EXPORT MANAGER

USER GUIDE

StiboSystems

STEP Trailblazer

Table of Contents

Exporting Data and Images

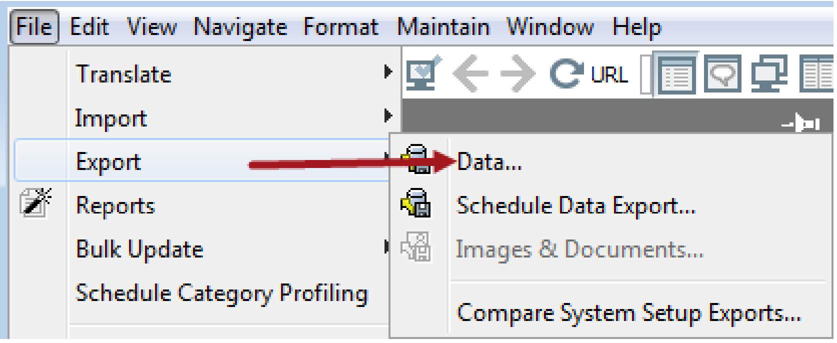
There are three primary methods for exporting data and images from STEP:

- **Export Manager** is used to export fixed data sets on demand or at scheduled intervals, which includes options to:
  - Export Data (objects, classifications, and entities)
  - Export Images & Documents (digital assets)
- **Asset Push** is used to automatically send assets to an external file system upon modification and/or approval. For information on asset push, see the **Asset Push User Guide**.
- **Outbound Integration Endpoints** are used to monitor the system and automatically export relevant data. For more information about Outbound Integration Endpoints, see the **Integration Endpoints for Data Exchange** documentation.

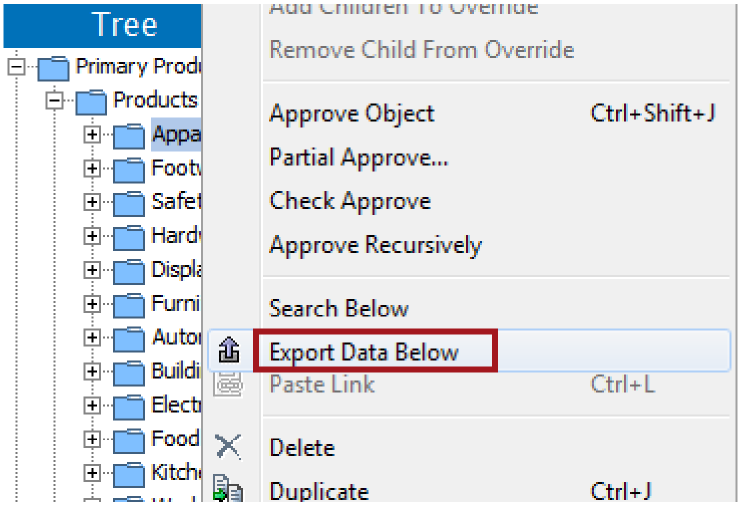
Exporting Data

Select the appropriate context in STEP before launching the Export Wizard. There are multiple ways to start a data export process:

- File menu > **Export** > **Data**



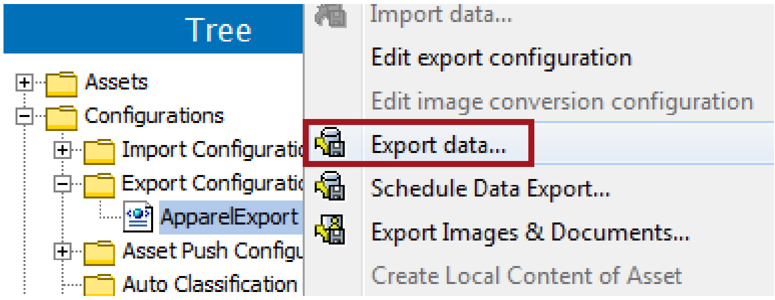
- In the Tree, select a classification, product group, publication, attribute group, or entity. Right-click and then click **Export Data Below**.



Using the Export Data Below option opens the wizard in step 2 with the hierarchy node applied. Depending on the type of hierarchy applied, the wizard automatically selects the type of objects to be exported.

The Export Data Below selection is non-inclusive. This means, for example, that when selecting a product node in the primary product hierarchy, the node selected is *not* included in the output file. However, when selecting an individual product, that has no children, the selected product is output.

- In the Tree, select an export configuration, right-click and then click **Export Data**.


Tree
Assets
Configurations
Import Configuratio
Export Configuratio
ApparelExport
Asset Push Configu
Auto Classification
Import data...
Edit export configuration
Edit image conversion configuration
Export data...
Schedule Data Export...
Export Images & Documents...
Create Local Content of Asset

Data Export Manager Wizard

The Data Export Wizard takes you through up to six steps. If you launched the wizard by using the **Export Data Below** method, the first two screens are skipped.

Step 1 - Select Configuration: This screen allows you to select a previously stored configuration. A configuration is useful for frequently-used exports where the parameters and options are always, or nearly always, the same. Instead of having to redefine your export options every time you do an export, the configuration feature lets you quickly select an appropriate set of options and parameters, and make adjustments if required.

Step 2 - Select Objects: In this screen, you select one or more hierarchy nodes. If you selected a configuration, the objects defined in that configuration are displayed in this screen. It is possible to add and remove objects.

Step 3 - Select Format: In this screen, you select the file format in which the extracted information is delivered.

Step 4 - Map Data: In this screen, you select which pieces of information are extracted per product.

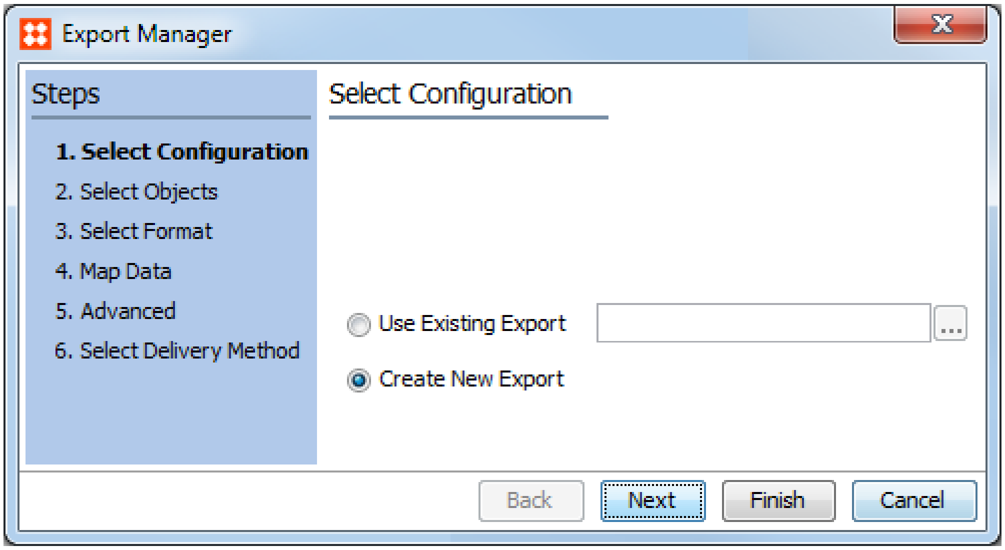
Step 5 - Advanced: In this screen, you specify whether exported data should be converted to another output format.

Step 6 - Select Delivery Method: In this screen, you can specify how information to be delivered.

Step 1 - Select Configuration

1. Select the appropriate context in STEP. All data is exported using the currently selected context.
2. Open the **Export Manager** wizard: File > Export > Data.
3. Select a radio button to use an existing export configuration or create a new export configuration.

- To create a new configuration, select **Create New Export**.



Export Manager

Steps

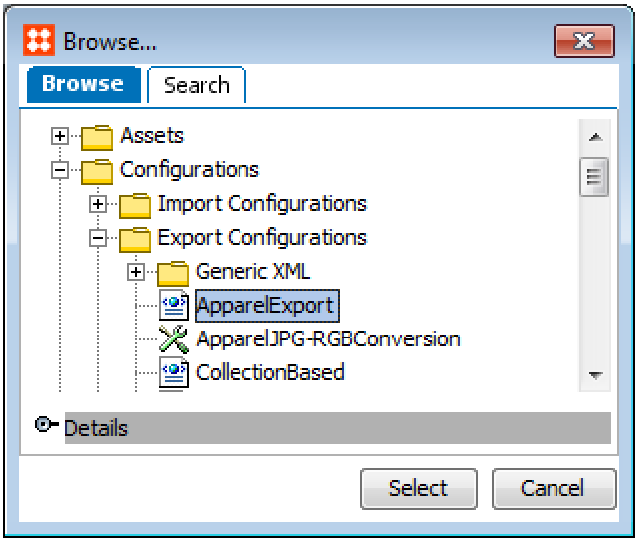
1. Select Configuration
2. Select Objects
3. Select Format
4. Map Data
5. Advanced
6. Select Delivery Method

Select Configuration

○ Use Existing Export

● Create New Export

- To use an existing export configuration, after selecting the Use Existing Export radio button, either **Browse** or **Search** to locate the configuration in STEP.



Browse...

4. Click **Next** to move on to the next step, or to use an existing configuration with no changes, click **Finish**.

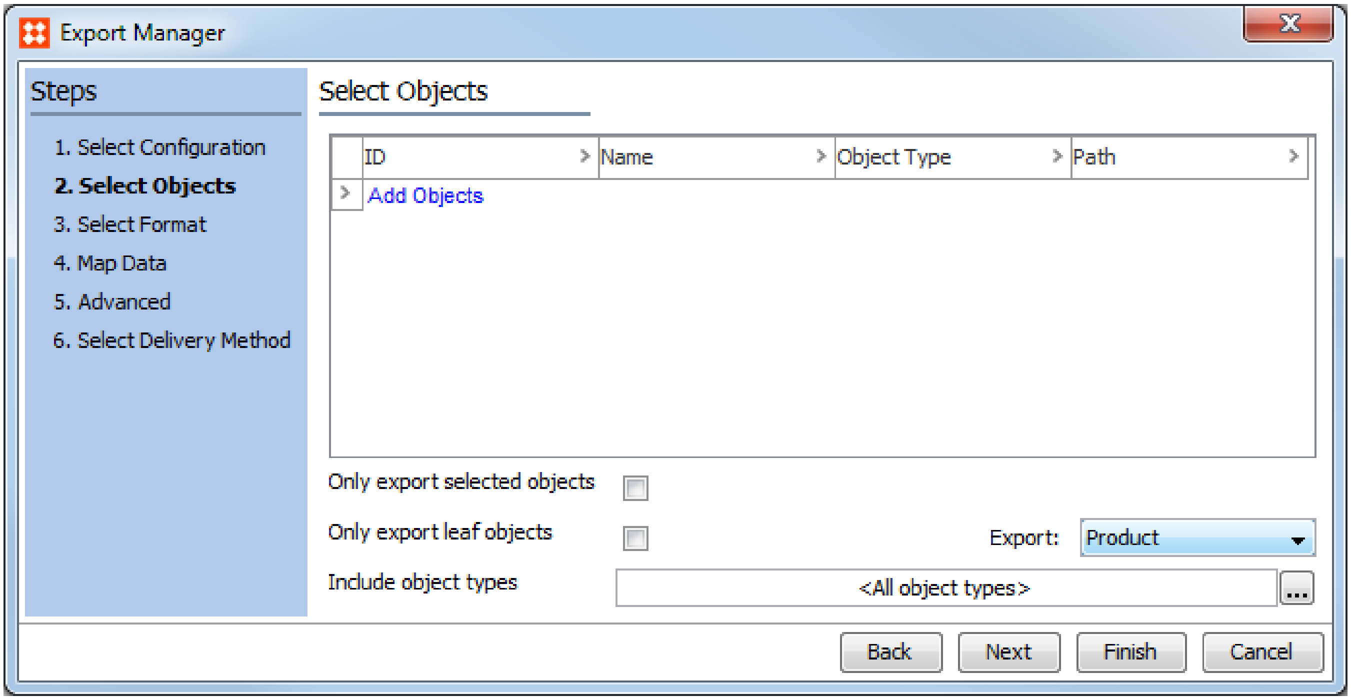
Note: You can create a new configuration by selecting an existing one, changing it according to your current requirements, and then saving the modified setup as a new configuration in the final step of the wizard.

Step 2 - Select Objects

The Export Manager works on all objects below and including the selected root node object. When the Export Manager is started by selecting by a hierarchy node in Tree and using the Export Data Below option, this step is skipped. However, you can use the Back button to return to this step to make changes.

1. Click the **Add Objects** link.

Note: Data can be exported when the Select Object tables is empty, based on the other wizard settings.



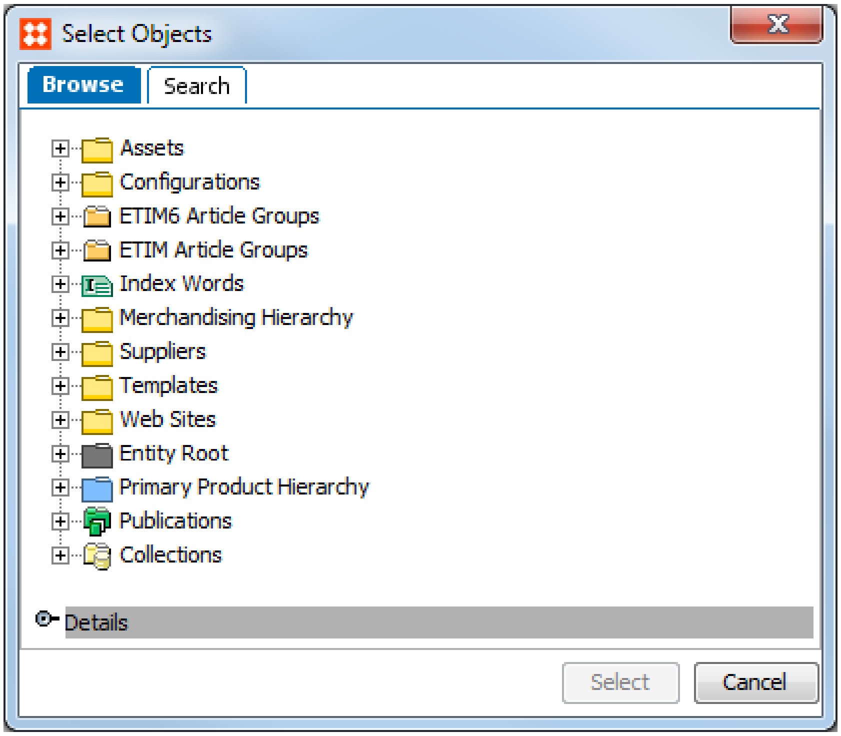
Only export selected objects □

Only export leaf objects □

Export: Product

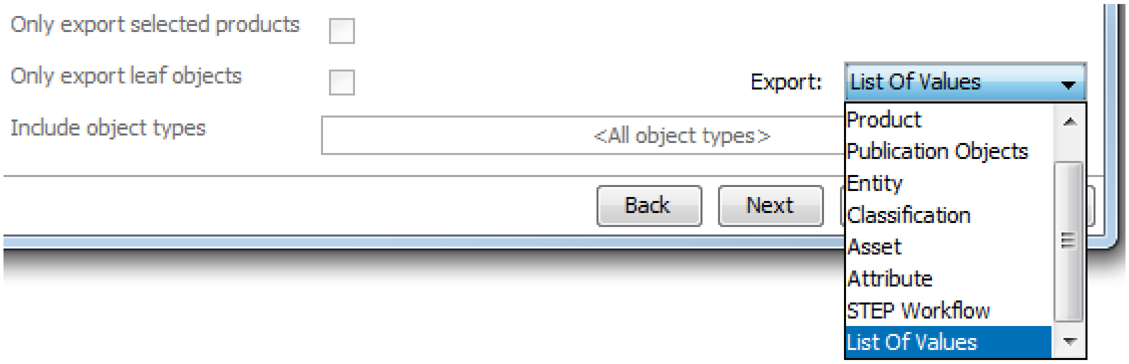
Include object types <All object types>

2. 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, or classification hierarchy, select the root note for the export. Multiple objects can be selected at the same time.



3. Click **Select**. The selected nodes appear in the **Select Objects** window.
4. 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.



Only export selected products □

Only export leaf objects □

Include object types <All object types>

Export: List Of Values

5. Select **Only export selected objects** to export only the root nodes
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.

   This option is ignored when exporting STEPXML format.

7. To limit the export to objects of a specific Object Type, click the ellipsis button **(...)** in the **Include object types** row, and select the object types to be exported. Leave this field blank to include all object types.
8. Click **Next** to move on to the next screen.

Selecting Objects for SmartSheet Export

If you are configuring an Excel SmartSheet export, the following settings apply:

- Do not change the **Export** and **Include object types** default settings.
- For Smartsheet multilevel data export, clear **Only export selected products** and **Only export leaf objects**. Otherwise, only the topmost selection or no products at all are exported.
- For Smartsheet template export, optionally select **Only export selected products** so that the Export Manager does not have to examine the child products of the selected template product.
- Distributing the selected products across different sheets based on value validity can limit potential empty attribute cells.

For Smartsheet template export, note the following:

- 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 selected in the Web UI.
- The selected objects are not exported in Smartsheet templates. However, the first product in the list or selection will be the parent of top-level products created when the edited template is imported.
- Selecting **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.

Recommendations for Selecting Products

We recommend that you review exactly what products you are about to extract and the attributes that you want to extract with them before you open the Export Manager wizard. We also recommended that you test your export settings on a small product line first.

It is unlikely that all the attributes of one product range are also valid for other product ranges. So while the resultant file has columns of attribute data, in many cases one or more of those attributes are relevant to a subset of the products in the file.

There is no way to distinguish between a missing value or a value that is empty only because the attribute is not valid for a particular product. Therefore, in Smartsheets, the selected products might be distributed across different sheets based on value validity.

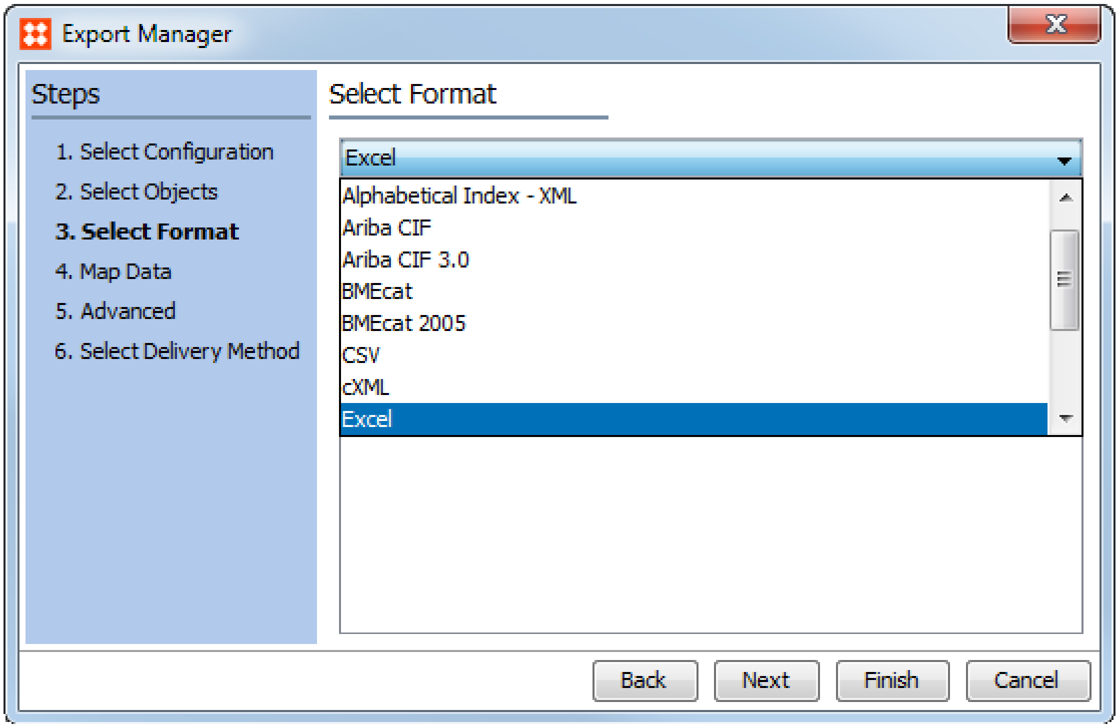
Also, if the file shows certain cells of data as empty, in many cases it will not be apparent that those cells are meant be empty because the attribute is not relevant for that product.

Preparing Object Selection

- 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, include the manufacturer's name as an attribute, and then split the extracted file into split it into separate files for each manufacturer. In this way, your attributes are valid for the products in each file.
- If you know what attributes you want to extract before you launch the Export Wizard, it is easier to select the appropriate attributes when you reach the Data Map screen.

Step 3 - Select Format



The selected option determines the file format of the extracted data.

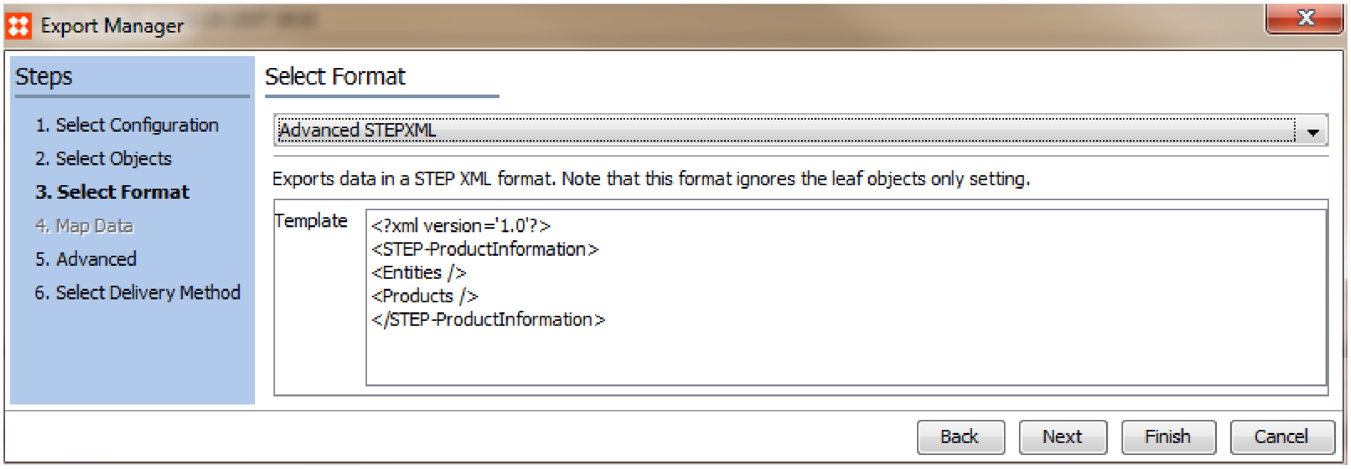
- Advanced STEPXML
- Alphabetical Index - XML is used by customers with the InDesign solution. See **Creating Document Indexes** in the **STEP'n'design** documentation.
- Ariba CIF
- Ariba CIF 3.0
- BMEcat - For more information about the standard, see www.bmecat.org.
- BMEcat 2005
- CSV
- cXML
- Excel
- Excel Smartsheet
- Generic XML
- IDoc MATMAS05

- STEPXML
- xCBL

Exporting Advanced STEPXML Format

Selecting the **Advanced STEPXML** format displays the **Template** field for the data to be exported. The template must be based on the STEPXML XSD.

See the **STEP API Documentation** for a link to the XSD file.



Advanced STEPXML

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

Template

```
<?xml version='1.0'?>
<STEP-ProductInformation>
<Entities />
<Products />
</STEP-ProductInformation>
```

A sample Advanced STEPXML template is shown below. For information about the tags that can be used to output the desired STEP data, see the **Advanced STEPXML** documentation.

```
<STEP-ProductInformation ExportDerivedAttrs="false" FollowOverrideSubProducts="true">
    <DeleteAssets/>
    <DeleteProducts/>
    <DeleteClassifications/>
    <DeleteEntities/>
<Products ExportSize="Selected">
<Product>
   <Name/>
   <ProductCrossReference IncludeInherited="true"/>
<Product/>
<OverrideSubProduct/>
</Product>
</Products>
/STEP-ProductInformation>
```

Advanced STEPXML

Selecting Advanced STEPXML displays a default template in the Select Format window. Advanced STEPXML tags determine how data is exported from STEP. Update or replace the default template with Advanced STEPXML tags to generate the desired output.

The most commonly required output can be generated using the following tags, XML attributes, and setup:

- Attribute Link Tag in STEPXML
- Attribute Value Filtering in STEPXML
- Classification Tag in STEPXML
- Deleting Product References in STEPXML
- Deleting Product, Classifications, and Assets in STEPXML
- Inheriting References in STEPXML
- Inheriting Values in STEPXML
- Product Filtering in STEPXML
- Products Tag in STEPXML
- Reference Filtering in STEPXML
- Referenced and Embedded XML Attributes in STEPXML
- Sequence Product Tag in STEPXML
- STEP Product Information Tag in STEPXML
- Unique Key Tag in STEPXML

Additionally, several STEPXML tags can be used within an Advanced STEPXML template. For more information about the specific tags available, see the **Tags Available for Advanced STEPXML Messages** section.

Attribute Link 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

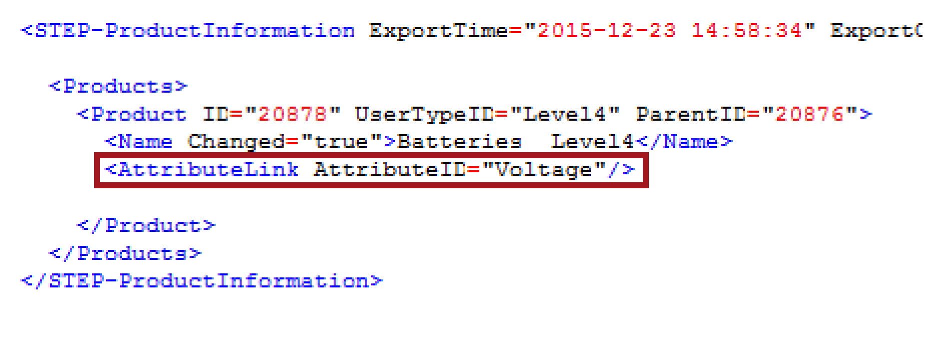
No Inheritance	IncludeInherited

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

Results

Only local attribute links for the exported product are output.

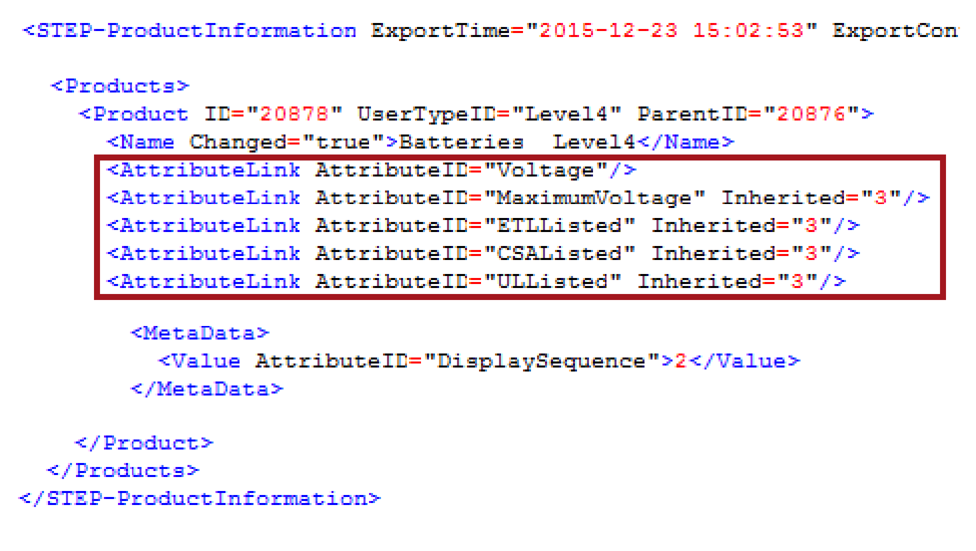
No Inheritance



```
<STEP-ProductInformation ExportTime="2015-12-23 14:58:34" ExportC

  <Products>
    <Product ID="20878" UserTypeID="Level4" ParentID="20876">
      <Name Changed="true">Batteries  Level4</Name>
      <AttributeLink AttributeID="Voltage"/>

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

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

IncludeInherited



```
<STEP-ProductInformation ExportTime="2015-12-23 15:02:53" ExportCon

  <Products>
    <Product ID="20878" UserTypeID="Level4" ParentID="20876">
      <Name Changed="true">Batteries  Level4</Name>
      <AttributeLink AttributeID="Voltage"/>
      <AttributeLink AttributeID="MaximumVoltage" Inherited="3"/>
      <AttributeLink AttributeID="ETLListed" Inherited="3"/>
      <AttributeLink AttributeID="CSAListed" Inherited="3"/>
      <AttributeLink AttributeID="ULListed" Inherited="3"/>

        <MetaData>
          <Value AttributeID="DisplaySequence">2</Value>
        </MetaData>

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

Attribute Value Filtering in STEPXML

Use the **Values** tag to output all the attribute values for a specific object.

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" ExportDerivedAtt
rs="true">

<Products>

    <Product IncludeParent="true">

       <Name/>

       <Values>

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

       </Value>

       <AssetCrossReference Type="Main"/>

       <ProductCrossReference Type="Acccessory"/>

       <ClassificationReference Type="Website"/>

    </Product>

</Products>

</STEP-ProductInformation>
```

Classifications Tag in STEPXML

The Classifications tag can include an XML attribute. When an XML attribute is absent, the default setting is used. For example:

```
<Classifications IncludeParent="true">
```

The following XML attribute is available:

IncludeParent

- False - default setting
- True - when a classification is exported from STEP, the file includes the approved classification as well as its parents.

Additionally, several STEPXML tags can be used within an Advanced STEPXML template. For more information about the specific tags available, see the **STEPXML Tags Available Within Advanced STEPXML Tags** section.

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:

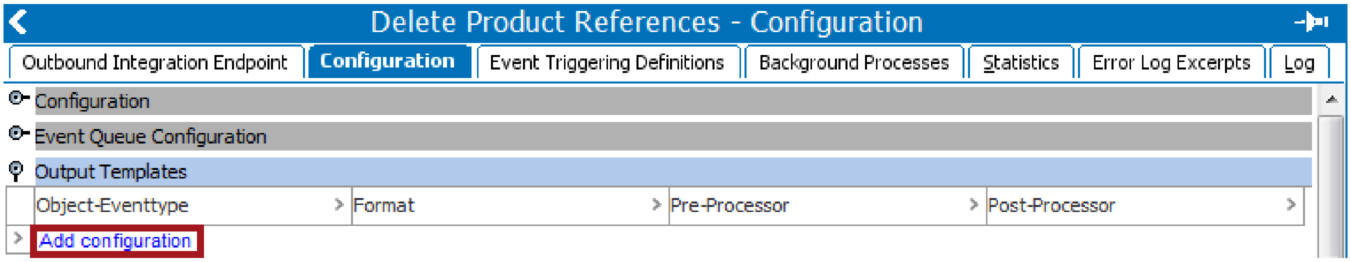
```
<STEP-ProductInformation>
<Products IncludeClassificationOwned="false">
<Product>
</Product>
</Products>
</STEP-ProductInformation>
```

Deleting Product References in STEPXML

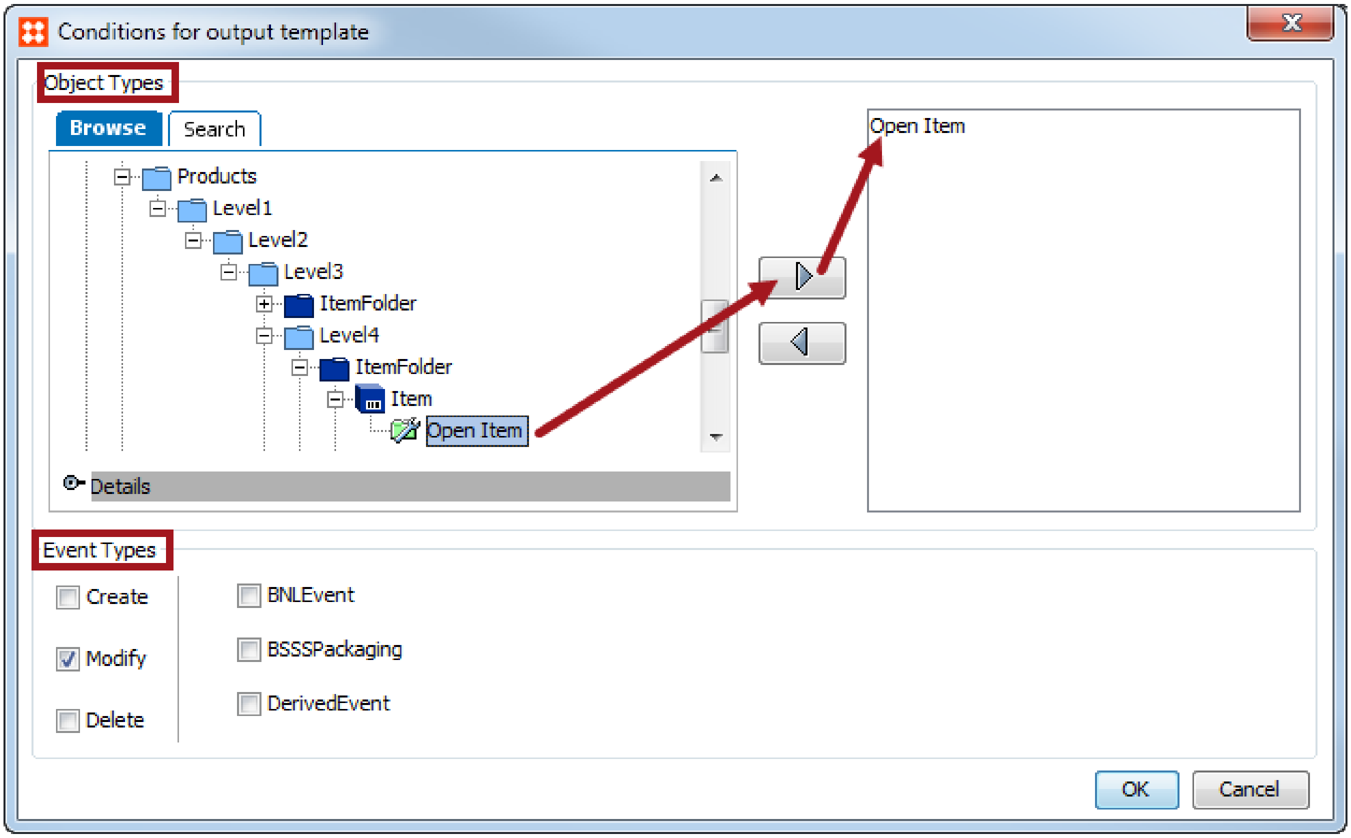
When an Outbound Integration Endpoint (OIEP) is configured to listen for Modify events, use the **STEP-ProductInformation** tag with the **ExportDeletedData** XML attribute to output information about deleted product references. After deleting the reference, approving the product triggers the OIEP.

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.



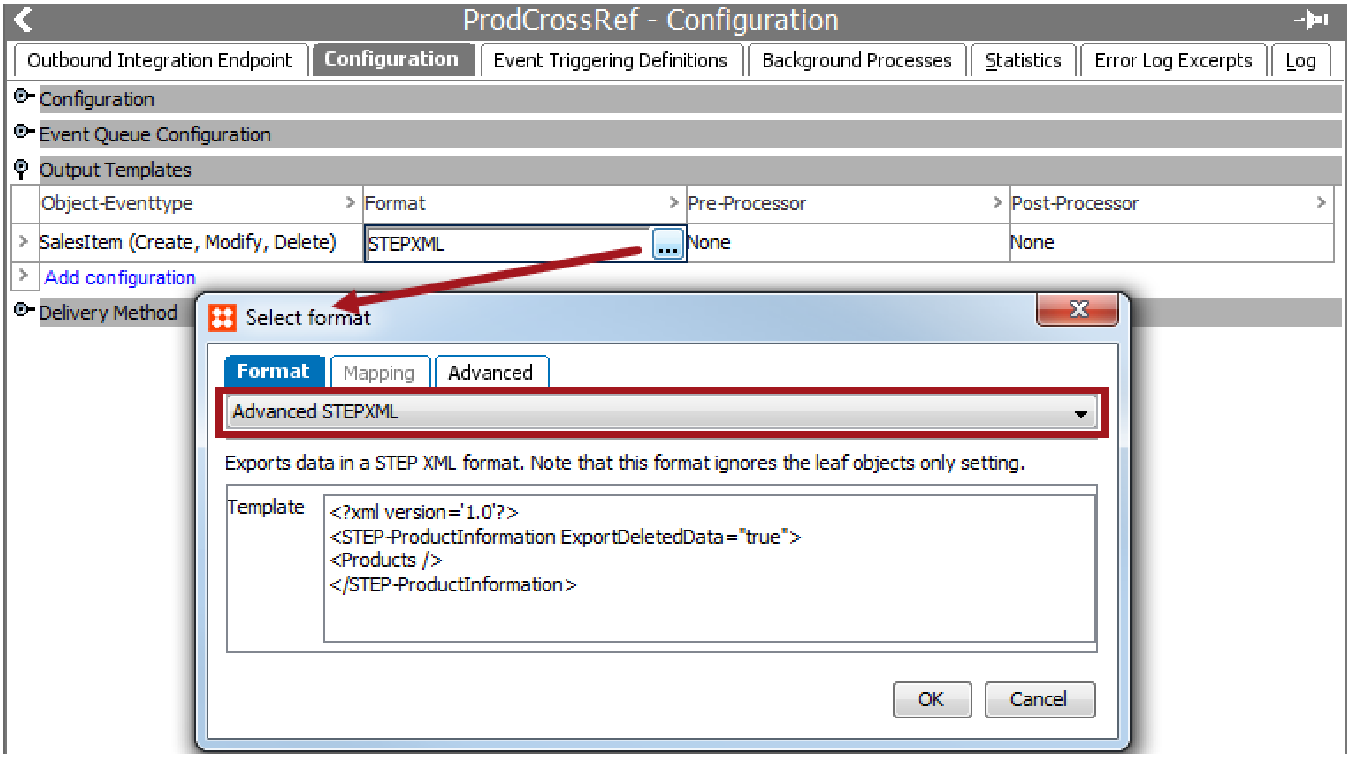
☐ Create
☑ Modify
☐ Delete
☐ BNLEvent
☐ BSSSPackaging
☐ DerivedEvent

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>
```



Object-Eventtype	Format	Pre-Processor	Post-Processor
SalesItem (Create, Modify, Delete)	STEPXML	None	None

Format: Advanced STEPXML

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

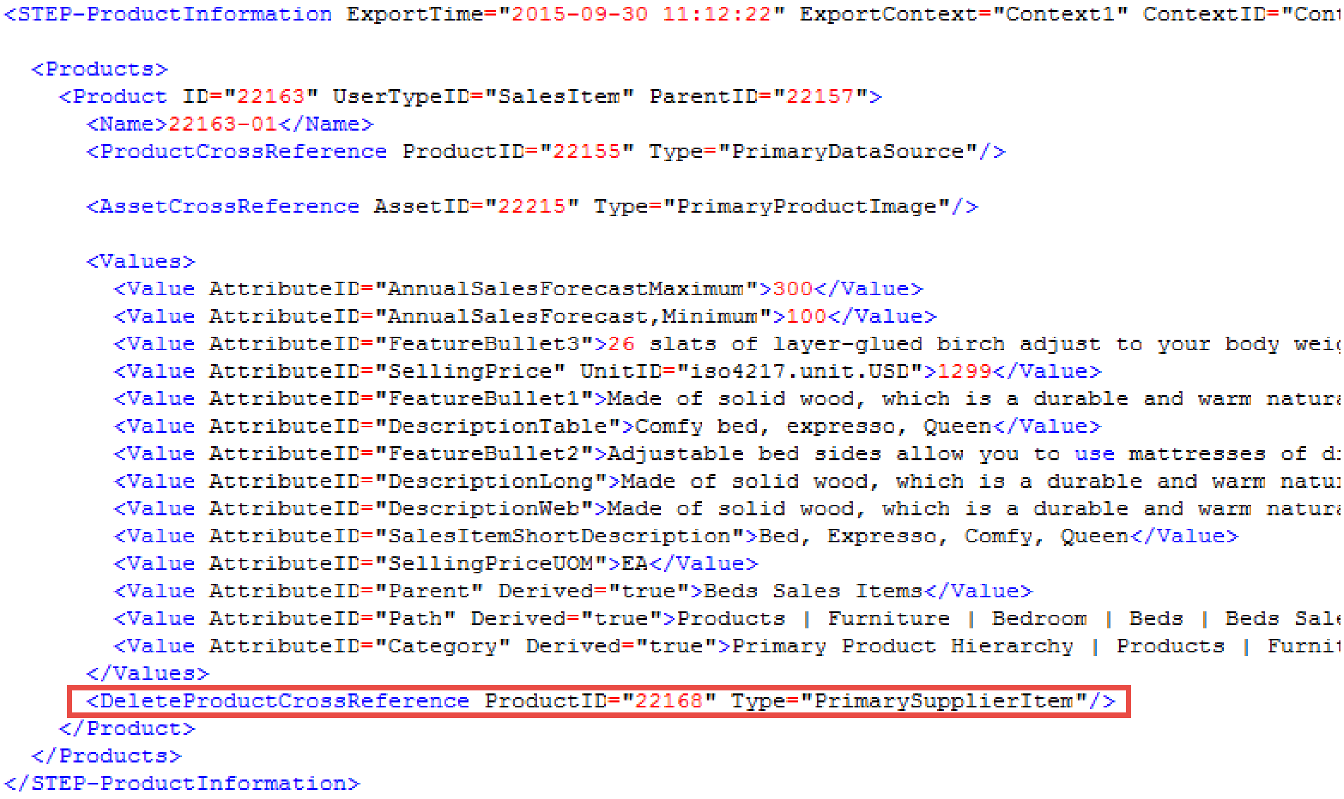
Template

```
<?xml version='1.0'?>
<STEP-ProductInformation 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="Con

  <Products>
    <Product ID="22163" UserTypeID="SalesItem" ParentID="22157">
      <Name>22163-01</Name>
      <ProductCrossReference ProductID="22155" Type="PrimaryDataSource"/>

      <AssetCrossReference AssetID="22215" Type="PrimaryProductImage"/>

      <Values>
        <Value AttributeID="AnnualSalesForecastMaximum">300</Value>
        <Value AttributeID="AnnualSalesForecast,Minimum">100</Value>
        <Value AttributeID="FeatureBullet3">26 slats of layer-glued birch adjust to your body wei
        <Value AttributeID="SellingPrice" UnitID="iso4217.unit.USD">1299</Value>
        <Value AttributeID="FeatureBullet1">Made of solid wood, which is a durable and warm natur
        <Value AttributeID="DescriptionTable">Comfy bed, expresso, Queen</Value>
        <Value AttributeID="FeatureBullet2">Adjustable bed sides allow you to use mattresses of d
        <Value AttributeID="DescriptionLong">Made of solid wood, which is a durable and warm natu
        <Value AttributeID="DescriptionWeb">Made of solid wood, which is a durable and warm natur
        <Value AttributeID="SalesItemShortDescription">Bed, Expresso, 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 Sal
        <Value AttributeID="Category" Derived="true">Primary Product Hierarchy | Products | Furni
      </Values>
      <DeleteProductCrossReference ProductID="22168" Type="PrimarySupplierItem"/>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Deleting Products, Classifications, and Assets in STEPXML

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/>
```

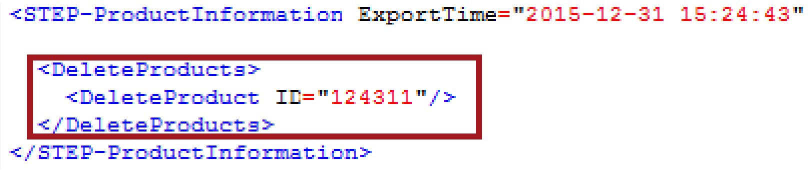
Note: The Force Delete option is not available for products, classifications, or assets when an OIEP is configured to deliver deletion information.

Template

```
<STEP-ProductInformation ExportDeletedData="true">
   <Products>
      <Product>
         <Name/>
      </Product>
   </Products>
<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>
```

Inheriting References in STEPXML

References within STEP are output against the object type where the references are defined. 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" ExportDerivedAtt
rs="true">

<Products>

    <Product IncludeParent="true">

       <Name/>

       <Values/>

       <Values IncludeInheritedAttributes="Level 3"/>

 <AssetCrossReference IncludeInheritedReferences="Level 3"/>  or <AssetCrossReference Inclu
deInherited=‟true"/>

       <ProductCrossReference Type="Acccessory"/>

       <ClassificationReference Type="Website"/>

    </Product>

</Products>

</STEP-ProductInformation>
```

Inheriting Values in STEPXML

Within the **Product** message, the **Values** tag outputs only local values. When no local value is available, even when an inherited value exists, no value is output.

```
<Values/>
```

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

Inherited values are output for each specified object type. Use multiple instances of this tag when duplicate values are required for inherited attributes.

```
<Values IncludeInheritedAttributes="ItemFolder"/>
<Values IncludeInheritedAttributes="Item"/>
```

Template

IncludeInherited	IncludeInheritedAttributes

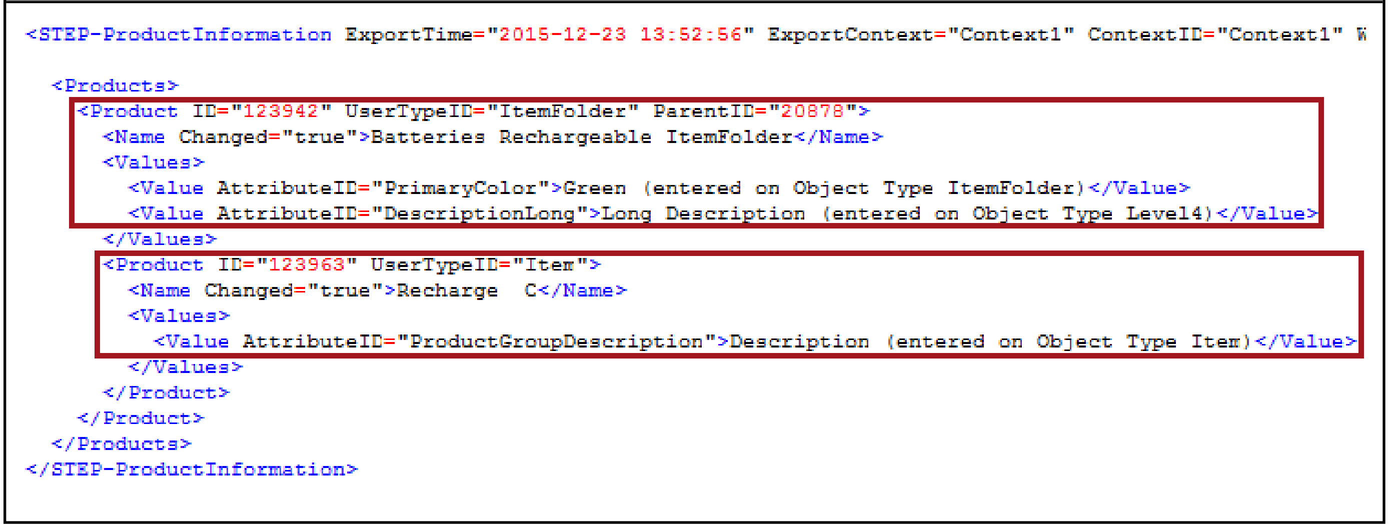
<?xml version="1.0" encoding="UTF-8"?>
<STEP-ProductInformation>
 <Products>
  <Product>
  <Name/>
  <Values **IncludeInherited="true"**/>
 	
  </Product>
 </Products>
</STEP-ProductInformation>
	<?xml version="1.0" encoding="UTF-8"?>
	<STEP-ProductInformation>
	 <Products>
	  <Product>
	  <Name/>
	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

Results

Two (2) levels are exported from the product hierarchy: ItemFolder and Item.

- Local and inherited values are output for the top level (ItemFolder)
- Only local values are displayed for the lower level (Item)

IncludeInherited

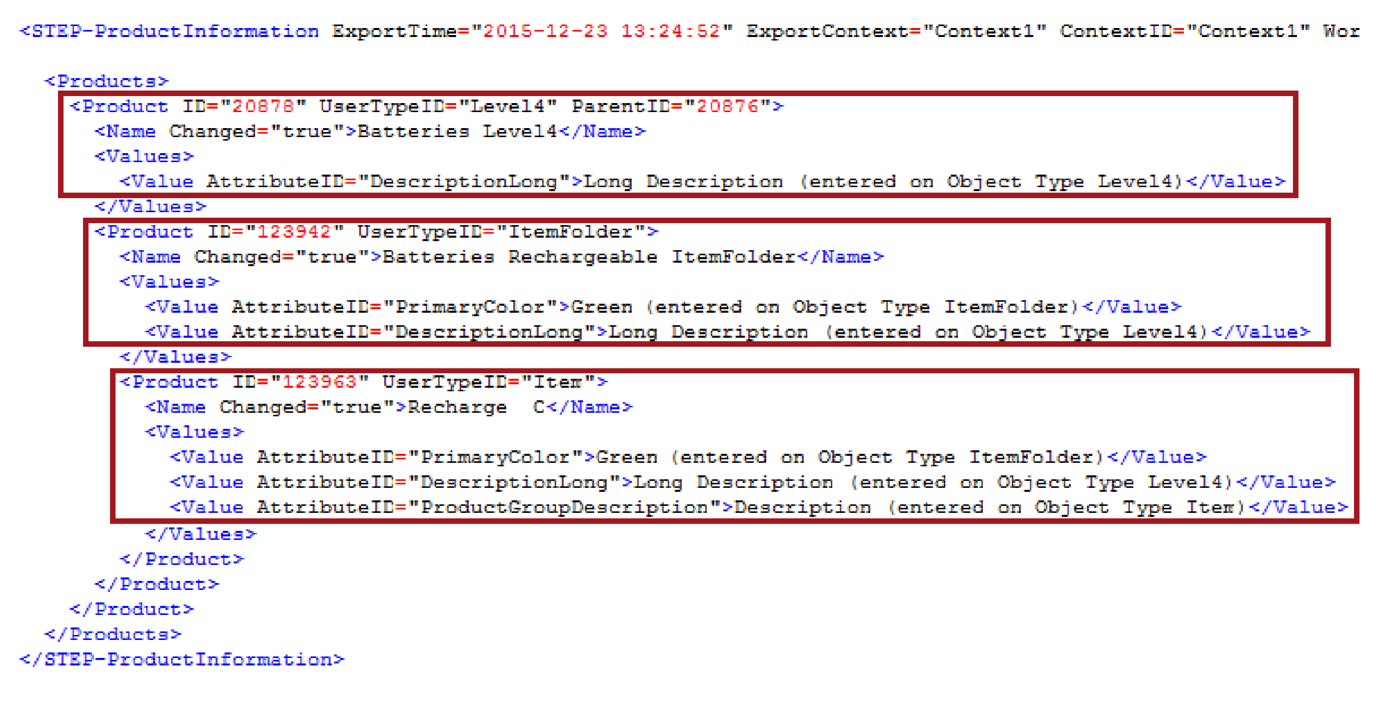


```
<STEP-ProductInformation ExportTime="2015-12-23 13:52:56" ExportContext="Context1" ContextID="Context1" W

  <Products>
    <Product ID="123942" UserTypeID="ItemFolder" ParentID="20878">
      <Name Changed="true">Batteries Rechargeable ItemFolder</Name>
      <Values>
        <Value AttributeID="PrimaryColor">Green (entered on Object Type ItemFolder)</Value>
        <Value AttributeID="DescriptionLong">Long Description (entered on Object Type Level4)</Value>
      </Values>
      <Product ID="123963" UserTypeID="Item">
        <Name Changed="true">Recharge  C</Name>
        <Values>
          <Value AttributeID="ProductGroupDescription">Description (entered on Object Type Item)</Value>
        </Values>
      </Product>
    </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 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 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>
      </Product>
    </Product>
  </Products>
</STEP-ProductInformation>
```

Product Filtering 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.

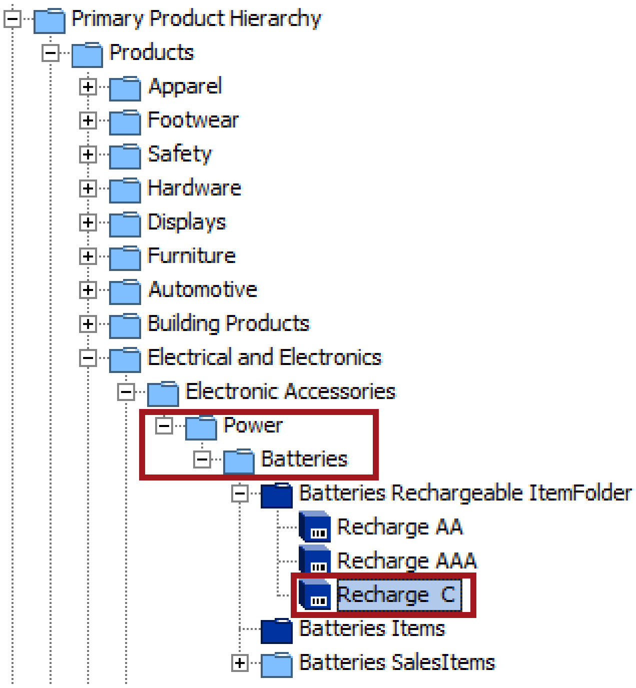
Use multiple instances of the **FilterUserType** tag to include all of the required objects and to filter out the objects that are not needed.

Template

Without Filtering	With Filtering
`<?xml version="1.0" encoding="UTF-8"?>`	`<?xml version="1.0" encoding="UTF-8"?>`
`<STEP-ProductInformation>`	`<STEP-ProductInformation>`
`<Products>`	`<Products>`
	`<FilterUserType ID="Level3"/>`
	`<FilterUserType ID="Level4"/>`
	`<FilterUserType ID="Level5"/>`
`<Product IncludeParent="true">`	`<Product IncludeParent="true">`
`<Name/>`	`<Name/>`
`<ProductCrossReference/>`	`<ProductCrossReference/>`
`</Product>`	`</Product>`
`</Products>`	`</Products>`
`</STEP-ProductInformation>`	`</STEP-ProductInformation>`

Modified Object

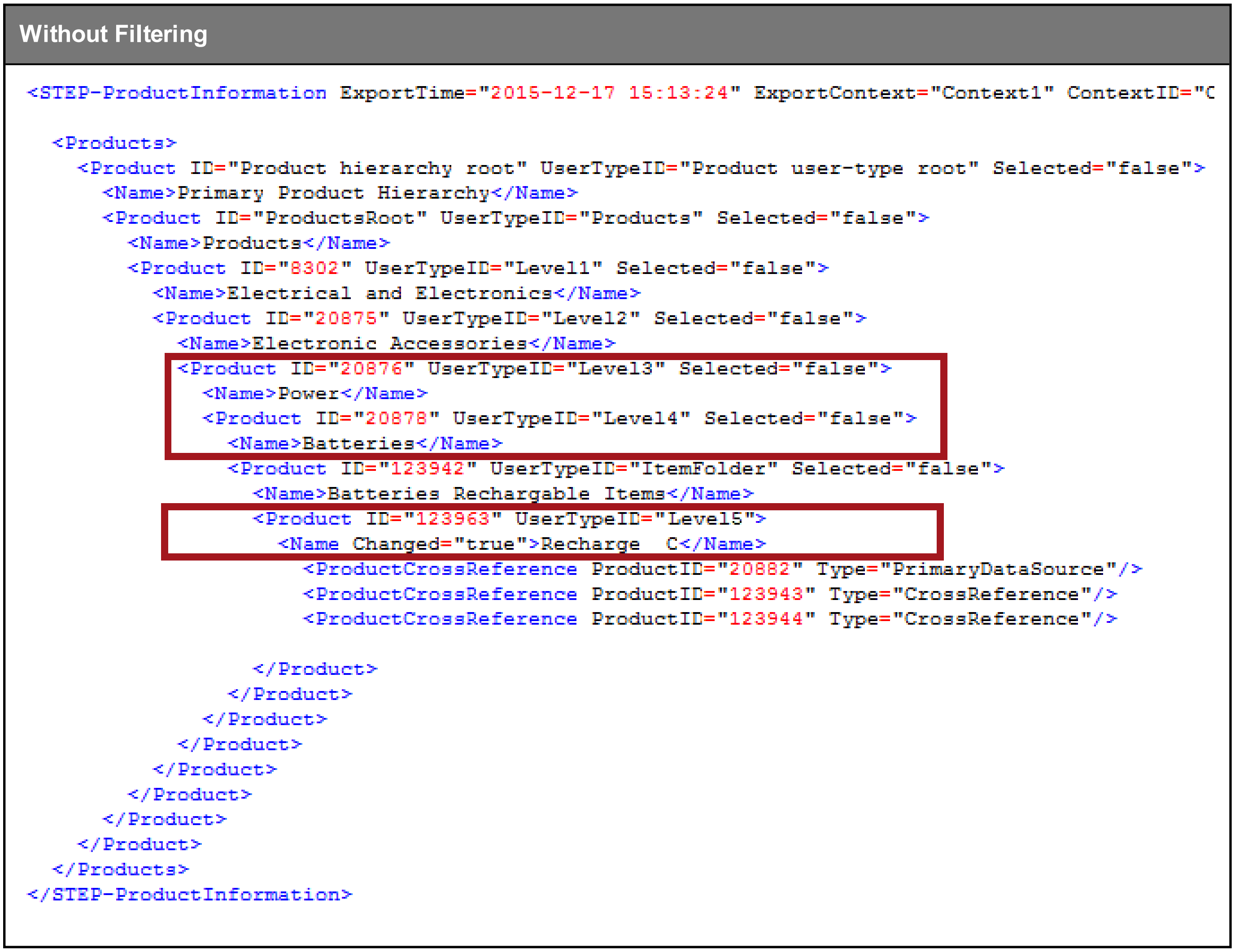
For each template, the same single item (Recharge C) is modified and approved, which could trigger an OIEP.



Primary Product Hierarchy

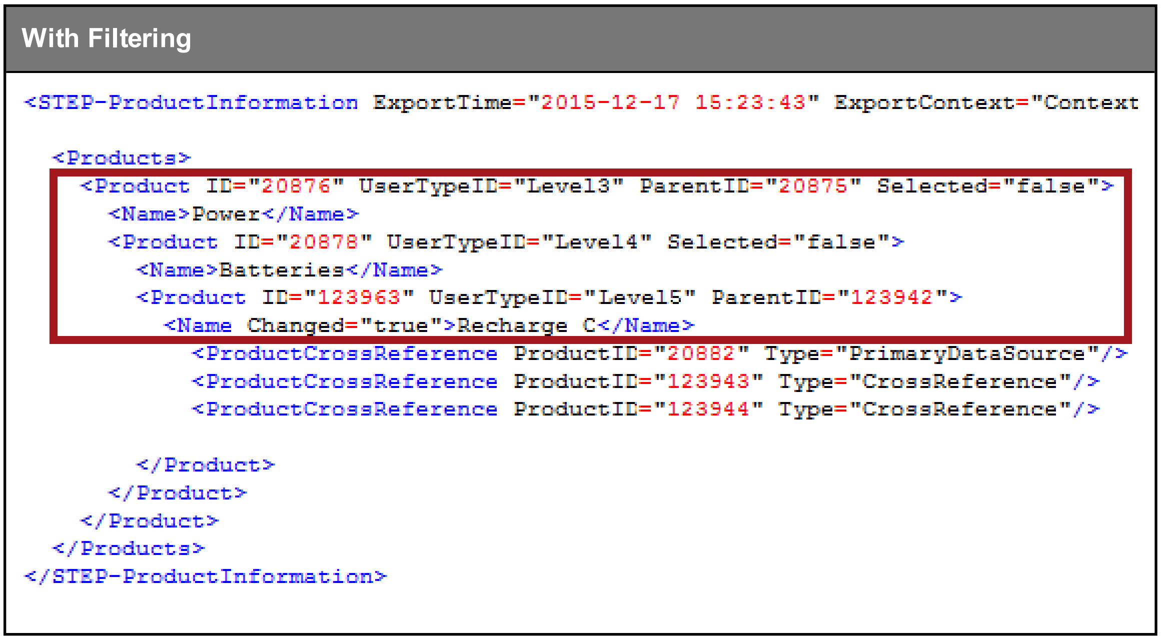
- Products
  - Apparel
  - Footwear
  - Safety
  - Hardware
  - Displays
  - Furniture
  - Automotive
  - Building Products
  - Electrical and Electronics
    - Electronic Accessories
      - Power
        - Batteries
          - Batteries Rechargeable ItemFolder
            - Recharge AA
            - Recharge AAA
            - Recharge C
          - Batteries Items
          - Batteries SalesItems

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



With Filtering

```
<STEP-ProductInformation ExportTime="2015-12-17 15:23:43" ExportContext="Context

  <Products>
    <Product ID="20876" UserTypeID="Level3" ParentID="20875" Selected="false">
      <Name>Power</Name>
      <Product ID="20878" UserTypeID="Level4" Selected="false">
        <Name>Batteries</Name>
        <Product ID="123963" UserTypeID="Level5" ParentID="123942">
          <Name Changed="true">Recharge C</Name>
            <ProductCrossReference ProductID="20882" Type="PrimaryDataSource"/>
            <ProductCrossReference ProductID="123943" Type="CrossReference"/>
            <ProductCrossReference ProductID="123944" Type="CrossReference"/>

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

Products Tag in STEPXML

The Products tag can include multiple options. When an option is absent, the default setting is used.

```
<Products ExportSize="Minimum" IncludeParent="true">
```

The following XML attributes are available:

ExportSize

- Minimum - default setting. Other options are Selected and All.

