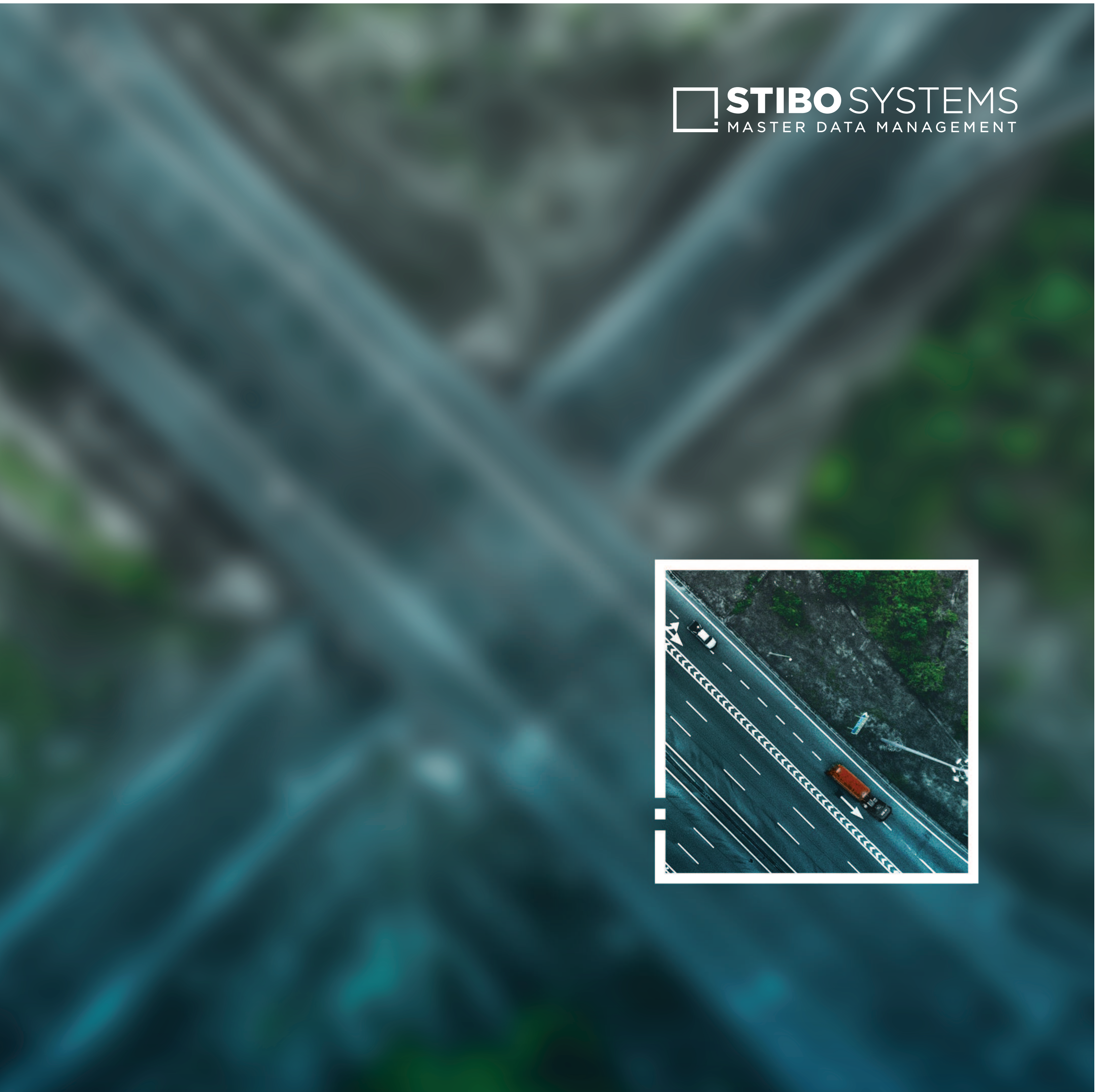


STIBO SYSTEMS
MASTER DATA MANAGEMENT

SETUP and USER GUIDE

Data Integration

Rel 10.2-MP3 (September 21, 2021)

Table of Contents

Data Integration

The topics in this section detail data integration between STEP and other systems.

It is important to have a solid understanding of how the platform works before working through the topics in this section. It also helps to be familiar with the third-party systems and their functionality.

- Asynchronous Translations
- Experian Email Validation Integration
- Loqate Integration
- Machine Learning-Based Auto Classification Integration
- Product Data Exchange
- Syndigo

Asynchronous Translations

This topic focuses on setting up and using the automatic service capabilities in STEP for translations so that translatable data can be sent to and from your translation service via their REST API, without human intervention. STEP supports Asynchronous Translations for SDL and Lionbridge translation services, with the differences in setup being only in the connection details.

Users who manage a local Across Language Server (referred to as 'Across' below) are working with a translation software, rather than a translation service agency. Translations done in Across can also use the asynchronous translations functionality to track the STEP data translation process. For more information on Across, search the web.

For users that do not have Across, Lionbridge or SDL, STEP offers the add-on component File Exchange Service. The File Exchange Service allows users to import / export translation files to folders without the need for an API.

Using an asynchronous translation configuration, translations are initiated via a business action and the jobs are monitored in the 'Translation Status Widget' in the Web UI.

Note: If you are new to translations in STEP, it is suggested that you read the high-level overview in the **Translations** documentation.

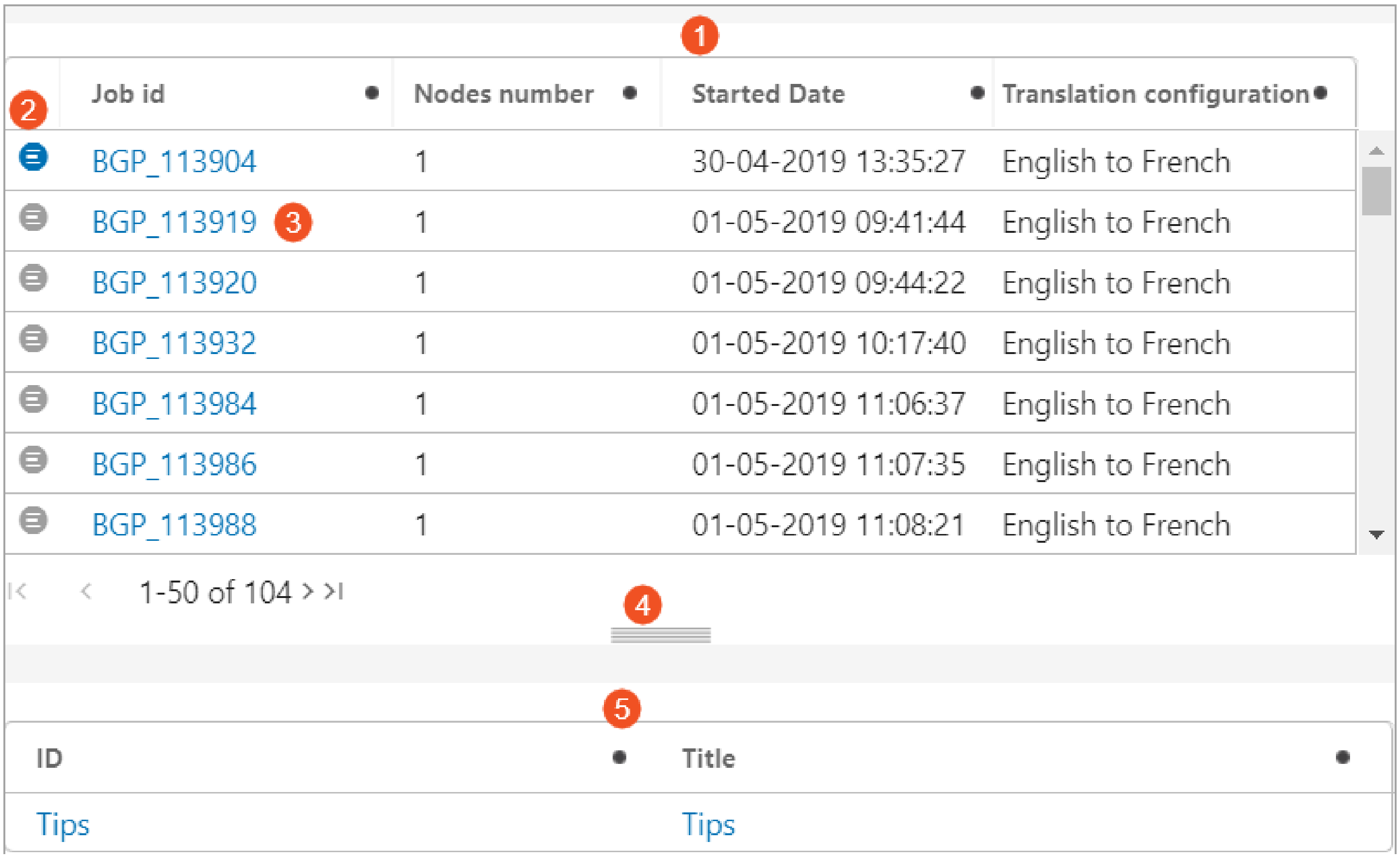
Asynchronous Translations can be configured using the following topics:

- Create the Asynchronous Services Object Type
- Configuring an Asynchronous Translation Service
- Configuring an Asynchronous File Exchange Service
- Setting Up a Translation Configuration
- Configuring Screens for Asynchronous Translation Status in Web UI
- Attribute Filters for Asynchronous Translation Services
- Business Rules for Asynchronous Translations
- Asynchronous Translations in Web UI
- Translation Status Widget Configuration

To access and use the File Exchange Service, Across, SDL, or Lionbridge components, they must first be activated as add-on components on your system and implemented with their corresponding framework. Instructions for installing components can be found in the **SPOT Program** topic in the **System Administration** documentation. Contact your account manager or partner manager to begin the process of enabling a license or licenses for your system.

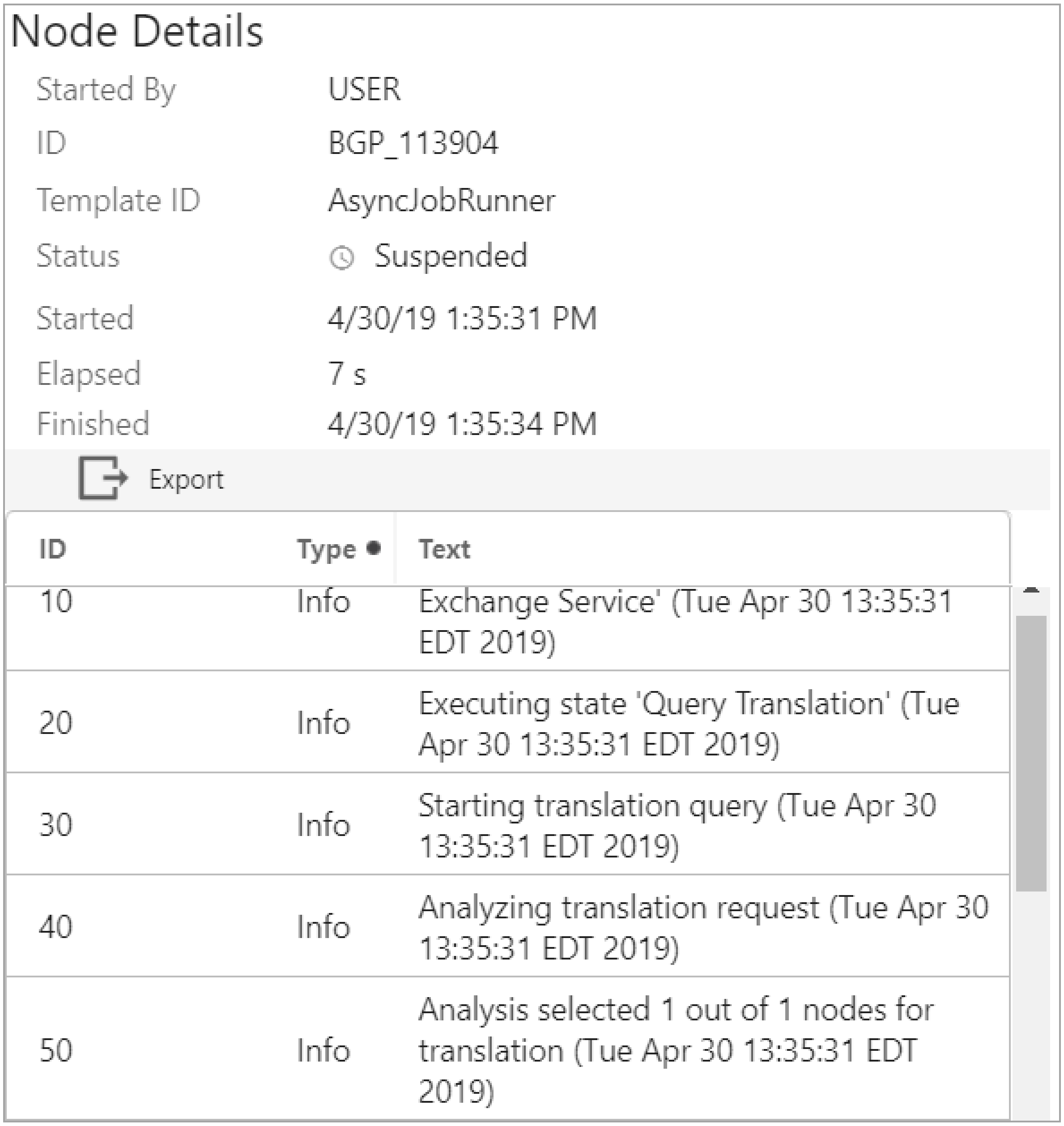
Asynchronous Translation Status in Web UI

Using the Translation Status Widget along with a combination of configured screens, users are able to check the status of asynchronous translation statuses in Web UI. Once configured, the widget, along with the screens, can be used to give very specific details about the asynchronous translation, as shown in the images below.



1. **The Job List Screen:** The job list screen includes the list of objects that contain translation statuses. The column headers for this screen are selected during configuration.
2. **Job Node List Screen Icon Selector**: Clicking this icon accesses the Job Node List Screen for the translation status of the selected object, which is shown below the handle (see number 5 and 6).
3. **Background Process (BGP) for Translation Status of Selected Object:** Clicking the BGP link will open a Node Details screen, which includes detailed translation status information about the selected object.
4. **Moveable Handle:** Separates Job List Screen (above the handle) and the Job Node List Screen (below the handle). Move this handle up or down to view more or less of the desired screen.
5. **Job Node List Screen:** Details information regarding selected translation. Header columns 'ID' and 'Title' are set by default, but can be modified during screen configuration.

When the BGP link is clicked (as described in number 3 above), a Node Details screen will open. This screen details specific information regarding the various states of translation for the selected object.



Prerequisites prior to configuring Web UI for asynchronous translations

Before accessing asynchronous translation status details in Web UI, users are required to:

1. In the workbench, create a translation configuration and include at least one asynchronous service in that configuration. For more information, refer to the **Asynchronous Translations** topic.
2. Configure the screens and mappings necessary to view asynchronous translation status details in Web UI. For more information, see the **Configuring Screens for Asynchronous Translation Status in Web UI** topic.

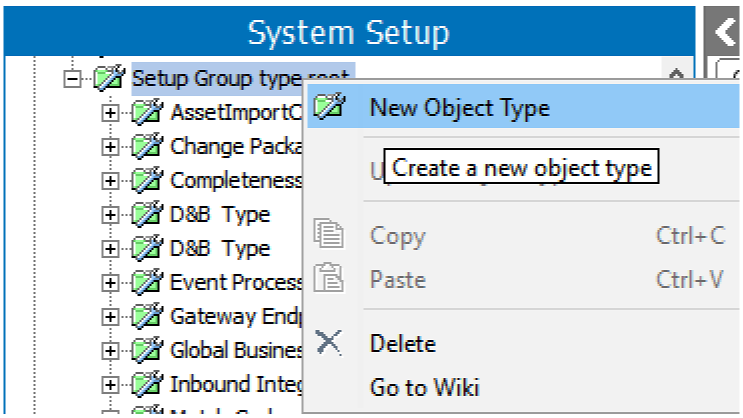
3. Configure the Translation Status widget for Web UI. see the **Translation Status Widget Configuration** topic.

Create the Asynchronous Services Object Type

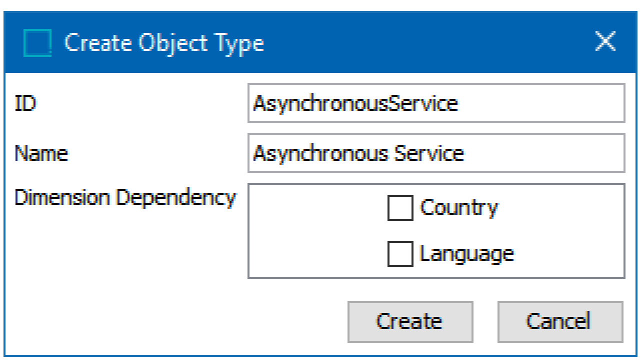
Before an asynchronous service can be set up and configured, an Asynchronous Services setup group root node and object type must be created. For more information about creating Setup Groups in general, see the **Setup Groups** section of the **System Setup / Super User Guide**.

Use the following steps to create the object type.

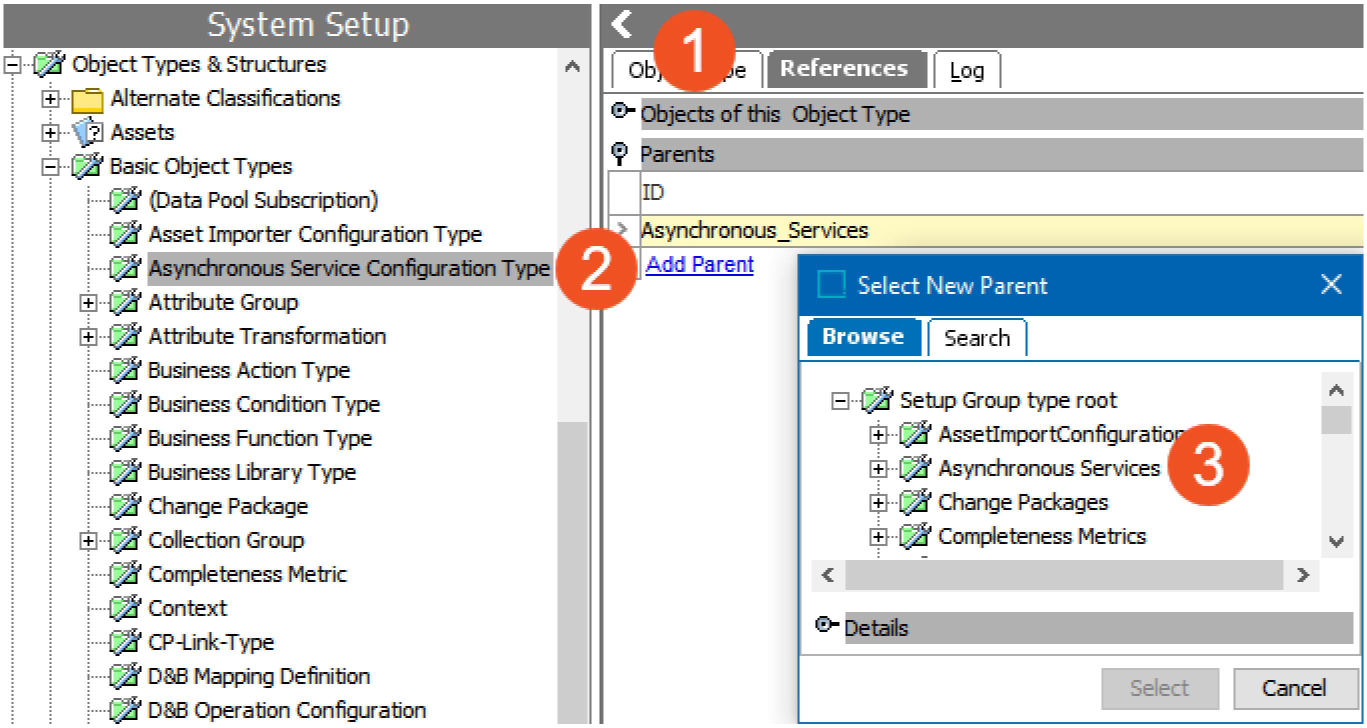
1. From 'System Setup', expand 'Object Types and Structures' and right-click **Setup Group type root**. Select **New Object Type**.



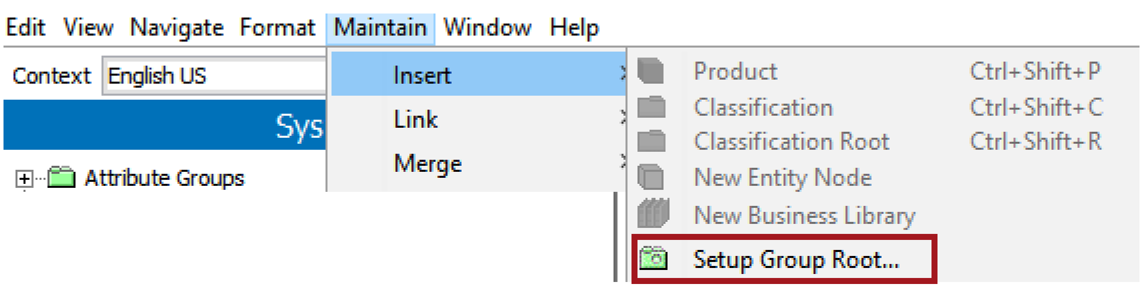
2. Enter a Name and ID in the **Create Object Type** dialog box.



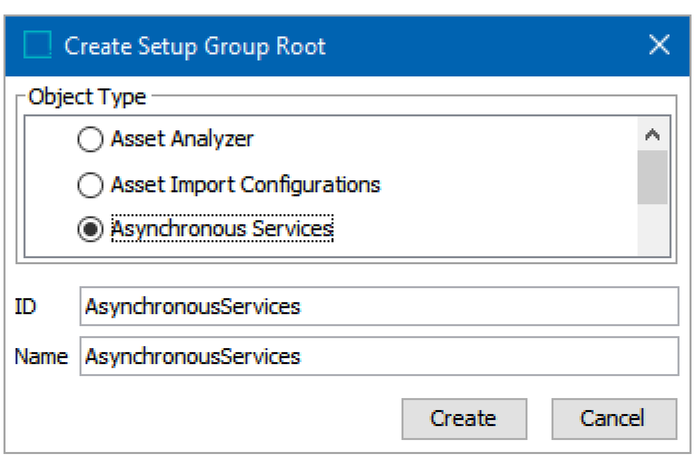
3. Remaining in 'Object Types & Structures', navigate to and expand 'Basic Object Types.' Select the 'Asynchronous Service Configuration Type.' Click on the References tab and select 'Add Parent.' In the **Select New Parent** dialog box, select the setup group root that was created in step two, and click the Select button to add it as a Parent.



4. Now the Asynchronous Service can be added as a root node under the 'System Setup' structure. In 'System Setup', select the **Maintain** menu, **Insert**, and **Setup Group Root**.



5. Select the Asynchronous Services root node that was created and give it an ID and a Name, and click **Create**.



The Asynchronous Services setup group root and object type have been created

6. Configure an asynchronous service as defined in the following topics:
   - Configuring an Asynchronous Translation Service
   - List Processing External Processing Operation

Configuring an Asynchronous Translation Service

This section covers setting up a new Asynchronous Translation Service. All steps in this section are applicable to Across, SDL, and Lionbridge services except in the **Server Connection Details**, which have been broken out by service.

> **Note:** For information regarding configuring an Asynchronous File Exchange Service, see the **Configuring an Asynchronous File Exchange Service** topic.

Prerequisites

1. For the Across Translation Server software only, before performing the Configuration below in STEP, create a 'STEP' document template on the Across server using the 'Tagged XML v2' option. This template enables Across to read STEPXML files. For more information, contact Across support.
2. For any asynchronous service option other than File Service Exchange, prior to configuration, clicking the **Server URL** dropdown parameter displays the required property name. Provide a selection for the dropdown parameter via the sharedconfig.properties file on the STEP application server using the appropriate case-sensitive property:

   - **Async.Kernel.Config.Across.ServerURL**
   - **Async.Kernel.Config.LionBridge.ServerURL**
   - **Async.Kernel.Config.SDL.ServerURL**

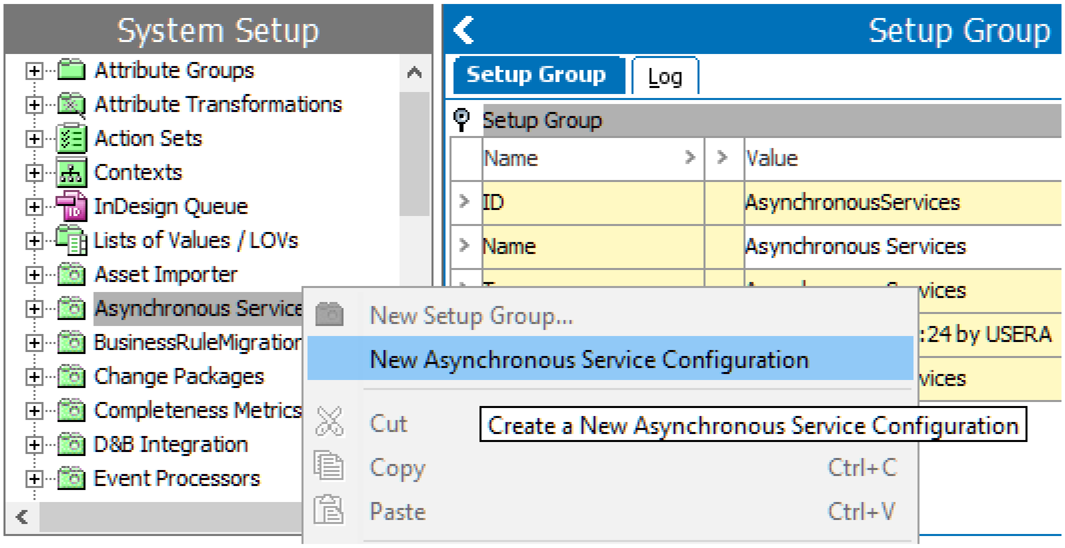
   The following is an example of a complete property entry for two Across systems:

   ```
   Async.Kernel.Config.Across.ServerURL=1=http://across.scloud.com/crossAPI/crossAPI.wsd
   l,2=http://across.scloud2.com/crossAPI/crossAPI.wsdl
   ```

   > **Note:** Across Translation Server software users have the option to add a custom port for communication between STEP and Across Translation Server software by adding the port to the property entry within the sharedconfig.properties file. For example, in the property entry shown below, the '5000' is the custom port.
   >
   > ```
   > Async.Kernel.Config.Across.ServerURL=1=http://127.0.0.1:5000
   > ```

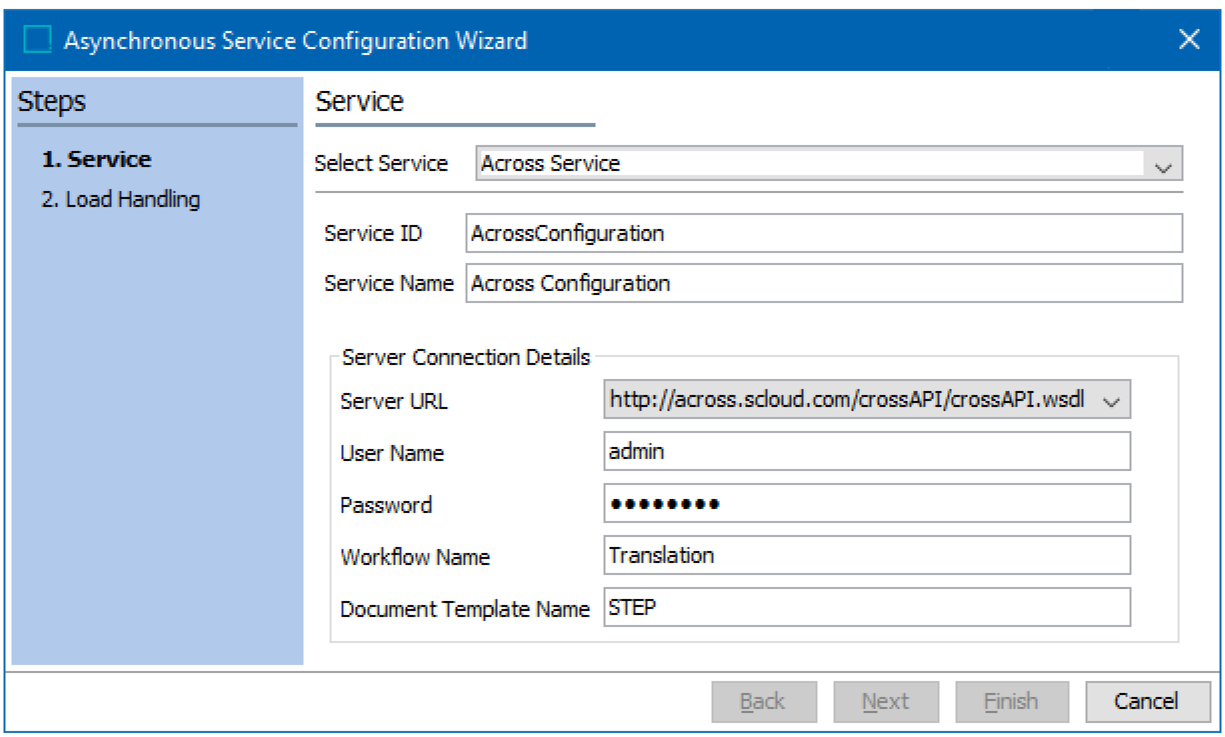
Configuration

1. In System Setup, right-click the Asynchronous Services root node and select **New Asynchronous Service Configuration**.



2. In the **Asynchronous Service Configuration Wizard**, complete the parameters on the **Service** step based on the required service:

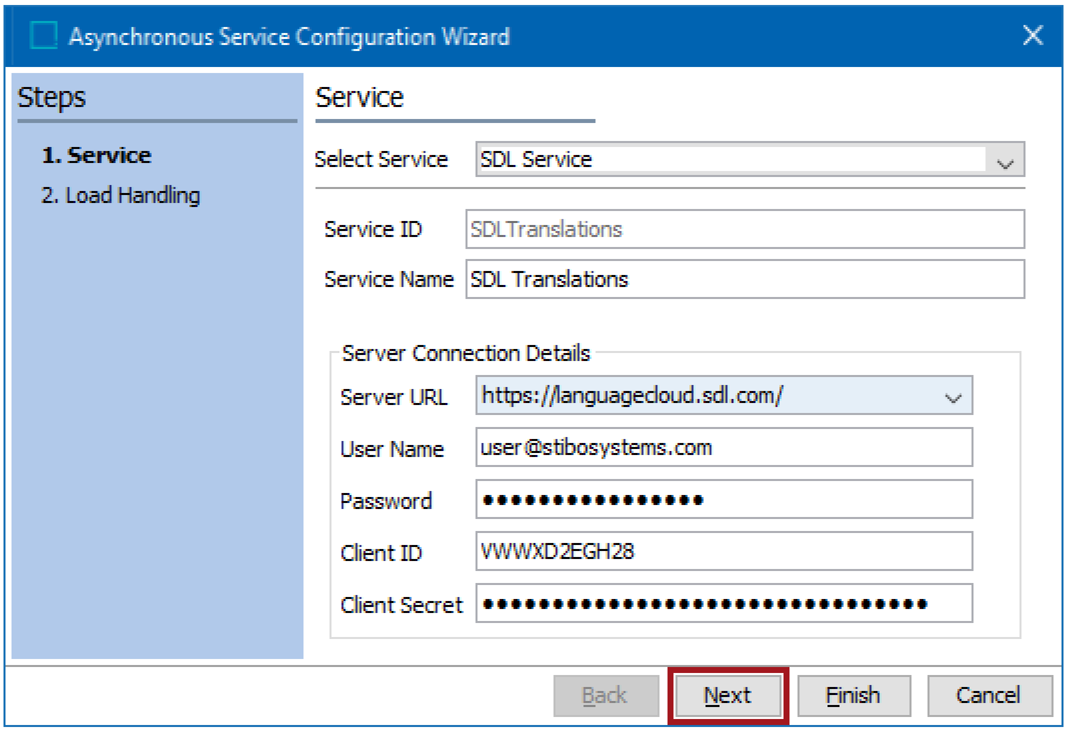
- For a local **Across** translation server, enter your own values for the parameters as shown in the example below.



- **Select Service** - Select **Across Service**
- **Service ID** - Enter the service ID
- **Service Name** - Enter the service name for your Across server

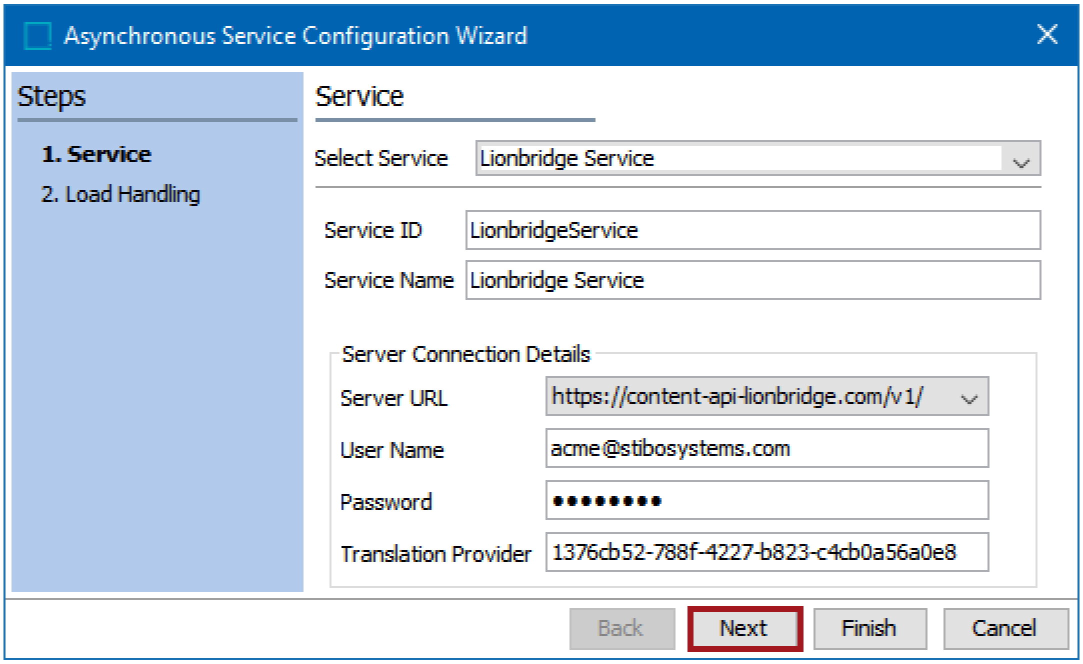
- **Server URL** - Select the Across server URL from the dropdown
- **User Name** - Enter the user name to communicate with Across
- **Password** - Enter the password for Across
- **Workflow Name** - Enter the name of the workflow in Across that will be used for the configuration. In this example the name of the Across workflow is 'Translation.'
- **Document Template Name** - Enter the name of the Across document template you created in the Prerequisites section. In the image above, the Across document template is named 'STEP.'

- For **SDL** translation service, most of the information in the Server Connection Details of the Service step is provided by SDL. Enter your own values for parameters as illustrated in the image below.



