it into a attribute named 'LongItemDescription'.

Map Data

Source:

<ID>	<Name>	<ShortItemDescription[DK]>	Brand Name	<Unique Key>
114852	Blue Light	Lightweight and durable	Flashlights Inc.	AJJKF22
114854	Green Flashlight	Carbon Case	Flashlights Inc.	AJJKF23
111204	LED Flashlight	200 voltz	Flashlights Inc.	668998643
114859	Pink light	20 volt light pink flashlight	Flashlights Inc.	AAJJKF24

Result:

Map to: Product

Auto Map

Map

Constant

Remove

Transform

Generate Profile

1. Select the 'Brand Name' column in the **Source** area and map it to the 'LongItemDescription' attribute.
2. In the **Result** area select the 'LongItemDescription = Brand Name' column and click **Transform**.
3. In the **Transformations** dialog , add the transformation **Append from source** and select the '<Name>' column.

Specify the sequence of transformations in the table below.

Target: LongItemDescription

☐ Mandatory

Value - Source: Brand Name

☐	Append	
☐	Append from	<Name> <ID> <Name> <ShortItemDescription[DK]> Brand Name <Unique Key>

[Add Transformation](#)

Dimension Point

☒ Constant: (From Import Context)

☐ Source: <ID>

[Add Transformation](#)

Preview

LongItemDescription=Brand Name	✓
Flashlights Inc. Blue Light	
Flashlights Inc. Green Flashlight	
Flashlights Inc. LED Flashlight	
Flashlights Inc. Pink light	
Flashlights Inc. Red light	
Flashlights Inc.	

Save Reset Cancel

In this example, a space has been added to the 'Brand Name' value before the **Append from source** transformation to separate the concatenated strings. You can add any number of transformations before the **Append from source** transformation. Also, you can add transformations that are applied to the entire string.



Specify the sequence of transformations in the table below.

Target:LongItemDescription

☐ Mandatory

Value - Source:Brand Name

×

Append

↑

↓

×

Change to

Lower

▼

case

↑

↓

×

Append from

<Name>

▼

↑

↓

Add Transformation

Dimension Point

☒ Constant: (From Import Context) ▼

☐ Source: <ID> ▼

Add Transformation

Preview

LongItemDescription=Brand Name ✓

flashlights inc. Blue Light

flashlights inc. Green Flashlight

flashlights inc. LED Flashlight

flashlights inc. Pink light

flashlights inc. Red light

flashlights inc.

<

>

Save

Reset

Cancel

Note: If you need to add a suffix value from a source or intermediate variable column, a **Prepend from source** transformation is available.

Dimension Point for Single Dimension Data Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

This example includes the following transformations:

- Dimension Point

This example demonstrates import of single dimension dependent data (language in this case) from either a CSV, Excel, or Generic XML file. The title of the column includes the language of the values (DK and UK). The input data is displayed in the Source section of the Map Data step.

1. Map the object's **ID** in the import tool.

Map Data

Source:

<ID>	>	<Name>	>	<ShortItemDescription[DK]>	>	<ShortItemDescription[UK]>	>
114852		Blue Light		Lightweight and durable		Durable, lightweight flashlight	
114854		Green Flashlight		Carbon Case		Durable Carbon case	
111204		LED Flashlight		200 voltz		120 voltz	
114859		Pink light		20 volt light pink flashlight		Light pink flashlight	

Result:
Map to: Product

ID = <ID> ✓
114852
114854
111204
114859

Auto Map
Map
Constant
Remove
Transform
Generate Profile

2. Map the column 'ShortItemDescription[DK]' to the desired attribute. In this case it is mapped to 'ShortItemDescription'. Once mapped, select the attribute column in the Results field and click the **Transform** button.

Map Data

Source:

<ID>	<Name>	<ShortItemDescription[D...>	<ShortItemDescription[U...>
114852	Blue Light	Lightweight and durable	Durable, lightweight flashlight
114854	Green Flashlight	Carbon Case	Durable Carbon case
111204	LED Flashlight	200 voltz	120 voltz
114859	Pink light	20 volt light pink flashlight	Light pink flashlight

Result:

Map to: Product

ID=<ID> ✓	ShortItemDescription=<ShortItemDescription[DK]> ✓
114852	Lightweight and durable
114854	Carbon Case
111204	200 voltz
114859	20 volt light pink flashlight

Auto Map

Map

Constant

Remove

Transform

Generate Profile

- In the Transformation window, under the Dimension Point section, select the Constant radio button, and select the desired language from the dropdown menu. In this case **Danish** is chosen. Click **Save**.

Specify the sequence of transformations in the table below.

Target: ShortItemDescription

☐ Mandatory

Value - Source: <ShortItemDescription[DK]>

[Add Transformation](#)

Dimension Point

☒ Constant: Danish

☐ Source: <ID>

[Add Transformation](#)

Preview

ShortItemDescription=<ShortItemDescription[DK]> ✓

Lightweight and durable

Carbon Case

200 voltz

20 volt light pink flashlight

Save Reset Cancel

The Dimension Point section is shown because the attribute has a single dimension dependency. The default behavior is to use the language of the import context. The dimension point can either be mapped to:

- **Constant** - when all values of the import are in the same language.
- **Source** - when the language is specified in a separate column. In that case transformations may be needed in order to map the values of the language column correctly to the dimension points.

Note: If the Mandatory option is checked on the Transformation screen, it will skip the objects with empty values, and those objects will not be imported / updated.

4. Select the column 'ShortItemDescription[UK]' and map to the same attribute that was chosen before, 'ShortItemDescription'. Once mapped, select the attribute column in the Results field and click **Transform** > **Constant** > **UK English**. Click **Save** on the Transformations dialog.

Map Data

Source:

<ID>	<Name>	<ShortItemDescription[D... >	<ShortItemDescription[U... >
114852	Blue Light	Lightweight and durable	Durable, lightweight flashlight
114854	Green Flashlight	Carbon Case	Durable Carbon case
111204	LED Flashlight	200 voltz	120 voltz
114859	Pink light	20 volt light pink flashlight	Light pink flashlight

Result:

Map to: Product

ID=<ID> ✓	ShortItemDescription=<ShortItem... ✓	ShortItemDescription=<ShortItem... ✓
114852	Lightweight and durable	Durable, lightweight flashlight
114854	Carbon Case	Durable Carbon case
111204	200 voltz	120 voltz
114859	20 volt light pink flashlight	Light pink flashlight

Auto Map Map Constant Remove **Transform** Generate Profile

Specify the sequence of transformations in the table below.

Target: ShortItemDescription

☐ Mandatory

Value - Source: <ShortItemDescription[UK]>

Add Transformation

Dimension Point

☒ Constant: UK English

☐ Source: <ID>

Add Transformation

Preview

ShortItemDescription=<ShortItemDescription[UK]> ✓

Durable, lightweight flashlight

Durable Carbon case

120 voltz

Light pink flashlight

Save Reset Cancel

5. Complete the remaining steps of the import as usual.

LOV Value-ID or Value Aspect

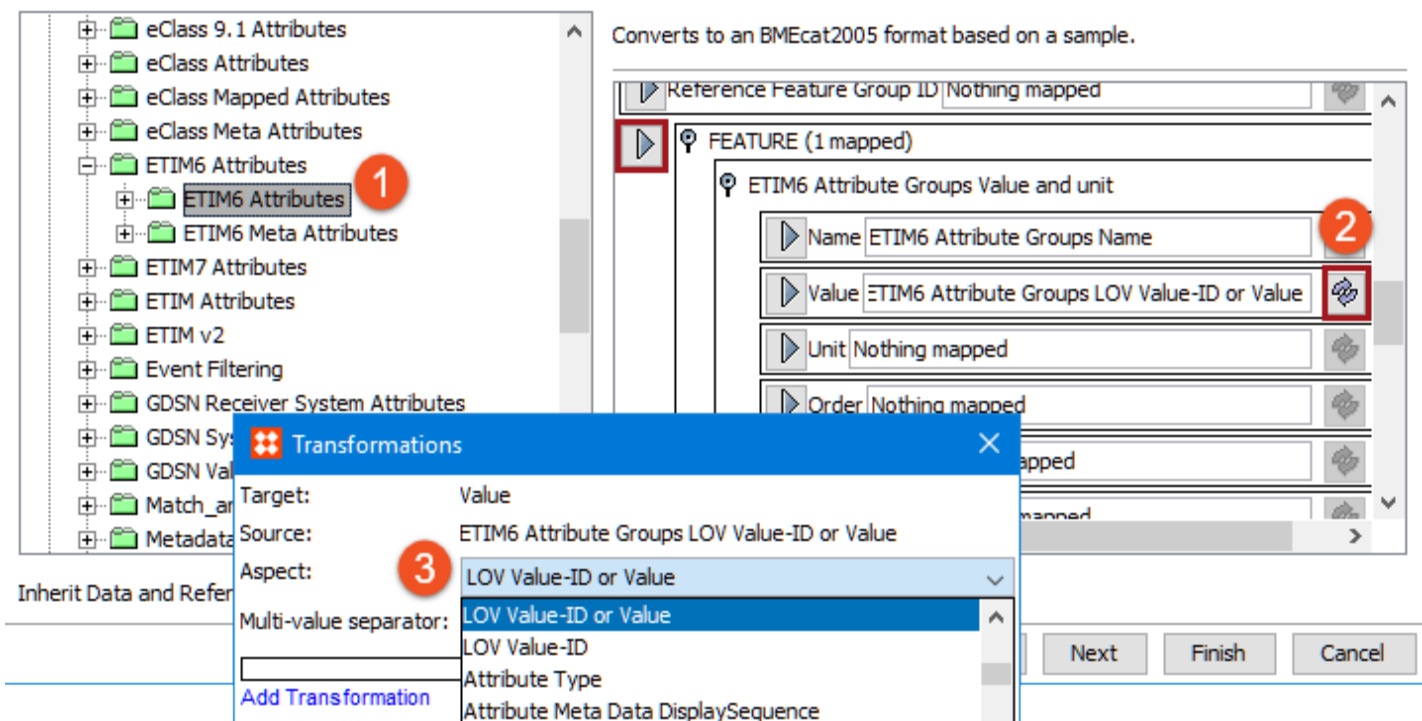
Sometimes it may be necessary to map LOV Value IDs instead of the LOV Values when mapping attribute groups. For instance, when exporting ETIM via BMEcat2005, you would map an attribute group to the feature field and the LOV Value IDs (versus the LOV Values) would be mapped to be exported.

The 'Value and Unit' aspect is the default aspect when mapping attribute groups and exports only the values. Using the 'LOV Value-ID or Value' aspect will allow for values to be exported for all attributes unless it is an LOV attribute with a Value ID.

Configuration

1. In the left panel, select the desired attribute or attribute group, and in the right panel, click the mapping arrow () where the values are to be mapped. In the following image, an attribute group called 'ETIM6 Attributes' is being mapped to the BMEcat2005 FEATURE row, which then creates the 'ETIM6 Attribute Groups' flipper in the right panel with its contents.
2. Click on the transformation button for the row where the LOV values will be mapped.
3. On the Transformations dialog, in the Aspect dropdown, select **LOV Value-ID or Value**. Selecting this means values will be exported for all attributes unless it is an LOV attribute with a Value ID. For attributes with a Value ID, the LOV Value ID will be exported.

Map Data



Converts to an BMEcat2005 format based on a sample.

Reference Feature Group ID: Nothing mapped

FEATURE (1 mapped)

ETIM6 Attribute Groups Value and unit

Name: ETIM6 Attribute Groups Name

Value: ETIM6 Attribute Groups LOV Value-ID or Value

Unit: Nothing mapped

Order: Nothing mapped

Transformations

Target: Value

Source: ETIM6 Attribute Groups LOV Value-ID or Value

Aspect: LOV Value-ID or Value

Multi-value separator: LOV Value-ID or Value

Add Transformation

Next Finish Cancel

Extract Characters at Position Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

This example describes how to extract and clean information from a legacy system, or groups of information stored in one column, and load it into separate attributes in STEP. If information is more technical or cryptic, see the last section of this topic on how to convert cryptic data.

This example includes the following transformations:

- Starting from position
- Replace whole value
- Change case
- Extract characters at position

Data being transformed

Below we use transformations for each of the columns that require data to be clarified.

Consider data in an input file that is cryptic, meaning that it is represented in a different than the way STEP organizes the data. For example, a user needs to extract the color from the material. In this case bl = blue, gr = green and br = brown. This data needs to be transformed into 'real' attribute values that STEP can recognize.

	A	B
1	<ID>	(Data)
2	107601	blcotton 2
3	20803	grwool 4
4	20805	brwool 3
5	109011	grcotton 2

For more on all available transformations, see the **Outbound Map Data - Transform** topic.

Applying transformations

First map the inbound data columns and then apply the needed transformations.

1. Map the '<ID>' column to the ID in STEP. For details, see the **ID or Key - Map Inbound** topic.
2. Map the '(Data)' column to the appropriate attributes. In this case, since the column holds both the color and the material it is mapped to the 'Primary Color' attribute and the 'Material' attribute.

A warning message is displayed saying that the source column has already been mapped. Click 'Yes' to continue Mapping. For details, see the **Mapping the Same Column Multiple Times** section of the **Inbound Map Data - Map** topic.

3. In the Results area, select the column mapped to the 'Primary Color' attribute, and click **Transform**.
4. Add required transformations:
 - The 'Starting from position' transformation is used so that the exact letters that represent the color can be separated from the material.
 - The 'Replace whole value' is used to update the two letter coding data to match the correct color data in STEP.
 - The 'Change case' is used to apply a capital letter to color names, with lower case letters that follow.
 - Click **Save** to close the Transformations dialog.

Specify the sequence of transformations in the table below.

Target: PrimaryColor

☐ Mandatory

Value - Source: (Data)

☒	Starting from position 1 extract 2 characters
☒	Replace bl with blue
☒	Replace gr with green
☒	Replace br with brown
☒	Change to Mixed case

[Add Transformation](#)

Dimension Point

☒ Constant:

☐ Source:

[Add Transformation](#)

[Save](#) [Reset](#) [Cancel](#)

Preview

PrimaryColor=(Data) ✓

Blue
Green
Brown
Green

Note: Transformations work in sequence from top to bottom.

5. In the Results area, select the column mapped to the 'Material' attribute, and click **Transform**.
6. Add the 'Extract Characters at position' transformation to indicate that the material information starts with character 3, and that the longest material is 6 letters long.

Specify the sequence of transformations in the table below.

Target:Material

☐ Mandatory

Value - Source:(Data)

Starting from position extract characters

[Add Transformation](#)

Dimension Point

☒ Constant: (From Import Context)

☐ Source: <ID>

[Add Transformation](#)

Preview

Material=((Data)) ✓

cotton
wool
wool
cotton

7. Once each column is properly transformed, the results field should now look accurate, and it is ready to go on to the next steps for import.

Map Data

Source:

<ID>	(Data)
107601	blcotton
20803	grwool
20805	brwool
109011	grcotton

Result:

Map to: Product

ID=<ID> ✓	PrimaryColor=((Data)) ✓	Material=((Data)) ✓
107601	Blue	cotton
20803	Green	wool
20805	Brown	wool
109011	Green	cotton

Extract Characters Between and Extract Characters After Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

Transformations can be used to separate attribute values from a single field of data, and create values for multiple attributes in STEP.

This example includes the following transformations:

- Extract characters between two pieces of text in the value
- Extract characters after a given text in the value

This example describes how to extract and clean information from one field of an import file, and prepare it to be loaded into separate attributes in STEP. The following data is being imported, and contains data for multiple STEP attributes within a single field 'Primary Color and List Price':

Map Data

Source:

<ID>	> Primary Color and List Price	>
107601	aBlackz 20	
20803	aGreenz 41	
20805	aRedz 33	
109011	aWhitez 27	

Result:

Map to: Product

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Configuration

1. Load the import file into the inbound tool. For more information, see the **Data Exchange** topic.
2. Map the object ID in the import tool. See **ID or Key - Map Inbound** for details.
3. Map the source column that holds multiple values to the required attributes. When mapping the same source column more than once, you must click Yes to confirm that you want to map it again.

Selected Column Already Mapped

The selected column has already been mapped. Do you want to map it again?

Yes

No

In this example, we map 'Primary Color and List Price' two times, once to the Primary Color attribute, and again to the List Price attribute. In our example, the Result section shows the newly mapped items in red because the current values are not valid.

Note: Hover over a red cell to display the error message.

Map Data

Source:

<ID>	> Primary Color and List Price >
107601	aBlackz 20
20803	aGreenz 41
20805	aRedz 33
109011	aWhitez 27

Result: Map to: Product ▼

ID = <ID> ✓	PrimaryColor=Primary Color and List Price ⛔	ListPrice=(Primary Color and List Price) ⛔
107601	aBlackz 20	aBlackz 20
20803	aGreenz 41	aGreenz 41
20805	aRedz 33	aRedz 33
109011	aWhitez 27	aWhitez 27

- Select the one of the result columns mapped to an attribute and click the **Transform** button.

Map Data

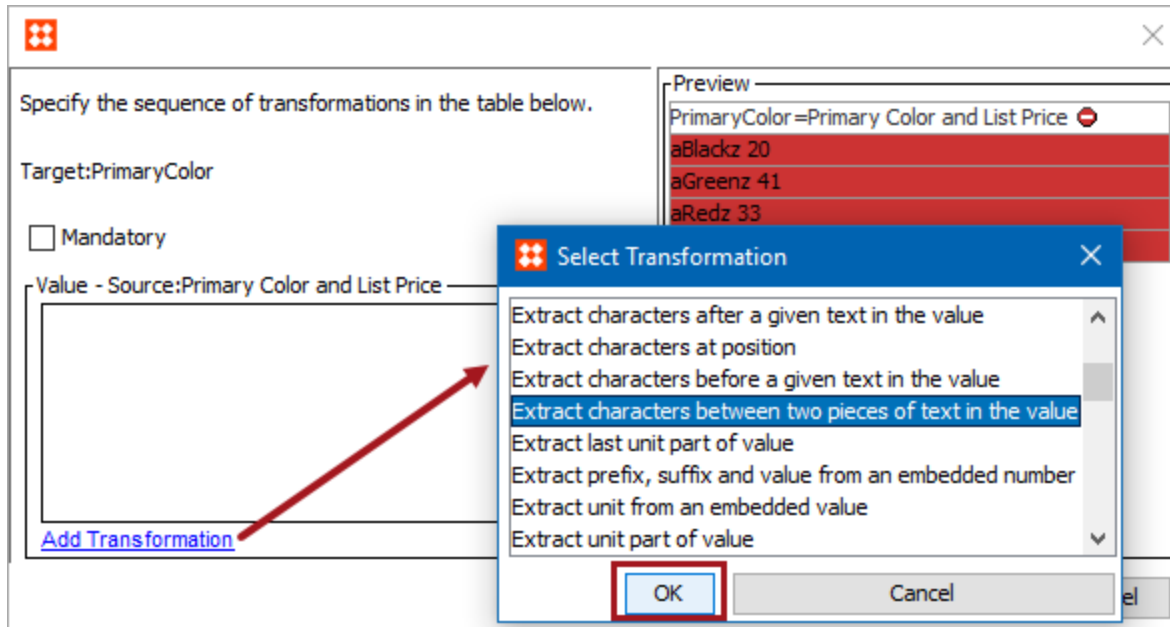
Source:

<ID>	> Primary Color and List Price >
107601	aBlackz 20
20803	aGreenz 41
20805	aRedz 33
109011	aWhitez 27

Result: Map to: Product ▼

ID = <ID> ✓	PrimaryColor=Primary Color and List Price ⛔	ListPrice=(Primary Color and List Price) ⛔
107601	aBlackz 20	aBlackz 20
20803	aGreenz 41	aGreenz 41
20805	aRedz 33	aRedz 33
109011	aWhitez 27	aWhitez 27

- In the Transformation window, click on the **Add Transformation** link, select the **Extract characters between two pieces of text in the value** transformation, and click **OK**.



- Enter the characters that should be used to determine that data that needs to be extracted and click out of the text fields. In this example, the color is always between the letters 'a' and 'z'. Click or tab out of the parameter to complete the configuration. The Preview pane shows the results of the transformation.

Click or tab out of the parameter to complete the configuration. The Preview pane shows the results of the transformation. If any additional transformations are needed for this data, click the **Add Transformation** to add them.

Important: Transformations are evaluated from the top down. Carefully consider the order of the transformations applied. To remove or rearrange transformations, use the Delete, Up arrow, and Down arrow buttons .

Specify the sequence of transformations in the table below.

Target: PrimaryColor

☐ Mandatory

Value - Source: Primary Color and List Price

Extract characters between and

[Add Transformation](#)

Preview

PrimaryColor = Primary Color and List Price ✓
Black
Green
Red
White

[Save](#) [Reset](#) [Cancel](#)

- If necessary, check the 'Mandatory' option to skip objects with empty values during import.
- Click **Save** to display the transformed data in the Map Data step.

Map Data

Source:

<ID>	Primary Color and List Price
107601	aBlackz 20
20803	aGreenz 41
20805	aRedz 33
109011	aWhitez 27

Result:

Map to:

ID = <ID> ✓	PrimaryColor = (Primary Color and List Price) ✓	ListPrice = (Primary Color and List Price) ⚠
107601	Black	aBlackz 20
20803	Green	aGreenz 41
20805	Red	aRedz 33
109011	White	aWhitez 27

[Auto Map](#) [Map](#) [Constant](#) [Remove](#) [Transform](#) [Generate Profile](#)

- Repeat step 3 for the remaining mapped column in the result area.
- Use the **Extract characters after a given text in the value** transformation in order to remove all text except the price. Although not visible in the following image, a 'space' character was added for the 'Extract characters after' parameter, and results in extracting only the price as shown below.

Specify the sequence of transformations in the table below.

Target: ListPrice

☐ Mandatory

Value - Source: Primary Color and List Price

☐ Extract characters after

[Add Transformation](#)

Unit

☐ Constant: \$

☒ Source: Primary Color and List Price

☐ Extract unit part of value

[Add Transformation](#)

Preview

ListPrice=(Primary Color and List Price) ✓

20
41
33
27

[Save](#) [Reset](#) [Cancel](#)

11. Click **Save** to display all transformed data in the Map Data step.

Map Data

Source:

<ID>	Primary Color and List Price
107601	aBlackz 20
20803	aGreenz 41
20805	aRedz 33
109011	aWhitez 27

Result:

Map to: Product

ID=<ID> ✓	PrimaryColor=(Primary Color and List Price) ✓	ListPrice=(Primary Color and List Price) ✓
107601	Black	20
20803	Green	41
20805	Red	33
109011	White	27

[Auto Map](#)

[Map](#)

[Constant](#)

[Remove](#)

[Transform](#)

[Generate Profile](#)

Format Number Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

Transformations can be used to format numbers, for example, to set a specified number of decimals, as required by an attribute. For example, the following data is being imported, and contains no decimal places:

Map Data

Source:

<ID>	ListPrice
179915	15
18212	17
124146	50
18216	7

Result:

Map to: Product

Auto Map

Map

Constant

Remove

Transform

Generate Profile

1. Map the object ID in the import tool.
2. Map the source column that holds the data to the required attribute. In this example, we map 'ListPrice' to the attribute of the same name.
3. Select the mapped column with the price data and click the **Transform** button.

Map Data

Source:

<ID>	ListPrice
179915	15
18212	17
124146	50
18216	7

Result:

Map to: Product

ID = <ID> ✓	ListPrice = (ListPrice) ✓
179915	15
18212	17
124146	50
18216	7

Auto Map

Map

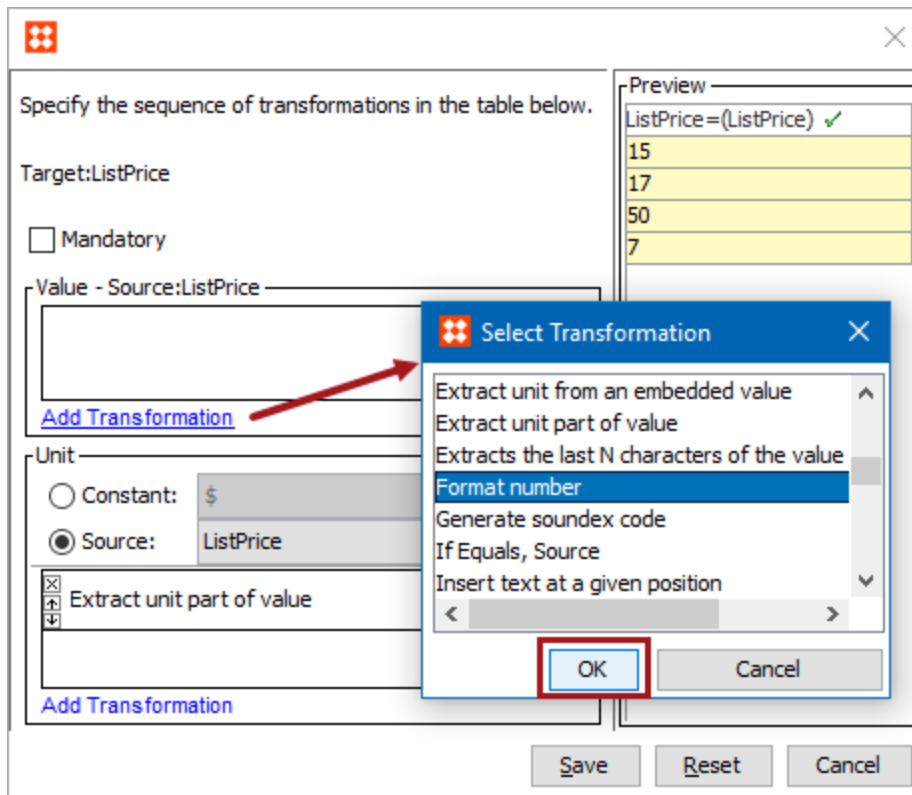
Constant

Remove

Transform

Generate Profile

- Click the **Add Transformation** link, and choose the 'Format number' transformation. Click **OK**.



- For the Unit section, select the **Constant** radio button, and use the dropdown to add the correct currency symbol to the value.

Specify the sequence of transformations in the table below.

Target: ListPrice

☐ Mandatory

Value - Source: ListPrice

Format number according to format #.00

Add Transformation

Unit

☒ Constant: \$

☐ Source: ListPrice

Add Transformation

Preview

ListPrice=(ListPrice) ✓

15.00 \$
17.00 \$
50.00 \$
7.00 \$

Save Reset Cancel

6. Click **Save** to display the results in the Map Data step.

Map Data

Source:

<ID>	ListPrice
179915	15
18212	17
124146	50
18216	7

Result:

Map to: Product

ID = <ID> ✓	ListPrice = (ListPrice) ✓
179915	15.00 \$
18212	17.00 \$
124146	50.00 \$
18216	7.00 \$

Auto Map Map Constant Remove Transform Generate Profile

Header and Value Aspects Example

This example shows a variety of mappings and exports using a single attribute, named 'Color'. Although the format illustrated is Excel, using other formats would produce the same results. For more information on each outbound mapping option, see the **Outbound Map Data - Data Source** documentation.

For more information on the available aspects, see the **Aspect - Transform Outbound** documentation.

Default Header Aspect and Default Value Aspect

The following products have values for the Color attribute and are displayed using the multi object editor:

Products			
Products References Referenced By			
View: Color			
	ID	Name	Color
Name	ID	Name	Color
AC-P7000-55	179624	AC-P7000-55	Black
AC-P7000-65	179625	AC-P7000-65	Blue
AC-P7000-79	179626	AC-P7000-79	Black
AC-P7000-83	179807	AC-P7000-83	Blue

The ID and the Color attribute are mapped using the default settings. When the flipper for the mapped column is opened (⌵), separate header and value parts are displayed. When mapping an attribute, by default, the attribute name is mapped to the header part and the value of the attribute is mapped to the value part.

Map Data

<Page Number>

All Attributes

- Attribute Group
- Attribute Group C
- Blacklisted - 2
- Blacklisted Sales Item Maintenance
- Business Rules
- Category Specific Attributes
- Customer Data
- Dates
- Description Attributes
- Display
 - Discontinued Product Maintenance
 - Item Maintenance
 - Item Description Information
 - Base Unit of Measure
 - Color

Exports data in Excel format.

Column (2 mapped)

<ID> ID

Color Value and unit

HeaderColor Name

ValueColor Value and unit

Inherit Data and References

Results

Exporting with this mapping produces the following result file:

	A	B
1	<ID>	Color
2	179624	Black
3	179625	Blue
4	179626	Black
5	179807	Blue

ID Header Aspect and Default Value Aspect

The data exported as header and value is determined by the aspect setting. You can include the attribute ID in the header using the transform button () and selecting the ID aspect for the header column.

Map Data

Exports data in Excel format.

Column (2 mapped)

<ID> ID

Color Value and unit

Header Color Name

Transformations

Target: Header

Source: Color Name

Aspect: Name

Name

Add Transform ID

Value

Value and unit

Unit

Unit ID

Base unit value

Base unit name

Inherit Data and References ☒

Results

Exporting with the transformation produces the following result file:

	A	B
1	<ID>	ColorID
2	179624	Black
3	179625	Blue
4	179626	Black
5	179807	Blue

ID Header Aspect and ID Value Aspect

You could also generate a file output with an additional data column like product ID, attribute ID, and the attribute value by adding the ID aspect for the value part of the mapping.

Map Data

<ID>

<Name>

<Parent ID>

<Object Type Name>

<Product-Override Child ID>

<AttributeLinks>

<Is deleted>

"Constant Value"

<Page Number>

All Attributes

Select Attribute

Classifications

Index Words

Product Classification Links

Product References

Asset References

Classification References

Entity References

STEP Workflow Task Info

Inherit Data and References

Exports data in Excel format.

Column (2 mapped)

<ID> ID

Color Value and unit

Header Color ID

Value Color Value and unit

Transformations

Target: Value

Source: Color Value and unit

Aspect: Value and unit

Name

Add Transformation

Value

Value and unit

Unit

Unit ID

Base unit value

Base unit name

Results

Exporting with this mapping results in the following file, which includes the attribute ID in both the header and the value for each product:

	A	B
1	<ID>	ColorID
2	179624	ColorID
3	179625	ColorID
4	179626	ColorID
5	179807	ColorID

Constant Value, ID Header Aspect, and ID Value Aspect

Now you can map a Constant Value to the header part by selecting "Constant Value" data source on the left and map it to the header as shown below.

The screenshot shows the 'Map Data' interface. On the left, a list of data sources includes '<ID>', '<Name>', '<Parent ID>', '<Object Type Name>', '<Product-Override Child ID>', '<AttributeLinks>', '<Is deleted>', 'Constant Value', '<Page Number>', 'All Attributes', 'Select Attribute', 'Classifications', 'Index Words', 'Product Classification Links', 'Product References', and 'Asset References'. The 'Constant Value' source is selected. A red arrow points from this source to the 'Header' field in the 'Color ID' aspect of the 'Column (2 mapped)' table. Below the table, an 'Enter Value' dialog box is open, prompting 'Enter constant value' with the text 'Attribute ID' entered. The dialog has 'Save' and 'Cancel' buttons. The 'Inherit Data and References' checkbox is checked at the bottom left.

After saving the constant value, there are two elements for the header.

The screenshot shows the 'Map Data' interface after saving the constant value. The 'Constant Value' source is still selected. The 'Header' field in the 'Color ID' aspect of the 'Column (2 mapped)' table now contains two elements: 'Color ID' and '"Attribute ID"'. A red box highlights these two elements. The 'Inherit Data and References' checkbox is checked at the bottom left.

For more information on using the Constant Value data source, see the **Constant Value - Data Source Outbound** documentation.

Results

Exporting with this mapping results in a file where the header is a concatenation of the attribute ID (ColorID) and the constant value 'Attribute ID'.

	A	B
1	<ID>	ColorIDAttribute ID
2	179624	ColorID
3	179625	ColorID
4	179626	ColorID
5	179807	ColorID

Constant Value and Value Aspect

Click the delete (X) button for the Header > Color ID element to remove the actual attribute ID from the header.

Add the third column by mapping the Color attribute again, and replace the header part with an 'Attribute Value' constant.

The full mapping now includes three columns, that contain information for the product ID, the ID for the attribute named Color, and the value for the attribute named Color.

Map Data

- <ID>
- <Name>
- <Parent ID>
- <Object Type Name>
- <Product-Override Child ID>
- <AttributeLinks>
- <Is deleted>
- Constant Value"
- <Page Number>
- + All Attributes
- Select Attribute
- + Classifications
- + Index Words
- + Product Classification Links
- + Product References
- + Asset References
- + Classification References
- + Entity References

Exports data in Excel format.

Column (3 mapped)

<ID> ID

Header "<ID>"

Value <ID> ID

Color ID

Header "Attribute ID"

Value Color ID

Color Value

Header "Attribute Value"

Value Color Value

Inherit Data and References ☒

Results

Exporting with this mapping results in the following file:

	A	B	C
1	<ID>	Attribute ID	Attribute Value
2	179624	ColorID	Black
3	179625	ColorID	Blue
4	179626	ColorID	Black
5	179807	ColorID	Blue

If Equals Source Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

Using the **If Equals, Source** transformation, you can make specific columns in a row or column based import file dependent on the content of a mapped column.

For example, a column in your import file specifies whether a product is for print or web publication, and another column contains color information in RGB Hex and CMYK. You now want to populate an attribute with CMYK information if the product is for print and RGB if it is for web.

First, map the column that contains the web and print information to the PrimaryColor attribute in STEP. Next, apply the If Equals, Source transformation. The transformation specifies to get data from the RGB Hex column if the value equals web and otherwise to get data from the CMYK column.

Steps

1. Select Configuration

2. Select Data Source

3. Select Format

4. Map Data

5. Generate Profile

Map Data

Source:

ID	Output	CMYK	RGB Hex
114852	web	C:0M:31Y:37K:0	#FFB0A1
114854	Print	C:0M:52Y:100K:64	#SC2C00

Specify the sequence of transformations in the table below.

Target:PrimaryColor

☐ Mandatory

Value - Source:Output

If value equals

web

Source

RGB Hex

Else source

CMYK

Add Transformation

Dimension Point

☒ Constant: (From Import Context)

☐ Source: ID

Add Transformation

Save

Reset

Cancel

Preview

PrimaryColor=Output ✓

#FFB0A1

C:0M:52Y:100K:64

Map to: Product

PrimaryColor=Output ✓

web

Print

Auto Map

Map

Constant

Remove

Transform

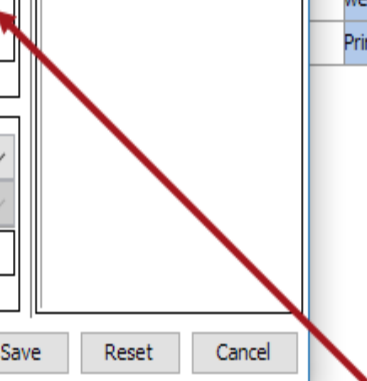
Generate Profile

Back

Next

Finish

Cancel



Insert Text Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

Inserting text into the attribute values while importing data is possible using the **Insert text before** and the **Insert text at a given position** transformations.

This example describes how to add new text to a field in an import file, and prepare it to be loaded into an attribute in STEP. The following data is being imported and contains incomplete data for the 'Manufacturer Name' attribute:

Map Data

Source:

<ID>	> Manufacturer Name >
107601	HanesBrands
20803	HanesBrands
20805	HanesBrands
109011	HanesBrands

Result:

Map to:
Product

Auto Map

Map

Constant

Remove

Transform

Generate Profile

1. Load the import file into the inbound tool. For more information, see the **Data Exchange** topic.
2. Map the object ID in the import tool. See **ID or Key - Map Inbound** for details.
3. Map the next source column to the required attribute. In this example, we map 'Manufacturer Name' to the Manufacturer Name attribute.

Map Data

Source:

<ID>	> Manufacturer Name >
107601	HanesBrands
20803	HanesBrands
20805	HanesBrands
109011	HanesBrands

Result:

Map to: Product ▼

ID = <ID> ✓	ManufacturerName = Manufacturer Name ✓
107601	HanesBrands
20803	HanesBrands
20805	HanesBrands
109011	HanesBrands

Auto Map

Map

Constant

Remove

Transform

Generate Profile

- Select the result column that contains data to be modified and click the **Transform** button.

Map Data

Source:

<ID>	> Manufacturer Name >
107601	HanesBrands
20803	HanesBrands
20805	HanesBrands
109011	HanesBrands

Result:

Map to: Product ▼

ID = <ID> ✓	ManufacturerName = Manufacturer Name ✓
107601	HanesBrands
20803	HanesBrands
20805	HanesBrands
109011	HanesBrands

Auto Map

Map

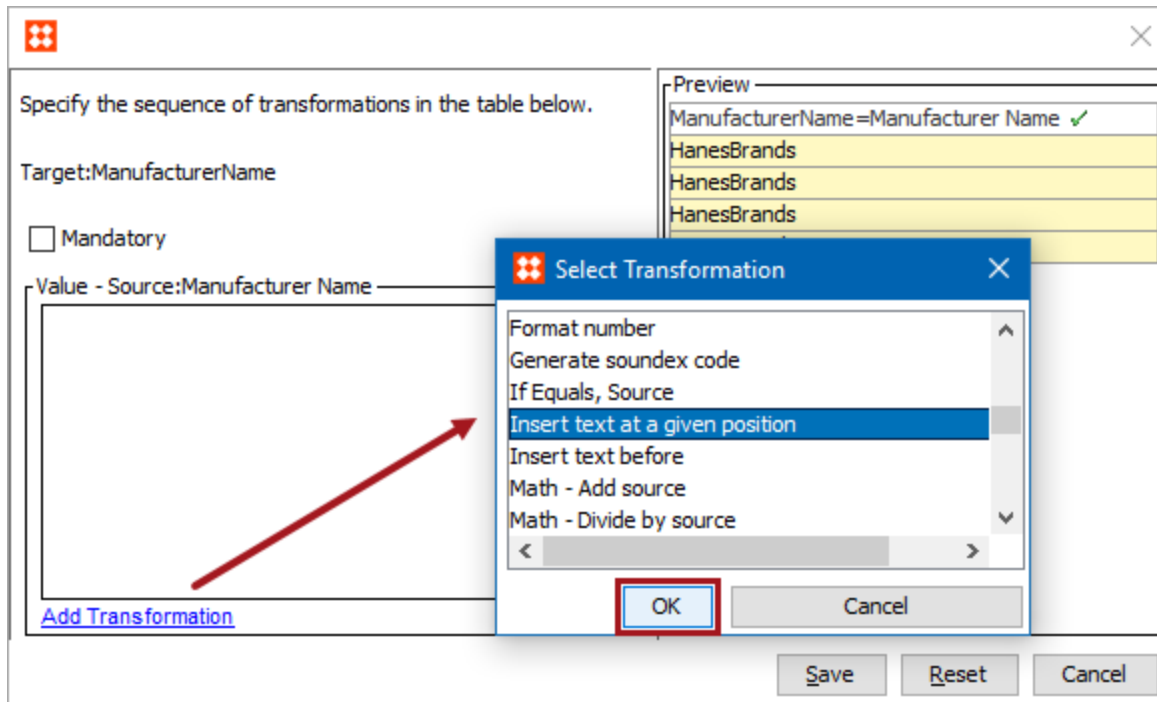
Constant

Remove

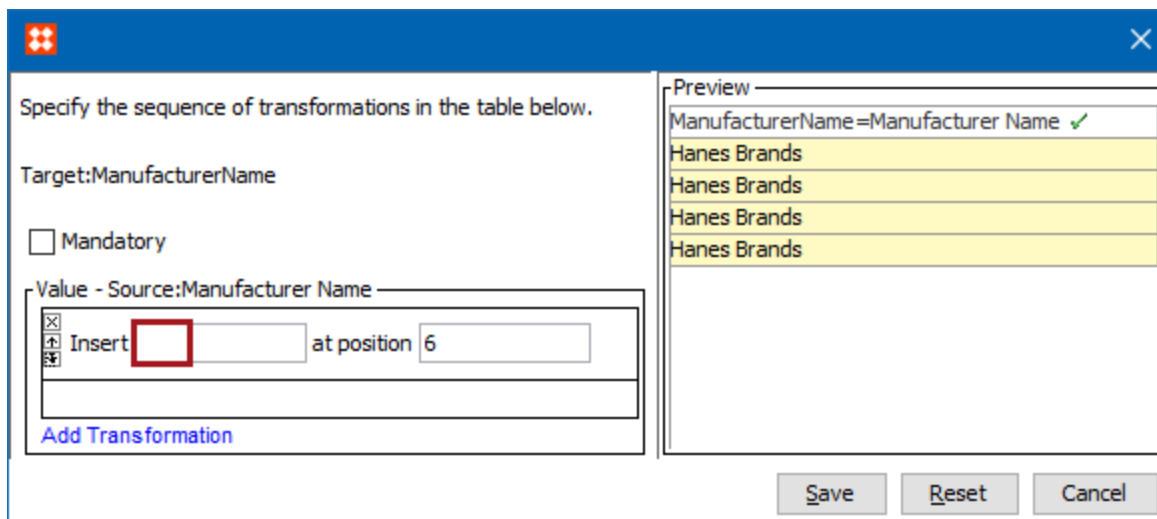
Transform

Generate Profile

- In the Transformation window, click on the **Add Transformation** link, select the **Insert text at a given position** transformation, and click **OK**.



6. Although not obvious in the following image, add a space in the **Insert** parameter, and indicate the position of the space by adding a number in the **at position** parameter. In this example, since the space belongs between the words 'Hanes' and 'Brands', we enter the number 6 for the position so the space is inserted as character 6. Click or tab out of the parameter to complete the configuration. The Preview pane shows the results of the transformation.



7. In the Transformation window, click on the **Add Transformation** link again, select the **Insert text before** transformation, and click **OK**.
8. In the **Insert before** parameter, add the text that should precede the existing value. Remember to add any necessary space characters. Click or tab out of the parameter to update the Preview pane.

Specify the sequence of transformations in the table below.

Target:ManufacturerName

☐ Mandatory

Value - Source:Manufacturer Name

Insert		at position	6
Insert	Ultimate	before	

[Add Transformation](#)

Preview

ManufacturerName=Manufacturer Name ✓
Ultimate Hanes Brands
Ultimate Hanes Brands
Ultimate Hanes Brands
Ultimate Hanes Brands

[Save](#) [Reset](#) [Cancel](#)

Important: Transformations are evaluated from the top down. Carefully consider the order of the transformations applied. To remove or rearrange transformations, use the Delete, Up arrow, and Down arrow buttons .

9. If necessary, check the 'Mandatory' option to skip objects with empty values during import.
10. Click **Save** to display the transformed data in the Map Data step.

Map Data

Source:

<ID>	> Manufacturer Name >
107601	HanesBrands
20803	HanesBrands
20805	HanesBrands
109011	HanesBrands

Result:

Map to: Product

ID = <ID> ✓	ManufacturerName = (Manufacturer Name) ✓
107601	Ultimate Hanes Brands
20803	Ultimate Hanes Brands
20805	Ultimate Hanes Brands
109011	Ultimate Hanes Brands

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Math Operation Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

Using the **Math operation (+, -, *, /)** transformation, basic math operations can be applied to inbound data mapped to attributes with Validation Base Types for numbers (for example, integer or number). Values can then be imported.

This example applies each of the math operations to a single field of the import file as shown below.

Map Data

Source:

<ID>	> Cost	>
107601	5	
20803	1	
20805	4	
109011	7	

Result:

Map to: Product

Auto Map

Map

Constant

Remove

Transform

Generate Profile

1. Load the import file into the inbound tool. For more information, see the **Data Exchange** topic.
2. Map the object ID in the import tool. See **ID or Key - Map Inbound** for details.
3. Map the next source column to the required attribute. In this example, we map 'Cost' to the Cost attribute.

Map Data

Source:

<ID>	> Cost	>
107601	5	
20803	1	
20805	4	
109011	7	

Result:

Map to: Product

ID = <ID> ✓	Cost = (Cost) ✓
107601	5
20803	1
20805	4
109011	7

Auto Map

Map

Constant

Remove

Transform

Generate Profile

4. Select the result column that contains data to be modified and click the **Transform** button.

Map Data

Source:

<ID>	> Cost
107601	5
20803	1
20805	4
109011	7

Result:

Map to: Product

ID=<ID> ✓	Cost=(Cost) ✓
107601	5
20803	1
20805	4
109011	7

5. In the Transformation window, click on the **Add Transformation** link, select the **Math operation (+,-,*,/)** transformation, and click **OK**.

Specify the sequence of transformations in the table below.

Target: Cost

☐ Mandatory

Value - Source: Cost

[Add Transformation](#)

Unit

☐ Constant: \$

☒ Source: Cost

☒ Extract unit part of value

[Add Transformation](#)

Select Transformation

- Insert text before
- Math - Add source
- Math - Divide by source
- Math - Multiply by source
- Math - Subtract source
- Math operation (+,-,*,/)**
- Parse Address

Preview

Cost=(Cost) ✓
5
1
4

6. In the **Value** parameter, use the dropdown to choose the **+** math operation, and in the text field, enter the number that should be added to the inbound value. In this example, 3 is added to the value.

Click or tab out of the parameter to complete the configuration. The Preview pane shows the results of the transformation.

Specify the sequence of transformations in the table below.

Target: Cost

☐ Mandatory

Value - Source: Cost

Value = Value + 3

Add Transformation

Unit

☐ Constant: \$

☒ Source: Cost

Extract unit part of value

Add Transformation

Preview

Cost=(Cost) ✓

10

6

9

12

Save Reset Cancel

7. Click **Save** to display the transformed data in the Map Data step.
8. To illustrate each math operation, three additional attributes for Cost are used: Cost1, Cost2, and Cost3. Repeat steps 3 - 6, selecting one of the additional Cost attributes each time, confirming to map the same column again, and selecting a different math operation each time as follows:
 - In the **Value** parameter, use the dropdown to choose the **-** math operation, and in the text field, enter the number that should be subtracted from the inbound value. In this example, 1 is subtracted from the value.

Value - Source: Cost

Value = Value - 1

Add Transformation

- In the **Value** parameter, use the dropdown to choose the ***** math operation, and in the text field, enter the number that should be multiplied with the inbound value. In this example, the value is multiplied by 2.

Value - Source:Cost

Value = Value * 2

Add Transformation

- In the **Value** parameter, use the dropdown to choose the / math operation, and in the text field, enter the number that should be used to divide the inbound value. In this example, the value is divided by 2.

Value - Source:Cost

Value = Value / 2

Add Transformation

- If necessary, check the 'Mandatory' option to skip objects with empty values during import.
- Click **Save** to display all of the transformations in the Map Data step Result section as follows:
 - The **Cost = (Cost)** column shows where the original value had 3 added.
 - The **Cost1 = (Cost)** column shows where the original value had 1 subtracted.
 - The **Cost2 = (Cost)** column shows where the original value was multiplied by 2.
 - The **Cost3 = (Cost)** column shows where the original value was divided by 2.

Map Data

Source:

<ID>	> Cost
107601	5
20803	1
20805	4
109011	7

Result:

Map to: Product

ID=<ID> ✓	Cost=(Cost) ✓	Cost1=(Cost) ✓	Cost2=(Cost) ✓	Cost3=(Cost) ✓
107601	8	4	10	2.5
20803	4	0	2	0.5
20805	7	3	8	2
109011	10	6	14	3.5

Auto Map

Map

Constant

Remove

Transform

Generate Profile

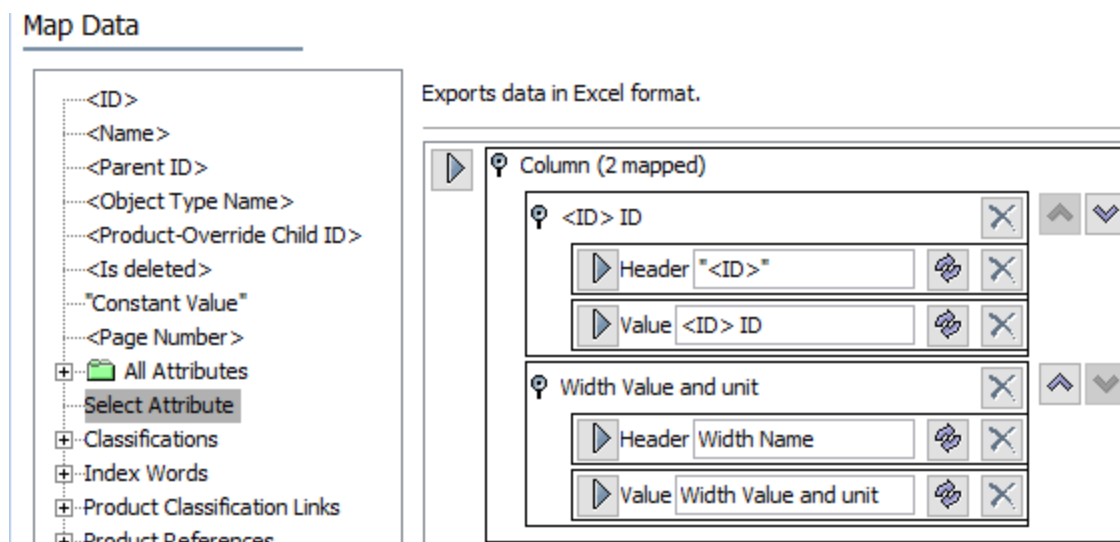
Merge Attributes and Constant Example

You can modify an output by concatenating separate pieces of data from STEP to display in the same field upon export.

For example, you can merge three attributes with the names width, depth, and height into one export field in the format 'width x depth x height', and also show only the value, not the unit. This mapping will include concatenating values (the measurement) with constant values (an abbreviation to indicate width, depth, or height).

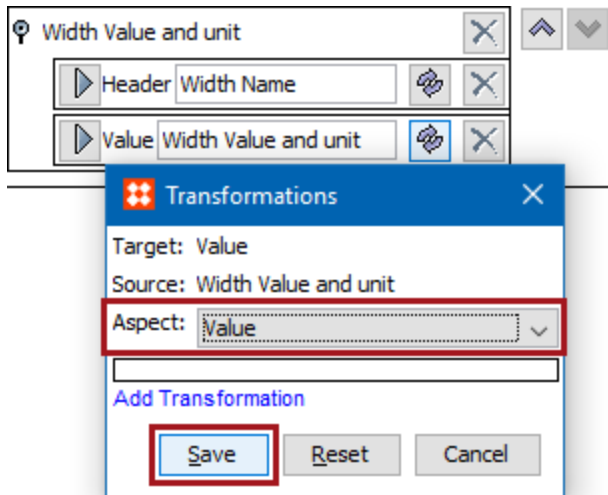
1. Map **<ID>**. See the **ID or Key - Data Source Outbound** topic for details.
2. Map **Select Attribute**, and search for the Width attribute. See the **Attributes and Data Containers - Data Source Outbound** topic for details.

Two items have been mapped and display in the right panel.

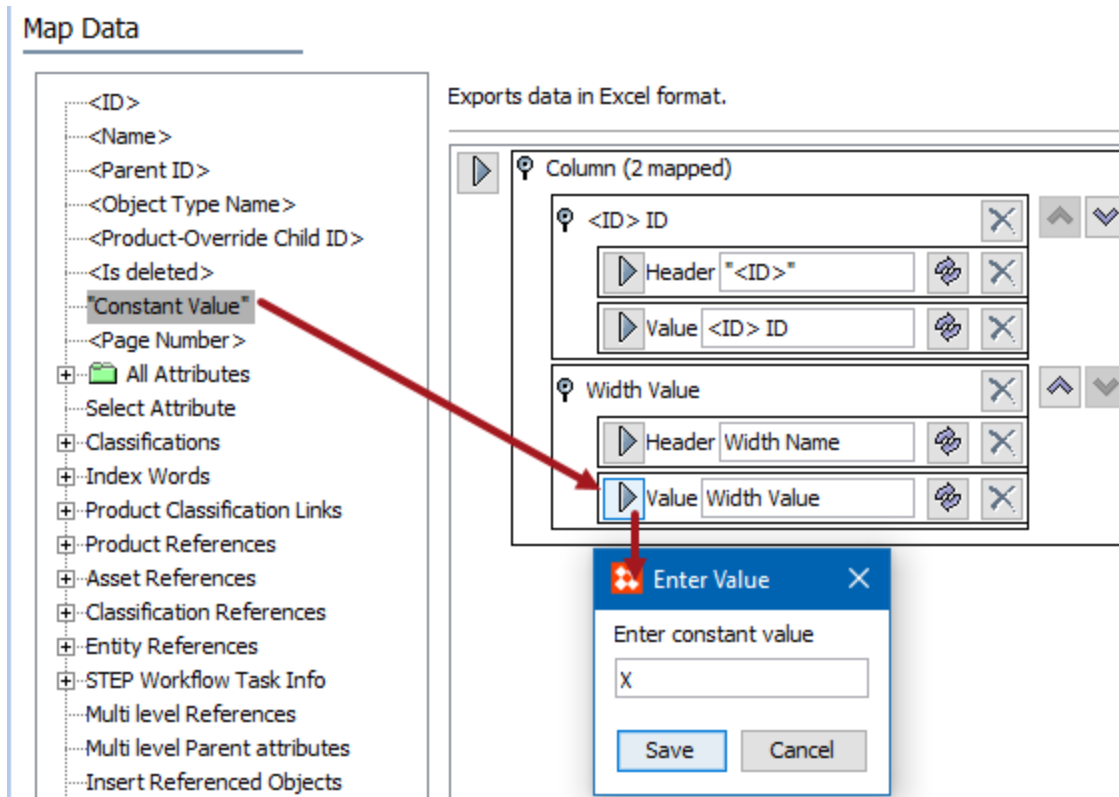


3. Open the Width attribute flipper to display the Header and Value elements.
4. Click the **Transformation** button () for the width attribute Value element to display the Transformations dialog.
5. From the **Aspect** parameter dropdown select **Value**, and click **Save** to apply the change.

Note: Although you removed the 'Value And Unit' aspect by choosing 'Value' aspect, if the attributes have valid units assigned to them, you can extract the units in the same manner.

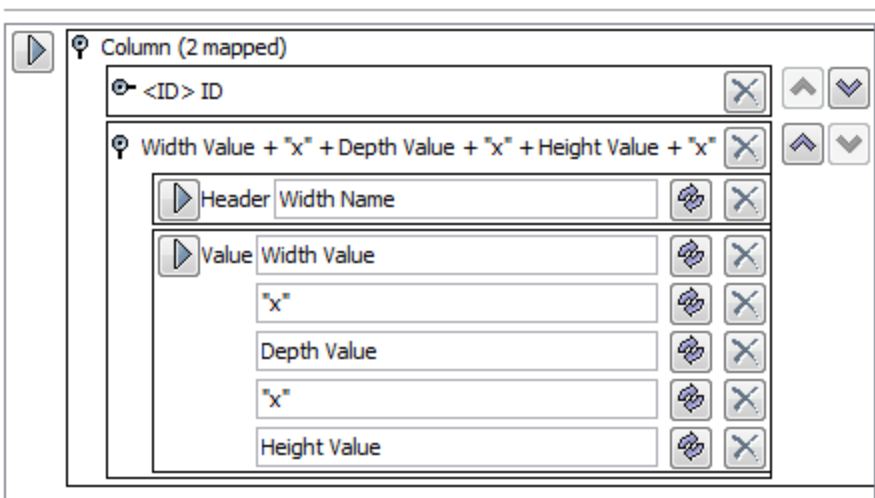


6. Map **Constant Value** to the Value element of your Width attribute in order to merge elements.
7. In the **Enter Value** dialog type **X** and click the **Save** button.



8. Repeat steps 2-7 and add the attributes depth and height, excluding the final X constant.
The right panel now looks like the following image.

Exports data in Excel format.



9. Complete any additional mappings and initiate the export.

Results

Using the mapping shown above, the following output is generated and includes the ID of the object and the values for the width, depth, and height attributes in a single column. When no value of attribute being merged is used along with a constant, the value will be skipped while the other attributes and constants appear as defined, as shown below when the depth value is not available.

	A	B
1	<ID>	Width
2	18212	13x10x12
3	124188	15xx16
4	124148	15x16x20
5	18216	12x11x12

When exporting in a tabular format, you can also add a user-defined column header by mapping a Constant Value for the Header element. Additionally, the header could be a combination of multiple pieces of data, using the merging steps above.

For more information on each outbound mapping option, see the **Outbound Map Data - Data Source** documentation.

Merge Data for Multi-Valued Data Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

Multiple values for a multi-valued attribute and multiple targets for a reference or link type can be imported from a single field in the import file. In this case when using a CSV, the values / Target IDs must be separated by a delimiter that is different from the delimiter used for separating columns (alternately, the value can be enclosed in quotes).

Below are the following examples of merging import data to result in values for a multi-valued attribute:

- **Merge single values from multiple columns** - in this scenario, the import file contains the values for a multi-valued attribute in more than one column. Transformations are used to combine the data into a single STEP attribute.
- **Merge single values from multiple columns using an Intermediate Variable** - in this scenario, the import file is the same as the first one (it contains the values for a multi-valued attribute in more than one column). Transformations are used on an intermediate variable to combine the data into a single STEP attribute. While the method is different, the result is the same as the first scenario.
- **Merge multiple values from multiple columns** - in this scenario, the import file includes two columns that each have multiple values in a single cell. The values from the first column should be paired with the corresponding value in the second column.

Merge single values from multiple columns

In this example, the import file contains the values for a multi-valued attribute in more than one column. Transformations are used to combine the data into a single STEP attribute. The end result is that the values of columns 'Color 1' and 'Color 2' are merged into a single multi-valued attribute named 'Color.'

1. Map the 'Color 1' column to the 'Color' attribute.

Map Data

Source:

<ID>	Color1	Color2
160323	Azure	Black
160324		Kiwi
160349	Canary	
160350	Denim	Emerald

Result:

ID = <ID> ✓

160323

160324

160349

160350

Auto Map Map

Map Color1 to

☐ ID
☐ Name
☒ Attribute
☐ Product Classification Link
☐ Product Reference
☐ Asset Reference
☐ Classification Reference
☐ Entity Reference
☐ Reference Meta-Data

Browse Search

Color (Color)

Search

Advanced

Name

> Color ID = Color

- In the **Result** area, select the 'Color=Color 1' column and click the **Transform** button to display the transformation dialog.
- In the **Before Multivalue Split** area, add the **Append text** with a semi-colon (;), and the **Append from source** transformation selecting the 'Color2' column from the dropdown.

Map Data

Source:

<ID>	> Co
160323	Az
160324	
160349	Ca
160350	De

Result:

ID=<ID> ✓	Color=Color 1 ✓
160323	Azure
160324	
160349	Canary
160350	Denim

Specify the sequence of transformations in the table below.

Target:Color

☐ Mandatory

Multivalue separator ;

Before Multivalue Split - Source:Color1

☒ Append ;

☒ Append from {Color2} ✓

Add Transformation

Value - Source:Color1

Add Transformation

Preview

Color=Color 1 ✓
Azure Black
Kiwi
Canary
Denim Emerald

Save Reset Cancel

Auto Map Map Constant Remove Transform Generate Profile

In the Preview area, verify that the rows with multiple values are displayed correctly with a pipe (|) separator. This is required for a successful import of multi-values.

- Click **Save** to display the result of the transformation in the Result area. Continue navigating through the mappings and rest of the wizard.

Map Data

Source:

<ID>	Color1	Color2
160323	Azure	Black
160324		Kiwi
160349	Canary	
160350	Denim	Emerald

Result:

Map to: Product

ID=<ID> ✓	Color=Color1 ✓
160323	Azure Black
160324	Kiwi
160349	Canary
160350	Denim Emerald

Auto Map
Map
Constant
Remove
Transform
Generate Profile

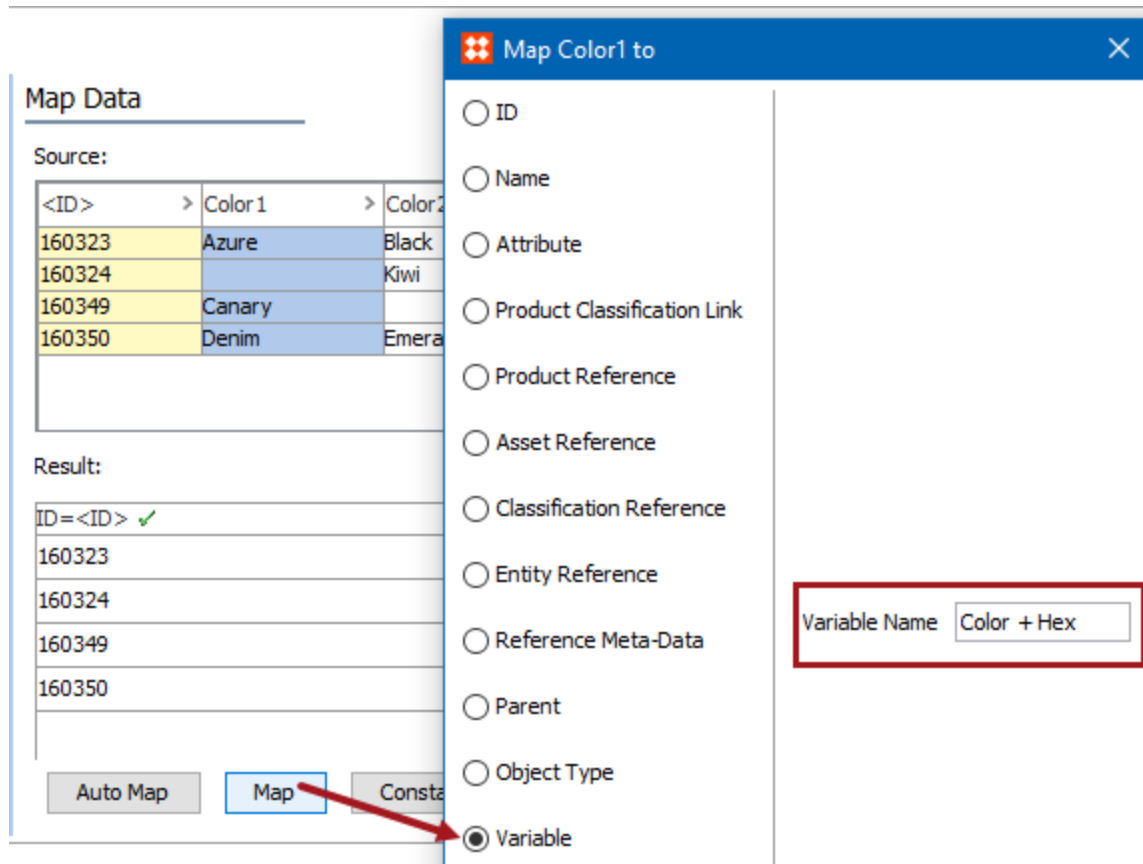
Viewing the products that were modified by the import, shows that the multi-valued Color attribute correctly includes multiple colors when available.

Products	
Products	References Referenced By
View: Color	
ID	Color
> ID	Color
> 160323	Azure Black
> 160324	Kiwi
> 160349	Canary
> 160350	Denim Emerald

Merge single values from multiple columns using an Intermediate Variable

In this example, the import file is the same as the one before (it contains the values for a multi-valued attribute in more than one column). Transformations are used on an intermediate variable to combine the data into a single STEP attribute. While the method is different, the result is the same as the first scenario, namely that the values of columns 'Color 1' and 'Color 2' are merged into a single multi-valued attribute named 'Color.'

1. Map the 'Color 1' column to the 'Color + Hex' variable.



- In the **Intermediate Variables** area, select the 'Color + Hex =Color 1' column and click the **Transform** button to display the transformation dialog.
- In the **Value** area, add the **Append text** with a semi-colon (;), and the **Append from source** transformation selecting the 'Color2' column from the dropdown.

Map Data

Source:

<ID>	Color 1	Color 2
160323	Azure	Black
160324		Kiwi
160349	Canary	

Intermediate Variables:

Color + Hex=Color1

Azure

Canary

Denim

Result:

ID=<ID> ✓

160323

160324

160349

Auto Map Map Constant Remove Transform Generate Profile

Specify the sequence of transformations in the table below.

Target: Color + Hex=Color1

☐ Mandatory

Value - Source: Color1

Append ;

Append from Color2

Add Transformation

Save Reset Cancel

Preview

Color + Hex=Color1 ✓

Azure;Black

;Kiwi

Canary;

Denim;Emerald

In the Preview area, verify that the rows with multiple values are displayed correctly with a semi-colon (;) separator.

- Click **Save** to display the result of the transformation in the Intermediate Variables area, select the 'Color + Hex =Color 1' variable column, and map it to the 'Color' attribute.

Map Data

Source:

<ID>	Color1	Color2
160323	Azure	Black
160324		Kiwi
160349	Canary	

Intermediate Variables:

```
Color + Hex=Color1
Azure;Black
;Kiwi
Canary;
Denim;Emerald
```

Result:

```
ID=<ID> ✓
160323
160324
160349
```

Map Color + Hex to

☐ ID
☐ Name
☒ Attribute
☐ Product Classification Link
☐ Product Reference
☐ Asset Reference
☐ Classification Reference
☐ Entity Reference
☐ Reference Meta-Data
☐ Parent
☐ Object Type

Browse Search

Color (Color)

Advanced

☒ Color ID = Color

- Click **Save** to display the result of the transformation and variable mapping in the Result area. Continue navigating through the mappings and rest of the wizard.

Map Data

Source:

<ID>	Color1	Color2
160323	Azure	Black
160324		Kiwi
160349	Canary	
160350	Denim	Emerald

Intermediate Variables:

Color + Hex=Color1
Azure;Black
;Kiwi
Canary;
Denim;Emerald

Result:

Map to: Product

ID = <ID> ✓	Color = Color + Hex ✓
160323	Azure Black
160324	Kiwi
160349	Canary
160350	Denim Emerald

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Viewing the products that were modified by the import, shows that the multi-valued Color attribute correctly includes multiple colors when available. However, note that when first column mapped has no value, indicated by the arrow below, that the import does not include the second value.

Products	
Products	References Referenced By
View: Color	
ID	Color
> ID	Color
> 160323	Azure Black
> 160324	
> 160349	Canary
> 160350	Denim Emerald

Merge multiple values from multiple columns

In this example, the import file includes two columns that each have multiple values in a single cell. The color values from the first column ('Colors') should be paired with the corresponding hexadecimal values in the second column ('Hex'). The end result is that the 'Colors' and 'Hex' columns are merged into a multi-valued attribute ('Color-Hex'), displaying each as a color-hexadecimal pair.

1. Map 'Colors' to 'Color-Hex'.

The screenshot shows the 'Map Data' interface. The 'Source' table has columns: <ID>, Colors, and Hex. The 'Result' table shows the mapped data. A 'Map Colors to' dialog is open, showing the 'Attribute' radio button selected. The 'Color-Hex (ColorHex)' field is highlighted with a red box, showing the mapping: > Color-Hex ID = ColorHex.

<ID>	Colors	Hex
160323	Black; Yellow	#000000; #ffd700
160324	Red	#cc0000
160349	Navy; Green	#000080; #008000
160350	Gray; Teal	#cccccc; #008080

Result:

ID = <ID> ✓
160323
160324
160349
160350

Map Colors to

☐ ID
☐ Name
☒ Attribute
☐ Product Classification Link
☐ Product Reference
☐ Asset Reference
☐ Classification Reference
☐ Entity Reference

Color-Hex (ColorHex)

Advanced

Name

> Color-Hex ID = ColorHex

2. In the **Result** area, select the mapped 'Color-Hex=Colors' column, and click the **Transform** button to display the transformation dialog.
3. In the **Before Multivalue Split** area, add the **Append from multivalue source** transformation selecting the 'Hex' column from the dropdown. The separator, '-' in the example, is optional.

Map Data

Source:

<ID>	Colors	Hex
160323	Black; Yellow	#000000; #ffd700
160324	Red	#cc0000
160349	Navy; Green	#000080; #008000
160350	Gray; Teal	#cccccc; #008080

Result:

ID=<ID>	Color-Hex=Colors
160323	Black Yellow
160324	Red
160349	Navy Green
160350	Gray Teal

Auto Map
Map
Constant
Remove
Transform
Generate Profile

Specify the sequence of transformations in the table below.

Target:Color-Hex

☐ Mandatory

Multivalue separator ;

Before Multivalue Split - Source:Colors

☒ Append from multivalue: Hex Multivalue Separator; Separator -

Add Transformation

Value - Source:Colors

Add Transformation

Save
Reset
Cancel

Preview

Color-Hex=Colors ✓

Black-#000000|Yellow-#ffd700

Red-#cc0000

Navy-#000080|Green-#008000

Gray-#cccccc|Teal-#008080

In the Preview area, verify that the rows with multiple values are displayed correctly with a pipe (|) separator. This is required for a successful import of multi-values.

- Click **Save** to display the result of the transformation in the Result area. Continue navigating through the mappings and rest of the wizard.

Map Data

Source:

<ID>	> Colors	> Hex
160323	Black; Yellow	#000000; #ffd700
160324	Red	#cc0000
160349	Navy; Green	#000080; #008000
160350	Gray; Teal	#cccccc; #008080

< >

Result: Map to: Product

ID = <ID> ✓	Color-Hex = Colors ✓
160323	Black-#000000 Yellow- #ffd700
160324	Red-#cc0000
160349	Navy-#000080 Green- #008000
160350	Gray-#cccccc Teal- #008080

Auto Map Map Constant Remove Transform Generate Profile

Viewing the products that were modified by the import, shows that the multi-valued Color-Hex attribute correctly includes multiple colors and hexadecimal values when available.

Products

Products References Referenced By

View: Color

ID	Color-Hex
160323	Black-#000000 Yellow- #ffd700
160324	Red-#cc0000
160349	Navy-#000080 Green- #008000
160350	Gray-#cccccc Teal- #008080

Replace Substrings of the Value Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

The **Replace substrings of the value** transformation can be used to find and replace all matches of a specified text within a value.

This example describes how text within a field in an import file is modified to be loaded into an attribute in STEP. The following data is being imported, and contains outdated data for the 'Manufacturer Name' attribute:

Map Data

Source:

<ID>	> Manufacturer Name >
107601	Ultimate Hanes Brands
20803	Ultimate Hanes Brands
20805	Ultimate Hanes Brands
109011	Ultimate Hanes Brands

Result:

Map to: Product

Auto Map

Map

Constant

Remove

Transform

Generate Profile

1. Load the import file into the inbound tool. For more information, see the **Data Exchange** topic.
2. Map the object ID in the import tool. See **ID or Key - Map Inbound** for details.
3. Map the next source column to the required attribute. In this example, we map 'Manufacturer Name' to the Manufacturer Name attribute.

Map Data

Source:

<ID>	> Manufacturer Name >
107601	Ultimate Hanes Brands
20803	Ultimate Hanes Brands
20805	Ultimate Hanes Brands
109011	Ultimate Hanes Brands

Result:

Map to: Product

ID=<ID> ✓	ManufacturerName=Manufacturer Name ✓
107601	Ultimate Hanes Brands
20803	Ultimate Hanes Brands
20805	Ultimate Hanes Brands
109011	Ultimate Hanes Brands

Auto Map

Map

Constant

Remove

Transform

Generate Profile

4. Select the result column that contains data to be modified and click the **Transform** button.

Map Data

Source:

<ID>	> Manufacturer Name
107601	Ultimate Hanes Brands
20803	Ultimate Hanes Brands
20805	Ultimate Hanes Brands
109011	Ultimate Hanes Brands

Result:

Map to: Product

ID = <ID> ✓	ManufacturerName = Manufacturer Name ✓
107601	Ultimate Hanes Brands
20803	Ultimate Hanes Brands
20805	Ultimate Hanes Brands
109011	Ultimate Hanes Brands

Auto Map Map Constant Remove Transform Generate Profile

5. In the Transformation window, click on the **Add Transformation** link, select the **Replace substrings of the value** transformation, and click **OK**.

Specify the sequence of transformations in the table below.

Target: ManufacturerName

☐ Mandatory

Value - Source: Manufacturer Name

[Add Transformation](#)

Select Transformation

- Parse Address
- Prepend from source
- Remove last unit from value
- Remove unit from value
- Replace substrings of the value**
- Replace substrings of the value using a regular expression
- Replace the whole value
- Replace whole value using Lookup Table

OK Cancel

Preview

ManufacturerName = Manufacturer Name ✓
Ultimate Hanes Brands
Ultimate Hanes Brands
Ultimate Hanes Brands

Save Reset Cancel

6. In the **Replace sub-strings** parameter, add the text that should be removed.
7. In the **with** parameter, add the replacement text. In this example, the word 'Hanes' will be replaced with the word 'Cotton'. If multiple instances of the word 'Hanes' existed, all would be replaced.

Click or tab out of the parameter to complete the configuration. The Preview pane shows the results of the transformation.

Specify the sequence of transformations in the table below.

Target:ManufacturerName

☐ Mandatory

Value - Source:Manufacturer Name

Replace sub-strings	Hanes	with	Cotton

Add Transformation

Preview

ManufacturerName=Manufacturer Name ✓

Ultimate Cotton Brands

Ultimate Cotton Brands

Ultimate Cotton Brands

Ultimate Cotton Brands

Save Reset Cancel

Important: Transformations are evaluated from the top down. Carefully consider the order of the transformations applied. To remove or rearrange transformations, use the Delete, Up arrow, and Down arrow buttons .

8. If necessary, check the 'Mandatory' option to skip objects with empty values during import.
9. Click **Save** to display the transformed data in the Map Data step.

Map Data

Source:

<ID>	Manufacturer Name
107601	Ultimate Hanes Brands
20803	Ultimate Hanes Brands
20805	Ultimate Hanes Brands
109011	Ultimate Hanes Brands

Result:

Map to: Product

ID = <ID> ✓	ManufacturerName=(Manufacturer Name) ✓
107601	Ultimate Cotton Brands
20803	Ultimate Cotton Brands
20805	Ultimate Cotton Brands
109011	Ultimate Cotton Brands

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Split and Extract Data Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

In this example, you have a file that has dimensional data in one of the columns. Now, you want to split that dimensional information into three separate values for loading into STEP.

- 1. Map the product ID in the import tool. Then select the column that holds the dimensional data and begin to map it to the desired three separate attributes in STEP.

Map Data

Source:

<ID>	> Dimensions >	
22165	30x28x16	
22167	17x20x22	
22168	17x20x22	

Result: Map to: Product

ID=<ID> ✓
22165
22167
22168

Auto Map

Map

Constant

Remove

Transform

Generate Profile

- 2. Manually map the column to the first attribute that will be populated with some part of the data. In this case, the first group of numbers are mapped to the attribute 'Length'. Notice that the Results field shows it in red- this is expected until proper transformation is set.

Map Data

Source:

<ID>	Dimensions
22165	30x28x16
22167	17x20x22
22168	17x20x22

Result:

Map to: Product

ID=<ID> ✓	Length=(Dimensions) ⚠
22165	30 x28x16
22167	17 x20x22
22168	17 x20x22

- Map the column that holds the dimension data again to the next two attributes that are needed. Each time a warning message is displays saying that the source column has already been mapped and asks if you want to map it again. Click **Yes**.

Selected Column Already Mapped ✕


 The selected column has already been mapped. Do you want to map it again?

In this example, the same column has been mapped to three different attributes: Length, Width, and Height.

Map Data

Source:

<ID>	Dimensions
22165	30x28x16
22167	17x20x22
22168	17x20x22

Result:

Map to: Product

ID=<ID> ✓	Length=(Dimensions) ⚙	Width=(Dimensions) ⚙	Height=(Dimensions) ⚙
22165	30 x28x16	30 x28x16	30 x28x16
22167	17 x20x22	17 x20x22	17 x20x22
22168	17 x20x22	17 x20x22	17 x20x22

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Note that in this particular case, each of the three attributes mapped to have a default unit in centimeters, 'cm'.

- In the **Results** pane, select the first attribute that you want to perform a transformation on. In this example that is 'Length'. Click **Transform**. The following window appears.



Specify the sequence of transformations in the table below.

Target: Length

☐ Mandatory

Value - Source: Dimensions

☒ Remove unit from value

Add Transformation

Unit

☐ Constant: cm

☒ Source: Dimensions

☒ Extract unit part of value

Add Transformation

Save Reset Cancel

Preview

Length=(Dimensions) ⚙

30 x28x16

17 x20x22

17 x20x22

- In the Unit field, select **Constant**, and choose the desired measurement from the dropdown if there are choices. In this case inches, 'in' are chosen. In the Value-Source: Dimensions field, remove the transformation **Remove unit from value**.

The screenshot shows a dialog box titled "Specify the sequence of transformations in the table below." with a close button (X) in the top right corner. The dialog is divided into two main sections: a configuration area on the left and a preview area on the right.

Configuration Area:

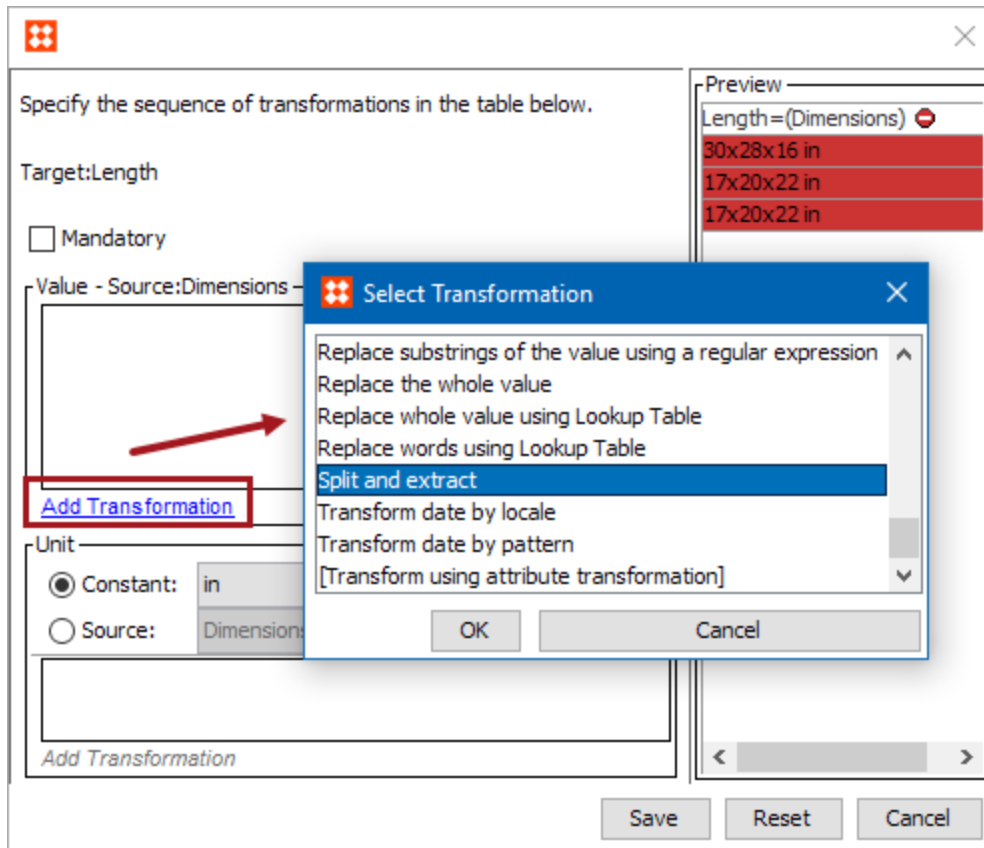
- Target:** Length
- Mandatory:** ☐
- Value - Source:** Dimensions (with a text input field below it)
- Add Transformation:** (blue link)
- Unit:**
 - ☒ **Constant:** in (dropdown menu)
 - ☐ **Source:** Dimensions (dropdown menu)
- Add Transformation:** (grey link)

Preview Area:

- Preview:** (title)
- Length=(Dimensions)** (with a red minus button)
- 30x28x16 in (highlighted in red)
- 17x20x22 in (highlighted in red)
- 17x20x22 in (highlighted in red)

Buttons: Save, Reset, Cancel

- Under the Value-Source: Dimensions field, click **Add Transformation**. A list of available transformation functions appears. Select the **Split and extract** transformation.



- In the **Split and extract** transformation, specify how the data in the original file's dimensional data column should be split. In this example, you want to split the data by looking at the information between the " x " (include the spaces), and extracting the data before the first occurrence of this string. Select the dropdown list in the extract field and choose '1'. This represents the first group of numbers in the series. Choose 2 to represent the second group of numbers, and so forth. (The maximum extract field is 10). The right panel gives a preview of the data that is extracted. If the preview is not displayed, click F2. Click **Save**.

Specify the sequence of transformations in the table below.

Target:Length

☐ Mandatory

Value - Source:Dimensions

Split by x and extract field 1

Add Transformation

Unit

☒ Constant: in

☐ Source: Dimensions

Add Transformation

SaveResetCancel

Preview

Length=(Dimensions) ✓

30 in

17 in

17 in

Note: If the 'Mandatory' option is checked, it will skip the empty values, and those objects will not be updated / imported into STEP.

8. In the **Map Data** screen, the column for Length shows a preview of the data that is loaded into STEP.

Map Data

Source:

<ID>	Dimensions
22165	30x28x16
22167	17x20x22
22168	17x20x22

Result:Map to: Product

ID=<ID> ✓	Length=(Dimensions) ✓	Width=(Dimensions) ⚠	Height=(Dimensions) ⚠
22165	30 in	30 x28x16	30 x28x16
22167	17 in	17 x20x22	17 x20x22
22168	17 in	17 x20x22	17 x20x22

Auto MapMapConstantRemoveTransformGenerate Profile

9. Repeat the transformation steps for the other two needed attributes, in this case 'Width' and 'Height'.

Map Data

Source:

<ID>	> Dimensions >
22165	30x28x16
22167	17x20x22
22168	17x20x22

Result:

Map to: Product

ID=<ID> ✓	Length=(Dimensions) ✓	Width=(Dimensions) ✓	Height=(Dimensions) ✓
22165	30 in	28 in	16 in
22167	17 in	20 in	22 in
22168	17 in	20 in	22 in

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Replace the Whole Value Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

This example describes how to extract and clean information from a legacy system, or groups of information stored in one column, and load it into separate attributes in STEP. If information is more technical or cryptic, see the last section of this topic on how to convert cryptic data.

This example includes the following transformations:

- Extract characters before a given text in the value (Extract Characters before)
- Change case
- Split and extract
- Replace the whole value

Data being transformed

Note: The preview displays how the transformations function on a sample of data, typically the first 20 rows. Prior to starting the import, verify that the first 20 objects contain the entries with transformations you want to verify.

Map Data

Source:

<ID>	> (Data)
107601	blue 2 cotton China
20803	green 4 wool USA
20805	brown 3 wool USA
109011	pink 2 cotton UK

Result:

Map to: Product

Auto Map

Map

Constant

Remove

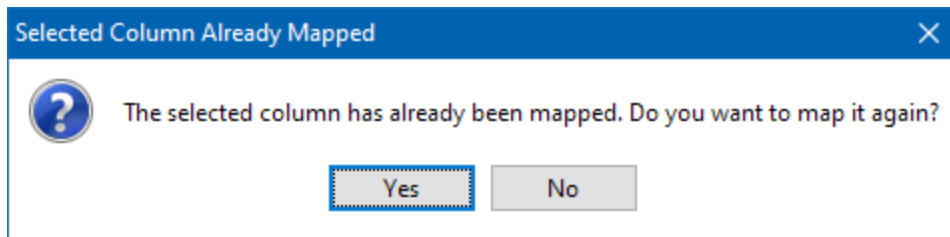
Transform

Generate Profile

A single import column contains four pieces of data that are to be stored separately in STEP. Using transformations, separate values can be created for the attributes. In this case it is: Primary Color, Hat Size, Material, and Country of Origin. For the first source row, the object should have Color = blue, Hat Size = 2, Material = Cotton, and Country of Origin = CN for China.

1. In the 'Map Data' step of an import wizard, map the object's ID. Then, map the column that holds the data to the four needed attributes. This means that the same column will be mapped 4 times. For more on how to map data, please see the **Import Manager - Map Data** topic or the **IIEP - Configure Processing Engine** topic in the **Data Exchange** documentation.

2. A warning message will display saying that the source column you are trying to map has already been mapped. Click 'Yes' to continue mapping to the needed attributes.



3. When finished with mapping, the Results field should look similar to the image below. Some columns may or may not be red depending on the attributes mapped and the validation base type for those attributes. Each column now needs to be transformed.

Map Data

Source:

<ID>	> (Data)
107601	blue 2 cotton China
20803	green 4 wool USA
20805	brown 3 wool USA
109011	pink 2 cotton UK

Result:

Map to: Product

ID=<ID> ✓	PrimaryColor=(Data) ✓	HatSize=(Data) ⚠	Material=(Data) ✓	CountryOfOrigin=(D... ⚠
107601	blue 2 cotton China	blue 2 cotton China	blue 2 cotton China	blue 2 cotton China
20803	green 4 wool USA	green 4 wool USA	green 4 wool USA	green 4 wool USA
20805	brown 3 wool USA	brown 3 wool USA	brown 3 wool USA	brown 3 wool USA
109011	pink 2 cotton UK	pink 2 cotton UK	pink 2 cotton UK	pink 2 cotton UK

Applying transformations

Below are examples of how transformations can be used for each of the columns to clarify the data.

For more on all available transformations, see the **Outbound Map Data - Transform** topic.

1. Select the first column that needs to be transformed, in this case it is 'PrimaryColor=(Data)', in the Results field, and click **Transform**.

Map Data

Source:

<ID>	>(Data)>
107601	blue 2 cotton China
20803	green 4 wool USA
20805	brown 3 wool USA
109011	pink 2 cotton UK

Result:

Map to: Product

ID=<ID> ✓	PrimaryColor=(Data) ✓	HatSize=(Data) ⚠	Material=(Data) ✓	CountryOfOrigin=(Data) ⚠
107601	blue 2 cotton China	blue 2 cotton China	blue 2 cotton China	blue 2 cotton China
20803	green 4 wool USA	green 4 wool USA	green 4 wool USA	green 4 wool USA
20805	brown 3 wool USA	brown 3 wool USA	brown 3 wool USA	brown 3 wool USA
109011	pink 2 cotton UK	pink 2 cotton UK	pink 2 cotton UK	pink 2 cotton UK

Auto Map

Map

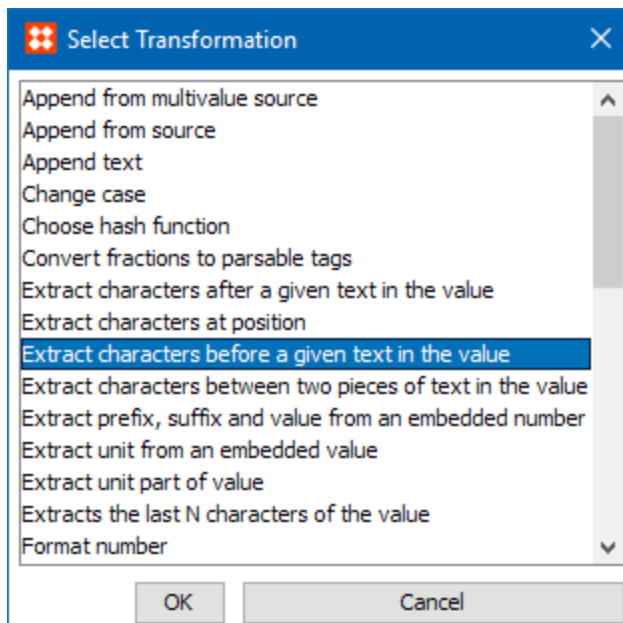
Constant

Remove

Transform

Generate Profile

- In the Transformation window, click on the blue **Add Transformation** link, to open the list of available transformations.



- Choose the needed transformation and enter in any necessary data. For example, two transformations were added for 'PrimaryColor=(Data)'. The first one, 'Extract characters before a given text in the value', shows up in the transformation table as 'Extract Characters before'. A space is put in the text field, though it is not visible

in the picture. Also, the 'Change case' transformation was added. 'Mixed' was chosen from the dropdown so that the first letter of the colors are capitalized and the rest appear lower case.

Specify the sequence of transformations in the table below.

Target:PrimaryColor

☐ Mandatory

Value - Source:(Data)

☒	Extract characters before	
☒	Change to	Mixed case

Add Transformation

Dimension Point

☒ Constant: (From Import Context)

☐ Source: (Data)

Add Transformation

Preview

PrimaryColor=(Data) ✓

Blue

Green

Brown

Pink

Save Reset Cancel

It is important to note that transitions are evaluated from the top down. Careful consideration needs to go into the order of the transformations applied. Each time a new transformation needs to be applied, **Add Transformation** needs to be clicked.

Note: If the 'Mandatory' option is checked, it will skip the empty values, and those objects will not be imported.

- Continue with transforming the next column. For example, in the 'HatSize=(Data)' column, the 'Split and extract' transformation can be selected to show the correct numbers. In the first field, a space, which can not be seen in the picture, was added. In the second field, the number 2 was selected from the dropdown menu. This is because the second grouping of data is what is desired to put into this field.

Specify the sequence of transformations in the table below.

Target: HatSize

☐ Mandatory

Value - Source: (Data)

Split by [] and extract field 2

Add Transformation

Preview

HatSize=(Data) ✓
2
4
3
2

Save Reset Cancel

5. In the third column, again select the 'Split and extract' transformation and select to extract field 3.
6. For the last column, select the 'Split and extract' transformation and select to extract field 4. Then select the 'Replace whole value with' transformation to change the countries to the needed abbreviations for STEP. Note that you will need to select this for each country type.

Specify the sequence of transformations in the table below.

Target: Material

☐ Mandatory

Value - Source: (Data)

Split by [] and extract field 4

Replace China with CN

Replace USA with US

Replace UK with GB

Add Transformation

Dimension Point

☒ Constant: (From Import Context)

☐ Source: <ID>

Add Transformation

Preview

Material=(Data) ✓
CN
US
US
GB

Save Reset Cancel

- Once each column is properly transformed, the results field should now look accurate, and it is ready to go on to the next steps for import.

Map Data

Source:

<ID>	> (Data)
107601	blue 2 cotton China
20803	green 4 wool USA
20805	brown 3 wool USA
109011	pink 2 cotton UK

Result:

Map to: Product

ID=<ID> ✓	PrimaryColor=((Data)) ✓	HatSize=((Data)) ✓	Material=((Data)) ✓	CountryOfOrigin=((Data)) ✓
107601	Blue	2	cotton	CN
20803	Green	4	wool	US
20805	Brown	3	wool	US
109011	Pink	2	cotton	GB

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Transform Date by Locale or by Pattern

Attributes with a Validation Base Type of Date, ISO Date, and ISO Date and Time can be transformed by locale (using the 'Transform date by locale' option) or by pattern (using the 'Transform date by pattern' option).

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

Transform date by locale

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

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

Transform date by pattern

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

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

Step-by-step instructions for export and import transformations are included below.

Considerations and Limitations

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

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

Export Transformations

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

1. Verify that the STEP Workbench Context is set correctly. This will impact locale transformations.
2. Choose a way to begin the export process using the wizard as described in **Creating a Data Export** and select the objects if necessary.
3. Select a format and continue to the Map Data step. Transformations are set up in Map Data step. If your selected format does not require mapping, this step is disabled. For details, see **Export Manager - Map Data**.
4. Begin by mapping the attributes to be exported. Pick the attributes from the tree and click the mapping icon.
5. Click the flipper to expand the options for the attribute you wish to transform. For 'Attribute N' values, you will see a Header row and a Value row.
6. Click the transformations button to the right of the Value row.
7. A Transformations dialog box appears. Change the Aspect field to 'Value' using the dropdown.
8. Click the 'Add Transformation' link and make a 'Transform date' selection: Transform date by locale / Transform date by pattern. Follow the directions for the applicable selection and then continue on to the next step in these directions.

Export Manager

Steps

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

Map Data

<ID>

<Name>

<Parent ID>

<Object Type Name>

<Product-Override Child ID>

<Is deleted>

"Constant Value"

<Page Number>

All Attributes

Select Attribute

Classifications

Index Words

Product Classification Links

Product References

Asset References

Classification References

Entity References

STEP Workflow Task Info

Multi level References

Multi level Parent attributes

Insert Referenced Objects

Custom Attributes

System Setup

Inherit Data and References

☒

Exports data in Excel format.

Column (3 mapped)

<ID> ID

<Name>

Attribute N Value

Header Attribute N Name

Value Attribute N Value

Transformations

Target: Value

Source: Attribute N Value

Aspect: Value

Add Transformation

Save

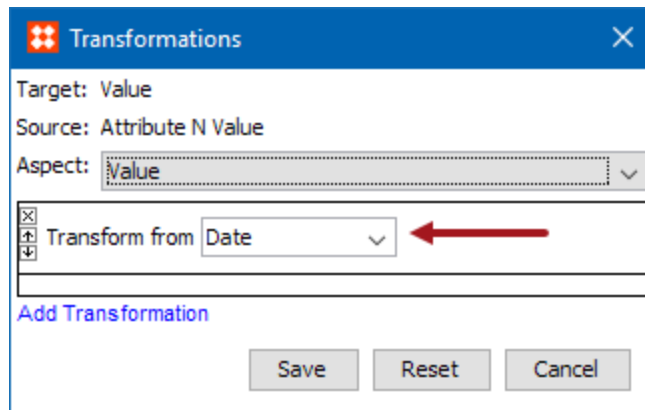
Reset

Cancel

Transform date by locale

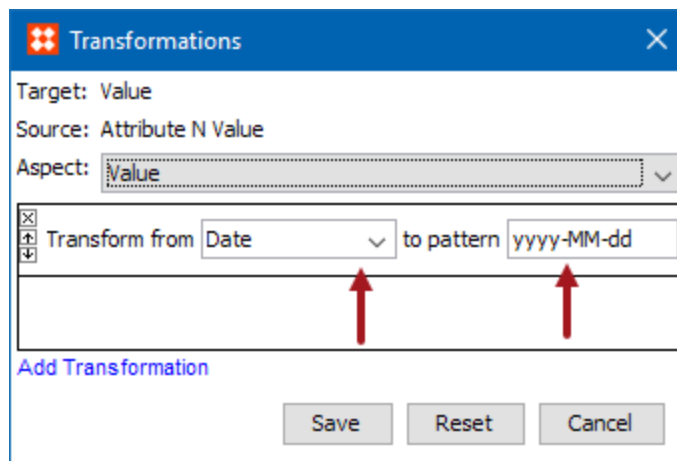
- Click on **Transform date by locale** and then click OK.
- From the Transformations screen, use the 'Transform from' dropdown to choose the validation base type of

the date attribute being exporting.



Transform date by pattern

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



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

9. Select Next to move to the Advanced step. For details, see **Export Manager - Advanced**. Make any additional selections, if desired. However, no additional configurations are required to complete the process, click Finish.
10. Optional: Make any Select Delivery Method selections and click Finish. For details, see **Export Manager - Select Delivery Method**.
11. Make selections on the Save Export Configuration screen, if desired. Click OK. For details, see **Running a Data Export**.

12. The 'Starting process Exporting' dialog will appear. For details, see **Monitoring a Data Export**. Choose 'Go to process' to access the file.
13. On the Background Process screen, determine if the export was successful or not.
 - If the process failed, go back and try the steps again (making sure the correct attribute validation base type was selected on the Transformations screen).
 - If the process was successful and the delivery method was set to file, an export file (expand the Result flipper, if needed) will be available at the bottom of the Background Process page. Double-click the file name to open the file, or click the Save icon to the right of the file name to save the file without immediately opening it.
14. Verify the exported data is correct.
 - If transforming by locale, the correct locale format should be showing. The locale is determined by the context set in the workbench at the time the export process begins.
 - If doing a pattern transformation, verify the date format matches the pattern entered. If the date pattern includes a time zone, the time zone is determined by the location of your server. It is not impacted by context set in the workbench.

Import Transformations

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

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

1. Choose a way to begin the import process using the wizard as described in **Creating a Data Import**.
2. In the Select Data Source step, link the file you are importing by searching / selecting or manually typing in (or pasting) the file location. For details, see **Import Manager - Select Data Source**.
3. Make a format selection in the Select Format step. For details, see **Import Manager - Select Format**.
4. Transformations are set up in Map Data step. If your selected format does not require mapping, this step is disabled. For details, see **Import Manager - Map Data**.
5. Begin data mapping by selecting Auto Map. If the date column is unmapped, then use Map to manually tie the attribute values to be imported to an existing attribute.
6. Under Result, select the column you want to transform during the import process. While the column is selected, click on the Transform button.

Import Manager

Steps

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

Map Data

Source:

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

Result:

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

Map to: Product

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

Navigation: Back, Next, Finish, Cancel

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

Specify the sequence of transformations in the table below.

Target:Date

☐ Mandatory

Value - Source:Manufacturing Expiration Date

Transform to: Date

Add Transformation

Preview

Date=Manufacturing Expiration Date ✓

15-Oct-2015

Buttons: Save, Reset, Cancel

Specify the sequence of transformations in the table below.

Target:Date

☐ Mandatory

Value - Source:Manufacturing Expiration Date

Transform from pattern	yyyy-MM-dd	to	Date

[Add Transformation](#)

Preview

Date=Manufacturing Expiration Date ✓

06-Apr-0021

Save Reset Cancel

Transform date by locale

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

Value - Source:Manufacturer Warranty 90-Day Notice

Transform to Date

Transform date by pattern

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

Value - Source:Manufacturer Warranty 90-Day Notice

Transform from pattern yyyy-MM-dd to Date

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

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

Transform Using Attribute Transformation Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

Using the '[Transform Using Attribute Transformation]' transformation, it is possible to apply a set of transformations specified in an attribute transformation to a value. The transformation takes a predefined attribute transformation as argument. This can be treated as a convenient setup to avoid duplication of such transformations, and it is easy to maintain once created. [Transform Using Attribute Transformation] are defined in System Setup, and are predominantly used for print and tables.

For example, a user can apply the required transformations.

The screenshot displays the 'System Setup' window on the left and the 'Attr_Trans - Attribute Transformation' configuration window on the right.

System Setup (Left Panel):

- Attribute Groups
- Attribute Transformations
 - Transformation
 - Attr_Trans (selected)
 - Commercial Price
 - Fraction
 - New Item
 - Replace Text
- Action Sets
- Contexts
- InDesign Queue
- Lists of Values / LOVs

Attr_Trans - Attribute Transformation (Right Panel):

Attribute Transformation

Name	Value
ID	Attr_Trans
Name	Attr_Trans

Transformations

- Append New Product
- Change to Upper case
- Replace whole value using Lookup Table: <up Table 1 (TransformationLookupTable1) ...

[Add Transformation](#)

Then select the attribute transformation to apply the full set of transformations during a data import.



Specify the sequence of transformations in the table below.

Target: ShortItemDescription

☐ Mandatory

Value - Source: <ShortItemDescription[DK]>

×

⬆

⬇

⬆

Attribute Transformation

Attr_Trans (Attr_Trans)

⋮

Add Transformation

Dimension Point

☒ Constant: (From Import Context) ▼

☐ Source: <ID> ▼

Add Transformation

Preview

ShortItemDescription = <ShortItemDescription[DK]> ✓
Lightweight and durable
Carbon Case
200 voltz
20 volt light pink flashlight
Lightweight and durable

<

>

Save

Reset

Cancel

Transformation Lookup Table Example

Transformations can improve the quality of the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

Transformation Lookup Tables can be used to transform words and entire values when importing tabular-based data. For information on lookup tables, see the **Transformation Lookup Tables** topic in **System Setup / Super User Guide** documentation.

Note: Transformations that use Lookup Tables can be saved as an Import Manager configuration. To use a saved Lookup Table in a configuration, the referenced Transformation Lookup Table must not be deleted and must have content in the context of the import.

For example, it is possible to replace all occurrences of a word with another word before data is imported into an attribute. Using a Transformation Lookup Table, you can add a 'Replace words using Lookup Table' transformation on the result column, and select the relevant Lookup Table. For this case, all instances of 'colour' are being changed to 'color' by using this transformation table.

<
▶
Lookup Terms rev.0.1 - Transformation Lookup Table

Transformation Lookup Table	
Description	
Name >>	Value >
> ID	LKUPTerms
> Name	Lookup Terms
> Object Type	Transformation Lookup Table
> Revision	0.1 Last edited by USERJ on Fri Sep 29 10:24:33 EDT 2017
> Approved	✖ Never Been Approved
> Translation	Not Translated
> Path	Classification 1 root/Classifications/AOT Office Documents/Lookup Terms

Lookup Table

☐ Replace with default value when no matches are found (Value Substitution only):

☐ Ignore Case

From >	To >
> Colour	color
> colours	colors
> flavour	flavor
> flavors	flavors
> Add Row	

4 Rows

Import From Clipboard

Apply

1. In the **Preview**, before any transformations are added to the Results field, the data looks like the table below:

Map Data

Source:

<ID>	>	<Name>	>
114852		Blue colour Light	
114854		Green colour Flashlight	
111204		LED Flashlight	
114859		Pink light colour	
114440		Red light	
888264			

Result:

Map to: Product

ID = <ID> ✓	Name = <Name> ✓
114852	Blue colour Light
114854	Green colour Flashlight
111204	LED Flashlight
114859	Pink light colour
114440	Red light
888264	

Auto Map

Map

Constant

Remove

Transform

Generate Profile

2. Select the 'Name' field in the Source table, then click **Transform**. Click the **Add Transformation** link and select the 'Replace words using Lookup Table' option, and choose the needed transformation table.

Specify the sequence of transformations in the table below.

Target: Name

☐ Mandatory

Value - Source: <Name>

Replace words using Lookup Table:

[Add Transformation](#)

Dimension Point

☒ Constant: (From Import Context)

☐ Source: <ID>

[Add Transformation](#)

Preview

Name = <Name> ✓
Blue colour Light
Green colour Flashlight
LED Flashlight
Pink light colour
Red light

Map to: Product

Name = <Name> ✓
Blue colour Light
Green colour Flashlight
LED Flashlight
Pink light colour
Red light

- When properly added, the transformation changes the input text to the proper format.


×

Specify the sequence of transformations in the table below.

Target:Name

☐ Mandatory

Value - Source: <Name>

×

⬆

⬇

⬆

⬇

Replace words using Lookup Table: <up Table 1 (TranformationLookupTable 1) ...

Add Transformation

Dimension Point

☒ Constant: (From Import Context)
 ☐ Source: <ID>

Add Transformation

Preview

Name = <Name> ✓
Blue color Light
Green color Flashlight
LED Flashlight
Pink light color
Red light

Save

Reset

Cancel

Note: When whole values are replaced with multi-valued attribute data, the replacement works on the separate entries in the multi-value when added in the Value section. When added in the Before Multivalue Split section it works on the entire string from the import file including the multi-value separators.

- Click **Save** to exit the Transformations dialog and update the Map Data step contents.

Map Data

Source:

<ID>	> <Name>
114852	Blue colour Light
114854	Green colour Flashlight
111204	LED Flashlight
114859	Pink light colour
114440	Red light
888264	

Result:

Map to: Product

ID=<ID> ✓	Name=(<Name>) ✓
114852	Blue color Light
114854	Green color Flashlight
111204	LED Flashlight
114859	Pink light color
114440	Red light
888264	

Auto ...

Map

Constant

Remove

Transform

Generate Profile

Export Manager

The Export Manager allows you to export data on demand or at scheduled intervals and to save or modify an export configuration. The final step of the wizard allows you to save the export configuration. This is useful for frequently-used exports where the parameters and options are always, or nearly always, the same. A saved export configuration saves setup time, and is required for scheduled exports.

Setup Requirements

Setting up and using the Export Manager involves the following steps:

1. Based on the data being exported, determine how to start the Export Manager wizard as described in **Creating a Data Export**.
2. In the wizard, choose a previously saved configuration or start a new export as described in **Export Manager - Select Configuration**.
3. In the wizard, select one or more hierarchy nodes or modify the nodes selected via a configuration as described in **Export Manager - Select Objects**.
4. In the wizard, select the file format in which the extracted information is delivered as described in **Export Manager - Select Format**.
5. In the wizard, when required, select which pieces of information are extracted per object exported as described in **Export Manager - Map Data**.
6. In the wizard, when required, specify if exported data should be converted to another output format as described in **Export Manager - Advanced**.
7. In the wizard, specify how information is to be delivered as described in **Export Manager - Select Delivery Method**.
8. In the wizard, save the configuration (if necessary), and run the export as described in **Running a Data Export**.
9. Monitor the export and download the output file as described in **Monitoring a Data Export**.

Additional Information

The following information is useful once an export configuration is saved:

1. Maintain a saved configuration as described in **Maintaining a Saved Export Configuration**.
2. Perform the selected export configuration one or more times, according to the scheduling parameters, via a background process as described in **Scheduling a Data Export**.
3. Remove a scheduled export process, which is required to run current parameters if the original configuration is modified, as described in **Deleting a Scheduled Data Export**.

Creating a Data Export

The Export Manager wizard allows you to export data on demand and to create or modify an export configuration. The final step of the wizard allows you to save the export configuration. This is useful for frequently-used exports where the parameters and options are always, or nearly always, the same.

Export data using the Export Manager

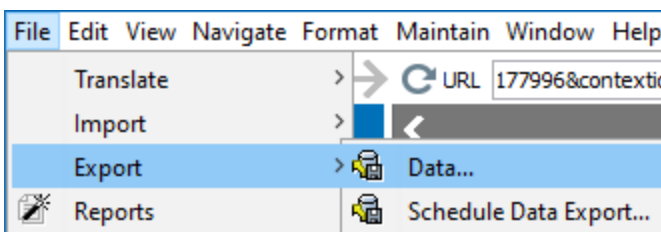
1. Select the appropriate STEP context. All data is exported using the currently selected context and the current workspace, typically Main.

Note: When exporting products, see the **Considerations for Exporting Products** section below.

2. Use one of the following methods to launch the Export Manager wizard, noting that method of starting the wizard results in exporting different objects. Regardless of the method selected, the same wizard is launched.

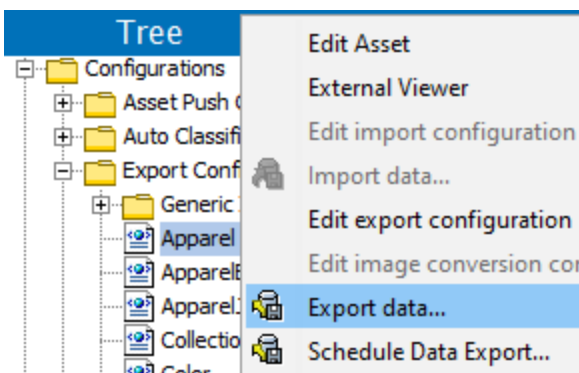
- **From the File Menu:** Click the File menu > Export > **Data**.

The wizard opens in Step 1 and allows you to select an existing configuration or create a new one.



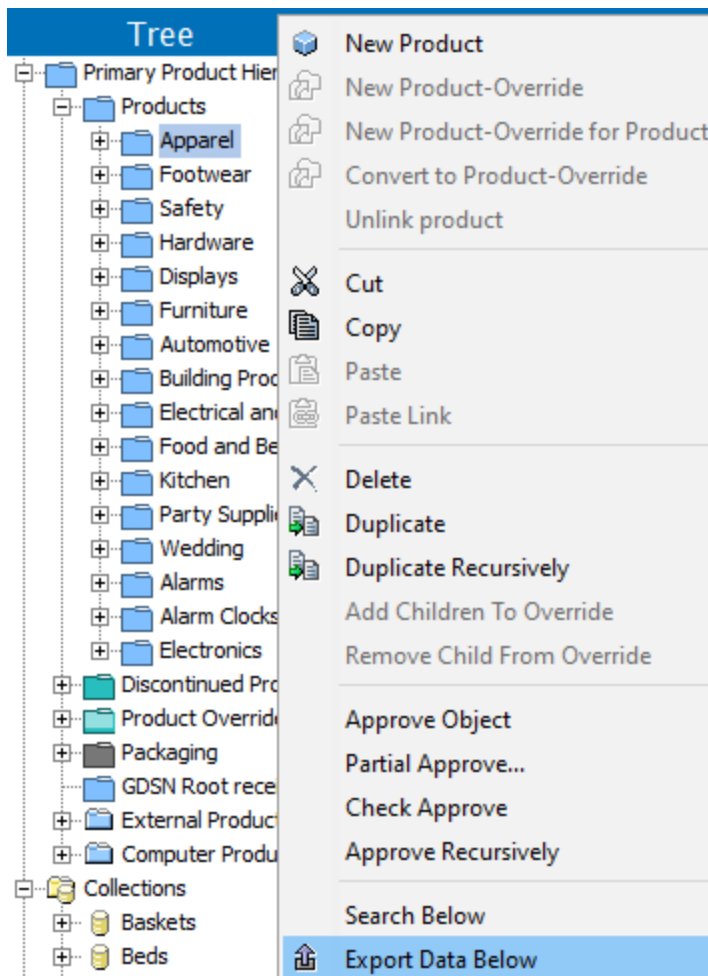
- **From an Export Configuration:** On the Tree, select an export configuration, right-click, and then select **Export Data**.

The wizard opens in Step 2 with the configuration and objects included in the configuration already selected.



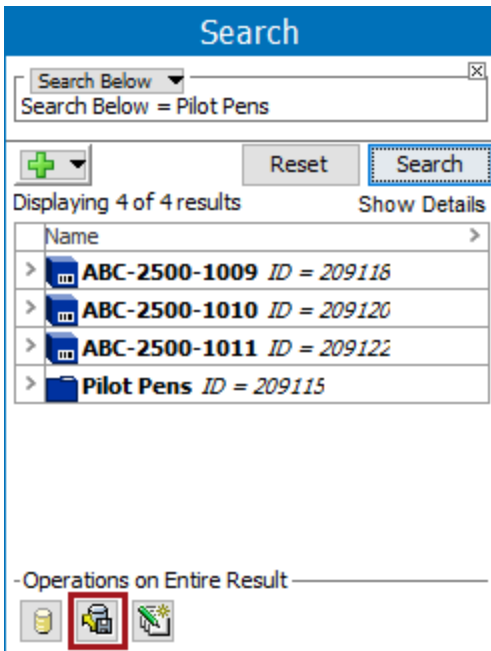
- **From an Object:** On the Tree, select a classification, product group, publication, or entity, right-click, and then select **Export Data Below**.

The wizard opens in Step 2 with the objects below the selection displayed. This selection is non-inclusive, meaning that the selected object is not exported. For example, when selecting a product node in the primary product hierarchy, the selected node is not exported, but the children are exported. When selecting one or more individual products that have no children, only the selected products are output.



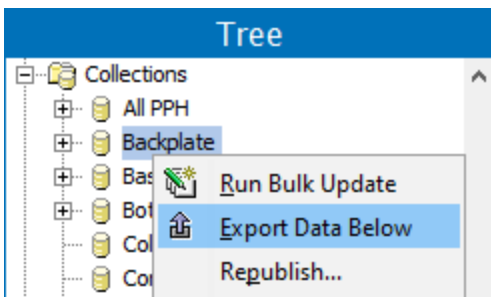
- **From a Search Operation:** On the Search tab, perform a search with results displayed. In the **Operations on Entire Result** section, click the export button.

The wizard opens in Step 2 with 'Product selection made by search URL' displayed.



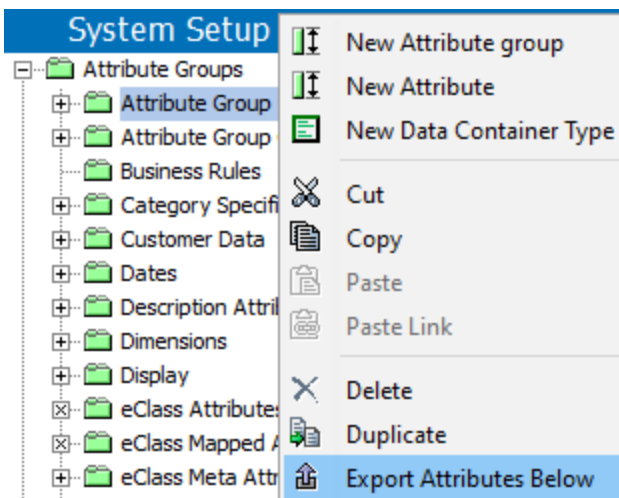
- **From a Collection:** On the Tree, select a collection, right-click, and then select **Export Data**. All object in the collection are included in the export.

The wizard opens in Step 3 ready to choose a format.

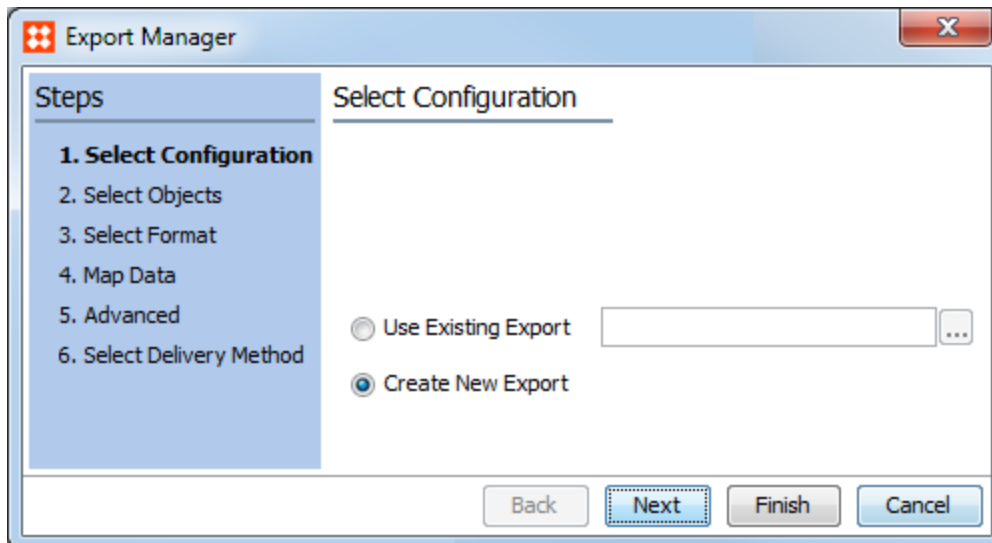


- **From an Attribute Group:** On System Setup, select an attribute group, right-click and then select **Export Attributes Below**.

The wizard opens in Step 2 with the attributes below the selection displayed.



3. After selecting one of the above ways to export, the Export Manager wizard displays and can involve the following steps:



- **Select Configuration** allows you to select a previously stored configuration or start a new export.
- **Select Objects** allows you to select one or more hierarchy nodes or modify the nodes selected via a configuration.
- **Select Format** allows you to select the file format in which the extracted information is delivered.
- **Map Data** allows you to select which pieces of information are extracted per object exported.
- **Advanced** allows you to specify if exported data should be converted to another output format.
- **Select Delivery Method** allows you to specify how information is to be delivered.

Considerations for Exporting Products

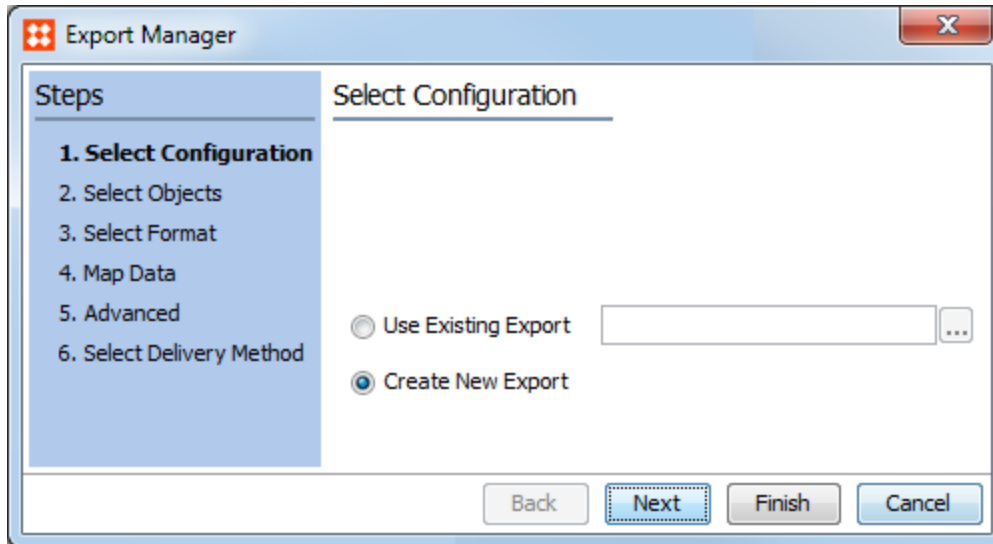
When exporting products, it is common setup to first review exactly the products and attributes that you want to extract. It is important to choose the exported attributes carefully since there is no distinction in the output between a missing (or blank) attribute value and a value that is empty because the attribute is not valid for a product. Including invalid attributes in an export results in output with blank attribute data and in many cases, the cause of the missing data is not apparent.

Determine the following information about your required export enables you to accurately set the wizard parameters:

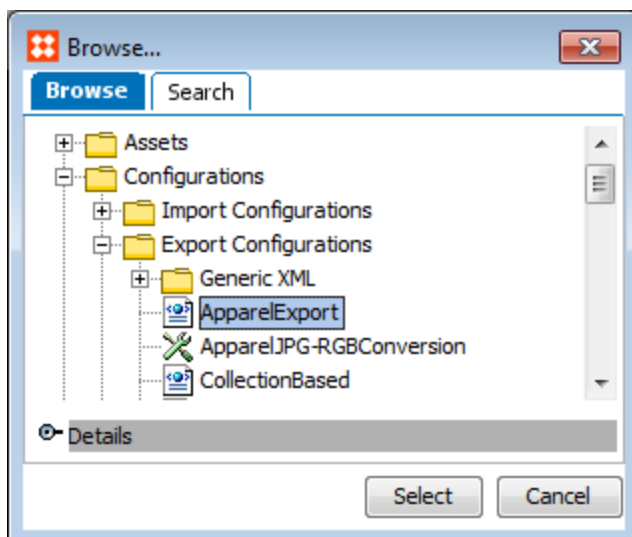
- If the attributes you want to extract are global, and for example, relevant for nearly all products, you can select products from disparate product lines, since the attributes would be valid for all those products.
- If the product lines you want to extract do not have common attributes, it can be useful to extract multiple files with separate sets of attributes.
- If your goal is to create an XLS file that manufacturers can return to you with missing data, extract the product lines, including the manufacturer's name as an attribute, and then split the extracted file into separate files for each manufacturer. In this way, the attributes are valid for the products in each file.
- If you know the attributes you want to extract before you launch the Export Wizard, it is easier to select the appropriate attributes on the Data Map step.

After completing the wizard steps, save the configuration so you can modify it during testing. Initially, use a small group of products to test your export settings. Refine the wizard parameters to resolve any issues and save the updated configuration.

Export Manager - Select Configuration

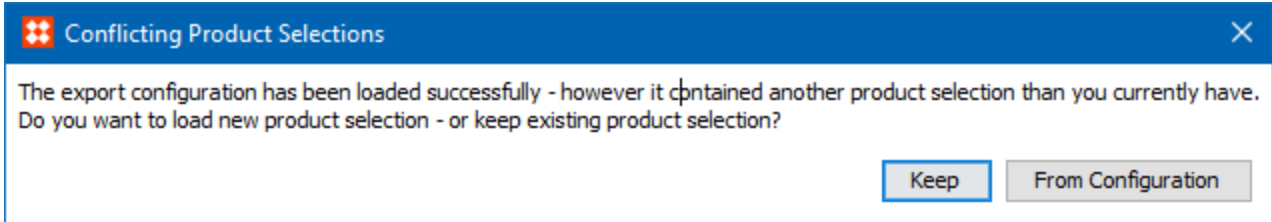


1. Select a radio button to use an existing export configuration, or create a new one. The following options are available when using an existing configuration:
 - To use an export configuration previously saved, select the **Use Existing Export** radio button, click the ellipsis button (...), and either Browse or Search to locate the configuration in STEP. For more information on saved configurations, see the **Maintaining a Saved Export Configuration** topic.



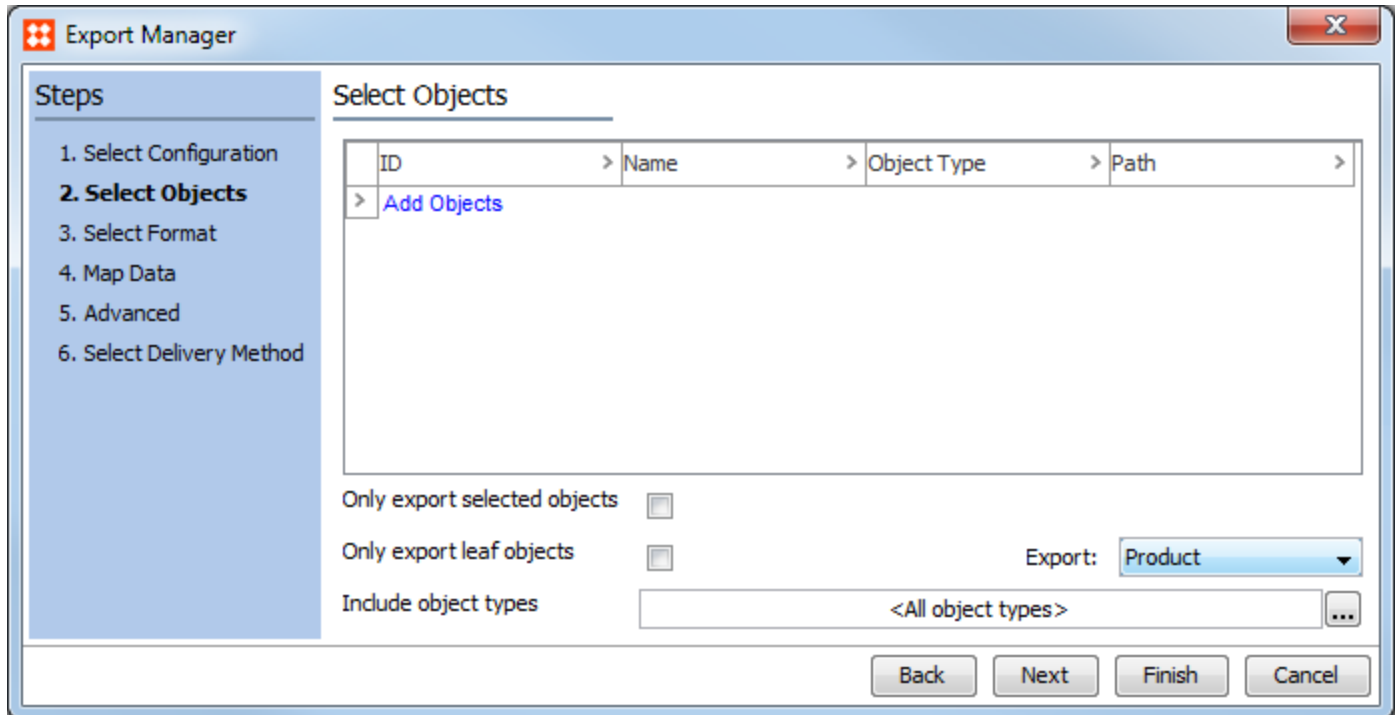
- To create a new export configuration based on an existing export configuration, select the **Use Existing Export** radio button, click the ellipsis button (...), and either Browse or Search to locate the existing configuration, use the other wizard steps to change the configuration to meet your current requirements, and then save the modified configuration in the final step of the wizard.

- When returning to the Select Configuration step after making a selection on the Select Objects step, select **Use Existing Export** to apply a saved configuration. In this scenario, if the objects selected differ from those in the saved configuration, the Conflicting Product Selections dialog is presented. Click **Keep** to continue with the manually selected objects, or click **From Configuration** to use the object selection in the saved configuration.



2. Click **Next** to display **Export Manager - Select Objects**, or click **Finish** to use an existing configuration without changes, and display the Save Export Configuration window as defined in **Running a Data Export**.

Export Manager - Select Objects



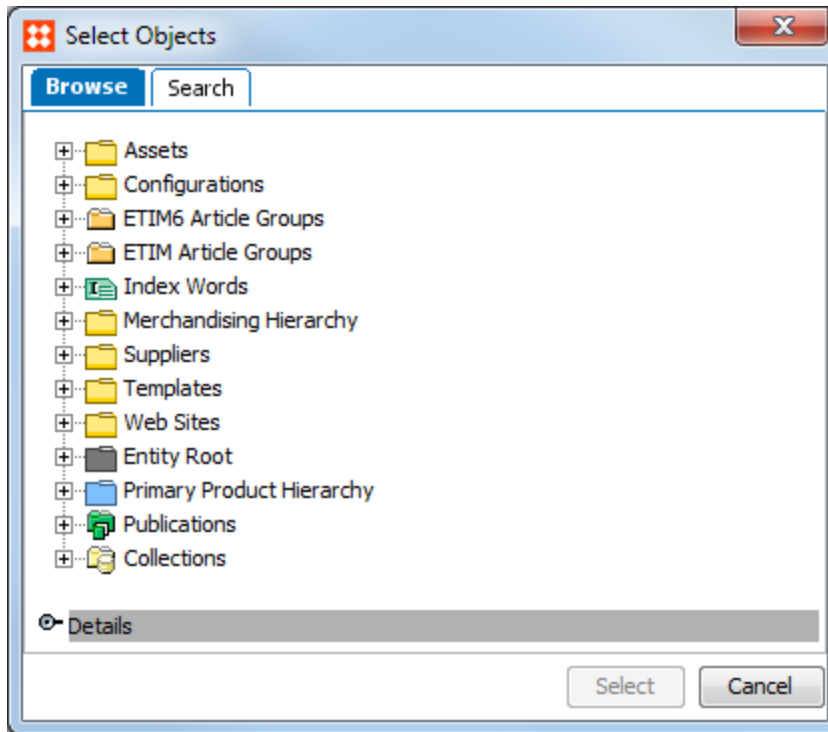
1. In the **Export** dropdown, select the object super type to export and allow the next step of the wizard to show only legal export formats.
 - **All** - all objects of the selected node are exported. Most formats are available.
 - **Asset** - only metadata on assets is exported from the selected node, export of the actual digital asset (asset content) is not determined by this selection. Most formats are available. For information on asset content exports, see **Exporting Assets** in the **Digital Asset Exchange** documentation.
 - **Attribute** - all attribute objects are exported for the selected attribute group. Most formats are available.
 - **Change Package** - only sealed change package objects are exported. Advanced STEPXML format is available with default template.
 - **Classification** - only classification objects are exported from the selected classification hierarchy. Most formats are available.
 - **Entity** - only entity objects are exported from the selected node. Most formats are available.
 - **List of Values** - only LOV objects are exported. Advanced STEPXML (with default template), STEPXML (with default parameter settings), and Generic XML formats are available.
 - **Product** - only product objects are exported from the selected product hierarchy. All formats are available.
 - **Publication Objects** - all publication-related objects are exported. When a publication group object is selected, this option defaults to Publication Objects and the format defaults to STEPXML, allowing the publication-related parameters to be set. The Publication Excel or Flatplan Excel formats are available.

- **STEP Workflow** - all available workflows are exported unless individual workflows are selected. Advanced STEPXML (with default template) and STEPXML (with default parameter settings) formats are available.

For example, when adding objects, you could select a classification hierarchy node, then choose to export products, which results in exporting all products linked in below the selected classification.

Note: Selecting an option from the Export dropdown impacts formats that require mapping. When the Map Data step is disabled, for example with STEPXML, this selection has no effect.

2. Click the **Add Objects** link to select objects, or leave the Select Object tables empty and export based on the other wizard parameters. If you started the export from a node using Export Below, the objects are selected automatically.
If exporting an Excel Smartsheet, see the **Selecting Objects for Excel Smartsheet Export** section below.
3. In the Select Objects window, use the **Browse** or **Search** tab to locate and select objects for export. The available objects displayed in the Select Objects window are determined by selection you made in the Export dropdown.
 - To export specific branches of the product, entity, collection, or classification hierarchy, select the root node for the export.
 - Multiple objects can be selected at the same time.
 - The selected root node object and any objects below will be exported, if allowed by other parameters in the wizard.

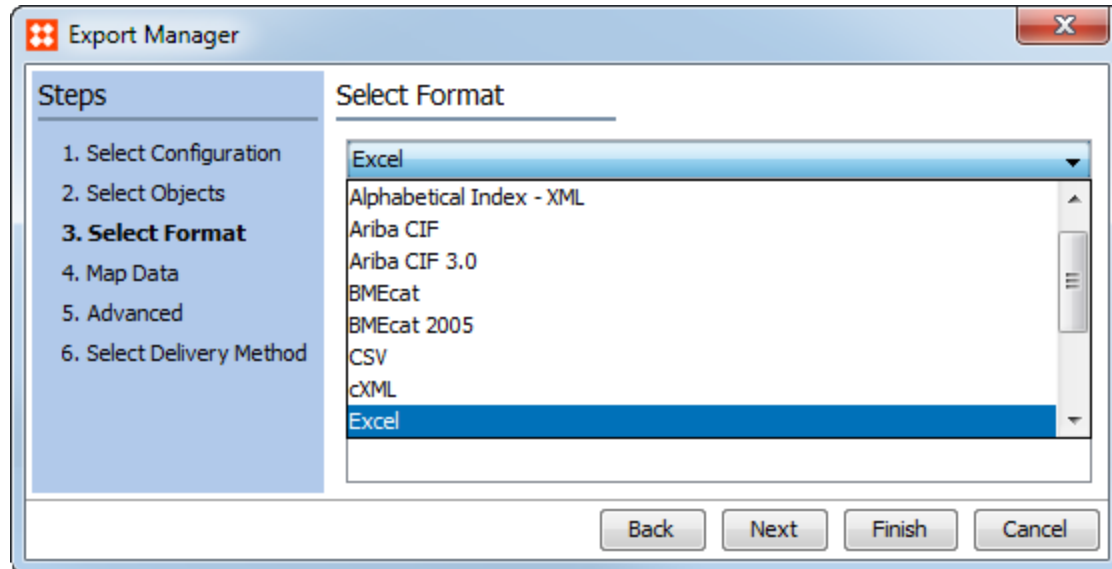


4. Click **Select**. The selected nodes appear in the **Select Objects** window.
5. Select **Only export selected objects** to export only the root nodes, and not the objects that live below the selected node. This option is only available when 'Only export leaf objects' is not selected. If available and unchecked, objects and their children are exported.
6. Select **Only export leaf objects** if you have selected a top hierarchy of products to be exported, but only want the lowest level on the Tree to be included in the export file. Selected objects are only included in the export if they have no children. This option is only available when 'Only export selected objects' is not selected. If available and unchecked, objects and their children are exported.

Note: This option is ignored when exporting STEPXML format.

7. The **Include object types** parameter is used to limit the export to objects of a specific Object Type. To modify the parameter, click the ellipsis button (...) and select the object types to be exported. Leave this field blank to include all object types.
8. Click **Next** to display **Export Manager - Select Format**.

Export Manager - Select Format

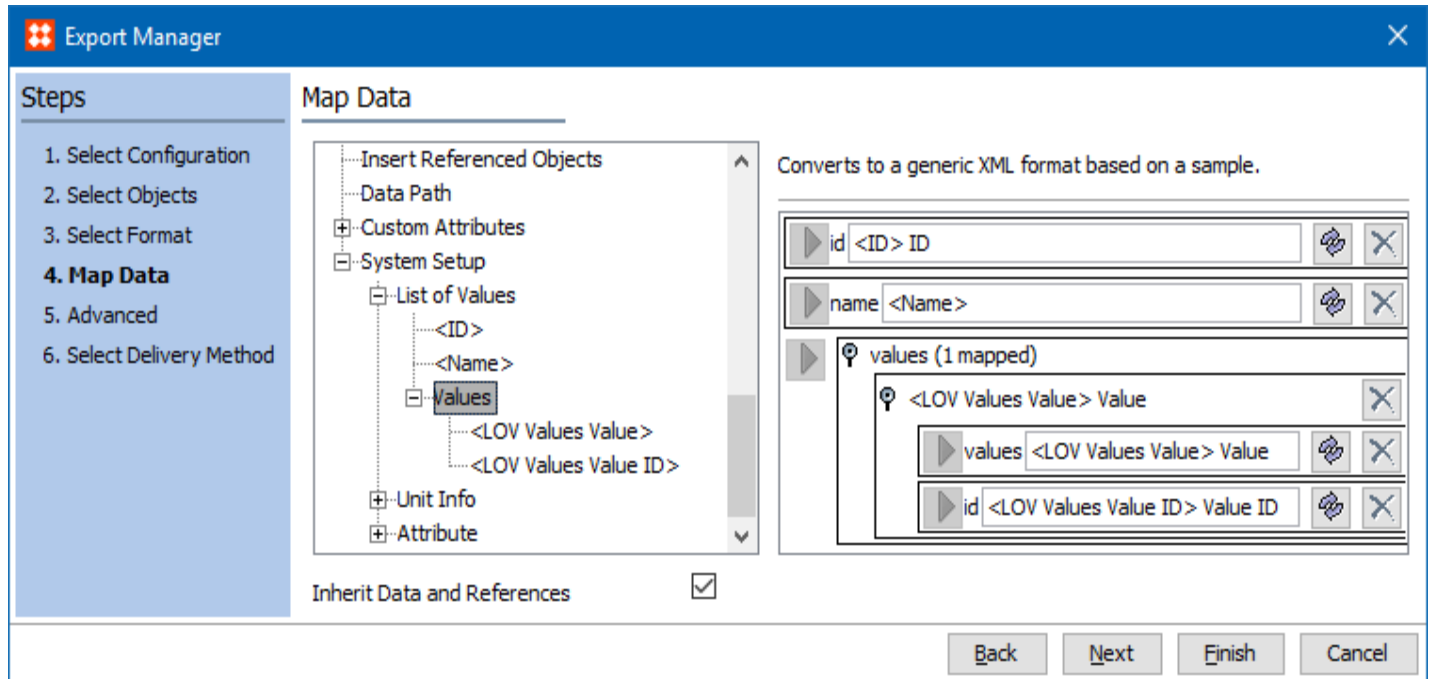


The selected option determines the file format of the extracted data.