For details about export size, see the **STEPXML Minimum, Referenced, and Selected** section.

IncludeParent

- False - default setting
- True - when a product is exported from STEP, the file includes the approved product as well as its parents.

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>
```

Reference Filtering 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"/>
```

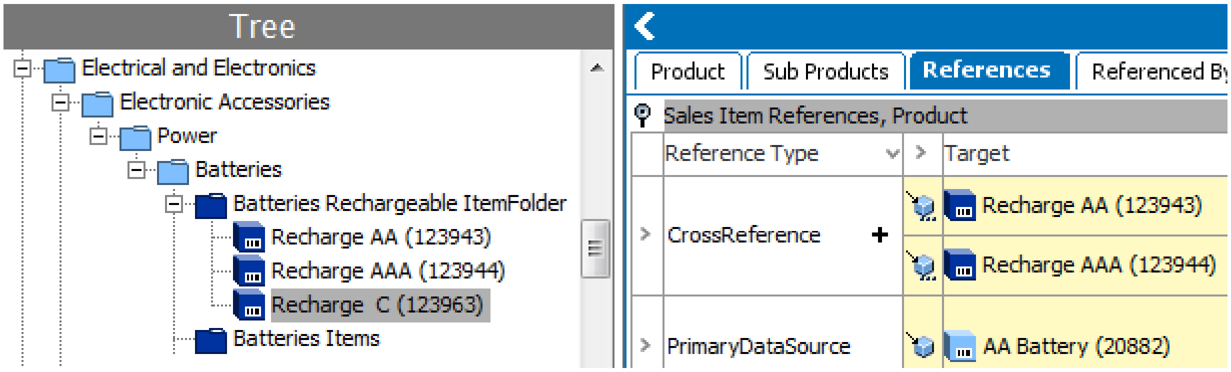
Template

Without Filtering	With Filtering

`<?xml version="1.0" encoding="UTF-8"?>`
`<STEP-ProductInformation>`
` <Products>`
`  <Product>`
`  <Name/>`
`  <ClassificationReference/>`
`  <ProductCrossReference/>`
`  </Product>`
` </Products>`
`</STEP-ProductInformation>`
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<STEP-ProductInformation>`
	` <Products>`
	`  <Product>`
	`  <Name/>`
	`  <ClassificationReference>`
	`  <ProductCrossReference` **`Type="CrossReference"`**`/>`
	`  </Product>`
	` </Products>`
	`</STEP-ProductInformation>`

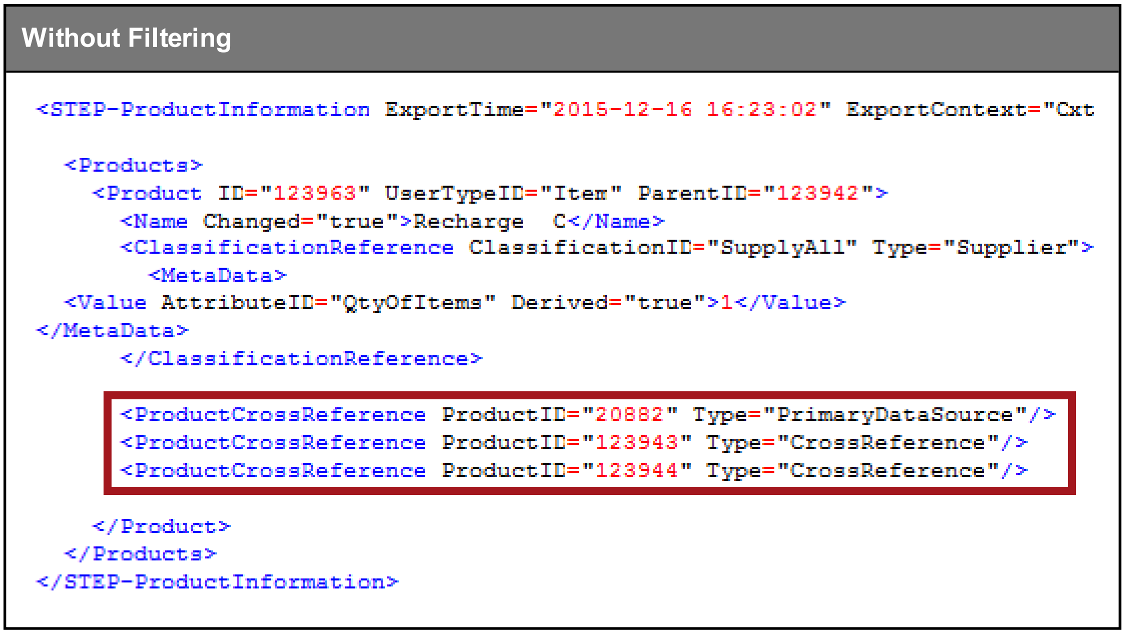
Modified Object

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



Results

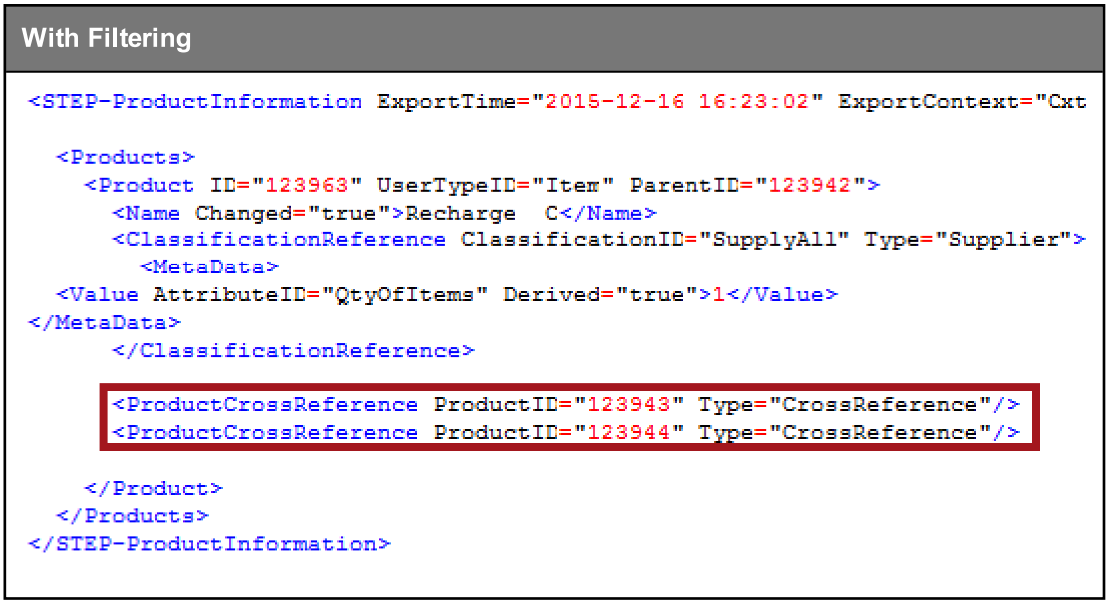
Without filtering, all references for the exported product are output.



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>
```

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">
```

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

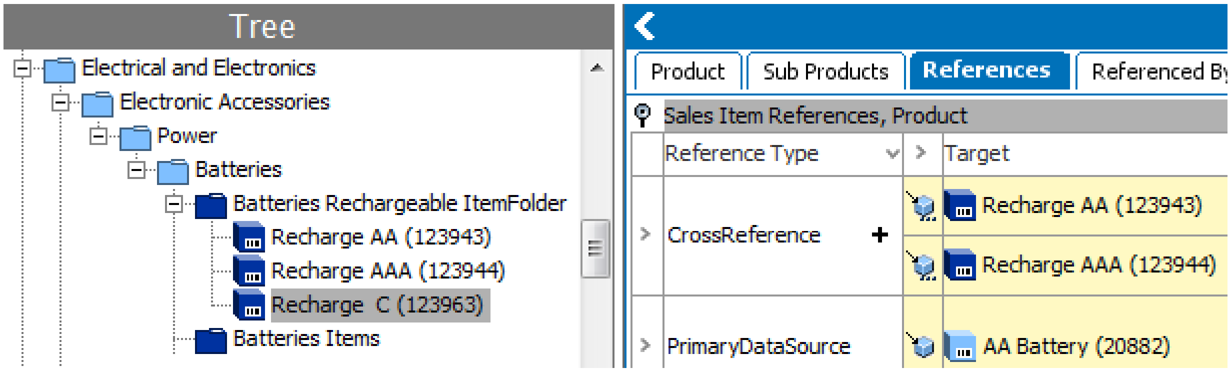
`<?xml version="1.0" encoding="UTF-8"?>`
`<STEP-ProductInformation ResolveInlineRefs="true">`
`<Products>`
`<FilterUserType ID="Item"/>`
`<Product IncludeParent="true">`
`<Name/>`
`<ProductCrossReference Type="CrossReference">`
`<MetaData/>`
`<Product Referenced="true">`
`<Name/>`
`<Values/>`
`</Product>`
`</ProductCrossReference>`
`</Product>`
	`<?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>`

Referenced Without Embedded	Referenced With Embedded

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

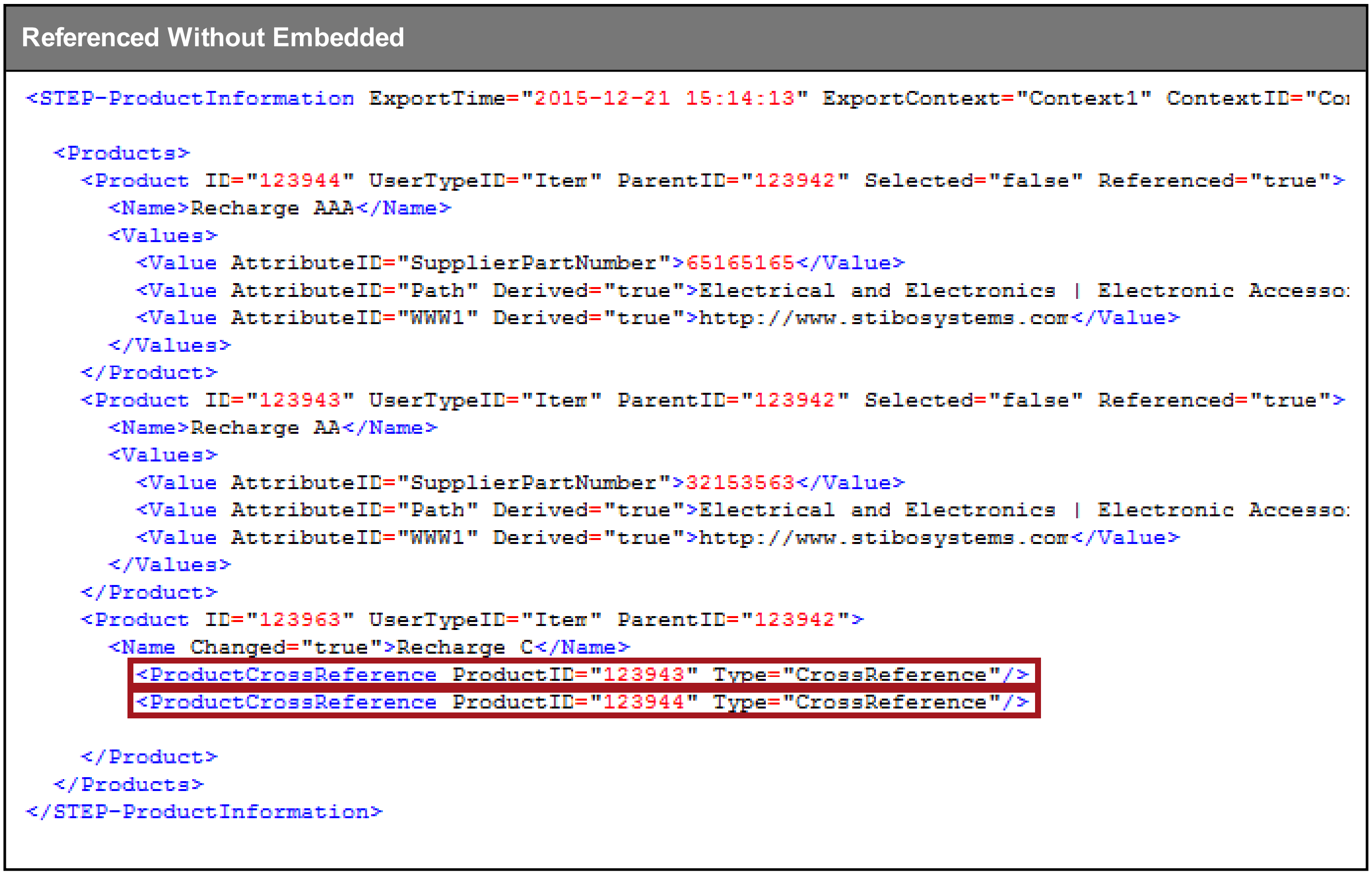
Modified Object

For each template, the same single item (Recharge C) is modified and approved, which triggers the OIEP.



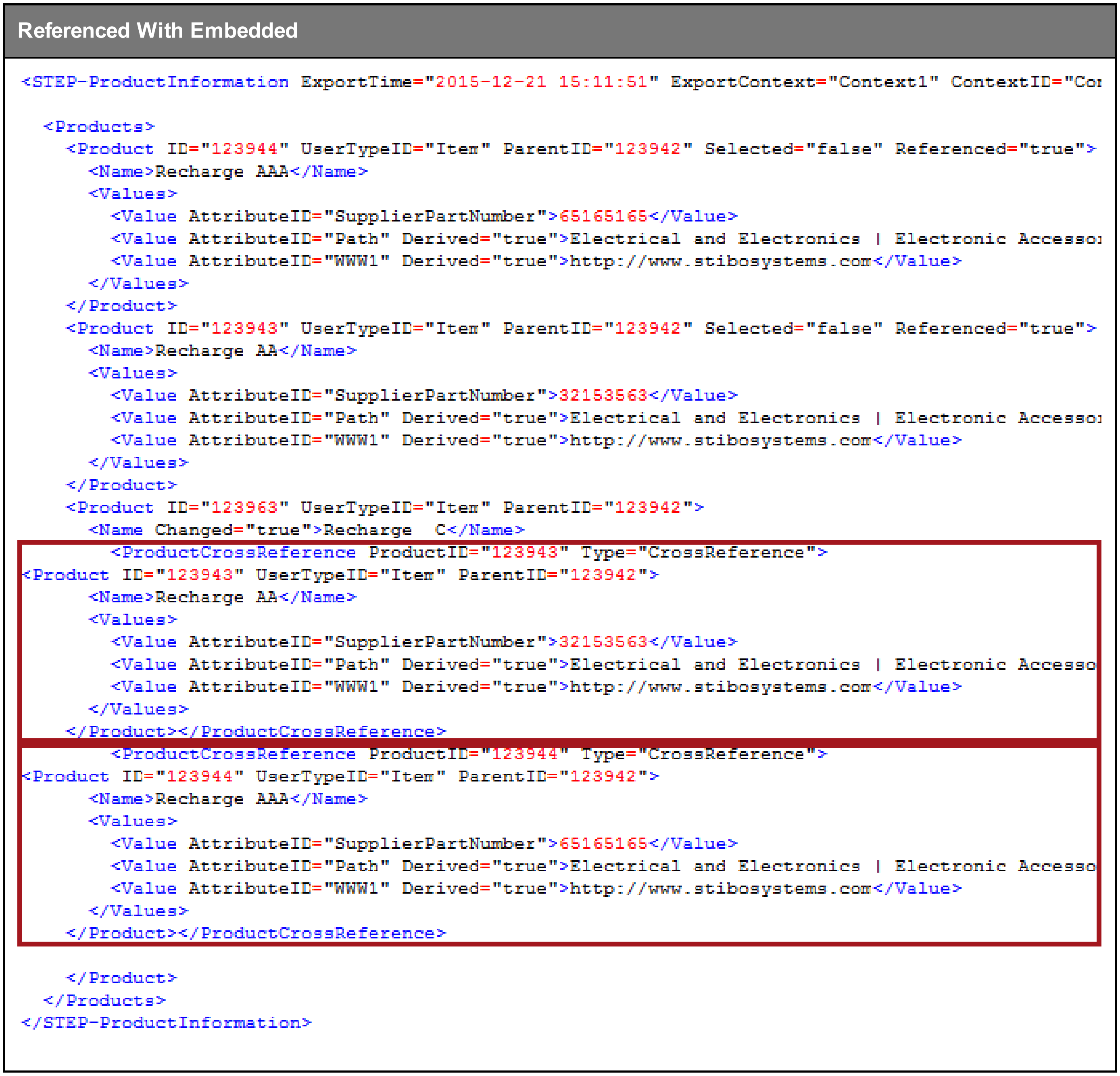
Reference Type	Target
CrossReference	Recharge AA (123943) Recharge AAA (123944)
PrimaryDataSource	AA Battery (20882)

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 Accesso
        <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
      </Values>
    </Product>
    <Product ID="123943" UserTypeID="Item" ParentID="123942" Selected="false" Referenced="true">
      <Name>Recharge AA</Name>
      <Values>
        <Value AttributeID="SupplierPartNumber">32153563</Value>
        <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accesso
        <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
      </Values>
    </Product>
    <Product ID="123963" UserTypeID="Item" ParentID="123942">
      <Name Changed="true">Recharge C</Name>
        <ProductCrossReference ProductID="123943" Type="CrossReference"/>
        <ProductCrossReference ProductID="123944" Type="CrossReference"/>

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

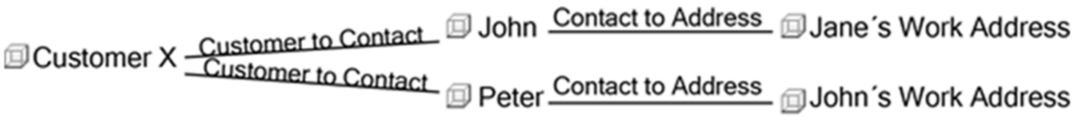


Referenced With Embedded

```
<STEP-ProductInformation ExportTime="2015-12-21 15:11:51" ExportContext="Context1" ContextID="Co

  <Products>
    <Product ID="123944" UserTypeID="Item" ParentID="123942" Selected="false" Referenced="true">
      <Name>Recharge AAA</Name>
      <Values>
        <Value AttributeID="SupplierPartNumber">65165165</Value>
        <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accesso
        <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
      </Values>
    </Product>
    <Product ID="123943" UserTypeID="Item" ParentID="123942" Selected="false" Referenced="true">
      <Name>Recharge AA</Name>
      <Values>
        <Value AttributeID="SupplierPartNumber">32153563</Value>
        <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accesso
        <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
      </Values>
    </Product>
    <Product ID="123963" UserTypeID="Item" ParentID="123942">
      <Name Changed="true">Recharge  C</Name>
        <ProductCrossReference ProductID="123943" Type="CrossReference">
<Product ID="123943" UserTypeID="Item" ParentID="123942">
      <Name>Recharge AA</Name>
      <Values>
        <Value AttributeID="SupplierPartNumber">32153563</Value>
        <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accesso
        <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
      </Values>
    </Product></ProductCrossReference>
        <ProductCrossReference ProductID="123944" Type="CrossReference">
<Product ID="123944" UserTypeID="Item" ParentID="123942">
      <Name>Recharge AAA</Name>
      <Values>
        <Value AttributeID="SupplierPartNumber">65165165</Value>
        <Value AttributeID="Path" Derived="true">Electrical and Electronics | Electronic Accesso
        <Value AttributeID="WWW1" Derived="true">http://www.stibosystems.com</Value>
      </Values>
    </Product></ProductCrossReference>

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

Additionally, several STEPXML tags can be used within an Advanced STEPXML template. For more information about the specific tags available, see the **STEPXML Tags Available Within Advanced STEPXML Tags** section.

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.



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>
```

Sequence Product 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.

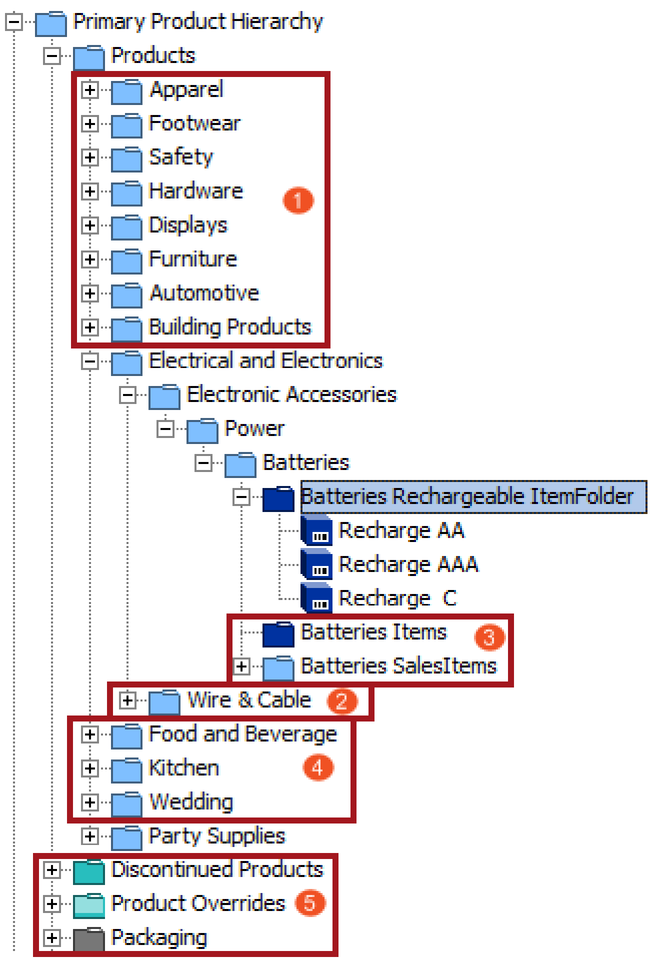
Template

With SequenceProduct	Without SequenceProduct

`<?xml version="1.0" encoding="UTF-8"?>`
`<STEP-ProductInformation>`
`<Products>`
`   <Product IncludeParent="true">`
` <SequenceProduct/>`
`   <Name/>`
`   </Product>`
`</Products>`
`</STEP-ProductInformation>`
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<STEP-ProductInformation>`
	`<Products>`
	`   <Product IncludeParent="true">`
	
	`   <Name/>`
	`   </Product>`
	`</Products>`
	`</STEP-ProductInformation>`

Modified Object

For each template, the same single item (Batteries Rechargeable ItemFolder) is modified and approved, which triggers the OIEP.



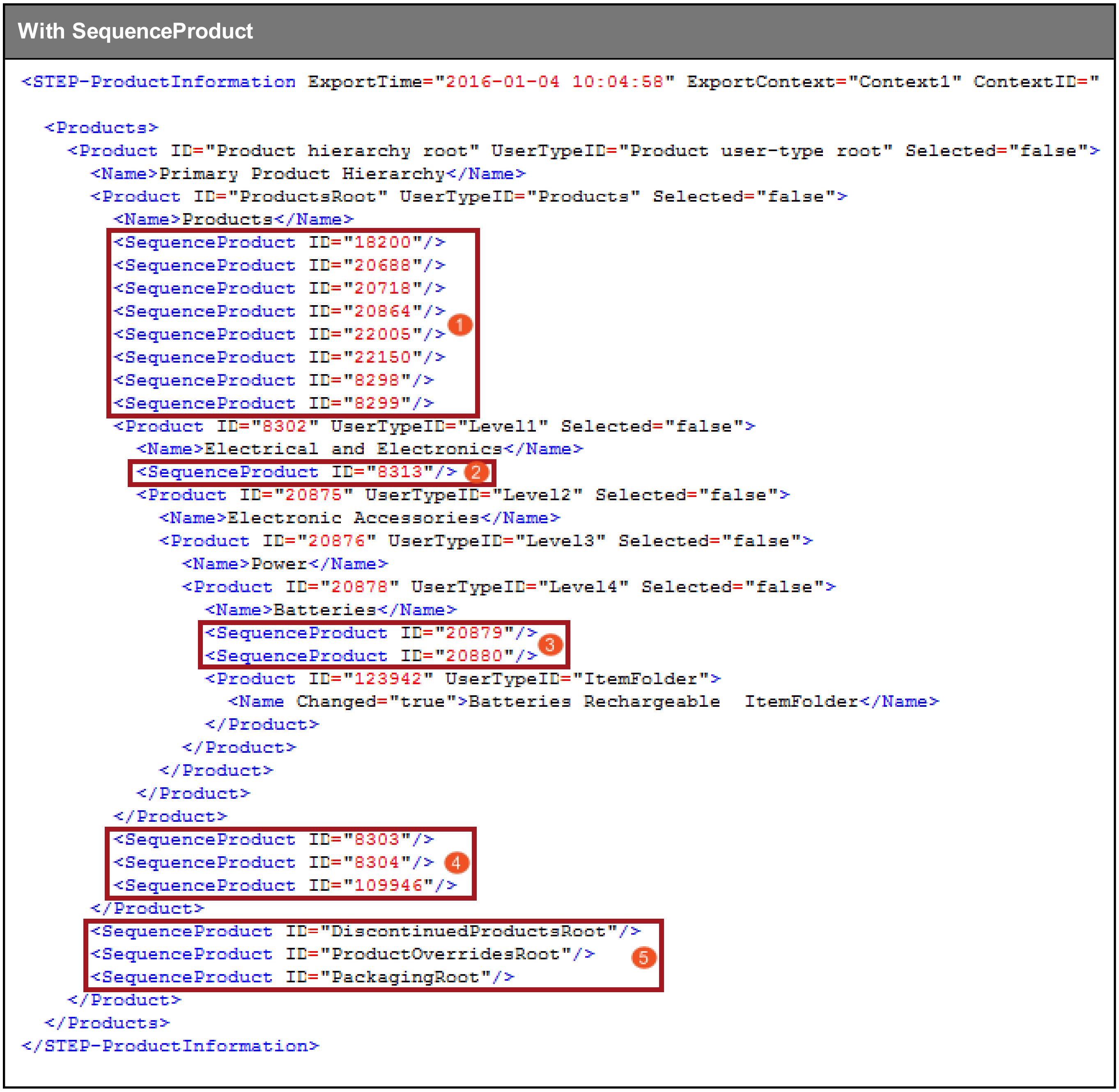
- Primary Product Hierarchy
  - Products
    - Apparel
    - Footwear
    - Safety
    - Hardware
    - Displays
    - Furniture
    - Automotive
    - Building Products
    - Electrical and Electronics
      - Electronic Accessories
        - Power
          - Batteries
            - Batteries Rechargeable ItemFolder
              - Recharge AA
              - Recharge AAA
              - Recharge C
            - Batteries Items
            - Batteries SalesItems
      - Wire & Cable
    - Food and Beverage
    - Kitchen
    - Wedding
    - Party Supplies
  - Discontinued Products
  - Product Overrides
  - Packaging

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.

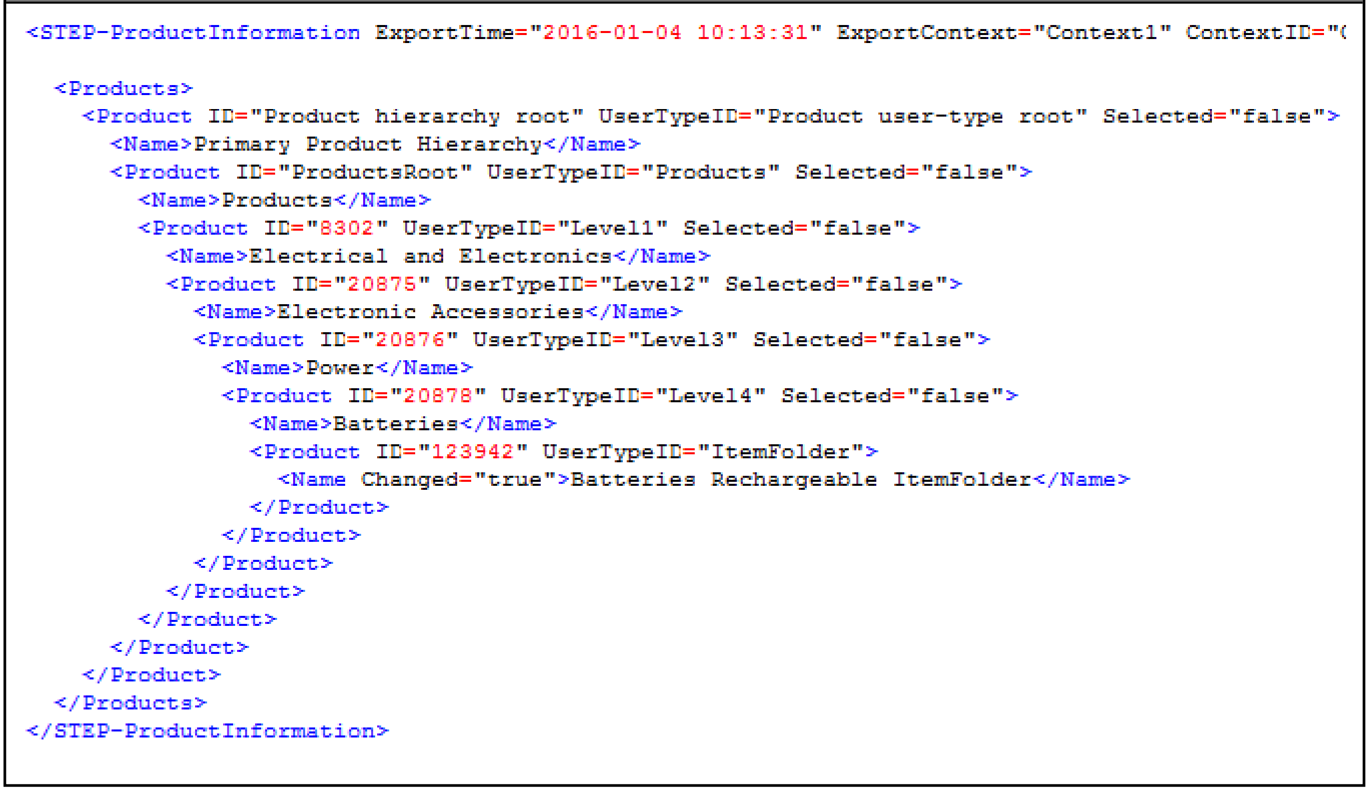


With SequenceProduct

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

STEP Product Information 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
- 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 XSD and DTD.

Tags Available for Advanced STEPXML Messages

You can use the following tags within a product, classification, or asset Advanced STEPXML message to output specific object data.

Tag	Output	Valid Parent tag
`<Name/>`	object name	• Product • Classification • Asset
`<Values/>`	attribute values	• Product • Asset
`<AssetCrossReference/>`	all asset cross references	• Product
`<ProductCrossReference/>`	all product cross references	• Product
`<ClassificationReference/>`	all classification cross references	• Product • Classification
`<MetaData/>`	metadata	• Classification • Asset

For more information about cross reference output, see the **Filtering References** section.

For more information about values output, see the **Inheriting Values** section.

Unique Key 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.

Unique keys are especially used when STEPXML is delivered to external systems like ERP systems , websites, and so on. These systems do not know 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 template is defined to export unique key IDs 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:

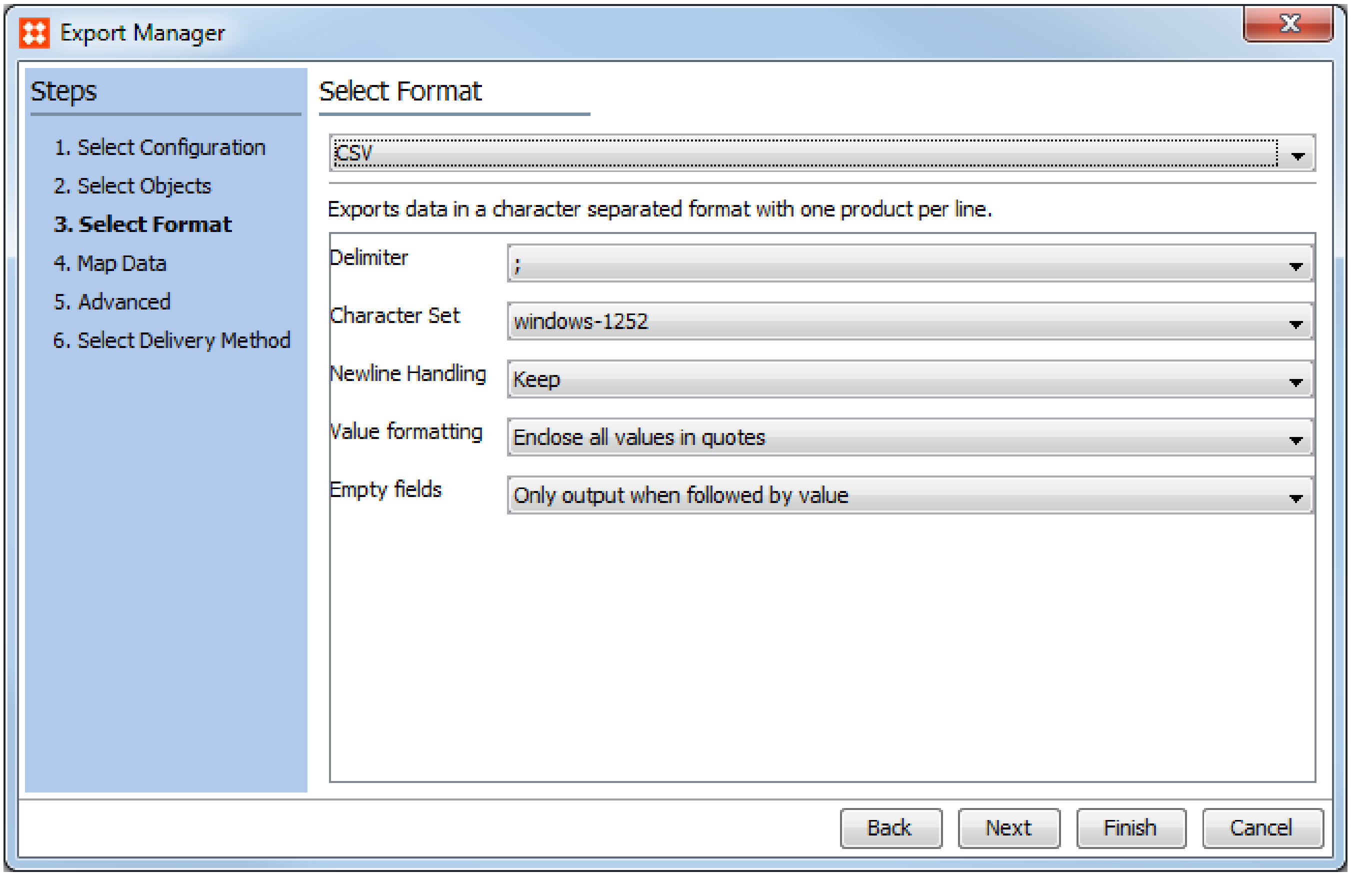
```
<STEP-ProductInformation ExportDerivedAttrs="false" FollowOverrideSubProducts="true">
<IncludeKey KeyID="SAP_ID"/>
<Products ExportSize="Minimum">
<Product>
   <Name/>
   <AttributeLink/>
   <ClassificationReference/>
   <ProductCrossReference/>
   <AssetCrossReference/>
   <Values/>
</Product>
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

Excel or CSV Format

- If you select Excel, you must also select the relevant version: Excel 97/2003 (limited to 65,536 rows) or Excel 2007 (limited to 1,048,576 rows). The character set used to export the data is UTF-8.
- If you select CSV, then you have a number of options:



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

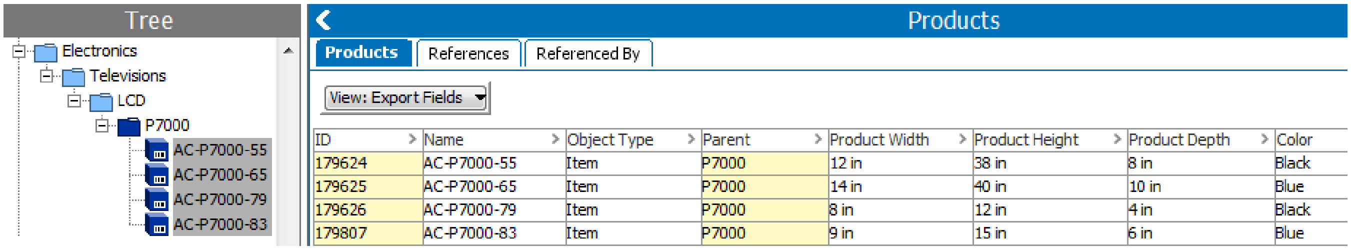
Empty fields: Only output when followed by value

1. In **Delimiter**, select the character to use as delimiter: semi-colon (;), comma (,), the tab character, the pipe symbol (|), or colon (:). The recommended setting is **tab**.
2. In **Character Set**, select the desired character set: 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.
3. In **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.

4. In **Value formatting**, select either 'Enclose all values in quotes' or 'Export without quotes unless value contains delimiter character(s)'.
5. In **Empty fields** select 'Only output when followed by value' or 'Always output'.

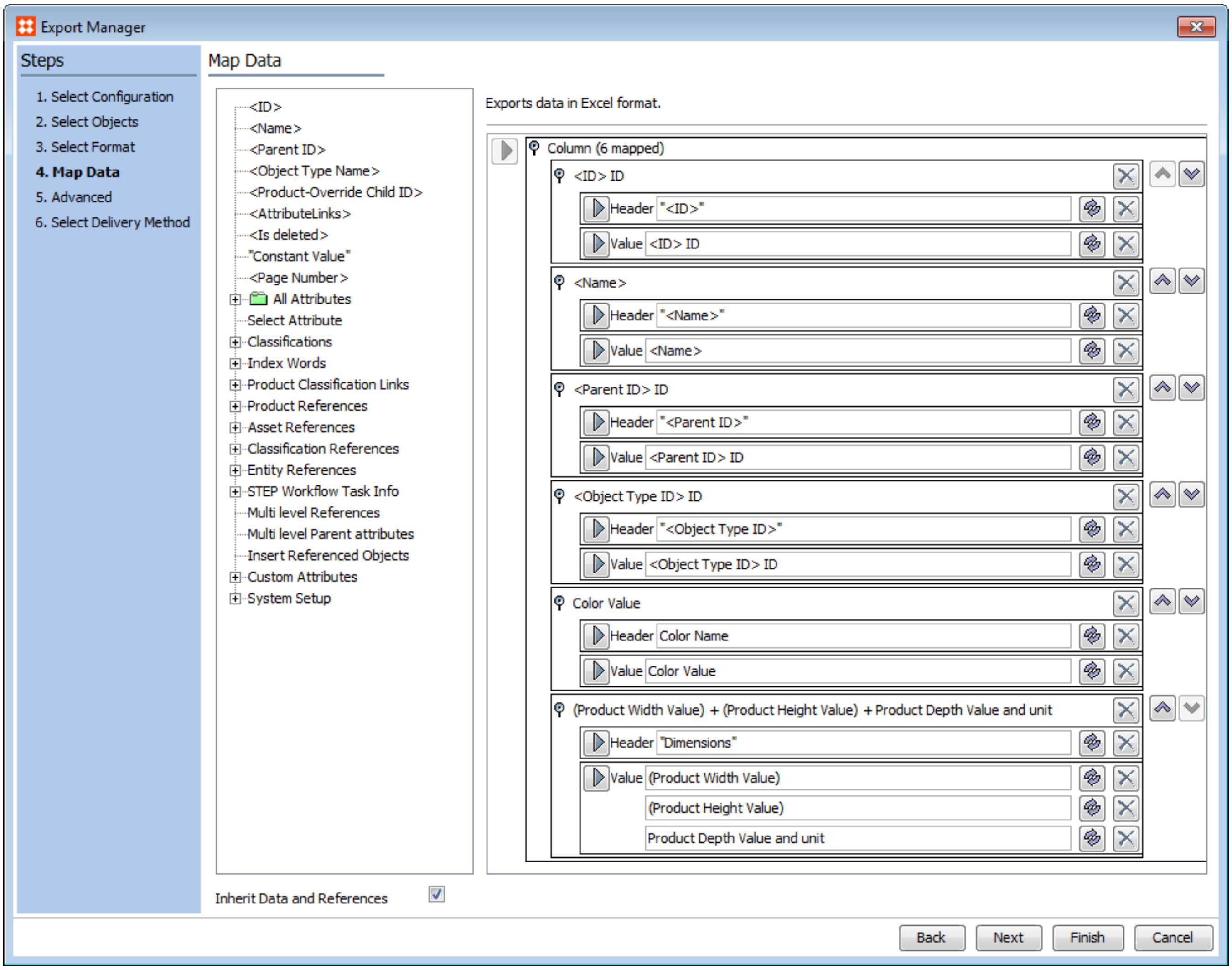
Excel or CSV Overview

The basic process of exporting to Excel / CSV involves identifying data in STEP, creating a mapping layout, adding necessary transformations, and then exporting. For example, the following products will be exported. Below, a view is used to display only the attributes that will be exported.



ID	Name	Object Type	Parent	Product Width	Product Height	Product Depth	Color
179624	AC-P7000-55	Item	P7000	12 in	38 in	8 in	Black
179625	AC-P7000-65	Item	P7000	14 in	40 in	10 in	Blue
179626	AC-P7000-79	Item	P7000	8 in	12 in	4 in	Black
179807	AC-P7000-83	Item	P7000	9 in	15 in	6 in	Blue

Next, a mapping is created in Export Manager for the selected objects.



Export Manager

Steps

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

Map Data

Exports data in Excel format.

Column (6 mapped)

<ID> ID
Header "<ID>"
Value <ID> ID

<Name>
Header "<Name>"
Value <Name>

<Parent ID> ID
Header "<Parent ID>"
Value <Parent ID> ID

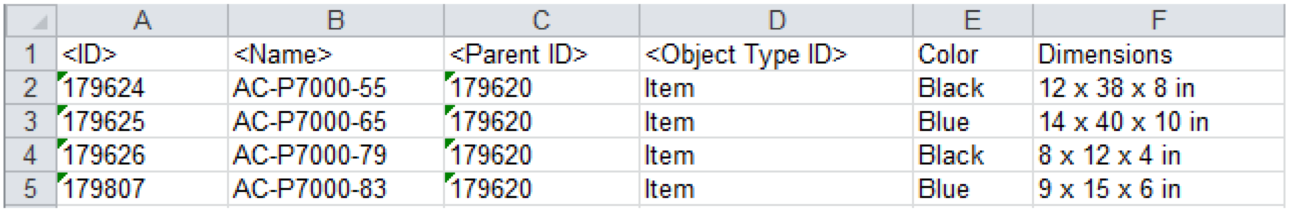
<Object Type ID> ID
Header "<Object Type ID>"
Value <Object Type ID> ID

Color Value
Header Color Name
Value Color Value

(Product Width Value) + (Product Height Value) + Product Depth Value and unit
Header "Dimensions"
Value (Product Width Value)
(Product Height Value)
Product Depth Value and unit

Inherit Data and References ☑

Finally, the Excel file is generated.



	A	B	C	D	E	F
1	<ID>	<Name>	<Parent ID>	<Object Type ID>	Color	Dimensions
2	179624	AC-P7000-55	179620	Item	Black	12 x 38 x 8 in
3	179625	AC-P7000-65	179620	Item	Blue	14 x 40 x 10 in
4	179626	AC-P7000-79	179620	Item	Black	8 x 12 x 4 in
5	179807	AC-P7000-83	179620	Item	Blue	9 x 15 x 6 in

The STEP Export Manager can use both Excel and CSV export formats to export the following node types / super types:

- Products
- Classifications
- Entities

- Assets (objects, not content)
- Attributes (objects and definitions

Export features include options to:

- Select the objects to export
- Select the export format
- Select the metadata to export
- Transform the data for export
- Select how data is delivered from STEP
- Save an Export Configuration to use again

The standard Excel / CSV export options generate an export file with data in the same arrangement as relational database tables. This means that each object is displayed as a single row in the export file and each object property item is displayed as a single column.

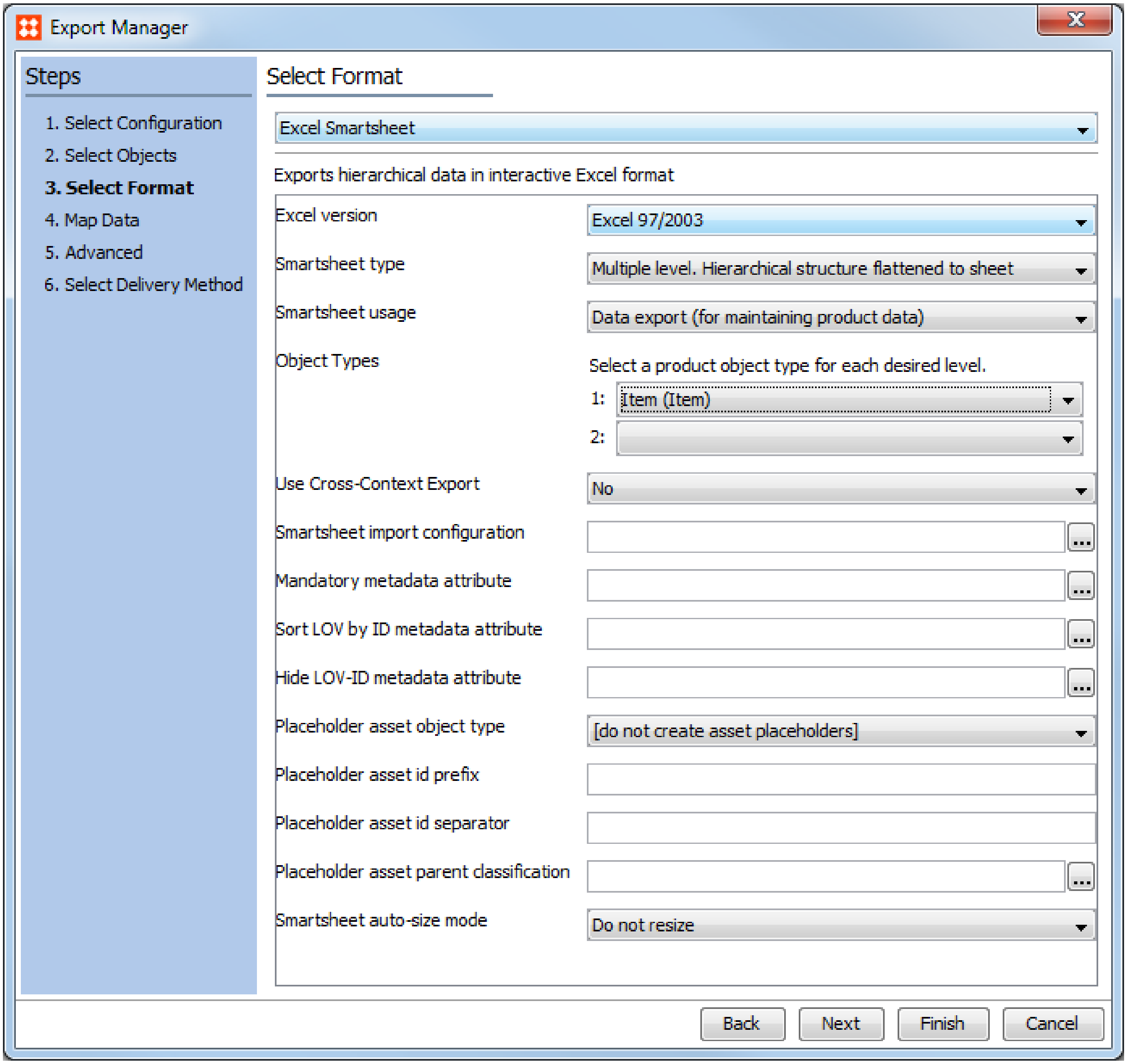
Planning Your Export

Before starting a data export, consider the following points:

- Standard Excel / CSV exports are context and workspace specific. By default, data is extracted from the context and workspace in use when the export is started.
- Rows for different node types / super types cannot be included in the same file. This is limited by the Export dropdown in the Export wizard.
- Exported files will always include a header row.
- For CSV exports, consider and appropriately handle cases where, for example, attribute values can contain the delimiter character selected.
- If you plan to import the data back into STEP, include the STEP IDs in the export.

Smartsheet Format

In this screen, you select Smartsheet as the file export 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	Data export (for maintaining product data)
Object Types	Select a product object type for each desired level. 1: Item (Item) 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

1. In **Excel version**, select the relevant Excel version. Excel 2007 is recommended.
2. In **Smartsheet type**, the following options are available:

- Select **Multiple level. Hierarchical structure flattened to sheet** when wanting to work with Multi-level Smartsheet. 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** when wanting to work with Multi-object Smartsheet. This handles multiple object types simultaneously, without them being hierarchically related; i.e. one object type should not necessarily be a child of the other. This functionality works best when there is a large overlap between the set of attributes that are valid for each of the object types.

3. In **Smartsheet usage**, the following options are available:
   - Select **Data export (for maintaining product data)** to export maintainable 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.
4. In **Object Types**, the following choices are available:
   - 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. Only 5 entries are allowed.
   - For Multi-object Smartsheets, specify the object type(s) in question. There is no limit on how many entries are allowed.
5. In **Use Cross-Context Export**, choose **Yes** or **No**.

   When set to **Yes**, the contexts that should be included in the export are specified. 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.
6. In **Smartsheet import configuration**, select an existing import configuration to apply to the Smartsheet. The import configuration contains predefined business rules and actions. This field is optional.
7. In **Mandatory metadata attribute**, select an attribute to mark columns as mandatory in the Smartsheet. The attribute must be of the attribute type Description and of the validation base types Integer or Numeric. The attribute must also be added as metadata to attributes or attribute links. For attributes that should appear as a mandatory column in the Smartsheet, the metadata must have a non-zero value.
8. **Sort LOV by ID metadata attribute** is used for individual attribute columns that have a LOV as validation base type in order to sort the (optionally filtered) list of values by value ID. This changes the order in drop-down lists and pop-up LOV selector in the Smartsheet column(s).

   If a Description attribute made valid for attributes or attribute links contains non-zero values for the particular column, the list of values will be sorted by value ID.

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

9. **Hide LOV-ID metadata attribute** is used to exclude external IDs from individual LOV-based Smartsheet columns.
10. **Placeholder asset object type** defines 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.

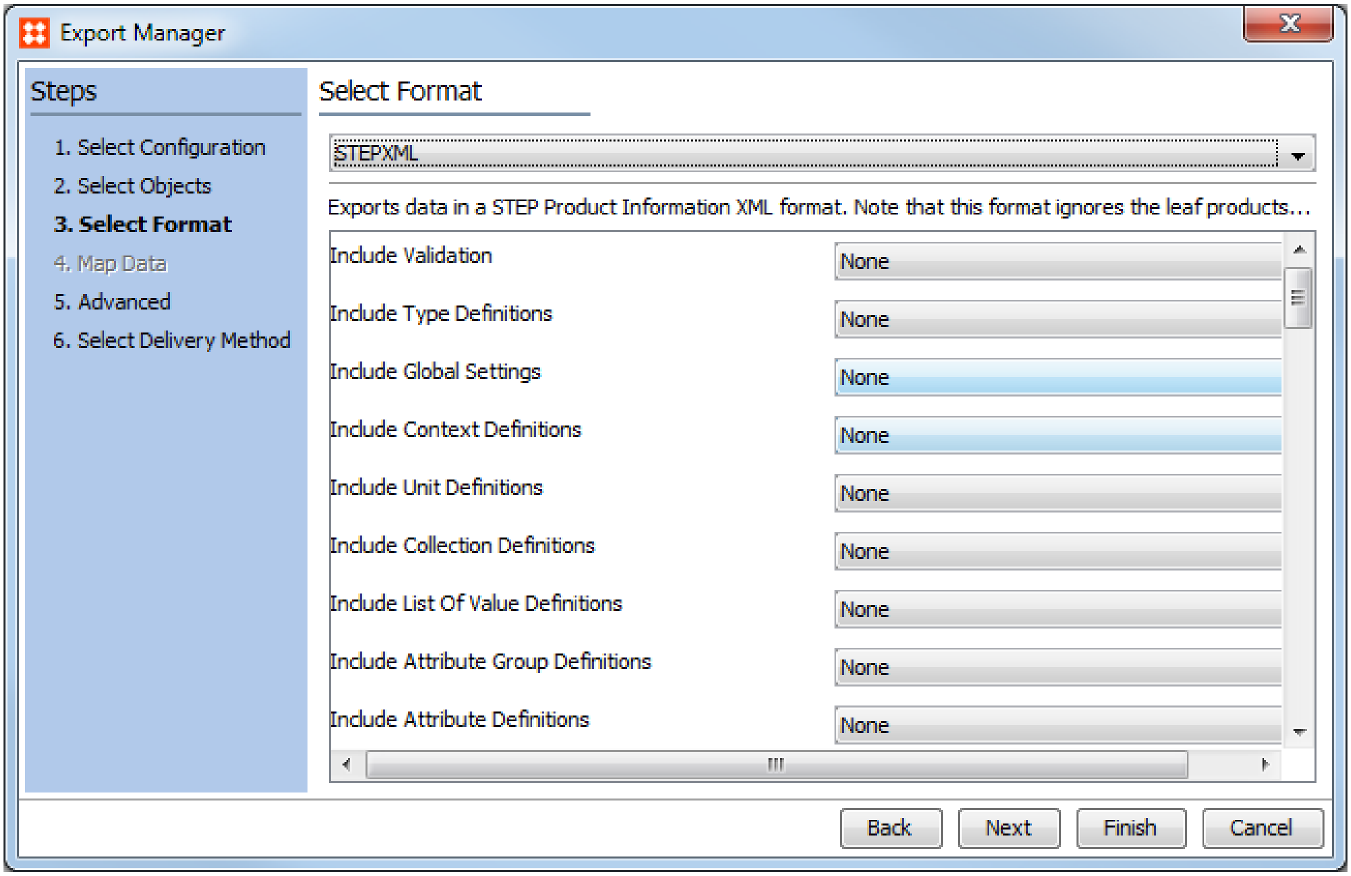
    Placeholder assets make it possible for the created assets to have a determinable ID so other processes are able to import and maintain actual asset content.

    If **[do not create asset placeholders]** is selected, the asset placeholder system is disabled and normal asset references are maintained in the Smartsheet.
11. **Placeholder asset ID prefix** is used during export from STEP Workbench to ensure that asset IDs are unique across all suppliers. The prefix will be changed during portal export. This field is optional.
12. **Placeholder asset ID separator** is inserted in the final ID between the prefix and the file name. This field is optional.

    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". 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.
13. **Placeholder asset parent classification** defines where placeholder assets are created when the Smartsheet is re-imported. It is configurable in the Export manager to allow the export to be run from STEP Workbench.
14. **Smartsheet auto-size mode** specifies whether to resize the exported Smartsheets either by column-header width or by cell-content width using word wrap, or not to resize at all.
15. Click **Next** to move on to the next screen.

STEPXML Format

If you select the STEPXML format, a number of parameters determine the data included in the export file. These parameters eliminate the need for the Map Data step.



STEPXML

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

Include Validation: None

Include Type Definitions: None

Include Global Settings: None

Include Context Definitions: None

Include Unit Definitions: None

Include Collection Definitions: None

Include List Of Value Definitions: None

Include Attribute Group Definitions: None

Include Attribute Definitions: None

For details about the available options, see **STEPXML Minimum, Referenced, and Selected** section of the **Exporting Data** documentation.

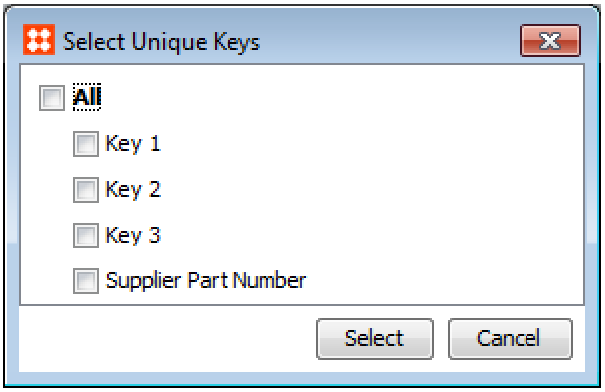
Parameter	Descriptions
Include Validation	Controls the type of validation applied, if any. The following options are available: None, Schema, and DTD. For more information about schema and validation, see the API documentation.
Include Type	Controls output of system specific object types, edge types, and system specific

Parameter	Descriptions
Definitions	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 Definitions	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 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.

Parameter	Descriptions
	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 outputs only the attributes used in the selected product hierarchy. **Important:** System specific attributes (having an ID that starts with **stibo.** or **asset.**) are included.
Include Assets	Controls output of description attributes and meta attributes on assets. If the asset contains an image or document, the exported file will not include the actual image or document. The following options are available: None, Minimum, and All. When exporting a product hierarchy or classification hierarchy, selecting Minimum outputs only the assets used in the selected product hierarchy.
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.

Parameter	Descriptions
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. 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 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

Parameter	Descriptions
	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: • 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.
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

Parameter	Descriptions
	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. **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.
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.  Select Unique Keys □ All □ Key 1 □ Key 2 □ Key 3 □ Supplier Part Number Select Cancel
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. 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 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. The following options are available: None and All.
Include Baskets	Controls output of details of baskets. The following options are available: None and All.
Include Pagination Rules	Controls output of details of pagination rules. The following options are available: None and All.

Parameter	Descriptions
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. The following options are available: None and All.
Include Matching Algorithms	Controls output of matching algorithms, 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 Web UI Configurations	Controls output of Web UIconfigurations in the exported file. The following options are available: None and All.
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	Controls output of event processor information, including ID, name, links, and

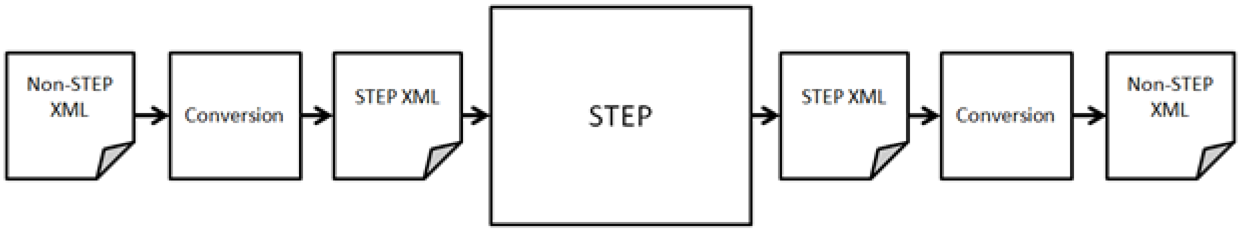
Parameter	Descriptions
Processors	configuration. The configurations are exported in a 64 bit encoding and can only be imported without modifications into another STEP system. The following options are available: None and All.
Include Setup Groups	Controls output of setup groups that are defined in system setup to hold integration endpoints, Web UI configurations, STEP workflows, and business rules. 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 Overview

STEPXML is the native XML format for STEP. Where other supported import / export formats are limited to specific super types and node types, with STEP XML it is 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 environment and for inbound integrations with systems / middleware capable of producing STEPXML.

As illustrated below, all data that enters and exits STEP via the import / export functionality (including Integration Endpoints) uses the STEPXML format. For example, when a CSV file is imported, the Import Manager converts the data to STEPXML, and then imports the STEPXML data.



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) cannot be represented in STEP XML.
- 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 / DTD files.

XSD and DTD files for STEP XML are available at:

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

A web view of the schema is available at:

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

STEPXML Examples

The following STEPXML file 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>
```