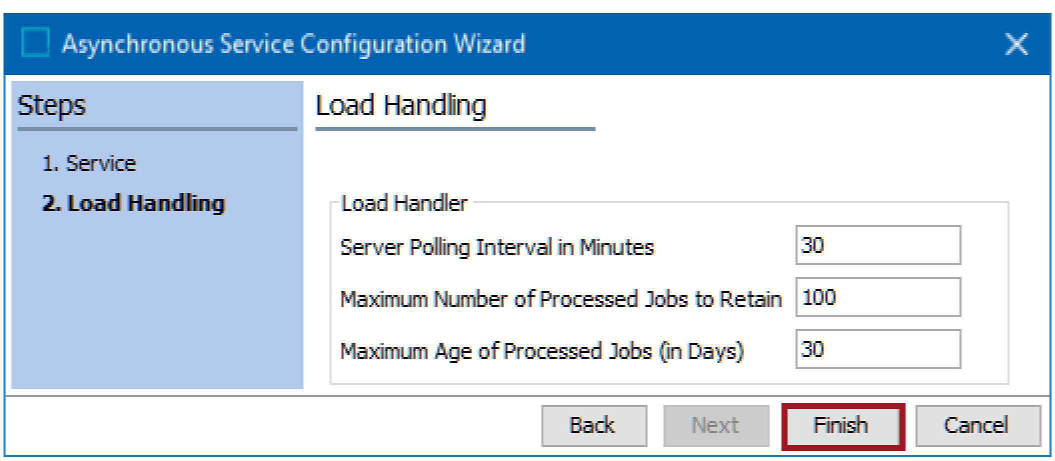
- **Select Service** - Select the **SDL Service** for SDL translations
- **Service ID** - Enter a service ID
- **Service Name** - Enter the service name for your SDL service
- **Server URL** - Select the server URL for SDL from the dropdown
- **User Name** - Enter the user name that will be used to communicate with SDL
- **Password** - Enter the password provided by SDL
- **Client ID** - Enter the client ID assigned to your application from SDL
- **Client Secret** - Enter the client secret assigned to your application from SDL

- For **Lionbridge** translation service, most of the information in the Server Connection Details of the Service step is provided by Lionbridge. Enter your own values for parameters as demonstrated in the image below.



- **Select Service** - Select the **Lionbridge Service** for Lionbridge translations
- **Service ID** - Enter a service ID
- **Service Name** - Enter the service name for your Lionbridge service
- **Server URL** - Select the server URL for Lionbridge from the dropdown
- **User Name** - Enter the user name that will be used to communicate with Lionbridge
- **Password** - Enter the password provided by Lionbridge
- **Translation Provider** - Enter the translation provider key provided by Lionbridge

3. Click **Next** to move on to the Load Handling step.
4. For **Load Handling**, provide the following data:



- **Server Polling Interval in Minutes -** The length of time, in minutes, the asynchronous service will poll the asynchronous translation service. The minimum is one (1) minute.
- **Maximum Number of Processed Jobs to Retain -** The maximum number of processed translation jobs to retain.
- **Maximum Age of Processed Jobs (in Days) -** The maximum number of days to retain a processed translation job.

> **Note:** The processed jobs are retained until either the maximum number of processed jobs or the maximum age of processed jobs limit has been met, whichever comes first.

5. Continue with the setup as defined in **Setting Up a Translation Configuration** topic.

Configuring an Asynchronous File Exchange Service

The File Exchange Service is an asynchronous translation add-on component that allows users that do not have a contract with any of the translation services that STEP supports, such as Lionbridge and SDL, a way to export and/or import translation files to folders instead of using a translation service API.

Note: In order to access the File Exchange Service, the 'file-exchange-translation' add-on component must be activated on your system. See your Stibo Systems representative for more information.

When initiated, the File Exchange Service exports translation files into an out folder. The user manually retrieves these files, makes desired changes to the text that needs translation, and places these files into the in folder. The updated files are then imported back into STEP during the asynchronous poller process.

For general information regarding Asynchronous Services, see the **Asynchronous Translations** topic.

For information regarding configuring other asynchronous translation services, see the **Configuring an Asynchronous Translation Service** topic.

For information regarding the use of the File Exchange Service component within the Web UI, see the **Translation Status Widget Configuration** topic.

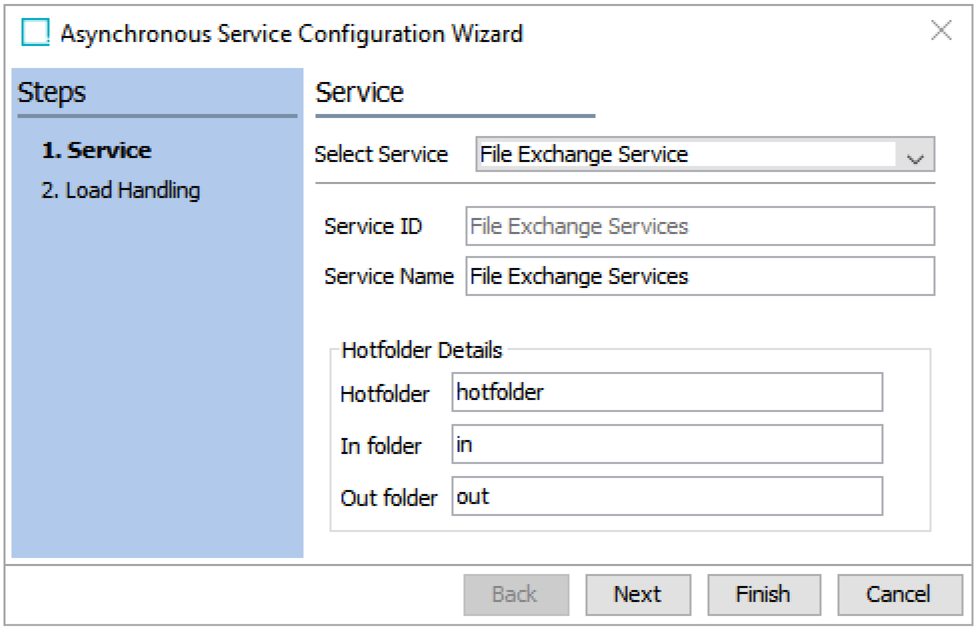
Configuring the File Exchange Service

The File Exchange Service must be set on an Asynchronous Service Configuration.

Important: Before an asynchronous service can be set up and configured, an Asynchronous Services setup group root node and object type must be created. For more information about creating Setup Groups in general, see the **Setup Groups** section of the **System Setup / Super User Guide**.

Use the following steps to configure the File Exchange Service:

1. With the workbench open and the System Setup tab selected, right-click on the Asynchronous Services root node, and select 'New Asynchronous Service Configuration.' The Asynchronous Service Configuration Wizard opens.
2. On the Service step:

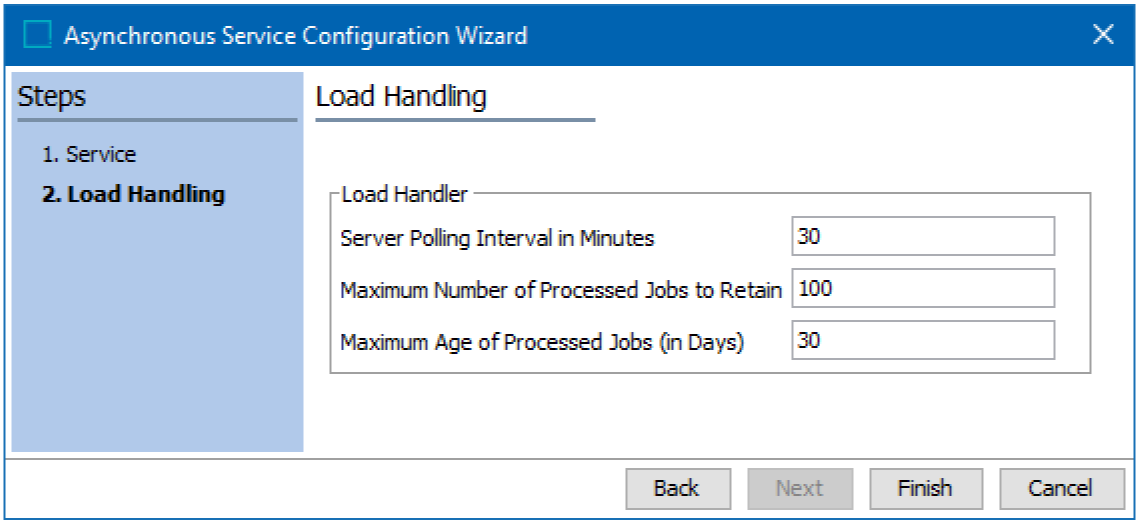


- For **Select Service**, select 'File Exchange Service' from the dropdown menu.
- For **Service ID** and **Service Name**, enter a name into the text fields.
- **Hotfolder -** enter the name of the parent folder that houses both the In and the Out folders. Use a descriptive name, like the name of the translation file exchange service.
- **In folder -** enter the name of the folder to hold completed translation file exchange service files. The translation file exchange service places files in this folder and STEP retrieves files from this folder to resume processing.

> **Note:** Inbound hotfolders must be unique for each ASC. The ASC Poller monitors the inbound folder defined by the ASC and cannot resolve return files across multiple ASCs.

- **Out folder -** enter the name of the folder to hold files pending translation files. STEP places files in this folder and the translation file exchange service retrieves files from this folder.
- Click the **Next** button.

3. On the Load Handling step:



For information on a parameter, hover over the parameter field to display help text.

- For **Server Polling Interval in Minutes**, update from the default of 30 minutes if needed. This is the time between attempts to poll the 'In folder' for an external file that has been processed. The minimum is 1 minute.
- For **Maximum Number of Processed Jobs to Retain**, update from the default of 100 if needed. This is the number of BGPs to be retained. When the maximum is reached, the oldest job is deleted automatically.
- For **Maximum Age of Processed Jobs (in Days)**, update from the default of 30 if needed. A BGP is automatically deleted when its age reaches the maximum number of days.

Note: The processed jobs are retained until either the maximum number of processed jobs or the maximum age of processed jobs limit has been met, whichever comes first.

- Click the **Finish** button.

4. Continue with the setup as defined in **Setting Up a Translation Configuration** topic.

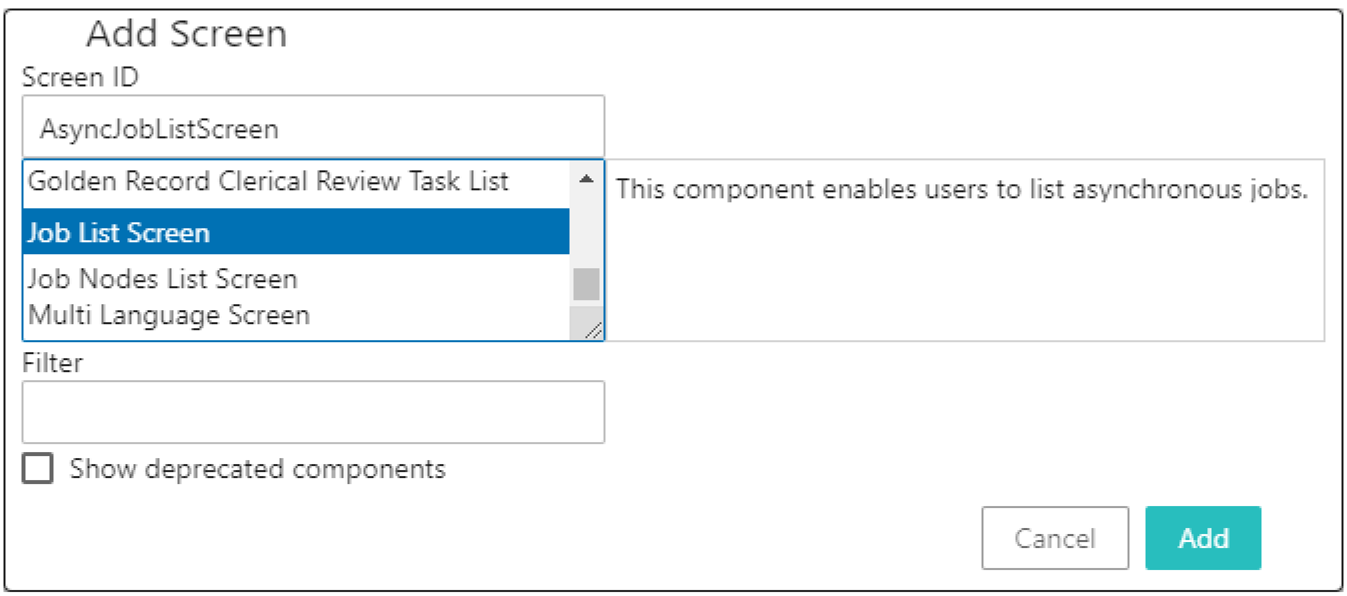
Configuring Screens for Asynchronous Translation Status in Web UI

Three separate screens must be configured prior to using the Translation Status widget to check the status of asynchronous translations. Although the following steps will detail how to configure the screens necessary to view the asynchronous translation statuses of products in the Web UI, it is assumed that users already have a working knowledge in regards to creating screens in the Web UI. For more information, see the **Design Mode Basics** topic in the **Web UI Getting Started** documentation. Additionally, users should familiarize themselves with the **Asynchronous Translations in Web UI** topic , which presents an overview of asynchronous translations in the Web UI.

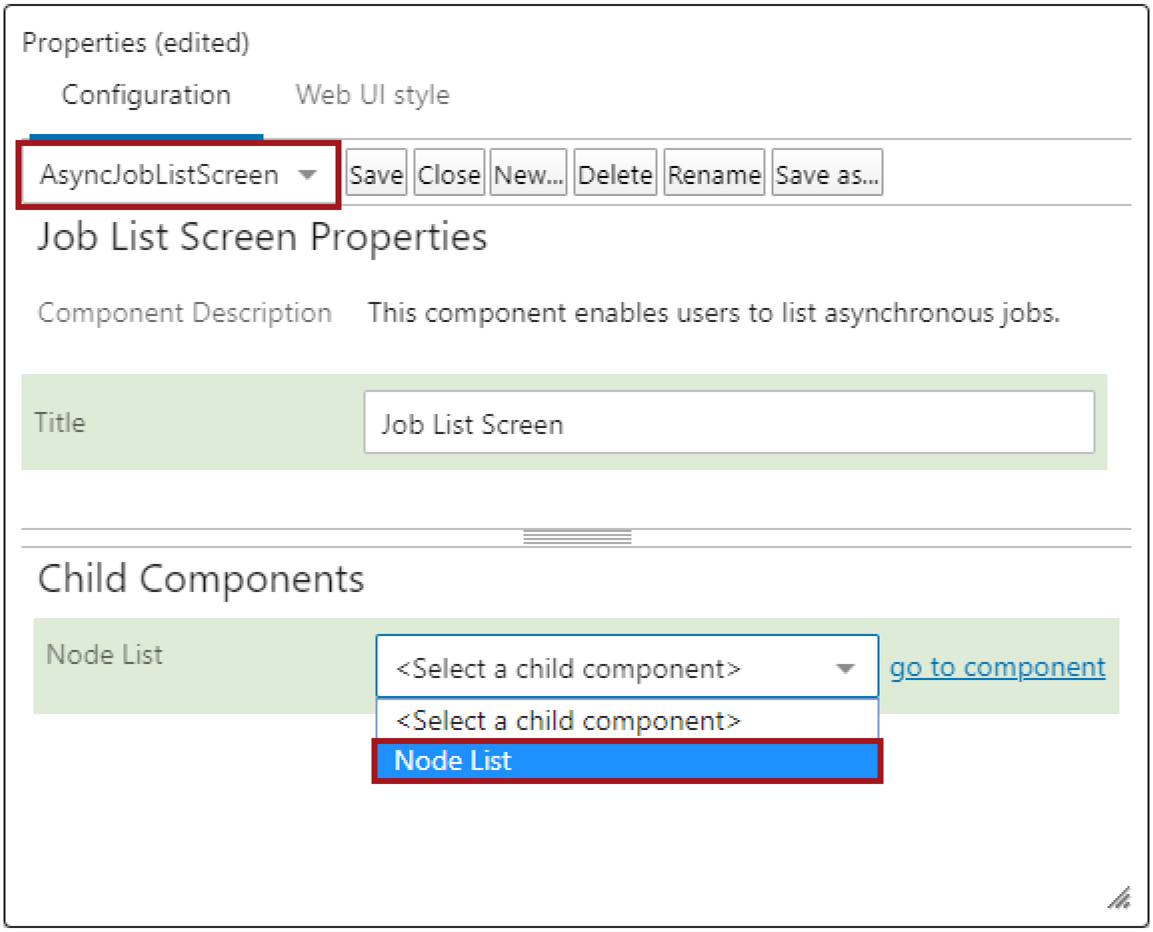
Screen Configuration

Follow the steps below to configure the screens necessary to access asynchronous translation status in Web UI.

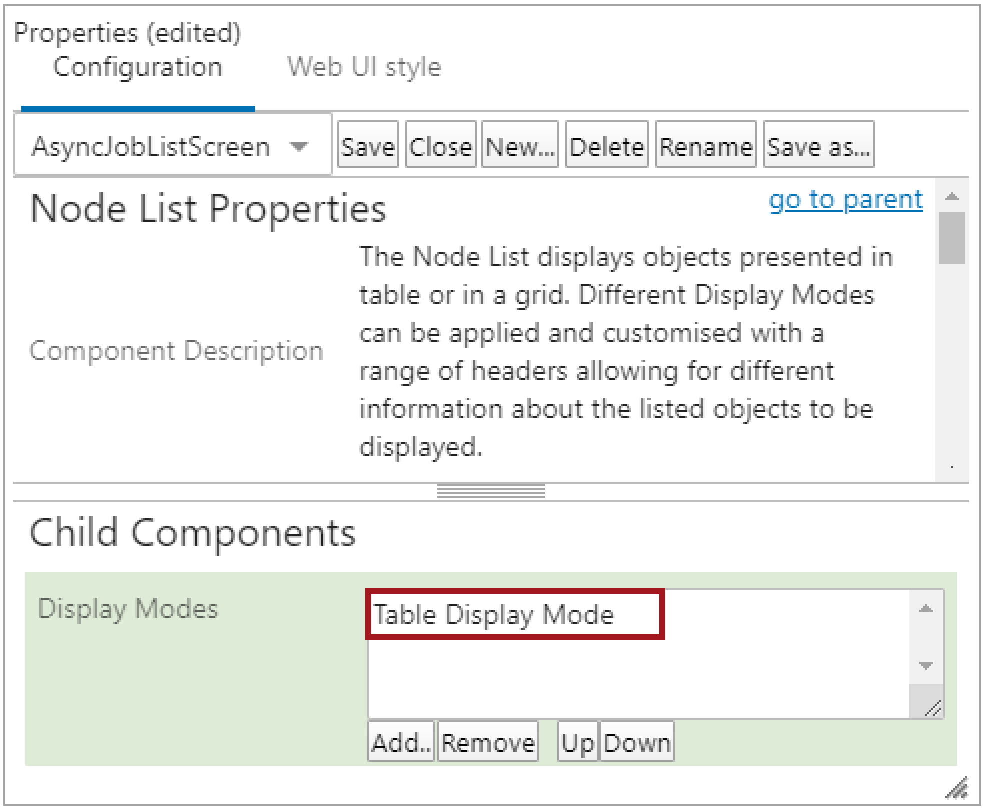
1. In the ' Add Screen' window, select ' Job List Screen' from the list of available screen options and enter a name for the screen in the Screen ID text field. In this example, the user has titled the screen 'AsyncJobListScreen.'



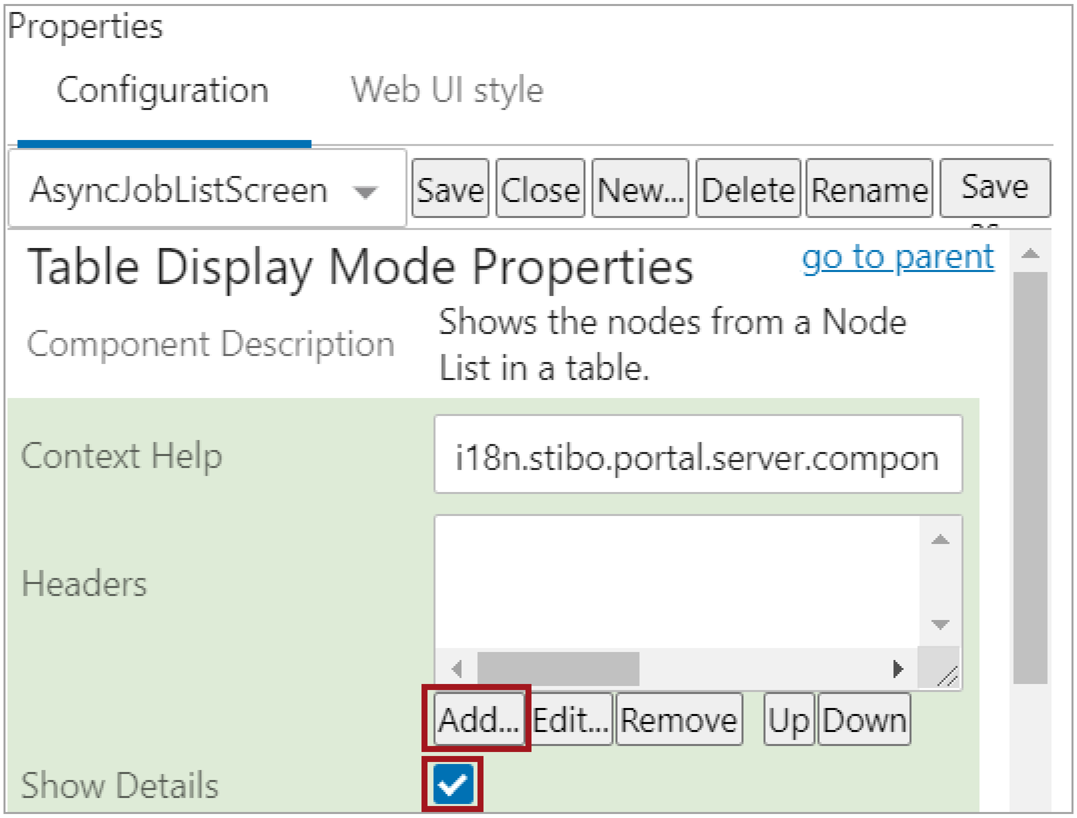
2. Click 'Add.'
3. In the designer, with the newly created screen 'AsyncJobListScreen' selected, select 'Node List' from the Node List dropdown.



4. Click 'go to component.'

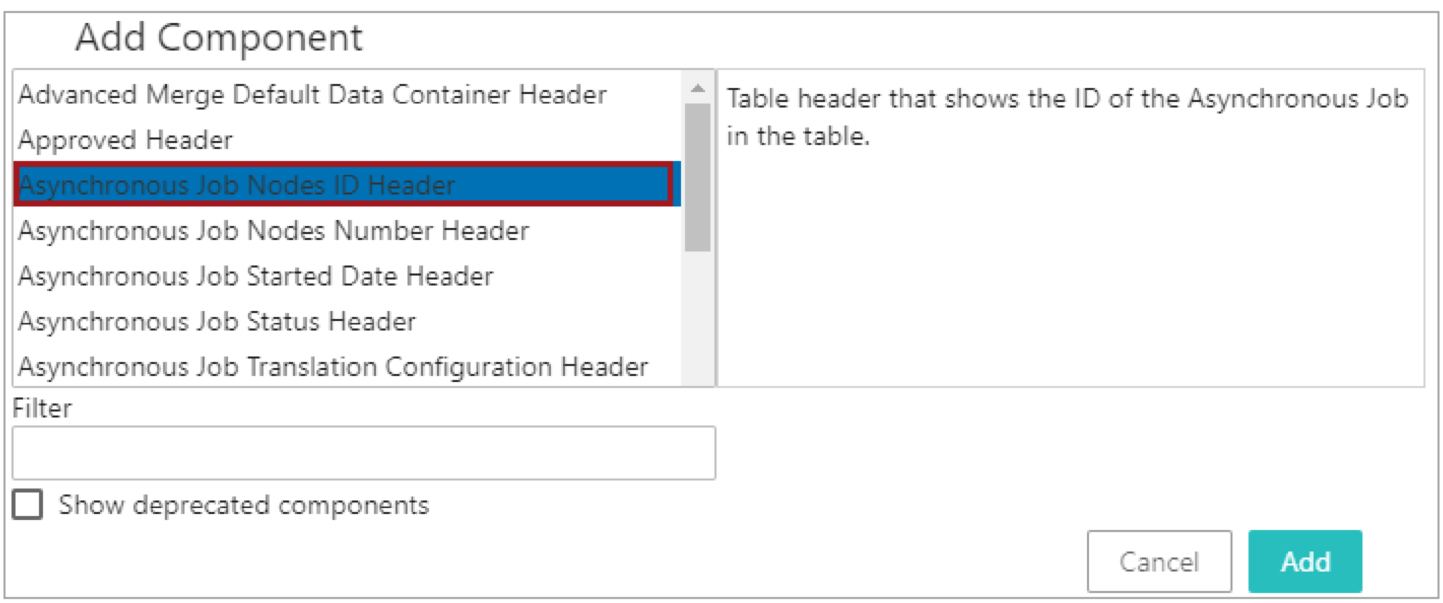


5. Double click 'Table Display Mode.'
6. In the Table Display Mode Properties dialog, check the 'Show Details' checkbox and click 'Add....'.



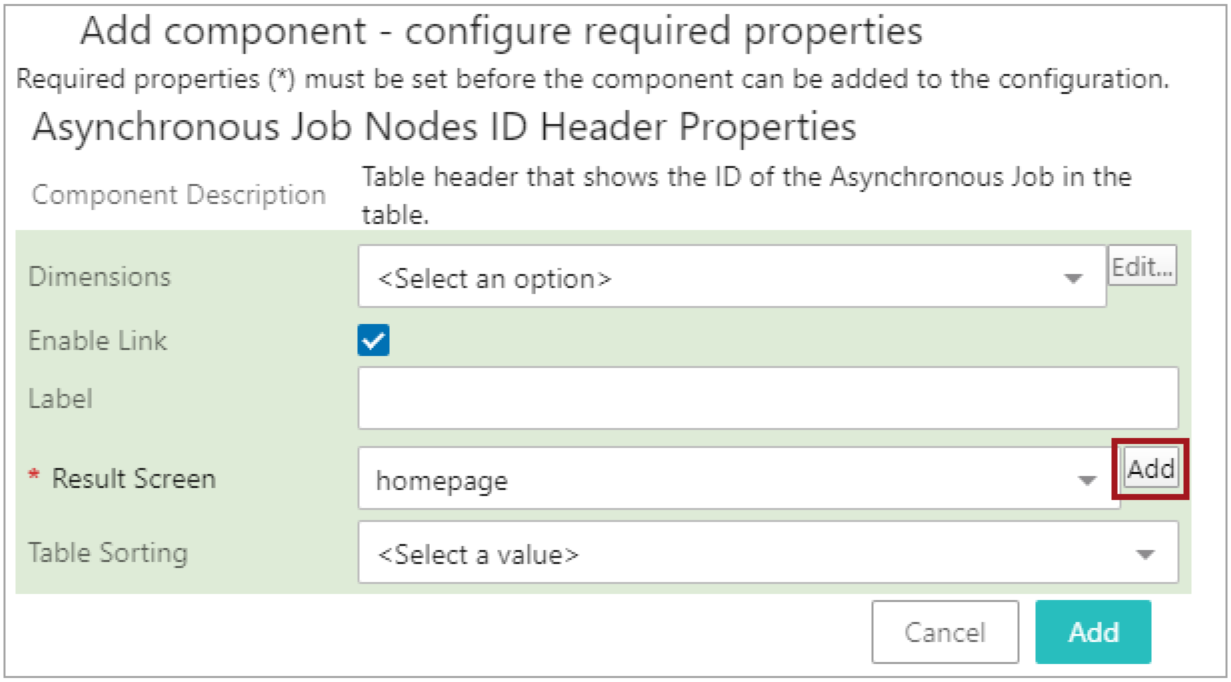
The Add Component dialog contains the table headers that will be used to identify in detail the data regarding translation statuses in the job list screen (in this example, titled 'AsyncJobListScreen'). Each pertinent header starts with the words 'Asynchronous Job.' Information about each header can be obtained by clicking on the header itself, as shown in the example below.

7. In the Add Component dialog, select the 'Asynchronous Job Nodes ID Header' component.

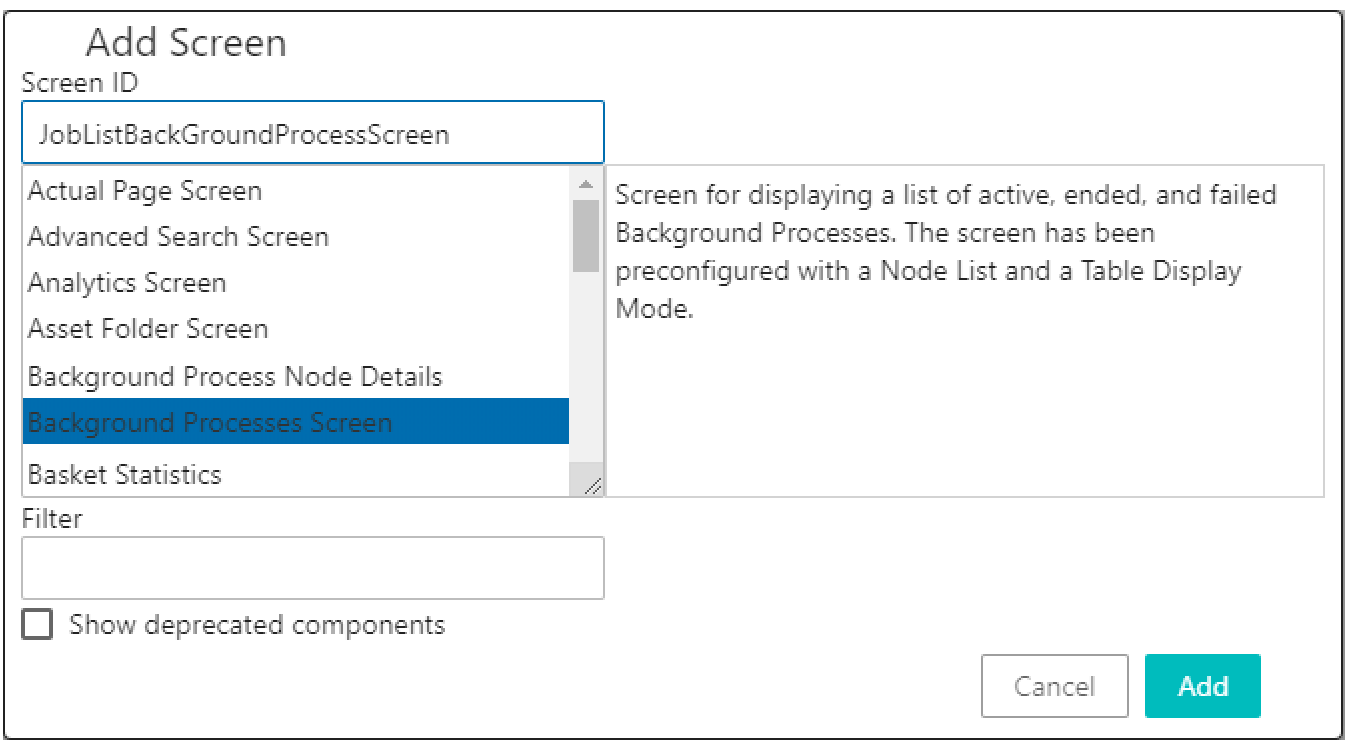


8. Click 'Add.'

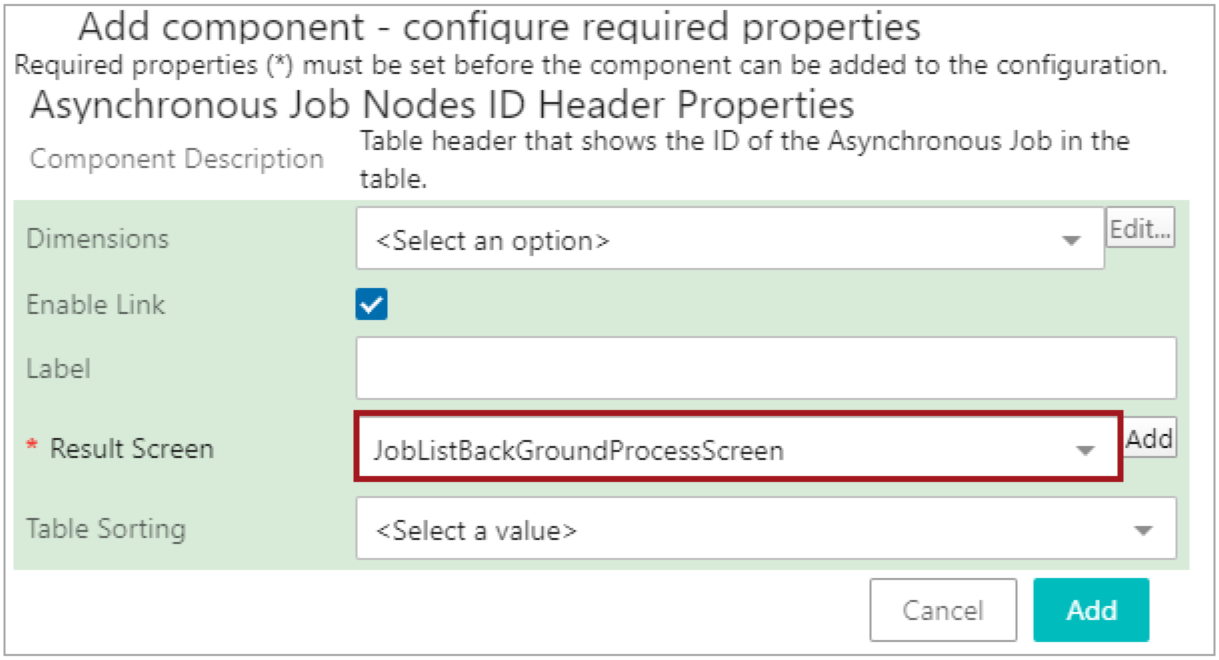
   The Asynchronous Job Nodes ID Header Properties dialog opens.



9. Select 'Add' next to the 'Result Screen' dropdown menu.
10. In the Add Screen dialog, select 'Background Processes Screen' and create a Screen ID. In this example, the user has created the Screen ID 'JobListBackGroundProcessScreen.'



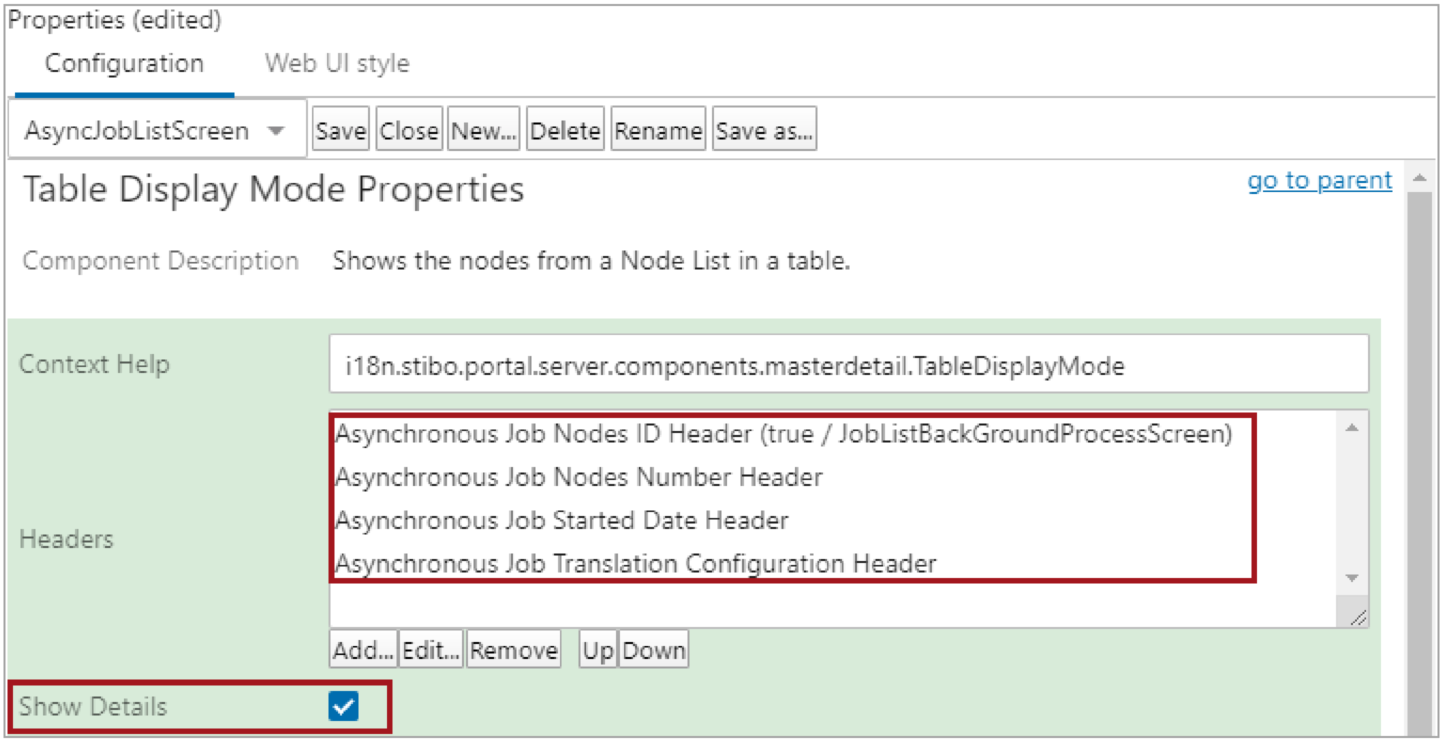
11. Click 'Add.'
12. Select the newly created result screen 'JobListBackgroundProcessScreen' from the Result Screen dropdown menu.



