

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 8.1

Table of Contents

Table of Contents	2
Data Exchange	13
Inbound Tools	13
Data Format	13
Receiver Method	13
Outbound Tools	13
Data Format	13
Delivery Method	13
Gateway Tool	14
Data Formats	14
Advanced STEPXML Format	16
Format Availability	16
Mapping	16
Inbound Data	17
Outbound Data	18
Alphabetical Index - XML Format	20
Format Availability	20
Mapping	20
Outbound Data	20
Ariba CIF Format	21
Format Availability	21
Mapping	21
Outbound Data	21
Ariba CIF 3.0 Format	23
Format Availability	23
Mapping	23

Outbound Data	23
BMEcat Format	25
Format Availability	25
Mapping	26
Inbound Data	26
Outbound Data	27
BMEcat 2005 Format	29
Format Availability	29
Mapping	29
Inbound Data	29
Outbound Data	31
CSV Format	32
Format Availability	33
Mapping	33
Inbound Data	33
Outbound Data	35
cXML Format	37
Format Availability	37
Mapping	37
Outbound Data	37
eCl@ss Format	39
Format Availability	40
Mapping	40
Multiple Versions	40
Inbound Data	41
eCl@ss Classification Import	41
Map eCl@ss Attributes to STEP Attributes	43

ETIM Format	45
Format Availability	45
Mapping	45
ETIM Configuration	45
ETIM Versions	46
ETIM Classification	46
ETIM Feature Types	48
Valid Values are Determined by ETIM Version	49
Valid Units are Determined by Classification to Attribute Link	49
Handling Import Errors	50
ETIM and ETIM v2 Format	51
ETIM 6 Format	53
Create Product to ETIM Classification Link	54
Set Validity on ETIM Attributes for Products	57
Manually Import ETIM Attribute Values	62
Excel Format	74
Format Availability	75
Mapping	75
Inbound Data	75
Outbound Data	77
Excel Smartsheet Format	78
Setup Requirements	79
Initiating Products Using a Smartsheet	79
Maintaining Products Using a Smartsheet	83
Smartsheet Import Configurations	86

Smartsheet Data and Template Configurations	88
Additional Information About Mapping	95
Using a Smartsheet	97
Additional Information About Smartsheets	101
FixedWidth Format	106
Format Availability	106
Mapping	106
Inbound Data	106
FixedWidth Format Configuration Examples	108
Flatplan Excel Format	114
Format Availability	114
Mapping	114
Inbound Data	114
Outbound Data	115
Generic XML Format	116
Limitations	116
Format Availability	116
Mapping	117
Inbound Data	117
Outbound Data	118
Generic XML Inbound Processing Instructions	120
Remove Redundant Elements	132
Add Record and Source Instructions	133
Add Repeated Instruction	134
Add Repeated Instruction With a Filter	135

Add MultiSource Instruction	136	Limitations	219
Generic XML Outbound Processing Instructions	138	Setup Requirements	219
Object Type and Event Type Selection	162	Inbound Data	219
Template	163	Global Product Classification Import	221
Mapping	164	Create a GPC Product to Classification Link Type	223
Results	164	IDoc MATMAS 05 Format	226
Object Type and Event Type Selection	165	Format Availability	226
Template	165	Mapping	226
Mapping	166	Inbound Data	226
Result	166	Outbound Data	228
Object Type and Event Type Selection	167	Mapping	230
Template	168	Inbound Data Mapping	230
Mapping	169	Outbound Data Mapping	230
Results	169	Inbound Map Data Options	231
Product	170	Map to an Existing Attribute	247
Object Type and Event Type Selection	170	Create and Map to a New Attribute	248
Template	171	Mapping ID	261
Mapping	171	Mapping Key	264
Results	172	Create Transformation Lookup Table Objects	289
Data Path Configuration	197	Outbound Map Data Options	291
Object Type and Event Type Selection	213	Transforming Data	309
Template	214	Outbound Add Transformation Options	310
Mapping	215	Data Container Separator - Transform Outbound	313
Results	215	Multi-Value Separator - Transform Outbound	314
Global Product Classification Format	218	Aspect - Transform Outbound	315
Format Availability	219		
Mapping	219		

Mapping Transformation Examples	317
Publication Excel Format	367
Format Availability	367
Mapping	368
Inbound Data	368
Outbound Data	369
STEPXML Format	370
Limitations	370
Schema	371
Format Availability	371
Mapping	371
Inbound Data	371
Outbound Data	372
STEPXML Outbound Parameters	374
STEPXML Tags and Examples	384
Name	390
Values	390
ClassificationReference	390
MetaData	390
Results	406
Name	419
Values	419
AssetCrossReference	419
ProductCrossReference	419
ClassificationReference	419
Nested structure	420
Non-nested structure	420

UNSPSC Format	433
Format Availability	433
Mapping	433
Inbound Data	433
xCBL Format	437
Format Availability	437
Mapping	437
Outbound Data	437
Export Manager	439
Setup Requirements	439
Additional Information	440
Creating a Data Export	440
Considerations for Exporting Products	443
Export Manager - Select Configuration	444
Export Manager - Select Objects	445
Export Manager - Select Format	447
Export Manager - Map Data	448
Export Manager - Advanced	449
Export Manager - Select Delivery Method	450
Running a Data Export	457
Maintaining a Saved Export Configuration	458
View a Saved Export Configuration	458
Edit a Saved Export Configuration	458
Monitoring a Data Export	459
Monitoring the Export Process	459
Execution Report	460
Additional Information About Export	461

Manager	Setup Requirements	484
Scheduling a Data Export	Additional Information	485
Scheduled Process Properties	Creating a Data Import	485
Deleting a Scheduled Data Export	Import Manager - Select Configuration ..	487
Gateway Integration Endpoints	Import Manager - Select Data Source ..	489
Setup Requirements	Import Manager - Select Format	492
Additional Information	Import Manager - Map Data	493
Initial Setup for a Gateway Integration	Import Manager - Identify Objects	494
Endpoint	Import Manager - Identify Destination ...	498
Create the Gateway Setup Group	Import Manager - Select Business Rules	499
Link the Gateway Object Type to Setup	Import Manager - Advanced Settings ...	502
Group	Running a Data Import	508
Create an Instance of the Gateway	Maintaining a Saved Import Configuration	510
Endpoint Object	View a Saved Import Configuration	510
Creating a Gateway Integration Endpoint .	Edit a Saved Import Configuration	511
Gateway Integration Endpoint	Monitoring a Data Import	511
Configuring a Gateway Integration	Monitoring the Import Process	512
Endpoint	Properties Flipper	512
Business Action Accesses Gateway	Execution Report	513
Integration Endpoint	Reimporting a Corrected File	514
Gateway Integration Endpoint Bind	Import Error Messages	515
Running a Gateway Integration Endpoint .	Import Error Message Examples	518
Gateway Integration Endpoint	Inbound Integration Endpoints	527
Maintaining a Gateway Integration	Setup Requirements	527
Endpoint	Additional Information	527
Monitoring a Gateway Integration	Initial Setup for an Inbound Integration	
Endpoint	Endpoint	528
Monitor from STEP	Create the Inbound Integration Setup	529
Monitor Outside of STEP		
Gateway Integration Endpoint Statistics		
Exporting Gateway Integration Endpoint		
Definitions as Comments		
Import Manager		

Group	
Link the IIEP Object Type to Setup Group	530
Create an Instance of the IIEP Object ...	531
Creating an Inbound Integration Endpoint	532
IIEP - Identify Endpoint	534
IIEP - Choose Receiver	535
Configuration	537
Parameters	537
Configuration	538
Parameters	538
Configuration	539
Parameters	539
Configuration	541
Parameters	541
Configuration	542
Parameters	542
Configuration Examples	543
FileInitialContextFactory JMS Receiver ...	543
JMS Websphere Delivery Using SSL	543
Apache ActiveMQ JMS Receiver	544
Configuration	544
Parameters	544
Configuration	545
Parameters	545
IIEP - Configure Endpoint	547
IIEP - Configure PreProcessor	549
IIEP - Configure Processing Engine	549

IIEP - Configure PostProcessor	557
IIEP - Schedule Endpoint	557
IIEP - Configure Error Reporter	558
Running an Inbound Integration Endpoint	559
Enable and Invoke an Inbound Integration Endpoint	559
Disable an Inbound Integration Endpoint	562
Maintaining an Inbound Integration Endpoint	562
Inbound Integration Endpoint Tab	563
Background Processes Tab	566
Statistics Tab	567
Error Log Excerpts Tab	567
Log Tab	568
Status Tab	568
Monitoring an IIEP via Background Process	568
Selecting the Background Process Queue	569
Background Process Queue Size Property	569
Monitoring an IIEP via External System ...	570
Monitoring Sensors	570
REST API Monitoring	570
Import Error Messages	571
Import Error Message Examples	574
Illegal Unit for Attribute	574
Unknown Parent Product	575
Maximum Length, Illegal Value	575
Out of Range, Illegal Value	576

Not a Number, Illegal Value	576	OIEPs	
Object Existed with Another Object Type.....	578	Initial Setup for an Outbound Integration Endpoint	593
Unknown Classification	578	Create the Outbound Integration Setup Group	593
Not Privileged to Create Object	578	Link the OIEP Object Type to Setup Group	595
Unknown Asset ID	579	Create an Instance of the OIEP Object ..	596
Array Operation Failed	579	Creating an Outbound Integration Endpoint	597
Optimistic Locking Errors	579	Creating an Event-Based Outbound Integration Endpoint	598
The Transaction has been Rolled Back: Unique Constraint	579	Process Engine	606
Additional Information About Inbound Integration Endpoints	584	Error Reporter	606
Exporting Inbound Integration Endpoint Definitions as Comments	584	Schedule	606
Handling Failed IIEP Background Processes	584	Queue for Endpoint	607
Inbound Integration Endpoint Structure ..	586	Queue for Endpoint Processes	608
Inbound Integration Endpoint Transactional Settings	587	Transactional Settings	608
Reference Target Lock Policy on Object Types	589	Number of Threads	608
Outbound Integration Endpoints	590	Maximum Number of Old Processes	608
Event-Based Outbound Integration Endpoint	590	Maximum Age of Old Processes	608
Setup Requirements for Event-Based OIEPs	590	Contexts	608
Additional Information for Event-Based OIEPs	591	Workspace	608
Select Objects Outbound Integration Endpoint	592	Configure the Object Types and Event Types	609
Setup Requirements for Select Objects OIEPs	592	Using Multiple OIEPS or Multiple Output Templates	610
Additional Information for Select Objects	592	Multiple OIEPs	610
		Multiple Output Templates	611
		Limitations of Multiple Output Templates ..	611

Configure the Format	611
Configure the Pre-processor and Post-processor	614
Pre-processor	614
Post-processor	615
OIEP - Event-Based - Pre-Processor Configuration Example	617
OIEP Post-Processor - Copy Context Dependent Values and References	620
Copy to Directory	622
Deploy	622
Email	622
FTP	623
JMS	623
Mongo	623
Oracle AQ	624
REST	624
REST Direct	625
SFTP	625
OIEP Delivery Methods	627
Copy to Directory Delivery Method	629
Deploy Delivery Method	631
Email Delivery Method	632
File Delivery Method	633
FTP Delivery Method	634
IBM Websphere MQ SSL Delivery Method	636
JMS Delivery Method	638
Sharedconfig.properties File Examples	639
FileInitialContextFactory JMS Delivery	639

Configuration Example	
JMS Websphere Delivery Using SSL Configuration Example	640
Apache Active MQ JMS Delivery	640
Mongo Delivery Method	641
Mongo Delivery Method Elements	643
Default Databases and Collection	645
STEP JSON and Mongo JSON	646
JavaScript Triggers	646
Bound Variables	647
Example JavaScript to Maintain a Collection in a Classification	647
Prerequisites for Configuring the MongoDB Adapter	649
Configure the OIEP	649
Configuring the JavaScript Triggers	650
MongoDB Adapter Setup Quick Guides	651
Configuring Mongo Authentication Quick Guide	651
SSL Configuration Quick Guide	652
Mongo Delivery Method Conversion Example	654
Values	654
References and Links	655
Attribute Types	657
Reference Types	658
Asset Push Locations	658
Tables	659
Table Formatting	659
Attribute Links	661

Unit	661
List Of Values	661
Oracle AQ Delivery Method	663
REST Delivery Method	664
REST Direct Delivery Method	665
Server Side Delivery Method	667
SFTP Delivery Method	668
Websphere Commerce Import Delivery Method	670
Triggering Object Types Flipper	671
Object Types	672
Event Filter	672
Generate Event	673
Triggering Attributes Flipper	674
Triggering Table Types Flipper	675
Reference Type Triggers Flipper	675
Miscellaneous Triggers Flipper	676
Creating a Select Objects Outbound Integration Endpoint	677
Process Engine	683
Error Reporter	684
Schedule	684
Queue for Endpoint	685
Queue for Endpoint Processes	685
Transactional Settings	686
Number of Threads	686
Maximum Number of Old Processes	686
Maximum Age of Old Processes	686

Contexts	686
Workspace	686
Configure the Object Types	688
Using Multiple OIEPS or Multiple Output Templates	690
Limitations of Multiple Output Templates ..	690
Configure the Format	690
Configure the Pre-processor and Post-processor	693
Pre-processor	693
Post-processor	693
OIEP Post-Processor - Copy Context Dependent Values and References	695
Copy to Directory	697
Deploy	697
Email	697
FTP	698
JMS	698
Mongo	698
Oracle AQ	699
REST	699
REST Direct	700
SFTP	700
OIEP Delivery Methods	702
Copy to Directory Delivery Method	704
Deploy Delivery Method	706
Email Delivery Method	707
File Delivery Method	708
FTP Delivery Method	709

IBM Websphere MQ SSL Delivery Method	711
JMS Delivery Method	713
Sharedconfig.properties File Examples	714
FileInitialContextFactory JMS Delivery Configuration Example	714
JMS Websphere Delivery Using SSL Configuration Example	715
Apache Active MQ JMS Delivery	715
Mongo Delivery Method	716
Mongo Delivery Method Elements	718
Default Databases and Collection	720
STEP JSON and Mongo JSON	721
JavaScript Triggers	721
Bound Variables	722
Example JavaScript to Maintain a Collection in a Classification	722
Prerequisites for Configuring the MongoDB Adapter	724
Configure the OIEP	724
Configuring the JavaScript Triggers	725
MongoDB Adapter Setup Quick Guides	726
Configuring Mongo Authentication Quick Guide	726
SSL Configuration Quick Guide	727
Mongo Delivery Method Conversion Example	729
Values	729
References and Links	730
Attribute Types	732
Reference Types	733

Asset Push Locations	733
Tables	734
Table Formatting	734
Attribute Links	736
Unit	736
List Of Values	736
Oracle AQ Delivery Method	738
REST Delivery Method	739
REST Direct Delivery Method	740
Server Side Delivery Method	742
SFTP Delivery Method	743
Websphere Commerce Import Delivery Method	745
Running an Outbound Integration Endpoint	745
Enable and Invoke an Outbound Integration Endpoint	746
Disable an Outbound Integration Endpoint	748
Maintaining an Outbound Integration Endpoint	749
Outbound Integration Endpoint Tab	749
Configuration Tab	750
Background Processes Tab	754
Statistics Tab	755
Error Log Excerpts Tab	755
Log Tab	756
Status Tab	756
Monitoring an OIEP via Background Process	756

Selecting the Background Process Queue	757
Background Process Queue Size Property	758
Monitoring an OIEP via External Systems	758
Monitoring Sensors	758
REST API Monitoring	758
Additional Information About All Outbound Integration Endpoints	759
Exporting Outbound Integration Endpoint Definitions as Comments	759
Handling Failed OIEP Background Processes	760
Outbound Integration Endpoint Structure	761
Additional Information About Event-Based OIEPs	762
Event-Based OIEP Event Batching	762
Event-Based OIEP Examples	763
Event-Based OIEP Forward, Rewind, Purge, and Republish	777
Event-Based OIEP Multithreading Support	782
Event-Based OIEP Queued Events	784
Event-Based OIEP Status and Queue Status	786

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 **Export Manager - 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 determines 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 delivery methods available are determined by the outbound tool used (see **OIEP Delivery Methods**).

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.	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.	Yes*	No

Format	Description	Inbound	Outbound
	*This import is only available via the eClass Classification Import wizard.		
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 6	XML Electro-Technical Information Model (ETIM) classification structure that includes multiple languages. See ETIM 6 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 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*
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 that must be created manually.

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**.

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		

<

>

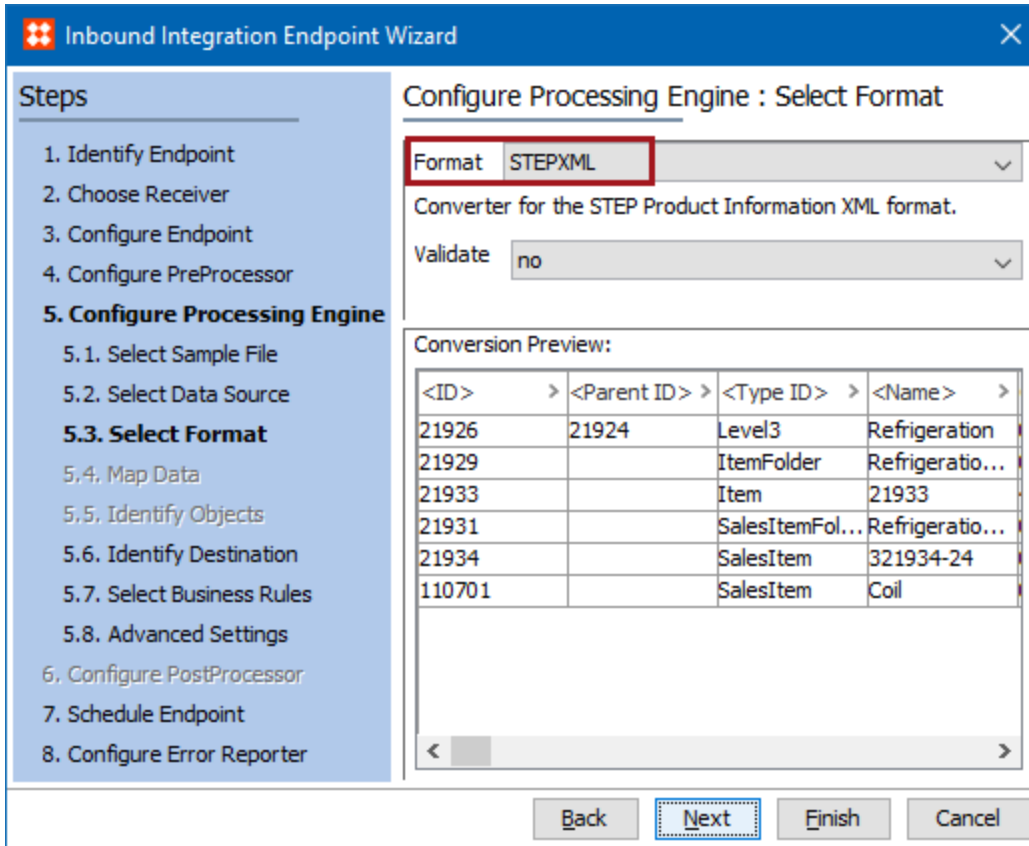
Back

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

Format: **STEPXML**

Converter for the STEP Product Information XML format.

Validate: no

Conversion Preview:

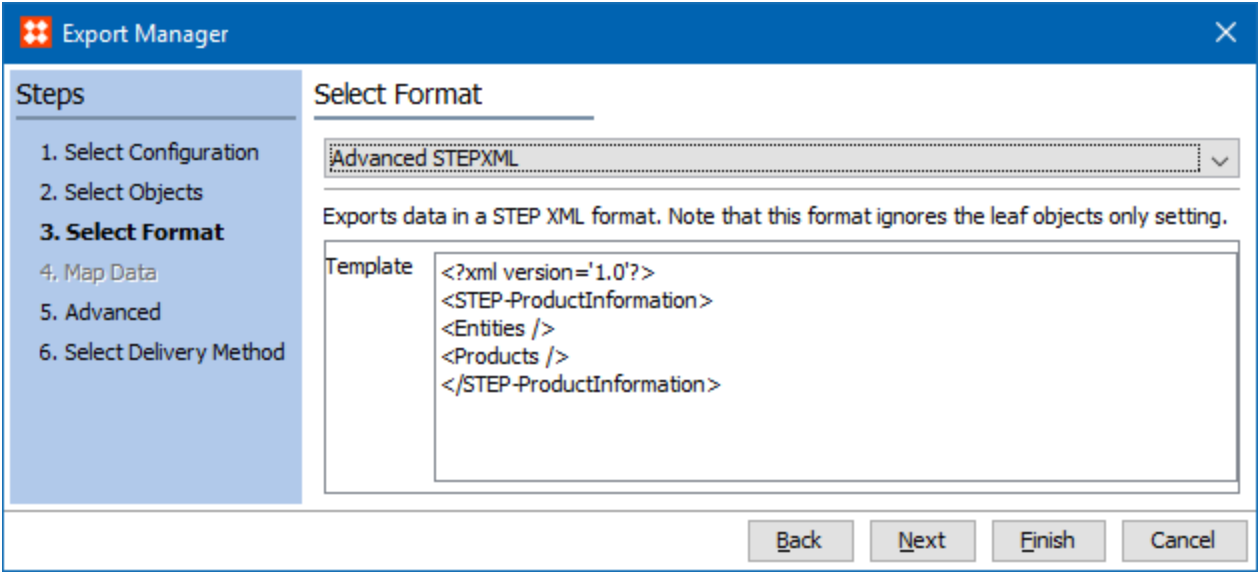
<ID>	<Parent ID>	<Type ID>	<Name>
21926	21924	Level3	Refrigeration
21929		ItemFolder	Refrigeratio...
21933		Item	21933
21931		SalesItemFol...	Refrigeratio...
21934		SalesItem	321934-24
110701		SalesItem	Coil

Buttons: Back, Next, Finish, Cancel

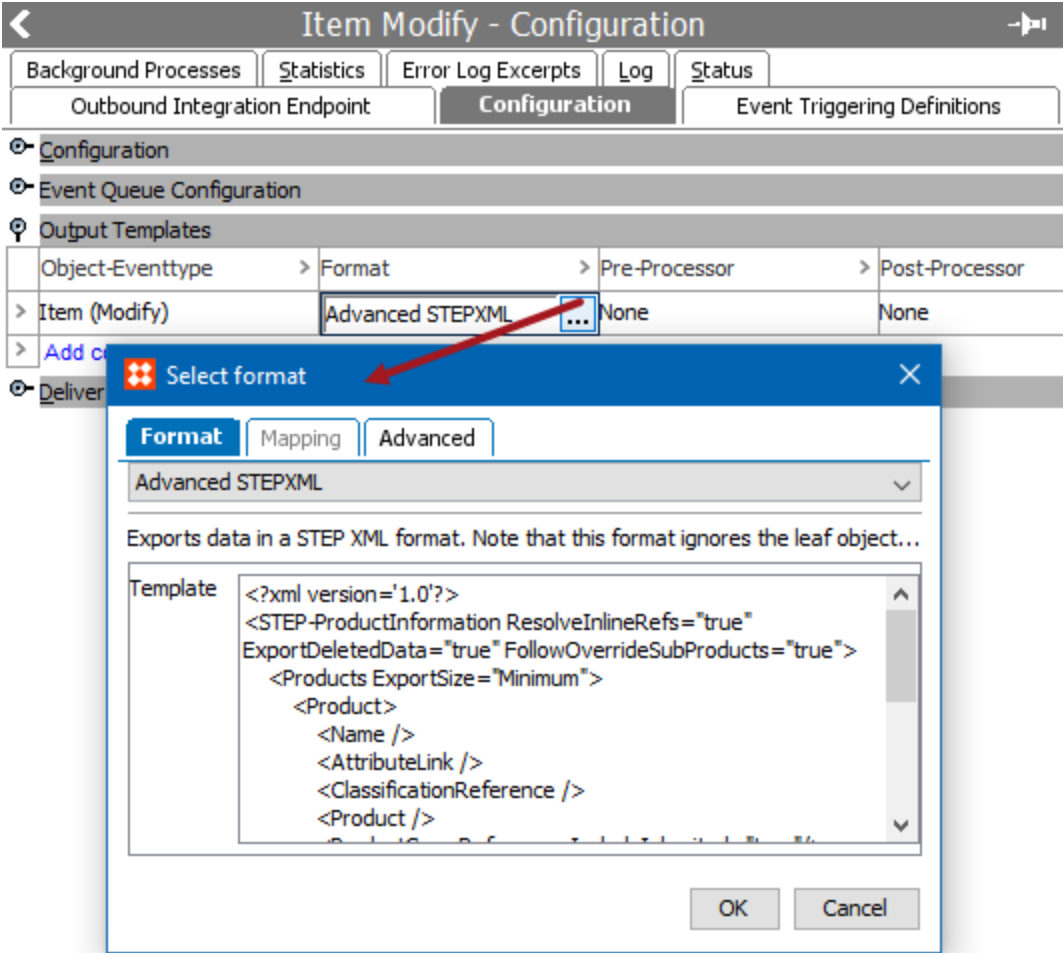
Outbound Data

The default template loaded for export will output all entities and products. Update it with your own template to configure the data and layout of the export. Available tags and examples using STEPXML are available in the **STEPXML Tags and Examples** documentation. Ultimately, the template must be based on the STEPXML XSD. See the **STEP API Documentation** 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. For more information, see **Creating Document Indexes** 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**

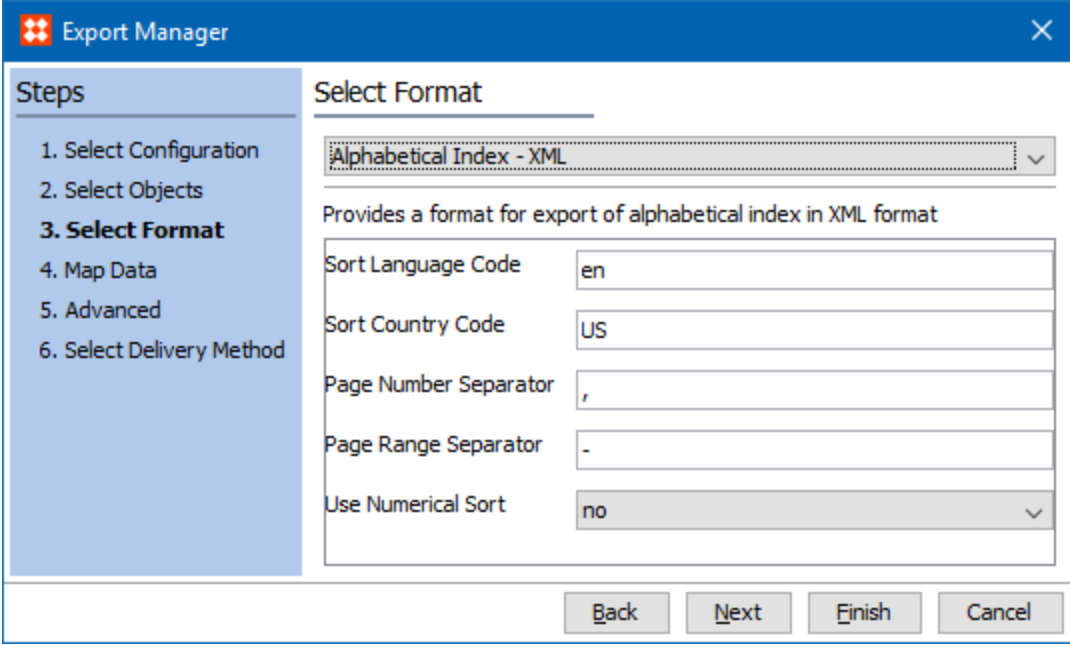
Mapping

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

Outbound Data

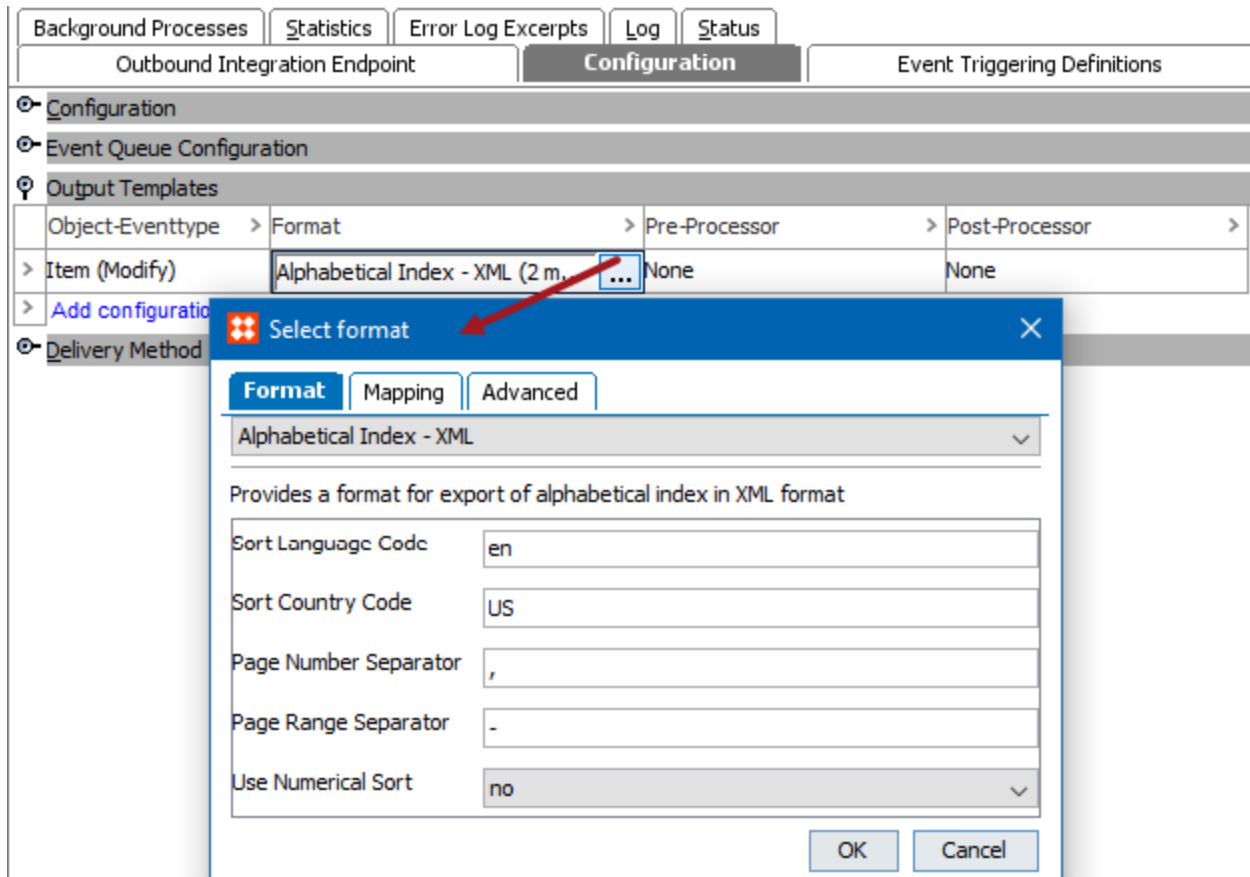
The same parameters are available in both Export Manager and OIEP.

Export Manager



The screenshot shows the 'Export Manager' dialog box with a blue title 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



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 **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

Ariba CIF

▼

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

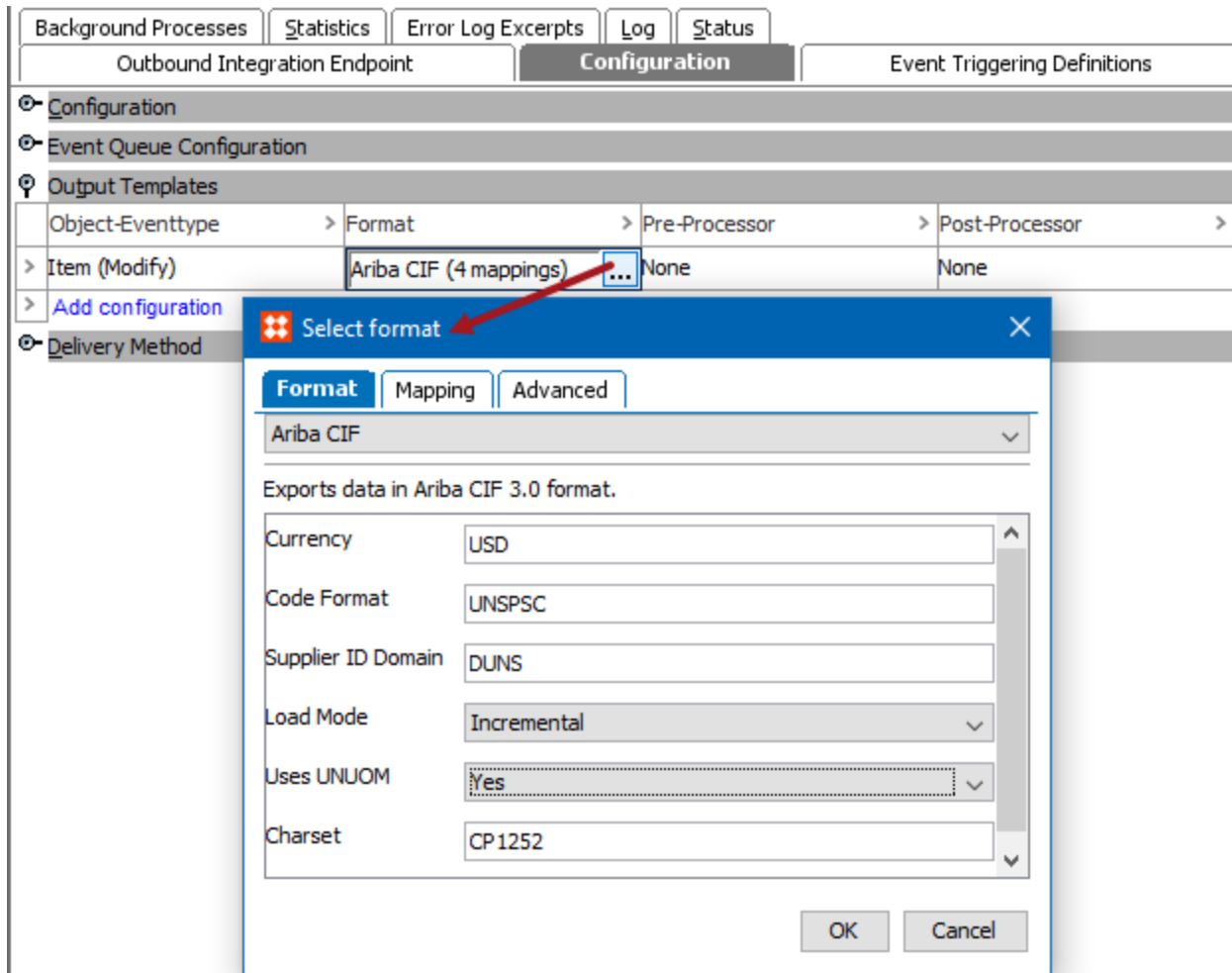
Back

Next

Finish

Cancel

OIEP



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 **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

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

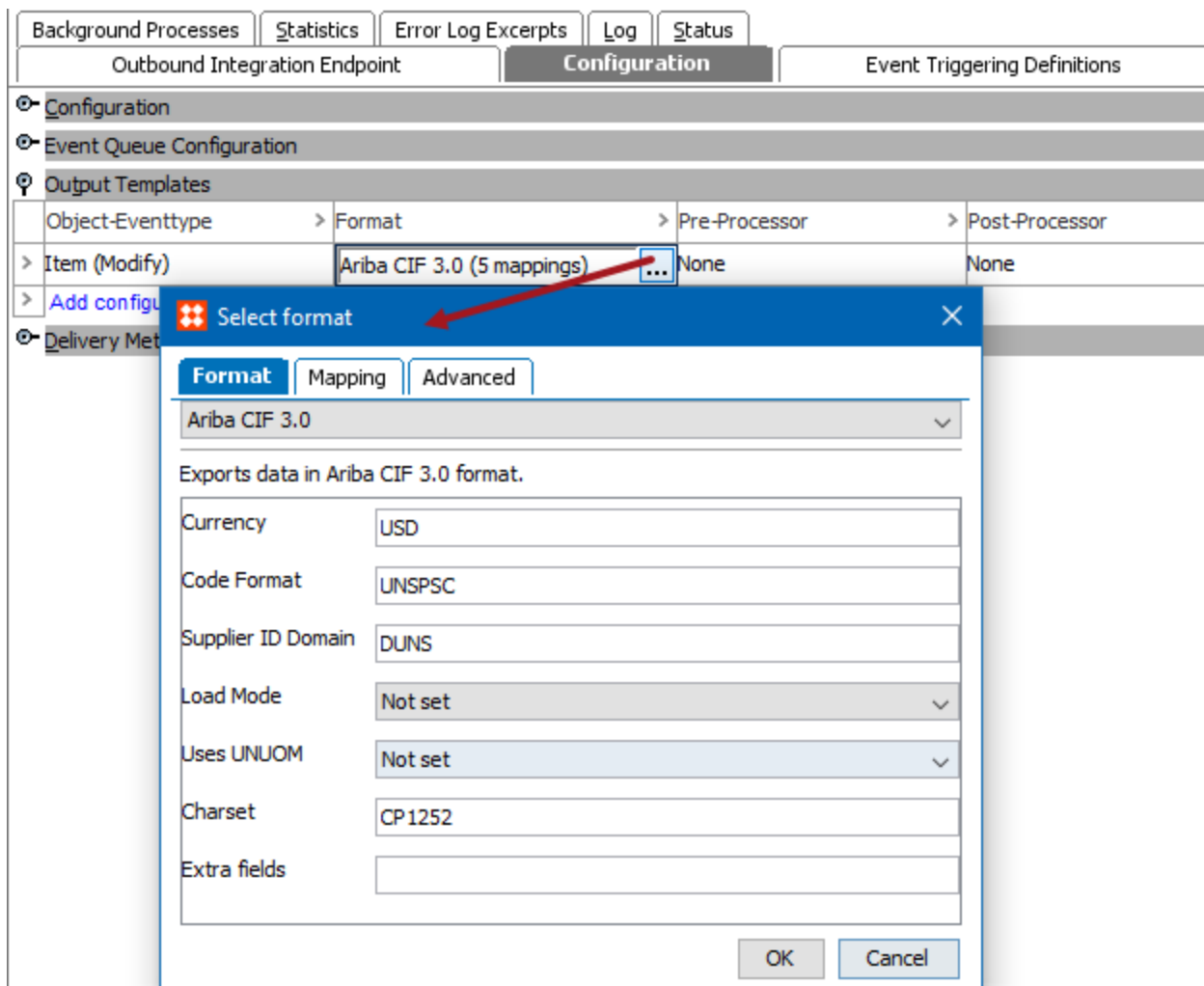
Back

Next

Finish

Cancel

OIEP



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 **Mapping**.

Inbound Data

Inbound Parameters

- **Validate** determines if the STEP checks the validity of the BMEcat file with 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

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		▼

< >

Back

Next

Finish

Cancel

Outbound Data

Exporting the BMEcat format involves supplying data in the parameters displayed in the wizard. Some are required, as is indicated on the Map Data step.

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

Price update only?

No

Update count

0

Generator Information

Generated by STEP BMEcat2 plugin.

Language

eng

Back

Next

Finish

Cancel

OIEP

Item Modify Non STEPXML - Configuration

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)	BMEcat (2 mappings)	None	None
> Add			

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

Price update only?

No

Update count

0

<

>

OK

Cancel

© Stibo Systems - Confidential - STEP Trailblazer 8.1

28

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 **Mapping**.

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

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 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>`

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

Back

Next

Finish

Cancel

Outbound Data

The sample file is loaded automatically when selecting BME2005 format for export.

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 2005

▼

Converts to an BMEcat2005 format based on a sample.

Sample

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

Allow empty tags

No

▼

Language

Catalog ID

Catalog version

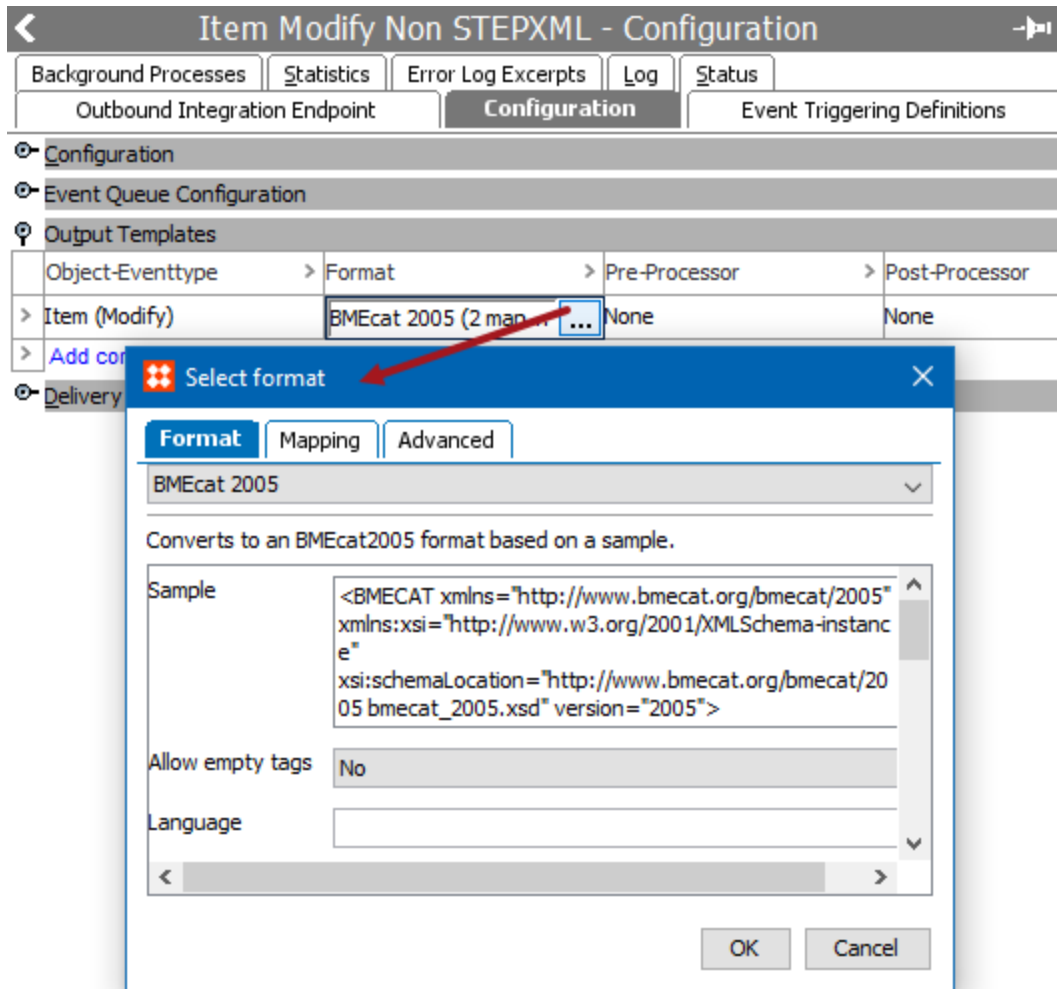
Back

Next

Finish

Cancel

OIEP



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 **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.

The following character sets are supported by the Import Manager for CSV files: UTF-8, UTF-16, Windows-1252, or ISO 8859-1 (Latin-1).

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.

- **Allow Multi Line Values** - select 'yes' if the file includes 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, and the Allow Multi Line Values must be set to yes.
- **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 recommended setting is **UTF-8**, unless you have a reason to do otherwise.

Note: If the file contains special symbols, like trademark symbols, Unicode character set UTF-8 or UTF-16 is required to import correctly.

- **Conversion Preview** - a sample of the first few lines of the file is displayed to allow verification that the selected options are correct.
- **Delimiter** - from the dropdown, select the character to use as delimiter: semi-colon (;), comma (,), the tab character, the pipe symbol (|), or colon (:). The recommended setting is **tab** to avoid splitting the data on printable characters that may occur within the values.

Note: Only the delimiters in the dropdown are allowed. If the file uses a different delimiter, before importing, first convert the file outside of STEP.

- **Has Header** - select Yes if the file has a header line. If 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. 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 leading and trailing spaces in values.

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

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

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.

Outbound Parameters

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

- **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 recommended setting is **UTF-8**, unless you have a reason to do otherwise. This Character Set text is also under the Inbound Data Exchange Parameters section above.

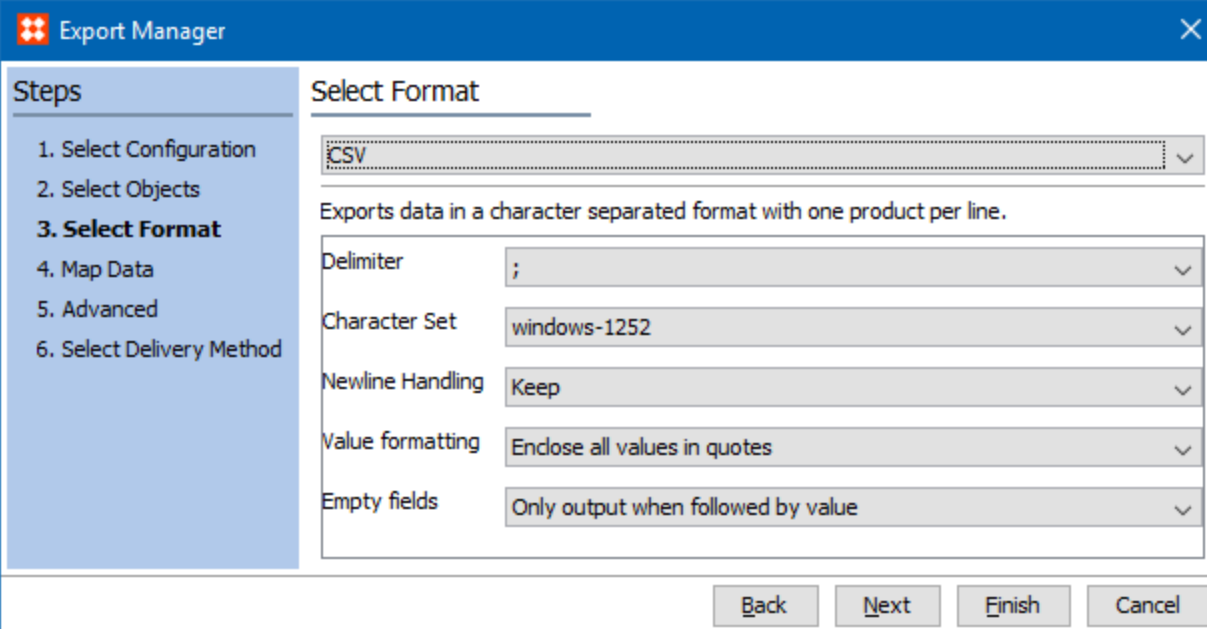
Note: If the file contains special symbols, like trademark symbols, Unicode character set UTF-8 or UTF-16 is required to import correctly.

- **Delimiter** - from the dropdown, select the character to use as delimiter: semi-colon (;), comma (,), the tab character, the pipe symbol (|), or colon (:). The recommended setting is **tab** to avoid splitting the data on printable characters that may occur within the values.

Note: Only the delimiters in the dropdown are allowed. If the file uses a different delimiter, before importing, first convert the file outside of STEP.

- **Empty fields** - select 'Only output when followed by value' or 'Always output'.
- **Newline Handling** - select to Keep the new line or Convert to Space. The recommended setting is **Convert to Space**. If new lines (return codes) exist in the data being exported, selecting Keep can cause issues with data integrity, since character-separated files do not expect new lines within a record. However, if the file recipient can handle these new lines, Keep allows them to be included.
- **Value formatting** - select either 'Enclose all values in quotes' or 'Export without quotes unless value contains delimiter character(s)'.

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

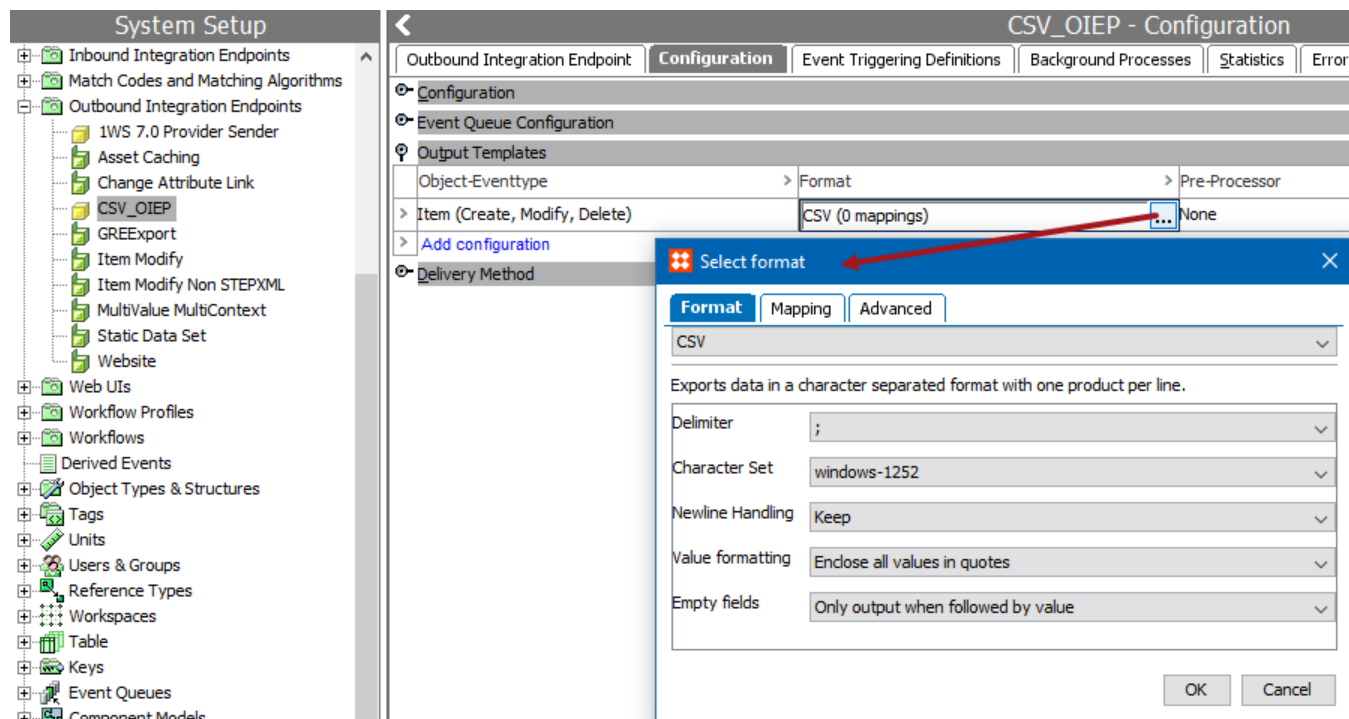
CSV

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



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 **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

icXML

▼

Exports data in the Ariba cXML 1.2 format.

Load Mode

Full

▼

Supplier ID

Supplier ID Domain

Comments

Language

en

Item Mode

IndexItemAdd

▼

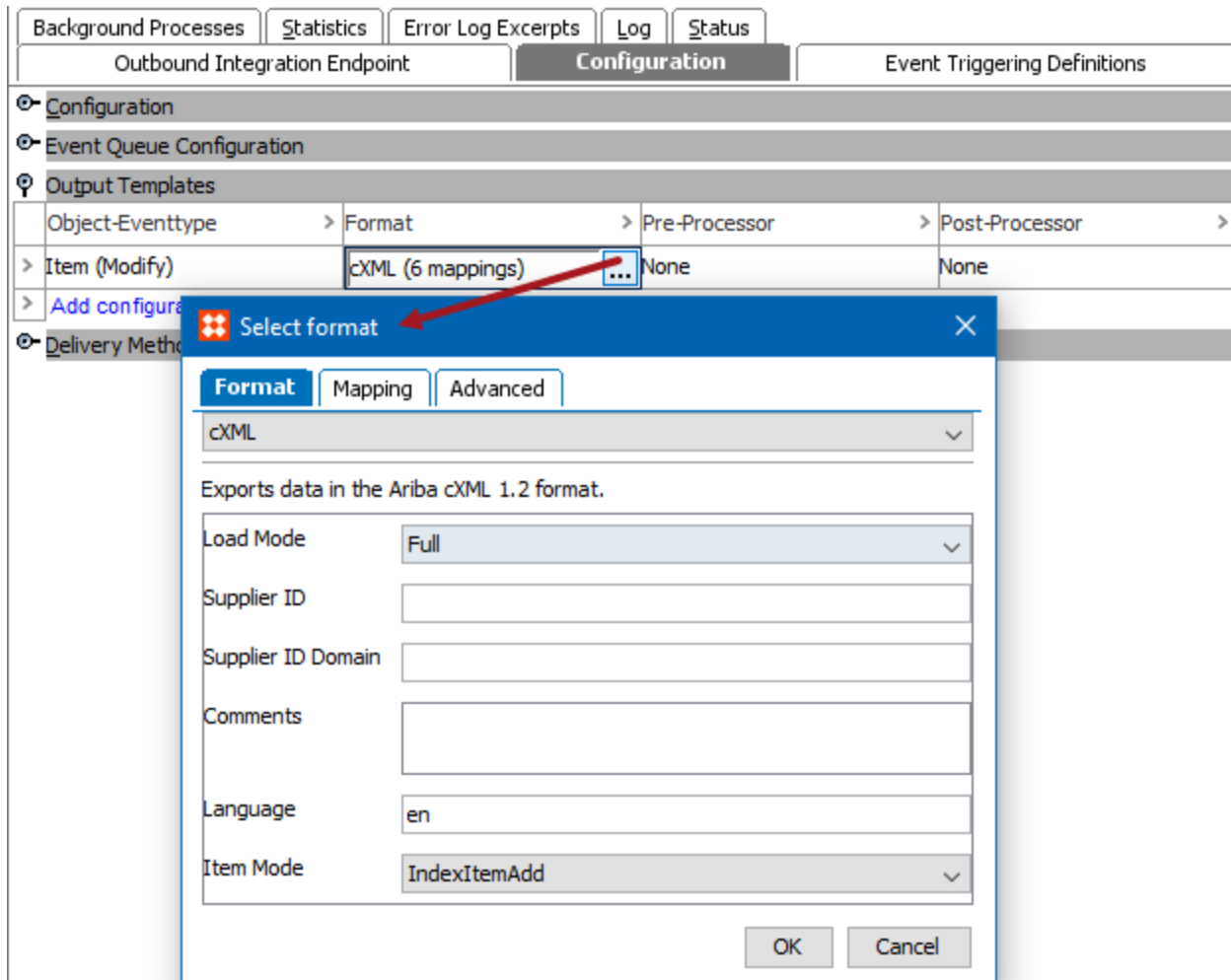
Back

Next

Finish

Cancel

OIEP

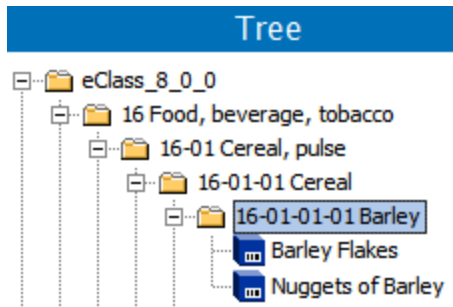


eCl@ss Format

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 format for any language using versions 7.0, 7.1, 8.0, 8.1, 9.0, and 9.1. For more information on the eCl@ss license, contact your Stibo 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

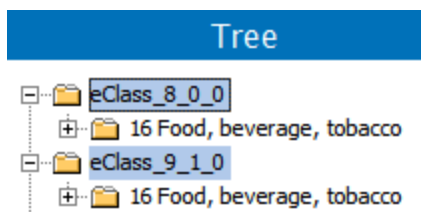
eCl@ss is available for selection only in the eCl@ss Classification Import wizard. For details, see **eCl@ss Classification Import**.

Mapping

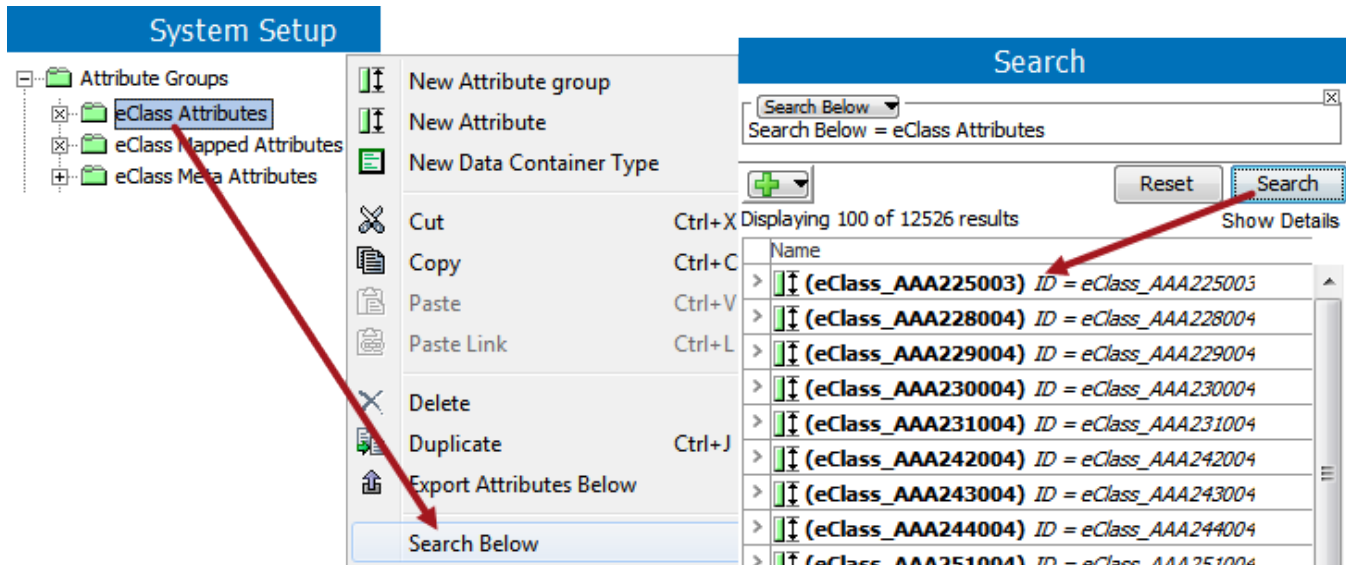
There is no need to recreate attributes you already have in STEP to gain the benefits of using standard eCl@ss attributes. After importing an eCl@ss classification hierarchy, you can map each eCl@ss attribute to a STEP attribute to merge the attributes and delete the eCl@ss attribute. For details, see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

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.

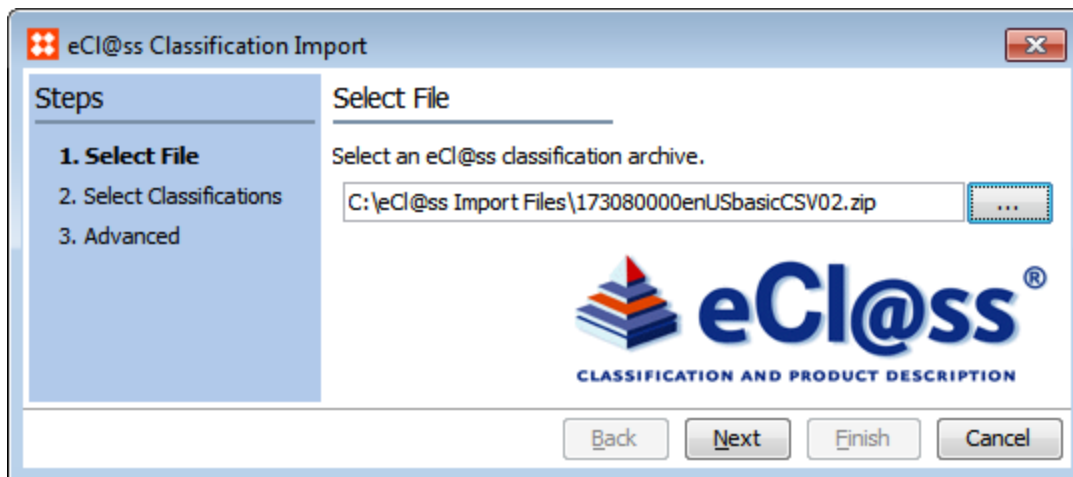


The import wizard also creates eCl@ss attributes, and saves all versions to the same attribute group. Depending on the number of classifications imported, System Setup may not allow browsing through the attributes created. However, you can view a portion of them, or find a particular one, using the right-click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



Inbound Data

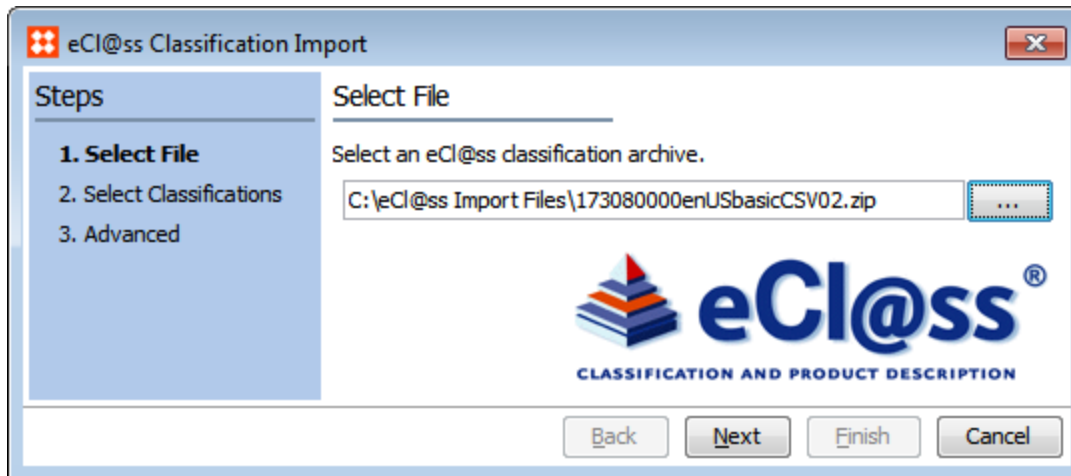
eCl@ss Classification Import



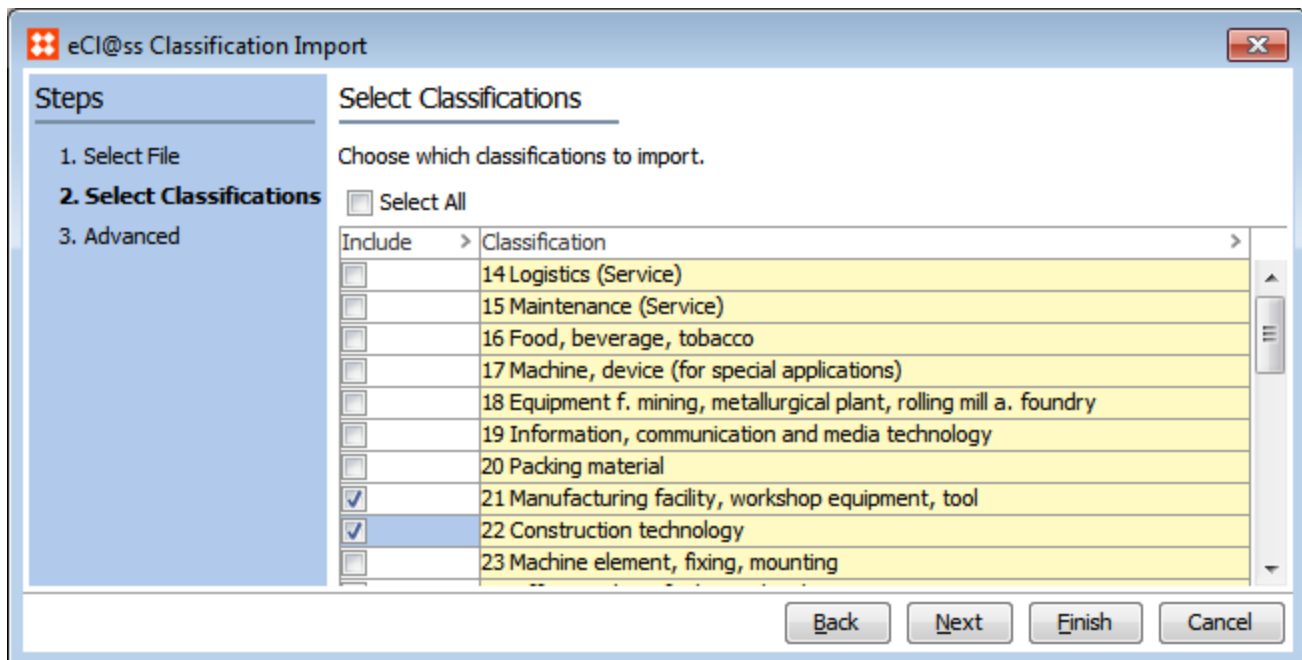
eCl@ss Classification Import

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**.

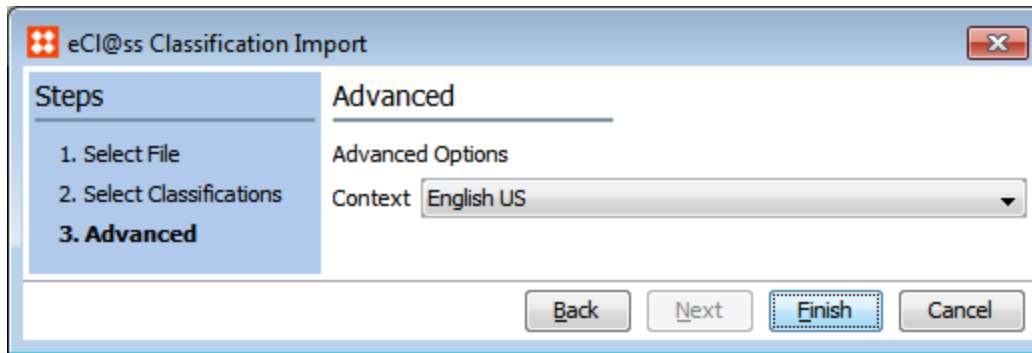
Note: The language of the file is indicated within the file name. For example, in the image below, '...enUS...' indicates that the file contains English / US.



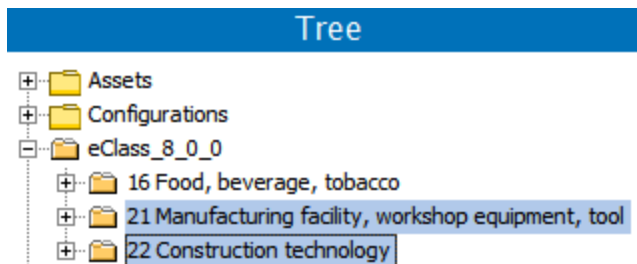
3. For **Select Classifications**, select all classifications to import, and click **Next**.



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



5. 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.



6. 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.

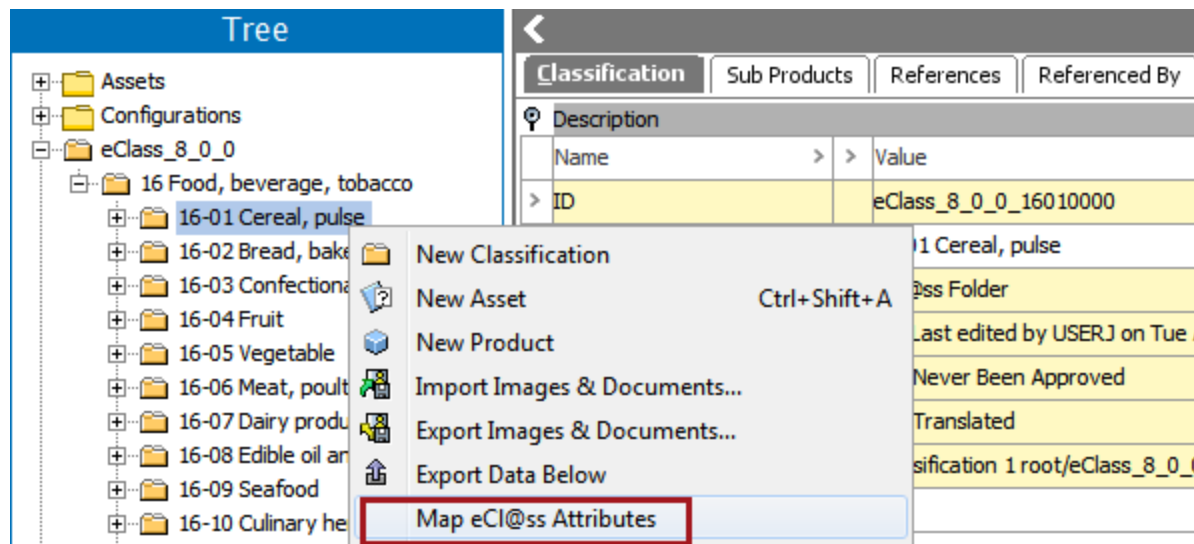
Map eCI@ss Attributes to STEP Attributes

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

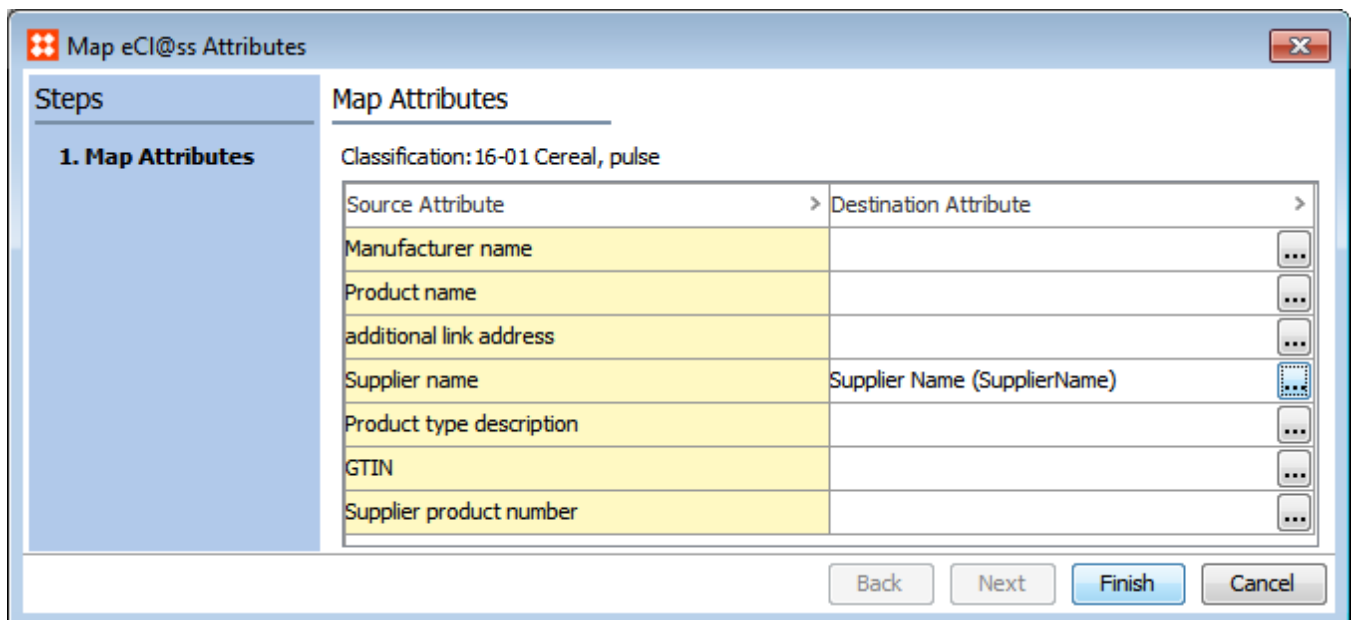
There is no need to recreate attributes you already have in STEP to gain the benefits of using standard eCI@ss attributes. After importing an eCI@ss classification hierarchy, you can map each eCI@ss attribute to a STEP attribute to merge the attributes and delete the eCI@ss attribute.

Mapping eCI@ss Attributes to STEP Attributes

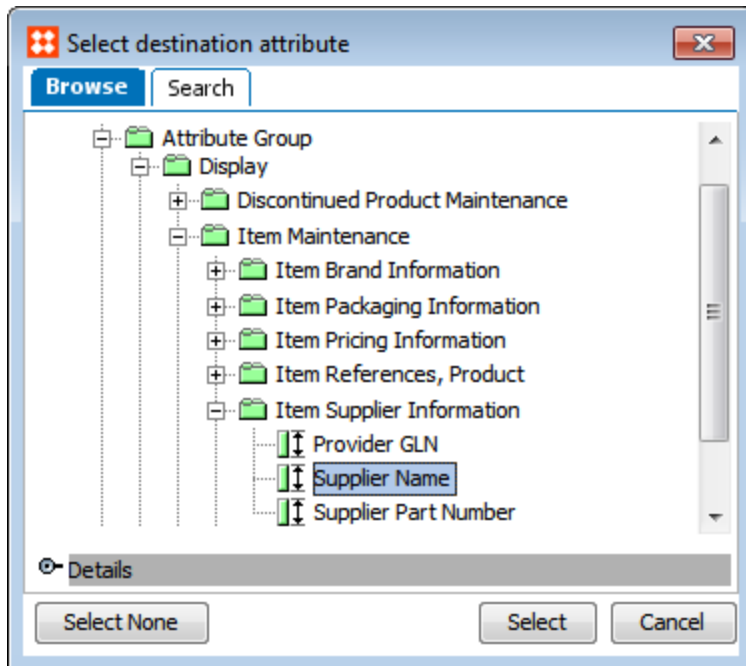
1. In the Tree, right-click the eCI@ss classification to map attributes *from* (the Source Attribute), and select the **Map eCI@ss Attributes** option.



2. In the wizard, click the ellipsis button (...) for the relevant eCI@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 6 classifications can be imported into STEP. The import tools cannot delete ETIM data or mark ETIM data as obsolete.

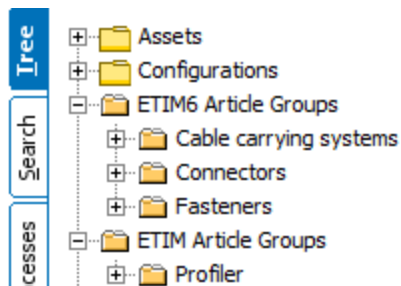
Both versions can exist in STEP simultaneously using two individual structures. Attributes, LOVs, Units, Object Types, and Classifications are not shared between ETIM versions.

- 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 uses XML and includes multiple languages (English, German, Dutch, and a number of other European languages). For more information about the ETIM 6 format, see **ETIM 6 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 6	
Attribute Label	Description	Attribute Label	Description
ID	Value displays ETIM preceding the ETIM ID	ID	Value displays ETIM6 preceding the ETIM ID
Name	The ETIM Description of the article group	Name	The ETIM Description of the article group
N/A	N/A	ETIM6 Abbreviation	Abbreviation from ETIM
ETIM Description	Description from ETIM, matches Name attribute	ETIM6 Description	Description from ETIM, matches Name attribute
ETIM ID	ID from ETIM, matches ID attribute	ETIM6 ID	ID from ETIM, matches ID attribute
ETIM Date	The date of the last change to the classification in ETIM	ETIM6 Date	The date of the last change to the classification in ETIM
ETIM Version	The Article Classification version from the ETIM system	ETIM6 Version	The Article Classification version from the ETIM system
ETIM Synonyms	Multivalued and contains 0 or more ETIM synonyms for the article classification	ETIM6 Synonyms	Multivalued and contains 0 or more ETIM synonyms for the article classification

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 | Tables | Statu

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/Accessories for industr
Visibility	
ETIM6 Abbreviation	abe
ETIM6 Date	11
ETIM6 Description	abe Accessories for industrial connectors
ETIM6 ID	abe EC002943
ETIM6 Synonyms	abe Accessories for industrial connectors Screen connector De-mounting tool Bracket Handgrip Accessories for heavy duty connectors Strain relief Mounting tool
ETIM6 Version	abe 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.
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	Features are imported as Number validation type.

Type	Type Description	Result
	number that allows a unit.	
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 6

- ETIM 6 - 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 6 format, see **ETIM 6 Format**.

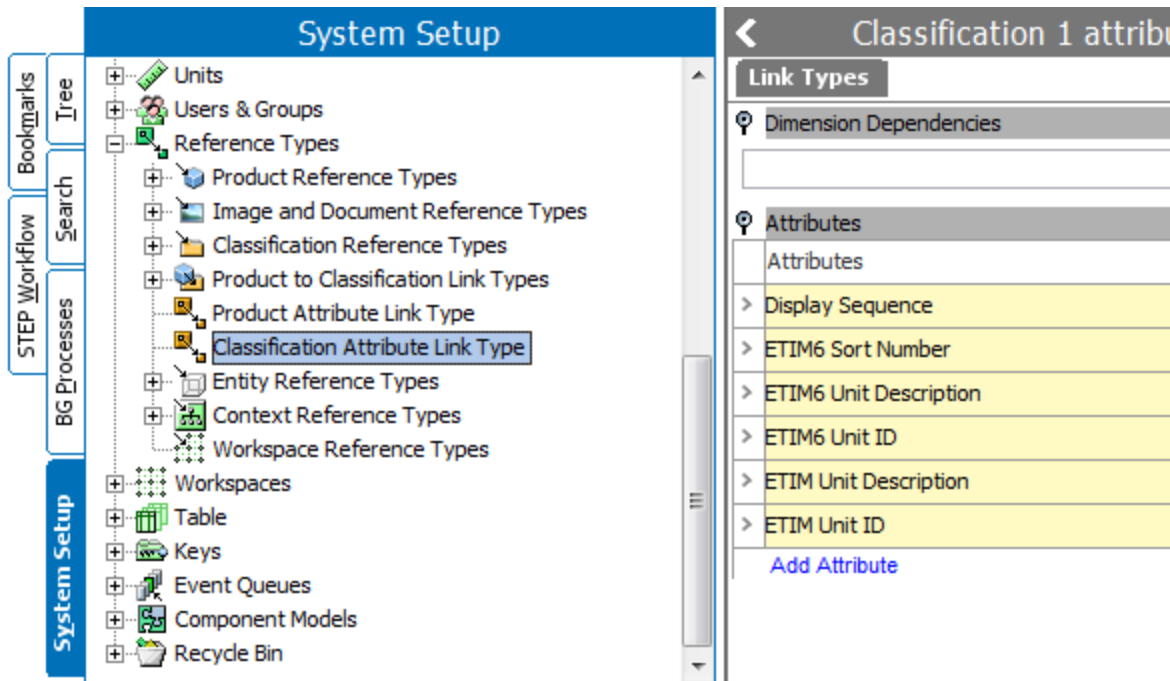
Valid Units are Determined by Classification to Attribute Link

An ETIM feature does not have a valid unit in ETIM 5 or ETIM 6. 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 satellitteknik	EG000033
Apparatlådor och skåp	EG000011
Arbetsutrustning	EG010330
Batterier och laddningsapparat	EG000053

Back
Next
Finish
Cancel

ETIM 6 Format

ETIM 6 uses XML 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 of the valid values. For more information about ETIM functionality in STEP, see **ETIM Format**.

Format Availability

ETIM 6 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 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.
- **Conversion Preview** displays the data selected in the Article Group ID(s) parameter in the selected language.

Note: Once the 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

FormatETIM 6

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

Languageen-GB

Article Group ID(s)EG000004, EG000049, EG000058

Conversion Preview:

Description	Article Group ID
Accessories for lighting	EG000030
Alarm installations, emergency call and signalling	EG000054
Antenna and satellite technique	EG000033
Auxiliary material	EG000060

BackNextFinishCancel

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

FormatETIM 6

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

Languageen-GB

Article Group ID(s)EG000023, EG000026

Conversion Preview:

Description	Article Group ID
Accessories for lighting	EG000030
Alarm installations, emergency call and signalling	EG000054
Antenna and satellite technique	EG000033
Auxiliary material	EG000060
Batteries, chargers, power cords	EG000053
Cabinet enclosures	EG000011
Cable and wire entry systems	EG000009
Cable carrying systems	EG000004

BackNextFinishCancel

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.

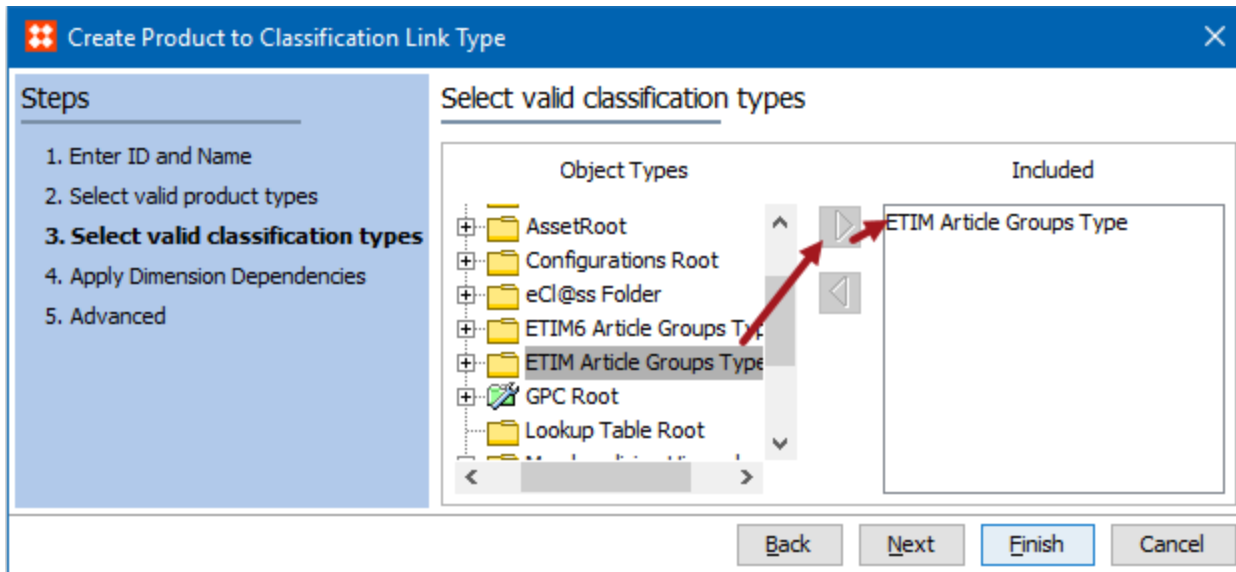
1. System Setup > Reference Types > **Product to Classification Link Types**, right-click and create a new Product to Classification Link Type.
2. Type an ID and Name for the new link type and click the **Next** button.

The screenshot shows the 'Create Product to Classification Link Type' dialog box. On the left, a 'Steps' panel 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 main area is titled 'Enter ID and Name' and contains two input fields: 'ID' with the value 'ETIMLink' and 'Name' with the value 'ETIM Link'. At the bottom, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

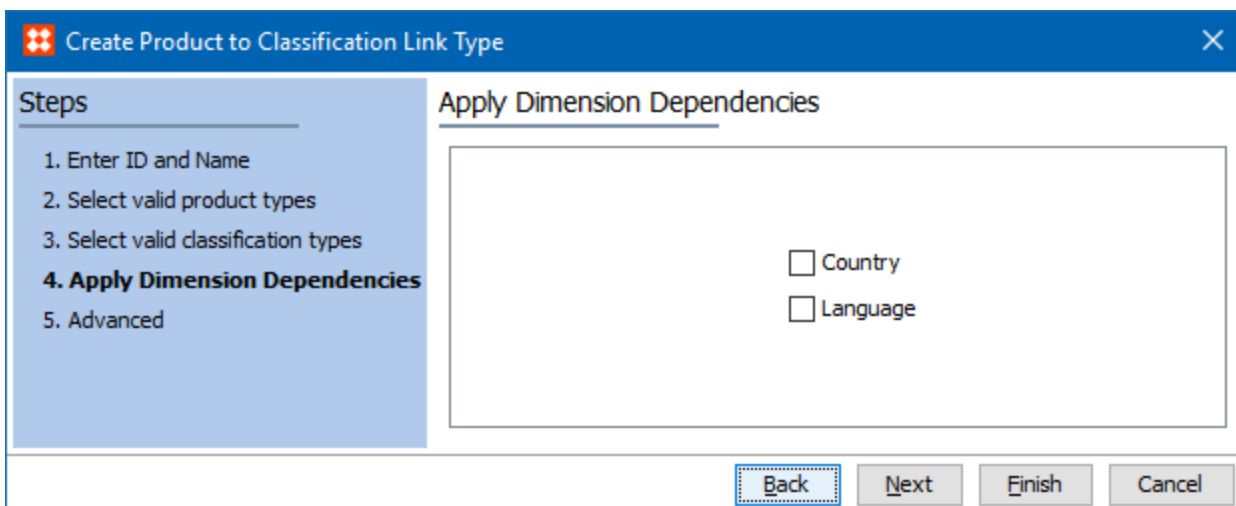
3. Select the valid product type(s), click the arrow button to move to the 'Included' list, and click the **Next** button.

The screenshot shows the 'Create Product to Classification Link Type' dialog box at Step 2: 'Select valid product types'. The 'Steps' panel on the left highlights step 2. The main area is divided into two panes: 'Object Types' on the left and 'Included' on the right. The 'Object Types' pane shows a tree structure with folders like 'External Products', 'Packaging Hierarchy', 'Primary Product Class', 'Product Overrides', 'Products', 'Level1', and 'Level2'. A red arrow points from 'Level1' in the 'Object Types' pane to the right-pointing arrow button between the two panes. The 'Included' pane is currently empty. At the bottom, there are four buttons: 'Back', 'Next', '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

eCl@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

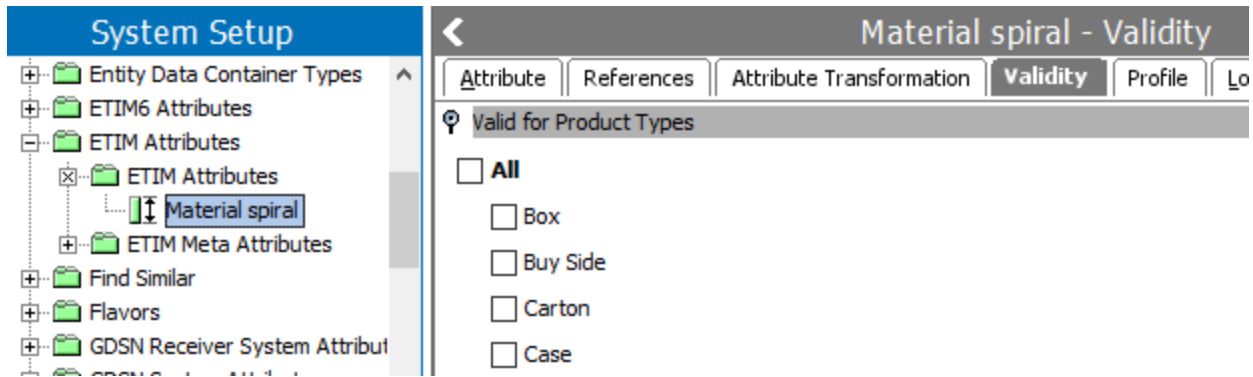
Set Validity on ETIM Attributes for Products

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

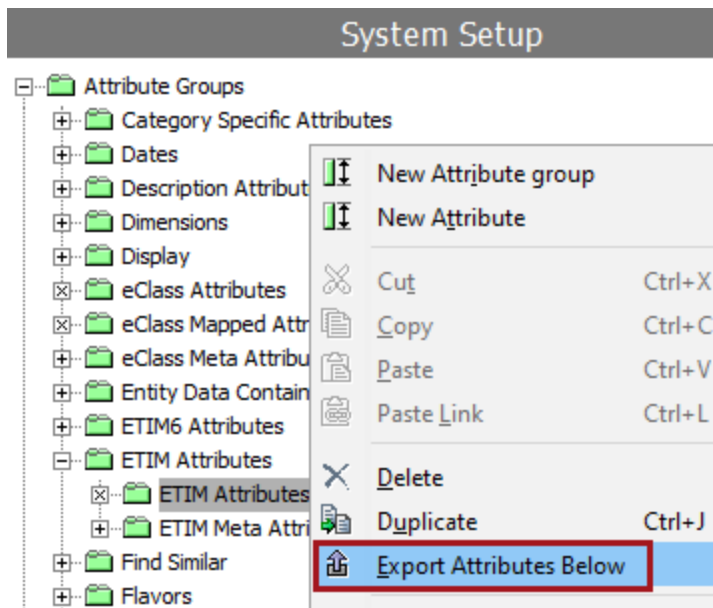
Use the following steps to set the valid product type on the Validity tab for the exported ETIM attributes.

© Stibo Systems - Confidential - STEP Trailblazer 8.1

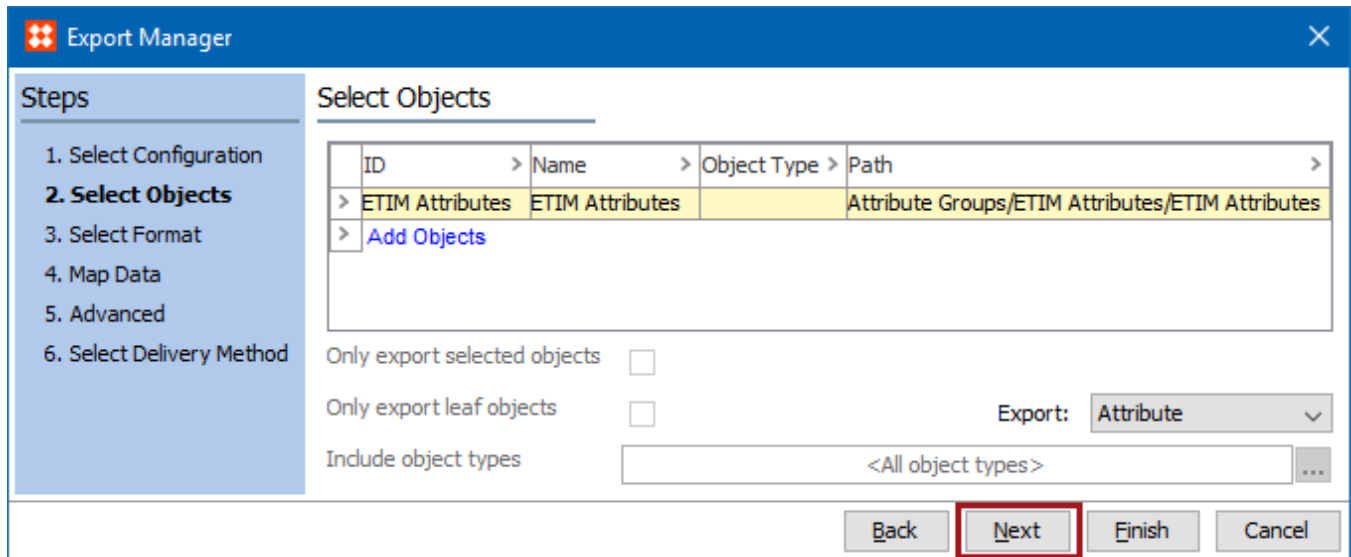
57



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

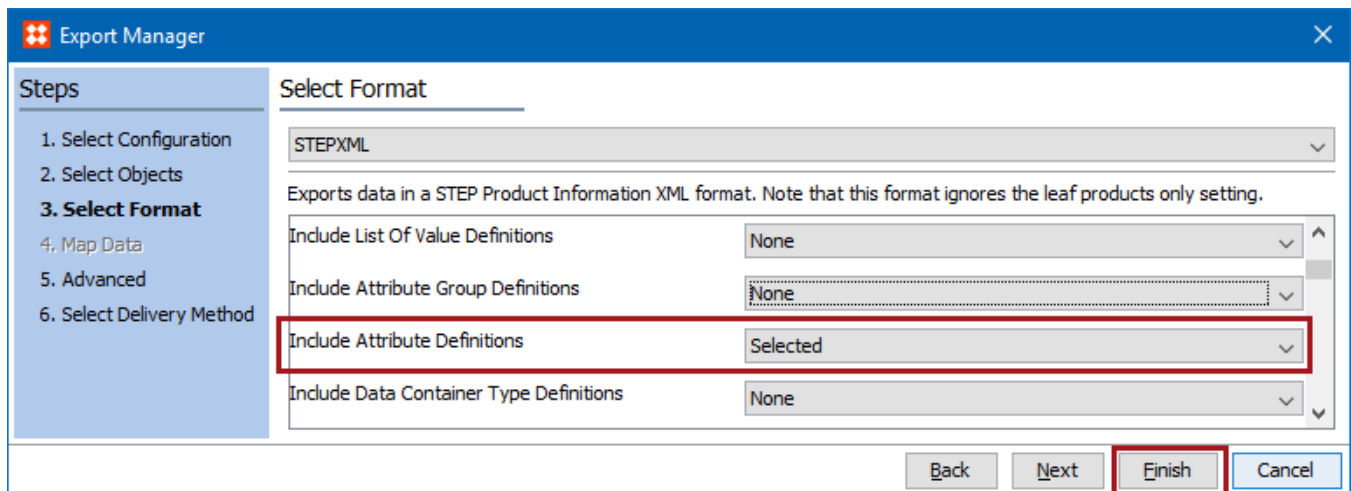
Select Objects

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
 Include object types: <All object types>

Export: Attribute

4. To ensure the smallest XML export file, on Select Format step:
 - Set the **Include Attribute Definitions** 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

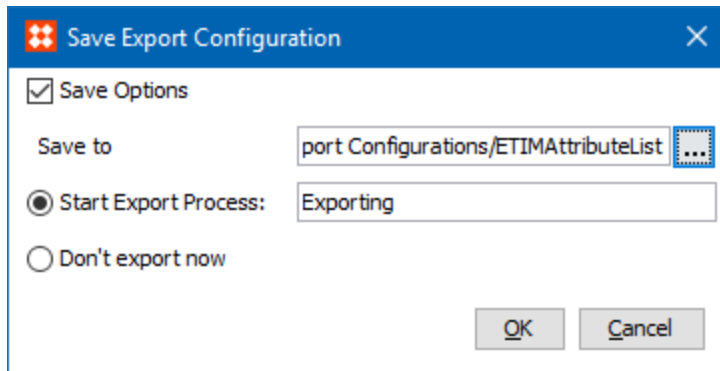
Select Format

STEPXML

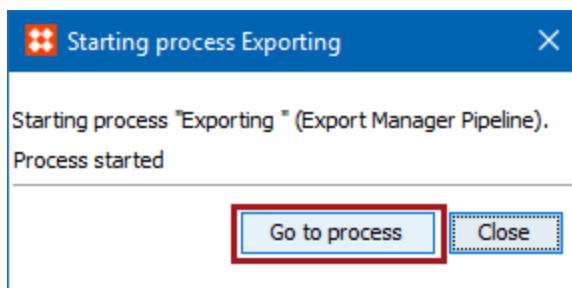
Exports data in a STEP Product Information XML format. Note that this format ignores the leaf products only setting.

Include List Of Value Definitions: None
 Include Attribute Group Definitions: None
Include Attribute Definitions: Selected
 Include Data Container Type Definitions: None

4. 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>

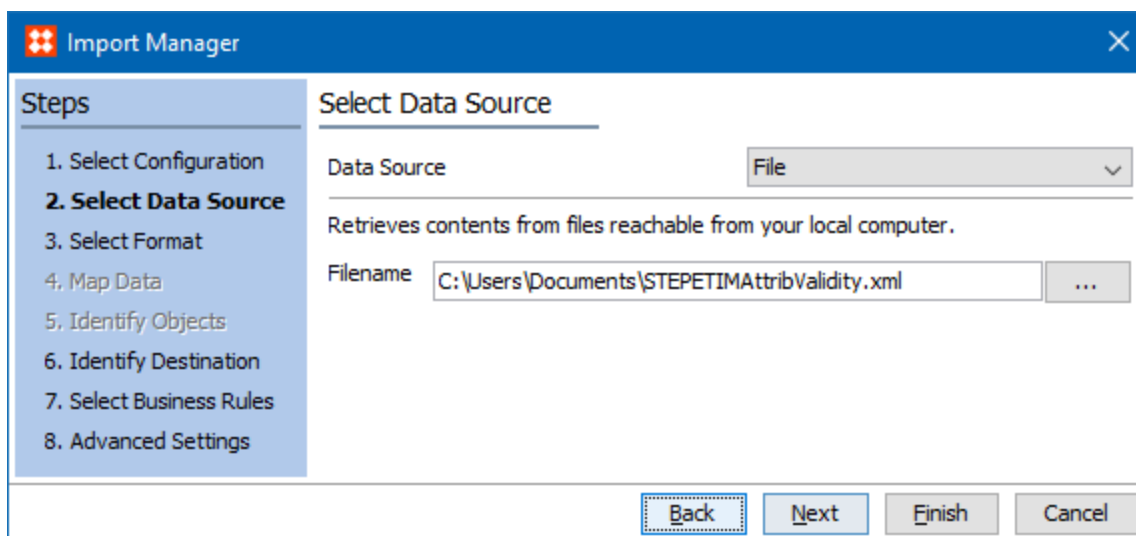
    <Attribute ID="ETIM EF021685 A" MultiValued="false" ProductMode="Normal" FullTextIndexed="false"
ExternallyMaintained="false" Derived="false" HierarchicalFiltering="true"
ClassificationHierarchicalFiltering="true" Selected="true" Referenced="true">
      <Name>Material spiral</Name>
      <ListOfValueLink 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>

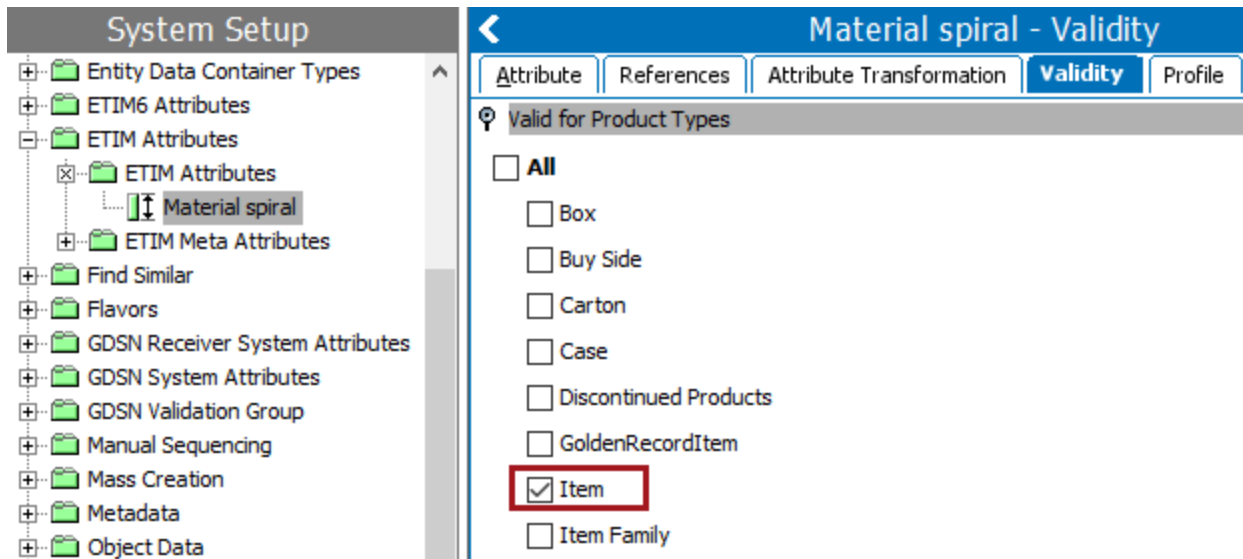
  </AttributeList>

</STEP-ProductInformation>
```

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



- 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.

- Start the Import Manager and complete the following steps:
 - Import Manager - Select Configuration**
 - Import Manager - Select Data Source**
- For the Import Manager Select Format step, select the appropriate ETIM version.
 - ETIM version 5 using **ETIM and ETIM v2 Format**
 - ETIM version 6 using **ETIM 6 Format**
- 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**

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

Navigation: 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

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

OK Cancel

Preview

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

Save Reset Cancel

9. Enter 'ETIM ' or 'ETIM6 ' (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

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**

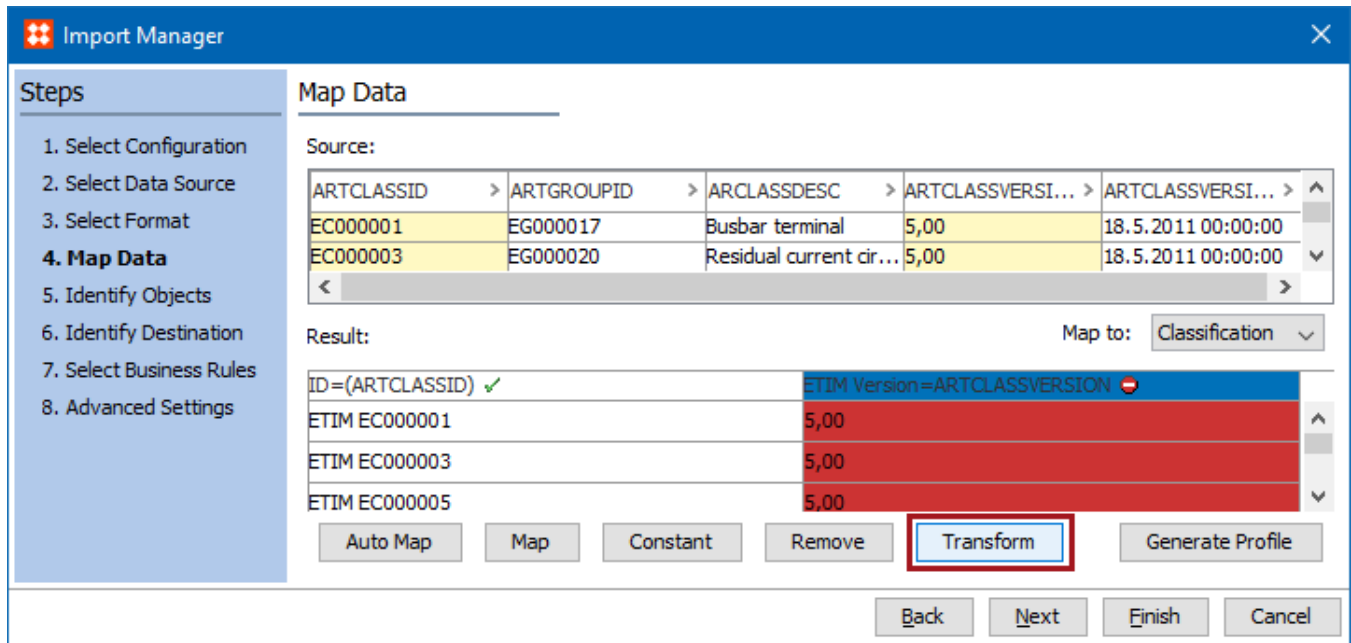
ETIM Version (ETIM Version)

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

Result:

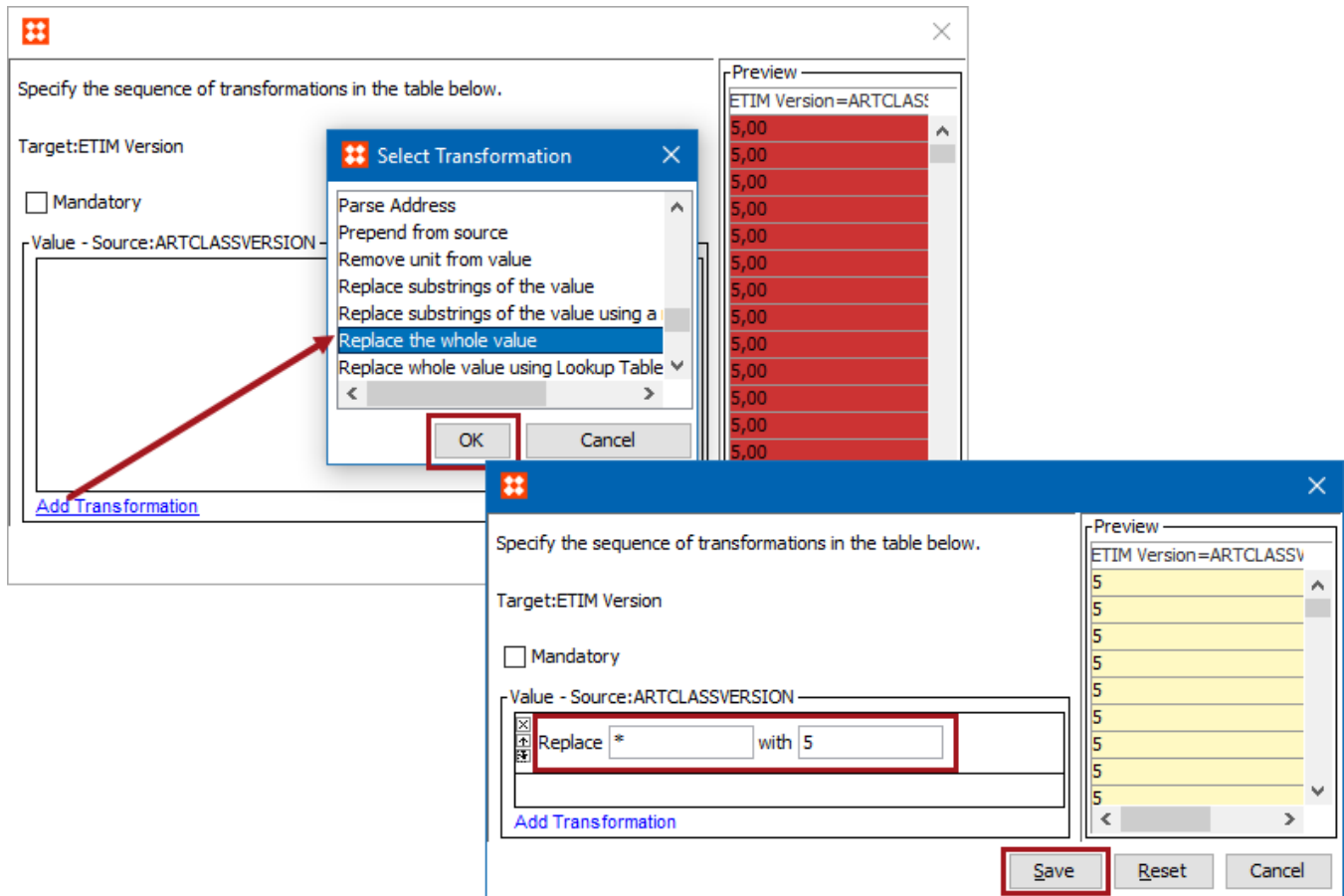
ID=(ARTCLASSID) ✓	ETIM Version=ARTCLASSVERSION ⚠
ETIM EC000001	5,00
ETIM EC000003	5,00
ETIM EC000005	5,00

Map to: Classification

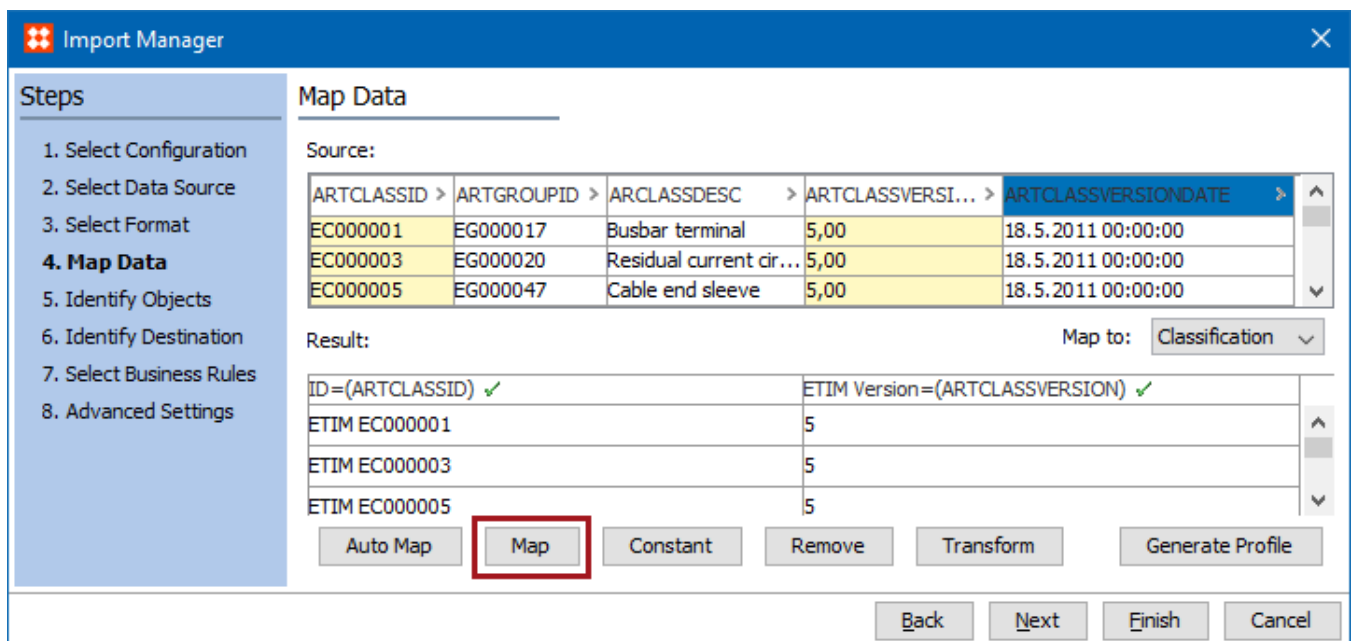
Auto Map Map Constant Remove **Transform** Generate Profile

Back Next Finish Cancel

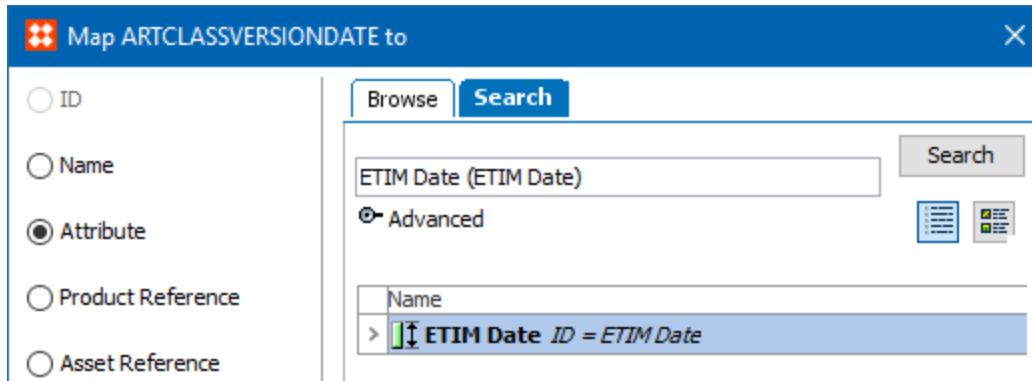
14. In the Transformation dialog, click the **Add Transformation** link, select the **Replace the whole value** option. For the values, add **replace * with** and enter a single digit (either 5 or 6) based on the version being imported, and click the **Save** button.



15. The Result table ID=ARTCLASSVERSION column now displays the 5 or 6 for all rows. In the Source table, select the **ARTCLASSVERSIONDATE** column and click the **Map** button below the Result table.

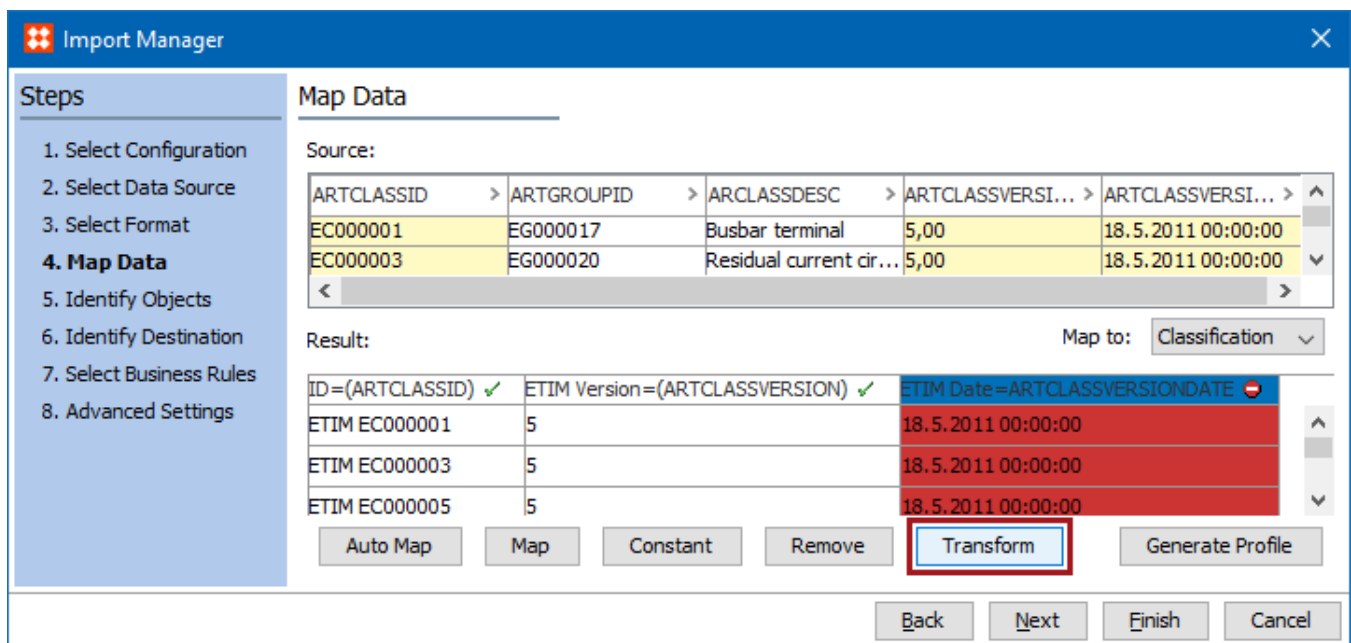


16. In the Map To window, select the **Attribute** radio button, search for and select the **ETIM Date** attribute, and click **OK**.



The dialog box titled "Map ARTCLASSVERSIONDATE to" has a close button (X) in the top right. On the left, there are five radio buttons: ID, Name, Attribute (selected), Product Reference, and Asset Reference. On the right, there are "Browse" and "Search" tabs. The "Search" tab is active, showing a search bar with "ETIM Date (ETIM Date)" and a "Search" button. Below the search bar, there is an "Advanced" section with a search icon and a list of results. The first result is "Name" and the second is "ETIM Date ID = ETIM Date", which is highlighted with a blue selection bar.

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.

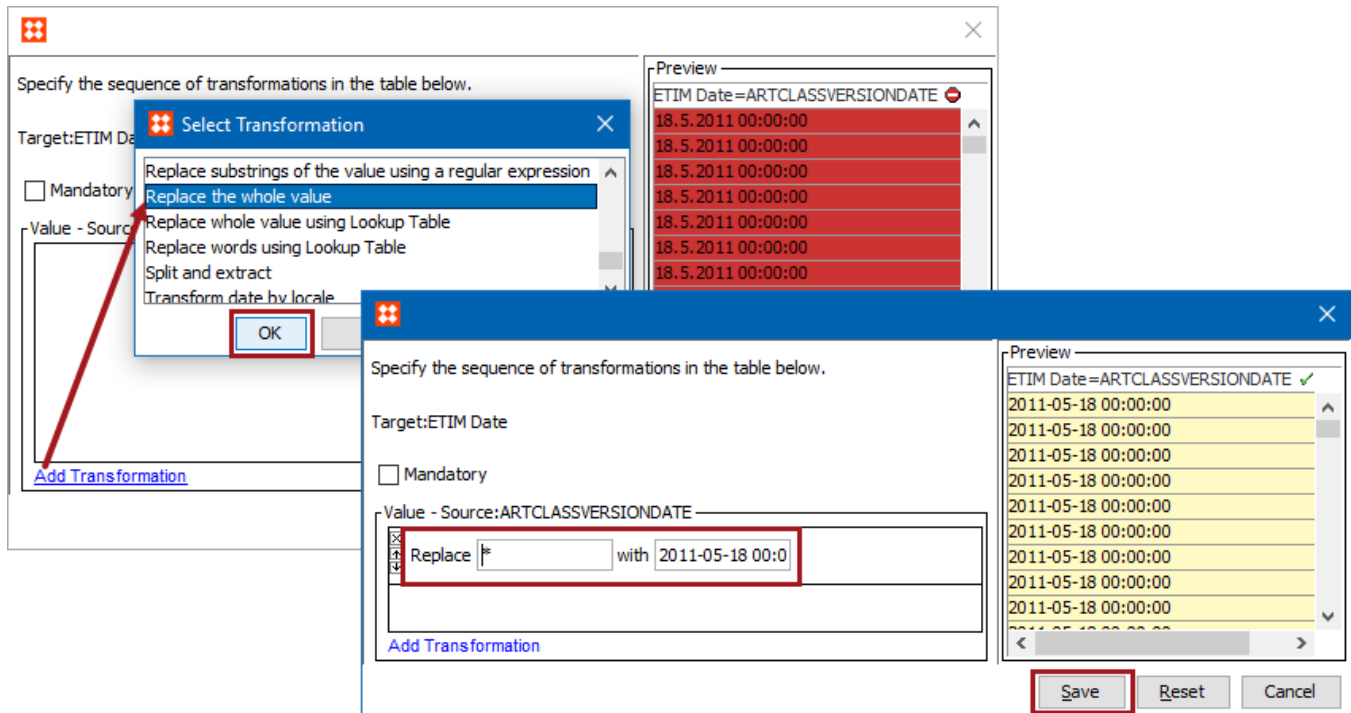


The "Import Manager" dialog box has a close button (X) in the top right. On the left, there is a "Steps" list with eight items: 1. Select Configuration, 2. Select Data Source, 3. Select Format, 4. Map Data (selected), 5. Identify Objects, 6. Identify Destination, 7. Select Business Rules, and 8. Advanced Settings. The main area is titled "Map Data" and contains a "Source" table and a "Result" table. The "Source" table has five columns: ARTCLASSID, ARTGROUPID, ARCLASSDESC, ARTCLASSVERSI..., and ARTCLASSVERSI... The "Result" table has three columns: ID=(ARTCLASSID) ✓, ETIM Version=(ARTCLASSVERSION) ✓, and ETIM Date=ARTCLASSVERSIONDATE ✗. The "ETIM Date=ARTCLASSVERSIONDATE" column is highlighted in red. Below the "Result" table, there are buttons: Auto Map, Map, Constant, Remove, Transform (highlighted with a red box), and Generate Profile. At the bottom, there are buttons: Back, Next, Finish, and Cancel.

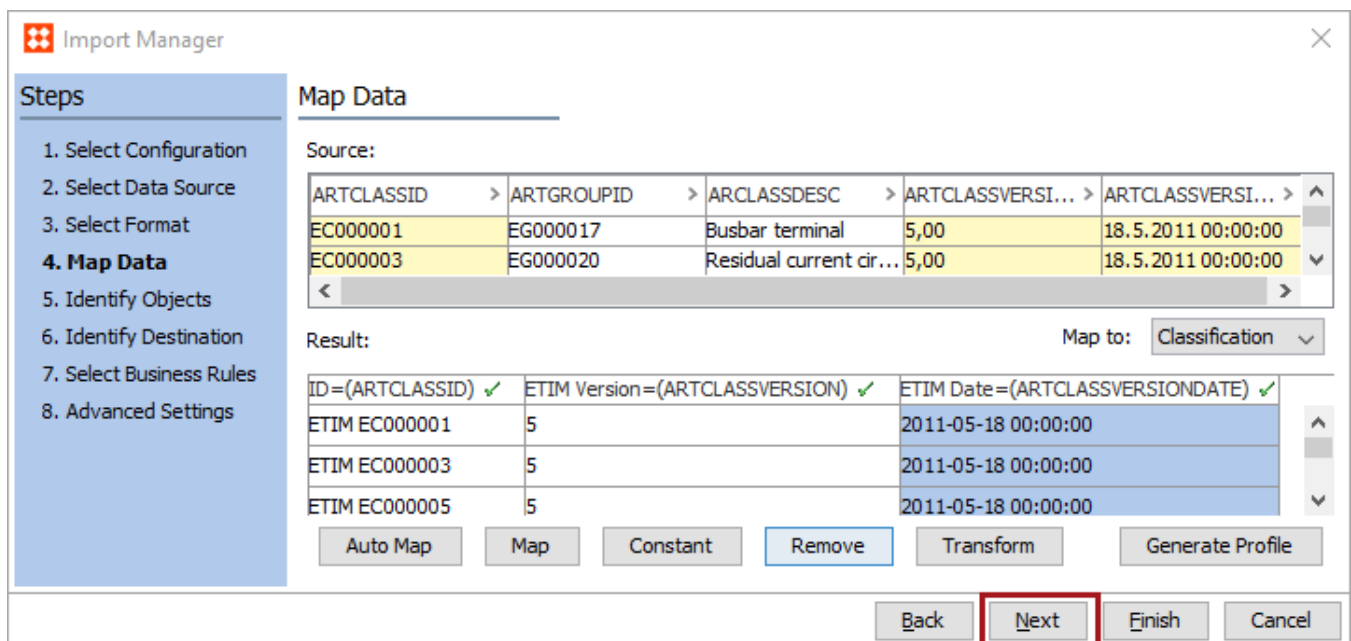
ARTCLASSID	ARTGROUPID	ARCLASSDESC	ARTCLASSVERSI...	ARTCLASSVERSI...
EC000001	EG000017	Busbar terminal	5,00	18.5.2011 00:00:00
EC000003	EG000020	Residual current cir...	5,00	18.5.2011 00:00:00

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

18. 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.

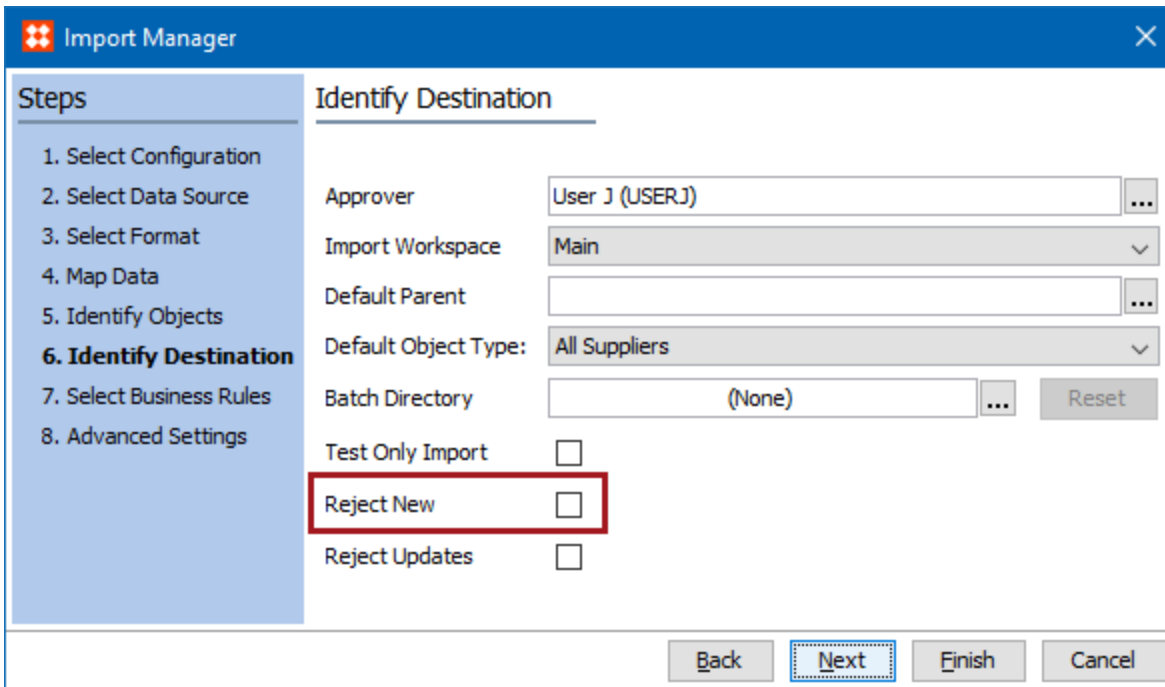


19. All three (3) mapped columns now show valid values in the Result table. Click **Next** (selecting defaults) until the Import Manager Identify Destination step displays.



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.



The screenshot shows the 'Import Manager' dialog box with the 'Identify Destination' step selected. The 'Steps' list on the left includes: 1. Select Configuration, 2. Select Data Source, 3. Select Format, 4. Map Data, 5. Identify Objects, 6. Identify Destination (highlighted), 7. Select Business Rules, and 8. Advanced Settings. The 'Identify Destination' section contains the following fields and options:

- Approver: User J (USERJ)
- Import Workspace: Main
- Default Parent: (empty)
- Default Object Type: All Suppliers
- Batch Directory: (None)
- Test Only Import: ☐
- Reject New: ☐** (highlighted with a red box)
- Reject Updates: ☐

At the bottom, there are buttons for Back, Next (highlighted with a blue dashed border), Finish, and Cancel. A 'Reset' button is also present next to the Batch Directory field.

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-/distribu
 - 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 & Documents
Tables
St

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 12:04:54 EST 2017
> Approved	✗ Never Been Approved
> Translation	Not Translated
> Path	Classification 1 root/ETIM 5 Article Groups/Distribution boards/Cable tree
> Visibility	
> ETIM Date	2011-05-18 00:00:00
> ETIM Description	abc Cable tree
> ETIM ID	abc EC000273
> ETIM Synonyms	abc Wiring set abc Cable tree abc 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 version for Excel within STEP are:

- Excel 97/2003 - limit 65,536 rows
- Excel 2007 - limit 1,048,576 rows

Keep the following points in mind when working with STEP data using 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 **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 TM, 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.

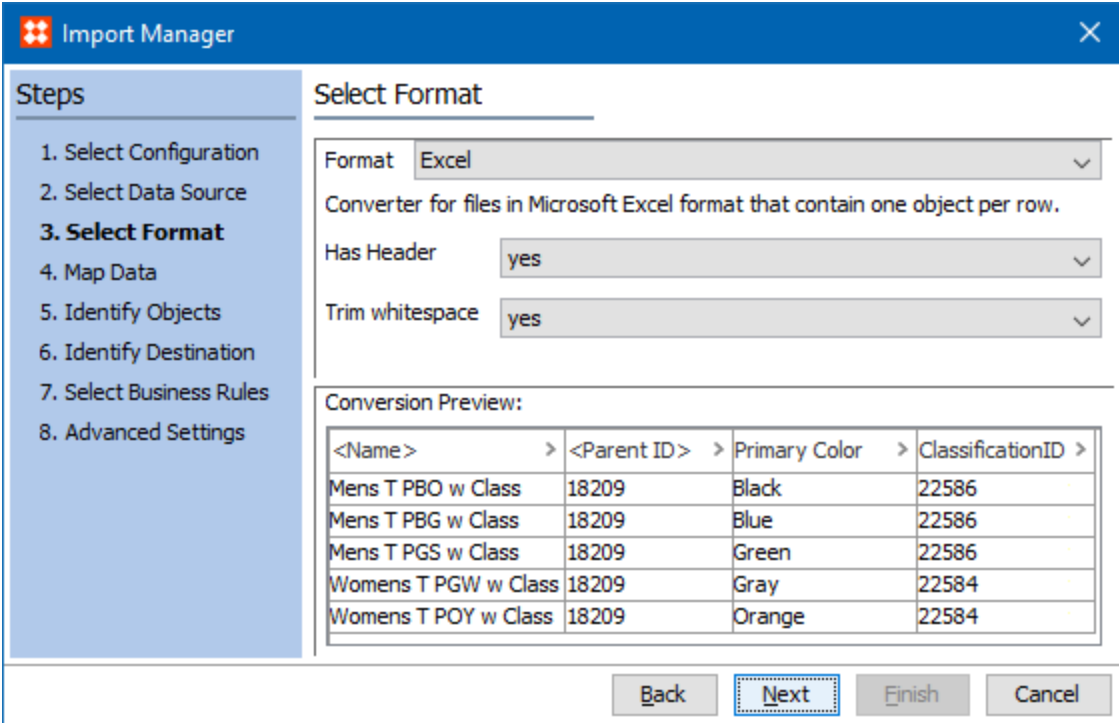
- To remove an attribute value, use the entry **[delete]** in your load file. Empty cells do not cause values to be deleted. If the value being deleted was inherited, the result is not a blank field, but the inherited value is restored.
- Enter attribute values and units in the same cell, with the units 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.

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.

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

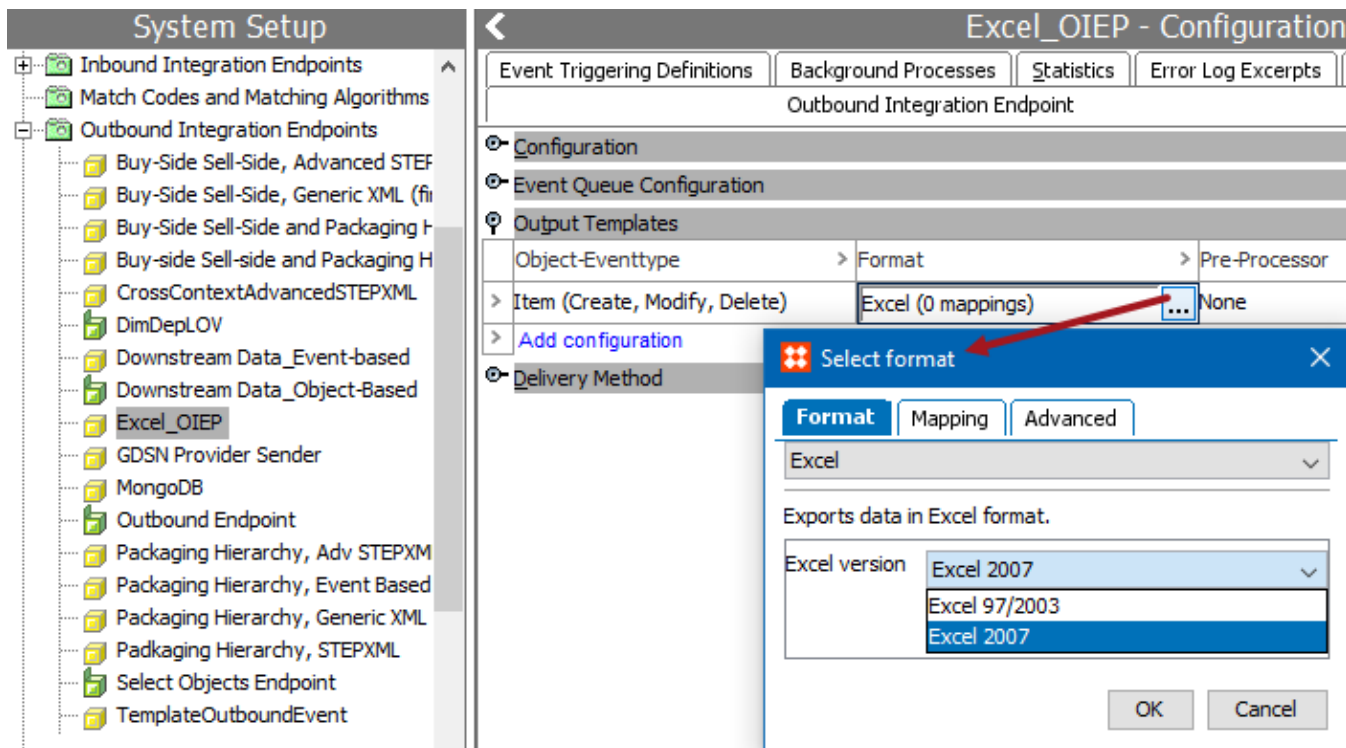
Excel

Exports data in Excel format.

Excel version
Excel 97/2003
Excel 97/2003
Excel 2007

Back
Next
Finish
Cancel

OIEP



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).

For more information on configuring Smartsheet Web UI components, see the **Smartsheets in Web UI** section of the **Web User Interfaces** 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.

Note that the Smartsheet format and Supplier Web UI may require special licenses to use. Ask your Stibo account manager for more information.

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

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 **Creating 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.

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

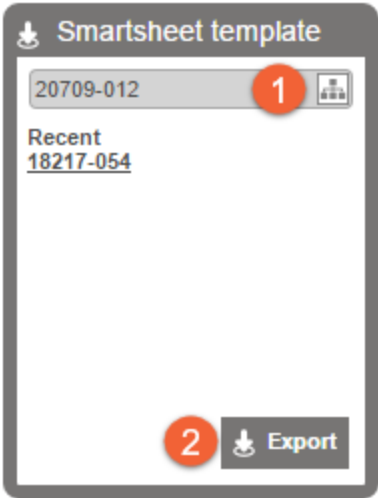
Several setup steps are required before Smartsheets can be used to initiate new products into STEP:

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 doc, and ultimately re-importing the edited Smartsheet back into the Supplier Web UI. Detailed steps are listed below:

- 1. In the **Smartsheet Export** widget, click the browse hierarchy icon  and select a valid node based on the object type(s) specified in the Smartsheet template. 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.
- 2. Click **Export**.

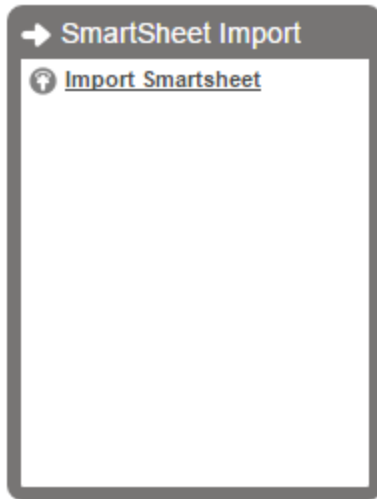


- 3. Enter product data into the exported Smartsheet.

	B		D	H	I
9	Validate sheet	Next error			
	Duplicate row	Delete row			
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	...

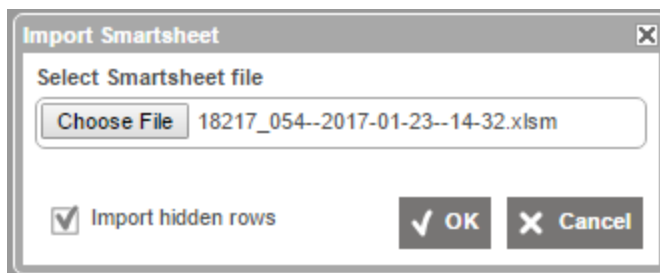
For more information on navigating and editing a Smartsheet, see the **Using a Smartsheet** section of the documentation.

4. 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.
5. Once the sheet has been validated, save the document.
6. Navigate back to the Supplier Web UI and click on the Smartsheet Import Action link.



7. 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.

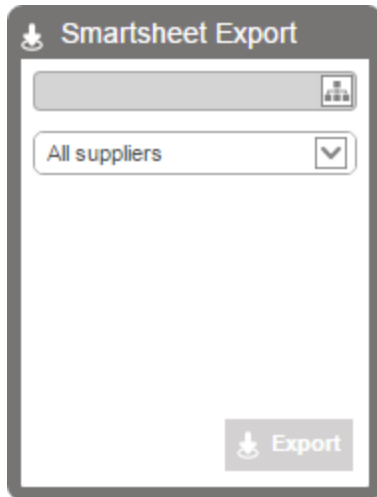
Exporting a Template with a Supplier Selector

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 drop down will not be present.

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		
	Next error		
9			
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, a business condition can be set up 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 in 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

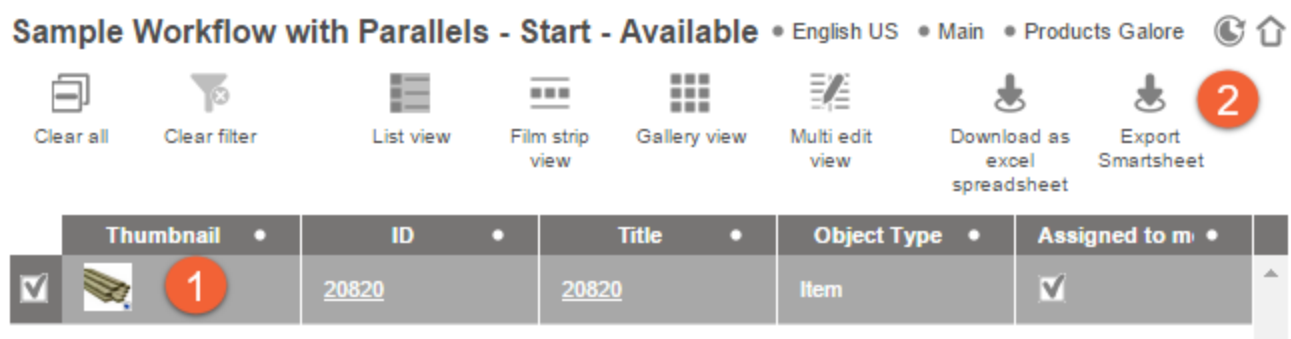
Several setup steps are required before Smartsheets can be used to maintain product data in STEP:

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.

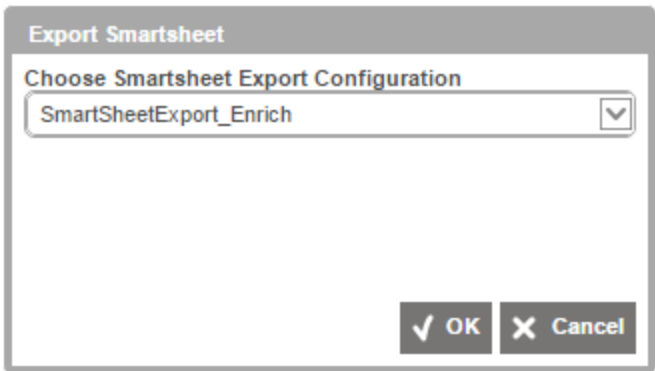
Product Maintenance

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, select the products you wish to export.
- 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**.

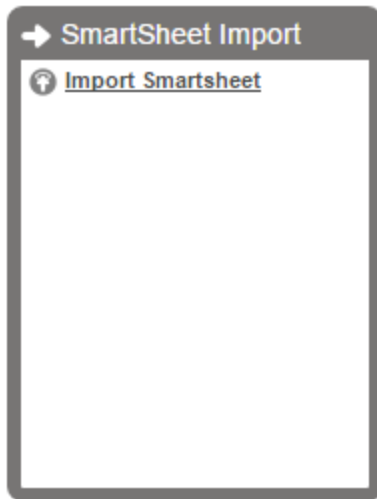


- 4. Modify the product data in the exported Smartsheet.

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

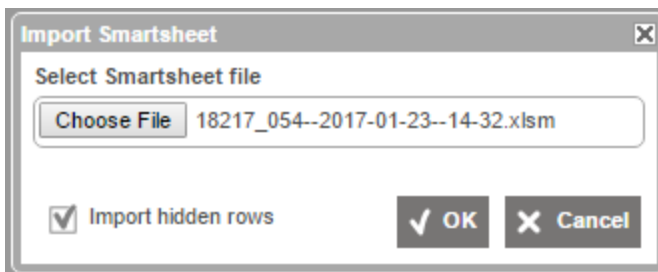
For more information on navigating and editing a Smartsheet, see the **Using a Smartsheet** section of the documentation.

5. 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.
6. Once the sheet has been validated, save the document.
7. Navigate back to the Supplier Web UI and click on the Smartsheet Import Action link.



8. 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 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.

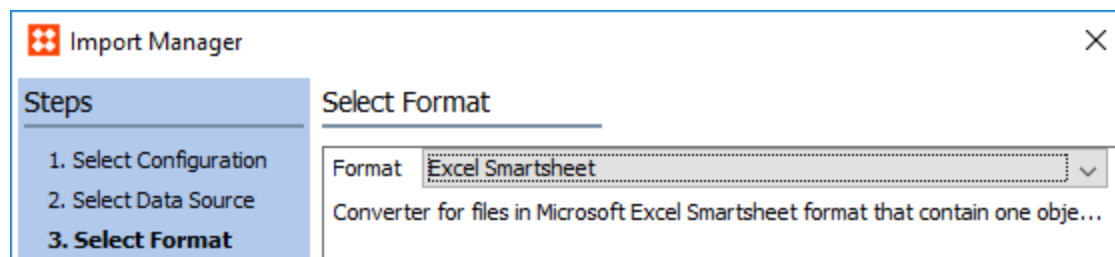
Creating an Import Configuration

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.

Step 3 - Select Format

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



Step 8 - Advanced Settings

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.

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

Step 2 - Select Objects

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
 <All object types>

Export: Product

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.

Step 3 - Select Format

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

Back

Next

Finish

Cancel

1. In the **Excel version** parameter, use the dropdown to select which Excel version to use for the export. Excel 2007 is recommended. This parameter is mandatory.
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 references.

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 step 2 of the wizard. A maximum of 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 step 5 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 drop-down 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 step 4 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.

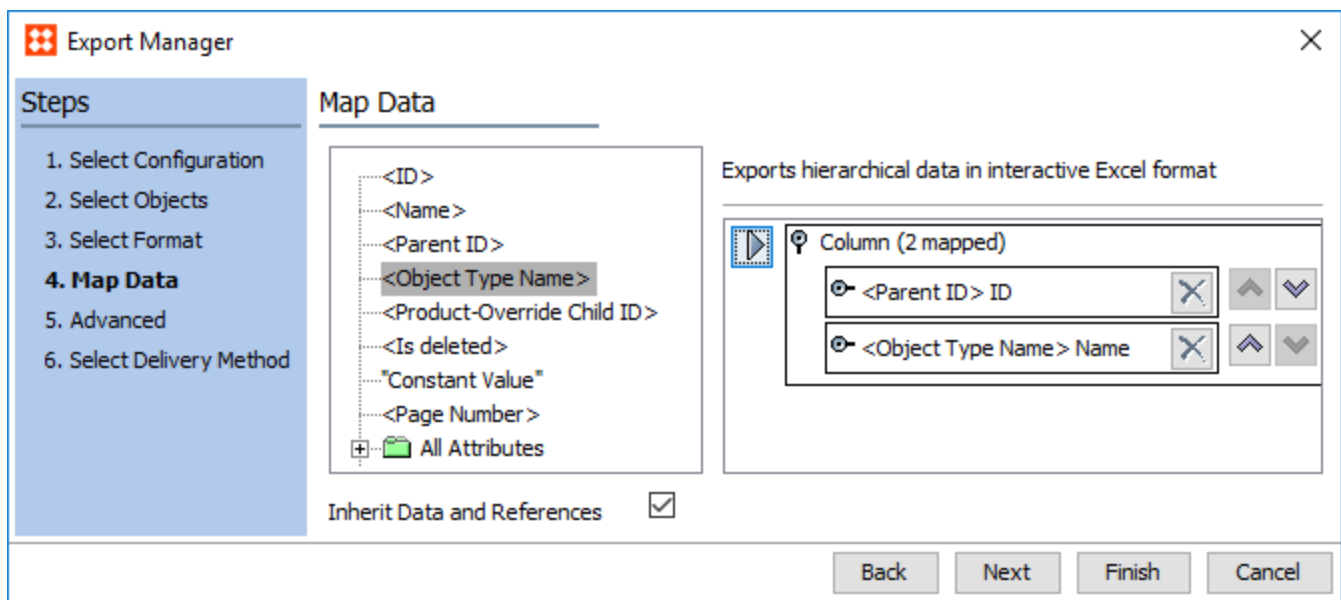
Step 4 - Mapping Data for Smartsheets

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.

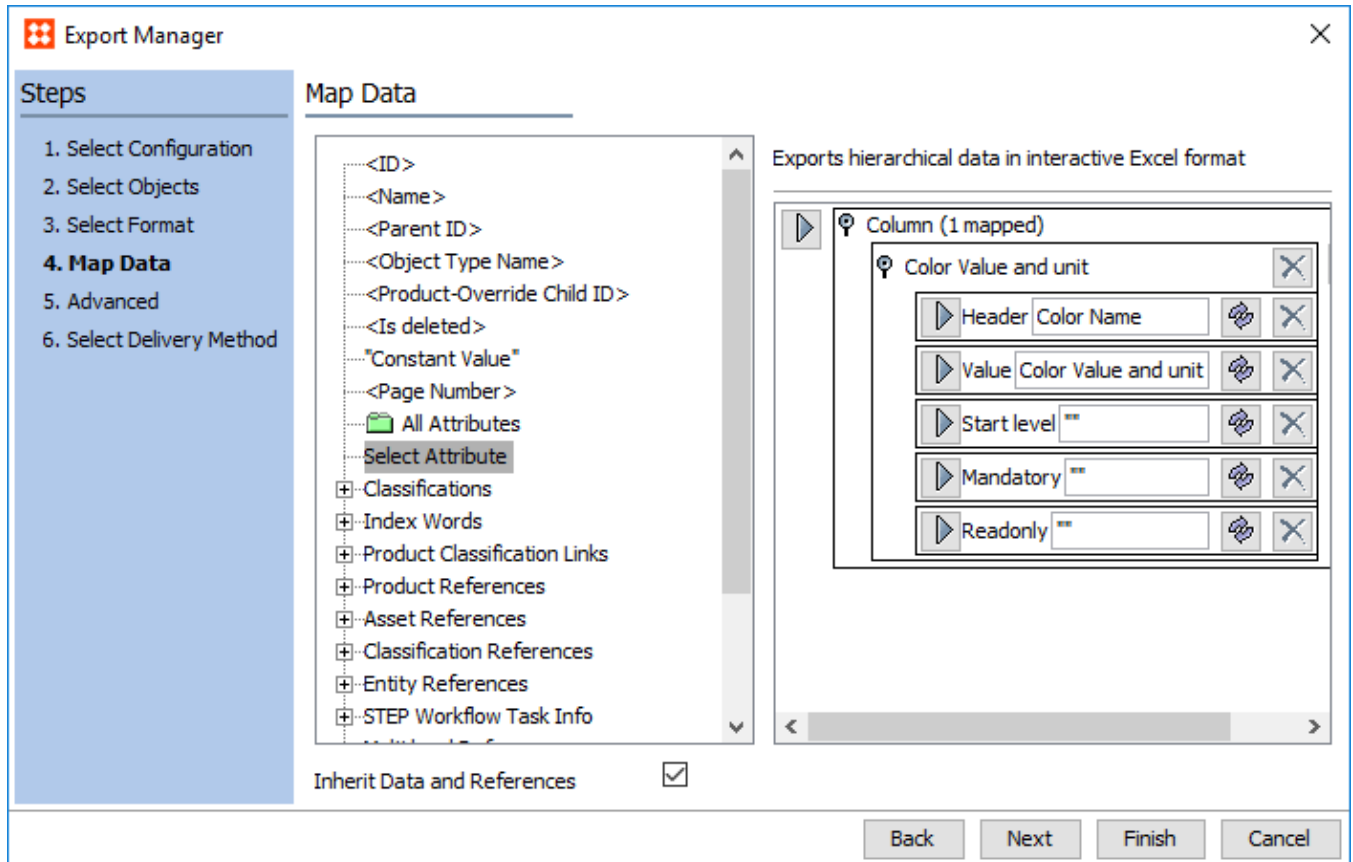
The Start level columns are also used for grouping products at each level. For example, all rows with 'T-Shirt' in the **Name** column (which is designated as a Start level column) become one product. All rows that have both 'Pink' in the **Color** column and 'T-Shirt' in the **Name** column become one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



Export Manager

Steps

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

Map Data

Exports hierarchical data in interactive Excel format

Column (1 mapped)

Color Value and unit

Header	Color Name		
Value	Color Value and unit		
Start level	**		
Mandatory	**		
Readonly	**		

Inherit Data and References ☒

Back Next Finish Cancel

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 recommend 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 a 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.

Step 5 - Advanced

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.

Step 6 - Select Delivery Method

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.

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.

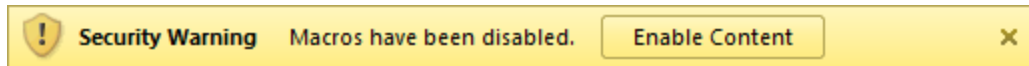
The screenshot displays the 'System Setup' interface. On the left, a tree view shows the navigation path: **System Setup** > **Users & Groups**. The main panel, titled 'System Settings', shows a list of configuration items. The 'Context Help metadata attribute' is highlighted with a red box, and its value is set to 'AttributeHelpText'.

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 classifiction to product link	SupplierLink

Using a Smartsheet

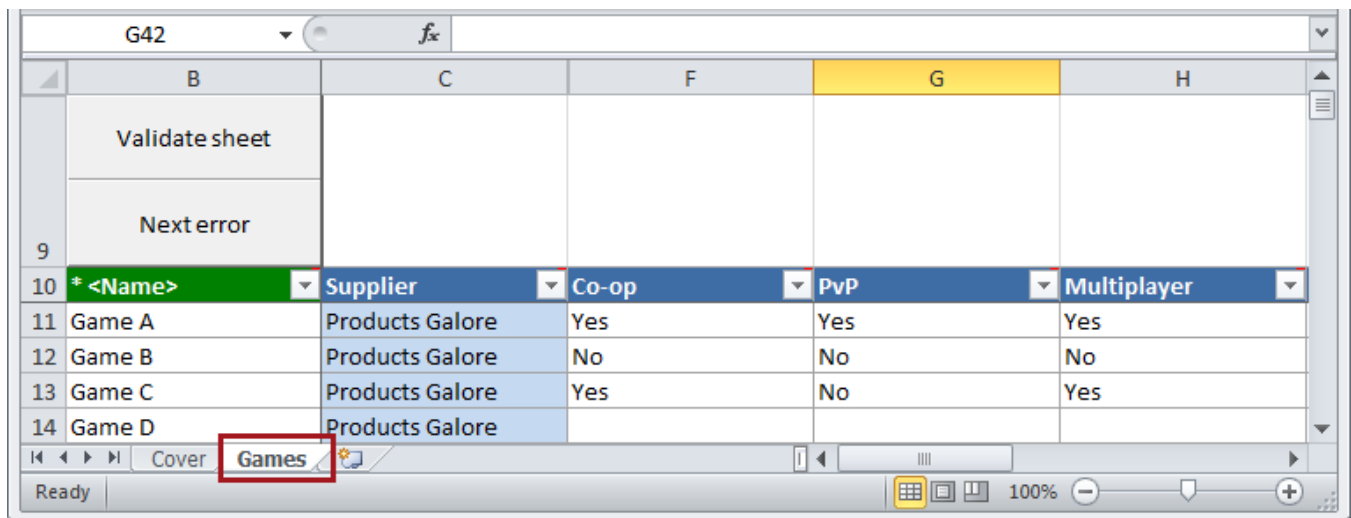
Unlike typical Excel sheets, Smartsheets are very restrictive and must be handled in a very specific way.

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.

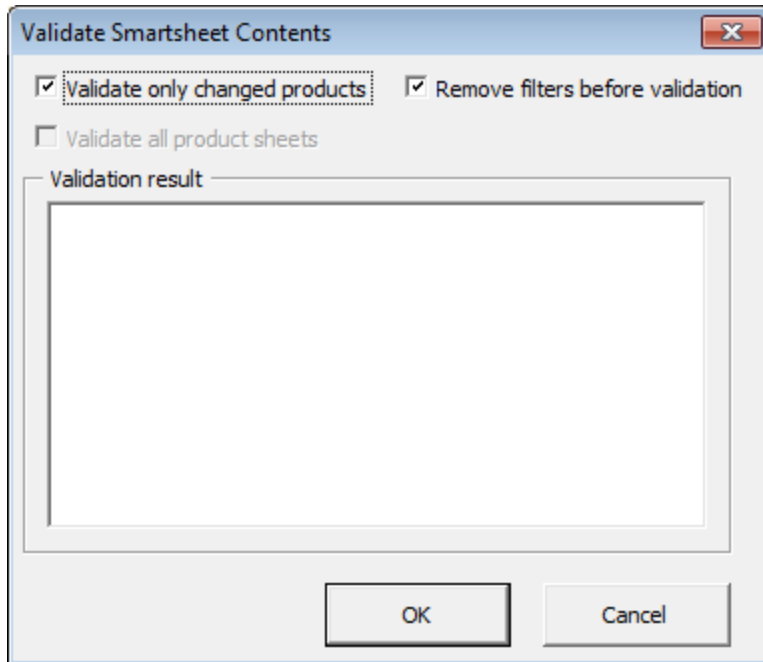


3. Enter the data for the objects to be imported.

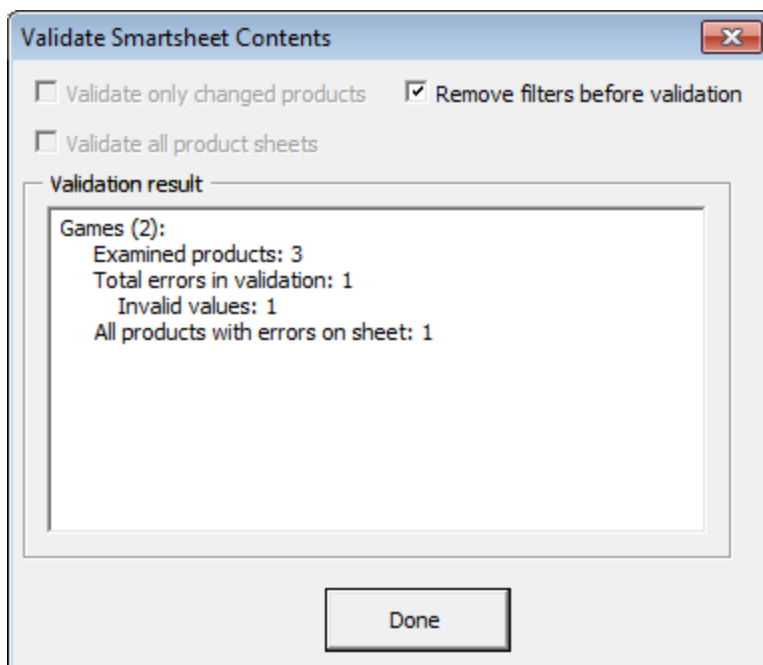
Note: In template sheets, all rows are editable, but any unmapped columns are locked to the user. In maintenance sheets, all unmapped rows and columns are locked.

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 check box 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 hid 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.

8. 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.

	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
14		
15	Microsoft Excel	
16	Click OK to delete the selected row, number 13.	
17	OK Cancel	
18		
19		
20		
21		

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.

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			
	Next error			
9				
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.

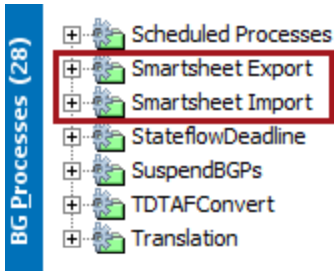
Important: Attributes mandatory for transitions are not enforced in Smartsheets as the selected transition is not known.

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 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 their 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.



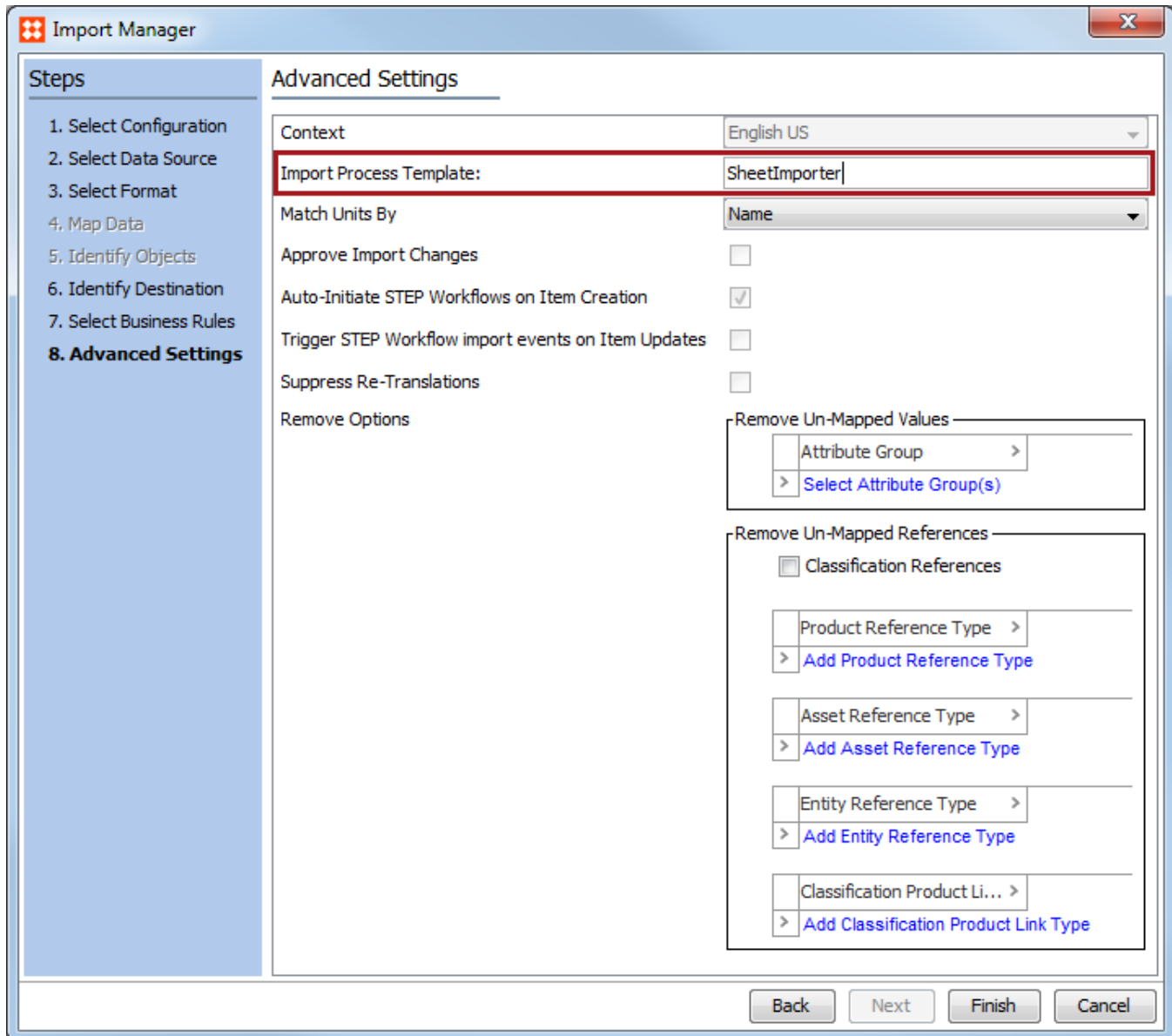
Important: 'BackgroundProcess.ProcessType.SheetExporter.Queue' and 'BackgroundProcess.ProcessType.SheetImporter.Queue' default to the same queue as Export Manager and Import Manager, and must be changed in sharedconfig.properties before they can run in their own queues. This step must be completed before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

Using Smartsheet Import Background Process

On Step 8 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](#)

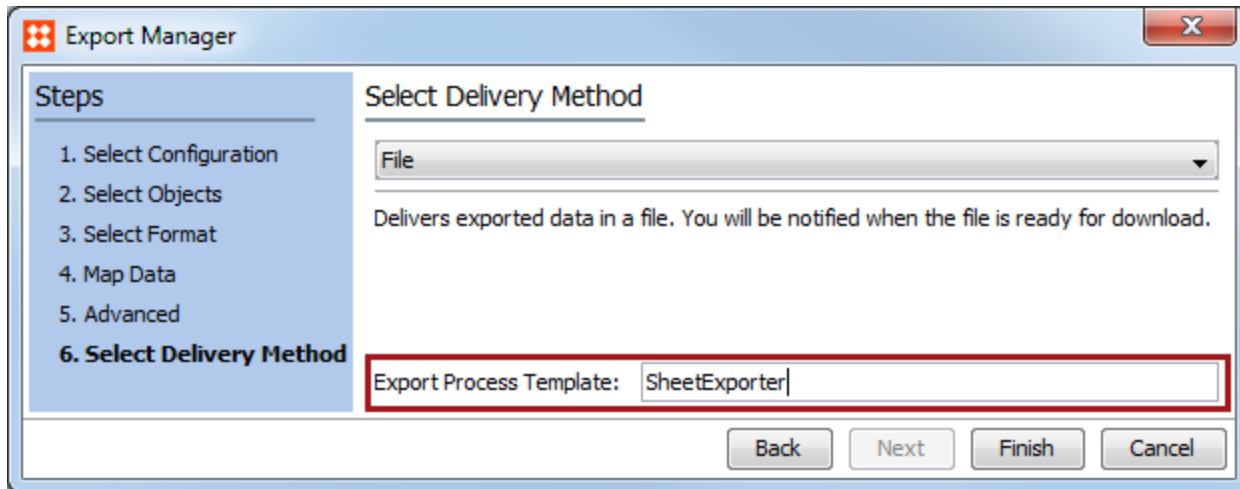
Back Next Finish Cancel

Using Smartsheet Export Background Process

On Step 6 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.



Smartsheet System Properties

Configuration properties are available for Smartsheets that 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 steered using a configuration property called 'Smartsheet.LimitSupplierBrowsing'.

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 is used from the client while maintaining product cross-references and asset references.

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's 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; i.e. it is possible to export a Smartsheet from PortalServerA and have it validated at PortalServerB as long as they share the same STEP backbone.

This setting is steered via the 'Smartsheet.ValidationService' property.

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, the following instructions steps to be followed:

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 STEP config.properties:
 - 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 **Mapping**.

Inbound Data

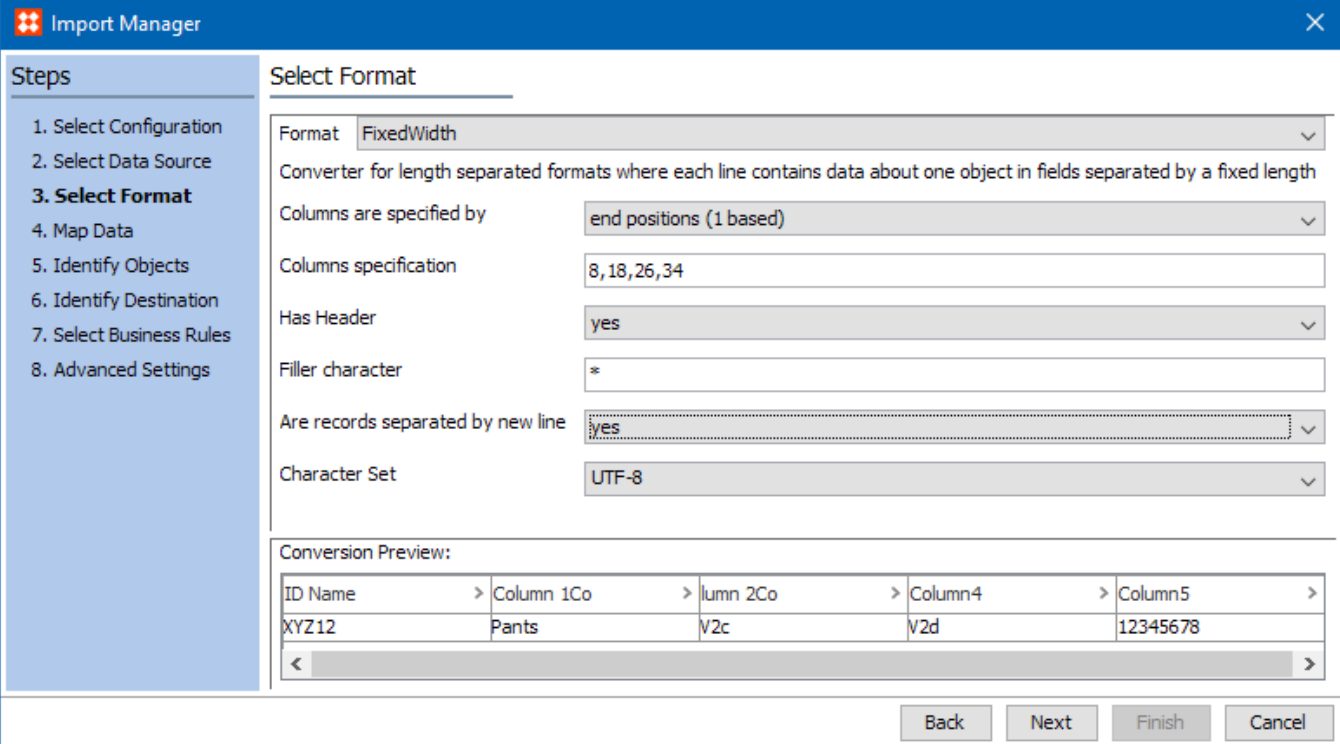
For examples of ways to modify fixed width data for import and further explanations on the inbound parameters, see **FixedWidth Format Configuration Examples**.

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

Buttons: 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 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	

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

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: **Flatplanner**
Imports an Flatplanner Excel files

Conversion Preview:

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

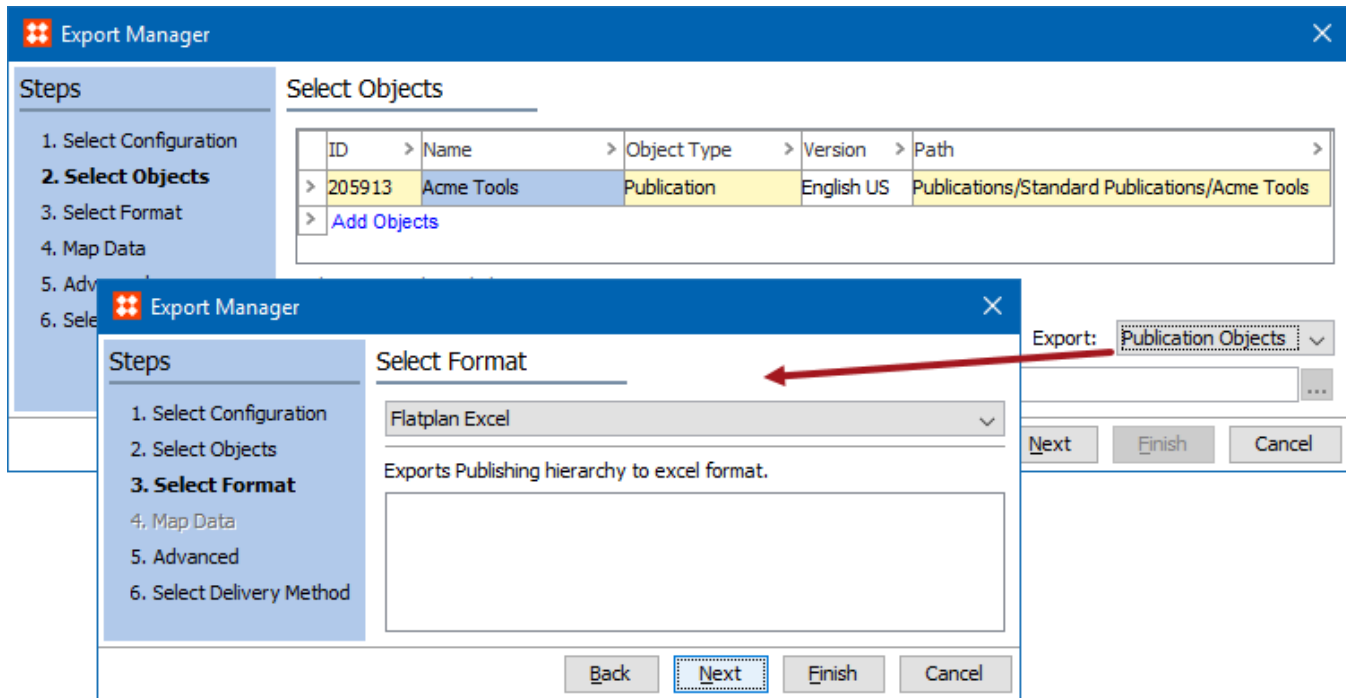
Back Next Finish Cancel

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.



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 and attribute 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 **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.

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

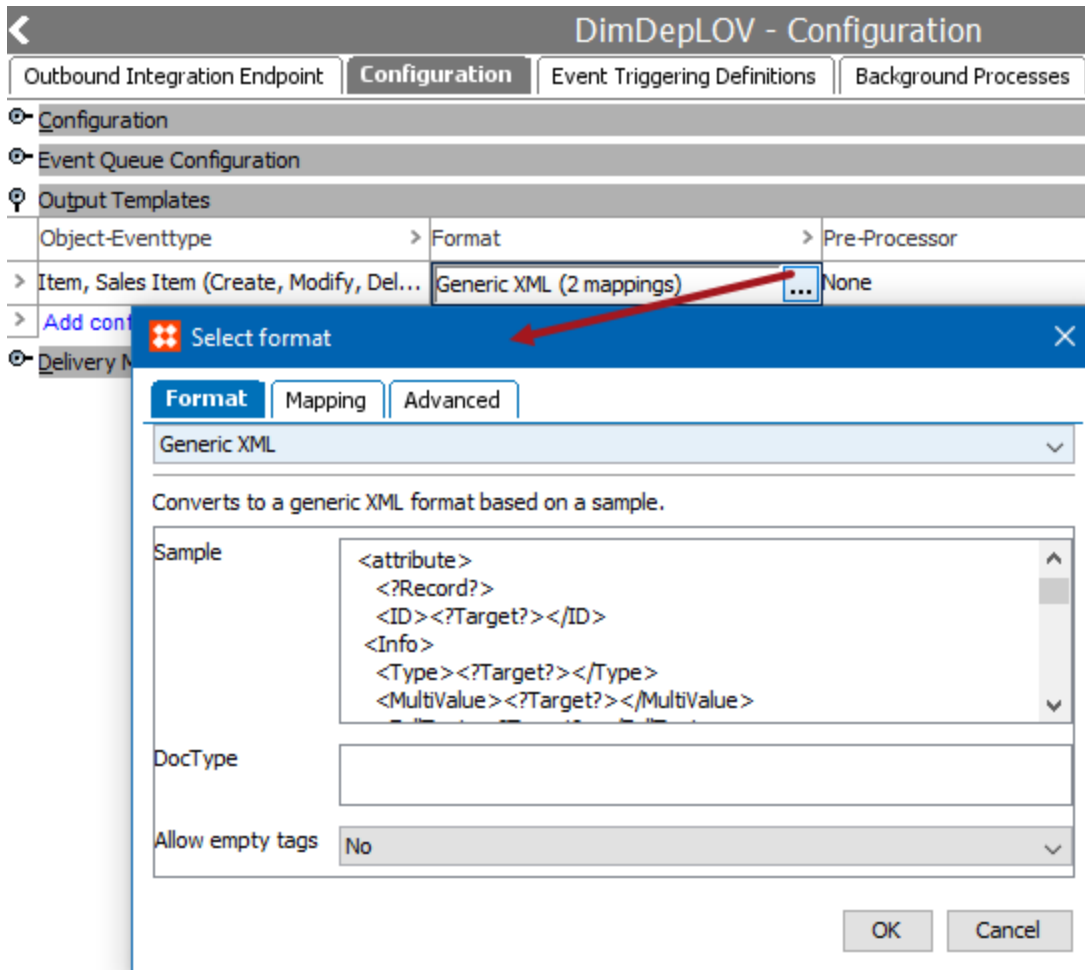
Back

Next

Finish

Cancel

OIEP



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

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.

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>
```