In the following STEPXML file, product, classification, and entity data are nested which reflects the Tree hierarchy structure. SalesItem-13112 and SalesItem-13113 are children of SalesItemFamily-1311.

```
<?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>
```

Same information in a 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>
```

STEPXML Multi-Context Data

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 Größe
n 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 A
cme 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.

STEPXML Backward Compatibility

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.

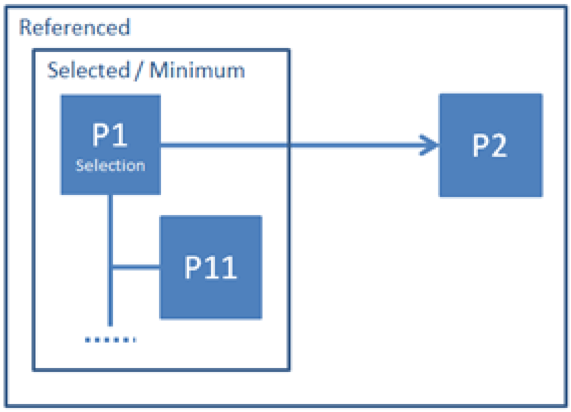
The legacy names that most frequently cause problems are:

- **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.
- **ProductMode** – attribute elements have a ProductMode attribute. This attribute specifies whether the attribute is a description attribute (value: Property) or a specification attribute (value: Normal).
- **Derived** – attribute element that specifies if the attribute is calculated.
- **ClassificationReference** – used inside both the Product and Asset elements. For assets, this is the parent link (assets can have multiple parents). For products, the element represents product-to-classification links.

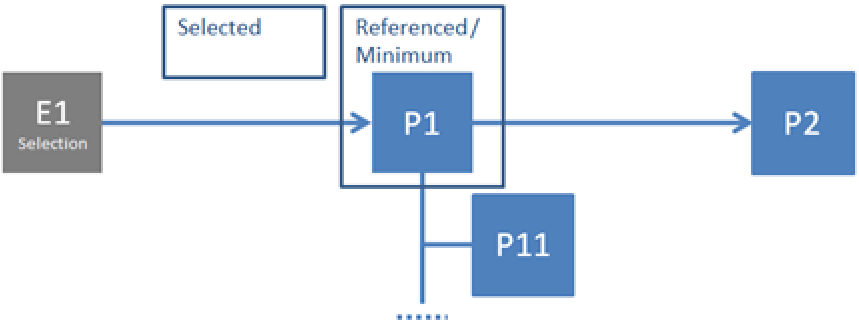
STEPXML Minimum, Referenced, and Selected

Many of the parameters for STEPXML format on the Select Format step of the wizard include the options Minimum, Referenced, and/or Selected.

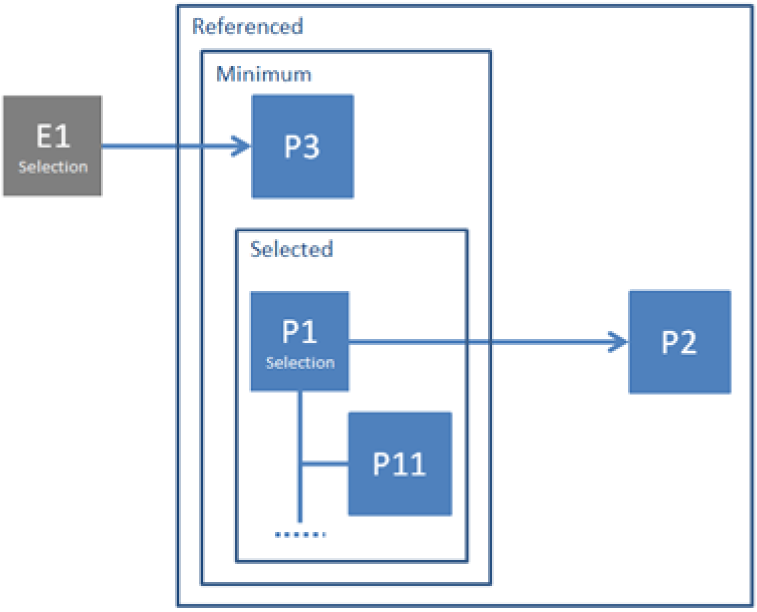
In the example below, product P1 has been selected as root node for the export. The Selected and Minimum options cause P1 and its descendants to be exported. The Referenced setting will also include P2 since it is referenced from P1.



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.

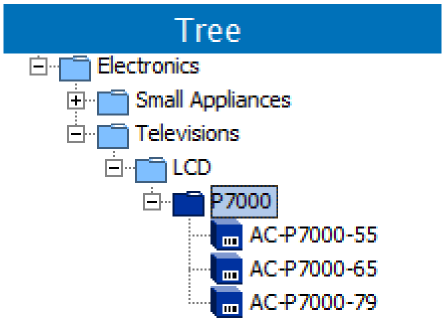


In the following example, the reason for having three options is demonstrated. Here, both a non-product (E1) and a product (P1) are selected as root nodes for the export. With this selection, Referenced, Minimum, and Selected will have three different meanings.



Hierarchy Export Including Objects Referenced

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.



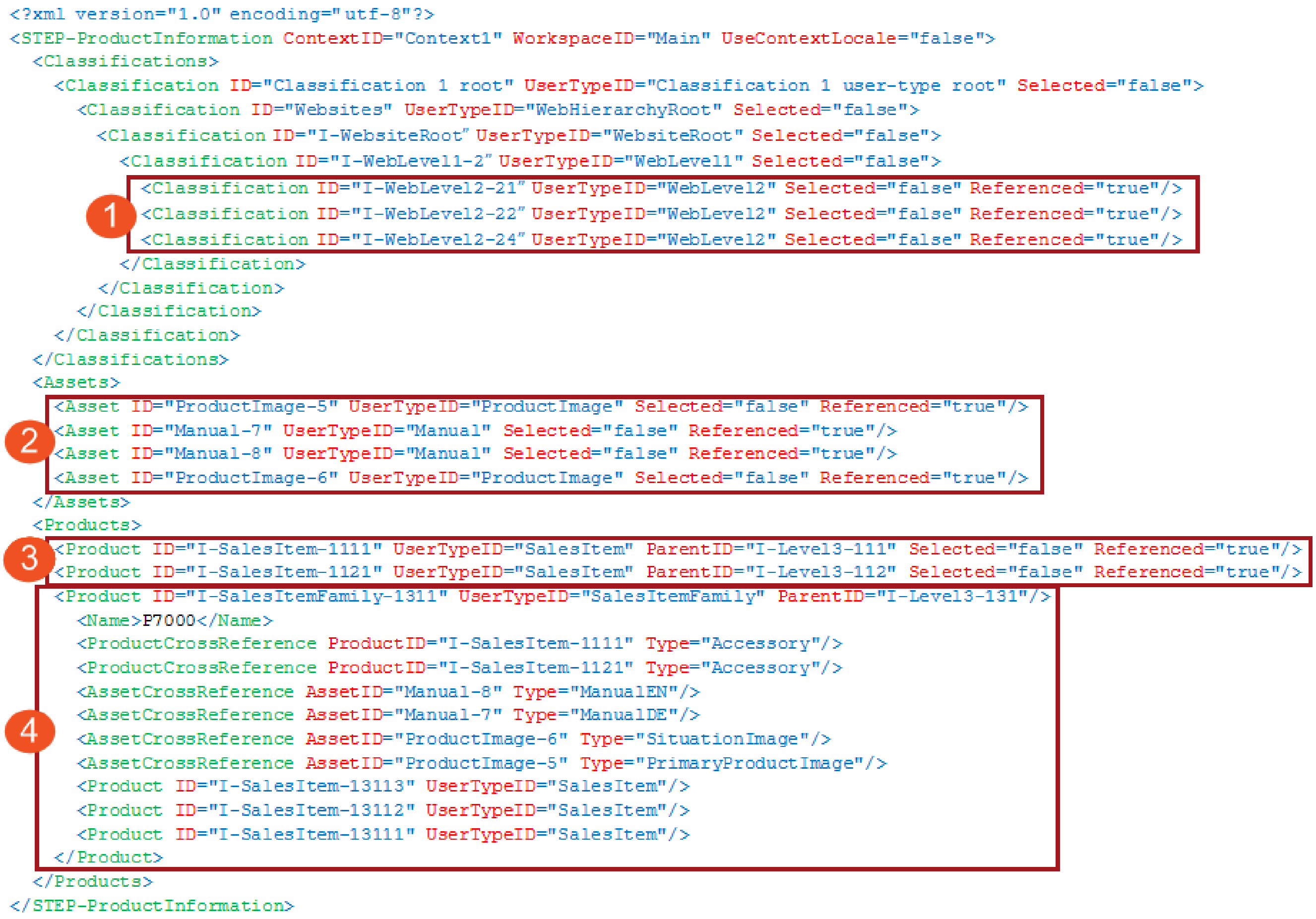
Product P7000 is selected as the root and the export uses the following settings:

- **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>
    </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

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**

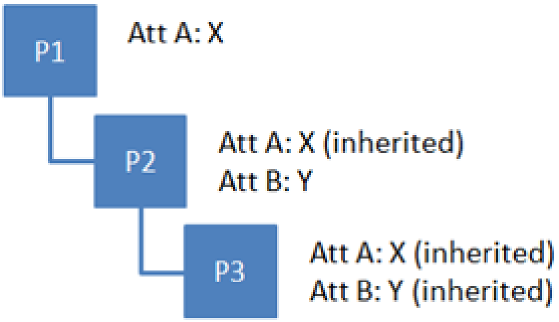
Below is a modified version of the results. All but one unit, one list of values group, one list of values and one attribute has been removed. Also, the exported product has again been stripped of name, references / links, and values.

```
<?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>
```

STEPXML Inheritance

By default, inherited values and references / links are not exported in STEPXML. This default behavior can be overruled by setting the **Export inherited values and references** parameter to yes.

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.

Generic XML Format

The Generic XML format in STEP allows you to import data from and export data to a variety of different XML formats without the need for extensions or customizations.

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 in the same way as exporting data in CSV or Excel.

Creating a Generic XML export file requires:

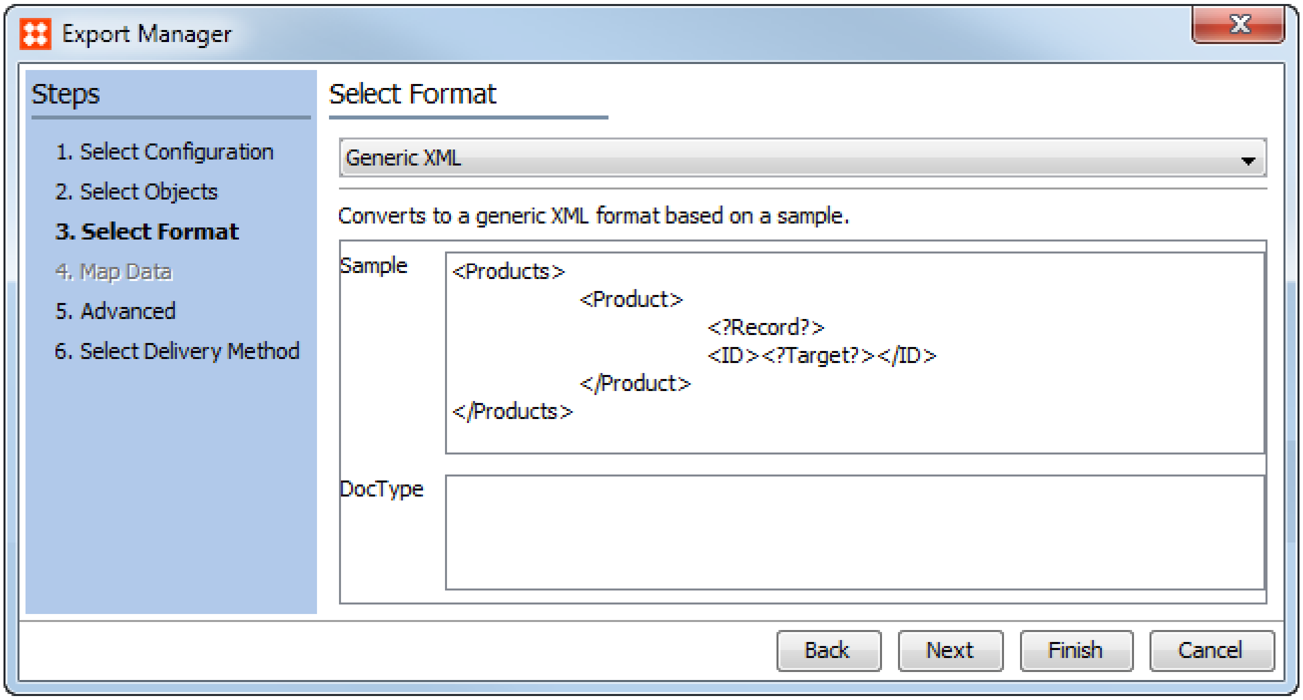
- Selecting objects to export - Information selected for export is then displayed as a list of **data sources**.
- Providing a template - A sample of the data structure for the output file determines the **mapping targets**.
- Associating data sources with mapping targets - **Mapping rules** determine where STEP information appears in the output file.

Data sources are evaluated against the object being exported to determine which data will be inserted into the output XML at positions specified by mapping targets.

Mapping Targets

The Sample field is used to provide a template for the output file. A basic Generic XML template is an XML document that represents the set of nested rules for the output.

Mapping targets, like <?Target?> in the screenshot below, are placeholders within the template. Mapping targets tell the export where to insert actual data.



Generic XML

Converts to a generic XML format based on a sample.

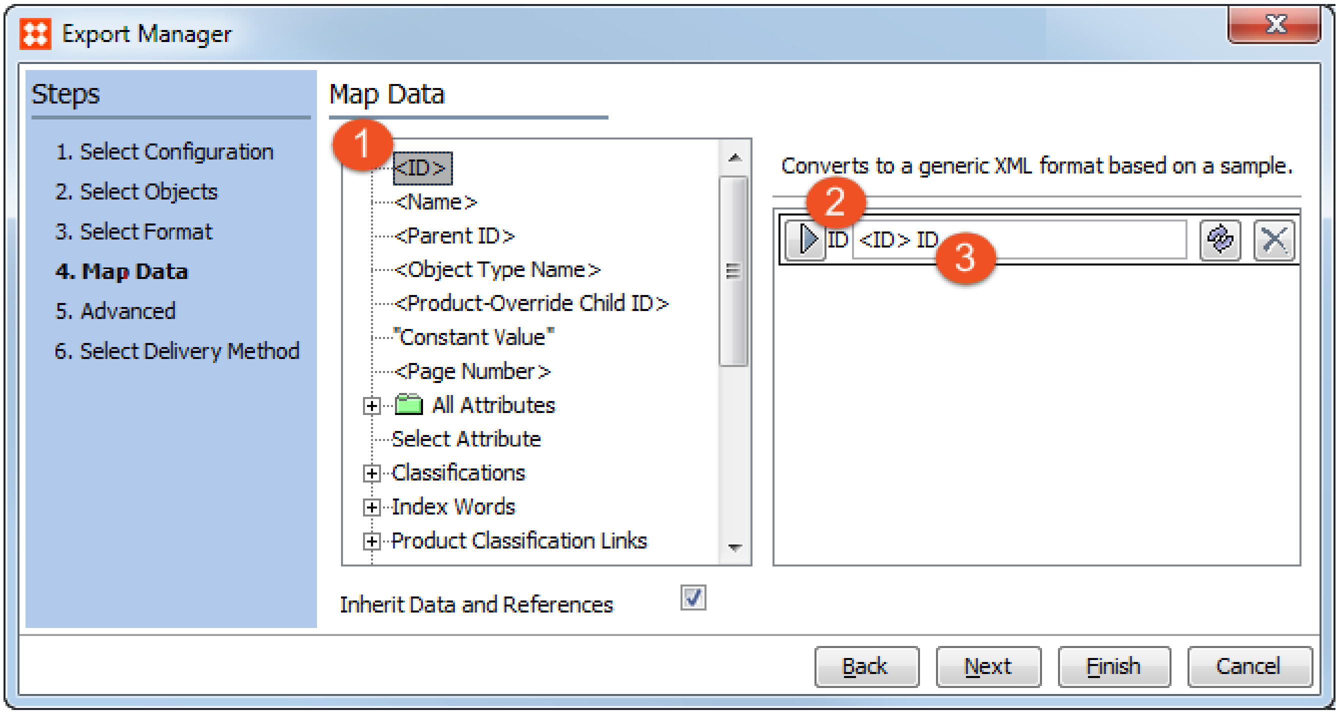
```
<Products>
            <Product>
                        <?Record?>
                        <ID><?Target?></ID>
            </Product>
</Products>
```

Data Sources

Data sources indicate the original location of the exported information. The Map Data page of the wizard provides a way to associate the mapping targets defined in the template with the STEP information available for the selected object types being exported.

Mapping Rules

Mapping rules are created by setting a relationship, also called a binding, between a data source and a mapping target. Bindings can be configured with transformations that will be applied to the output of a Data Source.



Converts to a generic XML format based on a sample.

ID <ID> ID

Inherit Data and References ☑

1. The objects selected in Step 2. Select Objects determine the **data sources** displayed in Step 4. Map Data.
2. The **mapping targets** in the template from Step 3. Select Format are displayed as mapping targets in Step 4. Map Data.
3. Selecting a data source and clicking a mapping target arrow button creates a **mapping rule**. This binds a data source to a mapping target.

See section Step 4 - Map Data of The Data Export Wizard documentation for more information.

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.

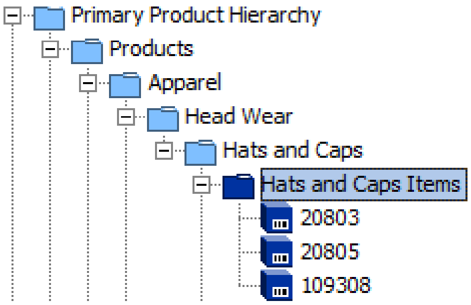
- The Generic XML import functionality cannot handle nested records.
- Neither data nor objects can be deleted when importing with Generic XML.

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.



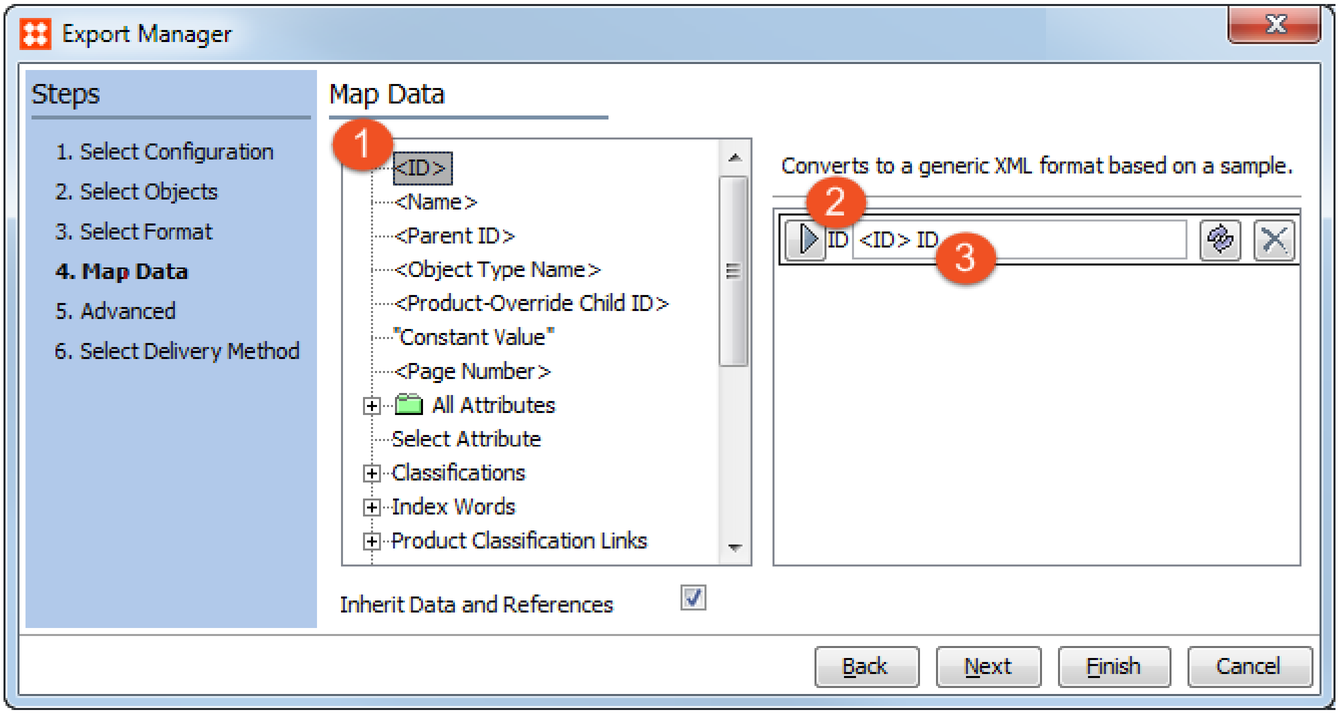
Primary Product Hierarchy
Products
Apparel
Head Wear
Hats and Caps
Hats and Caps Items
20803
20805
109308

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

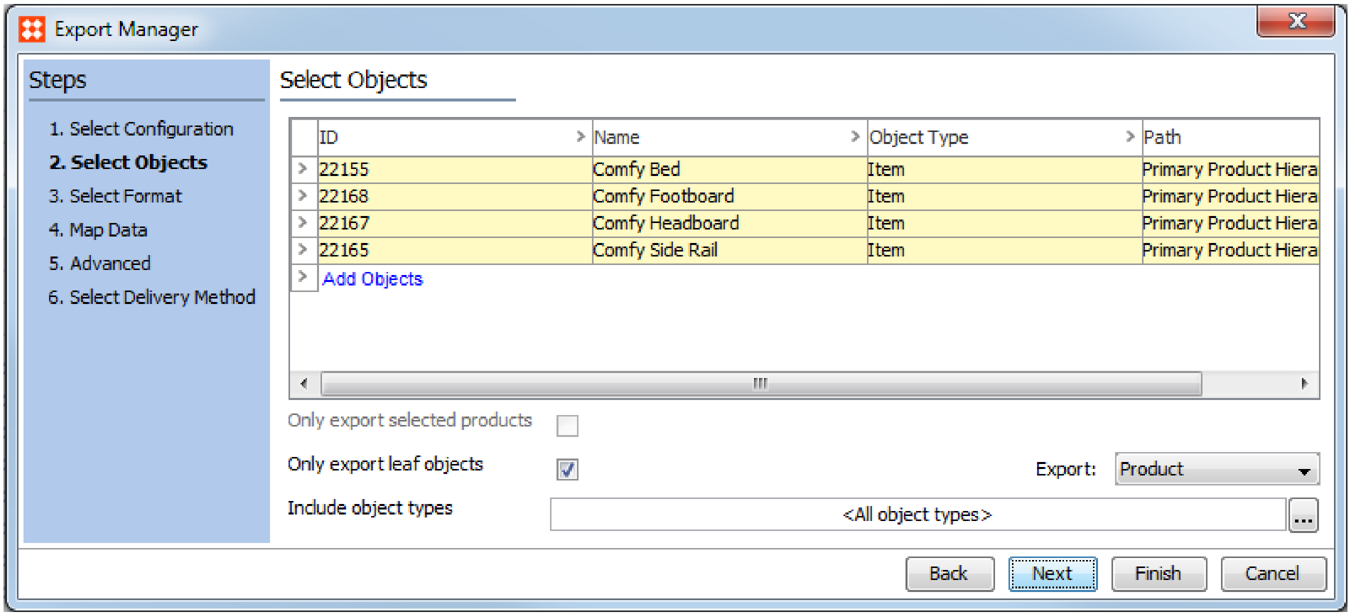
Generic XML Single Cardinality Tags

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
- Target Override Default

Objects



ID	Name	Object Type	Path
22155	Comfy Bed	Item	Primary Product Hiera
22168	Comfy Footboard	Item	Primary Product Hiera
22167	Comfy Headboard	Item	Primary Product Hiera
22165	Comfy Side Rail	Item	Primary Product Hiera
Add Objects			

Only export selected products □

Only export leaf objects ☑

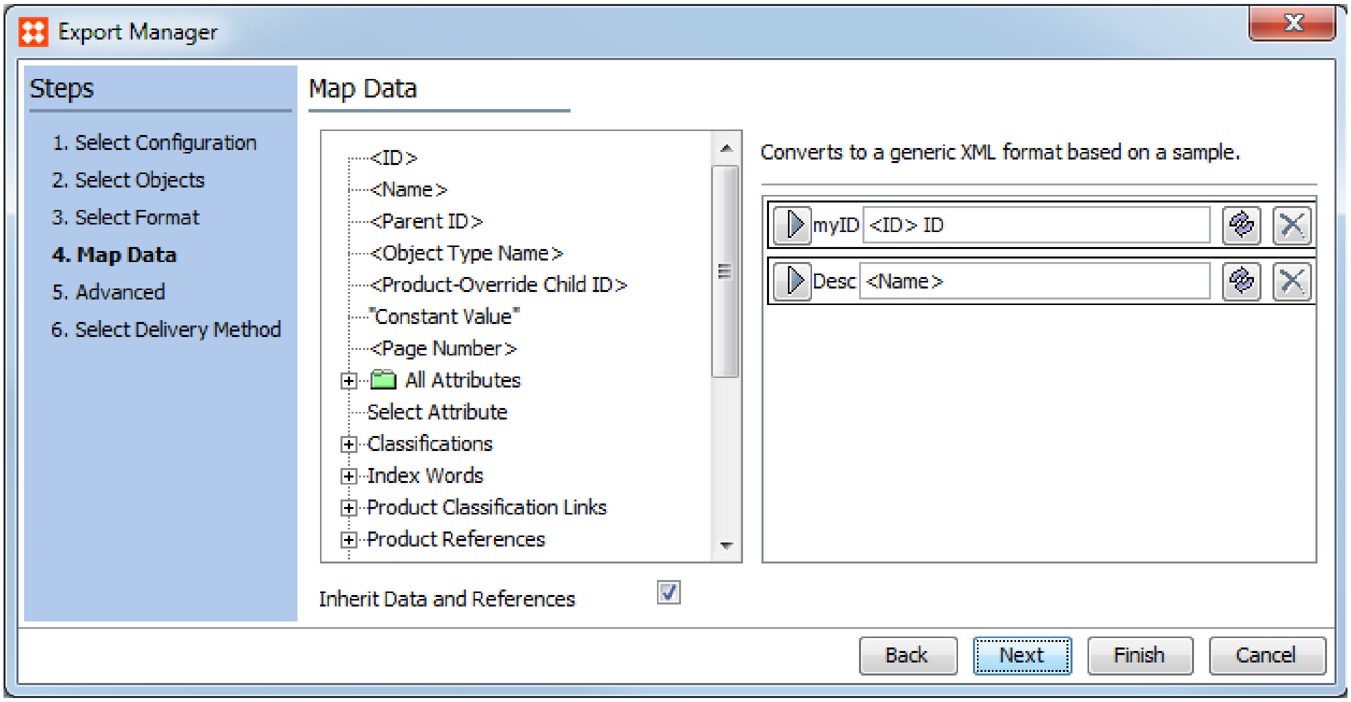
Export: Product

Include object types <All object types>

Template

```
<Products>
   <Product>
      <?Record?>
      <ID><?Target myID?></ID>
      <Desc><?Target?></Desc>
   </Product>
</Products>
```

Mapping



Converts to a generic XML format based on a sample.

myID <ID> ID

Desc <Name>

Inherit Data and References ☑

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.

Generic XML Multi-Cardinality Tags

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:

- Repeated Target
- MultiTarget
- Composite Mapping Targets

Objects

The following objects are used for each of the examples in this section.

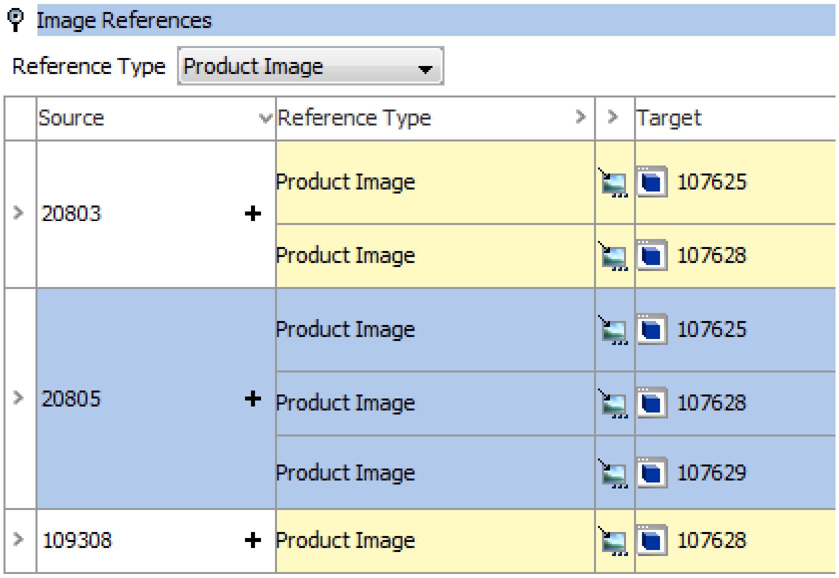


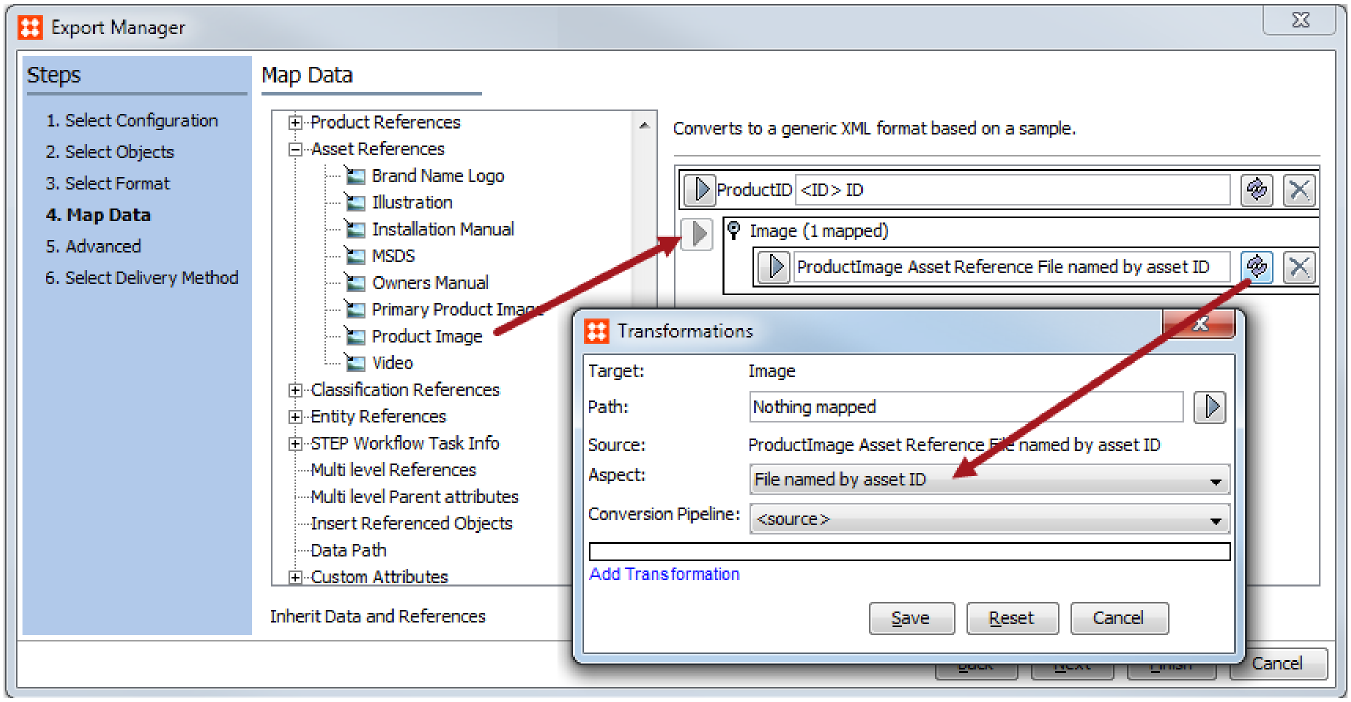
Image References

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.



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

Converts to a generic XML format based on a sample.

ProductID <ID> ID

Image (1 mapped)

ProductImage Asset Reference File named by asset ID

Target: Image

Path: Nothing mapped

Source: ProductImage Asset Reference File named by asset ID

Aspect: File named by asset ID

Conversion Pipeline: <source>

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>.

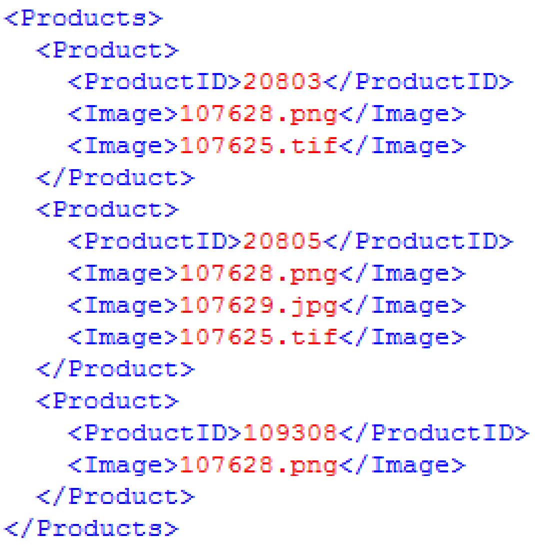
When using Repeated Targets, mapping target declarations must be repeated immediately and the tag name and mapping target ID must be identical. This technique **cannot** be used for mapping targets declared below composite mapping targets.

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.

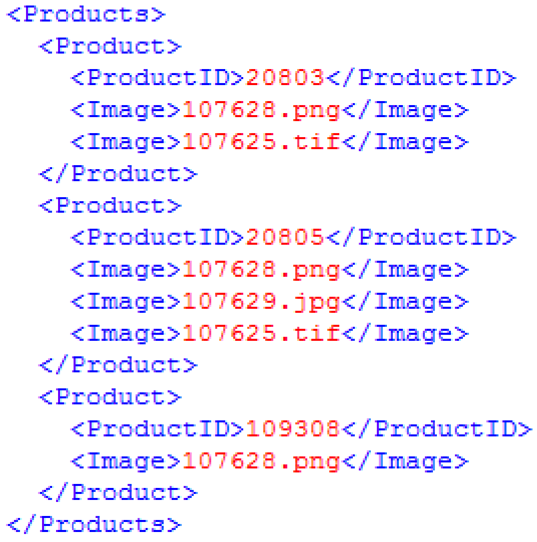
This technique **can** be used for mapping targets declared below composite mapping targets.

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.

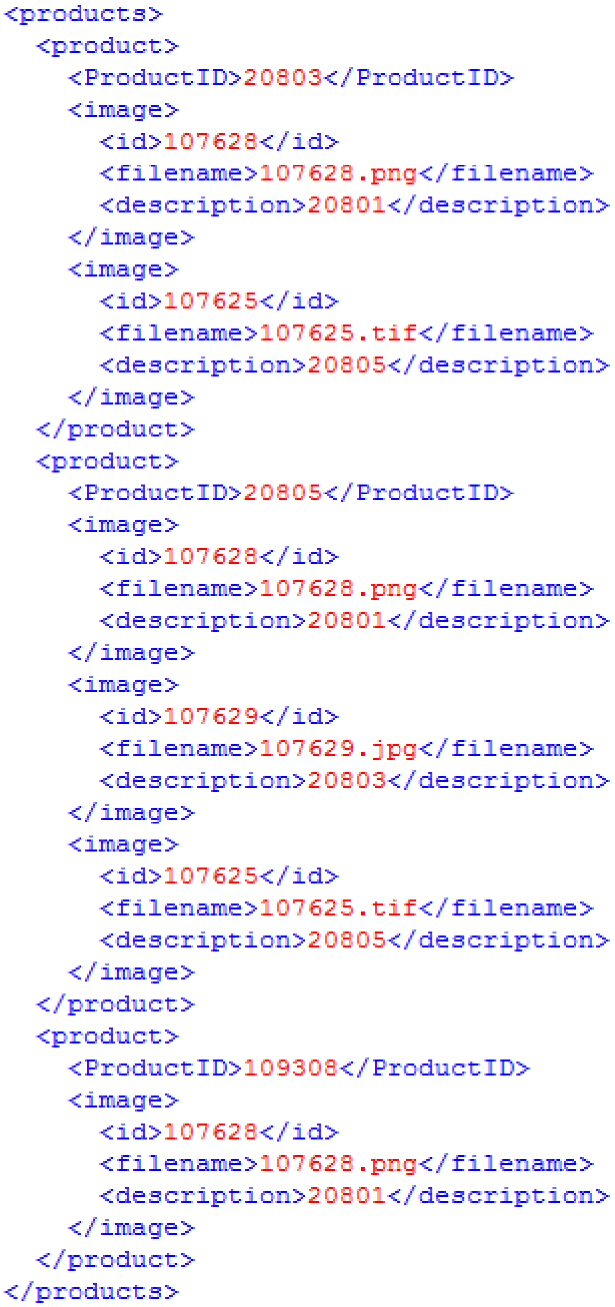
Mapping a data source to a composite mapping target will cause the composite substructure to be evaluated and written to the output once for each value produced by the data source.

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>
```

RemoveIfEmpty Tag

In an outbound Generic XML template, the Generic XML instruction **RemoveIfEmpty** makes it possible to define that a tag should be removed if the tag does not contain any children. For example, consider the template:

```
<Product>
   <id><?Target?></id>
   <ValueGroup id="group1">
      <Value1>
      <?Target?>
      </Value1>
      <Value2>
      <?Target?>
      </Value2>
   </ValueGroup>
</Product>
```

If the mapping for the id tag evaluates to 1234, but the mappings for Value1 and Value2 evaluate to nothing, this would generate the XML:

```
<Product>
   <id>1234</id>
   <ValueGroup id="group1"/>
</Product>
```

This is valid XML, but might not be right in the given case, since the ValueGroup should not have been output. To achieve this, use the **RemoveIfEmpty** instruction in the template:

```
<Product>
   <ValueGroup id="group1">
   <?RemoveIfEmpty?>
      <Value1>
      <?Target?>
      </Value1>
      <Value2>
      <?Target?>
      </Value2>
   </ValueGroup>
</Product>
```

Used with the same data as above, this generates XML where the ValueGroup tag has been removed since all the child XML elements of the tag are empty:

```
<Product>
   <id>1234</id>
</Product>
```

Recursive option

To remove empty XML elements in an XML hierarchy, add the **RemoveIfEmpty** tag to each level in the hierarchy that should be removed:

```
<Product>
   <id><?Target?></id>
   <ValueGroup id="group1">
   <?RemoveIfEmpty?>
      <ChildGroup id="childGroup">
      <?RemoveIfEmpty?>
         <Value1>
         <?Target?>
         </Value1>
         <Value2>
         <?Target?>
         </Value2>
      </ChildGroup>
   </ValueGroup>
</Product>
```

For deep-nested XML structures, adding tags to each level can be rather cumbersome. Instead, use the **recursive** option for the instruction. This removes the tag, and all child tags, if one of them does not contain child elements, as shown here:

```
<Product>
   <id><?Target?></id>
   <ValueGroup id="group1">
   <?RemoveIfEmpty recursive?>
      <ChildGroup id="childGroup">
         <Value1>
         <?Target?>
         </Value1>
         <Value2>
         <?Target?>
         </Value2>
      </ChildGroup>
   </ValueGroup>
</Product>
```

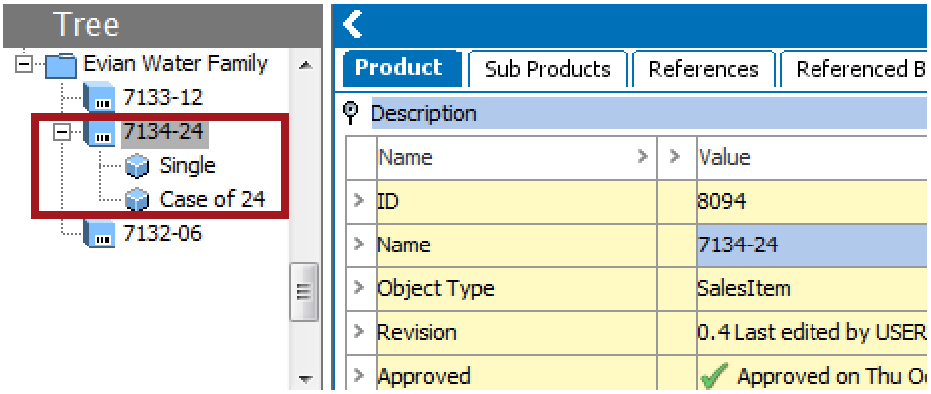
For information about the 'Allow empty tags' option, see the **Empty Values** section.

DoNotEmit Tag

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.



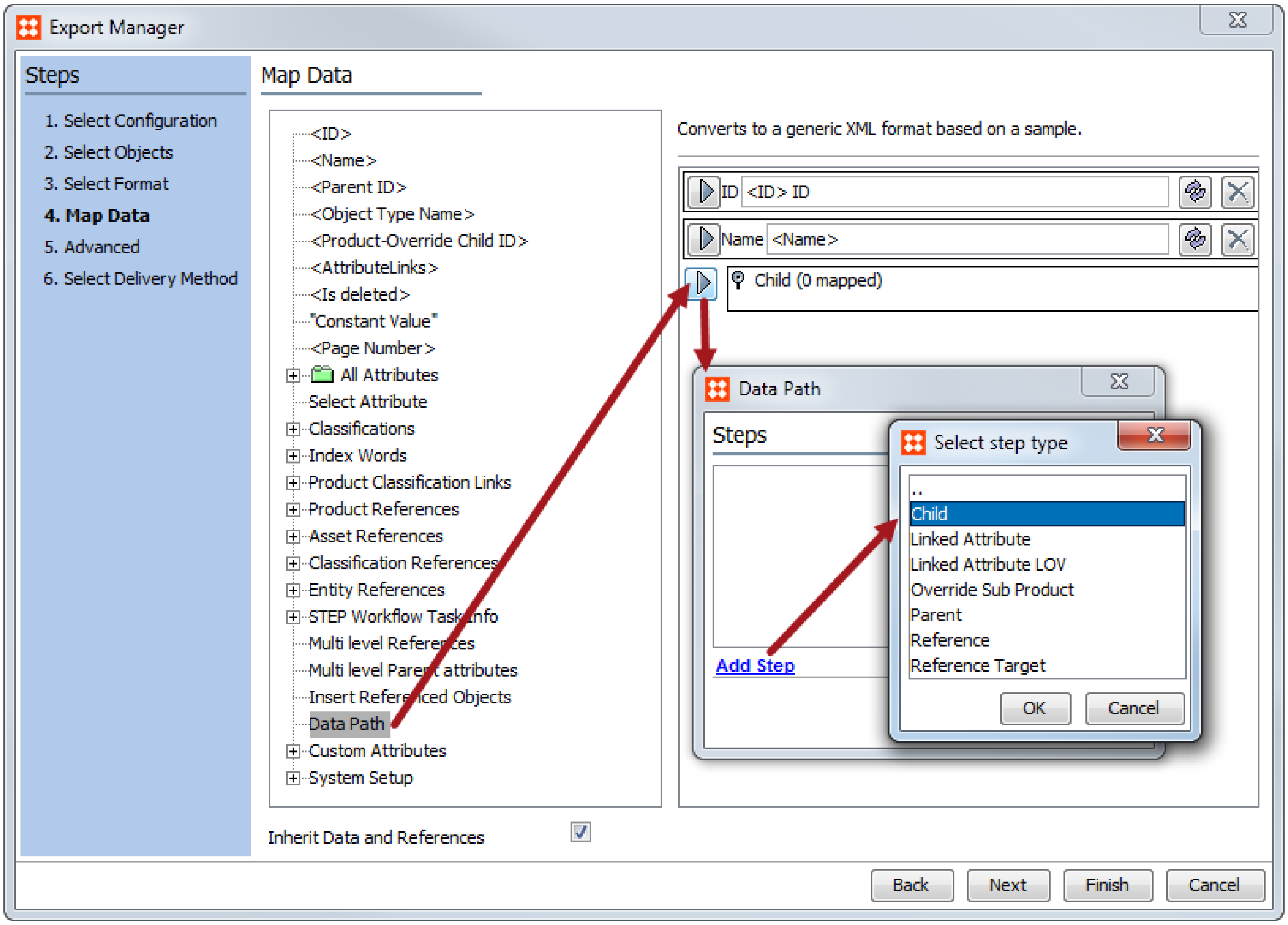
Name	Value
ID	8094
Name	7134-24
Object Type	SalesItem
Revision	0.4 Last edited by USER
Approved	✓ Approved on Thu O

Template

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

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.



Converts to a generic XML format based on a sample.

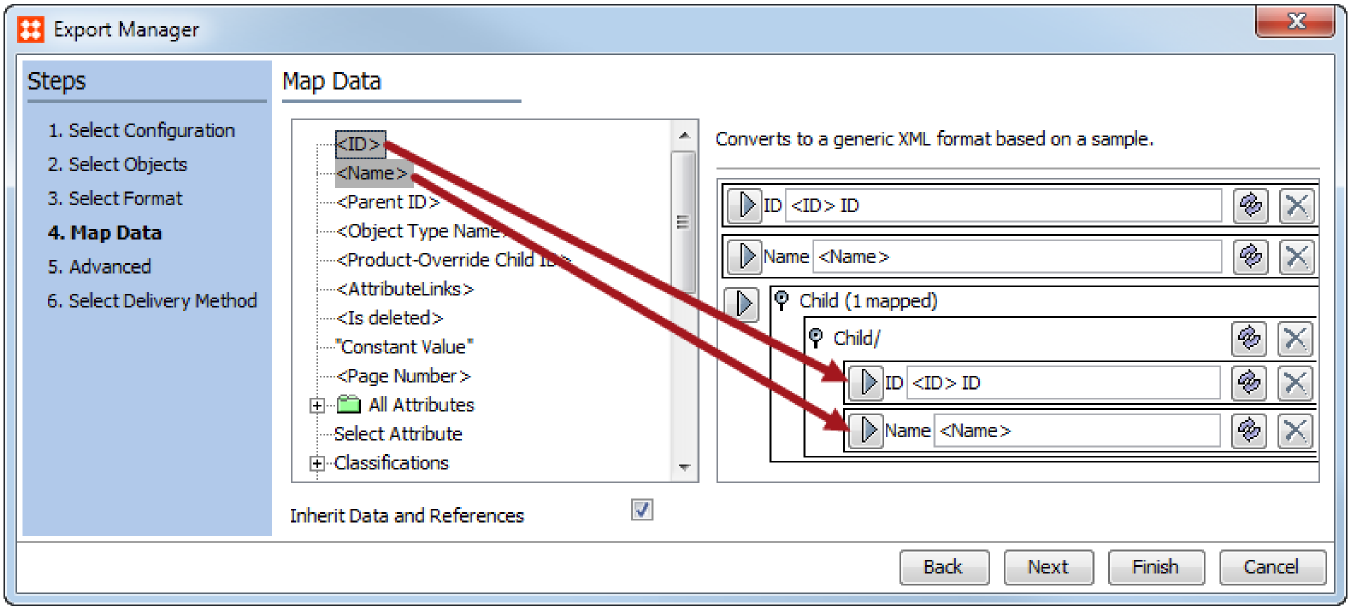
ID <ID> ID

Name <Name>

Child (0 mapped)

Inherit Data and References ☑

Finally, map the Child's ID and Name.



Export Manager

Steps

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

Map Data

Converts to a generic XML format based on a sample.

ID <ID> ID

Name <Name>

Child (1 mapped)

Child/

ID <ID> ID

Name <Name>

Inherit Data and References ☑

Results

Without DoNotEmit	With DoNotEmit

`<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>`
	`<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>`

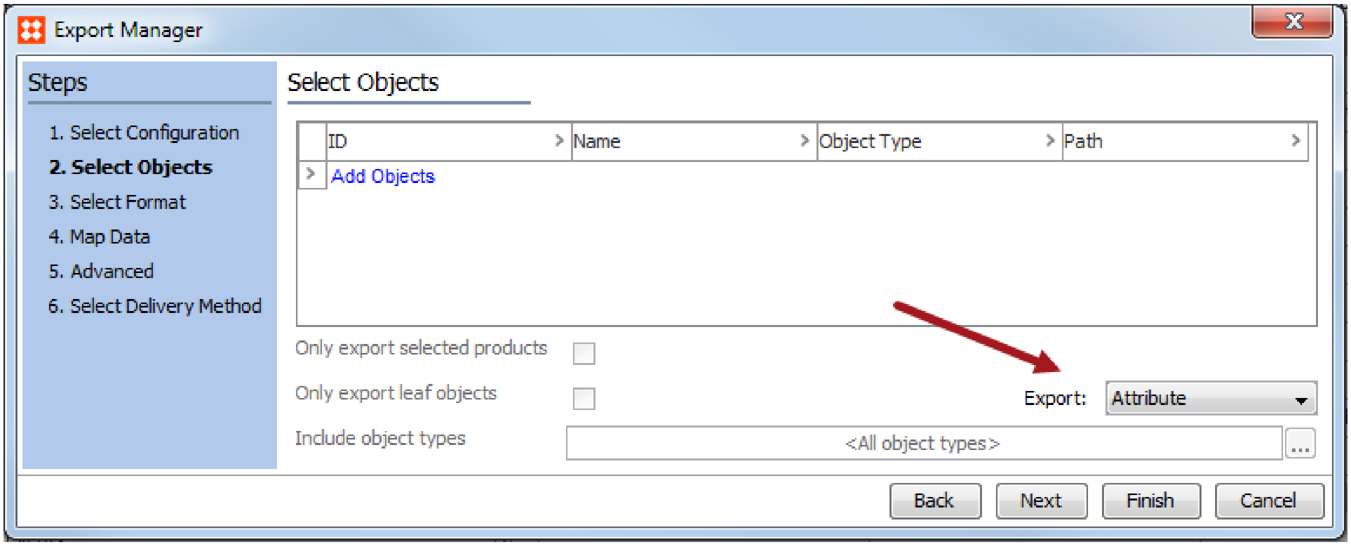
Attributes

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.



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			

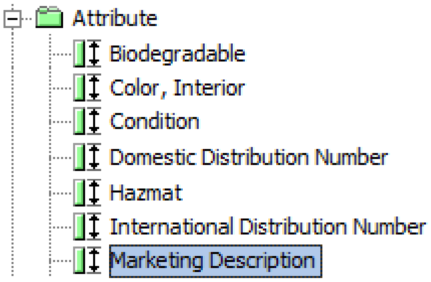
Only export selected products □

Only export leaf objects □

Export: Attribute

Include object types <All object types>

Use the **Add Objects** link to select a single attribute, multiple attributes, or an attribute group to export.



Attribute
- Biodegradable
- Color, Interior
- Condition
- Domestic Distribution Number
- Hazmat
- International Distribution Number
- Marketing Description

Note: If exporting a group, data for each field in the template is evaluated for each child in the group.

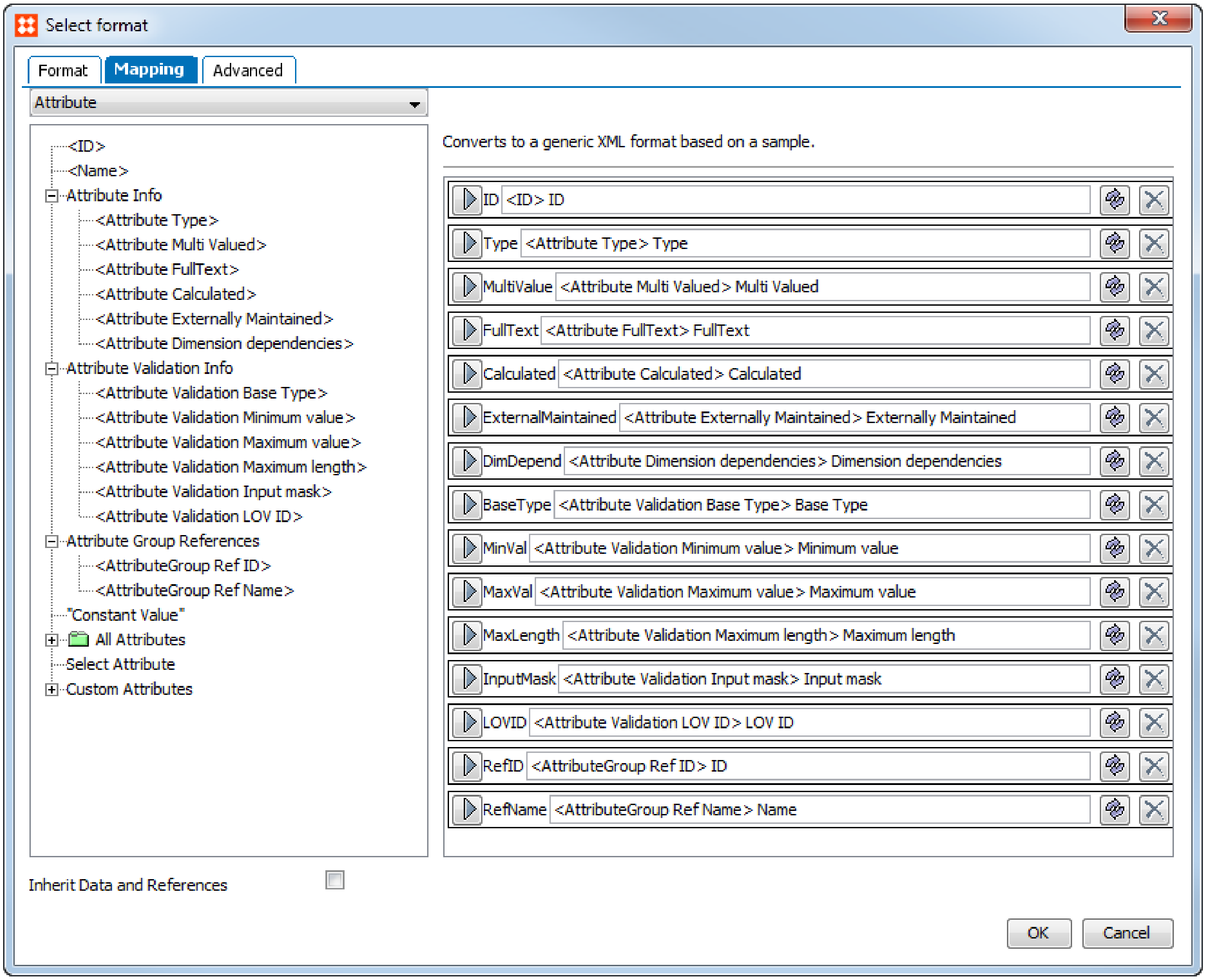
Template

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:



Attribute

<ID>
<Name>
Attribute Info
<Attribute Type>
<Attribute Multi Valued>
<Attribute FullText>
<Attribute Calculated>
<Attribute Externally Maintained>
<Attribute Dimension dependencies>
Attribute Validation Info
<Attribute Validation Base Type>
<Attribute Validation Minimum value>
<Attribute Validation Maximum value>
<Attribute Validation Maximum length>
<Attribute Validation Input mask>
<Attribute Validation LOV ID>
Attribute Group References
<AttributeGroup Ref ID>
<AttributeGroup Ref Name>
"Constant Value"
All Attributes
Select Attribute
Custom Attributes

Converts to a generic XML format based on a sample.

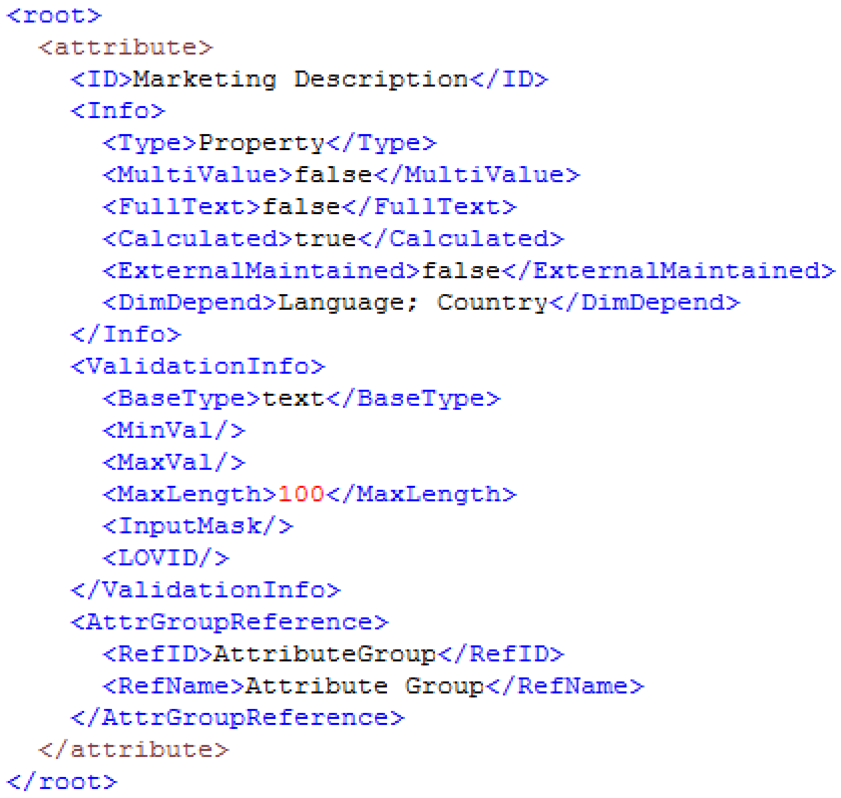
ID <ID> ID
Type <Attribute Type> Type
MultiValue <Attribute Multi Valued> Multi Valued
FullText <Attribute FullText> FullText
Calculated <Attribute Calculated> Calculated
ExternalMaintained <Attribute Externally Maintained> Externally Maintained
DimDepend <Attribute Dimension dependencies> Dimension dependencies
BaseType <Attribute Validation Base Type> Base Type
MinVal <Attribute Validation Minimum value> Minimum value
MaxVal <Attribute Validation Maximum value> Maximum value
MaxLength <Attribute Validation Maximum length> Maximum length
InputMask <Attribute Validation Input mask> Input mask
LOVID <Attribute Validation LOV ID> LOV ID
RefID <AttributeGroup Ref ID> ID
RefName <AttributeGroup Ref Name> Name

Inherit Data and References ☐

See the Attribute Aspects section of the Generic XML documentation for a description of each available data point.