- Advanced STEPXML Format
- Alphabetical Index - XML Format
- Ariba CIF Format
- Ariba CIF 3.0 Format
- BMEcat Format
- BMEcat 2005 Format
- CSV Format
- cXML Format
- Excel Format
- Excel Custom Template
- Excel Smartsheet Format
- Flatplanner Format
- Generic XML Format
- IDoc MATMAS05 Format
- Publication Excel Format
- STEPXML Format
- xCBL Format

Click the **Next** button to display **Export Manager - Map Data** when required by the format, or **Export Manager - Advanced** if mapping is not required.

Export Manager - Map Data



The options available for mapping are the same for all formats except Generic XML, which includes two extra data sources. Outbound mapping is discussed in detail in the **Outbound Map Data Options** documentation.

Generally, you will create mapping rules by selecting a data source (which indicates what you want to extract for each record) and linking that to a mapping target (which indicates where it belongs in the output).

For details on the available data sources (left pane), see the **Outbound Map Data - Data Source** documentation.

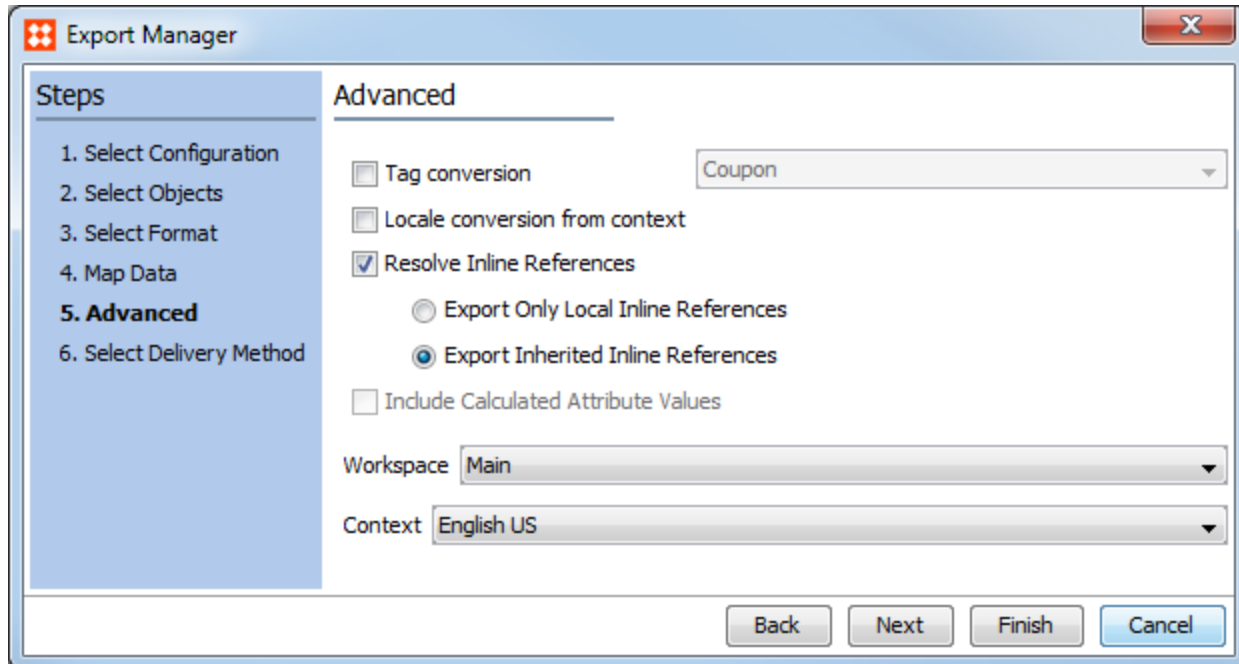
For details on mapping targets (right pane), see the **Outbound Map Data - Mapping Targets** documentation.

For details on transformations available using the  button, see the **Outbound Map Data - Transform** documentation.

For details on inheritance, see **Inherit Data and References - Data Source Outbound** documentation.

Click the **Next** button to display **Export Manager - Advanced**.

Export Manager - Advanced



Set the following additional export parameters as needed:

- **Tag conversion:** Optional. Allows for STEP Tags to be converted to match the selected output tag format.

For example, the STEP bold tag could be converted to the HTML equivalent. Tags and Tag Conversions are configured under Tags in System Setup. The selection in the dropdown determines the output formatting. For more details see the **Tags** topic, and for the available tag conversion formats, see the **Configuring Output Formatting** topic in the **System Setup / Super User Guide** documentation.

- **Locale conversion from context:** Optional. When checked, converts dates and numbers to comply with the selected locale for the Context selected before starting the Export Manager wizard according to the locale rules associated with the context.

Important: Selecting **Locale conversion from context** when exporting a Smartsheet can break the automatic conversion to the client's native decimal separator, causing the import of the Smartsheet to fail.

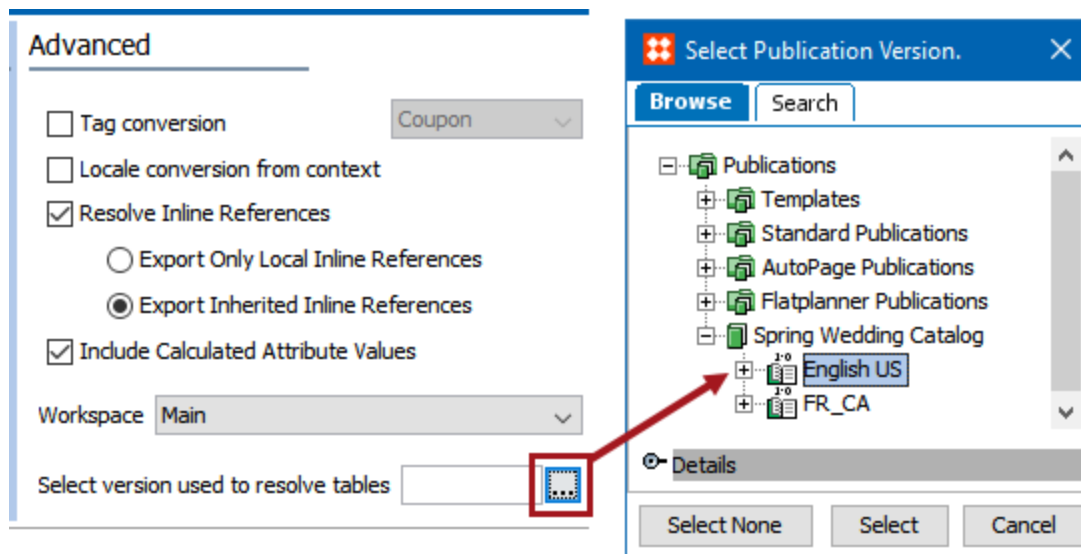
- **Resolve Inline References:** Determines how inline references are resolved when exported. By default, this option is checked, which means that if inline references are included in the data, they will be resolved (rather than tagged) in the export. If not checked, inline references are exported with the inline references tagging. Once checked, choose one of the following options:
 - Export Only Local Inline References means only local references are resolved and exported.
 - Export Inherited Inline References means references inherited from a parent are resolved and exported.
- **Include Calculated Attribute Values:** This option is only enabled (and is checked by default) when calculated attributes have been mapped. When checked, calculated attribute values are resolved upon export.

If not checked, calculated attribute values are exported with empty values, unless the Force Calculation option has been set. The Force Calculation option is available when mapping a calculated attribute using the Select Attribute data source, and also as a transformation. For more information, see **Force Calculation** in **Attribute - Data Source Outbound** and **Outbound Map Data - Transform**.

The value template is exported for each selected context, including a qualifier ID, which makes it possible to import the same data back into STEP. For information about calculated attribute values, see the **Calculated Attributes** topic in **System Setup / Super User Guide** documentation.

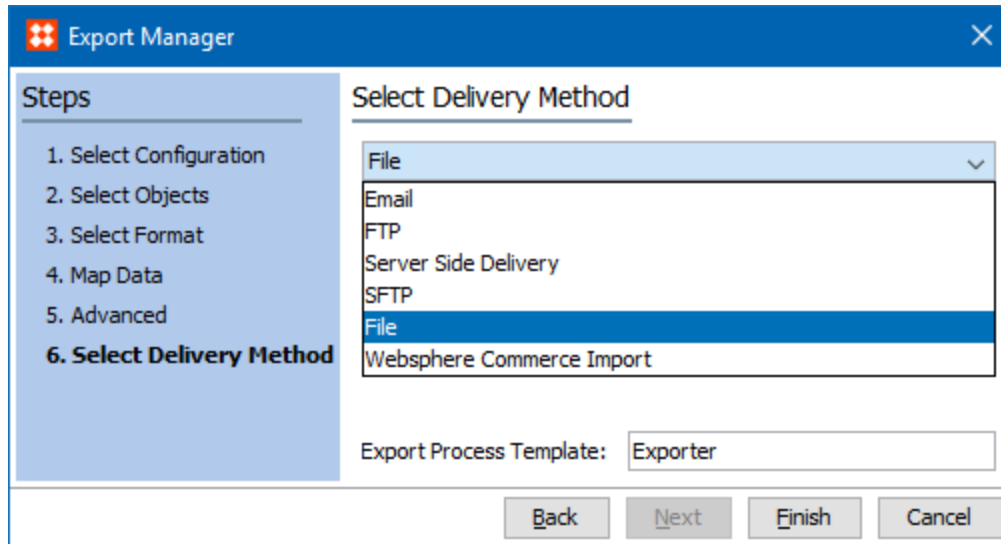
Important: If many complex calculated attribute values (traversing hierarchies and/or references) are used, consider if they should be exported, since they can impact performance. If required, consider scheduling for non-peak times. Simple calculations are not detrimental to an export, regardless of the quantity.

- **Workspace:** Allows the export to run from a different Workspace than the one selected when the Export was started. Only a single workspace can be selected for export.
- **Context:** Allows the export to run from a different Context than the one selected when the Export was started. For all formats except STEPXML, only a single context can be selected for export. For information about exporting multiple contexts with STEPXML, see the **Export Data for Selected Contexts** section of the **STEPXML Outbound Parameters** documentation.
- **Select version used to resolve tables:** This setting is only available if the format is STEPXML and 'Include Tables' is set to Yes. A publication version should be selected if the tables contain content relevant to a particular publication (such as column or row types that are only valid for certain publication types) or a publication version (such as commercial data). If a publication version is not selected, the version last used in the STEP workbench is used. For more information, see the **Exporting Resolved Tables** topic in the **Tables** documentation.



Click the **Next** button to display **Export Manager - Select Delivery Method**.

Export Manager - Select Delivery Method



Delivery methods are used by both outbound data tools but the available options vary. For information on the delivery methods available in OIEPs, see **OIEP Delivery Methods**.

1. In the **File Delivery** list, select one of the following export options:
 - **Email** delivers files as an email attachment and allows a zipped delivery.
 - **File** delivers a file via a background process and notifies when the file is ready for download.
 - **FTP** delivers files using file transfer protocol (FTP).
 - **Server Side Delivery** exports a file to a folder located on the STEP system application server.
 - **JDBC** exports data to a Relational database management system (RDBMS).
 - **SFTP** delivers a file using the Secure File Transfer Protocol (SFTP).
 - **Websphere Commerce Import** delivers a STEPXML file to a Websphere Commerce server.
2. For the **Export Process Template** parameter, leave the default setting, which is required to start the background process for the export.
3. Click **Finish** to display the Save Export Configuration window as defined in **Running a Data Export**.

Email Delivery Method

The email delivery option is very straight forward and, like 'Copy to directory', it allows the delivery to be zipped. It is typically used to test the export output, or to deliver a small export to one or more persons as a one-time occurrence.

This delivery method is available in Export Manager and in OIEPs. For information on a parameter, hover over the parameter to display help text.

Prerequisites

To deliver an exported file via email, the STEP server must have access to the SMTP server. Be aware of email file size limits, since depending on export contents, the file could be very large.

Configuration

For information on a parameter, hover over the parameter label to display help text.

Export Manager

Steps

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

Select Delivery Method

Email

Delivers exported data in an email.

Email address: inboundDataSource@customer.com

Subject text: Product data export

Body text: Please find enclosed the latest product data delivery

Zip export file: No

Export Process Template: Exporter

Back Next Finish Cancel

1. Click the **Select Delivery Method** parameter to display the dropdown and choose **Email**.
2. For the **Email** parameter, enter an email address.
3. For the **Subject text** parameter, enter the text that will display for the email subject line.
4. For the **Body text** parameter, enter the text that will display for the email body.
5. In **Zip export file**, select an option from the dropdown to specify if the output file should be delivered as an email attachment in a .ZIP (compressed) file format.
 - **Yes** zips the export file.
 - **No** does not zip the export file.

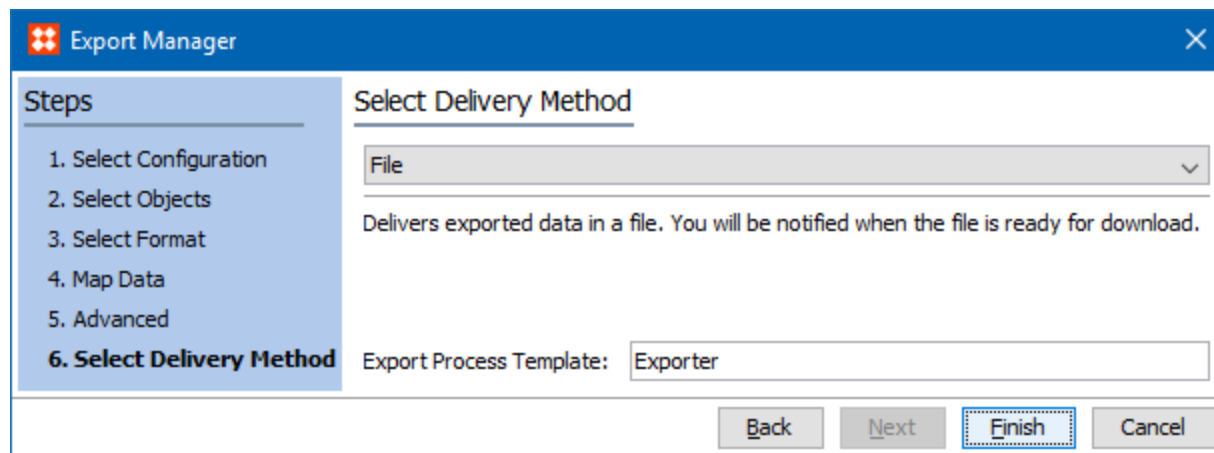
6. For the **Export Process Template** parameter, leave the default setting, which is required to start the background process for the export.
7. Click **Finish** to display the Save Export Configuration window as defined in **Running a Data Export**.

File Delivery Method

The file delivery option is only available in Export Manager. It makes the output available within STEP, via a background process.

Configuration

For information on a parameter, hover over the parameter label to display help text.



Export Manager

Steps

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

Select Delivery Method

File

Delivers exported data in a file. You will be notified when the file is ready for download.

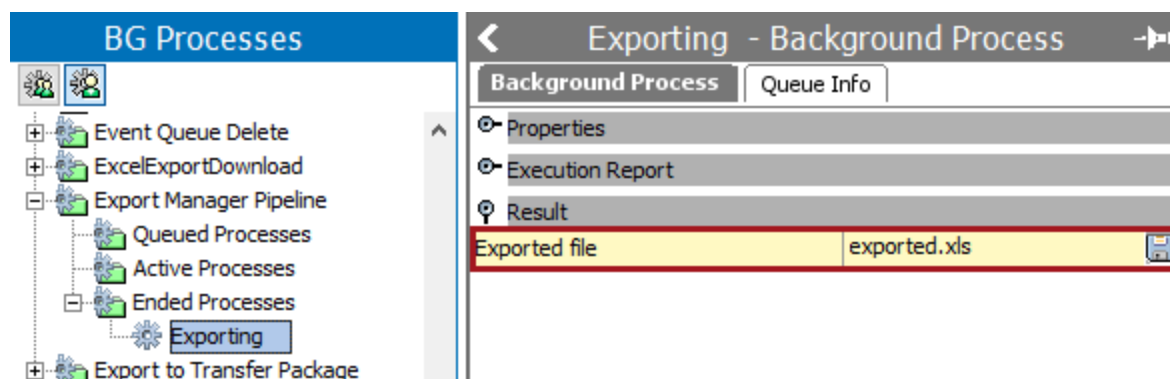
Export Process Template: Exporter

Back Next **Finish** Cancel

1. Select **File** from the dropdown.
2. For the **Export Process Template** parameter, leave the default setting, which is required to start the background process for the export.
3. Click **Finish** to display the Save Export Configuration window as defined in **Running a Data Export**.

Output

By default, the output is saved on the Background Processes tab, under the Export Manager Pipeline node, which can be easily accessed as described in **Monitoring a Data Export**.



BG Processes

- Event Queue Delete
- ExcelExportDownload
- Export Manager Pipeline
 - Queued Processes
 - Active Processes
 - Ended Processes
 - Exporting**
- Export to Transfer Package

Exporting - Background Process

Background Process Queue Info

Properties

Execution Report

Result

Exported file	exported.xls

FTP Delivery Method

The FTP delivery method allows an exported file to be delivered to an external system and is often used when the output files are large, or when a different or remote system is in use. FTP is also available in OIEP.

The preferred method for FTP delivery is the SFTP secure delivery method. For information on the SFTP delivery method, see **SFTP Delivery Method**.

Configuration

For information on a parameter, hover over the parameter label to display help text.

Export Manager

Steps

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

Select Delivery Method

FTP

Delivers exported data to a remote server via the File Transfer Protocol (FTP).

Host name: FTPServer

Proxy host name: 10.232.10.201

User name: user7

Password: ••••••••

File name template: \$filename-\$timestamp.\$extension

Zip before upload: No

Export Process Template: Exporter

Back Next Finish Cancel

1. In **Select Delivery Method**, choose **FTP** from the dropdown.
2. In **Host name**, enter the host name or IP address of the FTP server to be used for the delivery.
3. In **Proxy host name**, add the host name to be used for the server proxy. This field is optional.
4. In **User name**, enter the user name that has access to log on to the FTP server.
5. In **Password**, enter the password that will be used to log on to the FTP server.
6. In **File name template**, to create the file name used for the output, enter template variables, static text, or use a combination. The default template is \$filename-\$timestamp.\$extension and creates a file name like 'GenericXML-1490813266535.xml' based on the settings of the OIEP. Each variable is described below:

- **\$filename** This variable is replaced with text to indicate the format of the delivered file as specified in the Output Templates flipper. For example, the output file name would include the text 'excel' or 'GenericXML' for those formats, or the Event ID for event-based STEPXML. Using the default file name template, a comma-separated value file would be named 'csv-1490808860412.csv' and 'GenericXML-1490810861593.xml' would indicate that Generic XML was used.
- **\$timestamp** This variable is replaced with the time in milliseconds between January 1, 1970, and the time when the file is uploaded, which allows for a unique identifier. For example, 1479230247017 is used for an export on 15 NOV 2016.

Alternatively, the following options can be used to generate a readable date and can include static characters to separate the timestamp elements. Individual elements can be ordered as required, as shown below.

\$timestamp(ddMMyyyy) displays day, month, and year and is output '15112016' for 15 NOV 2016.

\$timestamp(yyyy-MM-dd_HH.mm.ss) displays year, month, day, hour, minutes, and seconds and is output as '2016-11-15_09.32.43' for 2016 NOV 15 at 9:32.43 AM.

- **\$extension** This variable is replaced with the extension of the output file based on the selected format in the Output Templates flipper. For XML-based formats, the output file is set with the .XML extension, while Excel files use .XLS, and comma-separated values use .CSV.

Note: The File Name Template does not support conversions of file formats and can only be used to deliver files in the format specified on the Configuration tab > Output Templates flipper of the outbound integration endpoint.

7. In **Zip export file**, select an option from the dropdown to specify if the output file should be delivered in a .ZIP (compressed) file format.
 - **Yes** uses the File Name Template and the extension .ZIP. When a file is zipped, the File Name Template is used for the .ZIP file. The \$filename template is replaced with 'result_0', and the selected file type is compressed with the name 'exported'. For example, a zipped STEPXML output with the default File Name Template exported on 15 NOV 2016 results in an output .ZIP file named 'result_0-1479230247017.zip'.
 - **No** uses the File Name Template for the file name along with the appropriate extension for the selected data format.
8. In **Export Process Template**, leave the default setting, which is required to start the background process for the export.
9. Click the **Finish** button to complete the wizard and begin the export as described in **Running a Data Export**.

JDBC Delivery Method

The JDBC delivery method allows STEP data to be delivered directly to tables in Relational database management systems (RDBMS) like Oracle, MySQL, MS SQL Server, and PostgreSQL. Though the ways the JDBC plugin can be deployed are various, one of its prime uses is to send STEP data to data analytics tools for the purpose of displaying STEP data in a data analytics dashboard. This delivery method requires CSV format and is available in both the Export Manager and in Outbound Integration Endpoint (OIEP).

For more information on the OIEP option, see **JDBC Delivery Method** in the **Outbound Integration Endpoints** documentation.

Prerequisites

Important: For complete setup requirements, see the **Exporting Data via JDBC with CSV Format** topic in the **Data Format** documentation.

Install the required drivers

JDBC specification 4.1-compliant drivers should be placed in a directory accessible from all application servers. These drivers can then be made available for the delivery plugin via the dynamic properties `JDBCDeliveryPlugin.DriverPath.[n]` and `JDBCDeliveryPlugin.DriverClass.[n]`. For more information regarding applicable Java drivers, review the RDBMS vendor's homepage on the web.

Configure data for the dropdown parameters

1. Prior to configuration, clicking the **Driver Location** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive `JDBCDeliveryPlugin.DriverPath.[n]` property. As an example:

```
JDBCDeliveryPlugin.DriverPath.1 = L:/shared/mysql-connector-java-5.1.42-bin.jar
```

In this example, the drivers are stored on the application server's L:/shared drive.

2. Prior to configuration, clicking the **Driver Class** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive `JDBCDeliveryPlugin.DriverClass.[n]` property. As an example:

```
JDBCDeliveryPlugin.DriverClass.1 = com.mysql.jdbc.Driver
```

In this example, the drivers class used is 'com.mysql.jdbc.Driver.'

3. Prior to configuration, clicking the **Database URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive `JDBCDeliveryPlugin.URL.[n]` property.

```
JDBCDeliveryPlugin.URL.1 = jdbc:mysql://localhost:3306/mydb
```

In this example, the database URL used is 'jdbc:mysql://localhost:3306/mydb.'

Configuration

For information on a parameter, hover over the parameter label to display help text.

Export Manager

Steps

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

Select Delivery Method

JDBC

Delivers exported data in CSV format into a database table via JDBC.

Driver Location: L:/jdbc-drivers/mysql-connector-java-5.0.8-bin.jar

Driver Class: com.mysql.jdbc.Driver

Database URL: jdbc:mysql://localhost:3306/mydb

Username: user1

Password:

Table Name: stepdata

Key Columns: id, datetime

Delete Key Columns: id

Export Process Template: Exporter

Back Next Finish Cancel

1. In **Select Delivery Method**, choose **JDBC** from the dropdown.
2. In **Driver Location**, select from the dropdown one of the paths to the relevant .jar file.
3. In **Driver Class**, select from the dropdown one of the pre-configured driver class.
4. In **Database URL**, select from the dropdown one of the pre-configured URLs to the destination database.
5. In **Username**, enter the username required to access the destination database.
6. In **Password**, enter the password required to access the destination database.
7. In **Table Name**, enter the name of the table in the destination database to which STEP will be publishing data
8. In **Key Columns**, list the names of the columns appearing on both the exported CSV file and the table in the destination database, separated by commas (and no spaces), into which STEP will publish data
9. In **Delete Key Columns**, list the names of the columns appearing on both the exported CSV file and the table in the destination database, separated by commas (and no spaces), from which STEP will delete data. The headers contained in this field can differ from the headers in the 'Key Columns' field, but they must also be part of the upsert key definition.
10. In **Export Process Template**, leave the default setting, which is required to start the background process for the export.
11. Click the **Finish** button to complete the wizard and begin the export as described in **Running a Data Export**.

Server Side Delivery Method

The server side delivery option is only available in Export Manager and saves the output file on the STEP application server via a background process. This delivery method is typically used to export a large file to a specified location, while an external system (consumer of the data) is looking for an input file in this specified location.

Configuration

For information on a parameter, hover over the parameter label to display help text.

1. Select **Server Side Delivery** from the dropdown.
2. For **Path**, specify the application server directory where the file should be delivered. The location of the delivery is included in the background process execution report. In the image above, 'workarea' would indicate that the output would be saved at <root>\workarea.
3. For the **Export Process Template** parameter, leave the default setting, which is required to start the background process for the export.
4. Click **Finish** to display the Save Export Configuration window as defined in **Running a Data Export**.

Output

By default, the output is saved on the Background Processes tab, under the Export Manager Pipeline node, which can be easily accessed as described in **Monitoring a Data Export**.

The screenshot shows the 'Exporting - Background Process' window. On the left, the 'BG Processes' tree is visible, with 'Exporting' selected under 'Ended Processes'. The right pane displays the details for the selected process, including 'Properties', 'Execution Report', and 'Result'. The 'Result' section shows 'Exported file' as 'exported.xls'.

SFTP Delivery Method

The preferred method for FTP delivery is the SFTP secure delivery method. The SFTP delivery method allows an exported file to be delivered to an external system and is often used when the output files are large, or when a different or remote system is in use.

SFTP is also available in OIEP. For information on the FTP delivery method, see **FTP Delivery Method**.

Configuration

For information on a parameter, hover over the parameter label to display help text.

1. In **Select Delivery Method**, choose **SFTP** from the dropdown.
2. In **Host name**, enter the host name of the SFTP server to be used for the delivery.
3. In **Proxy host name**, add the host name to be used for the server proxy. This field is optional.
4. In **User name**, enter the user name that has access to log on to the FTP server.
5. In **Password**, enter the password that will be used to log on to the FTP server.
6. In **File name template**, to create the file name used for the output, enter template variables, static text, or use a combination. The default template is \$filename-\$timestamp.\$extension and creates a file name like 'GenericXML-1490813266535.xml' based on the settings of the OIEP. Each variable is described below:
 - **\$filename** This variable is replaced with text to indicate the format of the delivered file as specified in the Output Templates flipper. For example, the output file name would include the text 'excel' or 'GenericXML'

for those formats, or the Event ID for event-based STEPXML. Using the default file name template, a comma-separated value file would be named 'csv-1490808860412.csv' and 'GenericXML-1490810861593.xml' would indicate that Generic XML was used.

- **\$timestamp** This variable is replaced with the time in milliseconds between January 1, 1970, and the time when the file is uploaded, which allows for a unique identifier. For example, 1479230247017 is used for an export on 15 NOV 2016.

Alternatively, the following options can be used to generate a readable date and can include static characters to separate the timestamp elements. Individual elements can be ordered as required, as shown below.

\$timestamp(ddMMyyyy) displays day, month, and year and is output '15112016' for 15 NOV 2016.

\$timestamp(yyyy-MM-dd_HH.mm.ss) displays year, month, day, hour, minutes, and seconds and is output as '2016-11-15_09.32.43' for 2016 NOV 15 at 9:32.43 AM.

- **\$extension** This variable is replaced with the extension of the output file based on the selected format in the Output Templates flipper. For XML-based formats, the output file is set with the .XML extension, while Excel files use .XLS, and comma-separated values use .CSV.

Note: The File Name Template does not support conversions of file formats and can only be used to deliver files in the format specified on the Configuration tab > Output Templates flipper of the outbound integration endpoint.

7. In **Zip before upload**, select an option from the dropdown to specify if the output file should be delivered in a .ZIP (compressed) file format.
 - **Yes** uses the File Name Template and the extension .ZIP. When a file is zipped, the File Name Template is used for the .ZIP file. The \$filename template is replaced with 'result_0', and the selected file type is compressed with the name 'exported'. For example, a zipped STEPXML output with the default File Name Template exported on 15 NOV 2016 results in an output .ZIP file named 'result_0-1479230247017.zip'.
 - **No** uses the File Name Template for the file name along with the appropriate extension for the selected data format.
8. In **Export Process Template**, leave the default setting, which is required to start the background process for the export.
9. Click the **Finish** button to complete the wizard and begin the export as described in **Running a Data Export**.

Websphere Commerce Import Delivery Method

The file delivery option is only available in Export Manager. It delivers a STEPXML file to a Websphere Commerce server.

Configuration

For information on a parameter, hover over the parameter label to display help text.

Export Manager

Steps

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

Select Delivery Method

Websphere Commerce Import

Imports STEPXML into Websphere Commerce Server

Server Command: WCSImport

Arguments: "%file%"

Stdout Prefix:

Stderr Prefix:

Abort Criteria: # Specify abort criteria:
exitcode: -1
exitcode: 1-255
stderr: not recognized as an internal or external command

Export Process Template: Exporter

Back Next Finish Cancel

1. Select **Websphere Commerce Import** from the dropdown.
2. For each of the available parameters, add the required information.
3. For the **Export Process Template** parameter, leave the default setting, which is required to start the background process for the export.
4. Click **Finish** to display the Save Export Configuration window as defined in **Running a Data Export**.

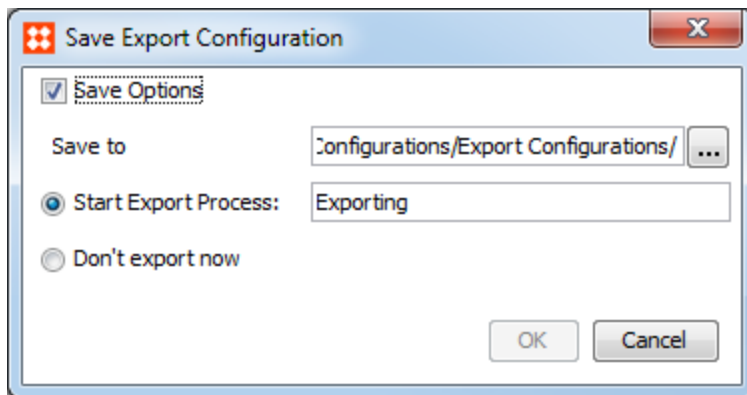
Running a Data Export

The Save Export Configuration dialog appears when you click the Finish button in the Export Manager in workbench. A saved configuration provides a way to keep all the parameter selections set while using the export wizard. Saving a configuration allows you to perform the same export again, without having to go through each step of the wizard. A saved configuration is required when scheduling exports. Whether you save your configuration or not, this dialog also allows you to start the export process.

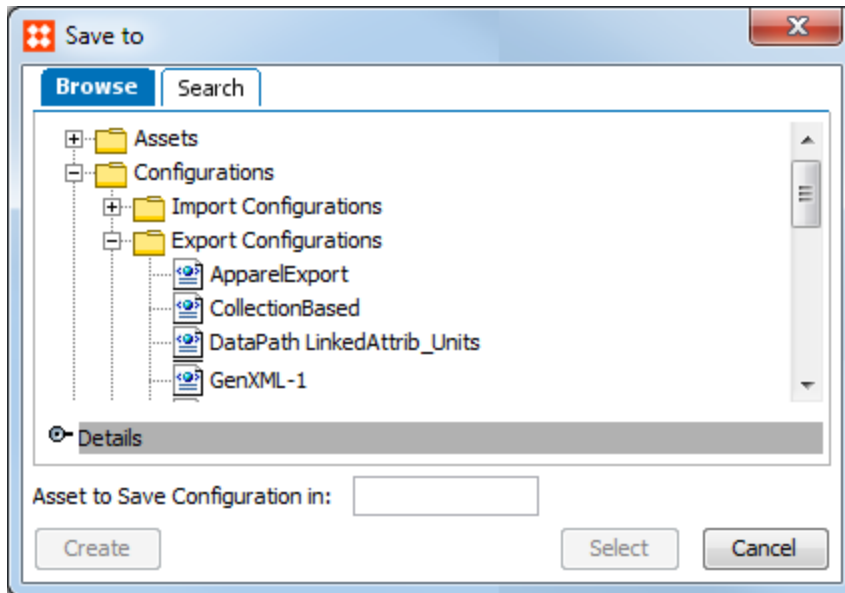
Save an Export Configuration

1. The **Save Options** checkbox is checked by default. Leave this setting if:
 - You made changes to an existing configuration and want to save those changes.
 - You want to save a new configuration.

Uncheck the Save Options checkbox if you do not want to save this configuration.



2. When the Save Options checkbox is checked, click the ellipsis button (...) on the Save to field to display the **Save to** dialog.
3. On the **Save to** dialog, choose the option for your situation.
 - To save a new configuration: use the Browse or Search option to select the folder for the new configuration. Type the name of the new configuration in the **Asset to Save Configuration in** parameter, and click the **Create** button that is now enabled.
 - To update an existing configuration: use the Browse or Search option to select the configuration, and click the **Select** button which is now enabled.



4. Back on the Save Export Configuration dialog, choose a radio button:
 - Select **Start Export Process** and add a name in the field to make it easier to locate the specific process later.
 - Select **Don't export now** to only save changes to the configuration.
5. Click **OK** to complete the wizard. If you started an export, the **Starting Process** dialog appears. For more information, see **Monitoring a Data Export**.

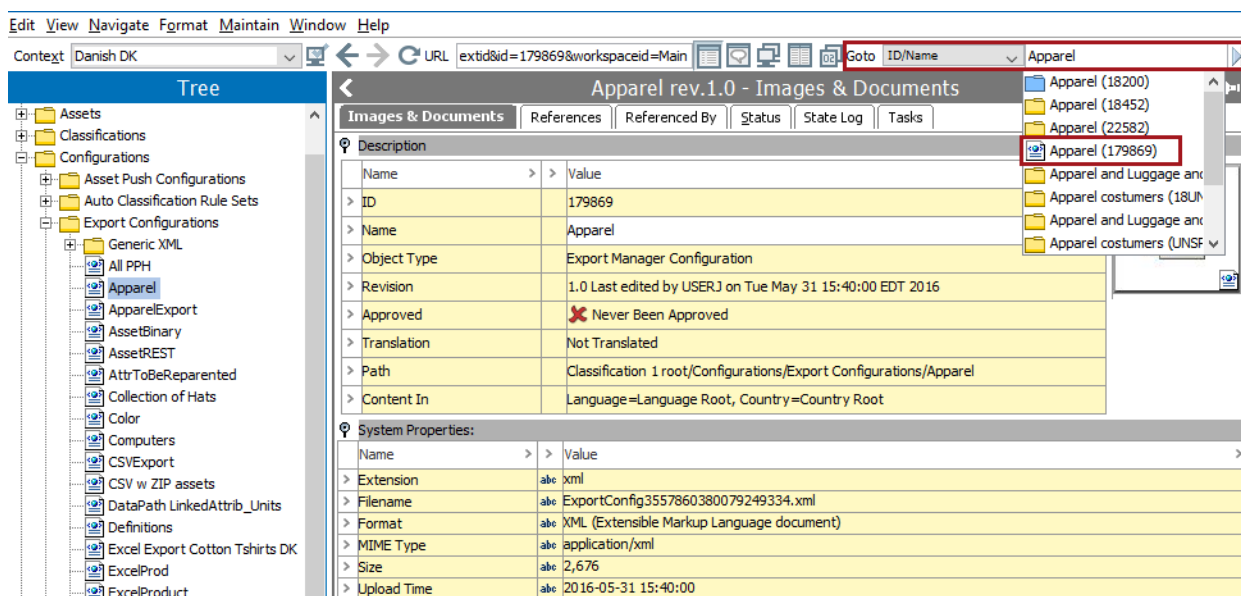
Maintaining a Saved Export Configuration

A saved export configuration allows you to repeatedly export a group of objects using the same wizard parameter settings. Completing the Export Manager wizard in workbench includes an option to save an export configuration (see **Running a Data Export**). Viewing a saved configuration allows you to confirm the current settings. Editing a saved configuration affects future exports that use the configuration, and can be a valuable step in the process of creating and testing an export to ensure the correct data is output in the proper manner.

View a Saved Export Configuration

After clicking the Finish button on the Export Manager wizard, the option to save the configuration is displayed. Although you can save an export configuration to any classification folder, using a specific folder for export configurations makes it easier to locate them for review and running.

Note: If you know the name of the configuration, use the **Goto** search option to find it anywhere in the Tree, using the icon to distinguish the configuration from other objects.



The object type of a saved export configuration is Export Manager Configuration. The metadata for the configuration displayed in the System Properties flipper cannot be edited.

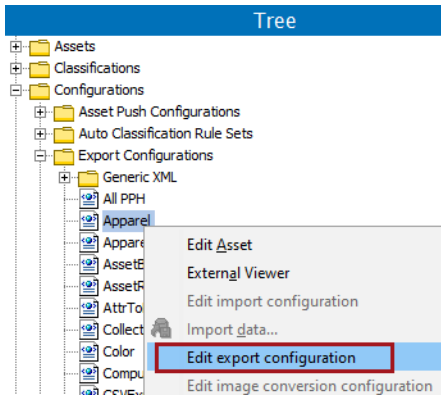
Edit a Saved Export Configuration

Use the following steps to apply updates to the configuration selected for editing.

- To change only the name of the configuration, simply edit the Name field in the configuration editor.
- To edit an existing configuration and save with a new name, see the steps in the **Running a Data Export** section.
- To change the settings of the existing configuration parameters, use the edit option or edit via XML. Both methods are defined below.

Edit export configuration via Export Manager

- 1. Select an export configuration, right-click and select **Edit export configuration** option.



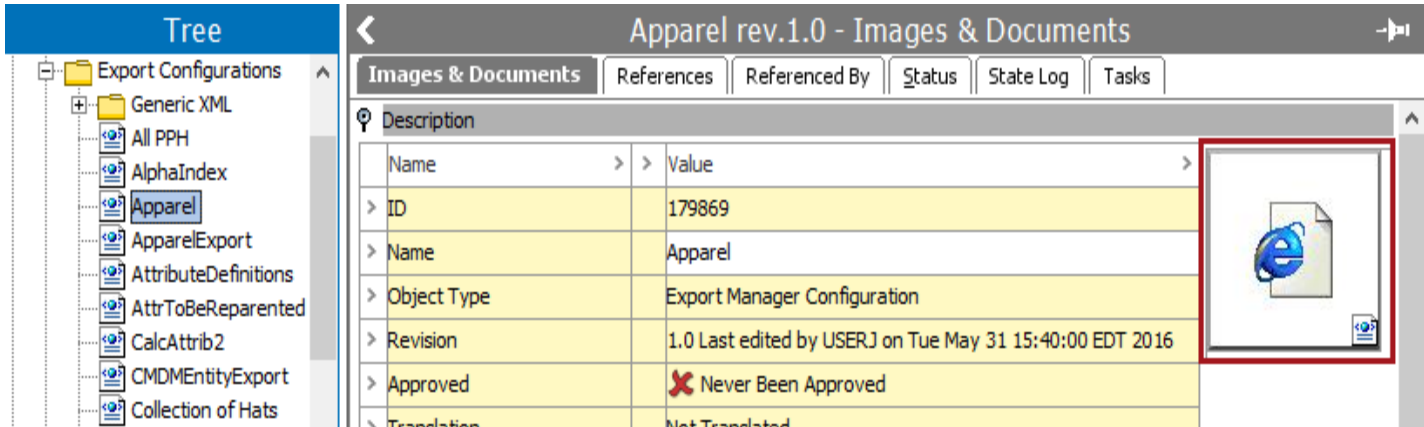
- 2. The Export Manager wizard opens on the Select Objects step.
- 3. Make all changes in each step and click the **Finish** button to update the selected configuration.

Edit export configuration via XML

This method should be attempted only if user is well versed with STEPXML. Even a minor error may corrupt the configuration file.

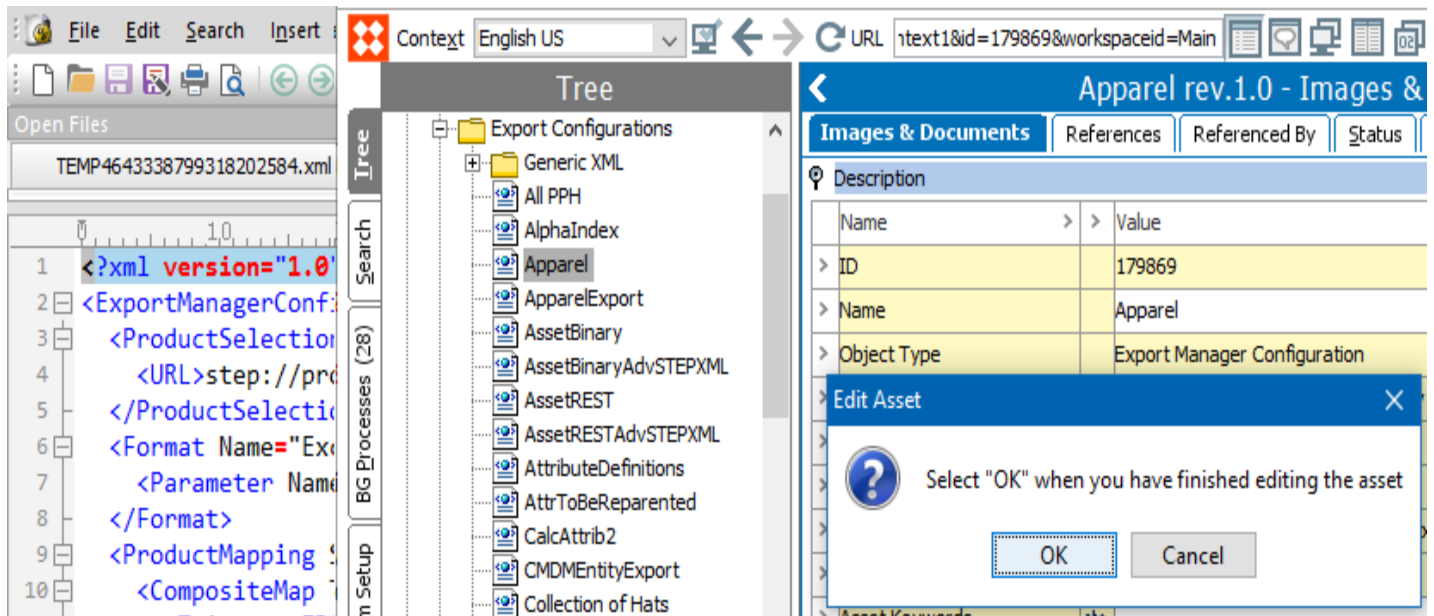
Note: Changing the name of the export configuration via XML file is not allowed.

- 1. Select an export configuration, double-click the XML file displayed on the Images & Documents tab.



The XML opens in an editor, along with an Edit Asset dialog.

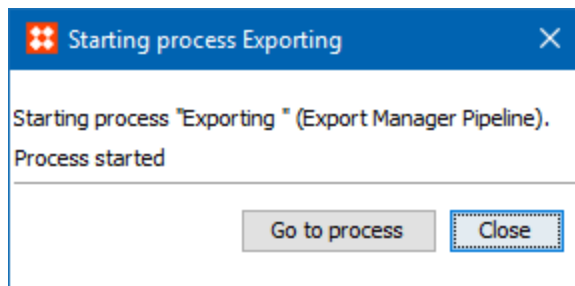
Important: Do not dismiss the Edit Asset dialog until updates to the XML file are complete.



2. Make necessary changes in the XML file editor, save, and close the file.
3. In the Edit Asset dialog, click **OK** to save the changes, or **Cancel** to discard them.

Monitoring a Data Export

When you launch a data export process (as described in the **Running a Data Export** topic), the system starts a background process. It assigns a process number to that particular data export, and gives it the name assigned on the Save Export Configuration dialog. The starting process dialog is displayed:



Monitoring a data export allows users to view the process' activity and status, take action based on the process' result, analyze the execution report for details, and truncate the execution report to eliminate unneeded information.

Monitoring the Export Process

There are two ways to monitor a particular data export process:

- In the **Starting Process** window (that appears when you launch the export), click the **Go To Process** button.
- On the **BG Processes** tab select the **Export Manager Pipeline** node. For processes not yet started, open the **Queued Processes** folder. For processes that are currently being performed, look in **Active Processes** folder. For processes that have completed, look in **Ended Processes** folder.

The Properties flipper includes details about the export process.

BG Processes

Event Consumer

Event Processor

Event Queue Delete

ExcelExportDownload

Export Manager Pipeline (3)

Queued Processes

Active Processes

Ended Processes (3)

Export to Transfer Package

GDSNConverter

GDSNReceiverSubImport

GDSNValidation

GetPDFOfActualPage

Global Update

ISOStrict

Import Manager Pipeline

Import Transfer Package

ImportInDesignPackage

Install Change-Package

Match Code Processes

Matching Pair Export

Matching Processes

Migrate assets/DTP documents to the file

MountDocForPlan

MultipleRevive

NodeListToCollection

PDF of planned page

PageDone

Pre-generate thumbnail cache

Publication Duplication

Publish to Website

Exporting - Background Process

Background Process Queue Info

Properties

Property	Value
Started by	USERJ
Id	BGP_179834
Description	Exporting
Execution Server	doc-dev
Progress	Done
Status	succeeded
Created	Thu May 26 15:37:09 EDT 2016
Started	Thu May 26 15:37:10 EDT 2016
Finished	Thu May 26 15:37:11 EDT 2016
Processing Time	0 m 1 s
Time in Queue	0 m 1 s
# of warnings	0
# of errors	0

Execution Report

- Export process started (Thu May 26 15:37:10 EDT 2016)
- Logging on to PIM server doc-dev as USERJ...
- Logged on
- Analysis started. (Thu May 26 15:37:10 EDT 2016)
- Analyzed 1 objects from initial object selection in 0 seconds. (Thu May 26 15:37:10 EDT 2016)
- Analyzed 0 children objects (including aggregates) in 0 seconds. (Thu May 26 15:37:10 EDT 2016)
- Analyzed 6 initial parents in 0 seconds. (Thu May 26 15:37:10 EDT 2016)
- Analyzed 3 objects referenced from other objects in 0 seconds. (Thu May 26 15:37:11 EDT 2016)
- Analyzed 0 referenced parents in 0 seconds. (Thu May 26 15:37:11 EDT 2016)

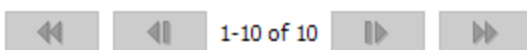
Execution Report

The execution information includes the status of the process and the number of warnings and errors.

Typically, you do not monitor the process as it is being carried out. However, if the process fails and you need to investigate why or if you need to contact Stibo Help Desk, the information can be useful.

Navigation

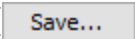
After the first entry, the Execution Report displays the following navigation buttons:

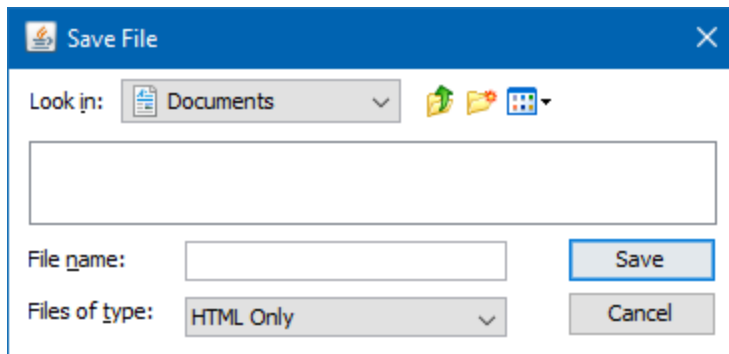


The buttons are enabled when until the log reaches over 100 lines.

- 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.

Save

1. Click the Save button () to save the Execution Report as an 'HTML Only' file type.

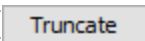


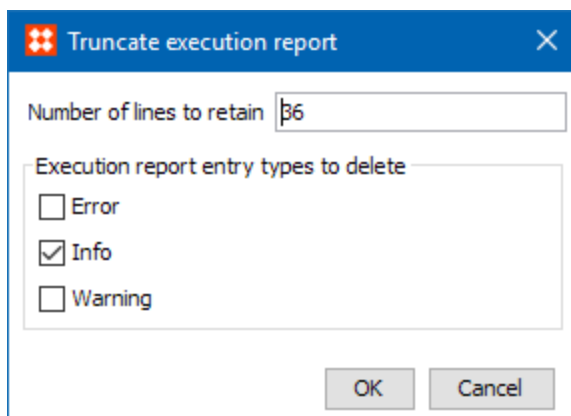
Note: Changing the default format manually may result in an error.

2. Enter a name for the log report and click **Save**.

Truncate

When the Execution Report become cluttered or only specific information is needed, use the Truncate option to permanently eliminate selected rows.

1. Click the Truncate button () to display the Truncate execution report dialog:



2. In **Number of lines to retain**, the current number of log entries will display. Type the number of lines to keep starting from the end of the log, and permanently deleting the oldest rows, based on the selected entry types below.
3. In **Execution report entry types to delete**, select the types that should be permanently removed.

Note: The Properties flipper includes a count of the Warnings and Errors included in the Execution Report.

4. Click OK to permanently remove the selected number and types of execution report entries.

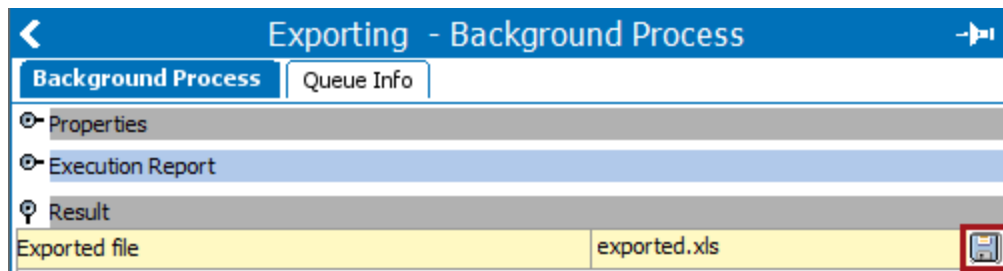
Result

The selected delivery option determines if you can download the file from the Background Process tab. When available, the file is stored on the STEP application server, and can typically be found at workarea\background-processarea\Exporter, in a folder named for the background process ID (for example, BGP_179834 in the image above),.

Depending on your system setup, you may also be able to edit the file from within STEP.

Download an Export File

1. Click the save icon to the right of the file name.



2. In the **Save** window, find the location where to save the file, and click **Save**.

You can now open and edit the file. When you save the file, you can also update the file in STEP.

View an Export File

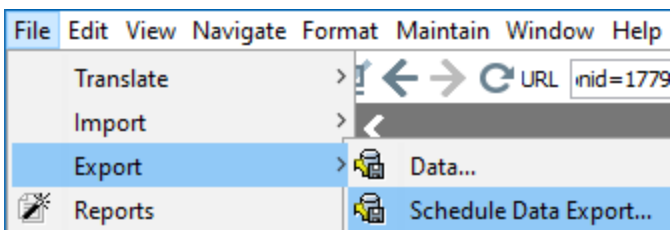
Double-click the name of the exported file to open it using an available editor for the output format. In the image above, the file will be opened in Excel.

Scheduling a Data Export

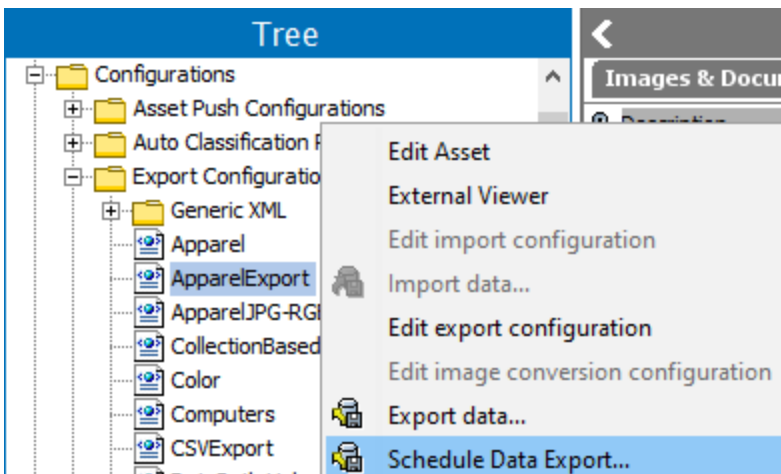
Scheduling a data export provides additional functionality for the Export Manager, allowing users to repeat an export with regularity. For example, sending the same data to the same location or recipient on a regular basis, to be consumed by a downstream system. Scheduling requires that an export configuration is saved under any of the classification folders. A saved configuration eliminates the need to set the export parameters each time the data should be output.

Create a Scheduled Export

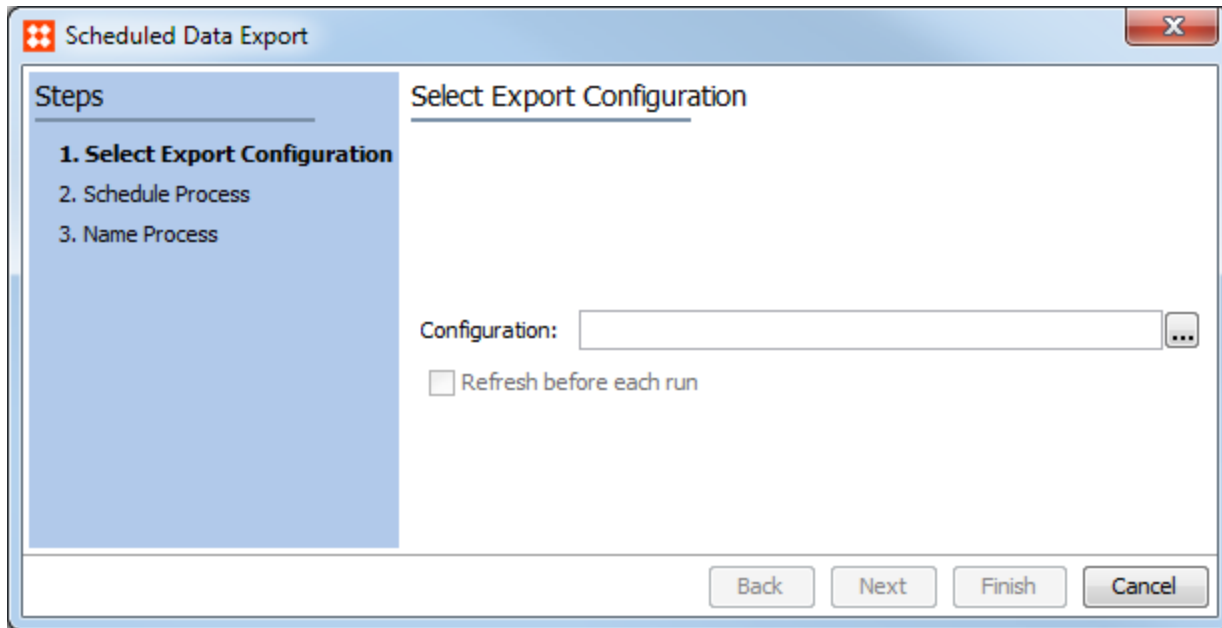
1. Select the appropriate context in STEP.
2. Verify the required Export Configuration already exists for the export you want to schedule. To create a new one, use the steps described in the **Creating a Data Export** documentation and then save the configuration as described in the **Running a Data Export** section.
3. Open the Schedule Data Export wizard using one of these methods:
 - From the File menu > Export > **Schedule Data Export**.



- In the Tree, select an existing export configuration, right-click, and select **Schedule Data Export**.

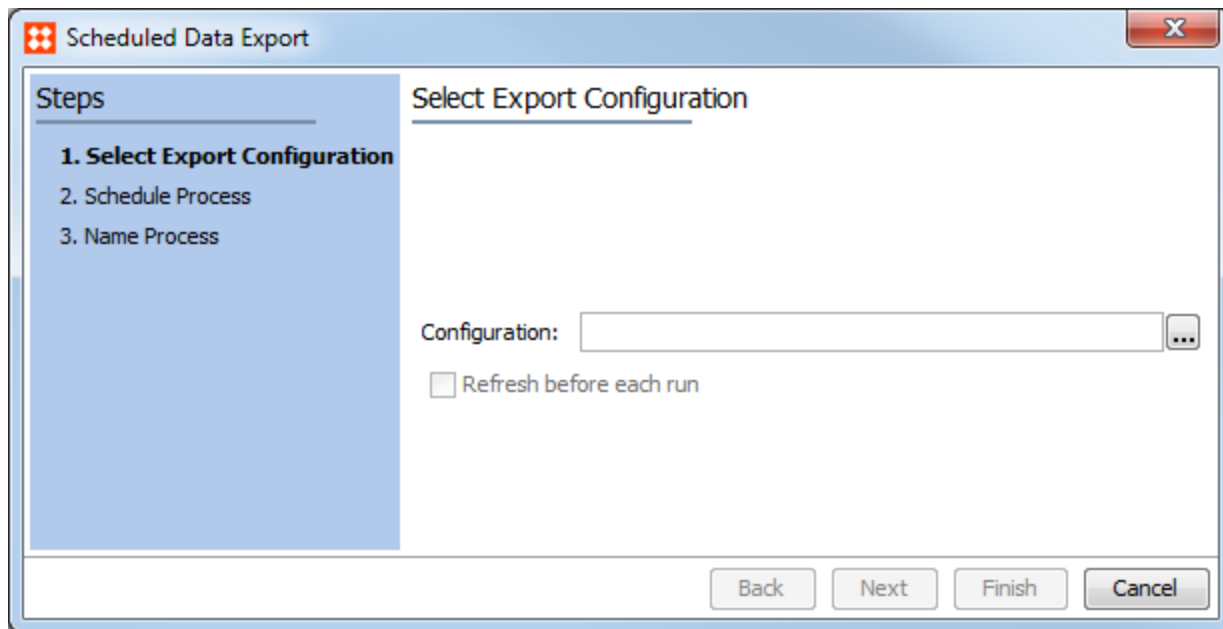


4. After selecting one of the above options to start the Schedule Data Export wizard, the following dialog is displayed and includes these steps:



- **Select Export Configuration** allows you to select the relevant export configuration.
 - **Schedule Process** allows you to schedule the export.
 - **Name Process** allows you to name the process so that you can locate it on the BG Processes tab.
5. Manage scheduled data exports using the following information:
- **Scheduled Process Properties** displays the status of a scheduled process from queued, to active, and finally when it has ended.
 - **Deleting a Scheduled Data Export** removes a scheduled export process, which is required to run current parameters when the original configuration is modified.

Scheduled Data Export - Select Export Configuration



1. For **Configuration**, use a method to select the required configuration:
 - Type the configuration name or ID into the text box and select it from the matches displayed in the dropdown.
 - click the ellipsis button (...). The Select Export Configuration window displays. Browse or search for the configuration, and click **Select**.
2. If the configuration is based on a Collection, the **Refresh before each run** checkbox is enabled. For more on collections, see the **Collections** documentation.
 - Check the box to update the collection before each scheduled export.
 - Leave the box unchecked if you want to only export the objects currently in the collection at the time of export.
3. Click **Next** to display **Scheduled Data Export - Schedule Process**.

Scheduled Data Export - Schedule Process

1. Select one of the following options.

- **Now** runs the process only once when you click **Finish**.
- **Later** runs the process only once at the specified date and time.
- **Weekly** runs the process on the days select in the specified period until the process is deleted.
- **Monthly** runs the process every month at the specified day and time until the process is deleted.
- **Later and repeat** runs the process at the specified start date and time, and repeatedly after a specified number of minutes between attempts to export until the process is deleted.

Note: Since a scheduled export process continues to use the configuration it was started on, if you change the parameters of the export configuration, you should delete the relevant scheduled processes. See **Deleting a Scheduled Data Export**.

2. Click **Next** to display **Scheduled Data Export - Name Process**.

Scheduled Data Export - Name Process

1. For **Schedule Process Name**, enter a name for the scheduled process. This name is displayed on the process on the BG Processes tab > Scheduled Processes node for all schedule options except 'Now.' The default value includes the name of the export configuration and scheduling details.
2. In the **Process Name** field, enter a name for the exporting process. This name is displayed on the BG Processes tab > Export Manager Pipeline. The default value includes the name of the export configuration.
3. Click **Finish**.

To monitor scheduled processes, see the **Scheduled Process Properties** section.

Scheduled Process Properties

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

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

A scheduled process starts in the Queues Processes folder until the scheduled time arrives. Then the process appears in the Active Processes folder until it is completed. The same process again goes into the Scheduled Processes folder until the next scheduled time. The process is moved to Ended Processes after successful completion of schedule or if it is aborted while not running. If a user aborts the process before the schedule is complete, or if there is an error in the BGP, the process moves to the Active Processes folder with red text indicating an error execution.

The screenshot shows the 'BG Processes' editor. On the left, a tree view shows the hierarchy: BG Processes > Scheduled Processes > Queued Processes > 'Export using 'Collection of Hats''. The main panel displays the properties of this process in a table format.

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

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

- **Started by** shows the user who scheduled the export process.
- **ID** shows the background process ID.
- **Description** is the text added in the 'Scheduled Process Name' field when the scheduled export was created.
- **Schedule** defines the frequency that the export is scheduled to run.
- **Execution Server** shows the name of the database server running the export.
- **Status** can be waiting, active, or aborted, depending on the current situation.
- **Created** shows the day, date, and time that the scheduled export was created.

- **Started** is the day, date, and time that the scheduled export first ran.
- **# of warnings** is the number of warning messages included in the execution report.
- **# of errors** is the number of error messages included in the execution report.
- **Next Run** is the day, date, and time that the export will run again.

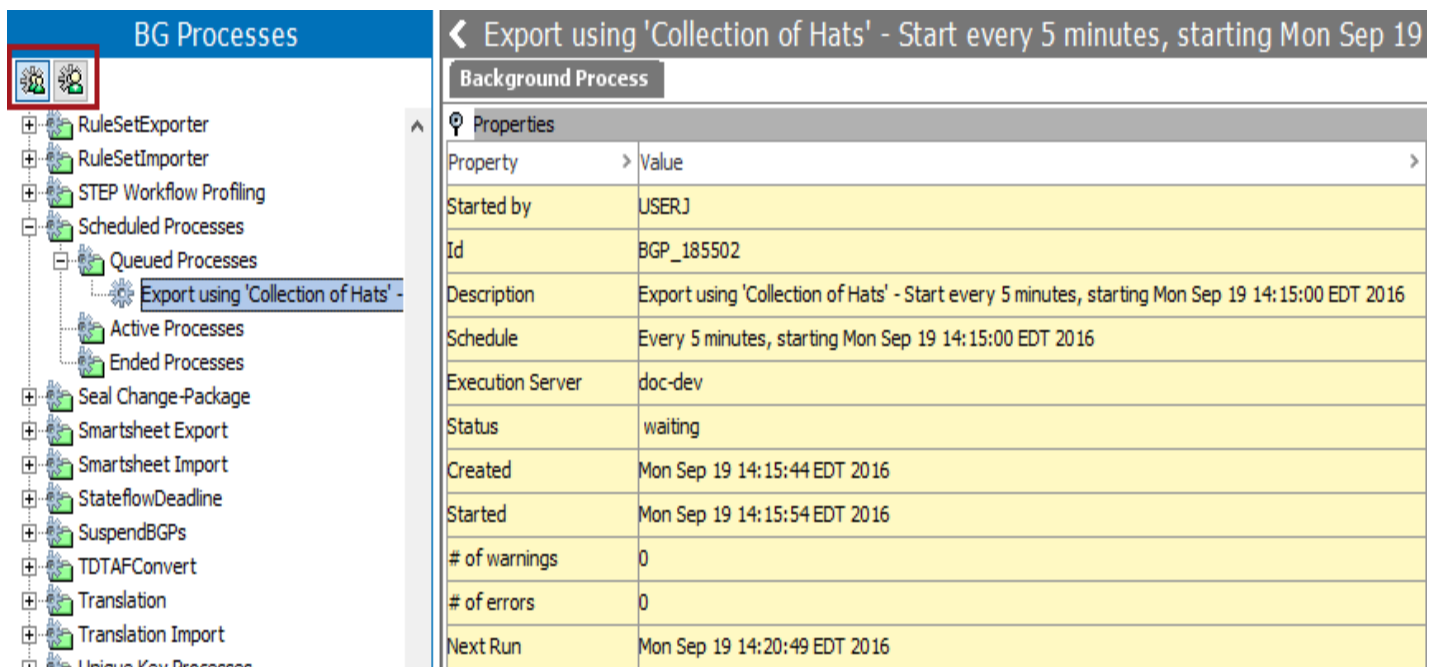
Deleting a Scheduled Data Export

A scheduled export process continues to use the configuration it was started on. Therefore, if you change the parameters of the export configuration, you should delete the relevant scheduled processes.

Once an export 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 Data Export

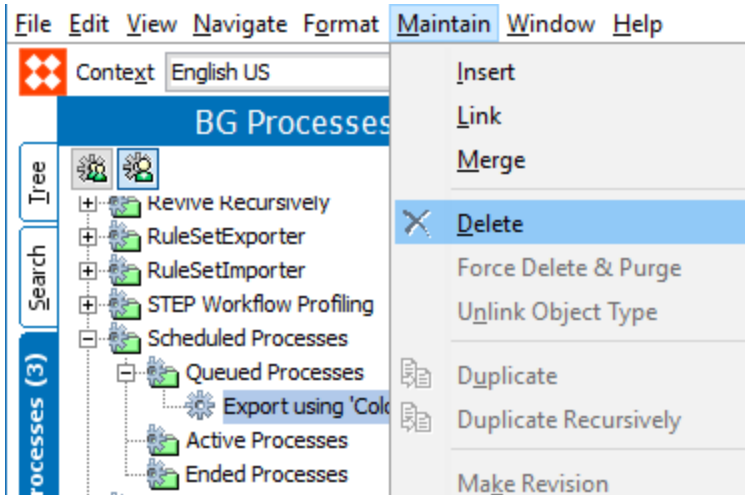
1. On the BG Processes tab, expand Scheduled Processes > **Queued Processes**. If needed, use the filter buttons at the top of the BG Processes tab to display processes started by other users.



The screenshot displays the 'BG Processes' interface. On the left, a tree view shows the hierarchy: BG Processes > Scheduled Processes > Queued Processes. The process 'Export using 'Collection of Hats' -' is selected. On the right, the 'Background Process' properties are shown in a table.

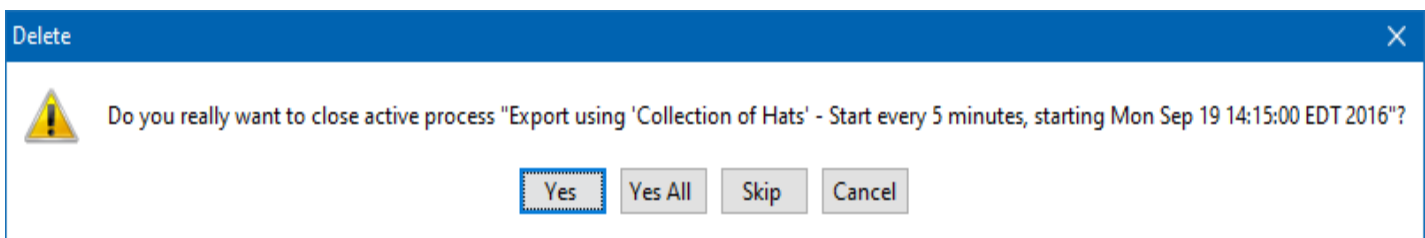
Property	Value
Started by	USERJ
Id	BGP_185502
Description	Export using 'Collection of Hats' - Start every 5 minutes, starting Mon Sep 19 14:15:00 EDT 2016
Schedule	Every 5 minutes, starting Mon Sep 19 14:15:00 EDT 2016
Execution Server	doc-dev
Status	waiting
Created	Mon Sep 19 14:15:44 EDT 2016
Started	Mon Sep 19 14:15:54 EDT 2016
# of warnings	0
# of errors	0
Next Run	Mon Sep 19 14:20:49 EDT 2016

2. Select the one or more processes to delete, and from the **Maintain** menu, click **Delete**.



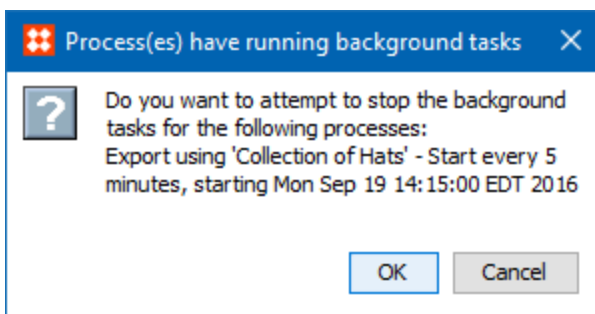
Important: This step moves the scheduled process to an aborted status and displays it in red text as an Ended Process. The details of the process continue to display, however, an aborted scheduled process cannot be started again. To run it again, use the details to recreate the scheduled process.

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.

The screenshot displays the Stibo Systems interface. On the left, a tree view under 'BG Processes' shows 'Export using 'Collection of Hats' - Start every 5 minutes, starting Mon Sep 19 14:15:00 EDT 2016' selected under 'Ended Processes'. On the right, a properties table for this process is shown:

Property	Value
Started by	USERJ
Id	BGP_185502
Description	Export using 'Collection of Hats' - Start every 5 minutes, starting Mon Sep 19 14:15:00 EDT 2016
Schedule	Every 5 minutes, starting Mon Sep 19 14:15:00 EDT 2016
Execution Server	doc-dev
Status	aborted

Below the properties table, a 'Delete' dialog box is open, asking: 'Do you really want to delete the ended process "Export using 'Collection of Hats' - Start every 5 minutes, starting Mon Sep 19 14:15:00 EDT 2016"?'. The dialog has four buttons: 'Yes', 'Yes All', 'Skip', and 'Cancel'.

Gateway Integration Endpoints

Gateway integration endpoints enable STEP to access external systems through business rules that make calls to fetch data or update status. For example, to create objects in STEP using an ID from an external Enterprise Resource Planning (ERP) system, STEP can use a gateway endpoint to access and retrieve the external system ID.

The gateway endpoint controls all access to a given system. It holds the information that is required to access the external system including the user name, password, and the server URL. This means the information does not have to be included in the JavaScript business action and that the duration and frequency of calls can be monitored and logged.

The business rule uses synchronous REST calls from JavaScript. For more information about REST and the REST API, click the **STEP API Documentation** button on the STEP WebStart page.

Note: Keystore is not supported.

Setup Requirements

Setting up and using a gateway integration endpoint involves the following steps:

1. Create a setup group to hold the endpoint as described in **Initial Setup for a Gateway Integration Endpoint**.
2. Create a gateway endpoint as described in **Creating a Gateway Integration Endpoint**.
3. Configure the endpoint as described in **Configuring a Gateway Integration Endpoint**.
4. Create a business rule to access the gateway endpoint as defined in **Gateway Integration Endpoint Bind**.
5. Enable the endpoint and test connectivity as described in **Running a Gateway Integration Endpoint**.

Additional Information

The following information is useful once a gateway integration endpoint is set up:

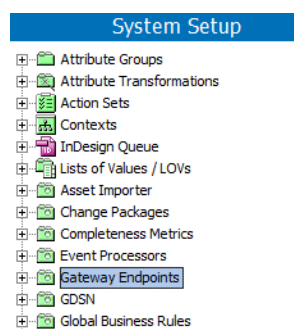
1. Maintain or modify the endpoint as described in **Maintaining a Gateway Integration Endpoint**.
2. Monitor the endpoint as described in **Monitoring a Gateway Integration Endpoint**.
3. Export gateway integration endpoint definition for comparison purposes in an external source control system in **Exporting Gateway Integration Endpoint Definitions as Comments**.

Initial Setup for a Gateway Integration Endpoint

Before creating a gateway integration endpoint, one or more setup groups must be created that are allowed to hold gateway integration endpoints. You must also specify the setup group(s) in which gateway integration endpoints can be created. This setup only needs to be performed once, and most systems will already have it completed.

Only users with the relevant privileges can view or maintain gateway integration endpoints. For detailed information, see the **Action Sets** section and the **Users and Groups** section in the **System Setup / Super User** documentation.

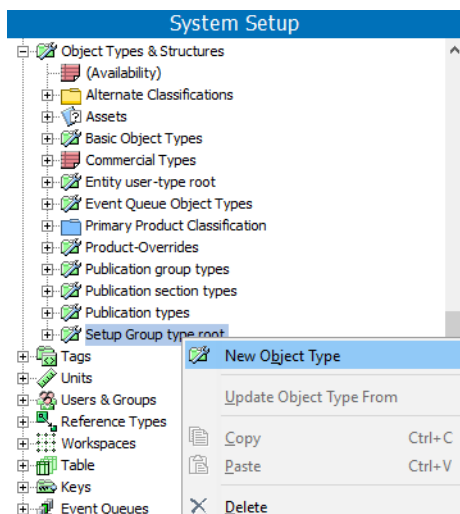
Review your System Setup tab to determine if one or more gateway integration endpoint nodes already exist. The name of the node on your system should include the word 'gateway' but is not required to match the one in the image below.



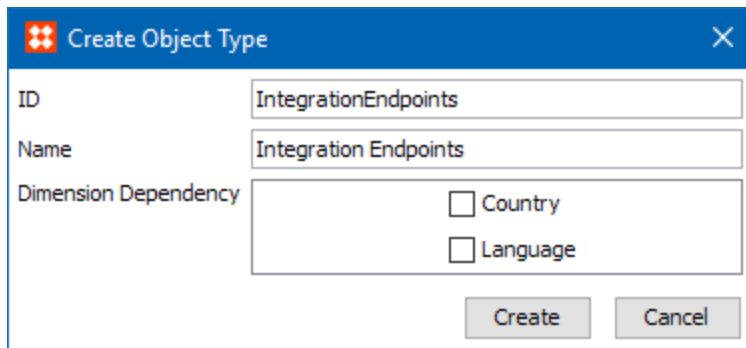
Once the setup has been completed, the steps in this section are only needed if additional levels of organization are desired.

Create the Gateway Setup Group

1. In System Setup, expand **Object Types & Structures**.
2. Right-click 'Setup Group type root', and choose **New Object Type**.

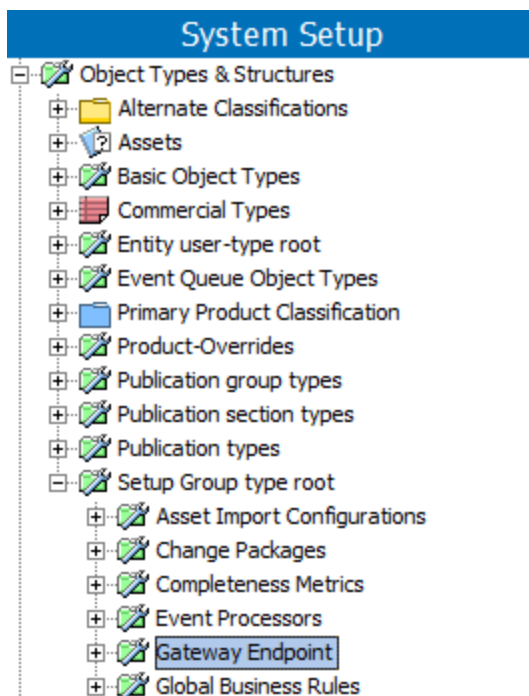


3. Enter an **ID** and a **Name**, select any required Dimension Dependencies, and click **Create**.



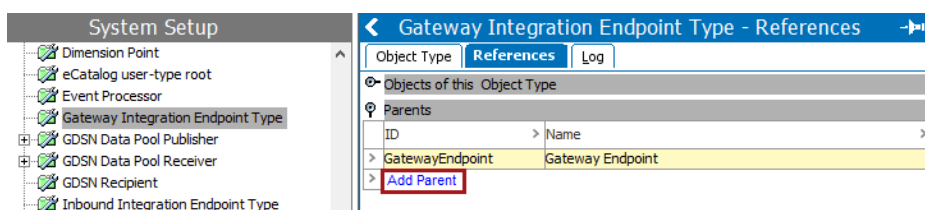
The 'Create Object Type' dialog box has a blue header with a Stibo logo and a close button. It contains three input fields: 'ID' with the value 'IntegrationEndpoints', 'Name' with the value 'Integration Endpoints', and 'Dimension Dependency' which is a list box containing 'Country' and 'Language', both with unchecked checkboxes. At the bottom are 'Create' and 'Cancel' buttons.

4. The new setup group appears in System Setup under 'Object Types & Structures' as a child in the **Setup Group type root**.

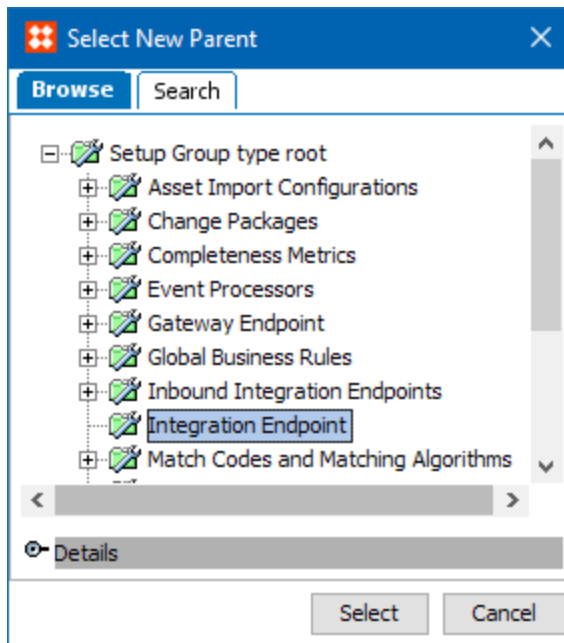


Link the Gateway Object Type to Setup Group

1. In Object Types & Structures > expand Basic Object Types > select **Gateway Integration Endpoint Type**.
2. On the References tab > Parents flipper > click the **Add Parent** link.

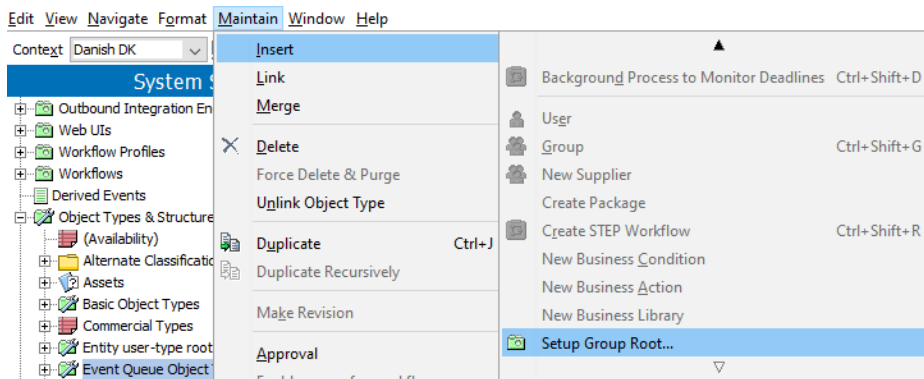


3. In the Select New Parent dialog, select the setup group you created, and click **Select** to make it a valid parent.

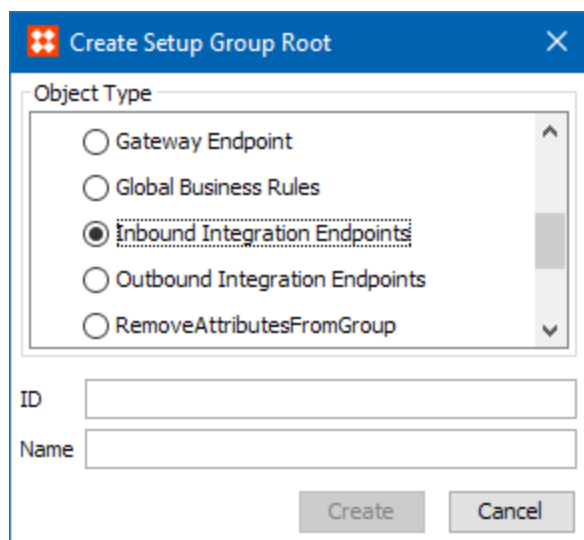


Create an Instance of the Gateway Endpoint Object

1. On the System Setup tab, select any object in the **System Setup** hierarchy to activate the following Maintain menu selection.
2. Click the Maintain menu, point to Insert, and select **Setup Group Root**.

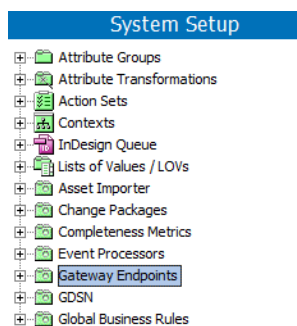


3. In the Create Setup Group Root dialog, select the gateway object type, enter an **ID** and a **Name**, and click **Create**.



The dialog box is titled "Create Setup Group Root" and has a close button (X) in the top right corner. It contains a section labeled "Object Type" with a list of five radio button options: "Gateway Endpoint", "Global Business Rules", "Inbound Integration Endpoints" (which is selected and has a dotted border), "Outbound Integration Endpoints", and "RemoveAttributesFromGroup". Below this list are two text input fields labeled "ID" and "Name". At the bottom right are two buttons: "Create" and "Cancel".

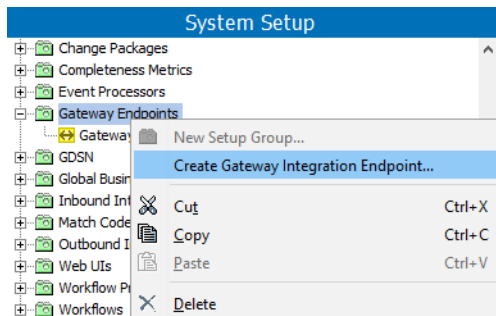
A setup group is created as a node in the System Setup hierarchy. Gateway integration endpoints can now be created under this new node.



Creating a Gateway Integration Endpoint

After creating a setup group for gateway integration endpoints, create a gateway integration endpoint to access data on an external server.

1. In System Setup, right-click the Gateway Endpoints setup group, and click **Create Gateway Integration Endpoint**.



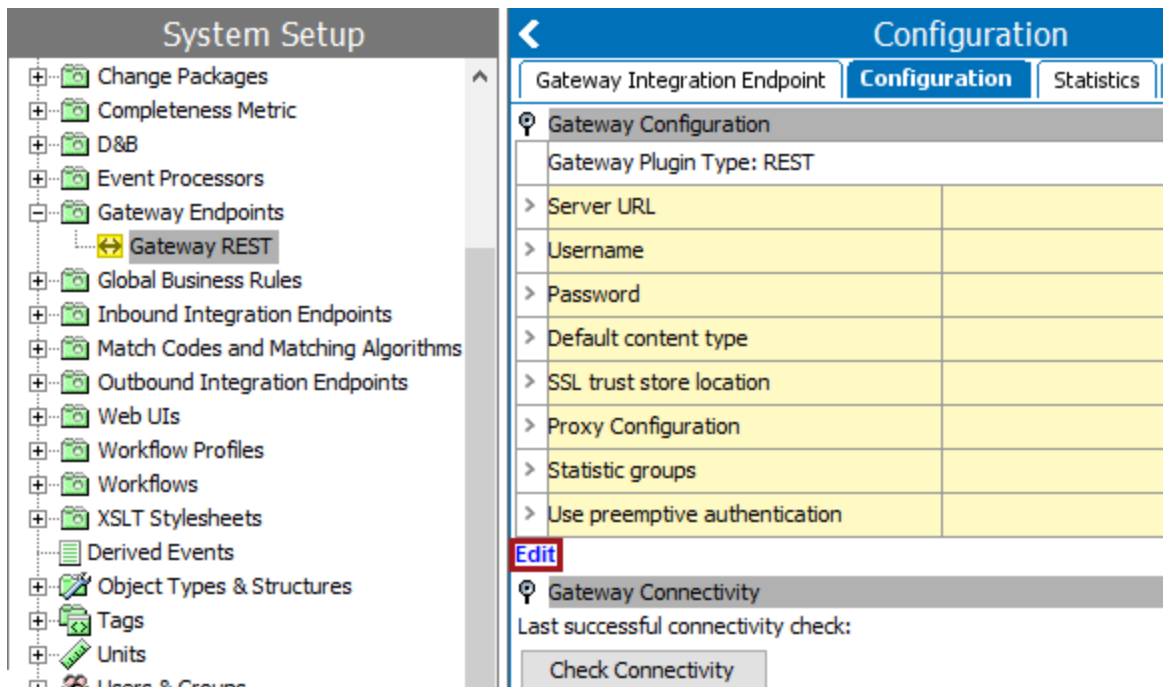
2. Enter an **ID** and a **Name** for the integration endpoint. The Description parameter is optional.

3. Click **Create** and then configure the endpoint. For more information, see the **Configuring a Gateway Integration Endpoint** documentation.

Configuring a Gateway Integration Endpoint

Once a gateway integration endpoint is created, the configuration settings allow you to identify the server housing the required data.

1. In System Setup, expand the gateway endpoints setup group, and select the relevant gateway integration endpoint.
2. On the Configuration tab, click the **Edit** link to display the Gateway Integration Endpoint Configuration dialog.



3. On the Gateway Integration Endpoint Configuration dialog, use the following parameters to specify which external system the gateway integration endpoint will access.

Gateway Integration Endpoint Configuration

REST

Server URL: http://usest.stibo.com/restapi/

Username: connusername

Password:

Default content type: application/xml; charset=UTF-8

SSL trust store location:

Proxy Configuration: ProxyConfig1

Statistic groups: */products/*

Use preemptive authentication: ☐

Save Cancel

- **Server URL** - The URL of the server to be accessed. The available URLs are system dependent and must be configured in the sharedconfig.properties file using the RESTGateway.ServerURL property. If necessary, use a comma to separate multiple URLs. The following is an example of a complete property entry for two systems named 'qa' and 'stage,' as well as their URLs 'http://step-qa' and 'http://step-stage':

```
RESTGateway.ServerURL=qa=http://step-qa,stage=http://step-stage
```

- **Username** - Login required to establish the connection.
- **Password** - Password required to establish the connection.
- **Default content type** - Data format or content type to be used for the endpoint. Type your own or select from the dropdown list of the most commonly used options.
- **SSL trust store location** - The trust store holds certificates that verify that the system can be trusted. To display a trust store in the dropdown, first update the 'RESTGateway.SSLTrustStoreLocation' property in the sharedconfig.properties file. Contact Stibo Systems technical support for assistance in adding a trust store to your system.

If no trust store locations are selected, the gateway connection is not configured to use SSL encryption and the connection will be established using a certificate signed by a recognized CA such as Verisign or Thawte.

- **Proxy Configuration** - Select from this dropdown if the gateway connection must first pass through a proxy server with its own login requirement. The configurations that appear in this dropdown are populated from properties added into the sharedconfig.properties file using the dynamic property 'RESTGateway.ProxyConfiguration'. See the **Proxy Configuration Setup** section below for details.

- **Statistic groups** - Optional. Regular expression can be used to group executed REST methods on the endpoint's Statistics tab. The regular expression must match the entire URL, for example, `.*products/.*`. Use the regular expression syntax available in Java in `java.util.regex.Pattern` class. For more information, see the **Gateway Integration Endpoint Statistics** section or the **Regular Expression** topic in **System Setup / Super User Guide** documentation.
 - **Use preemptive authentication** - Optional. Check this box to allow privileged users to grant this gateway the ability to override the default usage of normal basic authentication and use preemptive authentication instead. With preemptive authentication, every time the gateway binding invokes a request (GET, PUT, POST, etc.), the authentication header containing the user name and password will be applied to the HTTP request.
4. Click **Save** to complete the configuration.
 5. Create a business action to access the gateway endpoint. For details, see **Gateway Integration Endpoint Bind**.

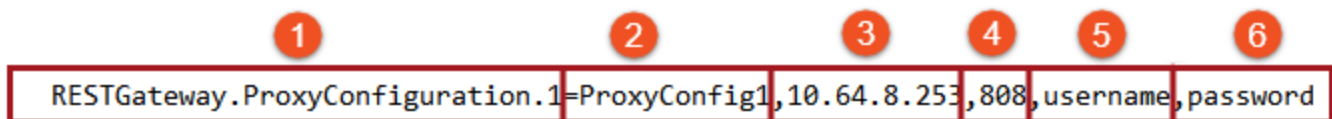
Proxy Configuration Setup

The following are two examples of how the dynamic `RESTGateway.ProxyConfiguration` property can be configured:

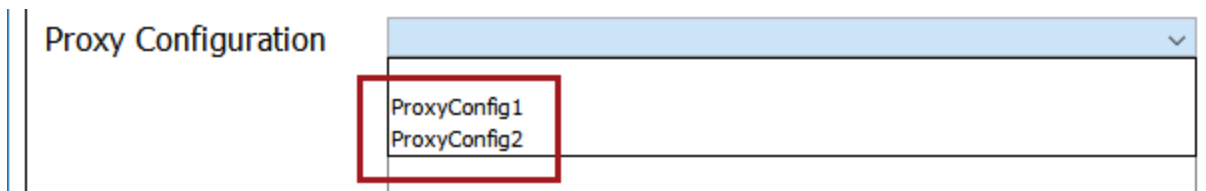
```
RESTGateway.ProxyConfiguration.1=ProxyConfig1,10.64.8.253,808,username,password
```

```
RESTGateway.ProxyConfiguration.2=ProxyConfig2,10.64.9.253,64,user10,password1
```

To break the first of these two properties into its component parts (each element separated by commas), please reference the screenshot and the numbered list below:



1. Required text to enable the REST Gateway Proxy configuration property
2. The name of the configuration. This is what will appear in the dropdown as a selectable option, like so:

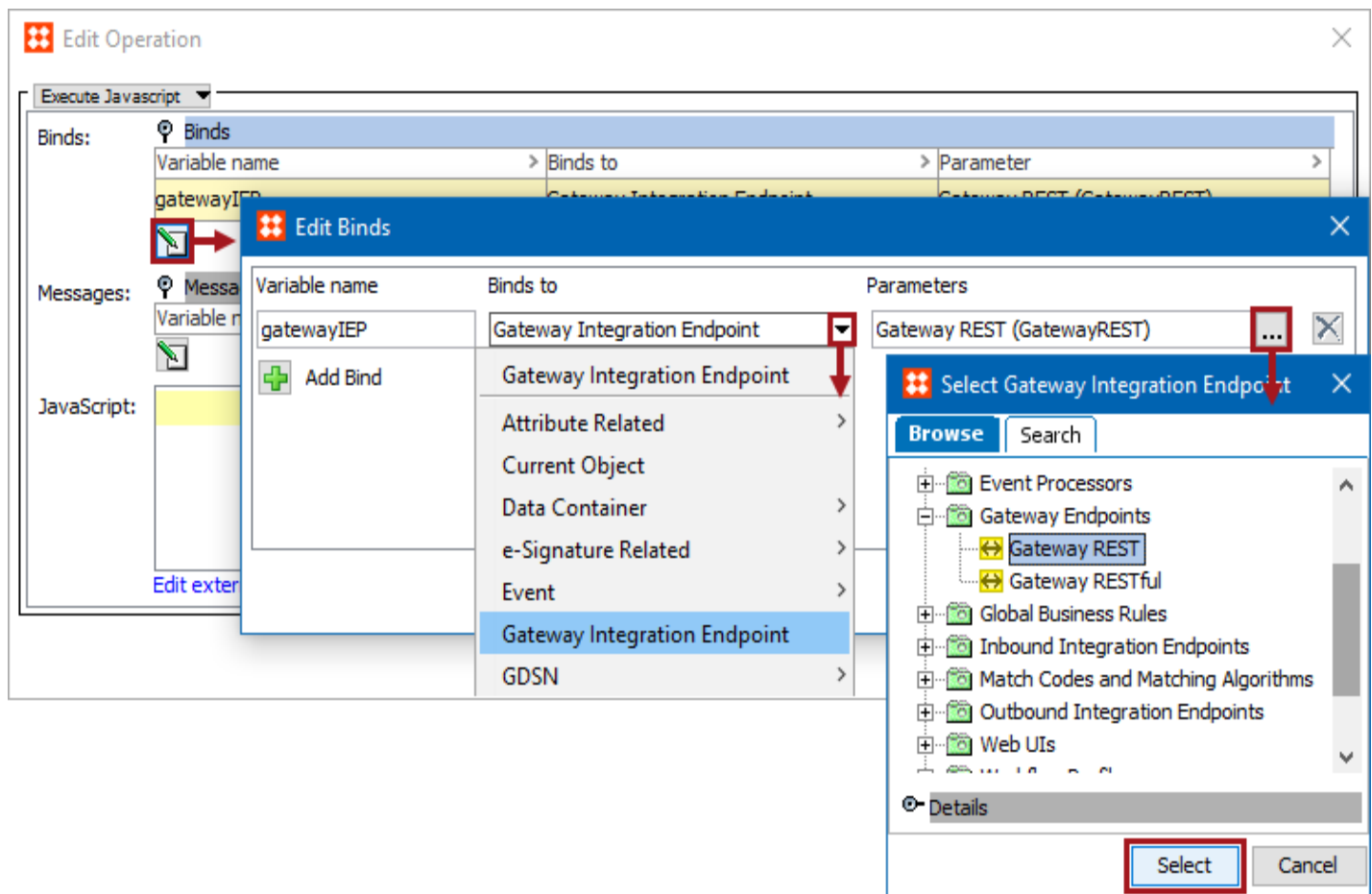


3. The IP of the proxy server being accessed
4. The port of the proxy server being accessed
5. The username for the proxy server being accessed (optional)
6. The password for the proxy server being accessed (optional)

Gateway Integration Endpoint Bind

Gateway Integration Endpoints provide real-time inbound and outbound integration under STEP's control. The Gateway Integration Endpoint bind is accessed from JavaScript in business rule conditions and actions. The bind can work with the REST methods included in the **Gateway REST Methods** section below.

Note: A gateway IEP functions as an HTTP client and supports all of the standard HTTP methods. Therefore a sound basic understanding of the HTTP protocol is essential to be able to configure and use a gateway IEP.



Configuration

To use any bind:

1. Create a business rule as defined in the **Creating a Business Rule or Library** topic.
2. Edit the business rule as defined in the **Editing a Business Rule** topic.
3. In the Edit Operation dialog, add the bind to a business rule, as defined in the **Adding a Bind** topic.
4. In the Edit Operation dialog, optionally add Messages, as defined in the **Adding a Localized Business Rule Message** topic.
5. In the Edit Operation dialog, add JavaScript to call the bind.

Gateway REST Methods

This bind can work with the following REST methods. For more information about these methods, click the **STEP API Documentation** button on the STEP WebStart page.

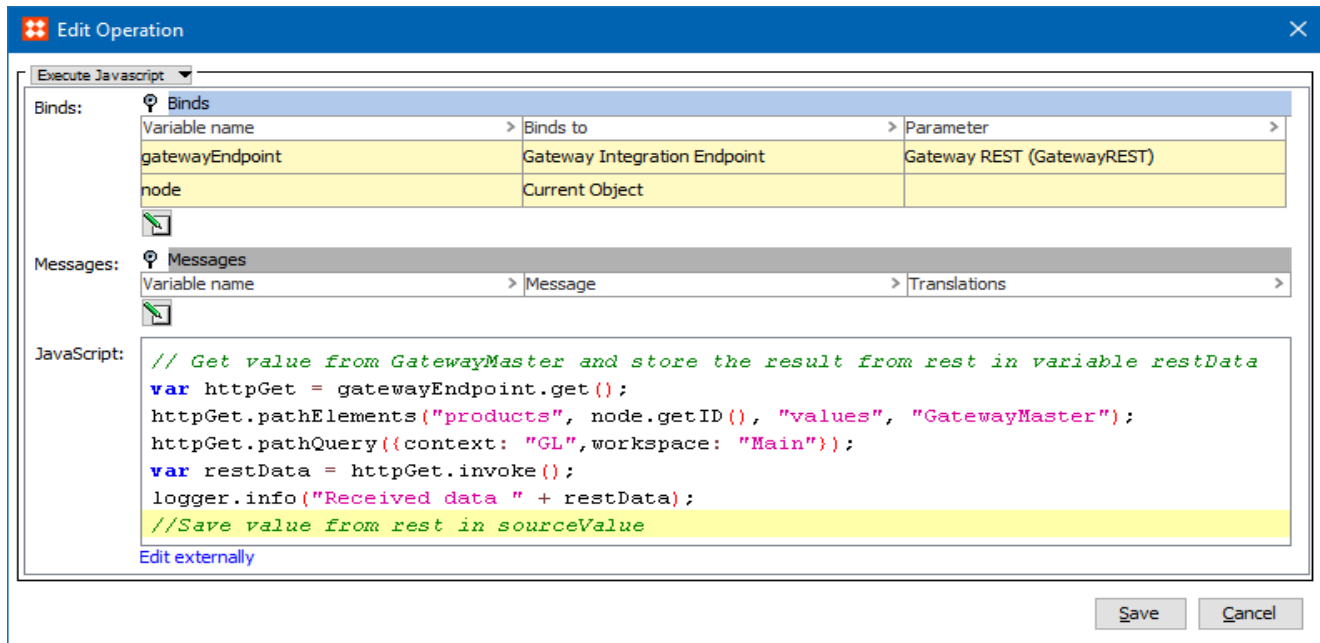
REST Method	Syntax Example
delete	<code>delete().path("products").pathQuery({ workspace: "Approved", context: "GL"}).body("Test").invoke();</code>
get	<code>get().path("products").pathQuery({ workspace: "Approved", context: "GL"}).invoke();</code>
head	<code>head().path("products").pathQuery({ workspace: "Approved", context: "GL"}).invoke();</code>
options	<code>options().path("products").pathQuery({ workspace: "Approved", context: "GL"}).invoke();</code>
post	<code>post().path("products").pathQuery({ workspace: "Approved", context: "GL"}).invoke();</code>
put	<code>put().path("products").pathQuery({ workspace: "Approved", context: "GL"}).body("Test").invoke();</code>

Example

The following is example JavaScript that uses this bind.

Important: The example scripts should not be used as-is without thorough testing, including updating the script to match object and link types that exist on your system.

The following example uses a REST gateway integration endpoint to access the external server for information.

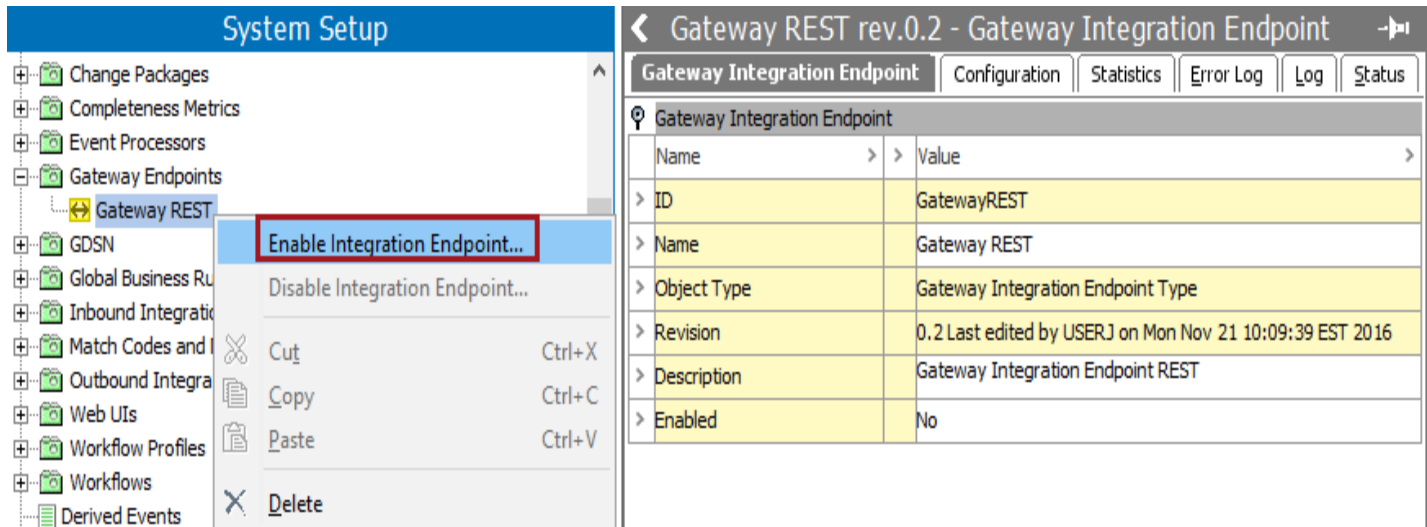


When the business action is executed, the gateway endpoint is accessed. The gateway endpoint Statistics tab displays an overview of executed REST methods. For more information, see the **Gateway Integration Endpoint Statistics** section.

Running a Gateway Integration Endpoint

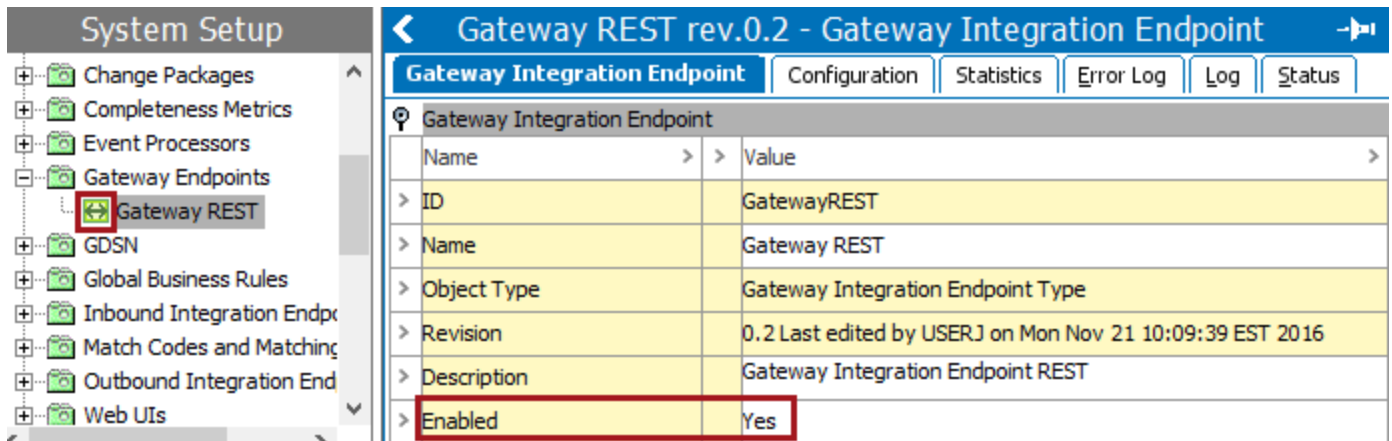
After configuring a gateway endpoint, it must be enabled and connected before it can run.

1. On System Setup expand the Gateway Endpoints node to display all existing gateway endpoints.
2. Select a disabled gateway endpoint (indicated by the yellow gateway icon  meaning it has been disabled by user request), right-click, and select **Enable Integration Endpoint** from the menu.



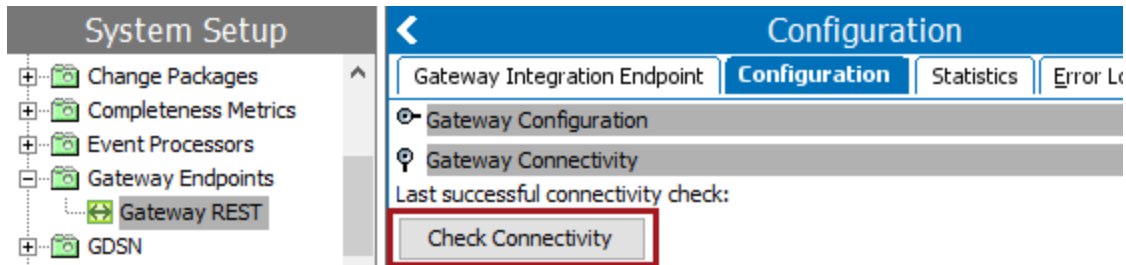
Name	Value
ID	GatewayREST
Name	Gateway REST
Object Type	Gateway Integration Endpoint Type
Revision	0.2 Last edited by USERJ on Mon Nov 21 10:09:39 EST 2016
Description	Gateway Integration Endpoint REST
Enabled	No

The tree navigator icon is now green () indicating it is active and ready to use. The Gateway Integration Endpoint flipper 'Enabled' parameter now displays 'Yes'.



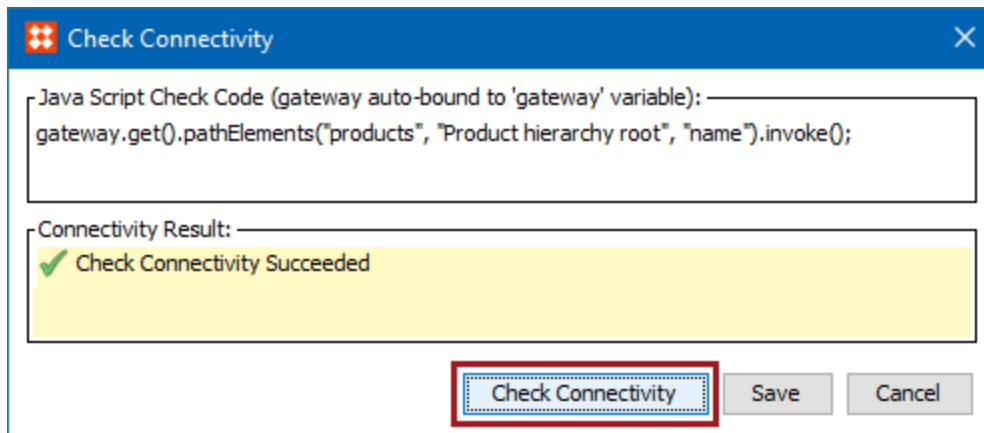
Name	Value
ID	GatewayREST
Name	Gateway REST
Object Type	Gateway Integration Endpoint Type
Revision	0.2 Last edited by USERJ on Mon Nov 21 10:09:39 EST 2016
Description	Gateway Integration Endpoint REST
Enabled	Yes

3. On the Configuration tab under the Gateway Connectivity flipper, click the **Check Connectivity** button.



4. In the **Check Connectivity** dialog, add JavaScript to access the external system.

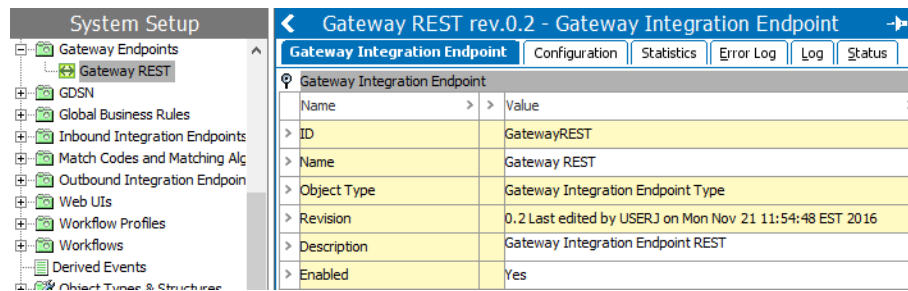
In the following example, the JavaScript gets the name of the "Product Hierarchy root". For information about the JavaScript syntax, click the **STEP API Documentation** button on the STEP WebStart page.



5. Click the **Check Connectivity** button to display the connectivity result.

Maintaining a Gateway Integration Endpoint

Gateway integration endpoint settings can be viewed and edited from the Gateway Integration Endpoint editor.



The editor has the following tabs:

- **Gateway Integration Endpoint:** holds basic information like ID, Name, and Enable status.
- **Configuration:** holds information about the external server, user credentials and allows you to test the connection to the external server. Also allows editing of this information via the Edit link in the Gateway Configuration flipper. For more information, see the **Configuring a Gateway Integration Endpoint** section.
- **Statistics:** provides statistics about frequency and timings of external calls using the gateway. For more information, see the **Gateway Integration Endpoint Statistics** section.
- **Error Log:** shows errors reported from external calls using the gateway. For more information, see the **Monitoring a Gateway Integration Endpoint** section.
- **Log:** provides information about changes to the gateway configuration.
- **Status:** provides information about revisions, hidden values and diagnostics.

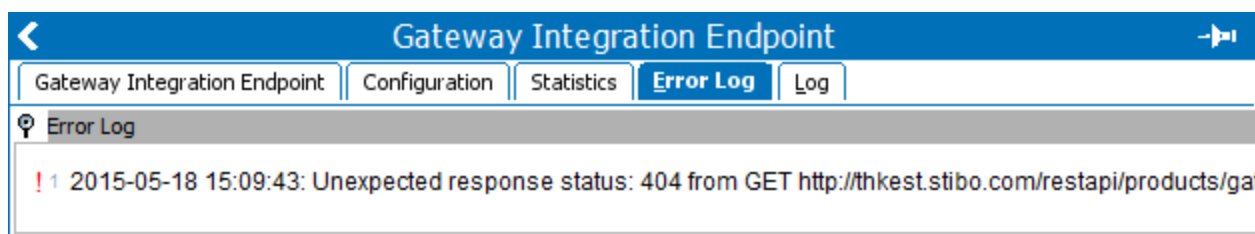
Monitoring a Gateway Integration Endpoint

Errors may occur when business actions are used to access gateway endpoints. Typical errors include not being able to access the external system or the external system not containing the data being requested in the business action. Errors can be monitored from within STEP or by directly accessing the endpoint via a URL.

Monitor from STEP

Any unexpected response status is received and logged in the in the Error Log tab on the gateway integration endpoint. This provides an overview of all logged errors on the gateway integration endpoint.

In the following example, an unexpected response status is logged when a method has been executed to retrieve data on an object which did not exist on the external system.



Monitor Outside of STEP

The following URLs allow you to get an overview of the status and any errors on processes without logging onto STEP.

View all endpoints

Use the following URL, replacing <appserver> with your own application server name.

```
http://<appserver>/admin/monitoring
```

For example:

```
http://myappserver/admin/monitoring
```

Endpoints are listed by name in the Sensor column of the list displayed.

View a specific gateway endpoint

Use the following URL, replacing <appserver> with your own application server name, and replacing <Gateway-ID> with the STEP ID of the desired endpoint. The ID is displayed in the gateway integration endpoint editor. For more information, see **Maintaining a Gateway Integration Endpoint**.

```
http://<appserver>/admin/monitoring/GatewayIntegrationEndpointStatus-<Gateway-ID>
```

For example:

```
http://myappserver/admin/monitoring/GatewayIntegrationEndpointStatus-GatewayEndpointCheckConnection
```

A report similar to the one below is displayed for the identified gateway endpoint with the ID 'GatewayEndpointCheckConnection'.

Sensor status for GatewayIntegrationEndpointStatus-GatewayEndpointCheckConnection

Plugin GatewayIntegrationEndpointStatus
Sensor GatewayEndpointCheckConnection
Status Warning
Created Thu May 21 11:01:02 CEST 2015 (28 seconds ago)
TTL 30 seconds
Short message Endpoint is enabled, and has errors - but has no recent errors since 2015-05-20 11:01:02
Long message The endpoint has errors:
 2015-05-18 15:09:43: Unexpected response status: 404 from GET http://thkj-test.stibo.com/restapi/products/gatewayproductdoesnotexist/values/Description?context=GL&workspace=Main http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"/ Error 404 Not Found HTTP ERROR 404 Problem accessing /restapi/products/gatewayproductdoesnotexist/values/Description. Reason: Not Found / Powered by Jetty://

Gateway Integration Endpoint Statistics

The Statistics tab includes information for executed REST methods grouped by the HTTP request types (GET, POST, PUT, and so on).

Gateway Integration Endpoint

Configuration

Statistics

Error Log

Log

Performance Statistics

Load

1 hour

From

2015-05-18 15:39:56

...

To

2015-05-18 15:39:56

...

Description	Minimum duration	Average duration	Maximum duration	Max. duration URL	Total duration	Invocations
DELETE */products/. *	68 ms	68,00 ms	68 ms	http://thkest.stibo.com/re...	0,07 s	1
GET */products/. *	135 ms	135,00 ms	135 ms	http://thkest.stibo.com/re...	0,14 s	1
POST	57 ms	57,00 ms	57 ms	http://thkest.stibo.com/re...	0,06 s	1
POST */products/. *	54 ms	54,00 ms	54 ms	http://thkest.stibo.com/re...	0,05 s	1
PUT	29 ms	29,00 ms	29 ms	http://thkest.stibo.com/re...	0,03 s	1

For more granular groupings, regular expression can be used to display individual REST calls. This is set up on the Configuration tab > Gateway Configuration flipper > 'Statistic groups' parameter. For more information, see the **Statistic groups** section of the **Configuring a Gateway Integration Endpoint** documentation or the **Regular Expression** topic in **System Setup / Super User Guide** documentation.

Exporting Gateway Integration Endpoint Definitions as Comments

Integration endpoint definitions can be exported as comments using Advanced STEPXML. These exports are intended to be used for submission to external source control systems for comparison purposes. Users can import them into source code repository systems where they can be compared from version to version. Editing and/or importing of these files is not supported (e.g., users may not export, edit the comments, and re-import in STEP).

To export an integration endpoint definition for external comparison, use Advanced STEPXML and set the DefinitionsAsComments tag to 'true', as shown in the following sample template:

```
<?xml version='1.0'?>
<STEP-ProductInformation DefinitionsAsComments="true">
<OutBoundIntegrationEndpoint ExportSize="All"/>
</STEP-ProductInformation>
```

An example of the output for an integration endpoint definition as comments is shown below:

```
<OutBoundIntegrationEndpoint...>
<!--Definition: [This will be the IEP definition. Removed for brevity. -->
[Remaining configuration is included]
</OutBoundIntegrationEndpoint>
```

Note: The content of the comment field is not part of the STEPXML XSD and therefore Stibo Systems reserves the right to change the format of the output content at any time.

Additionally, integration endpoints can be exported using STEPXML. For information, see the Configuration > **Include Integration Endpoints** section of the **STEPXML Outbound Parameters** documentation.

Import Manager

The Import Manager allows you to import data on demand and to save or modify an import configuration. The final step of the wizard allows you to save the import configuration. This is useful for frequently-used imports where the parameters and options are always, or nearly always, the same.

Importing formats that require mapping gives access to a number of data quality features using transformations. The mapping process allows you to spot errors in the data before importing, to trim white space, or otherwise manipulate data to meet your specifications. Checking the data prior to import can prevent background process failures due to invalid data formatting.

Setup Requirements

Setting up and using the Import Manager involves the following steps:

1. Based on the data being imported, determine how to start the Import Manager wizard as described in **Creating a Data Import**.
2. In the wizard, choose a previously saved configuration or start a new import as described in **Import Manager - Select Configuration**.
3. In the wizard, select the data file to be imported as described in **Import Manager - Select Data Source**.
4. In the wizard, select the format of the data file, which determines the import method as described in **Import Manager - Select Format**.
5. In the wizard, when required, select which pieces of information are imported as described in **Import Manager - Map Data**.
6. In the wizard, when required, specify if imported data is new or if it matches data already in STEP as described in **Import Manager - Identify Objects**.
7. In the wizard, specify how and where information is to be imported as described in **Import Manager - Identify Destination**.
8. In the wizard, specify business conditions and/or actions that affect the import as described in **Import Manager - Select Business Rules**.
9. In the wizard, when required, select a context and specify if imported data should be modified as described in **Import Manager - Advanced Settings**.
10. In the wizard, save the configuration (if necessary), and run the import as described in **Running a Data Import**.
11. Monitor the import and download the output file as described in **Monitoring a Data Import**.

Additional Information

The following information is useful once an import configuration is saved:

1. Maintain a saved configuration as described in **Maintaining a Saved Import Configuration**.

Creating a Data Import

The Import Manager wizard allows you to import data on demand and to create or modify an import configuration. The final step of the wizard allows you to save the import configuration. This is useful for frequently-used imports where the parameters and options are always, or nearly always, the same.

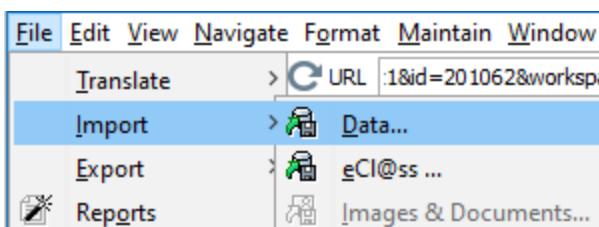
The user running the import should have the following setup privilege action sets: Create domain, Create value in hard domain, Create value in medium domain, Delete domain, Merge entire domain, and View domain. For more information, see **Action Sets** in the **System Setup / Super User Guide** documentation.

To import data using the Import Manager:

1. Verify that all attributes and reference / link types being imported already exist in STEP.
2. Use one of the following methods to launch the Import Manager wizard, noting that using an import configuration pre-loads the import file and settings.

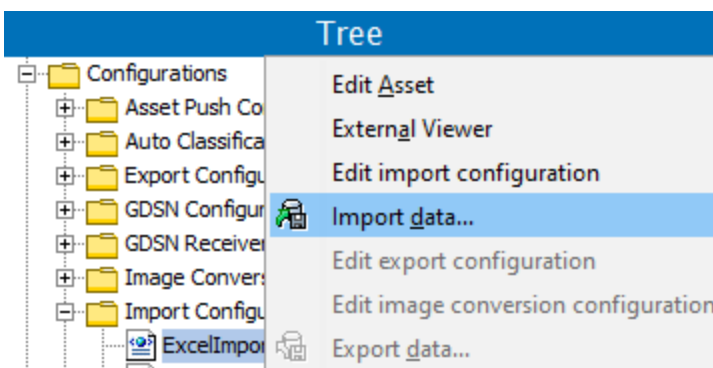
- **From the File Menu:** Click the File menu > Import > **Data**.

The wizard opens in Step 1 and allows you to select an existing configuration or create a new one.

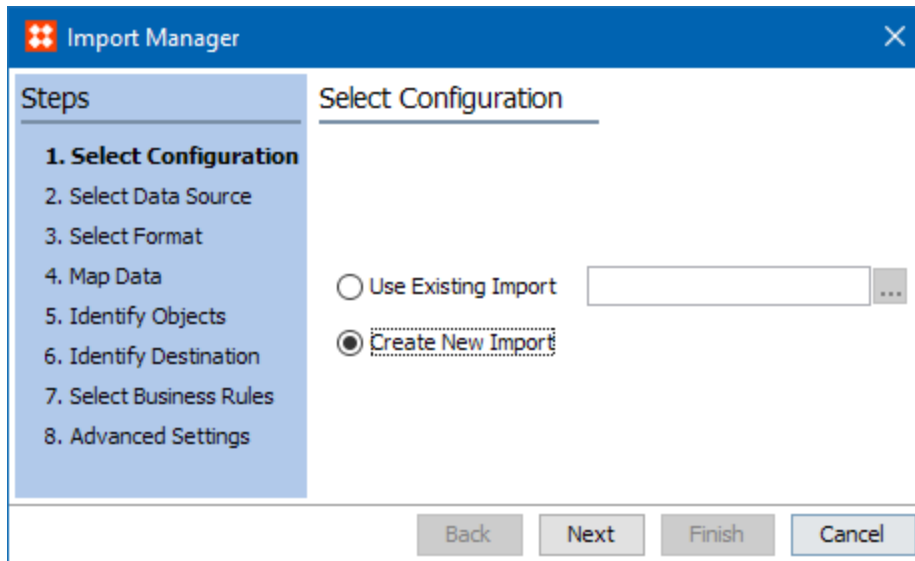


- **From an Import Configuration:** On the Tree, select an import configuration, right-click, and then select **Import Data**.

The wizard opens in Step 2 with the configuration and objects included in the configuration already selected.

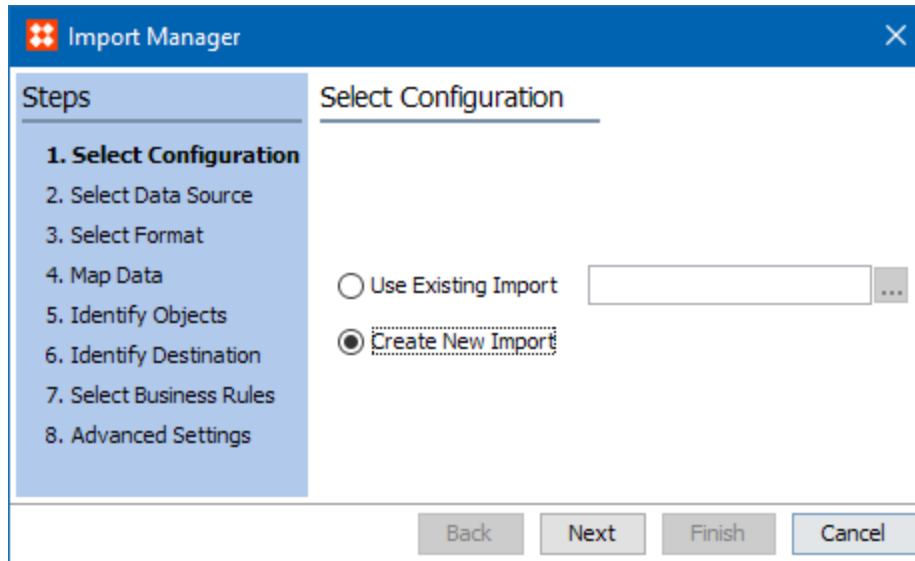


3. The Import Manager wizard displays and can involve the following steps:



- **Select Configuration** allows you to specify an existing import configuration or create a new one. If you frequently load data files manually, and they have the same data in the same columns, you can save a configuration with all necessary settings.
 - **Select Data Sources** allows you to select the source of the data. If the source is a file, open a browser and select the file that you want to use. You can also choose an FTP connection.
 - **Select Format** allows you to select the type of file to import and the basic parameters of the file.
 - **Map Data** when required, allows you to map the columns of data in the import file to the equivalent fields in STEP.
 - **Identify Objects** allows you to confirm if objects in the input file are new or existing and specify to locate objects by attribute value (when the file does not contain STEP IDs), by the object ID, or by name.
 - **Identify Destination** allows you to specify a default object type and parent for new objects.
 - **Select Business Rules** allows you to select business conditions and business actions to test / execute in relation to the data being imported. Data to be imported is processed based on the selected business rules.
 - **Advanced Settings** allows you to set infrequently used operations, such as removing attribute values that are not listed in the input file or setting the approval status of imported objects.
4. Follow the steps defined in **Running a Data Import** to start the import background process.

Import Manager - Select Configuration

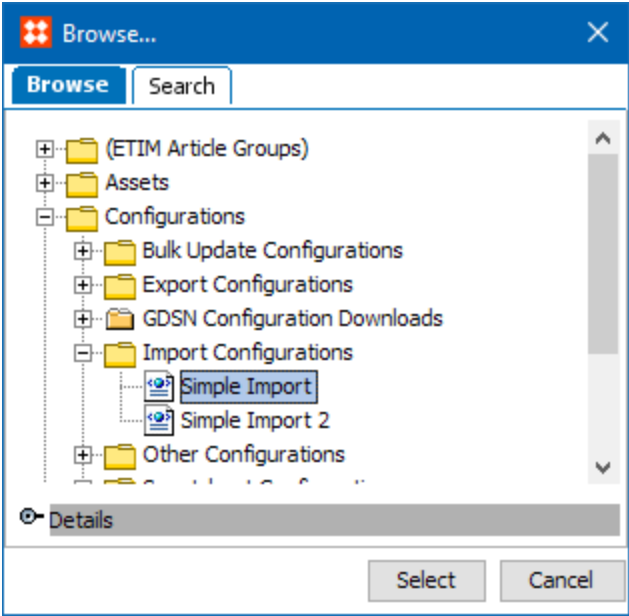


Once created, an import configuration can be saved as an asset in STEP for use with other import files. A configuration file contains an XML representation of the parameter settings in Import Manager.

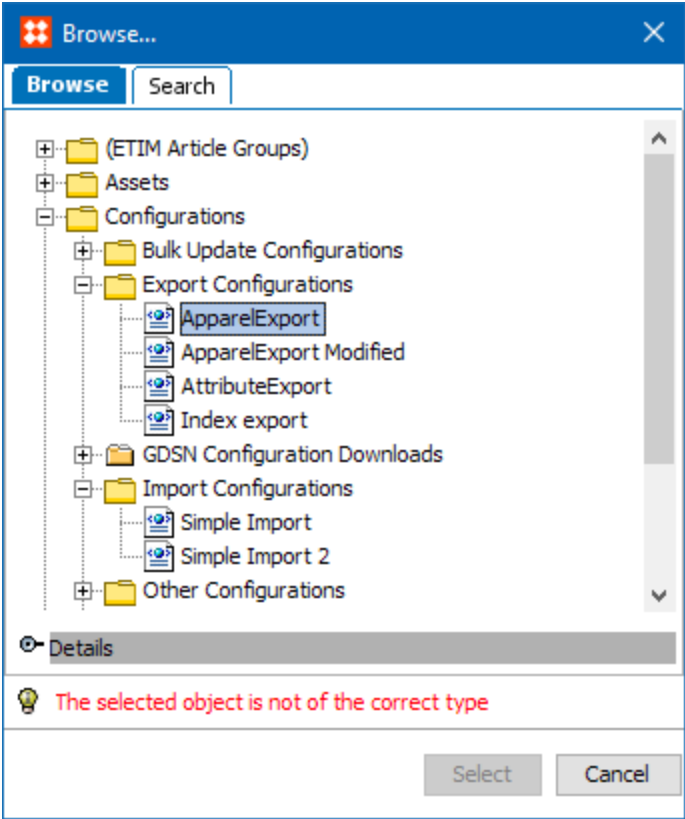
1. Select a configuration option:
 - **Use Existing Import** - allows you to select a previously saved configuration. Click the ellipsis button (...) to browse or search for the configuration in STEP.
 - To create a new configuration, Select **Create New Import**.
2. Click **Next** to display **Import Manager - Select Data Source** or when using an existing configuration, if no additional changes are necessary, click **Finish** to display Save Import Configuration dialog as defined in **Running a Data Import**.

Using a Saved Import Configuration

For example, if a user frequently loads data files that always have the same attributes in the same columns, they can store a configuration that has all the import options already set, including the mappings of the columns etc. This way, all the user has to do is select that configuration and (if necessary) point to the file they want to load.



Note: Selecting an incompatible configuration, such as a bulk update or export configuration, displays a message and the user must make a new selection.



Import Manager - Select Data Source

If you selected an import configuration in the previous step, the saved data is displayed. When creating a new import configuration, you must select the source from the following options. The steps for each are described below.

- **FTP** allows you to select a file from an FTP site.
- **File** allows you to select a file from your local hard drive or from your network.

Using FTP

Import Manager

Steps

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

Select Data Source

Data Source: **FTP**

Retrieves data from a remote server using the File Transfer Protocol (FTP).

Host: myserver.mycompany.com

User: myname

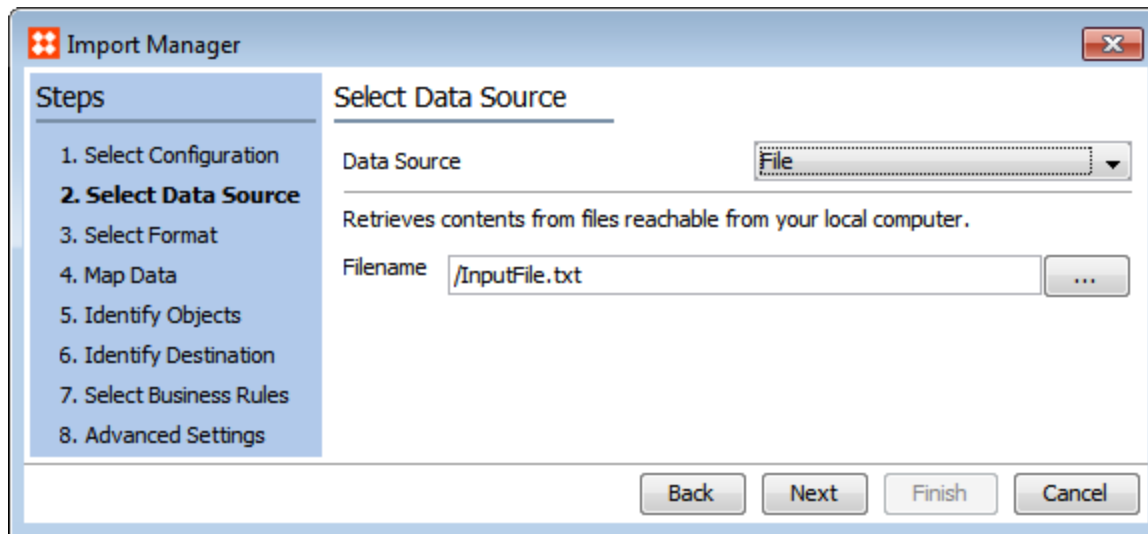
Password:

Filename: /myfile.txt

Back Next Finish Cancel

1. Select **FTP** from the Data Source list.
2. Enter the **Host** name, the **User** name, and the **Password** (if any).
3. Enter the **Filename** including the path, and then click **Next** to display **Import Manager - Select Format**.

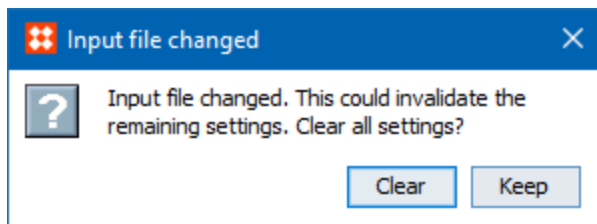
Using File



It is common practice to copy very large files to your local drive prior to importing.

1. Select **File** from the Data Source list and click the ellipsis button (...) on the Filename field.
2. Locate the relevant file, and then click **Select File**, and then click **Next** to display **Import Manager - Select Format**.

When using a previously saved configuration (described in **Running a Data Import**), selecting a file that is different from the one saved in the configuration can result in invalidating the configuration settings. The following dialog allows the user to proceed with the saved settings, or clear them.



- Click **Clear** to remove the configuration settings and then reset them manually. Use this option, for example, when the configuration expects a Fixed Width file but the newly selected file is an Excel file.
- Click **Keep** to apply the saved configuration settings to the new file. Use this option, for example, when the new file selected is in the layout expected by the configuration but the data is different.

Import Manager - Select Format

Import Manager

Steps

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

Select Format

Format Excel

Converter for files in Microsoft Excel format that contain one object per row.

Has Header yes

Trim whitespace yes

Use date and number formatting from sheet yes

Conversion Preview:

<ID>	> <Parent ID>	> <Primary Color>	> <Secondary Color>	>
Mens T PBO	18209	Black	Orange	
Mens T PBG	18209	Blue	Green	
Mens T PGS	18209	Green	Silver	
Mens T PGW	18209	Gray	White	
Mens T POY	18209	Orange	Yellow	

Back
Next
Finish
Cancel

Select the format of the file manually if it is not automatically detected by the Import Manager. Depending on the format selected, a number of additional settings become available. Before proceeding, some formats populate the Conversion Preview section to allow for visual inspection of the data.

The following formats are available:

- Advanced STEPXML Format
- BMEcat Format
- BMEcat 2005 Format
- CSV Format
- eCI@ss Format
- ETIM Format
- ETIM 6 Format
- ETIM v2 Format
- Excel Format
- Excel Smartsheet Format

- FixedWidth Format
- Flatplanner Format
- Generic XML Format
- Global Product Classification Format
- IDoc MATMAS 05 Format
- Publication Excel Format
- STEPXML Format
- UNSPSC Format

Click the **Next** button to display **Import Manager - Map Data** when required, or **Import Manager - Identify Objects** if mapping is not needed.

Import Manager - Map Data

Import Manager

Steps

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

Map Data

Source:

<Name>	<Parent ID>	Primary Color	Secondary Color
Mens T PBO	18209	Black	Orange
Mens T PBG	18209	Blue	Green
Mens T PGS	18209	Green	Silver
Mens T PGW	18209	Gray	White
Mens T POY	18209	Orange	Yellow

Result:

Map to: **Product**

Name = <Name> ✓	Parent = <Parent ID> ✓	PrimaryColor = Primar... ✓	SecondaryColor = Se... ✓
Mens T PBO	Cotton T-Shirts	Black	Orange
Mens T PBG	Cotton T-Shirts	Blue	Green
Mens T PGS	Cotton T-Shirts	Green	Silver
Mens T PGW	Cotton T-Shirts	Gray	White
Mens T POY	Cotton T-Shirts	Orange	Yellow

Mapping is required when importing some file types. Mapping relates the import data to the appropriate element in the STEP data model and specifies the desired object type.

The mapping step for Import Manager and IIEPs include the same options. In addition to mapping data, additional functionality is available that allows you to modify data and generate a profile to verify the quality of the inbound data.