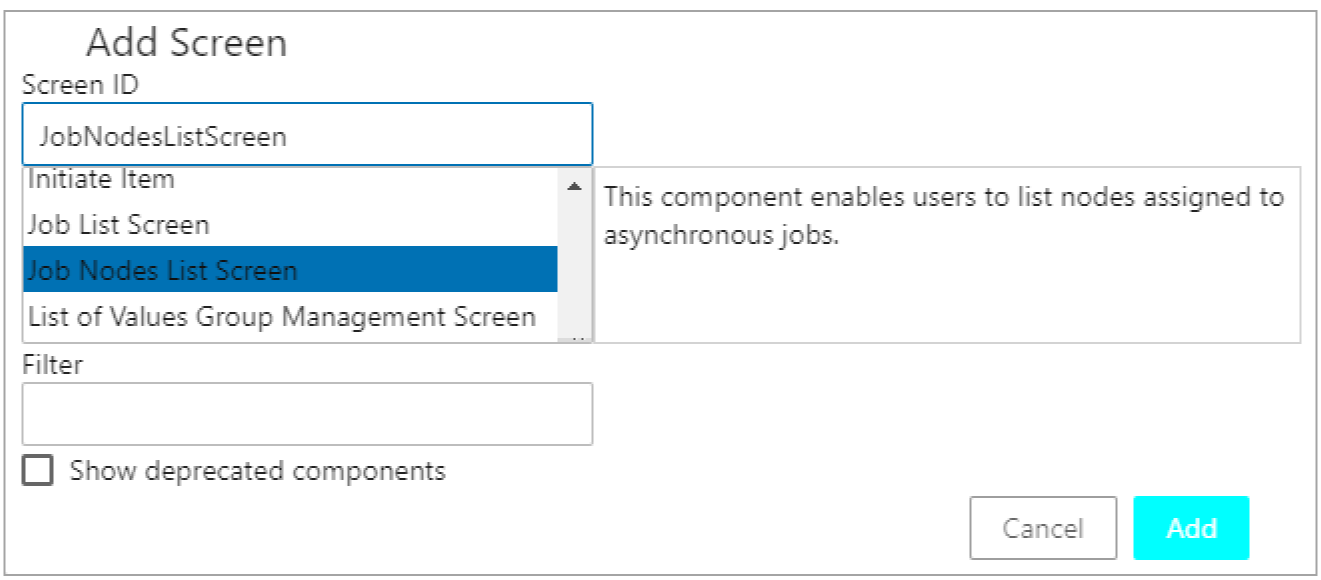
13. Click 'Add.'
14. In the Table Display Mode Properties, dialog window, click 'Add... ' to add more Asynchronous Job headers.

Note: Header components must be added one at a time. After a desired component header is selected, click 'Add.'

The example below shows the four headers that the user has selected.



15. Click 'Save'.
16. In the designer, click 'New...' to add a new screen.
17. In the Add Screen dialog, select 'Job Nodes List Screen' and create a Screen ID. In this example, the user has created the Screen ID 'JobNodesListScreen.'



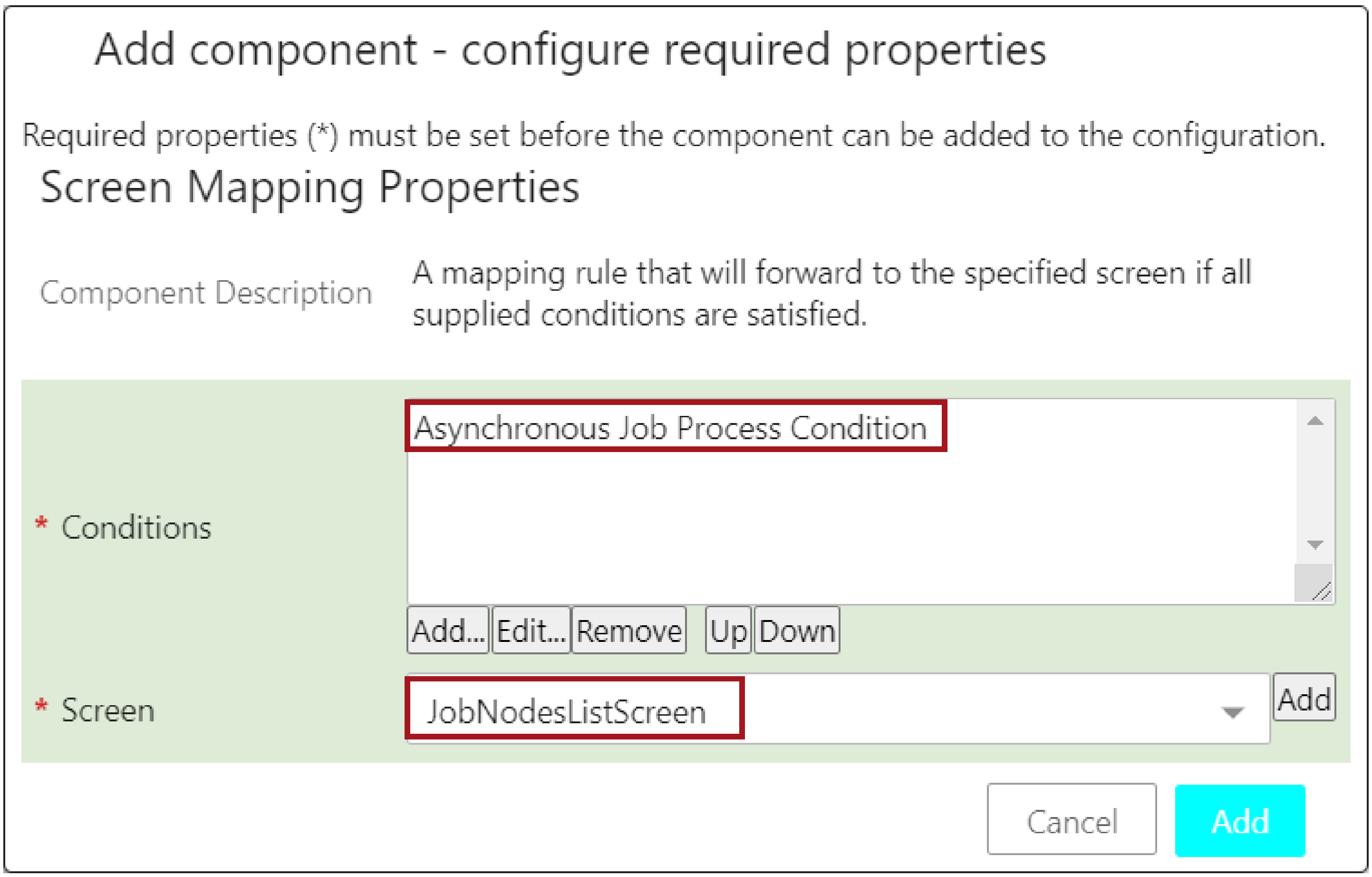
18. Click 'Add' to close the dialog.
19. In the designer, click 'Save.'

Screen Mapping

Now that the three screens have been configured, the Job Nodes List screen (in this example, titled 'JobNodesListScreen') needs to be mapped.

> **Note:** Although the following steps will detail how to map any screens necessary to view the asynchronous translation statuses of products in the Web UI, it is highly recommended that users already have a working knowledge of mapping within Web UI. For more information, see the **Mapping Workflow States in Web UI** topic in the **Using Web UI** documentation.

1. In the Screen Mapping Properties dialog, select 'Asynchronous Job Process Condition' as the condition and 'JobNodesListScreen' as the screen.



2. Click 'Add' to close the dialog.

   Mapping is complete.

3. Click 'Save' and 'Close' to close the designer.

Now that the screens are configured and mapped, the Translation Status widget needs to be configured. Details regarding that process can be found in the **Translation Status Widget Configuration** topic.

Setting Up a Translation Configuration

Configuration of an asynchronous service can be used to communicate translation data outside of STEP.

Prerequisites

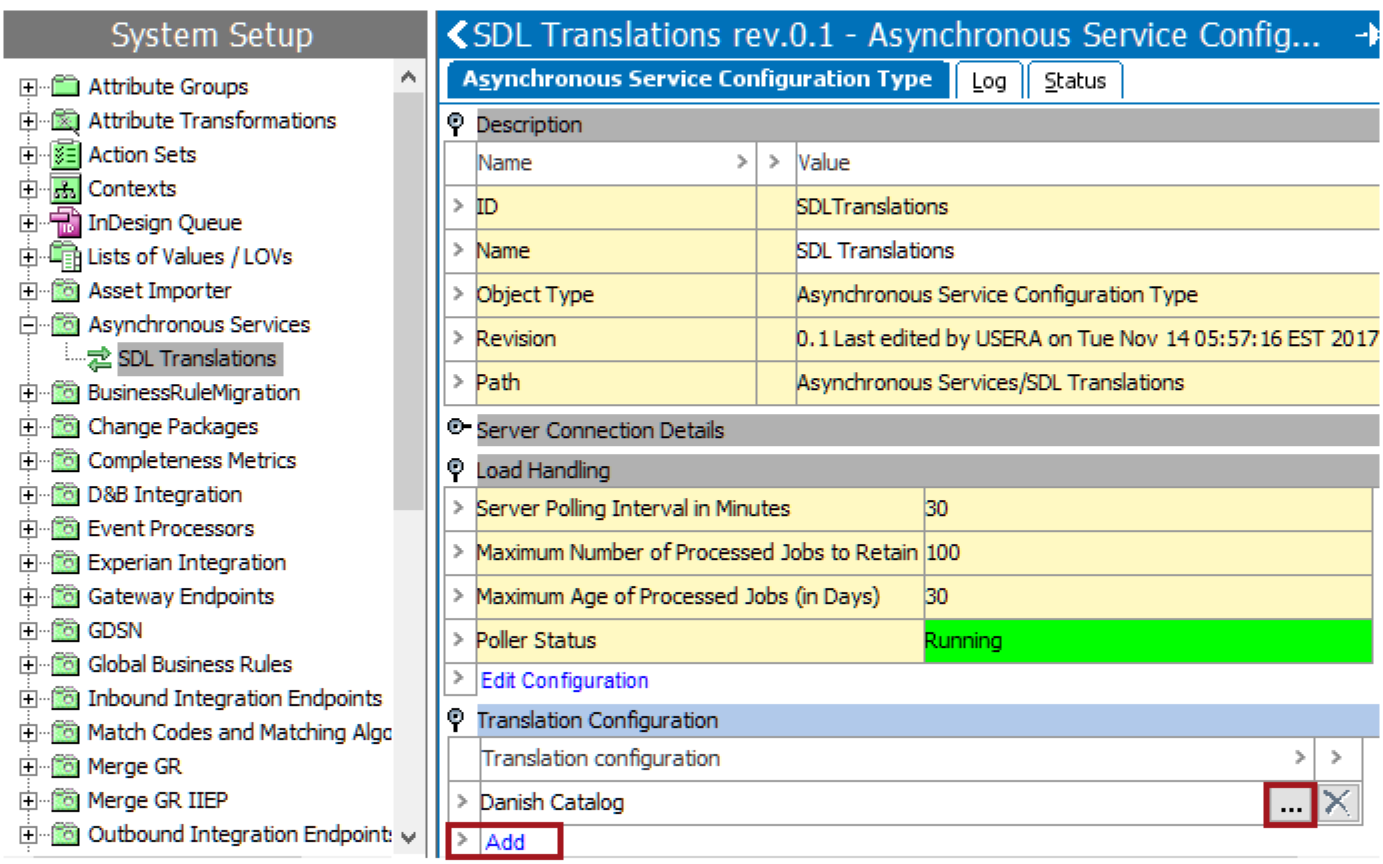
1. Create an asynchronous service object type as described in the **Create the Asynchronous Services Object Type** topic.
2. Run the Asynchronous Service Configuration Wizard as described in the **Configuring an Asynchronous Translation Service** topic.

   For information about asynchronous file exchange services, see the **Configuring an Asynchronous File Exchange Service** topic.

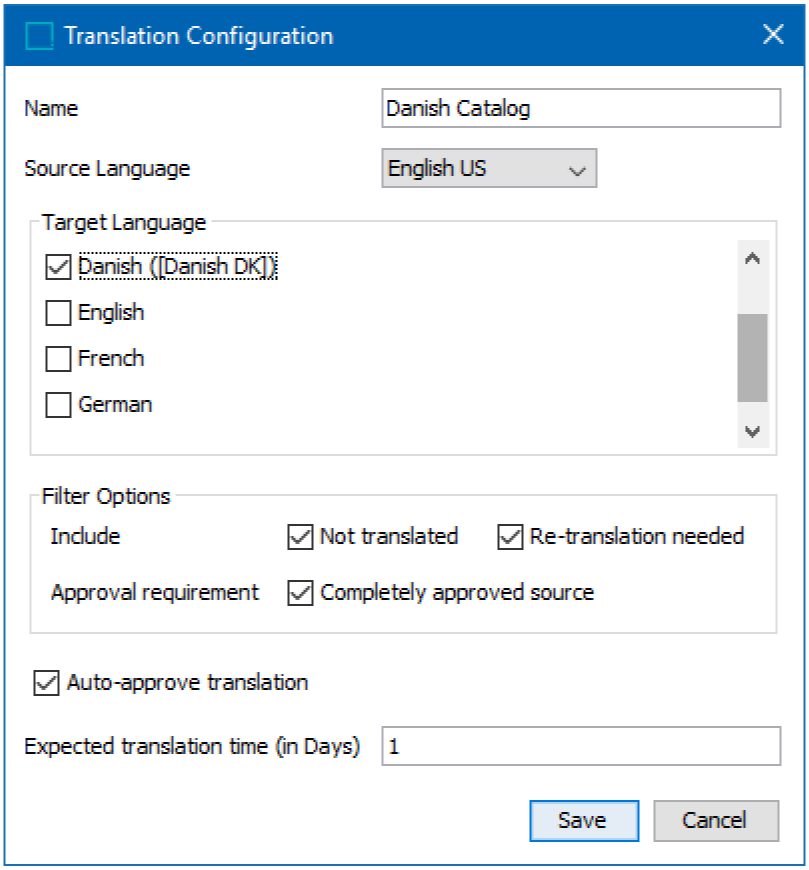
Set Up a Translation Configuration

After the wizard has been run, the **Translation Configuration** needs to be set up as described below.

1. In System Setup, select the asynchronous service configuration (named 'SDL Translations' in the image below), and on the first tab open the Translation Configuration flipper.



2. Click the 'Add' link for a new configuration, or click the ellipsis button (...) for an existing configuration to display the Translation Configuration dialog.



> **Important:** If the expected source and/or target translations are not displaying in the Translation Configuration dialog, see below for a troubleshooting guide.

3. Set the options in the dialog as follows:

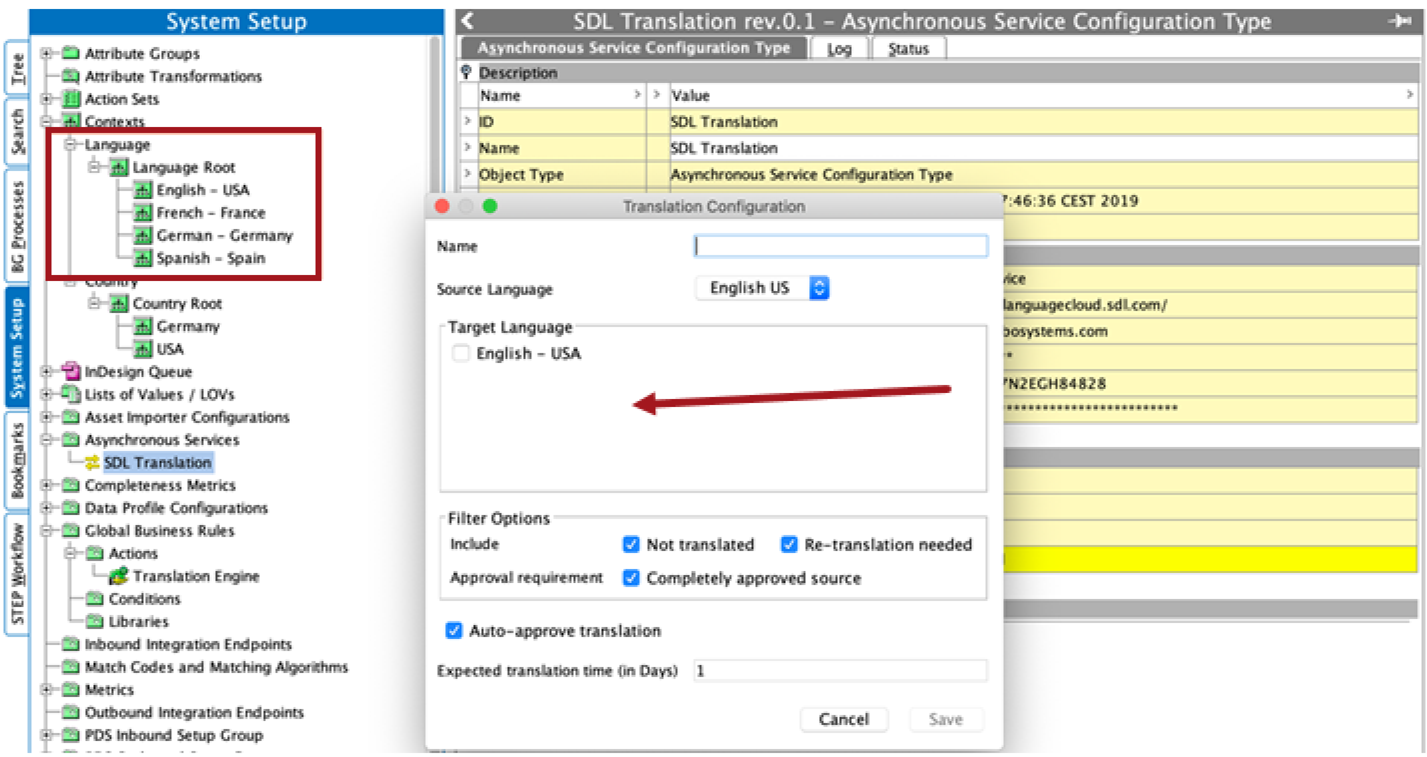
- **Name** - Enter the name of the Translation Configuration.
- **Source Language** - Select the context from the dropdown to indicate the source language.
- **Target Language** - Check the language(s) that the content should be translated into.
- **Filter Options: Include** - select at least one option to enable the **Save** button:
- 'Not translated' includes untranslated values in the filter
- 'Re-translation needed' includes values that have been amended and are thus designated as needing re-translation
- **Filter Options: Approval Requirement** - Uncheck the 'Completely approved source' box to waive the requirement that all objects must be fully approved prior to being translated.

- **Auto-approve translation -** By default, this option is checked and inbound translation updates are automatically approved and moved to the Approved workspace. When unchecked, inbound translation updates remain in the Main (maintenance) workspace.
- **Expected translation time (in Days) -** Used to communicate the number of days the translation is expected back from the service. This is usually populated based on the service level agreement, assuming there is one, between the translation service and the partner.

4. Click the **Save** button to complete the translation configuration.
5. Right-click the asynchronous service configuration and select **Start Polling** to activate the process.
6. Continue with the required setup, creating a business rule to trigger the translation as defined in the **Business Rules for Asynchronous Translations** topic.

Troubleshooting Source and Target translation

If the source and/or target translations are not displaying in the Translation Configuration dialog like in the image below, then the **Language** and **Country** fields on the Contexts editor need to be configured.

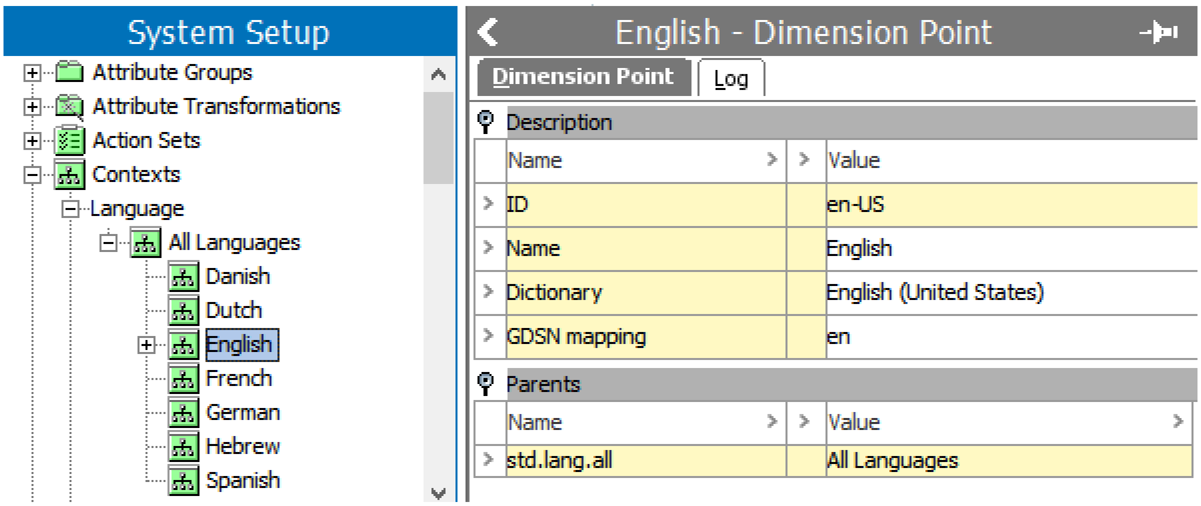


For more information, see the **Maintaining Contexts** topic.

Language Codes for Asynchronous Services

When sending files to be translated, some services require that the ISO-639 language code needs to be sent, followed by the ISO-3166 country code. For example, if a file was being translated from English US to Danish Denmark for SDL Translation, the required language codes would be 'en-US' and 'da-DK.' As shown below, the language dimension point IDs follow the required format, so a transformation lookup table is not required.

Note: When the language and country codes are not in the correct format in the IDs of the language dimension points, a Transformation Lookup Table is required.



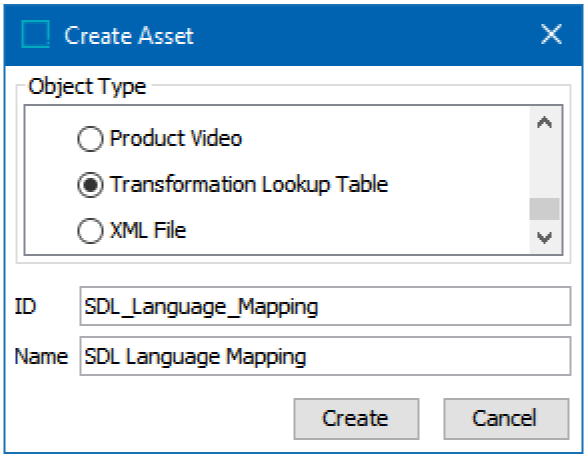
For Across translation, since the Windows language 'LCID' (Language Code Identifier) is required for the language code, a Lookup Table must be used.

Transformation Lookup Table for Language Codes

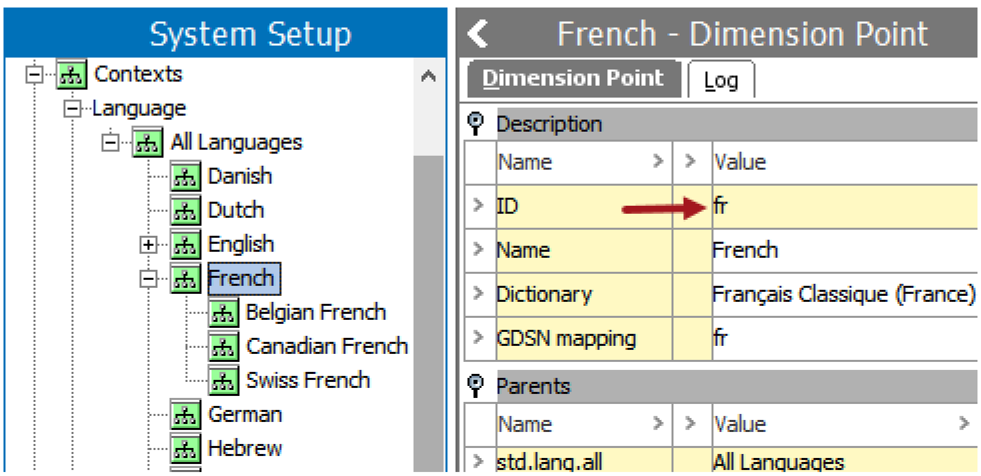
Use the following steps to create a Transformation Lookup Table that will convert the language ID in the language dimension point into ISO language / country qualifiers.

Important: Allowed languages are determined by the translation service or software. Attempting to translate for a language that is not allowed, even when both the target and source languages exist in the lookup table, results in an error in the BGP Execution Report. For example, this error is reported by Across when the target language was not allowed by the software: 'Failed to find target LCID: 1030. Supported target LCIDs are 10249, 10252, 1031, 1033, 1036, 11273, 11276...'.

1. Create a Transformation Lookup Table. For more information, see the **Transformation Lookup Tables** topic in the **Resource Materials** online help.

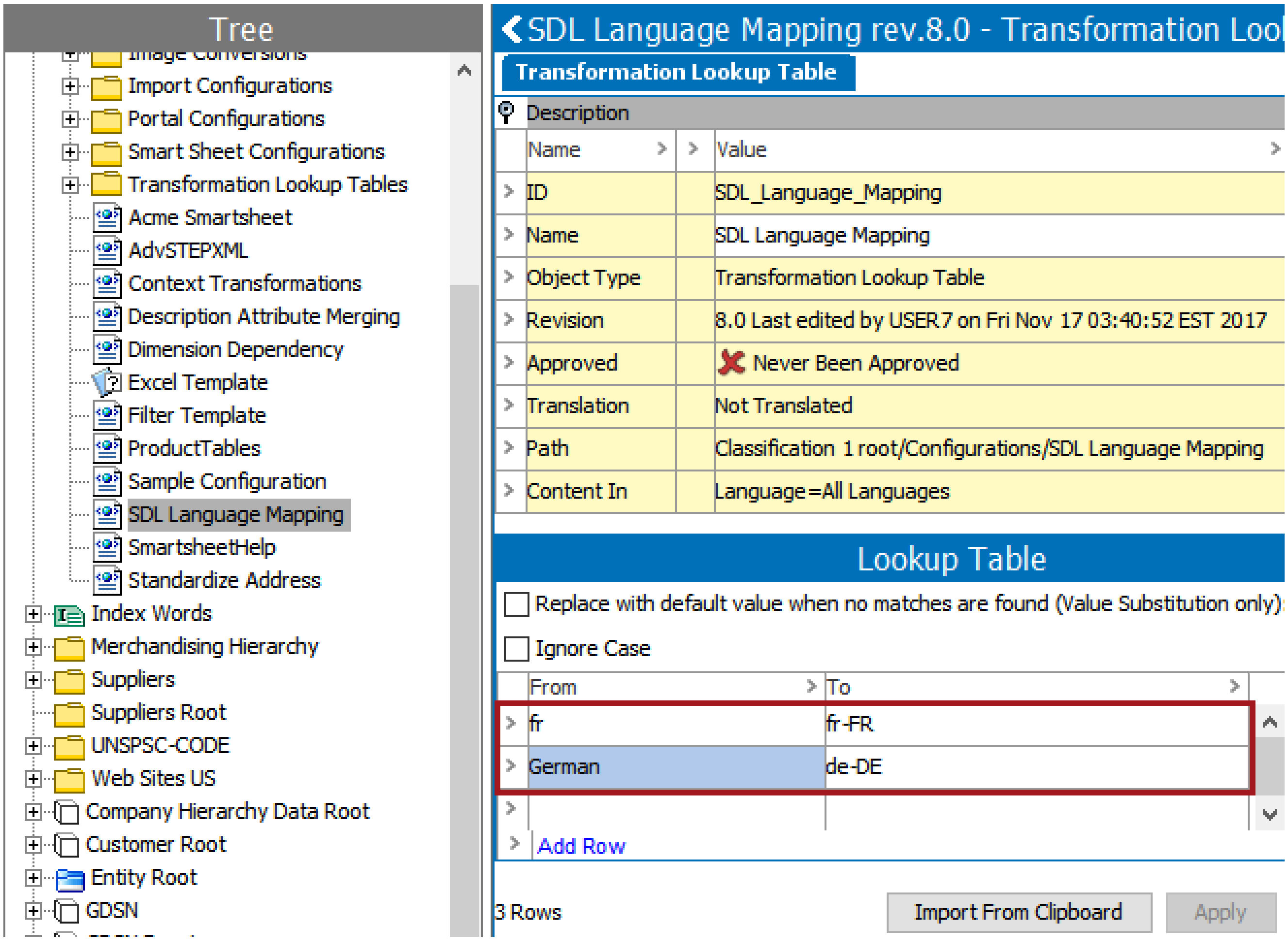


2. Find the dimension IDs for the languages that need qualifiers set up.

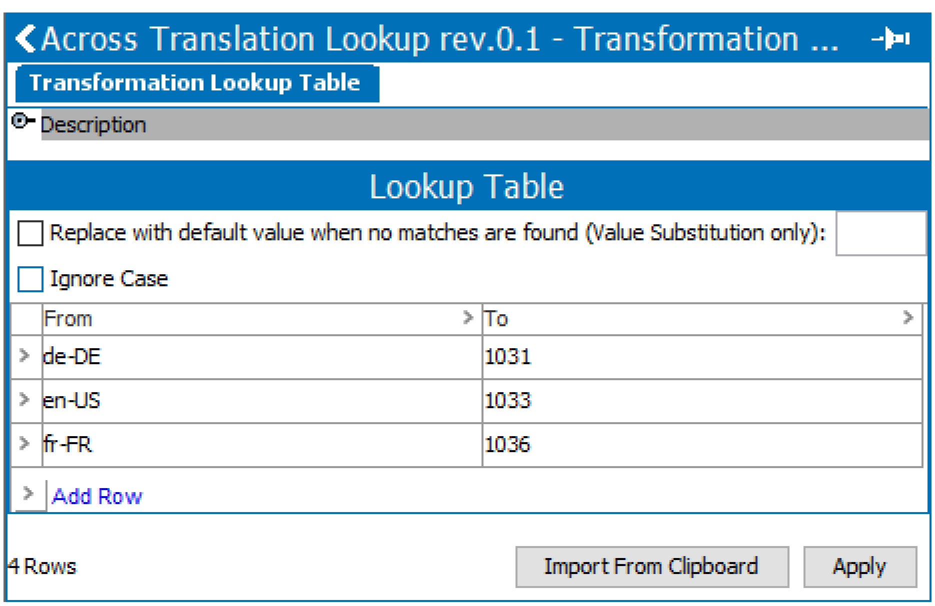


3. In the Transformation Lookup Table, determine the type of transformation needed to convert the language ID in the language dimension point to:

- **ISO language / country qualifiers** - Enter the language IDs in the 'From' column and the corresponding ISO language / country qualifier in the 'To' column as shown below.



- **Windows Language Code Identifier (LCID)** - Enter the language IDs in the 'From' column and the corresponding ISO language / country qualifier in the 'To' column as shown below.



4. Add one or more of the following case-sensitive properties to the sharedconfig.properties file on the STEP application server to identify one or more lookup tables for translations, and stop / start the server to apply the change.

   - **AsyncTranslation.TargetLanguageLookupTable.Across**
   - **AsyncTranslation.TargetLanguageLookupTable.Lionbridge**
   - **AsyncTranslation.TargetLanguageLookupTable.SDL**

   ```
   AsyncTranslation.TargetLanguageLookupTable.Across=AcrossLanguageMapping
   ```

   In this property example, 'AcrossLanguageMapping' is the ID of the lookup table.

   - **AsyncTranslation.TargetLanguageLookupTable** - This original property only allowed a single lookup table, even when multiple translation processes were used. Although it is deprecated, it still functions.

   ```
   AsyncTranslation.TargetLanguageLookupTable=SDL_Language_Mapping
   ```

   In this property example, 'SDL_Language_Mapping' is the ID of the lookup table.

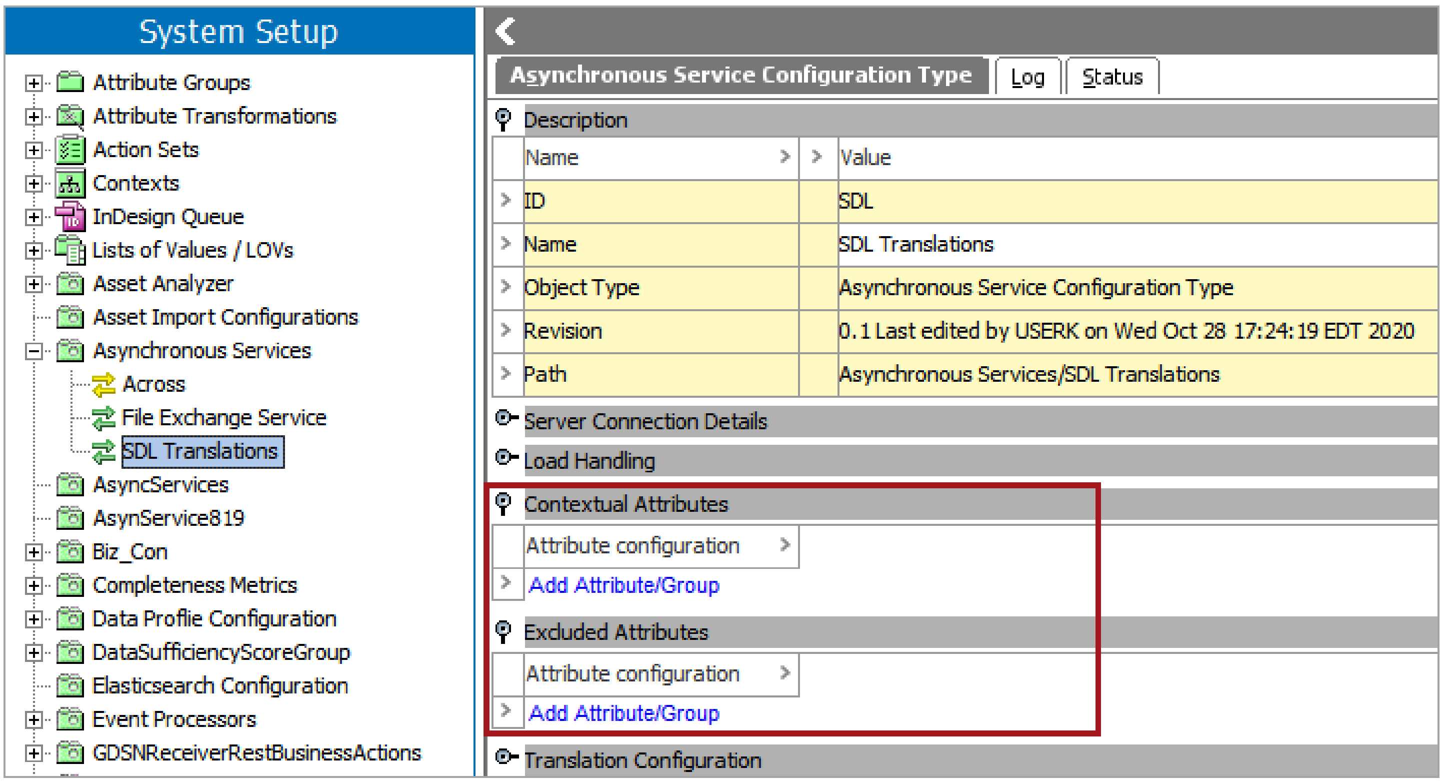
Filtering Attributes for Translation

Within an asynchronous service configuration, users are able to select language-dependent attributes that they would like to be excluded for translation when the translation service is run; users are also able to select contextual attributes that they would like to be included for translation when the translation service is run. For more information, see the **Attribute Filters for Asynchronous Translation Services** topic.