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. A repeated node may not be declared as a record node as well.

The following template could be used to extract the data from the above Generic XML:

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

FormatGeneric 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

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
< >												

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>
        </ProductInformation>
      </PrimarySpecs>
    </Item>
  </Items>
</ItemLoad>

```

```

        <Name>Battery</Name>
        <Value>C2</Value>
    </Variants>
    <Manuals>
        <Manual type="ManualDE">Manual-9</Manual>
        <Manual type="ManualEN">Manual-10</Manual>
    </Manuals>
</ProductInformation>
</PrimarySpecs>
<SyncProtocols>
    <Protocol>Acme</Protocol>
    <Protocol>Sony</Protocol>
    <Protocol>Sharp</Protocol>
</SyncProtocols>
</Item>
</Items>
</ItemLoad>

```

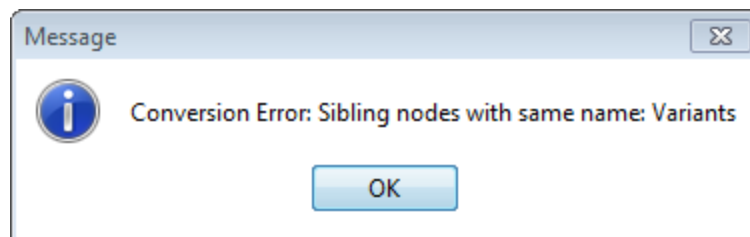
Creating a Template

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>

```



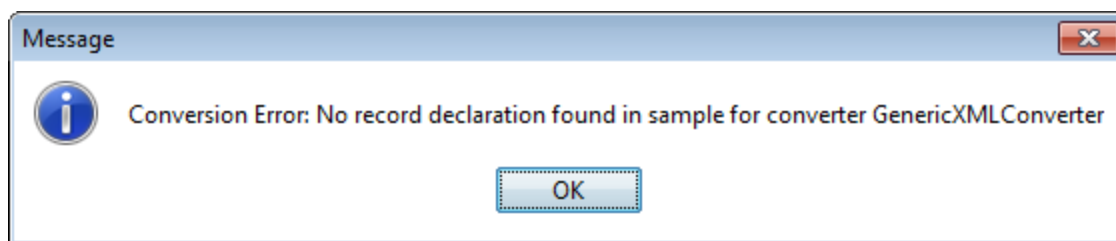
```

<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...	>
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...	>
2700524977488		AC-UZ444		Active 3D Glasses	
				AvailableFrom	>
				2015-01-01 00:00:00	
				Value	>
				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...	>
2700524977488		AC-UZ444		Active 3D Glasses	
				AvailableFrom	>
				2015-01-01 00:00:00	
				Color.Value	>
				Gray	
				Battery.Value	>
				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>
<SyncProtocols>
  <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-UZ444	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
- Deleting Assets with a Generic XML OIEP
- Deleting Classifications with a Generic XML OIEP
- Deleting Entities with a Generic XML OIEP
- Deleting Product References with a Generic XML OIEP
- 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
- 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 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 **RemoveIfEmpty Instruction in Generic XML** section.

Objects

The object selected for export includes attributes without values.

Product	Sub Products	References	Referenced By	Images & Documents	Commercial
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

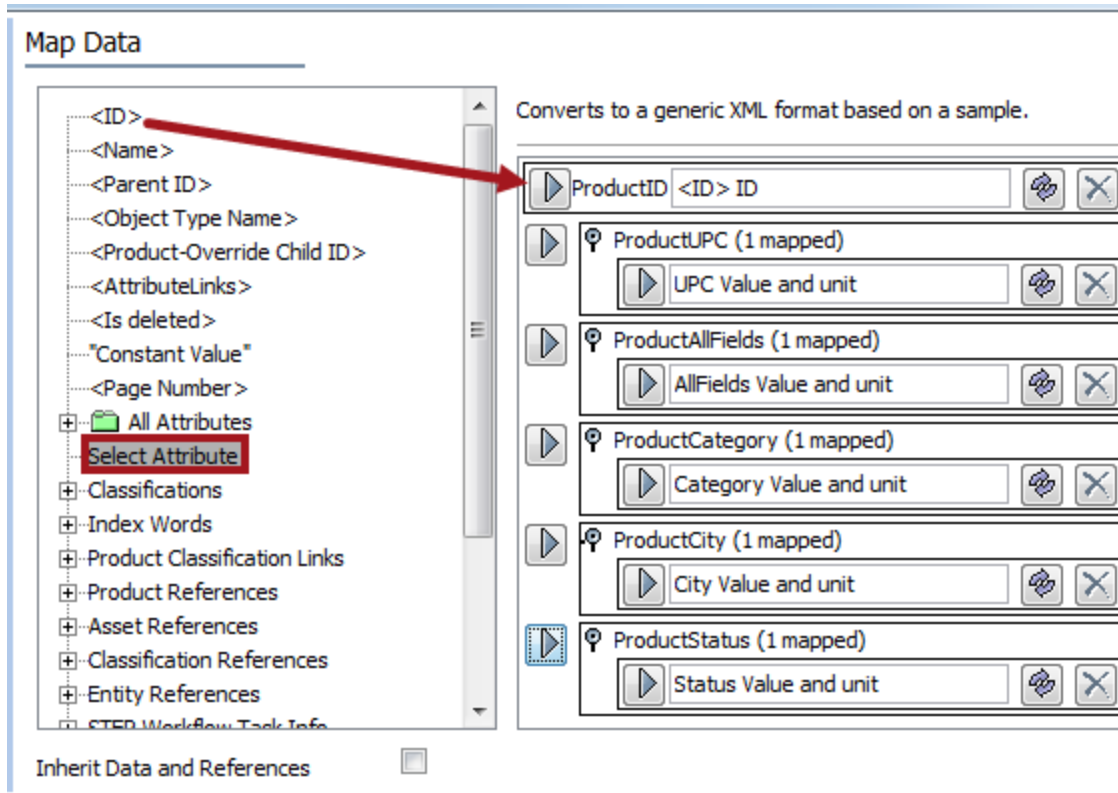
Generic XML

Converts to a generic XML format based on a sample.

Sample	<pre> <Products> <Product> <?Record?> <ProductID><?Target?></ProductID> <ProductUPC><?MultiTarget?></ProductUPC> <ProductAllFields><?MultiTarget?></ProductAllFields> <ProductCategory><?MultiTarget?></ProductCategory> <ProductCity><?MultiTarget?></ProductCity> <ProductStatus><?MultiTarget?></ProductStatus> </Product> </Products> </pre>
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>
```

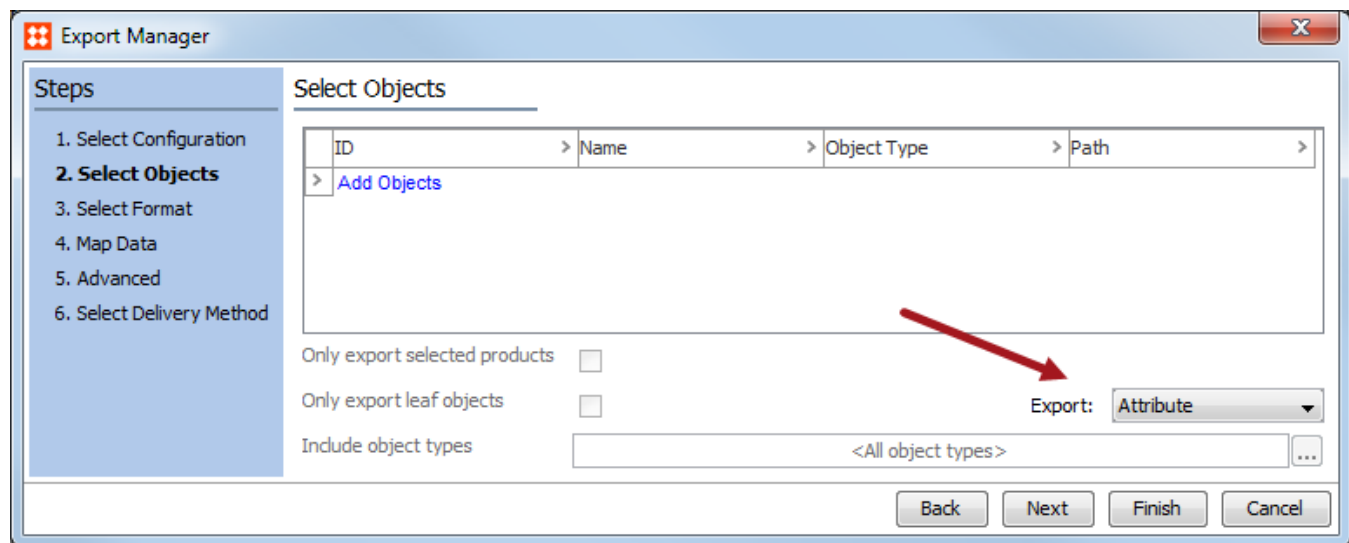
Attributes Instruction in Generic XML

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.

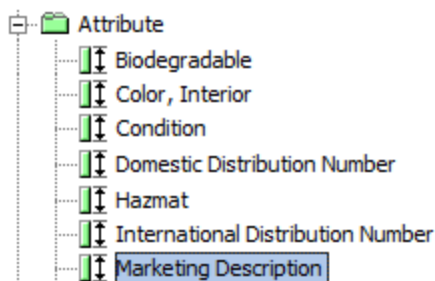
Follow the steps below to extract all available attribute data. If some data points on attributes are not required, they can be removed from the template and the mapping.

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:

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 Validation Info

Attribute Group References

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.

Aspect	Description
	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 Validation Minimum value	Extracts the minimum value for the attribute, if available. 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/>).

Aspect	Description
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 **Export Manager - Map Data** section of **Exchanging Data** documentation.

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.

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 ☐

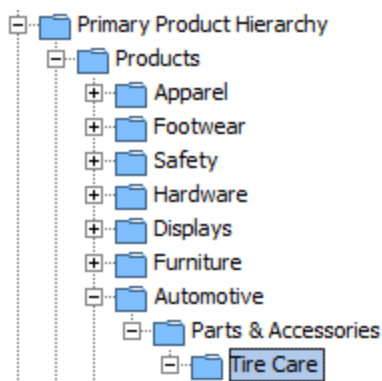
Only export leaf objects ☐

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>
```

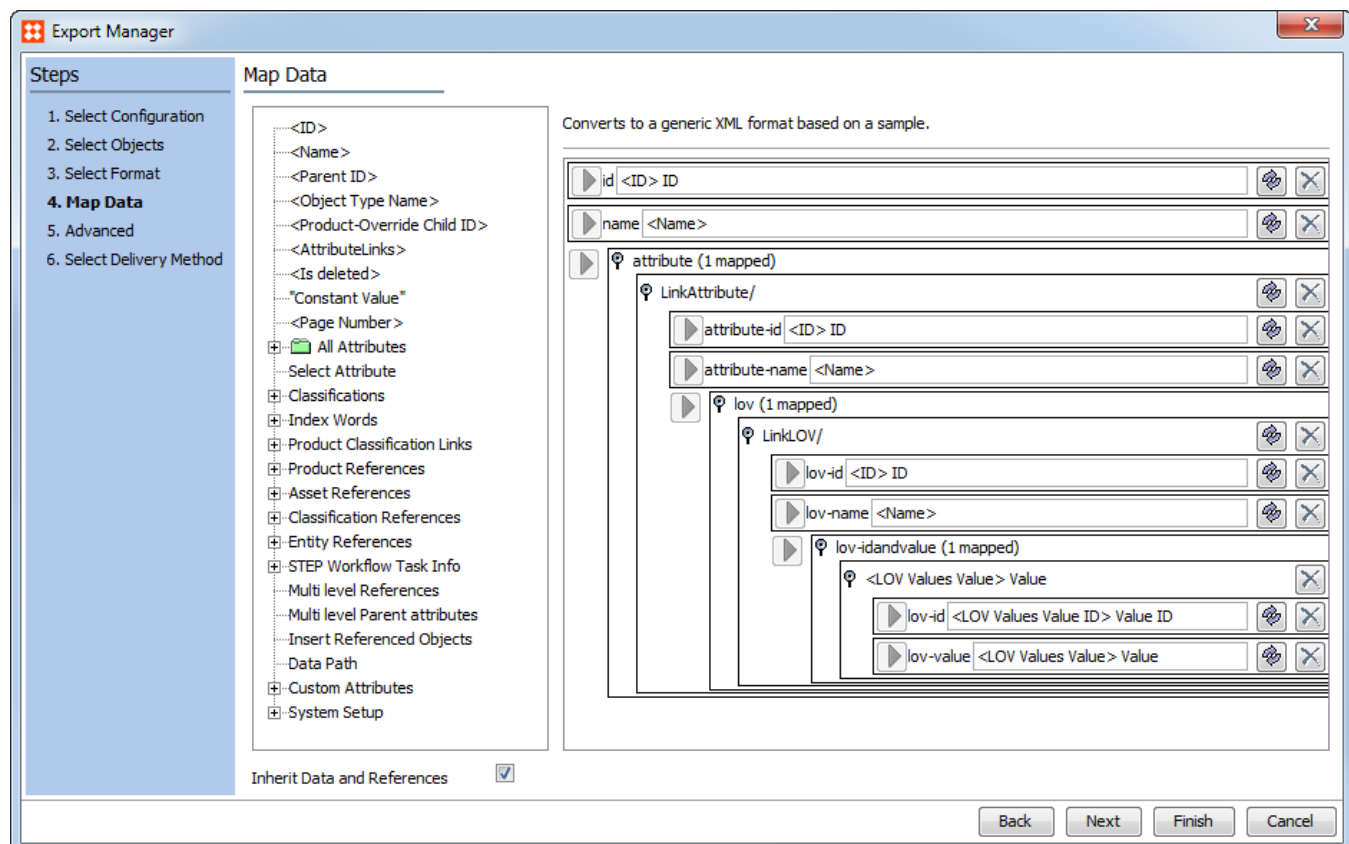
```

<attribute>
  <?MultiTarget?>
  <id><?Target attribute-id?></id>
  <name><?Target attribute-name?></name>
</attribute>
<lov>
  <?MultiTarget?>
  <id><?Target lov-id?></id>
  <name><?Target lov-name?></name>
</lov>
<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

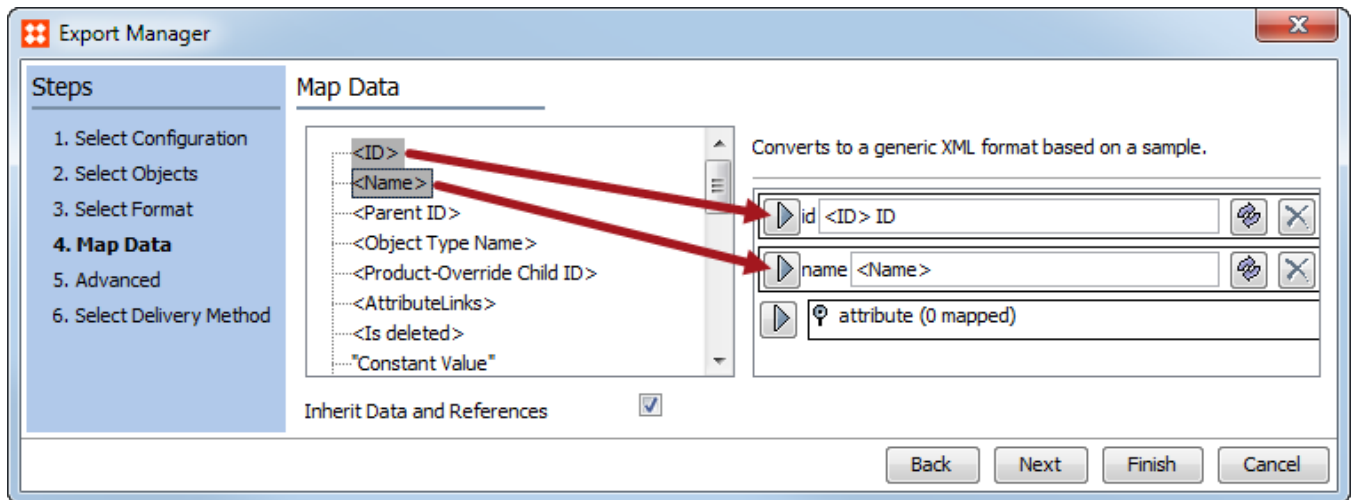
The Product, its linked Attributes, and any LOVs (and the values) used by the linked attributes are exported.

```
<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>
```

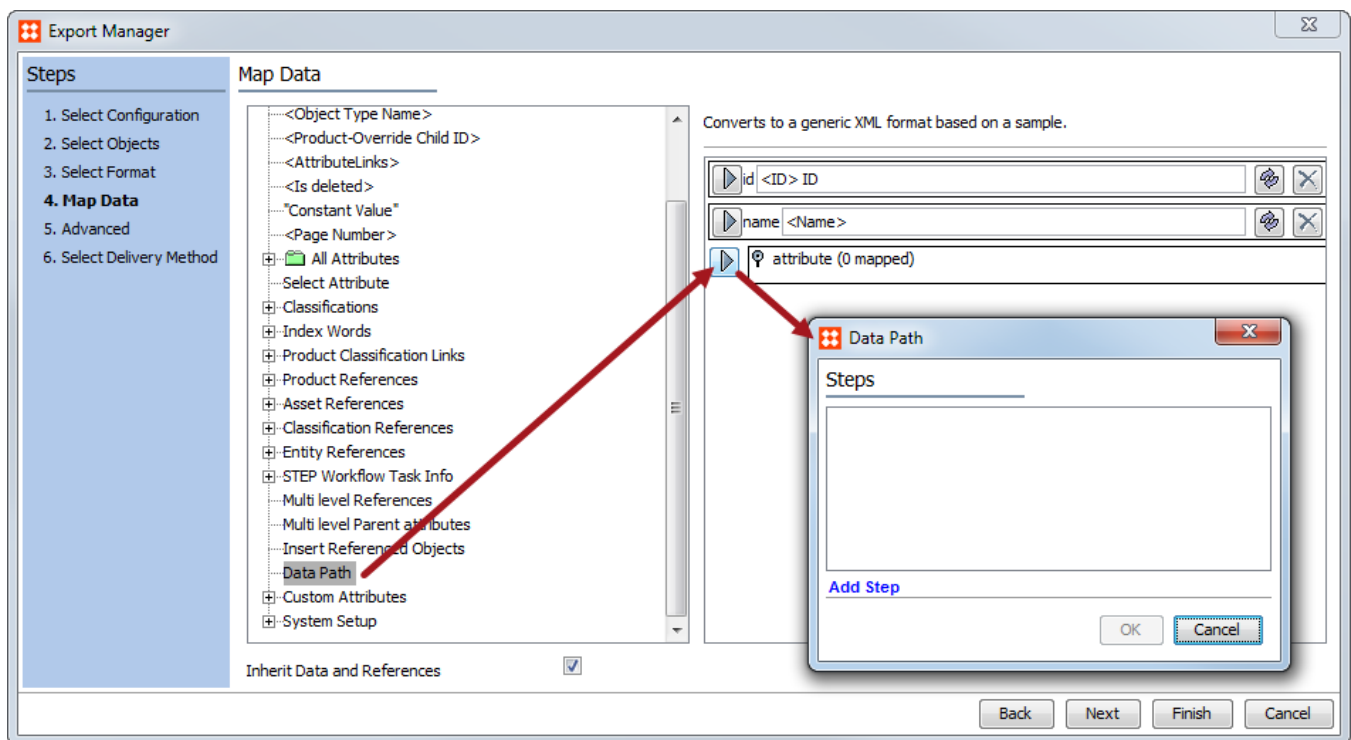
Mapping Attribute Links, LOVs, and LOV Values in Generic XML

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.

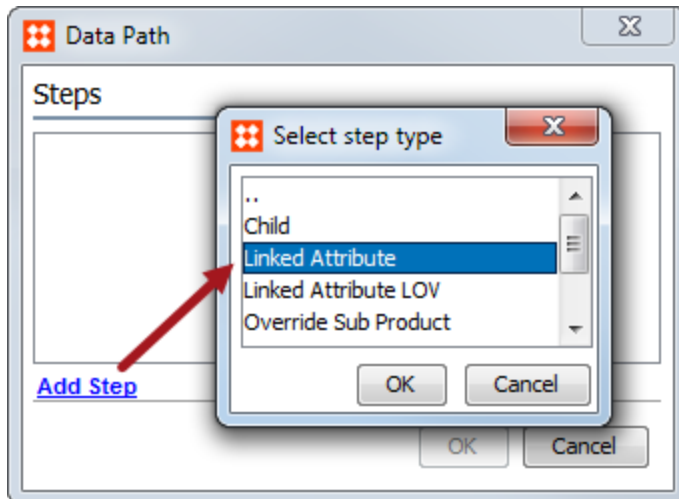
1. Map **ID** and **Name** for the Product.



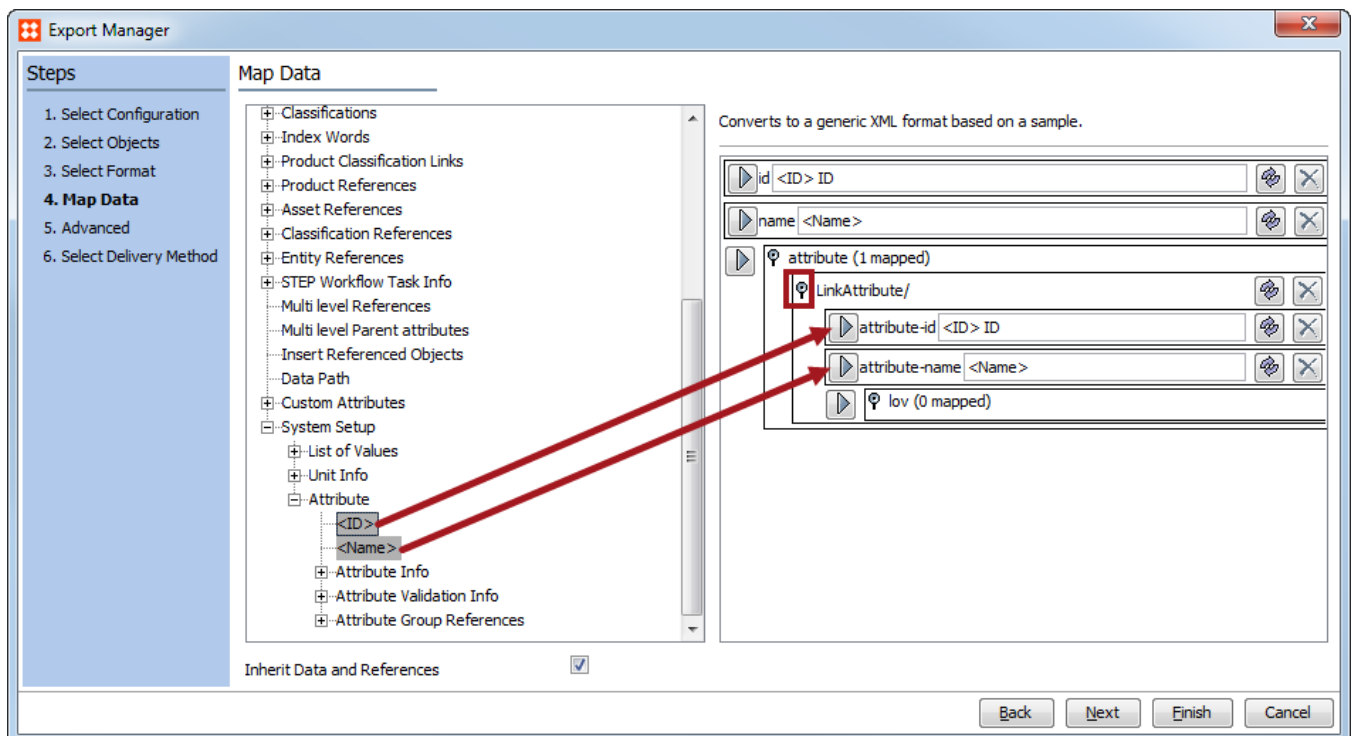
2. Map **Data Path** to attribute.



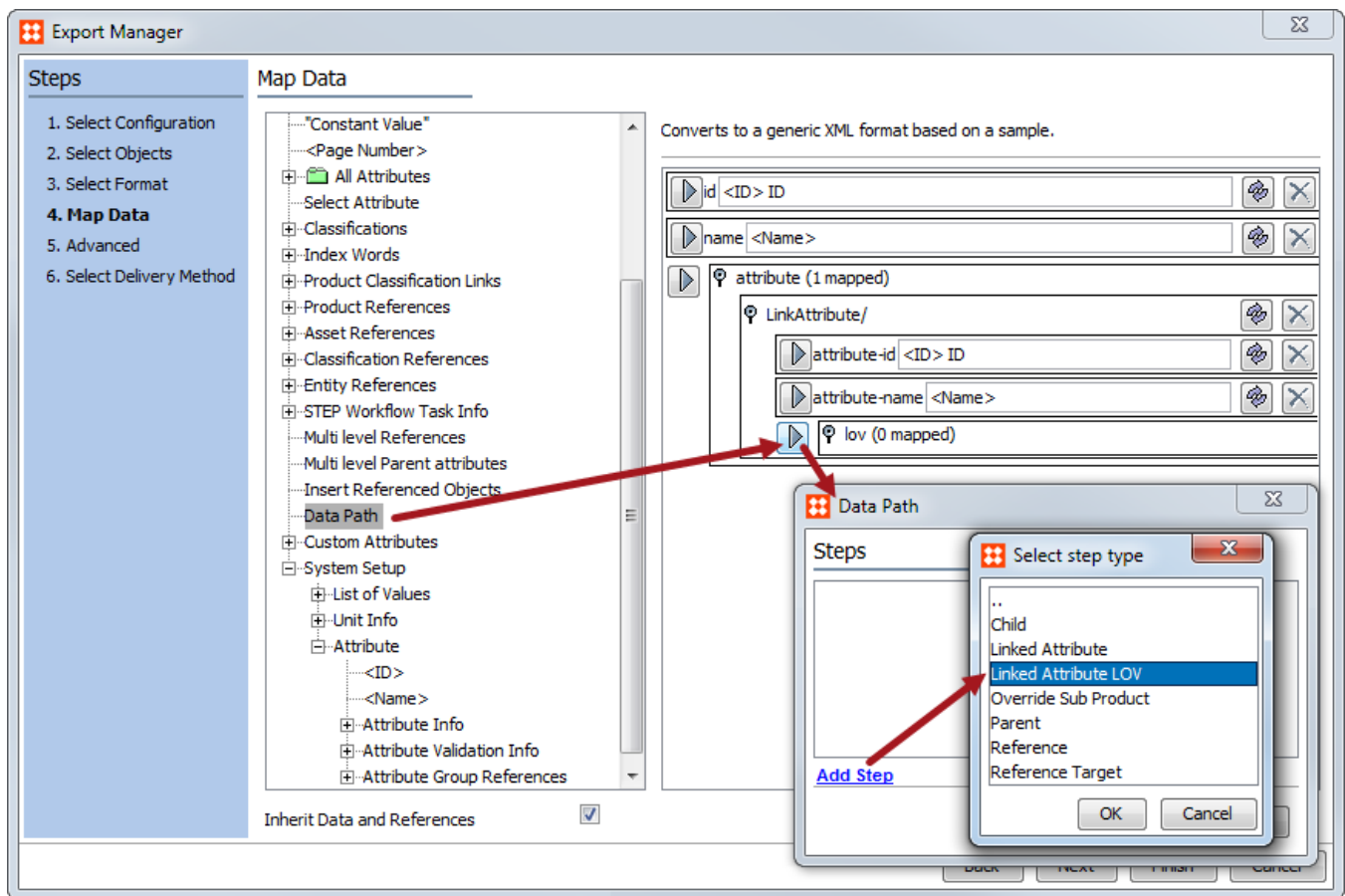
3. Click the **Add Step** link and select **Linked Attribute**.



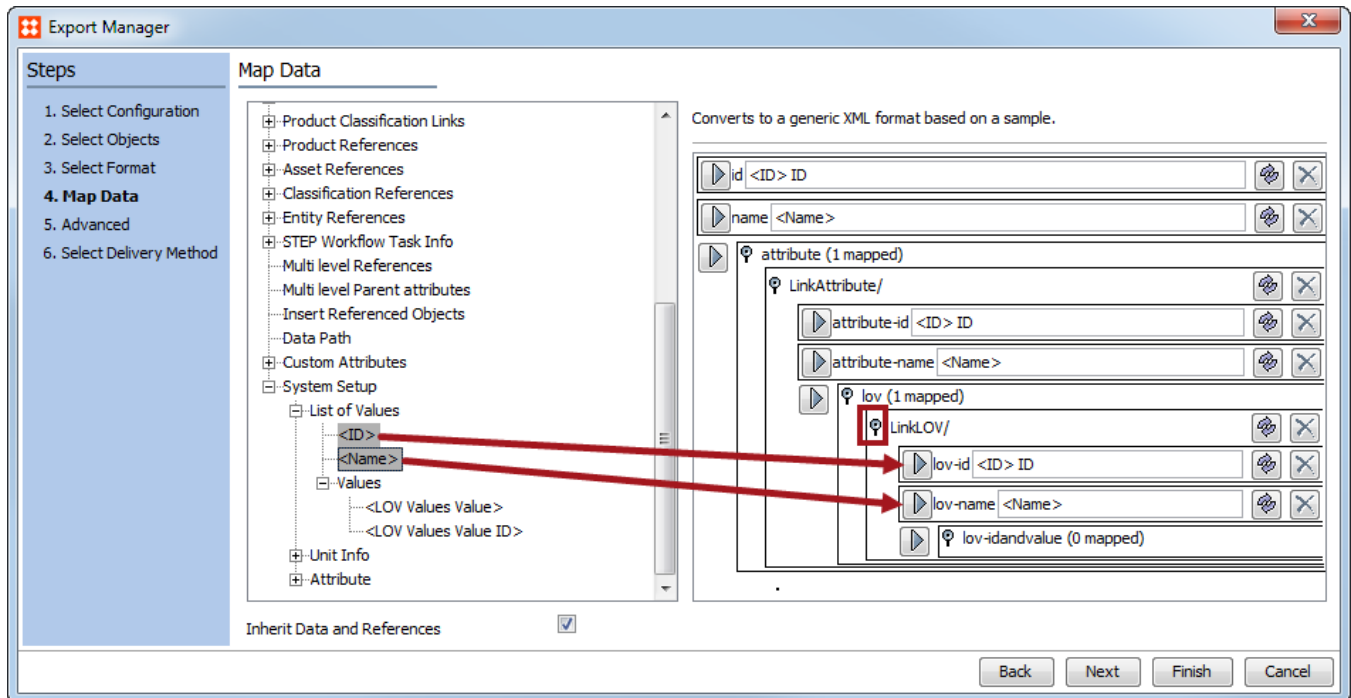
4. Click OK to close the **Select step type** and **Data Path** dialogs.
5. Open the LinkAttribute/ flipper and map Attribute **ID** and **Name**.



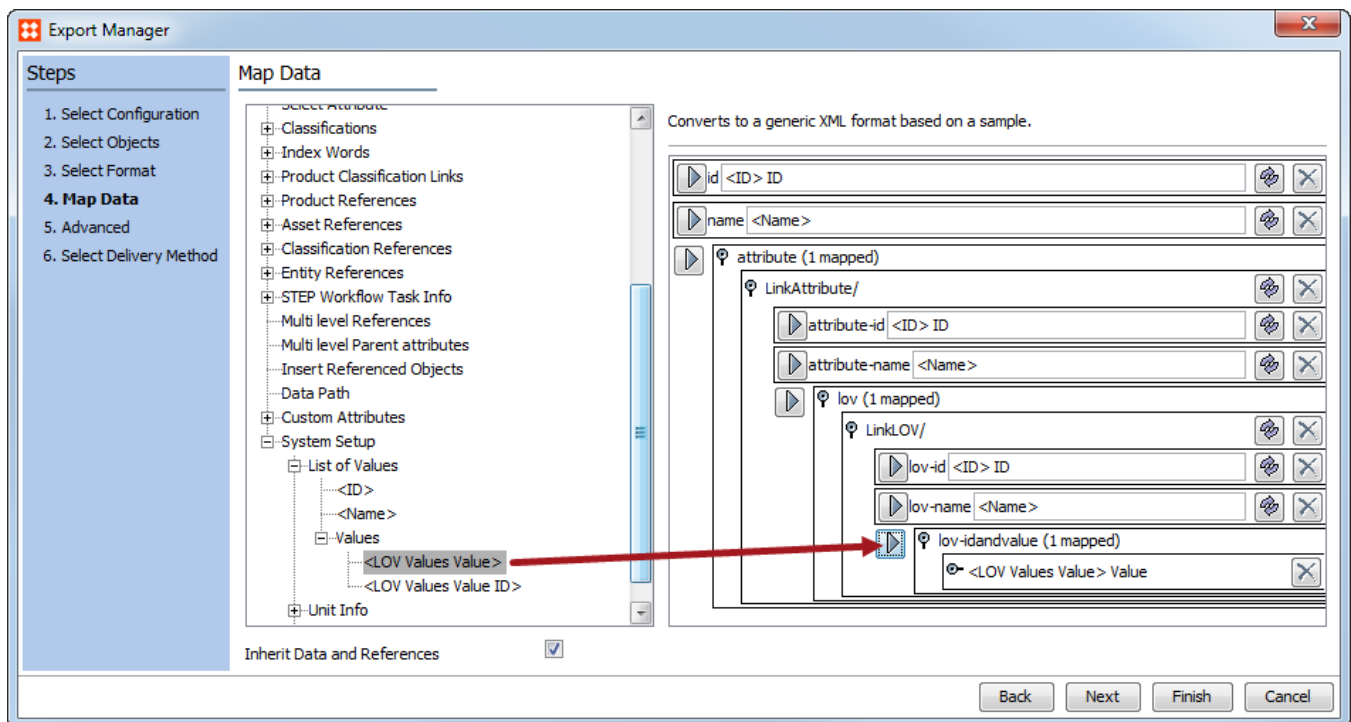
6. Map **Data Path** to lov, click **Add Step** and select **Linked Attribute LOV**.



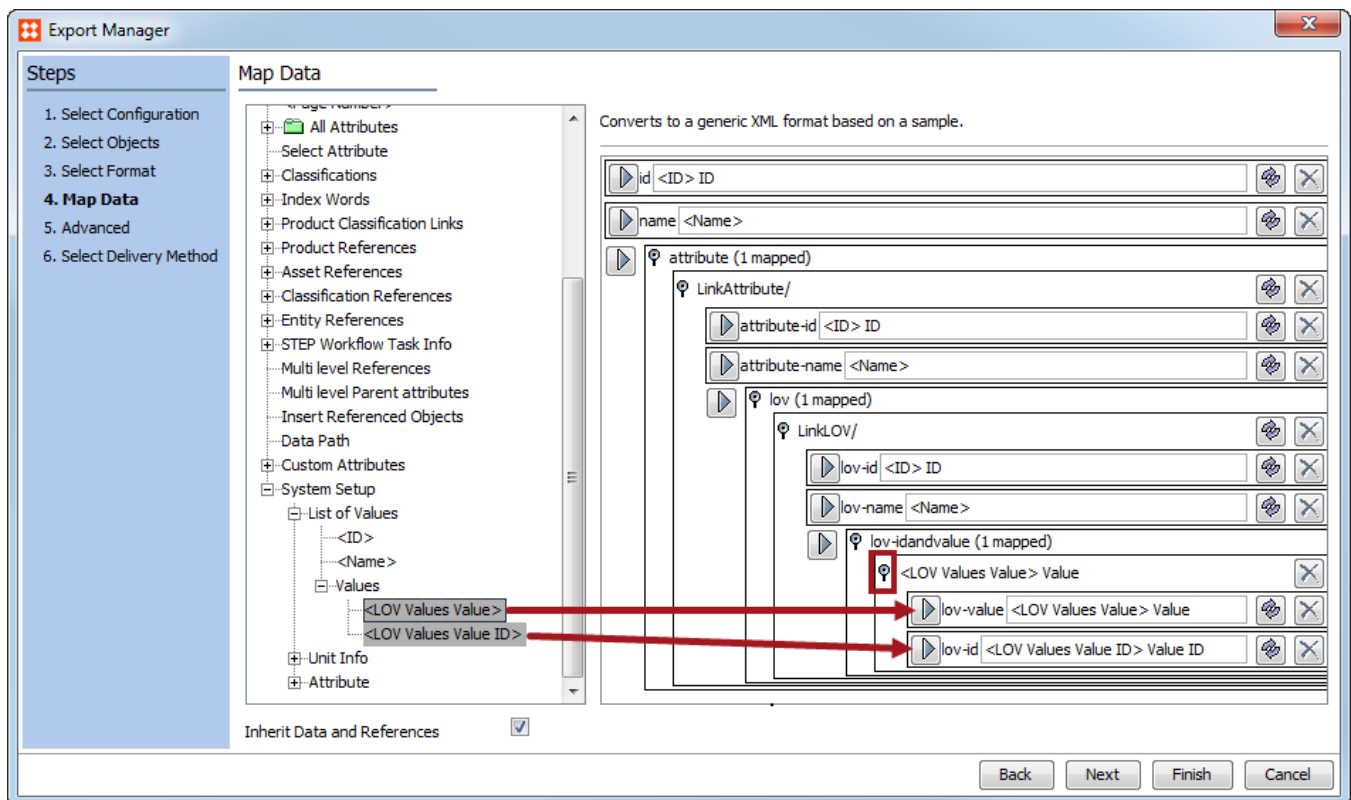
7. Click OK to close the **Select step type** and **Data Path** dialogs.
8. Open the LinkLOV/ flipper and map Attribute **ID** and **Name**.



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:

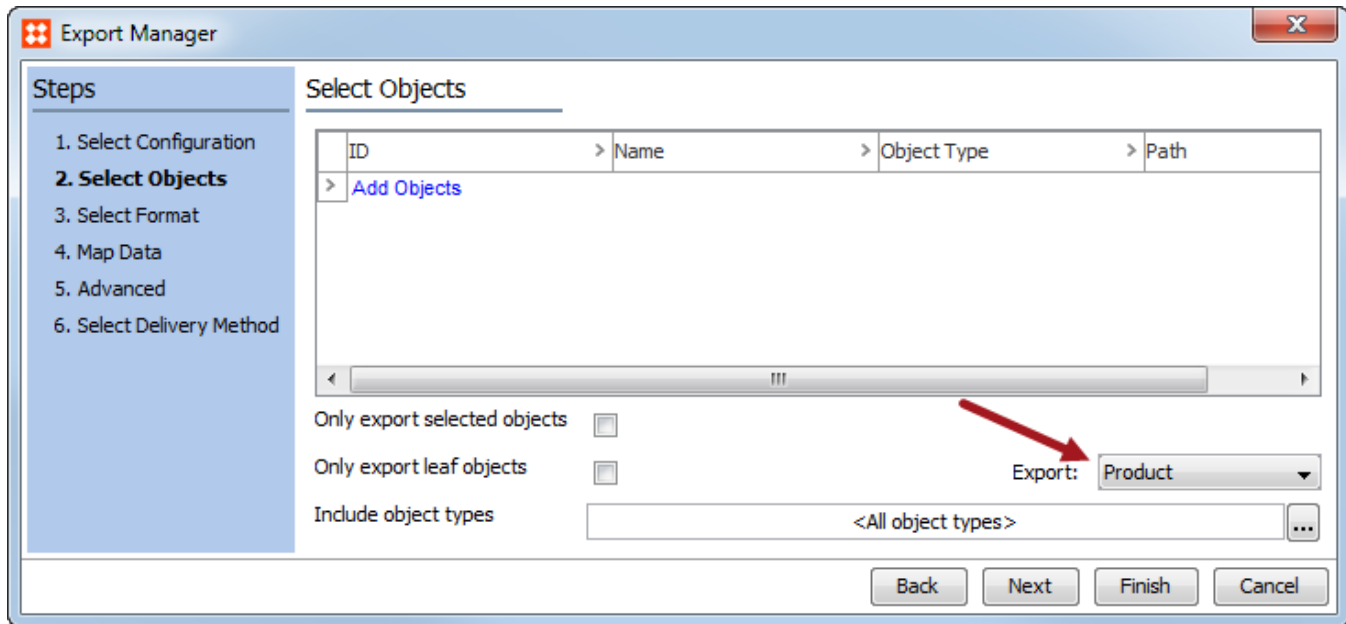
- the ID and Name of the Product
 - the ID and Name of the linked attributes on the Product
 - the ID, Name, and Base of the Unit on any of the linked attributes

LOVs for linked attributes can also be mapped and exported as described in **Attribute Links and LOVs Instruction in Generic XML**.

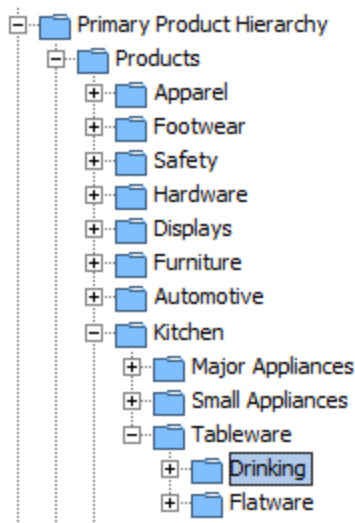
The data path data source is also available using the IDoc MATMAS 05 format.

Objects

Choose **Product** as object type to export.



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>
  </product>
</products>
```

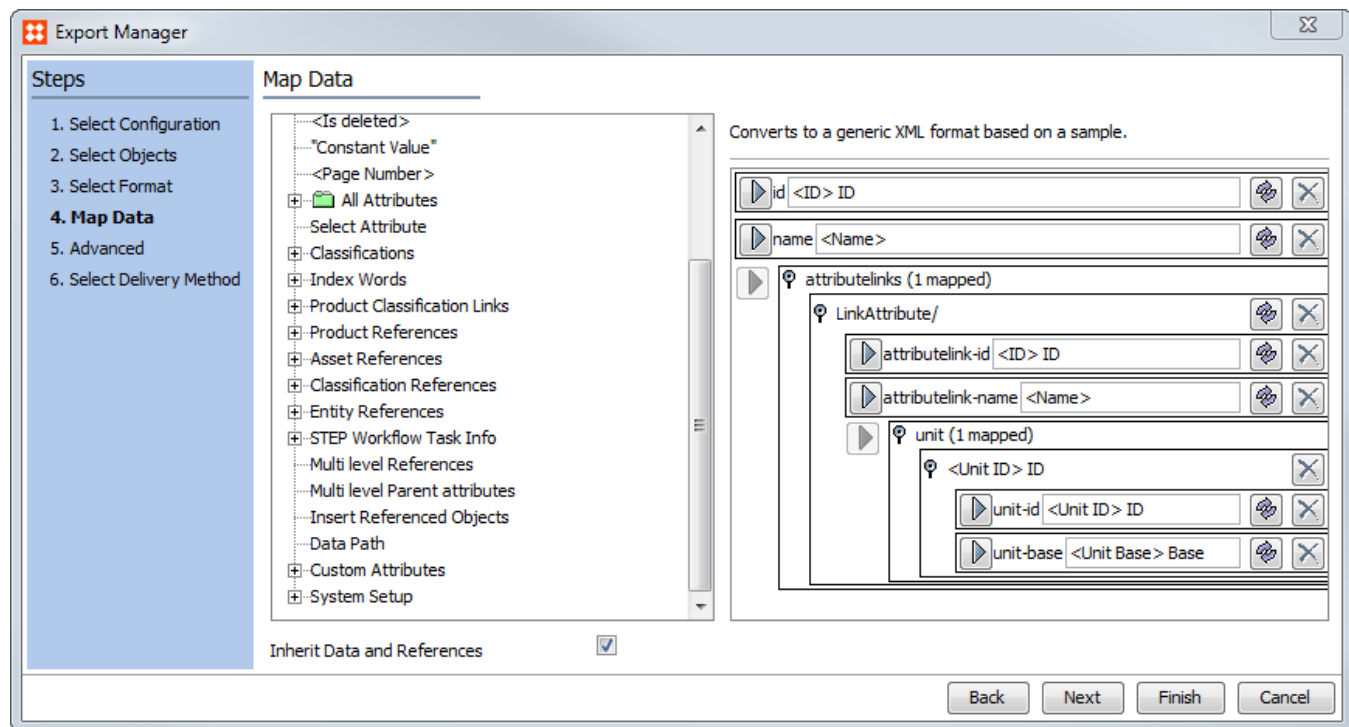
```

<?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.

```

<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>

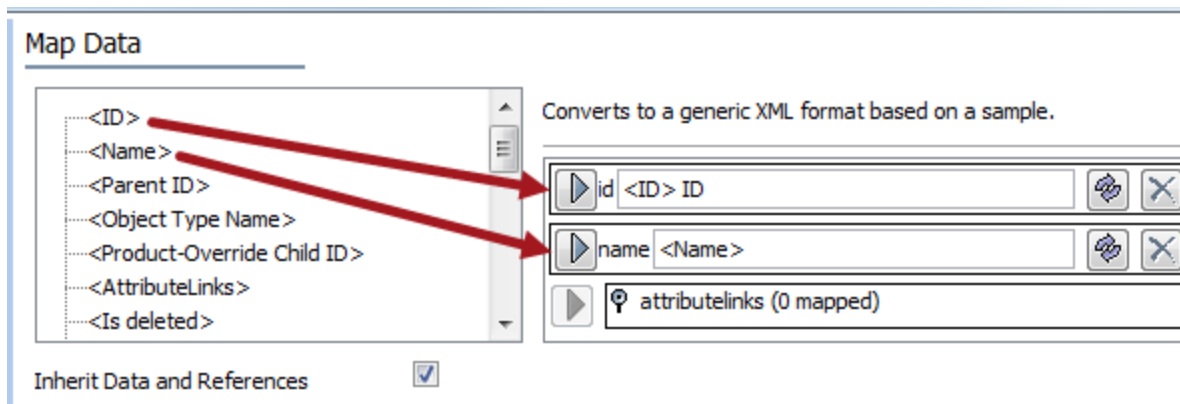
```

Mapping Attribute Links, Units, and Unit Base in Generic XML

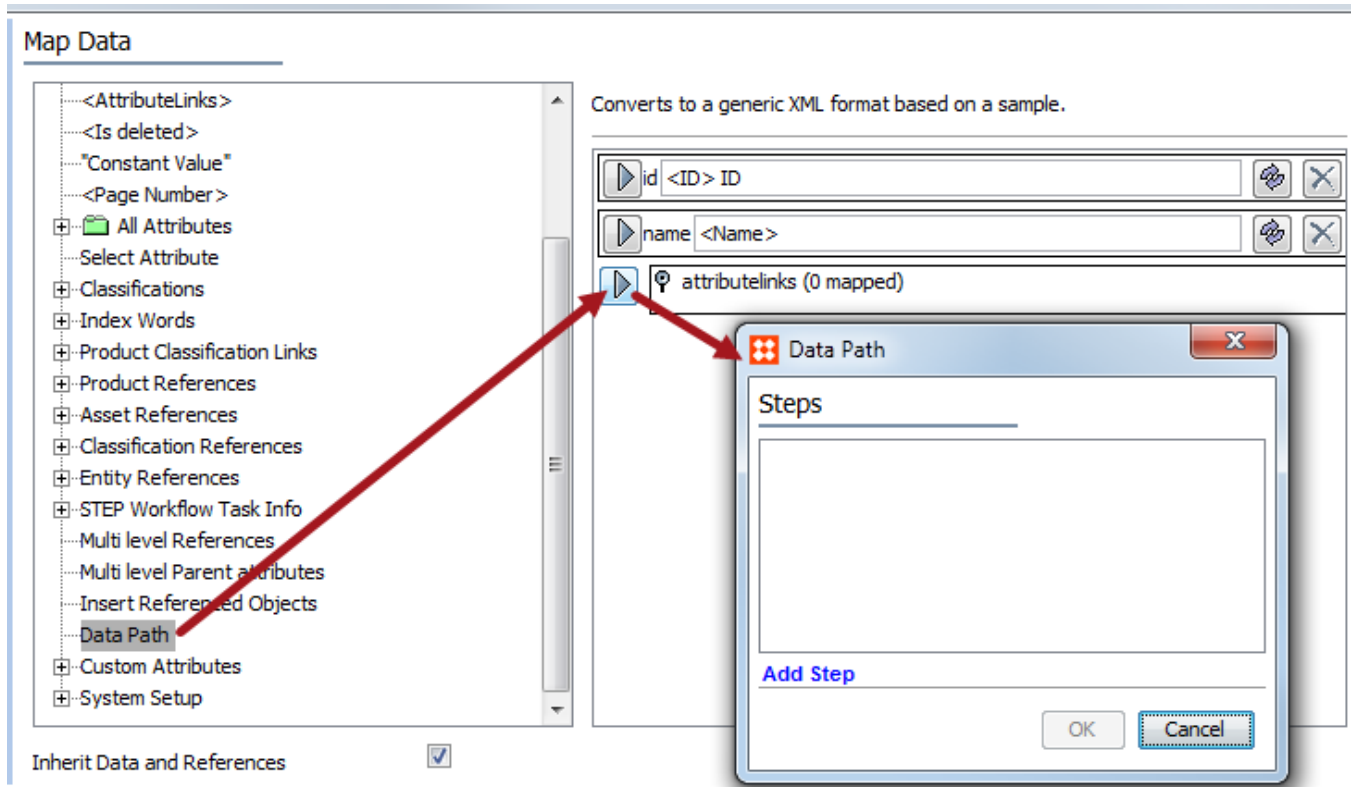
The **Attribute Links and Units Instruction in Generic XML** section discusses additional setup for a Generic XML export. Use the following steps to map the Product, linked attributes, and their Unit information.

The data path data source is also available using the IDoc MATMAS 05 format.

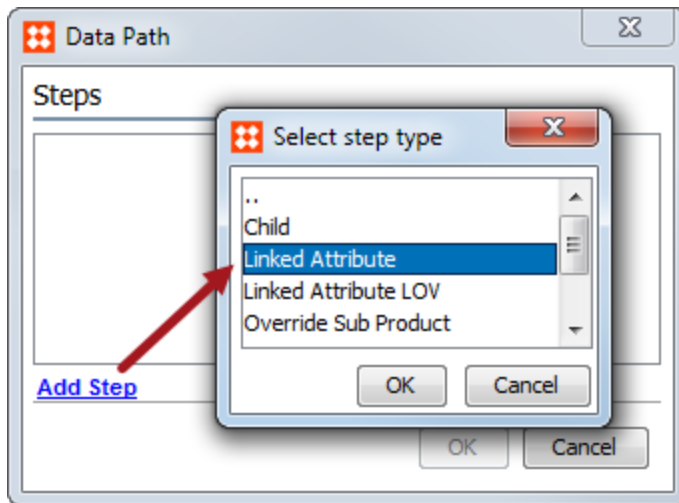
1. Map **ID** and **Name** for the Product.



2. Map **Data Path** to attributelinks.



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**.

Map Data

Converts to a generic XML format based on a sample.

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

Inherit Data and References ☒

id <ID> ID

name <Name>

attributelinks (1 mapped)

- LinkAttribute/
 - attributelink-id <ID> ID
 - attributelink-name <Name>
 - unit (0 mapped)

6. Map **Unit ID** to unit.

Map Data

Converts to a generic XML format based on a sample.

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

Inherit Data and References ☒

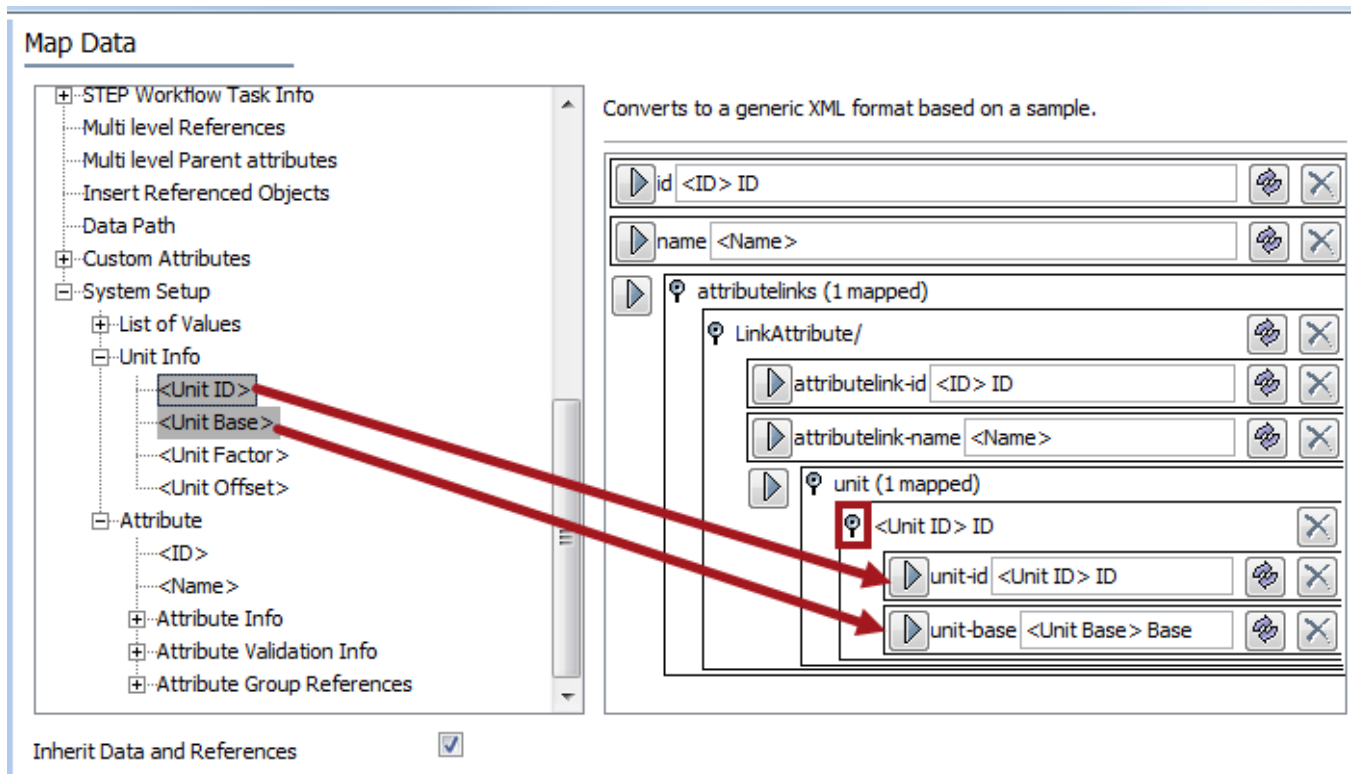
id <ID> ID

name <Name>

attributelinks (1 mapped)

- LinkAttribute/
 - attributelink-id <ID> ID
 - attributelink-name <Name>
 - unit (1 mapped)
 - <Unit ID> ID

7. Open the <Unit ID> flipper and map Unit **ID** and **Base**.

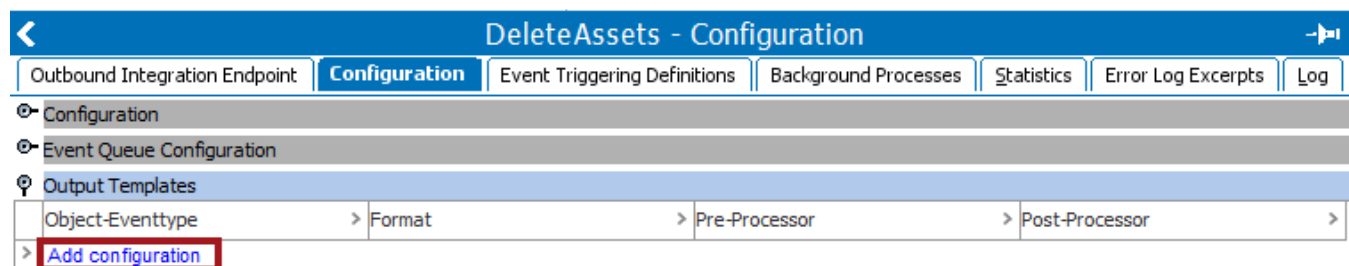


Deleting Assets with a Generic XML OIEP

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.

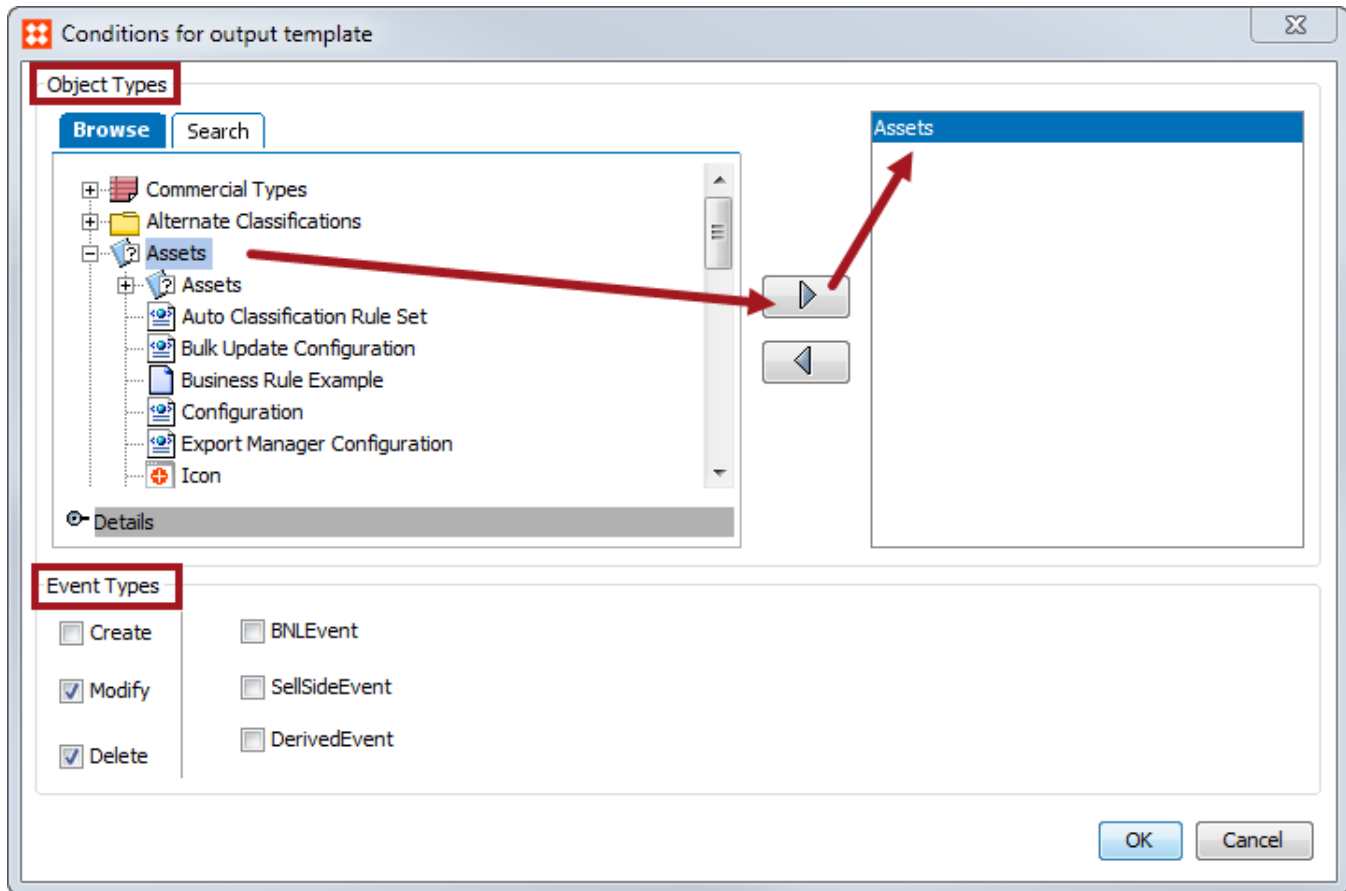
Object Type and Event Type Selection

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.



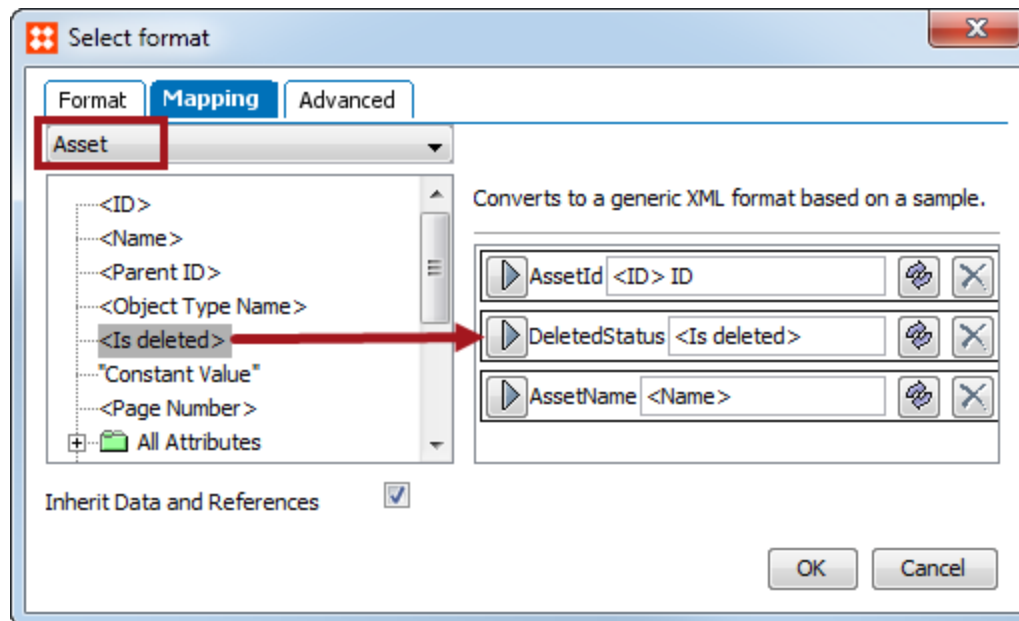
Template

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.

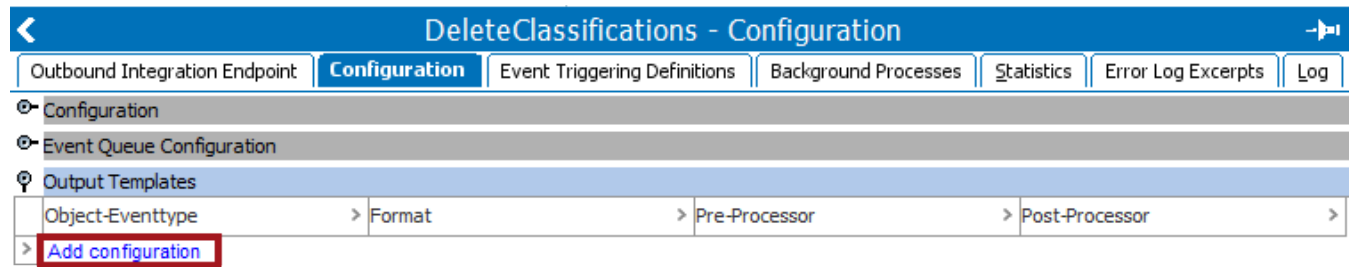
For modified Assets, the Deleted Status displays " ", which indicates the asset is not deleted. All mapped fields are exported for Create and Modify events.

Deleting 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.

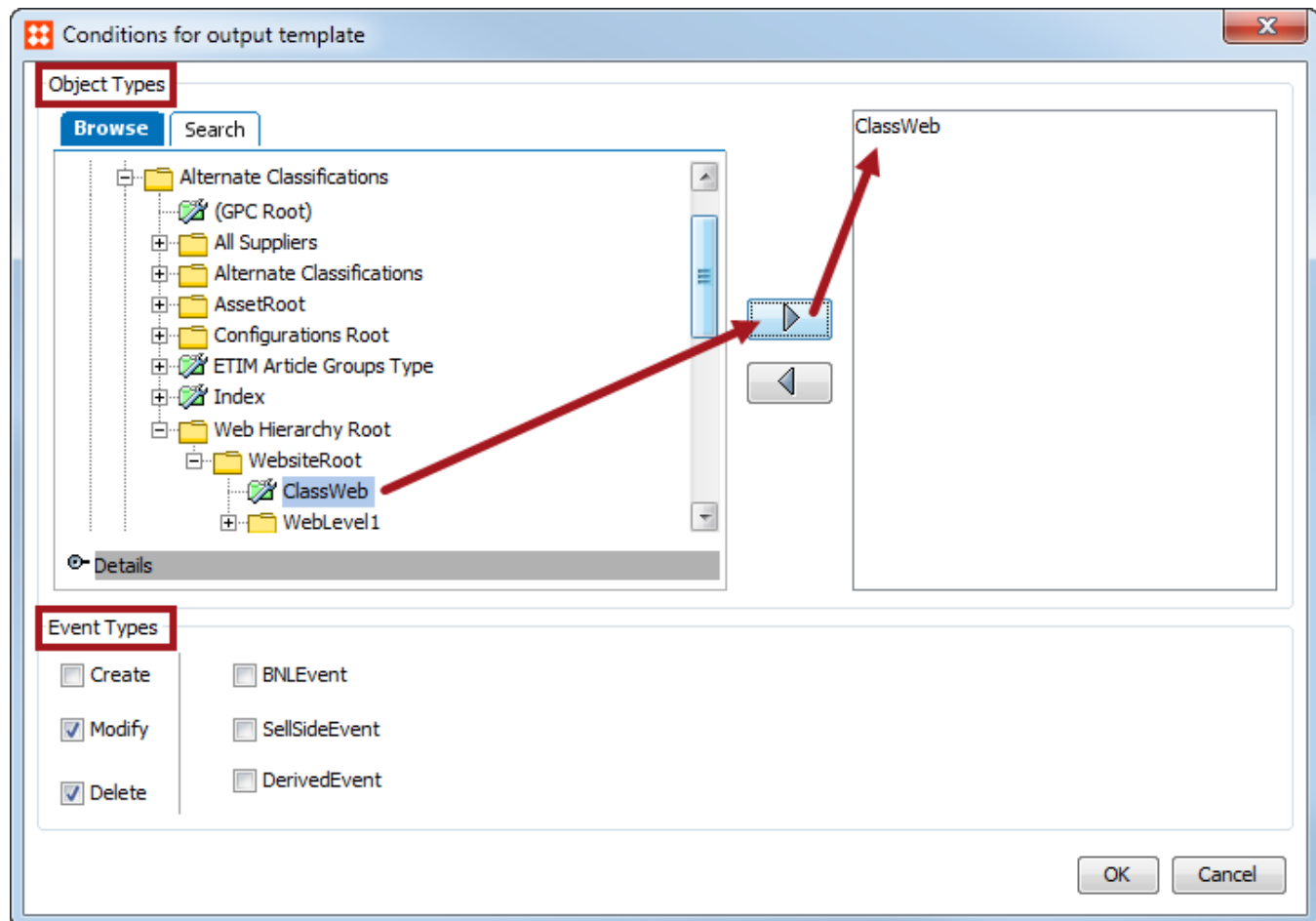
Object Type and Event Type Selection

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.

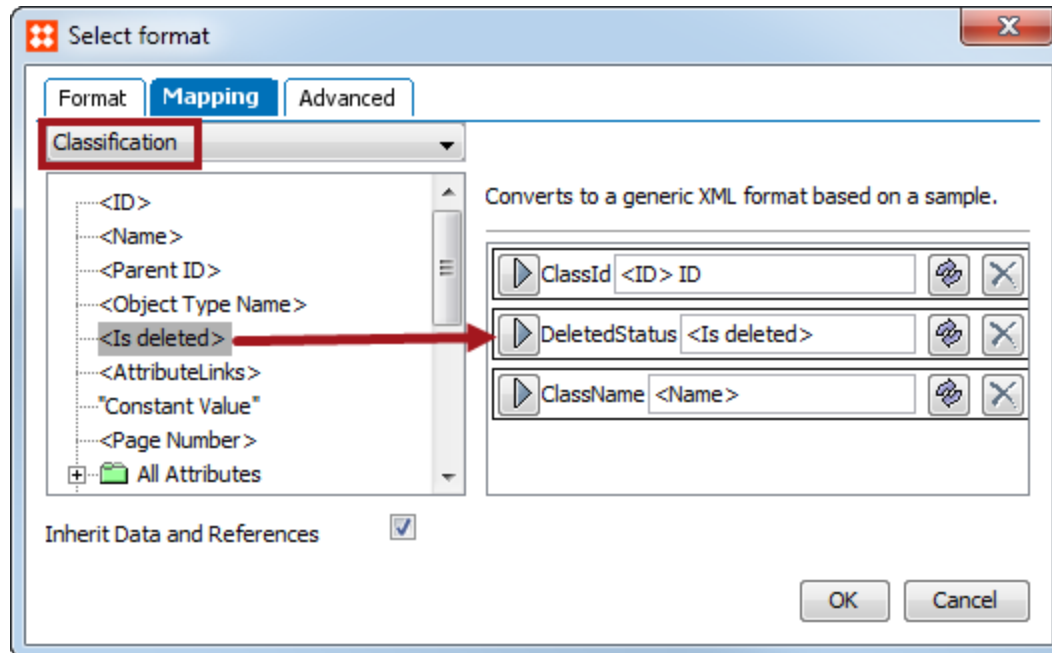
```
<root>
```

```

<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.

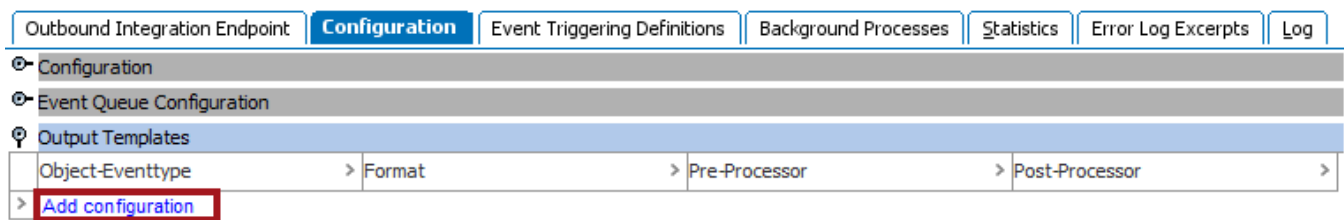
For modified Classifications, the Deleted Status displays **" "**, which indicates the classification is not deleted. All mapped fields are exported for Create and Modify events.

Deleting 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.

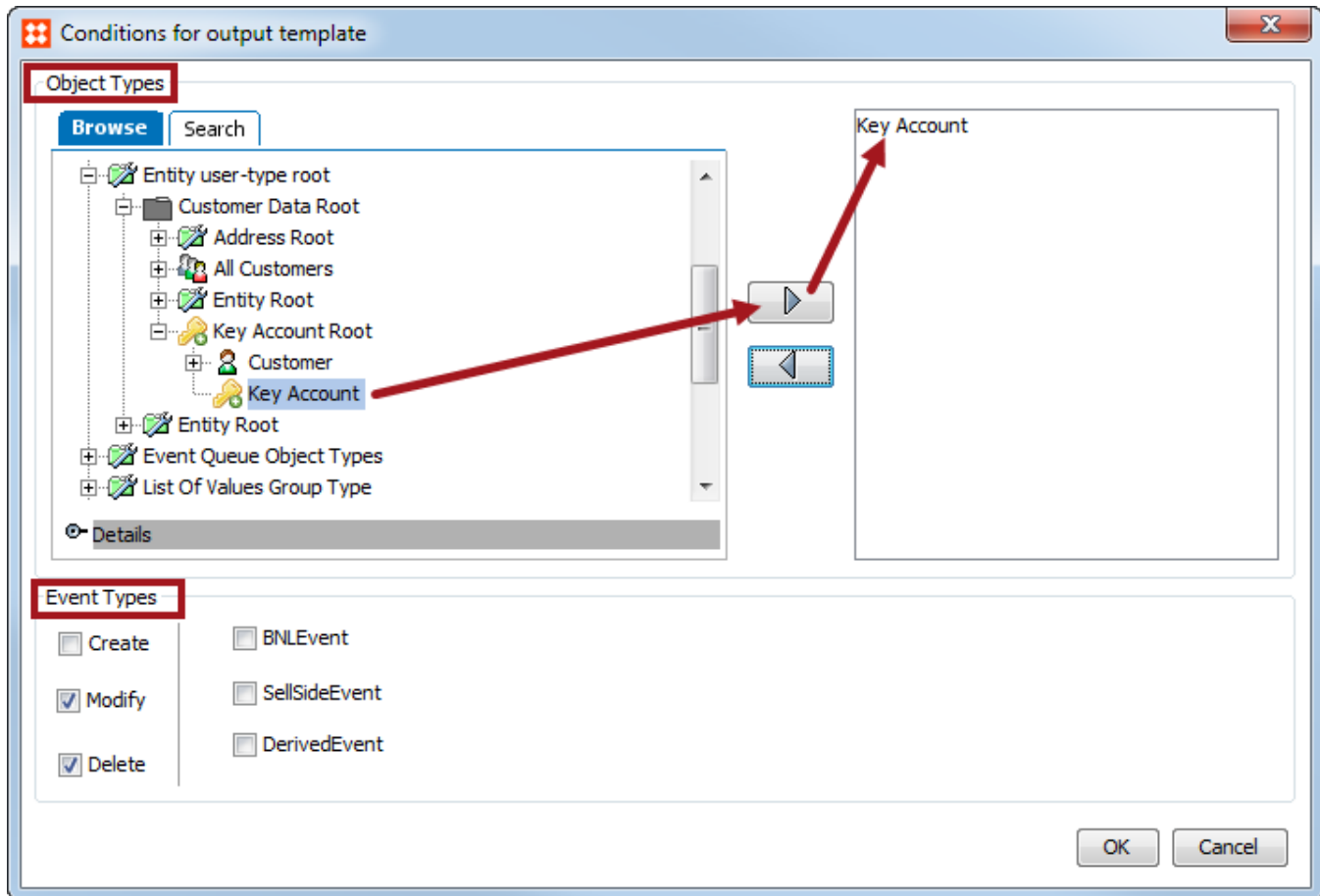
Object Type and Event Type Selection

On the OIEP, open the Configuration tab. Under the Output Templates flipper, click the **Add configuration** link.



Set the Object Types and Event Types.

For this example, Modify events are also included to demonstrate the available functionality.



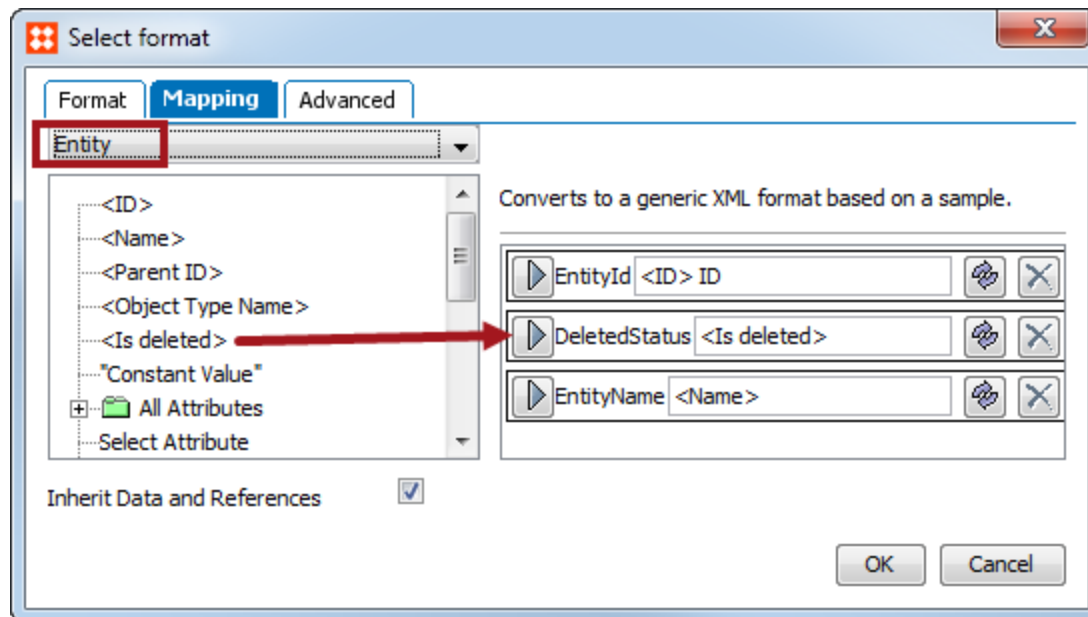
Template

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.

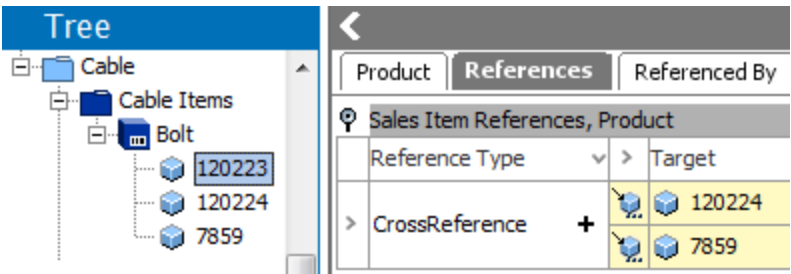
For modified Entities, the Deleted Status displays **" "**, which indicates the entity is not deleted. All mapped fields are exported for Create and Modify events.

Deleting 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.

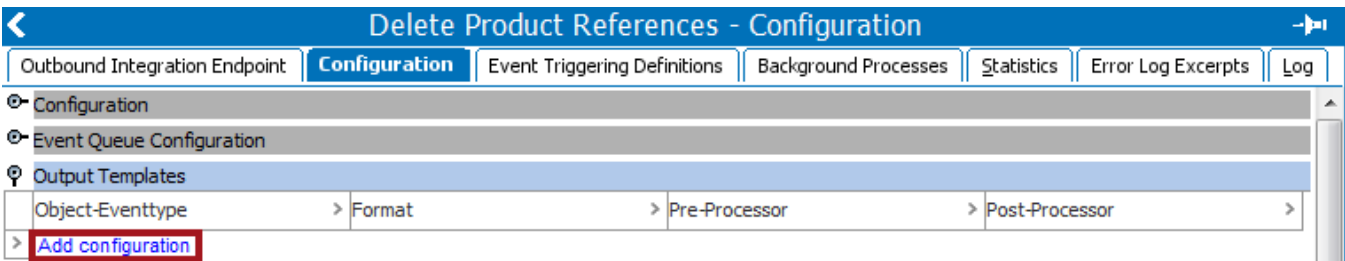
Product

On product 120223, the Reference Type of CrossReference to Target 7859 will be deleted in order to trigger the OIEP.

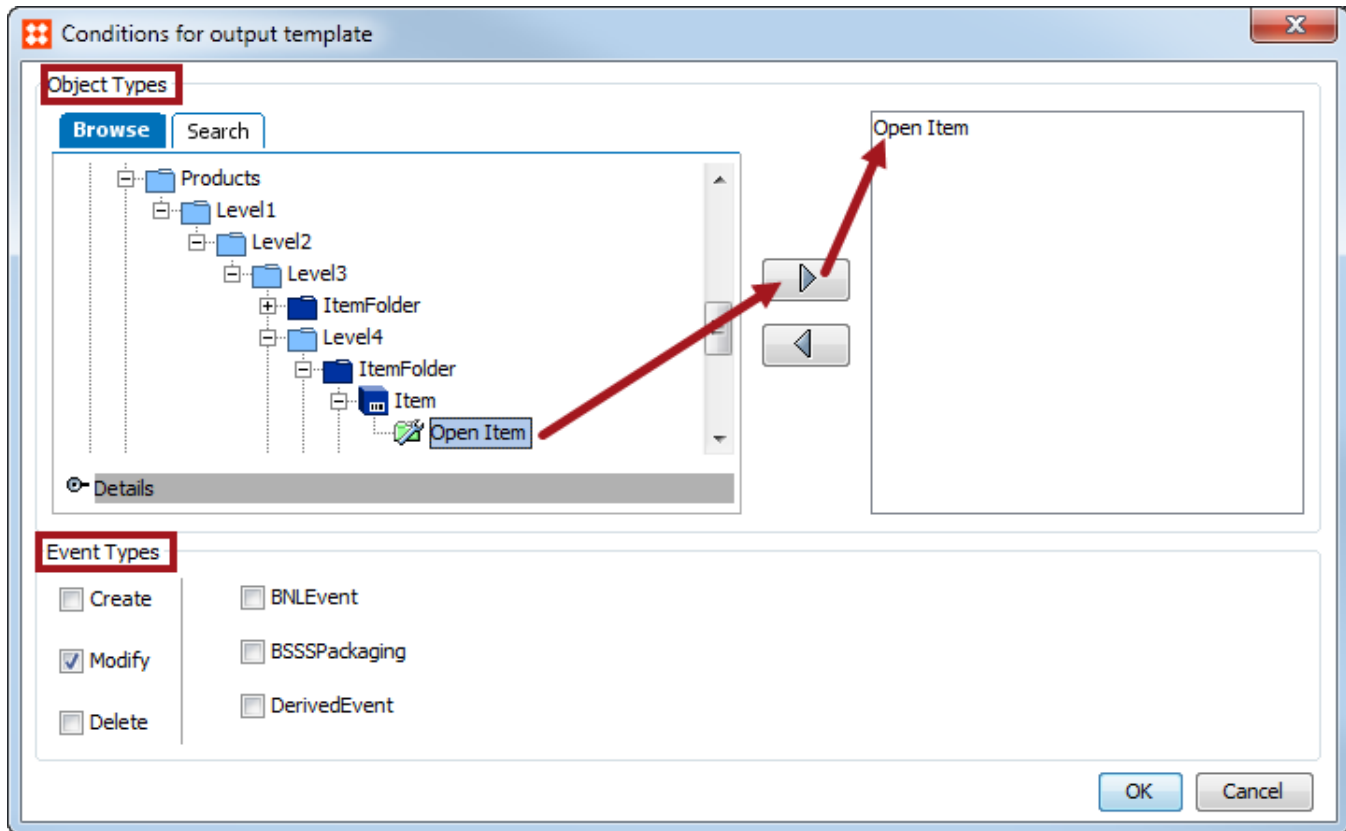


Object Type and Event Type Selection

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

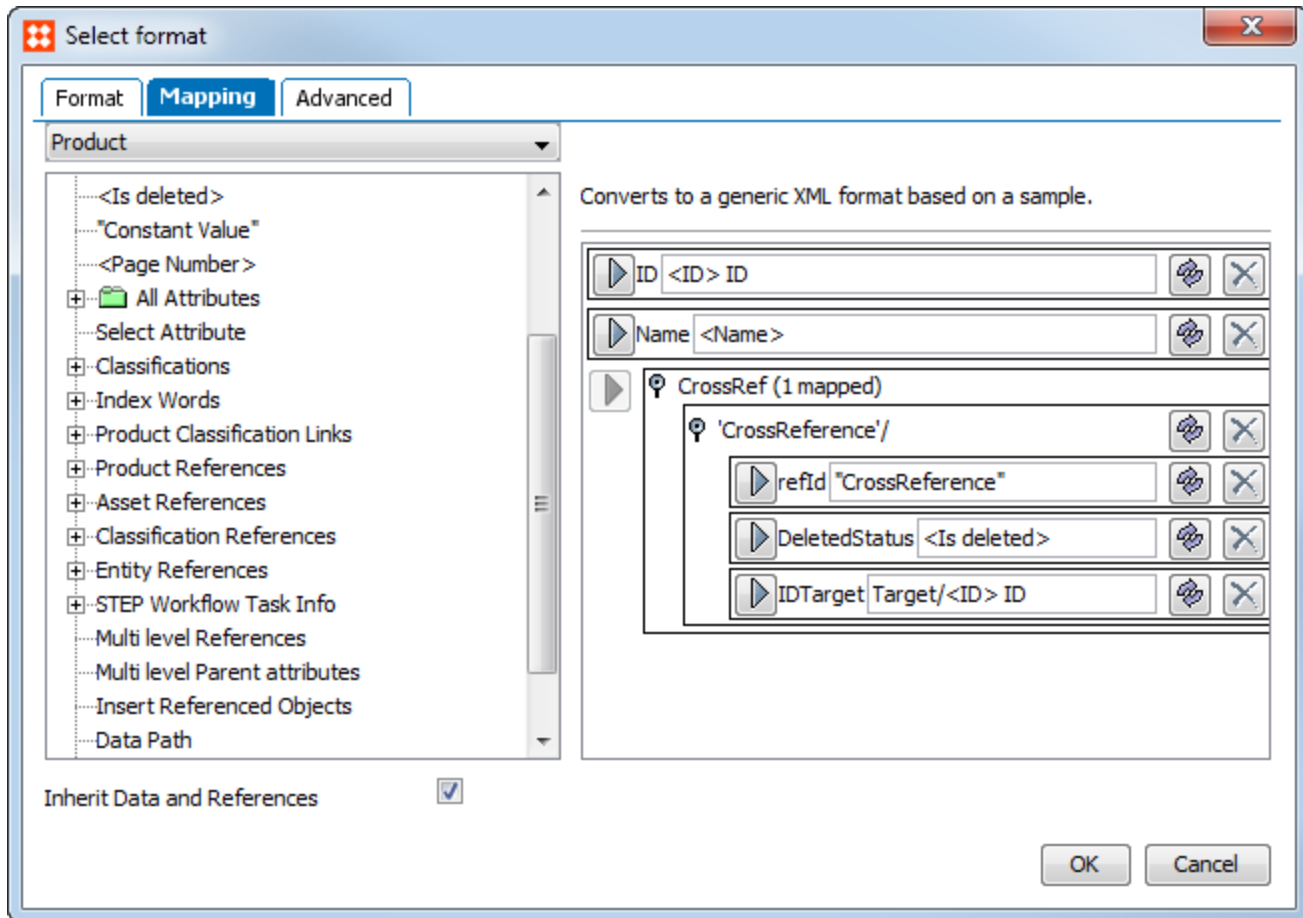
In the Output Template, click the ellipsis button (...), and set the format to **Generic XML**.

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 Delete Product References with Generic XML OIEP** documentation.

Results

When the Product Reference for item 7859 is deleted and the product 120223 is approved, the OEIP is triggered. Invoking the OIEP generates the following file:

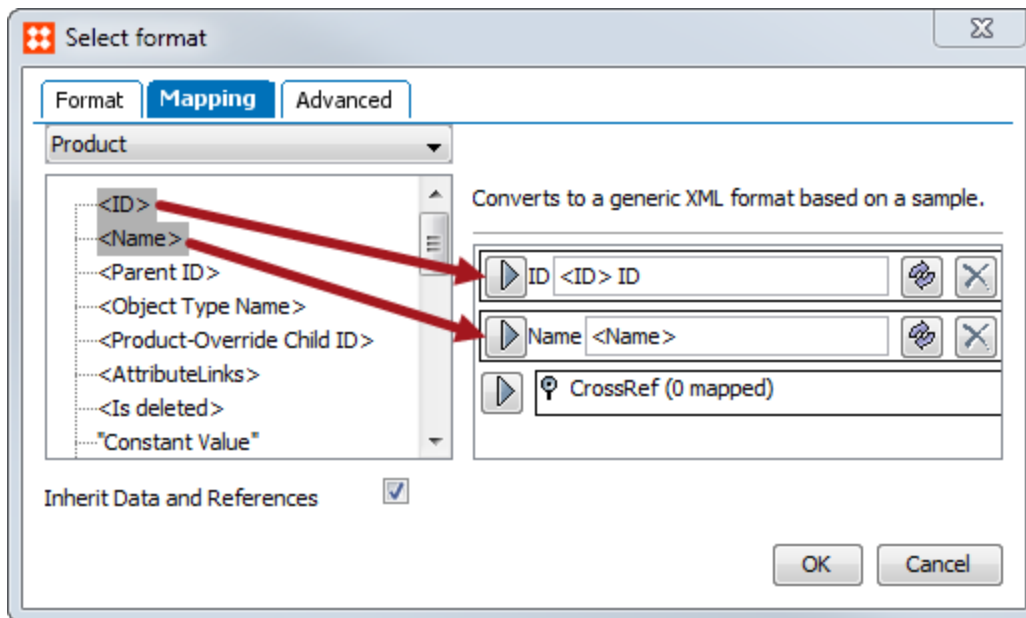
```
<Products>
  <Product ID="120223">
    <Name>120223</Name>
    <CrossRefs>
      <CrossRef refTypeID="CrossReference" DeletedStatus="">
        <IDTarget>120224</IDTarget>
      </CrossRef>
      <CrossRef refTypeID="CrossReference" DeletedStatus="IsDeleted">
        <IDTarget>7859</IDTarget>
      </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.

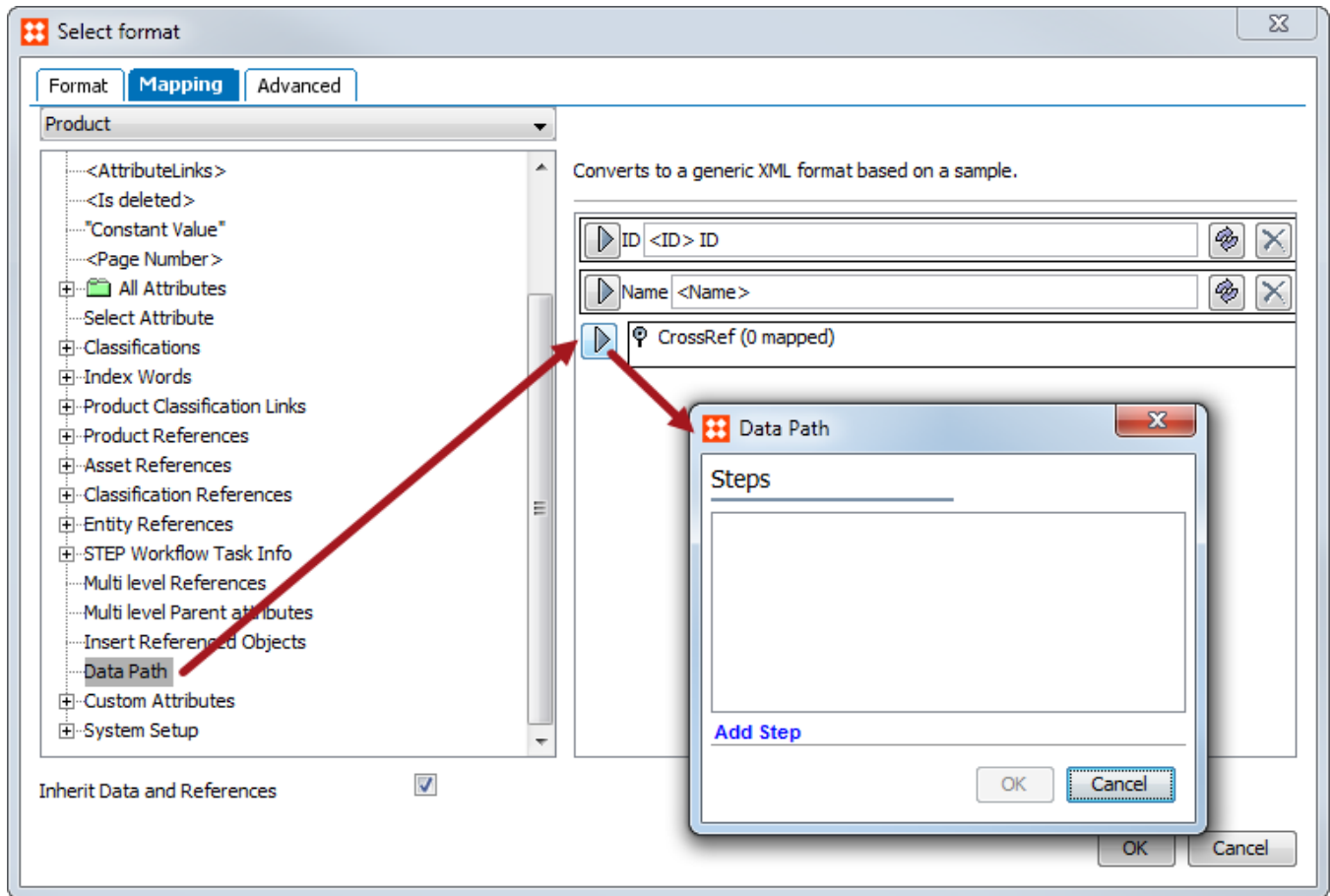
Mapping to Delete Product References with a Generic XML OIEP

The **Deleting 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.

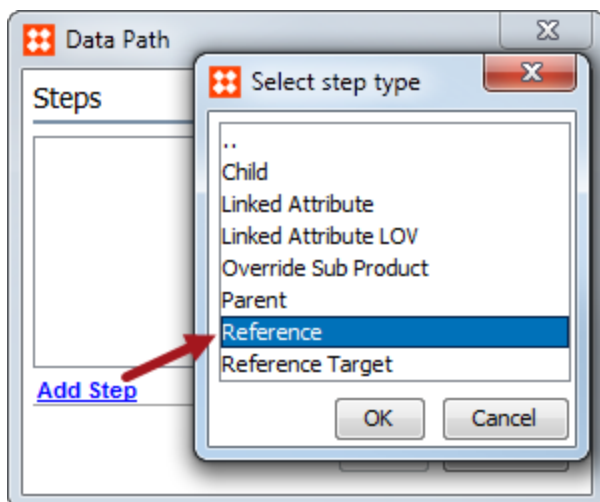
1. Map **ID** and **Name** for the Product.



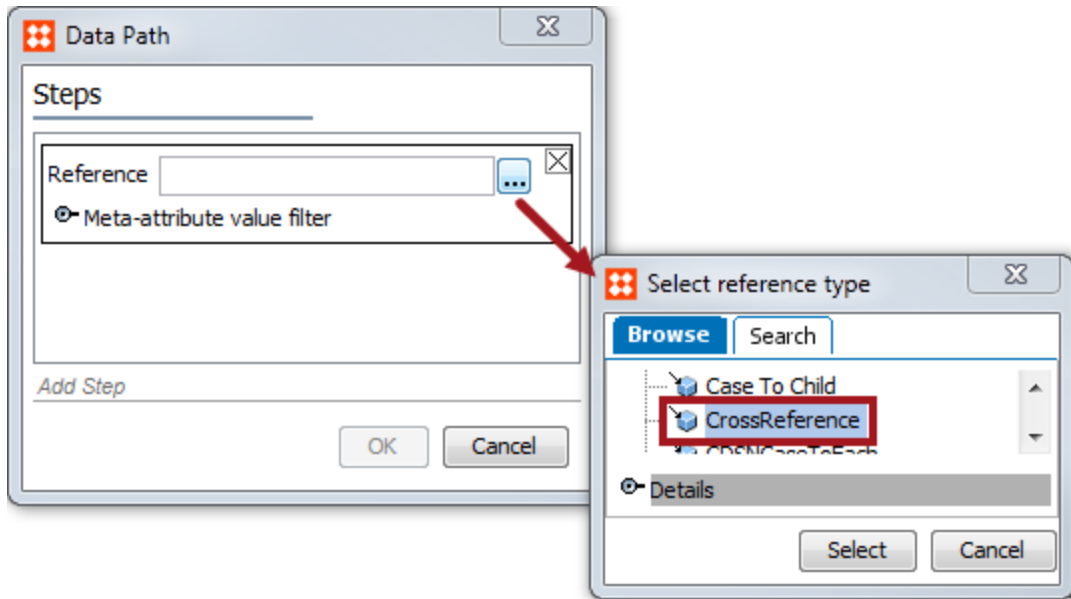
2. Map **Data Path** to CrossRef.



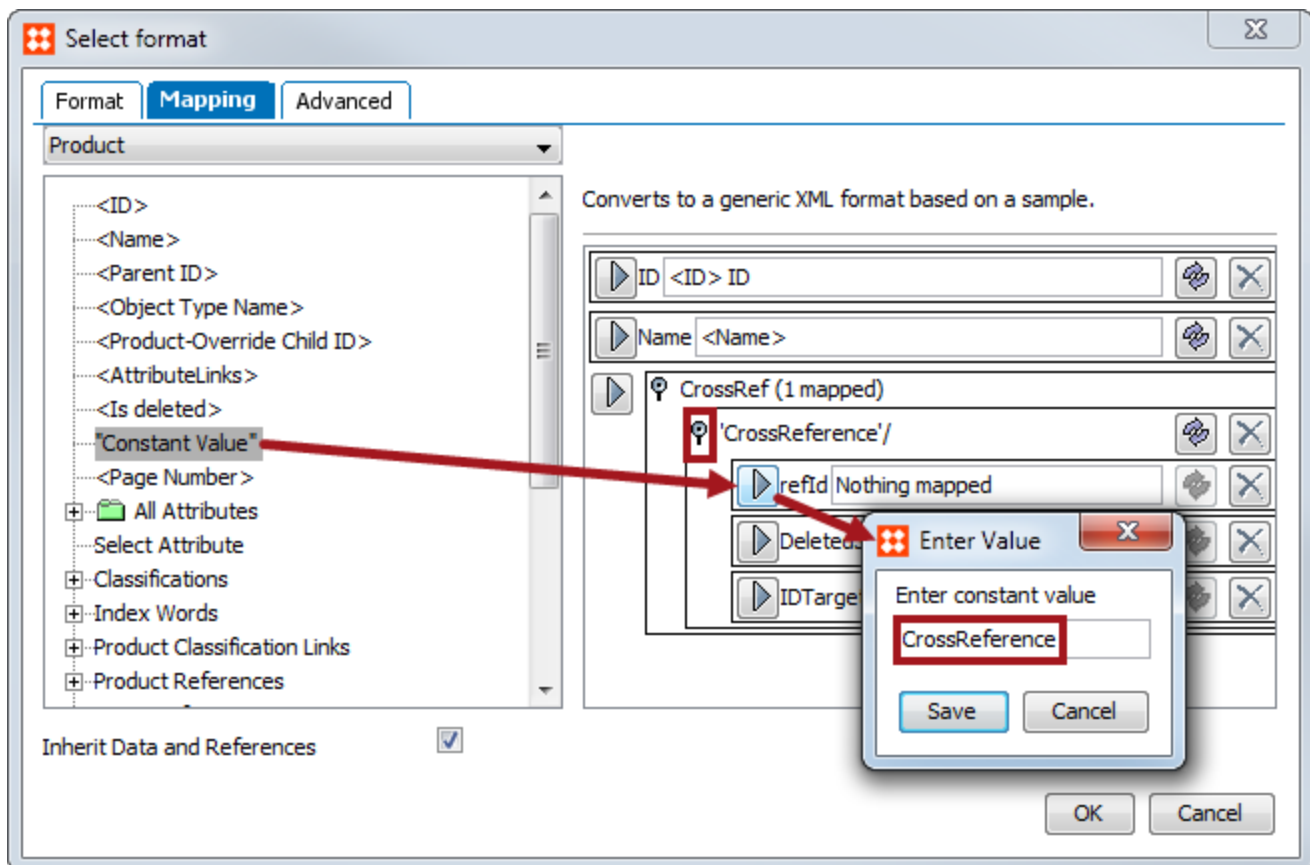
3. Click the **Add Step** link and select **Reference**.



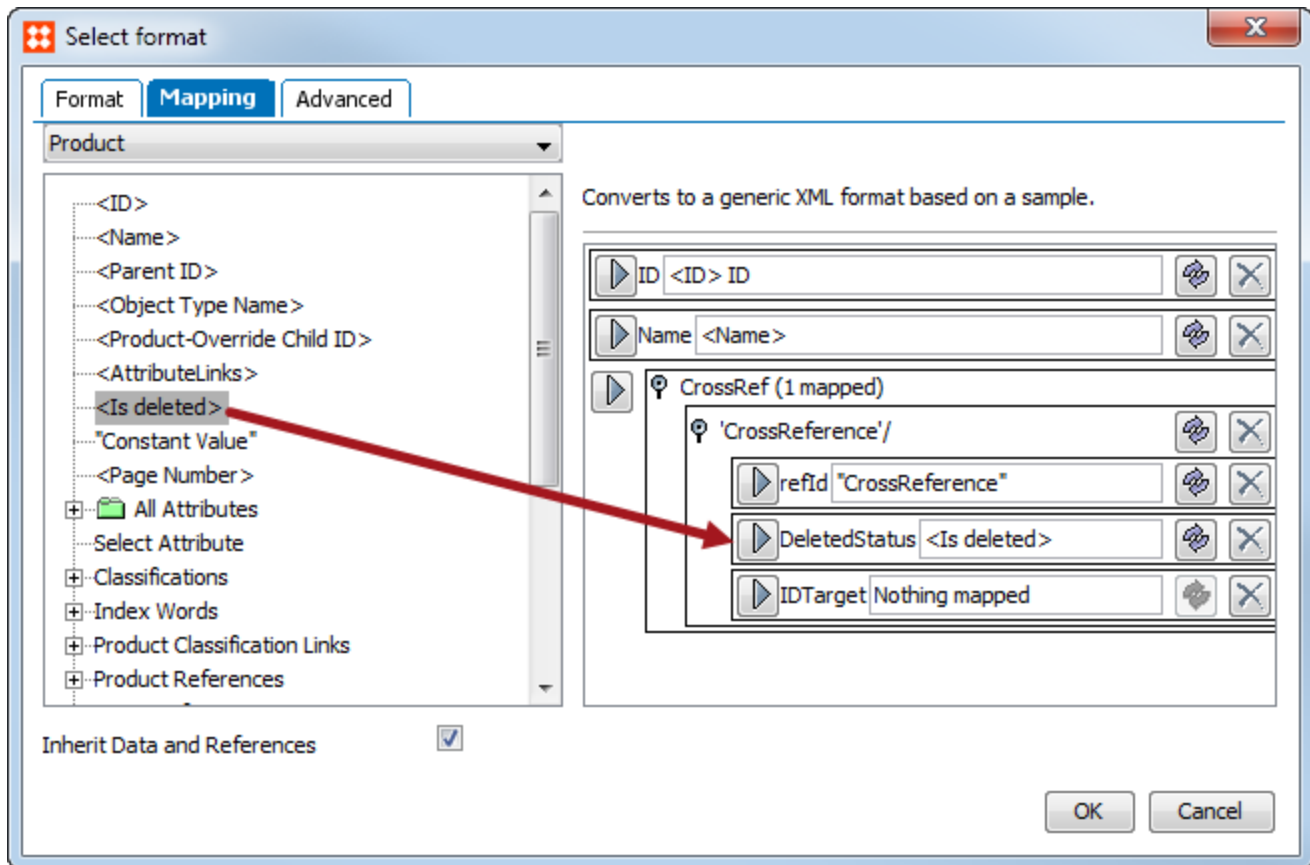
4. Click **OK** to close the **Select step type** dialog.
5. On the Data Path dialog, click the ellipsis button (...) and select the **Reference Type** to monitor, then click the **Select** button.



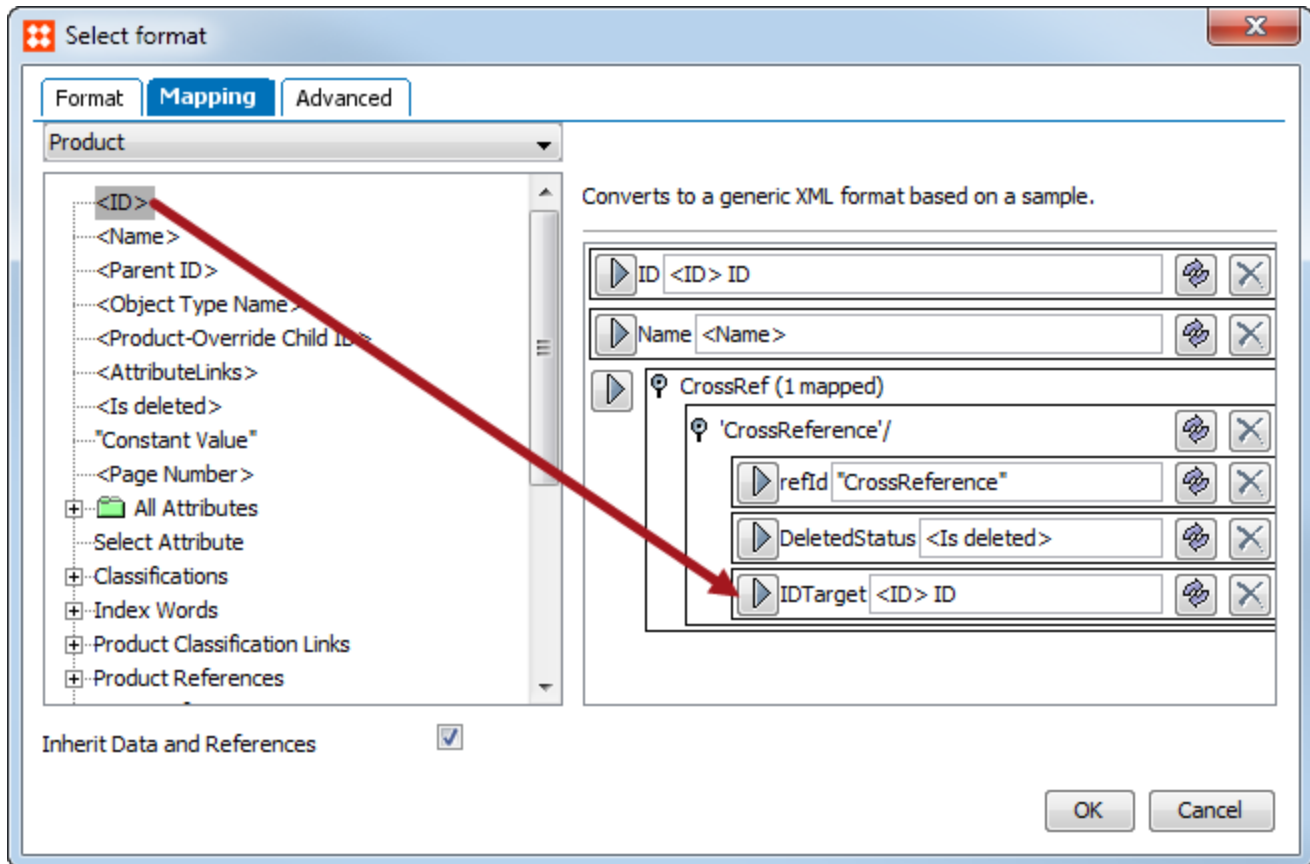
6. Click **OK** to close the **Data Path** dialog.
7. Open the 'CrossReference' flipper, map **Constant Value** to refID, add the text, and click the **Save** button.



