STIBO SYSTEMS
MASTER DATA MANAGEMENT

SOLUTION ENABLEMENT

Data Management ECLASS Advanced

2025.4 - December 2025

Table of Contents

ECLASS: European Classification for Advanced e-Commerce

ECLASS data standard plays a crucial role in developing an international data standard that provides unique and consistent descriptions for products, materials, systems, and services in the e-commerce industry. This document aims to provide an overview of ECLASS and handling of ECLASS data standard in the STEP system.

ECLASS is an industrial language designed to facilitate the exchange of standardized, machine-readable product data in a consistent manner. ECLASS standard ensures that information provided across different platforms and systems is interoperable and easily understood. It facilitates seamless communication, exchange, and integration of product data throughout the supply chain. By adopting this standard, organizations can eliminate manual data cleansing, transformation, and other time-consuming tasks. ECLASS simplifies the process of data exchange, allowing organizations to seamlessly share information with their partners, customers, and other stakeholders.

ECLASS Basic and ECLASS Advanced are two variations of the ECLASS standard, each designed to cater specific needs and applications. Below are the key differences between the two:

- ECLASS Basic offers a simple and flat classification hierarchy, making it easy to understand and implement. It supports data exchange through CSV and XML formats, ensuring machine readability. ECLASS Basic is primarily used in eProcurement applications, enabling organizations to efficiently manage their procurement processes.
- On the other hand, ECLASS Advanced features a more complex classification hierarchy. It utilizes XML format for data exchange, allowing for greater flexibility and customization. Implementing ECLASS Advanced requires more technical expertise due to its complexity. This variation is specifically tailored for applications in engineering, IoT (Internet of Things), and digital twin technologies. It excels in handling technically complex products, providing detailed and comprehensive descriptions.

STEP supports both, ECLASS basic and ECLASS Advanced. By leveraging the ECLASS standard, STEP empowers organizations to achieve greater accuracy, efficiency, and consistency in managing and exchanging product data across various platforms and systems.

This will be the comprehensive guide to data handling within STEP in the ECLASS Advanced standard. This document serves as a valuable resource for understanding the intricate aspects of data management within the ECLASS Advanced framework. As you navigate through the sections ahead, you will gain comprehensive insights into the various facets of handling data in accordance with the ECLASS Advanced standard.

This portion of Solution Enablement material introduces the ECLASS Advanced functionality, including the use of Easy Setup and other necessary configurations.

Click on a title in the left navigation panel to expand the topics under it, or click the main topics below:

- ECLASS Advanced Quick Start Guide
- ECLASS Advanced Reference Guide

It is recommended that users be familiarized in using STEP and the STEP Online Help topics before beginning ECLASS Advanced functions.

ECLASS Advanced Quick Start Guide

This guide introduces to the ECLASS Advanced solution available in STEP 11.2 and newer versions. It covers the necessary actions an admin must take to set up the solution, as well as providing an overview of the end user functionality that is provided with the solution after Easy Setup actions for ECLASS Advanced standard have been completed by an admin.

The ECLASS Advanced commercial license is required to use this functionality. Contact Stibo Systems to begin the process of enabling a license or licenses for your system.

> **Note:** Details on the specific data models for ECLASS Advanced standard are not covered in this documentation.

This section addresses:

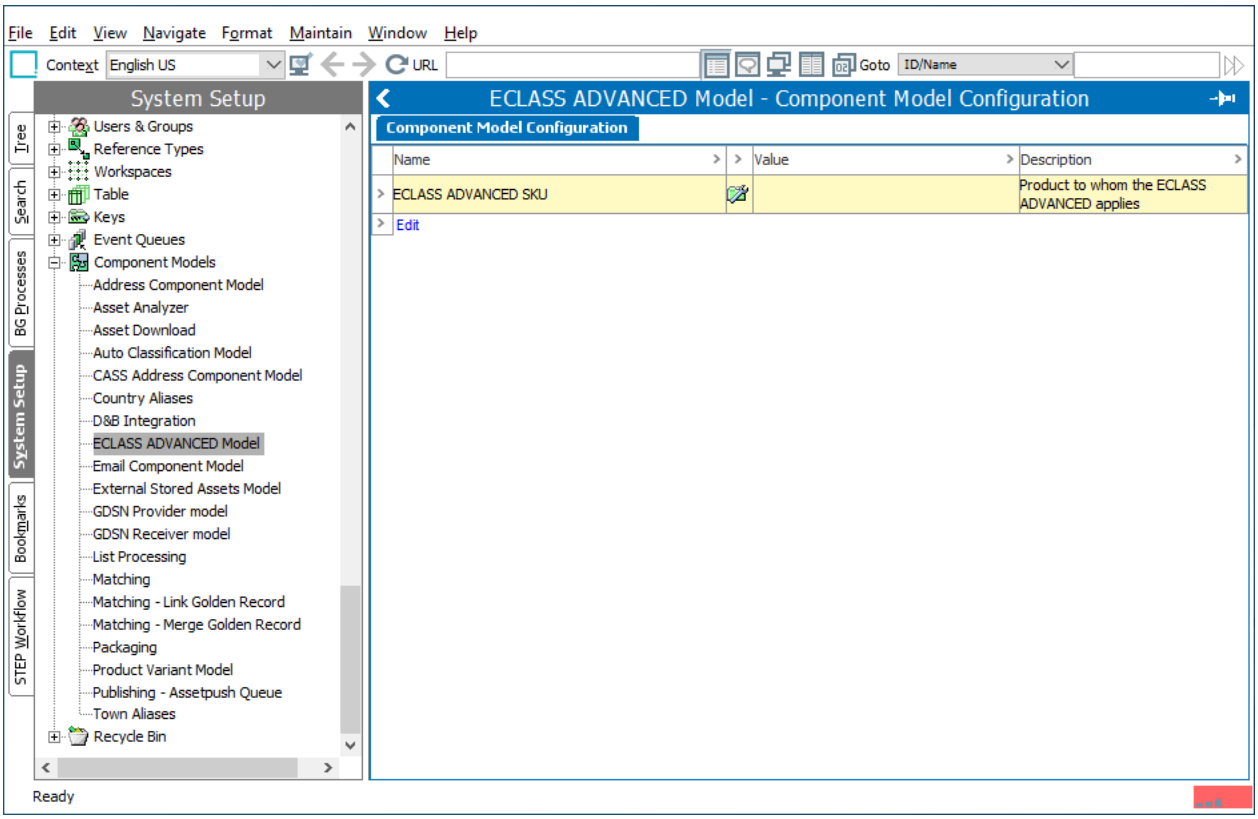
- ECLASS Advanced Data Models and Easy Setup
- ECLASS Advanced Quick Start Setup for Admins
- ECLASS Standard Supported Versions and Formats

ECLASS Advanced Data Models and Easy Setup

The ECLASS Advanced solution requires a data model for setting up the standard. When Easy Setup actions outlined in this documentation are completed, many of the configurations for the ECLASS Advanced solution are automatically configured within the 'ECLASS ADVANCED Model' that is available within the workbench > System Setup > Component Models.

The Easy Setup actions are not designed to provide a complete solution but rather give admins a jump start on completing many of the necessary configurations.

Below is a sample of the 'ECLASS ADVANCED Model' that is to be configured for ECLASS standard.



ECLASS Advanced Quick Start Setup for Admins

This section addresses the necessary actions an admin must take to set up the ECLASS Advanced solution.

Prerequisites

It is assumed that the admin has knowledge of STEP administrative functions and experience working in System Setup, including creating and editing workflows, business rules, Web UIs, attributes, etc. Therefore, this documentation does not provide introductory material for these concepts and instead targets only the specific information needed for a knowledgeable STEP admin to complete the ECLASS Advanced solution setup. If additional information is needed, refer to the ECLASS Format topic.

Quick Start Setup Actions

Below are the required setup actions:

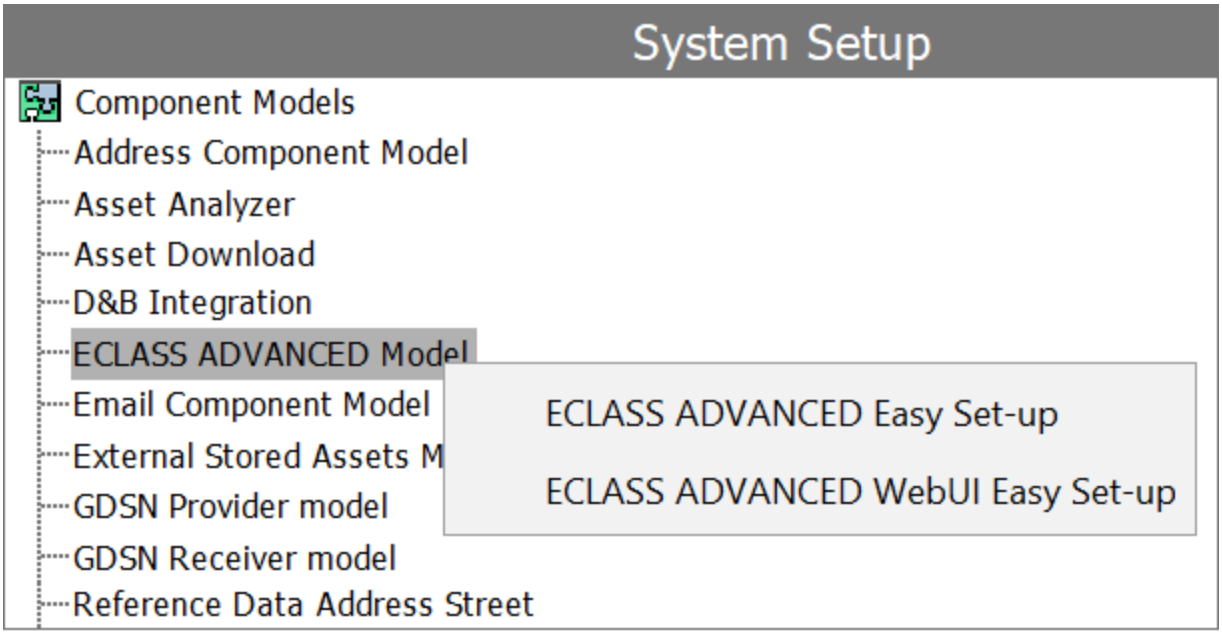
1. Run Easy Setup of ECLASS ADVANCED Industry Standard
2. Update IIEPs
3. Prepare the Language Dimension Mapping

> **Important:** The setup actions must be performed in the order in which they are listed. All steps are considered required for the setup, unless explicitly stated otherwise.

Run Easy Setup of ECLASS Advanced Industry Standard

The ECLASS Advanced industry standard includes an Easy Setup wizard that creates elements necessary to support the standard's core functionality. The process involves following two actions that must be run sequentially:

1. ECLASS ADVANCED Easy Setup
2. ECLASS ADVANCED Web UI Easy Setup



Note: If an object already exists on a system, running Easy Setup will not change it. Therefore, users can run Easy Setup as needed to deploy new functionality, without risk of disrupting or changing current processes. If manual changes have been made to an object following creation by Easy Setup, these changes are retained if setup is subsequently re-run. This also means that when enablement of new functionality requires changing the setup of an existing object, that change must be made manually on existing implementations (while Easy Setup can handle it automatically in new implementations). Because of this, it is important to pay attention to information included in patch notes and carry out any manual configurations needed to enable new functionality.

A brief description is provided below about what each of the two setup actions creates. For a detailed information on the elements created by the Easy Setup action, refer to Elements Created by Easy Setup Action topic in the **ECLASS Advanced Quick Start Setup for Admins** section of the **ECLASS Advanced** documentation.

ECLASS ADVANCED Easy Set-up: This step of the Easy Setup action creates the elements needed to support the data model consisting of object types, hierarchies, attributes, LOVs, and references.

ECLASS ADVANCED Web UI Easy Set-up: During the execution of this setup action, it's necessary to choose an existing Web UI where the widgets will be created and configured. Users have the option to either create a new Web UI and then run the Easy Setup on it, or utilize an existing one to implement the modifications. Running this setup action on an existing Web UI won't alter the settings or configurations that are already established within the Web UI.

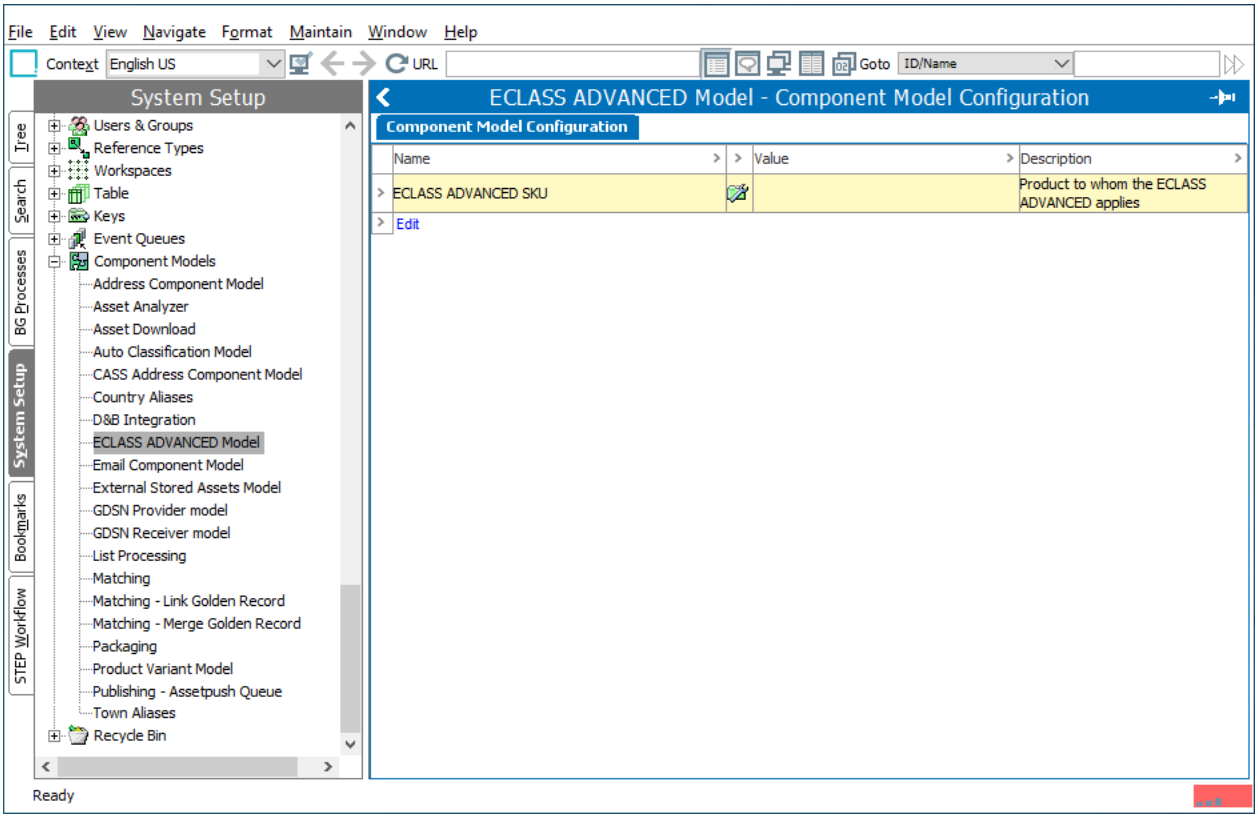
Configuration Steps

The following steps describe how to configure ECLASS Advanced industry standard using the Easy Setup method.

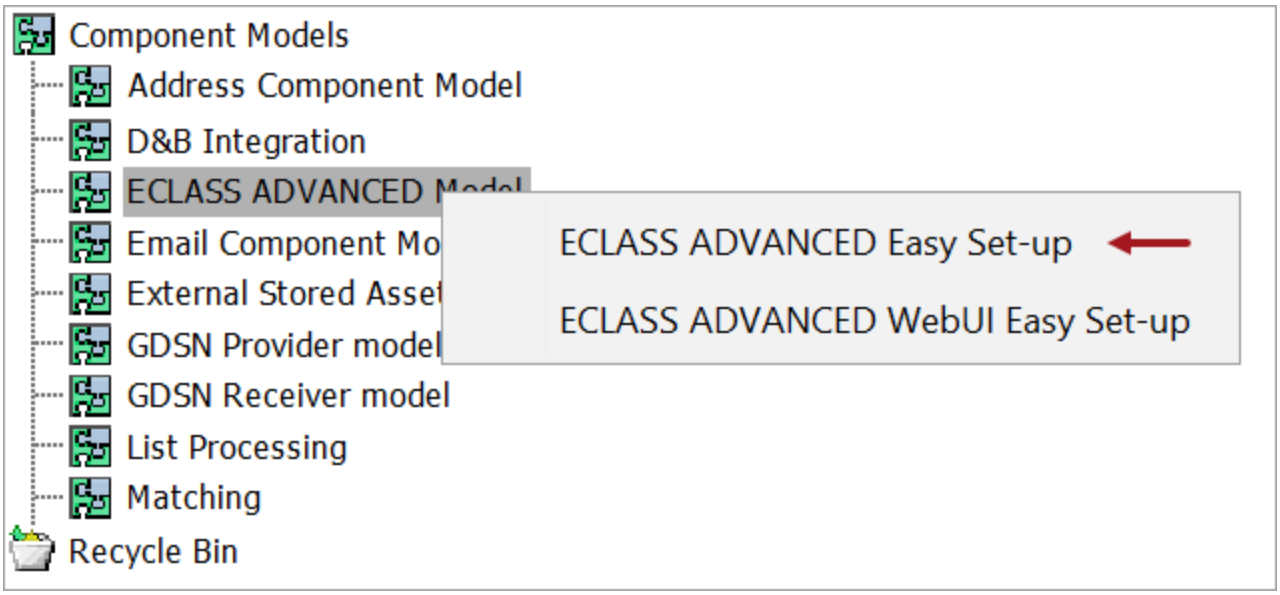
1. Go to **Context** and select the relevant context option from the dropdown.

> **Note:** Consider your relevant STEP Context before you trigger the Easy Setup. Pay attention to the language-dependent data such as attribute names and others. If you perform the Easy Setup in a second attempt using a different STEP context, the imported data from the first attempt will not be updated.

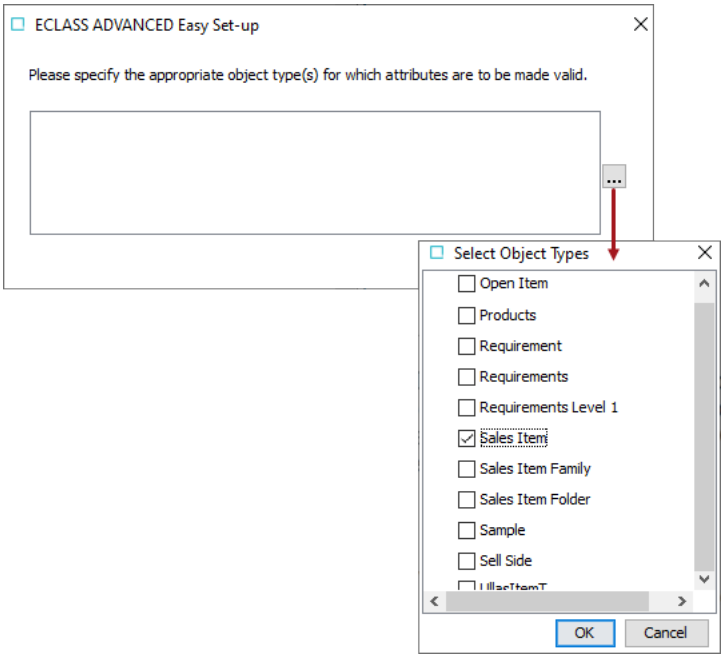
2. Go to **System Setup** > **Component Models** > **ECLASS ADVANCED Model**



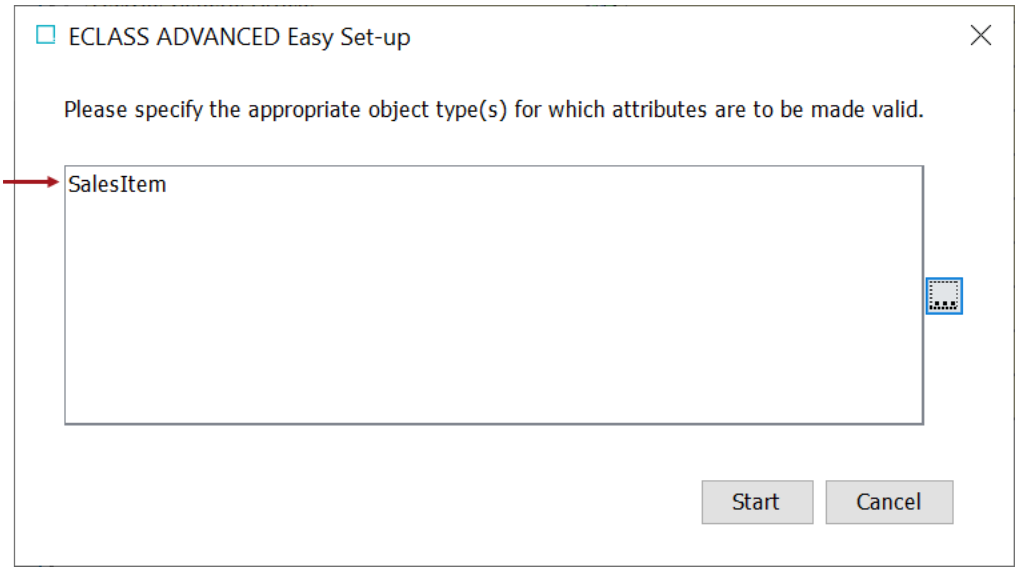
3. Right-click on the **ECLASS ADVANCED Model** and select **ECLASS ADVANCED Easy Set-up**.



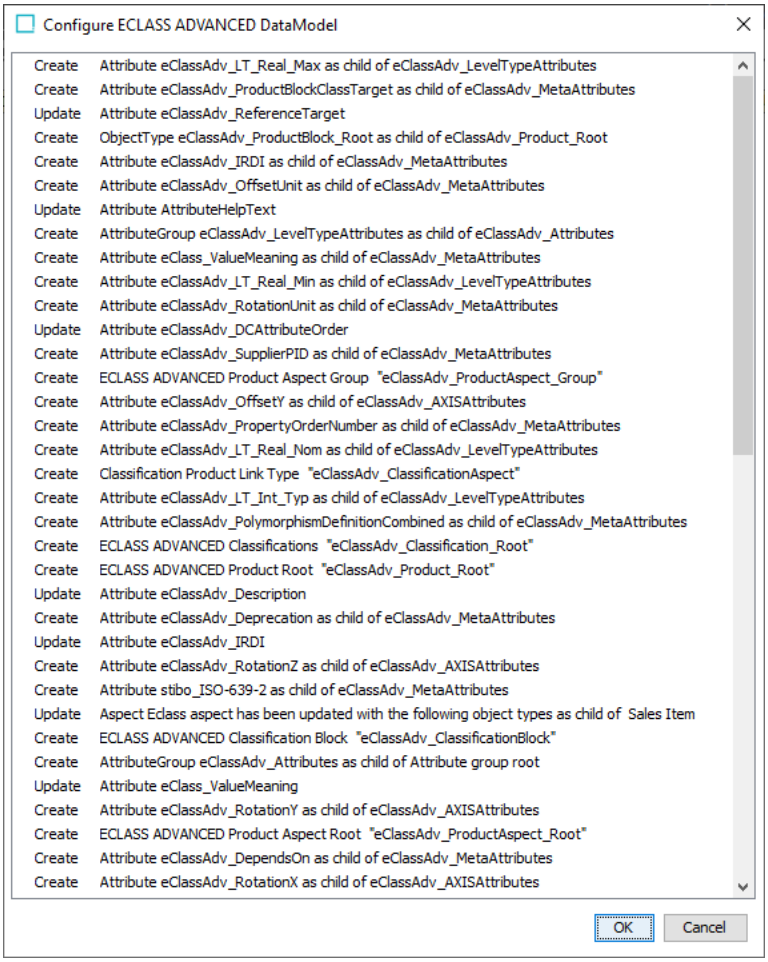
4. The ECLASS ADVANCED Easy Set-up dialog will display prompting the users to select the object type(s) for which the attributes are to be made valid. To select an object type, click the ellipsis button (...) that is available within the dialog to display the Select Object Types dialog (as shown below). It is also possible to select multiple object types.



5. Select the required object types and then click OK. All the selected object types will be listed within the ECLASS ADVANCED Easy Set-up dialog.

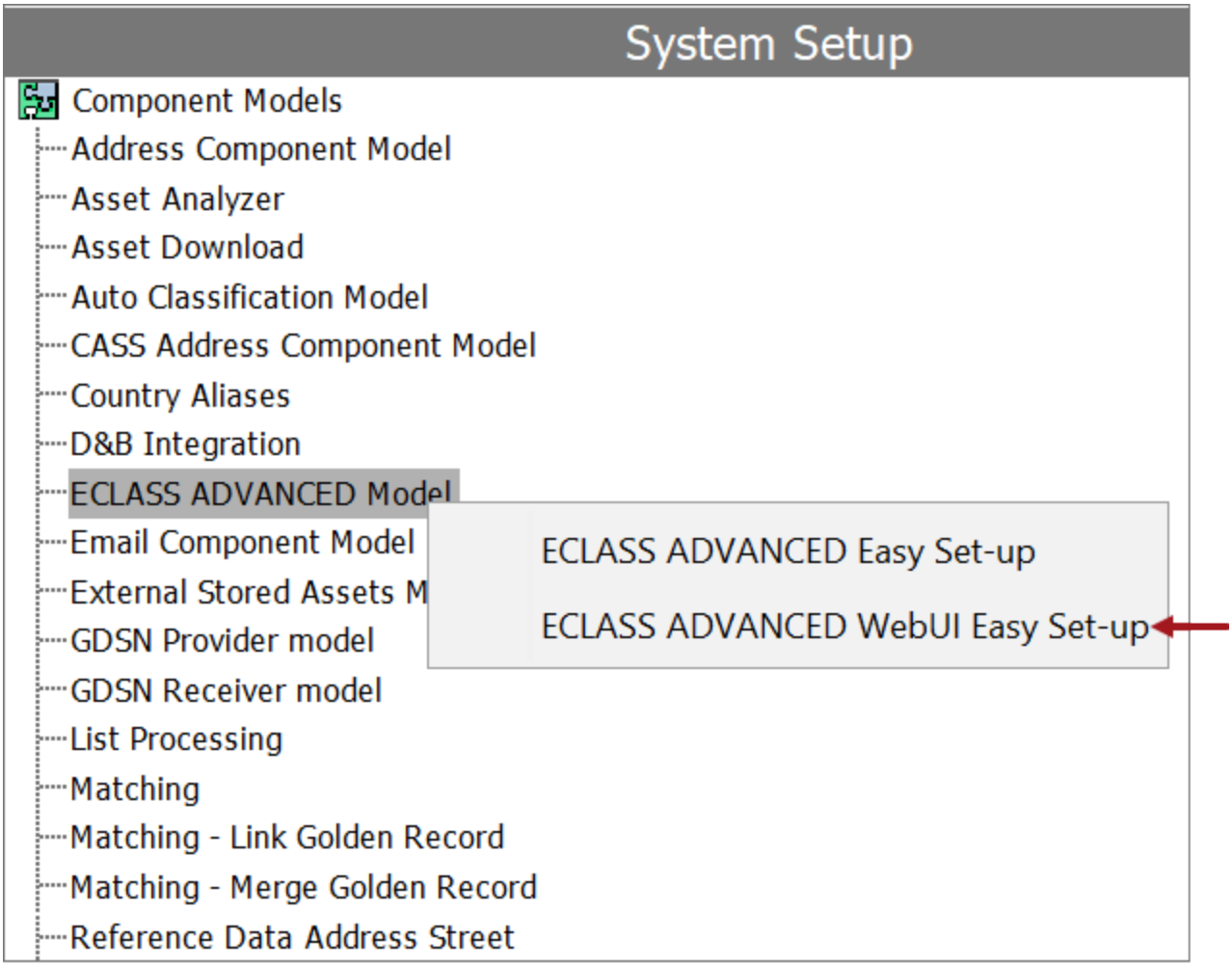


6. When all the required object types are included, click the **Start** button. A dialog will display stating the changes that will be made by running the process. If you would like to record the changes, you may do so by taking a screenshot of the dialog.

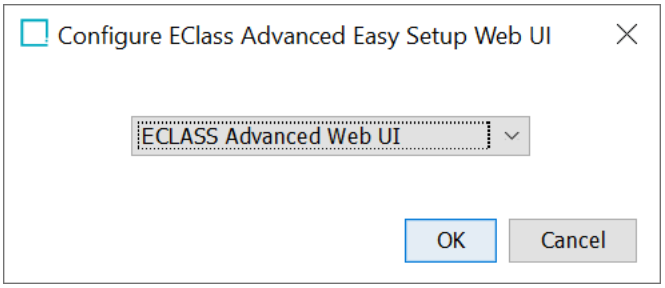


7. When you are ready to start the configuration process, click the **OK** button. The system will create all necessary elements to support the applicable process. This will typically take less than a minute, and when complete, a dialog will display listing each change that was made.

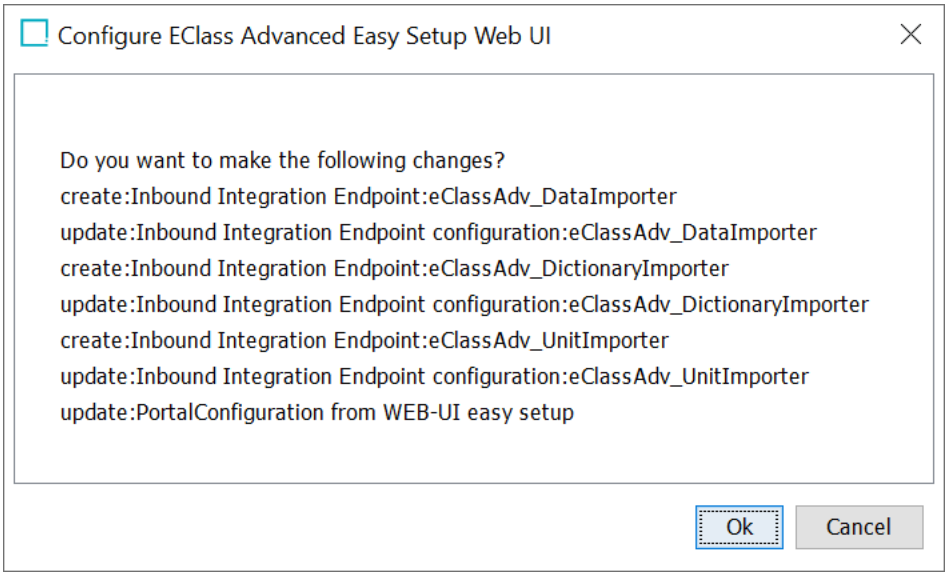
8. Right-click on the **ECLASS ADVANCED Model** and select **ECLASS ADVANCED WebUI Easy Set-up**.



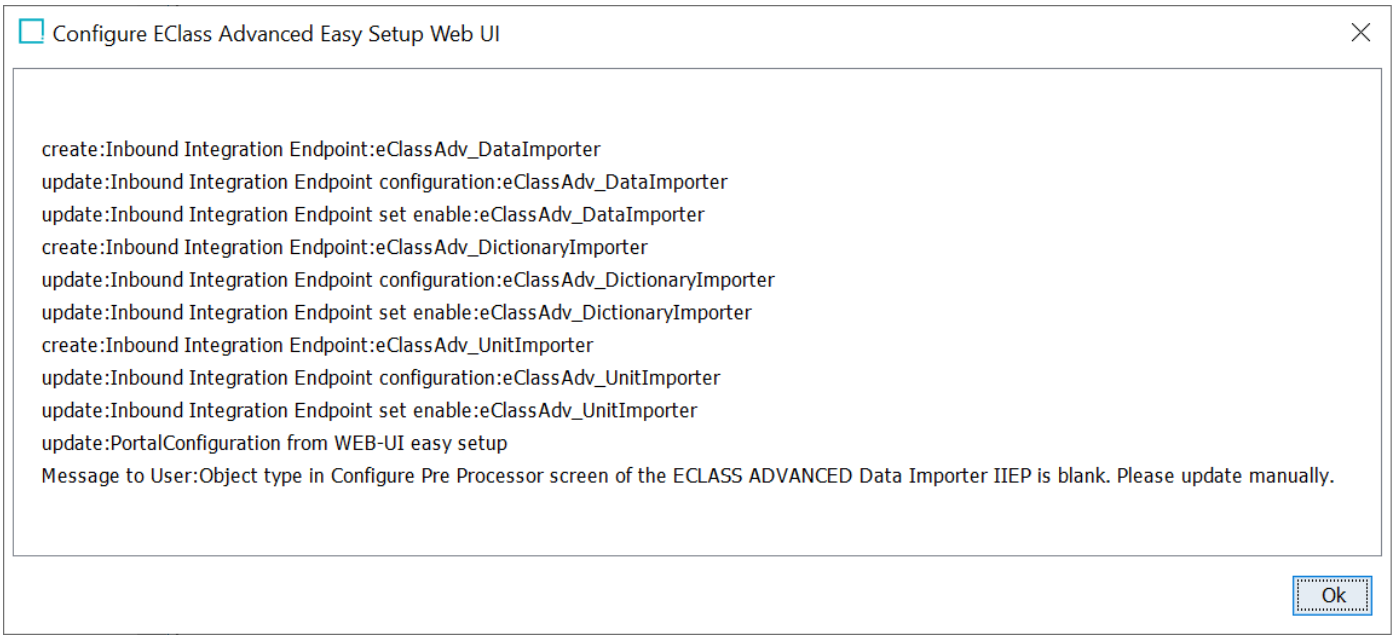
9. The 'Configure ECLASS Advanced Easy Setup Web UI' dialog will display asking the users to select a Web UI for which the changes have to be implemented. Click the dropdown available within the dialog to find and select a Web UI. In the example below, a Web UI named 'ECLASS Advanced Web UI' is selected.



10. When the required Web UI is selected, click the **OK** button. A dialog will display stating the changes that will be made by running the process. If you would like to record the changes, you may do so by taking a screenshot of the dialog.

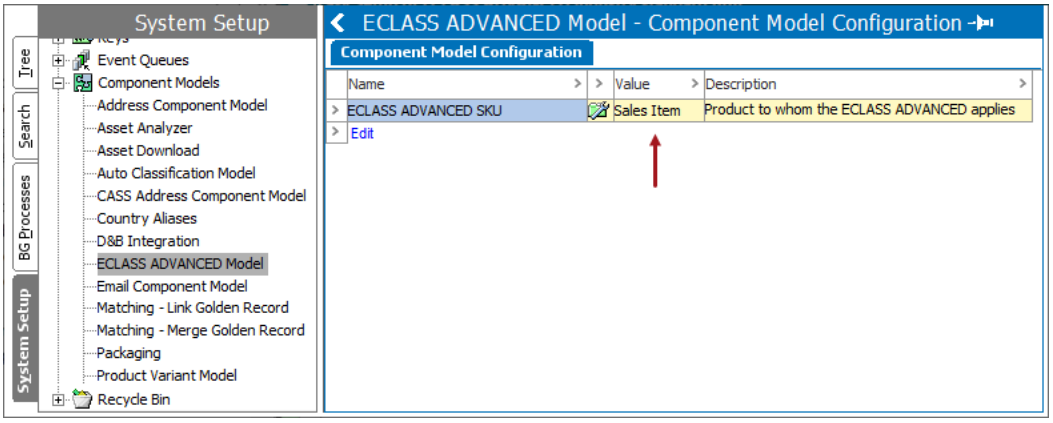


11. When you are ready to start the configuration process, click the **OK** button. The system will create all necessary elements to support the applicable process. This will typically take less than a minute, and when complete, a dialog will display listing each change that was made.



12. Click the **OK** button to close the dialog and resume normal activities on the system.

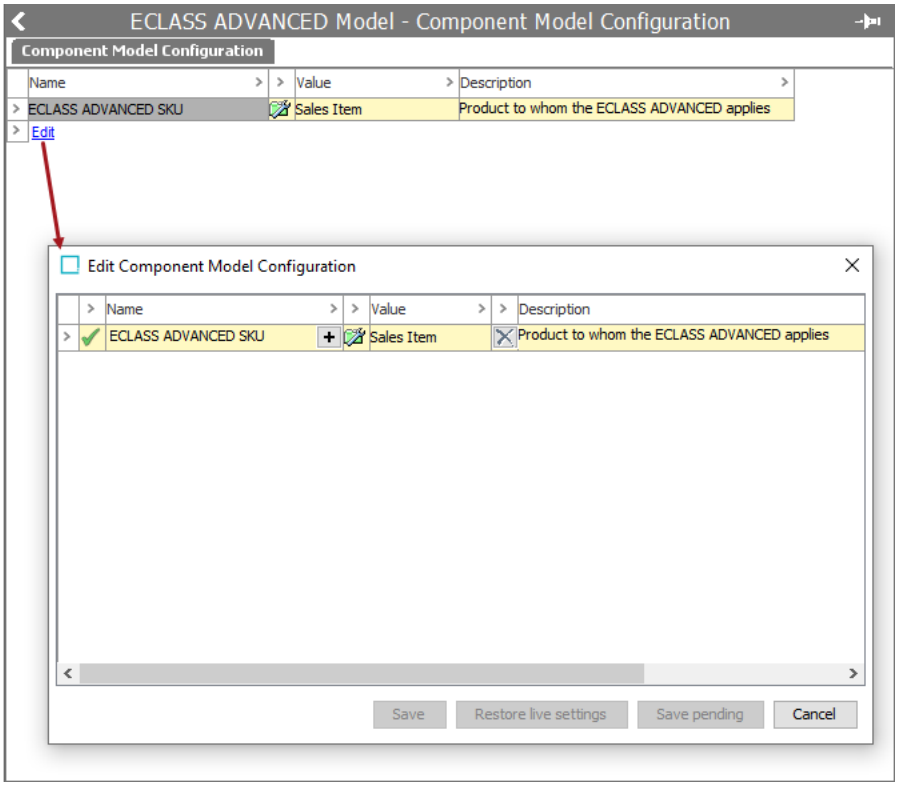
The selected object types that are mapped to the ECLASS ADVANCED Component Model will be listed within the Value column of the Component Model Configuration window.



Add / Remove object types from the ECLASS ADVANCED Component Model

An alternate way to add object types to the ECLASS ADVANCED Component Model is by accessing the Component Model Configuration window. Follow the steps below to add a new object type or to remove an object type from the component model:

- Click the blue Edit link shown at the bottom of the table. This will open the editor, allowing you to add, edit, and remove object type mappings.



2. When editing the mappings, double-click the + button to make value additions and the X button to remove any existing values. If the + button is inactive, then the value must be removed before trying to add a new one.

Elements Created by Easy Setup Action

This topic explains the elements that are created by the Easy Setup actions available within the 'ECLASS ADVANCED Model' component model. Running the ' ECLASS ADVANCED Easy Set-up' and 'ECLASS ADVANCED Web UI Easy Set-up' actions of the 'ECLASS ADVANCED Model' component model automatically creates and configures the following.

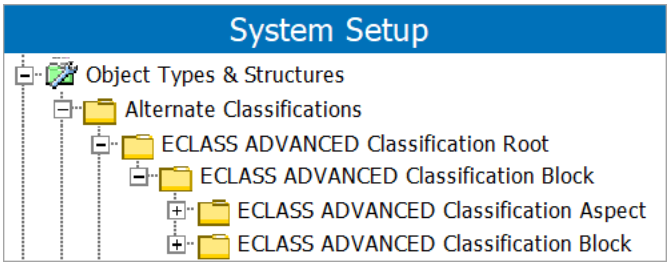
Users:

The following users are created:

USER Name	USER ID
ECLASS ADVANCED UNIT IMPORTER	ECLASSADVANCEDUNITIMPORTER
ECLASSADVANCEDDICTIONARYIMPORTER	ECLASSADVANCEDDICTIONARYIMPORTER
ECLASSADVANCEDDATAIMPORTER	ECLASSADVANCEDDATAIMPORTER

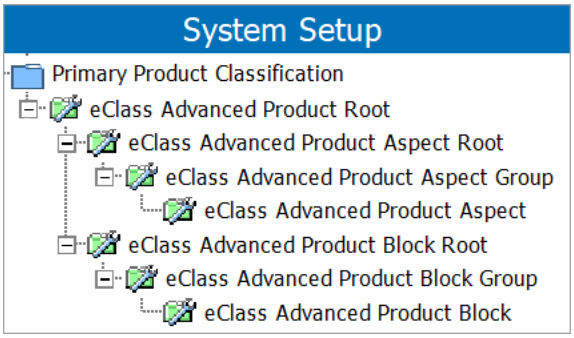
Non-Version dependent Object Type Model:

The following Classification Object Types are created:



Object Type Name	Object Type ID
ECLASS ADVANCED Classification Root	eClassAdv_Classification_Root
ECLASS ADVANCED Classification Block	eClassAdv_ClassificationBlock
ECLASS ADVANCED Classification Aspect	eClassAdv_ClassificationAspect

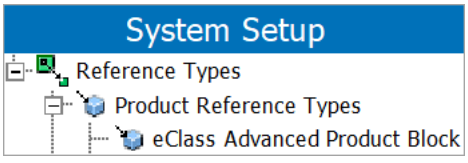
The following Product Object Types are created:



Object Type Name	Object Type ID
eClass Advanced Product Root	eClassAdv_Product_Root
eClass Advanced Product Block Root	eClassAdv_ProductBlock_Root
eClass Advanced Product Block Group	eClassAdv_ProductBlock_Group
eClass Advanced Product Block	eClassAdv_ProductBlock
eClass Advanced Product Aspect Root	eClassAdv_ProductAspect_Root
eClass Advanced Product Aspect Group	eClassAdv_ProductAspect_Group
eClass Advanced Product Aspect	eClassAdv_ProductAspect

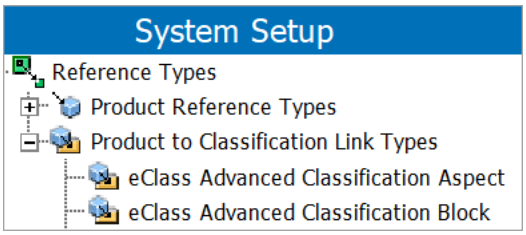
References between the relevant objects:

The following Product Reference Types are created:



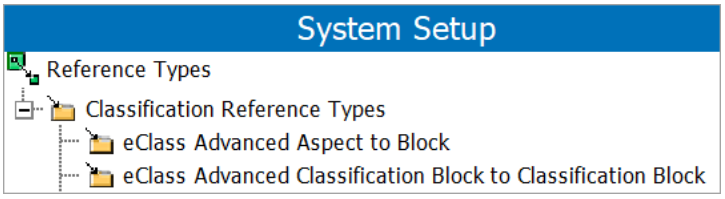
Reference Type Name	Reference Type ID	Validity From (ID)	Validity To (ID)
eClass Advanced Product Block	eClassAdv_ProductBlock	eClassAdv_ ProductBlock	eClassAdv_ ProductBlock

The following Product to Classification Link Types are created:



Reference Type Name	Reference Type ID	Validity From (ID)	Validity To (ID)
eClass Advanced Classification Block	eClassAdv_ ClassificationBlock	eClassAdv_ ProductBlock	eClassAdv_ ClassificationBlock
eClass Advanced Classification Aspect	eClassAdv_ ClassificationAspect	eClassAdv_ ProductAspect	eClassAdv_ ClassificationAspect

The following Classification Reference Types are created:

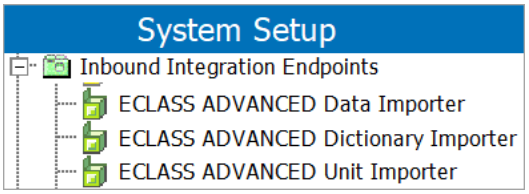


Reference Type Name	Reference Type ID	Validity From (ID)	Validity To (ID)
eClass Advanced	eClassAdv_	eClassAdv_	eClassAdv_

Reference Type Name	Reference Type ID	Validity From (ID)	Validity To (ID)
Aspect to Block	ClassAspectToClassBlock	ClassificationAspect	ClassificationBlock
eClass Advanced Classification Block to Classification Block	eClassAdv_ ClassBlockToClassBlock	eClassAdv_ ClassificationBlock	eClassAdv_ ClassificationBlock

Inbound Integration Endpoints:

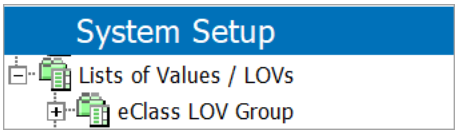
The following Inbound Integration Endpoints are created:



Inbound Integration Endpoint Name	Inbound Integration Endpoint ID
ECLASS ADVANCED Dictionary Importer	eClassAdv_DictionaryImporter
ECLASS ADVANCED Unit Importer	eClassAdv_UnitImporter
ECLASS ADVANCED Data Importer	eClassAdv_DataImporter

LOV Group Creation:

The following LOV group is created:



LOV Group Name	LOV Group ID
eClass LOV Group	eClassLOVGroup

Relevant Meta Data Attributes:

The following Meta Data Attributes are created:

Attribute Name	Attribute ID	Description
eCl@ss Description	EclassDescription	This is a Stibo Systems created meta data attribute holding the version of the imported ECLASS data
eCl@ss ID	EclassID	
eCl@ss Import Version	EclassImportVersion	This is a Stibo Systems created meta data attribute holding the version of the imported ECLASS data For ECLASS Advanced only maintained on the Unit.
eClass Advanced Attribute Type	eClassAdv_AttributeType	This is a Stibo Systems created meta data attribute holding the Type of the property based on the ECLASS Advanced definition
eClass Advanced Cardinality	eClassAdv_Cardinality	This is a Stibo Systems created meta data attribute holding flag if a block is per definition a cardinality block
eClass Advanced Data Container Attribute Order	eClassAdv_DCAttributeOrder	This is a Stibo Systems created Meta Data determine the order of the attribute how it is shown in the Web UI within the Data Container (Level

Attribute Name	Attribute ID	Description
		Type and Axis Type)
eClass Advanced Data Container Link Type Attribute	eClassAdv_ DataContainerLinkTypeAttribute	Determines which Attribute is used to hold the Meta Information for the Data Container Link. Is used for Level Type and Axis Type Data Containers.
eClass Advanced Depends On	eClassAdv_DependsOn	This is a Stibo Systems created meta data attribute holding ID of the Attribute that the current Attribute is depending on
eClass Advanced Deprecation	eClassAdv_Deprecation	This is a Stibo Systems created meta data attribute holding the information if a Classification Block or Classification Aspect is deprecated.
eClass Advanced Description	eClassAdv_Description	This is a Stibo Systems created meta data attribute holding the description of the ECLASS Advanced specific data model components
eClass Advanced IRDI	eClassAdv_IRDI	This is a Stibo Systems created meta data attribute holding the ECLASS IRDI.
eClass Advanced Product Block Classification Target	eClassAdv_ProductBlockClassTarget	This is a Stibo Systems created meta data attribute holding the Classification Block ID of the referenced Product Block
eClass Advanced	eClassAdv_PropertyDomainType	This is a Stibo Systems created meta

Attribute Name	Attribute ID	Description
Property Domain Type		data attribute holding the property domain type based on the ECLASS Advanced definition
eClass Advanced Property Order Number	eClassAdv_PropertyOrderNumber	This is a Stibo Systems created meta data attribute holding the property order number
eClass Advanced Reference Target	eClassAdv_ReferenceTarget	This is a Stibo Systems created meta data attribute holding the ID of the Block that this Attribute is referencing to
eClass Advanced Supplier PID	eClassAdv_SupplierPID	This is a Stibo Systems created Meta Data holding the original Supplier Product ID for the Product Block Group since for the STEP ID it needs to be hashed to not exceed 40 characters within the STEP ID
eClass Advanced Version	eClassAdv_Version	This is a Stibo Systems created meta data attribute displayed on the root node of an ECLASS Advanced Version displaying the version in a human readable format
Eclass Keyword	EclassKeyword	This is a Stibo Systems created meta data attribute holding the eclass description
eClass Primary Key	EclassPrimaryKey	This is a Stibo Systems created meta data attribute holding the eclass id (harmonized, but without prefix).

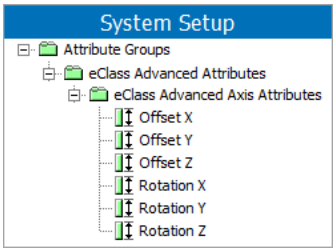
Attribute Name	Attribute ID	Description
FID	eClassAdv_FID	This is a Stibo Systems created meta data attribute holding the FID Information of the Importer that builds the hierarchy
FPARENT_ID	eClassAdv_FPARENT_ID	This is a Stibo Systems created meta data attribute holding the FPARENT_ID Information of the Importer that builds the hierarchy
ISO-639-2	stibo_ISO-639-2	This is a Stibo Systems created Meta Data holding the Language ISO Code for Language Dimension Points. Using the ISO Standard https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
Polymorphism Controlling Value	eClassAdv_ PolymorphismControllingValue	This is a Stibo Systems created meta data attribute holding the value that determines the Polymorphism of the current level
Polymorphism Defining Property	eClassAdv_ PolymorphismDefiningProperty	This is a Stibo Systems created meta data attribute holding the information of the Polymorphisms attribute of the specific Block
Polymorphism Definition Combined	eClassAdv_ PolymorphismDefinitionCombined	This is a Stibo Systems created meta data attribute holding the information of the Polymorphisms attribute of the specific Block plus the value ID that needs to be selected to get to this block. Separator is pipe (|)

Attribute Name	Attribute ID	Description
Unit of the Offset	eClassAdv_OffsetUnit	This is a Stibo Systems created meta data attribute holding the Offset unit of the Axis Type Property which is depending on the Classification where it is used
Unit of the Rotation	eClassAdv_RotationUnit	This is a Stibo Systems created meta data attribute holding the Rotation unit of the Axis Type Property which is depending on the Classification where it is used
eClass Advanced Imported Versions	eClassAdv_ImportedVersions	This is a Stibo Systems created Meta Data telling in which Versions this Classification was imported.
eClass Advanced BMEcat ECLASS Version	eClassAdv_BMEcatECLASSVersion	This is a Stibo Systems created Meta Data telling the BMEcat ECLASS Version of the current ECLASS Root Node
ECLASS Value Meaning	eClass_ValueMeaning	This is a Stibo Systems created meta data attribute that applies only to List of Values (LOV). This attribute contains the 'value meaning information' (human readable value), the “STEP value ID” and the “value string” (machine readable value), provided in the dictionary. This meta-attribute might be used to provide the user in a data maintenance scenario (Editor) with all information needed

Attribute Name	Attribute ID	Description
		to select the correct value.
eClass Advanced Product SKU ID	eClassAdv_ProductSKUID	This is a Stibo Systems Product SKU ID which is used get the SKU details in Aspect and Block.
eClass Advanced Version	eClassAdv_Version_Harmonized	This is a Stibo Systems created meta data attribute displayed on the root node of an ECLASS Advanced Product Block and Product Aspect Group. Used for easy and harmonized version identification.

AXIS Attributes:

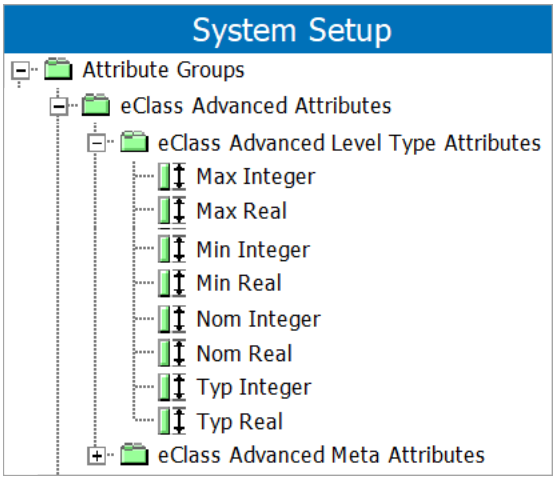
The following AXIS attributes are created:



Offset X	eClassAdv_OffsetX
Offset Y	eClassAdv_OffsetY
Offset Z	eClassAdv_OffsetZ
Rotation X	eClassAdv_RotationX
Rotation Y	eClassAdv_RotationY
Rotation Z	eClassAdv_RotationZ

Static Level Type Attributes:

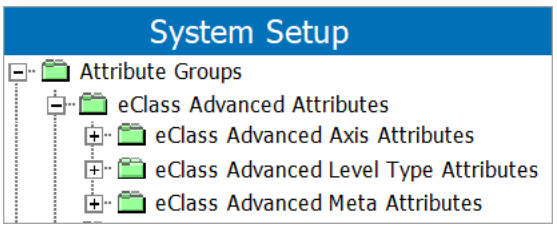
The following Static Level Type attributes are created:



Min Integer	eClassAdv_LT_Int_Min
Max Integer	eClassAdv_LT_Int_Max
Nom Integer	eClassAdv_LT_Int_Nom
Min Real	eClassAdv_LT_Real_Min
Max Real	eClassAdv_LT_Real_Max
Nom Real	eClassAdv_LT_Real_Nom
Typ Integer	eClassAdv_LT_Int_Typ
Typ Real	eClassAdv_LT_Real_Typ

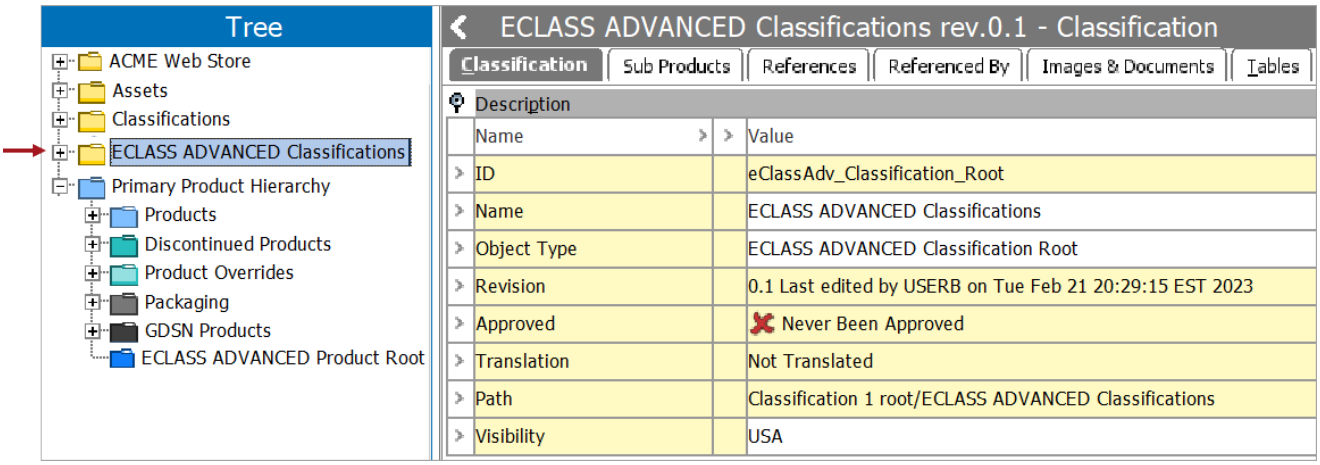
Relevant Root Nodes:

The following attribute groups are created:

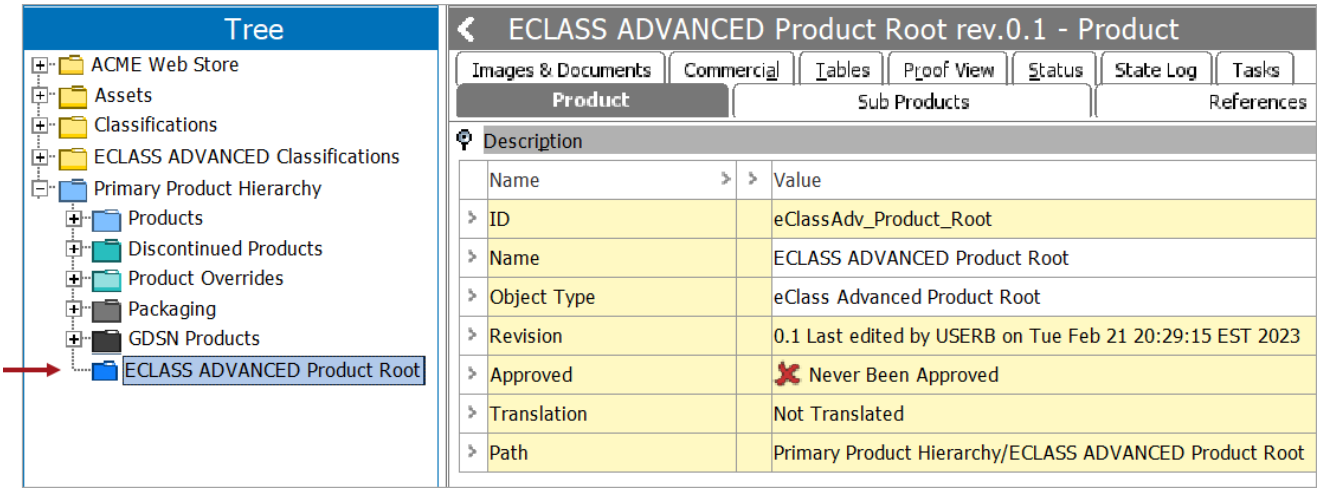


Attribute Group Name	
eClass Advanced Axis Attributes	eClassAdv_AXISAttributes
eClass Advanced Level Type Attributes	eClassAdv_LevelTypeAttributes
eClass Advanced Meta Attributes	eClassAdv_MetaAttributes

A classification folder named 'ECLASS ADVANCED Classifications' (ID = eClassAdv_Classification_Root) is created as shown below.

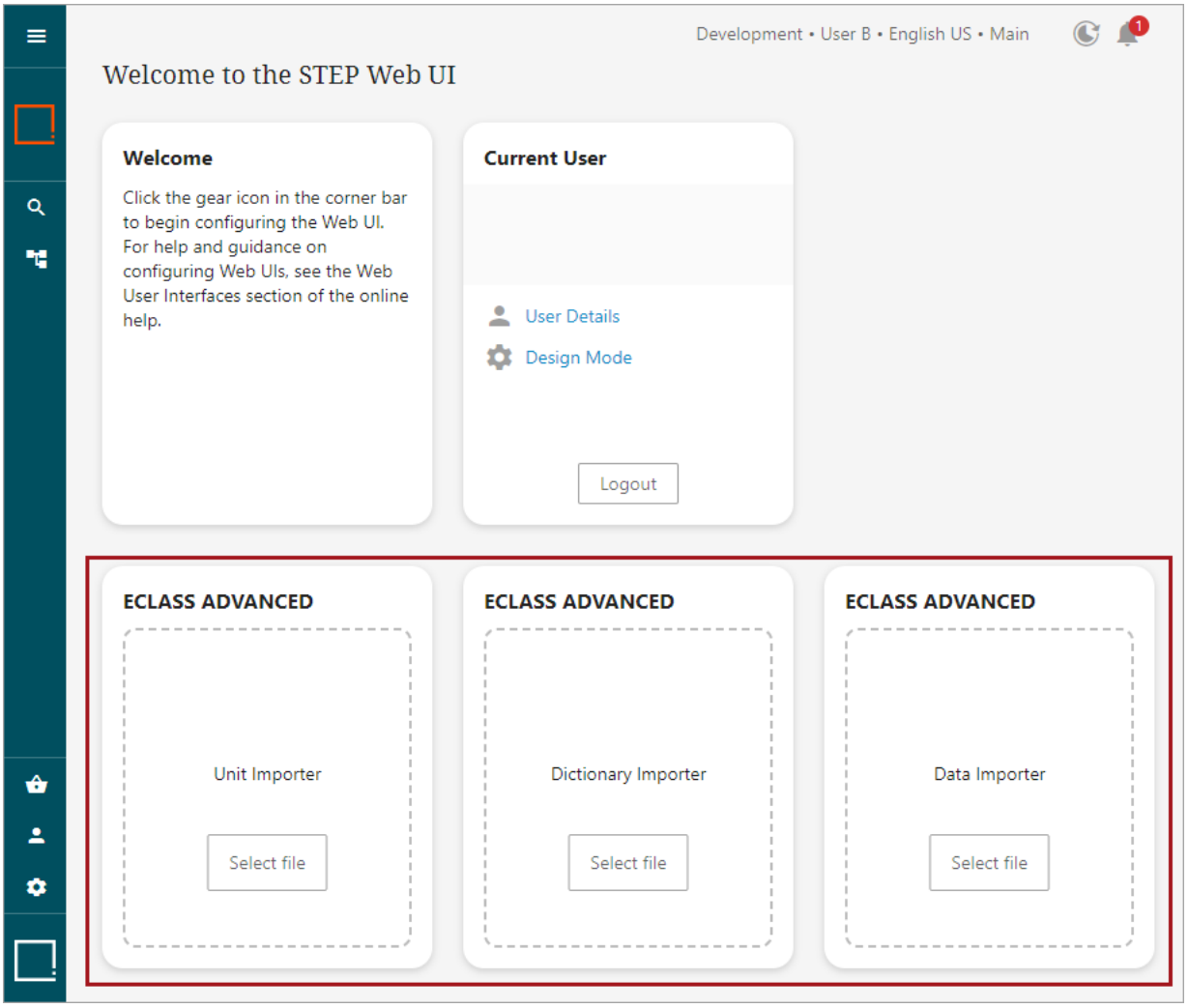


A product folder named 'ECLASS ADVANCED Product Root' (ID = eClassAdv_Product_Root) is created under the Primary Product Hierarchy folder as shown below.



File Loading Widgets in the Web UI:

The following File Loading Widgets are created in the homepage of the Web UI:



- ECLASS Advanced Unit Importer
- ECLASS Advanced Dictionary Importer
- ECLASS Advanced Data Importer

Note: It is expected that the Web UI administrator will modify the default configurations to provide access to customer-specific data and processes as needed, so your Web UI may look different from what is shown above. However, it is likely that the default configurations have been expanded upon rather than removed so your display should be comparable to what is described. If not, contact your administrator for additional information.

Easy Setup Generic Considerations

When executing the Easy Setup action, the following considerations come into play:

- If a user modifies Data Model objects, rerunning the Easy Setup will not overwrite such changes. For example, alterations to validation base type in an attribute, multivalued settings, or defining if the object is externally maintained or not, etc.
- If a customer removes Data Model objects, rerunning the Easy Setup will recreate them. This applies, for example, when deleting static or metadata attributes, the Easy Setup restores them.
- When a customer adds Data Model object(s), rerunning the Easy Setup will not eliminate these objects. The Easy Setup will never delete / erase data. For example, newly added attributes, metadata attributes, or objects remain intact.
- Once the Easy Setup is unintentionally executed, reverting changes is not possible. Manual removal of created data may become necessary in extreme cases.
- The Easy Setup action will always be executed on the logged-on user's credentials.

> **Note:** Under no circumstances shall the Stibo Systems ECLASS Advanced Data Model be changed by any user.

Prepare the Language Dimension Mapping

Prior to importing the Unit or Dictionary file into the system, it is essential to establish a relationship (commonly referred to as mapping) between the ECLASS language code and the language dimension points in the system. Familiarity with the ECLASS language codes contained within the ECLASS XML file is essential for this process.

Within the Unit file, the language code will be specified within the 'content_description' tags, as depicted below. The Unit file always comes in English and German language.

```
<content_description language_code="ENG">eCl@ss 10.1</content_
description>
```

Although the Unit file contains both English and German languages, Unit Groups and Units are exclusively created in English. Consequently, it is recommended to import Units into the Global context. Alternatively, if there is a desire to import Units in German, users should switch the tag as follows:

From:

```
<content_description language_code="ENG">ECLASS12.0</content_description>
```

To:

```
<content_description language_code="DEU">ECLASS12.0</content_description>
```

The Dictionary file, the language code may vary depending on the specific file being imported.

For the file that is in English, the language code within the 'content_description' tags will be as

```
<content_description language_code="ENG">eCl@ss 10.1 ADVANCED</content_
description>
```

For French, the language code within the 'content_description' tags will be as

```
<content_description language_code="FRA">eCl@ss 10.1 ADVANCED</content_
description>
```

For German, the language code within the 'content_description' tags will be as

```
<content_description language_code="DEU (or GER)">eCl@ss 10.1
ADVANCED</content_description>
```

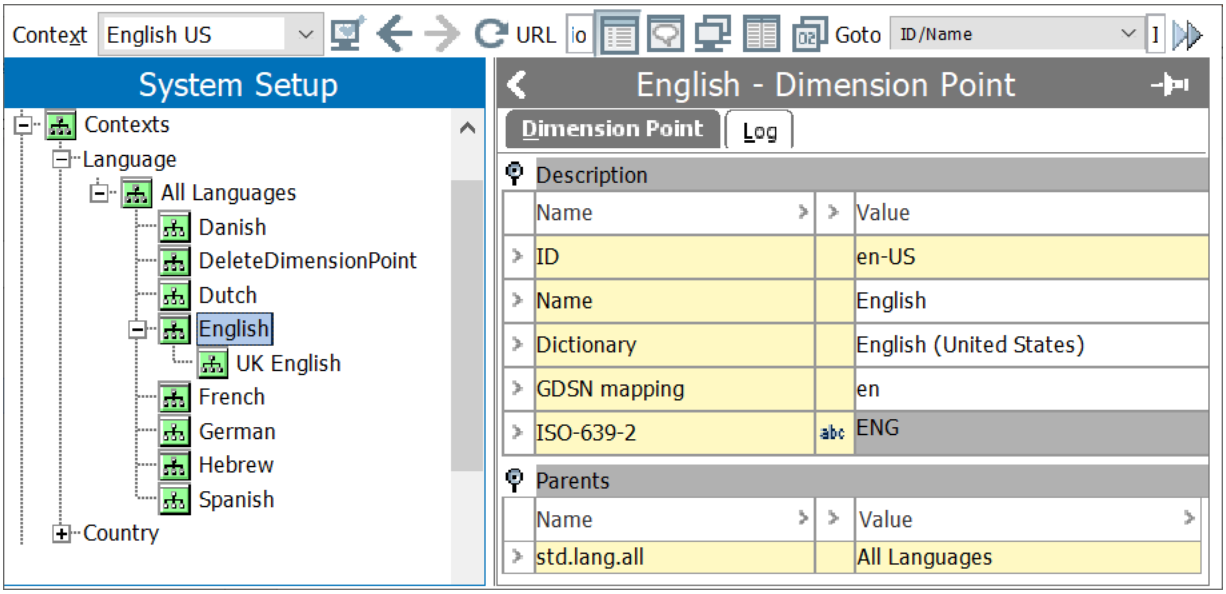
Users must be aware of the appropriate language code to be used within the 'content_description' tags based on the language of the Dictionary file they intend to import.

Context (Dimension Point) Identification

Mapping ECLASS Advanced language codes is a straightforward process. As a system administrator, you need to enter the appropriate language code value into the metadata attribute ISO-639-2 (ID = stibo_ISO-639-2) for the desired language dimension point. By following these guidelines, you can effectively identify and import language-specific data into the appropriate contexts.