Attribute Filters for Asynchronous Translation Services

The Asynchronous Service Configuration Type object allows users to refine the list of attribute values required to be translated. Configure the following optional parameters as needed:

- 'Contextual Attributes' includes contextual attributes /attribute groups for translation.
- 'Excluded Attributes' excludes language-dependent attributes / attribute groups for translation.



Note: Although calculated attributes can be selected as contextual attributes or translatable attributes when configuring attribute filtering, they will be omitted from the export as it is not possible to export calculated attributes into a translation file.

Prerequisites

Before proceeding with the following steps found in this topic, users must:

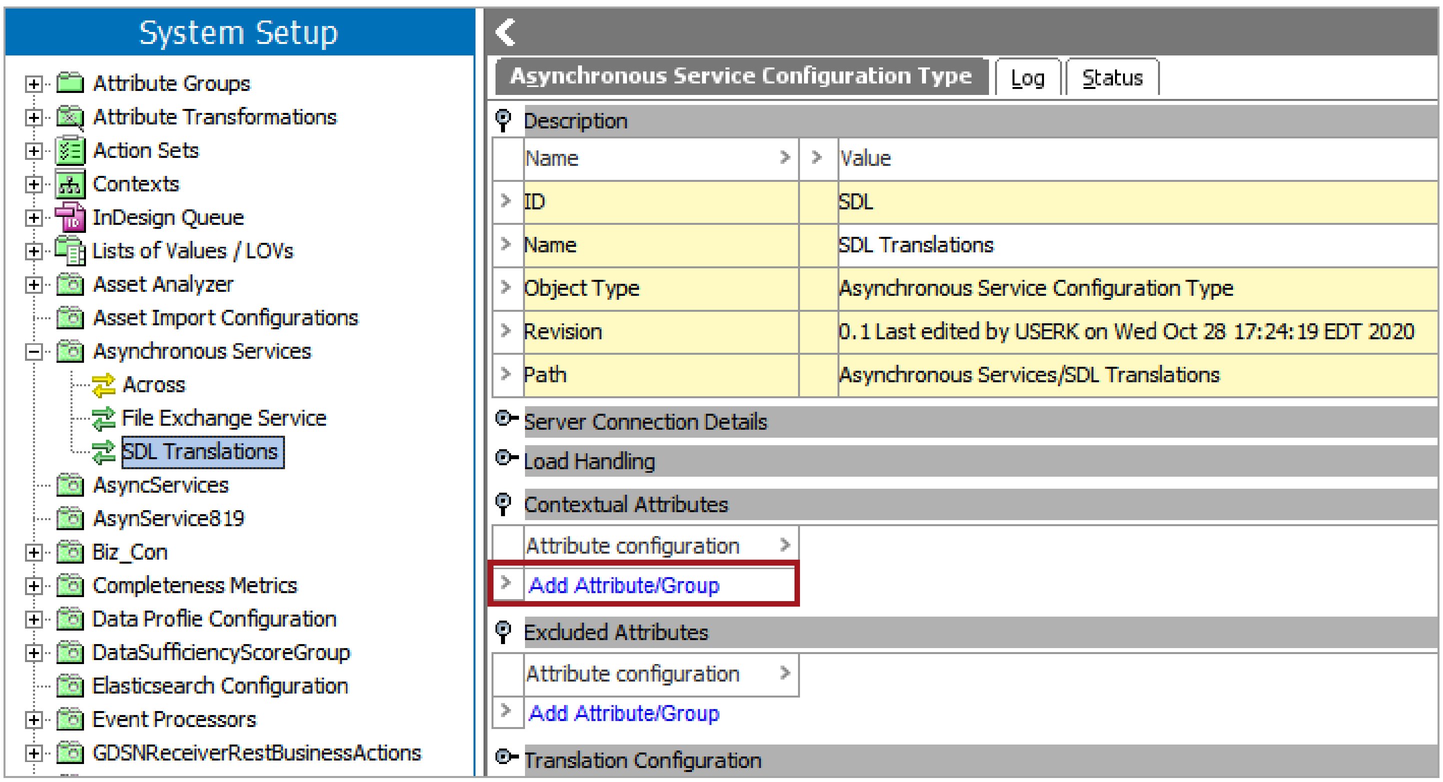
- Create an asynchronous service object type as described in the **Create the Asynchronous Services Object Type** topic.
- Complete the Asynchronous Service Configuration Wizard as described in the **Configuring an Asynchronous Translation Service** topic.

- Set up a Translation Configuration as described in the **Setting Up a Translation Configuration** topic.

Include Contextual Attributes for Asynchronous Translations

Contextual attributes are attributes that although are language-independent, provide the translator with contextual information about the object which contains the language-dependent attributes being translated. Contextual attributes selected are included within an asynchronous translation, while all others are ignored by the translation service.

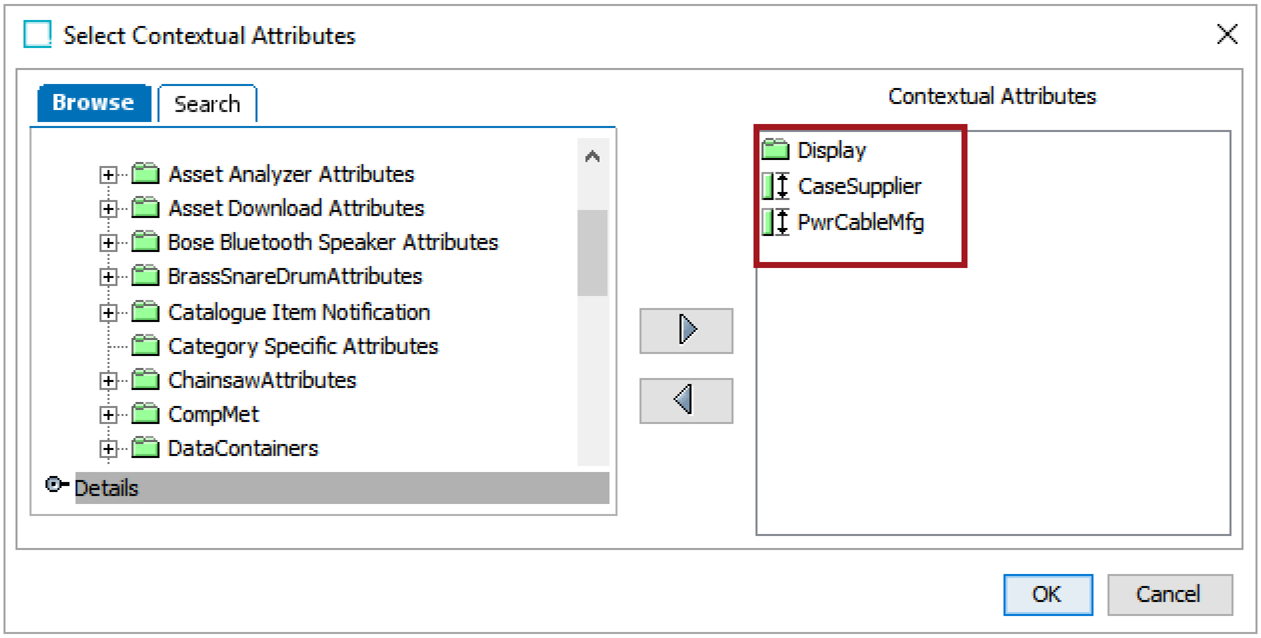
1. Select the desired asynchronous translation service (in this example, 'SDL Translations'), open the 'Contextual Attributes' flipper open, and click the 'Add Attribute/Group' link.



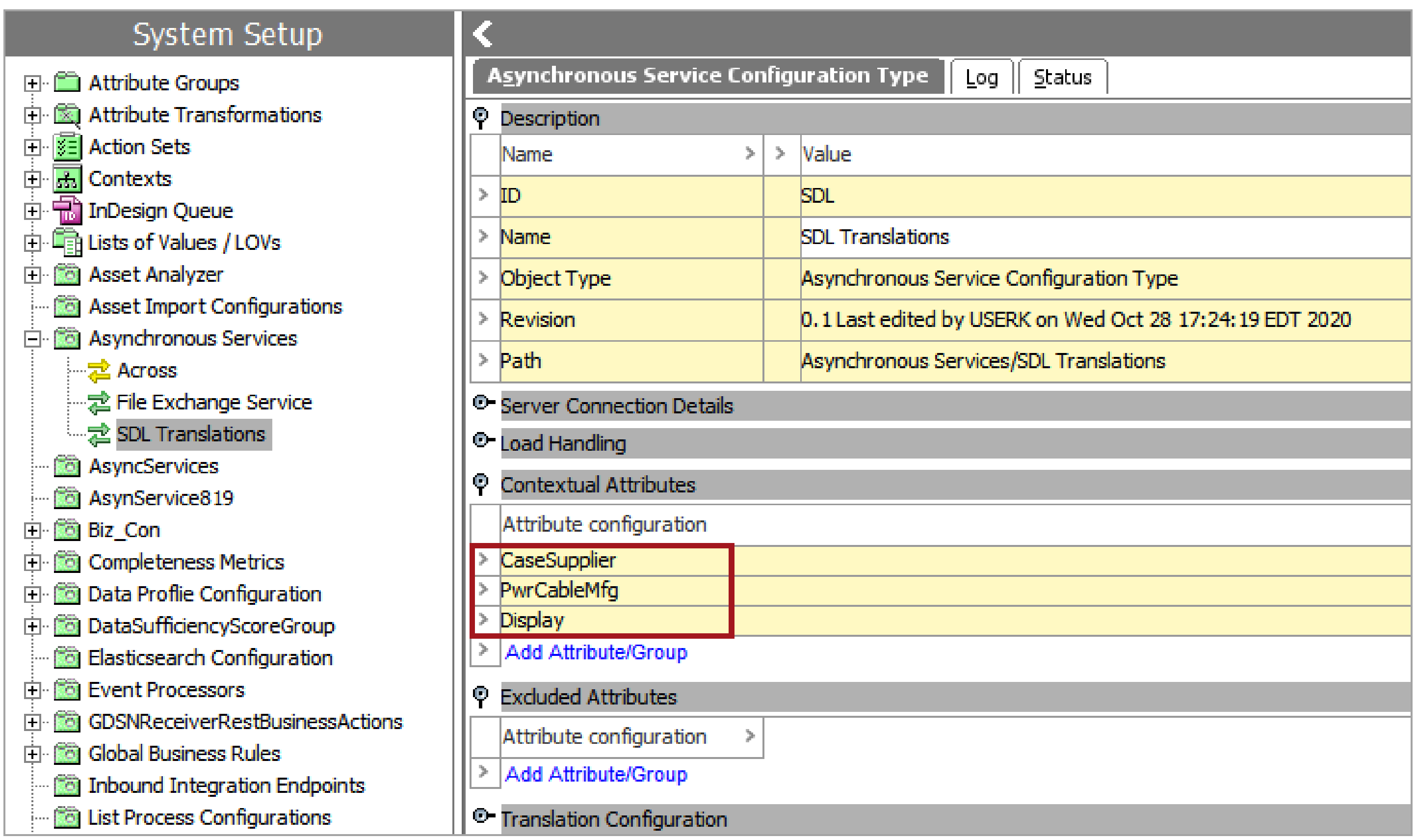
2. In the 'Select Contextual Attributes' dialog, select the desired contextual attributes / attribute groups to include for translation when a translation is run using the selected asynchronous service, and use the right arrow ( ) button to move those selections to the 'Contextual Attributes' column.

   - If desired, use the left arrow ( ) button to remove the selected attributes / attribute groups from the 'Contextual Attributes' column.

   In this example, the user has selected the attribute group 'Display' and the attributes 'CaseSupplier' and 'PwrCableMfg.'



3. Click 'OK' to save the settings and close the 'Select Contextual Attributes' dialog. The included attributes and attribute groups are displayed in the 'Contextual Attribute' flipper.



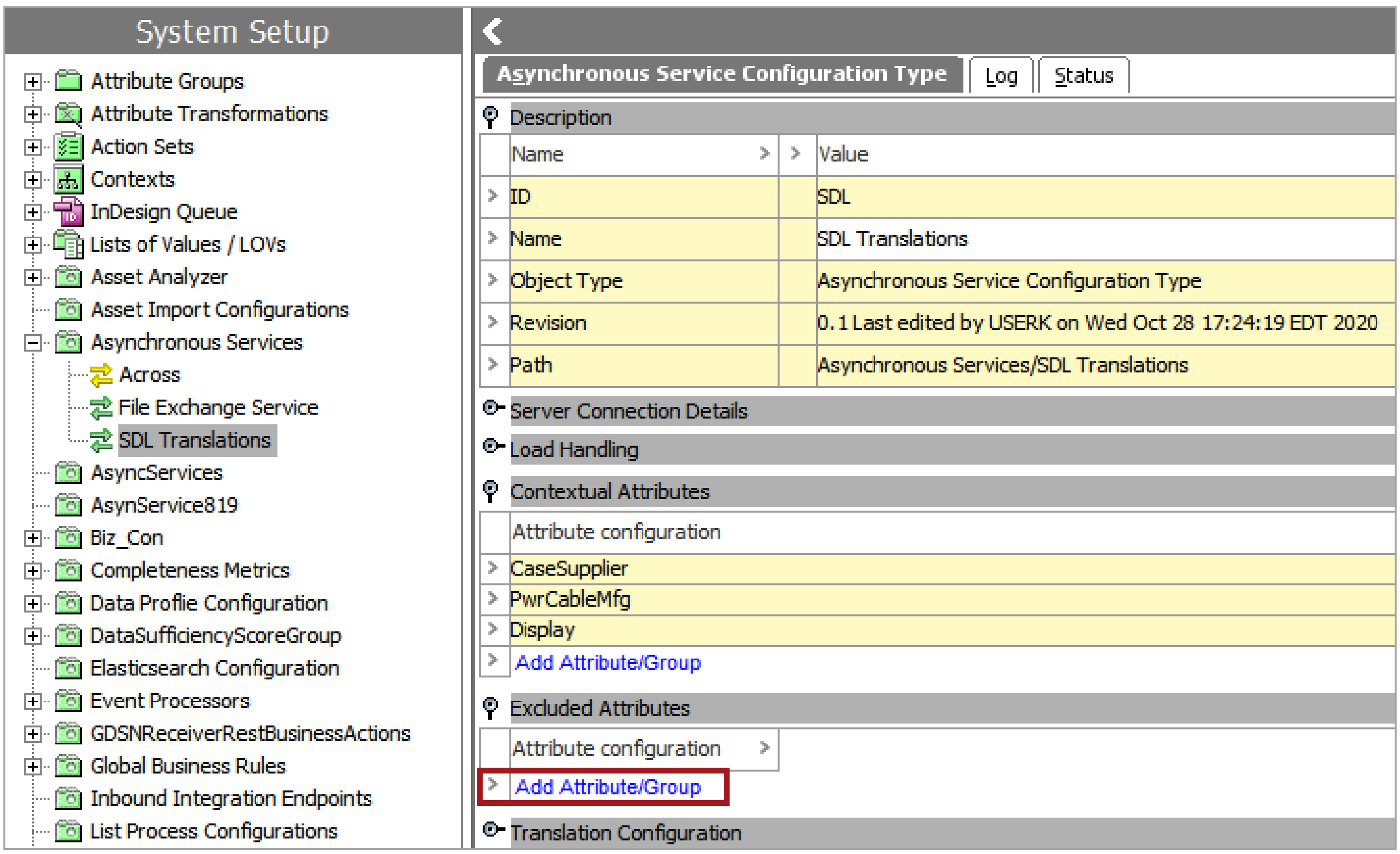
Only the selected contextual attributes / attribute groups will be included when a translation is run using the selected asynchronous service.

Exclude Attributes for Asynchronous Translations

Individual language-dependent attributes or entire language-dependent attribute groups may be irrelevant for translation. The 'Excluded Attributes' parameter allows users to define the attributes that should not be translated.

To exclude attributes and/or attribute groups from an asynchronous translation:

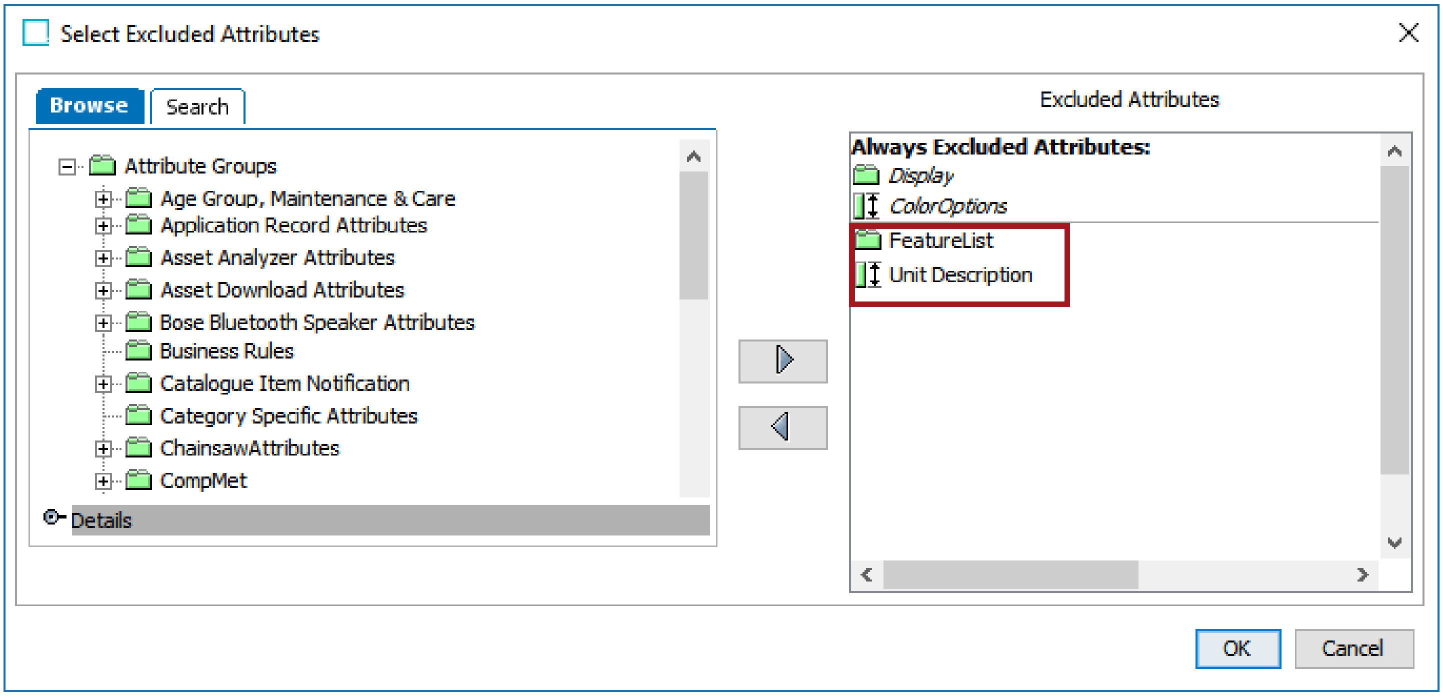
1. Select the desired asynchronous translation service (in this example, 'SDL Translations'), open the 'Excluded Attributes' flipper, and click the 'Add Attribute/Group' link.



2. In the 'Select Excluded Attributes' dialog, select the desired language-dependent attributes and attribute groups to exclude from translation when a translation is run using the selected asynchronous service, and click the right arrow ( ) button to move those selections to the 'Excluded Attributes' column.

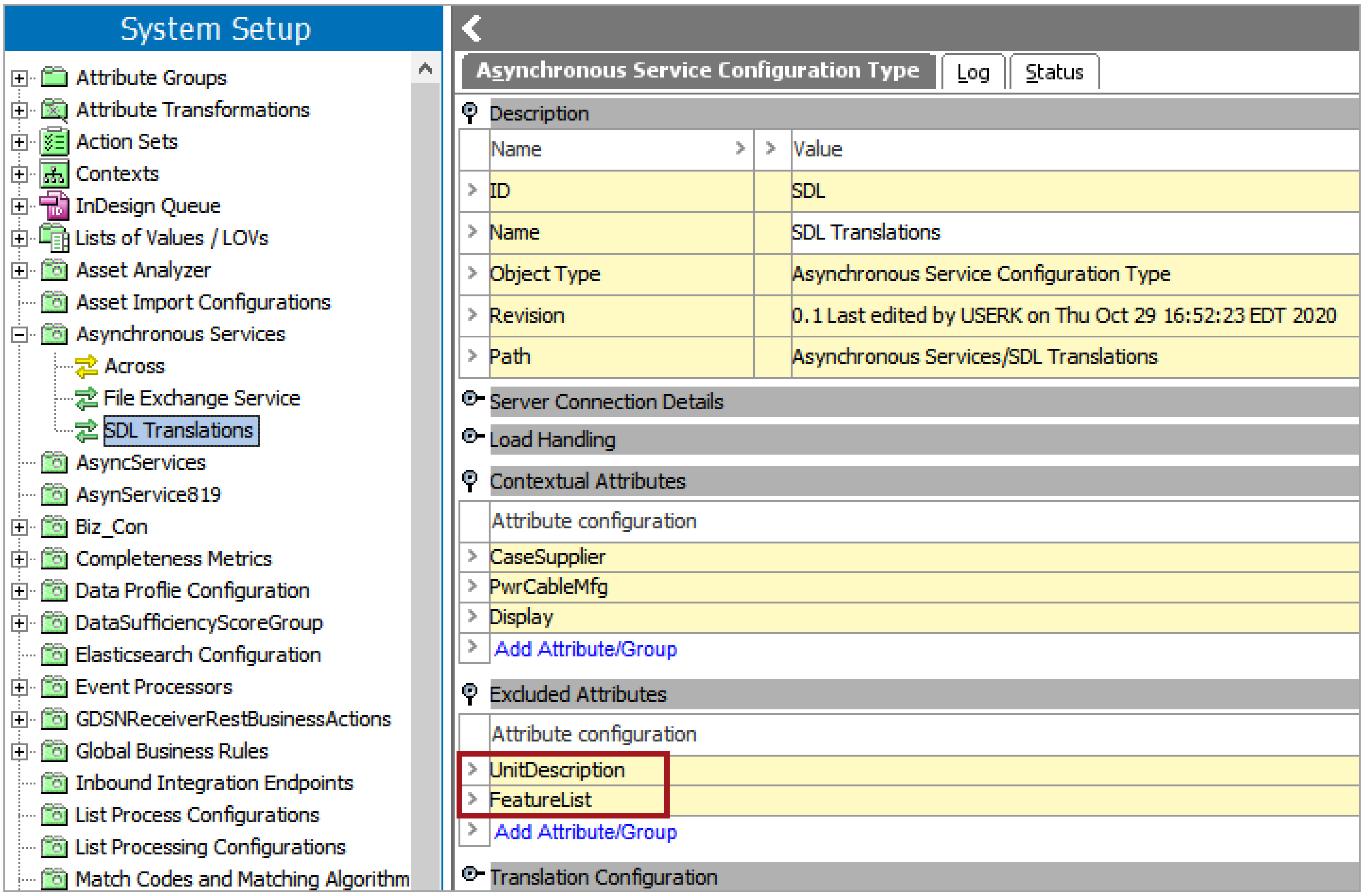
   - To remove selected attributes and attribute groups from the 'Excluded Attributes' column, select the left arrow ( ) button.

   In this example, the user has selected the 'FeatureList' attribute group and the 'Unit Description' attribute to be excluded from translation.



Note: In the image above, the attribute group 'Display' and the attribute 'Color Options,' which are located under 'Always Excluded Attributes,' are selected at the global level within the system settings. For all translation methods, these attributes will always be excluded. For more information, see the **Translation Settings** topic in the **System Settings** section of the **System Setup / Super User Guide** documentation.

3. Click 'OK' to save the settings and close the 'Select Excluded Attributes' dialog. The excluded attributes and attribute groups are displayed in the 'Excluded Attribute' flipper.



Only the selected language-dependent attributes / attribute groups will be excluded when a translation is run using the selected asynchronous service.

Business Rules for Asynchronous Translations

Asynchronous translations can be initiated via a business rule which can be triggered in different ways, for example, from within a STEP workflow.

This section assumes that users are familiar with business rules. Most of the information about setting up and managing business rules can be found in the **Business Rules** section of the online help. However, this topic addresses the one business action that is required to initiate an asynchronous translation in a workflow.

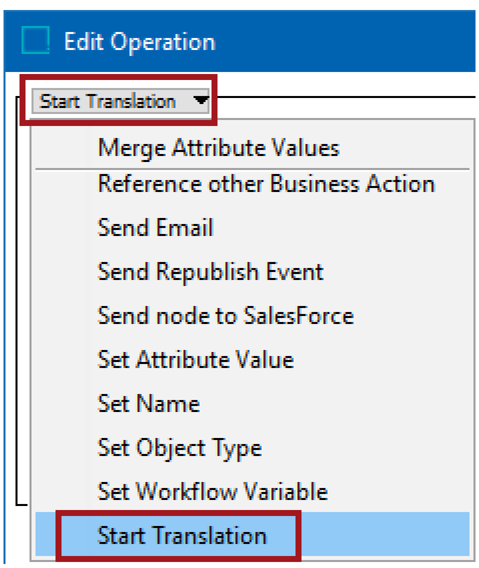
Use the following steps to configure a business action that will start a translation job for asynchronous translation.

Prerequisites

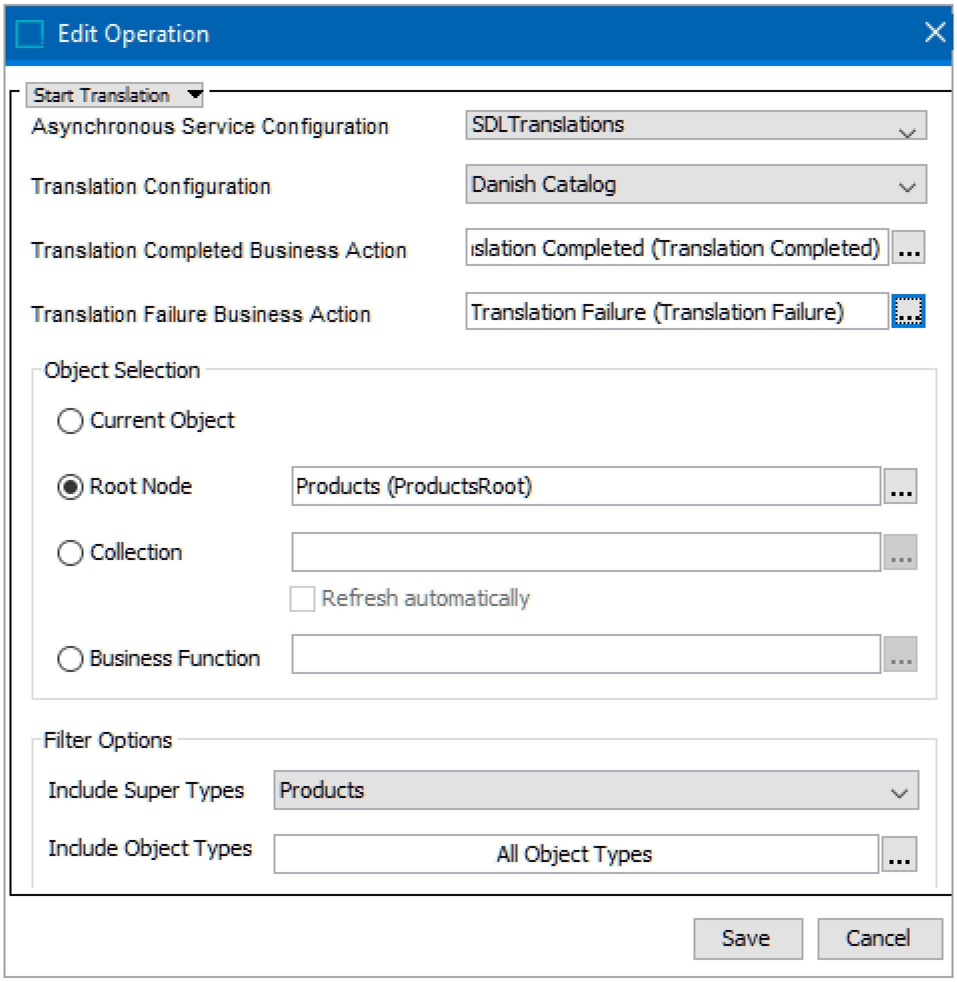
- The 'Translation Completed Business Action' and 'Translation Failure Business Action' can be created prior to being selected for the **Start Translation** Business Action. These actions can move the translation job through the workflow.
- The **Start Translation** Business Action only displays if the functionality for asynchronous translations has been activated on your system. See your Stibo Systems account representative about adding the functionality, if desired.

Configuration

1. Create a business action in System Setup, then right-click and select Edit Business Rule.
2. In the Business Rule Editor, give the rule an ID and Name, then click **Add new Business Action** to open the Edit Operation dialog.
3. In the Edit Operation dialog, select the **Start Translation** business operation, as shown in the image below.



4. Supply the Start Translation operation parameters shown below as required:



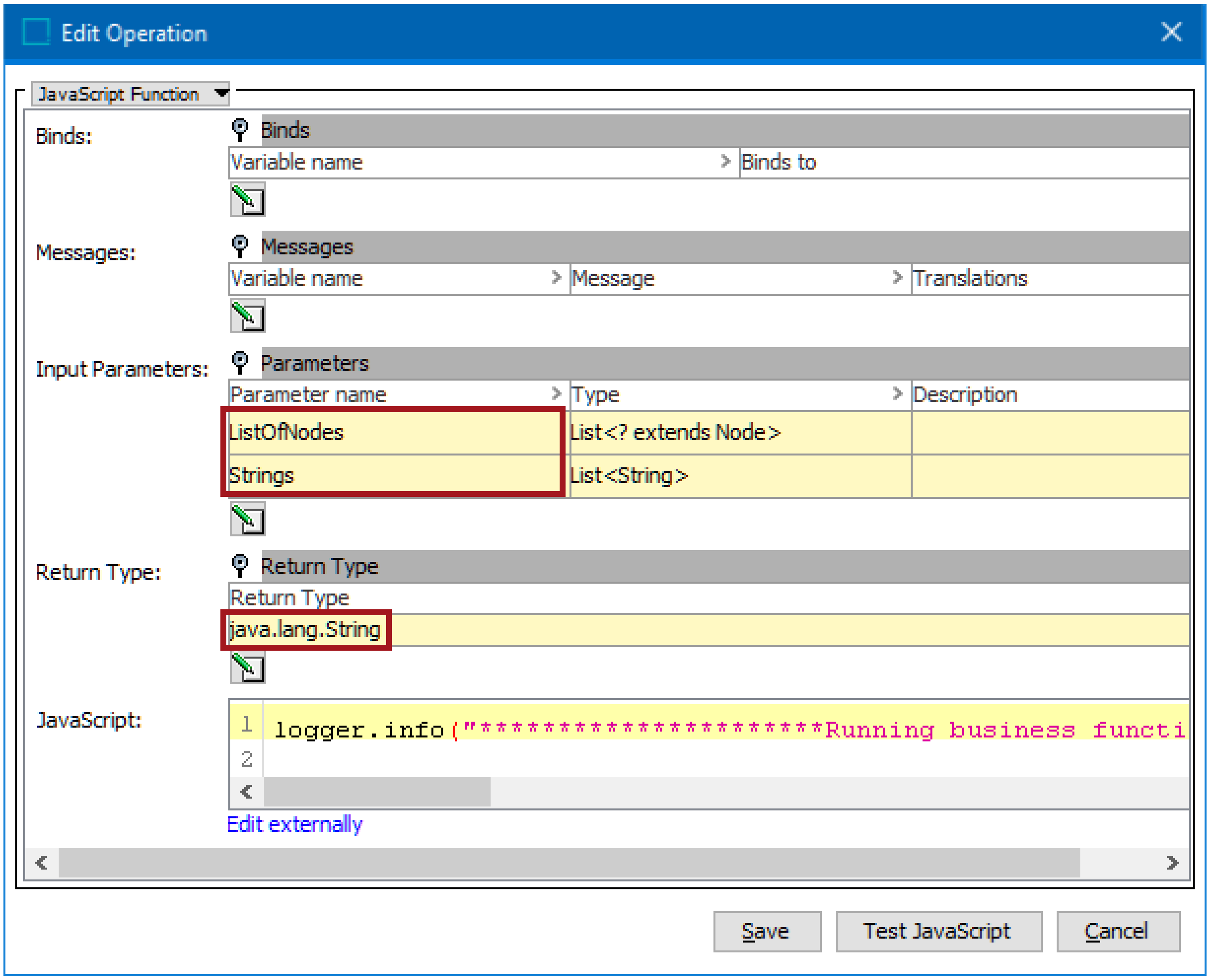
- For **Asynchronous Service Configuration** - Use this dropdown to select the Service Configuration you want to initiate.
- For **Translation Configuration** - Select the Translation Configuration to which the business rule applies. There can be many Translation Configurations for the same Service Configuration but the business action must be configured to use just one translation configuration.
- For **Translation Completed Business Action** - Select the business action that must be executed once a translation job is completed. If the translation was initiated in a STEP workflow task, you can use the Trigger STEP Workflow Event business action to move the workflow task to the next state.
- For **Translation Failure Business Action** - Select a business action that will be executed when a translation job fails.
- For **Object Selection** Options - Choose one of the following options:
  - **Current Object** - For accessing the STEP object that the business rule is being evaluated or executed against.

- **Root Node -** For selecting objects directly from the Tree. These can include products, assets, classifications, attributes, and LOVs. Click the ellipsis button (...) to either browse the Tree or search for a specific root node.

> **Important:** If a user selects LOVs or Attributes / Attribute Groups from the list of available nodes, the **Filter Options** (detailed below) will be inactive.

- **Collection -** For selecting a manually selected grouping of various object types (which can, potentially, be from multiple super types) grouped in a collection. In the Collection field, click the ellipsis button (...) and either browse to or search for the collection to be translated. If the collection is based on a search and you want to ensure that new or amended values in the collection are captured prior to exporting for translation, click the box beside **Refresh Automatically**. Choosing this option activates the selections in the **Filter Options** section of the dialog. See the 'Filter Options' subsection below for more details.
- **Business Function -** For creating output from STEP to send to the translation service. Click the ellipsis button (...) to browse to or search for the relevant business function. For more detailed information about how to use business functions, see the **Business Functions** section of the **Business Rules** documentation.

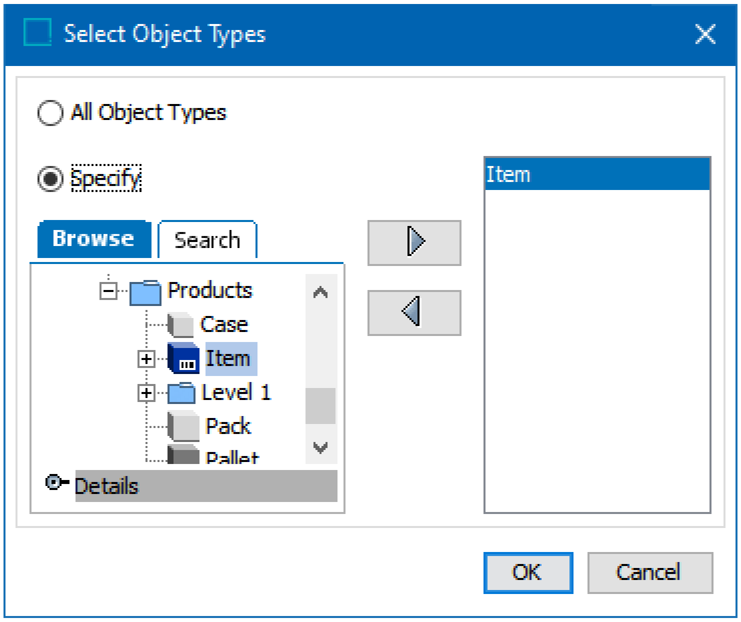
  As shown in the following image, to use the Business Function feature for asynchronous translations, two input parameter types are required: **List<? extends Node>** and **List<String>**. The Return Type **java.lang.String** must also be used.