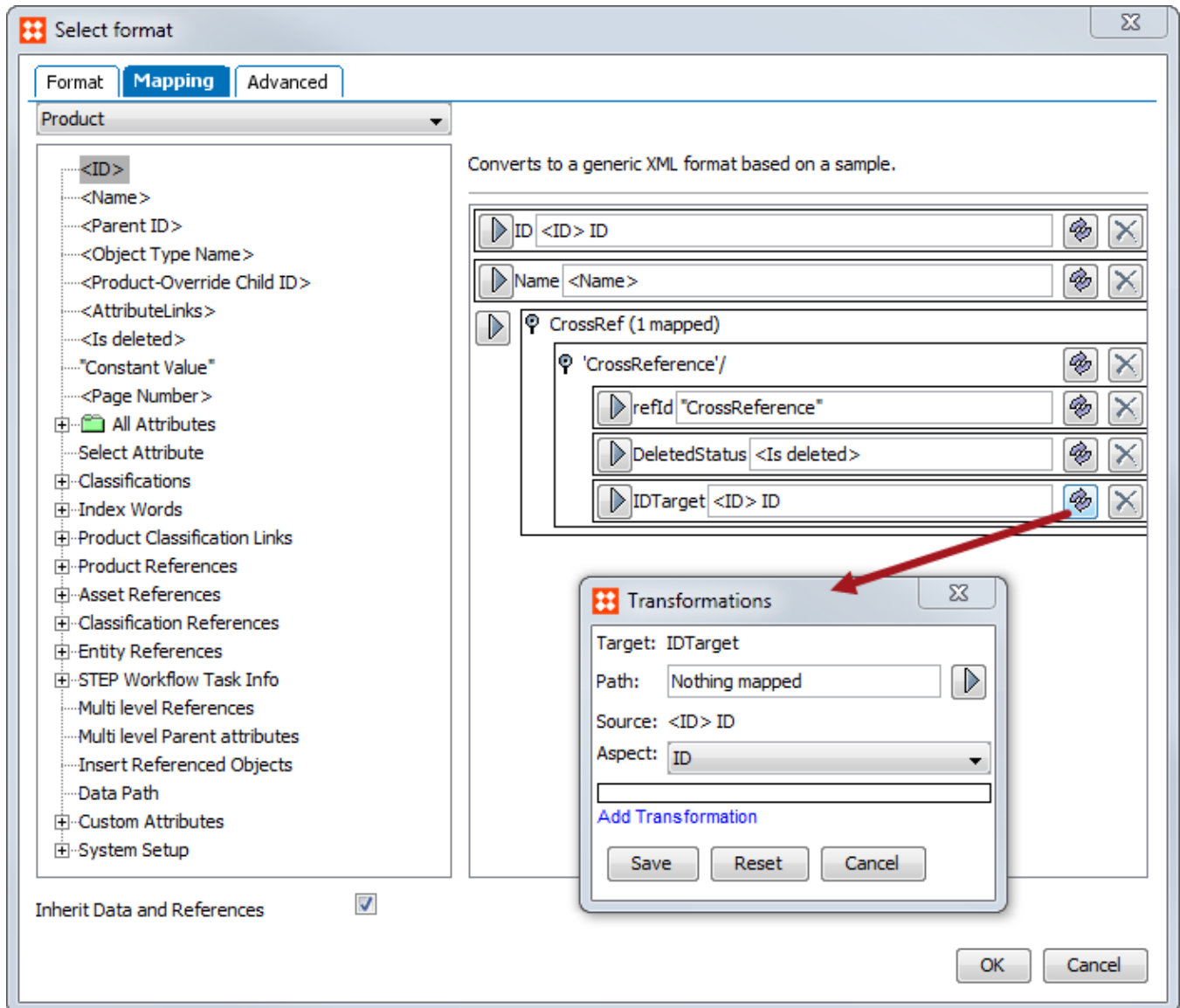
- Map **<Is deleted>** to DeletedStatus.



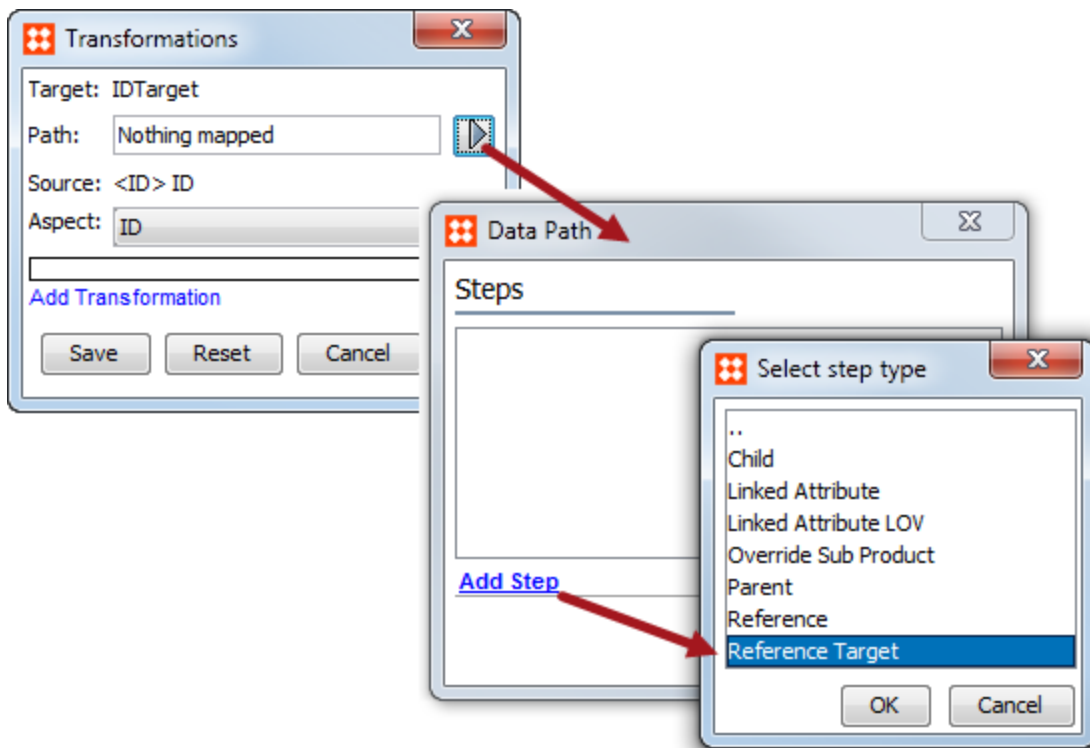
- Map **ID** to IDTarget.



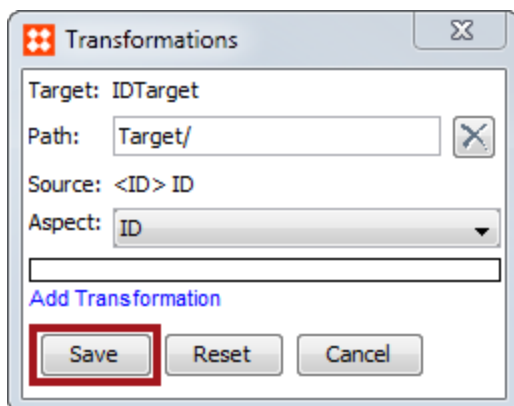
10. Click the **Transformations** button.



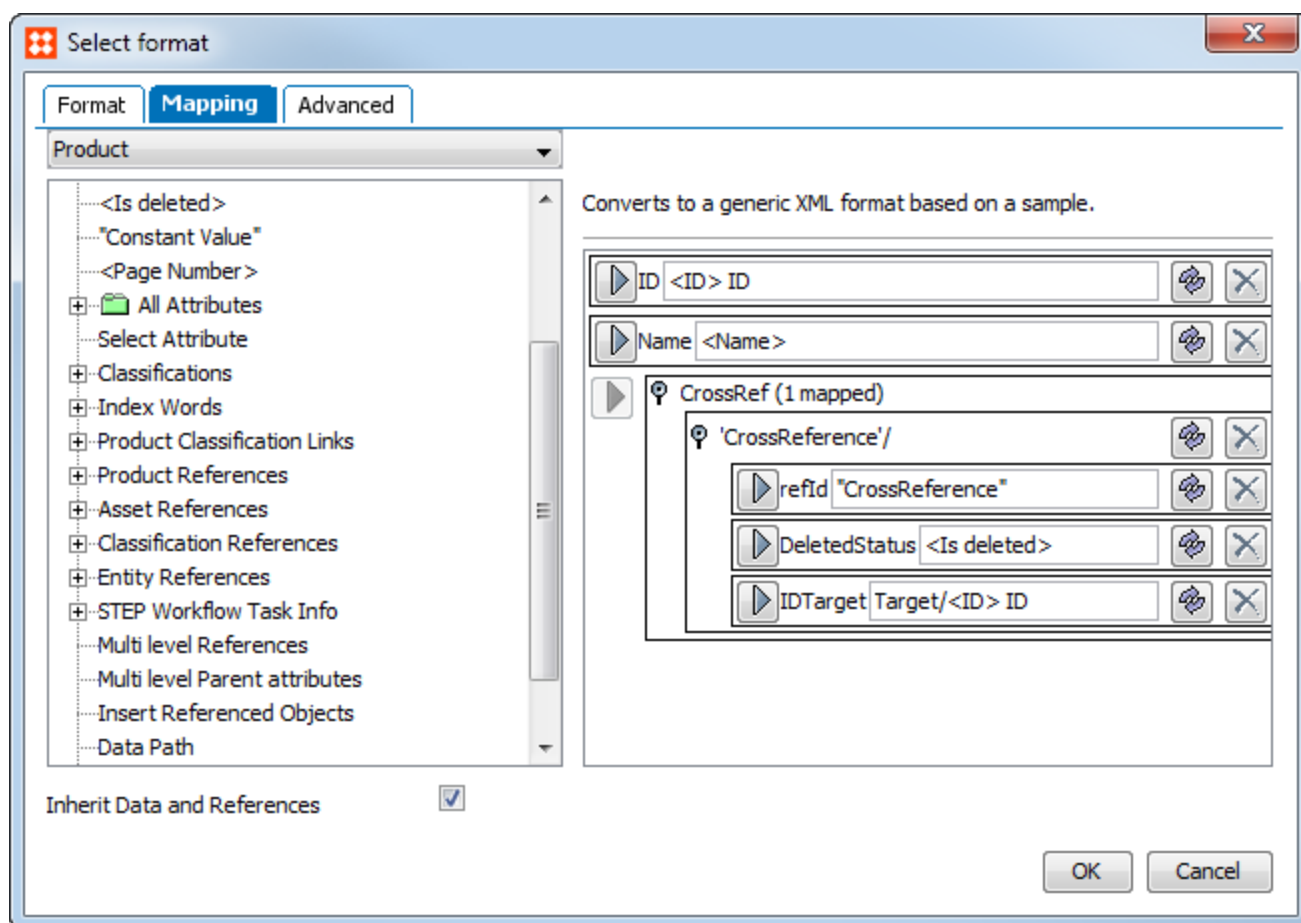
11. Click the **Path** button, click the **Add Step** link, and select **Reference Target**.



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.

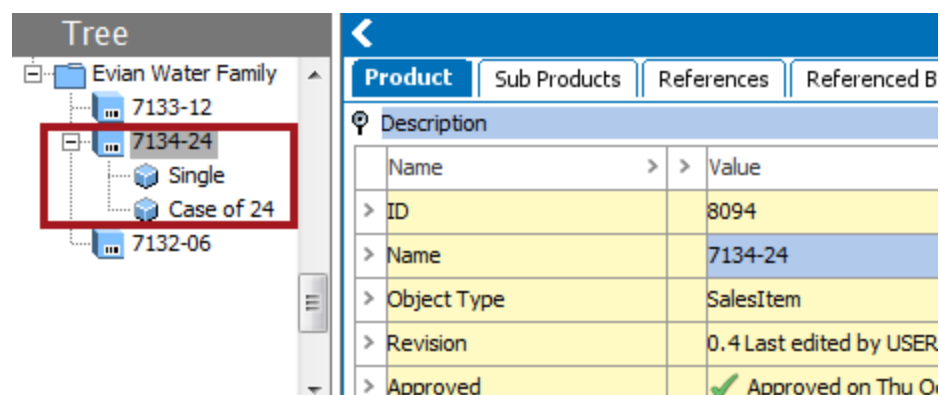


DoNotEmit Instruction in Generic XML

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 SalesItem and children are selected for export.

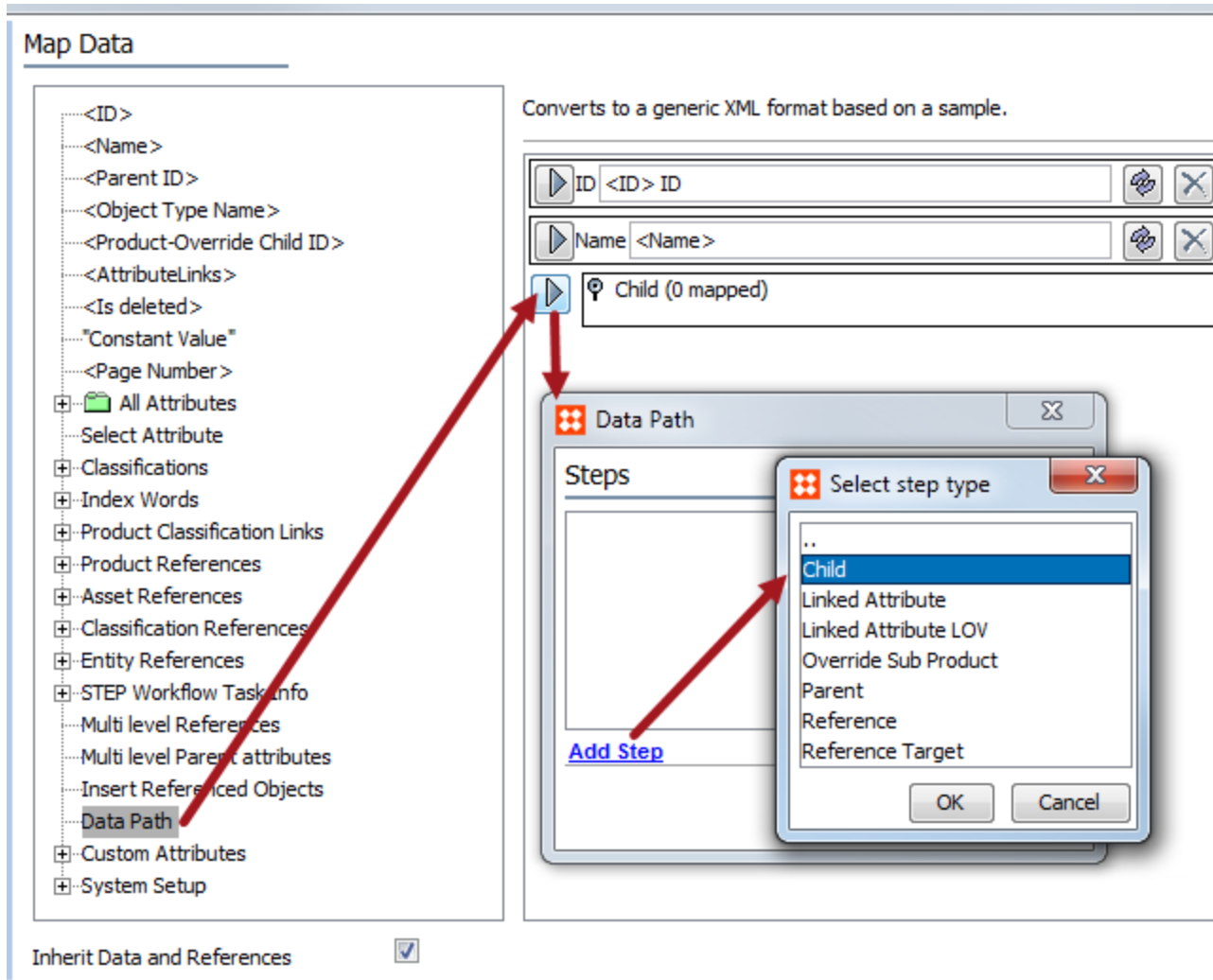


Template

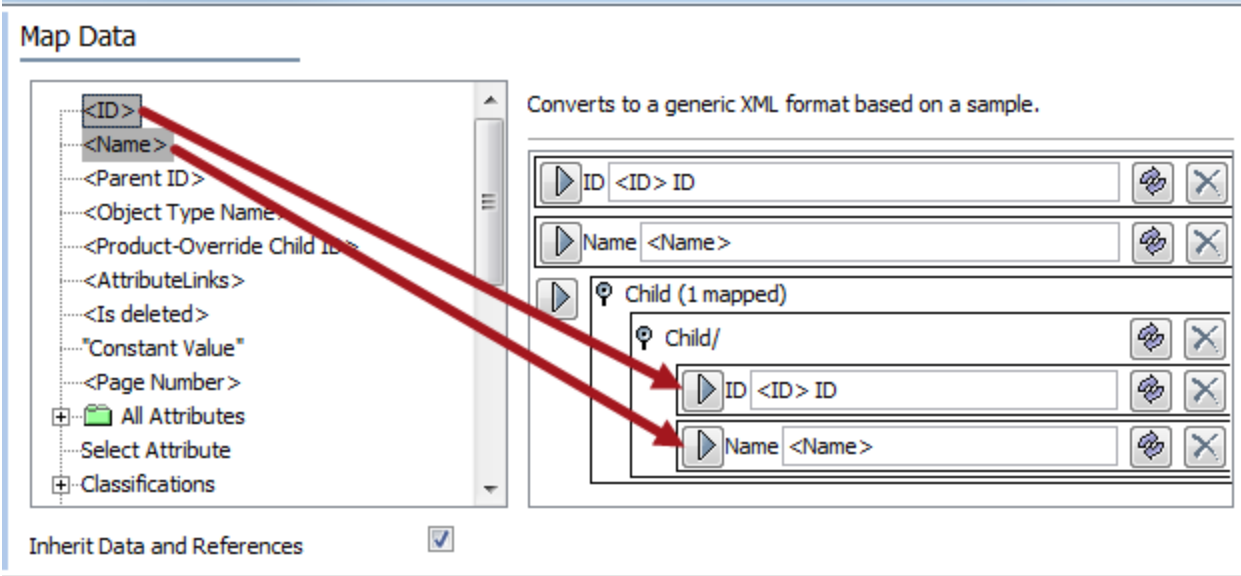
Without DoNotEmit	With DoNotEmit
<pre><Families> <Family ID="[/Target ID?]"> <?Record?> <Name><?Target Name?></Name> <Children> <Child> <?MultiTarget?> <ID><?Target ID?></ID> <Name><?Target Name?></Name> </Child> </Children> </Family> </Families></pre>	<pre><Families> <Family ID="[/Target ID?]"> <?Record?> <Name><?Target Name?></Name> <Children> <Child> <?DoNotEmit?> <?MultiTarget?> <ID><?Target ID?></ID> <Name><?Target Name?></Name> </Child> </Children> </Family> </Families></pre>

Mapping

Mapping is the same for both templates. After mapping the SalesItem ID and Name, use the Data Path - Child option to map the Child target.



Finally, map the Child's ID and Name.



Results

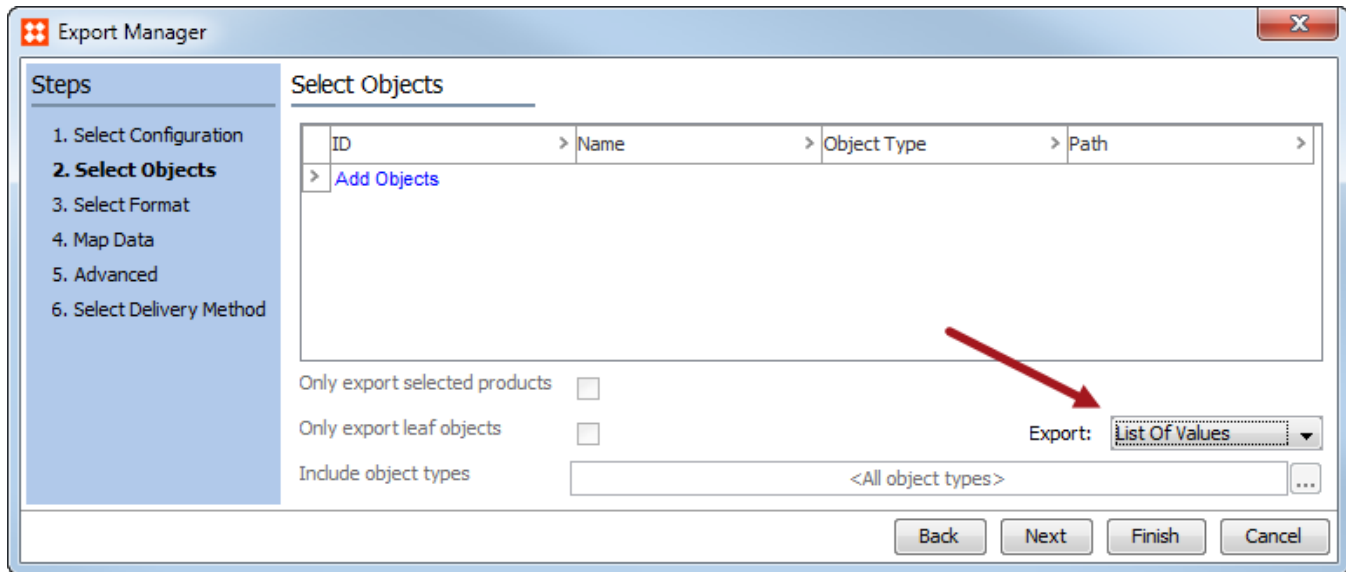
Without DoNotEmit	With DoNotEmit
<pre><Families> <Family ID="8094"> <Name>7134-24</Name> <Children> <Child> <ID>8082</ID> <Name>Single</Name> </Child> <Child> <ID>107807</ID> <Name>Case of 24</Name> </Child> </Children> </Family> </Families></pre>	<pre><Families> <Family ID="8094"> <Name>7134-24</Name> <Children> <ID>8082</ID> <Name>Single</Name> <ID>107807</ID> <Name>Case of 24</Name> </Children> </Family> </Families></pre>

List of Values Object Type in Generic XML

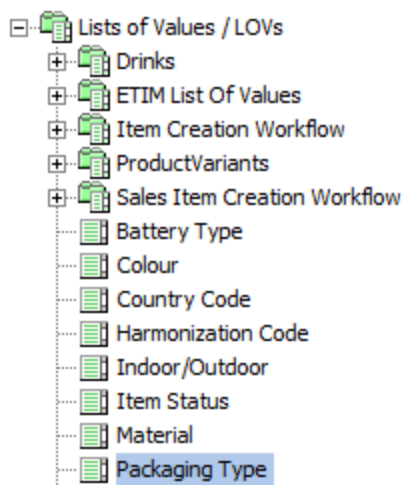
Export Manager requires that you select objects and provide both a sample template and mapping when exporting Generic XML.

Objects

Choose List of Values (LOV) as object type to export.



Use the **Add Objects** link to select the data to export.



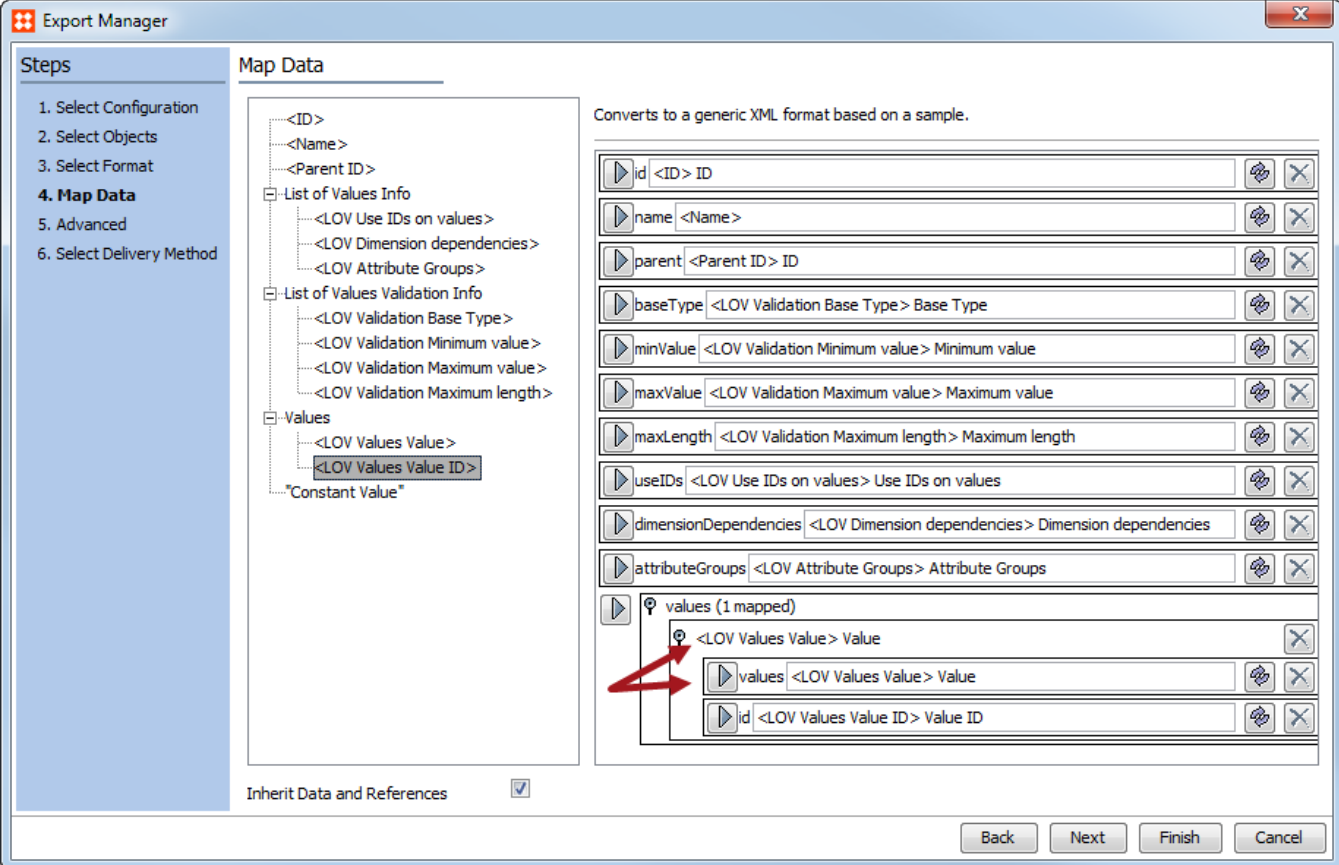
Template

Choose **Generic XML** for **Format** and provide the following template text in the Sample field.

```
<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



The LOV Values Value node is mapped twice to allow the Value ID and content to be displayed on the same line.

Results

```
<root>
  <lov>
    <id>PackagingTypeLOV</id>
    <name>Packaging Type</name>
    <parent>List Of Values group root</parent>
    <baseType>text</baseType>
    <minValue/>
    <maxValue/>
    <maxLength>100</maxLength>
    <useIDs>true</useIDs>
    <dimensionDependencies>Country; Language</dimensionDependencies>
    <attributeGroups>AttributeGroup</attributeGroups>
    <values id2="SPL">Spool</values>
    <values id2="TU">Tube</values>
    <values id2="BX">Box</values>
    <values id2="CT">Carton</values>
    <values id2="BP">Banded Package</values>
    <values id2="LSE">Loose</values>
    <values id2="SW">Shrinkwrapped</values>
    <values id2="CS">Case</values>
    <values id2="CY">Cylinder</values>
    <values id2="BG">Bag</values>
    <values id2="PX">Pallet</values>
    <values id2="CG">Card</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.

Map Data

All Attributes

Attribute Group

Blacklisted - 2

Blacklisted Sales Item Maintenance

Business Rules

Category Specific Attributes

Customer Data

Dates

DescriptionAttribute

Display

Item Maintenance

Item Description Information

Item Import/Export Information

Country of Origin

Harmonization Code

Converts to a generic XML format based on a sample.

ProductID <ID> ID

Value (1 mapped)

Country of Origin Value and unit

Inherit Data and References ☒

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.

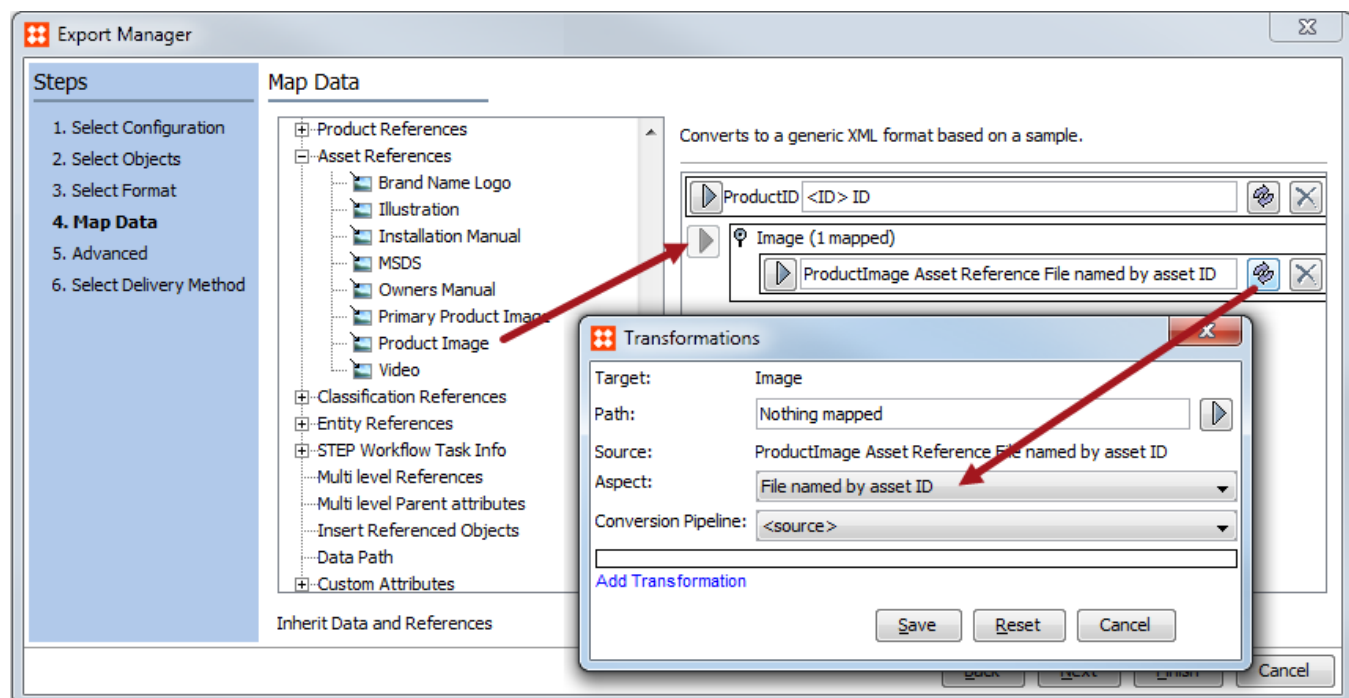
Image References

Reference Type: Product Image

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

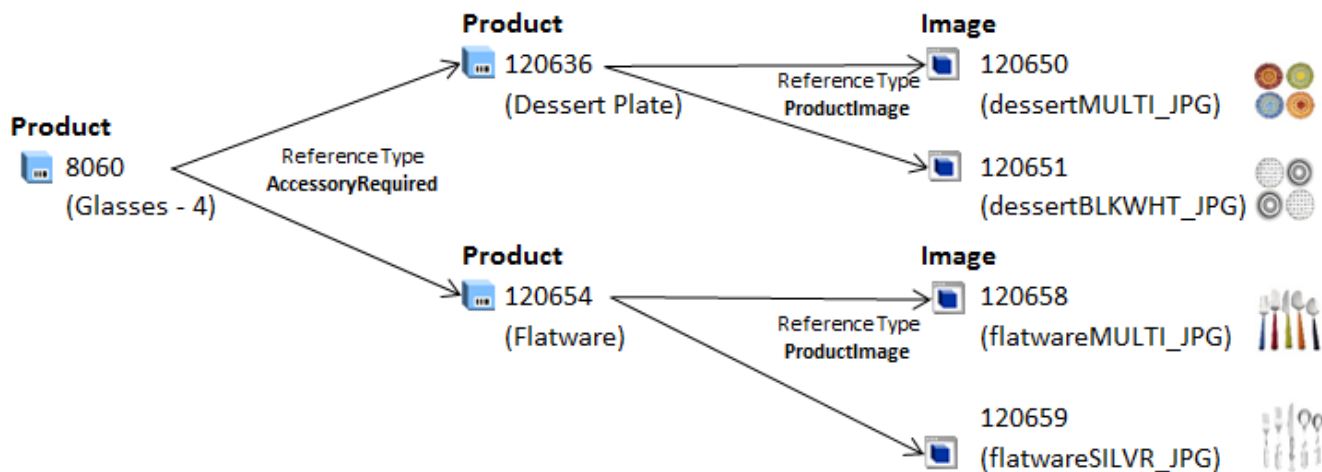
The data sources that are to be mapped below a Data Path source require special support for evaluating relative to the objects / references identified by the paths. The following standard data sources support such nesting:

- 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.

Results

© Stibo Systems - Confidential - STEP Trailblazer 8.1

198

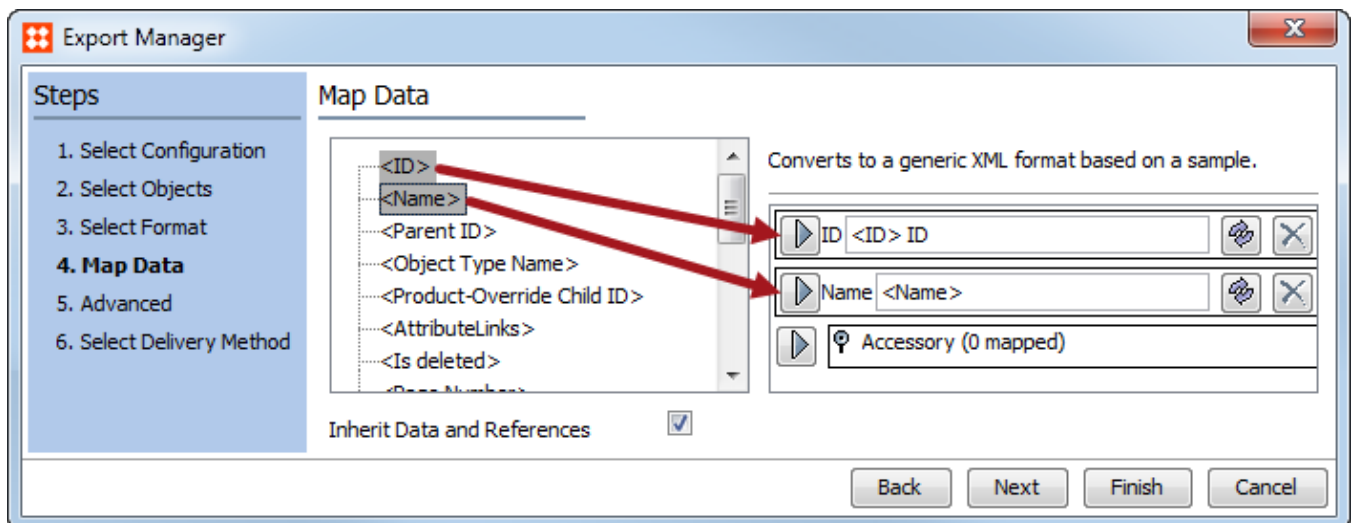
```

<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>

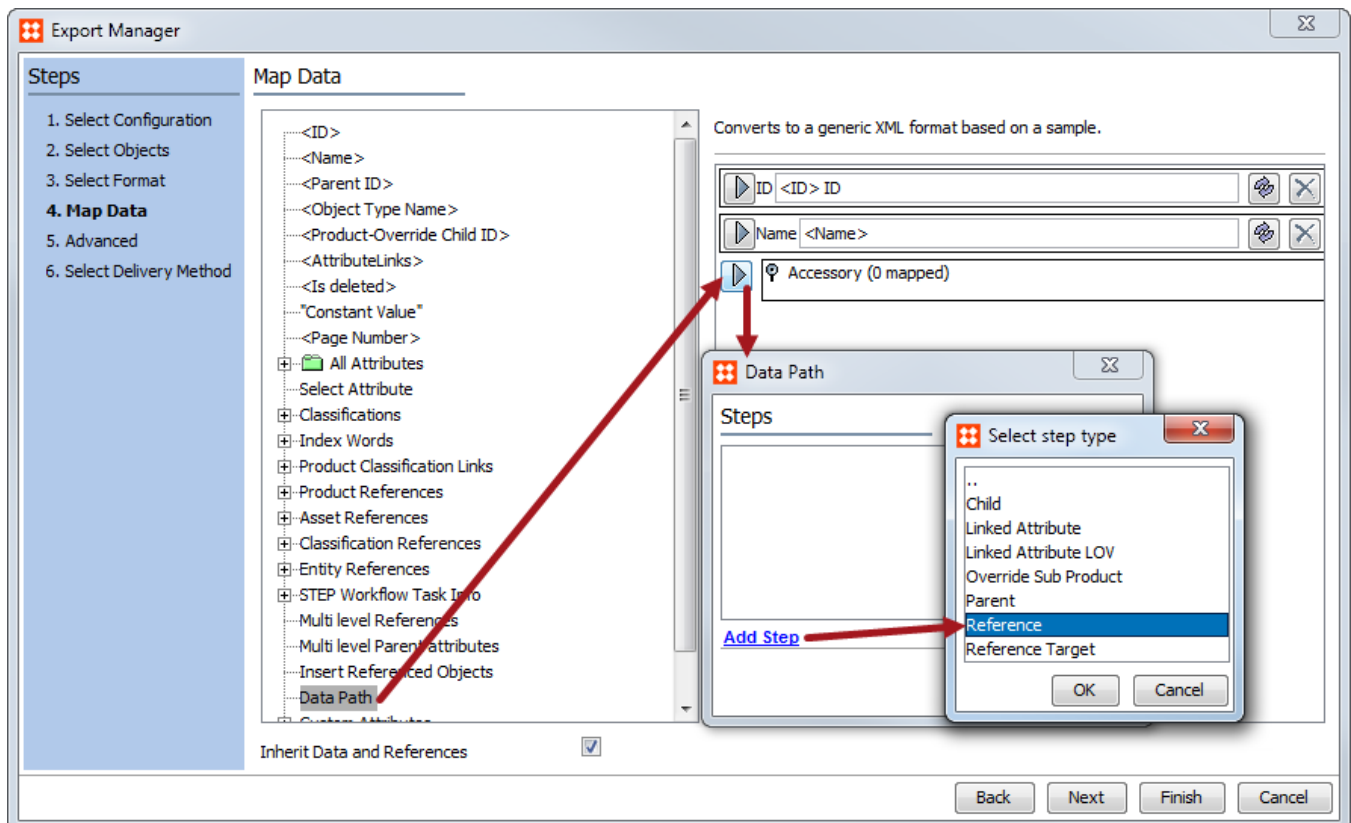
```

Mapping for Nestable Data Paths in Generic XML Example

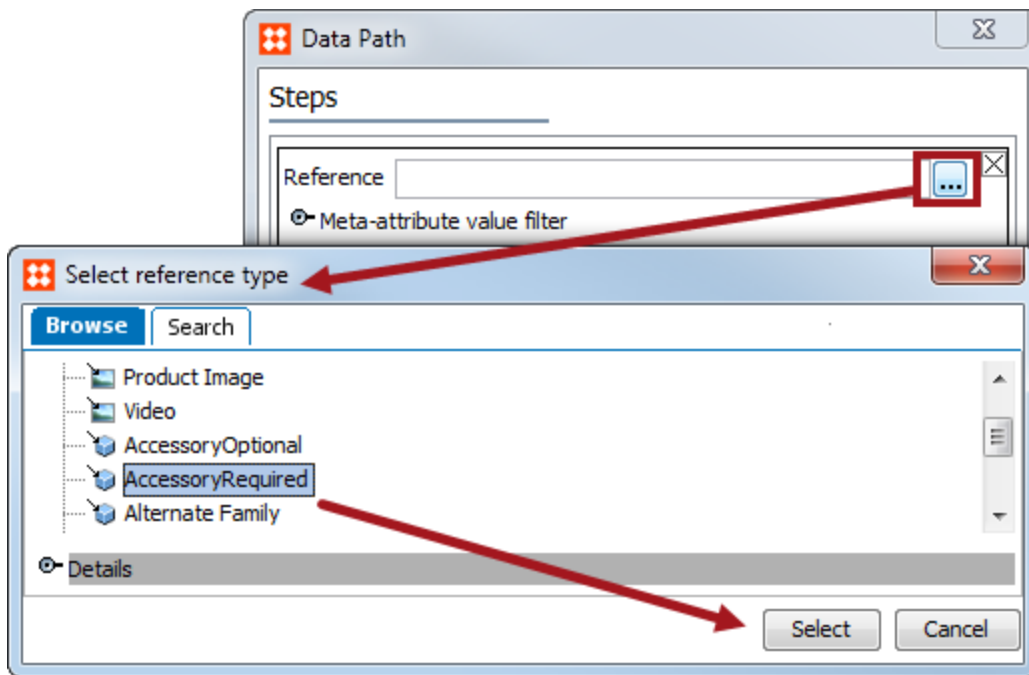
1. Map **ID** and **Name** for Level 1.



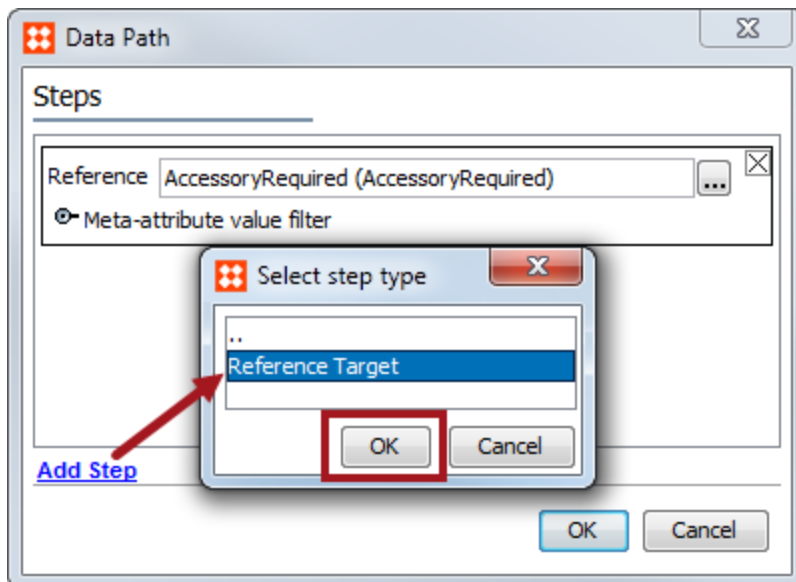
2. Map **Data Path** to Accessory, click the **Add Step** link, select the **Reference** step type, and click **OK**.



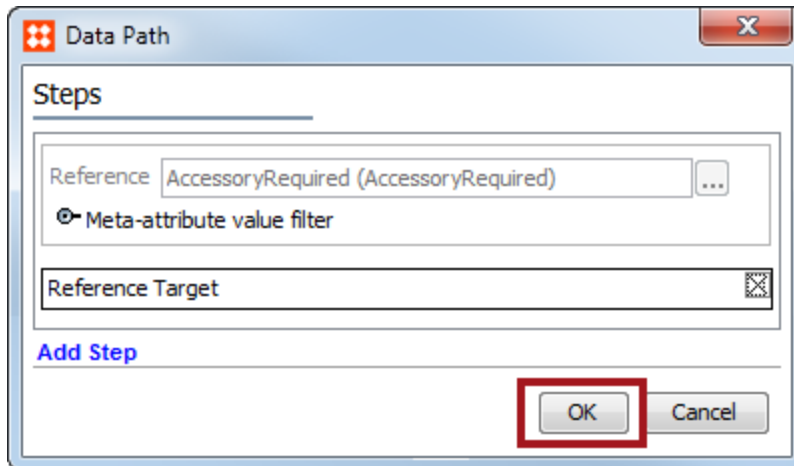
3. On the Data Path dialog, click the ellipsis button (...), select AccessoryRequired reference type, and click the **Select** button.



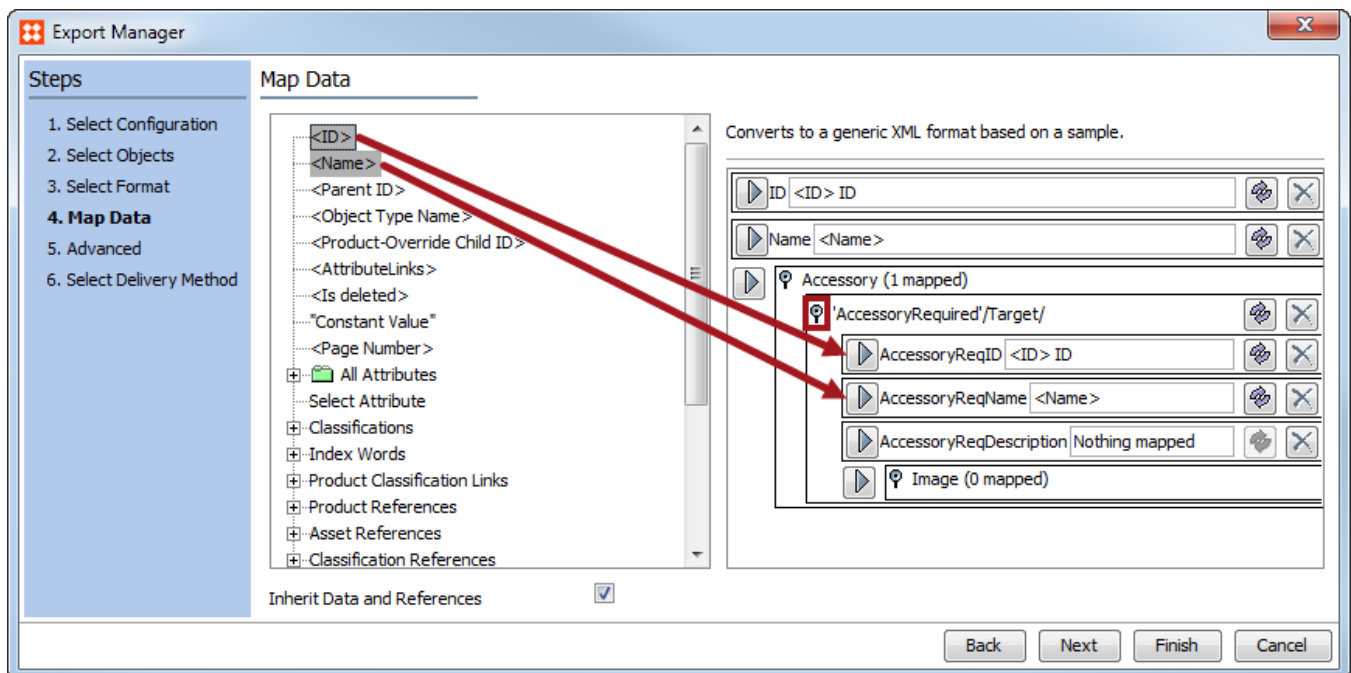
4. Click the **Add Step** link, select **Reference Target** for Accessory and click **OK**.



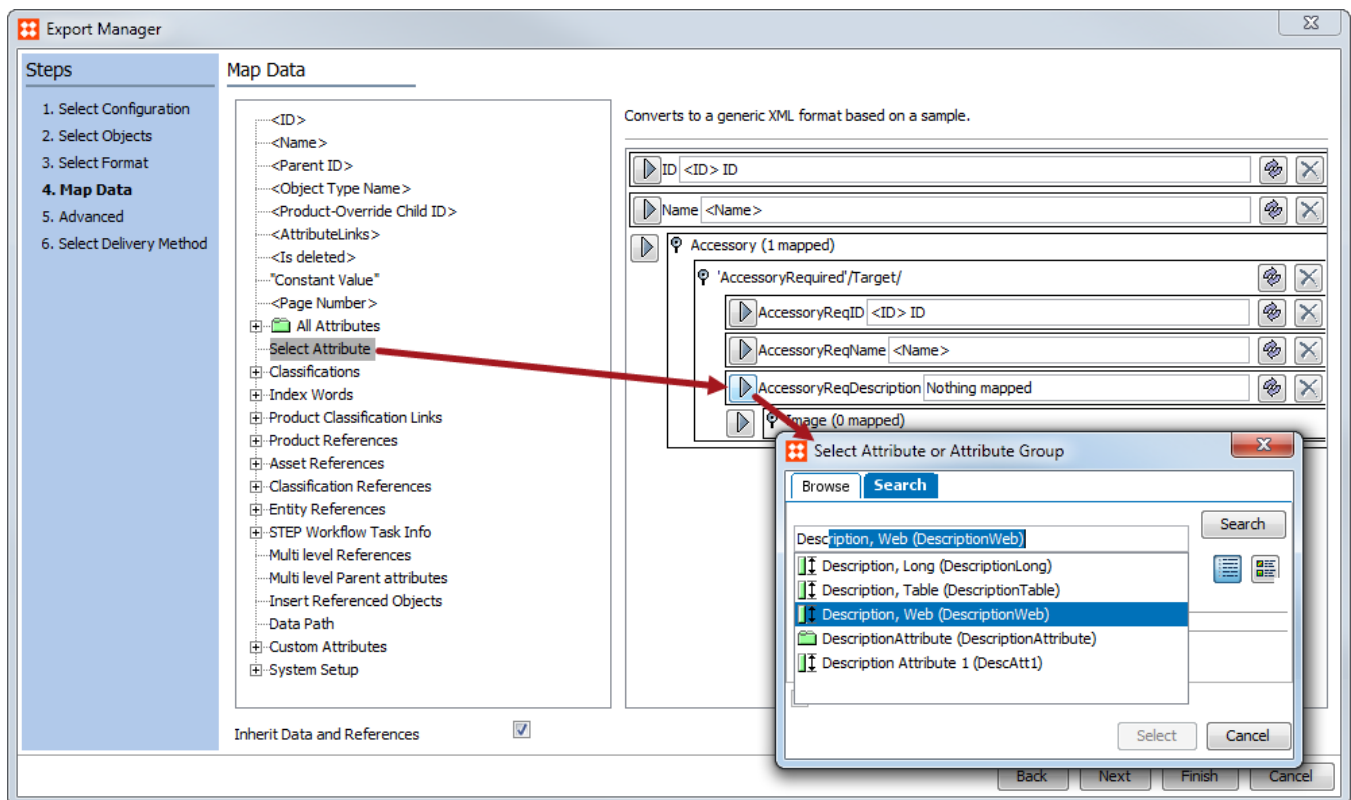
5. Click **OK** to confirm the Reference and Reference Target Data Path fields for the Accessory.



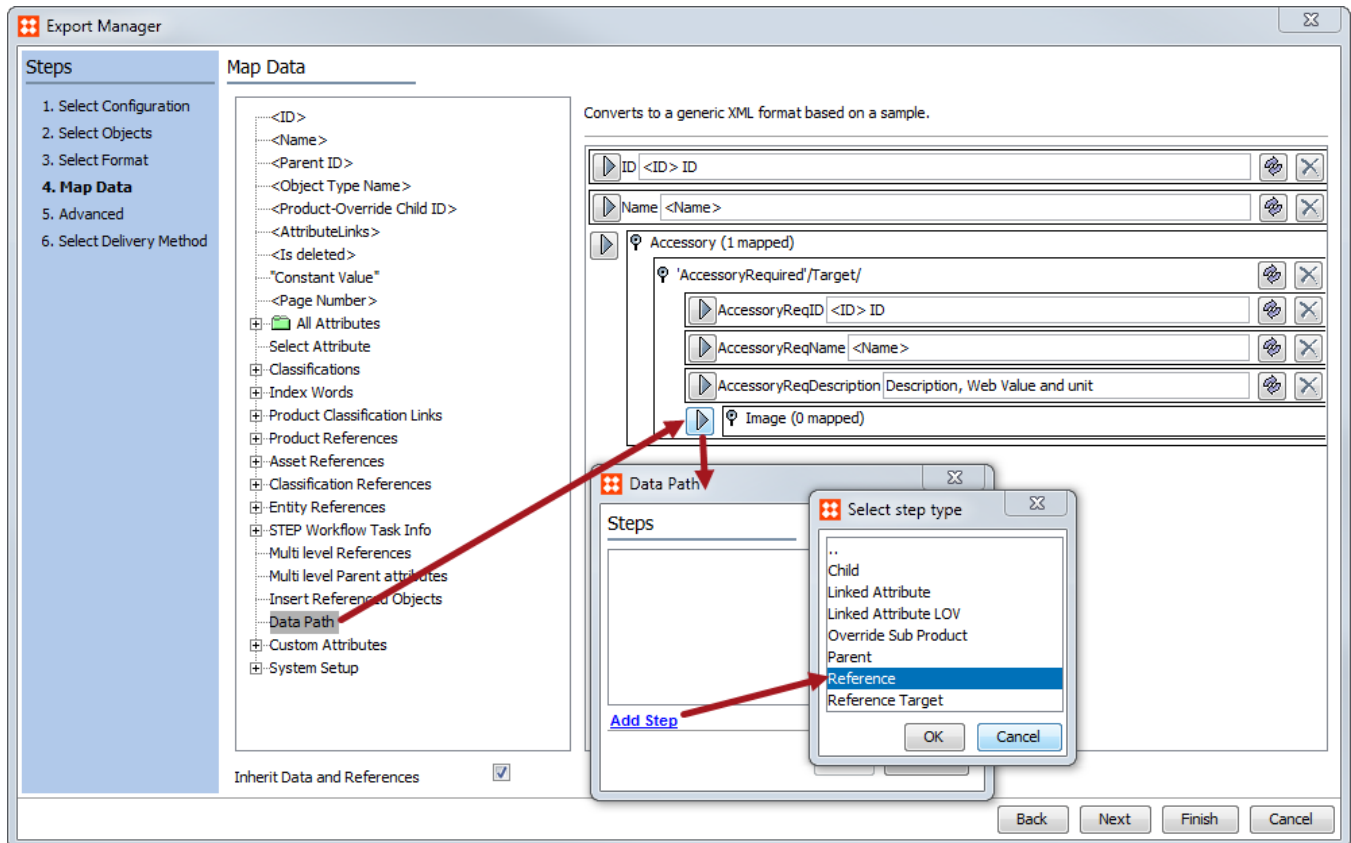
- Open the 'AccessoryRequired'/Target/ flipper to display the nested mapping options. Map **ID** and **Name** for the Level 2 products being exported.



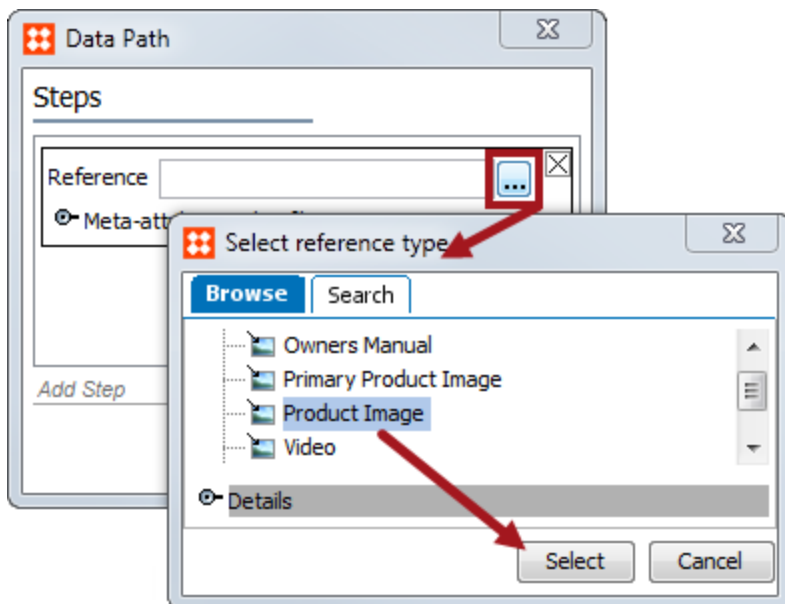
- Use **Select Attribute** to map 'Description, Web' for Level 2 (products) and click the **Select** button.



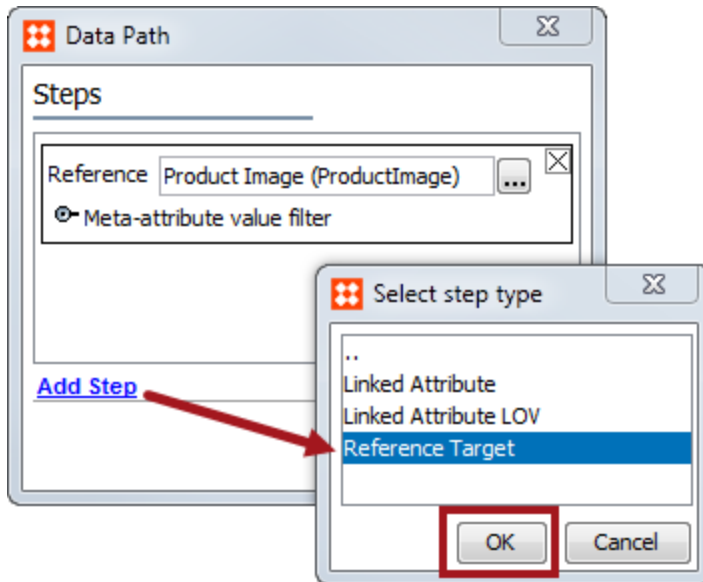
8. Map **Data Path** to Image, click the **Add Step** link, select the **Reference** step type, and click **OK**.



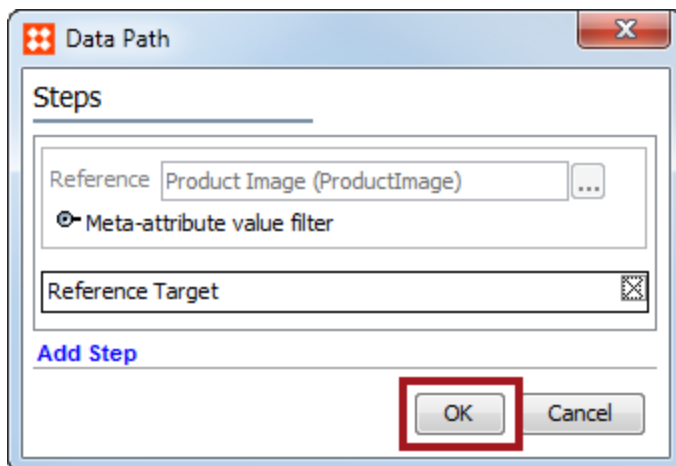
- On the Data Path dialog, click the ellipsis button (...), select ProductImage reference type, and click the **Select** button.



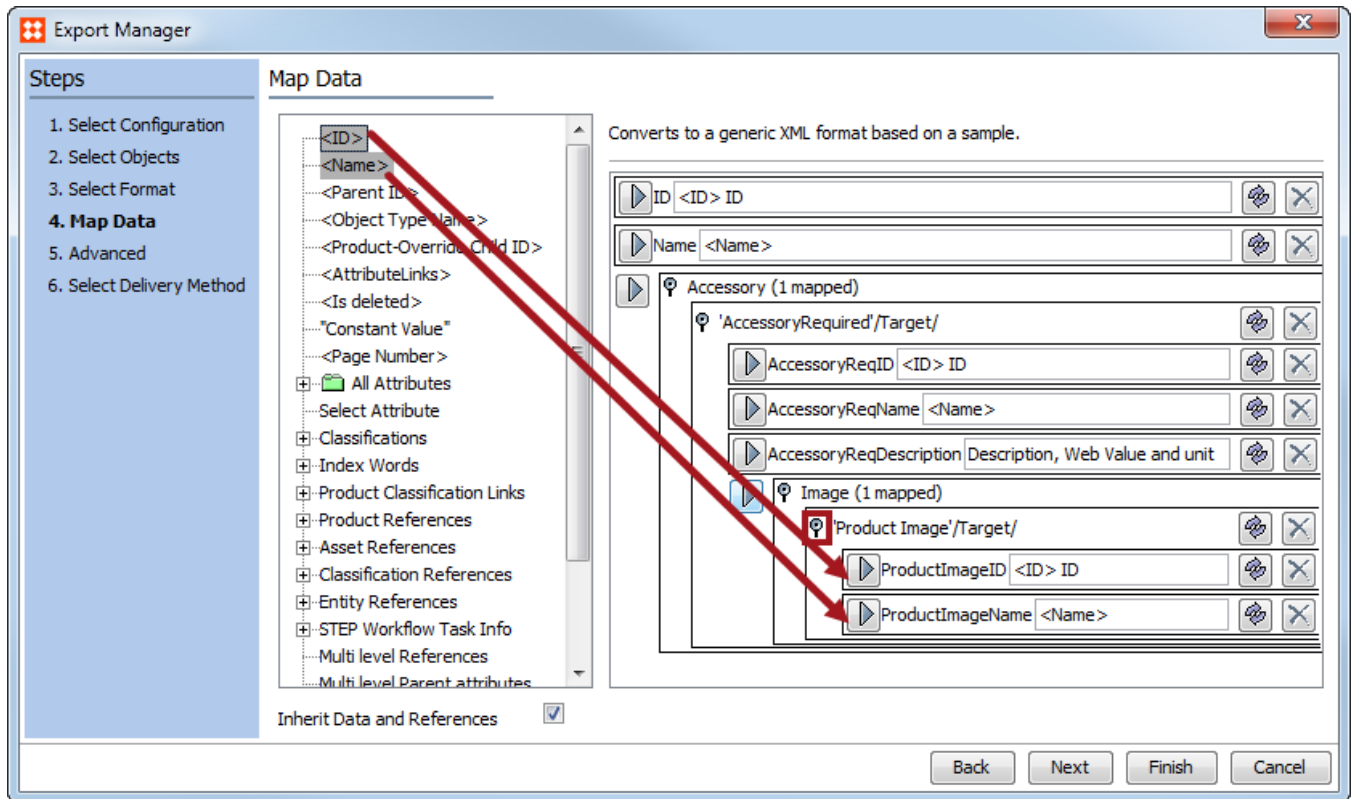
- Click the **Add Step** link, select **Reference Target** for Image and click **OK**.



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> <ProviderGLN><?Target?></ProviderGLN> </mandatoryAttributes> <optionalAttributes> <?RemoveIfEmpty?> <PartNo><?Target?></PartNo> <Size><?Target?></Size> </optionalAttributes> </Product> </Products> </pre>	<pre> <Products> <Product> <?Record?> <ID><?Target?></ID> <NAME><?Target?></NAME> <mandatoryAttributes> <Parent><?Target?></Parent> <BrandName><?Target?></BrandName> <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.

<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

Inherit Data and References ☒

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>

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**

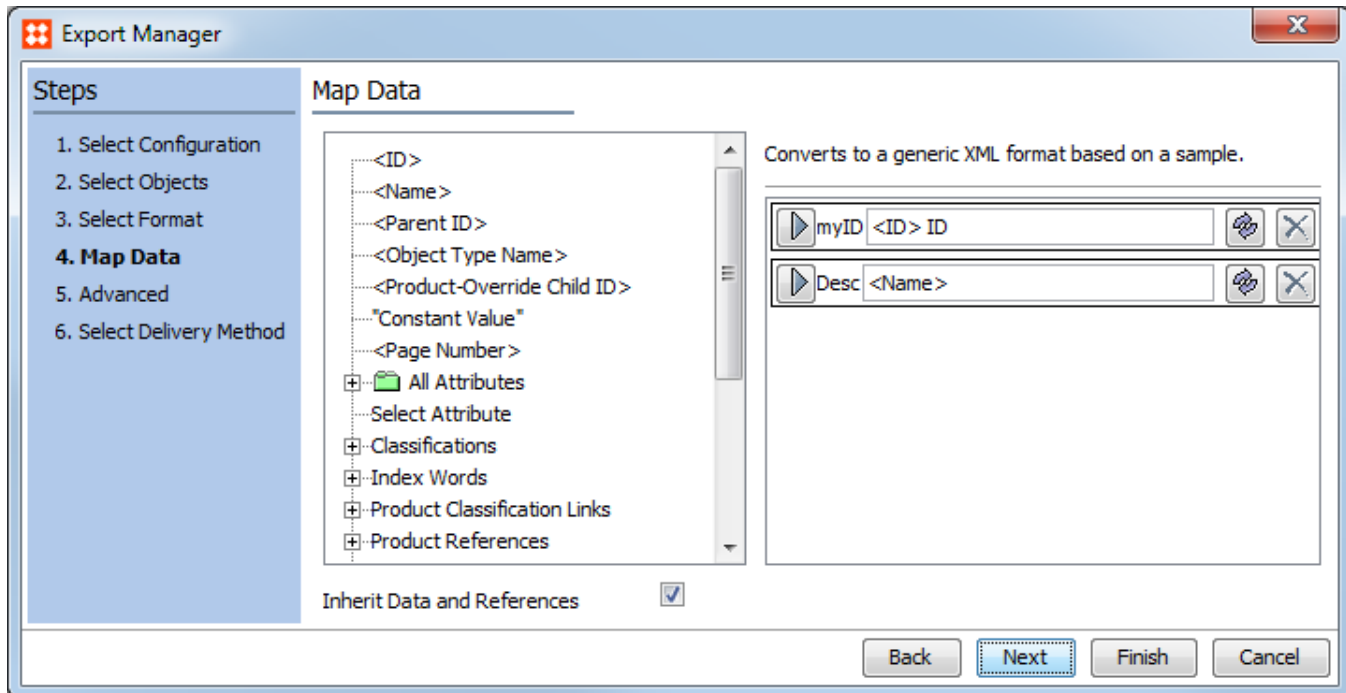
Include object types

Buttons: 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

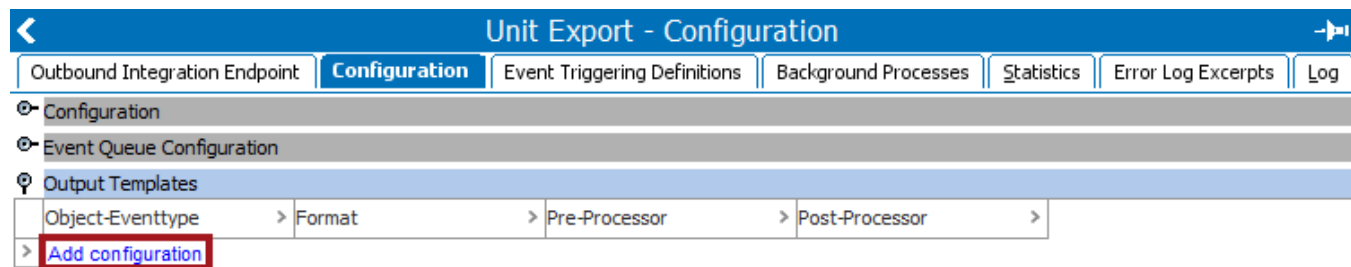
- There must be exactly one <?Record?> in the template.
- Mapping targets may only be used inside <?Record?> scope.
- Each mapping target must have a unique label.

Units with a Generic XML OIEP

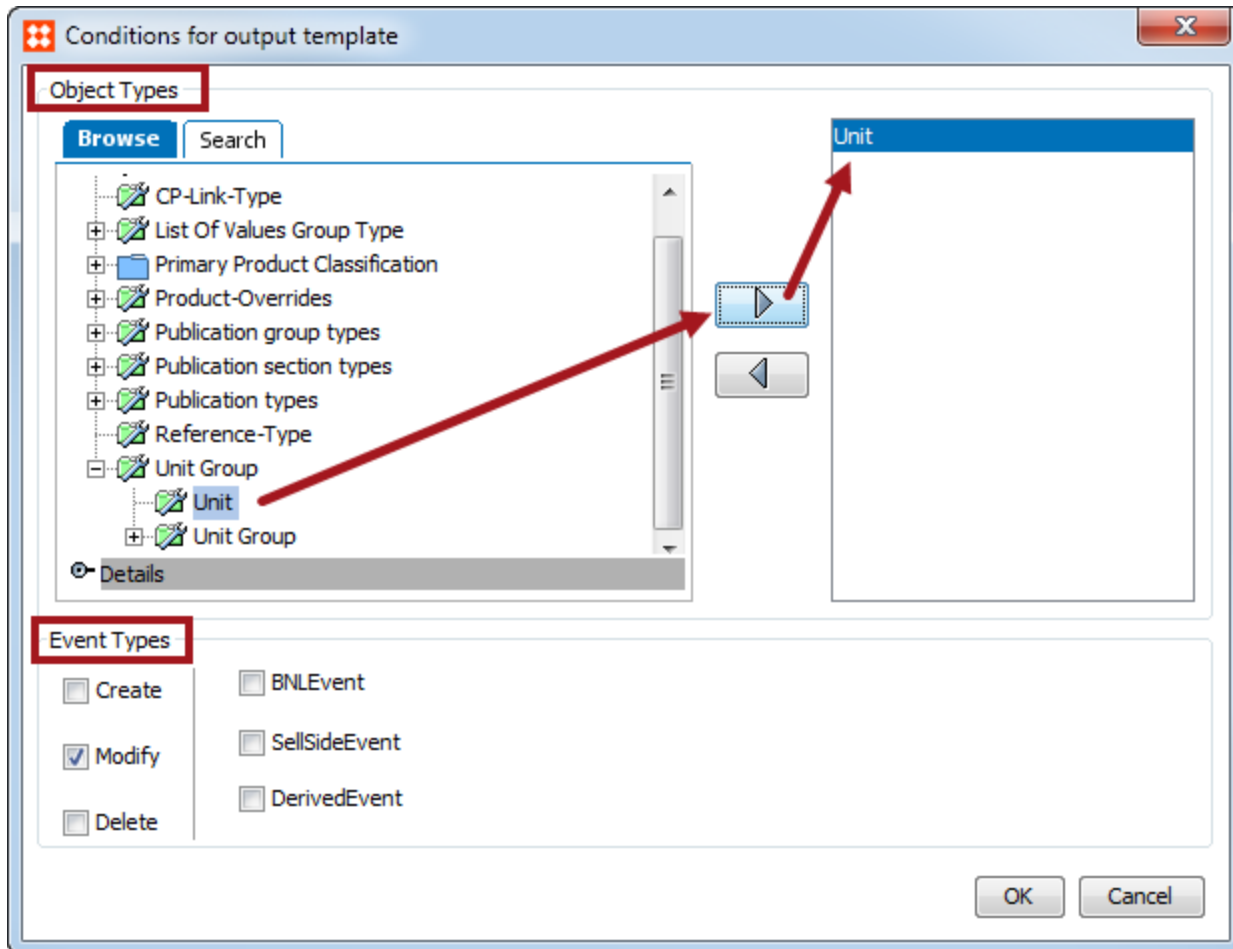
Outbound Integration Endpoints (OIEPs) allow the export of unit information.

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

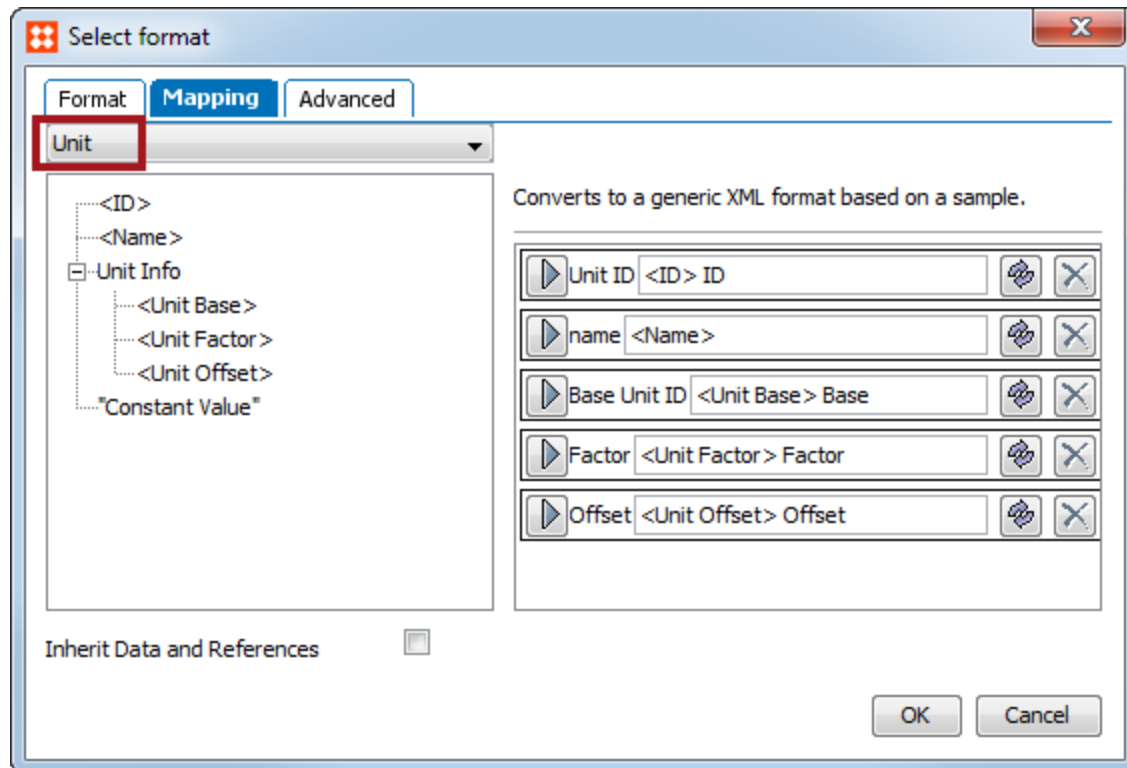
In the Output Template, click the ellipsis button (...), and set the format to **Generic XML**.

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.



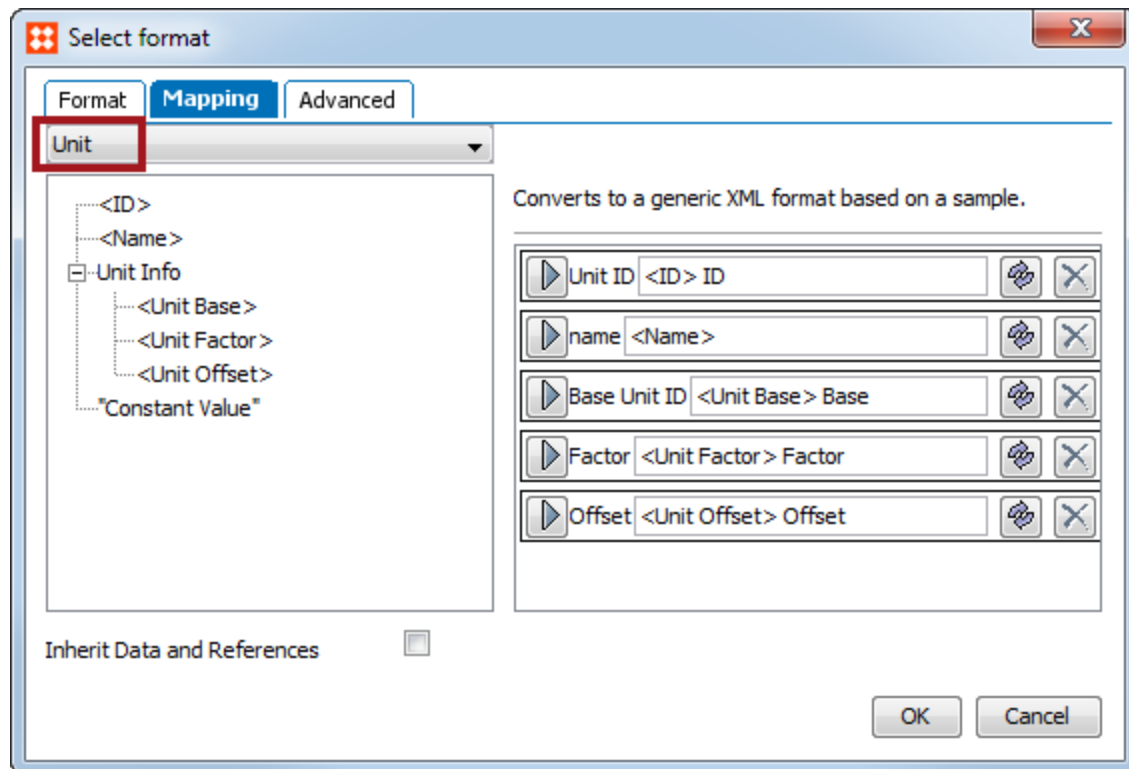
See the **Mapping Units Aspects with Generic XML OIEP** section for more information.

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. For example, if a unit has a base unit conversion of 'value(g) = .001 * value(kg),' then '.001' is

Aspect	Description
	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.

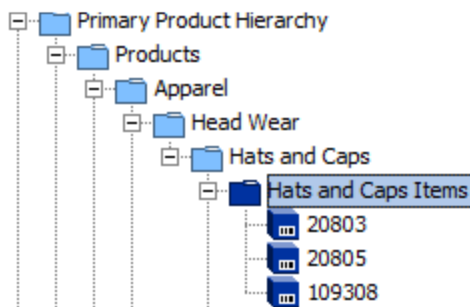
For a description of Constant Value, see the **Constant Value - Data Source Outbound** section.

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

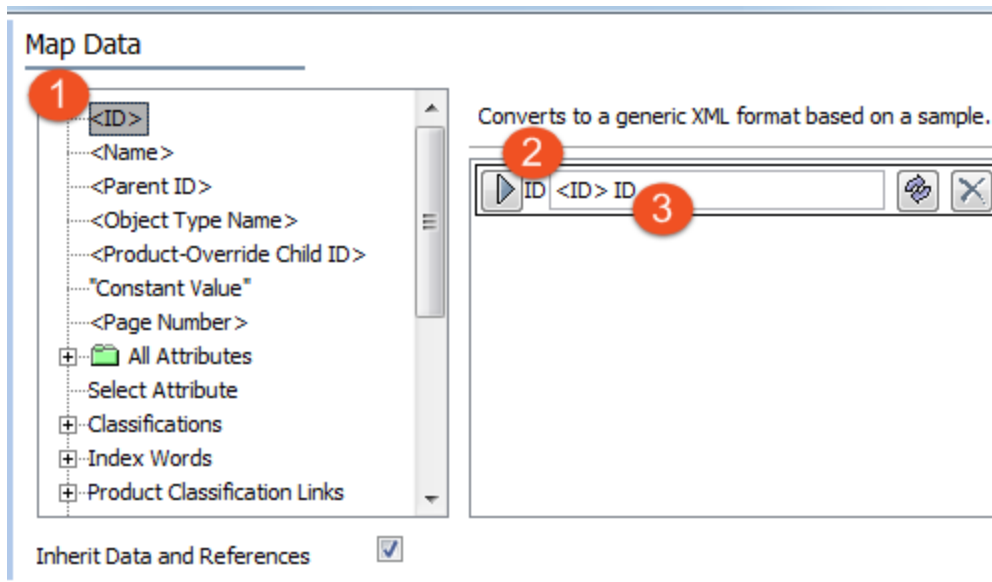
Choose **Generic XML** for **Format** and provide the following template text in the Sample field.

```

<Products>
  <Product>
    <?Record?>
    <ID><?Target?></ID>
    <Desc><?Target?></Desc>
  </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:

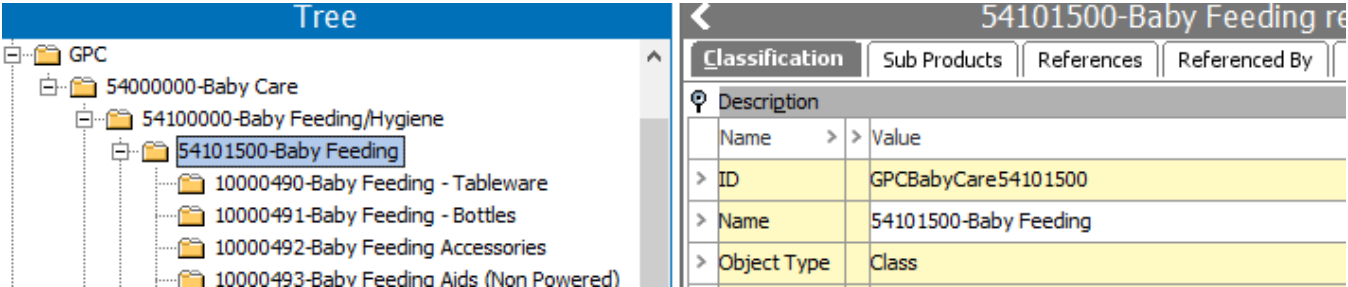
```
<Products>
  <Product>
    <ID>20803</ID>
  </Product>
  <Product>
    <ID>20805</ID>
  </Product>
  <Product>
    <ID>109308</ID>
  </Product>
</Products>
```

- 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.



Classification	
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.

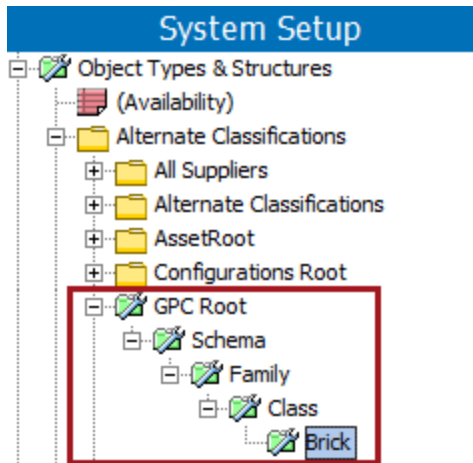
Setup Requirements

Setting up and using GPC Classifications involves the following steps:

1. Obtain the GPC download files and extract zipped data. Each segment of the current production release GPC can be downloaded from the GS1 website.
2. 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:

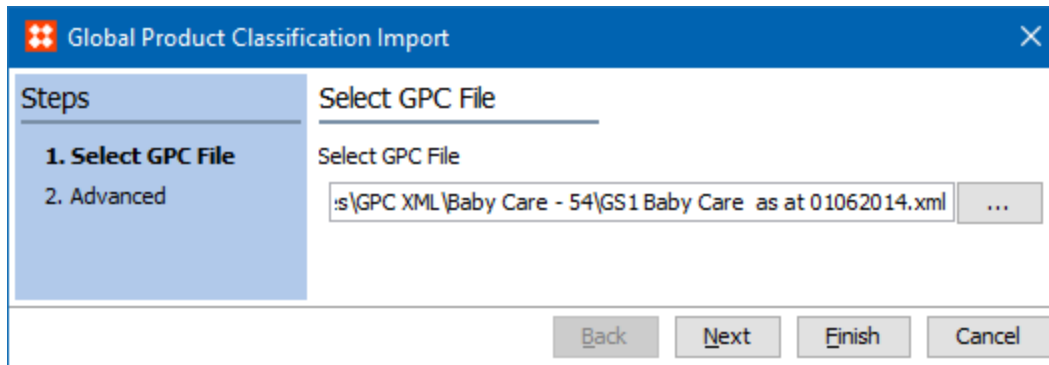
- Alternate Classification Object Types (Segment, Family, Class, and Brick)
- Alternate Classification hierarchy objects for the imported segment
- Attributes for the imported segment. Attributes will be created under the STEP standard attribute group 'GPC'.
- LOVs for the imported segment attributes. 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:

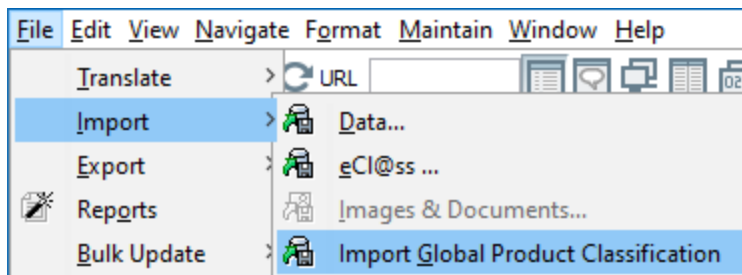
Level	GPC Name	STEP ID	STEP Name	STEP Parent Name
1	GPC (GS1 - Global Product Classification)	GPC Root	GPC Root	Alternate Classifications
2	Segment	GPC Schema	Schema	GPC Root
3	Family	GPC Family	Family	Schema
4	Class	GPC Classification	Class	Family
5	Brick	GPC Brick	Brick	Class

Global Product Classification Import

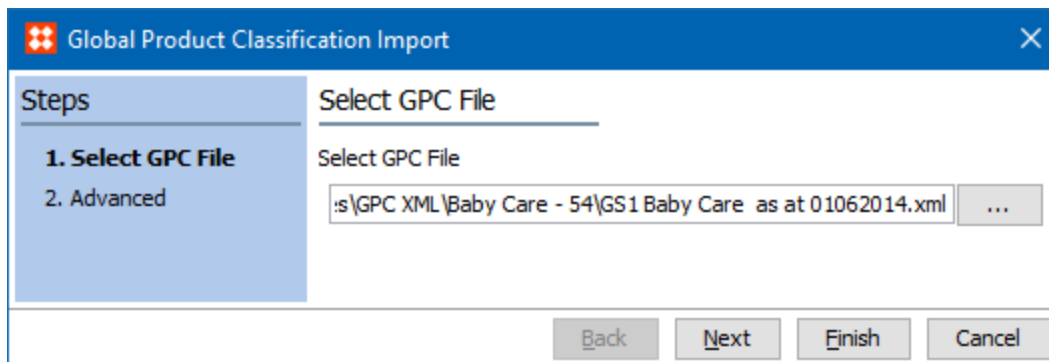


Global Product Classification Import

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:

The screenshot shows the 'Global Product Classification Import' dialog box with the 'Advanced' step selected. The 'Steps' pane on the left lists '1. Select GPC File' and '2. Advanced'. The 'Advanced' pane contains the following options:

- ID for GPC Classification Folders and GPC Attributes:**
 - ☒ Prefix with text: (text field)
 - ☐ Postfix with text: (text field)
- Name for GPC Classification Folder:**
 - ☒ Prefix with GPC Code
 - ☐ Postfix with GPC Code
 - ☐ No Prefix or Postfix
- Additional Options:**
 - ☐ Approve Import Changes

At the bottom are buttons for 'Back', 'Next', 'Finish' (highlighted with a dashed border), and 'Cancel'.

- **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 Product to Classification Link Types** section of the **Reference and Link Types** 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

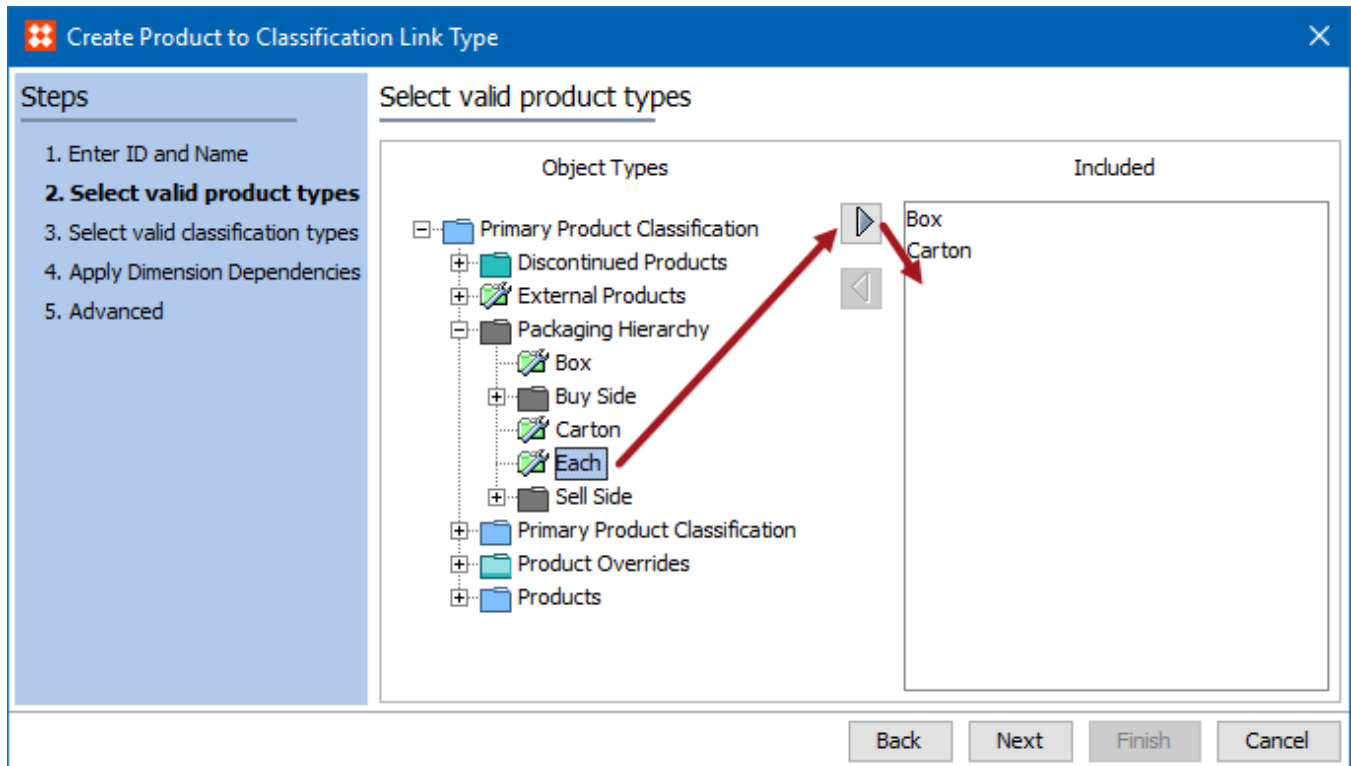
Enter ID and Name

ID: GPC

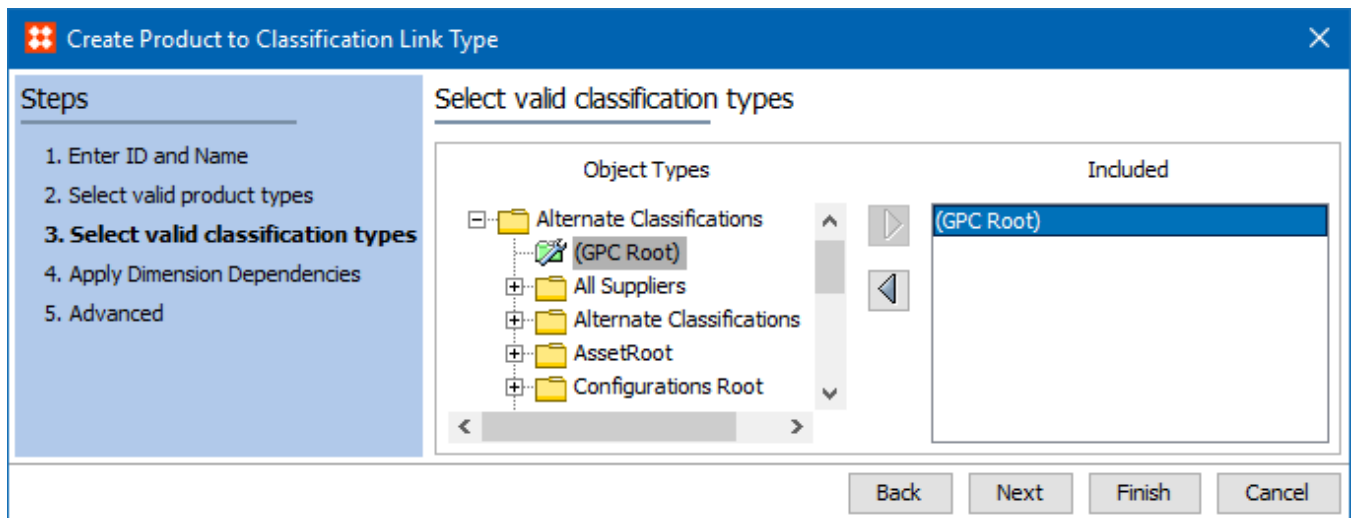
Name: GPC

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

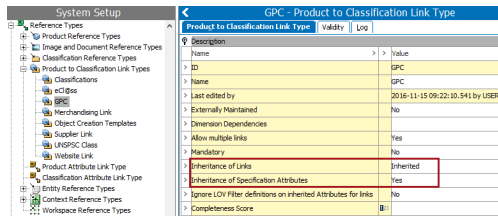
☐ Mandatory

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.



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 **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>
    </E1MARAM SEGMENT>
  </IDOC>
</MATMAS05>
```

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.
- **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

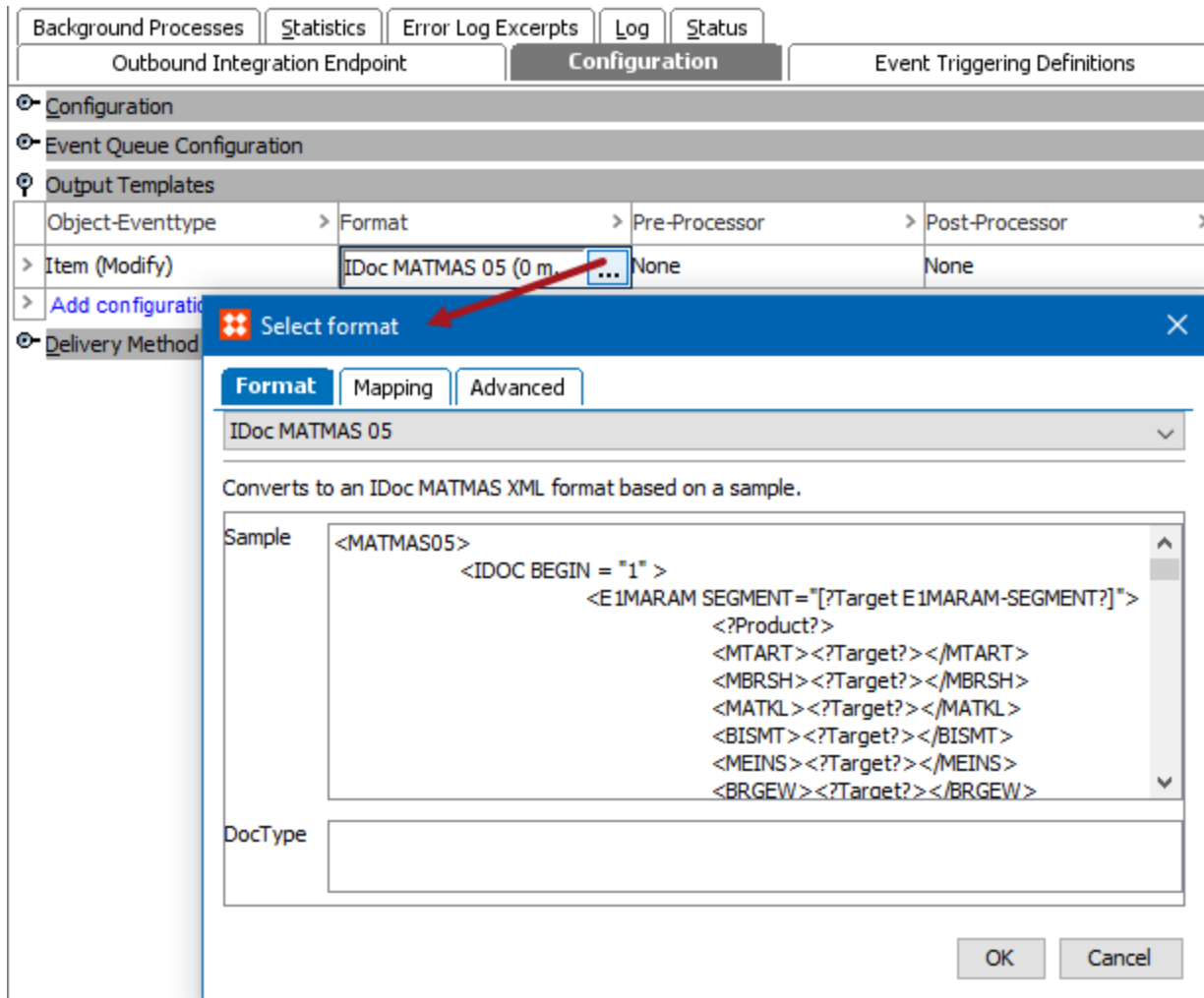
Back

Next

Finish

Cancel

OIEP



Mapping

Mapping is required to sync data from external systems with STEP data. 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 inbound 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 wizard 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 Import Manager 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 wizard 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

Header Text	Description
<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
<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 Result panel. If an auto mapping is unsuccessful, you can map that column manually.

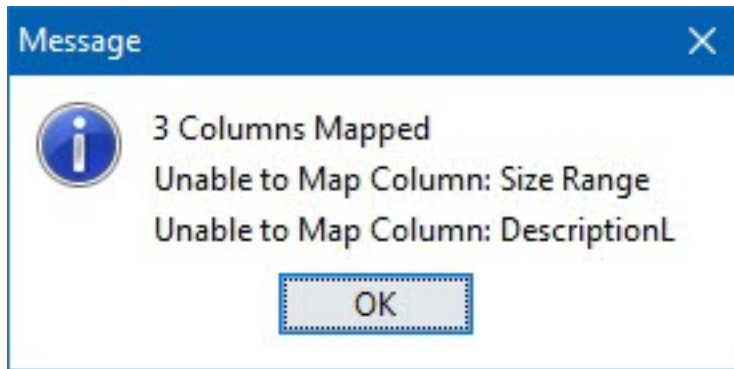
Auto Map Example

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.	

1. 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 / 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

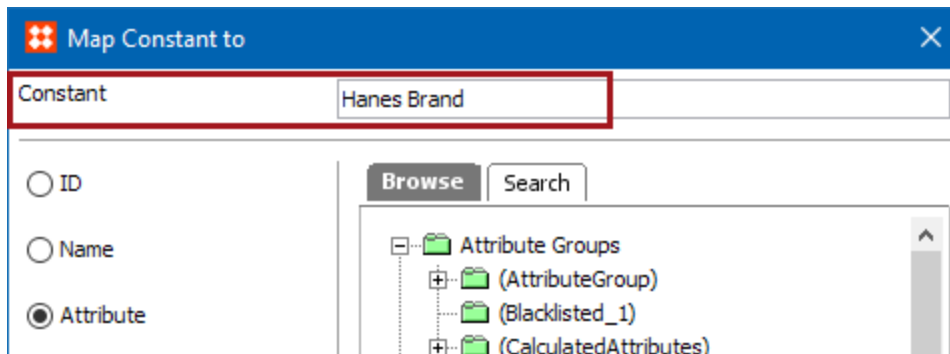
Name=<Name> ✓	Parent=<Parent ID> ✓	PrimaryColor=Primar... ✓	SecondaryColor=Se... ✓
Mens T PBO	Cotton T-Shirts	Black	Orange
Mens T PBG	Cotton T-Shirts	Blue	Green

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

For more information about the additional wizard 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" being 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 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' radio button is a 'Search' panel. This panel has a 'Browse' button and a 'Search' button. Below these is a search bar containing 'Color (Color)' and a 'Search' button. Underneath is a section labeled 'Advanced' with a search icon. Below that is a table with a header 'Name' and one row: '> Color ID = Color'. At the bottom of the dialog, there is a 'Create' button, a 'Mandatory' checkbox (which is unchecked), and a 'Match value ID in LOV' checkbox (which is 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

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

Auto Map
Map
Constant
Remove
Transform

Generate Profile

For more information about the additional wizard 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.

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:

<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:

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

Map to: Product

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

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>	83.33%	
<ShortItemDescription[DK]>	83.33%	
<ShortItemDescription[UK]>	83.33%	
<Unique Key>	83.33%	

Filtering enabled

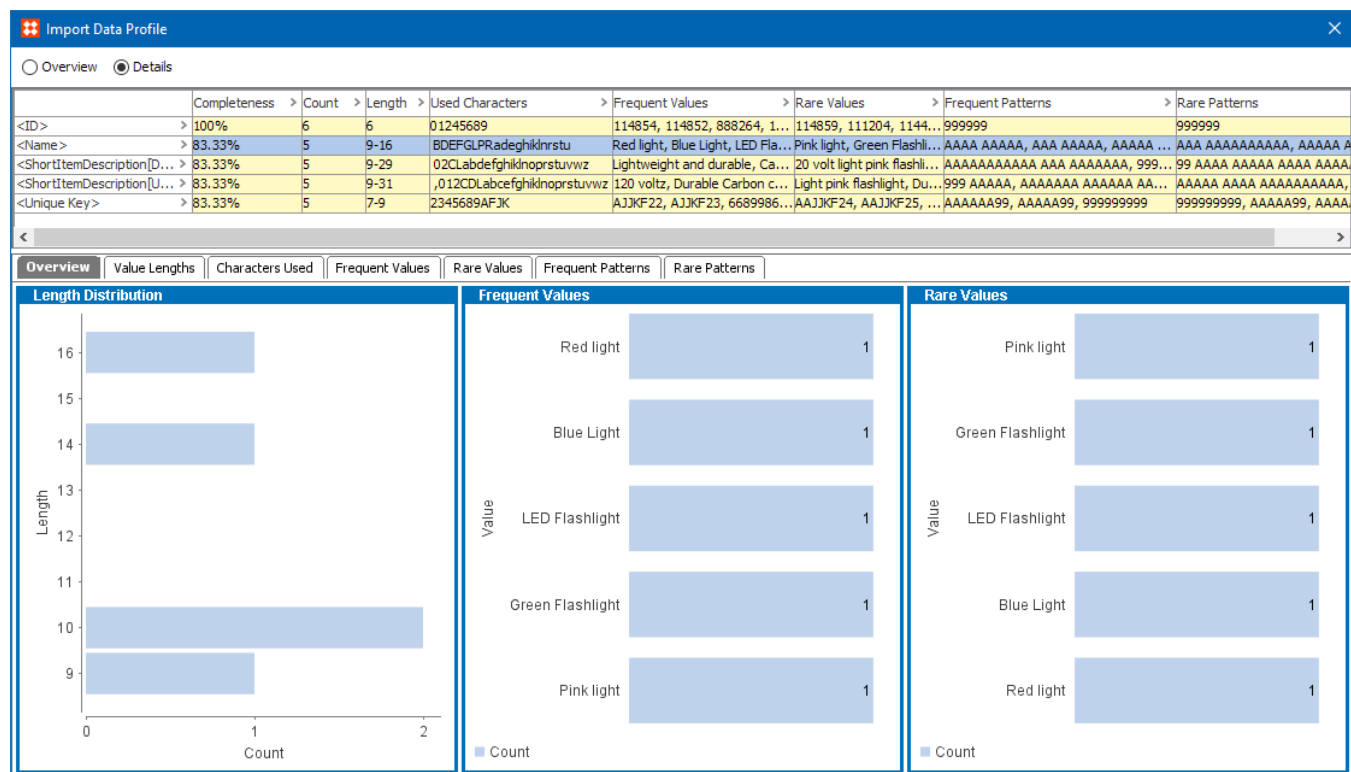
Copy table for Excel

When **Filtering Enabled** is selected, you will get a drop down list which allows you to filter the based on: Empty Values, Non Empty Values, Equals, Begins with, Ends with, Contains, '<' (less 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.

- 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.
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	

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- Transformation** 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

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

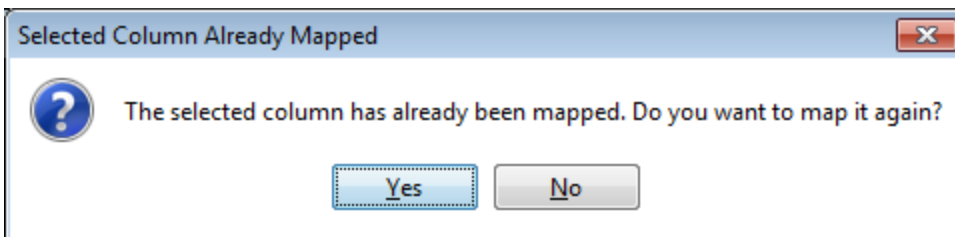


Illegal source selection: Please select cells in one column only.

OK

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.



Mapping Options

See the following topics for examples on manually mapping different kinds of data:

- Asset Reference - Map Inbound
- Attribute - Map Inbound
- Classification Reference - Map Inbound
- Data Container Attribute - Map Inbound
- Entity Reference - Map Inbound
- ID or Key - Map Inbound
- Name - Map Inbound
- Parent - Map Inbound
- Product Classification Links - Map Inbound

- Product Reference - Map Inbound
- Reference Meta-Data - Map Inbound

In addition to the one-on-one mapping of inbound data to an object in STEP, the following actions can be achieved via the Map feature:

- Change Object Type - Map Inbound
- Delete Values - Map Inbound
- Create and Map to a New Attribute in the Attribute - Map Inbound topic

Asset Reference - Map Inbound

You can reference products, classifications, or entities IDs / Unique keys listed in one column of the input file to assets IDs / Unique keys listed in another column of the input file, and make a reference between them in STEP. When you make this selection, you must specify which reference type to use. The available reference types are based on the asset reference types that exist in STEP.

Before you create additional mappings, note the following:

- 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 links from objects to classifications or to other objects or assets, you must map the product ID / Unique key in the input file.
- When making product-to-asset references, you cannot specify the asset reference type in the input file. You must select the asset reference type in the wizard screen. You can map the same column again and specify a different asset reference type in case it is needed.

Map Classification 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

Target ID Aspect: ID

☒ (BrandNameLogo)
☐ (Illustration)
☐ (InstallationManual)
☐ (MSDS)
☐ (OwnersManual)
☐ (PrimaryProductImage)
☐ (ProductImage)
☐ (Video)

☐ Mandatory

Cancel OK

If you check the 'Mandatory' box at the bottom of the wizard, row(s) will be skipped if a Target ID value does not exist in the input file.

Click the **OK** button when finished, and the mapping will show in the results section.

Attribute - Map Inbound

Mapping for attributes is available for both attributes that already exist in STEP and those that need to be created. Use the information below to map appropriately.

Map to an Existing Attribute

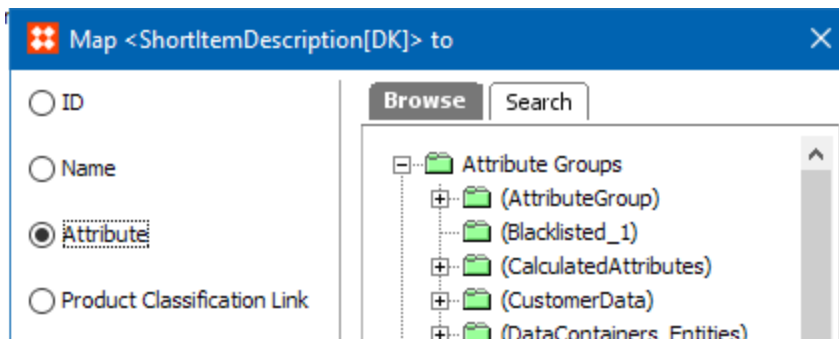
You can map a data column in your input file to an attribute that already exists in STEP. For Classification(s) you can map the data to a description attribute only.

1. In the Source section, select the column of data that you want to map to an attribute and click the **Map** button.

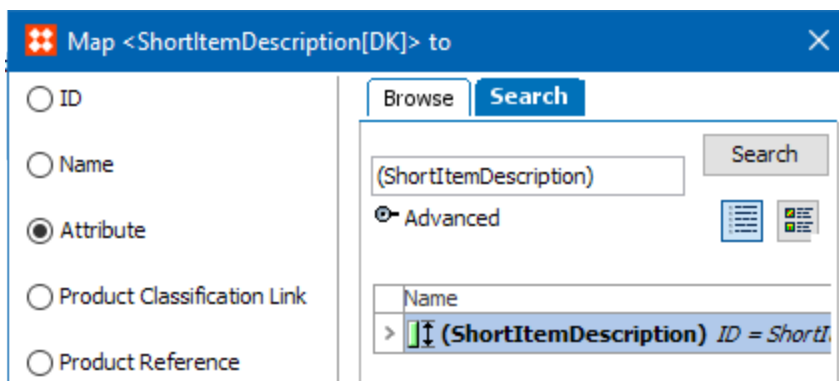
Source:

<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** window, 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. Check the **Mandatory** option to import only the objects that have a value in the file. If unchecked, all rows are imported, even those without an attribute value.



☐ Add child to override

☒ Mandatory

Cancel OK

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:

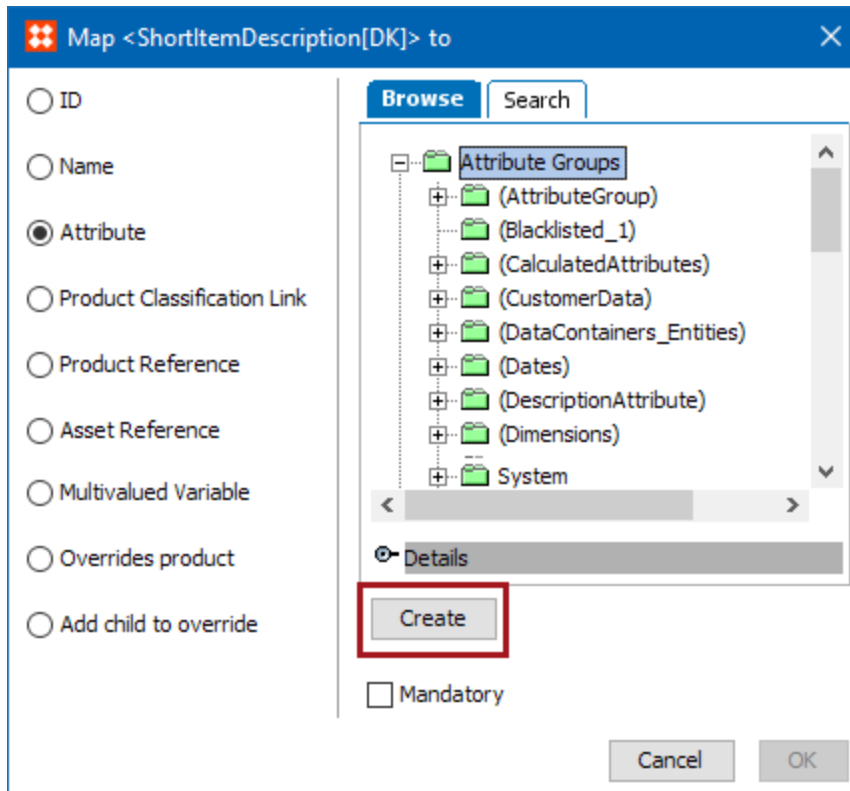
Map to: Product

ID = <ID> ✓	Name = <Name> ✓	Parent = <Parent ID> ✓	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

Create and Map to a New Attribute

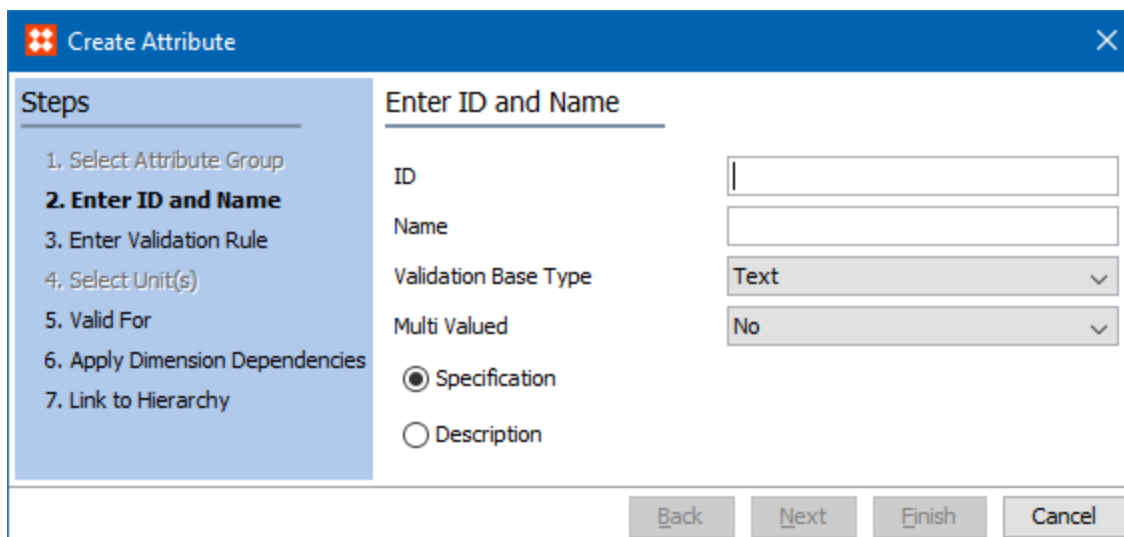
When a data column in your input file is an attribute that does not yet exist in STEP, you can create the new STEP attribute during the import process, and then map the data column to it.

1. 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.

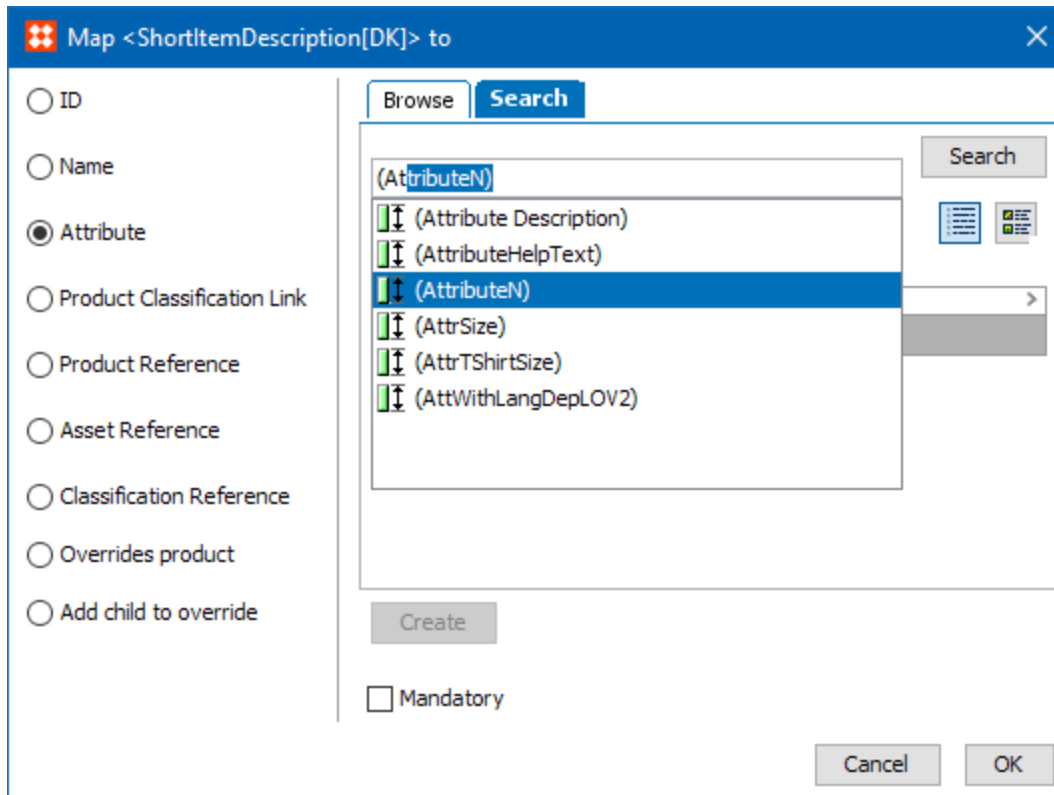


2. In the Create Attribute window, provide required information for the new attribute. For details on how to create an attribute, see the **Creating Attributes** topic in the **System Setup** documentation.

Note: Only description attributes are valid for classifications



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.



5. Click **OK** to map this new attribute to the previously selected data column.

Classification Reference - Map Inbound

You can reference products, classifications, or entities IDs / Unique keys listed in one column to entities IDs / Unique keys listed in another column in the input file, and make a reference between them in STEP. When you make this selection, you must specify which reference type to use. The available reference types are based on the entity reference types that exist in STEP.

If you want to link objects to a classification, verify that the object and classification exist in STEP, and that the reference type is valid for the object types in the input file. You must also verify that you have specified the classification ID in the input file, and not the name.

Before additional mappings are created, note the following:

- When mapping references, there must be at least two columns in the 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 mapping these columns, you can specify the reference type that should be using 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 during the import.

Map <Name> 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

Target ID Aspect

ID

☐ Mandatory

Cancel

OK

An example of how it input a classification reference is below:

A	B
<ID>	Classification ID
18210	18454
18212	18454
18213	18454
18216	18454

1. In the Source window, select the ID column and map the ID column.
2. Select the Target Column in the Source section and then click the **Map** button.

Map Data

Source:

<ID>	> Classification ID
18210	18454
18212	18454
18213	18454
18216	18454

Result:

Map to: Product

ID = <ID> ✓
18210
18212
18213
18216

Auto Map

Map

Constant

Remove

Transform

Generate Profile

- In the 'Map To' window, select the Classification Reference radio button option and select 'ID' in the Target 'ID Aspect' drop down list. Make sure that the desired reference type is selected.

Map Classification 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

Target ID Aspect: ID

☒ Classification Product

☐ Mandatory

Cancel OK

Note: If the input file contains the unique key, you can map via key by selecting the appropriate key in the dropdown list.

If the 'Mandatory' box at the bottom of the wizard is checked, row(s) will be skipped if a target ID value does not exist in the input file.

4. Click the **OK** button, and it will show the mapping in the results section.

Map Data

Source:

<ID>	> Classification ID >
18210	18454
18212	18454
18213	18454
18216	18454

Result:

Map to: Product

ID = <ID> ✓	ClassificationProduct ClassificationCrossReference... ✓
18210	T-shirts
18212	T-shirts
18213	T-shirts
18216	T-shirts

Auto Map

Map

Constant

Remove

Transform

Generate Profile

5. It is possible to map the same target column again to a different classification reference type if it is needed.

Change Object Types - Map Inbound

During import, it is possible to change the object type of an existing object. This might be useful during data migration, data clean-up activities, or when the wrong object type was originally assigned.

Important: When changing object types, is advisable 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.

The following conditions must be met in order to change the object type of an existing object during import:

1. 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 (shown in the image below).
3. The parent node must allow children of the new object type being selected.
4. The attribute validations must be the same between the original and the new object type.
5. The dimension dependencies must be the same between the original and the new object type.

To illustrate, consider an object which has the valid object type options of Case, Pack, and Pallet. 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 & Documents	Commercial	Tables	Proof View	Status	State Log
Product		References			
Description					
Name	>	>	Value		
ID	>		184435		
Name	>		Acme Hot Dogs pk		
Object Type	>		Pack		
Revision	>		Case		
Approved	>		Pack		
Translation	>		Pallet		
			Not translated		

To create the import file, we first export the object that needs to have a new object type. This ensures we have the correct ID (or unique key) and parent. Although this example uses Excel, we could use any format that is valid for both inbound and outbound exchange. The only required data is the object ID and the Object Type. In this example, we also included the name (to easily identify the object being edited) and the parent ID.

To reparent the object in addition to changing the object type, include the new Parent ID in the file to be imported. For this example, we left the object in its current location within the hierarchy.

	A	B	C	D
1	<ID>	<Name>	<Parent ID>	<Object Type Name>
2	184435	Acme Hot Dogs pk	SellSidePackagingRoot	Pack

Now, we update the file to include the data needed for import and save it. In this example we changed the name to reflect the object type change and also updated the object type.

	A	B	C	D
1	<ID>	<Name>	<Parent ID>	<Object Type Name>
2	184435	Acme Hot Dogs case	SellSidePackagingRoot	Case

Import the file to make the update in STEP, mapping all four columns on the Map Data step. At a minimum, you must map the ID and object type. Because the ID is mapped, the object is recognized as already existing in STEP.

Map Data

Source:

<ID>	>	<Name>	>	<Parent ID>	>	<Object Type Name>	>
184435		Acme Hot Dogs case		SellSidePackagingRoot		Case	

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

After the import finishes, the same object is refreshed and reviewed in the Tree and shows the updated object type.

Acme Hot Dogs case rev.0.5 - Product	
Images & Documents	Commercial
Tables	Proof View
Status	State Log
Tasks	
Product	References
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

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.

- In the Source section, select the column of data that is the first attribute of the data container, and click the **Map** button.

Note: In the example below, the Shipping Address.<ID> column represents the data container IDs being imported. The data container type ID column should not be mapped separately as it will be included as part of the attribute configuration just described.

Source:

<ID>	>	<Name>	>	Shipping Address.<ID>	>	Shipping Address.City	>	Shipping A...	>	Shipping
CUS_114563		Customer D		ShippingAddress_Kenne		Kennesaw#Toronto#Paris		United States...		US#CA#
ADD_120042		3550 George ...								
ADD_135924		Axel Kiers Vej...								
ADD_184139		Stibo Canada								

2. In the **Map to** dialog, select the **Data Container Attribute** radio button.
3. Click the ellipsis button (...) for **Data Container type** and select the data container type associated with the attribute column being mapped.
4. Select the **Attribute** using the drop down. Only attributes valid for the data container type mapped in the previous step will display.
5. For **Data Container ID column**, use the drop down to select the header of the data container ID column applicable to the data container attribute being mapped.
6. 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 Address.City (shown) and Shipping Address.State (not shown) will share the same Data Container ID.

Map Shipping Address.City 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 (ShippingAddress) ...
 Attribute: City
 Data Container ID column: Shipping Address. <ID>
 Data Container separator: #

☐ Mandatory

Cancel OK

7. Check the **Mandatory** option to import only the objects that have a value in the file. If unchecked, all rows are imported, even those without an attribute value.
8. 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 text to remove values for attributes that have already been populated. Keep in mind the following points:

- The delete command is case-sensitive and must be included in the import file as: **[delete]**
- If the value being deleted was inherited, the result is not a blank field, but restoration of the inherited value.

- Only the value of an attribute is deleted, and not the object itself.

The delete function works for both tabular and XML formats and 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 description would be removed, for the mapped column.

```

23 <Product>
24   <Products>
25     <ID>SU4-SalesItem-111</ID>
26     <Description>[delete]</Description>
27   </Product>
28 </Products>

```

Entity Reference - Map Inbound

You can reference products, classifications, or entities IDs / Unique keys listed in one column to entities IDs / Unique keys listed in another column in the input file, and make a reference between them in STEP. When you make this selection, you must specify which reference type to use. The available reference types are based on the entity reference types that exist in STEP.

Before additional mappings are created though, note the following:

- When mapping references, there must be at least two columns in the 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 mapping these columns, you can specify the reference type that should be using 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 during the import.

Map <Name> 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

Target ID Aspect: ID

☒ GDSN Product Subscription

☐ Mandatory

Cancel OK

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.

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

3. In the **Map To** window, 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

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Back

Next

Finish

Cancel

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** window, 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

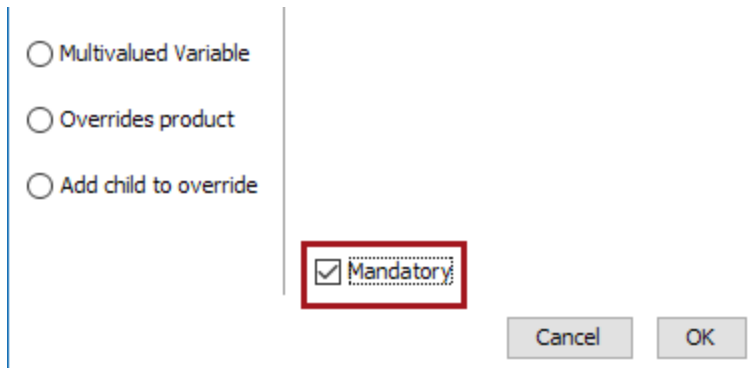
Key: Supplier Part Number

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 dialog box with a light blue border. On the left, there are three radio buttons stacked vertically: 'Multivalued Variable', 'Overrides product', and 'Add child to override'. To the right of these is a checked checkbox labeled 'Mandatory', which is enclosed in a red rectangular box. At the bottom right of the dialog 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.

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.

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
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

- In the **Map to** window, select the **Name** radio button.

Map <Name> to

☐ ID

☒ Name

☐ Attribute

- Check the **Mandatory** option appropriately:

- If **Checked**, it will require that imported objects have a name in the file. If an object does not have a name, it will be skipped and not imported.
- If **Unchecked**, all objects are imported, even those without a name.

☐ Add child to override

☒ Mandatory

Cancel

OK

- 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

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

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** window, 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:

- **Check** to require that imported objects have a parent in the file. If a parent ID does not exist for an object, then that object will not be imported.
- **Unchecked** if there are missing parent IDs and you want all objects 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

Product Classification Links - Map Inbound

You can link a product in the input file to one or more classifications in STEP. Ensure that the import file contains IDs / Unique Keys of the classifications required for linking to the products IDs / Unique Keys. The system automatically links the products using the valid product-to-classification link type. Note that each classification object type can have only one product-to-classification link type assigned to it. Therefore, the reference type is not selected during configuration as the system identifies the classifications by the provided IDs and applies the valid link type (of which only one will be available).

When using the Product Classification option, consider the following information:

- If you want to making 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.
- When mapping links from classifications to other objects, the ID / Unique key in the input file must be mapped.
- When making classification-to-classification references , there must be at least two columns in the input file. One with the ID of the source object, and the other with the ID of the target object. When mapping these columns, the reference type that should be used can be specified when linking to a classification.
- When making classification-to-classification references, it is not possible to specify the classification reference type in the input file. The classification reference type must be selected on the wizard screen, and the classification reference type should be valid for the object types in the input file. The same column can be mapped again and a classification reference type can be specified if needed.

- When making classification -to-asset references, it is not possible to specify the asset reference type in the input file. The asset reference type must be selected on the wizard screen. The same column can be mapped again and a asset reference type can be specified if needed.

Mapping Product Classification

1. In the 'Map To' window, select the **Product Classification Link** radio button.
2. In the 'Classification ID Aspect' dropdown option and select 'ID' or the specific key if mapping via a unique key.

Map (BillOfMaterials) Product Reference 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

Classification ID Aspect: ID

☐ Mandatory

Cancel OK

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 using 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.

Mapping Product Reference

1. In the Source window, select the ID column and map the ID column. For more information see the **ID or Key - Map Inbound** documentation.
2. Select the Target Column in the Source section and then click the 'Map' button.

Map Data

Source:

<ID>	Target
M1801	1801
L1802	1802
K1803	1803
J1804	1804

Result:

Map to: Product

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

3. In the 'Map To' window, select the **Product Reference** radio button option and select 'ID' in the Target 'ID Aspect' drop down list. Make sure that the desired Reference type is selected.

Note that if the input file contains the unique key, you can map via key by selecting the appropriate key in the drop down list.

If you check the 'Mandatory' box at the bottom of the wizard, row(s) will be skipped if a Target ID value does not exist in the input file.

4. Click the OK button, and it will show the mapping in the results 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...MapConstantRemoveTransformGenerate Profile

5. It is possible to map the same Target column again to a different product reference type if it is needed.

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 MapMapConstantRemoveTransformGenerate Profile

Reference Meta-Data - Map Inbound

You can import values into meta attributes on references or links. You can import meta attribute values into all types of reference types or link types.

For example, the reference tab on a product has been selected.

Tree

Primary Product Hierarchy

Products

Apparel

Upper Body Wear

T-shirts

T-shirts Items

88723-12

Cotton T-Shirts

Mens T PBO

Mens T PBG

Mens T PGS

Mens T PGW

Mens T POY

Mens T PBO w Class

Mens T PBG w Class

Product

Sub Products

References

Referenced By

Attribute Group

Discontinued Product Maintenance

Document References

Image References

Reference Type	Target
> Brand Name Logo	
> Illustration	
> Primary Product ...	107625
> Product Image	
> Video	

To import meta attribute values into the Accessory Display sequence metadata attribute Description, an input file must contain columns holding the metadata attribute values, the Product IDs, and the referenced target IDs.

<ID>	Target	sequence
134854	20695	2
134887	20695	5
179682	20695	1
129666	20695	3

In the Source window, select the ID column and map accordingly. For more information see **ID or Key - Map Inbound** documentation.

Map Data

Source:

<ID>	Target	sequence
134854	20695	2
134887	20695	5
179682	20695	1
129666	20695	3

Result: Map to: Product

Next select the meta attribute values column ' sequence' and click the map button. On 'Map References ID to' select the **Reference Meta-Data** radio button and fill out the following fields.

© Stibo Systems - Confidential - STEP Trailblazer 8.1

273

☐ 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

Reference Type

(PrimaryProductImage) ...

Attribute

(ProductsGaloreAttribute) ...

Target ID Column

Classification ID v

Target ID Aspect

ID v

Match value ID in LOV

☐

☐ Mandatory

Cancel

OK

- In **Reference Meta-Data** select the desired reference type. For example, in this case the reference type 'Primary Product Image'.
- For the **Attribute** field, select the desired meta attribute. In this example the attribute 'ProductsGaloreAttribute' is selected.
- The **Target ID Column** show the columns present in the source. It allows you to select from the dropdown list the column which contains the Reference Product's IDs in the input file.
- In **Target ID Aspect** , select the ID in the Target ID Aspect Field. (If the input file contains the Unique Key, then select the respective key in the dropdown list in the Target ID Aspect field.)

- **Match value ID in LOV:** If the meta attribute holds a hard domain LOV with value IDs, the attribute column in the Excel sheet can hold the value IDs. The IDs are matched against the IDs in the LOV and values are applied based on the match.
- If you check **Mandatory**, an Excel row is skipped if a meta attribute value does not exist in the Excel sheet.

Additionally, the following attribute data is available for mapping Reference Meta-Data:

- <Attribute Type>
- <Attribute Multi Valued>
- <Attribute FullText>
- <Attribute Externally Maintained>
- <Attribute Calculated>
- <Attribute Calculated Value Template>
- <Attribute Calculated Unit Template>
- <Attribute Mandatory>
- <Attribute Completeness>
- <Attribute Default Unit ID>
- <Attribute Dimension Ref ID>
- <Attribute Object Type Ref ID>
- <Attribute Unit Ref ID>
- <Attribute Validation Base Type>
- <Attribute Validation Minimum value>
- <Attribute Validation Maximum value>
- <Attribute Validation Maximum length>
- <Attribute Validation Input mask>
- <Attribute Validation LOV ID>
- <AttributeGroup Ref ID>

Inbound Map Data - Map To

The mapping step for the Import Manager and IIEP wizards include the same options. For more information about the additional wizard steps, see **Creating a Data Import** or **Creating an Inbound Integration Endpoint**.

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

Mapping Based on Object Type

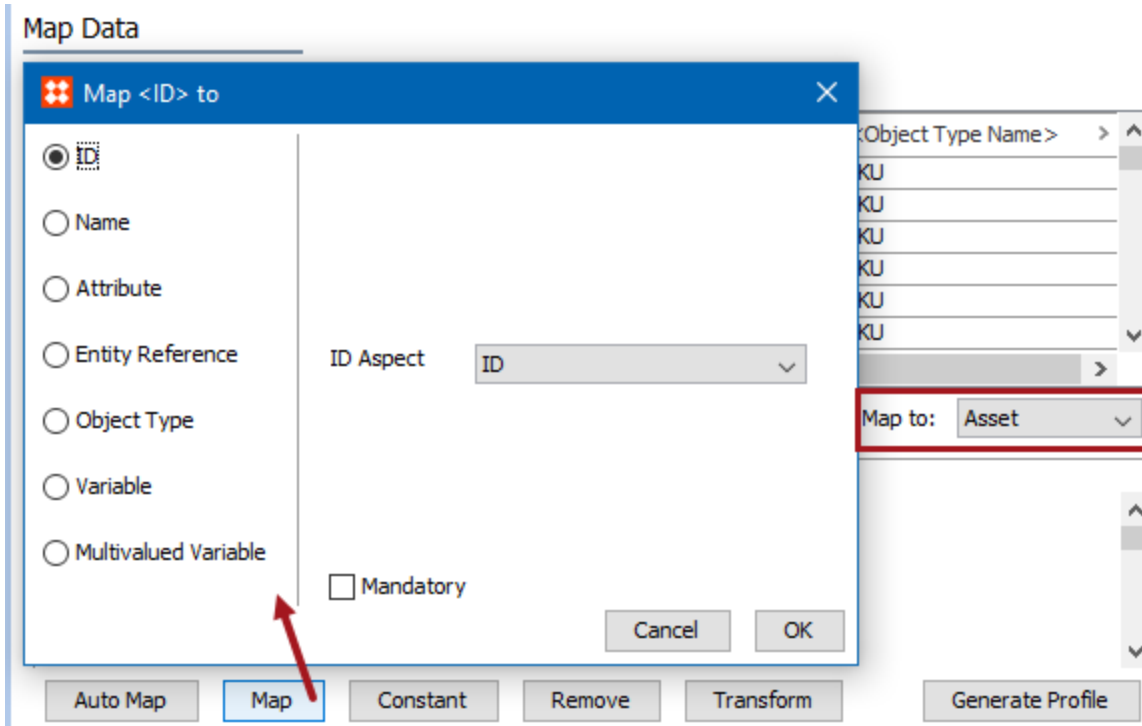
The following object types can be mapped:

- 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

When setting Map to 'Asset', then clicking the Map button, the following options are available:

- ID
- Name
- Attribute
- Entity Reference
- Parent
- Object Type
- Variable
- Multivalued Variable

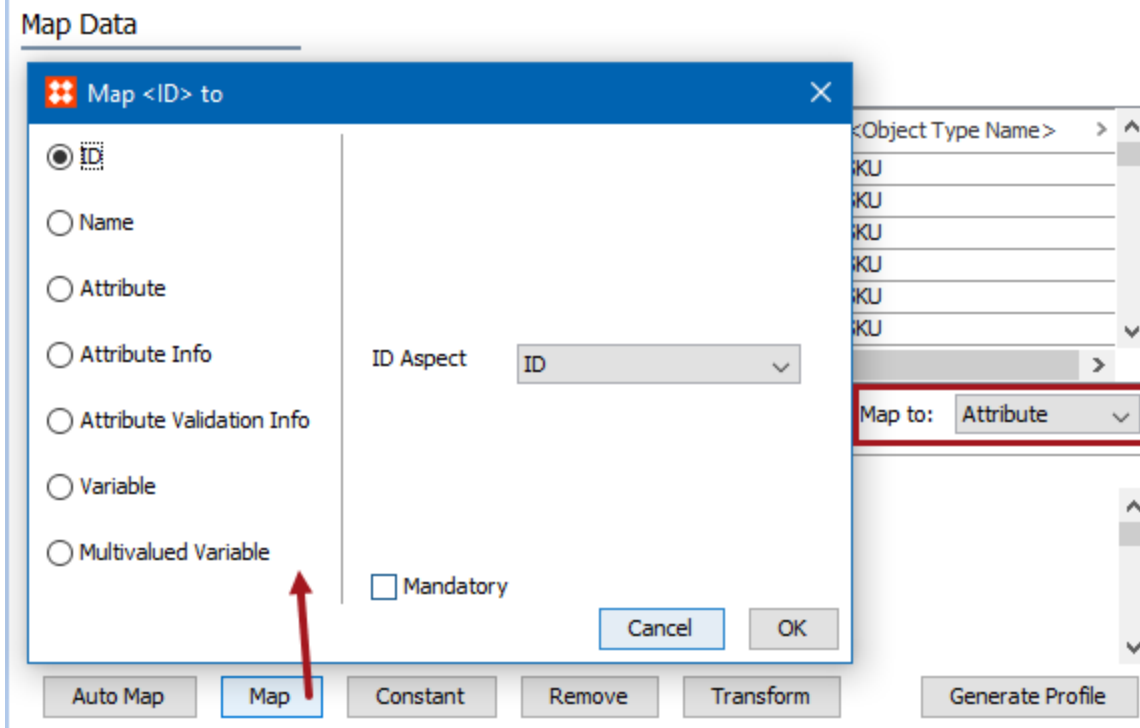


Attribute Object Type - Map To Inbound

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.

When setting Map to 'Attribute', then clicking the Map button, the following options are available:

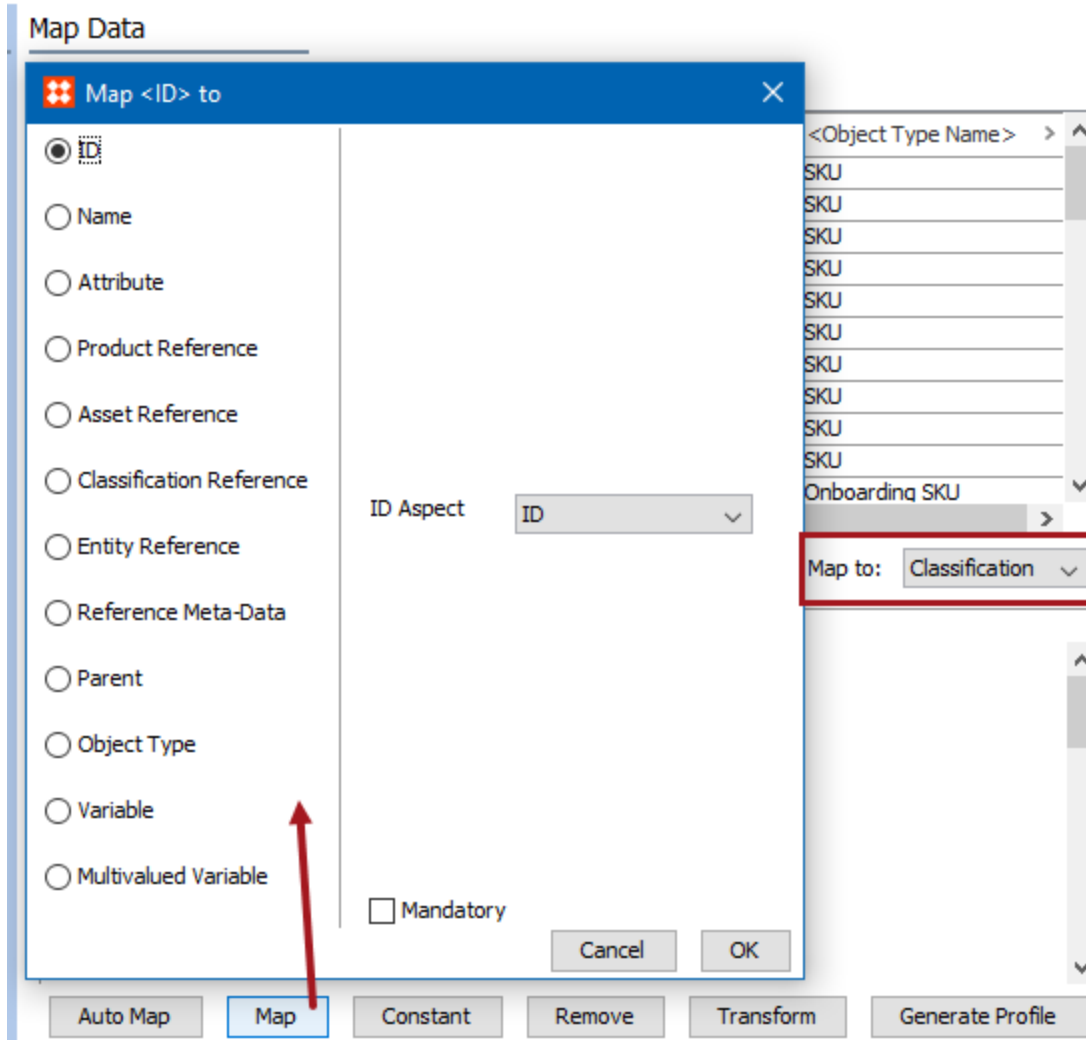
- ID
- Name
- Attribute
- Attribute Info
- Attribute Validation Info
- Variable
- Multivalued Variable



Classification Object Type - Map To Inbound

When setting Map to 'Classification ', then clicking the Map button, the following options are available:

- ID
- 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

When setting Map to 'Entity', then clicking the Map button, the following options are available:

- ID
- Name
- Attribute
- Product Reference
- Asset Reference
- Classification Reference
- Entity Reference
- Reference Meta-Data
- Parent
- Object Type

- Variable
- Multivalued Variable
- Data Container Attribute

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
☐ Data Container Attribute

ID Aspect: ID

☐ Mandatory

Cancel OK

Auto Map Map Constant Remove Transform Generate Profile

<Object Type Name>
 SKU
 SKU
 SKU
 SKU
 SKU
 SKU
 SKU
 SKU
 SKU
 Onboarding SKU
 Onboarding SKU
 Map to: Entity

Product Object Type - Map To Inbound

When setting Map to 'Product', then clicking the Map button, the following options are available:

- ID
- 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

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

☐ Mandatory

Cancel

OK

<Object Type Name>

SKU
SKU
SKU
SKU
SKU
SKU
SKU
SKU
SKU
SKU
Onboarding SKU
Onboarding SKU
Onboarding SKU
Level 3

Map to:

Auto Map

Map

Constant

Remove

Transform

Generate Profile

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 wizard 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.