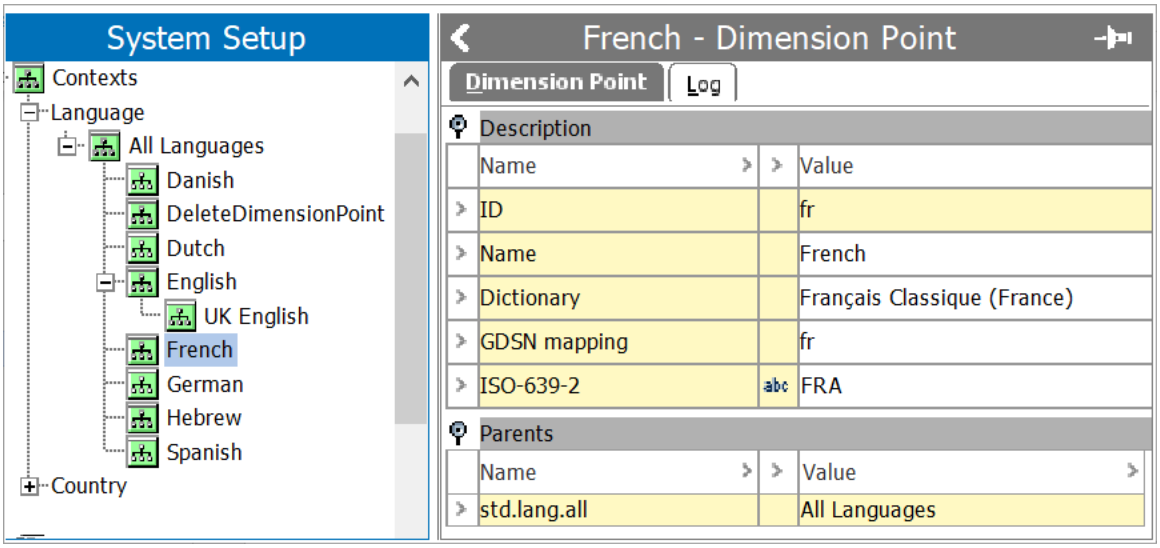
Example:

In the following example, the language code 'ENG' is applied to the English language dimension. Consequently, language-dependent data will be imported in the English language and associated with the context that holds the English dimension point.

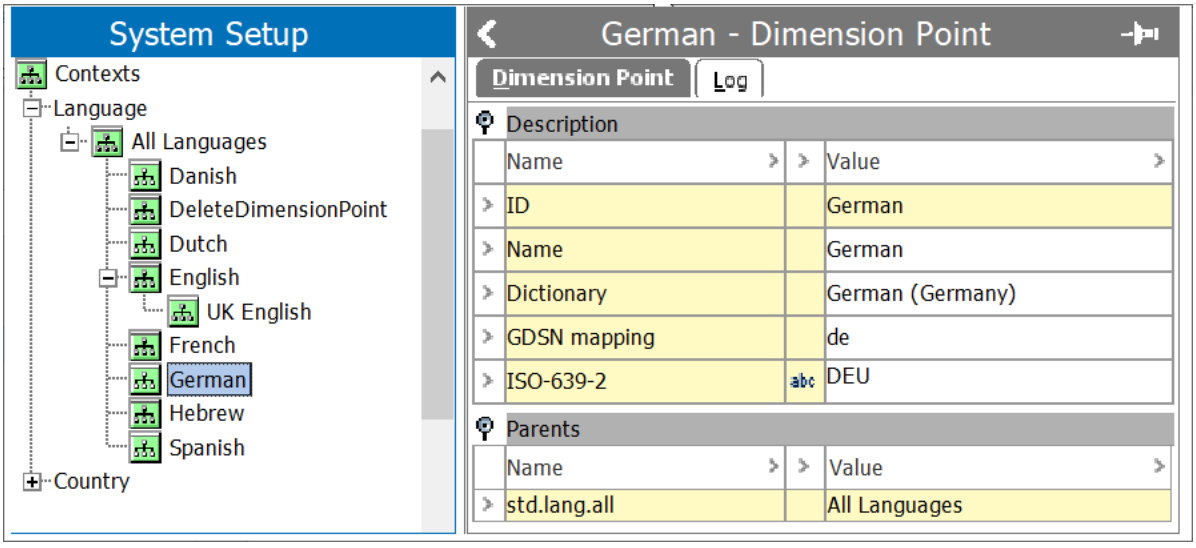


Note: As illustrated in the example picture above, selecting the All Languages option (as it is the parent of all other language dimension points) ensures that language-dependent data are inherited across all dimension points.

For ECLASS Advanced packages in the French language, the language code 'FRA' should be applied to the French language dimension point. As a result, language-dependent data will now be imported in the French language, associated with the context holding the French dimension point.



Similarly, for ECLASS Advanced packages in the German language, the language code 'DEU' should be applied to the German language dimension point. Consequently, language-dependent data will be imported in the German language, linked to the context holding the German dimension point.



Potential Scenarios During Dictionary File Import

During the import of ECLASS Dictionary files, certain scenarios may arise due to irregularities in language mapping. We have outlined the following possible situations:

- Absence of matching language dimension point:

  If the system lacks a language dimension point that corresponds to the language code provided in the file, the language-dependent data will be imported into the context specified in the Inbound Integration Endpoint. In this way, data is seamlessly imported even if an exact language match is unavailable.

- Multiple contexts with the same language dimension point:

  In cases where the same language dimension point exists in multiple contexts within the system, the import process identifies the first context containing the relevant language dimension point. For instance, if the 'Eng' language code is found in contexts named 'UK-Eng' and 'US-Eng,' the English dimension point from the 'UK-Eng' context will be selected for the mapping. Ultimately, the data is imported into the English language dimension.

Important Considerations for Language Codes

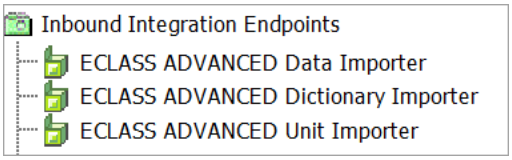
When dealing with the mapping of language codes, it is essential to consider the following points.

- Case Insensitivity of Language Codes: Language codes are case insensitive, meaning they can be mapped to the system without concern for the letter case. For example, language codes such as 'Eng,' 'eng,' or 'ENG' will all receive identical treatment.
- Default Language for Unit Imports: Units are typically imported in the English language. As a result, the import process automatically defaults to the dimension point mapped with the 'ENG' language code. In case the language code is not mapped to any dimension point in the system, the import will default to the context determined in the Inbound Integration Endpoint (IIEP).
- Multiple Dimension Points with the Same Language Code: If the same language code is mapped to multiple dimension points, the parser will consider the first language dimension point detected by the IIEP during the import.
- No Language Code Mapping: If the language code is not applied to any of the language dimension points, the import will default to the context determined in the IIEP.

- Erroneous Language Code Mapping: Mapping an incorrect or erroneous language code will also result in the import defaulting to the context determined in the IIEP.
- For IIEPs involved in importing Unit and Dictionary files, the ISO-639-2 attribute is defined as the Context Attribute. It is of utmost importance not to remove this attribute once defined. Accidentally removing it will cause the IIEP to fail in determining the language code provided in the file, leading to the default context being considered.
- After importing a file, to gain insights into the context determined for that import, users can refer to the Execution Report, which provides comprehensive information on the import process.

Update IIEPs

The second step of the Easy Setup action '2. ECLASS ADVANCED Web UI Easy Setup' creates three inbound integration endpoints for each file type supported by the solution. You can locate these endpoints under the Inbound Integration Endpoints node within the System Setup tab.



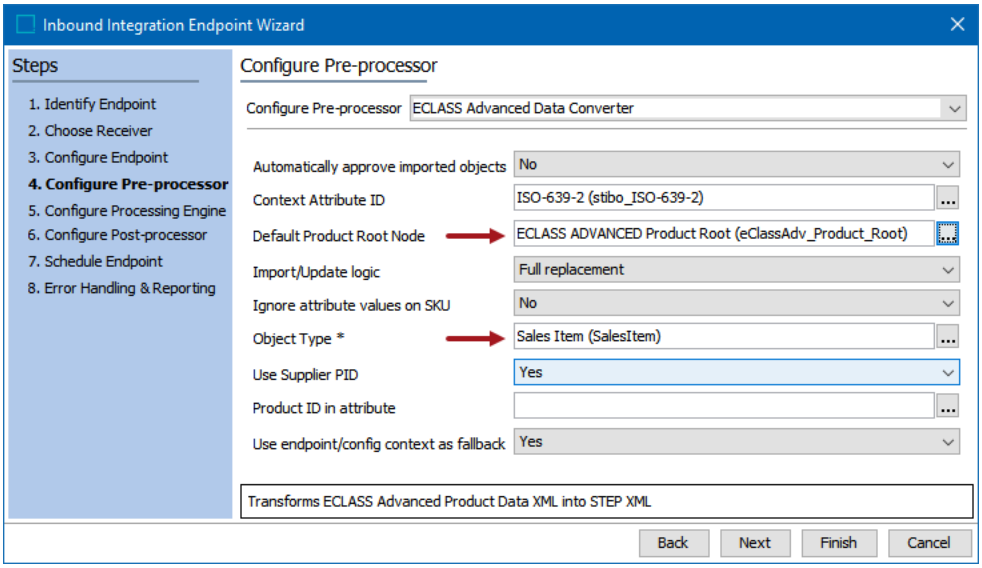
Although many options specific to the IIEPs are set and configured, it is imperative to manually set up certain parameters to ensure the IIEPs are fully prepared for usage.

While the 'ECLASS ADVANCED Unit Importer' and the 'ECLASS ADVANCED Dictionary Importer' are ready for immediate use, the 'ECLASS ADVANCED Data Importer' requires further configuration, as detailed below:

Configuration Steps

Update 'ECLASS ADVANCED Data Importer' Endpoint Parameters:

1. Right-click on the ECLASS ADVANCED Data Importer endpoint and select Edit Inbound Integration Endpoint. This action will launch the Inbound Integration Endpoint Wizard.
2. Navigate to the 'Configure Pre-processor' tab, where the 'ECLASS Advanced Data Converter' will be displayed as the pre-processor (refer to the screenshot below)
3. Update the following parameter:
   - **Object Type:** Click the ellipsis button (...) next to the 'Object Type' parameter to find and select the main SKU / Product Object Type. Selected object type has to correspond to the ECLASS data validity. Thus, ensure that the configured object type matches with the configuration set during the "1. ECLASS ADVANCED Easy Set-up" action (within the Easy Setup). For more information, refer to Run Easy Setup of ECLASS Advanced Industry Standard topic.



Only information pertaining to initial setup of the 'ECLASS ADVANCED Data Importer' is explained in this topic. For more detailed information on the rest of the parameters, refer to Configuring an IIEP for ECLASS Advanced Data Imports topic within the **ECLASS Advanced Importers** section of the **ECLASS Advanced Reference Guide**.

ECLASS Standard Supported Versions and Formats

The following are the supported import and export versions and/or formats for the ECLASS Advanced standard.

- **Data Importer:** BMEcat 2005.1
- **Data Exporter:** BMEcat 2005.1

ECLASS Advanced Reference Guide

This guide describes specific ECLASS Advanced reference material beyond what is provided in the ECLASS Quick Start Guide Introduction topic. This includes a detailed description of user functionalities that is provided with the solution after Easy Setup actions for the ECLASS Advanced standard has been completed by an admin.

This guide addresses the following topics:

- Importing ECLASS Advanced Files
- Exporting in BMEcat 2005.1 Format
- ECLASS Advanced Actions

Importing ECLASS Advanced Files

The ECLASS Advanced solution provides extensive import capabilities. The intention of the import aspect of the ECLASS Advanced solution is to provide pre-configured importers for different ECLASS Advanced files. Each customer can subsequently implement their unique validations, business processes, and data management protocols. To do this successfully, it is crucial to understand the import framework described in the ECLASS Advanced Import Framework topic.

Easy Setup creates and configures three File Loading Widgets in the Web UI that can be used for importing ECLASS Advanced files. Each of the ECLASS Advanced file types has their own importer, and more information for each can be found within their respective sections below.

> ⚠ **Important:** Imports should be executed in the given order for best results, as there are some dependencies between the imports.

1. ECLASS Advanced Unit Importer
2. ECLASS Advanced Dictionary Importer
3. ECLASS Advanced Data Importer

For information on supported versions, refer to the ECLASS Standard Supported Versions and Formats topic.

ECLASS Advanced Import Framework

All ECLASS Advanced IIEPs are configured to use Hotfolders Receiver.

Import Process Overview

Once a valid ECLASS Advanced file is uploaded using a File Loading Widget (or uploaded directly to a hotfolder), the file is picked up by an IIEP, and the IIEP starts a background process.

Uploading Files via Web UI

To upload a file to a specific hotfolder via a Web UI, a File Loading Widget can be used. For generic information on how to use a File Loading Widget, refer to File Loading Widget topic. Further, details relevant to specific ECLASS Advanced importers are described in the following topics:

- Using ECLASS Advanced Unit Importer
- Using ECLASS Advanced Dictionary Importer
- Using ECLASS Advanced Data Importer

Uploading Files via the Hotfolder

To upload a file to a specific hotfolder:

1. Navigate to /upload/hotfolders/[Importer].
2. Drop the file into the folder with a name aligning with the importer (e.g., ECLASS ADVANCED Data Importer should have files dropped into the 'EclassAdvDataImporter').
3. The endpoint will pick up the file at the next scheduled polling, and the file load will begin.

> **Note:** The endpoints are created with a default schedule of polling the hotfolder once per minute, but this setting is adjustable for each endpoint, so it may vary between importers and implementations.

ECLASS Advanced Unit Importer

The primary objective of the ECLASS Advanced Unit Importer is to offer a convenient out-of-the-box solution for importing Unit data in a supported XML format.

Purpose and Considerations of Unit File Import

The ECLASS Advanced Dictionary file can include Units that may not have been generated during a prior ECLASS Basic import process in STEP. As a result, it is essential to perform a full import of the ECLASS Unit file before proceeding with the import of the ECLASS ADVANCED Dictionary file.

Following are key considerations:

- The ECLASS Basic and ECLASS Advanced Unit files are identical.
- Typically, the Unit file is a part of the ECLASS-provided .zip archive, which must be manually extracted to access the Unit XML file. The Unit Importer exclusively accepts XML file formats.
- The Unit file may contain Unit Groups without allocated Units. These Unit Groups will be imported into STEP, irrespective of whether Units are present.
- Units within the Unit file may exist without a designated Unit group. Such Units will be listed under a generated Unit group named 'eClass ungrouped units (ID = eClass_UngroupedUnits).'
- STEP does not support structuring Units beyond the Unit Group level. Therefore, all Unit Groups are created on the same level as children of the 'Unit' node.
- The creation of Unit Group and Unit STEP IDs and Names follows the ECLASS Basic structure logic.
- Version dependencies for Unit Groups and Units are not taken into account. The Unit metadata attribute 'EclassImportVersion' will contain the latest version number.
- Unit conversions are not considered, and consequently, no data related to Unit conversions is imported.
- The ECLASS Unit file does not contain any Units related to Currencies. However, Currency Units are managed within the ECLASS Advanced Dictionary file. Therefore, the ECLASS Advanced Dictionary Importer will create the Unit Group called 'eClass Currency (ID = eClass_Currency),' which will include the relevant currency Units found in the Dictionary file, such as EUR.

- Despite the presence of both English and German languages in the Unit file, Unit Groups and Units are exclusively created in English. Consequently, it is recommended to import Units into the Global context. Alternatively, if you intend to import the units in the German language, users should switch the tag as follows:
  - From:

    ```
    <content_description language_code="ENG">ECLASS12.0</content_
    description>
    ```

  - To:

    ```
    <content_description language_code="DEU">ECLASS12.0</content_
    description>
    ```

This section includes information on:

- Using ECLASS Advanced Unit Importer
- Configuring ECLASS Advanced Unit Importer

Configuring ECLASS Advanced Unit Importer

> **Note:** If the Easy Setup actions for the ECLASS Advanced Component model have been completed, then the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

The following topics provide the configuration steps necessary to allow users to be able to drag and drop Unit files onto a configured File Loading Widget, and monitor the progress of the import file in the created IIEP Background Process in the workbench.

- Configuring an IIEP for ECLASS Advanced Unit Imports
- Configuring a File Loading Widget for ECLASS Advanced Unit Imports

Configuring an IIEP for ECLASS Advanced Unit Imports

> **Note:** If the Easy Setup actions for the ECLASS Advanced Component model have been completed, then the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

An inbound integration endpoint (IIEP) can be configured in the workbench to help the process of importing Unit files into STEP. Once an IIEP is configured for ECLASS Advanced Unit imports, Unit files can be imported after they are uploaded either to a configured hotfolder, or to a File Loading Widget on a Web UI Homepage. For more information, refer to ECLASS Advanced Unit Importer topic.

This section describes how to configure an IIEP that can allow for the automated processing of Unit files. Each screenshot example within this section provides recommended values for the parameters and ECLASS Advanced Unit Importer.

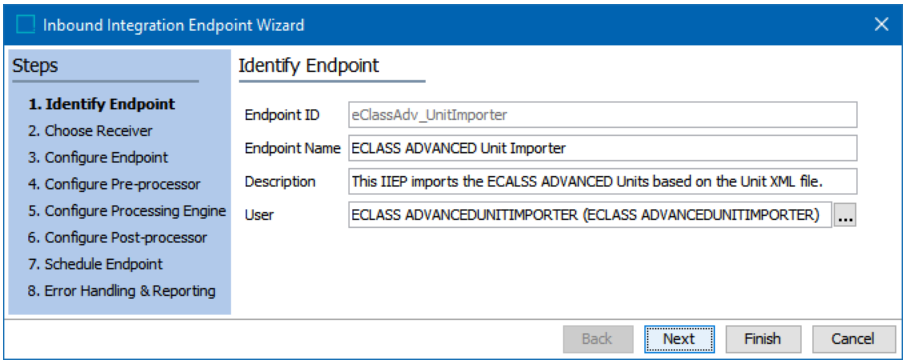
Prerequisites

This topic aims to acquaint users with the IIEP specifically designated for the import of Unit files. It does not cover general IIEP functionalities. It is assumed that individuals configuring an IIEP for ECLASS Advanced Unit Import are well-versed in configuring and processing standard inbound integration endpoints. For a comprehensive understanding of the standard functionalities provided in inbound integration endpoints, refer to Inbound Integration Endpoints topic within the Data Exchange documentation.

Configuration Steps

1. In the workbench, go to System Setup, select and right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.

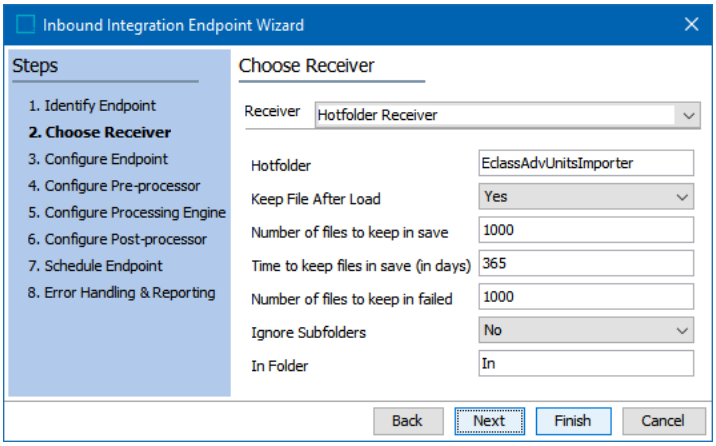
2. Once the Inbound Integration Endpoint Wizard displays, The parameters are to be populated as recommended and shown below.



For more information about the parameters available within the Identify Endpoint step, refer to IIEP - Identify Endpoint topic within the Data Exchange documentation.

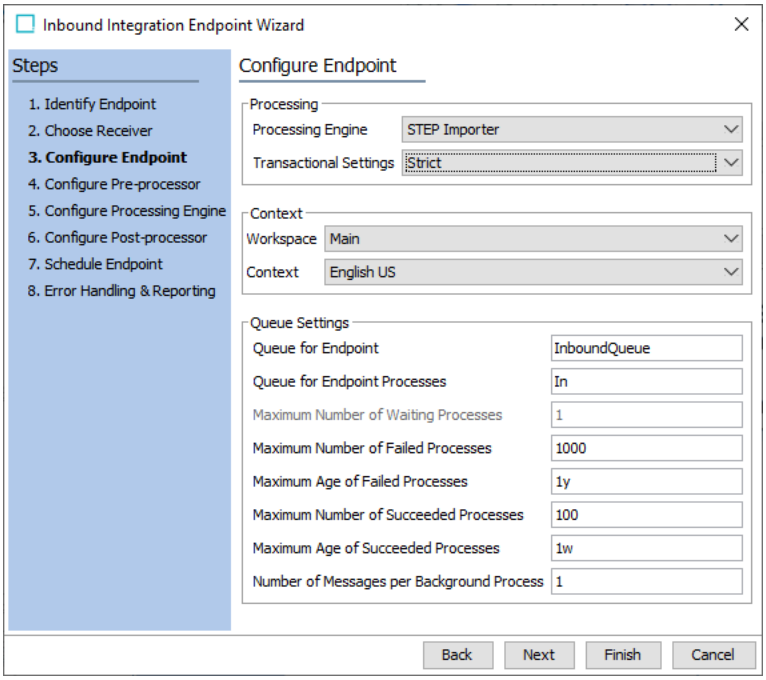
3. Click the **Next** button, and the Choose Receiver parameters will display. The parameters are to be populated as recommended and shown below. The mandatory parameter Hotfolder must be populated with a hotfolder name before the Next button will enable. In the screenshot below, the Hotfolder parameter is populated with the value 'EClassAdvUnitImporter.'

> **Note:** The value within this hotfolder parameter will be used to create the new hotfolder, once the IIEP Wizard is complete.



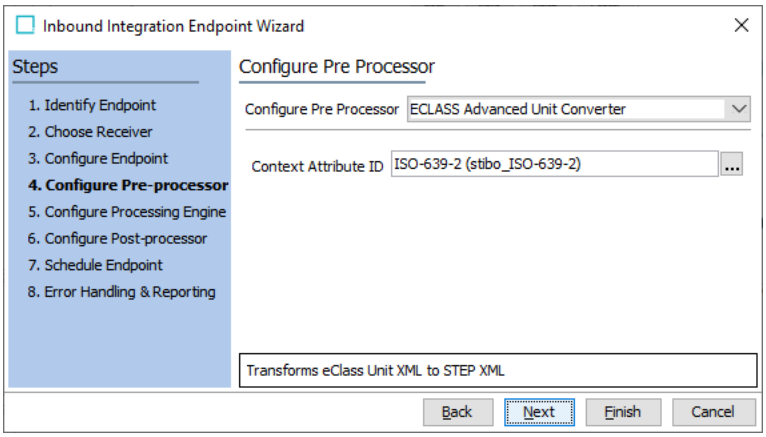
For more information about the parameters, refer to IIEP - Choose Receiver topic within the Data Exchange documentation.

4. Click the **Next** button, and the Configure Endpoint parameters will display. The parameters are to pre-populated with the recommended values as shown below.



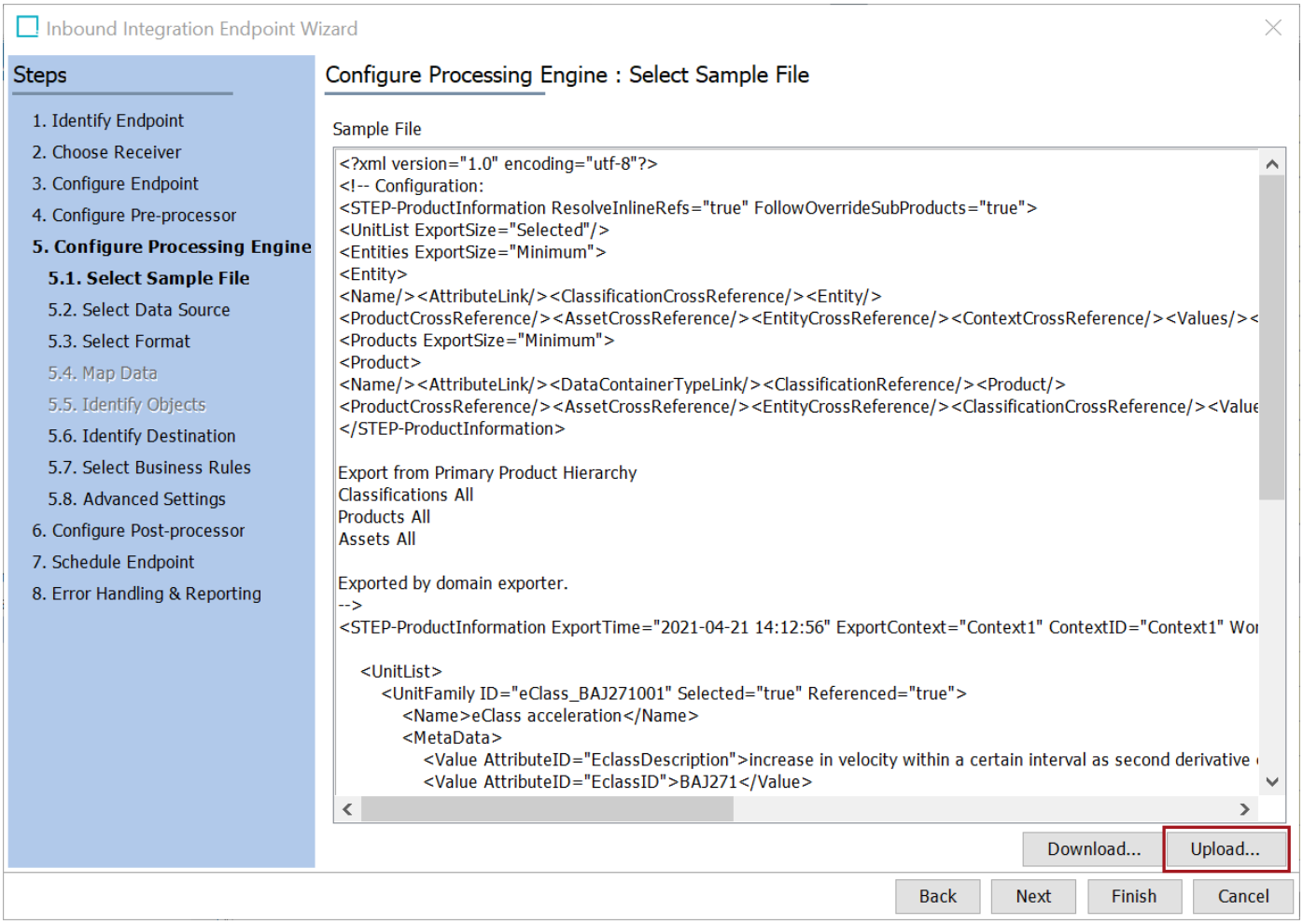
For more information about the parameters, refer to IIEP - Configure Endpoint topic within the Data Exchange documentation.

5. Click the **Next** button, and the Configure Pre-processor parameter will display. The selection of the pre-processor within this step makes the IIEP unique for importing Unit files. The parameters are to be populated as recommended and shown below:



- Configure Pre-processor has to be populated with ECLASS Advanced Unit Converter option. This is an exclusive pre-processor for importing Unit files. For more information about the parameter, refer to IIEP - Configure Pre-processor topic within the Data Exchange documentation.
- Context Attribute ID parameter is to be populated with the attribute ISO-639-2 (ID = stibo_ISO-639-2). This attribute is created by the Easy Setup action and holds the language mappings. For more information about language mappings, refer to Prepare the Language Dimension Mapping.

6. Click the **Next** button, and the Configure Processing Engine: Select Sample File field for the STEP Importer processing engine will display.



- In **5.1 Select Sample File** step, click the Upload button to upload a sample STEPXML file. For information about how to upload a sample file, refer to IIEP - Configure STEP Importer Processing Engine topic within the Data Exchange documentation.

The basic data structure of a sample Unit file is provided below:

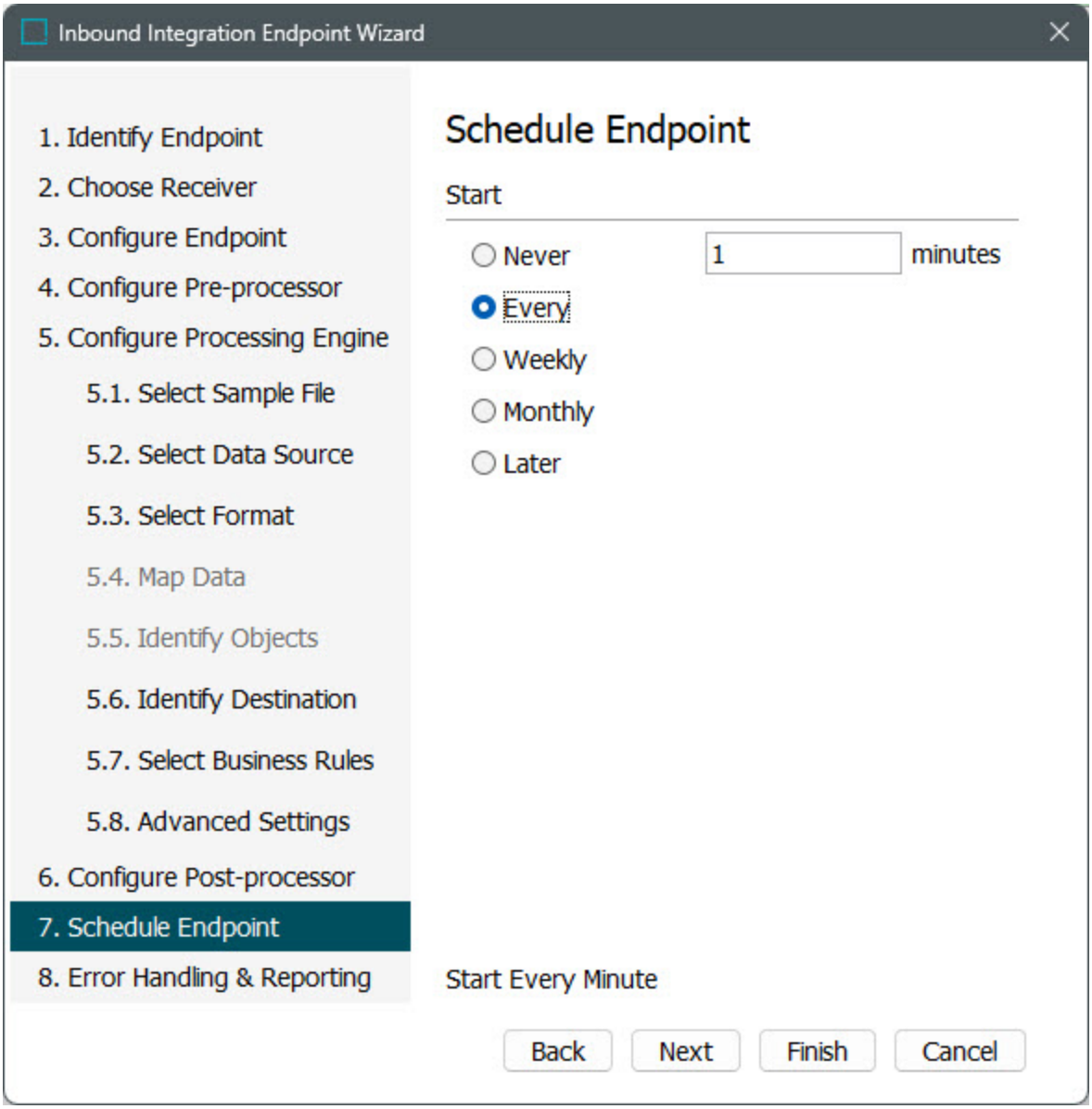
```
<?xml version="1.0" encoding="utf-8"?>
<!-- Configuration:
<STEP-ProductInformation ResolveInlineRefs="true"
FollowOverrideSubProducts="true">
<UnitList ExportSize="Selected"/>
<Entities ExportSize="Minimum">
<Entity>
<Name/><AttributeLink/><ClassificationCrossReference/><Entity/>

<ProductCrossReference/><AssetCrossReference/><EntityCrossReference/>
<ContextCrossReference/><Values/></Entity></Entities>
<Products ExportSize="Minimum">
<Product>

<Name/><AttributeLink/><DataContainerTypeLink/><ClassificationReferen
ce/><Product/>

<ProductCrossReference/><AssetCrossReference/><EntityCrossReference/>
<ClassificationCrossReference/><Values/><OverrideSubProduct/></Produc
t></Products>
</STEP-ProductInformation>

Export from Primary Product Hierarchy
Classifications All
Products All
Assets All

Exported by domain exporter.
-->
<STEP-ProductInformation ExportTime="2021-04-21 14:12:56"
ExportContext="Context1" ContextID="Context1" WorkspaceID="Main"
UseContextLocale="false">
```

```
    <UnitList>
        <UnitFamily ID="eClass_BAJ271001" Selected="true"
Referenced="true">
            <Name>eClass acceleration</Name>
            <MetaData>
                <Value AttributeID="EclassDescription">increase in
velocity within a certain interval as second derivative of the
distance per time</Value>
                <Value AttributeID="EclassID">BAJ271</Value>
                <Value
AttributeID="EclassPrimaryKey">BAJ271001</Value>
            </MetaData>
            <Unit ID="eClass_AAA225002" Selected="true"
Referenced="true">
                <Name>ft/sÂ²</Name>
                <MetaData>
                    <Value AttributeID="EclassDescription">unit foot
according to the Anglo-American and the Imperial system of units
divided by the power of the SI base unit second with the exponent 2
with the relation according to NIST: 1 ft/sÂ² = 0,304 8 m/sÂ²</Value>
                    <Value
AttributeID="EclassImportVersion">10.1</Value>
                    <Value AttributeID="EclassID">AAA225</Value>
                    <Value
AttributeID="EclassPrimaryKey">AAA225002</Value>
                </MetaData>
            </Unit>
            <Unit ID="eClass_AAA597002" Selected="true"
Referenced="true">
                <Name>m/sÂ²</Name>
                <MetaData>
                    <Value AttributeID="EclassDescription">SI base
unit metre divided by the power of the SI base unit second and the
```

```
exponent 2</Value>
                    <Value
AttributeID="EclassImportVersion">10.1</Value>
                    <Value AttributeID="EclassID">AAA597</Value>
                    <Value
AttributeID="EclassPrimaryKey">AAA597002</Value>
                </MetaData>
            </Unit>
        </UnitFamily>
    </UnitList>
</STEP-ProductInformation>
```

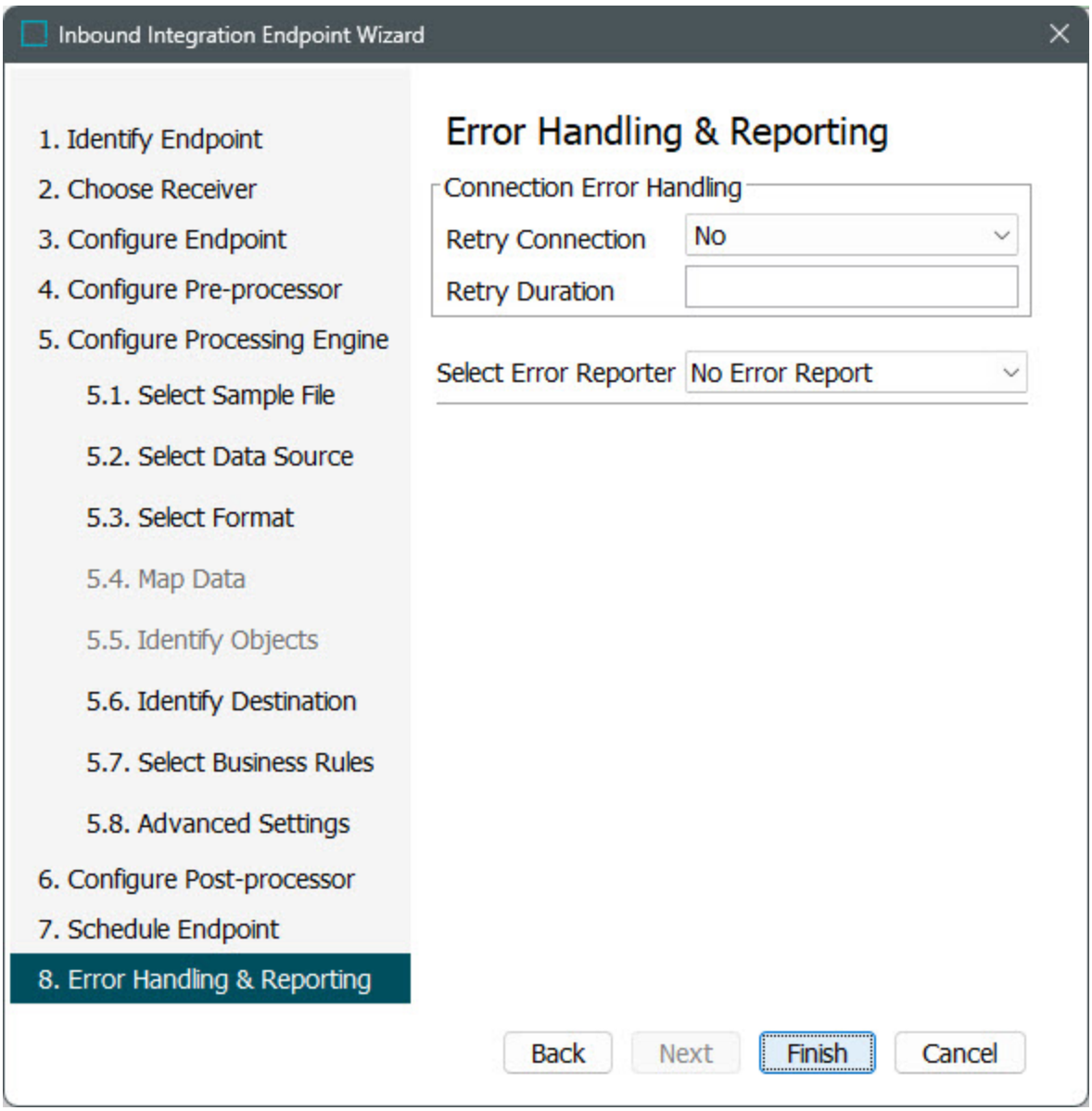
- Further potential sub-steps underneath the Configure Processing Engine are optional. For more information about these steps, refer to IIEP - Configure STEP Importer Processing Engine topic within the Data Exchange documentation.

7. Click the **Next** button to display the Schedule Endpoint parameters (bypass the Configure Post-processor parameters). By default, 'Never' is selected. Optionally, update the values to those shown below.



For more information about the parameters available within this step, refer to IIEP - Schedule Endpoint topic within the Data Exchange documentation.

8. Click the **Next** button, and the Error Handling & Reporting step will display. The parameters are to be populated as recommended and shown below:



For more information about the parameters available within this step, refer to IIEP - Error Handling & Reporting topic within the Data Exchange documentation.

9. Click the **Finish** button, the Inbound Integration Endpoint Wizard will close, and the newly created endpoint will display within the workbench.

> ⚠ **Important:** An endpoint must be enabled before it can start processing data. For more information, refer to Running an Inbound Integration Endpoint topic within the Data Exchange documentation.

If users need to access the IIEP via a Web UI, then the IIEP must be configured within a File Loading Widget. For more information, refer to the Configuring a File Loading Widget for ECLASS Advanced Data Imports topic .

Configuring a File Loading Widget for ECLASS Advanced Unit Imports

When configured, Web UI users can import Unit files into STEP using a File Loading Widget.

Prerequisites

It is expected that anyone configuring the ECLASS Advanced Unit Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, refer to Designer Access topic within the Web User Interfaces documentation.

Before configuring the Web UI portion of this solution, an IIEP for an ECLASS Advanced Unit Importer must be configured within the workbench. For more information, refer to Configuring an IIEP for ECLASS Advanced Unit Imports topic.

Additionally, it is helpful to know how to add a widget to a Web UI Homepage. Details on how to do this can be found in Adding Widgets to a Homepage topic in the Getting Started documentation.

Configuration

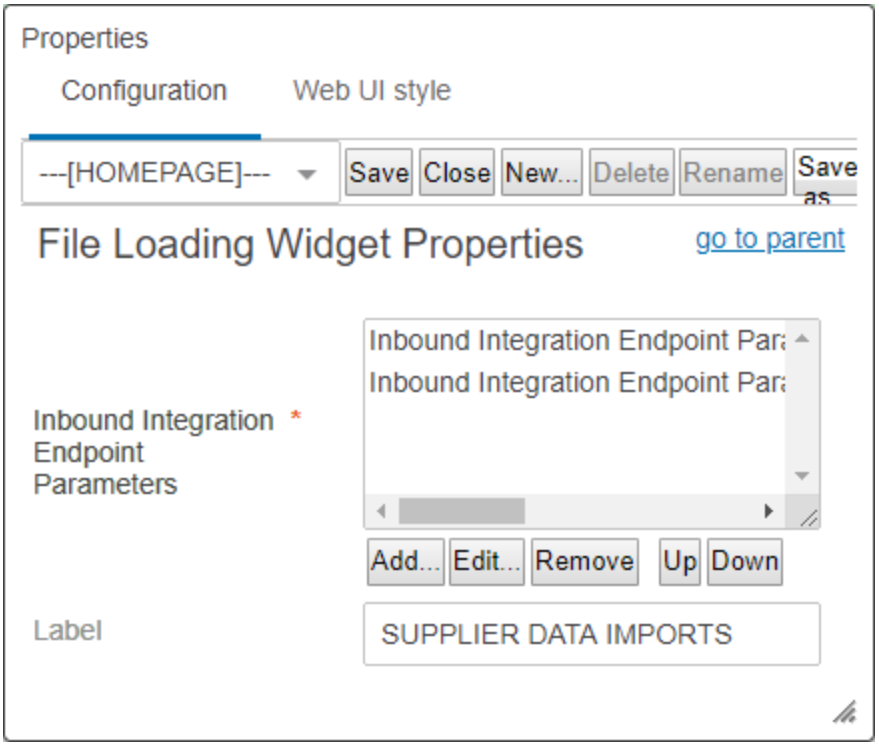
Each screenshot example within this section provides recommended values for the parameters in ECLASS Advanced Unit Importer.

This topic describes how to configure a File Loading Widget so that users can drag and drop ECLASS Advanced Unit files onto a File Loading Widget on a Web UI Homepage.

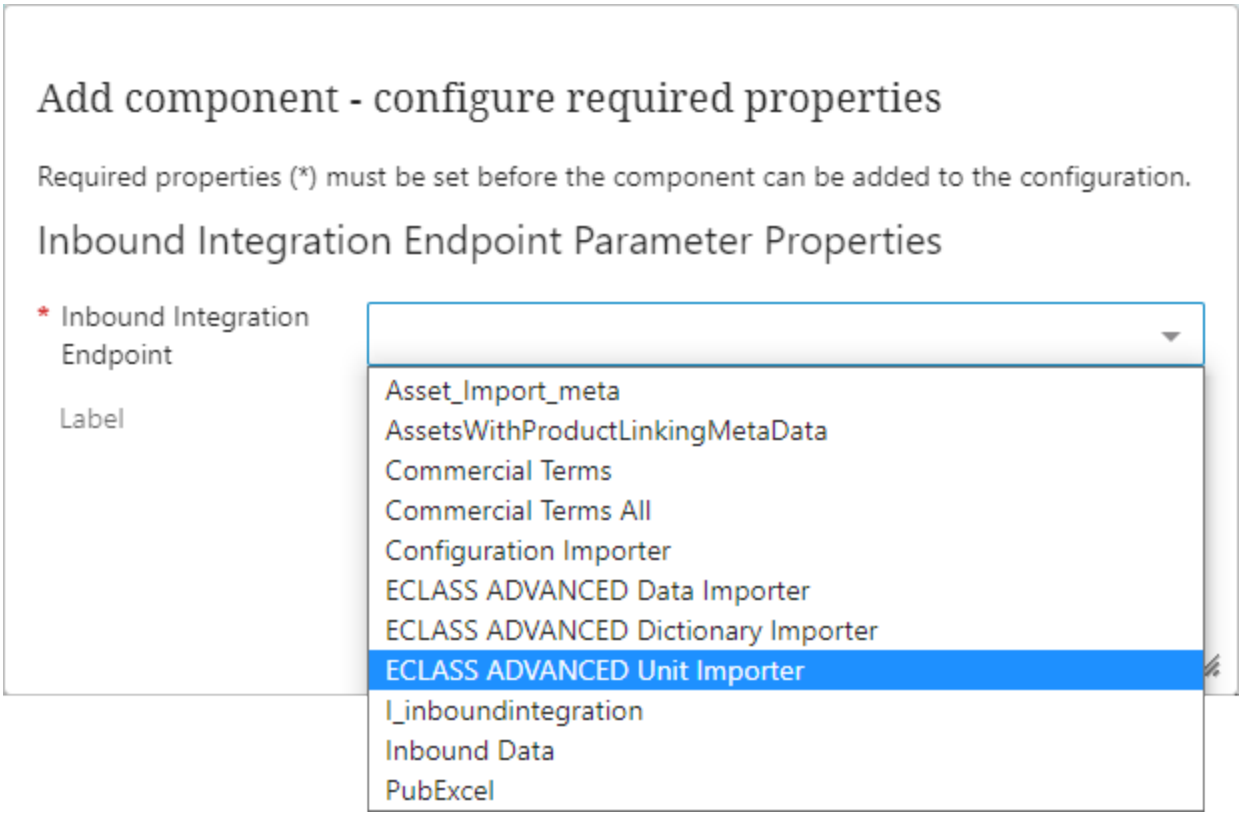
> **Note:** If Easy Setup actions for the ECLASS Advanced solution have been completed as described in the Run Easy Setup of ECLASS Advanced Industry Standard topic of the 'ECLASS Quick Start Guide', then the 'Unit Importer' File Loading Widget will automatically be added to the Web UI Homepage as shown in the examples below. Otherwise, the steps below can be used to complete configuration.

1. In the designer, select an existing File Loading Widget to be used, or add a new File Loading Widget to the Homepage Widget Grid component. For more information, refer to the File Loading Widget topic within the Web User Interfaces documentation.

2. Go to the Inbound Integration Endpoint Parameters field, click the **Add** button, and the Inbound Integration Endpoint Parameter Properties dialog will display.



3. Click the dropdown for the Inbound Integration Endpoint parameter, and select **ECLASS ADVANCED Unit Importer** (the IIEP created for ECLASS Advanced Unit imports).

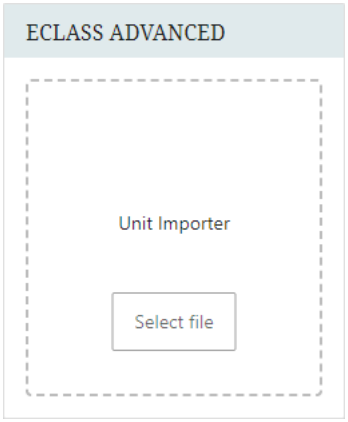


Note: If the desired IIEP does not display in the dropdown, then it can be created using the steps described in Configuring an IIEP for ECLASS Advanced Unit Imports topic.

4. Optionally, provide a label to be displayed within the drop zone of the widget.

   In the example below:

   - A File Loading Widget labeled as 'ECLASS ADVANCED' is displayed above its configurations.
   - The File Loading Widget and its configurations are shown with the default configurations provided automatically when Easy Setup actions for the ECLASS Advanced component are completed.



5. Click the **Save** and **Close** buttons to save the changes and close the designer.

Using ECLASS Advanced Unit Importer

ECLASS Advanced Unit file can be imported into STEP by uploading it to either a configured hotfolder, or through a File Loading Widget on a Web UI Homepage. The intention of the Unit Importer Web UI setup included within this topic is to provide an out-of-the-box solution for importing data included within a supported ECLASS Advanced Unit format.

Prerequisites

If you have completed the Easy Setup actions for the ECLASS ADVANCED Model, the functionalities outlined in this topic should be readily available for use. Otherwise, configuration is required. For information on how to configure ECLASS Advanced Unit importer, refer to Configuring ECLASS Advanced Unit Importer topic.

The Unit Importer exclusively supports XML file formats. In case the XML file is zipped, it is essential to manually extract the .zip file to access the Unit file.

Import Process Overview

Once a valid ECLASS Advanced file is uploaded using a File Loading Widget (or uploaded directly to a hotfolder), the file is picked up by an IIEP, and the IIEP starts a background process.

Procedure

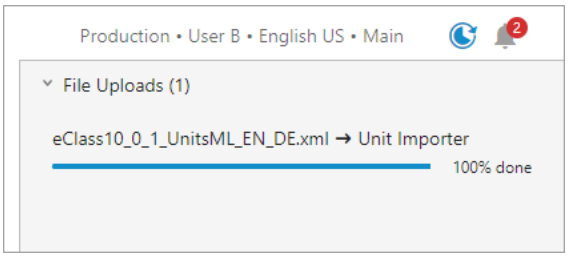
1. Access the ECLASS Advanced Web UI Homepage.
2. Upload a valid Unit file to the hotfolder (root/upload/hotfolders/EclassAdvUnitImporter/In), or use the 'Unit Import' File Loading Widget.

   For information about File Loading Widget, refer to File Loading Widget topic within the Web User Interfaces documentation.

   For more information about uploading files directly to the hotfolder, refer to ECLASS Advanced Import Framework topic.

3. Once the upload has started, users can view the progress of the upload using the 'Recent background processes' side panel.

   In the example below, the 'Recent background processes' side panel is expanded, and the file 'eClass10_0_1_UnitsML_EN_DE.xml' is 100% processed.



   For more information on using the 'Recent background processes' side panel, refer to Recent Background Processes Side Panel topic within the Web User Interfaces documentation.

After uploading the file, the IIEP picks it up and initiates a Background Process for the import process. The file loading widget does not provide background process monitoring in the Web UI. You can monitor the import status within the workbench through the IIEP Background Process that is generated.

For more information about monitoring the IIEP via background process, refer to Monitoring an IIEP via Background Process topic within the Data Exchange documentation.

ECLASS Advanced Dictionary Importer

The main goal of the ECLASS Advanced Dictionary Importer is to provide a seamless ready-to-use solution for importing Dictionary data in a supported XML format.

The Dictionary file is typically bundled within the ECLASS-provided .zip file. To access the Dictionary file, you will need to manually unzip the archive. The Dictionary Importer exclusively accepts XML file formats.

This section includes information on:

- Using ECLASS Advanced Dictionary Importer
- Configuring ECLASS Advanced Dictionary Importer

Configuring ECLASS Advanced Dictionary Importer

> **Note:** If the Easy Setup actions for the ECLASS Advanced Component model have been completed, then the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

The following topics provide the configuration steps necessary to allow users to be able to drag and drop Dictionary files onto a configured File Loading Widget, and monitor the progress of the import file in the created IIEP Background Process in the workbench.

- Configuring an IIEP for ECLASS Advanced Dictionary Imports
- Configuring a File Loading Widget for ECLASS Advanced Dictionary Imports

Configuring an IIEP for ECLASS Advanced Dictionary Imports

Note: If the Easy Setup actions for the ECLASS Advanced Component model have been completed, then the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

An inbound integration endpoint (IIEP) can be configured in the workbench to help the process of importing Dictionary files into STEP. Once an IIEP is configured for ECLASS Advanced Dictionary imports, Dictionary files can be imported after they are uploaded either to a configured hotfolder, or to a File Loading Widget on a Web UI Homepage. For more information, refer to ECLASS Advanced Data Importer topic.

This section describes how to configure an IIEP that can allow for the automated processing of Dictionary files. Each screenshot example within this section provides recommended values for the parameters in ECLASS Advanced Dictionary Importer.

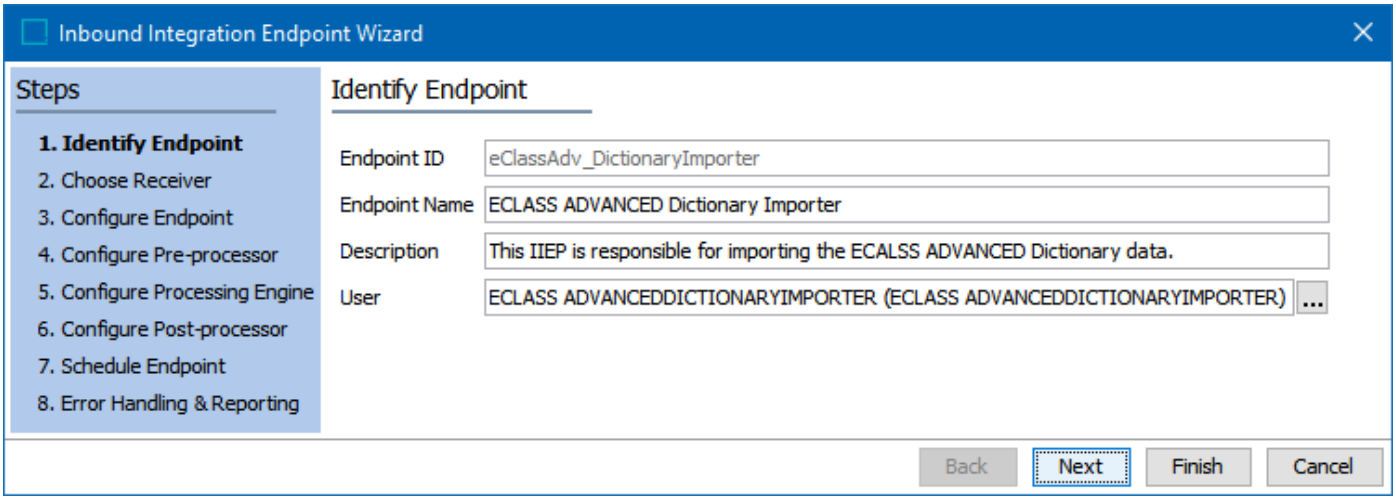
Prerequisites

This topic aims to acquaint users with the IIEP specifically designated for the import of Dictionary files. It does not cover general IIEP functionalities. It is assumed that individuals configuring an IIEP for ECLASS Advanced Dictionary Import are well-versed in configuring and processing standard inbound integration endpoints. For a comprehensive understanding of the standard functionalities provided in inbound integration endpoints, refer to Inbound Integration Endpoints topic within the Data Exchange documentation.

Configuration Steps

1. In the workbench, go to System Setup, select and right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.

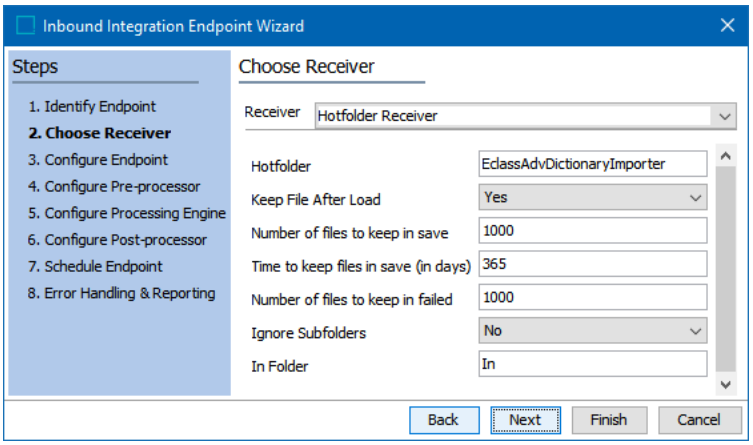
2. Once the Inbound Integration Endpoint Wizard displays, The parameters are to be populated as recommended and shown below.



For more information about the parameters available within the Identify Endpoint step, refer to IIEP - Identify Endpoint topic within the Data Exchange documentation.

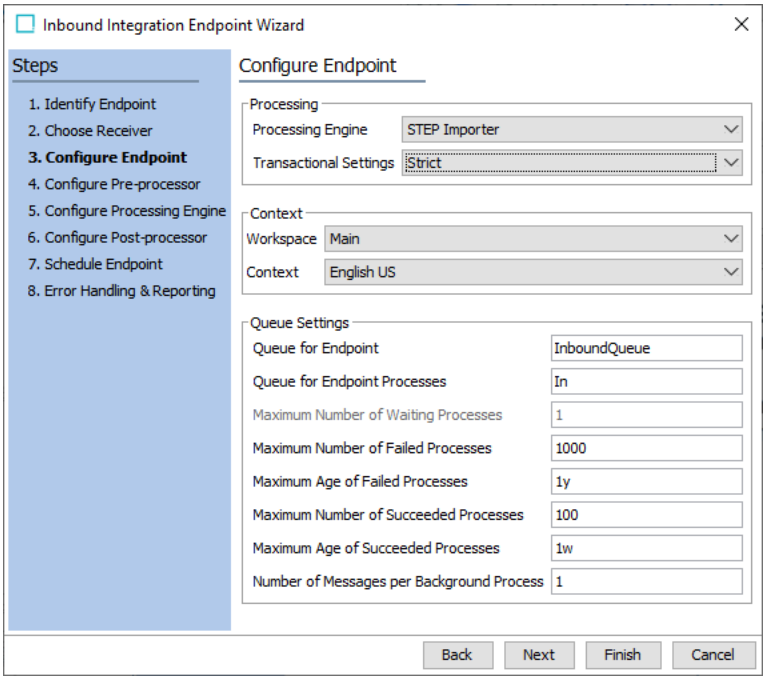
3. Click the **Next** button, and the Choose Receiver parameters will display. The parameters are to be populated as recommended and shown below. The mandatory parameter Hotfolder must be populated with a hotfolder name before the Next button will enable. In the screenshot below, the Hotfolder parameter is populated with the value 'EClassAdvDictionaryImporter.'

> **Note:** The value within this hotfolder parameter will be used to create the new hotfolder, once the IIEP Wizard is complete.



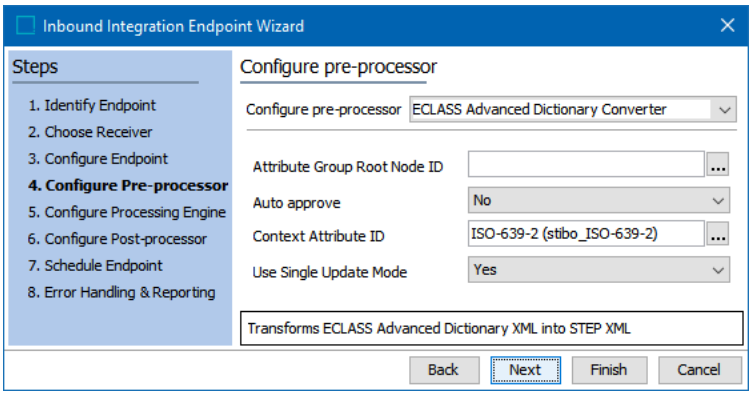
For more information about the parameters, refer to IIEP - Choose Receiver topic within the Data Exchange documentation.

4. Click the **Next** button, and the Configure Endpoint parameters will display. The parameters are to pre-populated with the recommended values as shown below.



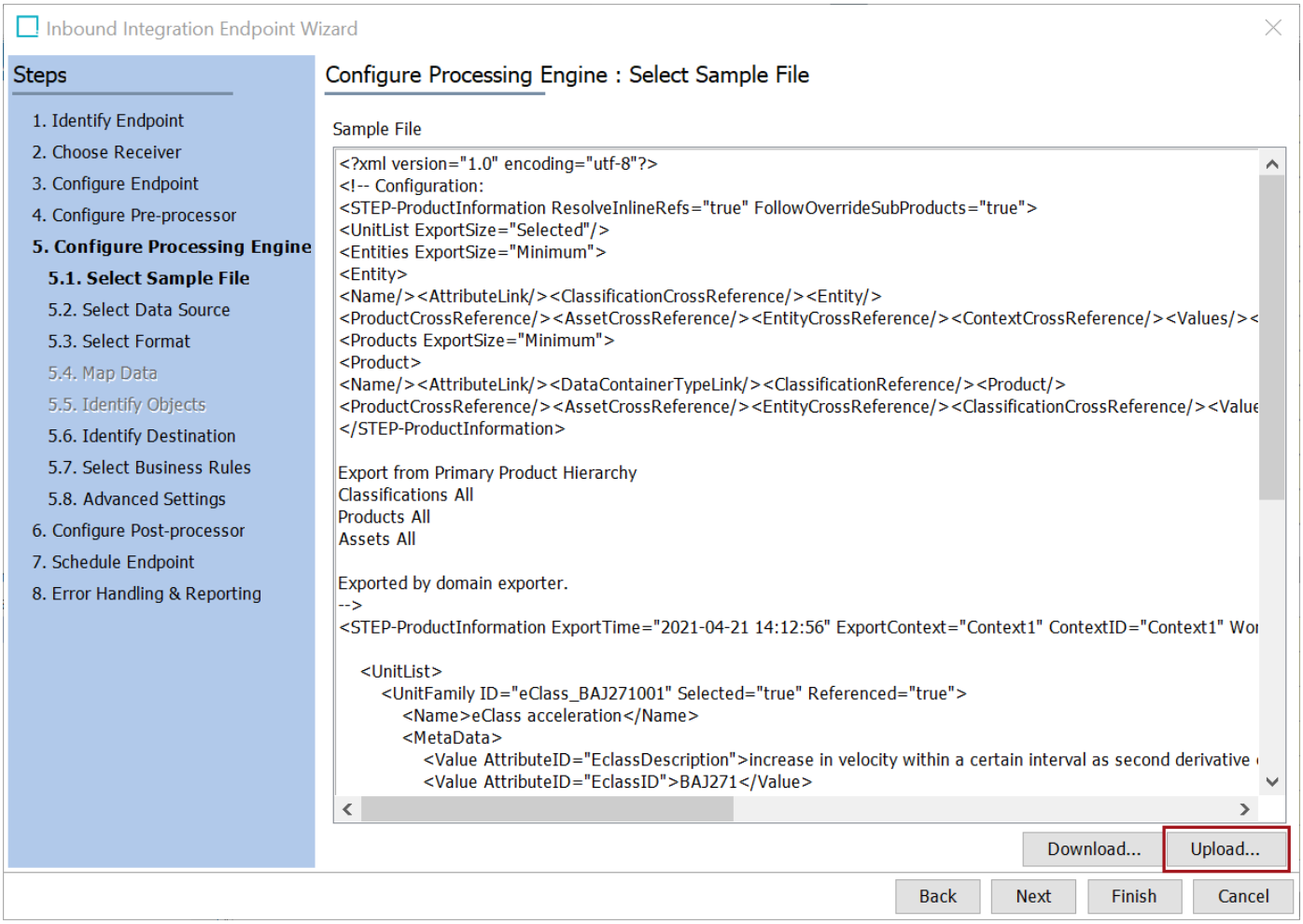
For more information about the parameters, refer to IIEP - Configure Endpoint topic within the Data Exchange documentation.

5. Click the **Next** button, and the Configure Pre-processor parameter will display. The selection of the pre-processor within this step makes the IIEP unique for importing Dictionary files. The parameters are to be populated as recommended and shown below:



- **Configure Pre-processor:** This parameter has to be populated with ECLASS Advanced Dictionary Converter option. This is an exclusive pre-processor for importing Dictionary files. For more information about the parameter, refer to IIEP - Configure Pre-processor topic within the Data Exchange documentation.
- **Attribute Group Root Node ID:** Upon defining a value (root node) within this parameter, it serves as a directive for determining the specific location for attribute import. If left empty, the attributes will be imported beneath the standard top Attribute Group Root node.
- **Auto approve:**This parameter determines whether the imported classification nodes are to be automatically approved or not. The default setting is No.
- **Context Attribute ID:** This parameter is to be populated with the attribute ISO-639-2 (ID = stibo_ISO-639-2). This attribute is created by the Easy Setup action and holds the language mappings. For more information about language mappings, refer to Prepare the Language Dimension Mapping topic.

6. Click the **Next** button, and the 'Configure Processing Engine: Select Sample File' field for the STEP Importer processing engine will display.



- In **5.1 Select Sample File** step, click the Upload button to upload a sample STEPXML file. For information about how to upload a sample file, refer to IIEP - Configure STEP Importer Processing Engine topic within the Data Exchange documentation.