Results

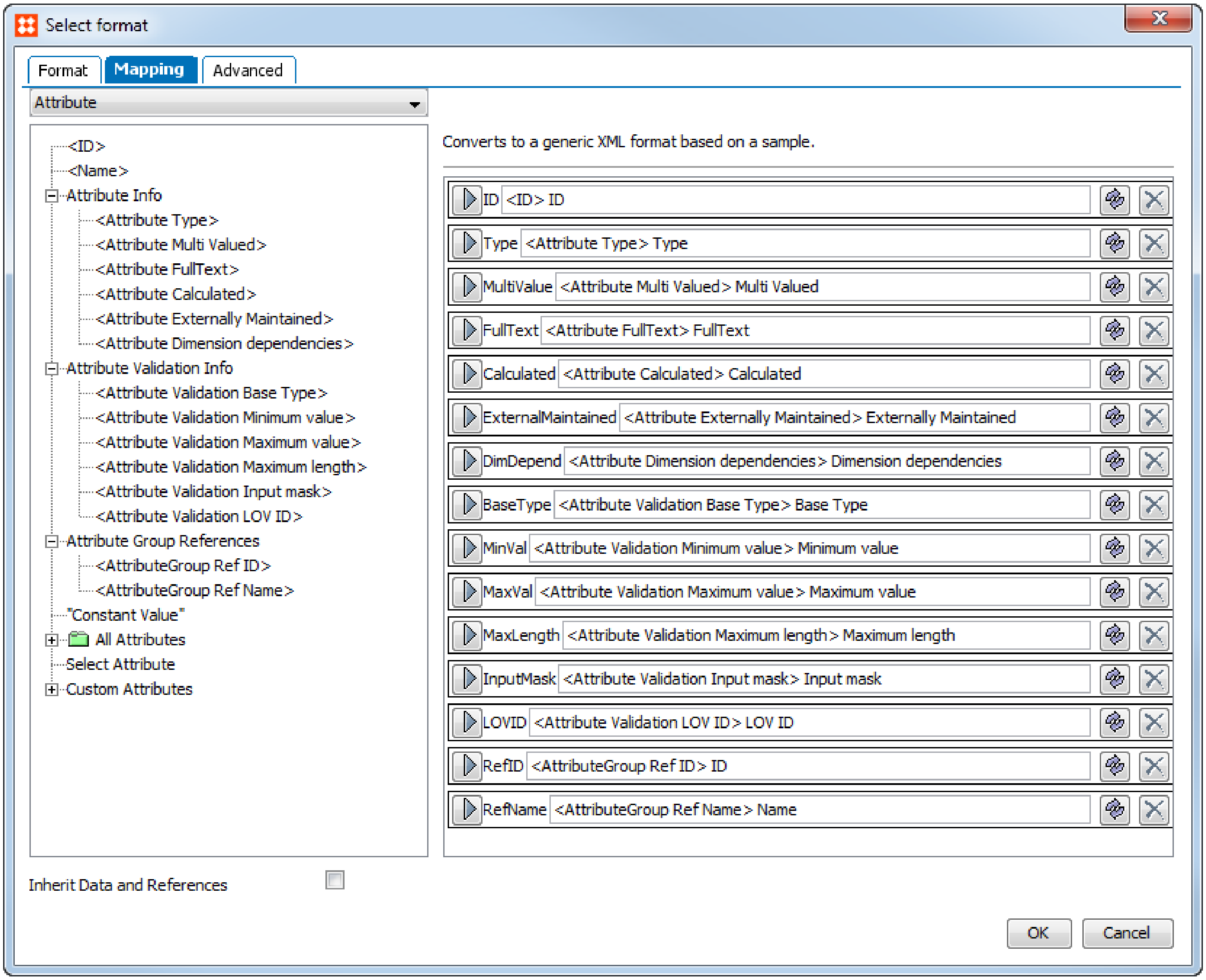
A sample output of the above mapping is shown below.



```xml
<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

Aspects, also called data sources, are displayed when mapping Attributes for export. The description of each aspect is included below.


Select format
Format
Mapping
Advanced
Attribute
<ID>
<Name>
Attribute Info
<Attribute Type>
<Attribute Multi Valued>
<Attribute FullText>
<Attribute Calculated>
<Attribute Externally Maintained>
<Attribute Dimension dependencies>
Attribute Validation Info
<Attribute Validation Base Type>
<Attribute Validation Minimum value>
<Attribute Validation Maximum value>
<Attribute Validation Maximum length>
<Attribute Validation Input mask>
<Attribute Validation LOV ID>
Attribute Group References
<AttributeGroup Ref ID>
<AttributeGroup Ref Name>
"Constant Value"
All Attributes
Select Attribute
Custom Attributes
OK
Cancel

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 □

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.

Aspect	Description
Name	Extracts only the name of the attribute and not the attribute value. For example, if an attribute has ID 1234 and name 'Weight,' then the word 'Weight' is extracted.
Attribute Type	Extracts the type of the attribute, i.e., either Description or Specification. For example, for a description attribute the word 'Description' is extracted.
Attribute Multi Valued	Extracts if the attribute is multi-valued. If the attribute is a multi-valued attribute the word 'true' is extracted, otherwise the word 'false' is extracted.
Attribute FullText	Extracts if the attribute is full text indexable. If the attribute is a full text indexable attribute the word 'true' is extracted, otherwise the word 'false' is extracted.
Attribute Calculated	Extracts if the attribute is a calculated attribute. If the attribute is a calculated attribute the word 'true' is extracted, otherwise the word 'false' is extracted.
Attribute Externally Maintained	Extracts if the attribute is externally maintained. If the attribute is externally maintained the word 'true' is extracted, otherwise the word 'false' is extracted.
Attribute Dimension Dependencies	Extracts if the attribute has dimension dependencies. If the attribute has dimension dependencies they are listed and separated by a semicolon (;).
Attribute Validation Base Type	Extracts the validation base type of the attribute. For example, if the attribute has Text as validation base type, the word 'text' is extracted.
Attribute 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/>).

Aspect	Description
Attribute Validation Maximum value	Extracts the maximum value for the attribute, if available. For example, for a number attribute with a maximum value of 500, the number '500' is extracted. For a number attribute without a maximum value, a single target tag is extracted with a trailing slash (i.e., <MaxVal/>).
Attribute Validation Maximum length	Extracts the maximum length of the attribute, if available. For example, for an attribute with a maximum length of 20, the number '20' is extracted. For a number attribute without a maximum length, a single target tag is extracted with a trailing slash (i.e., <MaxLen/>).
Attribute Validation Input mask	Extracts the attribute validation input mask, if available. For example, if an attribute has an attribute validation mask of 'F0,' then 'F0' is extracted. For an attribute without an input mask, a single target tag is extracted with a trailing slash (i.e., <InputMask/>).
Attribute Validation LOV ID	Extracts the ID of the List Of Values, if available. For example, for an attribute configured to use the List Of Value with ID Colors, the word 'Colors' is extracted. For an attribute without an LOV, a single target tag is extracted, with a trailing slash (i.e., <LOVID/>).
AttributeGroup Ref ID	Extracts a list of IDs of the attribute groups that include the attribute. Each group ID is separated by a semicolon (;). For example, if the attribute is in attribute groups with ID Group1 and ID Group2 then 'Group1;Group2' is extracted.
AttributeGroup Ref Name	Extracts a list of names of the attribute groups that include the attribute. Each group name is separated by a semicolon (;). For example, if the attribute is in attribute groups named GroupName1 and GroupName2, then 'GroupName1;GroupName2' is extracted.

For a description of the Constant Value, All Attributes, Select Attributes and Custom Attributes aspects, see the Map Data section of Exporting Data documentation.

Attribute Links and LOVs

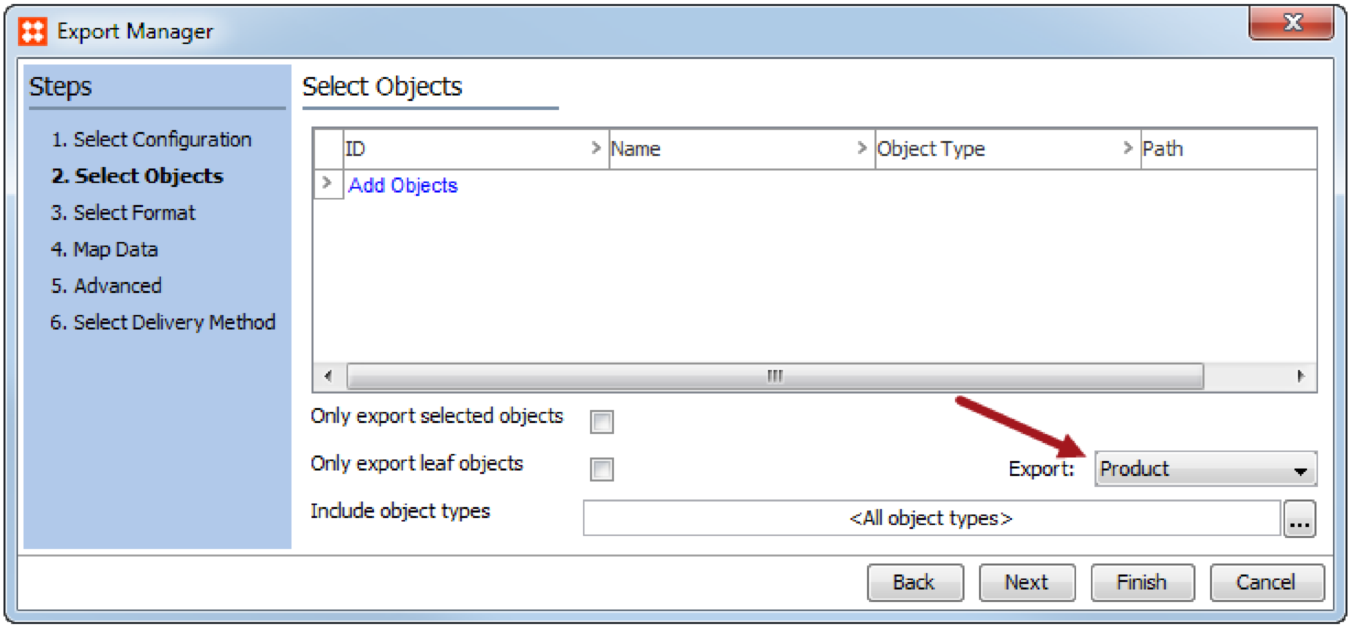
Using the following Generic XML template and mapping allows the export of System Setup information along with Product information. In this scenario, the following is exported:

- 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.

Objects

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



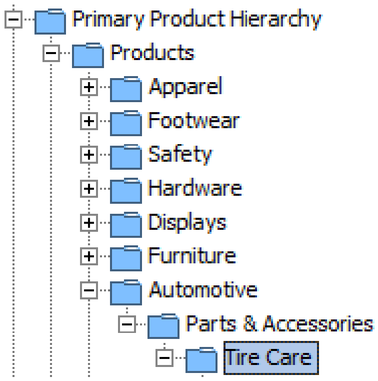
Only export selected objects □

Only export leaf objects □

Export: Product

Include object types <All object types>

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



- Primary Product Hierarchy
  - Products
    - Apparel
    - Footwear
    - Safety
    - Hardware
    - Displays
    - Furniture
    - Automotive
      - Parts & Accessories
        - Tire Care

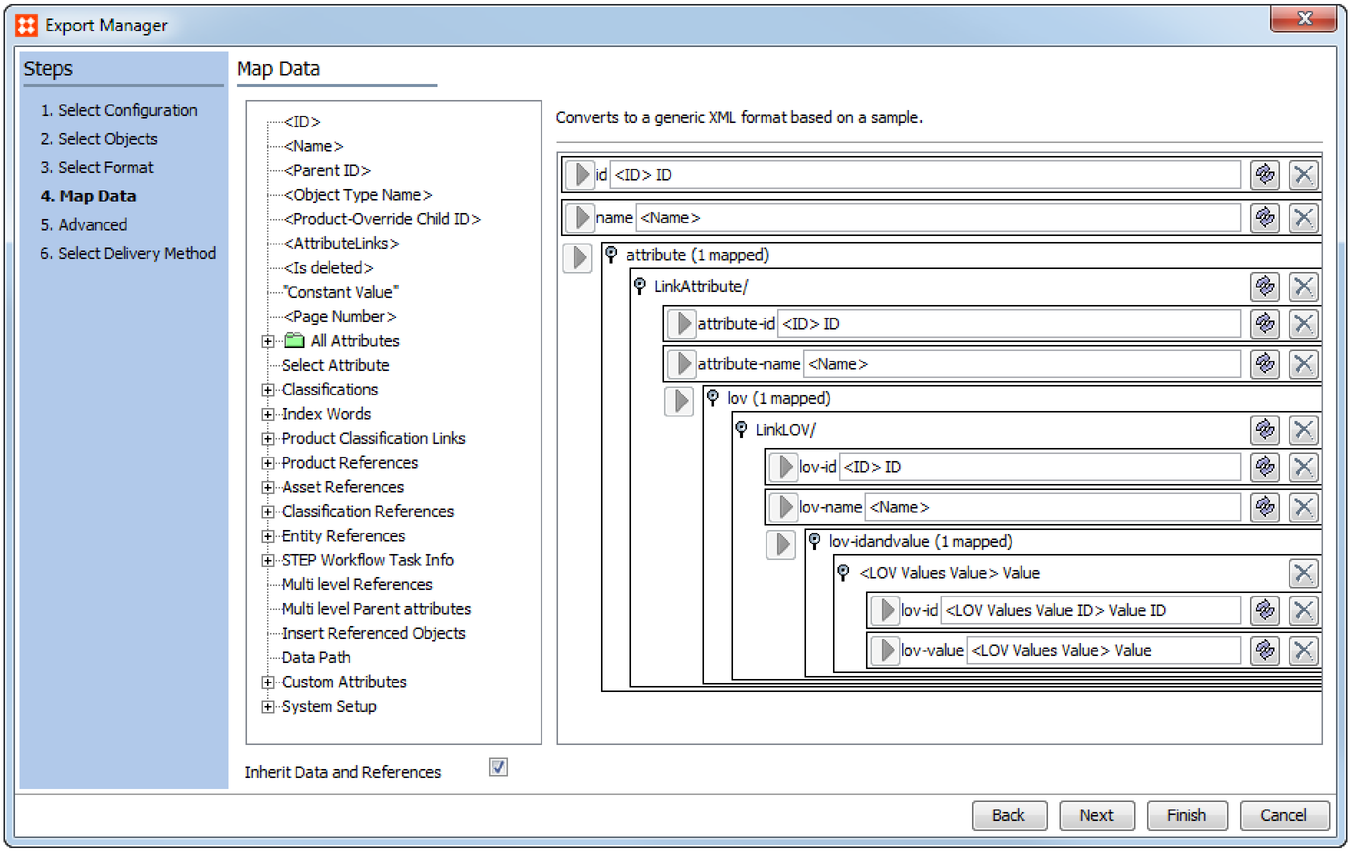
Template

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

```
<products>
   <product>
      <?Product?>
      <id><?Target?></id>
      <name><?Target?></name>
   <attribute>
      <?MultiTarget?>
      <id><?Target attribute-id?></id>
      <name><?Target attribute-name?></name>
   <lov>
      <?MultiTarget?>
      <id><?Target lov-id?></id>
      <name><?Target lov-name?></name>
   <lov-idandvalue>
      <?MultiTarget?>
      <value><?Target lov-value?></value>
      <id><?Target lov-id?></id>
   </lov-idandvalue>
   </lov>
   </attribute>
   </product>
</products>
```

Mapping

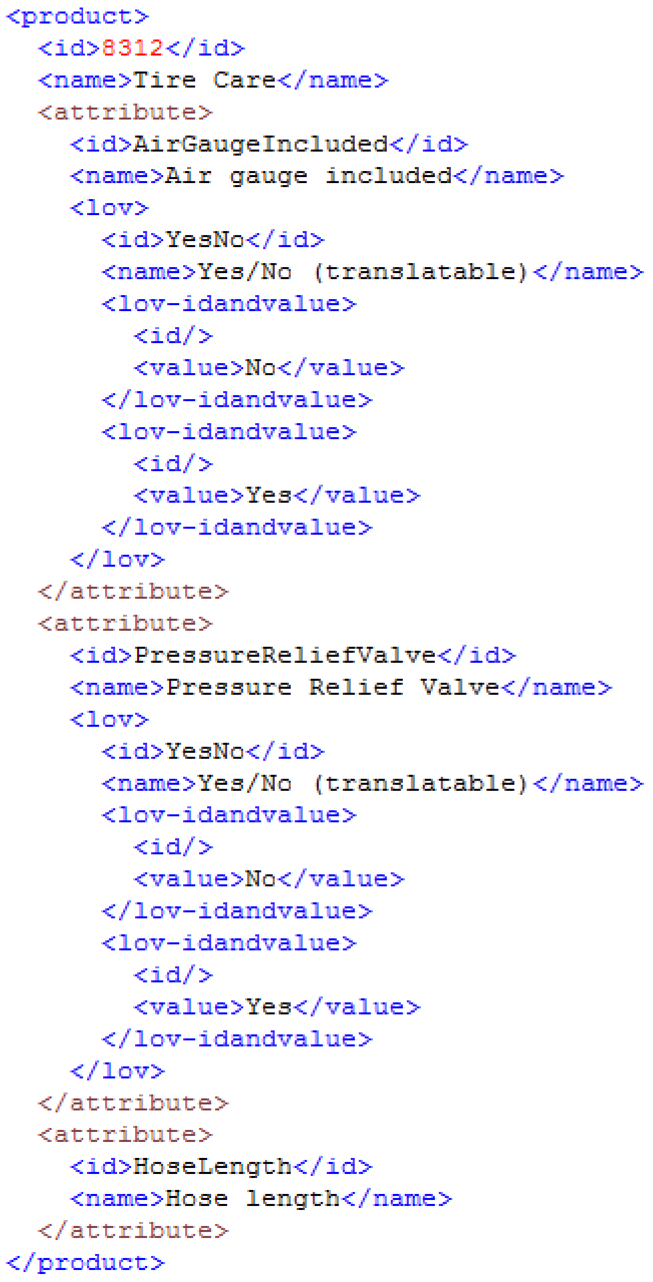
The Data Path aspect enables the complete set of mapping rules displayed below:



See the 'Mapping Attribute Links, LOVs, and LOV Values' section of the STEP Export Manager User Guide / Exporting Data and Images documentation for detailed mapping steps.

Results

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

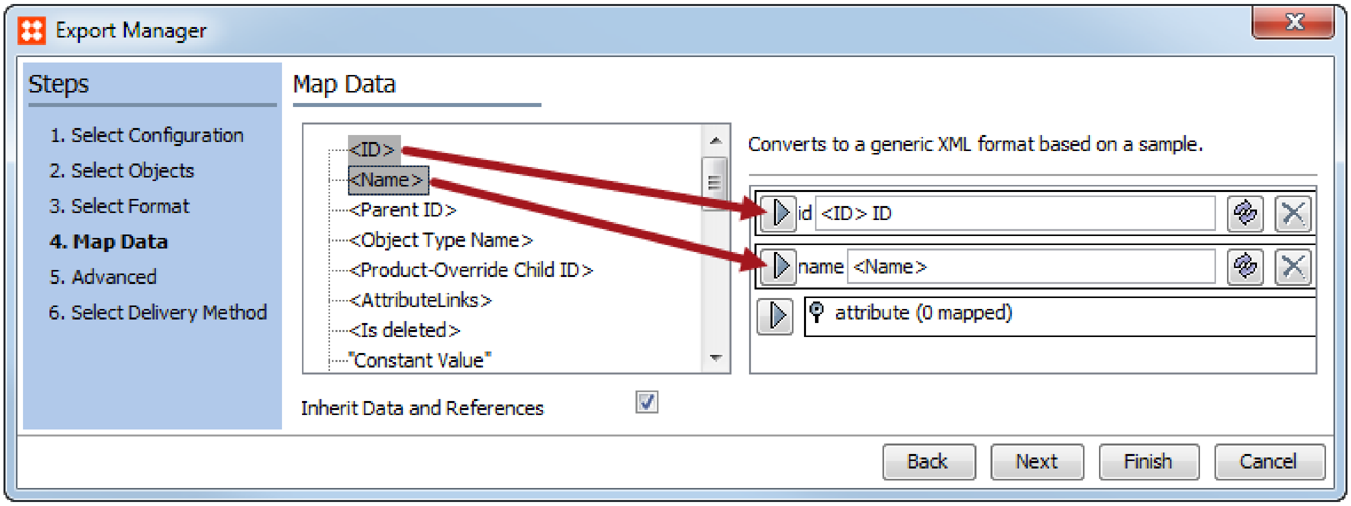


```
<product>
  <id>8312</id>
  <name>Tire Care</name>
  <attribute>
    <id>AirGaugeIncluded</id>
    <name>Air gauge included</name>
    <lov>
      <id>YesNo</id>
      <name>Yes/No (translatable)</name>
      <lov-idandvalue>
        <id/>
        <value>No</value>
      </lov-idandvalue>
      <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

The Attribute Links and LOVs section discusses additional setup for a Generic XML export. Use the following steps to map the Product, linked attributes, and their LOV information.

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



Converts to a generic XML format based on a sample.

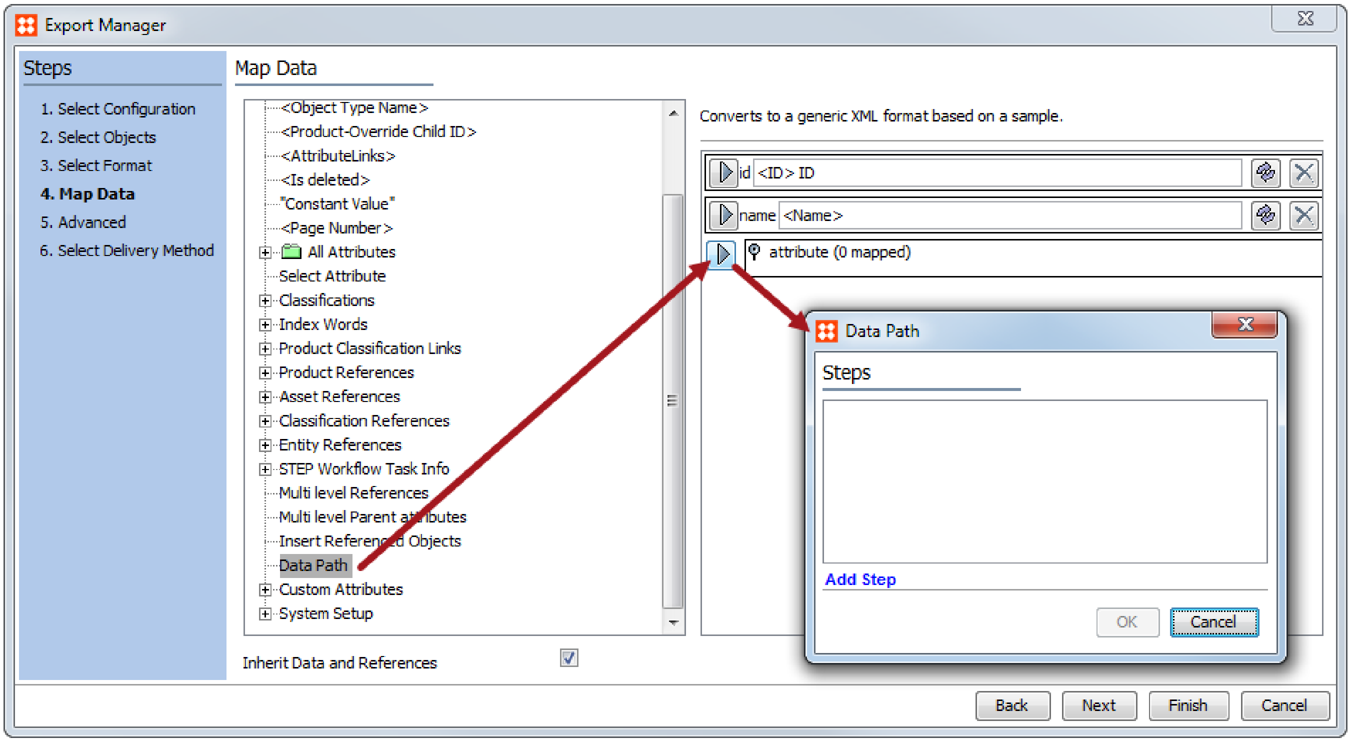
id <ID> ID

name <Name>

attribute (0 mapped)

Inherit Data and References ☑

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



Converts to a generic XML format based on a sample.

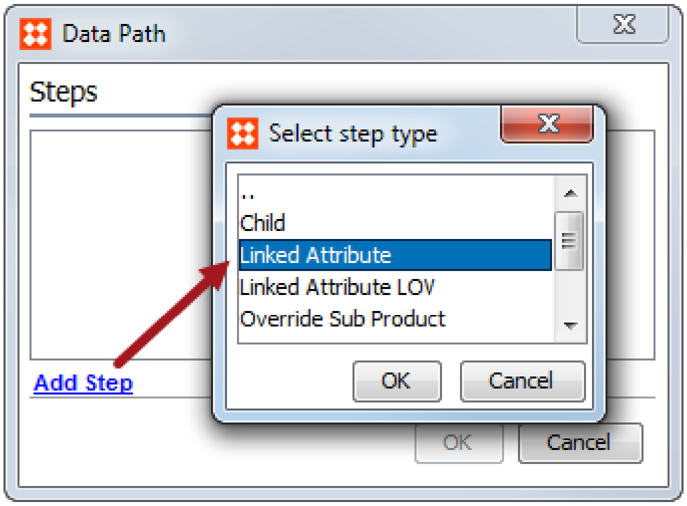
id <ID> ID

name <Name>

attribute (0 mapped)

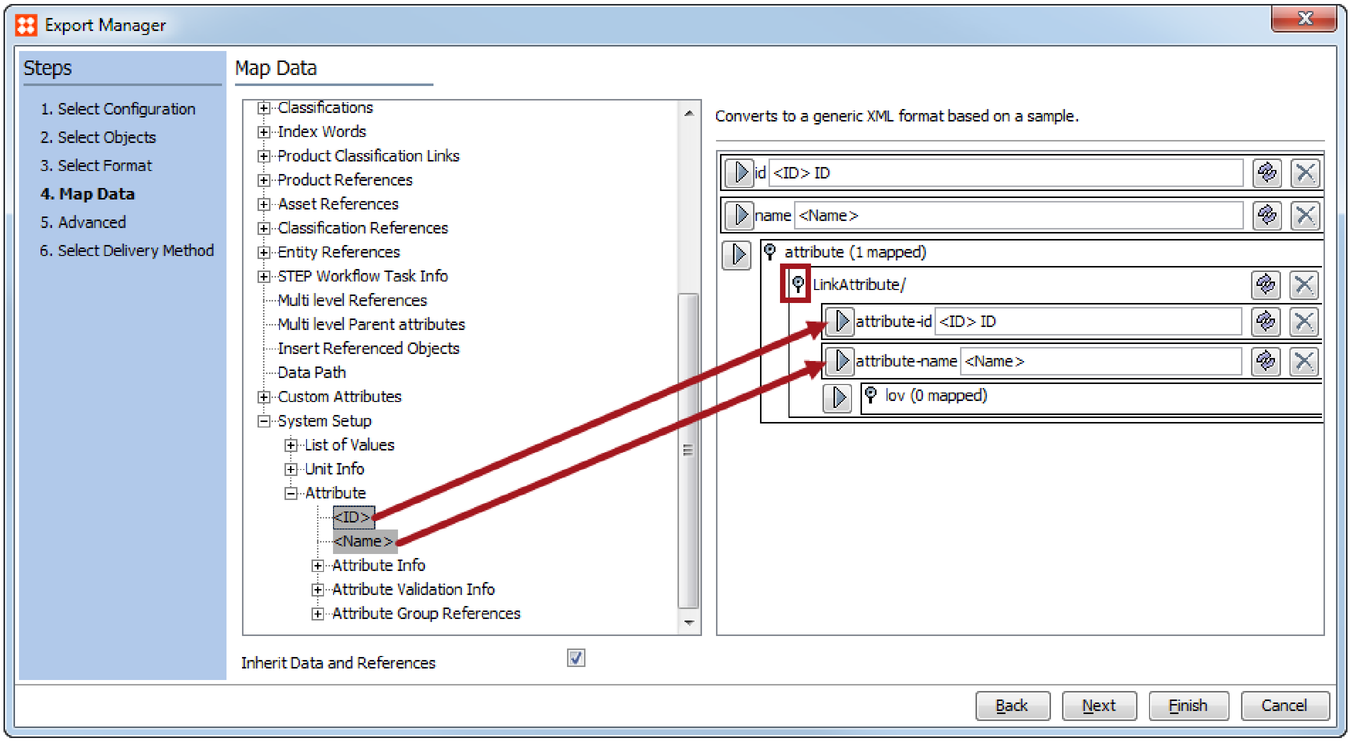
Inherit Data and References ☑

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**.



Converts to a generic XML format based on a sample.

id <ID> ID

name <Name>

attribute (1 mapped)

LinkAttribute/

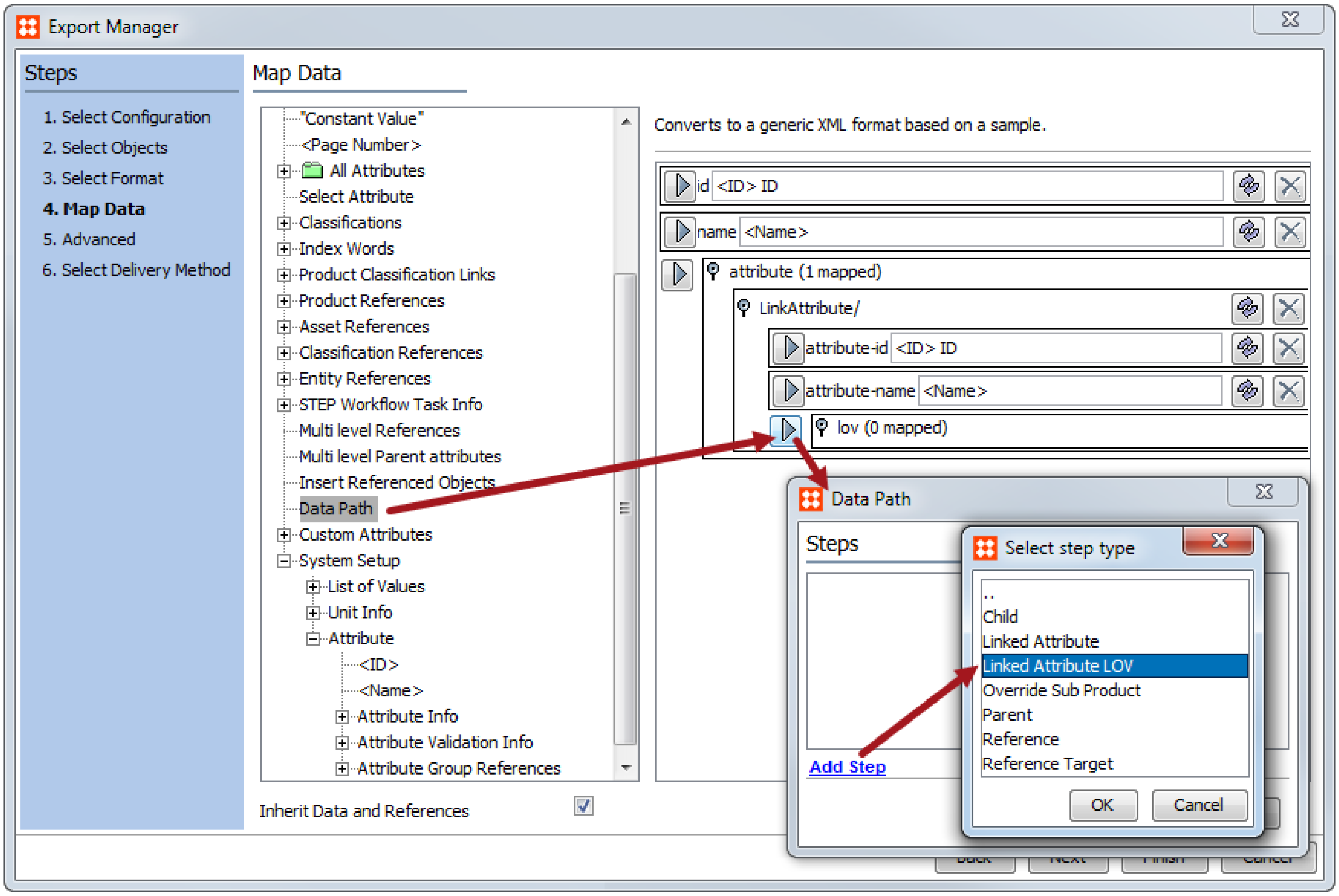
attribute-id <ID> ID

attribute-name <Name>

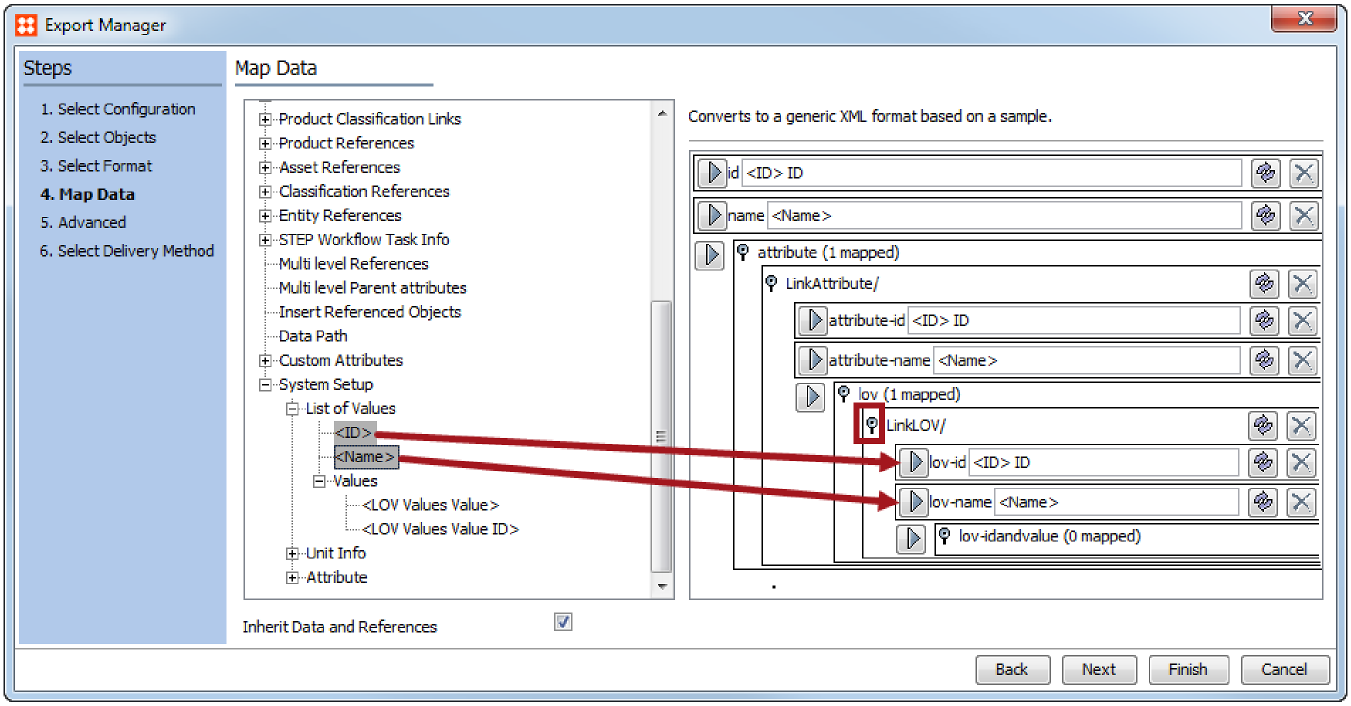
lov (0 mapped)

Inherit Data and References ☑

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**.

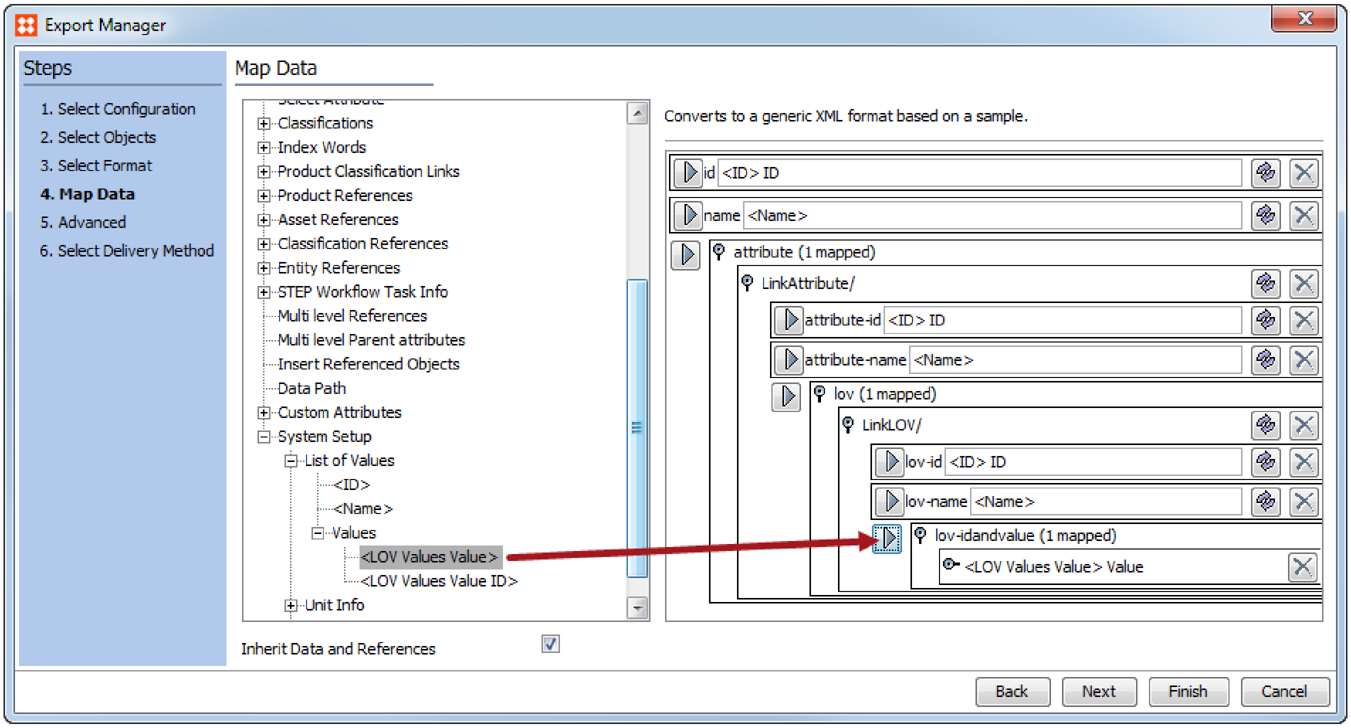


Converts to a generic XML format based on a sample.

id <ID> ID
name <Name>
attribute (1 mapped)
LinkAttribute/
attribute-id <ID> ID
attribute-name <Name>
lov (1 mapped)
LinkLOV/
lov-id <ID> ID
lov-name <Name>
lov-idandvalue (0 mapped)

Inherit Data and References ☑

9. Map **LOV Values Value** to lov-idandvalue.

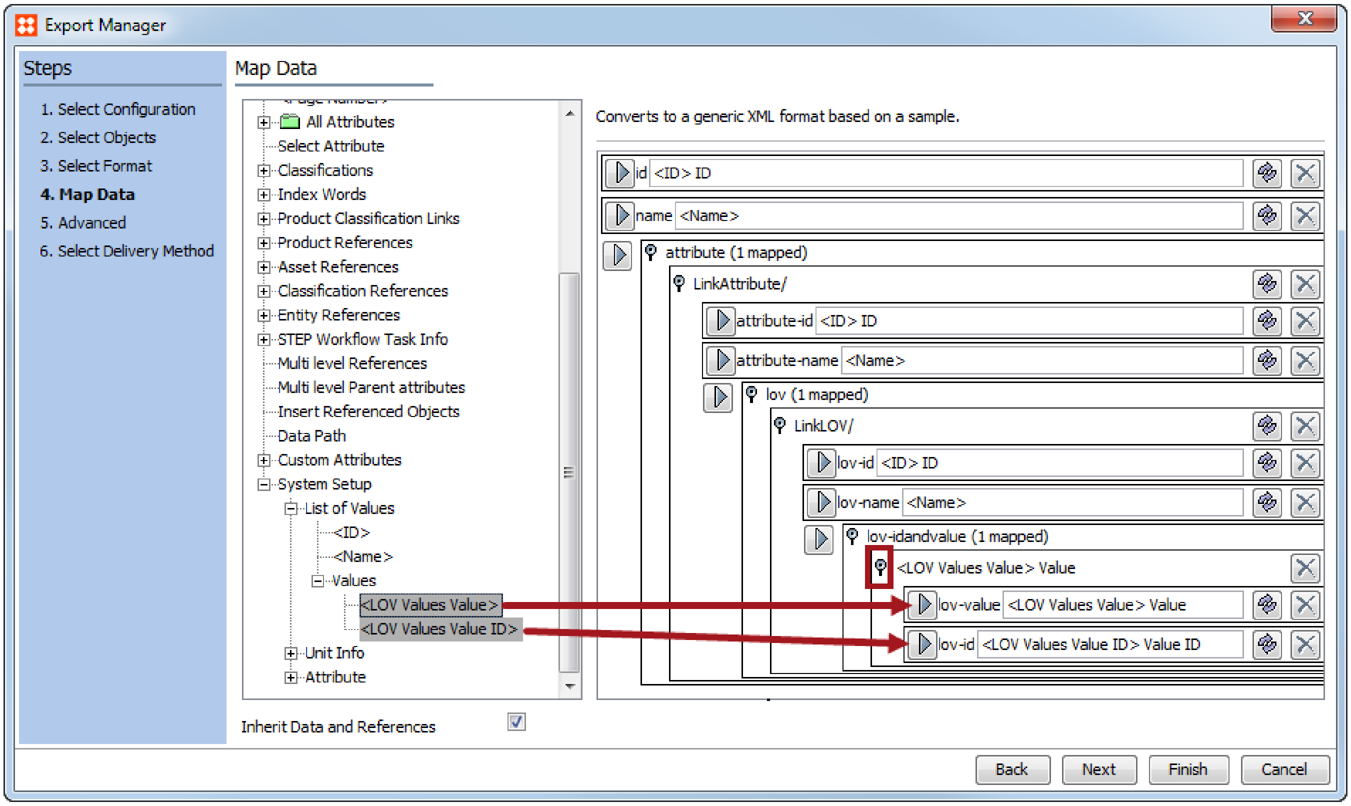


Converts to a generic XML format based on a sample.

id <ID> ID
name <Name>
attribute (1 mapped)
LinkAttribute/
attribute-id <ID> ID
attribute-name <Name>
lov (1 mapped)
LinkLOV/
lov-id <ID> ID
lov-name <Name>
lov-idandvalue (1 mapped)
<LOV Values Value> Value

Inherit Data and References ☑

10. Open the <LOV Values Value> Value flipper and map **LOV Values Value** and **LOV Values ID**.



Converts to a generic XML format based on a sample.

```
id <ID> ID
name <Name>
attribute (1 mapped)
  LinkAttribute/
    attribute-id <ID> ID
    attribute-name <Name>
    lov (1 mapped)
      LinkLOV/
        lov-id <ID> ID
        lov-name <Name>
        lov-idandvalue (1 mapped)
          <LOV Values Value> Value
            lov-value <LOV Values Value> Value
            lov-id <LOV Values Value ID> Value ID
```

Inherit Data and References ☑

Attribute Links and Units

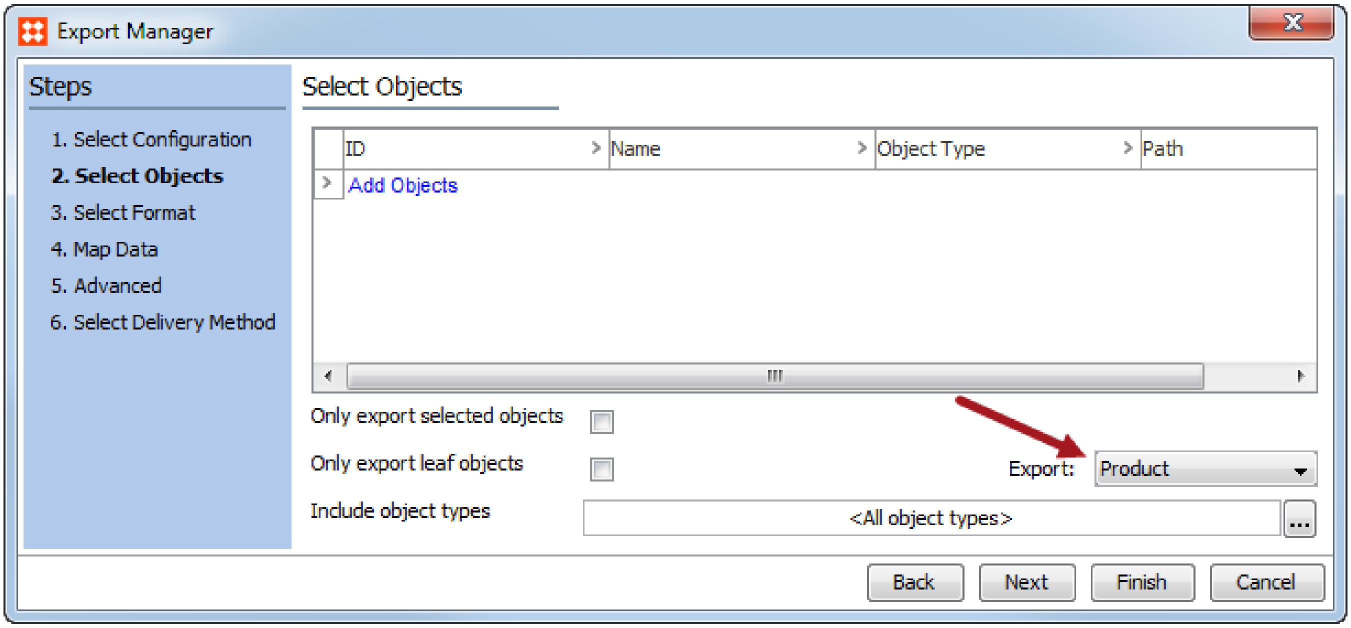
Using the following Generic XML template and mapping allows the export of System Setup information along with Product information. In this scenario, the following is exported:

- 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.

Objects

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



ID	Name	Object Type	Path
Add Objects			

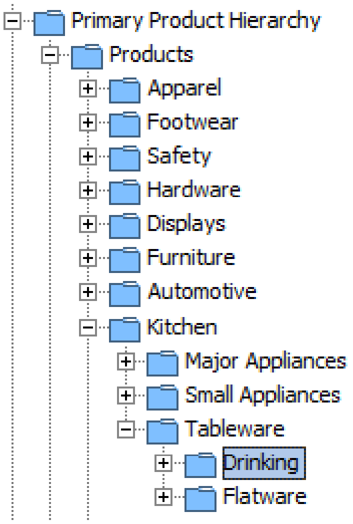
Only export selected objects □

Only export leaf objects □

Export: Product

Include object types <All object types> ...

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



- Primary Product Hierarchy
  - Products
    - Apparel
    - Footwear
    - Safety
    - Hardware
    - Displays
    - Furniture
    - Automotive
    - Kitchen
      - Major Appliances
      - Small Appliances
      - Tableware
        - Drinking
        - Flatware

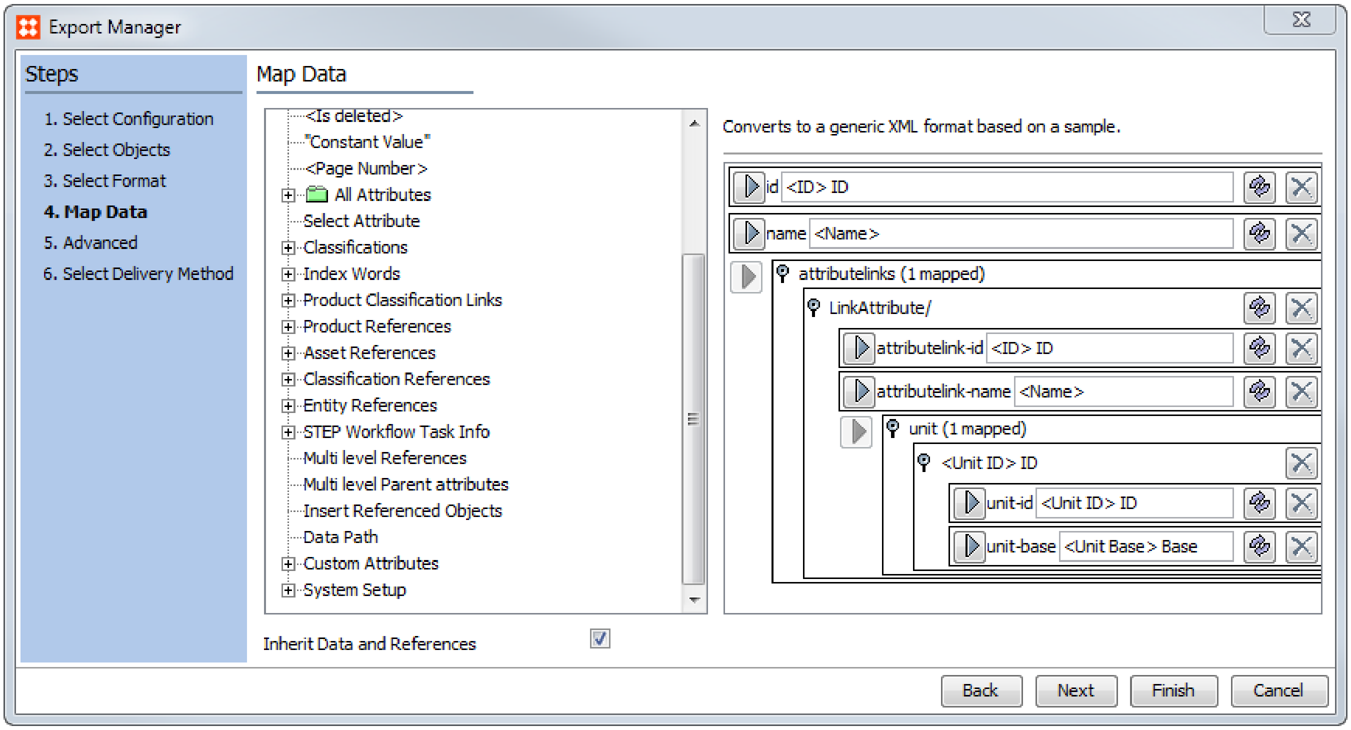
Template

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

```
<products>
   <product>
      <?Product?>
      <id><?Target?></id>
      <name><?Target?></name>
   <attributelinks>
      <?MultiTarget?>
      <id><?Target attributelink-id?></id>
      <name><?Target attributelink-name?></name>
   <unit>
      <?MultiTarget?>
      <attributelink-unit-id><?Target unit-id?></attributelink-unit-id>
      <attributelink-unit-base><?Target unit-base?></attributelink-unit-base>
   </unit>
   </attributelinks>
   </product>
</products>
```

Mapping

The Data Path aspect enables the complete set of mapping rules displayed below:



Export Manager

Steps

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

Map Data

Converts to a generic XML format based on a sample.

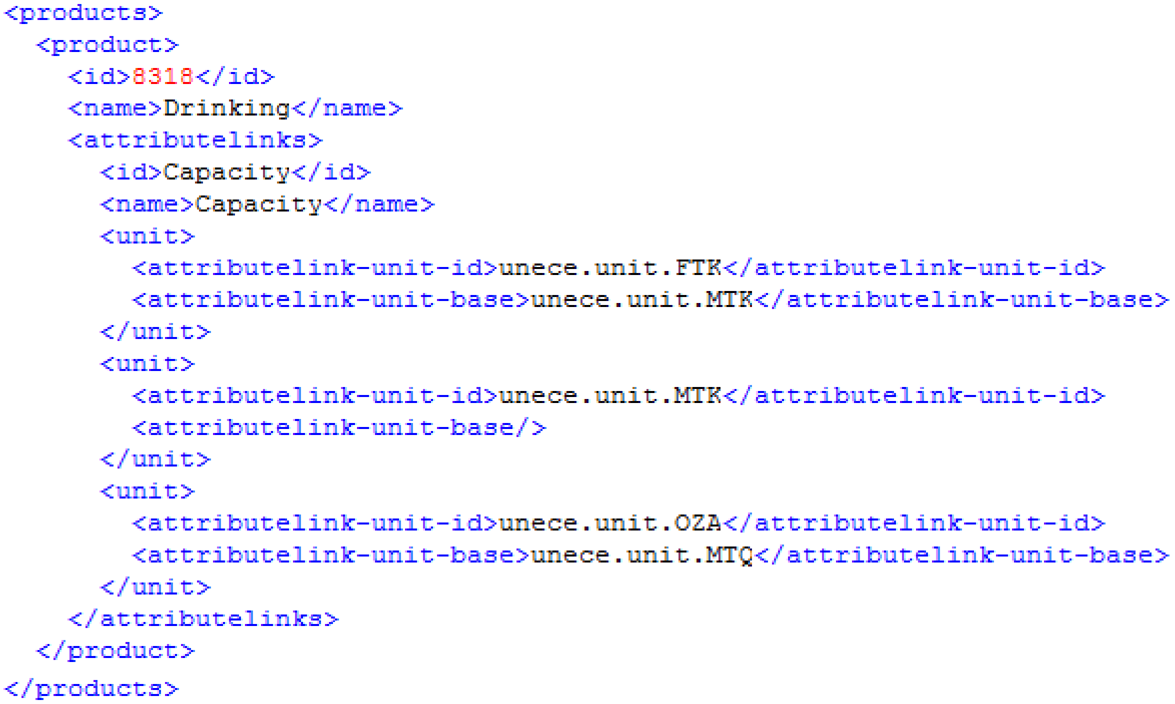
- id <ID> ID
- name <Name>
- attributelinks (1 mapped)
  - LinkAttribute/
    - attributelink-id <ID> ID
    - attributelink-name <Name>
    - unit (1 mapped)
      - <Unit ID> ID
        - unit-id <Unit ID> ID
        - unit-base <Unit Base> Base

Inherit Data and References ☑

See the 'Mapping Attribute Links, Units, and Unit Base' section of the STEP Export Manager User Guide / Exporting Data and Images documentation for detailed mapping steps.

Results

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

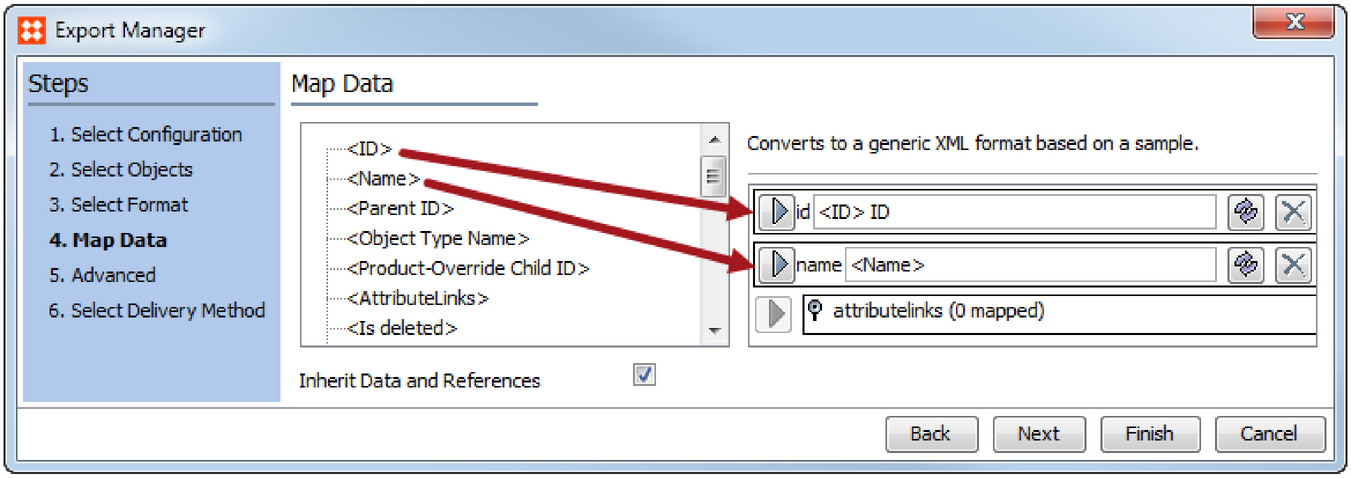


```
<products>
  <product>
    <id>8318</id>
    <name>Drinking</name>
    <attributelinks>
      <id>Capacity</id>
      <name>Capacity</name>
      <unit>
        <attributelink-unit-id>unece.unit.FTK</attributelink-unit-id>
        <attributelink-unit-base>unece.unit.MTK</attributelink-unit-base>
      </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>
```

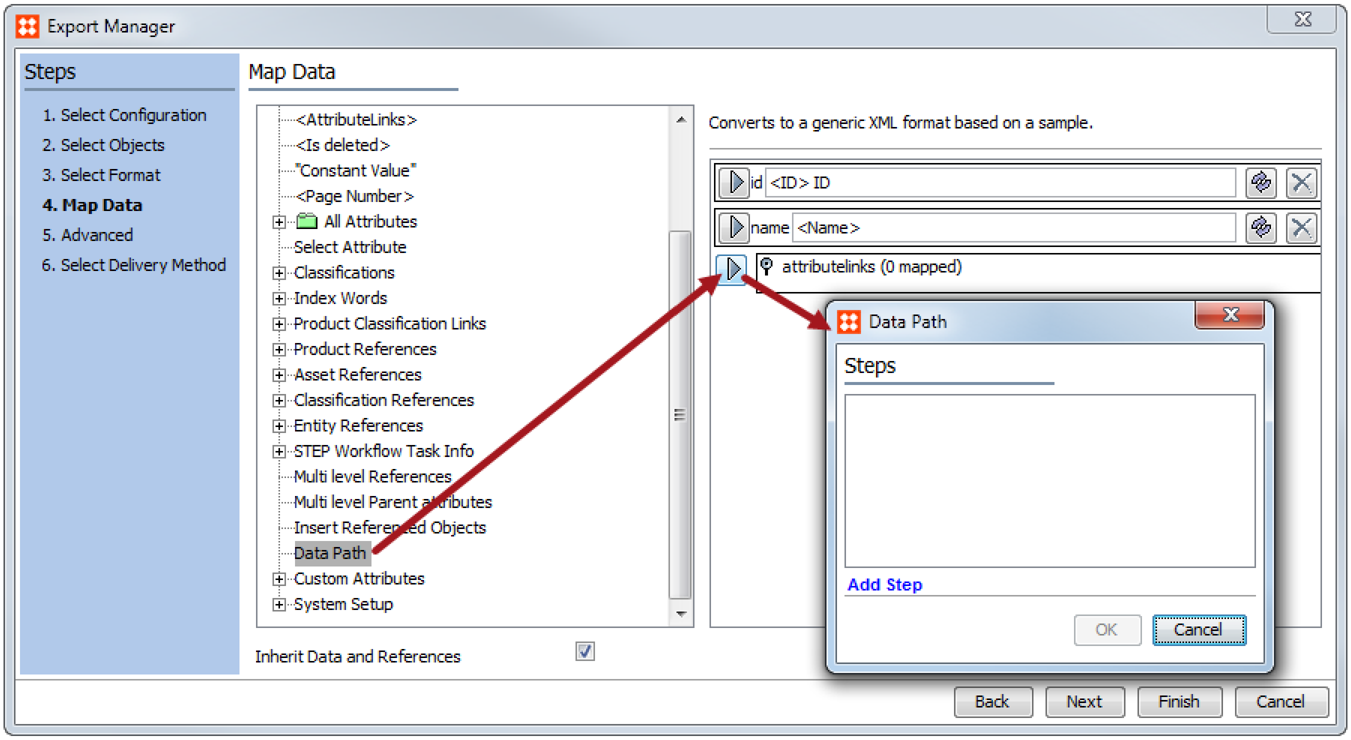
Mapping Attribute Links, Units, and Unit Base

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

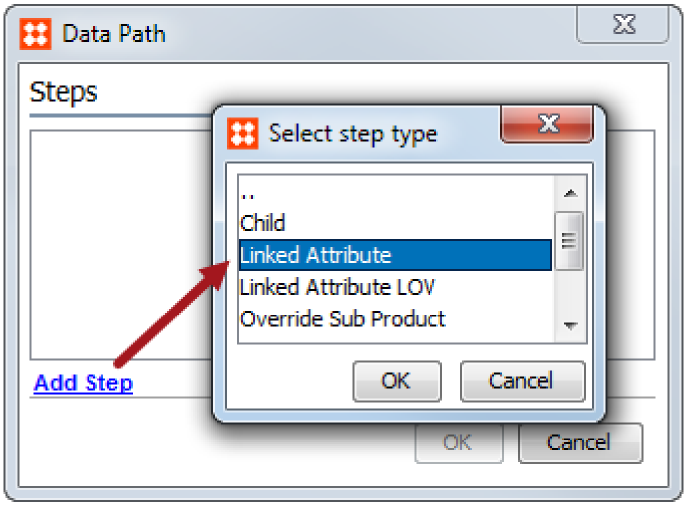
1. Map **ID** and **Name** for the Product.