- For **Filter Options** - set the parameters as required:
  - **Include Super Types** - From the dropdown list, select which super types should be translated. This field is only activated if **Root Node, Collection, or Business Function** was chosen for Object Selection. The super type options are Products (default), Classifications, Assets, or All.
  - **Include Object Types** - Click the ellipsis button (**...**) to display the **Select Object Types** dialog and select the object types to be included in the translation export. This field is only activated if **Root Node, Collection,** or **Business Function** was chosen for Object Selection.

    In the dialog, click the **Specify** radio button to enable the Browse and Search tabs for the Tree window. Select the relevant object type(s), then click the right-facing arrow ( ) to add the object type(s) to the filter. To remove an object type from the filter, select it and click the left-facing arrow ( ).

    Click **OK** to close the dialog.



5. Click the **Save** button to close the Edit Operation dialog and add the operation to the business rule.

Note: Exported translation files can contain either data objects (products, classifications, assets, etc.) **or** system setup objects (attributes, LOV, etc.); they cannot contain a combination of the two.

Running the Start Translation Business Rule

After setup of the business action is complete, testing or running the business rule starts a background process that is displayed on the BG Processes tab under the Async Job Runner node. The job displays a status of 'suspended' until the translation job is complete.

Translation Status Widget Configuration

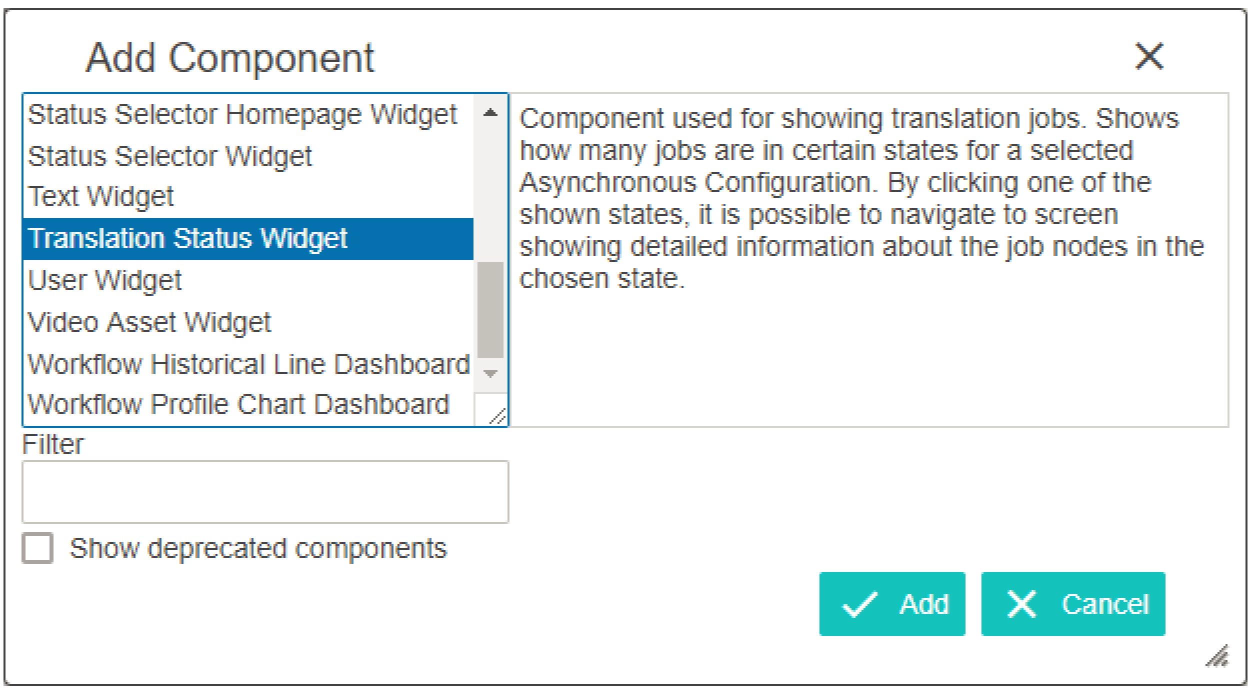
The Translation Status Widget that, once configured along with the necessary screens, enables users to see the number of translation jobs that are in certain states for a selected asynchronous configuration, and select those states for more detailed information.

Configuring the Translation Status Widget

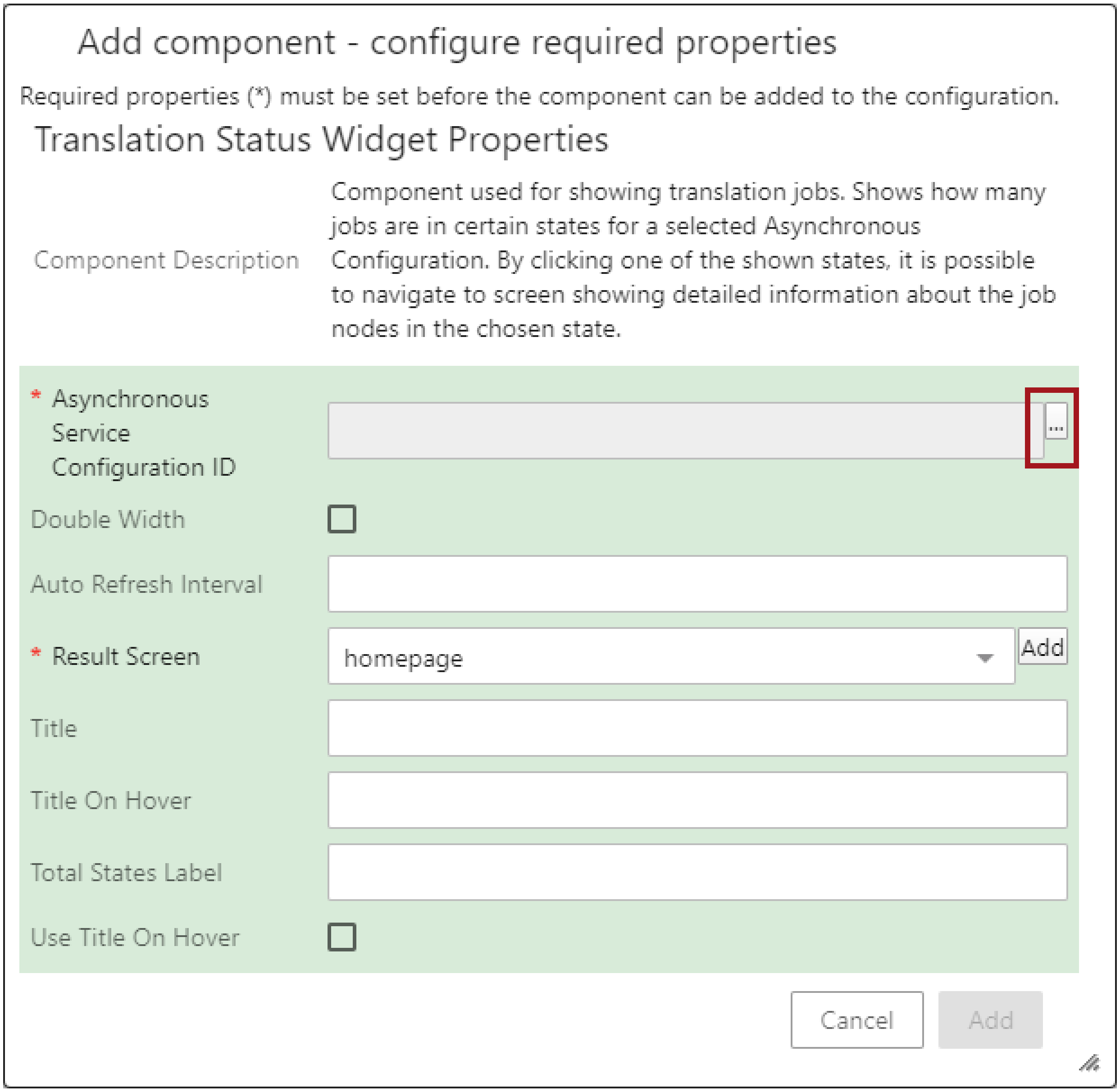
Note: Before configuring the Translation Status widget, users are recommended to read the **Asynchronous Translations in Web UI** topic , and need to complete the steps presented in the topic **Configuring Screens for Asynchronous Translation Status in Web UI**.

While the following steps will detail adding and configuring a Translation Status widget, it is assumed that the reader has a working knowledge of how to add a widget to the homepage. For more information regarding homepage widgets, including adding widgets to a homepage, see the **Homepage Widgets** topic in the **Using Web UI** documentation.

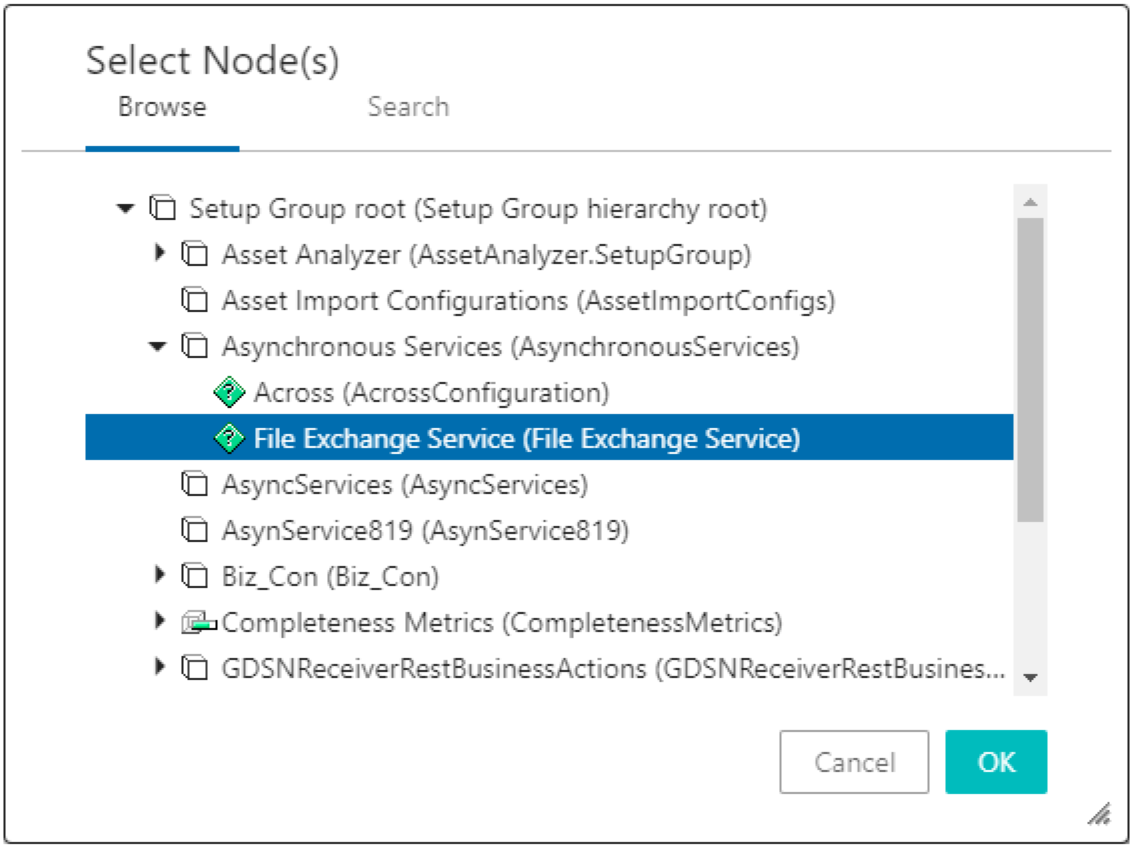
1. In the Add Component dialog, select 'Translation Status Widget' and click 'Add.'



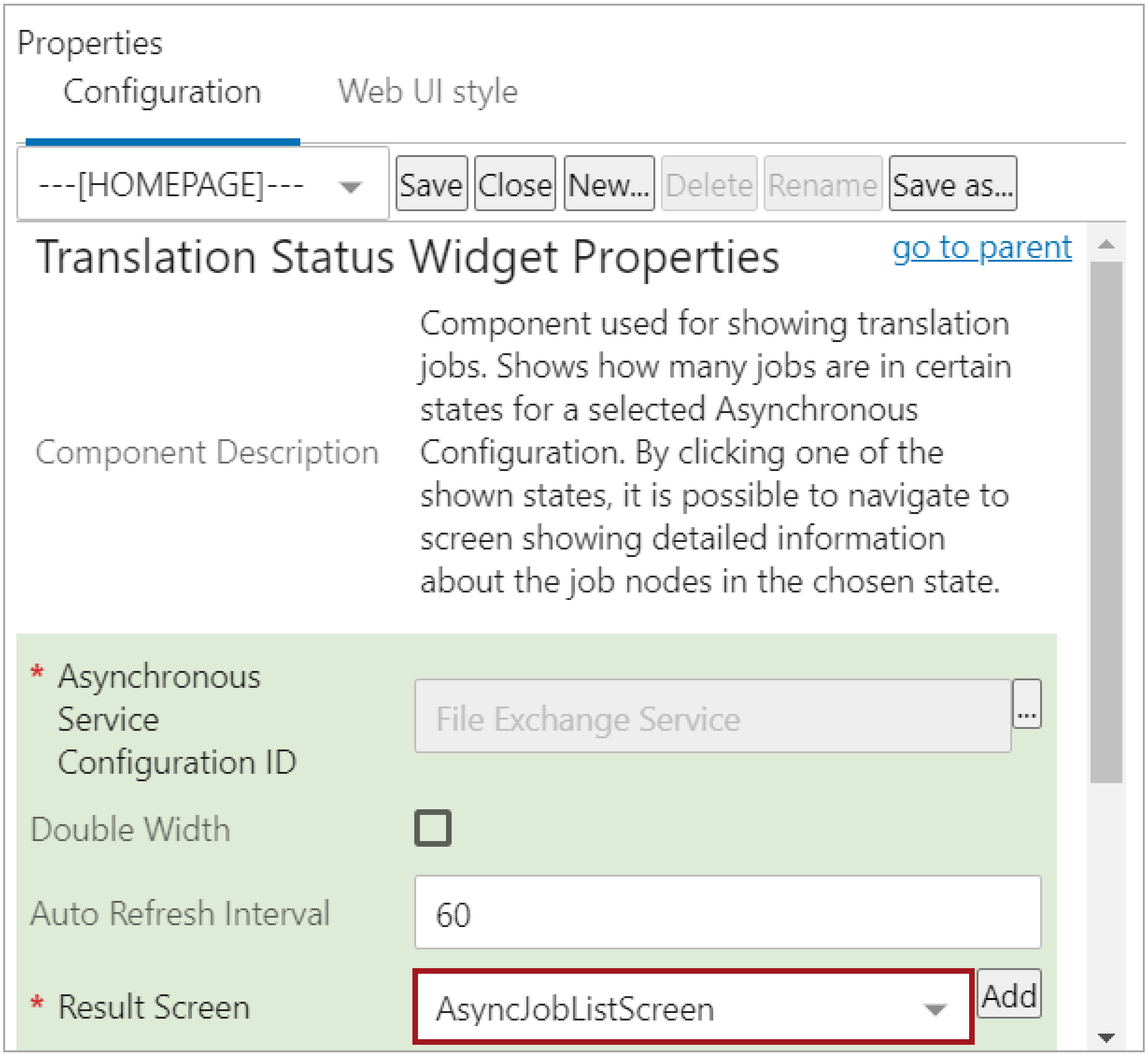
2. Click the ellipsis button (...) located next to the Asynchronous Service Configuration ID.



3. Select the desired asynchronous service (in this example, 'File Exchange Service') and click 'OK.'



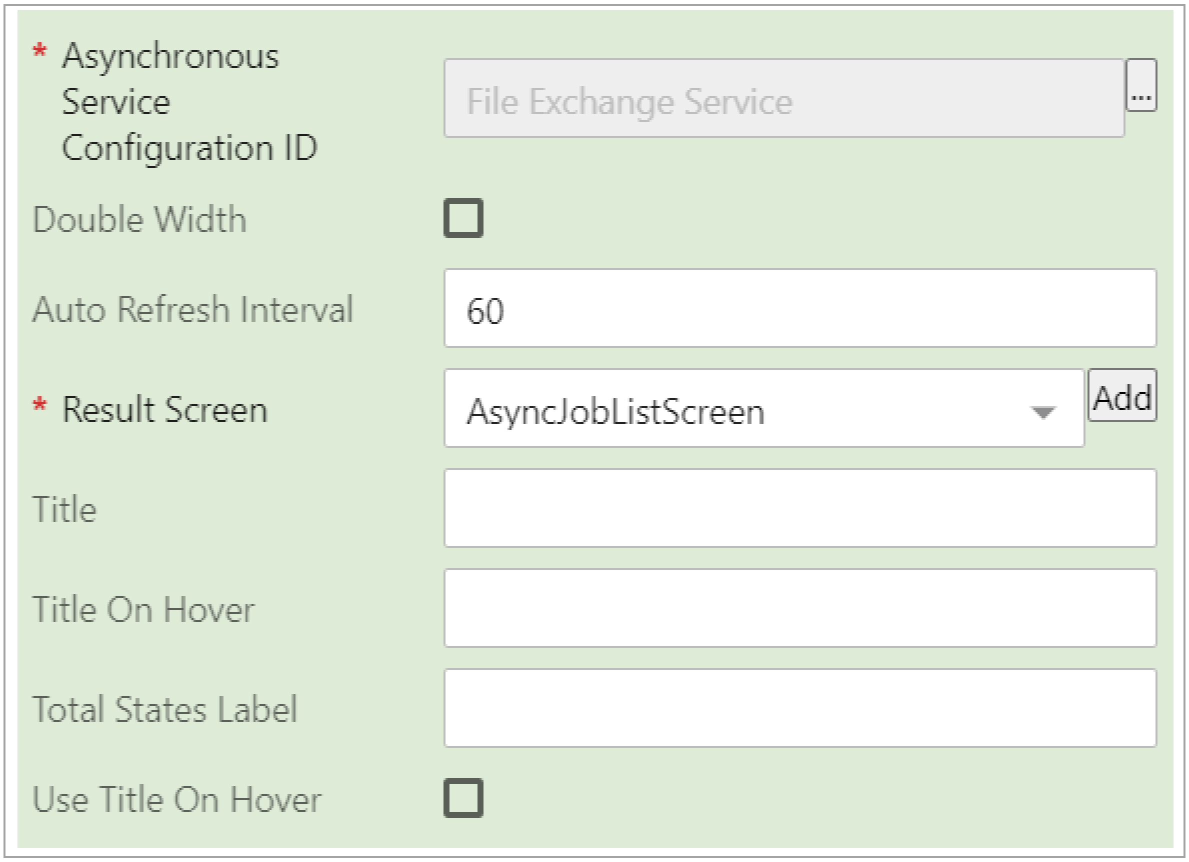
4. Open the 'Result Screen' dropdown menu and select the 'AsyncJobList Screen' screen. This screen was configured as part of the **Configuring Screens for Asynchronous Translation Status in Web UI** topic.



5. Click 'Save' to save the Translation Status widget settings.
6. Click 'Close' to close the designer.

Optional Parameters in the Properties dialog

The two parameters, 'Asynchronous Service Configuration ID' and 'Result Screen', are mandatory and are identified as such by asterisks. All other parameters are optional and described in detail below.



Double Width

Doubles the width of the widget on the Home Page.

Auto Refresh Interval

If desired, enter the number of seconds that should pass before the widget is automatically updated. Automatic updates are disabled if this field is left blank.

Title

Enter the title for the Translation Widget.

Title On Hover

If a title is entered in the 'Title' parameter, and this parameter is selected, the assigned title will appear when the user hovers the mouse over the title area within the widget.

Total States Label

If given a value, this will include the total translation states within the selected asynchronous service. In this example, the user has provided a value of 'Total States' for the parameter.

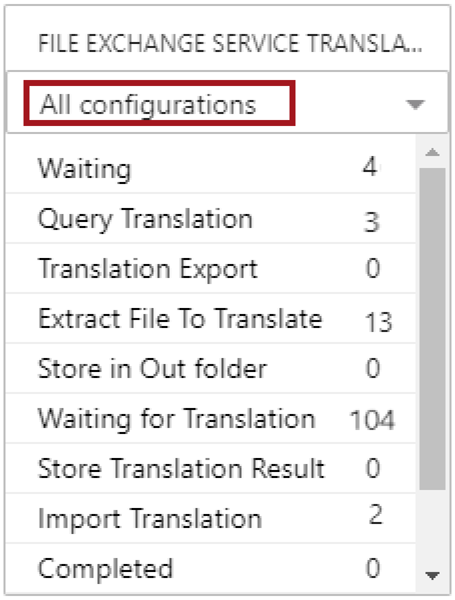
Use Title On Hover

If a title is provided in the 'Title" parameter, the title will appear when the user hovers the mouse over the top of the Translation Status widget.

Using the Translation Status Widget

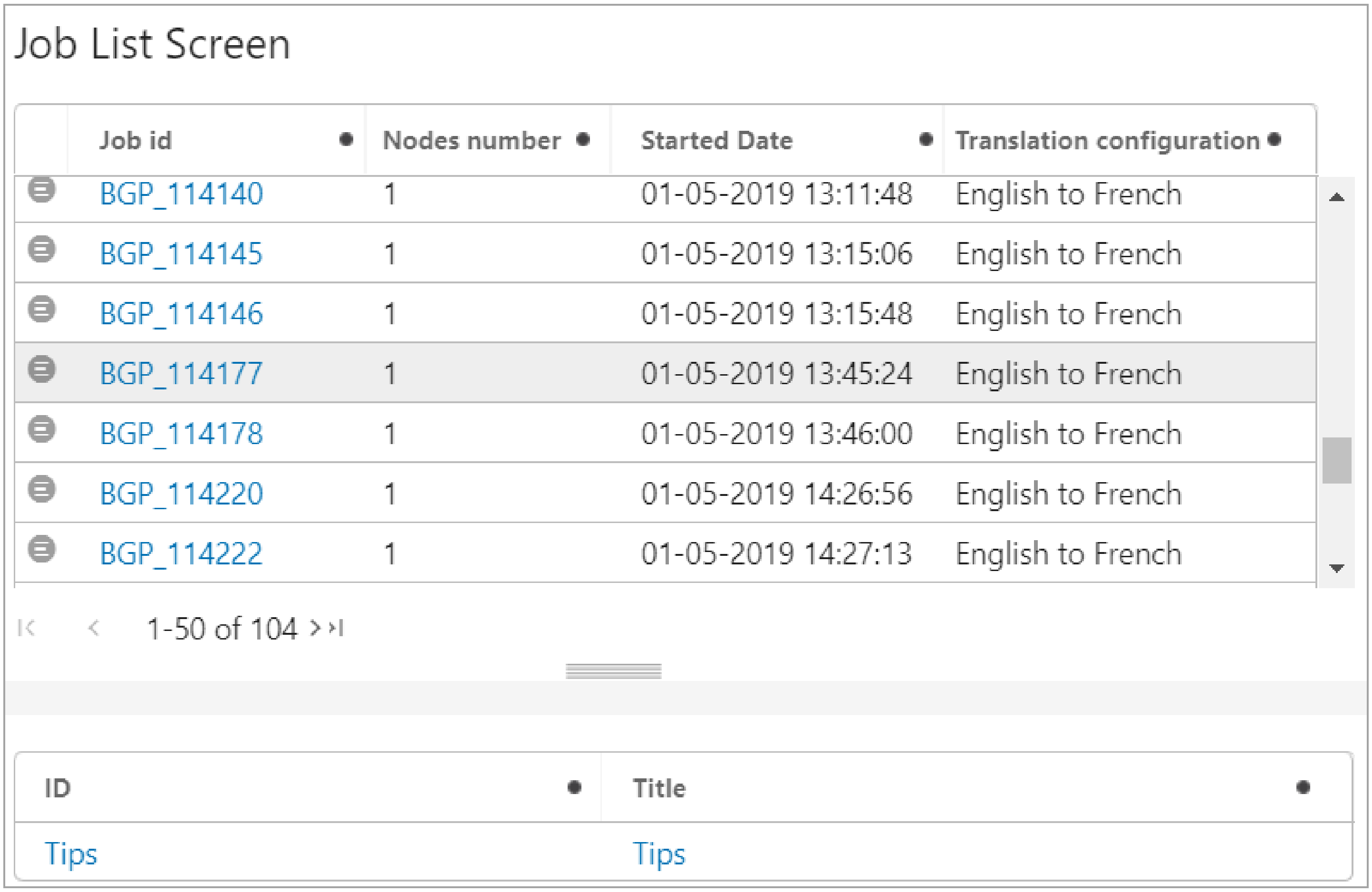
The Translation Status widget should now be visible on the Web UI homepage.

1. From the dropdown menu located in the Translation Status widget, select a translation configuration. In this example, the user has selected 'All configurations.'



2. Select a translation state (e.g., 'Waiting', 'Query Translation', etc.). In this example, 'Waiting for Translation' has been selected.

   The Job List Screen and Job List Node Screen, separated by a moveable handle, opens.



Job List Screen

	Job id	Nodes number	Started Date	Translation configuration
	BGP_114140	1	01-05-2019 13:11:48	English to French
	BGP_114145	1	01-05-2019 13:15:06	English to French
	BGP_114146	1	01-05-2019 13:15:48	English to French
	BGP_114177	1	01-05-2019 13:45:24	English to French
	BGP_114178	1	01-05-2019 13:46:00	English to French
	BGP_114220	1	01-05-2019 14:26:56	English to French
	BGP_114222	1	01-05-2019 14:27:13	English to French

1-50 of 104

ID	Title
Tips	Tips

For more information on using these two screens to obtain further translation status details, see the **Asynchronous Translations in Web UI** topic.

Dun & Bradstreet Integration

Both STEP Workbench and Web UI support matching data from the Dun & Bradstreet (D&B) database and allow for enriching customer information in STEP with the D&B data. These records, brought in either manually or through inbound integrations, can be matched using confidence scoring and are enriched automatically through business actions or in bulk using event processors and workflows.

The integration to D&B services can be implemented following a synchronous and/or an asynchronous integration process.

The centralized **synchronous** integration process is started manually and is effective, for example, to onboard individual suppliers. It uses the following elements:

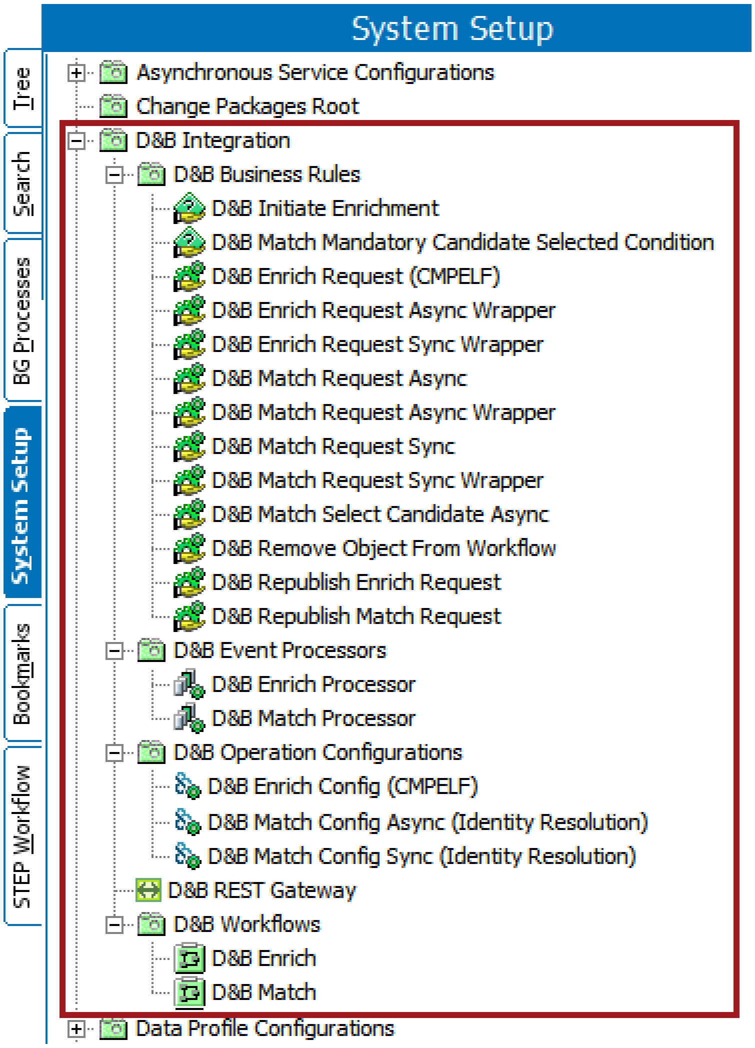
- Business actions for matching, candidate selection, and enriching
- D&B Integration component model

The **asynchronous** integration process is effective, for example, to match and enrich customers in bulk. It uses the following elements:

- Business actions for matching, candidate selection, enriching, removal, and republishing
- D&B Integration component model
- Workflows
- Event processors

To assist with proper configuration, Easy Setup creates a D&B Integration (ID: DnBIntegration) setup folder in System Setup that contains the following elements, as appropriate:

- **D&B Business Rules** - Business actions named according to their function.
- **D&B Workflows** - One workflow for each service (D&B Match and D&B Enrich) used to control the information flow.
- **D&B Event Processors** - Event processors used to request changes from third-party services.
- **D&B Operation Configurations** - A node for each D&B operation with a configuration specific to each operation.
- **D&B REST Gateway** - The default gateway for all communication with the Direct+ API.



Additionally, attribute groups, attributes, and data containers are created as part of the Easy Setup. Over 200 attributes / data containers (including those for candidate matching and enriching) are created. For the full list of attributes, see the **D&B Attributes and Data Containers** topic.

Prerequisites

Important: The Dun & Bradstreet Direct+ Enrichment Service commercial license is required to use the D&B functionality and enables access to a new system license (X.DnBIntegration) and the dnb-integration add-on component. Also, you can make D&B Direct API calls using Secure Sockets Layer (SSL) with the URL of the D&B endpoint: https://direct.dnb.com.

- Configuring the component model requires that users are familiar with the workbench System Setup tab and how to configure within this area (e.g., creation and maintenance of object types, attributes, and references). Users must also have the privileges required to carry out these tasks.

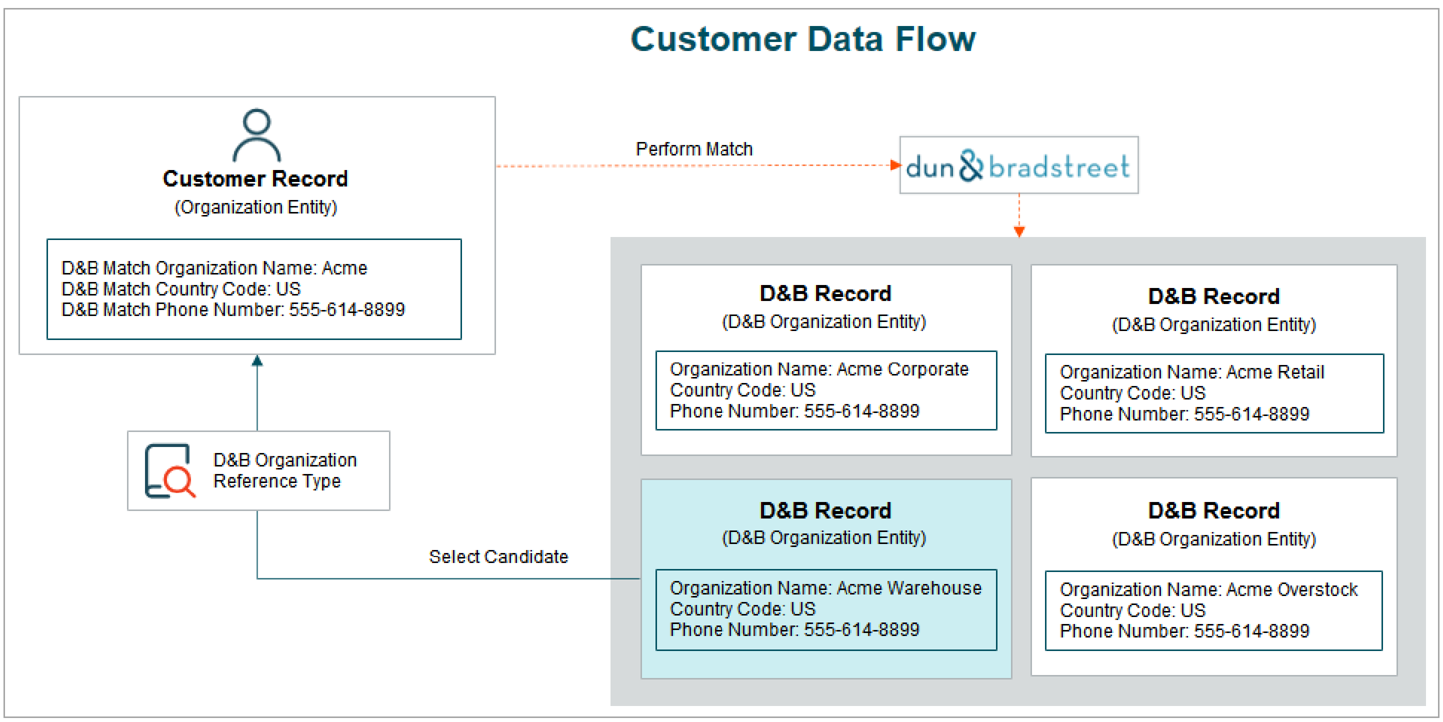
- Users should be familiar with business rules, workflows, event processors, and it is expected that anyone configuring Web UI components is familiar with Web UI designer. Most of these concepts are covered outside of this D&B documentation. For more information, search online help to understand these concepts and processes.
- Configuring and using the D&B integration requires the proper privileges and understanding of how all the pieces work together. The Easy Setup is designed so very little configuration is needed to get up and running as soon as possible.

Important: The Event Processors used for D&B Matching and D&B Enrichment each process one request to D&B's web services at a time. The response times from these web services vary. Commonly, 1,500 - 3,000 D&B Enrichment requests and 3,000 - 7,000 D&B Matching request can be processed per hour.

Customer Data Model

The D&B process diagram below shows why the setup is important:

- Input for D&B Match requests is taken from a configurable set of attributes on the organization record.
- Data returned from D&B is stored in an entity, in a fixed data model created by the Easy Setup that is comprised of data containers and attributes.
- D&B Matching creates the D&B organization entities and creates a reference from the organization record to the D&B organization record.



Additional information on the D&B Integration includes:

- D&B Processing Examples
- D&B Integration Configuration and Easy Setup
- D&B Matching
- D&B Enriching
- D&B in Web UI
- D&B Attributes and Data Containers
- D&B Data Storage
- D&B Direct+ API Products
- D&B Error Handling with Integration Status

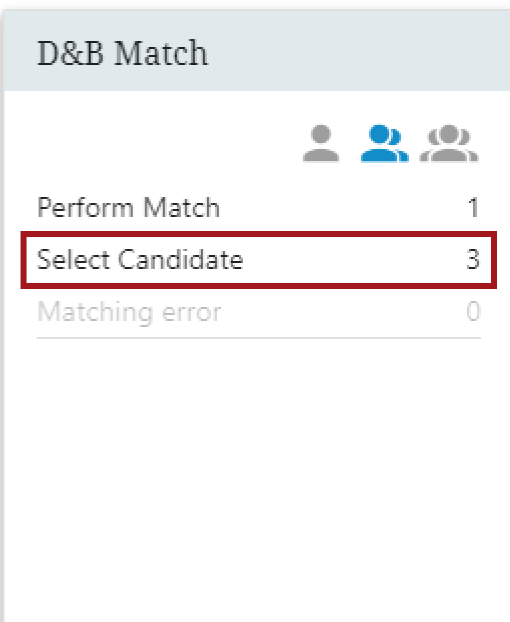
For information about migrating from D&B V2 to Direct+, see the **Dun & Bradstreet Migration: Direct 2.0 to Direct+** topic in the **System Administration** documentation.

D&B Processing Examples

The Dun & Bradstreet (D&B) matching and enriching processes have similar steps regardless of the process. The differences are noted in the example shown below. True functionality depends on the actual setup. Web UI is recommended for all D&B user activities.