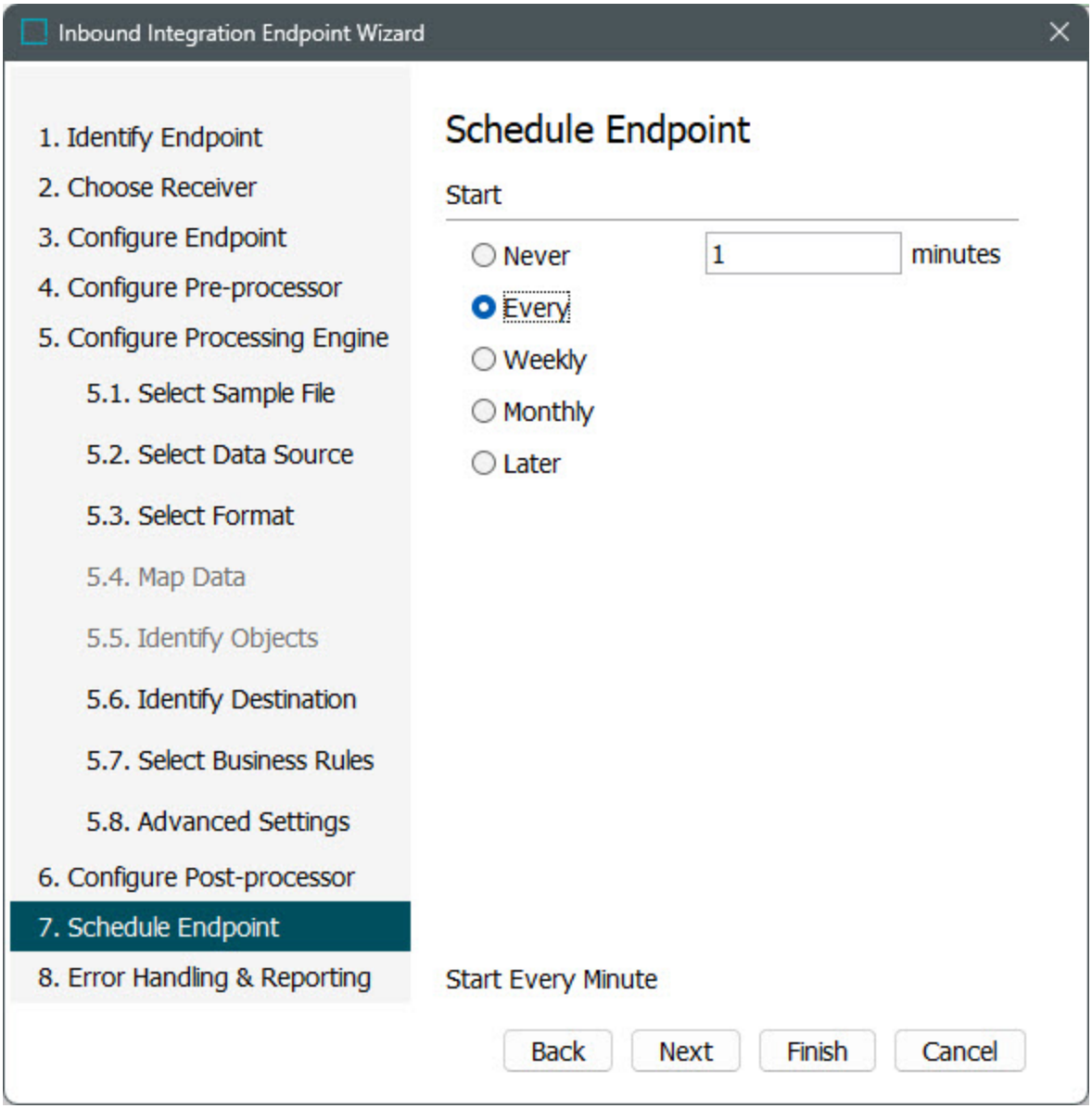
```
<?xml version="1.0" encoding="UTF-8"?>
<STEP-ProductInformation ExportTime="2021-06-15 12:53:15"
ContextID="Context1" WorkspaceID="Main" UseContextLocale="false"
SingleUpdateMode="Y">

    <AttributeGroupList>
        <AttributeGroup ID="Attribute group root"
ShowInWorkbench="true" ManuallySorted="false">
            <AttributeGroup ID="eClassAdv_Attributes"
```

```
ShowInWorkbench="true" ManuallySorted="false">
                <Name>eClass Advanced Attributes</Name>
                <AttributeGroup ID="eClassAdv_101_Attributes"
ShowInWorkbench="true" ManuallySorted="false">
                    <Name>eClass Advanced 101 Attributes</Name>
                </AttributeGroup>
                <AttributeGroup ID="eClassAdv_101_DataContainers"
ShowInWorkbench="true" ManuallySorted="false">
                    <Name>eClass Advanced 101 Data Containers</Name>
                </AttributeGroup>
                <AttributeGroup ID="eClassAdv_101_
DCLinkTypeAttributes" ShowInWorkbench="true" ManuallySorted="false">
                    <Name>eClass Advanced 101 Data Container Link
Type Attributes</Name>
                </AttributeGroup>
            </AttributeGroup>
        </AttributeGroup>
    </AttributeGroupList>

    <AttributeList>
        <Attribute ID="eClass_BAE519002" MultiValued="true"
ProductMode="Normal" Referenced="false" Selected="true"
FullTextIndexed="false" ExternallyMaintained="false" Derived="false"
Mandatory="false">
            <Name>Driver</Name>
            <Validation BaseType="text" MinValue="" MaxValue=""
MaxLength="" InputMask=""/>
            <DimensionLink DimensionID="Language"/>
            <MetaData>
                <Value AttributeID="EclassID">BAE519</Value>
                <Value
AttributeID="EclassPrimaryKey">BAE519002</Value>
                <Value AttributeID="eClassAdv_IRDI">0173-1#02-
BAE519#002</Value>
```

```
                <Value AttributeID="eClassAdv_
PropertyDomainType">TRANSLATABLE_STRING_TYPE_Type</Value>
                <Value AttributeID="eClassAdv_Description">small
additional programs enabling the operation of a device connected to
the PC, e.g. modem, printer or sound card. they operate as
interpreters between the operating system and he connected device.
Drivers are included in the hardware purchase and are normally also
included in the operating system. the operation of almost all devices
can be regularly updated via the Internet. To this end, simply
download the manufacturer's free driver update from the Website.
Links to the download sites of the provider: www.treiber.de</Value>
                <Value AttributeID="eClassAdv_AttributeType">NON_
DEPENDENT_P_DET_Type</Value>
                <Value AttributeID="eClassAdv_DependsOn"></Value>
                <Value AttributeID="eClassAdv_
ReferenceTarget"></Value>
            </MetaData>
            <AttributeGroupLink AttributeGroupID="eClassAdv_101_
Attributes"/>
            <UserTypeLink UserTypeID="SKU"/>
            <UserTypeLink UserTypeID="eClassAdv_ProductBlock"/>
        </Attribute>
    </AttributeList>

</STEP-ProductInformation>
```

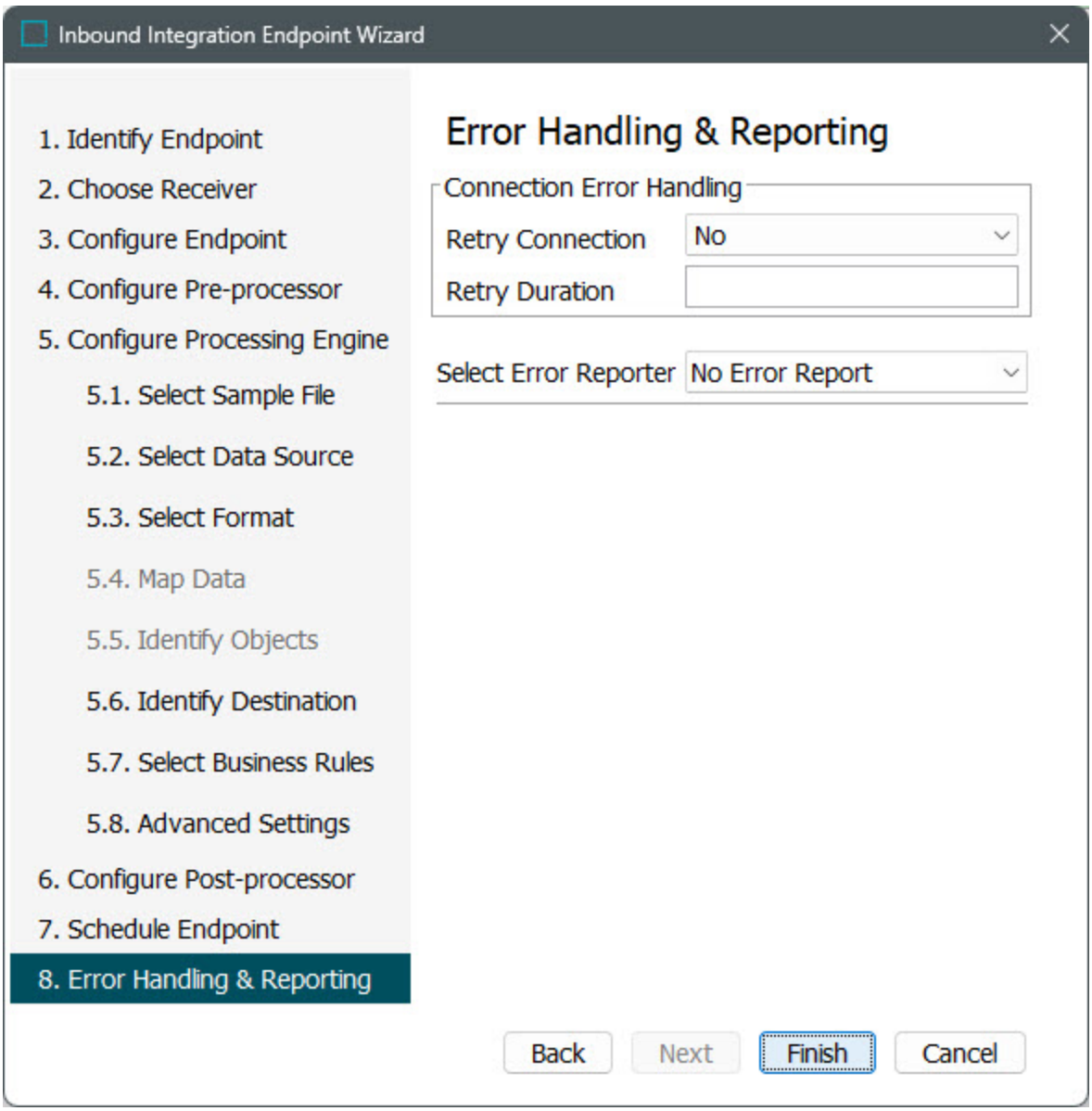
- Further potential sub-steps underneath the Configure Processing Engine are optional. For more information about these steps, refer to IIEP - Configure STEP Importer Processing Engine topic within the Data Exchange documentation.

7. Click the **Next** button to display the Schedule Endpoint parameters (bypass the Configure Post-processor parameters). Update the values to those shown below.



For information about the parameters available within this step, refer to IIEP - Schedule Endpoint topic within the Data Exchange documentation.

8. Click the **Next** button, and the Error Handling & Reporting step will display. By default, the parameters are populated as recommended and shown below.



For information about the parameters available within this step, refer to IIEP - Error Handling & Reporting topic within the Data Exchange documentation.

9. Click the **Finish** button, the Inbound Integration Endpoint Wizard will close, and the newly created endpoint will display within the workbench.

> ⚠ **Important:** An endpoint must be enabled before it can start processing data. For more information, refer to Running an Inbound Integration Endpoint topic within the Data Exchange documentation.

If users need to access the IIEP via a Web UI, then the IIEP must be configured within a File Loading Widget. For more information, refer to the Configuring a File Loading Widget for ECLASS Advanced Data Imports topic.

Configuring a File Loading Widget for ECLASS Advanced Dictionary Imports

Web UI users can import Dictionary files into STEP using a File Loading Widget.

If Easy Setup actions for the ECLASS Advanced solution have been completed as described in Run Easy Setup of ECLASS Advanced Industry Standard topic of the 'ECLASS Quick Start Guide', then the 'Dictionary Importer' File Loading Widget will automatically be added to the Web UI Homepage as shown in the examples below. Otherwise, the steps below can be used to complete configuration.

Prerequisites

It is expected that anyone configuring the ECLASS Advanced Dictionary Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, refer to Designer Access topic within the Web User Interfaces documentation.

Before configuring the Web UI portion of this solution, an IIEP for an ECLASS Advanced Dictionary Importer must be configured within the workbench. For more information, refer to Configuring an IIEP for ECLASS Advanced Dictionary Imports topic.

Additionally, it is helpful to know how to add a widget to a Web UI Homepage. Details on how to do this can be found in Adding Widgets to a Homepage topic in the Getting Started documentation.

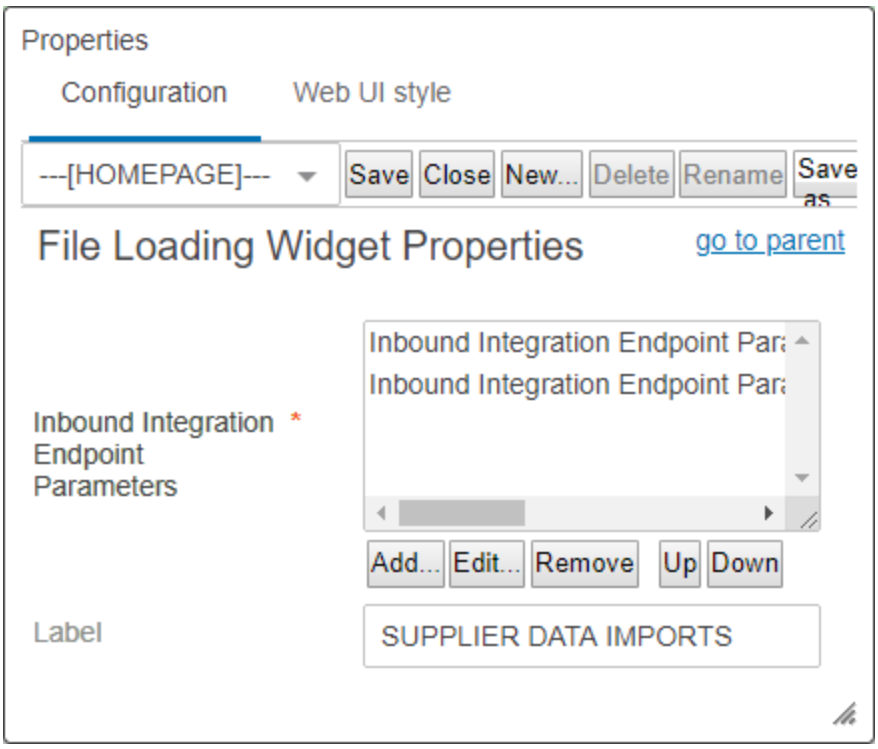
Configuration

Each screenshot example within this section provides recommended values for the parameters in ECLASS Advanced Dictionary Importer.

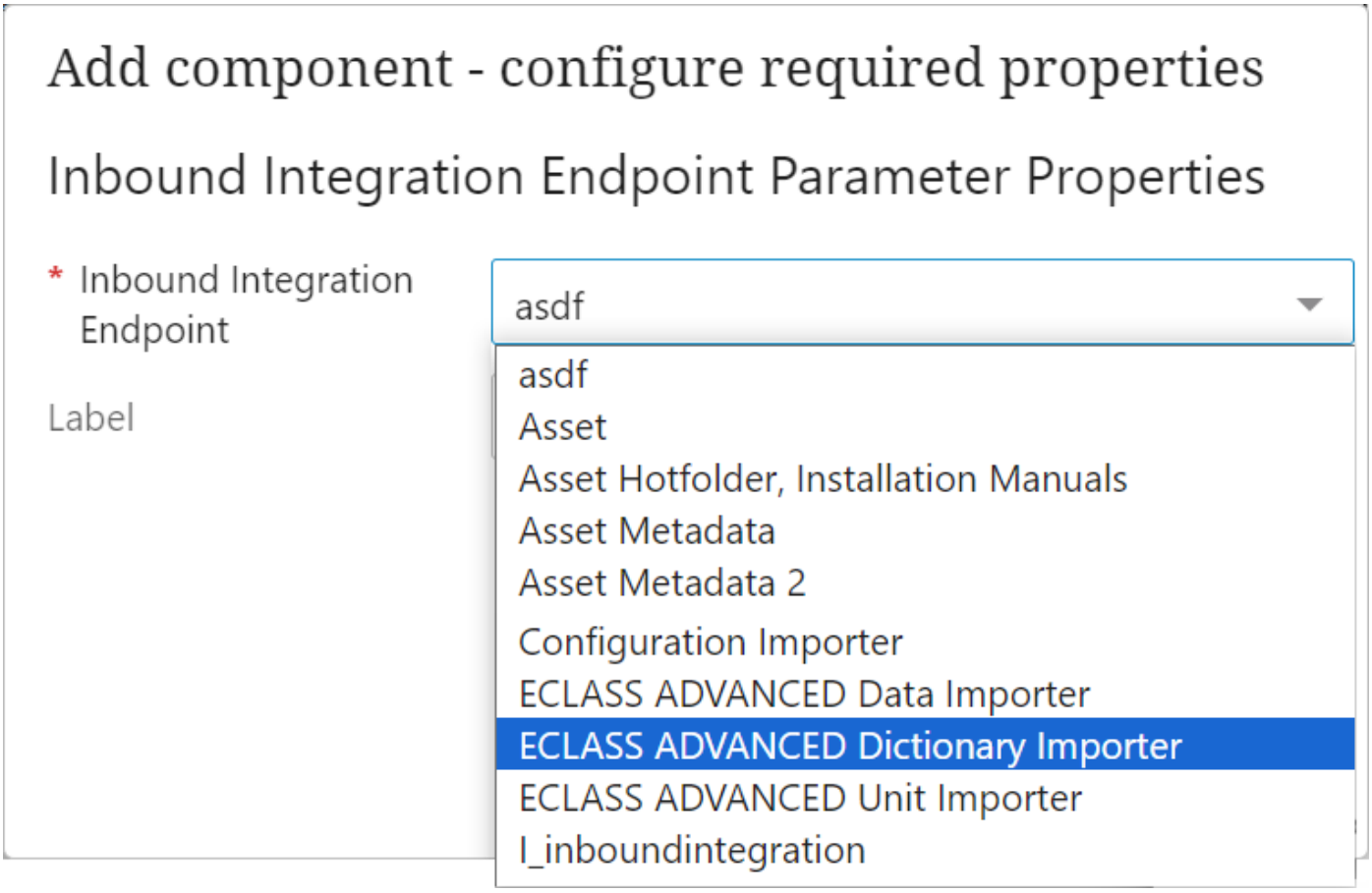
This topic describes how to configure a File Loading Widget so that users can drag and drop ECLASS Advanced Dictionary files onto a File Loading Widget on a Web UI Homepage.

1. In the designer, select an existing File Loading Widget to be used, or add a new File Loading Widget to the Homepage Widget Grid component. For more information, refer to the File Loading Widget topic within the Web User Interfaces documentation.

2. Go to the Inbound Integration Endpoint Parameters field, click the **Add** button, and the Inbound Integration Endpoint Parameter Properties dialog will display.



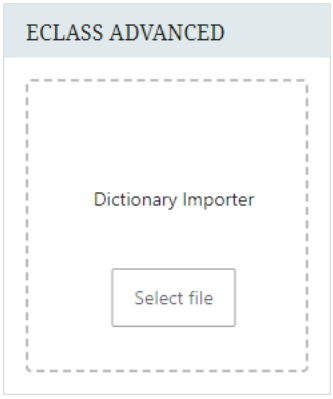
3. Click the dropdown for the Inbound Integration Endpoint parameter and select **ECLASS ADVANCED Dictionary Importer** (the IIEP created for ECLASS Advanced Dictionary imports).



Note: If the desired IIEP does not display in the dropdown, then it can be created using the steps described in the Configuring an IIEP for ECLASS Advanced Unit Imports topic.

4. Optionally, provide a label to be displayed within the drop zone of the widget.

In the example below, a File Loading Widget labeled as 'ECLASS ADVANCED' is displayed above its configurations.



5. Click the **Save** and **Close** buttons to save the changes and close the designer.

Using ECLASS Advanced Dictionary Importer

Dictionary file can be imported into STEP by uploading it to either a configured hotfolder, or through a File Loading Widget on a Web UI Homepage. The intention of the Dictionary Importer Web UI setup included within this topic is to provide an out-of-the-box solution for importing data included within a supported ECLASS Advanced Dictionary format.

Prerequisites

If you have completed the Easy Setup actions for the ECLASS ADVANCED Model, the functionalities outlined in this topic should be readily available for use. Otherwise, configuration is required. For information on how to configure ECLASS Advanced Dictionary importer, refer to Configuring ECLASS Advanced Dictionary Importer topic.

The Dictionary Importer exclusively supports XML file formats. In case the XML file is zipped, it is essential to manually extract the .zip file to access the Dictionary file.

Import Process Overview

Once a valid ECLASS Advanced file is uploaded using a File Loading Widget (or uploaded directly to a hotfolder), the file is picked up by an IIEP, and the IIEP starts a background process.

Note: The File Loading Widget permits a maximum file size of 20 MB for uploading purposes. Should the files intended for import exceed this 20 MB threshold, we recommend a direct upload to the designated hotfolder. Additionally, if the file upload process exceeds 30 seconds, the web server may time out. In such cases, it is advisable to use the direct hotfolder upload method. This functionality is managed by a configuration property that is not available in the Self-Service UI. Contact Stibo Systems Support for assistance. The following config properties setting enables uploading files larger than 20 MB by adjusting the value as HotfolderUpload.MaxFileSize=4096.

Procedure

1. Access the ECLASS Advanced Web UI Homepage.
2. Upload a valid Dictionary file to the hotfolder (root/upload/hotfolders/EclassAdvDictionaryImporter/In), or use the 'Dictionary Import' File Loading Widget.

   For information about File Loading Widget, refer to File Loading Widget topic within the Web User Interfaces documentation.

   For more information about uploading files directly to the hotfolder, refer to ECLASS Advanced Import Framework topic.
3. Once the upload has started, users can view the progress of the upload using the 'Recent background processes' side panel.

   For more information on using the 'Recent background processes' side panel, refer to Recent Background Processes Side Panel topic within the Web User Interfaces documentation.

After uploading the file, the IIEP picks it up and initiates a Background Process for the import process. The file loading widget doesn't provide background process monitoring in the Web UI. You can monitor the import status within the workbench through the IIEP Background Process that is generated.

For more information about monitoring the IIEP via background process, refer to Monitoring an IIEP via Background Process topic within the Data Exchange documentation

ECLASS Advanced Data Importer

The primary objective of the ECLASS Advanced Data Importer is to offer a convenient out-of-the-box solution for importing BMEcat 2005.1 file in a supported XML format. To ensure a successful upload, it is essential to verify that the version of the Data file you intend to upload is listed among the Supported Versions and Formats. You can confirm the file version being uploaded is listed within ECLASS Standard Supported Versions and Formats topic.

The Data Importer exclusively accepts XML file formats.

This section includes information on:

- Using ECLASS Advanced Data Importer
- Configuring ECLASS Advanced Data Importer

Configuring ECLASS Advanced Data Importer

> **Note:** If the Easy Setup actions for the ECLASS Advanced Component model have been completed, then the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

The following topics provide the configuration steps necessary to allow users to be able to drag and drop Data files onto a configured File Loading Widget and monitor the progress of the import file in the created IIEP Background Process in the workbench.

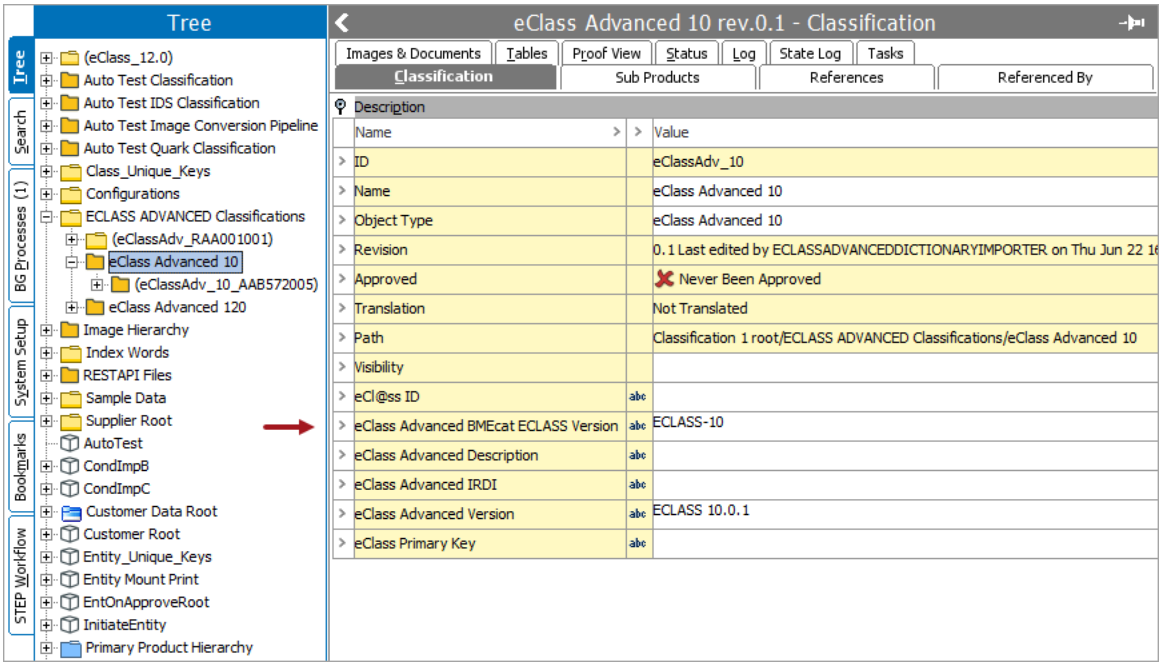
- Mapping BMEcat 2005.1 Reference Feature System with the Corresponding ECLASS Advanced Version
- Configuring an IIEP for ECLASS Advanced Data Imports
- Configuring a File Loading Widget for ECLASS Advanced Data Imports

Mapping BMEcat 2005.1 Reference Feature System with the Corresponding ECLASS Advanced Version

To successfully import any BMEcat 2005.1 file into STEP, it is essential to establish a mapping between the BMEcat 2005.1 reference feature system and the corresponding ECLASS ADVANCED version. This mapping is necessary due to the lack of alignment between the naming conventions used in BMEcat 2005.1 and the Dictionary file, specifically when indicating versions.

For example, while BMEcat 2005.1 employs the term '10.0' as its reference feature system, the ECLASS ADVANCED Dictionary refers to the version / revision as '10.0.1'. In situations like this, it becomes imperative for STEP to have a mechanism that indicates how the BMEcat data should be associated with the appropriate ECLASS ADVANCED version.

This mapping process must be applied to the metadata attribute 'eClass Advanced BMEcat ECLASS Version,' which is located on the version dependent object named 'eClassAdv_[Version]_Root.'



For the attribute, the value should be retrieved from the value available within the following BMEcat 2005.1 file tag:

```
"<REFERENCE_FEATURE_SYSTEM_NAME>xxx</REFERENCE_FEATURE_SYSTEM_NAME>"
```

For example, within the ECLASS Advanced 10.0 file, the value will be displayed as:

```
"<REFERENCE_FEATURE_SYSTEM_NAME>ECLASS-10.0</REFERENCE_FEATURE_SYSTEM_
NAME>"
```

Note: If the value is not applied, no data import takes place and the import execution report will show: "*Failed to identify a valid version number. Ensure the version is registered in the attribute 'eClassAdv_BMEcatECLASSVersion …….. The product will not be imported*"

Configuring an IIEP for ECLASS Advanced Data Imports

> **Note:** If the Easy Setup actions for the ECLASS Advanced Component model have been completed, then the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

An inbound integration endpoint (IIEP) can be configured in the workbench to help the process of importing Data (BMEcat 2005.1) files into STEP. Once an IIEP is configured for ECLASS Advanced Data imports, Data files can be imported after they are uploaded either to a configured hotfolder, or to a File Loading Widget on a Web UI Homepage. For more information, refer to ECLASS Advanced Data Importer topic.

This section describes how to configure an IIEP that can allow for the automated processing of Data files. Each screenshot example within this section provides recommended values for the parameters in ECLASS Advanced Data Importer.

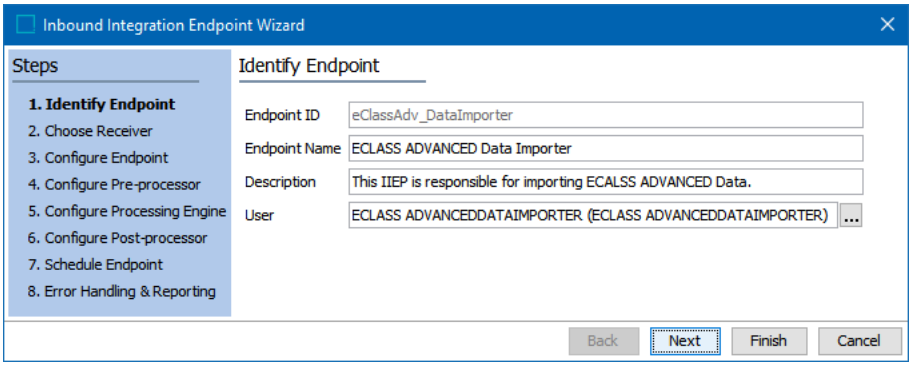
Prerequisites

This topic aims to acquaint users with the IIEP specifically designated for the import of BMEcat 2005.1 data files. It does not cover general IIEP functionalities. It is assumed that individuals configuring an IIEP for ECLASS Advanced Data Import are well-versed in configuring and processing standard inbound integration endpoints. For a comprehensive understanding of the standard functionalities provided in inbound integration endpoints, refer to Inbound Integration Endpoints topic within the Data Exchange documentation.

Configuration Steps

1. In the workbench, go to System Setup, select and right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.

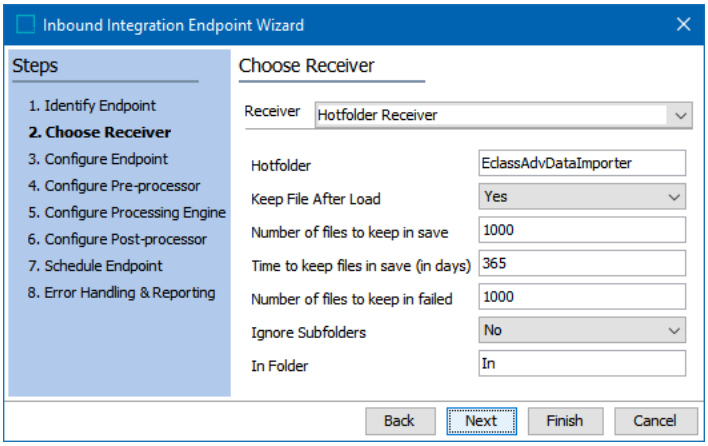
2. Once the Inbound Integration Endpoint Wizard displays, the parameters are to be populated as recommended and shown below.



For more information about the parameters available within the Identify Endpoint step, refer to IIEP - Identify Endpoint topic within the Data Exchange documentation.

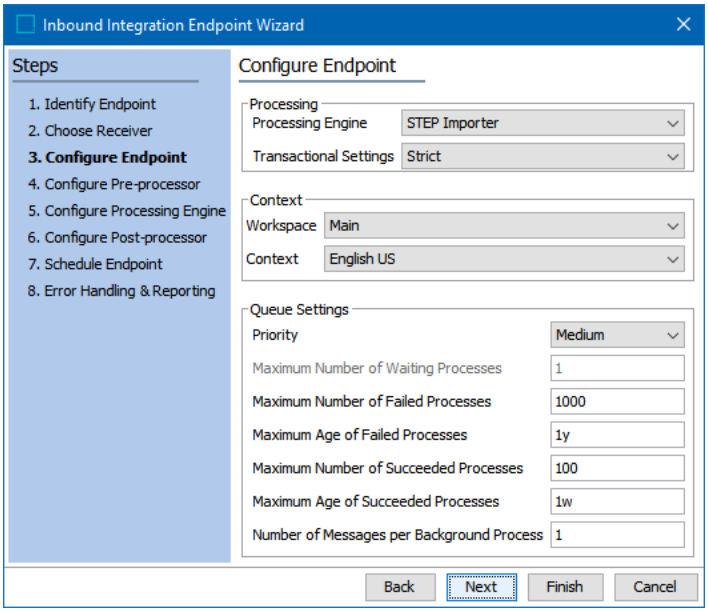
3. Click the **Next** button, and the Choose Receiver parameters will display. The parameters are to be populated as recommended and shown below. The mandatory parameter Hotfolder must be populated with a hotfolder name before the Next button will enable. In the screenshot below, the Hotfolder parameter is populated with the value 'EclassAdvDataImporter.'

> **Note:** The value within this hotfolder parameter will be used to create the new hotfolder, once the IIEP Wizard is complete.



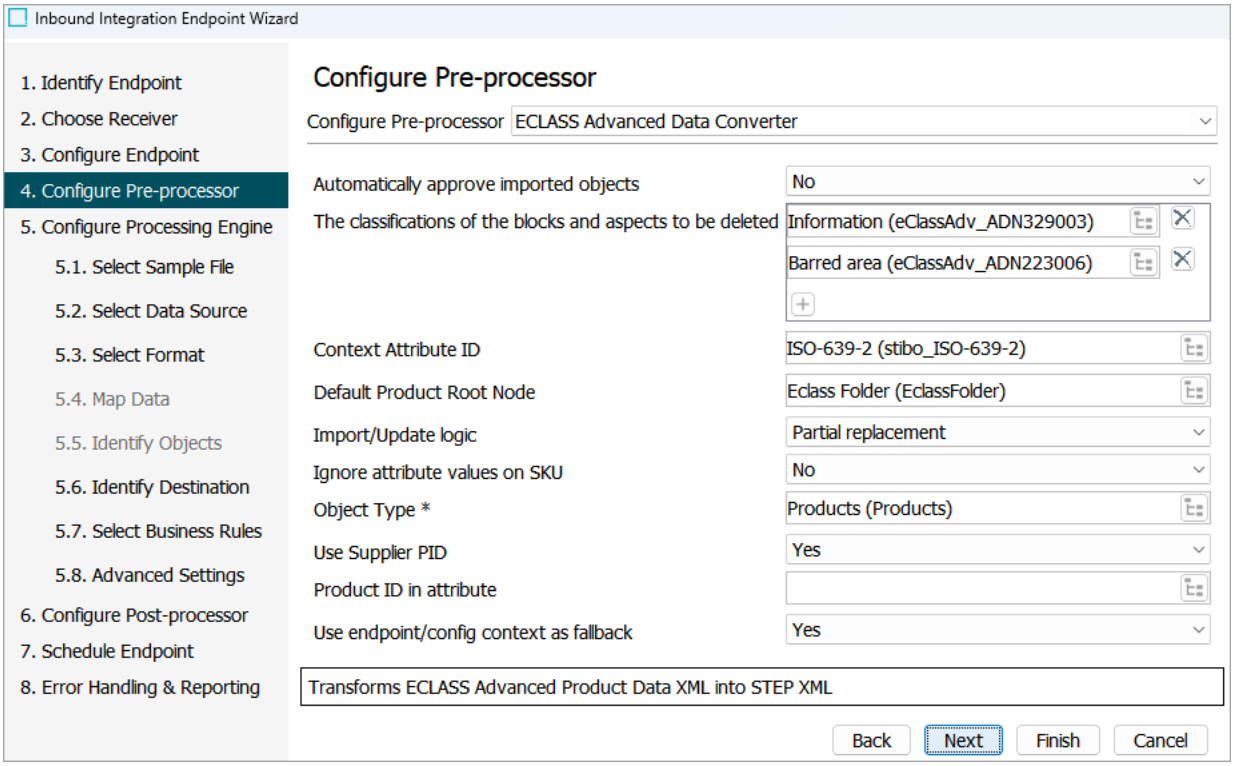
For more information about the parameters, refer to IIEP - Choose Receiver topic within the Data Exchange documentation.

4. Click the **Next** button, and the Configure Endpoint parameters will display. The parameters are to pre-populated with the recommended values as shown below.



For more information about the parameters, refer to IIEP - Configure Endpoint topic within the Data Exchange documentation.

5. Click the **Next** button, and the Configure Pre-processor parameter will display. The selection of the pre-processor within this step makes the IIEP unique for importing BMEcat 2005.1 data files. The parameters are to be populated as recommended and shown below:



- **Configure Pre-processor:** This parameter has to be populated with ECLASS Advanced Dictionary Converter option. This is an exclusive pre-processor for importing Dictionary files. For more information about the parameter, refer to IIEP - Configure Pre-processor topic within the Data Exchange documentation.
- **Automatically approve imported objects:** This parameter is for the automated approval of SKUs / Products, Product Blocks, and Aspects. When configured as 'No' (default), no approval process is initiated. However, selecting 'Yes' results in the automatic approval of all imported elements, including Product Blocks, Aspects, SKU / Product, ECLASS ADVANCED attributes, and their corresponding product root nodes.
- **Classifications of Blocks and Aspects to be Deleted:** This parameter supports a partial update deletion policy, allowing you to selectively update parts of product data from specific sources during the import process. This parameter enables you to specify exactly which classifications, blocks, or aspects should be removed or replaced, without requiring a full data deletion. By defining specific classification IDs within this parameter, you gain precise control over data being imported.

- **Context Attribute ID:** This parameter is to be populated with the attribute ISO-639-2 (ID = stibo_ISO-639-2). This attribute is created by the Easy Setup action and holds the language mappings. For more information about language mappings, refer to Prepare the Language Dimension Mapping topic.
- **Default Product Root Node:** Providing an existing product root node enables the creation of new products during the import process. These products are established beneath the specified product root node. Upon selecting a node, it's imperative to designate the corresponding object type. Leaving the field empty, however, restricts the creation of new products, permitting solely the update of existing ones.
- **Import/Update logic:** This parameter provides the following options:
    - **Full replacement:** Selecting this option leads to the prior-to-import removal of all Product Blocks and Aspects, along with the associated link to the Application Class. The actual Product / SKU remains unaltered.
    - **Reject updates on import:** If the Application Class for the same Product / SKU ID in the import file matches the Application Class referenced in STEP, the data import for the corresponding SKU is skipped. If the Application Class for the same Product / SKU in the import file does not match the Application Class referenced in STEP, the import is also skipped.
    - **Allow updates - for same Application class**: By selecting this option, the data import for the relevant SKU is updated if the Application Class in the import file matches the Application Class referenced in STEP. Conversely, if the Application Class in the import file does not correspond to the referenced Application Class in STEP, the data import for the relevant SKU is skipped.
    - **Allow updates - replace when different Application Class:** If the Application Class in the import file does not match the Application Class referenced in STEP, a full replacement of the Product / SKU takes place.
    - **Allow updates: for same Application Class with full replacement:** If the Application Class in the import file match the Application Class referenced in STEP, a full replacement of the Product / SKU takes place.
- **Ignore attribute values on SKU:** Setting this option to 'Yes' will result in the import not updating attributes directly maintained on the SKU. Conversely, when set to 'No,' the import will update attributes directly maintained on the SKU.

- **Object Type:** This mandatory parameter has to be defined with the main SKU / Product Object Type. The specified object type must be from the one for which the ECLASS Advanced attributes were deemed valid. Meaning, this is the same object type configured during the execution of the Easy Setup action. For information about the SKU / Product Object type, refer to the Run Easy Setup of ECLASS Advanced Industry Standard topic.

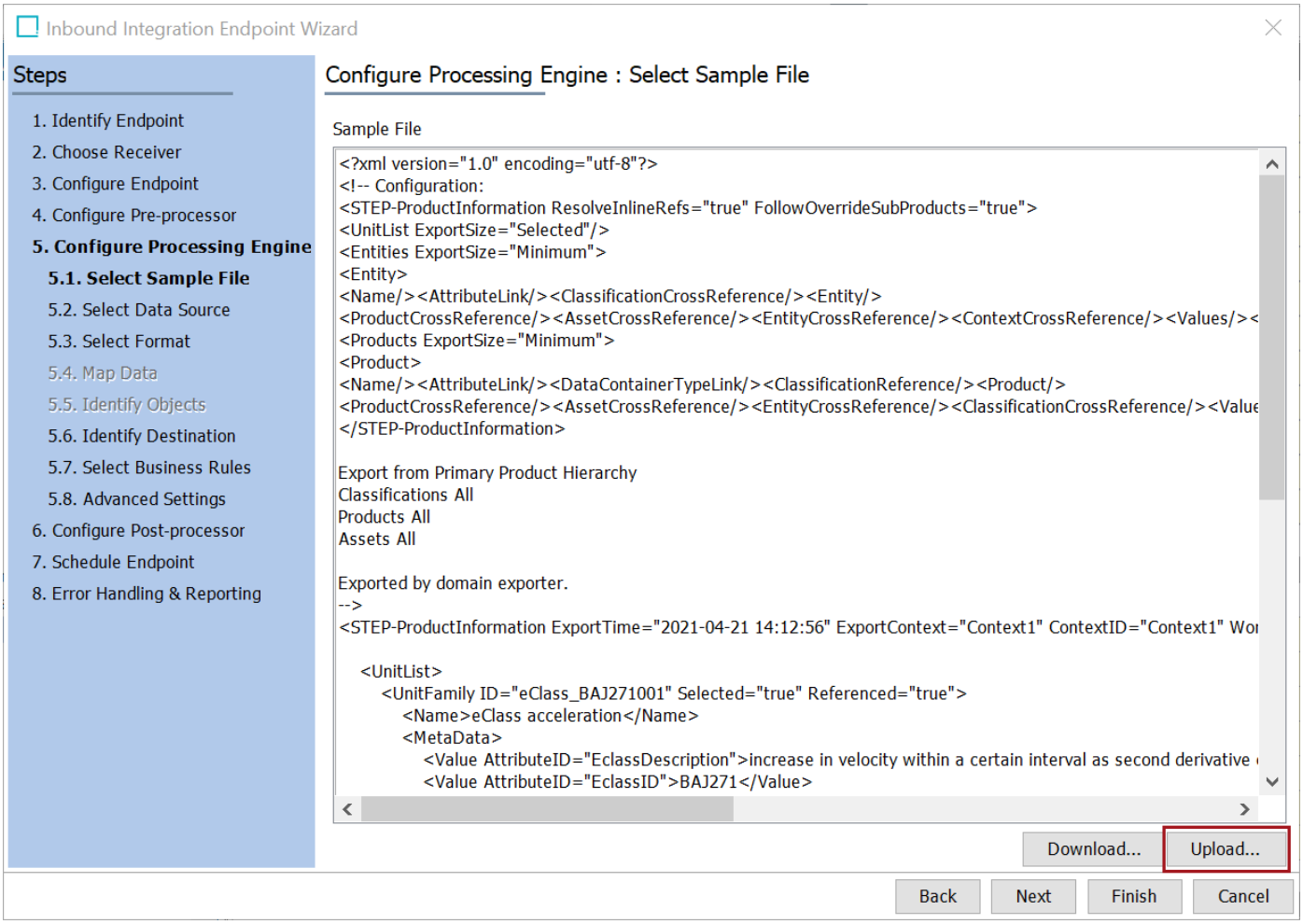
  > **Note:** This parameter is not automatically populated by the Easy Setup action, given that the solution may encompass multiple SKU / Product object types that are defined during the Easy Setup process.

- **Use Supplier PID:** When identifying the SKU through the 'SUPPLIER_PID,' opt for Yes (default). Alternatively, when using the 'MANUFACTURER_PID' for SKU identification, select 'No.'
- **Product ID in attribute:** This parameter facilitates the updating of existing SKUs in the system. If the SKU ID is specified within an attribute, then that attribute can be defined within this parameter. The uniqueness of the ID has to be taken into account while defining such attributes. In the event this field remains blank, the identifier will be automatically set to the Step ID.
- **Use endpoint/config context as fallback:** The option selected within this parameter (Yes or No) reacts based on a combination of factors including:
  - Language declaration as part of the BMEcat 2005.1 header
  - The language tag provided or not provided as part of the actual translatable free text value
  - The STEP Context configured in the Inbound Integration Endpoint (IIEP)
  - The actual language mapping metadata attribute ISO-639-2 (ID = stibo_ISO-639-2) to the Language Dimension Points

  If for some reason free text value translations cannot be identified or are missing in the import file, by setting the value to 'Yes' will ensure that the relevant values are imported into the STEP Context configured in the IIEP (fallback context).

  Setting the parameter to 'No' will require correctly placed language tags.

6. Click the **Next** button, and the 'Configure Processing Engine: Select Sample File' field for the STEP Importer processing engine will display.



- In **5.1 Select Sample File** step, click the Upload button to upload a sample STEPXML file. For information about how to upload a sample file, refer to IIEP - Configure STEP Importer Processing Engine topic within the Data Exchange documentation.

  The basic data structure of a sample Data file is provided below:

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

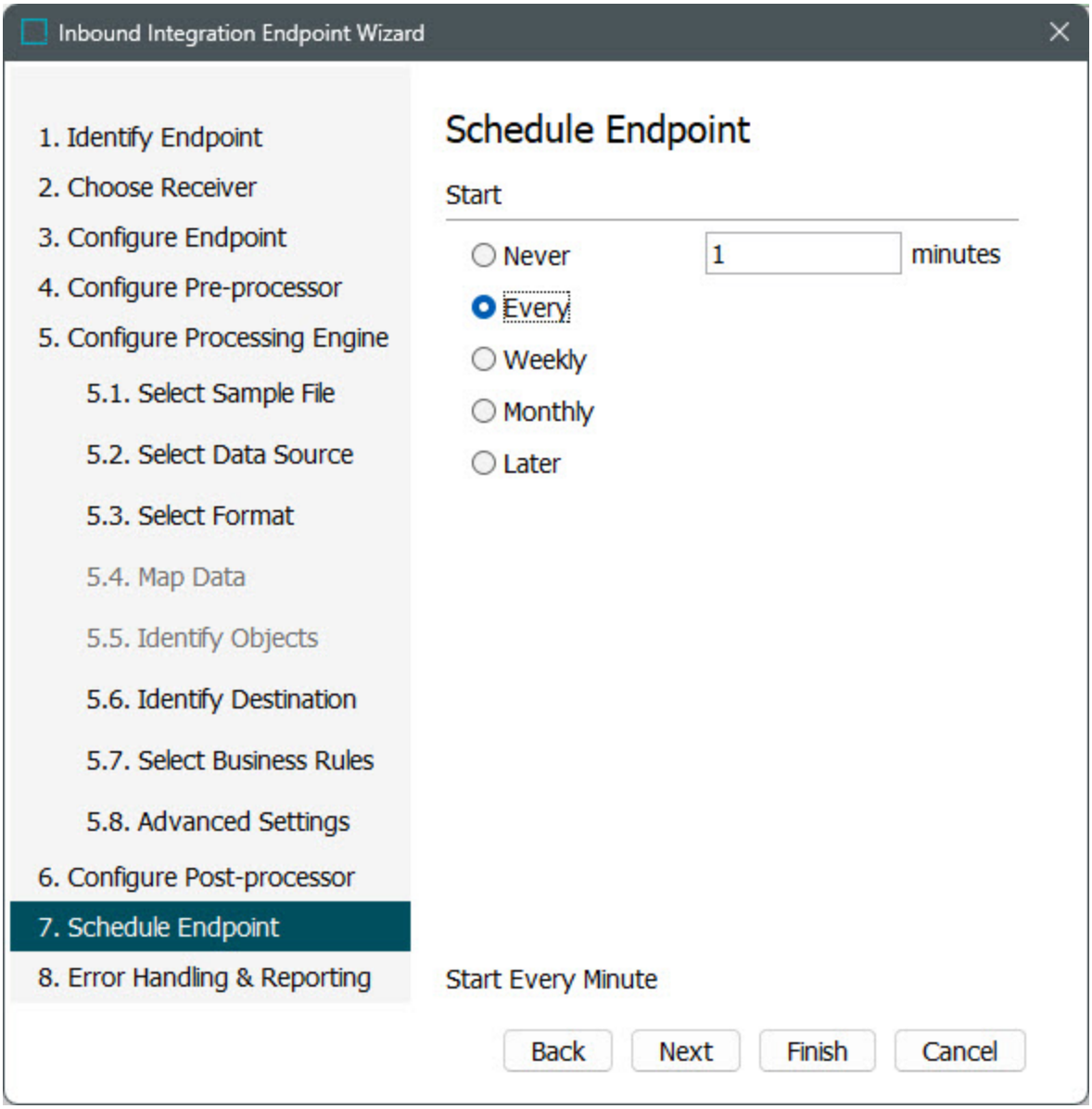
```
<Entities ExportSize="Minimum">
<Entity>
<Name/><AttributeLink/><ClassificationCrossReference/><Entity/>

<ProductCrossReference/><AssetCrossReference/><EntityCrossReference/>
<ContextCrossReference/><Values/></Entity></Entities>
<Products ExportSize="Minimum">
<Product>

<Name/><AttributeLink/><DataContainerTypeLink/><ClassificationReferen
ce/><Product/>

<ProductCrossReference/><AssetCrossReference/><EntityCrossReference/>
<ClassificationCrossReference/><Values/><OverrideSubProduct/></Produc
t></Products>
</STEP-ProductInformation>

Export from Primary Product Hierarchy
Classifications All
Products All
Assets All

Exported by domain exporter.
-->
<STEP-ProductInformation ExportTime="2021-04-21 14:12:56"
ExportContext="Context1" ContextID="Context1" WorkspaceID="Main"
UseContextLocale="false">

    <UnitList>
        <UnitFamily ID="eClass_BAJ271001" Selected="true"
Referenced="true">
            <Name>eClass acceleration</Name>
            <MetaData>
                <Value AttributeID="EclassDescription">increase in
```

```
velocity within a certain interval as second derivative of the
distance per time</Value>
                <Value AttributeID="EclassID">BAJ271</Value>
                <Value
AttributeID="EclassPrimaryKey">BAJ271001</Value>
            </MetaData>
            <Unit ID="eClass_AAA225002" Selected="true"
Referenced="true">
                <Name>ft/sÂ²</Name>
                <MetaData>
                    <Value AttributeID="EclassDescription">unit foot
according to the Anglo-American and the Imperial system of units
divided by the power of the SI base unit second with the exponent 2
with the relation according to NIST: 1 ft/sÂ² = 0,304 8 m/sÂ²</Value>
                    <Value
AttributeID="EclassImportVersion">10.1</Value>
                    <Value AttributeID="EclassID">AAA225</Value>
                    <Value
AttributeID="EclassPrimaryKey">AAA225002</Value>
                </MetaData>
            </Unit>
            <Unit ID="eClass_AAA597002" Selected="true"
Referenced="true">
                <Name>m/sÂ²</Name>
                <MetaData>
                    <Value AttributeID="EclassDescription">SI base
unit metre divided by the power of the SI base unit second and the
exponent 2</Value>
                    <Value
AttributeID="EclassImportVersion">10.1</Value>
                    <Value AttributeID="EclassID">AAA597</Value>
                    <Value
AttributeID="EclassPrimaryKey">AAA597002</Value>
                </MetaData>
```

```
            </Unit>
        </UnitFamily>
    </UnitList>
</STEP-ProductInformation>
```

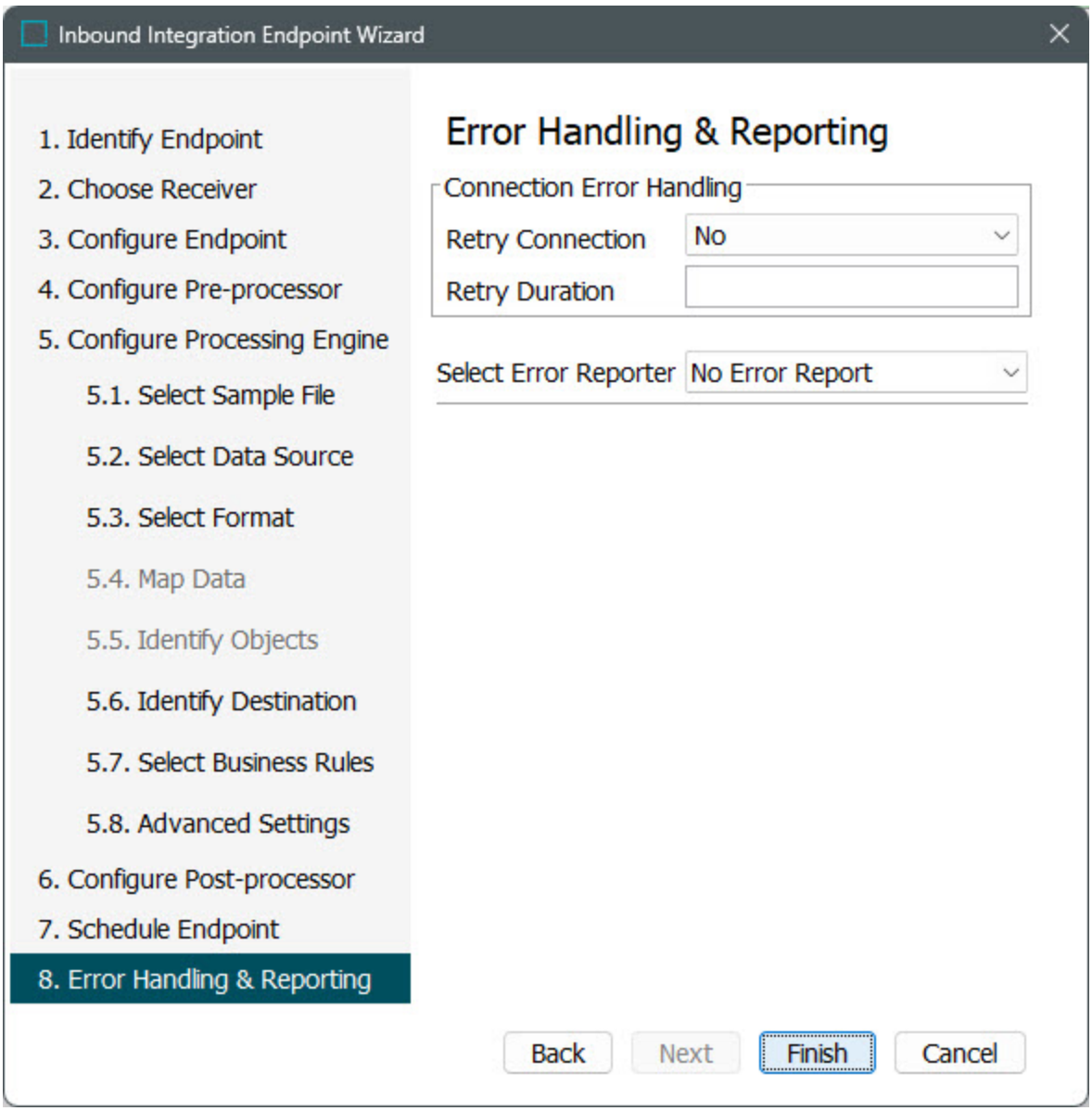
- Further potential sub-steps underneath the Configure Processing Engine are optional. For more information about these steps, refer to IIEP - Configure STEP Importer Processing Engine topic within the Data Exchange documentation.

7. Click the **Next** button to display the Schedule Endpoint parameters (bypass the Configure Post-processor parameters). Update the values to those shown below.



For information about the parameters available within this step, refer to IIEP - Schedule Endpoint topic within the Data Exchange documentation.

8. Click the **Next** button, and the Error Handling & Reporting step will display. The parameters are to be populated as recommended and shown below:



For more information about the parameters available within this step, refer to IIEP - Error Handling & Reporting topic within the Data Exchange documentation.

9. Click the **Finish** button, the Inbound Integration Endpoint Wizard will close, and the newly created endpoint will display within the workbench.

> ⚠ **Important:** An endpoint must be enabled before it can start processing data. For more information, refer to the Running an Inbound Integration Endpoint topic within the Data Exchange documentation.

If users need to access the IIEP via a Web UI, then the IIEP must be configured within a File Loading Widget. For more information, refer to the Configuring a File Loading Widget for ECLASS Advanced Data Imports topic.

Configuring a File Loading Widget for ECLASS Advanced Data Imports

Web UI users can import Data files into STEP using a File Loading Widget. If Easy Setup actions for the ECLASS Advanced solution have been completed as described in 2. Run Easy Setup of ECLASS ADVANCED Industry Standard topic of the **ECLASS Quick Start Guide**, then the 'Data Importer' File Loading Widget will automatically be added to the Web UI Homepage as shown in the examples below. Otherwise, the steps below can be used to complete configuration. This topic describes how to configure a File Loading Widget so that users can drag and drop ECLASS Advanced Data files onto a File Loading Widget on a Web UI Homepage.

Prerequisites

Before configuring the Web UI portion of this solution, an IIEP for an ECLASS Advanced Data Importer must be configured within the workbench. For more information, refer to Configuring an IIEP for ECLASS Advanced Data Imports topic.

It is expected that anyone configuring the ECLASS Advanced Data Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, refer to Designer Access topic within the Web User Interfaces documentation.

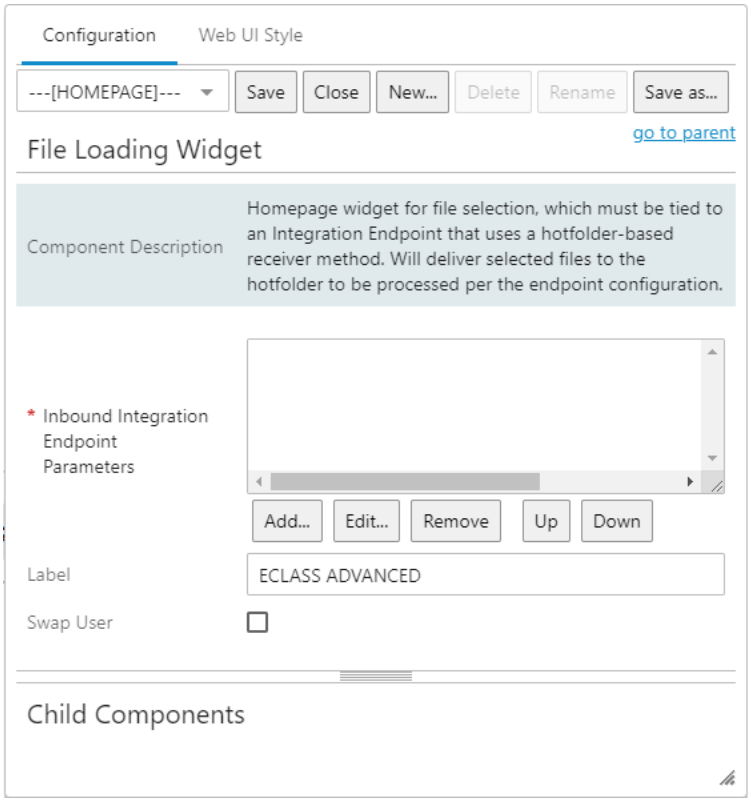
Additionally, it is helpful to know how to add a widget to a Web UI Homepage. Details on how to do this can be found in Adding Widgets to a Homepage topic in the Web User Interfaces documentation.

Configuration

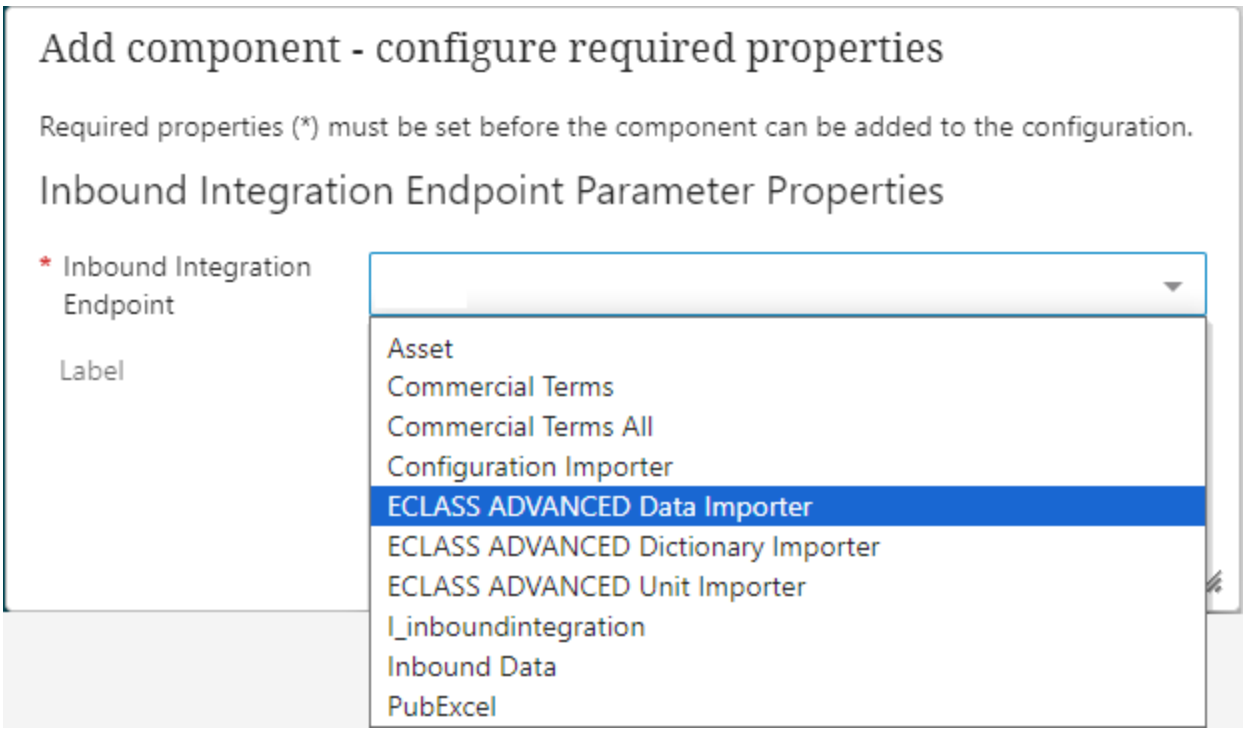
Each screenshot example within this section provides recommended values for the parameters in ECLASS Advanced Data Importer.

1. In the designer, select an existing File Loading Widget to be used, or add a new File Loading Widget to the Homepage Widget Grid component. For more information, refer to File Loading Widget topic within the Web User Interfaces documentation.

2. Go to the Inbound Integration Endpoint Parameters field, click the **Add** button, and the Inbound Integration Endpoint Parameter Properties dialog will display.



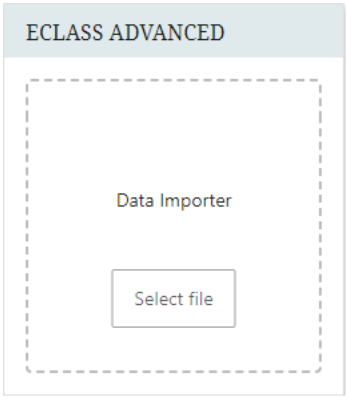
3. Click the dropdown for the Inbound Integration Endpoint parameter, and select **ECLASS ADVANCED Data Importer** (the IIEP created for ECLASS Advanced Data imports).



Note: If the desired IIEP does not display in the dropdown, then it can be created using the steps described in Configuring an IIEP for ECLASS Advanced Data Imports topic.

4. Optionally, provide a label to be displayed within the drop zone of the widget.

   In the example below, a File Loading Widget labeled as 'ECLASS ADVANCED' is displayed above its configurations.



5. Click the **Save** and **Close** buttons to save the changes and close the designer.

Using ECLASS Advanced Data Importer

ECLASS Advanced Data file can be imported into STEP by uploading it to either a configured hotfolder, or through a File Loading Widget on a Web UI Homepage. The purpose of the Data Importer Web UI setup outlined in this topic is to offer a readily available solution for importing data in a supported ECLASS Advanced Data format. Only the BMEcat 2005.1 data is allowed to be imported as the ECLASS Advanced Data file.

Because only supported versions will successfully upload, before attempting to upload a Data file, confirm the file version being uploaded is listed within the ECLASS Standard Supported Versions and Formats topic.

Considerations before initiating an import

Before importing a BMEcat 2005.1 file into the system, it is important to consider the following:

- BMEcat 2005.1 is purposefully designed for transporting ECLASS Advanced data, making it exclusively compatible with ECLASS Advanced.
- BMEcat data corresponds to specific ECLASS Advanced version(s), requiring the prior import of the corresponding / matching ECLASS Advanced version before commencing the BMEcat data import.
- The BMEcat 2005.1 Importer exclusively handles ECLASS Advanced data. Standard BMEcat (1.2 or 2005) fields will not undergo import.
- Unlike the ECLASS Advanced Dictionary file, which is language-specific, the BMEcat 2005.1 file includes multiple languages. Therefore, it's imperative to execute Language Mapping before initiating the import of BMEcat 2005.1 data.

Prerequisites

The Data Importer exclusively supports BMEcat 2005.1 file format. In case the XML file is zipped, it is essential to manually extract the .zip file to access the Data file.

Before engaging in any BMEcat 2005.1 import scenarios, it is essential to import the related ECLASS Advanced unit and ECLASS Advanced Dictionary file into the system.

The BMEcat 2005.1 Reference Feature System should be mapped with the corresponding ECLASS Advanced Version. For information about how to map BMEcat 2005.1 Reference Feature System with the Corresponding ECLASS Advanced Version, refer to Mapping BMEcat 2005.1 Reference Feature System with the Corresponding ECLASS Advanced Version topic available within the ECLASS Advanced Quick Start Guide.

Language Mapping has to be established. For information about how to perform Language Mapping, refer to Prepare the Language Dimension Mapping topic available within the ECLASS Advanced Quick Start Guide.

If you have completed the Easy Setup actions for the ECLASS Advanced Model, the functionalities outlined in this topic should be readily available for use. Unlike other ECLASS Advanced importers, the IIEP used for ECLASS Advanced Data Importer requires some manual configuration before using it for the first time. For information about updating the newly Easy Setup created IIEP, refer to Update IIEPs topic available within the ECLASS Advanced Quick Start Guide. For a detailed information on how to configure ECLASS Advanced Data importer, refer to Configuring ECLASS Advanced Data Importer topic.

Import Process Overview

Once a valid ECLASS Advanced file is uploaded using a File Loading Widget (or uploaded directly to a hotfolder), the file is picked up by an IIEP, and the IIEP starts a background process.

Procedure

1. Access the ECLASS Advanced Web UI Homepage.
2. Drag and drop a valid Data file into the 'Data Import' File Loading Widget, or upload to the hotfolder (root/upload/hotfolders/EclassAdvDataImporter/In).

   For information about File Loading Widget, refer to File Loading Widget topic within the Web User Interfaces documentation.

   For more information about uploading files directly to the hotfolder, refer to ECLASS Advanced Import Framework topic.

3. Once the upload has started, users can view the progress of the upload using the 'Recent background processes' side panel.

   For more information on using the 'Recent background processes' side panel, refer to Recent Background Processes Side Panel topic within the Web User Interfaces documentation.

After uploading the file, the IIEP picks it up and initiates a Background Process for the import process. The file loading widget does not provide background process monitoring in the Web UI. You can monitor the import status within the workbench through the IIEP Background Process that is generated.

For more information about monitoring the IIEP via background process, refer to Monitoring an IIEP via Background Process topic within the Data Exchange documentation.

Exporting in BMEcat 2005.1 Format

Exporting the BMEcat 2005.1 format involves supplying header data and catalog / product data via the parameters displayed on the Select Format step and during the Map Data step. Some information is required, as is indicated on the Map Data step.

> **Note:** While it is possible to initiate the export process even in the absence of mandatory fields, it's important to note that the background process (BGP) will fail.

Format Availability

BMEcat 2005.1 XML can be exported from:

- Export Manager - refer to Creating a Data Export
- OIEP - refer to Creating an Outbound Integration Endpoint

Mapping

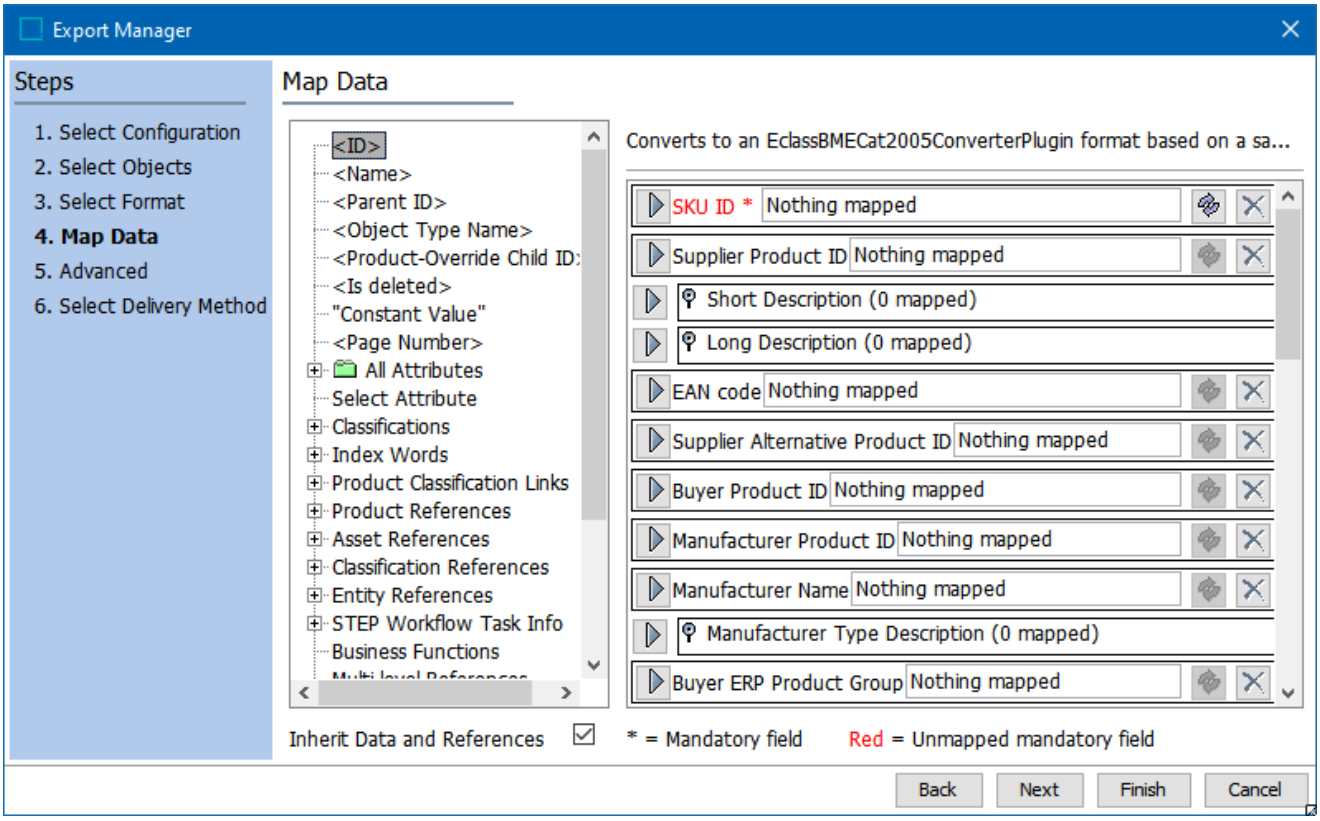
This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For further details on mapping, refer to Data Mapping.

Selecting the BMEcat 2005.1 Export format will automatically generate the data within the <PRODUCT_FEATURES></PRODUCT_FEATURES> Tag with the in STEP available ECLASS Advanced content. The extension plug-in will produce the needed information for the output.