For information about the formats that require mapping and all of the available options, see the **Inbound Map Data Options** section.

Click the Next button to display **Import Manager - Identify Objects**.

Import Manager - Identify Objects

Import Manager

Steps

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

Identify Objects

Source:

ID = <ID>
114852
114854
111204
114859
114440
888264

Result:

Match Result	ID = <ID>
114852	114852
114854	114854
111204	111204
114859	114859
New	114440
New	888264

Match Remove

Back Next Finish Cancel

Identify Objects allows you to verify if loaded objects are existing data in STEP or will be created as new objects. Additionally, you can specify to locate existing objects by attribute value rather than by the ID or name.

Importing With Object IDs

When the ID has been mapped, the Result panel displays the first 20 results:

- Objects found in STEP are displayed in green
- New objects display 'New' in the Match Result column

You cannot introduce new products by using Identify Objects (in this step). When importing with object ID, all products must exist already in STEP. If there is no match of the attribute value to a Object ID, the record is ignored.

You can select any attribute that you have already mapped in the prior Map Data screen, but you can only match on one attribute. For example, if you have several attribute columns that are 'alternate part numbers' or 'replacement part numbers', you cannot select multiple attributes at one time in order to find the actual Object ID in STEP.

Matching can only be done on one attribute at one time. However, you can try several columns on the same import by using the Remove button and by selecting another attribute column in the top panel.

Click the Next button to display **Import Manager - Identify Destination**.

Importing with Unique Key

Note: Unique Key must be activated before the import process is started.

When the key has been mapped, the Result panel displays the first 20 results:

- The Result pane shows a column respective IDs for the unique key.
- Objects found in STEP are displayed in green.

Import Manager

Steps

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

Identify Objects

Source:

ID KeyID=SupplierPartNumber = <Unique Key>

AJJKF22

AJJKF23

668998643

AAJJKF24

Result:

Match Result	ID KeyID=SupplierPartNumber = <Unique Key>
114852	AJJKF22
114854	AJJKF23
111204	668998643
114859	AAJJKF24

Match Remove

Back Next Finish Cancel

Importing Without Object IDs

If new objects without IDs are to be imported, an ID Pattern must be set for the target object type. Otherwise the objects cannot be created.

You can select any column that you have already mapped in the prior screen, the Map Data screen, but you can only match on one column. So if you have several columns that are 'alternate part numbers' or 'replacement part numbers', etc. you cannot select multiple columns at one time in order to find the actual Object ID in STEP. Matching can only be done on a column at once. However, you can try several columns by using the 'Remove' button and by selecting another column in the top panel.

Identifying Objects by Attribute Values

When matching on an attribute other than STEP ID, prior to running live imports, test typical data to determine if performance is negatively affected.

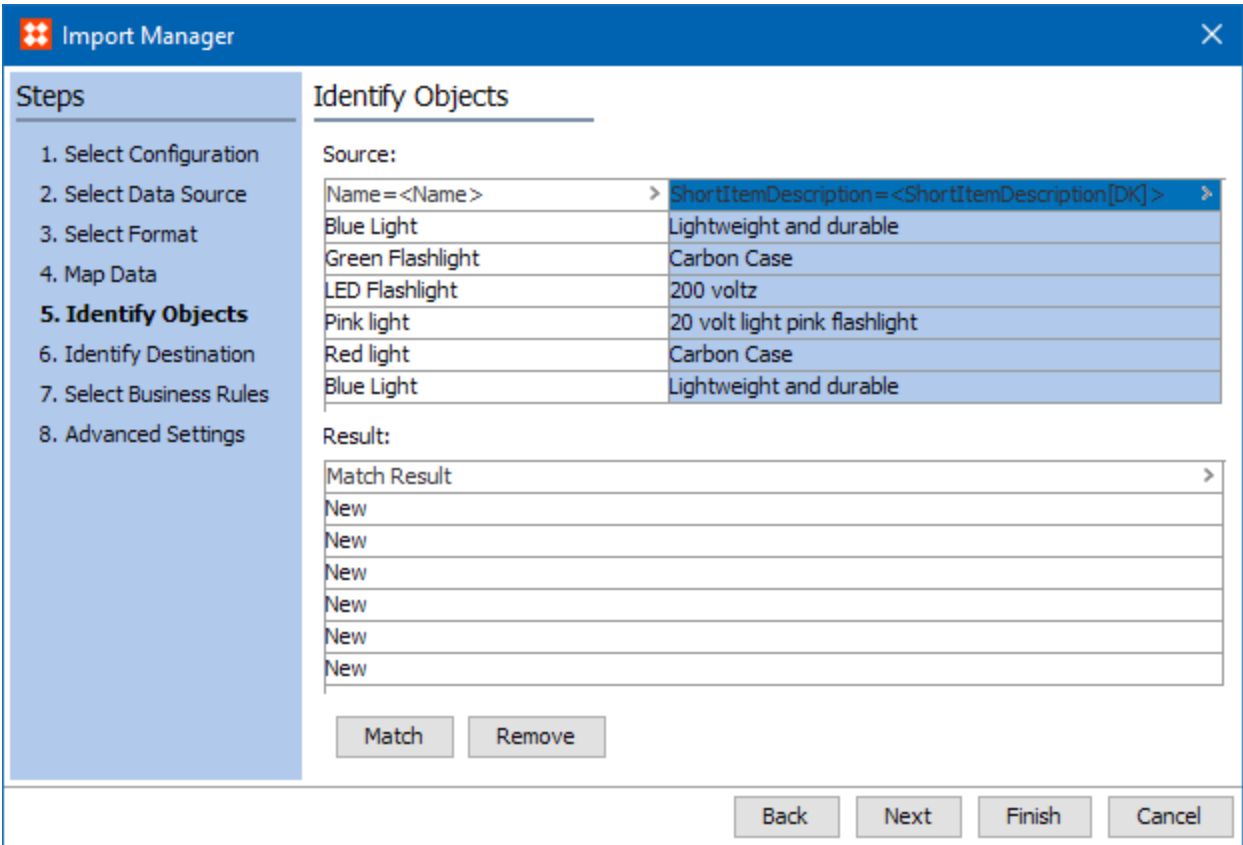
The following example describes how to identify products by attribute values. The product data is loaded from a file with no product IDs listed but contains the manufacturer’s own part numbers.

Important: New products are not loaded to STEP when they are located by attribute values. Even if you have set up the system to auto-generate an ID for new products, and even if you allow new products to be created, the system rejects any new products in this file when the products are located by attribute values.

Normally, the manufacturer part number would be an attribute of your own product. If the manufacturer’s part number is unique, you can then load the file by having the system locate the products by an attribute value – the manufacturer’s part number - instead of mapping a column of data to the product ID as you would normally do.

A file with two columns of data is imported to STEP. Both columns are mapped to attributes. The first one is mapped to the attribute 'Manufacturer Part Number', and the second is mapped to 'Description'.

- 1. Select a column in the top panel, and click **Match**. The system attempts to locate products by the values in that attribute.



Note: The Match button is only enabled if you have not mapped a column to a product ID in the Map Data step. STEP assumes that if you defined a column of data as the ID, no matching of attribute values to Objects IDs is required.

If the match is successful, the result is displayed in green in the **Results** pane.

Import Manager

×

Steps

1. Select Configuration

2. Select Data Source

3. Select Format

4. Map Data

5. Identify Objects

6. Identify Destination

7. Select Business Rules

8. Advanced Settings

Identify Objects

Source:

Name = <N... >	ShortItemDescription = <ShortItemDescription[DK] >
Blue Light	Lightweight and durable
Green Flashli...	Carbon Case
LED Flashlight	200 voltz
Pink light	20 volt light pink flashlight
Red light	Carbon Case
Blue Light	Lightweight and durable

Result:

Match Result	Name = <Name >
114852	Blue Light
New	Green Flashlight
111204	LED Flashlight
114859	Pink light
New	Red light
114852	Blue Light

Match

Remove

Back

Next

Finish

Cancel

The first column is the ID of the product in STEP. The second column is the attribute value that matches that product.

Match only checks the first 20 products in the file. If it is unable to find a match it the result in the first column is 'New'.

2. Click the Next button to display **Import Manager - Identify Destination** or click **Finish** if you have made all required selections and want to launch the import process as defined in **Running a Data Import**.

Import Manager - Identify Destination

- **Approver** is used to specify the user name of the person that is allowed to approve the products after the import process is complete. When you click the ellipsis button (...), you can search or browse for the user. You are the default Approver. Normally, this is used in conjunction with a customized workflow. See also the 'Approve Import Changes' option on the Advanced Settings step in this wizard.
- **Import Workspace** is used to specify which workspace the data is loaded into. You can use this feature if you have set up more than one maintenance type of workspace in your system, such as 'Staging'. The default is Main. You can never import data into the Approved workspace
- **Default Parent Product** is used to specify where new products should be placed in STEP. **A parent is required to create new objects, entities, assets, and classifications in STEP.** Use this parameter if no column has been mapped to the Parent ID, or if the entry in the Parent ID column is invalid (for example, if the parent product listed in the input file does not exist in STEP). The import does not check all of the records in the load file. This means that if any of the records in your load file is a new product, and does not have a parent, it cannot be created in STEP. Similarly, even if you have mapped a column in the load file as the Parent ID, the import has not validated all those entries before this point. When parent data is missing in the import file, it can sometimes be derived via transformations or added by mapping a constant value in the Map Data step.

- **Default Object Type** is used to specify which object type to use when there are new products in the load file. **An object type is required to create new objects, entities, assets, and classifications in STEP.** Valid object types are based on the selected default parent product. It is not unusual to have different object types for categories, subcategories, families, or individual items. It is also not unusual for objects to have several possible object types, depending on your system setup. Common setup is to use this feature when all products in the load file are of the same object type.
- **Batch Directory** is an obsolete option and selecting a directory has no effect on the import. Instead of using this option to link products to a classification folder, use the Map Data step to link a product to one or more classifications with the specified Reference / Link Type. For more information on importing to classifications, see the **Map Product Classification Links** section of the **Inbound Map Data - Map** documentation.
- **Test Import Only** is used to load a small file into STEP without saving or committing any changes. No data is imported into STEP when this option is checked. The Import Manager wizard goes through all the steps of importing the file, but does not perform the final step of confirming the changes in STEP. This allows you to review the execution report so that you address any warnings or errors encountered when loading the file. You can then make appropriate changes to the load file and do the load again, until all errors have been resolved.
- **Reject New** is used to specify a reject to any new products that are listed in the load file. If selected, new products are not loaded or created. The execution report lists the products that have been rejected as a result of this selection. This is useful, for example, when loading price files.
- **Reject Updates** is used to specify whether only new products should be imported from a load file that contains both new and existing products. This way you avoid updating existing products in STEP. Existing products are left untouched.

Click the **Next** button to display **Import Manager - Select Business Rules**.

Import Manager - Select Business Rules

Import Manager

Steps

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

Select Business Rules

Reject Objects if the following conditions are not met:

Name	Description
Object Damaged	

> Add condition

Execute these actions for each valid Object:

Name	Description
Object For Review	

> Add action

Back Next Finish Cancel

Import data is processed based on any selected business rules. All business rules that exist on your system are available for selection. First, all business conditions are validated in order against objects during the import process, and then all business actions are executed in order on the valid objects. If a business rule results in an error, the import process skips the object and logs to the Import Execution Report. For more information, review the business rule limitations below.

1. If necessary, click the **Add Condition** link to add conditions. Then use Browse or Search to locate the condition, and click **Select**. Repeat until all the relevant conditions are added.

Add condition

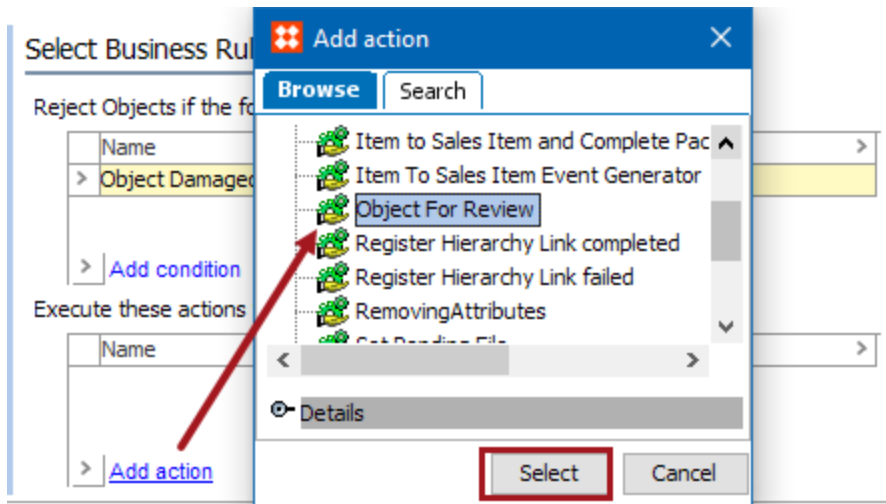
Browse Search

- Donate Shoes
- Ignore Buy Side Objects
- NewCondition
- Object Damaged
- TestCondition
- TranslateMsgC

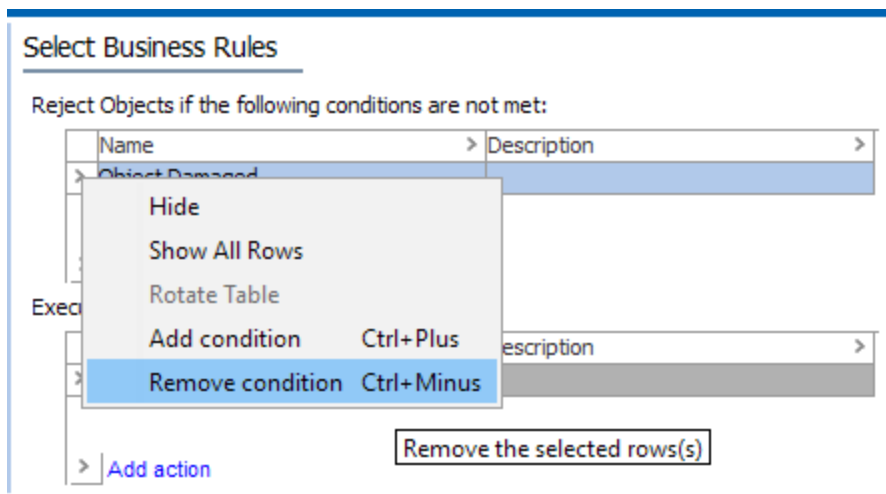
Details

Select Cancel

- If necessary, click the **Add Action** link to add actions. Then use Browse or Search to locate the action, and click **Select**. Repeat until all the relevant actions are added.



- If necessary, to remove a **condition**, select it, right-click, and then select **Remove condition**.



- If necessary, to remove an **action**, select it, right-click, and then select **Remove action**.
- Click **Next** to display **Import Manager - Advanced Settings** or click **Finish** to launch the import process as defined in **Running a Data Import**.

Business Rule Limitations

- Objects must be imported one at a time for the validation mechanism to work. It is therefore not possible to import nested STEPXML documents when you are using business rules.
- If one or more business rules are selected, the Import Manager uses domain mode.
- All selected business rules are run in sequence, from top to bottom.

- Changes are not detected for references and links. Objects with references in the import are therefore always reported as changed.
- Sometimes imported references are deferred if they depend on objects that have not been imported yet. As a consequence, the deferred parts of an imported object are not present while it is being validated. Therefore, use caution when using business conditions to validate references.
- When actions are used, the 'Approve Import Changes' on Advanced step is disabled - importer cannot automatically auto approve imported data because side effects from business actions are unknown. However it is still possible to approve imported objects via a business action.

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

Import Manager - Advanced Settings

Import Manager

Steps

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

Advanced Settings

Context
Danish DK

Import Process Template:
Importer

Match Units By
Name

Approve Import Changes
☐

Auto-Initiate STEP Workflows on Item Creation
☒

Trigger STEP Workflow import events on Item Updates
☐

Suppress Re-Translations
☐

Auto select Import Mode
☐

Force Database Import
☐

Force Domain Import
☒

Remove Options

Remove Un-Mapped Values

Attribute Group
>

> [Select Attribute Group\(s\)](#)

Remove Un-Mapped Data Containers

Data Container Type
>

> [Select Data Container Type\(s\)](#)

Remove Un-Mapped References

☐ Classification References

Product Reference Type
>

> [Add Product Reference Type](#)

Asset Reference Type
>

> [Add Asset Reference Type](#)

Entity Reference Type
>

> [Add Entity Reference Type](#)

Classification Product Li...
>

> [Add Classification Product Link Type](#)

Back

Next

Finish

Cancel

The Advanced Settings screen is only available in licensed versions. It provides a series of options that are not required on a day-to-day basis for data imports. Options are displayed as disabled, or are hidden completely, when the user running the import does not have valid permissions to run the option.

Important: Advanced Settings options are generally used to clean up data within STEP. However, if you are not careful, you can cause data integrity issues. Double-check your load file and your System Setup before using any of these options. Common setup is to perform a test on a test server first, and to start testing with small files to verify the results before changing large amounts of data.

After setting the options as defined below, click the **Finish** button to display the Save Import Configuration dialog as described in **Running a Data Import**.

General Advanced Settings

- **Context** - If you are creating a new configuration, the context shown is the one that is currently selected in STEP. That is usually the safest way to import data. You can, however, select a context different than the currently select one. If you selected an existing configuration, the context that was set up in that configuration is shown. Dimension Point specific mappings will overrule this setting. Consider the context setting when creating an Import Configuration that will be used repeatedly. You can only import to one context at a time, so if you have several languages, you will have to do several imports.

Note: If there are transformations applied in the Map Data step, to import the data in different contexts, the same will still be applicable. Additionally, when importing ETIM6, you selected a Language in the Select Format step. Verify that you choose the Context that should be updated with the selected language.

- **Import Process Template** - This is used to specify a workflow process. The default is Importer, which is the standard workflow.
- **Match Units By** - When the system loads data, it has the capability of trying to separate actual values from their units in the load file. Use this option to indicate if units in the data file should be matched on the unit's ID or the unit's name (default).
- **Approve Import Changes** - This option is used to approve the changes that the import causes. When checked, the import mode option 'Force Domain Import' is automatically selected and cannot be changed. If business rules are applied from previous step this option cannot be selected - instead it is possible to approve entire object from business rules.
- **Auto-Initiate STEP Workflows on Item Creation** - If a workflow is valid for either products, classifications, assets, or entities, it is possible to have an instance of a given workflow created whenever an object of the valid object type(s) is created. That is, a workflow is automatically started when the object is created. This functionality is available for workflows set to Auto Initiate on Object Creation. Users with the privilege Disable STEP Workflow Auto-initiation in Imports are allowed to disable auto-initiation in the Import Manager for tabular imports, or directly in the XML for XML imports by setting AutoInitiate="N" in the STEP-ProductInformation tag.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<STEP-ProductInformation CContextID="Context1" WorkspaceID="Main"AutoInitiate="N"
```

- **Trigger STEP Workflow import events on Item updates** - This option controls whether or not the import triggers events. In XML, the functionality is enabled / disabled using the STEP-ProductInformation tag attribute 'STEPWorkflowImportEvent' that can be set to "Y" or "N". The option is typically used in imports based on pre-configured Import Configurations. When checked, the import mode option 'Force Domain Import' is automatically selected and cannot be changed.
- **Suppress Re-Translations** - Checking Suppress Re-Translation, prevents products, classifications, entities or assets from getting the status **Re-Translation Needed**. This option only has effect if the data import is set to be approved too.
- **Import Mode options** - If the previous two options 'Approve Import Changes' or 'Trigger STEP Workflow import events on Item Updates' are selected, these options are disabled and cannot be modified.

Important: The following options are displayed and enabled based two factors: the privileges of the user running the import and the content of the import. When importing products, the user must have all user privileges to the product hierarchy root (without any restrictions). When importing entities, the user must have all user privileges to the entity hierarchy root (without any restrictions). Restrictions could include some attribute groups, or dimension points, or having limited privileges assigned to any of the object type being imported.

Auto Select Import Mode - Choose this option to allow STEP to analyze the import file and its size and run either Domain or Database import mode, as appropriate.

Force Domain Import - Access of the user who started the import to the hierarchies or attributes being modified is verified during the import. Generally, a more detailed execution report is delivered. If access is not allowed, the import fails and the error "Not privileged to change value ..." is included in the import execution report. Since this option checks all user privileges, it can run slowly and is only advisable for small import files.

Force Database Import - Access of the user who started the import to the hierarchies or attributes being modified is ignored, and the import succeeds. Typically this setting results in a faster import time and is useful for: very large imports, data migration, when working with externally maintained data, commercial data, or small changes to existing data. This mode is not beneficial when the import will create references.

Remove Options

Setting any of the Remove Options means that any existing data for the selected option type will be removed on the objects included in the import file, and **only** the data of the selected option type for the objects in the import file will exist in STEP after the import. Essentially, the remove options allow you to remove existing data from STEP and replace it with the data in the inbound file for the objects being imported.

Leave the Remove Options set as default to add the imported data to the existing data in STEP.

Important: The Remove Options can both add and remove data from STEP. Prior to using them, it is a good idea to first create an export file of the data being modified. This provides a way to restore the unmodified data, in the event that the import does not have the desired effect.

Remove Un-Mapped Values

This option enables you to remove attribute values from one or more selected attribute groups and replace them with the values specified in the load file, leave them blank, or return them to their inherited value.

Remove Un-Mapped Values

Attribute Group

Select Attribute Group(s)

Important: Use this option with care as it can both add and remove data from STEP.

Click the 'Select Attribute Group(s)' link. In the Select Attribute Group dialog, select the Attribute groups available in the STEP that should be modified by the import. Only values included in the import file will exist for the selected groups after the successful import.

Approve Import Changes

Auto-Initiate STEP Workflows on Item Creation

Trigger STEP Workflow import events on Item Updates

Suppress Re-Translations

Auto select Import Mode

Force Database Import

Force Domain Import

Remove Options

☐
☒
☐
☐
☐
☐
☒

Remove Un-Mapped Values

Attribute Group

Select Attribute Group(s)

Select Attribute Group

Browse
Search

Attribute Groups

- (AttributeGroup)
- (Blacklisted_1)
- (CalculatedAttributes)
- (CustomerData)
- (DataContainers_Entities)

Details

Select
Cancel

For example, you have 200 products listed in your load file, and they are all gloves of one type or another. You use 8 product attributes to define these gloves. Now you want to use just four attributes. Further, all of the original 8 attributes are in their own attribute group. You load the 200 products using just the 4 attributes that you want to use, and you then select the attribute group under the 'Remove Un-Mapped Values' option. STEP removes all attribute values in that attribute group for those 200 products only, and then loads the attribute values as specified in the load file. Thus, it has removed all 8 old attribute values for all 200 products, and it will show only the 4 attribute values present in the load file.

Remove Un-Mapped Data Containers

Data container types can be removed from the entities being imported into STEP via the Data Containers type option. A dialog allows you to confirm that you want to remove all unmapped types for the imported objects.

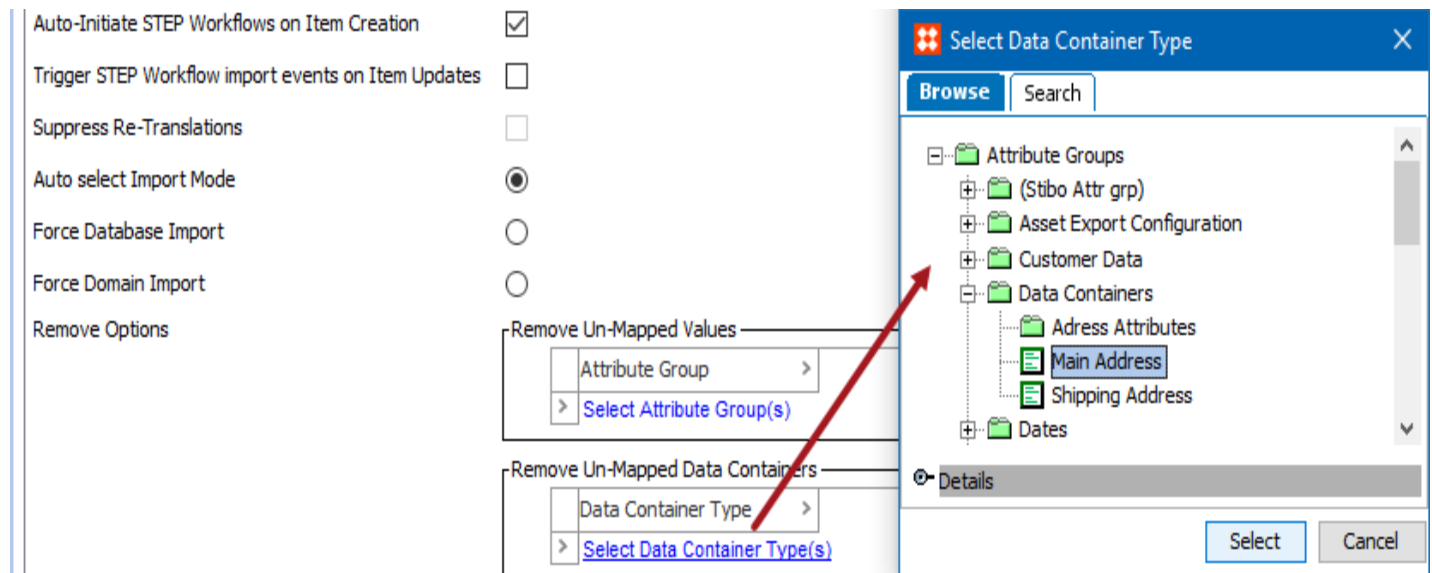
Remove Un-Mapped Data Containers

Data Container Type

Select Data Container Type(s)

Important: Use this option with care as it can both add and remove data from STEP.

Click the 'Select Data Container Type(s)' link. In the Select Data Container Type dialog, select the Data Container type available in the STEP that should be modified by the import. After a successful import, only values for the selected Data Container Types included in the import file will exist on the imported objects.



For example, you have a data container type that allows for multiple data containers (e.g., Shipping Address). Your warehouse usually ships to all the locations in Shipping Address, but now your warehouse has been designated to ship to addresses in a different data container type named Shipping Addresses - West. Instead of adding all new addresses to Shipping Addresses, you can do an import and un-map the original data container type and import only the applicable data container type and the associated data containers.

Remove Un-Mapped References

Selecting a reference option allows a user to specify which reference type they would like to remove. Information and examples for each reference type follows. For each type, a dialog allows you to confirm that you want to remove all unmapped types for the imported objects.

Remove Un-Mapped References

☐ Classification References

Product Reference Type >
> Add Product Reference Type

Asset Reference Type >
> Add Asset Reference Type

Entity Reference Type >
> Add Entity Reference Type

Classification Product Link Type >
> Add Classification Product Link Type

For details on how to add or remove references types, see the **Adding and Removing Un-Mapped References** section below.

Important: Use these options with care as they can both add and remove data from STEP.

- **Classification References** - This is a legacy option that no longer affects the import. Based on the mapped classification reference or link, use the Classification Product Link Type or Classification Reference Type option below to choose one or more classification types.
- **Product Reference Type** - This option enables you to clean up product-to-product references, classification-to-product references, or entity-to-product references.

For example, if you have products in your system that reference other products using the reference type 'Replacement Product'. Over time, products that really are replacement products have been linked to other products using other reference types such as 'Obsolete Product' or 'Similar Product'. To clean up, you list the products and their valid reference types in your load file, map them, and then select the reference type(s) that you want to remove from these products.

The system then reads your load file and finds the source product, which is mapped to the ID and will be linked to one or more target products. All references are then removed from that product to all other products that have the reference type(s) selected in this import wizard screen. The source product is linked in the load file to the target product(s) using the reference type that was selected when you specified the target product.

For example, Product 123 is linked to products 456, 567, 678, and 789 with the reference type of 'Obsolete'. All those links are incorrect. Therefore, you specify in your load file that 123 should link to 567 with the reference type of 'Replacement Product', and nothing else. There are no other entries for product 123 are in the load file. In this screen, you select the reference type 'Obsolete'. After this load file has been processed, there are no links from product 123 to any product with the reference type 'Obsolete', and there is a link from 123 to 567 with the reference type of 'Replacement Product'.

Note: You can only unlink the specified reference types. To unlink **all** references from a product, regardless of the type, you must select each reference type.

To make two references between two products (and all the other product ID pairs listed in the input file), you must load the file twice, specifying one reference type the first time, and the other reference type the second time. For the second import, do not use the Remove Un-Mapped References option.

- **Asset Reference Type** - This option refers to product-to-asset references, classification-to-asset references, or entity-to-asset references. It enables you to clean up references, for example, by removing some or all references between products and assets and applying only those listed in the load file.

For example, in STEP you have products that are linked to images using the reference type 'Primary Image'. However, images that are really manufacturer logos have been linked to various products using this reference type, instead of using the 'Vendor Logo' reference type as intended. You therefore list the products and images in the load file together with the correct reference type, map the columns, and then select the reference type to be removed ('Primary Image') from the product / image. The system removes the selected reference type(s) and only the reference type(s) listed in the load file is available after import.
- **Entity Reference Type** - This option refers to any reference with a target of entity. It enables you to clean up references, for example, by removing some or all references between classifications, products, assets, entities,

or publications and entities, and applying only those listed in the load file.

For example, in STEP you have entities that are linked to entities using the reference type 'Address 1'. However, for identification purposes, they should have been linked using the 'Primary Address' reference type. You therefore list the entities and the address data (also an entity object in STEP) in the load file together with the correct reference type, map the columns, and then select the reference type to be removed ('Address 1') from the entity. The system removes the selected reference type and only the reference type listed in the load file is available after import.

- **Classification Reference Type** - When you have mapped a Classification Reference Type, this option allows you to specify the type(s) that should be removed during the import. This reference type is used when an asset / product / classification is the source, and a classification is the target. When this option is displayed, the Classification Product Link Type option is not displayed. For more information on classification reference types, see Reference Types in the System Setup / Super User Guide documentation.
- **Classification Product Link Type** - When you have mapped a Product to Classification Link Type, this option allows you to specify the type(s) that should be removed during the import. This is used to link a classification as the source to a product as the target, and display the product as a child of the classification. When this option is displayed, the Classification Reference Type option is not displayed. For more information on classification product link types, see Product to Classification Link Types in the System Setup / Super User Guide documentation.

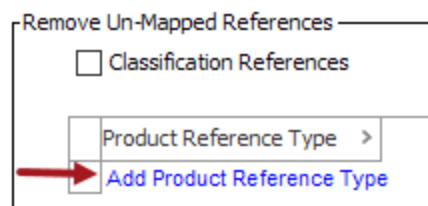
For example, if you have products that are linked multiple times to different classifications, you can use this option to clean up product to classification links not specified by the import file. You therefore define certain products that are linked only to the classifications you list in your load file and not to any other classification.

Do that by listing the products and classifications in the load file, mapping them, and then selecting the types that should be removed.

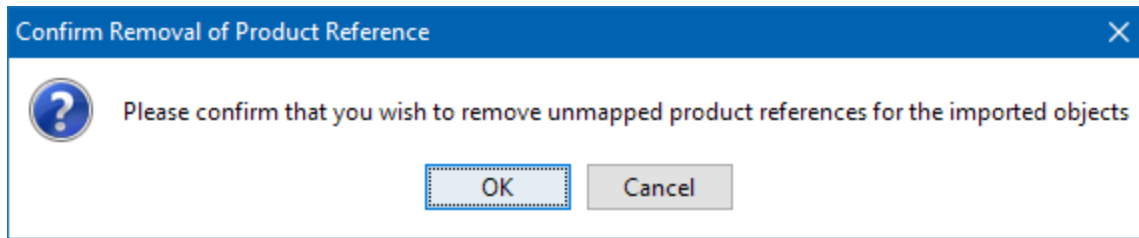
Adding and Removing Un-Mapped References

Interacting with the various **Remove Un-Mapped References** options is the same. An example of how to add and remove a reference type is explained below using the 'Product Reference Type'.

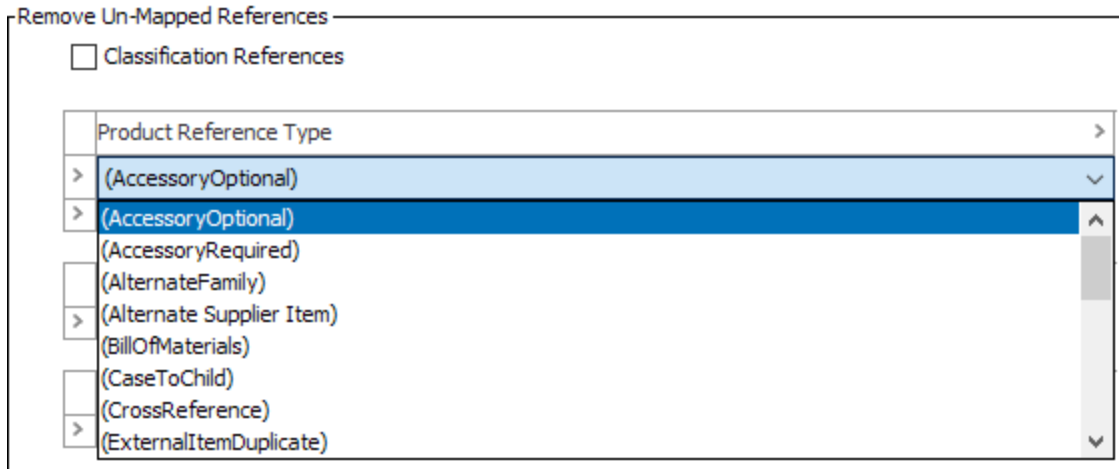
1. Click the option 'Add Product Reference Type'.



2. A dialog allows the user to confirm that they want to remove all unmapped Product references for the imported object.

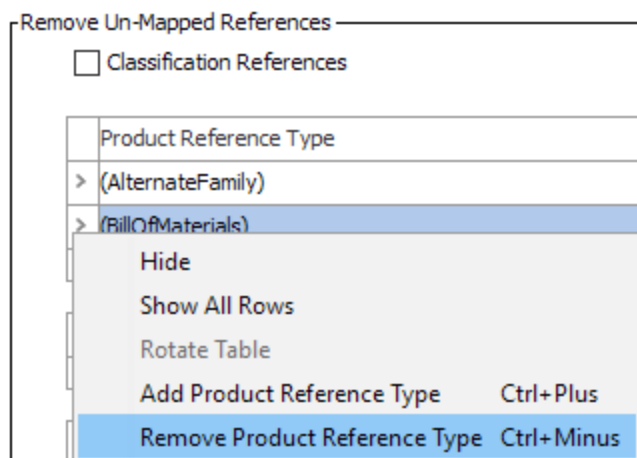


3. Click OK, and then select the appropriate reference type from the dropdown list.



4. Right-click on the field of the Reference type will give the following options:

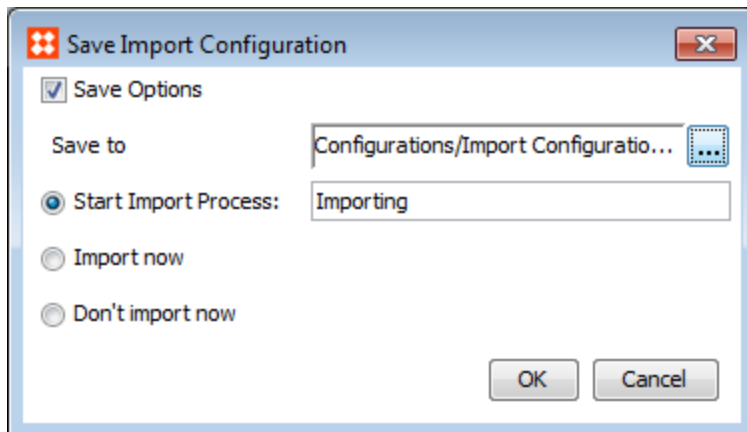
- **Hide** - hides the selected reference type from the dropdown list.
- **Show all Rows** - shows all the selected reference types.
- **Add Reference Type** - adds an additional reference.
- **Remove Reference Type** - removes the selected reference type.



Running a Data Import

After clicking the Finish button on the Import Manager wizard in workbench, the **Save Import Configuration** window displays and allows you to save the configuration and then start the import.

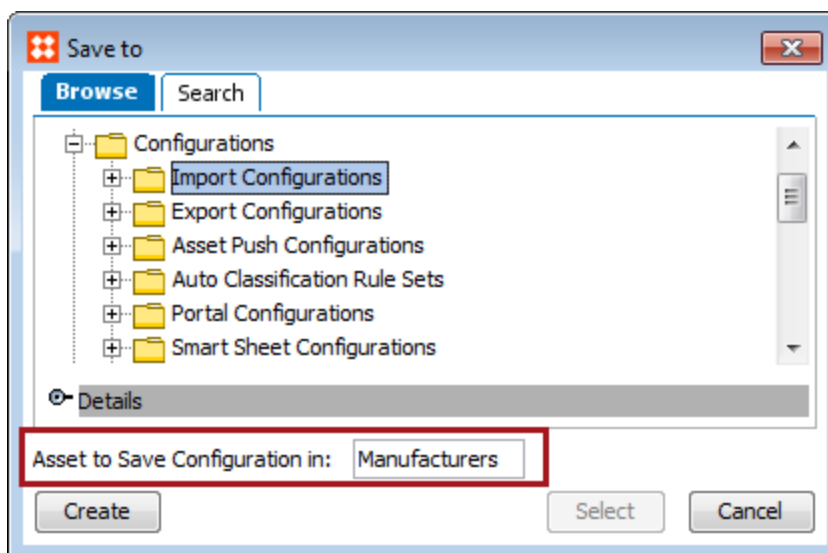
Saving a configuration allows you to modify it (instead of recreating it) during testing. Initially, use a small group of objects to test your settings. Refine the wizard parameters to resolve any issues and save the updated configuration.



1. To save the configuration so that it can be used for other imports, check **Save Options**, click the ellipsis button (...), and then use the **Browse** or **Search** tab to select a location. Configurations are saved as a special type of asset and live in the classification hierarchy.

When updating an existing configuration, the path and name of the configuration that you selected on the first wizard screen is automatically displayed in the **Save To** box.

2. In the **Asset to Save Configuration in** field, enter a name for the configuration, and then click **Create**.



Note: Saving the import configuration is optional, you can also start the data import process without saving the configuration

3. Optionally, you can add a name for the import process in the **Start Import Process** text field. This makes finding the process on the BG Processes tab easier.
4. Determine the final action:
 - Select the **Start Import Process** radio button and click **OK** to send the job to a background process.
 - Select the **Import Now** radio button, which is available if the import file is less than 100 KB on a default system. This allows you to import the file without the use of a workflow. Click **OK** to send the job to a background process.
 - Select **Don't Import Now** to only save the settings, assuming you selected the Save Options checkbox. Click **OK** to close without importing.

Note: No data is imported into STEP if, on the **Identify Destination** step, you checked the **Test Only Import** checkbox. Details of the test import are available on the **Import Report** dialog when selecting the Import Now option; details are available in the BG Process when selecting the Start Import Process option.

Maintaining a Saved Import Configuration

Completing the Import Manager wizard in workbench includes an option to save an import configuration (see **Running a Data Import**). Viewing a saved configuration allows you to confirm the current settings. Editing a saved configuration affects future imports and can be a valuable step in the process of creating and testing an import to ensure the correct data is received in the proper manner.

View a Saved Import Configuration

After clicking the Finish button on the Import Manager wizard, the option to save the configuration is displayed. Although all classification folders are available, using a specific folder for import configurations makes it easier to locate them for review.

Note: If you know the name of the configuration, use the **Goto** search option to find it anywhere in the Tree.

Context: English US URL: Context&workspaceid=Main Goto: ID/Name ExcelImport

Tree

- Assets
- Classifications
- Configurations
 - Asset Push Configurations
 - Auto Classification Rule Sets
 - Export Configurations
 - GDSN Configuration Downloads
 - GDSN Receiver Configuration Downloads
 - Image Conversions
 - Import Configurations
 - ExcelImportEntity
 - ExcelImportProduct
 - ImportBinaryAsset
 - IMPORTEDBINARY
 - List of Values
 - Simple Import
 - TelevisionImportVersion23
 - UNSPSC13
 - UNSPSC17
 - Portal Configurations
 - Smart Sheet Configurations
 - Transformation Lookup Tables
 - AdvSTEPXML
 - Description Attribute Merging
 - Excel Template
 - Filter Template
 - Sample Configuration

ExcelImportProduct rev.4.0 - Images & Documents

Images & Documents | References | Referenced By | Status | State Log

Description

Name	Value
ID	180525
Name	ExcelImportProduct
Object Type	Import Manager Configuration
Revision	4.0 Last edited by USERJ on Mon Jun 27 16:55:49 ED...
Approved	Never Been Approved
Translation	Not Translated
Path	Classification 1 root/Configurations/Import Configur...
Content In	Language=Language Root, Country=Country Root
Asset Keywords	abc
Asset Object Type	Import Manager Configuration
Calculated Asset File Name	180525-ExcelImportProduct

System Properties:

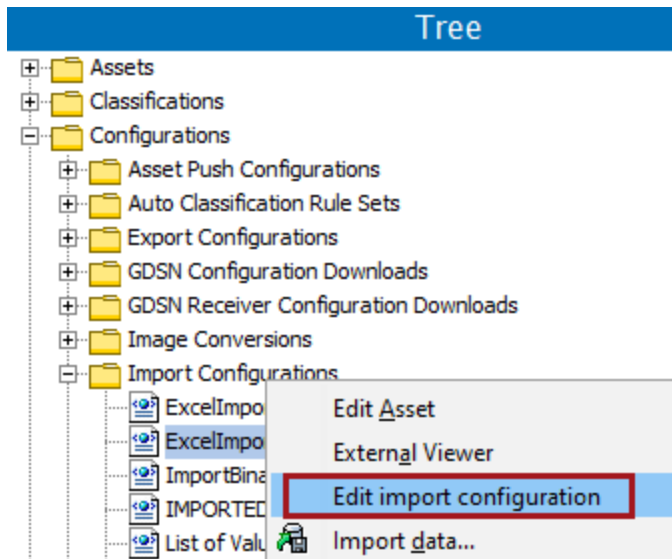
Name	Value
Extension	xml
Filename	ImportConfig5662778875734351770.xml
Format	XML (Extensible Markup Language document)
MIME Type	application/xml
Size	1,697
Upload Time	2016-06-27 16:55:49

Edit a Saved Import Configuration

These editing options are available to update the selected configuration.

- To change only the name of the configuration, simply edit the Name field in the configuration editor.
- To edit an existing configuration and save with a new name, see the steps in the **Running a Data Import** section.
- To change the settings of the existing configuration parameters, follow the steps below.

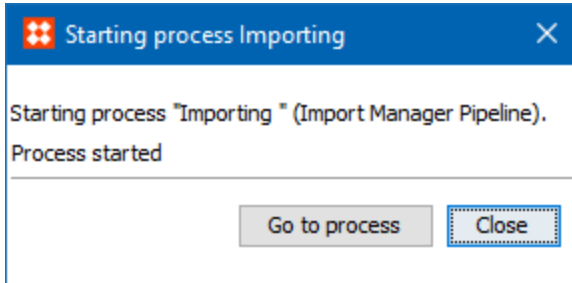
1. Select an import configuration, right-click and select **Edit import configuration** option.



2. The Import Manager wizard opens on Select Data Source.
3. Make all changes and click the **Finish** button to update the selected configuration.

Monitoring a Data Import

When you launch a data import process, STEP starts a background process. It assigns a unique process number and gives it the name assigned on the Save Import Configuration dialog. The starting process dialog is displayed:



Click the **Go To Process** button to display the current background process on the BG Processes tab.

Monitoring the Import Process

On the BG Process tab, select the **Import Manager Pipeline** node. Select the process to monitor based on its status:

- For processes not yet started, open the **Queued Processes** folder.
- For processes that are currently being performed, look in **Active Processes** folder.
- For processes that have completed but had errors, look in **Completed with Errors** folder.
- For processes that have completed, look in **Ended Processes** folder.

The Properties flipper includes details about the import process.

For more information on background processes, see **Background Process Queues** in the **System Setup / Super User Guide** documentation.

BG Processes

- Import Manager Pipeline
 - Queued Processes
 - Active Processes
 - Completed with Errors
 - Ended Processes
 - Importing**
- Import Transfer Package
- ImportInDesignPackage
- Install Change-Package
- Match Code Processes
- Matching Pair Export
- Matching Processes
- Migrate assets/DTP documents to the
- MountDocForPlan
- MultipleRevive
- NodeListToCollection
- PDF of planned page
- PageDone
- Pre-generate thumbnail cache
- Publication Duplication
- Publish to Website
- Purge Dimension Point
- Purge Matching Data
- Purge Revisions
- Purge Workspace
- Re-link Document Assets
- Recycle Bin and Object Purge
- Refresh Node Collection
- Remove Items From STEP Workflow
- Report Process
- Republish nodes
- Revive Recursively
- RuleSetExporter
- RuleSetImporter

Importing - Background Process

Background Process Queue Info

Properties

Property	Value
Started by	USERJ
Id	BGP_179903
Description	Importing
Execution Server	doc-dev
Progress	Done
Status	succeeded
Created	Thu Jun 02 16:17:51 EDT 2016
Started	Thu Jun 02 16:17:56 EDT 2016
Finished	Thu Jun 02 16:17:57 EDT 2016
Processing Time	0 m 1 s
Time in Queue	0 m 5 s
# of warnings	0
# of errors	0

Execution Report

- 1 Retrieval started (Thu Jun 02 16:17:56 EDT 2016)
- 2 Retrieved 36864 bytes (Thu Jun 02 16:17:56 EDT 2016)
- 3 Conversion started (Thu Jun 02 16:17:56 EDT 2016)
- 4 Converted 5 objects (Thu Jun 02 16:17:56 EDT 2016)
- 5 Logged on
- 6 Mapping started (Thu Jun 02 16:17:56 EDT 2016)
- 7 Mapping completed (Thu Jun 02 16:17:56 EDT 2016)
- 8 Import Started (Thu Jun 02 16:17:56 EDT 2016)
- 9 Logged On
- 10 Using import mode "domain"
- 11 Starting first import pass (creating system setup objects)
- 12 Starting second import pass (importing data)

Properties Flipper

The following data is displayed in for the selected background process:

1. **Started by** displays the ID of the user who started the import.
2. **ID** is the Background Process ID which is generated sequentially and always starts with "BGP_" for each process that is started by a user.
3. **Description** is the name that is given by the user for the import process.

4. **Execution Server** is, by default, the application server where STEP is installed. However, it could also be a cluster server.
5. **Progress** displays status bar with the completion percentage of the process.
6. **Status** shows the status of the process, as defined below:
 - Completed means that the process completed successfully.
 - Completed with Errors indicates that the process was partially completed but with the errors included in the error file.
 - Failed means that the process there was a fatal error and nothing was processed.
 - Aborted means that the process was aborted by a user while the process was executing.
7. **Created** is the date and time when the process was created and moved to a particular background process queue.
8. **Started** is the date and time when the process started.
9. **Finished** is the date and time when the process ended.
10. **Processing Time** displays the number of hours, minutes, and seconds it took to complete the process.
11. **Time in Queue** displays the total amount of time that the process was in the queue before it started executing.
12. **# of Warnings** shows the number of warnings that were present after the process was executed. Corrections may be desired.
13. **# of Errors** shows the number of errors that were present after the process was executed. Corrections are required to complete the import successfully.

Execution Report

The execution report gives information about the data import, including validation errors found during import.

Typically, an execution report can be very long, and it can be useful to copy and paste the information into a standard text editor. You can then go through the file to locate errors and warnings.

Handling Import Errors

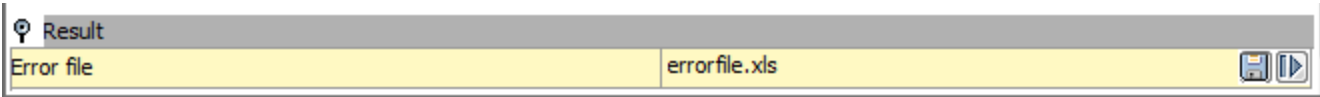
Some imports result in errors, for example, importing a new object (product) that is linked to another new object (classification or asset) that is also being imported. The background process fails, but the user can use the execution report to identify the error and resolve it by referring to the list of common errors.

For more information, see the **Import Error Messages** section and the **Import Error Message Examples** section.

When the following errors are reported, the creation of new objects is skipped entirely:

- Specified parent object does not exist
- Specified Object Type is not valid below specified parent
- No ID is provided and no ID pattern is specified for the target Object Type

When an import process completes with errors, a file is created that contains just the rows with erroneous data. The file can be saved to disk or opened directly in the Import Manager.



When a data error is encountered, the error message reports the data it was attempting to load, including the attribute name and the object ID for the value, and the type of error encountered. The line number and column number are also given, but that is not always useful since it refers to the interim XML file that is created.

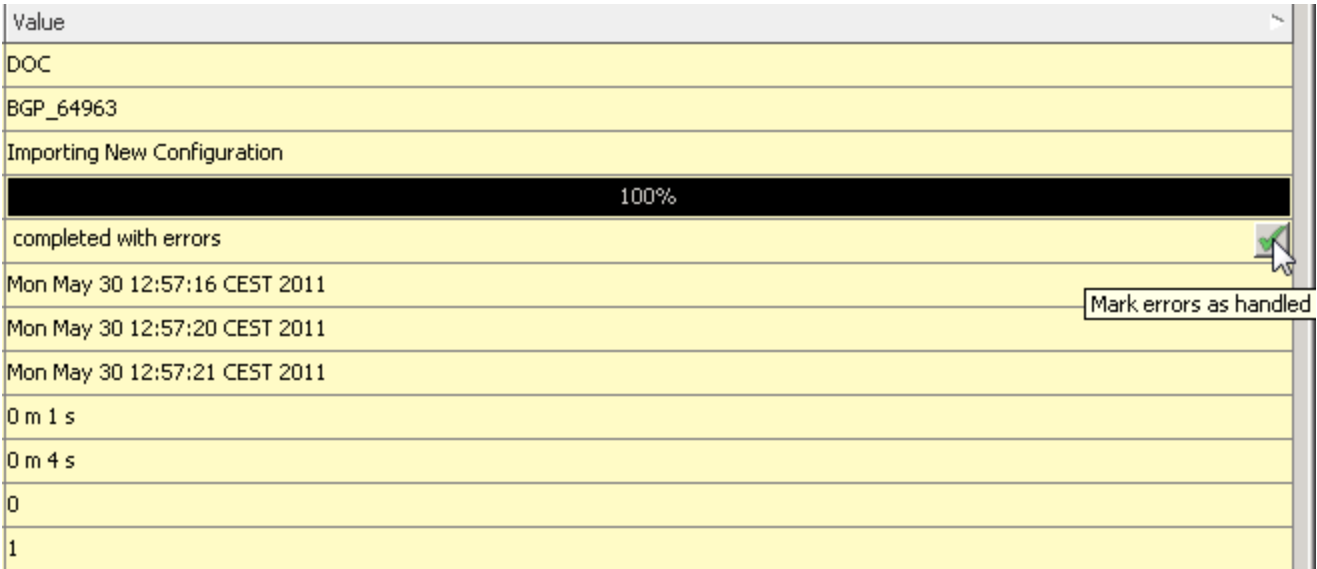
For Excel and CSV imports, the error file contains the rows from the original file with the defect data.

For XML files, the error file contains the objects with errors.

Reimporting a Corrected File

Use the error file to identify the problematic data in the import file and determine the next step:

- If you can correct the data errors with transformations, re-import the same file and create the necessary transformation. When you re-import a file, the status of the background process changes from 'completed with errors' to 'succeeded' and the process is moved to the **Ended Processes** group.
- If transformations will not resolve the issues, edit the import file offline, save it and then import it. You can make the state transition by clicking the **Mark errors as handled** button as shown below.



Import Error Messages

The following table shows most of the different types of error message that you are likely to encounter, with a brief explanation of the most common reason for the error. The most common errors allow users to troubleshoot and then perform the import. There are instances where issues are not related to the import file and are related to the servers, network latency, etc. For these scenarios, contact the Stibo Support team for assistance.

The error messages marked with an asterisk (*) are errors that you will only encounter when you load an XML file. That is because in an XML import, the wizard does not prevent you from entering an invalid value. However, when importing CSV or XLS files, since you often select values from a list, you cannot select an invalid value.

Error Message	Description
Illegal value x for attribute y in product z	The value in the load file does not meet the validity or other constraint that is set up for the attribute.
Not in legal values list	You tried to load an attribute value into an attribute that uses an LOV, and that value does not exist in the LOV, and it is set up not to accept additional entries. This error is not displayed when an LOV is configured to 'Allow Users to Add Values = Yes'.
Attribute not legal for object type	The product's object type is not valid for the attribute. That is, the attribute is not allowed to hold a value for products with the specified object type.
*Illegal unit x for attribute y in product z	In the XML file, you have tried to assign a non-existing unit to an attribute.
Unknown Classification x in product y.	In the XML file, you have probably requested a product-to-classification link using a specified reference type, but the classification you specified is not found.
Unknown Parent product x for product y	You tried to assign a product to a parent ID, but that parent ID cannot be found in the system.
*Unknown product reference target	You attempted to link one product to another via a reference type, but the target product does not exist.
*Unknown attribute	You specified an attribute in the XML file, but that attribute does not exist.

Error Message	Description
Unknown unit	The unit encountered in the load file does not exist in STEP.
Rejected new product x	You set the option to "Reject New Products" in the wizard, and a new product was encountered in the load file.
Illegal unit for attribute	The unit exists but is not valid for the attribute you are trying to load the value into.
*Illegal Object Type for product position	You tried to create a new product via an XML load and you specified an object type that is not a valid choice, given the parent product that you specified.
*Unknown Object Type	You specified an object type (e.g. for a product) in the XML file, but that object type does not exist.
*Unknown reference Type	You specified a reference type (e.g. for a product-to-product reference) in the XML file, but that reference type does not exist.
Insufficient privileges to ...	You do not have permission to carry out one of the following actions (specified in the error message): • set value • set name • create product • move product • modify Object Type • classify product • create product reference
Product x contains more than one reference to product y of Type z, reference data will not be imported.	You cannot make a product-to-product reference from the same source product to the same target product more than once.
Product x contains more than one reference to asset y of Type z, reference data will not be	You cannot make a product-to-asset reference from the same source product to the same target asset more than

Error Message	Description
imported.	once.
*Classification x contains more than one reference to Classification y of Type z, reference data will not be imported.	You cannot make a classification-to-classification reference from the same source classification to the same target classification more than once.
*Reference Type x is not valid for importing a product cross reference from a to b.	In your XML file, you have asked for a product reference that refers to a non-existent reference type.
*Reference Type x is not valid for importing an asset cross reference from a to b.	In your XML file, you have asked for an asset reference that refers to a non-existent reference type.
*Reference Type x is not valid for importing a Classification cross reference from a to b.	In your XML file, you have asked for a classification reference that refers to a non-existent reference type.
*Product x already contains a reference to Classification y but it is invisible in the specified context. The reference and any data on it will not be imported.	You have probably made a product-to-classification link dimension-dependent, and the context you are importing into cannot "see" that link, but it is still there.
Conversion Error	This run time exception error is returned when the XML file is loaded is poorly formed. Download the error file, make the indicated correction, and import the file again.
Conversion Error: Not a Valid STEPXML File (Encoding)	When any encoding type other than UTF-8 is entered in a STEPXML import file, this error is displayed in wizard on the 'Select Format'.
Unknown validator	Importing STEPXML that includes an attribute with a validator type that does not exist in STEP. Validation types with multiple words must replace the space with an underscore. For example, 'Numeric Text' must be written as 'Numeric_Text'.
Context 'XYZ' not found	Importing STEPXML where the Context ID is either misspelled or does not exist.
No ContextID specified in STEPXML to import	Importing STEPXML without a Context ID.

Error Message	Description
No WorkspaceID specified in STEPXML to import	Importing STEPXML without a Workspace ID. Workspace ID should always be an editable workspace, often 'Main'. Since the Approved workspace cannot be edited, it cannot receive imports.

Import Error Message Examples

The most common errors encountered during data import refer to an invalid attribute value. Sometimes there is a mismatch with the attribute's validation type, other times there is a mismatch with one of the other setups. For example, the attribute's constraints, such as masks, minimum and maximum values, maximum length, LOVs, the object's own object type, and so on. Therefore, when you encounter an error, look at the attribute setup and review the constraints.

Even if you load attribute values for an object, not all values will be automatically available in STEP. Attributes must be made valid for an object before the values can be accessed via references. STEP will always load attribute values if the object's type is valid for the attribute, and the attribute values meet the validity criteria. But, the attribute itself must be a valid attribute for that object, that is, linked somewhere in the object hierarchy or classification hierarchy where the object resides.

Use the following examples to assist in troubleshooting problems with data imports.

Illegal Unit for Attribute

The tab-delimited input file below had two columns mapped: the first column was mapped both to the Product ID and Product Name, and the second column was mapped to the attribute 'Description'.

```
T100-3526      RED 1/2"  STRAIGHT WIDGET SSU05PL
T100-2625      BLUE 1/4"  ANGLED GADGET BZT10GL
T100-2827      ORANGE 3/4"  ELL WHATSIT PLU04HF
```

When the file was imported, the execution report showed three illegal unit errors:

- Illegal unit "/2" STRAIGHT WIDGET SSU05PL" for attribute Description in product T100-3526 (UnknownUnit).
Source: Line=2, Column=Column2.
- Illegal unit "/4" ANGLED GADGET BZT10GL" for attribute Description in product T100-2625 (Unknown Unit).
Source: Line=3, Column=Column2.
- Illegal unit "/4" ELL WHATSIT PLU04HF" for attribute Description in product T100-2827 (Unknown Unit).
Source: Line=4, Column=Column2.

Upon inspection of the attribute with the name of 'Description', it was found that the assigned data validity check was 'Numeric_Text', and that it had a unit assigned to it: mm. Further, it was not selected as the default unit.

STEP could not handle the input text, since the attribute's validity was defined as being NumericText and had only one legal unit, and that unit could not be matched with anything in the attribute value provided by the import. In this case, either the attribute's setup is incorrect, or a different product attribute should be used that has an appropriate data validity type assigned.

Unknown Parent Product

The tab-delimited input file includes three columns. Column 1 was mapped to both Product ID and Product Name (new products), and Column 3 was mapped to the Parent ID.:

```
T100-3526      RED 1      Hand Tools
```


T100-2625	1mm ANGLED GADGET	Hand Tools
T100-2827	ORANGE 3mm	Hand Tools

When the file was imported, the execution report showed the following three errors:

- Unknown Parent Product "Hand Tools" for product "T100-3526" (1 product(s) was rejected due to this error). Source: Line=2, Column=Column3.
- Unknown Parent Product "Hand Tools" for product "T100-2625" (1 product(s) was rejected due to this error). Source: Line=3, Column=Column3.
- Unknown Parent Product "Hand Tools" for product "T100-2827" (1 product(s) was rejected due to this error). Source: Line=4, Column=Column3.

STEP could not find the Parent IDs given in Column 3 of the input file, and therefore the new products were rejected. Even though a default Parent ID was selected in the Identify Destination screen of the import wizard, STEP accepts the value for the Parent ID in the input file as an override, and rejects the new product anyway.

A reason for this execution report error could be that one of the input file's columns was mapped to be the Parent ID, and STEP could not find it. If the product existed already, it would not be moved anywhere, it would stay with its current parent. If the product did not exist, it would not be created.

Maximum Length, Illegal Value

The input file had two columns mapped: the first column was mapped both to the Product ID and Product Name (these were new products), and the second column was mapped to the attribute "Manufacturer Part Number".

G100-352654	REDD-52626278-YDS-777777-1/FFGH00KK7733249-UHP
J100-267225	REDD-5262627
D100-289827	K7733249-UHP

The file was imported and the execution report showed the following three errors:

- Illegal value "REDD-52626278-YDS-777777-1/FFGH00KK7733249-UHP" for attribute Manufacturer Part Number in product G100-352654 (Length of the value exceeds max length of domain Is trying to insert a value with '46' characters into a domain that has a maximum length of '40' characters.). Source: Line=10, Column=Column2.
- Line= at line 9: : Length of the value exceeds max length of domain Is trying to insert a value with '46' characters into a domain that has a maximum length of '40' characters.
- Line= at line 9: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEFF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_INS' ; SQL return code=20291; SQL return code=0

These errors are reporting a single problem. The first and second errors say that one of the imported part numbers had a value in the Manufacturer Part Number column that was too long, as defined by the attribute's setup. You are given the part number, the attribute name, and the value that was trying to be imported. The third error message is useful for a programmer.

Note: This type of error does not prevent the new part number from being created. The value for the Manufacturer Part Number, however, is left blank.

Out of Range, Illegal Value

This is another example of an illegal value error:

- Illegal value "250" for attribute Focal Length (ft) in product G100-352654 (Value error: Value '250' is out of range). Source: Line=9, Column=Column2.
- Line= at line 8: : Value error: Value '250' is out of range
- Line= at line 8: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEFF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_INS' ; SQL return code=20291; SQL return code=0

This group of three error messages are reporting a single problem. The first two messages indicate that a value of 250 is too large for the maximum value allowed for the attribute.

Not a Number, Illegal Value

The following tab-delimited input file had two columns mapped: the first column was mapped to the Product ID (the products already existed in STEP), and the second column was mapped to an attribute called 'Hole Diameter'.

G100-352654	1-1/2
J100-267225	2-3/4
D100-289827	17-3/16

The file was imported and the execution report showed the following nine errors:

- Illegal value "1-1/2" for attribute Hole Diameter in product G100-352654 (Value error: Value '1-1/2' is not a number). Source: Line=9, Column=Column2.
- Line= at line 8: : Value error: Value '1-1/2' is not a number
- Line= at line 8: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEFF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_INS' ; SQL return code=20291; SQL return code=0
- Illegal value "2-3/4" for attribute Hole Diameter in product J100-267225 (Value error: Value '2-3/4' is not a number). Source: Line=14, Column=Column2.
- Line= at line 13: : Value error: Value '2-3/4' is not a number
- Line= at line 13: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEFF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_INS' ; SQL return code=20291; SQL return code=0

- Illegal value "17-3/16" for attribute Hole Diameter in product D100-289827 (Value error: Value '17-3/16' is not a number). Source: Line=19, Column=Column2.
- Line= at line 18: : Value error: Value '17-3/16' is not a number
- Line= at line 18: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEf_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEf_INS' ; SQL return code=20291; SQL return code=0

Each set of three errors addresses one of three problems. That is, different information is reported for the same error in three individual messages. The first message in each group is what is most useful to the end user. In the above series, notice that the first lines that indicate that the fractional value in the input file is not allowed for the attribute 'Hole Diameter' because the value 'is not a number'. In this case, the attribute's validation type was set to Number, which does not allow fractional values.

This is another example:

- Illegal value "Hand tools" for attribute Focal Length (ft) in product J100-267225 (Value error: Value 'Hand tools' is not a number). Source: Line=14, Column=Column2.
- Line= at line 13: : Value error: Value 'Hand tools' is not a number
- Line= at line 13: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEf_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEf_INS' ; SQL return code=20291; SQL return code=0

The first message in the group tells you that the attribute value of 'Hand tools' in the input file is not allowed since the attribute is set up to expect a number.

And another example:

- Illegal value "50,70" for attribute Focal Length (ft) in product D100-289827 (Value error: Value '50,70' is not a number). Source: Line=19, Column=Column2.
- Line= at line 18: : Value error: Value '50,70' is not a number
- Line= at line 18: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEf_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEf_INS' ; SQL return code=20291; SQL return code=0

The first message in the group tells you that the attribute value of '50,70' in the input file does not match the attribute's validation type of Number, which does not allow commas.

Object Existed with Another Object Type

The following tab-delimited input file had two columns mapped: the first column was mapped to the Product ID (the products already existed in STEP), and the second column was mapped to an attribute called "MFOB".

G100-352654	Yes
J100-267225	No
D100-289827	Maybe

The file was imported and the execution report showed the following three warnings and one error:

- Line=2: Product 'G100-352654' existed with another object type - supplied object type not set
- Line=3: Product 'J100-267225' existed with another object type - supplied object type not set
- Line=4: Product 'D100-289827' existed with another object type - supplied object type not set
- Illegal value "Maybe" for attribute MFOB in product D100-289827 (Not in legal values list). Source: Line=4, Column=Column2.

The first three warnings say that the products already existed in STEP with the object type of 'Product'. When the file was imported, the Identify Destination screen was skipped, and the default object type that was set in that screen was different from Product. STEP reported that it did not change the object type of the products.

The final message is an error that says that the attribute MFOB used an LOV and the value in the input file was not valid. Further investigation showed that LOV had only two valid values: Yes and No. Additionally, the LOV was set up so that no modifications could be made to the LOV by loading in data. So the value "Maybe" was rejected.

Unknown Classification

The following tab-delimited input file had two columns mapped: the first column was mapped to the product ID (the products already existed in STEP), and the second was mapped as a classification ID.

G100-352654	Hand tools
J100-267225	Hand tools
D100-289827	Hand tools

The file was imported and the execution report showed the following three errors:

- Unknown Classification "Hand tools" in product "G100-352654" (Classification 'Hand tools' does not exist in the import workspace). Source: Line=6, Column=Column2.
- Unknown Classification "Hand tools" in product "J100-267225" (Classification 'Hand tools' does not exist in the import workspace). Source: Line=10, Column=Column2.
- Unknown Classification "Hand tools" in product "D100-289827" (Classification 'Hand tools' does not exist in the import workspace). Source: Line=14, Column=Column2.

In this case, the import file requested that the products be linked to a classification folder, but STEP could not find a classification with the specified ID 'Hand tools'.

Not Privileged to Create Object

A file was imported and the execution report showed the following error:

- Line=2: Not privileged to Create product

Two scenarios can cause this error message:

1. Importing new products when the import user does not have the privilege to create new products. The import user must have all access required for the action to be performed during the import.

2. Importing a product with a product-to-product Reference Type by mapping one column of data to the Product ID and another column to a Reference, while the target product of the reference does not already exist in STEP. A reference can only be imported when the target already exists.

Unknown Asset ID

A file was imported and the execution report showed the following error:

- Line=2: Unknown Asset ID: AC200-627

This error is reported when attempting to link a Product ID to an Asset ID, but the asset does not exist in STEP.

Note: The Asset ID (not the Asset Name) is required in an import file.

Array Operation Failed

A file was imported and the execution report showed the following error:

- Line= at line 8: : Array operation failed (1 times): Error during array operation: ORA-20042: Value rejected: Attribute Gap not valid for this usertype ORA-06512: at "STEPSYS.VALUEMAP_BEf_INS", line 41 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEf_INS'; SQL return code=20042; SQL return code=0

This error is reported when attempting to load an attribute value (not shown in the error) to an attribute that is not legal for the product's object type. Upon further investigation, we found that the product's object type was 'Product'. The error message is saying that the attribute that has the ID of 'Gap' is not valid for products with the object type of 'Product'. Since the value is invalid, it is ignored.

Optimistic Locking Errors

When a file was imported, the below error was displayed in the execution report:

- OptimisticVerificationException: Optimistic locking errors were detected when flushing to the data store. This indicates that some objects were concurrently modified in another transaction. Failed objects:
[com.stibo.core.persistence.ProcessPO@33bf33bf: BGP_3604361] [java.util.ArrayList]

This error occurs when a user is trying to perform an import by modifying an attribute value for an existing product in STEP and at the same time there is another user who has access to the same product and is modifying the attribute value. In these scenario's, the product is locked for any modification and the import fails.

The Transaction has been Rolled Back: Unique Constraint

A file was imported to create a new product and the below error was thrown:

- Caught FatalDataStoreException at Mon Jun 06 13:10:16 CEST 2016: The transaction has been rolled back. See the nested exceptions for details on the errors that occurred., caused by SQLException: ORA-00001: unique constraint (STEPSYS.NODE_NODEID_UIX) violated {prepstmnt 523703057 INSERT INTO NODE (CHECKID, NAME, NODEID, NODETYPE, USERTYPE, USERTYPEID) VALUES (?, ?, ?, ?, ?, ?) [reused=5]} [code=1, state=23000]

The above error was thrown when a new product was being created but, the ID provided to create the new product already existed in STEP. ID's for objects that are created in STEP is always Unique and if a value for an ID already exists then repeating the same again is not accepted in STEP.

Note: ID's used for one product cannot be used again for another product, however, the same ID can be used for to create a Classification folder.

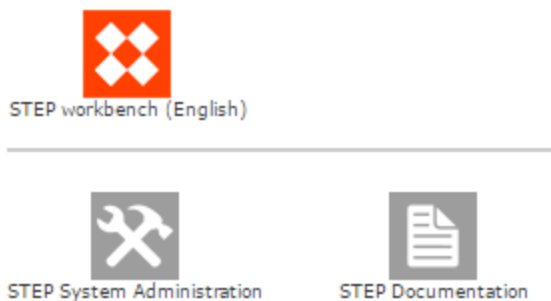
Import errors are shown in the Background processes under the Execution report and the number of errors and warning are also displayed as shown below:

Description	Importing
Execution Server	doc-rel
Progress	100%
Status	completed with errors
Created	Sun Feb 05 06:49:19 EST 2017
Started	Sun Feb 05 06:49:20 EST 2017
Finished	NA
Processing Time	0 m 2 s
Time in Queue	0 m 1 s
# of warnings	0
# of errors	1

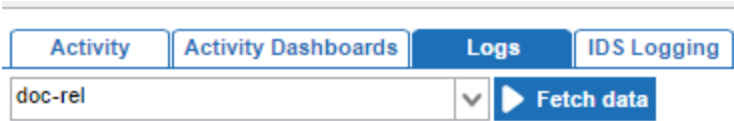
Execution Report	
1	Retrieval started (Sun Feb 05 06:49:21 EST 2017)
2	Retrieved 8817 bytes (Sun Feb 05 06:49:21 EST 2017)
3	Conversion started (Sun Feb 05 06:49:21 EST 2017)
4	Converted 6 objects (Sun Feb 05 06:49:24 EST 2017)
5	Logged on
6	Mapping started (Sun Feb 05 06:49:24 EST 2017)
7	Mapping completed (Sun Feb 05 06:49:24 EST 2017)
8	Import Started (Sun Feb 05 06:49:24 EST 2017)
9	Logged On
10	Using import mode "domain"
11	Starting first import pass (creating system setup objects)
12	Starting second import pass (importing data)
! 13	Row 6, Column : The attribute with ID ' ShortItemDescription ' isn't valid for object product with ID ' 888264
14	Imported 2 new products, 0 new classifications, 0 new entities and 0 new assets.
15	Processed 4 existing products, 0 existing classifications, 0 existing entities and 0 existing assets.
16	Skipped 0 products, 0 classifications, 0 entities and 0 assets.
17	Deleted 0 products, 0 classifications, 0 entities and 0 assets.
18	Found 0 warnings
19	Found 1 errors
20	Import completed (Sun Feb 05 06:49:26 EST 2017)
21	Error file generation started (Sun Feb 05 06:49:26 EST 2017)
22	Error file with 1 object(s) generated (Sun Feb 05 06:49:30 EST 2017)

Similarly, the execution report can also be viewed in the System Administration page by following the below steps:

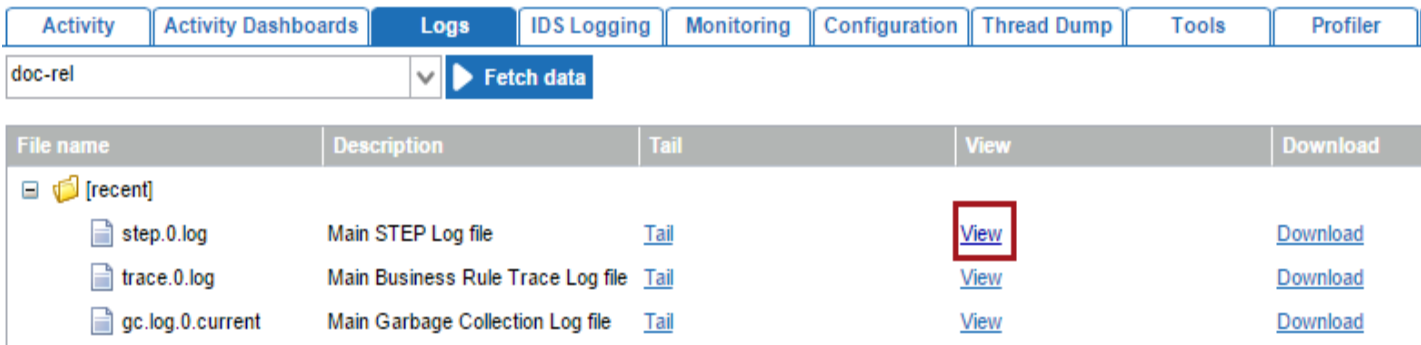
- 1. Launch the STEP WebStart page
- 2. Click on the System Administration link



- 3. Once logged in, click on the Logs Tab
- 4. Select the Server in the dropdown list and click on 'Fetch Data'



- 5. Expand [recent] and then click on view option for step.0.log file



- 6. A new window will open with the log details shown
- 7. Copy the background process ID from STEP

Background Process		Queue Info
⚙ Properties		
Property	> Value	
Started by	USERL	
Id	BGP_115400	
Description	Importing	
Execution Server	doc-rel	
Progress	100%	
Status	completed with errors	
Created	Sun Feb 05 06:49:19 EST 2017	
Started	Sun Feb 05 06:49:20 EST 2017	
Finished	NA	
Processing Time	0 m 2 s	
Time in Queue	0 m 1 s	
# of warnings	0	
# of errors	1	

- Press Ctrl + F on the keyboard and search for the background process ID to find the complete log detail with more additional details.

```
Submitting bg-process (BGP_115400) with template (stibo.Importer) to queue (IN)
2017/02/05-06:49:20 2c8 com.stibo.servicemanager.beans.DefaultBackgroundProcessInvocationImpl run INFO: Start
handling of bg-process (BGP_115400) (recovery=false)
2017/02/05-06:49:21 2c8|USERL|BGP com.stibo.util.unstable.plugin.DefaultPlugin init WARNING: Unable to find
parameter name: ValueSubstitutionAssetIDName
2017/02/05-06:49:21 2c8|USERL|BGP com.stibo.util.unstable.plugin.DefaultPlugin init WARNING: Unable to find
parameter name: WordSubstitutionAssetIDName
2017/02/05-06:49:21 2c8|USERL|BGP com.stibo.util.unstable.plugin.DefaultPlugin init WARNING: Unable to find
parameter name: LibraryAttributeTransformationIDName
2017/02/05-06:49:25 2c8|USERL|BGP com.stibo.systemconfig.ConfigUtil getPropertyTypeCheck WARNING
The property Import.IsMarkNodeFilterValuesModification is not documented, check
com.stibo.core.domain.impl.importer.ValueFilterHandlerDomain.<clinit>(ValueFilterHandlerDomain.java:38) and read
http://confluence.stibo.com/display/RD/STEP+5+configuration+properties
2017/02/05-06:49:30 2c8|USERL|BGP com.stibo.core.domain.impl.backgroundprocess.BackgroundProcessImpl updateStatus
INFO: Setting status on succeeded bg-process (BGP_115400)
2017/02/05-06:49:30 2c8|USERL|BGP com.stibo.core.domain.impl.backgroundprocess.BackgroundProcessImpl$12
lambda$run$0 INFO: Succeeded updating status of bg-process (BGP_115400)
2017/02/05-06:49:30 2c8 com.stibo.servicemanager.beans.DefaultBackgroundProcessInvocationImpl run INFO: Finished
handling of bg-process (BGP_115400)
2017/02/05-07:00:16 4f com.stibo.systemconfig.ConfigUtil getPropertyTypeCheck WARNING
The property Log.ConfigurationPackagesLogRoot is not documented, check
com.stibo.admin.FileUtil.getConfigurationPackagesLogFileNames(FileUtil.java:265) and read
http://confluence.stibo.com/display/RD/STEP+5+configuration+properties
2017/02/05-07:00:53 4e com.stibo.systemconfig.ConfigUtil getPropertyTypeCheck WARNING
The property Log.ConfigurationPackagesLogRoot is not documented, check
```


As explained above, there are common errors and the execution logs can be viewed and then correction can be made accordingly and then the file can be imported. There are other errors which can occur for different reason based on the scenario. If the errors are not correctable then those report logs can be collected and then reported to the Stibo Support team to have the issue fixed.

Inbound Integration Endpoints

Inbound Integration Endpoints (IIEPs) offer a centralized interface for monitoring and maintaining integrations with systems that send data to STEP in the form of files or messages.

IIEPs can poll different data sources for new files or messages to import on scheduled intervals—the minimum interval being one minute. As such, IIEPs offer functionality for 'near real-time' integrations. However, when the data source is REST, a POST will invoke the endpoint. For alternative real-time integrations with STEP, see the SOAP and REST APIs.

IIEPs provide a way to automatically manage the functionality for the standard STEP import functionality. Without any extensions, files, or messages being imported, data is handled by the STEP Importer, and the same options for mapping and manipulation are available as when importing manually.

Setup Requirements

Setting up and using an inbound integration endpoint involves the following steps:

1. Create a setup group to hold the endpoint as described in **Initial Setup for an Inbound Integration Endpoint**.
2. Launch the inbound integrations endpoint wizard as described in **Creating an Inbound Integration Endpoint**.
3. In the wizard, add an ID, name, description, and user as described in **IIEP - Identify Endpoint**.
4. In the wizard, select the receiver of the data as described in **IIEP - Choose Receiver**.
5. In the wizard, specify processing, context, and queue settings for the endpoint as described in **IIEP - Configure Endpoint**.
6. In the wizard, if available, configure defined preprocessors for customer-specific solutions as described in **IIEP - Configure PreProcessor**.
7. In the wizard, specify the data format and map the data, among other options as described in **IIEP - Configure Processing Engine**.
8. In the wizard, if available, configure defined postprocessors for customer-specific solutions as described in **IIEP - Configure PostProcessor**.
9. In the wizard, specify how often the endpoint should search for data to be processed as described in **IIEP - Schedule Endpoint**.
10. In the wizard, specify, select, and configure an error reporter plug-in that is activated if an endpoint-related background process fails as described in **IIEP - Configure Error Reporter**.
11. Enable the endpoint and invoke it as described in **Running an Inbound Integration Endpoint**.

Additional Information

The following information is useful once an inbound integration endpoint is set up:

1. Understand each element of an IIEP as described in **Inbound Integration Endpoint Structure**.
2. Understand the effects of the transactional setting on an IIEP as described in **Inbound Integration Endpoint Transactional Settings**.
3. Maintain or modify the endpoint as described in **Maintaining an Inbound Integration Endpoint**.
4. Monitor the endpoint as described in **Monitoring an IIEP via Background Process**.
5. Monitor the endpoint as described in **Monitoring an IIEP via External System**.
6. Resolve failed background processes as described in **Handling Failed IIEP Background Processes**.
7. Export inbound integration endpoint definition for comparison purposes in an external source control system in **Exporting Inbound Integration Endpoint Definitions as Comments**.
8. Parallel imports involving multiple references on object types as described in **Reference Target Lock Policy on Object Types**.

Initial Setup for an Inbound Integration Endpoint

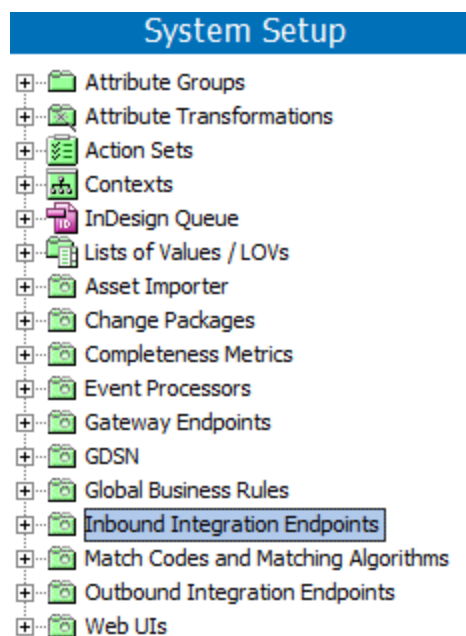
Before creating an inbound integration endpoint, the following items must exist:

- An Inbound Integration Setup Group
- Linking between the IIEP Object Type and the Setup Group
- An instance of the IIEP Object

This setup only needs to be performed once, and most systems will already have it completed.

Only users with the relevant privileges can view or maintain inbound integration endpoints. For detailed information, see the **Action Sets** section and the **Users and Groups** section in the **System Setup / Super User Guide** documentation.

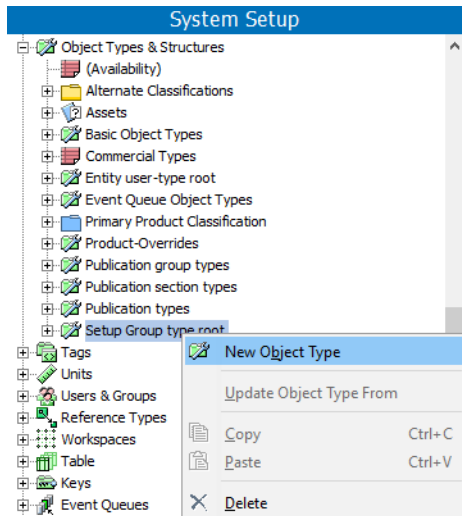
Review your System Setup tab to determine if one or more inbound integration endpoint nodes already exist. The name of the node on your system is not required to match the one in the image below.



Once the setup has been completed, the steps in this section are only needed if additional levels of organization are desired.

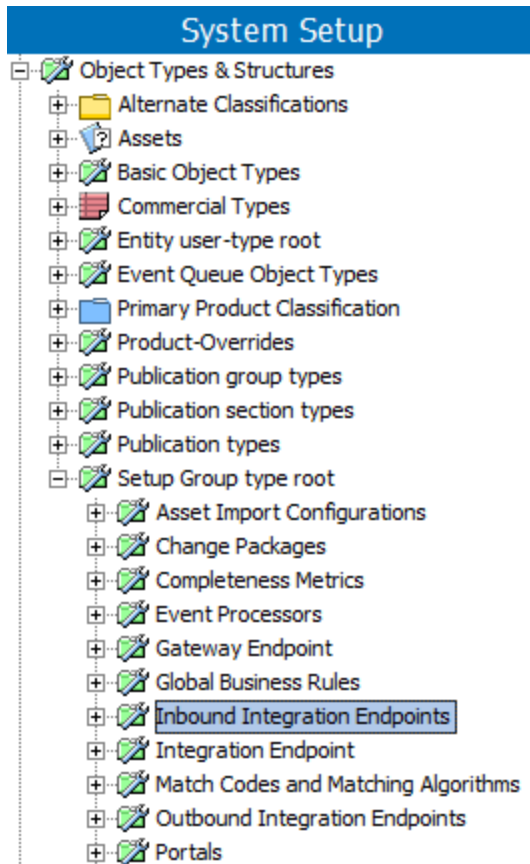
Create the Inbound Integration Setup Group

1. In System Setup, expand **Object Types & Structures**.
2. Right-click 'Setup Group type root', and choose **New Object Type**.



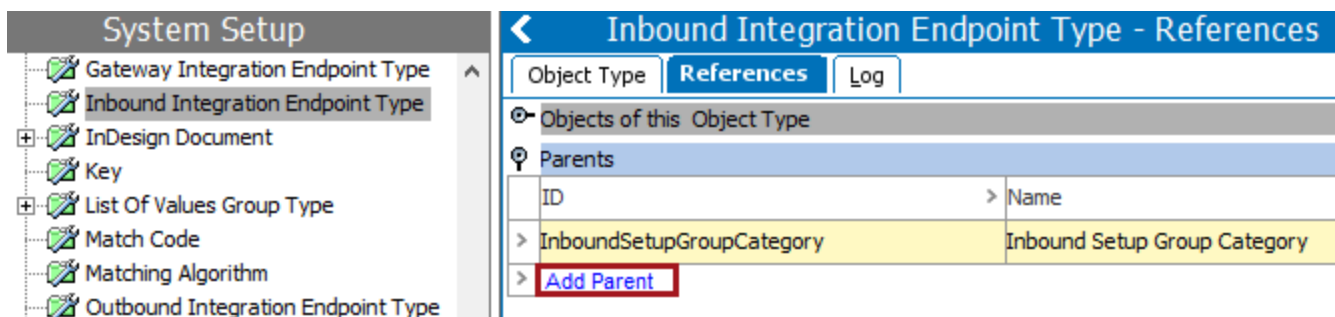
3. Enter an **ID** and a **Name**, select any required Dimension Dependencies based on your system, and click **Create**.

4. The new setup group appears in System Setup under 'Object Types & Structures' as a child in the **Setup Group type root**.

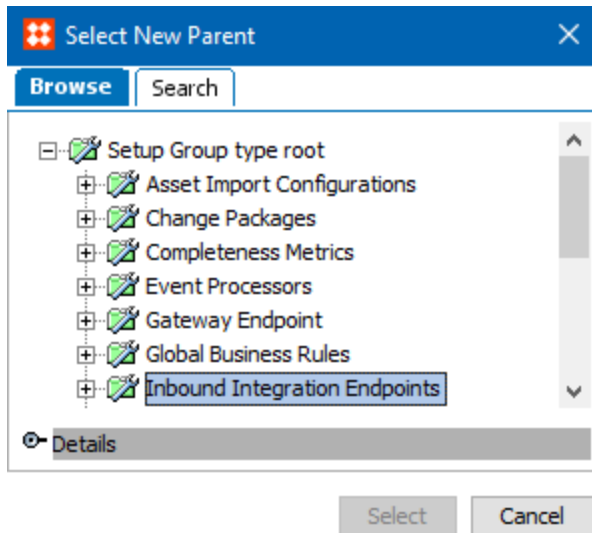


Link the IIEP Object Type to Setup Group

1. In Object Types & Structures > expand Basic Object Types > select **Inbound Integration Endpoint Type**.
2. On the References tab > Parents flipper > click the **Add Parent** link.

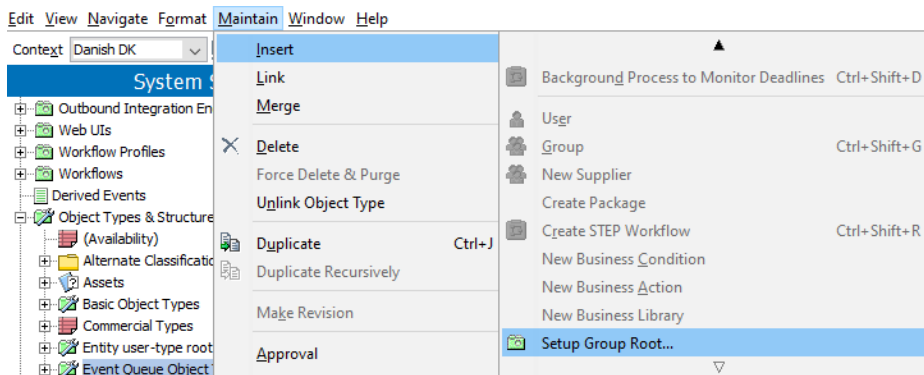


3. In the Select New Parent dialog, select the setup group you created, and click **Select** to make it a valid parent.

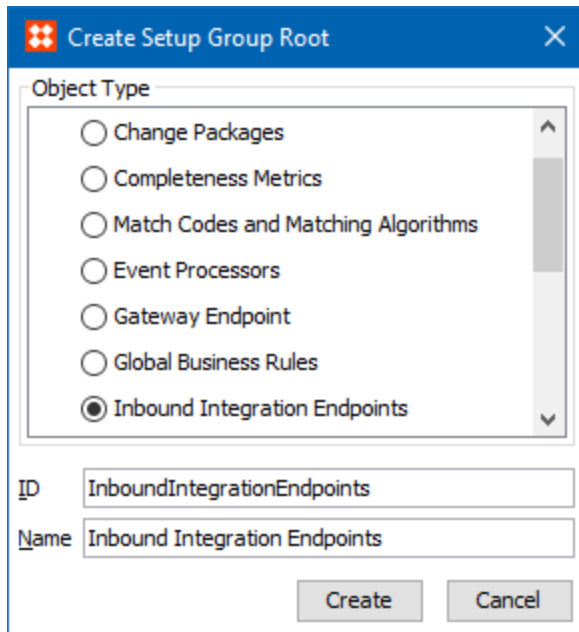


Create an Instance of the IIEP Object

1. On the System Setup tab, select any object in the **System Setup** hierarchy to activate the following Maintain menu selection.
2. Click the Maintain menu, point to Insert, and select **Setup Group Root**.

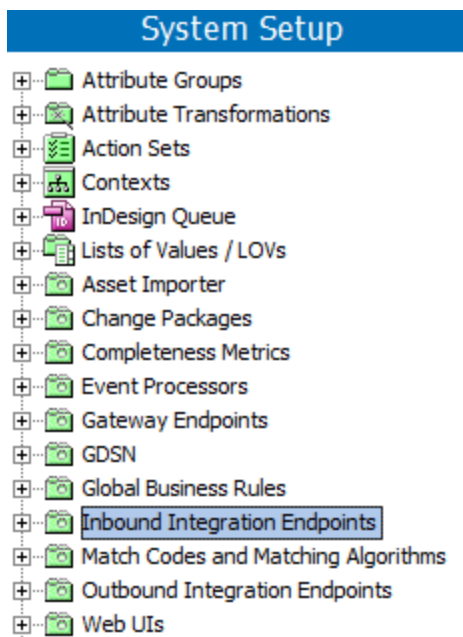


3. In the Create Setup Group Root dialog, select the inbound integration endpoints object type, enter an **ID** and a **Name**, and click **Create**.



The dialog box titled "Create Setup Group Root" has a close button (X) in the top right corner. It contains a section labeled "Object Type" with a list of radio buttons: "Change Packages", "Completeness Metrics", "Match Codes and Matching Algorithms", "Event Processors", "Gateway Endpoint", "Global Business Rules", and "Inbound Integration Endpoints". The "Inbound Integration Endpoints" option is selected. Below this list are two text input fields: "ID" with the value "InboundIntegrationEndpoints" and "Name" with the value "Inbound Integration Endpoints". At the bottom right are "Create" and "Cancel" buttons.

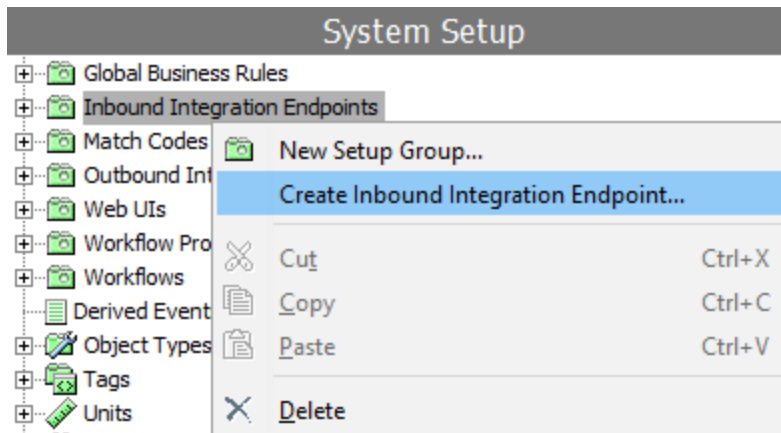
A setup group is created as a node in the System Setup hierarchy. Inbound integration endpoints can now be created under this new node.



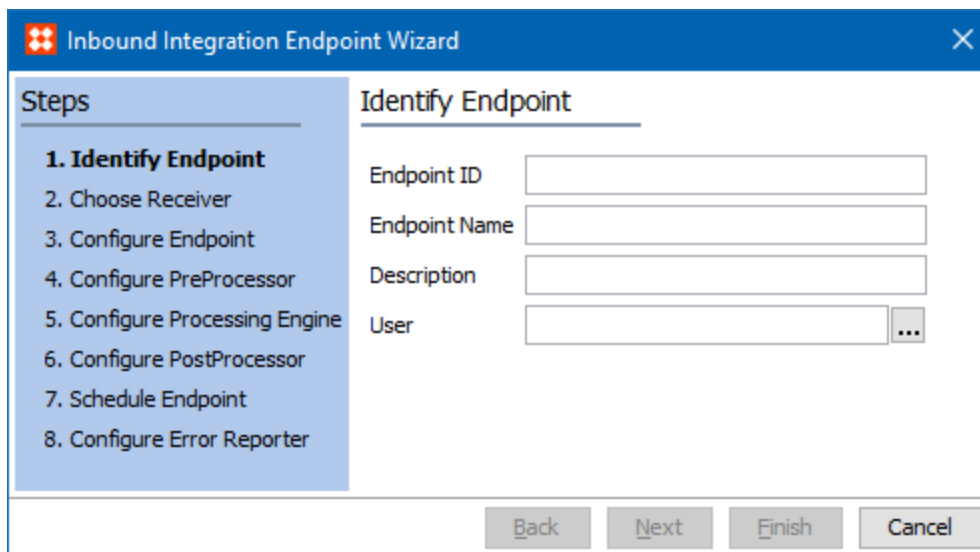
Creating an Inbound Integration Endpoint

After creating a setup group for inbound integration endpoints, create an inbound integration endpoint to monitor and maintain integrations with systems that send data to STEP in the form of files or messages.

1. In System Setup, right-click the Inbound Integrations Endpoints setup group, and click **Create Inbound Integration Endpoint**.



2. The Inbound Integration Endpoint wizard displays and can involve the following steps:



- **Identify Endpoint** allows you to specify the name and ID of the endpoint and identify the user whose privileges are applied.
- **Choose Receiver** allows you to specify the receiver of the data.
- **Configure Endpoint** allows you to specify processing, context and queue settings for the integration endpoint.

- **Configure PreProcessor**, when available, allows you to configure defined preprocessors for customer-specific solutions.
 - **Configure Processing Engine** allows you to specify the data format and map the data, among other options.
 - **Configure PostProcessor**, when available, allows you to configure defined post-processes for customer-specific solutions.
 - **Schedule Endpoint** allows you to specify how often the endpoint should search for data to be processed.
 - **Configure Error Reporter** allows you to specify, select, and configure an error reporter plug-in that is activated if an endpoint-related background process fails.
3. Follow the steps defined in **Running an Inbound Integration Endpoint** to enable and run the endpoint.

IIEP - Identify Endpoint

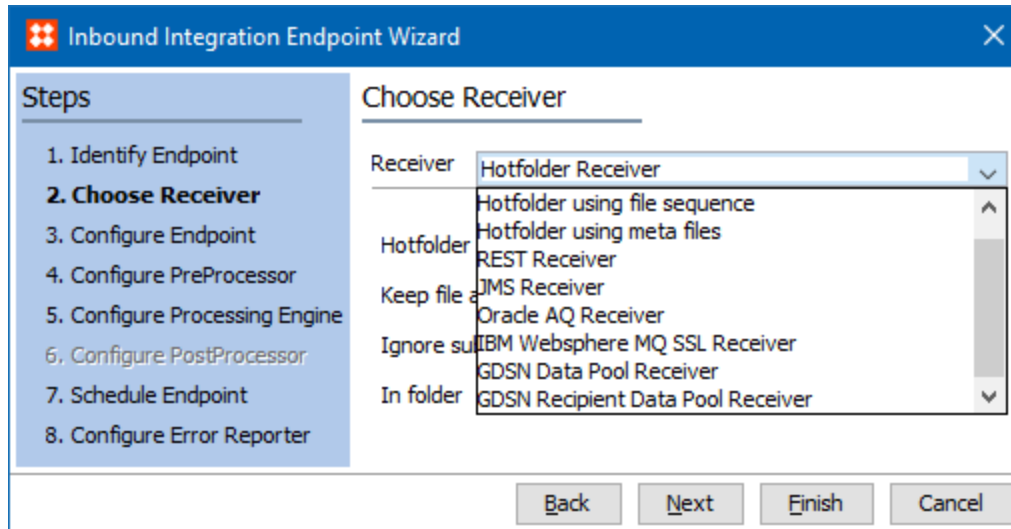
The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' list contains eight items: 1. Identify Endpoint (highlighted), 2. Choose Receiver, 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area is titled 'Identify Endpoint' and contains four input fields: 'Endpoint ID' with the value 'InboundEndpointID', 'Endpoint Name' with 'Inbound Endpoint Name', 'Description' with 'Explanation of the purpose of the IIEP', and 'User' with 'User (USER)' and a browse button (...). At the bottom are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

1. For **Endpoint ID**, enter an ID. Common setup is to use no spaces or punctuation.
2. For **Endpoint Name**, enter a name. Common setup is to repeat the ID with added spaces for readability.
3. For **Description**, enter an optional description for the IIEP.
4. For **User**, search or browse for a user. The privileges of the selected user determine which actions the inbound integration endpoint can perform when it processes data.

Note: It is common setup to create a special system user account for use by each IIEP. This allows you to track changes made by the integration and also allows you to restrict (via privileges) what data the integration can change. This is especially relevant for STEPXML based IIEPs since STEPXML allows you to make significant changes to both data and configuration settings.

5. Click the **Next** button to display **IIEP - Choose Receiver**.

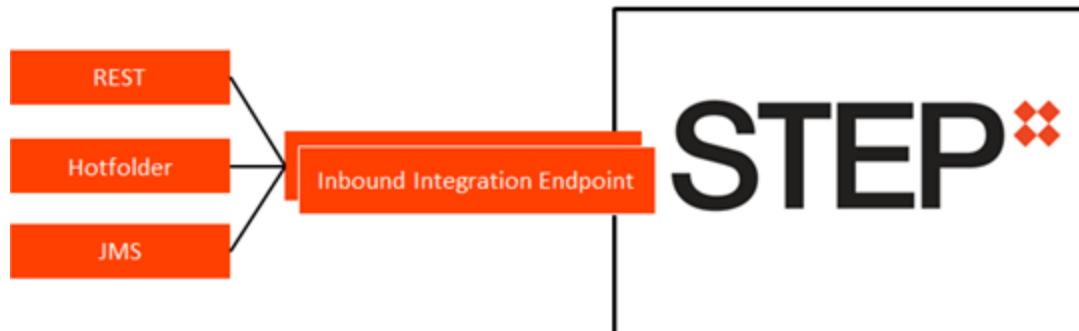
IIEP - Choose Receiver



1. Select the receiver to be used for the inbound endpoint. The available standard receivers are:
 - GDSN Data Pool Receiver (available with GDSN implementation)
 - GDSN Recipient Data Pool Receiver (available with GDSN implementation)
 - Hotfolder Receiver (default receiver and required for Asset Importer processing engine)
 - Hotfolder Receiver Using File Sequence
 - Hotfolder Receiver Using Meta Files
 - IBM Websphere MQ SSL Receiver
 - JMS Receiver
 - Oracle AQ Receiver
 - REST Receiver
2. Click the **Next** button to display **IIEP - Configure Endpoint**.

IIEP Receiver Methods

Data being imported into STEP via an IIEP can originate using any of the IIEP receiver methods described below. When the data is available, the IIEP processes based on its configuration, and then performs the import.



Import Manager does not require a receiver method since the user chooses a single file for import. For more on Import Manager, see **Import Manager**.

The following receiver options are only available when importing data using an IIEP, and some are based on a license.

Method	Description
GDSN Data Pool Receiver	The GDSN solution must be implemented fully before the GDSN Data Pool Receiver method is available and functional. See GDSN Provider documentation.
GDSN Recipient Data Pool Receiver	The GDSN solution must be implemented fully before the GDSN Recipient Data Pool Receiver method is available and functional. See GDSN Receiver documentation.
Hotfolder Receiver	Enables setup of a standard data hotfolder, typically on the application server. See Hotfolder Receiver documentation.
Hotfolder using file sequence	Enables setup of a standard data hotfolder, typically on the application server, where file names include sequence ID to determine the file processing order. See Hotfolder Receiver Using File Sequence documentation.
Hotfolder using meta files	Enables the setup of a standard data hotfolder, typically on the application server, where one or more simple .txt meta files determine the file processing order. See Hotfolder Receiver Using Meta Files documentation.

Method	Description
IBM Websphere MQ SSL	IBM Websphere MQ SSL, uses MQ series Secure Sockets Layer, and enables data exchange across IBM and non-IBM platforms. See IBM Websphere MQ SSL documentation.
JMS Receiver	Java Message Service (JMS) receivers are system dependent. By default, the JMS Receiver option lets you consume and dequeue messages on defined queues. See JMS Receiver documentation.
Oracle AQ Receiver	Oracle Advanced Queuing (Oracle AQ) enables messages to be exchanged between two systems. See Oracle AQ Receiver documentation.
REST Receiver	Representational State Transfer (REST) is a software architecture style used to design web services. See REST Receiver documentation.

Hotfolder Receiver

This default hotfolder option enables the setup of a monitored directory as the receiver. Files are placed into this folder and are processed by the integration endpoint. Typically this is on the application server or can be reached from the application server. When the endpoint polls the hotfolder and finds more than one file, the files are handled in sequence starting with the oldest file first, based on the last modified data stamp. All formats supported by the standard import functionality are available via hotfolder import.

Note: The Asset Importer Processing Engine requires the use of the Hotfolder Receiver. For more information, see the **IIEP - Configure Asset Importer Processing Engine** documentation.

To control the order of the files to be imported, set up either a **Hotfolder Receiver Using Meta Files** or a **Hotfolder Receiver Using File Sequence** instead.

Prerequisites

The location of the 'Hotfolder' and 'In folder' parameters are determined by the sharedconfig.properties file **Install.HotfolderRoot** property. Changes to the properties file are implemented when the server is restarted.

Configuration

After completing the prerequisite steps, edit the receiver of the IIEP. Use the following parameters to configure the IIEP. For information on a parameter, hover over the parameter field to display help text.

The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' panel lists eight steps: 1. Identify Endpoint, 2. Choose Receiver (highlighted), 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area is titled 'Choose Receiver' and contains the following configuration fields:

- Receiver:** A dropdown menu with 'Hotfolder Receiver' selected.
- Hotfolder:** A text input field containing 'ReceivedFiles'.
- Keep file after load:** A dropdown menu with 'Yes' selected.
- Ignore sub folders:** A dropdown menu with 'No' selected.
- In folder:** A text input field containing 'INCOMING'.

At the bottom of the wizard, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

1. For **Receiver**, choose **Hotfolder Receiver**.
2. For **Hotfolder**, enter the name used to create a folder on the application server in the 'upload' directory.
3. For **Keep file after load**, specify if files dropped in the hotfolder should be removed after processing the files.

- Yes - requires periodic manual cleanup on the server.
 - No - removes the files automatically and is the common setup to prevent the need for manual cleanup.
4. For **Ignore sub folders**, specify if subfolders within the hotfolder (which contain files) should be ignored.
- Yes - files in these folders will not be imported.
 - No - in order to process all files, regardless of their location within the hotfolder or a folder below it is the common setup.
5. For **In folder** is optional. Enter a name for an 'In' folder. The name entered is used to create a new folder under the directory specified above in the 'Hotfolder' parameter. If no folder is specified, files are dropped at the hotfolder top level.
6. Click the **Next** button to continue with the wizard, or the **Finish** button to close the wizard.

Hotfolder Receiver Using File Sequence

The 'Hotfolder using File Sequence' receiver enables setup of a standard data hotfolder where a sequence ID (included in the file name) determines the file processing order. Files are placed into this folder and are processed by the integration endpoint. Typically this is on the application server or can be reached from the application server. All formats supported by the standard import functionality are available via hotfolder import.

Prerequisites

The location of the 'Hotfolder' and 'In folder' parameters are determined by the sharedconfig.properties file **Install.HotfolderRoot** property. Changes to the properties file are implemented when the server is restarted.

Configuration

After completing the prerequisite steps, edit the receiver of the IIEP. Use the following parameters to configure the IIEP. For information on a parameter, hover over the parameter field to display help text.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
- 2. Choose Receiver**
3. Configure Endpoint
4. Configure PreProcessor
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Choose Receiver

Receiver: **Hotfolder using file sequence**

Hotfolder: **ReceivedFiles**

Keep file after load: **Yes**

Ignore sub folders: **No**

In folder: **INCOMING**

Pattern of sequence file: **ToSTEP#.xls**

Sequence Number: **1**

Back **Next** **Finish** **Cancel**

1. For **Receiver**, choose **Hotfolder using file sequence**.
2. For **Hotfolder**, the name entered is used to create a folder on the application server in the 'upload' directory.
3. For **Keep file after load**, specify if files dropped in the hotfolder should be removed after processing the files. Selecting 'Yes' requires periodic manual cleanup on the server. Selecting 'No' removes the files automatically. Common setup is to select 'No' to prevent the need for manual cleanup.
4. For **Ignore sub folders**, specify if subfolders within the hotfolder (which contain files) should be ignored. If set to Yes, files in these folders will not be imported. Common setup is to select 'No' in order to process all files, regardless of their location within the hotfolder or a folder below it.

5. For **In folder** is optional. Enter a name for an 'In' folder. The name entered is used to create a new folder under the directory specified above in the 'Hotfolder' parameter. If no folder is specified, files are dropped at the hotfolder top level.
6. For **Pattern of Sequence File** is required. Enter the relevant file pattern. For example, if the hotfolder is used to parse XML files, the pattern could be: file#.xml. In this case, a file name must begin with *file* and have the extension *xml*. The # indicates a sequence number.
7. For **Sequence Number** is required. Enter the sequence number of the first file to be processed. For example, if the first number is 1, file1.xml is processed first, then file2.xml, and so on. The sequence number will automatically increment so that it is always possible to see the number for the next file. With this option, sequences can only be skipped if you manually reconfigure the 'Sequence Number'.
8. Click the **Next** button to continue with the wizard, or the **Finish** button to close the wizard.

Hotfolder Receiver Using Meta Files

This option enables the setup of a standard data folder where one or more meta files determine the file processing order of the data files. Files are placed into this folder and are processed by the integration endpoint. Typically this is on the application server or can be reached from the application server. All formats supported by the standard import functionality are available via hotfolder import.

Prerequisites

The location of the 'Hotfolder' and 'In folder' parameters are determined by the sharedconfig.properties file **Install.HotfolderRoot** property. Changes to the properties file are implemented when the server is restarted.

A meta file is a simple .txt file dropped into the hotfolder together with files to be imported. The meta file must list the files to be processed, only one file is listed per line. Press the Enter key to create a new line within the meta file.

Configuration

After completing the prerequisite steps, edit the receiver of the IIEP. Use the following parameters to configure the IIEP. For information on a parameter, hover over the parameter field to display help text.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
- 2. Choose Receiver**
3. Configure Endpoint
4. Configure PreProcessor
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Choose Receiver

Receiver:

Hotfolder:

Pattern of meta file:

Keep file after load:

Ignore sub folders:

In folder:

1. For **Receiver**, choose **Hotfolder using meta files**.
2. For **Hotfolder**, enter the name used to create a folder on the application server in the 'upload' directory.
3. For **Pattern of meta file** is required. Enter the name of the meta file that will be used to define the order of imported files. You can enter:
 - An **exact name** of the meta file, for example, meta.txt. In this case, additional meta files cannot be dropped in the hotfolder at the same time. The hotfolder must complete the import specified in the meta file before additional meta file(s) and files can be imported.

- A **wildcard name** of the meta file, for example, sequence_*.txt. In this case, multiple meta files with names that match the wildcard name (for example, meta1.txt, meta2.txt, and so on) can exist in a hotfolder and will be processed. The hotfolder handles the meta files based on last modification date. This means that if you drop a meta1.txt and meta2.txt and the modification date in meta2.txt is older than meta1.txt, the hotfolder will process the files specified in meta2.txt before the files specified in meta1.txt.
3. For **Keep file after load**, specify if files dropped in the hotfolder should be removed after processing the files. Selecting 'Yes' requires periodic manual cleanup on the server. Selecting 'No' removes the files automatically. Common setup is to select 'No' to prevent the need for manual cleanup.
 4. For **Ignore sub folders**, specify if subfolders within the hotfolder (which contain files) should be ignored. If set to Yes, files in these folders will not be imported. Common setup is to select 'No' in order to process all files, regardless of their location within the hotfolder or a folder below it.
 5. For **In folder** is optional. Enter a name for an 'In' folder. The name entered is used to create a new folder under the directory specified above in the 'Hotfolder' parameter. If no folder is specified, files are dropped at the hotfolder top level.
 6. Click the **Next** button to continue with the wizard, or the **Finish** button to close the wizard.

IBM Websphere MQ SSL Receiver

For information on connecting to IBM Websphere MQ in a non-SSL way, see **JMS Receiver**.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. Prior to configuration, clicking the **Connection URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **WSMQSSLProviderURL** property. The elements are needed for each URL in the format [host]:[port]/[channel]:
 - [host] = hostname or IP of the Websphere MQ server
 - [port] = port number for the channel
 - [channel] = name of the channel
2. Prior to configuration, clicking the **Queue Manager** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **WSMQSSLQueueManager** property.
3. Prior to configuration, clicking the **Queue Name** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **WSMQSSLQueue** property.
4. Prior to configuration, clicking the **Key Store** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **WSMQSSLKeyStoreLocation** property.

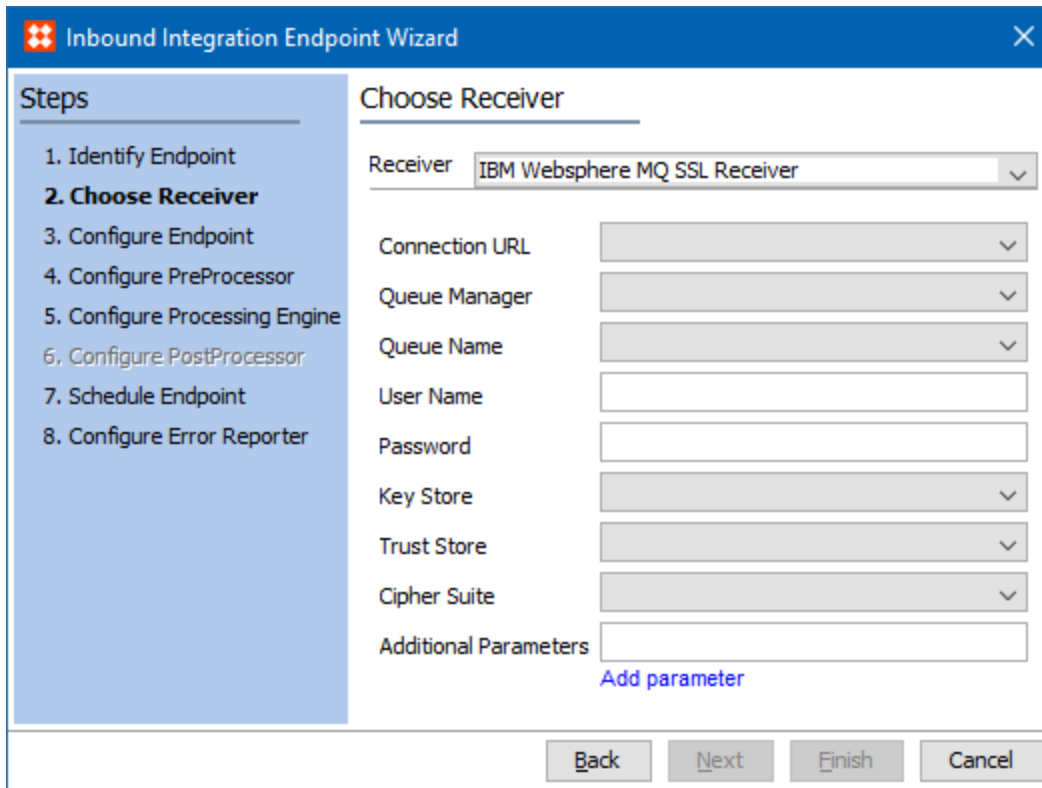
This must be in jks format, with the personal certificate for the Queue Manager. To generate, consult the manual for IBM Websphere MQ. The password for the key store must be configured in sharedconfig.properties using the case-sensitive **WSMQSSLKeyStorePassword** property.
5. Prior to configuration, clicking the **Trust Store** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **WSMQSSLTrustStoreLocation** property.

The Trust Store can be the same file as the Key Store. To generate, consult the manual for IBM Websphere MQ. The password for the trust store must be configured in sharedconfig.properties using the case-sensitive **WSMQSSLTrustStorePassword** property.
6. Prior to configuration, clicking the **Cipher Suite** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **WSMQSSLCipherSuite** property.

Set this property to the same value as SSL CipherSuite in Websphere MQ.

Configuration

After completing the prerequisite steps, edit the receiver of the IIEP. Use the following parameters to configure the IIEP. For information on a parameter, hover over the parameter field to display help text.



The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' sidebar lists eight steps: 1. Identify Endpoint, 2. Choose Receiver (highlighted), 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area is titled 'Choose Receiver' and contains a 'Receiver' dropdown menu set to 'IBM Websphere MQ SSL Receiver'. Below this are several configuration fields: 'Connection URL', 'Queue Manager', and 'Queue Name' (all dropdown menus); 'User Name', 'Password', and 'Additional Parameters' (all text input fields); and 'Key Store', 'Trust Store', and 'Cipher Suite' (all dropdown menus). A blue 'Add parameter' link is located below the 'Additional Parameters' field. At the bottom of the window are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

1. For **Receiver**, choose **IBM Websphere MQ SSL Receiver**.
2. For **Connection URL**, select the URL for connection.
2. For **Queue Manager**, select the Queue Manager.
3. For **Queue Name**, select the name of the Queue for the connection.
4. For **User Name**, if required, enter the user name that will be used to log onto the JMS provider.
5. For **Password**, if required, enter the password that will be used to log onto the JMS provider.
6. For **Key Store**, select the key store.
7. For **Trust Store**, select the trust store with the CA for the Queue Manager.
8. For **Cipher Suite**, since STEP is running on non-IBM jre, this must be the same value as configured in the Queue Manager.
9. For **Additional Parameters**, if required, click the **Add parameter** link, then enter the Key and Value pairs. For possible keys and values, consult the manual for IBM Websphere MQ.
10. Click the **Next** button to continue with the wizard, or the **Finish** button to close the wizard.

JMS Receiver

The available options for the Java Message Service (JMS) receiver are system dependent. By default, the JMS Receiver option lets you consume and dequeue messages (instead of files) on queues defined below. Supported formats include STEPXML, Generic XML, CSV (text messages). JMS is the preferred option where compatible enterprise middleware is available and there is a need for time-critical incremental updates from external systems.

If additional JME support is required, contact your Stibo Systems account manager for customizations.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. Prior to configuration, clicking the **JMS Connection Factory Name** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JMSConnectionFactoryName** property. For example, two factory names are displayed with the following property setting:

```
JMSConnectionFactoryName=1=ConnectionFactory1,2=ConnectionFactory2
```

2. Prior to configuration, clicking the **JMS Provider URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JMSProviderURL** property. For example, a single URL is displayed with the following property setting:

```
JMSProviderURL=1=tcp://[IP address]:[Port, ActiveMQ default is 61616]
```

3. Prior to configuration, clicking the **JMS Queue** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JMSQueue** property.

```
JMSQueue=1=[Queue]
```

Configuration

After completing the prerequisite steps, edit the receiver of the IIEP. Use the following parameters to configure the IIEP. For information on a parameter, hover over the parameter field to display help text.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
- 2. Choose Receiver**
3. Configure Endpoint
4. Configure PreProcessor
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Choose Receiver

Receiver:

JMS Connection Factory Class:

Connection Factory Name:

JMS Provider URL:

JMS Queue:

User Name:

Password:

Additional Parameters:

[Add parameter](#)

1. For **Receiver**, choose **JMS Receiver**.
2. For **JMS Connection Factory Class**, select one of the available JMS connection factory classes.
 - **ActiveMQInitialContextFactory**: Allows set up of a JMS receiver connecting to Apache ActiveMQ. For information, search the web.
 - **FileInitialContextFactory**: Enables setting up JMS Websphere Inbound and Outbound Integration Endpoints which reference a binding file (created from JMS Websphere Client software). The binding file is a configuration file which includes all details of how STEP should interact with JMS Websphere. For information about JMS Websphere Client Software, search the web.

Note: WMQInitialContextFactory is no longer supported, use FileInitialContextFactory instead.

2. For **Connection Factory Name**, select a connection factory name in the list.
3. For **JMS Provider URL**, select a JMS URL in the list.
4. For **JMS Queue**, select the JMS Queue to be used on Apache ActiveMQ or Websphere MQ.
5. For **User Name** and **Password**, if required to log onto the JMS provider, enter a user name and a password.
6. For **Additional Parameters**, if required, click the **Add parameter** link, then enter the Key and Value pairs. For possible keys and values, consult the manual for IBM Websphere MQ.
7. Click the **Next** button to continue with the wizard, or the **Finish** button to close the wizard.

Configuration Examples

The following images demonstrate the parameters and setup for a variety of JMS Receivers.

FileInitialContextFactory JMS Receiver

System Setup

- Global Business Rules
- Google Business Rule Group
- IEPCSVHotfolderwBAandBC
- Integration Endpoints
 - autotest_name
 - Inbound Integration Endpoints
 - CSVAssetHotfolderwBAandBC
 - CSVHotfolderwBAandBC
 - HotfolderErrorHandling
 - Hotfolder Receiver
 - HotfolderUsingFileSequence
 - Hotfolder Using Meta Files
 - HotfolderWithTermsList
 - JMS Apache
 - JMSWebsphereJNDIIN**

JMSWebsphereJNDIIN rev.0.1 - Inbound Integr

Inbound Integration Endpoint | Background Processes | Statistics | Error Log

Description

Configuration

JMS Receiver Configuration

ID	Name
> JMS Connection Factory Class	FileInitialContextFactory
> Connection Factory Name	QM.STIBO
> JMS Provider URL	
> JMS Queue	TEST.Q1
> User Name	
> Password	
> Additional Parameters	

JMS Websphere Delivery Using SSL

IBM Websphere MQ SSL Receiver Configuration	
ID	Name
> Connection URL	webspheremq-qa.stibo.com:1417/STEP.SVRCONN
> Queue Manager	QM.STIBO_SSL
> Queue Name	TEST.Q1
> User Name	
> Password	
> Key Store	file:/workarea/JMSWebsphereSSLKeystore/keyStore.jks
> Trust Store	file:/workarea/JMSWebsphereSSLKeystore/keyStore.jks
> Cipher Suite	TLS_RSA_WITH_AES_256_CBC_SHA256
> Additional Parameters	

Apache ActiveMQ JMS Receiver

Delivery Method

JMS Delivery

>	JMS Connection Factory Class	ActiveMQInitialContextFactory
>	Connection Factory Name	ConnectionFactory
>	JMS Provider URL	tcp://ATTCM3S9:61616
>	JMS Queue	testqueue
>	Binary Payload	No
>	User Name	
>	Password	
>	Additional Parameters	
>	Edit Delivery	

Oracle AQ Receiver

Oracle Advanced Queuing (Oracle AQ) enables messages to be exchanged between two systems.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. Prior to configuration, clicking the **Connection URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **OracleAQReceiverConnectionURL** property. For example, two URLs are displayed with the following property setting:

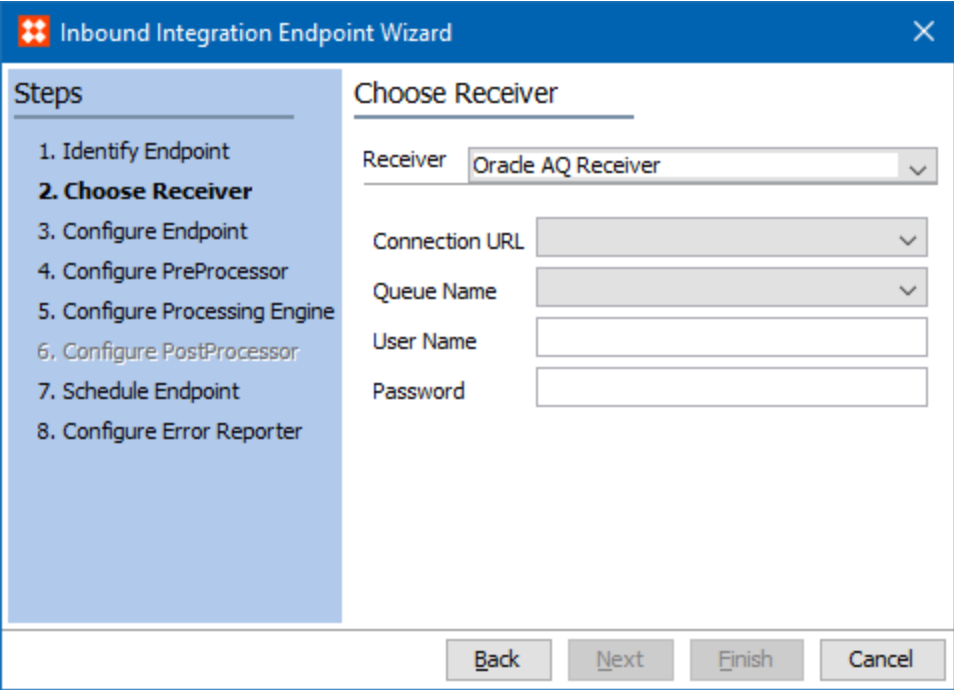
```
OracleAQReceiverConnectionURL=1=ConnectionURL1,2=ConnectionURL2
```

2. Prior to configuration, clicking the **Queue Name** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **OracleAQReceiverQueueName** property. For example, a single URL is displayed with the following property setting:

```
OracleAQReceiverQueueName=1=QueueName
```

Configuration

After completing the prerequisite steps, edit the receiver of the IIEP. Use the following parameters to configure the IIEP. For information on a parameter, hover over the parameter field to display help text.



The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' sidebar lists eight steps: 1. Identify Endpoint, 2. Choose Receiver (highlighted), 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area is titled 'Choose Receiver' and contains a 'Receiver' dropdown menu with 'Oracle AQ Receiver' selected. Below this are four input fields: 'Connection URL' (a dropdown menu), 'Queue Name' (a dropdown menu), 'User Name' (a text box), and 'Password' (a text box). At the bottom of the window are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

1. For **Receiver**, choose **Oracle AQ Receiver**.
2. For **Connection URL**, select a URL pointing to Oracle AQ.
3. For **Queue name**, select an Oracle AQ queue name.
4. For **User Name**, enter a user name to be used to log on Oracle.
5. For **Password**, enter a password to be used to log on Oracle.
6. Click the **Next** button to continue with the wizard, or the **Finish** button to close the wizard.

REST Receiver

REST stands for Representational State Transfer. It is a software architecture style that you can use to design web services. REST as an architecture style does not require processing and is naturally more flexible. Unlike SOAP, REST doesn't have to use XML to provide the response. You can find REST-based Web services that output the data in Command Separated Value (CSV), JavaScript Object Notation (JSON), and Really Simple Syndication (RSS). Supported formats include XML files using STEPXML or Generic XML.

If you want to learn more about REST, search for resources on the web.

For information about how to use the STEP REST API, click the **STEP API Documentation** button on the STEP WebStart page.

Prerequisites

The path used for the file upload is system specific, and is automatically used when new REST receivers are created. In the configured path, a folder named REST is created and files that are uploaded using REST API appear in this folder.

When an IIEP has been configured to use the REST Receiver, XML files can be posted the IIEP via HTTP POST to the following URL:

```
[Host]/restapi/integrationendpoints/[Endpoint ID]/upload?context=[Existing Context]&workspace=Main
```

The request header 'Content-Type' must have the value 'text/xml' and basic authorization must be used. For example, the 'Authorization' header must have the value 'Basic ' combined with a 64 bit encoding of [Username]: [Password]. For example, 'Basic c3RlcHN5czpzdGVwc3lz'.

Notice that in the URL shown above, the context and workspace qualifiers have no significance, but they are required for the REST API.

The POST will in itself invoke the endpoint, so there is no reason to schedule a REST-based endpoint. The resource for the REST POST invoke request is:

```
[Host]/restapi/integrationendpoints/[Endpoint ID]/invoke
```

Configuration

After completing the prerequisite steps, edit the receiver of the IIEP. Use the following parameters to configure the IIEP. For information on a parameter, hover over the parameter field to display help text.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
- 2. Choose Receiver**
3. Configure Endpoint
4. Configure PreProcessor
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Choose Receiver

Receiver: REST Receiver

Keep file after load: Yes

Back Next Finish Cancel

1. For **Receiver**, choose **REST Receiver**.
2. For **Keep file after load**, specify if files dropped in the hotfolder should be removed after processing the files. This option is available because files posted via REST are temporarily stored in a directory on the application server (under Hotfolder root).
 - **Yes** requires periodic manual cleanup on the server.
 - **No** removes the files automatically. Common setup is to select 'No' to prevent the need for manual cleanup.
3. Click the **Next** button to continue with the wizard, or the **Finish** button to close the wizard.

IIEP - Configure Endpoint

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
- 3. Configure Endpoint**
4. Configure PreProcessor
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Endpoint

Processing

Processing Engine:

Transactional settings:

Context

Workspace:

Context:

Queue Settings

Queue for endpoint:

Queue for endpoint processes:

Maximum number of waiting processes:

Maximum number of old processes:

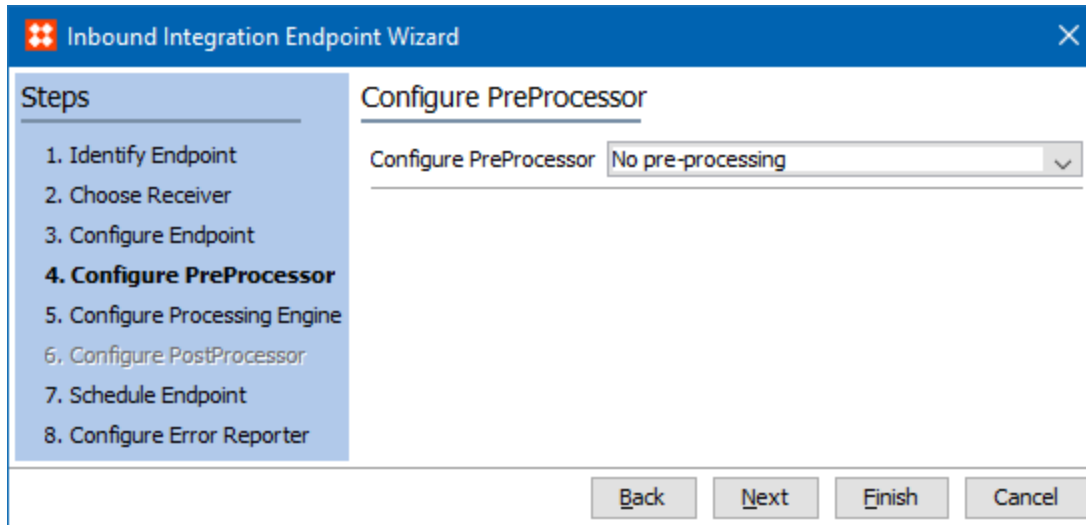
Maximum age of old processes:

Number of messages per background process:

1. For **Processing Engine** on a standard STEP system, you can select an available option:
 - **STEP Importer** - enables you to use the same import functionality as in the STEP Import Manager. Additional setup is required on the upcoming Configure Processing Engine step.
 - **Asset Importer** - allows you to use the same import functionality as Asset Importer. Additional setup is required on the upcoming Configure Processing Engine step.
 - **GDSN Inbound message processor** - available when the GDSN solution is implemented.
 - **GDSN Receiver Inbound message processor** - available when the GDSN solution is implemented.
 - **STEP Match and Merge Importer** - available when the Matching and Merging deduplication solution is implemented, and requires additional configuration.
2. For **Transactional Settings** specify if the processes will be chain, strict, or none. For details on the effects of each option, see **Inbound Integration Endpoint Transactional Settings** documentation.
3. The queue size of a background process with strict or chain transactional setting must be 1. Only a transactional setting of None allows multiple background processes to work in parallel / concurrently. For more information, see **Inbound Integration Endpoint Transactional Settings** documentation.

4. For the **Context** options, select the **Workspace** and **Context** that will receive the imported data. Common setup is to use Main since it allows objects that require approval to be displayed in the Main workspace before being approved and moved to the Approved workspace. For externally maintained attributes or global maintained objects, importing to the Main workspace also adds the data to the Approved workspace, just as it does when adding data manually.
 - The selected context and workspace affects import formats where an import context and workspace is not included in the import file.
 - If the endpoint imports STEPXML files, the context and workspace specified in the STEPXML file override the context and workspace selected in this option.
5. For **Queue Settings**, set up the background processes to be used for the endpoint.
 - For **Queue for endpoint**, enter a name for the queue that is used by the IIEP Background Process to poll the endpoint. The first time you activate the endpoint, a queue with the specified name is created if it does not already exist. If in doubt about how to populate this parameter, create a new queue for the IIEP, for example with a name that includes the IIEP ID.
 - For **Queue for endpoint processes**, enter a name for the queue that is used by the Background Processes started by the endpoint to handle the actual import. The queue is automatically created on the system if it does not already exist. High priority integrations or integrations with long-running processes should typically have their own queue, for example with a name that includes the IIEP ID.
 - For **Maximum number of waiting processes**, specify how many background processes the endpoint is allowed to start. A background process can be in the state "waiting" or "active". This field is disabled if the 'Transactional settings' parameter (discussed above) is Strict.
 - For **Maximum number of old processes**, specify the maximum number of ended background processes the system is allowed to retain. When the set limitation has been exceeded, this auto-cleanup function deletes succeeded and ended background processes that were started when the IIEP was invoked. The oldest processes are deleted first. The default for this setting is 100. All background processes, regardless of their ending state (CompletedWithErrors, Aborted, Failed, Succeeded), are subject to the auto-delete functionality. Those that are in a 'Suspended' state are exempt.
 - For **Maximum age of old processes**, specify the maximum age of ended background processes the system is allowed to retain. To populate this value, the following case-sensitive notations should be used: y = years, M = months, w = weeks, d = days, h=hours, m = minutes, and s = seconds. The default for this setting is 1wk. All background processes, regardless of their ending state (CompletedWithErrors, Aborted, Failed, Succeeded), are subject to the auto-delete functionality. Those that are in a 'Suspended' state are exempt.
 - For **Number of messages per background process**, specify the number of messages that will be handled in one background process before a new background process is generated. This is useful if the endpoint handles many small messages that would otherwise generate a large number of background processes. Common setup is to let several messages / files be handled per background process. This field is only enabled if the transactional setting is strict.
6. Click the **Next** button to display **IIEP - Configure PreProcessor** when available, or display **IIEP - Configure Processing Engine**.

IIEP - Configure PreProcessor



The screenshot shows the 'Inbound Integration Endpoint Wizard' window. The title bar is blue with a red Stibo logo and the text 'Inbound Integration Endpoint Wizard'. The window is divided into two main sections. On the left, a 'Steps' sidebar lists eight steps: 1. Identify Endpoint, 2. Choose Receiver, 3. Configure Endpoint, 4. Configure PreProcessor (highlighted in blue), 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area on the right is titled 'Configure PreProcessor' and contains a dropdown menu labeled 'Configure PreProcessor' with 'No pre-processing' selected. At the bottom of the window, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

A preprocessor has access to the files / messages delivered by the receiver and can manipulate or discard them. This allows you to create extensions to STEP in order to handle your specific inbound data processing needs. For assistance in creating a preprocessor, contact Stibo Systems Support.

1. If necessary, choose a preprocessor and follow the steps in the linked topics for setup:
 - **No pre-processing option** is selected by default.
 - **XML Normalizer** preprocessor is available to make minor modifications to the layout of an inbound XML file so that STEP can process it using the Generic XML format. It allows a specified tag to be copied from one place within the file to another place within the same tag. For details, see the **IIEP - Configure XML Normalizer PreProcessor** documentation.
 - For details on the **XSLT** preprocessor, see **IIEP - Configure Transformation by XSLT PreProcessor** documentation.
 - **Transformation by Import Configuration** can be used to transform Excel and CSV files to STEPXML on import.
2. Click the **Next** button to display **IIEP - Configure Processing Engine**.

IIEP - Configure Transformation by XSLT PreProcessor

The Transformation by XSLT preprocessor allows an XSLT 2.0-compliant stylesheet to be used during import to transform valid inbound XML into STEPXML, or into non-STEPXML that can then be imported using the Generic XML format option.

This allows the XML file to be parsed via an XSLT. The data is extracted and represented in a tabular format that can be handled with the standard Import Manager functionality.

Note: To use this preprocessor successfully, the incoming file or message size must be 50 MB or smaller.

Prerequisites

Before configuring the Transformation by XSLT Preprocessor, you must first:

1. Perform the **Initial Setup for XSLT Stylesheets** - this one-time setup is described below.
2. **Create an XSLT Stylesheet Object** - these steps must be performed for each separate stylesheet needed and is described below.

Once both of the prerequisites have been completed, proceed to the **Configure the Transformation by XSLT Preprocessor** section below.

Initial Setup for XSLT Stylesheets

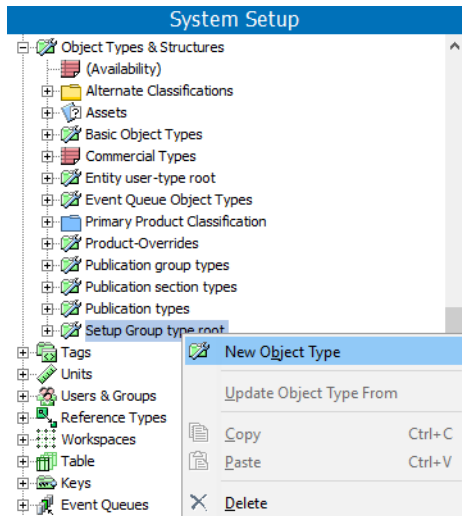
Creating an XSLT stylesheet setup group must only be performed once, unless additional levels of organization are desired.

Navigate to System Setup > Object Types & Structures > Setup Group type root, and determine if an entry exists for the XSLT Stylesheet.

- If an XSLT Stylesheet setup group does not exist, continue with the steps below to create a setup group, link the XSLT object type, and create an instance of the XSLT object.
- If an XSLT Stylesheet setup group exists, but has no children, proceed to the **Link the XSLT Stylesheets object type to setup group** section below.
- If an XSLT Stylesheet setup group exists, and has an XSLT Stylesheet Object as a child, proceed directly to the **Creating an XSLT Stylesheet** section below.

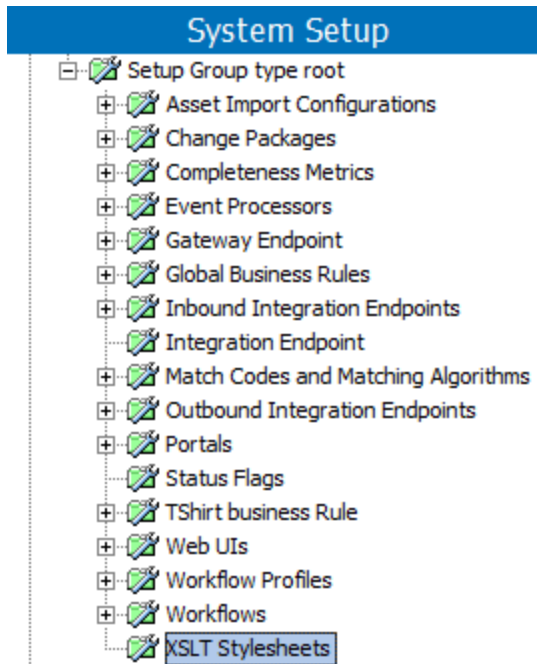
Create the XSLT stylesheet setup group

1. In System Setup, expand **Object Types & Structures**.
2. Right-click 'Setup Group type root', and choose **New Object Type**.



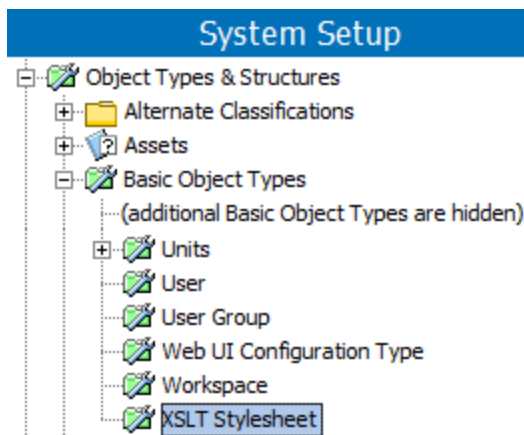
3. Enter an **ID** and a **Name**, select any required Dimension Dependencies, and click **Create**.

4. The new setup group appears in System Setup under 'Object Types & Structures' as a child in the **Setup Group type root**.

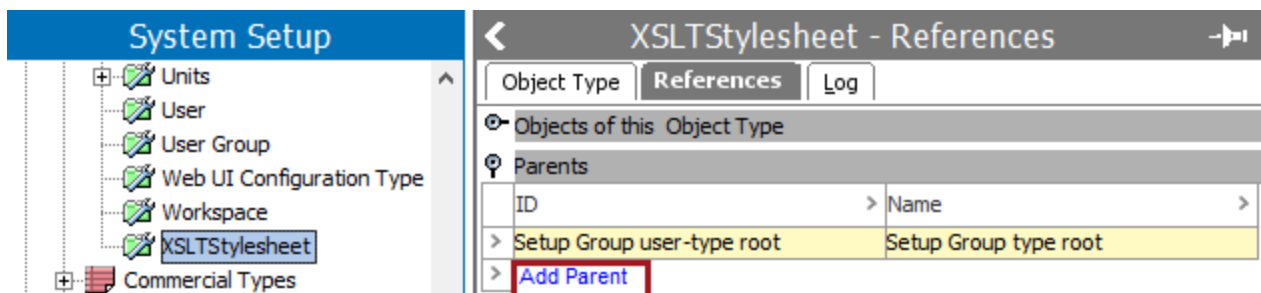


Link the XSLT stylesheet object type to the XSLT setup group

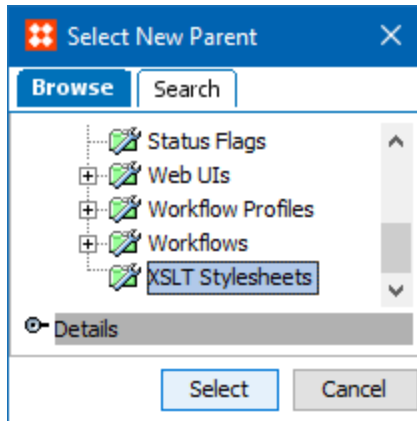
1. In Object Types & Structures > expand Basic Object Types > select **XSLT Stylesheet**.



2. On the References tab > Parents flipper > click the **Add Parent** link.



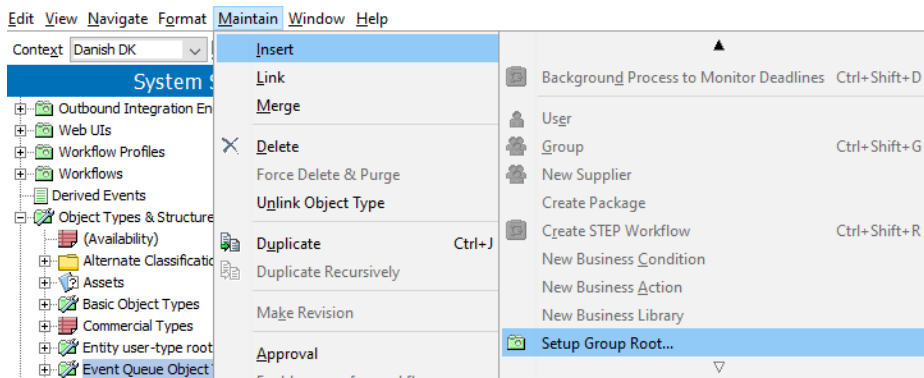
3. In the Select New Parent dialog, select the setup group type root you created, and click **Select** to make it a valid parent.



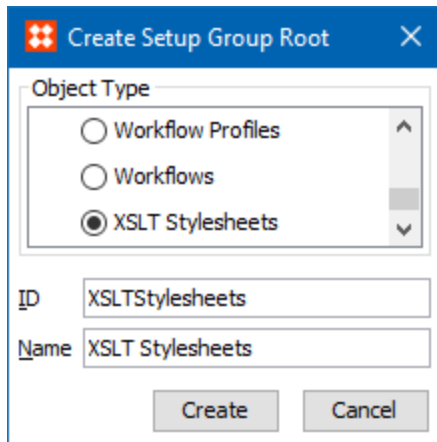
The Setup Group type root node now displays XSLT Stylesheets with a child of XSLT Stylesheet.

Create an instance of the XSLT Stylesheet object

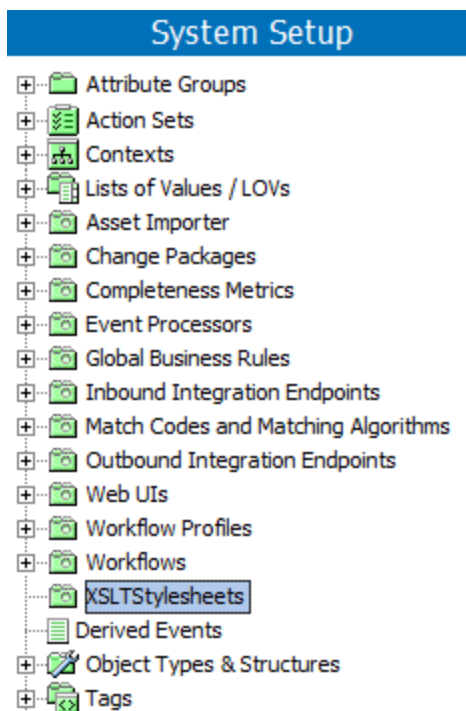
1. On the System Setup tab, select any object in the **System Setup** hierarchy to activate the following **Maintain** menu selection.
2. Click the Maintain menu, point to Insert, and select **Setup Group Root**.



3. In the Create Setup Group Root dialog, select the XSLT Stylesheets object type, enter an **ID** and a **Name**, and click **Create**.



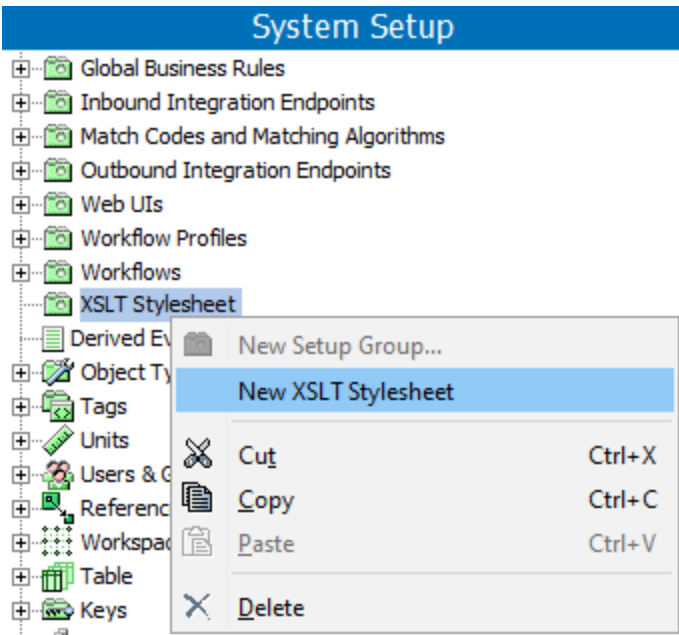
A setup group is created as a node in the System Setup hierarchy. XSLT Stylesheets can now be created under this new node.



Create an XSLT Stylesheet Object

Once a setup group node has been created to hold XSLT Stylesheets, you can add your own stylesheet, which will be available for selection in the IIEP preprocessor.

1. In System Setup, select your XSLT Stylesheet node, right-click, and choose **New XSLT Stylesheet**.



2. Enter an **ID** and a **Name**, and click **Create**.

A screenshot of the 'Create New XSLT Stylesheet' dialog box. It has a title bar with a close button. Inside, there are two text input fields: 'ID' and 'Name'. Both fields contain the text 'AnXSLTStylesheet'. At the bottom right, there are two buttons: 'Create' and 'Cancel'.

3. Select the new stylesheet object that displays below the XSLT node and click the **Edit** link.

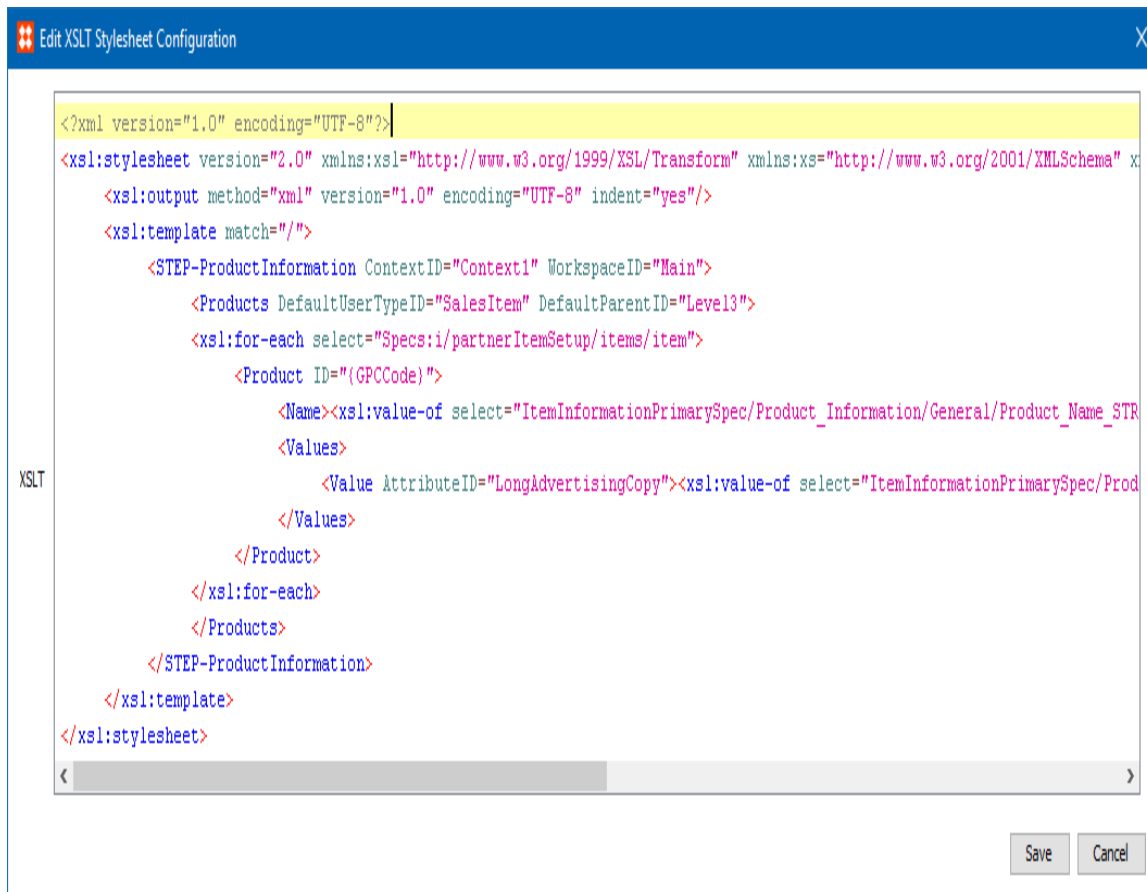
A screenshot of the 'System Setup' application window. The 'XSLT Stylesheet' item in the tree view is expanded, and a sub-item named 'An XSLT Stylesheet' is selected and highlighted with a blue box.

A screenshot of the configuration page for 'An XSLT Stylesheet rev.0.1'. The page has a title bar and tabs for 'XSLT Stylesheet', 'Log', and 'Status'. Below the tabs is a 'Description' section with a table of properties. The 'Edit' link at the bottom is highlighted with a red box.

Name	Value
ID	AnXSLTStylesheet
Name	An XSLT Stylesheet
Object Type	XSLT Stylesheet
Revision	0.1 Last edited by USERJ on Mon Apr 24 15:52:54 EDT 2017
Path	XSLT Stylesheet/An XSLT Stylesheet

Edit

4. Type or copy your XSLT 2.0-compliant stylesheet into the text box, and click the **Save** button.



The XSLT stylesheet is now ready for selection in the IIEP wizard.

Configure the Transformation by XSLT Preprocessor

Once the prerequisite tasks have been completed, you are ready to configure the preprocessor in the IIEP wizard.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
- 4. Configure PreProcessor**
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure PreProcessor

Configure PreProcessor: Transformation by XSLT

XSLT Stylesheet: An XSLT Stylesheet (AnXSLTStylesheet)

XSLT preprocessor to be used for transforming small XML files/messages.
Size of file/messages to handle must not exceed 50 MB.
Referenced XSLT Stylesheet must hold XSLT 2.0 compliant stylesheet.

Back Next Finish Cancel

1. In the **Configure PreProcessor** dropdown, select **Transformation by XSLT**.
2. In the **XSLT Stylesheet** parameter, click the ellipsis button (...) to display the Select XSLT Stylesheet dialog.
3. Select your stylesheet and click the **Select** button.

Select XSLT Stylesheet

Browse Search

.....XSLT An XSLT Stylesheet

Details

Select Cancel

The importer will use the selected XSLT stylesheet to transform inbound XML file, so that STEP can process it using STEPXML or Generic XML format, based on the selection in the Configure Processing Engine step.

4. On the wizard dialog, click the **Next** button to display **IIEP - Configure Processing Engine**.

IIEP - Configure XML Normalizer PreProcessor

While this preprocessor can be used for any inbound XML file, it has been useful for GDSN solutions when processing the following scenarios:

- The GDSN BMS Receiver requires a flattened version of the recursive XML structures that are in the 'tradelItem' tags in the XML when dealing with child nodes in the packaging hierarchy.
- A GDSN CIC is sent for the top node in the packaging hierarchy and contains the CIC status for all nodes in the hierarchy.

Important: For a successful import, verify that the node data is valid to exist in the new location.

For a simple example, as shown in the following XML input file (which could be available to the IIEP via the selected receiver method), although the 'AvailableFrom' tag is present in the file, it is required by STEP to be included at a lower level. The import file includes it at the 'Items' level, but it should be imported into STEP as an attribute on each of the two products within each 'Item' tag.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemLoad>
  <Items>
    <AvailableFrom>2015-01-01 00:00:00</AvailableFrom>
    <Item>
      <EAN>9900524977488</EAN>
      <ProductName>AC-UZ444</ProductName>
    </Item>
    <Item>
      <EAN>9900524977490</EAN>
      <ProductName>AC-UZ311</ProductName>
    </Item>
  </Items>
</ItemLoad>
```

Configure and Use the XML Normalizer

1. In the **Configure PreProcessor** dropdown, select **XML Normalizer**.
2. Use the following parameters to identify the text to be copied and the new location for it to be copied.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
- 4. Configure PreProcessor**
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure PreProcessor

Configure PreProcessor: XML Normalizer

XPath to select the file type:

Search for nodes to copy below:

Name of new tag:

Node to copy:

The XML Normalizer is use to normalize an XML file so that it is possible to use generic XML to import the file. This is done by copying selected parts of the XML file to other places in the XML file. The search path specifies a path where the nodes to copy are found. The found nodes are copied to a new tag below the search path. The name of the new tag is specified. The search path may be an absolute path (/root/parent) or an relative path (/parent) matching any path containing the parent tag in the XML file. The node to copy is relative to the parent, i.e. child would copy the child node in /root/parent/child.