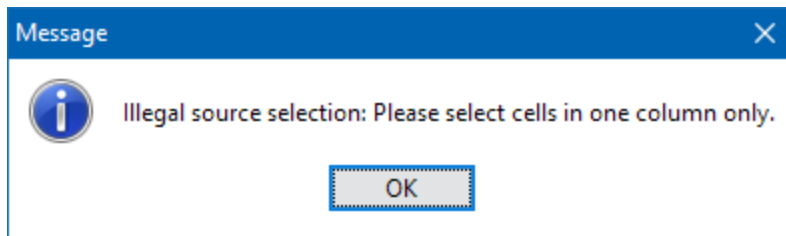
The **Result** panel no longer displays the column, and the removed data will not be imported.

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.



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 **Mapping Transformation Examples**.

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 more information about the additional wizard 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 Add Transformation Options

Transformation	Description
Append from multi-value 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-

Transformation	Description
	Valued Data Example.
Append text	Adds the specified text, such as a suffix, to the end of the value. For an example using this transformation, see Concatenate Data Using Variables Example .
Change case	Sets values to all upper case, all lower case, or mixed case. Mixed case sets the first letter of the value to upper case and the remainder to lower case. For an example using this transformation, see Split, Extract, and Convert Data Example .
Change hash function	Converts the attribute values to a cryptographic text using two cryptographic functions MD5 and SHA-1.
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: <code><StartNumerator/><bldin/>1<FractionBar/><StartDenominator/>2<EndFraction/>cm</code> You have to set up an output formatting to Indesign for <code><StartNumerator/></code>, <code><bldin/></code>, <code><FractionBar/></code>, <code><StartDenominator/></code> and <code><EndFraction/></code> to mount the fractions.
Extract characters after a given text in the value	Extracts a part of the value, starting after the occurrence of the specified text string.
Extract characters at position	Extracts a specified number of characters after a specified position in a text string. For an example using this transformation, see Split, Extract, and Convert Data Example .
Extract characters before a given text in the value	Extracts a part of the value, starting from the beginning of the text string and ending at the occurrence of the specified text string. For an example using this transformation, see Split and Extract Identifiable Data Example .
Extract characters 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.

Transformation	Description
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. For an example using this transformation, see Split, Extract, and Convert Data Example .
Format number	Extracts numerical data with the specified decimal places, and enables you to localize the number.
Generate Soundex code	Soundex code for the value. Soundex is a general algorithm which creates the same type of 4 alphanumeric code for the words with similar pronunciation.
If equals, source	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	Inserts a series of characters into the attribute value at the specified position.
Insert text before	Adds the specified text, such as a prefix, before the value.
Math - Add Source	Performs the addition operation on the value with the source. Only works for numeric values.
Math - Divide by source	Performs the division operation on the value with the source. Only works for numeric values.
Math - Multiply	Performs the multiplication operation on the value with the source. Only works for numeric

Transformation	Description
by source	values.
Math - Subtract source	Performs the subtract operation on the value with the source. Only works for numeric values.
Math Operation (+, -, *, /)	Performs the standard math operations multiplication, division, addition, and subtraction on the value.
Parse address	Splits an address contained in a single field into multiple fields using <multisep/>. This allows addresses to be normalized and validated. For example, the import of "1200 Mail Rd Hays KS 11111" should be transformed to "KS<multisep/>HAYS<multisep/>11111<multisep/>MAIL<multisep/>RD<multisep/>1200".
Prepend from source	Adds the text from the source, such as a prefix, at the beginning of the 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	Changes the specified text string with the replacement text string. All matches on the specified text are replaced.
Replace substrings of the value using a regular expression	Replaces sub-strings of text in the value with another string of text. Replaces the text on all matches or on the first match.
Replace the whole value	Replaces the specified value with a different value. You must enter the entire value. Only works on text-based attributes. For an example using this transformation, see Split, Extract, and Convert Data Example .
Replace the whole value using Lookup Table	Replaces the specified value with another value from the selected Lookup Table. For setup of a transformation table, see the Creating a Transformation Lookup Table topic in this documentation. For an example using this transformation, see Transformation Lookup Table Example .

Transformation	Description
Replace words using Lookup Table	Replaces the specified words with another value from the selected Lookup Table. For setup of a transformation table, see the Creating a Transformation Lookup Table topic in this documentation. For an example using this transformation, see Transformation Lookup Table Example .
Split and extract	Extracts data based on a specified search text string and its occurrence. 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. For 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. For 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.
Transform date by pattern	For 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. For 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.
[Transform using attribute transformation]	Extracts data based on an attribute transformation that is already defined in System Setup under Attribute Transformations. Makes it possible to apply a set of transformations specified in an Attribute Transformation to a value. The transformation takes a predefined attribute transformation as argument. For more information, see the Attribute Transformations documentation. For an example using this transformation, see Transform Using Attribute Transformation Example .

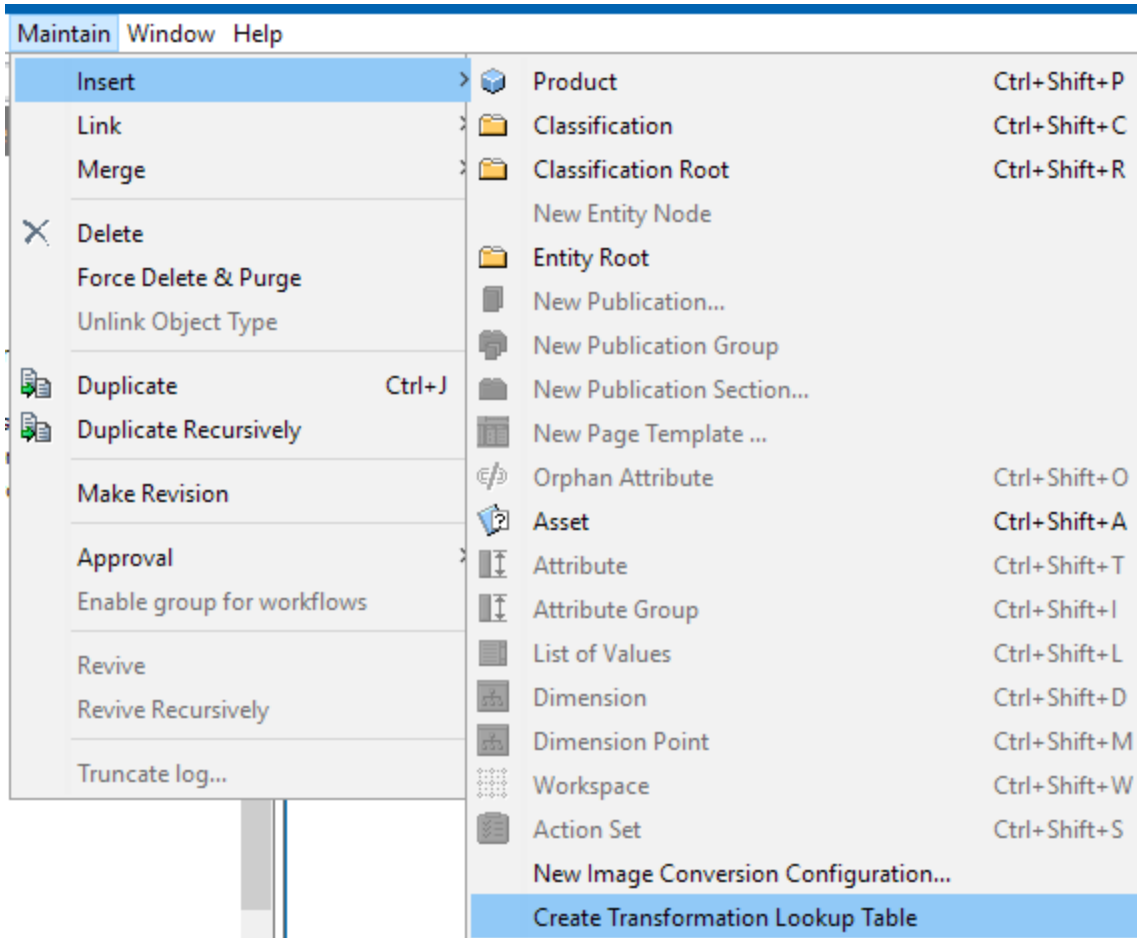
Creating a Transformation Lookup Table

It is possible to use transformations to improve the quality the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

Use transformation lookup tables when you want to transform data by replacing either specific words in values or entire values. The transformations **Replace whole value using Lookup Table** and **Replace words using Lookup Table** are available in the list of available transformations in the Import Manager wizard, when working with tables and on attribute transformations.

Create Transformation Lookup Table Objects

1. In the **Tree**, expand the **Classification Hierarchy**, and then select the preferred classification.
2. From the **Maintain** menu, point to **Insert**, and then choose **Create Transformation Lookup Table**.



3. In the **Create Transformation Lookup Table** dialog enter an ID and a name, and then click **OK**.
4. In the **Transformation Lookup Table** editor, define the from-to replacement pairs, and then click **Apply**.

Note: If you already have substitution pairs defined in an Excel sheet, you can use the **Import From Clipboard** button to paste the data into the table.

Transformation Lookup Table 1 rev.0.1 - Transformation Lookup Table

Transformation Lookup Table	
Description	
Name	Value
ID	TransformationLookupTable1
Name	Transformation Lookup Table 1
Object Type	Transformation Lookup Table
Revision	0.1 Last edited by USERL on Tue Feb 07 10:47:10 EST 2017
Approved	Never Been Approved
Translation	Not Translated
Path	Classification 1 root/Suppliers/Products Galore/Batches/Transformation Lookup Table 1
(Asset Object Type)	Transformation Lookup Table
Asset Keywords	abc
Calculated Asset File Name	TransformationLookupTable1-Transformation Lookup Table 1
Name Frequency Max Value	123
Name Frequency Min Value	123

Lookup Table

☐ Replace with default value when no matches are found (Value Substitution only):

☐ Ignore Case

From	To
Colour	color
colours	colors
flavour	flavor
flavours	flavors

Add Row

5 Rows

Import From Clipboard

Apply

Above the substitution table there are two options:

- Replace with default value when no matches are found (Value Substitution only)** - Only select this option when you use the Transformation Lookup Table to replace entire values. When selected, you can specify a default replacement value that will be used if the values you are transforming do not match any of the entries in the **From** column. For example, if you are importing values into a List of Values and want to make sure all values match entries already in the LOV.
- Ignore Case** - If selected, case is ignored when values or words in the data to be transformed are matched with entries in the From column. In this example, 'Colour', 'COLOUR' and 'colour' are all replaced with 'color'.

Note: Transformation Lookup Tables are asset objects. If asset content is configured to be dimension dependent on the system, this also applies to these objects. You can therefore have different replacement pairs defined on the same object in different contexts.

See the **Transformation Lookup Table Example** topic in the **Data Exchange** for an example.

Outbound Map Data Options

The mapping step for the Export Manager and OIEP wizards include the same user interface. For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

Formats

Mapping differs slightly based on the type of data being output.

Tabular Formats

The following tabular formats require mapping for outbound data:

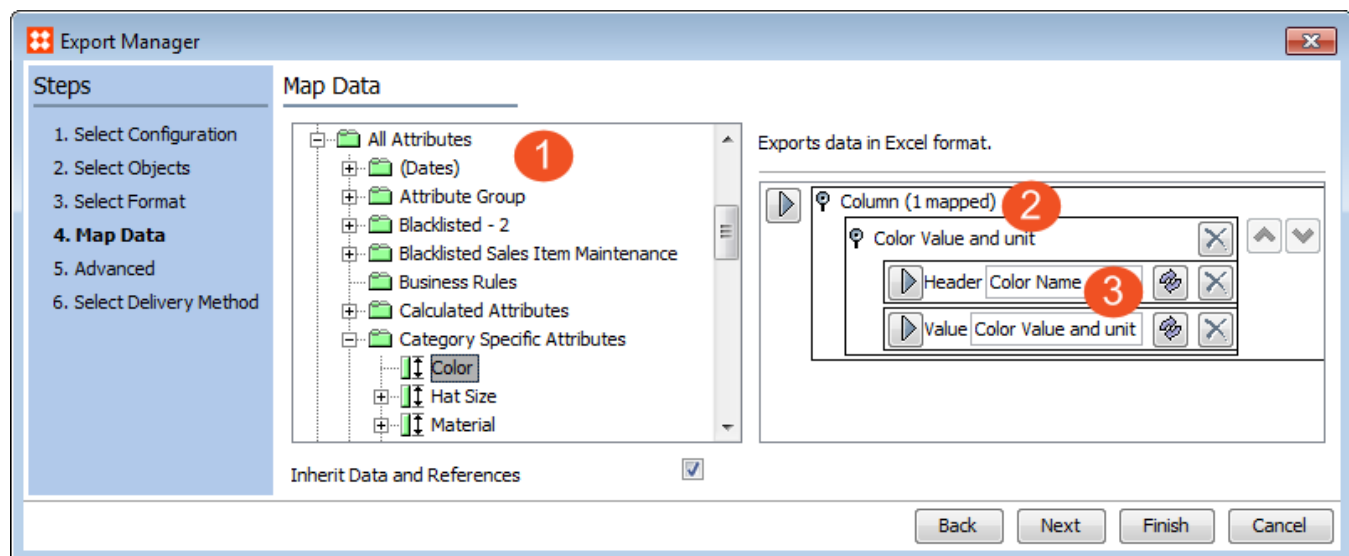
- CSV Format
- Excel Format
- Smartsheet

XML Formats

The following XML formats require mapping for outbound data:

- BMEcat Format
- BMEcat 2005 Format
- Generic XML Format
- IDoc MATMAS 05 Format

As defined below, in addition to mapping data, you can also modify data before export.



Mapping

In Map Data you specify data sources that you want to extract for each record. When the Map Data step is active in the wizard, only data that is mapped is exported. For Excel / CSV, each STEP data source mapped is displayed as a column in the output.

The numbered elements in the Map Data image above include:

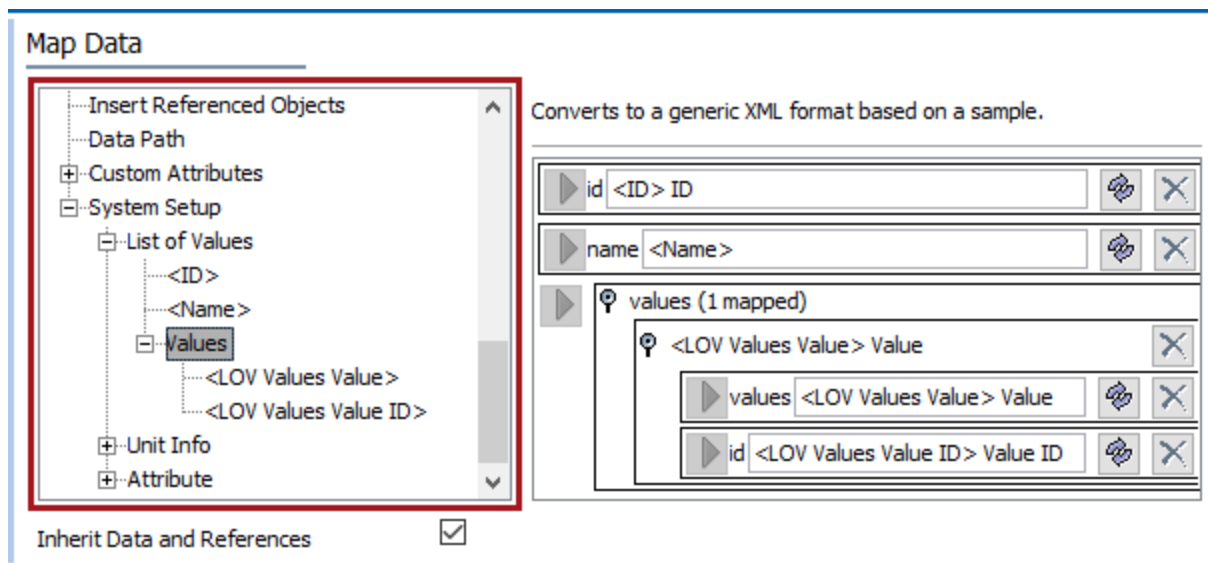
1. **Data Sources** indicate the original location of the information to be exported. The Map Data page of the wizard provides a way to associate the mapping targets with the STEP information available for the selected object types being exported. For more information on each source option, see the **Outbound 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 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:

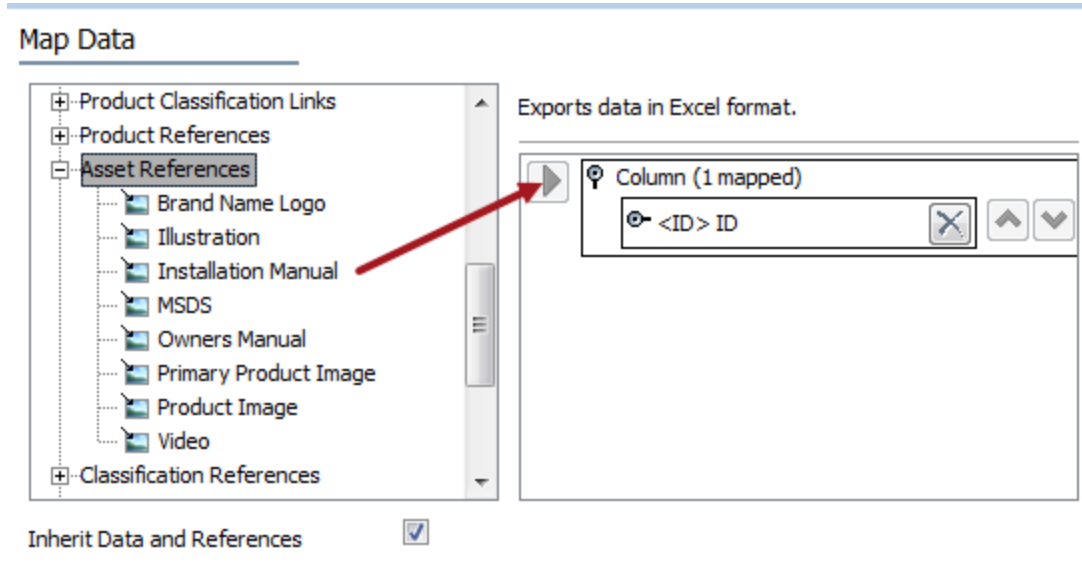
- All Attributes - Data Source Outbound
- Asset References - Data Source Outbound
- Attribute Links - Data Source Outbound or - for Generic XML and IDoc MATMAS 05 only
- Classification References - Data Source Outbound
- Classifications - Data Source Outbound
- Constant Value - Data Source Outbound
- 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
- Inherit Data and References - Data Source Outbound
- Insert Referenced Objects - Data Source Outbound
- Is Deleted - Data Source Outbound
- 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
- 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

Asset References - Data Source Outbound

Each selected asset reference is extracted into a separate field in the output. After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Asset References

1. In the left panel, expand **Asset References**, and select the desired asset reference type.

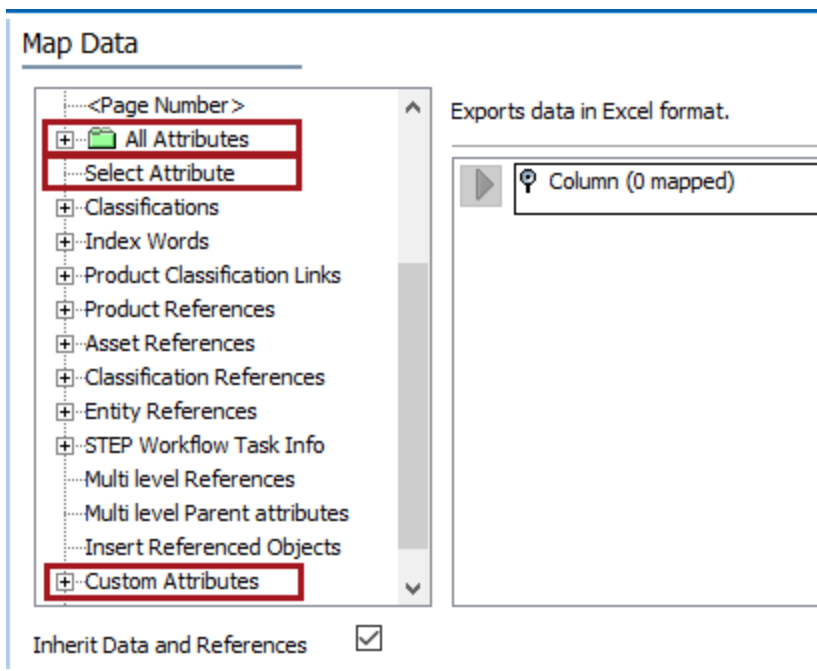


2. Click the right arrow button in the right panel to add the selected asset reference type as a column.
3. Add additional asset reference types, if needed.

Attribute - Data Source Outbound

Multiple options are available to export attributes:

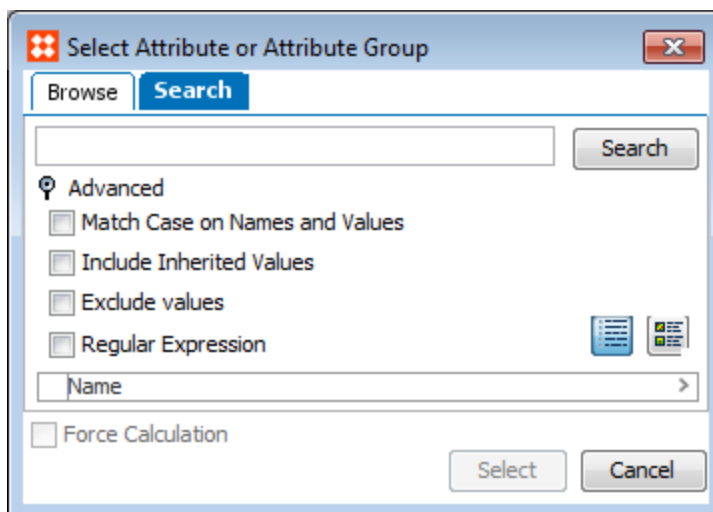
- **All Attributes** shows attributes that are relevant for the objects that are selected for export. You can select and map attribute groups in the hierarchy, allowing you to map all attributes in the group. The top level node All Attributes can also be mapped, allowing you to map all attributes relevant for the selected object(s).
- **Select Attribute** allows you to browse or search for an attribute.
- **Custom Attribute** selection is system dependent and is only available special scripts exist that operate on certain attributes.



After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Attributes

1. In the left panel, click **Select Attribute**.
2. In the right panel, click the right arrow to display the **Select Attribute or Attribute Group** dialog.

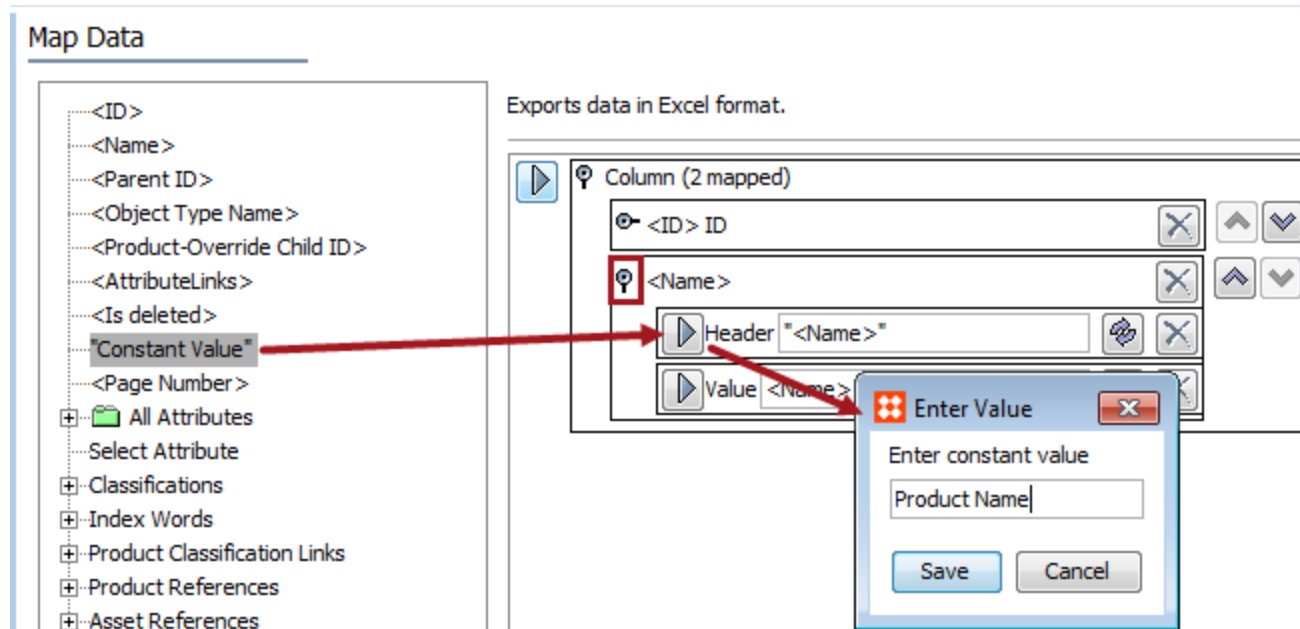


3. Enter part of the name of the attribute you want to map, and then click **Search**. Open the flipper to use the advanced search options.

4. Select the relevant attribute, and then click **Select**.
5. Repeat these steps to add all other required attributes.

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.



After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Mapping Constant Value

1. In the left panel, select **Constant Value**.
2. In the right panel, click the right arrow for the field that you want to replace.
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.

For an expanded example of mapping aspects and constants, see the **Header and Value Aspects Example** documentation.

Data Path - Data Source Outbound

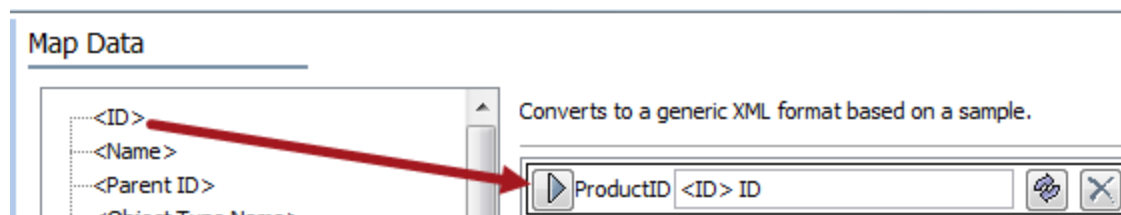
The Data Path data source is only available with the Generic XML and IDoc MATMAS 05 format. The following mapping examples use the 'Data Path' data source:

- Nestable Data Paths in Generic XML
- Nestable Data Paths in Generic XML Example
- Mapping for Nestable Data Paths in Generic XML Example

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

ID or Key - Data Source Outbound

When mapping a STEP ID, use the ID data source and connect to the mapping target intended to hold the output ID.



When a unique key is used in place of the STEP ID by systems outside of STEP, the key(s) must be mapped in during the Map Data step of an export tool. For more on unique keys, see the **Unique Keys** section of the **System Setup / Super User Guide** documentation.

Mapping ID or Key

1. In the right panel, click the flipper icon next to the ID that requires a key.
2. Under **Value**, click the **Transformation** icon .
3. In the **Transformations** dialog, select the relevant key from the **Aspect** list and click **Save**.

Map Data

The screenshot shows the 'Map Data' window. On the left, a tree view lists attributes: <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, and STEP Workflow Task Info. The 'All Attributes' folder is expanded. On the right, a section titled 'Exports data in Excel format.' shows a 'Column (1 mapped)' configuration. It has a 'Header' field with '<ID>' and a 'Value' field with '<ID> ID'. A red box highlights the 'Value' field, and a red arrow points to a 'Transformations' dialog box. The 'Transformations' dialog has 'Target: Value' and 'Source: <ID> ID'. The 'Aspect' dropdown is set to 'ID', and a list of keys is shown: ID, Key: Key 1, Key: Key 2, Key: Key 3, and Key: Supplier Part Number. The 'Key: Key 2' option is selected.

Inherit Data and References - Data Source Outbound

The 'Inherit Data and References' option shown below is a global setting that applies to all attributes selected in the Map Data window. When checked, the output includes values that are inherited from objects at a higher level in the hierarchy.

The screenshot shows the 'Map Data' window. On the left, the same attribute tree is visible. At the bottom, the 'Inherit Data and References' checkbox is checked and highlighted with a red box. On the right, the 'Exports data in Excel format.' section shows a 'Column (3 mapped)' configuration. It has three rows: '<ID> ID', '<Name>', and '<Parent ID> ID'. At the bottom, there are buttons for 'Back', 'Next', 'Finish', and 'Cancel'.

Insert Referenced Objects - Data Source Outbound

Using the 'Insert Referenced Objects' data source causes additional rows to be exported. All objects referenced from the selected reference type are exported 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.

In this example, the product being exported has three references to images:

The screenshot shows a tree view on the left with the following structure:

- Electronics
 - Televisions
 - LCD
 - P7000
 - AC-P7000-55
 - AC-P7000-65
 - AC-P7000-79
 - AC-P7000-83

On the right, the 'References' tab is active, showing 'Image References' for the selected product. The table lists references to images:

Reference Type	Image ID
Primary Product Image	107629
Product Image	107625
Product Image	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
	<ID>	<Name>	Ref Owner ID of PrimaryProductImage	Ref Owner ID of ProductImage
1	107628	107628	179626	179626
2	107625	107625	179626	179626
3	107629	107629	179626	179626
4	179626	AC-P7000-79	179626	179626

After mapping, most output options can be altered using transformations. For more information, see the **Outbound Map Data - Transform** documentation.

Is Deleted - Data Source Outbound

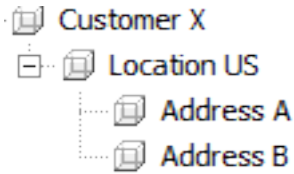
The following mapping examples use the 'Is Deleted' data source:

- Deleting Assets with a Generic XML OIEP
- Deleting Classifications with a Generic XML OIEP
- Deleting Entities with a Generic XML OIEP

Multi Level Parent Attributes - Data Source Outbound

Multi level Parent attributes enables the exporting of data on parent objects as a separate column for each object being exported.

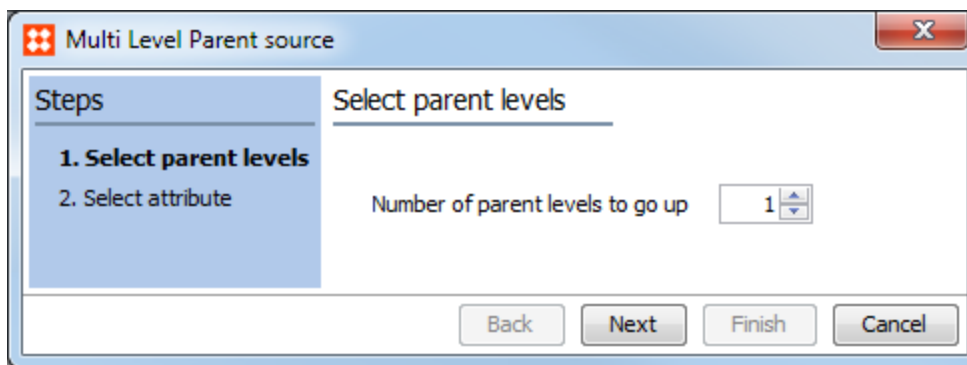
For example, a customer hierarchy is created in STEP using entity nodes. Addresses are grouped into locations.



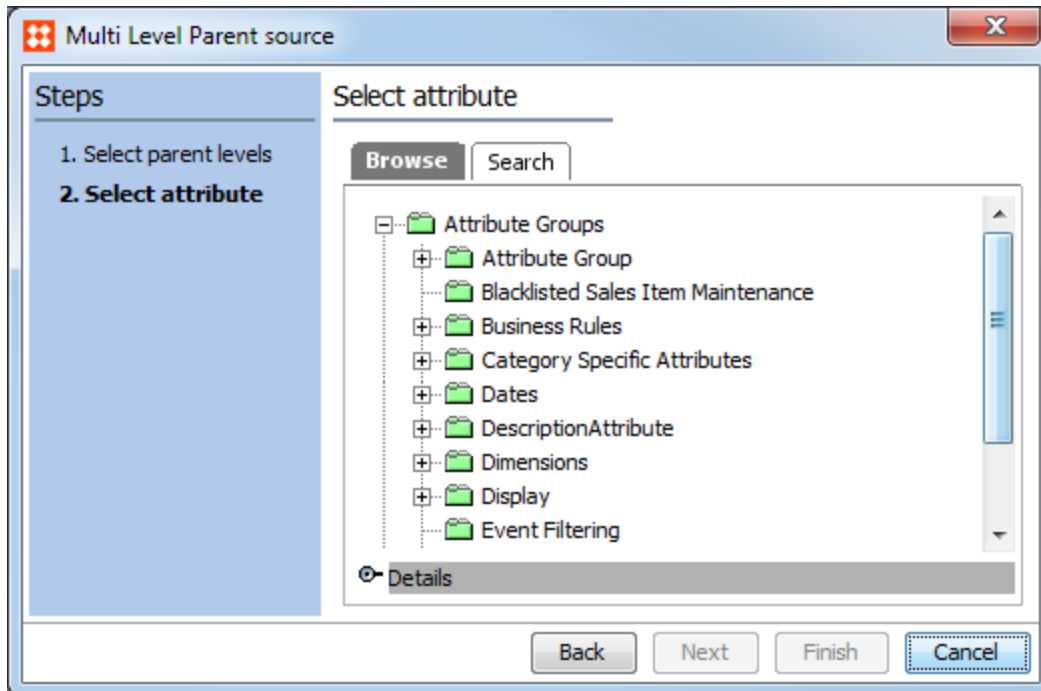
When this customer hierarchy is exported, you can export attribute values on parent nodes. This means that the row that holds Address A and Address B could also include values that only exist on the Location US or Customer X entity objects.

Mapping Multi Level Parent Attributes

1. In the left panel of the Export Manager wizard, select **Multi level Parent attributes**.
2. In the right panel, click the right arrow icon.
3. In **Select parent levels**, specify the number of parent levels to go up within the hierarchy, and click **Next**.



4. In **Select attribute**, use **Browse** or **Search** to locate the attribute to be exported, and click **Finish**.



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.

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.



The following can be exported

- Meta data attributes on references.
- Names of referenced objects.
- Attribute values on referenced objects.

Handling Multiple References

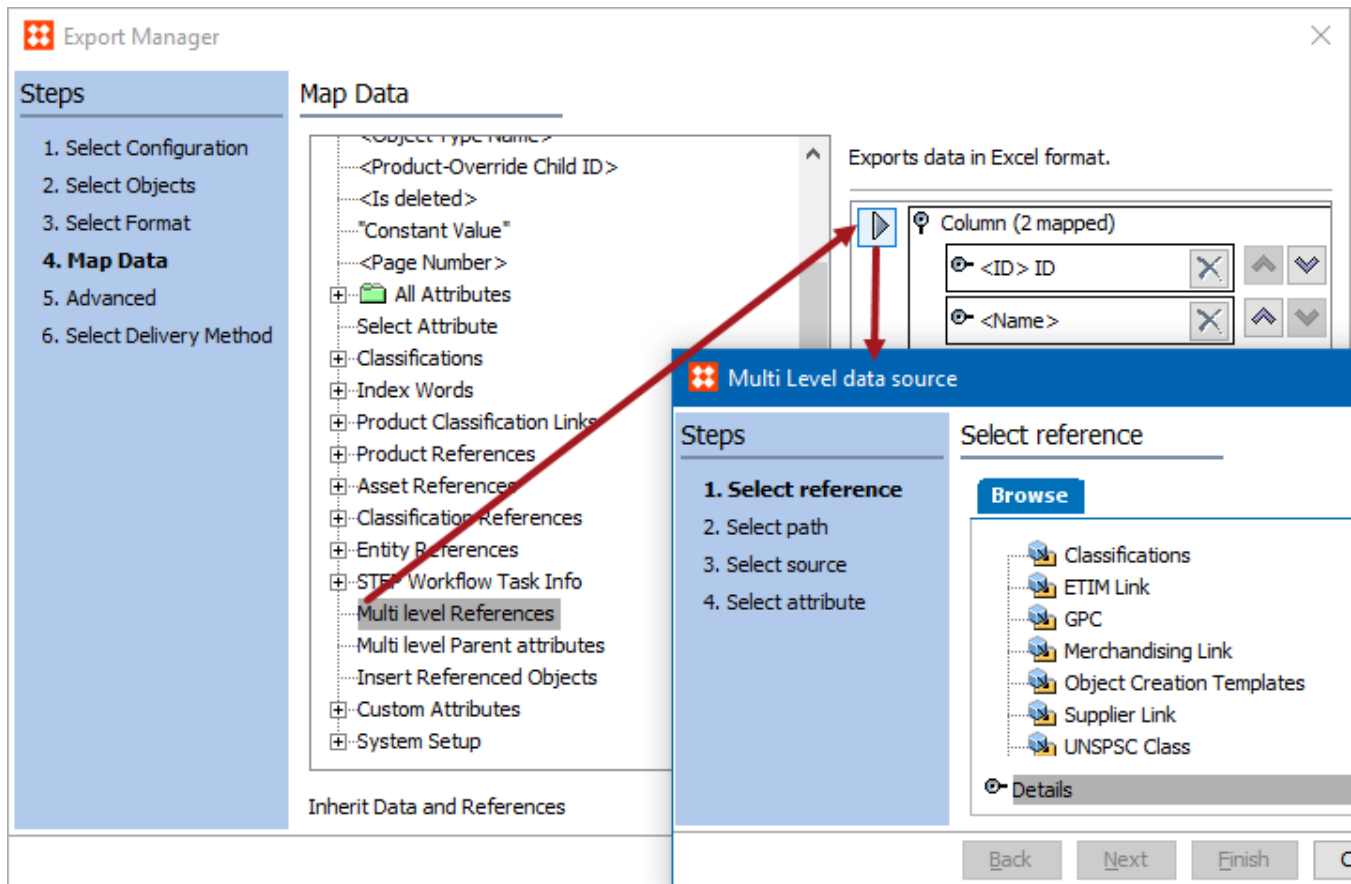
When multiple references exist, there are two options for the output:

- 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.

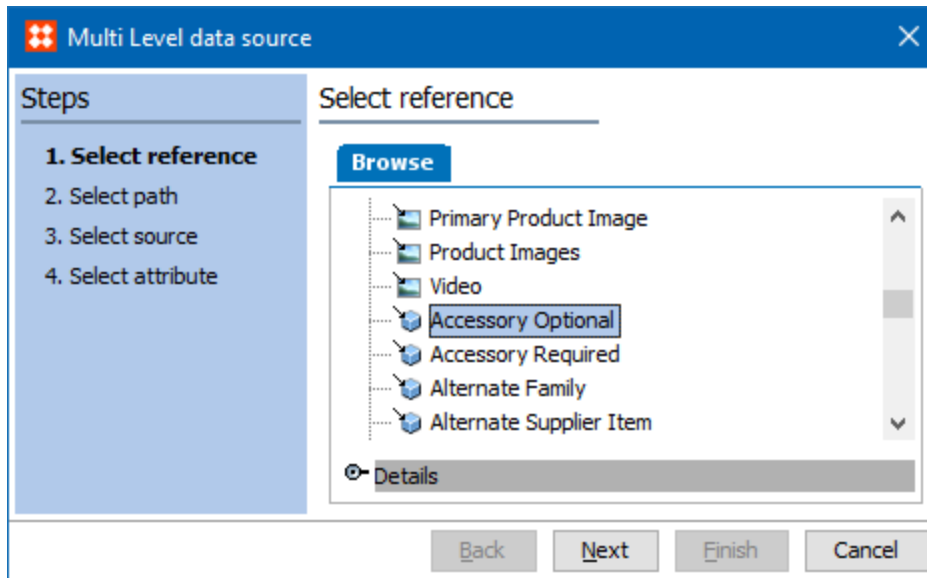
Details for using each option is described below.

Mapping Multi Level References

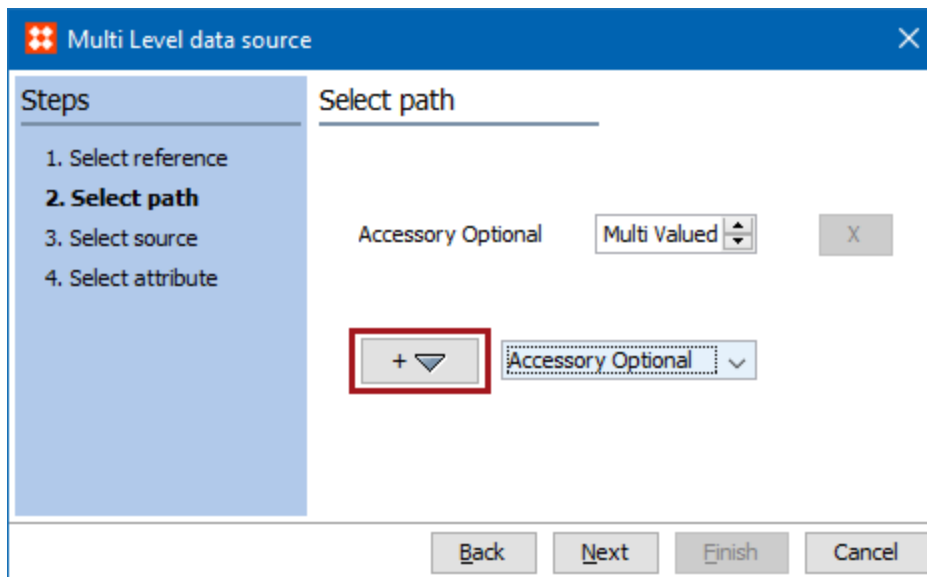
1. In the left panel of the wizard, select **Multi level References** data source.
2. In the right panel, click the right arrow icon to display the **Multi Level data source** wizard.



3. In **Select reference**, highlight the relevant source reference type and click **Next**.



4. 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, .



5. 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 

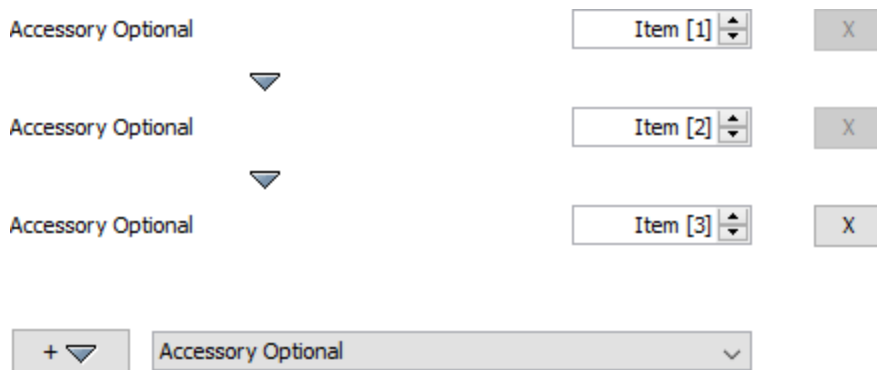
- **Multivalued** - exports data from all referenced objects of the selected type. All references are output in a single field, with a separator character.



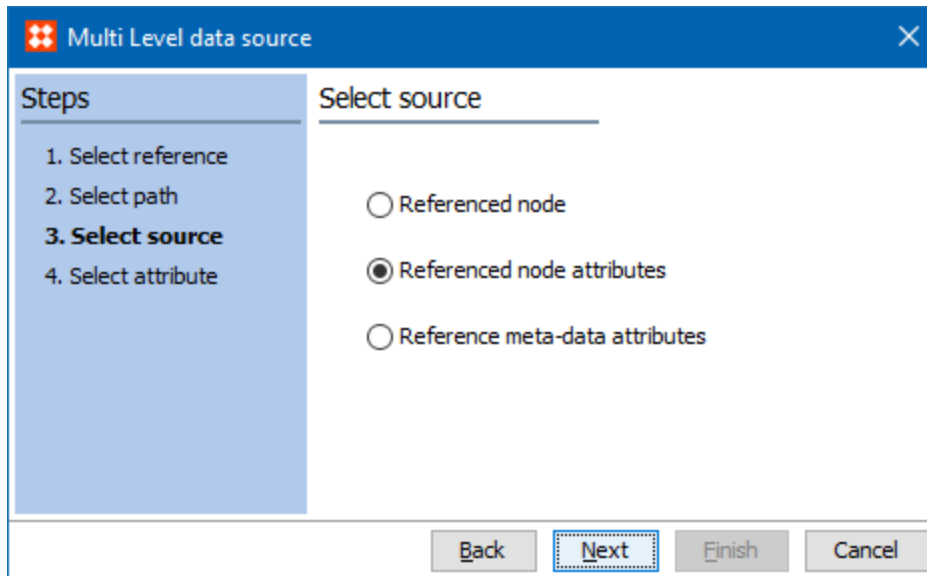
- **Item []** - exports data from only a single referenced object, determined by the selected Item number. For example, Item [1] only exports data from the first referenced object.



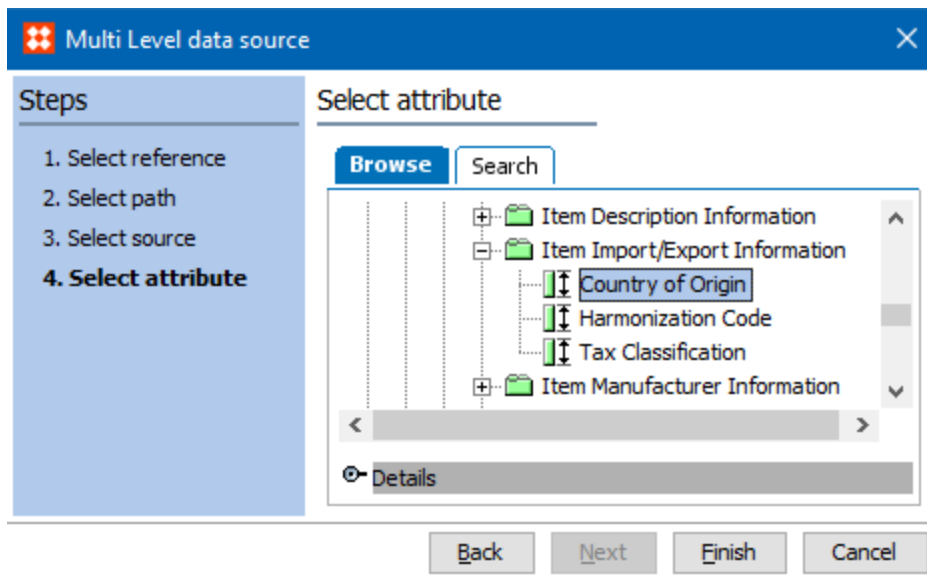
- **Item [] with multiple references** - exports multiple references in individual fields (not a single, separated field as with the Multivalued option), first determine the maximum number of possible instances for the selected reference type per referencing object. Then change Multi Valued to Item [] option and use the plus / down arrow button, , until the maximum number is displayed.



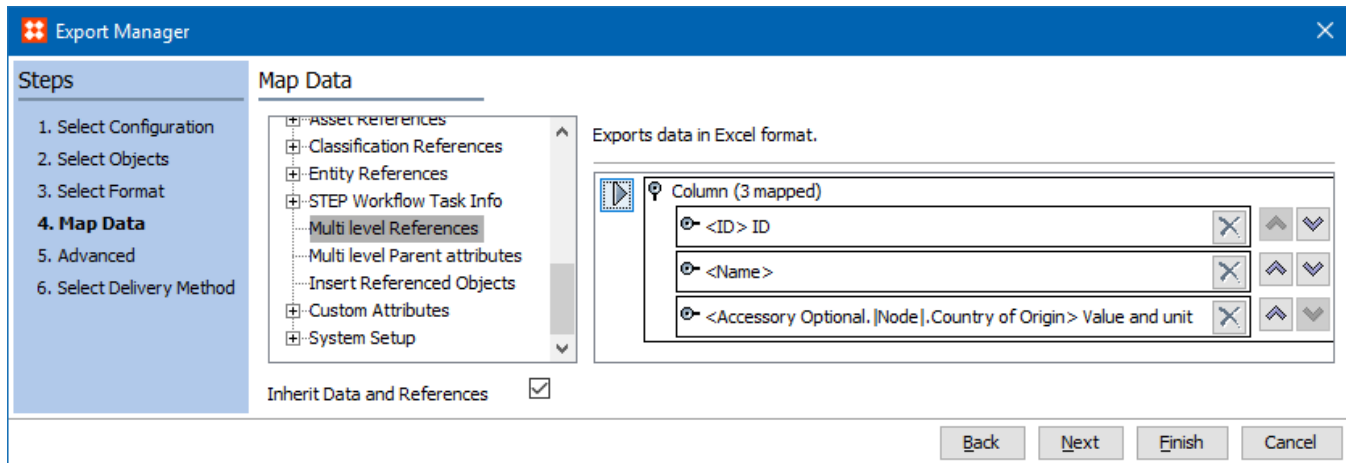
- **X** - Use the delete button to remove any unnecessary levels.
6. In **Select Source**, choose the type of data to be exported:
- **Referenced node** exports the names of the objects for the selected attribute. Click **Next** to display the **Select Attribute** screen.
 - **Referenced node attributes** 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**.



7. 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.

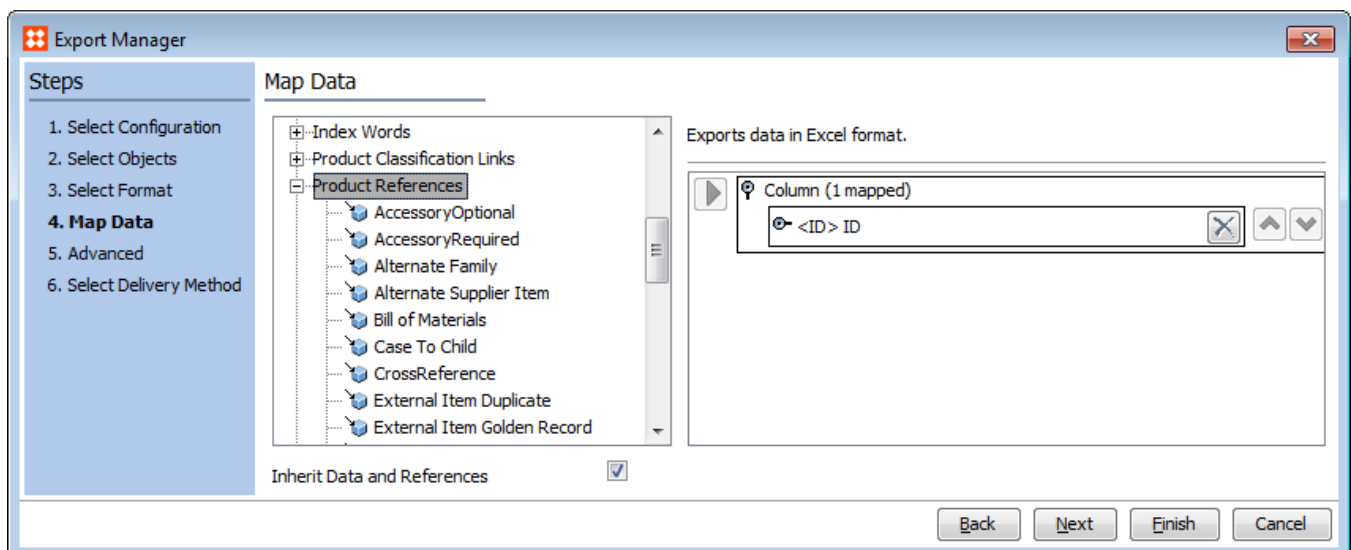


The single object output includes the values for both of the Accessory Optional references. Although this example used the Excel format, any other format would produce the same output.

	A	B	C
1	<ID>	<Name>	<Accessory Optional, Node , Country of Origin>
2	20805	Baseball Cap	SPAIN;CHINA

Product References - Data Source Outbound

1. In the left panel, expand **Product References**, and then select the preferred reference type from the list of those available in your system.



2. In the right panel, click the right arrow button to add the selected reference type as a column.
3. Add additional reference types, if needed. Each is extracted into a separate field in the output.

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

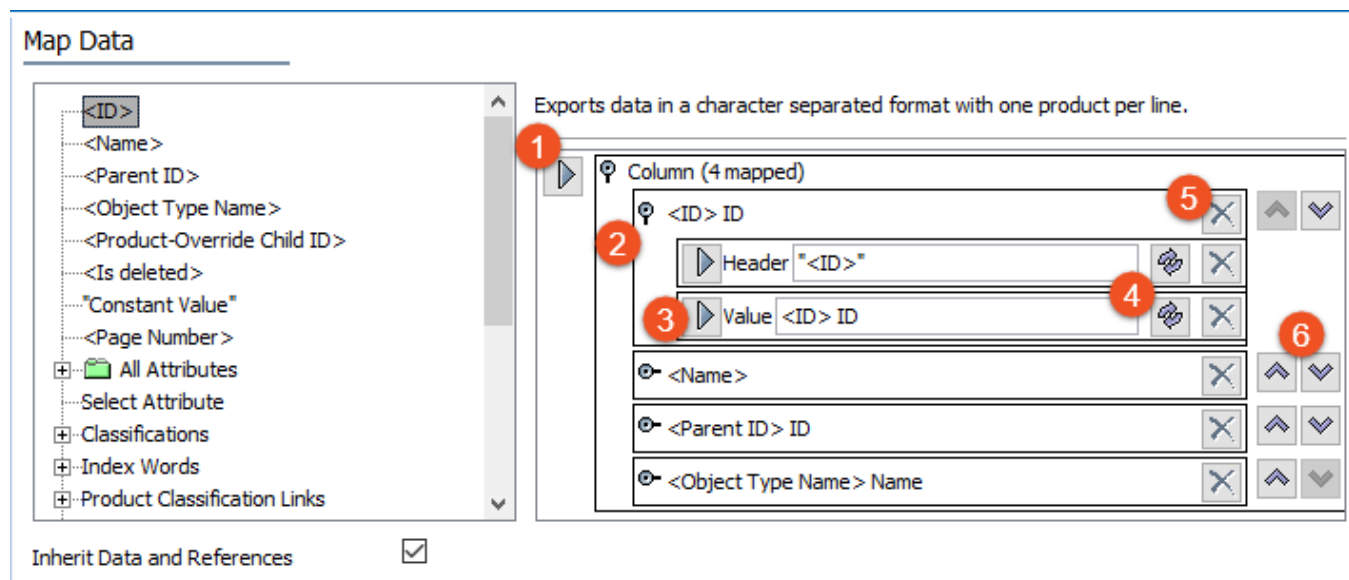
Mapping targets, displayed in the right panel, indicate the order and details of the data being exported. The process to map data differs only slightly based on the data source selected.

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.

For more information, see the **Outbound Map Data Options** documentation.

Tabular Format Targets

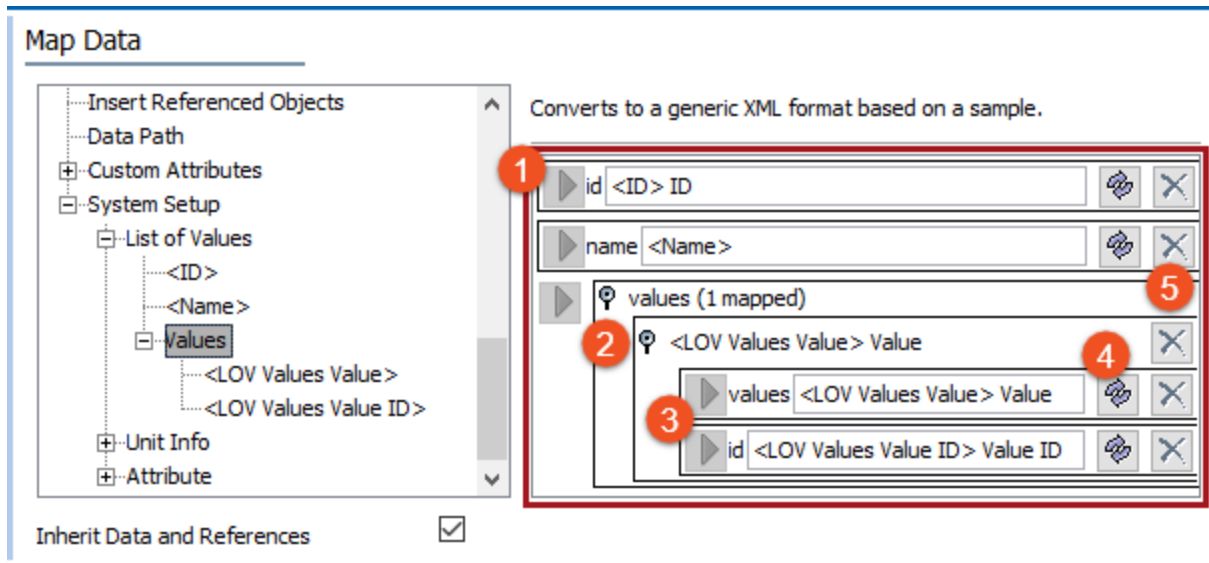
The numbered options in the following image are available for creating mapping targets for outbound data.



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.

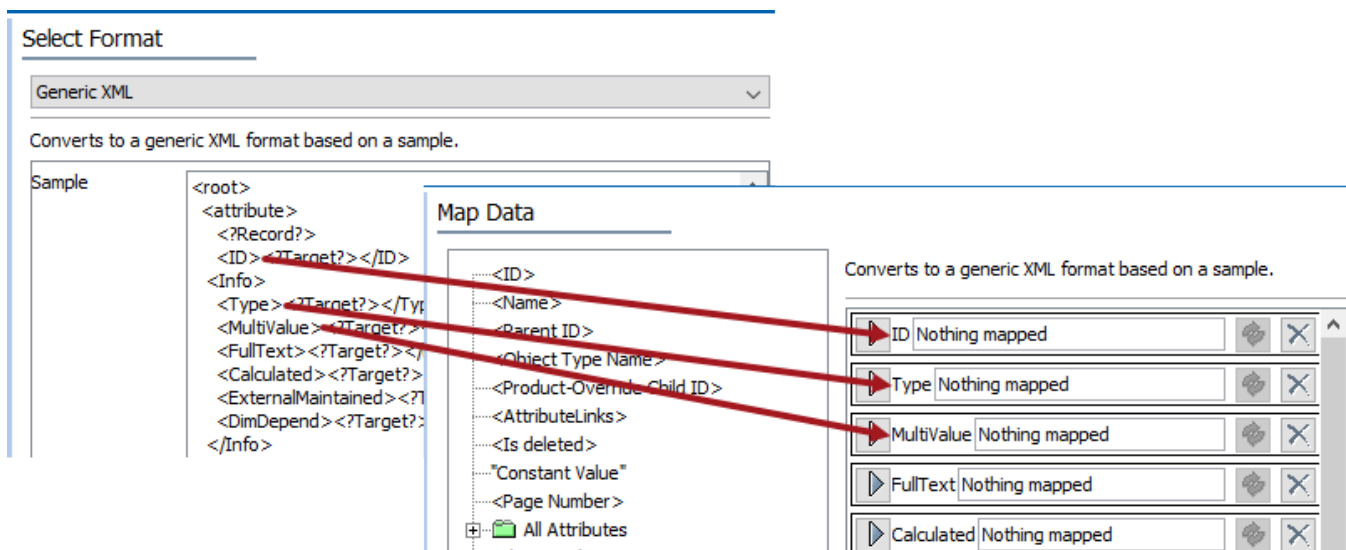
XML Format Targets

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.



Outbound Map Data - Transform

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.

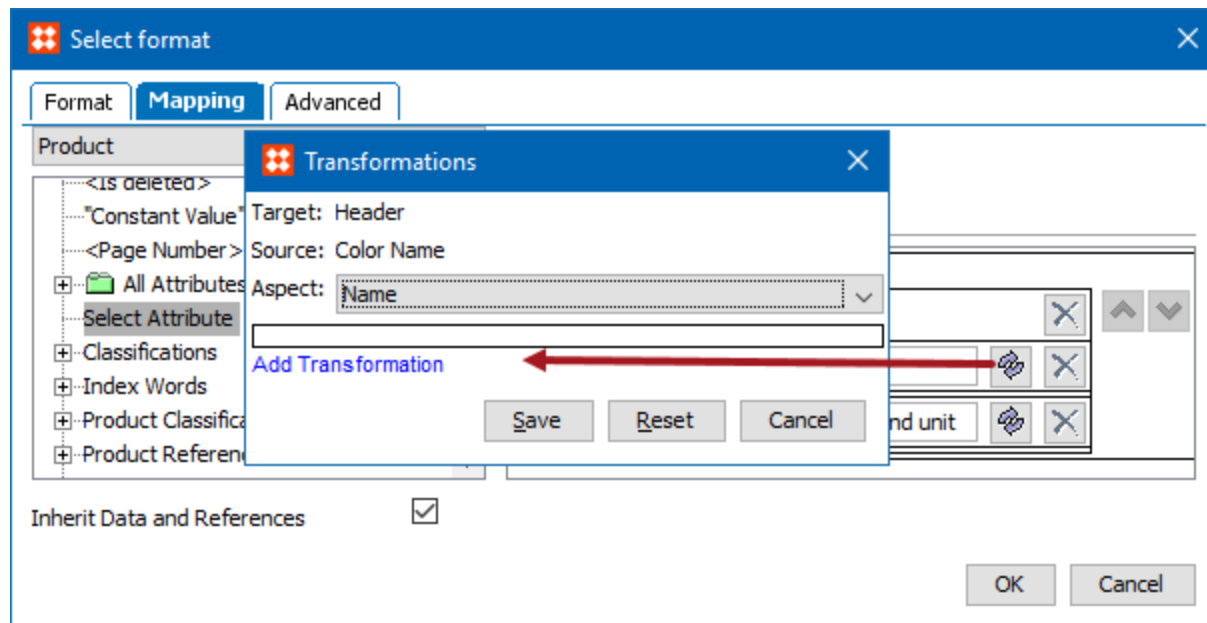
Transformations that are textual in nature and cannot be applied to attribute values that are not textual.

For examples of the most common transformations, see **Mapping Transformation Examples**.

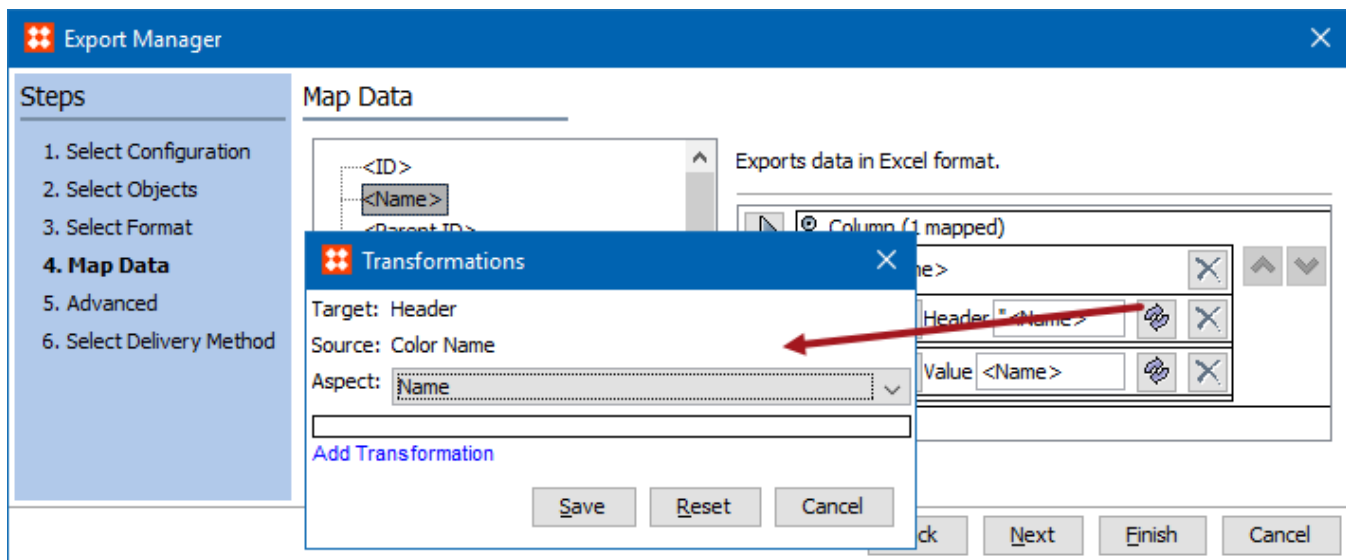
Transforming Data

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 > Output Templates flipper > edit the Format column > select the Mapping tab.



In the Export Manager, the transformations dialog is accessed the Map Data step.



The following parameters are displayed on the Transformations dialog, based on the item being transformed, and allow you to complete the transformation:

- Aspect
- Data Container Separator
- Multi-Value Separator

For more information about the additional wizard steps, see **Creating a Data Export** or **Creating an Outbound Integration Endpoint**.

Outbound Add Transformation Options

Transformation	Description
Append text	Adds the specified text, such as a suffix, to the end of the value. For an example using this transformation, see Concatenate Data Using Variables Example .
Change case	Sets values to all upper case, all lower case, or mixed case. Mixed case sets the first letter of the value to upper case and the remainder to lower case. For an example using this transformation, see Split, Extract, and Convert Data Example .
Convert fractions to parsable tags	Scans a text string for fractions. If a fraction is found, then the fraction is converted into parsable tags. A text string that contains the fraction 1/2 cm is transformed to: <StartNumerator/><bflidin/> 1<FractionBar/><StartDenominator/>2<EndFraction/>cm You have to set up an output formatting to Indesign for <StartNumerator/>, <bflidin/> ,

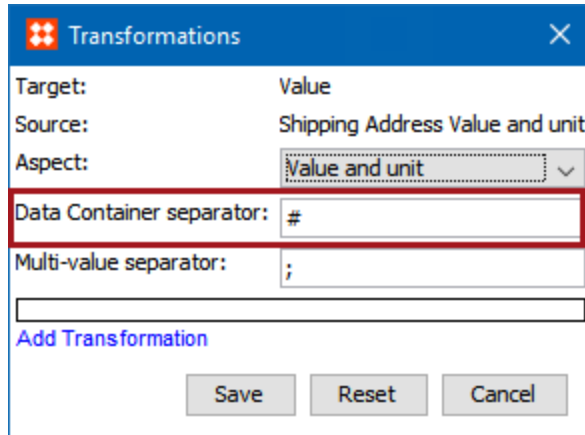
Transformation	Description
	<FractionBar/>, <StartDenominator/> and <EndFraction/> to mount the fractions.
Extract characters after a given text in the value	Extracts a part of the value, starting after the occurrence of the specified text string.
Extract characters at position	Extracts a specified number of characters after a specified position in a text string. For an example using this transformation, see Split, Extract, and Convert Data Example .
Extract characters before a given text in the value	Extracts a part of the value, starting from the beginning of the text string and ending at the occurrence of the specified text string. For an example using this transformation, see Split, Extract, and Convert Data Example .
Extract characters 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.
Extracts the last N characters of the value	Extracts a specified number of characters of a value, counting from the end. For an example using this transformation, see Split, Extract, and Convert Data Example .
Format number	Extracts numerical data with the specified decimal places, and enables you to localize the number.
Format the number with unit	Formats the number with the selected unit.
Insert text at a given position	Inserts a series of characters into the attribute value at the specified position.
Insert text before	Adds the specified text, such as a prefix, before the value.
Math Operation (+, -, *, /)	Performs the standard math operations multiplication, division, addition, and subtraction on the value.

Transformation	Description
Replace substrings of the value	Changes the specified text string with the replacement text string. All matches on the specified text are replaced.
Replace substrings of the value using a regular expression	Replaces sub-strings of text in the value with another string of text. Replaces the text on all matches or on the first match.
Replace the whole value	Replaces the specified value with a different value. You must enter the entire value. Only works on text-based attributes. For an example using this transformation, see Split, Extract, and Convert Data Example .
Split and extract	Extracts data based on a specified search text string and its occurrence. 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. Locale is determined by the context being used in workbench at the time of export.
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 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.

Data Container Separator - Transform Outbound

If a data container attribute is being mapped to a column, 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.

Important: 'Data Container separator' is only available / visible in the Transformations dialog for 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

Here is an example of how data container attribute values may look when exported. In this example, Shipping Address is the data container type and the attribute is City. The City values are separated based on the Transformations setup.

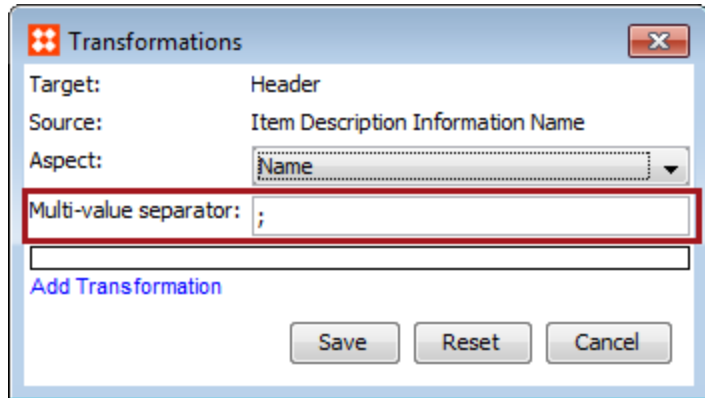
Shipping Address.City
Kennesaw#Toronto#Paris#

←

Multi-Value Separator - Transform Outbound

When multivalued attributes are mapped to a column, you can specify a character to be used as value delimiter.

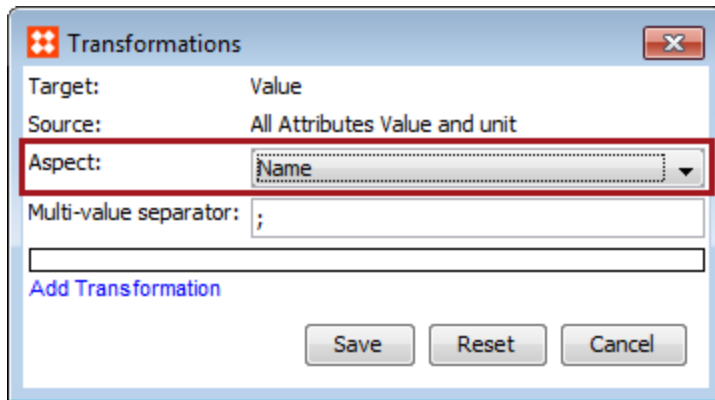
When exporting a tabular format, select a different character from the one used to delimit columns.



The screenshot shows a 'Transformations' dialog box. It has a title bar with a close button. Inside, there are labels for 'Target:', 'Source:', and 'Aspect:'. The 'Target:' is set to 'Header', 'Source:' is 'Item Description Information Name', and 'Aspect:' is 'Name'. Below these is a 'Multi-value separator:' label followed by a text field containing a semicolon (;). This text field is highlighted with a red rectangular border. Below the text field is a link that says 'Add Transformation'. At the bottom of the dialog are three buttons: 'Save', 'Reset', and 'Cancel'.

Aspect - Transform Outbound

Aspects are used to modify the data exported for the attribute and to map the same attribute multiple times while applying a different element to each occurrence. This allows you to extract and create multiple columns of different data from one attribute.



The screenshot shows a 'Transformations' dialog box. It has a title bar with a close button. Inside, there are labels for 'Target:', 'Source:', 'Aspect:', and 'Multi-value separator:'. The 'Target' is set to 'Value', the 'Source' is 'All Attributes Value and unit', and the 'Aspect' dropdown menu is open, showing 'Name' as the selected option. The 'Multi-value separator' is set to ';'. Below these fields is a text area and a blue 'Add Transformation' button. At the bottom are 'Save', 'Reset', and 'Cancel' buttons.

For example, assume the value '3' is associated with the unit 'ft'. The unit itself is associated with the base unit 'mm', so the conversion to mm for the value of 3 is 914.4. Mapping this attribute four separate times allows the following four columns of data to be exported: 3, ft, mm, and 914.4.

For an extended example of mapping aspects and constants, see the **Header and Value Aspects Example**.

Aspect	Description
Name	Extracts the name of the attribute and not the attribute value. For example, if an attribute has the ID '1234' and the name 'Weight', then the word 'Weight' is extracted per product.
ID	Extracts the ID of the attribute not the attribute value. For example, if an attribute has the ID '1234' and the name 'Weight', then '1234' is extracted per product.
Value	Extracts the product attribute value only. If the attribute has units assigned to it, and the values have units assigned to them, then units are not extracted.
ValueAndUnit	Extracts the product attribute value and the related unit (if any).
Unit	Extracts the name of the unit associated with the product attribute value but not the actual

Aspect	Description
	value.
UnitID	Extracts the ID of the unit associated with the product attribute value but not the actual value.
BaseUnitValue	Extracts the attribute value according to the base unit. Can be applied when established associations exist between units, where one unit is the base unit. For example, if the value is '3', the unit is 'ft', and the unit 'ft' is associated with the base unit of 'mm', then the value is converted from 3 ft to the corresponding number in mm.
BaseUnitName	Extracts the name of the base unit that is associated with the unit of the product attribute value. For example, if the unit 'ft' is associated with the base unit name 'mm (with the ID 'millimeters')', the output is 'millimeters'.
BaseUnitID	Extracts the ID of the base unit that is associated with the unit of the product attribute value. For example, if the unit 'ft' is associated to the base unit name 'mm' (with the ID of 'millimeters'), then the output is 'millimeters'.

These additional aspects are also available:

- LOV ID
- LOV Value-ID
- Dimension Dependencies
- Attribute Type
- Attribute Meta Data ETIM ID
- Attribute Meta Data ETIM Description
- Attribute Meta Data ETIM Feature ID
- Attribute Meta Data ETIM6 ID
- Attribute Meta Data ETIM6 Abbreviation
- Attribute Meta Data ETIM6 Description
- Attribute Meta Data ETIM6 Feature ID
- Attribute Meta Data Attribute Description
- Attribute Meta Data Attribute Help Text
- Attribute Meta Data Condition
- Attribute Meta Data Completeness Score

- Attribute Meta Data Display Name
- Attribute Meta Data Display Sequence
- Attribute Meta Data Purpose
- Attribute Meta Data Translate Item Data
- Attribute Meta Data Table Header
- Value and unit Meta Data Unit Description
- Value and unit Meta Data ETIM ID
- Value and unit Meta Data ETIM Description
- Value and unit Meta Data ETIM6 ID
- Value and unit Meta Data ETIM6 Description
- Value and unit Meta Data ETIM6 Abbreviation
- Value and unit Meta Data GDSN unit map (publisher)

Mapping Transformation Examples

Regardless if the data is being imported or exported, generally the transformation options have the same effect. 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
- Header and Value Aspects Example
- If Equals Source Example
- Merge Attributes and Constant Example
- Merge Data for Multi-Valued Data Example
- Remove or Add Units and Values
- Split and Extract Data Example
- Split, Extract, and Convert Data Example
- Transform Using Attribute Transformation Example
- Transformation Lookup Table Example

Concatenate Column with Unit Data Example

In this example, the source file is a CSV or XLS file with product data where one column has a value and another 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. Two options are available to properly load this into STEP. One is an external option, the other takes place during the mapping phase.

Using the external procedure, the two columns are concatenated in the source file. Next, the concatenated column values are saved as text. Then, the column is mapped to the attribute.

However, you can also accomplish this concatenation using the following procedure in the mapping / transformation phase as illustrated below.

- 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 mean 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.

Units

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:

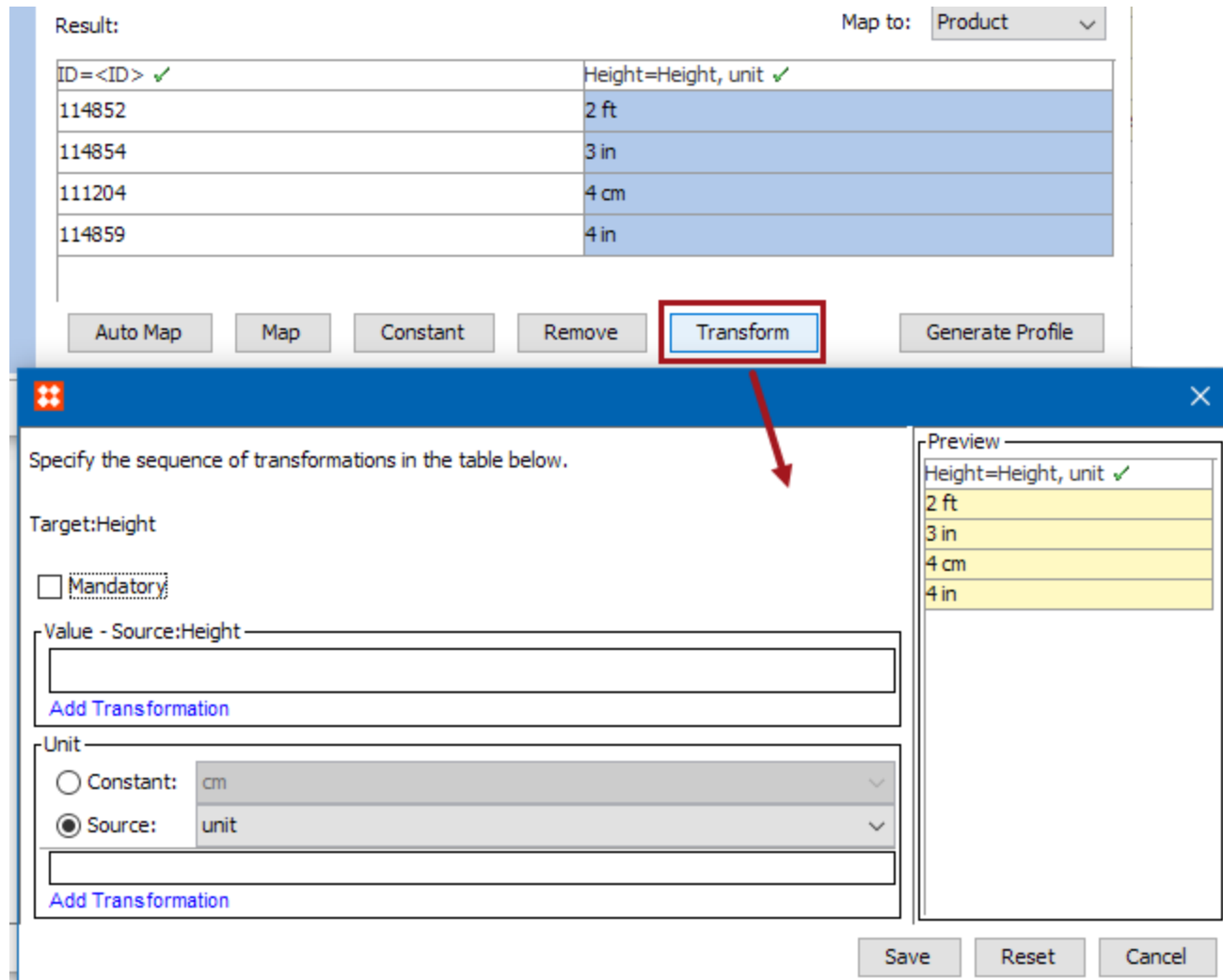
Map to: Product

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 automapping of the 'units' column to the attribute values, the units should be placed in the consecutive column of the column that holds the attribute values in the imported file.

- 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

It is possible to use transformations to improve the quality of 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.

In this example, you have an import file with 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 Variable and enter a name.


Map Width (in) 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

Variable Name

☐ Mandatory

Cancel

OK

3. Click OK. A new intermediate level is displayed between the Source and Result panels.

×

Map Data

Source:

<ID>	> Height (in)	> Width (in)
20859	12	4
20859	40	7

Intermediate Variables:

Variable1=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

Back

Next

Finish

Cancel

- 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**.

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:

<ID>	Height (in)	Width (in)
20859	12	4
20859	40	7

Intermediate Variables:

Variable1=Width (in)
10.16
17.78

Specify the sequence of transformations in the table below.

Target: Variable1=Width (in)

☐ Mandatory

Value - Source: Width (in)

Value	Operator	Value
Value	*	2.54

[Add Transformation](#)

Preview

Variable1=Width (in) ✓
10.16
17.78

Map to: Product

[Transform](#) [Generate Profile](#)

[Next](#) [Finish](#) [Cancel](#)

[Save](#) [Reset](#) [Cancel](#)

5. 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

<ID>
 Height (in)
 Width (in)
 Variable1

Add Transform

Preview

ShippingRestrictions=(Height (in)) ✓

30.48 x10.16

101.6 x17.78

Save Reset Cancel

- The variable now displays the transformed value, which can be mapped in the same manner as source columns are mapped. If you save the import configuration, variables and transformations are saved, as well as the mappings for use each time the same import is run.

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:

<ID>	Height (in)	Width (in)
20859	12	4
20859	40	7

< >

Intermediate Variables:

Variable 1=Width (in)

10.16

17.78

Result:

Map to: Product

ShippingRestrictions=(Height (in)) ✓

30.48 x10.16

101.6 x17.78

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Back

Next

Finish

Cancel

Concatenate into Single-Valued Data Example

It is possible to use transformations to improve the quality 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.

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

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 indicates the language of the values. For this case the following spreadsheet is used as input:

1. In the Import Manager 'Map Data' step, map the object's **ID**.

Import Manager
✕

Steps

- Select Configuration
- Select Data Source
- Select Format
- Map Data**
- Identify Objects
- Identify Destination
- Select Business Rules
- Advanced Settings

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

Back
Next
Finish
Cancel

- Map the second 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

3. 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. Hit **Save**.

Note the Dimension Point section, which is shown because the attribute has a single dimension dependency. Here the dimension point can either be mapped to a Constant or a Source. The Constant is used in case all values of the import are in the same language. The default behavior is to use the language of the import context. The 'Source' radio button is used in case 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. Again, in this example the Constant mapping meets the requirement.

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 third 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**. Hit **Save** when finished.

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.

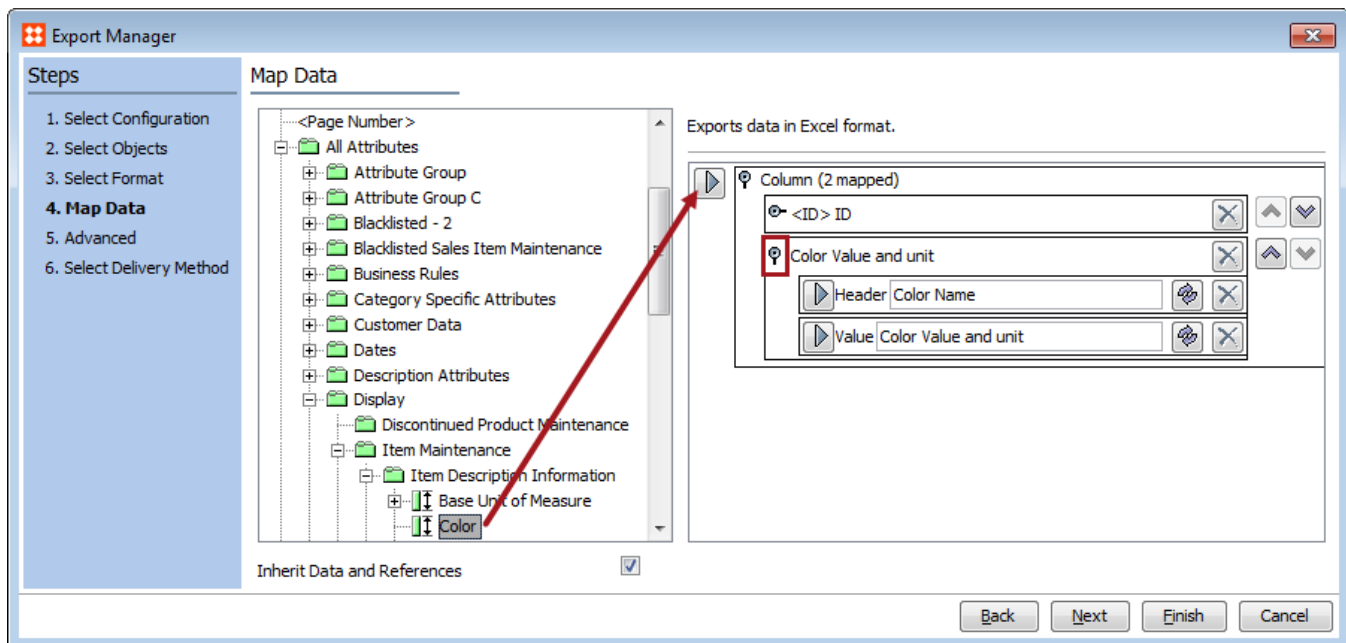
Header and Value Aspects Example

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.

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

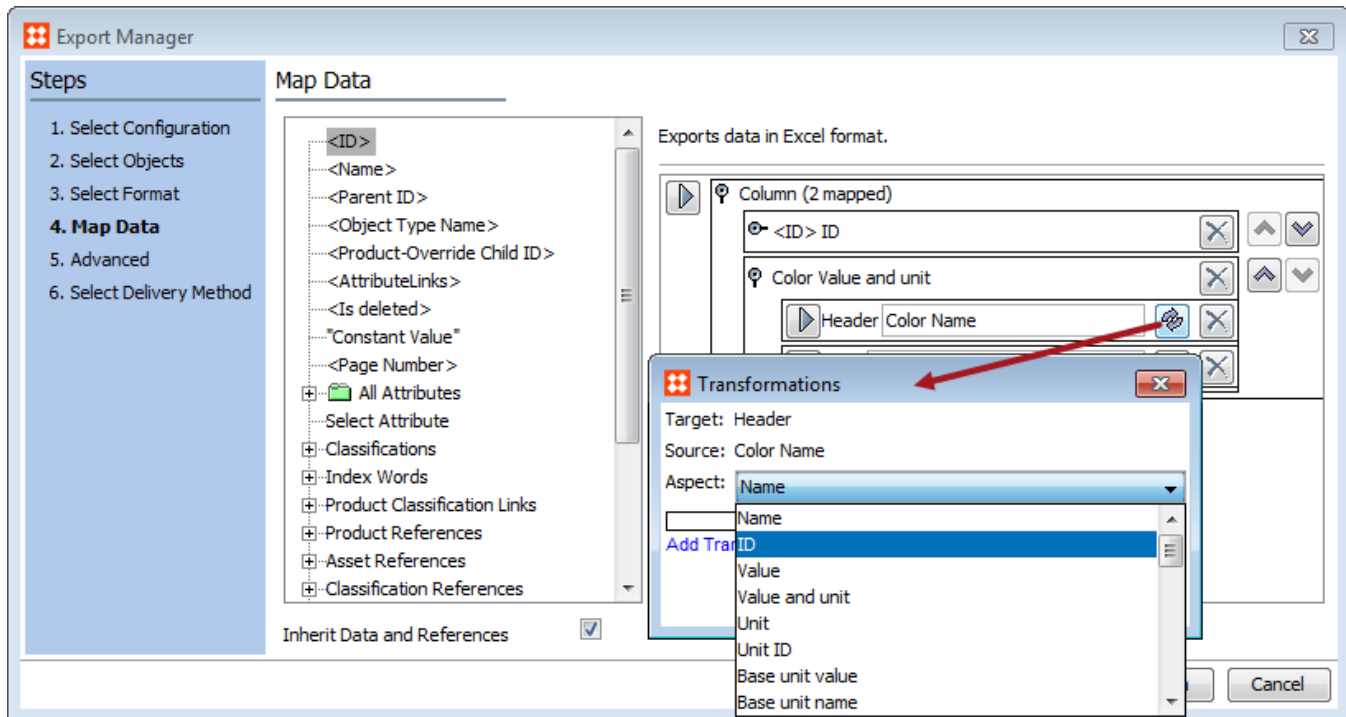
The ID and the Color attribute are mapped using the default settings. When the flipper for the mapped column is opened, separate header and value parts are displayed. When mapping an attribute, by default, the attribute name is mapped to the header part and the value of the attribute is mapped to the value part.



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

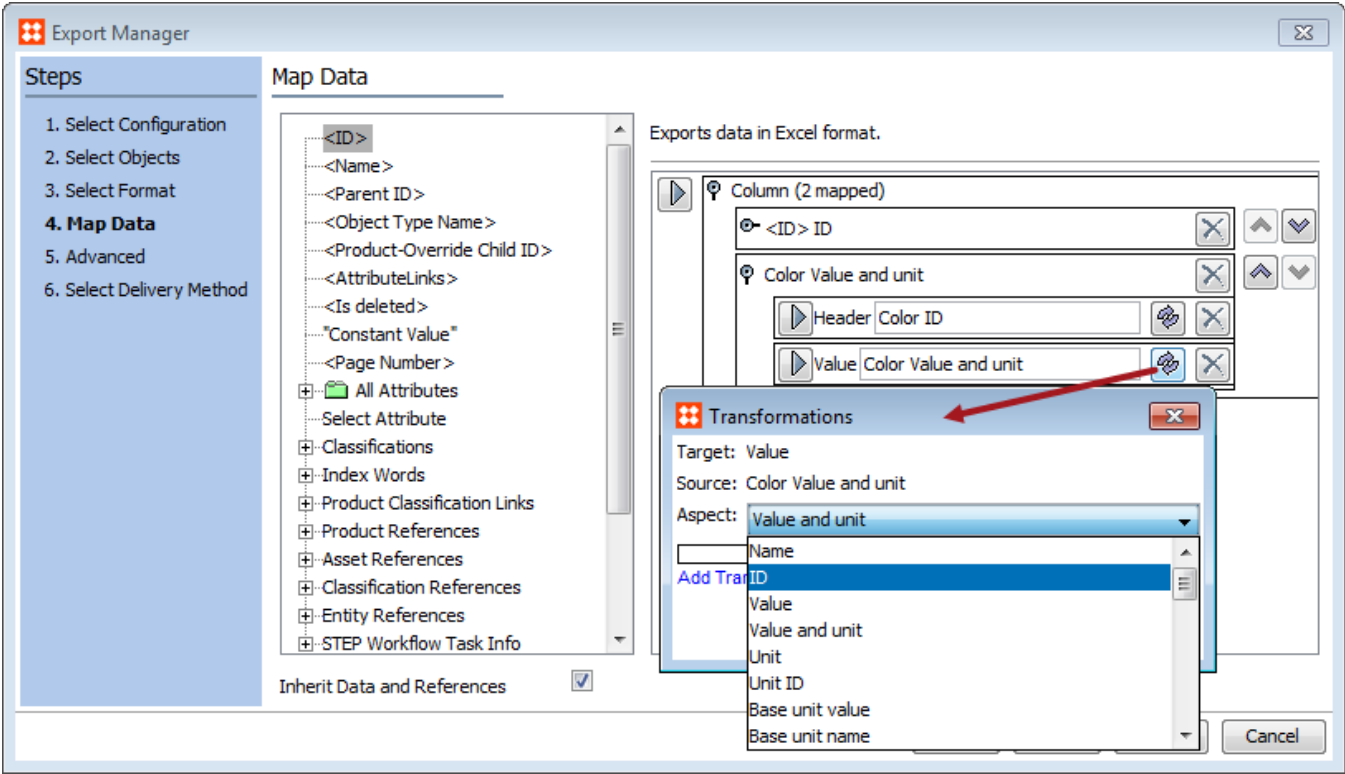
The data exported as header and value is determined by the aspect setting. You can include the attribute ID in the header using the transform button and selecting the ID aspect for the header column.



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

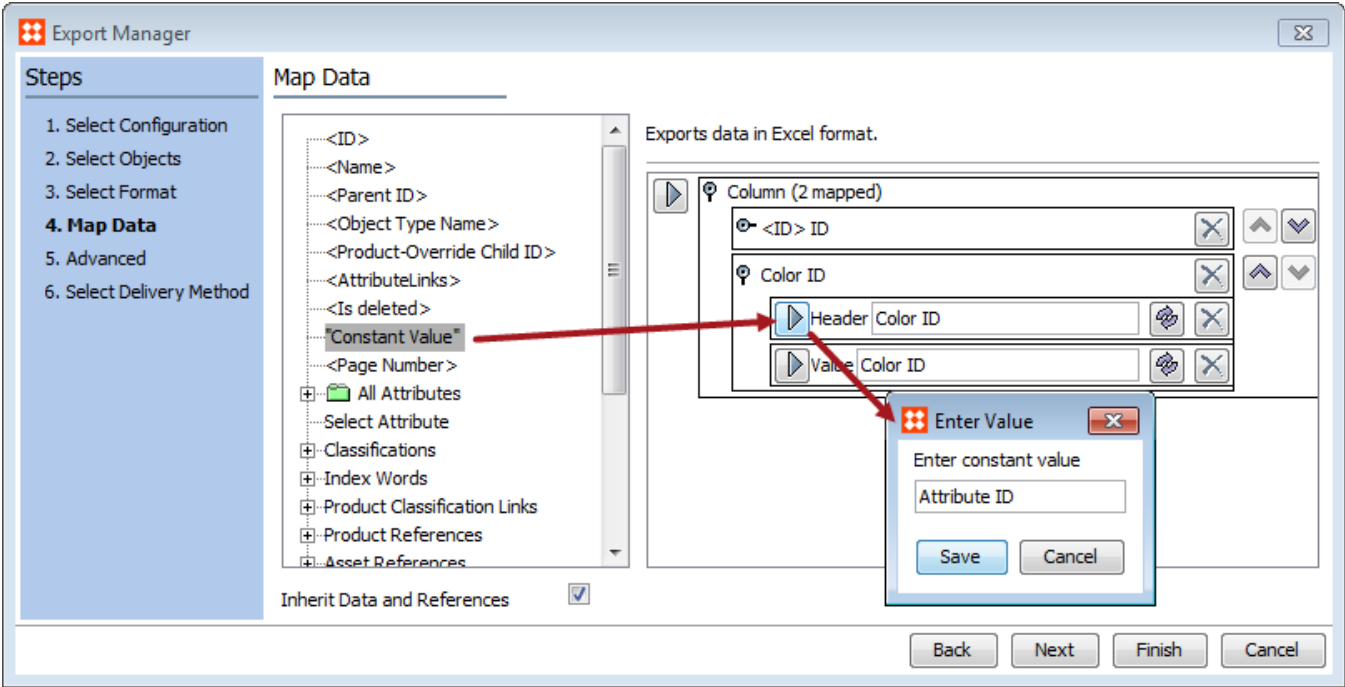
You could also generate a file output with an additional data column like product ID, attribute ID, and the attribute value by adding the ID aspect for the value part of the mapping.



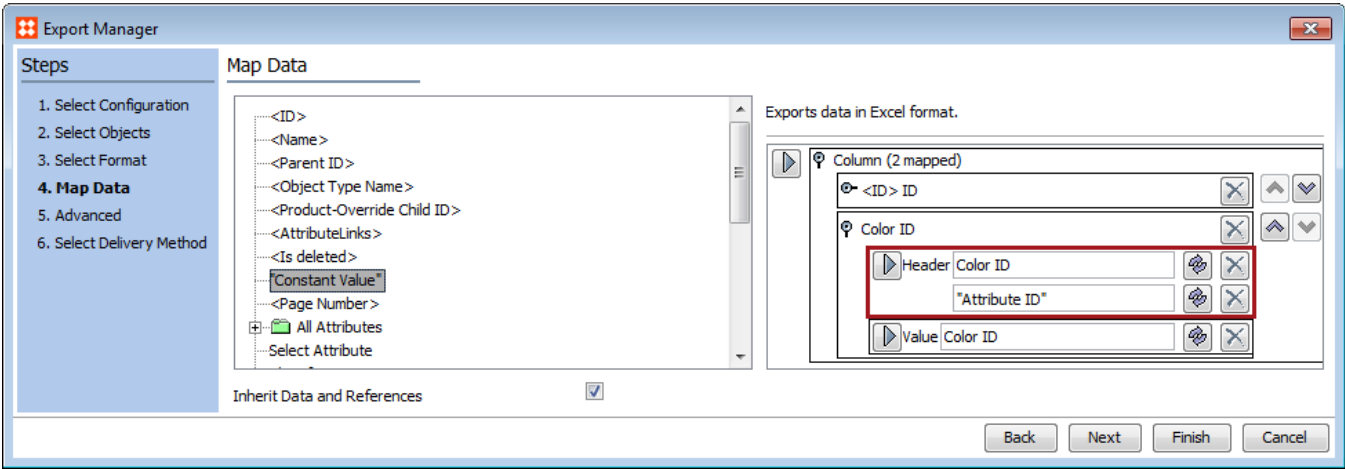
Exporting with this mapping results in the following file, which includes the attribute ID in both the header and the value for each product:

	A	B
1	<ID>	ColorID
2	179624	ColorID
3	179625	ColorID
4	179626	ColorID
5	179807	ColorID

Now you can map a Constant Value to the header part by selecting "Constant Value" on the left and map it to the header as shown below.



After saving the constant value, there are two elements for the header.



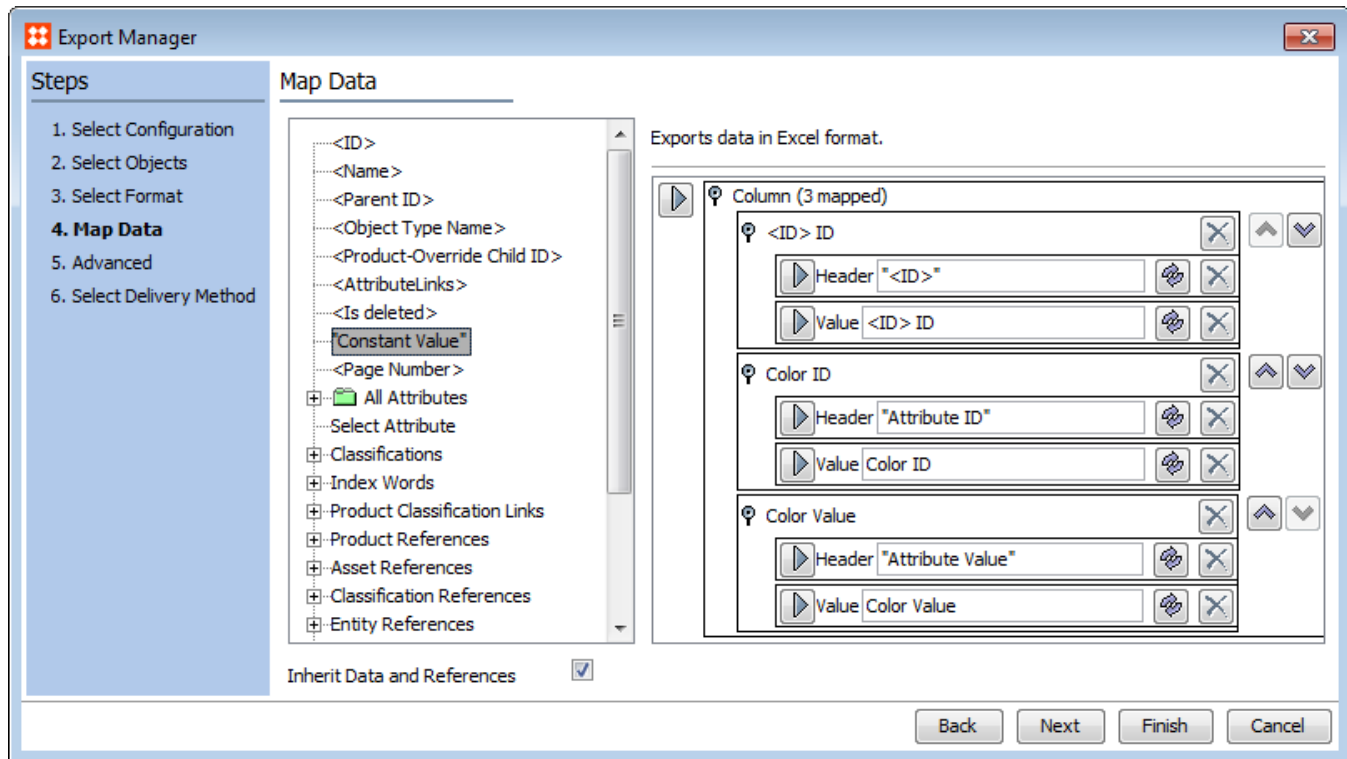
Exporting with this mapping results in a file where the header is a concatenation of the attribute ID (ColorID) and the constant value 'Attribute ID'.

	A	B
1	<ID>	ColorIDAttribute ID
2	179624	ColorID
3	179625	ColorID
4	179626	ColorID
5	179807	ColorID

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.



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

It is possible to use transformations to improve the quality 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.

The screenshot shows the 'Map Data' dialog in STEP Trailblazer 8.1. The 'Steps' panel on the left indicates the current step is '4. Map Data'. The 'Source' table lists two rows: ID 114852 with 'web' in the 'Output' column and ID 114854 with 'Print' in the 'Output' column. The 'CMYK' column contains 'C:0M:31Y:37K:0' for 'web' and 'C:0M:52Y:100K:64' for 'Print'. The 'RGB Hex' column contains '#FFB0A1' for 'web' and '#SC2C00' for 'Print'. A modal window titled 'Specify the sequence of transformations in the table below.' is open, showing a transformation for 'Target: PrimaryColor'. The transformation is 'If value equals' with 'web' in the 'Value' field. The 'Source' is 'RGB Hex' and the 'Else source' is 'CMYK'. A red arrow points from the 'Transform' button in the modal to the 'Transform' button in the main dialog. The main dialog also shows a 'Preview' section with 'PrimaryColor=Output' and the corresponding color values. The 'Map to' dropdown is set to 'Product'.

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

Auto Map Map Constant Remove Transform Generate Profile

Back Next Finish Cancel

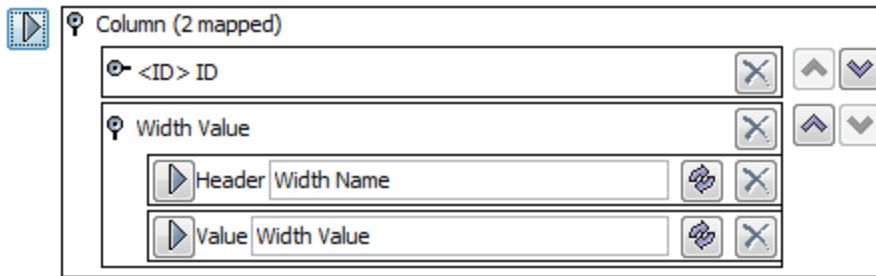
Merge Attributes and Constant Example

You can modify an output by concatenating separate pieces of data from STEP to display in the same field upon export.

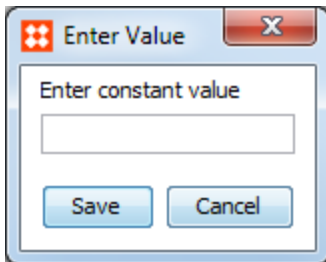
For example, you can merge three attributes with the names width, depth, and height into one export field in the format 'width x depth x height', and also show only the value, not the unit. This mapping will include concatenating values with constant values.

1. In the left panel, click <ID> and then in the right panel, click the right arrow icon.
2. In the left panel, click **Select Attribute**, and then in the right panel, click the right arrow icon. In the **Select Attribute or Attribute Group** dialog, search for the width attribute, and then click **Select**.
3. Click the **Transformation** button next to the width attribute you just selected.

4. In the **Transformations** dialog, from the **Aspect** list, select **Value**.
5. In the left panel select the **Constant Value**, and then click the right arrow icon next to the attribute you just selected. This arrow is used when you want to merge attributes.



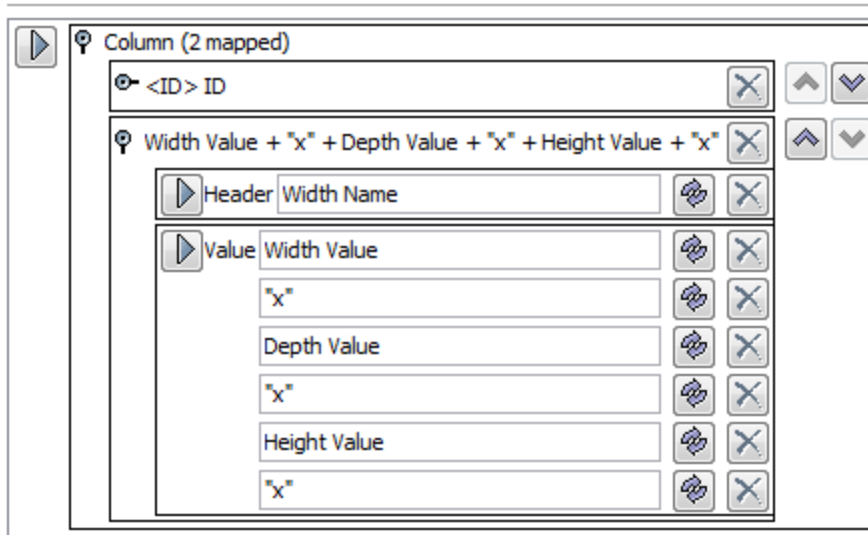
6. In the **Enter Value** dialog type **X**.



7. Repeat steps 2-6 and add the attributes depth and height.

The right panel now looks like the following image.

Exports data in Excel format.



Note: In the above example, you did not use the ValueAndUnit for the dimensions. However, if those dimension attributes did have valid units assigned to them, you could have extracted the units in the same manner.

When exporting in Excel / CSV format, you can 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 the data you want to import. For general information about transformations, see the **Inbound Map Data - Transform** section.

Multiple values for a single multivalued 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).

If STEP does not automatically recognize the import data as a 'multi value' after the data has been mapped to either a multivalued attribute or a reference / link type that allows multiple targets, the result column can be transformed and the delimiter / separator specified as shown below.

In Example 1, you are going to merge the values of the two columns "Color 1" and a "Color 2" into a single multi-valued attribute named "Colors".

1. Map 'Color 1' to 'PrimaryColor'.

Map Color 1 to

☐ ID

☐ Name

☒ Attribute

☐ Product Classification Link

☐ Product Reference

☐ Asset Reference

☐ Classification Reference

☐ Entity Reference

☐ Reference Meta-Data

☐ Parent

☐ Object Type

☐ Variable

☐ Multivalued Variable

Browse Search

(PrimaryColor) Search

Advanced

Name
> (PrimaryColor) ID = PrimaryColor

Color 1 >

green

Orange

Pink

Green

Map

Back Next

2. Transform the mapped data by clicking on the 'PrimaryColor=Color 1' column and hitting the **Transform** button. Add the **Append text** and **Append from source** transformations.

Specify the sequence of transformations in the table below.

Target: PrimaryColor

☐ Mandatory

Value - Source: Color 1

Append ;

Append from Color 2

<ID>
<Name>
<ShortItemDescription[DK]>
Brand Name
<Unique Key>
Color 1
Color 2

Constant: ☒ Source: ☐

Add Transformation

Preview

PrimaryColor=Color 1 ✓

green;black
Orange;Blue
Pink;Purple
Green;blue
Orange;Red
Red;Pink

Save Reset Cancel

3. Click **OK**, and continue navigating through the mappings and rest of the wizard.

Specify the sequence of transformations in the table below.

Target: PrimaryColor

☐ Mandatory

Value - Source: Color 1

Append

Append from Color 2

Append from multivalue: <ID> Multivalue Separator ; Separator -

Add Transformation

Dimension Point

Constant: (From Import Context) Source: <ID>

Add Transformation

Preview

PrimaryColor=Color 1 ✓

green black-114852
Orange Blue-114854
Pink Purple-111204
Green blue-114859
Orange Red-114440
Red Pink-888264

Save Reset Cancel

In Example 2, you have a list of colors in one column, 'Colors' and the corresponding hexadecimals in another, 'Hexadecimals'. You want to merge these two columns into a multi-valued attribute, 'PrimaryColors', so that each entry in the multi-value is a color-hexadecimal pair.

4. Map 'Colors' to 'PrimaryColors'.

Map Data

Source:

<ID>	<Name>	Color 1	Hexadecimals
114852	Blue Light	green; black	#063D10; #000000
114854	Green Flashlight	Orange;Blue	#FA8602; #0202F5
111204	LED Flashlight	Pink;Purple	#DB5ACC; #8C5ADB
114859	Pink light	Green;blue	#063D10; #0202F5
114440	Red light	Orange;Red	#FA8602; #F71B31
888264		Red;Pink	#F71B31; #DB5ACC

Result:

☐ ID
☐ Name
☒ Attribute
☐ Product Classification Link
☐ Product Reference
☐ Asset Reference
☐ Classification Reference

Map Color 1 to

Browse Search

(PrimaryColor) Search

Advanced

Name

> (PrimaryColor) ID = PrimaryColor

Auto Map Map

5. Transform the mapped data using the **Append from multivalue source** transformation as shown below. The value separator, '-' in the example, is optional.

Specify the sequence of transformations in the table below.

Target: PrimaryColor

☐ Mandatory

Value - Source: Color 1

Append from multivalue: Multivalue Separator: Separator:

[Add Transformation](#)

Dimension Point

☒ Constant: (From Import Context) ☐ Source: <ID>

[Add Transformation](#)

Save Reset Cancel

Preview

PrimaryColor=Color 1	✓
green-#063D10; black-#000000	
Orange-#FA8602;Blue-#0202F5	
Pink-#DB5ACC;Purple-#8C5ADB	
Green-#063D10;blue-#0202F5	
Orange-#FA8602;Red-#F71B31	
Red-#F71B31;Pink-#DB5ACC	

Remove or Add Units and Values Example

When working with values that may include units, use the Aspect option to control what data is exported, as defined below.

Removing Units from Values

1. In the right panel, click the flipper for the attribute to extract without unit.
2. Click the **Transformation** icon.

Column (2 mapped)

☒ <ID> ID	X	Up	Down
☒ Height Value and unit	X	Up	Down
Header Height Name	X		
Value Height Value and unit	X	Transformation	

Transformations

Target: Value

Source: Height Value and unit

Aspect:

[Add Transformation](#)

Save Reset Cancel

3. In the **Transformations** dialog, from the **Aspect** list, select **Value**.

- Repeat the previous steps to remove units from other attributes if needed.

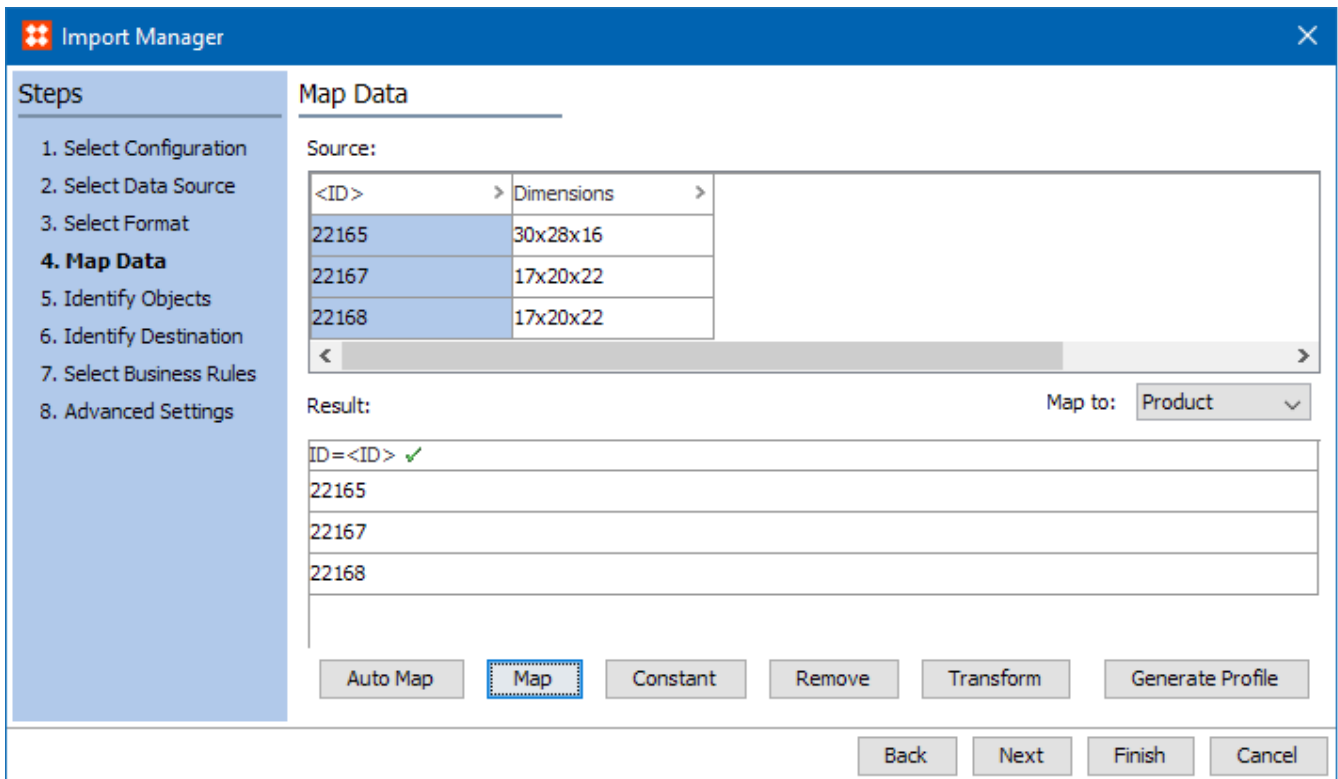
Adding Units to Values

- In the right panel, click the flipper for the attribute to extract with unit.
- Click the **Transformation** icon .
- In the **Transformations** dialog, select the relevant aspect from the **Aspect** list.
- Repeat the previous steps to add units to other attributes if needed.

Split and Extract Data Example

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.

- In the Import Manager **Map Data** step, map the product ID. Then select the column that holds the dimensional data and begin to map it to the desired three separate attributes in STEP.



<ID>	Dimensions
22165	30x28x16
22167	17x20x22
22168	17x20x22

Result	Map to: Product
ID=<ID> ✓	
22165	
22167	
22168	

- 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

Preview

Length=(Dimensions)

30 x28x16

17 x20x22

17 x20x22

Save

Reset

Cancel

- 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 configuration window for transformations. It has a blue header bar with a logo and a close button. The main area is divided into two sections. The left section is titled 'Specify the sequence of transformations in the table below.' and contains a 'Target:Length' field, a 'Mandatory' checkbox, and a 'Value - Source:Dimensions' field. Below the 'Value' field is an 'Add Transformation' button. The right section is titled 'Preview' and shows a list of dimensions: 'Length=(Dimensions)', '30x28x16 in', '17x20x22 in', and '17x20x22 in'. At the bottom of the window are 'Save', 'Reset', and 'Cancel' buttons.

Specify the sequence of transformations in the table below.

Target:Length

☐ Mandatory

Value - Source:Dimensions

Add Transformation

Unit

☒ Constant: in

☐ Source: Dimensions

Add Transformation

Preview

Length=(Dimensions)

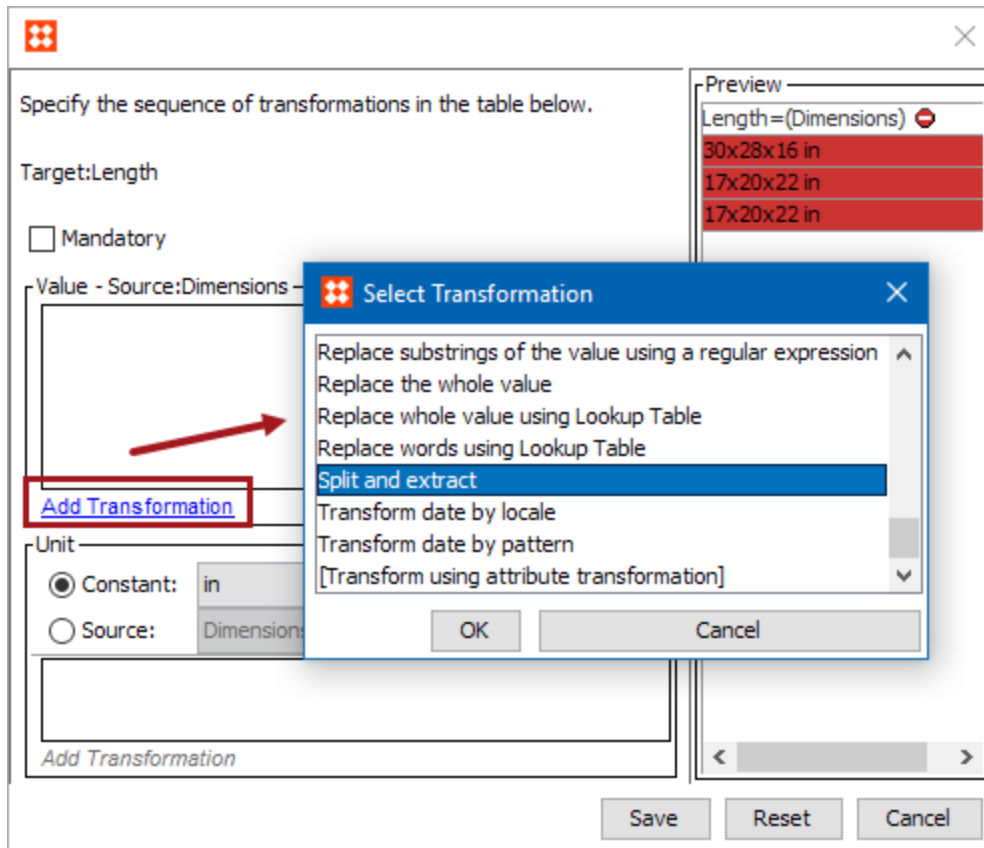
30x28x16 in

17x20x22 in

17x20x22 in

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.



7. 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 and extract field

Add Transformation

Unit

☒ Constant:

☐ Source:

Add Transformation

Preview

Length=(Dimensions) ✓
30 in
17 in
17 in

Save Reset Cancel

Note: If the 'Mandatory' option is checked, it will skip the empty values, and those objects will not be updated / imported into STEP.

- 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 Map Map Constant Remove Transform Generate Profile

9. Repeat the transformation steps for the other two needed attributes, in this case 'Width' and 'Height'.

Import Manager

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

Back Next Finish Cancel

Split, Extract, and Convert Data Example

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.

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.

<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 ▼

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

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.

Selected Column Already Mapped

?

The selected column has already been mapped. Do you want to map it again?

Yes No

- 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

Auto Map

Map

Constant

Remove

Transform

Generate Profile

Using 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 **Attribute Transformations** topic in the **Super User / System Setup** documentation.

- 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

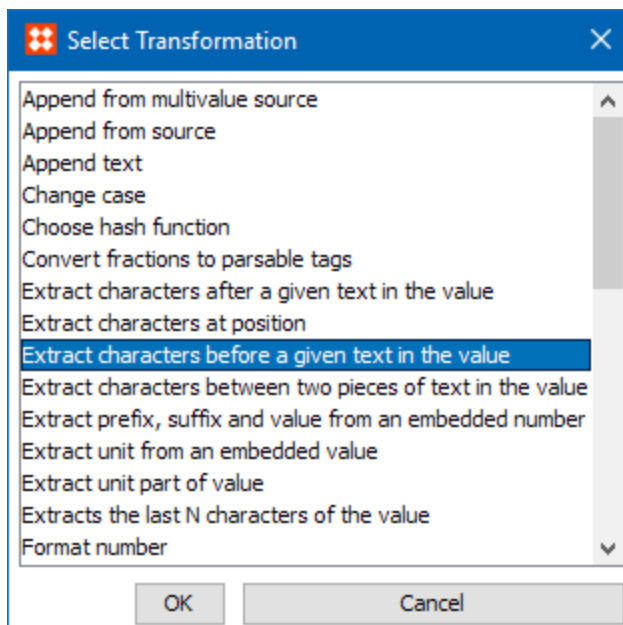
Constant

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'. Here 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)

Preview

PrimaryColor=(Data) ✓

Blue

Green

Brown

Pink

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

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

×

↑

↓

Replace with

×

↑

↓

Replace with

×

↑

↓

Replace with

Add Transformation

Dimension Point

☒ Constant:
☐ Source:

Add Transformation

Save

Reset

Cancel

Preview

Material=(Data) ✓
CN
US
US
GB

- 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

Back

Next

Finish

Cancel

Converting Cryptic Data

Sometimes data in an input file contains data that is different then the way STEP organizes the data, and is thus more cryptic.

In the file below, 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

- 1. To do this, map the first column 'ID' to the ID in STEP, and then map the 'Data' column, in this case, to the 'Primary Color' attribute and the 'Material' attribute. Note that a warning message is displayed saying that the source column has already been mapped. Click 'Yes' to continue Mapping.
- 2. Next, each attribute column in the results field needs to be transformed. In this example, we will focus on extracting the color from the material. Select the 'Primary Color' column in the results field, and click Transform.
- 3. There are a few transformations that need to happen:

- The 'Starting from position' transformation is selected, so that the exact letters can be separated from the material.
- The 'Replace whole value' needs to be set so that the letter coding is the correct color when entering STEP
- The 'Change case' is selected, so that the colors are capitalized with lower case letters that follow.
- Click Save.

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: (From Import Context)

☐ Source: <ID>

[Add Transformation](#)

[Save](#) [Reset](#) [Cancel](#)

Preview

PrimaryColor=(Data) ✓

Blue

Green

Brown

Green

Note: Transformations will work in the sequence order from top to bottom.

5. For the material column, in this example the 'Extract Characters at position' transformation was used- though other transformations could have been chosen. This allowed the last 6 characters to be chosen, as the longest material was 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

Save Reset Cancel

Preview

Material=(Data) ✓

cotton
wool
wool
cotton

6. 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.

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:

<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

Auto ... Map Constant Remove Transform Generate Profile

Back Next Finish Cancel

Transform Using Attribute Transformation Example

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.

System Setup

- Attribute Groups
 - Attribute Transformations
 - Transformation
 - Attr_Trans
 - Commercial Price
 - Fraction
 - New Item
 - Replace Text
- Action Sets
- Contexts
- InDesign Queue
- Lists of Values / LOVs

Attr_Trans - 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

Transformation Lookup Tables can be used to transform words and entire values when importing row-column based data.

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.

Transformation Lookup Table	
Description	
Name	> Value
ID	TransformationLookupTable1
Name	Transformation Lookup Table 1
Object Type	Transformation Lookup Table
Revision	0.1 Last edited by USERL on Tue Feb 07 10:47:10 EST 2017
Approved	Never Been Approved
Translation	Not Translated
Path	Classification 1 root/Suppliers/Products Galore/Batches/Transformation Lookup Table 1
(Asset Object Type)	Transformation Lookup Table
Asset Keywords	abc
Calculated Asset File Name	TransformationLookupTable1-Transformation Lookup Table 1
Name Frequency Max Value	123
Name Frequency Min Value	123

Lookup Table	
☐ Replace with default value when no matches are found (Value Substitution only):	
☐ Ignore Case	
From	To
> Colour	color
> colours	colors
> flavour	flavor
> flavours	flavors
>	
> Add Row	

5 Rows

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

Back

Next

Finish

Cancel

A user then selects the 'Name' field > Transform > Add Transformation > 'Replace words using Lookup Table' > and chooses 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

Save Reset Cancel

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

Auto Map Map Constant Remove Transform Generate Profile

When properly added, the transformation changes the needed 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

Save Reset Cancel

Preview

Name = <Name> ✓
Blue color Light
Green color Flashlight
LED Flashlight
Pink light color
Red light

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

Back
Next
Finish
Cancel

Transformations that use Lookup Tables can be saved in Import Manager Configurations. To save a Lookup Table the referenced Transformation Lookup Table must not be deleted and must have content in the context of the import.

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.

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

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 Publication Excel

Imports a publicationExcel files

Conversion Preview:

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

< >

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: **Publication Excel**
Imports a publicationExcel files

Conversion Preview:

Publicatio... >	Publicatio... >	Publicatio... >	RuleSettings >	VersionName >	Ver >
Acme Pet Su...	Default publi...	107822		English US	Cor

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

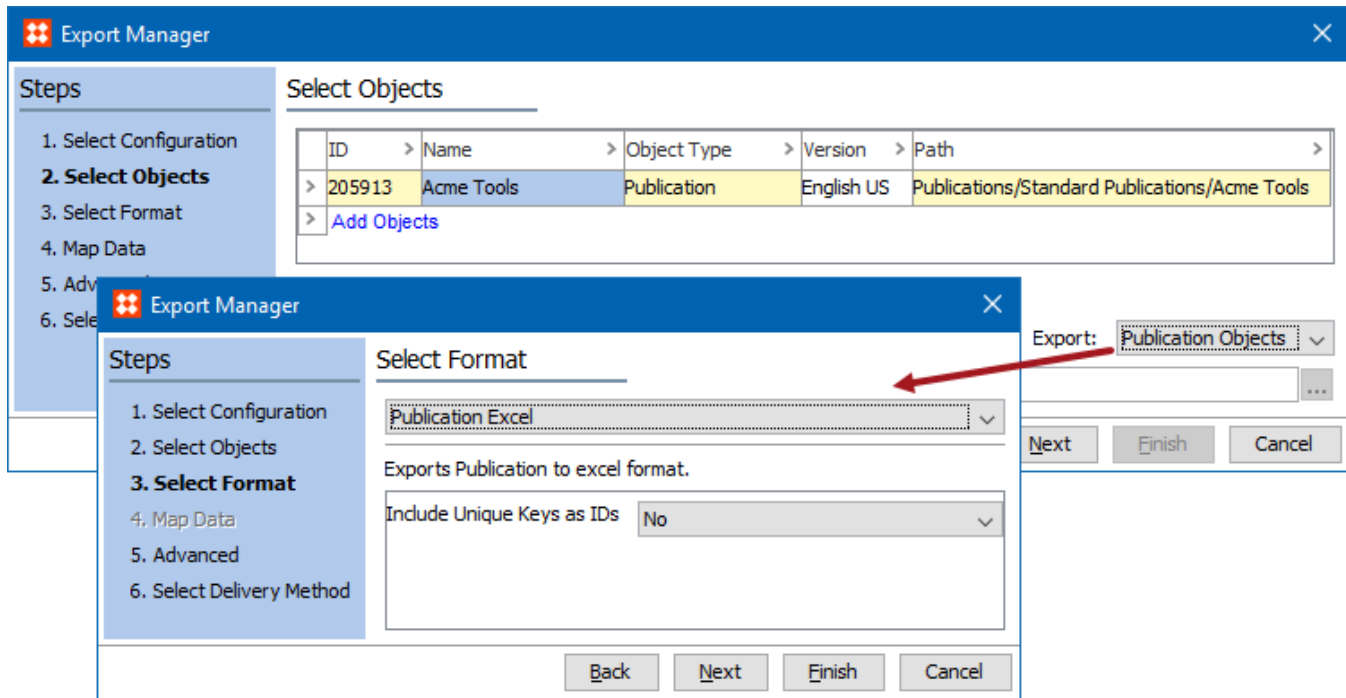
Outbound Data

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.



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.



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.

Schema

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

The XSD file for STEP XML 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

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).

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

- Select Configuration
- Select Data Source
- Select Format**
- Map Data
- Identify Objects
- Identify Destination
- Select Business Rules
- 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

- Identify Endpoint
- Choose Receiver
- Configure Endpoint
- Configure PreProcessor
- Configure Processing Engine**
 - Select Sample File
 - Select Data Source
 - Select Format**
 - Map Data
 - Identify Objects
 - Identify Destination
 - Select Business Rules
 - Advanced Settings
- Configure PostProcessor
- Schedule Endpoint
- 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.

Include Assets

None

Include Asset Content

None

Include Classifications

None

Include Products

Minimum

Include Entities

Minimum

Include Product Attribute Values

All

Include Entity Attribute Values

All

Include Custom Attribute Values

All

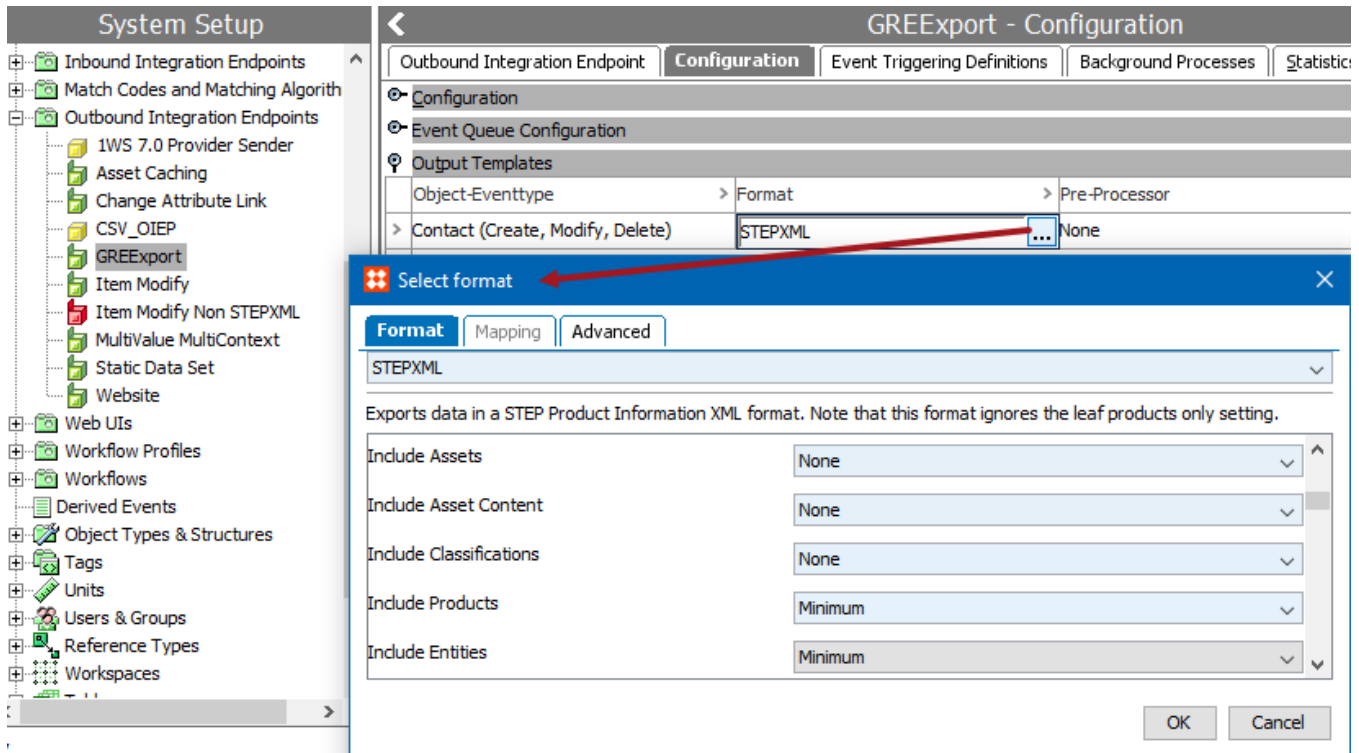
Back

Next

Finish

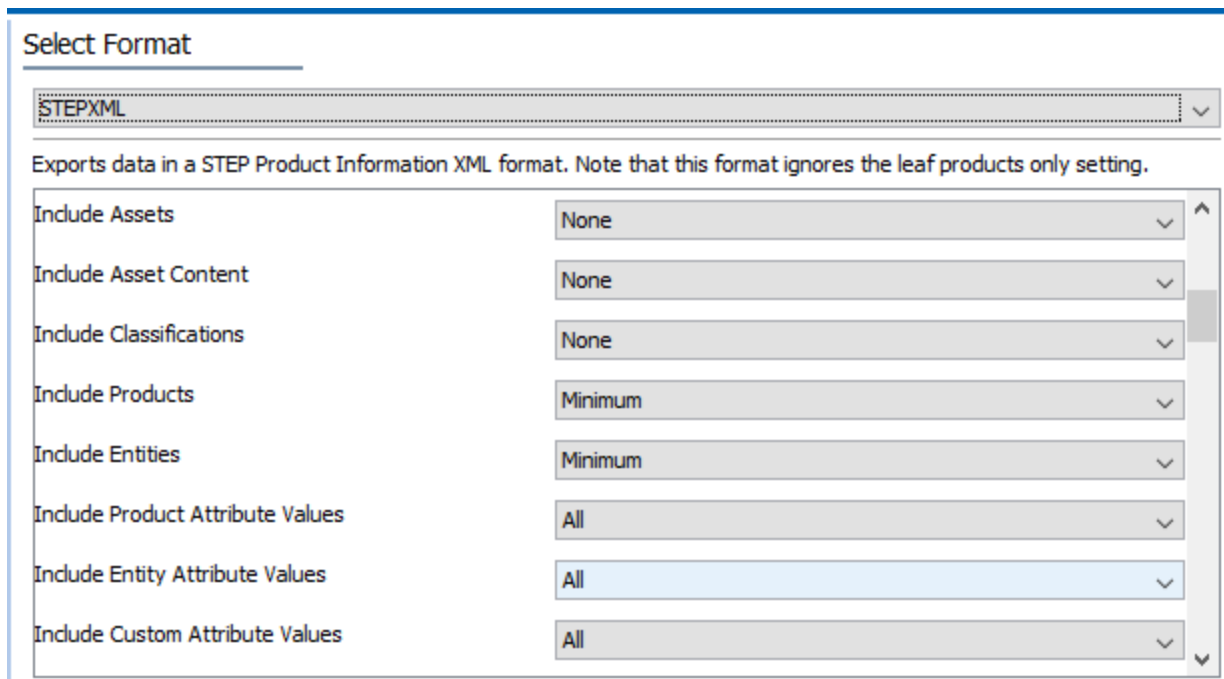
Cancel

OIEP



STEPXML Outbound Parameters

The following parameters are available when exporting using the STEPXML format.



For details about the available options, see **Minimum, Referenced, and Selected in STEPXML** documentation.

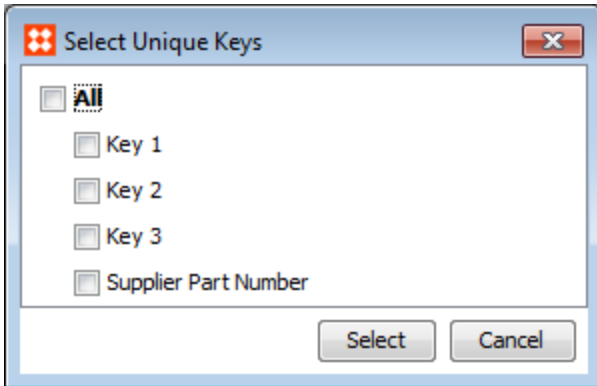
Parameter	Descriptions
Include Validation	Controls the type of validation applied, if any. The following options are available: None and Schema, and DTD. For more information about schema and validation, see the API documentation on the webstart page. For an Advanced STEPXML template, this option is included in the STEP-ProductInformation Tag. For more information, see STEP-ProductInformation Tag in STEPXML.
Include Type Definitions	Controls output of system specific object types, edge types, and system specific reference types. The following options are available: None, Selected, Minimum, and All. When exporting a product hierarchy, selecting Minimum outputs only the types in the selected product hierarchy. Important: Handle reference types and object types with an ID that start with stibo. carefully since they are fundamental objects.
Include Global Settings	Controls output of dimension settings applied to asset contents and calculated templates. The settings are defined on the Users and Groups root node. The following options are available: None and All.
Include Context Definitions	Controls output of dimensions, dimension points, contexts, applied dimension points, and locale settings. The following options are available: None and All.
Include Unit Definitions	Controls output of unit groups and units, including applied meta attributes, values, and base unit conversions. The following options are available: None, Selected, Minimum, and All. When exporting a product hierarchy, selecting Minimum outputs only the units used in the selected product hierarchy.
Include Collection	Controls output of collections groups and collections, and includes search URLs in the output. The exported file will not contain products, classifications, and assets included in a

Parameter	Descriptions
Definitions	collection. The following options are available: None and All.
Include List of Value Definitions	Controls output of LOV definitions and values in LOVs, including applied meta attributes, validation base types, and applied units. The following options are available: None, Minimum, and All. When exporting a product hierarchy, selecting Minimum outputs only the LOVs used in the selected product hierarchy.
Include Attribute Group Definitions	Controls output of attribute groups, including view definitions and settings applied attribute Groups. The following options are available: None, Minimum, and All. When exporting a product hierarchy, selecting Minimum outputs only the attribute groups used in the selected product hierarchy.
Include Attribute Definitions	Controls output of validation base types, dimension dependencies applied units, applied LOV hierarchy filters, calculated templates and so on. The following options are available: None, Minimum, and All. When exporting a product hierarchy, selecting Minimum will only output the attributes used in the selected product hierarchy. Important: System specific attributes (having an ID that starts with stibo. or asset.) are included.
Include Data Container Type Definitions	Controls output of data container types. The following options are available: None, Selected, Minimum, and All. When exporting an entity hierarchy, selecting Minimum outputs only the data container types in the selected entity hierarchy.
Include Assets	Controls output of description attributes and meta data attributes on assets, and determines if the following 'Include Asset Content' parameter is active. The following options are available: None, Minimum, and All. When exporting a product hierarchy or classification hierarchy, selecting Minimum will only output the assets used in the selected product hierarchy.

Parameter	Descriptions
	When exporting asset content using the Include Asset Content parameter below, the Include Assets parameter must be set to Minimum or All. None on the Include Assets parameter indicates that neither asset text, references, nor digital contents will be exported. For an Advanced STEPXML template, this option is included in the Assets Tag. For more information, see Assets Tag in STEPXML.
Include Asset Content	Controls the digital asset contents to be exported along with the asset text and references when the previous 'Include Assets' parameter is set to Minimum or All. The following options are available: Binary and REST. Both options require an selection of at least one Image Conversion. For details about setting up export of asset content, see Export Manager - Assets and Content with STEPXML in the Digital Asset Exchange documentation
Include Classifications	Controls output of classifications and applied meta attributes, including attribute links and references from classifications. The following options are available: None, Minimum, and All. When exporting a product hierarchy or classification hierarchy, selecting Minimum outputs only the classifications used in the selected product hierarchy.
Include Products	Controls output of products and meta attributes applied, including attribute links, attribute values, and includes references from products. The following options are available: None, Selected, Minimum, Referenced, and All. When exporting a product hierarchy, selecting Minimum or Selected outputs only the products in the selected product hierarchy. Additionally, only product references to products in the selected hierarchy are exported. When exporting a product hierarchy, selecting Referenced outputs all products in the hierarchy and products referenced from products in the hierarchy.
Include Entities	Controls output of entity objects and meta attributes applied to entity objects. The following options are available: None, Selected, Minimum, Referenced, and All. When exporting an entity hierarchy, selecting Minimum or Selected outputs only entities in the selected entity hierarchy. Additionally, only entity references to entities in the selected hierarchy are exported.