Standard BMEcat fields can be mapped with the usual mapping process. For more information on mapping, refer to the Outbound Map Data Options topic in the Data Exchange documentation.

To map for ECLASS Advanced, the STEP ID of the SKU / Product must be mapped to the mandatory field 'SKU ID.'

Should other relevant fields need to be mapped, they can be mapped as needed.



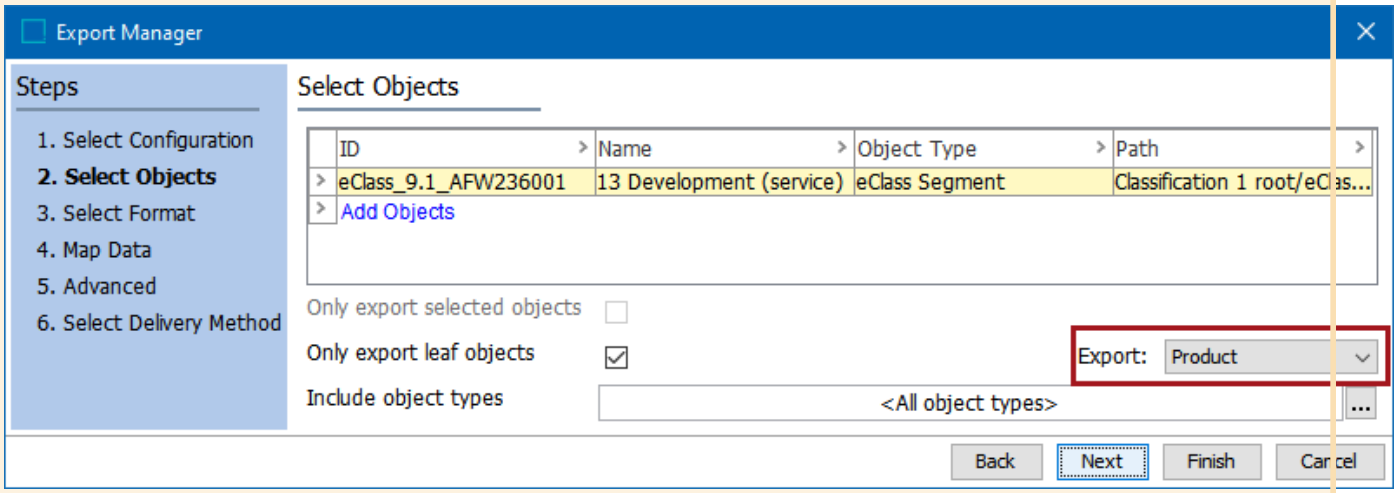
Configuring BMEcat Exporter

The technical starting point for the Exporter is always the SKU / Product. Therefore, the user should trigger the exporter from either the SKU / Product or a relevant ECLASS Advanced Classification node from the top ECLASS Advanced level downwards.

Note: Only mapped languages are determined for output.

1. In the workbench, initiate the export for the product(s) you intend to export. For information on how to initiate an export refer to the Creating a Data Export topic in the Data Exchange documentation.

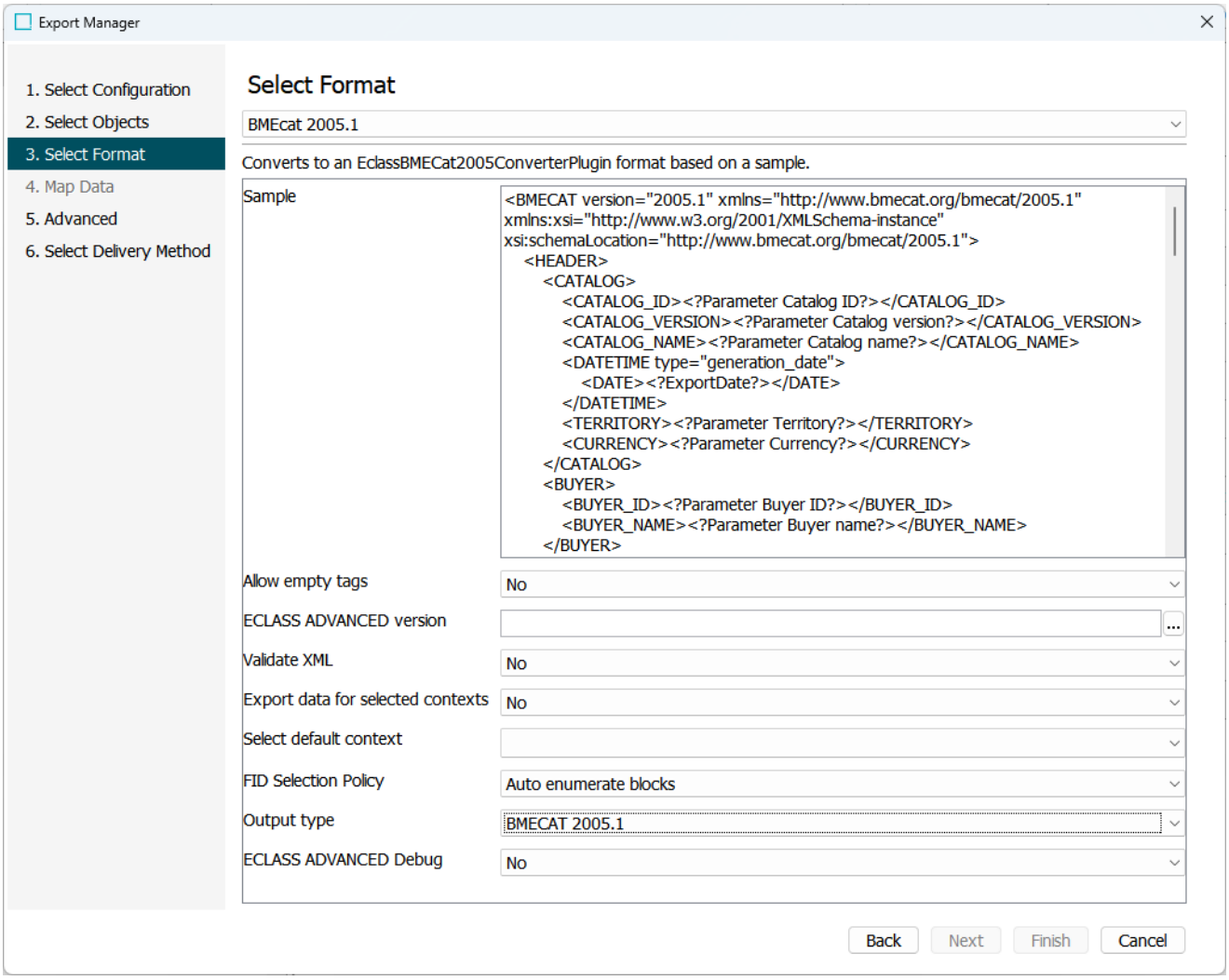
Important: When exporting from a Classification node, in the Export dropdown select 'Product.'



2. On the Select Format step of the Export Manager, fill out the following Outbound Parameters.

Outbound Parameters

The parameters that are available to fill out are controlled by the XML template that is uploaded. The fields listed below are displayed for all users.



- **Sample**: An XML template file is loaded automatically but can be modified as required. This defines the format of the XML file to be exported.

> **Note:** For more information on Generic XML instructions, refer to the Generic XML Outbound Processing Instructions topic in the Data Exchange documentation.

- **Allow empty tags**: Selecting **Yes** indicates that export tags with empty values are included in the output. If set to **No**, tags containing empty values are not included in the export.
- **ECLASS ADVANCED Version**: This field is mandatory. If no version is selected, then there will be no SKU / Products exported. This field specifies the root classification node of the version that should be exported. To select the relevant ECLASS Advanced Version, click the eclipses button, and navigate to the relevant version

in the ECLASS Advanced structure.

- **Validate XML**: Selecting **Yes** indicates that the BMEcat 2005.1 export file is validated against an XSD included in STEP. When validation fails, the background process also fails, and the problem is reported in the BGP execution report. If set to **No**, the validation is skipped and the BGP does not fail due to differences found when comparing to the XSD.
- **Export data for selected contexts**: This allows for the output values for language dependent attributes, for example, the ETIM Short Description, Long Description, Keyword, and Remark values, and others. Select **Yes** in the dropdown to display the **Select Contexts** link. Click the link, select the required contexts for the export from the Select Contexts dialog, and click the **Select** button. The selected contexts are listed in a text box.

  The chosen contexts will be presented as <LANGUAGE> elements within the <HEADER> section of the exported XML file. If the system contains the specified context-dependent attribute values, they will be incorporated into the <PRODUCT_DETAILS> segment.

  > **Important:** When using an OIEP for BMECat format export, ensure consistency between the contexts specified in the 'Export data for selected contexts' parameter and OIEP > Configuration tab > Configuration flipper > Contexts parameter. A mismatch may lead to contexts being presented differently in the exported file: those added in OIEP > Configuration tab > Configuration flipper > Contexts parameter will appear in the <PRODUCT_DETAILS> element, and those in 'Export data for selected contexts' parameter will be in the <HEADER> element.

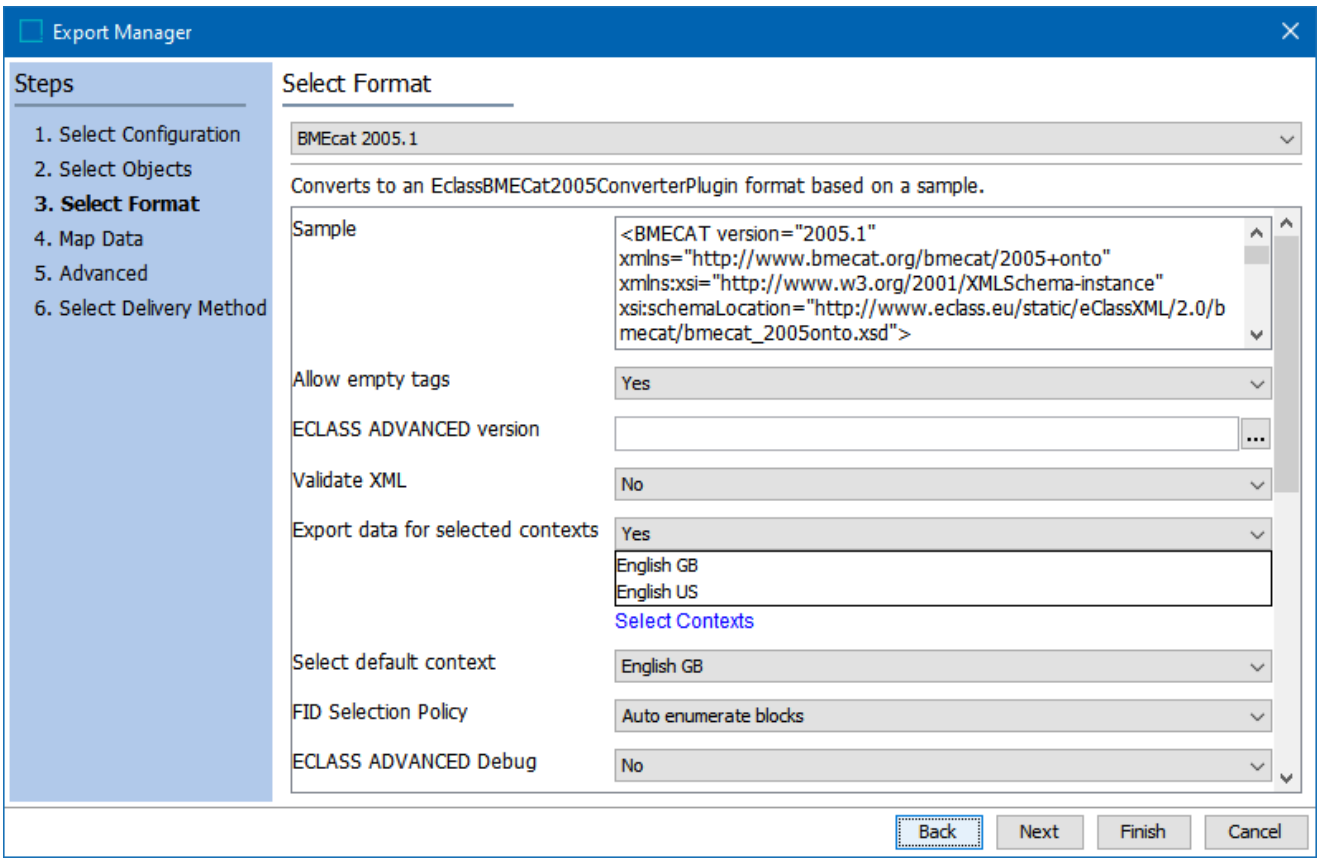
Select default context: This shows the default context that is selected for the output. The parameter only displays contexts that were selected in the **Export data for selected contexts** parameter, and automatically selects the first context from the list. However, this can be changed by clicking the dropdown menu and selecting a different option.

> **Note:** For exports done in the BMEcat file format, all cross-context exports, by default, always contain suppressed references. However, to omit suppressed references, the following configuration property must be added: `ExportManager.Omit.SuppressedReferences.CrossContext=true` This configuration property is not available in the Self-Service UI. Contact Stibo Systems Support for assistance.

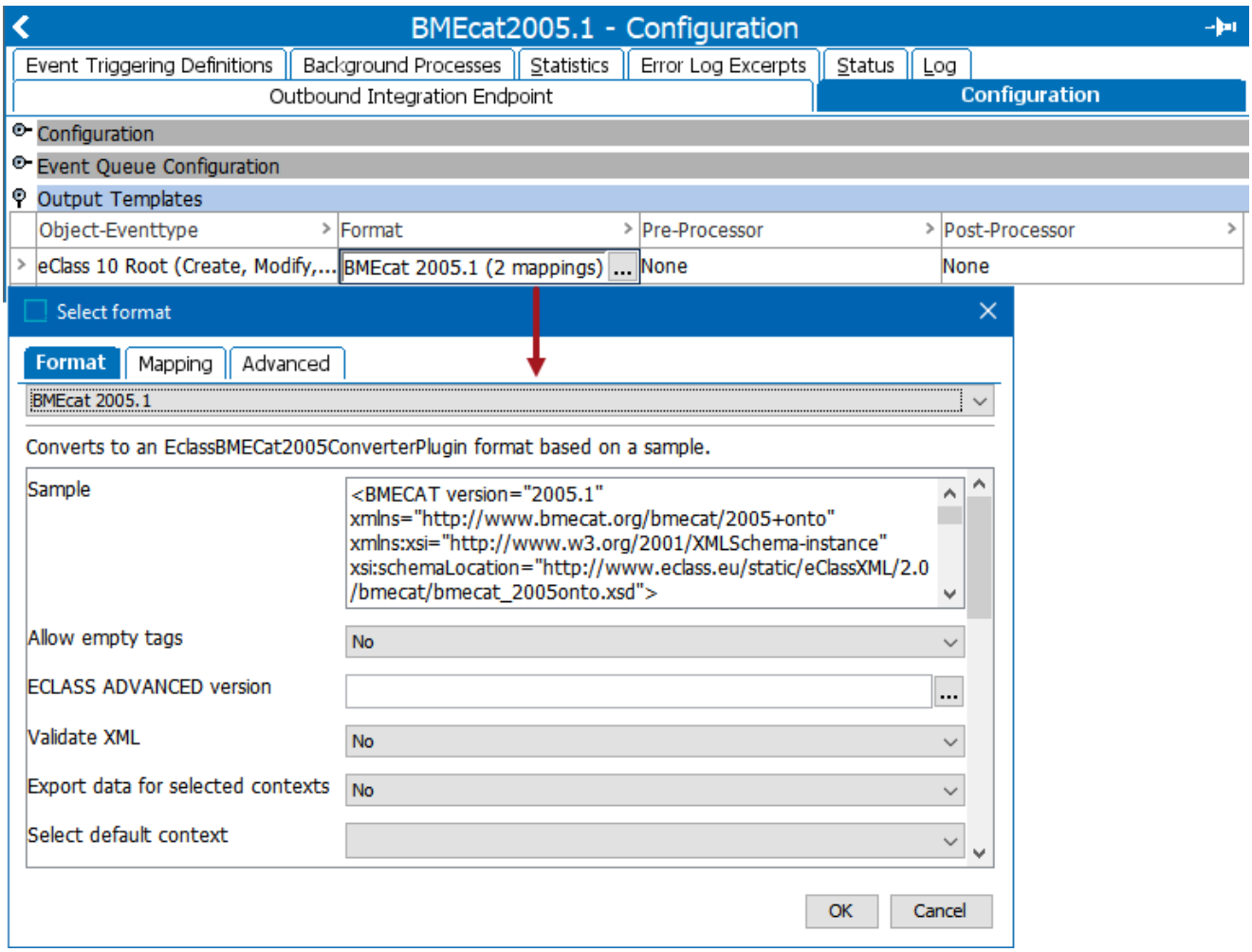
- **FID Selection Policy**: The FID in STEP remains in contrast to the new FID values generated on export. The options are:
  - **Auto enumerate blocks**: When selected, the FID of product blocks stored in STEP will be output with a continuous enumeration starting at "0."
  - **Warn and skip element and hierarchy**: When selected, product blocks, where metadata attributes have no value, will not be part of the output. All other directly related Product Blocks will consequently also not be part of the output. All other Product Blocks will be output with the FID from STEP. The execution report will list the 'leading' product block with missing FID values.
  - **Warn and auto enumerate block**: When selected, product blocks, were metadata attributes have no value, will appear in the output with an auto FID as described. All other product blocks will be output with the FID from STEP. The execution report will list the product blocks with missing FID values.
- **Output type**: This allows to select the version of the BMEcat file. The current options include, BMECat 2005.1 and BMEcat 2005.2. Switching between the output types will change the sample file populated in the Sample parameter above.
- **ECLASS ADVANCED Debug**: When selected, the output will show the respective STEP ID of the 'Features' in the 'StepID' tag. The StepID tag is intended for easier internal output debug investigation

For an explanation of the remaining parameters, search the web. No validation is performed on the text entered or the selections made, but if Validate XML = Yes, errors will be reported in the execution report, as defined above.

Export Manager


Export Manager
Steps
1. Select Configuration
2. Select Objects
3. Select Format
4. Map Data
5. Advanced
6. Select Delivery Method
Select Format
BMEcat 2005.1
Converts to an EclassBMECat2005ConverterPlugin format based on a sample.
Sample
<BMECAT version="2005.1"
xmlns="http://www.bmecat.org/bmecat/2005+onto"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.eclass.eu/static/eClassXML/2.0/b
mecat/bmecat_2005onto.xsd">
Allow empty tags
Yes
ECLASS ADVANCED version
Validate XML
No
Export data for selected contexts
Yes
English GB
English US
Select Contexts
Select default context
English GB
FID Selection Policy
Auto enumerate blocks
ECLASS ADVANCED Debug
No
Back
Next
Finish
Cancel

OIEP



Note: When using an OIEP for BMECat format export, there is a subtle distinction in how contexts are filled in the <HEADER> and <PRODUCT_DETAILS> elements. Specifically, for the context-dependent attribute values populated in the <PRODUCT_DETAILS> element, the values set in OIEP > Configuration tab > Configuration flipper > Contexts parameter take precedence and are populated accordingly. However, the contexts presented as <LANGUAGE> elements within the <HEADER> section are determined by the settings made in the above mentioned 'Export data for selected contexts' parameter itself.

ECLASS Advanced Actions

The ECLASS Advanced solution offers multiple unique functionalities that could be executed in the system. The topics listed below this section explain those functionalities that are exclusively applicable for the ECLASS Advanced solution. Following are the ECLASS Advanced Actions:

- Approve version Action
- Delete version Action
- Duplicate version Action
- Validate version Action

Approve version Action

The 'Approve version' action initiates an approval process for the selected SKU/Product and its associated ECLASS Advanced data.

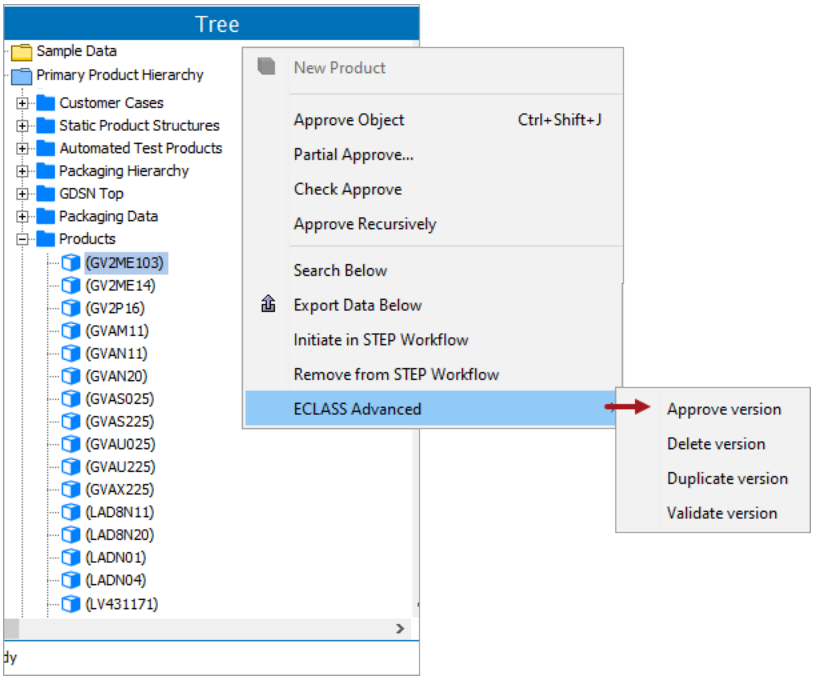
The approval encompasses the following actions:

- Approval of all Product Blocks and Aspects along with their respective root nodes for the selected version(s)
- Partial approval of the following elements of the SKU / Product:
  - STEP Name
  - Relevant ECLASS Advanced classification reference(s)
  - Relevant ECLASS Advanced attributes

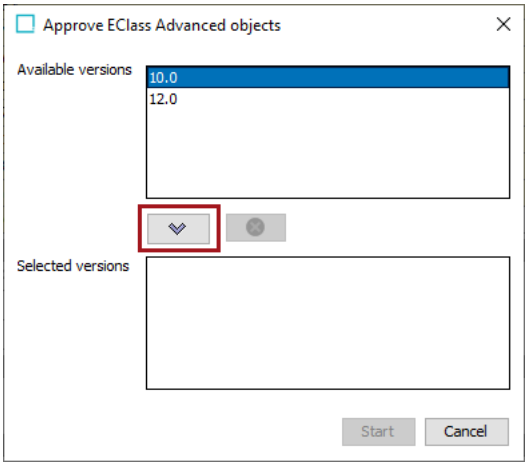
Procedure to approve versions for an SKU / Product in the workbench

Below are the steps to approve an SKU / Product:

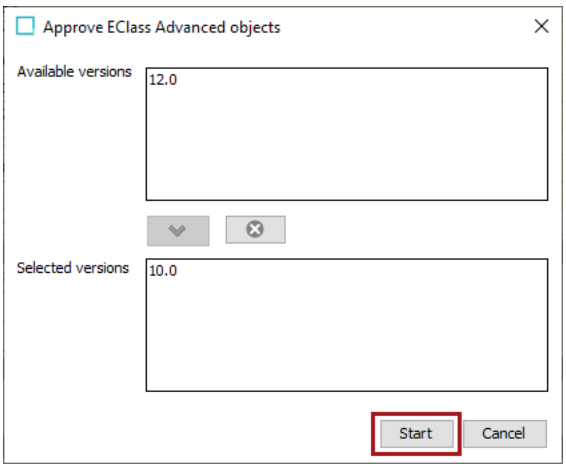
1. Navigate to the Tree tab and select the desired object.
2. Right-click and hover the cursor over ECLASS Advanced menu, and then select 'Approve version' option.



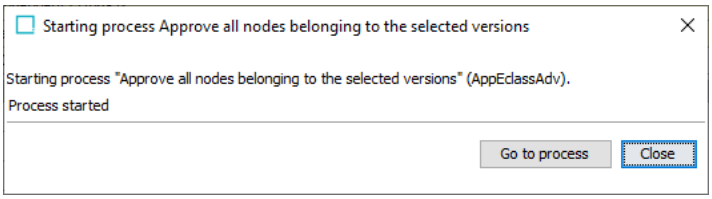
3. In the 'Approve ECLASS Advanced objects' dialog that is displayed, select the version(s) to be approved and click the Down icon ( ).



4. After the desired version is being populated in the 'Selected versions' field, click the 'Start' button to initiate the approval process. In the example below, ECLASS Advanced data for version 10.0 is to be approved.



5. The approval action is run as a background process, and you can access the Background Process information through the execution report. In the dialog labeled 'Starting process Approve all nodes belonging to the selected versions,' click the 'Go to process' button to navigate to the actual background process.



The report provides error details if any issues are encountered. In case of a successful process or approval, the report logs product information. Consequently, the report displays no errors and no warnings when the process is error-free and successful.

Delete version Action

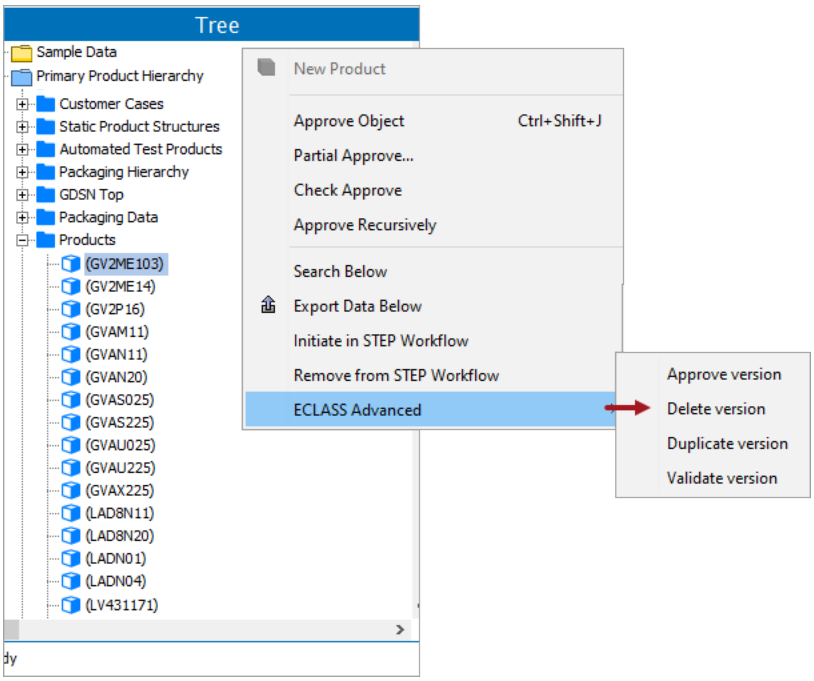
This functionality is designed to remove ECLASS ADVANCED data from specific versions within STEP. This functionality is exclusively accessible when working within the workbench environment.

The deletion process eliminates all ECLASS Advanced data, including referenced Product Blocks and Product Aspects along with their immediate parent nodes. In addition, the Application Class reference is also removed for the selected version(s). The values associated with the actual SKU/Product remain unchanged.

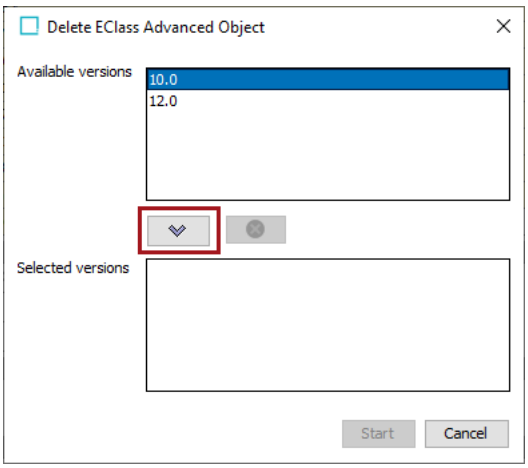
Procedure to remove version from SKU / Product

Below are the steps to remove an SKU / Product from being applicable to a specific version:

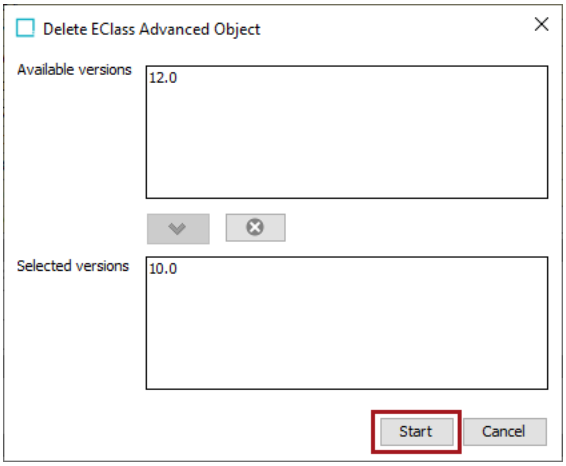
1. Navigate to the Tree tab and select the desired object.
2. Right-click and hover the cursor over ECLASS Advanced menu, and then select 'Delete version' option.



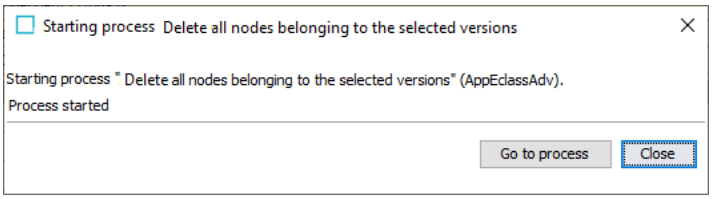
3. In the 'Delete ECLASS Advanced object' dialog that is displayed, select the version(s) to be deleted and click the Down icon ( ).



4. After the desired version is being populated in the 'Selected versions' field, click the 'Start' button to initiate the deletion process. In the example below, ECLASS Advanced data for version 10.0 is to be deleted.



5. The deletion action is run as a background process, and you can access the Background Process information through the execution report. In the dialog labeled 'Starting process Delete all nodes belonging to the selected versions,' click the 'Go to process' button to navigate to the actual background process.



The report provides error details if any issues are encountered. Consequently, the report displays no errors and no warnings when the process is error-free and successful.

> **Note:** The deletion process is irreversible.

Duplicate Version Action

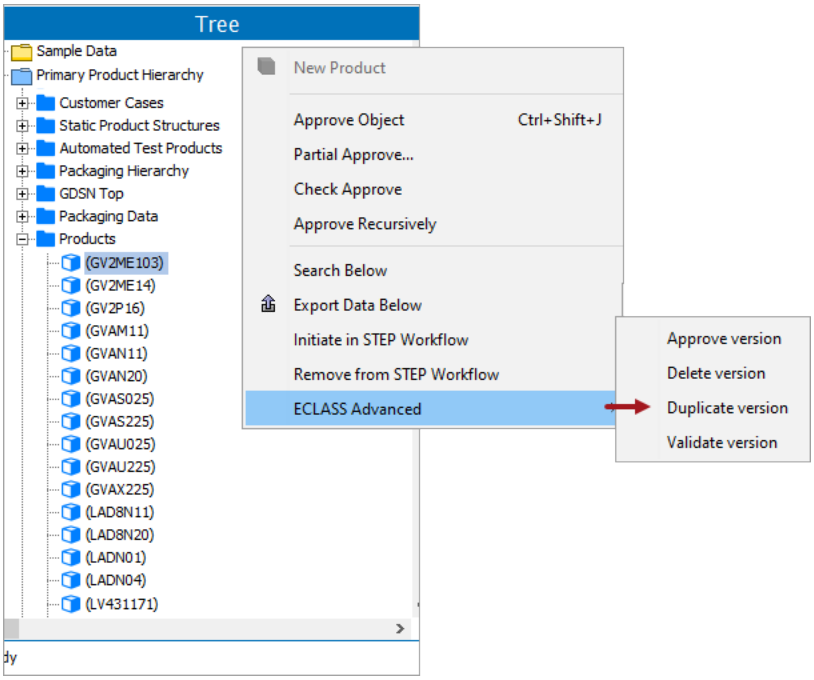
The duplication action enables the duplication of ECLASS Advanced version(s) along with their associated Product Blocks and Aspects. The user retains the choice of either generating a new SKU / Product or duplicating the data into an existing SKU / Product.

If you choose to duplicate versions for an existing SKU / Product rather than creating a new one, the selected SKU / Product will be enriched with additional versions. Yet, if a version is already defined within the SKU / Product (determined through the ECLASS Advanced classification link), the duplication of that version will not take place.

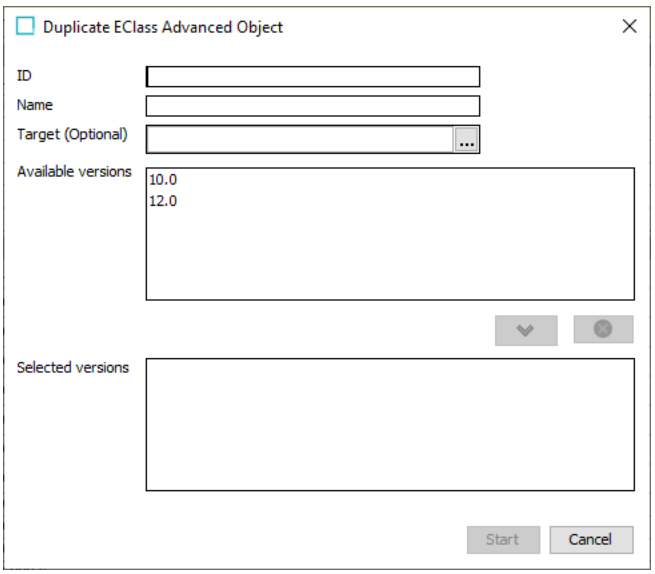
Procedure to Duplicate Version(s)

Below are the steps to duplicate ECLASS Advanced versions.

1. Navigate to the Tree tab and select the desired object.
2. Right-click and hover the cursor over ECLASS Advanced menu, and then select 'Duplicate version' option.

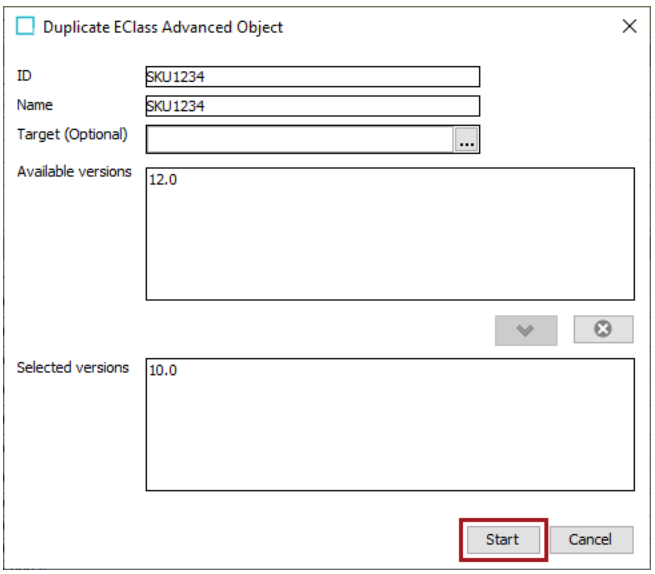


3. In the 'Duplicate ECLASS Advanced object' dialog that is being displayed, populate the following parameters:

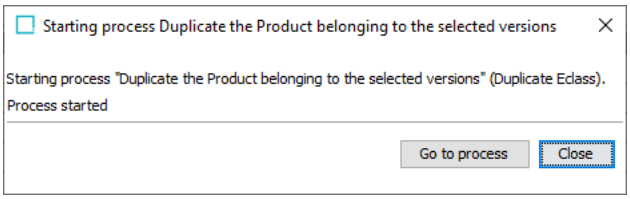


- **ID:** This parameter is to be populated if you are duplicating the data into a new SKU / Product. Type the ID of the new SKU / Product.
- **Name:** Type a name for the new SKU / Product.
- **Target (Optional):** This parameter is to be populated if you are duplicating the data into an existing SKU / Product. To select an existing SKU / Product, click the ellipsis button (...).
- **Available versions / Selected versions:** The 'Available versions' parameter enumerates all the versions to which the current SKU / Product is applicable. Select the relevant version(s) and click the Down icon ( ).

4. After the desired version is being populated in the 'Selected versions' field, click the 'Start' button to initiate the duplication process. In the example below, ECLASS Advanced data from version 10.0 is selected to be duplicated into a new product 'SKU1234.'



5. The duplication is run as a background process, and you can access the Background Process information through the execution report. In the dialog labeled 'Starting process Duplicate the Product belonging to the selected versions,' click the 'Go to process' button to navigate to the actual background process.



The report provides error details if any issues are encountered. Consequently, the report displays no errors and no warnings when the process is error-free and successful.

You can locate the duplicated Product Block and Aspect Groups, as well as Product Blocks and Aspects, within the root nodes named eClassAdv_[Version]ProductBlocks and eClassAdv[Version]_ProductAspects.

Validate Version Action

The 'Validation version' action tests the structure of the data imported through BMEcat 2005.1 file.

This validation process encompasses the following actions:

- Checks the validity of Product Blocks and Aspect hierarchy in terms of classifications
- Checks for the cardinality consistency
- Checks for orphan values

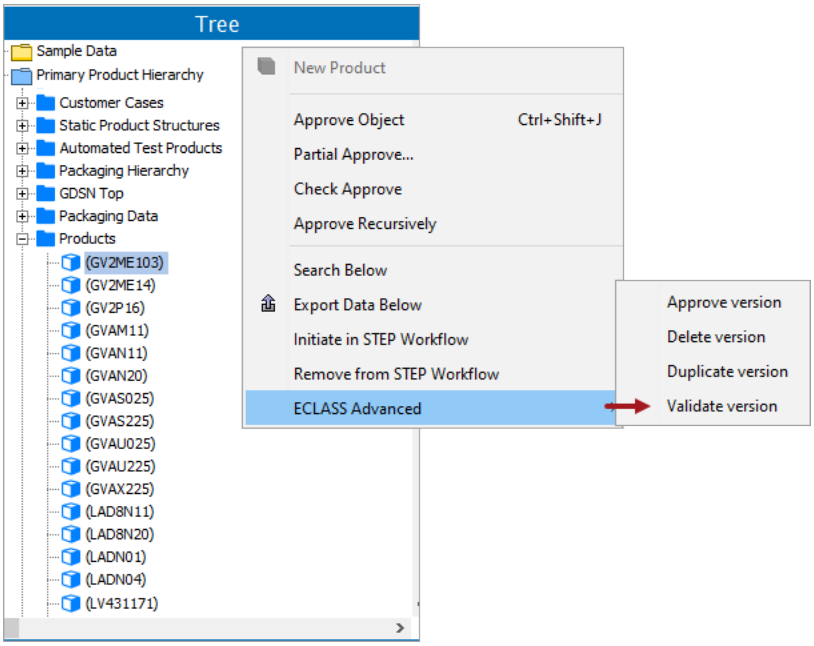
However, the validation process does not cover the following:

- It does not validate anything related to data containers
- It does not validate data for data quality purpose

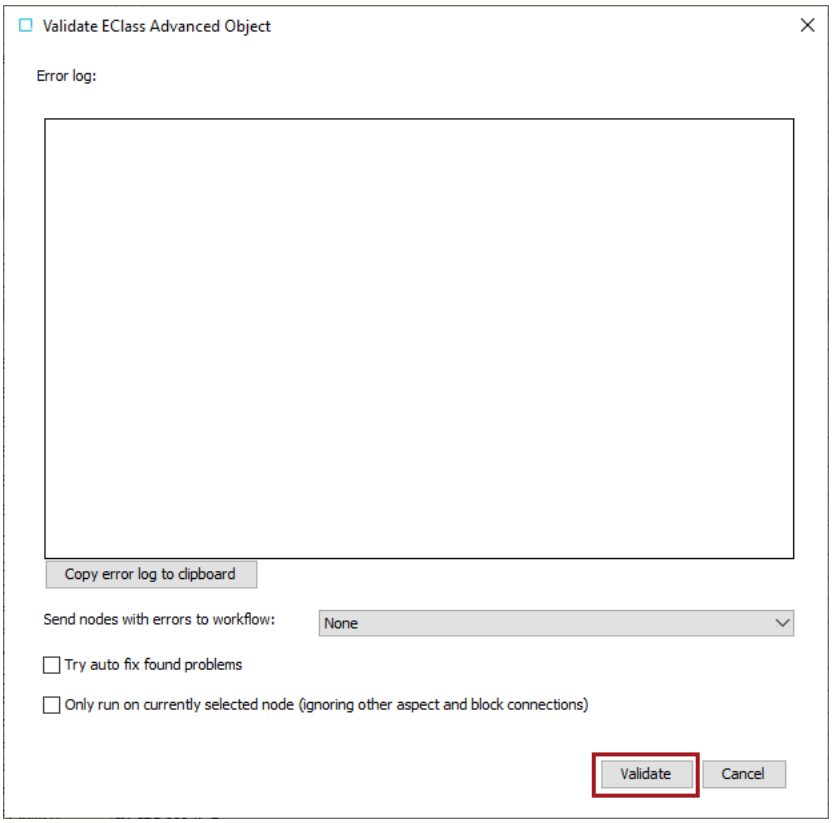
Procedure to validate an SKU / Product in the workbench

Below are the steps to validate an SKU / Product:

1. Navigate to the Tree tab and select the desired object. The object can be an SKU / Product or Product Block or Product Aspect.
2. Right-click and hover the cursor over ECLASS Advanced menu, and then select 'Validate version' option.

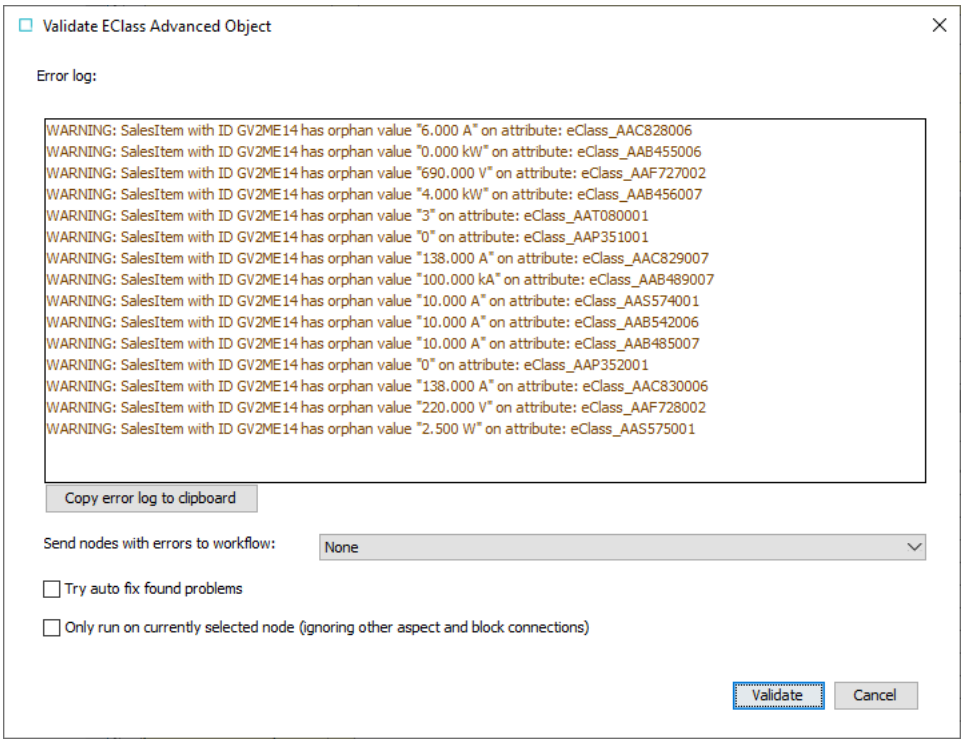


3. In the 'Validate ECLASS Advanced objects' dialog that is displayed, click 'Validate' button.



4. After the 'Error log' parameter is populated with the error messages (as shown below), you can copy the error messages by clicking the 'Copy error log to clipboard' button.

   Meanwhile, the following are the parameters available within the 'Validate ECLASS Advanced Editor Screen Advanced objects' dialog:



- **Error log:** This field displays each warning or fatal error presented on a distinct line. Users have the option to click on a line and use Ctrl+C to copy the content of the respective line.
- **Copy error log to clipboard:** This button allows you to copy all lines of the error log to the clipboard.
- **Send notes with errors to workflow:** This dropdown allows users to select a STEP Workflow. If a node contains validation errors, the system will push those nodes into the selected workflow.
- **Try auto fix found problems:** When this checkbox is selected, the system will attempt to automatically resolve any found validation errors. Note that some errors may necessitate manual intervention.
- **Only run on currently selected node:** Enabling this checkbox confines the validation check to the currently selected node. Validation check on the referenced nodes are excluded.
- **Validate:** Clicking this button will initiate the validation process.

5. To close the dialog, click 'Cancel' button.

Types of Validation Warnings / Errors

The below table outlines different types of validation warnings and errors:

Type	Importance	Test on object type	Description	Can be auto-fixed
Has Classification Link	FATAL	SKU, Aspect, Block	Checks whether the node has at least one ECLASS Advanced Classification Link	No
Check for Orphan Values	WARNING	SKU, Aspect, Block	Checks if the node has any orphan values	Yes
Do Cardinality Attributes' values match number of references of that type	FATAL	Aspect, Block	Checks if the node has any connected attributes of CONDITION_DET_TYPE in which case it check	Yes
Does SKU uphold 1:1 relationship constraints for its NON_DEPENDENT_P_DET_TYPE attribute targets	FATAL	SKU	Checks if there is only one reference of the type corresponding to the classification's attributes types of NON_DEPENDENT_P_DET_TYPE	No

Scalable ECLASS: European Classification for Advanced e-Commerce

> ⚠ **Important:** The scalable ECLASS Advanced data model is a redesigned approach for managing ECLASS Advanced structures within STEP. These changes require a migration that must be carefully planned, agreed upon, and executed to prevent system disruptions. If your environment is impacted, contact your Stibo Systems Account Manager to discuss next steps.

The ECLASS data standard provides structured, machine-readable classifications and descriptions for products, materials, systems, and services. It enables consistent and interoperable data exchange across platforms, supporting integration in industrial and e-commerce environments. This document outlines the ECLASS standard and its implementation within the STEP system.

ECLASS serves as a semantic framework for standardized product data communication. By adopting ECLASS, organizations can automate data exchange processes, reduce manual transformation efforts, and ensure compatibility across systems. This facilitates efficient data handling throughout the supply chain.

ECLASS Basic and ECLASS Advanced are two variants of the ECLASS standard, each serving distinct application requirements. Below are the features of both variants:

Feature	ECLASS Basic	ECLASS Advanced
Primary Use Cases	Procurement, catalog management	Engineering, IoT, digital twin, complex product modeling
Data Granularity	Basic attribute-level data	ECLASS Basic and ECLASS Advanced share the same attribute structure, with Basic being a perfect subset of Advanced. However, ECLASS Advanced introduces a hierarchical classification that enables more detailed product descriptions.
Integration Scope	Suitable for basic ERP and procurement systems	Ideal for PLM, CAD / CAM, and advanced digital ecosystems

STEP supports both ECLASS Basic and ECLASS Advanced, enabling organizations to harness the full potential of the ECLASS standard. Through this integration, STEP enhances the precision, efficiency, and consistency of product data management and exchange across diverse systems and platforms.

This will be the comprehensive guide to data handling within STEP in the ECLASS Advanced standard. This document serves as a valuable resource for understanding the intricate aspects of data management within the ECLASS Advanced framework. As you navigate through the sections ahead, you will gain comprehensive insights into the various facets of handling data in accordance with the ECLASS Advanced standard.

This portion of Solution Enablement material introduces the ECLASS Advanced functionality, including the use of Easy Setup and other necessary configurations.

Click on a title in the left navigation panel to expand the topics under it, or click the main topics below:

- ECLASS Advanced Terminology
- ECLASS Advanced Quick Start Guide
- ECLASS Advanced Reference Guide

It is recommended that users be familiar with STEP and the STEP Online Help topics before beginning ECLASS Advanced functions.

ECLASS Advanced Terminology

STEP supports multiple industry standards across various sectors. To work efficiently with the ECLASS standard in STEP, users should familiarize themselves with key industry-specific terminology. This topic introduces essential ECLASS terms commonly encountered in daily operations.

For a detailed explanation of the ECLASS logic and structural elements, refer to the official ECLASS documentation: https://eclass.eu/support/technical-specification/structure-and-structural-elements.

Aspects, Blocks, and Properties

In ECLASS Advanced, the terms Aspects and Blocks relate to how product data is structured for detailed, standardized representation.

Blocks Blocks are groups of properties that logically belong together and describe specific characteristics or functional areas of a product. They organize properties into meaningful clusters, helping to structure product information in a clear, modular way.

For example, for an electric motor, a Block might be 'Electrical Characteristics,' containing properties like voltage, power consumption, current type, etc.

Another example of a Block could be 'Mechanical Data,' which might include properties like shaft diameter, housing material, etc.

Aspects: Aspects are higher-level groupings used to organize multiple Blocks together, representing a broader segment or section of product data. They act as containers for Blocks, providing a structured hierarchy and sequence in how product information is displayed or transmitted.

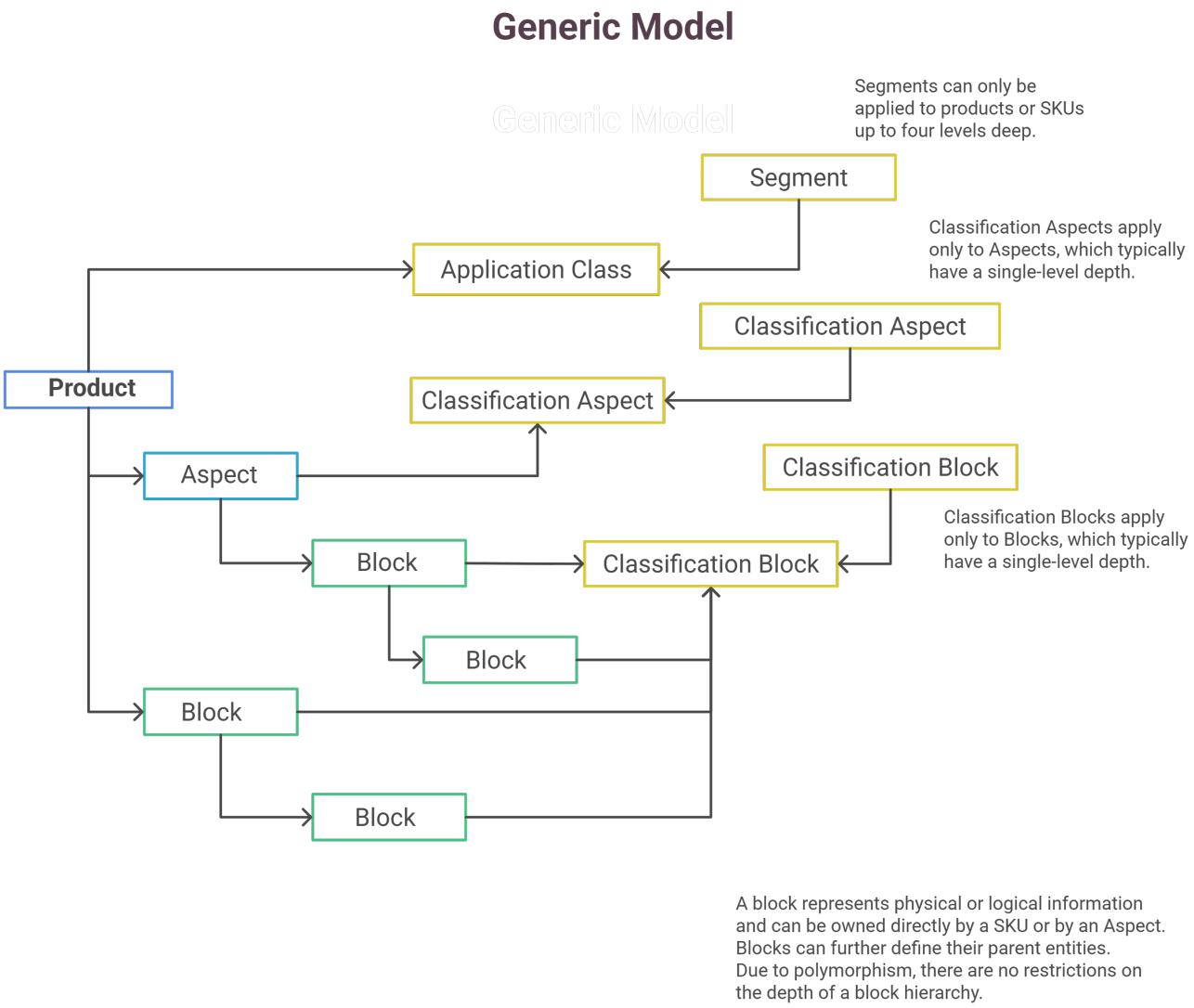
Example: An Aspect called 'Technical Data' might contain several Blocks:

- Electrical Characteristics
- Mechanical Data
- Performance Parameters

Another example of an Aspect might be 'Commercial Information,' grouping Blocks like packaging data, pricing information, delivery time, etc.

Property: A Property is the smallest unit of information used to describe a characteristic or feature of a product in ECLASS Advanced. Properties carry the actual values.

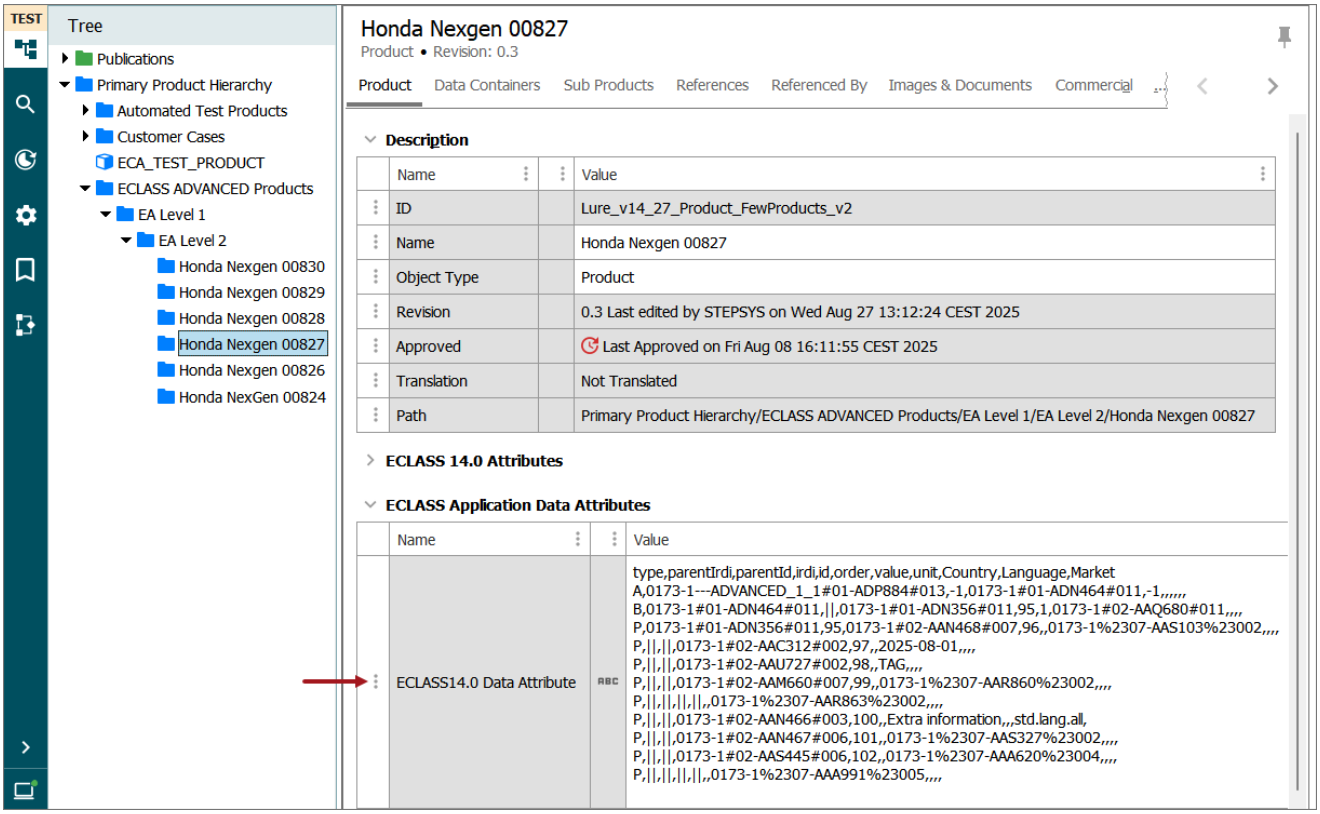
As an example, a Block called 'Technical Data' might contain an aspect called Electrical Characteristics. The properties within it could be Voltage, Power Consumption, etc.



While the text and diagram above explains the generic definition of aspects, blocks and properties, the below paragraph explains how these are stored in a STEP system.

How STEP handles Aspects, Blocks, and Properties in the system

The representation of Aspects, Blocks, and Properties in STEP is version-specific and product-dependent. In STEP, the Aspects, Blocks, and Properties are not stored as free-floating objects but are instead tied to a product. For each product, all applicable Aspects and Blocks are stored in CSV format within a version-specific attribute. Meaning, STEP stores the relevant Aspects and Blocks for a product in a CSV format inside an attribute that belongs to that product. These attributes are organized under the attribute group ECLASS Application Data Attributes and follow a version-based naming convention, such as ECLASS 14.0 Data Attribute for version 14.0 or ECLASS 13.0 Data Attribute for version 13.0.



Important: It is strictly prohibited to manually edit, alter, or tamper with the CSV structure stored within the version-specific attributes under ECLASS Application Data Attributes. These CSV files are system-generated and follow a precise schema required for correct interpretation by STEP. Any unauthorized modifications may lead to data corruption, loss of linkages between Aspects, Blocks, and Properties, system errors during import/export operations, and incompatibility with future ECLASS versions or STEP updates. Always use the designated interfaces (like ECLASS Advanced Editor screen in the Web UI) and tools supported by STEP for managing product attributes.

Cardinality and Polymorphism

In ECLASS Advanced, Cardinality and Polymorphism are structural and functional concepts that define how data is organized, repeated, or substituted within Aspects and Blocks.

Cardinality: Cardinality defines how many times a Block can appear or be repeated within a given context. The primary purpose of cardinality is to control or specify whether a block:

- Must appear exactly once
- Can appear optionally
- Can repeat multiple times

Example use cases:

- An aspect named Electrical Characteristics has a cardinality of 1..n. This means each product must include *at least one* electrical connection, but may have multiple.
- A block named Technical Data has a cardinality of 1..1. This indicates that the block is mandatory and must appear *exactly once*.

Polymorphism: Polymorphism allows one Block to be substituted with another equivalent block, depending on the context or product type.

The primary purpose of polymorphism is to introduce flexibility in the data structure by enabling:

- Different products to represent different data structures while still conforming to the standard.
- Substitution of Blocks based on product variants.

Example use case:

Consider two Blocks:

- Block A: Electrical Characteristics
- Block B: Pneumatic Characteristics

A product may use either Block A or Block B, depending on whether it operates electrically or pneumatically. The system dynamically substitutes one block with the other without violating the standard structure.

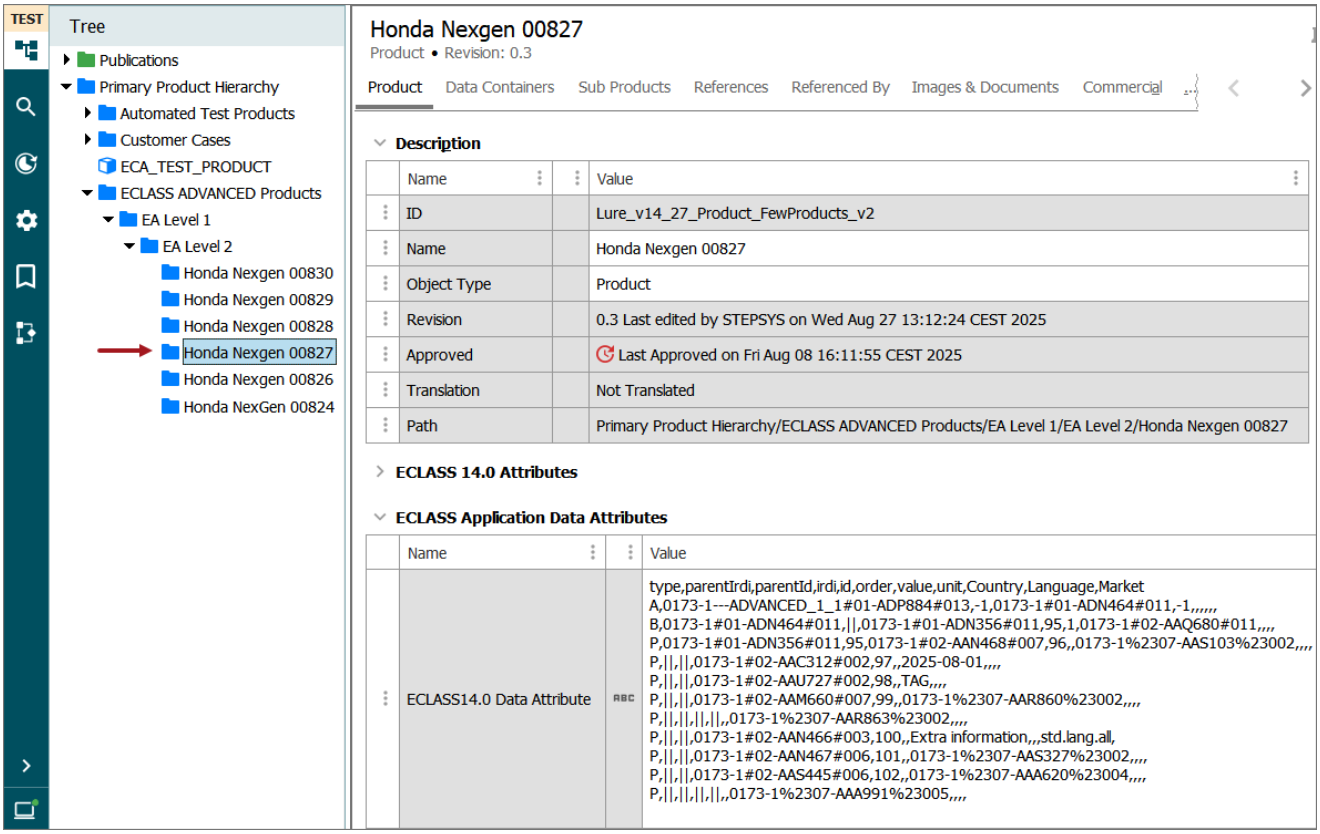
Product and Applications

Product: A Product is a specific instance of a real-world item or service that is described using the structured classification and property system defined by ECLASS.

It represents a concrete offering from a manufacturer or supplier, such as a motor, valve, connector, or even a software service. A product is classified under one or more ECLASS Classes.

As an example, a Product is the actual item like a 230V Electric Motor Model X.

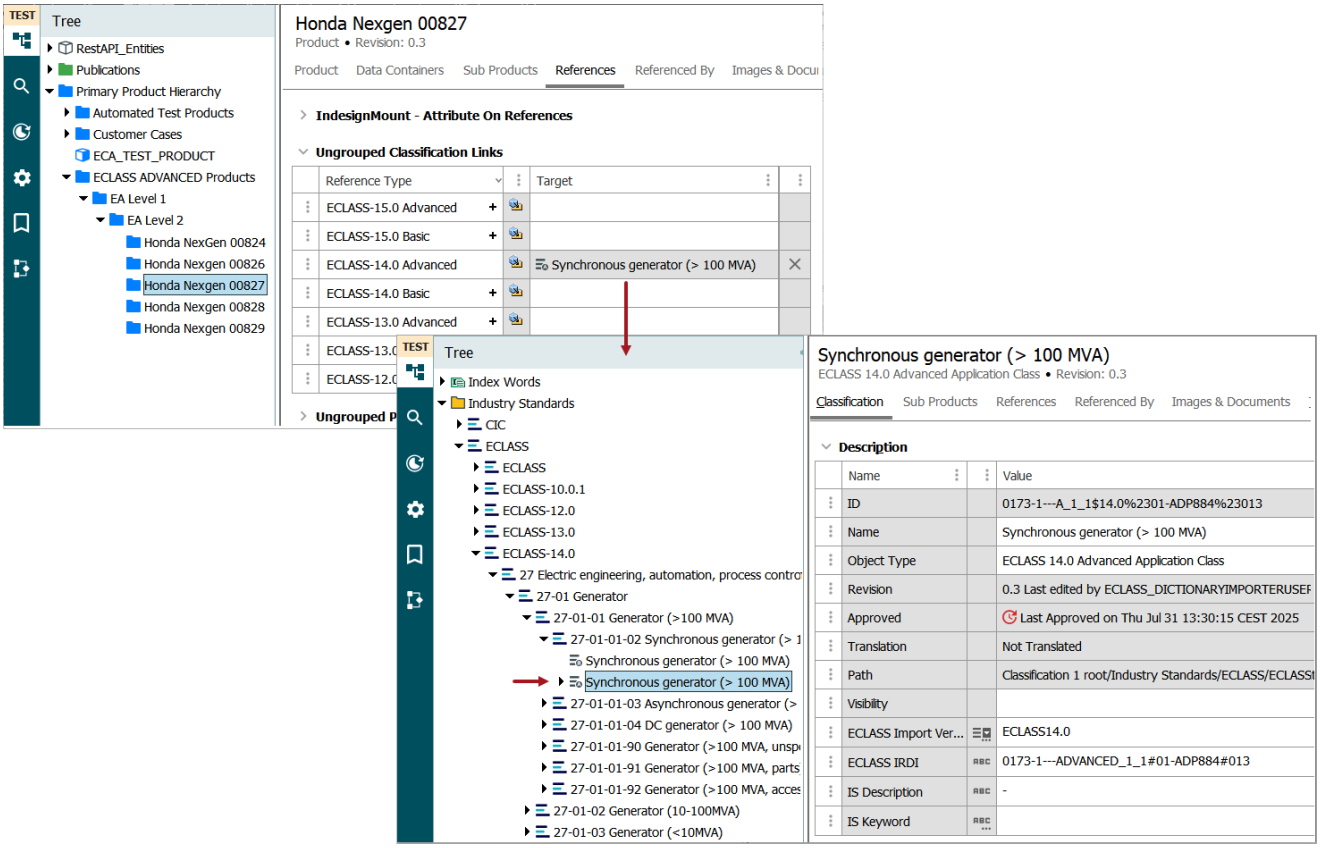
A product in STEP is always stored under the Primary Product Hierarchy within the Tree tab, as shown in the screenshot below.



Application: An Application in ECLASS refers to the context or condition under which a property is valid or relevant for a product. It is a way to define when and how a property should be applied.

Applications are stored within the classification folder called Industry Standards under their respective ECLASS version hierarchy and are linked to products through a reference type.

For example, let us say you have a product called Honda Nexgen 00827. In ECLASS version 14, this product might be applicated (linked via an application) to Synchronous Generator (>100 MVA). This means that the properties defined under that application, such as cooling method, insulation class, or efficiency are valid only when the product is used as a synchronous generator in that power range.



ECLASS Advanced Templates

Templates help ensure that your counterpart receives exactly the data they need by filtering elements such as Aspects, Blocks, and Properties, defining mandatory elements to guarantee completeness, and restricting values to those appropriate for each element. Templates do not create new elements; they only limit the existing ECLASS definition. Acting as a blueprint for describing products within a class, they specify what information is required and how it is structured. STEP implementation of these templates emphasizes filtering and validation rather than extending the ECLASS definition.

More detailed information about the XML Schema Definition (XSD) file used for defining templates in the ECLASS Advanced Template mechanism is available in the Understanding the ECLASS Advanced Template XML Schema topic.