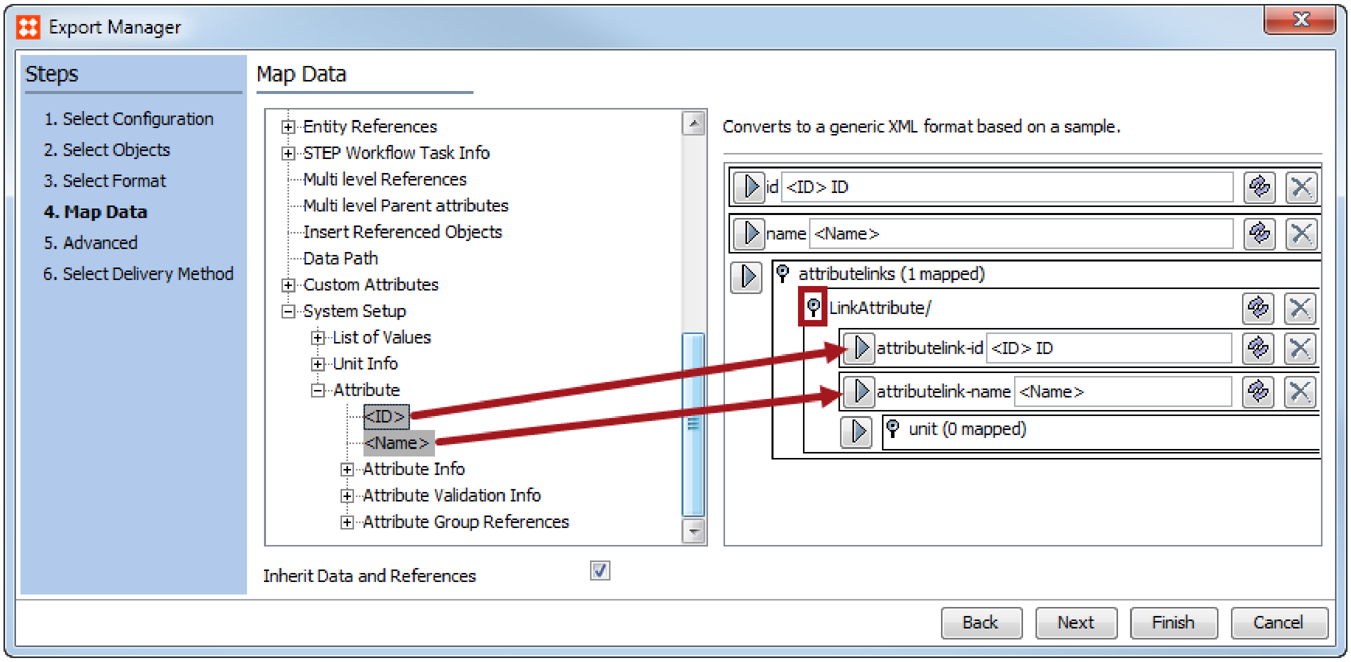
2. Map **Data Path** to attributelinks.



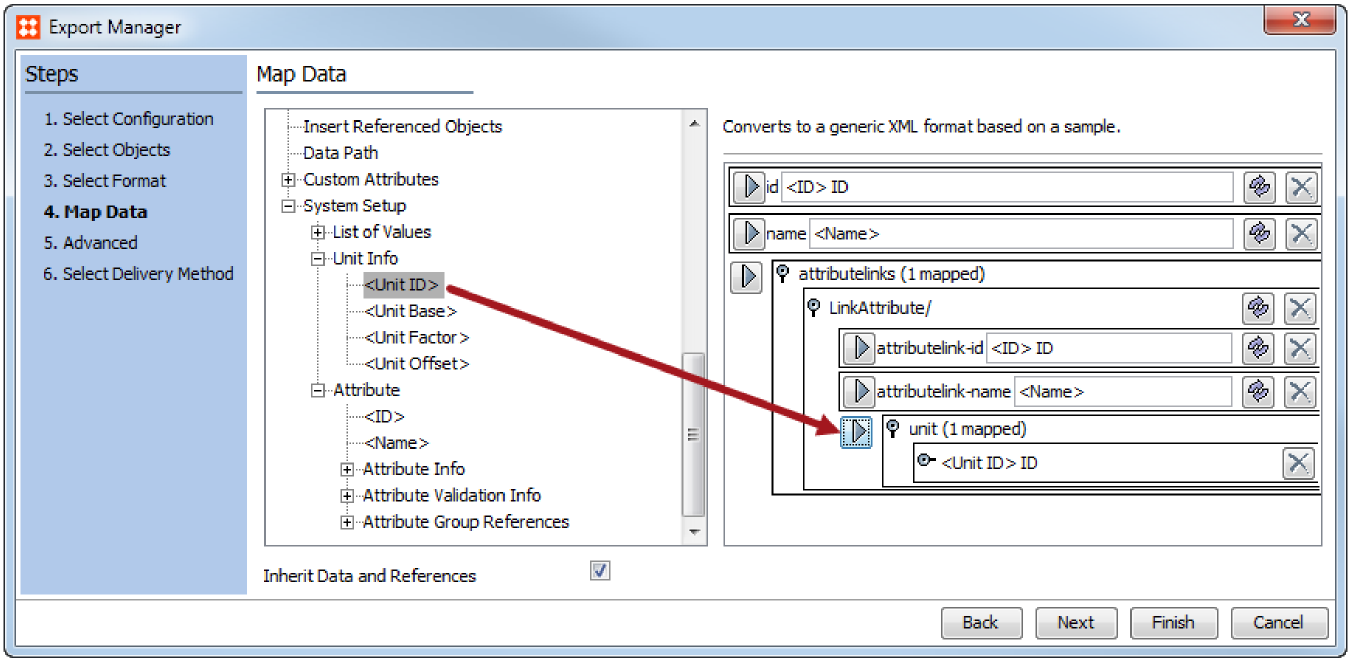
3. Click the **Add Step** link and select **Linked Attribute**.



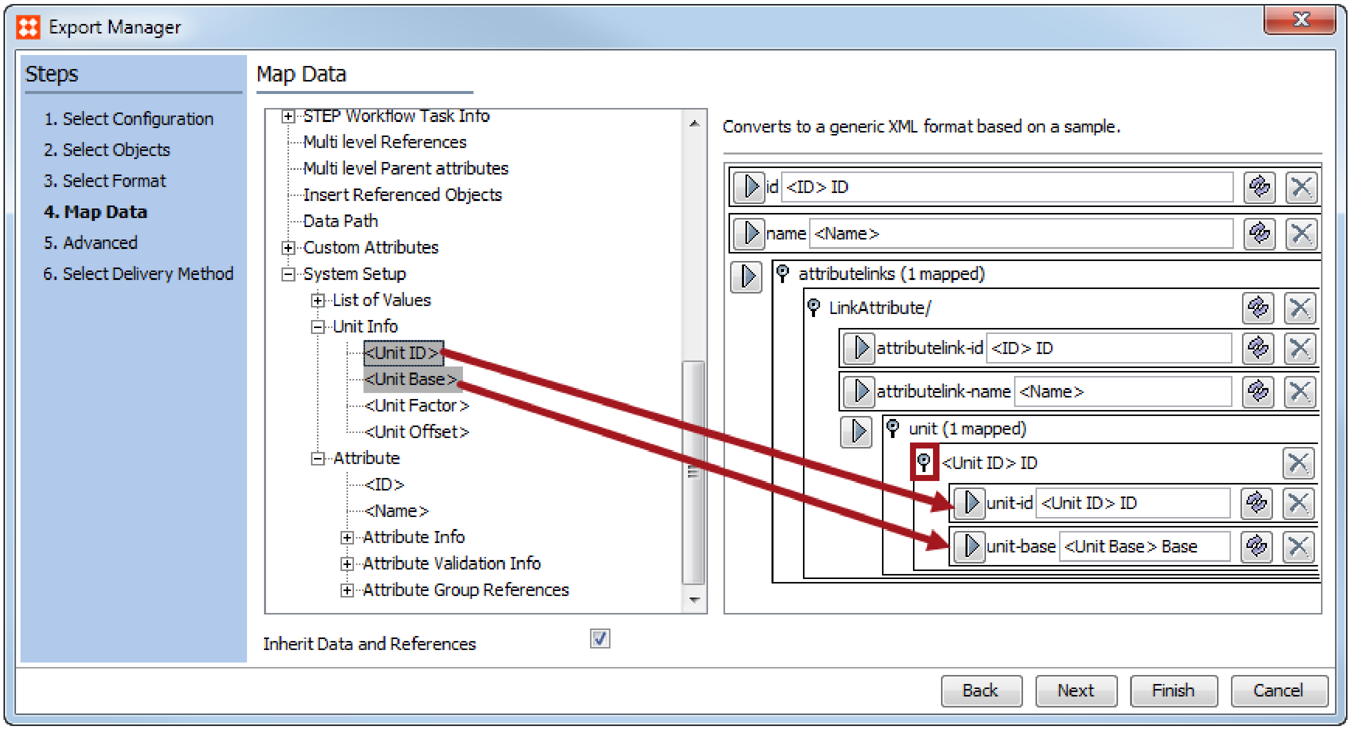
4. Click OK to close the **Select step type** and **Data Path** dialogs.
5. Open the LinkAttribute/ flipper and map Attribute **ID** and **Name**.



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



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

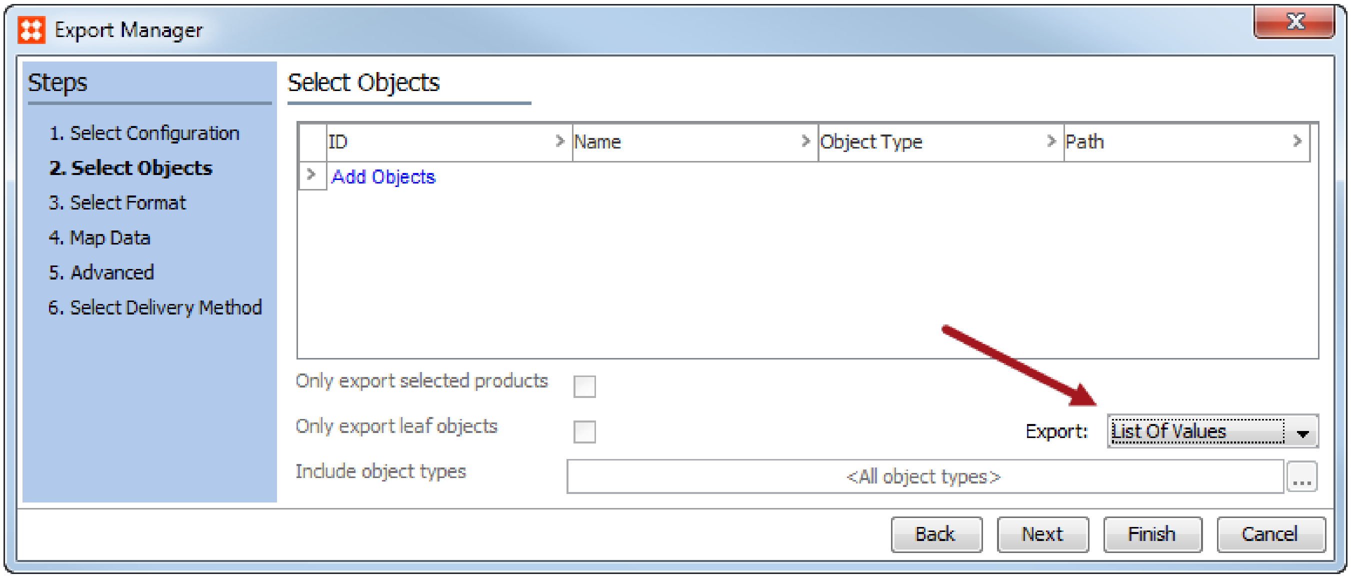


List of Values

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.



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

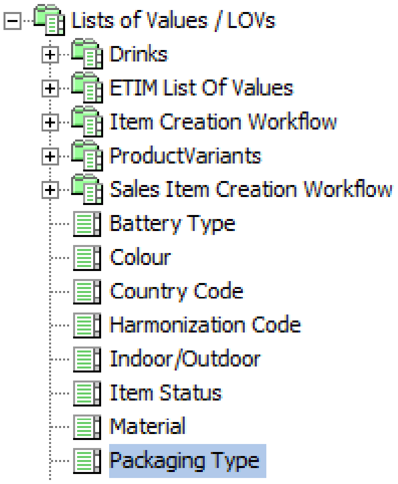
Only export selected products □

Only export leaf objects □

Export: List Of Values

Include object types <All object types>

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



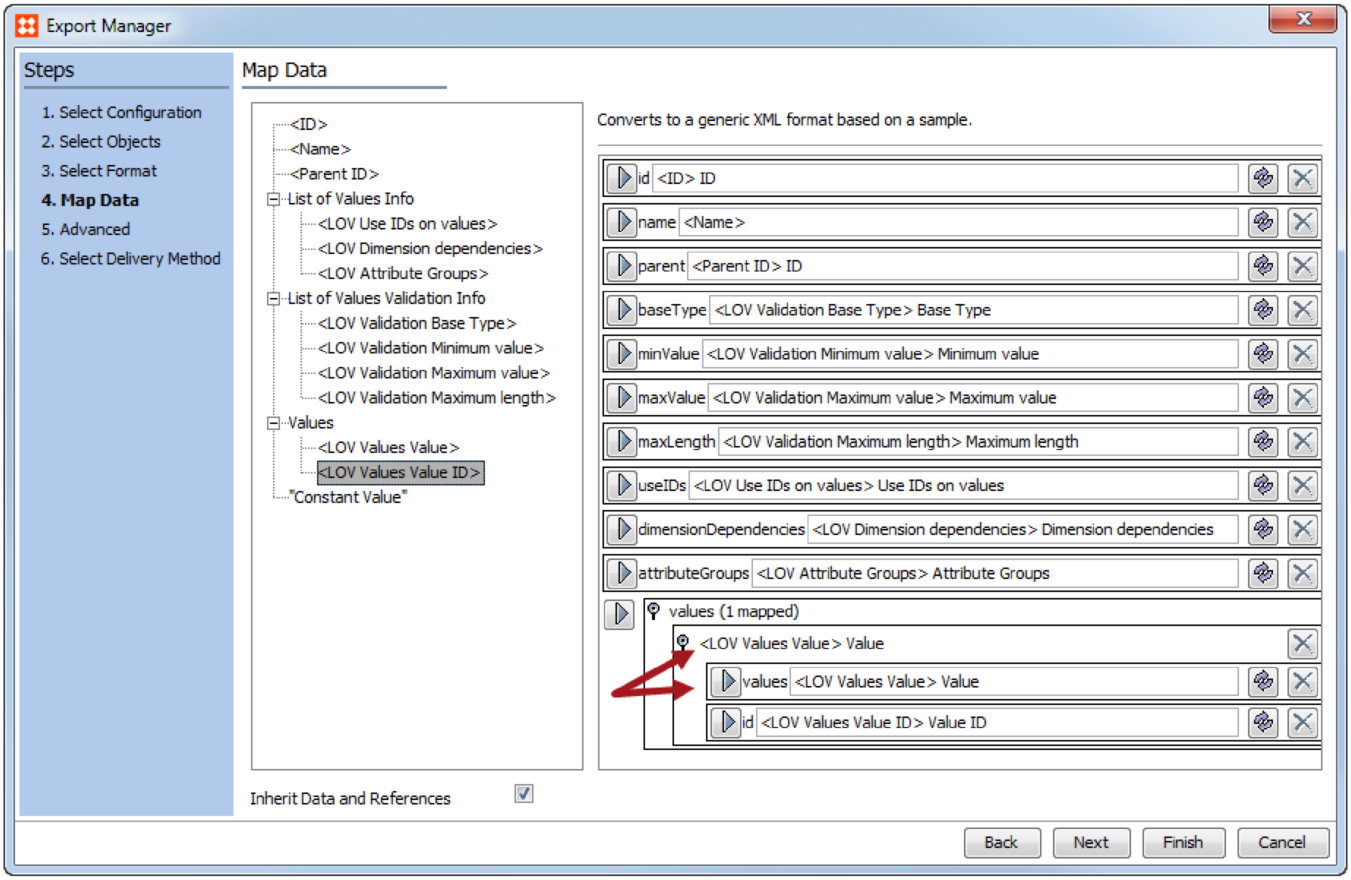
- Lists of Values / LOVs
  - Drinks
  - ETIM List Of Values
  - Item Creation Workflow
  - ProductVariants
  - Sales Item Creation Workflow
  - Battery Type
  - Colour
  - Country Code
  - Harmonization Code
  - Indoor/Outdoor
  - Item Status
  - Material
  - Packaging Type

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



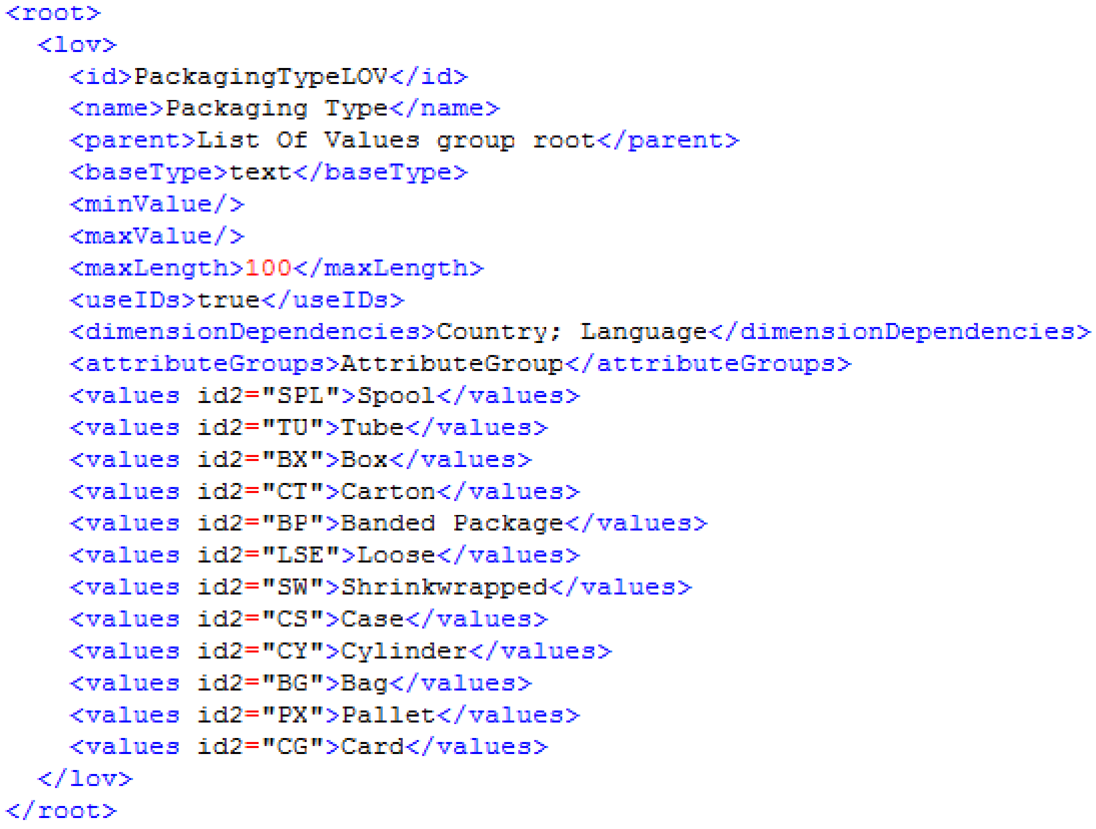
Converts to a generic XML format based on a sample.

id <ID> ID
name <Name>
parent <Parent ID> ID
baseType <LOV Validation Base Type> Base Type
minValue <LOV Validation Minimum value> Minimum value
maxValue <LOV Validation Maximum value> Maximum value
maxLength <LOV Validation Maximum length> Maximum length
useIDs <LOV Use IDs on values> Use IDs on values
dimensionDependencies <LOV Dimension dependencies> Dimension dependencies
attributeGroups <LOV Attribute Groups> Attribute Groups
values (1 mapped)
<LOV Values Value> Value
values <LOV Values Value> Value
id <LOV Values Value ID> Value ID

Inherit Data and References ☑

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>
```

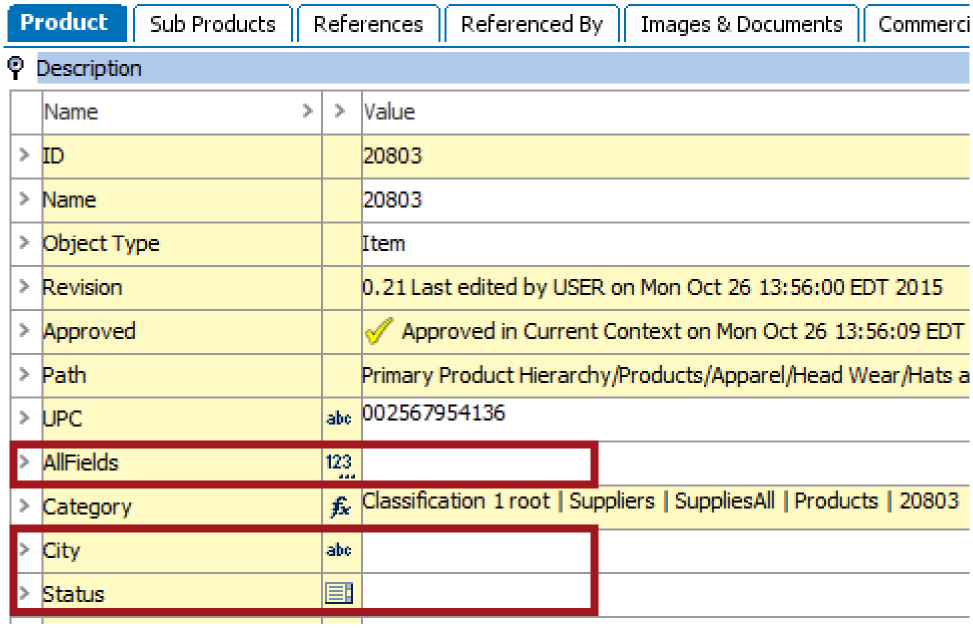
Allow Empty Tags

The 'Allow empty tags' option exports mapped Attributes even when no value is present.

If tags that do not contain children values should be removed, see the **RemoveIfEmpty Tag** section.

Objects

The object selected for export includes Attributes without values.



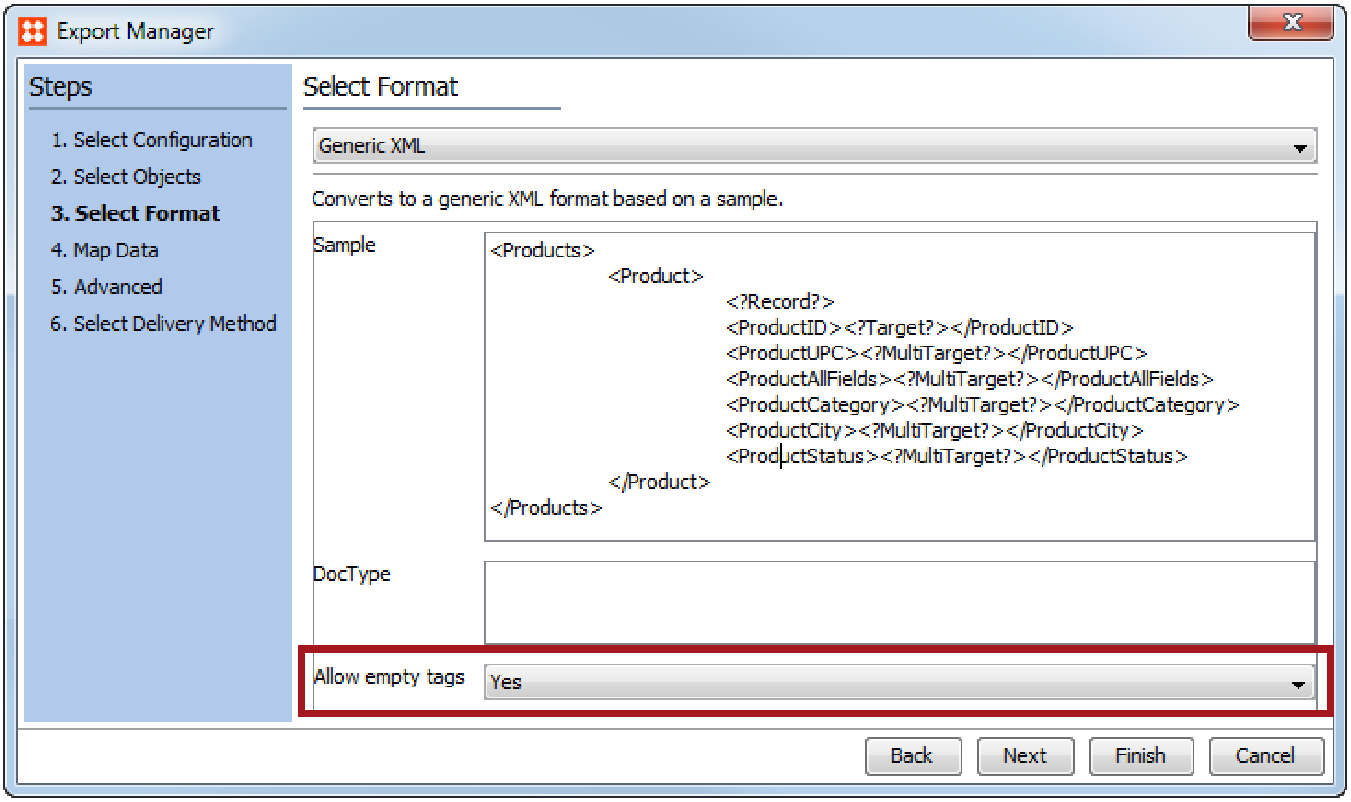
Name		Value
ID		20803
Name		20803
Object Type		Item
Revision		0.21 Last edited by USER on Mon Oct 26 13:56:00 EDT 2015
Approved		✓ Approved in Current Context on Mon Oct 26 13:56:09 EDT
Path		Primary Product Hierarchy/Products/Apparel/Head Wear/Hats a
UPC	abc	002567954136
AllFields	123	
Category	fx	Classification 1 root | Suppliers | SuppliesAll | Products | 20803
City	abc	
Status	▤	

Template

Choose **Generic XML** format and provide the following template text 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** to **Yes**.



Generic XML

Converts to a generic XML format based on a sample.

Sample

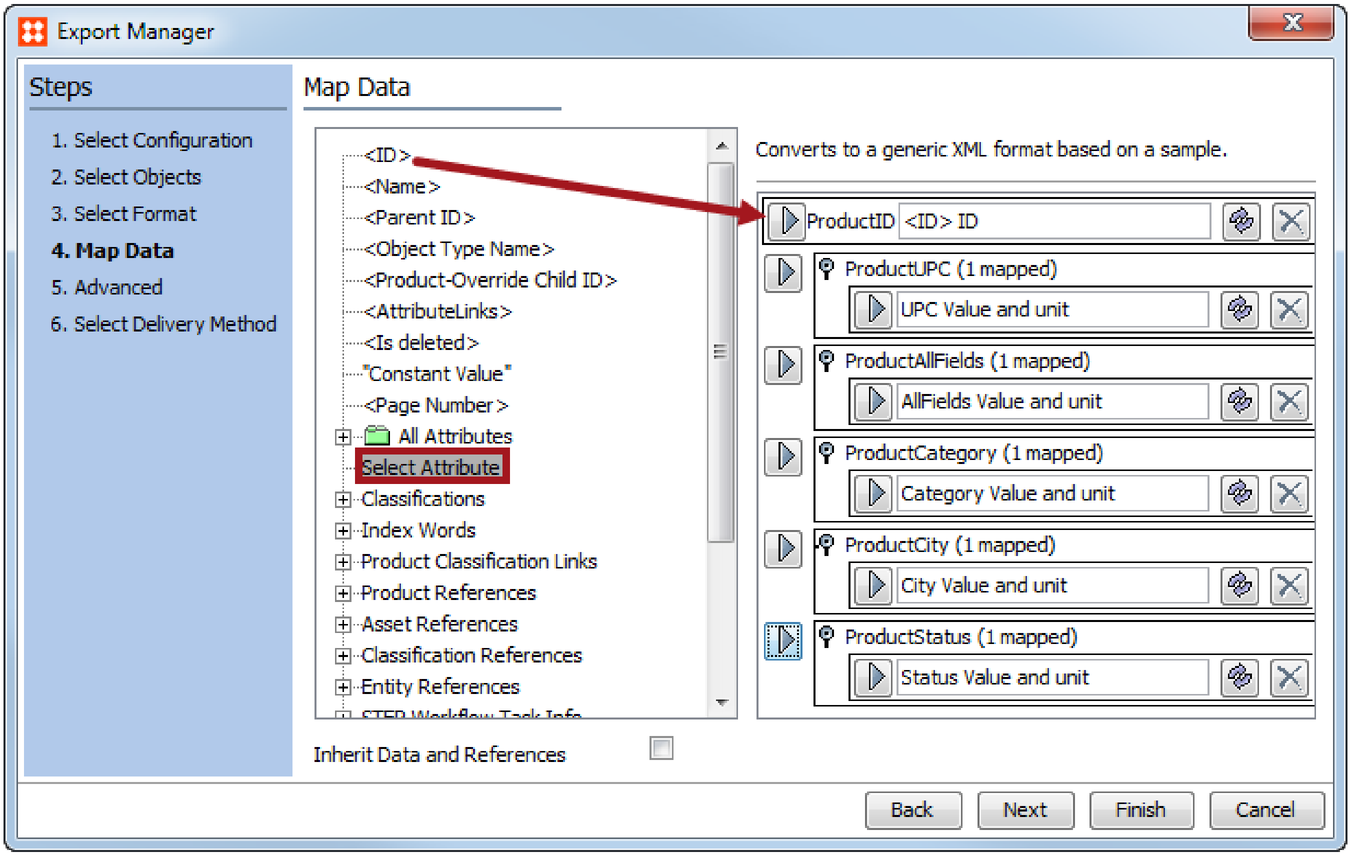
```
<Products>
        <Product>
                <?Record?>
                <ProductID><?Target?></ProductID>
                <ProductUPC><?MultiTarget?></ProductUPC>
                <ProductAllFields><?MultiTarget?></ProductAllFields>
                <ProductCategory><?MultiTarget?></ProductCategory>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

DocType

Allow empty tags: Yes

Mapping

Use the **Select Attribute** aspect for the attribute data targets after ID.



Results

Empty 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>
```

Multi Valued

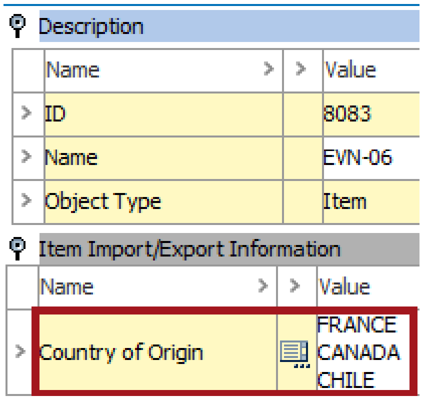
There are two Generic XML options for exporting fields that allow multiple values.

- 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:

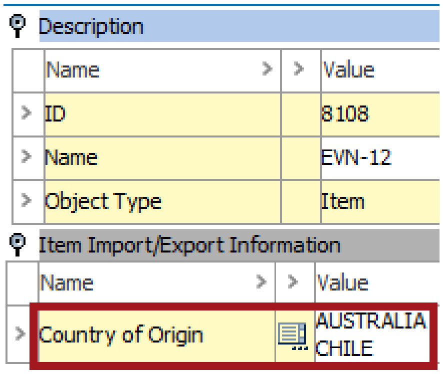


Description

Name	Value
ID	8083
Name	EVN-06
Object Type	Item

Item Import/Export Information

Name	Value
Country of Origin	FRANCE CANADA CHILE

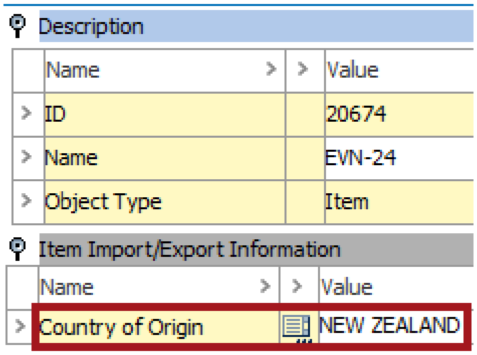


Description

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

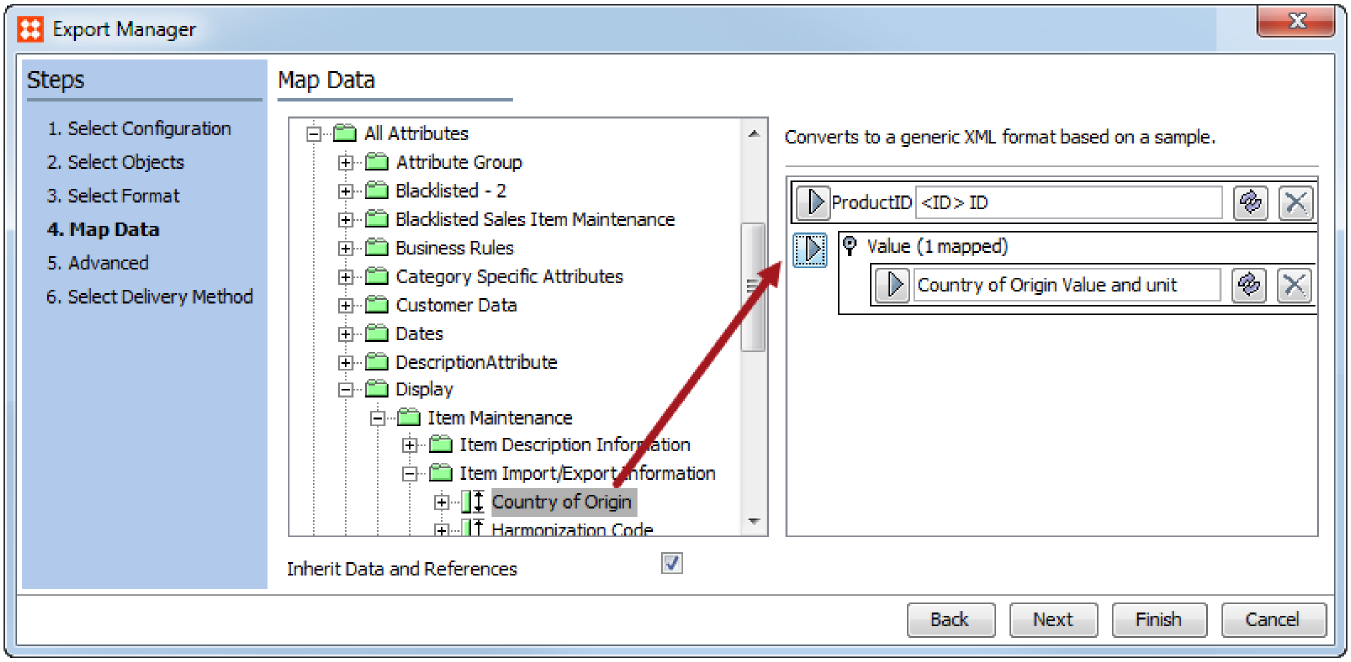
Name	Value
Country of Origin	NEW ZEALAND

Template

Repeated Target	MultiTarget
<Products> <Product> <?Record?> <ProductID><?Target?></ProductID> <Value><?Target?></Value> <Value><?Target?></Value> </Product> </Products>	<Products> <Product> <?Record?> <ProductID><?Target?></ProductID> <Value><?MultiTarget?></Value> </Product> </Products>

Mapping

The mapping steps are the same regardless of the Multi Valued option used.



Export Manager

Steps

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

Map Data

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

`<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>`
	`<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>`

Nestable Data Paths

The Nestable Data Path feature is only available with Export Formats that support output of nested repeated structures, such as Generic XML.

The Nestable Data Path Example illustrates the concepts and capabilities provided by this feature.

Data Path Concept

The mapping step of the Data 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

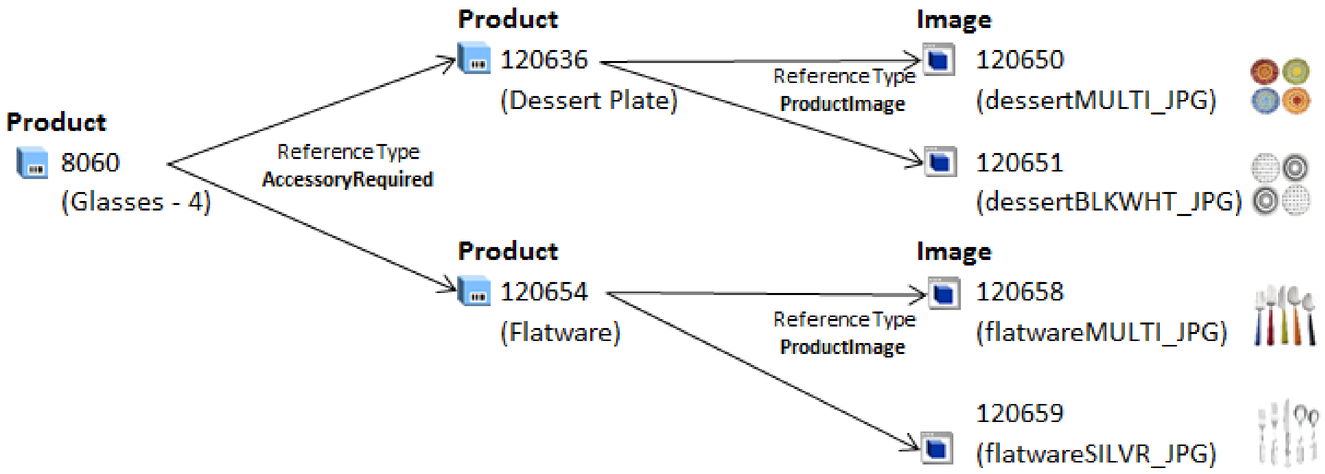
The data sources that are to be mapped below a Data Path source require special support for evaluating relative to the objects / references identified by the paths. The following standard data sources support such nesting:

- 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 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>
```

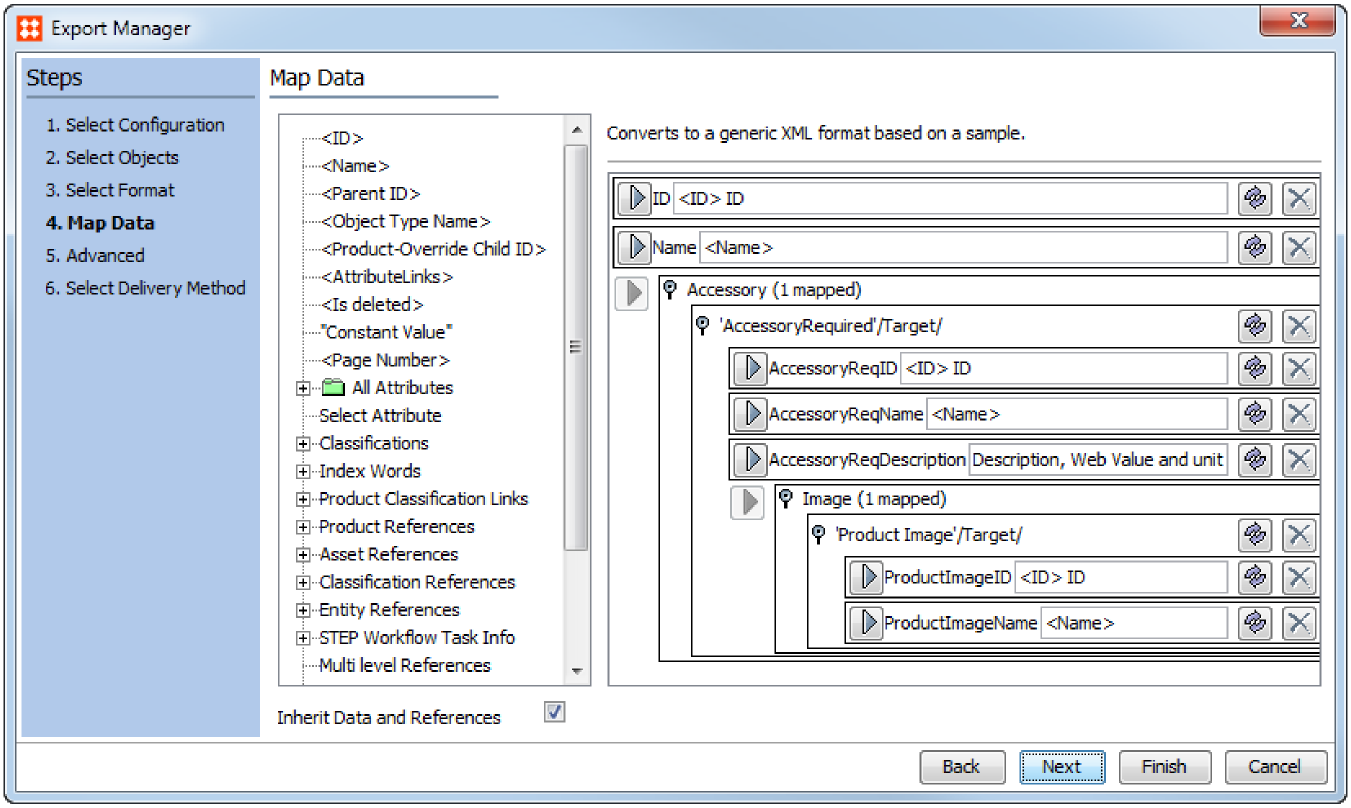
```
         <Image>
         <?MultiTarget?>
         <ID><?Target ProductImageID?></ID>
         <Name><?Target ProductImageName?></Name>
         </Image>
      </Accessory>
   </Product>
</Products>
```

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



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/

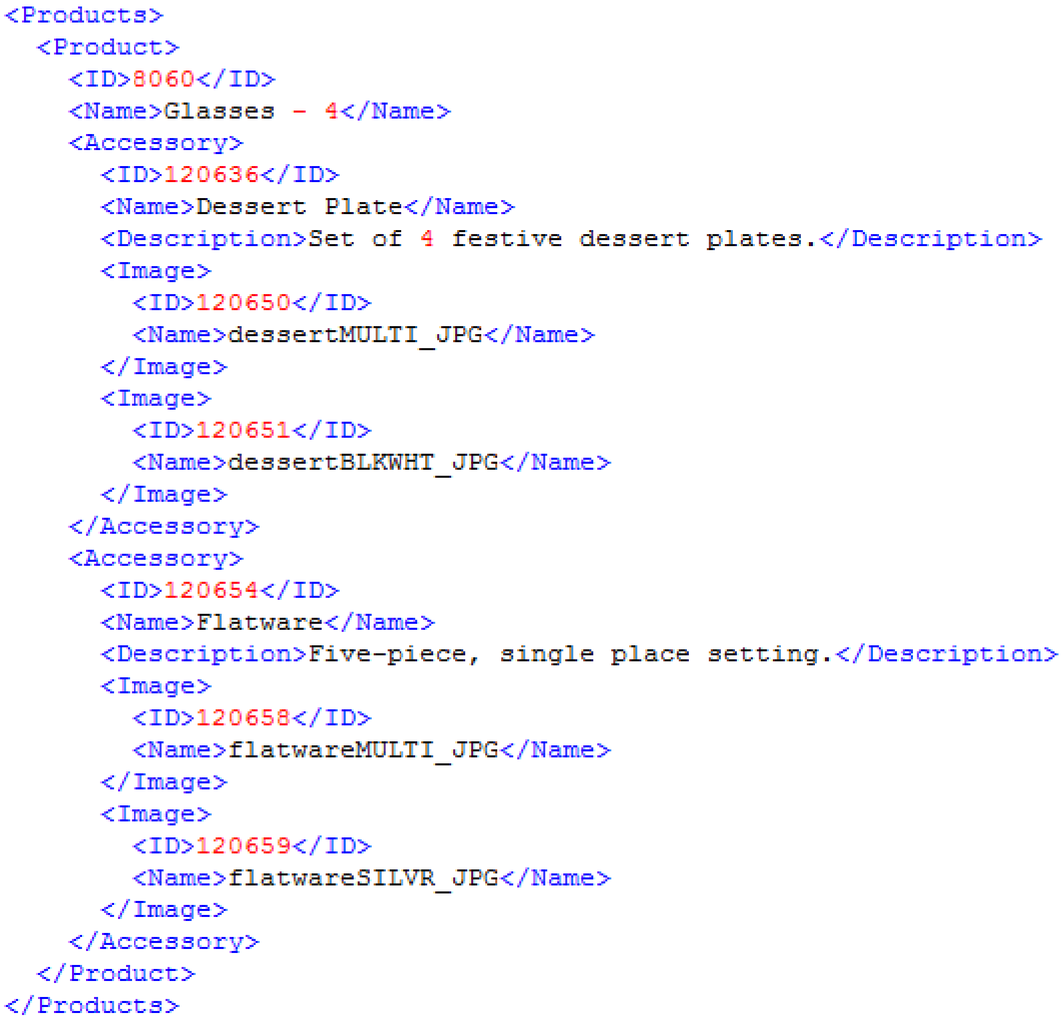
ProductImageID <ID> ID

ProductImageName <Name>

Inherit Data and References ☑

For step-by-step mapping instructions, see the Mapping Nestable Data Paths section of the STEP Export Manager User Guide / Exporting Data and Images documentation.

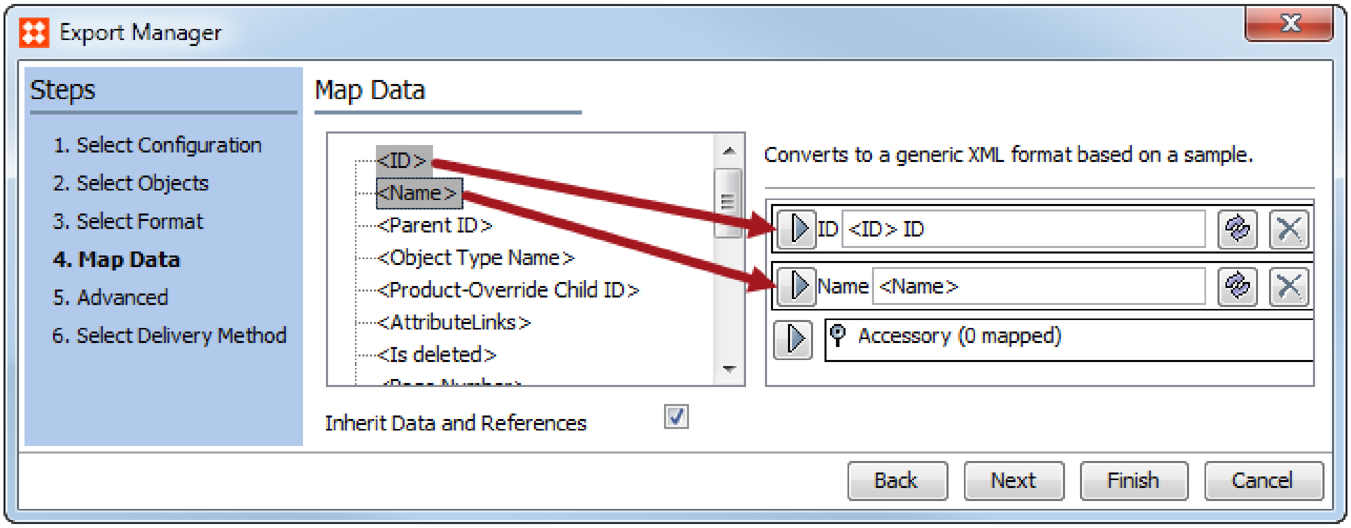
Results



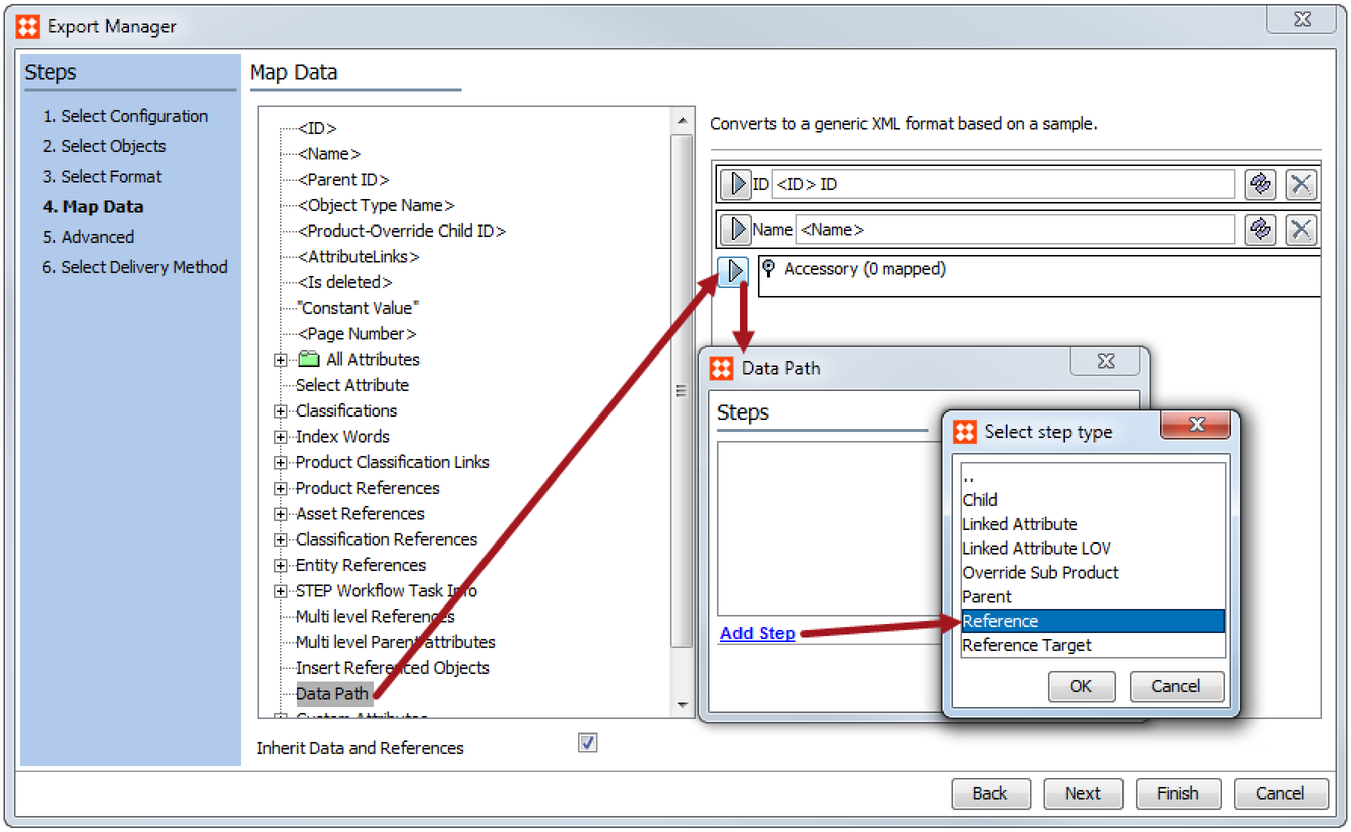
```
<Products>
  <Product>
    <ID>8060</ID>
    <Name>Glasses - 4</Name>
    <Accessory>
      <ID>120636</ID>
      <Name>Dessert Plate</Name>
      <Description>Set of 4 festive dessert plates.</Description>
      <Image>
        <ID>120650</ID>
        <Name>dessertMULTI_JPG</Name>
      </Image>
      <Image>
        <ID>120651</ID>
        <Name>dessertBLKWHT_JPG</Name>
      </Image>
    </Accessory>
    <Accessory>
      <ID>120654</ID>
      <Name>Flatware</Name>
      <Description>Five-piece, single place setting.</Description>
      <Image>
        <ID>120658</ID>
        <Name>flatwareMULTI_JPG</Name>
      </Image>
      <Image>
        <ID>120659</ID>
        <Name>flatwareSILVR_JPG</Name>
      </Image>
    </Accessory>
  </Product>
</Products>
```

Mapping Nestable Data Paths

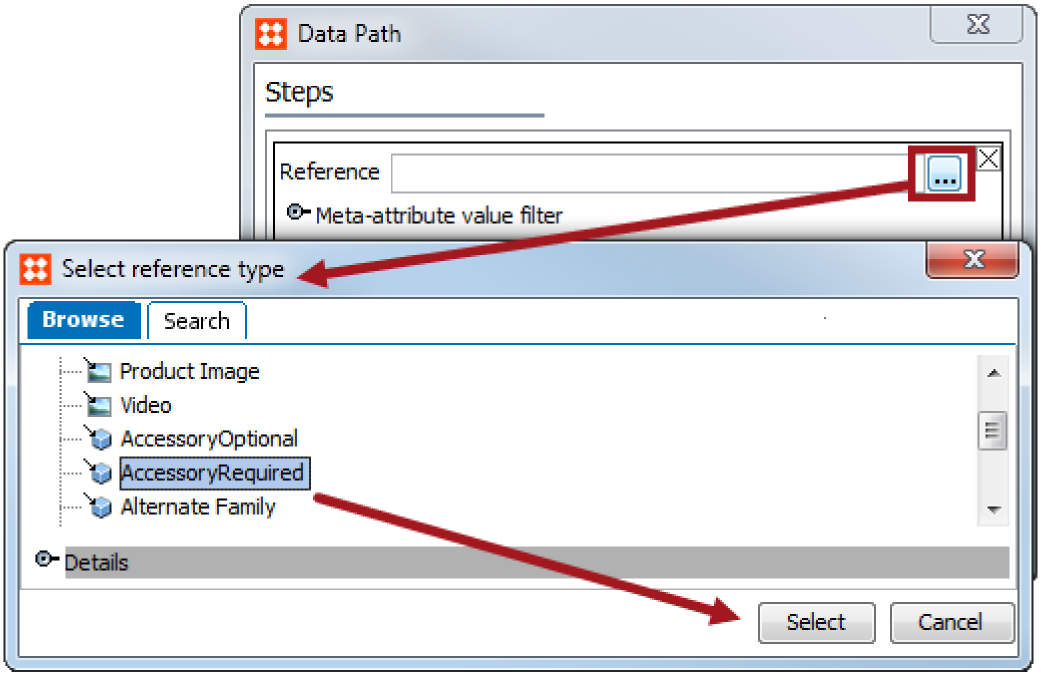
1. Map **ID** and **Name** for Level 1.



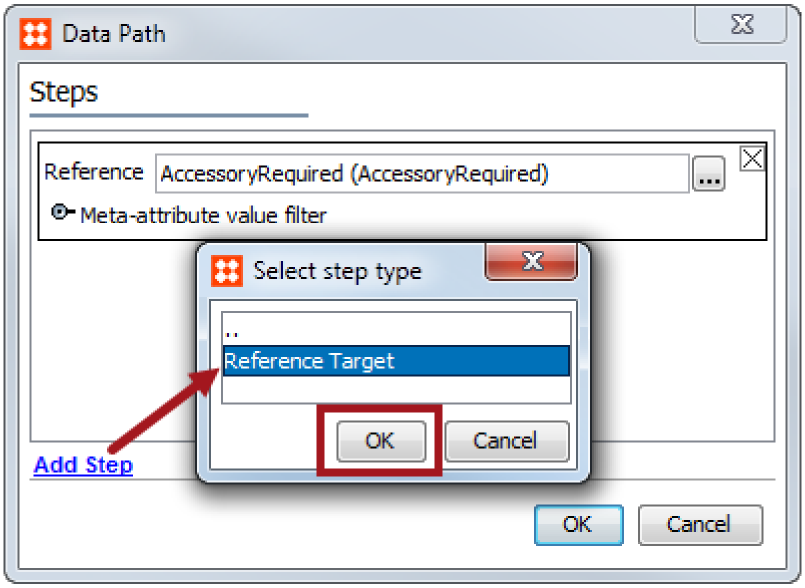
2. Map **Data Path** to Accessory, click the **Add Step** link, select the **Reference** step type, and click **OK**.



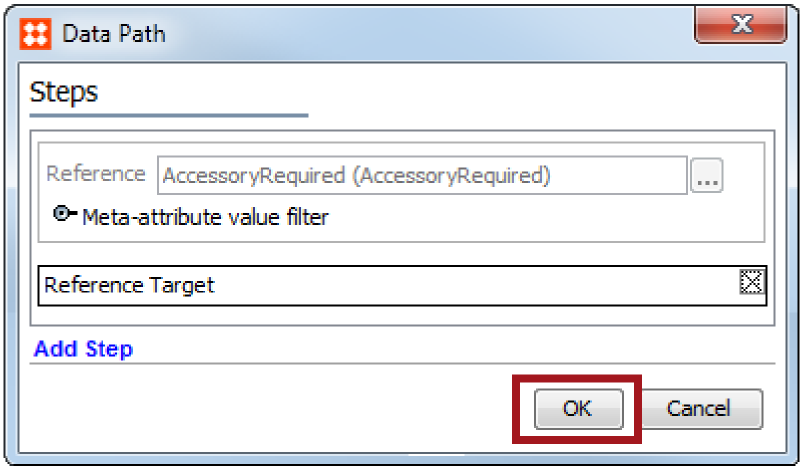
3. On the Data Path dialog, click the ellipsis button **(…)**, select AccessoryRequired reference type, and click the **Select** button.



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.