Parameter	Descriptions
	When exporting an entity hierarchy, selecting Referenced outputs all entities in the hierarchy and entities referenced from entities in the hierarchy.
Include Product Attribute Values	Controls output of product values. The following options are available: None , Minimum , and All .
Include Entity Attribute Values	Controls output of entity values. The following options are available: None , Minimum , and All .
Include Custom Attribute Values	Only relevant for customized STEP solutions. Controls output of custom attribute values. The following options are available: None and All .
Include Data Containers	Controls whether or not data containers, if applicable, are included in the export file when exporting entities.. The following options are available: None and All .
Include Tag Definitions	Controls output of style tags, character tags, special characters, footnotes, and hyperlinks, including information about rendering, short cuts, and applied output formatting. The following options are available: None and All .
Include Context Qualifiers	Controls 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. The following options are available: None and All .
Include Privilege Definitions	Controls output of a list of all actions applied to each action set. The following options are available: None and All .
Include User Definitions	Controls 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. The following options are available: None and All .
Include System Setup	Controls output of the default settings applied to the users and groups root node from the following flippers:

Parameter	Descriptions
	• Product Information Manager Default Settings • Flatplanner Default Settings • DTP Default Settings • GDSN Default Settings • Terms List Settings • WebServices Default Settings • Web UI Default Settings • Default Reference Type of Primary Image • Table Default width and height The following options are available: None and All.
Put product values before child products	Only relevant for customized STEP solutions. Controls output of product values prior to the child products. The following options are available: No and Yes.
Export Inherited Values and References	Controls output of inherited values and references. The following options are available: No and Yes. For more information, see Export Inheritance in STEPXML Example.
Flatten Hierarchy	When set to Yes, values and references that are localized on parent objects are included on each object below the parent (on each leaf object). When set to No, values and references are included as they exist in the hierarchy. Note: The Flatten Hierarchy option does not analyze the validity of the attributes. The following options are available: No and Yes.
Include GDSN Packaging Hierarchy	Controls output of the GDSN Packaging hierarchy. The following options are available: None, All, and All and additionally include the referenced product.
Include Tables	Controls output of STEP tables as they appear in the preview tab in STEP. The following options are available: No and Yes.

Parameter	Descriptions
	Note: Table transformations and commercial data are not included in the exported file. Importing tables is not available.
Include Table Type Definitions	Controls output of row type definitions, column type definitions, and table type definitions, including applied transformations. The following options are available: None, Minimum, and All.
Use Cross-Context Export	Controls output of one or more contexts based on selection. The following options are available: No and Yes. For more information, see Multi-Context Data in STEPXML.
Include Unique Keys as IDs	Controls output of unique key as ID, based on selection of one or more keys. For more on unique keys, see the Unique Keys section of the System Setup / Super User Guide documentation. The following options are available: No and Yes.  For an Advanced STEPXML template, this option is included in the IncludeKey Tag. For more information, see IncludeKey Tag in STEPXML.
Include Overridden Products	Controls output of product overrides. The following options are available: None, Minimum, and All.
Include Key Definitions	Controls output of definitions of objects created as keys. The following options are available: None, Minimum, and All.

Parameter	Descriptions
Include Derived Event Types	Controls output of objects created in System Setup as Derived Event Type. For more information, see Derived Events in the System Setup / Super User Guide. The following options are available: None and All.
Include E-Catalogs	Controls output of eCatalog configurations and price lists. Does not include product selection lists. The following options are available: None and All.
Include Web-Sites	Controls output of websites, with and without workspace. The following options are available: None and All.
Include Event-Queues	Controls output of event queues and their applied message templates, output formatting, and triggering definitions. The following options are available: None and All.
Include Attribute Transformations	Controls output of attribute transformations and their configurations. The following options are available: None and All.
Include Import Configurations	Controls output of assets saved as import configurations. The following options are available: None, Minimum, and All. When exporting a classification hierarchy, selecting Minimum outputs only the import configurations used in the selected product hierarchy.
Include Export Configurations	Controls output of assets saved as export configurations. The following options are available: None, Minimum, and All. When exporting a classification hierarchy, selecting Minimum only export the configurations used in the selected product hierarchy.
Include Bulk Update Configurations	Controls output of assets saved as bulk update configurations. The following options are available: None, Minimum, and All. When exporting a classification hierarchy, selecting Minimum outputs only the bulk update configurations used in the selected product hierarchy.

Parameter	Descriptions
Include Transformation Lookup Table Configurations	Controls output of assets saved as transformation lookup table configurations. The following options are available: None, Minimum, and All. When exporting a classification hierarchy, selecting Minimum outputs only transformation lookup table configurations used in the selected product hierarchy.
Include Component Models	Controls 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. The following options are available: None and All.
Include Publications	Controls output of details of publications. The following options are available: None and All.
Include Publication Groups	Controls output of details of publication groups. The following options are available: None and All.
Include Publication Sections	Controls output of details of publication sections. The following options are available: None and All.
Include Planned Pages	Controls output of details of planned pages. The following options are available: None and All.
Include Actual Page Data	Controls output of details of actual page data. The following options are available: None and All.
Include AutoPage References	Controls output of details of AutoPage references. For more information, see About STEP AutoPage in the AutoPage documentation. The following options are available: None and All.
Include Baskets	Controls output of details of Flatplanner baskets. The following options are available: None and All.

Parameter	Descriptions
Include Pagination Rules	Controls output of details of pagination rules. For more information, see AutoPage Pagination Rules in the AutoPage documentation. The following options are available: None and All.
Include Data Pools	Controls output of data pools. The following options are available: None, Exclude data pool subscriptions, and All.
Include STEP Workflows	Controls output of workflow information, including ID, name, links, valid object types, and configuration. The configurations are exported in a 64 bit encoding and can only be imported without modifications into another STEP system. The following options are available: None, Selected, and All.
Include Global Business Rules	Controls output of global business rules information, including ID, name, links, on approve setting, dependencies, valid object types, and template. The templates are exported in a 64 bit encoding and can only be imported without modifications into another STEP system. The following options are available: None, Selected, All, and Referenced.
Include Match Codes	Controls output of match code information, including ID, name, links, setup group, valid entity hierarchies, and valid object types. For more information, see the Matching and Linking documentation. The following options are available: None and All.
Include Matching Algorithms	Controls output of matching algorithms, including ID, name, links, and configuration. For more information, see the Matching and Linking documentation. The configurations are exported in a 64 bit encoding and can only be imported without modifications into another STEP system. The following options are available: None and All.
Include Web UI Configurations	Controls output of Web UI configurations in the exported file. The following options are available: None and All.

Parameter	Descriptions
Include Integration End Points	Controls output of ID, name, and configurations. The configurations are exported in a 64 bit encoding and can only be imported without modifications into another STEP system. The following options are available: None and All.
Include Event Processors	Controls 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 a 64 bit encoding and can only be imported without modifications into another STEP system. The following options are available: None and All.
Include Setup Groups	Controls 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 / Super User Guide documentation. The following options are available: None and All.
Include Setup Entities	Controls output of setup entities including ID, name, links, and configuration. The configurations are exported in a 64 bit encoding and can only be imported without modifications into another STEP system. The following options are available: None and All.
Include STEP Workflow Task Info	Exports STEP workflow task info if set to Yes. The following options are available: No and Yes.

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:

- 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 Configuration Definitions as Comments
- 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
- Update Objects in STEPXML

Assets Tag in STEPXML

The assets tag can include multiple additional tags, each is described below.

A single <Assets/> tag implies several other default tags. Consider this structure:

```
<Assets>
  <Asset>
    <Name/>
    <Values/>
    ...
  </Asset>
</Assets>
```

If, for example, you only wanted the asset name, you can achieve this using 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. For more information about values output, see the **Inherit Values in STEPXML** section.

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></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <Product> <Name/> <AttributeLink IncludeInherited="true"> <MetaData/> </AttributeLink></pre>

No Inheritance	IncludeInherited
<pre> </Product> </Products> </STEP-ProductInformation> </pre>	<pre> </Product> </Products> </STEP-ProductInformation> </pre>

Results

Only local attribute links for the exported product are output.

No Inheritance
<pre> <STEP-ProductInformation ExportTime="2015-12-23 14:58:34" ExportCon <Products> <Product ID="20878" UserTypeID="Level4" ParentID="20876"> <Name Changed="true">Batteries Level4</Name> <AttributeLink AttributeID="Voltage"/> </Product> </Products> </STEP-ProductInformation> </pre>

Local and inherited attribute links for the exported product are output.

IncludeInherited
<pre> <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> </pre>

AutoApprove in STEPXML

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).

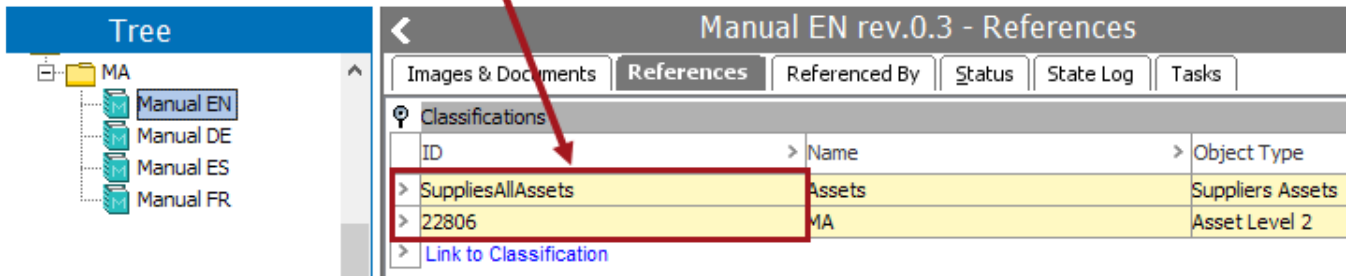
```
<Attribute ID="Depth" Multivalued="false" DefaultUnitID="unece.unit.CMT" ProductMode="Normal" FullTextIndex="false"
  ExternallyMaintained="false" Derived="false" Selected="true" Referenced="true"></Attribute>
```

- **Derived** – attribute element that specifies if the attribute is calculated.

```
<Value AttributeID="CountryOfOrigin" Derived="true">GB</Value>
<Value AttributeID="ManufacturerEAN" Derived="true">9513864492123</Value>
```

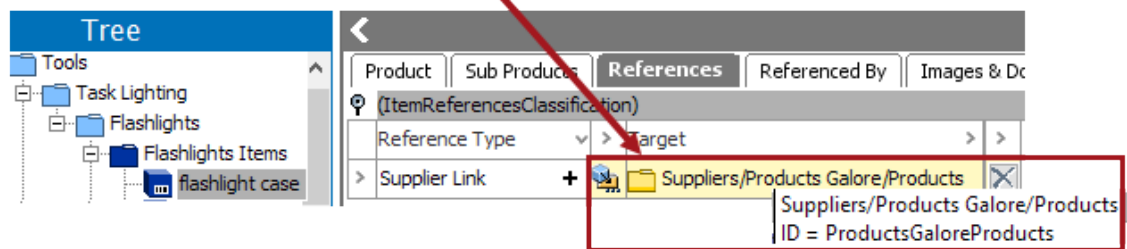
- **ClassificationReference** – used inside both the Product and Asset elements. For assets, this is the parent link (assets can have multiple parents).


```
<ClassificationReference ClassificationID="SuppliesAllAssets" />
<ClassificationReference ClassificationID="22806" />
```



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"
```



Classifications Tag in STEPXML

The Classification tag can include an XML attribute. 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.

```
<STEP-ProductInformation>
```

```
<Classifications>
  <Classification ID="Game Consoles" UserTypeID="Classification Folder"
ParentID="Electronics" Referenced="true">
    <ProductReference ProductID="XBox 360" Type="Classification Owned Product to
Classification Link"/>
  </Classification>
</Classifications>
<Products>
  <Product ID="Xbox 360" UserTypeID="Sellable unit" ParentID="Microsoft Products">
    <ClassificationReference ClassificationID="Game Consoles" Type="Classification Owned
Product to Classification Link"/>
  </Product>
</Products>
</STEP-ProductInformation>
```

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.

```
<STEP-ProductInformation>
  <Classifications>
    <Classification ID="Game Consoles" UserTypeID="Classification Folder"
ParentID="Electronics" Referenced="true">
      <ProductReference ProductID="XBox 360" Type="Classification Owned Product to
Classification Link"/>
    </Classification>
  </Classifications>
  <Products>
```

```
<Product ID="Xbox 360" UserTypeID="Sellable unit" ParentID="Microsoft Products">
  <ClassificationReference ClassificationID="Game Consoles" Type="Classification Owned
Product to Classification Link"/>
</Product>
</Products>
</STEP-ProductInformation>
```

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. 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.

If an ID pattern is specified for the object type being created, the ID can be excluded. For more information, see **Autogenerate using Name Pattern and ID Pattern**.

Also, a default parent and object type can be specified using the products tag attributes. For more information, see **Products Tag in STEPXML**.

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" DefaultUserTypeID="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" UserTypeID="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>

```

Here 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>
```

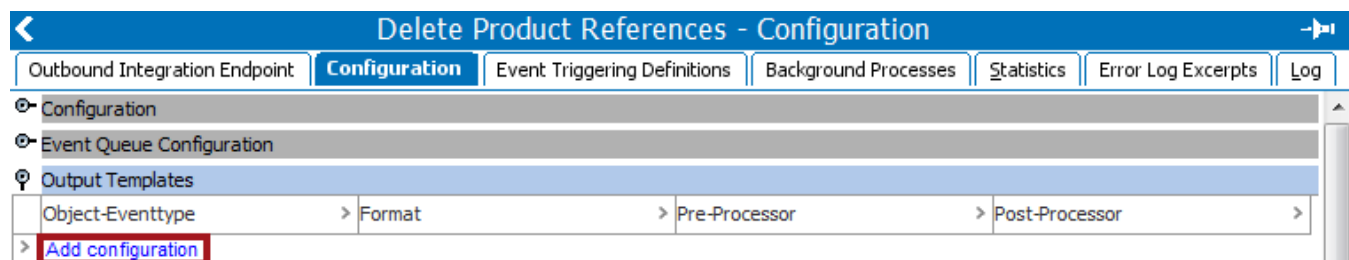
For more information on event-based OIEPs, see the **Outbound Integration Endpoints** documentation.

Delete Product Reference in STEPXML

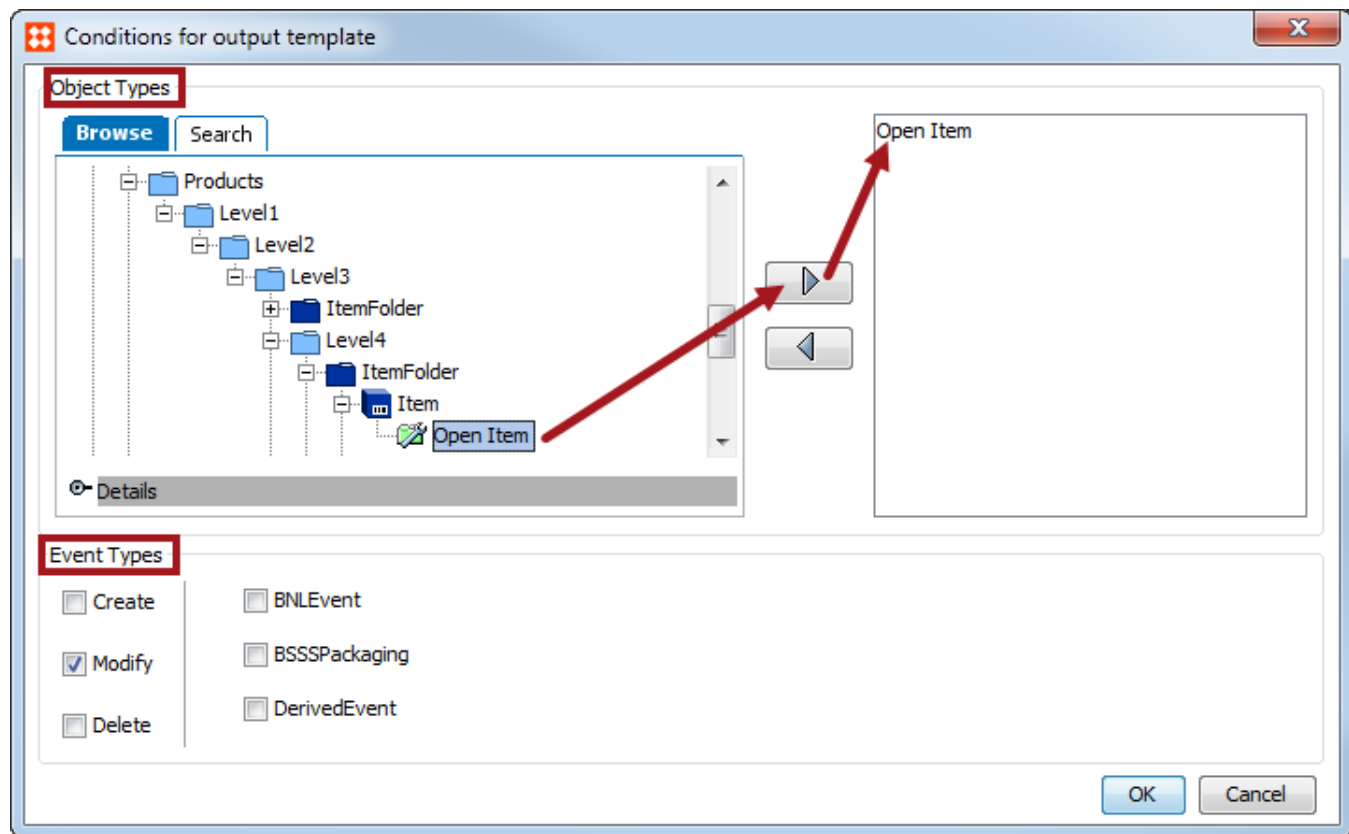
When an Outbound Integration Endpoint (OIEP) is configured to listen for Modify events, use the **STEP-ProductInformation** tag with the **ExportDeletedData** STEPXML attribute to output information about deleted product references. After deleting the reference, approving the product triggers the OIEP.

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.

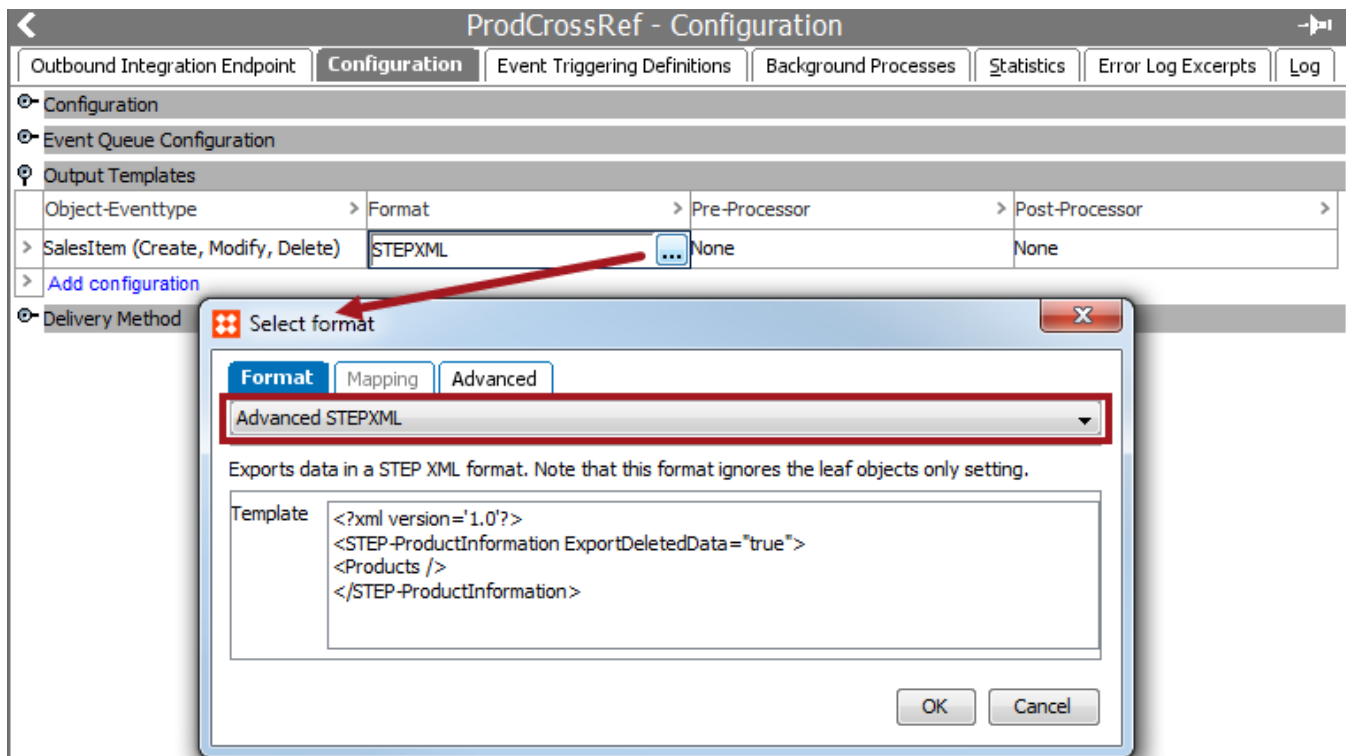


Template

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="Context1">
  <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 weight</Value>
        <Value AttributeID="SellingPrice" UnitID="iso4217.unit.USD">1299</Value>
        <Value AttributeID="FeatureBullet1">Made of solid wood, which is a durable and warm natural material</Value>
        <Value AttributeID="DescriptionTable">Comfy bed, espresso, Queen</Value>
        <Value AttributeID="FeatureBullet2">Adjustable bed sides allow you to use mattresses of different thicknesses</Value>
        <Value AttributeID="DescriptionLong">Made of solid wood, which is a durable and warm natural material</Value>
        <Value AttributeID="DescriptionWeb">Made of solid wood, which is a durable and warm natural material</Value>
        <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 Sales Items</Value>
        <Value AttributeID="Category" Derived="true">Primary Product Hierarchy | Products | Furniture</Value>
      </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 Configuration Definitions as Comments

Workflows, Web UIs, Integration Endpoints (IEPs), and Business Rules 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 import of these files is not supported (e.g. users may not export, edit the comments, and re-import).

Inclusion of configuration definitions as comments is accomplished by setting the DefinitionsAsComments tag to 'true' in an Advanced STEPXML template.

For example:

```
<?xml version='1.0'?>
<STEP-ProductInformation DefinitionsAsComments="true">
<STEPWorkflows ExportSize="All"/>
</STEP-ProductInformation>
```

For more on exporting Workflow definitions as comments, see the **Exporting Workflow Definitions as Comments for External Comparison** section of the **Exporting Workflow Definitions** documentation.

For more on exporting Web UI definitions as comments, see the **Exporting Web UI Definitions as Comments** section of the **Web UI Getting Started** documentation.

For more on exporting integration endpoint definitions as comments, see the **Exporting Inbound Integration Endpoint Definitions as Comments** documentation or the **Exporting Outbound Integration Endpoint Definitions as Comments** documentation.

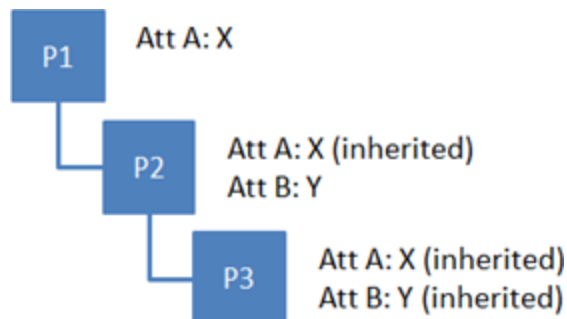
For more on exporting Business Rule definitions as comments, see the **Exporting Business Rule Definitions as Comments** section of the **STEP Business Rules** documentation.

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.

Export Inheritance in STEPXML Example

By default, inherited values and references / links are not exported in STEPXML. This default behavior can be overridden by setting the **Export inherited values and references** parameter to yes. For more information, see **STEPXML Outbound Parameters**.

Only values and references / links inherited from objects above the exported objects in the Tree are localized. For example, exporting the product hierarchy below:



- If only P3 is exported, the product element includes values for both Att A and Att B.
- If P2 and P3 are exported, no values are shown for P3, while P2 has both Att A and Att B (local value).

- If P1, P2 and P3 are exported, only local values are output.

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.

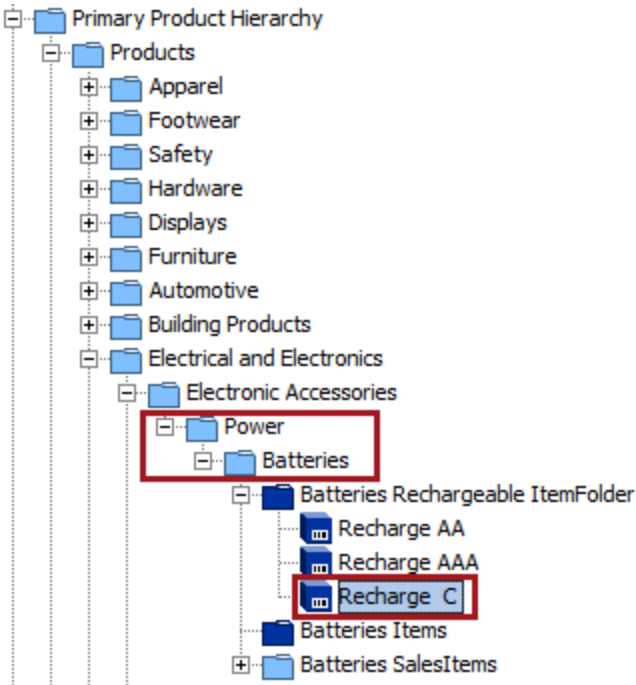
Use multiple instances of the **FilterUserType** tag to include all of the required objects and to filter out the objects that are not needed.

Template

Without Filtering	With Filtering
<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <Product IncludeParent="true"> <Name/> <ProductCrossReference/> </Product> </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> </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>
  </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"/> </Product> </Products> </STEP-ProductInformation></pre>

Modified Object

For each template, the same single item (Recharge C) is modified and approved, which can trigger 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

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 attribute ID of the attribute group.

For example, the following product message template includes the Brand attribute value and the ProductStatus Attributes group.

```
<?xml version="1.0" encoding="UTF-8"?>

<STEP-ProductInformation ExportDeletedData="true" ResolveInlineRefs="true"
ExportDerivedAttrs="true">

  <Products>
    <Product IncludeParent="true">
      <Name/>
      <Values>

        <Value AttributeID="Brand"/> or <Value AttributeID="ProductStatusAttributes"/>
```

```

        </Value>
        <AssetCrossReference Type="Main"/>
        <ProductCrossReference Type="Accessory"/>
        <ClassificationReference Type="Website"/>
    </Product>
</Products>
</STEP-ProductInformation>

```

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 ....>`. For example:

```

<STEP-ProductInformation ExportDerivedAttrs="false" FollowOverrideSubProducts="true">
  <IncludeKey KeyID="SAP_ID"/>
  <Products ExportSize="Minimum">
    <Product>
      <Name/>
      <AttributeLink/>
      <ClassificationReference/>
      <ProductCrossReference/>
      <AssetCrossReference/>
      <Values/>
    </Product>
  </Products>
</STEP-ProductInformation>

```

This instructs the exporter to also export keys of the corresponding IDs:

```

<STEP-ProductInformation
...
  <DeleteProduct ID="77">
    <KeyValue KeyID="SAP_ID">SAP_77</KeyValue>
  </DeleteProduct>

```

```

...
<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 highlight the product object type that references should be inherited to.

For example:

```
<AssetCrossReference IncludeInheritedReferences="Level 3"/>
```

Template

```

<?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 IncludeInheritedReferences="Level 3"/> or <AssetCrossReference
IncludeInherited="true"/>
      <ProductCrossReference Type="Accessory"/>
      <ClassificationReference Type="Website"/>
    </Product>

```

```
</Products>
</STEP-ProductInformation>
```

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.

Template

IncludeInherited	IncludeInheritedAttributes
<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <Product> <Name/> <Values IncludeInherited="true"/> </Product> </Products> </STEP-ProductInformation></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation> <Products> <Product> <Name/> <Values IncludeInheritedAttributes="ItemFolder"/> <Values IncludeInheritedAttributes="Item"/> </Product> </Products> </STEP-ProductInformation></pre>

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" %
  <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" Wor

<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.

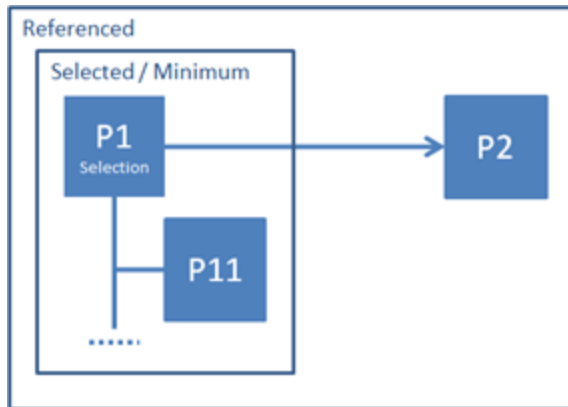
Select Format

STEPXML

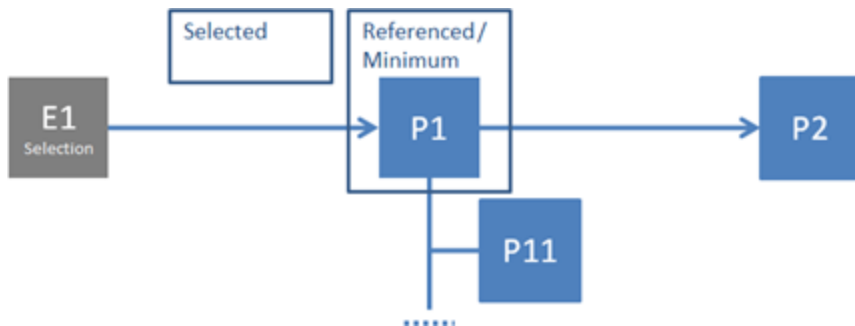
Exports data in a STEP Product Information XML format. Note that this format ignores the leaf products only setting.

Include Assets	None
Include Asset Content	None
Include Classifications	None
Include Products	Minimum
Include Entities	Minimum
Include Product Attribute Values	All
Include Entity Attribute Values	All
Include Custom Attribute Values	All

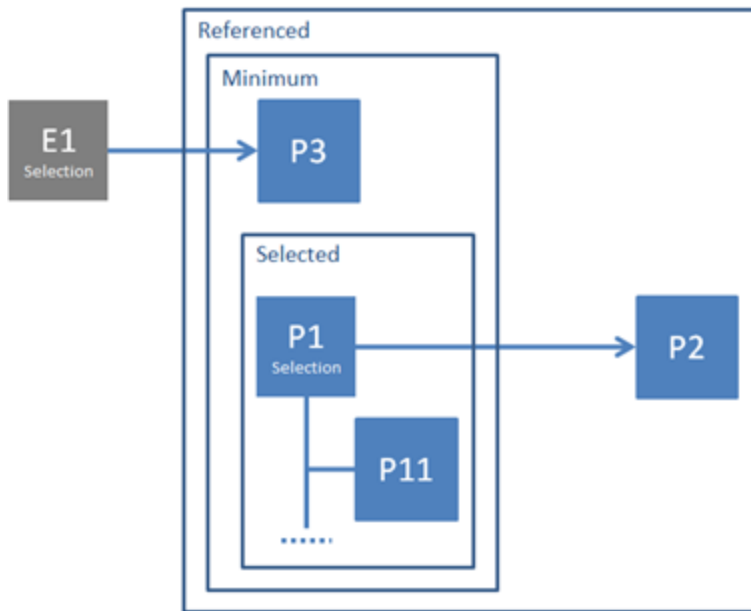
In the example below, product P1 has been selected as root node for the export. The Selected and Minimum options cause P1 and its descendants to be exported. The Referenced setting will also include P2 since it is referenced from P1.



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. Here, Referenced and Minimum will instead have the same meaning, and both cause P1 to be exported, as shown below.

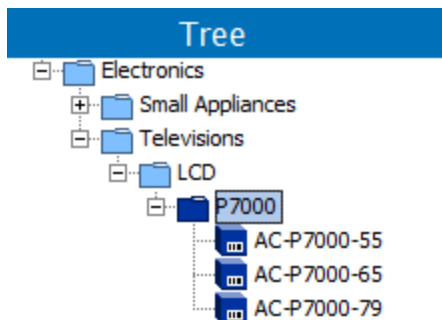


In the following example, the reason for having three options is demonstrated. Here, both a non-product (E1) and a product (P1) are selected as root nodes for the export. With this selection, Referenced, Minimum, and Selected will have three different meanings.



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 **Flatten hierarchy** 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>
    </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:

- **Include Unit Definitions: Minimum** – The Units used by the Attributes being exported.
- **Include List of Value Definitions: Minimum** – The LOVs used by the exported Attributes.
- **Include Attribute Definitions: Minimum** – The set of Attributes used by the objects being exported.

- **Include Products: Selected** – Only the selected product.
- **Include Product Attribute Values: All**

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>
  <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 / exported.

The following example shows data for a single product being exported and includes:

- Contexts are 'US-eng' (Context1) and 'DE-de' (Context3).
- Language-dependent attributes ('LongAdvertisingCopy', 'ConsumerShortDescription' and 'Color').
- Attributes that are not dimension dependent ('Weight' and 'EAN').
- One country-dependent attribute ('Warranty').
- Language dimension points are 'eng' and 'de'.
- Country dimension points are 'US' and 'DE'.

```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main">
  <Qualifiers>
    <Qualifier ID="eng">
      <DimensionPointLink DimensionPointID="eng"/>
      <Context ID="Context1" Priority="1"/>
    </Qualifier>
    <Qualifier ID="DE">
      <DimensionPointLink DimensionPointID="DE"/>
      <Context ID="Context3" Priority="5"/>
    </Qualifier>
    <Qualifier ID="de">
      <DimensionPointLink DimensionPointID="de"/>
      <Context ID="Context3" Priority="1"/>
    </Qualifier>
    <Qualifier ID="US">
      <DimensionPointLink DimensionPointID="US"/>
      <Context ID="Context1" Priority="5"/>
    </Qualifier>
  </Qualifiers>
  <Products>
    <Product ID="I-SalesItem-1121" UserTypeID="SalesItem" ParentID="I-Level3-112">
      <Name>AC-XNPR746</Name>
      <Values>
        <ValueGroup AttributeID="LongAdvertisingCopy">
          <Value QualifierID="eng">The perfect fit for Acme LCD P Series TVs sized 32" to 65"</Value>
          <Value QualifierID="de">Die perfekte Ergänzung für Acme P Series LCD-Fernseher in den
            Größen 32" bis 65"</Value>
        </ValueGroup>
```

```
<ValueGroup AttributeID="Color" ID="BK">
  <Value ID="BK" LOVQualifierID="eng">Black</Value>
  <Value ID="BK" LOVQualifierID="de">Schwarz</Value>
</ValueGroup>
<Value AttributeID="Weight" UnitID="unece.unit.KGM">0.74</Value>
<ValueGroup AttributeID="Warranty">
  <Value UnitID="unece.unit.MON" QualifierID="US">6</Value>
  <Value UnitID="unece.unit.MON" QualifierID="DE">24</Value>
</ValueGroup>
<Value AttributeID="EAN">4905524975872</Value>
<Value AttributeID="ConsumerShortDescription" QualifierID="eng">Wall Bracket for P Series
Acme TVs</Value>
</Values>
</Product>
</Products>
</STEP-ProductInformation>
```

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.

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.

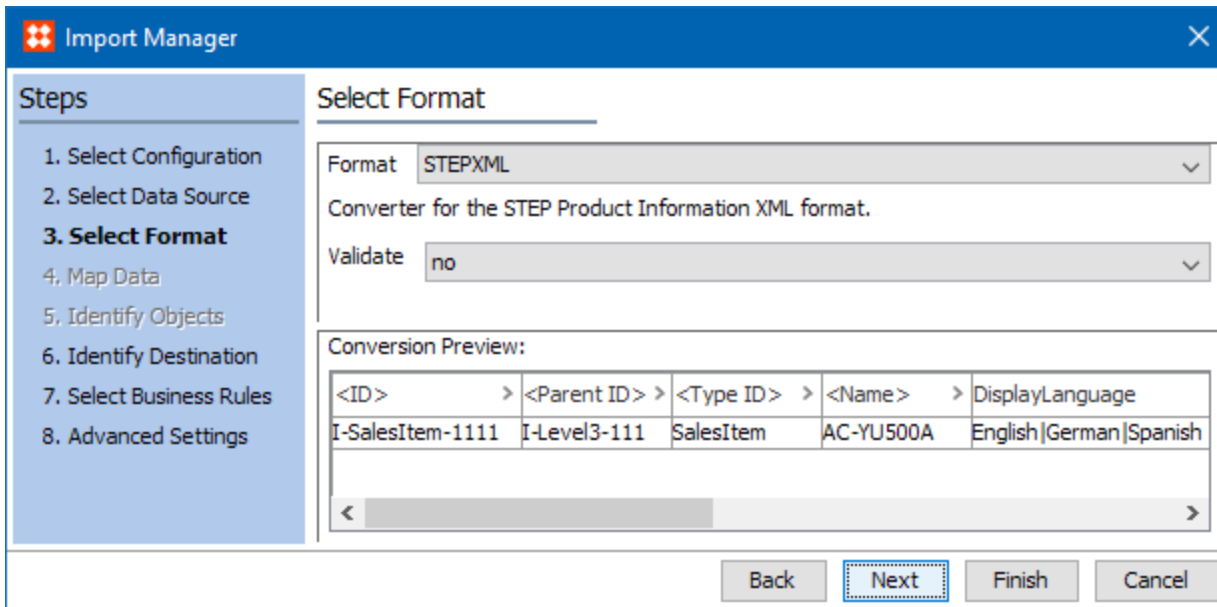
```
<?xml version="1.0" encoding="utf-8"?>
<STEP-ProductInformation ContextID="Context1" WorkspaceID="Main">
  <Products>
    <Product ID="I-SalesItem-1111" UserTypeID="SalesItem" ParentID="I-Level3-111">
      <Name>AC-YU500A</Name>
```



```

<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:

Converter for the STEP Product Information XML format.

Validate:

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.

```

<STEP-ProductInformation>
  <Classifications>

```

```

    <Classification ID="Game Consoles" UserTypeID="Classification Folder"
    ParentID="Electronics" Referenced="true">
      <ProductReference ProductID="XBox 360" Type="Classification Owned Product to
      Classification Link"/>
    </Classification>
  </Classifications>
</Products>
  <Product ID="Xbox 360" UserTypeID="Sellable unit" ParentID="Microsoft Products">
    <ClassificationReference ClassificationID="Game Consoles" Type="Classification Owned
    Product to Classification Link"/>
  </Product>
</Products>
</STEP-ProductInformation>

```

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>

```

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> </ProductCrossReference> </Product> </Products> </STEP-ProductInformation> </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/> </Product> </ProductCrossReference> </Product> </Products> </STEP-ProductInformation> </pre>

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">65165165</Value>
        <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accessor:
        <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 Accessor:
        <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 Accessor
      <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 Accessor
      <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 Accessor
      <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 Accessor
      <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>
  </Entity>
</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>
  </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
- ReplaceAttributeLinks
- 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 DeleteProduct and DeleteProductCrossReference tags are only available for outbound XML when using event-based integration endpoints as discussed in **Delete Objects in STEPXML** and **Delete Product Referenc 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.

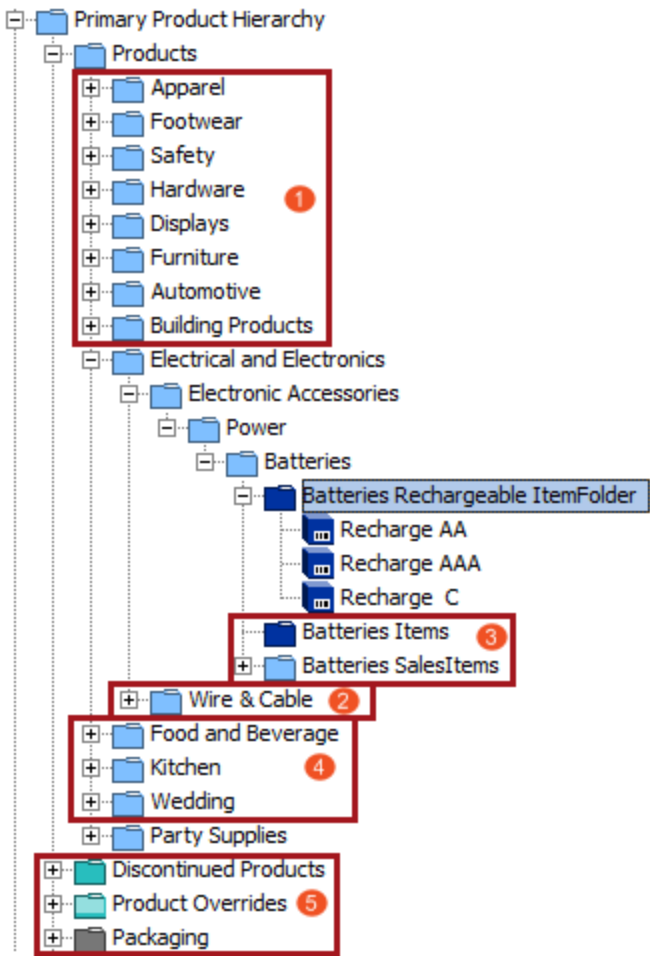
Template

With SequenceProduct	Without SequenceProduct
<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <STEP-ProductInformation></pre>

With SequenceProduct	Without SequenceProduct
<pre><Products> <Product IncludeParent="true"> <SequenceProduct/> <Name/> </Product> </Products> </STEP-ProductInformation></pre>	<pre><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"/>
        <SequenceProduct ID="22150"/>
        <SequenceProduct ID="8298"/>
        <SequenceProduct ID="8299"/>
        <Product ID="8302" UserTypeID="Level1" Selected="false">
          <Name>Electrical and Electronics</Name>
          <SequenceProduct ID="8313"/>
          <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"/>
                <Product ID="123942" UserTypeID="ItemFolder">
                  <Name Changed="true">Batteries Rechargeable ItemFolder</Name>
                </Product>
              </Product>
            </Product>
          </Product>
        </Product>
        <SequenceProduct ID="8303"/>
        <SequenceProduct ID="8304"/>
        <SequenceProduct ID="109946"/>
      </Product>
      <SequenceProduct ID="DiscontinuedProductsRoot"/>
      <SequenceProduct ID="ProductOverridesRoot"/>
      <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>
</Products>
</STEP-ProductInformation>
```

SingleUpdateMode in STEPXML

Importing certain changes requires that the STEP system enters single user 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 User 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">
```

DefinitionsAsComments

- False - default setting, definitions are not exported
- True - configuration definitions for Business Rules, Web UIs, IEPs, or Workflows are exported as comments.

For more details, see the **Export Configuration Definitions as Comments** section of the **Configuration Management** documentation.

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.

ExportDerivedAttrs

- False - default setting
- True - calculated attribute values are derived when an object is output from STEP.

FollowOverrideSubProducts

- False - default setting
- True - all Sub-Products of Product Overrides are included in the output XML.

ResolveInlineRefs

- False - default setting
- True - all inline references are resolved within the attributes values that are exported.

Validation

- None - default. Other options are Schema and DTD.

Update Objects in STEPXML

When updating most objects upon 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 / STEP Super User Guide** documentation.

It is only necessary to include the information to be updated. By default, existing data that is not included in the import file will not be modified.

Limitations

Note that the following update cases cannot be handled by STEPXML:

- Removing or adding individual values for multivalued attributes
- Removing the legal source and target object types for reference / link type definitions
- Removing legal units for attributes

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.

Tree

UNSPSC

Apparel and Luggage and Personal Care Products

Clothing

Athletic wear

Boys athletic wear

Girls athletic wear

Mens athletic wear

Mens T PBG

Mens T PBO

Mens athletic wear rev.0.1 -

Classification

Sub Products

References

Referenced By

Image

Description

Name

>>

Value

>

ID

UNSPSC000.53102902

>

Name

Mens athletic wear

>

Object Type

UNSPSC

>

Revision

0.1 Last edited by USERJ on Thu Aug 04 1

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:

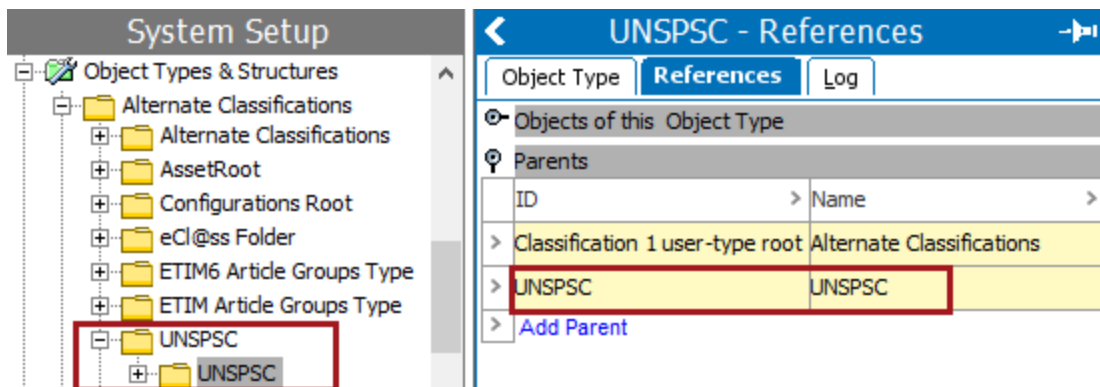
Version	File Types Included in ZIP	File Prep Required	File to Import
13	One (1) Excel file with six (6) tabs	Extract before import	Excel file
17, 18, 19	Six (6) individual .TXT files	None	.ZIP file

Multiple versions of UNSPSC classifications can exist simultaneously in STEP, provided different UNSPSC classification ID prefixes are used.

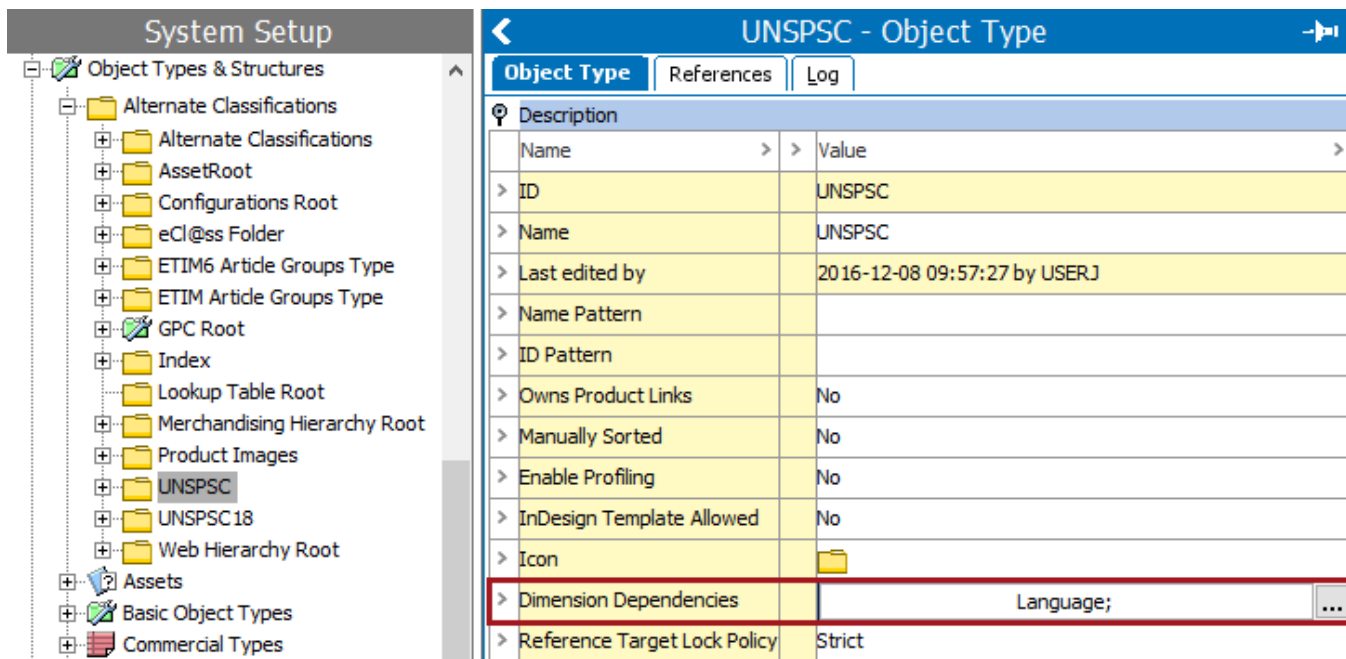
Prerequisite

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.



To import multiple UNSPSC languages for different contexts, on the top-level UNSPSC folder, set the Dimension Dependencies parameter to Language.



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

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: **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**

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 **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

Default Currency Code

Schema name

Schema description

Schema version

Schema standard

en

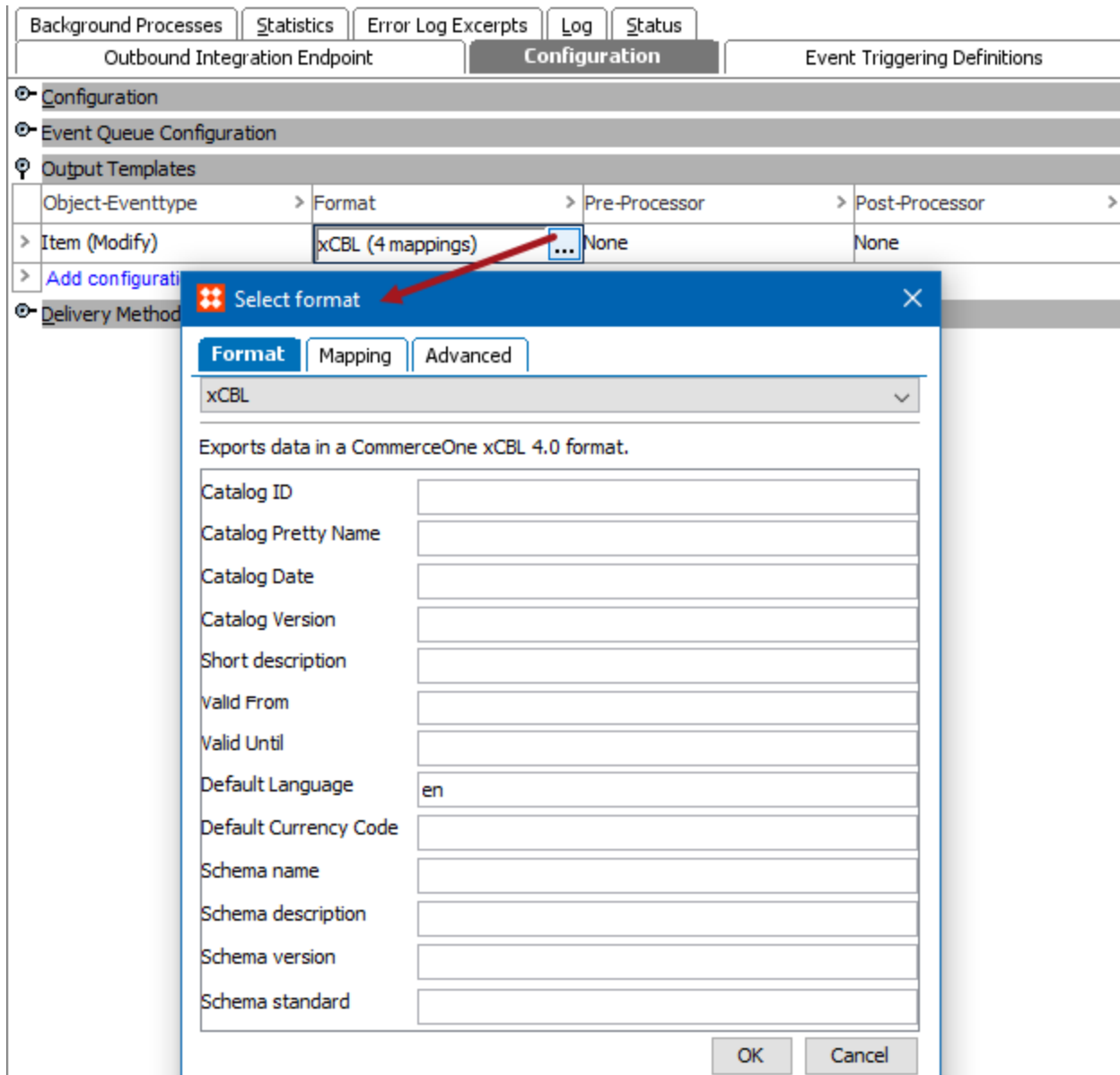
Back

Next

Finish

Cancel

OIEP



Export Manager

The Export Manager allows you to export data on demand or at scheduled intervals and to save or modify an export configuration. A saved export configuration is useful for frequently-used exports where the parameters and options are always, or nearly always, the same.

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. An export configuration is useful for frequently-used exports where the parameters and options are always, or nearly always, the same.

To export data using the Export Manager:

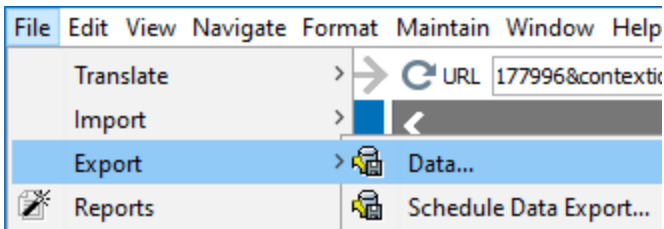
1. Select the appropriate STEP context. All data is exported using the currently selected context.

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.

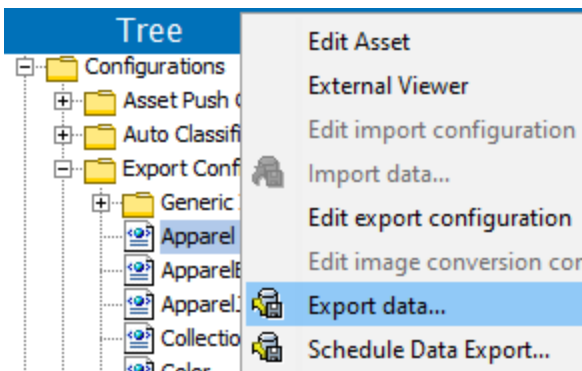
- **From the File Menu:** Click the File menu > Export > **Data**.

The wizard opens in Step 1 and allows you to select an existing configuration or create a new one.



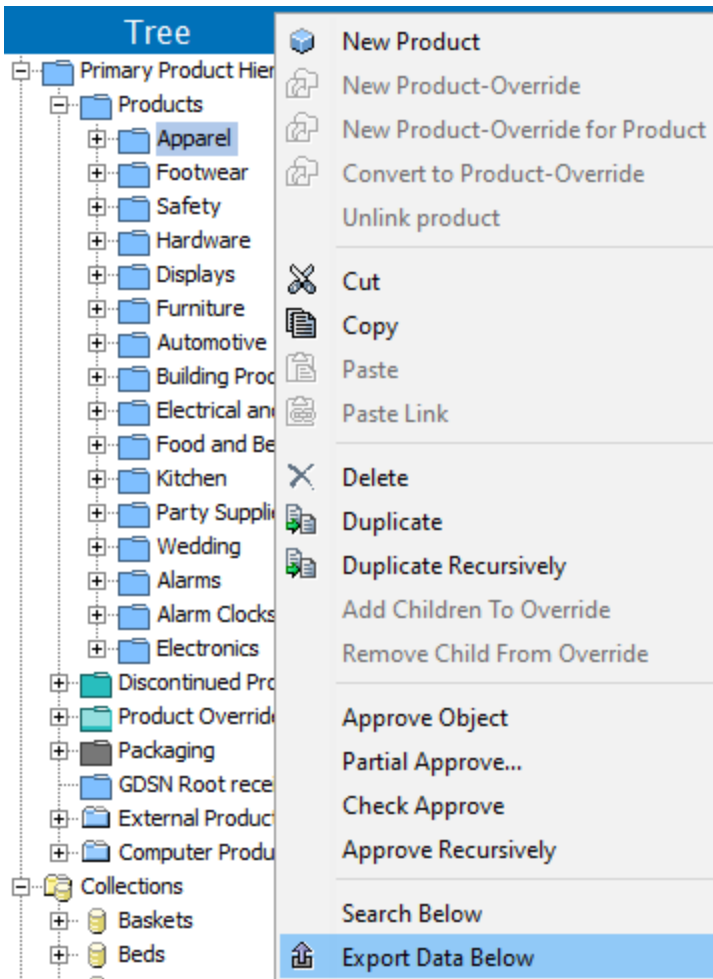
- **From an Export Configuration:** On the Tree, select an export configuration, right-click, and then select **Export Data**.

The wizard opens in Step 2 with the configuration and objects included in the configuration already selected.



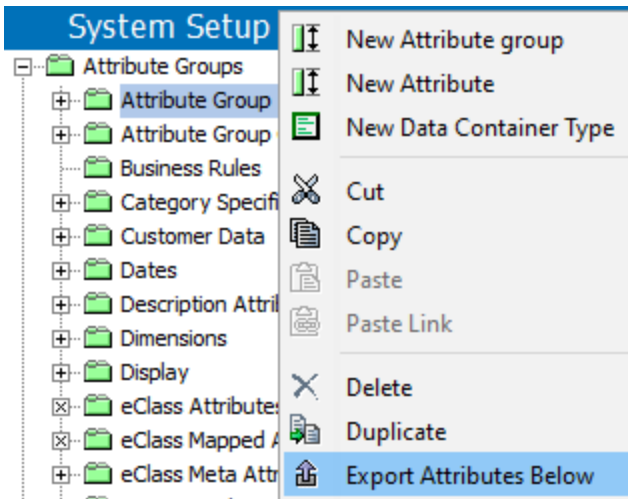
- **From an Object:** On the Tree, select a classification, product group, publication, or entity, right-click, and then select **Export Data Below**.

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.

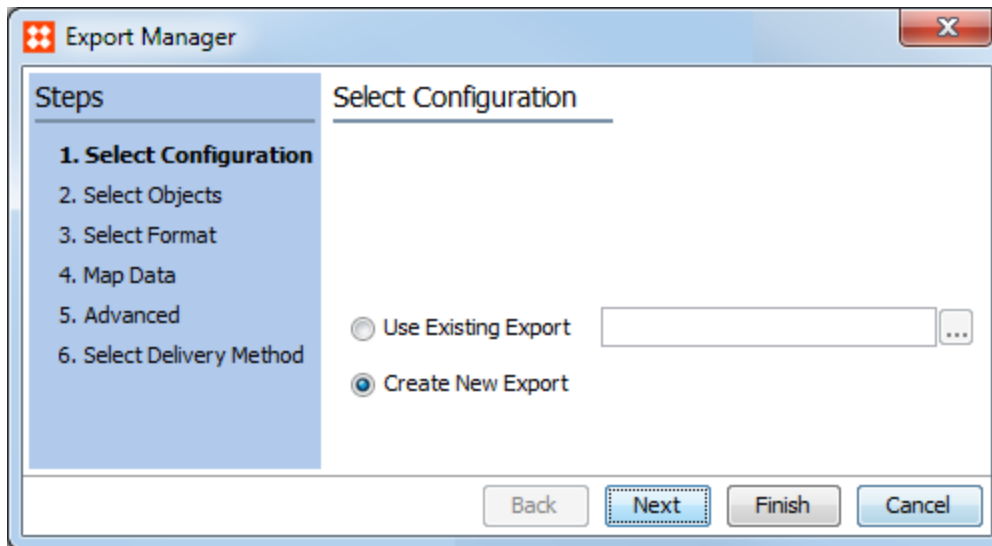


- **From an Attribute Group:** On System Setup, select an attribute group, right-click and then select **Export Attributes Below**.

The wizard opens in Step 2 with the attributes below the selection displayed.



- 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 particular 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.

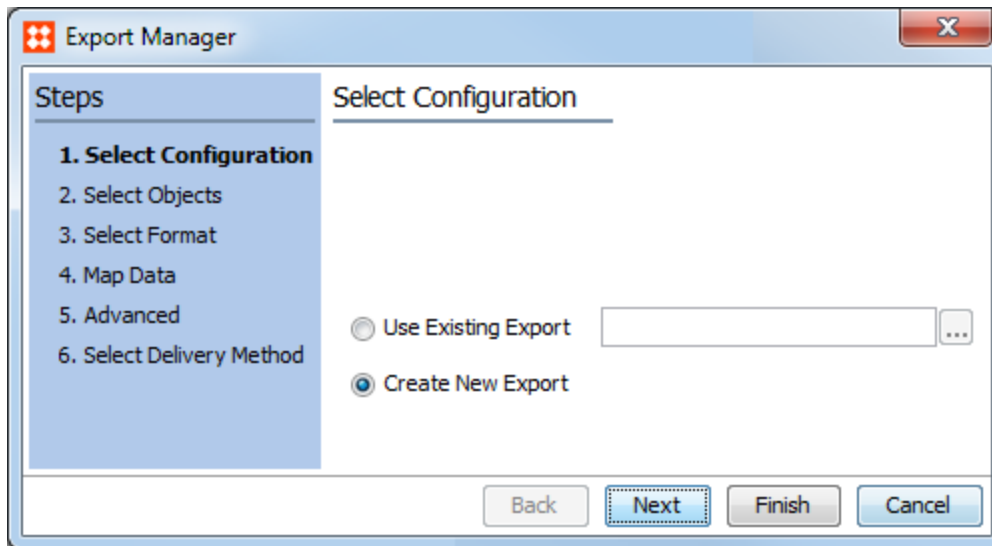
Determine the following information about your required export enables you to accurately set the wizard parameters:

- 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.

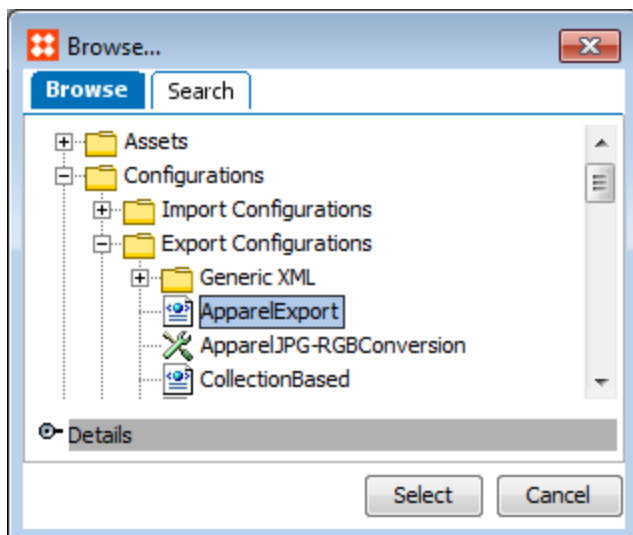
- If you know the attributes you want to extract before you launch the Export Wizard, it is easier to select the appropriate attributes on the Data Map step.

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:
 - To use an existing export configuration, select the Use Existing Export radio button, and either **Browse** or **Search** to locate the configuration in STEP.
 - To create a new export configuration based on an existing export configuration, select the Use Existing Export radio button, select an existing configuration, use the following steps to change it to meet your current requirements, and then save the modified setup in the final step of the wizard.



- 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

- Click the **Add Objects** link to select objects, or leave the Select Object tables empty and export based on the other wizard parameters.

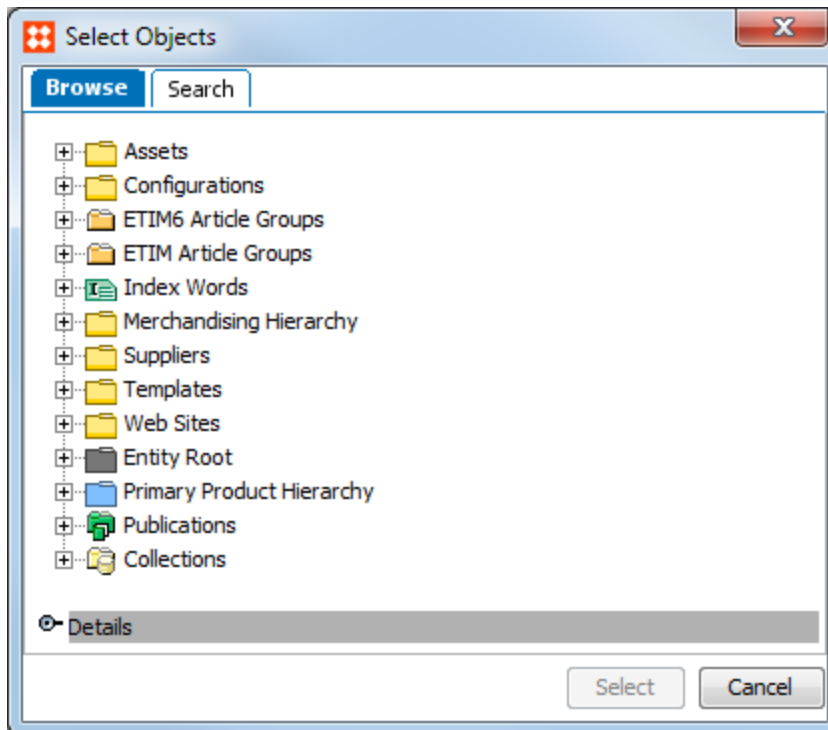
If exporting an Excel Smartsheet, see the **Selecting Objects for Excel Smartsheet Export** section below.

- In the **Export** dropdown, select the type of data to export. The type is used in the next step of the wizard to show only legal export formats.

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 export data type only impacts formats that require mapping. When the Map Data step is disabled, for example with STEPXML, this selection has no effect.

3. In the Select Objects window, use the **Browse** or **Search** tab to locate and select objects for export.
 - 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.

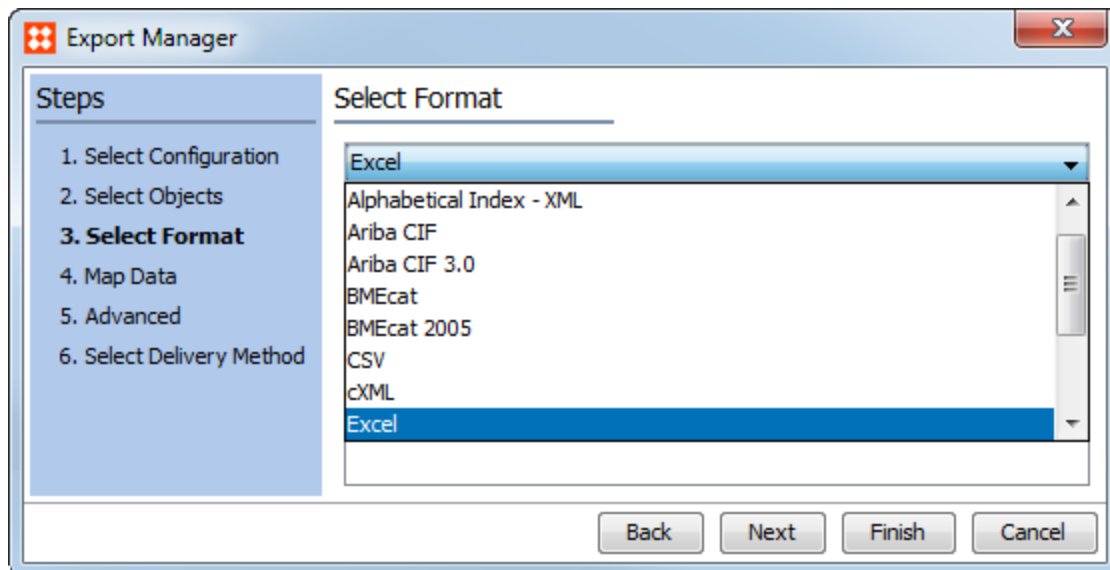
Note: This option is ignored when exporting STEPXML format.

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 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, or **Export Manager - Advanced** if mapping is not needed.

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).

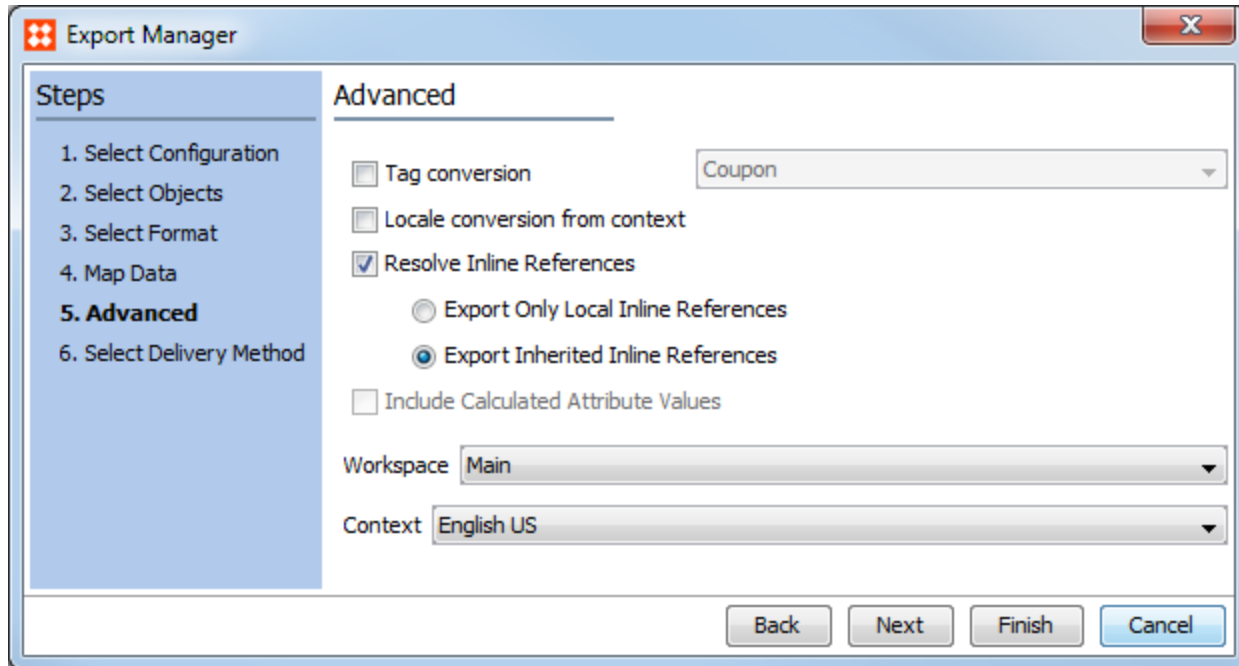
For details on data sources, see the **Outbound Map Data - Data Source** documentation.

For details on mapping targets, see the **Outbound Map Data - Mapping Targets** documentation.

For details on transformations, see the **Outbound Map Data - Transform** documentation.

Clicking the Next button displays **Export Manager - Advanced**.

Export Manager - Advanced



Set the following additional export parameters as needed:

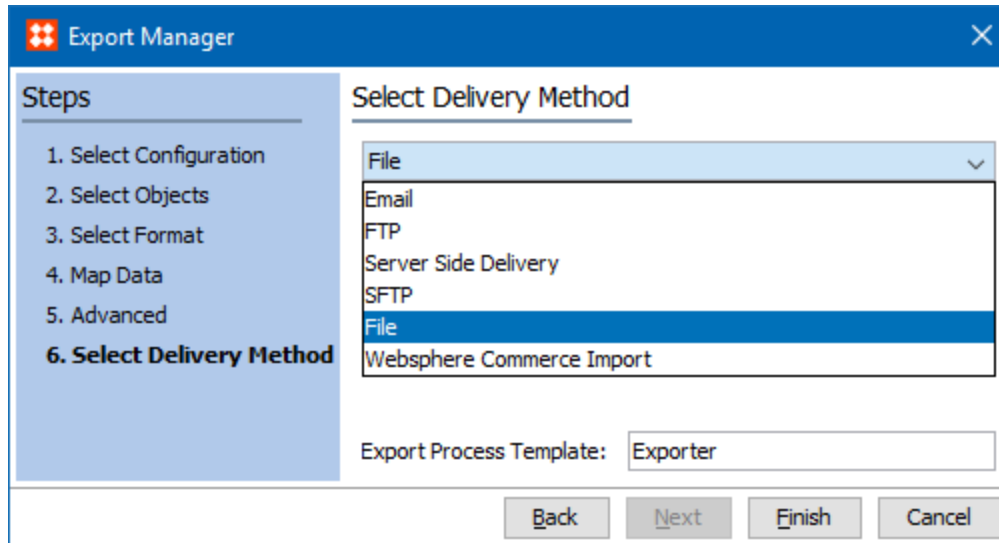
- **Tag conversion:** Optional. Allows for STEP Tags to be converted to match the selected output tag format. For example, the STEP bold tag could be converted to the HTML equivalent. Tags and Tag Conversions are configured under Tags in System Setup. The selection in the dropdown determines the output formatting. For more details see the **Tags** topic and 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. After checking the checkbox, click the ellipsis button (...) and select the preferred locale in the **Select Locale** dialog.

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. Checked 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.
- **Include Calculated Attribute Values:** Resolves calculated attribute values when they are exported. If not checked, calculated attribute values are exported with empty values.
- **Workspace:** Allows the export to run from a different Workspace than the one selected when the Export was started.
- **Context:** Allows the export to run from a different Context than the one selected when the Export was started.

Clicking the Next button displays **Export Manager - Select Delivery Method**.

Export Manager - Select Delivery Method

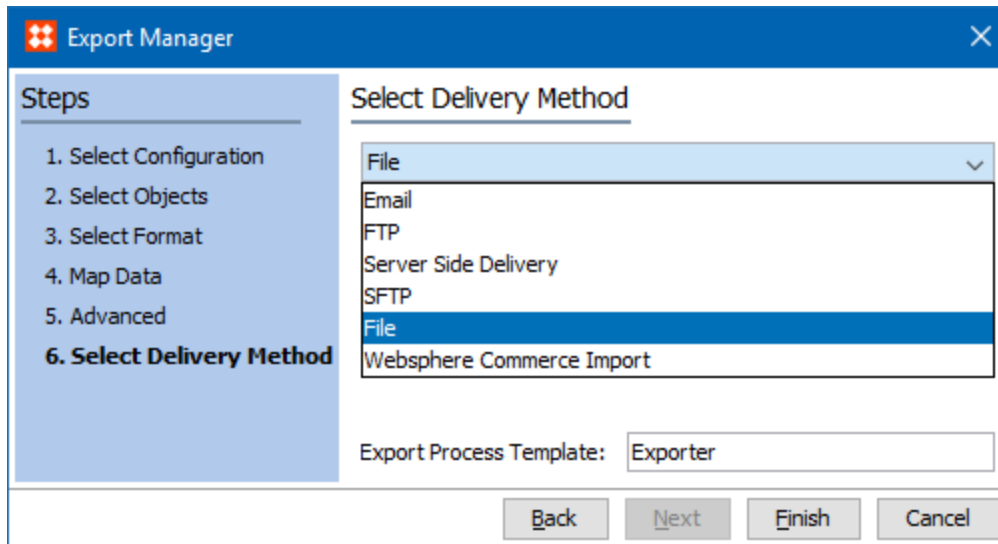


1. In the **File Delivery** list, select the relevant export option. For details about each method, see **Export Manager Delivery Methods**.
 - **Email**: Delivers a file as an email attachment.
 - **File**: Delivers a file in the background process.
 - **FTP**: Delivers a file using file transfer protocol (FTP).
 - **Server Side Delivery**: Exports the file into a folder (with read / write access) located on the Application Server of the STEP system.
 - **SFTP**: Delivers a file using the Secure File Transfer Protocol (SFTP).
 - **Websphere Commerce Import**: Availability depends on your system setup.
2. Click **Finish** to display the Save Export Configuration window as defined in **Running a Data Export**.

Export Manager Delivery Methods

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**.

In Export Manager, the delivery method is selected within the wizard.



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

Email

FTP

Server Side Delivery

SFTP

File

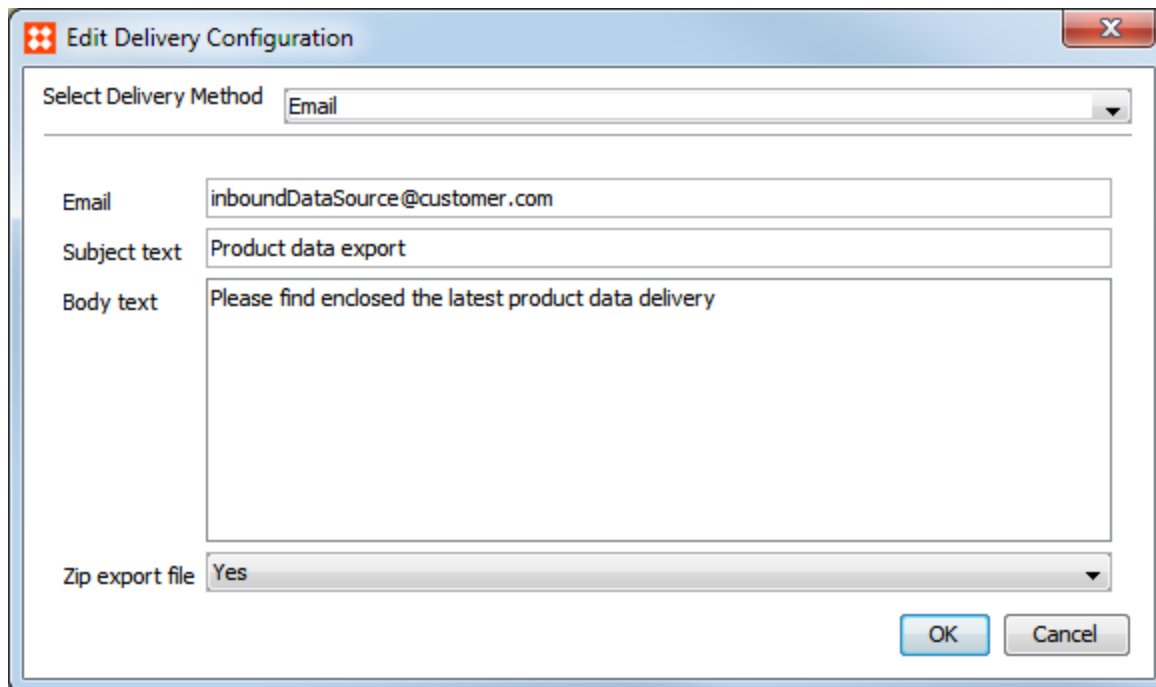
Websphere Commerce Import

Export Process Template:

The following delivery method options are available when exporting using Export Manager.

Method	Description
Email	Delivers files as an email attachment and allows a zipped delivery. See Email Delivery Method documentation.
File	Delivers a file via a background process and notifies when the file is ready for download. See File Delivery Method documentation.
FTP	Delivers files using file transfer protocol (FTP). You must specify the host name, user name, password, and the file name used to deliver the file. See FTP Delivery Method documentation.
Server Side Delivery	Exports a file to a folder located on the STEP system application server. See Server Side Delivery Method documentation.
SFTP	Delivers a file using the Secure File Transfer Protocol (SFTP). You must specify the host name, user name, password, and file name. See SFTP Delivery Method documentation.
Websphere Commerce Import	Delivers a STEPXML file to a Websphere Commerce server. See Websphere Commerce Import Delivery Method documentation.

Email Delivery Method



Edit Delivery Configuration

Select Delivery Method: **Email**

Email:

Subject text:

Body text:

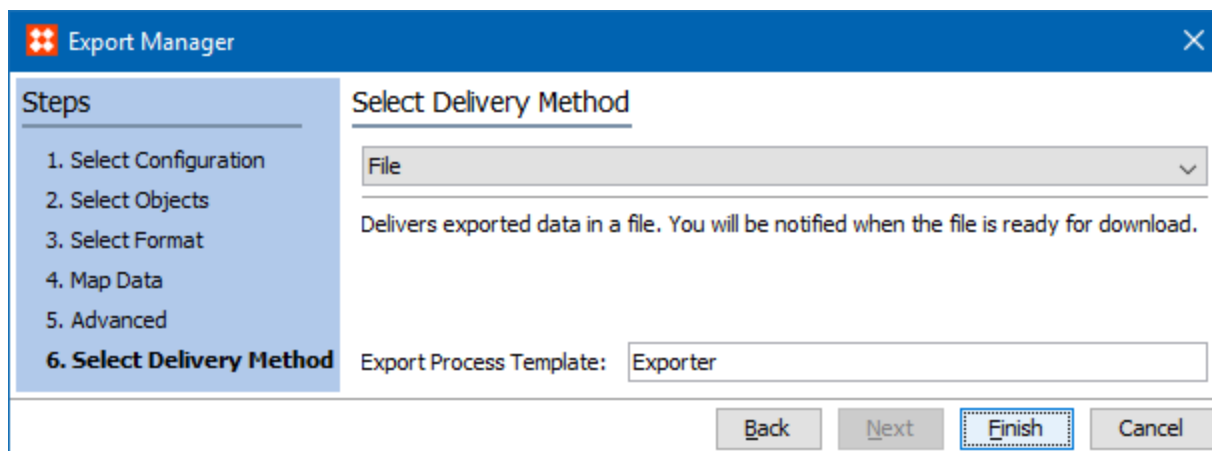
Zip export file: **Yes**

OK **Cancel**

The email delivery option is very straight forward and, like 'Copy to directory', it allows the delivery to be zipped. This delivery method is available in Export Manager and in OIEPs.

1. In **Select Delivery Method**, choose **Email**.
2. Enter an email address, subject text, and body text.
3. In **Zip export file**, specify if you want to deliver the exported contents (email attachment) as a .ZIP file.

File Delivery Method



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.

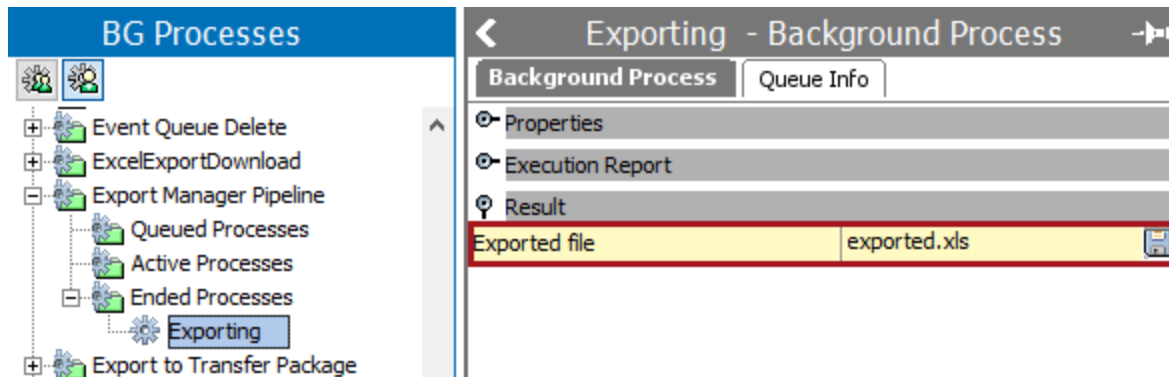
Export Process Template:

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

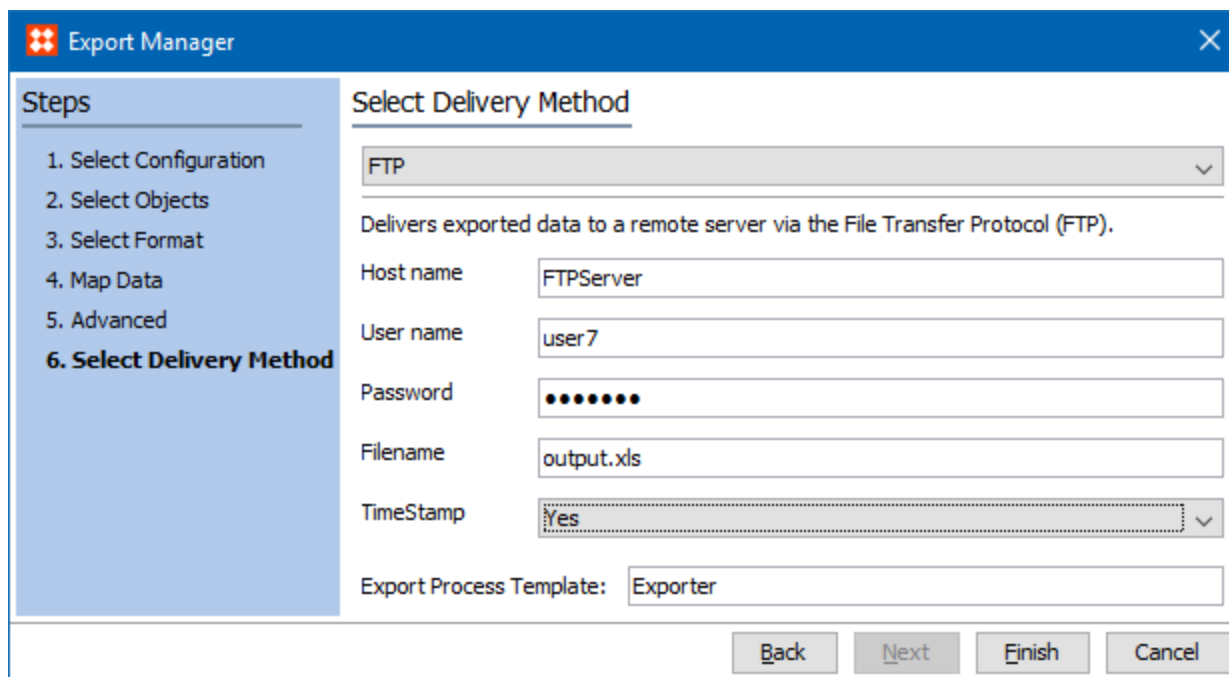
The file delivery option is only available in Export Manager. It makes the output available within STEP, via a background process. There are no additional parameter settings.

For the **Export Process Template** parameter, leave the default setting, which is required to start the background process for the export.

By default, the output is saved on the Background Processes tab, under the Export Manager Pipeline node. While displaying the process, open the Result flipper on the Background Process tab.



FTP Delivery Method



The FTP delivery method allows an exported file to be delivered to an external system. The FTP delivery method in OIEP includes additional options for file names and allows for the file to be zipped before it is uploaded. For more information on the OIEP option, see **FTP Delivery Method** in the **Outbound Integration Endpoints** documentation.

1. In **Select Delivery Method**, choose **FTP** from the dropdown.
2. In **Host name**, type the host name or IP address of the FTP server to be used for the delivery.
3. In **User name**, enter the user name that will be used to log on to the FTP server.
4. In **Password**, enter the password that will be used to log on to the FTP server.
5. In **Filename**, enter a name for the file being delivered. Include the appropriate extension based on the Select Format step. For example, when exporting Excel, using 'output.xls' would allow the receiver to easily open the file with Excel.

Note: Using a static file name and setting the Timestamp parameter to 'no' can result in multiple files with the same name, which causes the background process to fail. To generate unique file names without any static text, add the '\$timestamp' variable to the Filename parameter. The result is a file name of the same string of numbers generated by the timestamp parameter below.

6. In **Timestamp**, select **yes** to create unique exported file names by modifying the static file name with a timestamp. The time is represented in milliseconds between January 1, 1970 and the time when the file is uploaded (e.g., 1438634758430). For example, the delivered file for the screenshot above would be 'output-20172903104220386.xls' when this parameter is set to 'yes,' and 'output.xls' when set to 'no.'
7. In **Export Process Template**, leave the default setting, which is required to start the background process for the export.
8. Click the **Finish** button to complete the wizard and begin the export as described in **Running a Data Export**.

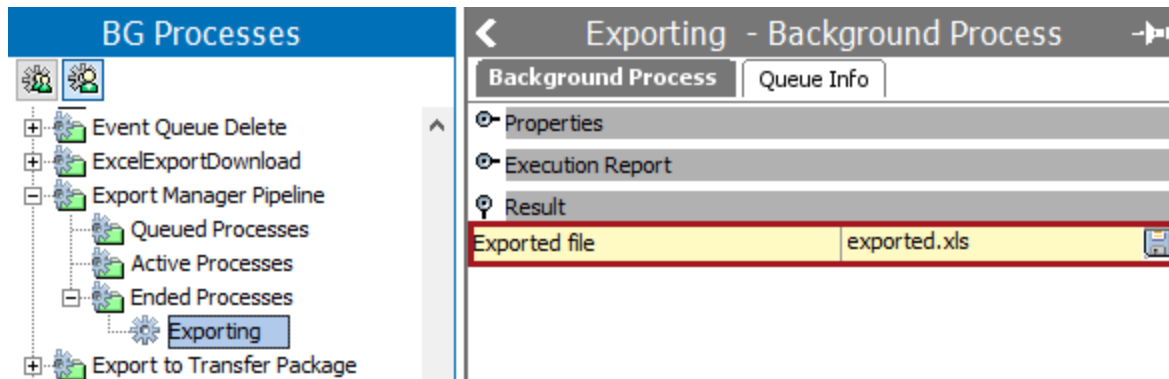
Server Side Delivery Method

The screenshot shows the 'Export Manager' window with the 'Steps' pane on the left. Step 6, 'Select Delivery Method', is active. The main area shows a dropdown menu with 'Server Side Delivery' selected. Below it, the text 'Deliver output file to location on server' is displayed. There are three input fields: 'Path' with the value 'workarea', an empty 'Filename' field, and an 'Export Process Template' field with the value 'Exporter'. At the bottom, there are four buttons: 'Back', 'Next', 'Finish' (which is highlighted with a blue border), and 'Cancel'.

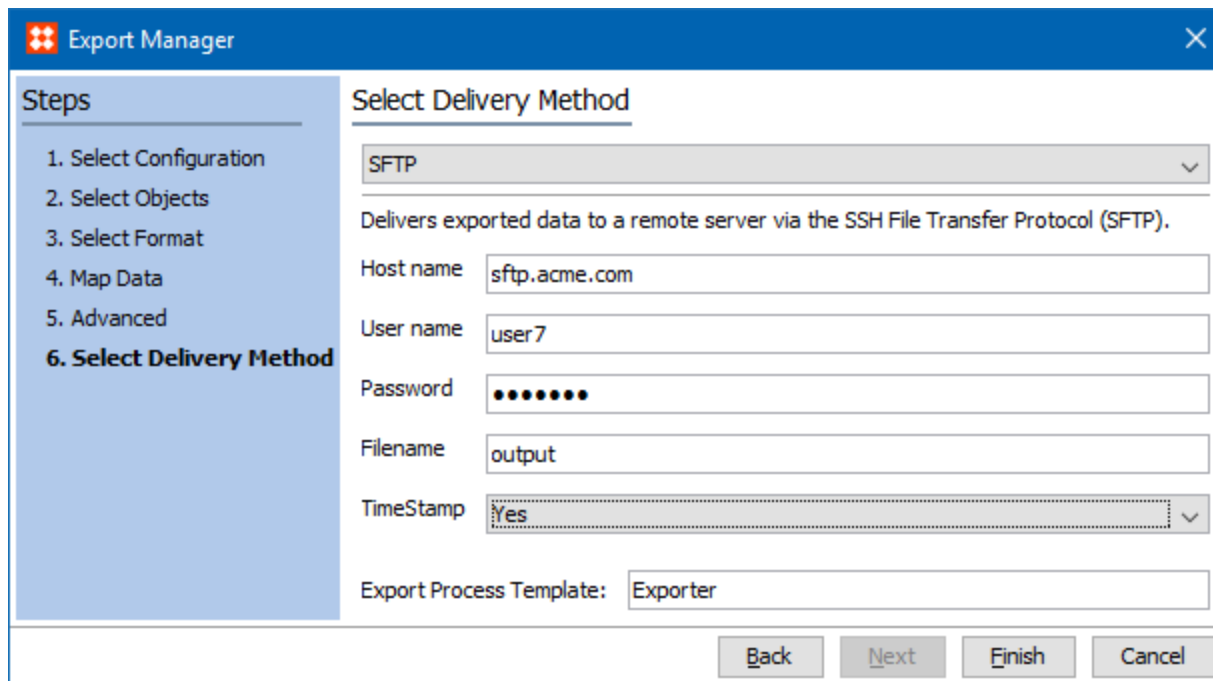
The server side delivery option is only available in Export Manager. It makes the output available within STEP, via a background process.

For the **Export Process Template** parameter, leave the default setting, which is required to start the background process for the export.

By default, the output is saved on the Background Processes tab, under the Export Manager Pipeline node. While displaying the process, open the Result flipper on the Background Process tab.



SFTP Delivery Method



The SFTP delivery method allows an exported file to be delivered to an external system. The SFTP delivery method in OIEP includes additional options for file names and allows for the file to be zipped before it is uploaded. For more information on the OIEP option, see **SFTP Delivery Method** in the **Outbound Integration Endpoints** documentation.

1. In **Select Delivery Method**, choose **SFTP** from the dropdown.
2. In **Host name**, type the host name or IP address of the FTP server to be used for the delivery.
3. In **User name**, enter the user name that will be used to log on to the FTP server.
4. In **Password**, enter the password that will be used to log on to the FTP server.

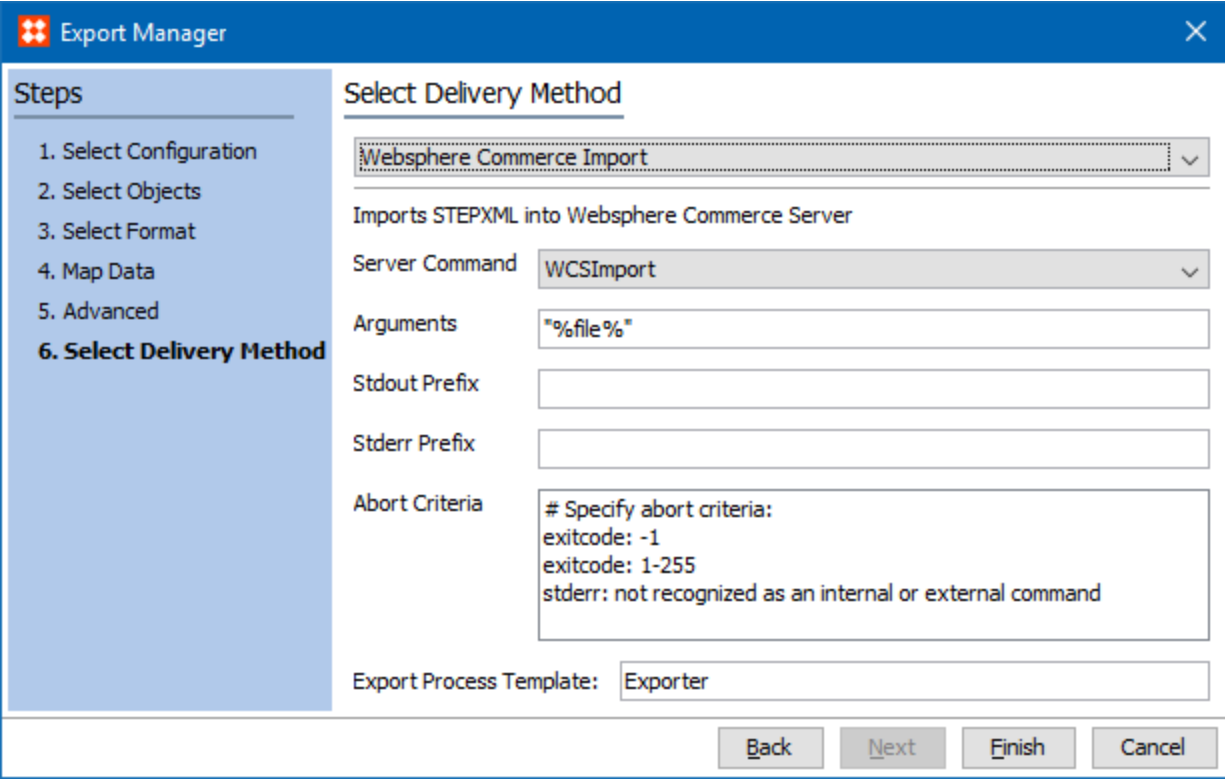
- In **Filename**, enter a name for the file being delivered. Include the appropriate extension based on the Select Format step. For example, when exporting Excel, using 'output.xls' would allow the receiver to easily open the file with Excel.

Note: Using a static file name and setting the Timestamp parameter to 'no' can result in multiple files with the same name, which causes the background process to fail. To generate unique file names without any static text, add the '\$timestamp' variable to the Filename parameter. The result is a file name of the same string of numbers generated by the timestamp parameter below.

- In **Timestamp**, select **yes** to create unique exported file names by modifying the static file name with a timestamp. The time is represented in milliseconds between January 1, 1970 and the time when the file is uploaded (e.g., 1438634758430). For example, the delivered file for the screenshot above would be 'output-20172903104220386.xls' when this parameter is set to 'yes,' and 'output.xls' when set to 'no.'
- In **Export Process Template**, leave the default setting, which is required to start the background process for the export.
- 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.



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

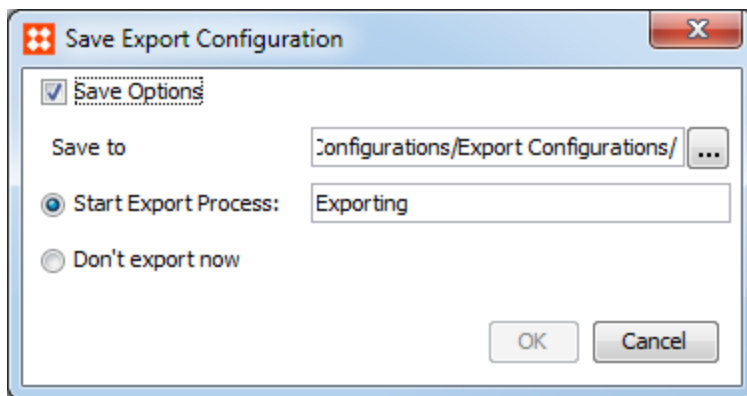
For the **Export Process Template** parameter, leave the default setting, which is required to start the background process for the export.

Running a Data Export

The Save Export Configuration dialog appears when you click Finish in the Export Manager. A 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.

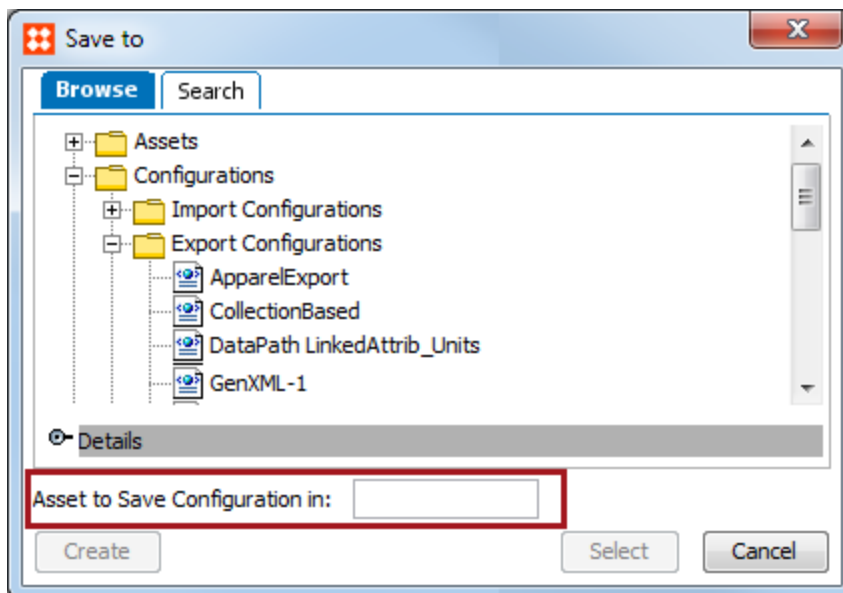
1. Check **Save Options** if:

- You made changes to an existing configuration and want to save those changes.
- You want to save a new configuration.



2. When saving a new configuration, click the ellipsis button (...) on the Save to field to display the **Save to** dialog.

3. In the **Asset to Save Configuration in** field, type the name of the new configuration.



4. Use browse or search and select the folder for the saved configuration. Click the **Create** button that is now enabled.
5. 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.
6. Click **OK** to start the completed the wizard. If you started an export, the **Starting Process** dialog appears. For more information, see **Monitoring a Data Export**.

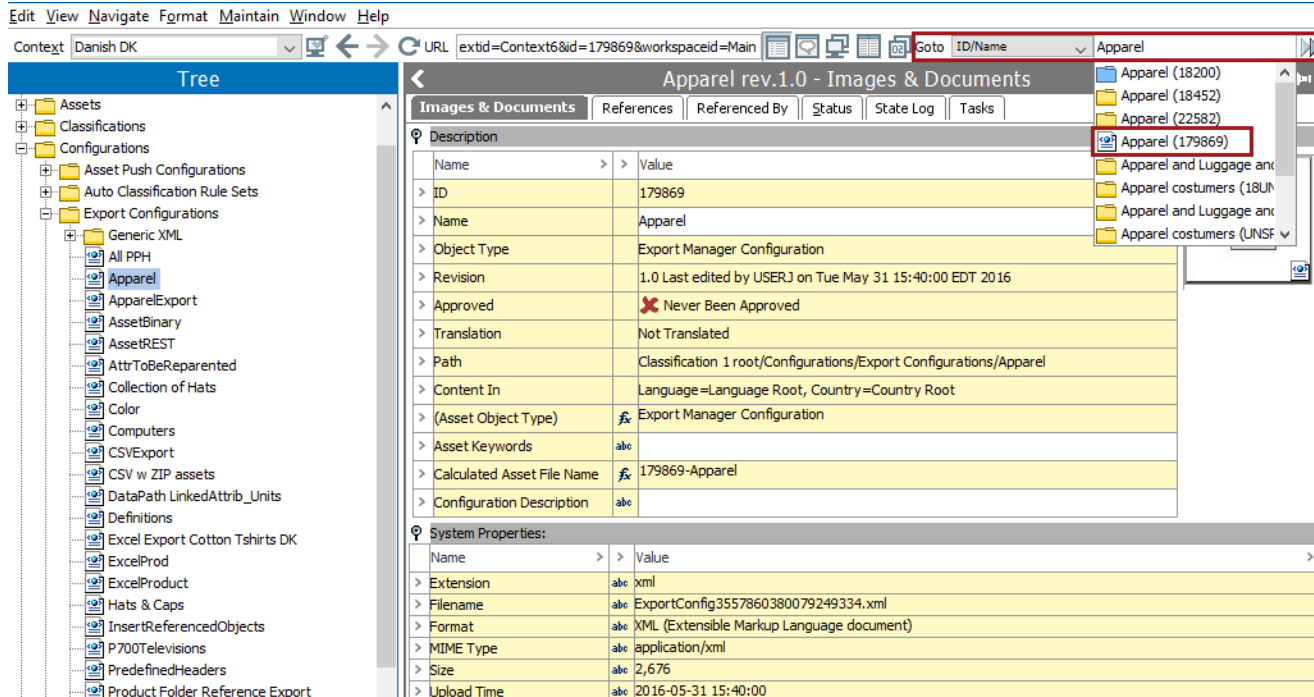
Maintaining a Saved Export Configuration

Completing the Export Manager wizard 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 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 all classification folders are available, using a specific folder for export 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.



The screenshot shows the Stibo Systems interface. On the left, a tree view displays the hierarchy: Context: Danish DK, Assets, Classifications, Configurations, Asset Push Configurations, Auto Classification Rule Sets, Export Configurations, Generic XML, All PPH, Apparel, ApparelExport, AssetBinary, AssetREST, AttrToBeReparented, Collection of Hats, Color, Computers, CSVExport, CSV w ZIP assets, DataPath LinkedAttrib_Units, Definitions, Excel Export Cotton Tshirts DK, ExcelProd, ExcelProduct, Hats & Caps, InsertReferencedObjects, P700Televitions, PredefinedHeaders, Product Folder Reference Export. The 'Apparel' folder is selected. On the right, the 'Apparel rev.1.0 - Images & Documents' configuration details are shown. The 'Goto' search bar is highlighted, showing 'Apparel (179869)'. The configuration details include a table with the following data:

Name	Value
ID	179869
Name	Apparel
Object Type	Export Manager Configuration
Revision	1.0 Last edited by USERJ on Tue May 31 15:40:00 EDT 2016
Approved	Never Been Approved
Translation	Not Translated
Path	Classification 1 root/Configurations/Export Configurations/Apparel
Content In	Language=Language Root, Country=Country Root
(Asset Object Type)	Export Manager Configuration
Asset Keywords	abc
Calculated Asset File Name	179869-Apparel
Configuration Description	abc

Below the configuration details, the 'System Properties' table is shown:

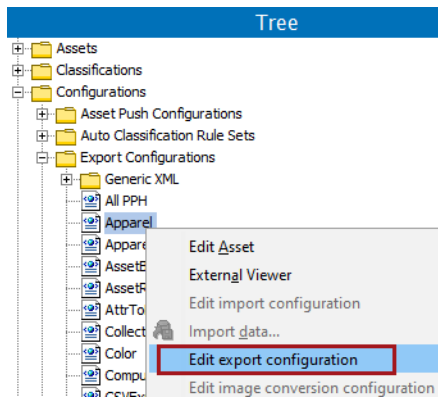
Name	Value
Extension	xml
Filename	ExportConfig3557860380079249334.xml
Format	XML (Extensible Markup Language document)
MIME Type	application/xml
Size	2,676
Upload Time	2016-05-31 15:40:00

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, follow the steps below.

1. Select an export configuration, right-click and select **Edit export configuration** option.

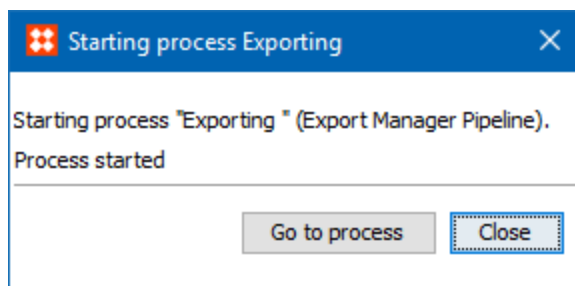


2. The Export Manager wizard opens on step 2 Select Objects.

3. Make all changes and click the **Finish** button to update the selected configuration.

Monitoring a Data Export

When you launch a data export process, 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 the Export Process

There are two ways to monitor a particular data import process:

- In the **Starting Process** window (that appears when you launch the export), click the **Go To Process** button.
- On the **BG Process** 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.

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

- 1 Export process started (Thu May 26 15:37:10 EDT 2016)
- 2 Logging on to PIM server doc-dev as USERJ...
- 3 Logged on
- 4 Analysis started. (Thu May 26 15:37:10 EDT 2016)
- 5 Analyzed 1 objects from initial object selection in 0 seconds. (Thu May 26 15:37:10 EDT 2016)
- 6 Analyzed 0 children objects (including aggregates) in 0 seconds. (Thu May 26 15:37:10 EDT 2016)
- 7 Analyzed 6 initial parents in 0 seconds. (Thu May 26 15:37:10 EDT 2016)
- 8 Analyzed 3 objects referenced from other objects in 0 seconds. (Thu May 26 15:37:11 EDT 2016)
- 9 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.

Download an Export File

If the file delivery option is set to File, you can download the file from the lower part of the Background Process tab. Depending on your system setup, you may also be able to edit the file from within STEP.

1. Click the save icon to the right of the file name.

Exporting - Background Process

Background Process Queue Info

Result

Exported file	exported.xls

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.

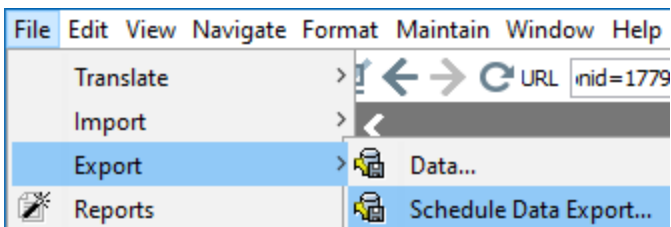
Additional Information About Export Manager

This information is relevant to understanding additional functionality available for implementing and using the Export Manager. Included are details on the following features:

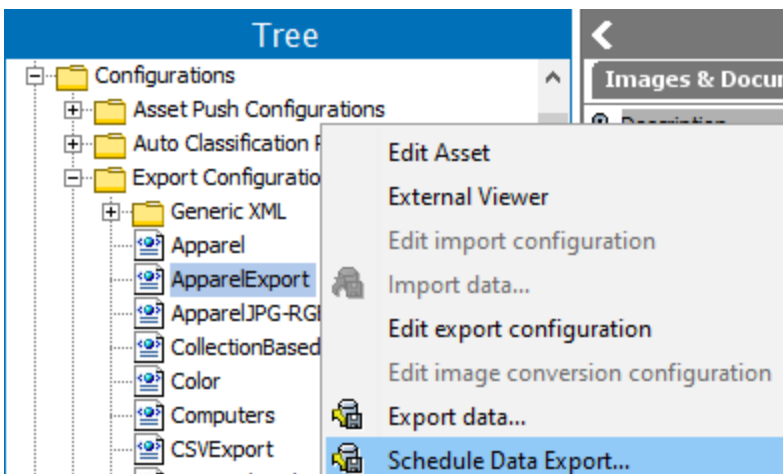
- **Scheduling a Data Export** performs the selected export configuration one or more times via a background process, according to the scheduling parameters.
- **Scheduled Process Properties** displays the status of a scheduled process from queued, to active, and finally when it has ended.
- **Deleting a Scheduled Data Export** removes a scheduled export process, which is required to run current parameters when the original configuration is modified.

Scheduling a Data Export

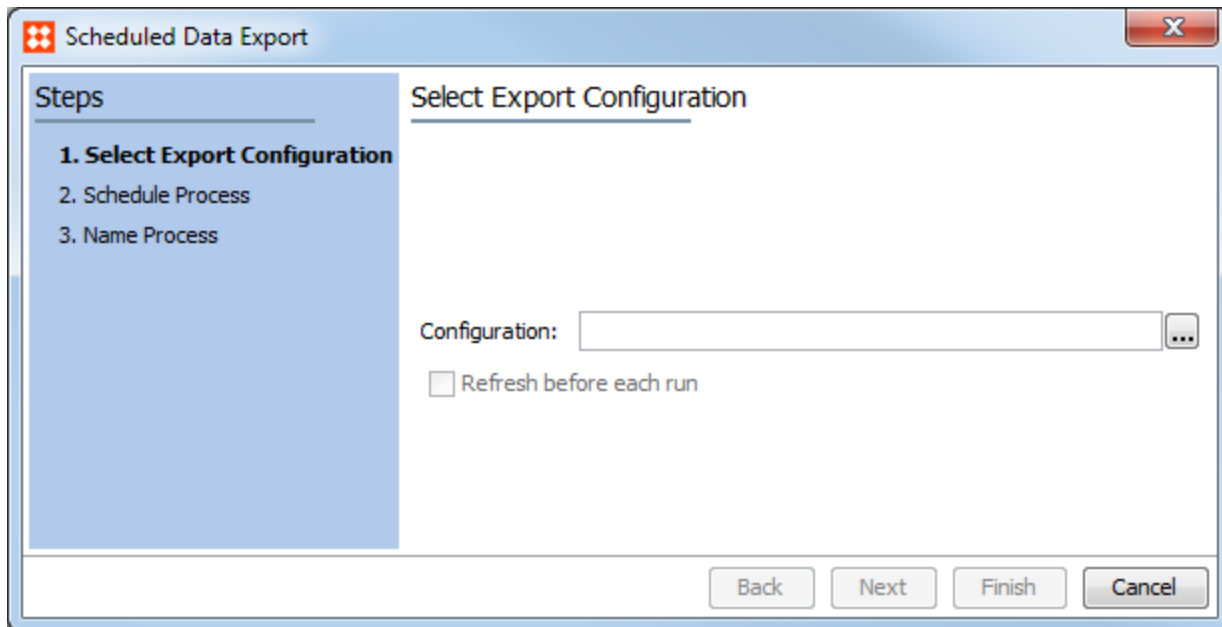
1. Select the appropriate context in STEP.
2. Verify the required Export Configuration already exists for the export you want to schedule. To create a new one, use the steps described in the **Creating a Data Export** documentation and then save the configuration as described in the **Running a Data 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**.

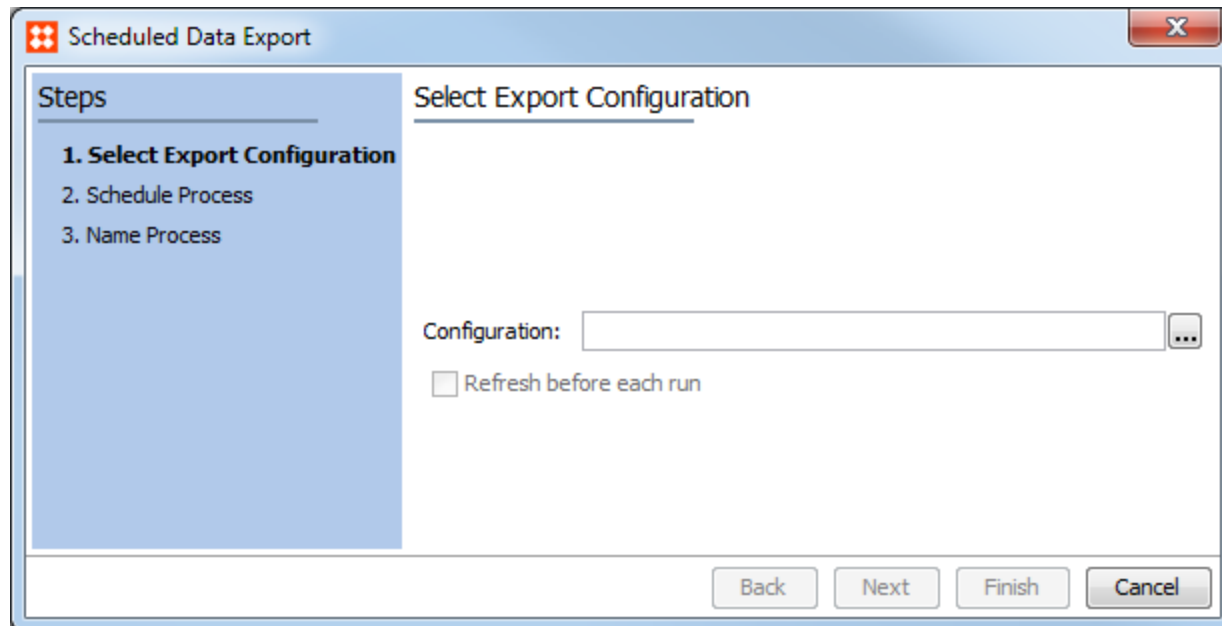


4. After selecting one of the above options to start the Schedule Data Export wizard, the following dialog is displayed and includes these steps:



- **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.

Scheduled Data Export - Select Export Configuration



1. For **Configuration**, click the ellipsis button (...). The Select Export Configuration window displays.

2. Browse or search for the relevant configuration, and click **Select**.
3. If the configuration is based on a Collection, the **Refresh before each run** checkbox is enabled. Check the box to refresh the collection before each scheduled export. Leave it unchecked if you want to only export the objects currently in the collection. For more on collections, see the **Collections** documentation.
4. Click **Next** to display **Scheduled Data Export - Schedule Process**.

Scheduled Data Export - Schedule Process

Scheduled Data Export - DataPath LinkedAttrib_Units

Steps

1. Select Export Configuration
- 2. Schedule Process**
3. Name Process

Schedule Process

Start

☐ Now
 ☐ Later
 ☐ Weekly
 ☒ Monthly
 ☐ Later and repeat

Start at (hh:mm): 09:34
 Start on (yyyy-mm-dd): 2015-11-04
 End on (yyyy-mm-dd): -
 Every: First
 Monday

Start every first mon at 09:34, starting 2015-11-04

Back Next Finish Cancel

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 exports 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

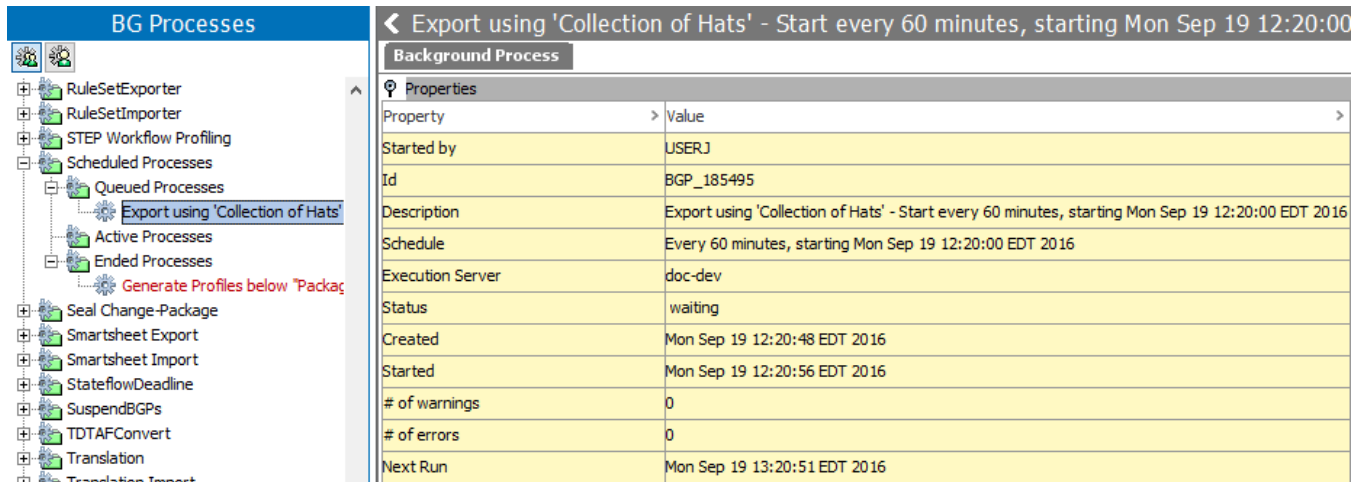
1. For **Schedule Process Name**, enter a name for the scheduled process. This makes it easier for you to locate the process on the BG Processes tab > Scheduled Processes.
2. In the **Process Name** field, enter a name for the exporting process. This makes it easier for you to locate the process on the BG Processes tab > Export Manager Pipeline.
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.



BG Processes

- RuleSetExporter
- RuleSetImporter
- STEP Workflow Profiling
- Scheduled Processes
 - Queued Processes
 - Export using 'Collection of Hats'**
 - Active Processes
 - Ended Processes
 - Generate Profiles below 'Packag
- Seal Change-Package
- Smartsheet Export
- Smartsheet Import
- StateflowDeadline
- SuspendBGPs
- TDATFConvert
- Translation
- Translation Import

Background Process

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

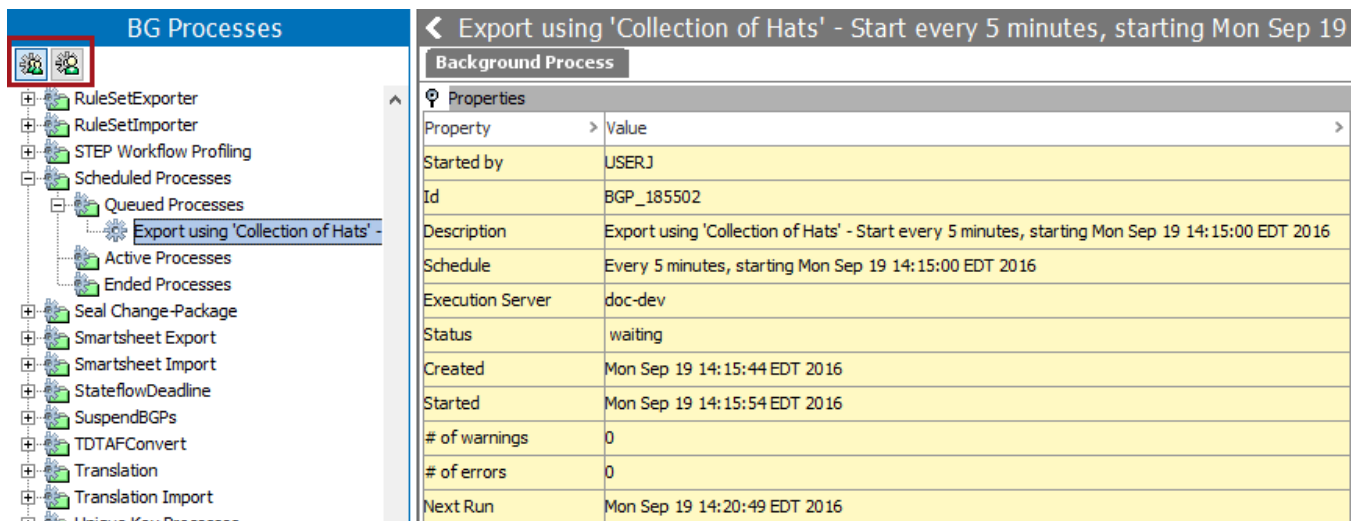
The Background Processes editor tab displays the following data about the selected export:

- **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.

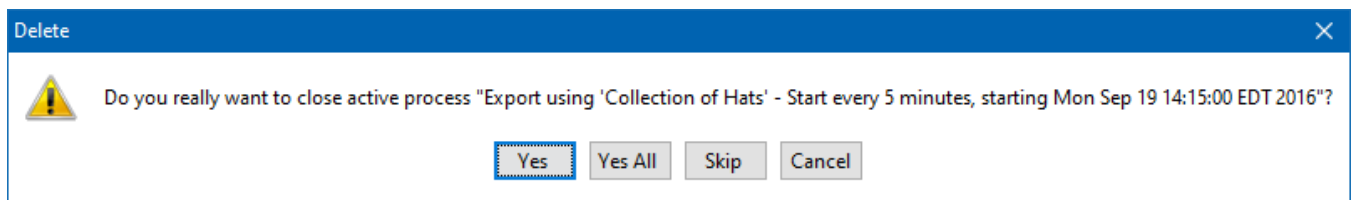
1. On the BG Processes tab, expand Scheduled Processes > **Queued Processes**. If needed, use the filter buttons at the top of the BG Processes tab to display processes started by other users.



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

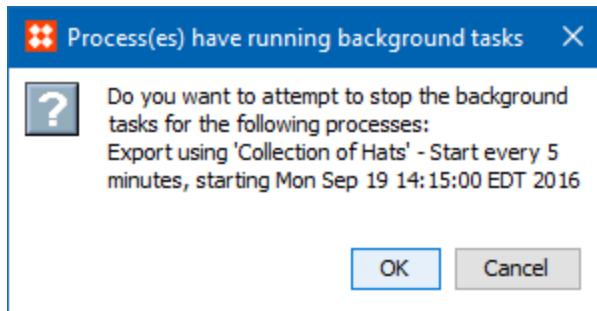
2. Select the one or more processes to delete, and from the Maintain menu, click **Delete** to display the Delete dialog.

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. 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.

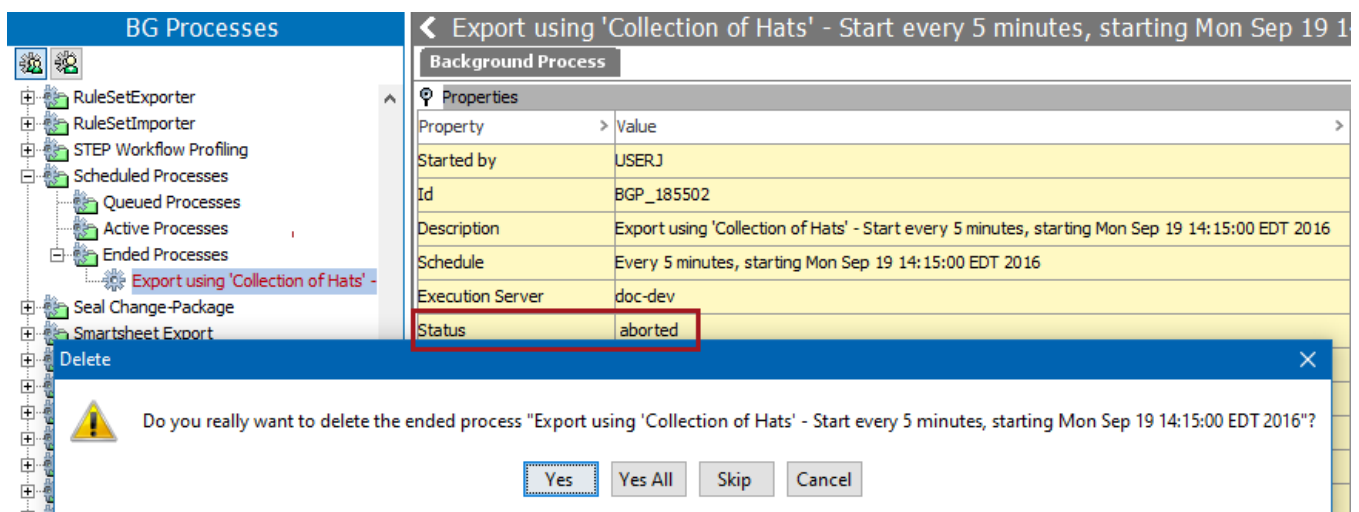
Since deleting a queued process attempts to move it to Ended Processes, you must confirm that you want to stop the background tasks.



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.



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, see the STEP API documentation.

Note: Connecting to a REST destination via an external proxy is not supported. 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. 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:

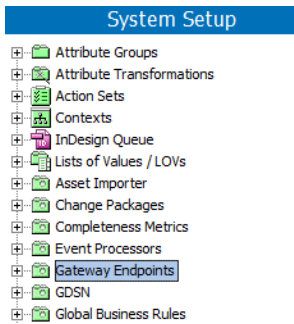
1. Maintain or modify the endpoint as described in **Maintaining a Gateway Integration Endpoint**.
2. Monitor the endpoint as described in **Monitoring a Gateway Integration Endpoint**.
3. Export gateway integration endpoint definition for comparison purposes in an external source control system in **Exporting 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 / STEP Super User** documentation.

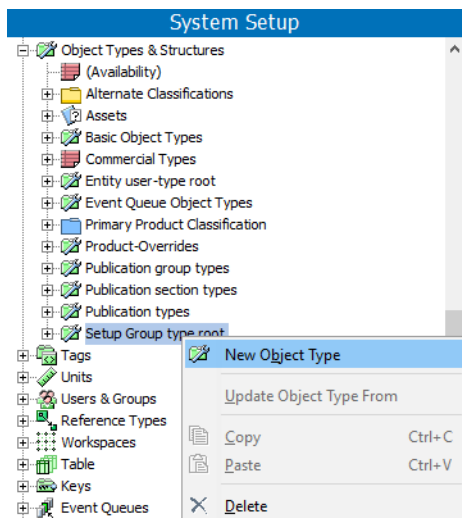
Review your System Setup tab to determine if one or more gateway integration endpoint nodes already exist. The name of the node on your system should include the word 'gateway' but is not required to match the one in the image below.



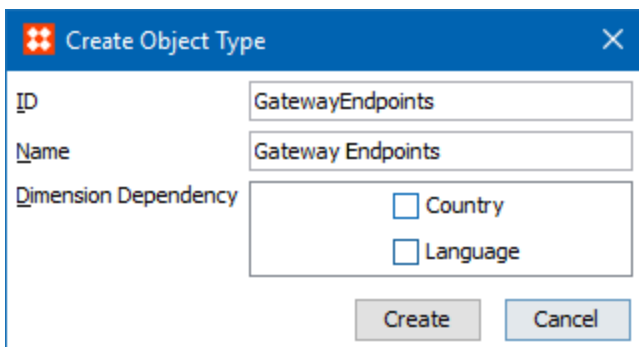
Once the setup has been completed, the steps in this section are only needed if additional levels of organization are desired.

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**.

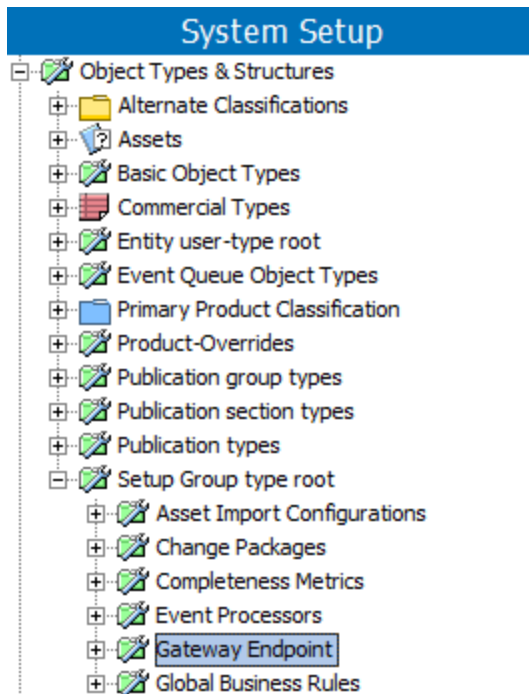


3. Enter an **ID** and a **Name**, select any required Dimension Dependencies, and click **Create**.



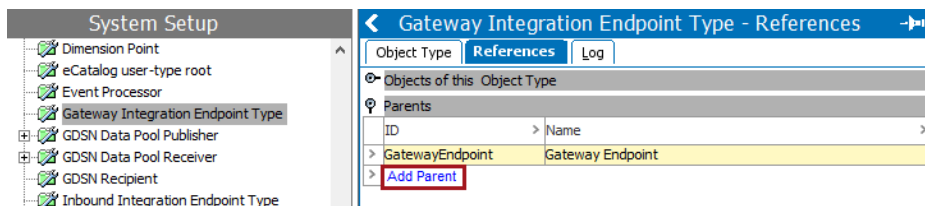
The screenshot shows the 'Create Object Type' dialog box. The 'ID' field contains 'GatewayEndpoints' and the 'Name' field contains 'Gateway Endpoints'. Under 'Dimension Dependency', the 'Country' and 'Language' checkboxes are unchecked. The 'Create' button is highlighted.

- The new setup group appears in System Setup under 'Object Types & Structures' as a child in the **Setup Group type root**.

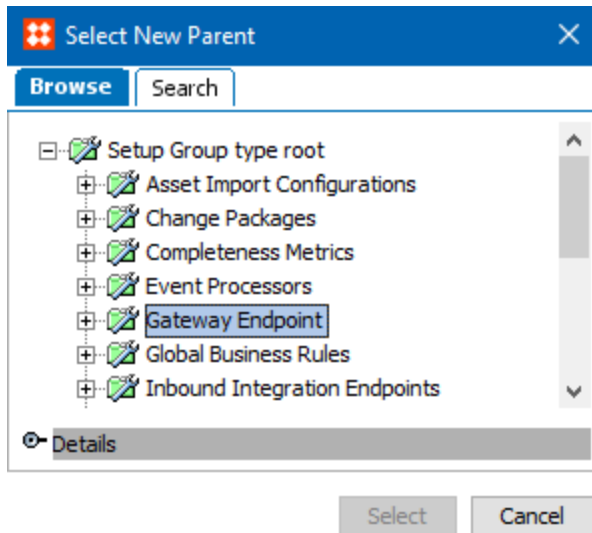


Link the Gateway Object Type to Setup Group

- In Object Types & Structures > expand Basic Object Types > select **Gateway Integration Endpoint Type**.
- On the References tab > Parents flipper > click the **Add Parent** link.

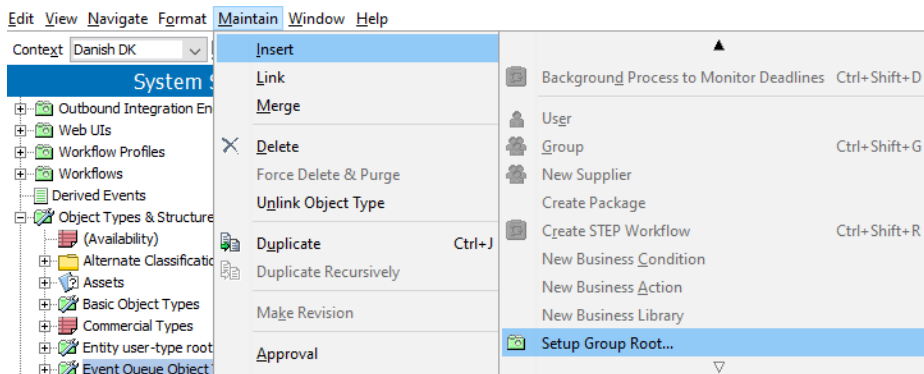


- In the Select New Parent dialog, select the setup group you created, and click **Select** to make it a valid parent.

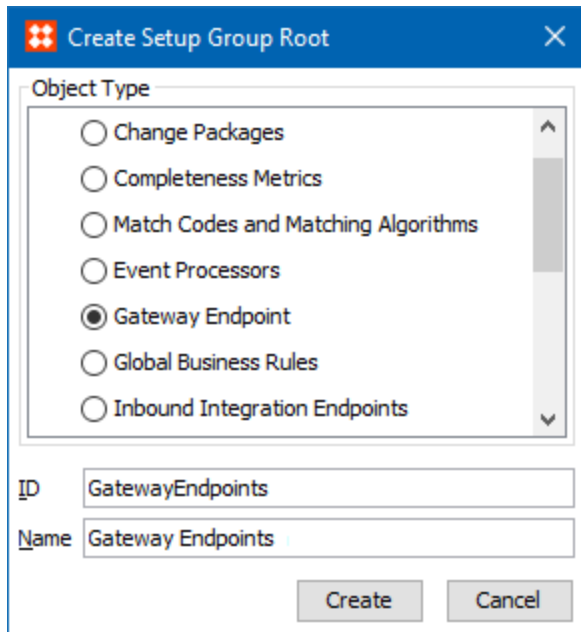


Create an Instance of the Gateway Endpoint Object

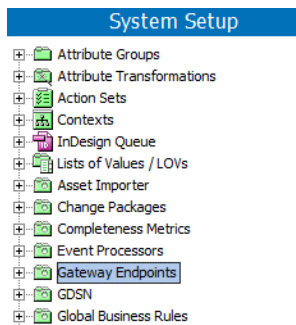
1. On the System Setup tab, select any object in the **System Setup** hierarchy to activate the following Maintain menu selection.
2. Click the Maintain menu, point to Insert, and select **Setup Group Root**.



3. In the Create Setup Group Root dialog, select the gateway object type, enter an **ID** and a **Name**, and click **Create**.



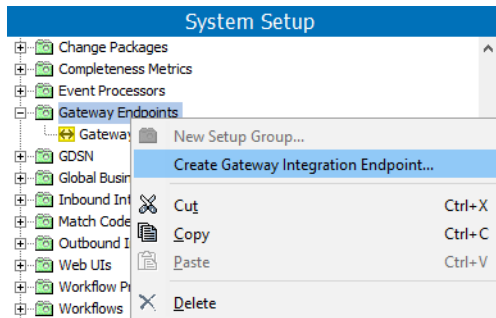
A setup group is created as a node in the System Setup hierarchy. Gateway integration endpoints can now be created under this new node.



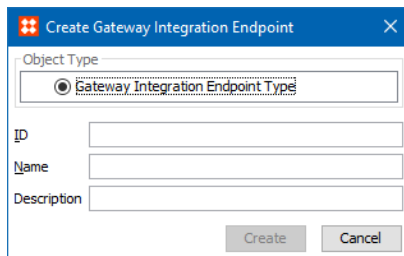
Creating a Gateway Integration Endpoint

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.

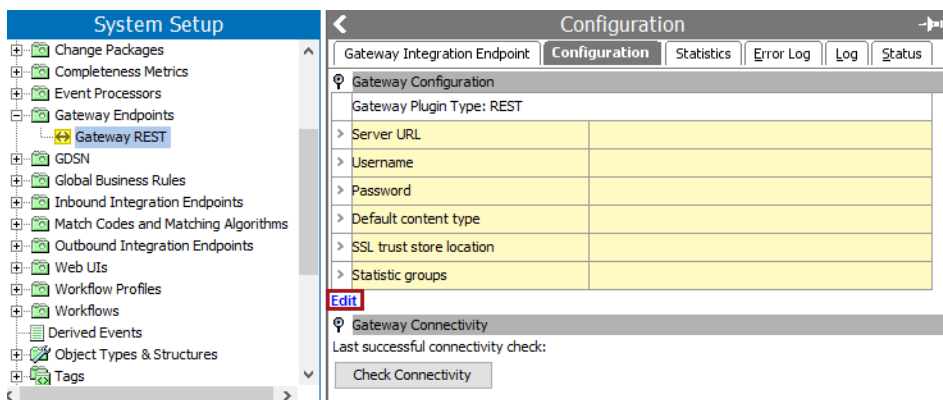


3. Click **Create** and then configure the endpoint. For more information, see the **Configuring a Gateway Integration Endpoint** documentation.

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, click the Gateway Endpoints setup group, and select the relevant gateway integration endpoint.
2. On the Configuration tab, click the **Edit** link.



3. On the Gateway Integration Endpoint Configuration dialog, use the following parameters to specify which external system the gateway integration endpoint will access.

- **Server URL** - The URL of the server to be accessed. To display a Server URL in the dropdown, first update the 'RESTGateway.ServerURL' property in the sharedconfig.properties file. Contact Stibo Systems technical support for assistance in adding a Server URL to your system.
- **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.

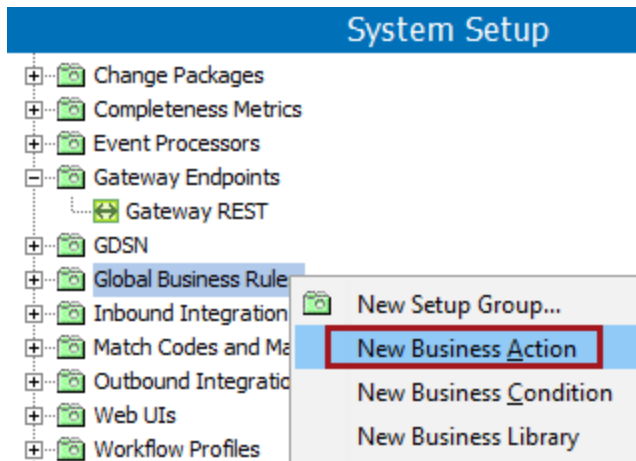
If no trust store locations are selected, the gateway connection is not configured to use SSL encryption and the connection will be established using a certificate signed by a recognized CA such as Verisign or Thawte.

- **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.
4. Click **Save** to complete the configuration.
 5. Create a business action to access the gateway endpoint. For details, see **Business Action Accesses Gateway Integration Endpoint**.

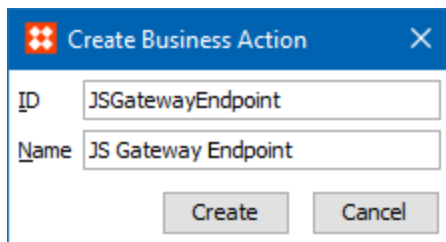
Business Action Accesses Gateway Integration Endpoint

Before a business action can access a gateway endpoint, use the following steps to first create the business action and then bind the gateway endpoint into the JavaScript code.

1. In System Setup, select the Business Rules setup group, and then right-click and select **New Business Action**.



2. Add an **ID** and a **Name** for the new business action and click the **Create** button.



3. Right-click the new business action and choose **Edit Business Rule**, or click the Edit Business Rule link on the editor.