Understanding the ECLASS Advanced Template XML Schema

This topic provides a detailed explanation of the XML Schema Definition (XSD) file used for defining templates in the ECLASS Advanced Template mechanism. This XSD file acts as a blueprint that describes how the XML document (template definition) should be structured, specifying the elements and attributes allowed, their data types, and the expected order of appearance.

Below is a brief description of the main elements in the XSD file used for defining templates in the ECLASS Advanced Template mechanism.

- **xs:schema:** This is the root element of the XML Schema document. It defines the form of the elements and attributes in an XML document.
- **xs:annotation** and **xs:documentation**: These tags are used to add comments and notes within the schema. They don't affect the structure or validation of the XML documents that the schema describes.
- **xs:element:** This defines an element that can appear in the XML document. The example schema below uses this to define the name as 'EclassFilter.'
- **xs:complexType:** This defines a complex type that groups a set of element and attribute declarations. Complex types like 'Header,' 'Name,' 'Application,' and others are defined in the example schema below.
- **xs:sequence:** This specifies that the child elements must appear in a sequence. Each child element can occur from zero to any number of times.
- **xs:attribute:** This defines an attribute for an element. The attribute can be used in the start tag of an element. Attributes like 'id,' 'irdi,' 'orderNumber,' and others are defined in the example schema below.
- **xs:simpleType** and **xs:restriction:** These define a simple type and its restrictions. For example, the 'StepID' simple type is a string with a maximum length of 40 is defined in the example schema below.
- **xs:choice:** This allows only one out of several possible elements to be present. In the example schema below, the 'Application' complex type is defined with a choice between of 'Property,' 'Block,' and 'Aspect' elements.

Each of these elements and attributes has a specific purpose and rules that it must follow. The annotations in the schema provide additional information about the purpose of each element and attribute. For example, the 'EclassFilter' element is the root element for ECLASS application filters, and its 'id' attribute specifies the ID of the ECLASS application filters.

Refer to the online version of this topic for the example.

Below is an example XSD file:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://www.stibosystems.com/eclass_filter"
           xmlns="http://www.stibosystems.com/eclass_filter"
           elementFormDefault="qualified" attributeFormDefault="unqualified">

    <xs:annotation>
        <xs:documentation>
            This XML schema definition describes a valid definition of the filters used in ECLASS Advanced filtering
            mechanism.
        </xs:documentation>
    </xs:annotation>

    <xs:element name="EclassFilter">
        <xs:complexType>
            <xs:annotation>
                <xs:documentation>
                    Root element for ECLASS Advanced application filters.
                </xs:documentation>
            </xs:annotation>
            <xs:sequence>
                <xs:element name="Header" type="Header" minOccurs="0" maxOccurs="1"/>
                <xs:element name="Application" type="Application" minOccurs="0" maxOccurs="unbounded"/>
```

```
            </xs:sequence>
            <xs:attribute name="id" type="StepID" use="required">
                <xs:annotation>
                    <xs:documentation>
                        Specifies the ID of the ECLASS Advanced application filters, but is not used in ECLASS Advanced editor.
                        The filter ID is taken from XML asset id (filter definition).
                    </xs:documentation>
                </xs:annotation>
            </xs:attribute>
        </xs:complexType>
    </xs:element>

    <xs:complexType name="Header">
        <xs:annotation>
            <xs:documentation>
                Optional filter information not used in ECLASS Advanced editor.
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="Name" type="Name" minOccurs="1" maxOccurs="unbounded"/>
            <xs:element name="CreationDate" type="xs:dateTime" minOccurs="0" maxOccurs="1">
                <xs:annotation>
                    <xs:documentation>
                        Specifies optional creation date.
```

```
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element
name="ModificationDate" type="xs:dateTime" minOccurs="0" maxOccurs="1">
                <xs:annotation>
                    <xs:documentation>
                        Specifies optional last modification date.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element
name="Creator" type="xs:string" minOccurs="0" maxOccurs="1">
                <xs:annotation>
                    <xs:documentation>
                        Specifies optional name of creator.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="Name">
        <xs:annotation>
            <xs:documentation>
                Describes the name of the application filters, but is not used
in ECLASS Advanced editor.
            </xs:documentation>
```

```
        </xs:annotation>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="lang" type="Lang" use="optional">
                    <xs:annotation>
                        <xs:documentation>
                            Specifies optional language code of the name. Not used in ECLASS Advanced editor.
                        </xs:documentation>
                    </xs:annotation>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>

    <xs:simpleType name="StepID">
        <xs:annotation>
            <xs:documentation>
                STEP ID value
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:maxLength value="40"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="Application">
        <xs:annotation>
```

```
            <xs:documentation>
                Describes application class from a particular ECLASS Advanced version that the current XML asset (filter definition) is valid to. If this tag does not include any child tags like:
                Property, Block, or Aspect, then all nested data (aspects, blocks, properties) belonging to that application class will be displayed without any filtering.
            </xs:documentation>
        </xs:annotation>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element name="Property" type="Property" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="Block" type="Block" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="Aspect" type="Aspect" minOccurs="0" maxOccurs="unbounded"/>
        </xs:choice>
        <xs:attribute name="irdi" type="IRDI" use="required">
            <xs:annotation>
                <xs:documentation>
                    Specifies the reference to the application IRDI id, required value.
                </xs:documentation>
            </xs:annotation>
        </xs:attribute>
    </xs:complexType>

    <xs:complexType name="Property">
        <xs:annotation>
```

```
121             <xs:documentation>
122                 Describes the property that requires to be filtered. If this tag does not include any Value tags then all value tags is considered for display by default (no further filtering of values).
123             </xs:documentation>
124         </xs:annotation>
125         <xs:sequence>
126             <xs:element name="Context" type="Context" minOccurs="0" maxOccurs="unbounded"/>
127             <xs:element name="Value" type="Value" minOccurs="0" maxOccurs="unbounded"/>
128         </xs:sequence>
129         <xs:attribute name="irdi" type="IRDI" use="required">
130             <xs:annotation>
131                 <xs:documentation>
132                     Specifies the reference to the property IRDI id, required value.
133                 </xs:documentation>
134             </xs:annotation>
135         </xs:attribute>
136         <xs:attribute name="orderNumber" type="OrderNumber" use="optional"/>
137         <xs:attribute name="isRequired" type="xs:boolean" use="optional" default="false">
138             <xs:annotation>
139                 <xs:documentation>
140                     Specifies that the property value is required. If provided, then in ECLASS Advanced editor the
141                     field is marked as a required field.
142                 </xs:documentation>
```

```
143             </xs:annotation>
144         </xs:attribute>
145     </xs:complexType>
146 
147     <xs:complexType name="Value">
148         <xs:annotation>
149             <xs:documentation>
150                 Restricted value for the property for filtering values in the
property by STEP LOV id or IRDI id. One of
151                 attributes has to be provided: IRDI id or STEP LOV id.
152             </xs:documentation>
153         </xs:annotation>
154         <xs:simpleContent>
155             <xs:extension base="xs:string">
156                 <xs:attribute name="irdi" type="IRDI" use="optional">
157                     <xs:annotation>
158                         <xs:documentation>
159                             Specifies the reference to the value IRDI id (full
number).
160                         </xs:documentation>
161                     </xs:annotation>
162                 </xs:attribute>
163                 <xs:attribute name="id" type="StepID" use="optional">
164                     <xs:annotation>
165                         <xs:documentation>
166                             Specifies the reference to the STEP LOV id.
167                         </xs:documentation>
168                     </xs:annotation>
```

```
169                 </xs:attribute>
170                 <xs:attribute
name="orderNumber" type="OrderNumber" use="optional"/>
171             </xs:extension>
172         </xs:simpleContent>
173     </xs:complexType>
174 
175     <xs:complexType name="Block">
176         <xs:annotation>
177             <xs:documentation>
178                 This tag outlines the block definition. In the absence of any
child tags such as Property or Block, all the nested data, including blocks and
properties associated with that block, will be presented without any filtration.
179             </xs:documentation>
180             </xs:documentation>
181         </xs:annotation>
182         <xs:choice minOccurs="0" maxOccurs="unbounded">
183             <xs:element
name="Context" type="Context" minOccurs="0" maxOccurs="unbounded"/>
184             <xs:element
name="Property" type="Property" minOccurs="0" maxOccurs="unbounded"/>
185             <xs:element
name="Block" type="Block" minOccurs="0" maxOccurs="unbounded"/>
186         </xs:choice>
187         <xs:attribute name="irdi" type="IRDI" use="required">
188             <xs:annotation>
189                 <xs:documentation>
190                     Specifies the reference to the block IRDI id, required
value.
```

```
                </xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:attribute name="orderNumber" type="OrderNumber" use="optional"/>
    </xs:complexType>

    <xs:complexType name="Aspect">
        <xs:annotation>
            <xs:documentation>
                  This tag outlines the aspect definition. In the absence of any child tags such as Property or Block, all the nested data, including blocks and properties associated with that aspect, will be presented without any filtration.
            </xs:documentation>
        </xs:annotation>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element name="Context" type="Context" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="Property" type="Property" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="Block" type="Block" minOccurs="0" maxOccurs="unbounded"/>
        </xs:choice>
        <xs:attribute name="irdi" type="IRDI" use="required">
            <xs:annotation>
                <xs:documentation>
                    Specifies the reference to the aspect IRDI id, required value.
                </xs:documentation>
```

```
            </xs:annotation>
        </xs:attribute>
        <xs:attribute name="orderNumber" type="OrderNumber" use="optional"/>
    </xs:complexType>

    <xs:complexType name="Context">
        <xs:annotation>
            <xs:documentation>
                Describes context path. The filter structure can be written as a tree structure as defined below:

                application (app)
                    property 1
                    property 2
                    block 3
                        block 3.1
                        block 3.2
                            property 3.2.1
                            property 3.2.2
                        block 3.3

                However, it also can be written as a flat structure as shown below:

                application (app)
                    property 1
                    property 2
                    block 3
```

```
                    block 3.1
                    block 3.2
                    property 3.2.1
                    property 3.2.2
                    block 3.3

                In a tree structure, the relationship between nodes is determined by nested tags. Conversely, in a flat structure, the connection between nodes is established through the context of the elements.

                In context, tag are list of elements defining path to the parent object. If context is not
                provided, then the parent object is obtained from the parent tag.

                For the above given example, in order to define the tree structure, nested blocks and properties have to contain
                context tag with proper elements as shown below:

                application (app)
                    property 1
                    property 2
                    block 3
                    block 3.1
                        context
                            element: ref="app"
                            element: ref="3"
                    block 3.2
```

```
                        context
                            element: ref="app"
                            element: ref="3"
                    property 3.2.1
                        context
                            element: ref="app"
                            element: ref="3"
                            element: ref="3.2"
                    property 3.2.2
                        context
                            element: ref="app"
                            element: ref="3"
                            element: ref="3.2"
                    block 3.3
                        context
                            element: ref="app"
                            element: ref="3"
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element
name="Element" type="Element" minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="Element">
        <xs:annotation>
            <xs:documentation>
```

```
289                 Describes element path. The element defines the parent element
of the node. The order of elements
290                 should be aligned from application to final element.
291             </xs:documentation>
292         </xs:annotation>
293         <xs:attribute name="ref" type="IRDI" use="required">
294             <xs:annotation>
295                 <xs:documentation>
296                     Specifies the reference parent element IRDI
297                 </xs:documentation>
298             </xs:annotation>
299         </xs:attribute>
300     </xs:complexType>
301
302     <xs:simpleType name="IRDI">
303         <xs:annotation>
304             <xs:documentation>
305                 Describes IRDI "International Registration Data Identifier". It
is a unique identifier used within
306                 the International Registration Data Identifier (IRDI) system,
which is a part of the International
307                 Registration Data Access Protocol (IRDAP).
308             </xs:documentation>
309         </xs:annotation>
310         <xs:restriction base="xs:string">
311             <xs:maxLength value="40"/>
312         </xs:restriction>
313     </xs:simpleType>
```

```
314
315     <xs:simpleType name="Lang">
316         <xs:annotation>
317             <xs:documentation>
318                 Lanugage represented by ISO 639-2 alpha-3 code.
319             </xs:documentation>
320         </xs:annotation>
321         <xs:restriction base="xs:string">
322             <xs:maxLength value="3"/>
323         </xs:restriction>
324     </xs:simpleType>
325
326     <xs:simpleType name="OrderNumber">
327         <xs:annotation>
328             <xs:documentation>
329                 It specifies the order number of an element. If the OrderNumber
    is used, it supersedes the existing ECLASS specification order and utilizes the
    number for arranging from minimum to maximum values.
330             </xs:documentation>
331         </xs:annotation>
332         <xs:restriction base="xs:integer">
333             <xs:minInclusive value="0"/>
334         </xs:restriction>
335     </xs:simpleType>
336
337 </xs:schema>
```

The following is a sample of a template definition XML file, which is composed according to the previously mentioned XML Schema Definition (Refer to the online version of this topic for the example.):

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EclassFilter id="FilterID" xmlns="http://www.stibosystems.com/eclass_filter">
    <Header>
        <Name>Information</Name>
    </Header>
    <Application irdi="0173-1---ADVANCED_1_1#01-ADO120#007">
        <Property irdi="0173-1#01-ADN329#000"/>
        <Property irdi="0173-1#01-ADN329#001"/>
        <Aspect irdi="0173-1#01-ADN329#002">
            <Property irdi="0173-1#01-ADN329#001"/>
            <Block irdi="0173-1#01-ADN334#001">
                <Property irdi="0173-1#01-ADN330#001"/>
                <Property irdi="0173-1#01-ADN330#002"/>
                <Property irdi="0173-1#01-ADN330#003"/>
            </Block>
        </Aspect>
    </Application>
    <Application irdi="0173-1---ADVANCED_1_1#01-ADO120#006">
        <Aspect irdi="0173-1#01-ADN329#002"/>
    </Application>
</EclassFilter>
```

ECLASS Advanced Quick Start Guide

This guide introduces to the ECLASS Advanced solution available in STEP 2025.3 and newer versions. It covers the necessary actions an admin must take to set up the solution, as well as providing an overview of the end user functionality that is provided with the solution after Easy Setup actions for ECLASS Advanced standard have been completed by an admin.

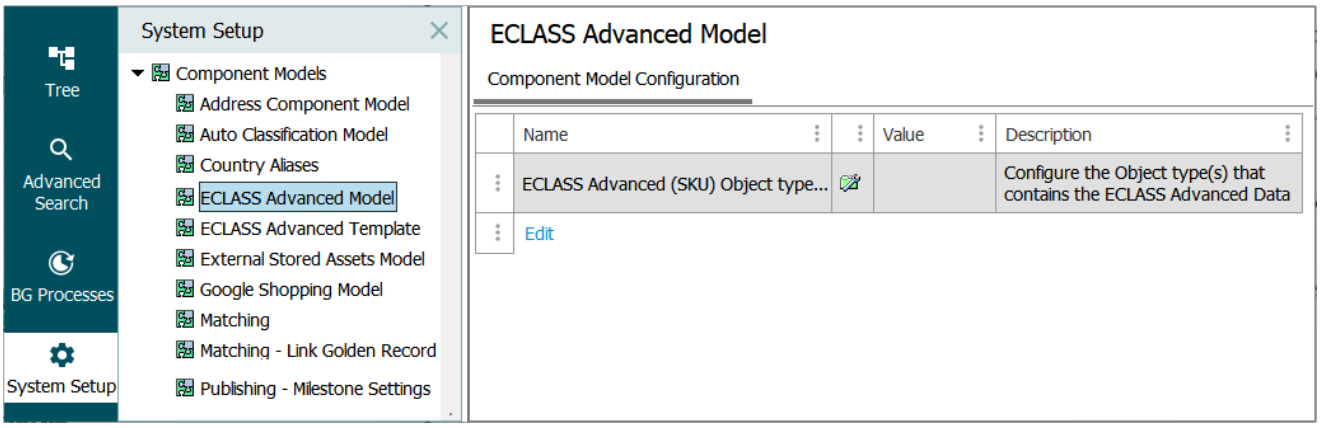
The ECLASS Advanced commercial license is required to use this functionality. Contact Stibo Systems to begin the process of enabling a license or licenses for your system.

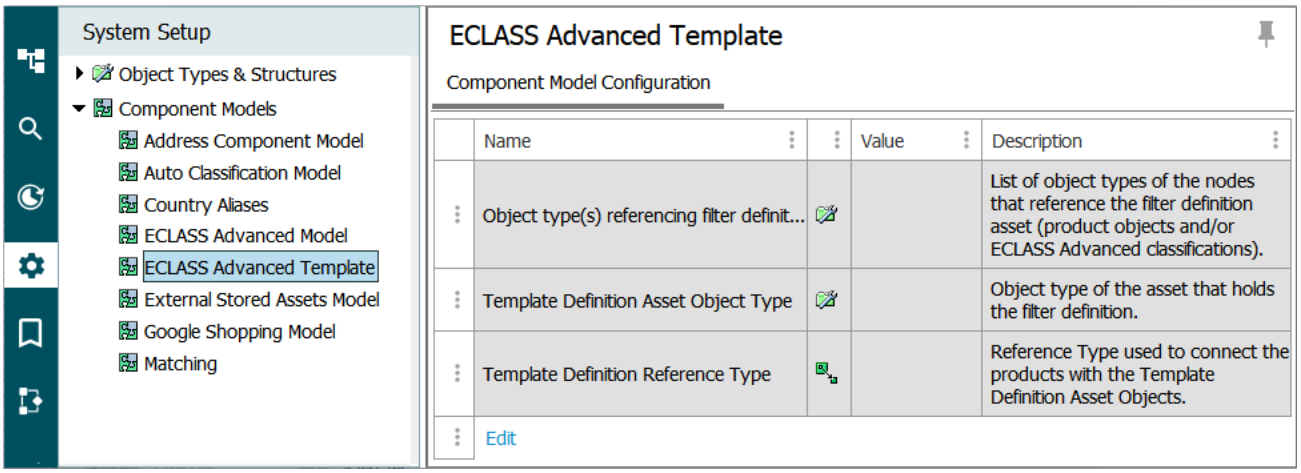
ECLASS Advanced Data Models and Easy Setup

The ECLASS Advanced solution requires a data model for setting up the standard. When Easy Setup actions outlined in this documentation are completed, many of the configurations for the ECLASS Advanced solution are automatically configured within the 'ECLASS Advanced Model' and 'ECLASS Advanced Template' that are available within the workbench > System Setup > Component Models.

The Easy Setup actions are not designed to provide a complete solution but rather give admins a jump start on completing many of the necessary configurations.

Below are the screenshots of the component models that are to be configured for ECLASS standard.





Easy Setup Generic Considerations

When executing the Easy Setup action, the following considerations come into play:

- If a user modifies Data Model objects, rerunning the Easy Setup will not overwrite such changes. For example, alterations to validation base type in an attribute, multivalued settings, or defining if the object is externally maintained or not, etc.
- If a customer removes Data Model objects, rerunning the Easy Setup will recreate them. This applies, for example, when deleting static or metadata attributes, the Easy Setup restores them.
- When a customer adds Data Model object(s), rerunning the Easy Setup will not eliminate these objects. The Easy Setup will never delete / erase data. For example, newly added attributes, metadata attributes, or objects remain intact.
- Once the Easy Setup is unintentionally executed, reverting changes is not possible. Manual removal of created data may become necessary in extreme cases.
- The Easy Setup action will always be executed on the logged-on user's credentials.

> **Note:** Under no circumstances shall the Stibo Systems ECLASS Advanced Data Model be changed by any user.

ECLASS Advanced Quick Start Setup for Admins

This section addresses the necessary actions an admin must take to set up the ECLASS Advanced solution.

Prerequisites

It is assumed that the admin has knowledge of STEP administrative functions and experience working in System Setup, including creating and editing workflows, business rules, Web UIs, attributes, etc. Therefore, this guide does not provide introductory material for these concepts and instead targets only the specific information needed for a knowledgeable STEP admin to complete the ECLASS Advanced solution setup. If additional information is needed, refer to the STEP Online Help.

Quick Start Setup Actions

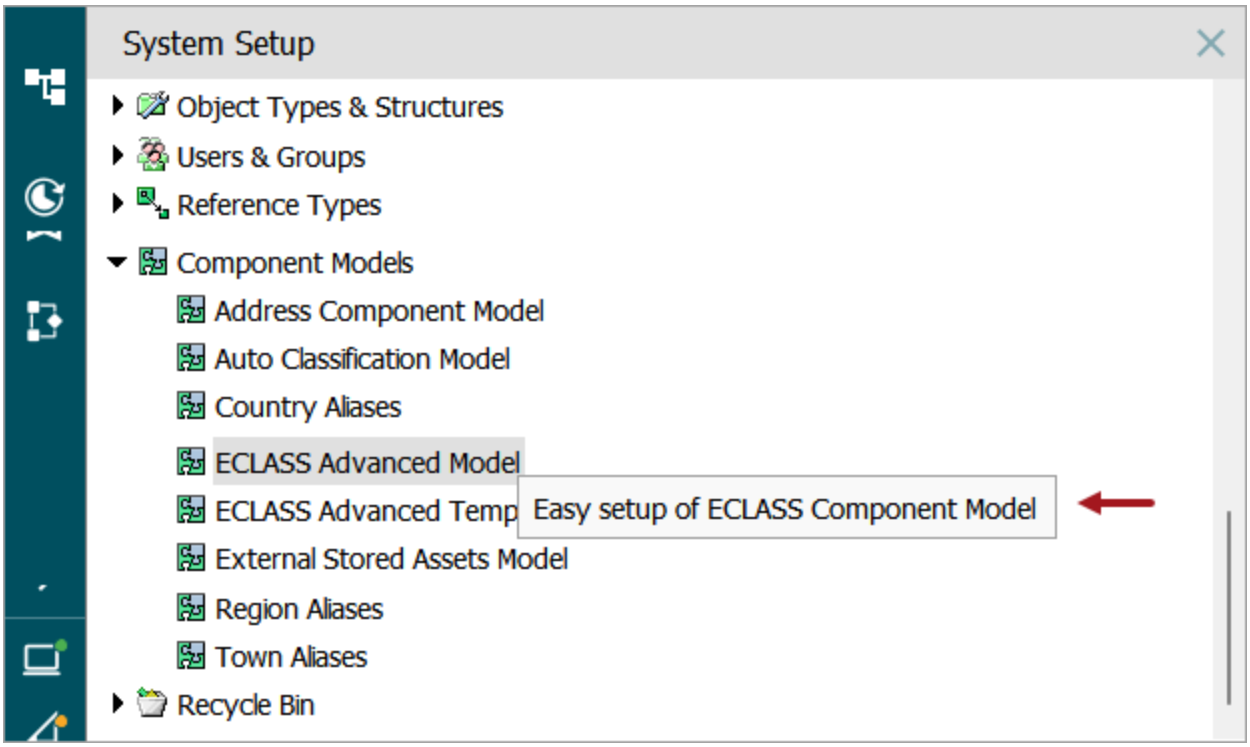
Below are the required setup actions:

1. Run Easy Setup of ECLASS Advanced Industry Standard
2. Prepare the Language Dimension Mapping
3. Update IIEPs
4. Initial Setup for ECLASS Advanced Editor Filtering

> **Important:** The setup actions must be performed in the order in which they are listed. All steps are considered required for the setup, unless explicitly stated otherwise.

Run Easy Setup of ECLASS Advanced Industry Standard

The ECLASS Advanced industry standard includes an Easy Setup wizard that creates elements necessary to support the standard's core functionality. The process involves running an action called 'Easy setup of ECLASS Component Model,' which is available within the ECLASS Advanced Model component model.



Running this action creates the elements needed to support the data model consisting of object types, hierarchies, attributes, LOVs, and references.

For detailed information on the elements created by the Easy Setup action, refer to Elements Created by Easy Setup Action topic in the **ECLASS Advanced Quick Start Setup for Admins** section of the **ECLASS Advanced** documentation.

Note: If an object already exists on a system, running Easy Setup will not change it. Therefore, users can run Easy Setup as needed to deploy new functionality, without risk of disrupting or changing current processes. If manual changes have been made to an object following creation by Easy Setup, these changes are retained if setup is subsequently re-run. This also means that when enablement of new functionality requires changing the setup of an existing object, that change must be made manually on existing implementations (while Easy Setup can handle it automatically in new implementations). Because of this, it is important to pay attention to information included in patch notes and carry out any manual configurations needed to enable new functionality.

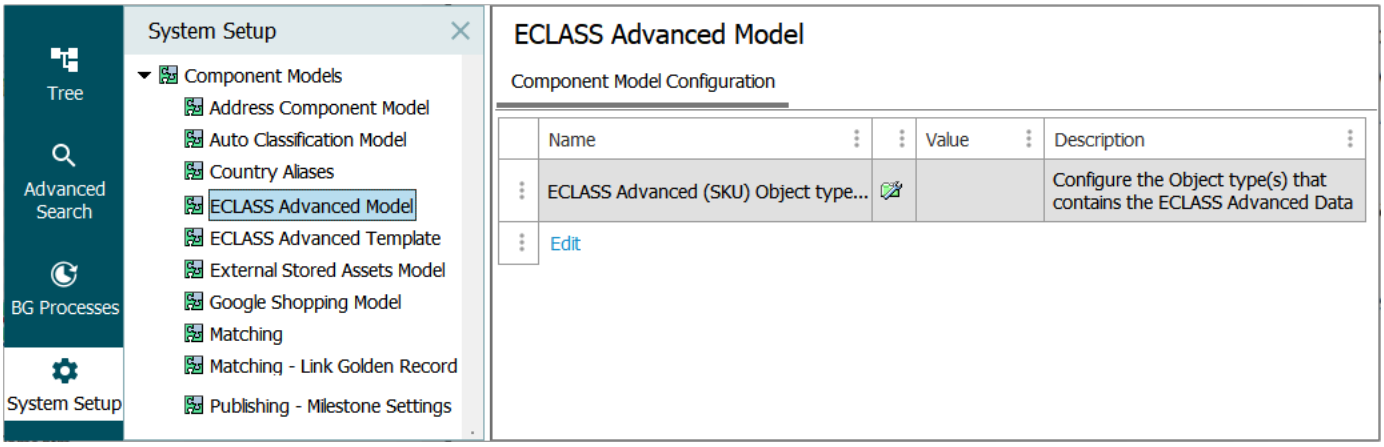
Configuration Steps

The following steps describe how to configure ECLASS Advanced industry standard using the Easy Setup method.

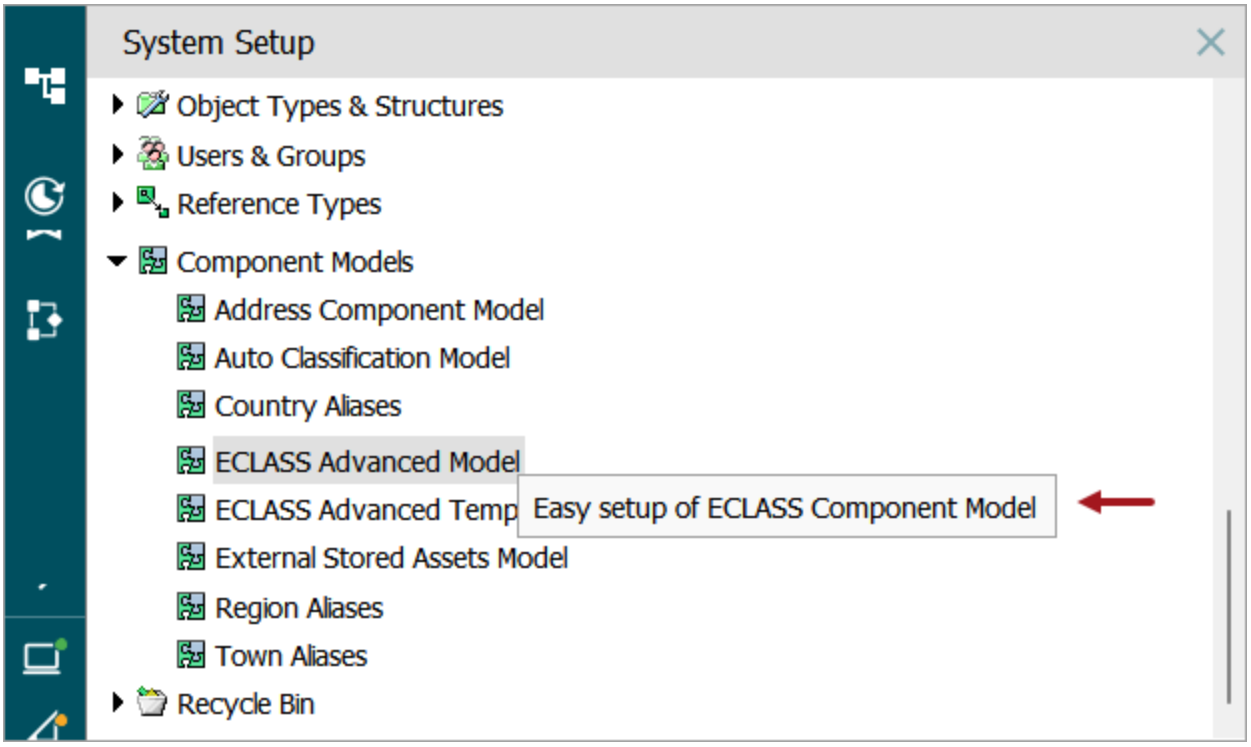
1. Go to **Context** and select the relevant context option from the dropdown.

Note: Consider your relevant STEP Context before you trigger the Easy Setup. Pay attention to the language-dependent data such as attribute names and others. If you perform the Easy Setup in a second attempt using a different STEP context, the imported data from the first attempt will not be updated.

2. Go to **System Setup** > **Component Models** > **ECLASS Advanced Model**

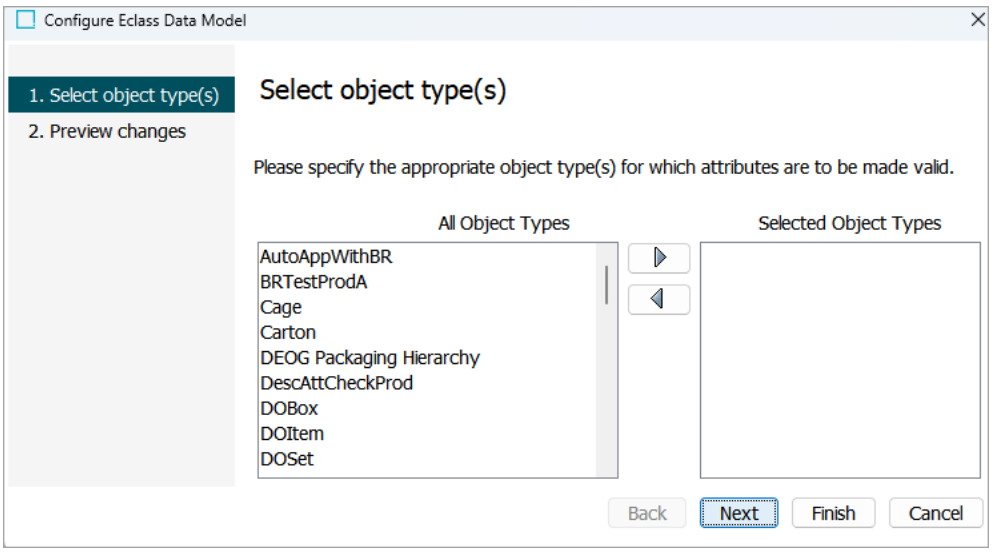


3. Right-click on the **ECLASS Advanced Model** and select **Easy setup of ECLASS Component Model**.

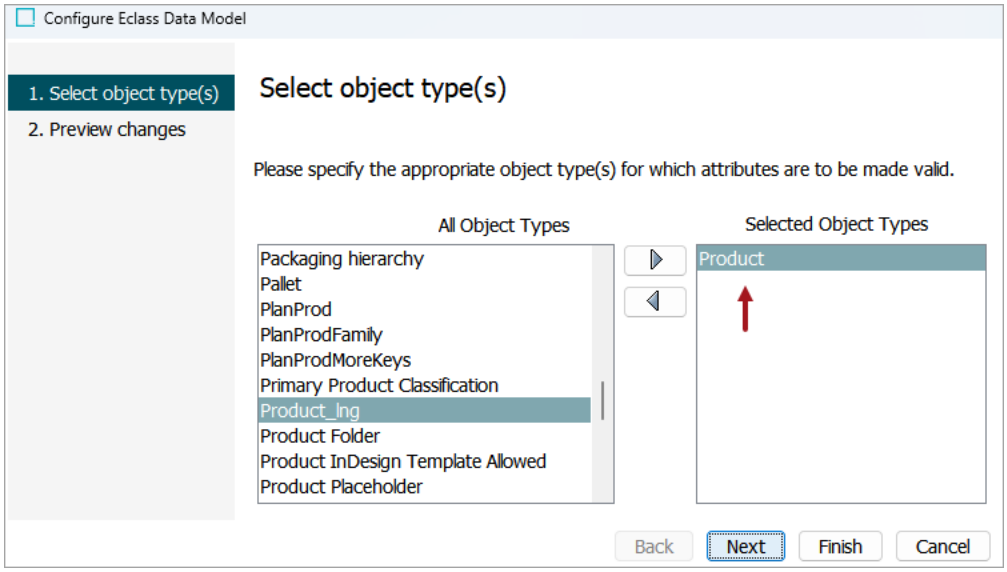


4. The Configure Eclass Data Model dialog will display, prompting the users to select the object type(s) for which the ECLASS attributes are to be made valid.

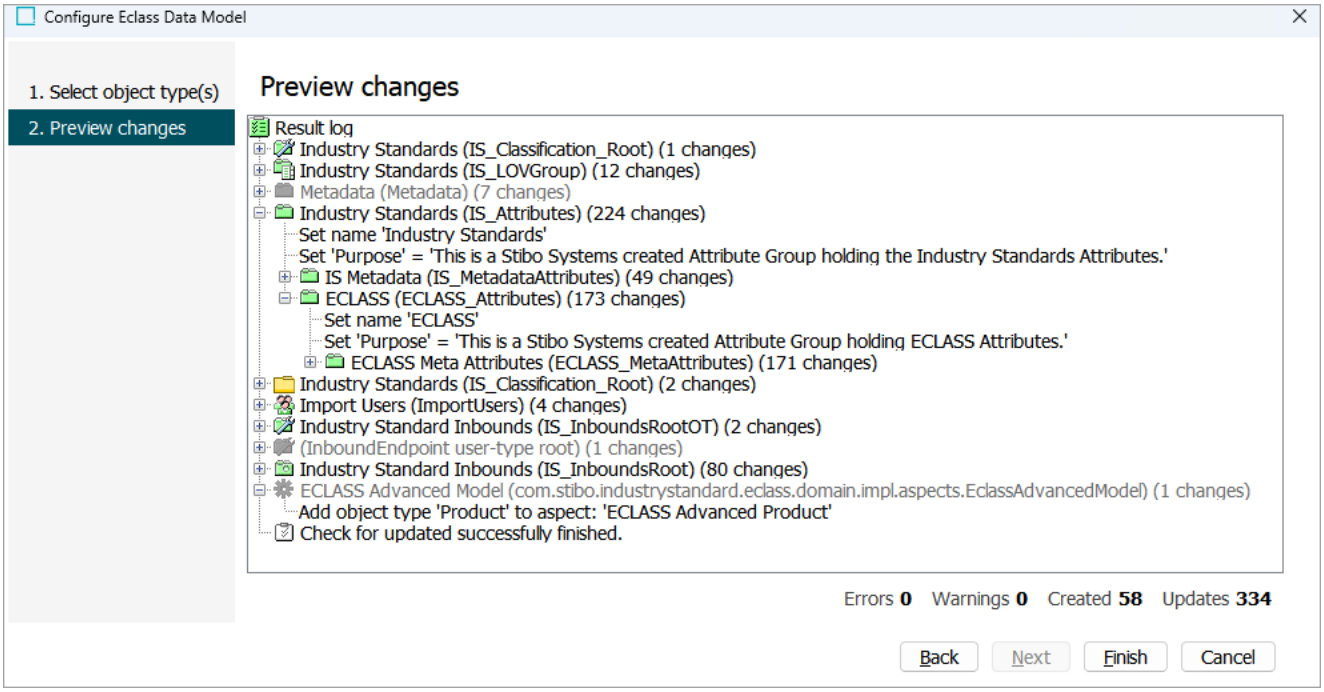
   Select an object type from the left panel (All Object Types) and then click the right arrow ( ). The selected object type will be listed in the right column (Selected Object Types). It is also possible to add multiple object types.



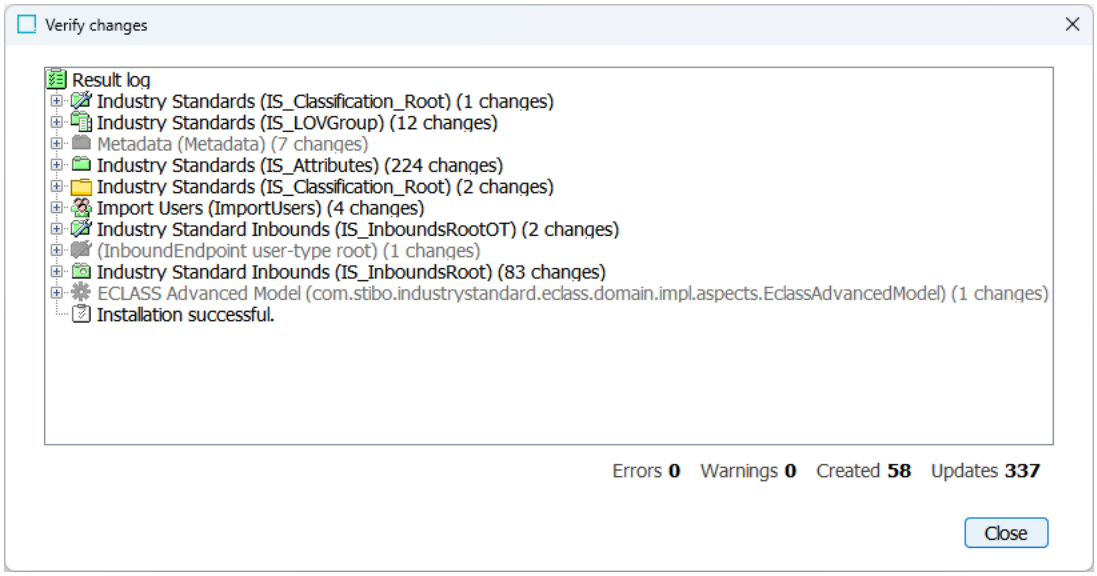
In the screenshot below, the object type 'Product' is selected.



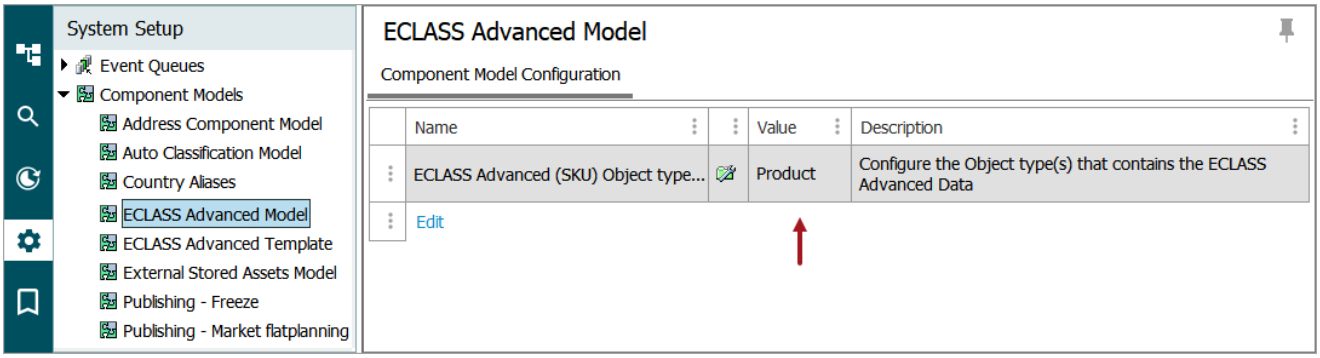
5. When all the required object types are included, click the **Next** button. The next step will display, stating the changes that will be made by running the process. You can expand each of the element to see the details of objects, attributes, and LOVs being created. If you would like to record the changes, you may do so by taking a screenshot of the dialog.



6. When you are ready to start the configuration process, click the **Finish** button. The system will create all necessary elements to support the applicable process. This will typically take less than a minute, and when complete, a dialog will display listing each change that was made.



The selected object types that are mapped to the ECLASS Advanced Model component model will be listed within the Value column of the Component Model Configuration window.



Elements Created by Easy Setup Action

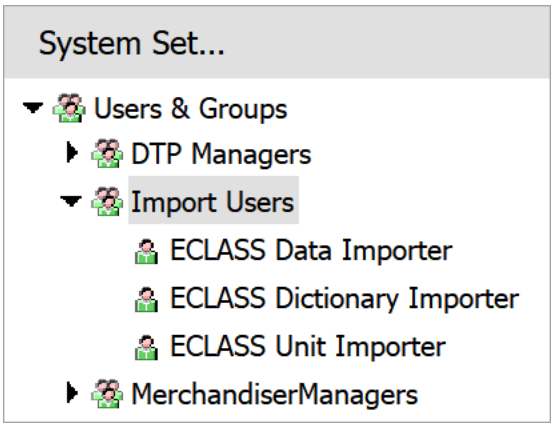
This topic explains the elements created by the Easy Setup actions available within the 'ECLASS Advanced Model' component model. Running the ' ECLASS ADVANCED Easy Set-up' action of the 'ECLASS Advanced Model' component model automatically creates and configures the elements described below.

Users

The following users are created along with a user group called Import Users (ID = ImportUsers):

USER Name	USER ID
ECLASS Unit Importer	ECLASS_UNITIMPORTERUSER
ECLASS Dictionary Importer	ECLASS_DICTIONARYIMPORTERUSER
ECLASS Data Importer	ECLASS_DATAIMPORTERUSER

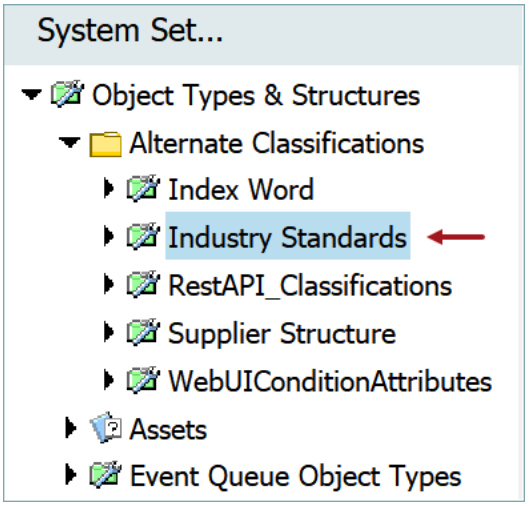
Below is a screenshot of the users created in the system:



Note: Administrators must review and adjust the privileges assigned to Import users created through Easy Setup actions. Currently, these privileges are overly broad, which can expose the data model to unauthorized modifications.

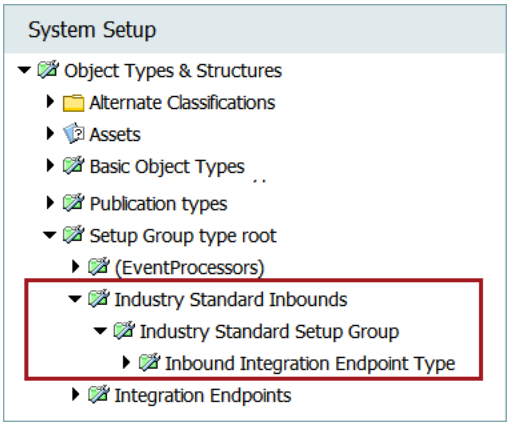
Object Type Model

The following Classification Object Type is created:



Object Type Name	Object Type ID
Industry Standards	IS_Classification_Root

A Setup Group Object Type is created that holds industry standard's integration endpoints.

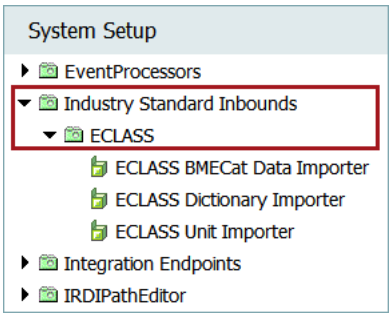


Object Type Name	Object Type ID
Industry Standard Inbounds	IS_InboundsRootOT

Object Type Name	Object Type ID
Industry Standard Setup Group	IS_InboundsSetupGroupOT
Inbound Integration Endpoint Type	InboundEndpoint user-type root

Setup Groups

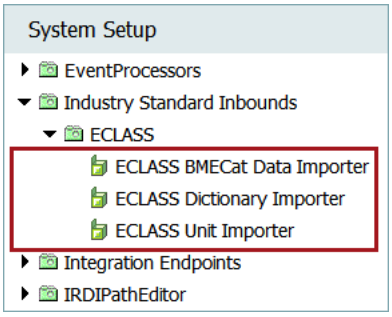
A setup group root and setup group under which the inbound integration endpoints are created.



Setup Group	Setup Group ID
Industry Standard Inbounds	IS_InboundsRoot
ECLASS	ECLASS_InboundsSetupGroup

Inbound Integration Endpoints

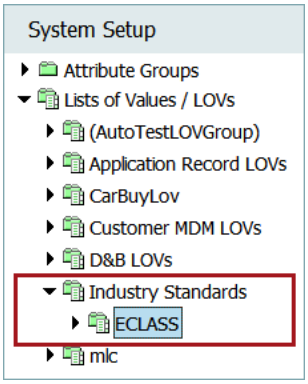
The following inbound integration endpoints are created:



Inbound Integration Endpoint Name	Inbound Integration Endpoint ID
ECLASS Dictionary Importer	ECLASS_DictionaryImporter
ECLASS Unit Importer	ECLASS_UnitImporter
ECLASS BMECat Data Importer	ECLASS_DataImporter

LOV Group creation

The following LOV group is created:



LOV Group Name	LOV Group ID
Industry Standards	IS_LOVGroup
ECLASS	ECLASS_LOVGroup

Attributes (ECLASS-specific and Generic Industry Standard)

Easy Setup actions create both ECLASS-specific attributes and generic attributes that can be used with other industry standards.

ECLASS-Specific Relevant Metadata Attributes:

The following attributes, which are relevant to the ECLASS standard, are created:

Attribute Name	Attribute ID	Description
ECLASS Attribute Type	ECLASS_AttributeType	This is a Stibo Systems-created metadata attribute holding the Type of the property based on the ECLASS definition.
ECLASS Data Container Attribute Order	ECLASS_DCAttributeOrder	This is a Stibo Systems-created metadata attribute that determines the order of the attributes shown in the Web UI within the data container (Level Type and Axis Type).
ECLASS Data Container Link Type Attribute	ECLASS_ DataContainerLinkTypeAttribute	Determines which attribute is used to hold the metadata information for the Data Container Link. Used for Level Type and Axis Type data containers.
ECLASS Depends On	ECLASS_DependsOn	This is a Stibo Systems-created metadata attribute holding the ID of the attribute that the current attribute is dependent on.
ECLASS Hierarchical Position	ECLASS_HierarchicalPosition	This is a Stibo Systems-created metadata attribute holding the ID of the hierarchical position.

Attribute Name	Attribute ID	Description
ECLASS IRDI	ECLASS_IRDI	This is a Stibo Systems-created metadata attribute holding the ECLASS IRDI.
ECLASS Polymorphism Controlling Value	ECLASS_ PolymorphismControllingValue	This is a Stibo Systems-created metadata attribute holding the value that determines the Polymorphism of the current level.
ECLASS Polymorphism Defining Property	ECLASS_ PolymorphismDefiningProperty	This is a Stibo Systems-created metadata attribute holding the information of the Polymorphisms attribute of the specific Block.
ECLASS Polymorphism Definition Combined	ECLASS_ PolymorphismDefinitionCombined	This is a Stibo Systems-created metadata attribute holding the information of the Polymorphisms attribute of the specific Block plus the value ID that needs to be selected to get to this block. Separator is pipe (|).
ECLASS Property Domain Type	ECLASS_PropertyDomainType	This is a Stibo Systems-created metadata attribute holding the property domain type based on the

Attribute Name	Attribute ID	Description
		ECLASS definition.
ECLASS Reference Target	ECLASS_ReferenceTarget	This is a Stibo Systems-created metadata attribute holding the ID of the Block this attribute is referencing.
ECLASS Unit of the Offset	ECLASS_OffsetUnit	This is a Stibo Systems-created metadata attribute holding the Offset unit of the Axis Type Property which is dependent on the classification where it is used.
ECLASS Unit of the Rotation	ECLASS_RotationUnit	This is a Stibo Systems-created metadata attribute holding the Rotation unit of the Axis Type Property which is dependent on the classification where it is used.
ECLASS Value Meaning Attribute	ECLASS_ValueMeaning	This is a Stibo Systems-created metadata attribute holding the Value Meaning for the ECLASS definition.

Generic Attributes:

The following attributes are generic and relevant to the ECLASS standard, but can also be used with other industry standards:

Attribute Name	Attribute ID	Description
IS Deprecation	IS_Deprecation	This is a Stibo Systems-created Industry Standard metadata attribute holding the information about deprecation. For ECLASS Advanced, deprecated elements are not imported or used during an import. The deprecation attribute is therefore not utilized by ECA.
IS Description	IS_Description	This is a Stibo Systems-created Industry Standard metadata attribute holding the Description.
IS ISO-639-2	stibo_ISO-639-2	This is a Stibo Systems-created Industry Standard metadata attribute holding the Language ISO-639-2 (3 alfa, eg. ENG, DEU) code for language dimension points. Using the ISO standard https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes.
IS Keyword	IS_Keyword	This is a Stibo Systems-created industry standard metadata attribute holding the keyword.
IS Name	IS_Name	This is a Stibo Systems-created industry standard metadata attribute holding the name.
IS Order Number	IS_OrderNumber	This is a Stibo Systems-created metadata attribute holding the property order number.

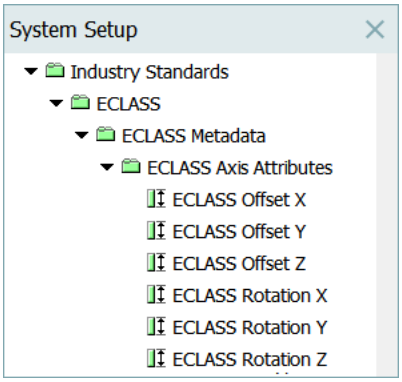
Import Version attribute

This attribute stores information about the ECLASS version file from which the object was imported.

Attribute Name	Attribute ID	Description
ECLASS Import Version	ECLASS_EclassVersion	This is a Stibo Systems-created attribute o display the ECLASS Import Version.

AXIS Attributes

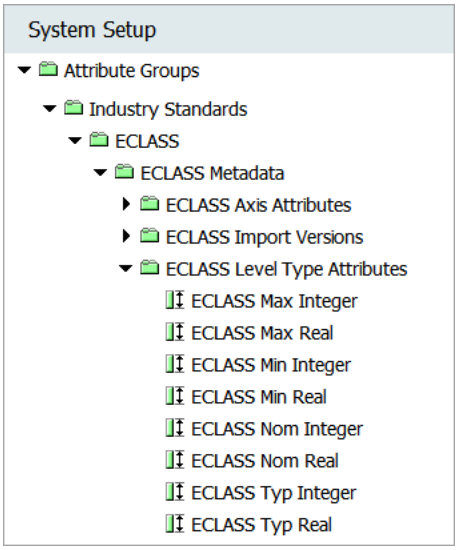
The following AXIS attributes are created:



Attribute Name	
ECLASS Offset X	ECLASS_OffsetX
ECLASS Offset Y	ECLASS_OffsetY
ECLASS Offset Z	ECLASS_OffsetZ
ECLASS Rotation X	ECLASS_RotationX
ECLASS Rotation Y	ECLASS_RotationY
ECLASS Rotation Z	ECLASS_RotationZ

Static Level Type Attributes

The following Static Level Type attributes are created:

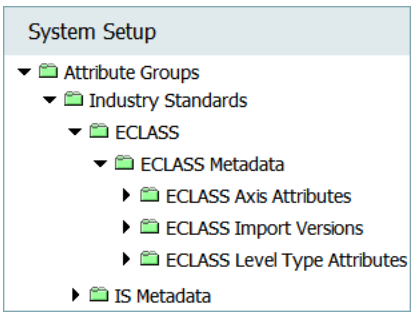


Attribute Name	
ECLASS Max Integer	ECLASS_LT_Int_Max
ECLASS Max Real	ECLASS_LT_Real_Max
ECLASS Min Integer	ECLASS_LT_Int_Min
ECLASS Min Real	ECLASS_LT_Real_Min
ECLASS Nom Integer	ECLASS_LT_Int_Nom
ECLASS Nom Real	ECLASS_LT_Real_Nom
ECLASS Typ Integer	ECLASS_LT_Int_Typ
ECLASS Typ Real	ECLASS_LT_Real_Typ

Relevant Root Nodes

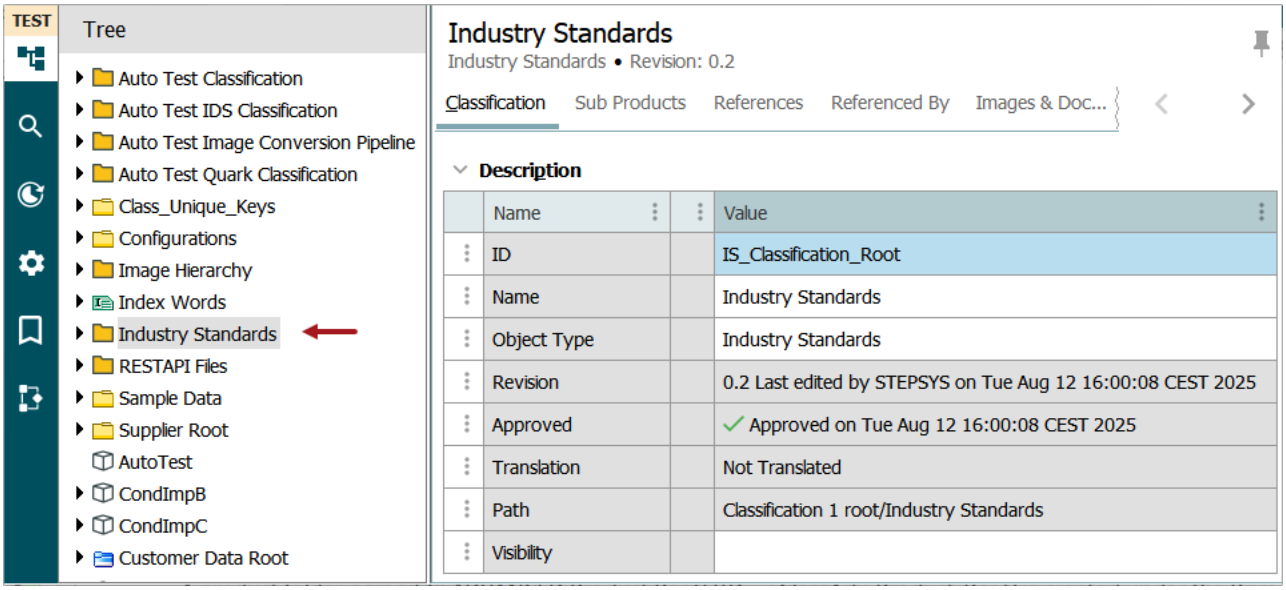
Attribute Group creation:

The following attribute groups are created:



Attribute Group Name	Attribute Group ID
Industry Standards	IS_Attributes
ECLASS	ECLASS_Attributes
ECLASS Metadata	ECLASS_MetaAttributes
ECLASS Axis Attributes	ECLASS_AXISAttributes
ECLASS Import Versions	ECLASS_EclassVersions
ECLASS Level Type Attributes	ECLASS_LevelTypeAttributes
IS Metadata	IS_MetadataAttributes

A classification folder named 'Industry Standards' (ID = IS_Classification_Root) is created, as shown in the screenshot below.


TEST
Tree
Auto Test Classification
Auto Test IDS Classification
Auto Test Image Conversion Pipeline
Auto Test Quark Classification
Class_Unique_Keys
Configurations
Image Hierarchy
Index Words
Industry Standards
RESTAPI Files
Sample Data
Supplier Root
AutoTest
CondImpB
CondImpC
Customer Data Root
Industry Standards
Industry Standards • Revision: 0.2
Classification
Sub Products
References
Referenced By
Images & Doc...
Description
Name
Value
ID
IS_Classification_Root
Name
Industry Standards
Object Type
Industry Standards
Revision
0.2 Last edited by STEPSYS on Tue Aug 12 16:00:08 CEST 2025
Approved
Approved on Tue Aug 12 16:00:08 CEST 2025
Translation
Not Translated
Path
Classification 1 root/Industry Standards
Visibility

ECLASS Name and ID Pattern

All ECLASS-related data and configuration objects created in STEP during Unit and Dictionary imports follow a specific ID pattern. These patterns are typically applied when objects are generated automatically, either through imports or from an upstream system.

This topic explains the ID pattern STEP uses when creating or updating objects. Understanding this pattern helps users know how IDs are generated and enables them to define custom ID patterns, if needed, to ensure uniqueness during imports.

Note: Objects created by Easy Setup actions have predefined names and IDs. Do not modify these unless absolutely necessary. For reference, objects created by Easy Setup actions are listed in the Elements Created by Easy Setup Action topic.

Defining the Name and ID Pattern

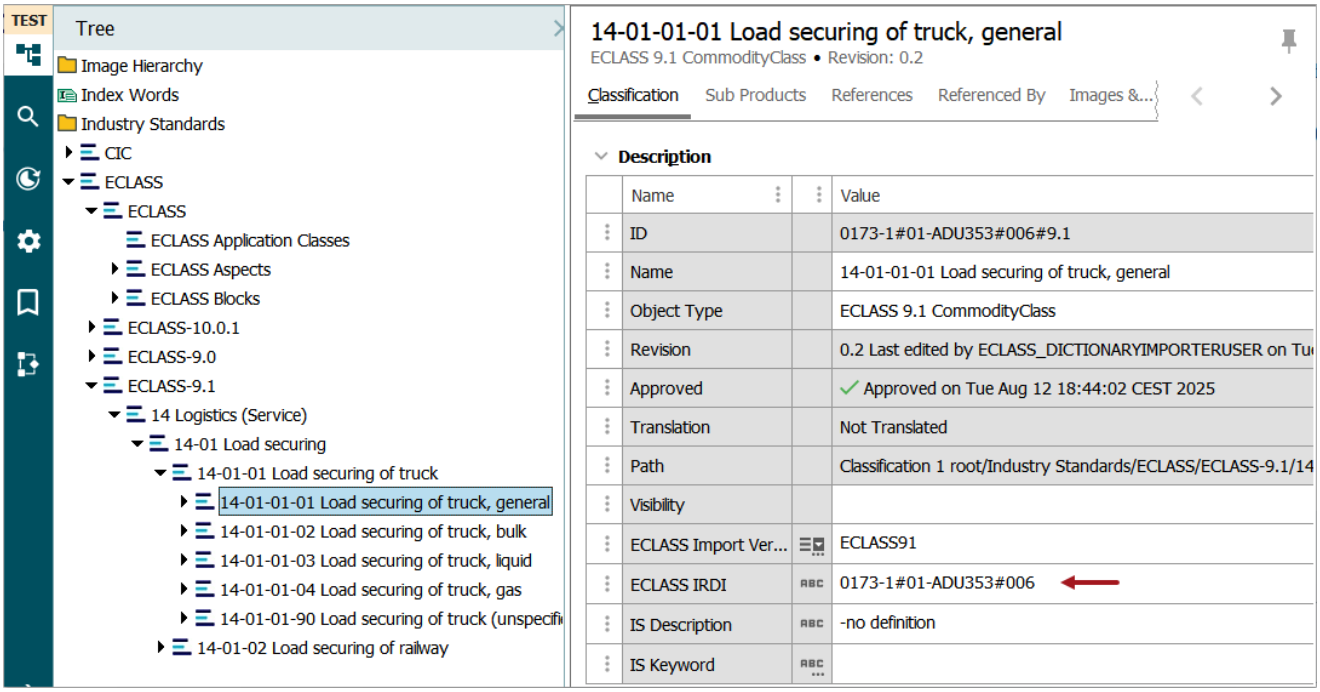
The Name and ID of created objects follow a defined pattern. These patterns resemble the ECLASS IRDI (International Registration Data Identifier) code because the ID string is derived from text within the IRDI code. To understand STEP's ID pattern, users must first be familiar with the IRDI code.

IRDI

ECLASS assigns globally unique identifiers (IRDI – International Registration Data Identifier) to every object in its standard. IRDI complies with international standards ISO/IEC 11179-6, ISO 29002, and ISO/IEC 6523. It consists of multiple segments:

In STEP, each object's IRDI is stored in the attribute ECLASS IRDI (ID = ECLASS_IRDI).

An example of an IRDI stored in an object is displayed in the screenshot below:



IRDI for objects created by the Dictionary Importer contains three segments, separated by '#'

- First Segment: This is comprised of the International Code Designator (ICD) per ISO/IEC 6523, followed by the Organization Identifier (OI).

  For example, in the screenshot above, 0173 represents ECLASS. Other registered organizations include ISO (0112), ODETTE (0177), SIEMENS (0175), GTIN (0160), etc. The updated list of registered organizations is available at http://iso6523.info/icd_list.pdf.

- Second Segment: The Code Space Identifier (CSI) identifies the object type (e.g., 01 for classification class, 02 for property), followed by the Concept Code. In the example above, 01 indicates a class, and ADU353 is the unique identifier for that class.

  The list of CSIs for ECLASS is available at https://eclass.eu/support/technical-specification/structure-and-elements/irdi.

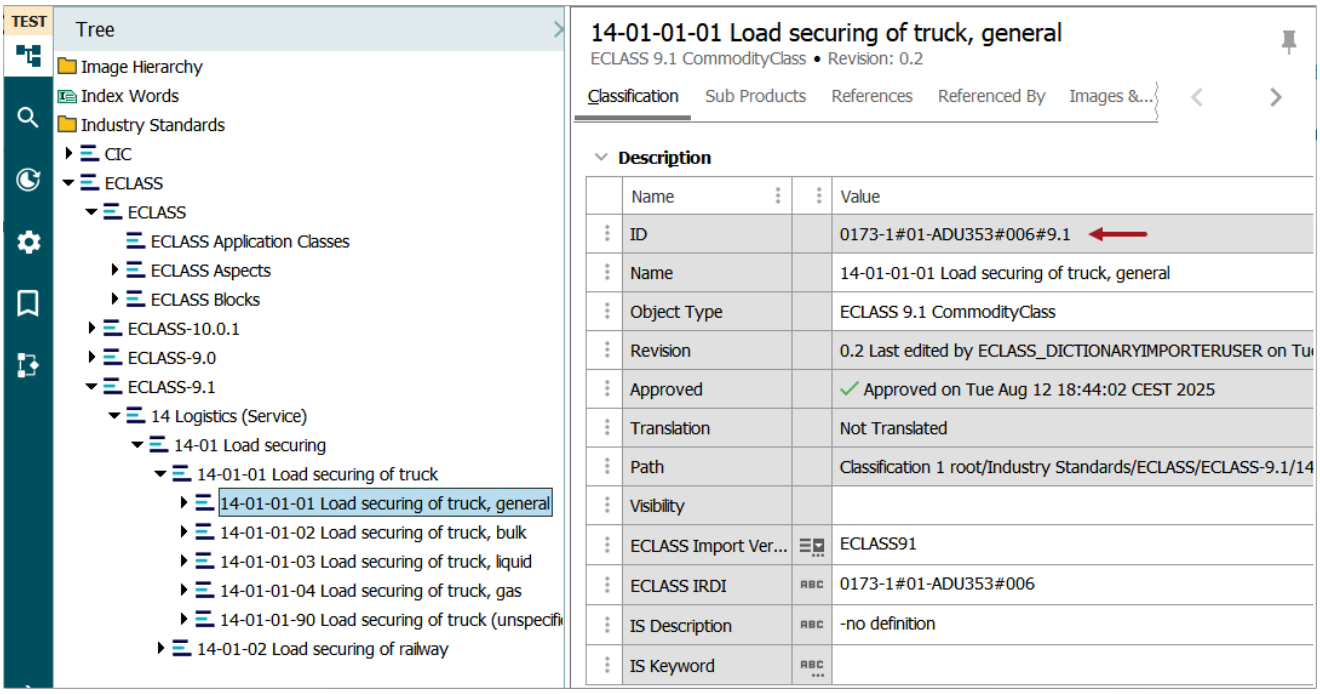
- Third Segment: The version identifier (e.g., 001 for the first version). In the example, 006 means the object has undergone six revisions.

STEP ID

Although IRDI is unique, it cannot be used directly as a STEP ID because STEP IDs are limited to 40 characters, whereas IRDI can exceed this length. Therefore, STEP derives the ID from selected IRDI segments.

In addition to the three IRDI segments, STEP IDs may include a fourth segment for classification objects. This segment indicates the ECLASS version from which the object originated.

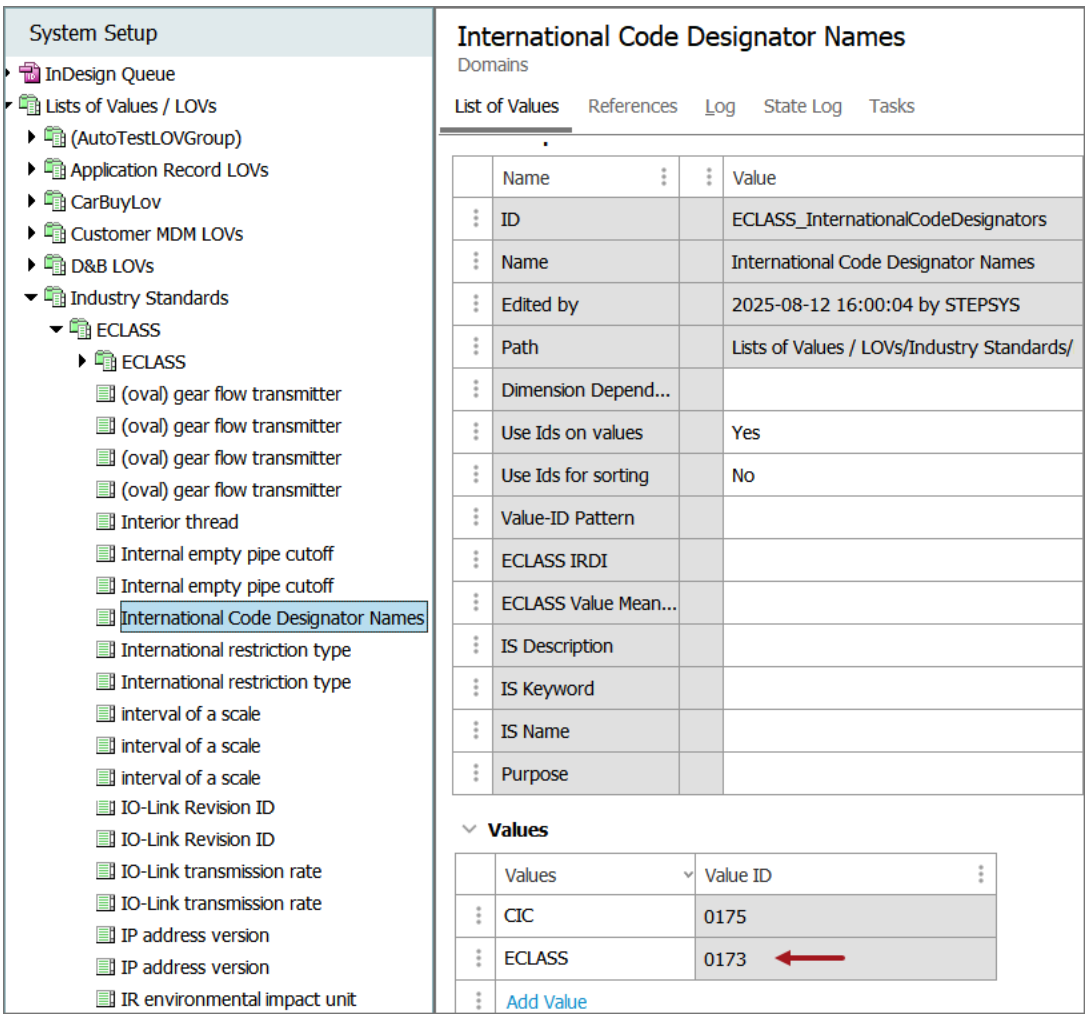
In the example below, the first three segments come from the IRDI, and the fourth segment (e.g., 9.1) shows the ECLASS version.