Back Next Finish Cancel

- In **XPath to select the file type**, add an XPath if needed. This parameter is optional.
- In **Search for nodes to copy below**, identify the path of the parent tag where the copied text should be inserted. This can be an absolute path or a relative path matching any path containing the parent tag in the XML file.

Using the original example input file above, the path `/ItemLoad/Items/Item` indicates that the copied node will be placed within the 'Item' node.


Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
- 4. Configure PreProcessor**
5. Configure Processing Engine

Configure PreProcessor

Configure PreProcessor

XPath to select the file type

Search for nodes to copy below

```

<?xml version="1.0" encoding="utf-8"?>
1 <ItemLoad>
2   <Items>
    <AvailableFrom>2015-01-01 00:00:00</AvailableFrom></AF_New>
3   <Item>
      <EAN>9900524977488</EAN>
    
```

- In **Name of new tag**, type the new tag identifier. You must use different text than is currently used to additionally identify the tag that is copied. This tag must appear in the Sample file supplied in the 'Select Sample File' sub-step of the Configure Processing Engine wizard step. For more information on the sample file, see **IIEP - Configure STEP Importer Processing Engine** documentation.

In the wizard image, 'AF_New' is the tag that will be added as a prefix to the information being copied. The normalized XML file below shows the newly copied tag with the '<AF_New>' wrapper.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
- 4. Configure PreProcessor**
5. Configure Processing Engine
6. Configure PostProcessor

Configure PreProcessor

Configure PreProcessor: XML Normalizer

XPath to select the file type:

Search for nodes to copy below: /ItemLoad/Items/Item

Name of new tag: **AF_New**

```
<?xml version="1.0" encoding="utf-8"?>
<ItemLoad>
  <Items>
    <Item>
      <EAN>2700524977488</EAN>
      <ProductName>AC-UZ444</ProductName>
      <AF_New><AvailableFrom>2015-01-02 00:00:00</AvailableFrom></AF_New>
```

- In **Node to copy**, identify the path and the tag to be copied.

Using the original example input file, the node to copy (AvailableFrom) is currently located at /ItemLoad/Items/AvailableFrom.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
- 4. Configure PreProcessor**
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint

Configure PreProcessor

Configure PreProcessor
XML Normalizer

XPath to select the file type

Search for nodes to copy below
/ItemLoad/Items/Item

Name of new tag
AF_New

Node to copy
/ItemLoad/Items/AvailableFrom

```

<?xml version="1.0" encoding="utf-8"?>
1<ItemLoad>
2<Items>
3<AvailableFrom>2015-01-01 00:00:00</AvailableFrom>
  <Item>
    <EAN>9900524977488</EAN>
  
```

3. Include the new tag in the expected location, within the sample file supplied on the **Select Sample File** sub-step. For details, see the **IIEP - Configure STEP Importer Processing Engine** documentation.

Using the original example input file, you could create the following Sample File which includes the new tag in the required location (as determined by the XML Normalizer parameters) :

```

<?xml version="1.0" encoding="utf-8"?>
<ItemLoad>
  <Items>
    <Item>
      <EAN>2700524977488</EAN>
      <ProductName>AC-UZ444</ProductName>
      <AF_New><AvailableFrom>2015-01-02 00:00:00</AvailableFrom></AF_New>
    </Item>
  </Items>
</ItemLoad>

```

4. Include the new tag in the expected location, within the parsing template supplied on the **Select Format** sub-step. For details, see the **IIEP - Configure STEP Importer Processing Engine** documentation.

Using with the original example, you could create the following Template which includes the instruction to retrieve data for the new tag:

```
<?xml version="1.0" encoding="utf-8"?>
<ItemLoad>
  <Items>
    <Item>
      <?Record?>
        <EAN><?Source?></EAN>
        <ProductName><?Source?></ProductName>
        <AF_New><AvailableFrom><?Source?></AvailableFrom></AF_New>
    </Item>
  </Items>
</ItemLoad>
```

5. Click the **Next** button to display **IIEP - Configure Processing Engine**.

Preprocessor Results

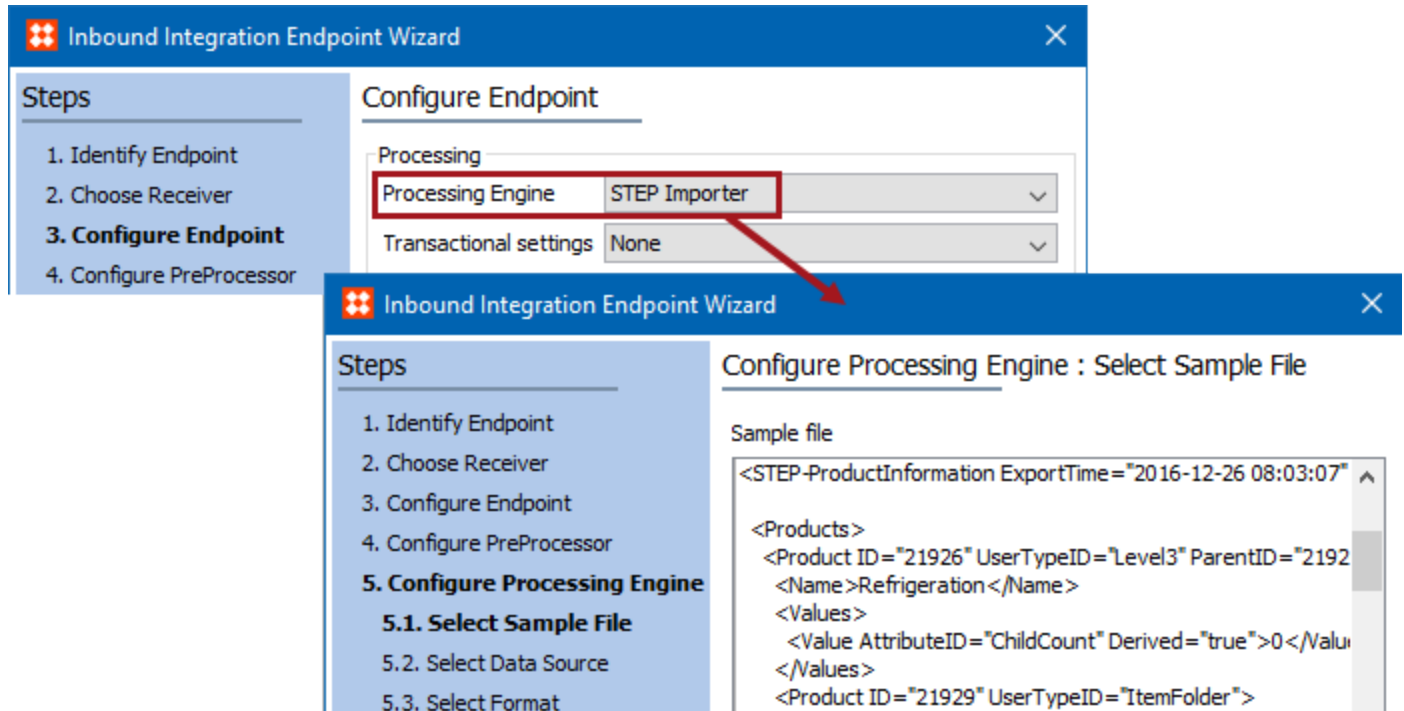
Continuing the original example, the input file is normalized by the preprocessor as follows:

```
<ItemLoad>
  <Items>
    <AvailableFrom>2015-01-01 00:00:00</AvailableFrom>
    <Item>
      <EAN>9900524977488</EAN>
      <ProductName>AC-UZ444</ProductName>
      <AF_New><AvailableFrom>2015-01-01 00:00:00</AvailableFrom></AF_New>
    </Item>
    <Item>
      <EAN>9900524977490</EAN>
      <ProductName>AC-UZ311</ProductName>
      <AF_New><AvailableFrom>2015-01-01 00:00:00</AvailableFrom></AF_New>
    </Item>
  </Items>
</ItemLoad>
```

And ultimately, the STEP attributes are updated based on the normalized Generic XML, as shown in the first product from the input file.

AC-UZ444 rev.0.1 - Product				
Product	Sub Products	References	Referenced By	Images & Documents
	Name	>	>	Value
>	ID			SalesItem-111203
>	Name			AC-UZ444
>	Object Type			Sales Item
>	Revision			0.1 Last edited by USERY on Mon Feb 27
>	Approved			 Never Been Approved
>	Available From			2015-01-01 00:00:00
>	EAN			9900524977488

IIEP - Configure Processing Engine



The Processing Engine is responsible for performing the actual import. The Processing Engine can be replaced via custom extension, for example, if some special format not supported in the standard STEP Importer is to be handled.

The prior Configure Endpoint step determines the options available for the Configure Processing Engine step.

1. Select the processing engine and use the following topics to configure the necessary sub-steps. Options may include:
 - Asset Importer
 - STEP Importer
 - GDSN inbound message processor - only available when the GDSN solution is implemented
 - GDSN Receiver Inbound message processor - only available when the GDSN solution is implemented
 - STEP Match and Merge Importer - only available when the Matching & Merging deduplication solution is implemented
2. Click the **Next** button to display **IIEP - Configure PostProcessor** when available, or display **IIEP - Schedule Endpoint**.

IIEP - Configure Asset Importer Processing Engine

The screenshot shows the 'Inbound Integration Endpoint Wizard' window, specifically the 'Configure Processing Engine : Configuration' step. On the left, a 'Steps' sidebar lists 8 steps, with '5. Configure Processing Engine' and its sub-step '5.1. Configuration' highlighted. The main area contains the following configuration options:

- Asset Importer Configuration:** A dropdown menu set to 'Image Importer (Image Importer)' with a three-dot menu icon to its right.
- Handle content as:** A dropdown menu set to 'Files'.
- Override configuration options for subfolders:** A table with columns 'Folder', 'Configuration Step', and 'Configuration'.

Folder	Configuration Step	Configuration
> JPEG	Import Validator	Override parameters
> TIFF	Import Validator	Override parameters
> GIF	Import Validator	Override parameters
> Add Folder Override		

At the bottom of the wizard are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

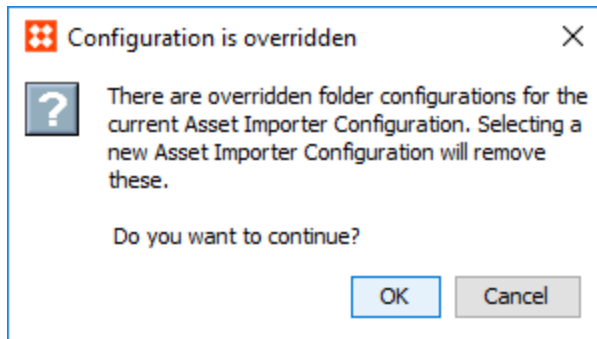
When using an Asset Importer endpoint, it is necessary to specify an asset importer configuration and define how the imported content will be received.

Prerequisites

The following settings are important when using this processing engine:

- On the Choose Receiver step, the 'Hotfolder Receiver' option must be selected as defined in the **IIEP - Choose Receiver** topic.
- PostProcessors are not available for the Asset Importer.
- If desired, folder overrides can also be configured, allowing users to replace certain aspects of the asset import configuration for sub folders within the top-level hot folder. For required configuration, see the **Asset Importer Configuration** section of the **Asset Importer** documentation.

Note: Note that if you attempt to change the asset importer configuration *after* configuring overrides, a prompt will appear warning you that the overrides will be removed.



Configuration

1. In the **Asset Importer Configuration** parameter, click the ellipsis button (...) and select an asset importer configuration from the node list. Click **Select**.
2. In the **Handle content as** parameter, define how files will be delivered to the asset importer by selecting the appropriate receiver option from the dropdown. The options include:
 - **Files** - This option should be used when importing single asset files into STEP. No additional parameters are required.
 - **Metafile** - This option should be used when a metadata file is necessary to supply information related to the assets, which may be the case if the assets and meta file reside in different locations. By configuring this parameter, users can point imported metadata files to related assets that are about to be imported or updated (via a path accessible by the STEP application server process). See the **Metadata File Format** section below.

Metafile Error Handling *	Report missing files
Filename Column Header *	Filename
Column Delimiter *	,

* Mandatory Fields

While importing the assets using the metafile, note the following required locations for the files:

- Assets are placed in the default hotfolder root (not the configured asset hotfolder). For example, while adding assets on the application server, the path will be `\[hotfolder for Windows system]` or `/upload/[hotfolder for Linux system]`.
- Metafile (the .TXT file) is placed in the `[configured asset folder]/IN` directory.

The **Metafile Error Handling** parameter determines how the errors are handled. Options include: 'Don't report inconsistencies', 'Report missing files', and 'Report all inconsistencies'. The default selection, 'Report missing files', should be selected if errors should be reported when the metafile references assets which cannot be found. 'Don't report inconsistencies' should be selected if inconsistencies between the metafile references and the assets found should not be reported as errors. This means that 'Report all

inconsistencies' should be selected if any and all inconsistencies between the metafile references and the assets should be reported as errors.

In the **Filename Column Header** parameter, specify the column heading of the metafile that contains the asset names. Additionally, a relative path can be used in the file name if the corresponding asset is in a subfolder below the root hotfolder directory. For example, 'Supplier1/Asset.jpg' corresponds to the asset name 'Asset.jpg' found within the 'Supplier1' subfolder.

In the **Column Delimiter** parameter, specify which character should be used to separate each of the columns in the metadata file. The default value is a comma (,) due to backward compatibility, but it is recommended to use either a semicolon or a pipe character, which is less likely to be included in any metadata fields.

- **Zip File** - This option should be used when importing zip files that only contain assets. The file is unpacked during the import process and all containing asset files are imported according to the remaining configuration options. No additional parameters are required.
- **ZIP Metafile** - This option should be used when importing a ZIP file that contains both the metadata file and the corresponding asset files. When importing with this configuration, asset placeholders are created from the metafile, and assets are subsequently loaded into STEP to update the placeholder content. See the **Metadata File Format** section below.

Name of metafile within ZIP file *	Assetfiles.csv
Metafile Error Handling *	Report missing files ▼
Filename Column Header *	Filename
Column Delimiter *	,

* Mandatory Fields

In the **Name of the metafile within ZIP file** parameter, indicate the file name of the metafile so the system can distinguish between it and the asset files. Valid file formats are .CSV and .TXT.

The **Metafile Error Handling** parameter determines how the errors are handled. Options include: 'Don't report inconsistencies', 'Report missing files', and 'Report all inconsistencies'. The default selection, 'Report missing files', should be selected if errors should be reported when the metafile references assets which cannot be found. 'Don't report inconsistencies' should be selected if inconsistencies between the metafile references and the assets found should not be reported as errors. Naturally this means that 'Report all inconsistencies' should be selected if any and all inconsistencies between the metafile references and the assets should be reported as errors.

In the **Filename Column Header** parameter, specify the column heading of the metafile that contains the asset names. Additionally, a relative path can be used in the file name if the corresponding asset is in a subfolder below the root hotfolder directory. For example, 'Supplier1/Asset.jpg' corresponds to the asset name 'Asset.jpg' found within the 'Supplier1' subfolder.

In the **Column Delimiter** parameter, specify which character should be used to separate each of the columns in the metadata file. The default value is a coma (,) due to backwards compatibility, but it is recommended to use either a semicolon or a pipe character, which is less likely to be included in any metadata fields.

3. In the **Override configuration options for subfolders** area of the wizard, click the **Add Folder Override** link to add an override to the existing configuration. This will add a row to the override screen, in which an individual configuration aspect can be specified.

Override configuration options for subfolders:

The screenshot shows a table with three columns: 'Folder', 'Configuration Step', and 'Configuration'. The 'Folder' column has a dropdown arrow next to it. The 'Configuration Step' column has a dropdown menu with 'Import Validator' selected. The 'Configuration' column has a button labeled 'Override parameters'. A red arrow points from the 'Add Folder Override' link in the 'Folder' column to the 'Override parameters' button. The dialog box 'Configuration Parameters' is open, showing fields for 'Width Dimension (Pixels)', 'Height Dimension (Pixels)', 'Valid Color Spaces', 'Valid MIME Types', 'DPI', and 'Max File Size (MB)'. The 'Max File Size (MB)' field is set to 100. The dialog box has 'OK' and 'Cancel' buttons.

Folder	Configuration Step	Configuration
> TIFF	Import Validator	Override parameters
> JPEG		
> Product Images		
> Add Folder Override		

- In the **Folder** field, specify a subfolder to override. If only one configuration difference is required between the parent and child folder, only a single configuration (row) is needed for each subfolder. However, if more than one configuration difference exists between the parent and child, each difference requires an individual row to be added (with the Folder name shared between multiple rows).
- In the **Configuration Step** dropdown, choose which aspect of the configuration you want to override. Almost every step in the Asset Importer configuration wizard is available for override, including: Content Importer, Import Validator, Hierarchy Builder, Asset Matcher, Product Linker, Approver, Business Rules, Metadata Importer, and Workflow Handler. The Configuration Step selection determines the dialog displayed by the Override parameters button (below).
- In the **Configuration** field, click the **Override parameters** button and make any of the necessary changes in the screen that appears and click **OK**. The parameters presented are based on the Configuration Step selection.
- Additional options are available by clicking the arrow in the first column of the override configuration options table. This displays a menu with options including Add Folder Override and Remove Folder Override.

By properly utilizing override subfolders, a single asset importer configuration can be used to control an entire hierarchy of hotfolders. All subfolders will inherit the asset importer configuration as normal, but those

configured in this way can override specific aspects of the base configuration, allowing for flexibility within a single asset importer configuration.

For example, different subfolders may be associated with different MIME Types (such as JPEG, TIFF, etc.). As part of the override configuration pictured above, this use case is addressed by overriding the 'Import Validator' step of the importer configuration and changing the 'MIME Type' parameter to '/jpeg'.

4. Click the **Next** button to display **IIEP - Schedule Endpoint**.

Metadata File Format

By configuring an IIEP with the Metafile or ZIP Metafile receiver options, the asset importer is able to process metadata files. To import such a file, a header row must be included in the imported file or the metadata file to be imported, and values must be provided for each column that references each asset.

The standard metafile configuration options described in the above section allow for a maximum of three data fields to be read from the file. However, if the 'Metadata Importer' is configured, the asset importer configuration can support additional columns, each representing an asset metadata field to be imported. If additional information is included in the file but not mapped in the metadata importer, it will be ignored.

For more information on the Metadata Importer, see the **Metadata Importer** section of the **Asset Importer** documentation.

Each of the standard fields is described below:

- **File Name Field:** Used in the Metafile and ZIP Metafile receiver selections to identify the file name of the asset. This field is required in each metafile as it relates the asset files to the content in the metafile.
- **Match Metadata Field:** Used in the Metadata Product Linker configuration to specify the value for the product match criteria.
- **Reference Type Metadata Field:** Used in the Metadata Product Linker configuration to specify the STEP ID of the reference that will be used to link the asset and the product.

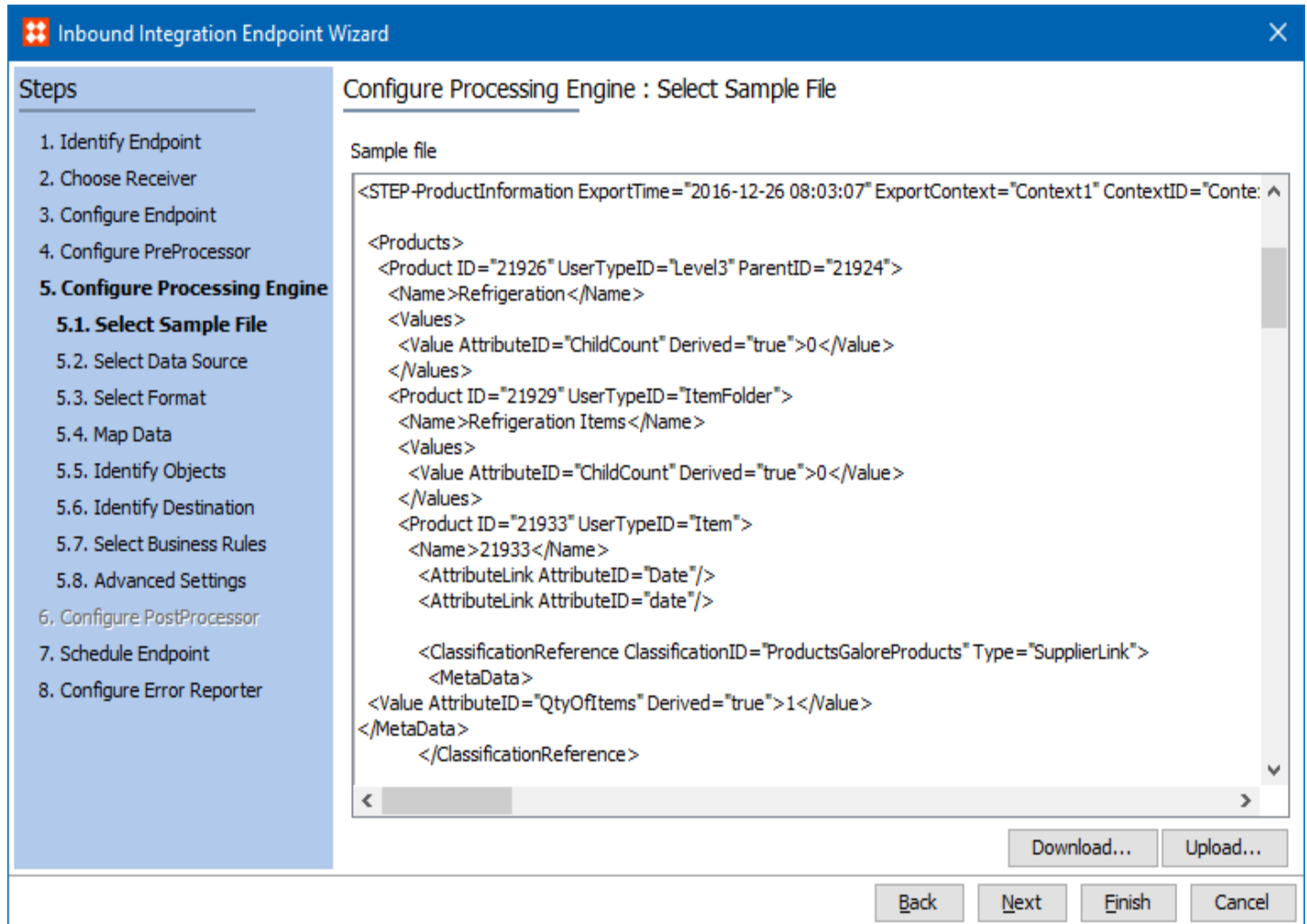
For more information on the Product Linker, see the **Product Linker** section of the **Asset Importer** documentation

Examples

1. Including metadata info to import (Metadata Importer):
 - **Filename;Product;Reference;Title;Photographer;PhotoCategory**
 - 1234.jpg;Item-12345;PrimaryImage;Glacier Bay Alaska;Peter Parker;Landscapes
2. Using a relative path (Metafile Receiver)
 - **Filename;Product;Reference**
 - Supplier1/1234.jpg;Item-12345;PrimaryImage
3. Multiple lines of the same asset to create multiple product references

- **Filename;Product;Reference**
- 1234.jpg;Item-12345;PrimaryImage
- 1234.jpg;Item-67890;PrimaryImage

IIEP - Configure STEP Importer Processing Engine



When using the STEP Importer processor, this step and its potential sub-steps, will typically be identical to the steps in the STEP Import Manager. Additionally, the same options are available for mapping and transforming data, and for referencing business rules to be run during import.

1. Create a sample file that contains the basic structure of the data, as shown in the image above.
2. In **5.1 Select Sample File**, click the **Upload** button to upload the sample (representative) file for the files or messages the IIEP will process. This file provides data for mapping (when required) and configures how to import data, and is stored in the database, but is not imported into STEP. This step is not available in the Import Manager since it allows you to only import a single file at a time.

If you want to modify a sample file that has already been uploaded, click the **Download** button and save the sample file to your computer. Make your changes and then click the **Upload** button to upload the modified sample file to the endpoint.

Note: When a preprocessor transforms the file to import, use a sample file that matches whatever format is produced by the preprocessor plugin. If the sample file is Excel, no preview is shown.

- When using the **Transformation by XSLT** preprocessor, use a sample file that matches your selected format (STEPXML or Generic XML) otherwise an error is returned that the format is not valid. For more information, see **IIEP - Configure Transformation by XSLT Preprocessor** documentation.
 - When using the **XML Normalizer** preprocessor, verify that the sample file includes both the new tag, and the copied tag data in the correct location. For more information, see **IIEP - Configure XML Normalizer PreProcessor** documentation.
 - When importing **ETIM6**, modify the import file to include only the <IXF >, <Header>, and <Groups> sections and save it as your sample.
4. In **5.2 Select Data Source** file, if the IIEP is used to update commercial data, also called Terms Lists, select the Termslist Targets. For more information, see topics in the **Commercial Data** documentation.
 5. In **5.3 Select Format** choose from the following options to indicate the type and basic parameters of the file you are importing.

When using the XML Normalizer preprocessor, verify that the template file includes both the new tag, and the copied tag data in the correct location. For more information, see **IIEP - Configure PreProcessor** documentation.

- Advanced STEPXML Format
 - BMEcat Format
 - BMEcat 2005 Format
 - CSV Format
 - ETIM Format
 - ETIM IXF Format
 - ETIM v2 Format
 - Excel Format
 - Excel Smartsheet Format
 - FixedWidth Format
 - Flatplanner Format
 - Generic XML Format
 - IDoc MATMAS 05 Format
 - Publication Excel Format
 - STEPXML Format
 - UNSPSC Format
5. In **5.4 – Map Data** match the columns of data in the import file to the appropriate STEP equivalents, and include any transformations. For information about all of the available mapping options, see the **Inbound Map Data Options** section.

6. In **5.5 – Identify Objects** allows you to verify if loaded objects are existing data in STEP or will be created as new objects. Additionally, you can specify to locate existing objects by attribute value rather than by the ID, key, or name. For more information, see **Import Manager - Identify Objects**.
7. In **5.6 – Identify Destination** specify settings such as the user who is responsible for approving the objects after import, where to place new objects in STEP, and the object type of new objects. Be cautious if setting default parent and object types. For more information, see **Import Manager - Identify Destination**.
8. In **5.7 – Select Business Rules** specify which business rule(s) to apply when data is imported. For more information, see **Import Manager - Select Business Rules**.
9. In **5.8 – Advanced Settings** are not required on a day-to-day basis for data imports and require a license. Generally these setting are used to clean up data within STEP. Common setup is to perform a test on the test server first, and start testing with small files since these options can cause data integrity issues. For more information, see **Import Manager - Advanced Settings**.
10. Click the **Next** button to display **IIEP - Configure PostProcessor** when available, or display **IIEP - Schedule Endpoint**.

IIEP - Configure Match and Merge Importer

The STEP Match and Merge Importer plays an integral role in the Matching & Merging deduplication solution, and is responsible for importing, processing, and standardizing data, and ultimately matching that data with existing golden records. When using the STEP Match and Merge Importer in an IIEP, it is not possible to make use of a postprocessor.

Configure Endpoint

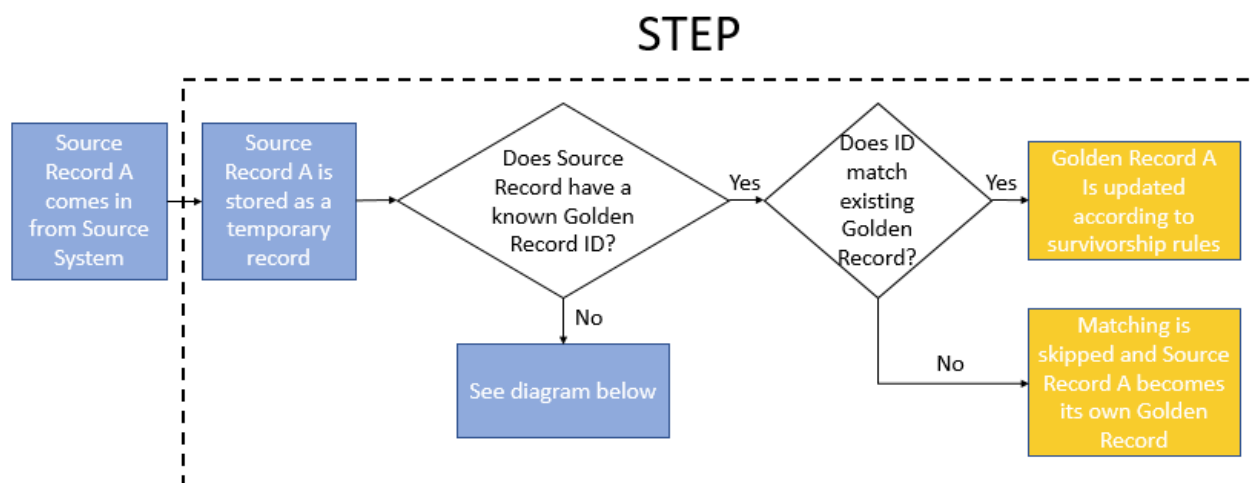
Processing	
Processing Engine	STEP Match and Merge Importer
Transactional settings	Strict

Match and Merge Importing Process and Logic

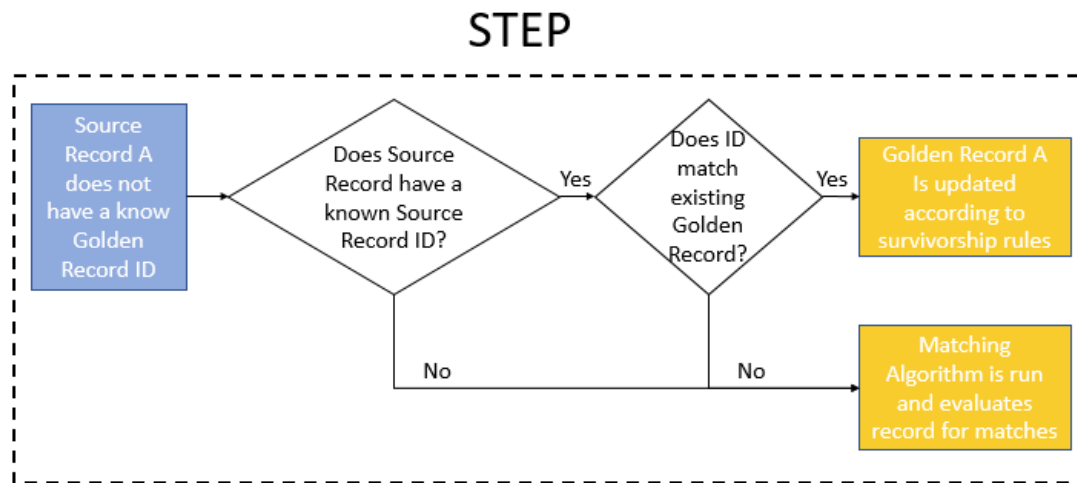
During an import, source records listed in the imported file are created, and treated as temporary STEP objects. Despite their temporary nature, business rules can act on them, and they can be matched via a matching algorithm. When being matched, survivorship rules determine which (if any) source values get promoted to existing golden records. After the import process is complete, these temporary objects are discarded.

Imported data is processed in the following order:

1. The IIEP checks the temporary source record to see if it has a known golden record ID. If it does have an ID, then the corresponding golden record in STEP is updated based on the survivorship rules configured in the matching algorithm. If no existing golden record shares the same ID, then the source record is ignored and is not imported to a golden record. A relevant error message is provided.



2. The IIEP checks the temporary source record to see if it has a known source record ID. If an existing active (or not deactivated) golden record is found that has the same source ID, survivorship rules are applied to promote source values to the golden record.



3. If neither a golden record ID or source record ID is found, the configured matching algorithm is invoked, and matches the temporary source records to existing golden records in STEP. If a source record matches one or more golden records above the auto threshold, then survivorship rules are applied to promote source values to the golden record that is matched with the highest score (if multiple golden records match above auto threshold, then golden records are not merged in this process). Those temporary source records that are matched below the auto threshold become their own golden records.

Configuration Wizard Steps

Configuration of the STEP Match and Merge Importer begins on the 'Configure Endpoint' step of the Inbound Integration Endpoint wizard.

1. On the **Configure Endpoint** step, click the 'Processing Engine' dropdown and select 'STEP Match and Merge Importer'.

Note: The **Transactional settings** and **Maximum number of waiting processes** parameters are pre-configured, and cannot be changed.

For details on how the other parameters in this step may be configured, see the **IIEP - Configure Endpoint** documentation.

The screenshot shows the 'Inbound Integration Endpoint Wizard' window, specifically the 'Configure Endpoint' step. On the left, a 'Steps' sidebar lists the following steps: 1. Identify Endpoint, 2. Choose Receiver, 3. **Configure Endpoint** (highlighted), 4. Configure PreProcessor, 5. Configure Processing Engine (with sub-step 5.1. Configuration), 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area is titled 'Configure Endpoint' and contains several configuration sections:

- Processing**: Includes 'Processing Engine' set to 'STEP Match and Merge Importer' and 'Transactional settings' set to 'Strict'.
- Context**: Includes 'Workspace' set to 'Main' and 'Context' set to 'GL'.
- Queue Settings**: Includes 'Queue for endpoint' (InboundQueue), 'Queue for endpoint processes' (In), 'Maximum number of waiting processes' (1), 'Maximum number of old processes' (100), 'Maximum age of old processes' (1w), and 'Number of messages per background process' (1).

At the bottom right, there are four buttons: 'Back' (highlighted with a dashed border), 'Next', 'Finish', and 'Cancel'.

2. On the **Configure PreProcessor** step, an existing import configuration can be used to transform Excel / CSV files into STEPXML, so that the processing engine can read those files.
 - To do this, select 'Transformation by Import Configuration' from the 'Configure PreProcessor' dropdown.

When making use of this option, the relevant import configuration may contain Excel / CSV mappings to STEPXML. If these files contain columns for Source System and Source Record ID, then the Reference Meta-Data option must be used to map the Source System to the relevant References Type, and the Source Record ID must be mapped to the relevant meta-data attribute on the Reference Type in question.
 - For the 'Import Configuration' parameter, click the ellipsis button (...) and browse or search for an import configuration to use for transforming incoming Excel / CSV files.
 - For the 'Allow open ZIP file' parameter, select 'Yes' from the dropdown to allow the importer to open ZIP files and convert the files within.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
- 4. Configure PreProcessor**
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure PreProcessor

Configure PreProcessor: Transformation by Import Configuration

Import Configuration: ...

Allow open ZIP file: Yes

This pre-processor enables the user to point out an already existing import configuration that can be used to transform incoming Excel and CSV files to the STEPXML required by the STEP Match and Merge Importer process engine used in Inbound Integration Endpoint. The conversion will be done by selecting an existing import configuration stored in an Asset. The pre-processor also allows traversing ZIP files if requested.

Back Next Finish Cancel

If STEP IDs are not typically available in your imports, it is recommended to use an import configuration that can create entity-to-entity references via Source Record ID.

For more information on Excel / CSV imports, see the **Entity Reference via Source Record ID - Map Inbound** section of the **Data Exchange** documentation.

For more information on XML imports, see the **Source Record ID in STEPXML** section below.

3. On the **Configure Processing Engine** step, business actions can be configured to process the temporary STEP source objects in order to standardize the data, which allows for more accurate matching.

For both of the parameters, click the link again to add items. Execution is performed in the order displayed. Click the ellipsis button (...) to change an item, click the arrows to change the order, and click the X button to remove an item.

- If a business action is required, click the **Add action** link and browse or search for the relevant business action.
- In the 'Matching Algorithms' parameter, click the **Add Matching Algorithm** link and browse or search for a matching algorithm. The matching algorithms configured are used to match the inbound source records against existing golden records and invoke survivorship rules.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
- 5. Configure Processing Engine**
 - 5.1. Configuration**
 6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Processing Engine : Configuration

Standardization Business Actions: Standardize Address Action [Add action]

Matching Algorithms: Individual Match Algorithm [Add Matching Algorithm]

Back Next Finish Cancel

Source Record ID in STEPXML

The Source Record ID of a given record is stored in the Source Record ID Attribute of that record. This attribute is a metadata attribute placed on the record's Source Relation Reference, which is a reference between the source record and the Source System in question.

See below for a STEPXML example of the relevant tags:

```
<Entity>
  <Name>Carl Lucas</Name>
  <Values>
    <Value AttributeID="FirstName">Carl</Value>
    <Value AttributeID="MiddleName"></Value>
    <Value AttributeID="LastName">Lucas</Value>
    <Value AttributeID="Email">l.cage@email.com</Value>
    <Value AttributeID="PhoneNo"></Value>
  </Values>
  <EntityCrossReference EntityID="SourceSystem_Test" Type="MergeSourceRelation">
    <MetaData>
      <Value AttributeID="SourceRecordID">SST_00001</Value>
    </MetaData>
  </EntityCrossReference>
  <EntityCrossReference Type="CustomerToOrganization">
    <Entity>
      <EntityCrossReference EntityID="SAP_Test" Type="MergeSourceRelation">
```

```
        <MetaData>
            <Value AttributeID="SourceRecordID">SSB_00002</Value>
        </MetaData>
    </EntityCrossReference>
</Entity>
</EntityCrossReference>
</Entity>
```

IIEP - Configure PostProcessor

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Identify Endpoint

Endpoint ID: Inbound Data

Endpoint Name: Inbound Data

Description:

User: user (USER)

By default, STEP does not include any pre-configured post-processor plugins and the wizard step is disabled. When created, a customized post-processor has access to import events (information about what has changed) and, based on the events, can trigger any required system change.

Typically, post-processor logic is implemented via business rules referenced from the processing engine configuration. For assistance in creating a post-processor, contact Stibo Systems Support.

Click the **Next** button to display **IIEP - Schedule Endpoint**.

IIEP - Schedule Endpoint

The screenshot shows the 'Inbound Integration Endpoint Wizard' window, specifically the 'Schedule Endpoint' step. The window has a blue title bar with the Stiibo logo and a close button. On the left, a 'Steps' pane lists 8 steps: 1. Identify Endpoint, 2. Choose Receiver, 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine (with sub-steps 5.1 to 5.8), 6. Configure PostProcessor, 7. Schedule Endpoint (highlighted), and 8. Configure Error Reporter. The main area is titled 'Schedule Endpoint' and contains a 'Start' section with five radio buttons: 'Never' (selected), 'Every', 'Weekly', 'Monthly', and 'Later'. Below these is the text 'Start Not scheduled'. At the bottom, there are four buttons: 'Back', 'Next' (highlighted with a dashed border), 'Finish', and 'Cancel'.

Use Schedule Endpoint to specify how often the endpoint should search for data to be processed considering the following:

- IIEPs that do not use the REST Receiver method should typically be scheduled to poll the data source for new files / messages with regular intervals.
 - IIEPs that do use the REST Receiver method can be invoked via a REST POST request. With this functionality, for example, a system delivering data to STEP could potentially invoke the IIEP after having uploaded a file to a Hotfolder. For more information, see the **REST Receiver** documentation.
1. Choose a radio button to select the necessary schedule:
 - **Never** - invoke the endpoint 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 endpoint repeatedly, every selected number of minutes. One (1) minute is the shortest interval allowed and is closest to real time. Enter the number of minutes in the text box. The selection is summarized at the bottom of the dialog.

☐ Never
 minutes

☒ Every

☐ Weekly

☐ Later

Start every minute

- Weekly** - automatically run the endpoint repeatedly, based on the selected time, start and end dates, and days of the week. Use this option if a daily schedule is needed. The 'Start at' parameter determines the time of day that the endpoint will run. The 'Start on' parameter determines the date the endpoint will first run, while the 'End on' parameter determines the date of the endpoint's final run. The 'Every' checkboxes determine the days of the week when the endpoint will run. The selections are summarized at the bottom of the dialog.

☐ Never
 ☐ Every
 ☒ Weekly
 ☐ Later

Start at (hh:mm):

Start on (yyyy-mm-dd):

End on (yyyy-mm-dd):

Every:
 ☒ Mon
 ☒ Sat
 ☐ Tue
 ☐ Sun
 ☒ Wed
 ☐ Thu
 ☐ Fri

Start every Mon, Wed, Sat 21:10:00 EDT, starting Thu Jul 06 2017

- Monthly** - automatically run the endpoint 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 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' dropdown parameter selections for the week of the month and the day of the week determine when the endpoint will run. The selections are summarized at the bottom of the dialog.

☐ Never
 ☐ Every
 ☐ Weekly
 ☒ Monthly
 ☐ Later

Start at (hh:mm):

Start on (yyyy-mm-dd):

End on (yyyy-mm-dd):

Every:

Start every third Thu 22:00:00 EDT, starting Thu Jul 06 2017

- **Later** - automatically run the endpoint only once, at the time and date specified. The selections are summarized at the bottom of the dialog.

☐ Never	Start at (hh:mm):	20:30
☐ Every	Start on (yyyy-mm-dd):	2017-07-06
☐ Weekly		
☒ Later		

Start at Thu Jul 06 20:30:00 EDT 2017

2. Click the **Next** button to display **IIEP - Configure Error Reporter**.

IIEP - Configure Error Reporter

The screenshot shows the 'Inbound Integration Endpoint Wizard' window. The title bar is blue with a red cross icon and the text 'Inbound Integration Endpoint Wizard'. The window is divided into two main sections. On the left is a 'Steps' sidebar with a list of steps: 1. Identify Endpoint, 2. Choose Receiver, 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine (with sub-steps 5.1 to 5.8), 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter (which is highlighted in blue). The main area on the right is titled 'Configure Error Reporter'. It contains a dropdown menu labeled 'Configure Error Reporter' with 'Send Error Report' selected. Below this is a text input field labeled 'Send report to address' containing 'user@acme.com'. At the bottom of the main area is a text box with the following text: 'This plug-in will send an error report to the e-mail address given in the input field. If no address has been supplied, the e-mail will be sent to the address configured for the endpoint's responsible User.' At the very bottom of the window are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

1. On **Configure Error Reporter**, select an option that is activated if an endpoint-related background process fails. The following options are standard with STEP:
 - **Send Error Report** emails information about the failed endpoint, the background process, the failed file, the failing process step, the cause of the error, and a copy of the file that triggered the error. Leaving the 'Send report to address' parameter blank means the error report is sent to the email address of the user who created the integration endpoint. If no email is defined for this user, or if no mail server is defined in the configuration, the error report is written to the failed Background Process Execution Log.
 - **Not Defined**, the error reporter is disabled, no notification by email is sent.
2. In **Send report to address**, enter the e-mail address of recipient of the error report. Use a semicolon (;) to separate multiple email addresses.
3. Click **Finish**. The endpoint is created and appears in the setup group.
4. The endpoint must be enabled before it can start processing data. For more information, see the **Running an Inbound Integration Endpoint** documentation.

Running an Inbound Integration Endpoint

After configuring an inbound endpoint, it must be enabled before it can run on schedule or be invoked manually.

The following icons are used to indicate the status of the inbound integration endpoint. The endpoint status is displayed in the System Setup hierarchy and on the endpoint editor in the Description flipper.

Icon	Endpoint Status	Description
	Enabled	The integration endpoint is active, connected, and running.
	Disabled	The integration endpoint has stopped and no data is being imported. Newly created endpoints, and endpoints that have been manually disabled, will have this status.
	Failed	An error occurred during processing that caused the integration endpoint background process to fail. No data will be imported until the endpoint has been manually resumed / reactivated. For more information, see Handling Failed IIEP Background Processes documentation.

Enable and Invoke an Inbound Integration Endpoint

Once configured, use the following steps to run the endpoint.

1. On System Setup expand the Inbound Integrations Endpoints node to display all existing inbound endpoints.
2. Select a  disabled inbound endpoint or a  failed endpoint, right-click, and select **Enable Integration Endpoint** from the menu.

The screenshot shows the 'System Setup' tree on the left with 'Inbound Data' selected. A context menu is open over 'Inbound Data' with 'Enable Integration Endpoint...' highlighted. The main panel shows the 'Inbound Data rev.1.1 - Inbound Integration Endpoint' configuration page. The 'Description' tab is active, displaying a table of parameters.

Name	Value
ID	Inbound Data
Name	Inbound Data
Object Type	Inbound Integration Endpoint Type
Revision	1.1 Last edited by USERJ on Tue Jun 28 10:43:57 EDT 2016
Description	
Enabled	No
Endpoint Status	Stopped
Last run	-
Next run	-

Below the table, there are sections for 'Configuration', 'Hotfolder Receiver Configuration', and 'Integration Endpoint Log'.

The tree navigator icon now shows enabled. On the Inbound Integration Endpoint tab, the Description flipper includes the 'Enabled' parameter, which now displays 'Yes'.

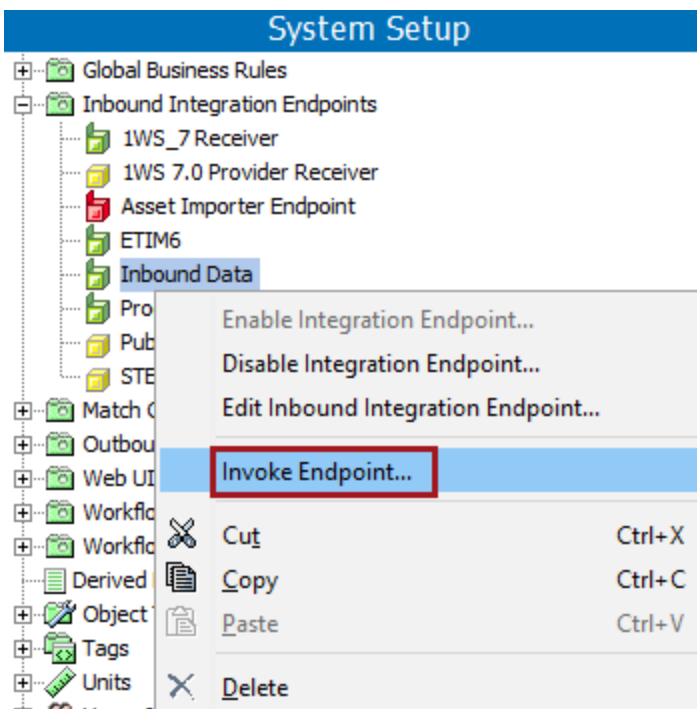
The screenshot shows the 'System Setup' tree on the left with 'Inbound Data' selected. The main panel shows the 'Inbound Data rev.0.1 - Inbound Integration Endpoint' configuration page. The 'Description' tab is active, displaying a table of parameters.

Name	Value
ID	Inbound Data
Name	Inbound Data
Object Type	Inbound Integration Endpoint Type
Revision	0.1 Last edited by Stiibo Systems on Wed Jan 27 11:43:03 EST 2...
Description	
Enabled	Yes
Endpoint Status	Running
Last run	-
Next run	-

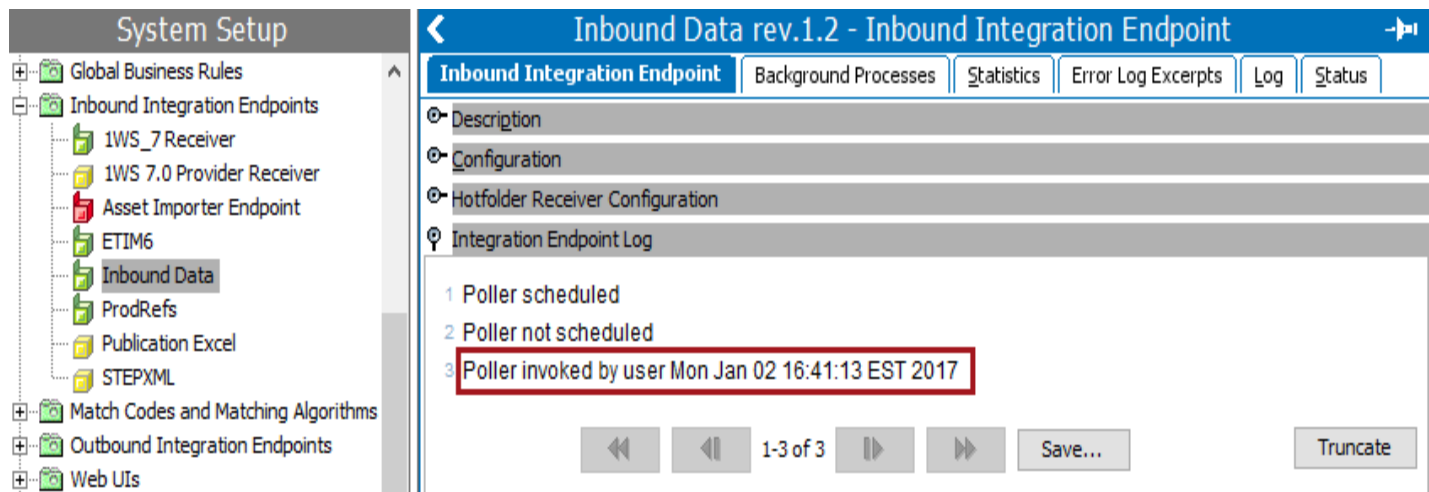
- Open the Configuration flipper to view the Schedule parameter.
 - If scheduled, the Schedule parameter shows when the IIEP will be invoked automatically.

Inbound Integration Endpoint		Background Processes	Statistics	Error Log Excerpts	Log	Status
Description						
Configuration						
Pre-Processor	No pre-processing					
Process Engine	GDSN Receiver Inbound message processor					
Post-Processor	No post-processing					
Error reporter	Not Defined					
Schedule	Start every Mon, Tue, Wed, Thu, Fri, Sat, Sun 21:55:00 EST, starting Wed Jan 04 2...					...
Queue for endpoint	InboundQueue					
Queue for endpoint processes	In					
Transactional settings	Strict					

- If the endpoint is not scheduled, or if the IIEP needs to be invoked on demand, right-click the enabled IIEP and click **Invoke Endpoint**.

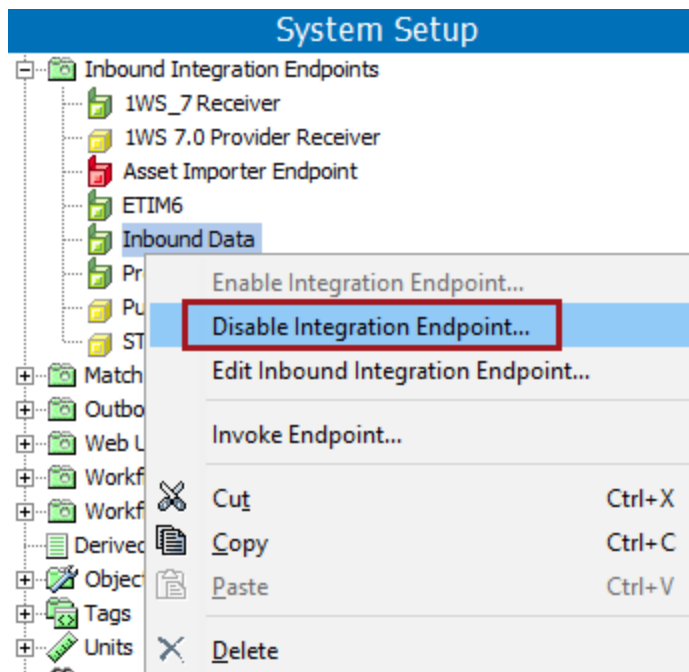


- When invoked on schedule and manually, files or messages available in the location identified by the IIEP receiver configuration are processed. The Integration Endpoint Log flipper displays the time the endpoint was invoked. For scheduled IIEPs, the log includes a link to the background process that was started. For more information, see **Maintaining an Inbound Integration Endpoint**.



Disable an Inbound Integration Endpoint

When an IIEP displaying the enabled icon (🟢) should be stopped from processing data, you can disable it. The configuration remains unchanged and no further data is processed. After selecting the IIEP, right-click and select **Disable Integration Endpoint** to set the IIEP to disabled and display the disabled icon (🔴).

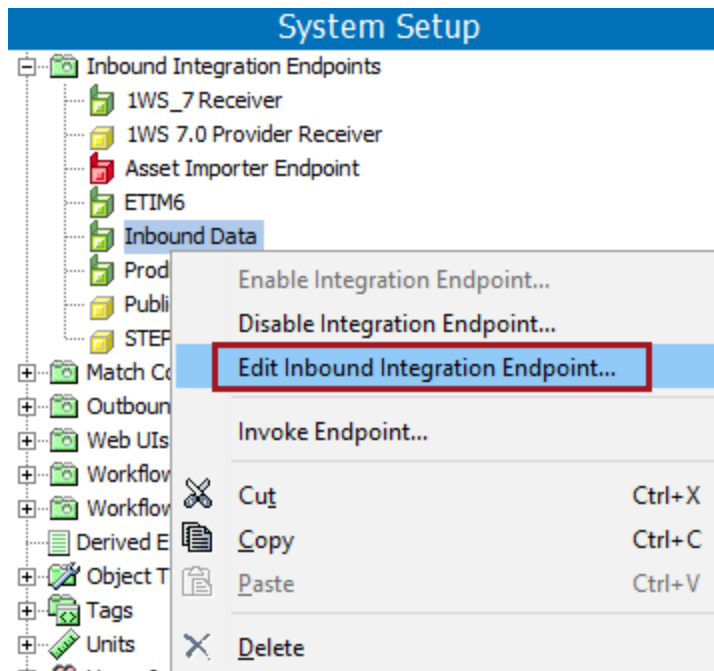


Once disabled, the inbound integration endpoint can be restarted using the Enable Integration Endpoint step defined above.

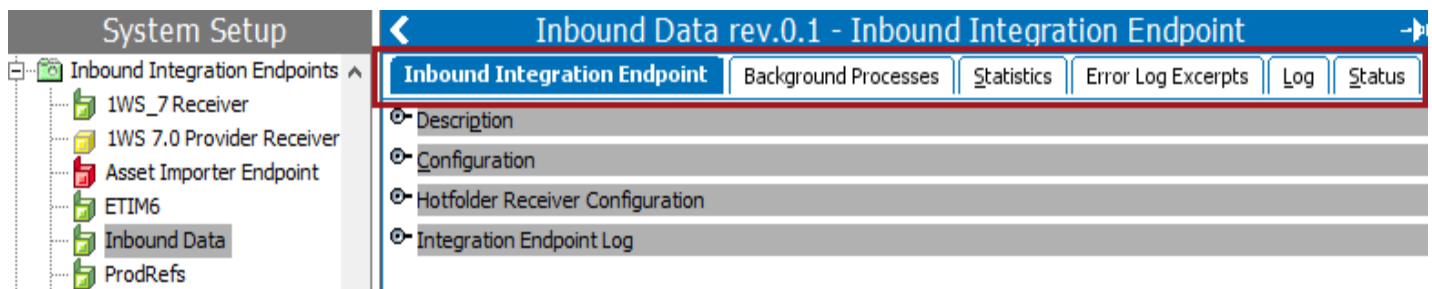
Additional information about restarting a failed endpoint, indicated by the failed icon (🔴) can be found in **Handling Failed IIEP Background Processes** documentation.

Maintaining an Inbound Integration Endpoint

Inbound integration endpoint settings can be viewed in the Inbound Integration Endpoint editor. To edit the settings, reopen the wizard using the hyperlinks defined below, or with the **Edit Inbound Endpoint** option on the IIEP's right-click menu.



Each tab on the editor is defined below and includes parameters to maintain the IIEP.



Inbound Integration Endpoint Tab

This tab holds basic information within the following flippers:

Description Flipper

The Description flipper includes basic information to identify the IIEP. Only the name, description, and enabled parameters can be edited. This data is originally set up in the IIEP wizard, see **IIEP - Identify Endpoint** documentation.

For information on the Endpoint Status, see **Running an Inbound Integration Endpoint** documentation.

Inbound Integration Endpoint		Background Processes	Statistics	Error Log Excerpts	Log	Status
Description						
Name	>	>	Value			>
ID			Inbound Data			
Name			Inbound Data			
Object Type			Inbound Integration Endpoint Type			
Revision			1.2 Last edited by USERJ on Wed Jan 04 08:29:46 EST 2017			
Description						
Enabled			Yes			
Endpoint Status			Running			
Last run			2017-01-04 08:29:03			
Next run			2017-01-04 08:30:03			
Configuration						
Hotfolder Receiver Configuration						
Integration Endpoint Log						

Configuration Flipper

The Configuration flipper displays many of the settings determined through the wizard.

- The **Edit Configuration** link opens the wizard and allows you to modify the following parameters:
- For information on the Pre-Processor, see **IIEP - Configure PreProcessor**.
- For information on the Process Engine, see **IIEP - Configure Processing Engine**.
- For information on the Post-Processor, see **IIEP - Configure PostProcessor**.
- For information on the Error reporter, see **IIEP - Configure Error Reporter**.
- Changes to the schedule can be made easily using the Schedule parameter ellipsis button (...). This opens the wizard on the Schedule Endpoint step, see **IIEP - Schedule Endpoint**.
- For information on the Queue for endpoint, Queue for endpoint process, Transactional settings, Maximum number of waiting processes, Maximum number of old processes, Maximum age of old processes, Number of messages per background processes, Contexts, and Workspace, see **IIEP - Configure Endpoint**.

Inbound Integration Endpoint
Background Processes
Statistics
Error Log Excerpts
Log
Status

Description
Configuration

Pre-Processor	No pre-processing
Process Engine	STEP Importer
Post-Processor	No post-processing
Error reporter	Not Defined
Schedule	Not scheduled
Queue for endpoint	InboundQueue
Queue for endpoint processes	In
Transactional settings	Chain
Maximum number of waiting processes	1000
Maximum number of old processes	1000
Maximum age of old processes	1 year
Number of messages per background process	1
Contexts	English US
Workspace	Main

> Edit Configuration

Hotfolder Receiver Configuration
Integration Endpoint Log

Receiver Configuration Flipper

The Receiver Configuration flipper displays details about the receiver method selected for the IIEP. For more information about receiver methods, see **IIEP Receiver Methods**.

- The **Edit Receiver Plugin** link opens the wizard on the Choose Receiver step and allows you to modify the settings. For more information, see **IIEP - Choose Receiver**.

Inbound Integration Endpoint
Background Processes
Statistics
Error Log Excerpts
Log
Status

Description
Configuration

Hotfolder Receiver Configuration

ID	> Name	>
> Hotfolder	in	
> Keep file after load	Yes	
> Ignore sub folders	No	
> In folder	in	

> Edit Receiver Plugin

Integration Endpoint Log

Integration Endpoint Log Flipper

The Integration Endpoint Log flipper contains information about the endpoint background process. An entry is generated each time the endpoint is invoked, when background processes generated by the endpoint are started, and when processing errors occur.

- The background process (**BGP**) link opens the individual Importer Background Processes.

Background Processes Tab

This tab holds information about any related processes. Hover over an ID to display the background process (**BGP**) link, click it to open the individual Importer Background Processes.

- Queued Processes** are background processes that are started in the state 'waiting'. The maximum number of background processes that can be started in the state waiting is specified in the integration endpoint configuration.
- Active Processes** are background processes that are currently being processed. If data is processed as strict or chained, the endpoint only has one active background process at a time. Otherwise, it is possible to apply more background process queues to work concurrently. In this case the endpoint can have more active background processes at a time.
- Failed Processes** are background processes that are no longer running. For more information, see **Handling Failed IIEP Background Processes**.
- Completed with errors** have run, but an error has been detected.
- Ended Processes** have completed successfully. The number of ended processes is determined by the configuration.

For more information, see the **Monitoring an IIEP via Background Process** section.

Inbound Integration Endpoint	Background Processes	Statistics	Error Log Excerpts	Log	Status		
Queued Processes							
Id	Description	Status	Progress	Start Date	Started By	Errors	Created
Active Processes							
Id	Description	Status	Progress	Start Date	Started By	Errors	Created
Completed with Errors							
Id	Description	Status	Progress	Start Date	Started By	Errors	Created
Ended Processes (1)							
Id	Description	Status	Progress	Start Date	Started By	Errors	Created
BGP_203450	Import started for endpoint...	succeeded	100%	Wed Jan 04 08: ...	USER	0	Wed Jan 04 08: ...

Statistics Tab

This tab provides statistics about the number of requests handled, run times, and wait times. Key measures are Endpoint up-time, Waiting in data source, Number of failed requests (Background Processes with errors), and Mean run time (average processing time per import). This data cannot be exported.

Note: Receiver plugins poll the data source to check for waiting files or messages more frequently than the IIEPs are invoked.

Inbound Integration Endpoint	Background Processes	Statistics	Error Log Excerpts	Log	Status
Statistics					
Last run		2017-01-04 08:29:03			
Next run		2017-01-04 08:30:03			
Endpoint uptime		1 day 19 hours 5 m 32 s			
Number of handled requests		1			
Waiting in data source		0			
Requests waiting to be processed		0			
Number of failed requests		0			
Number of running requests		0			
Minimum wait time		5 s			
Mean wait time		5 s			
Maximum wait time		5 s			
Minimum run time		-			
Mean run time		-			
Maximum run time		-			

Error Log Excerpts Tab

This tab shows data from the main Java log file related to failed background processes with Log Level > 'Info'. Click a hyperlink to a failed background process and correct the cause of the failed background process. For more information, see **Handling Failed IIEP Background Processes**.

Inbound Integration Endpoint	Background Processes	Statistics	Error Log Excerpts	Log	Status
Integration Endpoint Log					
Background Pro...	Log Item No	Text			
> BGP_206613	130	Row 2: Parent classification 'AddressRoot' for classification '<a href="step			
> BGP_206613	150	Row 3: Parent classification '18203' for classification '<a href="step://das			

Log Tab

This tab provides information about changes to the IIEP configuration.

Inbound Integration Endpoint	Background Processes	Statistics	Error Log Excerpts	Log	Status
Showing page 1 of 1					
2015-07-16 14:49:29 'USER': Created 2015-07-16 14:49:29 'USER': Name modified from 'null' 2015-07-16 14:49:29 'USER': Modified 2016-02-12 14:00:19 'USER': Modified 2016-06-28 10:43:57 'USERJ': Modified 2017-01-02 16:36:57 'USERJ': Modified 2017-01-04 08:02:39 'USERJ': Modified 2017-01-04 08:10:09 'USERJ': Modified 2017-01-04 08:25:27 'USERJ': Modified 2017-01-04 08:27:15 'USERJ': Modified 2017-01-04 08:28:03 'USERJ': Modified 2017-01-04 08:29:46 'USERJ': Modified 2017-01-04 10:51:35 'USERJ': Modified					

Status Tab

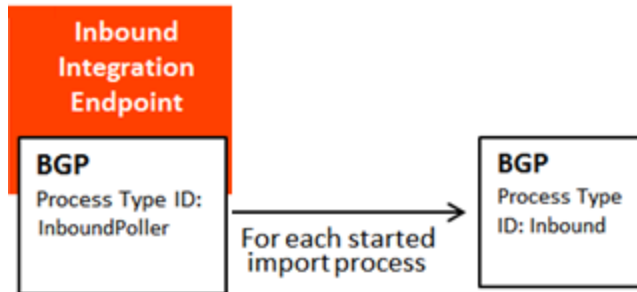
This tab provides information about revisions, hidden values and diagnostics.

Inbound Integration Endpoint	Background Processes	Statistics	Error Log Excerpts	Log	Status
Revisions					
Revision	Created	Edited	Major		
> 1.2	Mon Jan 02 16:33:01 EST ...	Wed Jan 04 10:51:35 EST...			
> 1.1	Tue Jun 28 10:43:57 EDT ...	Tue Jun 28 10:43:57 EDT ...			
> 1.0	Fri Feb 12 14:00:19 EST 2...	Fri Feb 12 14:00:19 EST 2...	X		
> 0.1	Wed Jan 27 11:43:03 EST...	Wed Jan 27 11:43:03 EST...			
Hidden values					
Diagnostics					

Monitoring an IIEP via Background Process

You can monitor an IIEP from within STEP by way of a background process.

An active inbound integration endpoint uses an associated background process to handle the scheduled invocation of the endpoint. The execution report for this background process is integrated in the workbench endpoint editor and includes invocation information as well as messages logged from the plugins. When using the standard STEP Importer processing engine, the actual import is handled in separate background processes started by the endpoint, as illustrated below.



By default, the started integration endpoint background processes are displayed within the IIEP editor on the Background Processes tab, and are not visible on the BG Processes tab.

Inbound Integration Endpoint	Background Processes	Statistics	Error Log Excerpts	Log	Status		
Queued Processes							
Id	Description	Status	Progress	Start Date	Started By	Errors	Created
Active Processes							
Id	Description	Status	Progress	Start Date	Started By	Errors	Created
Completed with Errors							
Id	Description	Status	Progress	Start Date	Started By	Errors	Created
Ended Processes (1)							
Id	Description	Status	Progress	Start Date	Started By	Errors	Created
BGP_203450	Import started for endpoin...	succeeded	100%	Wed Jan 04 08: ...	USER	0	Wed Jan 04 08:

For more information, see **Maintaining an Inbound Integration Endpoint**.

When configuring an inbound integration endpoint that uses the STEP Importer processing engine, you will specify a background process queue to use for each of the two types of background processes:

- Queue for endpoint
- Queue for endpoint processes

Typically, all background processes of the same type (same process type ID) will use the same queue. However, integration endpoint processes can be tied to different integration endpoints and use different queues.

Selecting the Background Process Queue

The main background process uses the default **InboundQueue** background process queue for IIEPs.

However, when you create an integration endpoint, you can specify that you want to use a different IIEP background process queue for the main IIEP background process queue. For more information about this setting, see the **IIEP - Configure Endpoint** documentation.

Background Process Queue Size Property

The queue size setting can have great impact on processing. The background process queue size property determines the number of background processes can be executed concurrently on the queue. If the size is 1, only one background process can run at a time. If the size is 2, two processes can run concurrently, and so forth.

Important: Although the integration endpoint configuration wizards suggest the number of queues to use, beware that if you create multiple endpoints and do not change the default queue suggestions, your setup could include endpoints that block each other since the queues have a default size of 1.

This is not a big concern for the endpoint processes that, most of the time, are idle and only take up a slot on the queue briefly when the endpoint is invoked (such as when polling a hotfolder for new files). However, it can be an issue for the generated background processes that perform the actual export processing. The decision as to whether different endpoints should use the same or different processing queues and the size of the Queues should therefore be an informed one.

Overwriting default values for a queue causes a new queue to be created with the queue size of 1. Use the following `sharedconfig.properties` entry to modify the default queue size value:

```
BackgroundProcess.Queue.[name of queue].Size = [number of allowed concurrent processes]
```


Monitoring an IIEP via External System

Integration endpoints can be monitored from external systems using Monitoring Sensors or the REST API.

Monitoring Sensors

For each integration endpoint created in STEP, a Monitoring Sensor is automatically created. Monitoring Sensors allow external systems to query the status of individual endpoints via HTTP, without authentication. In the following sample URLs, substitute your own system URL and endpoint ID to access your own endpoints.

A Monitoring Sensor can return the following information:

1. **Simple traffic light response** returns OK, WARNING, CRITICAL, or UNKNOWN for inbound endpoints via:

```
http://[System URL]/admin/monitoring/InboundIntegrationEndpointStatus-[Endpoint ID]/status
```

2. **Nagios friendly response** returns detailed statistics in Nagios friendly format for inbound endpoints via:

```
http://[System URL]/admin/monitoring/InboundIntegrationEndpointStatus-[Endpoint ID]/nagios
```

3. **Full detailed XML response** returns XML with detailed statistics for inbound endpoints via:

```
http://[System URL]/admin/monitoring/InboundIntegrationEndpointStatus-[Endpoint ID]/xml
```

REST API Monitoring

An alternative to Monitoring Sensors is to use the REST API, which requires authentication.

1. **Overview of all** configured integration endpoints via:

```
[Host]/restapi/integrationendpoints?context=[Context]&workspace=[Main]
```

2. **Detailed statistics** for specific endpoint via:

```
[Host]/restapi/integrationendpoints/[Endpoint ID]?context=[Context]&workspace=[Main]
```

3. **Execution report** for specific endpoint via:

```
[Host]/restapi/integrationendpoints/[Endpoint ID]/log?context=[Context]&workspace=[Main]
```

4. **Java log entries related to errors in Background Processes** generated by specific endpoint via:

```
[Host]/restapi/integrationendpoints/[Endpoint ID]/errorexcerpts?context=[Context]&workspace=[Main]
```

5. **Overview of Background Processes** generated by specific endpoint via:

```
[Host]/restapi/integrationendpoints/[Endpoint ID]/backgroundprocesses?context=[Context]&workspace=[Main]
```

Import Error Messages

The following table shows most of the different types of error message that you are likely to encounter, with a brief explanation of the most common reason for the error. The most common errors allow users to troubleshoot and then perform the import. There are instances where issues are not related to the import file and are related to the servers, network latency, etc. For these scenarios, contact the Stibo Support team for assistance.

The error messages marked with an asterisk (*) are errors that you will only encounter when you load an XML file. That is because in an XML import, the wizard does not prevent you from entering an invalid value. However, when importing CSV or XLS files, since you often select values from a list, you cannot select an invalid value.

Error Message	Description
Illegal value x for attribute y in product z	The value in the load file does not meet the validity or other constraint that is set up for the attribute.
Not in legal values list	You tried to load an attribute value into an attribute that uses an LOV, and that value does not exist in the LOV, and it is set up not to accept additional entries. This error is not displayed when an LOV is configured to 'Allow Users to Add Values = Yes'.
Attribute not legal for object type	The product's object type is not valid for the attribute. That is, the attribute is not allowed to hold a value for products with the specified object type.
*Illegal unit x for attribute y in product z	In the XML file, you have tried to assign a non-existing unit to an attribute.
Unknown Classification x in product y.	In the XML file, you have probably requested a product-to-classification link using a specified reference type, but the classification you specified is not found.
Unknown Parent product x for product y	You tried to assign a product to a parent ID, but that parent ID cannot be found in the system.
*Unknown product reference target	You attempted to link one product to another via a reference type, but the target product does not exist.
*Unknown attribute	You specified an attribute in the XML file, but that attribute does not exist.

Error Message	Description
Unknown unit	The unit encountered in the load file does not exist in STEP.
Rejected new product x	You set the option to "Reject New Products" in the wizard, and a new product was encountered in the load file.
Illegal unit for attribute	The unit exists but is not valid for the attribute you are trying to load the value into.
*Illegal Object Type for product position	You tried to create a new product via an XML load and you specified an object type that is not a valid choice, given the parent product that you specified.
*Unknown Object Type	You specified an object type (e.g. for a product) in the XML file, but that object type does not exist.
*Unknown reference Type	You specified a reference type (e.g. for a product-to-product reference) in the XML file, but that reference type does not exist.
Insufficient privileges to ...	You do not have permission to carry out one of the following actions (specified in the error message): • set value • set name • create product • move product • modify Object Type • classify product • create product reference
Product x contains more than one reference to product y of Type z, reference data will not be imported.	You cannot make a product-to-product reference from the same source product to the same target product more than once.
Product x contains more than one reference to asset y of Type z, reference data will not be	You cannot make a product-to-asset reference from the same source product to the same target asset more than

Error Message	Description
imported.	once.
*Classification x contains more than one reference to Classification y of Type z, reference data will not be imported.	You cannot make a classification-to-classification reference from the same source classification to the same target classification more than once.
*Reference Type x is not valid for importing a product cross reference from a to b.	In your XML file, you have asked for a product reference that refers to a non-existent reference type.
*Reference Type x is not valid for importing an asset cross reference from a to b.	In your XML file, you have asked for an asset reference that refers to a non-existent reference type.
*Reference Type x is not valid for importing a Classification cross reference from a to b.	In your XML file, you have asked for a classification reference that refers to a non-existent reference type.
*Product x already contains a reference to Classification y but it is invisible in the specified context. The reference and any data on it will not be imported.	You have probably made a product-to-classification link dimension-dependent, and the context you are importing into cannot "see" that link, but it is still there.
Conversion Error	This run time exception error is returned when the XML file is loaded is poorly formed. Download the error file, make the indicated correction, and import the file again.
Conversion Error: Not a Valid STEPXML File (Encoding)	When any encoding type other than UTF-8 is entered in a STEPXML import file, this error is displayed in wizard on the 'Select Format'.
Unknown validator	Importing STEPXML that includes an attribute with a validator type that does not exist in STEP. Validation types with multiple words must replace the space with an underscore. For example, 'Numeric Text' must be written as 'Numeric_Text'.
Context 'XYZ' not found	Importing STEPXML where the Context ID is either misspelled or does not exist.
No ContextID specified in STEPXML to import	Importing STEPXML without a Context ID.

Error Message	Description
No WorkspaceID specified in STEPXML to import	Importing STEPXML without a Workspace ID. Workspace ID should always be an editable workspace, often 'Main'. Since the Approved workspace cannot be edited, it cannot receive imports.

Import Error Message Examples

The most common errors encountered during data import refer to an invalid attribute value. Sometimes there is a mismatch with the attribute's validation type, other times there is a mismatch with one of the other setups. For example, the attribute's constraints, such as masks, minimum and maximum values, maximum length, LOVs, the object's own object type, and so on. Therefore, when you encounter an error, look at the attribute setup and review the constraints.

Even if you load attribute values for an object, not all values will be automatically available in STEP. Attributes must be made valid for an object before the values can be accessed via references. STEP will always load attribute values if the object's type is valid for the attribute, and the attribute values meet the validity criteria. But, the attribute itself must be a valid attribute for that object, that is, linked somewhere in the object hierarchy or classification hierarchy where the object resides.

Use the following examples to assist in troubleshooting problems with data imports.

Illegal Unit for Attribute

The tab-delimited input file below had two columns mapped: the first column was mapped both to the Product ID and Product Name, and the second column was mapped to the attribute 'Description'.

```
T100-3526      RED 1/2"  STRAIGHT WIDGET SSU05PL
T100-2625      BLUE 1/4"  ANGLED GADGET BZT10GL
T100-2827      ORANGE 3/4" ELL WHATSIT PLU04HF
```

When the file was imported, the execution report showed three illegal unit errors:

- Illegal unit "/2" STRAIGHT WIDGET SSU05PL" for attribute Description in product T100-3526 (UnknownUnit).
Source: Line=2, Column=Column2.
- Illegal unit "/4" ANGLED GADGET BZT10GL" for attribute Description in product T100-2625 (Unknown Unit).
Source: Line=3, Column=Column2.
- Illegal unit "/4" ELL WHATSIT PLU04HF" for attribute Description in product T100-2827 (Unknown Unit).
Source: Line=4, Column=Column2.

Upon inspection of the attribute with the name of 'Description', it was found that the assigned data validity check was 'Numeric_Text', and that it had a unit assigned to it: mm. Further, it was not selected as the default unit.

STEP could not handle the input text, since the attribute's validity was defined as being NumericText and had only one legal unit, and that unit could not be matched with anything in the attribute value provided by the import. In this case, either the attribute's setup is incorrect, or a different product attribute should be used that has an appropriate data validity type assigned.

Unknown Parent Product

The tab-delimited input file includes three columns. Column 1 was mapped to both Product ID and Product Name (new products), and Column 3 was mapped to the Parent ID.:

```
T100-3526      RED 1      Hand Tools
```

T100-2625	1mm ANGLED GADGET	Hand Tools
T100-2827	ORANGE 3mm	Hand Tools

When the file was imported, the execution report showed the following three errors:

- Unknown Parent Product "Hand Tools" for product "T100-3526" (1 product(s) was rejected due to this error). Source: Line=2, Column=Column3.
- Unknown Parent Product "Hand Tools" for product "T100-2625" (1 product(s) was rejected due to this error). Source: Line=3, Column=Column3.
- Unknown Parent Product "Hand Tools" for product "T100-2827" (1 product(s) was rejected due to this error). Source: Line=4, Column=Column3.

STEP could not find the Parent IDs given in Column 3 of the input file, and therefore the new products were rejected. Even though a default Parent ID was selected in the Identify Destination screen of the import wizard, STEP accepts the value for the Parent ID in the input file as an override, and rejects the new product anyway.

A reason for this execution report error could be that one of the input file's columns was mapped to be the Parent ID, and STEP could not find it. If the product existed already, it would not be moved anywhere, it would stay with its current parent. If the product did not exist, it would not be created.

Maximum Length, Illegal Value

The input file had two columns mapped: the first column was mapped both to the Product ID and Product Name (these were new products), and the second column was mapped to the attribute "Manufacturer Part Number".

G100-352654	REDD-52626278-YDS-777777-1/FFGH00KK7733249-UHP
J100-267225	REDD-5262627
D100-289827	K7733249-UHP

The file was imported and the execution report showed the following three errors:

- Illegal value "REDD-52626278-YDS-777777-1/FFGH00KK7733249-UHP" for attribute Manufacturer Part Number in product G100-352654 (Length of the value exceeds max length of domain Is trying to insert a value with '46' characters into a domain that has a maximum length of '40' characters.). Source: Line=10, Column=Column2.
- Line= at line 9: : Length of the value exceeds max length of domain Is trying to insert a value with '46' characters into a domain that has a maximum length of '40' characters.
- Line= at line 9: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEFF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_INS' ; SQL return code=20291; SQL return code=0

These errors are reporting a single problem. The first and second errors say that one of the imported part numbers had a value in the Manufacturer Part Number column that was too long, as defined by the attribute's setup. You are given the part number, the attribute name, and the value that was trying to be imported. The third error message is useful for a programmer.

Note: This type of error does not prevent the new part number from being created. The value for the Manufacturer Part Number, however, is left blank.

Out of Range, Illegal Value

This is another example of an illegal value error:

- Illegal value "250" for attribute Focal Length (ft) in product G100-352654 (Value error: Value '250' is out of range). Source: Line=9, Column=Column2.
- Line= at line 8: : Value error: Value '250' is out of range
- Line= at line 8: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEFF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_INS' ; SQL return code=20291; SQL return code=0

This group of three error messages are reporting a single problem. The first two messages indicate that a value of 250 is too large for the maximum value allowed for the attribute.

Not a Number, Illegal Value

The following tab-delimited input file had two columns mapped: the first column was mapped to the Product ID (the products already existed in STEP), and the second column was mapped to an attribute called 'Hole Diameter'.

G100-352654	1-1/2
J100-267225	2-3/4
D100-289827	17-3/16

The file was imported and the execution report showed the following nine errors:

- Illegal value "1-1/2" for attribute Hole Diameter in product G100-352654 (Value error: Value '1-1/2' is not a number). Source: Line=9, Column=Column2.
- Line= at line 8: : Value error: Value '1-1/2' is not a number
- Line= at line 8: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEFF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_INS' ; SQL return code=20291; SQL return code=0
- Illegal value "2-3/4" for attribute Hole Diameter in product J100-267225 (Value error: Value '2-3/4' is not a number). Source: Line=14, Column=Column2.
- Line= at line 13: : Value error: Value '2-3/4' is not a number
- Line= at line 13: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEFF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_INS' ; SQL return code=20291; SQL return code=0

- Illegal value "17-3/16" for attribute Hole Diameter in product D100-289827 (Value error: Value '17-3/16' is not a number). Source: Line=19, Column=Column2.
- Line= at line 18: : Value error: Value '17-3/16' is not a number
- Line= at line 18: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEf_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEf_INS' ; SQL return code=20291; SQL return code=0

Each set of three errors addresses one of three problems. That is, different information is reported for the same error in three individual messages. The first message in each group is what is most useful to the end user. In the above series, notice that the first lines that indicate that the fractional value in the input file is not allowed for the attribute 'Hole Diameter' because the value 'is not a number'. In this case, the attribute's validation type was set to Number, which does not allow fractional values.

This is another example:

- Illegal value "Hand tools" for attribute Focal Length (ft) in product J100-267225 (Value error: Value 'Hand tools' is not a number). Source: Line=14, Column=Column2.
- Line= at line 13: : Value error: Value 'Hand tools' is not a number
- Line= at line 13: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEf_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEf_INS' ; SQL return code=20291; SQL return code=0

The first message in the group tells you that the attribute value of 'Hand tools' in the input file is not allowed since the attribute is set up to expect a number.

And another example:

- Illegal value "50,70" for attribute Focal Length (ft) in product D100-289827 (Value error: Value '50,70' is not a number). Source: Line=19, Column=Column2.
- Line= at line 18: : Value error: Value '50,70' is not a number
- Line= at line 18: : Array operation failed (1 times): Error during array operation: ORA-20291: Parent key not found (valuemap.edgeid) ORA-06512: at "STEPSYS.VALUEMAP_BEf_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEf_INS' ; SQL return code=20291; SQL return code=0

The first message in the group tells you that the attribute value of '50,70' in the input file does not match the attribute's validation type of Number, which does not allow commas.

Object Existed with Another Object Type

The following tab-delimited input file had two columns mapped: the first column was mapped to the Product ID (the products already existed in STEP), and the second column was mapped to an attribute called "MFOB".

G100-352654	Yes
J100-267225	No
D100-289827	Maybe

The file was imported and the execution report showed the following three warnings and one error:

- Line=2: Product 'G100-352654' existed with another object type - supplied object type not set
- Line=3: Product 'J100-267225' existed with another object type - supplied object type not set
- Line=4: Product 'D100-289827' existed with another object type - supplied object type not set
- Illegal value "Maybe" for attribute MFOB in product D100-289827 (Not in legal values list). Source: Line=4, Column=Column2.

The first three warnings say that the products already existed in STEP with the object type of 'Product'. When the file was imported, the Identify Destination screen was skipped, and the default object type that was set in that screen was different from Product. STEP reported that it did not change the object type of the products.

The final message is an error that says that the attribute MFOB used an LOV and the value in the input file was not valid. Further investigation showed that LOV had only two valid values: Yes and No. Additionally, the LOV was set up so that no modifications could be made to the LOV by loading in data. So the value "Maybe" was rejected.

Unknown Classification

The following tab-delimited input file had two columns mapped: the first column was mapped to the product ID (the products already existed in STEP), and the second was mapped as a classification ID.

```
G100-352654      Hand tools
J100-267225      Hand tools
D100-289827      Hand tools
```

The file was imported and the execution report showed the following three errors:

- Unknown Classification "Hand tools" in product "G100-352654" (Classification 'Hand tools' does not exist in the import workspace). Source: Line=6, Column=Column2.
- Unknown Classification "Hand tools" in product "J100-267225" (Classification 'Hand tools' does not exist in the import workspace). Source: Line=10, Column=Column2.
- Unknown Classification "Hand tools" in product "D100-289827" (Classification 'Hand tools' does not exist in the import workspace). Source: Line=14, Column=Column2.

In this case, the import file requested that the products be linked to a classification folder, but STEP could not find a classification with the specified ID 'Hand tools'.

Not Privileged to Create Object

A file was imported and the execution report showed the following error:

- Line=2: Not privileged to Create product

Two scenarios can cause this error message:

1. Importing new products when the import user does not have the privilege to create new products. The import user must have all access required for the action to be performed during the import.

2. Importing a product with a product-to-product Reference Type by mapping one column of data to the Product ID and another column to a Reference, while the target product of the reference does not already exist in STEP. A reference can only be imported when the target already exists.

Unknown Asset ID

A file was imported and the execution report showed the following error:

- Line=2: Unknown Asset ID: AC200-627

This error is reported when attempting to link a Product ID to an Asset ID, but the asset does not exist in STEP.

Note: The Asset ID (not the Asset Name) is required in an import file.

Array Operation Failed

A file was imported and the execution report showed the following error:

- Line= at line 8: : Array operation failed (1 times): Error during array operation: ORA-20042: Value rejected: Attribute Gap not valid for this usertype ORA-06512: at "STEPSYS.VALUEMAP_BEf_INS", line 41 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEf_INS'; SQL return code=20042; SQL return code=0

This error is reported when attempting to load an attribute value (not shown in the error) to an attribute that is not legal for the product's object type. Upon further investigation, we found that the product's object type was 'Product'. The error message is saying that the attribute that has the ID of 'Gap' is not valid for products with the object type of 'Product'. Since the value is invalid, it is ignored.

Optimistic Locking Errors

When a file was imported, the below error was displayed in the execution report:

- OptimisticVerificationException: Optimistic locking errors were detected when flushing to the data store. This indicates that some objects were concurrently modified in another transaction. Failed objects:
[com.stibo.core.persistence.ProcessPO@33bf33bf: BGP_3604361] [java.util.ArrayList]

This error occurs when a user is trying to perform an import by modifying an attribute value for an existing product in STEP and at the same time there is another user who has access to the same product and is modifying the attribute value. In these scenario's, the product is locked for any modification and the import fails.

The Transaction has been Rolled Back: Unique Constraint

A file was imported to create a new product and the below error was thrown:

- Caught FatalDataStoreException at Mon Jun 06 13:10:16 CEST 2016: The transaction has been rolled back. See the nested exceptions for details on the errors that occurred., caused by SQLException: ORA-00001: unique constraint (STEPSYS.NODE_NODEID_UIX) violated {prepstmnt 523703057 INSERT INTO NODE (CHECKID, NAME, NODEID, NODETYPE, USERTYPE, USERTYPEID) VALUES (?, ?, ?, ?, ?, ?) [reused=5]} [code=1, state=23000]

The above error was thrown when a new product was being created but, the ID provided to create the new product already existed in STEP. ID's for objects that are created in STEP is always Unique and if a value for an ID already exists then repeating the same again is not accepted in STEP.

Note: ID's used for one product cannot be used again for another product, however, the same ID can be used for to create a Classification folder.

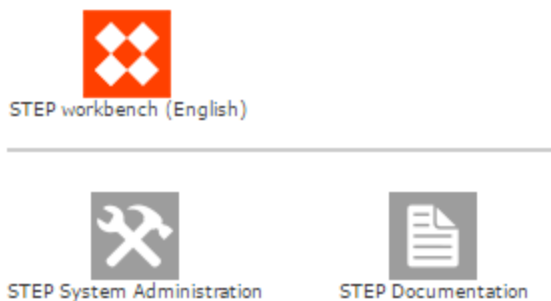
Import errors are shown in the Background processes under the Execution report and the number of errors and warning are also displayed as shown below:

Description	Importing
Execution Server	doc-rel
Progress	100%
Status	completed with errors
Created	Sun Feb 05 06:49:19 EST 2017
Started	Sun Feb 05 06:49:20 EST 2017
Finished	NA
Processing Time	0 m 2 s
Time in Queue	0 m 1 s
# of warnings	0
# of errors	1

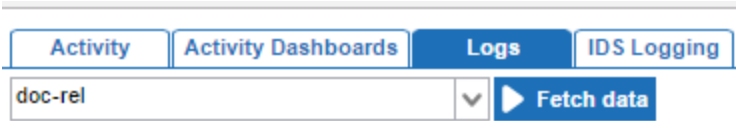
Execution Report	
1	Retrieval started (Sun Feb 05 06:49:21 EST 2017)
2	Retrieved 8817 bytes (Sun Feb 05 06:49:21 EST 2017)
3	Conversion started (Sun Feb 05 06:49:21 EST 2017)
4	Converted 6 objects (Sun Feb 05 06:49:24 EST 2017)
5	Logged on
6	Mapping started (Sun Feb 05 06:49:24 EST 2017)
7	Mapping completed (Sun Feb 05 06:49:24 EST 2017)
8	Import Started (Sun Feb 05 06:49:24 EST 2017)
9	Logged On
10	Using import mode "domain"
11	Starting first import pass (creating system setup objects)
12	Starting second import pass (importing data)
! 13	Row 6, Column : The attribute with ID ' ShortItemDescription ' isn't valid for object product with ID ' 888264
14	Imported 2 new products, 0 new classifications, 0 new entities and 0 new assets.
15	Processed 4 existing products, 0 existing classifications, 0 existing entities and 0 existing assets.
16	Skipped 0 products, 0 classifications, 0 entities and 0 assets.
17	Deleted 0 products, 0 classifications, 0 entities and 0 assets.
18	Found 0 warnings
19	Found 1 errors
20	Import completed (Sun Feb 05 06:49:26 EST 2017)
21	Error file generation started (Sun Feb 05 06:49:26 EST 2017)
22	Error file with 1 object(s) generated (Sun Feb 05 06:49:30 EST 2017)

Similarly, the execution report can also be viewed in the System Administration page by following the below steps:

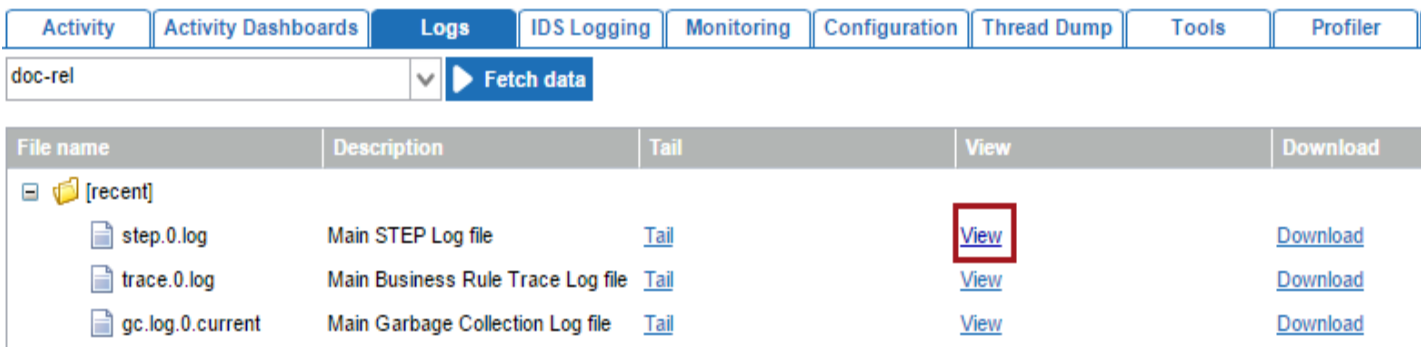
- 1. Launch the STEP WebStart page
- 2. Click on the System Administration link



- 3. Once logged in, click on the Logs Tab
- 4. Select the Server in the dropdown list and click on 'Fetch Data'



- 5. Expand [recent] and then click on view option for step.0.log file



- 6. A new window will open with the log details shown
- 7. Copy the background process ID from STEP

Background Process		Queue Info
Properties		
Property	> Value	
Started by	USERL	
Id	BGP_115400	
Description	Importing	
Execution Server	doc-rel	
Progress	100%	
Status	completed with errors	
Created	Sun Feb 05 06:49:19 EST 2017	
Started	Sun Feb 05 06:49:20 EST 2017	
Finished	NA	
Processing Time	0 m 2 s	
Time in Queue	0 m 1 s	
# of warnings	0	
# of errors	1	

- Press Ctrl + F on the keyboard and search for the background process ID to find the complete log detail with more additional details.

```
Submitting bg-process (BGP_115400) with template (stibo.Importer) to queue (IN)
2017/02/05-06:49:20 2c8 com.stibo.servicemanager.beans.DefaultBackgroundProcessInvocationImpl run INFO: Start
handling of bg-process (BGP_115400) (recovery=false)
2017/02/05-06:49:21 2c8|USERL|BGP com.stibo.util.unstable.plugin.DefaultPlugin init WARNING: Unable to find
parameter name: ValueSubstitutionAssetIDName
2017/02/05-06:49:21 2c8|USERL|BGP com.stibo.util.unstable.plugin.DefaultPlugin init WARNING: Unable to find
parameter name: WordSubstitutionAssetIDName
2017/02/05-06:49:21 2c8|USERL|BGP com.stibo.util.unstable.plugin.DefaultPlugin init WARNING: Unable to find
parameter name: LibraryAttributeTransformationIDName
2017/02/05-06:49:25 2c8|USERL|BGP com.stibo.systemconfig.ConfigUtil getPropertyTypeCheck WARNING
The property Import.IsMarkNodeFilterValuesModification is not documented, check
com.stibo.core.domain.impl.importer.ValueFilterHandlerDomain.<clinit>(ValueFilterHandlerDomain.java:38) and read
http://confluence.stibo.com/display/RD/STEP+5+configuration+properties
2017/02/05-06:49:30 2c8|USERL|BGP com.stibo.core.domain.impl.backgroundprocess.BackgroundProcessImpl updateStatus
INFO: Setting status on succeeded bg-process (BGP_115400)
2017/02/05-06:49:30 2c8|USERL|BGP com.stibo.core.domain.impl.backgroundprocess.BackgroundProcessImpl$12
lambda$run$0 INFO: Succeeded updating status of bg-process (BGP_115400)
2017/02/05-06:49:30 2c8 com.stibo.servicemanager.beans.DefaultBackgroundProcessInvocationImpl run INFO: Finished
handling of bg-process (BGP_115400)
2017/02/05-07:00:16 4f com.stibo.systemconfig.ConfigUtil getPropertyTypeCheck WARNING
The property Log.ConfigurationPackagesLogRoot is not documented, check
com.stibo.admin.FileUtil.getConfigurationPackagesLogFileNames(FileUtil.java:265) and read
http://confluence.stibo.com/display/RD/STEP+5+configuration+properties
2017/02/05-07:00:53 4e com.stibo.systemconfig.ConfigUtil getPropertyTypeCheck WARNING
The property Log.ConfigurationPackagesLogRoot is not documented, check
```

As explained above, there are common errors and the execution logs can be viewed and then correction can be made accordingly and then the file can be imported. There are other errors which can occur for different reason based on the scenario. If the errors are not correctable then those report logs can be collected and then reported to the Stibo Support team to have the issue fixed.

Additional Information About Inbound Integration Endpoints

This information is relevant to understanding additional functionality available for implementing and using an inbound integration endpoint. Included are details on the following features:

- Export inbound integration endpoint definition for comparison purposes in an external source control system in **Exporting Inbound Integration Endpoint Definitions as Comments**.
- Resolve failed background processes as described in **Handling Failed IIEP Background Processes**.
- Understand each element of an IIEP as described in **Inbound Integration Endpoint Structure**.
- Understand the effects of the transactional setting on an IIEP as described in **Inbound Integration Endpoint Transactional Settings**.
- Parallel imports involving multiple references on object types as described in **Reference Target Lock Policy on Object Types**.

Exporting Inbound Integration Endpoint Definitions as Comments

Integration endpoint definitions can be exported as comments using Advanced STEPXML. These exports are intended to be used for submission to external source control systems for comparison purposes. Users can import them into source code repository systems where they can be compared from version to version. Editing and/or importing of these files is not supported (e.g., users may not export, edit the comments, and re-import in STEP).

To export integration endpoint definitions (both inbound and outbound) for external comparison, use Advanced STEPXML and set the DefinitionsAsComments tag to 'true', as shown below. This template exports event queues and configurations, outbound and inbound integrations and configurations.

On the Select Format step of the outbound tool (either Export Manager or OIEP), choose the Advanced STEPXML format, then copy and paste the following text into the Template field:

```
<?xml version='1.0'?>
<STEP-ProductInformation DefinitionsAsComments="true">
<IntegrationEndpoints ExportSize="All"/>
</STEP-ProductInformation>
```

Select Format

Advanced STEPXML

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

Template

```
<?xml version='1.0'?>
<STEP-ProductInformation DefinitionsAsComments="true">
<IntegrationEndpoints ExportSize="All"/>
</STEP-ProductInformation>
```

An example of the output for an inbound integration endpoint definition as comments is shown below:

```
<InBoundIntegrationEndpoint...>
<!--Definition: [This will be the IEP definition. Removed for brevity.-->
[Remaining configuration is included]
</InBoundIntegrationEndpoint>
```

Note: The content of the comment field is not part of the STEPXML XSD and therefore Stibo Systems reserves the right to change the format of the output content at any time.

For more information, see the **STEP-ProductInformation Tag in STEPXML** section of the **Data Exchange** documentation.

Additionally, details about integration endpoint configuration can be exported using STEPXML. For information, see the Configuration > **Include Integration Endpoints** section of the **STEPXML Outbound Parameters** documentation.

Handling Failed IIEP Background Processes

You can reactivate a failed integration endpoint using Disable / Enable or Resume.

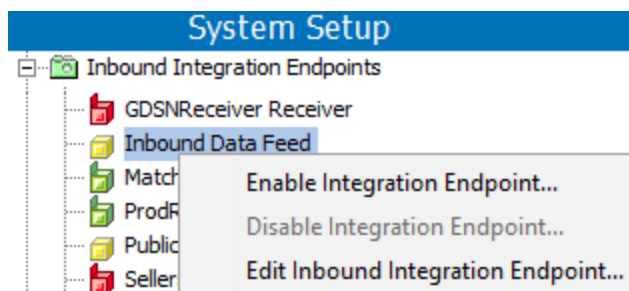
- If a background process fails and the processing of data is set to **strict**, you have to restart the endpoint before the endpoint is able to start another background process.
- If a background process fails and the processing of data is set to **chain**, the remaining background processes that belongs to the chain will also fail. You have to correct the failed background process before the endpoint is able to start another background process that belong to the same chain. The endpoint will, however, continue to process data from other chained background processes.

For more information about the strict and chain settings, see **Inbound Integration Endpoint Transactional Settings** documentation.

Disable and Enable a Background Process

Setting an integration endpoint to **Disable** and then **Enable** *does clear* the log file.

- Right-click the endpoint and select **Disable Integration Endpoint**.
- Right-click the endpoint again, and select **Enable Integration Endpoint**.

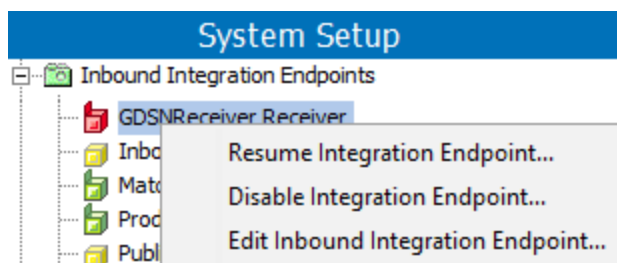


Resume a Background Process

Resuming an integration endpoint *does not clear* the log file

Note: Whenever the STEP system is patched or restarted, the main background process is automatically restarted. The integration endpoint log still exists, and continues to show an overview of the started background processes.

- Right-click the endpoint, and select **Resume Integration Endpoint**.

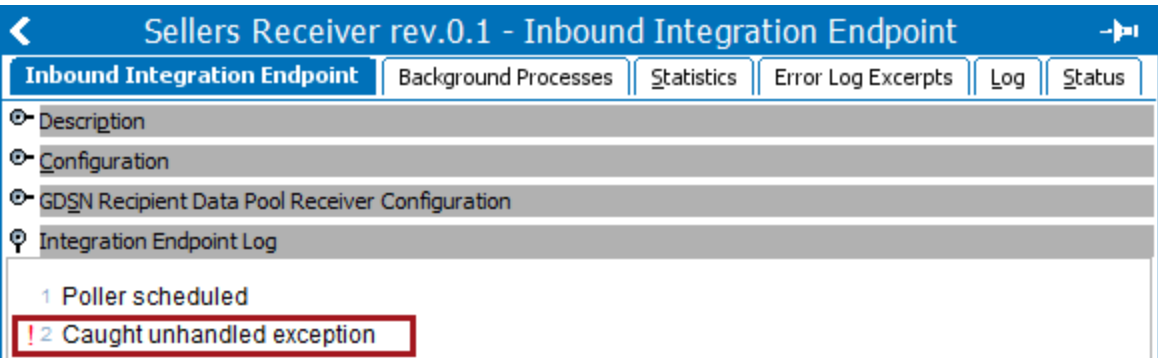


Resolving a Failed Background Process

Errors can happen before the background process starts, and also after it is running.

IIEP Fails Before the BGP Starts

In cases where the IIEP fails before the BGP is initiated, a report is available in Integration Endpoint Log on the Inbound Integration Endpoint tab. Refer to this log for information to make needed corrections.



IIEP Fails After the BGP Starts

The Error Log Excerpts tab includes details about failures within the background process.

1. In **System Setup** select the relevant integration endpoint, and then click the **Error Log Excerpts** tab to display a list of failed background processes and the description of the failure.
You can also see if a process has been suspended because it is dependent on a failed process.
2. In the **Background Process** column, hover over the background process to correct and click the **BGP** link.

Inbound Data Feed - Error Log Excerpts		
Inbound Integration Endpoint	Background Processes	Statistics
Error Log Excerpts	Log	Status
Integration Endpoint Log		
Background Pro...	Log Item No	Text
> BGP_206611	30	com.stibo.importmanager.ConversionError: Not a valid STEP XML file
> BGP_206611	40	Caught RuntimeException at Mon Feb 06 13:36:58 EST 2017: com.stibo

The Background Process editor displays.

Import started for endpoint 'Inbound Data' (2017-02-06)

Background Process
Queue Info

Properties

Property	Value
Started by	USER
Id	BGP_206611
Description	Import started for endpoint 'Inbound Data' (2017-02-06 13:36:53)
Execution Server	doc-dev
Progress	0%
Status	failed   
Created	Mon Feb 06 13:36:53 EST 2017
Started	Mon Feb 06 13:36:57 EST 2017
Finished	Mon Feb 06 13:36:58 EST 2017
Processing Time	0 m 1 s
Time in Queue	0 m 4 s
# of warnings	0
# of errors	2

Execution Report

```

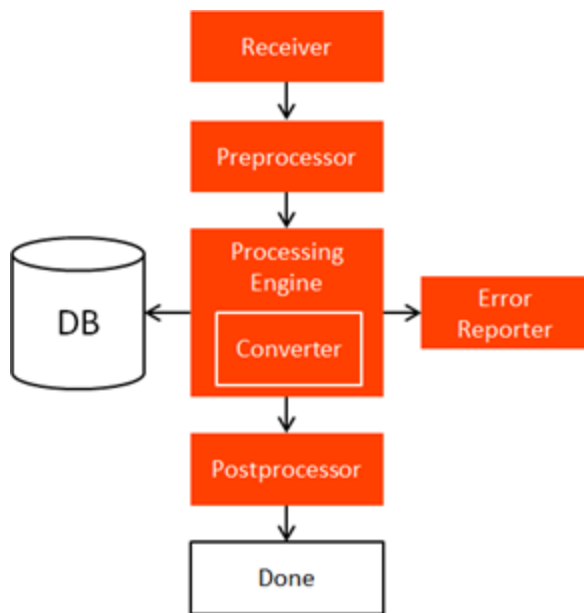
1 Processing file(s) 'upload/hotfolders/in/save/ClassProdEntity.xls' (Mon Fe
2 Conversion started (Mon Feb 06 13:36:58 EST 2017)
! 3 com.stibo.importmanager.ConversionError: Not a valid STEP XML file
   Not a valid STEP XML file
! 4 Caught RuntimeException at Mon Feb 06 13:36:58 EST 2017: com.stibo.ir

```

- In the Background Process editor **Execution Report**, view the progress of the background process, and where it failed.
- In the Properties area > **Value** column, click the Save button () near the failed status, and save the file locally.
- Open the file in a relevant editor and make the required changes.
- After completing the changes, upload the file back to the same background process. In the Background Process editor > Properties flipper > Status parameter Value column, click the Upload button () near the failed status.
- Restart the process in the Properties area > Status parameter Value column, click the Restart button () near the failed status. The background process is restarted.
- If the background process is part of a batch of chained processes, restart the other processes that are part of the chain.

Inbound Integration Endpoint Structure

The IIEP functionality has been created to allow for easy extensions and customizations, and is a plugin framework with interchangeable parts. The graphic below illustrates the structure and the colored boxes represent the interchangeable parts, which are described below.



Receiver

The receiver plugin is responsible for retrieving files or messages and passing them on to the succeeding plugins. The available receiver options are described in the **IIEP Receiver Methods** section.

Preprocessor

A preprocessor plugin (for example, an XML normalizer) has access to the file or message delivered by the receiver plugin and can manipulate or discard it. STEP does not include any predefined preprocessor plugins. For more information, see the **IIEP - Configure PreProcessor** documentation.

Processing Engine / Converter

The processing engine is responsible for performing the actual data import. The STEP Importer and Asset Importer engines are available with core functionality. If a format not supported in the standard STEP Importer is required, the processing engine can be replaced via an extension. Most often, however, creating a converter plugin for the STEP Importer is sufficient for such cases. This approach also has the benefit that the same converter plugin is available for manual imports. For more information, see the **IIEP - Configure Processing Engine** documentation.

Error Reporter

STEP includes a single Send Error Report option to report errors. An email can be sent to a specified address if errors occur when files or messages are handled by the processing engine. For more information, see the **IIEP - Configure Error Reporter** documentation.

Postprocessor

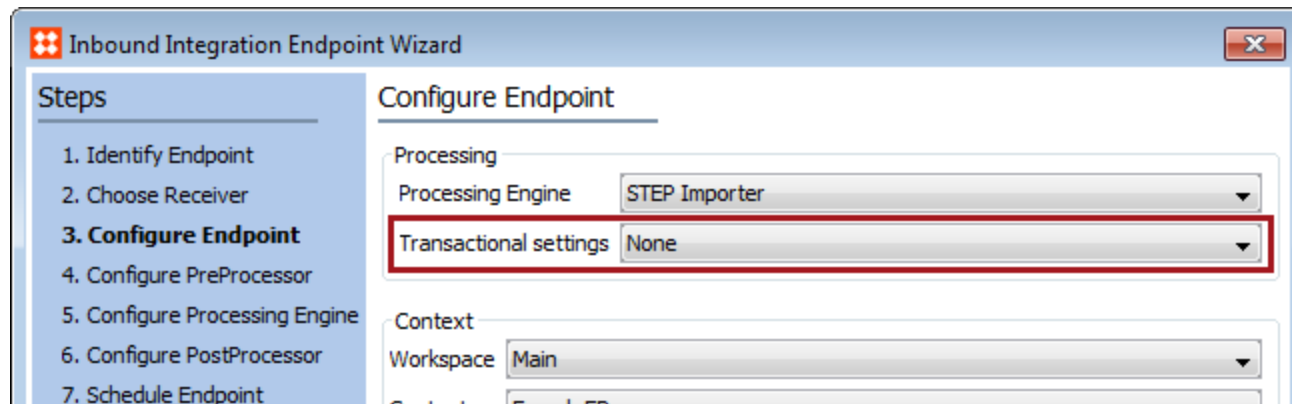
A customized postprocessor plugin has access to import events (information about what has changed) and, based on the events, can trigger any required system change. For more information on events, see the **Events** documentation in the **System Setup**.

By default, STEP does not include any pre-configured post-processor plugins, since postprocessor logic is also typically implemented via business rules referenced from the processing engine configuration.

Inbound Integration Endpoint Transactional Settings

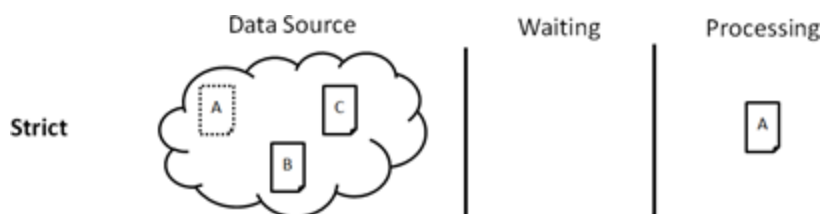
Transactional Settings specify how background processed will be generated, how the messages will be processed, and what will happen when it fails. The selected Receiver and Transactional Setting determines what other parameters are available in the additional wizard steps. There are three possible transactional settings: Strict, Chain, and None.

For more information, see the **IIEP - Configure Endpoint** documentation.



Strict

Using a strict transactional setting processes data in a strict order, using one background process at a time. When one process completes successfully, the endpoint starts the next background process. If a background process fails with an error, or if it cannot deliver a file, the next process is not started, and the 'Failed' status is set. This is common setup, unless there is a reason to use Chain or None (for assets). Strict is allowed for all receiver methods.



The queue size of a background process with strict transactional settings must be one (1). A strict transactional setting does not allow multiple background processes to work in parallel / concurrently regardless of the queue size. By definition, no new background processes can be generated while one is active.

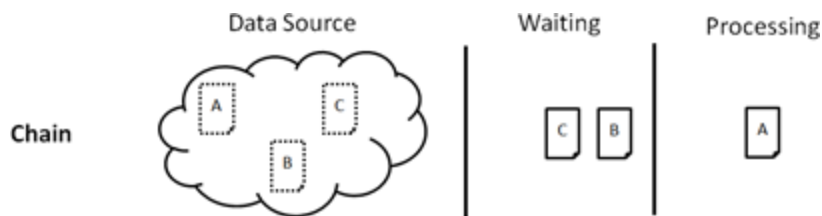
Chain

Using a chained transactional setting processes data in batches of chained background processes. A batch is generated each time the endpoint polls for data and finds more than one message or file that has not yet been processed. Chain is allowed for all receiver methods.

The **Maximum number of waiting processes** field allows you to specify the maximum number of background processes with the status 'Waiting' that is allowed. If a background process in a batch fails, the remaining background processes in the batch will also fail. However, the endpoint remains active with the status 'Running', and continues to process data in the next batch of chained background processes. A chained endpoint stops if a file cannot be delivered.

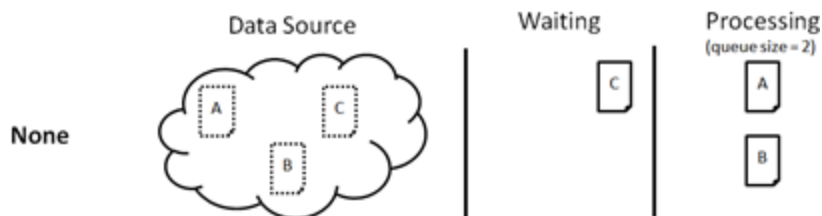
Note: When creating or modifying an endpoint for Transactional Setting = Chain, after selecting the Chain option, close the wizard and reopen it to display the 'Maximum number of waiting processes' parameter.

The queue size of a background process with chained transactional settings must be one (1). A chained transactional setting does not allow multiple background process to work in parallel / concurrently regardless of the queue size. By definition, no new background processes can be generated while one is active.



None

Using the 'none' transactional setting processes data concurrently, without any transactional restrictions or data dependencies. This is useful, for example, when processing assets. This allows the queue size to be larger than one (1). Data is not processed in a strict order and if one background process fails, the endpoint continues to process data in the next background process in the queue. None is not allowed for the Hotfolder Using File Sequence Receiver or Hotfolder Using Meta Files Receiver methods.



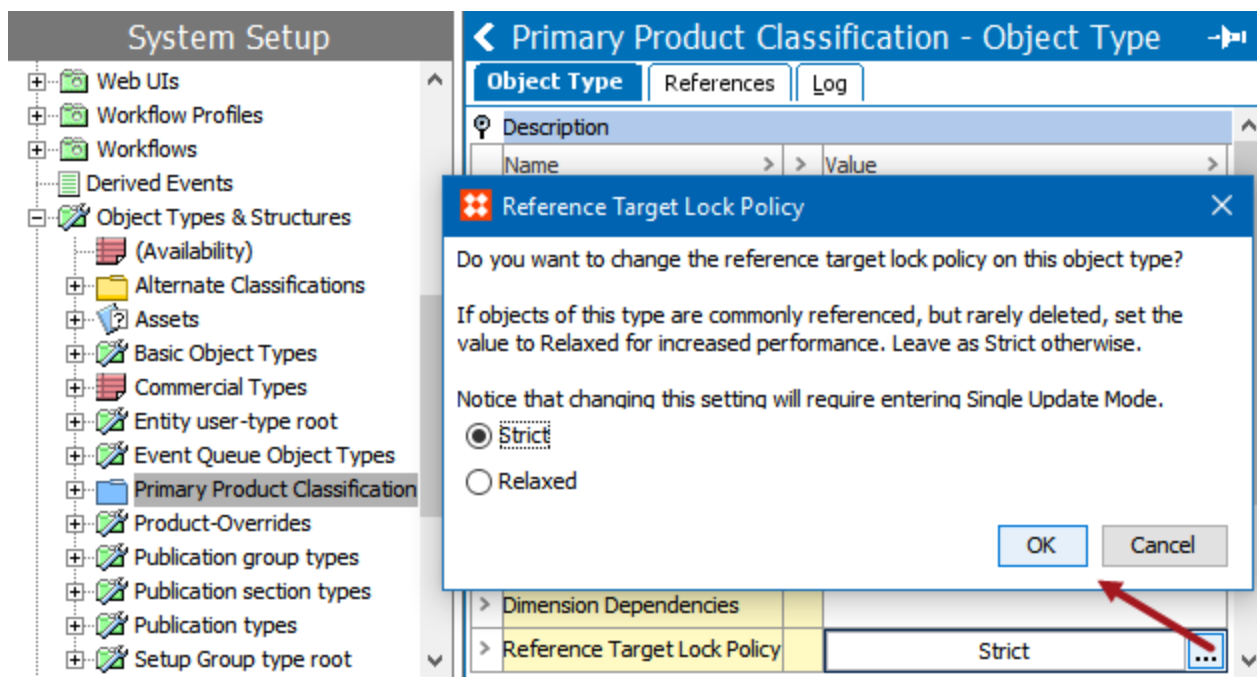
Reference Target Lock Policy on Object Types

The 'Reference Target Lock Policy' is used for the Assets, Entities, Classifications, and Products object types to manage how objects should be locked while they are being referenced. When a reference between two objects is being created in STEP, the target object is locked to ensure that it is not being deleted while the reference is being created, meaning only one process or user can edit it.

When a reference between two objects is being created in STEP, the target object will be 'locked' to ensure that it is not being deleted while the reference is being created. In most of these cases, deletion of the object is not a deletion from the database, but the deletion that occurs due to revision control when an insert of a new history entry occurs. This can be a major issue when running parallel imports where the first import locks the object type being referenced, and the second import eventually stops running because it cannot access the locked object. Although STEP continues to retry the import, this can cause inbound feeds to be negatively affected.

The 'Relaxed' setting on the property puts a less restrictive lock on the reference target objects being edited so that they can be updated concurrently by more than one process and/or user, with a full lock being employed only in the case that a deletion is attempted. STEP no longer locks the object. Instead the object type ID receives a shared 'user lock.' This means that multiple imports in parallel can reference the same objects without issues.

This setting, when used with object types that are frequently referenced but rarely deleted, will improve the performance and stability of parallel inbound imports, bulk updates, and users concurrently creating references to the same objects.



These Relaxed / Strict settings can be changed back and forth as needed, but changing the setting requires the system to enter Single Update Mode for a short time, only as long as it takes to update the setting. Keep in mind, however, that changing this setting during peak production hours can have a negative impact on performance.

Outbound Integration Endpoints

Outbound Integration Endpoint (OIEP) functionality can, in many ways, be thought of as a way to manage the standard STEP export functionality. Two types of OIEPs are available to meet the data export needs. As defined below, when creating either type of OIEP, you will use a wizard for the initial configuration, and then use the OIEP editor to manually further define your required OIEP.

Event-Based Outbound Integration Endpoint

OIEPs can be used for incremental output with an **Event-Based OIEP** which publishes data based on events occurring in STEP. An event, for example, could be generated when an object in STEP is approved. Multiple OIEPs can be configured to listen for such an event, and each will have an associated event queue for registered events. The OIEP can then, with scheduled intervals (minimum interval being once every minute), check the event queue for new events and publish data accordingly. For details on events, see the **Events** section of the **System Setup** documentation. For information about setting up an event processor, see **Event Processors** in the **System Setup**.