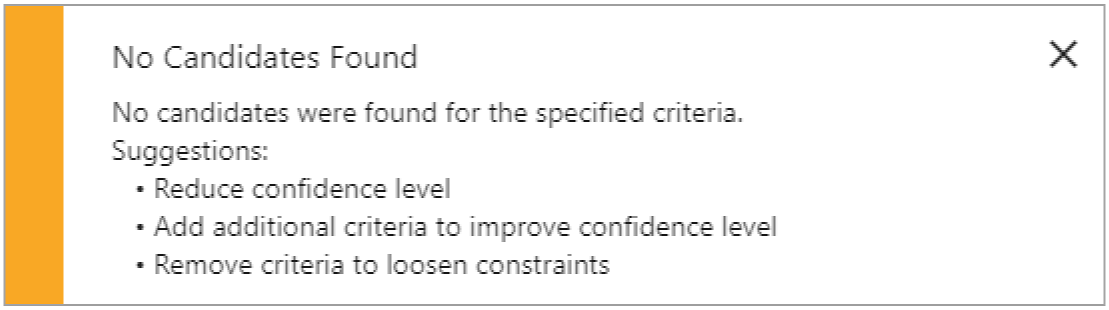
Basic functionality includes:

1. Start matching based on the selected process:
   - For **asynchronous** processing, entities are initiated into the D&B Match workflow.
   - For **synchronous** processing, a user clicks the configured action button, for example, a 'D&B Match Now' button.
2. For **asynchronous** processing only:
   - Once matching is done, the entity moves to the Select Candidate state.
   - On the D&B Match status selector, the user clicks the Select Candidate state.

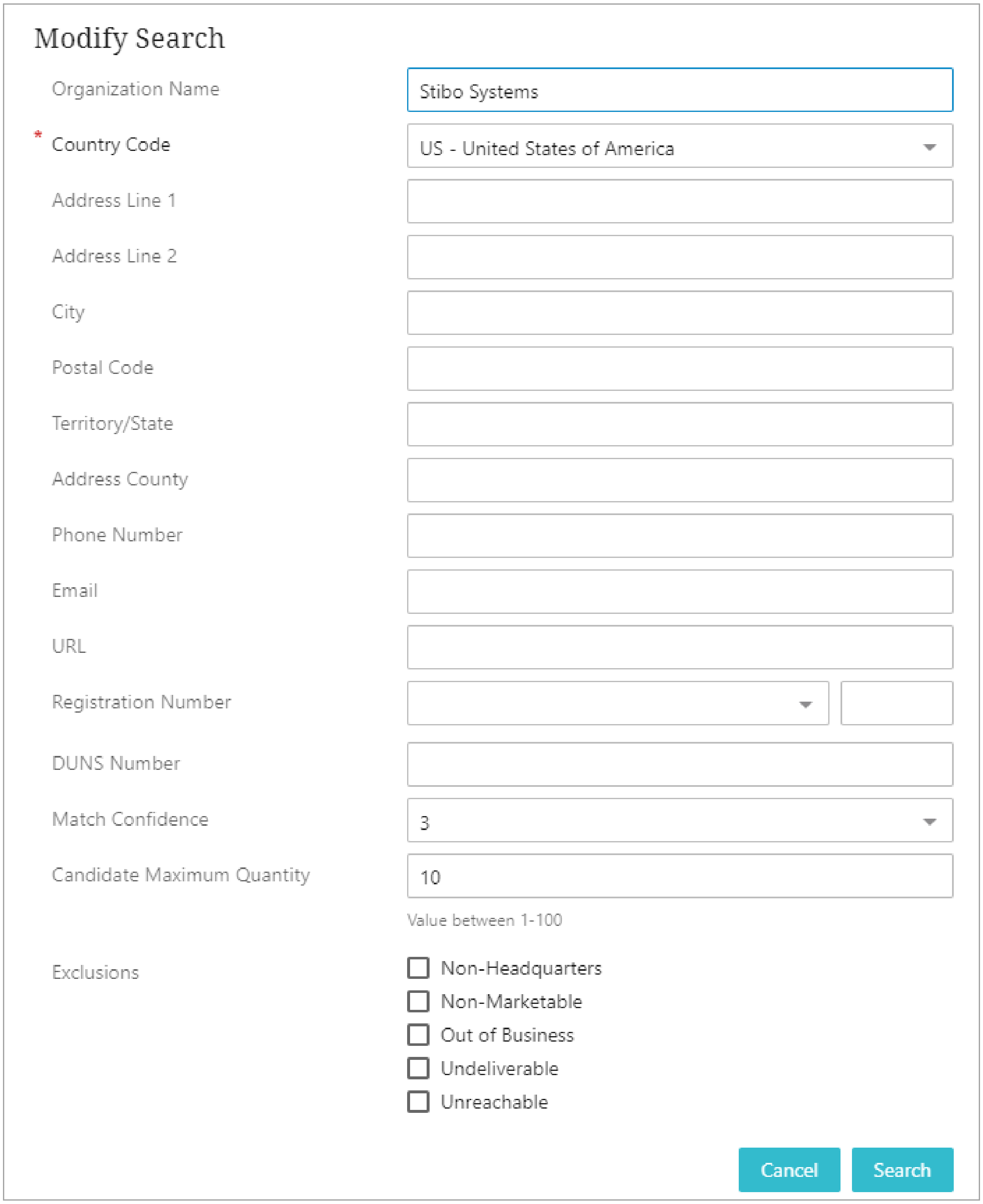


   - On the Task List, the user selects a candidate by clicking on the ID or Title link within a row.
3. For both processing options, a search is performed. The results shown are determined by a D&B algorithm used to match the D&B records to the customer.

- If no candidates are found, a message displays.

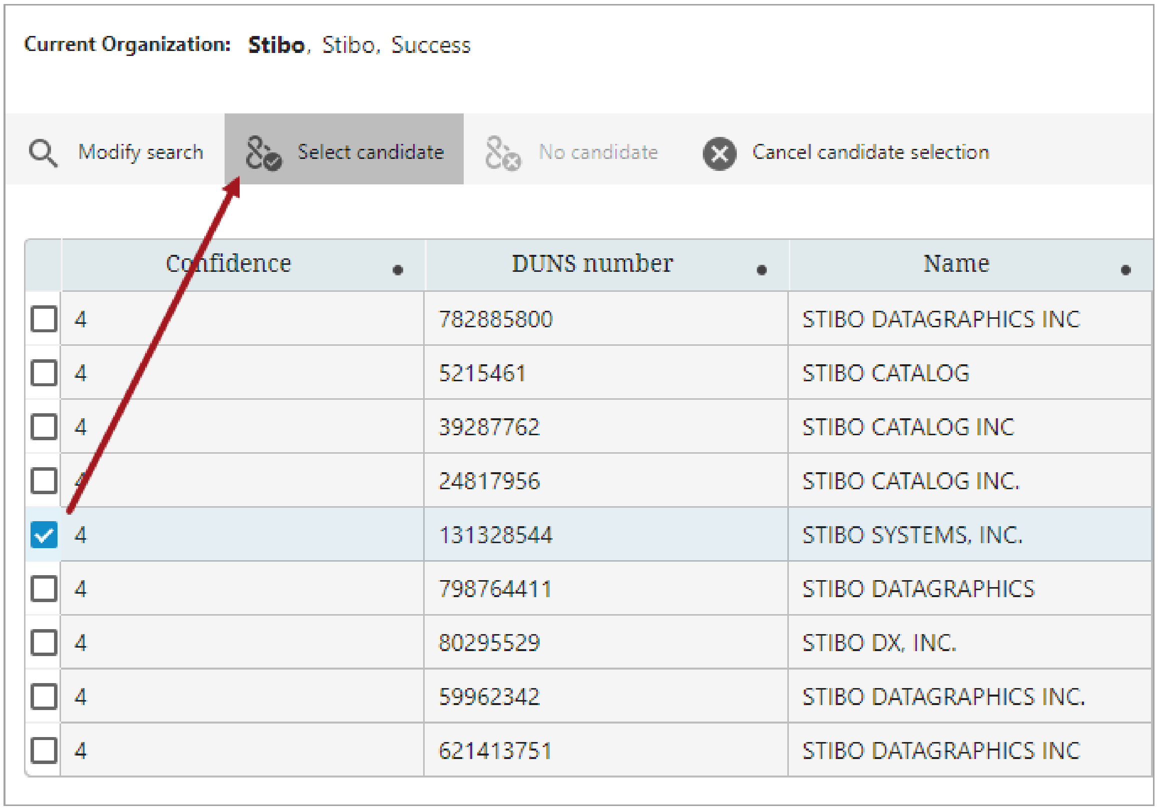


The user clicks the 'Modify search' button on the screen to edit the search. The exclusions (enabled via workbench) can be used to modify the search.

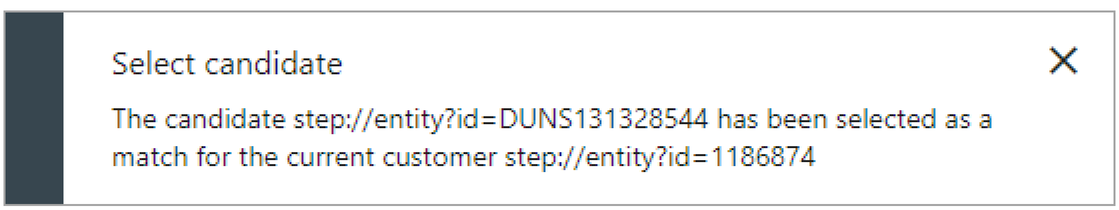


- If multiple candidates are returned, the configured asynchronous or synchronous D&B Match Candidates screen is displayed, and the user compares customer record information with the D&B records list shown.

If a suitable candidate is displayed, the user clicks the D&B record row and clicks the **Select candidate** button. Only one candidate can be selected. The customer record and the D&B record are linked via the D&B Organization Reference Type mapping in the component model.



A message confirms a candidate selection:



If no satisfactory candidate is found the user clicks the **No candidate** button (with no rows selected). The component model setting for 'D&B No Match User Decision' attribute is set to 'true'.

4. When matching is complete, the enrichment process is handled as follows:

   - For **asynchronous** processing, entities can be initiated into the D&B Enrich workflow.
   - For **synchronous** processing, enriching happens automatically.

D&B Integration Configuration and Easy Setup

The Dun & Bradstreet (D&B) integration Easy Setup creates all the needed files, attributes, and data containers. Follow this topic to ensure your data model allows the D&B integration to function properly.

Prerequisites

Define the relevant data structures required for the easy setup component model:

1. Create the objects for the new D&B organization entity records as returned from D&B:

   - **Organization Entity Type** - entity object type for the customer records.
   - **D&B Organization Entity Root** - root where D&B records are saved.
   - **D&B Organization Entity Type** - entity object type for the D&B records.

   For more information, see the **Object Types and Structures** topic in the **System Setup / Super User Guide** documentation.

2. Create customer reference types and links for the D&B records that are referenced by a customer record:

   - **D&B Match Candidate Reference Type** - reference type for D&B match candidates. A reference is made from each candidate to the applicable customer record; once a match is made all the non-matched records are removed from the system and this reference is no longer needed for that particular customer record since matching is complete.
   - **D&B Organization Reference Type** - reference type for D&B matches. Once a match candidate selection is made, the D&B record is linked to the customer record with this reference type.

     Ensure both reference types have the following settings:

     - Only **one** selection for the Valid Source Types flipper which holds the information to be compared.
     - Only **one** selection for the Valid Target Types flipper which is the D&B record that holds the D&B results.

   For more information, see the **Reference and Link Types** topic in the **System Setup / Super User Guide** documentation.

3. Consider creating calculated attributes based on attributes in the organization entity to display data for synchronous and asynchronous matching.

4. Request a **Consumer Key** from your Stibo Systems account manager or partner manager. By default, customers can access up to 500 free D&B ‘match’ transactions. When the 500 ‘match’ transactions are used, customers may purchase ‘match’ and ‘enrich’ licenses directly from Stibo Systems.

   Note: The ‘enrich’ license includes access to both ‘match’ and ‘enrich’ transactions. One transaction of ‘enrich' gives you the ability to do a match first and then obtain a detailed company enrichment. Regardless of the license you choose, only one license key will be provided to cover both 'match' and 'enrich' transactions. Contact your Stibo Systems account manager or partner manager for more information.

Configure D&B Integration

Follow these steps to configure the D&B Integration.

1. Configure the system with your D&B **Consumer Key** and **Consumer Secret** for Direct+. Contact your Stibo Systems account manager or partner manager with questions about this process.

   In the sharedconfig.properties file on the application server, for the case-sensitive **DnBDirectPlus.Credentials.#** parameter, using the example below, add the name to display in workbench (any string except for special characters, e.g., DnB Production), consumer key, and consumer secret for use in the REST Gateway Integration Endpoints. Multiple entries can be added by increasing the number represented by #. `DnBDirectPlus.Credentials.[#]=[Name],[Consumer Key],[Consumer Secret]`

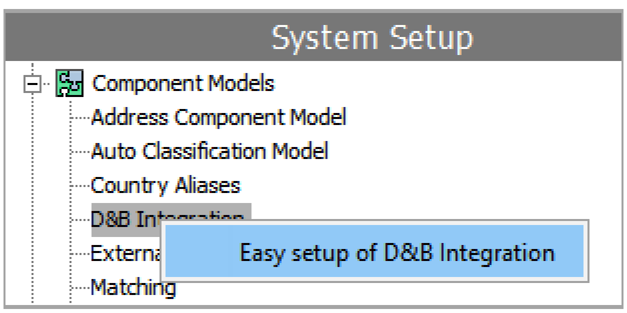
   For example:

   ```
   DnBDirectPlus.Credentials.1 = dnbConfig1,consumerKey,consumerSecret
   ```

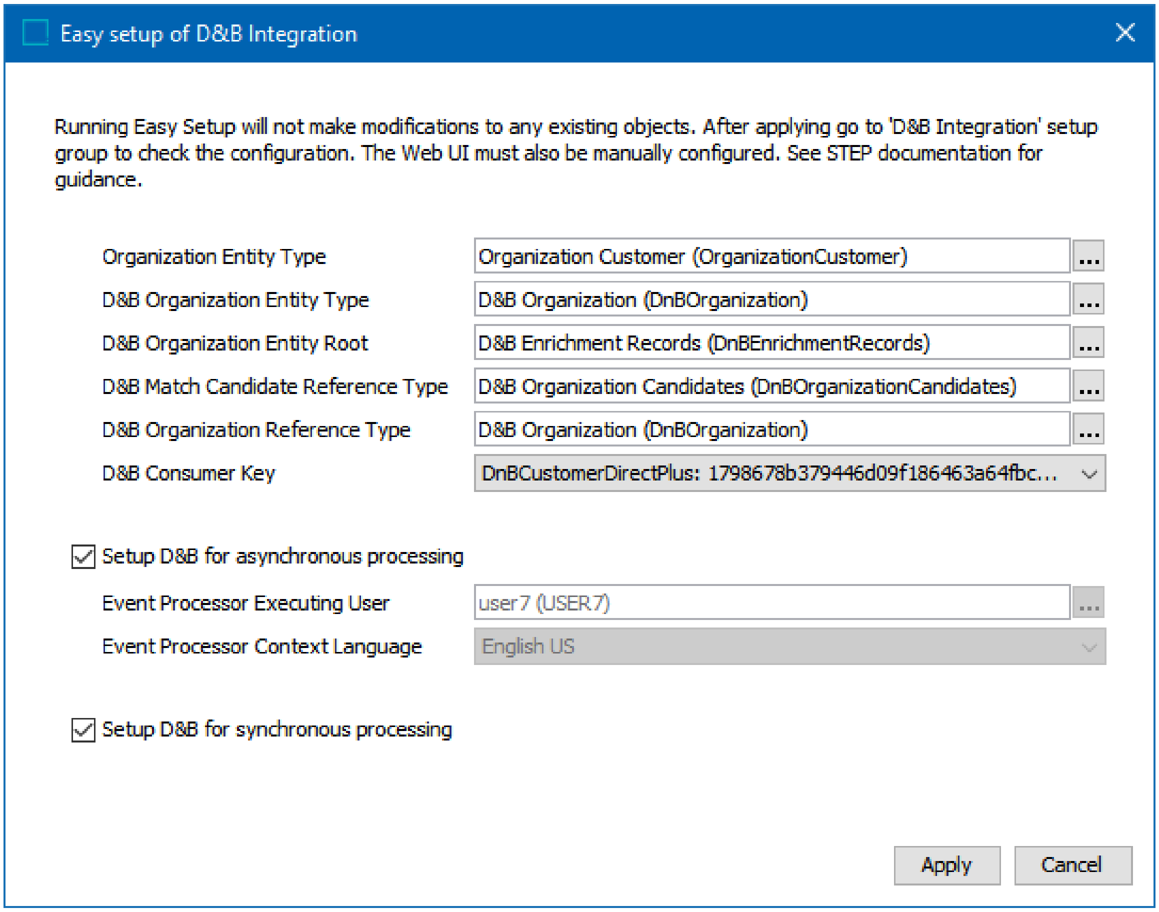
2. Verify or configure the gateway integration endpoint. For details, see the **Initial Setup for a Gateway Integration Endpoint** topic in the **Data Exchange** documentation.

3. On the System Setup tab, open the Component Models node, and select **D&B Integration**.

4. Right-click D&B Integration and click the **Easy setup of D&B Integration** option.



5. On the 'Easy setup of D&B Integration' dialog select the objects and set the options as follows:



Click the ellipsis button (...) or click the dropdown for each parameter to make a selection:

- For the **Organization Entity Type** parameter, select the entity object type for the customer records.
- For the **D&B Organization Entity Type** parameter, select the entity object type for the D&B records.
- For the **D&B Organization Entity Root** parameter, select the root where D&B records are saved.
- For the **D&B Match Candidate Reference Type** parameter, select the reference type for D&B match candidates.
- For the **D&B Organization Reference Type** parameter, select the reference type for D&B matches.
- For the **D&B Consumer Key** parameter, select the Consumer Key received from your Stibo Systems representative.
- For the **Setup D&B for asynchronous processing** parameter, check the box to allow match and enrich processing using workflows. For example, doing matching (perhaps overnight) and enriching automatically when matching is complete.

Note: Prior to STEP version 10.2, all D&B processing was done asynchronously.

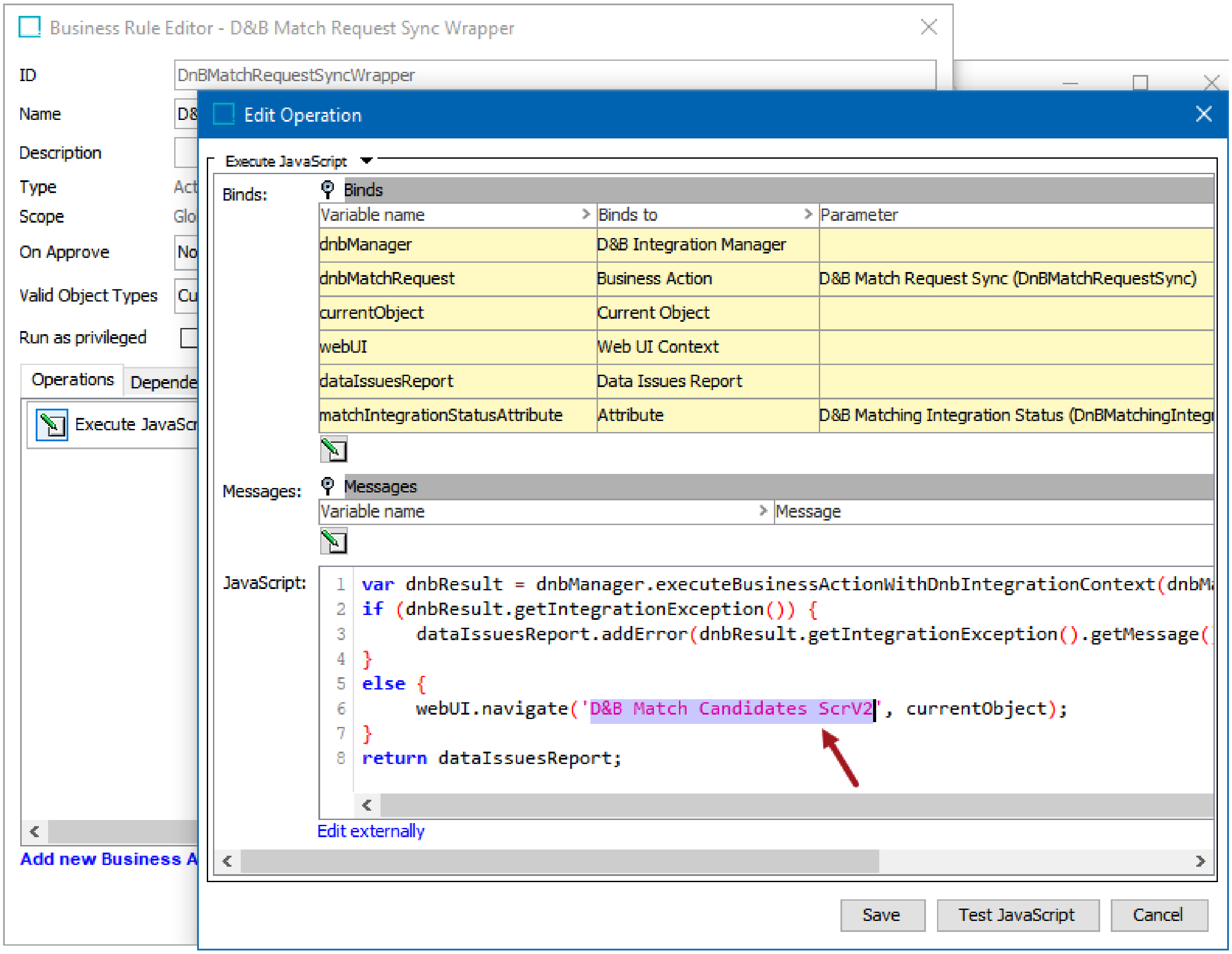
- For the **Event Processor Executing User** parameter, select the user to be associated with all of automatic event processing done as part of the D&B processes.
- For the **Event Processor Context Language** parameter, select the STEP Context that the event processor will use.

- For the **Setup D&B for synchronous processing** parameter, check the box to allow individual match and enrich processing manually when onboarding (outside of a standard workflow).
- Click the **Apply** button. The component model displays the information provided, the setup folders are created, and candidate matching can begin.

6. Configure Web UI based on your D&B processing selections, see the **D&B in Web UI** topic.
7. For **synchronous** processing only, update the name of the 'Select Candidate Screen' to match the name of your Web UI screen configured for synchronous processing. (Asynchronous processing is handled differently, and this update is not required.)

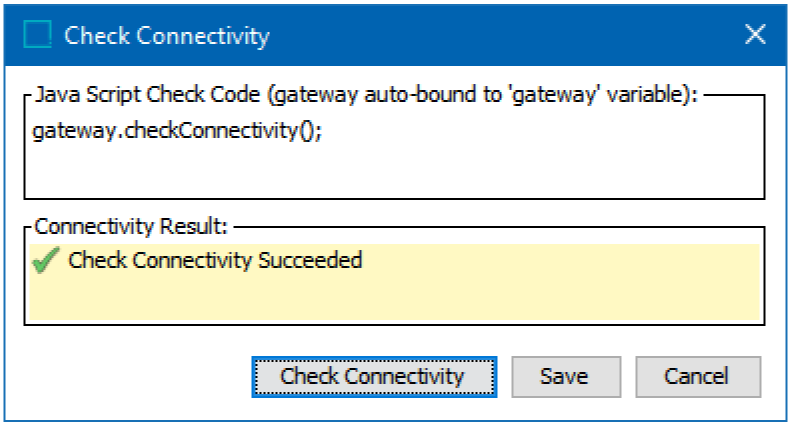
- Open the business action 'D&B Match Request Sync Wrapper (DnBMatchRequestSyncWrapper)'.
- Edit the JavaScript highlighted in the image below: `webUI.navigate('<name of synch Select Match Candidate screen>',currentObject);`

- Click the **Save** button.



8. Open the configured D&B REST Gateway endpoint and on the Configuration tab click the **Check Connectivity** button.

   - If the connectivity result fails, review the setup steps and gateway configuration steps to correct the error. For more information, see the **Configuring a Gateway Integration Endpoint - REST** topic in the **Data Exchange** documentation.



9. Open the Component Models node, select the D&B Integration node, and review the selections.

Component Model Configuration

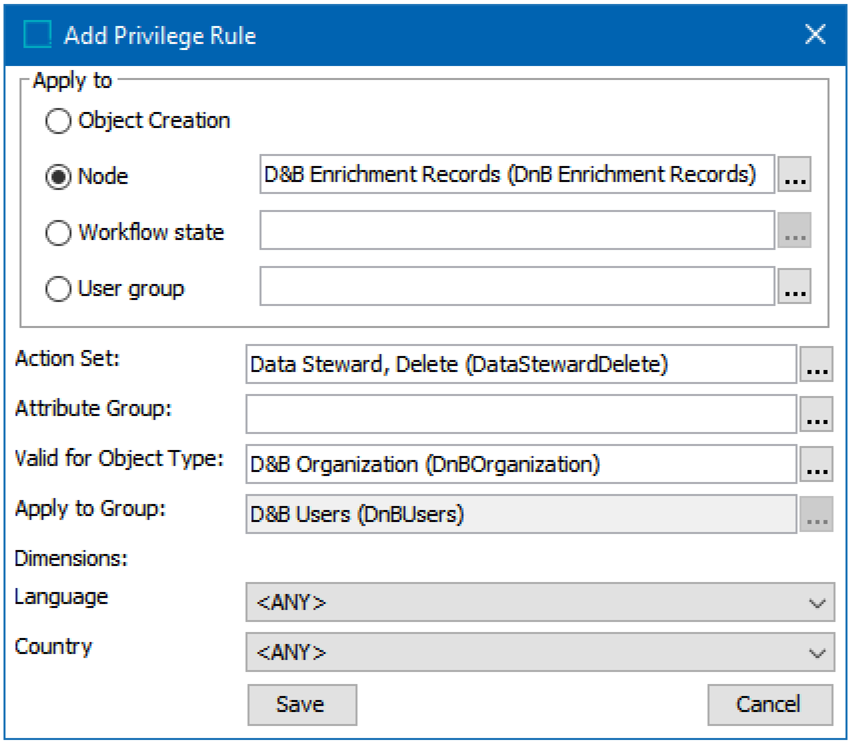
Name	Value	Description
D&B Organization Entity Type	D&B Organization	Used to set the entity type used for the D&B Organization Records.
Organization Entity Type	Organization Customer	Used to set the entity type used for Organization Entities.
D&B DUNS Number	D&B DUNS Number	Attribute on the D&B record to hold DUNS number. Used when running Detailed Company Profile.
D&B Match Address Line 1	D&B Match Address Line 1	Attribute on the customer record to hold first address line used for matching.
D&B Match Address Line 2	D&B Match Address Line 2	Attribute on the customer record to hold second address line used for matching.
D&B Match Confidence	D&B Match Confidence	Attribute on the D&B candidate references to hold confidence value on candidate references.
D&B Match Country Code	D&B Match Country Code	Attribute on the customer record to hold country pdActionCode used for matching.
D&B Match County	D&B Match County	Attribute on the customer record to hold the county used for matching.
D&B Match DUNS number	D&B Match DUNS Number	Attribute on the customer record to hold DUNS number used for matching.
D&B Match Email	D&B Match Email	Attribute on the customer record to hold the email used for matching.
D&B Match Grade	D&B Match Grade	Attribute on the D&B candidate references to hold the rating scored by the D&B match engine.
D&B Match Organization Name	D&B Match Organization Name	Attribute on the customer record to hold the organization name used for matching
D&B Match Phone Number	D&B Match Phone Number	Attribute on the customer record to hold the phone number of a company.
D&B Match Postal Code	D&B Match Postal Code	Attribute on the customer record to hold the postal pdActionCode used for matching.
D&B Match Reason Code		Attribute on the customer record to hold the reason pdActionCode if searching for a company located in Germany.
D&B Match Registration Number	D&B Match Registration Number	Attribute on the customer record to hold the registration number used for matching.
D&B Match Registration Number Type	D&B Match Registration Number Type	Attribute on the customer record to hold the registration number type used for matching.
D&B Match Response Data	D&B JSON Match Data	Attribute on the customer record to hold the raw JSON response data received when matching.
D&B Match State/Territory	D&B Match Territory/State	Attribute on the customer record to hold the state or territory name used for matching.
D&B Match Town Name	D&B Match Primary Town	Attribute on the customer record to hold the primary town name used for matching.
D&B Match URL	D&B Match URL	Attribute on the customer record to hold the URL used for matching.
D&B No Match User Decision	D&B No Match User Decision	Attribute on the customer record to hold the status of 'no match'.
D&B Match Candidate Reference Type	D&B Organization Candidates	Reference type used to link candidates to a D&B Organization record.
D&B Organization Reference Type	D&B Organization	Reference type used to link organization records to their D&B Organization Records.
Edit		

- D&B Match Confidence is a candidate reference attribute that is set by the D&B response (from 1 to 8) based on the quality of the matched record.
- If there were issues with the easy setup and values were not mapped, the issues are displayed within the component model. The Description column provides context for how the object types, reference types, or attributes / data containers are to be used.
- The following Request Attributes (aspect values in the component model) map to the corresponding D&B Attributes:

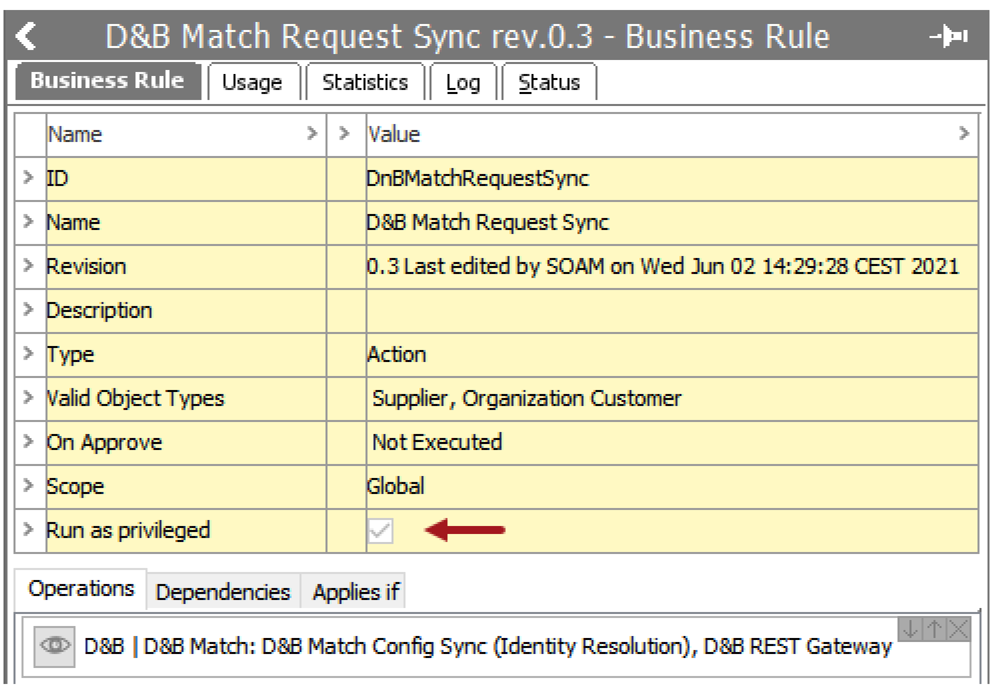
* D&B Match Reason Code is only needed when submitting match / enrichment requests for companies located in Germany. The attribute assignment is blank in the component model by default and must be created if needed.

Request Attributes	D&B Attributes
D&B Match Address Line 1	streetAddressLine1
D&B Match Address Line 2	streetAddressLine2
D&B Match Country Code	countryISOAlpha2Code
D&B Match DUNS Number	duns
D&B Match Organization Name	name
D&B Match Postal Code	postalCode
D&B Match Registration Number	registrationNumbers
D&B Match Registration Number Type	registrationNumberType
D&B Match State/Territory	addressRegion
D&B Match Town Name	addressLocality
D&B Match Phone Number	telephoneNumber
D&B Match Reason Code*	orderReason
D&B Match County	addressCounty
D&B Match Email	email
D&B Match URL	url

10. For all D&B users, on the **D&B Organization Entity Root** (D&B Enrichment Records (DnB Enrichment Records) in the images above), add the 'Delete entity reference' action, for the **D&B Organization Entity Type** (D&B Organization (DnBOrganization)), so that possible D&B Record candidate references can be removed when a single candidate is selected. For more information, see the **D&B Matching** topic.



11. To prevent privilege errors during auto matching, review all D&B business actions and check the 'Run as privileged checkbox. For details about editing business rules, see the **Editing a Business Rule** topic in the **Business Rules** documentation.



12. This completes the setup and users can begin matching existing customer records.

D&B Matching

The Dun & Bradstreet (D&B) data integration allows for users to do a matching request on a customer record. Regardless of the process used, matching involves:

- Receiving match candidates from D&B.
- Running the matching process based on the operation configuration to compare the D&B records and the STEP records.
- Selecting the preferred candidate when more than one candidate is received.

For an **asynchronous** process, the D&B Match Candidate Workflow allows users to keep track of the required matching tasks.

For a **synchronous** process, no workflow is used since matching and enriching is done at the time of onboarding.

Prerequisites

Prior to starting the matching process, the setup for D&B integration must be complete including the Easy Setup process and verifying the D&B Integration Component Model. For details, see the **D&B Integration Configuration and Easy Setup** topic.

Receiving Match Candidates

Values from your customer record are matched against D&B records.

For **synchronous** processing, matching is done as needed, typically while onboarding. No events or workflows are involved. A response is received from D&B and sent back to STEP that writes the data into D&B Organization records.

- If only one match candidate is returned from D&B or if multiple candidates are returned but only one exceeds the Autolink Threshold configured on the 'D&B Match Config Sync (Identity Resolution)' (ID: DnBMatchOpConfigSync) operation configuration, this candidate is automatically selected by creating the D&B Organization reference. The D&B Organization record is then automatically enriched.
- If multiple candidates are returned from D&B, and none or more than two candidates exceed the Autolink Threshold configured on the 'D&B Match Config Sync (Identity Resolution)' (ID: DnBMatchOpConfigSync) operation configuration, the customer record moves to the D&B Match Candidates screen for a user to select a candidate. A user evaluates the D&B records based on designated criteria to make a best-match choice and selects one candidate D&B record to reference. The select candidate process is best completed in Web UI as described below.

  Until a selection is made, 'D&B Match Candidate' references exist from the customer record to the possible D&B Record candidates. These references and non-match candidates are deleted when a match is selected, and the selected D&B record gets a 'D&B Organization' reference to the customer record.