Importing non-ECLASS dictionary files

You can also import dictionary files belonging to registered organizations listed at http://iso6523.info/icd_list.pdf. However, ensure that the International Code Designator (ICD) within STEP is updated before proceeding.

By default, the system includes the International Code Designator (ICD) for the ECLASS organization (0173) in the list of values (LOV) named **International Code Designator Names**(ID = ECLASS_ InternationalCodeDesignators). Before importing dictionary files from other organizations, update this LOV as shown in the screenshot below. For instructions on adding values to an LOV, refer to the Adding Values to an LOV documentation.


System Setup
InDesign Queue
Lists of Values / LOVs
(AutoTestLOVGroup)
Application Record LOVs
CarBuyLov
Customer MDM LOVs
D&B LOVs
Industry Standards
ECLASS
ECLASS
(oval) gear flow transmitter
(oval) gear flow transmitter
(oval) gear flow transmitter
(oval) gear flow transmitter
Interior thread
Internal empty pipe cutoff
Internal empty pipe cutoff
International Code Designator Names
International restriction type
International restriction type
interval of a scale
interval of a scale
interval of a scale
IO-Link Revision ID
IO-Link Revision ID
IO-Link transmission rate
IO-Link transmission rate
IP address version
IP address version
IR environmental impact unit
International Code Designator Names
Domains
List of Values
References
Log
State Log
Tasks
Name
Value
ID
ECLASS_InternationalCodeDesignators
Name
International Code Designator Names
Edited by
2025-08-12 16:00:04 by STEPSYS
Path
Lists of Values / LOVs/Industry Standards/
Dimension Depend...
Use Ids on values
Yes
Use Ids for sorting
No
Value-ID Pattern
ECLASS IRDI
ECLASS Value Mean...
IS Description
IS Keyword
IS Name
Purpose
Values
Values
Value ID
CIC
0175
ECLASS
0173
Add Value

Prepare the Language Dimension Mapping

Prior to importing the Unit or Dictionary file into the system, it is essential to establish a relationship (commonly referred to as mapping) between the ECLASS language code and the language dimension points in the system. Familiarity with the ECLASS language codes contained within the ECLASS XML file is essential for this process.

Within the Unit file, the language code will be specified within the 'content_description' tags, as depicted below. The Unit file always comes in English and German languages.

```
<content_description language_code="ENG">eCl@ss 10.1</content_
description>
```

Although the Unit file contains both English and German languages, Unit Groups and Units are exclusively created in English. Consequently, it is recommended to import Units into the Global context. Alternatively, if there is a desire to import Units in German, users should switch the tag as follows:

From:

```
<content_description language_code="ENG">ECLASS12.0</content_description>
```

To:

```
<content_description language_code="DEU">ECLASS12.0</content_description>
```

The Dictionary file, the language code may vary depending on the specific file being imported.

For the file that is in English, the language code within the 'content_description' tags will be as

```
<content_description language_code="ENG">eCl@ss 10.1 ADVANCED</content_
description>
```

For French, the language code within the 'content_description' tags will be as

```
<content_description language_code="FRA">eCl@ss 10.1 ADVANCED</content_
description>
```

For German, the language code within the 'content_description' tags will be as

```
<content_description language_code="DEU (or GER)">eCl@ss 10.1
ADVANCED</content_description>
```

Language codes within the content_description tags are automatically managed when the administrator maps ISO 639-2 codes to each language dimension in STEP. Ensure that no duplicate codes exist in the mapping.

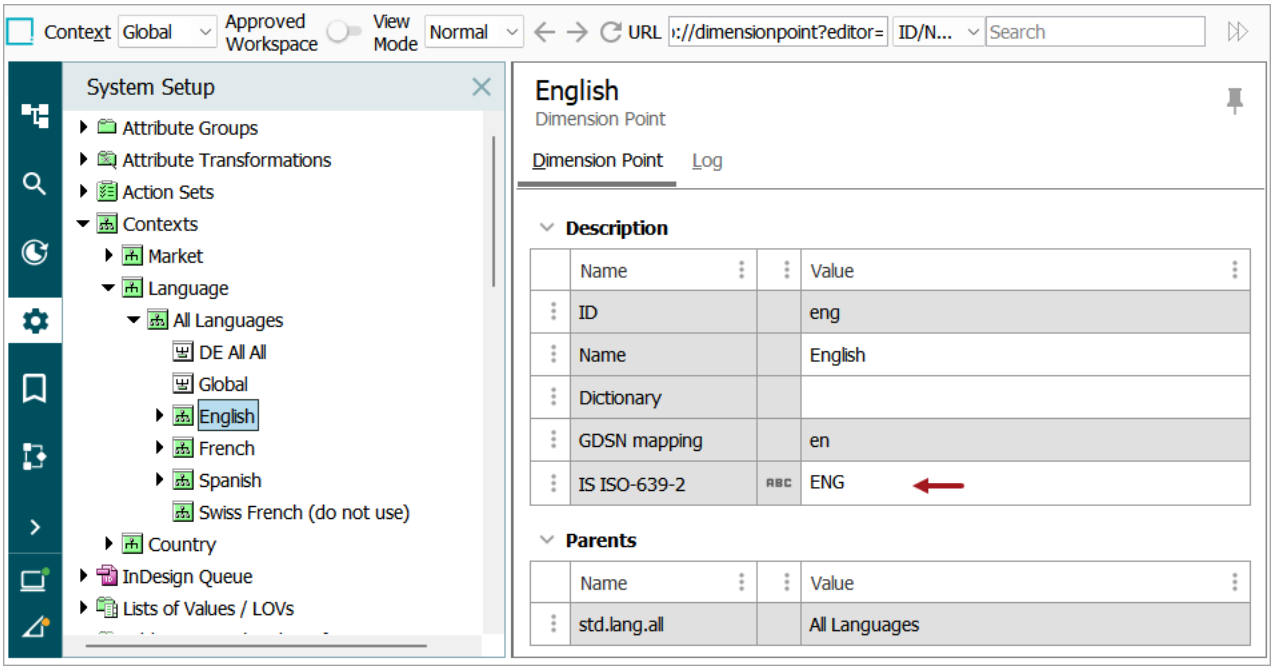
Context (Dimension Point) Identification

Mapping ECLASS Advanced language codes is a straightforward process. As a system administrator, you need to enter the appropriate language code value into the metadata attribute IS ISO-639-2 (ID = IS_ISO_639_2) for the desired language dimension point. By following these guidelines, you can effectively identify and import language-specific data into the appropriate contexts.

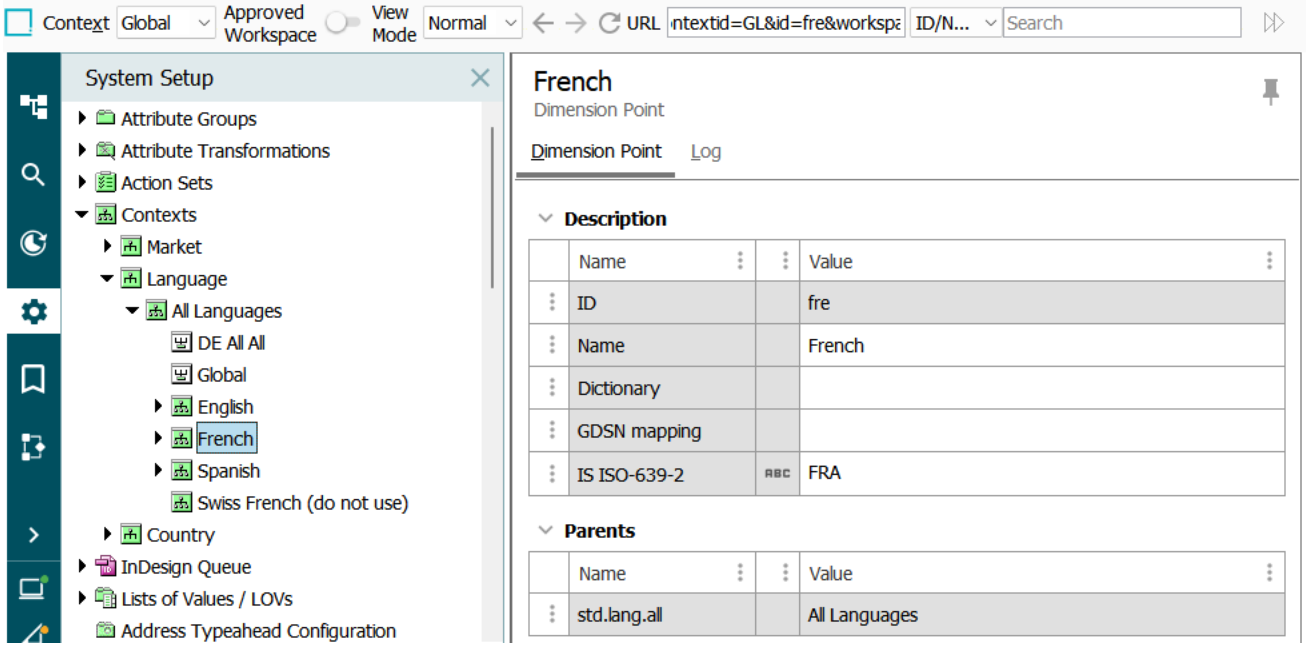
The examples below assume that the relevant language dimension points already exist in the system. If a language dimension point is not available, you will need to create one. For more information on how to create a language dimension point, refer to refer to Maintaining Dimensions and Dimension Points documentation.

Example:

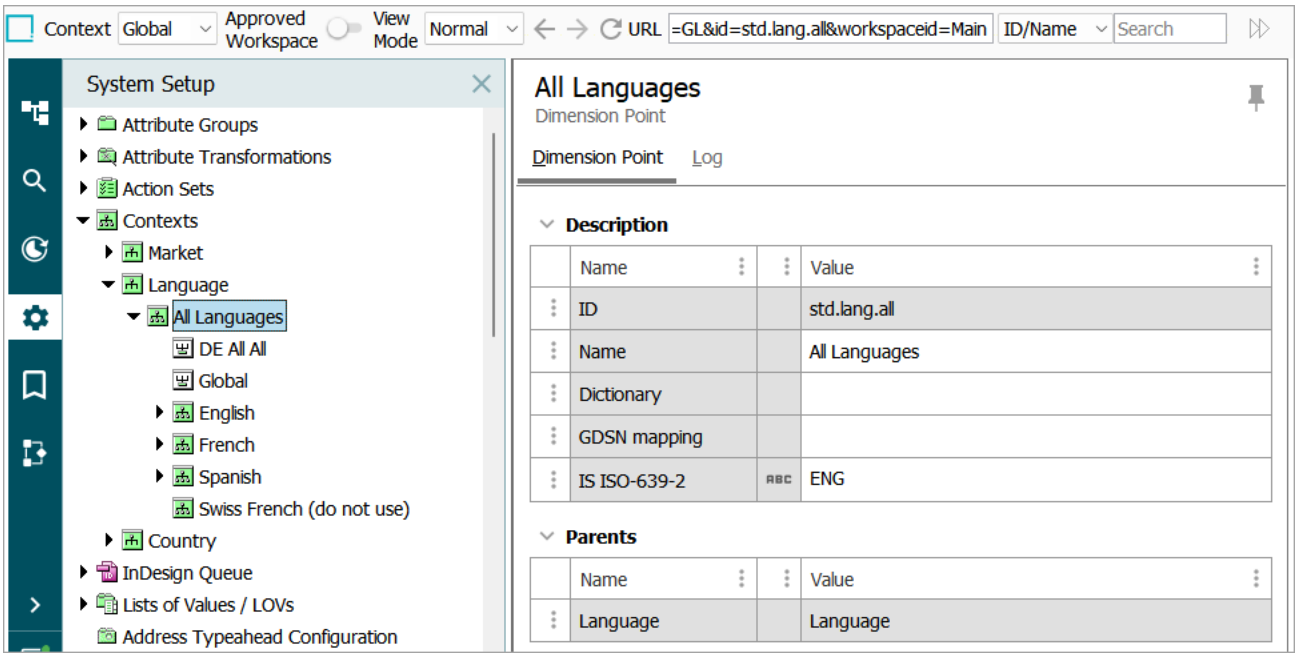
In the following example, the language code 'ENG' is applied to the English language dimension. Consequently, language-dependent data will be imported in the English language and associated with the context that holds the English dimension point.



For ECLASS Advanced packages in the French language, the language code 'FRA' should be applied to the French language dimension point. As a result, language-dependent data will now be imported in the French language, associated with the context holding the French dimension point.

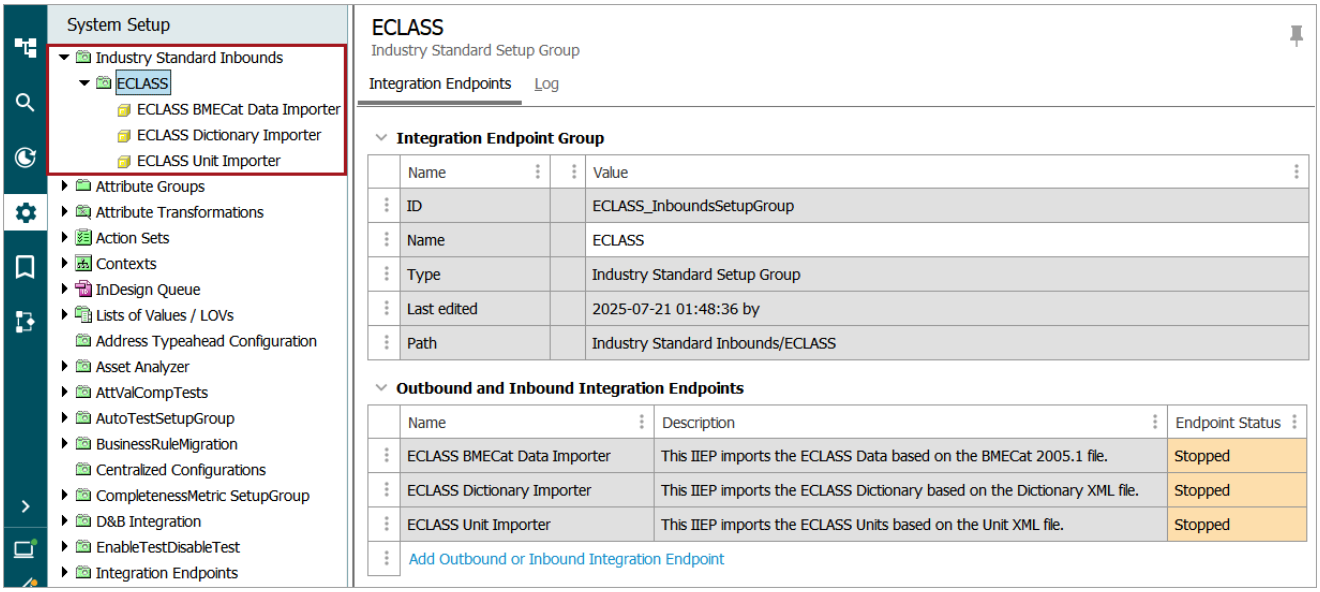


Note: As shown in the example image below, selecting the All Languages option (which acts as the parent for all other language dimension points) and populating the metadata attribute IS ISO-639-2 with a language code ensures that language-dependent data are inherited across all dimension points.



Update IIEPs

The Easy Setup action creates three inbound integration endpoints for each file type supported by the solution. These endpoints can be found under Industry Standard Inbounds → ECLASS within the System Setup tab.



Although many options specific to the IIEPs are set and configured, it is imperative to manually set up certain parameters to ensure the IIEPs are fully prepared for usage.

While the 'ECLASS Unit Importer' and the 'ECLASS Dictionary Importer' are ready for immediate use, the 'ECLASS BMECat Data Importer' may require further configuration, as detailed below:

Configuration Steps

Update 'ECLASS BMECat Data Importer' Endpoint Parameters:

1. Right-click on the ECLASS BMECat Data Importer endpoint and select Edit Inbound Integration Endpoint. This action will launch the Inbound Integration Endpoint Wizard.

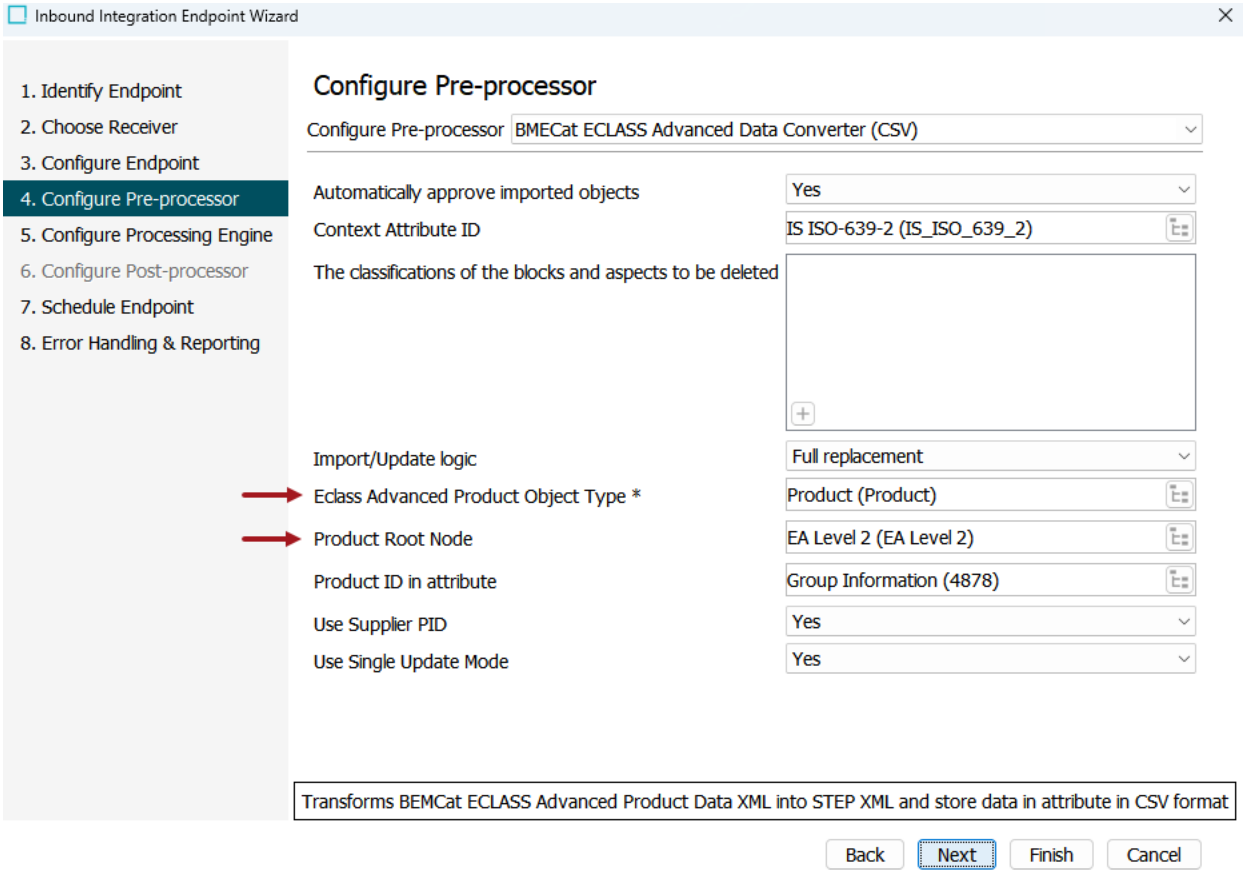
2. Navigate to the 'Configure Pre-processor' tab, where the 'BMECat ECLASS Advanced Data Converter (CSV)' will be displayed as the pre-processor (refer to the screenshot below).

3. Update the following parameter:

- **Product Root Node:**Use the node picker to select the root node under which newly imported products will be placed. This node must be created manually in advance, as it serves as the parent container for all new product records. If this parameter is left empty, the system will only update existing products and will not import any new ones. In the example below, a folder named EA Level 2 is populated.
- **ECLASS Advanced Product Object Type:** Use the node picker next to this parameter to find and select the primary SKU / product Object Type.

  The selected Object Type must align with ECLASS data validity requirements. Ensure that it matches the object type configured during the 'Easy setup of ECLASS Component Model' action (within the Easy Setup). For details on the object type configured during Easy Setup, refer to the Run Easy Setup of ECLASS Advanced Industry Standard topic.

  In the screenshot below, an object type named Product is used as an example.



Only information pertaining to initial setup of the 'ECLASS ADVANCED Data Importer' is explained in this topic. For more detailed information on the rest of the parameters, refer to Configuring an IIEP for ECLASS Advanced Data Imports topic within the **ECLASS Advanced Importers** section of the **ECLASS Advanced Reference Guide**.

Setup ECLASS Advanced Editor Templates

ECLASS Advanced Templates are a set of rules that define the filtering criteria along with additional validation principles. These are the predefined structures that groups a specific set of Aspects, Blocks, and Properties tailored for a particular ECLASS Class (i.e., product category).

Using the filtering mechanism requires a one-time setup, including creating the necessary elements (object type, reference type and a classification folder) and configuring the component model.

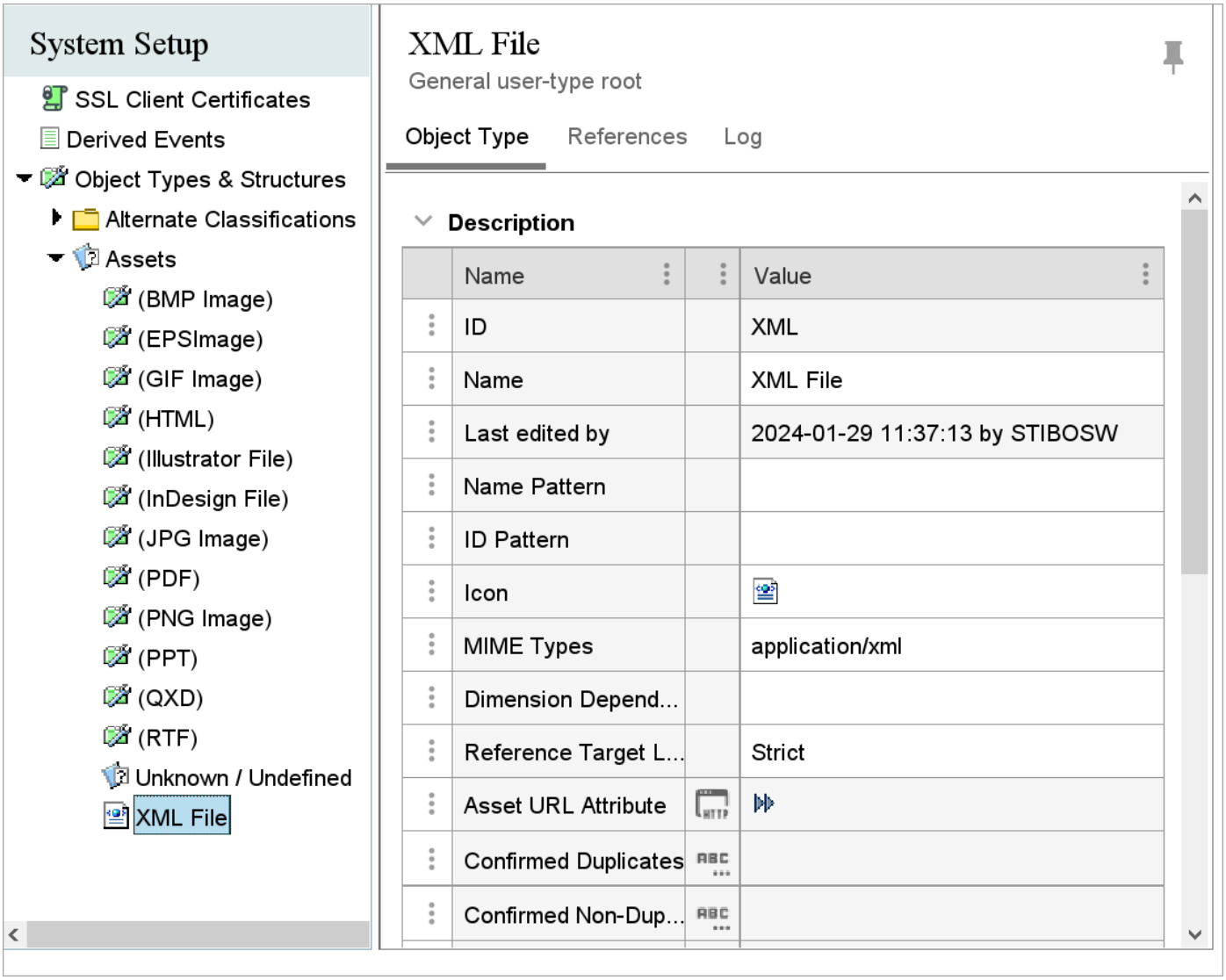
This topic covers each of the items required to fully set up the handling of the ECLASS Templates:

- Create the Required Elements
- Configure the ECLASS Advanced Filter Model

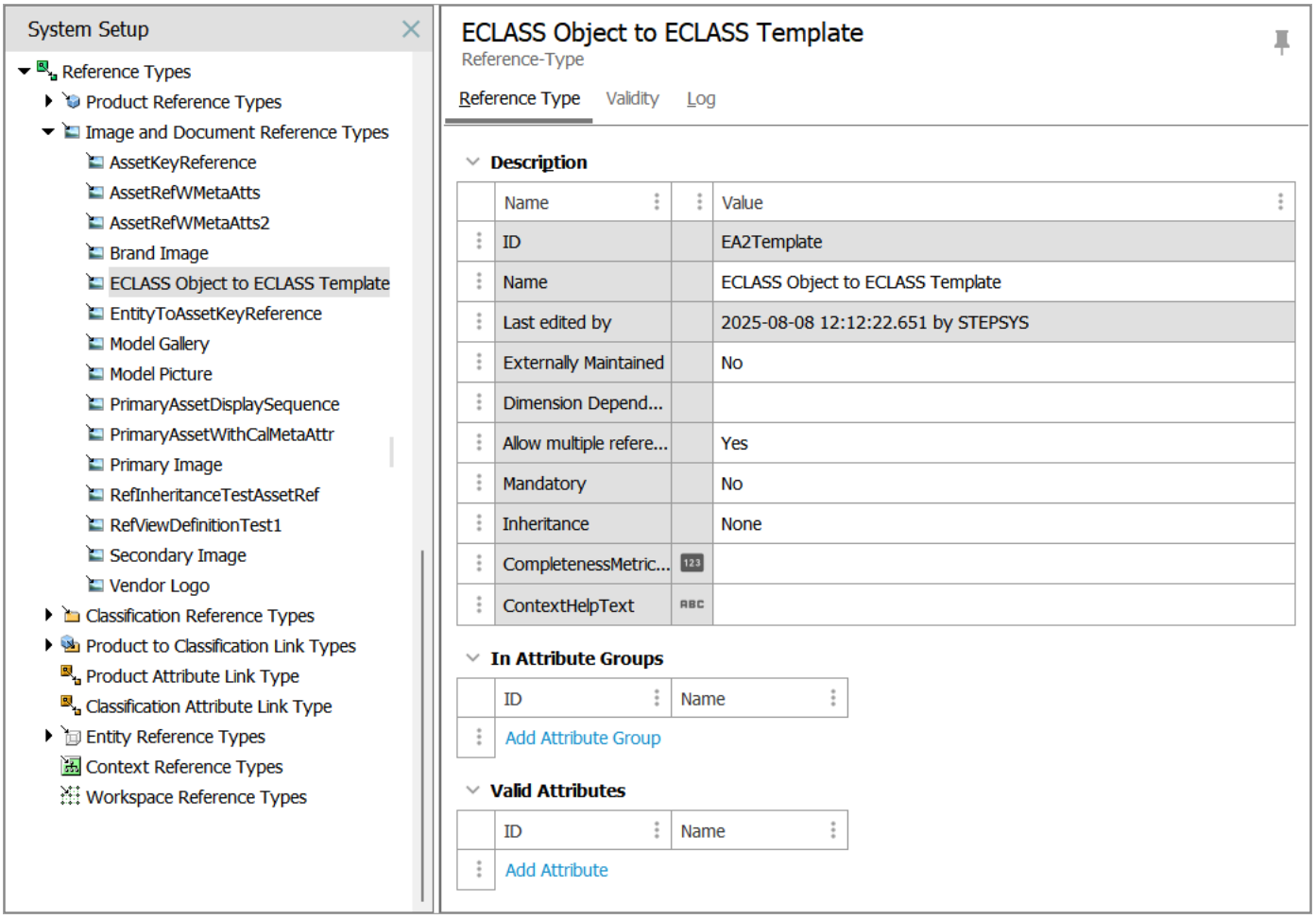
Create the Required Elements

To work with a template, you must first create an Asset object type for the template and at least one Classification folder to store it. Follow the procedure below to complete this setup.

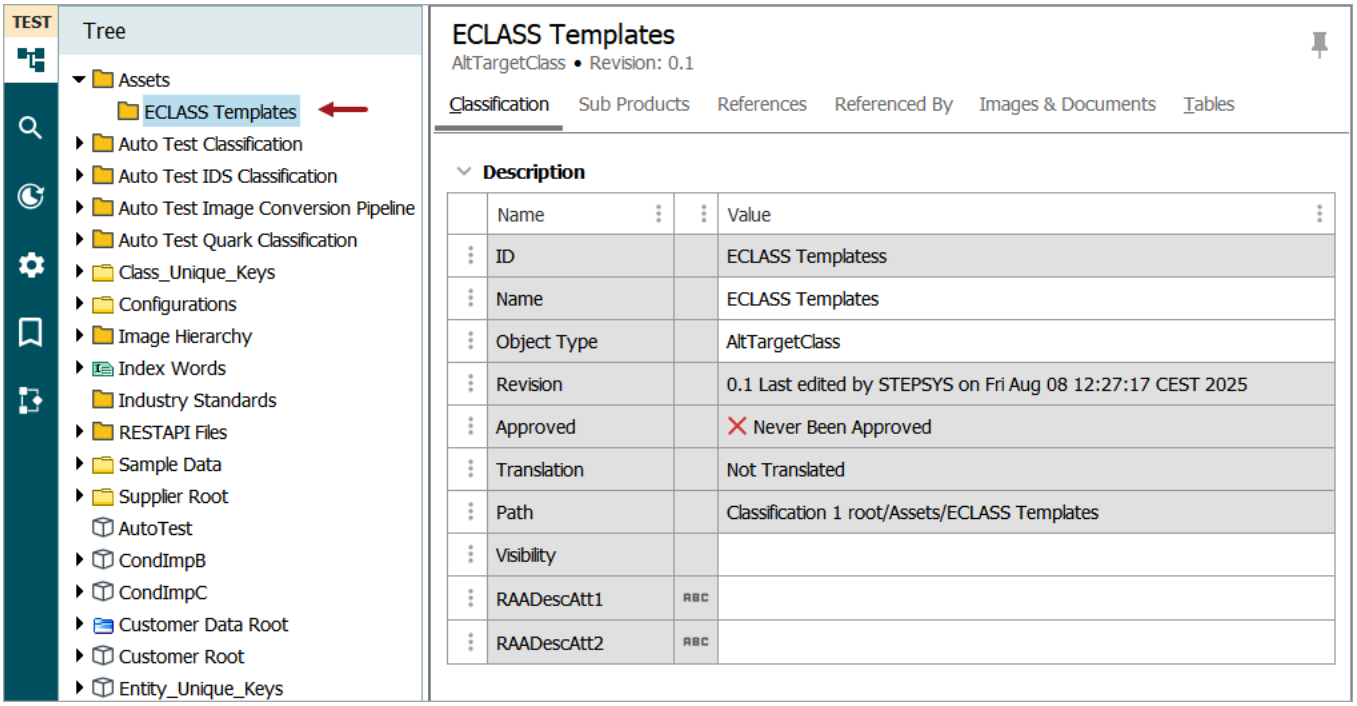
1. In System Setup, create one or more asset object types to support creating the XML assets stored in the assets folder. For information on creating the object type, Creating an Object Type topic in the System Setup documentation.



2. In System Setup, create a new asset reference type to hold the relation between product and the XML asset. For information on creating a reference type see Creating a Reference Type topic in the System Setup documentation.



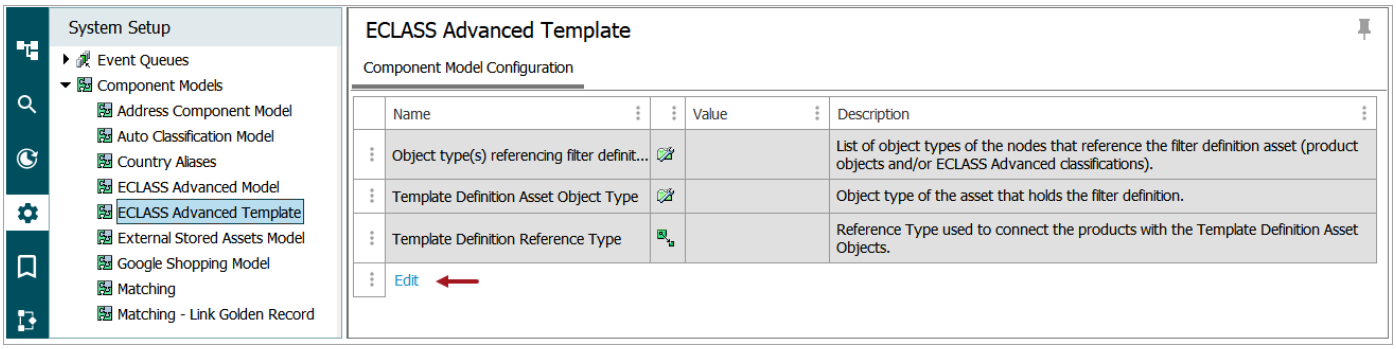
3. On the Tree tab, create a classification hierarchy to store assets in STEP. For information on creating a classification folder, see Classifications topic in the Getting Started documentation.



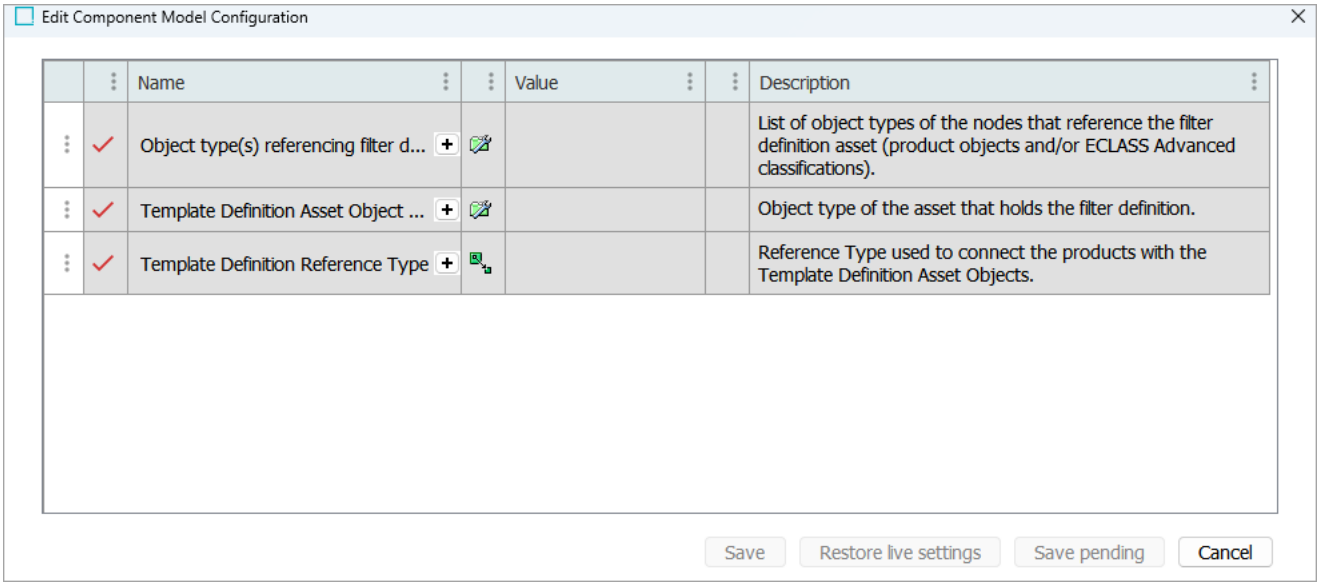
Configure the ECLASS Advanced Template Model

Set the newly created asset object type, reference type, and product object type in the component model.

1. In System Setup, open the Component Models node, select the ECLASS Advanced Template node, and click the **Edit** link to open the editor.

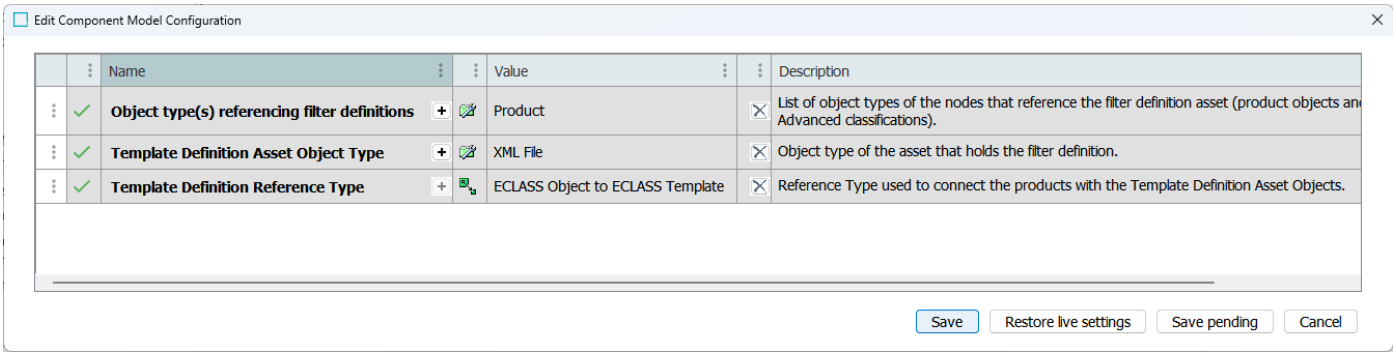


2. Double-click the plus button (+) to display a dialog to assign the required objects to the model.



- Object type(s) referencing filter definitions: Choose the object types of the nodes that reference the ECLASS Templates asset. This can be the object type of product objects and/or ECLASS ADVANCED classifications.
- Template definition asset object type: Choose the asset object type(s) that represent the XML assets (ECLASS Templates) in STEP.
- Template definition reference type: Choose the reference type that is used to connect the products with the ECLASS Template asset objects.

3. Make any required corrections:



- The plus button () remains active for parameters that allow multiple selections, such as the 'Template definition asset object type' parameter and the 'Object type(s) referencing filter definitions' parameter above. Select another object type if required.
- Double-click the delete button () to remove a selection.
- The first column shows the status of the parameter. A green check () means the configuration has no errors; a red X () means additional setup is required.
- Click the **Save pending** button to save your work while error(s) exists.
- Click the **Save** button to save a configuration once it has no errors.
- Click the **Restore live settings** button to undo the changes made to a previously saved and error-free configuration.
- Click the **Cancel** button to undo all changes made in this dialog.

ECLASS Standard Supported Versions and Formats

The following are the supported import and export versions and/or formats for the ECLASS Advanced standard.

- **Data Importer:** BMEcat 2005.1, BMEcat 2005.2
- **Data Exporter:** BMEcat 005.1, BMEcat 2005.2

ECLASS Advanced Reference Guide

This guide describes specific ECLASS Advanced reference material beyond what is provided in the ECLASS Advanced Quick Start Guide topic. This includes a detailed description of user functionalities that are provided with the solution after Easy Setup actions for the ECLASS Advanced standard has been completed by an admin.

This guide addresses the following topics:

- Importing ECLASS Advanced Files
- Exporting in BMEcat Formats using BMECat 2005.1 Converter

Importing ECLASS Advanced Files

The ECLASS Advanced solution provides extensive import capabilities. The intention of the easy setup action of the ECLASS Advanced solution is to provide pre-configured importers for different ECLASS Advanced files. Each customer can subsequently implement their unique validations, business processes, and data management protocols. To do this successfully, it is crucial to understand the import framework described in the ECLASS Advanced Import Framework topic.

Easy Setup only creates the required IIEPs for the importers. However, for the ease of operation it is necessary that you create and configure three File Loading Widgets in the Web UI that can be used for importing ECLASS Advanced files. Each of the ECLASS Advanced file types has their own importer, and more information for each can be found within their respective sections below.

> ⚠ **Important:** Imports must be executed in the specified order, otherwise, the process will fail due to dependencies between imports. Units only need to be imported once per version of ECLASS Advanced (ECA). After that, you can import Segments in any order.

1. ECLASS Advanced Unit Importer
2. ECLASS Dictionary Importer
3. ECLASS BMECat Data Importer

For information on supported versions, refer to the ECLASS Standard Supported Versions and Formats topic.

ECLASS Advanced Import Framework

All ECLASS Advanced IIEPs are configured to use the hotfolders receiver because they are built as preprocessors that utilize the STEP XML import engine. As a result, files imported through the workbench cannot be used for these IIEPs.

Import Process Overview

Once a valid ECLASS Advanced file is uploaded using a File Loading Widget (or uploaded directly to a hotfolder), the file is picked up by an IIEP, and the IIEP starts a background process.

Uploading Files via Web UI

To upload a file to a specific hotfolder via a Web UI, a File Loading Widget can be used. For generic information on how to use a File Loading Widget, refer to File Loading Widget topic. Further, details relevant to specific ECLASS Advanced importers are described in the following topics:

- Using ECLASS Advanced Unit Importer
- Using ECLASS Advanced Dictionary Importer
- Using ECLASS Advanced Data Importer

Uploading Files via the Hotfolder

To upload a file to a specific hotfolder:

1. Navigate to /upload/hotfolders/[Importer].
2. Drop the file into the folder with a name aligning with the importer (e.g., ECLASS ADVANCED Data Importer should have files dropped into the 'EclassAdvDataImporter').
3. The endpoint will pick up the file at the next scheduled polling, and the file load will begin.

Note: The endpoints are created with a default schedule of polling the hotfolder once per minute, but this setting is adjustable for each endpoint, so it may vary between importers and implementations. Stibo recommends using the File Upload widget and setting the IIEP schedule to Never, at least for Unit and Dictionary imports, since these functions are typically used only once a year. Continuous polling every minute would unnecessarily consume system resources.

ECLASS Advanced Unit Importer

The primary objective of the ECLASS Advanced Unit Importer is to offer a convenient out-of-the-box solution for importing Unit data in a supported XML format.

Purpose and Considerations of Unit File Import

The ECLASS Advanced Dictionary file can include Units that may not have been generated during a prior ECLASS Basic import process in STEP. As a result, it is essential to perform a full import of the ECLASS Unit file before proceeding with the import of the ECLASS ADVANCED Dictionary file.

Following are key considerations:

- The ECLASS Basic Unit file and ECLASS Advanced Unit file are identical.
- Typically, the Unit file is a part of the ECLASS-provided .zip archive, which must be manually extracted to access the Unit XML file. The Unit Importer exclusively accepts XML file formats.
- The Unit file may contain Unit Groups without allocated Units. These Unit Groups will be imported into STEP, irrespective of whether Units are present.
- Units within the Unit file may exist without a designated Unit group. Such Units will be listed under a generated Unit group named 'eClass ungrouped units (ID = eClass_UngroupedUnits).'
- STEP does not support structuring Units beyond the Unit Group level. Therefore, all Unit Groups are created on the same level as children of the 'Unit' node.
- The creation of Unit Group and Unit STEP IDs and Names follows the ECLASS Basic structure logic.
- Version dependencies for Unit Groups and Units are not taken into account. The Unit metadata attribute 'EclassImportVersion' will contain the latest version number.
- Unit conversions are not considered, and consequently, no data related to Unit conversions is imported.
- The ECLASS Unit file does not contain any Units related to Currencies. However, Currency Units are managed within the ECLASS Advanced Dictionary file. Therefore, the ECLASS Advanced Dictionary Importer will create the Unit Group called 'eClass Currency (ID = eClass_Currency),' which will include the relevant currency Units found in the Dictionary file, such as EUR.

- Despite the presence of both English and German languages in the Unit file, Unit Groups and Units are exclusively created in English. Consequently, it is recommended to import Units into the Global context. Alternatively, if you intend to import the units in the German language, users should switch the tag as follows:
  - To:

    ```
    <content_description language_code="DEU">ECLASS12.0</content_
    description>
    ```

  - From:

    ```
    <content_description language_code="ENG">ECLASS12.0</content_
    description>
    ```

This section includes information on:

- Using ECLASS Advanced Unit Importer
- Configuring ECLASS Advanced Unit Importer

Configuring ECLASS Advanced Unit Importer

Note: If the Easy Setup actions for the ECLASS Advanced Component model have been completed, the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

The following topics provide the configuration steps necessary to allow users to drag and drop Unit files onto a configured File Loading Widget and monitor the progress of the import file in the created IIEP Background Process in the workbench.

- Configuring an IIEP for ECLASS Advanced Unit Imports
- Configuring a File Loading Widget for ECLASS Advanced Unit Imports

Configuring an IIEP for ECLASS Advanced Unit Imports

> **Note:** If the Easy Setup actions for the ECLASS Advanced Component model have been completed, the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

An inbound integration endpoint (IIEP) can be configured in the workbench to import Unit files into STEP. Once an IIEP is configured for ECLASS Advanced Unit imports, Unit files can be imported after they are uploaded either to a configured hotfolder, or to a File Loading Widget on a Web UI Homepage. For more information, refer to the ECLASS Advanced Unit Importer topic.

This section describes how to configure an IIEP that can allow the automated processing of Unit files. Each screenshot example within this section provides recommended values for the parameters and ECLASS Advanced Unit Importer.

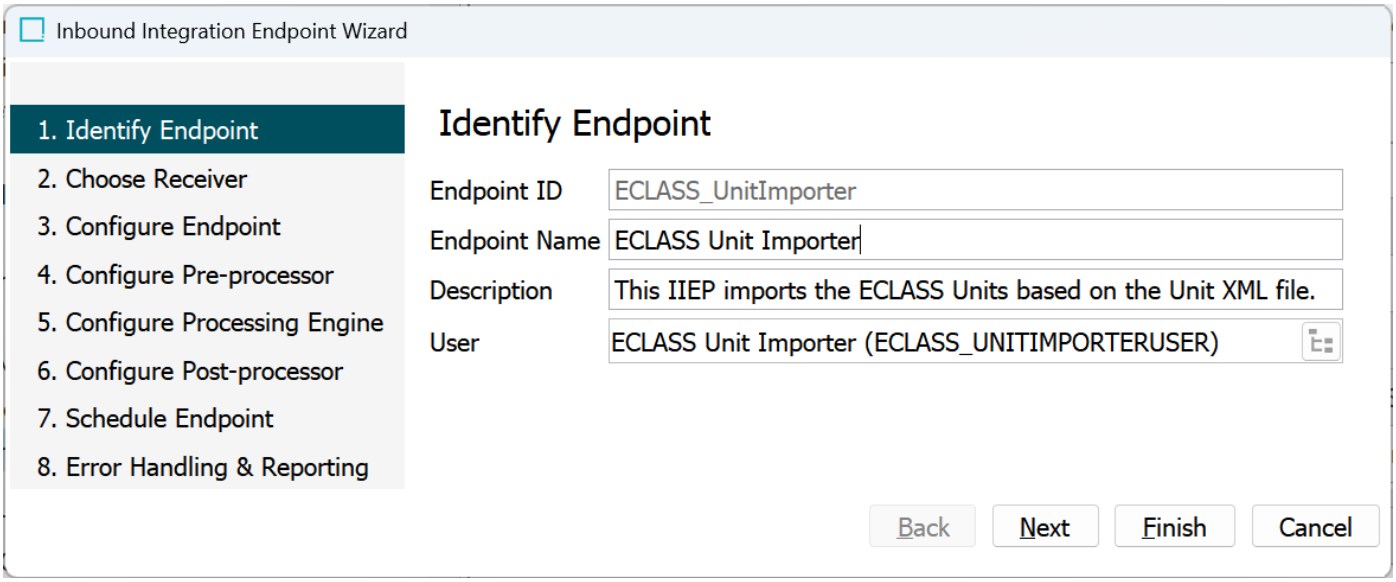
Prerequisites

This topic aims to acquaint users with the IIEP specifically designated for the import of Unit files. It does not cover general IIEP functionalities. It is assumed that individuals configuring an IIEP for ECLASS Advanced Unit Import are well-versed in configuring and processing standard inbound integration endpoints. For a comprehensive understanding of the standard functionalities provided in inbound integration endpoints, refer to Inbound Integration Endpoints topic within the Data Exchange documentation.

Configuration Steps

1. In the workbench, go to System Setup, select and right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.

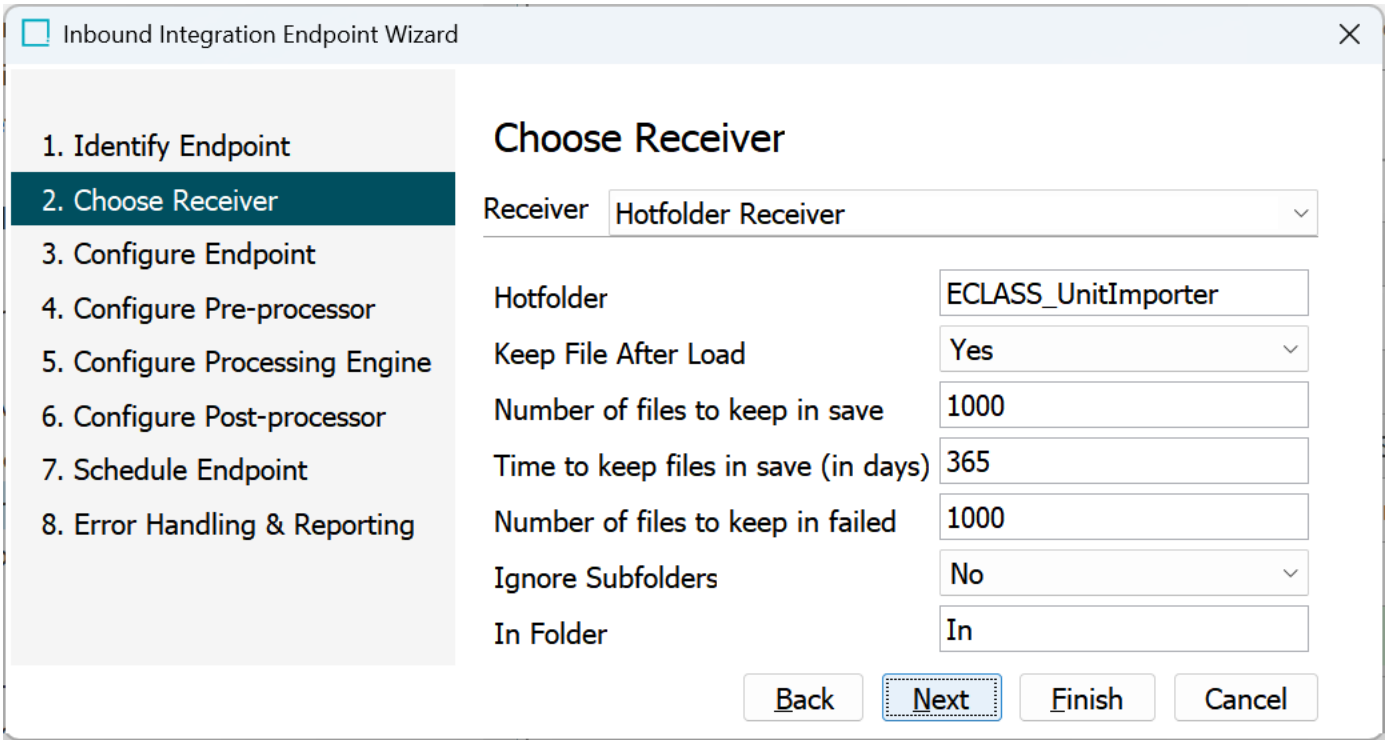
2. On the Inbound Integration Endpoint Wizard, populate the parameters as recommended and shown below.



For more information about the parameters available within the Identify Endpoint step, refer to IIEP - Identify Endpoint topic within the Data Exchange documentation.

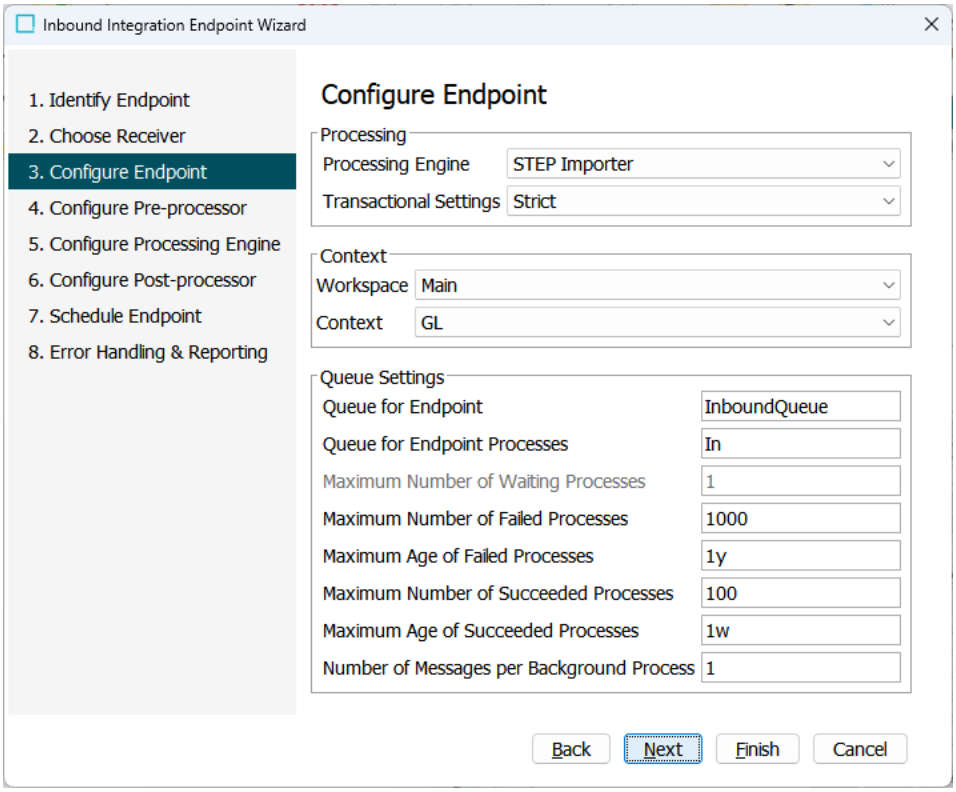
3. Click the **Next** button, and the Choose Receiver parameters will display. The parameters are to be populated as recommended and shown below. The mandatory parameter Hotfolder must be populated with a hotfolder name before the Next button will be enabled. In the screenshot below, the Hotfolder parameter is populated with the value 'EClassAdvUnitImporter.'

> **Note:** The value within this hotfolder parameter will be used to create the new hotfolder once the IIEP Wizard is complete.



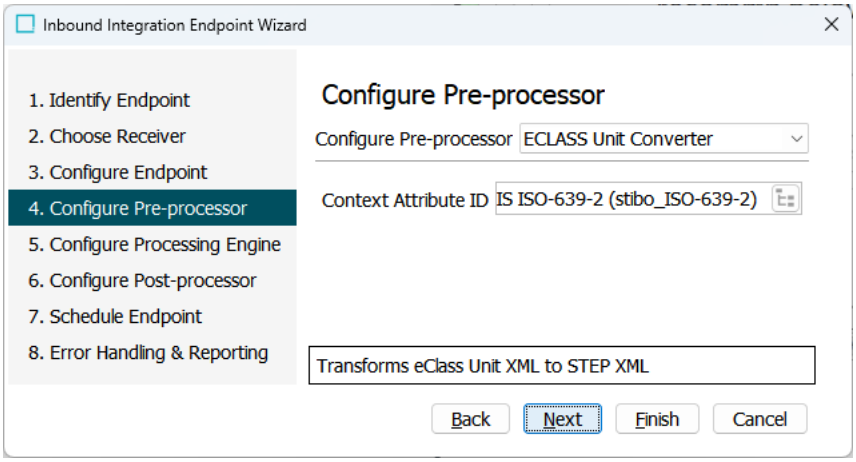
For more information about the parameters, refer to IIEP - Choose Receiver topic within the Data Exchange documentation.

4. Click the **Next** button, and the Configure Endpoint parameters will display. The parameters are pre-populated with the recommended values as shown below.



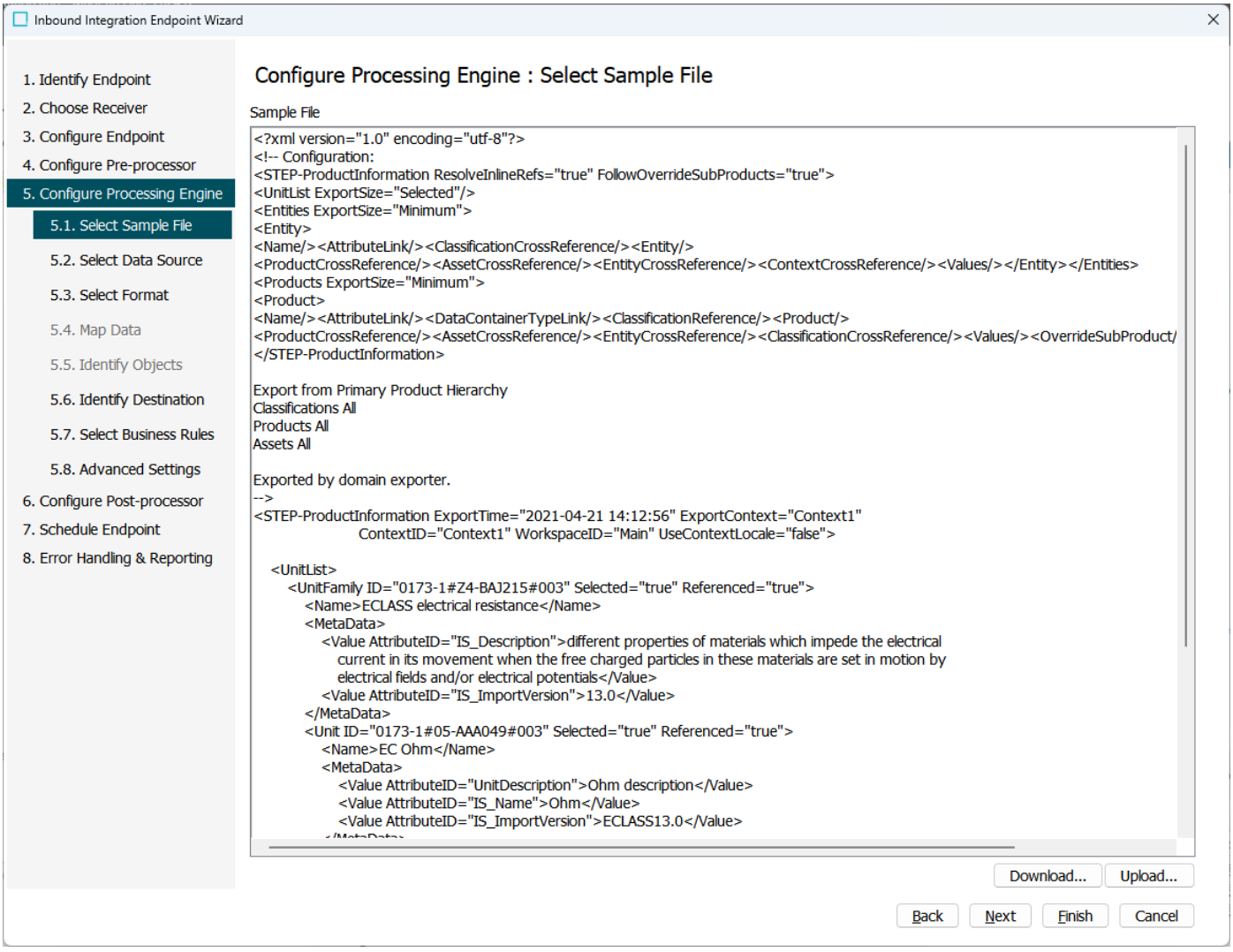
For more information about the parameters, refer to IIEP - Configure Endpoint topic within the Data Exchange documentation.

5. Click the **Next** button, and the Configure Pre-processor parameter will display. The selection of the pre-processor within this step makes the IIEP unique for importing Unit files. The parameters are to be populated as recommended and shown below:



- Configure Pre-processor must be populated with ECLASS Advanced Unit Converter option. This is an exclusive pre-processor for importing Unit files. For more information about the parameter, refer to IIEP - Configure Pre-processor topic within the Data Exchange documentation.
- Context Attribute ID parameter must be populated with the attribute IS ISO-639-2 (ID = stibo_ISO-639-2). This attribute is created by the Easy Setup action and holds the language mappings. For more information about language mappings, refer to Prepare the Language Dimension Mapping topic.

6. Click the **Next** button, and the Configure Processing Engine: Select Sample File step for the STEP Importer processing engine will display.



- In **5.1 Select Sample File** step, click the Upload button to upload a sample STEPXML file. For information about how to upload a sample file, refer to IIEP - Configure STEP Importer Processing Engine topic within the Data Exchange documentation.

  The basic data structure of a sample Unit file is provided below:

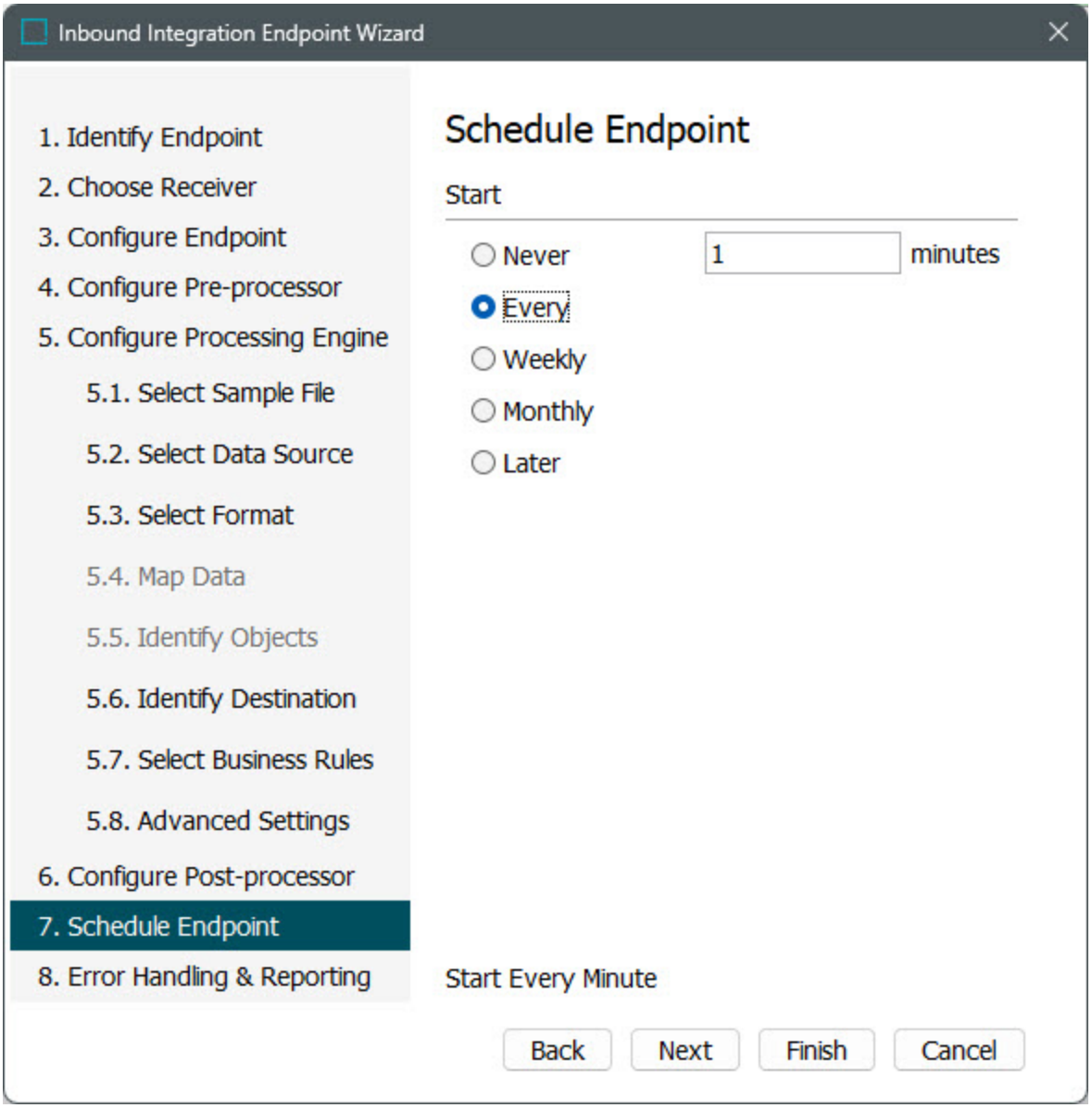
```
<?xml version="1.0" encoding="utf-8"?>
<!-- Configuration:
<STEP-ProductInformation ResolveInlineRefs="true"
FollowOverrideSubProducts="true">
<UnitList ExportSize="Selected"/>
<Entities ExportSize="Minimum">
<Entity>
<Name/><AttributeLink/><ClassificationCrossReference/><Entity/>

<ProductCrossReference/><AssetCrossReference/><EntityCrossReference/>
<ContextCrossReference/><Values/></Entity></Entities>
<Products ExportSize="Minimum">
<Product>

<Name/><AttributeLink/><DataContainerTypeLink/><ClassificationReferen
ce/><Product/>

<ProductCrossReference/><AssetCrossReference/><EntityCrossReference/>
<ClassificationCrossReference/><Values/><OverrideSubProduct/></Produc
t></Products>
</STEP-ProductInformation>

Export from Primary Product Hierarchy
Classifications All
Products All
Assets All

Exported by domain exporter.
-->
```

```
<STEP-ProductInformation ExportTime="2021-04-21 14:12:56"
ExportContext="Context1"
                         ContextID="Context1" WorkspaceID="Main"
UseContextLocale="false">

    <UnitList>
        <UnitFamily ID="0173-1#Z4-BAJ215#003" Selected="true"
Referenced="true">
            <Name>ECLASS electrical resistance</Name>
            <MetaData>
                <Value AttributeID="IS_Description">different
properties of materials which impede the electrical
                    current in its movement when the free charged
particles in these materials are set in motion by
                    electrical fields and/or electrical
potentials</Value>
                <Value AttributeID="IS_ImportVersion">13.0</Value>
            </MetaData>
            <Unit ID="0173-1#05-AAA049#003" Selected="true"
Referenced="true">
                <Name>EC Ohm</Name>
                <MetaData>
                    <Value AttributeID="UnitDescription">Ohm
description</Value>
                    <Value AttributeID="IS_Name">Ohm</Value>
                    <Value AttributeID="IS_
ImportVersion">ECLASS13.0</Value>
                </MetaData>
            </Unit>
            <Unit ID="0173-1#05-AAA460#003" Selected="true"
Referenced="true">
                <Name>EC microohm</Name>
                <MetaData>
```

```
                    <Value AttributeID="UnitDescription">microohm
description</Value>
                    <Value AttributeID="IS_Name">microohm</Value>
                    <Value AttributeID="IS_
ImportVersion">ECLASS13.0</Value>
                </MetaData>
            </Unit>
        </UnitFamily>
    </UnitList>
</STEP-ProductInformation>
```

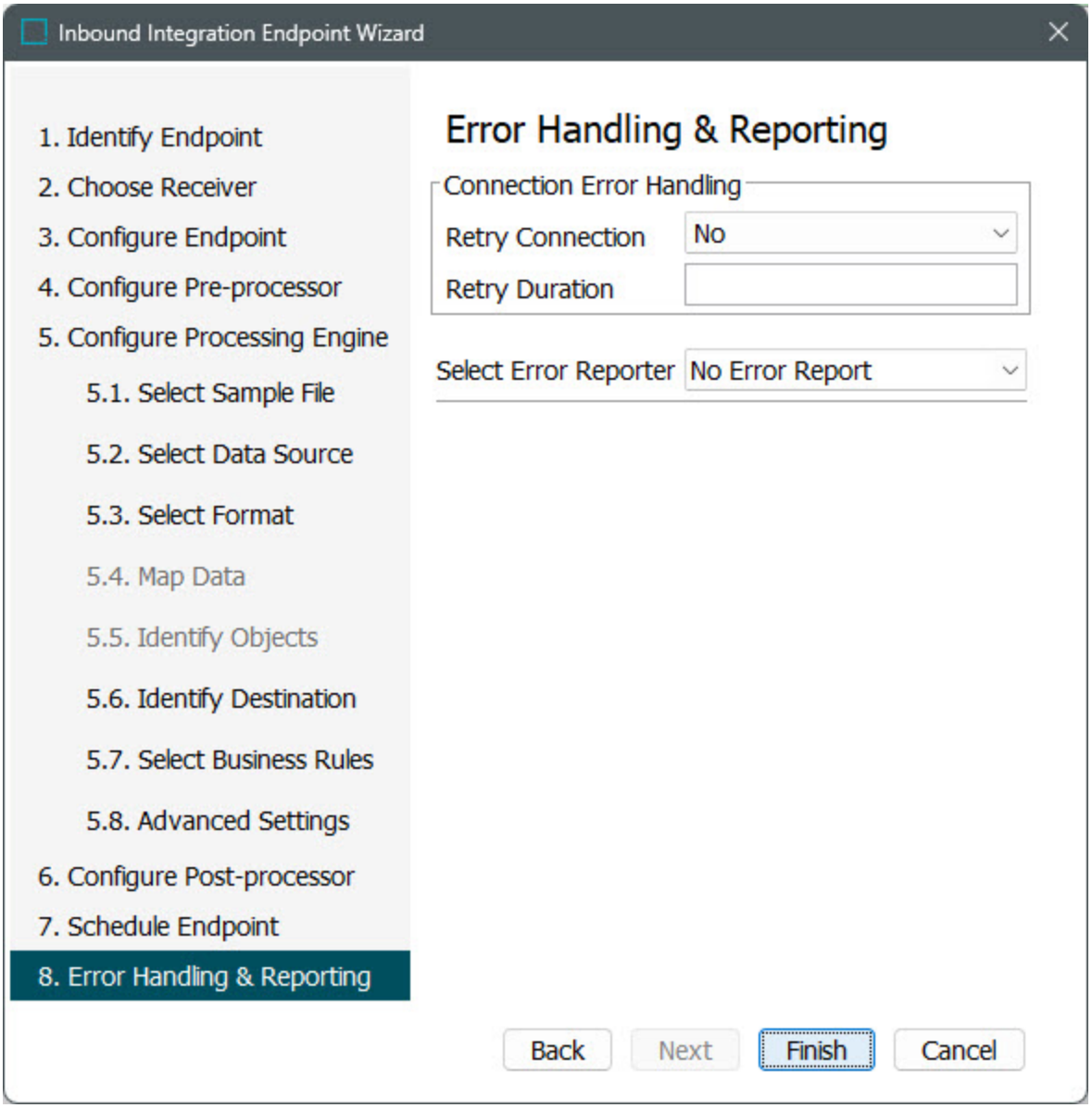
- Further potential sub-steps below the Configure Processing Engine are optional. For more information about these steps, refer to IIEP - Configure STEP Importer Processing Engine topic within the Data Exchange documentation.