Setup Requirements for Event-Based OIEPs

Setting up and using a select object outbound integration endpoint involves the steps described in Creating an Event-Based OIEP:

1. Create a setup group to hold the endpoint as described in **Initial Setup for an Outbound Integration Endpoint**.
2. Launch the outbound integrations endpoint wizard as described in **Creating an Event-Based Outbound Integration Endpoint**.
3. In the wizard, add an ID, name, description, and user as described in **OIEP - Event-Based - Identify Endpoint**.
4. In the wizard, choose Select Objects for the data source as described in **OIEP - Event-Based - Choose Data Source**.
5. In the wizard, specify processing and context settings for the endpoint as described in **OIEP - Event-Based - Configure Endpoint**.
6. In the OIEP editor, configure to send an email if an endpoint-related background process fails as described in the **Error Reporter** section of the **OIEP - Configuration Flipper** documentation.
7. In the OIEP editor, specify determine when the OIEP runs as described in the **Schedule** section of the **OIEP - Configuration Flipper** documentation.
8. In the OIEP editor, specify the objects to be output, the format and pre- or post-processors (if any) as described in **OIEP - Event-Based - Output Templates Flipper**.
9. In the OIEP editor, determine how data is delivered as described in **OIEP - Delivery Method Flipper**.
10. In the OIEP editor, define what data changes will cause the OIEP to run as described in the **OIEP - Event-Based - Event Triggering Definitions Tab**.
11. In the OIEP editor, set the queue status as described in the **Event-Based OIEP Status and Queue Status**.

12. Enable the endpoint and invoke it as described in **Running an Outbound Integration Endpoint**, paying particular attention to the **Prerequisite for Event-Based OIEPs** section.

Additional Information for Event-Based OIEPs

The following information is useful once an inbound integration endpoint is set up:

1. Review use cases for an event-based OIEP as described in **Event-Based OIEP Examples**.
2. Maintain or modify the endpoint from within STEP as described in **Monitoring an OIEP via Background Process**.
3. Maintain or modify the endpoint outside of STEP as described in **Monitoring an OIEP via External Systems**.
4. Understand the effects of batching on an OIEP as described in **Event-Based OIEP Event Batching**.
5. Understand the effects of manipulating events on an OIEP as described in **Event-Based OIEP Forward, Rewind, Purge, and Republish**.
6. Understand the multithreading functionality on an OIEP as described in **Event-Based OIEP Multithreading Support**.
7. Understand queued events on an OIEP as described in **Event-Based OIEP Queued Events**.
8. Understand the relationship between OIEP Status and Queue Status as described in **Event-Based OIEP Status and Queue Status**.
9. Understand each element of an OIEP as described in **Outbound Integration Endpoint Structure**.
10. Export outbound integration endpoint definition for comparison purposes in an external source control system in **Exporting Outbound Integration Endpoint Definitions as Comments**.

Select Objects Outbound Integration Endpoint

In the simplest form, a **Select Object OIEP** uses a static set of data and works exactly like a scheduled export (created in Export Manager). This type of OIEP publishes data non-incrementally from selected hierarchies to external systems on scheduled intervals. The only difference between a Select Objects OIEP and a scheduled export in Export Manager is that the OIEP has more standard delivery options than Export Manager and offers extended monitoring capabilities.

Setup Requirements for Select Objects OIEPs

Setting up and using a select object outbound integration endpoint involves the following steps:

1. Create a setup group to hold the endpoint as described in **Initial Setup for an Outbound Integration Endpoint**.
2. Launch the outbound integration endpoint wizard as described in **Creating a Select Objects Outbound Integration Endpoint**.
3. In the wizard, add an ID, name, description, and user as described in **OIEP - Select Objects - Identify Endpoint**.

4. In the wizard, choose Select Objects for the data source as described in **OIEP - Select Objects - Choose Data Source**.
5. In the wizard, specify processing and context settings for the endpoint as described in **OIEP - Select Objects - Configure Endpoint**.
6. In the OIEP editor, configure to send an email if an endpoint-related background process fails as described in the **Error Reporter** section of the **OIEP - Configuration Flipper** documentation.
7. In the OIEP editor, determine when the OIEP runs as described in the **Schedule** section of the **OIEP - Configuration Flipper** documentation.
8. In the OIEP editor, specify the objects to be output, the format, and pre- or post-processors (if any) as described in **OIEP - Select Objects - Output Templates Flipper**.
9. In the OIEP editor, determine how data is delivered as described in **OIEP - Delivery Method Flipper**.
10. Enable the endpoint and invoke it as described in **Running an Outbound Integration Endpoint**.

Additional Information for Select Objects OIEPs

The following information is useful once a select objects outbound integration endpoint is set up:

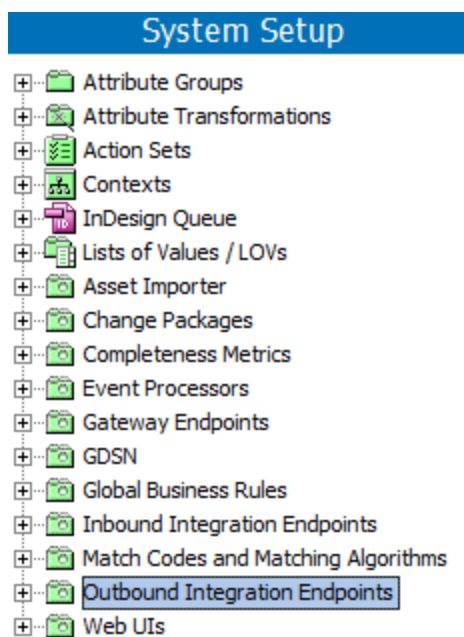
1. Determine the progress of the endpoint from within STEP as described in **Monitoring an OIEP via Background Process**.
2. Determine the progress of the endpoint outside of STEP as described in **Monitoring an OIEP via External Systems**.
3. Maintain or modify the endpoint as described in **Maintaining an Outbound Integration Endpoint**.
4. Understand each element of an OIEP as described in **Outbound Integration Endpoint Structure**.
5. Export outbound integration endpoint definitions for comparison purposes in an external source control system as described in **Exporting Outbound Integration Endpoint Definitions as Comments**.

Initial Setup for an Outbound Integration Endpoint

Before creating an outbound integration endpoint, one or more setup groups must be created that are allowed to hold outbound integration endpoints. You must also specify the setup group(s) in which outbound integration endpoints can be created. This setup only needs to be performed once, and most systems will already have it completed.

Only users with the relevant privileges can view or maintain outbound integration endpoints. For detailed information, see the **Action Sets** section and the **Users and Groups** section in the **System Setup / Super User Guide** documentation.

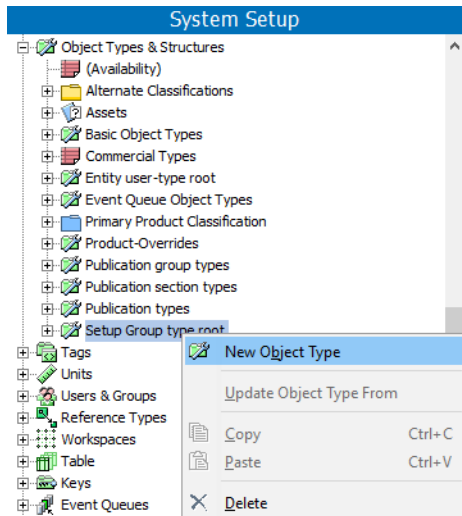
Review your System Setup tab to determine if one or more outbound integration endpoint nodes already exist. The name of the node on your system is not required to match the one in the image below.



Once the setup has been completed, the steps in this section are only needed if additional levels of organization are desired.

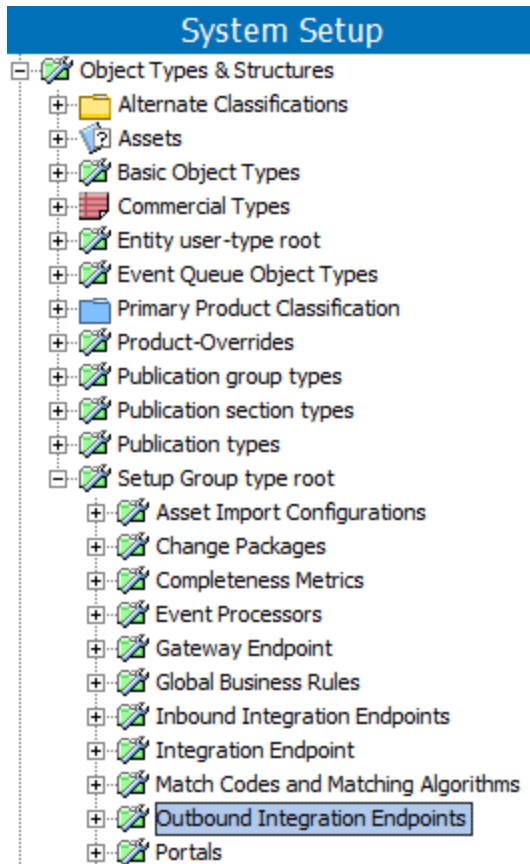
Create the Outbound Integration Setup Group

1. In System Setup, expand **Object Types & Structures**.
2. Right-click 'Setup Group type root', and choose **New Object Type**.



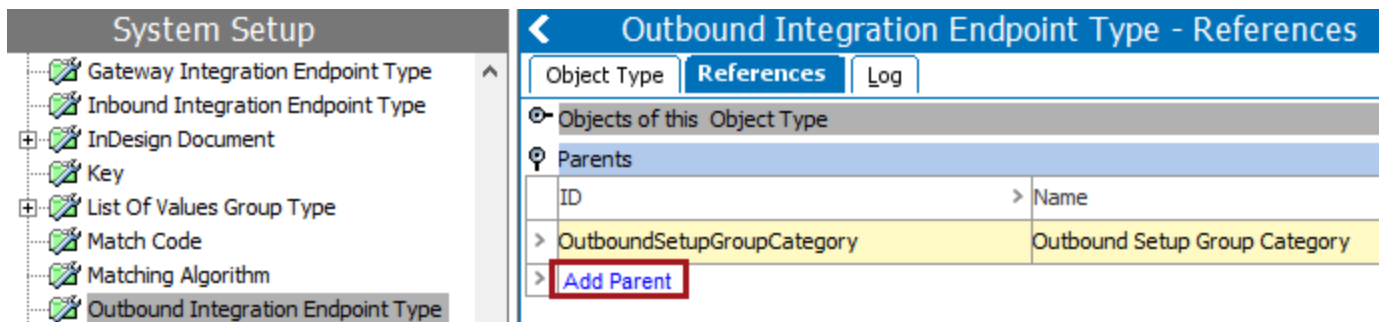
3. Enter an **ID** and a **Name**, select any required **Dimension Dependencies**, and click **Create**.

4. The new setup group appears in System Setup under 'Object Types & Structures' as a child in the **Setup Group type root**.

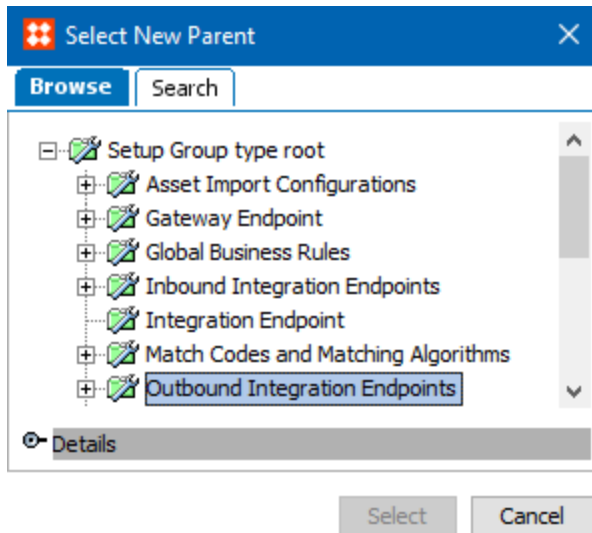


Link the OIEP Object Type to Setup Group

1. In Object Types & Structures > expand Basic Object Types > select **Outbound Integration Endpoint Type**.
2. On the References tab > Parents flipper > click the **Add Parent** link.

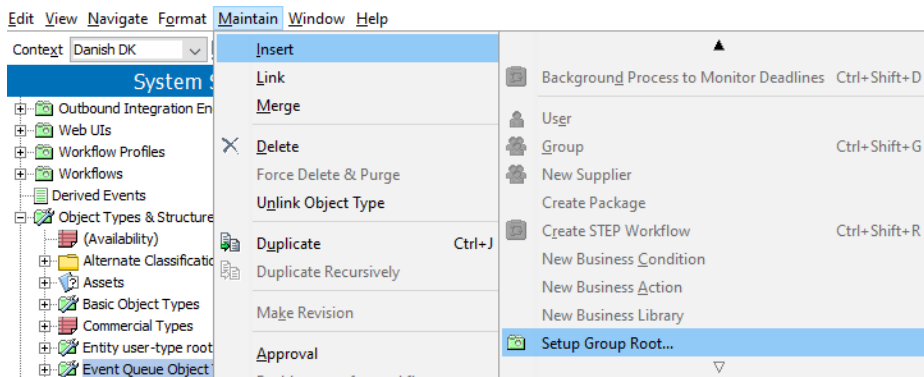


3. In the Select New Parent dialog, select the setup group you created, and click **Select** to make it a valid parent.

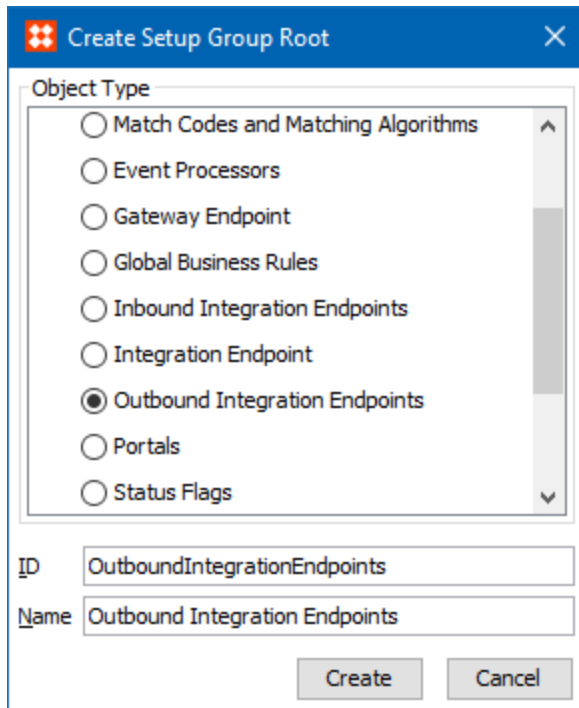


Create an Instance of the OIEP Object

1. On the System Setup tab, select any object in the **System Setup** hierarchy to activate the following Maintain menu selection.
2. Click the Maintain menu, point to Insert, and select **Setup Group Root**.



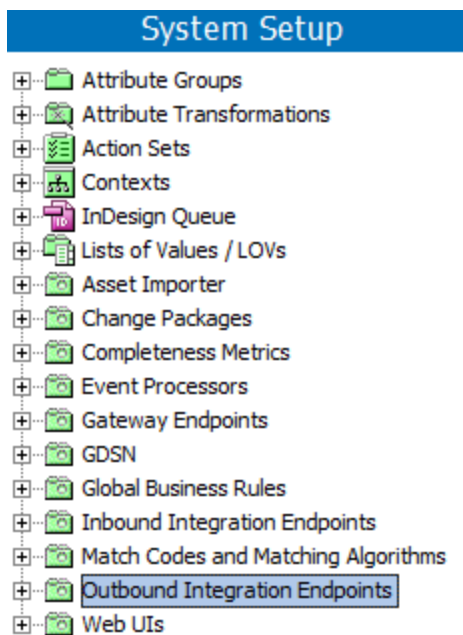
3. In the Create Setup Group Root dialog, select the outbound integration endpoints object type, enter an **ID** and a **Name**, and click **Create**.



The dialog box is titled "Create Setup Group Root" and contains the following elements:

- Object Type:** A list of radio buttons with "Outbound Integration Endpoints" selected.
 - ☐ Match Codes and Matching Algorithms
 - ☐ Event Processors
 - ☐ Gateway Endpoint
 - ☐ Global Business Rules
 - ☐ Inbound Integration Endpoints
 - ☐ Integration Endpoint
 - ☒ Outbound Integration Endpoints
 - ☐ Portals
 - ☐ Status Flags
- ID:** A text field containing "OutboundIntegrationEndpoints".
- Name:** A text field containing "Outbound Integration Endpoints".
- Buttons:** "Create" and "Cancel" buttons at the bottom right.

A setup group is created as a node in the System Setup hierarchy. Outbound integration endpoints can now be created under this new node.



Creating an Outbound Integration Endpoint

After creating a setup group for outbound integration endpoints, use the following topics to create the type of OIEP to meet your export needs:

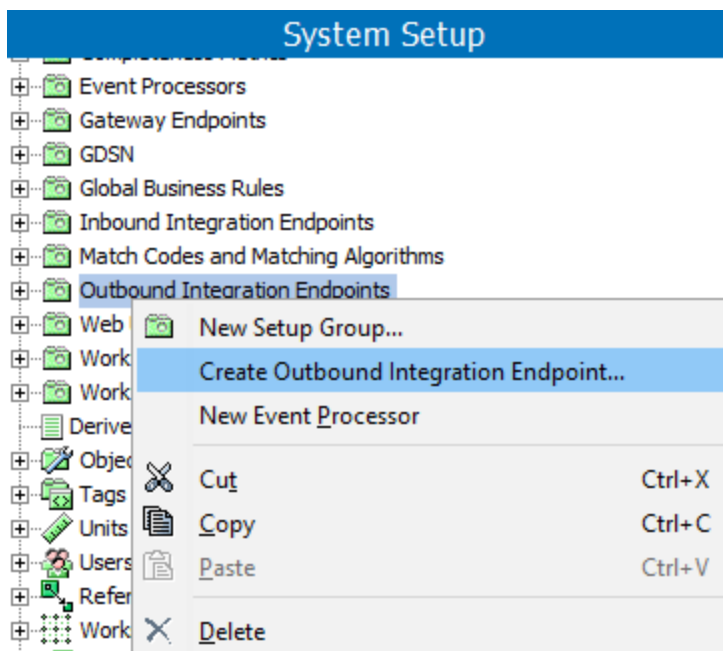
- To create an event-based OIEP that can monitor STEP for changes and export as needed, see **Creating an Event-Based Outbound Integration Endpoint**.
- To create an OIEP to send a static set of STEP data , see **Creating a Select Objects Outbound Integration Endpoint**.

Creating an Event-Based Outbound Integration Endpoint

An event-based OIEP is used to monitor for approved data changes in STEP (based on the revisability setting of the changed data), and to send the changed data to one or more external systems.

After creating a setup group for outbound integration endpoints, create an event-based outbound integration endpoint to monitor changes in STEP and output data. For information on a select objects OIEP, that can send a static set of data from STEP, see **Creating a Select Objects Outbound Integration Endpoint**.

1. In System Setup, right-click the Outbound Integrations Endpoints setup group, and click **Create Outbound Integration Endpoint**.



2. The Outbound Integration Endpoint wizard displays and involves the following steps:

Outbound Integration Endpoint Wizard

Steps

- 1. Identify Endpoint**
2. Choose Data Source
3. Configure Endpoint

Identify Endpoint

Endpoint ID

Endpoint Name

Description

User ...

Back Next Finish Cancel

- Complete setup require providing data in the wizard as well as some manual configuration. All setup includes the following sections:
 - In the wizard, **OIEP - Event-Based - Identify Endpoint** allows you to specify the name and ID of the endpoint and identify the user whose privileges are applied.
 - In the wizard, **OIEP - Event-Based - Choose Data Source** allows you to specify a static set of data or an event queue.
 - In the wizard, **OIEP - Event-Based - Configure Endpoint** allows you to specify processing and context settings for the integration endpoint.
 - In the OIEP editor, **OIEP - Event-Based - Manual Configuration** directs you to complete the required manual setup once the OIEP wizard steps are performed.

OIEP - Event Based - Identify Endpoint

The screenshot shows a window titled "Outbound Integration Endpoint Wizard" with a close button (X) in the top right corner. On the left, a "Steps" panel lists three steps: "1. Identify Endpoint" (highlighted in blue), "2. Choose Data Source", and "3. Configure Endpoint". The main area is titled "Identify Endpoint" and contains four input fields: "Endpoint ID" with the value "OutboundEndpoint", "Endpoint Name" with the value "Outbound Endpoint", "Description" with the value "Static data exported on schedule", and "User" with the value "User J (USERJ)" and a search/browse button (three dots). At the bottom, there are four buttons: "Back", "Next" (highlighted with a dashed border), "Finish", and "Cancel".

An event-based OIEP allows you to export data to be published as incremental changes are made and approved, based on the object's revisability.

1. Enter an **ID**. Common setup is to use no spaces or punctuation.
2. Enter a **Name**. Common setup is to repeat the ID with added spaces for readability.
3. Enter an optional **Description**. It is good practice to add description for future reference.
4. Use the search or browse for a **User**. The privileges of the selected user determine which actions the integration endpoint can perform and what data can be processed. A common setup is to create a special system user for this purpose so that the effects of the endpoint are easily identified and are not dependent on a particular STEP user.
5. Click **Next** to display **OIEP - Event-Based - Choose Data Source**.

OIEP - Event-Based - Choose Data Source

Outbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
- 2. Choose Data Source**
3. Configure Endpoint

Choose Data Source

Choose Data Source: Event Queue Data Source

Days to retain events: 0

Event Batching

☐ No event batching - one message is generated and handled for each event and appropriate message template will be used depending on object type and event type.

☒ Event batching - one message is generated and handled for multiple events that match the same template.

Specify maximum number of events to read for each generated message: 1000

Bundle Messages

☒ No message bundling - each message is delivered separately.

☐ Bundle messages - multiple message results will be delivered together. Specify maximum number of message results to bundle: 1

Back Next Finish Cancel

1. Choose the data source **Event Queue Data Source** for incremental exports where data should be published based on events. Creating an event-based OIEP automatically creates an event queue to process events.

Note: A **Select Objects** data source requires a separate setup, for detailed steps, see the **Creating an Outbound Integration Endpoint** documentation.

2. For the **Days to Retain Events** field, enter the number of days for the endpoint to retain events once they are read. Setting this number > 0 means that read events will stay on the queue for the specified number of days. This can be used as a security measure. If something goes wrong with the receiver, retaining read events will allow you to republish data for the retained events again later. Common setup is to use between 5 and 10 days.
3. For the **Event Batching** field, select an option to determine how many events should be included in each exported file. For more information, see **Event-Based OIEP Event Batching**.
4. For the **Bundle Messages** field, select an option to determine if multiple files generated by the same background process should be delivered together in the same message.
5. Click **Next** to display **OIEP - Event-Based - Configure Endpoint**.

OIEP - Event Based - Configure Endpoint

The screenshot shows the 'Outbound Integration Endpoint Wizard' window, specifically the 'Configure Endpoint' step. On the left, a 'Steps' sidebar lists: 1. Identify Endpoint, 2. Choose Data Source, and 3. Configure Endpoint (which is highlighted). The main area is titled 'Configure Endpoint' and contains two sections: 'Processing' and 'Multi context'. In the 'Processing' section, 'Processing Engine' is set to 'STEP Exporter', 'Transactional settings' is set to 'Strict', and 'Number of threads' is set to '1'. In the 'Multi context' section, 'Workspace' is set to 'Approved', and the 'Contexts' list contains 'English US' and an 'Add Context' link. At the bottom, there are four buttons: 'Back', 'Next', 'Finish' (which is highlighted with a dashed border), and 'Cancel'.

1. Set the **Processing** options:

- For the **Processing Engine** select STEP Exporter which uses the same functionality as the Export Manager. This is the only processing engine option on a standard STEP system and performs the actual data export. If you need a customized processing engine, please contact Stibo Systems. For more information, see **Outbound Integration Endpoint Structure** documentation.
- The **Transactional Settings** option does not have the same significance for OIEPs as they do for Inbound Integration Endpoints. The 'Strict' mode is set automatically, and while you can specify the Transactional Setting for OIEPs based on a static selection, the setting has no impact on the processing. For more information, see the **Inbound Integration Endpoint Transactional Settings** documentation.
- The **Number of threads** option is only valid for an Event Queue Data Source endpoint. The default thread setting is one (1), in which case the endpoint produces a single message at a time, with all events in the batch processed serially. Increasing the thread number results in each batch size being divided by the thread number so that the contents of a batch can be processed in parallel. When increasing the number of threads, you must also consider hardware limitations. For more information, see **Event-Based OIEP Multithreading Support**.

2. Set the **Multi context** options that the endpoint should use to export data:

- For **Workspace**, use the dropdown to select the desired workspace. A common setup is to use the Approved workspace, except when you need to generate events for objects before they are approved, for example, during import or from a workflow.
- For **Contexts**, when exporting data in the STEPXML format or when using a custom cross-context enabled format, one file can contain data from multiple contexts. For other formats, the standard 'Context splitter

Post-processor' should be used to generate separate files for the selected contexts. Click the **Add Context** link and choose one or more contexts.

3. Click **Finish** to complete the wizard.
4. Perform the manual configuration described in **OIEP - Event-Based - Manual Configuration**.

OIEP - Event-Based - Manual Configuration

After completing the Outbound Integration Endpoint wizard for an event-based OIEP, the OIEP editor includes additional parameters that must be set manually before the OIEP can run.

The configuration set from the wizard can also be modified manually using the Configuration tab.

Follow these steps to complete the manual configuration and set the event-based OIEP to run:

1. In the OIEP editor, configure to send an email if an endpoint-related background process fails as described in the **Error Reporter** section of **OIEP - Configuration Flipper**.
2. In the OIEP editor, determine when the OIEP runs as described in the **Schedule** section of **OIEP - Configuration Flipper**.
3. In the OIEP editor, specify the objects to be output, the format, and pre- or post-processors (if any) as described in **OIEP - Event-Based - Output Templates Flipper**.
4. In the OIEP editor, determine how data is delivered as described in **OIEP - Delivery Method Flipper**.

Outbound Endpoint - Configuration

Background Processes
Statistics
Error Log Excerpts
Log
Status

Outbound Integration Endpoint
Configuration
Event Triggering Definitions

Configuration

Process Engine	STEP Exporter
Error reporter 1	Not Defined
Schedule 2	Not scheduled
Queue for endpoint	OutboundQueue
Queue for endpoint processes	Out
Transactional settings	Strict
Number of threads	1
Maximum number of old processes	100
Maximum age of old processes	1w
Contexts	English US
Workspace	Approved

Event Queue Configuration

Event Actions:
Forward
Rewind
Purge
Republish

> Days to retain events	0
> Number of events to batch	1000
> Number of event batches to include per delivery	1
> Queue Status	Read Events
> Unread events (approximated)	Click to estimate ...

[Edit Configuration](#)

Output Templates 3

Object-Eventtype	> Format	> Pre-Processor	> Post-Processor	>
> Item, Sales Item (Create, ...	STEPXML	None	None	

[Add configuration](#)

Delivery Method 4

Copy to directory

> Directory	Outbound/Event
> File Name Template	\$filename-\$timestamp.\$extension
> Zip content	Yes

[Edit Delivery](#)

- In the OIEP editor, define what data changes will cause the OIEP to run as described in the **OIEP - Event-Based - Event Triggering Definitions Tab**.

Outbound Endpoint - Event Triggering Definitions

Background Processes
Statistics
Error Log Excerpts
Log
Status

Outbound Integration Endpoint
Configuration
5 Event Triggering Definitions

Triggering Object Types

Object Types	Event Filter	Generate Event
> Item, Sales Item	...	...

Add Object Type

Triggering Attributes

Name
> Item Pricing Information
> Item Supplier Information

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

6. Set the Queue Status as described in **Event-Based OIEP Status and Queue Status**.

System Setup

UnitOIEP
Website
web ssl
Web UIs
Workflow Profiles
Workflows
Derived Events
Object Types & Structures
Tags
Units
Users & Groups
Reference Types
Workspaces

Website - Configuration

Outbound Integration Endpoint
Configuration
Event Triggering Definitions
Backgrc

Configuration

Event Queue Configuration

Event Actions:
Forward
Rewind
Purge
Republish

> Days to retain events	0
> Number of events to batch	1000
> Number of event batches to include per delivery	1
> Queue Status	6 Discard Events
> Unread events (approximated)	Read Events Discard Events

Edit Configuration

7. Enable the endpoint and invoke it as described in **Running an Outbound Integration Endpoint**, paying particular attention to the **Prerequisite for Event-Based OIEPs** section.

Outbound Endpoint rev.0.3 - Outbound Integration Endpoint

Background Processes
Statistics
Error Log Excerpts
Log
Status

Outbound Integration Endpoint
Configuration
Event Triggering Definitions

Description

Name	Value
ID	Outbound Endpoint
Name	Outbound Endpoint
Object Type	Outbound Integration Endpoint Type
Revision	0.3 Last edited by USERJ on Tue Jan 17 08:10:09 EST 2017
User	user (USER)
Description	Text
Enabled	No
Endpoint Status	Stopped
Last run	2015-08-06 12:52:22
Next run	9998-01-12 00:31:00

Integration Endpoint Log

OIEP - Configuration Flipper

The OIEP Configuration flipper includes the same parameters for both Event-Based and Select Objects endpoints. Each parameter is described below.

Outbound Integration Endpoint	Configuration	Event Trigg
🔑 Configuration		
Process Engine	STEP Exporter	
Error reporter	Not Defined	
Schedule	Not scheduled	
Queue for endpoint	OutboundQueue	
Queue for endpoint processes	Out	
Transactional settings	Strict	
Number of threads	1	
Maximum number of old processes	100	
Maximum age of old processes	1w	
Contexts	Danish DK, English US	
Workspace	Approved	
⊖ Event Queue Configuration		
⊖ Output Templates		
⊖ Delivery Method		

Process Engine

STEP Exporter is the only processing engine option on a standard STEP system. If you need a customized processing engine, please contact Stiibo Systems.

The processing engine is originally selected in the wizard when configuring the endpoint. For more information, see **OIEP - Event-Based - Configure Endpoint** documentation or **OIEP - Select Objects - Configure Endpoint** documentation.

Error Reporter

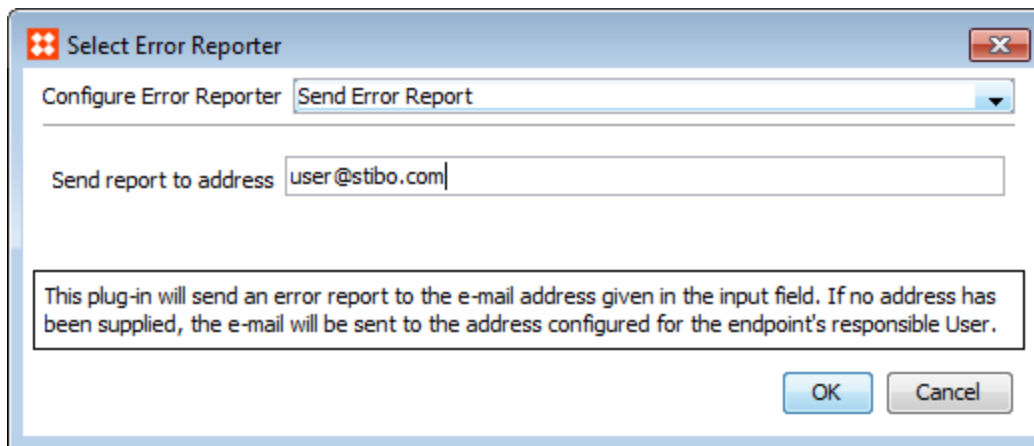
The 'Error reporter' parameter is not included in the wizard, but it can be configured to send an error report email if a background process fails. The email contains information about the failed endpoint, including the background process, failed file, failing process step, cause of the error, and a copy of the file that triggered the error. A standard STEP system (without custom extension) includes only the 'Send Error Report' option.

By default, error reports are sent when either errors or warnings are encountered. To disable error reports when only warnings are reported, update the sharedconfig.properties file on the app server to include the following property:

```
Integration.Endpoint.SendEmailErrorReportOnWarning=false
```

Configure the Error Report Email

1. On the relevant OIEP > Configuration tab > Configuration flipper > click into the **Error Report** parameter to display the ellipsis button (...).
2. Click the ellipsis button (...) to display the **Select Error Reporter** dialog.



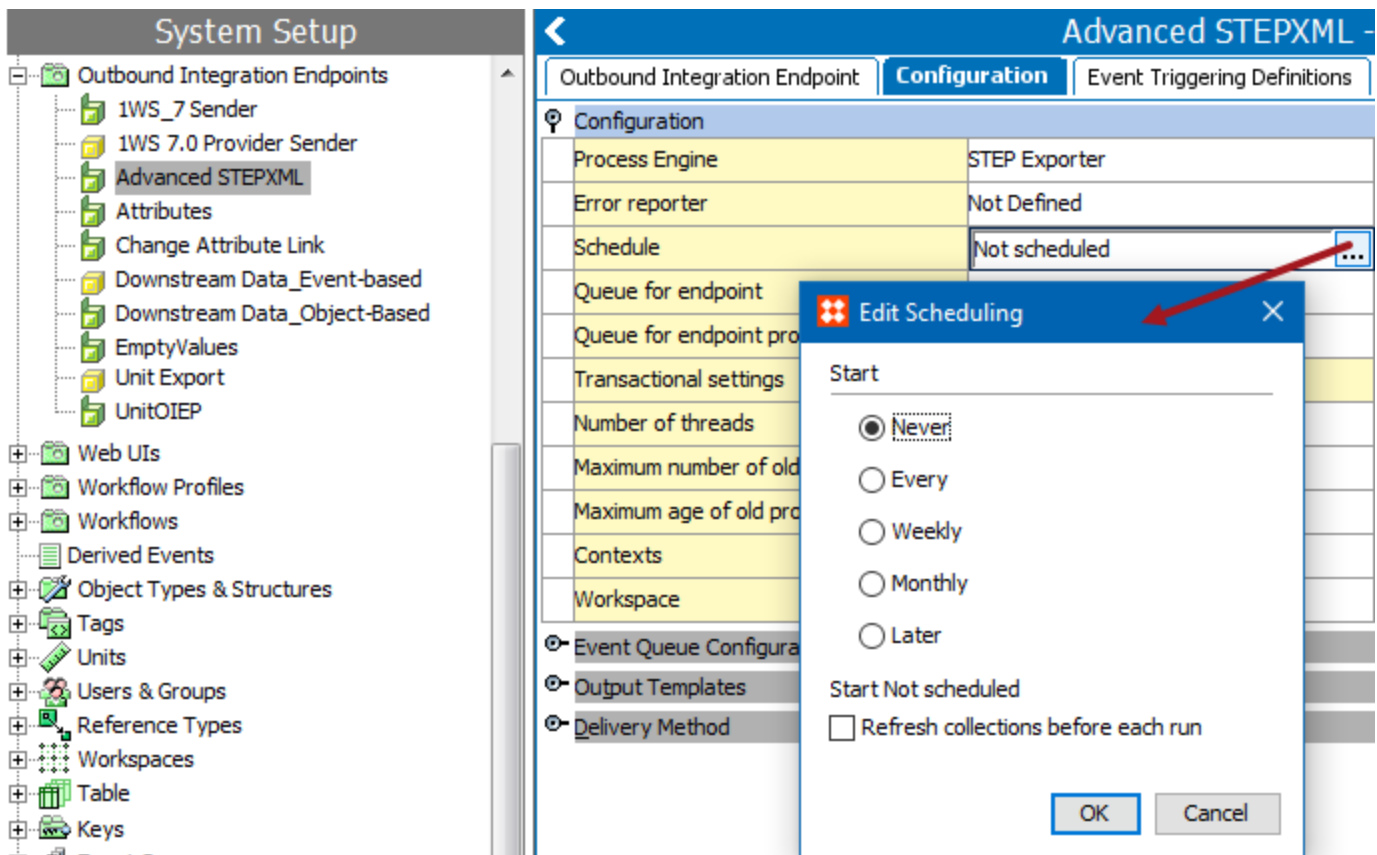
3. In Configure Error Reporter field, select **Send Error Report**.
4. In **Send report to address** field, enter one or more email addresses for the error report recipient(s). Use a semicolon-separated list to add more than one email address.

Note: If no email address is entered, the error report will be sent to the email address of the user who created the integration endpoint. If no email is defined for this user, or if no mail server is defined in the configuration, the error report will be written to the failed Background Process Execution Log.

Schedule

The schedule parameter is not included in the wizard, but it can be configured to repeatedly run the OIEP. For Select Objects OIEPs, the background processes runs as scheduled, regardless of changes to the data. For Event-Based OIEPs, the background process runs as scheduled but only when events are available for processing.

1. On the relevant OIEP > Configuration tab > Configuration flipper > click into the **Schedule** parameter to display the ellipsis button (...).
2. Click the ellipsis button (...), and then select one of the following options:



- **Never** - invoke the endpoint 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 endpoint repeatedly, every selected number of minutes. One (1) minute is the shortest interval allowed and is closest to real time. Enter the number of minutes in the text box. The selection is summarized at the bottom of the dialog.

☐ Never
 minutes

☒ Every

☐ Weekly

☐ Later

Start every minute

- **Weekly** - automatically run the endpoint repeatedly, based on the selected time, start and end dates, and days of the week. Use this option if a daily schedule is needed. The 'Start at' parameter determines the time of day that the endpoint will run. The 'Start on' parameter determines the date the endpoint will first run, while the 'End on' parameter determines the date of the endpoint's final run. The 'Every' checkboxes determine the days of the week when the endpoint will run. The selections are summarized at the bottom of the dialog.

☐ Never	Start at (hh:mm):	21:10
☐ Every	Start on (yyyy-mm-dd):	2017-07-06
☒ Weekly	End on (yyyy-mm-dd):	-
☐ Later	Every:	☒ Mon ☒ Sat ☐ Tue ☐ Sun ☒ Wed ☐ Thu ☐ Fri

Start every Mon, Wed, Sat 21:10:00 EDT, starting Thu Jul 06 2017

- **Monthly** - automatically run the endpoint 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 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' dropdown parameter selections for the week of the month and the day of the week determine when the endpoint will run. The selections are summarized at the bottom of the dialog.

☐ Never	Start at (hh:mm):	22:00
☐ Every	Start on (yyyy-mm-dd):	2017-07-06
☐ Weekly	End on (yyyy-mm-dd):	-
☒ Monthly	Every:	Third
☐ Later		Thursday

Start every third Thu 22:00:00 EDT, starting Thu Jul 06 2017

- **Later** - automatically run the endpoint only once, at the time and date specified. The selections are summarized at the bottom of the dialog.

☐ Never	Start at (hh:mm):	20:30
☐ Every	Start on (yyyy-mm-dd):	2017-07-06
☐ Weekly		
☒ Later		

Start at Thu Jul 06 20:30:00 EDT 2017

2. Select **Refresh collections before each run** if the outbound integration endpoint contains collections that should be automatically refreshed before each run.

Queue for Endpoint

'Queue for endpoint' is the name for the queue that is used by the OIEP Background Process to poll the endpoint. The first time you activate the endpoint, a queue with the specified name is created if it does not already exist. If in doubt about how to populate this parameter, create a new queue for the OIEP, for example, including the OIEP ID in the name.

Queue for Endpoint Processes

'Queue for endpoint processes' is the name for the queue that is used by the background processes started by the endpoint to handle the actual export. The queue is automatically created on the system if it does not already exist. High priority integrations or integrations with long-running processes should typically have their own queue, for example, including the OIEP ID in the queue name.

Transactional Settings

'Transactional Settings' do not have the same significance for OIEPs as they do for Inbound Integration Endpoints. The 'Strict' mode is set automatically, and while you can specify the Transactional Setting for OIEPs based on a static selection, the setting has no impact on the processing. For more information about Transactions Settings, see the **Inbound Integration Endpoint Transactional Settings** documentation.

Number of Threads

Although 'Number of threads' is available on the Configuration tab for both static and event-based OIEPs, the setting only affects event-based processes. The default thread setting is one (1), in which case the endpoint produces a single message at a time, with all events in the batch processed serially. Increasing the thread number results in each batch size being divided by the thread number so that the contents of a batch can be processed in parallel. For more information, see **Event-Based OIEP Multithreading Support**.

Maximum Number of Old Processes

'Maximum number of old processes' specifies the maximum number of ended processes the system retains. This auto-cleanup option deletes succeeded and ended processes started by the OIEP when the limitation is exceeded. The oldest processes are deleted first.

By default, the maximum number of old processes is set to 1000.

Maximum Age of Old Processes

'Maximum age of old processes' specifies the maximum age of succeeded and ended processes. Use the following case-sensitive notation: y = years, M = months, w = weeks, d = days, h=hours, m = minutes, and s = seconds.

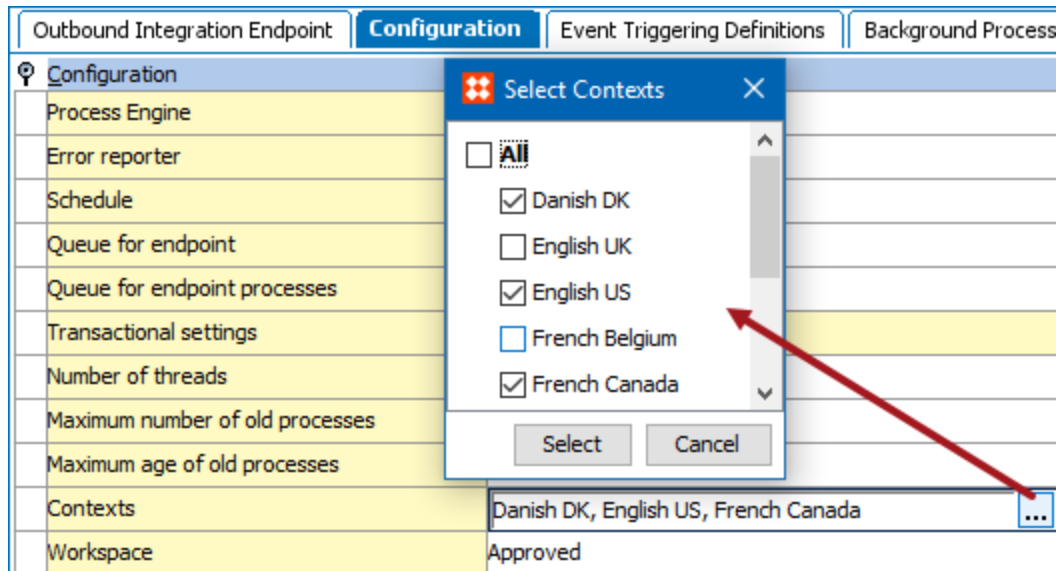
By default, the maximum age of old processes is set to one year.

Contexts

When exporting data in the STEPXML format or when using a custom cross-context enabled format, one file can contain data from multiple contexts. For other formats, the standard context splitter post-processor plugin should be used. Using this plugin, separate files are generated for the selected contexts. For more information, see the **OIEP - Post-Processor - Context Splitter** documentation.

By default , the context is set to the one which is currently being used while creating the OIEP.

Changing Selected Contexts



1. Click the Contexts field to display an ellipsis button.
2. Click the ellipsis button (...) to display the 'Select Contexts' dialog.
3. Choose the desired contexts and click the **Select** button to save.

Workspace

Workspace defines which data is used for the export. Common setup is to use the Approved workspace, except when you need to generate events for objects before they are approved, for example, during import or from a workflow.

By default, the Approved workspace is selected.

OIEP - Event-Based - Output Templates Flipper

An output template defines the data to be exported and connects it to the desired format. This enables you to use one output format for a specific object type, combined with a specific event type, and another output format for a different object type, combined with a different specific event type. Without an output template, even if events are triggered (based on the Event Triggering Definitions parameters), nothing is exported.

At least one output is required, but for event-based OIEPs, using multiple output templates can allow different messages to be generated based on the object type / event type combinations. Keep in mind that batching is restricted to a single template, so only use multiple templates when required. See more information below under the **Using Multiple OIEPs or Multiple Output Templates** section.

Configuring an output template involves the following steps:

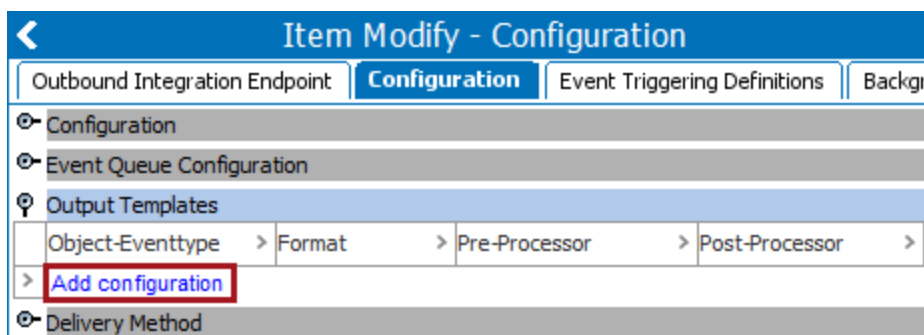
- Configure the object types and event types
- Specify the format
- Configure preprocessor and post-processor, based on availability on your system.

Important: For an event-based outbound integration endpoint, an output template must be created for each event trigger, otherwise the trigger will not output any data. For more information, see **OIEP - Event-Based - Event Triggering Definitions Tab**.

Configure the Object Types and Event Types

The object type and event type selections determine the STEP data that will be exported. Different output templates (export configurations) can be used to export data based on the event type (core events or derived events) and the object type for which the event is generated.

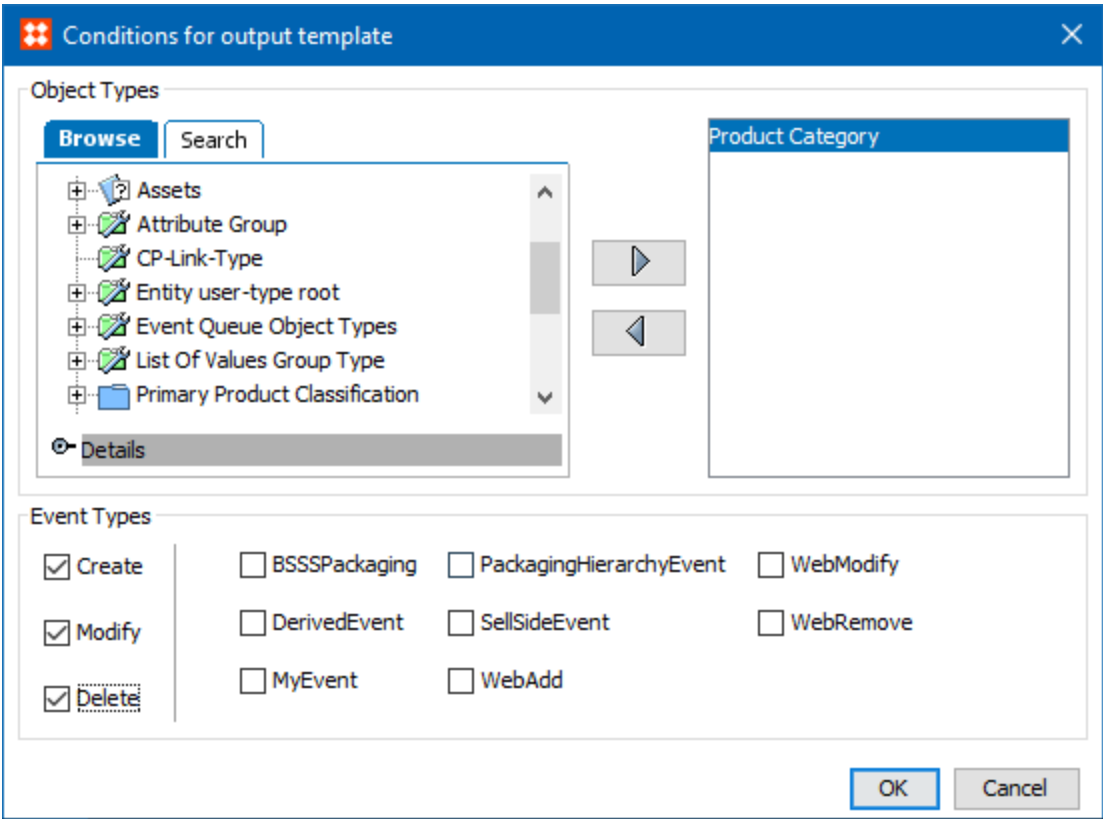
1. In System Setup, select the relevant OIEP > Configuration tab > Output Templates flipper > click the **Add configuration** link to display the 'Conditions for output template' dialog.



2. In the 'Conditions for output template' dialog, first, for the **Object Types** area at the top, select the object type (s) using the Browse or Search options. Click the right arrow button (▶) to move the object type to the right side of the dialog. To remove an object type, click the left arrow button (◀).

For the **Event Types** area at the bottom, next, select the event type. Core event types (create, modify, and delete) display on the left side and available derived event types display on the right side. Unlike core events, derived events must be created before they will display. For more information, see the **Derived Events** section of the **System Setup \ Super User Guide** documentation.

Note: Selecting an object and event type defines the object types for export, but does not necessarily define the objects that are included in the actual message, since the format of the template may dictate that parent, child, or referenced objects are included. Additionally, the advanced settings for format determine if the selected objects, their children, or both should be considered for the export.



3. Repeat the process until all required formats are displayed, keeping in mind the limitations of multiple output templates defined in the next section. Event-based endpoints can have one template for each object type / event type combination. Note that all of the available system event types (create, modify, delete) were selected for each template shown in the following image:

Output Templates			
Object-Eventtype	Format	Pre-Processor	Post-Processor
> Product Category (Create, Modify, Delete)	Generic XML (1 mappings)	None	None
> Product Family (Create, Modify, Delete)	Generic XML (1 mappings)	None	None
> Product (Create, Modify, Delete)	Generic XML (1 mappings)	None	None
> Add configuration			

4. To edit existing Object-Event type settings, click into the desired field to display an ellipsis button (...). Click the ellipsis button (...) to display the 'Conditions for output template' dialog discussed above.

Note: Only one (1) output template is allowed for each Object type and Event type combination.

Using Multiple OIEPs or Multiple Output Templates

Multiple OIEPs

- Triggering events for event-based endpoints are global. For example, you may want a change in STEP name to trigger an event on some objects, but not others. This scenario requires a different configuration of triggers on multiple endpoints.
- Delivery method is per endpoint, not per output format. If you need a different output to go to each external system, you must use multiple endpoints.
- Messages generated via events and those generated via object selection cannot be combined in a single endpoint. Even when the format is the same, this scenario requires separate message generation and therefore requires separate endpoints.

Multiple Output Templates

Multiple output templates can be created to allow data output based on a combination of specific object types and event types.

Important: Be aware that events are processed sequentially and are added to batches, as allowed, by the required output templates. Each batch also involves overhead processing operations. Depending on the data being exported, multiple output templates can result in many batches with relatively few events in each. To optimize processing, it is better to have fewer large batches instead of many small ones.

Using multiple output templates prevents the same data from being exported across all chosen object types and event types. Multiple output templates also help to reduce data output, which increases performance both for STEP and for downstream systems. In addition, the use of multiple output templates should reduce the need for customizations.

For example, consider an event-based OIEP that is configured so that when a change is made to a 'Product Category' object, the export *should not* include child objects, but when a change is made to a 'Product Family' object, the export *should* include child products. This can be achieved by configuring multiple XML output templates, and by selecting the specific object and event type.

Limitations of Multiple Output Templates

- Output files are only delivered to one destination, regardless of the number of output templates.
- Output templates encompassing all object types are only available if all object types are chosen when making the configuration. This prevents excess output from the endpoints.
- Batching occurs per template type.
- Different output formats cannot be defined for the same object type. Common setup is to avoid combining different output formats (e.g., STEPXML and CSV) in the same endpoint.

Configure the Format

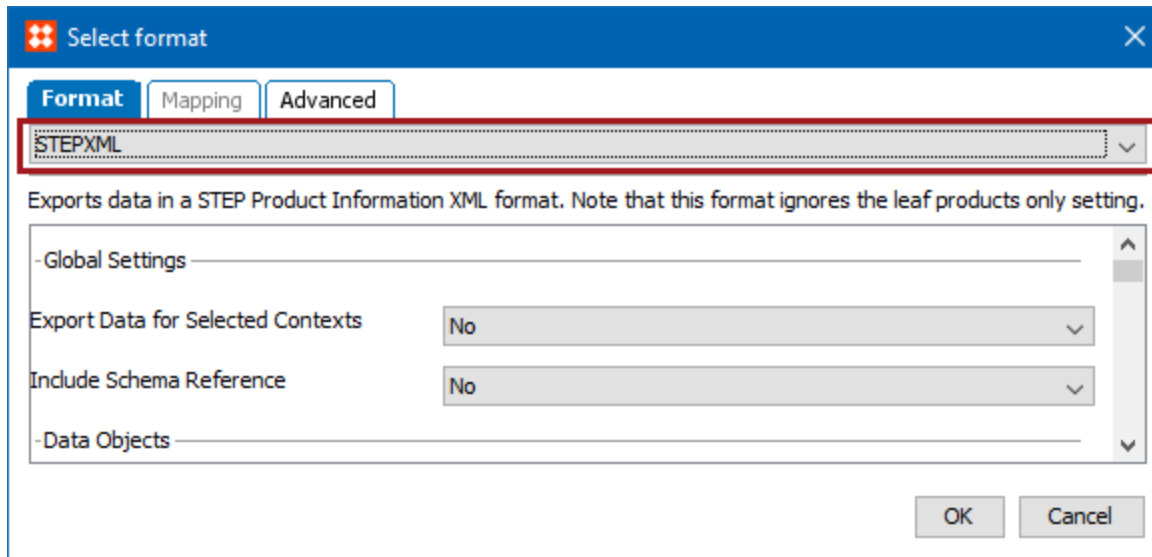
The format selection determines how the STEP data is formatted (XML or tabular).

1. Click into the format column to display the ellipsis button (...). Click the ellipsis button (...) to display the 'Select format' dialog.

Output Templates			
	Object-Eventtype	Format	Post-Processor
>	Product Category (Create, Modify, Delete)	Generic XML (1 mappings) ...	None
>	Product Family (Create, Modify, Delete)	Generic XML (1 mappings)	None
>	Add configuration		

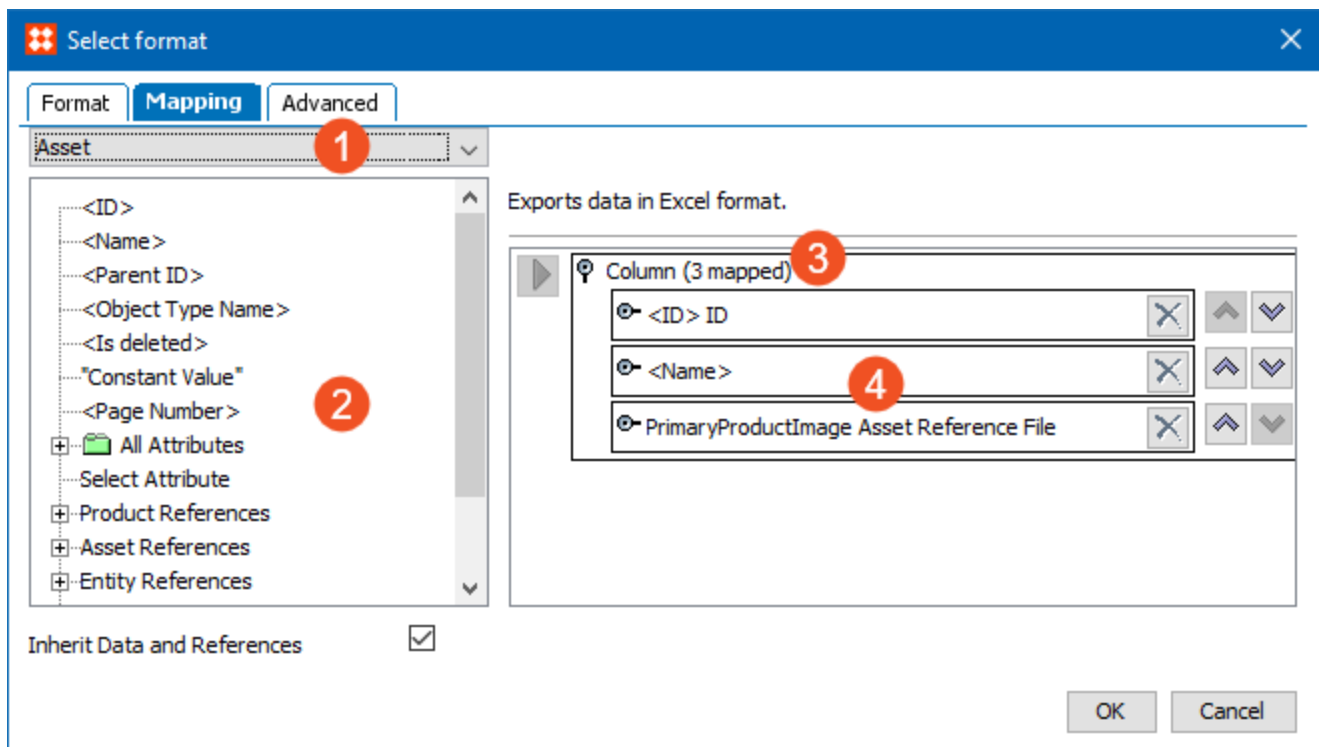
2. In the **Select format** window, on the **Format** tab, select the desired format from the dropdown. The options are the same as in the Export Manager, based on your system licenses. Use the following links for additional information:

- Advanced STEPXML Format
- Alphabetical Index - XML Format
- Ariba CIF Format
- Ariba CIF 3.0 Format
- BMEcat Format
- BMEcat 2005 Format
- CSV Format
- cXML Format
- Excel Format
- Excel Smartsheet Format
- Flatplanner Format
- Generic XML Format
- IDoc MATMAS05 Format
- Publication Excel Format
- STEPXML Format
- xCBL Format



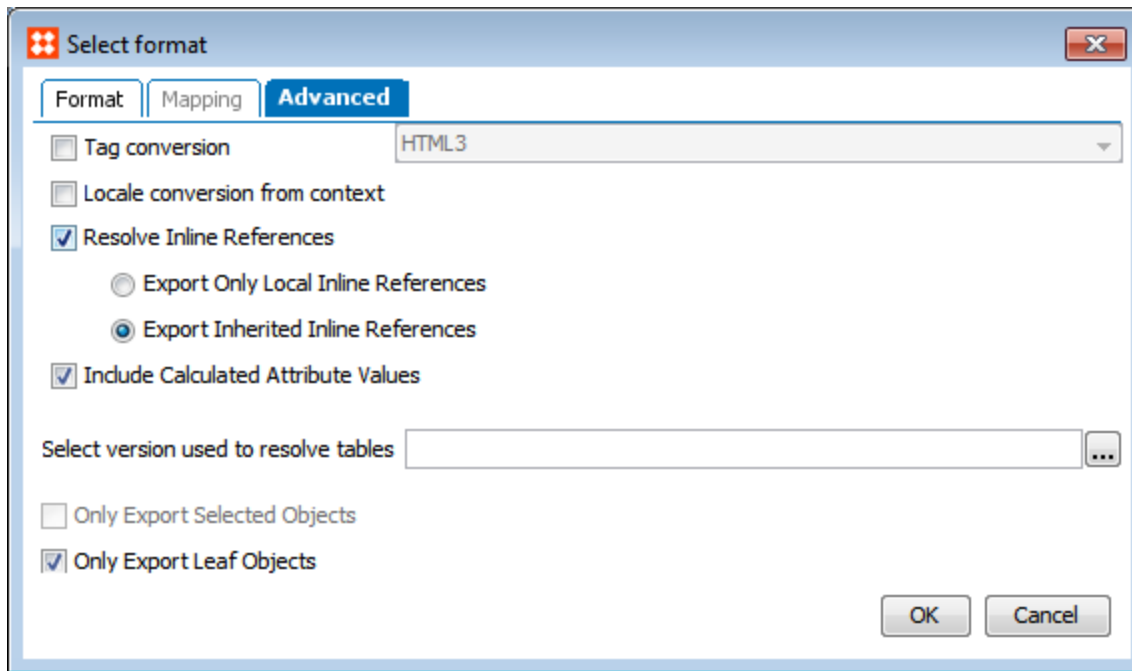
- When available, specify how to map data to the export format, including the elements identified in the image below: object super type dropdown, data source, mapping target, and mapping rules.

For more information about the object super type dropdown, see the **Export Manager - Select Objects** topic. For more information about the other elements, see the **Export Manager - Map Data** topic.



Note: The **Mapping** tab is enabled when the format selected requires mapping. Selected formats, including STEPXML and Advanced STEPXML, are automatically mapped so the Mapping tab is disabled.

- On the **Advanced** tab, a number of advanced export options can be specified.



- Tag conversion:** Converts tags to match the selected output format. This setting is optional.
 If this box is left unchecked, any formatting tags that are included in attribute values will not be converted in the outbound message. For example, if an attribute value contains bold text (which must be made bold with a STEP Style Tag, e.g., <bold>/</bold>), this tag will not be converted to its corresponding HTML output format (e.g., /) in the export. For more information on STEP Tags, see the **Tags** topic in **System Setup** documentation.
- Locale conversion from context:** Converts numbers into the numeric format that corresponds to the selected locale. By default, this option is not checked. **Note:** If Smartsheets are used for format, this must remain unchecked.
- Resolve Inline References:** Resolves inline references when they are exported. This box is checked by default, along with the option 'Export Inherited Inline References.'
 If unchecked, inline references will be exported with the inline reference tagging instead of the actual content pulled in by the inline reference. For example, if an attribute called 'Product Number' has an inline reference to pull in the STEP ID of the object (e.g. 12345), the attribute value will not contain 12345. Instead, it will contain tagging similar to the following:

```
<ref attrid="" equalsign="" includeattrname="false" resolveto="objid" separator=";" />
```


 This setting would be valid if the user planned to re-import the file at a later time and not overwrite the inline reference with a static value. For more information on inline references, see **Inline References in Attribute Values** in the **Getting Started** documentation.
- Include Calculated Attribute Values:** Resolves calculated attribute values when they are exported. By default, this option is checked. If not checked, calculated attribute values are exported with empty values,

unless the Force Calculation option has been set. The Force Calculation option is available when mapping a calculated attribute using a transformation. For more information, see **Outbound Map Data - Transform**.

The value template is exported for each selected context, including a qualifier ID, which makes it possible to import the same data back into STEP. For information about calculated attribute values, see the **Calculated Attributes** topic in **System Setup / Super User Guide** documentation.

Important: If many complex calculated attribute values (traversing hierarchies and/or references) are used, consider if they should be exported, since they can impact performance. If required, consider scheduling for non-peak times. Simple calculations are not detrimental to an export, regardless of the quantity.

- **Select version used to resolve tables:** This setting is only available if the format is STEPXML and 'Include Tables' is set to Yes. A publication version should be selected if the tables contain content relevant to a particular publication (such as column or row types that are only valid for certain publication types) or a publication version (such as commercial data). If a publication version is not selected, the version last used in the STEP workbench is used. For more information, see the **Exporting Resolved Tables** topic in the **Tables** documentation.
- **Only Export Selected Objects:** Specifies that only objects from the output template are exported. No children are exported. This setting is unchecked by default and is only available if 'Only Export Leaf Objects' is unchecked. This setting does not apply to STEPXML.
- **Only Export Leaf Objects:** Specifies that only the leaf objects (lowest level of the export object for both events and selected objects) of the selected top hierarchy are included in the export. Selected objects and triggering objects are only included if they have no children. This setting applies to the CSV or Excel format.

Note: Uncheck both 'Only Export Selected Objects' and 'Only Export Leaf Objects' if all objects (both for triggering events and object selection) and their children should be included in the export.

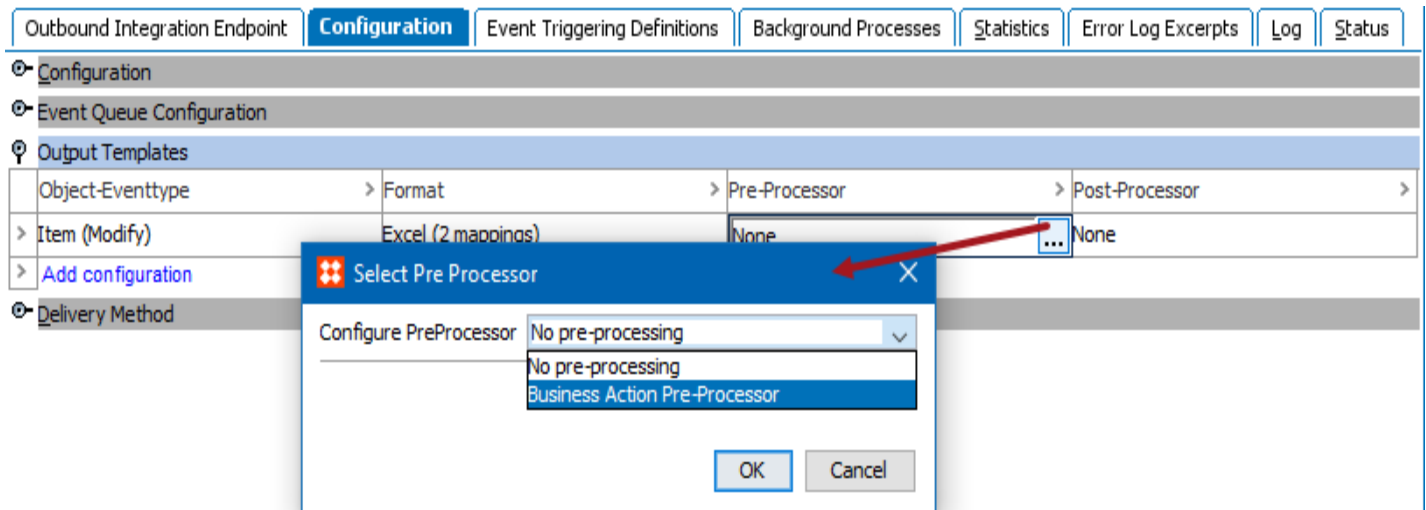
For more information, see **Export Manager - Advanced** documentation.

Configure the Pre-processor and Post-processor

Preprocessor and postprocessor options are available depending on your system configuration.

Pre-processor

For an event-based OIEP, the Business Action Pre-Processor can be applied before exporting events from a batch in an event queue. Applying the preprocessor means a node can be added to a current batch, or an event can be removed from the current batch before a message is created for an export. You can configure a preprocessor for each of the output templates. For more information, see **OIEP - Pre-Processor - Business Action**.

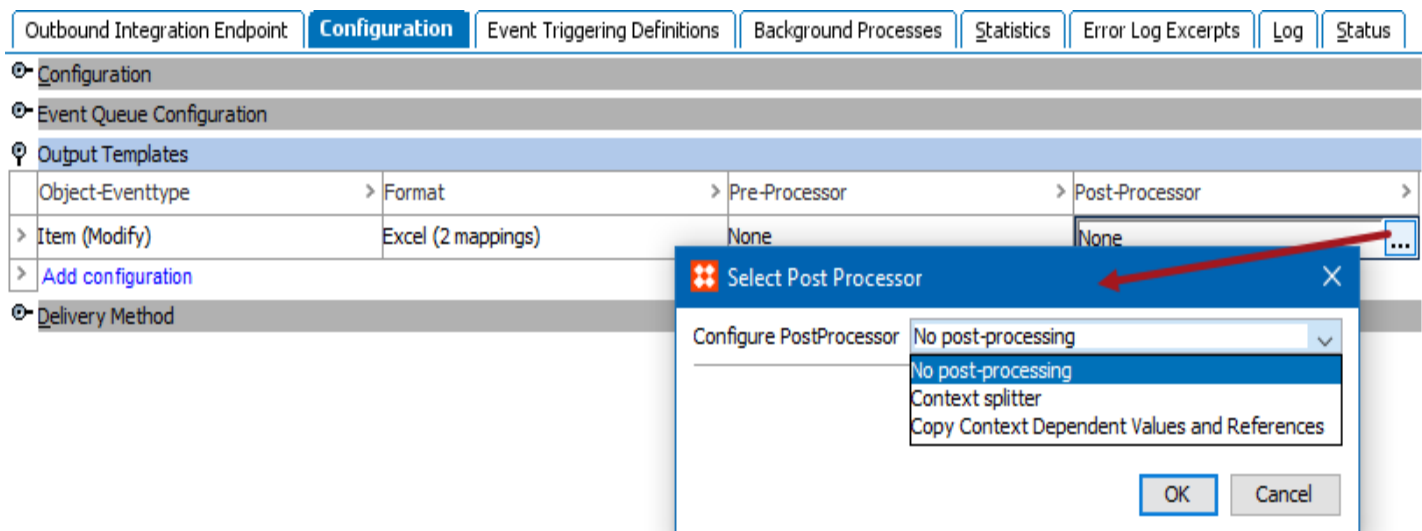


A Triggering Object Type event filter or Generate Event option can be used in place of a preprocessor. For more information, see the **OIEP - Event-Based - Event Triggering Definitions Tab** documentation.

Post-processor

Standard post-processing options should be evaluated if multiple contexts are included in the output. You can configure a post-processor for each of the output templates. The Triggering Object Type generate event option can be used in place of a post-processor. For more information, see the **OIEP - Event-Based - Event Triggering Definitions Tab** documentation.

1. Click into the Post-Processor column to display the ellipsis button (...). Click the ellipsis button (...) to display the 'Select Post Processor' dialog.



2. In the **Configure Post-processor**, select from the following options:

- **No post-processing** exports files using the standard export functionality.
- **Context splitter** generates an export file for each configured context. Each exported file will only contain context specific data. For more information, see the **OIEP - Post-Processor - Context Splitter** documentation.

Set **Copy inherited product values** parameter to **Yes** to copy and save inherited product values to the child product; or set to **No** to use inherited product values in the export, but leave the child product unmodified.

When using any format, and multiple contexts are configured for export, you must select Context Splitter, or a single context will be included in the export.

- **Copy Context Dependent Values and References** to export context-dependent values and references and add the corresponding ContextID attribute to the value or reference in question. The endpoint generates one file containing values and references for each context specified in the wizard **OIEP - Configure Endpoint**.

Set **Copy inherited product values** to **Yes** to copy and save inherited product values to the child product; set to **No** to use inherited product values in the export, but leave the child product unmodified.

If you use the **Copy Context Dependent Values and References** post-processor to add Context IDs to a cross-context export (exporting data for selected contexts), the downstream system can use the <ContextID> tag to identify specific contexts. Additionally, this facilitates re-importing the exported data into the correct contexts in STEP after processing by a third-party service or application.

For more information, see the **OIEP - Post-Processor - Copy Context Dependent Values and References** section.

OIEP - Pre-Processor - Business Action

By default, no pre-processor is included in an OIEP output template. Applying the Business Action Pre-Processor means a business rule can be run against the data, prior to the export. For example, a node can be added to a current batch, or an event can be removed from the current batch before a message is created for an export.

Using this pre-processor requires:

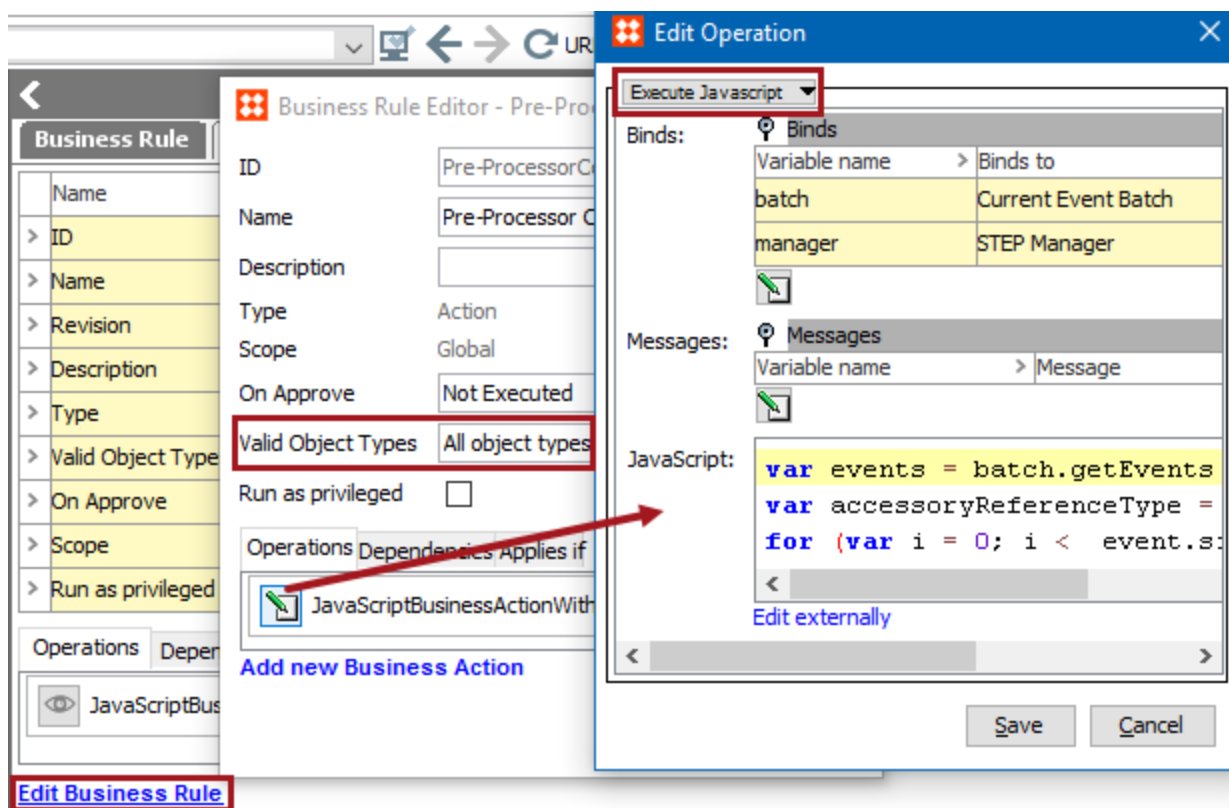
1. A business action - in this example, it will add referenced products to the products that are contained in the message.
2. An event-based OIEP configured with the business action - in this example, it is configured to deliver messages that inform a downstream system about changes to a product.

Business Action Configuration

Create a business action that is valid for 'All Object Types' (or specifically for the object types being handled by the event queue). For more information on creating business rules, see the **Specifying a Business Action** topic.

For a full JavaScript code example, see the online version of this topic.

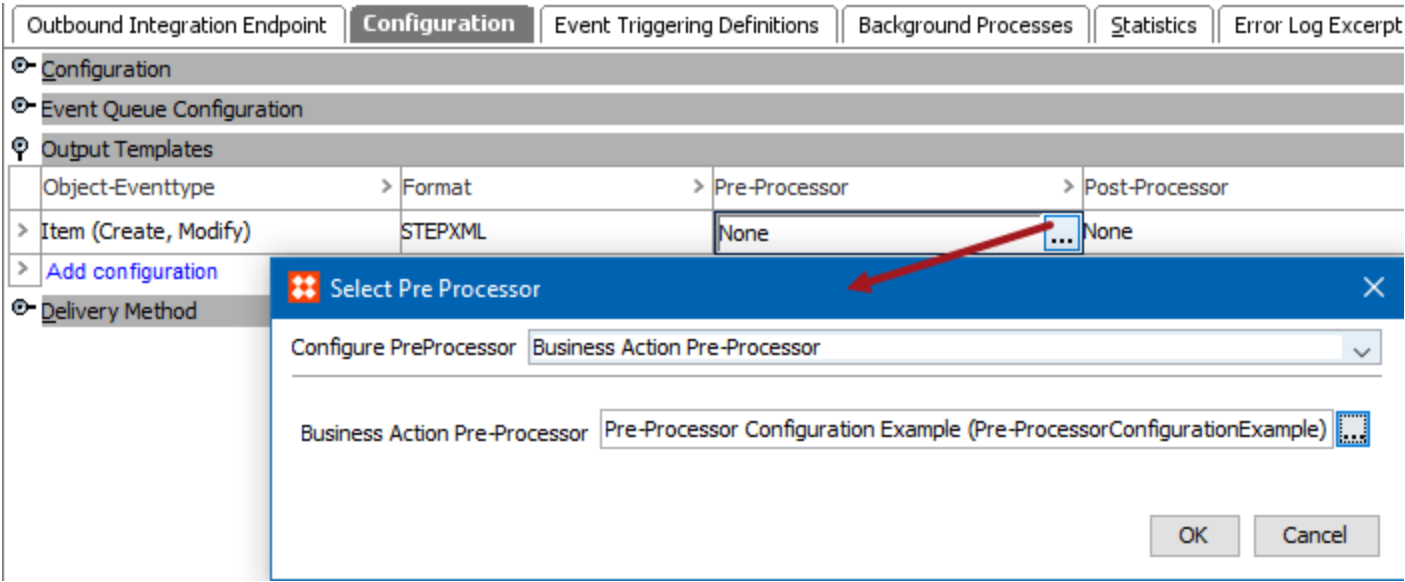
The business rule is configured in System Setup > Global Business Rules as displayed below:



OIEP Configuration

Apply the business action as a pre-processor for any of the output templates on an OIEP.

- 1. Create an event-based OIEP. For details, see the **Creating an Event-Based Outbound Integration Endpoint** documentation.
- 2. On the **Configuration** tab, open the **Output Templates** flipper and locate the output template that requires the business action.



- 3. In the **Pre-Processor** column, click the ellipsis button (...) to display the Select Pre Processor dialog.
- 4. In **Configure PreProcessor**, select the **Business Action Pre-Processor** option.
- 5. In **Business Action Pre-Processor**, select the business action created earlier. The output template is updated to show the pre-processor will run.

Output Templates			
Object-Eventtype	Format	Pre-Processor	Post-Processor
> Item (Create, Modify)	STEPXML	Business Action Pre-Processor	None
> Add configuration			

OIEP - Post-Processor - Context Splitter

The Context Splitter post processor splits the export file into individual files based on the contexts on which the data is dependent. This can be used when the data from different contexts should be sent to the external systems separately. It is also commonly used for reporting purposes for individual contexts.

The context splitter adds the <ContextID> without a <Qualifier ID>. This makes the file similar to an export of local values done from a single context. The <ContextID> tag specifies the ID of the context in which the object is exported, but does not provide the specific dimension point details which can only be supplied by the <QualifierID> tag. Using the context splitter assumes that the external system does not have any details of dependency points from STEP.

The following example illustrates using an outbound integration endpoint with the Context Splitter post-processor. It includes attribute values, but the post-processor also works with inherited references such as classification or asset references.

The product has a language-dependent attribute, 'Description, Long', with values defined in three contexts: English US, Danish DK, and French FR.

<

18217-0542 rev.0.30 - Compare Contexts

100% complete

→

Compare Contexts

References

Referenced By

View: Description, Long

	> English US	> Danish DK	> French FR
> ID	100305	100305	100305
> Name	18217-0542	18217-0542	18217-0542
> Description, Long	The best-selling T-Shirt For over 35 years, it has set the standard for T-shirt comfort and quality. Today it's better than ever, offering greater durability and less shrinkage than you'll get with ordinary tees.	Den bedst sælgende T-shirt I over 35 år har den sat standarden for T-shirt komfort og kvalitet. I dag er det bedre end nogensinde, hvilket giver større holdbarhed og mindre krympning, end du får med almindelige tees.	Le T-Shirt le plus vendu Depuis plus de 35 ans, il a établi la norme pour le confort et la qualité du T-shirt. Aujourd'hui, c'est mieux que jamais, offrant une plus grande durabilité et moins de retrait que vous ne le verrez avec les T ordinaires.

XML Output Example – Context Splitter

Using the context splitter, a separate file is generated for each context. As shown below, the names of the files output identify the Context IDs from which the objects were exported.

Name	Type
 Context1##exported_0.xml	XML File
 Context2##exported_0.xml	XML File
 Context6##exported_0.xml	XML File

The <ContextID> tag has been added to the export. The files can be distinguished by the Context ID and there is no <Qualifier ID> or any information about the dimension points. The XML output for each of the files is:

```

19 <STEP-ProductInformation ExportTime="2017-08-07 11:51:18" ExportContext="Context6" ContextID="Context1" WorkspaceID="Main" Us
20 <Products>
21 <Product ID="100305" UserTypeID="SalesItem" ParentID="18206">
22 <Name>18217-0542</Name>
23 <Values>
24 <Value AttributeID="DescriptionLong">The best-selling T-Shirt For over 35 years, it has set the standard for T-shirt

```

```

19 <STEP-ProductInformation ExportTime="2017-08-07 11:51:18" ExportContext="Context6" ContextID="Context2" WorkspaceID="Main" Us
20 <Products>
21 <Product ID="100305" UserTypeID="SalesItem" ParentID="18206">
22 <Name>18217-0542</Name>
23 <Values>
24 <Value AttributeID="DescriptionLong">Le T-Shirt le plus vendu Depuis plus de 35 ans, il a établi la norme pour le con

```

```

19 <STEP-ProductInformation ExportTime="2017-08-07 11:51:18" ExportContext="Context6" ContextID="Context6" WorkspaceID="Main" Us
20 <Products>
21 <Product ID="100305" UserTypeID="SalesItem" ParentID="18206">
22 <Name>18217-0542</Name>
23 <Values>
24 <Value AttributeID="DescriptionLong">Den bedst sælgende T-shirt I over 35 år har den sat standarden for T-shirt komfo

```

XML Output Example - Standard Cross-Context Export

Using a standard cross-context export (exporting data for selected contexts) without the context splitter, the single output file includes only the attribute values specific to particular dimension points (indicated by QualifierIDs). The XML output is:

```

129 <ValueGroup AttributeID="DescriptionLong">
130 <Value QualifierID="en-US">The best-selling T-Shirt For over 35 years, it has set the stan
131 <Value QualifierID="fr">Le T-Shirt le plus vendu Depuis plus de 35 ans, il a établi la nor
132 <Value QualifierID="Danish">Den bedst sælgende T-shirt I over 35 år har den sat standarden
133 </ValueGroup>

```


OIEP - Post-Processor - Copy Context Dependent Values and References

This post processor reduces the complexity of inheritance of values in cross-context exports, which allows downstream systems to receive a multi-context export in one single file.

Important: Using the 'Copy Context Dependent Values and References' post processor requires a strong understanding of the STEP Qualifier concept that defines in which dimensions and at which inheritance levels values are present.

If you want to further process the information in a form similar to a typical single-context export, we strongly recommend using the context-splitter post processor instead. For more information, see the **OIEP - Post-Processor - Context Splitter** topic in the **Outbound Integration Endpoints** documentation.

ContextID and QualifierID XML Tags

The 'Copy Context Dependent Values and References' post processor adds the <ContextID> and <QualifierID> XML tags to the export. A standard cross-context export (exporting data for selected contexts) only adds the <QualifierID>. The <QualifierID> tag specifies the ID of the Dimension Point in which the attribute value exists but does *not* provide the specific context. For example, the <QualifierID> of 'std.lang.all' could be the ID of the 'Language Root' Dimension Point (which would typically only be used in a Global context). But, without the addition of the <ContextID> to the export, the additional contexts that an attribute value has inherited down to cannot be determined.

When the <ContextID> XML tag is used in combination with the <QualifierID> XML tag, it is possible for the downstream system to determine the inheritance relationship between a value or reference and a context. This is useful when a service tries to re-import processed data into STEP. Because the downstream system knows the context the data must be imported into, it is easier to target specific contexts within STEP for import.

This example illustrates the differences between using the Export Manager for a standard cross-context export (exporting data for selected contexts) and using an outbound integration endpoint with the **Copy Context Dependent Values and References** post-processor enabled.

In the following, attribute values are used, but the post-processor also works with inherited references such as classification or asset references.

The product has a language-dependent attribute, 'Body Copy', with values defined in two different contexts: 'GL (Global)' and 'FR All All (French)'.

Global context attribute value:

Name	Value
Body Copy	abc Stylish, modern and lightweight, ideal for use in cafes, bars, bistros and clubs.

French context attribute value:

Name	Value
Body Copy	abc Élégantes, modernes et légères, idéales pour les cafés, les bars, les bistros ou les discothèques. Assortis aux

A third context, 'DE All All (German)', inherits its attribute values from the Global context. This is because no values have been entered in the DE All All context yet, and DE All All has an inheritance relationship with the GL context:

German context attribute value:

Name	Value
Body Copy	Stylish, modern and lightweight, ideal for use in cafes, bars, bistros and clubs.

XML Output Example - Standard Cross-Context Export

Using a standard cross-context export (exporting data for selected contexts), the XML looks like the following:

```

1 <Product ID="CHAIR" UserTypeID="Product Folder" ParentID="MyChairFolder">
2   <Name QualifierID="std.lang.all">SitWell Armchair</Name>
3   <Values>
4     <ValueGroup AttributeID="Body Copy">
5       <Value QualifierID="std.lang.all">Stylish, modern and lightweight, ideal for use in
6       <Value QualifierID="fre">Elégantes, modernes et légères, idéales pour les cafés, le
7     </ValueGroup>
8   </Values>
9 </Product>

```

Only the attribute values specific to particular dimension points—indicated by QualifierIDs—are exported, and only the attribute values that are different from the Global context. Global values are taken from 'std.lang.all' and the French values are taken from 'fre'. Since the 'Name' attribute is inherited from the 'std.lang.all' qualifier, no specific French or German 'Name' is exported. Because standard export methods collapse duplicate values, only the top value of the inheritance tree is exported.

XML Output Example - with Copy Context Dependent Values and References post processor

The following shows an XML output of an OIEP cross-context export (exporting data for selected contexts) with the **Copy Context Dependent Values and References** post-processor enabled:

```

1 <Product ID="CHAIR" UserTypeID="Product Folder" ParentID="MyChairFolder">
2   <Name ContextID="GL" QualifierID="std.lang.all">SitWell Armchair</Name>
3   <Name ContextID="FR All All" QualifierID="std.lang.all">SitWell Armchair</Name>
4   <Name ContextID="DE All All" QualifierID="std.lang.all">SitWell Armchair</Name>
5   <Values>
6     <Value AttributeID="Body Copy" ContextID="GL" QualifierID="std.lang.all">Stylish, modern and lightwei
7     <Value AttributeID="Body Copy" ContextID="FR All All" QualifierID="fre">Elégantes, modernes et légèr
8     <Value AttributeID="Body Copy" ContextID="DE All All" QualifierID="std.lang.all">Stylish, modern and
9   </Values>
10 </Product>

```

The <ContextID> tag has been added to the export along with the <QualifierID> tag, and each value—including inherited values—is explicitly exported. This means that the German values are included in the export, even though the DE All All context inherits from the GL context. Once the value is overwritten in the German context, the <QualifierID> would change to the ID of the German language dimension point.

OIEP - Delivery Method Flipper

The Delivery Method flipper in an OIEP includes the same parameters for both Event-Based and Select Objects endpoints. The delivery method controls how the output is delivered to an external system.

Open the Delivery Method flipper to display the selected delivery method. As shown below, the parameters vary based on the selected delivery method. Each is described in the **OIEP Delivery Methods** topic.

Outbound Integration Endpoint	Configuration	Event Triggering
☐	☐	☐
☐	☐	☐
☐	☐	☐
☐	☐	☐
Copy to directory		
> Directory	MultiValuePostProcessor	
> File Name Template	\$filename-\$timestamp.\$extension	
> Zip content	No	
> Edit Delivery		

Amazon SQS

For details on this delivery method, see **Amazon SQS Delivery Method**.

Amazon SQS	
> Server URL	https://sqs.us-east-2.amazonaws.com
> Credentials path	C:\Users\stibosw\aws\credentials
> Credentials profile	default
> Queue Name	myQueue
> Zip export file	Yes
> Edit Delivery	

Copy to Directory

For details on this delivery method, see **Copy to Directory Delivery Method**.

Copy to directory	
> Directory	OUT
> File Name Template	\$filename-\$timestamp.\$extension
> Zip content	Yes
> Edit Delivery	

Deploy

For details on this delivery method, see **Deploy Delivery Method**.

Deploy

> Extract to directory	workarea
> Copy to file	\$tmpdir/zip-\$timestamp.\$extension
> Edit Delivery	

Email

For details on this delivery method, see **Email Delivery Method**.

Email

> Email	OutboundData@customer.com
> Subject text	Product data export
> Body text	Please find enclosed the latest product data delivery
> Zip export file	Yes
> Edit Delivery	

FTP

For details on this delivery method, see **FTP Delivery Method**.

FTP

> Hostname	FTPDeliveryHostServerName
> Username	OIEPUser
> Password	xxxxxxxx
> File Name Template	\$filename-\$timestamp.\$extension
> Zip before upload	Yes
> Edit Delivery	

IBM Websphere MQ SSL Delivery

For details on this delivery method, see **IBM Websphere MQ SSL Delivery Method**.

IBM Websphere MQ SSL Delivery

>	Connection URL	
>	Queue Manager	
>	Queue Name	
>	Binary Payload	No
>	User Name	
>	Password	
>	Key Store	
>	Trust Store	
>	Cipher Suite	
>	Additional Parameters	
>	Edit Delivery	

JDBC

For details on this delivery method, see **JDBC Delivery Method**.

JDBC

>	Driver Location	L:/jdbc-drivers/mysql-connector-java-5.0.8-bin.jar
>	Driver Class	com.mysql.jdbc.Driver
>	Database URL	jdbc:mysql://localhost:3306/mydb
>	Username	root
>	Password	xxxx
>	Table Name	stepdata
>	Key Columns	id,datetime
>	Delete Key Columns	id
>	Edit Delivery	

JMS

For details on this delivery method, see **JMS Delivery Method**.

JMS Delivery

> JMS Connection Factory Class	ActiveMQInitialContextFactory
> Connection Factory Name	ConnectionFactory
> JMS Provider URL	tcp://ATTCM3S9:61616 =
> JMS Queue	JMSQueueForSTEP
> Binary Payload	No
> User Name	OIEPUser
> Password	xxxxxxxxxx
> Additional Parameters	
> Edit Delivery	

Mongo

For details on this delivery method, see **Mongo Delivery Method**.

Mongo delivery

> Server(s)	mongohost1:27017,mongohost1:27018,mongohost1:27019
> Raw DB prefix	raw
> User name	userauth
> Password	xxxxxxxxxxxxxx
> Cluster Enabled Configuration	Yes
> Use SSL to encrypt network traffic	No
> Key store location	
> Trust store location	
> Actions	WorkflowReview
> Edit Delivery	

Oracle AQ

For details on this delivery method, see **Oracle AQ Delivery Method**.

Oracle AQ Delivery

> Connection URL	jdbc:oracle:thin:@//server:1521/step
> Queue Name	OradeAQRecQueue
> Binary Payload	No
> User Name	OIEPUser
> Password	xxxxxxxxxx
> Edit Delivery	

REST

For details on this delivery method, see **REST Delivery Method**.

REST

> URL	http://www.Customer.com/sqlrest
> User name	OIEPUser
> Password	xxxxxxxxxx
> Zip content	Yes
> Edit Delivery	

REST Direct

For details on this delivery method, see **REST Direct Delivery Method**.

REST Direct

> URL	http://www.CustomerDirect.com/sqlrestdirect
> User name	OIEPUser
> Password	xxxxxxxxxx
> Zip content	Yes
> Header	
> Footer	
> Edit Delivery	

SFTP

For details on this delivery method, see **SFTP Delivery Method**.

SFTP

> Hostname	SFTPDeliveryHostServerName
> Username	OIEPUser
> Password	xxxxxxxx
> File Name Template	\$filename-\$timestamp.\$extension
> Zip before upload	Yes
> Edit Delivery	

Wiki

For details on this delivery method, see **Wiki Delivery Method**.

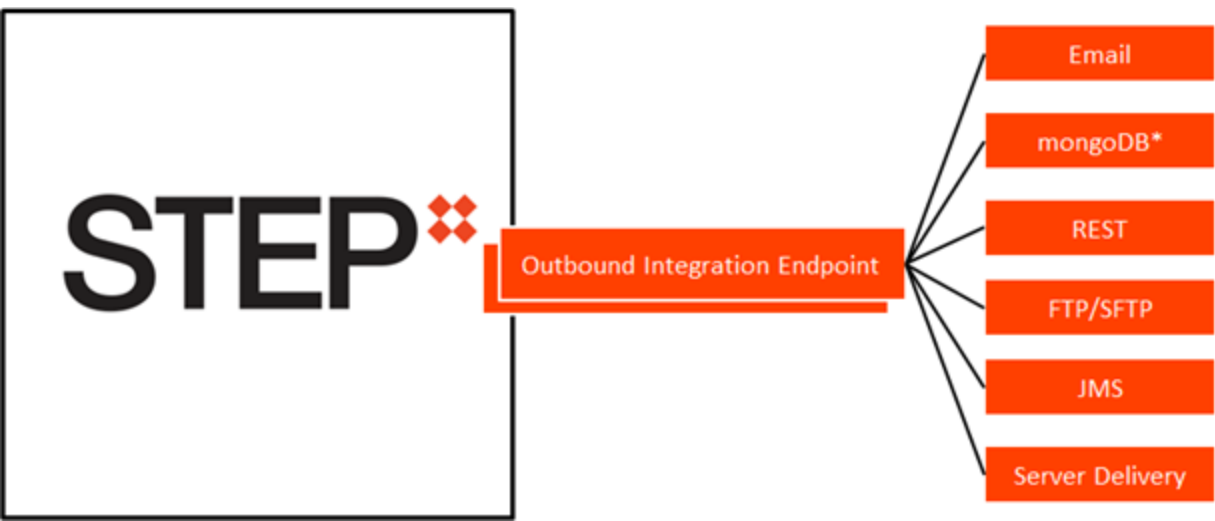
WikiDelivery

>	WikiMainSpace	STEP/
>	WikisName	wikis/xwiki/
>	WikiPassword	admin
>	WikiRestSpace	rest/
>	WikiUrl	http://vp487.vps.ohs.ca/
>	WikiUsername	admin
>	Edit Delivery	

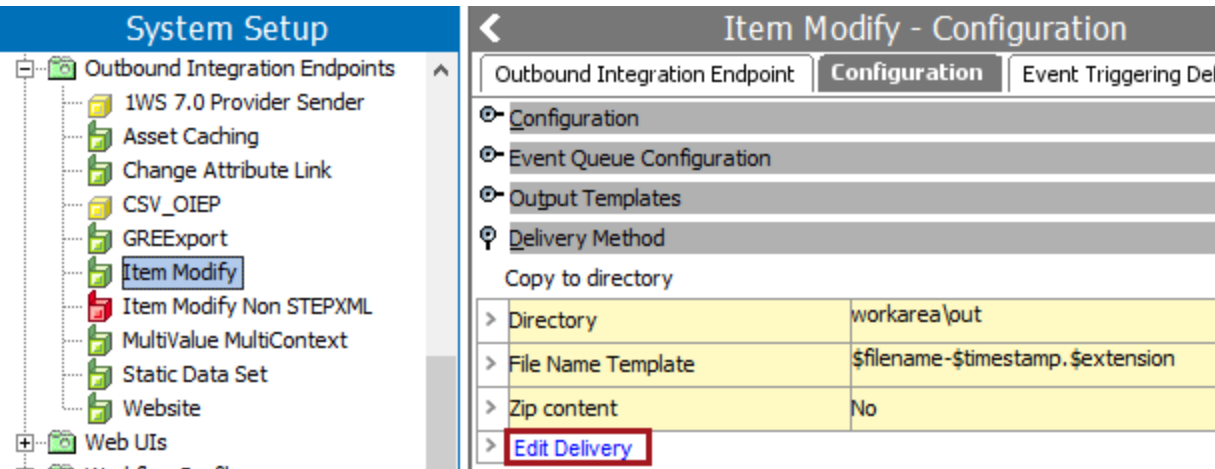
OIEP Delivery Methods

Delivery methods are used by both outbound data tools (Export Manager and OIEP) but the available options vary. For information on the delivery methods available in Export Manager, see **Export Manager - Select Delivery Method**.

An OIEP takes the data determined by its configuration and delivers it using the selected delivery method to the downstream system or on a directory on the application server.



In an OIEP, the delivery method is set in the editor on the Configuration tab, under the Delivery Method flipper.



The following delivery method options are available (although some are based on a license) when exporting using an OIEP.

Method	Description
Amazon SQS	Delivers messages to the Amazon SQS (Amazon Simple Queue Service). See Amazon SQS Delivery Method documentation.
Copy to Directory	Default selection for new OIEPs delivers files to a directory on the application server and allows the delivery to be zipped. See Copy to Directory Delivery Method documentation.
Deploy	Delivers files to a directory on the application server. Does not allow the delivery to be zipped. See Deploy Delivery Method documentation.
Email	Delivers files as an email attachment and allows a zipped delivery. See Email Delivery Method documentation.
FTP	Delivers files using file transfer protocol (FTP). See FTP Delivery Method documentation.
GDSN Data Pool	The GDSN solution must be implemented fully before the GDSN Data Pool method is available and functional. See GDSN Provider documentation.
GDSN Receiver Data Pool	The GDSN solution must be implemented fully before the GDSN Receiver Data Pool method is available and functional. See GDSN Receiver documentation.
IBM Websphere MQ SSL	IBM Websphere MQ SSL, uses MQ series Secure Sockets Layer, and enables data exchange across IBM and non-IBM platforms. See IBM Websphere MQ SSL Delivery Method documentation.
JDBC	The JDBC delivery plugin enables delivery of data to RDBMS-type databases like Oracle, MySQL, MS SQL Server, PostgreSQL, etc. See JDBC Delivery Method documentation.
JMS	The standard JMS delivery plugins (Apache Active MQ, Websphere MQ, and Oracle AQ) can be used to deliver messages to a JMS receiver without further customizations. See JMS Delivery Method documentation.
Mongo	Receives data from a STEP event queue and loads it into a MongoDB database. The MongoDB is often used for web site back-end, reporting, and high performance feeds to other back-end systems. See Mongo Delivery Method documentation.
Oracle AQ	Oracle Advanced Queuing (Oracle AQ) enables messages to be exchanged between two systems. See Oracle AQ Delivery Method documentation.
REST	The REST Delivery Method delivers a call-back URL to the REST service and does not include

Method	Description
	actual STEP data. See REST Delivery Method documentation.
REST Direct	The REST Direct delivery method differs from the standard REST delivery method in that the data is delivered directly to the REST service and no call-back URL is required. See REST Direct Delivery Method documentation.
SFTP	Delivers a file using the Secure File Transfer Protocol (SFTP). See SFTP Delivery Method documentation.
Wiki	Delivers metadata to the metadata wiki platform. See Wiki Delivery Method documentation.

Amazon SQS Delivery Method

The Amazon SQS Delivery Method delivers messages limited to 256KB to the Amazon Simple Queue Service (Amazon SQS). Common setup is to use file compression to meet the size limitation. This delivery option is only available in OIEPs and requires a separate installation recipe. Contact your Stibo representative for the latest recipe.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. For the 'Credentials profile' parameter, create a credentials file following the instructions at <http://docs.aws.amazon.com/cli/latest/userguide/cli-config-files.html>. Place the credentials file (named 'credentials', without any file extension) in the required path on the application server.
2. Prior to configuration, clicking the **Server URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **SQSDeliveryServerUrl** property. The following is an example of a property entry for a single server:

```
SQSDeliveryServerUrl=1=https://sqs.us-east-2.amazonaws.com
```

3. Prior to configuration, clicking the **Credentials Path** dropdown parameter displays the required property name. Provide a selection, including the name of the credentials file, for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **SQSDeliveryCredentialsPath** property. If necessary, use a comma and increment the number to separate multiple paths as shown in the example below.

```
SQSDeliveryCredentialsPath=1=C:\\Users\\stibosw\\.aws\\credentials,2=C:\\Users\\stibosw\\.aws\\credentials123
```

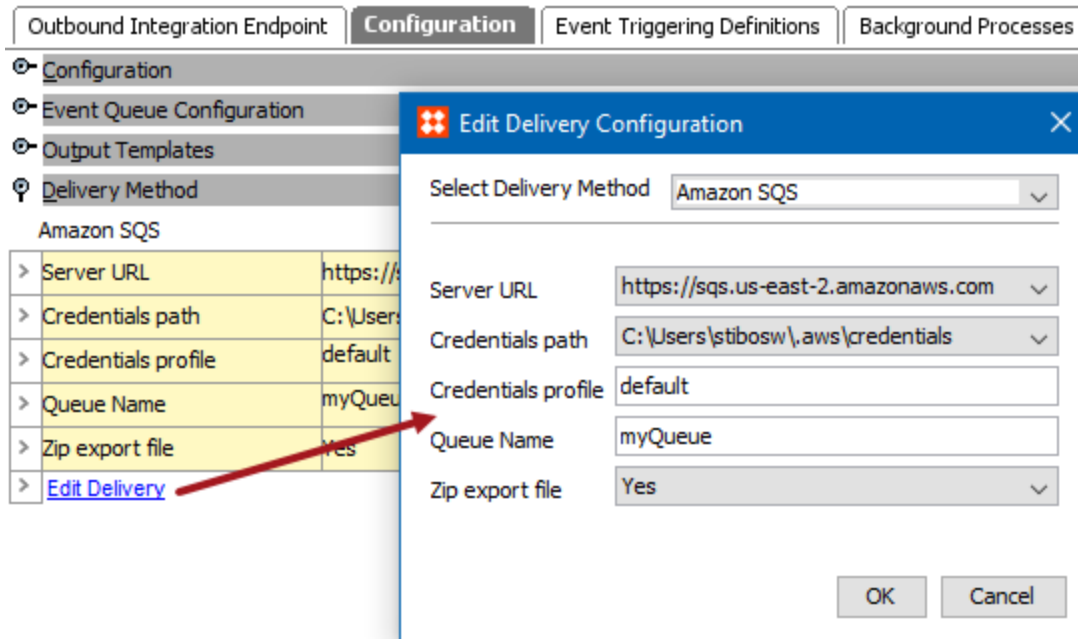
4. For the 'Queue name' parameter, if necessary, create a new queue on the SQS server. Use the steps defined in the **Amazon SQS Queue Configuration** section below.

Contact your Stibo Systems representative if you need assistance with setup.

Configuration

After completing the prerequisite steps, edit the delivery method of the OIEP. Use the following steps to configure the OIEP to use the Amazon SQS delivery option.

For information on a parameter, hover over the parameter field to display help text.

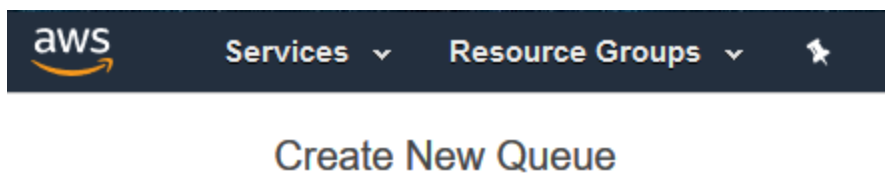


1. For **Select Delivery Method**, choose **Amazon SQS**.
2. In **Server URL**, select the URL that points to the SQS server where the data will be delivered.
3. In **Credentials path**, select the path to the desired SQS credentials file.
4. In **Credentials profile**, enter the profile name included in the SQS credentials profile.
5. In **Queue Name**, enter the Amazon queue created for this delivery.
6. In **Zip export file**, specify whether to zip the contents before delivery.
7. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Amazon SQS Queue Configuration

When required, use the following steps to create a new Amazon SQS queue.

1. Log in to the Amazon server based on your region.
2. Search for **Simple Queue Service** and select it to display the queue list page.
3. Click the **Create New Queue** button to display the Create New Queue page.



4. For **Queue Name**, type a queue name including the **.fifo** extension.
5. For the **type of queue** section, select **FIFO Queue**.

- Click the **Configure Queue** button at the bottom of the page to display the default Queue Attributes parameters.
- Check the **Content-Based Deduplication** parameter box to enable the queue.

Queue Settings

Default Visibility Timeout ⓘ 30 seconds Value must be between 0 seconds and 12 hours.

Message Retention Period ⓘ 4 days Value must be between 1 minute and 14 days.

Maximum Message Size ⓘ 256 KB Value must be between 1 and 256 KB.

Delivery Delay ⓘ 0 seconds Value must be between 0 seconds and 15 minutes.

Receive Message Wait Time ⓘ 0 seconds Value must be between 0 and 20 seconds.

Content-Based Deduplication ⓘ ☒

- Click the **Create Queue** button. The new queue is displayed in the list.

Copy to Directory Delivery Method

For OIEPs, there is no single delivery option for server side delivery, but the 'Deploy' and 'Copy to directory' options provide the same function in that they both deliver files to a directory on the application server. Common setup is to use 'Copy to directory' since it is more flexible because it allows the delivery to be zipped. Neither of these options is available in Export Manager.

For more information, see the **Deploy Delivery Method** documentation.

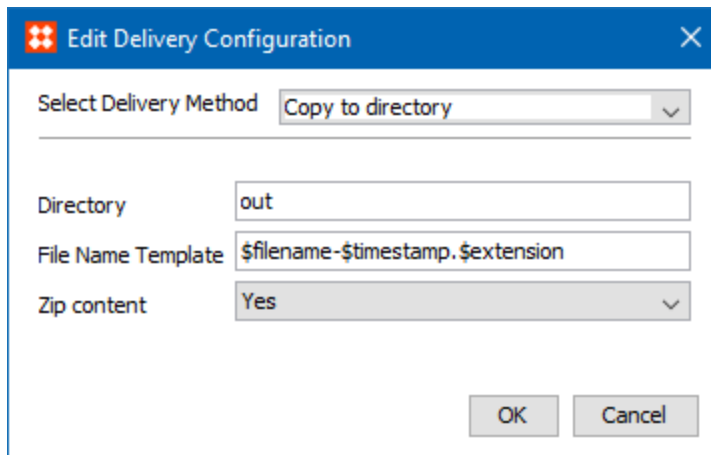
Prerequisites

The default delivery path is **opt/stibo/step** on the application server. This can be modified via the sharedconfig.properties file on the STEP application server using the case-sensitive **DirectoryDeliveryPlugin.RootDir** property. Changes to the properties file are implemented when the server is restarted.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. On the **Configuration** tab, navigate to the **Delivery Method** flipper, then click **Edit Delivery**.
2. Click the **Select Delivery Method** dropdown and choose **Copy to Directory**.



3. In **Directory**, specify the directory on the application server that will house the exported files. Do not start this text with a slash (/). Text entered will be appended to the path as defined in the Prerequisites section.
In the image above, assuming the default path is being used, the file will be delivered to 'opt/stibo/step/out'.
4. In **File Name Template**, enter the file naming template to be used for the deliveries. Each element of the template (indicated by the \$ character) is replaced with text as defined below. The template allows you to format the exported file name as required by external systems. The default template is: \$filename-\$timestamp.\$extension. Note that the hyphen (-) and the period (.) are actual characters that are used to build the complete name of the exported file.

- **\$filename** For event-based OIEPs, this variable is replaced with 'csv' for CSV format, 'excel' for Excel format, and the event ID for STEPXML format exports. For Select Object OIEPs, \$filename is replaced with 'exported'. Text can be added before or after the \$filename template, and \$filename can also be omitted altogether. When exporting multiple contexts, '##' is used as a separator within the filename to distinguish the context name for each output.
- **\$timestamp** This variable is replaced with the time in milliseconds between January 1, 1970, and the time when the file is uploaded, which allows for a unique identifier. For example, 1479230247017 is used for an export on 15 NOV 2016.

Alternatively, the following options can be used to generate a readable date and can include static characters to separate the timestamp elements. Individual elements can be ordered as required, as shown below.

\$timestamp(ddMMyyyy) displays day, month, and year and is output '15112016' for 15 NOV 2016.

\$timestamp(yyyy-MM-dd_HH.mm.ss) displays year, month, day, hour, minutes, and seconds and is output as '2016-11-15_09.32.43' for 2016 NOV 15 at 9:32.43 AM.

- **\$extension** This variable is replaced with the extension of the selected output file format.

Note: The File Name Template does not support the conversion of file formats and can only be used to deliver files in the format specified on the Configuration tab of the outbound integration endpoint.

5. For **Zip content**, select an option from the dropdown to specify if the output file should be delivered in a .ZIP (compressed) file format.
 - **Yes** uses the File Name Template and the extension .ZIP.
 - **No** uses the File Name Template for the file name along with the appropriate extension for the selected data format.
6. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Deploy Delivery Method

For OIEPs, there is no single delivery plugin for server side delivery, but the 'Deploy' and 'Copy to directory' options provide the same functionality in that they both deliver files to a directory on the application server. Common setup is to use 'Copy to directory' since it is more flexible because it allows the delivery to be zipped. Neither of these options is available in Export Manager.

For more information, see the **Copy to Directory Delivery Method** documentation.

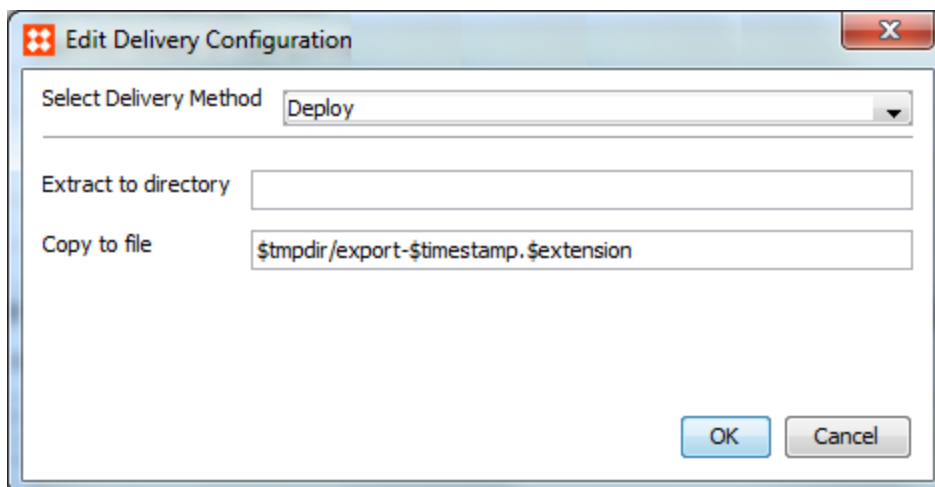
Prerequisites

The default delivery path is **opt/stibo/step** on the application server. This can be modified via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **DirectoryDeliveryPlugin.RootDir** property. Changes to the properties file are implemented when the server is restarted.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. On the **Configuration** tab, navigate to the **Delivery Method** flipper, then click **Edit Delivery**.
2. Click the **Select Delivery Method** field to display the dropdown and choose **Deploy**.



3. In **Extract to Directory**, specify a path on the application server where the exported .ZIP file should be unzipped. The field will only have an effect if the exported file is a .ZIP file, otherwise this field will be ignored.
4. In **Copy to file**, specify a location and file name template to define a path where exported file should be delivered. Static text is also allowed in addition to, or instead of, the template options (which are indicated by the initial \$). The elements in the example template shown in the image are described below:
 - **\$tmpdir** This variable points to the directory path defined by the **ApplicationServer.TmpDir** entry in the `sharedconfig.properties` file. Remove '\$tmpdir' to enforce a location relative to the **DirectoryDeliveryPlugin.RootDir** delivery path in the `sharedconfig.properties` file.

- **export** This represents and can be replaced with any static text required in the file name. For example, the Copy to File field could read 'out/EmergencySKUExport-\$timestamp.\$extension' instead, and would deliver the output to the 'out' directory below the path indicated in the properties file, and the file name would start with the text 'EmergencySKUExport-' before adding the timestamp, a period, and the file extension.
- **\$timestamp** This variable is replaced with the time in milliseconds between January 1, 1970, and the time when the file is uploaded, which allows for a unique identifier. For example, 1479230247017 is used for an export on 15 NOV 2016.

Alternatively, the following options can be used to generate a readable date and can include static characters to separate the timestamp elements. Individual elements can be ordered as required, as shown below.

\$timestamp(ddMMyyyy) displays day, month, and year and is output '15112016' for 15 NOV 2016.

\$timestamp(yyyy-MM-dd_HH.mm.ss) displays year, month, day, hour, minutes, and seconds and is output as '2016-11-15_09.32.43' for 2016 NOV 15 at 9:32.43 AM.

- **\$extension** This variable is replaced with the extension of the selected output file format.

Note: The 'Deploy' template does not support conversions of file formats and can only be used to deliver files in the format specified on the Configuration tab of the outbound integration endpoint.

5. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Email Delivery Method

The email delivery option is very straight forward and, like 'Copy to directory', it allows the delivery to be zipped. It is typically used to test the export output, or to deliver a small export to one or more persons as a one-time occurrence.

This delivery method is available in Export Manager and in OIEPs. For information on a parameter, hover over the parameter to display help text.

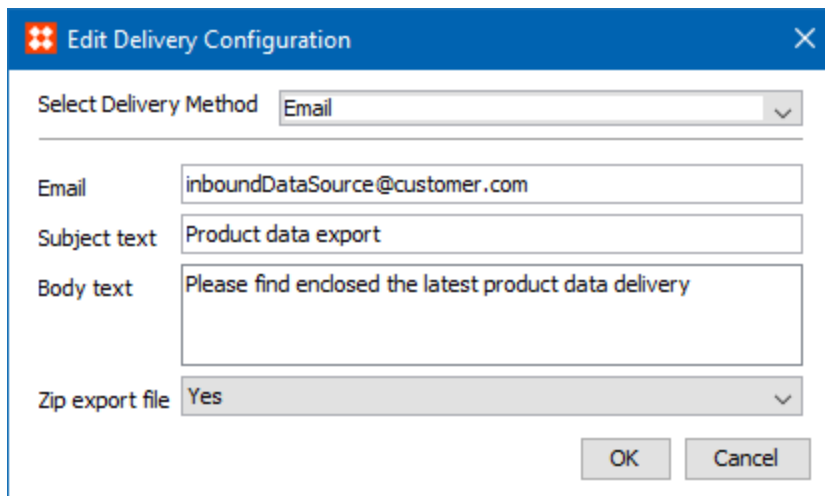
Prerequisites

To deliver an exported file via email, the STEP server must have access to the SMTP server. Be aware of email file size limits, since depending on export contents, the file could be very large.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. Click the **Select Delivery Method** parameter to display the dropdown and choose **Email**.
2. For the **Email** parameter, enter an email address.
3. For the **Subject text** parameter, enter the text that will display for the email subject line.
4. For the **Body text** parameter, enter the text that will display for the email body.



The screenshot shows a dialog box titled "Edit Delivery Configuration". It has a blue header bar with a close button (X) on the right. Below the header, there is a "Select Delivery Method" dropdown menu currently showing "Email". Underneath, there are three text input fields: "Email" with the value "inboundDataSource@customer.com", "Subject text" with the value "Product data export", and "Body text" with the value "Please find enclosed the latest product data delivery". At the bottom, there is a "Zip export file" dropdown menu showing "Yes". Finally, there are "OK" and "Cancel" buttons at the bottom right of the dialog.

5. In **Zip export file**, select an option from the dropdown to specify if the output file should be delivered as an email attachment in a .ZIP (compressed) file format.
 - **Yes** zips the exported file.
 - **No** does not zip the exported file.
6. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

FTP Delivery Method

The FTP delivery method allows an exported file to be delivered to an external system and is also available in Export Manager.

The preferred method for FTP delivery is the SFTP secure delivery method. For information on the OIEP SFTP delivery method, see **SFTP Delivery Method**.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

Prior to configuration, clicking the **HostName** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **FTPDeliveryHostName** property. The required format of the property is (square brackets not included):

```
SFTPDeliveryHostName=1=[host1],2=[host2]
```

Using the host name shown in the image below, the entry in the properties file would be:

```
FTPDeliveryHostName=1=FTPDeliveryHostServerName
```

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. In **Select Delivery Method**, choose **FTP**.

2. In **Host name**, select the host name or IP address of the FTP server to be used for the delivery.
3. In **Proxy host name**, add the host name to be used for the server proxy. This field is optional.

4. In **User name**, enter the user name that has access to log on to the FTP server.
5. In **Password**, enter the password that will be used to log on to the FTP server.
6. In **File name template**, to create the file name used for the output, enter template variables, static text, or use a combination. The default template is \$filename-\$timestamp.\$extension and creates a file name like 'GenericXML-1490813266535.xml' based on the settings of the OIEP. Each variable is described below:

- **\$filename** This variable is replaced with text to indicate the format of the delivered file as specified in the Output Templates flipper. For example, the output file name would include the text 'excel' or 'GenericXML' for those formats, or the Event ID for event-based STEPXML. Using the default file name template, a comma-separated value file would be named 'csv-1490808860412.csv' and 'GenericXML-1490810861593.xml' would indicate that Generic XML was used. For select object OIEPs, \$filename is replaced with 'exported.'
- **\$timestamp** This variable is replaced with the time in milliseconds between January 1, 1970, and the time when the file is uploaded, which allows for a unique identifier. For example, 1479230247017 is used for an export on 15 NOV 2016.

Alternatively, the following options can be used to generate a readable date and can include static characters to separate the timestamp elements. Individual elements can be ordered as required, as shown below.

\$timestamp(ddMMyyyy) displays day, month, and year and is output '15112016' for 15 NOV 2016.

\$timestamp(yyyy-MM-dd_HH.mm.ss) displays year, month, day, hour, minutes, and seconds and is output as '2016-11-15_09.32.43' for 2016 NOV 15 at 9:32.43 AM.

- **\$extension** This variable is replaced with the extension of the output file based on the selected format in the Output Templates flipper. For XML-based formats, the output file is set with the .XML extension, while Excel files use .XLS, and comma-separated values use .CSV.

Note: The File Name Template does not support conversions of file formats and can only be used to deliver files in the format specified on the Configuration tab > Output Templates flipper of the outbound integration endpoint.

7. In **Zip before upload**, select an option from the dropdown to specify if the output file should be delivered in a .ZIP (compressed) file format.
 - **Yes** uses the File Name Template and the extension .ZIP. When a file is zipped, the File Name Template is used for the .ZIP file. The \$filename template is replaced with 'result_0', and the selected file type is compressed with the name 'exported'. For example, a zipped STEPXML output with the default File Name Template exported on 15 NOV 2016 results in an output .ZIP file named 'result_0-1479230247017.zip'.
 - **No** uses the File Name Template for the file name along with the appropriate extension for the selected data format.
8. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

IBM Websphere MQ SSL Delivery Method

For information on connecting to IBM Websphere MQ in a non-SSL way, see **JMS Delivery Method**. This delivery option is only available in OIEPs.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. Prior to configuration, clicking the **Connection URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLProviderURL** property. Use the `[host]:[port]/[channel]` format as defined below:
 - `[host]` = hostname or IP of the Websphere MQ server
 - `[port]` = port number for the channel
 - `[channel]` = name of the channel
2. Prior to configuration, clicking the **Queue Manager** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLQueueManager** property.
3. Prior to configuration, clicking the **Queue Name** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLQueue** property.
4. Prior to configuration, clicking the **Key Store** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLKeyStoreLocation** property.

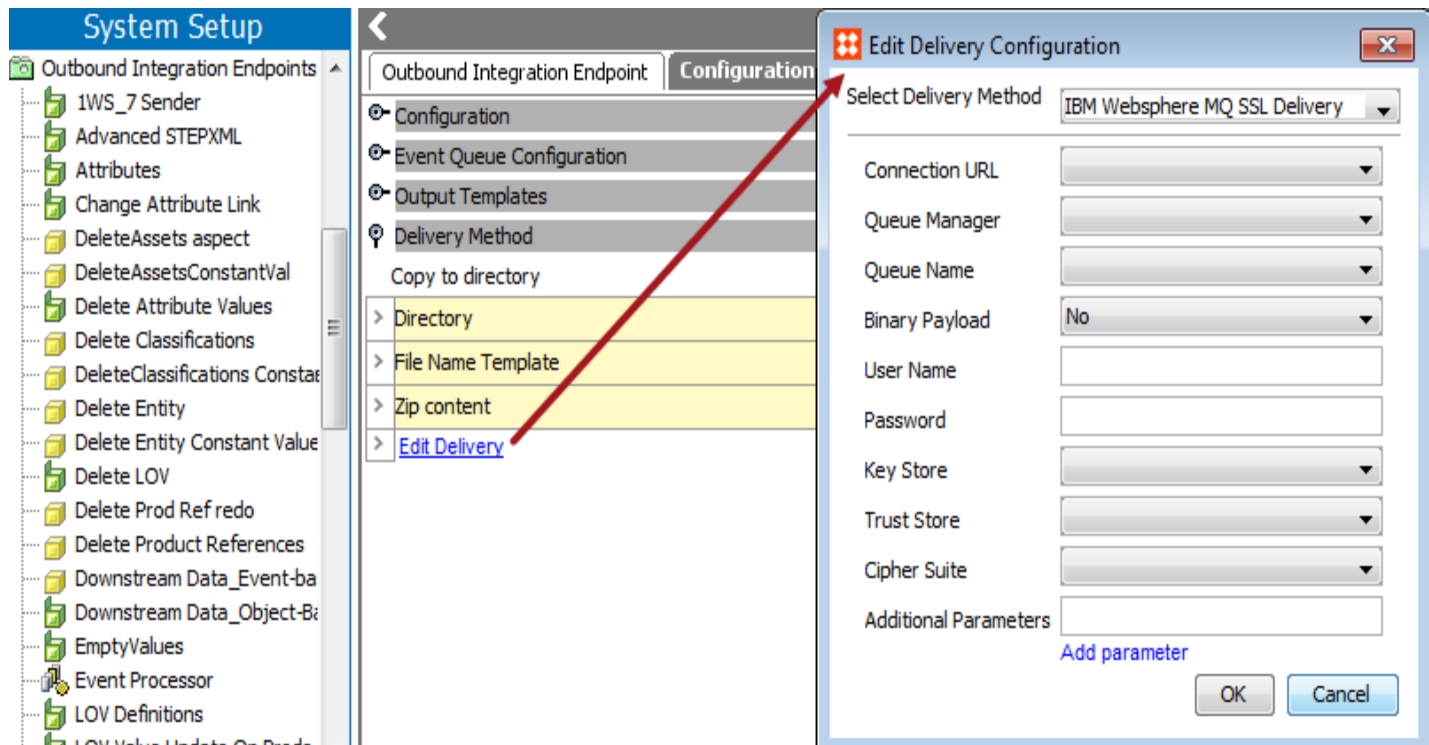
 Use the `jks` format with the personal certificate for the Queue Manager. To generate this, consult the manual for IBM Websphere MQ. The password for the key store must be configured in `sharedconfig.properties` file using the **WSMQSSLKeyStorePassword** property.
5. Prior to configuration, clicking the **Trust Store** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLTrustStoreLocation** property.

 This can be the same file as key store. To generate this, consult the manual for IBM Websphere MQ. The password for the trust store must be configured in `sharedconfig.properties` or `config.properties` using **WSMQSSLTrustStorePassword**.
6. Prior to configuration, clicking the **Cipher Suite** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLCipher Suite** property. STEP is running on non-IBM JRE, so this must be the same value as configured in the Queue Manager.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. In the **Select Delivery Method** field, choose **IBM Websphere MQ SSL Delivery**.



2. In **Connection URL**, select the URL for the connection.
3. In **Queue Manager**, select the name of the Queue Manager.
4. In **Queue Name**, select the name of the Queue for the connection.
5. In **Binary Payload**, select 'Yes' if the message contents will be in a binary format (such as Excel).
6. In **User Name**, if required, enter the user name that will be used to log onto the JMS provider.
7. In **Password**, if required, enter the password that will be used to log onto the JMS provider.
8. In **Key Store**, select the keystore in jks format, with the personal certificate for the Queue Manager.
9. In **Trust Store**, select the trust store with the CA for the Queue Manager.
10. In **Cipher Suite**, set to the same value as SSL CipherSuite in Websphere MQ.
11. If **Additional Parameters** are required, click Add parameter, then enter the Key and Value. For possible keys and values, consult the manual for IBM Websphere MQ.
12. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

JDBC Delivery Method

The JDBC delivery method allows STEP data to be delivered directly to tables in Relational database management systems (RDBMS) like Oracle, MySQL, MS SQL Server, and PostgreSQL. Though the ways the JDBC plugin can be deployed are various, one of its prime uses is to send STEP data to data analytics tools for the purpose of displaying STEP data in a data analytics dashboard. This delivery method requires CSV format and is available in both the Export Manager and in Outbound Integration Endpoint (OIEP).

For more information on the Export Manager option, see **JDBC Delivery Method** in the **Export Manager** documentation.

Prerequisites

Important: For complete setup requirements, see the **Exporting Data via JDBC with CSV Format** topic in the **Data Format** documentation.

Install the required drivers

JDBC specification 4.1-compliant drivers should be placed in a directory accessible from all application servers. These drivers can then be made available for the delivery plugin via the dynamic properties `JDBCDeliveryPlugin.DriverPath.[n]` and `JDBCDeliveryPlugin.DriverClass.[n]`. For more information regarding applicable Java drivers, review the RDBMS vendor's homepage on the web.

Configure data for the dropdown parameters

Changes to the properties file are implemented when the server is restarted.

1. Prior to configuration, clicking the **Driver Location** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **JDBCDeliveryPlugin.DriverPath.[n]** property. As an example:

```
JDBCDeliveryPlugin.DriverPath.1 = L:/shared/mysql-connector-java-5.1.42-bin.jar
```

In this example, the drivers are stored on the application server's L:/shared drive.

2. Prior to configuration, clicking the **Driver Class** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **JDBCDeliveryPlugin.DriverClass.[n]** property. As an example:

```
JDBCDeliveryPlugin.DriverClass.1 = com.mysql.jdbc.Driver
```

In this example, the drivers class used is 'com.mysql.jdbc.Driver.'

3. Prior to configuration, clicking the **Database URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **JDBCDeliveryPlugin.URL.[n]** property.

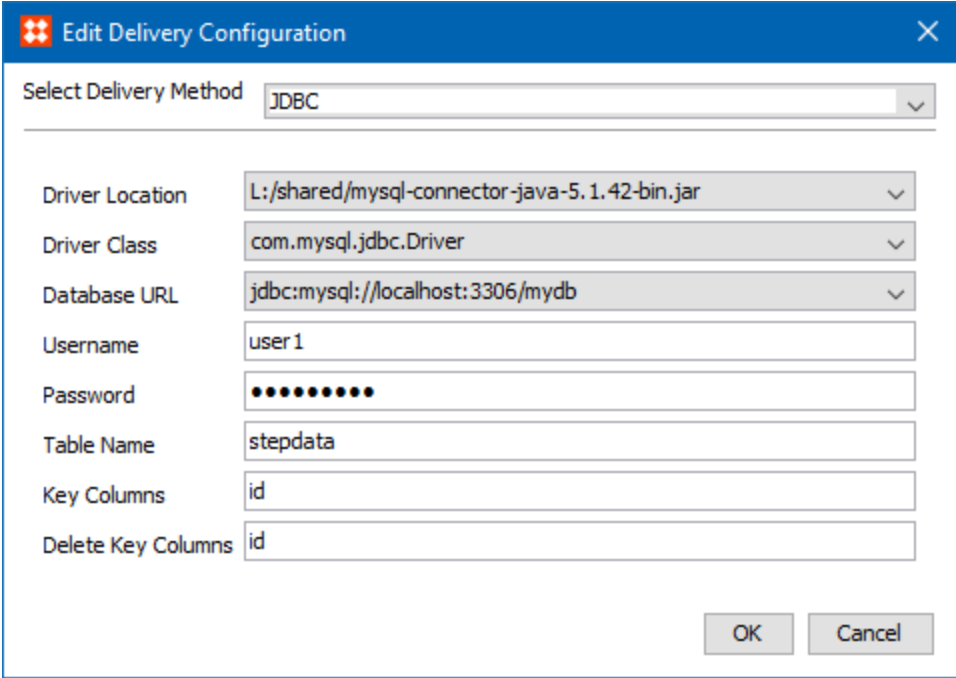
```
JDBCDeliveryPlugin.URL.1 = jdbc:mysql://localhost:3306/mydb
```

In this example, the database URL used is 'jdbc:mysql://localhost:3306/mydb.'

Configuration

For information on a parameter, hover over the parameter field to display help text.

In the OIEP editor on the **Configuration** tab, navigate to the **Delivery Method** flipper, then click **Edit Delivery**.



1. In **Select Delivery Method**, choose **JDBC** from the dropdown.
2. In **Driver Location**, select from the dropdown one of the paths to the relevant .jar file.
3. In **Driver Class**, select from the dropdown one of the pre-configured driver class.
4. In **Database URL**, select from the dropdown one of the pre-configured URLs to the destination database.
5. In **Username**, enter the username required to access the destination database.
6. In **Password**, enter the password required to access the destination database.
7. In **Table Name**, enter the name of the table in the destination database to which STEP will be publishing data
8. In **Key Columns**, list the names of the columns appearing on both the exported CSV file and the table in the destination database, separated by commas (and no spaces), into which STEP will publish data
9. In **Delete Key Columns**, list the names of the columns appearing on both the exported CSV file and the table in the destination database, separated by commas (and no spaces), from which STEP will delete data. The headers contained in this field can differ from the headers in the 'Key Columns' field, but they must also be part of the upsert key definition.
10. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

JMS Delivery Method

The available options for the Java Message Service (JMS) delivery method are system dependent. STEP has three standard JMS delivery plugins available: Apache Active MQ, Websphere MQ, and Oracle AQ. Each of these can be used to deliver messages to a JMS receiver without any further customizations. To deliver messages to other JMS providers, contact your Stibo Systems account manager for further customizations. This delivery option is only available in OIEPs.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

See the **Sharedconfig.properties File Examples** section below for more information.

1. Prior to configuration, clicking the **Connection Factory Name** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JMSConnectionFactoryName** property.
2. Prior to configuration, clicking the **JMS Provider URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JMSProviderURL** property. If necessary, use a comma to separate multiple URLs.

The following is an example of a complete property entry for two systems named 'qa' and 'stage,' as well as their URLs 'http://step-qa' and 'http://step-stage':

```
JMSProviderURL=qa=http://step-qa,stage=http://step-stage
```

3. Prior to configuration, clicking the **JMS Queue** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JMSQueue** property.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. On the **Configuration** tab, in the **Delivery Method** area, click **Edit Delivery**.
2. In **Select Delivery Method**, choose **JMS Delivery**.

3. In **JMS Connection Factory Class**, choose the JMS connection factory class that corresponds to the driver class from the JMS provider vendor. On a standard system, the following options are available:

- **ActiveMQInitialConnectionFactory**: Connects the JMS delivery to Apache Active MQ. For information, search the web.

Note: WMQInitialContextFactory is no longer supported, use FileInitialContextFactory instead.

- **FileInitialContextFactory**: Enables setting up JMS Websphere Inbound and Outbound Integration Endpoints which reference a binding file (created from JMS Websphere Client software). The binding file is a configuration file which includes all details of how STEP should interact with JMS Websphere. For information about JMS Websphere Client Software, search the web.

4. In **Connection Factory Name**, select a connection factory name from the list.

5. In **JMS Provider URL**, select a JMS Provider URL from the list.

6. In **JMS Queue**, select the physical name of the JMS Queue to be used on Apache Active MQ or Websphere MQ.

7. In **Binary Payload**, select 'Yes' if the message contents will be in a binary format (such as Excel).

8. In **User Name**, if required, enter the user name that will be used to log onto the JMS provider.

9. In **Password**, if required, enter the password that will be used to log onto the JMS provider.

10. If additional parameters are required, click **Add parameter**, then enter the Key and Value.

11. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Sharedconfig.properties File Examples

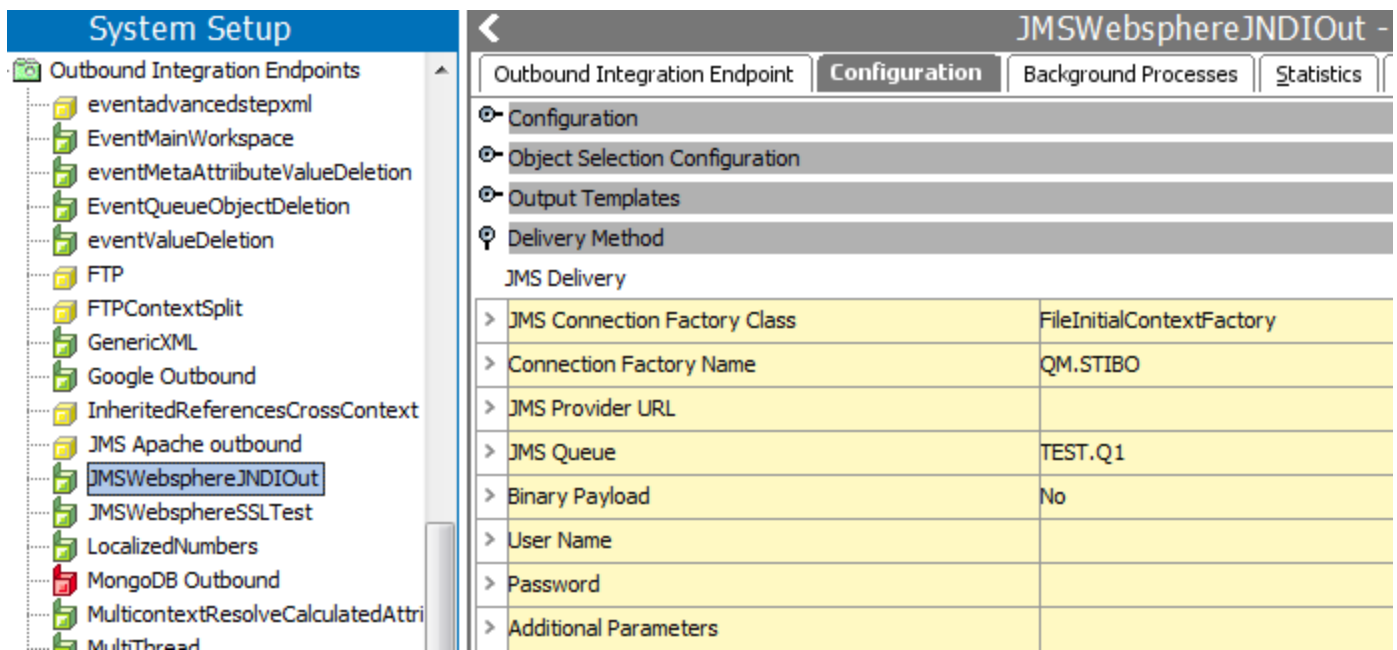
Each of these properties has a key/value pair as a value:

```
JMSConnectionFactoryName=FirstFactoryName=MyFactoryName
JMSProviderURL=ProviderURL=https://myproviderurl.com/url
JMSQueue=Queue=MyqueueName
```

Each of the properties can contain more than one value in a comma separated list:

```
JMSConnectionFactoryName=FirstFactoryName=MyFactoryName1,SecondFactoryName=MyFactoryName2
JMSProviderURL=FirstProviderURL=https://myproviderurl.com/url,SecondProviderURL=https://myproviderurl2.com/url
JMSQueue=Queue1=FirstqueueName,Queue2=SecondqueueName2
```

FileInitialContextFactory JMS Delivery Configuration Example



The screenshot shows the 'System Setup' window with a tree view on the left and a configuration panel on the right. The tree view includes 'Outbound Integration Endpoints' and various endpoints like 'eventadvancedstepxml', 'EventMainWorkspace', etc. The 'JMSWebSphereJNDIOut' endpoint is selected. The configuration panel on the right has tabs for 'Outbound Integration Endpoint', 'Configuration', 'Background Processes', and 'Statistics'. The 'Configuration' tab is active, showing a list of configuration items: 'Configuration', 'Object Selection Configuration', 'Output Templates', and 'Delivery Method'. The 'Delivery Method' is expanded, showing 'JMS Delivery' settings in a table.