Data Path

Steps

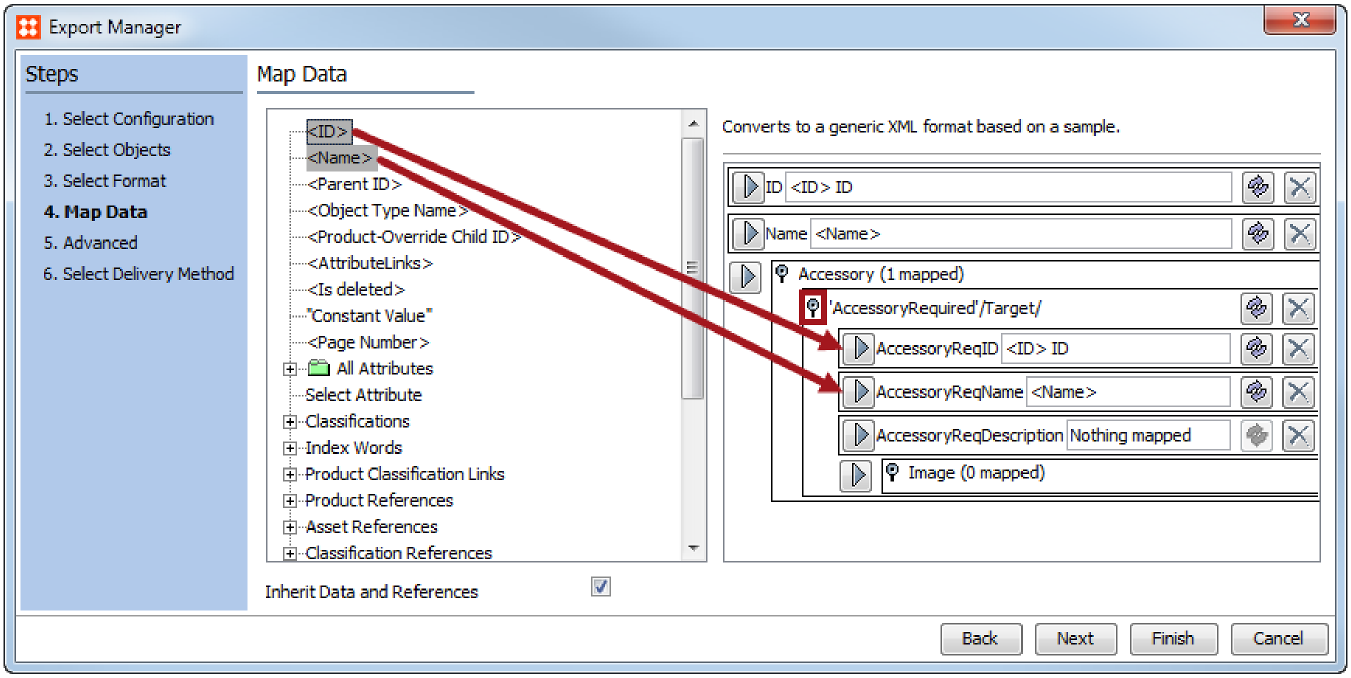
Reference AccessoryRequired (AccessoryRequired)

Meta-attribute value filter

Reference Target

Add Step

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



Export Manager

Steps

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

Map Data

Converts to a generic XML format based on a sample.

ID <ID> ID

Name <Name>

Accessory (1 mapped)

'AccessoryRequired'/Target/

AccessoryReqID <ID> ID

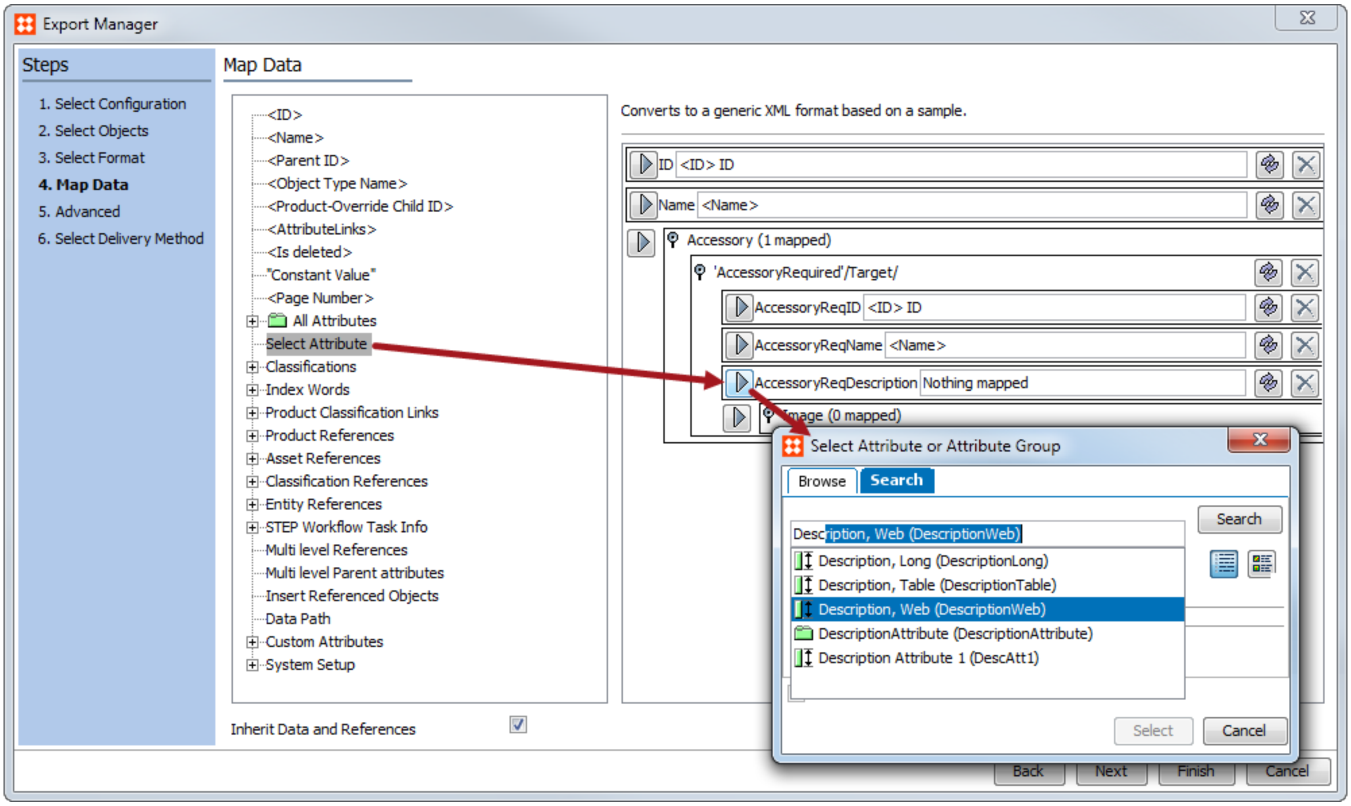
AccessoryReqName <Name>

AccessoryReqDescription Nothing mapped

Image (0 mapped)

Inherit Data and References ☑

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



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>

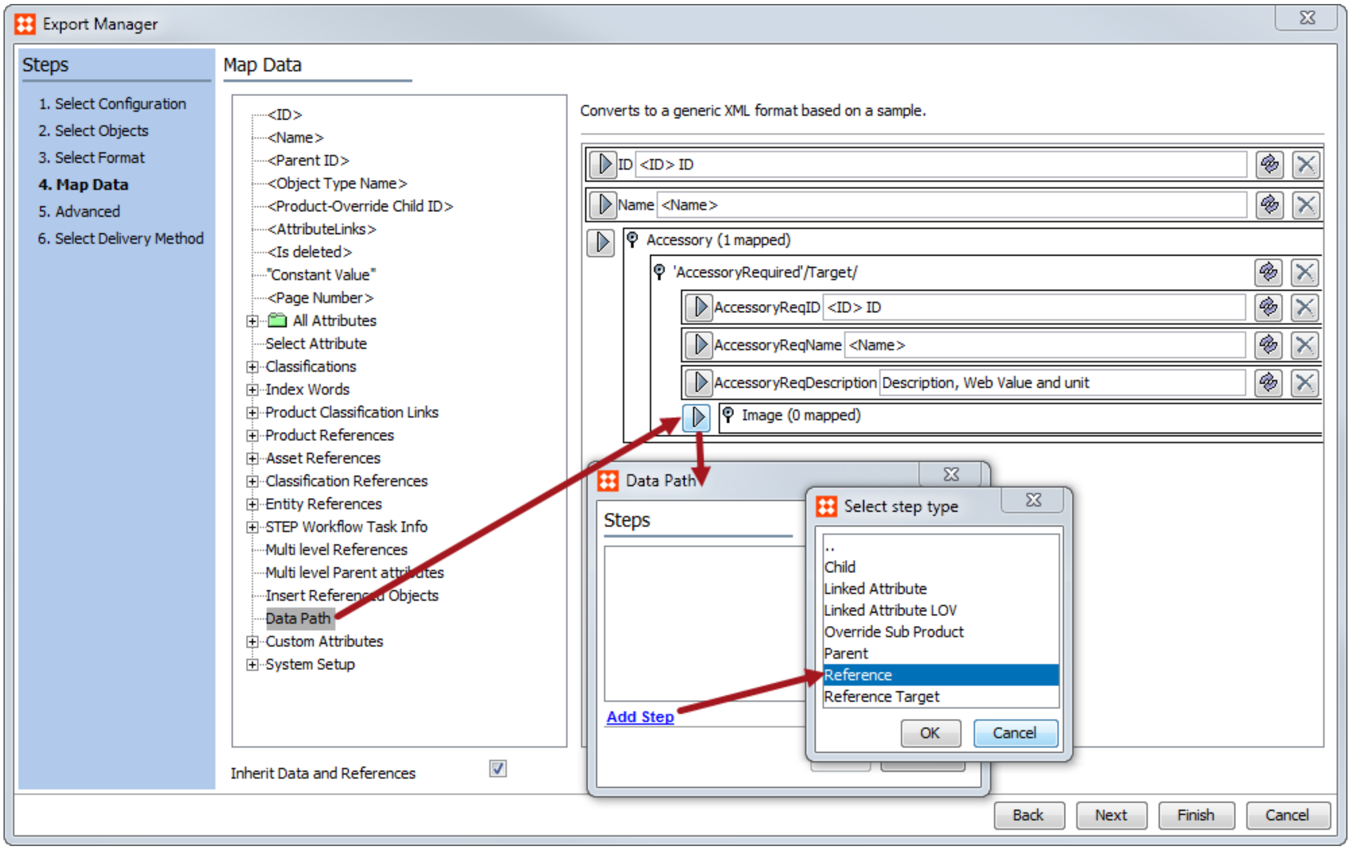
AccessoryReqDescription Nothing mapped

Image (0 mapped)

Description, Web (DescriptionWeb)

Inherit Data and References ☑

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



Converts to a generic XML format based on a sample.

ID <ID> ID

Name <Name>

Accessory (1 mapped)

'AccessoryRequired'/Target/

AccessoryReqID <ID> ID

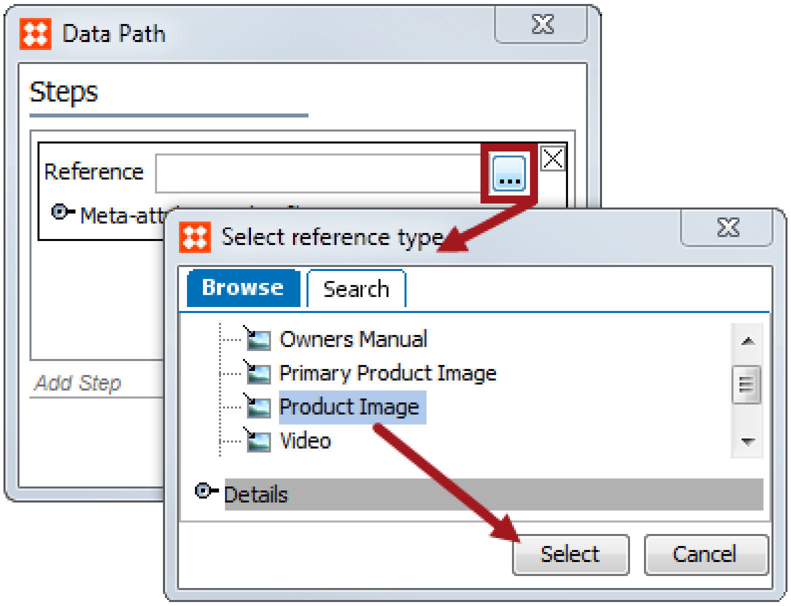
AccessoryReqName <Name>

AccessoryReqDescription Description, Web Value and unit

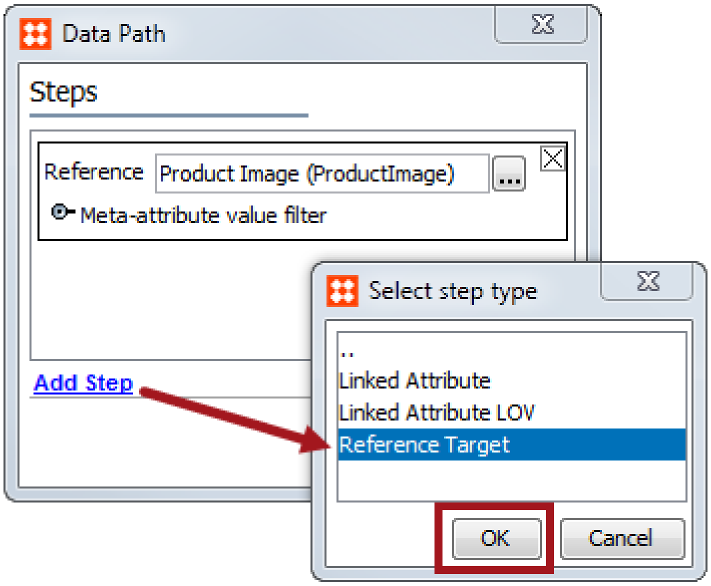
Image (0 mapped)

Inherit Data and References ☑

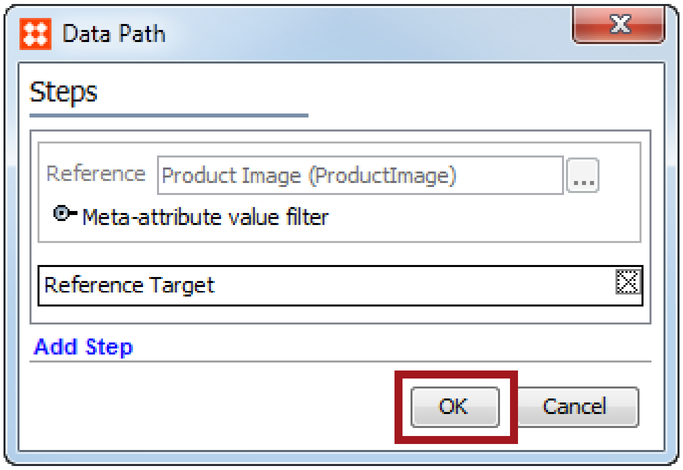
9. On the Data Path dialog, click the ellipsis button **(…)**, select ProductImage reference type, and click the **Select** button.



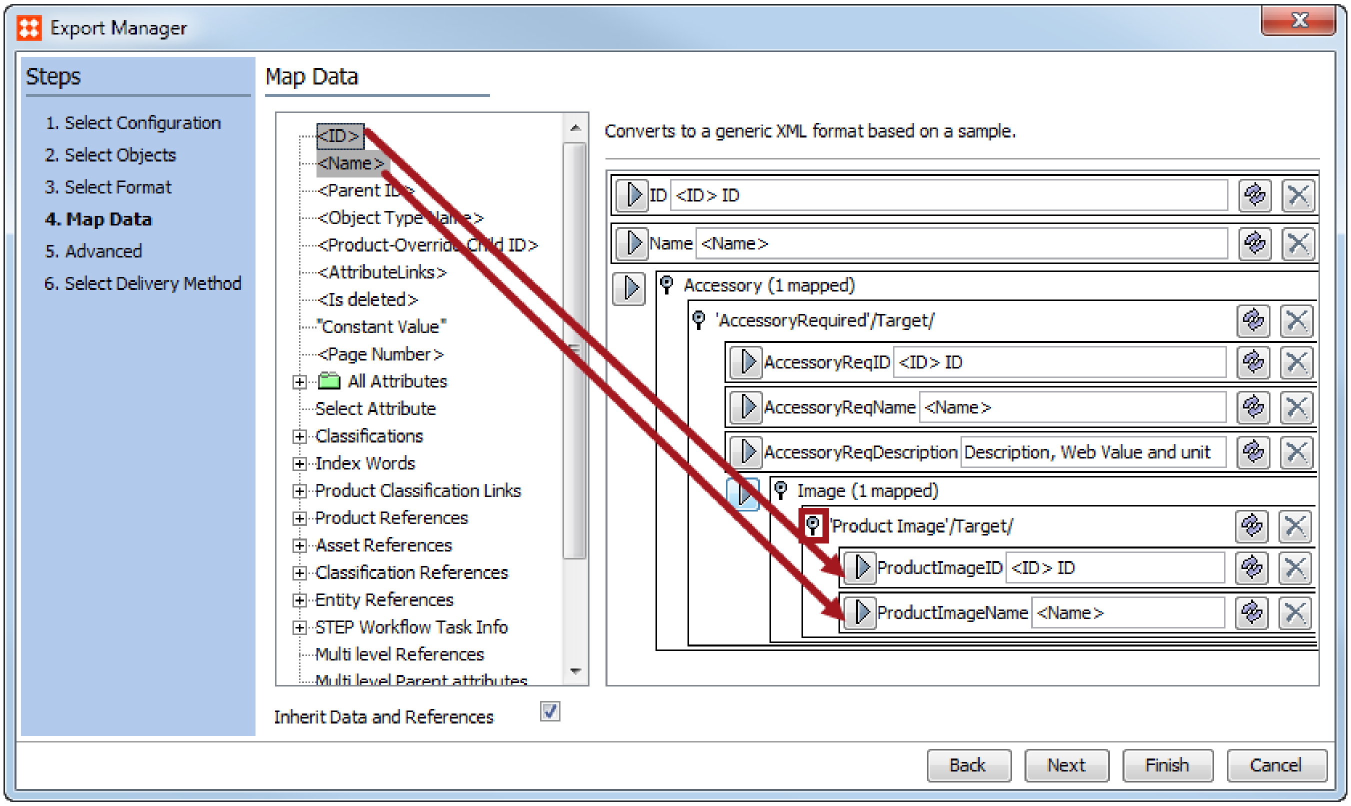
10. Click the **Add Step** link, select **Reference Target** for Image and click **OK**.



11. Click **OK** to confirm the Reference and Reference Target for the Image.



12. Open the 'ProductImage'/Target/ flipper to display the nested mapping options. Map **ID** and **Name** for Level 3 (images).



Export Manager

Steps

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

Map Data

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

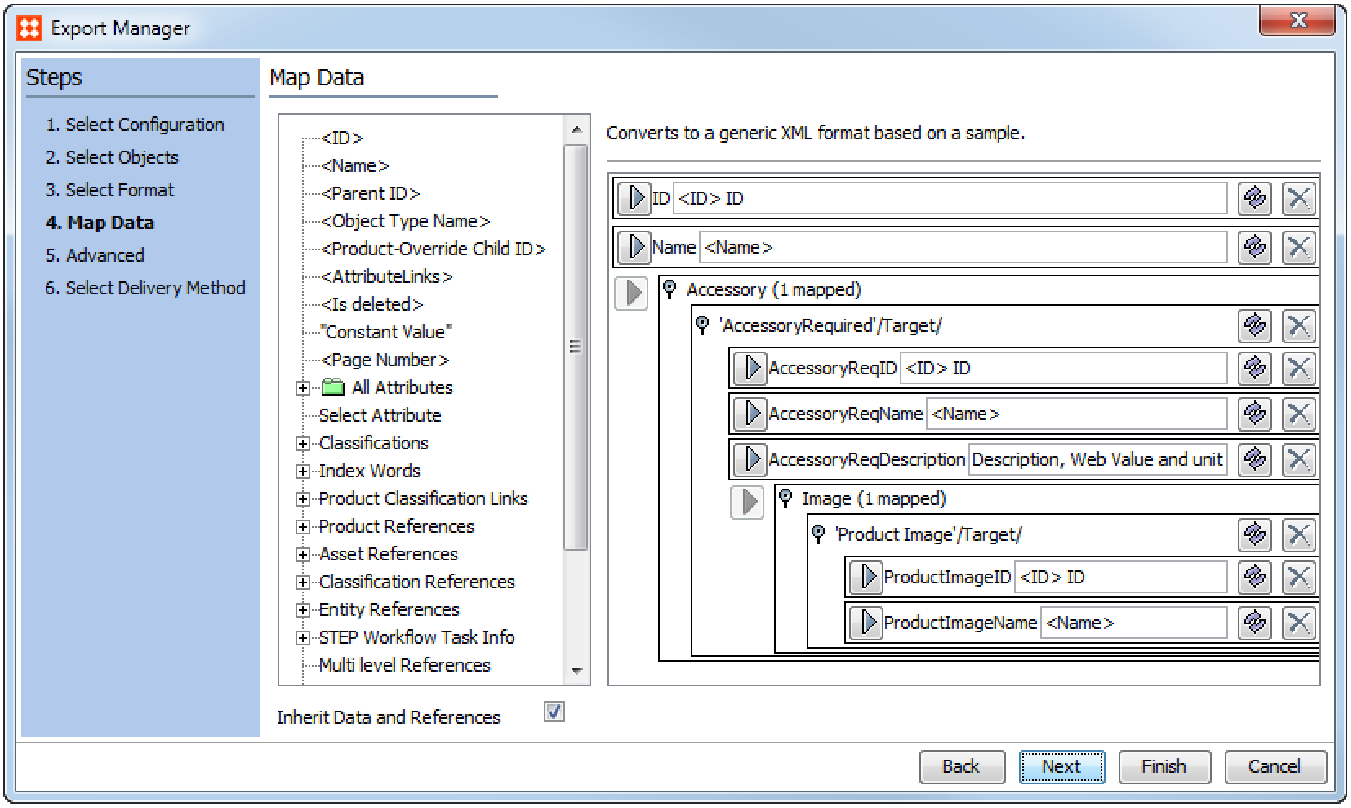
Image (1 mapped)

'Product Image'/Target/

ProductImageID	<ID> ID
ProductImageName	<Name>

Inherit Data and References ☑

All mapping is complete.



Export Manager

Steps

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

Map Data

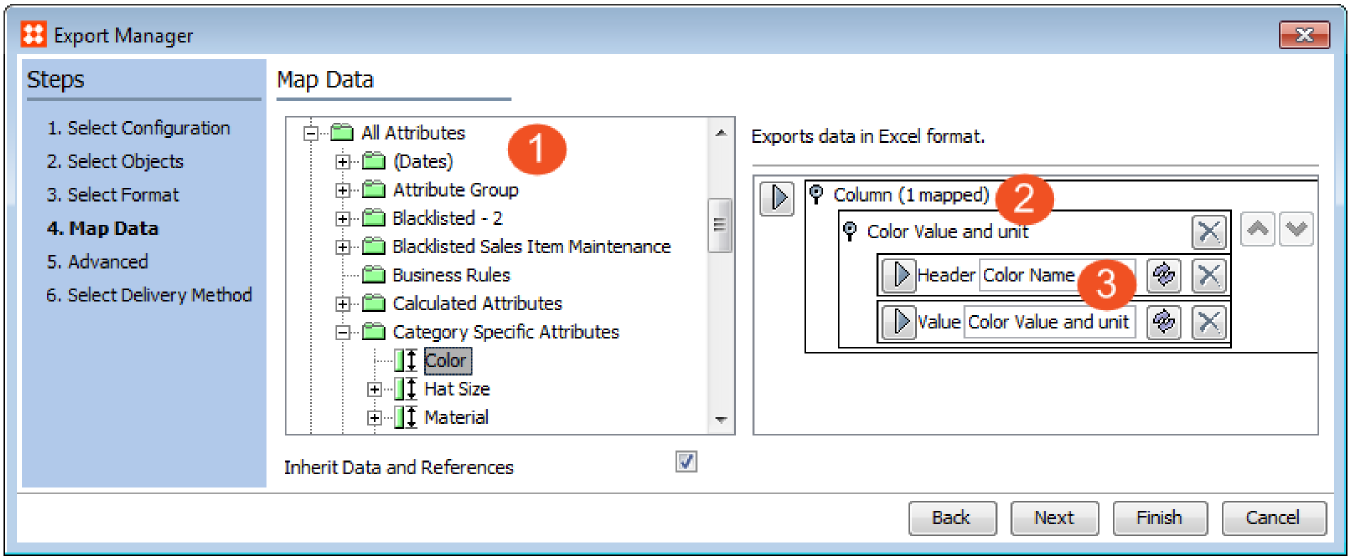
<ID>
<Name>
<Parent ID>
<Object Type Name>
<Product-Override Child ID>
<AttributeLinks>
<Is deleted>
"Constant Value"
<Page Number>
All Attributes
Select Attribute
Classifications
Index Words
Product Classification Links
Product References
Asset References
Classification References
Entity References
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 ☑

Step 4 - Map Data



Exports data in Excel format.

Inherit Data and References ☑

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 exported information. 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.
2. **Mapping Targets** are placeholders for the actual data being exported.
3. **Mapping Rules** are created by setting a relationship, also called a binding, between a data source and a mapping target. Bindings can be configured in a number of ways, including choosing a different aspect, adding a constant value, applying a transformation, or mapping additional STEP data.

For more information on how to map each data element, see the **Map Data Options** section of **Exporting Data** documentation.

For information about mapping data for multi-level Smartsheets, see the **Smartsheet Export Mapping** section of **Exporting Data** documentation.

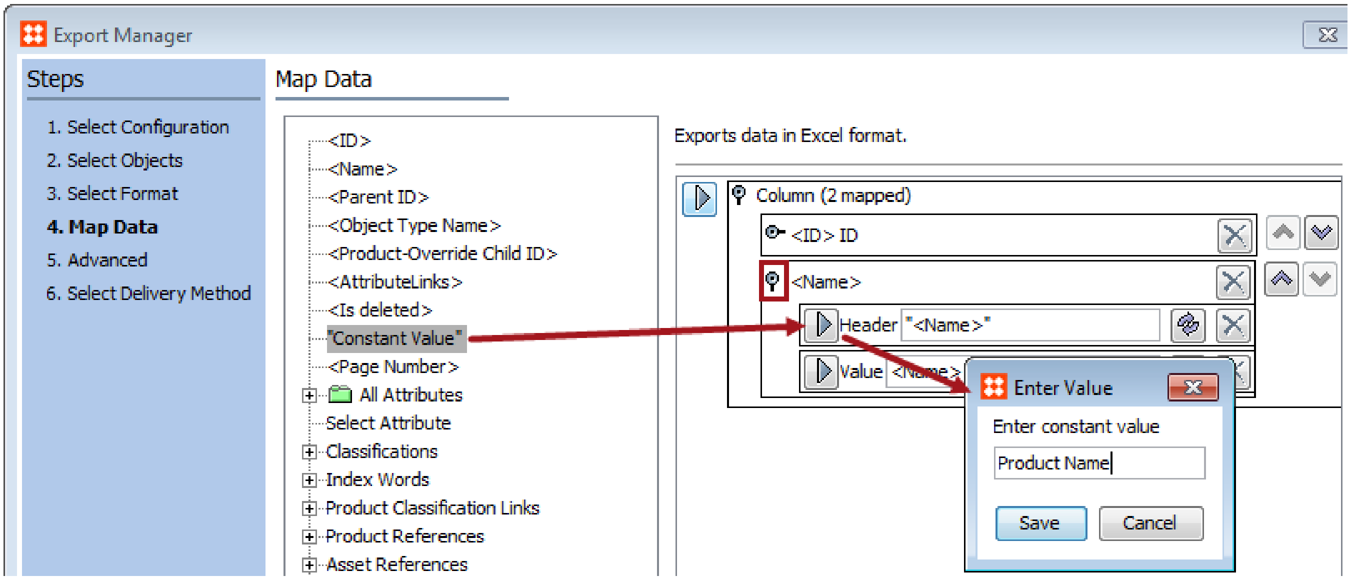
Map Data Options

The process to map data differs 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.

After mapping, most output options can be altered using transformations. For more information, see the **Transformations Overview** section of the **Exporting Data** documentation.

Mapping a Constant Value

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.



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 **Map Data Aspect and Constant Example** section of **Exporting Data** documentation.

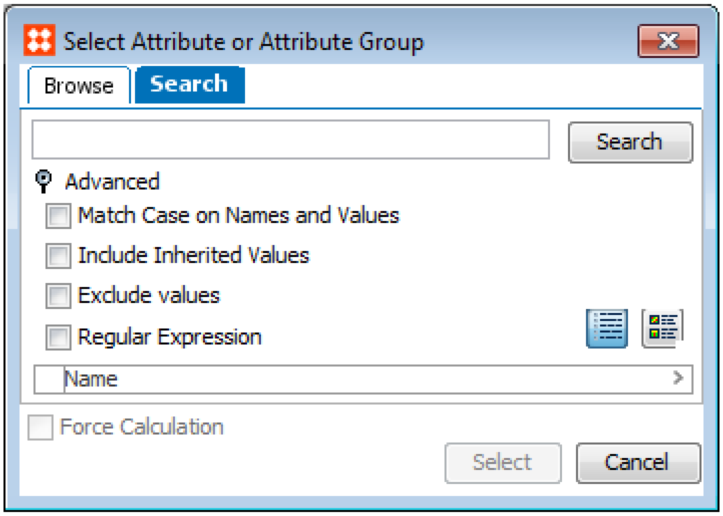
Mapping Attributes

Attributes can be mapped using the All Attributes hierarchy option or the Select Attribute option. The Custom Attribute selection is system dependent and is only available special scripts exist that operate on certain attributes.

The **All Attributes** option 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).

The **Select Attribute** option allows you to browse or search for an attribute.

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.



Select Attribute or Attribute Group

Browse | Search

Search

Advanced

□ Match Case on Names and Values

□ Include Inherited Values

□ Exclude values

□ Regular Expression

Name

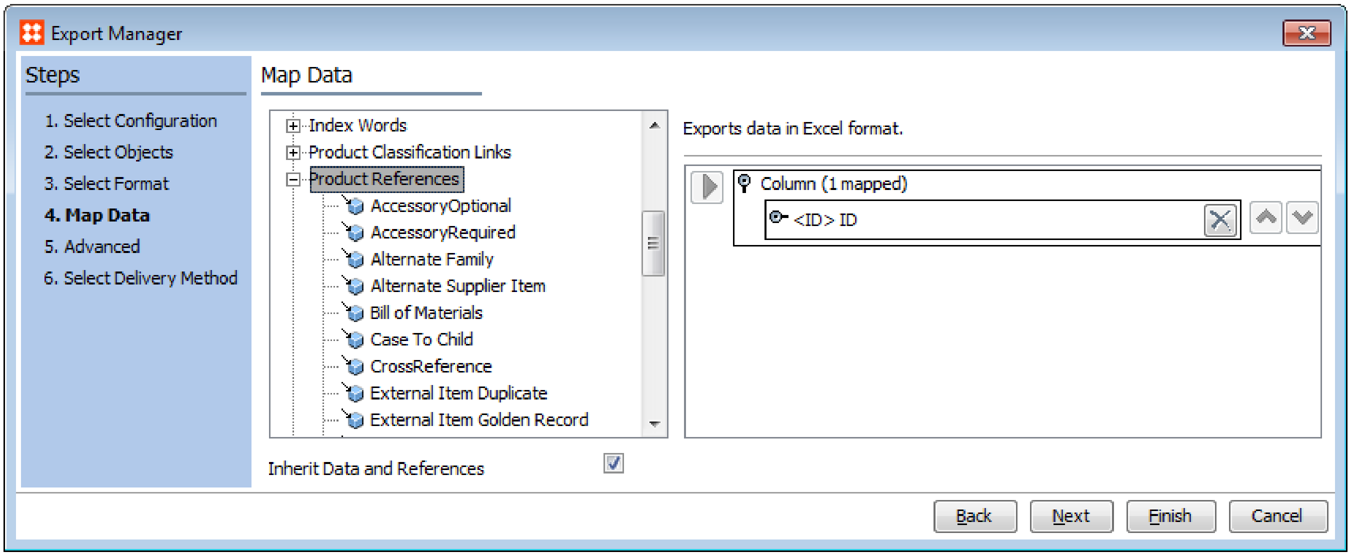
□ Force Calculation

Select | Cancel

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 steps 1 to 3 to add all other required attributes.

Product-To-Product References

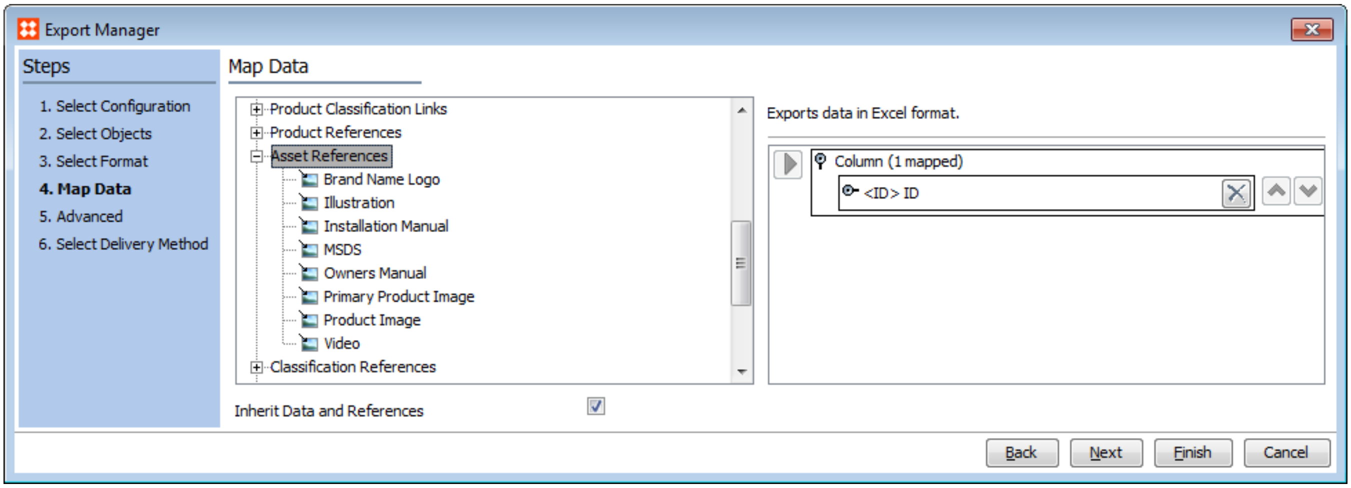
1. In the left panel, expand **Product References**, and then select the preferred reference type from the available list.



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.

Product-To-Asset References

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

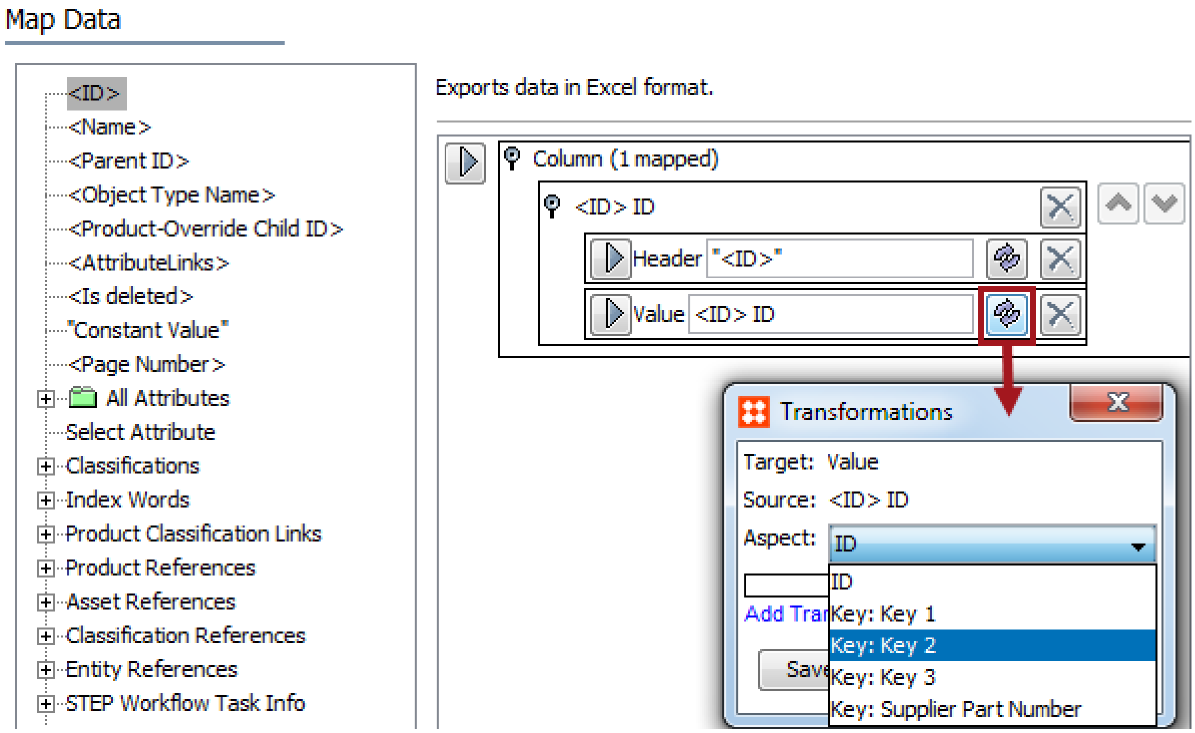


2. Click the right arrow button in the right panel to add the selected asset reference type as a column.
3. Add additional asset reference types, if needed. Each is extracted into a separate field in the output.

Mapping Keys

When unique key are 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 Export Manager. For more on unique keys, see the **Unique Keys** section of the **System Setup / Super User Guide** documentation.

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

- <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

Exports data in Excel format.

Column (1 mapped)

<ID> ID

Header "<ID>"

Value <ID> ID

Transformations

Target: Value

Source: <ID> ID

Aspect: ID

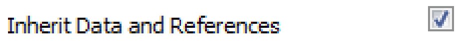
Changing the Order of the Fields

Fields in the export file appear in the same order as they are displayed in the Export Manager wizard. The top field will be first in the output.

Use the up ▲ and down ▼ arrows to change the order the fields.

Inherit Data and References

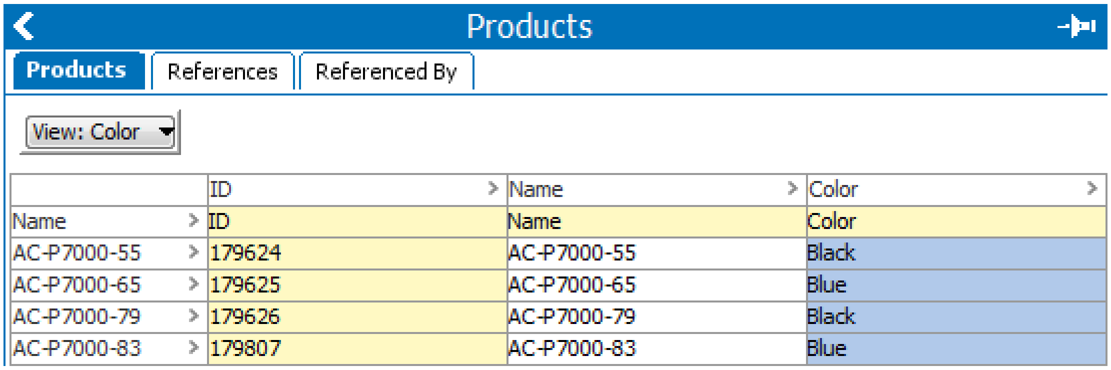
The **Inherit Data and References** option is a global setting that applies to all attributes selected in the Mapping Data window. When checked, the output includes values that are inherited from objects at a higher level in the hierarchy.



Inherit Data and References ☑

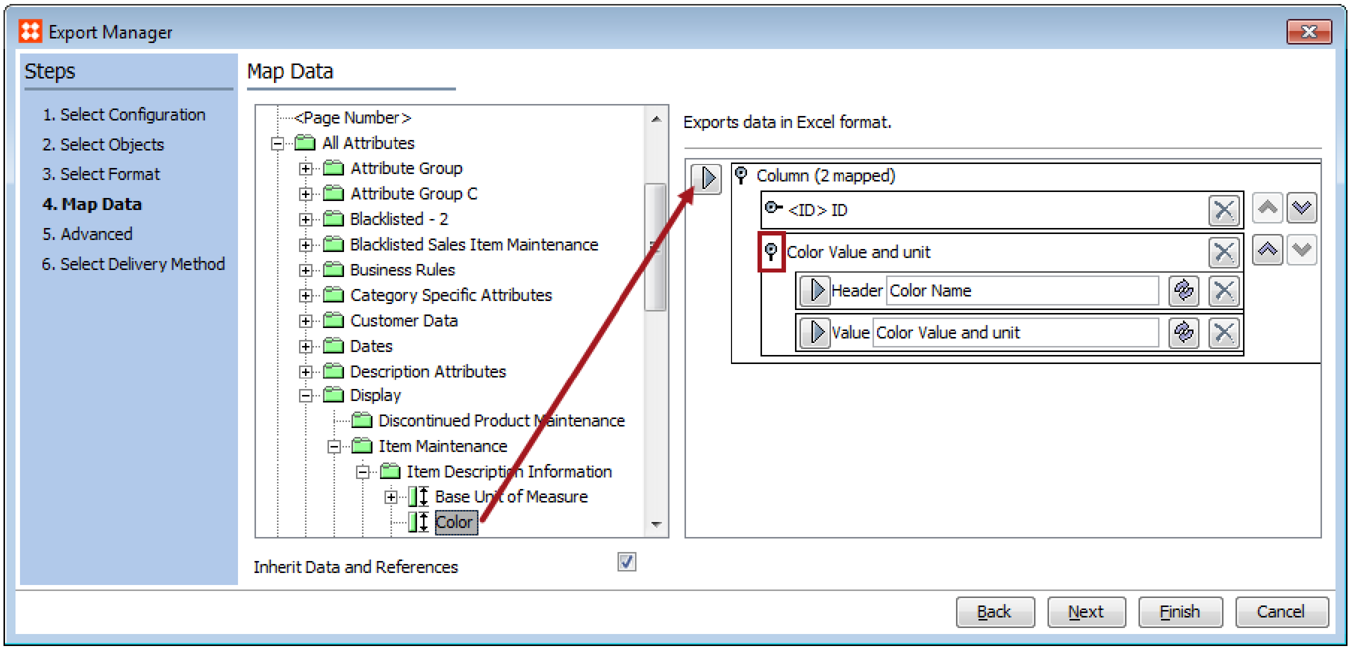
Map Data Aspect and Constant Example

This example shows a variety of mappings and exports using a single attribute, named 'Color'. The following products have values for the Color attribute and are displayed using the multi object editor:



	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.

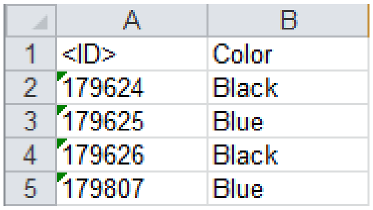


Exports data in Excel format.

Column (2 mapped)
<ID> ID
Color Value and unit
Header Color Name
Value Color Value and unit

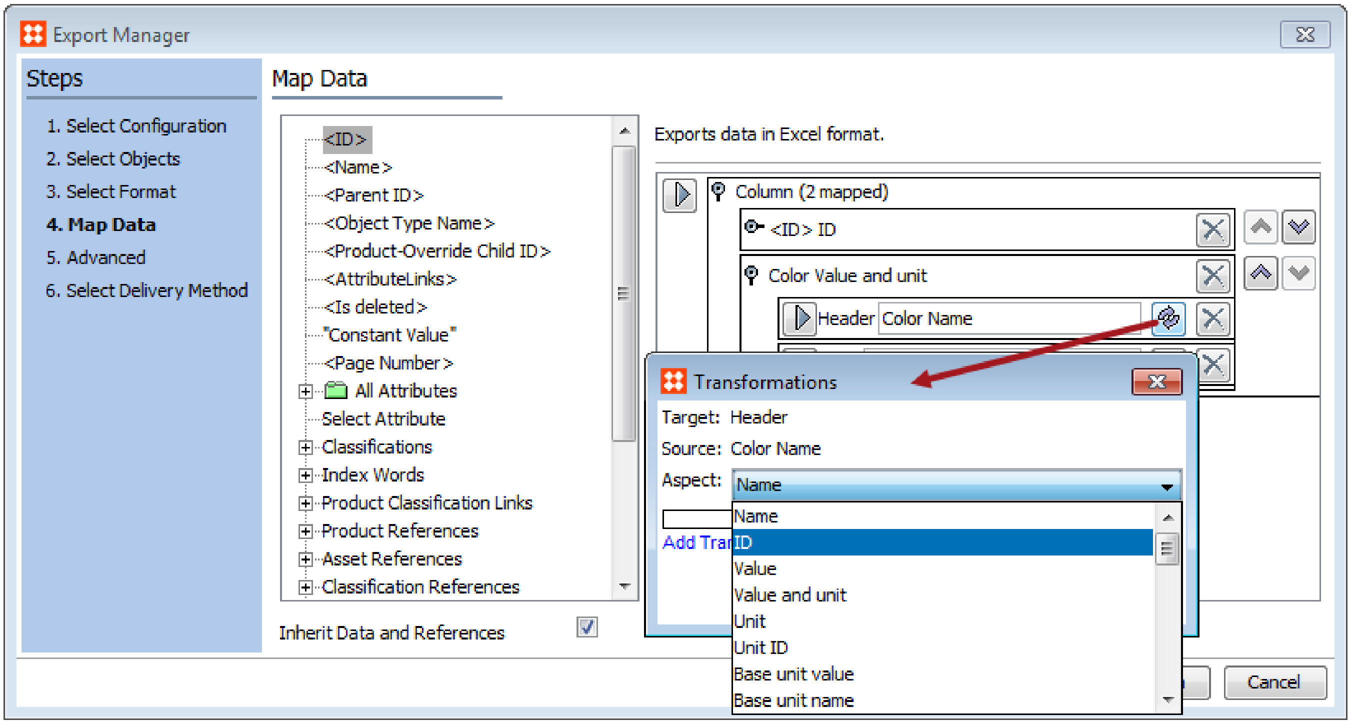
Inherit Data and References ☑

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.



Export Manager

Steps

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

Map Data

Exports data in Excel format.

Column (2 mapped)

<ID> ID

Color Value and unit

Header Color Name

Transformations

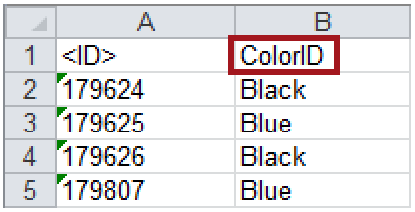
Target: Header

Source: Color Name

Aspect: Name

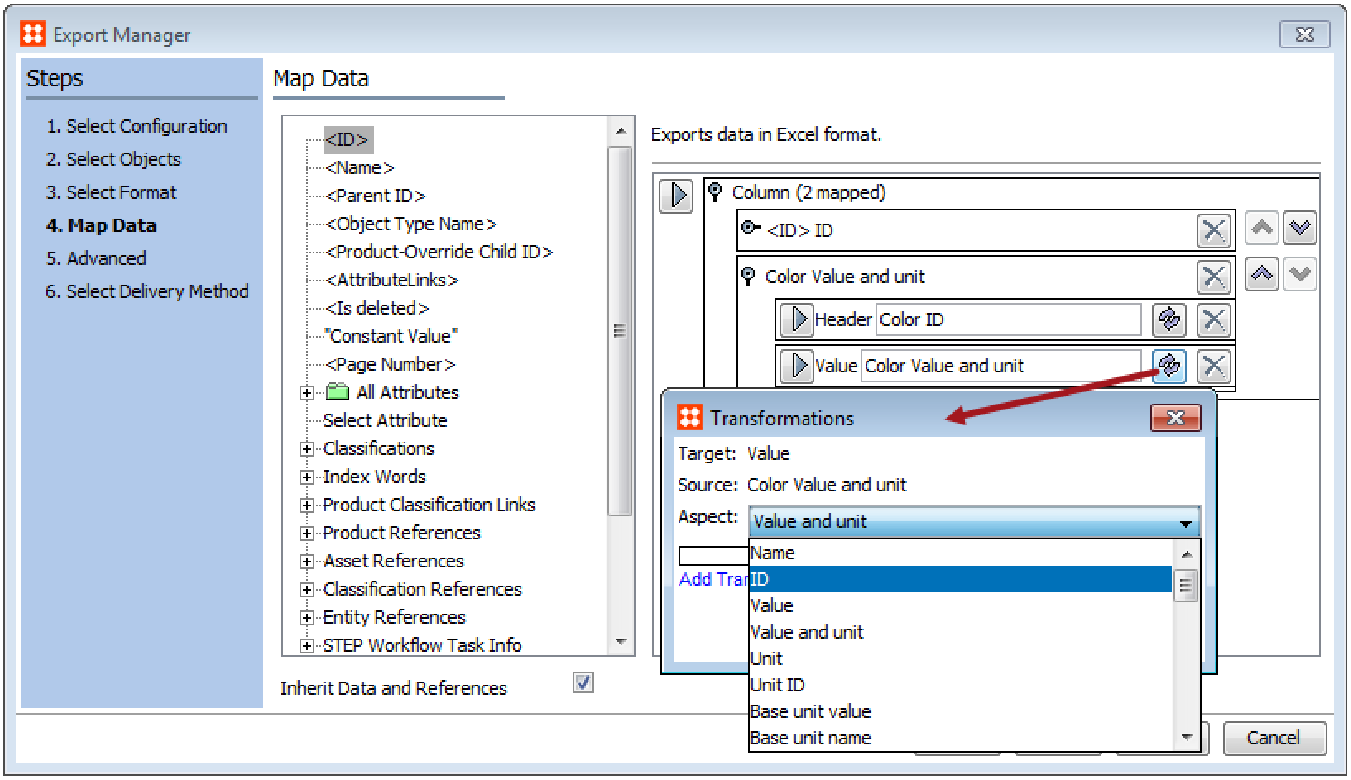
Inherit Data and References ☑

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.



Exports data in Excel format.

Column (2 mapped)

- <ID> ID
- Color Value and unit
  - Header Color ID
  - Value Color Value and unit

Transformations

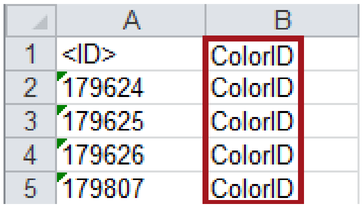
Target: Value

Source: Color Value and unit

Aspect: Value and unit

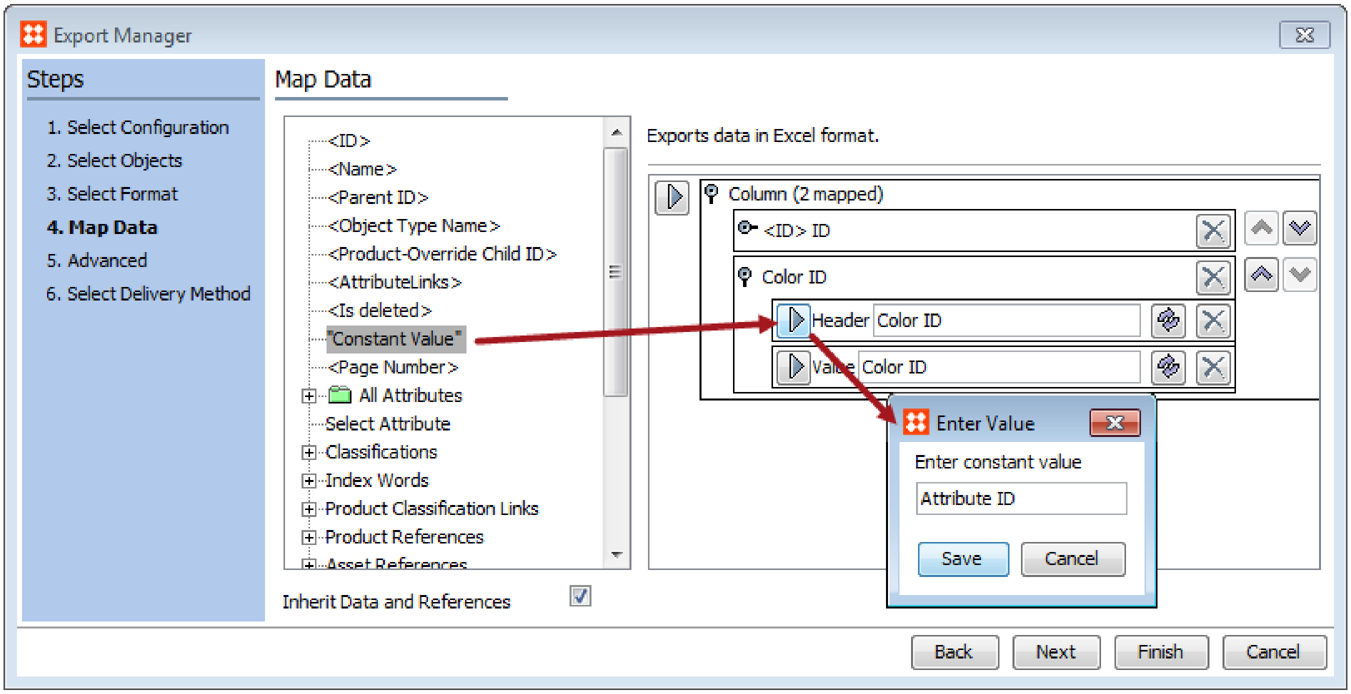
Inherit Data and References ☑

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.



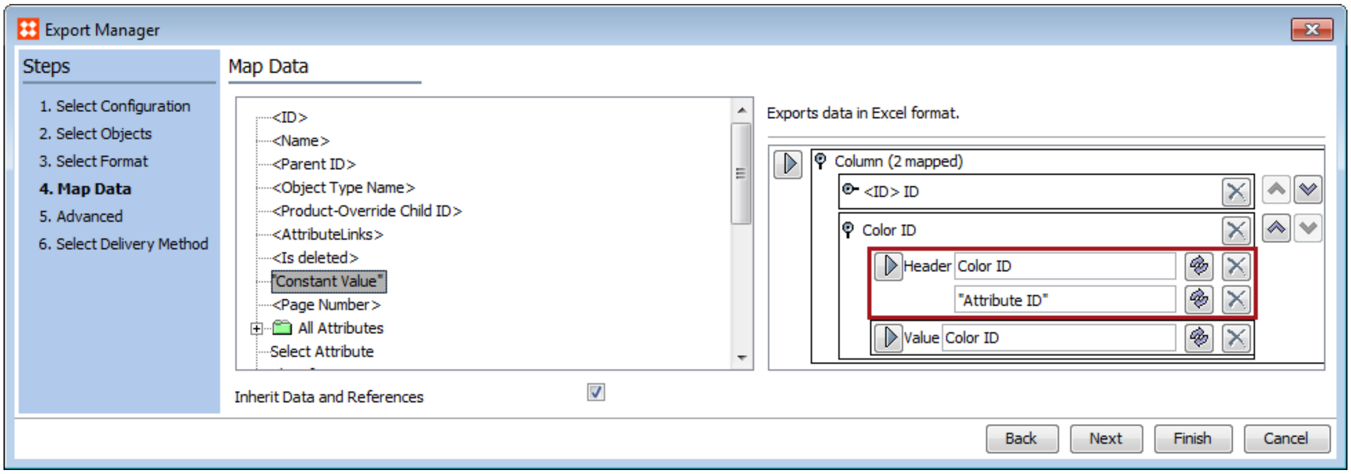
Header Color ID

Value Color ID

Enter constant value
Attribute ID

Inherit Data and References ☑

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



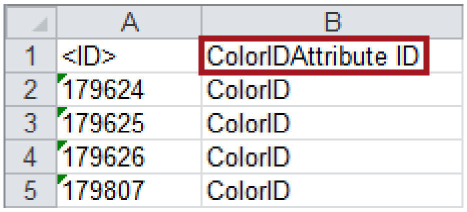
Header Color ID

"Attribute ID"

Value Color ID

Inherit Data and References ☑

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'.

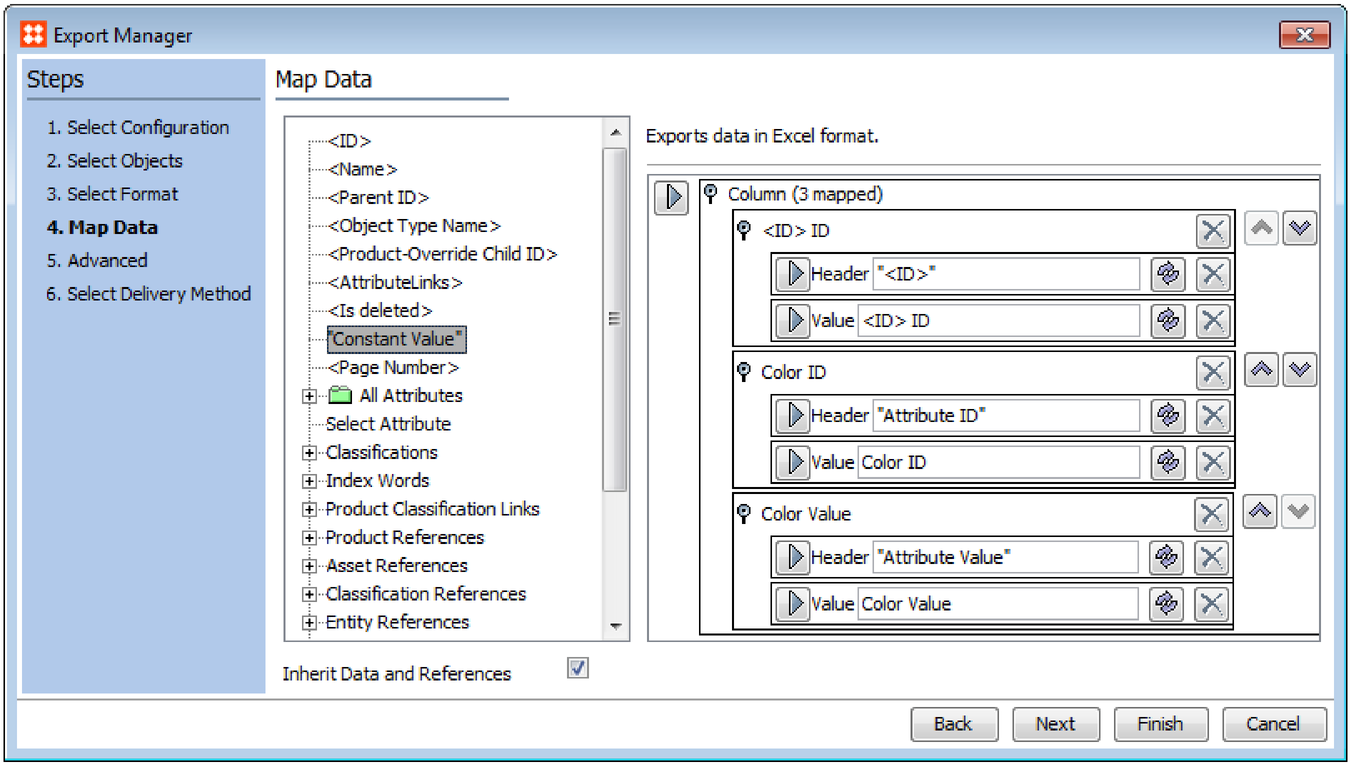


	A	B
1	<ID>	ColorIDAttribute ID
2	179624	ColorID
3	179625	ColorID
4	179626	ColorID
5	179807	ColorID

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.



Export Manager

Steps

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

Map Data

- <ID>
- <Name>
- <Parent ID>
- <Object Type Name>
- <Product-Override Child ID>
- <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

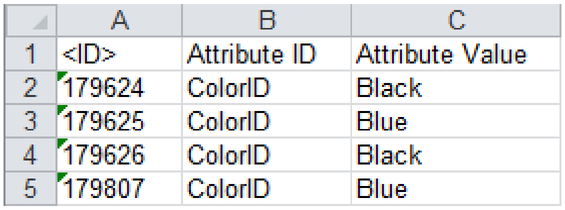
Inherit Data and References ☑

Exports data in Excel format.