7. Click the **Next** button to display the Schedule Endpoint parameters (bypassing the Configure Post-processor step). By default, 'Never' is selected. Optionally, update the values to those shown below.



For more information about the parameters available within this step, refer to IIEP - Schedule Endpoint topic within the Data Exchange documentation.

8. Click the **Next** button to display the Error Handling & Reporting step. The parameters are to be populated as recommended and shown below:



For more information about the parameters available within this step, refer to IIEP - Error Handling & Reporting topic within the Data Exchange documentation.

9. Click the **Finish** button, the Inbound Integration Endpoint Wizard will close, and the newly created endpoint will display within the workbench.

> ⚠ **Important:** An endpoint must be enabled before it can start processing data. For more information, refer to Running an Inbound Integration Endpoint topic within the Data Exchange documentation.

If users need to access the IIEP via a Web UI, the IIEP must be configured within a File Loading Widget. For more information, refer to the Configuring a File Loading Widget for ECLASS Advanced Unit Imports topic.

Configuring a File Loading Widget for ECLASS Advanced Unit Imports

When configured, Web UI users can import Unit files into STEP using a File Loading Widget.

Prerequisites

It is expected that anyone configuring the ECLASS Advanced Unit Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, refer to Designer Access topic within the Web User Interfaces documentation.

Before configuring the Web UI portion of this solution, an IIEP for an ECLASS Advanced Unit Importer must be configured within the workbench. For more information, refer to Configuring an IIEP for ECLASS Advanced Unit Imports topic.

Additionally, it is helpful to know how to add a widget to a Web UI Homepage. Details on how to do this can be found in Adding Widgets to a Homepage topic in the Web User Interfaces documentation.

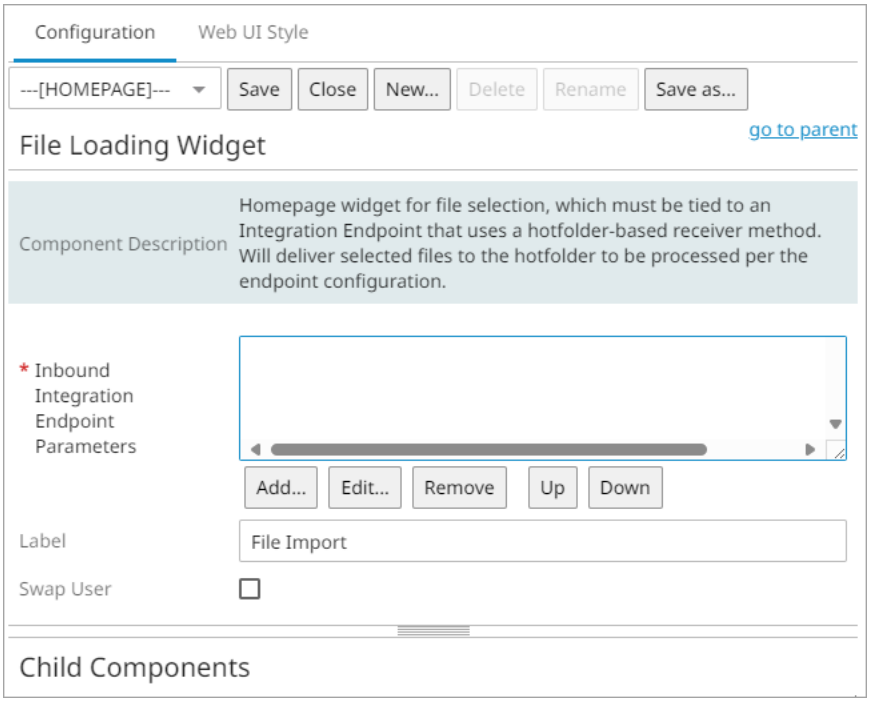
Configuration

Each screenshot example within this section provides recommended values for the parameters in ECLASS Advanced Unit Importer.

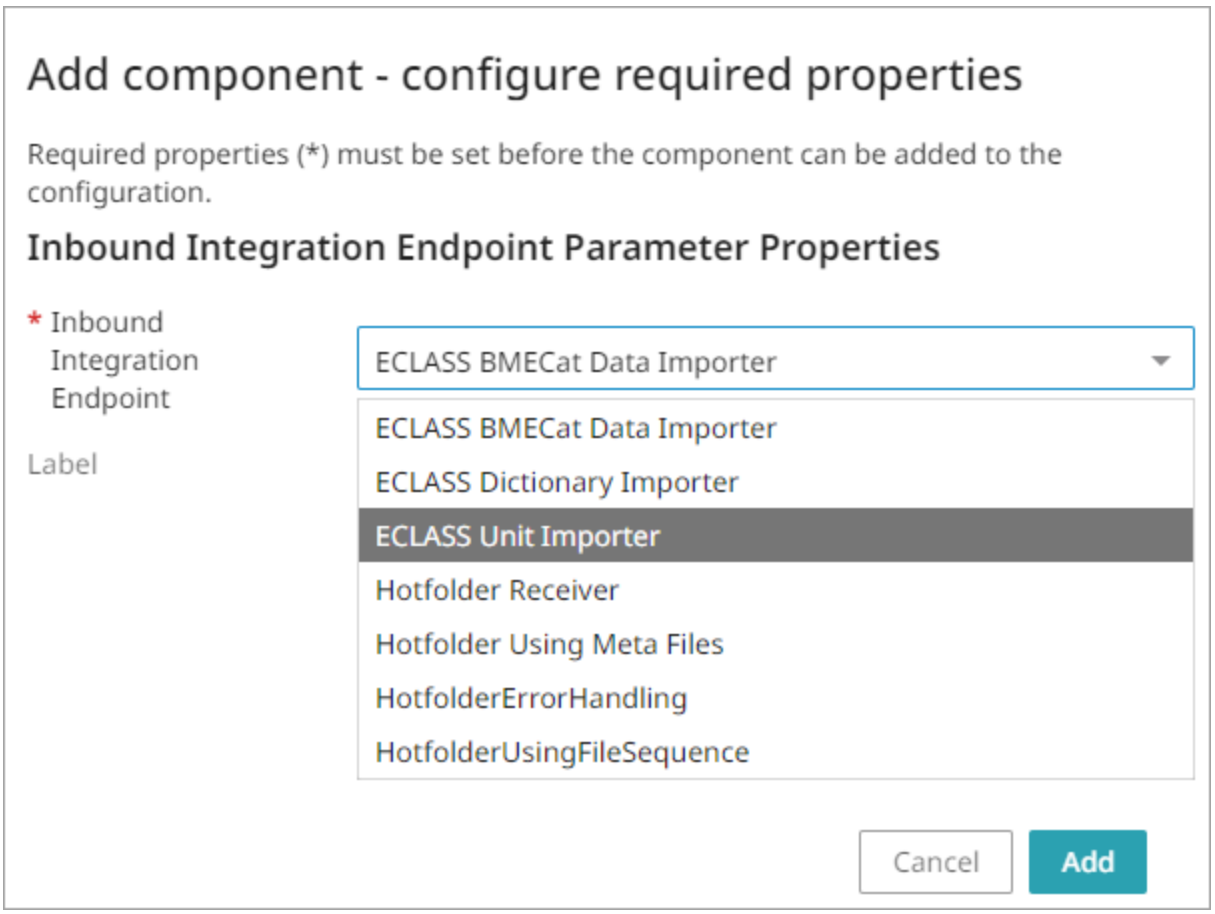
This topic describes how to configure a File Loading Widget so that users can drag and drop ECLASS Advanced Unit files onto a File Loading Widget on a Web UI Homepage.

1. In the designer, select an existing File Loading Widget to be used, or add a new File Loading Widget to the Homepage Widget Grid component. For more information, refer to the File Loading Widget topic within the Web User Interfaces documentation.

2. Go to the Inbound Integration Endpoint Parameters field, click the **Add** button, and the Inbound Integration Endpoint Parameter Properties dialog will display.



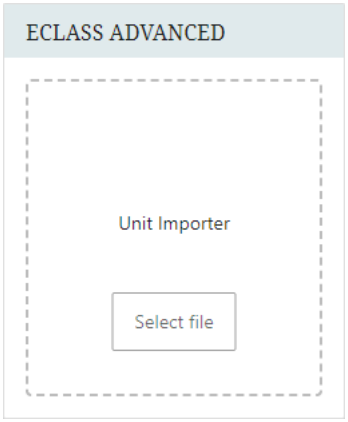
3. Click the dropdown for the Inbound Integration Endpoint parameter, and select **ECLASS Unit Importer** (the IIEP created for ECLASS Advanced Unit imports).



Note: If the desired IIEP does not display in the dropdown, it can be created using the steps described in Configuring an IIEP for ECLASS Advanced Unit Imports topic.

4. Optionally, provide a label to be displayed within the drop zone of the widget.

   In the example below, a File Loading Widget labeled as 'Unit Importer' is displayed.



5. Click the **Save** and **Close** buttons to save the changes and close the designer.

Using ECLASS Advanced Unit Importer

An ECLASS Advanced Unit file can be imported into STEP by uploading it to either a configured hotfolder or through a File Loading Widget on a Web UI Homepage. The intention of the Unit Importer Web UI setup included within this topic is to provide an out-of-the-box solution for importing data included within a supported ECLASS Advanced Unit format.

Prerequisites

If you have completed the Easy Setup actions for the ECLASS Advanced Model, the functionalities outlined in this topic should be readily available for use. Otherwise, configuration is required. For information on how to configure ECLASS Advanced Unit importer, refer to Configuring ECLASS Advanced Unit Importer topic.

The Unit Importer exclusively supports XML file formats. In case the XML file is zipped, it is essential to manually extract the .zip file to access the Unit file.

Import Process Overview

Once a valid ECLASS Advanced file is uploaded using a File Loading Widget (or uploaded directly to a hotfolder), the file is picked up by an IIEP, and the IIEP starts a background process.

Procedure

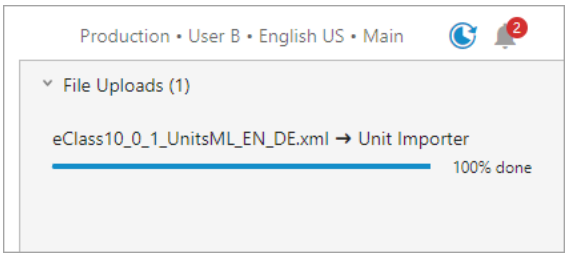
1. Access the ECLASS Advanced Web UI Homepage.
2. Upload a valid Unit file to the hotfolder (root/upload/hotfolders/EclassAdvUnitImporter/In), or use the 'Unit Import' File Loading Widget.

   For information about File Loading Widget, refer to File Loading Widget topic within the Web User Interfaces documentation.

   For more information about uploading files directly to the hotfolder, refer to ECLASS Advanced Import Framework topic.

3. Once the upload has started, users can view the progress of the upload using the 'Recent background processes' side panel.

   In the example below, the 'Recent background processes' side panel is expanded, and the file 'eClass10_0_1_UnitsML_EN_DE.xml' is 100% processed.



   For more information on using the 'Recent background processes' side panel, refer to Recent Background Processes Side Panel topic within the Web User Interfaces documentation.

After uploading the file, the IIEP picks it up and initiates a background process for the import. The file loading widget does not provide background process monitoring in the Web UI. You can monitor the import status within the workbench through the IIEP Background Process that is generated.

For more information about monitoring the IIEP via background process, refer to Monitoring an IIEP via Background Process topic within the Data Exchange documentation.

ECLASS Dictionary Importer

The main goal of the ECLASS Advanced Dictionary Importer is to provide a seamless ready-to-use solution for importing Dictionary data in a supported XML format.

The Dictionary file is typically bundled within the ECLASS-provided .zip file. To access the Dictionary file, you will need to manually unzip the archive. The Dictionary Importer exclusively accepts XML file formats.

This section includes information on:

- Using ECLASS Advanced Dictionary Importer
- Configuring ECLASS Advanced Dictionary Importer

Configuring ECLASS Advanced Dictionary Importer

> **Note:** If the Easy Setup actions for the ECLASS Advanced Component model have been completed, then the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

The following topics provide the configuration steps necessary to allow users to be able to drag and drop Dictionary files onto a configured File Loading Widget, and monitor the progress of the import file in the created IIEP Background Process in the workbench.

- Configuring an IIEP for ECLASS Advanced Dictionary Imports
- Configuring a File Loading Widget for ECLASS Advanced Dictionary Imports

Configuring an IIEP for ECLASS Advanced Dictionary Imports

> **Note:** If the Easy Setup actions for the ECLASS Advanced Component model have been completed, then the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

An inbound integration endpoint (IIEP) can be configured in the workbench to help the process of importing Dictionary files into STEP. Once an IIEP is configured for ECLASS Advanced Dictionary imports, Dictionary files can be imported after they are uploaded either to a configured hotfolder, or to a File Loading Widget on a Web UI Homepage. For more information, refer to ECLASS Advanced Dictionary Importer topic.

This section describes how to configure an IIEP that can allow for the automated processing of Dictionary files. Each screenshot example within this section provides recommended values for the parameters in ECLASS Advanced Dictionary Importer.

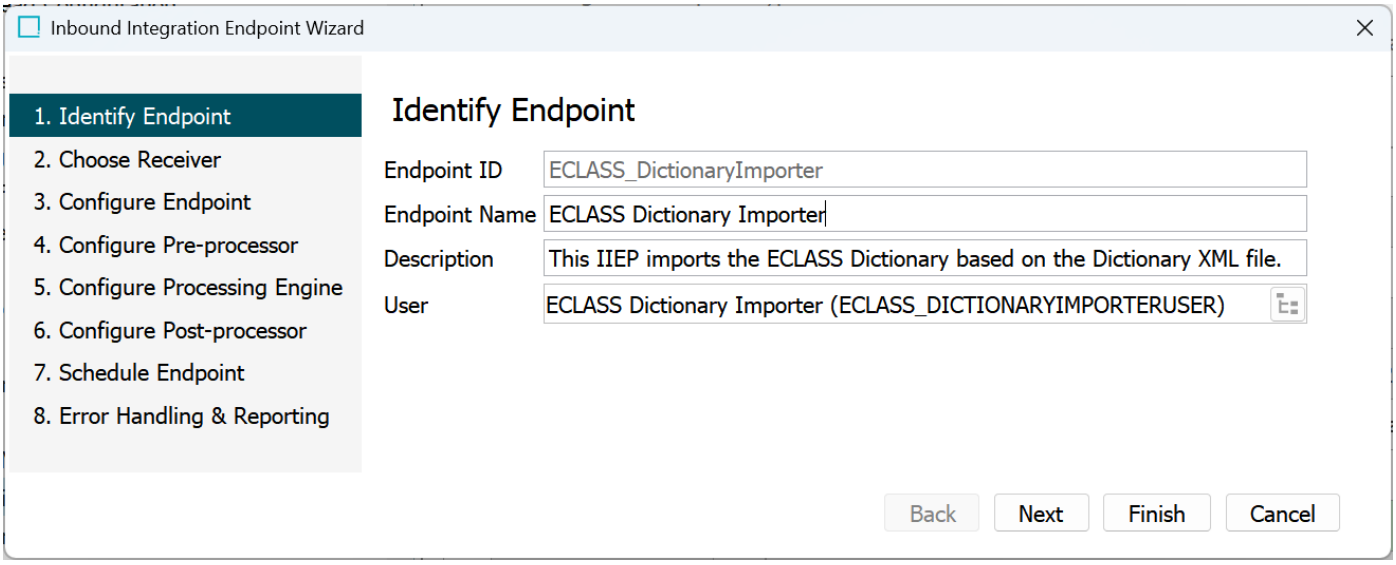
Prerequisites

This topic aims to acquaint users with the IIEP specifically designated for the import of Dictionary files. It does not cover general IIEP functionalities. It is assumed that individuals configuring an IIEP for ECLASS Advanced Dictionary Import are well-versed in configuring and processing standard inbound integration endpoints. For a comprehensive understanding of the standard functionalities provided in inbound integration endpoints, refer to Inbound Integration Endpoints topic within the Data Exchange documentation.

Configuration Steps

1. In the workbench, go to System Setup, select and right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.

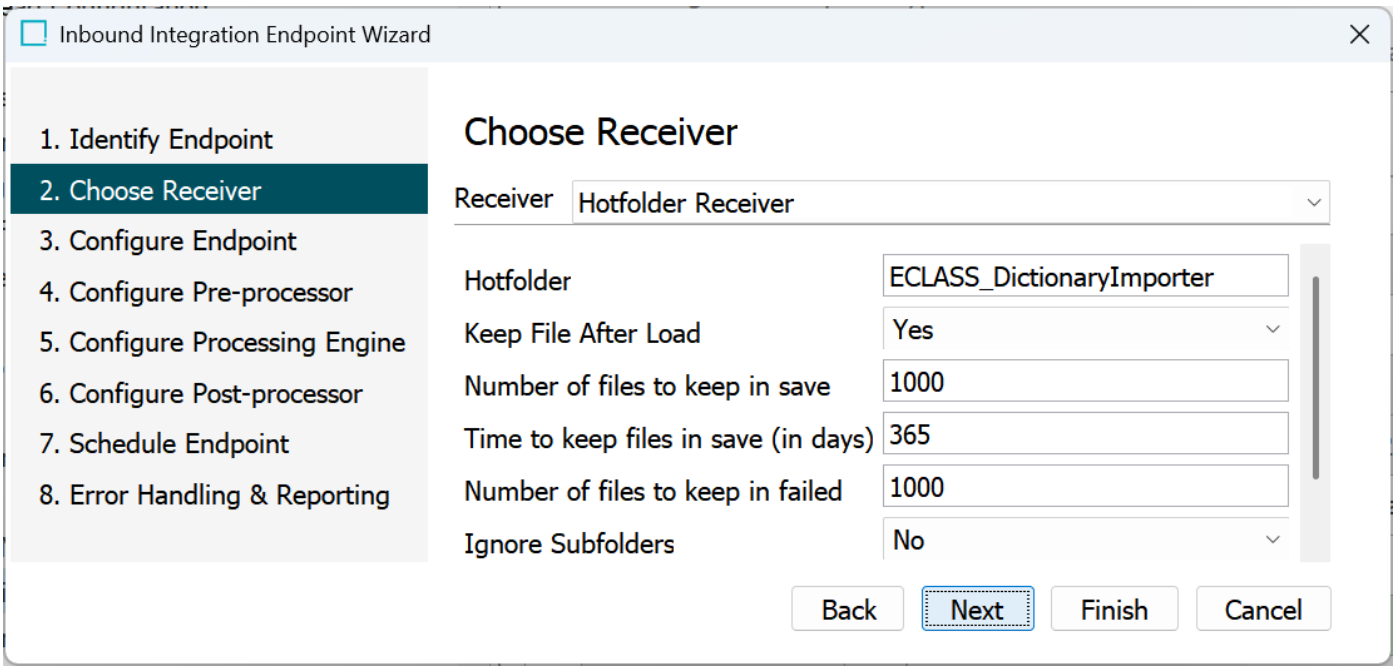
2. Once the Inbound Integration Endpoint Wizard displays, The parameters are to be populated as recommended and shown below.



For more information about the parameters available within the Identify Endpoint step, refer to IIEP - Identify Endpoint topic within the Data Exchange documentation.

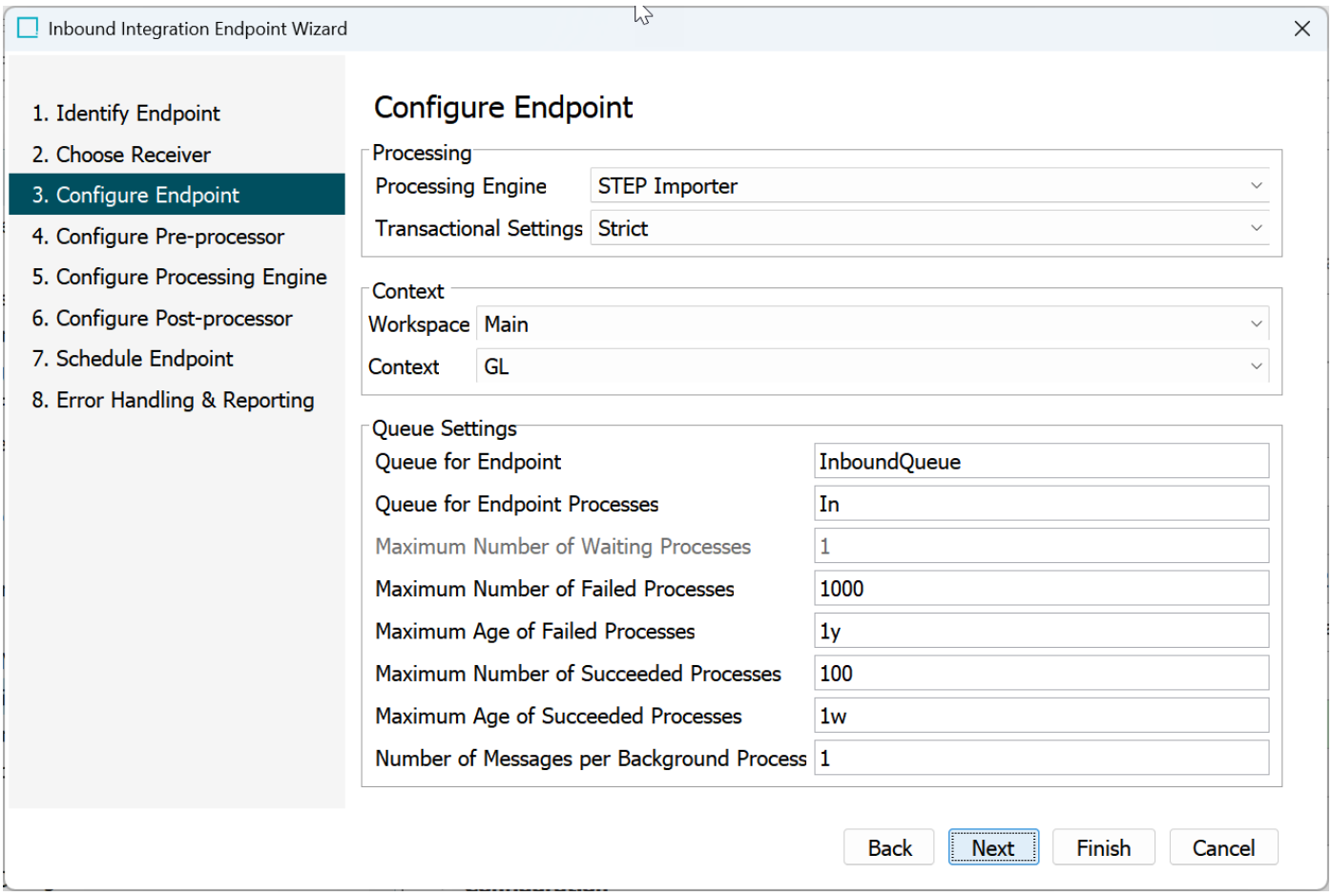
3. Click the **Next** button, and the Choose Receiver parameters will display. The parameters are to be populated as recommended and shown below. The mandatory parameter Hotfolder must be populated with a hotfolder name before the Next button will enable. In the screenshot below, the Hotfolder parameter is populated with the value 'EClassAdvDictionaryImporter.'

> **Note:** The value within this hotfolder parameter will be used to create the new hotfolder, once the IIEP Wizard is complete.



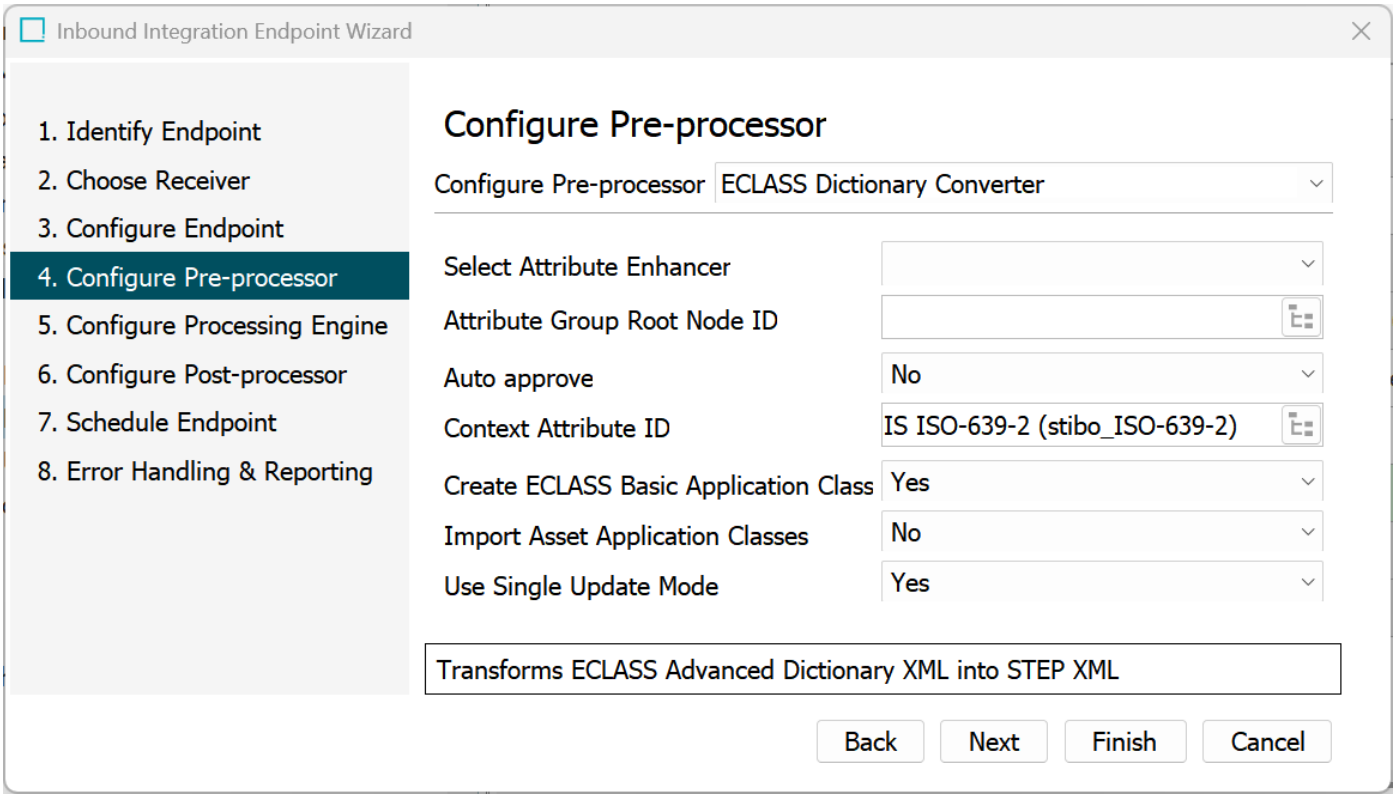
For more information about the parameters, refer to IIEP - Choose Receiver topic within the Data Exchange documentation.

4. Click the **Next** button, and the Configure Endpoint parameters will display. The parameters are pre-populated with the recommended values as shown below.



For more information about the parameters, refer to IIEP - Configure Endpoint topic within the Data Exchange documentation.

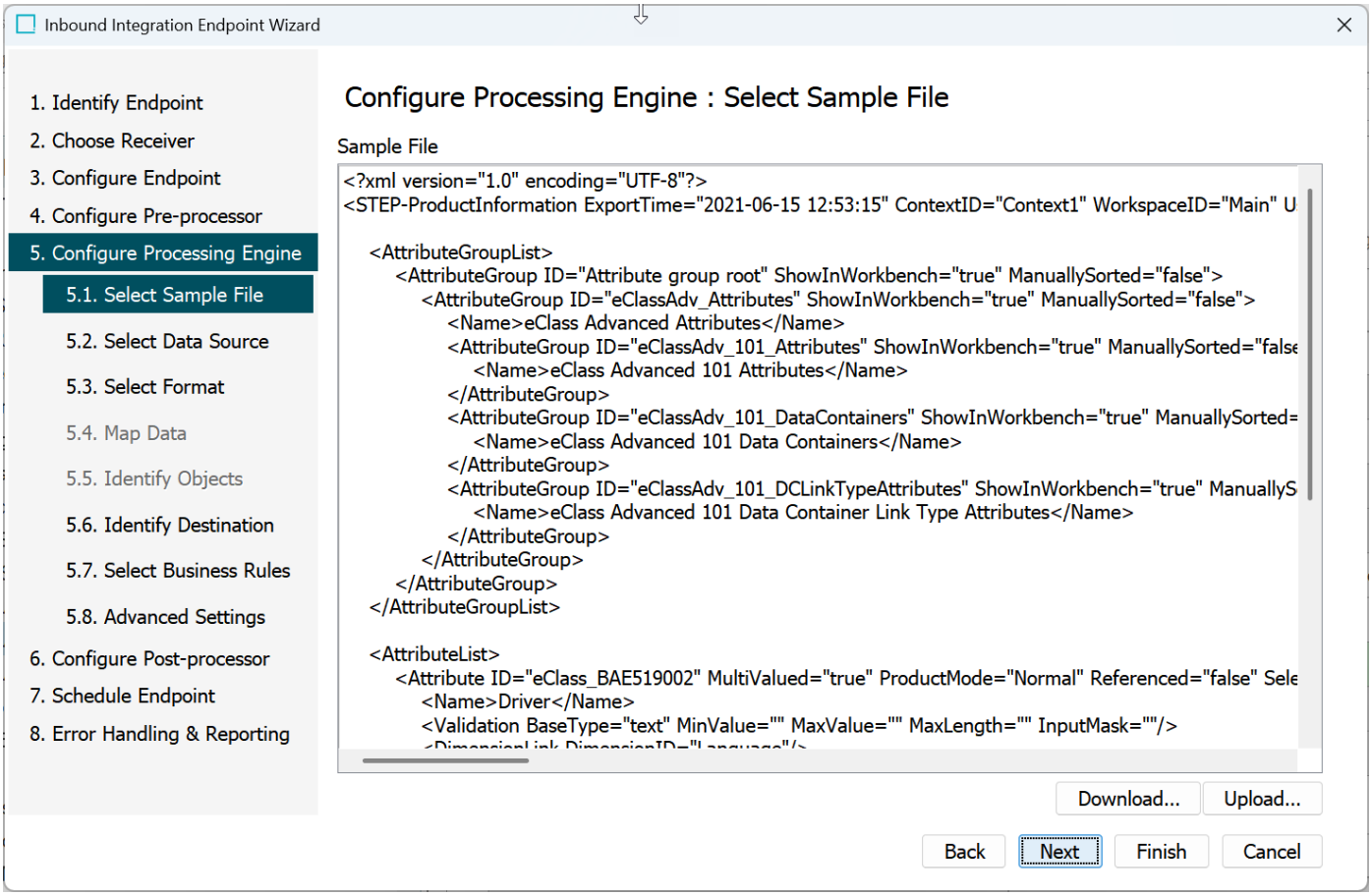
5. Click the **Next** button, and the Configure Pre-processor parameter will display. The selection of the pre-processor within this step makes the IIEP unique for importing Dictionary files. The parameters are to be populated as recommended and shown below:



- **Configure Pre-processor:** This parameter has to be populated with ECLASS Dictionary Converter option. This is an exclusive pre-processor for importing Dictionary files. For more information about the parameter, refer to IIEP - Configure Pre-processor topic within the Data Exchange documentation.
- **Select Attribute Enhancer:** This parameter is typically used to apply additional logic or enrichment to attributes during import. It allows you to choose an attribute enhancer from a predefined list. The enhancer applies rules to modify, validate, or enrich attribute values during import.
- **Attribute Group Root Node ID:** Upon defining a value (root node) within this parameter, it serves as a directive for determining the specific location for attribute import. If left empty, the attributes will be imported beneath the standard top Attribute Group Root node.
- **Auto approve:**This parameter determines whether the imported classification nodes are to be automatically approved or not. The default setting is No.
- **Context Attribute ID:** This parameter is to be populated with the attribute IS ISO-639-2 (ID = stibo_ISO-639-2). This attribute is created by the Easy Setup action and holds the language mappings. For more information about language mappings, refer to Prepare the Language Dimension Mapping topic.

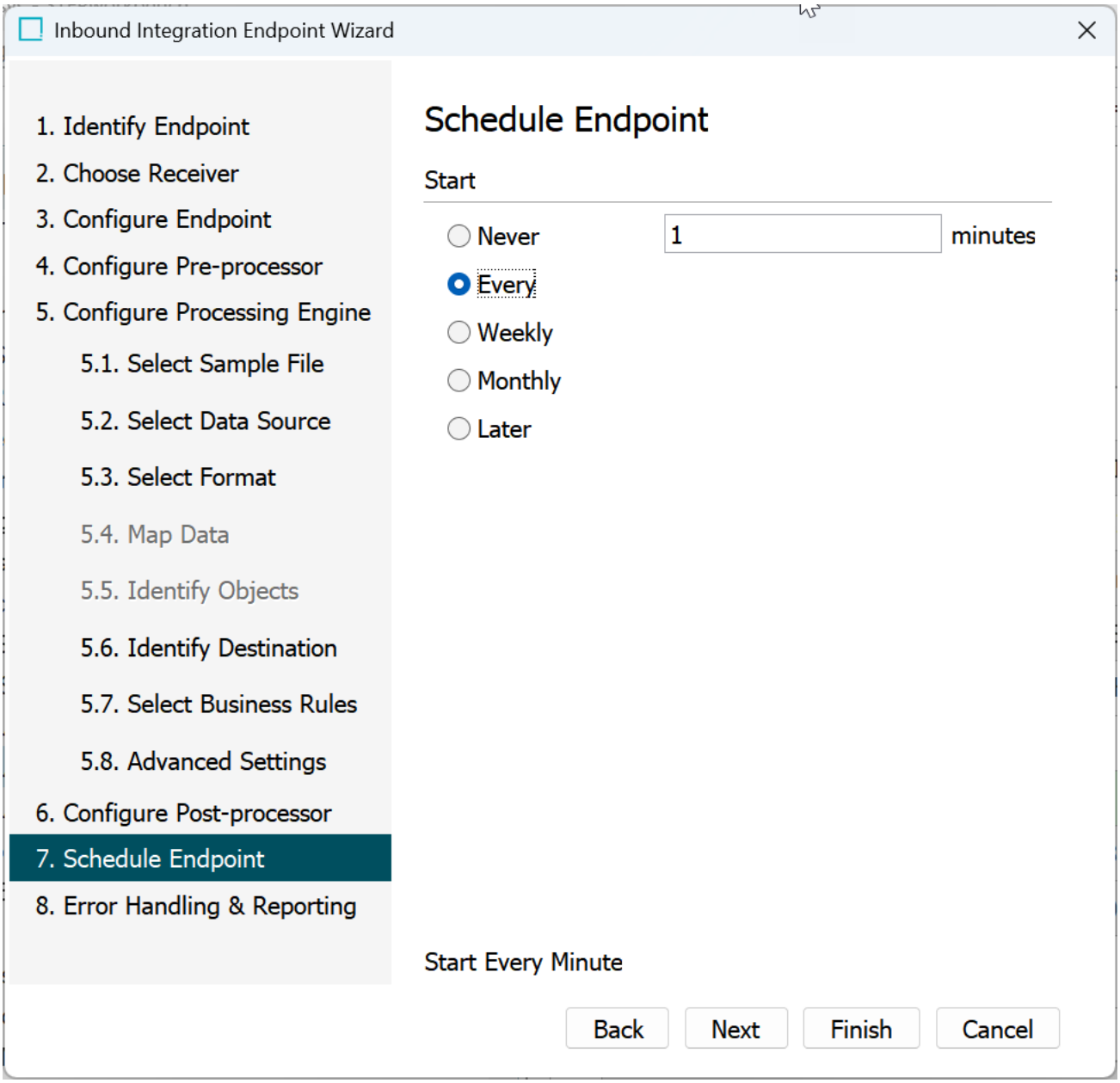
- **Enable ECLASS Basic:** The Enable ECLASS Basic parameter is used to determine whether the system should operate with the ECLASS Basic standard instead of ECLASS Advanced during import processes. When enabled, the system applies the ECLASS Basic logic for product classification, attributes, and mapping instead of the ECLASS Advanced model.
- **Import Asset Application Classes:** This parameter controls whether Asset Application Classes are imported into STEP during an import process. Enable this parameter if your implementation requires asset-related classification structures.
- **Use Single Update Mode:**This parameter controls whether the import process runs in single-update mode, which is a special lock state in STEP. When enabled, the system enters single-update mode during import. This means Only the import process can modify data in the database while the mode is active. All other users can still view data but cannot make changes until the import completes.

6. Click the **Next** button, and the 'Configure Processing Engine: Select Sample File' field for the STEP Importer processing engine will display.



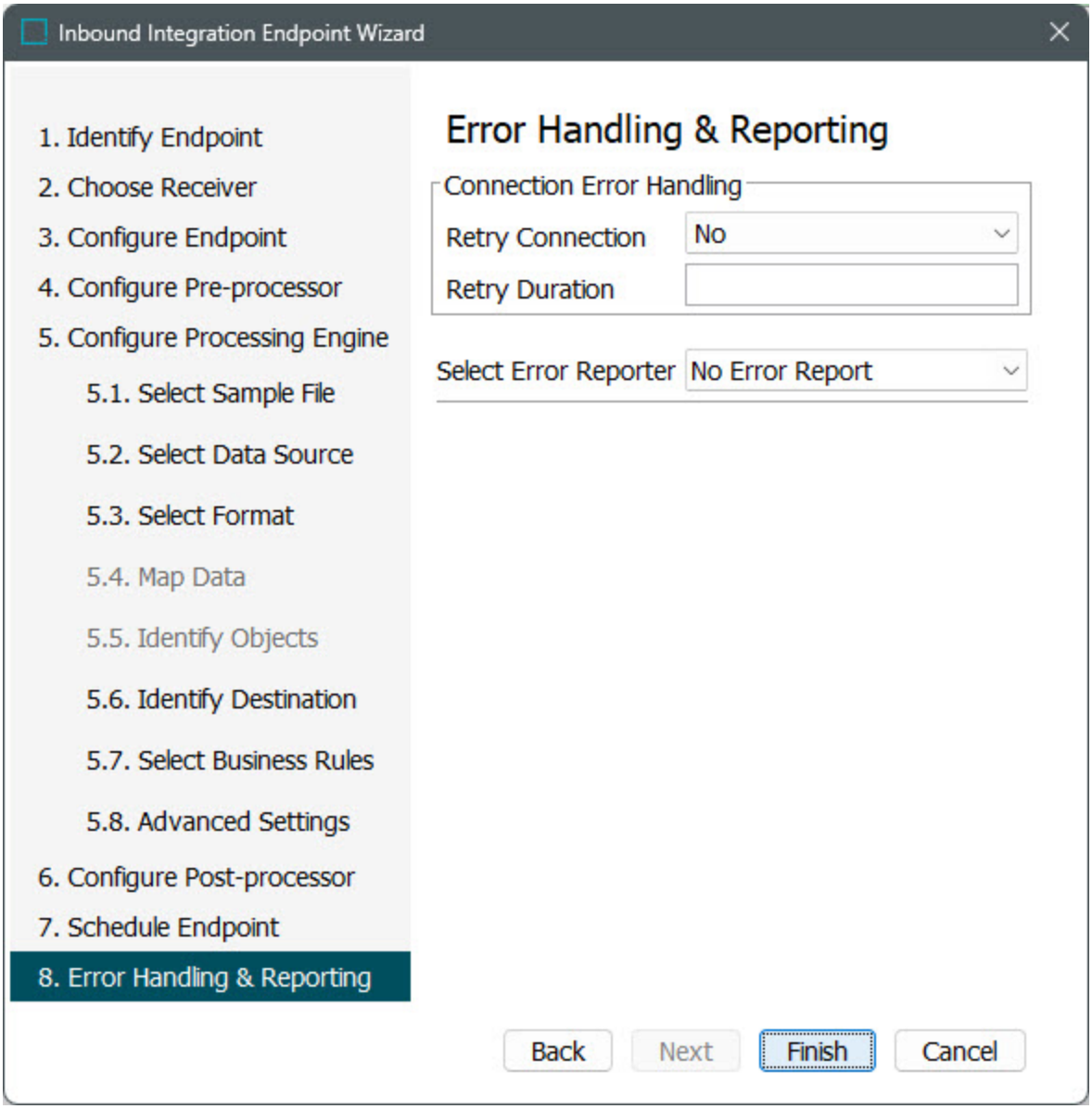
- Further potential sub-steps underneath the Configure Processing Engine are optional. For more information about these steps, refer to IIEP - Configure STEP Importer Processing Engine topic within the Data Exchange documentation.

7. Click the **Next** button to display the Schedule Endpoint parameters (bypass the Configure Post-processor parameters). Update the values to those shown below.



For information about the parameters available within this step, refer to IIEP - Schedule Endpoint topic within the Data Exchange documentation.

8. Click the **Next** button, and the Error Handling & Reporting step will display. By default, the parameters are populated as recommended and shown below.



For information about the parameters available within this step, refer to IIEP - Error Handling & Reporting topic within the Data Exchange documentation.

9. Click the **Finish** button, the Inbound Integration Endpoint Wizard will close, and the newly created endpoint will display within the workbench.

> **Important:** An endpoint must be enabled before it can start processing data. For more information, refer to Running an Inbound Integration Endpoint topic within the Data Exchange documentation.

If users need to access the IIEP via a Web UI, then the IIEP must be configured within a File Loading Widget. For more information, refer to the Configuring a File Loading Widget for ECLASS Advanced Dictionary Imports topic.

Configuring a File Loading Widget for ECLASS Advanced Dictionary Imports

When configured, Web UI users can import Dictionary files into STEP using a File Loading Widget.

Prerequisites

It is expected that anyone configuring the ECLASS Advanced Dictionary Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, refer to Designer Access topic within the Web User Interfaces documentation.

Before configuring the Web UI portion of this solution, an IIEP for an ECLASS Advanced Dictionary Importer must be configured within the workbench. For more information, refer to Configuring an IIEP for ECLASS Advanced Dictionary Imports topic.

Additionally, it is helpful to know how to add a widget to a Web UI Homepage. Details on how to do this can be found in Adding Widgets to a Homepage topic in the Getting Started documentation.

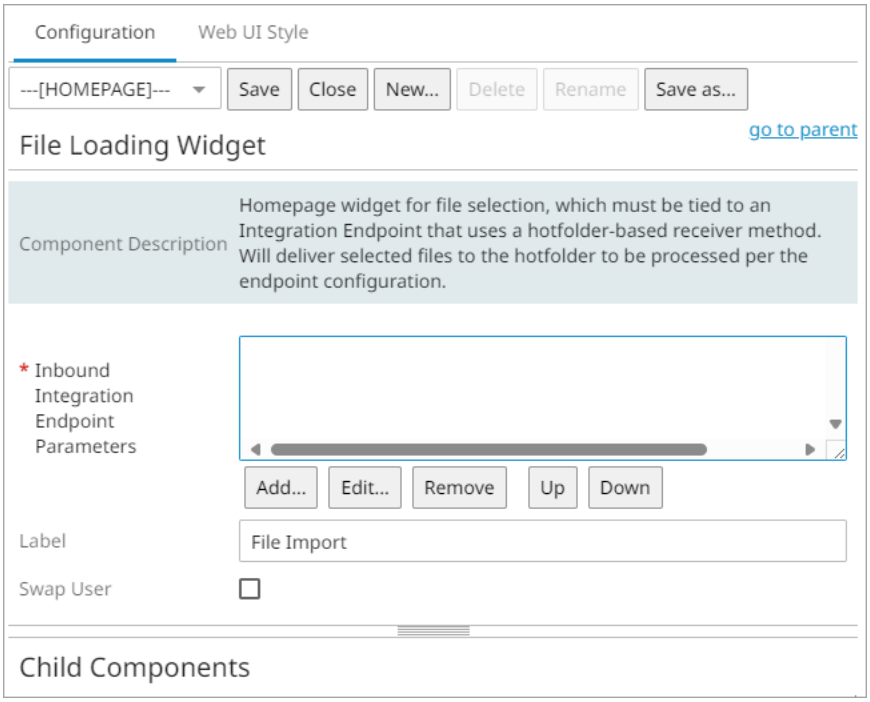
Configuration

Each screenshot example within this section provides recommended values for the parameters in ECLASS Advanced Dictionary Importer.

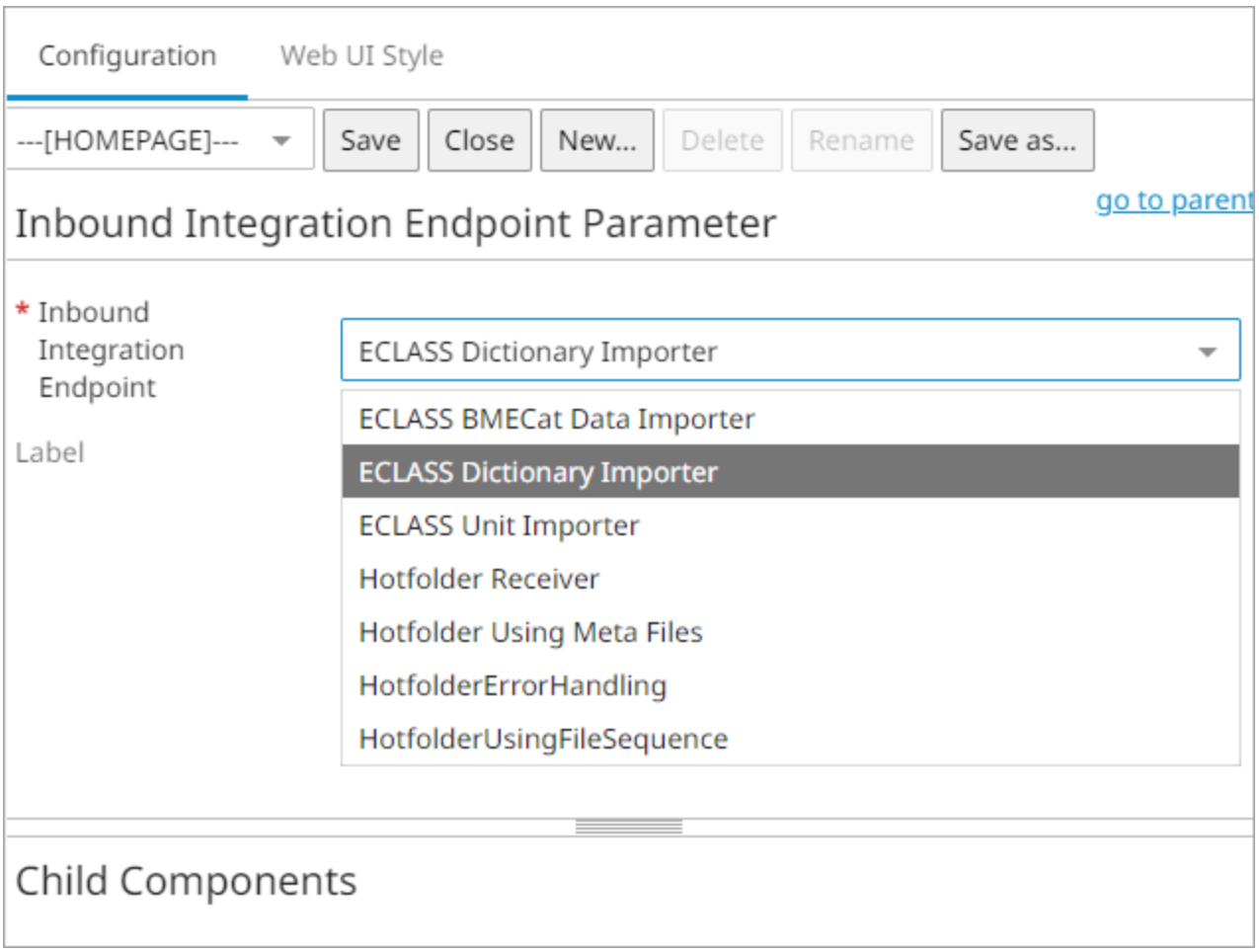
This topic describes how to configure a File Loading Widget so that users can drag and drop ECLASS Advanced Dictionary files onto a File Loading Widget on a Web UI Homepage.

1. In the designer, select an existing File Loading Widget to be used, or add a new File Loading Widget to the Homepage Widget Grid component. For more information, refer to the File Loading Widget topic within the Web User Interfaces documentation.

2. Go to the Inbound Integration Endpoint Parameters field, click the **Add** button, and the Inbound Integration Endpoint Parameter Properties dialog will display.



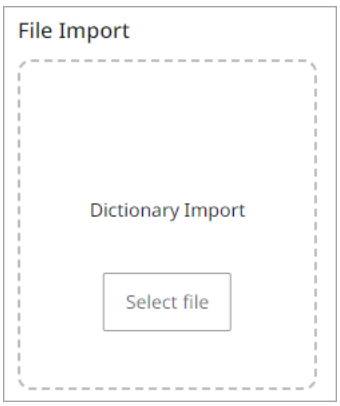
3. Click the dropdown for the Inbound Integration Endpoint parameter and select **ECLASS ADVANCED Dictionary Importer** (the IIEP created for ECLASS Advanced Dictionary imports).



Note: If the desired IIEP does not display in the dropdown, then it can be created using the steps described in the Configuring an IIEP for ECLASS Advanced Dictionary Imports topic.

4. Optionally, provide a label to be displayed within the drop zone of the widget.

   In the example below, a File Loading Widget labeled as 'ECLASS ADVANCED' is displayed above its configurations.



5. Click the **Save** and **Close** buttons to save the changes and close the designer.

Using ECLASS Advanced Dictionary Importer

Dictionary files can be imported into STEP by uploading it to either a configured hotfolder, or through a File Loading Widget on a Web UI Homepage. The intention of the Dictionary Importer Web UI setup included within this topic is to provide an out-of-the-box solution for importing data included within a supported ECLASS Advanced Dictionary format.

Prerequisites

If you have completed the Easy Setup actions for the ECLASS Advanced Model, the functionalities outlined in this topic should be readily available for use. Otherwise, configuration is required. For information on how to configure ECLASS Advanced Dictionary importer, refer to Configuring ECLASS Advanced Dictionary Importer topic.

The Dictionary Importer exclusively supports XML file formats. In case the XML file is zipped, it is essential to manually extract the .zip file to access the Dictionary file.

Import Process Overview

Once a valid ECLASS Advanced file is uploaded using a File Loading Widget (or uploaded directly to a hotfolder), the file is picked up by an IIEP, and the IIEP starts a background process.

Note: The File Loading Widget permits a maximum file size of 20 MB for uploading purposes. Should the files intended for import exceed this 20 MB threshold, we recommend a direct upload to the designated hotfolder. Additionally, if the file upload process exceeds 30 seconds, the web server may time out. In such cases, it is advisable to use the direct hotfolder upload method. This functionality is managed by a configuration property that is not available in the Self-Service UI. Contact Stibo Systems Support for assistance. The following config properties setting enables uploading files larger than 20 MB by adjusting the value as HotfolderUpload.MaxFileSize=4096.

Procedure

1. Access the ECLASS Advanced Web UI Homepage.
2. Upload a valid Dictionary file to the hotfolder (root/upload/hotfolders/ECLASS_DictionaryImporter/In), or use the 'Dictionary Import' File Loading Widget.

   For information about File Loading Widget, refer to File Loading Widget topic within the Web User Interfaces documentation.

   For more information about uploading files directly to the hotfolder, refer to ECLASS Advanced Import Framework topic.
3. Once the upload has started, users can view the progress of the upload using the 'Recent background processes' side panel.

   For more information on using the 'Recent background processes' side panel, refer to Recent Background Processes Side Panel topic within the Web User Interfaces documentation.

After uploading the file, the IIEP picks it up and initiates a Background Process for the import process. The file loading widget doesn't provide background process monitoring in the Web UI. You can monitor the import status within the workbench through the IIEP Background Process that is generated.

For more information about monitoring the IIEP via background process, refer to Monitoring an IIEP via Background Process topic within the Data Exchange documentation

Using ECLASS Advanced Dictionary Importer

Dictionary files can be imported into STEP by uploading it to either a configured hotfolder, or through a File Loading Widget on a Web UI Homepage. The intention of the Dictionary Importer Web UI setup included within this topic is to provide an out-of-the-box solution for importing data included within a supported ECLASS Advanced Dictionary format.

Prerequisites

If you have completed the Easy Setup actions for the ECLASS Advanced Model, the functionalities outlined in this topic should be readily available for use. Otherwise, configuration is required. For information on how to configure ECLASS Advanced Dictionary importer, refer to Configuring ECLASS Advanced Dictionary Importer topic.

The Dictionary Importer exclusively supports XML file formats. In case the XML file is zipped, it is essential to manually extract the .zip file to access the Dictionary file.

Import Process Overview

Once a valid ECLASS Advanced file is uploaded using a File Loading Widget (or uploaded directly to a hotfolder), the file is picked up by an IIEP, and the IIEP starts a background process.

Note: The File Loading Widget permits a maximum file size of 20 MB for uploading purposes. Should the files intended for import exceed this 20 MB threshold, we recommend a direct upload to the designated hotfolder. Additionally, if the file upload process exceeds 30 seconds, the web server may time out. In such cases, it is advisable to use the direct hotfolder upload method. This functionality is managed by a configuration property that is not available in the Self-Service UI. Contact Stibo Systems Support for assistance. The following config properties setting enables uploading files larger than 20 MB by adjusting the value as HotfolderUpload.MaxFileSize=4096.

Procedure

1. Access the ECLASS Advanced Web UI Homepage.
2. Upload a valid Dictionary file to the hotfolder (root/upload/hotfolders/ECLASS_DictionaryImporter/In), or use the 'Dictionary Import' File Loading Widget.

   For information about File Loading Widget, refer to File Loading Widget topic within the Web User Interfaces documentation.

   For more information about uploading files directly to the hotfolder, refer to ECLASS Advanced Import Framework topic.
3. Once the upload has started, users can view the progress of the upload using the 'Recent background processes' side panel.

   For more information on using the 'Recent background processes' side panel, refer to Recent Background Processes Side Panel topic within the Web User Interfaces documentation.

After uploading the file, the IIEP picks it up and initiates a Background Process for the import process. The file loading widget doesn't provide background process monitoring in the Web UI. You can monitor the import status within the workbench through the IIEP Background Process that is generated.

For more information about monitoring the IIEP via background process, refer to Monitoring an IIEP via Background Process topic within the Data Exchange documentation

ECLASS BMECat Data Importer

The primary objective of the ECLASS Advanced Data Importer is to offer a convenient out-of-the-box solution for importing BMEcat 2005.1 file in a supported XML format. To ensure a successful upload, it is essential to verify that the version of the Data file you intend to upload is listed among the Supported Versions and Formats. You can confirm the file version being uploaded is listed within ECLASS Standard Supported Versions and Formats topic.

The Data Importer exclusively accepts XML file formats.

This section includes information on:

- Using ECLASS BMECat Data Importer
- Configuring ECLASS BMECat Data Importer

Configuring ECLASS BMECat Data Importer

> **Note:** If the Easy Setup actions for the ECLASS Advanced Component model have been completed, then the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

The following topics provide the configuration steps necessary to allow users to be able to drag and drop Data files onto a configured File Loading Widget and monitor the progress of the import file in the created IIEP Background Process in the workbench.

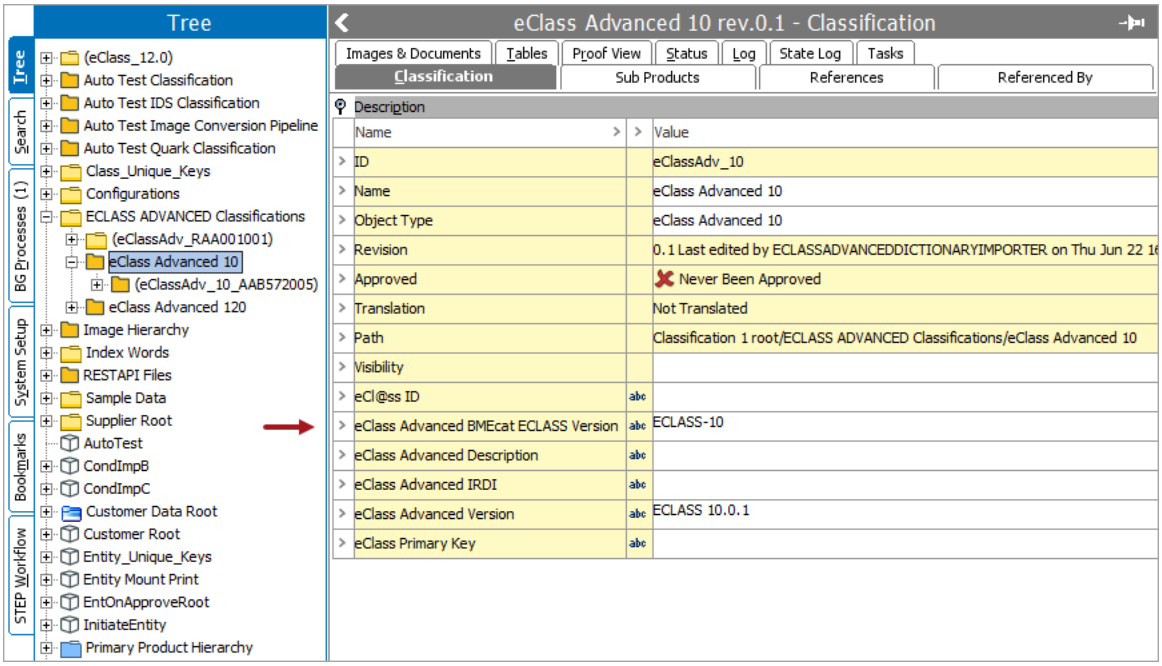
- Configuring an IIEP for ECLASS BMECat Data Imports
- Configuring a File Loading Widget for ECLASS BMECat Data Imports

Mapping BMEcat 2005.1 Reference Feature System with the Corresponding ECLASS Advanced Version

To successfully import any BMEcat 2005.1 file into STEP, it's essential to establish a mapping between the BMEcat 2005.1 reference feature system and the corresponding ECLASS ADVANCED version. This mapping is necessary due to the lack of alignment between the naming conventions used in BMEcat 2005.1 and the Dictionary file, specifically when indicating versions.

For example, while BMEcat 2005.1 employs the term '10.0' as its reference feature system, the ECLASS ADVANCED Dictionary refers to the version / revision as '10.0.1'. In situations like this, it becomes imperative for STEP to have a mechanism that indicates how the BMEcat data should be associated with the appropriate ECLASS ADVANCED version.

This mapping process must be applied to the metadata attribute 'eClass Advanced BMEcat ECLASS Version,' which is located on the version dependent object named 'eClassAdv_[Version]_Root.'



For the attribute, the value should be retrieved from the value available within the following BMEcat 2005.1 file tag:

```
"<REFERENCE_FEATURE_SYSTEM_NAME>xxx</REFERENCE_FEATURE_SYSTEM_NAME>"
```

For example, within the ECLASS Advanced 10.0 file, the value will be displayed as:

```
"<REFERENCE_FEATURE_SYSTEM_NAME>ECLASS-10.0</REFERENCE_FEATURE_SYSTEM_
NAME>"
```

Note: If the value is not applied, no data import takes place and the import execution report will show: "*Failed to identify a valid version number. Ensure the version is registered in the attribute 'eClassAdv_BMEcatECLASSVersion …….. The product will not be imported*"

Configuring an IIEP for ECLASS BMECat Data Imports

> **Note:** If the Easy Setup actions for the ECLASS Advanced Component model have been completed, then the configurations explained within this topic have been set up automatically. The purpose of this topic is to detail those settings to assist admins in adjusting their solution where necessary.

An inbound integration endpoint (IIEP) can be configured in the workbench to help the process of importing Data (BMEcat 2005.1) files into STEP. Once an IIEP is configured for ECLASS Advanced Data imports, Data files can be imported after they are uploaded either to a configured hotfolder, or to a File Loading Widget on a Web UI Homepage. For more information, refer to ECLASS BMECat Data Importer topic.

This section describes how to configure an IIEP that can allow for the automated processing of Data files. Each screenshot example within this section provides recommended values for the parameters in ECLASS Advanced Data Importer.

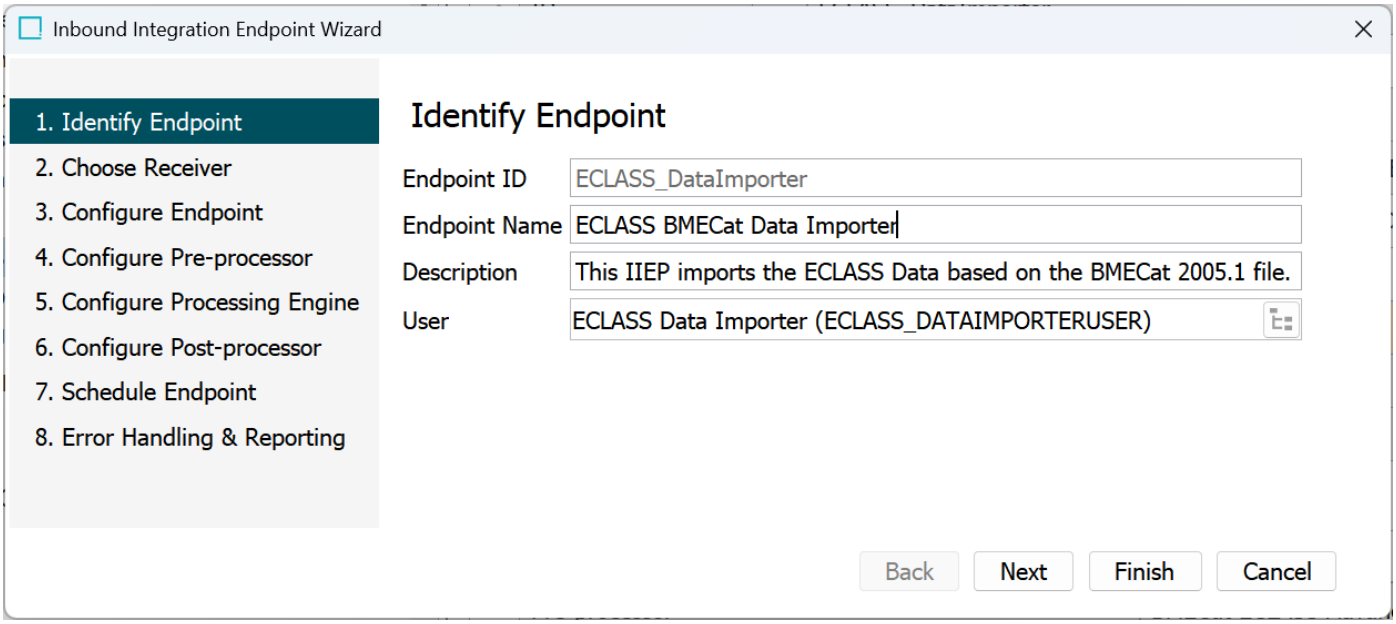
Prerequisites

This topic aims to acquaint users with the IIEP specifically designated for the import of BMEcat 2005.1 data files. It does not cover general IIEP functionalities. It is assumed that individuals configuring an IIEP for ECLASS Advanced Data Import are well-versed in configuring and processing standard inbound integration endpoints. For a comprehensive understanding of the standard functionalities provided in inbound integration endpoints, refer to Inbound Integration Endpoints topic within the Data Exchange documentation.