JMS Delivery	
> JMS Connection Factory Class	FileInitialContextFactory
> Connection Factory Name	QM.STIBO
> JMS Provider URL	
> JMS Queue	TEST.Q1
> Binary Payload	No
> User Name	
> Password	
> Additional Parameters	

JMS Websphere Delivery Using SSL Configuration Example

Delivery Method

IBM Websphere MQ SSL Delivery

> Connection URL	webspheremq-qa.stibo.com:1417/STEP.SVRCONN
> Queue Manager	QM.STIBO_SSL
> Queue Name	TEST.Q1
> Binary Payload	Yes
> User Name	
> Password	
> Key Store	file:/workarea/JMSWebsphereSSLKeystore/keyStore.jks
> Trust Store	file:/workarea/JMSWebsphereSSLKeystore/keyStore.jks
> Cipher Suite	TLS_RSA_WITH_AES_256_CBC_SHA256
> Additional Parameters	

Apache Active MQ JMS Delivery

Delivery Method

JMS Delivery

> JMS Connection Factory Class	ActiveMQInitialContextFactory
> Connection Factory Name	ConnectionFactory
> JMS Provider URL	tcp://ATTCM3S9:61616
> JMS Queue	testqueue
> Binary Payload	No
> User Name	
> Password	
> Additional Parameters	
> Edit Delivery	

Mongo Delivery Method

Prior to selecting the Mongo Delivery Method, ensure your setup meets the prerequisites explained in **Prerequisites for Configuring the MongoDB Adapter** documentation. This delivery option is only available in OIEPs.

For more information, see <https://docs.mongodb.com/manual/> or the **MongoDB Adapter Setup Quick Guides** documentation.

For more information on the Mongo Delivery Method, see the **Mongo Delivery Method Elements** documentation.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. If your system displays radio button options for the server selection, see the **Legacy Configuration** section below for more details.
2. Prior to configuration, clicking the **Server(s)** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive dynamic **MongoDB.Server.Configuration.[n]** property, where the [n] is used for differentiation. The first configuration entry must use number 1, and each additional entry must be numbered sequentially (MongoDB.Server.Configuration.1, MongoDB.Server.Configuration.2, MongoDB.Server.Configuration.3, etc.). Each host is separated from the respective port using a colon (:), and additional servers are added using a comma-separated list. For example:

```
MongoDB.Server.Configuration.1=host4:40004
```

indicates an independent MongoDB instance. A second and third configuration could be supplied for servers being used in a cluster using the format:

```
MongoDB.Server.Configuration.2=host1:10001,host2:20002,host3:30003
```

```
MongoDB.Server.Configuration.3=host4:40004,host5:50005,host6:60006,host7:70007
```

3. The **Key store location** is only required if the Mongo DB requires the clients to use SSL to encrypt connections.

The password for the key store is configured in sharedconfig.properties on the STEP application server, using the case-sensitive **MongoDB.SSLKeyStorePassword** property. For example:

```
MongoDB.SSLKeyStorePassword = keystore_password
```

4. In **Trust store location**, is only required if the Mongo DB requires the clients to use SSL to encrypt connections.

The password for the key store is configured in sharedconfig.properties on the STEP application server, using the case-sensitive **MongoDB.SSLTrustStorePassword** property. For example:

```
MongoDB.SSLTrustStorePassword = truststore_password
```

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. In **Select Delivery Method**, choose **Mongo delivery**.

Edit Delivery Configuration

Select Delivery Method: **Mongo delivery**

Server(s): **host1:10001,host2:20002,host3:30003**

Raw DB prefix: **MongoDB**

User name: **userauth**

Password: **.....**

Replace Dollar sign:

Replace Dot sign:

☐ Use SSL

Key store location:

Trust store location:

Actions: **Workflow Review**

[Add action](#)

OK **Cancel**

2. In **Server(s)** dropdown, select the desired MongoDB server configuration, either an independent MongoDB instance or a MongoDB cluster setup.
3. In **Raw DB prefix**, if required, enter a prefix for all raw databases the adapter creates. The output is [prefix] + [context], resulting in a database name of MongoDBContext1 using the data in the image above, and exporting for Context1. Optional.
4. In **User name**, if MongoDB is configured to use authentication, enter the MongoDB user. Optional.
5. In **Password**, if MongoDB is configured to use authentication, enter the MongoDB password. Optional.
6. In **Replace Dollar sign**, text entered replaces any dollar sign (\$) during delivery. If blank, all dollar signs are replaced automatically by the JSON converter with Unicode 'uFF04'. This allows overriding the default Unicode replacement with the specified character(s), for example, an underscore (_).
7. In **Replace Dot sign**, text entered replaces any dot sign (.) during delivery. If blank, all periods are replaced automatically by the JSON converter with Unicode 'uFF0E'. This allows overriding the default Unicode replacement with the specified character(s), for example, an underscore (_).
8. Check **Use SSL** if MongoDB requires the client to use SSL encryption. Optional. If no further parameters are configured, it is assumed that the MongoDB uses a SSL certificate that is issued by a trusted certificate authority (like Verisign or Thawte). If this is not the case, or if more security of the connection is requested, you can configure an SSL key store and an SSL trust store.

- In **Key store location**, enter a key store location, for example, file:///path/key_store.jks. The Key store holds the STEP Mongo Adapter private key and SSL certificate in a keystore file as generated by the Java utility 'keytool' (in jks format). The password for the key store is configured as defined in the **Prerequisites** section.
 - In **Trust store location**, enter a trust store location, for example, file:///path/trust_store.jks. The trust store holds the Mongo DB public key and certificate in a keystore file as generated by the Java utility 'keytool' (in jks format). The trust store configuration is only needed if the certificate is self-signed and not issued by a trusted certificate authority. The password for the key store is configured as defined in the **Prerequisites** section.
9. In **Actions**, click the **Add action** link to enter one or more STEP business actions used to create and populate aggregated collections. These actions are run after the raw collections have been populated.
 10. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Legacy Configuration

When the sharedconfig.properties file includes the MongoDB.Server property and the MongoDB.Port property, the dialog displays both a cluster option and a legacy option. Typically this happens when a STEP system had a MongoDB host / port configured prior to the update that introduced the clustered configuration functionality.

Note: Upgrading the sharedconfig.properties file to modify all configurations using the legacy properties to the cluster enabled property, and then removing the legacy properties completely, results in the radio button options being removed from the dialog, as shown in the image at the top of this topic.

When the **Cluster Enabled Configuration** radio button is selected, the **Server(s)** dropdown displays entries for the dynamic MongoDB.Server.Configuration.[n] property.

When the **Legacy Configuration** radio button is selected, the **Server** and **Port** dropdowns display entries for the MongoDB.Server and MongoDB.Port properties.

Multiple independent servers or ports can be listed using the following format:

```
MongoDB.Server=1=vps124887.vps.ovh.ca,2=cde,3=rty
```

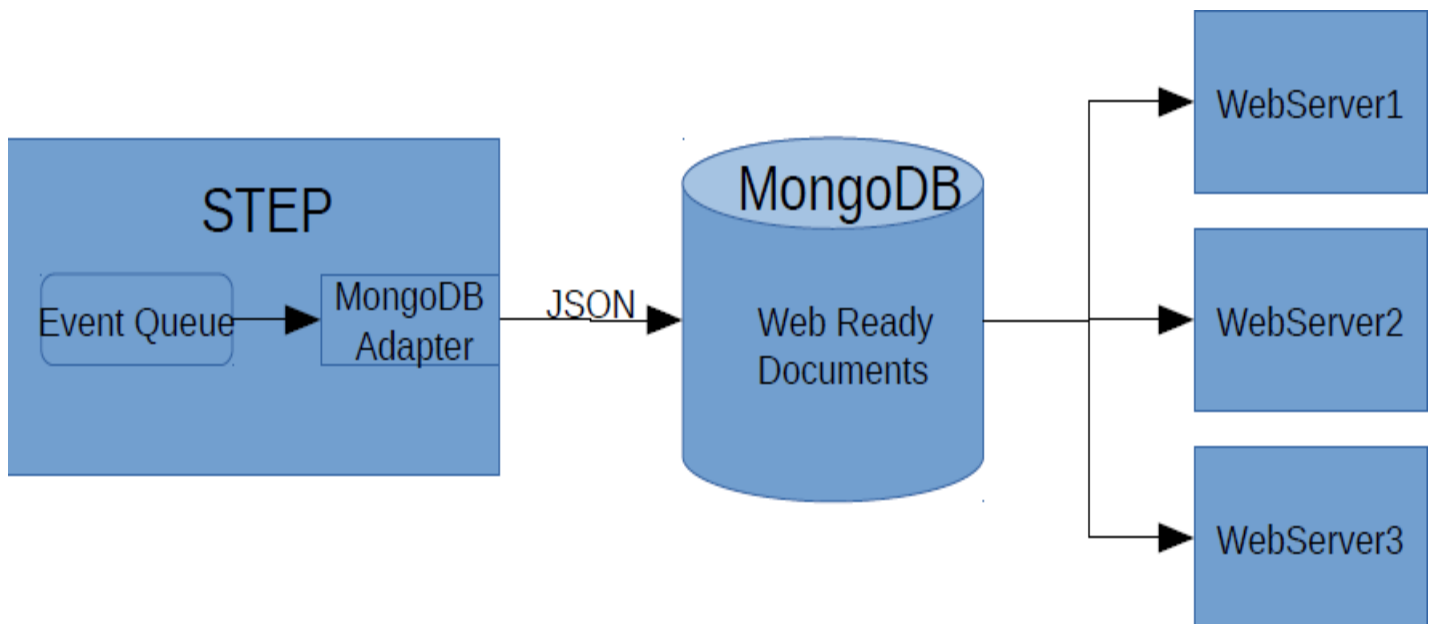
```
MongoDB.Port=1=30001,2=30002,3=30003
```

Mongo Delivery Method Elements

The STEP Mongo Adapter is an outbound integration endpoint delivery plugin that receives data from a STEP event queue and loads it into a MongoDB database. The MongoDB is often used for web site back-end, reporting, and high performance feeds to other back-end systems.

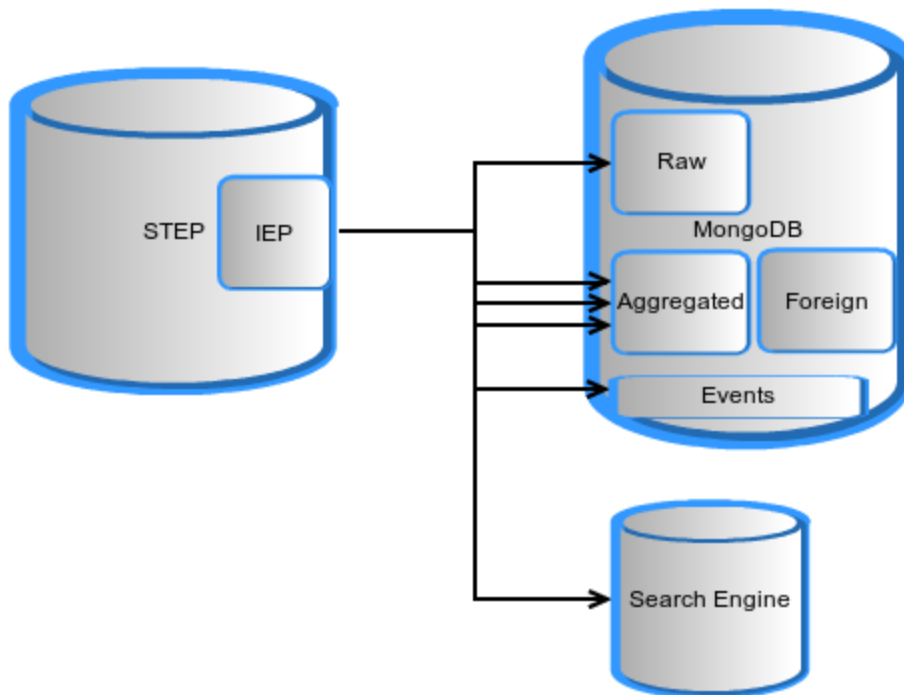
The Mongo adapter converts STEPXML to JSON and keeps a set of 'raw' collections in MongoDB that are in sync with STEP. Additionally, it allows you to add JavaScript triggers to maintain aggregated collections.

The following illustration shows how the Mongo Adapter can be used to feed a Mongo database that is used as the back-end to a website:



By keeping collections of web-ready JSON documents available in MongoDB, the JSON can be passed directly to the browser's JavaScript. This results in the web server having little processing to do, which means that it simply needs to pass the right collection result directly back to the browser.

The following drawing illustrates the collections involved:



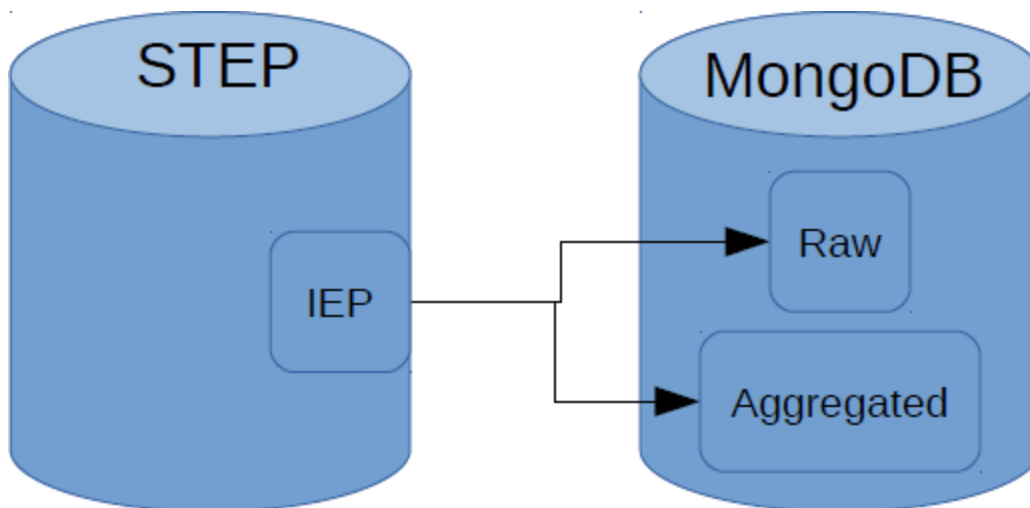
The STEP Mongo Adapter ensures that the raw collections are a one-to-one mirror of data in STEP. The adapter does not embed nodes found through references or child associations. These nodes are stored separately in the raw collections.

It is possible to configure the adapter so that it creates one or more customized, aggregated MongoDBs (databases or collections) where the documents can be tailored to meet the exact usage requirements, like reporting, analysis, and so on.

The JavaScript triggers allow maintaining aggregated collections, for example:

- List of children, including title and other data necessary in a tree-like view.
- Information from the target end of references that is needed in a given view.
- Mapping to a JSON structure that is closer to the web site data model.

The following image shows a MongoDB database containing a raw database that is maintained directly by the STEP Mongo Adapter and an aggregated database maintained by JavaScript triggers.



A JavaScript trigger is executed under certain conditions, for example, only for a specific object type. This reduces the amount of checking necessary in the JavaScript code. Once the raw collection is updated and the JavaScript triggers (if configured) have completed, an event is inserted into the events collection. Also, an update is sent to the search engine.

Note: Aggregated collections and raw collections do not need to be in the same MongoDB instance. However, if required, different collections in the aggregate can be combined into their own MongoDB instance.

Default Databases and Collections

The STEP Mongo Adapter maintains a number of MongoDBs, referred to as 'raw databases', and a number of collections in the raw databases, referred to as 'raw collections'. A raw database is created for each context that is exported and the database is named after the context. For example, if the two contexts EN and FR are exported, two raw databases are created, one named EN and one named FR.

It is possible to configure a string that prepend the raw database name when the STEP Mongo Adapter is configured. When data is exported from multiple contexts, a JSON document is created per context, per exported object. Continuing the previous example, two JSON documents are created for each object: one for the object in the EN context and one for the object in the FR context.

A raw collection is only created when the particular STEP data type is exported. So, if no assets are exported from STEP, no asset collection is created. Each raw database contains all or some of the following raw collections:

Raw Collection	Purpose
product	Exported products
asset	Exported assets
classification	Exported classifications

Raw Collection	Purpose
entity	Exported entities
attribute	Exported attribute definitions
referenceType	Exported reference types
listofvalues	Exported list of values (LOVs)
unit	Exported unit definitions

Inherited values, calculated values, and unit names on values are handled already in the raw collections. Since changes to these affect replicated information, STEP ensures that appropriate updates are made to sync the replicated information.

STEP JSON and Mongo JSON

The STEP Mongo Adapter converts STEPXML input to STEP JSON (JavaScript Object Notation) documents, and then stores the JSON documents in the MongoDB. The STEP JSON schema can be downloaded from your STEP server at: [http://\[enter step-server\]/files/StepSchema.json](http://[enter step-server]/files/StepSchema.json)

Formally, MongoDB documents are BSON documents, which is a binary representation of a JSON document. For more information, see the MongoDB documentation on the web.

For an expanded example of the basic conversion using STEP Mongo Adapter, see the **Mongo Delivery Method Conversion Example** documentation.

JavaScript Triggers

STEP business action triggers with JavaScript populate the aggregated MongoDB databases and collections. You can configure one or more business action triggers for the STEP Mongo Adapter. The business action triggers are fired after the STEP Mongo Adapter has populated the raw collections.

During configuration of the STEP Mongo Adapter, the configuration wizard allows you to select multiple JavaScript to run under different conditions by adding preconditions to the business actions. A precondition is, for example, a JavaScript trigger that is only fired for updates to a given type of objects. For more information, see the **Applies If Tab** section of the **Editing a Business Rule** documentation.

Due to the nature of inheritance and calculated attributes, the trigger may be called even if there are no changes to the underlying raw objects.

The business action triggers are executed once per JSON document sent to the MongoDB database. If two objects are exported from STEP, two JSON documents are sent to the MongoDB database and the business action triggers are executed twice. If the objects are exported in multiple contexts, one JSON document is

generated per context, per object. This means that if two products in two contexts are exported, the business action triggers are executed four times.

Business action triggers are executed on the STEP server. Business actions triggers need access to the MongoDB database to fetch data from the raw collections and write or read data to / from the aggregated collections. For the best performance, network latency between the STEP server and the MongoDB server should be minimal.

Important: Place aggregated collections in their own database so that the raw database contains only collections.

Bound Variables

The JavaScript trigger have accesses to information in the Mongo Adapter environment by binding script variables when the JavaScript trigger is defined.

Type	Gives access to
MongoDBContext	- The current STEP context. - The name of the raw MongoDB database. - The MongoDB database via the <code>com.mongodb.Mongo</code> object. For more information, see: http://api.mongodb.org/java/current/com/mongodb/Mongo.html. The actual Mongo object that is passed can be a wrapper, but it adheres to the interface
JSON context	The JSON object with the data that has been persisted in MongoDB.

For more information about executing JavaScript in business rules, see the **Business Action: Execute JavaScript** section of the **Business Rules** documentation.

Example JavaScript to Maintain a Collection in a Classification

The following code-snippet illustrates an example JavaScript that maintains a collection containing the list of products (ID and Name) in a classification.

The structure of the collection is:

```
{_id: "<classification-id>", products : { { id: "<product-id1>", name:
"<product-name1>"}, { id: "<product-id2>", name: "<product-name2>"}, ... }

function getClassificationIDs(classrefs) {
    var result = new Array();
    if (classrefs != null) {
        for (i = 0; i < classrefs.targets.size(); ++i) {
            result.push(new String(classrefs.targets[i].targetID));
        }
    }
}
```

```

    }
    return result;
}

var db = mongoContext.getMongo().getDB('extra');
var collection = db.getCollection('classificationproducts');

var classrefs = mongoData.references ? mongoData.references["Web
Classifications"] : null;

var classificationIDs = getClassificationIDs(classrefs);

var deletequery = {"products.id" : mongoData._id , "_id" : { $nin :
getClassificationIDs(classrefs) }};
var adelete = { $pull : { products : { id : mongoData._id } } };

collection.update(deletequery, adelete, false, true);

var query = {"products.id" : mongoData._id };
var name = mongoData.name;
var update = { $set : { "products.$.name" : name } };

collection.update(query, update, false, true);

classificationIDs.map(function(item) {
    var query = { _id : item };
    var update = { $addToSet : { products : { id : mongoData._id, name : name }
} };
    collection.update(query, update, true, false);
});

```


Prerequisites for Configuring the MongoDB Adapter

The following are prerequisites for setting up the MongoDB adapter:

1. A MongoDB instance with suitable amounts of disk space.
2. The STEP MongoDB Adapter uses a network port to access the MongoDB database on the server where the MongoDB is running. The network port must be open and accessible from the STEP servers.
3. Update `sharedconfig.properties` or `config.properties` with the **name of the MongoDB server**, using the `MongoDB.Server` configuration property and a list of number-to-value pairs, separated by commas, as follows:

```
MongoDB.Server = 1=MongoDBServer1, 2=MongoDBServer2, 3=MongoDBServer3
```

4. Update `sharedconfig.properties` or `config.properties` with the **MongoDB port** using the `MongoDB.Port` configuration property, and a list of number-to-value pairs, separated by commas, as follows:

```
MongoDB.Port = 1=10001, 2=2002, 3=3003
```

For example, to configure the adapter to use a MongoDB instance running on either 'Server1' or 'Server2' using port '27712' on both servers, the properties would be:

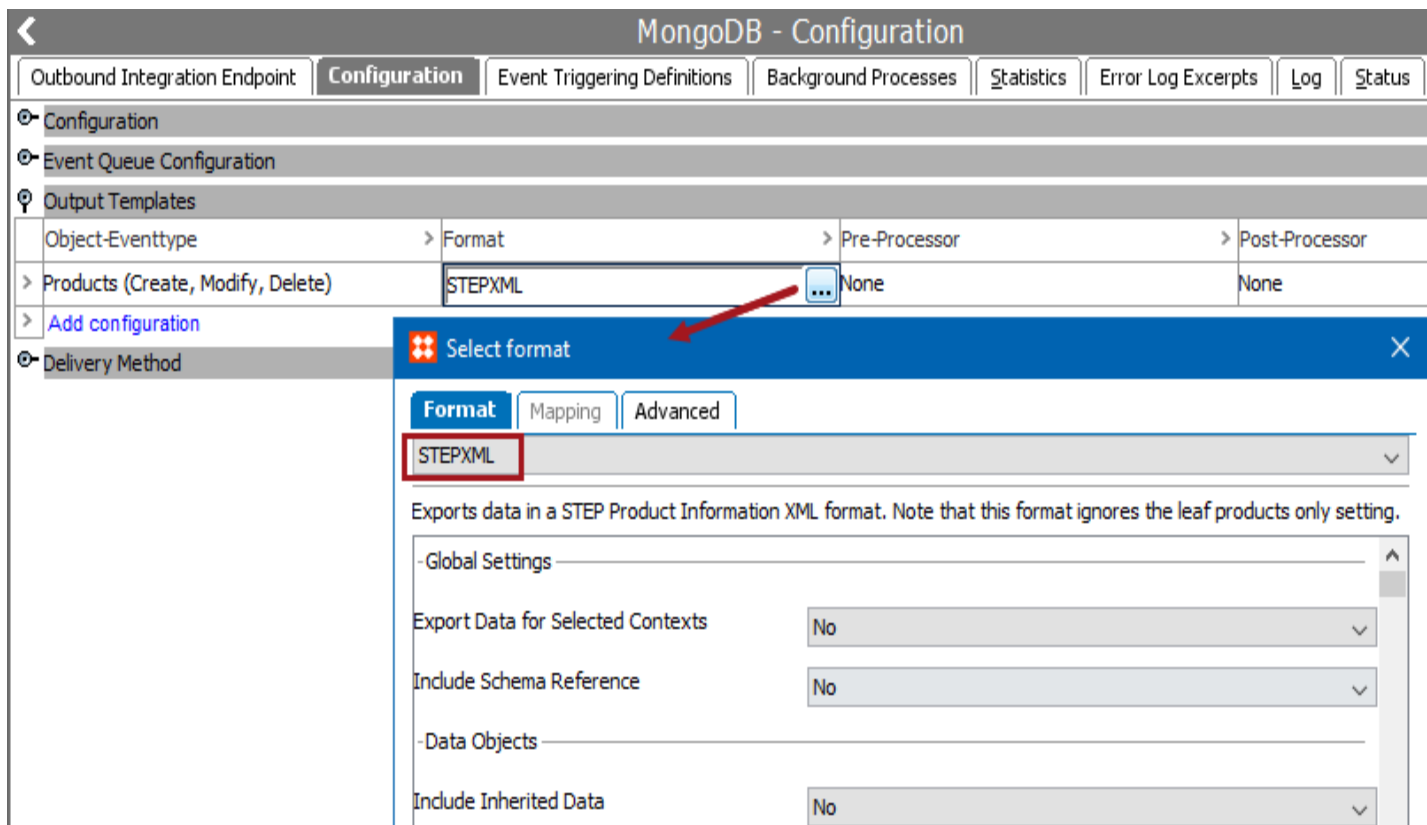
```
MongoDB.Server = 1=Server1, 2=Server2
```

```
MongoDB.Port = 1=27712
```

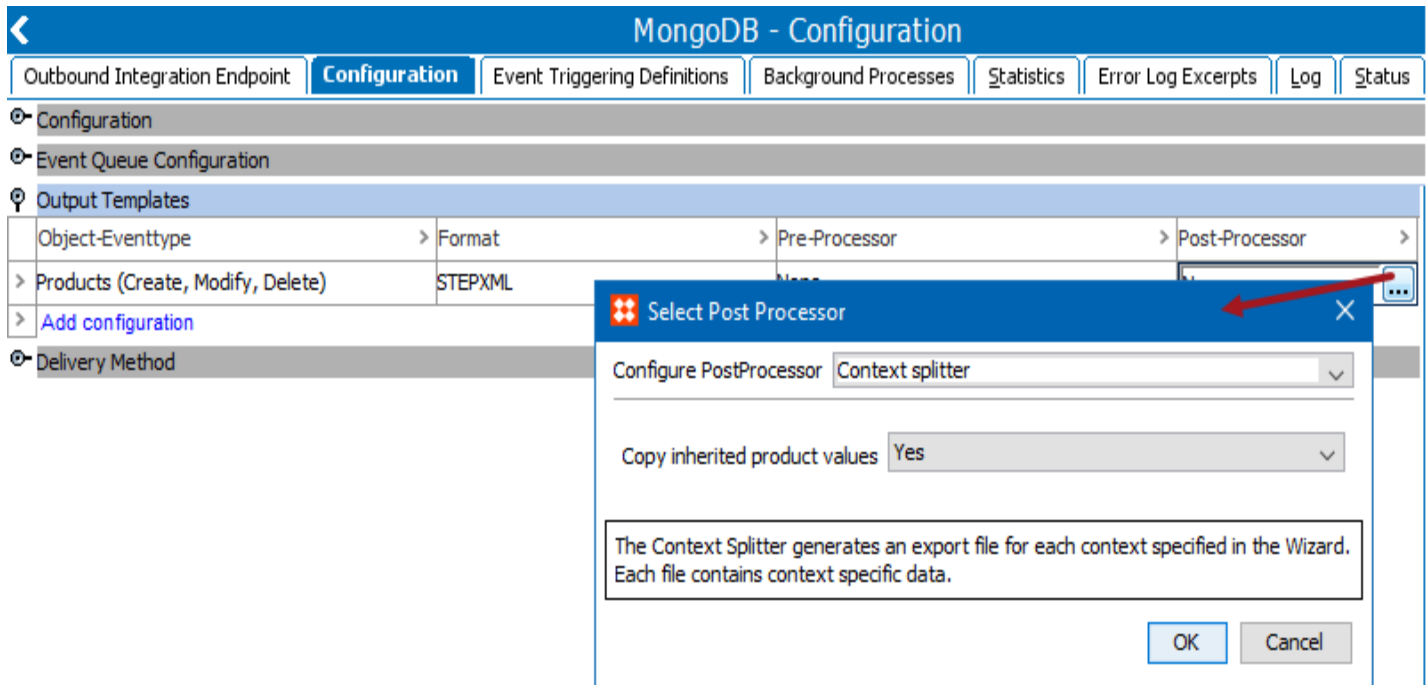
Configure the OIEP

The STEP MongoDB Adapter is run within an outbound integration endpoint (OIEP) and is configured as the delivery method for the endpoint.

1. In System Setup, create an OIEP, selecting STEP Exporter as the Process Engine. For details on creating a new OIEP, see the **Creating an Event-Based Outbound Integration Endpoint** topic.
2. For the new OIEP, open Configuration > Output Templates > Add configuration > Format > click the ellipsis button (...).



3. On the Select Format dialog, leave objects as default values except for the following parameters:
 - Select the **STEPXML** export format in the dropdown at the top of the dialog
 - Data Objects > **Flatten Hierarchies** = Yes
 - Data Objects > **Include Entities** = Selected
 - Data Objects > **Include Products** = Selected
 - Data Objects > **Include Classifications** = Selected
 - Data Objects > **Include Assets** = Selected
 - Configuration > **Include Attribute Groups** = Minimum
 - Configuration > **Include Link, Reference and Object Types** = Minimum
 - Configuration > **Include List of Values** = Minimum
 - Configuration > **Include Units** = Minimum
4. In the Post-Processor field, click the ellipsis button (...) and select **Context splitter**, then click OK.



- For Delivery Method, if MongoDB requires user authentication, configure a MongoDB user and password. The MongoDB user must be configured in the MongoDB admin database and have permissions to create new MongoDB databases (the MongoDB role "readWriteAnyDatabase").

Configuring the JavaScript Triggers

JavaScript triggers are fired for all updates to the raw collections. The purpose of the triggers is to maintain the aggregated collections. The JavaScript triggers are configured as business actions in STEP and then applied when configuring the MongoDB Adapter.

Continue with setting up the Mongo Delivery Method options, by following the **Mongo Delivery Method** documentation.

MongoDB Adapter Setup Quick Guides

The following information is available to assist in the setup of Mongo Delivery Method.

Configuring Mongo Authentication Quick Guide

The following quick guide describes how to configure a Mongo database to use authentication. Details may be found in the Mongo documentation on the web.

The guide assumes that the setup is used on a Mongo database installation running as a single server installation (i.e., the Mongo database is running as a standalone server).

1. Start the Mongo database without authentication.
2. Log onto the database using the Mongo client.
3. Create a system administrator user.

```
> use admin
switched to db admin
> db.createUser(
  {
    user: "admin",
    pwd: "password",
    roles:
    [
      {
        role: "userAdminAnyDatabase",
        db: "admin"
      }
    ]
  }
)
Successfully added user: {
  "user" : "admin",
  "roles" : [
    {
      "role" : "userAdminAnyDatabase",
      "db" : "admin"
    }
  ]
}
```

```
}
```

4. Create the user that will be used when the MongoDB adapter logs onto the Mongo database. This user should be created in the admin database with roles to read and write any databases in Mongo:

```
> use admin
switched to db admin
> db.createUser( {"user" : "stepsys", "pwd" : "stepsys", "roles" : [
"readWriteAnyDatabase" ] })
Successfully added user: { "user" : "stepsys", "roles" : [
"readWriteAnyDatabase" ] }
```

5. Stop the Mongo database and enable authentication. This may be done in the Mongo configuration file:

```
auth = true
```

or from the command line:

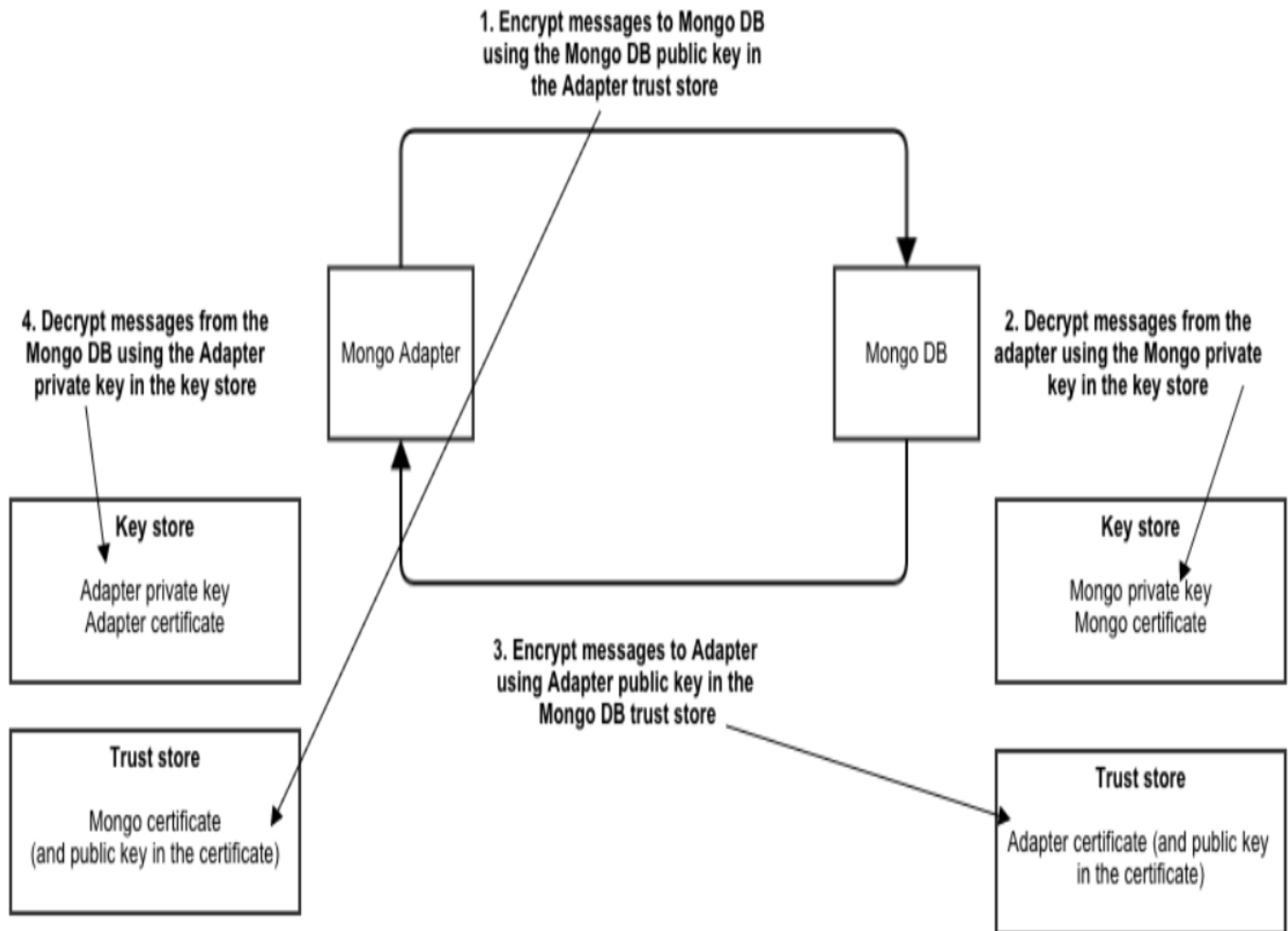
```
mongod --auth --config
```

SSL Configuration Quick Guide

When the Mongo DB adapter is configured for SSL, a key store and a trust store must be configured.

- The key store is a key store file in jks format containing the Mongo Adapter (STEP) private key and certificate.
- The trust store is a trust store file in jks format containing the Mongo DB public key and certificate.

Key store files may be created from the public and private key files and certificates using utilities like the java keytool utility. The diagram below illustrates the uses of the key store and trust store.



To configure SSL encryption on the STEP-to-Mongo database connection:

- Configure the Mongo database to use SSL: <http://docs.mongodb.org/manual/tutorial/configure-ssl/>
- Configure the Mongo client to connect to a Mongo database using SSL: <http://docs.mongodb.org/manual/tutorial/configure-ssl-clients/>

The Mongo database available as a free download does not support SSL. (SSL requires the Enterprise edition of Mongo.) Instead, download the appropriate source and compile your own version of the Mongo database from the web.: <https://docs.mongodb.com/manual/installation/>

Mongo Delivery Method Conversion Example

The following examples use this product with type = 'Item', ID 'XYZ', and Name = 'XYZ Name'. The product is located in the family 'FamilyXYZ2'.



ID: EXA-5002-1001

Name: Arm chair

Type: Chair

Parent ID: Chairs

The STEP Mongo Adapter transforms this to the STEP JSON document:

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
}
```

The id of the product is used as unique ObjectID in the MongoDB database, and therefore the STEPXML id field is transformed to _id. The UserTypeID XML attribute is converted to the objectTypeID field. The type field describes the kind of STEP object that is exported. For Basic STEP objects, the types are 'product', 'asset', 'classification', and 'entity'. Other STEP objects are described in the following topics.

Values

In this example, a product has two attributes:

- A single value attribute Color with the value 'Brown'
- A multi value attribute Height with the value = '43', the unit = 'inches', and the value = '120', with the unit = 'cm'.

When this is transformed to a STEP JSON document by the STEP MongoDB adapter, it results in the following:

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
  values : {
    Color : "Brown",
    Height : [ "43 inches", "120 cm"]
  }
}
```

The data displayed includes:

- Values of the product are stored as a subdocument with the name 'values'.
- The value of the single-valued attribute 'Color' is represented as a field-and-value pair: the ID of the attribute is the field, and the value is the value of the attribute represented as a string.
- The multi-valued attribute 'Height' is represented as a field-to-arrays of strings. Again, the ID of the attribute is the field, and the string array contains a string for each value.
- The name of the unit is appended to the attribute value with a space for all values with a unit. In the example above, '120 cm' is the concatenation of the value '120' and 'cm' from the unit named 'unece.unit.CM'.

Values of specification attributes that are inherited to the product are included in the JSON document of the product. For example, if the parent contains an attribute called Brand with the value 'Office chairs', the JSON document looks like the following:

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
  values : {
    color : "Brown",
    Height : [ "43 inches", "120 cm"],
    Brand: "Office chairs"
  }
}
```


LOV Value IDs

When values are included from an LOV that uses Value IDs (via the parameter 'Use Ids on values' = Yes on the LOV), the Value and the ValueID are stored as a subdocument with the name 'extValues'. For backward compatibility, the LOV value is also included in the 'value' subdocument, meaning this data is available twice, and can be extracted based on the need.

Note: If the LOV does not use Value IDs, the extValues subdocument is not written.

In this example, the LOV value of '4001' has a Value ID of '4001_CALYPSO'.

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
  values : {
    color : "Brown",
    Height : [ "43 inches", "120 cm"],
    Brand: "Office chairs"
    attribute_id: "4001",
  },
  extValues: {
    attribute_id: {
      value: "4001",
      valueID: "4001_CALYPSO"
    }
  }
}
```

The data displayed includes:

- The selected LOV value is stored within the subdocument with the name 'values' in the attribute_id field and again with the 'extValues' subdocument in the 'attribute_id: {value:}' field.
- The value ID of the selected LOV value is represented once with the 'extValues' subdocument in the 'attribute_id: {valueID:}' field.

References and Links

References and links are converted to STEP JSON in a subdocument called 'references'. This subdocument contains a field-and-value pair, where the field is the ID of the reference type, and the value is either a

subdocument that defines the target of a reference, or an array of subdocuments that each define the target of a reference.

If the reference can only reference one target for the same source, the value is a subdocument. If the reference can reference more targets for the same source, the value is an array.

In this example, the product uses a 'Primary Image' asset reference to reference the image for the product. The asset has the ID 'Image_EXA-5002-1001'. The STEP JSON then exports as:

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
  values : {
    color : "Brown",
    Height : [ "43 inches", "120 cm"],
    Brand: "Office chairs"
  }
  references : {
    Primary Image : { targetID : "Image-EXA-5002-1001" }
  }
}
```

To find the asset for the reference, it is necessary to find the reference type 'Primary Image', which is stored in the raw collection 'referenceType'. The target type of the reference comes from the reference type.

The JSON of the reference target is found by searching the raw collection given by the reference type of an object with 'targetID'.

In the example above, the target type of the 'Primary Image' reference type is 'asset'. By looking in the 'asset' raw collection, the asset given by Image-EXE-5002-1001 is found.

Adding the reference 'Secondary Image', which can reference more targets such as ('Image1' and 'Image2') to the product, results in the following STEP JSON:

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
```

```

values : {
  color : "Brown",
  Height : [ "43 inches", "120 cm"],
  Brand: "Office chairs"
}
references : {
  Primary Image : { targetID : "Image-EXA-5002-1001" },
  Secondary Image : [ { targetID : "Image1" }, { targetID : "Image2" } ]
}
}

```

If the reference contains a meta data attribute, the values of the meta data attribute are added as a subdocument to the reference subdocument that contains the value of the attribute:

```

{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
  values : {
    color : "Brown",
    Height : [ "43 inches", "120 cm"],
    Brand: "Office chairs"
  }
  references : {
    Primary Image : { targetID : "Image-EXA-5002-1001",
                     values: { ShowOnWeb : "true" },
                     },
    Secondary Image : [ { targetID : "Image1" }, { targetID : "Image2" } ]
  }
}

```

The reference type 'Primary Image' has a meta data attribute called 'ShowOnWeb'. The attribute is set to 'true' for the reference from the product to the asset 'PrimaryAsset'.

Classification to product links are special because they are owned by either the product or the classification. This will always be the same for a specific classification to product link type. The JSON document only contains the

classification-to-product links owned by the object. That is, the JSON document for a product only contains classification-to-product links owned by the product.

Attribute Types

Exporting an attribute type with the ID 'AttributeTypeID', and the name 'Name' linked in two parent attribute groups 'Parent1' and 'Parent2' results in the following JSON:

```
{
  _id : "AttributeTypeID",
  parentID : [ "Parent1", "Parent2"],
  name : "Name",
  listOfValuesID : "ListOfValuesID",
  validUnitIDs : ["unece.unit.MMT", "unece.unit.CMT"],
  type : "attribute"
}
```

Reference types are stored in the raw collection 'attribute'. To export reference types, verify that 'Include Attribute Definitions' is set to at least 'Selected' in the Process Engine configuration.

Reference Types

Exporting a product-to-product reference type with the ID 'ReferenceTypeID' and the name 'ReferenceName' generates the following JSON:

```
{
  _id : "ReferenceTypeID",
  name : "ReferenceName",
  type : "referenceType",
  targetType : "product"
}
```

In this example, 'targetType' is the target type of the reference. Possible target types are 'product', 'asset', 'classification', and 'entity'.

Reference types are stored in the raw collection 'referenceType'. When you export reference types, verify that Configuration > Include Link, Reference and Object Types is set to at least 'Minimum' in the Process Engine configuration.

Asset Push Locations

Asset Push Locations are stored in the assetPushLocations subdocument, in the root of the Asset documents in the asset collection. It provides the relative path to different versions of the asset. The assetPushLocations subdocument contains field-and-value pairs, where the field name is the assetpush configuration id, and the value is the relative path that the image file has been pushed to. The relative path is relative to the AssetPushClients root

folder, so to get the full path of the image, you have to prepend the path to the assetpush client root dir, as shown in the following example:

```
assetPushLocations : {
  AllAssets-approved : "AllAssets-approved/std.lang.all/73/15/7315.pdf",
  AllWebsiteImages-approved : "AllWebsiteImages-
approved/std.lang.all/73/15/7315.pdf",
  small : "small/std.lang.all/73/15/7315.jpg",
  large : "large/std.lang.all/73/15/7315.pdf"
}
```

Tables

Tables are defined on classifications or products. The table subdocument is stored in the root of the product and classification document, in the product and classification collection. The table subdocument contains field-and-value pairs, where the field name is the table Type name of the table on the product or classification.

A table is exported from STEP in a resolved and transformed state. That is, any transformations defined in STEP have already been applied to the table and the row and column layout has therefore been resolved.

Formatting is not applied to the table before it is exported. So, a web application is not bound by the STEP table formatting. However, if any formatting has been defined for the table in STEP, it is included in the export so that the website application can be configured to render the formatting on the website.

The table contains an array of column elements that contain the formatting meta data for each column and an array of rows. The row contains the formatting information for each row, plus an array of cells. Each cell has a number of formatting meta data attributes and a text field that contains the content of the table cell.

Table Formatting

Formatting meta data can be stored on table, column, row, and cell level, and is inherited downwards from table level to column level to row level to cell level.

If a column is specified to use an underlined text style, and a row is specified to be bold text style, the formatting information is accumulated so that the cell at the intersection of the row and column is rendered as both underlined and bold.

Formatting meta data defined at a lower level overrides meta data at a higher level. For example, if a table has a gray background color, and a column has a red background color, then cells in that row are rendered with a red background.

A simple way of formatting a table in HTML is to map the tableType, columnType, and rowType to element classes in HTML, and then use a CSS style sheet to define the table formatting, as shown in the following example.

```
tables : {
  Description Table : {
    columns :
```

```
[ { columnType : "Description" }, { columnType : "Description" }, {
columnType : "Description" } ],
rows : [
  { rowType : "Header",
    cells : [
      { ruleRight : "0.5 pt",
        text : "Header 1",
        verticalAlignment : "top",
        cellStoryDirection : "horizontal",
        backgroundColor : "Light Blue",
        column : "0",
        ruleBelow : "0.5 pt",
        ruleAbove : "0.5 pt",
        textStyle : "TableHeader-Description",
        ruleLeft : "0.5 pt" },
      { ruleRight : "0.5 pt",
        text : "Header 2",
        verticalAlignment : "top",
        cellStoryDirection : "horizontal",
        backgroundColor : "Light Blue",
        column : "1",
        ruleBelow : "0.5 pt",
        ruleAbove : "0.5 pt",
        textStyle : "TableHeader-Description",
        ruleLeft : "0.5 pt" } ] },
  { rowType : "Header",
    cells : [
      { ruleRight : "0.5 pt",
        text : "Value 1",
        verticalAlignment : "top",
        cellStoryDirection : "horizontal",
        backgroundColor : "Light Blue",
        column : "0",
        ruleBelow : "0.5 pt",
        ruleAbove : "0.5 pt",
```

```

        textStyle : "TableHeader-Description",
        ruleLeft : "0.5 pt" },
    { ruleRight : "0.5 pt",
      text : "Value 2",
      verticalAlignment : "top",
      cellStoryDirection : "horizontal",
      backgroundColor : "Light Blue",
      column : "1",
      ruleBelow : "0.5 pt",
      ruleAbove : "0.5 pt",
      textStyle : "TableHeader-Description",
      ruleLeft : "0.5 pt" }] }
  ] } }

```

Attribute Links

AttributeLinks define the validity of attributes for products linked into the product and classification hierarchies. The attributeLinks subdocument is stored at the root of the product and classification document structure. The attribute links can themselves contain a values subdocument that defines the meta data values attached to the attributeLink.

The attributeLinks subdocument contains field value pairs, where the field name is the attributeID, and the value is a subdocument that contains any meta data values related to the attribute link. The following is an example of an attributeLinks subdocument:

```

attributeLinks : { Voltage range : { }
                  Rotary address switches : { },
                  Default address : { },
                  Power consumption : { }
                  }

```

Unit

Exporting an unit with the ID 'unece.unit.CMT', and the name 'cm' results in the following JSON:

```

{
  "_id" : "unece.unit.CMT",
  "values" : {
    "4941" : "centimeter"
  },
  "conversionToBase" : {

```

```

    "factor" : "100",
    "unitID" : "unece.unit.MTR",
    "offset" : "0"
  },
  "name" : "cm",
  "type" : "unit"
}

```

The unit document contains the 'conversionToBase' object that can be used for converting values from a unit to its base unit. The 'conversionToBase' object of the above example shows how to convert values of the unit centimeters to meters.

Units are stored in the raw collection 'unit'. To export units, verify that Configuration > Include Units is set to at least 'Minimum' in the Process Engine configuration.

List Of Values

Exporting a ListOfValues object with the ID 'LOVID', and the name 'List of values' results in the following JSON:

```

{
  "_id" : "LOVID",
  "values" : {
    "6823" : "This is a metadata value of the lov"
  },
  "name" : "List of values",
  "validUnitIDs" : [
    "unece.unit.CMT",
    "unece.unit.MTR"
  ],
  "type" : "listofvalues"
}

```

List Of Values are stored in the raw collection 'listofvalues'. To export List Of Values, verify that 'Include List Of Values Definitions' is set to at least 'Minimum' in the Process Engine configuration.

Oracle AQ Delivery Method

This delivery option is only available in OIEPs.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. Prior to configuration, clicking the **Connection URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **OracleAQReceiverConnectionURL** property.
2. Prior to configuration, clicking the **Queue name** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **OracleAQReceiverQueueName** property.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. In the **Select Delivery Method** list, choose **Oracle AQ Delivery**.

The screenshot shows the 'Edit Delivery Configuration' dialog box. At the top, there's a title bar with a close button. Below it, a 'Select Delivery Method' dropdown menu is set to 'Oracle AQ Delivery'. Underneath, there are five configuration fields: 'Connection URL' (a dropdown menu with a tooltip that says 'configure the key 'OracleAQReceiverConnectionUrl' in config.properties'), 'Queue Name' (a dropdown menu with a tooltip that says 'configure the key 'OracleAQReceiverQueueName' in config.properties'), 'Binary Payload' (a dropdown menu with 'No' selected), 'User Name' (a text input field), and 'Password' (a text input field). At the bottom right, there are 'OK' and 'Cancel' buttons.

2. In the **Connection URL** list, select a URL that points to Oracle AQ.
3. In the **Queue name** list, select an Oracle AQ queue name.
4. For **Binary Payload**, select Yes or No. 'Yes' is selected for non-text files.
5. In the **User Name** field, enter the user name that will be used to log on to Oracle
6. In the **Password** field, enter the password that will be used to log on to Oracle.
7. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

REST Delivery Method

The REST Delivery Method delivers a call-back URL to the REST service and does not include actual STEP data. The data can be fetched from the call-back URL by the receiving REST service. This delivery option is only available in OIEPs.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

Prior to configuration, clicking the **URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **RESTDeliveryURL** property. If necessary, use a comma to separate multiple URLs.

This should be a URL to a REST PUT method, for example, http://server/rest_URL. A URL where the result of the endpoint can be fetched is returned in the PUT call.

The following is an example of a complete property entry for two systems named 'qa' and 'stage,' as well as their URLs 'http://step-qa' and 'http://step-stage':

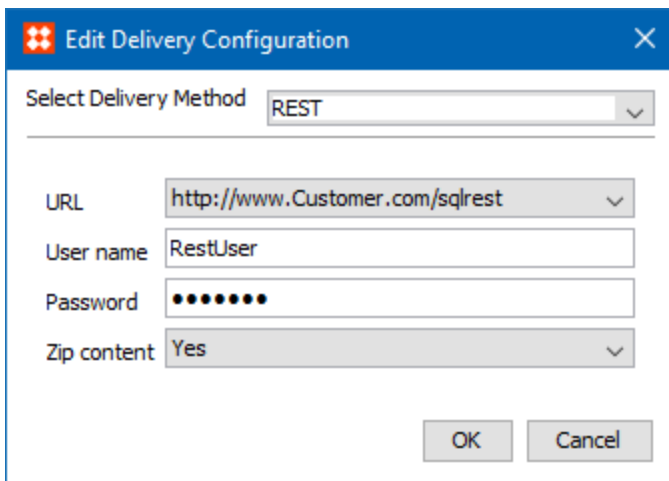
```
RESTDeliveryURL=qa=http://step-qa, stage=http://step-stage
```

Important: For information about how to use the REST API to upload files to REST, click the **STEP API Documentation** button on the STEP WebStart page.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. For **Select Delivery Method**, choose **REST**.



2. In **URL**, select the URL that points to the REST endpoint where you would like to receive the delivered data.
3. In **User Name**, enter the user name that will be used to log on to the REST endpoint.
4. In **Password**, enter the password that will be used to log on to the REST endpoint.

5. In **Zip content**, specify whether to zip the contents before upload.
6. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

REST Direct Delivery Method

The REST Direct delivery method differs from the standard REST delivery method in that the data is delivered directly to the REST service and no call-back URL is required.

This delivery option is only available in OIEPs and if the delivery fails, the OIEP will be disabled. There is no special resilience handling.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. The REST Direct delivery method reads the outgoing file into memory before sending. To handle the size of the outgoing file and prevent time-outs and rejections you must scale the heap size.
2. Prior to configuration, clicking the **URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **RestDirectDeliveryURL** property. If necessary, use a comma to separate multiple URLs. The image below shows the result of using the following example for a property entry involving two systems:

```
RestDirectDeliveryURL = 1=http://myfirstendpoint, 2=http://mysecondendpoint
```

URL	https://myfirstendpoint
HTTP Method	https://myfirstendpoint https://mysecondendpoint

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. For **Select Delivery Method**, choose **REST Direct**.

Edit Delivery Configuration

Select Delivery Method: REST Direct

URL: https://someendpoint.com

HTTP Method: POST

Query Parameters: version = 2
[Add parameter](#)

Headers: Content-Type = application/xml
[Add parameter](#)

Footer (Optional):

ZIP Content: No

Basic Authentication (Optional):

Username: someuser

Password: ••••••••

OK Cancel

2. In **URL**, select the URL that points to the REST endpoint where the delivered data should be received.
3. In the **HTTP Method** option, use the dropdown to select a POST, PUT, or PATCH method.

HTTP Method: POST

Query Parameters:

POST
PUT
PATCH

If required, in the **Query Parameters** field add any query parameters that are required for a successful REST transaction. You can click on the ellipsis (...) to add the 'Key' and 'Parameter' for the first parameter, or you can click on the **Add parameter** link to add the first and subsequent parameters.

- For **Headers**, click the **Add Parameter** link and add a key and a value. This is part of the HTTP network protocol.

- On the Add parameter dialog, click the **OK** button and the Header is displayed in the field. If multiple headers are needed, use the **Add Parameter** link to add each additional key and value pair.

Headers Content-Type = application/xml ... X
[Add parameter](#)

Note: Once a header is displayed, use the ellipsis button (...) to edit it or the X button to remove it.

For additional information about Headers, please see 'Additional Information About Headers' at the end of this section.

6. In **Footer**, add data required for the recipient to verify that the full message was received. This is part of the HTTP network protocol but not required.

The footer can be used to mark the end of a multi-part REST call (that is a REST call containing the payload split in more packages). The footer could also contain a checksum that the receiver can use to detect if the payload in a multi-part message has been changed by the middleware.

Headers	Content-Type = application/octet-stream ...
	Add parameter
Footer (Optional)	
ZIP Content	No ▼

7. In **Zip Content**, specify whether to zip the contents before upload.
8. In **Username**, optionally enter the user name that will be used to log on to the REST endpoint.
9. In **Password**, optionally enter the password that will be used to log on to the REST endpoint.

Basic Authentication (Optional)	
Username	
Password	

10. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Additional Request Information

By default, the REST Direct delivery plugins sends multipart/form-data POST requests with the exported file available in the part named 'file' with Content-Type application/octet-stream. The example below shows the properties for such a request:

Headers:

Accept-Encoding = gzip,deflate

Connection = close

Content-Length = 1354

Content-Type = multipart/form-data; boundary=JN_qruUpDaHqm9BgW_b4-adAHDftQbjuvDI

Body (Metadata):

Content-Disposition: form-data; name="file"; filename="result.zip"

Content-Type: application/octet-stream

Via the UI it is possible to change the HTTP method and also to overwrite the Content-Type header. For instance, this header could be set to 'application/xml' for a STEP XML export (unzipped) thereby overwriting the default 'multipart/form-data' value making the request properties be:

Headers:

Accept-Encoding: gzip,deflate

Connection: close

Content-Length: 1354

Content-Type: application/xml

Body: <The exported XML>

SFTP Delivery Method

The preferred method for FTP delivery is the SFTP secure delivery method. The SFTP delivery method allows an exported file to be delivered to an external system and is often used when the output files are large, or when a different or remote system is in use.

This delivery method is also available in Export Manager. For information on the OIEP FTP delivery method, see **FTP Delivery Method**.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

Prior to configuration, clicking the **Host name** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **FTPDeliveryHostName** property. The required format of the property is (square brackets not included): `FTPDeliveryHostName=1=[host1],2=[host2]`.

Using the host name shown in the image, the entry in the properties file would be:

```
FTPDeliveryHostName=1=FTPDeliveryHostServerName
```

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. Click the **Select Delivery Method** list to display the dropdown and select **SFTP**.

2. In **Host name**, select the host name of the SFTP server to be used for the delivery.
3. In **Proxy host name**, add the host name to be used for the server proxy. This field is optional.
4. In **User name**, enter the user name that has access to log on to the FTP server.

5. In **Password**, enter the password that will be used to log on to the FTP server.
6. In **File name template**, to create the file name used for the output, enter template variables, static text, or use a combination. The default template is \$filename-\$timestamp.\$extension and creates a file name like 'GenericXML-1490813266535.xml' based on the settings of the OIEP. Each variable is described below:

- **\$filename** This variable is replaced with text to indicate the format of the delivered file as specified in the Output Templates flipper. For example, the output file name would include the text 'excel' or 'GenericXML' for those formats, or the Event ID for event-based STEPXML. Using the default file name template, a comma-separated value file would be named 'csv-1490808860412.csv' and 'GenericXML-1490810861593.xml' would indicate that Generic XML was used.
- **\$timestamp** This variable is replaced with the time in milliseconds between January 1, 1970, and the time when the file is uploaded, which allows for a unique identifier. For example, 1479230247017 is used for an export on 15 NOV 2016.

Alternatively, the following options can be used to generate a readable date and can include static characters to separate the timestamp elements. Individual elements can be ordered as required, as shown below.

\$timestamp(ddMMyyyy) displays day, month, and year and is output '15112016' for 15 NOV 2016.

\$timestamp(yyyy-MM-dd_HH.mm.ss) displays year, month, day, hour, minutes, and seconds and is output as '2016-11-15_09.32.43' for 2016 NOV 15 at 9:32.43 AM.

- **\$extension** This variable is replaced with the extension of the output file based on the selected format in the Output Templates flipper. For XML-based formats, the output file is set with the .XML extension, while Excel files use .XLS, and comma-separated values use .CSV.

Note: The File Name Template does not support conversions of file formats and can only be used to deliver files in the format specified on the Configuration tab > Output Templates flipper of the outbound integration endpoint.

7. In **Zip before upload**, select an option from the dropdown to specify if the output file should be delivered in a .ZIP (compressed) file format.
 - **Yes** uses the File Name Template and the extension .ZIP. When a file is zipped, the File Name Template is used for the .ZIP file. The \$filename template is replaced with 'result_0', and the selected file type is compressed with the name 'exported'. For example, a zipped STEPXML output with the default File Name Template exported on 15 NOV 2016 results in an output .ZIP file named 'result_0-1479230247017.zip'.
 - **No** uses the File Name Template for the file name along with the appropriate extension for the selected data format.
8. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Wiki Delivery Method

The XWiki Delivery Method works with an OIEP configured to deliver metadata to a specified XWiki advanced open source enterprise wiki system. The contents of the wiki can be accessed by clicking a 'Go to Wiki' link from either Web UI or workbench.

The Wiki Metadata functionality is a licensed component and requires a separate installation recipe. For details and the latest recipe, contact your Stibo Systems representative.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. Review the **Wiki Metadata** topic for details on the functionality and setup.
2. Prior to configuration, clicking the **Base URL** dropdown parameter displays the required property name. Provide a Base URL selection for the dropdown parameter via the sharedconfig.properties file using the case-sensitive **Wiki.WikiUrl** property. A protocol (http:// or https://) and an ending slash (/) are mandatory. The following is an example of a property entry for two wikis.

```
Wiki.WikiUrl=1=http://vp487.vps.ohs.ca/,2=https://127.0.0.1:8080/
```

If only one URL is required, in the example above, eliminate the ',2=https://127.0.0.1:8080/' text.

Note: The format for multiple wikis is 1=url1,2=url2,3=url3, where each entry is incremented and unique.

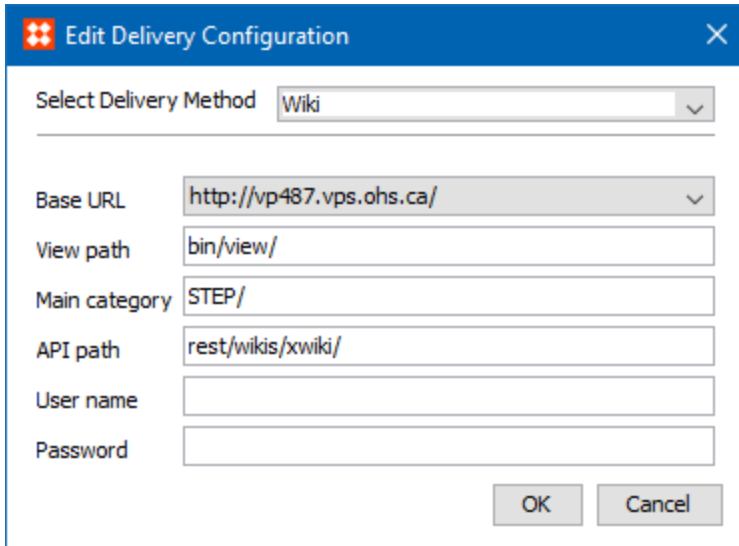
Configuration

For information on a parameter, hover over the parameter field to display help text.

After completing the prerequisite steps, edit the delivery method of the OIEP created during the installation. Use the following steps to configure the OIEP to use the XWiki delivery option.

Important: For the OIEP created automatically upon restarting the server after installing the STEP wiki metadata recipe, most of the following parameters are pre-populated and the default settings should not be changed. Review the parameter details below for more information.

1. For **Select Delivery Method**, choose **XWiki**.



Edit Delivery Configuration

Select Delivery Method: Wiki

Base URL: http://vp487.vps.ohs.ca/

View path: bin/view/

Main category: STEP/

API path: rest/wikis/xwiki/

User name:

Password:

OK Cancel

2. In **Base URL**, select the URL that points to the wiki that will be used for this STEP system. See the Prerequisites section for information on setup.
3. In **View Path**, enter the URL elements between the Base URL and the wiki pages, or main category if defined. This is used for the presentation of wiki pages and allows each user to watch pages they select. The slash at the end is mandatory. As shown in the image above, 'bin/view/' is the view path for the following URL:
http://vp487.vps.ohs.ca/**bin/view**/STEP/attribute/
4. In **Main category**, enter a name for the main category path. This allows you to organize the metadata pages and can be multiple levels by adding a slash, '/', between each one. The slash at the end is mandatory. As shown in the image above, 'STEP/' is the main category for the following URL:
http://vp487.vps.ohs.ca/bin/view/**STEP**/attribute/
5. In **API path**, specify the path for the wiki REST API, which is used to make modifications to the wiki pages. The slash at the end is mandatory.
6. In **User name**, enter the user with permission to call the REST API and who can make modifications to it.
7. In **Password**, enter the password for the user name.
8. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

OIEP - Event-Based - Event Triggering Definitions Tab

When Event Queue has been selected as the data source in an Outbound Integration Endpoint (OIEP), or when setting up an Event Processor (EP), you can specify if core events (modify, delete, or create events) should be collected by the OIEP or EP for specific object types.

- Workspace revisable objects, such as a product, generate an event when data changes are approved.
- Global revisable objects, such as an attribute or an entity storing patient data, generate an event as soon as the data changes because approval is not available or required.

Event triggering definitions define what changes (events) are important and should be monitored by the OIEP or EP. For an OIEP, changes to any element defined on this tab are then formatted based on the output template, and ultimately queued for export. For an EP, changes to any element defined on this tab are then handled by the selected core processor.

For details about event generation, see the **Events** section of the **System Setup** documentation.

For event-based OIEPs or EPs, the Event Triggering Definitions tab allows you to add or edit the available triggers. At a minimum, you must set an Object Type trigger before the OIEP or the EP can be enabled to queue events. Additional triggers can be added to further limit the data monitored by the OIEP. For more information, see the flipper section for each trigger below.

Important: For an OIEP, an output template must be created for each event trigger, otherwise the triggered data will not be exported. For more information, see **OIEP - Event-Based - Output Templates Flipper**.

System Setup

- 1WS 7.0 Provider Send
- 1WS 7.0 Receiver Send
- Asset Caching
- Change Attribute Link
- GDSN Receiver Data Po
- GREExport
- Item Modify Non STEPXI
- Item Modify
- MultiValue MultiContext
- PrintOIEP
- Sellers Sender
- Static Data Set
- Static Data Set Pt2
- Website**
- Web UIs
- Workflow Profiles
- Workflows
- XSLTStyle
- Derived Events
- Object Types & Structures
- Tags
- Units
- Users & Groups
- Reference Types
- Workspaces
- Table
- Keys
- Event Queues
- Component Models
- Recycle Bin

Website - Event Triggering Definitions

Background Processes | Statistics | Error Log Excerpts | Log | Status

Outbound Integration Endpoint | Configuration | **Event Triggering Definitions**

Triggering Object Types

Object Types	Event Filter	Generate Event
> Level 3, WebLevel3	Is Specific Hierarchy Item (IsProductItem)	Item To Sales Item Event Generator (ItemT ...

[Add Object Type](#)

Triggering Attributes

Name
> Landing Page Copy

[Add Attribute](#)

Triggering Table Types

Table Types
>

[Add Table Type](#)

Reference Type Triggers

Reference Types
>

[Add Reference Type](#)

Triggering Data Container Types

Data Container Types
>

[Add Data Container Type](#)

Miscellaneous Triggers

- ☒ Names enabled
- ☒ Parent links enabled
- ☒ Attribute-links enabled
- ☒ Index-word Hierarchy enabled

Triggering Object Types Flipper

System Setup

- 1WS 7.0 Provider Send
- 1WS 7.0 Receiver Send
- Asset Caching
- Change Attribute Link
- GDSN Receiver Data Po
- GREExport
- Item Modify Non STEPXI
- Item Modify
- MultiValue MultiContext
- PrintOIEP
- Sellers Sender
- Static Data Set
- Static Data Set Pt2
- Website**
- Web UIs
- Workflow Profiles
- Workflows
- XSLTStyle
- Derived Events
- Object Types & Structures
- Tags
- Units
- Users & Groups
- Reference Types
- Workspaces
- Table
- Keys
- Event Queues
- Component Models
- Recycle Bin

Website - Event Triggering Definitions

Background Processes | Statistics | Error Log Excerpts | Log | Status

Outbound Integration Endpoint | Configuration | **Event Triggering Definitions** | Background Processes | Statistics | Error Log Excerpts | Log | Status

Triggering Object Types

Object Types	Event Filter	Generate Event
> Level4		

[Add Object Type](#)

Triggering object types are used to filter the data being monitored by the OIEP. For example, if you are only interested in exporting data when a change is made on a specific object type, there is no reason for the OIEP to listen to events happening on other object types. Setting a trigger on only the required object type reduces the load on the OIEP.

Consider the following when setting up:

- If functionality should vary by object type, a unique endpoint should be created for each object type.
- Common setup is to include at least one attribute or attribute group unless the integration endpoint handles System Setup data only, or Create / Delete events only.
- If standard object types (product, entity, classification) are selected as triggers and the output template includes a Modify event, at least one attribute or attribute group must be selected to trigger the Modify event.

Object Types

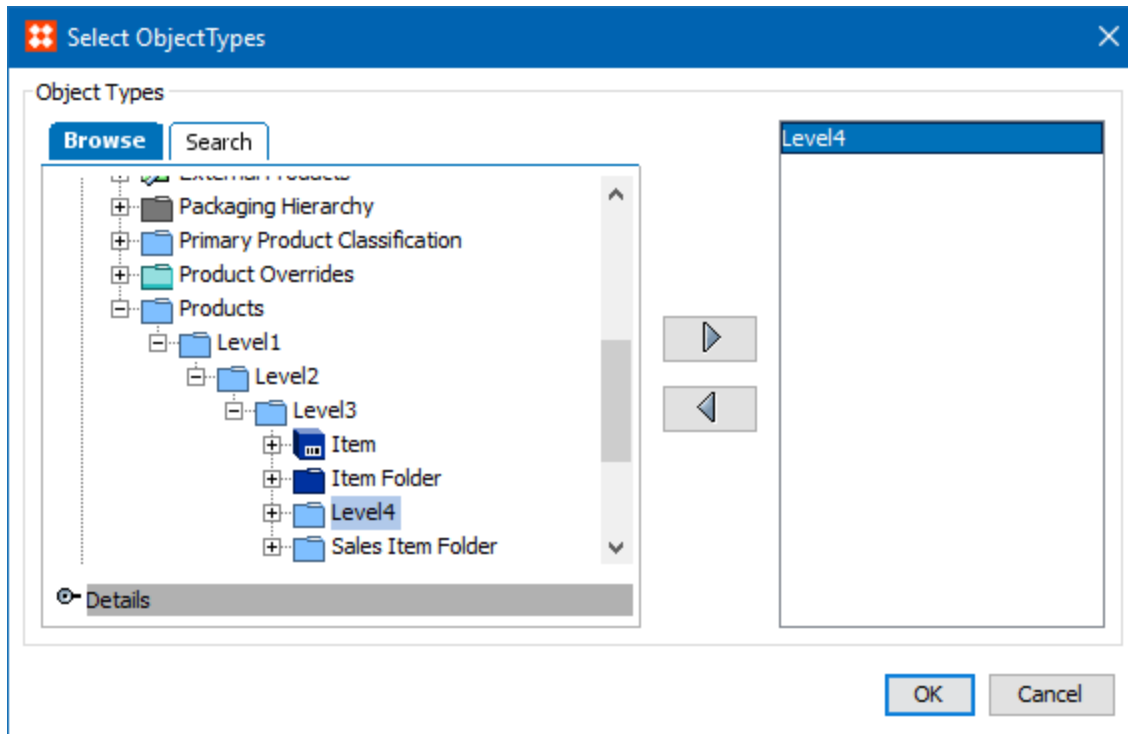
The Triggering Object Types section allows you to identify object type(s) that are important for this OIEP. The OIEP will listen for events on these object types in order to generate the desired output as defined by the output template configuration. When listening for events on changes to objects like attributes, the object type for these must be selected as well. All object types can be added in a single row unless business rules are used to filter and process events (functionality described below).

Keep in mind that events are triggered based on the revisability setting of the object type. Global revisable events are triggered on a change transaction. Workspace revisable events are triggered on approval. For more information, see the **Events** section of the **System Setup** documentation.

Important: The OIEP cannot be enabled or queue events until at least one object type is set as a trigger.

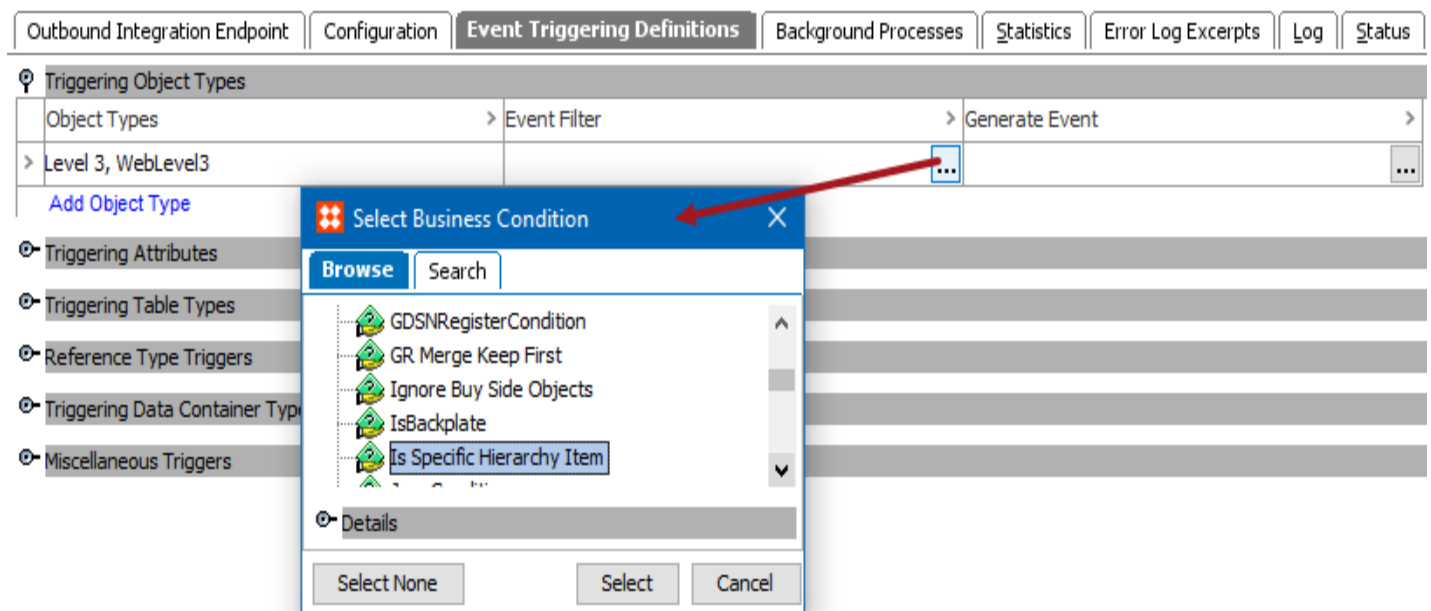
To select the object types:

- Click the **Add Object Type** link, then browse or search for the relevant object type.
- While the most commonly used object types are displayed in the browsing area, use Search to find other object types.
- Any System Setup data (references, attributes, units, etc.) can be selected.
- At least one object type **must** be selected.



Event Filter

Event filters are business conditions that can be added for each row of the Object Type Triggering section and can be used to apply additional event filtering. Event filters are run after approval (when approval is required), and can be used in place of a preprocessor. Despite being created on an OIEP, an event filter runs at the time of event generation. If the condition evaluates to true, the event is written and registered on the event queue; if the condition evaluates to false, the event is not written at all (unless a corresponding derived event is generated).



For example, when a change happens on a parent object, but you only want to output the children, you can use a business condition to filter out the parent. Then use a business action (see the Generate Event section below) to trigger an export of the children.

Conditions used as event filters are typically written in JavaScript since the built-in condition options, in most cases, will not provide the required flexibility.

For JavaScript based event filter conditions, the following available binds are of special relevance:

- **Event Type** binds in a derived event so you can check if the event currently being handed is of a specific derived event type.
- **Current Event Type** tests if the event currently being handed is a `BasicEventType` or a `DerivedEventType`. Assuming that 'Current Event Type' is bound to the variable 'currentEventType':

```
if (currentEventType instanceof
com.stibo.core.domain.eventqueue.BasicEventType) {
// It is a core event
}

if (currentEventType.equals
(com.stibo.core.domain.eventqueue.BasicEventType.Modify)) {
// It is a core Modify event
}
```

You can also test derived events and if you have created an 'Event Type' binding, test if the current event is of a specific type. Assuming that 'Current Event Type' is bound to the variable 'currentEventType' and the derived event 'webModify' is bound to the variable 'webModify':

```
if (currentEventType instanceof
com.stibo.core.domain.eventqueue.DerivedEventType) {
// It is a derived event
}

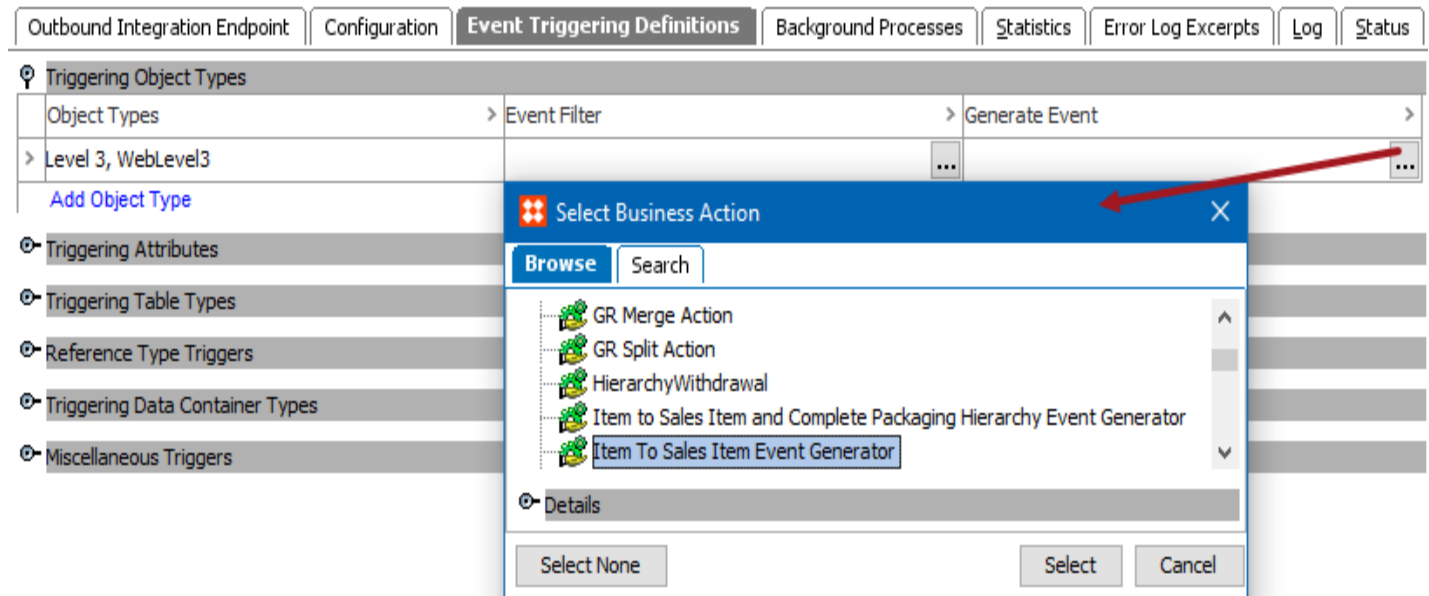
if (currentEventType.equals(webModify)) {
// It is a derived WebModify event
}
```

- **Approve Context** lets you compare an object in Main and Approved before the actual synchronization is carried out. It is important to understand that event filter conditions are evaluated immediately when events are generated. If the event is generated based on an approval, in the event filter condition, you will have access to the Approved Context. See the **Approve Context Bind** section of **Business Rules Actions** documentation or the 'ApprovePlugin.ApproveContext' interface in the Public Java API JavaDoc.

For more information, see **Event-Based OIEP Example Business Rules for Derived Events**.

Generate Event

Event generators are business actions that run after approval (when approval is required), and can be added for each row of the object type triggering section. The purpose of an event generator is to produce derived events based on other events (most often core events). New core events cannot be generated. For more information, see **Derived Events** of the **System Setup / Super User Guide** documentation.



Event generators have access to the current event regardless of the chosen event filter, and an event filter is not required. This means derived events can be generated from events discarded by the filter. As with event filters, the event generator logic is also executed immediately when an event is generated.

Generated actions must typically be formulated in JavaScript. In addition to the binds mentioned in the event filter section above, the following are of special relevance for event generators.

- **Current Event Queue** lets you bind in the current event queue so that you can put new derived events on the queue (after filtering). Assuming that 'Current Event Queue' is bound to 'currentEventQueue', the derived event type 'webModify' is bound to the variable 'webModify', and 'Current Object' is bound to 'currentObject', an event can be generated as follows:

```
currentEventQueue.queueDerivedEvent(webModify, currentObject);
```

- **Event Queue** lets you place an event on a queue different from current. Use this bind if you produce events from actions not hooked in as generators.

Important: Avoid creating an endless loop by considering that any derived event generated by an event generator for the current event queue will pass through the object type filter and any event filters and event generators again.

For more information, see **Event-Based OIEP Example Business Rules for Derived Events**.

Triggering Attributes Flipper

Website - Event Triggering Definitions

Outbound Integration Endpoint | Configuration | **Event Triggering Definitions** | Background Processes | Statistics | Error Log Excerpts | Log | Status

Triggering Object Types

Triggering Attributes

Name >

Add Attribute

Triggering Attributes are used to filter the data being monitored by the OIEP in conjunction with the Triggering Object Types section. When a Triggering Attribute is defined, and that attribute's value changes for objects of the object types selected in the Triggering Object Types section, events are registered. The 'Triggering Attributes' section is a positive list, so if no attributes are selected, events will not be registered when, for example, a value changes and the object holding the value is approved.

Be aware that you can select both revisable and externally maintained attributes, and events are generated as follows:

- Changes to values for externally maintained attributes and global revisable entities cause an event to be generated immediately.
- Changes to values for revisable attributes cause an event to be generated when the object holding the values is approved.

For more information, see the **Events** section of the **System Setup** documentation.

Outbound Integration Endpoint | Configuration | **Event Triggering Definitions** | Backgro

Triggering Object Types

Triggering Attributes

Name

> Landing Page Copy

Add Attribute

Triggering Table Types

Reference Type Triggers

Triggering Data Container Types

Miscellaneous Triggers

Select Attribute or Attribute Group

Browse Search

Dimensions

Width

Width (range)

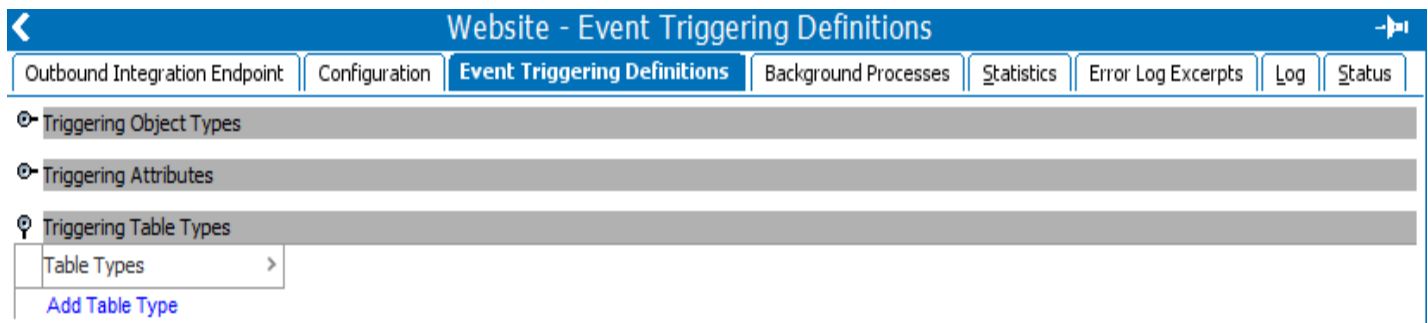
Height

Details

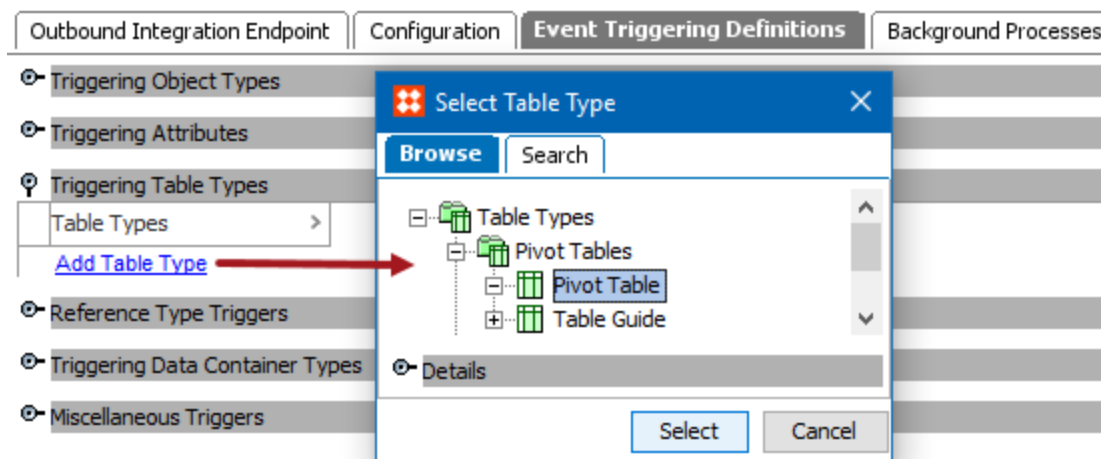
Select Cancel

- Click **Add Attribute**, then browse or search for the relevant attribute(s) and/or attribute group(s).
- At least one attribute or attribute group must be selected, unless the integration handles ONLY System Setup data OR Create / Delete events.
- When an attribute or attribute group is selected as a trigger, an output template must be configured with a Modify event or the endpoint will skip the event since it cannot find a corresponding template. For more information, see the **OIEP - Event-Based - Output Templates Flipper**.

Triggering Table Types Flipper

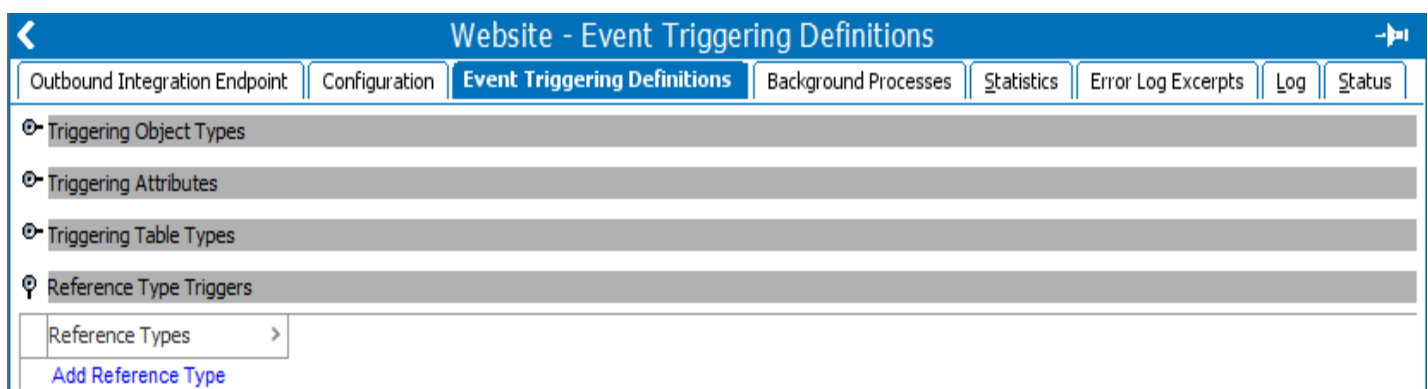


Allows you to specify (filter) table types for which you want events registered when tables owned by objects of the type listed in the Triggering Object Types section change. Common setup is to use tables and this setting only for print output.



- Click **Add Table Type**, then browse or search for the relevant table type. Modifications to the table configuration on the product. For example, adding, removing, or modifying the table rows or columns; or changing the data on a free text cell followed by an approval will trigger an event for the endpoint.

Reference Type Triggers Flipper

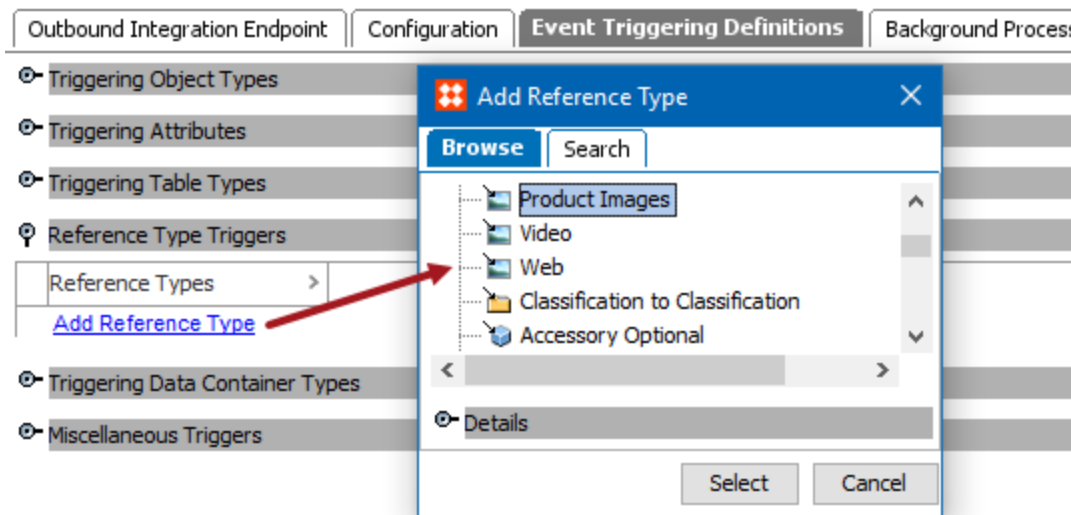


Use reference / link types to define (filter) which reference / link types to listen to for changes. This is a positive list so only the items listed are triggers. If functionality should vary by object type, a unique endpoint should be created for each object type.

Be aware that you can select both revisable and externally maintained reference / link types and events are generated as follows:

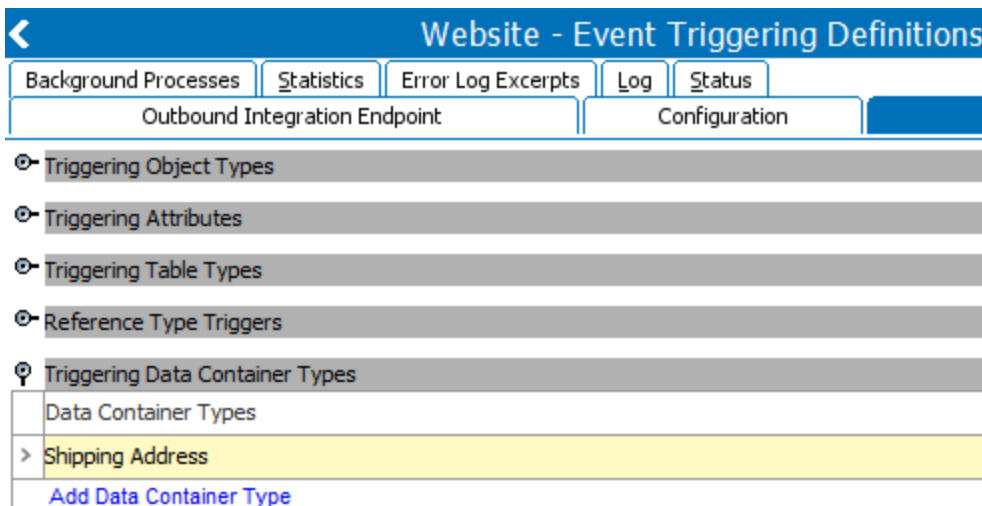
- Changes to values for externally maintained reference / link types cause an event to be generated immediately if an object type on which the reference is valid is selected as a triggering object type.
- Changes to values for revisable reference / link types cause an event to be generated when the object holding the values is approved if an object type on which the reference is valid is selected as a triggering object type.

For more information, see the **Events** section of the **System Setup** documentation.

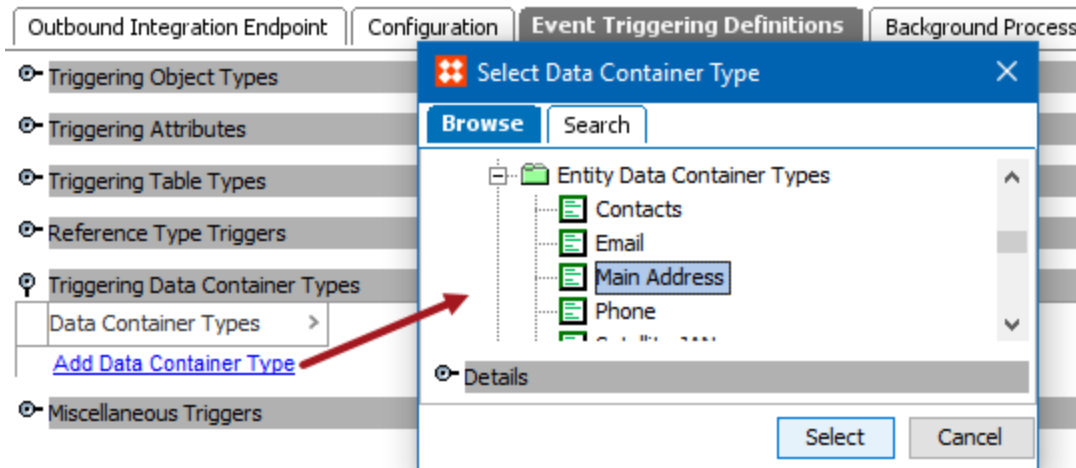


- Click **Add Reference Type**, then browse or search for the relevant reference type.

Triggering Data Container Types

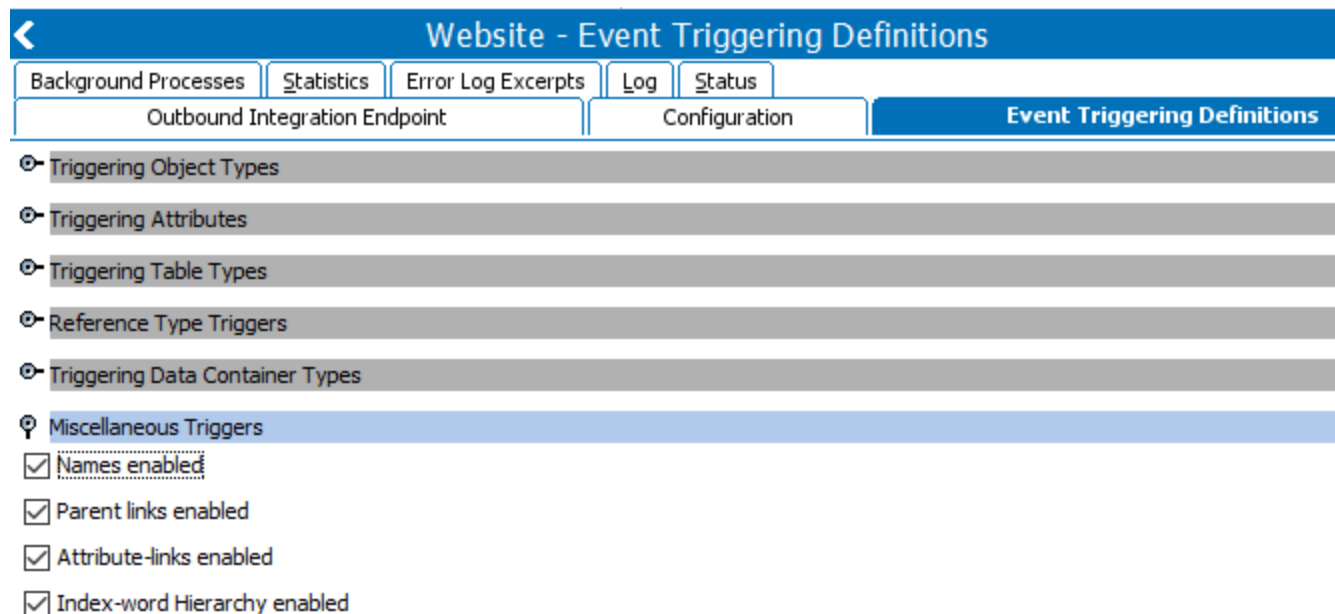


When monitoring changes on entity objects, specific data container types can be configured to trigger events upon creation / deletion of corresponding data containers. Any data container types omitted from the triggering definition will not trigger events.



- Click **Add Data Container Type**, then browse or search for the relevant data container type. Anytime a data container of this type is created / deleted it will trigger an event for the endpoint.

Miscellaneous Triggers Flipper



Miscellaneous triggers complement the triggers set in the other flippers. By default, the following miscellaneous triggers are set to additionally filter the data being monitored by the OIEP:

- **Names enabled:** Triggers an event when a change is made to the STEP Name of triggering object types for a global revisable object, or when a workspace revisable object is approved. An output template must be

configured with a Modify event or the endpoint will skip the event since it cannot find a corresponding template.

- **Parent links enabled:** Triggers an event when changes are made to the parentage of triggering object types for a global revisable object, or when a workspace revisable object is approved. For example, when a product or asset is moved into another parent folder. An output template must be configured with a Modify event or the endpoint will skip the event since it cannot find a corresponding template.
- **Attribute-links enabled:** Triggers an event when attribute hierarchy link is changed on an object of the types for a global revisable object, or when a workspace revisable object is approved. For example, when an attribute is linked to a product or classification.
- **Index-word hierarchy enabled:** Triggers an event when an index word changes, or an index hierarchy link is added on a triggering object type. Common setup is to use indexes and this option for print output.

OIEP - Event-Based - Event Queue Configuration Flipper

On the OIEP Configuration tab, the Event Queue Configuration flipper is only displayed when the selected OIEP uses the Event Queue data source, as defined on the Choose Data Source step of the wizard. For more information, see **OIEP - Event-Based - Choose Data Source**.

Click the Edit Configuration link to open the **Edit Configuration** dialog.

The screenshot shows the 'Outbound Integration Endpoint' configuration page. The 'Configuration' tab is active, and the 'Event Queue Configuration' flipper is expanded. Below the flipper, there are four buttons: 'Forward', 'Rewind', 'Purge', and 'Republish'. A table displays the following configuration parameters:

Parameter	Value
Days to retain events	0
Number of events to batch	1000
Number of event batches to include per delivery	1
Queue Status	Read Events
Unread events (approximated)	Click to estimate ...

Below the table, there is a link labeled 'Edit Configuration'. A red arrow points from this link to the 'Edit Configuration' dialog box. The dialog box has a title bar with a close button (X). Inside the dialog, the 'Choose Data Source' dropdown is set to 'Event Queue Data Source'. The 'Days to retain events' input field contains '0'. The 'Event Batching' section has two radio buttons: 'No event batching - one message is generated and handled for each event and appropriate message template will be used depending on object type and event type.' (unselected) and 'Event batching - one message is generated and handled for multiple events that match the same template.' (selected). The 'Specify maximum number of events to read for each generated message:' input field contains '1000'. The 'Bundle Messages' section has two radio buttons: 'No message bundling - each message is delivered separately.' (selected) and 'Bundle messages - multiple message results will be delivered together. Specify maximum number of message results to bundle:' (unselected). The 'Specify maximum number of message results to bundle:' input field contains '1'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

- For information on the **Event Actions** buttons and the **Days to retain events** parameter, see **Event-Based OIEP Forward, Rewind, Purge, and Republish**.

- For information on the **Number of events to batch** and **Number of event batches to include per delivery** (which are controlled by the Event Batching and Bundle Messages parameters on the Edit Configuration dialog), see **Event-Based OIEP Event Batching**.
- For information on the **Queue Status** parameter, see **Event-Based OIEP Status and Queue Status**.
- For information on the **Unread events** parameter, see **Event-Based OIEP Queued Events**.

Creating a Select Objects Outbound Integration Endpoint

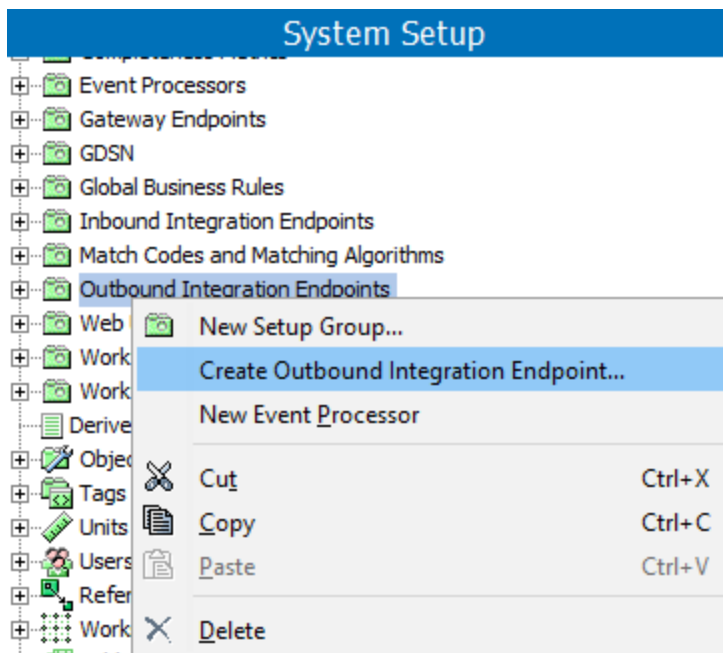
A Select Object OIEP publishes data non-incrementally from selected hierarchies (one or more root nodes) to external systems on scheduled intervals. This type of OIEP uses a static set of data and works exactly like a scheduled export created in Export Manager, with the differences being that the OIEP has more standard delivery options and offers extended monitoring capabilities.

For example, a Select Object OIEP can be used to send a specific set of data to an external system on regular intervals. Configuration of the output templates allow the same data to be delivered at the same time, using multiple formats, a pre-processor, or post-processor.

Select Object OIEP Setup

After creating a setup group for outbound integration endpoints, create a Select Object outbound integration endpoint to send a static set of data from STEP. For information on an event-based OIEP, that can monitor STEP for changes and export as needed, see **Creating an Event-Based Outbound Integration Endpoint**.

1. In System Setup, right-click the Outbound Integrations Endpoints setup group, and click **Create Outbound Integration Endpoint**.



2. The Outbound Integration Endpoint wizard displays and involves the following steps:

The screenshot shows a window titled "Outbound Integration Endpoint Wizard" with a close button (X) in the top right corner. On the left, a "Steps" sidebar lists three steps: "1. Identify Endpoint" (highlighted in blue), "2. Choose Data Source", and "3. Configure Endpoint". The main area is titled "Identify Endpoint" and contains four input fields: "Endpoint ID", "Endpoint Name", "Description", and "User". The "User" field has a dropdown arrow (three dots) to its right. At the bottom right, there are four buttons: "Back", "Next", "Finish", and "Cancel".

3. Complete setup requires providing data in the wizard as well as some manual configuration. All setup includes the following sections:
 - In the wizard, **OIEP - Select Objects - Identify Endpoint** allows you to specify the name and ID of the endpoint and identify the user whose privileges are applied.
 - In the wizard, **OIEP - Select Objects - Choose Data Source** allows you to specify a static set of data or an event queue.
 - In the wizard, **OIEP - Select Objects - Configure Endpoint** allows you to specify processing and context settings for the integration endpoint.
 - In the OIEP editor, **OIEP - Select Objects - Manual Configuration** directs you to complete the required manual setup.

OIEP - Select Objects - Identify Endpoint

The screenshot shows the 'Outbound Integration Endpoint Wizard' window. On the left, a 'Steps' pane lists three steps: '1. Identify Endpoint' (highlighted), '2. Choose Data Source', and '3. Configure Endpoint'. The main area is titled 'Identify Endpoint' and contains four input fields: 'Endpoint ID' with the value 'OutboundEndpoint', 'Endpoint Name' with 'Outbound Endpoint', 'Description' with 'Static data exported on schedule', and 'User' with 'User J (USERJ)' and an ellipsis button. At the bottom, there are four buttons: 'Back', 'Next' (which is highlighted with a dashed border), 'Finish', and 'Cancel'.

1. For **Endpoint ID**, enter an ID. Common setup is to use no spaces or punctuation.
2. For **Endpoint Name**, enter a **Name**. Common setup is to repeat the ID with added spaces for readability.
3. For **Description**, enter an optional **Description**. Common setup is to include a description for future reference.
4. For **User**, type a user name into the field, or click the ellipsis button (...) to search or browse for a user. The privileges of the selected user determine which actions the integration endpoint 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 endpoint are easily identified.

Once the Endpoint ID, Endpoint Name, and User are supplied, the Next button is enabled.

5. Click **Next** to display **OIEP - Select Objects - Choose Data Source**.

OIEP - Select Objects - Choose Data Source

Outbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
- 2. Choose Data Source**
3. Configure Endpoint

Choose Data Source

Choose Data Source: **Select Objects**

ID	Name	Object Type	Path
> Add Objects			

Back **Next** **Finish** **Cancel**

1. Choose the data source **Select Objects** for non-incremental exports where you want to publish data for specific hierarchies every time the endpoint is invoked.

Note: An **Event Queue Data Source** requires a separate setup. For detailed steps, see the **Creating an Outbound Integration Endpoint** documentation.

2. Click the **Add Objects** link to display the Select Objects dialog. Select one or more root node objects from the Tree hierarchies to be exported and click the **Select** button.

Note: Selecting a Tree hierarchy object is not required if the endpoint should export system-specific data, such as attributes, LOVs, and units that are not used in any specific product, classification, or entity hierarchy.

Outbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
- 2. Choose Data Source**
3. Configure Endpoint

Choose Data Source

Choose Data Source: **Select Objects**

ID	Name	Object Type	Path
> Add Objects			

Select Objects

Browse **Search**

- Products
 - Apparel
 - Upper Body Wear
 - Head Wear
 - Footwear
 - Safety

Details

Select **Cancel**

Back **Next** **Finish** **Cancel**

The selection(s) are displayed as a data source.

✦ Outbound Integration Endpoint Wizard

Steps

1. Identify Endpoint

2. Choose Data Source

3. Configure Endpoint

Choose Data Source

Choose Data Source

Select Objects

ID	Name	Object Type	Path
> 18201	Upper Body Wear	Level 2	Primary Product Hierarchy/Products/Apparel/Uppe...
> Add Objects			

<

>

Back

Next

Finish

Cancel

3. Click **Next** to display **OIEP - Select Objects - Configure Endpoint**.

OIEP - Select Objects - Configure Endpoint

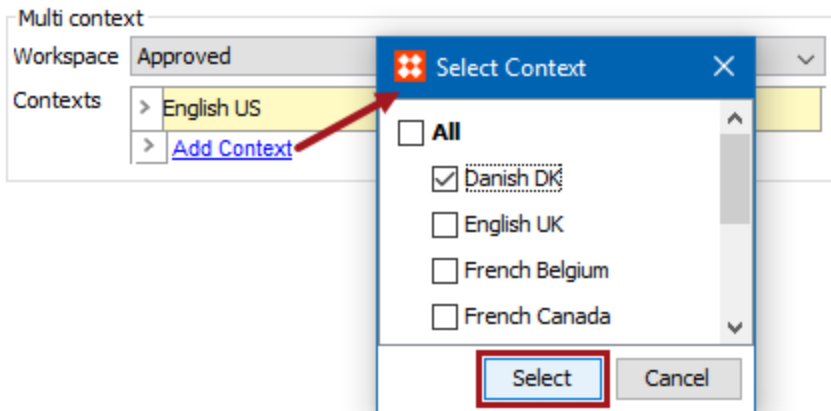
1. Set the **Processing** options:

- For the **Processing Engine** select **STEP Exporter** from the dropdown, in order to use the same functionality as the Export Manager. This is the only processing engine option on a standard STEP system and performs the actual data export. If you need a customized processing engine, please contact Stibo Systems. For more information, see **Outbound Integration Endpoint Structure** documentation.
- The **Transactional Settings** option does not have the same significance for OIEPs as they do for Inbound Integration Endpoints. The 'Strict' mode is set automatically, and while you can specify the Transactional Setting for OIEPs based on a static selection, the setting has no impact on the processing. For more information about Transactions Settings, see the **Inbound Integration Endpoint Transactional Settings** documentation.

- The **Number of threads** option is only valid for an Event Queue Data Source endpoint.
2. Set the **Multi context** options that the endpoint should use to export data. By default, the Workspace and Context selected is the one in which the Outbound Integration Endpoint has been created.

- For **Workspace**, use the dropdown to select the desired workspace. Common setup is to use the Approved workspace, except when you need to generate events for objects before they are approved, for example, during import or from a workflow.
- For **Contexts**, when exporting data in the STEPXML format or when using a custom cross-context enabled format, one file can contain data from multiple contexts. For other formats, the standard 'Context splitter Post-processor' should be used to generate separate files for the selected contexts.

Click the **Add Context** link and choose one or more contexts. Click the **Select** button to add them to the Contexts list.



3. Click **Finish** to complete the wizard.
4. Perform the manual configuration described in **OIEP - Select Objects - Manual Configuration**.

OIEP - Select Objects - Manual Configuration

After completing the Outbound Integration Endpoint wizard, for a Select Objects OIEP, the OIEP editor includes additional parameters that must be set manually before the OIEP can run.

To manually edit or review other parameter settings on an OIEP, see the **Maintaining an Outbound Integration Endpoint** documentation.

1. In the OIEP editor, configure to send an email if an endpoint-related background process fails as described in the **Error Reporter** section of the **OIEP - Configuration Flipper** documentation.
2. In the OIEP editor, determine when the OIEP runs as described in the **Schedule** section of the **OIEP - Configuration Flipper** documentation.
3. In the OIEP editor, specify the objects to be output, the format, and pre- or post-processors (if any) as described in **OIEP - Select Objects - Output Templates Flipper**.
4. In the OIEP editor, determine how data is delivered as described in **OIEP - Delivery Method Flipper**.

Static Data Set - Configuration

Outbound Integration Endpoint
Configuration
Background Processes
Statistics
Error Log Excerpts
Log
Status

Configuration

Process Engine	STEP Exporter
Error reporter 1	Not Defined ...
Schedule 2	Not scheduled ...
Queue for endpoint	OutboundQueue
Queue for endpoint processes	Out
Transactional settings	Strict
Number of threads	1
Maximum number of old processes	100
Maximum age of old processes	1w
Contexts	English US
Workspace	Approved

Object Selection Configuration

ID	Name	Object Type	Path
18209	Cotton T-Shirts	Item Family	Primary Product Hierarchy/Prod...

[Edit Configuration](#)

Output Templates 3

Object-Eventtype	Format	Pre-Processor	Post-Processor
Item Family, Level 1	Excel (2 mappings)	None	None

[Add configuration](#)

Delivery Method 4

Deploy

Extract to directory	workarea
Copy to file	\$tmpdir/zip-\$timestamp.\$extension

[Edit Delivery](#)

5. Enable the endpoint and invoke it as described in **Running an Outbound Integration Endpoint**.

Static Data Set rev.0.4 - Outbound Integration Endpoint				
Outbound Integration Endpoint			Configuration	Background Processes
			Statistics	Error Log Excerpts
			Log	Status
Description				
Name	>	>	Value	>
ID			StaticDataSet	
Name			Static Data Set	
Object Type			Outbound Integration Endpoint Type	
Revision			0.4 Last edited by USERJ on Tue Jan 10 11:15:47 EST 2017	
User			User J (USERJ)	...
Description				
Enabled			No	
Endpoint Status		5	Stopped	
Last run			2016-11-15 13:39:12	
Next run			9998-01-12 00:31:00	
Integration Endpoint Log				

OIEP - Configuration Flipper

The OIEP Configuration flipper includes the same parameters for both Event-Based and Select Objects endpoints. Each parameter is described below.

Outbound Integration Endpoint	Configuration	Event Trigg
🔑 Configuration		
Process Engine	STEP Exporter	
Error reporter	Not Defined	
Schedule	Not scheduled	
Queue for endpoint	OutboundQueue	
Queue for endpoint processes	Out	
Transactional settings	Strict	
Number of threads	1	
Maximum number of old processes	100	
Maximum age of old processes	1w	
Contexts	Danish DK, English US	
Workspace	Approved	
⊖ Event Queue Configuration		
⊖ Output Templates		
⊖ Delivery Method		

Process Engine

STEP Exporter is the only processing engine option on a standard STEP system. If you need a customized processing engine, please contact Stibo Systems.

The processing engine is originally selected in the wizard when configuring the endpoint. For more information, see **OIEP - Event-Based - Configure Endpoint** documentation or **OIEP - Select Objects - Configure Endpoint** documentation.

Error Reporter

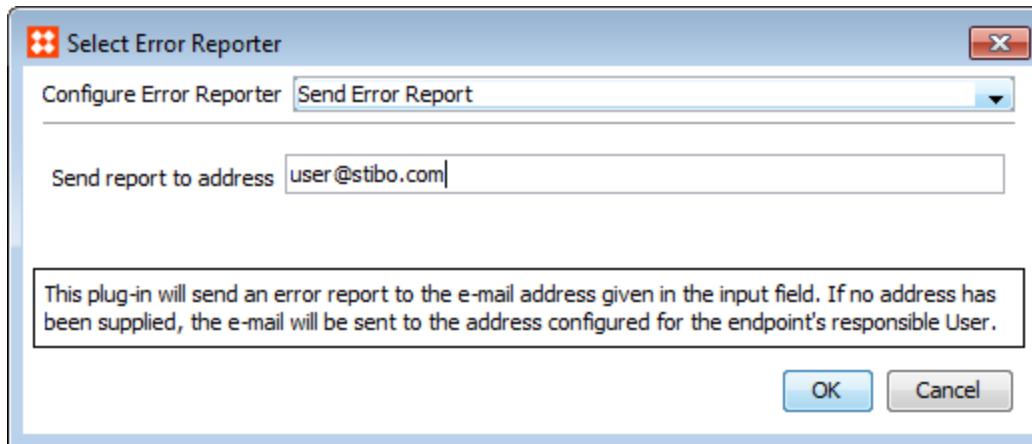
The 'Error reporter' parameter is not included in the wizard, but it can be configured to send an error report email if a background process fails. The email contains information about the failed endpoint, including the background process, failed file, failing process step, cause of the error, and a copy of the file that triggered the error. A standard STEP system (without custom extension) includes only the 'Send Error Report' option.

By default, error reports are sent when either errors or warnings are encountered. To disable error reports when only warnings are reported, update the sharedconfig.properties file on the app server to include the following property:

```
Integration.Endpoint.SendEmailErrorReportOnWarning=false
```

Configure the Error Report Email

1. On the relevant OIEP > Configuration tab > Configuration flipper > click into the **Error Report** parameter to display the ellipsis button (...).
2. Click the ellipsis button (...) to display the **Select Error Reporter** dialog.



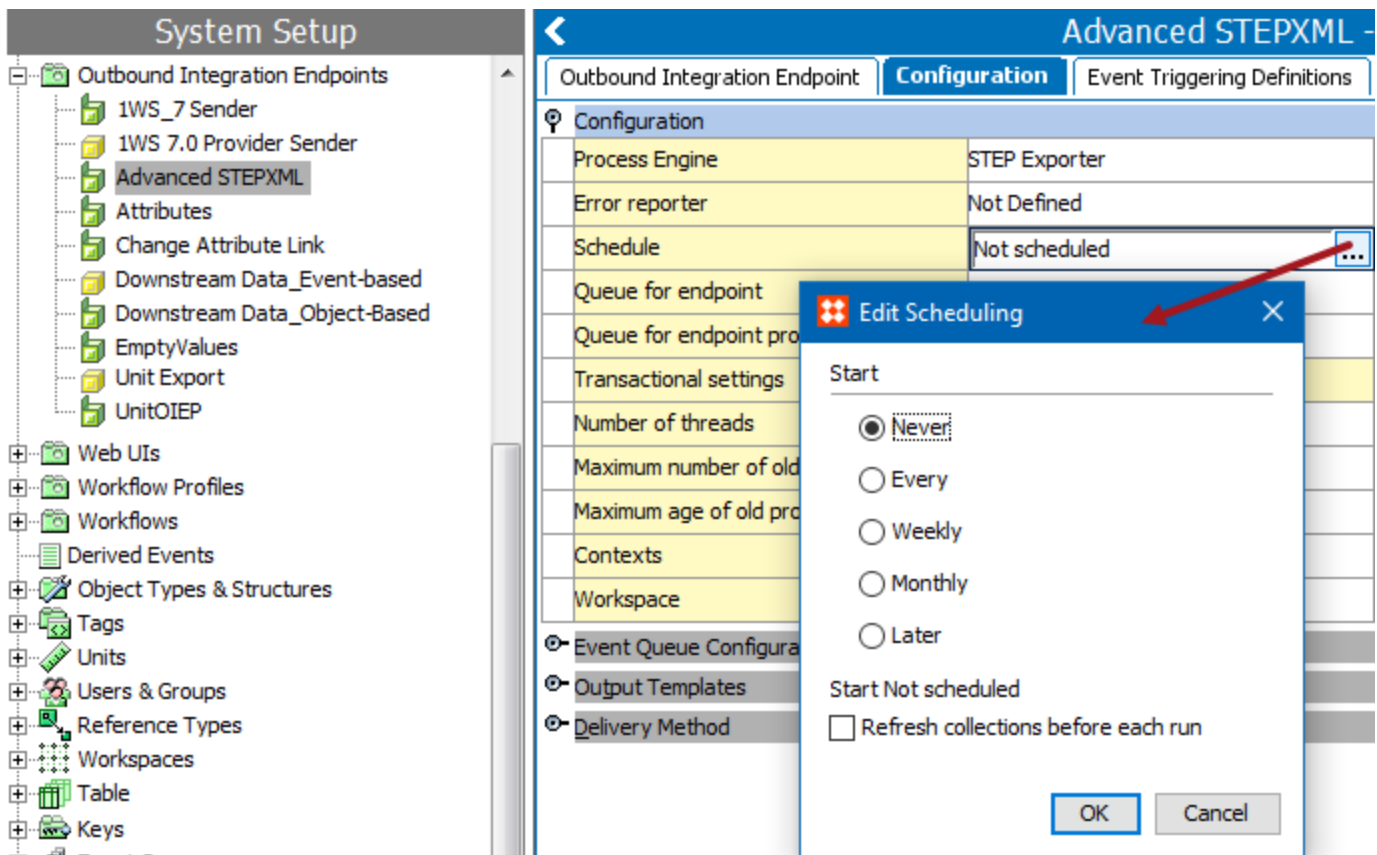
3. In Configure Error Reporter field, select **Send Error Report**.
4. In **Send report to address** field, enter one or more email addresses for the error report recipient(s). Use a semicolon-separated list to add more than one email address.

Note: If no email address is entered, the error report will be sent to the email address of the user who created the integration endpoint. If no email is defined for this user, or if no mail server is defined in the configuration, the error report will be written to the failed Background Process Execution Log.

Schedule

The schedule parameter is not included in the wizard, but it can be configured to repeatedly run the OIEP. For Select Objects OIEPs, the background processes runs as scheduled, regardless of changes to the data. For Event-Based OIEPs, the background process runs as scheduled but only when events are available for processing.

1. On the relevant OIEP > Configuration tab > Configuration flipper > click into the **Schedule** parameter to display the ellipsis button (...).
2. Click the ellipsis button (...), and then select one of the following options:



- **Never** - invoke the endpoint 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 endpoint repeatedly, every selected number of minutes. One (1) minute is the shortest interval allowed and is closest to real time. Enter the number of minutes in the text box. The selection is summarized at the bottom of the dialog.

☐ Never
 minutes
 ☒ Every
 ☐ Weekly
 ☐ Later

Start every minute

- **Weekly** - automatically run the endpoint repeatedly, based on the selected time, start and end dates, and days of the week. Use this option if a daily schedule is needed. The 'Start at' parameter determines the time of day that the endpoint will run. The 'Start on' parameter determines the date the endpoint will first run, while the 'End on' parameter determines the date of the endpoint's final run. The 'Every' checkboxes determine the days of the week when the endpoint will run. The selections are summarized at the bottom of the dialog.

☐ Never	Start at (hh:mm):	21:10
☐ Every	Start on (yyyy-mm-dd):	2017-07-06
☒ Weekly	End on (yyyy-mm-dd):	-
☐ Later	Every:	☒ Mon ☒ Sat ☐ Tue ☐ Sun ☒ Wed ☐ Thu ☐ Fri

Start every Mon, Wed, Sat 21:10:00 EDT, starting Thu Jul 06 2017

- **Monthly** - automatically run the endpoint 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 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' dropdown parameter selections for the week of the month and the day of the week determine when the endpoint will run. The selections are summarized at the bottom of the dialog.

☐ Never	Start at (hh:mm):	22:00
☐ Every	Start on (yyyy-mm-dd):	2017-07-06
☐ Weekly	End on (yyyy-mm-dd):	-
☒ Monthly	Every:	Third
☐ Later		Thursday

Start every third Thu 22:00:00 EDT, starting Thu Jul 06 2017

- **Later** - automatically run the endpoint only once, at the time and date specified. The selections are summarized at the bottom of the dialog.

☐ Never	Start at (hh:mm):	20:30
☐ Every	Start on (yyyy-mm-dd):	2017-07-06
☐ Weekly		
☒ Later		

Start at Thu Jul 06 20:30:00 EDT 2017

2. Select **Refresh collections before each run** if the outbound integration endpoint contains collections that should be automatically refreshed before each run.

Queue for Endpoint

'Queue for endpoint' is the name for the queue that is used by the OIEP Background Process to poll the endpoint. The first time you activate the endpoint, a queue with the specified name is created if it does not already exist. If in doubt about how to populate this parameter, create a new queue for the OIEP, for example, including the OIEP ID in the name.

Queue for Endpoint Processes

'Queue for endpoint processes' is the name for the queue that is used by the background processes started by the endpoint to handle the actual export. The queue is automatically created on the system if it does not already exist. High priority integrations or integrations with long-running processes should typically have their own queue, for example, including the OIEP ID in the queue name.

Transactional Settings

'Transactional Settings' do not have the same significance for OIEPs as they do for Inbound Integration Endpoints. The 'Strict' mode is set automatically, and while you can specify the Transactional Setting for OIEPs based on a static selection, the setting has no impact on the processing. For more information about Transactions Settings, see the **Inbound Integration Endpoint Transactional Settings** documentation.

Number of Threads

Although 'Number of threads' is available on the Configuration tab for both static and event-based OIEPs, the setting only affects event-based processes. The default thread setting is one (1), in which case the endpoint produces a single message at a time, with all events in the batch processed serially. Increasing the thread number results in each batch size being divided by the thread number so that the contents of a batch can be processed in parallel. For more information, see **Event-Based OIEP Multithreading Support**.

Maximum Number of Old Processes

'Maximum number of old processes' specifies the maximum number of ended processes the system retains. This auto-cleanup option deletes succeeded and ended processes started by the OIEP when the limitation is exceeded. The oldest processes are deleted first.

By default, the maximum number of old processes is set to 1000.

Maximum Age of Old Processes

'Maximum age of old processes' specifies the maximum age of succeeded and ended processes. Use the following case-sensitive notation: y = years, M = months, w = weeks, d = days, h=hours, m = minutes, and s = seconds.

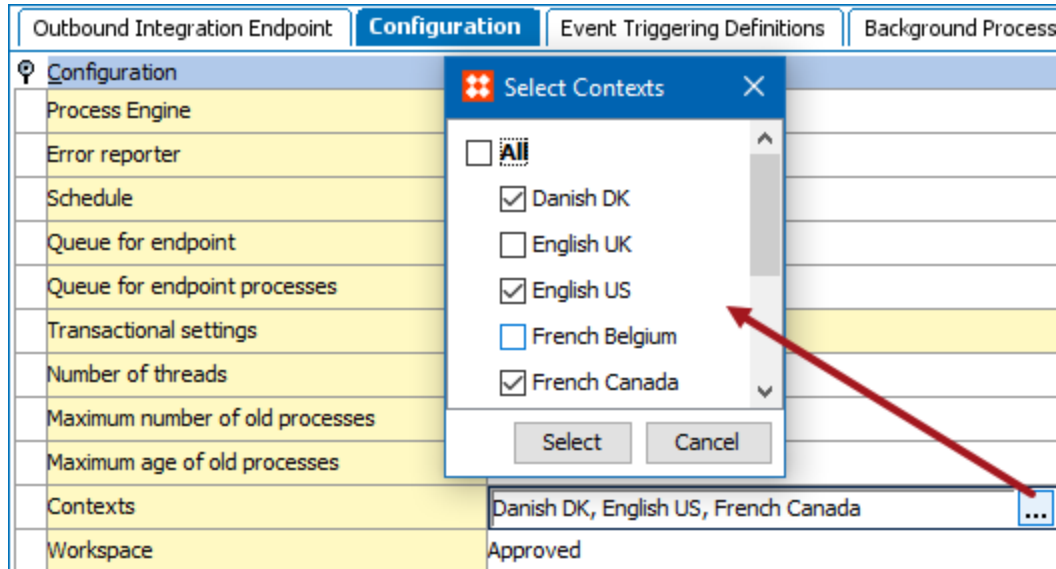
By default, the maximum age of old processes is set to one year.

Contexts

When exporting data in the STEPXML format or when using a custom cross-context enabled format, one file can contain data from multiple contexts. For other formats, the standard context splitter post-processor plugin should be used. Using this plugin, separate files are generated for the selected contexts. For more information, see the **OIEP - Post-Processor - Context Splitter** documentation.

By default , the context is set to the one which is currently being used while creating the OIEP .

Changing Selected Contexts



1. Click the Contexts field to display an ellipsis button.
2. Click the ellipsis button (...) to display the 'Select Contexts' dialog.
3. Choose the desired contexts and click the **Select** button to save.

Workspace

Workspace defines which data is used for the export. Common setup is to use the Approved workspace, except when you need to generate events for objects before they are approved, for example, during import or from a workflow.

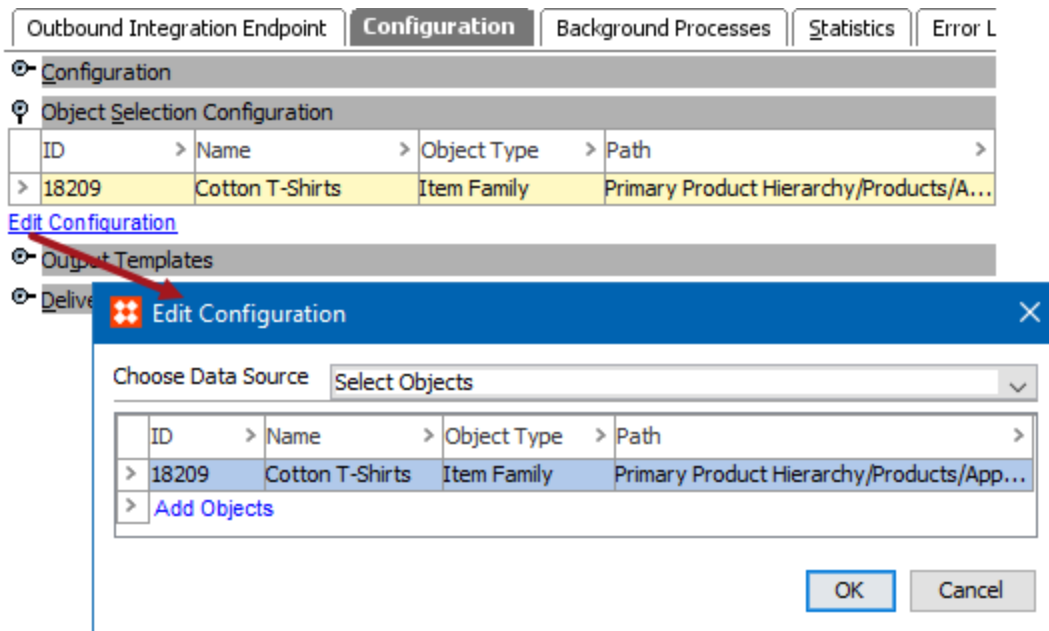
By default, the Approved workspace is selected.

OIEP - Select Objects - Object Selection Configuration Flipper

On the OIEP Configuration tab, the Object Selection Configuration flipper is only displayed when the selected OIEP uses the Select Objects data source, as defined on the Choose Data Source step of the wizard. For more information, see **OIEP - Select Objects - Choose Data Source**.

Note: Selecting a Tree hierarchy object is not required if the endpoint should export system-specific data, such as attributes, LOVs, and units that are not used in any specific product, classification, or entity hierarchy.

1. Click the Edit Configuration link to open the Edit Configuration dialog.

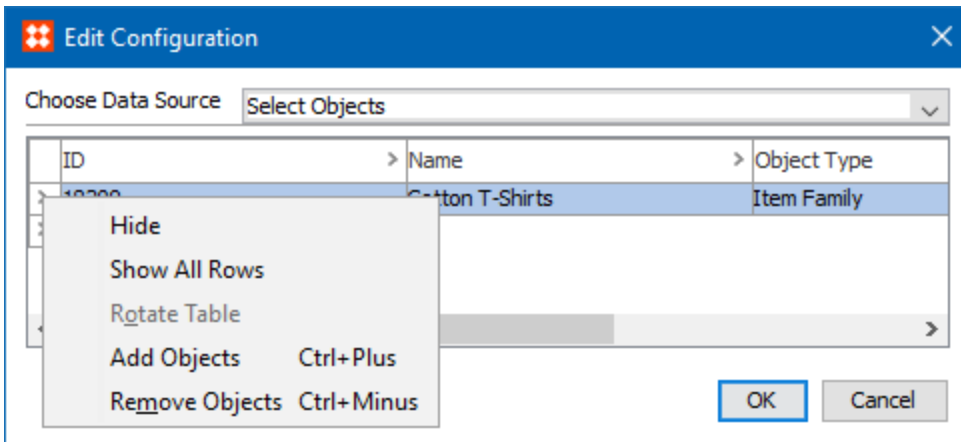


- Select an existing row and right-click to display additional options.

Hide and **Remove Objects** only affect the selected row.

Show All Rows and **Rotate Table** affect all rows.

Add Objects is the same option as accessed with the Add Objects link discussed below.



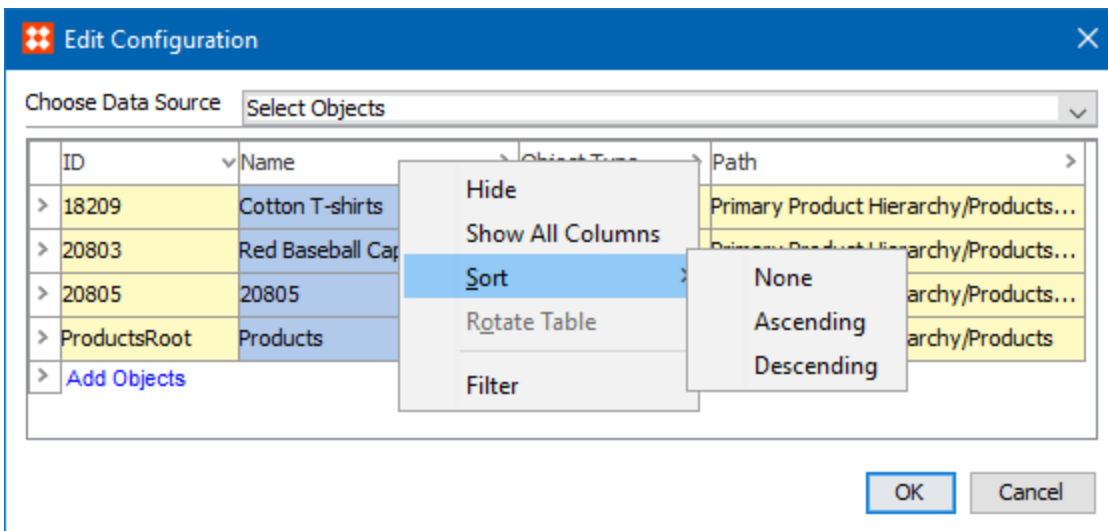
- Select an existing column and right-click to display additional options.

Hide only affects the selected column.