Column (3 mapped)

- <ID> ID
  - Header "<ID>"
  - Value <ID> ID
- Color ID
  - Header "Attribute ID"
  - Value Color ID
- Color Value
  - Header "Attribute Value"
  - Value Color Value

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

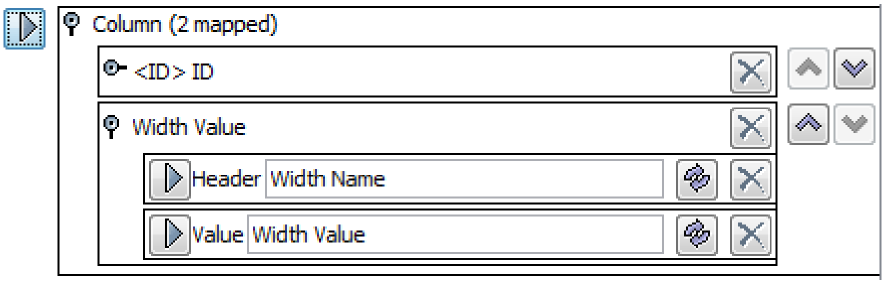
For more information on mapping each data element, see the **Map Data Options** section of **Exporting Data** documentation.

Merge Attributes With Mapping

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.



Column (2 mapped)

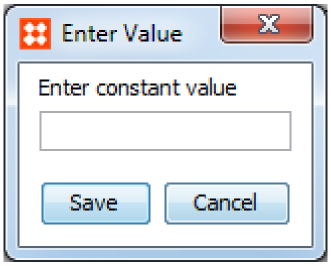
<ID> ID

Width Value

Header Width Name

Value Width Value

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

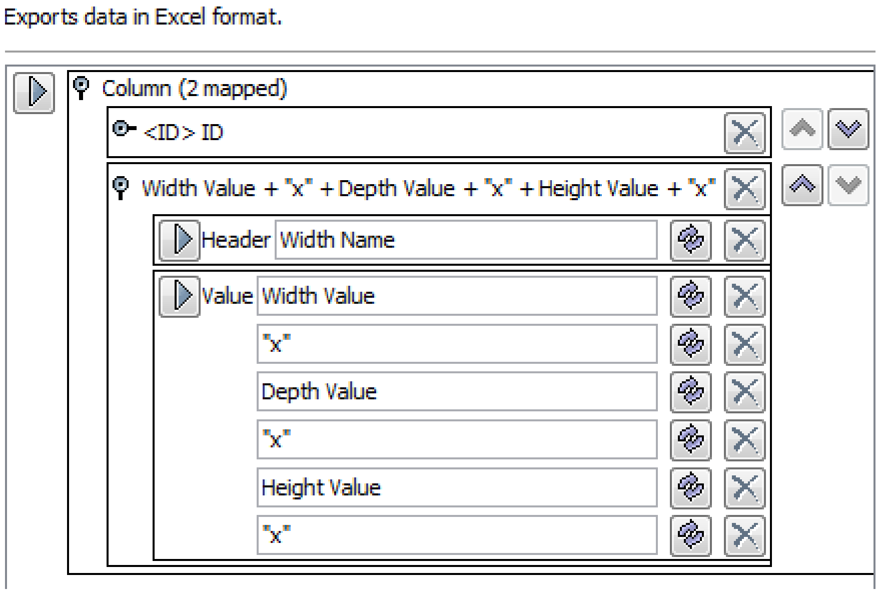


Enter Value

Enter constant value

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.

Column (2 mapped)

<ID> ID

Width Value + "x" + Depth Value + "x" + Height Value + "x"

Header Width Name

Value Width Value

"x"

Depth Value

"x"

Height Value

"x"

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.

Use the delete button (X) to remove any data that is not required in the output.

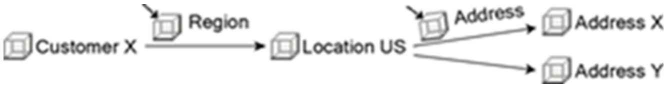
Exporting Data from Referenced Objects

From the Map Data screen, you can map data from referenced / ancestor objects in a way where the data is presented in a column for each object being exported. You can also map so that one or more additional rows are generated for referenced objects.

Mapping Multi Level References

Multi level References allows you to export values from the levels available with the reference links as a separate column for each object being exported.

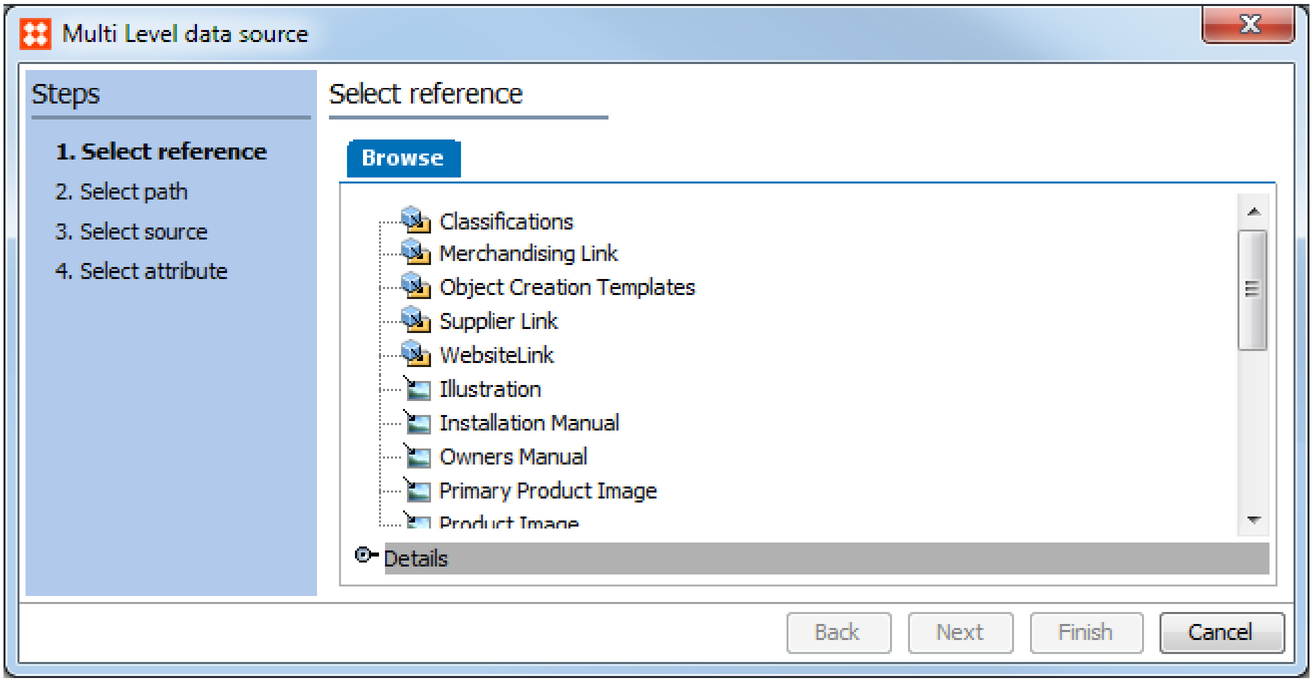
For example, a customer hierarchy is created in STEP using entity nodes. Each customer has a reference of the type Region to a Location. Each Location has a reference of type Address to specific addresses. When the customer hierarchy is exported, it is possible to follow the references and export data from the address objects.



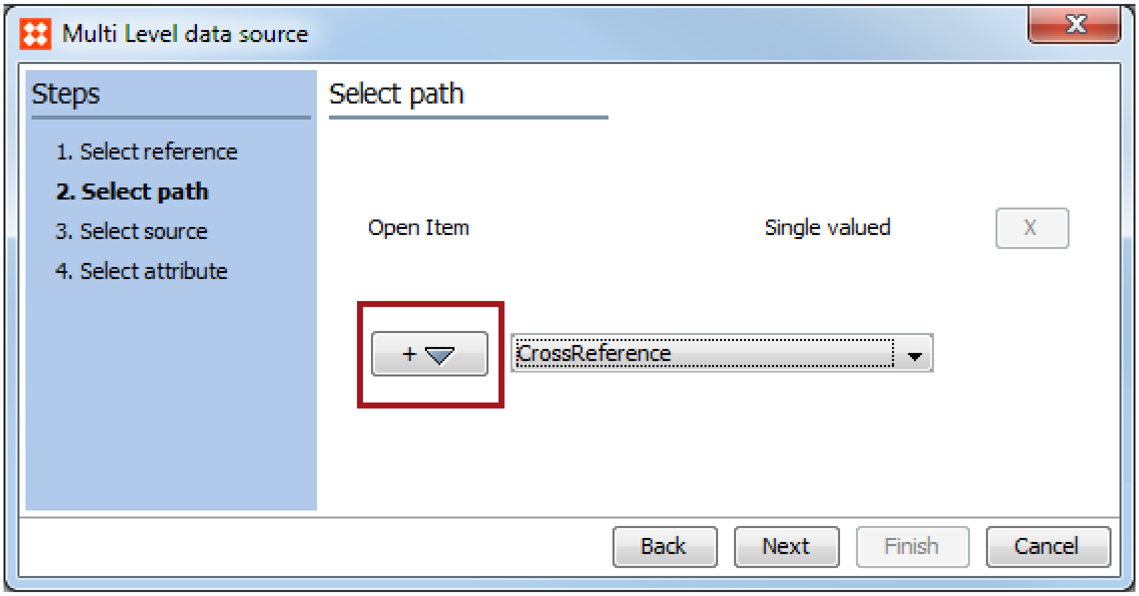
The following can be exported

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

1. In the left panel of the Export Manager wizard, select **Multilevel References**.
2. In the right panel, click the right arrow icon to display the **Multilevel data source** wizard.
3. In **Select reference**, highlight the relevant source reference type and click **Next**.



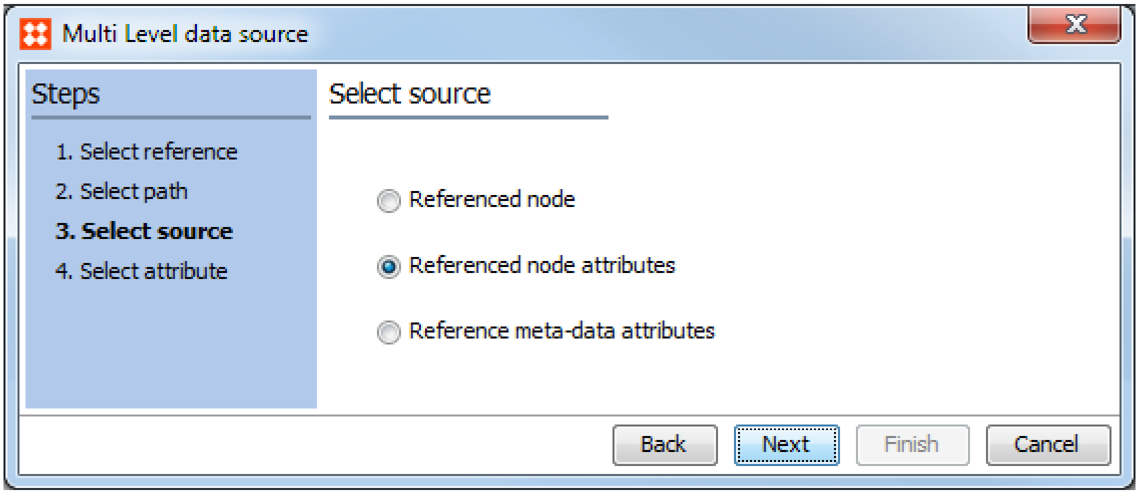
4. In **Select path**, the first legal reference type to follow is displayed. Keep the default, or select a different reference type. To add an additional reference type, click the plus / arrow button.



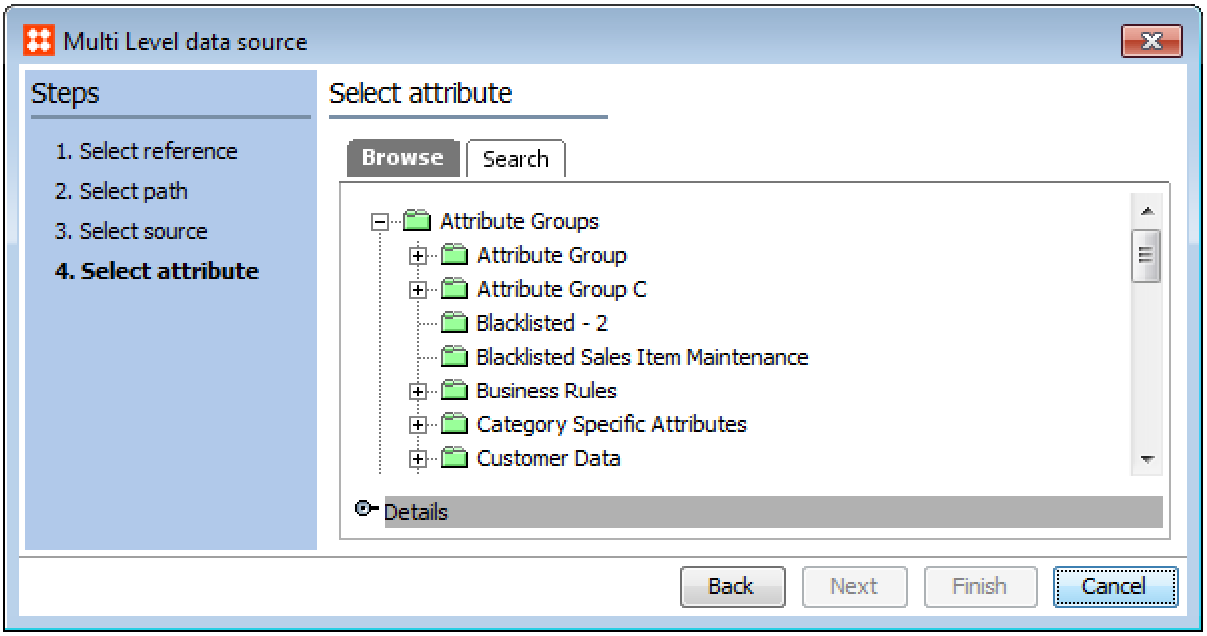
5. To export data from all referenced objects of the selected type, set the value to **Multivalued**. To export data from only a single referenced object, set the value to the desired Item number. For example, Item [1] only exports data from the first referenced object. Click **Next**.

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**.



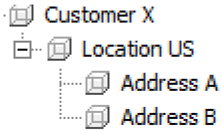
7. If needed, in **Select Attribute**, use **Browse** or **Search** to locate the attribute that holds the values to be exported and click **Finish**.



Mapping Multi Level Parent Attributes

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

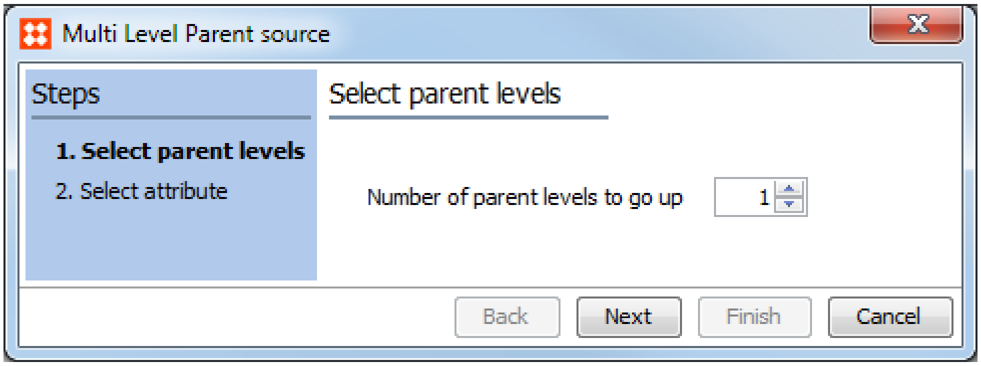
For example, a customer hierarchy is created in STEP using entity nodes. Addresses are grouped into locations.



- Customer X
- Location US
  - Address A
  - Address B

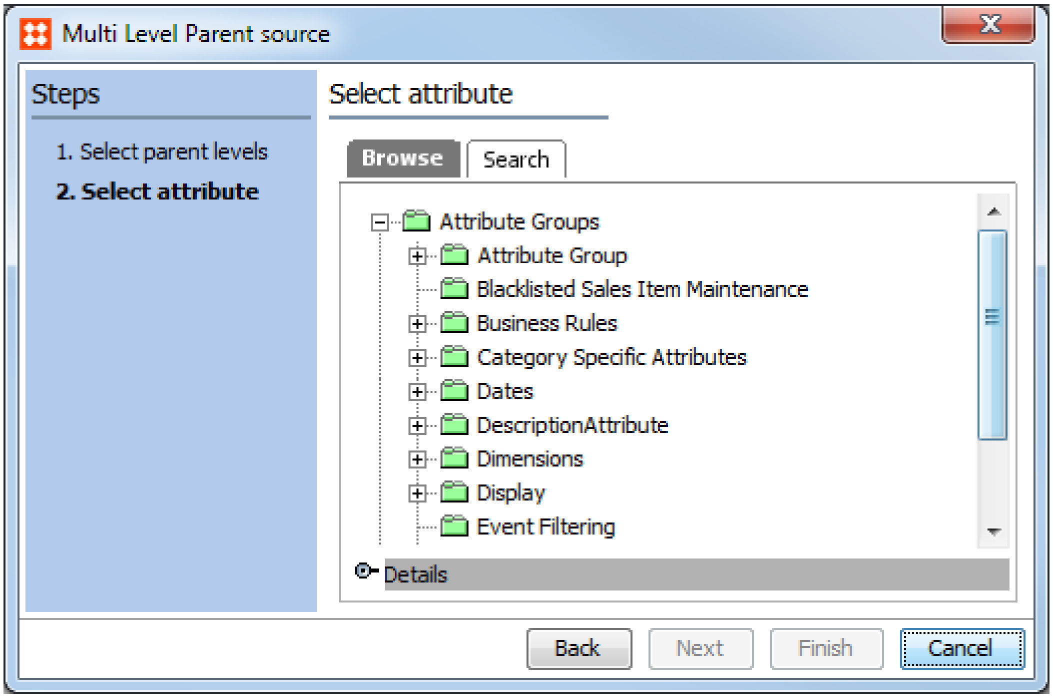
When this customer hierarchy is exported, you can export attribute values on parent nodes. This means that the row that holds Address A and Address B could also include values that only exist on the Location US or Customer X entity objects.

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**.



Number of parent levels to go up: 1

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



Multi Level Parent source

Steps

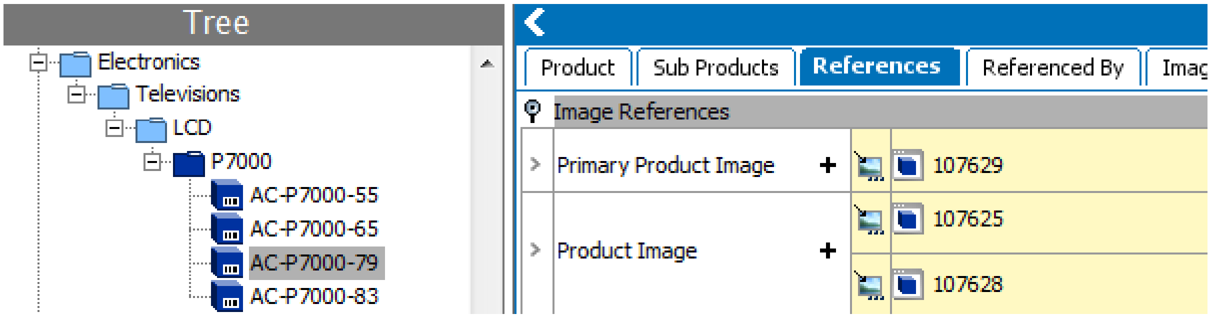
1. Select parent levels
2. Select attribute

Select attribute

Insert Referenced Objects

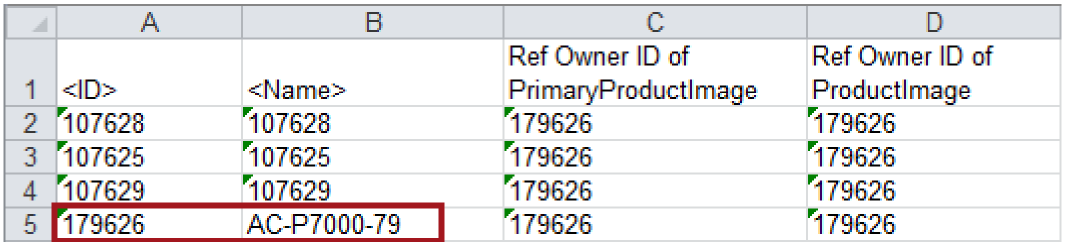
Using this mapping causes additional rows to be exported. All objects referenced from the selected reference type are exported before the object that is selected to be exported. The column added by this mapping contains the ID of the owner of the reference, in order to see that the objects belong together.

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



Primary Product Image	107629
Product Image	107625 107628

Using the Insert Referenced Objects option, the export shows the product ID once under the <ID> column and again for each of the two Ref Owner ID columns.



	A	B	C	D
1	<ID>	<Name>	Ref Owner ID of PrimaryProductImage	Ref Owner ID of ProductImage
2	107628	107628	179626	179626
3	107625	107625	179626	179626
4	107629	107629	179626	179626
5	179626	AC-P7000-79	179626	179626

Transformations Overview

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.

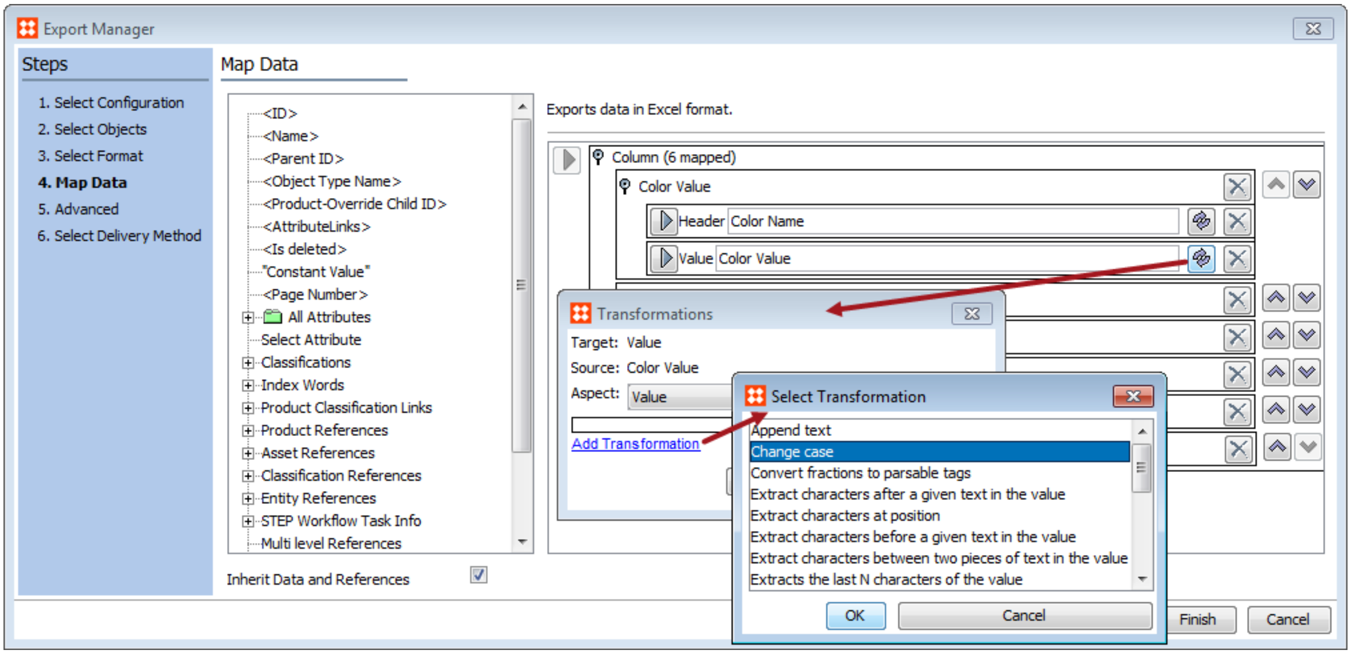
Transformations that are textual in nature and cannot be applied to attribute values that are not textual.

Note: There is no preview of the effect of transformation instructions. It is a good idea implement one or two transformations at a time, save the configuration, and gradually build the complete set of transformations.

Transforming Data

Generally, you can use the following steps to apply a transformation to a row in the list of columns being exported.

1. On the **Map Data** screen, in the right panel, open the flipper for the row that contains the data to transform
2. Click the transformation button .
3. In the **Transformations** dialog, click the **Add Transformations** link.
4. In the **Select Transformations** dialog, select the relevant transformation and click **OK**.



Exports data in Excel format.

Header: Color Name

Value: Color Value

Target: Value

Source: Color Value

Aspect: Value

5. Back on the **Transformations** dialog, enter the relevant information and click **Save**.

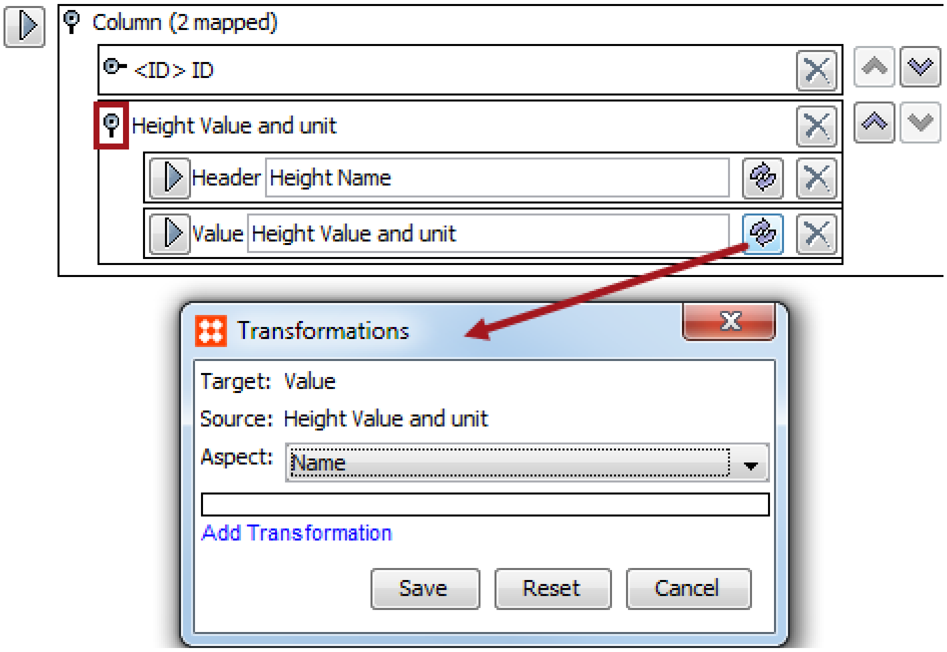
Steps for some of the most commonly-used transformations are included below.

For information about available transformations, see the **Transformation Options** section of the **Exporting Data** documentation.

For information about available transformation aspects, see the **Transformation Aspects** section of the **Exporting Data** documentation.

Removing Units from Values

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



Target: Value

Source: Height Value and unit

Aspect: Name

3. In the **Transformations** dialog, from the **Aspect** list, select **Value**.
4. Repeat the previous steps to remove units from other attributes if needed.

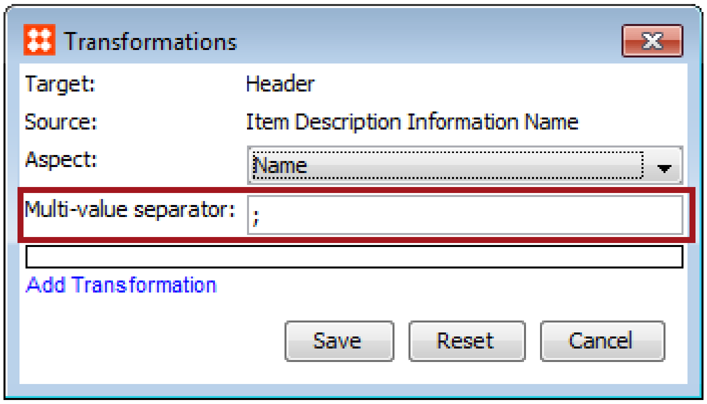
Adding Units to Values

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

Multi-value Separator

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

Note: When exporting Excel / CSV files, select a different character from the one used to delimit columns.



Target: Header

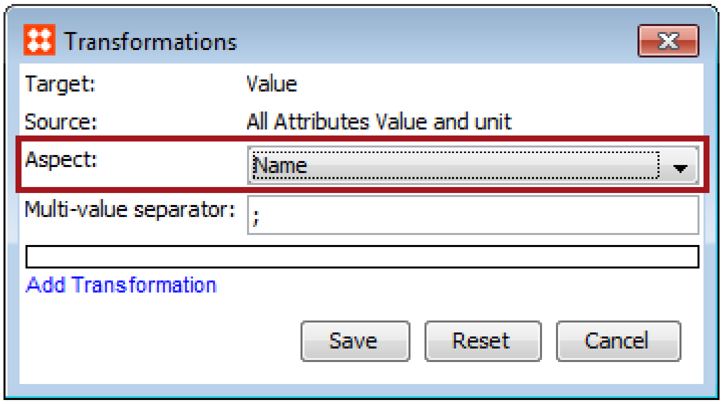
Source: Item Description Information Name

Aspect: Name

Multi-value separator: ;

Transformation Aspects

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.



Target: Value
Source: All Attributes Value and unit
Aspect: Name
Multi-value separator: ;

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.

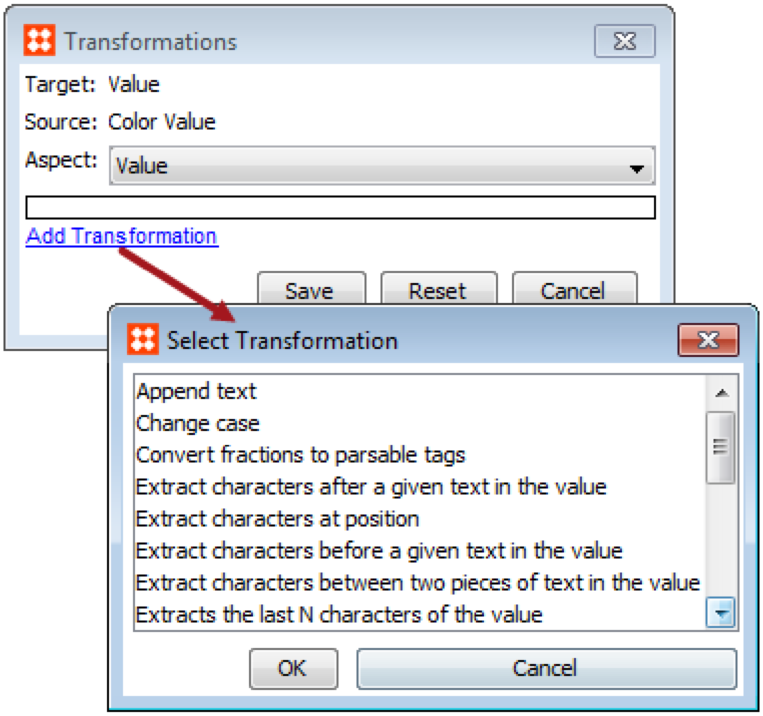
Aspect	Description
Name	Extracts the name of the attribute and not the attribute value. For example, if an attribute has the ID '1234' and the name 'Weight', then the word 'Weight' is extracted per product.
ID	Extracts the ID of the attribute not the attribute value. For example, if an attribute has 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).

Aspect	Description
Unit	Extracts the name of the unit associated with the product attribute value but not the actual 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'.

For an extended example of mapping aspects and constants, see the **Map Data Aspect and Constant Example** section of **Exporting Data** documentation.

Transformation Options

When you click the **Add Transformation** link on the Transformations dialog, the following options are available.



Transformations

Target: Value

Source: Color Value

Aspect: Value

Select Transformation

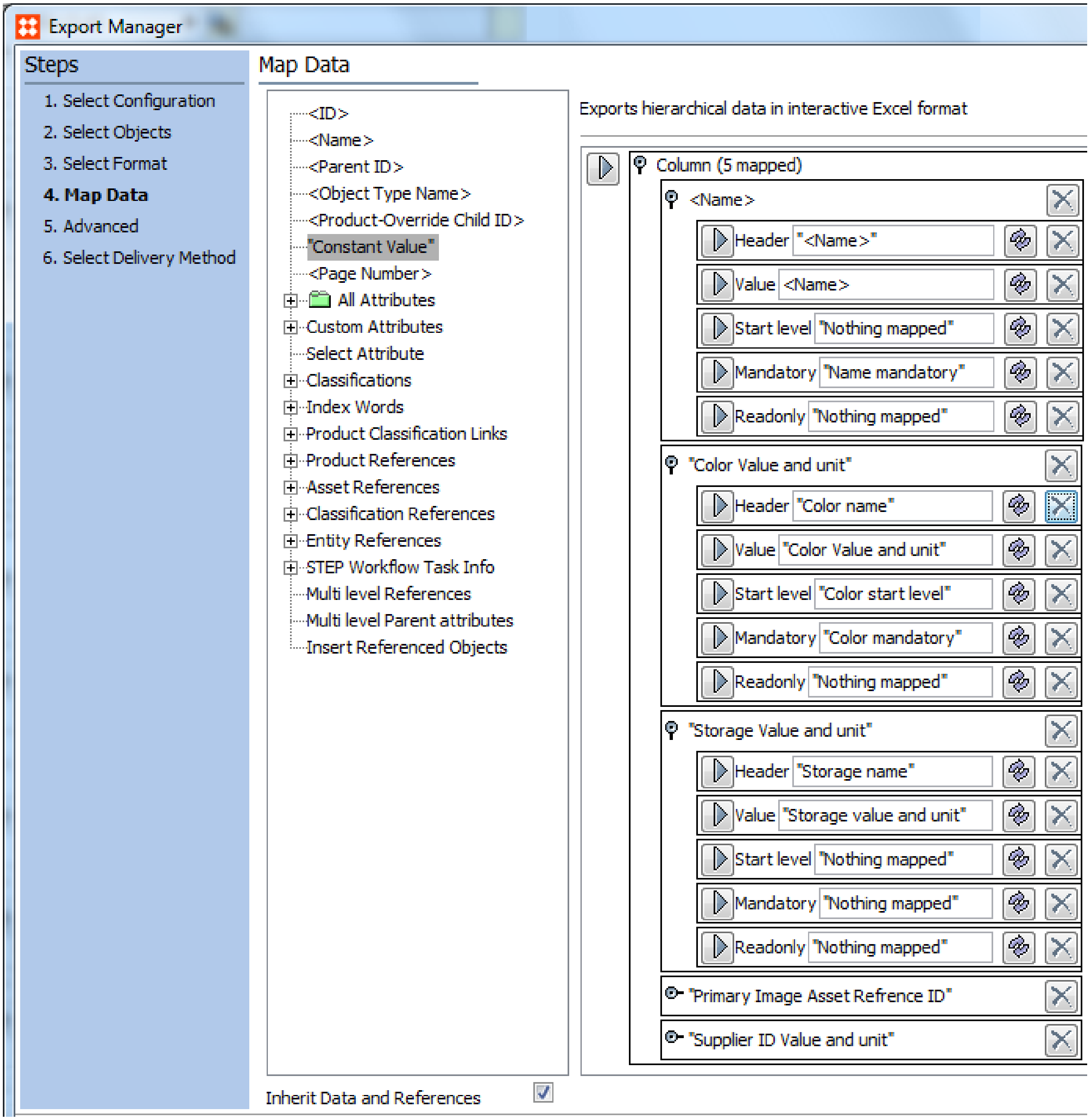
Append text
Change case
Convert fractions to parsable tags
Extract characters after a given text in the value
Extract characters at position
Extract characters before a given text in the value
Extract characters between two pieces of text in the value
Extracts the last N characters of the value

Transformations	Description
Append text	Adds the specified text, such as a suffix, to the end of the value.
Change case	Formats 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.
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/><bfldin/>1<FractionBar/><StartDenominator/>2<EndFraction/>cm You have to set up an output formatting to Indesign for <StartNumerator/>, <bfldin/>, <FractionBar/>, <StartDenominator/> and <EndFraction/> to mount the fractions.
Extract characters	Extracts a part of the value, starting after the occurrence of the specified text string.

Transformations	Description
after a given text in the value	
Extract characters at position	Extracts a specified number of characters after a specified position in a text string.
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.
Extract characters between two pieces of text in the value	Extracts the text between the specified starting text string and the specified ending text string.
Extract the last N characters of the value	Extracts the specified last character of the value.
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.
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	Replaces substrings of text in the value with another string of text. Replaces the text on all matches or on the first match.

Transformations	Description
value using a regular expression	
Replace the whole value	Replaces the specified value with a different value. You must enter the entire value. Only works on text-based attributes.
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.
Transform date by locale	Transforms Date, ISO Date, or ISO Date and Time attributes to localized formats. Localized dates follow the DateFormat.SHORT JavaScript formatting style. Times will be shown in the DateFormat.MEDIUM formatting style. 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.

Smartsheet Export Mapping



Export Manager

Steps

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

Map Data

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

Inherit Data and References ☑

Exports hierarchical data in interactive Excel format

Column (5 mapped)

<Name>
- Header "<Name>"
- Value <Name>
- Start level "Nothing mapped"
- Mandatory "Name mandatory"
- Readonly "Nothing mapped"

"Color Value and unit"
- Header "Color name"
- Value "Color Value and unit"
- Start level "Color start level"
- Mandatory "Color mandatory"
- Readonly "Nothing mapped"

"Storage Value and unit"
- Header "Storage name"
- Value "Storage value and unit"
- Start level "Nothing mapped"
- Mandatory "Nothing mapped"
- Readonly "Nothing mapped"

"Primary Image Asset Refrence ID"

"Supplier ID Value and unit"

The previous settings only apply to Smartsheet exports.

For Multi-object Smartsheets:

- The **Parent ID** setting is specific to Multi-object Smartsheets. It is used as a Parent ID selector in the sheets so users can indicate the correct parent IDs for the products they work with.
- The **Object Type Name** setting is specific to Multi-object Smartsheets and is used as an Object Type selector in the sheets so users can indicate the object types for the products they work with.

For Multi-level Smartsheets:

- The **Start level** setting is specific to Multi-level Smartsheets. The setting specifies that the selected column starts a new object level.
- The **Start level** can be left empty because it is implicit. However, a value must be provided for all other Start levels. The actual value is not important.

  The Start level columns are also used for grouping products at each level. For example, all sheet rows with "T-shirt" in the Name column become 1 product, and all "Pink" color columns on rows with "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.
- **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.

More About Attribute Smartsheet 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.

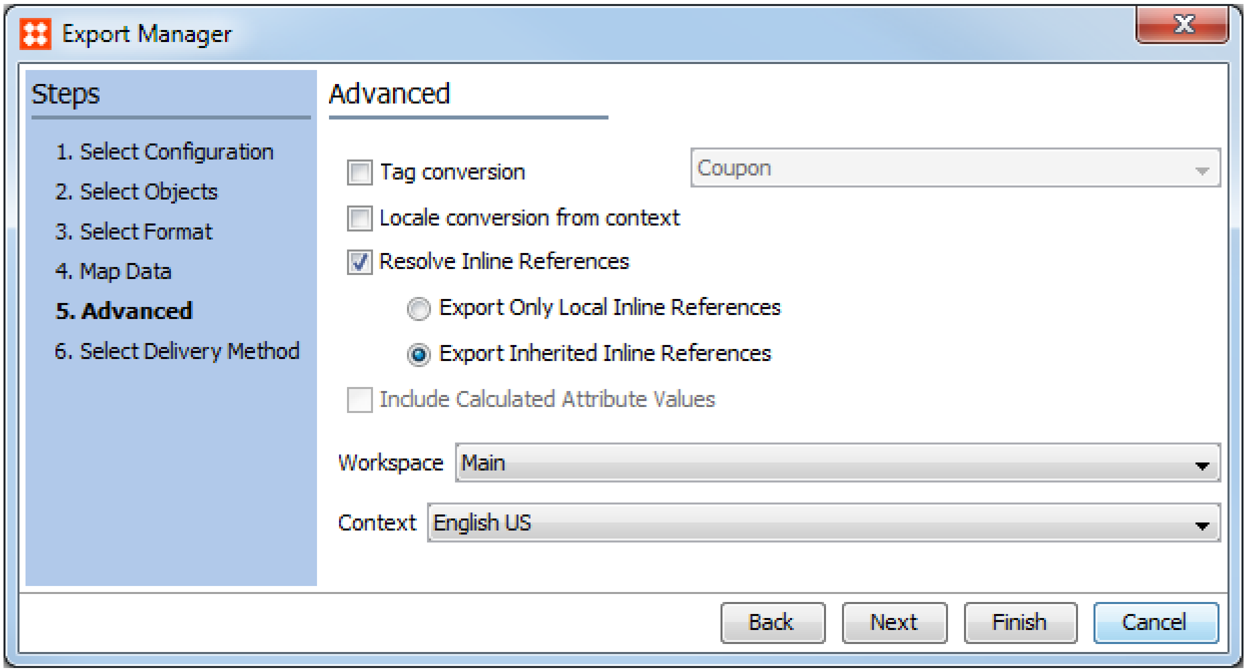
Important: Do not add any kind of transformation to the value of a mapping unless it is marked as read-only. Transformations are applied during Smartsheet data export. However, during Smartsheet import, the column will be handled as if it contains Value and Unit. This can lead to import errors or unexpected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

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: Smartsheet only supports LOV filters on attributes and attribute links.

Step 5 - Advanced



Export Manager

Steps

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

Advanced

☐ Tag conversion — Coupon

☐ Locale conversion from context

☑ Resolve Inline References

○ Export Only Local Inline References

● Export Inherited Inline References

☐ Include Calculated Attribute Values

Workspace: Main

Context: English US

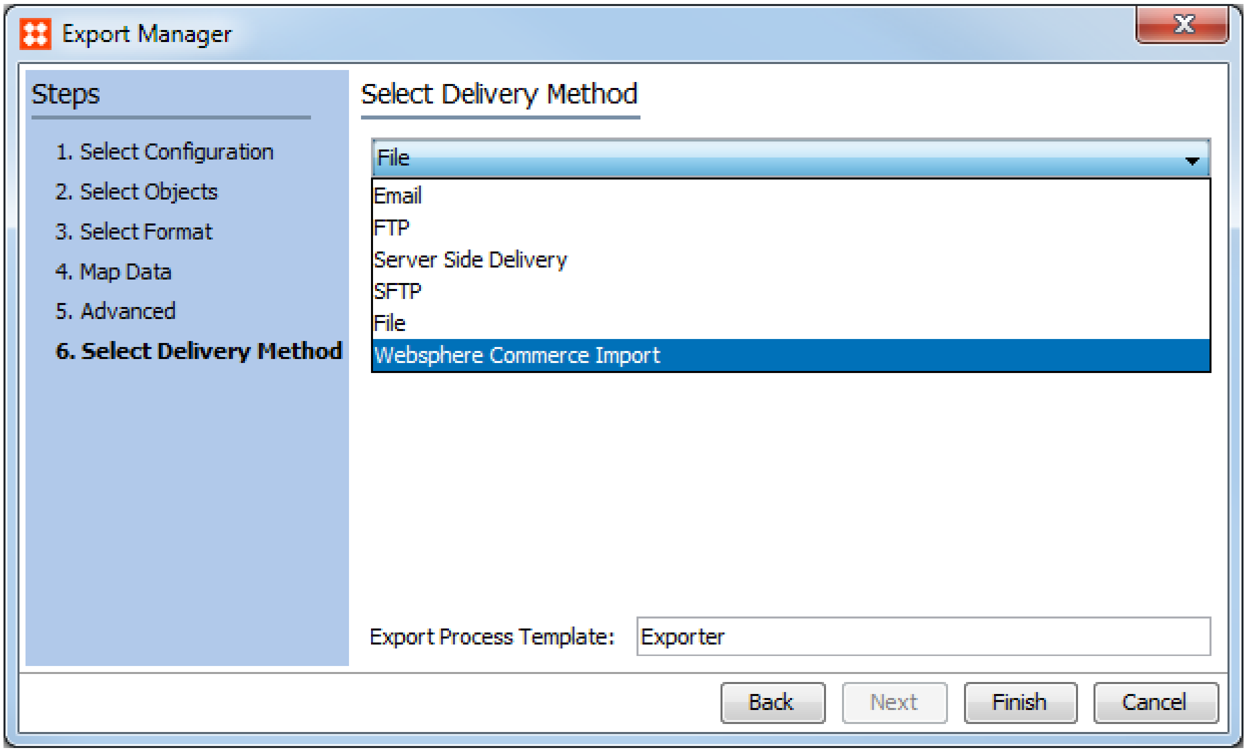
Set the following additional export parameters as needed:

- **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.
- **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.

Step 6 - Select Delivery Method



File

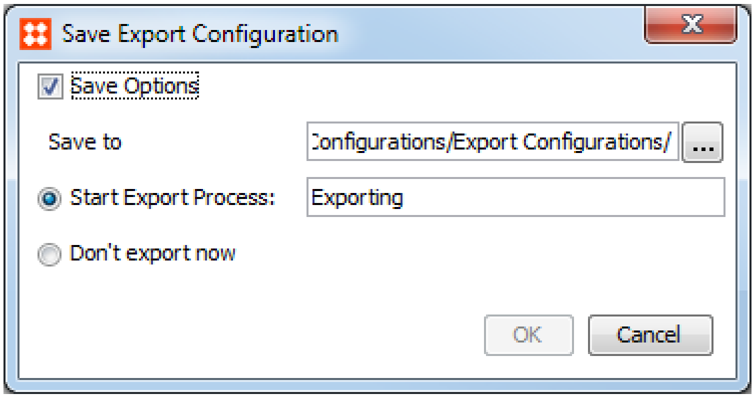
Export Process Template: Exporter

1. In the **File Delivery** list, select the relevant export option.
   - **Email**: Delivers a file as an email attachment. You must specify the email address of the recipient, a subject, and body text.
   - **File**: Delivers a file in the background process. You are notified when the file is ready for download.
   - **FTP**: Delivers a file using file transfer protocol (FTP). You must specify the host name, user name, password, and the file name used to deliver the file.
   - **Server Side Delivery**: Exports the file into a folder located on the Application Server of the STEP system. The file is only delivered if the specified path has read / write access on the server.
   - **SFTP**: Delivers a file using the Secure File Transfer Protocol (SFTP). You must specify the host name, user name, password, and file name.
   - **Websphere Commerce Import**: Depends on your system setup.
2. Click **Finish** to display the Saving the Configuration window.

Save Export Configuration

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



Save Export Configuration

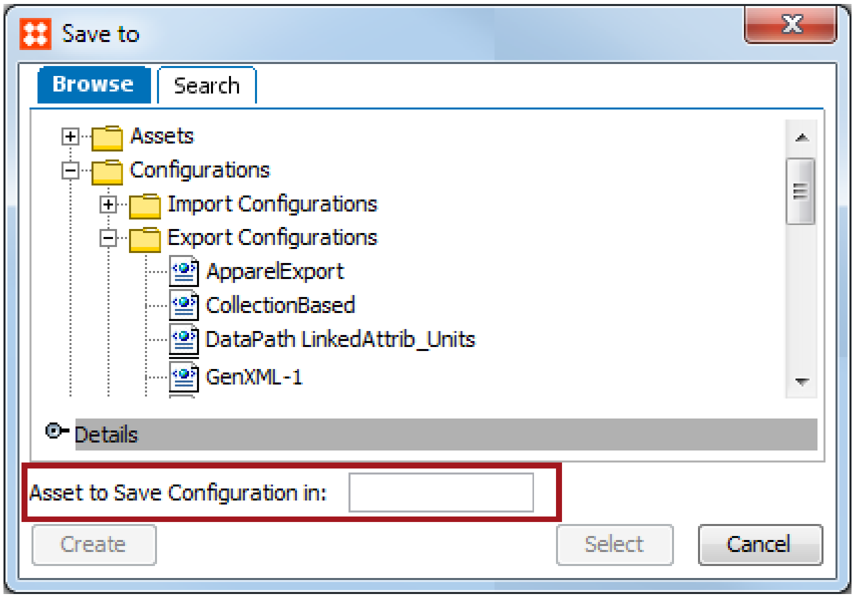
☑ Save Options

Save to: Configurations/Export Configurations/

● Start Export Process: Exporting

○ Don't export now

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.



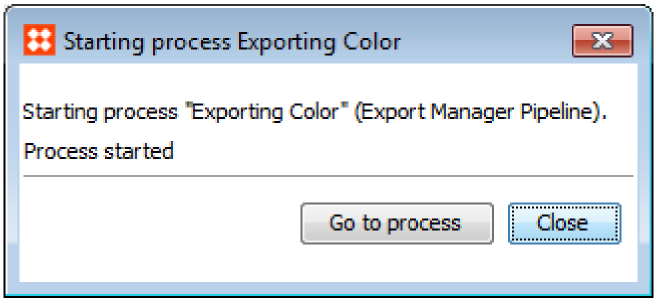
Save to

Asset to Save Configuration in:

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 complete the wizard. If you started an export, the **Starting Process** dialog appears. For more information, see **Checking Data Exports**.

Checking Data Exports

When you launch a data export process, the system starts a workflow process. It assigns a process number to that particular data export, and gives it the name you assigned when you saved the export.



Starting process "Exporting Color" (Export Manager Pipeline).

Process started

Monitoring the Export Process

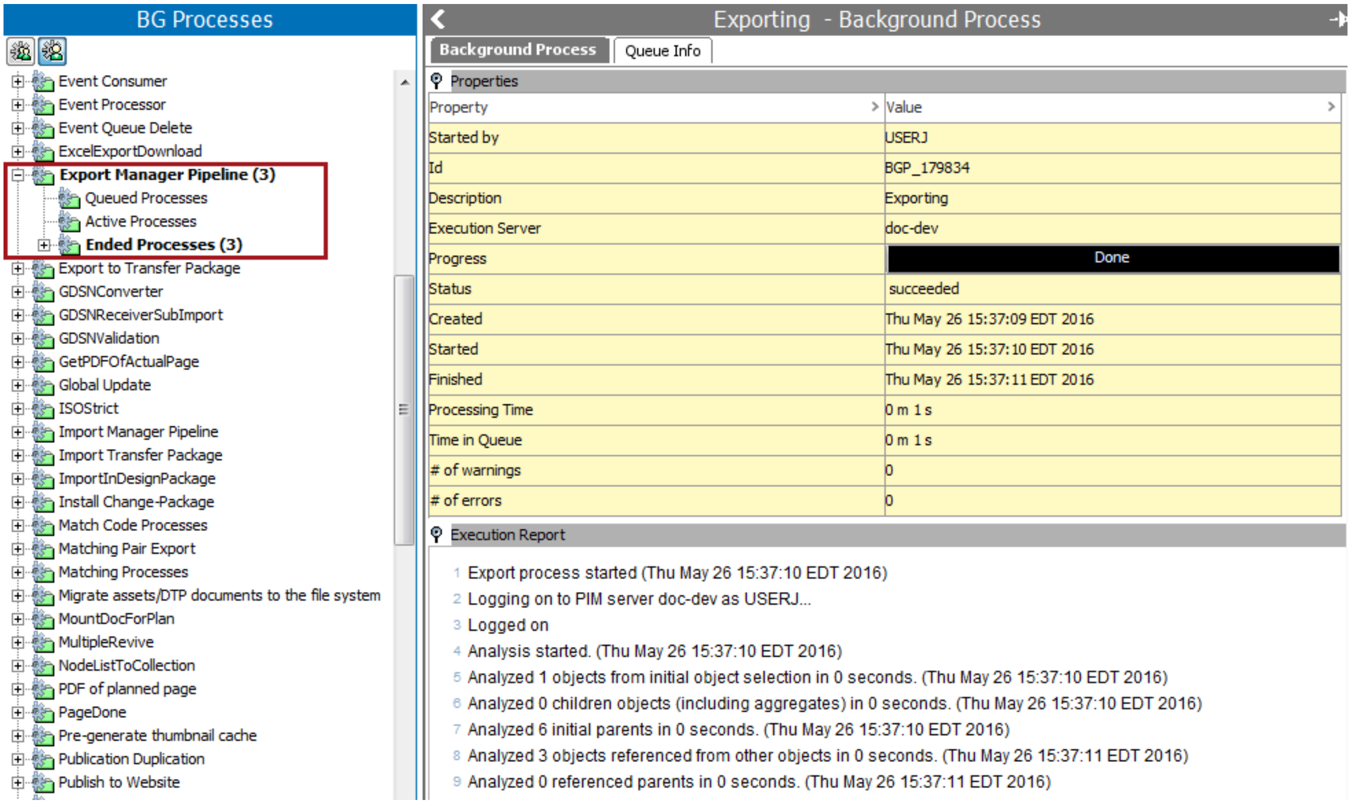
There are two ways to monitor a particular data import process:

- In the **Starting Process** window (that appears when you launch the import), click the **Go To Process** button.
- On the **BG Process** tab select the relevant process. For processes that have completed, look in **Ended Processes**. For processes that are currently being performed, look in **Active Processes**.

Execution Report

The process 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 HelpDesk, the information can be useful.



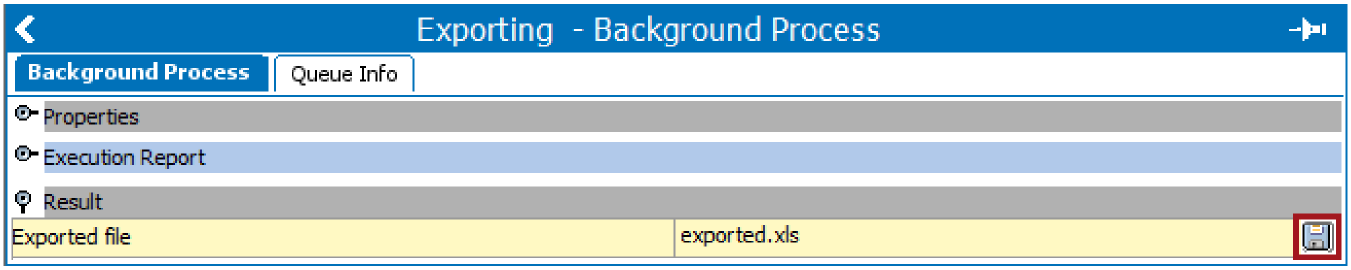
Property	Value
Started by	USERJ
Id	BGP_179834
Description	Exporting
Execution Server	doc-dev
Progress	Done
Status	succeeded
Created	Thu May 26 15:37:09 EDT 2016
Started	Thu May 26 15:37:10 EDT 2016
Finished	Thu May 26 15:37:11 EDT 2016
Processing Time	0 m 1 s
Time in Queue	0 m 1 s
# of warnings	0
# of errors	0

```
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)
```

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.



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.

Exporting Images and Documents

Assets are exported from STEP using the Export Images and Documents wizard. Since images are the most frequently loaded asset to the STEP there are a series of options and parameters you can set when exporting image assets.

This material describes standard asset exports only. For information on asset push, see the **Asset Push User Guide**.

Audience

1. You should have a good working knowledge of standard file formats and file extensions, such as .doc, .pdf, .gif, etc.
2. You should know about image file formats, and have a working knowledge of what they are used for, and what the ramifications are of up-sizing or down-sizing images on export.

Asset Exports - Functionality List

- You can export either a single asset or all assets directly linked to the specified folder.
- For images, you can specify whether or not to use standard file name extensions.
- For images, you can specify the image file format you want to export the images in, and a series of other image related options including resolution, sizing, and color management.
- For images, you can choose from a list of predefined file output formats and specifications called pipelines. This list varies depending on your system set-up. The list saves you from having to re-enter the same specifications each time you export images.

Starting an Export of Images or Documents

Before you launch the Export Images and Documents wizard, select the appropriate context in STEP.

There are two methods of starting an asset export process. In both cases, first select a folder to enable the Export Images & Documents option, then in the Tree:

- Select a classification folder, and then from the File menu point to Export, and select **Images & Documents**.
- Select a classification or asset, right-click and then click **Export Data Below**.

Export Images and Documents Wizard

The Export Images and Documents wizard allows you to export assets and can involve the following steps:

Select Export Location: In this screen you select the location that you want to download your assets to.

Image Conversion: In this screen you specify if you want to convert the format, size and color settings of the exported images. If you want images to be converted you can select a predefined image conversion configuration, script or pipeline. If you do not want to use a predefined image conversion configuration, script or pipeline, you can specify the conversion directly in the wizard by selecting 'Custom...'. This enables the remaining wizard screens.

File Format: In this screen you specify the file format of the images you want to export.

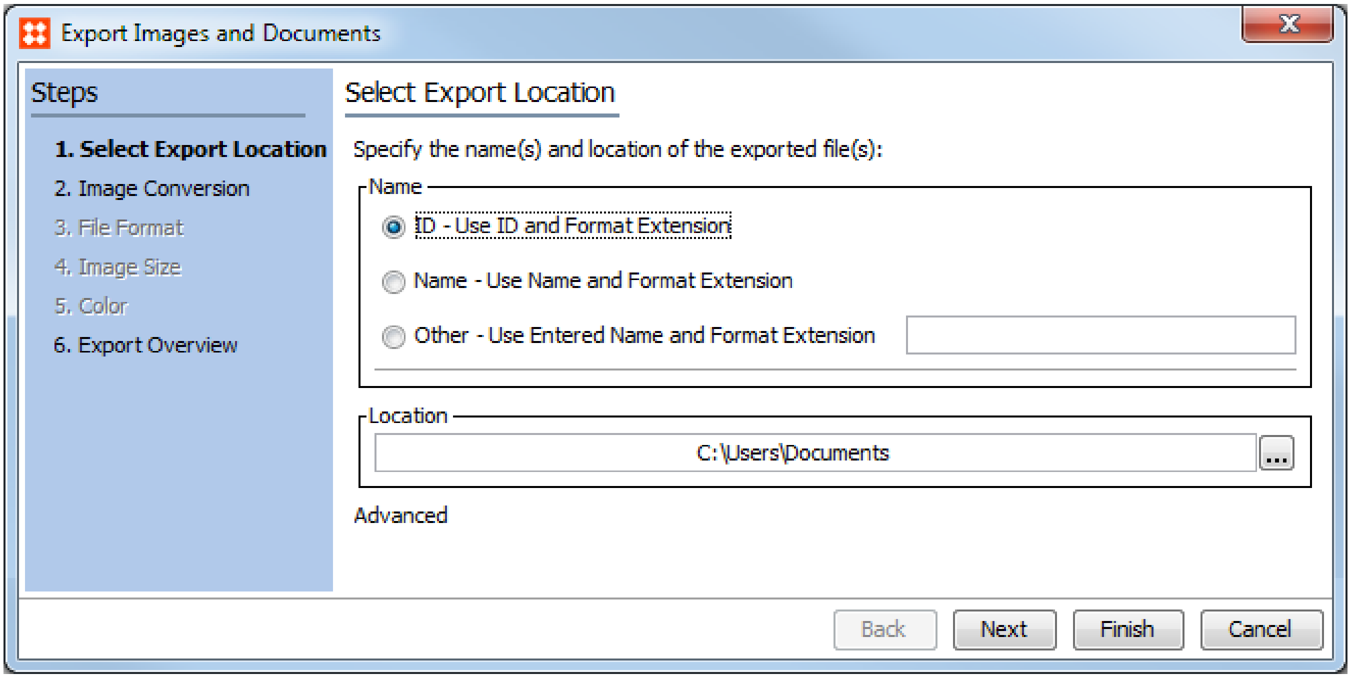
Image Size: In this screen you specify the size of the images upon export.

Color: In this screen you specify the color definition of the exported images.

Export Overview: In this screen you get an overview of the export settings so that you can determine if your options appear correct before you start the export.

Note: Any asset that is not an image is considered to be a document. You cannot modify an asset that is not an image. Documents are exported from STEP in the same format and manner that they were loaded into STEP.