Configuration Steps

1. In the workbench, go to System Setup, select and right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.

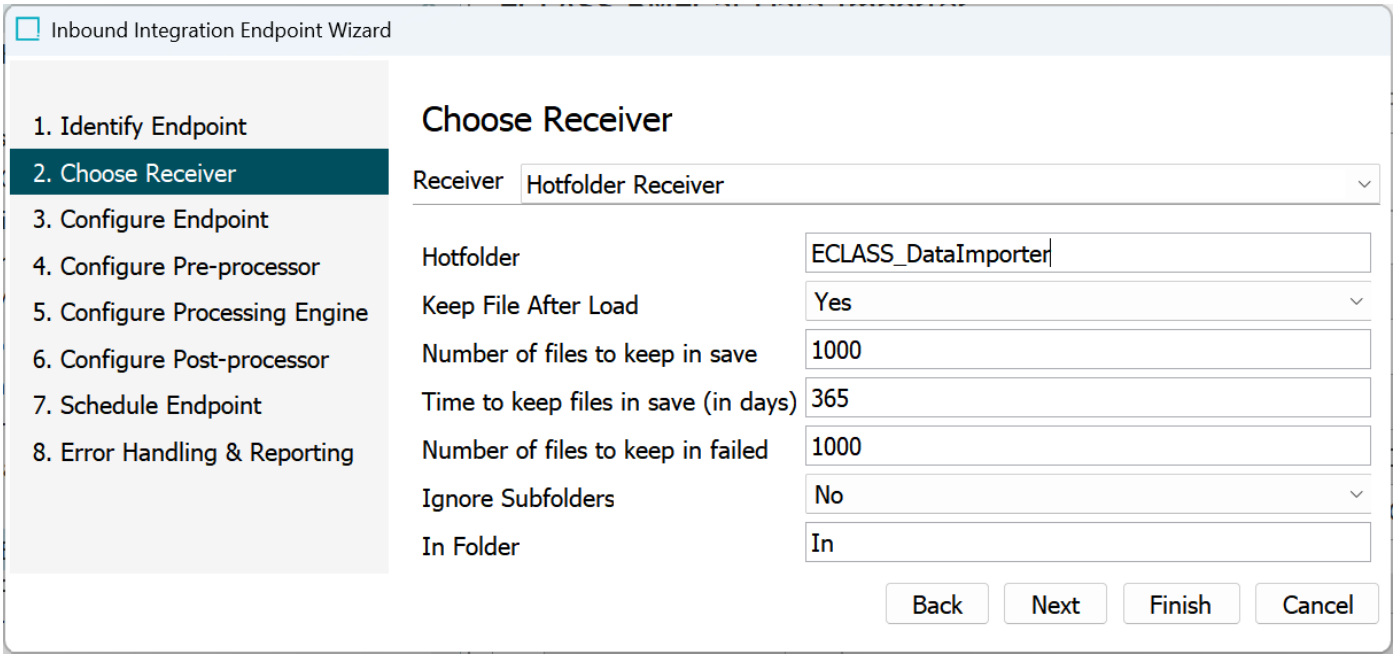
2. Once the Inbound Integration Endpoint Wizard displays, the parameters are to be populated as recommended and shown below.



For more information about the parameters available within the Identify Endpoint step, refer to IIEP - Identify Endpoint topic within the Data Exchange documentation.

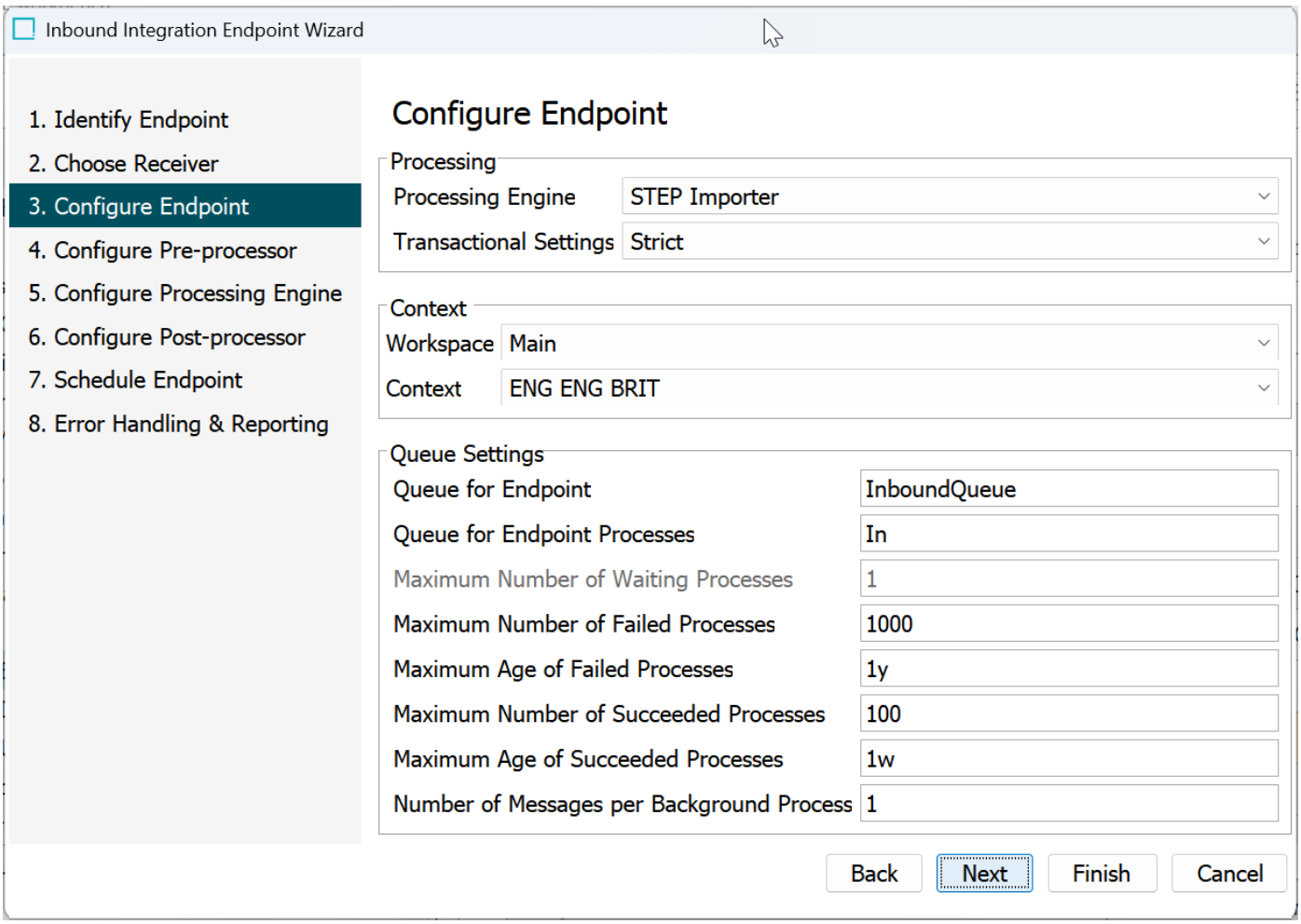
3. Click the **Next** button, and the Choose Receiver parameters will display. The parameters are to be populated as recommended and shown below. The mandatory parameter Hotfolder must be populated with a hotfolder name before the Next button will enable. In the screenshot below, the Hotfolder parameter is populated with the value 'ECLASS_DataImporter.'

> **Note:** The value within this hotfolder parameter will be used to create the new hotfolder, once the IIEP Wizard is complete.



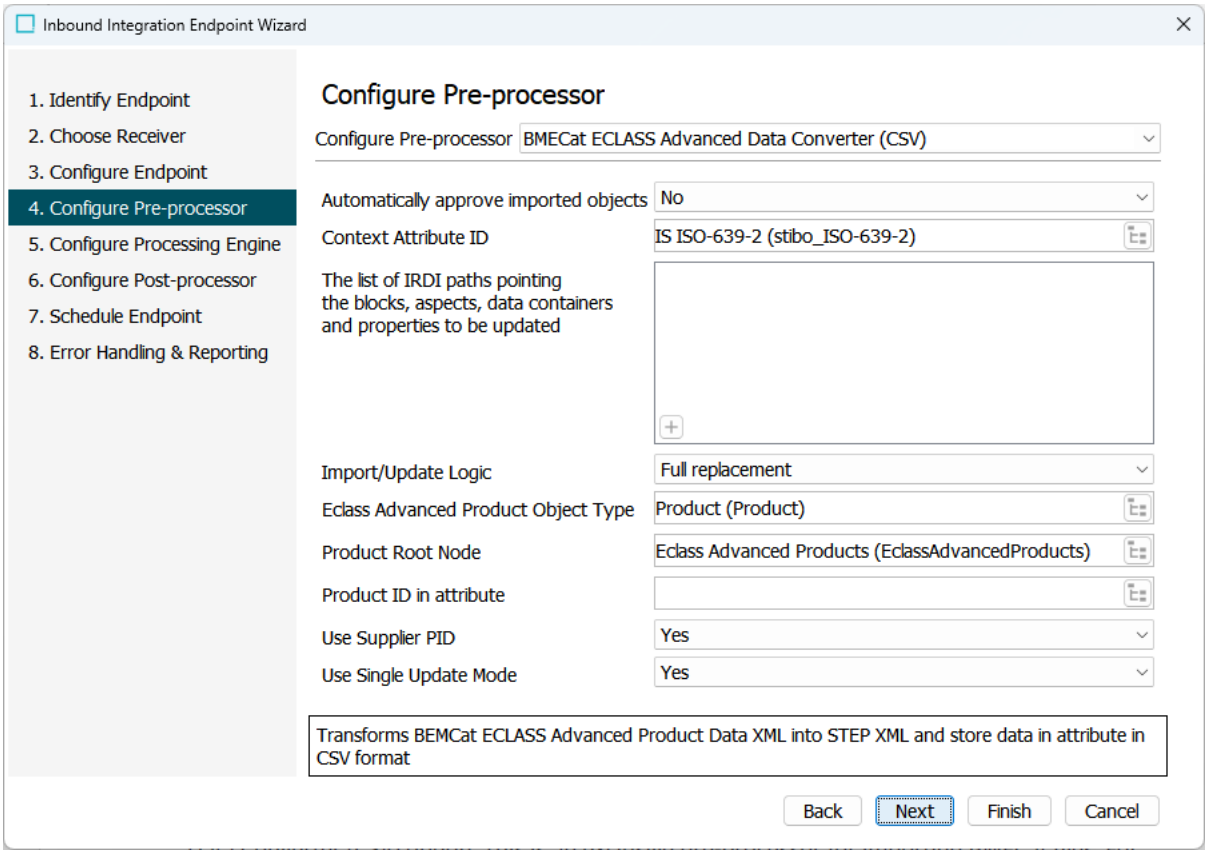
For more information about the parameters, refer to IIEP - Choose Receiver topic within the Data Exchange documentation.

4. Click the **Next** button, and the Configure Endpoint parameters will display. The parameters are to pre-populated with the recommended values as shown below.



For more information about the parameters, refer to IIEP - Configure Endpoint topic within the Data Exchange documentation.

5. Click the **Next** button, and the Configure Pre-processor parameter will display. The selection of the pre-processor within this step makes the IIEP unique for importing BMEcat 2005.1 data files. The parameters are to be populated as recommended and shown below:



- **Configure Pre-processor:** This parameter has to be populated with BMECat ECLASS Advanced Data Converter (CSV) option. This is an exclusive pre-processor for importing BMECat files. For more information about the parameter, refer toIIEP - Configure Pre-processor topic within the Data Exchange documentation.
- **Automatically approve imported objects:** This parameter is for the automated approval of SKUs / Products, Product Blocks, and Aspects. When configured as 'No' (default), no approval process is initiated. However, selecting 'Yes' results in the automatic approval of all imported elements, including Product Blocks, Aspects, SKU / Product, ECLASS ADVANCED attributes, and their corresponding product root nodes.
- **Context Attribute ID:** This parameter is to be populated with the attribute IS ISO-639-2 (ID = stibo_ISO-639-2). This attribute is created by the Easy Setup action and holds the language mappings. For more information about language mappings, refer toPrepare the Language Dimension Mapping topic.

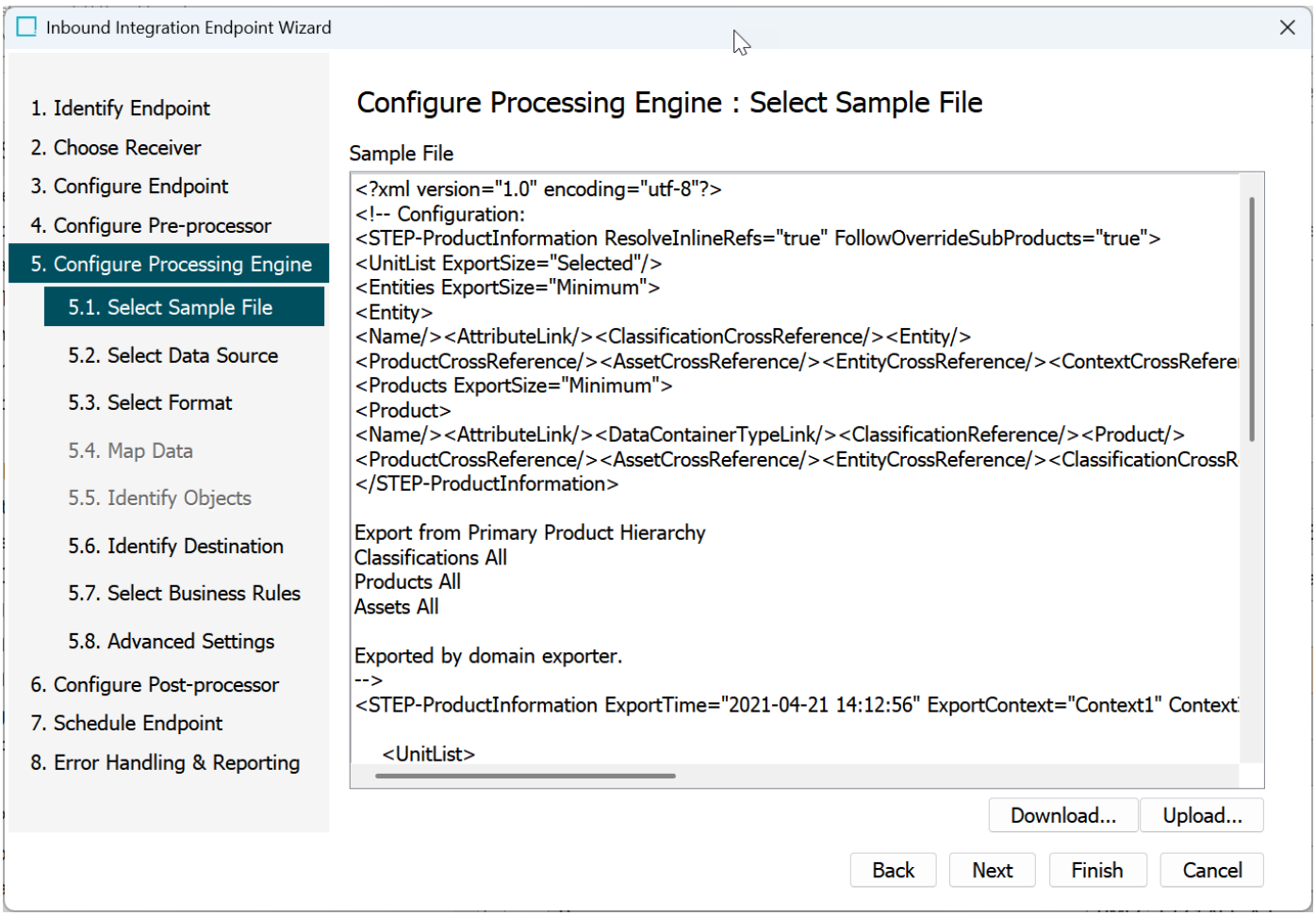
- **The list of IRDI paths pointing the blocks, aspects, data containers and properties to be updated:** Applicable only when the update strategy (Import/Update logic parameter defined below) is set to 'Partial replacement.' This parameter defines a targeted update scope during data import. It ensures that only the elements explicitly listed by their IRDI paths, such as blocks, aspects, data containers, and properties are updated. All other elements are excluded from the update process. The IRDI paths listed in this parameter act as filters that pinpoint which parts of the classification hierarchy should be updated. Only the elements matching these IRDI paths will be processed during import. This is particularly useful for partial replacement strategies, where you want to avoid overwriting the entire hierarchy.
- **Import/Update logic:** This parameter provides the following options:
    - **Full replacement:** Selecting this option leads to the prior-to-import removal of all Product Blocks and Aspects, along with the associated link to the Application Class. The actual Product / SKU remains unaltered.
    - **Import only new products:**If a product already exists in the system, its data will be excluded from the import. Only new products will be imported.
    - **Allow updates - for same Application class**: By selecting this option, the data import for the relevant SKU is updated if the Application Class in the import file matches the Application Class referenced in STEP. Conversely, if the Application Class in the import file does not correspond to the referenced Application Class in STEP, the data import for the relevant SKU is skipped.
    - **Allow updates - replace when different Application Class:** If the Application Class in the import file does not match the Application Class referenced in STEP, a full replacement of the Product / SKU takes place.
    - **Partial replacement:** This option updates only the blocks, aspects, data containers, and properties whose IRDI paths are defined in the parameter 'List of IRDI paths pointing to the blocks, aspects, data containers, and properties to be updated.' All other elements are excluded.
- **Eclass Advanced Product Object Type:** This parameter define the object type that should be assigned to imported products in the system if the import is intended to create a new product. If this parameter is left empty, the system will only update existing products and will not import any new ones. This parameter has to be configured with the main SKU / Product Object Type. The specified object type must be from the one for which the ECLASS Advanced attributes were deemed valid. Meaning, this is the same object type configured during the execution of the Easy Setup action. For information about the SKU / Product Object

type, refer to theRun Easy Setup of ECLASS Advanced Industry Standard topic.

> **Note:** This parameter is not automatically populated by the Easy Setup action, given that the solution may encompass multiple SKU / Product object types that are defined during the Easy Setup process.

- **Product Root Node:** Providing an existing product root node enables the creation of new products during the import process. These products are established beneath the specified product root node. Upon selecting a node, it is imperative to designate the corresponding object type. Leaving the field empty, however, restricts the creation of new products, permitting solely the update of existing ones.
- **Product ID in attribute:** This parameter facilitates the updating of existing SKUs in the system. If the SKU ID is specified within an attribute, then that attribute can be defined within this parameter. The uniqueness of the ID has to be taken into account while defining such attributes. In the event this field remains blank, the identifier will be automatically set to the Step ID.
- **Use Supplier PID:** When identifying the SKU through the 'SUPPLIER_PID,' opt for Yes (default). Alternatively, when using the 'MANUFACTURER_PID' for SKU identification, select 'No.'
- **Use Single Update Mode:** This parameter controls whether the STEP system enters a locked update state during certain import operations. When enabled, it ensures that only the import process can modify the database. All other users can still read data but are blocked from writing until the import completes.

6. Click the **Next** button, and the 'Configure Processing Engine: Select Sample File' field for the STEP Importer processing engine will display.



- In **5.1 Select Sample File** step, click the Upload button to upload a sample STEPXML file. For information about how to upload a sample file, refer to IIEP - Configure STEP Importer Processing Engine topic within the Data Exchange documentation.

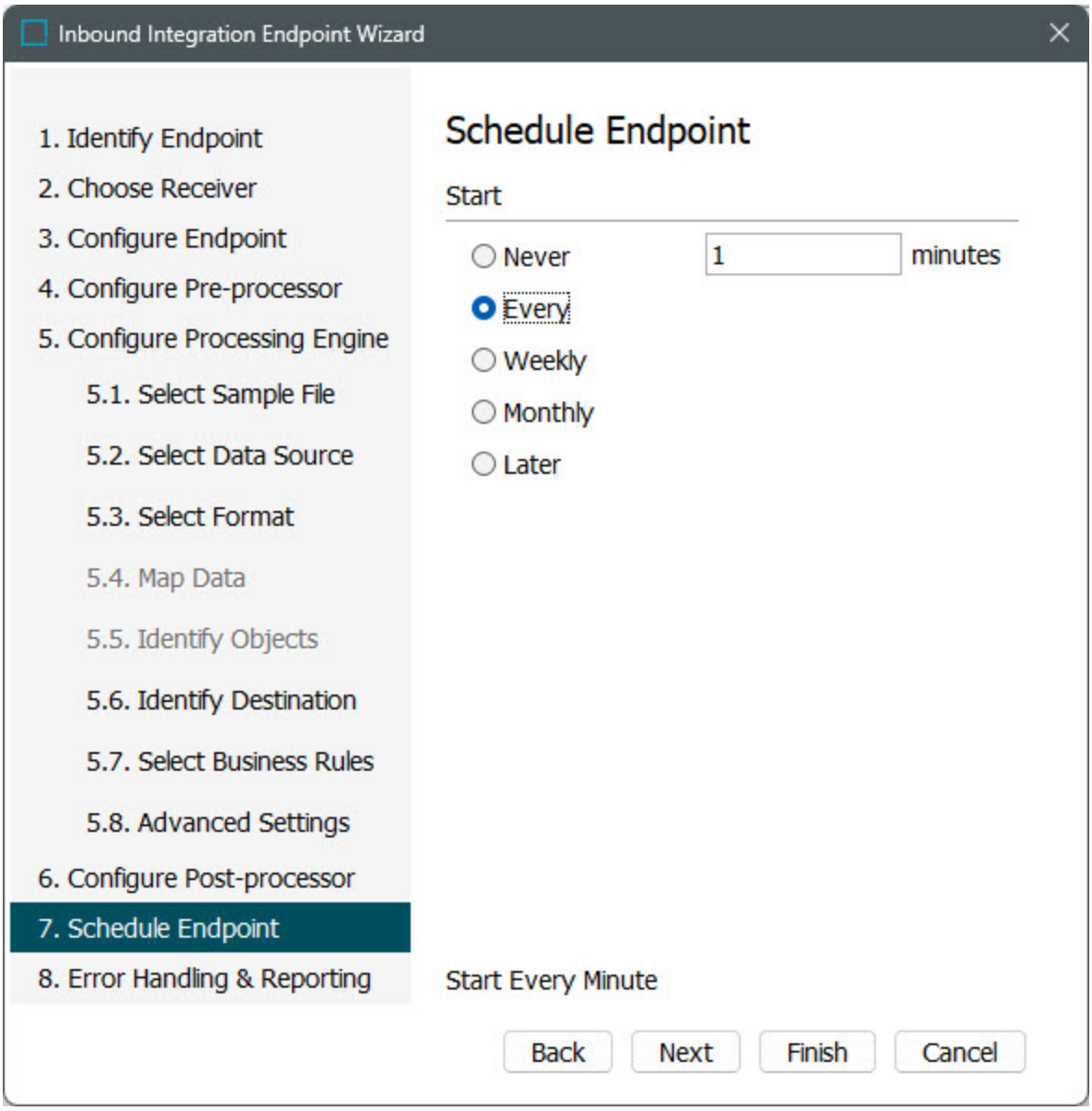
  The basic data structure of a sample Data file is provided below:

```
<?xml version="1.0" encoding="utf-8"?> <!-- Configuration: <STEP-
ProductInformation ResolveInlineRefs="true"
FollowOverrideSubProducts="true"> <UnitList ExportSize="Selected"/>
<Entities ExportSize="Minimum"> <Entity>
<Name/><AttributeLink/><ClassificationCrossReference/><Entity/>
<ProductCrossReference/><AssetCrossReference/><EntityCrossReference/>
<ContextCrossReference/><Values/></Entity></Entities> <Products
ExportSize="Minimum"> <Product>
```

<Name/><AttributeLink/><DataContainerTypeLink/><ClassificationReference/><Product/>
<ProductCrossReference/><AssetCrossReference/><EntityCrossReference/>
<ClassificationCrossReference/><Values/><OverrideSubProduct/></Product></Products> </STEP-ProductInformation> Export from Primary Product Hierarchy Classifications All Products All Assets All Exported by domain exporter. --> <STEP-ProductInformation ExportTime="2021-04-21 14:12:56" ExportContext="Context1" ContextID="Context1" WorkspaceID="Main" UseContextLocale="false"> <UnitList> <UnitFamily ID="eClass_BAJ271001" Selected="true" Referenced="true"> <Name>eClass acceleration</Name> <MetaData> <Value AttributeID="EclassDescription">increase in velocity within a certain interval as second derivative of the distance per time</Value> <Value AttributeID="EclassID">BAJ271</Value> <Value AttributeID="EclassPrimaryKey">BAJ271001</Value> </MetaData> <Unit ID="eClass_AAA225002" Selected="true" Referenced="true">
<Name>ft/sÂ²</Name> <MetaData> <Value AttributeID="EclassDescription">unit foot according to the Anglo-American and the Imperial system of units divided by the power of the SI base unit second with the exponent 2 with the relation according to NIST: 1 ft/sÂ² = 0,304 8 m/sÂ²</Value> <Value AttributeID="EclassImportVersion">10.1</Value> <Value AttributeID="EclassID">AAA225</Value> <Value AttributeID="EclassPrimaryKey">AAA225002</Value> </MetaData> </Unit> <Unit ID="eClass_AAA597002" Selected="true" Referenced="true">
<Name>m/sÂ²</Name> <MetaData> <Value AttributeID="EclassDescription">SI base unit metre divided by the power of the SI base unit second and the exponent 2</Value> <Value AttributeID="EclassImportVersion">10.1</Value> <Value AttributeID="EclassID">AAA597</Value> <Value AttributeID="EclassPrimaryKey">AAA597002</Value> </MetaData> </Unit> </UnitFamily> </UnitList> </STEP-ProductInformation>

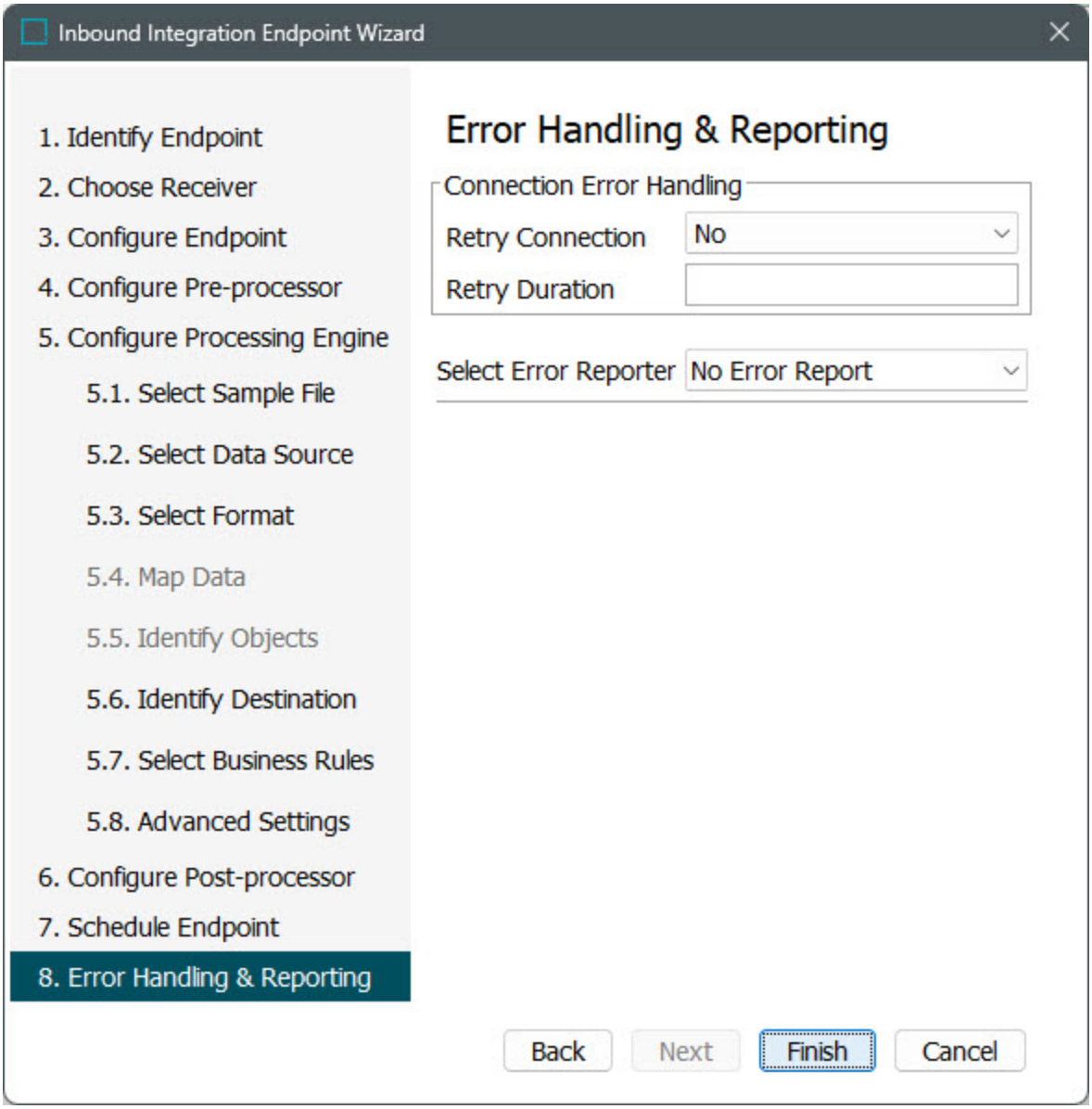
- Further potential sub-steps underneath the Configure Processing Engine are optional. For more information about these steps, refer to IIEP - Configure STEP Importer Processing Engine topic within the Data Exchange documentation.

7. Click the **Next** button to display the Schedule Endpoint parameters (bypass the Configure Post-processor parameters). Update the values to those shown below.



For information about the parameters available within this step, refer to IIEP - Schedule Endpoint topic within the Data Exchange documentation.

8. Click the **Next** button, and the Error Handling & Reporting step will display. The parameters are to be populated as recommended and shown below:



For more information about the parameters available within this step, refer to IIEP - Error Handling & Reporting topic within the Data Exchange documentation.

9. Click the **Finish** button, the Inbound Integration Endpoint Wizard will close, and the newly created endpoint will display within the workbench.

Important: An endpoint must be enabled before it can start processing data. For more information, refer to the Running an Inbound Integration Endpoint topic within the Data Exchange documentation.

If users need to access the IIEP via a Web UI, then the IIEP must be configured within a File Loading Widget. For more information, refer to the Configuring a File Loading Widget for ECLASS BMECat Data Imports topic.

Configuring a File Loading Widget for ECLASS BMECat Data Imports

Web UI users can import data files into STEP using a File Loading Widget. Follow the steps below to complete the configuration. This topic explains how to configure a File Loading Widget so users can drag and drop ECLASS BMECat data files onto it from the Web UI homepage.

Prerequisites

Before configuring the Web UI portion of this solution, an IIEP for an ECLASS Advanced Data Importer must be configured within the workbench. For more information, refer to Configuring an IIEP for ECLASS BMECat Data Imports topic.

It is expected that anyone configuring the ECLASS BMECat Data Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, refer to Designer Access topic within the Web User Interfaces documentation.

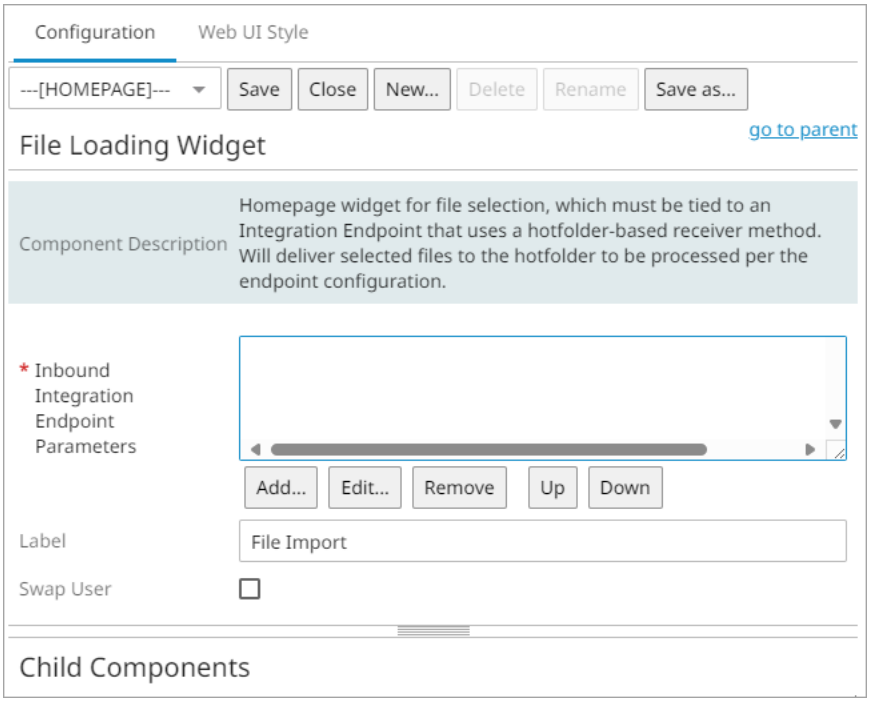
Additionally, it is helpful to know how to add a widget to a Web UI Homepage. Details on how to do this can be found in Adding Widgets to a Homepage topic in the Web User Interfaces documentation.

Configuration

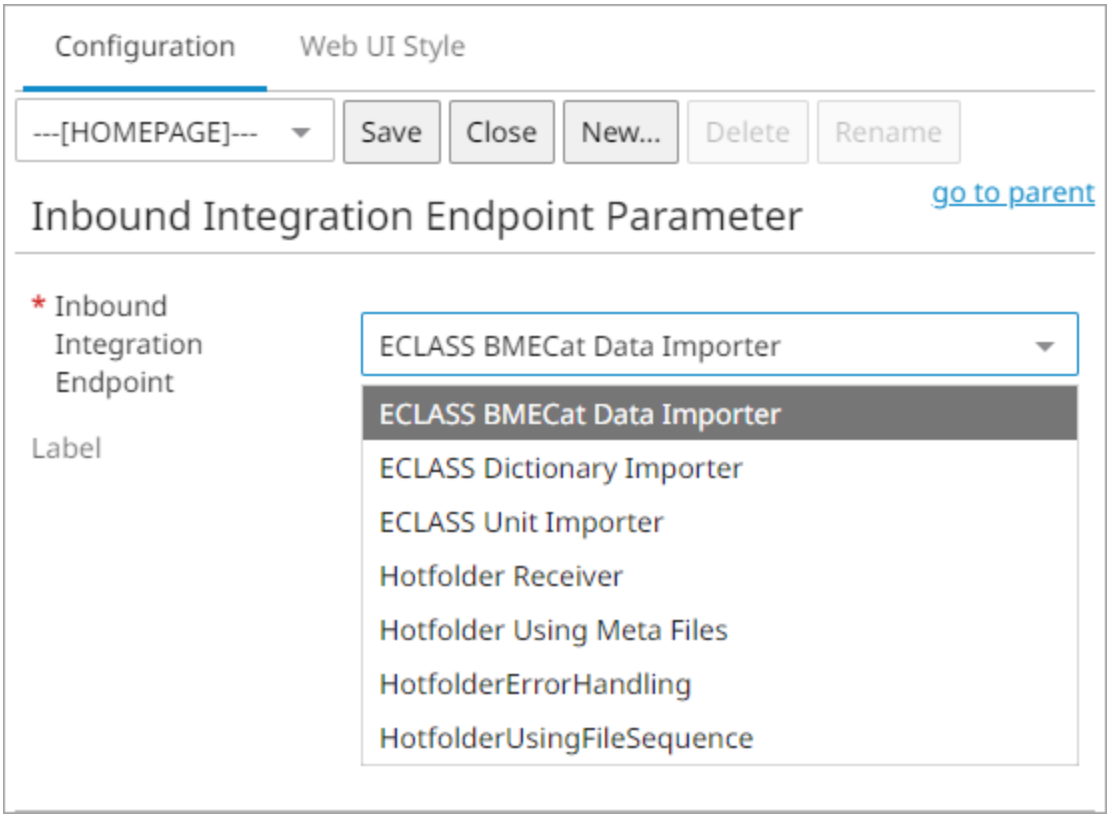
Each screenshot example within this section provides recommended values for the parameters in ECLASS BMECat Data Importer.

1. In the designer, select an existing File Loading Widget to be used, or add a new File Loading Widget to the Homepage Widget Grid component. For more information, refer to File Loading Widget topic within the Web User Interfaces documentation.

2. Go to the Inbound Integration Endpoint Parameters field, click the **Add** button, and the Inbound Integration Endpoint Parameter Properties dialog will display.



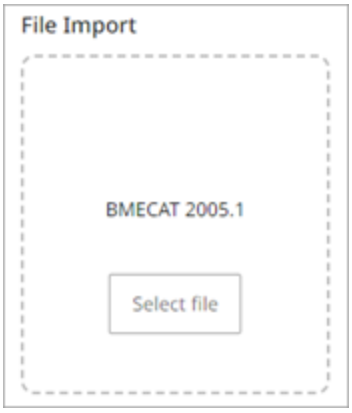
3. Click the dropdown for the Inbound Integration Endpoint parameter, and select **ECLASS ADVANCED Data Importer** (the IIEP created for ECLASS Advanced Data imports).



> **Note:** If the desired IIEP does not display in the dropdown, then it can be created using the steps described in Configuring an IIEP for ECLASS BMECat Data Imports topic.

4. Optionally, provide a label to be displayed within the drop zone of the widget.

   In the example below, a File Loading Widget labeled as 'ECLASS ADVANCED' is displayed above its configurations.



5. Click the **Save** and **Close** buttons to save the changes and close the designer.

Using ECLASS BMECat Data Importer

ECLASS BMECat Data file can be imported into STEP by uploading it to either a configured hotfolder, or through a File Loading Widget on a Web UI Homepage. The purpose of the Data Importer Web UI setup outlined in this topic is to offer a readily available solution for importing data in a supported ECLASS BMECat Data format. Only the BMEcat 2005.1 data is allowed to be imported as the ECLASS BMECat Data file.

Because only supported versions will successfully upload, before attempting to upload a Data file, confirm the file version being uploaded is listed within the ECLASS Standard Supported Versions and Formats topic.

Considerations before initiating an import

Before importing a BMEcat 2005.1 file into the system, it's important to consider the following:

- BMEcat 2005.1 is purposefully designed for transporting ECLASS BMECat Data, making it exclusively compatible with ECLASS Advanced.
- BMEcat data corresponds to specific ECLASS Advanced version(s), requiring the prior import of the corresponding / matching ECLASS Advanced version before commencing the BMEcat data import.
- The BMEcat 2005.1 Importer exclusively handles ECLASS BMECat data. Standard BMEcat (1.2 or 2005) fields will not undergo import.
- Unlike the ECLASS Advanced Dictionary file, which is language-specific, the BMEcat 2005.1 file includes multiple languages. Therefore, it's imperative to execute Language Mapping before initiating the import of BMEcat 2005.1 data.

Prerequisites

The Data Importer exclusively supports BMEcat 2005.1 file format. In case the XML file is zipped, it is essential to manually extract the .zip file to access the Data file.

Before engaging in any BMEcat 2005.1 import scenarios, it is essential to import the related ECLASS Advanced unit and ECLASS Advanced Dictionary file into the system.

The BMEcat 2005.1 Reference Feature System should be mapped with the corresponding ECLASS Advanced Version. For information about how to map BMEcat 2005.1 Reference Feature System with the Corresponding ECLASS Advanced Version, refer to Mapping BMEcat 2005.1 Reference Feature System with the Corresponding ECLASS Advanced Version topic available within the ECLASS Advanced Quick Start Guide.

Language Mapping has to be established. For information about how to perform Language Mapping, refer to Prepare the Language Dimension Mapping topic available within the ECLASS Advanced Quick Start Guide.

If you have completed the Easy Setup actions for the ECLASS Advanced Model, the functionalities outlined in this topic should be readily available for use. Unlike other ECLASS Advanced importers, the IIEP used for ECLASS BMECat Data Importer requires some manual configuration before using it for the first time. For information about updating the newly Easy Setup created IIEP, refer to Update IIEPs topic available within the ECLASS Advanced Quick Start Guide. For a detailed information on how to configure ECLASS BMECat Data importer, refer to Configuring ECLASS Advanced Data Importer topic.

Import Process Overview

Once a valid ECLASS Advanced file is uploaded using a File Loading Widget (or uploaded directly to a hotfolder), the file is picked up by an IIEP, and the IIEP starts a background process.

Procedure

1. Access the ECLASS Advanced Web UI Homepage.
2. Drag and drop a valid Data file into the 'Data Import' File Loading Widget, or upload to the hotfolder (root/upload/hotfolders/ECLASS_DataImporter/In).

   For information about File Loading Widget, refer to File Loading Widget topic within the Web User Interfaces documentation.

   For more information about uploading files directly to the hotfolder, refer to ECLASS Advanced Import Framework topic.
3. Once the upload has started, users can view the progress of the upload using the 'Recent background

processes' side panel.

For more information on using the 'Recent background processes' side panel, refer to Recent Background Processes Side Panel topic within the Web User Interfaces documentation.

After uploading the file, the IIEP picks it up and initiates a Background Process for the import process. The file loading widget does not provide background process monitoring in the Web UI. You can monitor the import status within the workbench through the IIEP Background Process that is generated.

For more information about monitoring the IIEP via background process, refer to Monitoring an IIEP via Background Process topic within the Data Exchange documentation.

ECLASS Standard Supported Versions and Formats

The following are the supported import and export versions and/or formats for the ECLASS Advanced standard.

- **Data Importer:** BMEcat 2005.1, BMEcat 2005.2
- **Data Exporter:** BMEcat 005.1, BMEcat 2005.2

Exporting in BMEcat Formats using BMECat 2005.1 Converter

The BMECat 2005.1 Converter format enables export of product data in BMECat 2005.1 and the BMECat 2005.2 XML standard.

Exporting the BMEcat formats involves supplying header data and catalog / product data via the parameters displayed on the Select Format step and during the Map Data step. Some information is required, as is indicated on the Map Data step.

> **Note:** While it is possible to initiate the export process even in the absence of mandatory fields, it is important to note that the background process (BGP) will fail.

Format Availability

BMECat 2005.1 Converter (which can export BMEcat 2005.1 and BMEcat 2005.2 XML) can be triggered from:

- Export Manager - refer to Creating a Data Export
- OIEP - refer to Creating an Outbound Integration Endpoint

Mapping

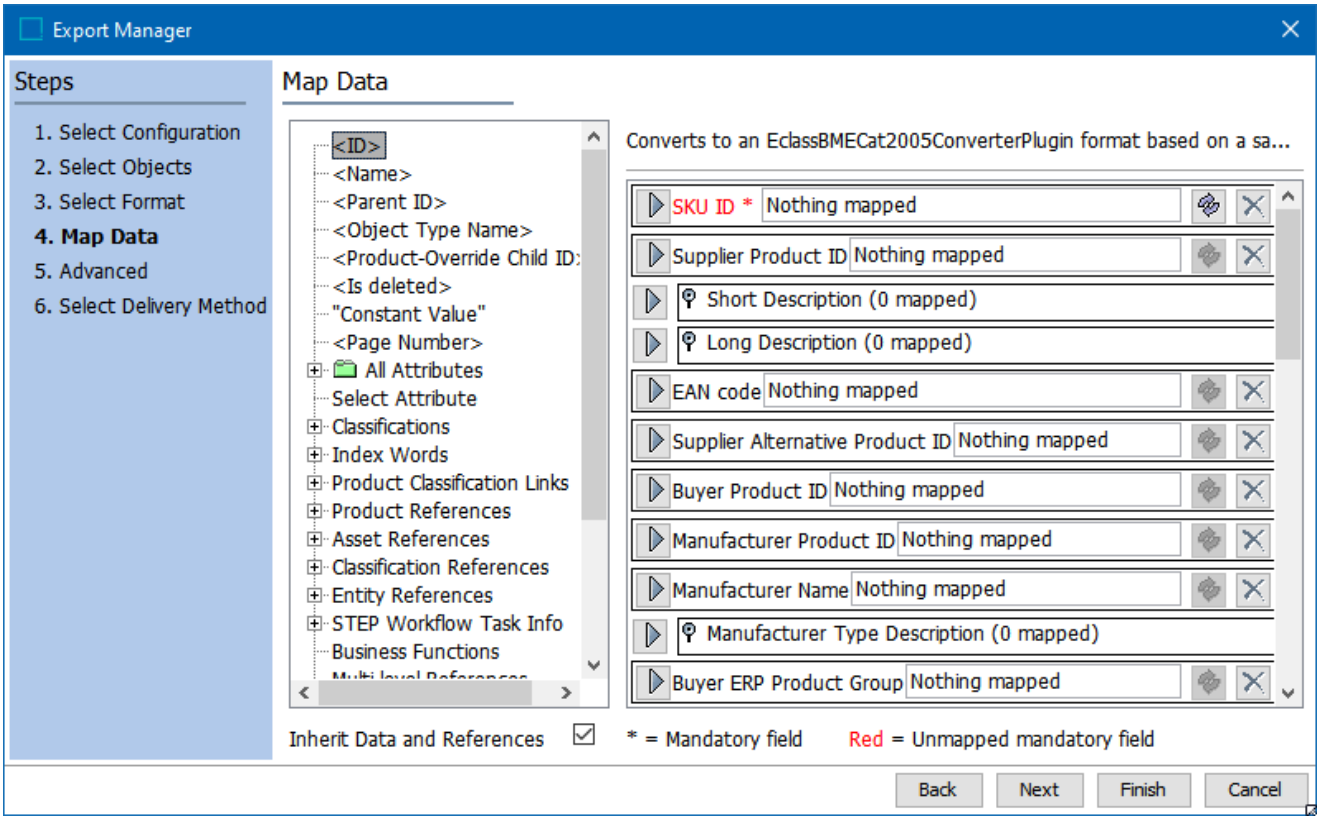
This format requires creating a data map between STEP and the data being processed, and may also include data transformations. For further details on mapping, refer to Data Mapping.

Selecting the BMEcat 2005.1 or BMEcat 2005.2 options within the BMEcat 2005.1 Converter Export format will automatically generate the data within the <PRODUCT_FEATURES></PRODUCT_FEATURES> Tag with the in STEP available ECLASS Advanced content. The extension plug-in will produce the needed information for the output.

Standard BMEcat fields can be mapped with the usual mapping process. For more information on mapping, refer to the Outbound Map Data Options topic in the Data Exchange documentation.

To map for ECLASS Advanced, the STEP ID of the SKU / Product must be mapped to the mandatory field 'SKU ID.'

Should other relevant fields need to be mapped, they can be mapped as needed.



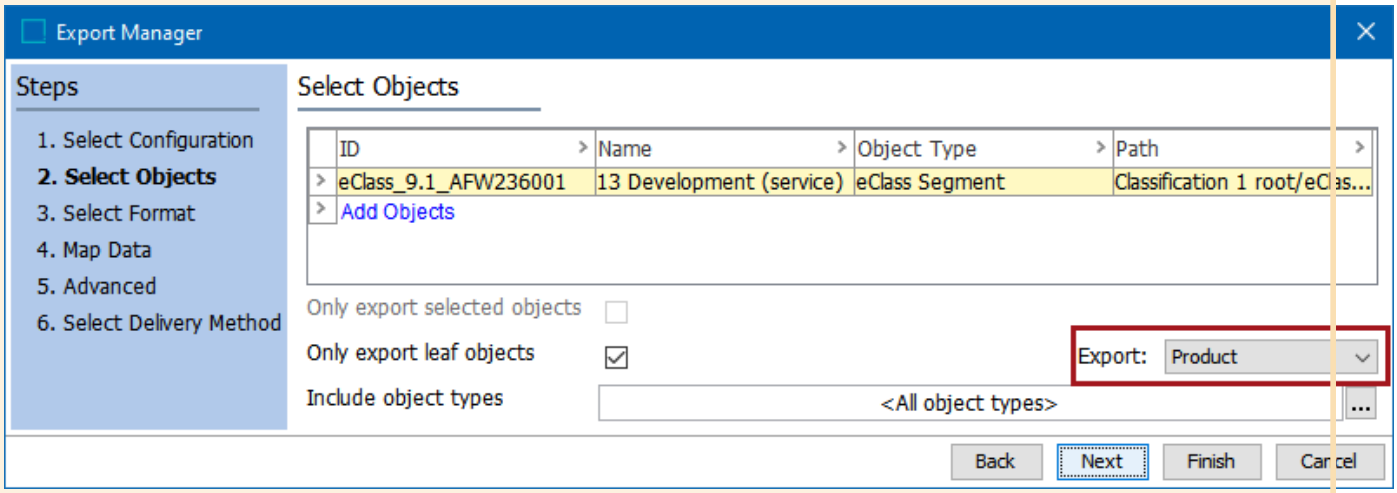
Configuring BMEcat Exporter

The technical starting point for the Exporter is always the SKU / Product. Therefore, the user should trigger the exporter from either the SKU / Product or a relevant Industry Standards classification node from the top Industry Standard level downwards.

Note: Only mapped languages are determined for output.

1. In the workbench, initiate the export for the product(s) you intend to export. For information on how to initiate an export refer to the Creating a Data Export topic in the Data Exchange documentation.

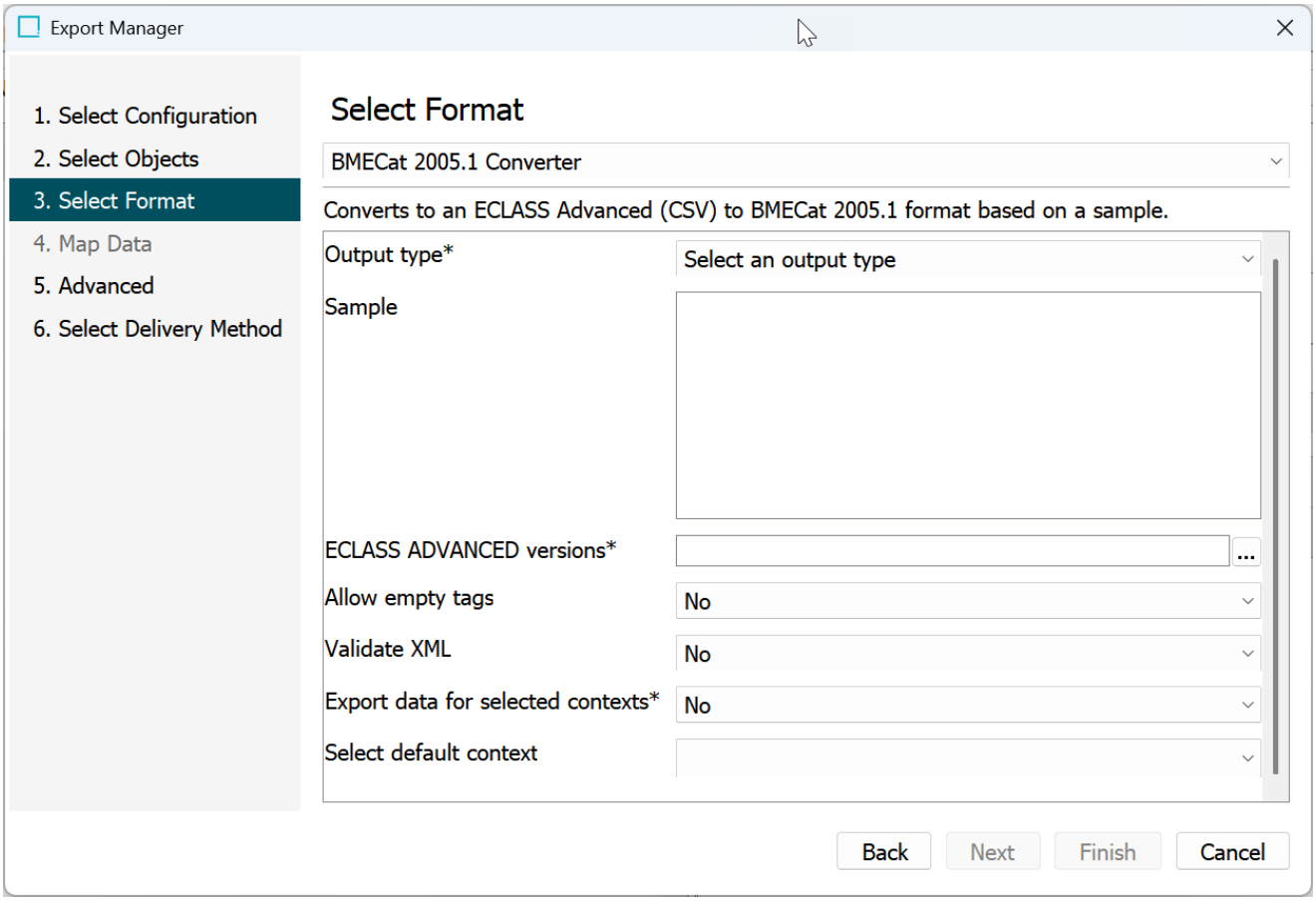
Important: When exporting from a Classification node, in the Export dropdown select 'Product.'



2. On the Select Format step of the Export Manager, fill out the following Outbound Parameters.

Outbound Parameters

The parameters that are available to fill out are controlled by the XML template that is uploaded. The fields listed below are displayed for all users.



- **Output Type**: This parameter determines the format and structure of the exported file. Available options are BMECat 2005.1 and BMECat 2005.2. Based on the selected format, the next parameter (Sample) is automatically populated with the corresponding template.

  Sample: An XML template file is loaded automatically but can be modified as required. This defines the format of the XML file to be exported.

  > **Note:** For more information on Generic XML instructions, refer to the Generic XML Outbound Processing Instructions topic in the Data Exchange documentation.

- **ECLASS ADVANCED versions**: This field is mandatory. If no version is selected, no SKUs or products will be exported.

  It specifies the root classification node of the version to be exported. To select the relevant ECLASS Advanced version(s), click the eclipses button, and choose the desired version from the list.

> **Note:** Only versions that have already been imported into the system are available for selection.

- **Allow empty tags**: Selecting **Yes** indicates that export tags with empty values are included in the output. If set to **No**, tags containing empty values are not included in the export.
- **Validate XML**: Selecting **Yes** indicates that the BMEcat 2005.1 or BMEcat 2005.2 export file is validated against an XSD included in STEP. When validation fails, the background process also fails, and the problem is reported in the BGP execution report. If set to **No**, the validation is skipped and the BGP does not fail due to differences found when comparing to the XSD.
- **Export data for selected contexts**: This allows you to filter and export data only for specific contexts rather than exporting all available contexts. Select **Yes** in the dropdown to display the **Select Contexts** link. Click the link, select the required contexts for the export from the Select Contexts dialog, and click the **Select** button. The selected contexts are listed in a text box.

  The chosen contexts will be presented as <LANGUAGE> elements within the <HEADER> section of the exported XML file. If the system contains the specified context-dependent attribute values, they will be incorporated into the <PRODUCT_DETAILS> segment.

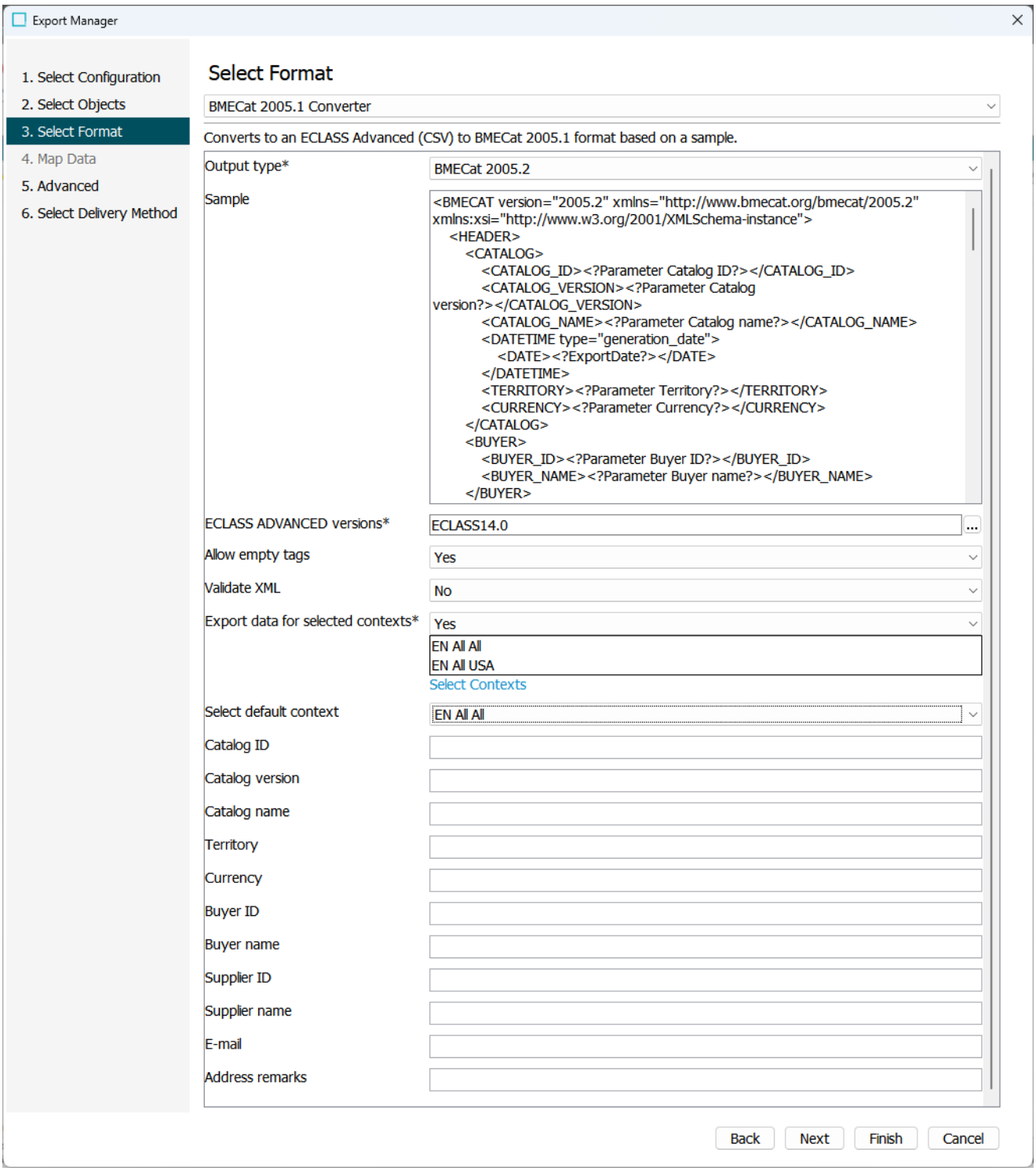
  > **Important:** When using an OIEP for BMECat format export, ensure consistency between the contexts specified in the 'Export data for selected contexts' parameter and OIEP > Configuration tab > Configuration flipper > Contexts parameter. A mismatch may lead to contexts being presented differently in the exported file: those added in OIEP > Configuration tab > Configuration flipper > Contexts parameter will appear in the <PRODUCT_DETAILS> element, and those in 'Export data for selected contexts' parameter will be in the <HEADER> element.

Select default context: This shows the default context that is selected for the output. The parameter only displays contexts that were selected in the **Export data for selected contexts** parameter, and automatically selects the first context from the list. However, this can be changed by clicking the dropdown menu and selecting a different option.

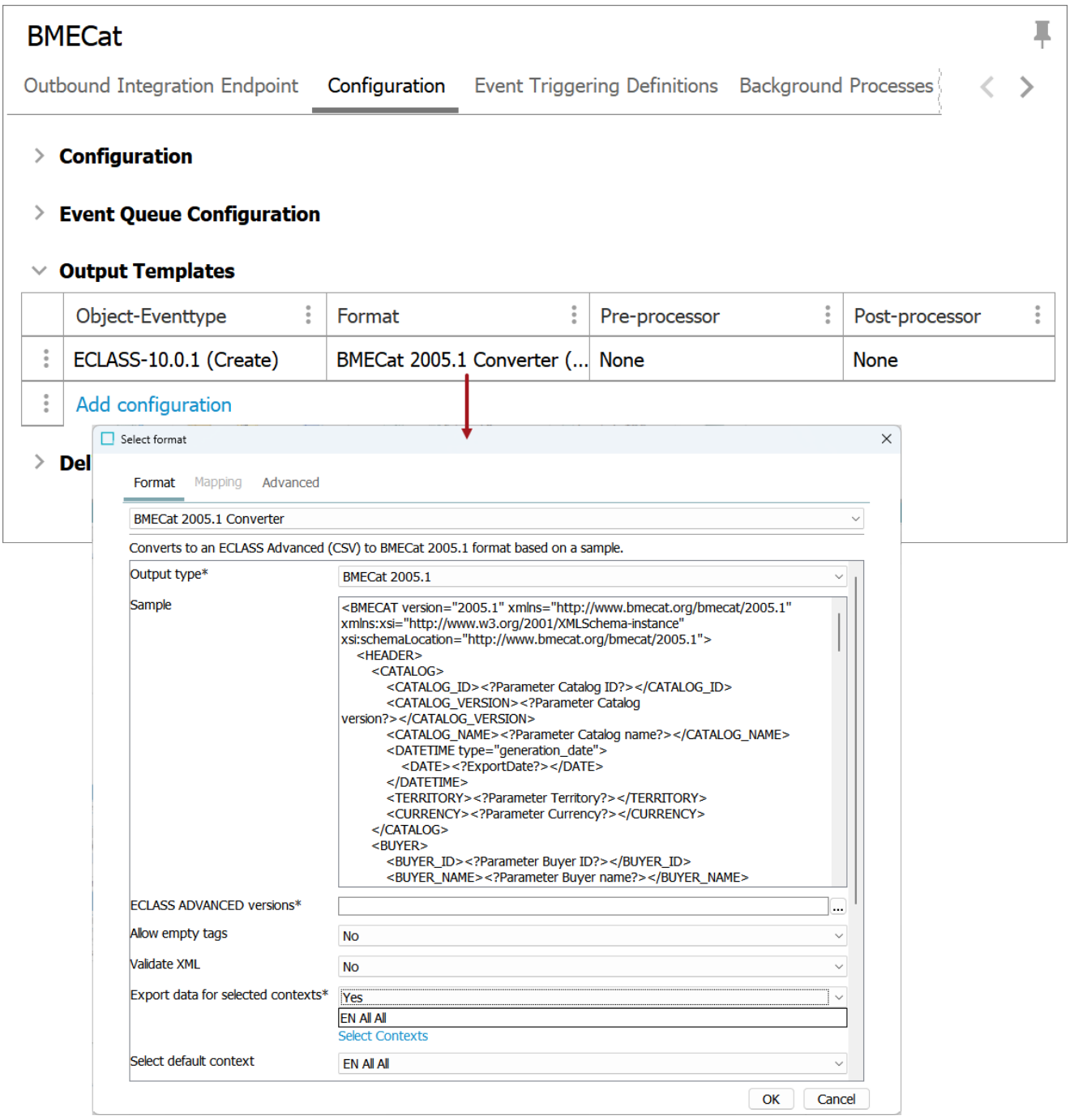
Note: For exports done in the BMEcat file format, all cross-context exports, by default, always contain suppressed references. However, to omit suppressed references, the following configuration property must be added: `ExportManager.Omit.SuppressedReferences.CrossContext=true` This functionality is managed by a configuration property that is not available in the Self-Service UI. Contact Stibo Systems Support for assistance.

For an explanation of the remaining parameters, search the web. No validation is performed on the text entered or the selections made, but if Validate XML = Yes, errors will be reported in the execution report, as defined above.

Export Manager


Export Manager
1. Select Configuration
2. Select Objects
3. Select Format
4. Map Data
5. Advanced
6. Select Delivery Method
Select Format
BMECat 2005.1 Converter
Converts to an ECLASS Advanced (CSV) to BMECat 2005.1 format based on a sample.
Output type*
BMECat 2005.2
Sample
<BMECAT version="2005.2" xmlns="http://www.bmecat.org/bmecat/2005.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<HEADER>
<CATALOG>
<CATALOG_ID><?Parameter Catalog ID?></CATALOG_ID>
<CATALOG_VERSION><?Parameter Catalog
version?></CATALOG_VERSION>
<CATALOG_NAME><?Parameter Catalog name?></CATALOG_NAME>
<DATETIME type="generation_date">
<DATE><?ExportDate?></DATE>
</DATETIME>
<TERRITORY><?Parameter Territory?></TERRITORY>
<CURRENCY><?Parameter Currency?></CURRENCY>
</CATALOG>
<BUYER>
<BUYER_ID><?Parameter Buyer ID?></BUYER_ID>
<BUYER_NAME><?Parameter Buyer name?></BUYER_NAME>
</BUYER>
ECLASS ADVANCED versions*
ECLASS14.0
Allow empty tags
Yes
Validate XML
No
Export data for selected contexts*
Yes
EN All All
EN All USA
Select Contexts
Select default context
EN All All
Catalog ID
Catalog version
Catalog name
Territory
Currency
Buyer ID
Buyer name
Supplier ID
Supplier name
E-mail
Address remarks
Back
Next
Finish
Cancel

OIEP


BMECat
Outbound Integration Endpoint
Configuration
Event Triggering Definitions
Background Processes
Configuration
Event Queue Configuration
Output Templates
Object-Eventtype
Format
Pre-processor
Post-processor
ECLASS-10.0.1 (Create)
BMECat 2005.1 Converter (...
None
None
Add configuration
Select format
Format
Mapping
Advanced
BMECat 2005.1 Converter
Converts to an ECLASS Advanced (CSV) to BMECat 2005.1 format based on a sample.
Output type*
BMECat 2005.1
Sample
<BMECAT version="2005.1" xmlns="http://www.bmecat.org/bmecat/2005.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.bmecat.org/bmecat/2005.1">
<HEADER>
<CATALOG>
<CATALOG_ID><?Parameter Catalog ID?></CATALOG_ID>
<CATALOG_VERSION><?Parameter Catalog
version?></CATALOG_VERSION>
<CATALOG_NAME><?Parameter Catalog name?></CATALOG_NAME>
<DATETIME type="generation_date">
<DATE><?ExportDate?></DATE>
</DATETIME>
<TERRITORY><?Parameter Territory?></TERRITORY>
<CURRENCY><?Parameter Currency?></CURRENCY>
</CATALOG>
<BUYER>
<BUYER_ID><?Parameter Buyer ID?></BUYER_ID>
<BUYER_NAME><?Parameter Buyer name?></BUYER_NAME>
ECLASS ADVANCED versions*
Allow empty tags
No
Validate XML
No
Export data for selected contexts*
Yes
EN All All
Select Contexts
Select default context
EN All All
OK
Cancel

Note: When using an OIEP for BMECat format export, there is a subtle distinction in how contexts are filled in the <HEADER> and <PRODUCT_DETAILS> elements. Specifically, for the context-dependent attribute values populated in the <PRODUCT_DETAILS> element, the values set in OIEP > Configuration tab > Configuration flipper > Contexts parameter take precedence and are populated accordingly. However, the contexts presented as <LANGUAGE> elements within the <HEADER> section are determined by the settings made in the above mentioned 'Export data for selected contexts' parameter itself.