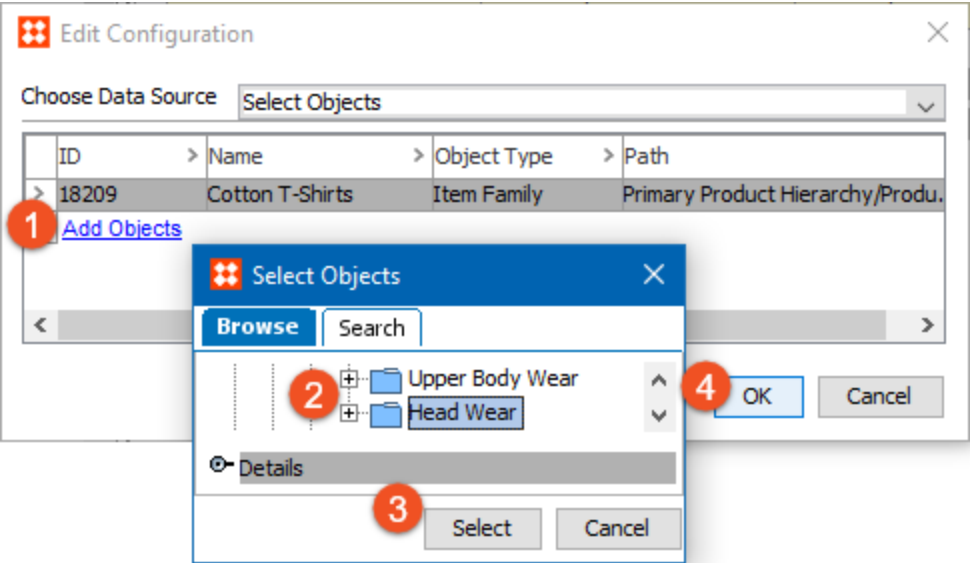
Show All Columns affects all columns.

Sort affects all rows and sorts based on the selected column.

Filter adds a filter option to each column.



- Click the **Add Objects** link to open the Select Objects dialog. Choose a new node and click the Select button. On the Edit Configuration dialog, click OK to include the newly selected data in the output.



OIEP - Select Objects - Output Templates Flipper

An output template defines the data to be exported and connects it to the desired format. This feature enables you to use a different output format for each selected object type if required.

At least one output template is required, but common setup is to use a single output template for a Select Objects OIEP since the selected data determines the template. Keep in mind that batching is restricted to a single template, so only use multiple templates when required.

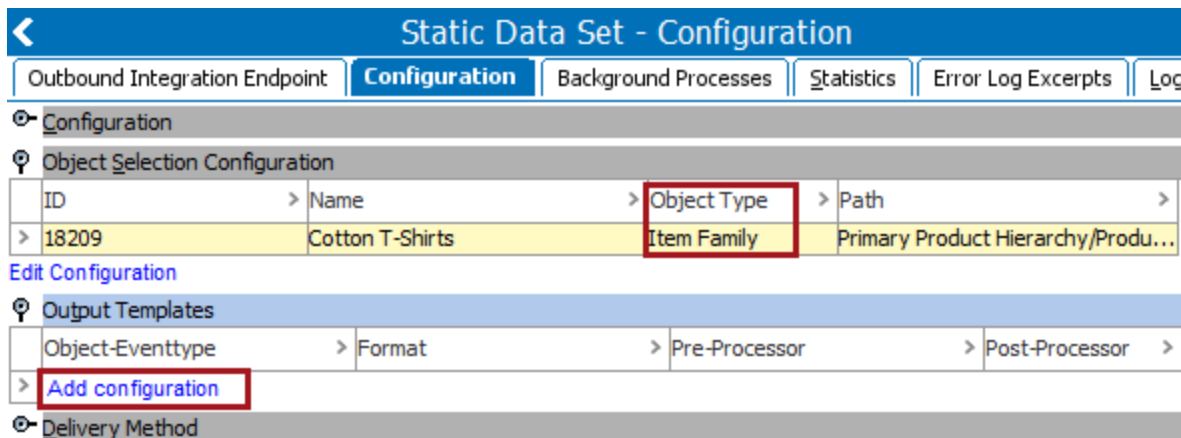
Configuring a select objects output template involves the following steps:

- Configure the object types
- Specify the format
- Configure preprocessor and postprocessor, based on availability on your system.

Configure the Object Types

The object type selection determines the STEP data that will be exported.

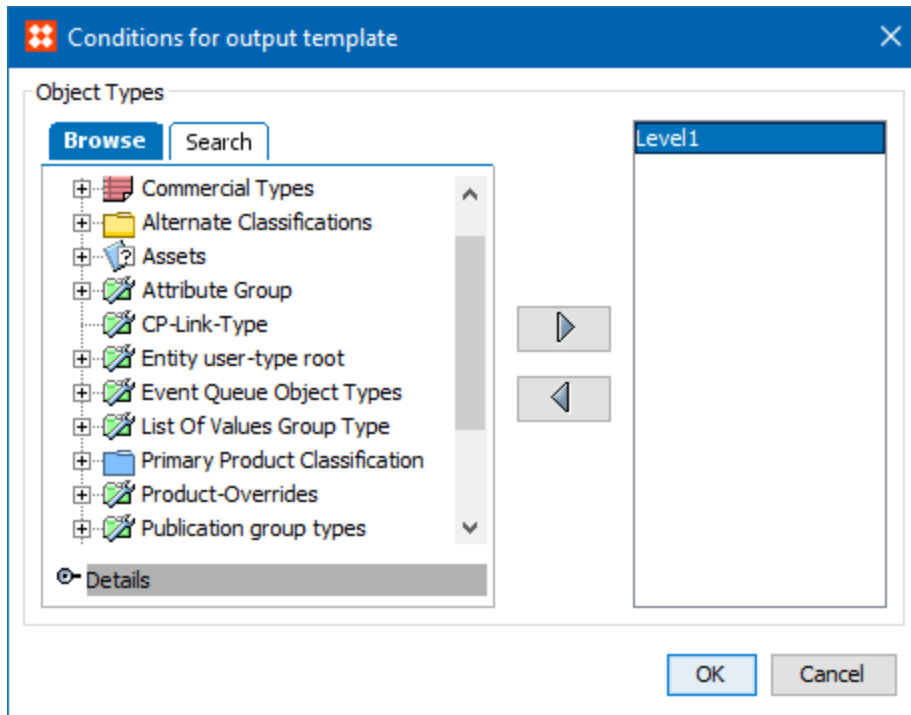
1. In System Setup, select the relevant OIEP > Configuration tab > Output Templates flipper > click the **Add configuration** link to display the 'Conditions for output template' dialog.



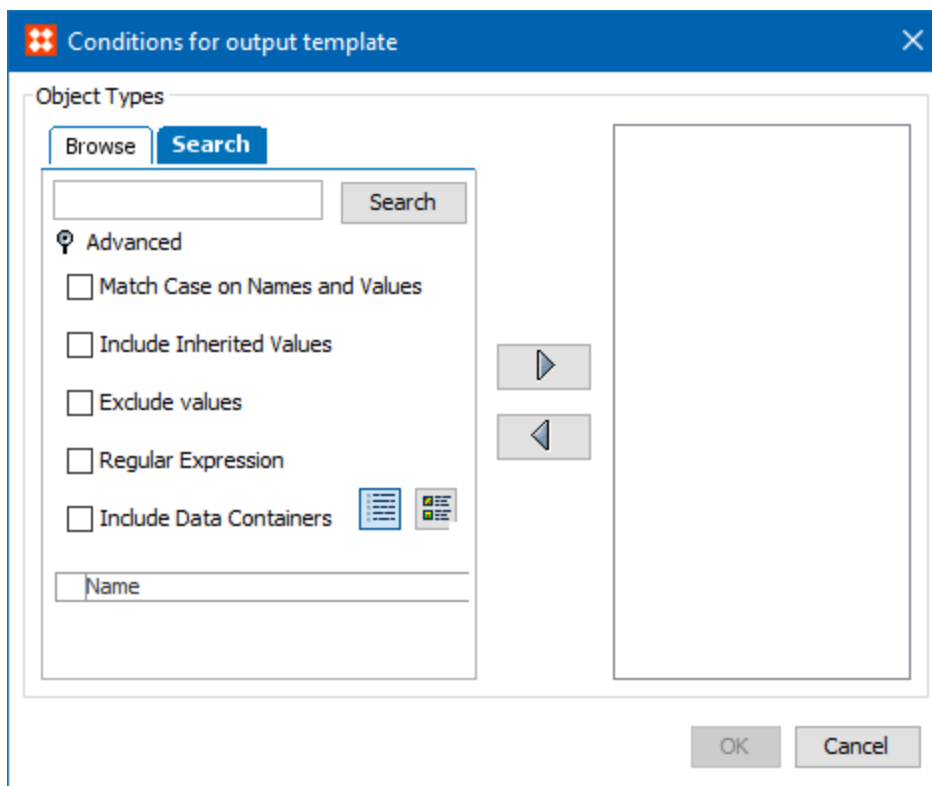
2. In the 'Conditions for output template' dialog, select object types that correspond at the root level to the objects selected in the wizard **OIEP - Select Objects - Choose Data Source** step, which is displayed in the Object Selection Configuration flipper under the Object Type column.

If Collections were selected in the wizard, in the 'Conditions for output template' dialog, add the relevant object types of the objects which are part of the collection.

Note: Selecting an output template object defines the object types for export, but does not necessarily define the objects that are included in the actual message, since the format of the template may dictate that parent, child, or referenced objects are included. Additionally, the advanced settings for format determine if the selected objects, their children, or both should be considered for the export.



On the Search tab, open the Advanced flipper to display additional search options.



Click the **OK** button.

- Select Objects endpoints can have only one template per object type. Repeat the previous steps until all required formats are displayed, keeping in mind the limitations of multiple output templates defined in the next section. Event type is not valid for this endpoint.

Output Templates				
Object-Eventtype	Format	Pre-Processor	Post-Processor	
> Level1	STEPXML	None	None	
> Level2	STEPXML	None	None	
> Add configuration				

- To edit existing Object-Eventtype settings, click into the desired field to display an ellipsis button (...). Click the ellipsis button (...) to display the 'Conditions for output template' dialog discussed above.

Using Multiple OIEPs or Multiple Output Templates

Since delivery method is per endpoint, not per output format, if you need a different output to go to each external system, you must use multiple OIEPs.

Multiple output templates can be created to allow data output based on specific object types. This prevents the same data from being exported across all chosen object types. Multiple output templates also help to reduce data output, which increases performance both for STEP and for downstream systems. In addition, the use of multiple output templates should reduce the need for customizations.

Limitations of Multiple Output Templates

- Output files are only delivered to one destination, regardless of the number of output templates.
- Output templates encompassing all object types are only available if all object types are chosen when making the configuration. This prevents excess output from the endpoints.
- Batching occurs per template type.
- Different output formats cannot be defined for the same object type. Common setup is to avoid combining different output formats (e.g., STEPXML and CSV) in the same endpoint.

Configure the Format

The format selection determines how the STEP data is formatted (XML or tabular).

- Click into the format column to display the ellipsis button (...). Click the ellipsis button (...) to display the 'Select format' dialog.

Output Templates				
Object-Eventtype	Format	Pre-Processor	Post-Processor	
> Item, Item Family, ...	STEPXML	None	Context splitter	
> Level 4	STEPXML	None	None	
> Open Sales Item	STEPXML	None	None	
> Add configuration				

2. In the **Select format** window, on the **Format** tab, select the desired format from dropdown. These are the same options as in the Export Manager, based on your licenses. Use the following links for additional information:

- Advanced STEPXML Format
- Alphabetical Index - XML Format
- Ariba CIF Format
- Ariba CIF 3.0 Format
- BMEcat Format
- BMEcat 2005 Format
- CSV Format
- cXML Format
- Excel Format
- Excel Smartsheet Format
- Flatplanner Format
- Generic XML Format
- IDoc MATMAS05 Format
- Publication Excel Format
- STEPXML Format
- xCBL Format

Select format

Format | Mapping | Advanced

STEPXML

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

-Global Settings-

Export Data for Selected Contexts No

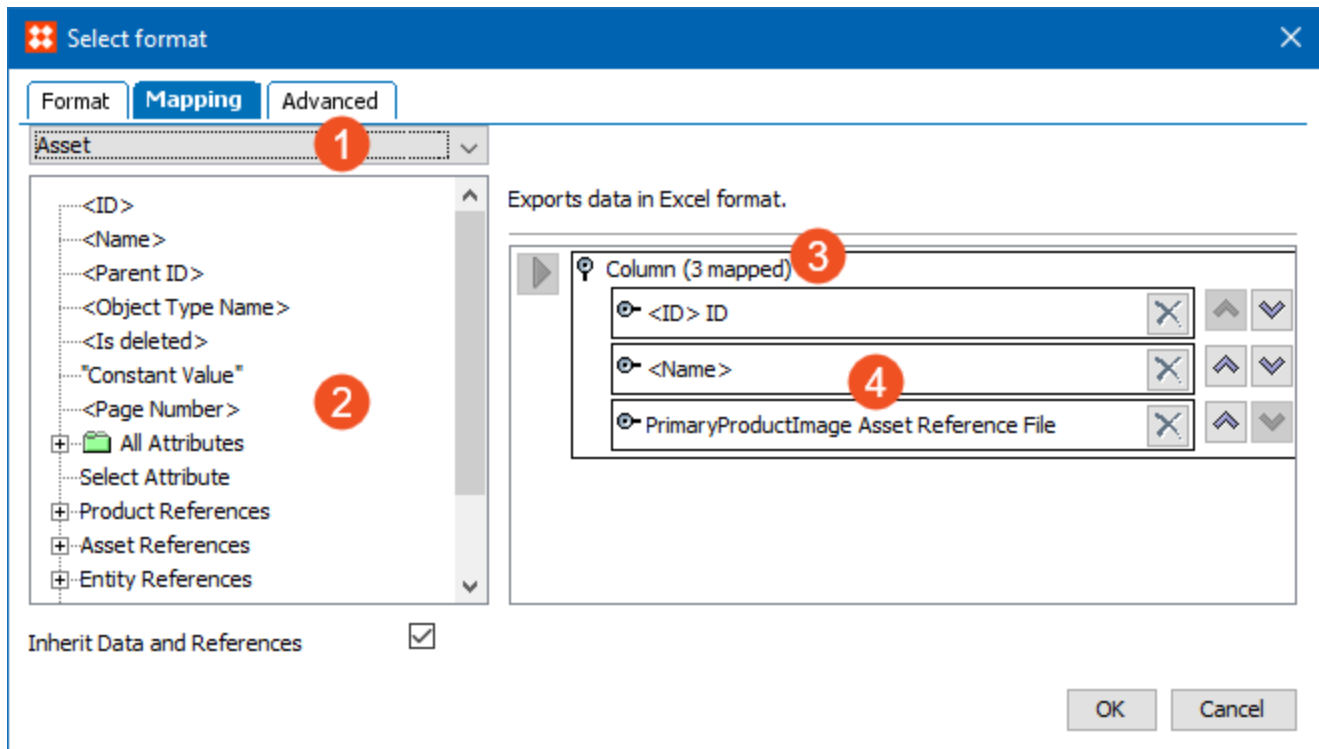
Include Schema Reference No

-Data Objects-

OK Cancel

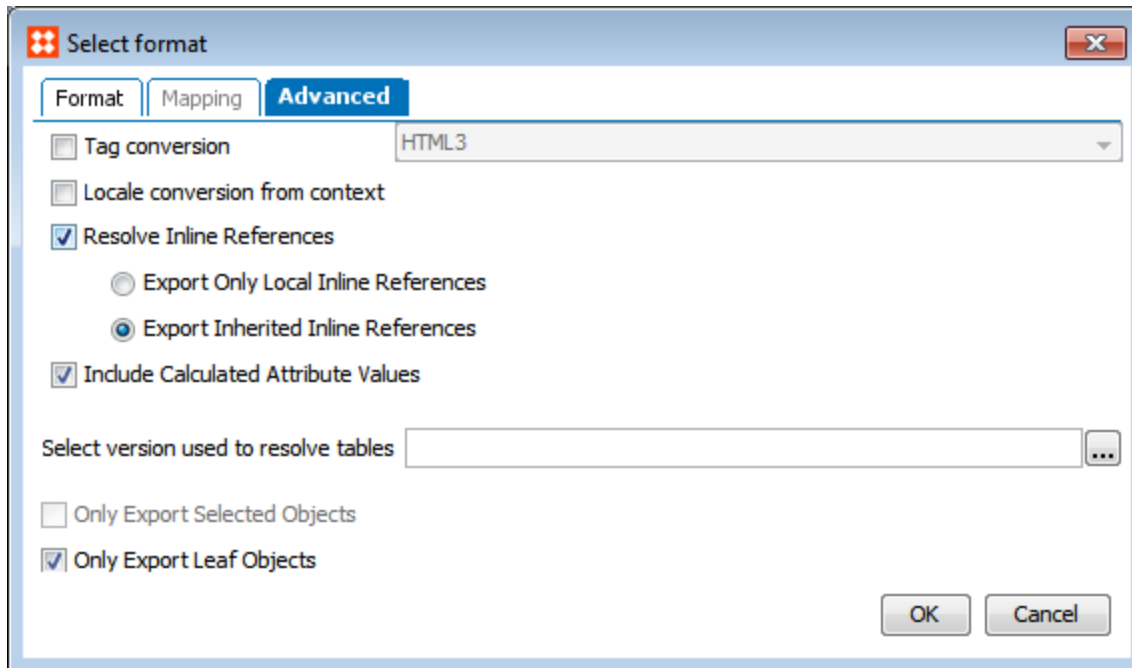
3. When available, specify how to map data to the export format, including the elements identified in the image below: object super type dropdown, data source, mapping target, and mapping rules.

For more information about the object super type dropdown, see the **Export Manager - Select Objects** topic.
For more information about the other elements, see the **Export Manager - Map Data** topic.



Note: The **Mapping** tab is enabled when the format selected requires mapping. Selected formats, including STEPXML and Advanced STEPXML, are automatically mapped so the Mapping tab is disabled.

4. On the **Advanced** tab, a number of advanced export options can be specified.



- **Tag conversion:** Converts tags to match the selected output format. This setting is optional.

If this box is left unchecked, any formatting tags that are included in attribute values will not be converted in the outbound message. For example, if an attribute value contains bold text (which must be made bold with a STEP Style Tag, e.g., <bold>/</bold>), this tag will not be converted to its corresponding HTML output format (e.g., /) in the export. For more information on STEP Tags, refer to System Setup > **Tags** in the online help.

- **Locale conversion from context:** Converts numbers into the numeric format that corresponds to the selected locale. If Smartsheets are used for format, this must remain unchecked. By default, the option is not checked.

- **Resolve Inline References:** Resolves inline references when they are exported. This box is checked by default, along with the option 'Export Inherited Inline References.'

If unchecked, inline references will be exported with the inline reference tagging instead of the actual content pulled in by the inline reference. For example, if an attribute called 'Product Number' has an inline reference to pull in the STEP ID of the object (e.g. 12345), the attribute value will not contain 12345. Instead, it will contain tagging similar to the following:

```
<ref attrid="" equalsign="" includeattrname="false" resolveto="objid"
separator=";" />
```

This setting would be valid if the user planned to re-import the file at a later time and not overwrite the inline reference with a static value. For more information on inline references, see **Inline References in Attribute Values** in the **Getting Started / STEP User Guide**.

- **Include Calculated Attribute Values:** Resolves calculated attribute values when they are exported. By default, the option is checked. If not checked, calculated attribute values are exported with empty values, unless the Force Calculation option has been set. The Force Calculation option is available when mapping a

calculated attribute using the Select Attribute data source, and also as a transformation. For more information, see **Force Calculation** in **Attribute - Data Source Outbound** and **Outbound Map Data - Transform**.

The value template is exported for each selected context, including a qualifier ID, which makes it possible to import the same data back into STEP. For information about calculated attribute values, see the **Calculated Attributes** topic in **System Setup / Super User Guide** documentation.

Important: If many complex calculated attribute values (traversing hierarchies and/or references) are used, consider if they should be exported, since they can impact performance. If required, consider scheduling for non-peak times. Simple calculations are not detrimental to an export, regardless of the quantity.

- **Select version used to resolve tables:** This setting is only available if the format is STEPXML and 'Include Tables' is set to Yes. A publication version should be selected if the tables contain content relevant to a particular publication (such as column or row types that are only valid for certain publication types) or a publication version (such as commercial data). If a publication version is not selected, the version last used in the STEP workbench is used. For more information, see the **Exporting Resolved Tables** topic in the **Tables** documentation.
- **Only Export Selected Objects:** Specifies that only objects from the output template are exported. No children are exported. This setting is unchecked by default and is only available if 'Only Export Leaf Objects' is unchecked. This setting does not apply to STEPXML.
- **Only Export Leaf Objects:** Specifies that only the leaf objects (lowest level of the export object for both events and selected objects) of the selected top hierarchy are included in the export. Selected objects and triggering objects are only included if they have no children. This setting applies to the CSV or Excel format.

Note: Uncheck both 'Only Export Selected Objects' and 'Only Export Leaf Objects' if all objects (both for triggering events and object selection) and their children should be included in the export.

For more information, see **Export Manager - Advanced** documentation.

Configure the Pre-processor and Post-processor

Depending on your system setup, preprocessor and postprocessor options may be available.

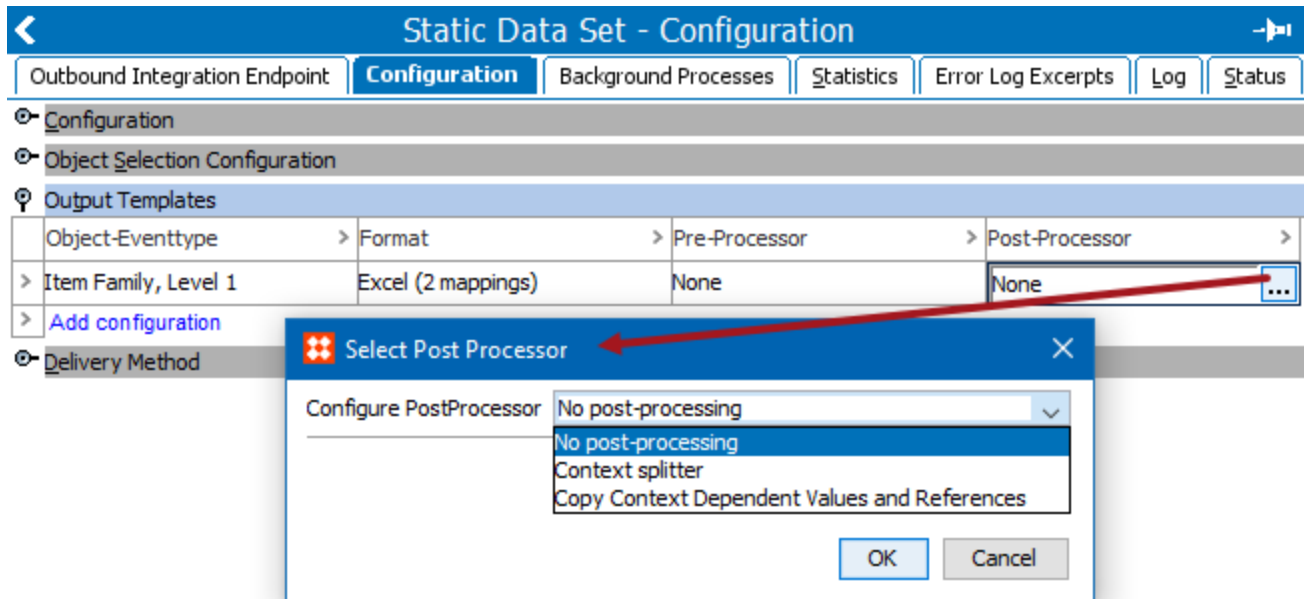
Pre-processor

Although a business action preprocessor can be set up for a select objects endpoint, it is not recognized and will not cause any changes to the data being processed.

Post-processor

Standard postprocessing options should be evaluated if multiple contexts are included in the output. You can configure a post-processor for each of the output templates.

1. Click into the Post-Processor column to display the ellipsis button (...). Click the ellipsis button (...) to display the 'Select Post Processor' dialog.



2. In the **Configure Post-processor**, select from the following options:

- **No post-processing** exports files using the standard export functionality.
- **Context splitter** generates an export file for each configured context. Each exported file will only contain context specific data. For more information, see the **OIEP - Post-Processor - Context Splitter** topic.

For the **Copy inherited product values** parameter, select **Yes** to copy and save inherited product values to the child product; or select **No** to use inherited product values in the export, but leave the child product unmodified.

When using Excel or CSV format, and multiple contexts are configured for export, you must select Context Splitter, or a single context will be included in the export.

- **Copy Context Dependent Values and References** to export context-dependent values and references and add the corresponding ContextID attribute to the value or reference in question. The endpoint generates one file containing values and references for each context specified. For more information, see the **OIEP Post-Processor - Copy Context Dependent Values and References** section.

For the **Copy inherited product values** parameter, select **Yes** to copy and save inherited product values to the child product; or select **No** to use inherited product values in the export, but leave the child product unmodified.

If you use the **Copy Context Dependent Values and References** post-processor to add ContextIDs to a cross-context export (exporting data for selected contexts), the downstream system can use the <ContextID> tag to identify specific contexts. Additionally, this facilitates re-importing the exported data into the correct contexts in STEP after processing by a third-party service or application.

OIEP - Pre-Processor - Business Action

By default, no pre-processor is included in an OIEP output template. Applying the Business Action Pre-Processor means a business rule can be run against the data, prior to the export. For example, a node can be added to a current batch, or an event can be removed from the current batch before a message is created for an export.

Using this pre-processor requires:

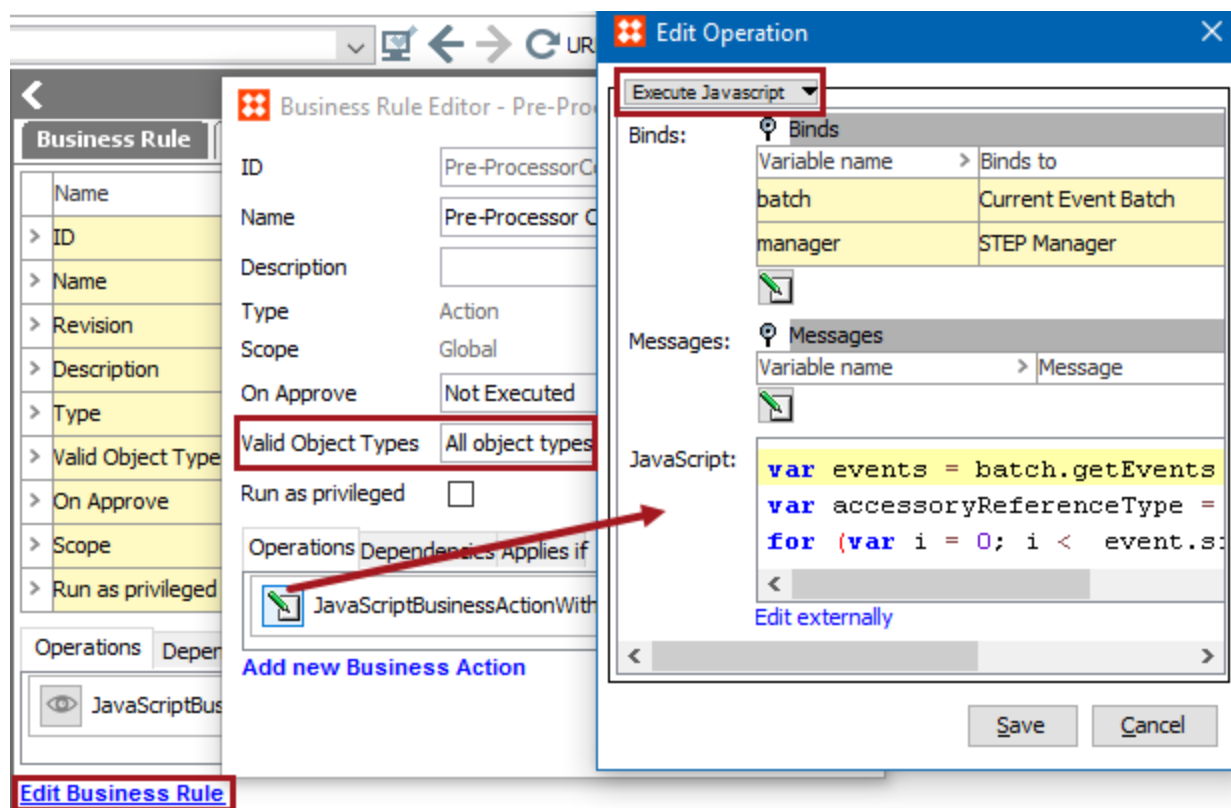
1. A business action - in this example, it will add referenced products to the products that are contained in the message.
2. An event-based OIEP configured with the business action - in this example, it is configured to deliver messages that inform a downstream system about changes to a product.

Business Action Configuration

Create a business action that is valid for 'All Object Types' (or specifically for the object types being handled by the event queue). For more information on creating business rules, see the **Specifying a Business Action** topic.

For a full JavaScript code example, see the online version of this topic.

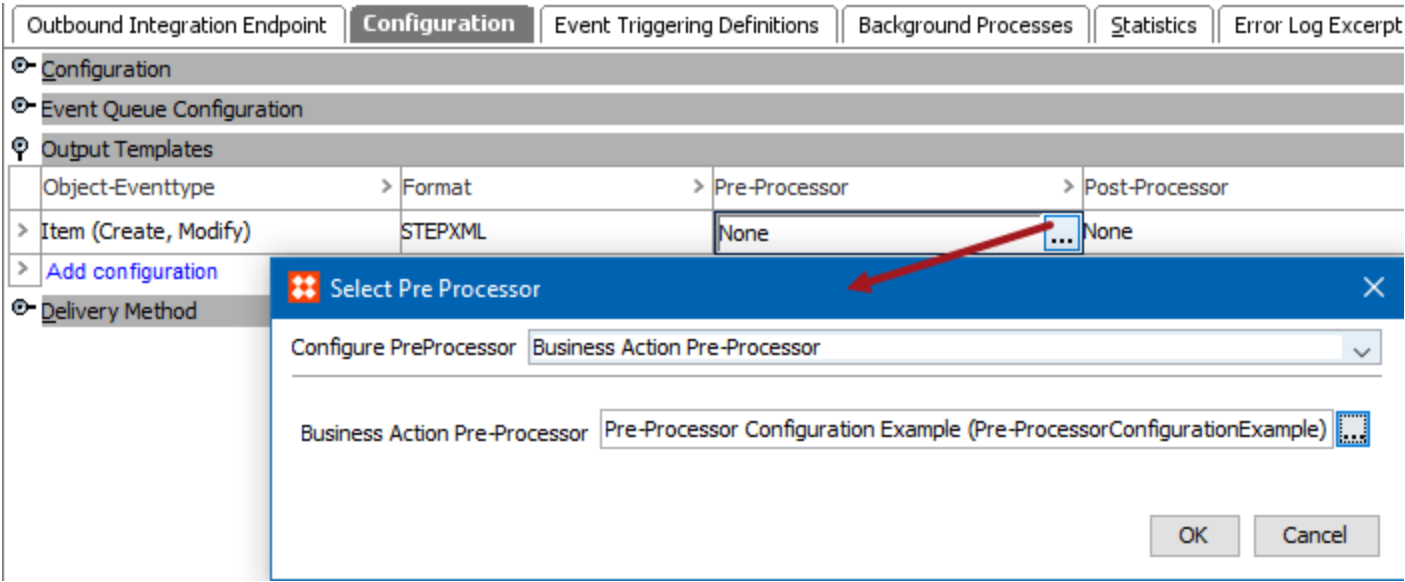
The business rule is configured in System Setup > Global Business Rules as displayed below:



OIEP Configuration

Apply the business action as a pre-processor for any of the output templates on an OIEP.

- 1. Create an event-based OIEP. For details, see the **Creating an Event-Based Outbound Integration Endpoint** documentation.
- 2. On the **Configuration** tab, open the **Output Templates** flipper and locate the output template that requires the business action.



- 3. In the **Pre-Processor** column, click the ellipsis button (...) to display the Select Pre Processor dialog.
- 4. In **Configure PreProcessor**, select the **Business Action Pre-Processor** option.
- 5. In **Business Action Pre-Processor**, select the business action created earlier. The output template is updated to show the pre-processor will run.

Output Templates			
Object-Eventtype	Format	Pre-Processor	Post-Processor
Item (Create, Modify)	STEPXML	Business Action Pre-Processor	None
Add configuration			

OIEP - Post-Processor - Context Splitter

The Context Splitter post processor splits the export file into individual files based on the contexts on which the data is dependent. This can be used when the data from different contexts should be sent to the external systems separately. It is also commonly used for reporting purposes for individual contexts.

The context splitter adds the <ContextID> without a <Qualifier ID>. This makes the file similar to an export of local values done from a single context. The <ContextID> tag specifies the ID of the context in which the object is exported, but does not provide the specific dimension point details which can only be supplied by the <QualifierID> tag. Using the context splitter assumes that the external system does not have any details of dependency points from STEP.

The following example illustrates using an outbound integration endpoint with the Context Splitter post-processor. It includes attribute values, but the post-processor also works with inherited references such as classification or asset references.

The product has a language-dependent attribute, 'Description, Long', with values defined in three contexts: English US, Danish DK, and French FR.

<

18217-0542 rev.0.30 - Compare Contexts

100% complete

→

Compare Contexts

References

Referenced By

View: Description, Long

	> English US	> Danish DK	> French FR
> ID	100305	100305	100305
> Name	18217-0542	18217-0542	18217-0542
> Description, Long	The best-selling T-Shirt For over 35 years, it has set the standard for T-shirt comfort and quality. Today it's better than ever, offering greater durability and less shrinkage than you'll get with ordinary tees.	Den bedst sælgende T-shirt I over 35 år har den sat standarden for T-shirt komfort og kvalitet. I dag er det bedre end nogensinde, hvilket giver større holdbarhed og mindre krympning, end du får med almindelige tees.	Le T-Shirt le plus vendu Depuis plus de 35 ans, il a établi la norme pour le confort et la qualité du T-shirt. Aujourd'hui, c'est mieux que jamais, offrant une plus grande durabilité et moins de retrait que vous ne le verrez avec les T ordinaires.

XML Output Example – Context Splitter

Using the context splitter, a separate file is generated for each context. As shown below, the names of the files output identify the Context IDs from which the objects were exported.

Name	Type
 Context1##exported_0.xml	XML File
 Context2##exported_0.xml	XML File
 Context6##exported_0.xml	XML File

The <ContextID> tag has been added to the export. The files can be distinguished by the Context ID and there is no <Qualifier ID> or any information about the dimension points. The XML output for each of the files is:

```

19 <STEP-ProductInformation ExportTime="2017-08-07 11:51:18" ExportContext="Context6" ContextID="Context1" WorkspaceID="Main" Us
20 <Products>
21 <Product ID="100305" UserTypeID="SalesItem" ParentID="18206">
22 <Name>18217-0542</Name>
23 <Values>
24 <Value AttributeID="DescriptionLong">The best-selling T-Shirt For over 35 years, it has set the standard for T-shirt

```

```

19 <STEP-ProductInformation ExportTime="2017-08-07 11:51:18" ExportContext="Context6" ContextID="Context2" WorkspaceID="Main" Us
20 <Products>
21 <Product ID="100305" UserTypeID="SalesItem" ParentID="18206">
22 <Name>18217-0542</Name>
23 <Values>
24 <Value AttributeID="DescriptionLong">Le T-Shirt le plus vendu Depuis plus de 35 ans, il a établi la norme pour le con

```

```

19 <STEP-ProductInformation ExportTime="2017-08-07 11:51:18" ExportContext="Context6" ContextID="Context6" WorkspaceID="Main" Us
20 <Products>
21 <Product ID="100305" UserTypeID="SalesItem" ParentID="18206">
22 <Name>18217-0542</Name>
23 <Values>
24 <Value AttributeID="DescriptionLong">Den bedst sælgende T-shirt I over 35 år har den sat standarden for T-shirt komfo

```

XML Output Example - Standard Cross-Context Export

Using a standard cross-context export (exporting data for selected contexts) without the context splitter, the single output file includes only the attribute values specific to particular dimension points (indicated by QualifierIDs). The XML output is:

```

129 <ValueGroup AttributeID="DescriptionLong">
130 <Value QualifierID="en-US">The best-selling T-Shirt For over 35 years, it has set the stan
131 <Value QualifierID="fr">Le T-Shirt le plus vendu Depuis plus de 35 ans, il a établi la nor
132 <Value QualifierID="Danish">Den bedst sælgende T-shirt I over 35 år har den sat standarden
133 </ValueGroup>

```


OIEP - Post-Processor - Copy Context Dependent Values and References

This post processor reduces the complexity of inheritance of values in cross-context exports, which allows downstream systems to receive a multi-context export in one single file.

Important: Using the 'Copy Context Dependent Values and References' post processor requires a strong understanding of the STEP Qualifier concept that defines in which dimensions and at which inheritance levels values are present.

If you want to further process the information in a form similar to a typical single-context export, we strongly recommend using the context-splitter post processor instead. For more information, see the **OIEP - Post-Processor - Context Splitter** topic in the **Outbound Integration Endpoints** documentation.

ContextID and QualifierID XML Tags

The 'Copy Context Dependent Values and References' post processor adds the <ContextID> and <QualifierID> XML tags to the export. A standard cross-context export (exporting data for selected contexts) only adds the <QualifierID>. The <QualifierID> tag specifies the ID of the Dimension Point in which the attribute value exists but does *not* provide the specific context. For example, the <QualifierID> of 'std.lang.all' could be the ID of the 'Language Root' Dimension Point (which would typically only be used in a Global context). But, without the addition of the <ContextID> to the export, the additional contexts that an attribute value has inherited down to cannot be determined.

When the <ContextID> XML tag is used in combination with the <QualifierID> XML tag, it is possible for the downstream system to determine the inheritance relationship between a value or reference and a context. This is useful when a service tries to re-import processed data into STEP. Because the downstream system knows the context the data must be imported into, it is easier to target specific contexts within STEP for import.

This example illustrates the differences between using the Export Manager for a standard cross-context export (exporting data for selected contexts) and using an outbound integration endpoint with the **Copy Context Dependent Values and References** post-processor enabled.

In the following, attribute values are used, but the post-processor also works with inherited references such as classification or asset references.

The product has a language-dependent attribute, 'Body Copy', with values defined in two different contexts: 'GL (Global)' and 'FR All All (French)'.

Global context attribute value:

Name	Value
Body Copy	abc Stylish, modern and lightweight, ideal for use in cafes, bars, bistros and clubs.

French context attribute value:

Name	Value
Body Copy	abc Élégantes, modernes et légères, idéales pour les cafés, les bars, les bistros ou les discothèques. Assortis aux

A third context, 'DE All All (German)', inherits its attribute values from the Global context. This is because no values have been entered in the DE All All context yet, and DE All All has an inheritance relationship with the GL context:

German context attribute value:

Name	Value
Body Copy	Stylish, modern and lightweight, ideal for use in cafes, bars, bistros and clubs.

XML Output Example - Standard Cross-Context Export

Using a standard cross-context export (exporting data for selected contexts), the XML looks like the following:

```

1 <Product ID="CHAIR" UserTypeID="Product Folder" ParentID="MyChairFolder">
2   <Name QualifierID="std.lang.all">SitWell Armchair</Name>
3   <Values>
4     <ValueGroup AttributeID="Body Copy">
5       <Value QualifierID="std.lang.all">Stylish, modern and lightweight, ideal for use in
6       <Value QualifierID="fre">Elégantes, modernes et légères, idéales pour les cafés, le
7     </ValueGroup>
8   </Values>
9 </Product>

```

Only the attribute values specific to particular dimension points—indicated by QualifierIDs—are exported, and only the attribute values that are different from the Global context. Global values are taken from 'std.lang.all' and the French values are taken from 'fre'. Since the 'Name' attribute is inherited from the 'std.lang.all' qualifier, no specific French or German 'Name' is exported. Because standard export methods collapse duplicate values, only the top value of the inheritance tree is exported.

XML Output Example - with Copy Context Dependent Values and References post processor

The following shows an XML output of an OIEP cross-context export (exporting data for selected contexts) with the **Copy Context Dependent Values and References** post-processor enabled:

```

1 <Product ID="CHAIR" UserTypeID="Product Folder" ParentID="MyChairFolder">
2   <Name ContextID="GL" QualifierID="std.lang.all">SitWell Armchair</Name>
3   <Name ContextID="FR All All" QualifierID="std.lang.all">SitWell Armchair</Name>
4   <Name ContextID="DE All All" QualifierID="std.lang.all">SitWell Armchair</Name>
5   <Values>
6     <Value AttributeID="Body Copy" ContextID="GL" QualifierID="std.lang.all">Stylish, modern and lightwei
7     <Value AttributeID="Body Copy" ContextID="FR All All" QualifierID="fre">Elégantes, modernes et légèr
8     <Value AttributeID="Body Copy" ContextID="DE All All" QualifierID="std.lang.all">Stylish, modern and
9   </Values>
10 </Product>

```

The <ContextID> tag has been added to the export along with the <QualifierID> tag, and each value—including inherited values—is explicitly exported. This means that the German values are included in the export, even though the DE All All context inherits from the GL context. Once the value is overwritten in the German context, the <QualifierID> would change to the ID of the German language dimension point.

OIEP - Delivery Method Flipper

The Delivery Method flipper in an OIEP includes the same parameters for both Event-Based and Select Objects endpoints. The delivery method controls how the output is delivered to an external system.

Open the Delivery Method flipper to display the selected delivery method. As shown below, the parameters vary based on the selected delivery method. Each is described in the **OIEP Delivery Methods** topic.

Outbound Integration Endpoint
Configuration
Event Triggering

Configuration
Event Queue Configuration
Output Templates
Delivery Method

Copy to directory

> Directory	MultiValuePostProcessor
> File Name Template	\$filename-\$timestamp.\$extension
> Zip content	No
> Edit Delivery	

Amazon SQS

For details on this delivery method, see **Amazon SQS Delivery Method**.

Amazon SQS

> Server URL	https://sqs.us-east-2.amazonaws.com
> Credentials path	C:\Users\stibosw\aws\credentials
> Credentials profile	default
> Queue Name	myQueue
> Zip export file	Yes
> Edit Delivery	

Copy to Directory

For details on this delivery method, see **Copy to Directory Delivery Method**.

Copy to directory

> Directory	OUT
> File Name Template	\$filename-\$timestamp.\$extension
> Zip content	Yes
> Edit Delivery	

Deploy

For details on this delivery method, see **Deploy Delivery Method**.

Deploy

> Extract to directory	workarea
> Copy to file	\$tmpdir/zip-\$timestamp.\$extension
> Edit Delivery	

Email

For details on this delivery method, see **Email Delivery Method**.

Email

> Email	OutboundData@customer.com
> Subject text	Product data export
> Body text	Please find enclosed the latest product data delivery
> Zip export file	Yes
> Edit Delivery	

FTP

For details on this delivery method, see **FTP Delivery Method**.

FTP

> Hostname	FTPDeliveryHostServerName
> Username	OIEPUser
> Password	xxxxxxxx
> File Name Template	\$filename-\$timestamp.\$extension
> Zip before upload	Yes
> Edit Delivery	

IBM Websphere MQ SSL Delivery

For details on this delivery method, see **IBM Websphere MQ SSL Delivery Method**.

IBM Websphere MQ SSL Delivery

>	Connection URL	
>	Queue Manager	
>	Queue Name	
>	Binary Payload	No
>	User Name	
>	Password	
>	Key Store	
>	Trust Store	
>	Cipher Suite	
>	Additional Parameters	
>	Edit Delivery	

JDBC

For details on this delivery method, see **JDBC Delivery Method**.

JDBC

>	Driver Location	L:/jdbc-drivers/mysql-connector-java-5.0.8-bin.jar
>	Driver Class	com.mysql.jdbc.Driver
>	Database URL	jdbc:mysql://localhost:3306/mydb
>	Username	root
>	Password	xxxx
>	Table Name	stepdata
>	Key Columns	id,datetime
>	Delete Key Columns	id
>	Edit Delivery	

JMS

For details on this delivery method, see **JMS Delivery Method**.

JMS Delivery

> JMS Connection Factory Class	ActiveMQInitialContextFactory
> Connection Factory Name	ConnectionFactory
> JMS Provider URL	tcp://ATTCM3S9:61616 =
> JMS Queue	JMSQueueForSTEP
> Binary Payload	No
> User Name	OIEPUser
> Password	xxxxxxxxxx
> Additional Parameters	
> Edit Delivery	

Mongo

For details on this delivery method, see **Mongo Delivery Method**.

Mongo delivery

> Server(s)	mongohost1:27017,mongohost1:27018,mongohost1:27019
> Raw DB prefix	raw
> User name	userauth
> Password	xxxxxxxxxxxxxx
> Cluster Enabled Configuration	Yes
> Use SSL to encrypt network traffic	No
> Key store location	
> Trust store location	
> Actions	WorkflowReview
> Edit Delivery	

Oracle AQ

For details on this delivery method, see **Oracle AQ Delivery Method**.

Oracle AQ Delivery

> Connection URL	jdbc:oracle:thin:@//server:1521/step
> Queue Name	OradeAQRecQueue
> Binary Payload	No
> User Name	OIEPUser
> Password	xxxxxxxxxx
> Edit Delivery	

REST

For details on this delivery method, see **REST Delivery Method**.

REST

> URL	http://www.Customer.com/sqlrest
> User name	OIEPUser
> Password	xxxxxxxxxx
> Zip content	Yes
> Edit Delivery	

REST Direct

For details on this delivery method, see **REST Direct Delivery Method**.

REST Direct

> URL	http://www.CustomerDirect.com/sqlrestdirect
> User name	OIEPUser
> Password	xxxxxxxxxx
> Zip content	Yes
> Header	
> Footer	
> Edit Delivery	

SFTP

For details on this delivery method, see **SFTP Delivery Method**.

SFTP

> Hostname	SFTPDeliveryHostServerName
> Username	OIEPUser
> Password	xxxxxxxxxx
> File Name Template	\$filename-\$timestamp.\$extension
> Zip before upload	Yes
> Edit Delivery	

Wiki

For details on this delivery method, see **Wiki Delivery Method**.

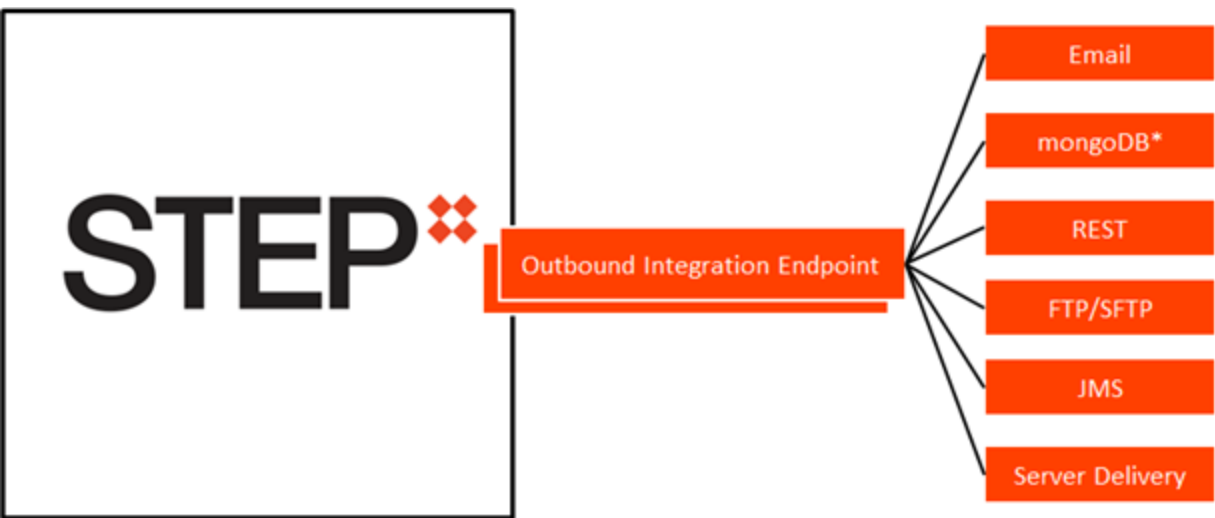
WikiDelivery

>	WikiMainSpace	STEP/
>	WikisName	wikis/xwiki/
>	WikiPassword	admin
>	WikiRestSpace	rest/
>	WikiUrl	http://vp487.vps.ohs.ca/
>	WikiUsername	admin
>	Edit Delivery	

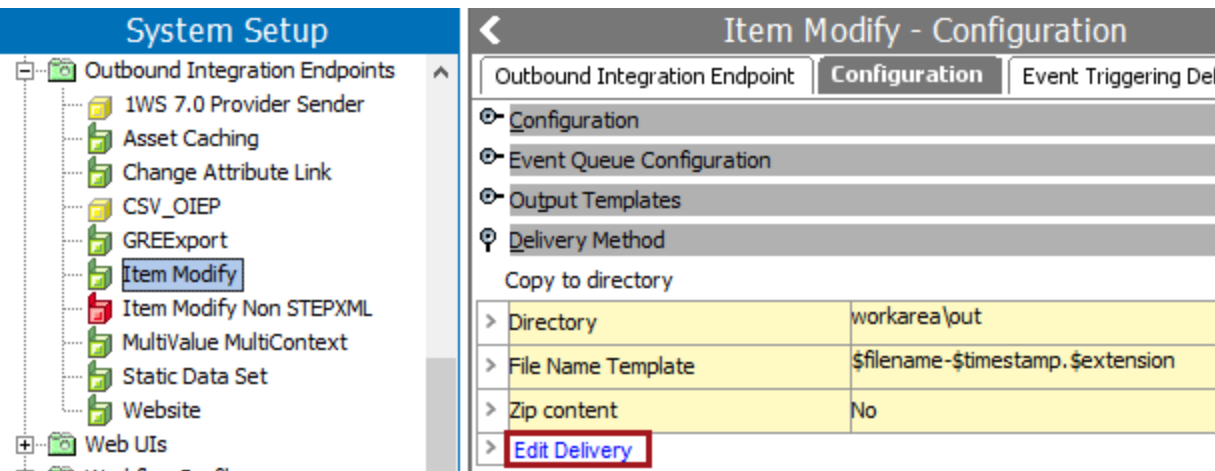
OIEP Delivery Methods

Delivery methods are used by both outbound data tools (Export Manager and OIEP) but the available options vary. For information on the delivery methods available in Export Manager, see **Export Manager - Select Delivery Method**.

An OIEP takes the data determined by its configuration and delivers it using the selected delivery method to the downstream system or on a directory on the application server.



In an OIEP, the delivery method is set in the editor on the Configuration tab, under the Delivery Method flipper.



The following delivery method options are available (although some are based on a license) when exporting using an OIEP.

Method	Description
Amazon SQS	Delivers messages to the Amazon SQS (Amazon Simple Queue Service). See Amazon SQS Delivery Method documentation.
Copy to Directory	Default selection for new OIEPs delivers files to a directory on the application server and allows the delivery to be zipped. See Copy to Directory Delivery Method documentation.
Deploy	Delivers files to a directory on the application server. Does not allow the delivery to be zipped. See Deploy Delivery Method documentation.
Email	Delivers files as an email attachment and allows a zipped delivery. See Email Delivery Method documentation.
FTP	Delivers files using file transfer protocol (FTP). See FTP Delivery Method documentation.
GDSN Data Pool	The GDSN solution must be implemented fully before the GDSN Data Pool method is available and functional. See GDSN Provider documentation.
GDSN Receiver Data Pool	The GDSN solution must be implemented fully before the GDSN Receiver Data Pool method is available and functional. See GDSN Receiver documentation.
IBM Websphere MQ SSL	IBM Websphere MQ SSL, uses MQ series Secure Sockets Layer, and enables data exchange across IBM and non-IBM platforms. See IBM Websphere MQ SSL Delivery Method documentation.
JDBC	The JDBC delivery plugin enables delivery of data to RDBMS-type databases like Oracle, MySQL, MS SQL Server, PostgreSQL, etc. See JDBC Delivery Method documentation.
JMS	The standard JMS delivery plugins (Apache Active MQ, Websphere MQ, and Oracle AQ) can be used to deliver messages to a JMS receiver without further customizations. See JMS Delivery Method documentation.
Mongo	Receives data from a STEP event queue and loads it into a MongoDB database. The MongoDB is often used for web site back-end, reporting, and high performance feeds to other back-end systems. See Mongo Delivery Method documentation.
Oracle AQ	Oracle Advanced Queuing (Oracle AQ) enables messages to be exchanged between two systems. See Oracle AQ Delivery Method documentation.
REST	The REST Delivery Method delivers a call-back URL to the REST service and does not include

Method	Description
	actual STEP data. See REST Delivery Method documentation.
REST Direct	The REST Direct delivery method differs from the standard REST delivery method in that the data is delivered directly to the REST service and no call-back URL is required. See REST Direct Delivery Method documentation.
SFTP	Delivers a file using the Secure File Transfer Protocol (SFTP). See SFTP Delivery Method documentation.
Wiki	Delivers metadata to the metadata wiki platform. See Wiki Delivery Method documentation.

Amazon SQS Delivery Method

The Amazon SQS Delivery Method delivers messages limited to 256KB to the Amazon Simple Queue Service (Amazon SQS). Common setup is to use file compression to meet the size limitation. This delivery option is only available in OIEPs and requires a separate installation recipe. Contact your Stibo representative for the latest recipe.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. For the 'Credentials profile' parameter, create a credentials file following the instructions at <http://docs.aws.amazon.com/cli/latest/userguide/cli-config-files.html>. Place the credentials file (named 'credentials', without any file extension) in the required path on the application server.
2. Prior to configuration, clicking the **Server URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **SQSDeliveryServerUrl** property. The following is an example of a property entry for a single server:

```
SQSDeliveryServerUrl=1=https://sqs.us-east-2.amazonaws.com
```

3. Prior to configuration, clicking the **Credentials Path** dropdown parameter displays the required property name. Provide a selection, including the name of the credentials file, for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **SQSDeliveryCredentialsPath** property. If necessary, use a comma and increment the number to separate multiple paths as shown in the example below.

```
SQSDeliveryCredentialsPath=1=C:\\Users\\stibosw\\.aws\\credentials,2=C:\\Users\\stibosw\\.aws\\credentials123
```

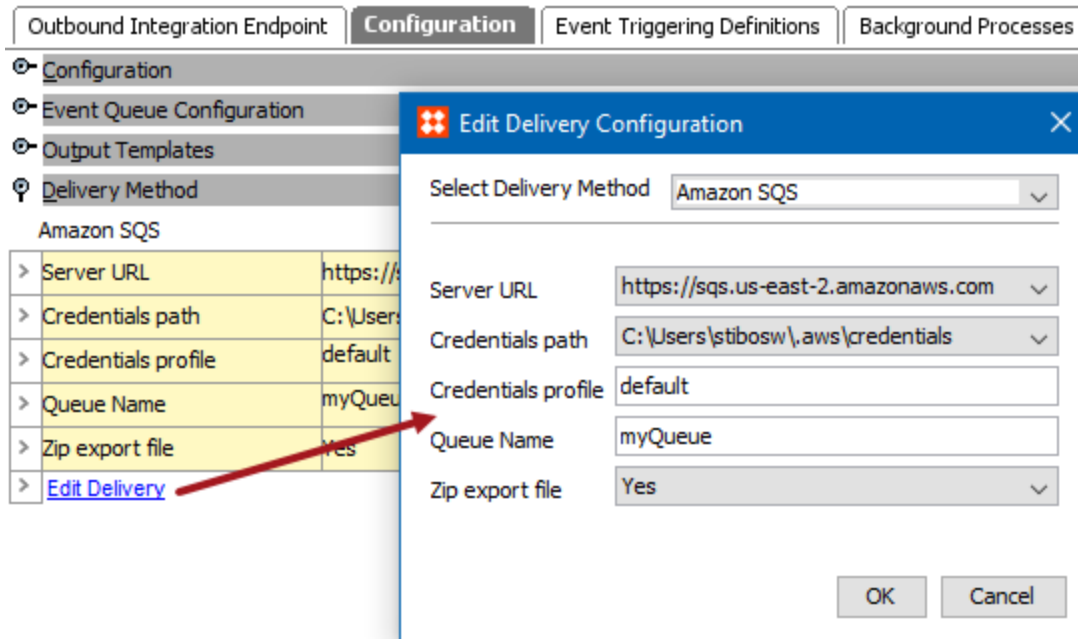
4. For the 'Queue name' parameter, if necessary, create a new queue on the SQS server. Use the steps defined in the **Amazon SQS Queue Configuration** section below.

Contact your Stibo Systems representative if you need assistance with setup.

Configuration

After completing the prerequisite steps, edit the delivery method of the OIEP. Use the following steps to configure the OIEP to use the Amazon SQS delivery option.

For information on a parameter, hover over the parameter field to display help text.

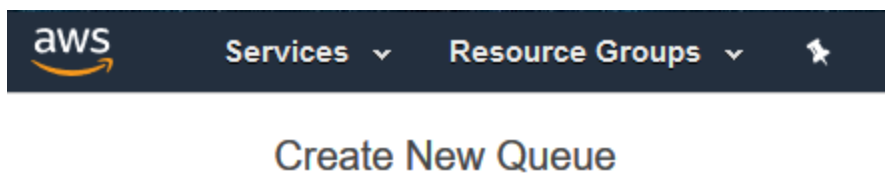


1. For **Select Delivery Method**, choose **Amazon SQS**.
2. In **Server URL**, select the URL that points to the SQS server where the data will be delivered.
3. In **Credentials path**, select the path to the desired SQS credentials file.
4. In **Credentials profile**, enter the profile name included in the SQS credentials profile.
5. In **Queue Name**, enter the Amazon queue created for this delivery.
6. In **Zip export file**, specify whether to zip the contents before delivery.
7. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Amazon SQS Queue Configuration

When required, use the following steps to create a new Amazon SQS queue.

1. Log in to the Amazon server based on your region.
2. Search for **Simple Queue Service** and select it to display the queue list page.
3. Click the **Create New Queue** button to display the Create New Queue page.



4. For **Queue Name**, type a queue name including the **.fifo** extension.
5. For the **type of queue** section, select **FIFO Queue**.

6. Click the **Configure Queue** button at the bottom of the page to display the default Queue Attributes parameters.
7. Check the **Content-Based Deduplication** parameter box to enable the queue.

Queue Settings

Default Visibility Timeout		30	seconds	Value must be between 0 seconds and 12 hours.
Message Retention Period		4	days	Value must be between 1 minute and 14 days.
Maximum Message Size		256	KB	Value must be between 1 and 256 KB.
Delivery Delay		0	seconds	Value must be between 0 seconds and 15 minutes.
Receive Message Wait Time		0	seconds	Value must be between 0 and 20 seconds.
Content-Based Deduplication  ☒				

8. Click the **Create Queue** button. The new queue is displayed in the list.

Copy to Directory Delivery Method

For OIEPs, there is no single delivery option for server side delivery, but the 'Deploy' and 'Copy to directory' options provide the same function in that they both deliver files to a directory on the application server. Common setup is to use 'Copy to directory' since it is more flexible because it allows the delivery to be zipped. Neither of these options is available in Export Manager.

For more information, see the **Deploy Delivery Method** documentation.

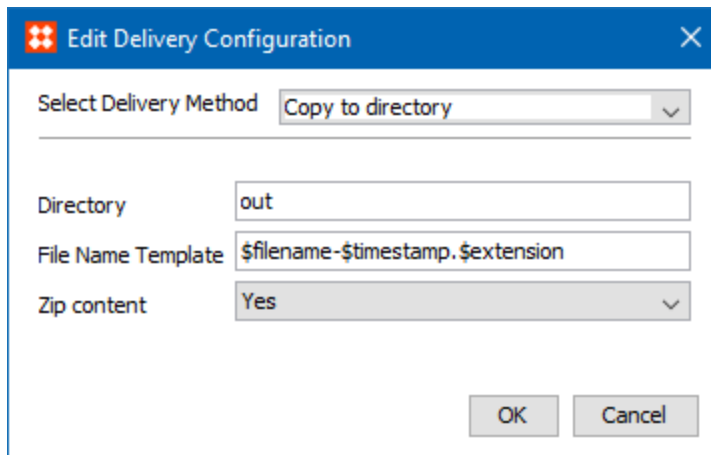
Prerequisites

The default delivery path is **opt/stibo/step** on the application server. This can be modified via the sharedconfig.properties file on the STEP application server using the case-sensitive **DirectoryDeliveryPlugin.RootDir** property. Changes to the properties file are implemented when the server is restarted.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. On the **Configuration** tab, navigate to the **Delivery Method** flipper, then click **Edit Delivery**.
2. Click the **Select Delivery Method** dropdown and choose **Copy to Directory**.



3. In **Directory**, specify the directory on the application server that will house the exported files. Do not start this text with a slash (/). Text entered will be appended to the path as defined in the Prerequisites section.
In the image above, assuming the default path is being used, the file will be delivered to 'opt/stibo/step/out'.
4. In **File Name Template**, enter the file naming template to be used for the deliveries. Each element of the template (indicated by the \$ character) is replaced with text as defined below. The template allows you to format the exported file name as required by external systems. The default template is: \$filename-\$timestamp.\$extension. Note that the hyphen (-) and the period (.) are actual characters that are used to build the complete name of the exported file.

- **\$filename** For event-based OIEPs, this variable is replaced with 'csv' for CSV format, 'excel' for Excel format, and the event ID for STEPXML format exports. For Select Object OIEPs, \$filename is replaced with 'exported'. Text can be added before or after the \$filename template, and \$filename can also be omitted altogether. When exporting multiple contexts, '##' is used as a separator within the filename to distinguish the context name for each output.
- **\$timestamp** This variable is replaced with the time in milliseconds between January 1, 1970, and the time when the file is uploaded, which allows for a unique identifier. For example, 1479230247017 is used for an export on 15 NOV 2016.

Alternatively, the following options can be used to generate a readable date and can include static characters to separate the timestamp elements. Individual elements can be ordered as required, as shown below.

\$timestamp(ddMMyyyy) displays day, month, and year and is output '15112016' for 15 NOV 2016.

\$timestamp(yyyy-MM-dd_HH.mm.ss) displays year, month, day, hour, minutes, and seconds and is output as '2016-11-15_09.32.43' for 2016 NOV 15 at 9:32.43 AM.

- **\$extension** This variable is replaced with the extension of the selected output file format.

Note: The File Name Template does not support the conversion of file formats and can only be used to deliver files in the format specified on the Configuration tab of the outbound integration endpoint.

5. For **Zip content**, select an option from the dropdown to specify if the output file should be delivered in a .ZIP (compressed) file format.
 - **Yes** uses the File Name Template and the extension .ZIP.
 - **No** uses the File Name Template for the file name along with the appropriate extension for the selected data format.
6. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Deploy Delivery Method

For OIEPs, there is no single delivery plugin for server side delivery, but the 'Deploy' and 'Copy to directory' options provide the same functionality in that they both deliver files to a directory on the application server. Common setup is to use 'Copy to directory' since it is more flexible because it allows the delivery to be zipped. Neither of these options is available in Export Manager.

For more information, see the **Copy to Directory Delivery Method** documentation.

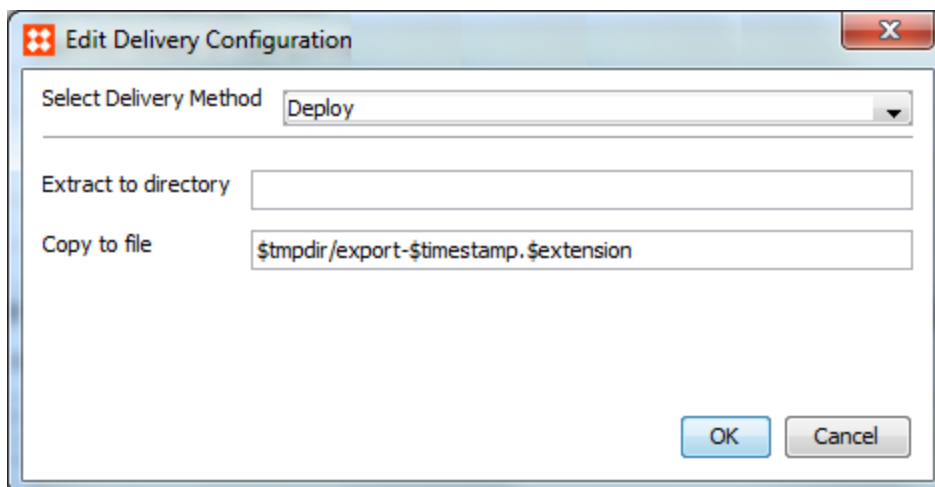
Prerequisites

The default delivery path is **opt/stiibo/step** on the application server. This can be modified via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **DirectoryDeliveryPlugin.RootDir** property. Changes to the properties file are implemented when the server is restarted.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. On the **Configuration** tab, navigate to the **Delivery Method** flipper, then click **Edit Delivery**.
2. Click the **Select Delivery Method** field to display the dropdown and choose **Deploy**.



3. In **Extract to Directory**, specify a path on the application server where the exported .ZIP file should be unzipped. The field will only have an effect if the exported file is a .ZIP file, otherwise this field will be ignored.
4. In **Copy to file**, specify a location and file name template to define a path where exported file should be delivered. Static text is also allowed in addition to, or instead of, the template options (which are indicated by the initial \$). The elements in the example template shown in the image are described below:
 - **\$tmpdir** This variable points to the directory path defined by the **ApplicationServer.TmpDir** entry in the `sharedconfig.properties` file. Remove '\$tmpdir' to enforce a location relative to the **DirectoryDeliveryPlugin.RootDir** delivery path in the `sharedconfig.properties` file.

- **export** This represents and can be replaced with any static text required in the file name. For example, the Copy to File field could read 'out/EmergencySKUExport-\$timestamp.\$extension' instead, and would deliver the output to the 'out' directory below the path indicated in the properties file, and the file name would start with the text 'EmergencySKUExport-' before adding the timestamp, a period, and the file extension.
- **\$timestamp** This variable is replaced with the time in milliseconds between January 1, 1970, and the time when the file is uploaded, which allows for a unique identifier. For example, 1479230247017 is used for an export on 15 NOV 2016.

Alternatively, the following options can be used to generate a readable date and can include static characters to separate the timestamp elements. Individual elements can be ordered as required, as shown below.

\$timestamp(ddMMyyyy) displays day, month, and year and is output '15112016' for 15 NOV 2016.

\$timestamp(yyyy-MM-dd_HH.mm.ss) displays year, month, day, hour, minutes, and seconds and is output as '2016-11-15_09.32.43' for 2016 NOV 15 at 9:32.43 AM.

- **\$extension** This variable is replaced with the extension of the selected output file format.

Note: The 'Deploy' template does not support conversions of file formats and can only be used to deliver files in the format specified on the Configuration tab of the outbound integration endpoint.

5. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Email Delivery Method

The email delivery option is very straight forward and, like 'Copy to directory', it allows the delivery to be zipped. It is typically used to test the export output, or to deliver a small export to one or more persons as a one-time occurrence.

This delivery method is available in Export Manager and in OIEPs. For information on a parameter, hover over the parameter to display help text.

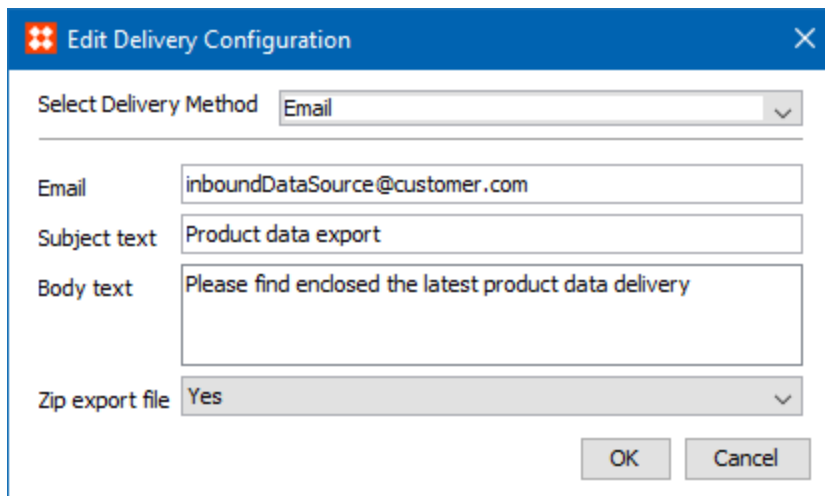
Prerequisites

To deliver an exported file via email, the STEP server must have access to the SMTP server. Be aware of email file size limits, since depending on export contents, the file could be very large.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. Click the **Select Delivery Method** parameter to display the dropdown and choose **Email**.
2. For the **Email** parameter, enter an email address.
3. For the **Subject text** parameter, enter the text that will display for the email subject line.
4. For the **Body text** parameter, enter the text that will display for the email body.



Edit Delivery Configuration

Select Delivery Method: Email

Email: inboundDataSource@customer.com

Subject text: Product data export

Body text: Please find enclosed the latest product data delivery

Zip export file: Yes

OK Cancel

5. In **Zip export file**, select an option from the dropdown to specify if the output file should be delivered as an email attachment in a .ZIP (compressed) file format.
 - **Yes** zips the exported file.
 - **No** does not zip the exported file.
6. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

FTP Delivery Method

The FTP delivery method allows an exported file to be delivered to an external system and is also available in Export Manager.

The preferred method for FTP delivery is the SFTP secure delivery method. For information on the OIEP SFTP delivery method, see **SFTP Delivery Method**.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

Prior to configuration, clicking the **HostName** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **FTPDeliveryHostName** property. The required format of the property is (square brackets not included):

```
SFTPDeliveryHostName=1=[host1],2=[host2]
```

Using the host name shown in the image below, the entry in the properties file would be:

```
FTPDeliveryHostName=1=FTPDeliveryHostServerName
```

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. In **Select Delivery Method**, choose **FTP**.

2. In **Host name**, select the host name or IP address of the FTP server to be used for the delivery.
3. In **Proxy host name**, add the host name to be used for the server proxy. This field is optional.

4. In **User name**, enter the user name that has access to log on to the FTP server.
5. In **Password**, enter the password that will be used to log on to the FTP server.
6. In **File name template**, to create the file name used for the output, enter template variables, static text, or use a combination. The default template is \$filename-\$timestamp.\$extension and creates a file name like 'GenericXML-1490813266535.xml' based on the settings of the OIEP. Each variable is described below:

- **\$filename** This variable is replaced with text to indicate the format of the delivered file as specified in the Output Templates flipper. For example, the output file name would include the text 'excel' or 'GenericXML' for those formats, or the Event ID for event-based STEPXML. Using the default file name template, a comma-separated value file would be named 'csv-1490808860412.csv' and 'GenericXML-1490810861593.xml' would indicate that Generic XML was used. For select object OIEPs, \$filename is replaced with 'exported.'
- **\$timestamp** This variable is replaced with the time in milliseconds between January 1, 1970, and the time when the file is uploaded, which allows for a unique identifier. For example, 1479230247017 is used for an export on 15 NOV 2016.

Alternatively, the following options can be used to generate a readable date and can include static characters to separate the timestamp elements. Individual elements can be ordered as required, as shown below.

\$timestamp(ddMMyyyy) displays day, month, and year and is output '15112016' for 15 NOV 2016.

\$timestamp(yyyy-MM-dd_HH.mm.ss) displays year, month, day, hour, minutes, and seconds and is output as '2016-11-15_09.32.43' for 2016 NOV 15 at 9:32.43 AM.

- **\$extension** This variable is replaced with the extension of the output file based on the selected format in the Output Templates flipper. For XML-based formats, the output file is set with the .XML extension, while Excel files use .XLS, and comma-separated values use .CSV.

Note: The File Name Template does not support conversions of file formats and can only be used to deliver files in the format specified on the Configuration tab > Output Templates flipper of the outbound integration endpoint.

7. In **Zip before upload**, select an option from the dropdown to specify if the output file should be delivered in a .ZIP (compressed) file format.
 - **Yes** uses the File Name Template and the extension .ZIP. When a file is zipped, the File Name Template is used for the .ZIP file. The \$filename template is replaced with 'result_0', and the selected file type is compressed with the name 'exported'. For example, a zipped STEPXML output with the default File Name Template exported on 15 NOV 2016 results in an output .ZIP file named 'result_0-1479230247017.zip'.
 - **No** uses the File Name Template for the file name along with the appropriate extension for the selected data format.
8. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

IBM Websphere MQ SSL Delivery Method

For information on connecting to IBM Websphere MQ in a non-SSL way, see **JMS Delivery Method**. This delivery option is only available in OIEPs.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. Prior to configuration, clicking the **Connection URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLProviderURL** property. Use the `[host]:[port]/[channel]` format as defined below:
 - `[host]` = hostname or IP of the Websphere MQ server
 - `[port]` = port number for the channel
 - `[channel]` = name of the channel
2. Prior to configuration, clicking the **Queue Manager** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLQueueManager** property.
3. Prior to configuration, clicking the **Queue Name** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLQueue** property.
4. Prior to configuration, clicking the **Key Store** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLKeyStoreLocation** property.

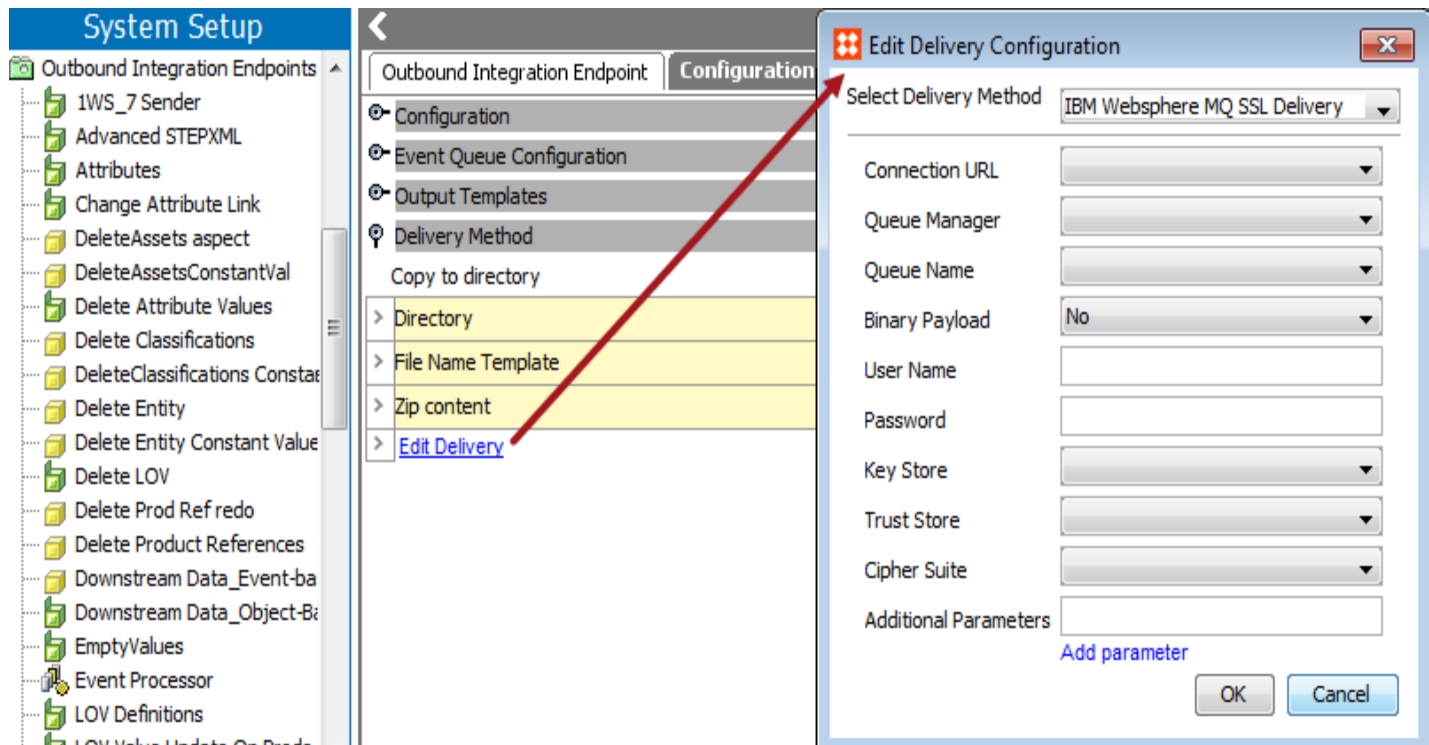
Use the `jks` format with the personal certificate for the Queue Manager. To generate this, consult the manual for IBM Websphere MQ. The password for the key store must be configured in `sharedconfig.properties` file using the **WSMQSSLKeyStorePassword** property.
5. Prior to configuration, clicking the **Trust Store** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLTrustStoreLocation** property.

This can be the same file as key store. To generate this, consult the manual for IBM Websphere MQ. The password for the trust store must be configured in `sharedconfig.properties` or `config.properties` using **WSMQSSLTrustStorePassword**.
6. Prior to configuration, clicking the **Cipher Suite** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **WSMQSSLCipher Suite** property. STEP is running on non-IBM JRE, so this must be the same value as configured in the Queue Manager.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. In the **Select Delivery Method** field, choose **IBM Websphere MQ SSL Delivery**.



2. In **Connection URL**, select the URL for the connection.
3. In **Queue Manager**, select the name of the Queue Manager.
4. In **Queue Name**, select the name of the Queue for the connection.
5. In **Binary Payload**, select 'Yes' if the message contents will be in a binary format (such as Excel).
6. In **User Name**, if required, enter the user name that will be used to log onto the JMS provider.
7. In **Password**, if required, enter the password that will be used to log onto the JMS provider.
8. In **Key Store**, select the keystore in jks format, with the personal certificate for the Queue Manager.
9. In **Trust Store**, select the trust store with the CA for the Queue Manager.
10. In **Cipher Suite**, set to the same value as SSL CipherSuite in Websphere MQ.
11. If **Additional Parameters** are required, click Add parameter, then enter the Key and Value. For possible keys and values, consult the manual for IBM Websphere MQ.
12. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

JDBC Delivery Method

The JDBC delivery method allows STEP data to be delivered directly to tables in Relational database management systems (RDBMS) like Oracle, MySQL, MS SQL Server, and PostgreSQL. Though the ways the JDBC plugin can be deployed are various, one of its prime uses is to send STEP data to data analytics tools for the purpose of displaying STEP data in a data analytics dashboard. This delivery method requires CSV format and is available in both the Export Manager and in Outbound Integration Endpoint (OIEP).

For more information on the Export Manager option, see **JDBC Delivery Method** in the **Export Manager** documentation.

Prerequisites

Important: For complete setup requirements, see the **Exporting Data via JDBC with CSV Format** topic in the **Data Format** documentation.

Install the required drivers

JDBC specification 4.1-compliant drivers should be placed in a directory accessible from all application servers. These drivers can then be made available for the delivery plugin via the dynamic properties `JDBCDeliveryPlugin.DriverPath.[n]` and `JDBCDeliveryPlugin.DriverClass.[n]`. For more information regarding applicable Java drivers, review the RDBMS vendor's homepage on the web.

Configure data for the dropdown parameters

Changes to the properties file are implemented when the server is restarted.

1. Prior to configuration, clicking the **Driver Location** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **JDBCDeliveryPlugin.DriverPath.[n]** property. As an example:

```
JDBCDeliveryPlugin.DriverPath.1 = L:/shared/mysql-connector-java-5.1.42-bin.jar
```

In this example, the drivers are stored on the application server's L:/shared drive.

2. Prior to configuration, clicking the **Driver Class** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **JDBCDeliveryPlugin.DriverClass.[n]** property. As an example:

```
JDBCDeliveryPlugin.DriverClass.1 = com.mysql.jdbc.Driver
```

In this example, the drivers class used is 'com.mysql.jdbc.Driver.'

3. Prior to configuration, clicking the **Database URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the `sharedconfig.properties` file on the STEP application server using the case-sensitive **JDBCDeliveryPlugin.URL.[n]** property.

```
JDBCDeliveryPlugin.URL.1 = jdbc:mysql://localhost:3306/mydb
```

In this example, the database URL used is 'jdbc:mysql://localhost:3306/mydb.'

Configuration

For information on a parameter, hover over the parameter field to display help text.

In the OIEP editor on the **Configuration** tab, navigate to the **Delivery Method** flipper, then click **Edit Delivery**.

1. In **Select Delivery Method**, choose **JDBC** from the dropdown.
2. In **Driver Location**, select from the dropdown one of the paths to the relevant .jar file.
3. In **Driver Class**, select from the dropdown one of the pre-configured driver class.
4. In **Database URL**, select from the dropdown one of the pre-configured URLs to the destination database.
5. In **Username**, enter the username required to access the destination database.
6. In **Password**, enter the password required to access the destination database.
7. In **Table Name**, enter the name of the table in the destination database to which STEP will be publishing data
8. In **Key Columns**, list the names of the columns appearing on both the exported CSV file and the table in the destination database, separated by commas (and no spaces), into which STEP will publish data
9. In **Delete Key Columns**, list the names of the columns appearing on both the exported CSV file and the table in the destination database, separated by commas (and no spaces), from which STEP will delete data. The headers contained in this field can differ from the headers in the 'Key Columns' field, but they must also be part of the upsert key definition.
10. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

JMS Delivery Method

The available options for the Java Message Service (JMS) delivery method are system dependent. STEP has three standard JMS delivery plugins available: Apache Active MQ, Websphere MQ, and Oracle AQ. Each of these can be used to deliver messages to a JMS receiver without any further customizations. To deliver messages to other JMS providers, contact your Stibo Systems account manager for further customizations. This delivery option is only available in OIEPs.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

See the **Sharedconfig.properties File Examples** section below for more information.

1. Prior to configuration, clicking the **Connection Factory Name** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JMSConnectionFactoryName** property.
2. Prior to configuration, clicking the **JMS Provider URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JMSProviderURL** property. If necessary, use a comma to separate multiple URLs.

The following is an example of a complete property entry for two systems named 'qa' and 'stage,' as well as their URLs 'http://step-qa' and 'http://step-stage':

```
JMSProviderURL=qa=http://step-qa,stage=http://step-stage
```

3. Prior to configuration, clicking the **JMS Queue** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **JMSQueue** property.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. On the **Configuration** tab, in the **Delivery Method** area, click **Edit Delivery**.
2. In **Select Delivery Method**, choose **JMS Delivery**.

3. In **JMS Connection Factory Class**, choose the JMS connection factory class that corresponds to the driver class from the JMS provider vendor. On a standard system, the following options are available:

- **ActiveMQInitialConnectionFactory**: Connects the JMS delivery to Apache Active MQ. For information, search the web.

Note: WMQInitialContextFactory is no longer supported, use FileInitialContextFactory instead.

- **FileInitialContextFactory**: Enables setting up JMS Websphere Inbound and Outbound Integration Endpoints which reference a binding file (created from JMS Websphere Client software). The binding file is a configuration file which includes all details of how STEP should interact with JMS Websphere. For information about JMS Websphere Client Software, search the web.

4. In **Connection Factory Name**, select a connection factory name from the list.

5. In **JMS Provider URL**, select a JMS Provider URL from the list.

6. In **JMS Queue**, select the physical name of the JMS Queue to be used on Apache Active MQ or Websphere MQ.

7. In **Binary Payload**, select 'Yes' if the message contents will be in a binary format (such as Excel).

8. In **User Name**, if required, enter the user name that will be used to log onto the JMS provider.

9. In **Password**, if required, enter the password that will be used to log onto the JMS provider.

10. If additional parameters are required, click **Add parameter**, then enter the Key and Value.

11. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Sharedconfig.properties File Examples

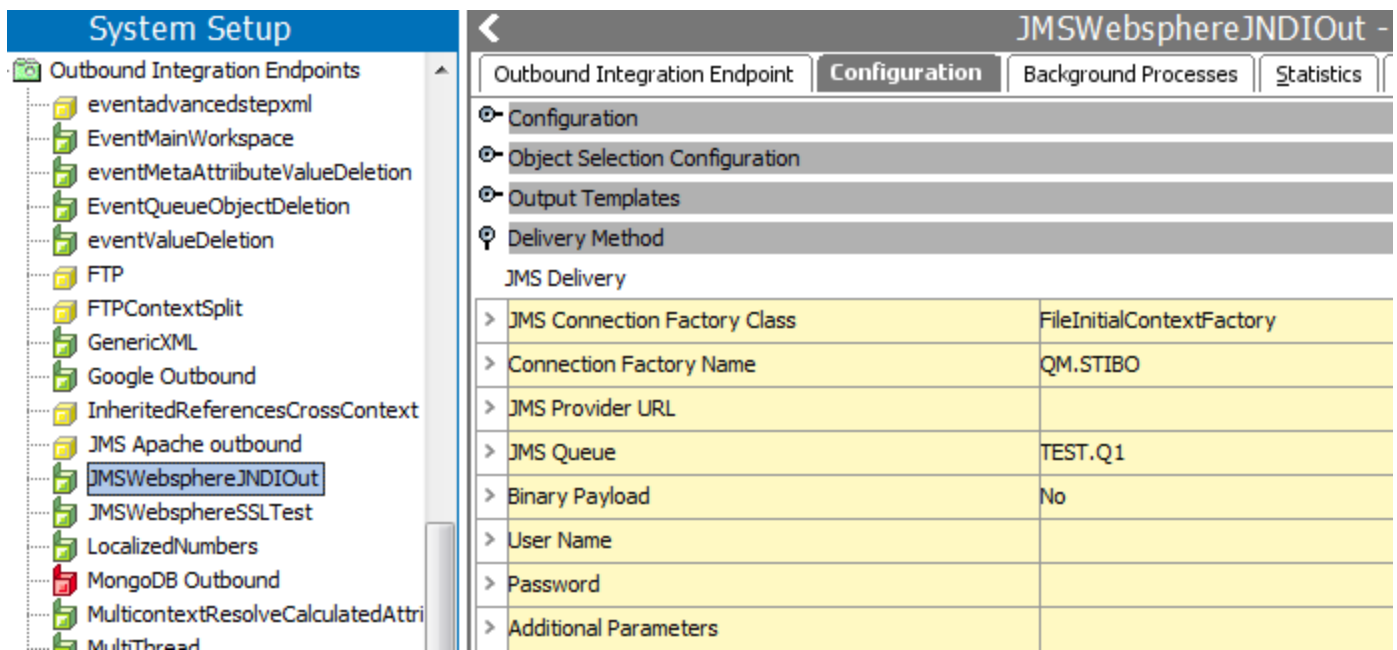
Each of these properties has a key/value pair as a value:

```
JMSConnectionFactoryName=FirstFactoryName=MyFactoryName
JMSProviderURL=ProviderURL=https://myproviderurl.com/url
JMSQueue=Queue=MyqueueName
```

Each of the properties can contain more than one value in a comma separated list:

```
JMSConnectionFactoryName=FirstFactoryName=MyFactoryName1,SecondFactoryName=MyFactoryName2
JMSProviderURL=FirstProviderURL=https://myproviderurl.com/url,SecondProviderURL=https://myproviderurl2.com/url
JMSQueue=Queue1=FirstqueueName,Queue2=SecondqueueName2
```

FileInitialContextFactory JMS Delivery Configuration Example



The screenshot shows the 'System Setup' window with a tree view on the left and a configuration panel on the right. The tree view includes 'Outbound Integration Endpoints' and various integration types. 'JMSWebSphereJNDIOut' is selected. The configuration panel shows tabs for 'Outbound Integration Endpoint', 'Configuration', 'Background Processes', and 'Statistics'. The 'Configuration' tab is active, showing a 'JMS Delivery' section with the following properties:

Property	Value
JMS Connection Factory Class	FileInitialContextFactory
Connection Factory Name	QM.STIBO
JMS Provider URL	
JMS Queue	TEST.Q1
Binary Payload	No
User Name	
Password	
Additional Parameters	

JMS Websphere Delivery Using SSL Configuration Example

Delivery Method

IBM Websphere MQ SSL Delivery

> Connection URL	webspheremq-qa.stibo.com:1417/STEP.SVRCONN
> Queue Manager	QM.STIBO_SSL
> Queue Name	TEST.Q1
> Binary Payload	Yes
> User Name	
> Password	
> Key Store	file:/workarea/JMSWebsphereSSLKeystore/keyStore.jks
> Trust Store	file:/workarea/JMSWebsphereSSLKeystore/keyStore.jks
> Cipher Suite	TLS_RSA_WITH_AES_256_CBC_SHA256
> Additional Parameters	

Apache Active MQ JMS Delivery

Delivery Method

JMS Delivery

> JMS Connection Factory Class	ActiveMQInitialContextFactory
> Connection Factory Name	ConnectionFactory
> JMS Provider URL	tcp://ATTCM3S9:61616
> JMS Queue	testqueue
> Binary Payload	No
> User Name	
> Password	
> Additional Parameters	
> Edit Delivery	

Mongo Delivery Method

Prior to selecting the Mongo Delivery Method, ensure your setup meets the prerequisites explained in **Prerequisites for Configuring the MongoDB Adapter** documentation. This delivery option is only available in OIEPs.

For more information, see <https://docs.mongodb.com/manual/> or the **MongoDB Adapter Setup Quick Guides** documentation.

For more information on the Mongo Delivery Method, see the **Mongo Delivery Method Elements** documentation.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. If your system displays radio button options for the server selection, see the **Legacy Configuration** section below for more details.
2. Prior to configuration, clicking the **Server(s)** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive dynamic **MongoDB.Server.Configuration.[n]** property, where the [n] is used for differentiation. The first configuration entry must use number 1, and each additional entry must be numbered sequentially (MongoDB.Server.Configuration.1, MongoDB.Server.Configuration.2, MongoDB.Server.Configuration.3, etc.). Each host is separated from the respective port using a colon (:), and additional servers are added using a comma-separated list. For example:

```
MongoDB.Server.Configuration.1=host4:40004
```

indicates an independent MongoDB instance. A second and third configuration could be supplied for servers being used in a cluster using the format:

```
MongoDB.Server.Configuration.2=host1:10001,host2:20002,host3:30003
```

```
MongoDB.Server.Configuration.3=host4:40004,host5:50005,host6:60006,host7:70007
```

3. The **Key store location** is only required if the Mongo DB requires the clients to use SSL to encrypt connections.

The password for the key store is configured in sharedconfig.properties on the STEP application server, using the case-sensitive **MongoDB.SSLKeyStorePassword** property. For example:

```
MongoDB.SSLKeyStorePassword = keystore_password
```

4. In **Trust store location**, is only required if the Mongo DB requires the clients to use SSL to encrypt connections.

The password for the key store is configured in sharedconfig.properties on the STEP application server, using the case-sensitive **MongoDB.SSLTrustStorePassword** property. For example:

```
MongoDB.SSLTrustStorePassword = truststore_password
```

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. In **Select Delivery Method**, choose **Mongo delivery**.

Edit Delivery Configuration

Select Delivery Method: **Mongo delivery**

Server(s): **host1:10001,host2:20002,host3:30003**

Raw DB prefix: **MongoDB**

User name: **userauth**

Password: **.....**

Replace Dollar sign:

Replace Dot sign:

☐ Use SSL

Key store location:

Trust store location:

Actions: **Workflow Review**

[Add action](#)

OK **Cancel**

2. In **Server(s)** dropdown, select the desired MongoDB server configuration, either an independent MongoDB instance or a MongoDB cluster setup.
3. In **Raw DB prefix**, if required, enter a prefix for all raw databases the adapter creates. The output is [prefix] + [context], resulting in a database name of MongoDBContext1 using the data in the image above, and exporting for Context1. Optional.
4. In **User name**, if MongoDB is configured to use authentication, enter the MongoDB user. Optional.
5. In **Password**, if MongoDB is configured to use authentication, enter the MongoDB password. Optional.
6. In **Replace Dollar sign**, text entered replaces any dollar sign (\$) during delivery. If blank, all dollar signs are replaced automatically by the JSON converter with Unicode 'uFF04'. This allows overriding the default Unicode replacement with the specified character(s), for example, an underscore (_).
7. In **Replace Dot sign**, text entered replaces any dot sign (.) during delivery. If blank, all periods are replaced automatically by the JSON converter with Unicode 'uFF0E'. This allows overriding the default Unicode replacement with the specified character(s), for example, an underscore (_).
8. Check **Use SSL** if MongoDB requires the client to use SSL encryption. Optional. If no further parameters are configured, it is assumed that the MongoDB uses a SSL certificate that is issued by a trusted certificate authority (like Verisign or Thawte). If this is not the case, or if more security of the connection is requested, you can configure an SSL key store and an SSL trust store.

- In **Key store location**, enter a key store location, for example, file:///path/key_store.jks. The Key store holds the STEP Mongo Adapter private key and SSL certificate in a keystore file as generated by the Java utility 'keytool' (in jks format). The password for the key store is configured as defined in the **Prerequisites** section.
 - In **Trust store location**, enter a trust store location, for example, file:///path/trust_store.jks. The trust store holds the Mongo DB public key and certificate in a keystore file as generated by the Java utility 'keytool' (in jks format). The trust store configuration is only needed if the certificate is self-signed and not issued by a trusted certificate authority. The password for the key store is configured as defined in the **Prerequisites** section.
9. In **Actions**, click the **Add action** link to enter one or more STEP business actions used to create and populate aggregated collections. These actions are run after the raw collections have been populated.
 10. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Legacy Configuration

When the sharedconfig.properties file includes the MongoDB.Server property and the MongoDB.Port property, the dialog displays both a cluster option and a legacy option. Typically this happens when a STEP system had a MongoDB host / port configured prior to the update that introduced the clustered configuration functionality.

Note: Upgrading the sharedconfig.properties file to modify all configurations using the legacy properties to the cluster enabled property, and then removing the legacy properties completely, results in the radio button options being removed from the dialog, as shown in the image at the top of this topic.

When the **Cluster Enabled Configuration** radio button is selected, the **Server(s)** dropdown displays entries for the dynamic MongoDB.Server.Configuration.[n] property.

When the **Legacy Configuration** radio button is selected, the **Server** and **Port** dropdowns display entries for the MongoDB.Server and MongoDB.Port properties.

Multiple independent servers or ports can be listed using the following format:

```
MongoDB.Server=1=vps124887.vps.ovh.ca,2=cde,3=rty
```

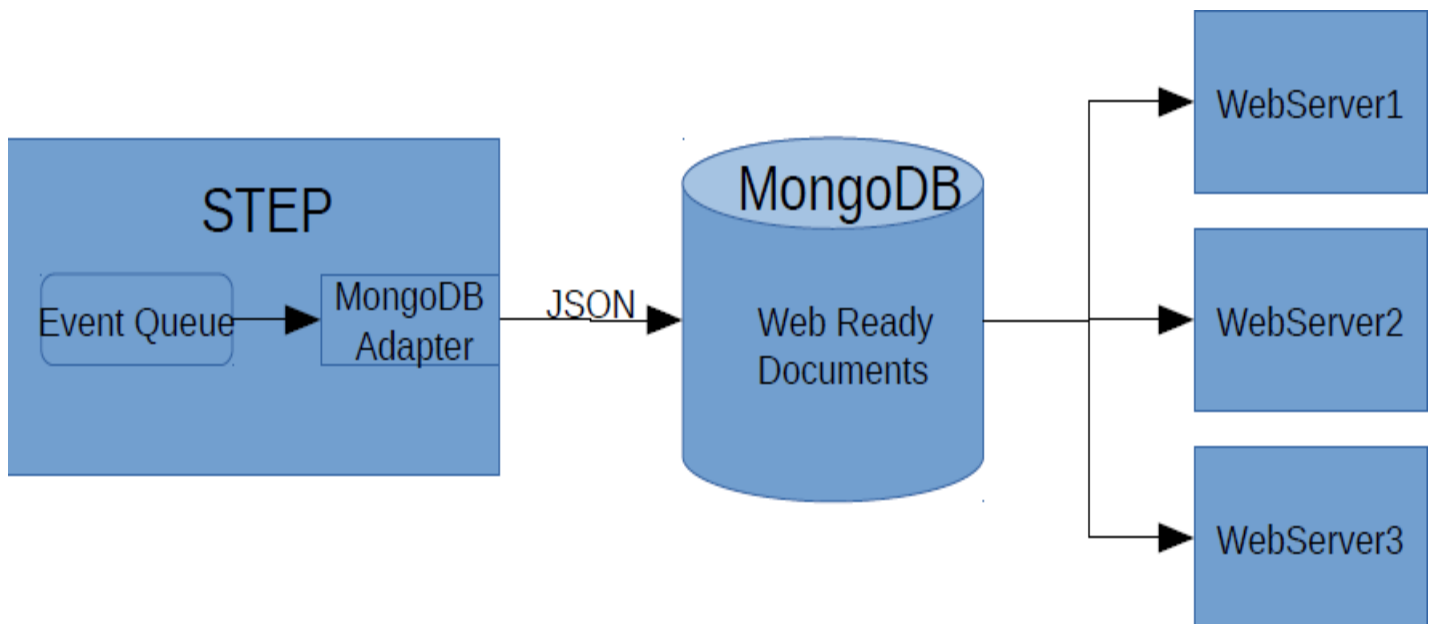
```
MongoDB.Port=1=30001,2=30002,3=30003
```

Mongo Delivery Method Elements

The STEP Mongo Adapter is an outbound integration endpoint delivery plugin that receives data from a STEP event queue and loads it into a MongoDB database. The MongoDB is often used for web site back-end, reporting, and high performance feeds to other back-end systems.

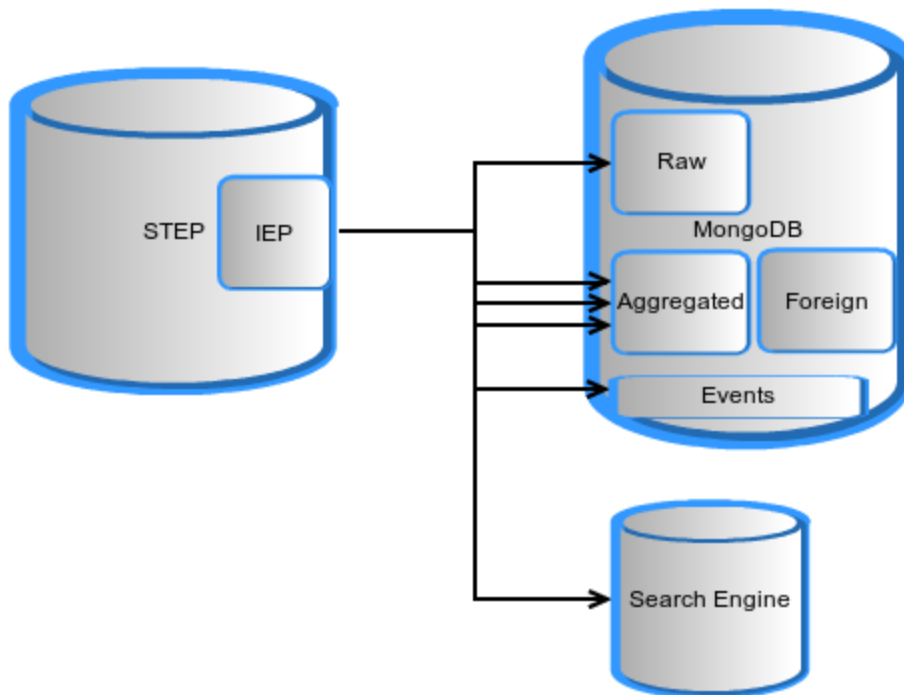
The Mongo adapter converts STEPXML to JSON and keeps a set of 'raw' collections in MongoDB that are in sync with STEP. Additionally, it allows you to add JavaScript triggers to maintain aggregated collections.

The following illustration shows how the Mongo Adapter can be used to feed a Mongo database that is used as the back-end to a website:



By keeping collections of web-ready JSON documents available in MongoDB, the JSON can be passed directly to the browser's JavaScript. This results in the web server having little processing to do, which means that it simply needs to pass the right collection result directly back to the browser.

The following drawing illustrates the collections involved:



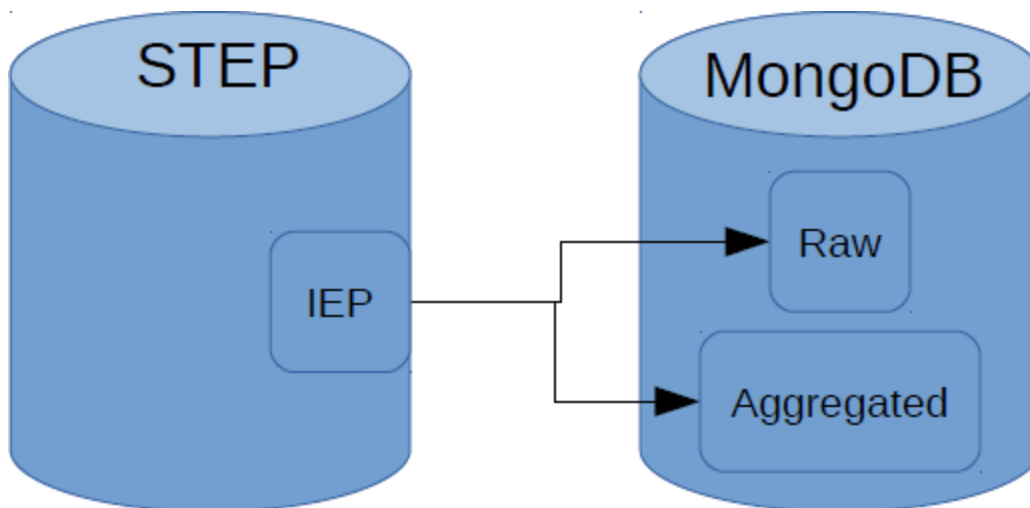
The STEP Mongo Adapter ensures that the raw collections are a one-to-one mirror of data in STEP. The adapter does not embed nodes found through references or child associations. These nodes are stored separately in the raw collections.

It is possible to configure the adapter so that it creates one or more customized, aggregated MongoDBs (databases or collections) where the documents can be tailored to meet the exact usage requirements, like reporting, analysis, and so on.

The JavaScript triggers allow maintaining aggregated collections, for example:

- List of children, including title and other data necessary in a tree-like view.
- Information from the target end of references that is needed in a given view.
- Mapping to a JSON structure that is closer to the web site data model.

The following image shows a MongoDB database containing a raw database that is maintained directly by the STEP Mongo Adapter and an aggregated database maintained by JavaScript triggers.



A JavaScript trigger is executed under certain conditions, for example, only for a specific object type. This reduces the amount of checking necessary in the JavaScript code. Once the raw collection is updated and the JavaScript triggers (if configured) have completed, an event is inserted into the events collection. Also, an update is sent to the search engine.

Note: Aggregated collections and raw collections do not need to be in the same MongoDB instance. However, if required, different collections in the aggregate can be combined into their own MongoDB instance.

Default Databases and Collections

The STEP Mongo Adapter maintains a number of MongoDBs, referred to as 'raw databases', and a number of collections in the raw databases, referred to as 'raw collections'. A raw database is created for each context that is exported and the database is named after the context. For example, if the two contexts EN and FR are exported, two raw databases are created, one named EN and one named FR.

It is possible to configure a string that prepend the raw database name when the STEP Mongo Adapter is configured. When data is exported from multiple contexts, a JSON document is created per context, per exported object. Continuing the previous example, two JSON documents are created for each object: one for the object in the EN context and one for the object in the FR context.

A raw collection is only created when the particular STEP data type is exported. So, if no assets are exported from STEP, no asset collection is created. Each raw database contains all or some of the following raw collections:

Raw Collection	Purpose
product	Exported products
asset	Exported assets
classification	Exported classifications

Raw Collection	Purpose
entity	Exported entities
attribute	Exported attribute definitions
referenceType	Exported reference types
listofvalues	Exported list of values (LOVs)
unit	Exported unit definitions

Inherited values, calculated values, and unit names on values are handled already in the raw collections. Since changes to these affect replicated information, STEP ensures that appropriate updates are made to sync the replicated information.

STEP JSON and Mongo JSON

The STEP Mongo Adapter converts STEPXML input to STEP JSON (JavaScript Object Notation) documents, and then stores the JSON documents in the MongoDB. The STEP JSON schema can be downloaded from your STEP server at: [http://\[enter step-server\]/files/StepSchema.json](http://[enter step-server]/files/StepSchema.json)

Formally, MongoDB documents are BSON documents, which is a binary representation of a JSON document. For more information, see the MongoDB documentation on the web.

For an expanded example of the basic conversion using STEP Mongo Adapter, see the **Mongo Delivery Method Conversion Example** documentation.

JavaScript Triggers

STEP business action triggers with JavaScript populate the aggregated MongoDB databases and collections. You can configure one or more business action triggers for the STEP Mongo Adapter. The business action triggers are fired after the STEP Mongo Adapter has populated the raw collections.

During configuration of the STEP Mongo Adapter, the configuration wizard allows you to select multiple JavaScript to run under different conditions by adding preconditions to the business actions. A precondition is, for example, a JavaScript trigger that is only fired for updates to a given type of objects. For more information, see the **Applies If Tab** section of the **Editing a Business Rule** documentation.

Due to the nature of inheritance and calculated attributes, the trigger may be called even if there are no changes to the underlying raw objects.

The business action triggers are executed once per JSON document sent to the MongoDB database. If two objects are exported from STEP, two JSON documents are sent to the MongoDB database and the business action triggers are executed twice. If the objects are exported in multiple contexts, one JSON document is

generated per context, per object. This means that if two products in two contexts are exported, the business action triggers are executed four times.

Business action triggers are executed on the STEP server. Business actions triggers need access to the MongoDB database to fetch data from the raw collections and write or read data to / from the aggregated collections. For the best performance, network latency between the STEP server and the MongoDB server should be minimal.

Important: Place aggregated collections in their own database so that the raw database contains only collections.

Bound Variables

The JavaScript trigger have accesses to information in the Mongo Adapter environment by binding script variables when the JavaScript trigger is defined.

Type	Gives access to
MongoDBContext	- The current STEP context. - The name of the raw MongoDB database. - The MongoDB database via the <code>com.mongodb.Mongo</code> object. For more information, see: http://api.mongodb.org/java/current/com/mongodb/Mongo.html. The actual Mongo object that is passed can be a wrapper, but it adheres to the interface
JSON context	The JSON object with the data that has been persisted in MongoDB.

For more information about executing JavaScript in business rules, see the **Business Action: Execute JavaScript** section of the **Business Rules** documentation.

Example JavaScript to Maintain a Collection in a Classification

The following code-snippet illustrates an example JavaScript that maintains a collection containing the list of products (ID and Name) in a classification.

The structure of the collection is:

```
{_id: "<classification-id>", products : { { id: "<product-id1>", name:
"<product-name1>"}, { id: "<product-id2>", name: "<product-name2>"}, ... }

function getClassificationIDs(classrefs) {
    var result = new Array();
    if (classrefs != null) {
        for (i = 0; i < classrefs.targets.size(); ++i) {
            result.push(new String(classrefs.targets[i].targetID));
        }
    }
}
```

```

    }
    return result;
}

var db = mongoContext.getMongo().getDB('extra');
var collection = db.getCollection('classificationproducts');

var classrefs = mongoData.references ? mongoData.references["Web
Classifications"] : null;

var classificationIDs = getClassificationIDs(classrefs);

var deletequery = {"products.id" : mongoData._id , "_id" : { $nin :
getClassificationIDs(classrefs) }};
var adelete = { $pull : { products : { id : mongoData._id } } };

collection.update(deletequery, adelete, false, true);

var query = {"products.id" : mongoData._id };
var name = mongoData.name;
var update = { $set : { "products.$.name" : name } };

collection.update(query, update, false, true);

classificationIDs.map(function(item) {
    var query = { _id : item };
    var update = { $addToSet : { products : { id : mongoData._id, name : name }
} };
    collection.update(query, update, true, false);
});

```


Prerequisites for Configuring the MongoDB Adapter

The following are prerequisites for setting up the MongoDB adapter:

1. A MongoDB instance with suitable amounts of disk space.
2. The STEP MongoDB Adapter uses a network port to access the MongoDB database on the server where the MongoDB is running. The network port must be open and accessible from the STEP servers.
3. Update `sharedconfig.properties` or `config.properties` with the **name of the MongoDB server**, using the `MongoDB.Server` configuration property and a list of number-to-value pairs, separated by commas, as follows:

```
MongoDB.Server = 1=MongoDBServer1, 2=MongoDBServer2, 3=MongoDBServer3
```

4. Update `sharedconfig.properties` or `config.properties` with the **MongoDB port** using the `MongoDB.Port` configuration property, and a list of number-to-value pairs, separated by commas, as follows:

```
MongoDB.Port = 1=10001, 2=2002, 3=3003
```

For example, to configure the adapter to use a MongoDB instance running on either 'Server1' or 'Server2' using port '27712' on both servers, the properties would be:

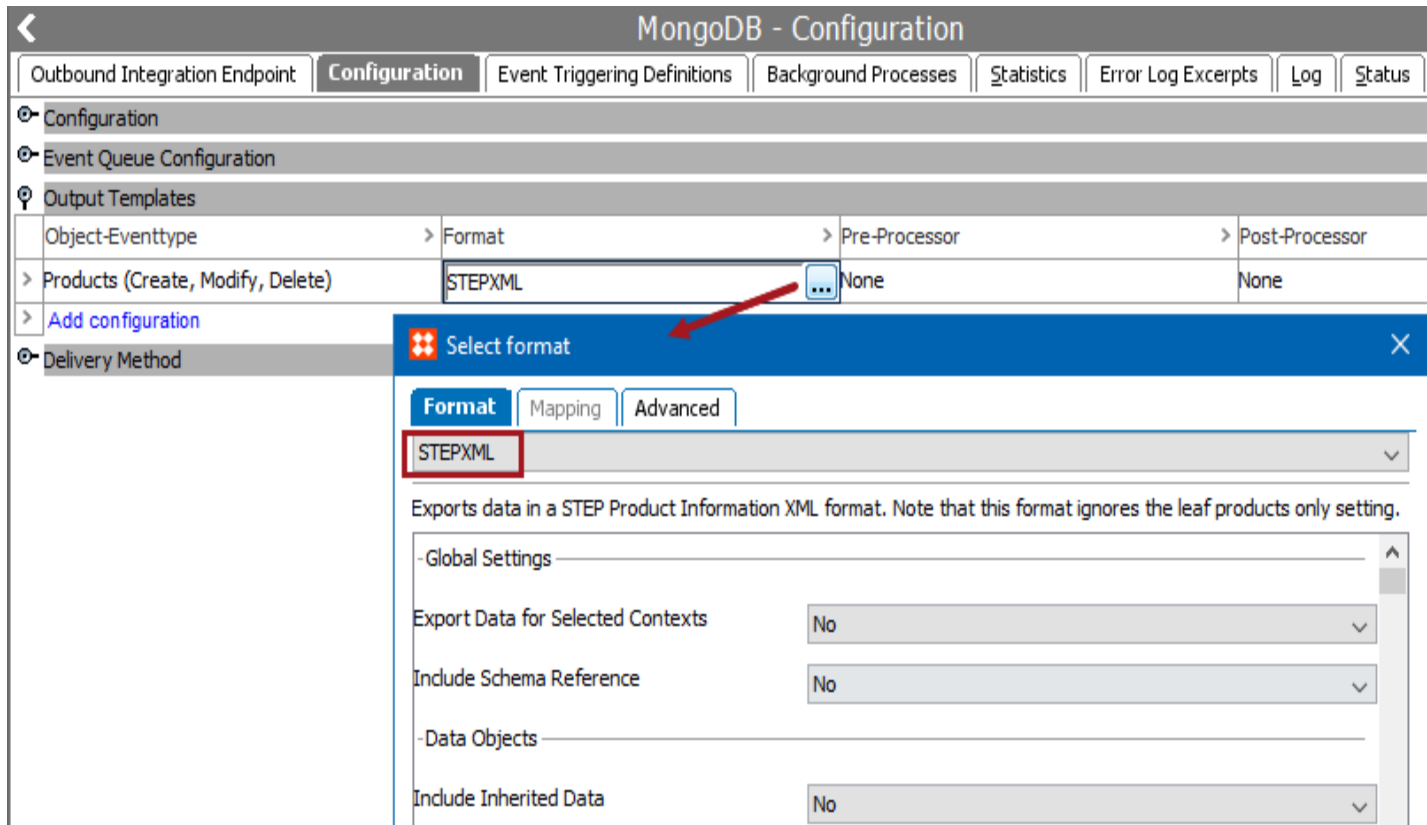
```
MongoDB.Server = 1=Server1, 2=Server2
```

```
MongoDB.Port = 1=27712
```

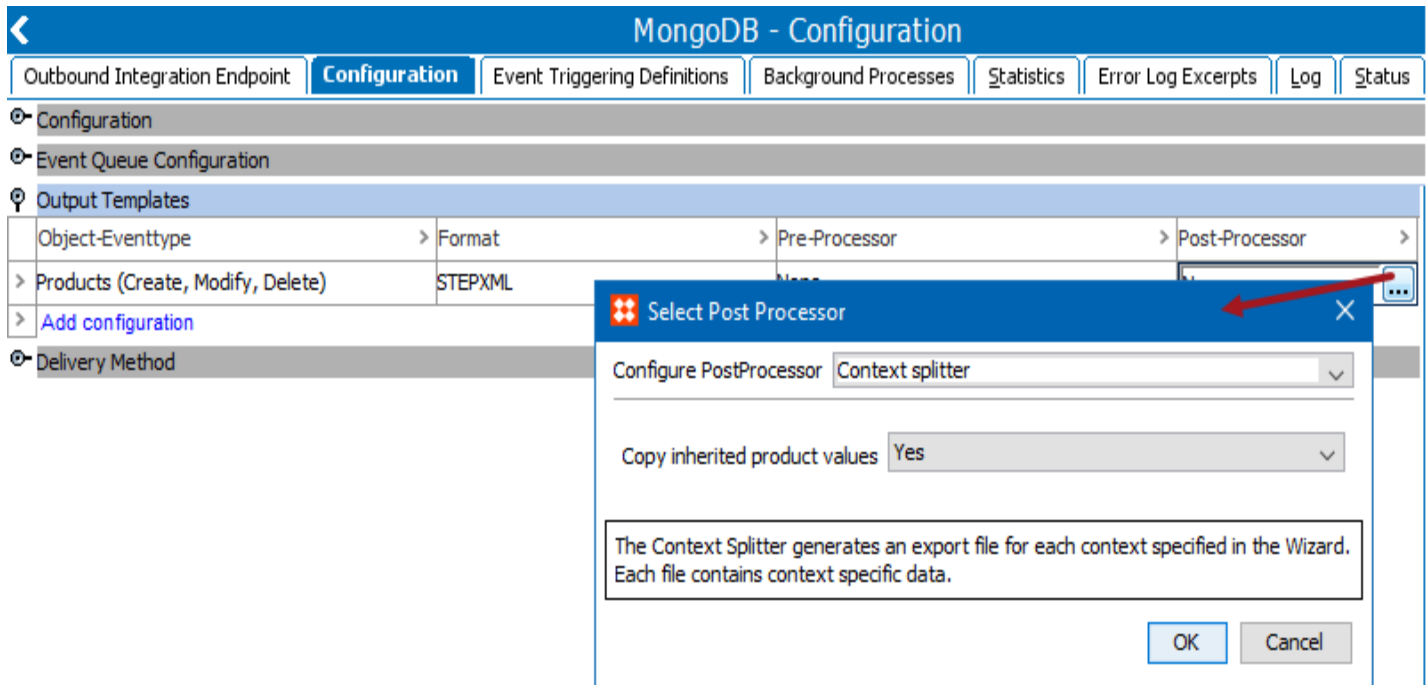
Configure the OIEP

The STEP MongoDB Adapter is run within an outbound integration endpoint (OIEP) and is configured as the delivery method for the endpoint.

1. In System Setup, create an OIEP, selecting STEP Exporter as the Process Engine. For details on creating a new OIEP, see the **Creating an Event-Based Outbound Integration Endpoint** topic.
2. For the new OIEP, open Configuration > Output Templates > Add configuration > Format > click the ellipsis button (...).



3. On the Select Format dialog, leave objects as default values except for the following parameters:
 - Select the **STEPXML** export format in the dropdown at the top of the dialog
 - Data Objects > **Flatten Hierarchies** = Yes
 - Data Objects > **Include Entities** = Selected
 - Data Objects > **Include Products** = Selected
 - Data Objects > **Include Classifications** = Selected
 - Data Objects > **Include Assets** = Selected
 - Configuration > **Include Attribute Groups** = Minimum
 - Configuration > **Include Link, Reference and Object Types** = Minimum
 - Configuration > **Include List of Values** = Minimum
 - Configuration > **Include Units** = Minimum
4. In the Post-Processor field, click the ellipsis button (...) and select **Context splitter**, then click OK.



- For Delivery Method, if MongoDB requires user authentication, configure a MongoDB user and password. The MongoDB user must be configured in the MongoDB admin database and have permissions to create new MongoDB databases (the MongoDB role "readWriteAnyDatabase").

Configuring the JavaScript Triggers

JavaScript triggers are fired for all updates to the raw collections. The purpose of the triggers is to maintain the aggregated collections. The JavaScript triggers are configured as business actions in STEP and then applied when configuring the MongoDB Adapter.

Continue with setting up the Mongo Delivery Method options, by following the **Mongo Delivery Method** documentation.

MongoDB Adapter Setup Quick Guides

The following information is available to assist in the setup of Mongo Delivery Method.

Configuring Mongo Authentication Quick Guide

The following quick guide describes how to configure a Mongo database to use authentication. Details may be found in the Mongo documentation on the web.

The guide assumes that the setup is used on a Mongo database installation running as a single server installation (i.e., the Mongo database is running as a standalone server).

1. Start the Mongo database without authentication.
2. Log onto the database using the Mongo client.
3. Create a system administrator user.

```
> use admin
switched to db admin
> db.createUser(
  {
    user: "admin",
    pwd: "password",
    roles:
    [
      {
        role: "userAdminAnyDatabase",
        db: "admin"
      }
    ]
  }
)
Successfully added user: {
  "user" : "admin",
  "roles" : [
    {
      "role" : "userAdminAnyDatabase",
      "db" : "admin"
    }
  ]
}
```

```
}
```

4. Create the user that will be used when the MongoDB adapter logs onto the Mongo database. This user should be created in the admin database with roles to read and write any databases in Mongo:

```
> use admin
switched to db admin
> db.createUser( {"user" : "stepsys", "pwd" : "stepsys", "roles" : [
"readWriteAnyDatabase" ] })
Successfully added user: { "user" : "stepsys", "roles" : [
"readWriteAnyDatabase" ] }
```

5. Stop the Mongo database and enable authentication. This may be done in the Mongo configuration file:

```
auth = true
```

or from the command line:

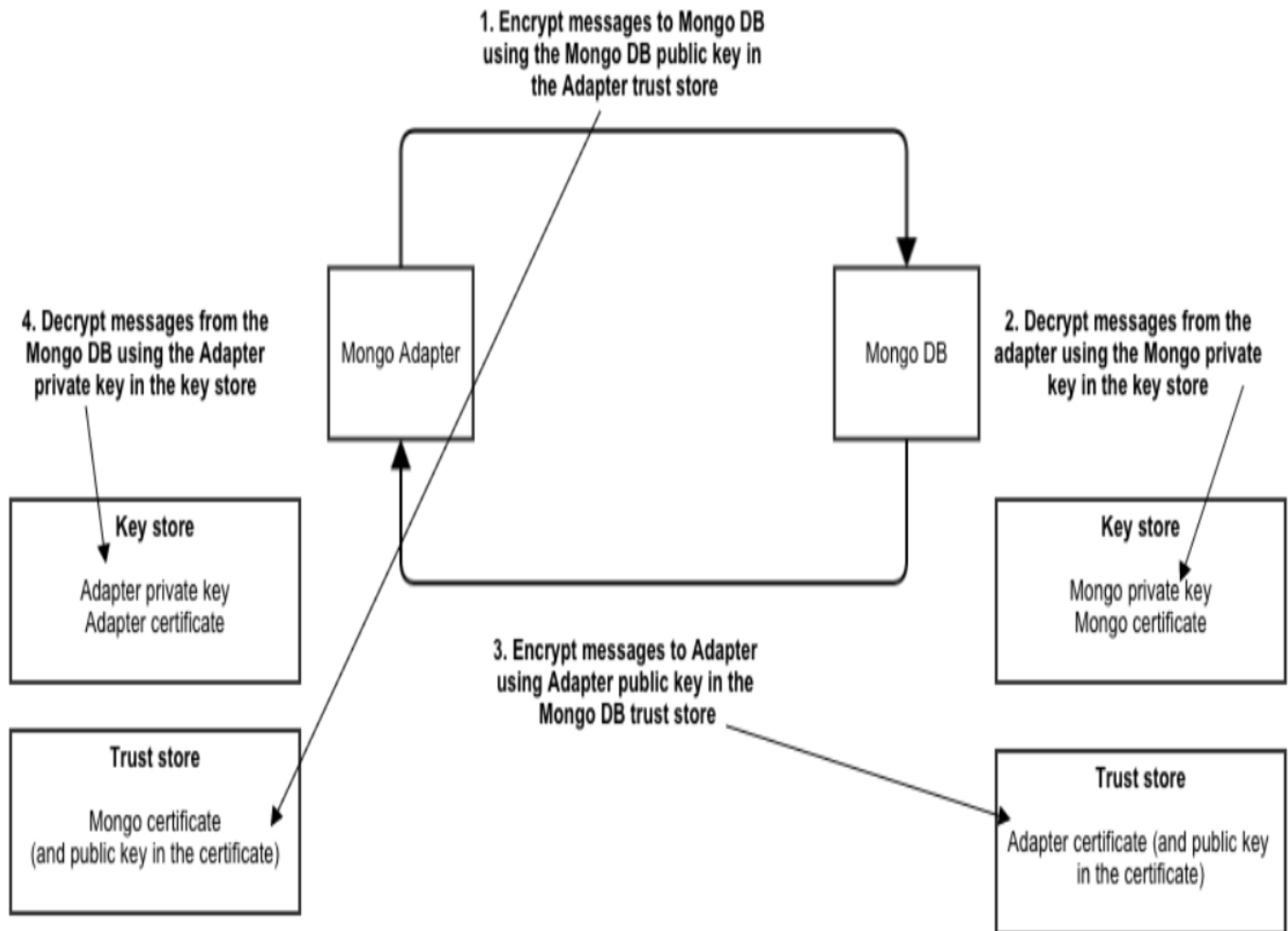
```
mongod --auth --config
```

SSL Configuration Quick Guide

When the Mongo DB adapter is configured for SSL, a key store and a trust store must be configured.

- The key store is a key store file in jks format containing the Mongo Adapter (STEP) private key and certificate.
- The trust store is a trust store file in jks format containing the Mongo DB public key and certificate.

Key store files may be created from the public and private key files and certificates using utilities like the java keytool utility. The diagram below illustrates the uses of the key store and trust store.



To configure SSL encryption on the STEP-to-Mongo database connection:

- Configure the Mongo database to use SSL: <http://docs.mongodb.org/manual/tutorial/configure-ssl/>
- Configure the Mongo client to connect to a Mongo database using SSL: <http://docs.mongodb.org/manual/tutorial/configure-ssl-clients/>

The Mongo database available as a free download does not support SSL. (SSL requires the Enterprise edition of Mongo.) Instead, download the appropriate source and compile your own version of the Mongo database from the web.: <https://docs.mongodb.com/manual/installation/>

Mongo Delivery Method Conversion Example

The following examples use this product with type = 'Item', ID 'XYZ', and Name = 'XYZ Name'. The product is located in the family 'FamilyXYZ2'.



ID: EXA-5002-1001

Name: Arm chair

Type: Chair

Parent ID: Chairs

The STEP Mongo Adapter transforms this to the STEP JSON document:

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
}
```

The id of the product is used as unique ObjectId in the MongoDB database, and therefore the STEPXML id field is transformed to _id. The UserTypeID XML attribute is converted to the objectTypeID field. The type field describes the kind of STEP object that is exported. For Basic STEP objects, the types are 'product', 'asset', 'classification', and 'entity'. Other STEP objects are described in the following topics.

Values

In this example, a product has two attributes:

- A single value attribute Color with the value 'Brown'
- A multi value attribute Height with the value = '43', the unit = 'inches', and the value = '120', with the unit = 'cm'.

When this is transformed to a STEP JSON document by the STEP MongoDB adapter, it results in the following:

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
  values : {
    Color : "Brown",
    Height : [ "43 inches", "120 cm"]
  }
}
```

The data displayed includes:

- Values of the product are stored as a subdocument with the name 'values'.
- The value of the single-valued attribute 'Color' is represented as a field-and-value pair: the ID of the attribute is the field, and the value is the value of the attribute represented as a string.
- The multi-valued attribute 'Height' is represented as a field-to-arrays of strings. Again, the ID of the attribute is the field, and the string array contains a string for each value.
- The name of the unit is appended to the attribute value with a space for all values with a unit. In the example above, '120 cm' is the concatenation of the value '120' and 'cm' from the unit named 'unece.unit.CM'.

Values of specification attributes that are inherited to the product are included in the JSON document of the product. For example, if the parent contains an attribute called Brand with the value 'Office chairs', the JSON document looks like the following:

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
  values : {
    color : "Brown",
    Height : [ "43 inches", "120 cm"],
    Brand: "Office chairs"
  }
}
```


LOV Value IDs

When values are included from an LOV that uses Value IDs (via the parameter 'Use Ids on values' = Yes on the LOV), the Value and the ValueID are stored as a subdocument with the name 'extValues'. For backward compatibility, the LOV value is also included in the 'value' subdocument, meaning this data is available twice, and can be extracted based on the need.

Note: If the LOV does not use Value IDs, the extValues subdocument is not written.

In this example, the LOV value of '4001' has a Value ID of '4001_CALYPSO'.

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
  values : {
    color : "Brown",
    Height : [ "43 inches", "120 cm"],
    Brand: "Office chairs"
    attribute_id: "4001",
  },
  extValues: {
    attribute_id: {
      value: "4001",
      valueID: "4001_CALYPSO"
    }
  }
}
```

The data displayed includes:

- The selected LOV value is stored within the subdocument with the name 'values' in the attribute_id field and again with the 'extValues' subdocument in the 'attribute_id: {value:}' field.
- The value ID of the selected LOV value is represented once with the 'extValues' subdocument in the 'attribute_id: {valueID:}' field.

References and Links

References and links are converted to STEP JSON in a subdocument called 'references'. This subdocument contains a field-and-value pair, where the field is the ID of the reference type, and the value is either a

subdocument that defines the target of a reference, or an array of subdocuments that each define the target of a reference.

If the reference can only reference one target for the same source, the value is a subdocument. If the reference can reference more targets for the same source, the value is an array.

In this example, the product uses a 'Primary Image' asset reference to reference the image for the product. The asset has the ID 'Image_EXA-5002-1001'. The STEP JSON then exports as:

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
  values : {
    color : "Brown",
    Height : [ "43 inches", "120 cm"],
    Brand: "Office chairs"
  }
  references : {
    Primary Image : { targetID : "Image-EXA-5002-1001" }
  }
}
```

To find the asset for the reference, it is necessary to find the reference type 'Primary Image', which is stored in the raw collection 'referenceType'. The target type of the reference comes from the reference type.

The JSON of the reference target is found by searching the raw collection given by the reference type of an object with 'targetID'.

In the example above, the target type of the 'Primary Image' reference type is 'asset'. By looking in the 'asset' raw collection, the asset given by Image-EXE-5002-1001 is found.

Adding the reference 'Secondary Image', which can reference more targets such as ('Image1' and 'Image2') to the product, results in the following STEP JSON:

```
{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
```

```

values : {
  color : "Brown",
  Height : [ "43 inches", "120 cm"],
  Brand: "Office chairs"
}
references : {
  Primary Image : { targetID : "Image-EXA-5002-1001" },
  Secondary Image : [ { targetID : "Image1" }, { targetID : "Image2" } ]
}

```

If the reference contains a meta data attribute, the values of the meta data attribute are added as a subdocument to the reference subdocument that contains the value of the attribute:

```

{
  _id : "EXA-5002-1001",
  objectTypeID : "Chair",
  parentID : "Chairs",
  name : "Arm chair",
  type : "product"
  values : {
    color : "Brown",
    Height : [ "43 inches", "120 cm"],
    Brand: "Office chairs"
  }
  references : {
    Primary Image : { targetID : "Image-EXA-5002-1001",
                      values: { ShowOnWeb : "true" },
                      },
    Secondary Image : [ { targetID : "Image1" }, { targetID : "Image2" } ]
  }
}

```

The reference type 'Primary Image' has a meta data attribute called 'ShowOnWeb'. The attribute is set to 'true' for the reference from the product to the asset 'PrimaryAsset'.

Classification to product links are special because they are owned by either the product or the classification. This will always be the same for a specific classification to product link type. The JSON document only contains the

classification-to-product links owned by the object. That is, the JSON document for a product only contains classification-to-product links owned by the product.

Attribute Types

Exporting an attribute type with the ID 'AttributeTypeID', and the name 'Name' linked in two parent attribute groups 'Parent1' and 'Parent2' results in the following JSON:

```
{
  _id : "AttributeTypeID",
  parentID : [ "Parent1", "Parent2"],
  name : "Name",
  listOfValuesID : "ListOfValuesID",
  validUnitIDs : ["unece.unit.MMT", "unece.unit.CMT"],
  type : "attribute"
}
```

Reference types are stored in the raw collection 'attribute'. To export reference types, verify that 'Include Attribute Definitions' is set to at least 'Selected' in the Process Engine configuration.

Reference Types

Exporting a product-to-product reference type with the ID 'ReferenceTypeID' and the name 'ReferenceName' generates the following JSON:

```
{
  _id : "ReferenceTypeID",
  name : "ReferenceName",
  type : "referenceType",
  targetType : "product"
}
```

In this example, 'targetType' is the target type of the reference. Possible target types are 'product', 'asset', 'classification', and 'entity'.

Reference types are stored in the raw collection 'referenceType'. When you export reference types, verify that Configuration > Include Link, Reference and Object Types is set to at least 'Minimum' in the Process Engine configuration.

Asset Push Locations

Asset Push Locations are stored in the assetPushLocations subdocument, in the root of the Asset documents in the asset collection. It provides the relative path to different versions of the asset. The assetPushLocations subdocument contains field-and-value pairs, where the field name is the assetpush configuration id, and the value is the relative path that the image file has been pushed to. The relative path is relative to the AssetPushClients root

folder, so to get the full path of the image, you have to prepend the path to the assetpush client root dir, as shown in the following example:

```
assetPushLocations : {
  AllAssets-approved : "AllAssets-approved/std.lang.all/73/15/7315.pdf",
  AllWebsiteImages-approved : "AllWebsiteImages-
approved/std.lang.all/73/15/7315.pdf",
  small : "small/std.lang.all/73/15/7315.jpg",
  large : "large/std.lang.all/73/15/7315.pdf"
}
```

Tables

Tables are defined on classifications or products. The table subdocument is stored in the root of the product and classification document, in the product and classification collection. The table subdocument contains field-and-value pairs, where the field name is the table Type name of the table on the product or classification.

A table is exported from STEP in a resolved and transformed state. That is, any transformations defined in STEP have already been applied to the table and the row and column layout has therefore been resolved.

Formatting is not applied to the table before it is exported. So, a web application is not bound by the STEP table formatting. However, if any formatting has been defined for the table in STEP, it is included in the export so that the website application can be configured to render the formatting on the website.

The table contains an array of column elements that contain the formatting meta data for each column and an array of rows. The row contains the formatting information for each row, plus an array of cells. Each cell has a number of formatting meta data attributes and a text field that contains the content of the table cell.

Table Formatting

Formatting meta data can be stored on table, column, row, and cell level, and is inherited downwards from table level to column level to row level to cell level.

If a column is specified to use an underlined text style, and a row is specified to be bold text style, the formatting information is accumulated so that the cell at the intersection of the row and column is rendered as both underlined and bold.

Formatting meta data defined at a lower level overrides meta data at a higher level. For example, if a table has a gray background color, and a column has a red background color, then cells in that row are rendered with a red background.

A simple way of formatting a table in HTML is to map the tableType, columnType, and rowType to element classes in HTML, and then use a CSS style sheet to define the table formatting, as shown in the following example.