System Setup

- Change Packages
- Completeness Metrics
- Event Processors
- Gateway Endpoints
 - Gateway REST
- GDSN
- Global Business Rules
 - CIC status updater
 - Execute Pending File
 - JS Gateway Endpoint
 - Register
 - Register
 - Set Pending
 - UnRegister
 - UnRegister
- Actions
 - BRLib
- Condition
- GDSN
- Libraries
- Inbound Integration Endpoints

JS Gateway Endpoint rev.0.1 - Business Rule

Business Rule | Usage | Statistics | Log | Status

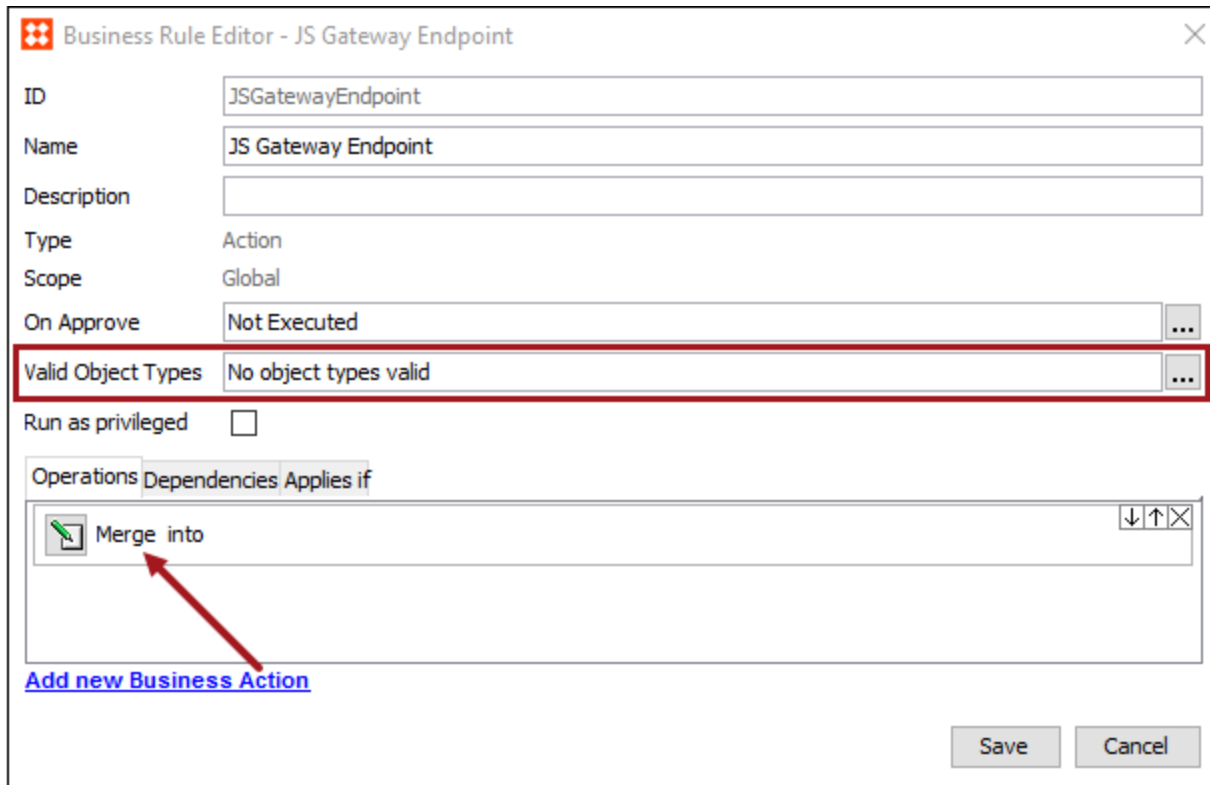
Name	Value
ID	JSGatewayEndpoint
Name	JS Gateway Endpoint
Revision	0.1 Last edited by USERJ on Mon No
Description	
Type	Action
Valid Object Types	No object types valid
Applies if	Not Executed
Global	Global
Enabled	☐

Dependencies | Applies if

Edit Business Rule

4. On the Business Rule Editor dialog:

- For the **Valid Object Types** parameter, choose the object types where business action is allowed to be executed.
- Click the **Add New Business Action** link to add an operation.



Business Rule Editor - JS Gateway Endpoint

ID: JSGatewayEndpoint

Name: JS Gateway Endpoint

Description:

Type: Action

Scope: Global

On Approve: Not Executed

Valid Object Types: No object types valid

Run as privileged: ☐

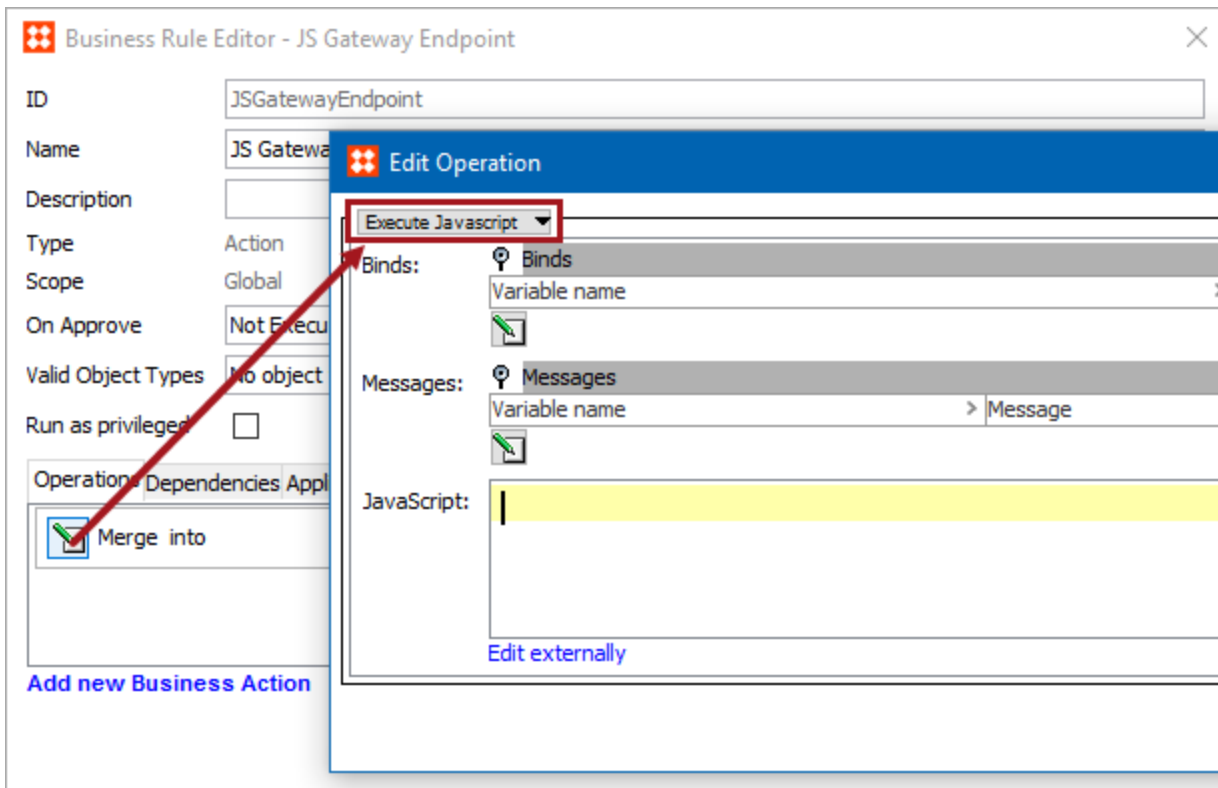
Operations: Dependencies Applies if

Merge into

[Add new Business Action](#)

Save Cancel

5. Edit the new operation and select **Execute JavaScript** from the operation dropdown.



Business Rule Editor - JS Gateway Endpoint

ID: JSGatewayEndpoint

Name: JS Gateway

Description:

Type: Action

Scope: Global

On Approve: Not Executed

Valid Object Types: No object types

Run as privileged: ☐

Operations: Dependencies Applies if

Merge into

[Add new Business Action](#)

Edit Operation

Execute Javascript

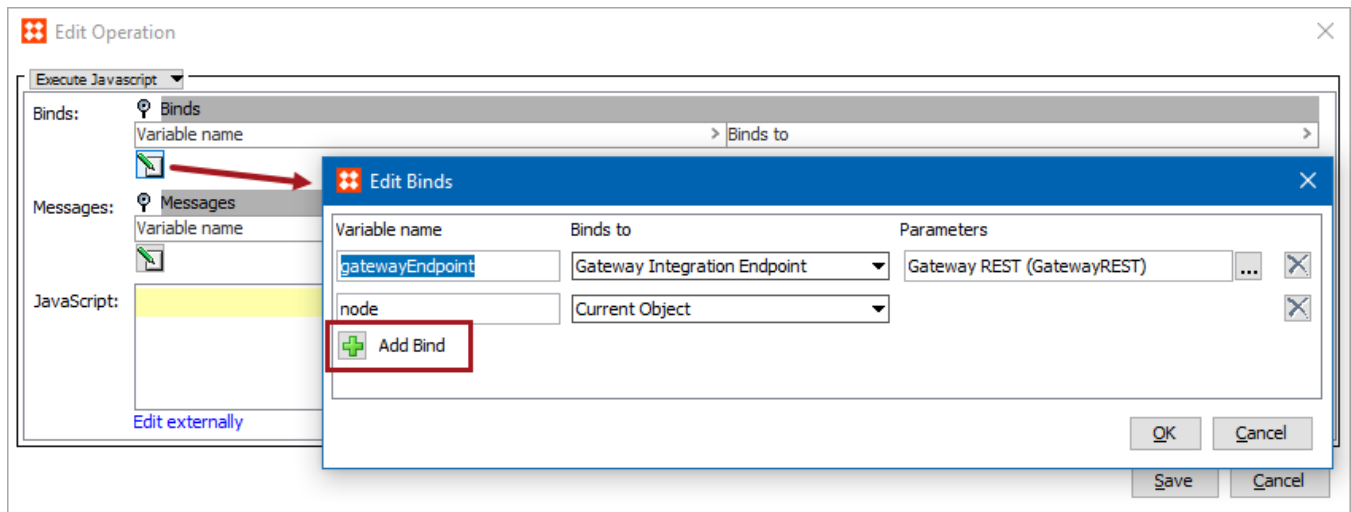
Binds: Variable name

Messages: Message

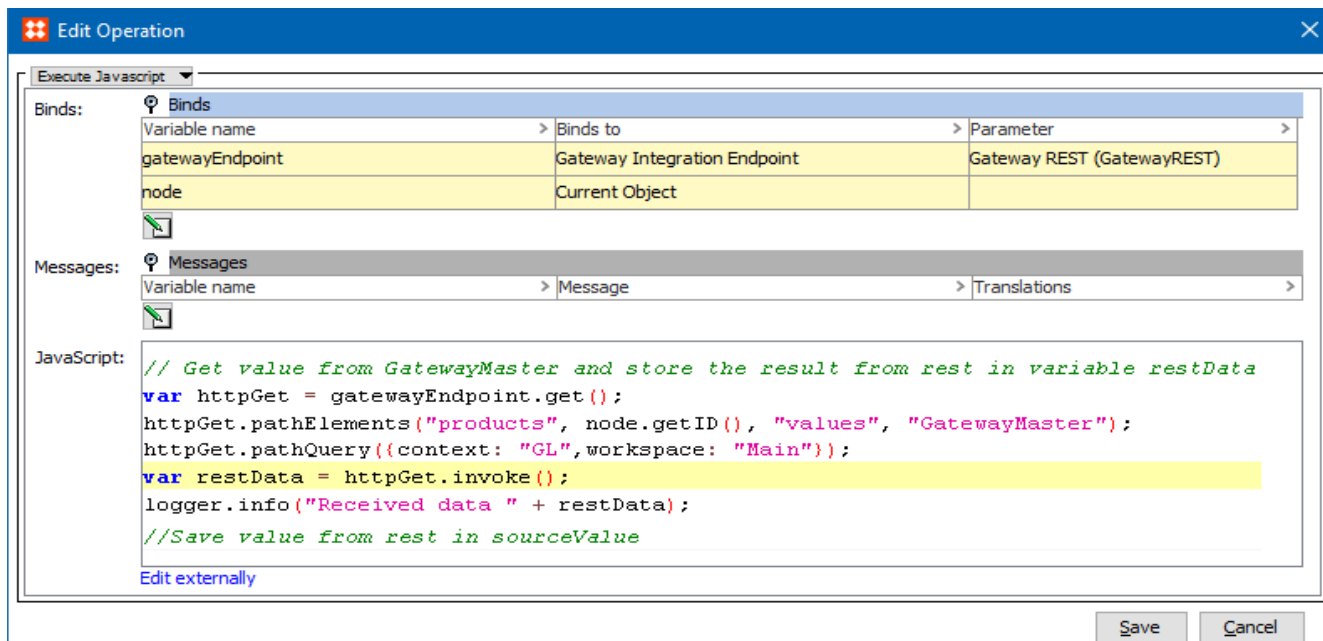
JavaScript:

[Edit externally](#)

- Click the **Edit Binds** button to display the Edit Binds dialog, then click the **Add Bind** button. Bind the JavaScript to the relevant gateway endpoint and click **OK**.



- In Edit Operation dialog, enter the JavaScript required to access the gateway. The following example uses a REST gateway integration endpoint to access the external server for information. For more information about the available REST methods, see the **Gateway Integration Endpoint Bind** section.



- In the **Edit Operations** dialog, click the **Save** button.

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.

Gateway Integration Endpoint Bind

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 table below.

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>

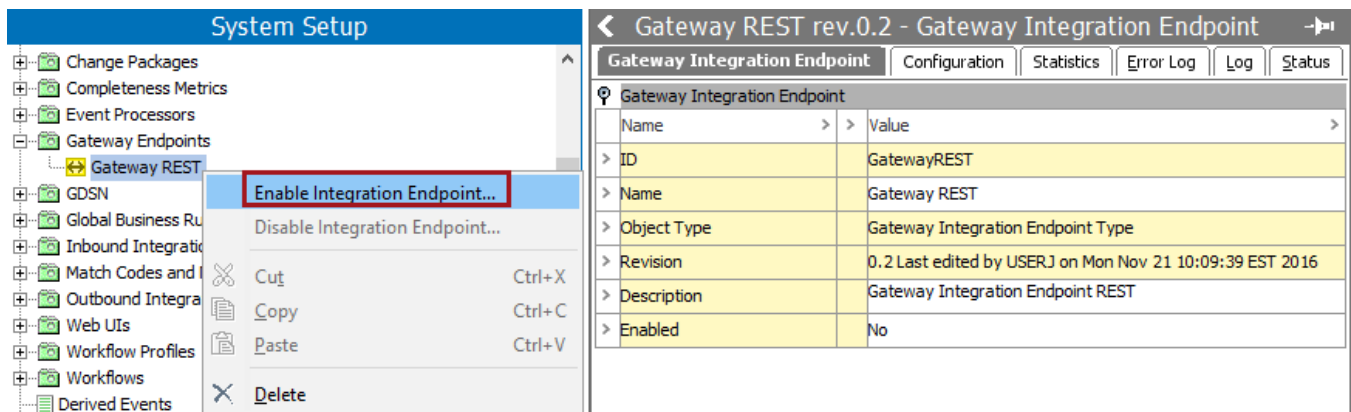
For more information, see the **Business Action Accesses Gateway Integration Endpoint** section.

For more information about the available methods, see the STEP API documentation.

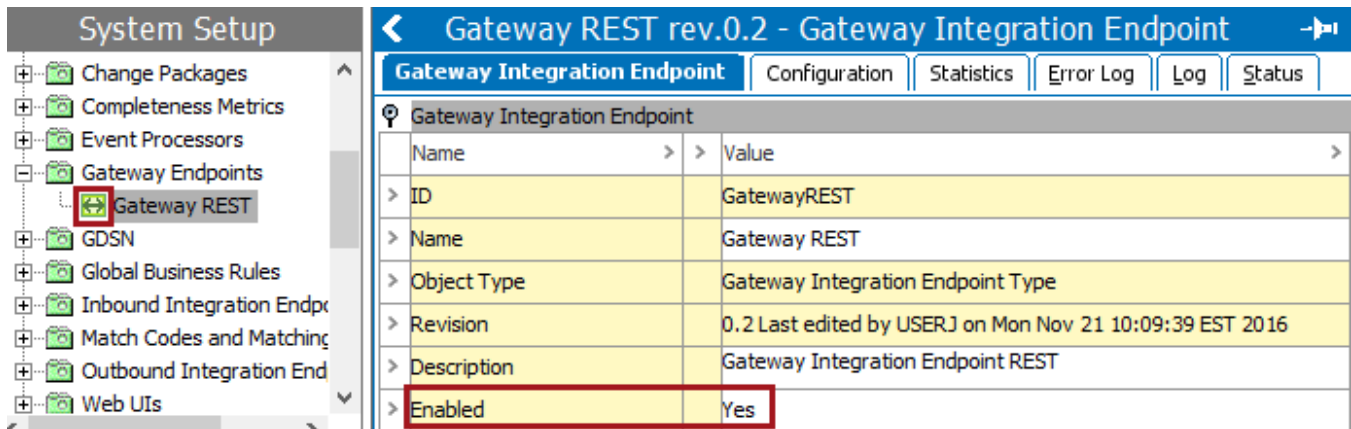
Running a Gateway Integration Endpoint

After configuring a gateway endpoint, it must be enabled and connected before it can run.

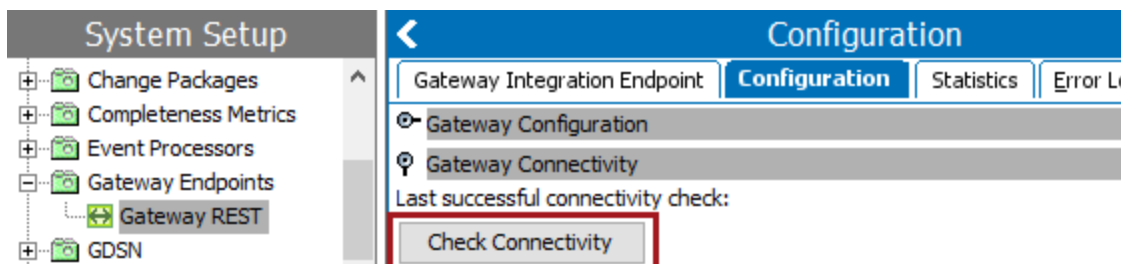
1. On System Setup expand the Gateway Endpoints node to display all existing gateway endpoints.
2. Select a disabled gateway endpoint (indicated by the yellow gateway icon  meaning it has been disabled by user request), right-click, and select **Enable Integration Endpoint** from the menu.



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'.

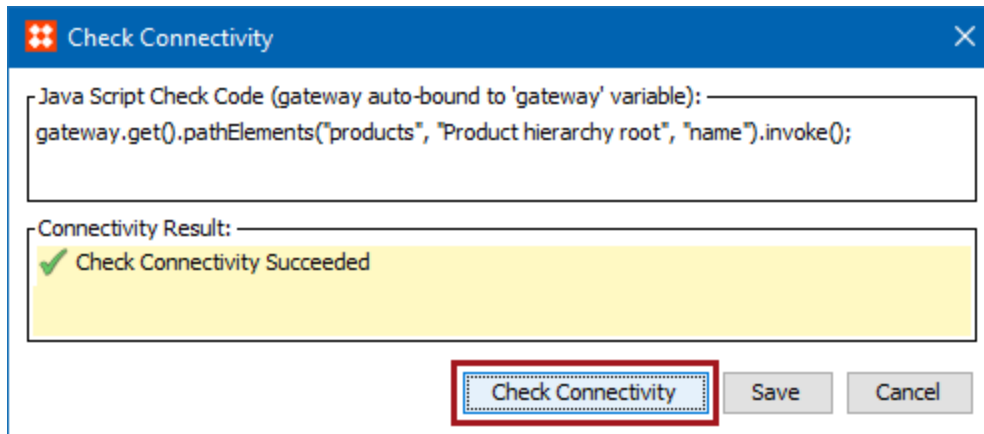


- On the Configuration tab under the Gateway Connectivity flipper, click the **Check Connectivity** button.



- 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, see the STEP API documentation.



Check Connectivity

JavaScript Check Code (gateway auto-bound to 'gateway' variable):

```
gateway.get().pathElements("products", "Product hierarchy root", "name").invoke();
```

Connectivity Result:

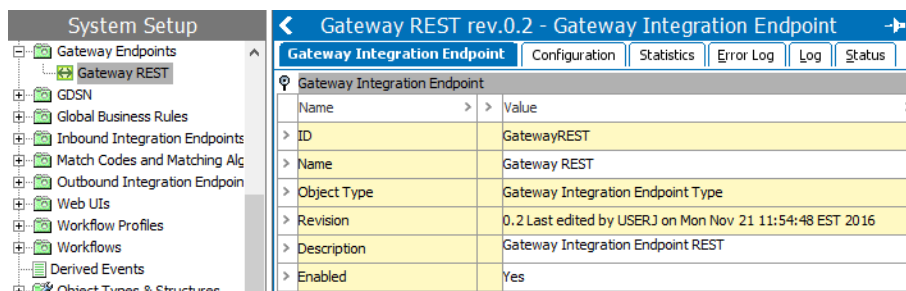
✓ Check Connectivity Succeeded

Check Connectivity Save Cancel

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.



System Setup

- Gateway Endpoints
 - Gateway REST
- 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

Gateway REST rev.0.2 - Gateway Integration Endpoint

Gateway Integration Endpoint | Configuration | Statistics | Error Log | Log | Status

Name	Value
ID	GatewayREST
Name	Gateway REST
Object Type	Gateway Integration Endpoint Type
Revision	0.2 Last edited by USERJ on Mon Nov 21 11:54:48 EST 2016
Description	Gateway Integration Endpoint REST
Enabled	Yes

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.

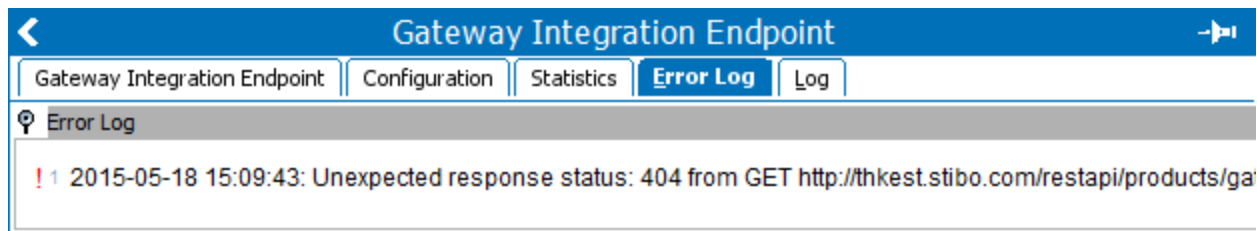
Monitoring a Gateway Integration Endpoint

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.

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: [IEP definition goes here. Removed for brevity.-->]
[Remaining configuration would populate here]
</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 **Include Integration End Points** 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. A saved import configuration 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

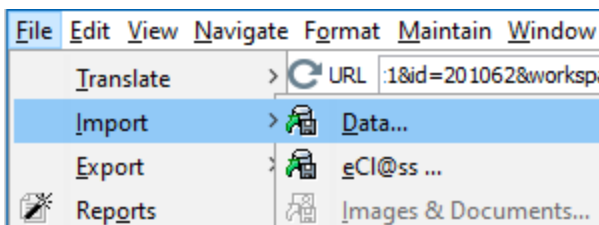
The Import Manager wizard allows you to import data on demand and to create or modify an import configuration. An import configuration is useful for frequently-used imports where the parameters and options are always, or nearly always, the same.

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:

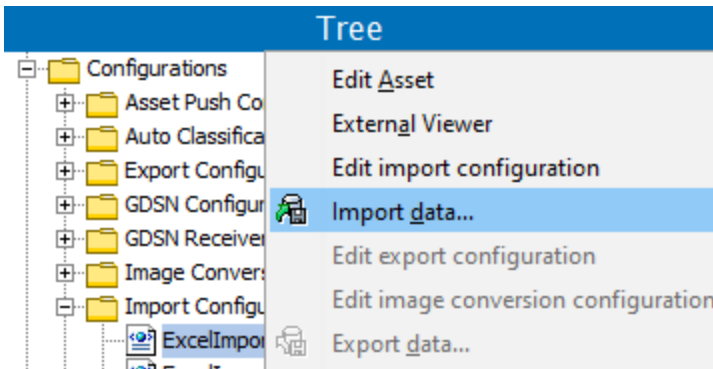
1. Verify that all attributes and reference / link types being imported already exist in STEP.
2. Use one of the following methods to launch the Import Manager wizard, noting that using an import configuration pre-loads the import file and settings.
 - **From the File Menu:** Click the File menu > Import > **Data**.

The wizard opens in Step 1 and allows you to select an existing configuration or create a new one.

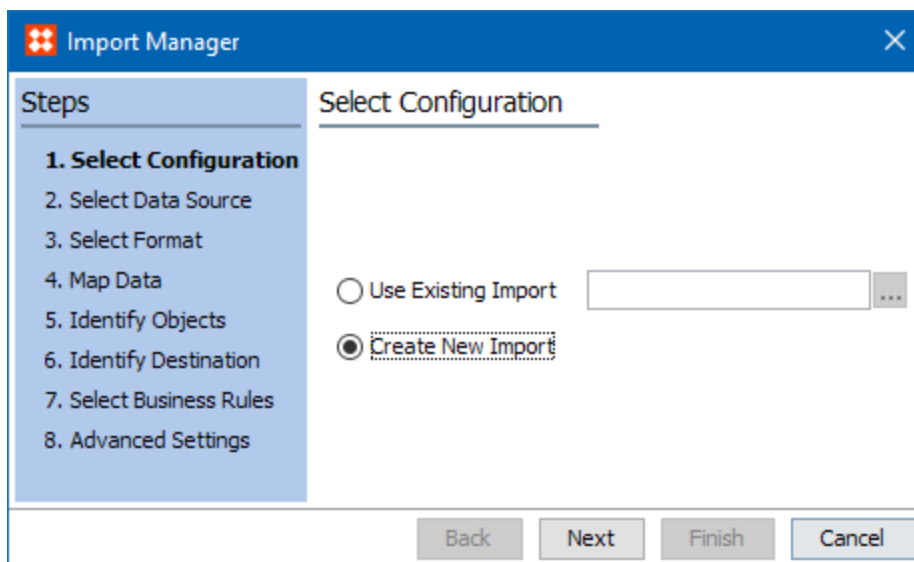


- **From an Import Configuration:** On the Tree, select an import configuration, right-click, and then select **Import Data**.

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 FTP or an ODBC 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.

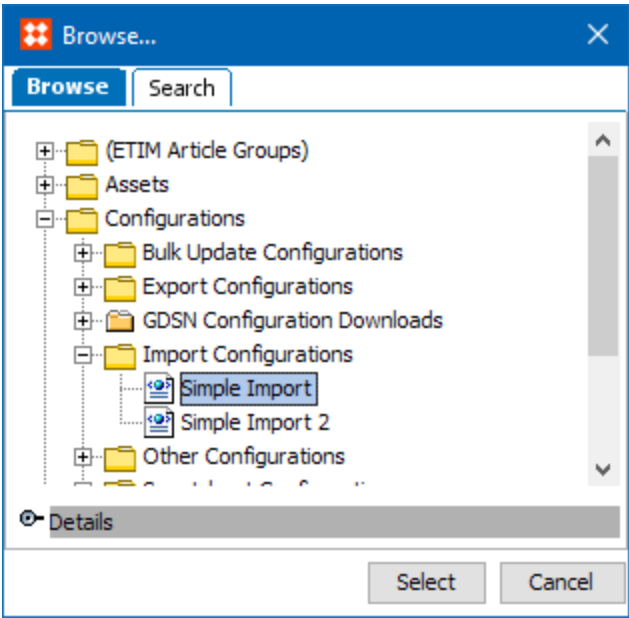
Import Manager - Select Configuration

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.

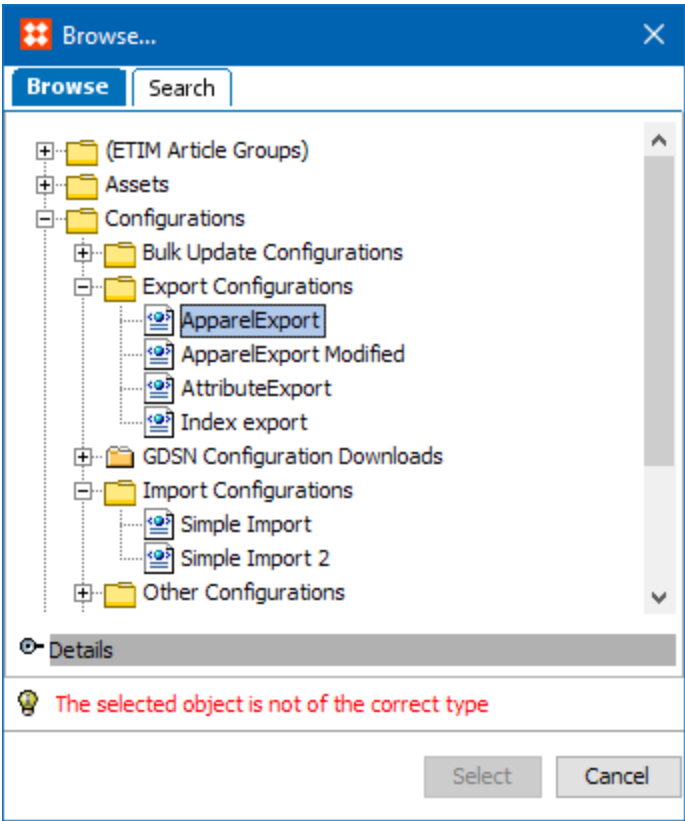
1. Select a configuration option:
 - **Use Existing Import** - allows you to select a previously saved configuration. Click the ellipsis button (...) to browse or search for the configuration in STEP.
 - To create a new configuration, Select **Create New Import**.
2. Click **Next** to display **Import 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:

- FTP allows you to select a file from an FTP site.
- ODBC (Open Database Connectivity) allows you to connect to an external database.
- 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 ODBC

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:

Retrieves data from a database using an Open Database Connection (ODBC).

Data Source:

User:

Password:

Query:

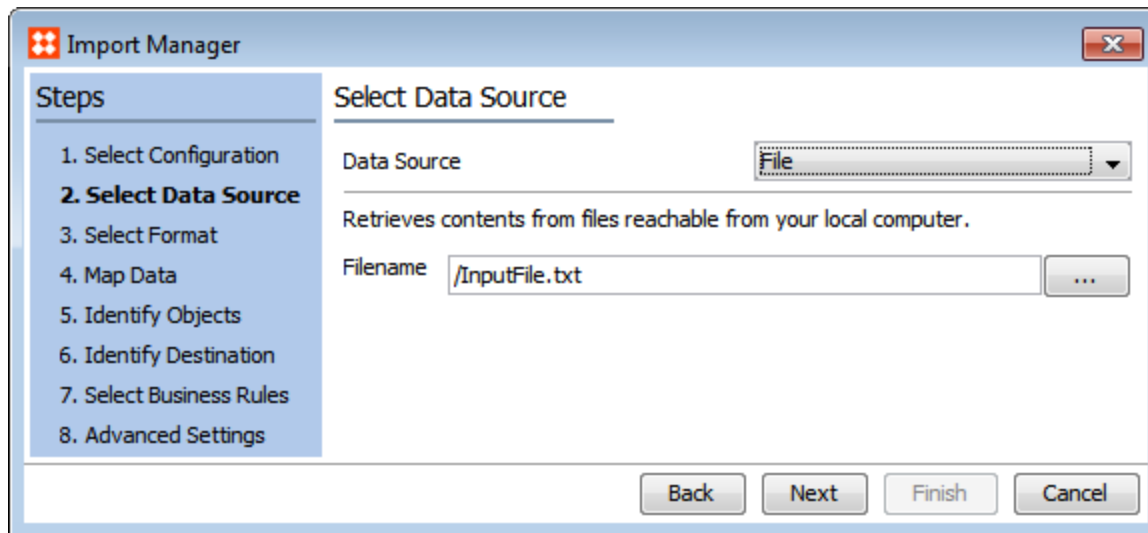
Back Next Finish Cancel

When importing via the ODBC connection, STEP converts the data into XML automatically, and loads data into STEP without any modifications.

Note: User should have the knowledge to establish an ODBC connection to the appropriate database file, and will need to know some SQL

1. Select **ODBC** from the Data Source list.
2. In the **Data Source** field, specify the target database.
3. Enter the database **User** ID and **Password** (if required).
4. In the **Query** field, write a SQL query that specifies which rows you want to import, 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.

When importing UNSPSC format, the file type required differs based on the version being imported. For more information, see the file preparation section of the **UNSPSC Format** documentation.

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**.

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**

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

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.

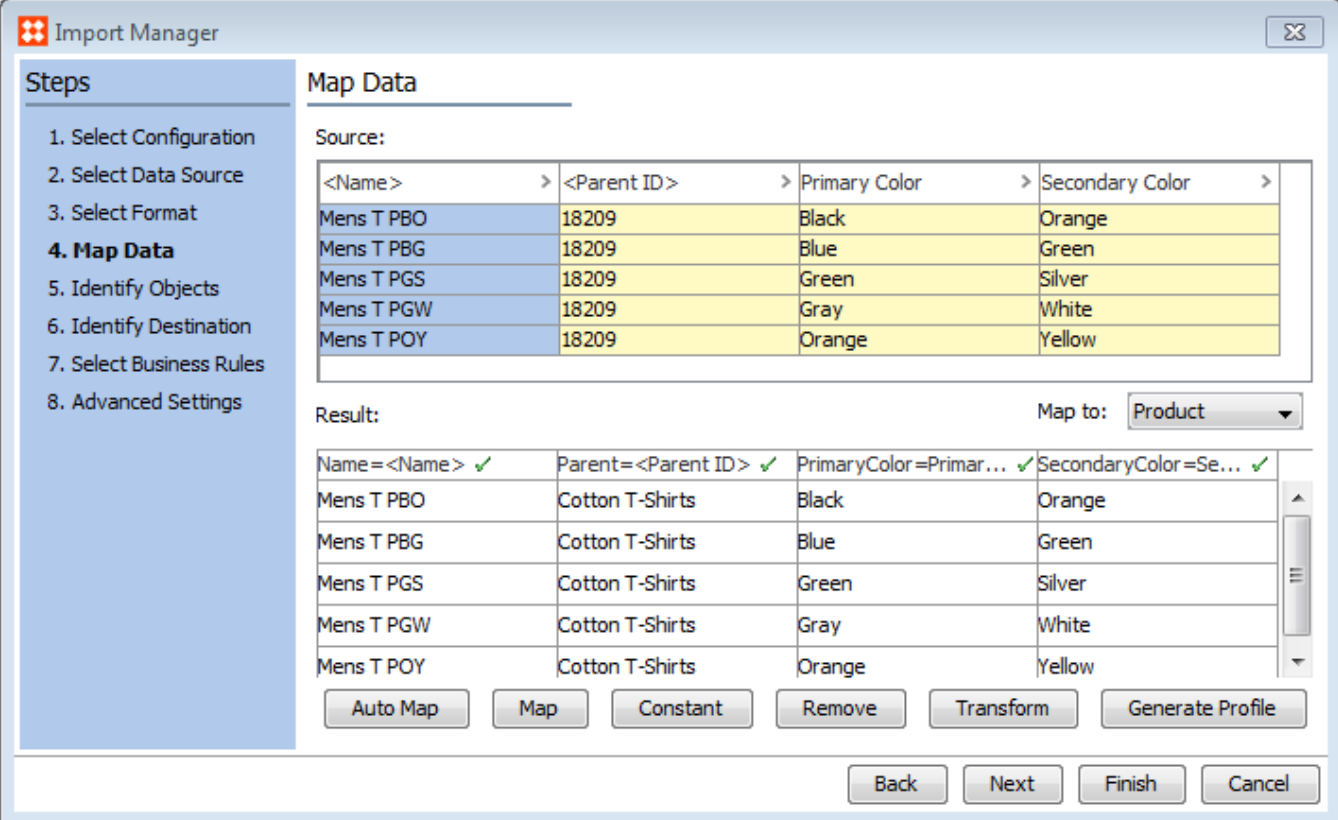
The following formats are available:

- 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

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 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 oype. 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.

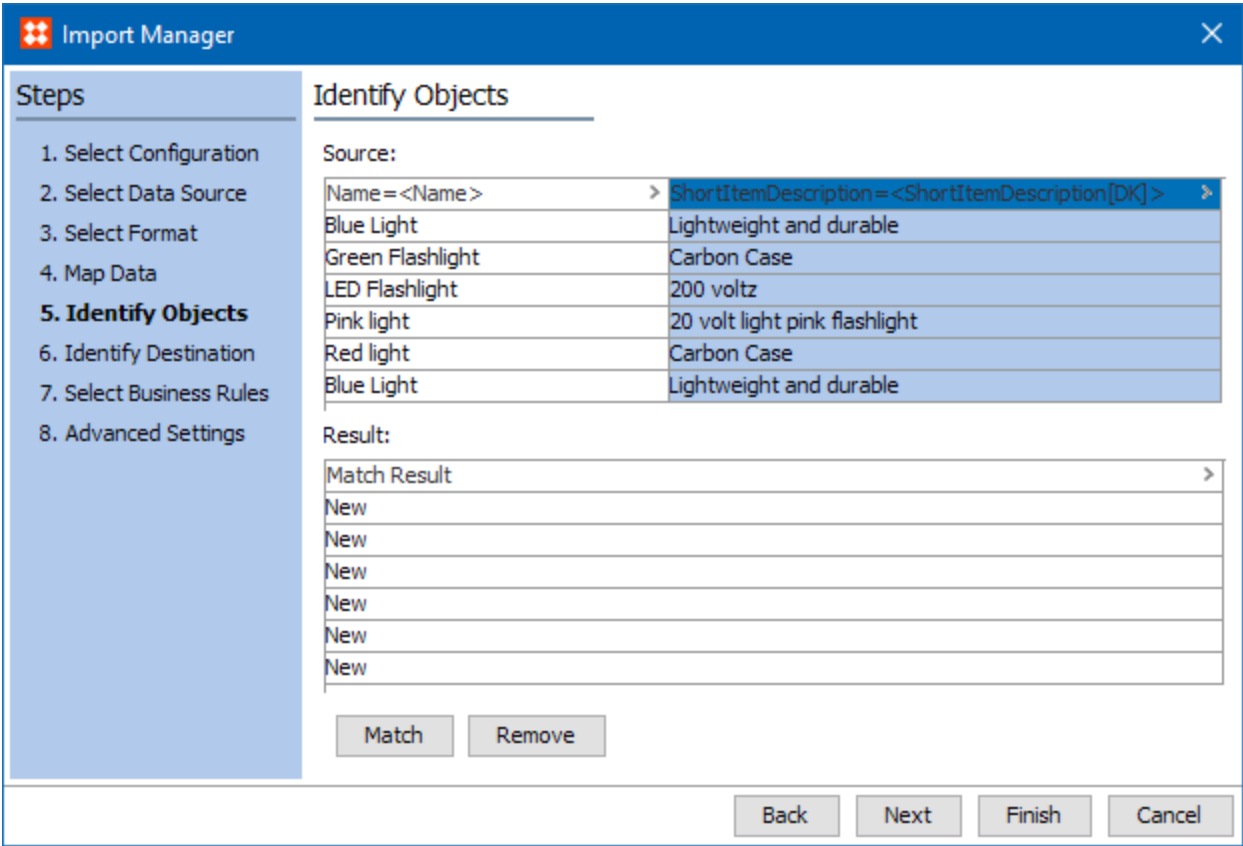
The following example describes how to identify products by attribute values. The product data is loaded from a file with no product IDs listed but contains the manufacturer's own part numbers.

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:

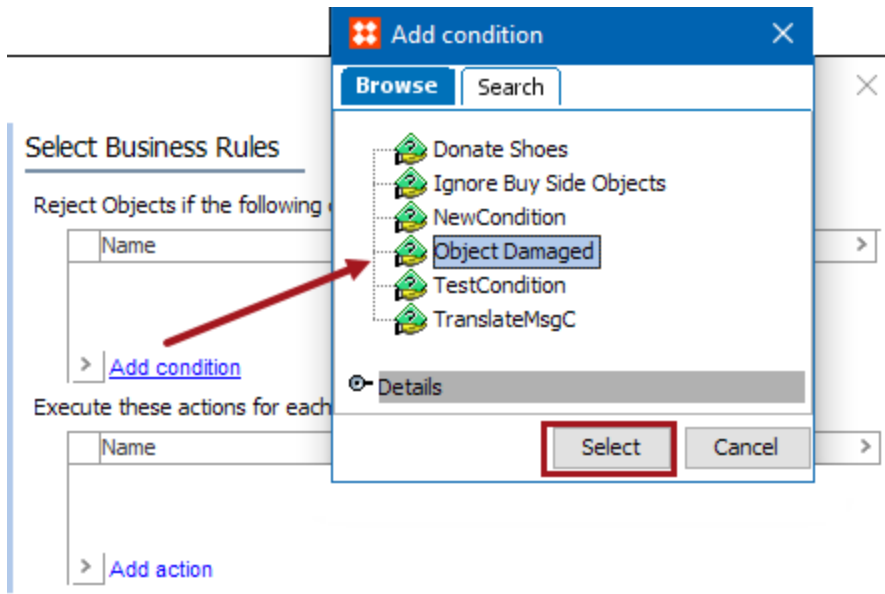
Name	Description
Object For Review	

> [Add action](#)

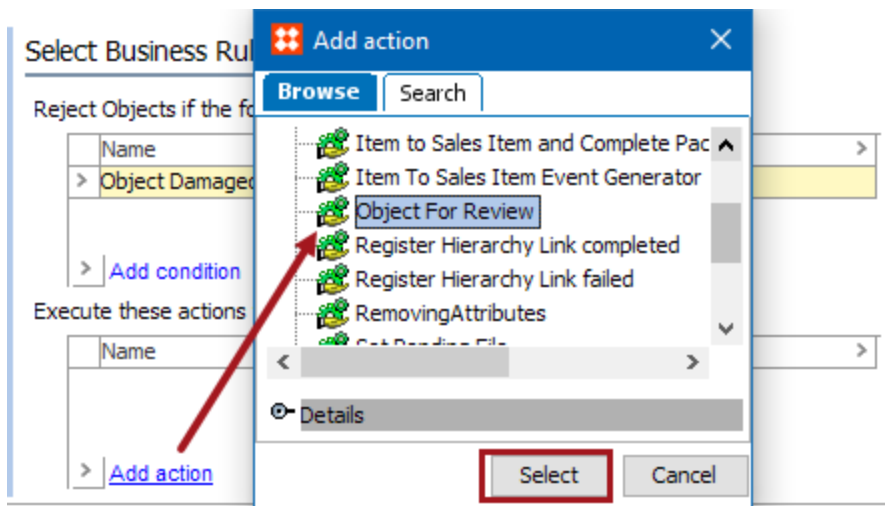
Back Next Finish Cancel

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.

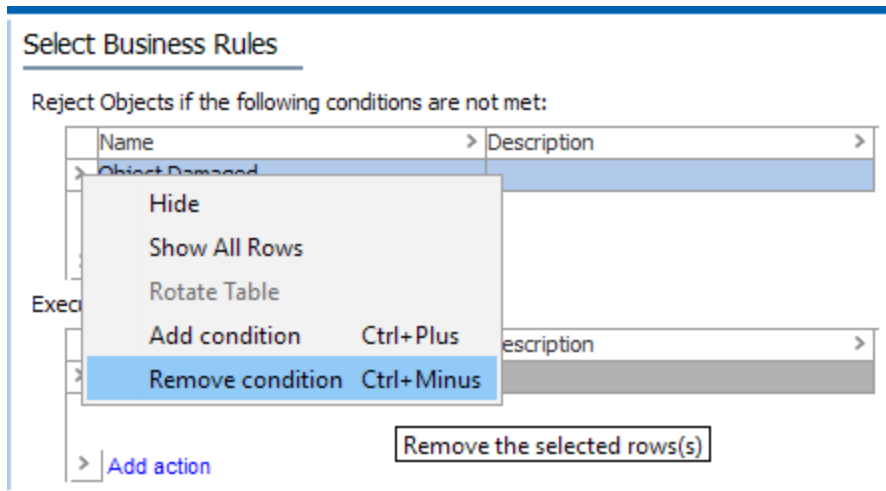
1. If necessary, click the **Add Condition** link to add conditions. Then use Browse or Search to locate the condition, and click **Select**. Repeat until all the relevant conditions are added.



2. 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.



3. If necessary, to remove a **condition**, select it, right-click, and then select **Remove condition**.



4. If necessary, to remove an **action**, select it, right-click, and then select **Remove action**.
5. 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 here 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. You use this option to indicate to the system on how to match the units on in the data file – the unit's ID or its name. The default is name.
- **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 (column based 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 COntextID="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

preconfigured 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** - These radio buttons are displayed and enabled when the user running the import has the super user privileges. For more information, see the **Users and Groups** topic in the **System Setup / Super User Guide** documentation. 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.

When displayed, the following options are available:

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 prior to starting the import. 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 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 Un-Mapped Values Setting

This option allows you to select one or more attribute groups. It enables you to remove attribute values from the selected attribute groups and replace them with the values specified in the load file, leave them blank, or return them to their inherited value.

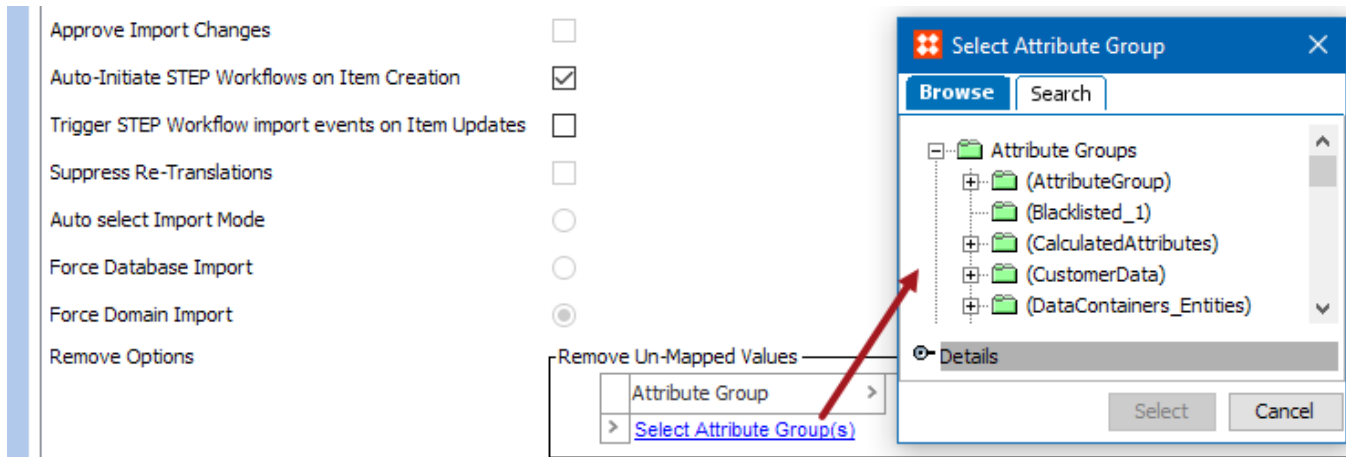
Remove Un-Mapped Values

Attribute Group

>

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

Click the 'Select Attribute Group(s)', and it will open the window to select the Attribute groups available in the STEP



Important: Use this option with care as it can both add and remove data from STEP.

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 References Settings

Remove Un-Mapped References

☐ Classification References

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

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

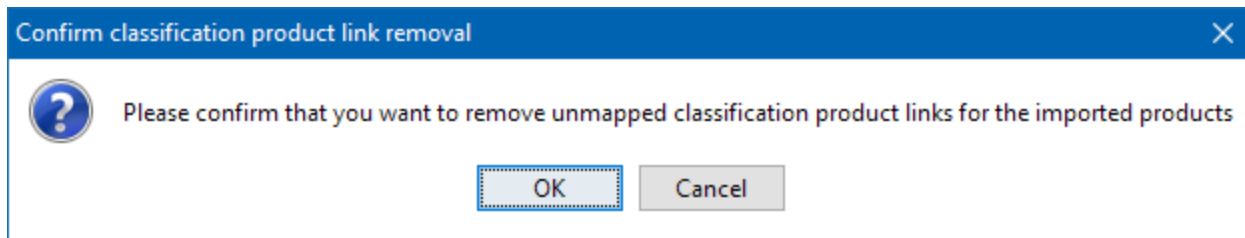
Entity Reference Type >
> [Add Entity Reference Type](#)

Classification Product Link Type >
> [Add Classification Product Link Type](#)

Selecting one of these options allows a user to specify which un-mapped reference type they would like to remove. Specific information for each reference type is explained below in greater detail.

Important: Use these options with care as they can both add and remove data from STEP.

- **Classification References** - Check to remove all product to classification links on the products being imported and only create mapped product to classification links. A dialog, as shown below, allows the user to confirm that you want to remove all unmapped classification references for the imported object.



- **Product Reference Type** - This option enables you to clean up your 'Product to Product References', 'Classification to Products References', or 'Entity to Product References'.

For example, if you have products in your system 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 will be linked to one or more target products). All links 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'.

You can only unlink the reference types that you have specified. To unlink all references from a product, no matter what type, you must add 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. Common setup is to deselect the Remove Un-Mapped References option the second time.

- **Asset Reference Type** - This is similar to the product reference type described earlier, except that it refers to 'Product to Asset References', 'Classification to Asset References', or 'Entity to Asset References'. It enables you to clean up, for example Product to Asset References by removing some or all references between products and assets and reinstating 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 Vendor Logo. 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 from the product /

image. The system removes the selected reference type(s) and reinstates only the reference type(s) listed in the load file.

- **Add Classification Product Link Type** - Click and select the specific 'Product to Classification Link Types' to be removed. A dialog allows you to confirm that you want to remove all unmapped classification product links for the imported products.

Only use this option when you have a load file where you have mapped one column to the product ID and another column to a classification ID. That is, you have products that you want to link to a classification.

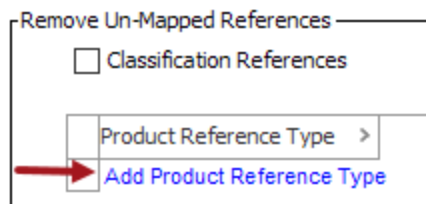
For example, if you have products that are linked multiple times to different classifications, you can use this option to clean up product to classification links not specified by the Import file. You therefore define certain products that are linked only to the classifications you list in your load file and not to any other classification.

Do that by listing the products and classifications in the load file, mapping them, and then selecting one of the two of the following options to remove remaining product to classification links.

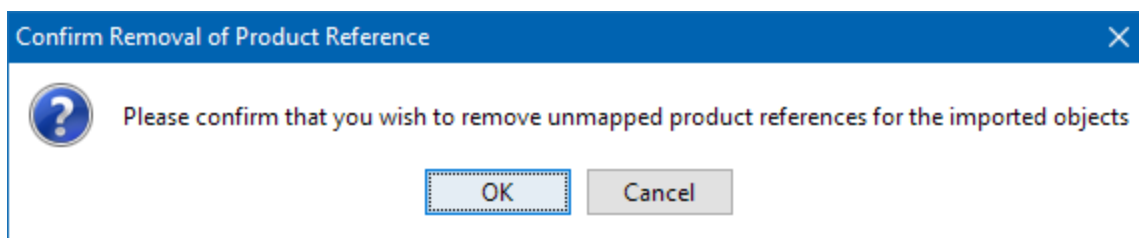
Adding and Removing Un-Mapped References

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'.

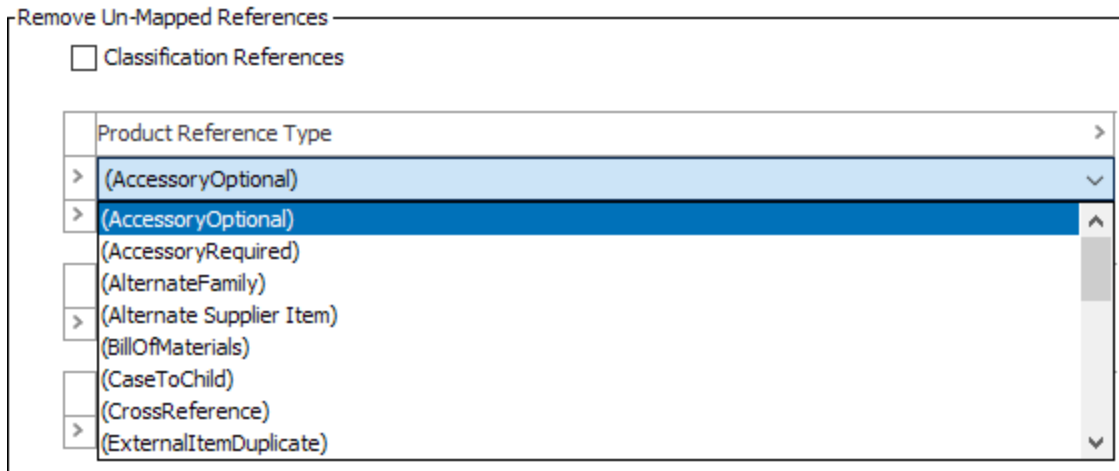
1. Click the option 'Add Product Reference Type'.



2. A dialog allows the user to confirm that they want to remove all unmapped Product references for the imported object.

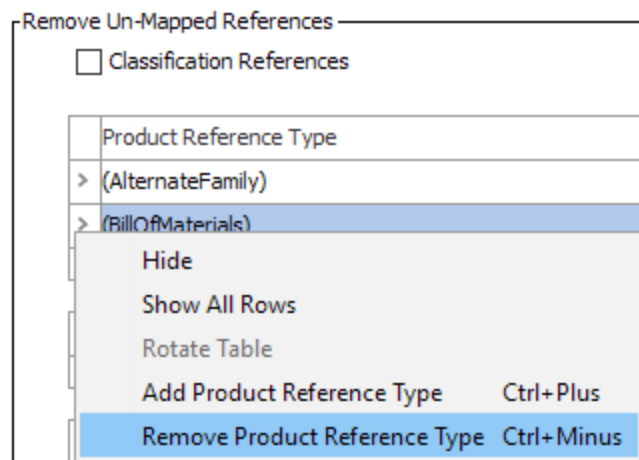


3. Click OK, and then select the appropriate reference type from the dropdown list.



4. Right-click on the field of the Reference type will give the following options

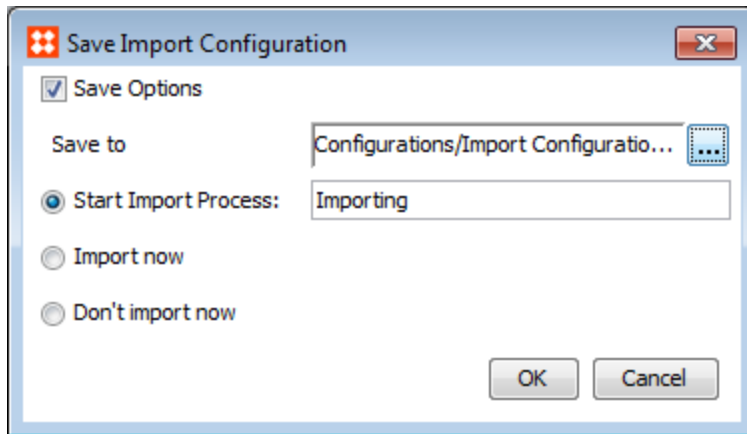
- **Hide** - to hide the selected reference type from the dropdown list
- **Show all Rows** - To show all the selected reference types
- **Add Product Reference Type** - To add one more product reference
- **Remove Product Reference Type** - To remove the selected reference type



Running a Data Import

After clicking the Finish button on the Import Manager wizard, the **Save Import Configuration** window displays and allows you to save the configuration and then start the import.

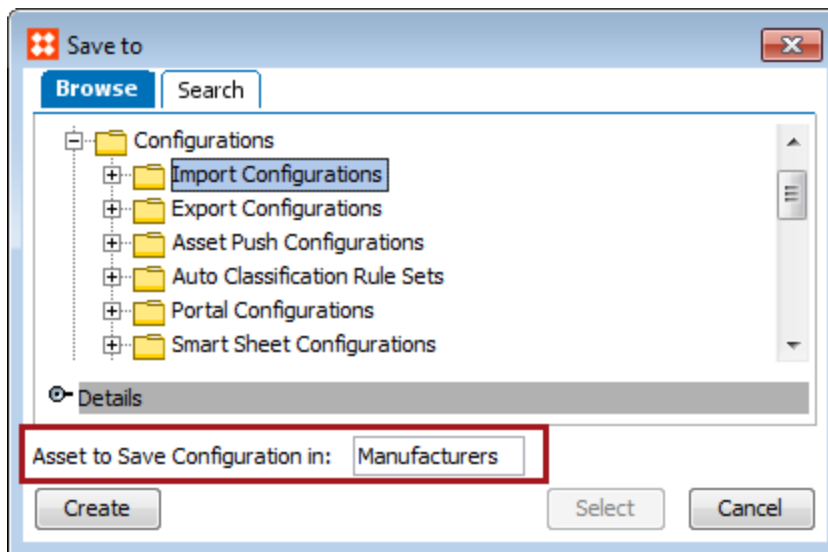
Saving a configuration allows you to modify it (instead of recreating it) during testing. Initially, use a small group of objects to test your settings. Refine the wizard parameters to resolve any issues and save the updated configuration.



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 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.

The screenshot displays the Stibo Systems interface. On the left, a tree view shows the hierarchy of configurations, with 'Import Configurations' expanded and 'ExcelImportProduct' selected. The right pane shows the details for 'ExcelImportProduct rev.4.0 - Images & Documents'. The 'Goto' search bar at the top right is set to 'ExcelImport'.

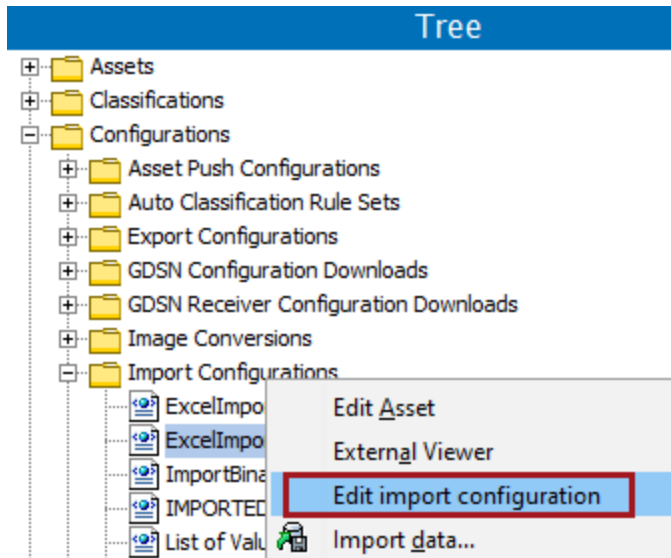
Name	Value
ID	180525
Name	ExcelImportProduct
Object Type	Import Manager Configuration
Revision	4.0 Last edited by USERJ on Mon Jun 27 16:55:49 ED...
Approved	Never Been Approved
Translation	Not Translated
Path	Classification 1 root/Configurations/Import Configur...
Content In	Language=Language Root, Country=Country Root
Asset Keywords	abc
Asset Object Type	Import Manager Configuration
Calculated Asset File Name	180525-ExcelImportProduct

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.

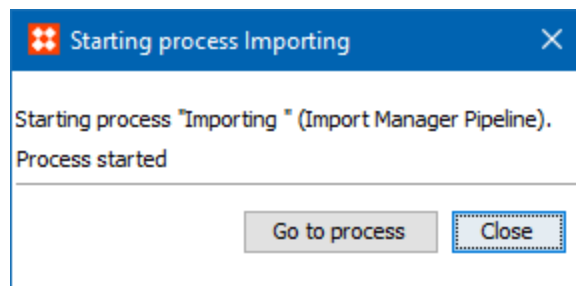
- To change only the name of the configuration, simply edit the Name field in the configuration editor.
 - To edit an existing configuration and save with a new name, see the steps in the **Running a Data Import** section.
 - To change the settings of the existing configuration parameters, follow the 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.

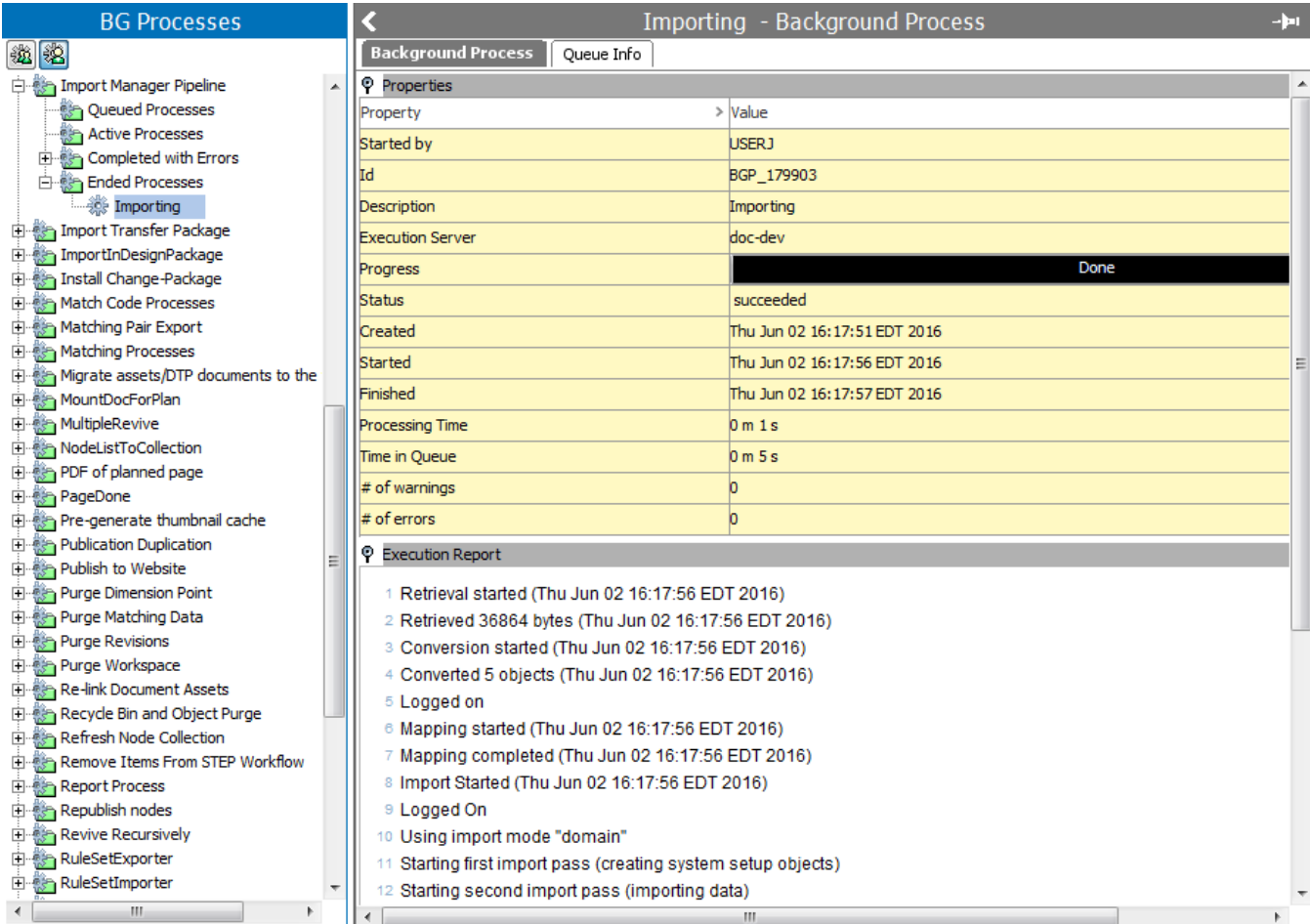
Monitoring the Import Process

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.



The screenshot displays the Stibo Systems interface. On the left, the 'BG Processes' tree is visible, with the 'Importing' process selected. On the right, the 'Properties' flipper is open, showing details for the selected background process.

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

Below the properties table, the 'Execution Report' is displayed, showing a list of steps:

- 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

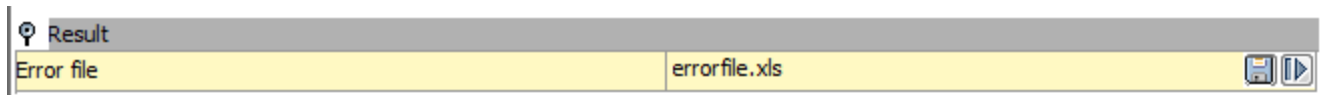
Some imports result in errors, for example, importing a new object (product) that is linked to another new object (classification or asset) that is also being imported. The background process fails, but the user can use the execution report to identify the error and resolve it by referring to the list 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.

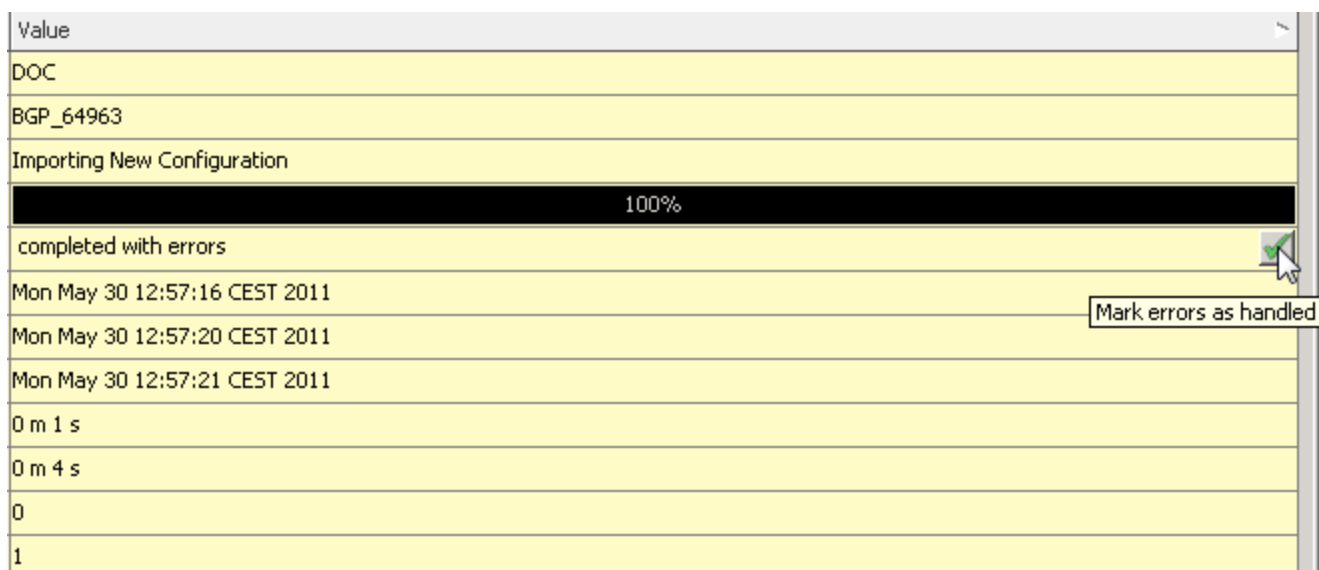
For Excel and CSV imports, the error file contains the rows from the original file with the defect data.

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.
Unknown unit	The unit encountered in the load file does not exist in

Error Message	Description
	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 imported.	You cannot make a product-to-asset reference from the same source product to the same target asset more than once.

Error Message	Description
*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.
No WorkspaceID specified in STEPXML to import	Importing STEPXML without a Workspace ID.

Error Message	Description
	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 "1/2" STRAIGHT WIDGET SSU05PL" for attribute Description in product T100-3526 (UnknownUnit).
Source: Line=2, Column=Column2.
- Illegal unit "1/4" ANGLED GADGET BZT10GL" for attribute Description in product T100-2625 (Unknown Unit).
Source: Line=3, Column=Column2.
- Illegal unit "3/4" 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_BE_F_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BE_F_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

Here 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_BEF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEF_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_BEF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEF_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_BEF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEF_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.

Here 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:

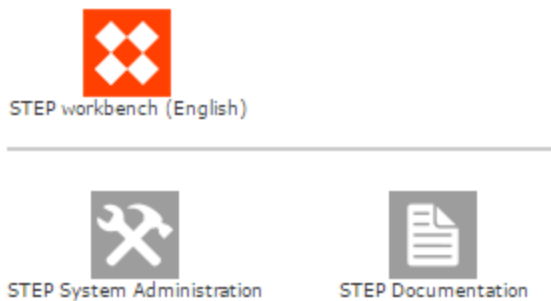
Description	Importing
Execution Server	doc-rel
Progress	100%
Status	completed with errors
Created	Sun Feb 05 06:49:19 EST 2017
Started	Sun Feb 05 06:49:20 EST 2017
Finished	NA
Processing Time	0 m 2 s
Time in Queue	0 m 1 s
# of warnings	0
# of errors	1

 Execution Report

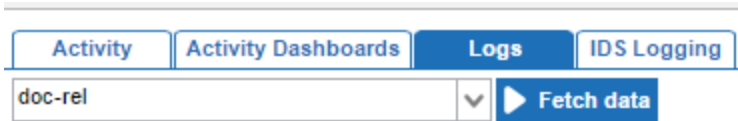
- 1 Retrieval started (Sun Feb 05 06:49:21 EST 2017)
- 2 Retrieved 8817 bytes (Sun Feb 05 06:49:21 EST 2017)
- 3 Conversion started (Sun Feb 05 06:49:21 EST 2017)
- 4 Converted 6 objects (Sun Feb 05 06:49:24 EST 2017)
- 5 Logged on
- 6 Mapping started (Sun Feb 05 06:49:24 EST 2017)
- 7 Mapping completed (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:

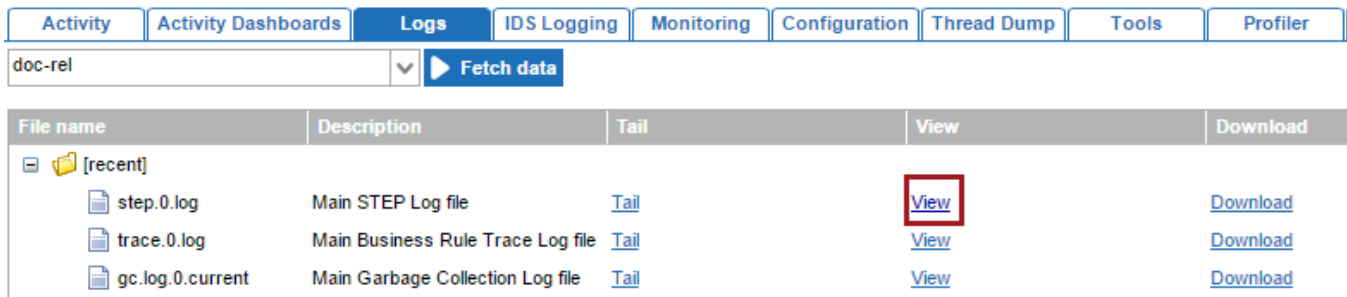
1. Launch the STEP Web start page
2. Click on the System Administration link



- 3. Once logged in, click on the Logs Tab
- 4. Select the Server in the drop down 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.getConfigurationsPackagesLogFileNames(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 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 pre-processors 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 post-processors 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:

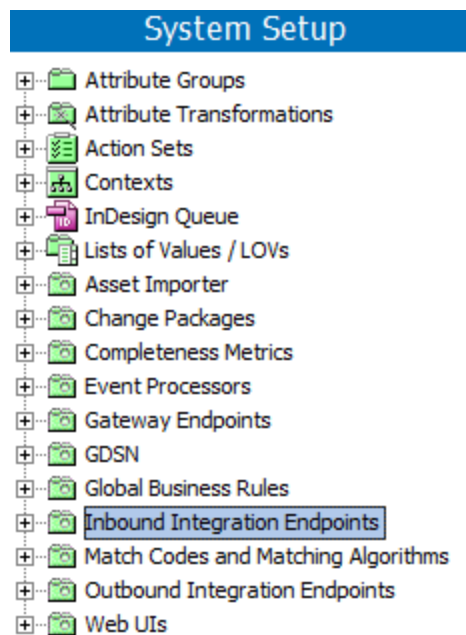
1. Understand each element of an IIEP as described in **Inbound Integration Endpoint Structure**.
2. Understand the effects of the transactional setting on an IIEP as described in **Inbound Integration Endpoint Transactional Settings**.
3. Maintain or modify the endpoint as described in **Maintaining 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, one or more setup groups must be created that are allowed to hold inbound integration endpoints. You must also specify the setup group(s) in which inbound 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 inbound integration endpoints. For detailed information, see the **Action Sets** section and the **Users and Groups** section in the **System Setup / STEP Super User** documentation.

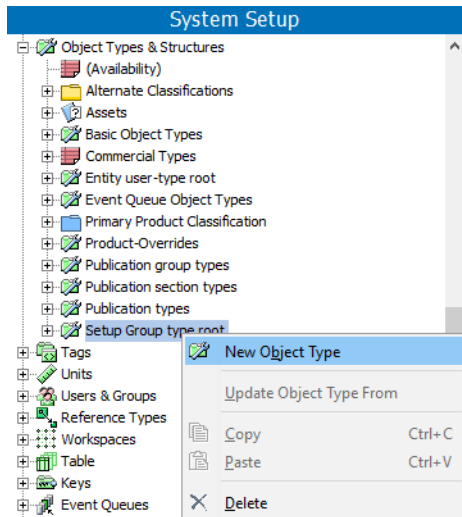
Review your System Setup tab to determine if one or more inbound integration endpoint nodes already exist. The name of the node on your system is not required to match the one in the image below.



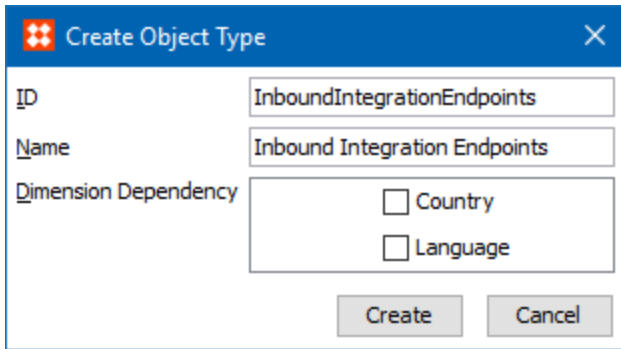
Once the setup has been completed, the steps in this section are only needed if additional levels of organization are desired.

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**.

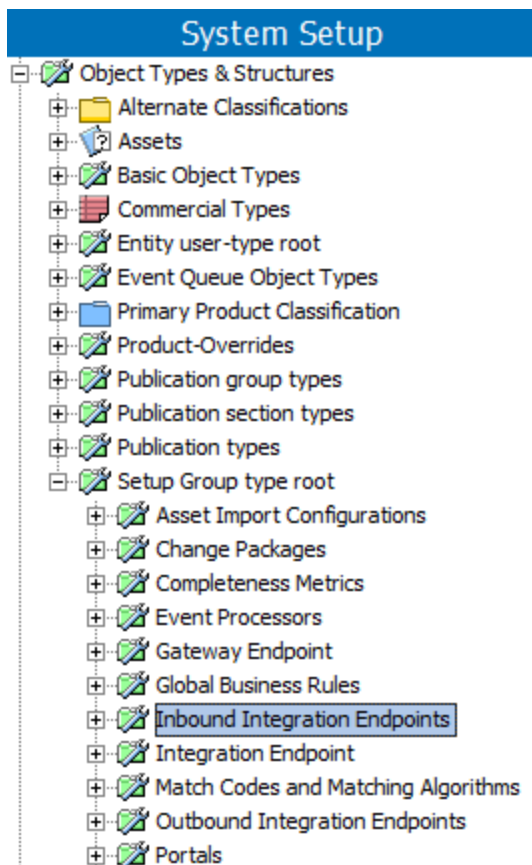


3. Enter an **ID** and a **Name**, select any required Dimension Dependencies, and click **Create**.



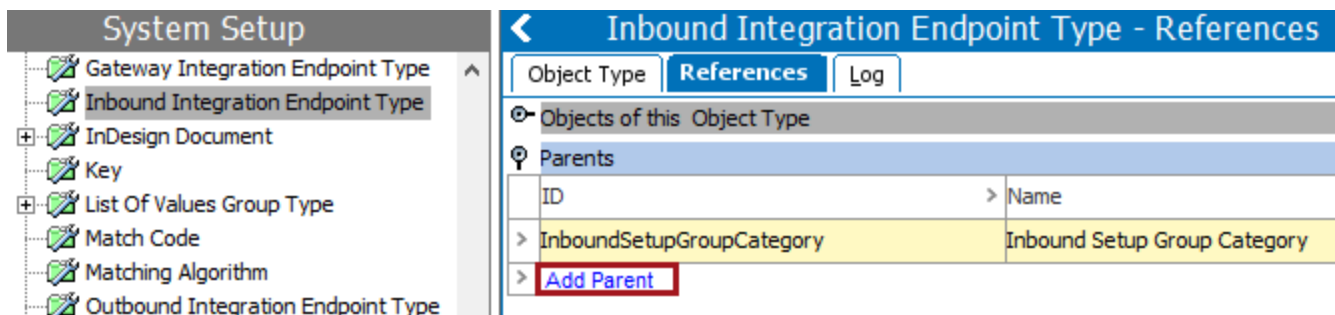
The screenshot shows the 'Create Object Type' dialog box. It has a blue title bar with the Stibo logo and a close button. The dialog contains three input fields: 'ID' with the value 'InboundIntegrationEndpoints', 'Name' with the value 'Inbound Integration Endpoints', and 'Dimension Dependency' which is a container for two checkboxes: 'Country' and 'Language', both of which are unchecked. At the bottom of the dialog are two buttons: 'Create' and 'Cancel'.

4. The new setup group appears in System Setup under 'Object Types & Structures' as a child in the **Setup Group type root**.

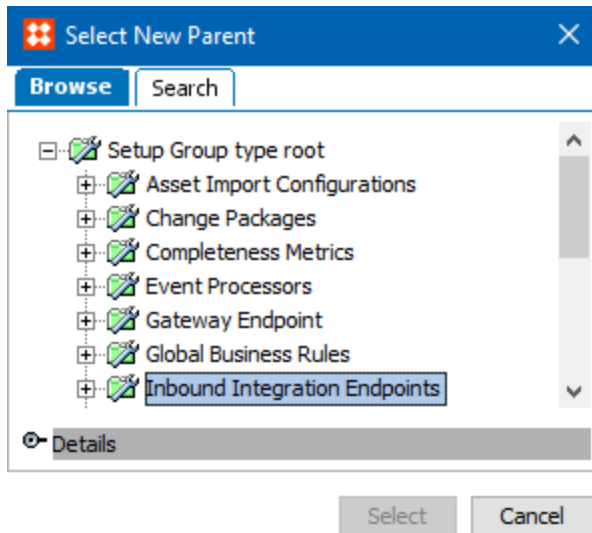


Link the IIEP Object Type to Setup Group

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.

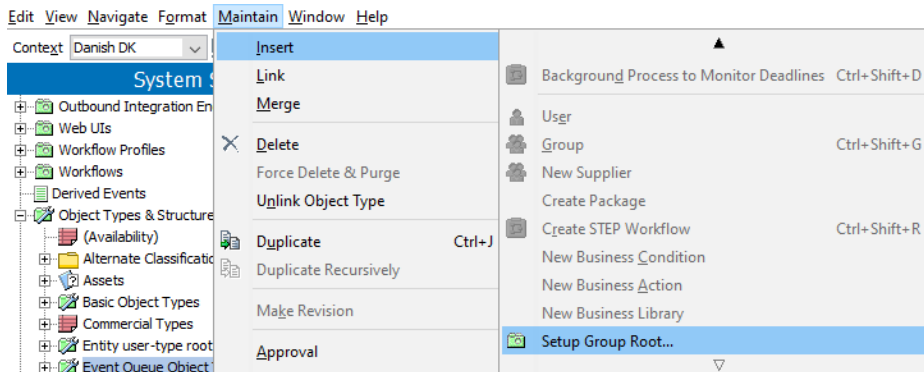


3. In the Select New Parent dialog, select the setup group you created, and click **Select** to make it a valid parent.

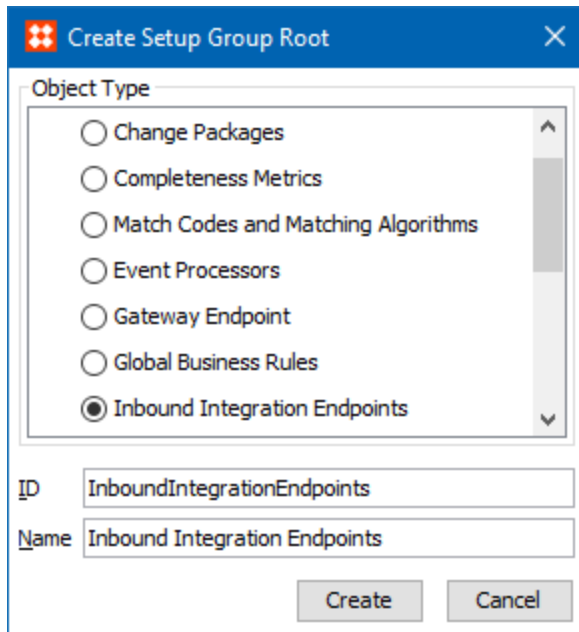


Create an Instance of the IIEP Object

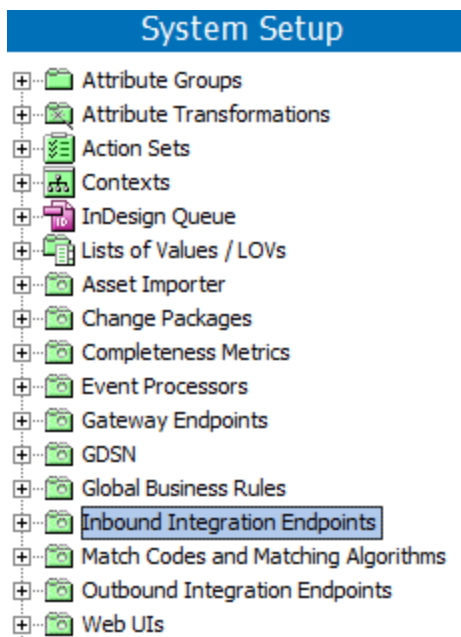
1. On the System Setup tab, select any object in the **System Setup** hierarchy to activate the following Maintain menu selection.
2. Click the Maintain menu, point to Insert, and select **Setup Group Root**.



3. In the Create Setup Group Root dialog, select the inbound integration endpoints object type, enter an **ID** and a **Name**, and click **Create**.



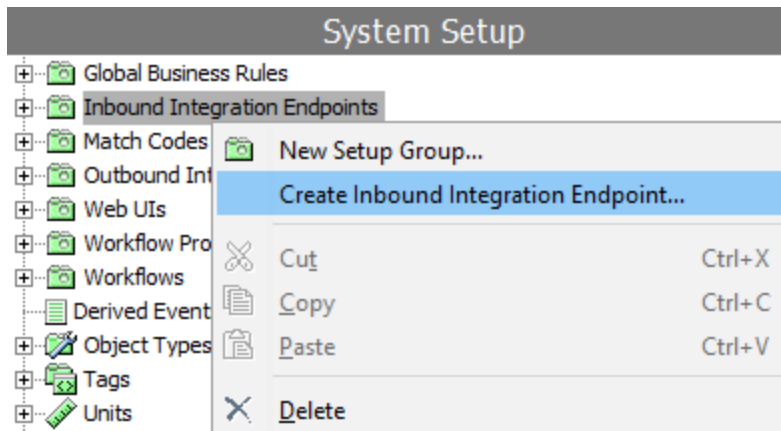
A setup group is created as a node in the System Setup hierarchy. Inbound integration endpoints can now be created under this new node.



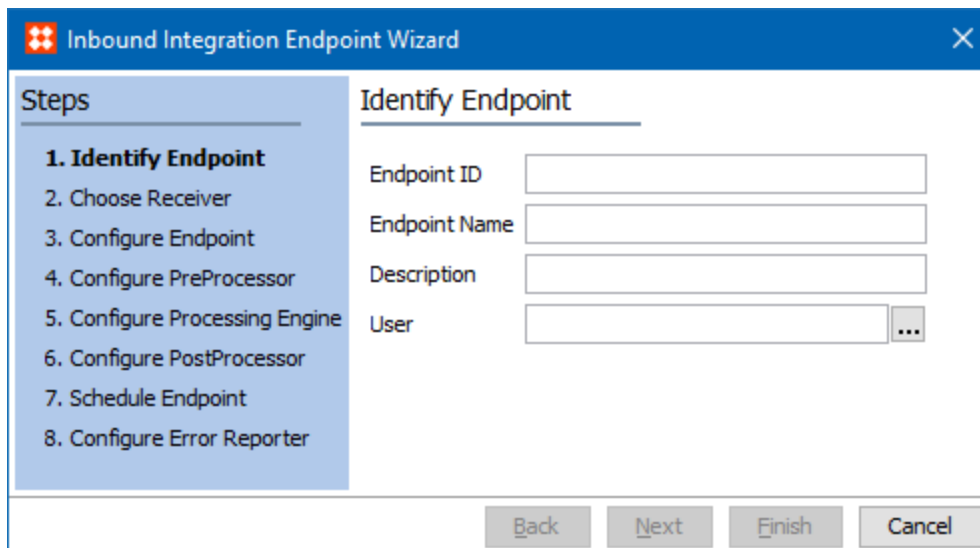
Creating an Inbound Integration Endpoint

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**.



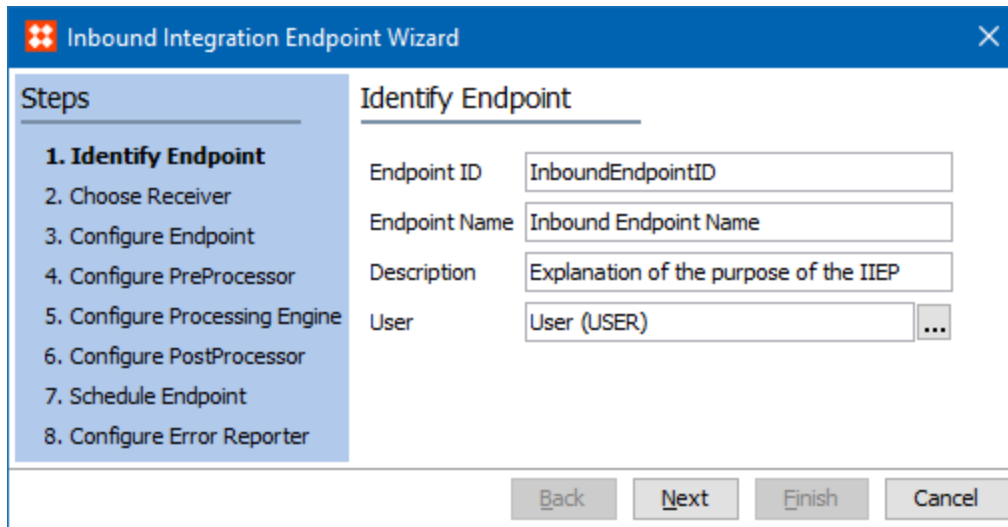
2. The Inbound Integration Endpoint wizard displays and can involve the following steps:



- **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 pre-processors 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



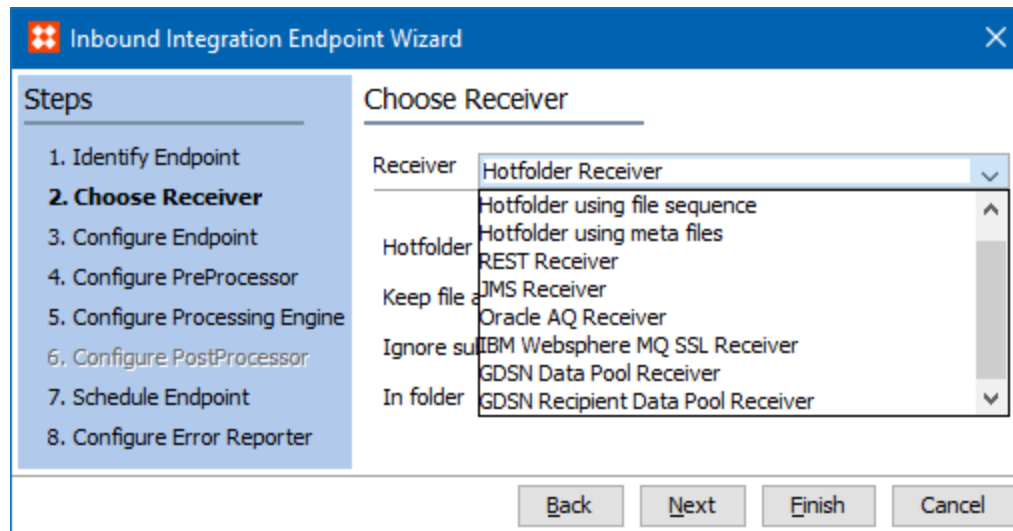
The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' list contains eight items: 1. Identify Endpoint (highlighted), 2. Choose Receiver, 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, 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 the value 'Inbound Endpoint Name', 'Description' with the value 'Explanation of the purpose of the IIEP', and 'User' with the value 'User (USER)' and a browse button (...). At the bottom are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

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**.
4. Search or browse for a **User**. The privileges of the selected user determine which actions the 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
- 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.

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 About 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.
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.

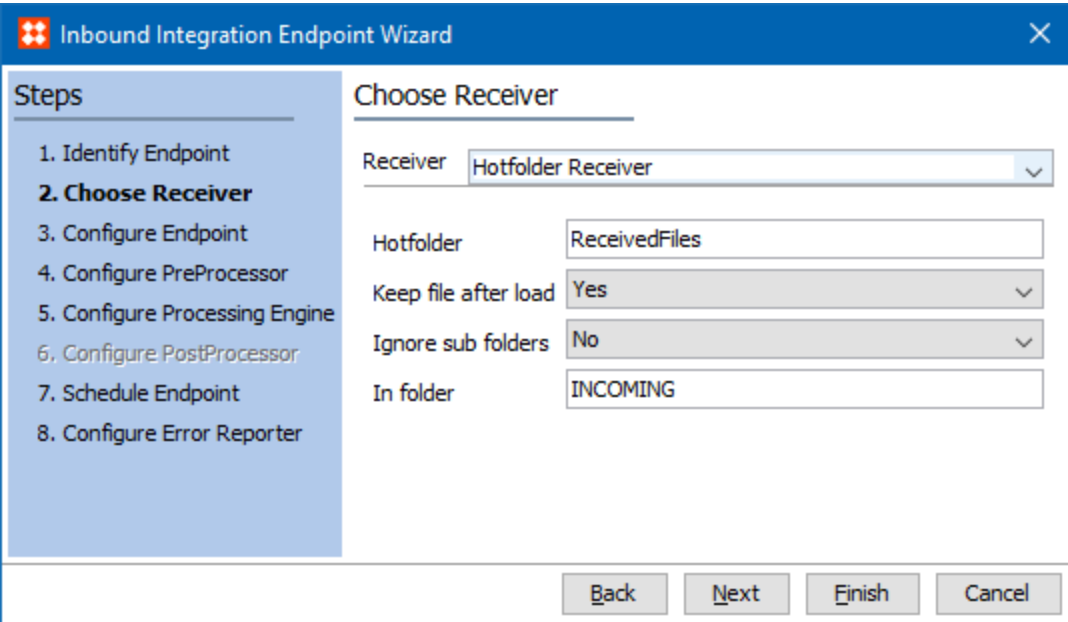
To control the order of the files to be imported, set up either a **Hotfolder Receiver Using Meta Files** or a **Hotfolder Receiver Using File Sequence** instead.

Configuration

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.

Parameters

The following parameters are available for this receiver:



The screenshot shows the 'Inbound Integration Endpoint Wizard' window, specifically the 'Choose Receiver' step. 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 four configuration fields: 'Receiver' (a dropdown menu set to 'Hotfolder Receiver'), 'Hotfolder' (a text box containing 'ReceivedFiles'), 'Keep file after load' (a dropdown menu set to 'Yes'), and 'Ignore sub folders' (a dropdown menu set to 'No'). Below these fields is an 'In folder' text box containing 'INCOMING'. At the bottom of the window are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

1. **Hotfolder** - The name entered is used to create a folder on the application server in the 'upload' directory.
2. **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.
3. **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.

4. **In folder** - Optional. Enter a name for an 'In' folder. The name entered here 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.

Hotfolder Receiver Using File Sequence

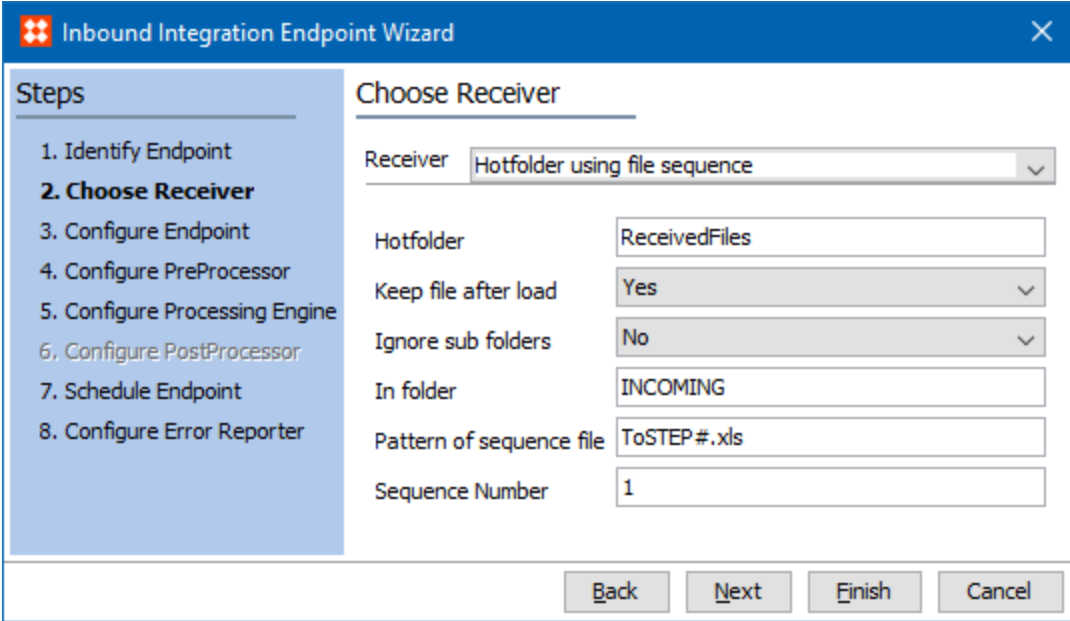
The 'Hotfolder using File Sequence' receiver enables setup of a standard data hotfolder where a sequence ID (included in the file name) determines the file processing order. Files are placed into this folder and are processed by the integration endpoint. Typically this is on the application server or can be reached from the application server. All formats supported by the standard import functionality are available via hotfolder import.

Configuration

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.

Parameters

The following parameters are available for this receiver:



The screenshot shows the 'Inbound Integration Endpoint Wizard' window, specifically the 'Choose Receiver' step. 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 'Hotfolder using file sequence'. Below this are several input fields: 'Hotfolder' with the value 'ReceivedFiles', 'Keep file after load' with a dropdown set to 'Yes', 'Ignore sub folders' with a dropdown set to 'No', 'In folder' with the value 'INCOMING', 'Pattern of sequence file' with the value 'ToSTEP#.xls', and 'Sequence Number' with the value '1'. At the bottom of the window are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

1. **Hotfolder** - The name entered is used to create a folder on the application server in the 'upload' directory.
2. **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.
3. **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.

4. **In folder** - Optional. Enter a name for an 'In' folder. The name entered here 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.
5. **Pattern of Sequence File** - 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.
6. **Sequence Number** - 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'.

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.

Configuration

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.

Parameters

The following parameters are available for this receiver:

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. **Hotfolder** - The name entered is used to create a folder on the application server in the 'upload' directory.
2. **Pattern of meta file** - 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. **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. **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. **In folder** - Optional. Enter a name for an 'In' folder. The name entered here 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.

IBM Websphere MQ SSL Receiver

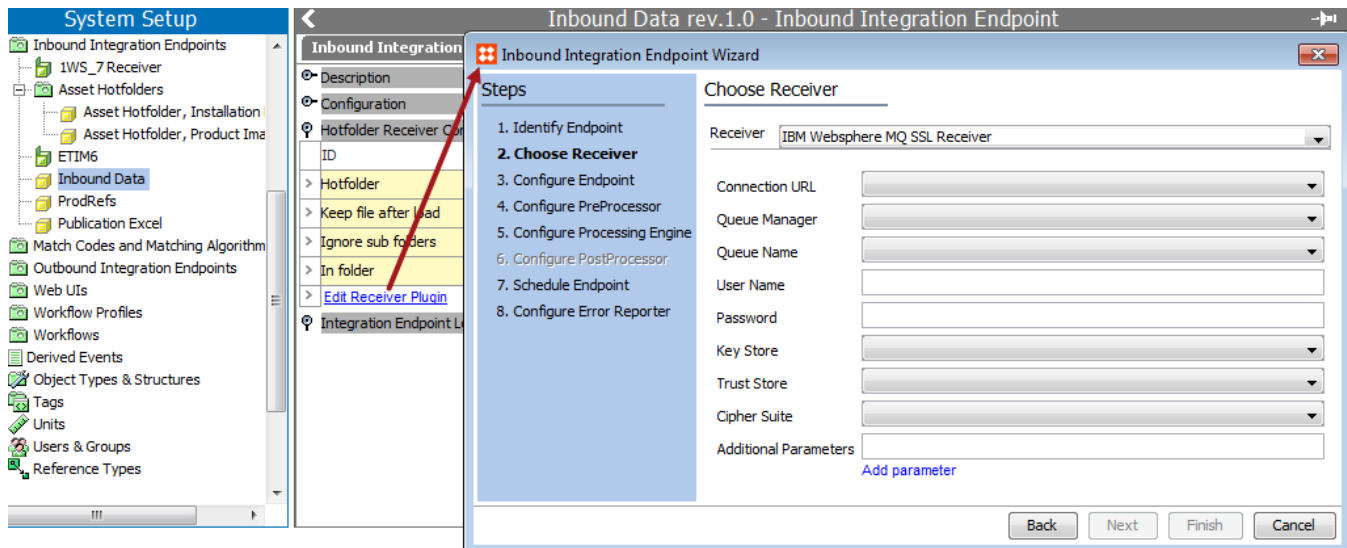
For information on connecting to IBM Websphere MQ in a non-SSL way, see **JMS Receiver**.

Configuration

The location of the folders is determined by the sharedconfig.properties file. Select a dropdown to display the required key name for each field. Changes to the properties file are implemented when the server is restarted.

Parameters

The following parameters are available for this receiver:



1. **Connection URL** - Select the URL for connection 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. **Queue Manager** - Select the name of the Queue Manager.
3. **Queue Name** - Select the name of the Queue for the connection.
4. **User Name** - If required, enter the user name that will be used to log onto the JMS provider.
5. **Password** - If required, enter the password that will be used to log onto the JMS provider.
6. **Key Store** - Select the key store in 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 or config.properties using 'WSMQSSLKeyStorePassword'.
7. **Trust Store** - Select the trust store with the CA for the Queue Manager. 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 use 'WSMQSSLTrustStorePassword'.
8. **Cipher Suite** - Set to the same value as SSL CipherSuite in Websphere MQ. STEP is running on non-IBM jre, so this must be the same value as configured in the Queue Manager.

9. **Additional Parameters** - If required, click Add parameter, then enter the Key and Value. For possible keys and values, consult the manual for IBM Websphere MQ.

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).

If additional JME support is required, contact your Stiibo Systems account manager for customizations.

Configuration

The location of the folders is determined by the sharedconfig.properties file. Select a dropdown to display the required key name for each field. Changes to the properties file are implemented when the server is restarted.

Parameters

The following parameters are available for this receiver:

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'. It features a 'Receiver' dropdown menu set to 'JMS Receiver'. Below this are several configuration fields: 'JMS Connection Factory Class' (set to 'ActiveMQInitialContextFactory'), 'Connection Factory Name', 'JMS Provider URL', 'JMS Queue', 'User Name', and 'Password'. Each of these fields has a dropdown arrow. At the bottom of the configuration area is an 'Additional Parameters' label and a blue 'Add parameter' link. At the very bottom of the window are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

1. **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. **Connection Factory Name** - Select a connection factory name in the list.
3. **JMS Provider URL** - Select a JMS URL in the list.
4. **JMS Queue** - Select the JMS Queue to be used on Apache ActiveMQ or Websphere MQ.
5. **User Name** and **Password** - If required to log onto the JMS provider, enter a user name and a password.
6. **Additional Parameters** - If required, click Add parameter, then enter the Key and Value. For possible keys and values, consult the manual for IBM Websphere MQ.

Configuration Examples

The following images demonstrate the parameters and setup for a variety of JMS Receivers.

FileInitialContextFactory JMS Receiver

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	websphermq-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

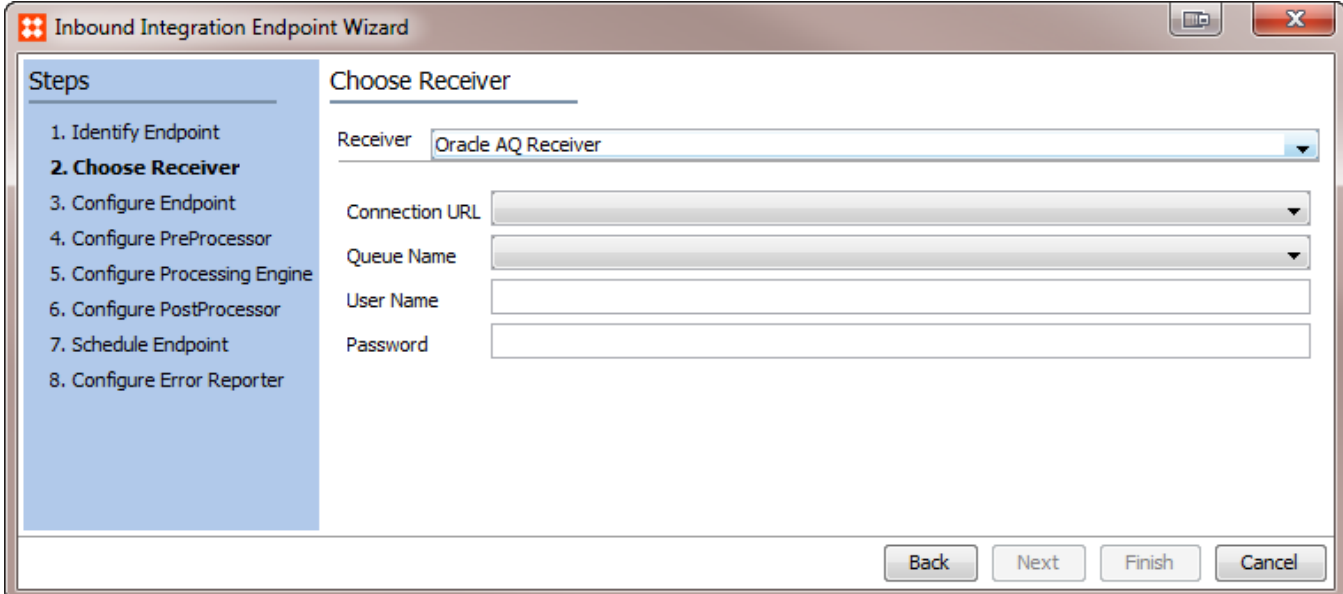
Oracle Advanced Queuing (Oracle AQ) enables messages to be exchanged between two systems.

Configuration

The location of the folders is determined by the sharedconfig.properties file. Select a dropdown to display the required key name for each field. Changes to the properties file are implemented when the server is restarted.

Parameters

The following parameters are available for this receiver:



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:

Connection URL:

Queue Name:

User Name:

Password:

1. **Connection URL** - Select a URL pointing to Oracle AQ.
2. **Queue name** - Select an Oracle AQ queue name.
3. **User Name** - Enter a user name to be used to log on Oracle.
4. **Password** - Enter a password to be used to log on Oracle.

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, you can find multiple resources on the internet.

For information about how to use the STEP REST API, see the STEP API documentation.

Configuration

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 c3RlcHN5czpzZGVwc3lz'.

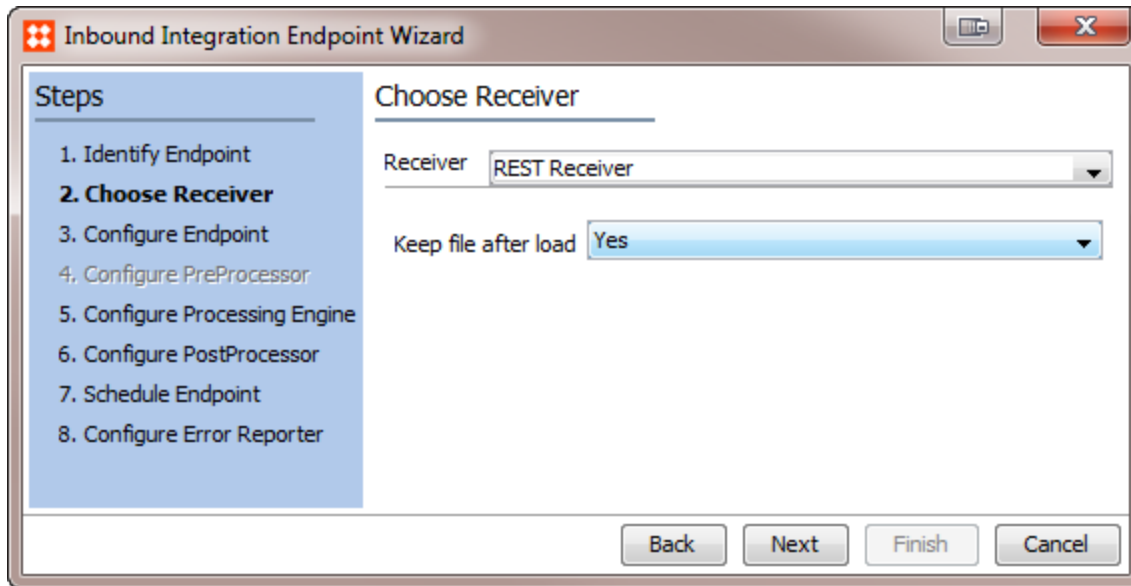
Notice that in the URL shown above, the context and workspace qualifiers have no significance, but they are required for the REST API.

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
```

Parameters

The following parameters are available for this receiver:



1. **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.

This option is available because files posted via REST are temporarily stored in a directory on the application server (under Hotfolder root).

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: **STEP Importer**

Transactional settings: **None**

Context

Workspace: **Main**

Context: **French FR**

Queue Settings

Queue for endpoint: **InboundQueue**

Queue for endpoint processes: **In**

Maximum number of old processes: **1000**

Maximum age of old processes: **1y**

Number of messages per background process: **1**

Navigation: Back, **Next**, Finish, Cancel

1. For **Processing Engine** on a standard STEP system, you can select **STEP Importer** or **Asset Importer**.
 - The 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.
 - The Asset Importer allows you to use the same import functionality as Asset Importer. Additional setup is required on the upcoming Configure Processing Engine step.
3. 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.
4. 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.
5. 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 here.
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.
 - 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.
 - 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. On the left, a 'Steps' list contains eight items: 1. Identify Endpoint, 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 'Configure PreProcessor' and features a dropdown menu labeled 'Configure PreProcessor' with 'No pre-processing' selected. At the bottom, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

1. When available, a pre-processor allows you to create extensions to STEP in order to handle your specific data processing needs. The options available depend on your particular implementation. For assistance in creating a pre-processor, contact Stibo Systems Support.
2. Click the **Next** button to display **IIEP - Configure Processing Engine**.

IIEP - Configure Processing Engine

The first screenshot shows the 'Inbound Integration Endpoint Wizard' at the 'Configure Endpoint' step. The 'Steps' list on the left highlights '3. Configure Endpoint' and '4. Configure PreProcessor'. The main area shows 'Processing Engine' set to 'STEP Importer' and 'Transactional settings' set to 'None'. A red box highlights the 'Processing Engine' dropdown, and a red arrow points from it to the second screenshot.

The second screenshot shows the 'Inbound Integration Endpoint Wizard' at the 'Configure Processing Engine : Select Sample File' step. The 'Steps' list on the left highlights '5. Configure Processing Engine' and its sub-steps: '5.1. Select Sample File', '5.2. Select Data Source', and '5.3. Select Format'. The main area is titled 'Configure Processing Engine : Select Sample File' and shows a 'Sample file' section with a text area containing XML data:


```
<STEP-ProductInformation ExportTime="2016-12-26 08:03:07">
  <Products>
    <Product ID="21926" UserTypeID="Level3" ParentID="2192
    <Name>Refrigeration</Name>
    <Values>
      <Value AttributeID="ChildCount" Derived="true">0</Valu
    </Values>
    <Product ID="21929" UserTypeID="ItemFolder">
```

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
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 text field containing 'Image Importer (Image Importer)' with an ellipsis button to its right.
- Handle content as:** A dropdown menu currently set to 'Files'.
- Override configuration options for subfolders:** A table with columns for 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 window 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. 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.

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 receiver options include:

- **Files** - This option should be used when importing single asset files into STEP.
- **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 the remaining configuration options.
- **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).

Metafile Error Handling *	Report missing files
Filename Column Header *	Filename
Column Delimiter *	,

Override configuration options for subfolders:

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.

- **ZIP Metafile** - This option should be used when importing a ZIP file that contain 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.

Name of metafile within ZIP file *	Assetfiles.csv
Metafile Error Handling *	Report missing files
Filename Column Header *	Filename
Column Delimiter *	,

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.

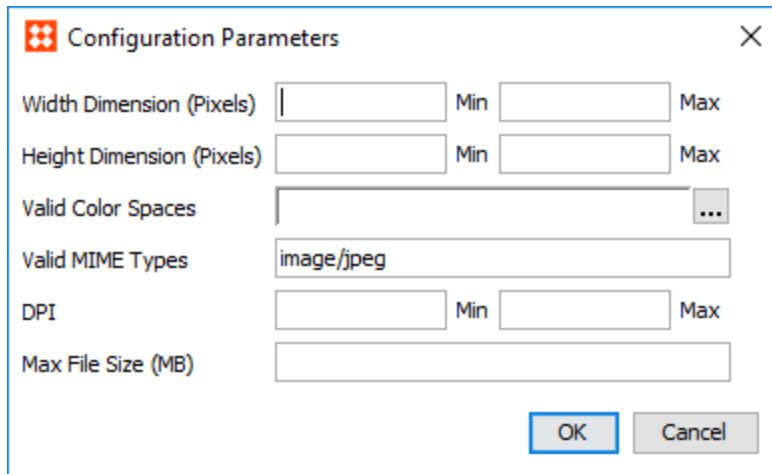
In the **Column Delimiter** parameter, specify which character should be used to separate each of the columns in the metadata file. The default value is a coma (,) due to backwards compatibility, but it is recommended to use either a semicolon or a pipe character, which is less likely to be included in any metadata fields.

3. In the **Override configuration options for subfolders** area of the wizard, click 'Add Folder Override' 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.

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 here 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 below, this use case is addressed by overriding the 'Import Validator' step of the importer configuration and changing the 'MIME Type' parameter to '/jpeg'.

Folder >	Configuration Step >	Configuration >
> JPEG	Import Validator	Override parameters
> TIFF	Import Validator	Override parameters
> Product Images	Product Linker	Override parameters
> Add Folder Override		

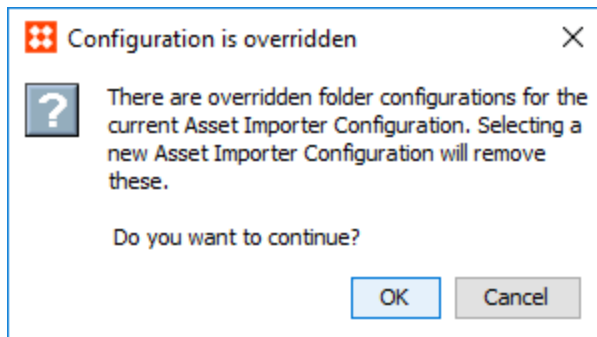


The 'Configuration Parameters' dialog box contains the following fields and controls:

- Width Dimension (Pixels):** A text input field followed by 'Min' and 'Max' labels and corresponding input fields.
- Height Dimension (Pixels):** A text input field followed by 'Min' and 'Max' labels and corresponding input fields.
- Valid Color Spaces:** A text input field with a three-dot menu button to its right.
- Valid MIME Types:** A text input field containing the value 'image/jpeg'.
- DPI:** A text input field followed by 'Min' and 'Max' labels and corresponding input fields.
- Max File Size (MB):** A text input field.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

- 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.
- In the **Configuration** field, click the **Override parameters** button and make any of the necessary changes in the screen that appears. Then click **OK**.

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.



The 'Configuration is overridden' dialog box contains the following elements:

- Icon:** A question mark icon in a square box.
- Text:** 'There are overridden folder configurations for the current Asset Importer Configuration. Selecting a new Asset Importer Configuration will remove these.'
- Text:** 'Do you want to continue?'
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

4. Click the **Next** button to display **IIEP - Schedule Endpoint**.

For more information on configuring Asset Importer, see the **Asset Importer Configuration** section of the **Asset Importer** documentation.

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 reference 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 **Step 6 - Metadata Importer** section of the **Asset Importer** documentation.

Each of the standard fields is described below:

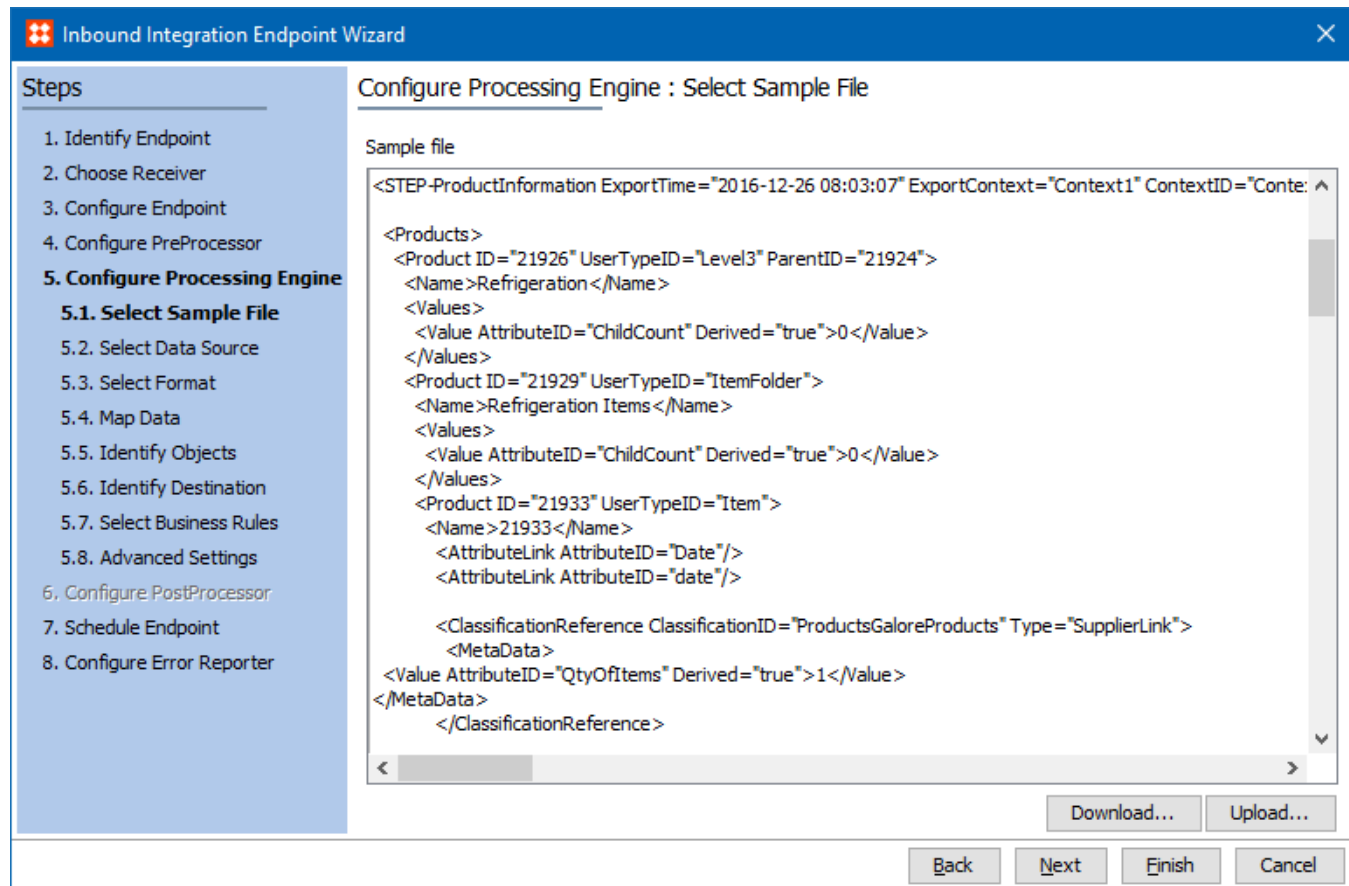
- **File Name Field:** Used in the Metafile and ZIP Metafile receiver selections to identify the file name of the asset. This field is required in each metafile as it relates the asset files to the content in the metafile.
- **Match Metadata Field:** Used in the Metadata Product Linker configuration 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 **Step 7 - 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.

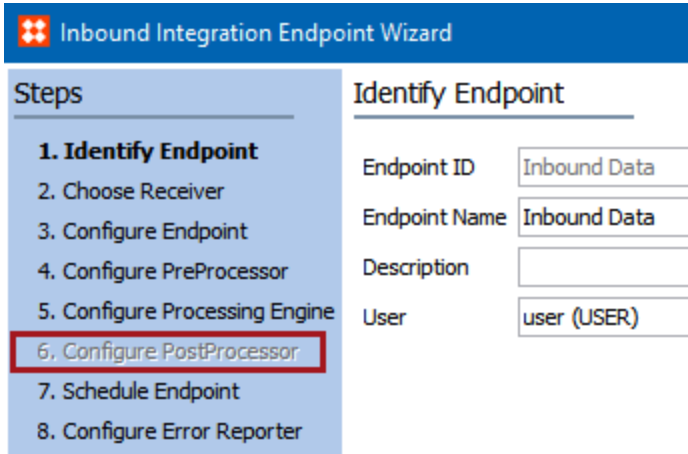
If you want to modify a sample file that has already been uploaded, click the **Download** button and save the sample file to your computer. Make your changes and then click the **Upload** button to upload the modified sample file to the endpoint.

Note: When importing ETIM6, modify the import file to include only the <IXF >, <Header>, and <Groups> sections and save it as your sample.

3. 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.
4. In **5.3 Select Format** choose from the following options to indicate the type and basic parameters of the file you are importing.

- Advanced STEPXML Format
 - BMEcat Format
 - BMEcat 2005 Format
 - CSV Format
 - eCl@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
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 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 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:

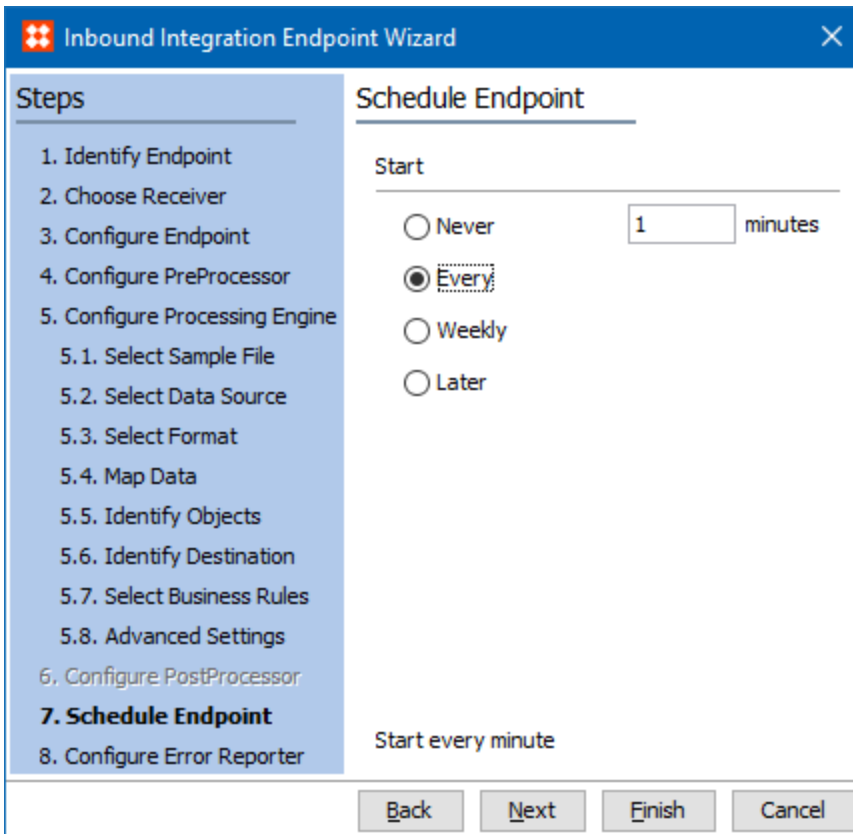
User: user (USER)

By default, STEP does not include any preconfigured 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



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

Schedule Endpoint

Start

☐ Never
 minutes

☒ Every

☐ Weekly

☐ Later

Start every minute

Use Schedule Endpoint to specify how often the endpoint should search for data to be processed considering the following:

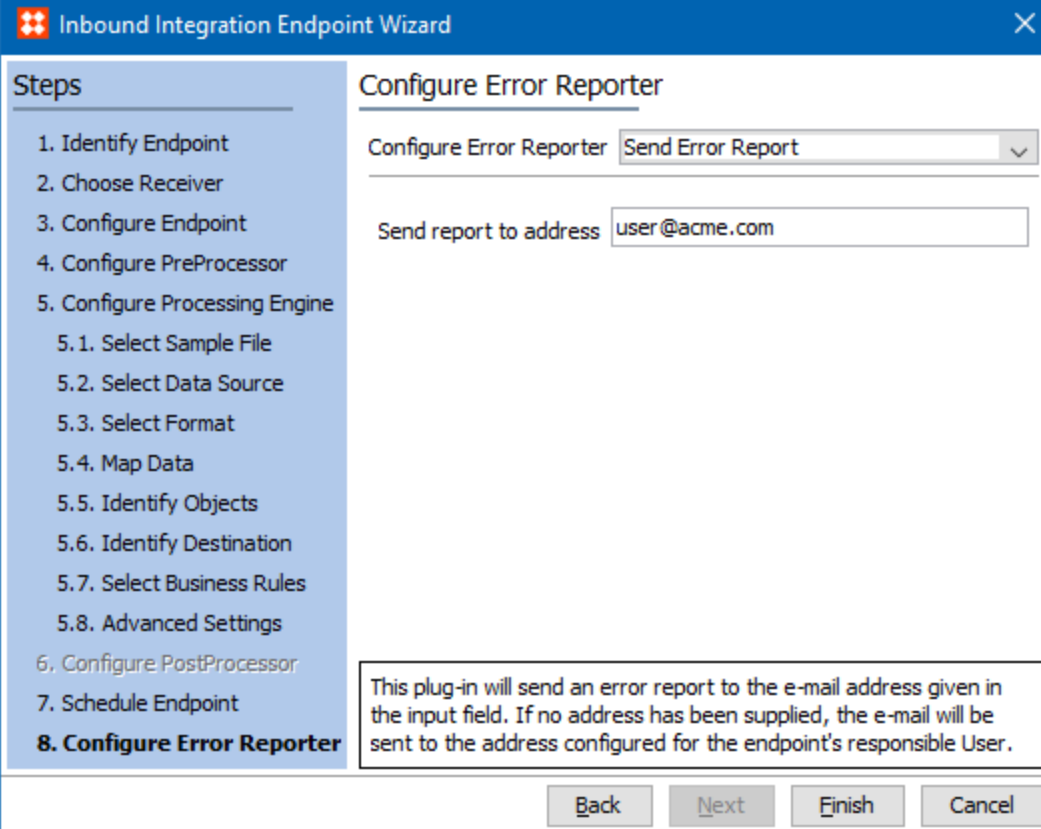
- IIEPs that do not use the REST Receiver method should typically be scheduled to poll the data source for new files / messages with regular intervals.
- IIEPs that do use the REST Receiver method can be invoked via a REST POST request. With this functionality, for example, a system delivering data to 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:

- Click **Never** if the endpoint process must be started manually. This is the default setting and should be used while testing your endpoint.
- Click **Every** to specify how many minutes will pass between runs of the endpoint process, with the minimum interval being once every minute.
- Click **Weekly** to specify specific days the process should run repeatedly, including start time, and start / stop date. Use this option if a daily schedule is needed.
- Click **Later** to run the endpoint process once at a future time.

2. Click the **Next** button to display **IIEP - Configure Error Reporter**.

IIEP - Configure Error Reporter



The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' list shows 8 steps, with '8. Configure Error Reporter' highlighted. The main area is titled 'Configure Error Reporter' and 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, there is a warning box stating: '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 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.
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.

System Setup

- Global Business Rules
- Inbound Integration Endpoints
 - 1WS_7 Receiver
 - 1WS 7.0 Provider Receiver
 - Asset Importer Endpoint
 - ETIM6
 - Inbound Data**
 - Enable Integration Endpoint...
 - Disable Integration Endpoint...
 - Edit Inbound Integration Endpoint...
 - Invoke Endpoint...
- Match Codes and Matching Algorithms
- Outbound Integration Endpoints
- Web UIs
- Workflow Profiles
- Workflows
- Derived Events
- Object Types & Structures
- Users & Groups

Inbound Data rev.1.1 - Inbound Integration Endpoint

Statistics | Error Log Excerpts | Log | Status

Inbound Integration Endpoint | Background Processes

Description

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	-

Configuration

- Hotfolder Receiver Configuration
- 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

- Global Business Rules
- Inbound Integration Endpoints
 - 1WS_7 Receiver
 - 1WS 7.0 Provider Receiver
 - Asset Importer Endpoint
 - ETIM6
 - Inbound Data** (enabled icon)
 - ProdRefs
 - Publication Excel
 - STEPXML
- Match Codes and Matching Algorithms
- Outbound Integration Endpoints
- Web UIs
- Workflow Profiles
- Workflows
- Derived Events
- Object Types & Structures

Inbound Data rev.0.1 - Inbound Integration Endpoint

Statistics | Error Log Excerpts | Log | Status

Inbound Integration Endpoint | Background Processes

Description

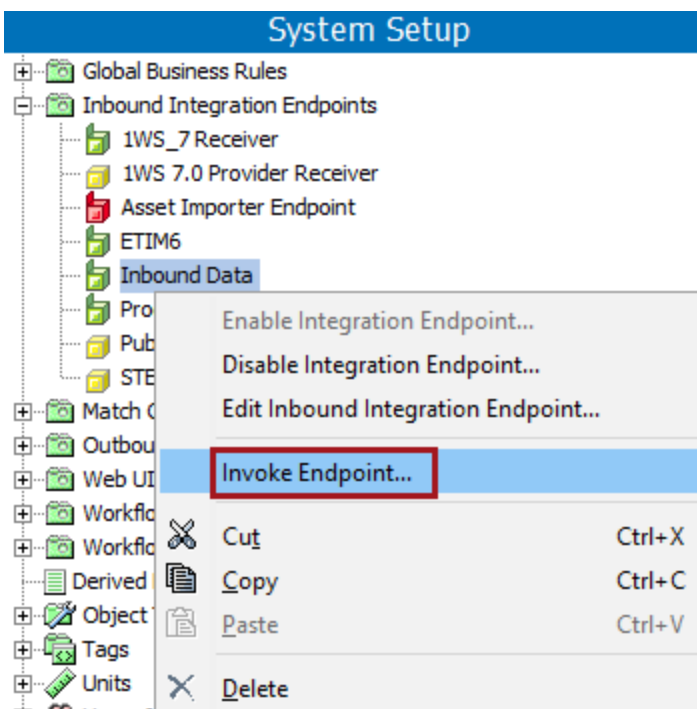
Name	Value
ID	Inbound Data
Name	Inbound Data
Object Type	Inbound Integration Endpoint Type
Revision	0.1 Last edited by Stibo Systems on Wed Jan 27 11:43:03 EST 2...
Description	
Enabled	Yes
Endpoint Status	Running
Last run	-
Next run	-

3. Open the Configuration flipper to view the Schedule parameter.

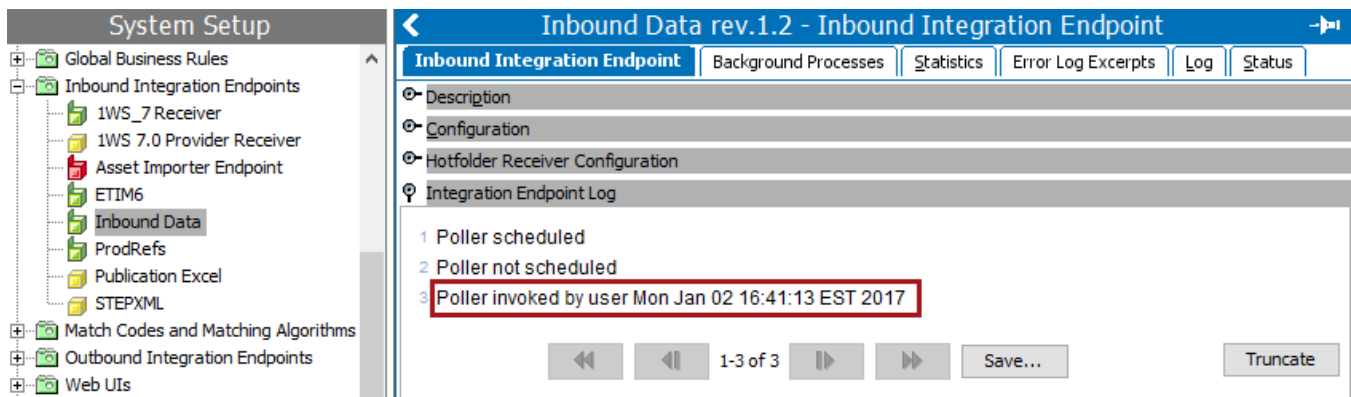
- If scheduled, the Schedule parameter shows when the IIEP will be invoked automatically.

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**.

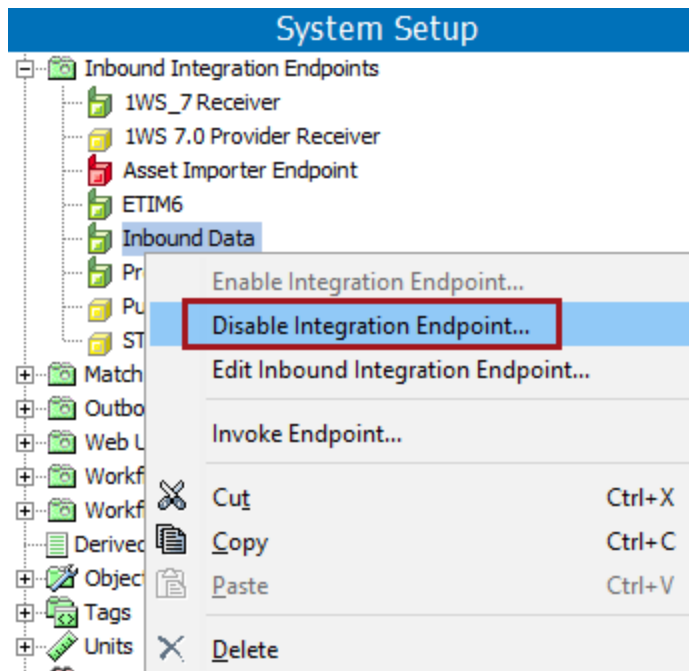


4. When invoked on schedule and manually, files or messages available in the location identified by the IIEP receiver configuration are processed. The Integration Endpoint Log flipper displays the time the endpoint was invoked. For scheduled IIEPs, the log includes a link to the background process that 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 (🟡).

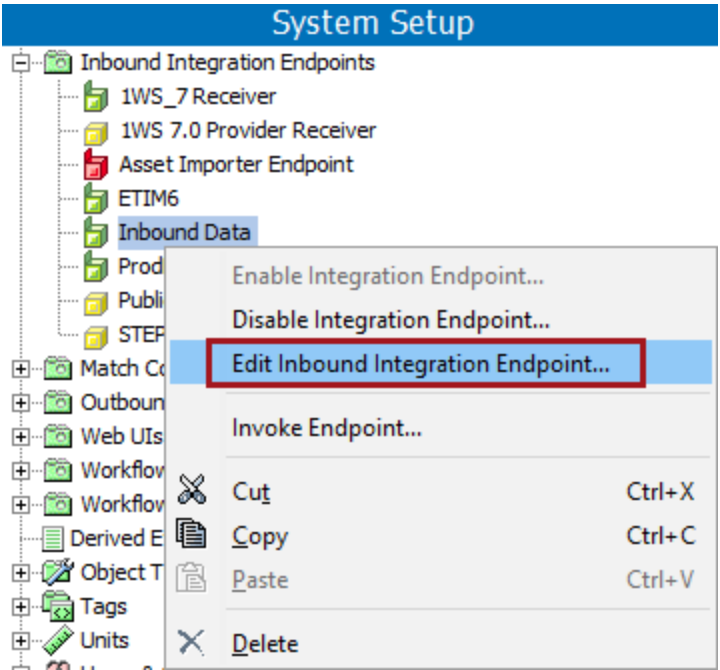


Once disabled, the inbound integration endpoint can be restarted using the Enable Integration Endpoint step defined above.

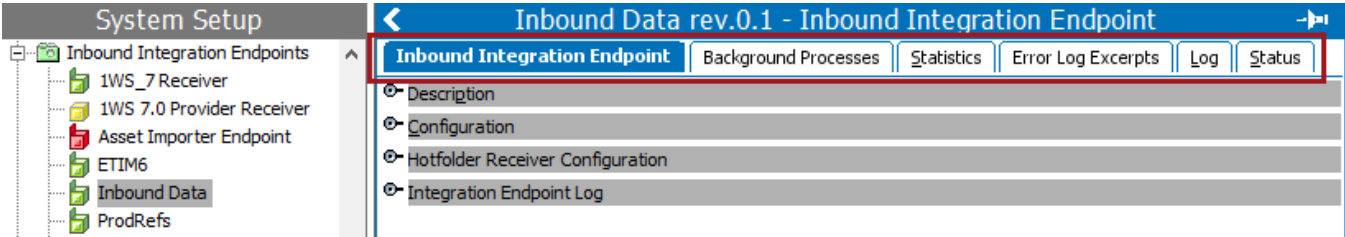
Additional information about restarting a failed endpoint, indicated by the failed icon (🔴) can be found in **Handling Failed IIEP Background Processes** documentation.

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. 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**.

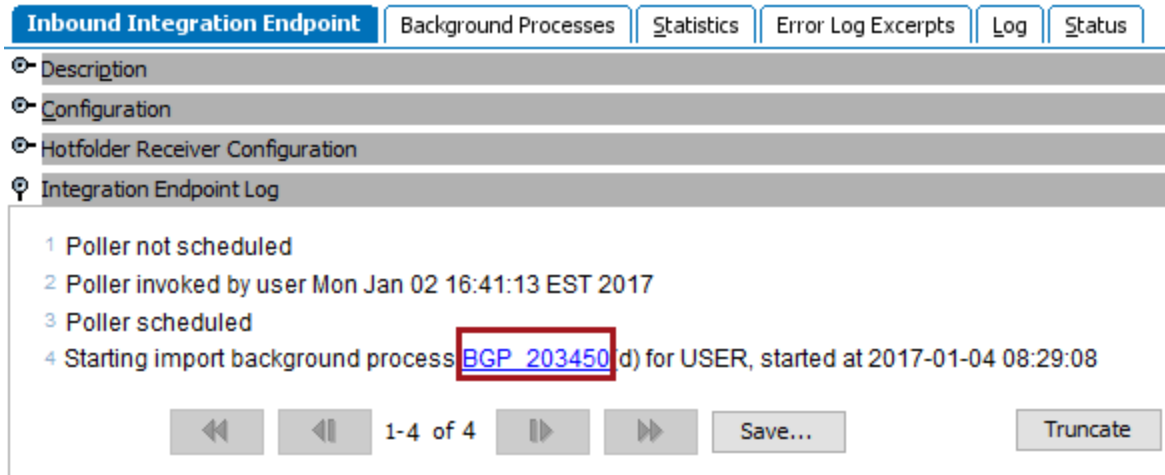
- The **Edit Receiver Plugin** link opens the wizard on the Choose Receiver step and allows you to modify the settings. For more information, see **IIEP - Choose Receiver**.

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 in parallel. 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					
Active Processes					
Completed with Errors					
Ended Processes (1)					

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).

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://clas			
> BGP_206613	150	Row 3: Parent classification '18203' for classification '<a href="step://clas			

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.

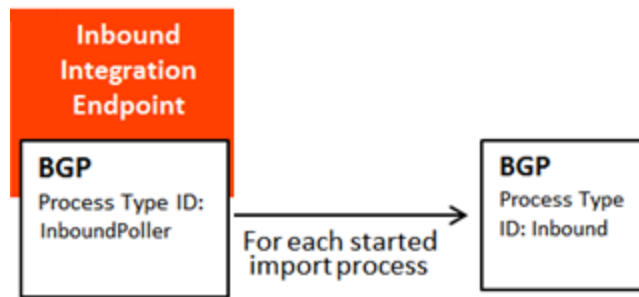
Inbound Integration Endpoint	Background Processes	Statistics	Error Log Excerpts	Log	Status
Revisions					
Revision	Created	Edited	Major		
> 1.2	Mon Jan 02 16:33:01 EST ...	Wed Jan 04 10:51:35 EST...			
> 1.1	Tue Jun 28 10:43:57 EDT ...	Tue Jun 28 10:43:57 EDT ...			
> 1.0	Fri Feb 12 14:00:19 EST 2...	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 endpoint...	succeeded	100%	Wed Jan 04 08:29:13 EST ...	USER	0	Wed Jan 04 08:29:08 EST ...

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 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 (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 parallel 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

Error Message	Description
	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.
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	You cannot make a product-to-product reference from

Error Message	Description
product y of Type z, reference data will not be imported.	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 imported.	You cannot make a product-to-asset reference from the same source product to the same target asset more than 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'.

Error Message	Description
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.
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 a 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 "1/2" STRAIGHT WIDGET SSU05PL" for attribute Description in product T100-3526 (UnknownUnit).
Source: Line=2, Column=Column2.
- Illegal unit "1/4" ANGLED GADGET BZT10GL" for attribute Description in product T100-2625 (Unknown Unit).
Source: Line=3, Column=Column2.
- Illegal unit "1/4" 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

Here 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_BEFF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_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.

Here 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_BEFF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_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_BEFF_INS", line 18 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_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_BEFF_INS", line 41 ORA-04088: error during execution of trigger 'STEPSYS.VALUEMAP_BEFF_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_UX) violated {prepstmt 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:

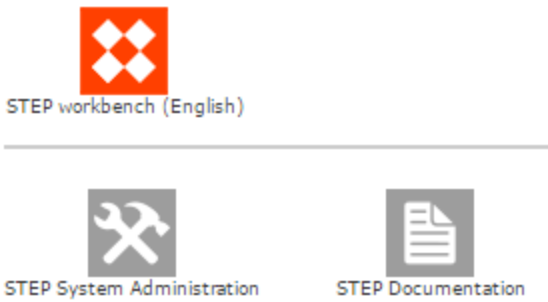
Description	Importing
Execution Server	doc-rel
Progress	100%
Status	completed with errors
Created	Sun Feb 05 06:49:19 EST 2017
Started	Sun Feb 05 06:49:20 EST 2017
Finished	NA
Processing Time	0 m 2 s
Time in Queue	0 m 1 s
# of warnings	0
# of errors	1

 Execution Report

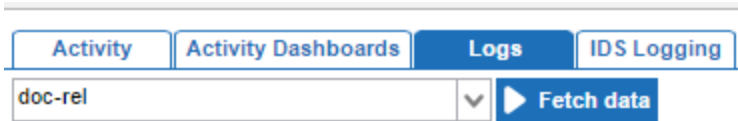
- 1 Retrieval started (Sun Feb 05 06:49:21 EST 2017)
- 2 Retrieved 8817 bytes (Sun Feb 05 06:49:21 EST 2017)
- 3 Conversion started (Sun Feb 05 06:49:21 EST 2017)
- 4 Converted 6 objects (Sun Feb 05 06:49:24 EST 2017)
- 5 Logged on
- 6 Mapping started (Sun Feb 05 06:49:24 EST 2017)
- 7 Mapping completed (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:

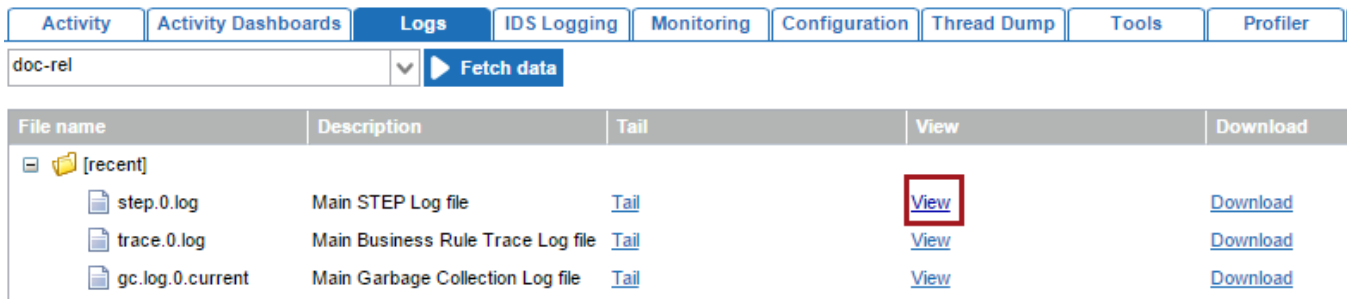
1. Launch the STEP Web start page
2. Click on the System Administration link



- 3. Once logged in, click on the Logs Tab
- 4. Select the Server in the drop down 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.getConfiguratonPackagesLogFileNames(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.

On the Select Format step of the outbound tool, 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>
```