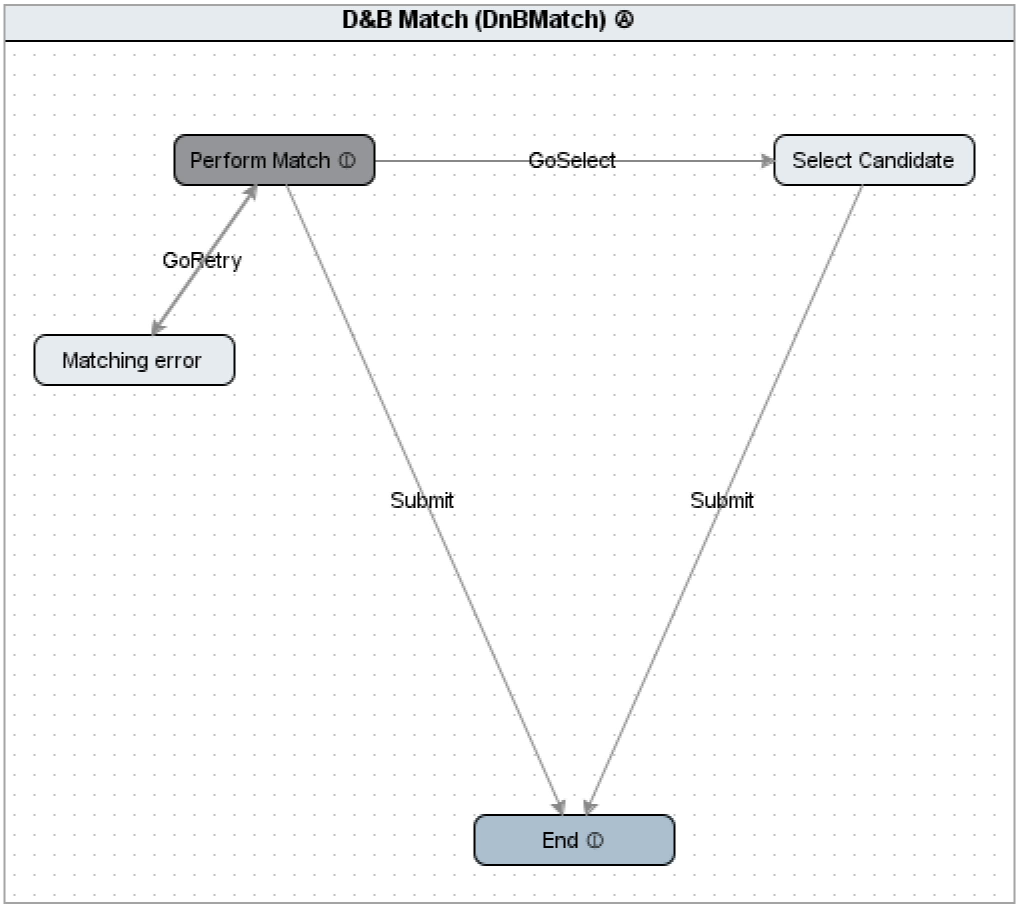
For **asynchronous** processing, matching is started by the initiation of the organization entity (as designated in the D&B Integration component model) into the D&B Match workflow. Standard methods are available for initiation including upon entity creation, through manual initiation, or by a variety of other actions or processes such as business rules, imports, and bulk updates.

Initiating customer records triggers an event that is then processed via an event processor. A business action triggers a call to D&B to match the customer record data to the D&B records. A response is returned to STEP that writes the data into D&B Organization records.

- If only one match candidate is returned from D&B or if multiple candidates are returned but only one exceeds the Autolink Threshold configured on the 'D&B Match Operation (Async)' (ID: DnBMatchOpConfig) operation configuration, this candidate is automatically selected by creating the D&B Organization reference. The customer record then transitions through the D&B Match workflow.
- If multiple candidates are returned from D&B and none or more than two candidates exceed the Autolink Threshold configured on the 'D&B Match Operation (Async)' (ID: DnBMatchOpConfig) operation configuration, the customer record moves to the Select Candidate workflow state. The select candidate process is best completed in Web UI as described below. To complete the Select Candidate task, a user evaluates the D&B records based on designated criteria to make a best-match choice and selects one of those candidate D&B records to reference.

  Until a selection is made, 'D&B Match Candidate' references exist from the customer record to the possible D&B Record candidates.

Important: The asynchronous process creates a queue of events that are performed as scheduled, which means users must allow time for each match request to process.



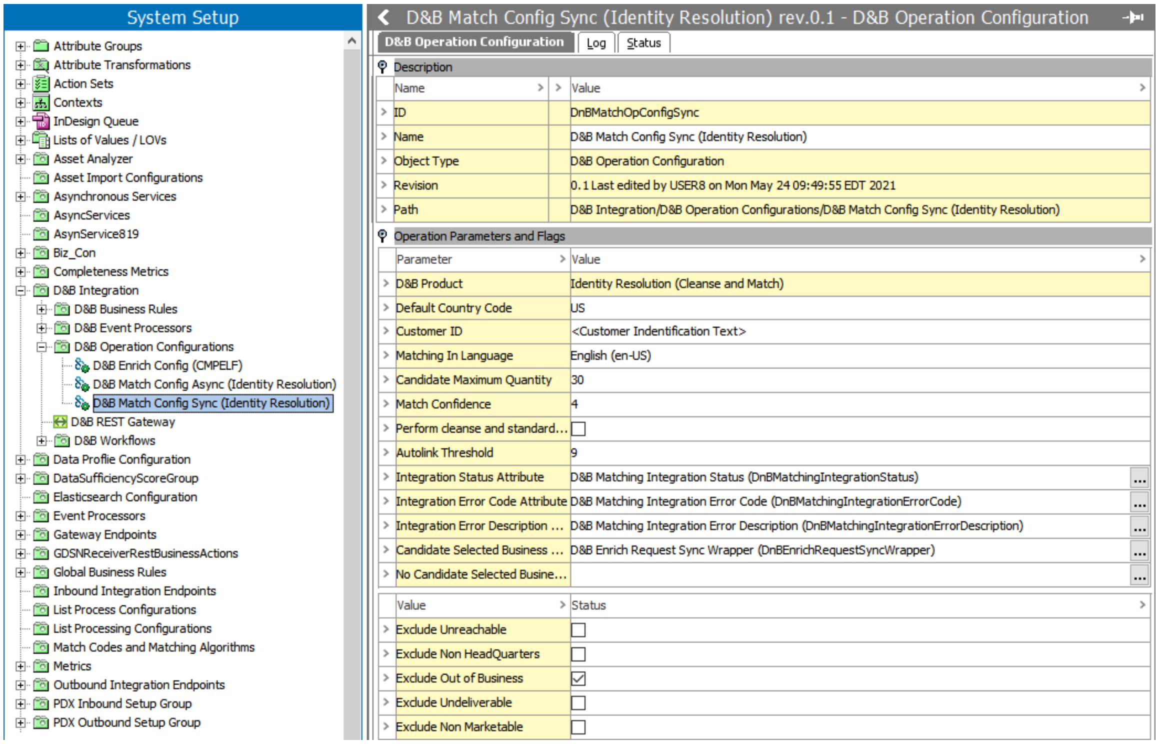
Matching Process

A matching request is comprised of attributes mapped in the D&B Integration component model. As explained in the **Dun & Bradstreet Integration** topic, input for D&B Match requests is taken from a configurable set of attributes on the organization record. Those attribute values are part of the request sent to D&B. More details about configuring the component model can be found in the **D&B Integration Configuration and Easy Setup** topic.

Operation Configuration

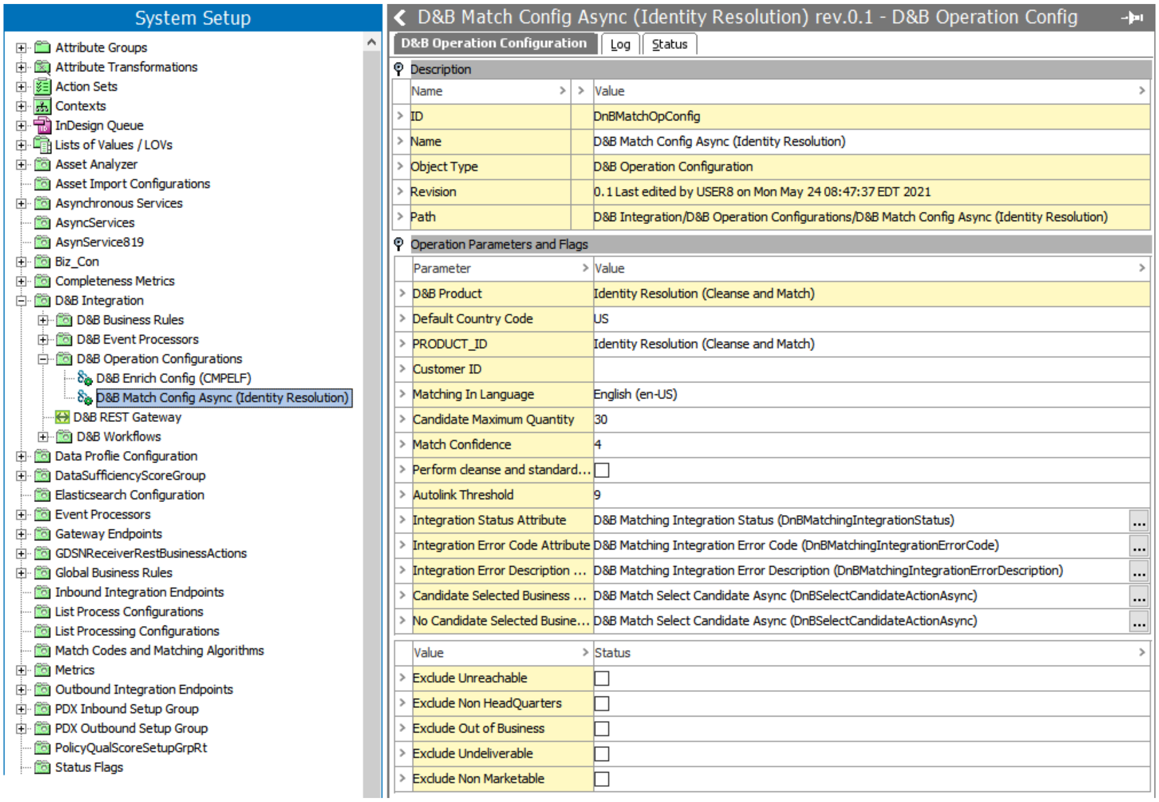
Exclusions can be enabled for both asynchronous and synchronous processing (shown in the images below) via workbench. These options are also honored when doing a 'Modify Search' action in Web UI.

For **synchronous** matching, the recommendation is to use the Web UI. The configuration is 'D&B Match Config Sync (Identity Resolution)' (ID: DnBMatchOpConfigSync).



Note: Matching candidates are based on the D&B Match Request Sync Wrapper created by the Easy Setup in the D&B Integration folder.

For **asynchronous** matching, use the Web UI (recommended) or workbench to start the initiation process. The configuration is 'D&B Match Config Async (Identity Resolution)' (ID: DnBMatchOpConfig).

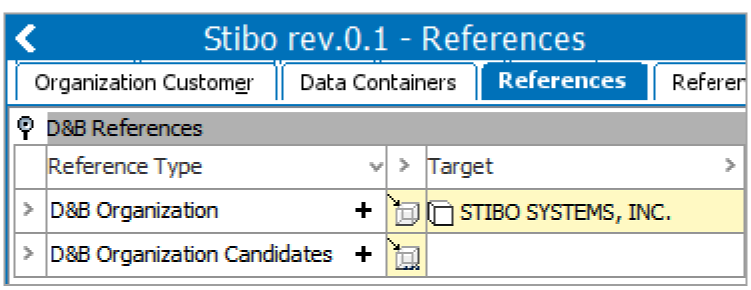


Note: Matching candidates are based on the D&B Match Processor created by the Easy Setup in the D&B folder. For information on maintaining event processors, see the **Event Processors** documentation.

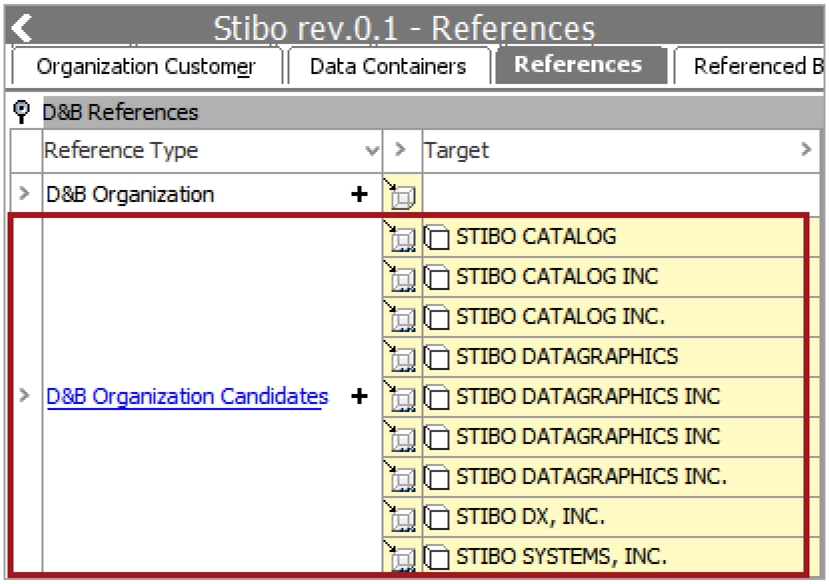
References and Match Candidates

The same references and candidates are used for both **synchronous** and **asynchronous** matching.

When only one match exists, no manual selection is required and the D&B Organization entity (D&B record) is automatically referenced by the organization entity (customer record). Initial setup is handled in the D&B Organization Reference Type mapping in the component model.



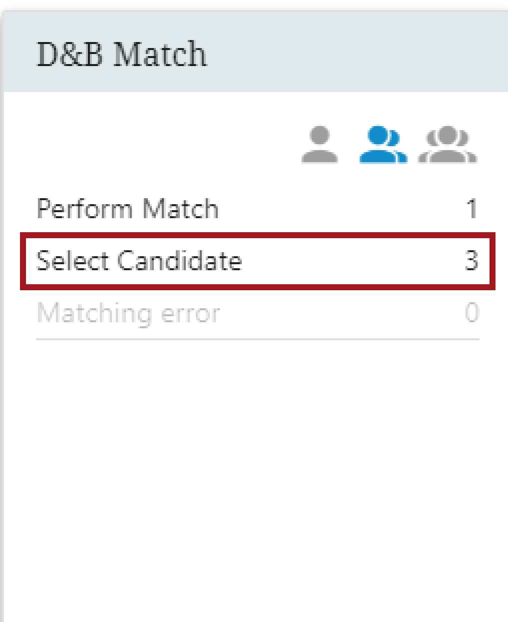
When multiple match candidates are found, a D&B reference is made from the customer record (organization entity) to each candidate record (D&B record / D&B organization entity) based on the D&B Match Candidate Reference Type in the component model.



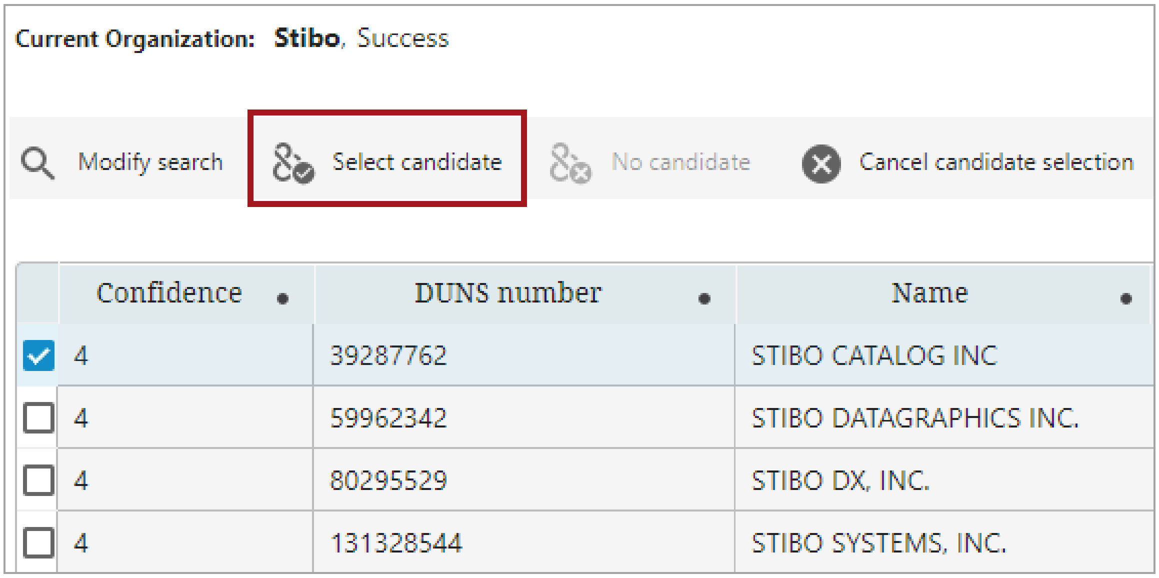
Selecting the Candidate

For organization entities that have multiple match candidates, a single candidate must be manually selected. The Web UI is recommended for this process. Detailed information for configuring and using the Web UI for D&B can be found in the **D&B in Web UI** topic.

For **asynchronous** processing only, selection tasks are available in a Status Selector Homepage Widget like the one shown below. This is not used in synchronous processing since no workflows are involved.



For both **synchronous** and **asynchronous** processing types, the following task list actions are available on the D&B Match Candidates Screen. Separate screens must be configured based on the desired processing type.



- Select a row and click the **Select candidate** action button to manually select the preferred candidate. The customer record and the D&B record are linked via the 'D&B Organization Reference Type' mapping in the component model.
- With no rows selected, click the **No candidate** button if no satisfactory candidate can be found even after using the 'Modify search' functionality. The component model setting for 'D&B No Match User Decision' attribute is set to 'true'.

Important: Matching errors must be handled before the record can be enriched. See the **D&B Error Handling with Integration Status** topic for more information.

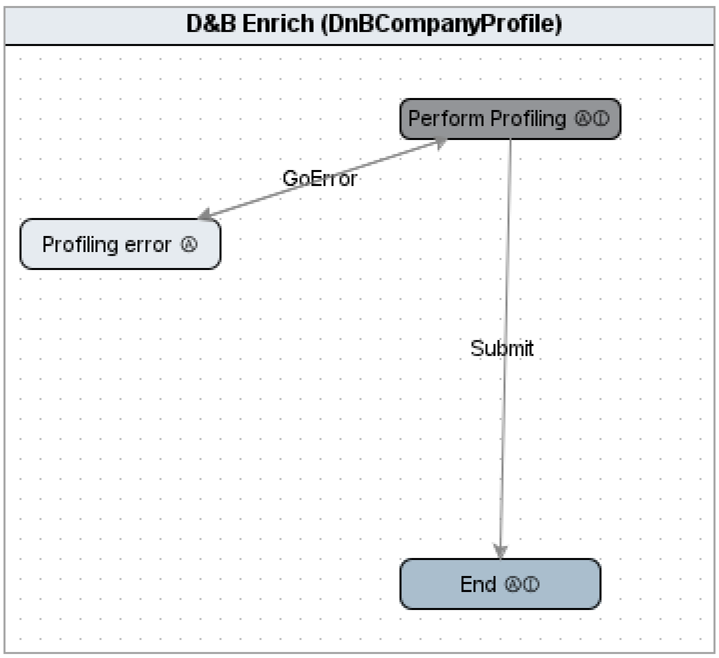
D&B Enriching

Dun & Bradstreet (D&B) integration enrichment happens for both asynchronous and synchronous processing with slight differences in the timing.

Asynchronous Processing

A customer entity must be linked to a D&B record to perform the asynchronous 'D&B Enrich' (ID:DnBCompanyProfile) enrichment workflow created by the D&B Easy Setup.

Initiation can be started from the workbench or Web UI. When entities are initiated into this workflow, an event is sent to the D&B Enrich Processor' (ID: DnBCompanyProfileProcessor). The event processor runs a business action that queries additional data from the D&B Enrich service, writes this information to the D&B Organization entity, and completes the workflow. For information on maintaining event processors, see the **Event Processors** documentation.



Important: Profiling errors that occur during the process must be handled via the workflow 'Profiling error' state. See the **D&B Error Handling with Integration Status** topic for more information.

Synchronous Processing

To perform synchronous D&B enrichment, the customer entity must be linked to a D&B record.

Enrichment is automatically performed immediately following the matching process. No workflow is required, nor are additional action buttons needed for enriching via the Web UI. The 'D&B Match Request Sync Wrapper' (ID: DnBMatchRequestSyncWrapper) completes both matching and enriching.

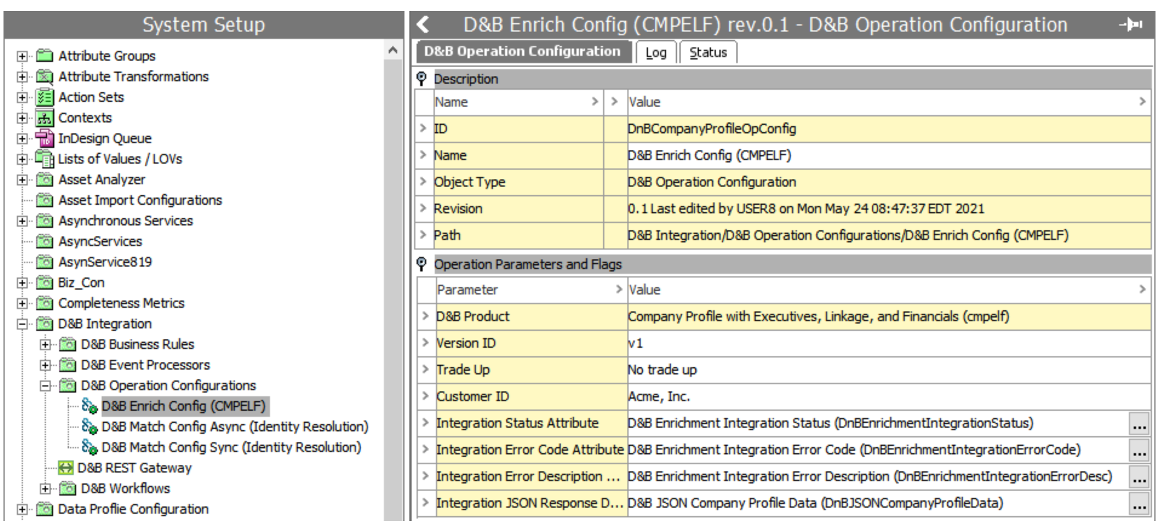
It is not necessary to initiate the enrichment process since it is a continuation of the synchronous matching process. However, to if manual initiation is desired, the recommendation is to use the Web UI.

Operation Configuration

For both **synchronous** and **asynchronous** processing, the following example of a D&B enrichment product configuration of CMPELF is shown below.

See Company Profile with Executives, Linkage, and Financials (cmpelf) for a list and description of the query parameters.

- **D&B Product**: Company Profile with Executives, Linkage, and Financials (cmpelf)
- **Version ID**: v1
- **Trade Up**: No trade up
- **Customer ID**: Acme, Inc.
- **Integration Status Attribute**: 'D&B Enrichment Integration Status' (ID: DnBEnrichmentIntegrationStatus)
- **Integration Error Code Attribute**: 'D&B Enrichment Integration Error Code' (ID: DnBEnrichmentIntegrationErrorCode)
- **Integration Error Description Attribute**: 'D&B Enrichment Integration Error Description' (ID: DnBEnrichmentIntegrationErrorDesc)
- **Integration JSON Response Data Attribute**: 'D&B JSON Company Profile Data' (ID: DnBJSONCompanyProfileData)



For information on the matching process, see the **D&B Matching** topic and the **D&B in Web UI** topic.

D&B in Web UI

The Dun & Bradstreet (D&B) integration allows customers to use the Web UI for matching and enriching once all setup is complete.

Asynchronous and synchronous D&B processing is available, and the configuration required for both are included in this topic.

Prerequisites

The D&B Web UI integration requires an understanding of the Web UI, how to create screens and widgets, and the necessary user permissions to do so. See the **Web UI Getting Started** documentation for more information on these topics.

Additionally, users should be familiar with the D&B integration setup and processes. For more information, see the **Dun & Bradstreet Integration** topic.

Asynchronous

For asynchronous processing, the following Web UI configuration is needed:

- **Status Selector Homepage Widget** or a **Status Selector on Global Navigation Panel** for the D&B Match Candidate workflow. See the **D&B Status Selector** topic.
- **D&B Candidate Matching** screen for an organization entity (customer record) with multiple D&B organization entity (D&B record) match candidates. See the **D&B Asynchronous Match Candidates Screen** topic.
- **Action buttons** to manually start the enrichment process via the D&B Enrich workflow. See the **D&B Match and Enrich Action Buttons** topic.

Synchronous

For synchronous processing, customer records enter the D&B Match workflow via standard methods such as in bulk or individually via Web UI or STEP Workbench. The following Web UI configuration is needed:

- No status selector is required since workflows are not used.
- **D&B Candidate Matching** screen for an organization entity (customer record) with multiple D&B organization entity (D&B record) match candidates. See the **D&B Synchronous Match Candidates Screen** topic.
- **Action button** to manually starting the match process (and optionally one to start enrichment). See the **D&B Match and Enrich Action Buttons** topic.

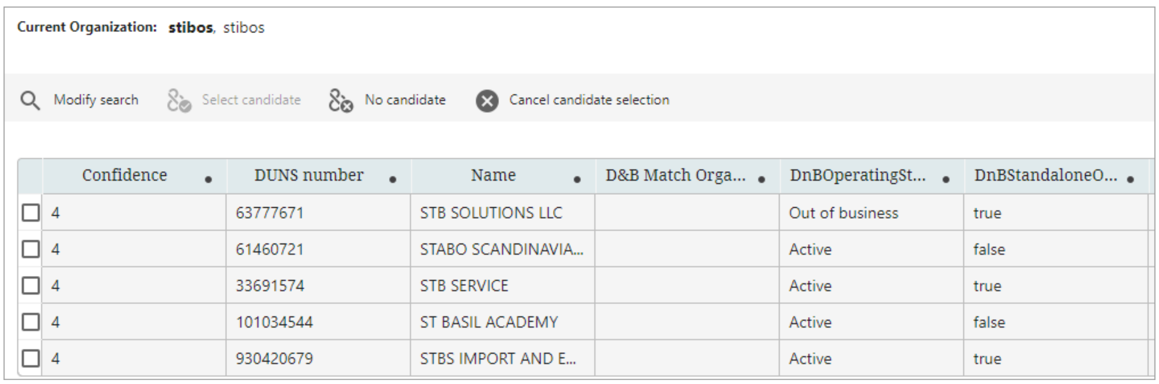
D&B Asynchronous Match Candidates Screen

The Dun & Bradstreet (D&B) integration allows customers to use the Web UI for matching and enriching once all setup is complete. For additional D&B Web UI configuration, see the **D&B in Web UI** topic.

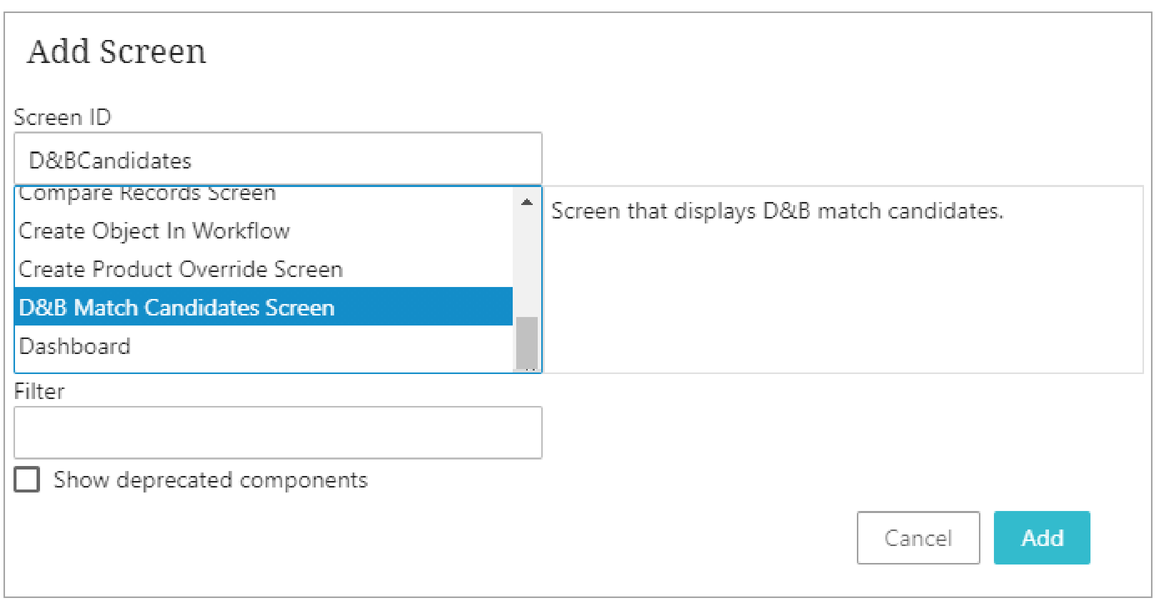
Important: Separate screens are required for asynchronous and synchronous processing.

Configure Asynchronous D&B Matching and Enriching

Follow the steps below to create a new Select Candidate Screen for asynchronous matching.

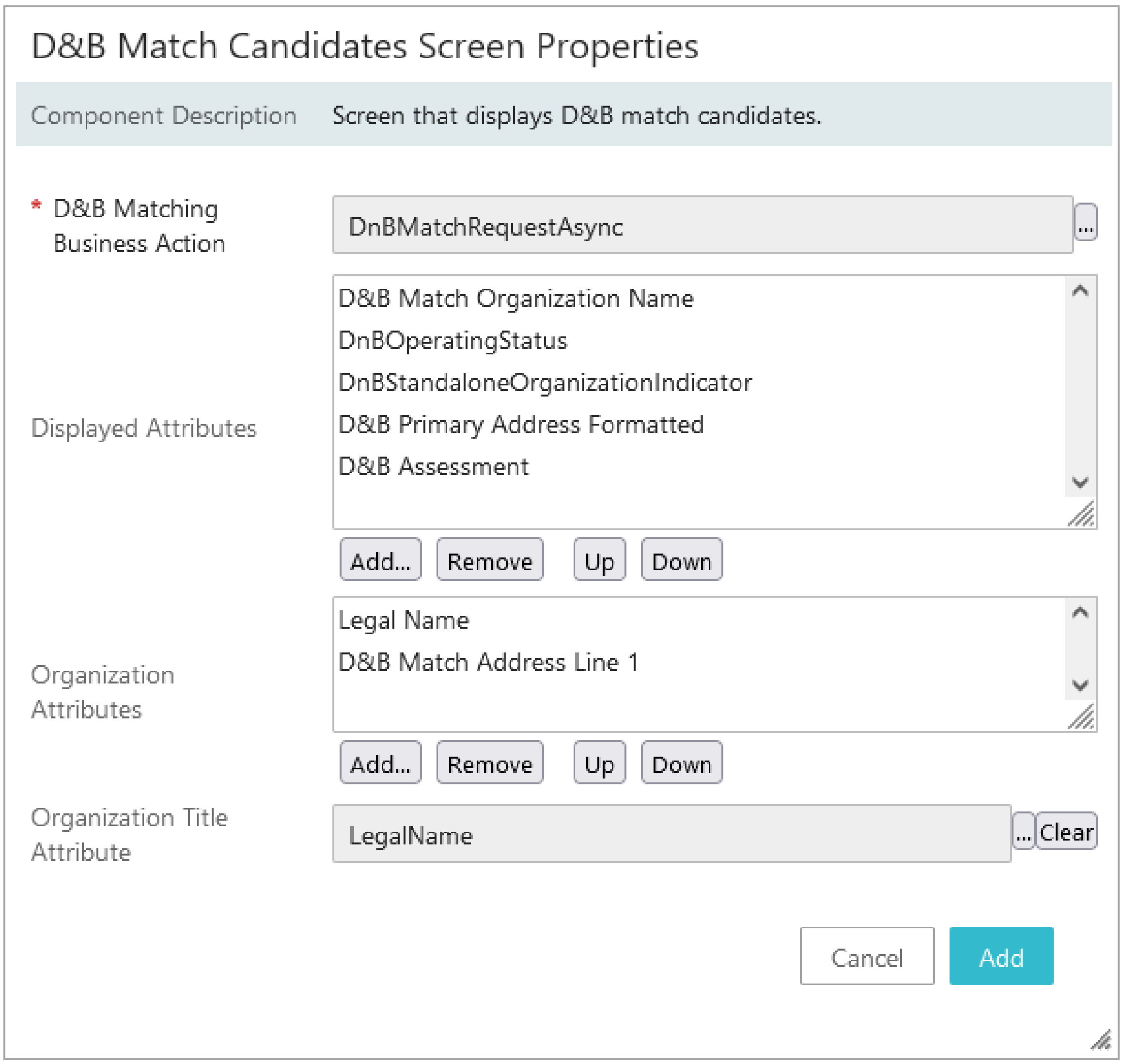


1. From the Web UI Designer, select **New**.
2. On the Add Screen dialog, select the **D&B Match Candidates Screen** option from the list. For **Screen ID**, add a name that is descriptive and distinguishable, such as D&BCandidates. Click the **Add** button.



3. On the **D&B Match Candidates Screen Properties** dialog:

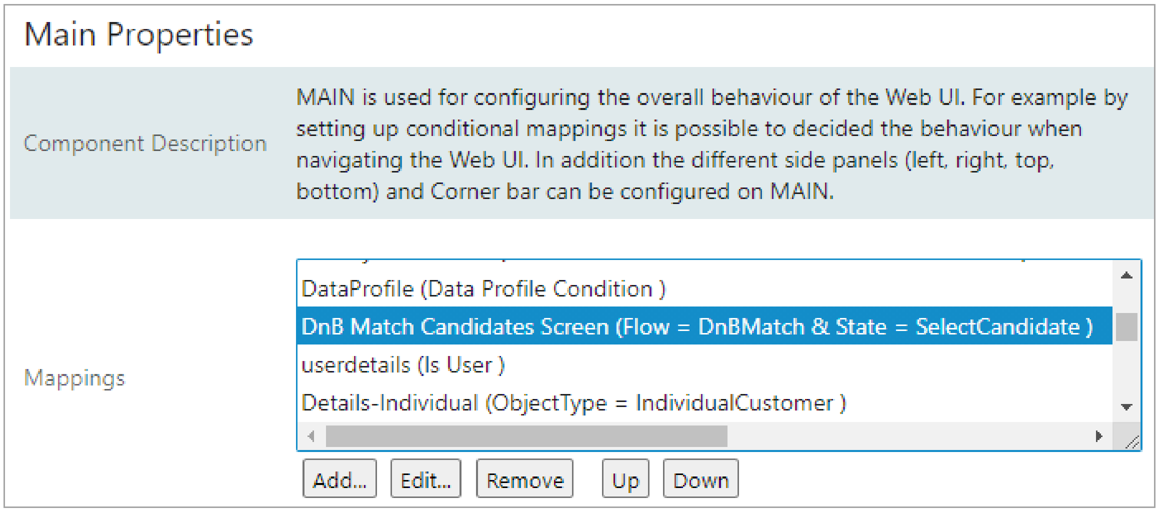
   - For the required **D&B Matching Business Action** parameter, click the ellipsis button (**...**) and select the **D&B Match Request Async (ID: DnBMatchRequestAsync)** business action. This business action performs D&B matching.
   - For the **Displayed Attributes** parameter, click the **Add** button and select the titles of the attribute columns that will be included (along with the default columns for Confidence score, DUNS number, and Name of the D&B record) in the task list screen.
   - For the **Organization Attributes** parameter, click the **Add** button and select the organization entity to appear above the task list for the current object.
   - For the **Organization Title Attribute** parameter, click the ellipsis button (**...**) and make a selection.



4. Click **Add** to complete the configuration or click **Cancel** to exit without saving changes.

5. Click **Save** to commit the changes to the D&B Match Candidates Screen Properties screen.

6. Map a Workflow Condition in the Main Properties to configure access to the D&B Match Candidates Screen. Create a Workflow Condition for each state of the D&B workflow and select the screen you just created for the Screen Mapping Properties. Setup details are included in the **Mapping Workflow States in Web UI** topic.



7. Configure the action button(s) as defined in the **D&B Match and Enrich Action Buttons** topic.

D&B Synchronous Match Candidates Screen

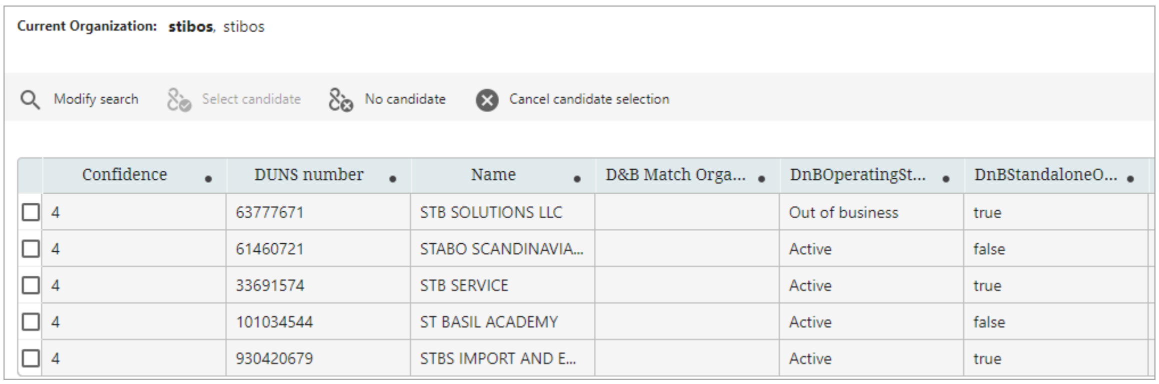
The Dun & Bradstreet (D&B) integration allows customers to use the Web UI for matching and enriching once all setup is complete. For additional D&B Web UI configuration, see the **D&B in Web UI** topic.