Step 1 - Select Export Location



Select Export Location

Specify the name(s) and location of the exported file(s):

Name

- ● ID - Use ID and Format Extension
- ○ Name - Use Name and Format Extension
- ○ Other - Use Entered Name and Format Extension

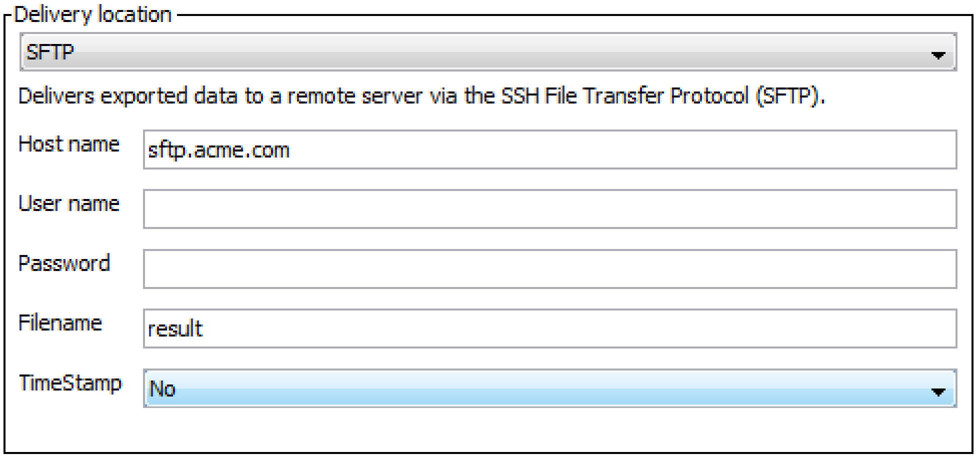
Location

C:\Users\Documents ...

Advanced

1. In the **Name** area, select one of the following options.
   - **ID – Use ID and Format Extension:** Builds the file names of the exported files by taking the asset's ID and appending the file format's associated extension. For example, a PDF document with the ID 1234567 is exported as 1234567.pdf
   - **Name – Use Name and Format Extension:** Builds the file names of the exported files by taking the asset's name and appending the file format's associated extension. For example, a PDF document with the file name Installation Manual is exported as Installation Manual.pdf.
   - **Other – Use Entered Name and Format Extension:** This option available when you have selected only one asset for export. You can enter any file name and extension that you want.
2. In the **Location** area, select the delivery location. You can toggle between **Advanced** and **Basic** export options, by clicking the Advanced / Basic text below the location field.
   - In **Basic** mode, click the ellipsis button **(…)**, and then navigate to the preferred location.
   - In **Advanced** mode, use the dropdown to choose a delivery option.
     - **FTP** delivers a file using file transfer protocol (FTP). You must specify the host name of the remote server, the user name and password used to log on to the remote server, a file name, and if the file name should be provided with a time stamp.

- **SFTP** delivers a file using the Secure File Transfer Protocol (SFTP). You must specify the host name of the remote server, the user name and password used to log on to the remote server, a file name, and if the file name should be provided with a time stamp.



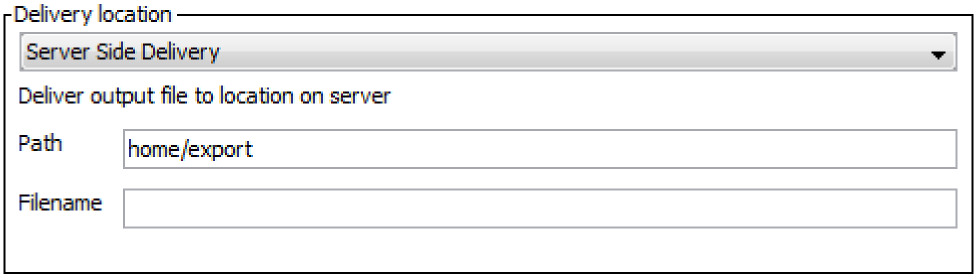
Delivery location

SFTP

Delivers exported data to a remote server via the SSH File Transfer Protocol (SFTP).

Host name	sftp.acme.com
User name	
Password	
Filename	result
TimeStamp	No

- **Server Side Delivery** exports the file into a folder located on the Application Server of the STEP system. The file is only delivered if the specified path has read / write access on the server. Enter the path of the application server and the name of the file.



Delivery location

Server Side Delivery

Deliver output file to location on server

Path	home/export
Filename	

Step 2 - Image Conversion

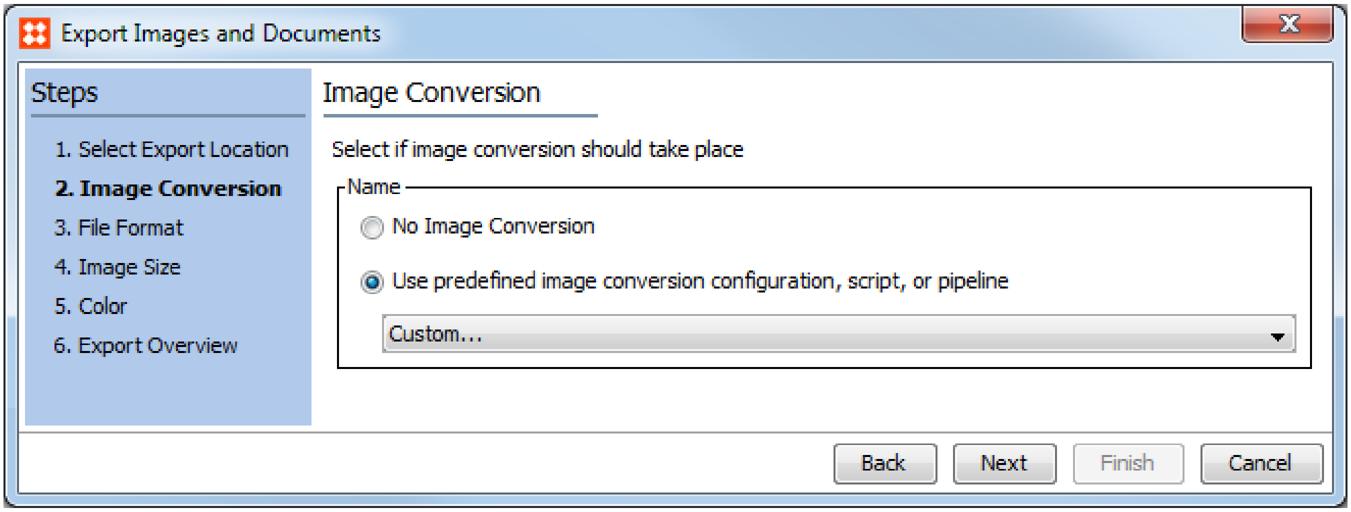


Image Conversion

Select if image conversion should take place

Name

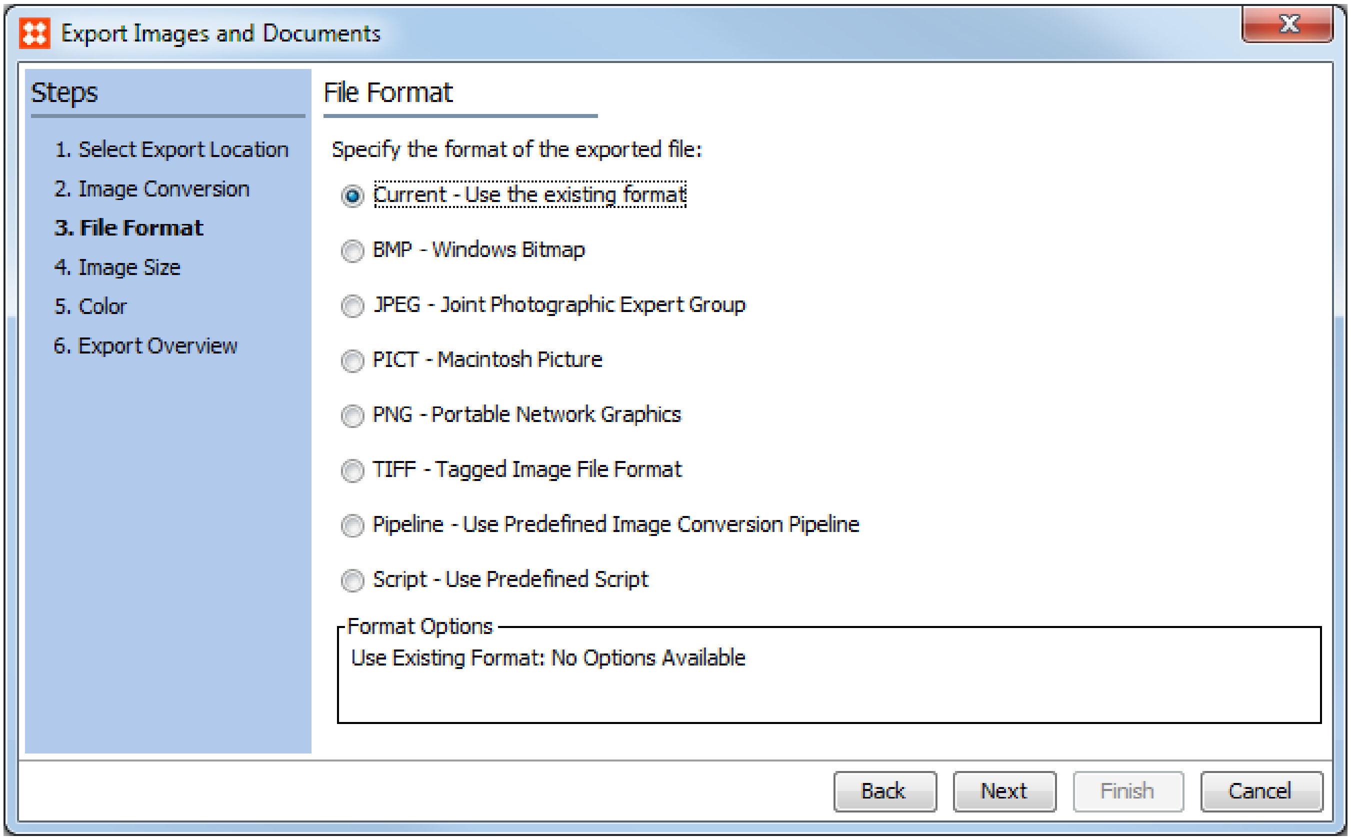
○ No Image Conversion

● Use predefined image conversion configuration, script, or pipeline

Custom...

1. For Name, select **No Image Conversion** to export the images as is. No additional information is required.
2. If an image conversion is required to change the image settings on export, select **Use predefined image conversion configuration, script or pipeline**.
3. Use the dropdown to select one of the predefined image conversion configurations, scripts, or pipelines available on your system. Select **Custom** to specify a new image conversion using the File Format, Image Size, and Color wizard steps.
4. Click **Next** to continue or click **Finish** to start the export.

Step 3 - File Format



File Format

Specify the format of the exported file:

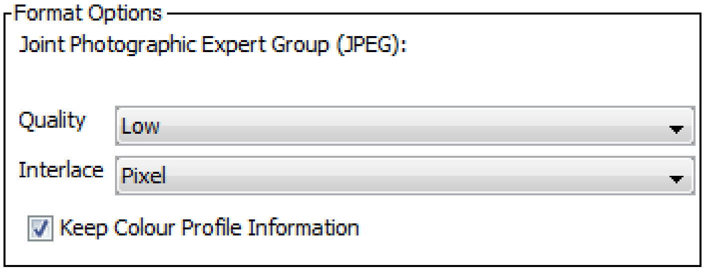
- ● Current - Use the existing format
- ○ BMP - Windows Bitmap
- ○ JPEG - Joint Photographic Expert Group
- ○ PICT - Macintosh Picture
- ○ PNG - Portable Network Graphics
- ○ TIFF - Tagged Image File Format
- ○ Pipeline - Use Predefined Image Conversion Pipeline
- ○ Script - Use Predefined Script

Format Options

Use Existing Format: No Options Available

1. Set the format required for export by selecting one of the following options:
    - **Current – Use the existing format**: Keeps the current (original) file format of the images selected for export.
    - **BMP – Windows bitmap**: Converts all image files in your selected folder to .BMP files.
    - **JPEG – Joint Photographic Expert Group**: Converts all image files in your selected folder to .JPG files. When you select this format, you have to specify the following:
        - **Quality Level**: Low, Medium, High Maximum
        - **Interlace**: Pixel or Plane

- **Keep Color Profile Information**



Format Options

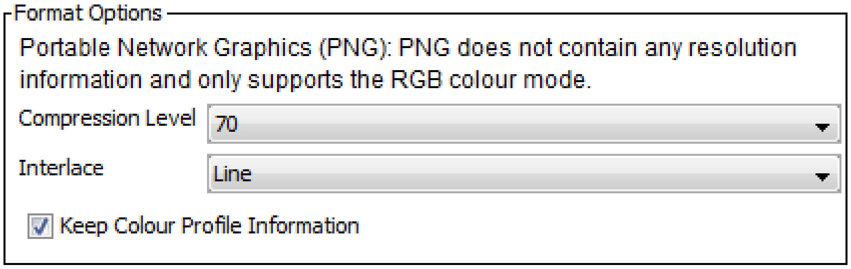
Joint Photographic Expert Group (JPEG):

Quality: Low

Interlace: Pixel

☑ Keep Colour Profile Information

- **PICT – Macintosh Picture**: Converts all image files in your selected folder to .PICT files.
- **PNG – Portable Network Graphics**: Converts all image files in your selected folder to .PNG files. When you select this format, you must specify the following:
  - **Compression Level**: From 0 to 90 in increments of 10
  - **Interlace**: Line or Pixel
  - **Keep Color Profile information**



Format Options

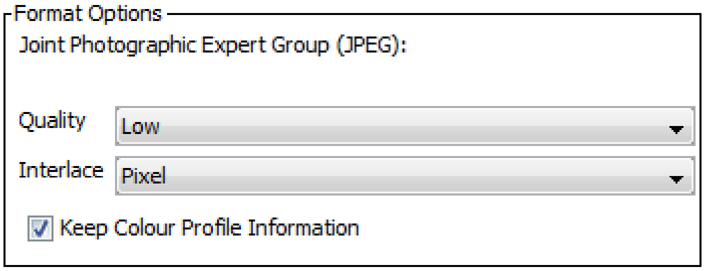
Portable Network Graphics (PNG): PNG does not contain any resolution information and only supports the RGB colour mode.

Compression Level: 70

Interlace: Line

☑ Keep Colour Profile Information

- **TIFF – Tagged Image File Format**: Converts all image files in your selected folder to .TIFF files. When you select this format, you must specify the following:
  - **Compression Type**: None, ZIP, JPEG, RLE, Fax, Group 4
  - **JPEG Quality**: Low, Medium, High Maximum
  - **Interlace**: Line, Pixel, Plane
  - **Keep Color Profile Information**



Format Options

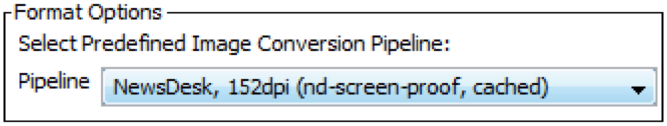
Joint Photographic Expert Group (JPEG):

Quality: Low

Interlace: Pixel

☑ Keep Colour Profile Information

- **Pipeline – Use Predefined Image Conversion Pipeline**: Uses a standard image pipeline. The available conversion pipelines depend on you system setup. Selecting a pipeline automatically provides the image conversion options each time you export images. No additional wizard settings are required. Click **Next** to display the **Export Overview** or **Finish** to start the export.



Format Options

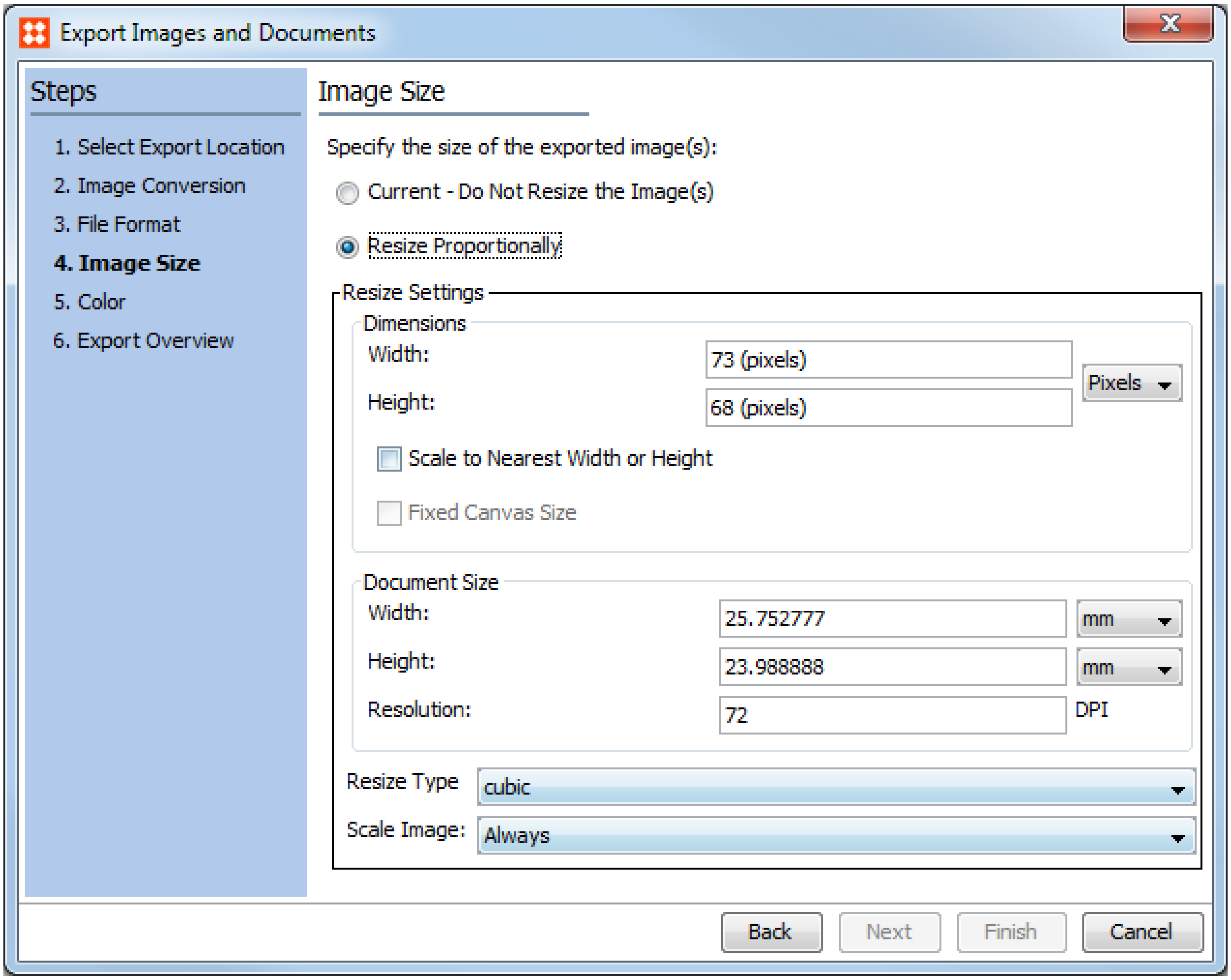
Select Predefined Image Conversion Pipeline:

Pipeline: NewsDesk, 152dpi (nd-screen-proof, cached)

- **Script - Use Predefined Script**: Uses a predefined script to convert the images. Scripts are system dependent. Selecting a script automatically provides the image conversion options each time you export images. No additional wizard settings are required. Click **Next** to display the **Export Overview** or **Finish** to start the export.

2. Click **Next** or click **Finish** to start the export.

Step 4 - Image Size



Export Images and Documents

Steps

1. Select Export Location
2. Image Conversion
3. File Format
4. **Image Size**
5. Color
6. Export Overview

Image Size

Specify the size of the exported image(s):

○ Current - Do Not Resize the Image(s)

● Resize Proportionally

Resize Settings

Dimensions

Width: 73 (pixels)

Height: 68 (pixels)

Pixels

□ Scale to Nearest Width or Height

□ Fixed Canvas Size

Document Size

Width: 25.752777 mm

Height: 23.988888 mm

Resolution: 72 DPI

Resize Type: cubic

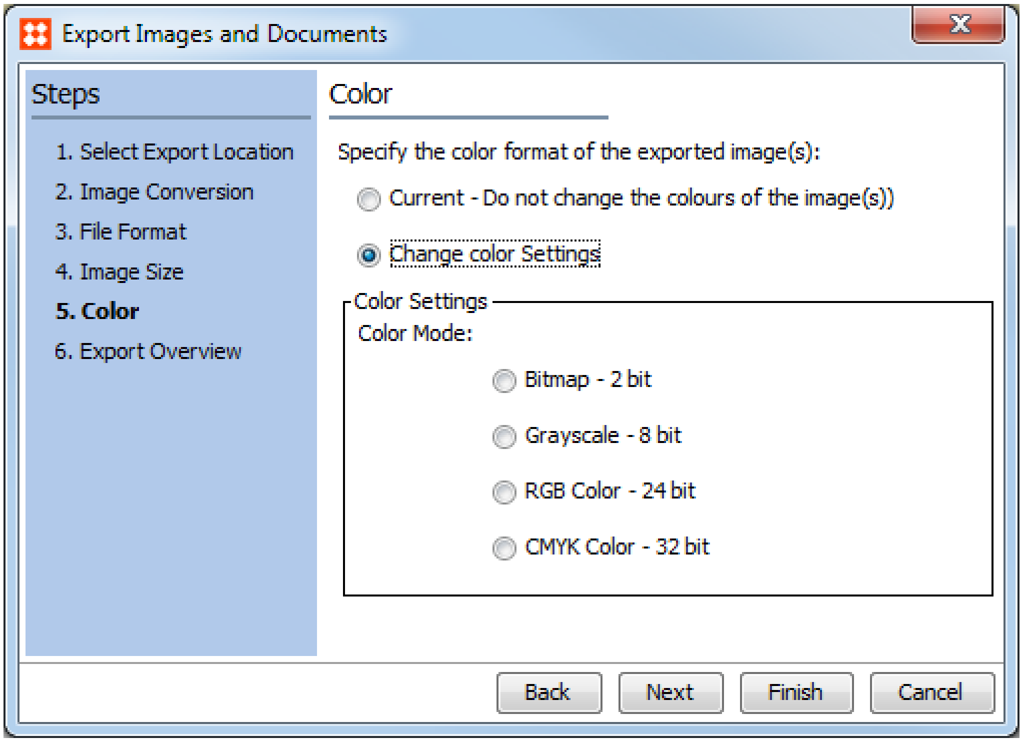
Scale Image: Always

1. Select **Resize Proportional** if you want to resize the images. If you do not want to resize the images select **Current - do not resize the image**.
2. For Resize Settings, use the **Dimensions** area enter the desired height *or* width in the selected unit of measure. Because the images are scaled proportionally, when you enter the first dimension, the other values change accordingly.
3. For Resize Settings, use the **Document Size** area to show the print size of the image and the resolution. In Resolution, specify the preferred resolution if different from the current resolution.
4. For **Resize Type**, select the preferred resize type: cubic, scale and subsample. These types are similar to those offered by Photoshop.

Note: It is easier to successfully reduce the resolution of an image than it is to increase it. Also, it is recommended to scale TIFF images by no more than 140%.

5. For **Scale Image**, select when you want the to scale the images from the following options:
    - **Always**: images are always scaled to the specified size.
    - **When Smaller**: images are scaled when they are smaller than the specified size.
    - **When Larger**: images are scaled when they are larger than the specified size.
6. Click **Next** to continue or click **Finish** to start the export.

Step 5 - Color



Specify the color format of the exported image(s):

○ Current - Do not change the colours of the image(s))

● Change color Settings

Color Settings

Color Mode:

○ Bitmap - 2 bit

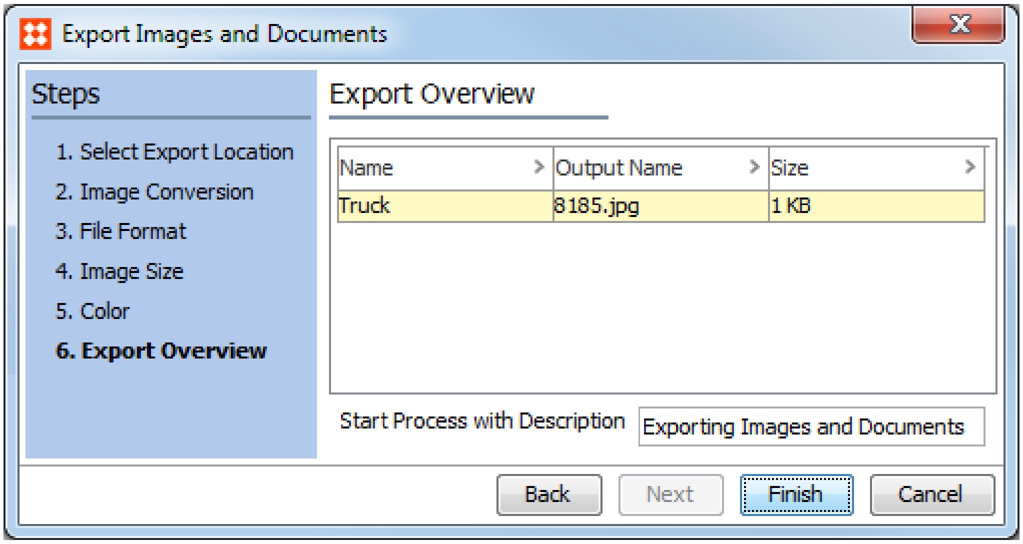
○ Grayscale - 8 bit

○ RGB Color - 24 bit

○ CMYK Color - 32 bit

1. Select **Change color settings** if you want to modify the image color, otherwise, select **Current - do not change the colors of the image**.
2. For **Color Settings**, select the preferred color mode option.
3. Click **Next** to continue or click **Finish** to start the export.

Step 6 - Export Overview



Name	Output Name	Size
Truck	8185.jpg	1 KB

Start Process with Description: Exporting Images and Documents

1. Review the STEP name of the asset, the file name to be used for export, and the file size.
2. To start the export process, click **Finish**.
3. Monitor the output location identified in Step 1 to ensure that all selected assets were exported.

Creating Image Conversion Configurations

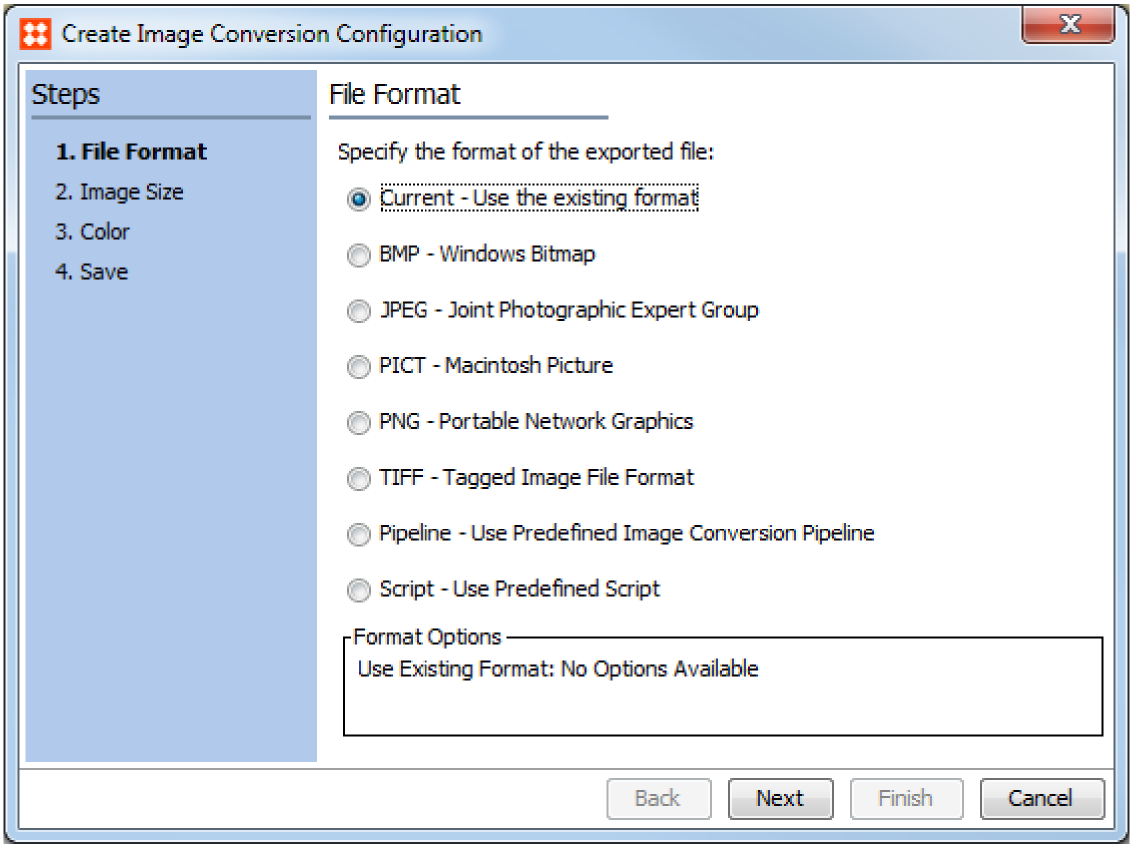
Image conversion configurations allow users to change the format, size, or image color settings of an asset when exporting images manually or with asset push. Once a configuration is created it is made available for all applicable asset pushes in the system.

For more information, see the **Asset Push Overview** section of the **Asset Push** documentation.

To Create an Image Conversion Configuration

1. Select a classification folder in **Tree**, then from the **Maintain** menu, hover over **Insert** and select **"New Image Conversion Configuration"**. The **Create Image Conversion Configuration** wizard appears.
2. Specify the file format of the images or keep the current format by selecting **Current - Use the existing format**, then click **Next**.

   For more information about the individual format options, see step 3 of the **Exporting Images and Documents** section of the **Export Manager** documentation.



Create Image Conversion Configuration

Steps

1. File Format
2. Image Size
3. Color
4. Save

File Format

Specify the format of the exported file:

● Current - Use the existing format

○ BMP - Windows Bitmap

○ JPEG - Joint Photographic Expert Group

○ PICT - Macintosh Picture

○ PNG - Portable Network Graphics

○ TIFF - Tagged Image File Format

○ Pipeline - Use Predefined Image Conversion Pipeline

○ Script - Use Predefined Script

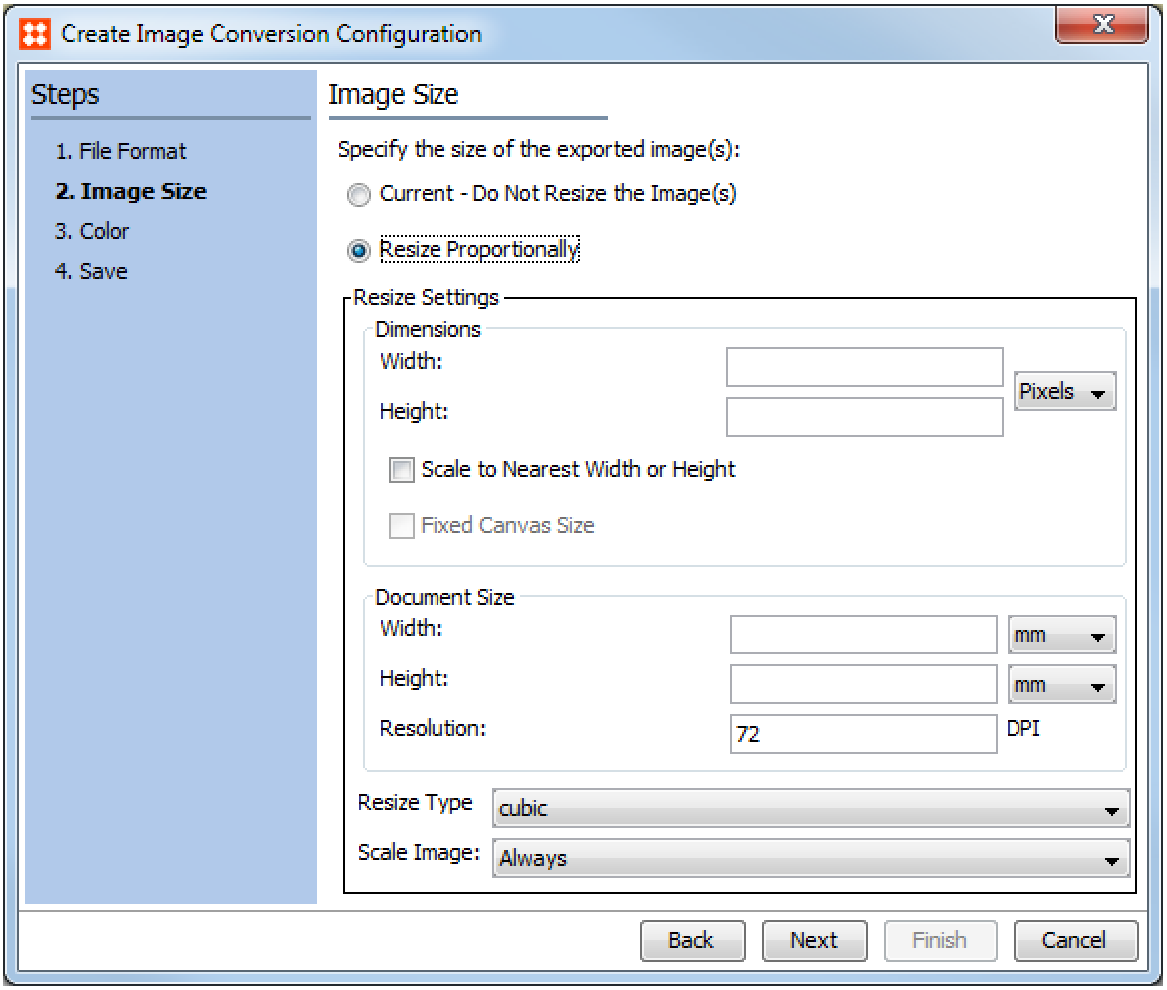
Format Options

Use Existing Format: No Options Available

Note: Steps 2 and 3 are disabled if the user selects the **Pipeline - use predefined image conversion pipeline option** or **Script - Use predefined script** options.

3. Select **Resize Proportionally** to change the size of the images, then specify the dimensions and document size. If no changes need to be made, select **Current - Do Not Resize the Image(s)** to export the images as they are. Once any required size changes are made, click **Next**.

   For more information about the individual format options, see step 4 of the **Exporting Images and Documents** section of the **Export Manager** documentation.



Create Image Conversion Configuration

Steps

1. File Format
2. **Image Size**
3. Color
4. Save

Image Size

Specify the size of the exported image(s):

○ Current - Do Not Resize the Image(s)

● Resize Proportionally

Resize Settings

Dimensions

Width:

Height:

Pixels

□ Scale to Nearest Width or Height

□ Fixed Canvas Size

Document Size

Width: mm

Height: mm

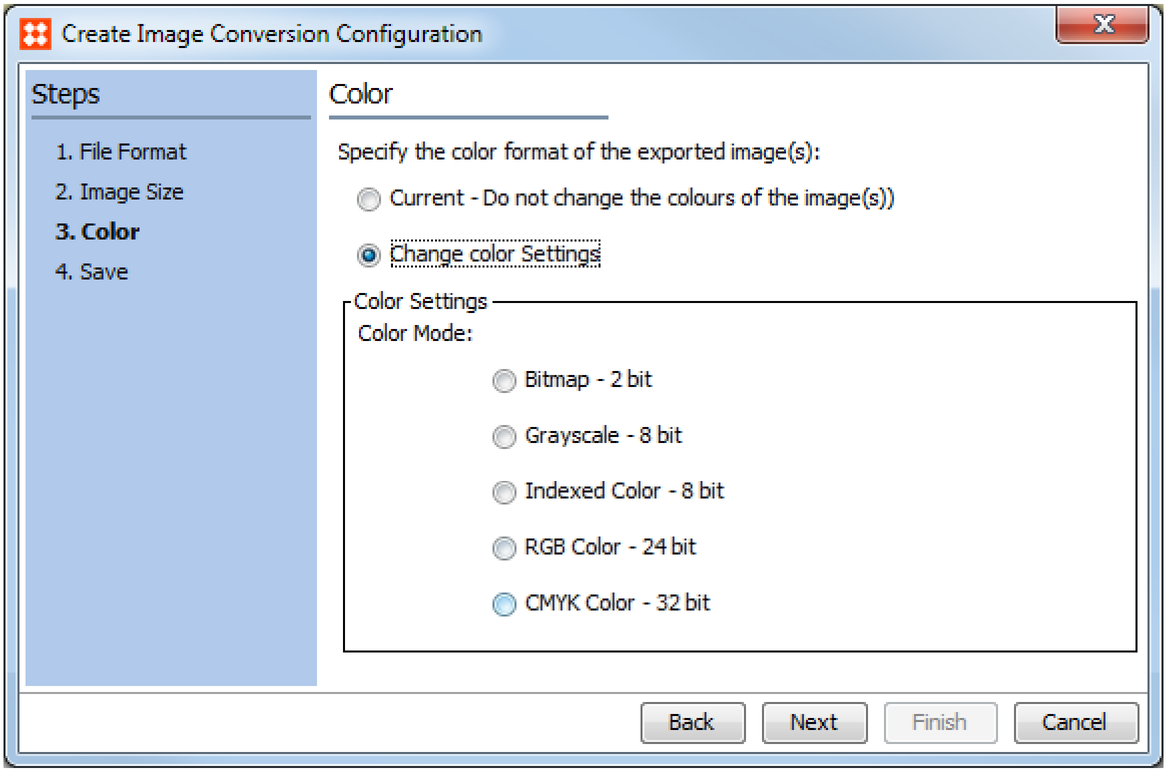
Resolution: 72 DPI

Resize Type cubic

Scale Image: Always

4. To change the color settings of the images, select **Change color Settings** and specify the desired color mode. If no changes need to be made, select **Current - Do not change the colours of the Image(s)** to export the images as they are.

   For more information about the individual format options, see step 5 of the **Exporting Images and Documents** section of the **Export Manager** documentation.



Create Image Conversion Configuration

Steps

1. File Format
2. Image Size
3. **Color**
4. Save

Color

Specify the color format of the exported image(s):

○ Current - Do not change the colours of the image(s))

● Change color Settings

Color Settings

Color Mode:

○ Bitmap - 2 bit

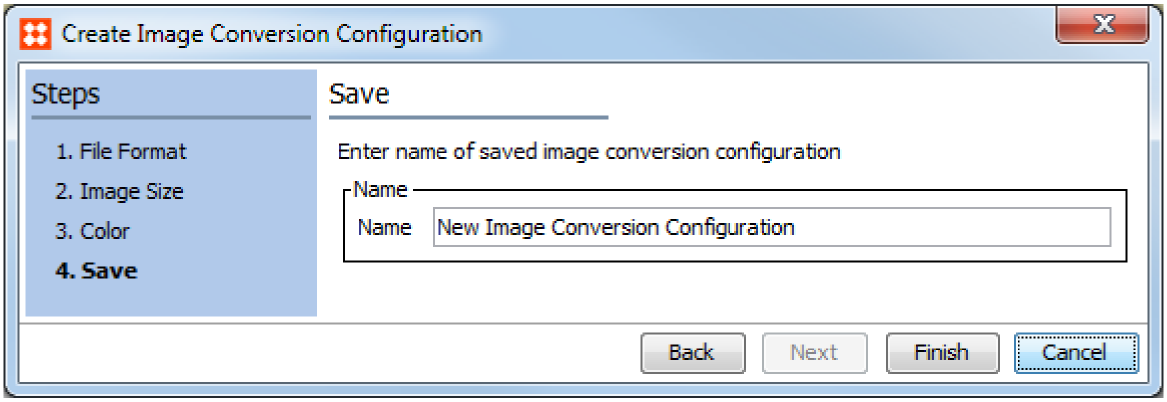
○ Grayscale - 8 bit

○ Indexed Color - 8 bit

○ RGB Color - 24 bit

○ CMYK Color - 32 bit

5. In the final screen, enter a name for the Image Conversion Configuration. Click **Finish** to save the configuration below the originally selected classification folder.



Create Image Conversion Configuration

Steps

1. File Format
2. Image Size
3. Color
4. **Save**

Save

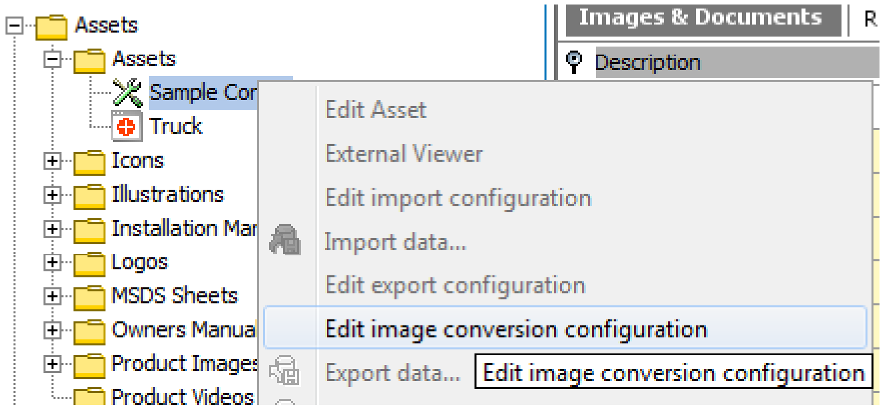
Enter name of saved image conversion configuration

Name

Name: New Image Conversion Configuration

Editing an Image Conversion Configuration

- In **Tree**, expand the relevant classification folder, right click the preferred image conversion configuration, and then select **Edit Image Conversion Configuration**.

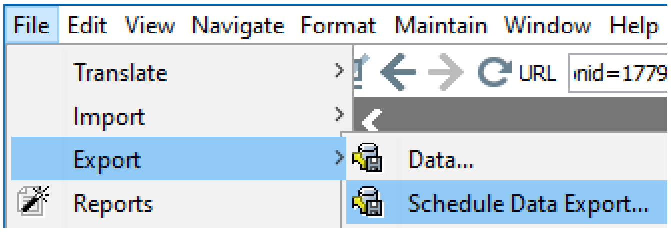


The **Edit Image Conversion Configuration** wizard appears. This wizard is identical to the Create Image Conversion Configuration wizard.

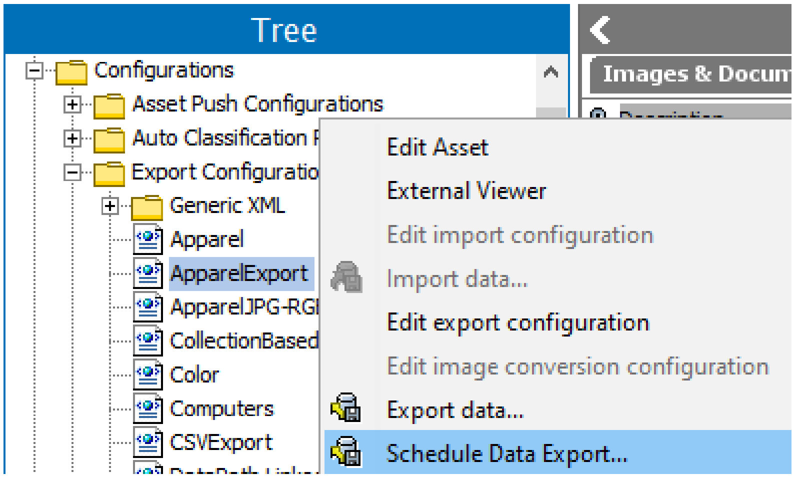
Scheduling Data Exports

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 **Data Export Manager Wizard** documentation and then save the configuration as described in the **Save Export Configuration** section.
3. Open the Schedule Data Export wizard using one of these methods:

- From the File menu > Export > **Schedule Data Export**.



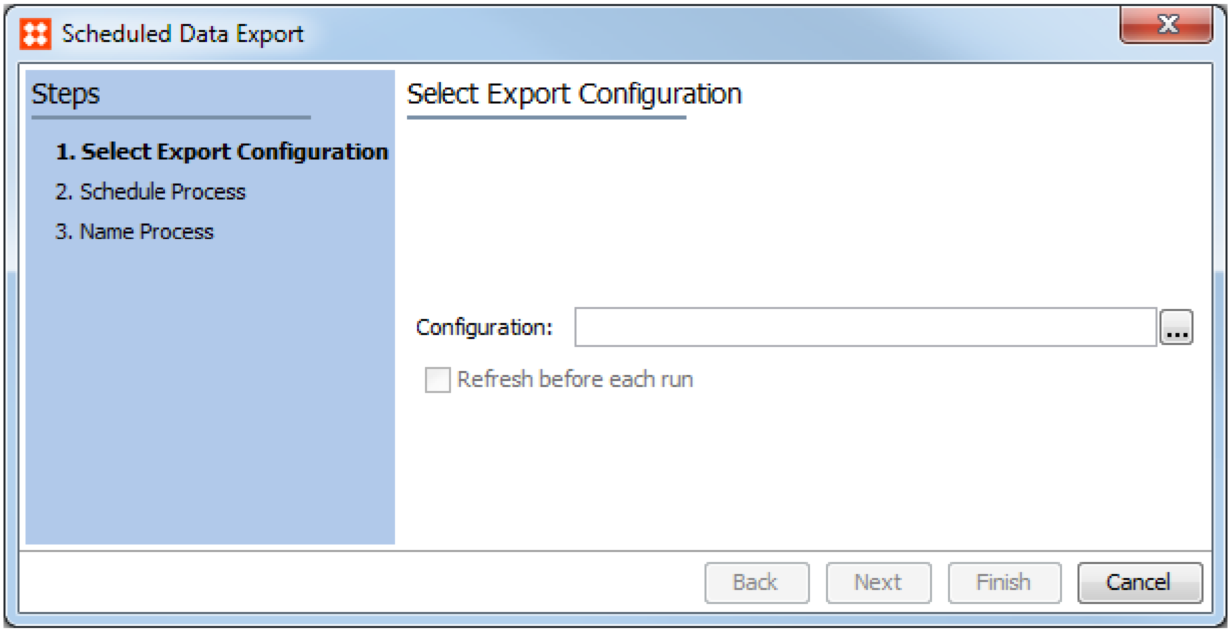
- In the Tree, select an existing export configuration, right-click, and select **Schedule Data Export**.



3. The Schedule Data Export wizard 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.

Step 1 - Select Export Configuration


Scheduled Data Export

Steps

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

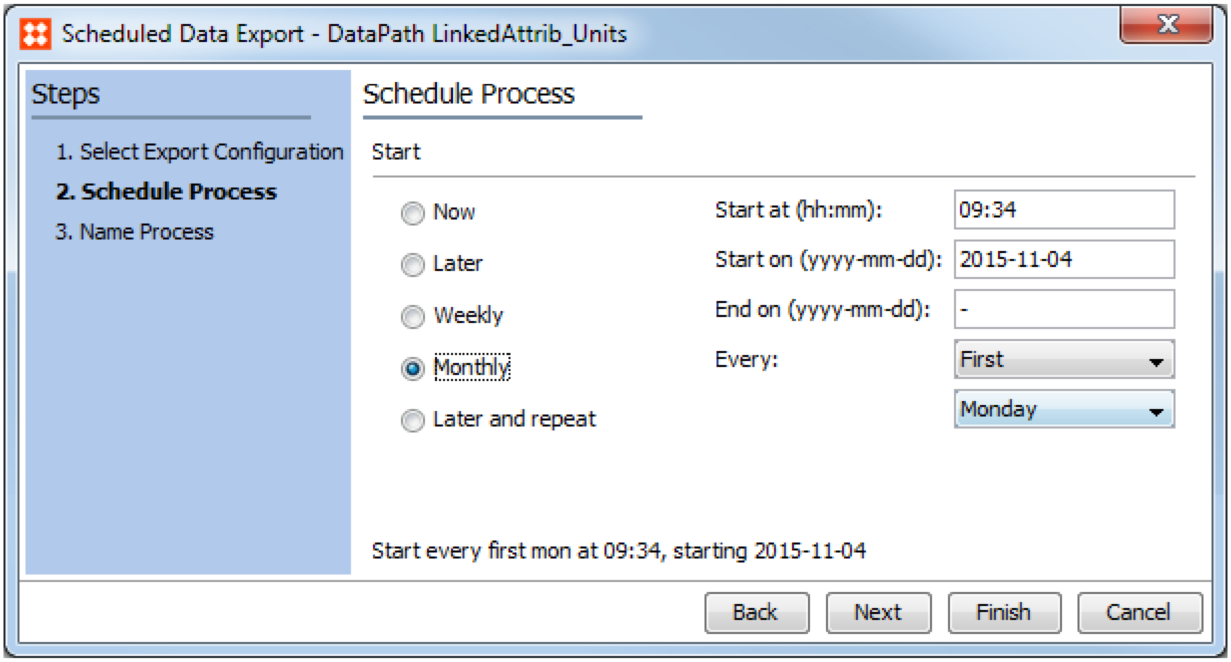
Select Export Configuration

Configuration:

□ Refresh before each run

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.
4. Click **Next**.

Step 2 - 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

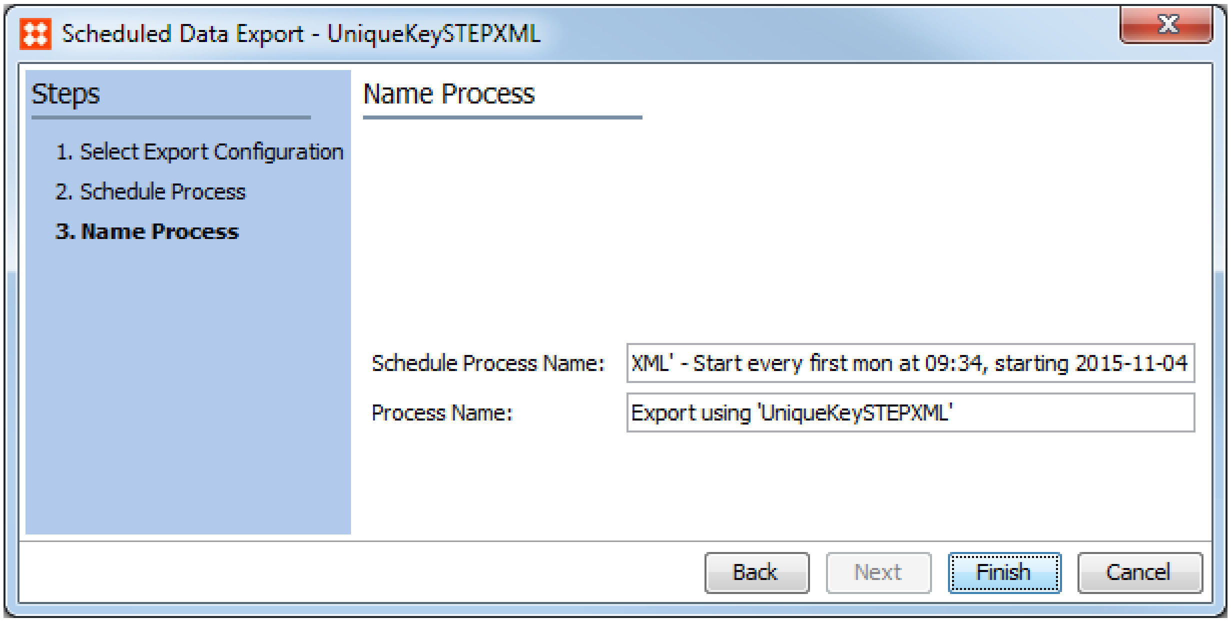
Start every first mon at 09:34, starting 2015-11-04

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 Scheduled Processes**.

2. Click **Next**.

Step 3 - Name Process



Schedule Process Name: XML' - Start every first mon at 09:34, starting 2015-11-04

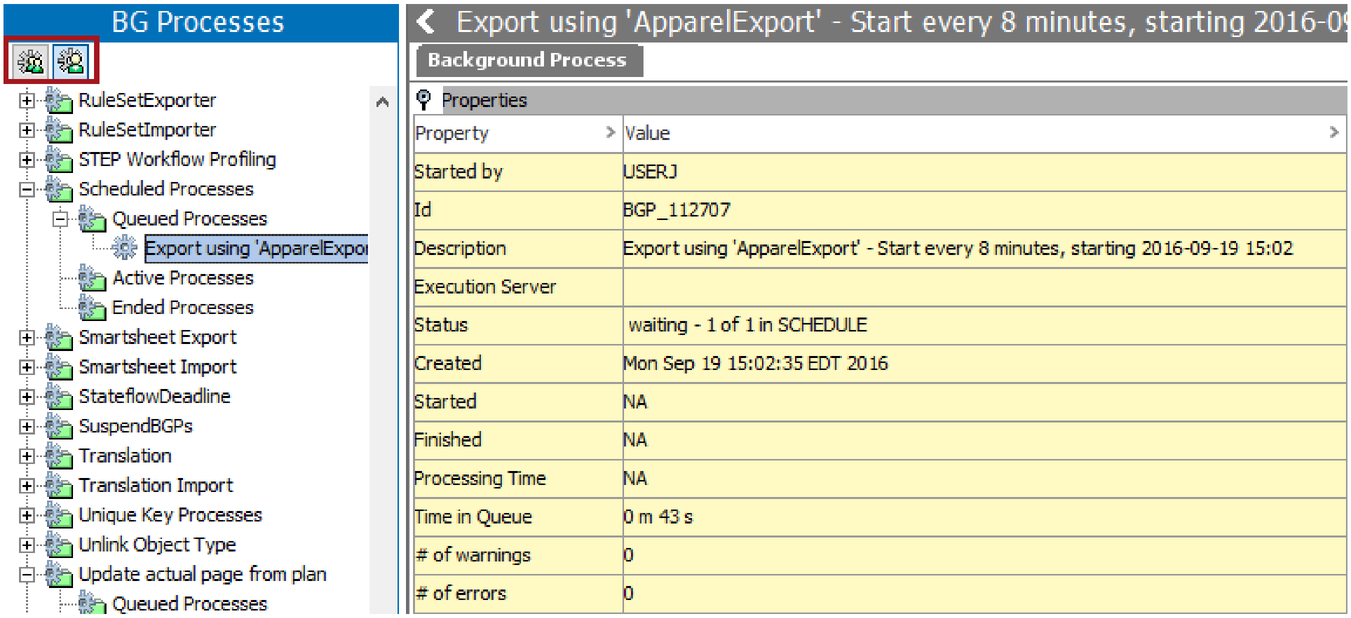
Process Name: Export using 'UniqueKeySTEPXML'

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**.

Deleting Scheduled Processes

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.

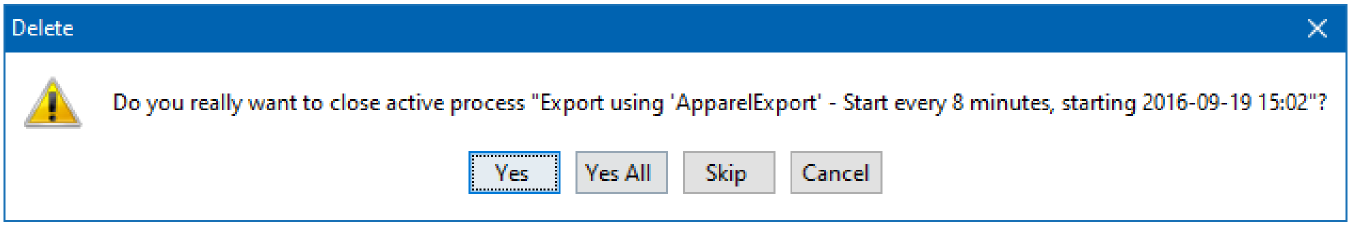


Export using 'ApparelExport' - Start every 8 minutes, starting 2016-0

Property	Value
Started by	USERJ
Id	BGP_112707
Description	Export using 'ApparelExport' - Start every 8 minutes, starting 2016-09-19 15:02
Execution Server	
Status	waiting - 1 of 1 in SCHEDULE
Created	Mon Sep 19 15:02:35 EDT 2016
Started	NA
Finished	NA
Processing Time	NA
Time in Queue	0 m 43 s
# of warnings	0
# of errors	0

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.



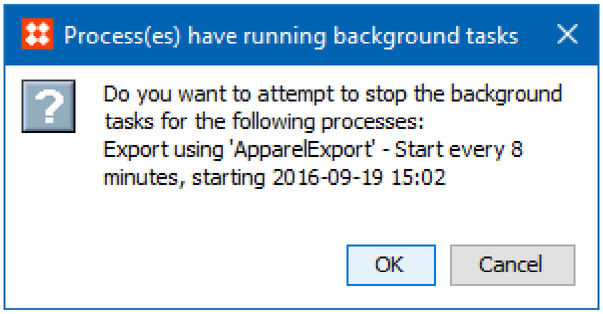
Do you really want to close active process "Export using 'ApparelExport' - Start every 8 minutes, starting 2016-09-19 15:02"?

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.



Process(es) have running background tasks

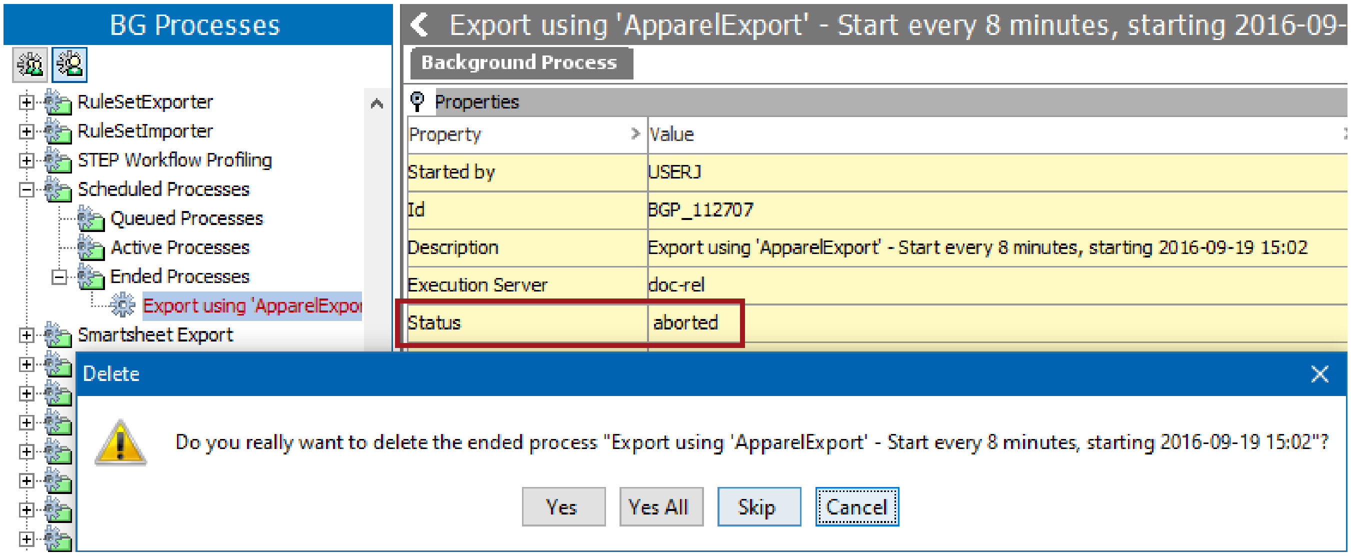
Do you want to attempt to stop the background tasks for the following processes:
Export using 'ApparelExport' - Start every 8 minutes, starting 2016-09-19 15:02

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.



Export using 'ApparelExport' - Start every 8 minutes, starting 2016-09-

Property	Value
Started by	USERJ
Id	BGP_112707
Description	Export using 'ApparelExport' - Start every 8 minutes, starting 2016-09-19 15:02
Execution Server	doc-rel
Status	aborted

Delete

Do you really want to delete the ended process "Export using 'ApparelExport' - Start every 8 minutes, starting 2016-09-19 15:02"?