An example of the output for an inbound integration endpoint definition as comments is shown below:

```
<InBoundIntegrationEndpoint...>
<!--Definition: [IIEP definition goes here. Removed for brevity.-->]
[Remaining configuration would populate here]
</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 **Include Integration End Points** 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 belong 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

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.

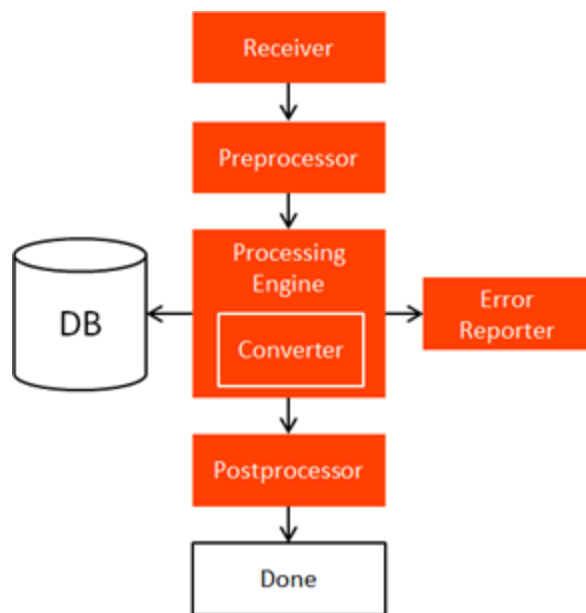
Error Log Excerpts		
Outbound Integration Endpoint	Configuration	Background Processes
Statistics	Error Log Excerpts	Log
Integration Endpoint Log		
Background Pro...	Log Item No	Text
> BGP_4730623	30	Object without matching format template. Object type id: 'Product', object id: 'A-1001-3'
> BGP_4730623	40	java.lang.NullPointerException
> BGP_4730623	50	Caught RuntimeException at Fri Jan 23 09:33:46 CET 2015: java.lang.NullPointerException
> BGP_4730626	30	Object without matching format template. Object type id: 'Product', object id: 'A-1001-3'

2. In the **Background Process** column, click the link for the background process to correct.
3. In the Background Process editor **Execution Report**, view the progress of the background process, and where it failed.
4. In the Properties area > **Value** column, click the Save  button near the failed status, and save the file locally.

5. Open the file in a relevant editor and make the required changes.
6. After completing the changes, upload the file back to the same background process. In the Background Process editor > Properties area > **Value** column, click the Upload  button near the failed status.
7. Restart the same process in the Properties area > **Value** column, click the Restart  button near to the failed status. The background process is restarted.
8. If the background process is part of a batch of chained processes, you also have to 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.

Pre-Processor

A pre-processor 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 pre-processor 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.

Post-Processor

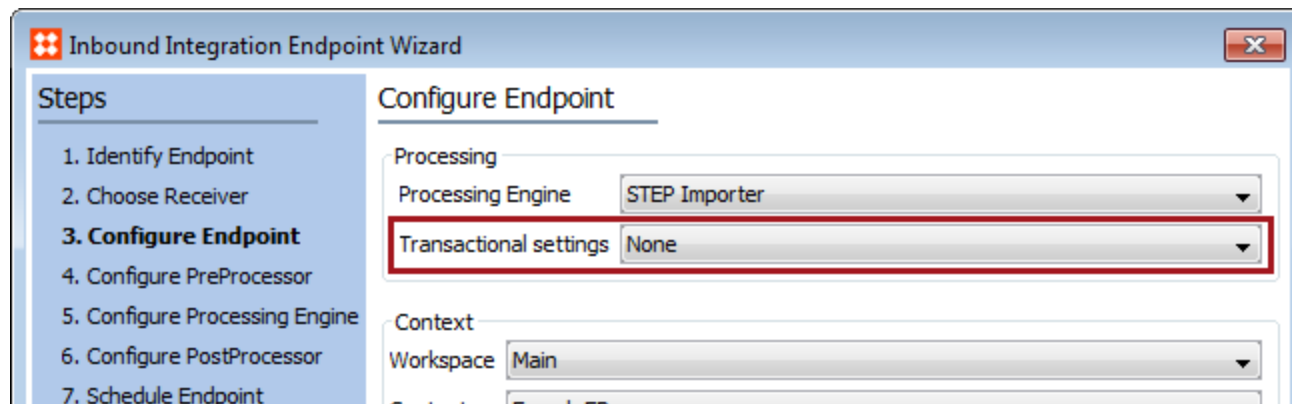
A customized post-processor plugin has access to import events (information about what has changed) and, based on the events, can trigger any required system change. For more information on events, see the **Events** documentation in the **System Setup / Super User Guide**.

By default, STEP does not include any preconfigured post-processor plugins, since post-processor logic is also typically implemented via business rules referenced from the processing engine configuration.

Inbound Integration Endpoint Transactional Settings

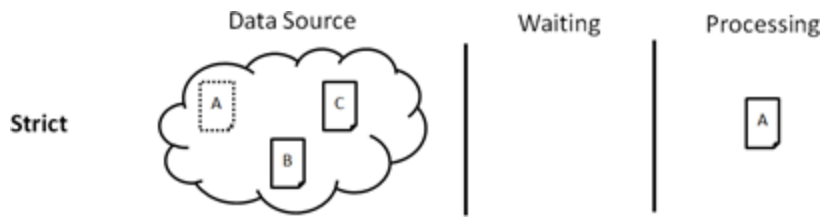
Transactional Settings specify how background processed will be generated, how the messages will be processed, and what will happen when it fails. The selected Receiver and Transactional Setting determines what other parameters are available in the additional wizard steps. There are three possible transactional 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.

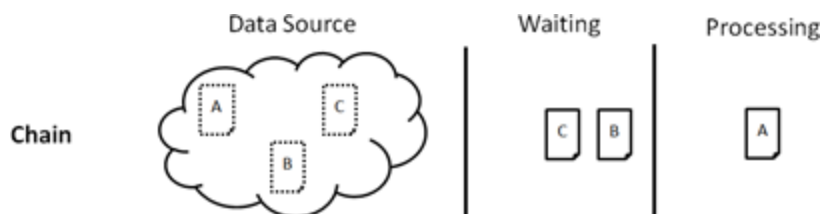
Chain

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.

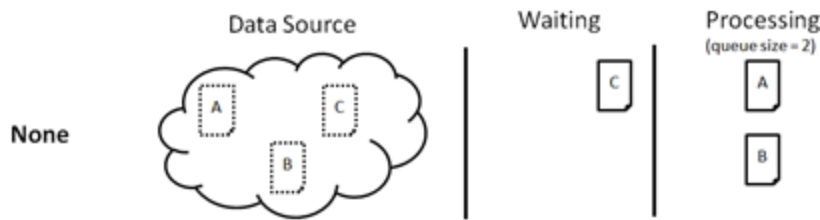
Note: When creating or modifying an endpoint for Transactional Setting = Chain, after selecting the Chain option, close the wizard and reopen it to display the 'Maximum number of waiting processes' parameter.

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 in parallel / 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. Non is not allowed for the Hotfolder Using File Sequence Receiver or Hotfolder Using Meta Files Receiver methods.

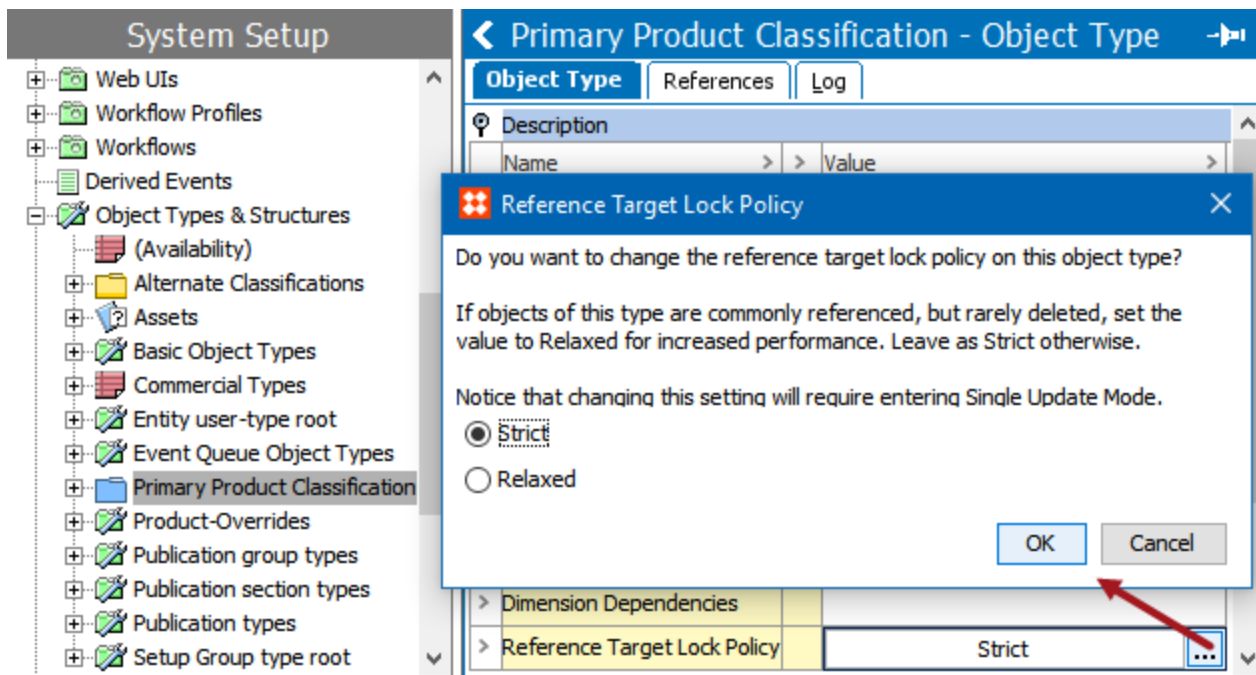


Reference Target Lock Policy on Object Types

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. This can be an issue when running parallel imports where the first import locks the object type being referenced and the second import eventually stops running because it cannot access the locked object. The default setting for this property is set to 'Strict' unless changed to 'Relaxed', and 'Relaxed' should NOT be used on an object type for objects that are often deleted – this could result in poor performance, and risk of locks and deadlocks.

When a reference between two objects is being created in STEP, the target object will be 'locked' to ensure that it is not being deleted while the reference is being created. In most of these cases, deletion of the object is not a deletion from the database, but the deletion that occurs due to revision control when an insert of a new history entry occurs. This can be a major issue when running parallel imports where the first import locks the object type being referenced, and the second import eventually stops running because it cannot access the locked object. Although STEP continues to retry the import, this can cause inbound feeds to be negatively affected.

The 'Relaxed' setting on the property puts a less restrictive lock on the reference target objects being edited so that they can be updated concurrently by more than one process and/or user, with a full lock being employed only in the case that a deletion is attempted. This setting, when used with object types that are frequently referenced but rarely deleted, will improve the performance and stability of parallel inbound imports, bulk updates, and users concurrently creating references to the same objects.



These Relaxed / Strict settings can be changed back and forth as needed, but changing the setting requires the system to enter Single Update Mode for a short time, only as long as it takes to update the setting. Keep in mind, however, that changing this setting during peak production hours can have a negative impact on performance.

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 / STEP Super User Guide** documentation. For information about setting up an event processor, see **Event Processors** in the **System Setup / STEP Super User Guide**.

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. Maintain or modify the endpoint from within STEP as described in **Monitoring an OIEP via Background Process**.
2. Maintain or modify the endpoint outside of STEP as described in **Monitoring an OIEP via External Systems**.
3. Understand the effects of batching on an OIEP as described in **Event-Based OIEP Event Batching**.
4. Understand the effects of manipulating events on an OIEP as described in **Event-Based OIEP Forward, Rewind, Purge, and Republish**.
5. Understand the multithreading functionality on an OIEP as described in **Event-Based OIEP Multithreading Support**.
6. Understand queued events on an OIEP as described in **Event-Based OIEP Queued Events**.
7. Understand the relationship between OIEP Status and Queue Status as described in **Event-Based OIEP Status and Queue Status**.
8. Understand each element of an OIEP as described in **Outbound Integration Endpoint Structure**.
9. 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:

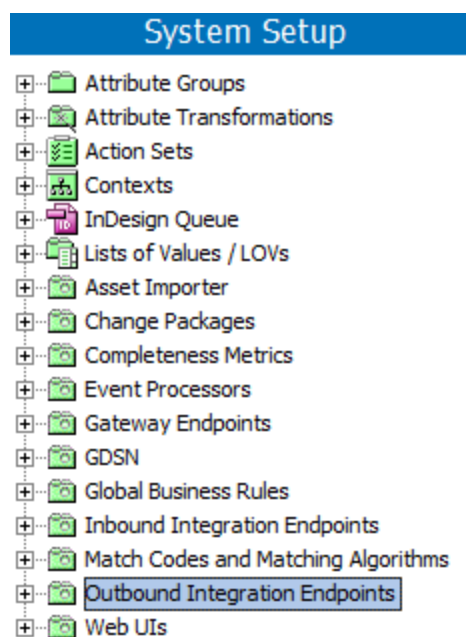
1. Determine the progress of the endpoint from within STEP as described in **Monitoring an OIEP via Background Process**.
2. Determine the progress of the endpoint outside of STEP as described in **Monitoring an OIEP via External Systems**.
3. Maintain or modify the endpoint as described in **Maintaining 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 / STEP Super User** documentation.

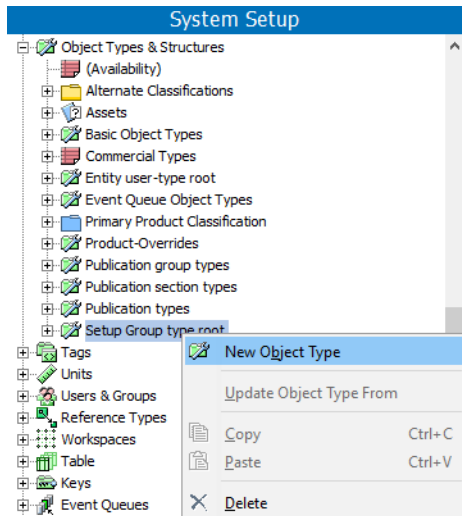
Review your System Setup tab to determine if one or more outbound integration endpoint nodes already exist. The name of the node on your system is not required to match the one in the image below.



Once the setup has been completed, the steps in this section are only needed if additional levels of organization are desired.

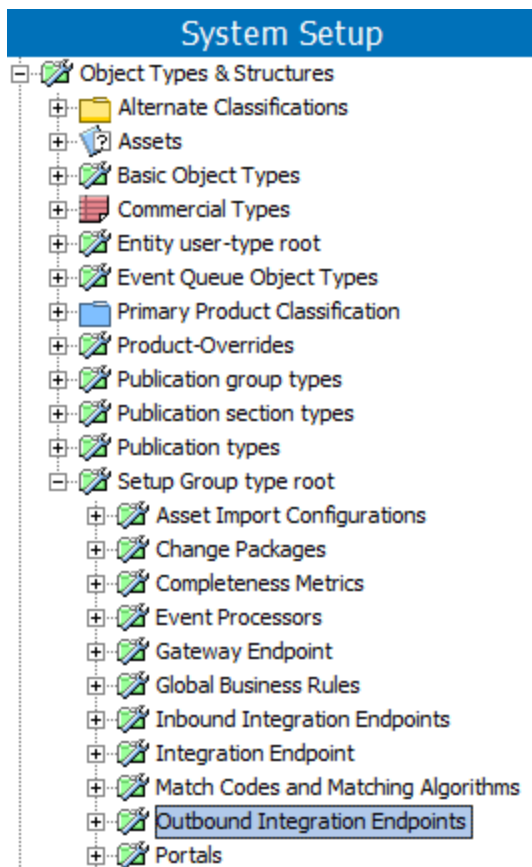
Create the Outbound Integration Setup Group

1. In System Setup, expand **Object Types & Structures**.
2. Right-click 'Setup Group type root', and choose **New Object Type**.



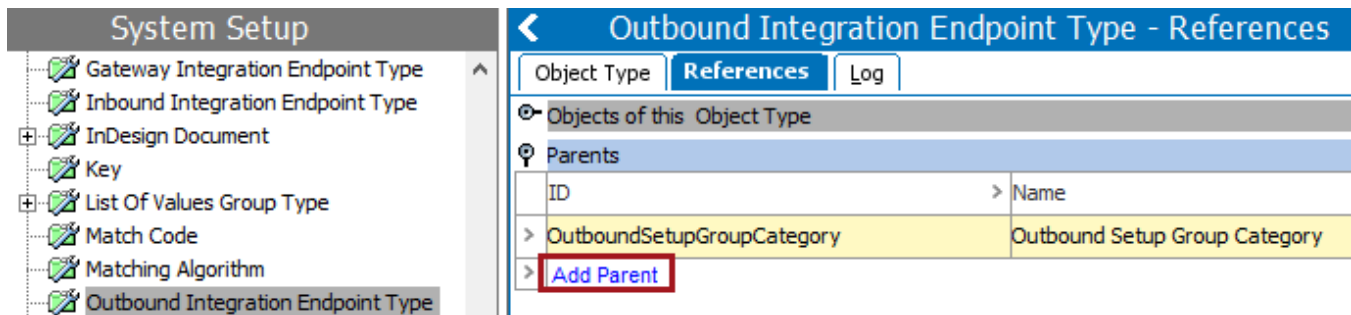
3. Enter an **ID** and a **Name**, select any required Dimension Dependencies, and click **Create**.

4. The new setup group appears in System Setup under 'Object Types & Structures' as a child in the **Setup Group type root**.

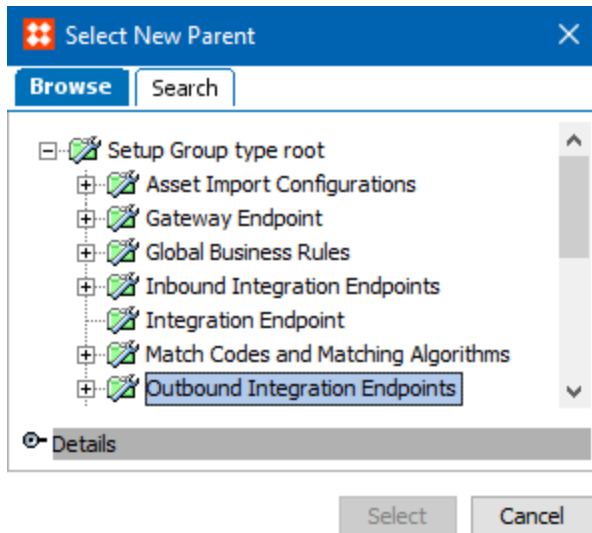


Link the OIEP Object Type to Setup Group

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.

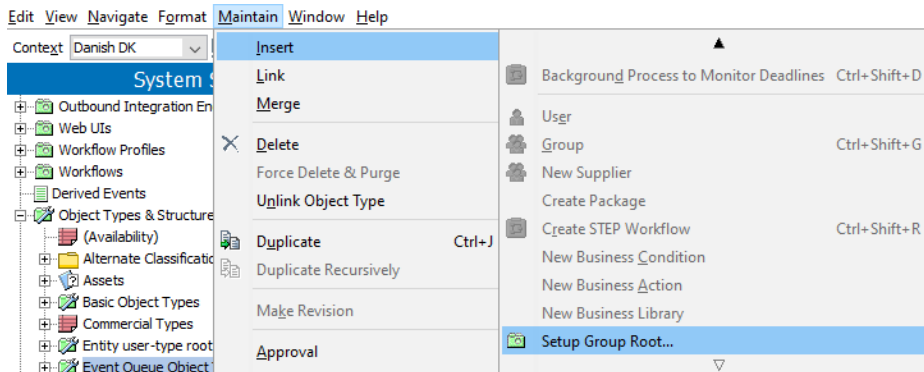


3. In the Select New Parent dialog, select the setup group you created, and click **Select** to make it a valid parent.

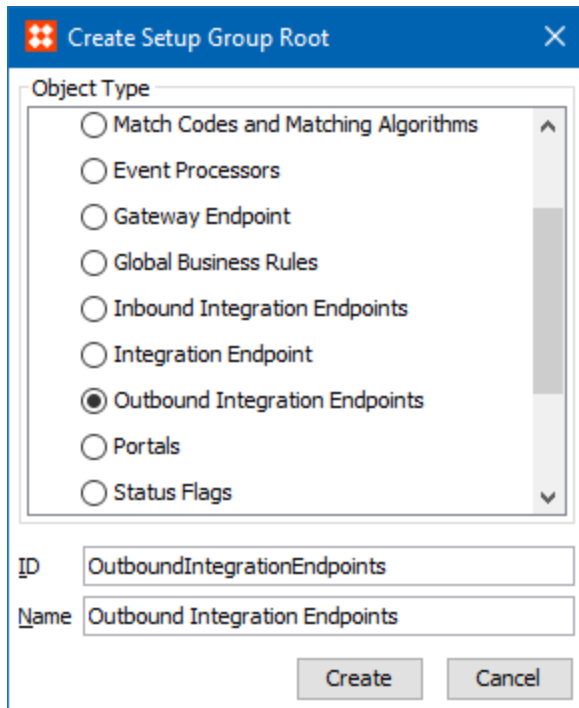


Create an Instance of the OIEP Object

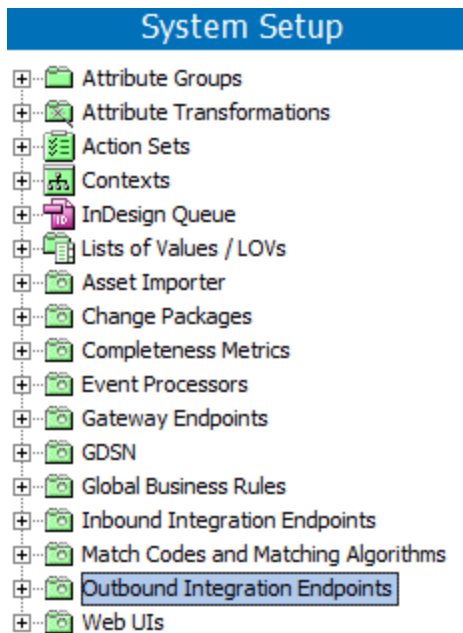
1. On the System Setup tab, select any object in the **System Setup** hierarchy to activate the following Maintain menu selection.
2. Click the Maintain menu, point to Insert, and select **Setup Group Root**.



3. In the Create Setup Group Root dialog, select the inbound integration endpoints object type, enter an **ID** and a **Name**, and click **Create**.



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 Outbound Integration Endpoint

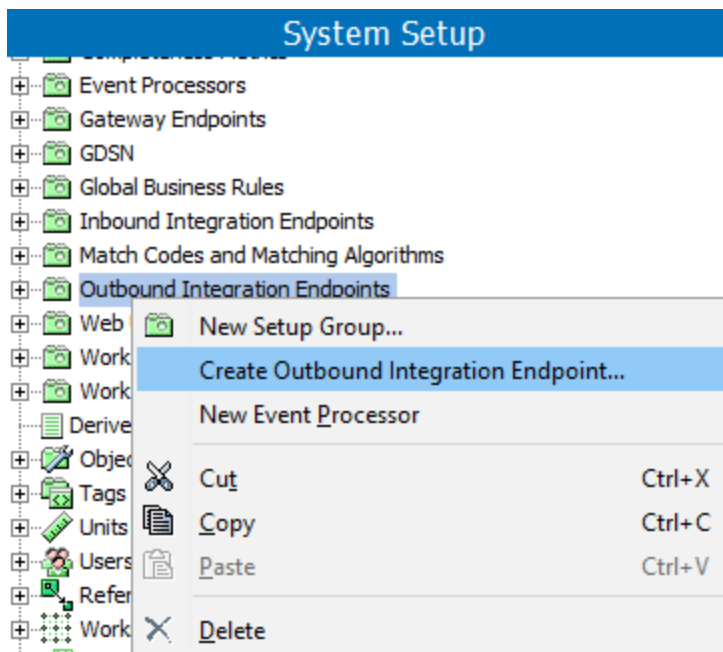
After creating a setup group for outbound integration endpoints, use the following topics to create the type of OIEP to meet your export needs:

- 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

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:

The screenshot shows the 'Identify Endpoint' step of the 'Outbound Integration Endpoint Wizard'. On the left, a 'Steps' sidebar lists three steps: '1. Identify Endpoint' (highlighted), '2. Choose Data Source', and '3. Configure Endpoint'. The main area contains four input fields: 'Endpoint ID', 'Endpoint Name', 'Description', and 'User'. The 'User' field has a dropdown arrow. At the bottom, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

3. 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.

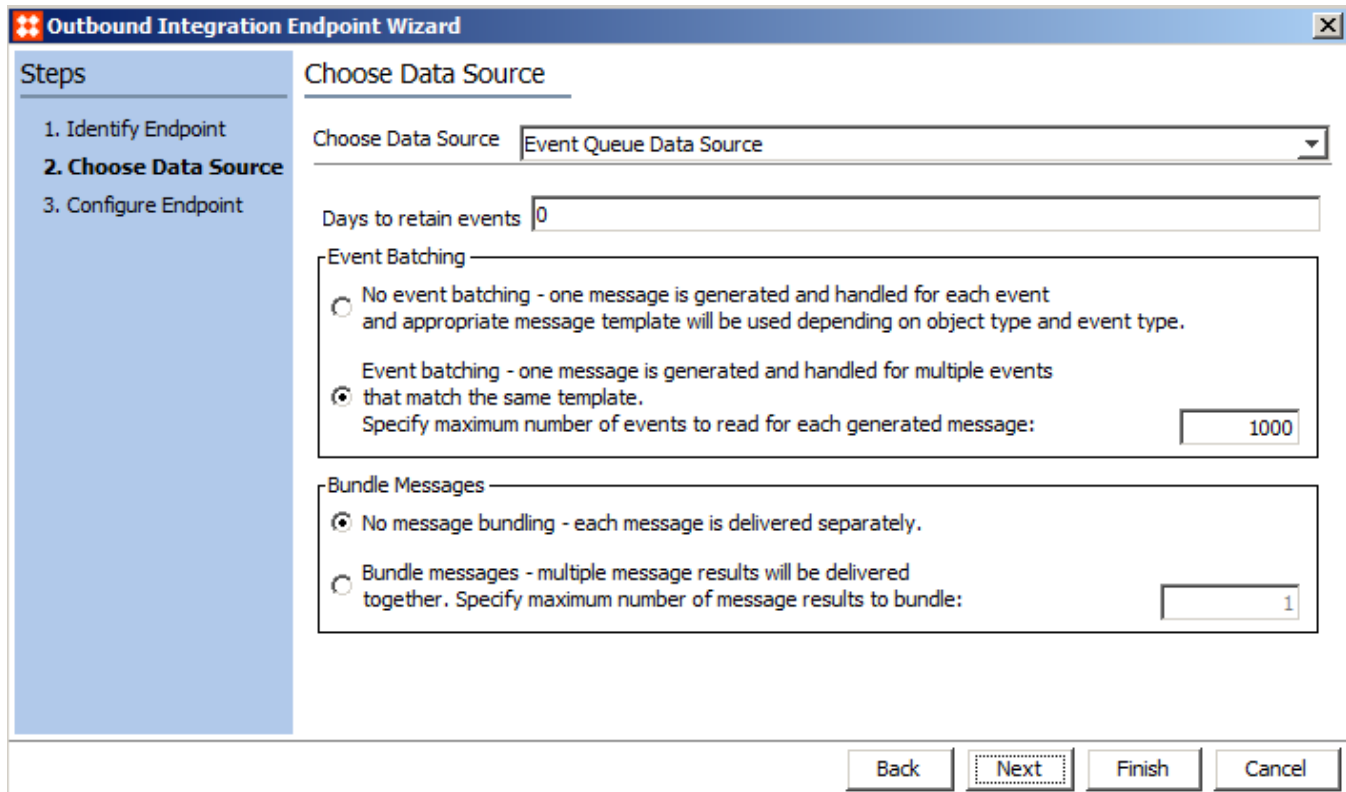
OIEP - Event Based - Identify Endpoint

This screenshot shows the 'Identify Endpoint' step with sample data entered into the fields. The 'Steps' sidebar is the same as in the previous image. The input fields now contain: 'Endpoint ID' (Outbound Endpoint), 'Endpoint Name' (Outbound Endpoint), 'Description' (Text), and 'User' (user (USER)). The 'Back', 'Next', 'Finish', and 'Cancel' buttons are at the bottom.

An event-based OIEP allows you to export data to be published as incremental changes are made.

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**.
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. Common setup is to create a special system user for this purpose so that the effects of the endpoint are easily identified.
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. Common setup is to use No Message Bundling since this option is only supported by some delivery plugins.
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 three steps: '1. Identify Endpoint', '2. Choose Data Source', and '3. Configure Endpoint', with the third step being the active one. The main content area is titled 'Configure Endpoint' and is divided into two sections. The 'Processing' section contains three fields: 'Processing Engine' set to 'STEP Exporter', 'Transactional settings' set to 'Strict', and 'Number of threads' set to '1'. The 'Multi context' section contains a 'Workspace' dropdown set to 'Approved' and a 'Contexts' list showing 'English US' and an 'Add Context' link. At the bottom right, there are four buttons: 'Back', 'Next', 'Finish', 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 Stiibo 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. 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. 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.

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	Discard Events
> Unread events (approximated)	Read Events

Edit Configuration

- 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 of the parameters are described below.

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

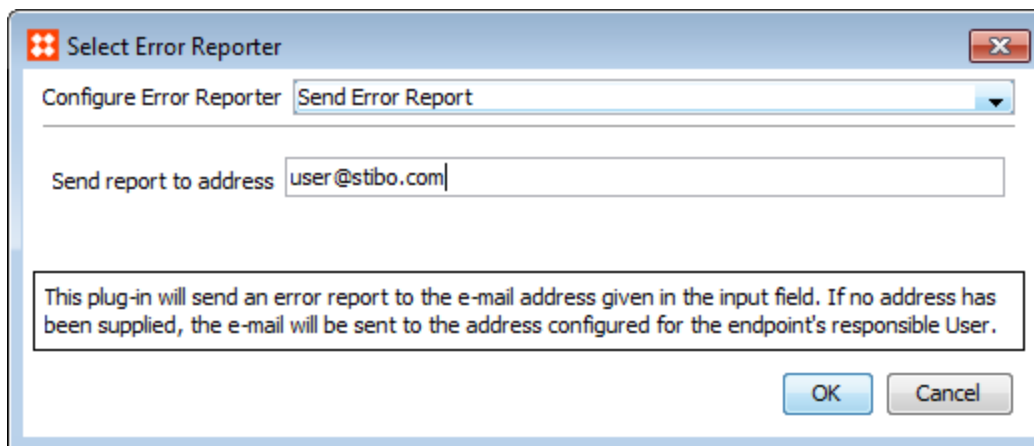
The processing engine is originally selected in the wizard when configuring the endpoint. For more information, see **OIEP - Event-Based - Configure Endpoint** documentation or **OIEP - Select Objects - Configure Endpoint** documentation.

Error Reporter

The 'Error reporter' parameter is not included in the wizard, but it can be configured to send an email error report 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 customizations) includes only the 'Send Error Report' option.

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 the email address of the error report recipient.

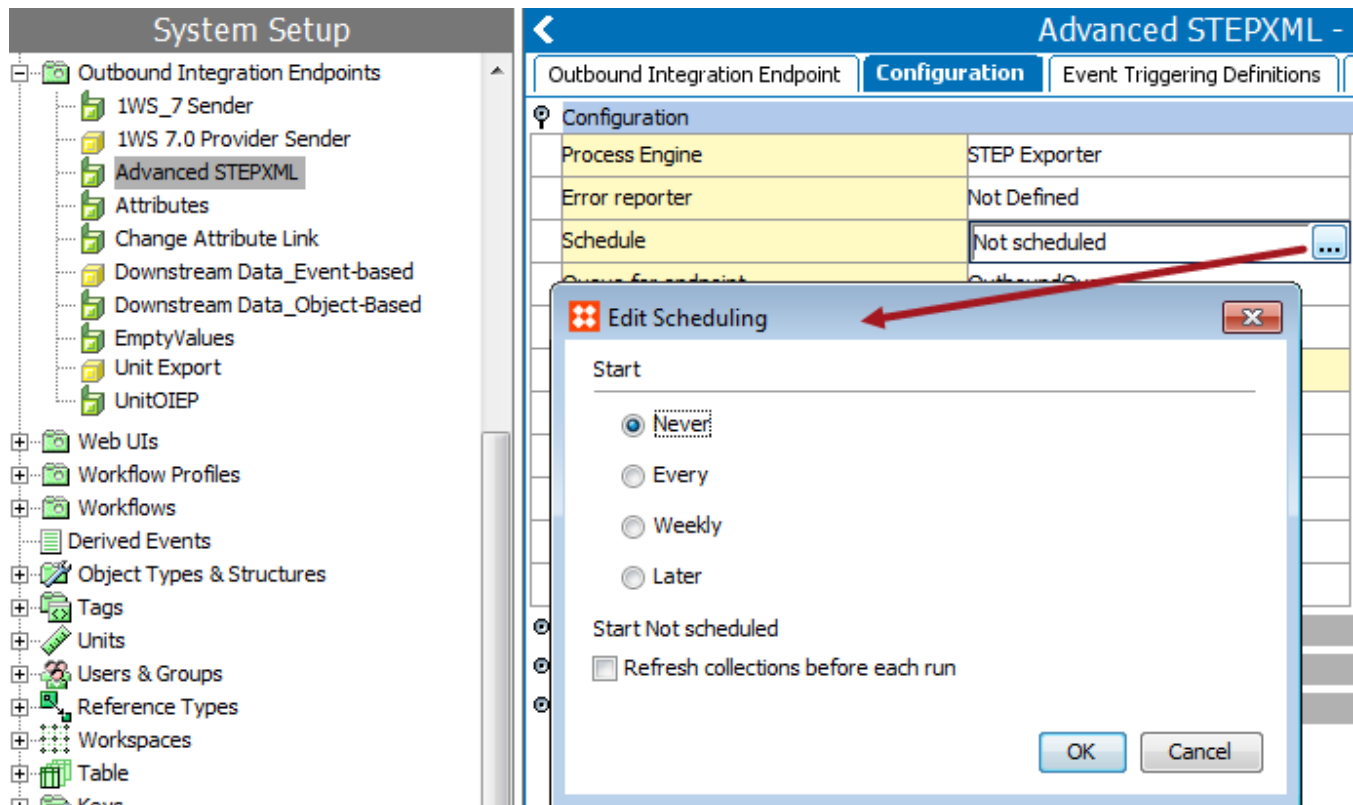
Note: If no email address is entered, the error report will be sent to the email address of the user who created the integration endpoint. If no email is defined for this user, or if no mail server is defined in the configuration, the error report will be written to the failed Background Process Execution 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** if you want to invoke the endpoint manually.
 - **Every, Weekly, or Later** to specify how often you want the endpoint to run. Different options are available depending on the frequency you select.



3. 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 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.

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.

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.

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.

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:

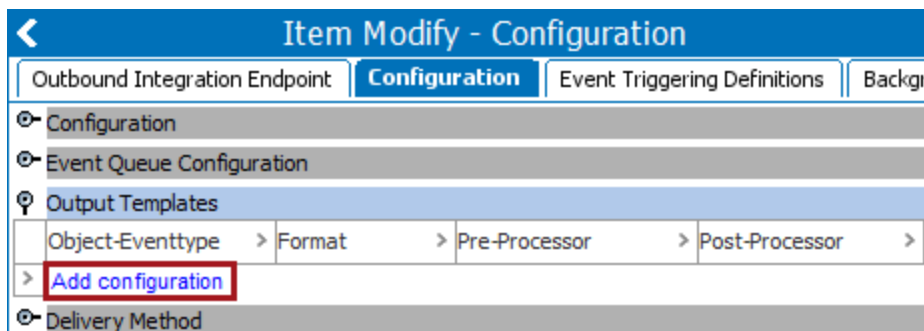
- Configure the object types and event types
- Specify the format
- Configure pre-processor and post-processor, based on availability on your system.

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

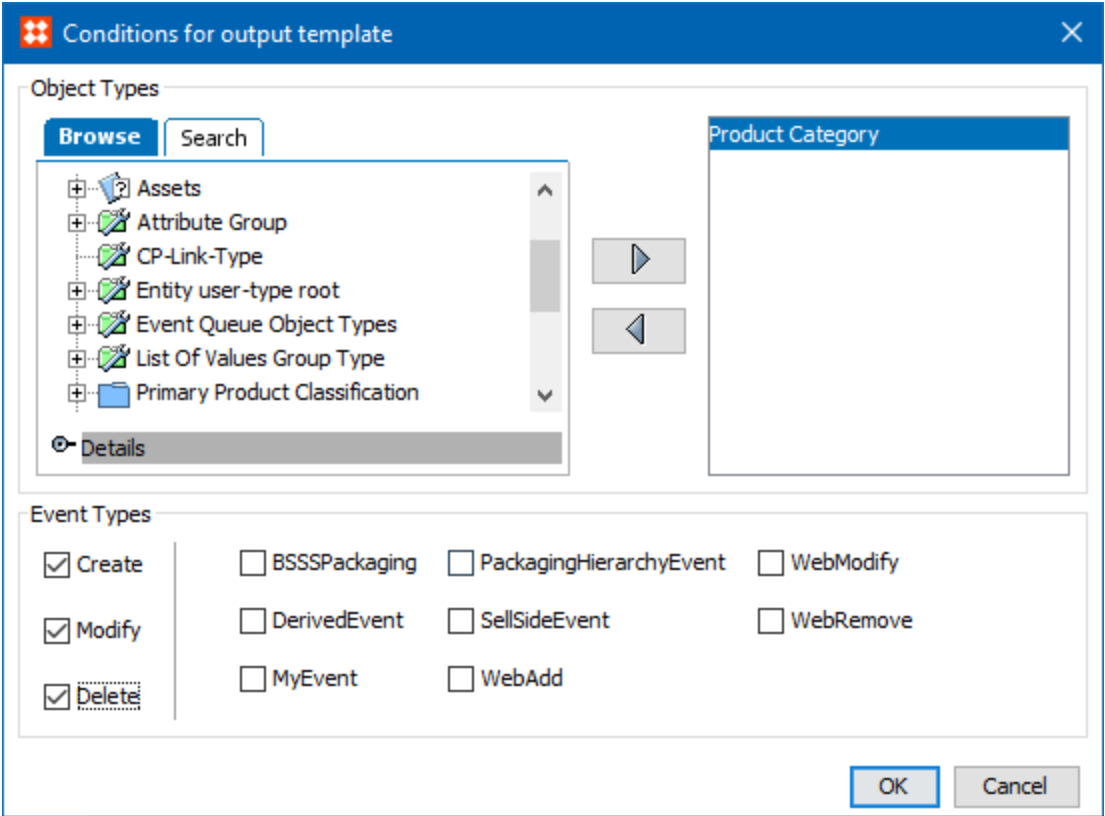
The object type and event type selections determine the STEP data that will be exported.

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 type(s) and event type(s). The basic system events (create, modify, and delete) are displayed in the **Event Types** area, along with any configured derived events. For more information, see **Derived Events** section of the **System Setup / STEP Super User Guide** documentation.

Note: Selecting an object and event type here 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-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

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 a 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 dropdown. These are the same options as in the Export Manager. 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.

Include Validation

None

Include Type Definitions

None

Include Global Settings

None

Include Context Definitions

None

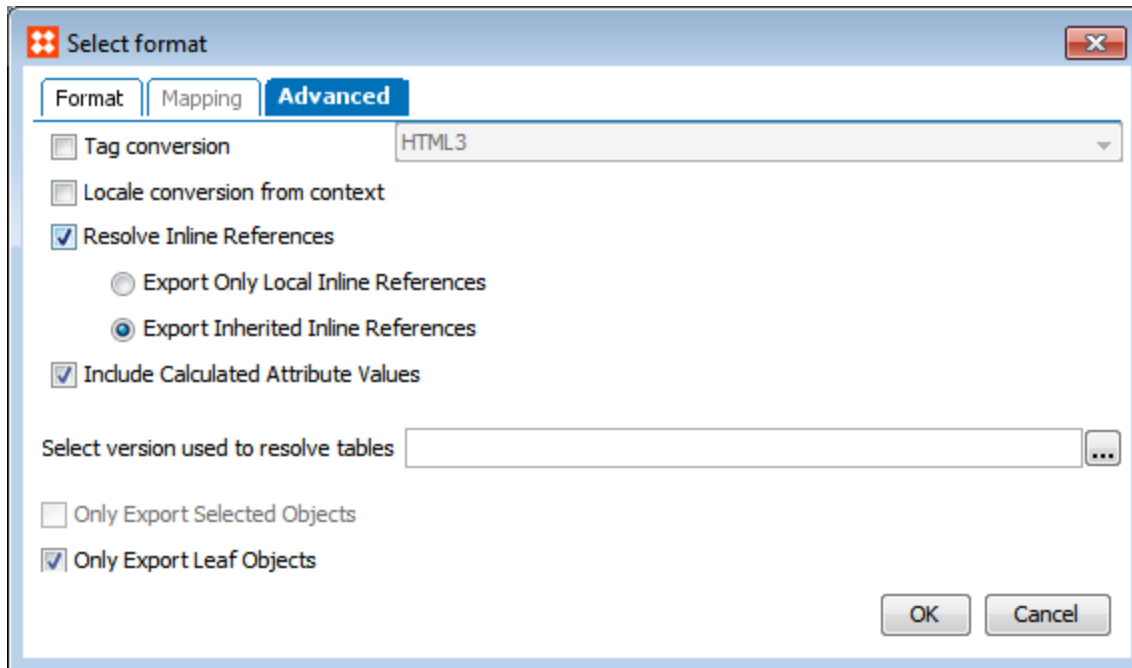
OK

Cancel

- Depending on the format selected, the **Mapping** tab may be enabled. When available, specify how to map data to the export format. For more information, see **Export Manager - Map Data** documentation.

If the selected format is automatically mapped (such as STEPXML or Advanced STEPXML), 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 / Super User Guide** documentation.

- **Locale conversion from context:** Converts numbers into the numeric format that corresponds to the selected locale. **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 / STEP User Guide**.

- **Include Calculated Attribute Values:** Resolves calculated attribute values when they are exported. If not selected, calculated attribute values are exported with empty values. 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. Specifies which version of a given publication should be used to resolve tables. If a publication is not selected, the version last used in STEP workbench is used.
- **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.

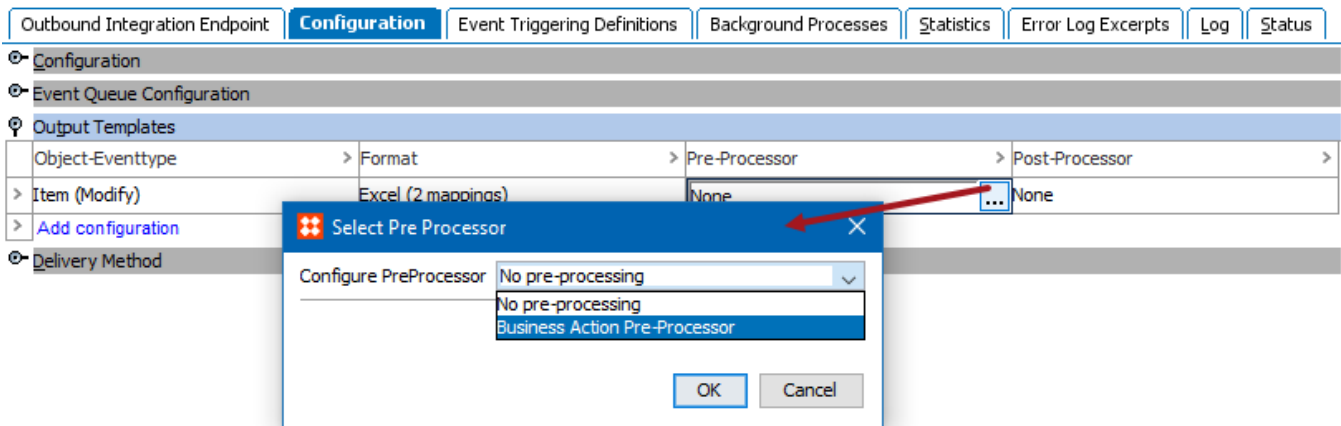
For more information, see **Export Manager - Advanced** documentation.

Configure the Pre-processor and Post-processor

Pre- and post-processor 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 pre-processor 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 pre-processor for each of the output templates. For more information, see **OIEP - Event-Based - Pre-Processor Configuration Example**.

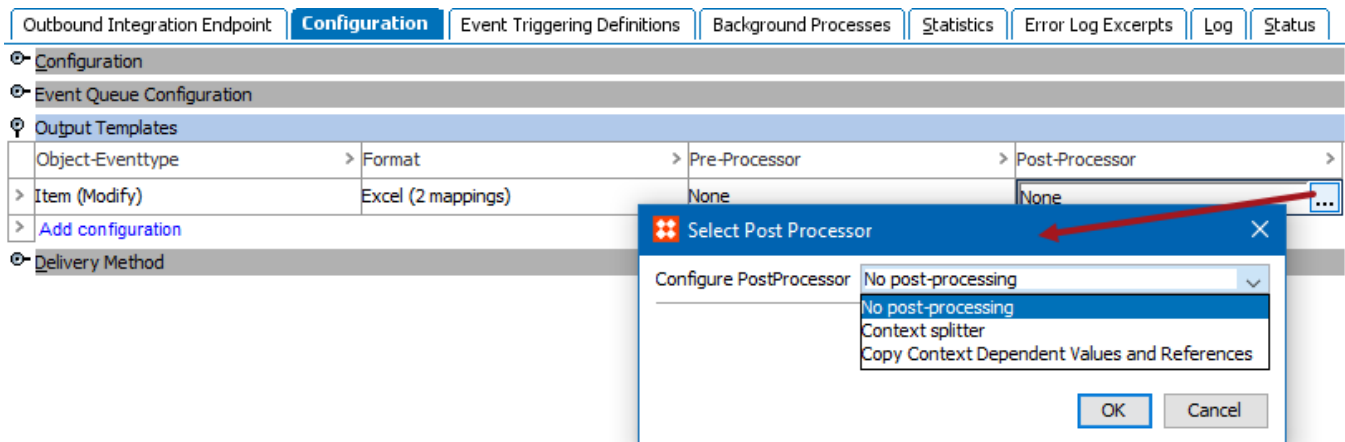


A Triggering Object Type event filter or Generate Event option can be used in place of a pre-processor. For more information, see the **OIEP - Event-Based - Event Triggering Definitions Tab** documentation.

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.

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 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 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.

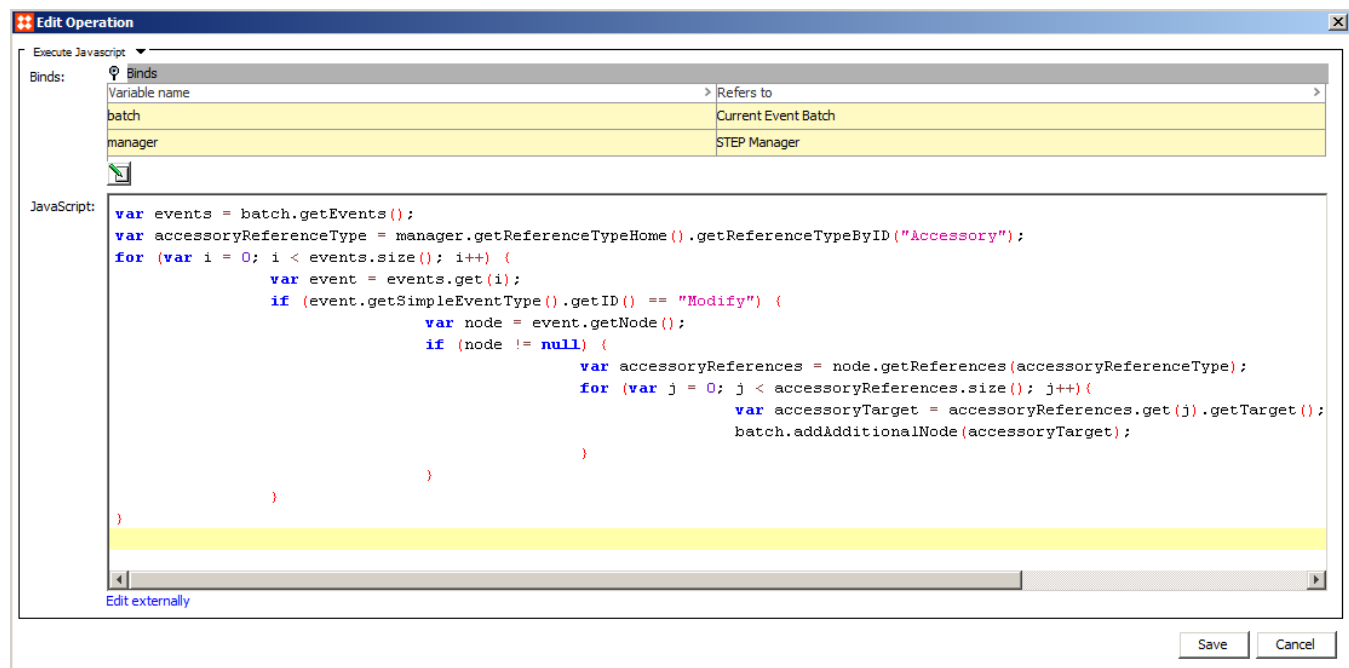
If you use the **Copy Context Dependent Values and References** post-processor to add ContextIDs to a cross-context export, the downstream system can use the <ContextID> tag to identify specific contexts. Additionally, this facilitates re-importing the exported data into the correct contexts in STEP after processing by a third-party service or application.

For more information, see the **OIEP Post-Processor - Copy Context Dependent Values and References** section.

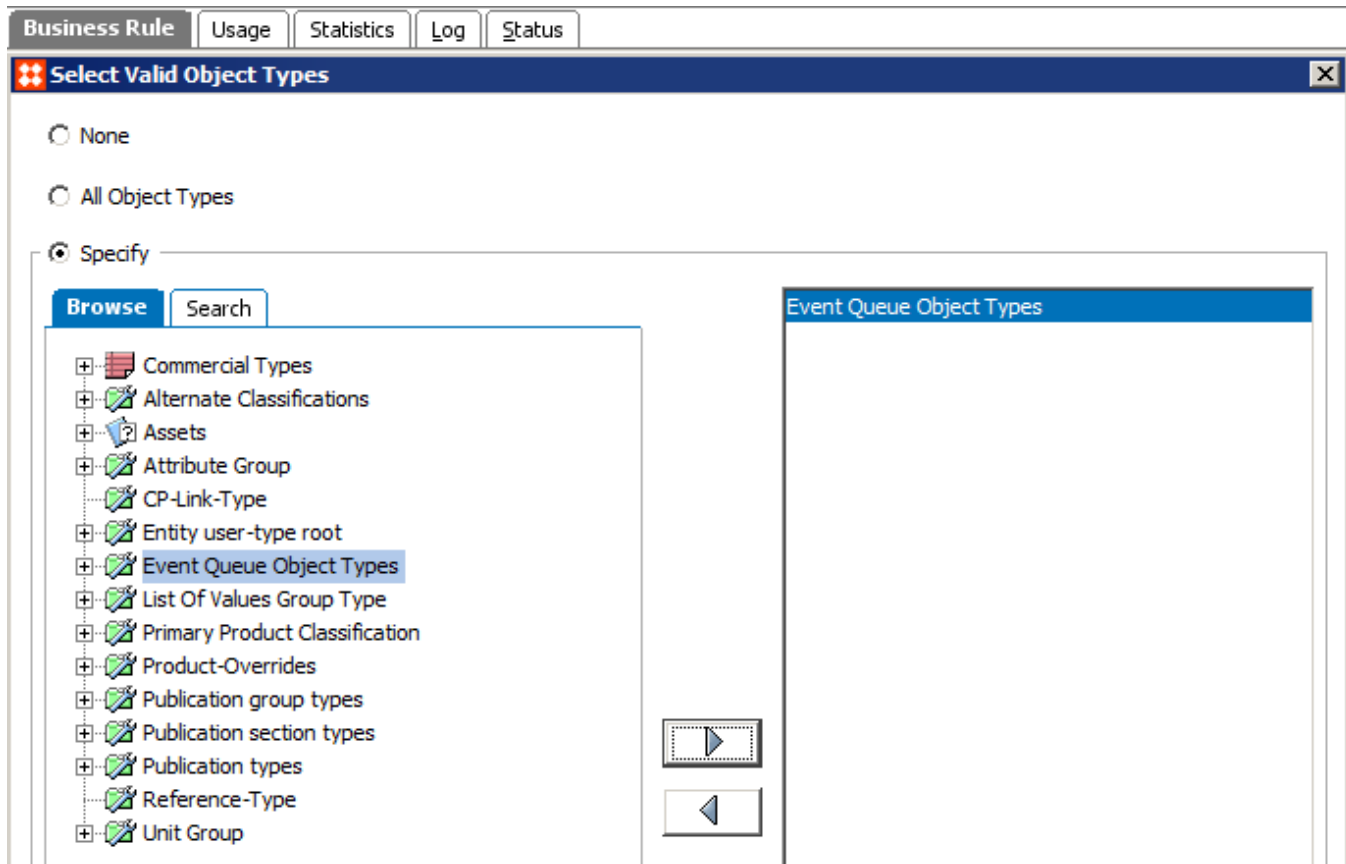
OIEP - Event-Based - Pre-Processor Configuration Example

In the following example, an event-based OIEP has been configured to deliver messages that inform a downstream system about changes to a product. A pre-processor business rule (business action) will be used to add referenced products to the products that are contained in the message.

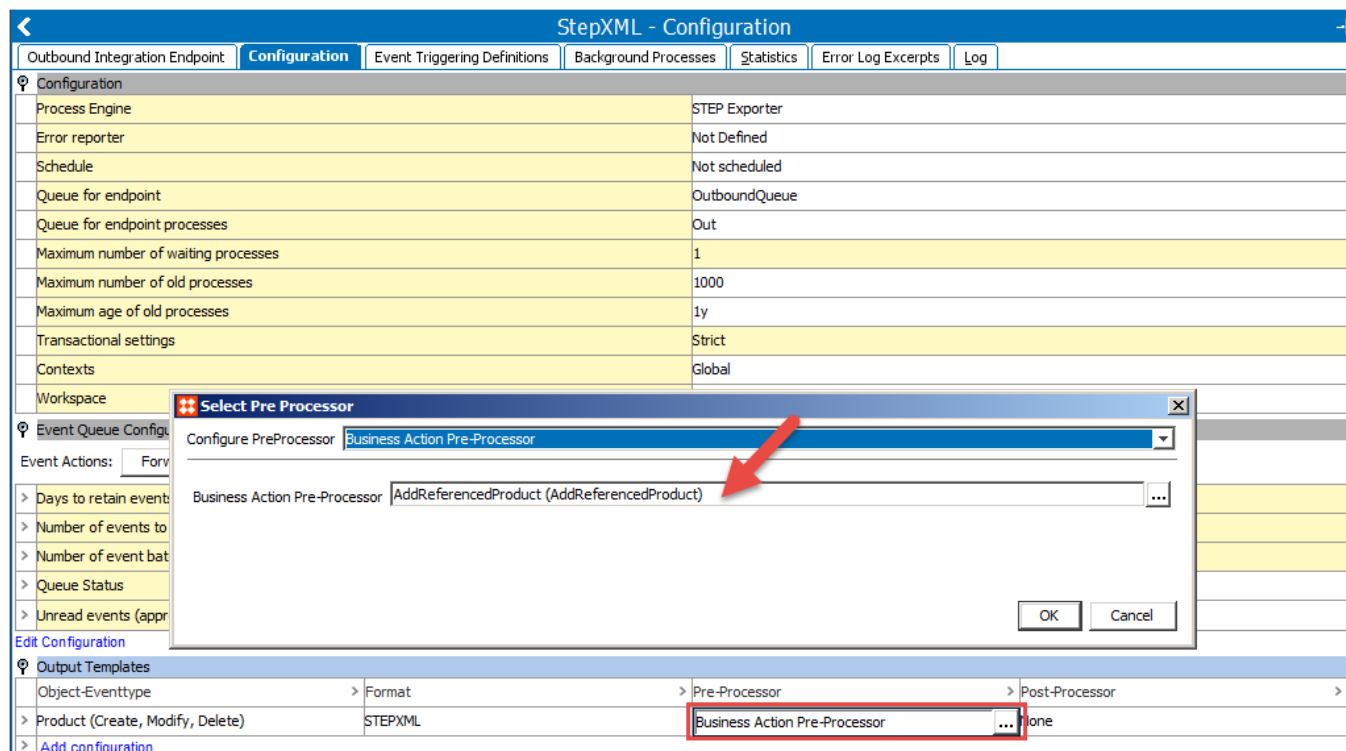
First, a business action is created where the current event batch is addressed to either add a node or to remove an event from the event batch. In this example, a node is added to each source product to include the target products that are referenced via the 'Accessory' Product Reference Type.



To make the business action valid, it must be made valid for 'All Object Types' or specifically for 'Event Queue Object Types':



A pre-processor can then be configured for any of the output templates:



1. In **System Setup**, locate the relevant outbound integration endpoint.
2. On the **Configuration** tab, navigate to the **Event Queue Configuration** flipper and locate the relevant output template. In the **Pre-Processor** column, click the ellipsis button (...), and then select the relevant pre-processor.
3. In **Configure PreProcessor**, the following options are available:
 - Select **No pre-processing** to export files without pre-processing.
 - Select **Business Action Pre-Processor** to execute a given business action on an event batch before export.
 - Select a pre-made business action to add a node or to remove an event from the event batch.

OIEP Post-Processor - Copy Context Dependent Values and References

The **Copy Context Dependent Values and References** post processor adds the <ContextID> and <QualifierID> XML tags to the export. A standard cross-context export 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 example illustrates the differences between using the Export Manager for a standard cross-context export 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 tables frêne page 282, 283.

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	abc 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, 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 cafes, bars, bistros and clubs.</Value>
6       <Value QualifierID="fre">Elégantes, modernes et légères, idéales pour les cafés, les bars, les bistros ou les discothèques.</Value>
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 - Copy Context Dependent Values and References

The following shows an XML output of an OIEP cross-context export 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 lightweight, ideal for use in cafes, bars, bistros and clubs.</Value>
7     <Value AttributeID="Body Copy" ContextID="FR All All" QualifierID="fre">Elégantes, modernes et légères, idéales pour les cafés, les bars, les bistros ou les discothèques.</Value>
8     <Value AttributeID="Body Copy" ContextID="DE All All" QualifierID="std.lang.all">Stylish, modern and lightweight, ideal for use in cafes, bars, bistros and clubs.</Value>
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 OIEP Delivery Method flipper includes the same parameters for both Event-Based and Select Objects endpoints.

Open the Delivery Method flipper to display the selected delivery method. The parameters vary based on the selected delivery method and each is described in **OIEP Delivery Methods**.

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	

Copy to Directory

For details on this delivery method, see **Copy to Directory Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
Configuration						
Object Selection Configuration						
Output Templates						
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**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
Configuration						
Object Selection Configuration						
Output Templates						
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**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
Configuration						
Object Selection Configuration						
Output Templates						
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**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
	Configuration					
	Object Selection Configuration					
	Output Templates					
	Delivery Method					
	FTP					
>	Hostname	FTPDeliveryHostServerName				
>	Username	OIEPUser				
>	Password	xxxxxxx				
>	File Name Template	\$filename-\$timestamp.\$extension				
>	Zip before upload	Yes				
>	Edit Delivery					

JMS

For details on this delivery method, see **JMS Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
	Configuration					
	Object Selection Configuration					
	Output Templates					
	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	xxxxxxxxx				
>	Additional Parameters					
>	Edit Delivery					

Mongo

For details on this delivery method, see **Mongo Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
- Configuration - Object Selection Configuration - Output Templates - Delivery Method						
Mongo delivery						
> Server						
> Port						
> Raw DB prefic						
> User name						
> Password						
> Use SSL to encrypt network traffic	No					
> Key store location						
> Trust store location						
> Actions						
> Edit Delivery						

Oracle AQ

For details on this delivery method, see **Oracle AQ Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
- Configuration - Object Selection Configuration - Output Templates - Delivery Method						
Oracle AQ Delivery						
> Connection URL	jdbc:oracle:thin:@//server:1521/step					
> Queue Name	OradeAQRecQueue					
> Binary Payload	No					
> User Name	OIEPUser					
> Password	xxxxxxx					
> Edit Delivery						

REST

For details on this delivery method, see **REST Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
- Configuration - Object Selection Configuration - Output Templates - Delivery Method						
REST						
> URL	http://www.Customer.com/sqlrest					
> User name	OIEUser					
> Password	xxxxxxxxxx					
> Zip content	Yes					
> Edit Delivery						

REST Direct

For details on this delivery method, see **REST Direct Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
- Configuration - Object Selection Configuration - Output Templates - Delivery Method						
REST Direct						
> URL	http://www.CustomerDirect.com/sqlrestdirect					
> User name	OIEUser					
> Password	xxxxxxxxxx					
> Zip content	Yes					
> Header						
> Footer						
> Edit Delivery						

SFTP

For details on this delivery method, see **SFTP Delivery Method**.

Outbound Integration Endpoint

Configuration

Background Processes

Statistics

Error Log Excerpts

Log

Status

Configuration

Object Selection Configuration

Output Templates

Delivery Method

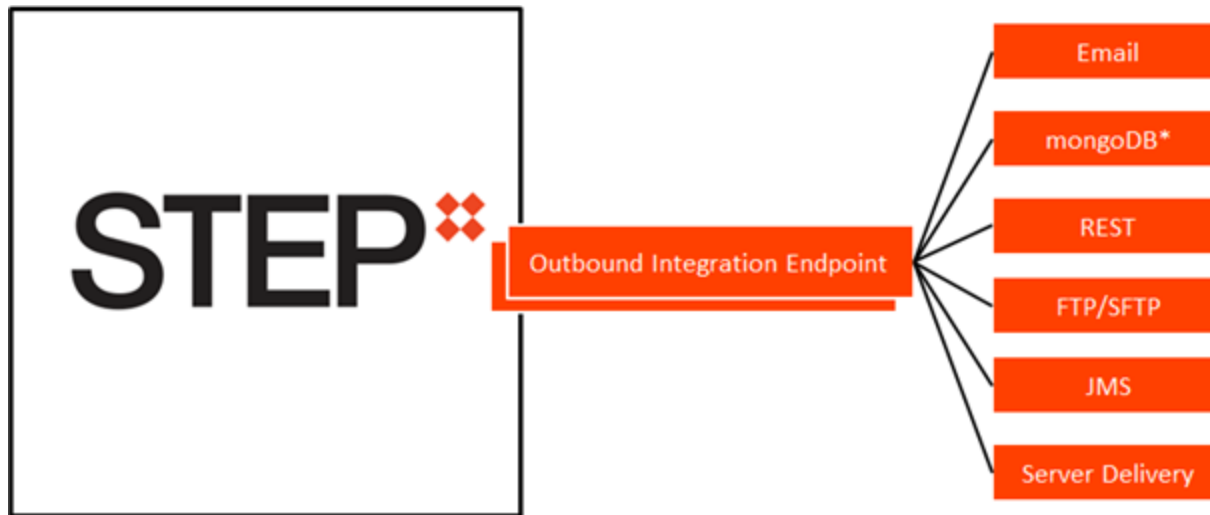
SFTP

> Hostname	SFTPDeliveryHostServerName
> Username	OIEPUser
> Password	xxxxxxxxxx
> File Name Template	\$filename-\$timestamp.\$extension
> Zip before upload	Yes
> Edit Delivery	

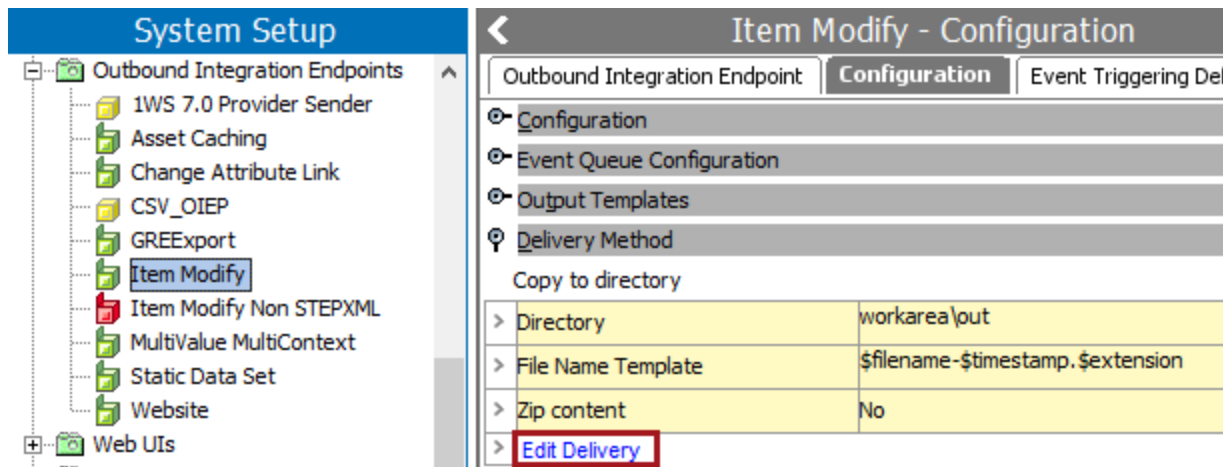
OIEP Delivery Methods

Delivery methods are used by both outbound data tools but the available options vary. For information on the delivery methods available in Export Manager, see **Export Manager Delivery Methods**.

An OIEP takes the data determined by its configuration and delivers it using the selected delivery method to the downstream system.



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 when exporting using an OIEP.

Method	Description
Copy to Directory	Delivers files to a directory on the application server and allows the delivery to be zipped. See Copy to Directory Delivery Method documentation.

Method	Description
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). You must specify the host name, user name, password, and the file name used to deliver the file. 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 About 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.
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 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). You must specify the host name, user name, password, and file name. See SFTP Delivery Method documentation.

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.

Note: The default delivery path is **opt/stibo/step** on the application server. This can be modified via the sharedconfig.properties file **DirectoryDeliveryPlugin.RootDir** property.

1. On the **Configuration** tab, navigate to the **Delivery Method** flipper, then click **Edit Delivery**.
2. In **Select Delivery Method**, choose **Copy to Directory**.

The screenshot shows a dialog box titled "Edit Delivery Configuration". It has a close button (X) in the top right corner. Inside the dialog, there is a section "Select Delivery Method" with a dropdown menu currently showing "Copy to directory". Below this, there are three fields: "Directory" (an empty text box), "File Name Template" (a text box containing "\$filename-\$timestamp.\$extension"), and "Zip content" (a dropdown menu currently showing "Yes"). At the bottom right of the dialog, there are two buttons: "OK" and "Cancel".

3. In **Directory**, specify the directory on the application server that will house the exported files. Text entered will be appended to the default path (opt/stibo/step) or to the updated path identified in the sharedconfig.properties file **DirectoryDeliveryPlugin.RootDir** property. Do not start this text with a slash (\).
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, '\$filename' 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.
 - **\$timestamp** Inserts 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, use \$timestamp(ddMMyyyy) to format the timestamp as a readable date. For example, 15112016 is used for an export on 15 NOV 2016.

- **\$extension** The extension of the selected output file format is inserted where \$extension is used.

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**, specify if the output file should be delivered in a .ZIP file format. 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'. When uncompressed, the file containing STEP data is named 'exported.xml'.

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.

Note: The default delivery path is **opt/stibo/step** on the application server. This can be modified via the sharedconfig.properties file **DirectoryDeliveryPlugin.RootDir** property.

1. On the **Configuration** tab, navigate to the **Delivery Method** flipper, then click **Edit Delivery**.
2. In **Select Delivery Method**, choose **Deploy**.

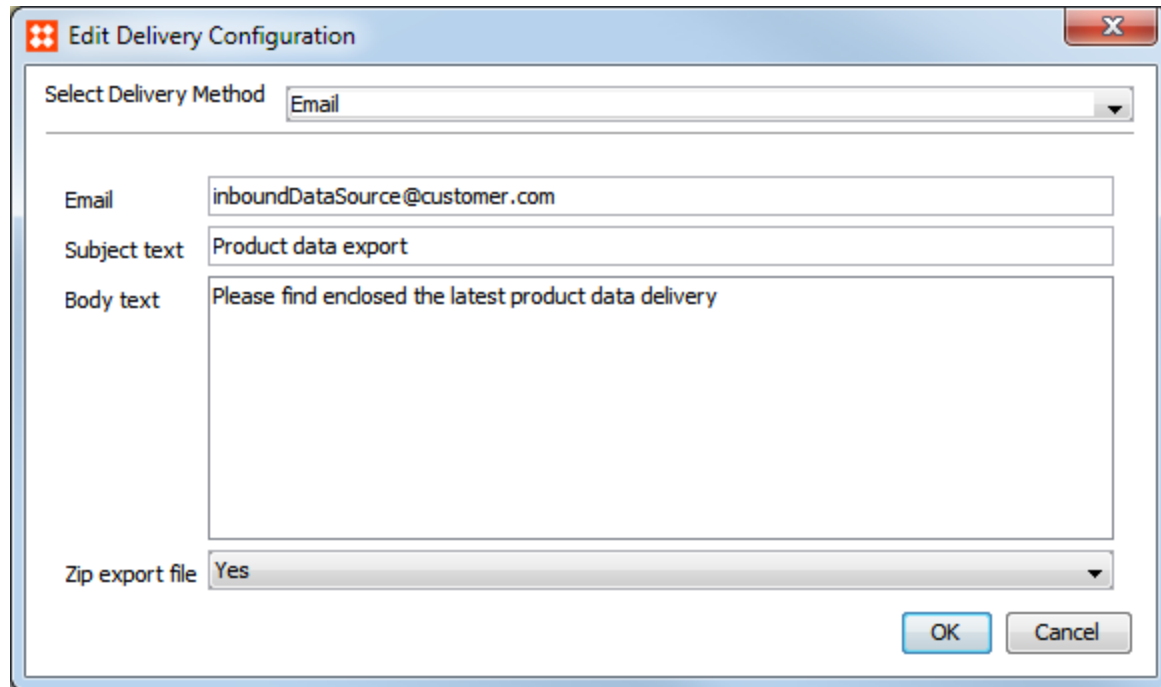
3. In **Extract to Directory**, specify a path on the application server where the exported .ZIP file should be unzipped. The field will only have an effect if the exported file is a .ZIP file, otherwise this field will be ignored.
4. In **Copy to file**, specify a file name template and a path where exported file should be saved.
 - **\$tmpdir** Points to the \$tmpdir path, defined in the configuration file of the server
 - **export-** Configurable text that can be replaced with any content that should be in the file name. For example, the Copy to File field could read '\$tmpdir/EmergencySKUExport-\$timestamp.\$extension' instead.
 - **\$timestamp** Inserts 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, use \$timestamp(ddMMyyyy) to format the timestamp as a readable date. For example, 15112016 is used for an export on 15 NOV 2016.
 - **\$extension** The extension of the selected output file format is inserted where \$extension is used.

Note: The 'Copy to file' 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.

Email Delivery Method

The email delivery option is very straight forward and, like 'Copy to directory', it allows the delivery to be zipped. This delivery method is available in Export Manager and in OIEPs.

1. In **Select Delivery Method**, choose **Email**.
2. Enter an email address, subject text, and body text.

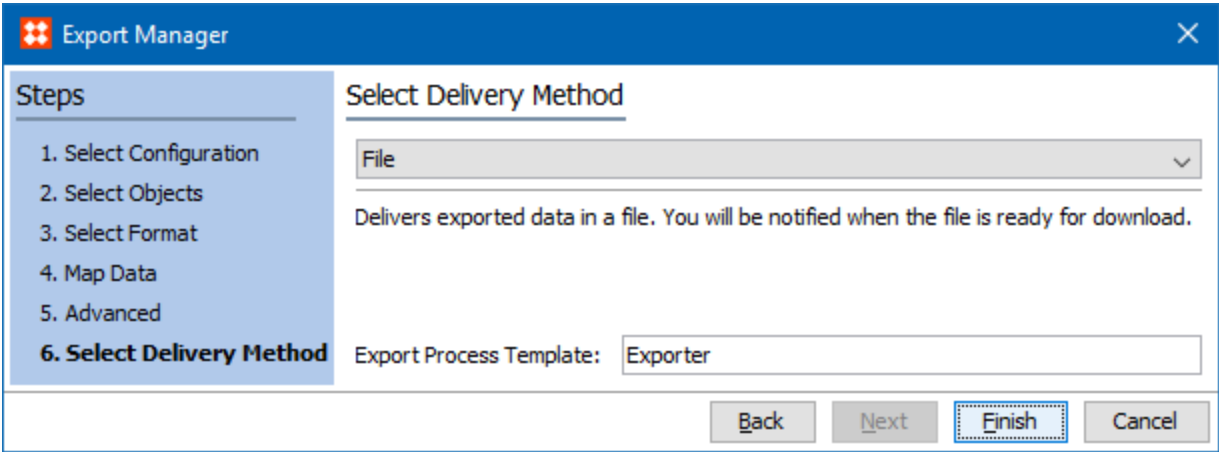


The screenshot shows a dialog box titled "Edit Delivery Configuration". At the top, there is a "Select Delivery Method" dropdown menu with "Email" selected. Below this, there are three input fields: "Email" with the value "inboundDataSource@customer.com", "Subject text" with the value "Product data export", 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 with "Yes" selected. The "OK" and "Cancel" buttons are located at the bottom right of the dialog box.

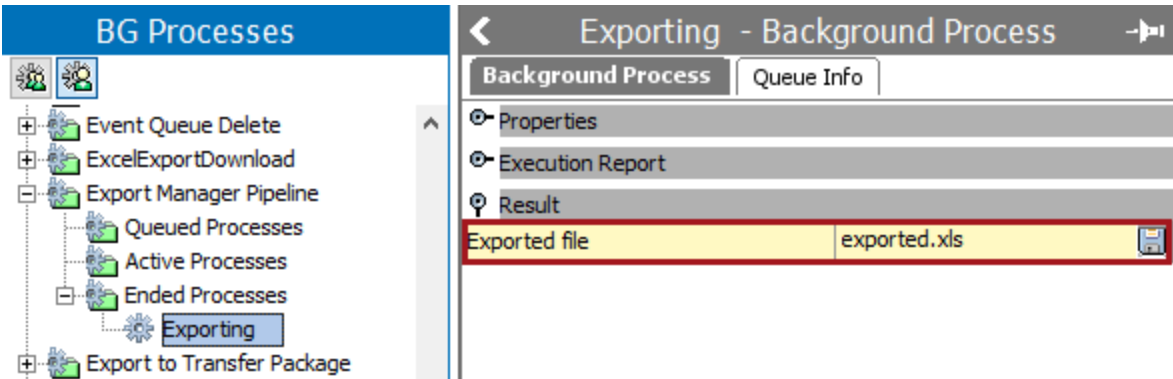
3. In **Zip export file**, specify if you want to deliver the exported contents (email attachment) as a .ZIP file.

File Delivery Method

The file delivery option is only available in Export Manager. It makes the output available within STEP, via a background process. There are no additional parameter settings.



By default, the output is saved on the Background Processes tab, under the Export Manager Pipeline node. While displaying the process, open the Result flipper on the Background Process tab.



FTP Delivery Method

The FTP delivery method allows an exported file to be delivered to an external system and is available in Export Manager and OIEP.

1. In **Select Delivery Method**, choose **FTP**.

The screenshot shows the 'Edit Delivery Configuration' window. At the top, there's a blue header with a Stibo logo and the title 'Edit Delivery Configuration'. Below the header, there's a 'Select Delivery Method' dropdown menu currently showing 'FTP'. Underneath, there are several input fields: 'Hostname' with a dropdown showing 'FTPHostName', 'Username' with a text box containing 'FTPUser', 'Password' with a masked text box (dots), 'File Name Template' with a text box containing '\$filename-\$timestamp.\$extension', and 'Zip before upload' with a dropdown showing 'Yes'. At the bottom right, there are 'OK' and 'Cancel' buttons.

2. In **Hostname**, select the host name or IP address of the FTP server to be used for the delivery. Options are available based on the FTPDeliveryHostName property in the sharedconfig.properties file on the application server. The required format of the property is (square brackets not included): FTPDeliveryHostName=1=[host1],2=[host2]. Using the hostname shown in the image above, the entry in the properties file would be: FTPDeliveryHostName=1=FTPDeliveryHostServerName. Contact Stibo Systems if you need assistance with configuration.
3. In **Username**, enter the user name that will be used to log on to the FTP server.
4. In **Password**, enter the password that will be used to log on to the FTP server.
5. 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 (e.g., 1438634758430). Using '\$timestamp(ddMMyyyy)' instead, formats the timestamp as a date. The placeholders for day, month, and year can be rearranged as needed to create the

desired date format. For example, a file generated on 29 March 2017 with the template '\$timestamp (ddMMyyyy).xls' is output as '29032017.xls' and the template '\$timestamp.xls' would use '1490810861593.xml' for the name.

- **\$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.

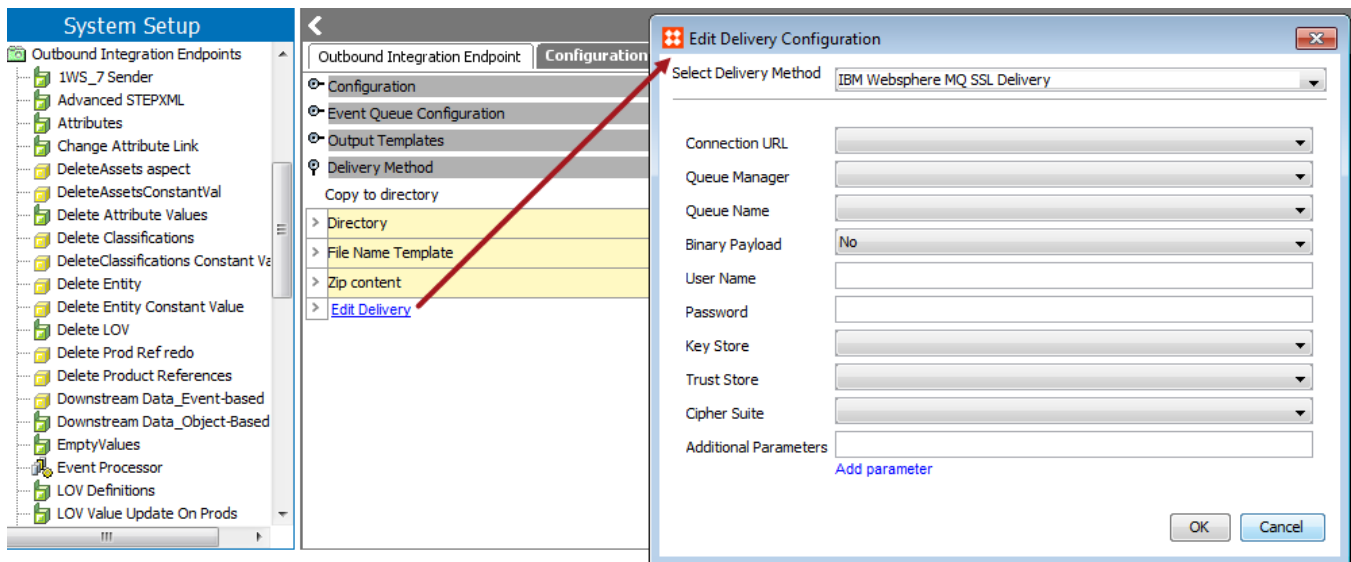
6. In **Zip before upload**, click **yes** to deliver a zipped file.

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.

Note: Options available on dropdown fields must be configured in sharedconfig.properties or config.properties. Select a dropdown to display the required key name for each field.

1. In the **Select Delivery Method** field, choose **IBM Websphere MQ SSL Delivery**.



2. In **Connection URL**, select the URL to connect to 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
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. To generate this, consult the manual for IBM Websphere MQ. The password for the key store must be configured in sharedconfig.properties or config.properties using 'WSMQSSLKeyStorePassword'.
9. In **Trust Store**, select the trust store with the CA for the Queue Manager. 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'.

10. In **Cipher Suite**, set to the same value as SSL CipherSuite in Websphere MQ. STEP is running on non-IBM jre, so this must be the same value as configured in the Queue Manager.
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.

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.

Note: Options available on dropdown fields must be configured in sharedconfig.properties using JMSConnectionFactoryName, JMSProviderURL, and JMSQueue. When no values are available, select a dropdown to display the required key name for each field. Contact Stibo Systems if you need assistance with configuration.

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. The list is determined by the JMSConnectionFactoryName property.
5. In **JMS Provider URL**, select a JMS Provider URL from the list. The list is determined by the JMSProviderURL property.
6. In **JMS Queue**, select the physical name of the JMS Queue to be used on Apache Active MQ or Websphere MQ. The list is determined by the JMSQueue property.
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.

Sharedconfig.properties File Examples

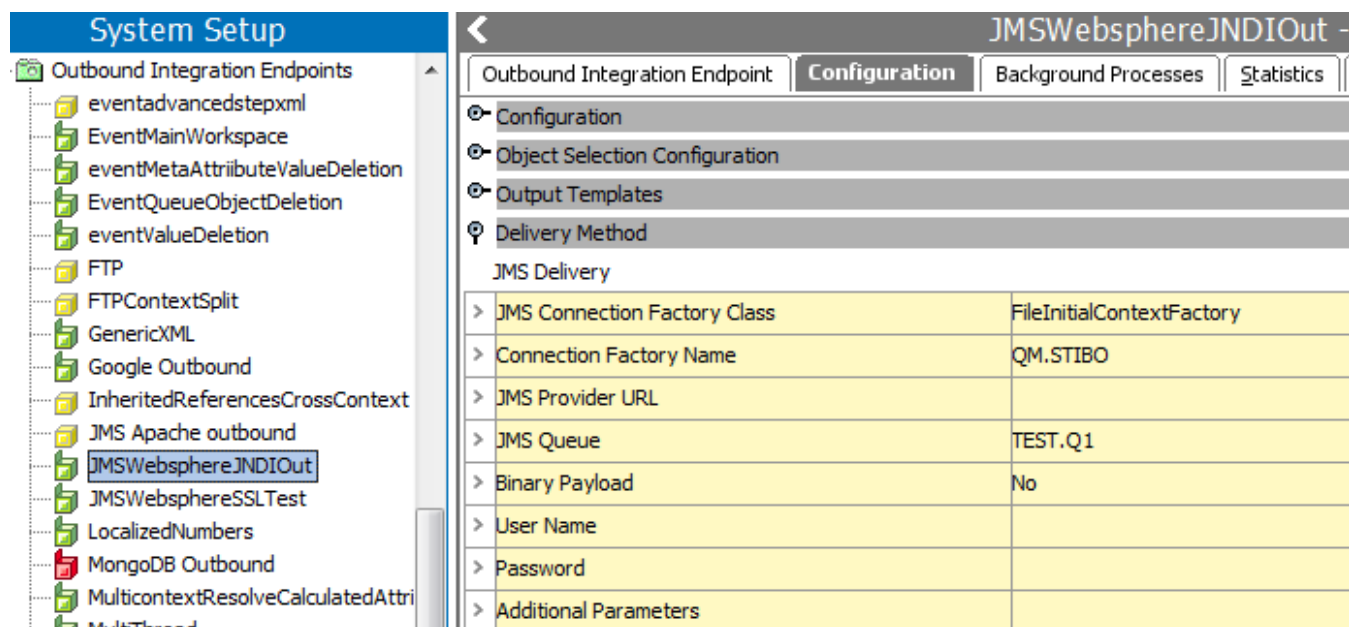
Each of these properties has a key/value pair as a value:

```
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=FirsProviderURL=https://myproviderurl.com/url, SecondProviderURL=https://mypr
oviderurl2.com/url
JMSQueue=Queue1=FirstqueueName, Queue2=SecondqueueName2
```

FileInitialContextFactory JMS Delivery Configuration Example



System Setup		JMSWebsphereJNDIOut -	
Outbound Integration Endpoints		Outbound Integration Endpoint	Configuration
eventadvancedstepxml			
EventMainWorkspace			
eventMetaAttributeValueDeletion			
EventQueueObjectDeletion			
eventValueDeletion			
FTP			
FTPContextSplit			
GenericXML			
Google Outbound			
InheritedReferencesCrossContext			
JMS Apache outbound			
JMSWebsphereJNDIOut			
JMSWebsphereSSLTest			
LocalizedNumbers			
MongoDB Outbound			
MulticontextResolveCalculatedAttri			
MultiThread			

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	websphermq-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 setting 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.

Note: Options available on dropdown fields must be configured in sharedconfig.properties or config.properties.

1. In **Select Delivery Method**, choose **Mongo delivery**.

The screenshot shows a dialog box titled "Edit Delivery Configuration". At the top, there is a dropdown menu labeled "Select Delivery Method" with "Mongo delivery" selected. Below this, there are several input fields: "Server" (a dropdown), "Port" (a dropdown), "Raw DB prefix" (a text box), "User name" (a text box), and "Password" (a text box). Below these, there is a checkbox labeled "Use SSL". To the right of the checkbox are two more dropdowns: "Key store location" and "Trust store location". At the bottom left, there is a section labeled "Actions" with a blue link "Add action". At the bottom right, there are "OK" and "Cancel" buttons.

2. In **Server**, enter the server where the MongoDB instance used by the endpoint is running. The available servers are configured using the MongoDB.server property.
3. In **Port**, enter the network port used by the MongoDB instance on the specified server. The possible values are configured using the MongoDB.port property.
4. In **Raw DB prefix**, if required, enter a prefix for all raw databases the adapter creates. Optional.
5. In **User name**, if MongoDB is configured to use authentication, enter the MongoDB user. Optional.
6. In **Password**, if MongoDB is configured to use authentication, enter the MongoDB password. Optional.
7. 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 in sharedconfig.properties or config.properties, using MongoDB.SSLKeyStorePassword. For example,

```
MongoDB.SSLKeyStorePassword = keystore_password
```

- 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 in sharedconfig.properties or config.properties, using MongoDB.SSLTrustStorePassword. For example,

```
MongoDB.SSLTrustStorePassword = truststore_password
```

8. 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.

For additional information about the Mongo Delivery Method, see the **Mongo Delivery Method Elements** documentation.

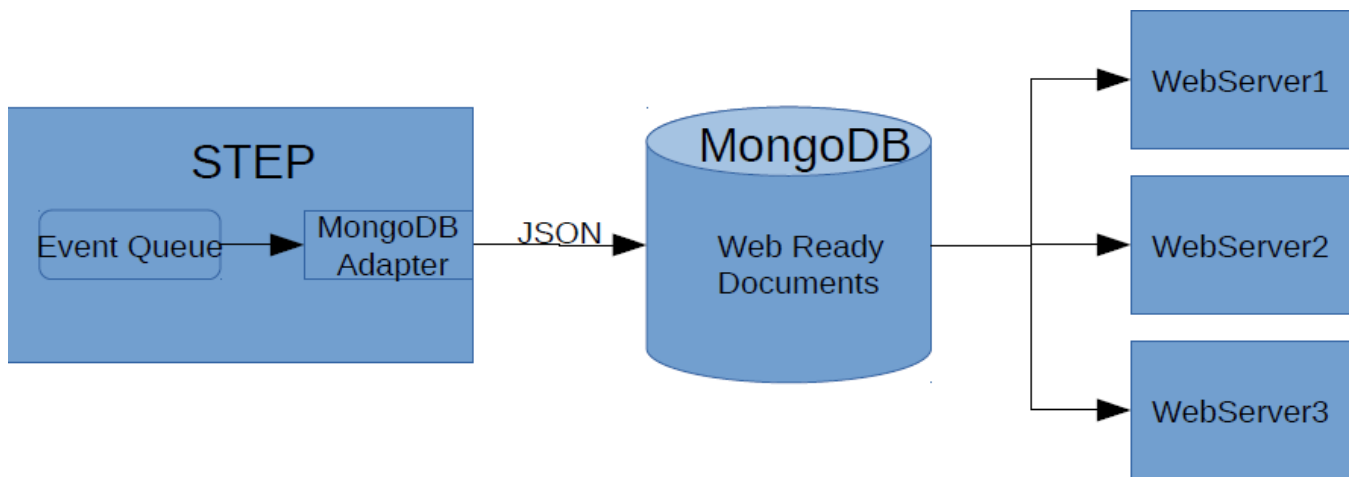
For more information, see **MongoDB Adapter Setup Quick Guides** documentation.

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.

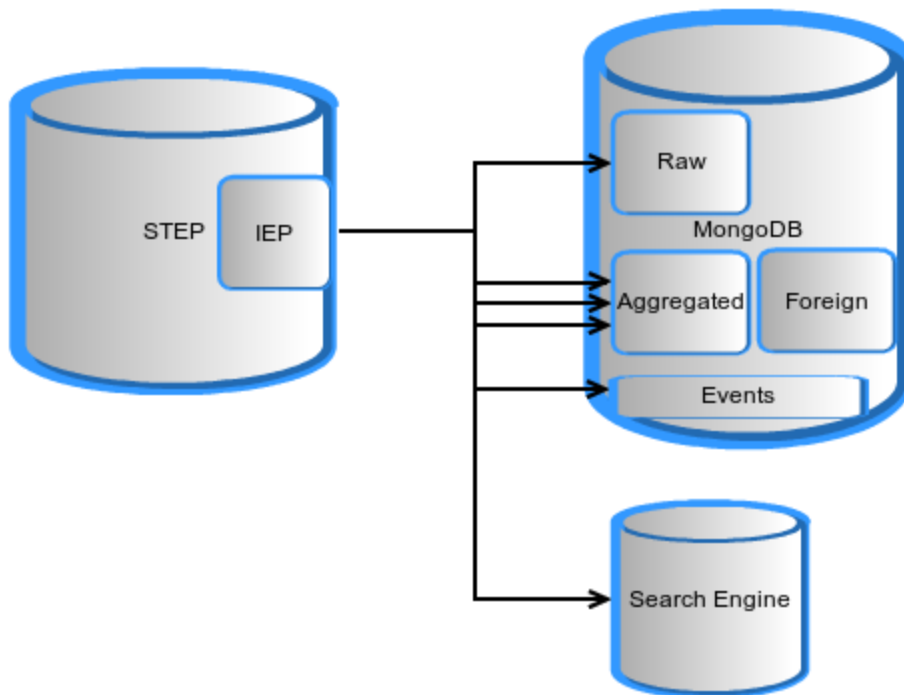
The Mongo adapter converts STEPXML to JSON and keeps a set of 'raw' collections in MongoDB that are in sync with STEP. Additionally, it allows you to add JavaScript triggers to maintain aggregated collections.

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:



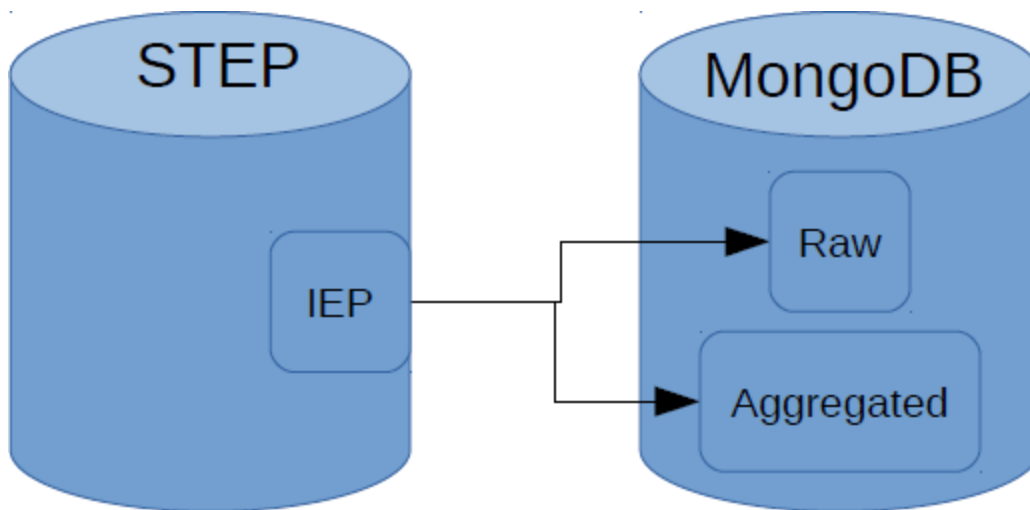
The STEP Mongo Adapter ensures that the raw collections are a one-to-one mirror of data in STEP. The adapter does not embed nodes found through references or child associations. These nodes are stored separately in the raw collections.

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 Collection

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 JavaScripts 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 **To Add Preconditions to a Business Rule** section of the **Editing Global Business Rules** 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 **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:

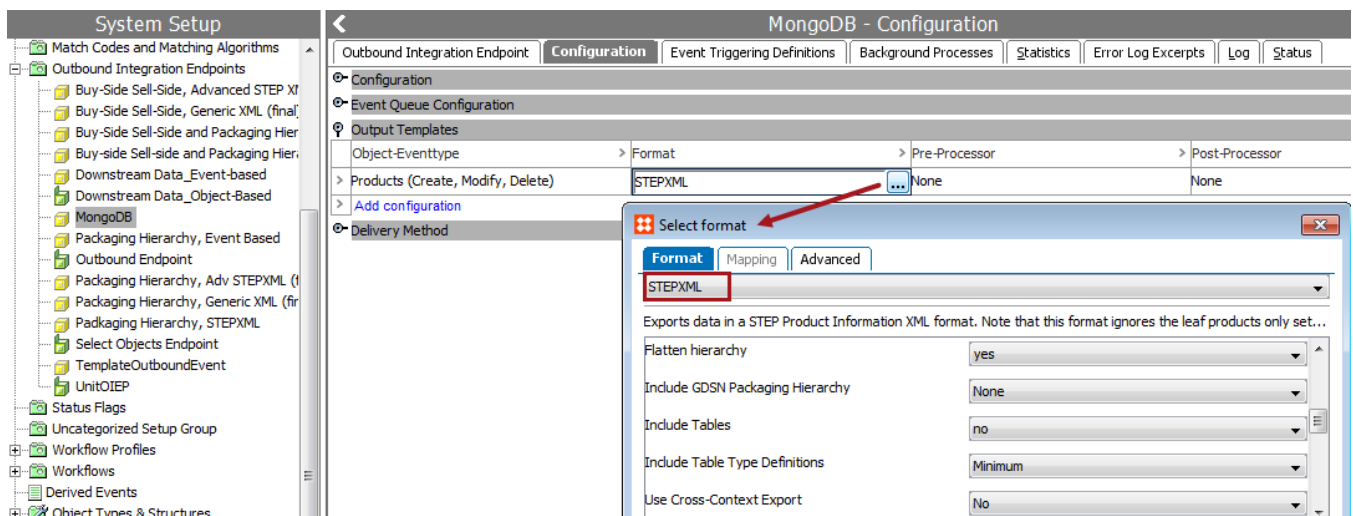
```
MongoDB.Server = 1=Server1, 2=Server2
```

```
MongoDB.Port = 1=27712
```

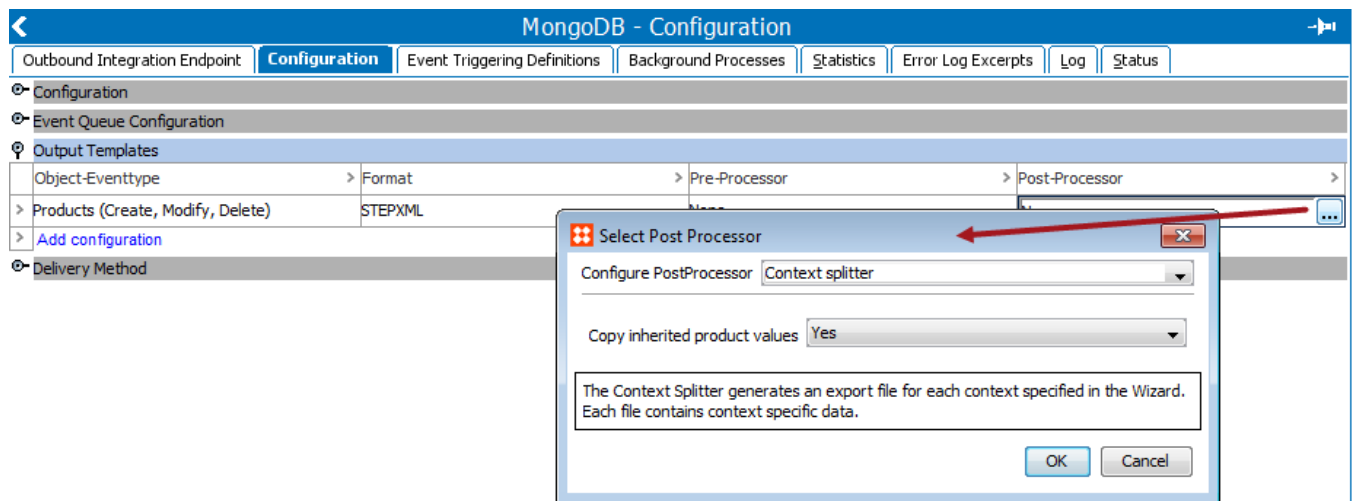
Configure the OIEP

The STEP MongoDB Adapter is run within an Outbound Integration Endpoint (OIEP) and configured as a delivery plugin for the endpoint.

1. Create an OIEP, selecting STEP Exporter as the Process Engine.
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 STEPXML export format in the dropdown at the top of the dialog
 - Set Flatten hierarchy = Yes
 - Set Include Type Definitions = Minimum
 - Set Include Unit Definitions = Minimum
 - Set Include List of Values Definitions = Minimum
 - Set Include Attribute Group Definitions = Minimum
 - Set Include Assets = Selected
 - Set Include Classifications = Selected
 - Set Include Products = Selected
 - Set Include Entities = Selected
4. In the Post-Processor field, click the ellipsis button (...) and select **Context splitter**, then click OK.



5. 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:

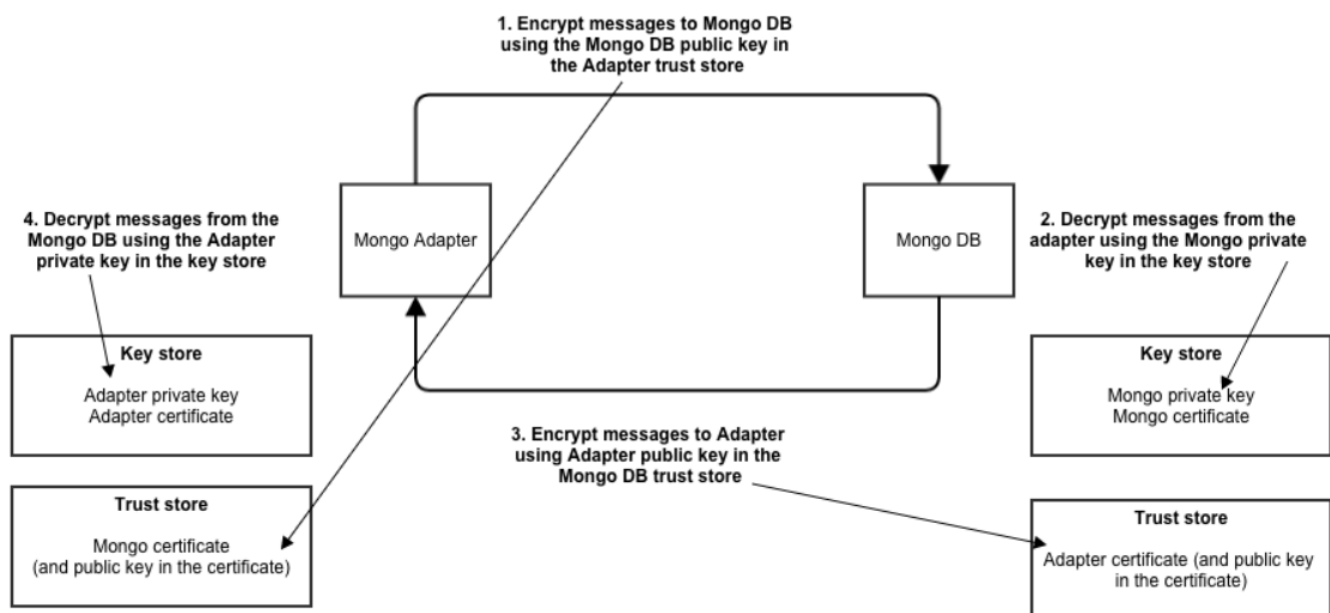
mongod --auth --config
```

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"
  }
}
```

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" } ]
  }
}

```

Here, 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 'Include Type Definitions' 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" : "centimetre"
  },
  "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 'Include Unit Definitions' 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"
}