```
tables : {
  Description Table : {
    columns :
```

```
[ { columnType : "Description" }, { columnType : "Description" }, {
columnType : "Description" } ],
rows : [
  { rowType : "Header",
    cells : [
      { ruleRight : "0.5 pt",
        text : "Header 1",
        verticalAlignment : "top",
        cellStoryDirection : "horizontal",
        backgroundColor : "Light Blue",
        column : "0",
        ruleBelow : "0.5 pt",
        ruleAbove : "0.5 pt",
        textStyle : "TableHeader-Description",
        ruleLeft : "0.5 pt" },
      { ruleRight : "0.5 pt",
        text : "Header 2",
        verticalAlignment : "top",
        cellStoryDirection : "horizontal",
        backgroundColor : "Light Blue",
        column : "1",
        ruleBelow : "0.5 pt",
        ruleAbove : "0.5 pt",
        textStyle : "TableHeader-Description",
        ruleLeft : "0.5 pt" }] },
  { rowType : "Header",
    cells : [
      { ruleRight : "0.5 pt",
        text : "Value 1",
        verticalAlignment : "top",
        cellStoryDirection : "horizontal",
        backgroundColor : "Light Blue",
        column : "0",
        ruleBelow : "0.5 pt",
        ruleAbove : "0.5 pt",
```

```

        textStyle : "TableHeader-Description",
        ruleLeft : "0.5 pt" },
    { ruleRight : "0.5 pt",
      text : "Value 2",
      verticalAlignment : "top",
      cellStoryDirection : "horizontal",
      backgroundColor : "Light Blue",
      column : "1",
      ruleBelow : "0.5 pt",
      ruleAbove : "0.5 pt",
      textStyle : "TableHeader-Description",
      ruleLeft : "0.5 pt" }] }
  ] } }

```

Attribute Links

AttributeLinks define the validity of attributes for products linked into the product and classification hierarchies. The attributeLinks subdocument is stored at the root of the product and classification document structure. The attribute links can themselves contain a values subdocument that defines the meta data values attached to the attributeLink.

The attributeLinks subdocument contains field value pairs, where the field name is the attributeID, and the value is a subdocument that contains any meta data values related to the attribute link. The following is an example of an attributeLinks subdocument:

```

attributeLinks : { Voltage range : { }
                  Rotary address switches : { },
                  Default address : { },
                  Power consumption : { }
                  }

```

Unit

Exporting an unit with the ID 'unece.unit.CMT', and the name 'cm' results in the following JSON:

```

{
  "_id" : "unece.unit.CMT",
  "values" : {
    "4941" : "centimeter"
  },
  "conversionToBase" : {

```

```

    "factor" : "100",
    "unitID" : "unece.unit.MTR",
    "offset" : "0"
  },
  "name" : "cm",
  "type" : "unit"
}

```

The unit document contains the 'conversionToBase' object that can be used for converting values from a unit to its base unit. The 'conversionToBase' object of the above example shows how to convert values of the unit centimeters to meters.

Units are stored in the raw collection 'unit'. To export units, verify that Configuration > Include Units is set to at least 'Minimum' in the Process Engine configuration.

List Of Values

Exporting a ListOfValues object with the ID 'LOVID', and the name 'List of values' results in the following JSON:

```

{
  "_id" : "LOVID",
  "values" : {
    "6823" : "This is a metadata value of the lov"
  },
  "name" : "List of values",
  "validUnitIDs" : [
    "unece.unit.CMT",
    "unece.unit.MTR"
  ],
  "type" : "listofvalues"
}

```

List Of Values are stored in the raw collection 'listofvalues'. To export List Of Values, verify that 'Include List Of Values Definitions' is set to at least 'Minimum' in the Process Engine configuration.

Oracle AQ Delivery Method

This delivery option is only available in OIEPs.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. Prior to configuration, clicking the **Connection URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **OracleAQReceiverConnectionURL** property.
2. Prior to configuration, clicking the **Queue name** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **OracleAQReceiverQueueName** property.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. In the **Select Delivery Method** list, choose **Oracle AQ Delivery**.

Edit Delivery Configuration

Select Delivery Method: Oracle AQ Delivery

Connection URL: configure the key 'OracleAQReceiverConnectionUrl' in config.properties

Queue Name: configure the key 'OracleAQReceiverQueueName' in config.properties

Binary Payload: No

User Name:

Password:

OK Cancel

2. In the **Connection URL** list, select a URL that points to Oracle AQ.
3. In the **Queue name** list, select an Oracle AQ queue name.
4. For **Binary Payload**, select Yes or No. 'Yes' is selected for non-text files.
5. In the **User Name** field, enter the user name that will be used to log on to Oracle
6. In the **Password** field, enter the password that will be used to log on to Oracle.
7. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

REST Delivery Method

The REST Delivery Method delivers a call-back URL to the REST service and does not include actual STEP data. The data can be fetched from the call-back URL by the receiving REST service. This delivery option is only available in OIEPs.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

Prior to configuration, clicking the **URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **RESTDeliveryURL** property. If necessary, use a comma to separate multiple URLs.

This should be a URL to a REST PUT method, for example, http://server/rest_URL. A URL where the result of the endpoint can be fetched is returned in the PUT call.

The following is an example of a complete property entry for two systems named 'qa' and 'stage,' as well as their URLs 'http://step-qa' and 'http://step-stage':

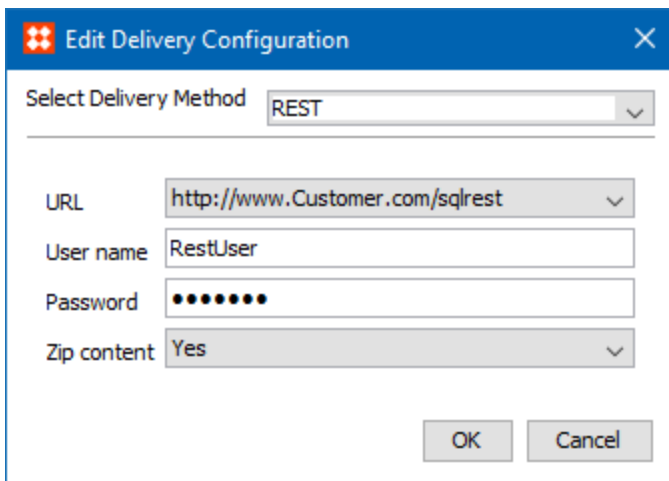
```
RESTDeliveryURL=qa=http://step-qa,stage=http://step-stage
```

Important: For information about how to use the REST API to upload files to REST, click the **STEP API Documentation** button on the STEP WebStart page.

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. For **Select Delivery Method**, choose **REST**.



2. In **URL**, select the URL that points to the REST endpoint where you would like to receive the delivered data.
3. In **User Name**, enter the user name that will be used to log on to the REST endpoint.
4. In **Password**, enter the password that will be used to log on to the REST endpoint.

5. In **Zip content**, specify whether to zip the contents before upload.
6. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

REST Direct Delivery Method

The REST Direct delivery method differs from the standard REST delivery method in that the data is delivered directly to the REST service and no call-back URL is required.

This delivery option is only available in OIEPs and if the delivery fails, the OIEP will be disabled. There is no special resilience handling.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. The REST Direct delivery method reads the outgoing file into memory before sending. To handle the size of the outgoing file and prevent time-outs and rejections you must scale the heap size.
2. Prior to configuration, clicking the **URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **RestDirectDeliveryURL** property. If necessary, use a comma to separate multiple URLs. The image below shows the result of using the following example for a property entry involving two systems:

```
RestDirectDeliveryURL = 1=http://myfirstendpoint, 2=http://mysecondendpoint
```

URL	https://myfirstendpoint
HTTP Method	https://myfirstendpoint https://mysecondendpoint

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. For **Select Delivery Method**, choose **REST Direct**.

Edit Delivery Configuration

Select Delivery Method: REST Direct

URL: https://someendpoint.com

HTTP Method: POST

Query Parameters: version = 2
[Add parameter](#)

Headers: Content-Type = application/xml
[Add parameter](#)

Footer (Optional):

ZIP Content: No

Basic Authentication (Optional):

Username: someuser

Password:

OK Cancel

2. In **URL**, select the URL that points to the REST endpoint where the delivered data should be received.
3. In the **HTTP Method** option, use the dropdown to select a POST, PUT, or PATCH method.

HTTP Method: POST

Query Parameters:

POST
PUT
PATCH

If required, in the **Query Parameters** field add any query parameters that are required for a successful REST transaction. You can click on the ellipsis (...) to add the 'Key' and 'Parameter' for the first parameter, or you can click on the **Add parameter** link to add the first and subsequent parameters.

4. For **Headers**, click the **Add Parameter** link and add a key and a value. This is part of the HTTP network protocol.

5. On the Add parameter dialog, click the **OK** button and the Header is displayed in the field. If multiple headers are needed, use the **Add Parameter** link to add each additional key and value pair.

Headers Content-Type = application/xml ... X
[Add parameter](#)

Note: Once a header is displayed, use the ellipsis button (...) to edit it or the X button to remove it.

For additional information about Headers, please see 'Additional Information About Headers' at the end of this section.

6. In **Footer**, add data required for the recipient to verify that the full message was received. This is part of the HTTP network protocol but not required.

The footer can be used to mark the end of a multi-part REST call (that is a REST call containing the payload split in more packages). The footer could also contain a checksum that the receiver can use to detect if the payload in a multi-part message has been changed by the middleware.

Headers	Content-Type = application/octet-stream ...
	Add parameter
Footer (Optional)	
ZIP Content	No ▼

7. In **Zip Content**, specify whether to zip the contents before upload.
8. In **Username**, optionally enter the user name that will be used to log on to the REST endpoint.
9. In **Password**, optionally enter the password that will be used to log on to the REST endpoint.

Basic Authentication (Optional)	
Username	
Password	

10. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Additional Request Information

By default, the REST Direct delivery plugins sends multipart/form-data POST requests with the exported file available in the part named 'file' with Content-Type application/octet-stream. The example below shows the properties for such a request:

Headers:

Accept-Encoding = gzip,deflate

Connection = close

Content-Length = 1354

Content-Type = multipart/form-data; boundary=JN_qruUpDaHqm9BgW_b4-adAHDftQbjuvDI

Body (Metadata):

Content-Disposition: form-data; name="file"; filename="result.zip"

Content-Type: application/octet-stream

Via the UI it is possible to change the HTTP method and also to overwrite the Content-Type header. For instance, this header could be set to 'application/xml' for a STEP XML export (unzipped) thereby overwriting the default 'multipart/form-data' value making the request properties be:

Headers:

Accept-Encoding: gzip,deflate

Connection: close

Content-Length: 1354

Content-Type: application/xml

Body: <The exported XML>

SFTP Delivery Method

The preferred method for FTP delivery is the SFTP secure delivery method. The SFTP delivery method allows an exported file to be delivered to an external system and is often used when the output files are large, or when a different or remote system is in use.

This delivery method is also available in Export Manager. For information on the OIEP FTP delivery method, see **FTP Delivery Method**.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

Prior to configuration, clicking the **Host name** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the case-sensitive **FTPDeliveryHostName** property. The required format of the property is (square brackets not included): `FTPDeliveryHostName=1=[host1],2=[host2]`.

Using the host name shown in the image, the entry in the properties file would be:

```
FTPDeliveryHostName=1=FTPDeliveryHostServerName
```

Configuration

For information on a parameter, hover over the parameter field to display help text.

1. Click the **Select Delivery Method** list to display the dropdown and select **SFTP**.

2. In **Host name**, select the host name of the SFTP server to be used for the delivery.
3. In **Proxy host name**, add the host name to be used for the server proxy. This field is optional.
4. In **User name**, enter the user name that has access to log on to the FTP server.

5. In **Password**, enter the password that will be used to log on to the FTP server.
6. In **File name template**, to create the file name used for the output, enter template variables, static text, or use a combination. The default template is \$filename-\$timestamp.\$extension and creates a file name like 'GenericXML-1490813266535.xml' based on the settings of the OIEP. Each variable is described below:

- **\$filename** This variable is replaced with text to indicate the format of the delivered file as specified in the Output Templates flipper. For example, the output file name would include the text 'excel' or 'GenericXML' for those formats, or the Event ID for event-based STEPXML. Using the default file name template, a comma-separated value file would be named 'csv-1490808860412.csv' and 'GenericXML-1490810861593.xml' would indicate that Generic XML was used.
- **\$timestamp** This variable is replaced with the time in milliseconds between January 1, 1970, and the time when the file is uploaded, which allows for a unique identifier. For example, 1479230247017 is used for an export on 15 NOV 2016.

Alternatively, the following options can be used to generate a readable date and can include static characters to separate the timestamp elements. Individual elements can be ordered as required, as shown below.

\$timestamp(ddMMyyyy) displays day, month, and year and is output '15112016' for 15 NOV 2016.

\$timestamp(yyyy-MM-dd_HH.mm.ss) displays year, month, day, hour, minutes, and seconds and is output as '2016-11-15_09.32.43' for 2016 NOV 15 at 9:32.43 AM.

- **\$extension** This variable is replaced with the extension of the output file based on the selected format in the Output Templates flipper. For XML-based formats, the output file is set with the .XML extension, while Excel files use .XLS, and comma-separated values use .CSV.

Note: The File Name Template does not support conversions of file formats and can only be used to deliver files in the format specified on the Configuration tab > Output Templates flipper of the outbound integration endpoint.

7. In **Zip before upload**, select an option from the dropdown to specify if the output file should be delivered in a .ZIP (compressed) file format.
 - **Yes** uses the File Name Template and the extension .ZIP. When a file is zipped, the File Name Template is used for the .ZIP file. The \$filename template is replaced with 'result_0', and the selected file type is compressed with the name 'exported'. For example, a zipped STEPXML output with the default File Name Template exported on 15 NOV 2016 results in an output .ZIP file named 'result_0-1479230247017.zip'.
 - **No** uses the File Name Template for the file name along with the appropriate extension for the selected data format.
8. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Wiki Delivery Method

The XWiki Delivery Method works with an OIEP configured to deliver metadata to a specified XWiki advanced open source enterprise wiki system. The contents of the wiki can be accessed by clicking a 'Go to Wiki' link from either Web UI or workbench.

The Wiki Metadata functionality is a licensed component and requires a separate installation recipe. For details and the latest recipe, contact your Stibo Systems representative.

Prerequisites

Changes to the properties file are implemented when the server is restarted.

1. Review the **Wiki Metadata** topic for details on the functionality and setup.
2. Prior to configuration, clicking the **Base URL** dropdown parameter displays the required property name. Provide a Base URL selection for the dropdown parameter via the sharedconfig.properties file using the case-sensitive **Wiki.WikiUrl** property. A protocol (http:// or https://) and an ending slash (/) are mandatory. The following is an example of a property entry for two wikis.

```
Wiki.WikiUrl=1=http://vp487.vps.ohs.ca/,2=https://127.0.0.1:8080/
```

If only one URL is required, in the example above, eliminate the ',2=https://127.0.0.1:8080/' text.

Note: The format for multiple wikis is 1=url1,2=url2,3=url3, where each entry is incremented and unique.

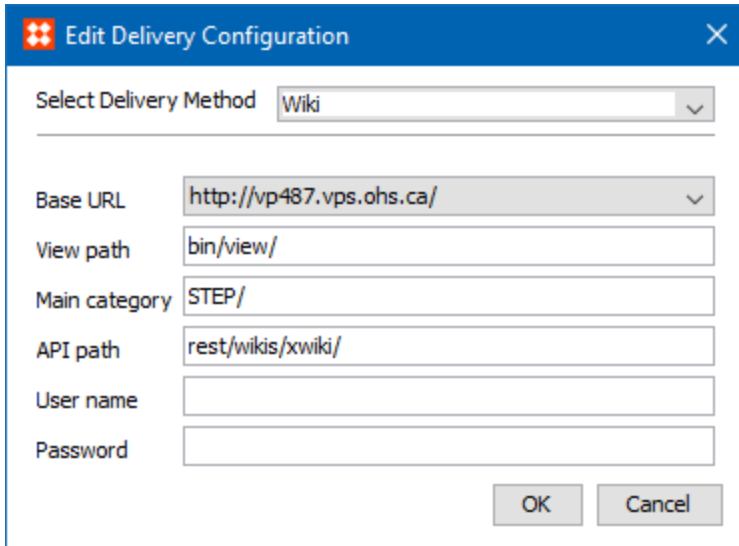
Configuration

For information on a parameter, hover over the parameter field to display help text.

After completing the prerequisite steps, edit the delivery method of the OIEP created during the installation. Use the following steps to configure the OIEP to use the XWiki delivery option.

Important: For the OIEP created automatically upon restarting the server after installing the STEP wiki metadata recipe, most of the following parameters are pre-populated and the default settings should not be changed. Review the parameter details below for more information.

1. For **Select Delivery Method**, choose **XWiki**.



Edit Delivery Configuration

Select Delivery Method: Wiki

Base URL: http://vp487.vps.ohs.ca/

View path: bin/view/

Main category: STEP/

API path: rest/wikis/xwiki/

User name:

Password:

OK Cancel

2. In **Base URL**, select the URL that points to the wiki that will be used for this STEP system. See the Prerequisites section for information on setup.
3. In **View Path**, enter the URL elements between the Base URL and the wiki pages, or main category if defined. This is used for the presentation of wiki pages and allows each user to watch pages they select. The slash at the end is mandatory. As shown in the image above, 'bin/view/' is the view path for the following URL:
http://vp487.vps.ohs.ca/**bin/view**/STEP/attribute/
4. In **Main category**, enter a name for the main category path. This allows you to organize the metadata pages and can be multiple levels by adding a slash, '/', between each one. The slash at the end is mandatory. As shown in the image above, 'STEP/' is the main category for the following URL:
http://vp487.vps.ohs.ca/bin/view/**STEP**/attribute/
5. In **API path**, specify the path for the wiki REST API, which is used to make modifications to the wiki pages. The slash at the end is mandatory.
6. In **User name**, enter the user with permission to call the REST API and who can make modifications to it.
7. In **Password**, enter the password for the user name.
8. On the **Edit Delivery Configuration** dialog, click the **OK** button to save the delivery method.

Running an Outbound Integration Endpoint

After configuring an outbound endpoint, it must be enabled before it can run on schedule or be invoked manually.

The following three colored icons are used to indicate the status of the integration endpoint. The endpoint status is displayed in the System Setup OIEP hierarchy and on the endpoint editor in the Description flipper > Endpoint Status parameter.

Icon	Endpoint Status	Description
	Enabled	The integration endpoint is active, connected, and running.
	Disabled	The integration endpoint has stopped and no data is being exported. Newly created endpoints, and endpoints that have been manually disabled, will have this status.
	Failed	An error occurred during processing that caused the integration endpoint background process to fail. No data will be exported until the endpoint has been manually resumed / reactivated. For more information, see Handling Failed IIEP Background Processes documentation.

Enable and Invoke an Outbound Integration Endpoint

Both event-based and select objects OIEPs must be enabled and invoked before they can run. Additionally, event-based OIEP must have the proper setting for queue status.

For example, if the external system is down and cannot accept data, keep the endpoint enabled but do not invoke the endpoint until the external system is ready to consume the data. In this way, data is collected from STEP and is then pushed to the external system only when invoked.

Prerequisite for Event-Based OIEPs

For an event-based OIEP, verify the Queue Status is set correctly. For information about how the Endpoint Status and Queue Status settings work together on event-based endpoints, see the **Event-Based OIEP Status and Queue Status** documentation.

For information about setting up an event processor, see **Event Processors** in the **System Setup / Super User Guide**.

Required Setup for All OIEPs

1. On System Setup expand the Outbound Integrations Endpoints node to display all existing outbound endpoints.
2. Select a  disabled endpoint or a  failed endpoint, right-click, and select **Enable Integration Endpoint** from the menu.

System Setup

- Match Codes and Matching Algorithms
- Outbound Integration Endpoints
 - 1WS 7.0 Provider Sender
 - Asset Caching
 - Change Attribute Link
 - GREExport
 - Item Modify
 - Item Modify Non STEPXML**
- Web
- Work
- Work
- Derive
- Obj
- Tags
- Units
- User
- Refer
- Work

Item Modify Non STEPXML rev.0.5 - Outbound Integration Endpoint

Background Processes | Statistics | Error Log Excerpts | Log | Status

Outbound Integration Endpoint | Configuration | Event Triggering Definitions

Description

Name	Value
ID	ItemModifyNonSTEPXML
Name	Item Modify Non STEPXML
Object Type	Outbound Integration Endpoint Type
Revision	0.5 Last edited by USERJ on Wed Jan 04 14:26:34 EST 2017
User	User J (USERJ)
Description	
Enabled	No
Endpoint Status	Stopped
Last run	2016-12-09 11:55:52
Next run	2016-12-09 11:56:50

Integration Endpoint Log

The tree navigator icon now shows enabled. On the Inbound Integration Endpoint tab, the Description flipper includes the 'Enabled' parameter, which now displays 'Yes'.

System Setup

- Match Codes and Matching Algorithms
- Outbound Integration Endpoints
 - 1WS 7.0 Provider Sender
 - Asset Caching
 - Change Attribute Link
 - GREExport
 - Item Modify
 - Item Modify Non STEPXML**
 - MultiValue MultiContext
 - Static Data Set
 - Website
- Web UIs
- Workflow Profiles
- Workflows
- Derived Events
- Object Types & Structures
- Tags
- Units

Item Modify Non STEPXML rev.0.5 - Outbound Integration Endpoint

Background Processes | Statistics | Error Log Excerpts | Log | Status

Outbound Integration Endpoint | Configuration | Event Triggering Definitions

Description

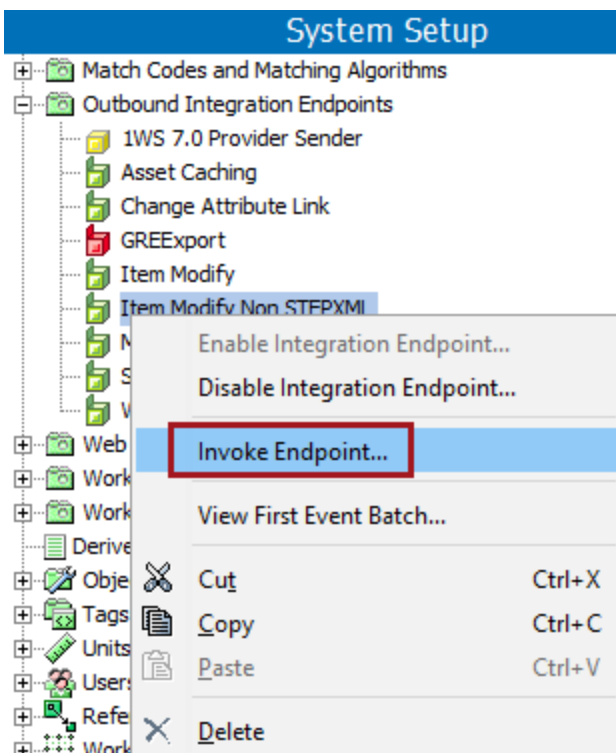
Name	Value
ID	ItemModifyNonSTEPXML
Name	Item Modify Non STEPXML
Object Type	Outbound Integration Endpoint Type
Revision	0.5 Last edited by USERJ on Fri Jan 06 14:25:07 EST 2017
User	User J (USERJ)
Description	
Enabled	Yes
Endpoint Status	Running
Last run	2016-12-09 11:55:52
Next run	2016-12-09 11:56:50

3. On the Configuration tab, open the Configuration flipper to view the Schedule parameter.

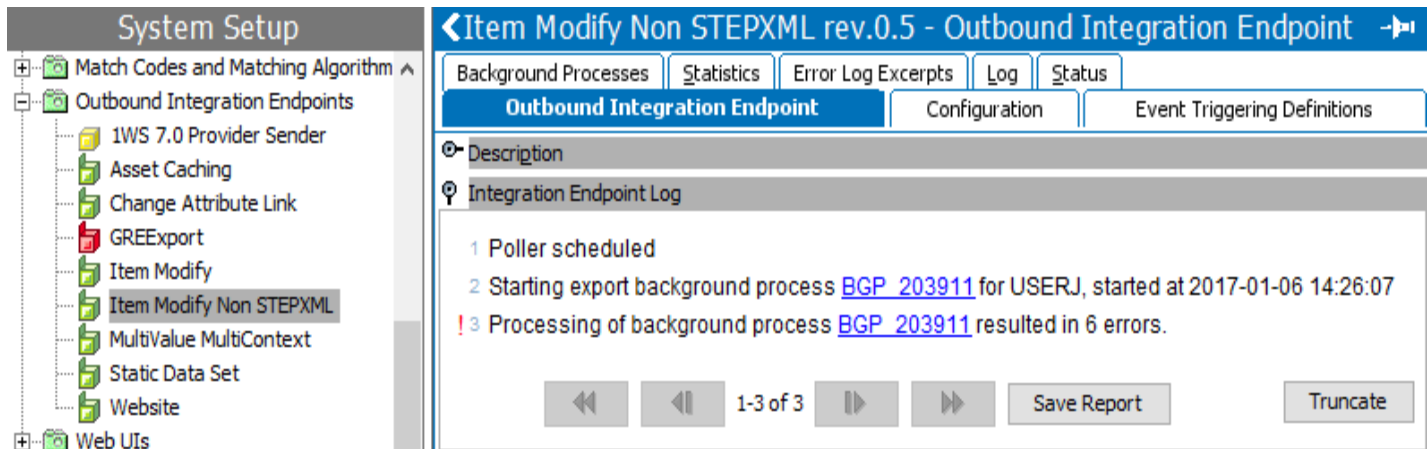
- If scheduled, the Schedule parameter shows when the IIEP will be invoked automatically.

Background Processes	Statistics	Error Log Excerpts	Log	Status
Outbound Integration Endpoint		Configuration	Event Triggering Definitions	
Configuration				
Process Engine	STEP Exporter			
Error reporter	Not Defined			
Schedule	Start every minute			
Queue for endpoint	OutboundQueue			
Queue for endpoint processes	Out			
Transactional settings	Strict			
Number of threads	1			
Maximum number of old processes	1000			
Maximum age of old processes	1y			
Contexts	English US			
Workspace	Approved			
Event Queue Configuration				
Output Templates				
Delivery Method				

- If 'Not scheduled', or if the IIEP needs to be invoked on demand, right-click the enabled IIEP and click **Invoke Endpoint**.



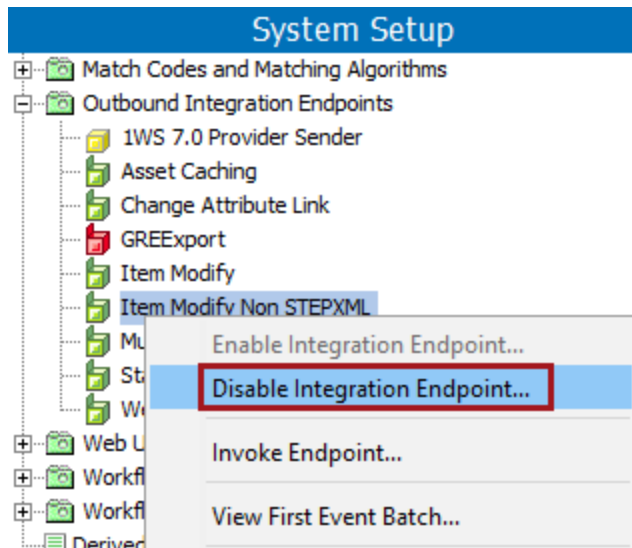
- When invoked on schedule and manually, files or messages ready for export as identified by the OIEP configuration are processed. The Outbound Integration Endpoint tab Integration Endpoint Log flipper displays the time the endpoint was invoked. For scheduled OIEPs, the log includes a link to the background process that was started. For more information, see **Maintaining an Outbound Integration Endpoint**.



Disable an Outbound Integration Endpoint

When an OIEP displaying the enabled icon (🟢) should be stopped from processing data, you can disable it. The configuration remains unchanged and no further data is processed. After selecting the OIEP, right-click and select **Disable Integration Endpoint** to set the OIEP to disabled and display the disabled icon (🟡).

For example, disable an endpoint when running a data refresh activity on the system and the objects should not be published. Disabling an endpoint does not alter the configuration.

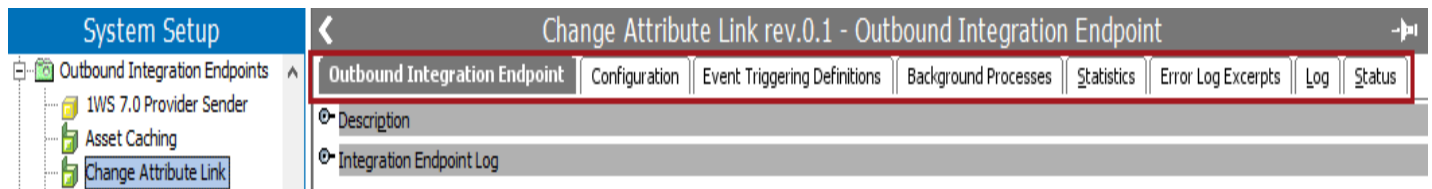


Once disabled, the outbound integration endpoint can be restarted using the Enable Integration Endpoint step defined above.

Additional information about restarting a failed endpoint, indicated by the failed icon () , can be found in the **Handling Failed OIEP Background Processes** documentation.

Maintaining an Outbound Integration Endpoint

Outbound integration endpoint settings can be viewed and edited from the Outbound Integration Endpoint editor. For information about creating a new outbound endpoint, see the **Creating an Outbound Integration Endpoint** section.



Each tab on the editor is defined below and includes parameters to maintain the OIEP.

Outbound Integration Endpoint Tab

This tab holds basic information within the following flippers:

Description Flipper

The Description flipper includes basic information to identify the OIEP. The name, user, description, and enabled parameters can be edited.

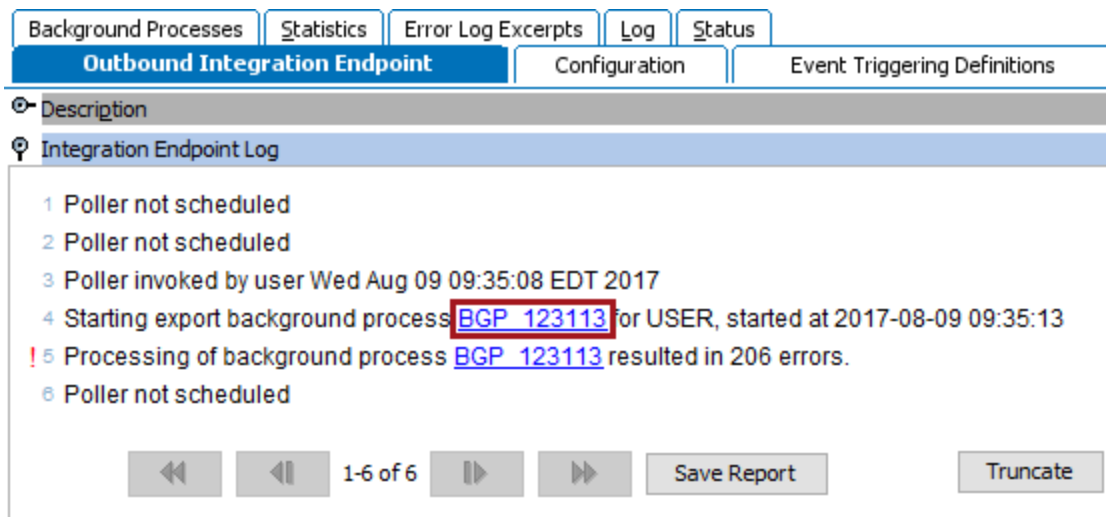
For information on the Endpoint Status, see **Running an Outbound Integration Endpoint** documentation.

Outbound Integration Endpoint		Configuration	Event Triggering Definitions	Background Processes	Statistics	Error Log Excerpts	Log	Status
Description								
Name	>	Value						
ID	>	Change Attribute Link						
Name	>	Change Attribute Link						
Object Type	>	Outbound Integration Endpoint Type						
Revision	>	0.1 Last edited by Stiibo Systems on Wed Jan 27 11:43:03 EST 2016						
User	>	User (USER)						
Description	>							
Enabled	>	Yes						
Endpoint Status	>	Running						
Last run	>	2015-10-23 16:36:40						
Next run	>	9998-01-12 00:31:00						
Integration Endpoint Log								

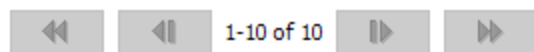
Integration Endpoint Log Flipper

The Integration Endpoint Log flipper contains information about the endpoint background process. An entry is generated each time the endpoint is invoked, when background processes generated by the endpoint are started,

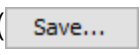
and when processing errors occur.

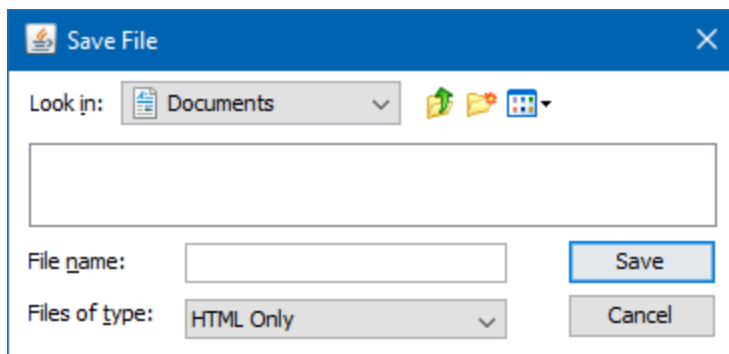


- **BGP link** - The background process opens the individual Importer Background Processes.
- **Navigation** - After the first entry, the Execution Report displays the following navigation buttons:

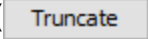


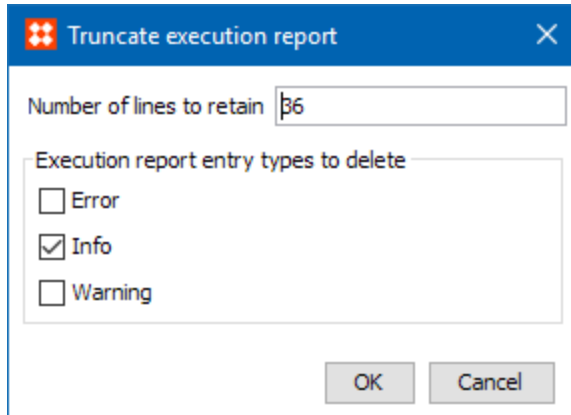
The buttons are enabled when until the log reaches over 100 lines.

- 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.
- **Save** - Click the Save button () to save the Execution Report as an 'HTML Only' file type, then enter a name for the log report and click **Save**.



Note: Changing the default format manually may result in an error.

- **Truncate** - When the Execution Report become cluttered or only specific information is needed, use the Truncate option to permanently eliminate selected rows. Click the Truncate button () to display the Truncate execution report dialog:



The dialog box titled "Truncate execution report" has a close button (X) in the top right corner. It contains a text input field labeled "Number of lines to retain" with the value "36" entered. Below this is a section titled "Execution report entry types to delete" containing three checkboxes: "Error" (unchecked), "Info" (checked), and "Warning" (unchecked). At the bottom are "OK" and "Cancel" buttons.

- In **Number of lines to retain**, the current number of log entries will display. Type the number of lines to keep starting from the end of the log, and permanently deleting the oldest rows, based on the selected entry types below.
- In **Execution report entry types to delete**, select the types that should be permanently removed.

Note: The Properties flipper includes a count of the Warnings and Errors included in the Execution Report.

- Click **OK** to permanently remove the selected number and types of execution report entries.

Configuration Tab

This tab contains many parameters required to successfully run an outbound integration endpoint.

Configuration Flipper

Most of the parameters on this flipper can be edited. The type of editing available varies based on the parameter as defined below.

Outbound Integration Endpoint	Configuration	Event Trigg
Configuration		
Process Engine	STEP Exporter	
Error reporter	Not Defined	
Schedule	Not scheduled	
Queue for endpoint	OutboundQueue	
Queue for endpoint processes	Out	
Transactional settings	Strict	
Number of threads	1	
Maximum number of old processes	100	
Maximum age of old processes	1w	
Contexts	Danish DK, English US	
Workspace	Approved	
Event Queue Configuration		
Output Templates		
Delivery Method		

The **Transactional setting** parameter cannot be edited and does not have the same significance as for IIEPs. For information, see **OIEP - Event-Based - Configure Endpoint** or **OIEP - Select Objects - Configure Endpoint**.

The following parameters can be edited within the OIEP editor as defined:

- Click the **Process engine** or **Workspace** parameter to display a dropdown button. Selections displayed depend on your system setup.

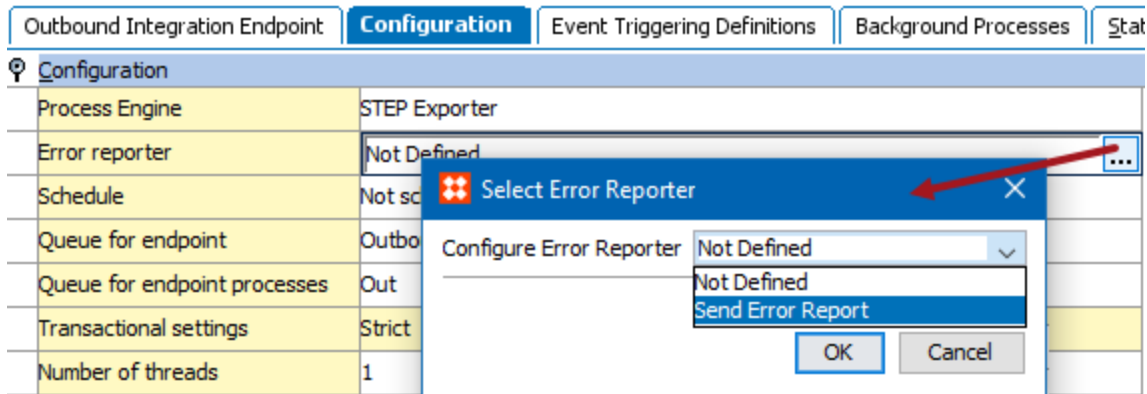
For information on the Process engine or Workspace, see **OIEP - Event-Based - Configure Endpoint** or **OIEP - Select Objects - Configure Endpoint**.

Workspace	Approved	▼
Event Queue Configuration	Approved	
Output Templates	Main	
Delivery Method	Staging	

- Click the **Error reporter**, **Schedule**, or **Contexts** parameter to display an ellipsis button (...). This opens a dialog for editing.

For information on the Error Reporter or Schedule, see the Error Reporter section of the **OIEP - Configuration Flipper**.

For information on Contexts, see **OIEP - Event-Based - Configure Endpoint** or **OIEP - Select Objects - Configure Endpoint**.



- Click the **Queue for endpoint**, **Queue for endpoint process**, **Number of threads**, **Maximum number of old processes**, or **Maximum age of old processes** parameter to edit the text with the field.

For information on the Transactional settings, Number of threads, Contexts, and Workspace, see **OIEP - Event-Based - Configure Endpoint** or **OIEP - Select Objects - Configure Endpoint**.

Outbound Integration Endpoint	Configuration	Event Triggering Definitio
Configuration		
Process Engine	STEP Exporter	
Error reporter	Not Defined	
Schedule	Not scheduled	
Queue for endpoint	OutboundQueue	
Queue for endpoint processes	Out	

Event Queue Configuration Flipper

The additional configuration flipper displayed is determined by the type of OIEP being viewed. Event-based OIEPs include the Event Queue Configuration flipper.

- The **Edit Configuration** link allows modification to the parameters displayed.
- The **Click to estimate** button displays a list of unread events.
- For more information on the Event Action buttons, see **Event-Based OIEP Forward, Rewind, Purge, and Republish**.

Outbound Integration Endpoint
Configuration
Event Triggering Definitions
Background F

Configuration

Event Queue Configuration

Event Actions:
Forward
Rewind
Purge
Republish

> Days to retain events	0
> Number of events to batch	1000
> Number of event batches to include per delivery	1
> Queue Status	Read Events
> Unread events (approximated)	Click to estimate ...

Edit Configuration

Output Templates

Delivery Method

Object Selection Configuration Flipper

the Object Selection Configuration flipper is only displayed when the selected OIEP uses the Select Objects data source, as defined on the Choose Data Source step of the wizard. For more information, see **OIEP - Select Objects - Choose Data Source**.

- The **Edit Configuration** link opens a dialog that allows modification to the objects selected for output. For details on the available options, see **OIEP - Select Objects - Object Selection Configuration Flipper**.

Outbound Integration Endpoint
Configuration
Background Processes
Statistics
Error L

Configuration

Object Selection Configuration

ID	Name	Object Type	Path
> 18209	Cotton T-Shirts	Item Family	Primary Product Hierarchy/Products/A...

Edit Configuration

Output Templates

Delivery Method

Output Templates Flipper

This flipper allows selection of object or event types, in addition to the format required for output. Preprocessors and postprocessors can also be set and availability is based on the system configuration.

- Click into a field to display an ellipsis button. Click the ellipsis button (...) to display the relevant editor.
- Click the **Add configuration** link to insert a new row in the table.

Outbound Integration Endpoint	Configuration	Event Triggering Definitions	Background Processes	Statistics
Configuration				
Event Queue Configuration				
Output Templates				
Object-Eventtype	Format	Pre-Processor	Post-Processor	
Sales Item Folder (M...	STEPXML	None	Copy Context Dependent Values and References	
Add configuration				
Delivery Method				

For details on the output templates, see **OIEP - Event-Based - Output Templates Flipper** or **OIEP - Select Objects - Output Templates Flipper**.

Delivery Method Flipper

Details about the delivery method selected are displayed.

- Click the **Edit Delivery** link to modify the parameters.

Outbound Integration Endpoint	Configuration	Event Triggering
Configuration		
Event Queue Configuration		
Output Templates		
Delivery Method		
Copy to directory		
Directory	MultiValuePostProcessor	
File Name Template	\$filename-\$timestamp.\$extension	
Zip content	No	
Edit Delivery		

For details on the available options, see **OIEP - Delivery Method Flipper**.

Background Processes Tab

This tab holds information about any related processes. Click the Background Process ID link to open the individual exporter background process.

- Queued Processes** are background processes that are started in the state 'waiting'. The maximum number of background processes that can be started in the state waiting is specified in the integration endpoint configuration.
- Active Processes** are background processes that are currently being processed. If data is processed as strict or chained, the endpoint only has one active background process at a time. Otherwise, it is possible to apply more background process queues to work concurrently. In this case the endpoint can have more active background processes at a time.

- **Failed Processes** are background processes that are no longer running. For more information, see **Handling Failed OIEP Background Processes**.
- **Completed with errors** have run, but an error has been detected.
- **Ended Processes** have completed successfully. The number of ended processes is determined by the configuration.

For more information, see the **Monitoring an OIEP via Background Processes** section.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status	
Queued Processes							
Id	Description	Status	Progress	Start Date	Started By		
Active Processes (2)							
Id	Description	Status	Progress	Start Date	Started By	Errors	Created
BGP_197160	Export started for endpoi...	failed	0%	Tue Nov 15 12:36:51 EST ...	USERJ	2	Tue Nov 15 12:36:46 E
BGP_197175	Export started for endpoi...	failed	35%	Tue Nov 15 13:32:45 EST ...	USERJ	2	Tue Nov 15 13:32:40 E
Completed with Errors							
Id	Description	Status	Progress	Start Date	Started By		
Ended Processes (7)							
Id	Description	Status	Progress	Start Date	Started By	Errors	Created
BGP_197164	Export started for endpoi...	succeeded	100%	Tue Nov 15 12:40:42 EST ...	USERJ	0	Tue Nov 15 12:40:37
BGP_197165	Export started for endpoi...	succeeded	100%	Tue Nov 15 12:41:32 EST ...	USERJ	0	Tue Nov 15 12:41:27

Statistics Tab

This tab provides statistics about the number of requests handled, run times, and wait times. The information cannot be edited nor exported.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
🔑 Statistics						
Last run	2016-11-15 13:39:12					
Next run	9998-01-12 00:31:00					
Endpoint uptime	1 month 3 weeks 1 day 3 hours 28 m 14 s					
Number of handled requests	9					
Waiting in data source	0					
Requests waiting to be processed	0					
Number of failed requests	2					
Number of running requests	0					
Minimum wait time	5 s					
Mean wait time	5 s					
Maximum wait time	5 s					
Minimum run time	-					
Mean run time						
Maximum run time	2 s					

Error Log Excerpts Tab

This tab shows data from the main Java log file related to failed background processes with Log Level > 'Info'. Click a hyperlink to a failed background process and correct the cause of the failed background process. For more information, see **Handling Failed OIEP Background Processes**.

Inbound Integration Endpoint	Background Processes	Statistics	Error Log Excerpts	Log	Status
🔑 Integration Endpoint Log					
Background Pro...	>	Log Item No	>	Text	
> BGP_206613		130		Row 2: Parent classification 'AddressRoot' for classification '<a href="step	
> BGP_206613		150		Row 3: Parent classification '18203' for classification '<a href="step://clas	

Log Tab

This tab provides information about changes to the OIEP configuration by a user, including the timestamp.

Showing page 1 of 1

2016-10-12 09:01:53 'USERJ': Created
2016-10-12 09:01:53 'USERJ': Modified
2016-10-12 09:54:18 'USERJ': Modified
2016-10-12 09:54:20 'USERJ': Modified
2016-10-12 15:51:04 'USERJ': Modified
2016-10-12 15:51:47 'USERJ': Modified
2016-10-12 15:52:39 'USERJ': Modified
2016-11-15 12:36:12 'USERJ': Modified
2016-11-15 12:36:28 'USERJ': Modified
2016-11-15 12:40:30 'USERJ': Modified
2016-11-15 12:43:32 'USERJ': Modified

Status Tab

This tab provides information about revisions, hidden values and diagnostics.

Revisions						
Revision	Created	Edited	Major	User	Comment	
> 0.3	Thu Dec 15 11:01:37 EST ...	Thu Dec 15 11:01:39 EST ...		USER6	Auto Generated	
> 0.2	Tue Nov 15 12:35:48 EST ...	Tue Nov 15 13:39:12 EST ...		USERJ	Auto generated - elapsed ...	
> 0.1	Wed Oct 12 09:01:53 EDT...	Wed Oct 12 15:52:39 EDT...		USERJ		
Hidden values						
Diagnostics						

Right-click on the > to the left of a revision row to display a menu that allows you to revert to the changes for the selected revision or delete (purge) the selected revision.

Background ProcessesStatisticsError Log ExcerptsLogStatus

Outbound Integration EndpointConfigurationEvent Triggering Definitions

Revisions

Revision	Created	Edited	Major	User	Comment
> 0.3	Mon Aug 07 15:28:59 EDT...	Mon Aug 07 15:43:32 EDT...		USERJ	Auto generated - el...
	5:38:15 EDT ...	Fri Jun 02 14:22:44 EDT 2...		USERJ	Auto generated - el...
	8:56:43 EST ...	Tue Dec 06 12:34:23 EST ...		USERJ	

CutCtrl+X

CopyCtrl+C

PasteCtrl+V

Paste LinkCtrl+L

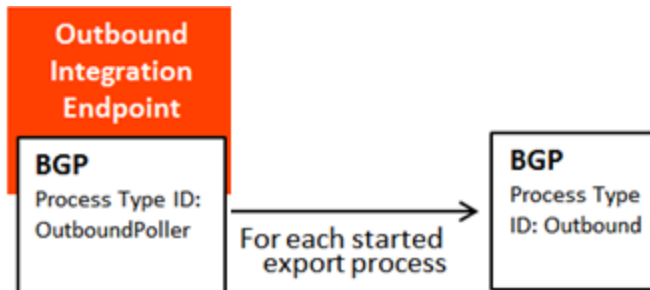
Purge

Revert to

For more information on revisions, see the **Managing Revisions in STEP** topic in **System Setup / Super User Guide** documentation.

Monitoring an OIEP via Background Process

An active outbound integration endpoint uses an associated background process to handle the scheduled invocation of the endpoint. The execution report for this background process is integrated in the endpoint editor and includes invocation information as well as messages logged from the plugins. When using the standard STEP Exporter processing engine, the actual export is handled in separate background processes started by the endpoint, as illustrated below.



By default, the started integration endpoint background processes are displayed within the OIEP editor on the Background Processes tab, and are not visible on the BG Processes tab. Click the background process hyperlink to display details about the actual process. For more information, see **Maintaining an Outbound Integration Endpoint**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
Queued Processes						
Id	Description	Status	Progress	Start Date	Started By	
Active Processes (2)						
Id	Description	Status	Progress	Start Date	Started By	Errors
BGP_197160	Export started for endpoi...	failed	0%	Tue Nov 15 12:36:51 EST ...	USERJ	2
BGP_197175	Export started for endpoi...	failed	35%	Tue Nov 15 13:32:45 EST ...	USERJ	2
Completed with Errors						
Id	Description	Status	Progress	Start Date	Started By	
Ended Processes (7)						
Id	Description	Status	Progress	Start Date	Started By	Errors
BGP_197164	Export started for endpoi...	succeeded	100%	Tue Nov 15 12:40:42 EST ...	USERJ	0
BGP_197165	Export started for endpoi...	succeeded	100%	Tue Nov 15 12:41:32 EST ...	USERJ	0

When configuring an outbound integration endpoint that uses the STEP Exporter processing engine, you will specify a background process queue to use for each of the two types of background processes:

- Queue for endpoint
- Queue for endpoint processes

Typically, all background processes of the same type (same process type ID) will use the same queue. However, integration endpoint processes can be tied to different integration endpoints and use different queues.

Selecting the Background Process Queue

The main background process uses the default **OutboundQueue** background process queue for OIEPs.

However, when you create an integration endpoint, you can specify that you want to use a different OIEP background process queue for the main OIEP background process queue. For more information about this setting, see the **OIEP - Configuration Flipper** documentation.

Background Process Queue Size Property

The queue size setting can have great impact on processing. The background process queue size property determines the number of background processes can be executed in parallel on the queue. If the size is 1, only one background process can run at a time. If the size is 2, two processes can run in parallel, and so forth.

Important: Although the integration endpoint configuration wizards suggest the number of queues to use, beware that if you create multiple endpoints and do not change the default queue suggestions, your setup could include endpoints that block each other since the queues have a default size of 1.

This is not a big concern for the endpoint processes that, most of the time, are idle and only take up a slot on the queue briefly when the endpoint is invoked. However, it can be an issue for the generated background processes that perform the actual export processing. The decision as to whether different endpoints should use the same or different processing queues and the size of the queues should therefore be an informed one.

Overwriting default values for a queue causes a new queue to be created with the queue size of 1. Use the following `sharedconfig.properties` entry to modify the default queue size value:

```
BackgroundProcess.Queue.[name of queue].Size = [number of allowed parallel
processes]
```

Monitoring an OIEP via External Systems

Integration endpoints can be monitored from external systems using Monitoring Sensors or the REST API. Nagios is one application that can monitor the status of integration endpoints.

Monitoring Sensors

For each integration endpoint created in STEP, a Monitoring Sensor is automatically created. Monitoring Sensors allow external systems to query the status of individual endpoints via HTTP, without authentication. In the following sample URLs, substitute your own system URL and endpoint ID to access your own endpoints.

A Monitoring Sensor can return the following information:

1. **Simple traffic light response** returns OK, WARNING, CRITICAL, or UNKNOWN for outbound endpoints via:

```
http://[System URL]/admin/monitoring/OutboundIntegrationEndpointStatus-
[Endpoint ID]/status
```

2. **Nagios friendly response** returns detailed statistics in Nagios friendly format for outbound endpoints via:

```
http://[System URL]/admin/monitoring/OutboundIntegrationEndpointStatus-
[Endpoint ID]/nagios
```

3. **Full detailed XML response** returns XML with detailed statistics for outbound endpoints via:

```
http://[System URL]/admin/monitoring/OutboundIntegrationEndpointStatus-
[Endpoint ID]/xml
```

REST API Monitoring

An alternative to Monitoring Sensors is to use the REST API, which requires authentication.

1. **Overview of all** configured integration endpoints via:

```
[Host]/restapi/integrationendpoints?context=[Context]&workspace=[Main]
```

2. **Detailed statistics** for specific endpoint via:

```
[Host]/restapi/integrationendpoints/[Endpoint ID]?context=[Context]&workspace=
[Main]
```

3. **Execution report** for specific endpoint via:

```
[Host]/restapi/integrationendpoints/[Endpoint ID]/log?context=
[Context]&workspace=[Main]
```

4. **Java log entries related to errors in Background Processes** generated by specific endpoint via:

```
[Host]/restapi/integrationendpoints/[Endpoint ID]/errorexcerpts?context=
[Context]&workspace=[Main]
```

5. **Overview of Background Processes** generated by specific endpoint via:

```
[Host]/restapi/integrationendpoints/[Endpoint ID]/backgroundprocesses?context=[Context]&workspace=[Main]
```


Additional Information About All Outbound Integration Endpoints

This information is relevant to understanding additional functionality available for implementing and using any outbound integration endpoint. Included are details on the following features:

- Export outbound integration endpoint definition for comparison purposes in an external source control system in **Exporting Outbound Integration Endpoint Definitions as Comments**.
- Understand how to handle failed BGPs for outbound integration endpoints as described in **Handling Failed OIEP Background Processes**.
- Understand each element of an OIEP as described in **Outbound Integration Endpoint Structure**.

Exporting Outbound Integration Endpoint Definitions as Comments

Integration endpoint definitions can be exported as comments using Advanced STEPXML. These exports are intended to be used for submission to external source control systems for comparison purposes. Users can import them into source code repository systems where they can be compared from version to version.

Editing and/or importing of these files is not supported (e.g., users may not export, edit the comments, and re-import in STEP). However, the exported file can be imported into another STEP system, which will result in creation of the same endpoint. For example, integration endpoints are typically built in a test or development system before being implemented in the production system. Once creation and testing is complete and satisfactory, the endpoint can be exported from the test environment and then imported into the production environment. This results in the same endpoint on both systems.

To export integration endpoint definitions (both inbound and outbound) for external comparison, use Advanced STEPXML and set the DefinitionsAsComments tag to 'true', as shown below.

On the Select Format step of the outbound tool (either Export Manager or OIEP), choose the Advanced STEPXML format, then copy and paste the following text into the Template field:

```
<?xml version='1.0'?>
<STEP-ProductInformation DefinitionsAsComments="true">
<IntegrationEndpoints ExportSize="All"/>
</STEP-ProductInformation>
```

Select Format

Advanced STEPXML

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

Template

```
<?xml version='1.0'?>
<STEP-ProductInformation DefinitionsAsComments="true">
<IntegrationEndpoints ExportSize="All"/>
</STEP-ProductInformation>
```

An example of the output for an outbound integration endpoint definition as comments is shown below:

```
<OutBoundIntegrationEndpoint...>
<!--Definition: [This will be the IEP definition. Removed for brevity.-->
[Remaining configuration is included]
</OutBoundIntegrationEndpoint>
```

Note: The content of the comment field is not part of the STEPXML XSD and therefore Stibo Systems reserves the right to change the format of the output content at any time.

For more information, see the **STEP-ProductInformation Tag in STEPXML** section of the **Data Exchange** documentation.

Additionally, details about integration endpoint configuration can be exported using STEPXML. For information, see the Configuration > **Include Integration Endpoints** section of the **STEPXML Outbound Parameters** documentation.

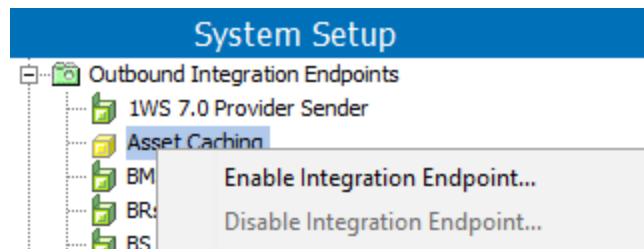
Handling Failed OIEP Background Processes

When a background process encounters an error, it is reported in the log. Errors can be reviewed in the log, resolved, and then the failed integration endpoint can be reactivated using Disable / Enable or Resume.

Disable and Enable a Background Process

Setting an integration endpoint to **Disable** and then **Enable** *does clear* the log file.

- Right-click the endpoint and select **Disable Integration Endpoint**.
- Right-click the endpoint again, and select **Enable Integration Endpoint**.

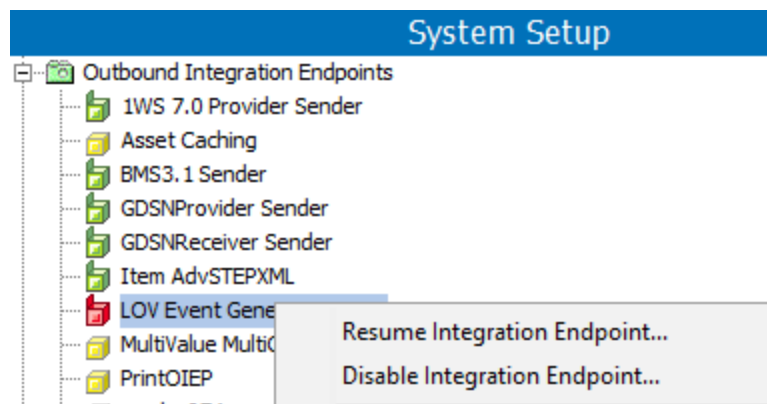


Resume a Background Process

Resuming an integration endpoint *does not clear* the log file

Note: Whenever the STEP system is patched or restarted, the main background process is automatically restarted. The integration endpoint log still exists, and continues to show an overview of the started background processes.

- Right-click the endpoint, and select **Resume Integration Endpoint**.



Resolving a Failed Background Process

Errors can happen before the background process starts, and also after it is running.

OIEP Fails Before the BGP Starts

In cases where the OIEP fails before the BGP is initiated, a report is available in Integration Endpoint Log on the Outbound Integration Endpoint tab. Refer to this log for information to make needed corrections.

LOV Event Generation rev.0.14 - Outbound Integration Endpoint		
Outbound Integration Endpoint	Configuration	Event Triggering Definitions
Description		
Integration Endpoint Log		
145	Starting export background process BGP_232939 for USER4, started at 2017-07-27 10:17:48	
146	Processing of background process BGP_232939 resulted in 1 error.	
147	Starting export background process BGP_232940 for USER4, started at 2017-07-27 10:18:48	
148	Processing of background process BGP_232940 resulted in 1 error.	
149	Starting export background process BGP_232941 for USER4, started at 2017-07-27 10:19:48	
150	Starting export background process BGP_232945 for USER4, started at 2017-07-27 10:22:37	
151	Starting export background process BGP_233013 for USER4, started at 2017-07-31 10:27:38	
152	Caught RuntimeException at Tue Aug 01 10:29:38 EDT 2017: Could not find service for type stibo.OutboundPoller	

OIEP Fails After the BGP Starts

The Error Log Excerpts tab includes details about failures within the background process.

- In **System Setup** select the relevant integration endpoint, and then click the **Error Log Excerpts** tab to display a list of failed background processes and the description of the failure.
You can also see if a process has been suspended because it is dependent on a failed process.
- In the **Background Process** column, hover over the background process to correct and click the **BGP link**.

REST_Direct - Error Log Excerpts		
Outbound Integration Endpoint	Configuration	Background Processes
Statistics		
Error Log Excerpts		
Log		
Status		
> BGP_232539	560	Failed to deliver batch number 1 due to Illegal character in query at index 39: https://pc
> BGP_232539	570	Caught RuntimeException at Tue Jul 18 10:01:36 EDT 2017: com.stibo.exportservices.i
> BGP_232541	560	Failed to deliver batch number 1 due to Illegal character in query at index 39: https://pc
> BGP_232541	570	Caught RuntimeException at Tue Jul 18 10:02:21 EDT 2017: com.stibo.exportservices.i

The Background Process editor displays.

<Export started for endpoint 'REST_Direct' (2017-07-18 10:02:16) - Background P

Background Process
Queue Info

Properties

Property	Value
Started by	USERA
Id	BGP_232541
Description	Export started for endpoint 'REST_Direct' (2017-07-18 10:02:16)
Execution Server	doc-dev
Progress	35%
Status	failed   
Created	Tue Jul 18 10:02:16 EDT 2017
Started	Tue Jul 18 10:02:21 EDT 2017
Finished	Tue Jul 18 10:02:21 EDT 2017
Processing Time	0 m 0 s
Time in Queue	0 m 5 s
# of warnings	0
# of errors	2

Execution Report

```

36 REST delivery, deliver file to https://pdtzxstzvf0.runscope.net
37 Create zip file
! 38 Failed to deliver batch number 1 due to Illegal character in query at index 39: https://pdtzxstzvf0.runscope.net
! 39 Caught RuntimeException at Tue Jul 18 10:02:21 EDT 2017: com.stiibo.exportservices.integrationendpoint

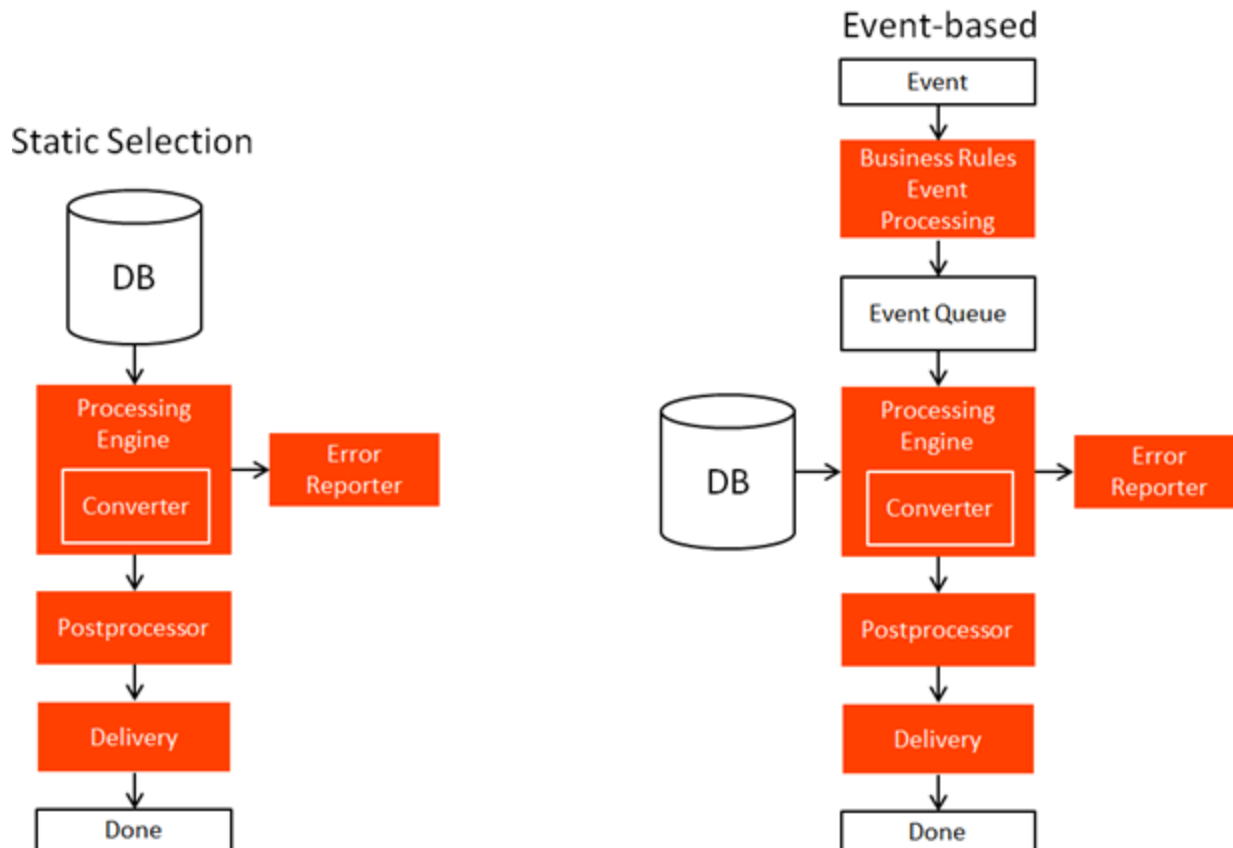
```

- In the Background Process editor **Execution Report**, view the progress of the background process, and where it failed.
- In the Properties flipper > Status parameter, if available, click the Save button () in the Value column, near the failed status, to save the file locally.
- Open the file in a relevant editor and make the required changes.
- After completing the changes, upload the file back to the same background process. In the Background Process editor > Properties flipper > Status parameter Value column, click the Upload button () near the failed status.
- Restart the process in the Properties area > Status parameter Value column, click the Restart button () near the failed status. The background process is restarted.
- If the background process is part of a batch of chained processes, restart the other processes that are part of the chain.

Outbound Integration Endpoint Structure

The OIEP functionality has been created to allow for easy extensions / customizations and can be seen as a plugin framework with interchangeable parts. The graphic below illustrates the structure for both a non-incremental endpoint based on a static selection and an event-based endpoint. The colored boxes represent the interchangeable parts, which are described below.

As the data source, an OIEP can either use a static selection of hierarchy root nodes, or events tied to selected objects in STEP. The delivery options and export formats are the same regardless of the data source selected. By default, this list will be the same as for the Export Manager.



Business Rules Event Processing

For event-based endpoints, business conditions can be used to filter events and business actions can be used to generate new events based on others. For more information, see the **Event Filter** and **Generate Event** sections of the **OIEP - Event-Based - Event Triggering Definitions Tab** documentation.

Processing Engine / Converter

The processing engine is responsible for performing the actual data export based either on events or a static selection. STEP Exporter is the only available option with the core functionality, and is also the standard functionality used for manual exports. If a special format not supported in the standard STEP Exporter is required,

the processing engine can be replaced via an extension. Most often, however, creating a converter plugin for the STEP Exporter is sufficient for such cases. This approach also has the benefit that the same converter plugin is available for manual exports. For more information, see the **Configure Endpoint** documentation for an **Event-Based OIEP** or a **Select Objects OIEP**.

Error Reporter

STEP includes a single Send Error Report plugin to report errors. An email can be sent to a specified address if errors occur when files / messages are handled by the processing engine. For more information, see the **Error Reporter** section of the **OIEP - Configuration Flipper** documentation.

Post-Processor

A post-processor plugin has access to the file delivered by the processing engine and can manipulate it and/or split it into multiple files. STEP includes two post-processor plugins, both used for manipulating files output by the processing engine with cross context information. For more information, see the **Output Templates Flipper** documentation for an **Event-Based OIEP** or a **Select Objects OIEP**.

Delivery

A delivery plugin has access to the file(s) output from the processing engine / post-processor. Additional delivery options can be added with the delivery plugin extension. For more information, see the **OIEP - Delivery Method Flipper** documentation.

Additional Information About Event-Based OIEPs

This information is relevant to understanding additional functionality available for implementing and using an event-based outbound integration endpoint. Included are details on the following features:

- Understand the effects of batching on an OIEP as described in **Event-Based OIEP Event Batching**.
- Understand the effects of manipulating events on an OIEP as described in **Event-Based OIEP Forward, Rewind, Purge, and Republish**.
- Understand the multithreading functionality on an OIEP as described in **Event-Based OIEP Multithreading Support**.
- Understand queued events on an OIEP as described in **Event-Based OIEP Queued Events**.
- Understand the relationship between OIEP Status and Queue Status as described in **Event-Based OIEP Status and Queue Status**.
- View some examples of event-based OIEPs as described in **Event-Based OIEP Examples**.

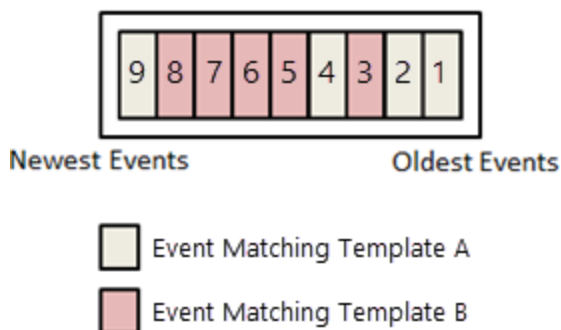
Event-Based OIEP Event Batching

The Event Batching option on an event-based OIEP does not determine the number of objects in a message, but rather, the number of events that will be processed and included in a single message. STEP does not wait for the number of events to occur before generating a message, instead, it caps the number of events in a single message so that 2+ messages are generated if more events are present than the batch number allows when the system is polled.

Unless it is a strict requirement that one file should be exported per event, common setup is to use event batching and leave the default value of 1000, or set it even higher if many events are registered and the endpoint is invoked infrequently. Appropriate batch size is typically based on the size of messages and downstream system processing capabilities, for example, use a smaller batch size for larger messages and a larger batch size for smaller messages.

STEP batches are based on the output template, which includes a combination of event type and object type. For more information, see **OIEP - Event-Based - Output Templates Flipper** documentation.

Consider the unread events in this illustration, which matches either 'Template A' or 'Template B' output templates:



If you work with a batch size of three (3), data related to the events will be published as follows:

- One file with data for Event 1 & 2 (Template A)
- One file with data for Event 3 (Template B)
- One file with data for Event 4 (Template A)
- One file with data for Event 5, 6 & 7 (Template B)
- One file with data for Event 8 (Template B)
- One file with data for Event 9 (Template A)

For more information, see the **Events** section of the **System Setup** documentation.

Event-Based OIEP Examples

This section includes event-based OIEP configuration examples for semi-advanced to advanced cases:

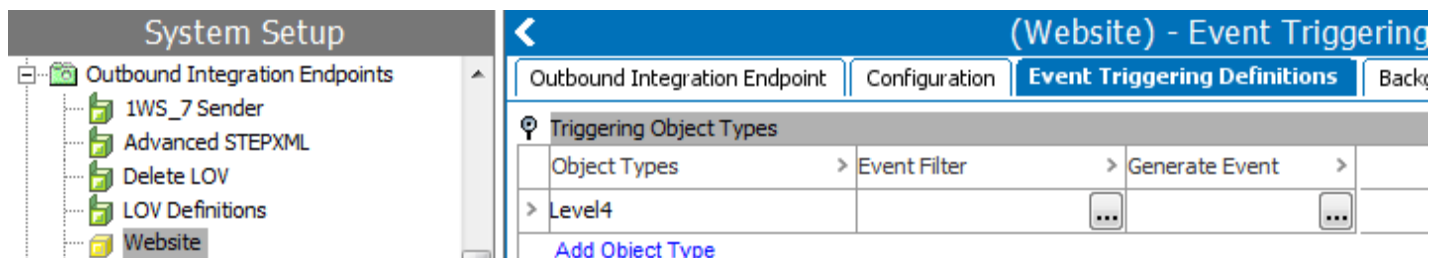
- Event-Based Example Business Rules for Derived Events
- Event-Based Example Publishing of Leaf Products with Inherited Data
- Event-Based Example Upward Inheritance
- Event-Based Example Products Linked to Classification

Event-Based Example Business Rules for Derived Events

Problem: A different output format is required for each derived event type.

Solution: Use JavaScript-based business rules in an event-based OIEP to:

- Create a business condition to filter out (discard) Republish events.
- Create a business action to generate derived events when products are linked, removed, or modified in a specific classification hierarchy and distinguishes if the products have been created, modified, or deleted in the classification structure.
- Apply the business rules to the OIEP on the Event Triggering Definitions tab, in the Triggering Object Types table.



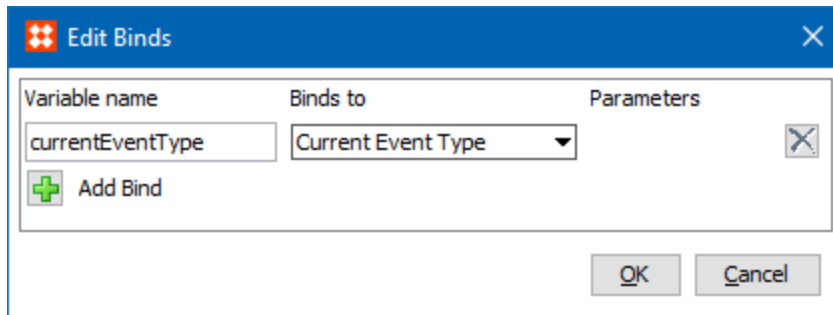
Important: The following scripts are only an example and should not be used as-is without thorough testing, including updating to match object and link types that exist on your system.

The following derived event types are used in this example:

- A product is linked into a classification and approved for the first time. In this case, a derived event with the event type 'website create' is generated.
- A product is approved and linked into a classification, but data on the product is approved with modifications. In this case, a derived event with event type 'website modify' is generated.
- A product is approved and linked into a classification, but approved with a deletion of the product to classification link. In this case, a derived event with event type 'website delete' is generated.

Event Filter - Using a bulk update or business action could generate Republish events, but they would be registered as core 'create' events. Instead, generate a derived event and then configure the relevant OIEP to discard all events that are not of this type.

Create a new **business condition** and make it valid for the appropriate object types. Use the **Evaluate JavaScript** operation and add a bind for the Current Event Type.

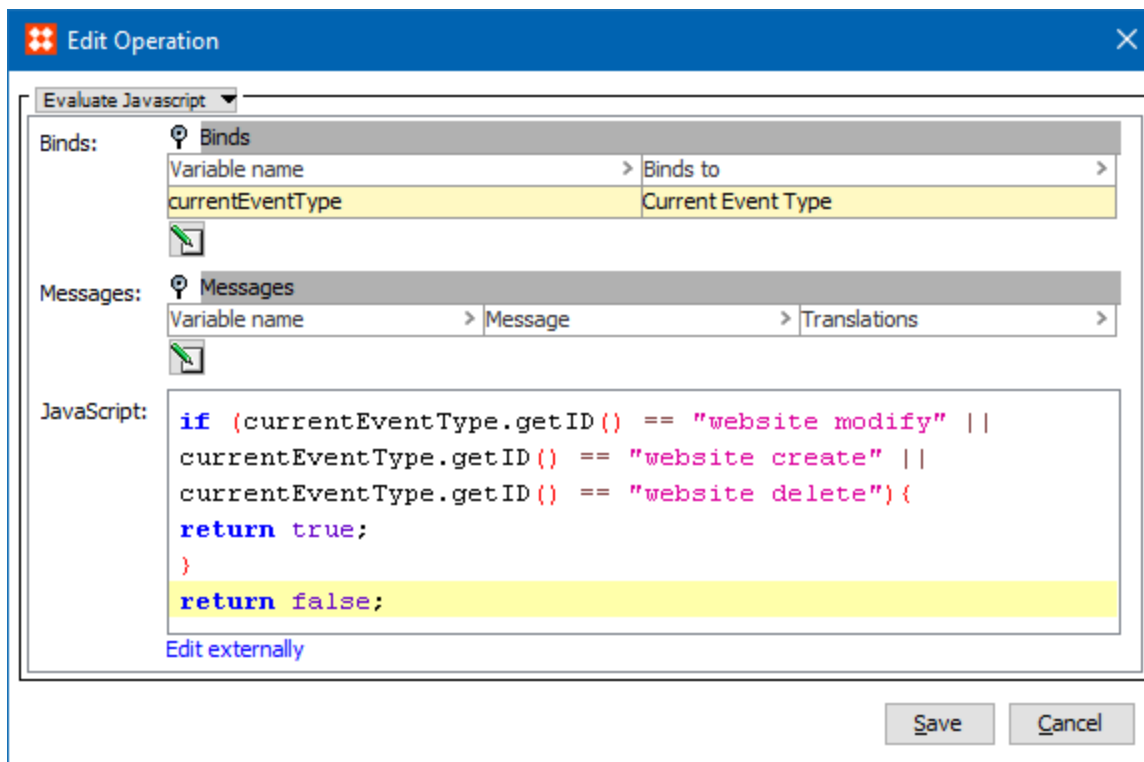


The 'Edit Binds' dialog box has a blue title bar with a close button. It contains three columns: 'Variable name', 'Binds to', and 'Parameters'. The 'Variable name' column has a text box with 'currentEventType'. The 'Binds to' column has a dropdown menu with 'Current Event Type' selected. There is a small 'X' icon in the 'Parameters' column. Below the columns is a green plus icon and the text 'Add Bind'. At the bottom right are 'OK' and 'Cancel' buttons.

Use the following JavaScript code to identify the derived event types:

```
if (currentEventType.getID() == "website modify" ||
    currentEventType.getID() == "website create" ||
    currentEventType.getID() == "website delete"){
    return true;
}

return false;
```



The 'Edit Operation' dialog box has a blue title bar with a close button. It contains a tab labeled 'Evaluate Javascript'. Below the tab are three sections: 'Binds', 'Messages', and 'JavaScript'. The 'Binds' section has a table with columns 'Variable name' and 'Binds to', containing one row with 'currentEventType' and 'Current Event Type'. The 'Messages' section has a table with columns 'Variable name', 'Message', and 'Translations'. The 'JavaScript' section has a text area with the following code:

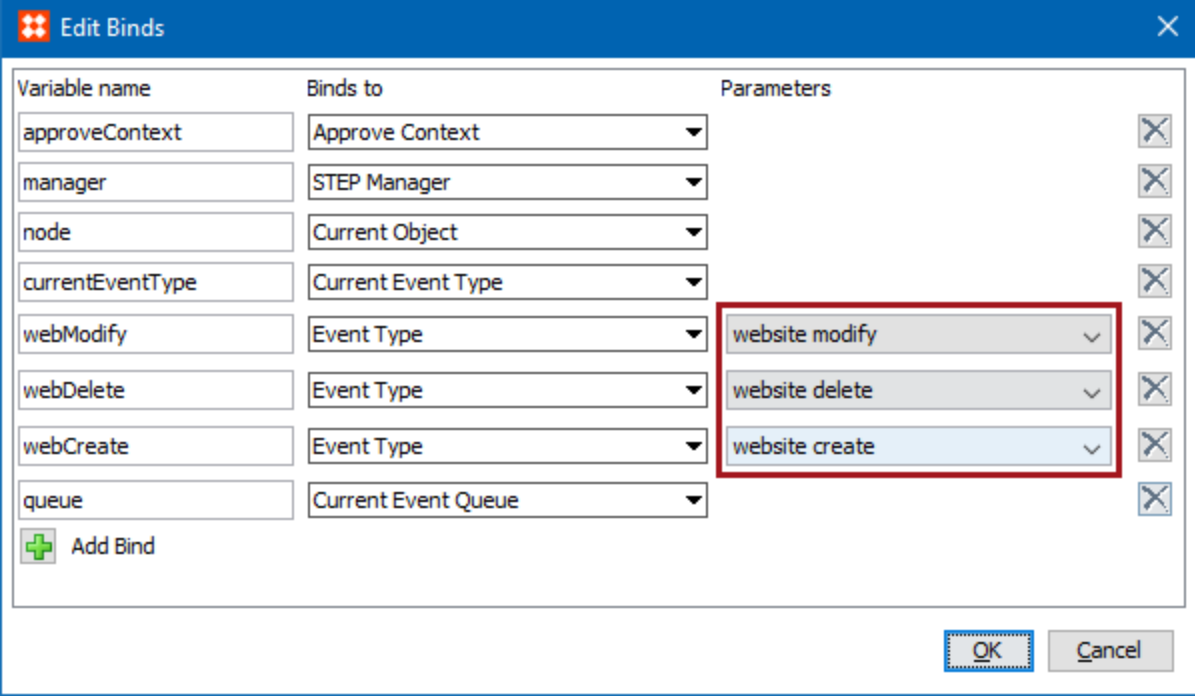

```
if (currentEventType.getID() == "website modify" ||
    currentEventType.getID() == "website create" ||
    currentEventType.getID() == "website delete"){
    return true;
}

return false;
```

 Below the code is a link 'Edit externally'. At the bottom right are 'Save' and 'Cancel' buttons.

Event Generator - Only standard derived event types can be used with JavaScript-based business actions. For additional information on JavaScript syntax for STEP, click the **STEP API Documentation** button on the STEP WebStart page.

Create a new **business action** and make it valid for the appropriate object types. Use the **Execute JavaScript** operation and add the following binds, including binds for the derived events:



Variable name	Binds to	Parameters
approveContext	Approve Context	
manager	STEP Manager	
node	Current Object	
currentEventType	Current Event Type	
webModify	Event Type	website modify
webDelete	Event Type	website delete
webCreate	Event Type	website create
queue	Current Event Queue	

+ Add Bind

OK Cancel

Use the following JavaScript code to create a derived event based on the approved change on the product:

```
if (approveContext){
    var linkType = manager.getLinkTypeHome
    ().getClassificationProductLinkTypeByID("Web Classification");

    var links = node.getClassificationProductLinks().get(linkType);
    var linkedToWebsite = links.size() > 0;
    var linkChanged = false;
    var partObjects = approveContext.getPartObjects();

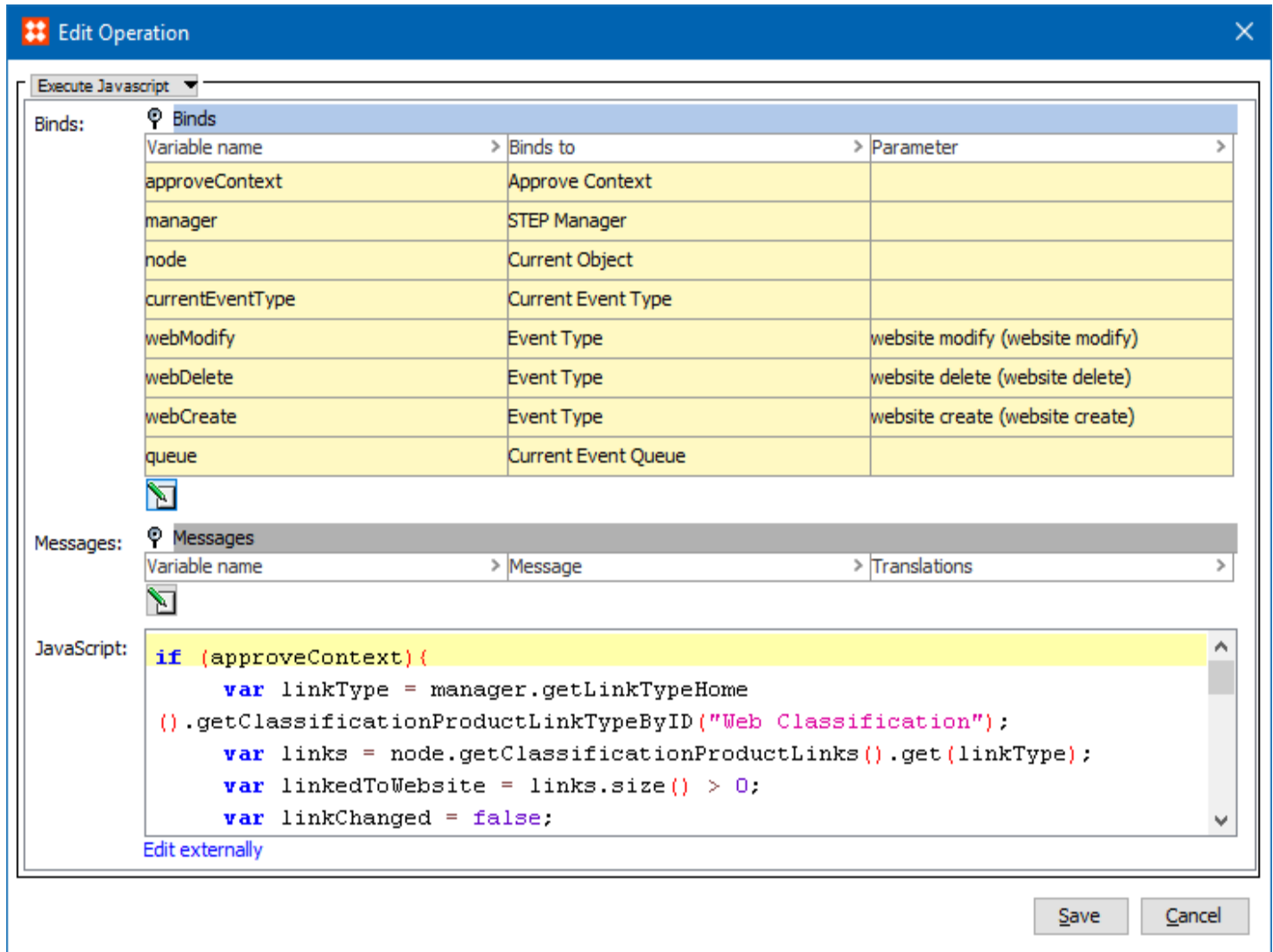
    var iter = partObjects.iterator();
    while (iter.hasNext()) {
        var part = iter.next();
        if (part instanceof
com.stibo.core.domain.partobject.ClassificationLinkPartObject){
            if (part.getLinkTypeID() == "Web Classification"){
                linkChanged = true;
            }
        }
    }

    if (linkedToWebsite && linkChanged){
        queue.queueDerivedEvent(webCreate, node);
    } else if (!linkedToWebsite && linkChanged) {
        queue.queueDerivedEvent(webDelete, node);
    }
}
```

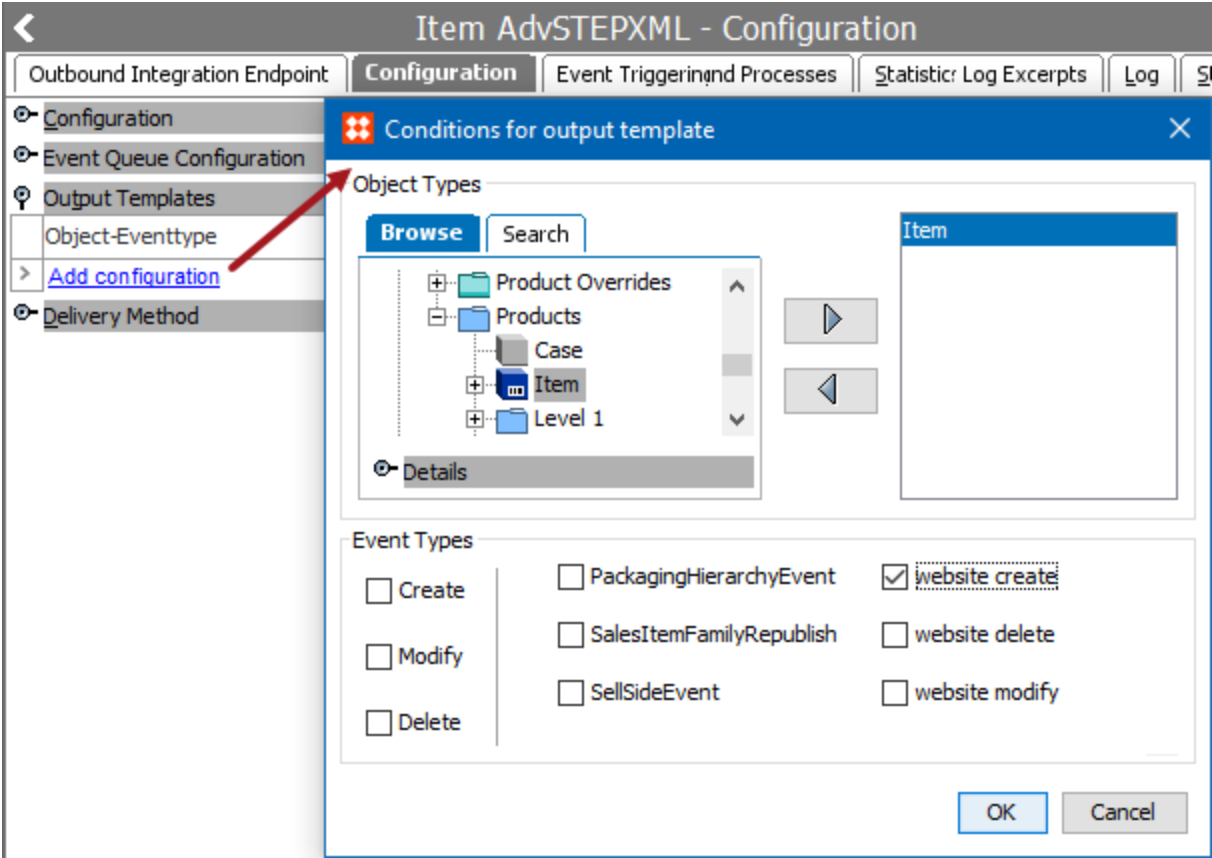
```

        } else if (linkedToWebsite && !linkChanged){
            queue.queueDerived
        }
    }
    Event(webModify, node);
}
}

```



Output Template - Using the configuration described above, you will need an output template for each different type of event. This can be one or multiple lines, dependent on the need for different messages for the different events. See **OIEP - Event-Based - Output Templates Flipper** topic for more information.



Event-Based Example Leaf Products with Inherited Data

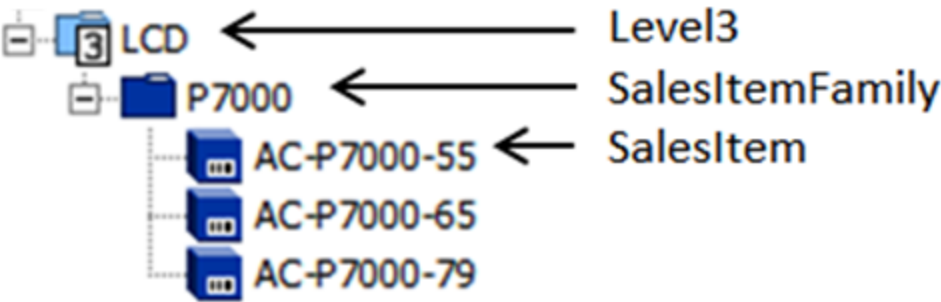
Problem: You only want to publish the leaf product level to a downstream system and the leaves inherit data from ancestors, so events will not be generated for the leaves when inherited data changes.

Solution: When values are inherited from the parent or grandparent level, use Event Filters and Event Generators in an event-based OIEP to ensure that:

- events generated for the parent or grandparent level are discarded
- derived events are generated for all descendant leaf level objects that exist in Approved mode (have been published earlier)

Important: The following scripts are only an example and should not be used as-is without thorough testing.

Using the following object types:



Use the following triggering object types setup:

Triggering Object Types			
Object Types	Event Filter	Generate Event	
> Sales Item		...	...
> Level 3, Sales Item Family	LP Always False (LPAlwaysFalse)	...	LP Republish Sales Item Descendants (...
Add Object Type			

No filters or generators are used for Sales Items, so events generated for this object type always get registered in the queue.

Use the following logic for the grandparent and parent levels (Level 3 and Sales Item Family):

Event Filter - No Events are registered for objects of these object types, but we still need to catch them to be able to generate derived events for descendant Sales Items. The event filter should be a JavaScript business condition that always evaluates to false, as shown below:

Execute Javascript

Bind

Variable name

Bind

Variable name

Bind

Variable name

Bind

Variable name

Bind

Message

Variable name

Message

Translations

JavaScript

return false;

Edit externally

Event Generator - The business action for event generation is a little more complex and uses the following binds (logger is optional):

Execute Javascript

Bind

Variable name

Bind

Variable name

Bind

Variable name

Bind

Variable name

Bind

Variable name

Bind

Variable name

Bind

Variable name

Bind

Variable name

Bind

Variable name

Bind

JavaScript

function prodIsInApp (prodID) {
var appProd;
manager.executeInWorkspace ("Approved", function (appManager) {
appProd = appManager.getProductHome ().getProductByID (prodID);
}
);
}

Edit externally

Notice the 'Event Type' bind to a derived event named 'salesItemRepublish'. This is the event that will be generated for descendant Sales Items and it must be defined in advance via the 'Derived Events' node in System Setup.

Important: The following script is only an example and should not be used as-is without thorough testing. For a script example of the derived events approval, see the online version of this topic.

The script uses as recursive function 'generateEventForApprovedLeafSalesItems' to find descendent Sales items that are in the Approved workspace.

Output Template - Using the configuration described above, you will need an output template for Sales Items and the different types of possible events. This can be one or multiple lines dependent on the need for different messages for the different events.

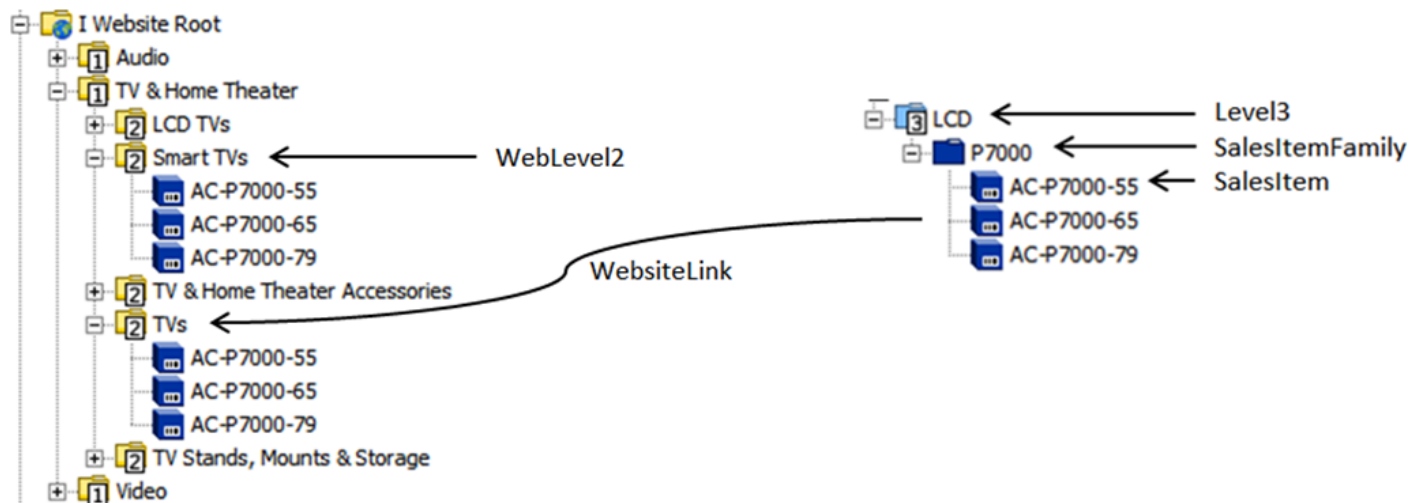
Outbound Integration Endpoint	Configuration	Event Triggering Definitions	Background Processes	Statistics	Error Log Excerpts
Configuration					
Event Queue Configuration					
Output Templates					
Object-Eventtype	> Format	> Pre-Processor	> Post-Processor	>	
> Sales Item (Create, Modify, Delete, salesItemRepublish) ...	Generic XML (4 mappings)	None	Context splitter		
> Add configuration					
Delivery Method					

Event-Based Example Products Linked to Classification

Problem: Sometimes an external system is only interested in data about products linked into a specific classification hierarchy (e.g., a website hierarchy) and needs different types of messages depending on whether a product has just been linked into the hierarchy ('create' to the external system), is in the hierarchy and has been edited ('modify' to the external system) or has been taken out of the hierarchy ('remove / delete' to the external system).

Assumptions: For this example the setup shown below is used and a number of assumptions are made.

- Only leaf level (Sales Item) can be linked into the classification hierarchy and only leaves should be published.
- The Product to Classification Link Type ('WebsiteLink') is only used for this specific website and all Sales Item objects in Approved that have the link should be on the website.



Solution: A naive approach to this problem could be to simply have an event filter condition hooked in for Sales Items that only let events for the objects with a WebsiteLink pass. The problem with this approach is that it would not allow you to detect the cases where a link has just been removed, so you would not be able to communicate this to the external system. Also, you would not be able to differentiate between cases where the link has just been added ('create' to an external system) and cases where a Sales Item already linked into the classification hierarchy has been modified.

A better approach is to use three derived events with IDs 'webAdd', 'webModify', and 'webRemove' in an event-based OIEP. The conditions for triggering the different events are listed below:

- **webAdd** - Sales Item with WebsiteLink is approved for the first time; earlier approved Sales Item where WebsiteLink has been added is approved
- **webModify** - Sales Item with WebsiteLink has been modified and is approved; externally maintained data for Sales Item with WebsiteLink in Approved is modified

- **webRemove** - Earlier approved Sales Item where WebsiteLink has been removed is approved; Sales Item with WebsiteLink is approve deleted (core Delete Event)

Important: The following scripts are only an example and should not be used as-is without thorough testing.

A setup for this case could use the following triggering object types:

Outbound Integration Endpoint

Configuration

Event Triggering Definitions

Background Processes

Statistics

Error Log Excerpts

Log

Statu

?

Triggering Object Types

Object Types	>	Event Filter	>	Generate Event	>
> Sales Item		WP Reject Core Events (WPRejectCoreEvents)	...	WP Generate Derived Events (WPGenerateDerivedEvents)	...
Add Object Type					

Event Filter - As data will be published based on derived events alone, all core events for Sales Items should be discarded. The script below does this and also, for debugging purposes, logs to the main Java log.

Edit Operation

Evaluate Javascript

Variable name

> Binds to

>

currentEventType

Current Event Type

logger

Logger

Messages

Variable name

> Message

> Translations

>

JavaScript

```

if(currentEventType instanceof com.stibo.core.domain.eventqueue.BasicEventType) {
    logger.info("Filter: Discarded core event of type '" + currentEventType.getID() + "'");
    return false;
}
else {
    logger.info("Filter: Let derived event of type '" + currentEventType.getID() + "' pass");
    return true;
}

```

[Edit externally](#)

Save

Cancel

If logging is not required, the script could be:

```

return !(currentEventType instanceof
com.stibo.core.domain.eventqueue.BasicEventType);

```

Event Generator - This logic is much more complex as it will have to perform the analysis to figure out which type of derived event should be generated. An example without logging is shown below. Notice that the business action should only generate derived events based on core events. If this check is not in place, an endless loop will be created.

The script uses the following bindings:

Edit Operation

Execute Javascript

Binds:

Binds

Variable name	Binds to	Parameter
logger	Logger	
currentEventQueue	Current Event Queue	
manager	STEP Manager	
currentEventType	Current Event Type	
appContext	Approve Context	
webAdd	Event Type	webAdd (webAdd)
webModify	Event Type	webModify (webModify)
webRemove	Event Type	webRemove (webRemove)
currentObject	Current Object	

Messages:

Messages

Variable name	Message	Translations

JavaScript:

```

var prodClassLinkTypeID = "WebsiteLink"
var ClassObjTypeID = "WebLevel12";

var linkType = manager.getLinkTypeHome().getClassificationProductLink

function classInApp(classID) {
    var res;
    manager.executeInWorkspace("Approved", function(appManager) {

```

Edit externally

Save
Cancel

Keeping in mind that the following script is strictly to illustrate the solution, and can be difficult to read, this list outlines the intended logic:

- Check if current event is a core event, do nothing if it is a derived event.
- Check if the Approve Context is available. If so, the event is generated based on an approval. If not, the event is based on some other change, for example, externally maintained data being modified.
- If Approve Context is available, use it to get hold of current object in both Main and Approved.
- Check if current core event is a Delete.
- Check for WebsiteLinks in Main and Approved.

© Stibo Systems - Confidential - STEP Trailblazer 8.3

1234

- If current event is a Delete, and current object has a WebsiteLink in Approved, generate a derived 'webRemove' event.
- If current object has WebsiteLink in both Main and Approved, generate a derived 'webModify' event.
- If none of the above, the Approve Context is used to figure out whether the addition or removal of a WebsiteLink is currently being approved (is a part object). Notice that currently you cannot get the link type directly from a ClassificationLinkPartObject and thus, the target classification object type is used to figure out if the link type is correct. Also, because you could risk having a situation where the target classification does not exist in Approved and you therefore would not get the link synchronized even though it is in the part object set, it is checked that the classification exists there.
- If WebsiteLink exists in Main and is to be approved, generate a derived 'webAdd' event.
- If WebsiteLink exists in Approved and is to be approved, generate a derived 'webRemove' event.
- If no Approve Context is available, check whether current object exists in Approved with WebsiteLink. If it does, generate derived 'webModify' event.

Output Template - If different messages are to be output dependent on the selected derived event type, three output templates should be configured as shown below. This can be one or multiple lines, depending on the need for different messages for the different events.

Outbound Integration Endpoint		Configuration	Event Triggering Definitions	Background Processes	Statistics	Error
⊖ Configuration						
⊖ Event Queue Configuration						
⊕ Output Templates						
	Object-Eventtype	> Format	> Pre-Processor	> Post-Processor	>	
>	Sales Item (WebAdd)	Generic XML (4 mappings)	None	Context splitter		
>	Sales Item (webAdd)	Generic XML (4 mappings)	None	Context splitter		
>	Sales Item (webRemove)	Generic XML (1 mappings)	None	Context splitter		
>	Add configuration					
⊖ Delivery Method						

Event-Based Example Upward Inheritance

Problem: For outbound integrations where not only the leaf level is published, sometimes data from children is used on parents, for instance, via calculated attributes. When the children change, the parents are not automatically republished.

For comparison, downward inheritance is typically handled by publishing all object descendants.

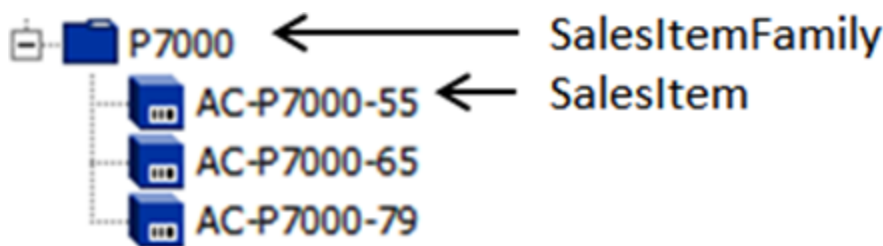
Solution:

In an event-based OIEP:

- If children get exported along with the parent, whenever an event is generated for a child with appropriate parent, catch it, discard it, and generate a derived event for the parent instead.
- If children are not published with parents, do not discard the children events.

Important: The following scripts are only an example and should not be used as-is without thorough testing.

The first option is described below, using the following object types:



Use the following triggering object types setup:

Outbound Integration Endpoint	Configuration	Event Triggering Definitions	Background Processes	Statistics	Error Log Excerpts	Log	Status
Triggering Object Types							
Object Types	Event Filter		Generate Event				
> Sales Item	UI Skip Sales Items Below Family (UISkipSalesItemsBelowFamily)		...	UI Republish Sales Item Family (UIRepublishSalesItemFamily)		...	
> SalesItemFamily			...			...	
Add Object Type							

No Filter or Generator is used for the Sales Item Family Object Type as all Events should pass for this type.

Use the following logic for the Sales Items:

Event Filter - If Sales Items can be children of both Sales Item Family and other types, a filter that checks the parent object type is required. The Script below discards events for Sales Items that are children of Sales Item Family objects:

Edit Operation

Evaluate Javascript

Binds

Variable name

Binds to

currentObject

Current Object

Messages

Variable name

Message

Translations

JavaScript

```
return !currentObject.getParent().getObjectType().getID().equals("SalesItemFamily");
```

[Edit externally](#)

Save

Cancel

If Sales Items can only exist below Sales Item Family objects, instead an 'Always False' condition like the one used in **OIEP Example Publishing of Leaf Products with Inherited Data** can be used instead.

Event Generator - This business action script is also fairly straight forward, notice that it is assumed that a 'Derived Event' with ID 'salesItemFamilyRepublish' has already been defined in STEP.

Edit Operation

Execute Javascript

Binds:

Variable name

Binds to

Parameter

currentObject

Current Object

currentEventQueue

Current Event Queue

salesItemFamilyRepublish

Event Type

SalesItemFamilyRepublish (SalesItemFamilyRepublish)

Messages:

Variable name

Message

Translations

JavaScript:

```
if (currentObject.getParent().getObjectType().getID().equals("SalesItemFamily")) {
    currentEventQueue.queueDerivedEvent(salesItemFamilyRepublish, currentObject.getParent());
}
```

[Edit externally](#)

Save

Cancel

© Stibo Systems - Confidential - STEP Trailblazer 8.3

1237

If Sales Items can only exist below Sales Item Family objects, the 'if' statement can be left out.

Output Template - Using the configuration described above, you will need one or more output templates that cover both Sales item and Sales Item Family, and the different types of possible events must be defined. Below, a single output template is used.

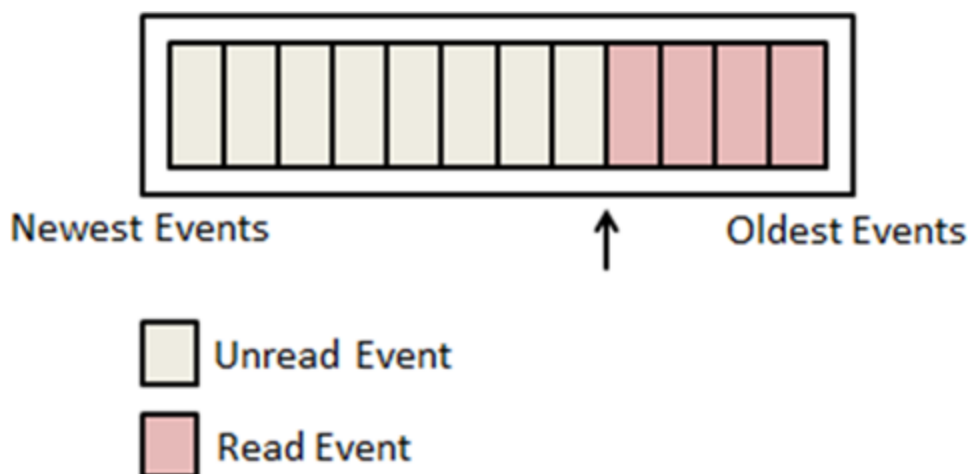
Outbound Integration Endpoint	Configuration	Event Triggering Definitions	Background Processes	Statistics	Error Log Excerpts	Log	?
⊞ Configuration							
⊞ Event Queue Configuration							
⊞ Output Templates							
	Object-Eventtype	>	Format	>	Pre-Processor	>	Post-Processor
>	Sales Item, SalesItemFamily (Create, Modify, Delete, SalesItemFamilyRepublish)		STEPXML		None		Context splitter
>	Add configuration						
⊞ Delivery Method							

Event-Based OIEP Forward, Rewind, Purge, and Republish

Creating an event-based OIEP automatically creates an associated event queue. After configuring the OIEP with Event Triggering Definitions, the OIEP then listens for events that match the configuration and sends those to the related queue. The OIEP polls the queue per the schedule, handles the events, marks them 'read', and publishes the data based on output templates.

The 'Days to retain events' parameter determines how long events are retained in the queue (when they are removed from the queue) after being read. Common setup for this parameter is between 5 and 10 days. When the setting is > 0, read events stay in the queue for the specified number of days. Retaining events allows you to republish them again should the receiving system lose messages. To do so, rewind the event queue and reprocess already processed events from a specific point in time. For information on setting the 'Days to retain events parameter', see the **OIEP - Event-Based - Event Queue Configuration Flipper** documentation.

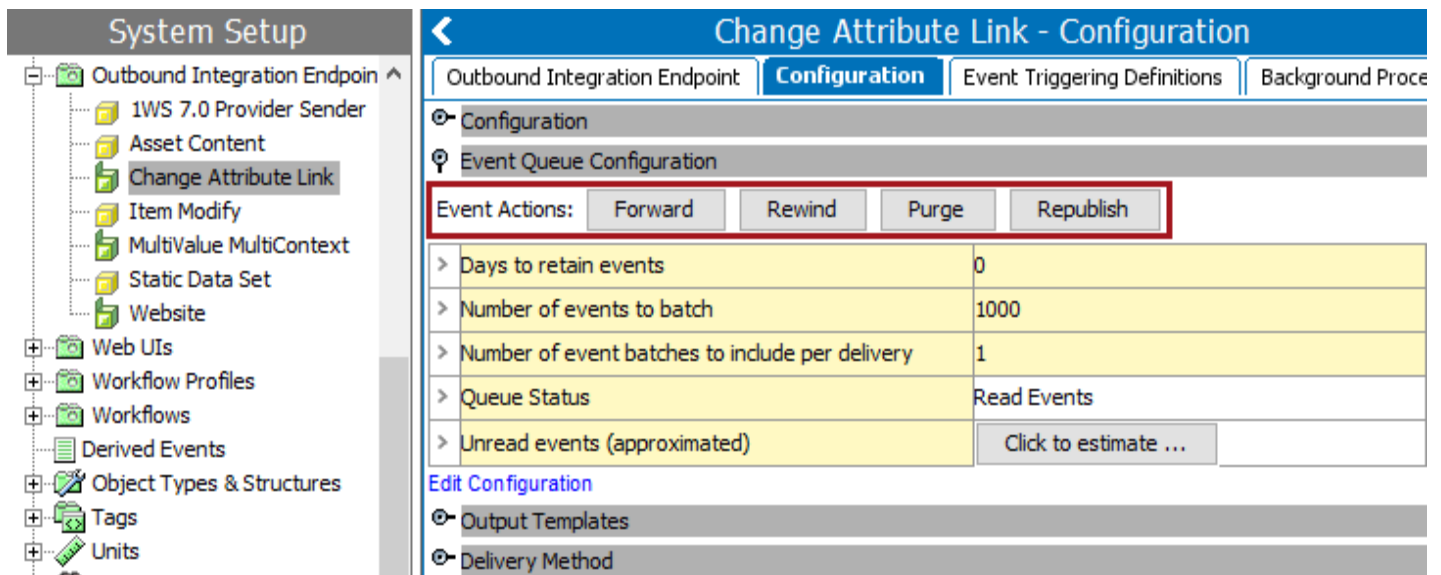
For example, if the queue is configured to retain events after they have been read, read events can be distinguished from unread events, as illustrated by the arrow in the following image.



The Forward and Rewind actions essentially lets the user move this arrow:

- Forward action marks unread Events as read (moving the pointer left)
- Rewind action marks read Events as unread (moving the pointer right)

On the OIEP Configuration tab, open the Event Queue Configuration flipper and use the buttons to modify the queue's events.



For information on the number and types of unread events, see the **Event-Based OIEP Queued Events** documentation.

Forward Events

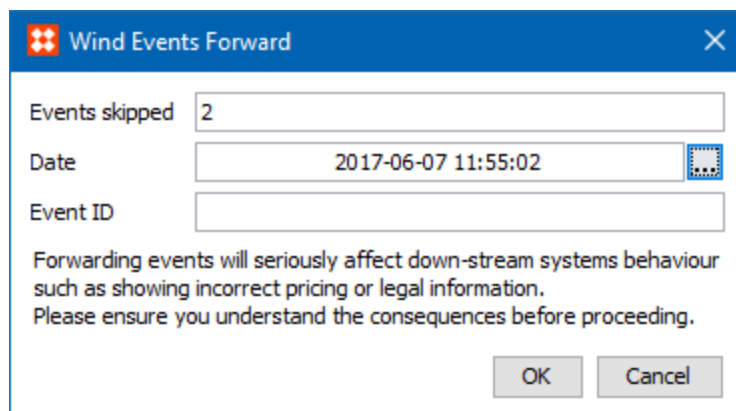
The Forward action requires that unread events exist in the queue. This option is typically used when the unread events in the queue should not be published to downstream systems. For example, consider that you performed a system refresh or data maintenance activity that generated events.

Important: Use caution when forwarding unread events as it may affect the downstream system's data, since the previous changes would be ignored by this activity.

1. In the **Wind Events Forward** dialog, click the ellipsis button (...) in the **Date** field.
2. In the **Date Picker**, select the date and time from which you want to skip events.

When you have selected a date and time, the **Events skipped** field shows the number of events that will be skipped.

Note: Event IDs are not shown in the UI but are read from the database, making this a troubleshooting option.



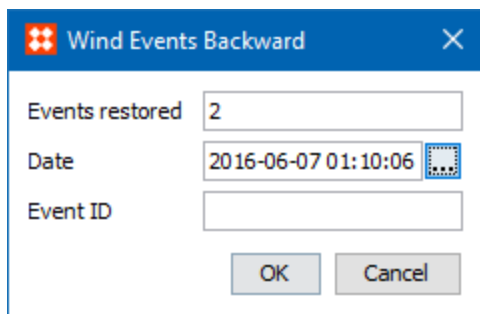
Rewind Events

The Rewind action is only available when the parameter 'Days to retain events' is set to a number greater than zero during creation or configuration of the OIEP. This option is typically used when a downstream system did not receive updates from the OIEP, there was a problem with file delivery, or any failure of the OIEP.

1. In **Wind Events Backward** dialog, click the ellipsis button (...) in the **Date** field.
2. In the **Date Picker**, select the date and time from which you want to restore events.

When you have selected a date and time, the **Events restored** field shows the number of events that will be restored.

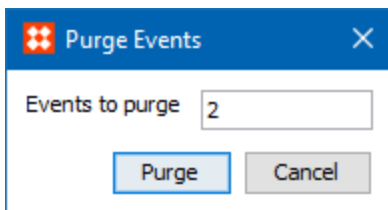
Note: Event IDs are not shown in the UI but are read from the database, making this a troubleshooting option.



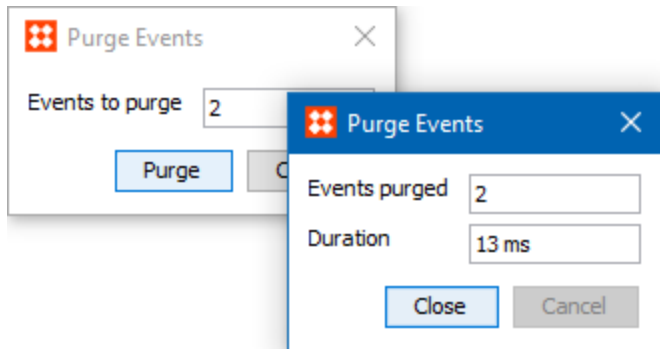
Purge Events

The Purge action is only available when the parameter 'Days to retain events' is set to a number greater than zero during creation or configuration of the OIEP and there are events in the queue and read events are older than the 'Days to retain events' parameter setting. This option is typically used when the unread events are expired, or when the events queued by the OIEP should not be published downstream.

1. In **Event Actions**, click **Purge**.
2. The Purge Events dialog, the number of **read** events that you can delete from the system is displayed.



3. Click **Purge** to display the events purged along with duration to purge those.

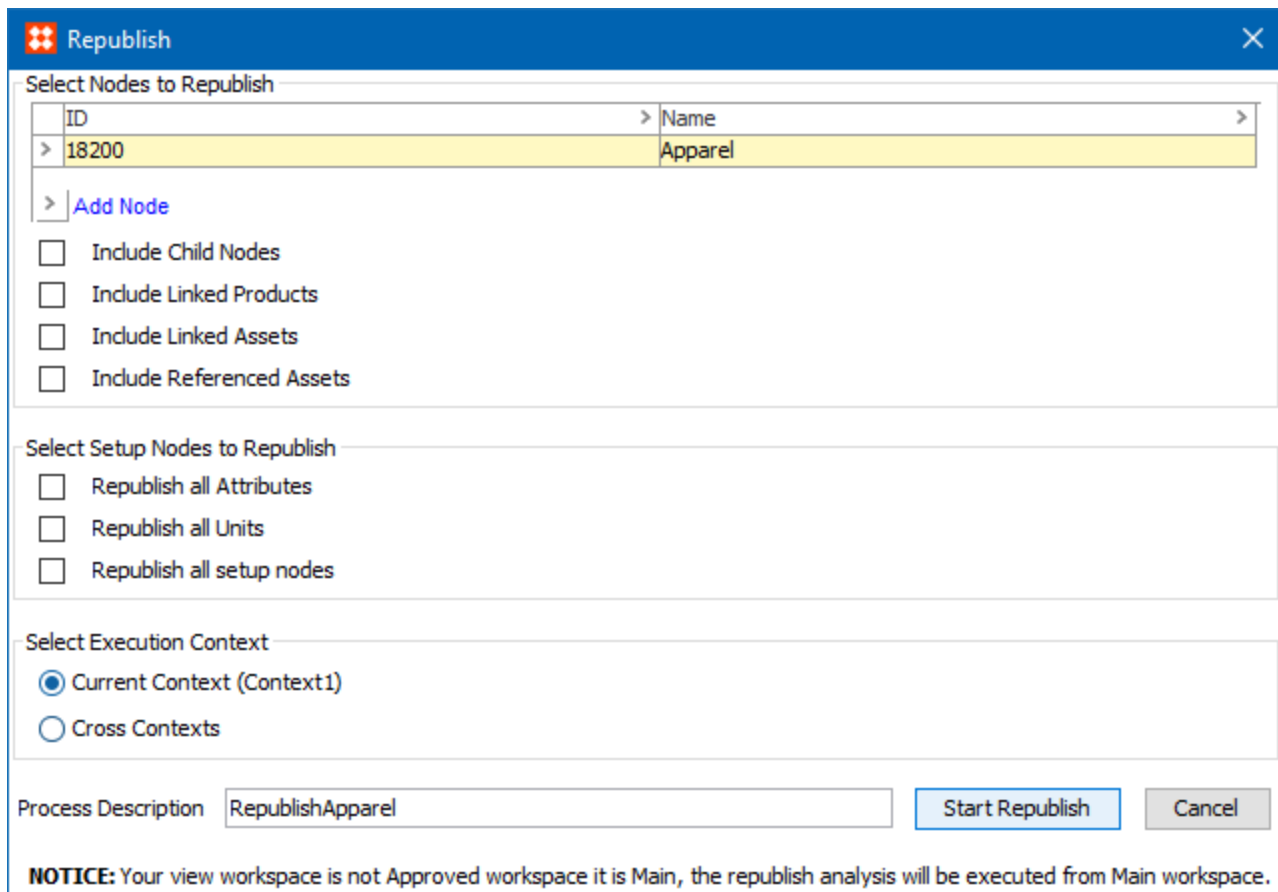


Note: When the number of read events in Purge Events shows 0, forward events to the current date and attempt to purge.

Republish Events

The Republish action generates events on products, classifications, or assets in one or more selected hierarchies. Common setup is to use this option for on-demand creation of republish events. For example, to generate events for all objects in a hierarchy that has never been published by an event queue, or events that were published but the downstream system did not accurately receive the data.

1. In **Event Actions**, click **Republish** to display the Republish dialog.



2. Click **Add node** and then browse or search for the hierarchy you want to republish.
3. Select the objects you want to republish. You have the following options:
 - **Include Child Nodes** republishes all children of the selected object.
 - **Include Linked Products** republishes all products linked to the selected object.
 - **Include Linked Assets** republishes all assets linked to the selected object.
 - **Include Referenced Assets** republishes all assets referenced by the selected object.
4. Select the setup nodes you want to republish. You have the following options:
 - **Republish All Attributes** republishes all attributes linked to the selected object.
 - **Republish all Units** republishes all units linked to the selected object.
 - **Republish all setup nodes** republishes all system setup objects.
5. Select the Execution Context. You have the following options:
 - Select **Current Context** to republish in the current context. (shown in parenthesis)
 - Select **Cross Contexts** to republish child nodes in all contexts.
6. In **Process Description** enter a name for the process and activate the Start Republish button. This is a mandatory field.
7. Click **Start Republish**. Events are triggered on objects that match the selection.

Republish Manually

Additionally, republish events can be generated manually in STEP Workbench as follows:

- From a collection using the Republish option. For more information, see the **Republish** section of the **Maintaining Collections** topic in the **Getting Started / User Guide** documentation.
- Via a business action using the Send Republish Event option, which is available from a business action operation. For more information, see the **Business Action: Send Republish Event** topic in the **Business Rules** documentation.
- Via Bulk Update using the Send Republish Event operation. For more information, see the **Send Republish Event Operation for Bulk Updates** topic in the **Bulk Updates** documentation.

Event-Based OIEP Multithreading Support

To increase performance when exporting from an Event-Based Outbound Integration Endpoint (OIEP) you can increase the number of threads for a given OIEP. Common setup is to use multithreading when there is a large amount of data going to a downstream system on a regular basis, and when the downstream system is capable of handling it.

Consider the following points before increasing the number of threads to more than one:

- The STEP system hardware should have enough resources to perform with multithreading.
- The downstream (receiving) system must be able to handle parallel events.
- Multithreading can negatively impact performance when a large amount of data is involved.
- Run the settings in a test environment, starting with a small amount of data and then increasing it, before implementing it in your production system.

Multithreading Setup

Although the **Number of threads** setting is available on the Configuration tab for both static and event-based OIEPs, the setting **only** affects event-based processes.

Outbound Integration Endpoint	Configuration	Event Triggering Definitions	Backgro
Configuration			
Process Engine	STEP Exporter		
Error reporter	Not Defined		
Schedule	Not scheduled		
Queue for endpoint	OutboundQueue		
Queue for endpoint processes	Out		
Transactional settings	Strict		
Number of threads	3		
Maximum number of old processes	1000		
Maximum age of old processes	1y		
Contexts	French, German		
Workspace	Approved		

The default thread setting is one (1), in which case the endpoint produces a single message at a time, with all events in the batch processed serially. Increasing the thread number results in each batch size being divided by the thread number so that the contents of a batch can be processed in parallel.

Since multiple event batches per message are not supported for multiple threads, when the 'Number of threads' is greater than one (1), then the 'Number of event batches to include per delivery' (shown in the image below) is automatically set to one (1). The following warning is included in the BGP execution report and also in the

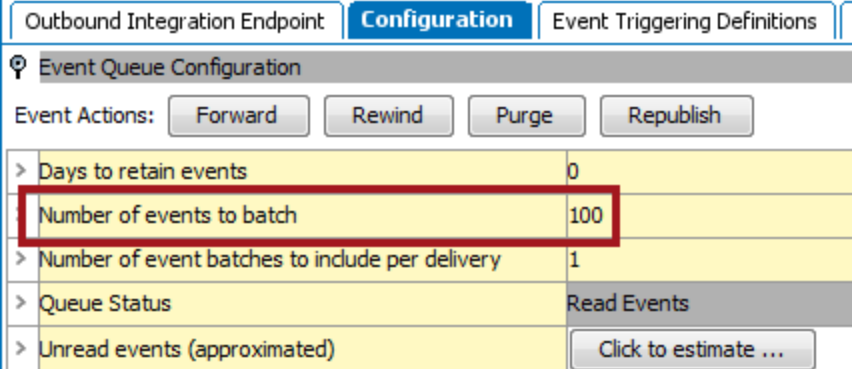
application server log: Number of threads are [x]. Number of batches per delivery will be set to 1 for integration endpoint [ID] as this is the only allowed value when multiple threads are used. Requested number of batches were [y].

When all threads are complete, the batch will still yield a single message, as is consistent with the 'Strict' transactional setting required by event-based OIEPs. Thread size is typically increased for particularly critical information where the speed with which the message is produced is key. However, increasing the setting beyond the capabilities of the hardware will impede the overall performance of the application server, so care must be taken in adjusting the thread size.

Additional Considerations

Increasing the Batch Size

When increasing the number of threads running, the size of the batch will decrease proportionally (e.g., a batch size of 100 using two threads will be split into two batches of 50). When adjusting the number of threads, it is therefore worthwhile to experiment with increasing the batch size, in order for each working thread to have a reasonable amount of data to process.



Event Queue Configuration	
Event Actions: Forward Rewind Purge Republish	
> Days to retain events	0
Number of events to batch	100
> Number of event batches to include per delivery	1
> Queue Status	Read Events
> Unread events (approximated)	Click to estimate ...

Serialized and Parallelized

The batch-fetching of the events is run serially and the data distributed as evenly as possible to each thread. From fetching the batch on through the delivery stage, the pre-processing, main processing, and post-processing take place in parallel. Following post-processing (if applicable), the data are handed over to the delivery stage which is again run serially. Events are therefore delivered as if everything was executed serially. With this in mind, it is important to consider the schedule of the endpoint and ensure that it is set to check for events as frequently as if the endpoint were single-threaded.

Note that when a batch is processed in parallel (via multiple threads), it is recorded in the execution report of the endpoint.

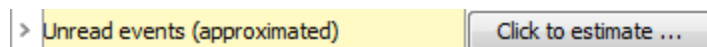
Event-Based OIEP Queued Events

As events are found that meet the criteria set in the event-based OIEP's configuration and event triggering definitions, an estimate and details are available for review in System Setup.

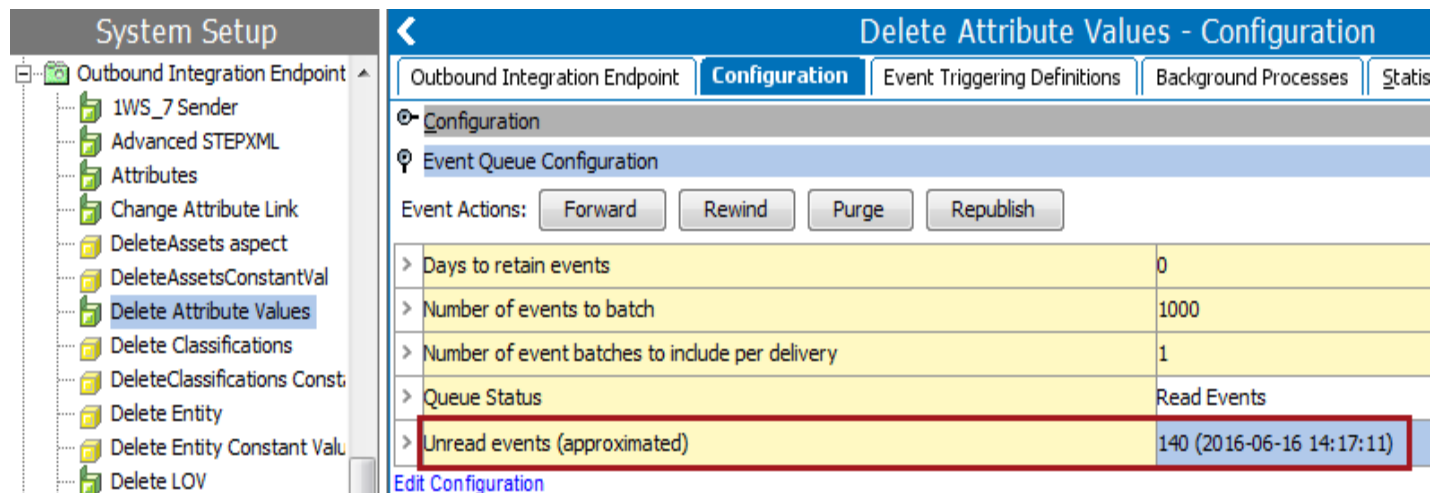
Estimate Unread Events

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.

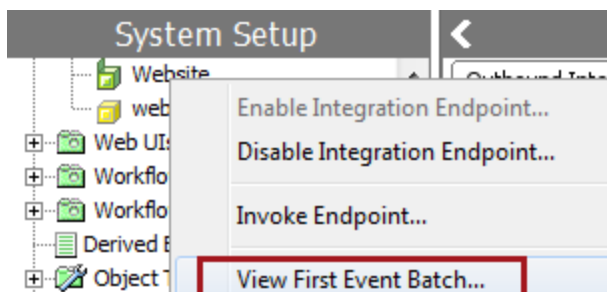


Using the forward, rewind, purge, or republish events options causes the number of unread events on the queue to change. For more information, see **Event-Based OIEP Forward, Rewind, Purge, and Republish**.

View First Event Batch

This option allows you to view more than just a count of the events.

1. Right-click the OIEP and select View First Event Batch.



- 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.

Current Event Batch			×
Time of fetch: 2017-06-08 11:37:46 - Size of batch: 100			
Amount	Origin Type	Event Type	
62	Item	Approve created	
1	Sales Item	Approve created	
5	Item	Approved	
1	Sales Item	Approved	Local Change
2	Outbound Integration Endpoint Type	Delete approved	
1	GDSN Registrations	Revived	
1	GDSN Target Market	Revived	
10	Item	Validation rule modified	
Details		Close	

- For more information about the events, click the **Details** button. The Origin column includes a link to the object being reported by the event.

Current Event Batch						×
Time of fetch: 2017-06-08 11:37:46 - Size of batch: 100						
	Number	Origin	Origin Type	Event Type	Generated	
>	1	20709-012	Sales Item	Approve created	2016-08-30 13:46:43	^
>	2	20695	Item	Approve created	2016-08-30 13:47:31	
>	3	18210 M B	Item	Approved	2016-09-23 14:06:23	
>	4	12-GGK799	Item	Validation rule modified	2016-09-26 16:12:05	
>	5	18210 M B	Item	Approved	2016-09-28 12:30:07	
>	6	12-GGK7 12-GGK799	Item	Approved	2016-10-28 09:15:42	
>	7	Cosmic P ID = 100703	Item	Approve created	2017-01-02 08:01:42	
>	8	Pink & Green Party Hat(2)(2)	Item	Approve created	2017-01-02 08:01:42	

- To postpone an event until the end of the queue, select a row and right-click on the > column, and click Postpone Event.

Important: Events can rely on data included in previous messages. Reordering the sequence may cause issues in the receiving system and should be run in a test environment first when possible. Carefully consider the warning displayed after clicking the Postpone Event option allows you to cancel the action.

Current Event Batch

Time of fetch: 2017-06-08 11:37:46 - Size of batch: 100

Number	Origin	Origin Type	Event Type	Generated
> 1	20709-012	Sales Item	Approve created	2016-08-30 13:46:43
> 2	20695	Item	Approve created	2016-08-30 13:47:31
> 3	18210 M B	Item	Approved	2016-09-23 14:06:23
> 4	12-GGK799	Item	Validation rule modified	2016-09-26 16:12:05
> 5	12-GGK799	Item	Approved	2016-09-28 12:30:07
> 6	12-GGK799	Item	Approved	2016-10-28 09:15:42
> 7	Cosmic Party Hat	Item	Approve created	2017-01-02 08:01:42
> 8	Pink & Purple	Item	Approve created	2017-01-02 08:01:42
> 9	Pink & Purple	Item	Approve created	2017-01-02 08:01:42
> 10	Politics Party Hat(2)(2)	Item	Approve created	2017-01-02 08:01:42
> 11	Purple	Item	Approve created	2017-01-02 08:01:42
> 12	Yellow	Item	Approve created	2017-01-02 08:01:42
> 13	Pink & Purple	Item	Approve created	2017-01-02 08:01:42
> 14	Purple	Item	Approve created	2017-01-02 08:01:42
> 15	Pink & Purple	Item	Approve created	2017-01-02 08:01:42
> 16	Politics Party Hat(2)(2)	Item	Approve created	2017-01-02 08:01:42
> 17	Purple & White Party Hat(2)(2)	Item	Approve created	2017-01-02 08:01:42
> 18	Yellow & Pink Party Hat(2)	Item	Approve created	2017-01-02 08:01:42

Overview

Close

Postpone Event

Postpone Event

!

Postponing events may seriously affect down-stream systems behaviour. Please ensure you understand the consequences before proceeding.

Postpone

Cancel

5. Click the **Overview** button to return to the previous view.

Event-Based OIEP Status and Queue Status

For event-based integration endpoint, the OIEP status (enabled / disabled) setting and the queue status (read events / discard events) setting are independent of each other, but work together to determine how data is processed.

This means that, for example, if a downstream system is offline but receives deliveries from an OIEP, disabling the endpoint stops publishing data, but allows the OIEP to continue registering events in STEP so that data can be published once the external system is back online.

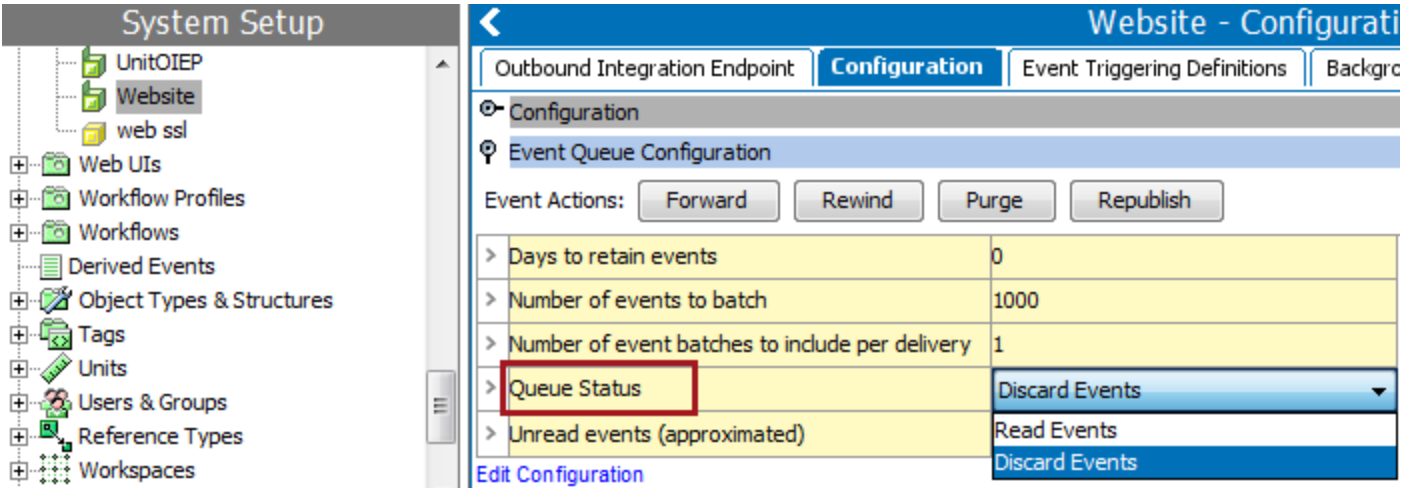
Common setup includes the following combination of queue status and endpoint status settings:

- **Enabled + Read Events = Active.** Use this setting for an active endpoint that should deliver data to downstream systems.
- **Disabled + Read Events = Paused.** Use this setting to temporarily disable the feed, while retaining access to events being generated, even while the endpoint is disabled.
- **Disabled + Discard Events = Inactive.** Use this setting when no new events should be processed (now or later) and data should not be delivered downstream.
- **Enabled + Discard Events = Transition.** Not commonly used, but can be employed when one endpoint will take over for another, or prior to running a bulk update process that should not be sent downstream. Allows the old endpoint to process queued events, but not generate any new ones as new events should be set to queue on the new endpoint (or discarded if bulk update should not be sent out).

Note: By default when a new OIEP is created and configured, the OIEP status is set to Disabled and Queue Status is set to Discard events.

Set the Queue Status

1. In System Setup, locate an event-based outbound integration endpoint and view the **Configuration** tab.
2. Open the **Event Queue Configuration** flipper.
3. In **Queue Status**, select the desired option.
 - **Read Events** allows the OIEP to register events as they occur, based on the OIEP configuration and event triggering definitions.
 - **Discard Events** ignores generated events that meet the OIEP configuration and event triggering definitions.



For information about setting the enabled / disabled parameter, see the **Running an Outbound Integration Endpoint**.