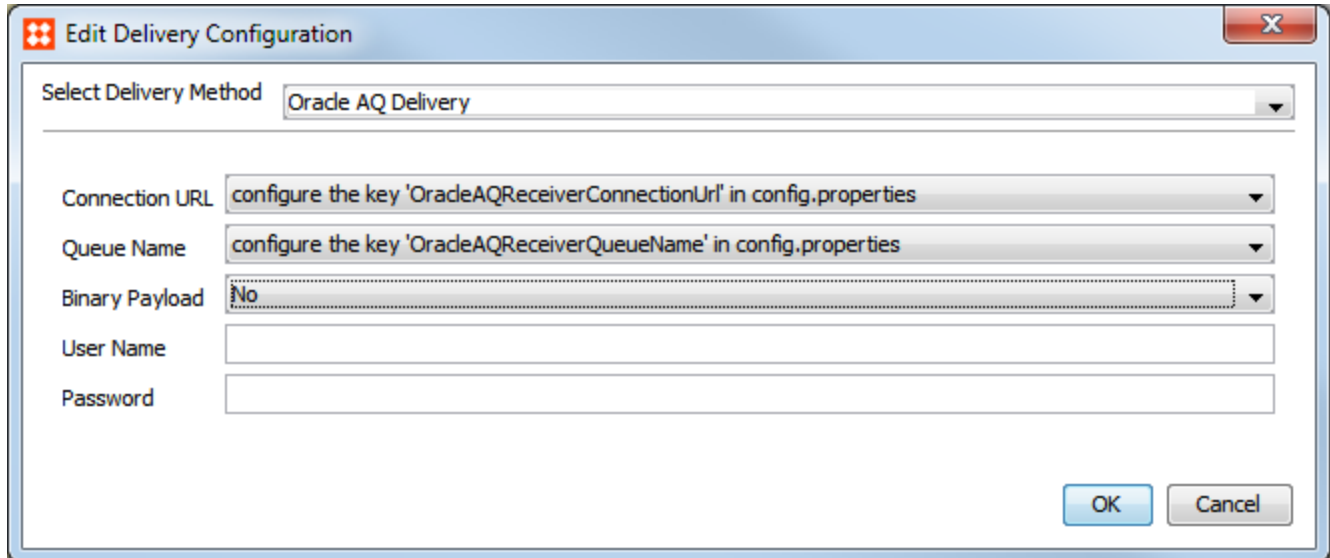
```

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.

1. In the **Select Delivery Method** list, choose **Oracle AQ Delivery**.



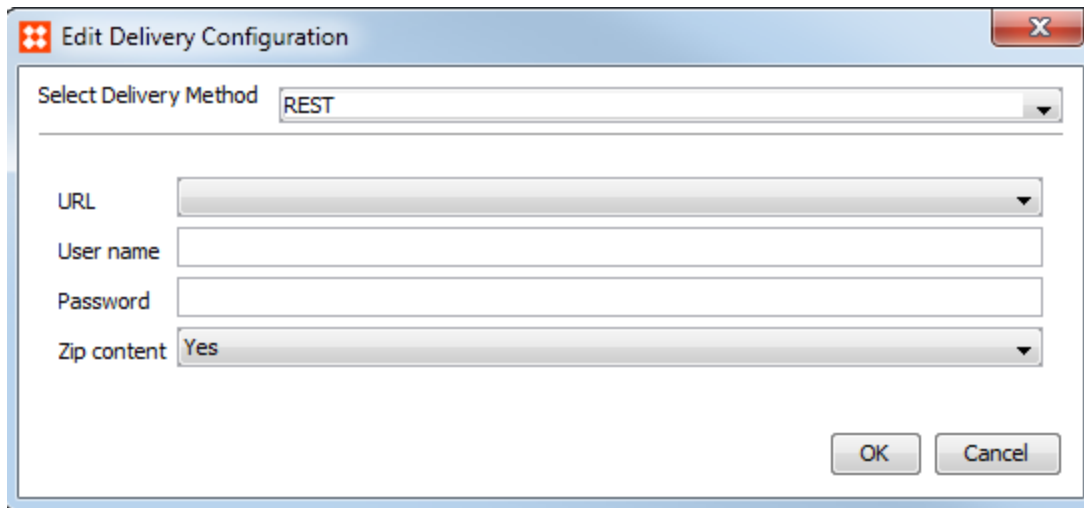
The screenshot shows a window titled "Edit Delivery Configuration". At the top, there is a "Select Delivery Method" dropdown menu with "Oracle AQ Delivery" selected. Below this, there are five configuration fields: "Connection URL" (a dropdown menu showing "configure the key 'OracleAQReceiverConnectionUrl' in config.properties"), "Queue Name" (a dropdown menu showing "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 of the window are "OK" and "Cancel" buttons.

2. In the **Connection URL** list, select a URL that points to Oracle AQ. Options are available based on the OracleAQReceiverConnectionURL property in the sharedconfig.properties file on the application server. Contact Stibo Systems if you need assistance with configuration.
3. In the **Queue name** list, select an Oracle AQ queue name. Options are available based on the OracleAQReceiverQueueName property in the sharedconfig.properties file on the application server. Contact Stibo Systems if you need assistance with configuration.
4. For **Binary Payload**, select Yes or No ('Yes' will be 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.

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.

1. For **Select Delivery Method**, choose **REST**.



The screenshot shows a dialog box titled "Edit Delivery Configuration". It has a close button (X) in the top right corner. Inside the dialog, there is a section labeled "Select Delivery Method" with a dropdown menu currently showing "REST". Below this, there are four input fields: "URL" (a text box), "User name" (a text box), "Password" (a text box), and "Zip content" (a dropdown menu currently showing "Yes"). At the bottom right of the dialog, there are two buttons: "OK" and "Cancel".

2. In **URL**, select the URL that points to the REST endpoint where you would like to receive the delivered data. 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.

Note: The available URLs are system dependent and must be configured in the `sharedconfig.properties` file under the property name `RESTDeliveryURL`. Contact Stibo Systems if you need assistance with configuration.

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.

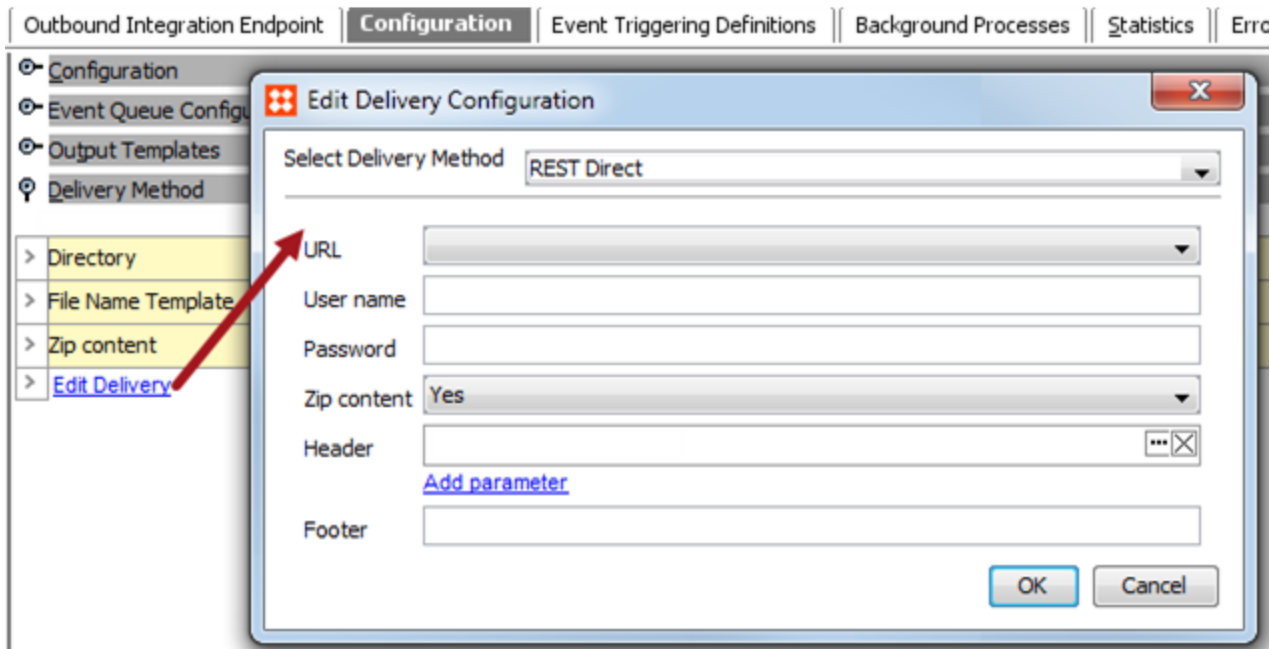
For information about how to use the REST API to upload files to REST, see the STEP API documentation.

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.

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. This delivery option is only available in OIEPs and if the delivery fails, the OIEP will be disabled. There is no special resilience handling.

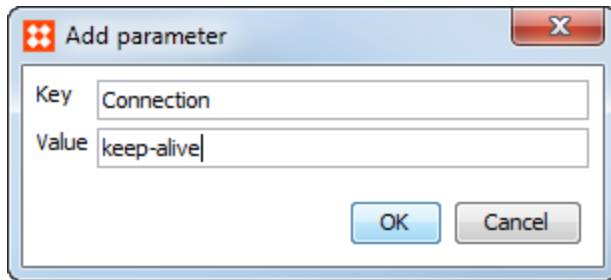
1. For **Select Delivery Method**, choose **REST Direct**.



2. In **URL**, select the URL that points to the REST endpoint where you would like to receive the delivered data.

Note: The available URLs are system dependent and must be configured in the sharedconfig.properties file under the property name RestDirectDeliveryURL. Contact Stibo Systems if you need assistance with configuration.

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. For **Header**, click the **Add Parameter** link and add a key and a value. This is part of the HTTP network protocol.



7. 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 pairs



Note: Once a header is displayed, use the ellipsis button (...) to edit it or the X button to remove it.

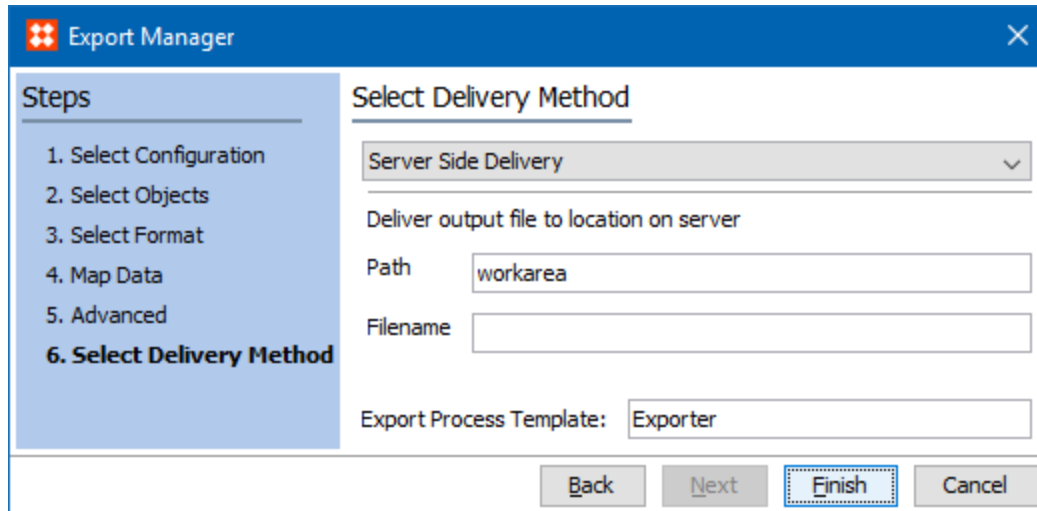
8. In **Footer**, add data required for the recipient to verify that the full message was received. This is part of the HTTP network protocol.

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.

9. On the Edit Delivery Configuration dialog, click the OK button to save the delivery method.

Server Side Delivery Method

The server side delivery option is only available in Export Manager. It makes the output available within STEP, via a background process.



Export Manager

Steps

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

Select Delivery Method

Server Side Delivery

Deliver output file to location on server

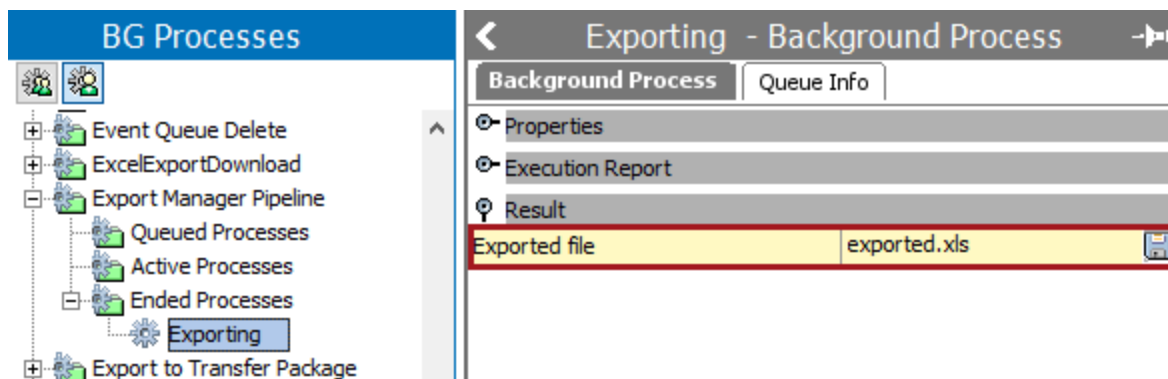
Path: workarea

Filename:

Export Process Template: Exporter

Back Next **Finish** Cancel

By default, the output is saved on the Background Processes tab, under the Export Manager Pipeline node. While displaying the process, open the Result flipper on the Background Process tab.



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

SFTP Delivery Method

This delivery method is available in Export Manager and in OIEPs.

1. In the Select Delivery Method list, select **SFTP** from the dropdown.

2. In **Hostname**, select the host name of the FTP server to be used for the delivery. Options are available based on the FTPDeliveryHostName property in the sharedconfig.properties file on the application server. The required format of the property is (square brackets not included): FTPDeliveryHostName=1=[host1],2=[host2]. Using the hostname shown in the image above, the entry in the properties file would be: FTPDeliveryHostName=1=FTPDeliveryHostServerName. Contact Stibo Systems if you need assistance with configuration.
3. In **Username**, enter the username that will be used to log on to the FTP server.
4. In **Password**, enter the password that will be used to log on to the FTP server.
5. 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 (e.g., 1438634758430). Using '\$timestamp(ddMMyyyy)' instead, formats the timestamp as a date. The placeholders for day, month, and year can be rearranged as needed to create the desired date format. For example, a file generated on 29 March 2017 with the template '\$timestamp

(ddMMyyyy).xls' is output as '29032017.xls' and the template \$timestamp.xls' would use '1490810861593.xml' for the name.

- **\$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.

6. In **Zip before upload**, click **yes** to deliver a zipped file.

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.

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

OIEP - Event-Based - Event Triggering Definitions Tab

When Event Queue has been selected as the data source in the Outbound Integration Endpoint wizard, you can specify if core events (modify, delete, or create events) for specific object types should be collected by the OIEP.

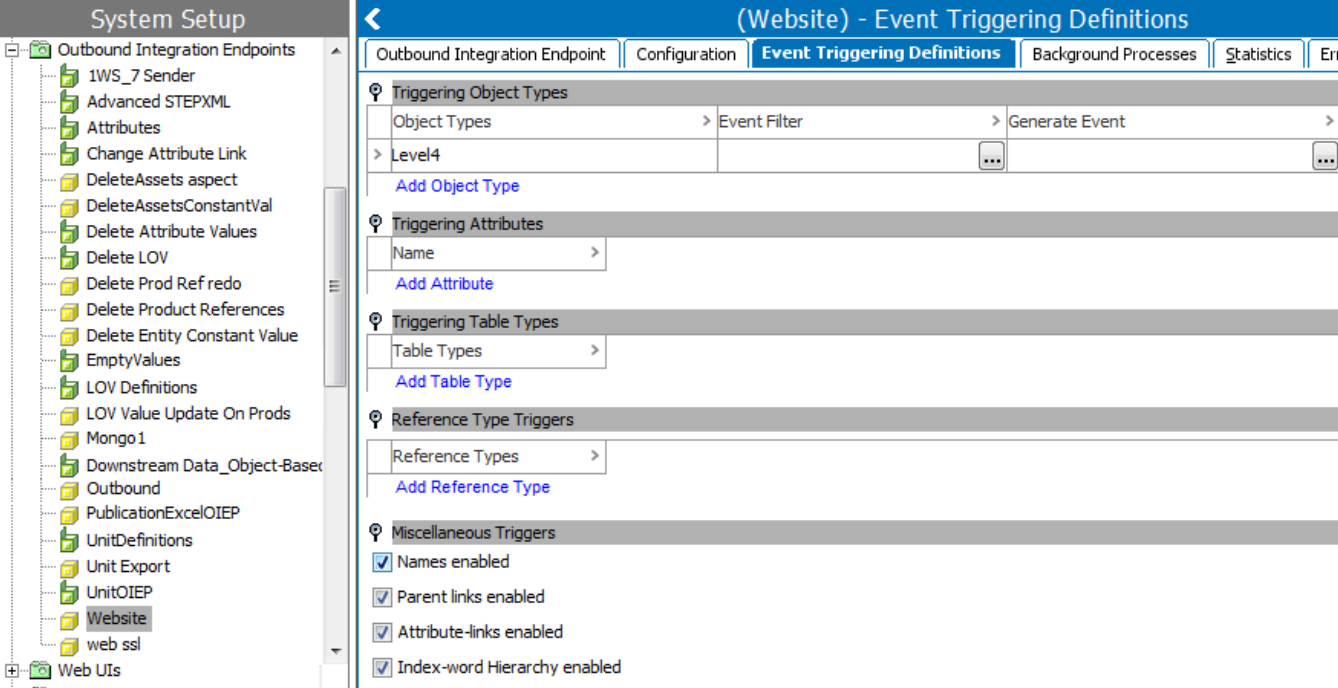
- Workspace revisable objects, such as a product, generate an event when data changes are approved.
- Global revisable objects, such as an attribute or an entity storing patient data, generate an event as soon as the data changes because approval is not available or required.

Event triggering definitions define what changes (events) are important to the OIEP. Changes to any element defined on this tab are then formatted based on the output template, and ultimately queued for export.

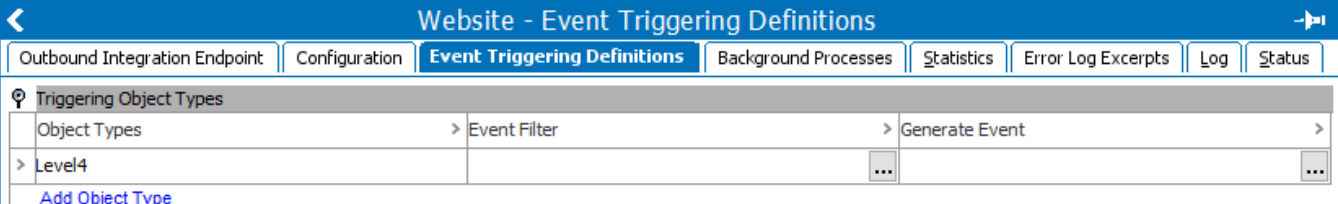
For details about event generation, see the **Events** section of the **System Setup / Super User Guide** documentation.

For event-based OIEPs, the **Event Triggering Definitions** tab allows you to add or edit all available triggers. No triggers are set by default, so all triggers that are required must be added manually.

Important: If no triggers are added, no events will be queued for the OIEP. Additionally, 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**.



Triggering Object Types Flipper



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 specific attribute is changed, there is no reason for the OIEP to listen to events happening on other attributes. Setting a trigger on the required attributes 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.

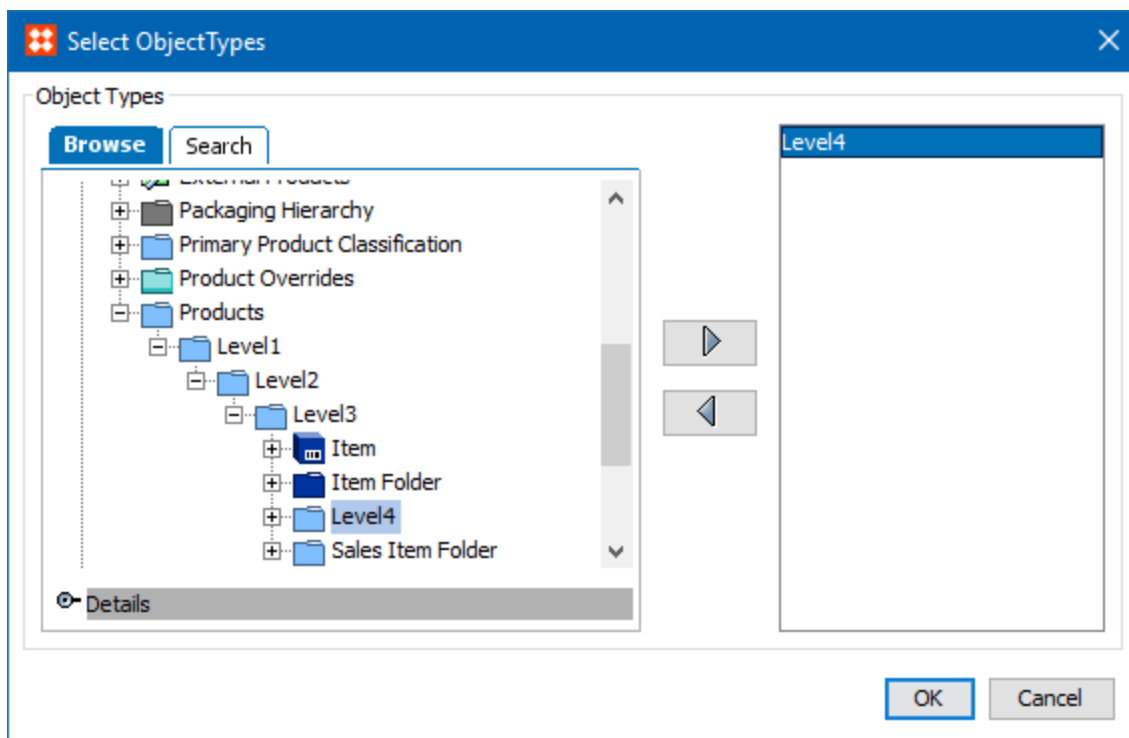
Object Types

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. If business rules are not used to filter and process events (functionality described below), all object types can be added in a single row.

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 / Super User Guide** documentation.

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 can be used in place of a pre-processor. 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 '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 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

Triggering Attributes are used to filter the data being monitored by the OIEP. You can set up an event-based OIEP so that it only registers events for objects of the object types selected in the Triggering Object Types section when values for specific attributes change. 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 here, 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 / Super User Guide** documentation.

- 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

The screenshot shows the 'Website - Event Triggering Definitions' interface. The 'Event Triggering Definitions' tab is active. Under the 'Triggering Table Types' section, there is a search bar with the text 'Table Types' and a right arrow. Below the search bar is a link labeled 'Add Table Type'.

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 System Setup configuration of the selected table types will trigger an event for the endpoint.

Reference Type Triggers Flipper

The screenshot shows the 'Website - Event Triggering Definitions' interface. The 'Event Triggering Definitions' tab is active. Under the 'Reference Type Triggers' section, there is a search bar with the text 'Reference Types' and a right arrow. Below the search bar is a link labeled 'Add Reference Type'.

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 here 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 / Super User Guide** documentation.

- Click **Add Reference Type**, then browse or search for the relevant reference type.

Miscellaneous Triggers 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

Triggering Table Types

Reference Type Triggers

Miscellaneous Triggers

☒ Names enabled

☒ Parent links enabled

☒ Attribute-links enabled

☒ Index-word Hierarchy enabled

The following miscellaneous triggers are available to 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.

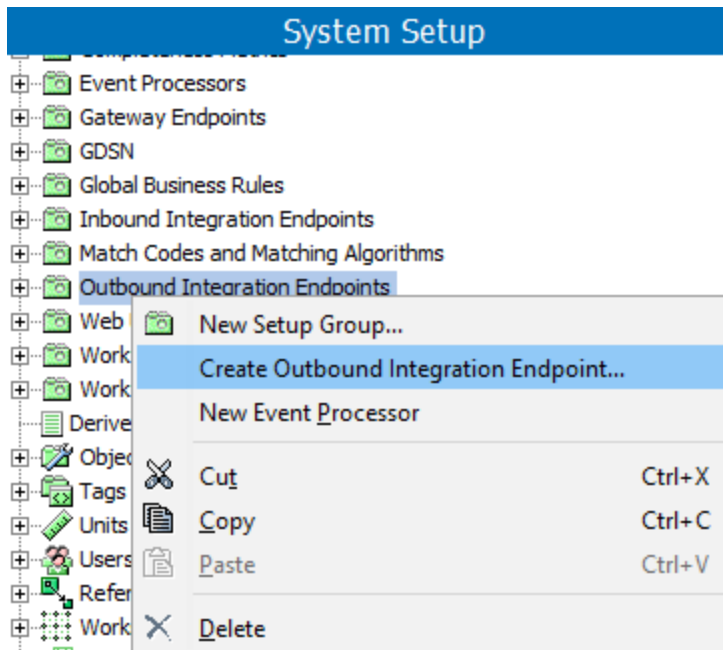
The screenshot shows the 'Outbound Integration Endpoint' configuration window. The 'Configuration' tab is active, showing the 'Event Queue Configuration' section. Below this, there are buttons for 'Event Actions': Forward, Rewind, Purge, and Republish. A table lists configuration parameters: 'Days to retain events' (0), 'Number of events to batch' (1000), 'Number of event batches to include per delivery' (1), 'Queue Status' (Read Events), and 'Unread events (approximated)' (Click to estimate ...). A red arrow points to the 'Edit Configuration' link. The 'Edit Configuration' dialog is open, showing the 'Choose Data Source' dropdown set to 'Event Queue Data Source'. It contains sections for 'Event Batching' (with radio buttons for 'No event batching' and 'Event batching' which is selected, with a text box for 'Specify maximum number of events to read for each generated message' set to 1000) and 'Bundle Messages' (with radio buttons for 'No message bundling' which is selected, and 'Bundle messages' which is not, with a text box for 'Specify maximum number of message results to bundle' set to 1). 'OK' and 'Cancel' buttons are at the bottom right.

- 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

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 the 'Outbound Integration Endpoint Wizard' dialog box. On the left, a 'Steps' pane lists three steps: '1. Identify Endpoint' (selected), '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. At the bottom, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

3. Complete setup require 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 'Identify Endpoint' step of the 'Outbound Integration Endpoint Wizard'. The left sidebar lists three steps: 1. Identify Endpoint (selected), 2. Choose Data Source, and 3. Configure Endpoint. The main area contains four input fields: 'Endpoint ID' with the value 'Outbound Endpoint', 'Endpoint Name' with the value 'Outbound Endpoint', 'Description' with the value 'Text', and 'User' with the value 'user (USER)' and a search icon. At the bottom are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

A Select Object OIEP allows one or more root node object hierarchies to be exported.

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**.
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. Common setup is to create a special system user for this purpose so that the effects of the endpoint are easily identified.
5. Click **Next** to display **OIEP - Select Objects - Choose Data Source**.

OIEP - Select Objects - Choose Data Source

The screenshot shows the 'Choose Data Source' step of the 'Outbound Integration Endpoint Wizard'. The left sidebar lists three steps: 1. Identify Endpoint, 2. Choose Data Source (selected), and 3. Configure Endpoint. The main area features a 'Choose Data Source' dropdown menu with 'Select Objects' selected. Below it is a table with columns: ID, Name, Object Type, and Path. The table contains one row with the following data: ID: 20688, Name: Footwear, Object Type: Level1, Path: Primary Product Hierarchy... There is also a link '> Add Objects' below the table. At the bottom are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

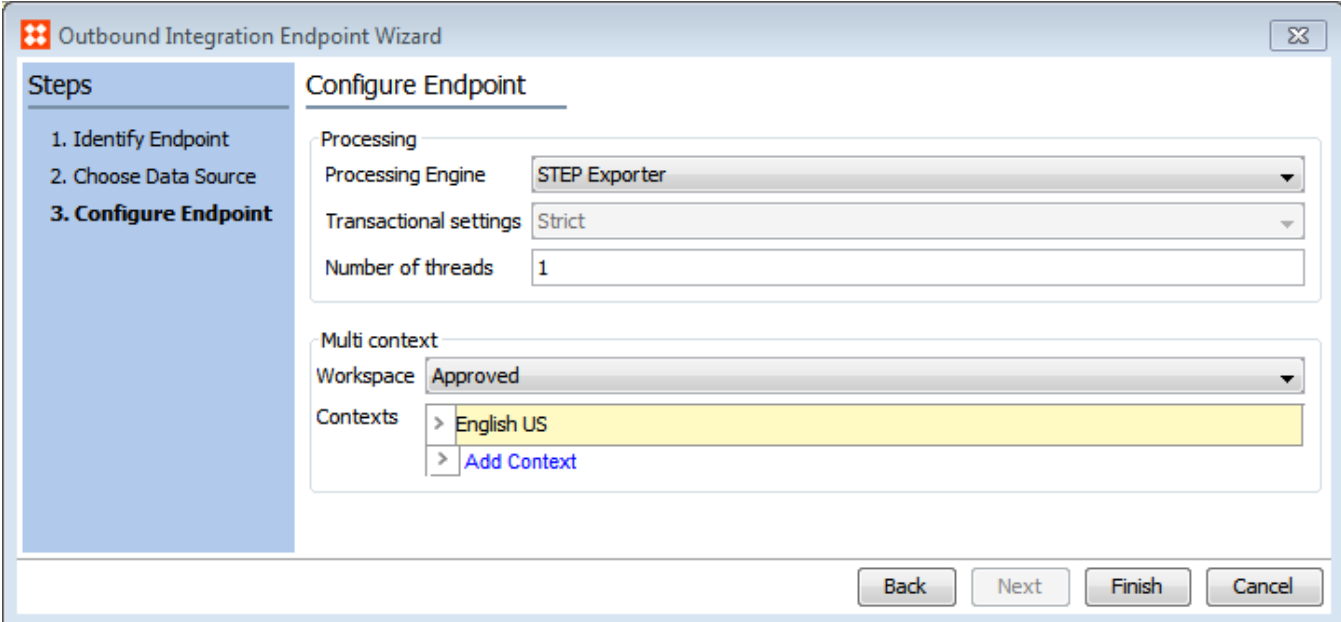
ID	Name	Object Type	Path
> 20688	Footwear	Level1	Primary Product Hierarchy...

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 **Add Objects** to select one or more root node object hierarchies to be exported. Selecting an object hierarchy is not required if the endpoint will export system-specific data, such as attributes, LOVs, and units that are not used in any specific product, classification, or entity hierarchy.
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 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 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:

- 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.
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.

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
Description			Statistics	Error Log Excerpts
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			Log	Status

OIEP - Configuration Flipper

The OIEP Configuration flipper includes the same parameters for both Event-Based and Select Objects endpoints. Each of the parameters are described below.

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

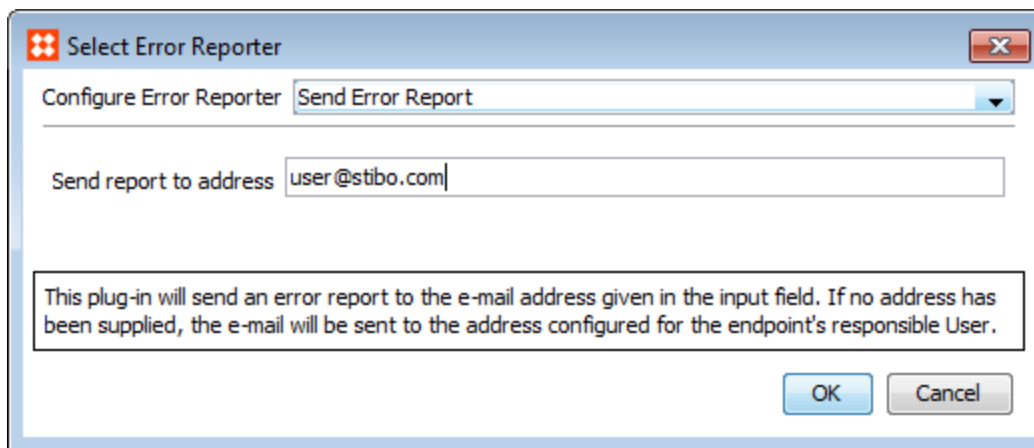
The processing engine is originally selected in the wizard when configuring the endpoint. For more information, see **OIEP - Event-Based - Configure Endpoint** documentation or **OIEP - Select Objects - Configure Endpoint** documentation.

Error Reporter

The 'Error reporter' parameter is not included in the wizard, but it can be configured to send an email error report 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 customizations) includes only the 'Send Error Report' option.

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 the email address of the error report recipient.

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.

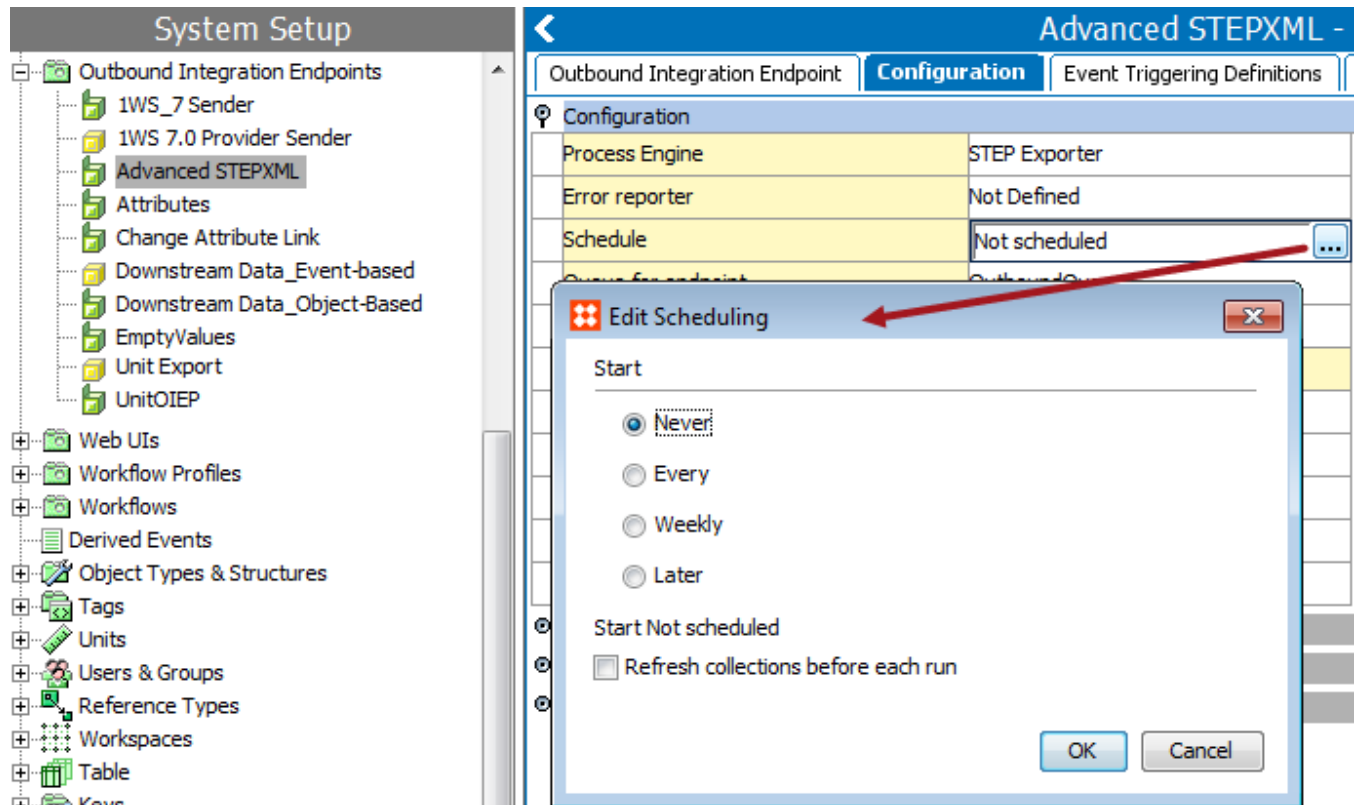
Schedule

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** if you want to invoke the endpoint manually.
- **Every, Weekly, or Later** to specify how often you want the endpoint to run. Different options are available depending on the frequency you select.



3. 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 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.

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.

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.

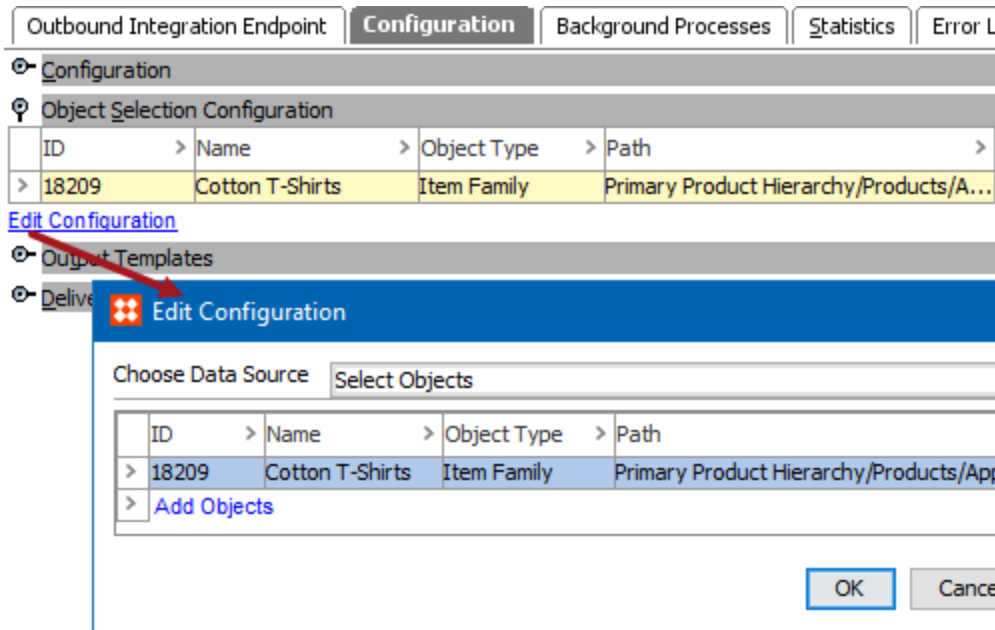
Workspace

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.

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**.

1. Click the Edit Configuration link to open the Edit Configuration dialog.

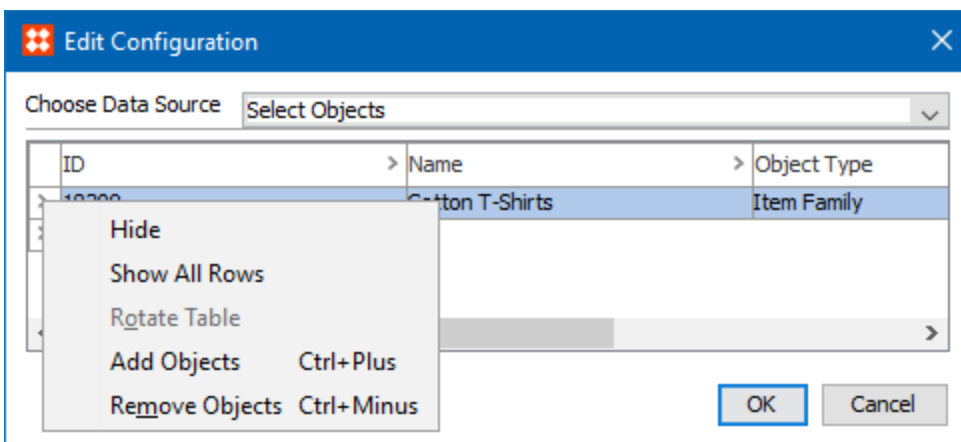


- Select an existing row and right-click to display additional options.

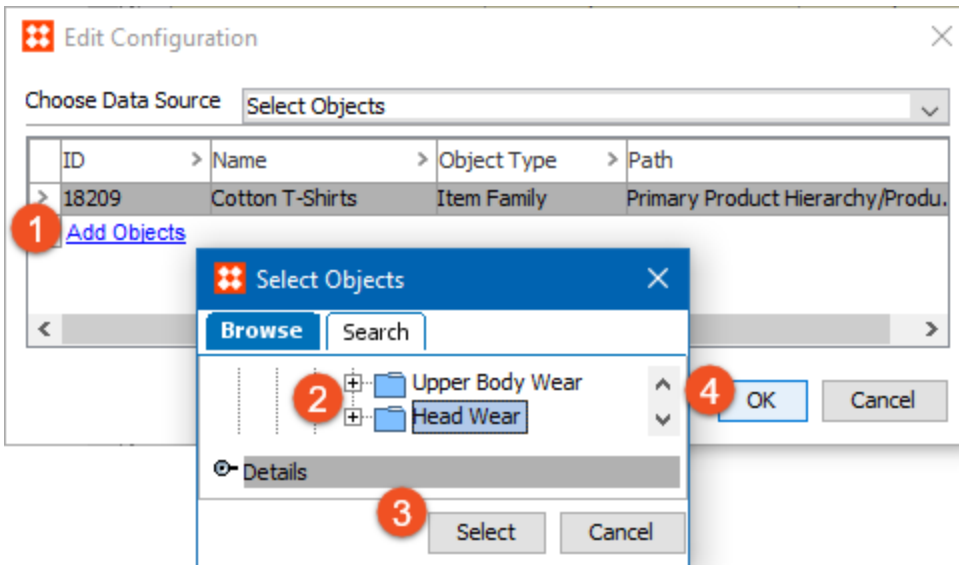
Hide and **Remove Objects** only affect the selected row.

Show All Rows and **Rotate Table** affect all rows.

Add Objects is the same option as accessed with the Add Objects link discussed below.



- 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 enables you to use one output format for a specific object type, and another format for a different object type.

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.

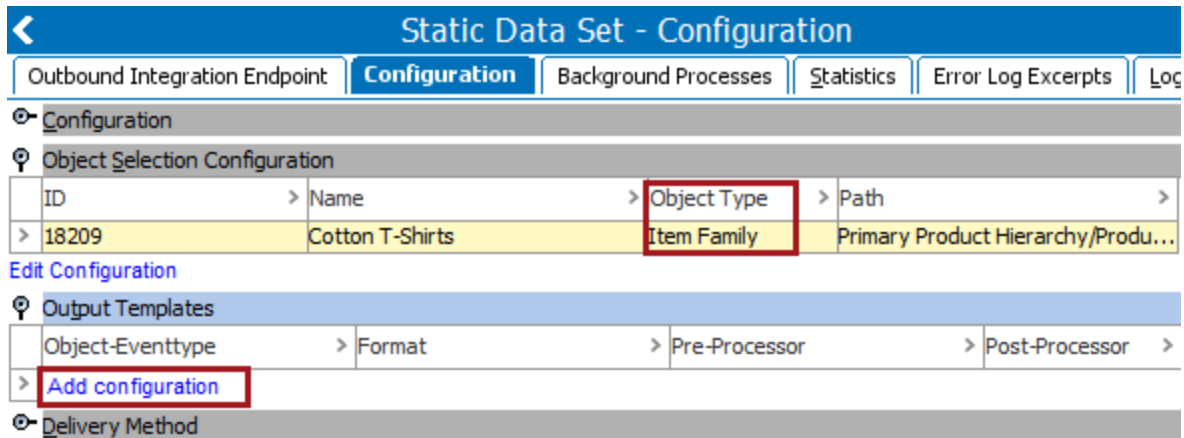
Configuring a select objects output template involves the following steps:

- Configure the object types
- Specify the format
- Configure pre-processor and post-processor, based on availability on your system.

Configure the Object Types

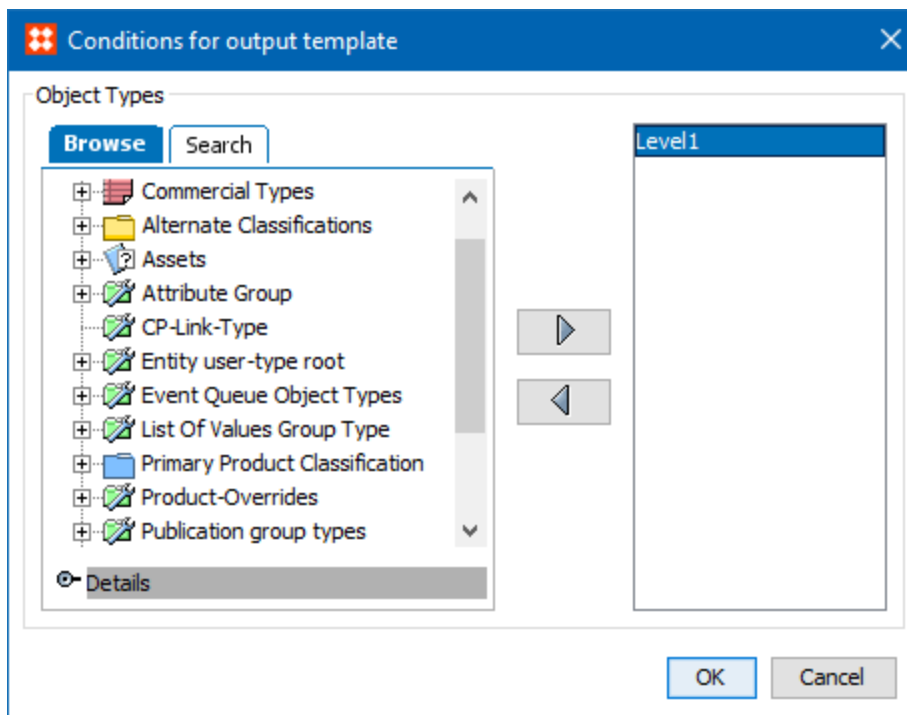
The object type selection determines the STEP data that will be exported.

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.



- In the 'Conditions for output template' dialog, select object types here 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, use Search to locate the relevant collection.

Note: Selecting an 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.



- Select Objects endpoints can have only one template per object type. Repeat the process 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	Post-Processor	
> Product Category (Create, Modify, Delete)	Generic XML (1 mappings) ...	None	
> Product Family (Create, Modify, Delete)	Generic XML (1 mappings)	None	
> Add configuration			

- 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. 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.

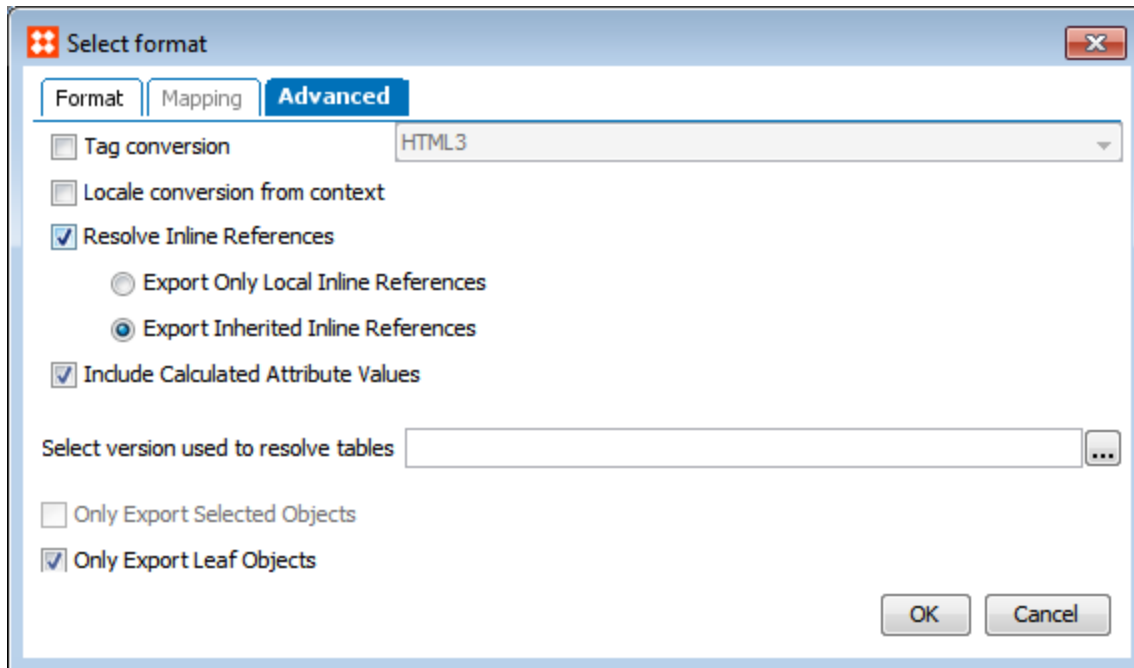
Include Validation	None
Include Type Definitions	None
Include Global Settings	None
Include Context Definitions	None

OK Cancel

- Depending on the format selected, the **Mapping** tab may be enabled. When available, specify how to map data to the export format. Mappings for each available format are described within the **Data Formats** documentation. For more information, see **Export Manager - Map Data** documentation.

If the selected format is STEPXML or Advanced STEPXML, the Mapping tab is disabled because these formats are automatically mapped.

- 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. **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 / STEP User Guide**.
- Include Calculated Attribute Values:** Resolves calculated attribute values when they are exported. If not selected, calculated attribute values are exported with empty values. 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. Specifies which version of a given publication should be used to resolve tables. If a publication is not selected, the version last used in STEP workbench is used.
- **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, pre- and post-processor options may be available.

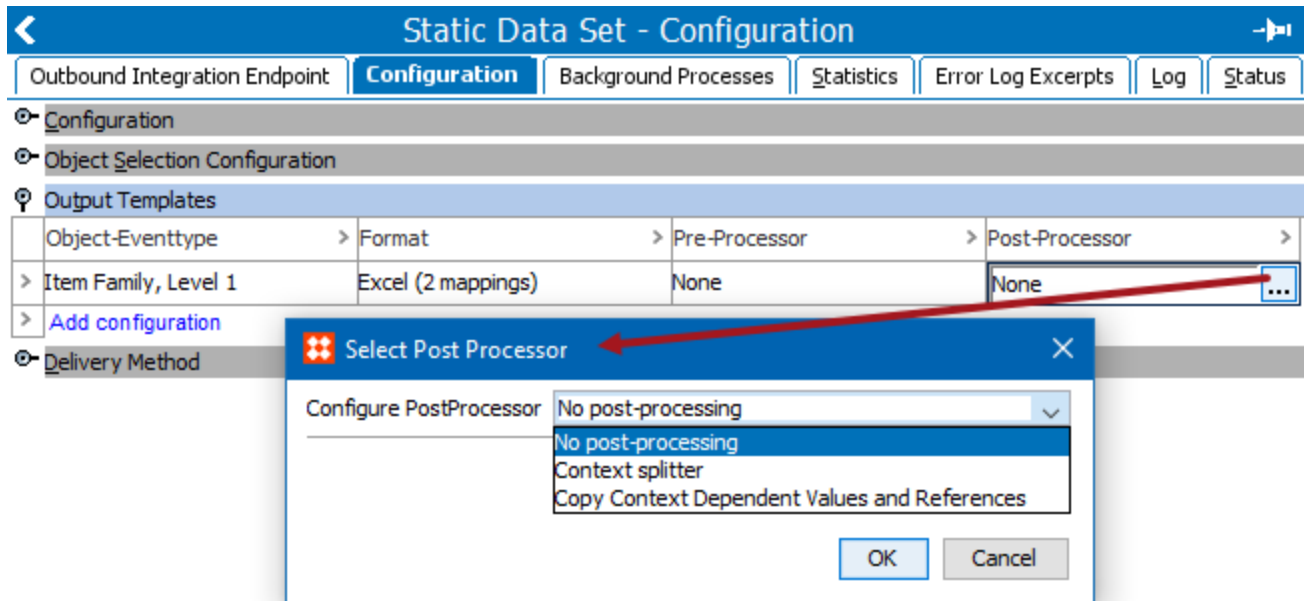
Pre-processor

Although a business action pre-processor can be set up for a select objects endpoint, it is not valid for Select Objects endpoints and will not cause any changes to the data being processed.

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.

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.

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 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 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 ContextIDs to a cross-context export, 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 Post-Processor - Copy Context Dependent Values and References

The **Copy Context Dependent Values and References** post processor adds the <ContextID> and <QualifierID> XML tags to the export. A standard cross-context export 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 example illustrates the differences between using the Export Manager for a standard cross-context export 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 tables frêne page 282, 283.

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	abc 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, 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 cafes, bars, bistros and clubs.</Value>
6       <Value QualifierID="fre">Elégantes, modernes et légères, idéales pour les cafés, les bars, les bistros ou les discothèques.</Value>
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 - Copy Context Dependent Values and References

The following shows an XML output of an OIEP cross-context export 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 lightweight, ideal for use in cafes, bars, bistros and clubs.</Value>
7     <Value AttributeID="Body Copy" ContextID="FR All All" QualifierID="fre">Elégantes, modernes et légères, idéales pour les cafés, les bars, les bistros ou les discothèques.</Value>
8     <Value AttributeID="Body Copy" ContextID="DE All All" QualifierID="std.lang.all">Stylish, modern and lightweight, ideal for use in cafes, bars, bistros and clubs.</Value>
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 OIEP Delivery Method flipper includes the same parameters for both Event-Based and Select Objects endpoints.

Open the Delivery Method flipper to display the selected delivery method. The parameters vary based on the selected delivery method and each is described in **OIEP Delivery Methods**.

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	

Copy to Directory

For details on this delivery method, see **Copy to Directory Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
Configuration						
Object Selection Configuration						
Output Templates						
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**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
Configuration						
Object Selection Configuration						
Output Templates						
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**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
Configuration						
Object Selection Configuration						
Output Templates						
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**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
Configuration						
Object Selection Configuration						
Output Templates						
Delivery Method						
FTP						
> Hostname	FTPDeliveryHostServerName					
> Username	OIEPUser					
> Password	xxxxxxx					
> File Name Template	\$filename-\$timestamp.\$extension					
> Zip before upload	Yes					
> Edit Delivery						

JMS

For details on this delivery method, see **JMS Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
Configuration						
Object Selection Configuration						
Output Templates						
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	xxxxxxxxx					
> Additional Parameters						
> Edit Delivery						

Mongo

For details on this delivery method, see **Mongo Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
Configuration Object Selection Configuration Output Templates Delivery Method						
Mongo delivery						
> Server						
> Port						
> Raw DB prefic						
> User name						
> Password						
> Use SSL to encrypt network traffic		No				
> Key store location						
> Trust store location						
> Actions						
> Edit Delivery						

Oracle AQ

For details on this delivery method, see **Oracle AQ Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
Configuration Object Selection Configuration Output Templates Delivery Method						
Oracle AQ Delivery						
> Connection URL		jdbc:oracle:thin:@//server:1521/step				
> Queue Name		OradeAQRecQueue				
> Binary Payload		No				
> User Name		OIEPUser				
> Password		xxxxxxxx				
> Edit Delivery						

REST

For details on this delivery method, see **REST Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
- Configuration - Object Selection Configuration - Output Templates - Delivery Method						
REST						
> URL	http://www.Customer.com/sqlrest					
> User name	OIEUser					
> Password	xxxxxxxxxx					
> Zip content	Yes					
> Edit Delivery						

REST Direct

For details on this delivery method, see **REST Direct Delivery Method**.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
- Configuration - Object Selection Configuration - Output Templates - Delivery Method						
REST Direct						
> URL	http://www.CustomerDirect.com/sqlrestdirect					
> User name	OIEUser					
> Password	xxxxxxxxxx					
> Zip content	Yes					
> Header						
> Footer						
> Edit Delivery						

SFTP

For details on this delivery method, see **SFTP Delivery Method**.

Outbound Integration Endpoint

Configuration

Background Processes

Statistics

Error Log Excerpts

Log

Status

Configuration

Object Selection Configuration

Output Templates

Delivery Method

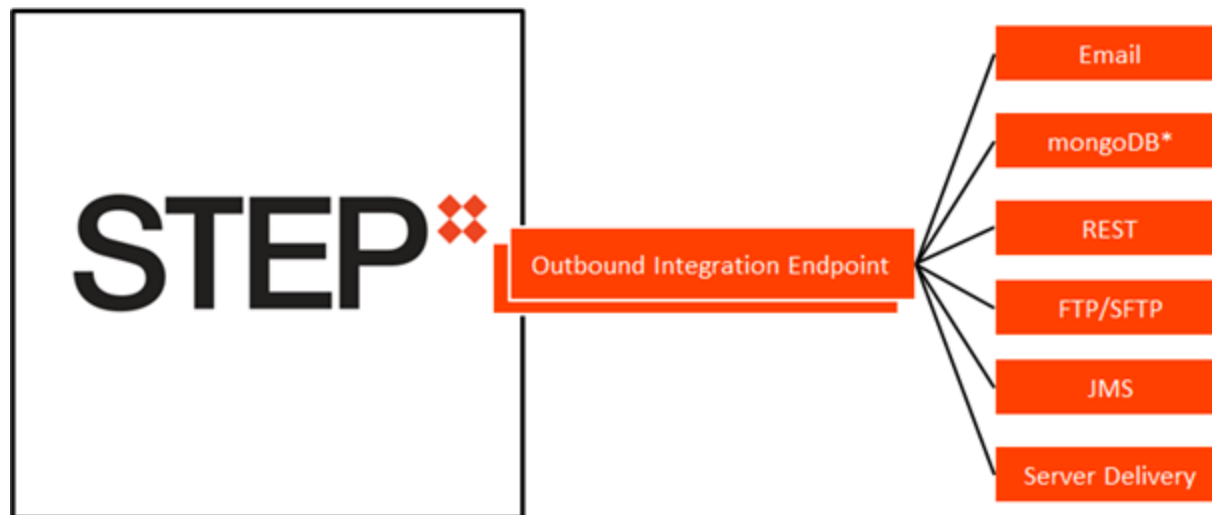
SFTP

> Hostname	SFTPDeliveryHostServerName
> Username	OIEPUser
> Password	xxxxxxxxxx
> File Name Template	\$filename-\$timestamp.\$extension
> Zip before upload	Yes
> Edit Delivery	

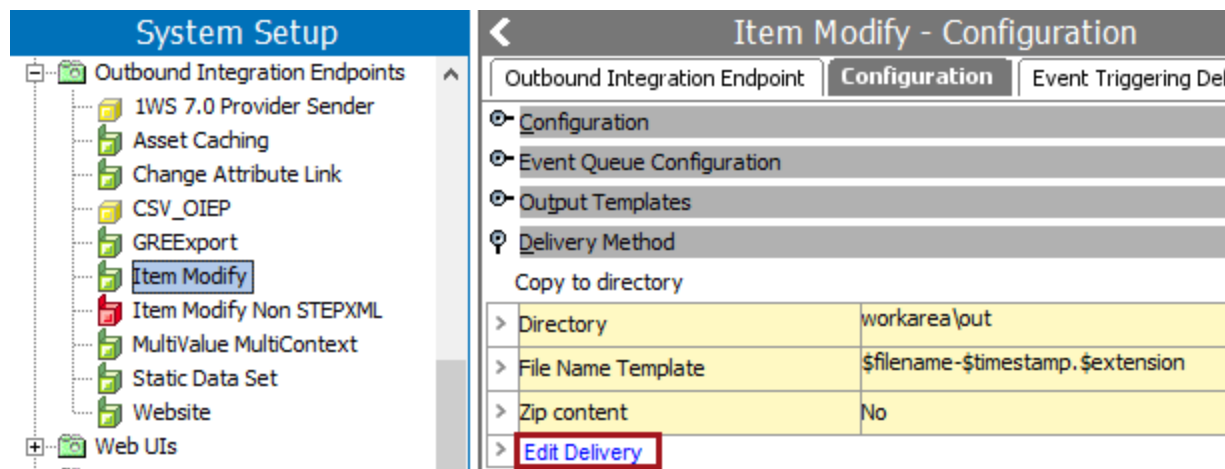
OIEP Delivery Methods

Delivery methods are used by both outbound data tools but the available options vary. For information on the delivery methods available in Export Manager, see **Export Manager Delivery Methods**.

An OIEP takes the data determined by its configuration and delivers it using the selected delivery method to the downstream system.



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 when exporting using an OIEP.

Method	Description
Copy to Directory	Delivers files to a directory on the application server and allows the delivery to be zipped. See Copy to Directory Delivery Method documentation.

Method	Description
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). You must specify the host name, user name, password, and the file name used to deliver the file. 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 About 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.
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 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). You must specify the host name, user name, password, and file name. See SFTP Delivery Method documentation.

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.

Note: The default delivery path is **opt/stibo/step** on the application server. This can be modified via the sharedconfig.properties file **DirectoryDeliveryPlugin.RootDir** property.

1. On the **Configuration** tab, navigate to the **Delivery Method** flipper, then click **Edit Delivery**.
2. In **Select Delivery Method**, choose **Copy to Directory**.

The screenshot shows a dialog box titled "Edit Delivery Configuration". It has a close button (X) in the top right corner. Inside the dialog, there is a section "Select Delivery Method" with a dropdown menu currently showing "Copy to directory". Below this, there are three fields: "Directory" (an empty text box), "File Name Template" (a text box containing "\$filename-\$timestamp.\$extension"), and "Zip content" (a dropdown menu currently showing "Yes"). At the bottom right of the dialog, there are two buttons: "OK" and "Cancel".

3. In **Directory**, specify the directory on the application server that will house the exported files. Text entered will be appended to the default path (opt/stibo/step) or to the updated path identified in the sharedconfig.properties file **DirectoryDeliveryPlugin.RootDir** property. Do not start this text with a slash (\).
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, 'filename' 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.
 - **\$timestamp** Inserts 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, use \$timestamp(ddMMyyyy) to format the timestamp as a readable date. For example, 15112016 is used for an export on 15 NOV 2016.

- **\$extension** The extension of the selected output file format is inserted where \$extension is used.

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**, specify if the output file should be delivered in a .ZIP file format. 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'. When uncompressed, the file containing STEP data is named 'exported.xml'.

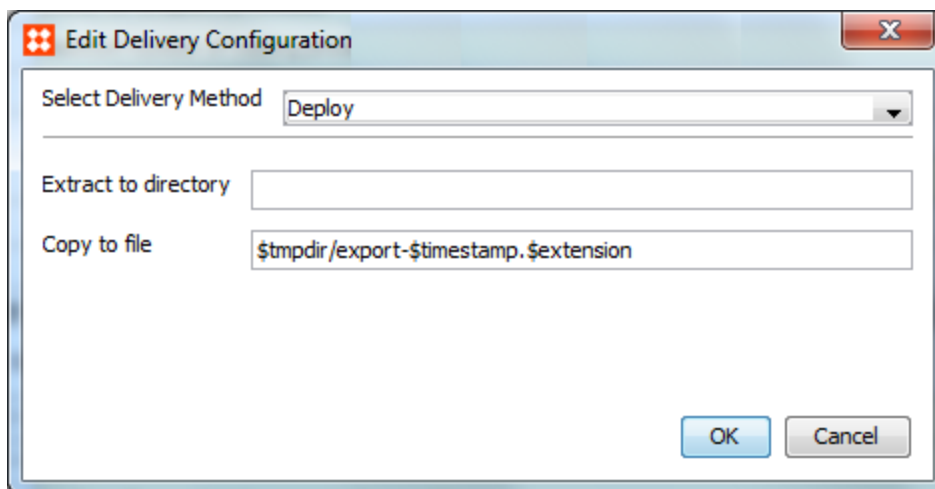
Deploy Delivery Method

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.

Note: The default delivery path is **opt/stibo/step** on the application server. This can be modified via the sharedconfig.properties file **DirectoryDeliveryPlugin.RootDir** property.

1. On the **Configuration** tab, navigate to the **Delivery Method** flipper, then click **Edit Delivery**.
2. In **Select Delivery Method**, choose **Deploy**.



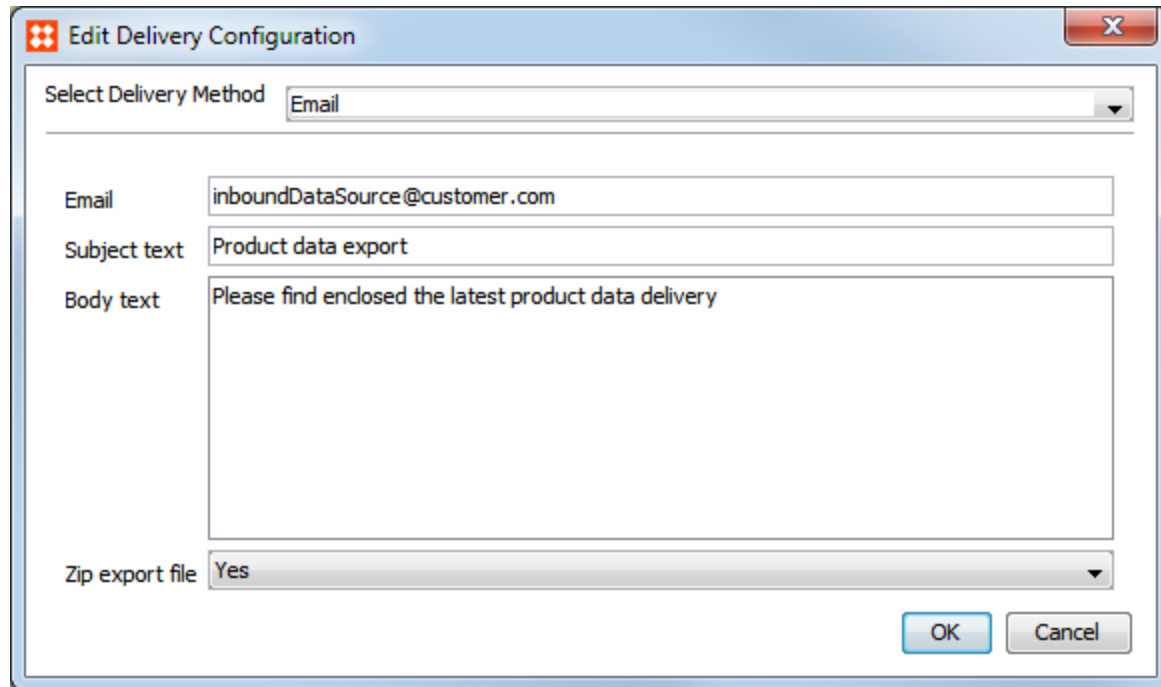
3. In **Extract to Directory**, specify a path on the application server where the exported .ZIP file should be unzipped. The field will only have an effect if the exported file is a .ZIP file, otherwise this field will be ignored.
4. In **Copy to file**, specify a file name template and a path where exported file should be saved.
 - **\$tmpdir** Points to the \$tmpdir path, defined in the configuration file of the server
 - **export-** Configurable text that can be replaced with any content that should be in the file name. For example, the Copy to File field could read '\$tmpdir/EmergencySKUExport-\$timestamp.\$extension' instead.
 - **\$timestamp** Inserts 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, use \$timestamp(ddMMyyyy) to format the timestamp as a readable date. For example, 15112016 is used for an export on 15 NOV 2016.
 - **\$extension** The extension of the selected output file format is inserted where \$extension is used.

Note: The 'Copy to file' 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.

Email Delivery Method

The email delivery option is very straight forward and, like 'Copy to directory', it allows the delivery to be zipped. This delivery method is available in Export Manager and in OIEPs.

1. In **Select Delivery Method**, choose **Email**.
2. Enter an email address, subject text, and body text.



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

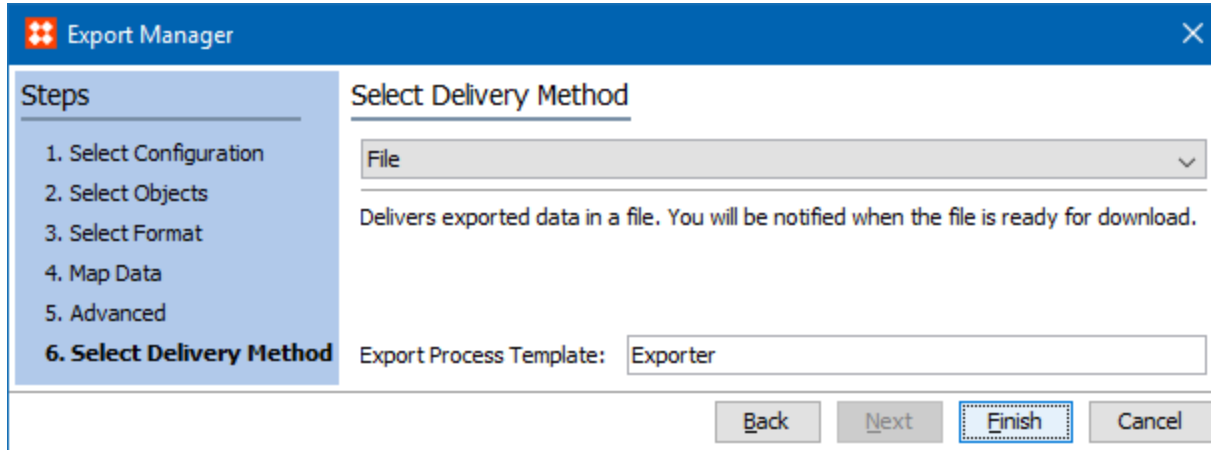
Zip export file: **Yes**

OK Cancel

3. In **Zip export file**, specify if you want to deliver the exported contents (email attachment) as a .ZIP file.

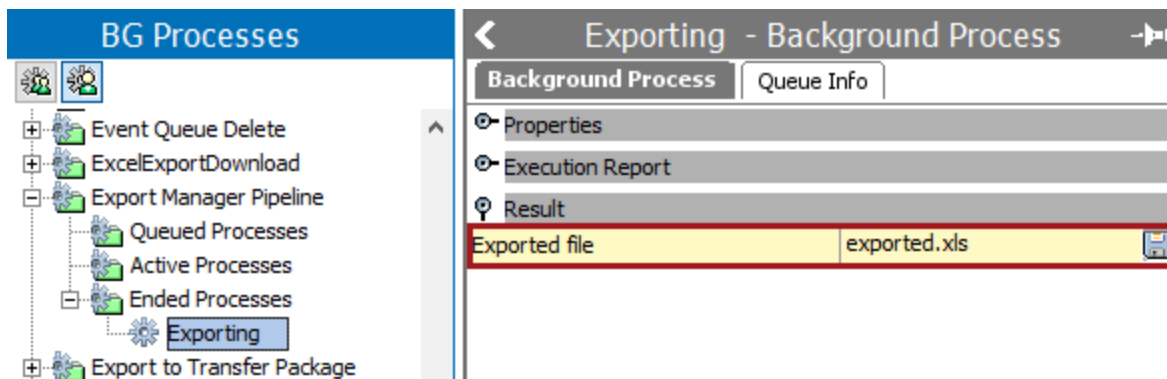
File Delivery Method

The file delivery option is only available in Export Manager. It makes the output available within STEP, via a background process. There are no additional parameter settings.



The image shows the 'Export Manager' dialog box with the 'Select Delivery Method' step selected. The 'File' method is chosen from the dropdown menu. The description states: 'Delivers exported data in a file. You will be notified when the file is ready for download.' The 'Export Process Template' is set to 'Exporter'. The 'Finish' button is highlighted.

By default, the output is saved on the Background Processes tab, under the Export Manager Pipeline node. While displaying the process, open the Result flipper on the Background Process tab.



The image shows the 'BG Processes' tab with the 'Export Manager Pipeline' node expanded. The 'Exporting' process is highlighted. The 'Exporting - Background Process' window is open, showing the 'Background Process' tab. The 'Result' flipper is open, displaying the 'Exported file' as 'exported.xls'.

FTP Delivery Method

The FTP delivery method allows an exported file to be delivered to an external system and is available in Export Manager and OIEP.

1. In **Select Delivery Method**, choose **FTP**.

2. In **Hostname**, select the host name or IP address of the FTP server to be used for the delivery. Options are available based on the FTPDeliveryHostName property in the sharedconfig.properties file on the application server. The required format of the property is (square brackets not included): FTPDeliveryHostName=1=[host1],2=[host2]. Using the hostname shown in the image above, the entry in the properties file would be: FTPDeliveryHostName=1=FTPDeliveryHostServerName. Contact Stibo Systems if you need assistance with configuration.
3. In **Username**, enter the user name that will be used to log on to the FTP server.
4. In **Password**, enter the password that will be used to log on to the FTP server.
5. 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 (e.g., 1438634758430). Using '\$timestamp(ddMMyyyy)' instead, formats the timestamp as a date. The placeholders for day, month, and year can be rearranged as needed to create the

desired date format. For example, a file generated on 29 March 2017 with the template '\$timestamp (ddMMyyyy).xls' is output as '29032017.xls' and the template '\$timestamp.xls' would use '1490810861593.xml' for the name.

- **\$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.

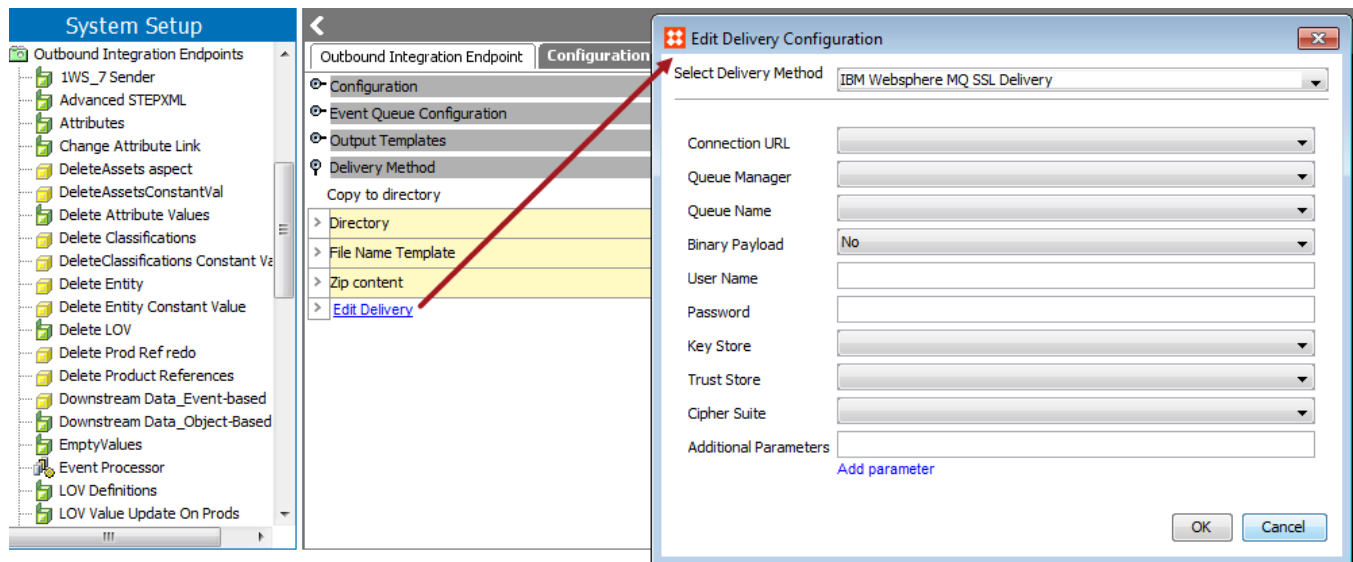
6. In **Zip before upload**, click **yes** to deliver a zipped file.

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.

Note: Options available on dropdown fields must be configured in sharedconfig.properties or config.properties. Select a dropdown to display the required key name for each field.

1. In the **Select Delivery Method** field, choose **IBM Websphere MQ SSL Delivery**.



2. In **Connection URL**, select the URL to connect to 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
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. To generate this, consult the manual for IBM Websphere MQ. The password for the key store must be configured in sharedconfig.properties or config.properties using 'WSMQSSLKeyStorePassword'.
9. In **Trust Store**, select the trust store with the CA for the Queue Manager. 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'.

10. In **Cipher Suite**, set to the same value as SSL CipherSuite in Websphere MQ. STEP is running on non-IBM jre, so this must be the same value as configured in the Queue Manager.
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.

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.

Note: Options available on dropdown fields must be configured in sharedconfig.properties using JMSConnectionFactoryName, JMSProviderURL, and JMSQueue. When no values are available, select a dropdown to display the required key name for each field. Contact Stibo Systems if you need assistance with configuration.

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. The list is determined by the JMSConnectionFactoryName property.
5. In **JMS Provider URL**, select a JMS Provider URL from the list. The list is determined by the JMSProviderURL property.
6. In **JMS Queue**, select the physical name of the JMS Queue to be used on Apache Active MQ or Websphere MQ. The list is determined by the JMSQueue property.
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.

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=FirsProviderURL=https://myproviderurl.com/url,SecondProviderURL=https://mypr
oviderurl2.com/url
JMSQueue=Queue1=FirstqueueName,Queue2=SecondqueueName2
```

FileInitialContextFactory JMS Delivery Configuration Example

The screenshot displays the 'System Setup' interface. On the left, a tree view shows various integration endpoints, with 'JMSWebsphereJNDIOut' highlighted. On the right, the configuration details for this endpoint are shown under the 'Configuration' tab. The 'JMS Delivery' section is expanded, showing the following settings:

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 setting 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.

Note: Options available on dropdown fields must be configured in sharedconfig.properties or config.properties.

1. In **Select Delivery Method**, choose **Mongo delivery**.

The screenshot shows the 'Edit Delivery Configuration' dialog box. The 'Select Delivery Method' dropdown is set to 'Mongo delivery'. Below it are fields for 'Server', 'Port', 'Raw DB prefix', 'User name', and 'Password'. There is a checkbox for 'Use SSL' and two dropdowns for 'Key store location' and 'Trust store location'. At the bottom, there is an 'Actions' section with an 'Add action' link and 'OK' and 'Cancel' buttons.

2. In **Server**, enter the server where the MongoDB instance used by the endpoint is running. The available servers are configured using the MongoDB.server property.
3. In **Port**, enter the network port used by the MongoDB instance on the specified server. The possible values are configured using the MongoDB.port property.
4. In **Raw DB prefix**, if required, enter a prefix for all raw databases the adapter creates. Optional.
5. In **User name**, if MongoDB is configured to use authentication, enter the MongoDB user. Optional.
6. In **Password**, if MongoDB is configured to use authentication, enter the MongoDB password. Optional.
7. 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 in sharedconfig.properties or config.properties, using MongoDB.SSLKeyStorePassword. For example,

```
MongoDB.SSLKeyStorePassword = keystore_password
```

- 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 in sharedconfig.properties or config.properties, using MongoDB.SSLTrustStorePassword. For example,

```
MongoDB.SSLTrustStorePassword = truststore_password
```

8. 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.

For additional information about the Mongo Delivery Method, see the **Mongo Delivery Method Elements** documentation.

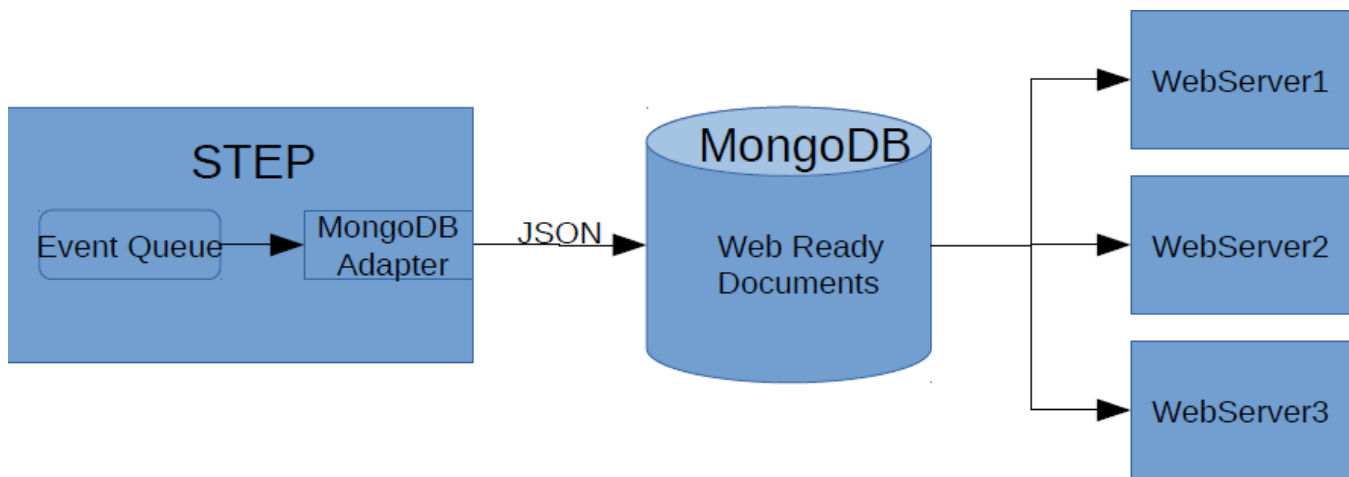
For more information, see **MongoDB Adapter Setup Quick Guides** documentation.

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.

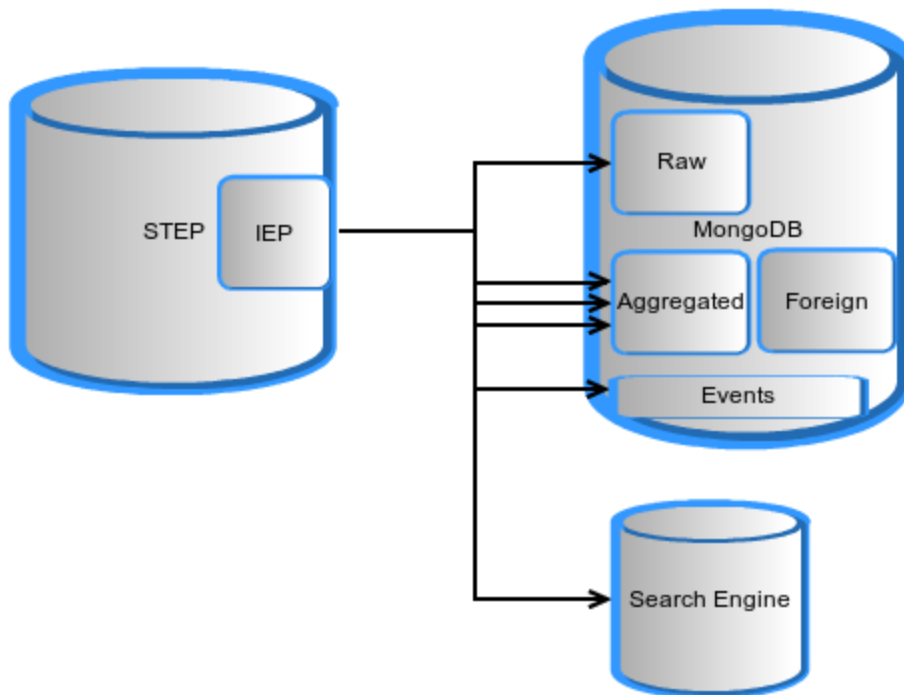
The Mongo adapter converts STEPXML to JSON and keeps a set of 'raw' collections in MongoDB that are in sync with STEP. Additionally, it allows you to add JavaScript triggers to maintain aggregated collections.

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:



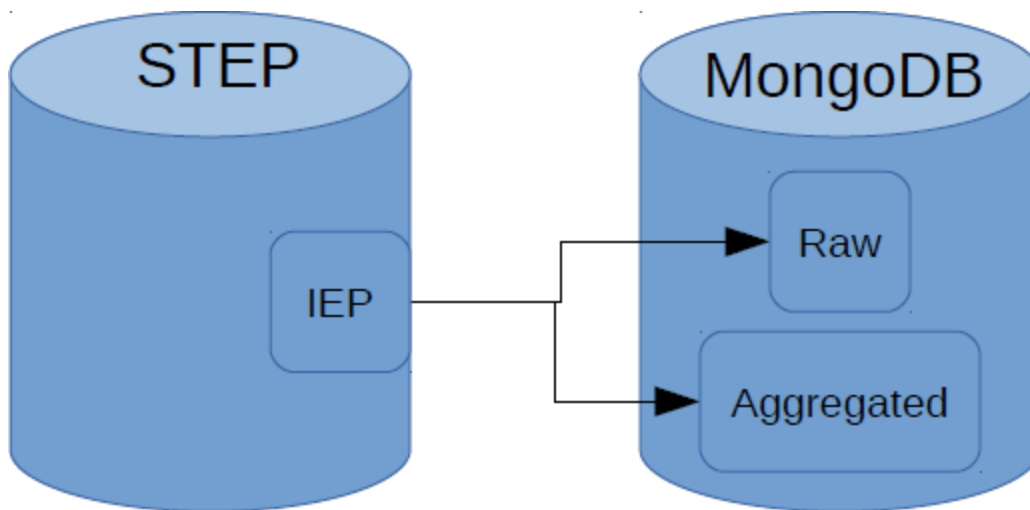
The STEP Mongo Adapter ensures that the raw collections are a one-to-one mirror of data in STEP. The adapter does not embed nodes found through references or child associations. These nodes are stored separately in the raw collections.

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 Collection

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 JavaScripts 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 **To Add Preconditions to a Business Rule** section of the **Editing Global Business Rules** 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 **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:

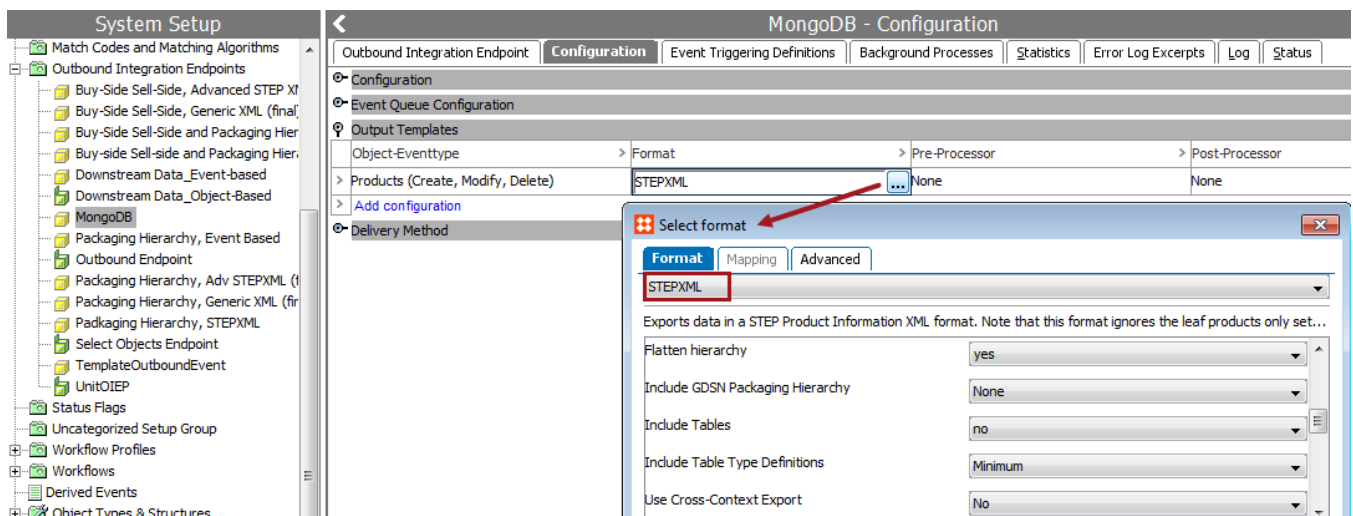
```
MongoDB.Server = 1=Server1, 2=Server2
```

```
MongoDB.Port = 1=27712
```

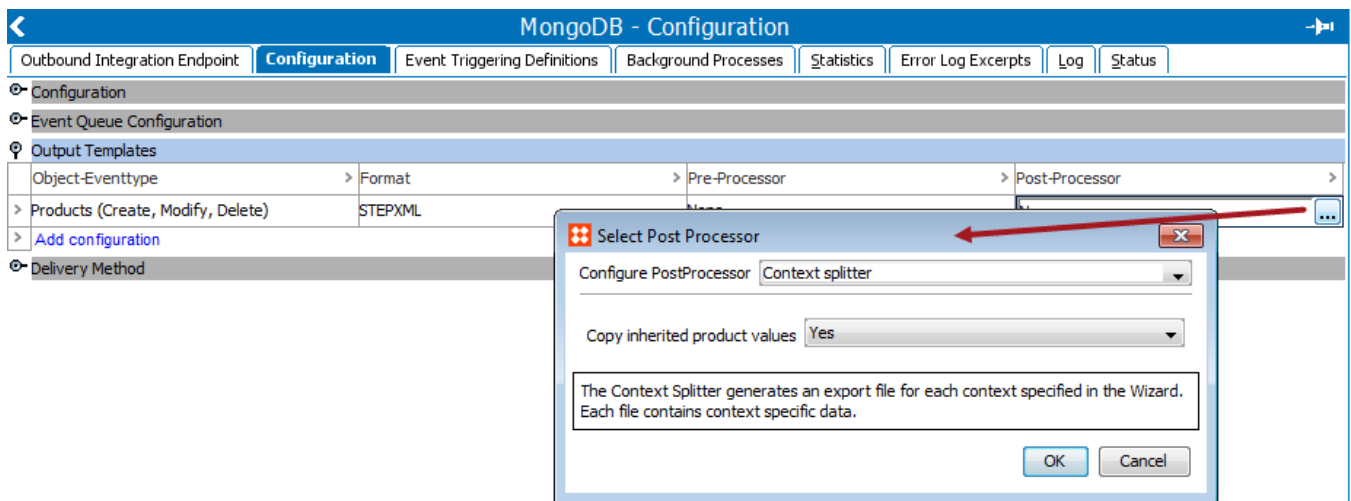
Configure the OIEP

The STEP MongoDB Adapter is run within an Outbound Integration Endpoint (OIEP) and configured as a delivery plugin for the endpoint.

1. Create an OIEP, selecting STEP Exporter as the Process Engine.
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 STEPXML export format in the dropdown at the top of the dialog
 - Set Flatten hierarchy = Yes
 - Set Include Type Definitions = Minimum
 - Set Include Unit Definitions = Minimum
 - Set Include List of Values Definitions = Minimum
 - Set Include Attribute Group Definitions = Minimum
 - Set Include Assets = Selected
 - Set Include Classifications = Selected
 - Set Include Products = Selected
 - Set Include Entities = Selected
4. In the Post-Processor field, click the ellipsis button (...) and select **Context splitter**, then click OK.



5. 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:

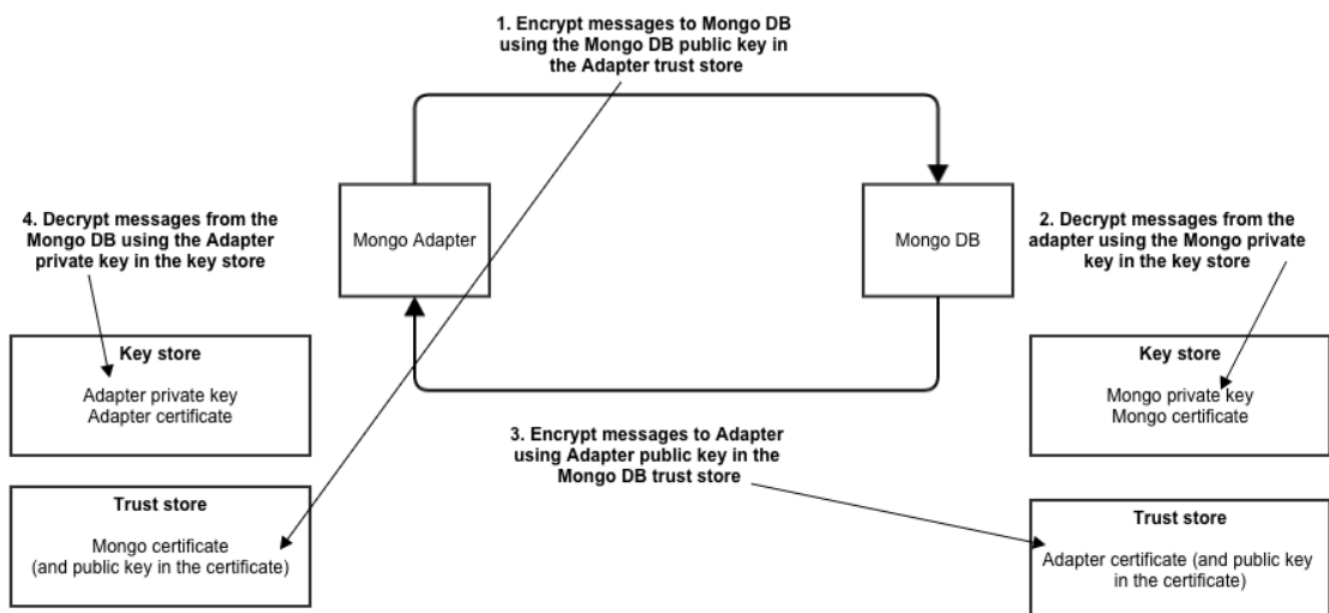
mongod --auth --config
```

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"
  }
}
```

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" } ]
  }
}

```

Here, 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 'Include Type Definitions' 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" : "centimetre"
  },
  "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 'Include Unit Definitions' 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"
}

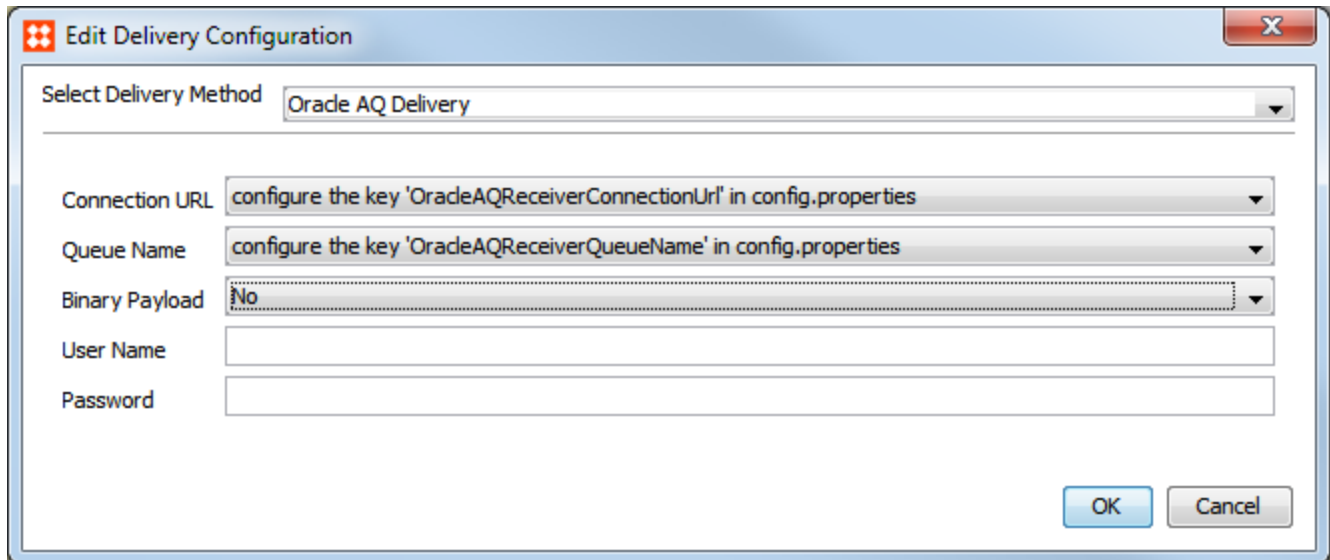
```

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.

1. In the **Select Delivery Method** list, choose **Oracle AQ Delivery**.



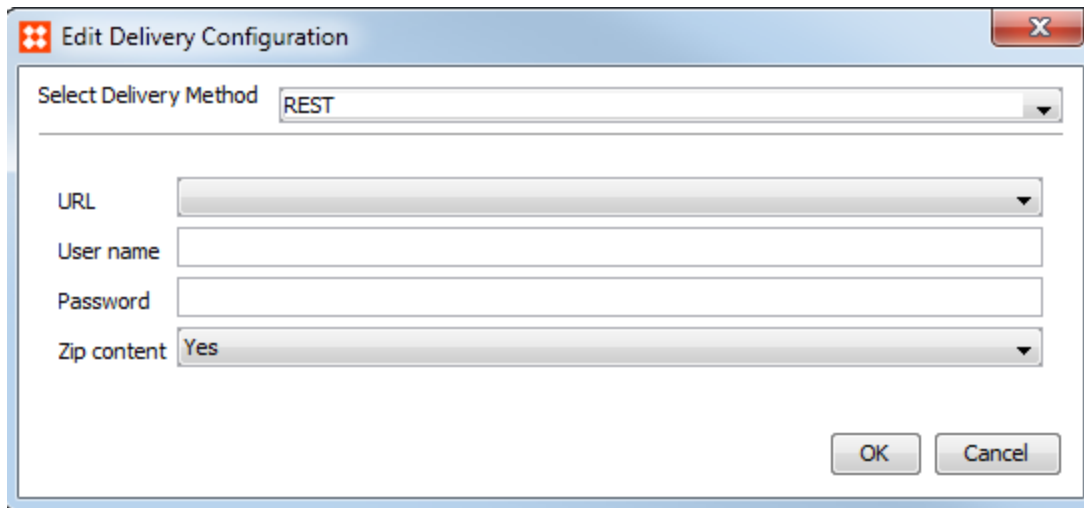
The screenshot shows a window titled "Edit Delivery Configuration". At the top, there is a "Select Delivery Method" dropdown menu with "Oracle AQ Delivery" selected. Below this, there are five configuration fields: "Connection URL" (a dropdown menu showing "configure the key 'OracleAQReceiverConnectionUrl' in config.properties"), "Queue Name" (a dropdown menu showing "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 of the window are "OK" and "Cancel" buttons.

2. In the **Connection URL** list, select a URL that points to Oracle AQ. Options are available based on the OracleAQReceiverConnectionURL property in the sharedconfig.properties file on the application server. Contact Stibo Systems if you need assistance with configuration.
3. In the **Queue name** list, select an Oracle AQ queue name. Options are available based on the OracleAQReceiverQueueName property in the sharedconfig.properties file on the application server. Contact Stibo Systems if you need assistance with configuration.
4. For **Binary Payload**, select Yes or No ('Yes' will be 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.

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.

1. For **Select Delivery Method**, choose **REST**.



The screenshot shows a dialog box titled "Edit Delivery Configuration". It has a close button (X) in the top right corner. Inside the dialog, there is a section labeled "Select Delivery Method" with a dropdown menu currently showing "REST". Below this, there are four input fields: "URL" (a text box with a dropdown arrow), "User name" (a text box), "Password" (a text box), and "Zip content" (a dropdown menu currently showing "Yes"). At the bottom right of the dialog are two buttons: "OK" and "Cancel".

2. In **URL**, select the URL that points to the REST endpoint where you would like to receive the delivered data. 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.

Note: The available URLs are system dependent and must be configured in the `sharedconfig.properties` file under the property name `RESTDeliveryURL`. Contact Stibo Systems if you need assistance with configuration.

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.

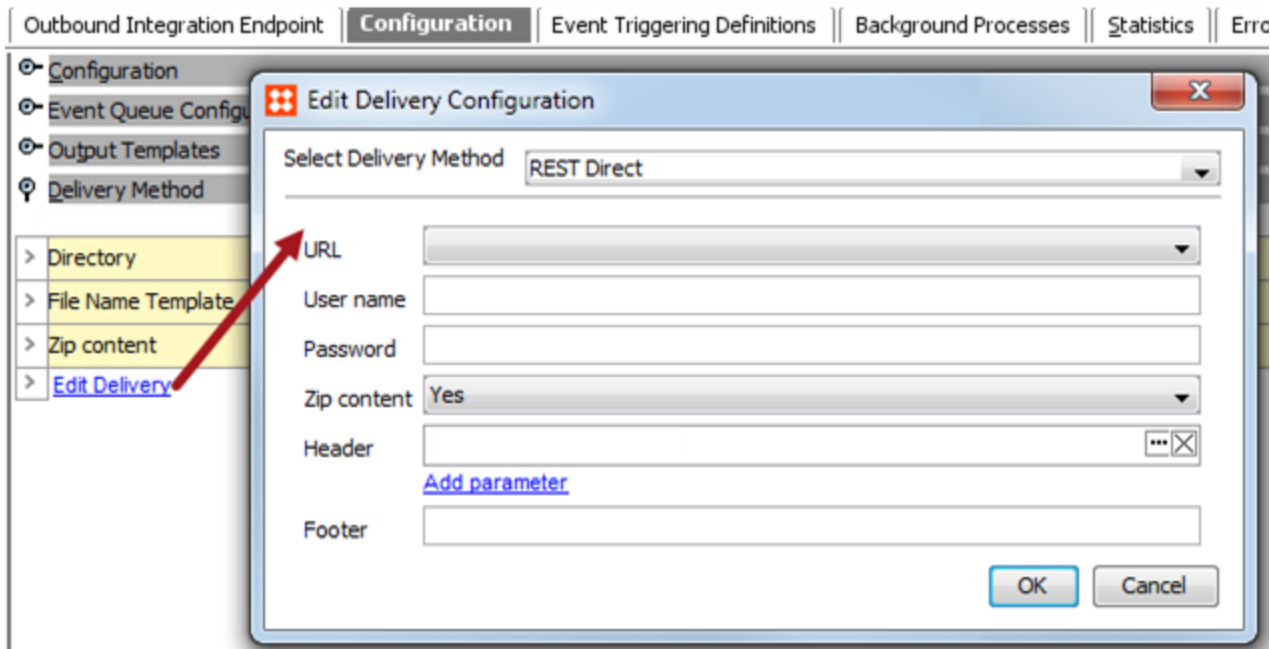
For information about how to use the REST API to upload files to REST, see the STEP API documentation.

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.

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. This delivery option is only available in OIEPs and if the delivery fails, the OIEP will be disabled. There is no special resilience handling.

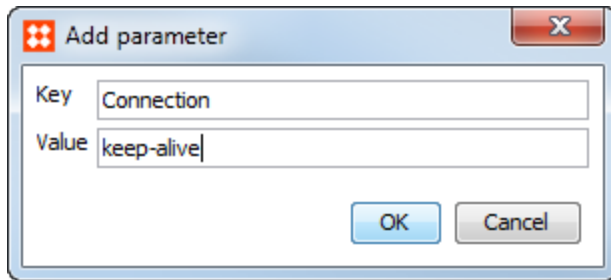
1. For **Select Delivery Method**, choose **REST Direct**.



2. In **URL**, select the URL that points to the REST endpoint where you would like to receive the delivered data.

Note: The available URLs are system dependent and must be configured in the sharedconfig.properties file under the property name RestDirectDeliveryURL. Contact Stibo Systems if you need assistance with configuration.

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. For **Header**, click the **Add Parameter** link and add a key and a value. This is part of the HTTP network protocol.



7. 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 pairs



Note: Once a header is displayed, use the ellipsis button (...) to edit it or the X button to remove it.

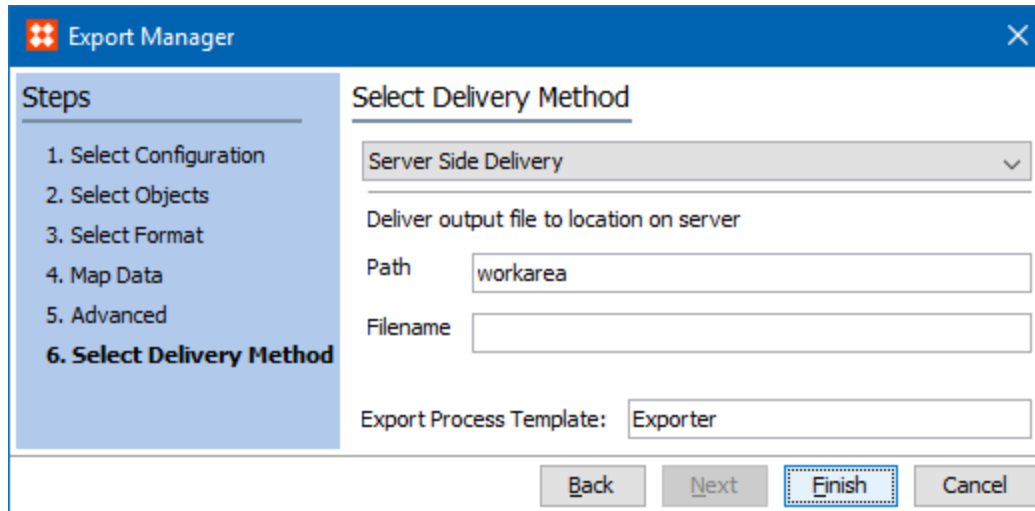
8. In **Footer**, add data required for the recipient to verify that the full message was received. This is part of the HTTP network protocol.

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.

9. On the Edit Delivery Configuration dialog, click the OK button to save the delivery method.

Server Side Delivery Method

The server side delivery option is only available in Export Manager. It makes the output available within STEP, via a background process.



Export Manager

Steps

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

Select Delivery Method

Server Side Delivery

Deliver output file to location on server

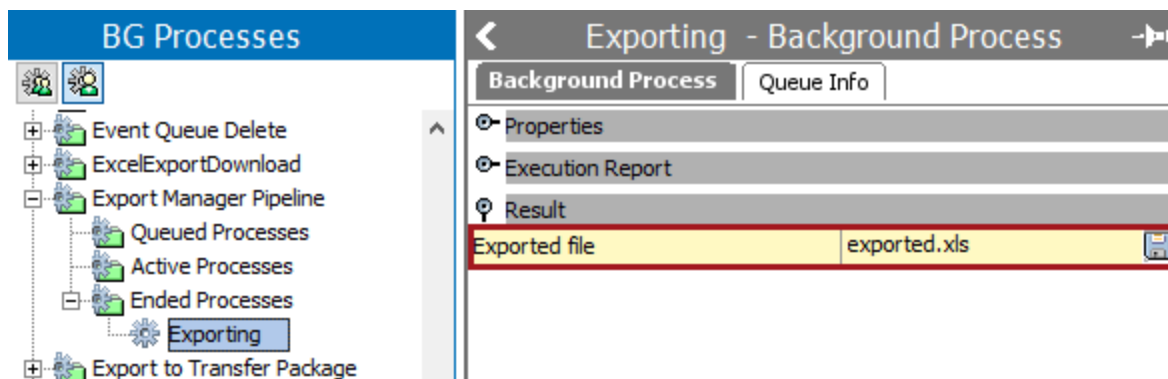
Path: workarea

Filename:

Export Process Template: Exporter

Back Next **Finish** Cancel

By default, the output is saved on the Background Processes tab, under the Export Manager Pipeline node. While displaying the process, open the Result flipper on the Background Process tab.



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

SFTP Delivery Method

This delivery method is available in Export Manager and in OIEPs.

1. In the Select Delivery Method list, select **SFTP** from the dropdown.

2. In **Hostname**, select the host name of the FTP server to be used for the delivery. Options are available based on the FTPDeliveryHostName property in the sharedconfig.properties file on the application server. The required format of the property is (square brackets not included): FTPDeliveryHostName=1=[host1],2=[host2]. Using the hostname shown in the image above, the entry in the properties file would be: FTPDeliveryHostName=1=FTPDeliveryHostServerName. Contact Stibo Systems if you need assistance with configuration.
3. In **Username**, enter the username that will be used to log on to the FTP server.
4. In **Password**, enter the password that will be used to log on to the FTP server.
5. 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 (e.g., 1438634758430). Using '\$timestamp(ddMMyyyy)' instead, formats the timestamp as a date. The placeholders for day, month, and year can be rearranged as needed to create the desired date format. For example, a file generated on 29 March 2017 with the template '\$timestamp

(ddMMyyyy).xls' is output as '29032017.xls' and the template \$timestamp.xls' would use '1490810861593.xml' for the name.

- **\$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.

6. In **Zip before upload**, click **yes** to deliver a zipped file.

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.

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

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 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 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.

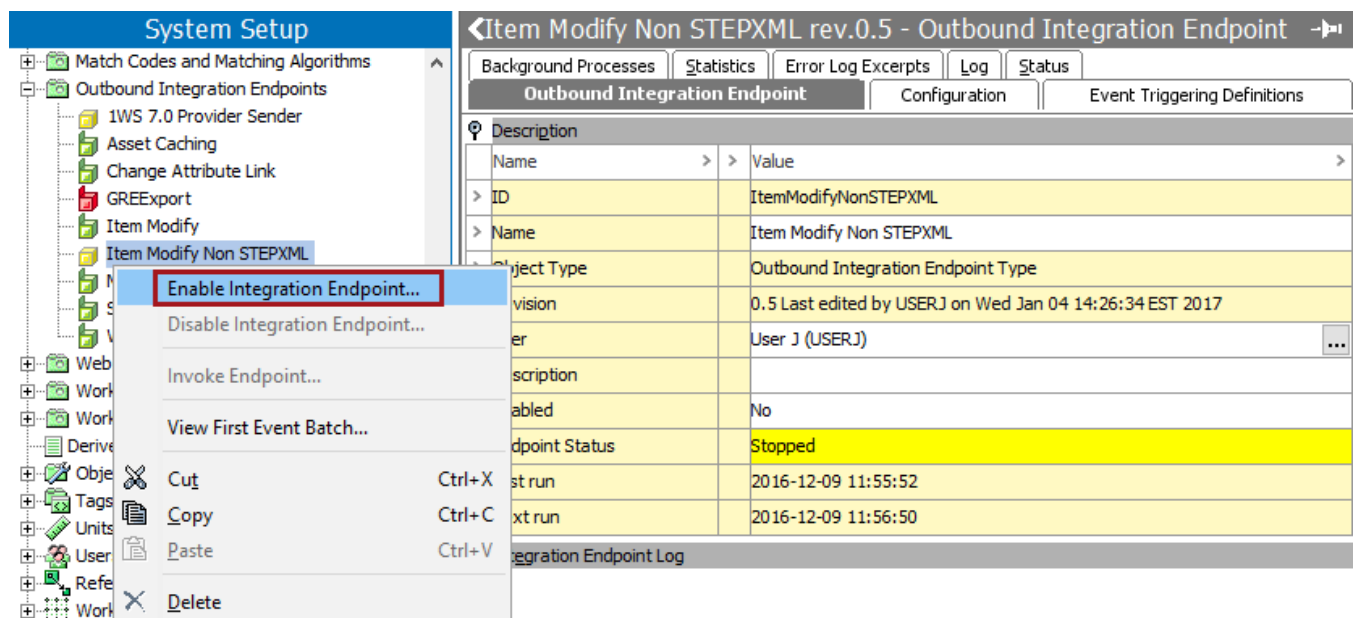
Prerequisite for Event-Based OIEPs

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.



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

- 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

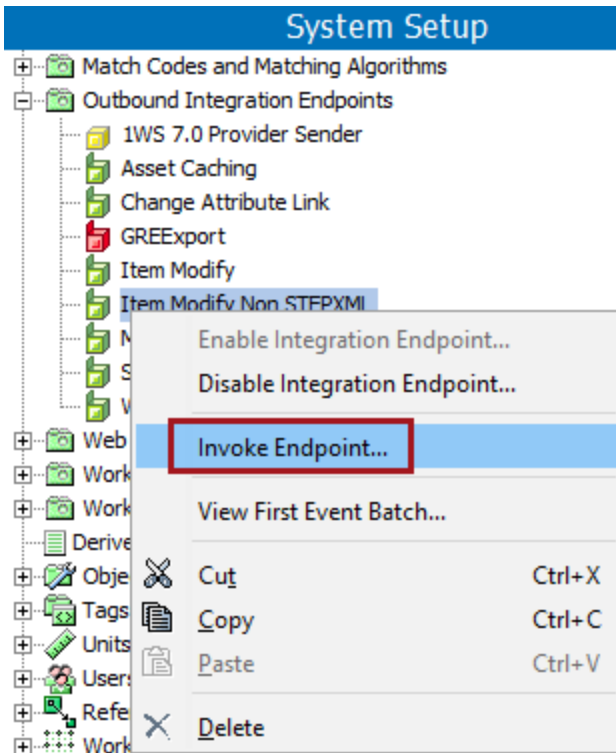
Process Engine	STEP Exporter
Error reporter	Not Defined
Schedule	Start every minute
Queue for endpoint	OutboundQueue
Queue for endpoint processes	Out
Transactional settings	Strict
Number of threads	1
Maximum number of old processes	1000
Maximum age of old processes	1y
Contexts	English US
Workspace	Approved

Event Queue Configuration

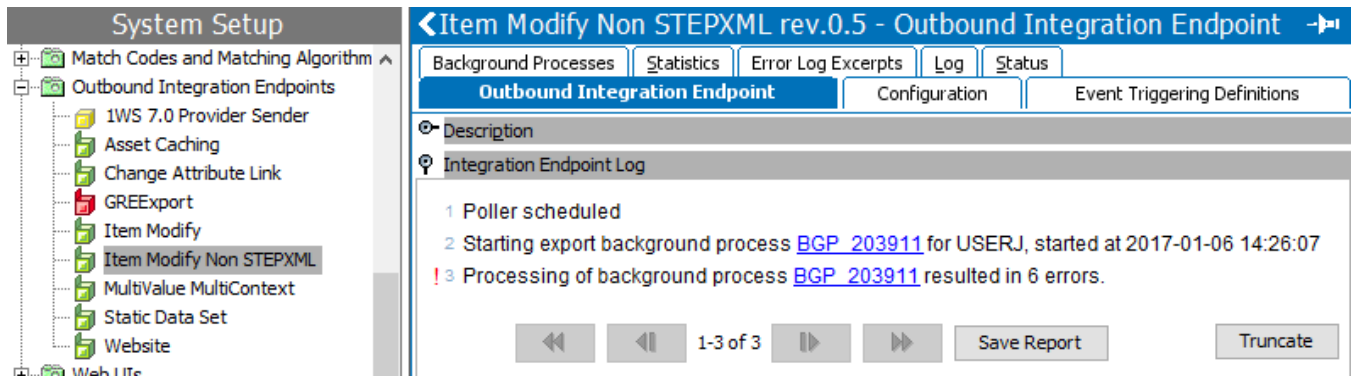
Output Templates

Delivery Method

- If 'Not scheduled', or if the IIEP needs to be invoked on demand, right-click the enabled IIEP and click **Invoke Endpoint**.

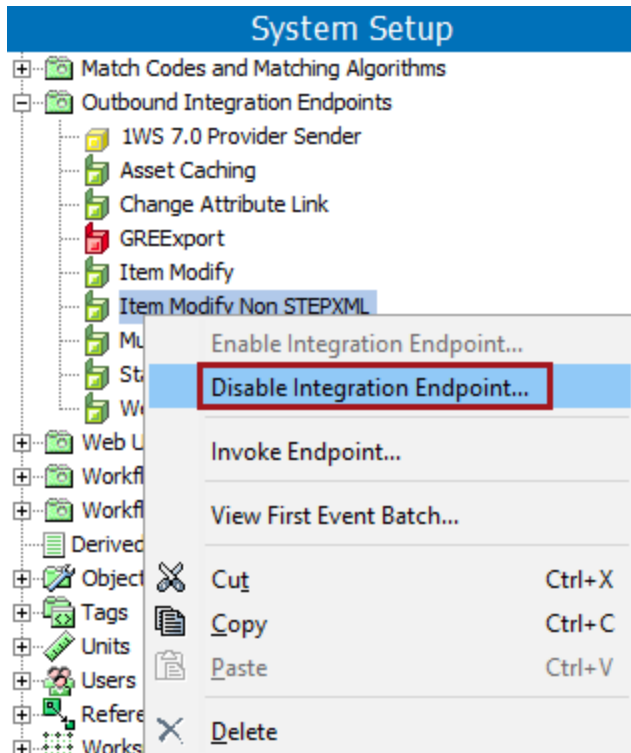


4. When invoked on schedule and manually, files or messages ready for export as identified by the OIEP configuration are processed. The Outbound Integration Endpoint tab Integration Endpoint Log flipper displays the time the endpoint was invoked. For scheduled OIEPs, the log includes a link to the background 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 (🟡).

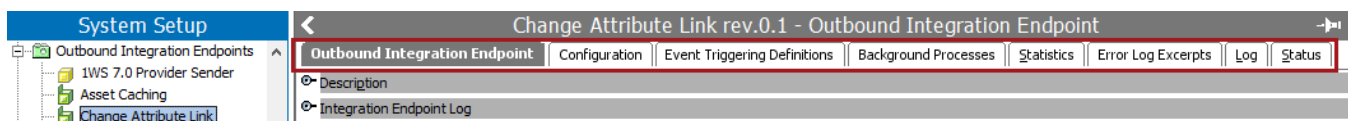


Once disabled, the outbound integration endpoint can be restarted using the Enable Integration Endpoint step defined above.

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 Stibo 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.

- The background process (**BGP**) link opens the individual Importer Background Processes.

Outbound Integration Endpoint	Configuration	Event Triggering Definitions	Background Processes	Statistics	Error Log Excerpts	Log	Status
Description							
Integration Endpoint Log							
1 Poller scheduled 2 Poller invoked by user Mon Oct 24 13:54:46 EDT 2016 3 Starting export background process BGP 189801 for USERJ, started at 2016-10-24 13:54:46 4 Starting export background process BGP 189804 for USERJ, started at 2016-10-24 14:07:47 5 Poller invoked by user Mon Oct 24 14:15:31 EDT 2016							

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.

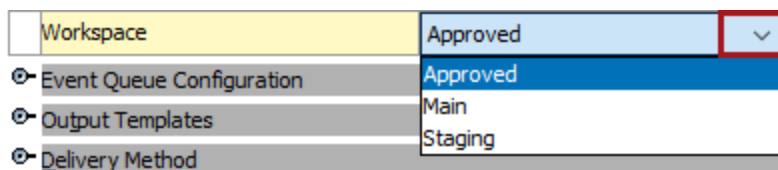
Outbound Integration Endpoint	Configuration	Event Triggering Definitions	Background Processes	Statistics	Error Log Excerpts	Log	Status
Configuration							
Process Engine	STEP Exporter						
Error reporter	Not Defined						
Schedule	Not scheduled						
Queue for endpoint	OutboundQueue						
Queue for 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.

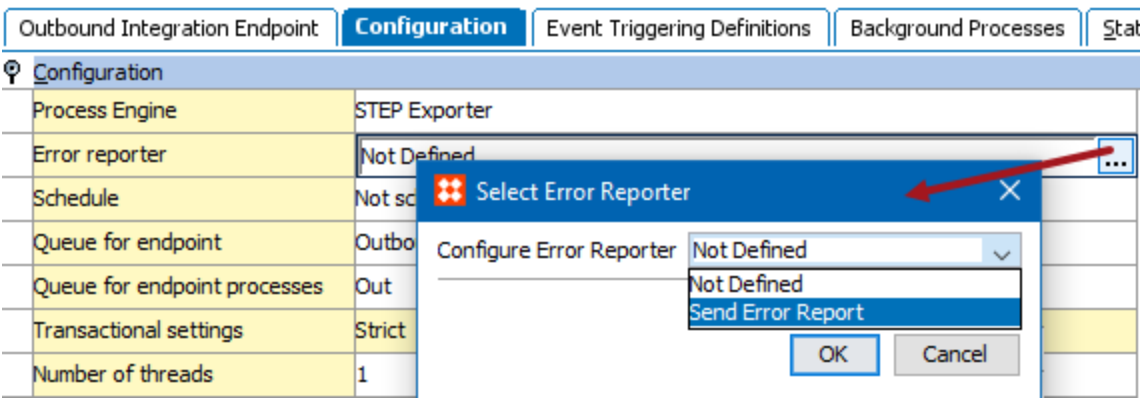
For information on the Process engine or Workspace, see **OIEP - Event-Based - Configure Endpoint** or **OIEP - Select Objects - Configure Endpoint**.



- 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. Pre-processors and post-processors 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				

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. Hover over an ID to display the background process (**BGP**) link, click it to open the individual Exporter 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 in parallel. 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
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

Statistics Tab

This tab provides statistics about the number of requests handled, run times, and wait times.

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 '18203'			
> BGP_206613	150	Row 3: Parent classification '18203' for classification '18203'			

Log Tab

This tab provides information about changes to the OIEP configuration.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
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						

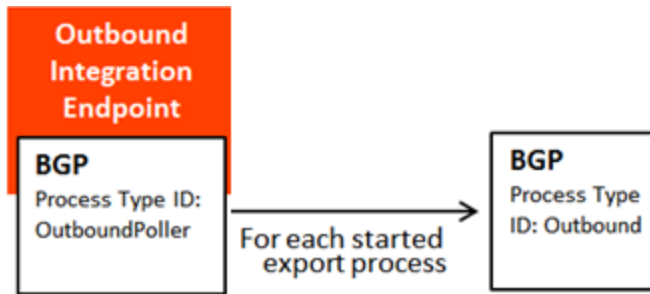
Status Tab

This tab provides information about revisions, hidden values and diagnostics.

Outbound Integration Endpoint	Configuration	Background Processes	Statistics	Error Log Excerpts	Log	Status
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						

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		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	

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.

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]
```

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 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).

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, 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>
```

An example of the output for an outbound integration endpoint definition as comments is shown below:

```
<OutBoundIntegrationEndpoint...>
<!--Definition: [IEP definition goes here. Removed for brevity.-->]
[Remaining configuration would populate here]
</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 **Include Integration End Points** section of the **STEPXML Outbound Parameters** documentation.

Handling Failed OIEP Background Processes

You can reactivate a failed integration endpoint 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

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.

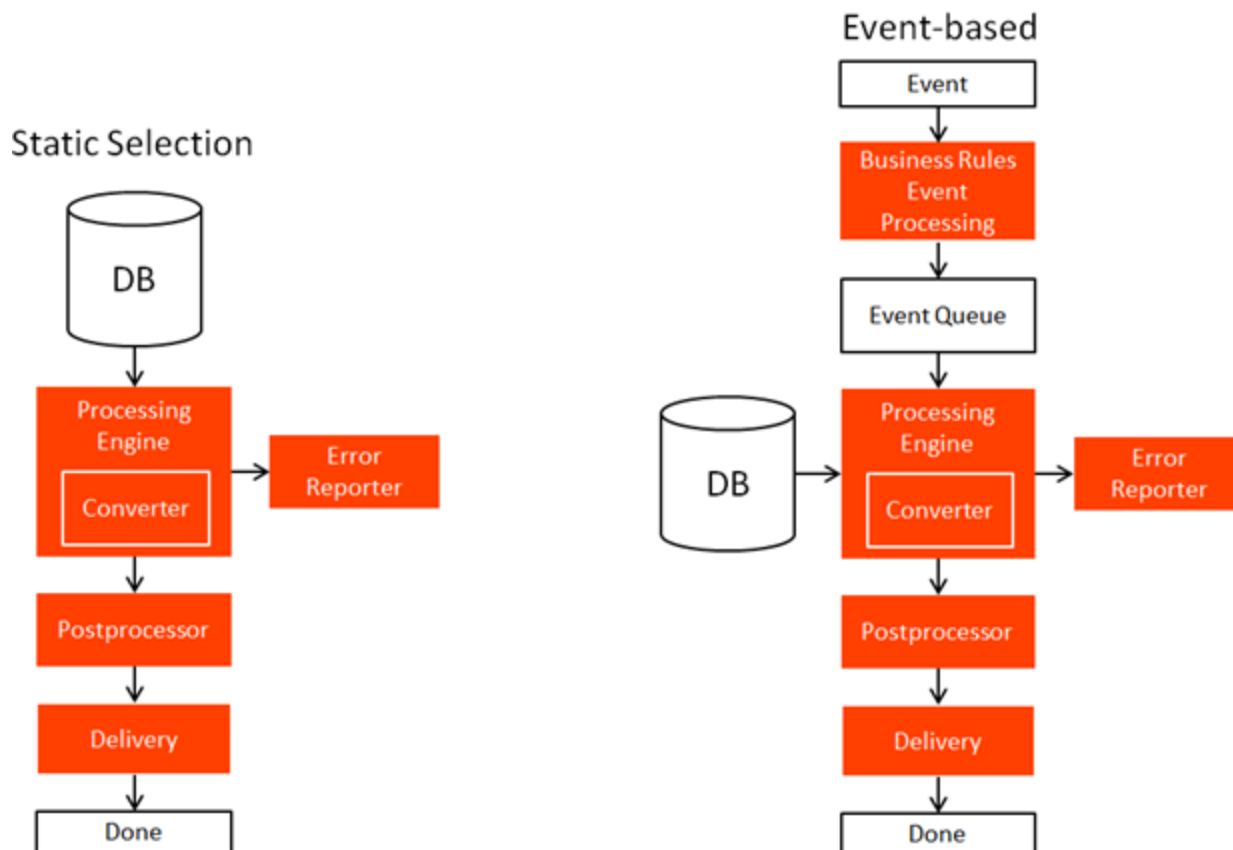
Error Log Excerpts		
Outbound Integration Endpoint	Configuration	Background Processes
Statistics	Error Log Excerpts	Log
Integration Endpoint Log		
Background Pro...	Log Item No	Text
> BGP_4730623	30	Object without matching format template. Object type id: 'Product', object id: 'A-1001-3'
> BGP_4730623	40	java.lang.NullPointerException
> BGP_4730623	50	Caught RuntimeException at Fri Jan 23 09:33:46 CET 2015: java.lang.NullPointerException
> BGP_4730626	30	Object without matching format template. Object type id: 'Product', object id: 'A-1001-3'

2. In the **Background Process** column, click the link for the background process to correct.
3. In the Background Process editor **Execution Report**, view the progress of the background process, and where it failed.
4. In the Properties area > **Value** column, click the Save  button near the failed status, and save the file locally.
5. Open the file in a relevant editor and make the required changes.
6. After completing the changes, upload the file back to the same background process. In the Background Process editor > Properties area > **Value** column, click the Upload  button near the failed status.
7. Restart the same process in the Properties area > **Value** column, click the Restart  button near to the failed status. The background process is restarted.

8. If the background process is part of a batch of chained processes, you also have to 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.



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**.

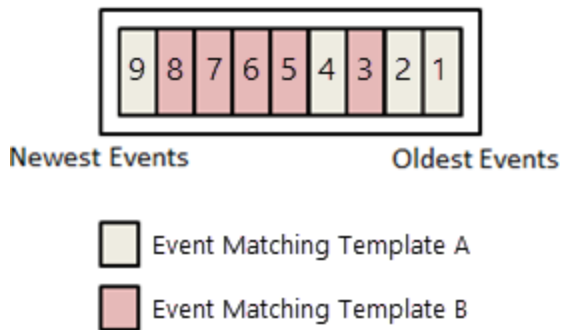
Event-Based OIEP Event Batching

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 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 based on 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':



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 / Super User Guide** documentation.

Event-Based OIEP Examples

This section includes event-based OIEP configuration examples for semi-advanced to advanced cases:

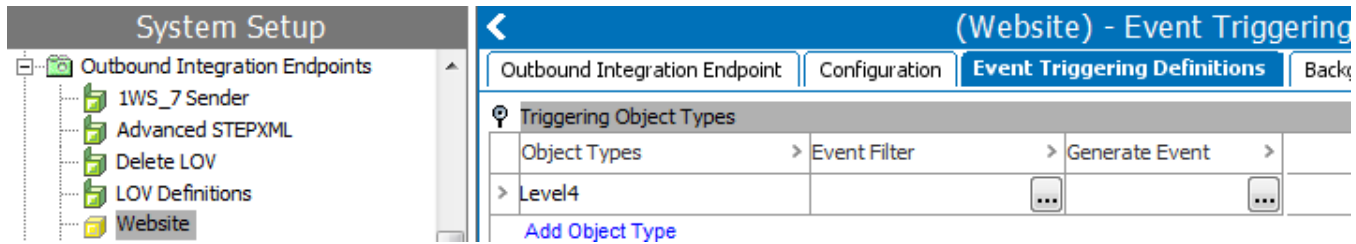
- Event-Based Example Business Rules for Derived Events
- Event-Based Example Publishing of Leaf Products with Inherited Data
- Event-Based Example Upward Inheritance
- Event-Based Example Products Linked to Classification

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.



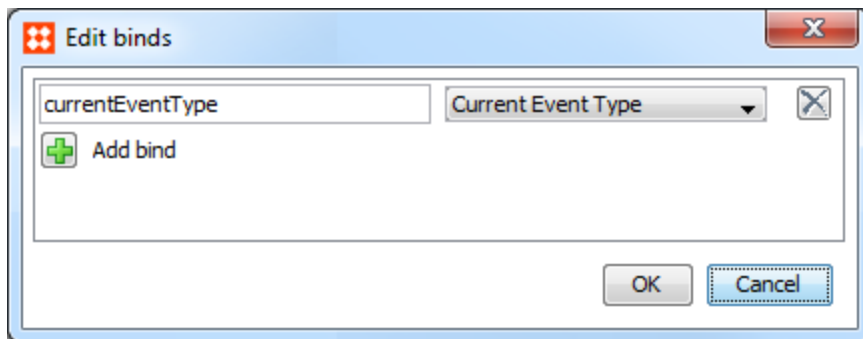
Important: The following scripts are only an example and should not be used as-is without thorough testing, including updating to match object and link types that exist on your system.

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.

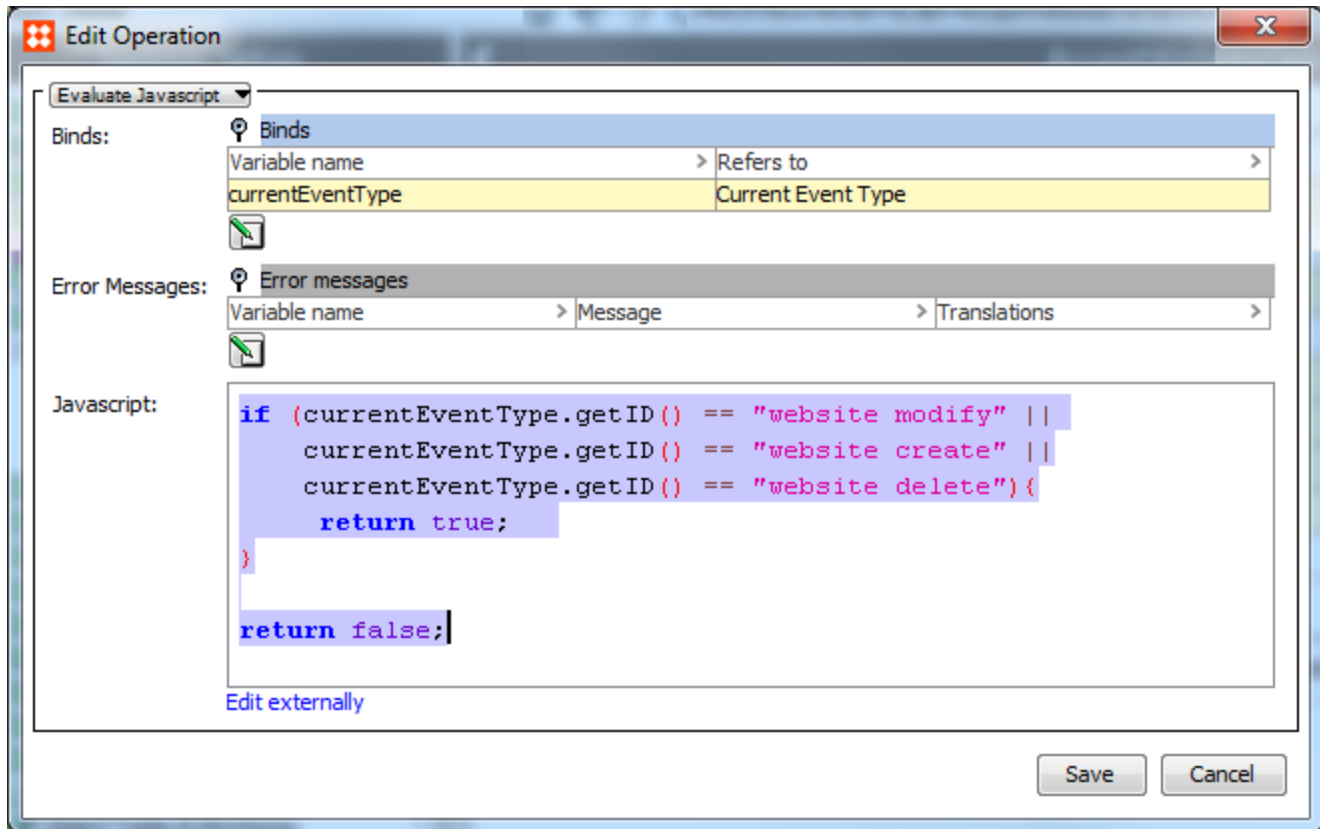
Create a new **business condition** and make it valid for the appropriate object types. Use the **Evaluate JavaScript** operation and add a bind for the Current Event Type.



Use the following JavaScript code to identify the derived event types:

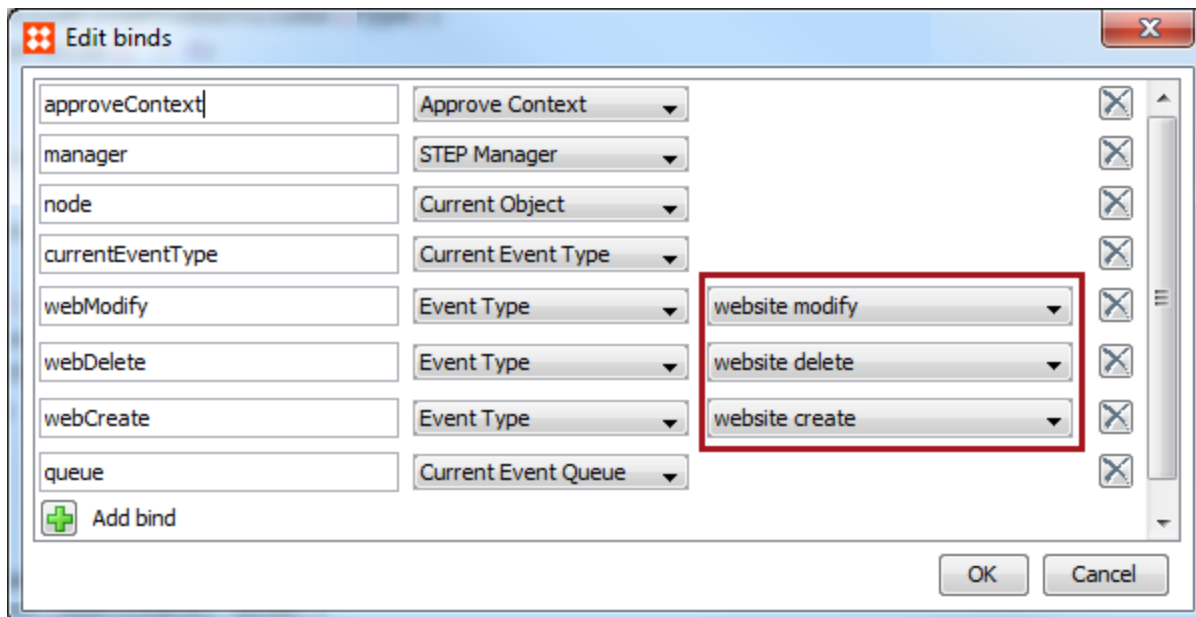
```
if (currentEventType.getID() == "website modify" ||
    currentEventType.getID() == "website create" ||
    currentEventType.getID() == "website delete"){
    return true;
}

return false;
```

Event Generator - Only standard derived event types can be used with JavaScript-based business actions. Additional information on JavaScript syntax for STEP is available in the STEP API documentation.

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:



Use the following JavaScript code to create a derived event based on the approved change on the product:

```

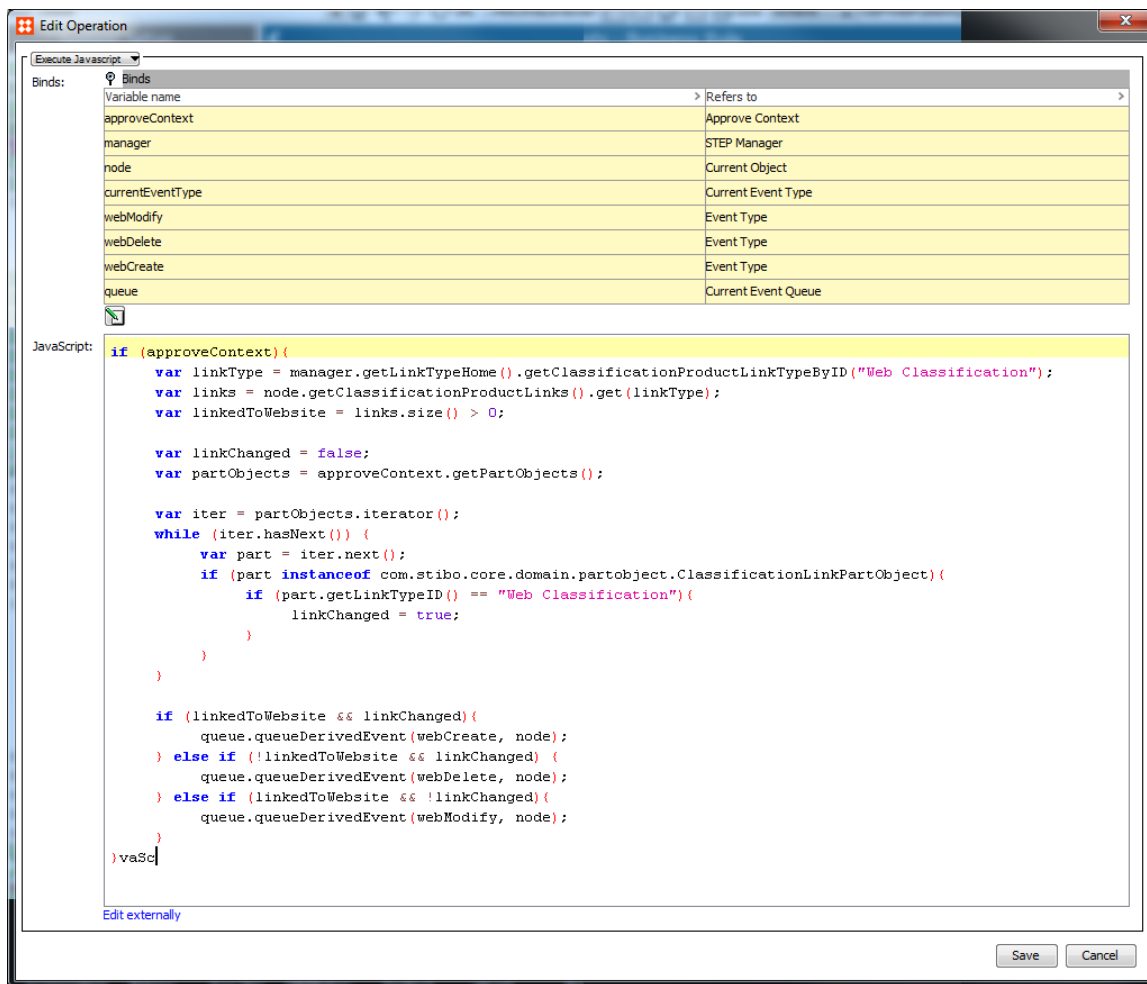
if (approveContext){
    var linkType = manager.getLinkTypeHome
    ().getClassificationProductLinkTypeByID("Web Classification");

    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.

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:

Triggering Object Types			
Object Types	>	Event Filter	>
> Sales Item		...	...
> Level 3, Sales Item Family	LP Always False (LPAlwaysFalse)	...	LP Republish Sales Item Descendants (...
Add Object Type			

Use the following logic for the grandparent and parent levels (Level 3 and Sales Item Family):

Evaluate Javascript

Binds

Variable name

>

Refers to

>



Error Messages

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):

currentObject	Current Object	
logger	Logger	
manager	STEP Manager	
currentEventQueue	Current Event Queue	
salesItemRepublish	Event Type	salesItemRepublish 

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.

The script to generate derived events for approved Sales Items could look as follows:

```
function prodIsInApp(prodID) {
    var appProd;
    manager.executeInWorkspace("Approved", function(appManager) {
        appProd = appManager.getProductHome().getProductByID(prodID);
    });
    return (appProd != null);
}

function generateEventForApprovedLeafSalesItems(ancestor) {
    var children = ancestor.getChildren();
    for (var i = 0; i < children.size(); i++) {
        var current = children.get(i);
        if (current.getObjectType().getID() == "SalesItem") {
            if (prodIsInApp(current.getID())) {
                logger.info("Generating 'salesItemRepublish' event for approved descendant Sales Item '" +
                    current.getID() + "'.");
                currentEventQueue.queueDerivedEvent(salesItemRepublish, current);
            }
        }
        else generateEventForApprovedLeafSalesItems(current);
    }
    generateEventForApprovedLeafSalesItems(currentObject);
}
```

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.

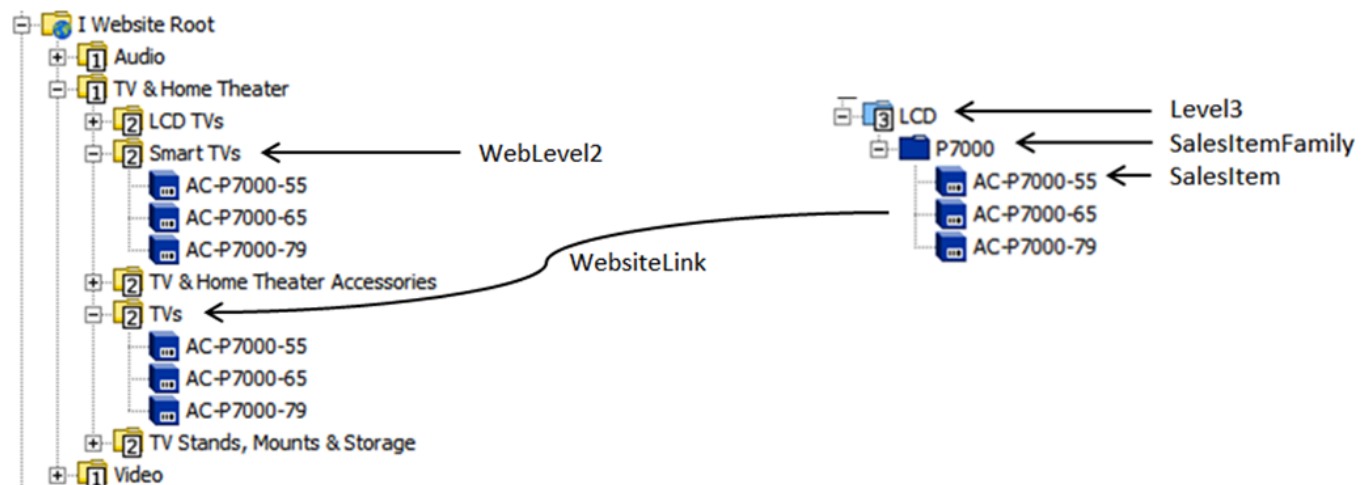
Output Templates			
Object-Eventtype	> Format	> Post-Processor	>
> Sales Item (Create, Modify, Delete, salesItemRepublish)	Generic XML (4 mappings)	MultiContextSplitter	
> Add configuration			

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 has the link should be on 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 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 approved 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:

Triggering Object Types			
Object Types	Event Filter	Generate Event	
> Sales Item	WP Reject Core Events (WPRejectCoreEvents)	...	WP Generate Derived Events (WPGenerateDerivedEv...
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

Refers to

currentEventType

Current Event Type

logger

Logger

Error 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:

logger	Logger		X
currentEventQueue	Current Event Queue		X
manager	STEP Manager		X
currentEventType	Current Event Type		X
appContext	Approve Context		X
webAdd	Event Type	webAdd	X
webModify	Event Type	webModify	X
webRemove	Event Type	webRemove	X
currentObject	Current Object		X
 Add bind			

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.
- 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 here, 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.


```

var prodClassLinkTypeID = "WebsiteLink"
var ClassObjTypeID = "WebLevel2";
var linkType = manager.getLinkTypeHome().getClassificationProductLinkTypeByID
(prodClassLinkTypeID);
//Utility function to check whether a Classification exists in Approved
function classInApp(classID) {
var res;
manager.executeInWorkspace("Approved", function(appManager) {
res = appManager.getClassificationHome().getClassificationByID(classID) != null;
});
return res;
}
if(currentEventType instanceof com.stibo.core.domain.eventqueue.BasicEventType) {
var appProd;
//Event generated based on an approval
if(appContext) {
var mainProd = appContext.getMainNode();
appProd = appContext.getApprovedNode();
var stepDelete = currentEventType ==
com.stibo.core.domain.eventqueue.BasicEventType.Delete;

var linkInMain = !mainProd.getClassificationProductLinks().get(linkType).isEmpty();
var linkInApp = false;
if(appProd) {
linkInApp = !appProd.getClassificationProductLinks().get(linkType).isEmpty();
}
//Case A - Prod has been delete approved and has link in Approved > WebRemove
if(stepDelete && linkInApp) {
currentEventQueue.queueDerivedEvent(webRemove, mainProd);
}
//Case B - Prod has link in both Main and Approved > WebModify
else if(linkInMain && linkInApp) {
currentEventQueue.queueDerivedEvent(webModify, mainProd);
}
else {
var partObjects = appContext.getPartObjects().toArray();
var linkToBeApproved = false;
for(var i = 0; i < partObjects.length; i++) {
if(partObjects[i] instanceof com.stibo.core.domain.partobject.ClassificationLinkPartObject)
{
var currentClassID = partObjects[i].getClassificationID();

```

```

var currentClassObjTypeID = manager.getClassificationHome().getClassificationByID
(currentClassID).getObjectType().getID();
if(currentClassObjTypeID == ClassObjTypeID && classInApp(currentClassID)) {
linkToBeApproved = true;
break;
}
}
}
//Case C - Prod has link in Main but not in Approved and link is part object > WebAdd
if(linkInMain && linkToBeApproved) {
currentEventQueue.queueDerivedEvent(webAdd, mainProd);
}
//Case D - Prod has link in Approved but not in Main and link is part object > WebRemove
if(linkInApp && linkToBeApproved) {
currentEventQueue.queueDerivedEvent(webRemove, mainProd);
}
}
}
//Event based on republish or change to externally maintained data
else {
manager.executeInWorkspace("Approved", function(appManager) {
appProd = appManager.getProductHome().getProductByID(currentObject.getID());
});
//Case E - Non approval based event has been generated and prod is in Approved with link >
WebModify
if (appProd && !appProd.getClassificationProductLinks().get(linkType).isEmpty()) {
currentEventQueue.queueDerivedEvent(webModify, currentObject);
}
}
}

```

Output Template - If different messages are to be output dependent on the derived event type, three output templates should be configured as shown below.

Output Templates		
Object-Eventtype	Format	Post-Processor
> Sales Item (webModify)	Generic XML (4 mappings)	MultiContextSplitter
> Sales Item (webAdd)	Generic XML (4 mappings)	MultiContextSplitter
> Sales Item (webRemove)	Generic XML (1 mappings)	MultiContextSplitter
> Add configuration		

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.

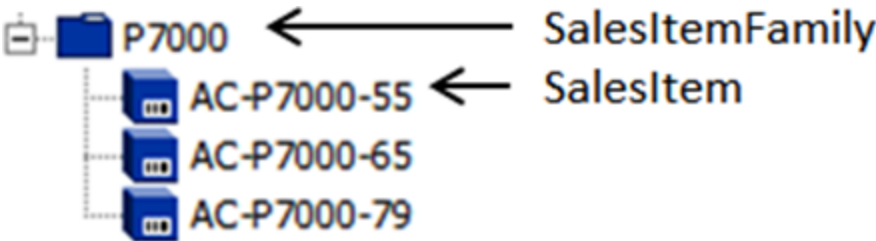
Solution:

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:



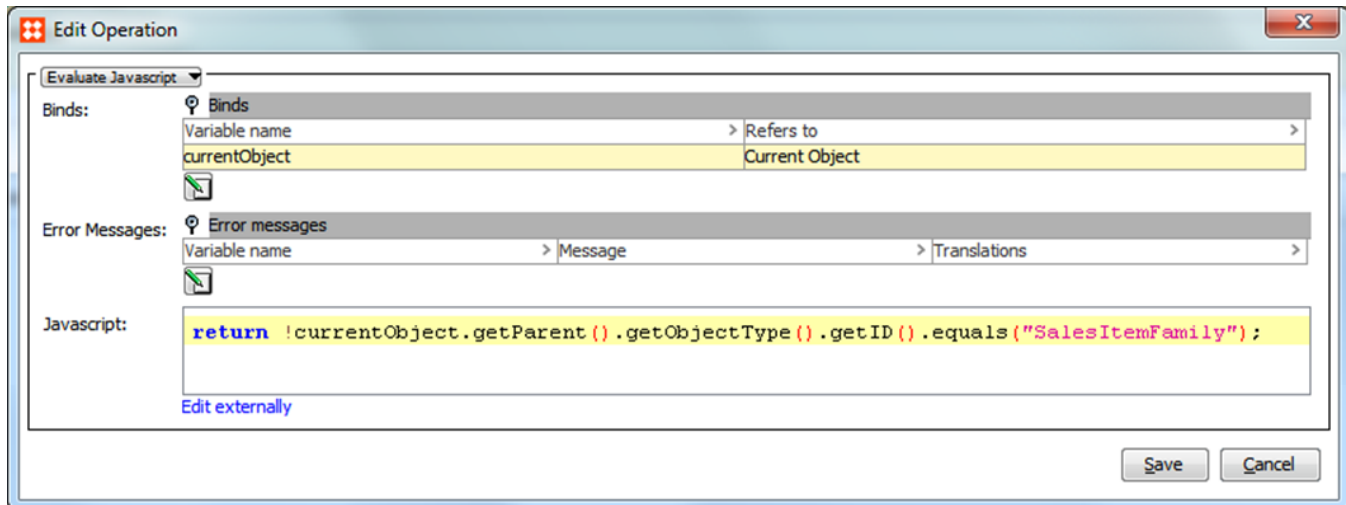
Use the following triggering object types setup:

Triggering Object Types		
Object Types	Event Filter	Generate Event
> Sales Item	UI Skip Sales Items Below Family (UISkipSalesItemsBelowFan...	UI Republish Sales Item Family (UIRepublishSalesItemFamily...
> Sales Item Family		
Add Object Type		

No Filter or Generator is used for the Sales Item Family Object Type as all Events should pass for this type.

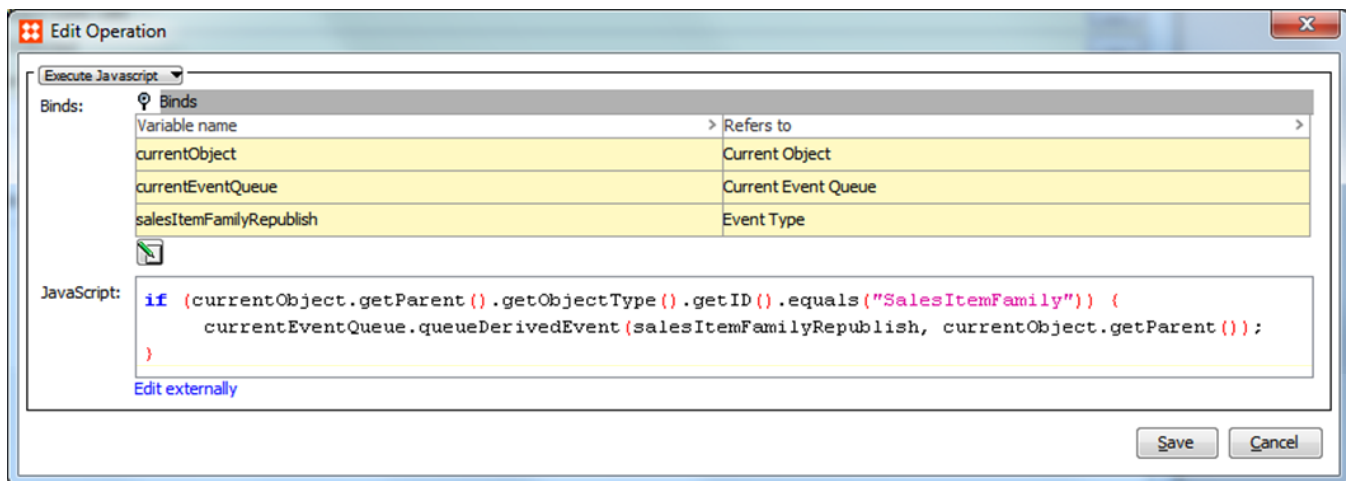
Use the following logic for the Sales Items:

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:



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.



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.

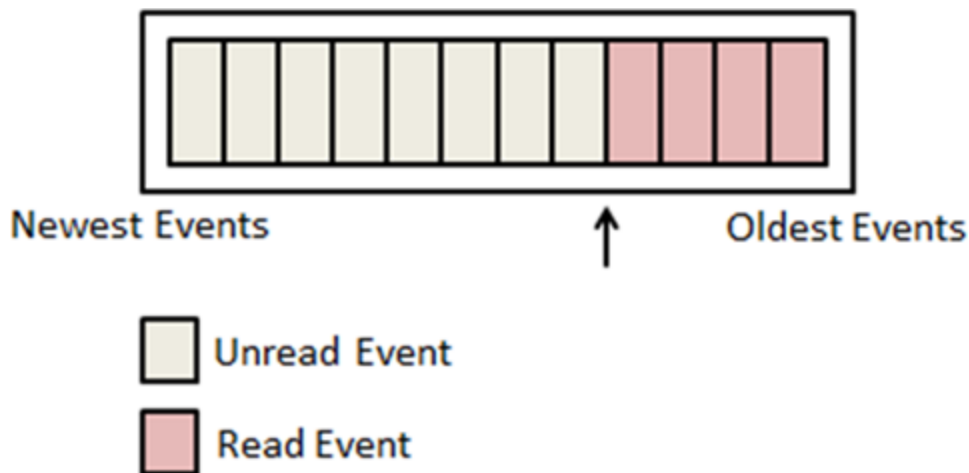
Output Templates		
	Format	Post-Processor
Object-Eventtype		
> Sales Item, Sales Item Family (Create, Modify, Delete, SalesItemFamilyRepublish)	STEPXML	MultiContextSplitter
> Add configuration		

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.

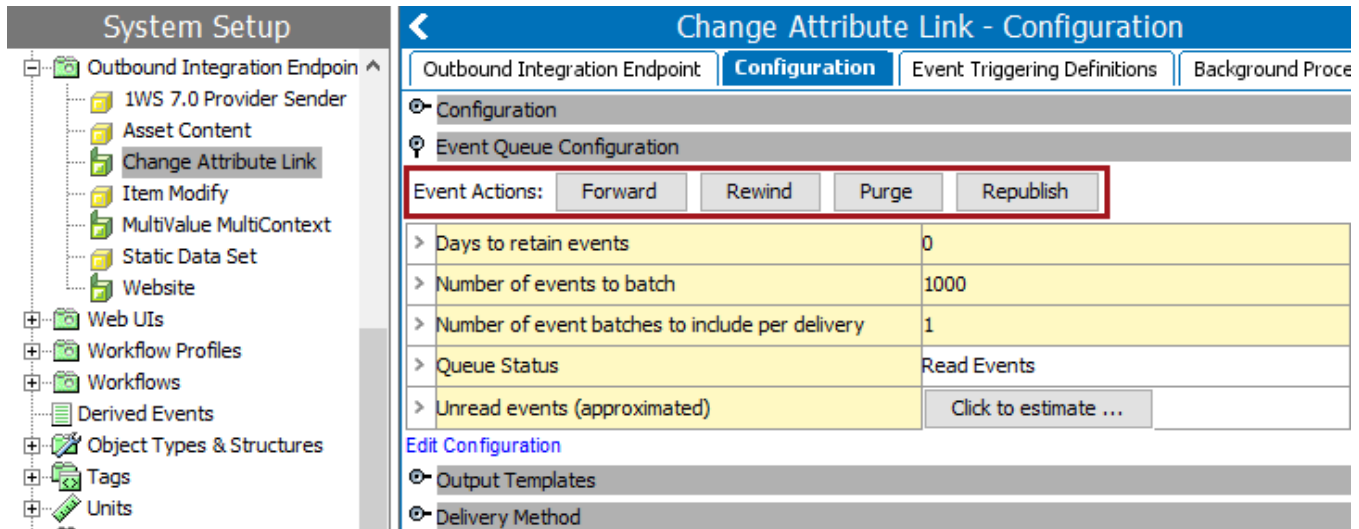
For example, if the queue is configured to retain events after they have been read, read events can be distinguished from unread events, as illustrated by the arrow in the following image.



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.



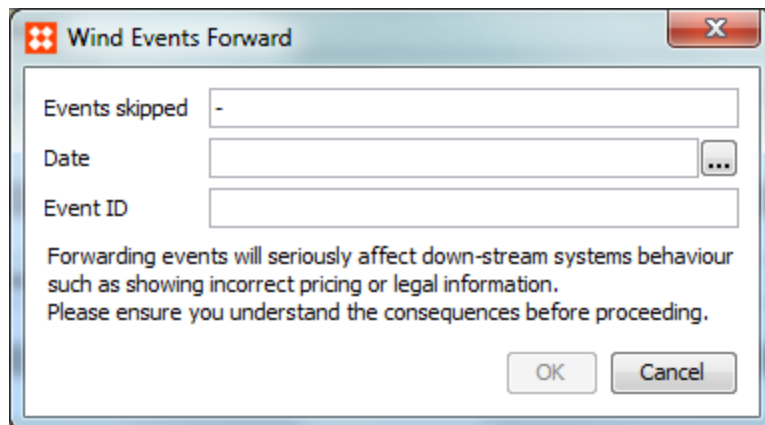
For information on the number and types of unread events, see the **Event-Based OIEP Queued Events** documentation.

Forward Events

1. In the **Wind Events Forward** dialog, click the ellipsis button (...) in the **Date** field.
2. In the **Date Picker**, select the date from which you want to skip events.

When you have selected a date, 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.

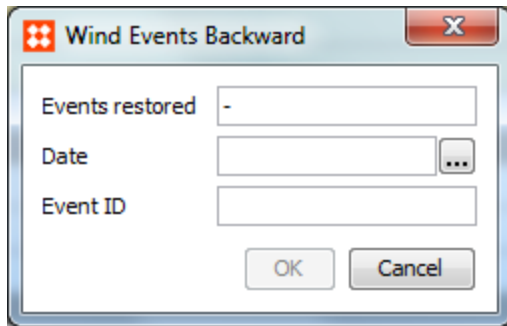


Rewind Events

1. In **Wind Events Backward** dialog, click the ellipsis button (...) in the **Date** field.
2. In the **Date Picker**, select the date from which you want to restore events.

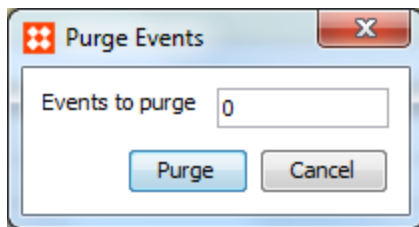
When you have selected a date, the **Events restored** field shows the number of events that will be restored.

Note: Event IDs are not shown in the UI but are read from the database.



Purge Events

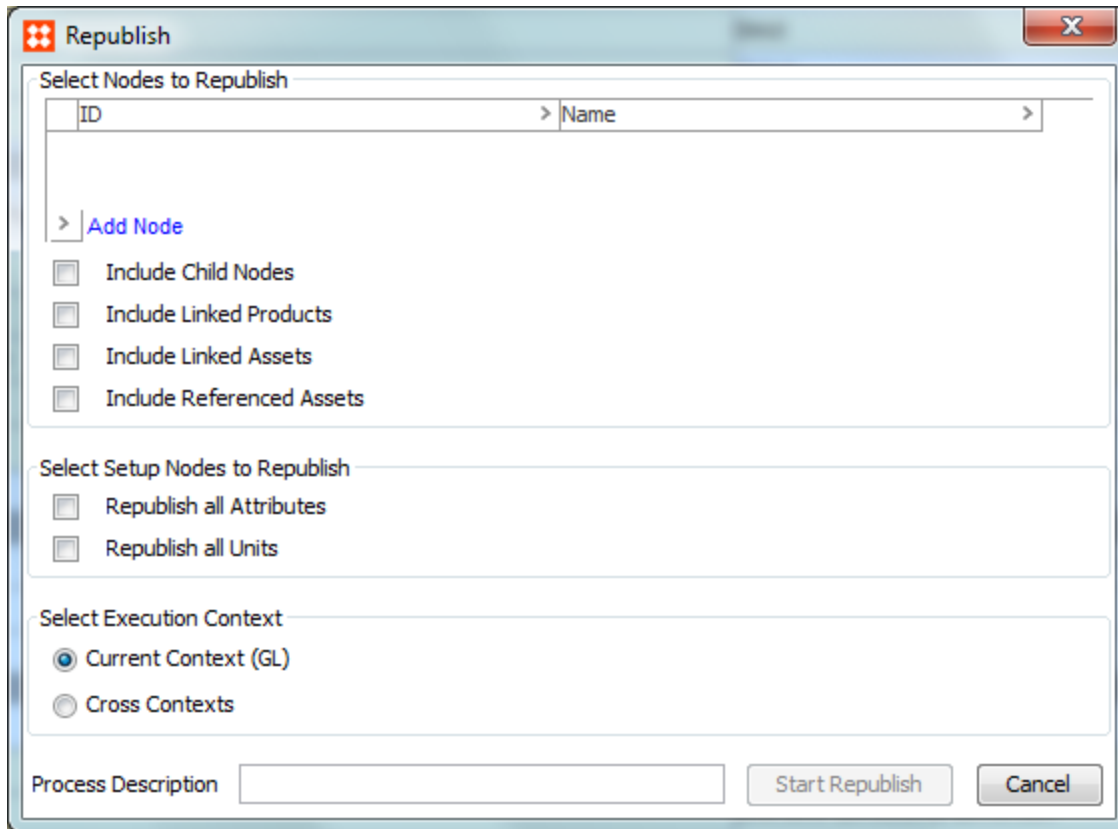
1. In **Event Actions**, click **Purge**.
2. The Purge Events dialog, the number of **read** events that you can delete from the system is displayed.
3. Click **Purge** to delete the events.



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.

1. In **Event Actions**, click **Republish** to display the Republish dialog.

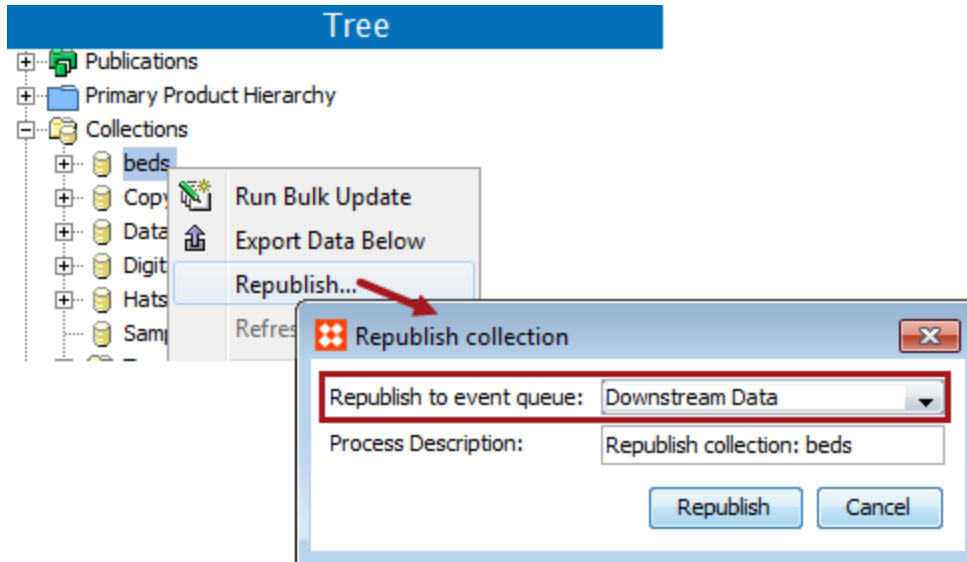


2. Click **Add node** and then browse or search for the hierarchy you want to republish.
3. Select the objects you want to republish. You have the following options:
 - **Include Child Nodes** republishes all children of the selected object.
 - **Include Linked Products** republishes all products linked 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.
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. This is a mandatory field.
7. Click **Start Republish**. Events are triggered on objects that match the selection.

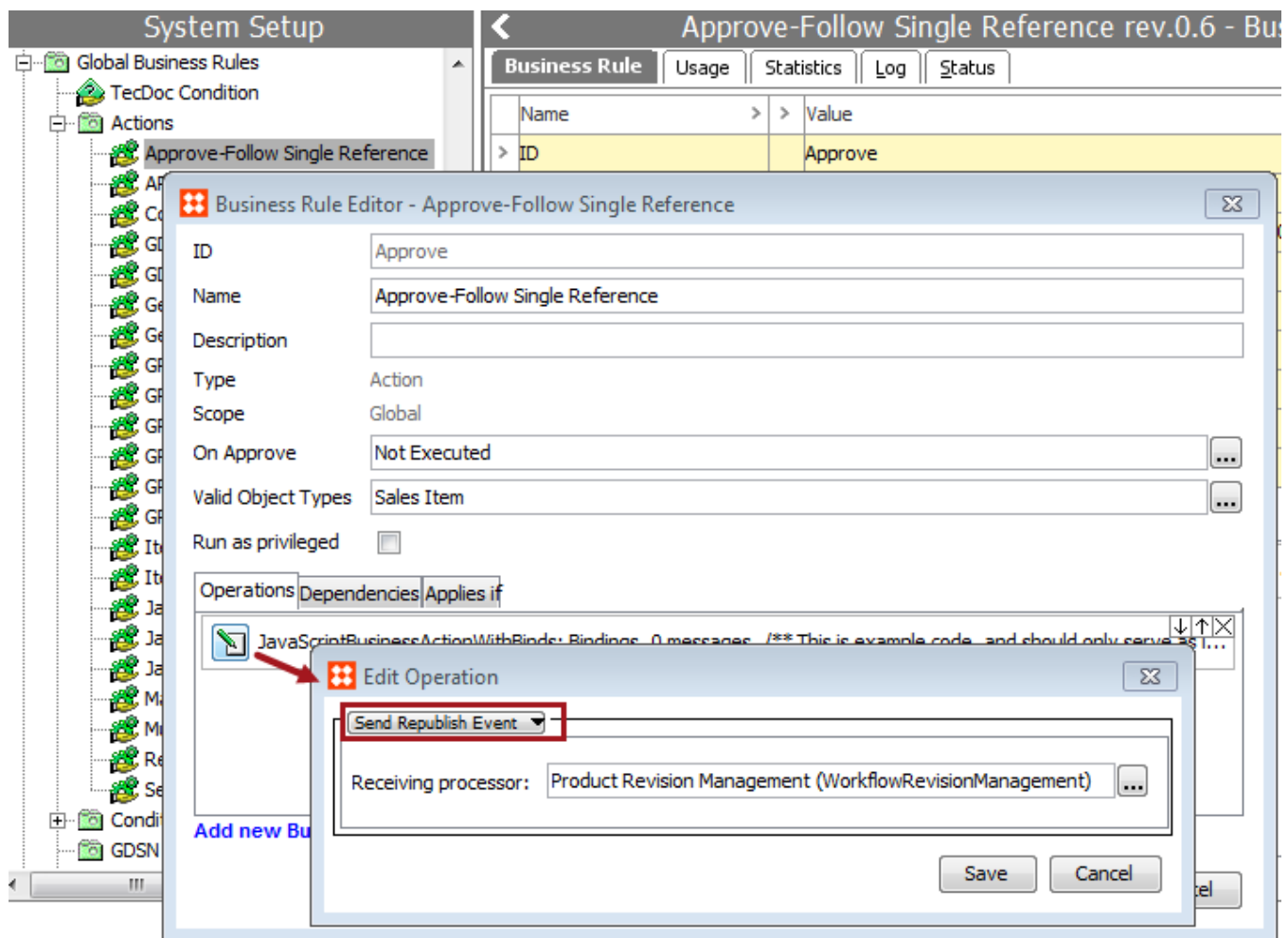
Republish Manually

Additionally, republish events can be generated manually in STEP Workbench as follows:

- From a collection



- Via the Send Republish Event option, which is available from a bulk update and a business action



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.

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. Note that 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.

Outbound Integration Endpoint	Configuration	Event Triggering Definitions
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, and estimate and details are available for review in System Setup.

Estimate Unread Events

An estimate of the current number of unread events in the Event Queue Configuration area 'Unread events (approximated)' parameter.

Click the 'Click to estimate' button to display the data.

> Unread events (approximated) Click to estimate ...

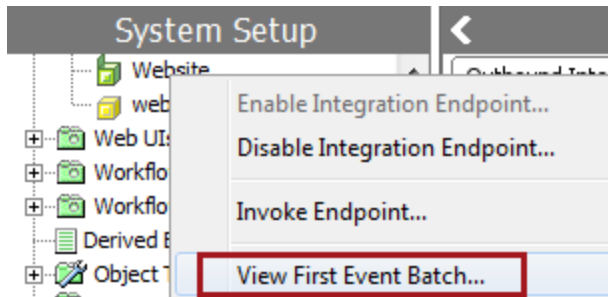
Delete Attribute Values - Configuration	
Configuration	
Event Queue Configuration	
Event Actions: Forward Rewind Purge Republish	
> Days to retain events	0
> Number of events to batch	1000
> Number of event batches to include per delivery	1
> Queue Status	Read Events
> Unread events (approximated)	140 (2016-06-16 14:17:11)

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**.

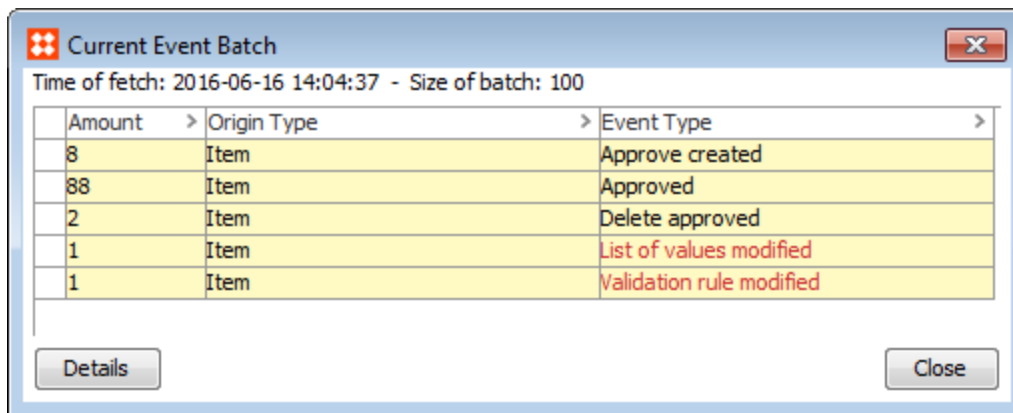
View First Event Batch

This option allows you to view more than just a count of the events.

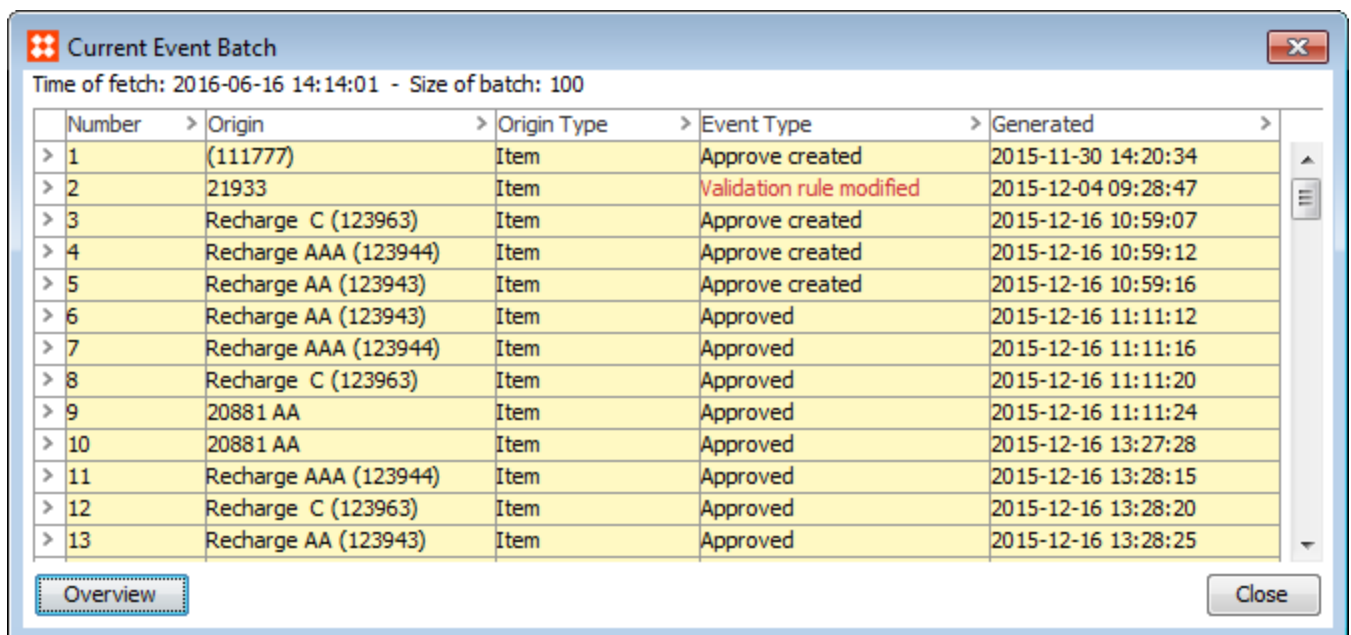
1. Right-click the OIEP and select View First Event Batch.



2. The Current Event Batch dialog displays the first 100 unread events on the queue. Derived events are represented by a number instead of by their ID.



3. For more information about the events, click the **Details** button. 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).

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.

The screenshot shows the 'System Setup' sidebar on the left with a tree view containing items like UnitOIEP, Website, web ssl, Web UIs, Workflow Profiles, Workflows, Derived Events, Object Types & Structures, Tags, Units, Users & Groups, Reference Types, and Workspaces. The main panel is titled 'Website - Configuration' and has tabs for 'Outbound Integration Endpoint', 'Configuration', 'Event Triggering Definitions', and 'Background'. The 'Configuration' tab is active, showing the 'Event Queue Configuration' section. Below this, there are buttons for 'Event Actions': Forward, Rewind, Purge, and Republish. A table lists configuration parameters: 'Days to retain events' (0), 'Number of events to batch' (1000), 'Number of event batches to include per delivery' (1), 'Queue Status' (set to 'Discard Events'), and 'Unread events (approximated)' (Read Events). The 'Queue Status' row is highlighted with a red box, and a dropdown menu is open showing 'Discard Events' and 'Read Events' options.

For information about setting the enabled / disabled parameter, see the **Running an Outbound Integration Endpoint**.