The D&B synchronous Match Candidates Screen automatically displays after matching is done. The process is described in the **D&B Matching** topic.

Important: Separate screens are required for asynchronous and synchronous processing.

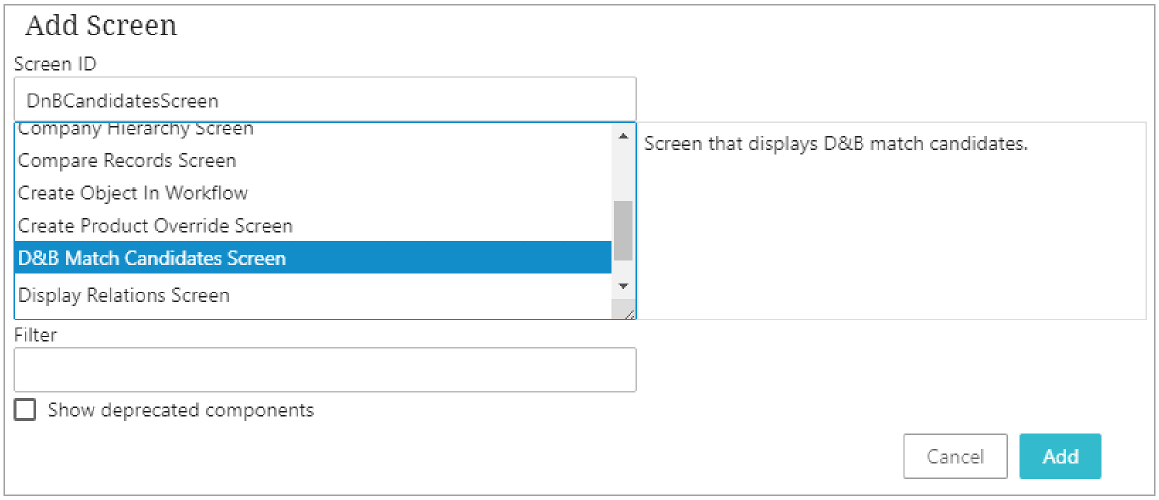
Configure Synchronous D&B Matching and Enriching

Follow the steps below to create a new Select Candidate Screen for synchronous matching.



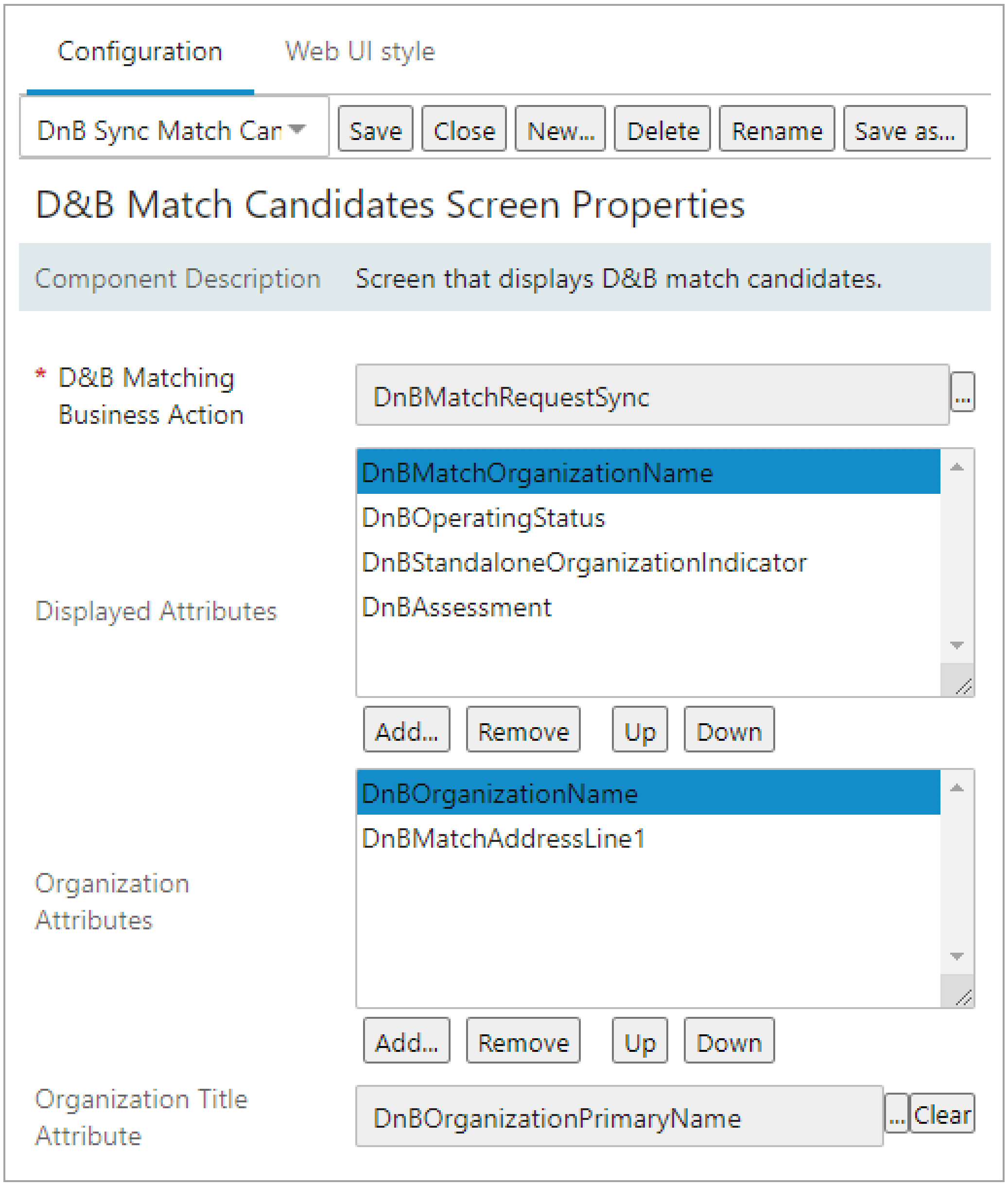
1. From the Web UI Designer, select **New**.
2. From the Add Screen prompt, select the **D&B Match Candidates Screen** option from the list. For **Screen ID**, add the required screen ID: DnBCandidatesScreenSync. Click the **Add** button.

   Note: The screen ID must match the 'navigate to ID' code used in the JavaScript for the business action 'D&B Match Request Sync Wrapper (DnBMatchRequestSyncWrapper)' as defined in the **D&B Integration Configuration and Easy Setup** topic.



3. On the **D&B Match Candidates Screen Properties** dialog:

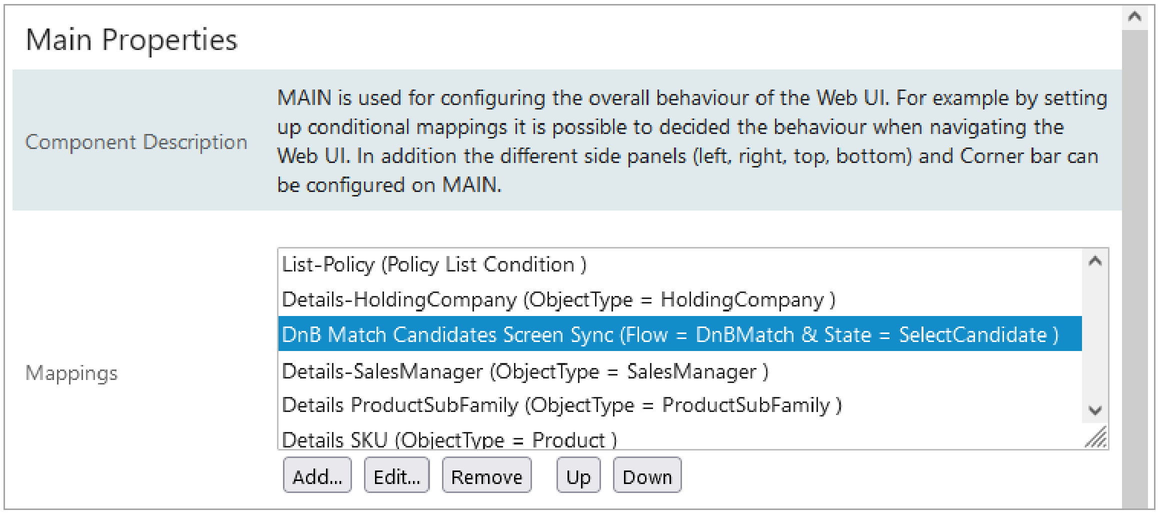
   - For the required **D&B Matching Business Action** parameter, click the ellipsis button (**...**) and select the **D&B Match Request Sync (ID: DnBMatchRequestSync)** business action. This business action performs D&B matching.
   - For the **Displayed Attributes** parameter, click the **Add** button and select the titles of the attribute columns that will be included (along with the default columns for Confidence score, DUNS number, and Name of the D&B record) in the task list screen.
   - For the **Organization Attributes** parameter, click the **Add** button and select the organization entity to appear above the task list for the current object.
   - For the **Organization Title Attribute** parameter, click the ellipsis button (**...**) and make a selection.



4. Click **Add** to complete the configuration.

5. Click **Save** to commit the changes to the D&B Match Candidates Screen Properties screen.

6. Map a Workflow Condition within the mapping field in Main Properties to configure how a user will access the D&B Match Candidates Screen. Create a Workflow Condition for each state of the D&B workflow and select the screen you just created for the Screen Mapping Properties. Details on how to do this can be found in the **Mapping Workflow States in Web UI** topic of the **Workflows in Web UI** documentation.



7. Click **Save** on the Main screen and then **Close** to exit design mode.
8. Configure the action button(s) as defined in the **D&B Match and Enrich Action Buttons** topic.

D&B Status Selector

The Dun & Bradstreet (D&B) integration allows customers to use the Web UI for matching and enriching once all setup is complete. For additional D&B Web UI configuration, see the **D&B in Web UI** topic.

For **asynchronous** processing only, to process a customer record for matching to D&B data, the record must go through the D&B Integration Matching workflow. A Status Selector widget gives access to tasks for this workflow.

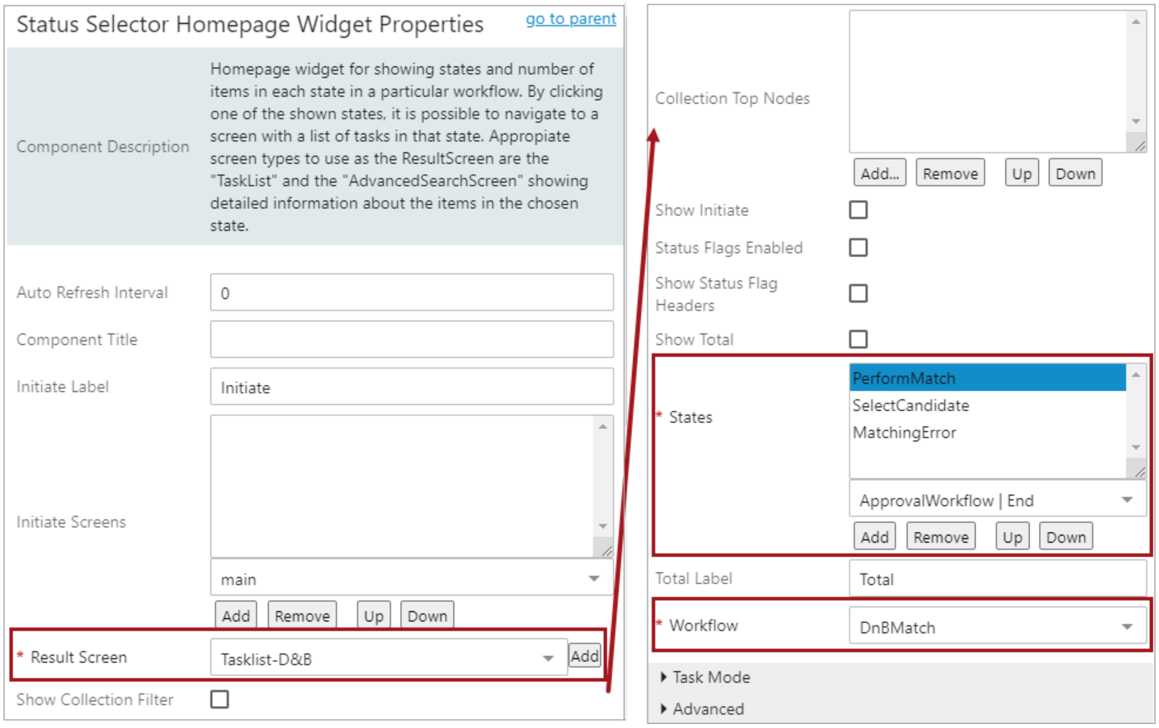
Configure a D&B Workflow Status Selector Widget

Use these steps to configure a **Status Selector Homepage Widget** and/or a **Status Selector on Global Navigation Panel** for the D&B Match Candidate workflow.

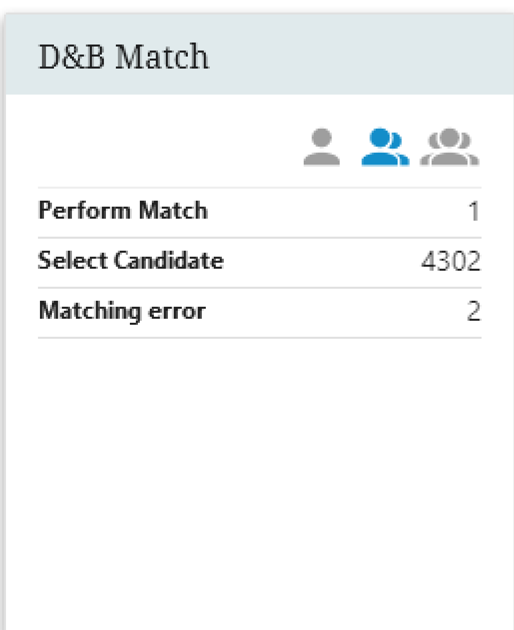
1. Set up the desired status selector(s) as defined in the following **Web User Interfaces / Web UI Getting Started** topics:
   - Status Selector Homepage Widget
   - Status Selector on Global Navigation Panel

   Additional setup information is included in the **Screen / Component Configuration and Mappings for Workflows** documentation.
2. Add the required parameters for the Properties dialog:
   - **Result Screen**: select the screen ID for the Task List screen to be used.
   - **States**: add the states for the applicable workflow.
   - **Workflow**: select the 'DnBMatch Candidate' workflow.
3. Configure additional parameters as desired.
4. Click **Save** and then **Close** to exit the Web UI Designer.

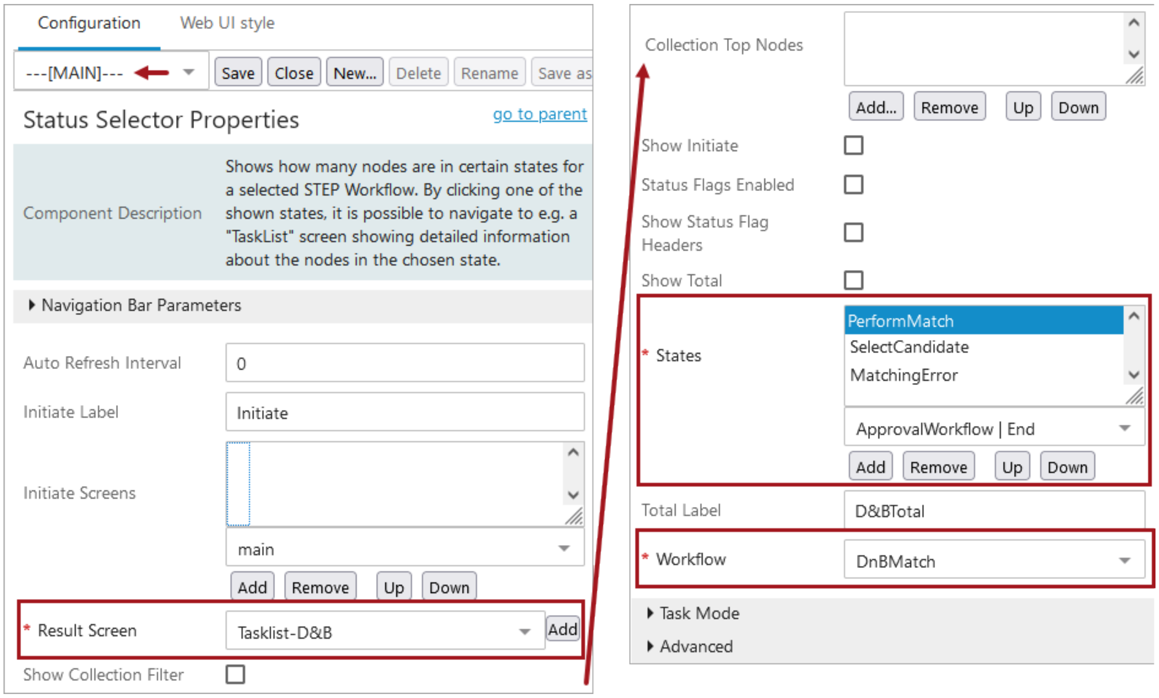
For this Status Selector Homepage widget example, the 'DnBMatch Candidate' workflow is added to the Web UI Homepage. (The single Properties dialog below has been split for easier display.)



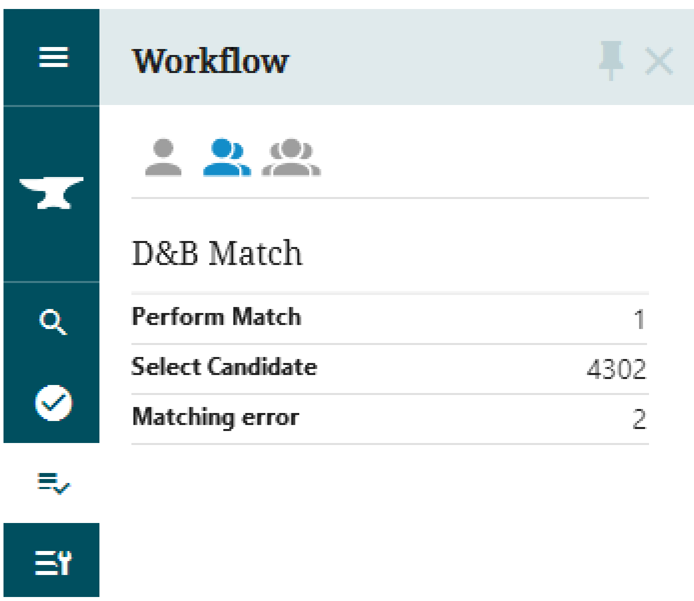
After the Web UI refreshes, the widget displays on the home screen of the Web UI.



For this Status Selector on Global Navigation Panel example, on MAIN in the Left child component, the Global Navigation Panel is updated with a Status Selector for the 'DnBMatch Candidate' workflow. (The single Properties dialog below has been split for easier display.)



After the Web UI refreshes, the workflow status selector displays on the global navigation panel of the Web UI.



D&B Match and Enrich Action Buttons

The Dun & Bradstreet (D&B) integration allows customers to use the Web UI for matching and enriching once all setup is complete. For additional D&B Web UI configuration, see the **D&B in Web UI** topic.

Action buttons enable users to manually initiate the matching and/or enriching processes.

Recommendations

Different action buttons are recommended based on the matching and/or enriching processing types.

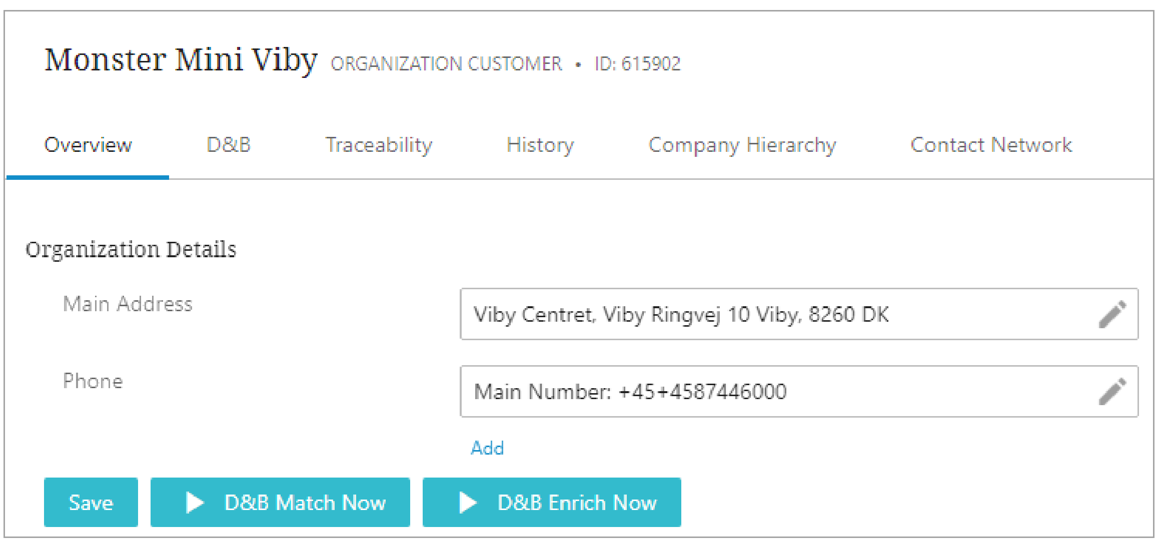
Synchronous Processing

In this scenario, enrichment happens automatically upon completion of matching, that is, when an automatic link is added (by the 'D&B Match Request Sync Wrapper (DnBMatchRequestSyncWrapper)' JavaScript business rule) or when a candidate is manually selected from the Synchronous Match Candidates screen.

For more information, see the 'Candidate Selected Business Action' parameter on the 'D&B Match Config Sync (Identity Resolution) (DnBMatchOpConfigSync)' configuration in the Operation Configurations section of the **D&B Matching** topic.

The following action buttons can be useful:

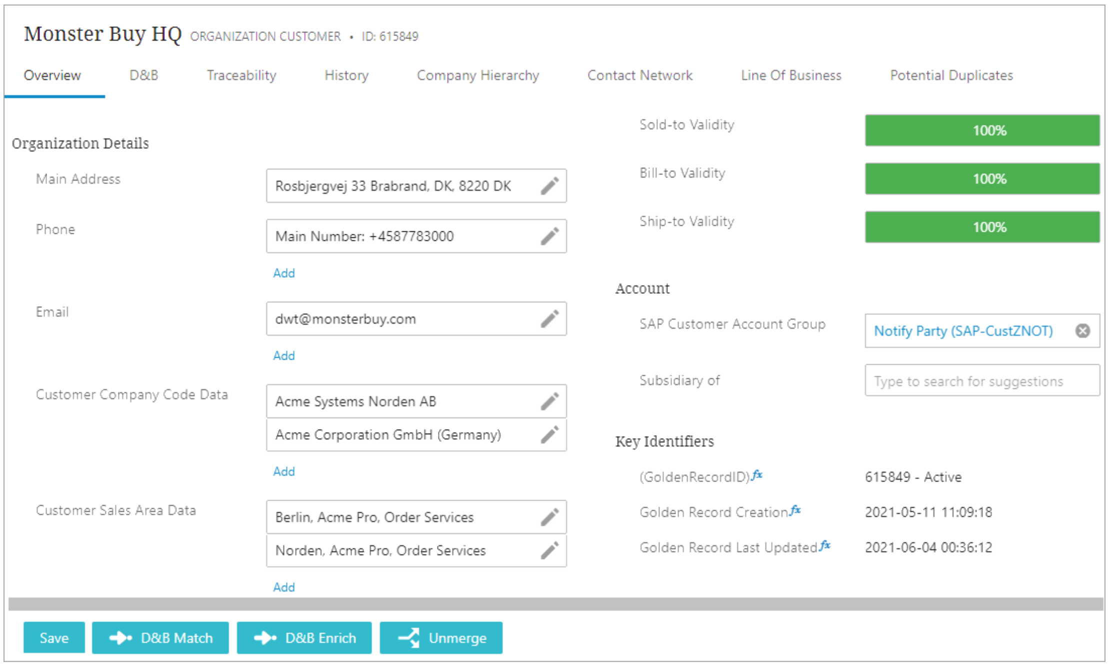
- On a Node Details screen, add a **match button** to start the matching process while viewing an individual record.
- On a Node Details screen, optionally add an enrich button. This is not required since enriching happens automatically after matching.



Asynchronous Processing

In this scenario, records are generally initiated into the matching and/or enriching workflows in bulk. However, an individual record can be initiated via action buttons. Additionally, configuration in workbench can initiate objects into the workflow using a bulk update or an automated business action rule.

- On a Node Details screen, add a **match button** and an **enrich button** to initiate customer records individually.

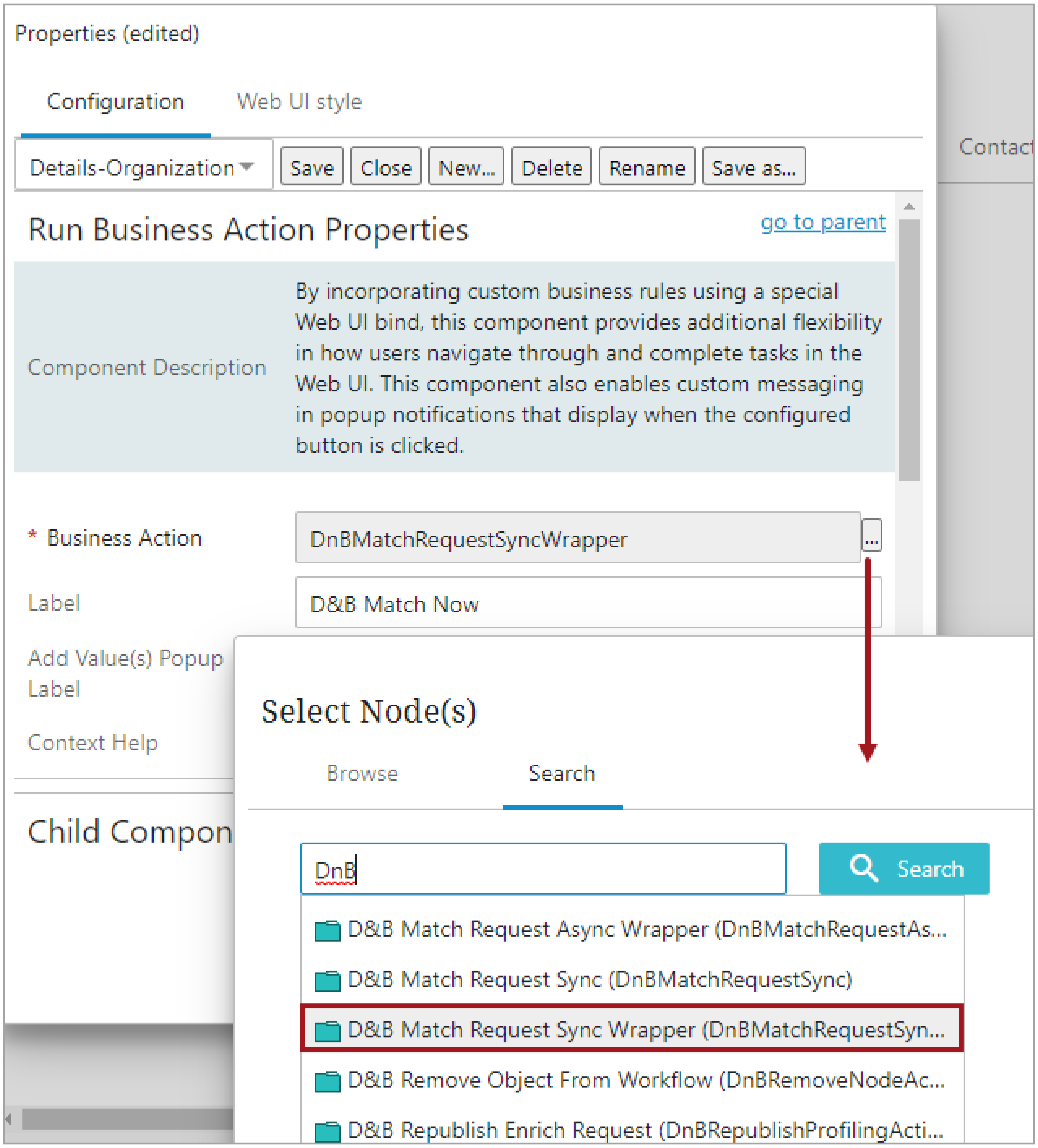


- On a Task List screen, add a **match button** to start the process and optionally add an enrich button to push a task through the workflow.

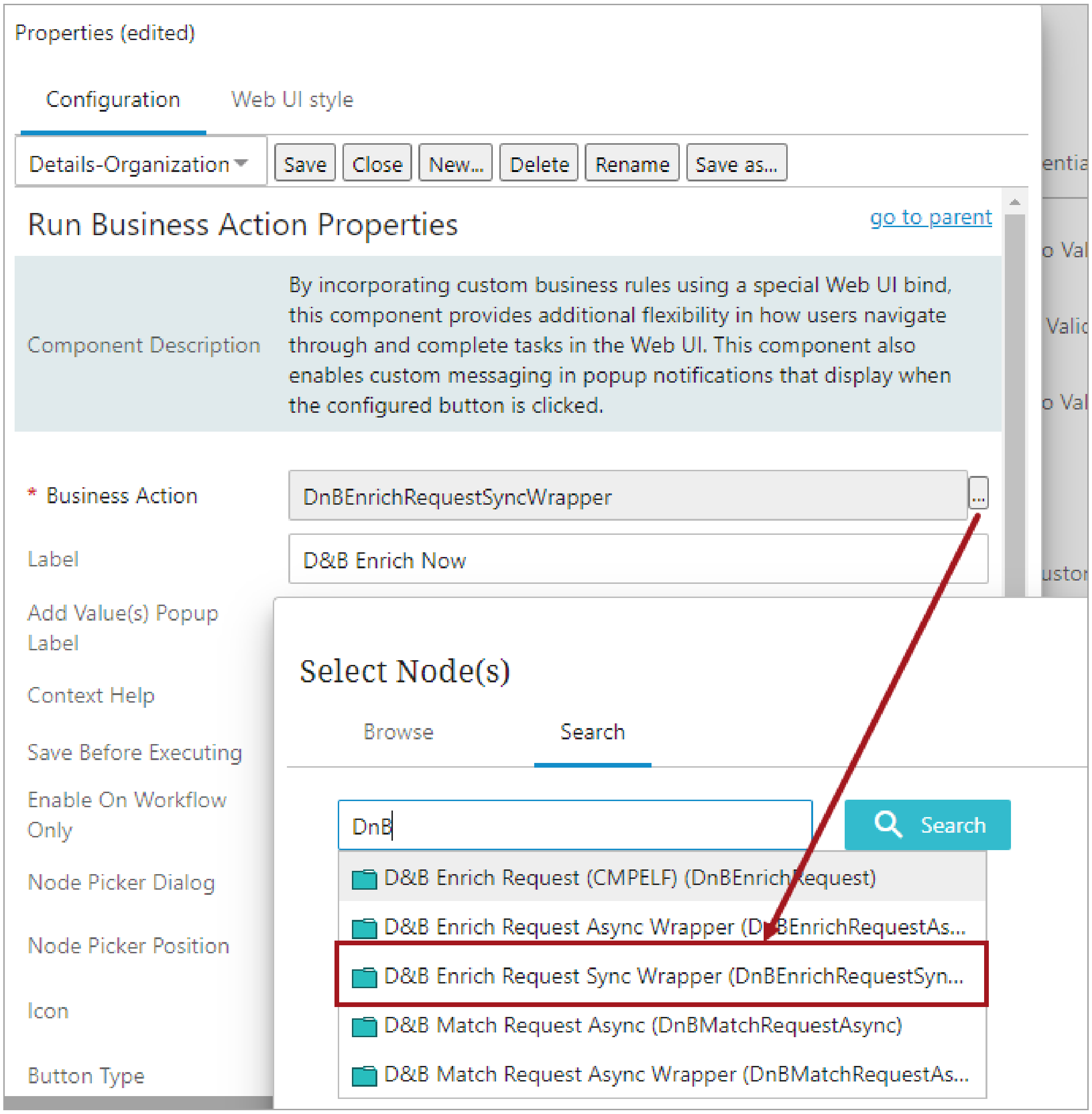
Configure Action Buttons for Synchronous Processing

The recommended buttons can be configured using the steps below.

1. In Web UI design mode, open your organization's **Node Details** screen or on a screen that is part of your customer organization onboarding process.
2. For matching, add a Run Business Action button for the 'D&B Match Request Sync Wrapper (DnBMatchRequestSyncWrapper)' business action. Add a descriptive label, such as 'D&B Match Now'.



3. For enriching, optionally add a Run Business Action button for the 'D&B Enrich Request Sync Wrapper (DnBEnrichRequestSyncWrapper)' business action. Add a descriptive label, such as 'D&B Enrich Now'.

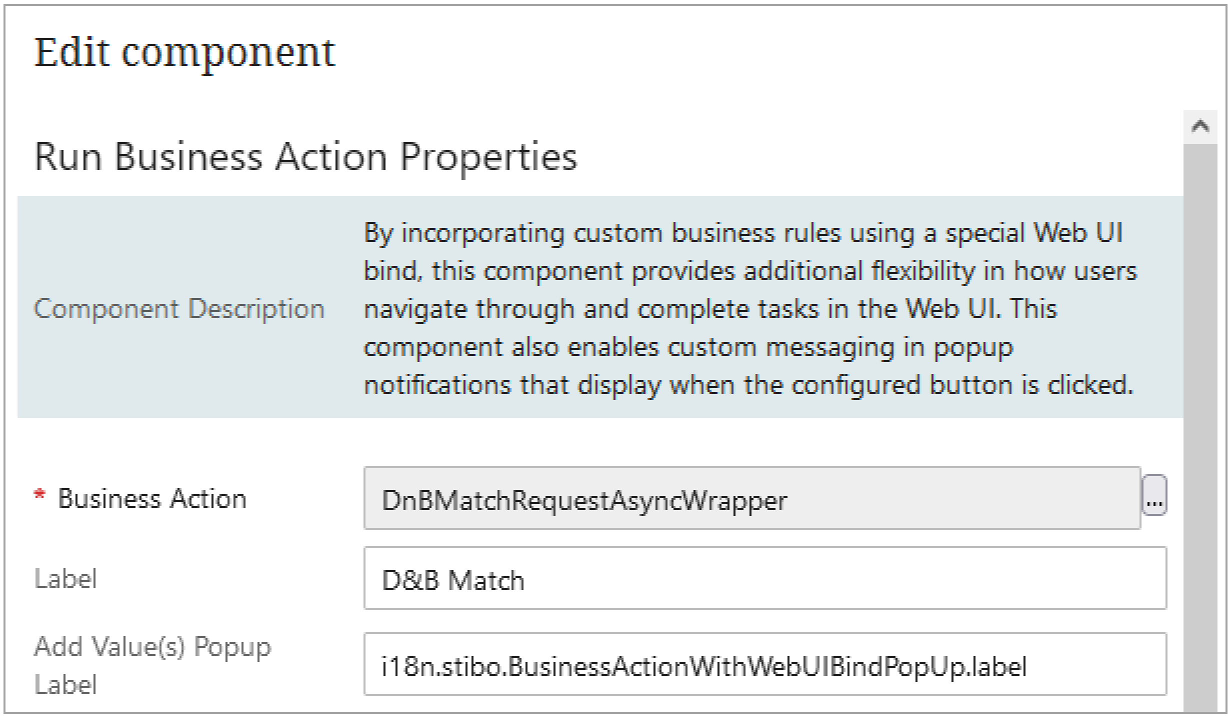


4. Click **Save** and then **Close** to exit the Web UI Designer.

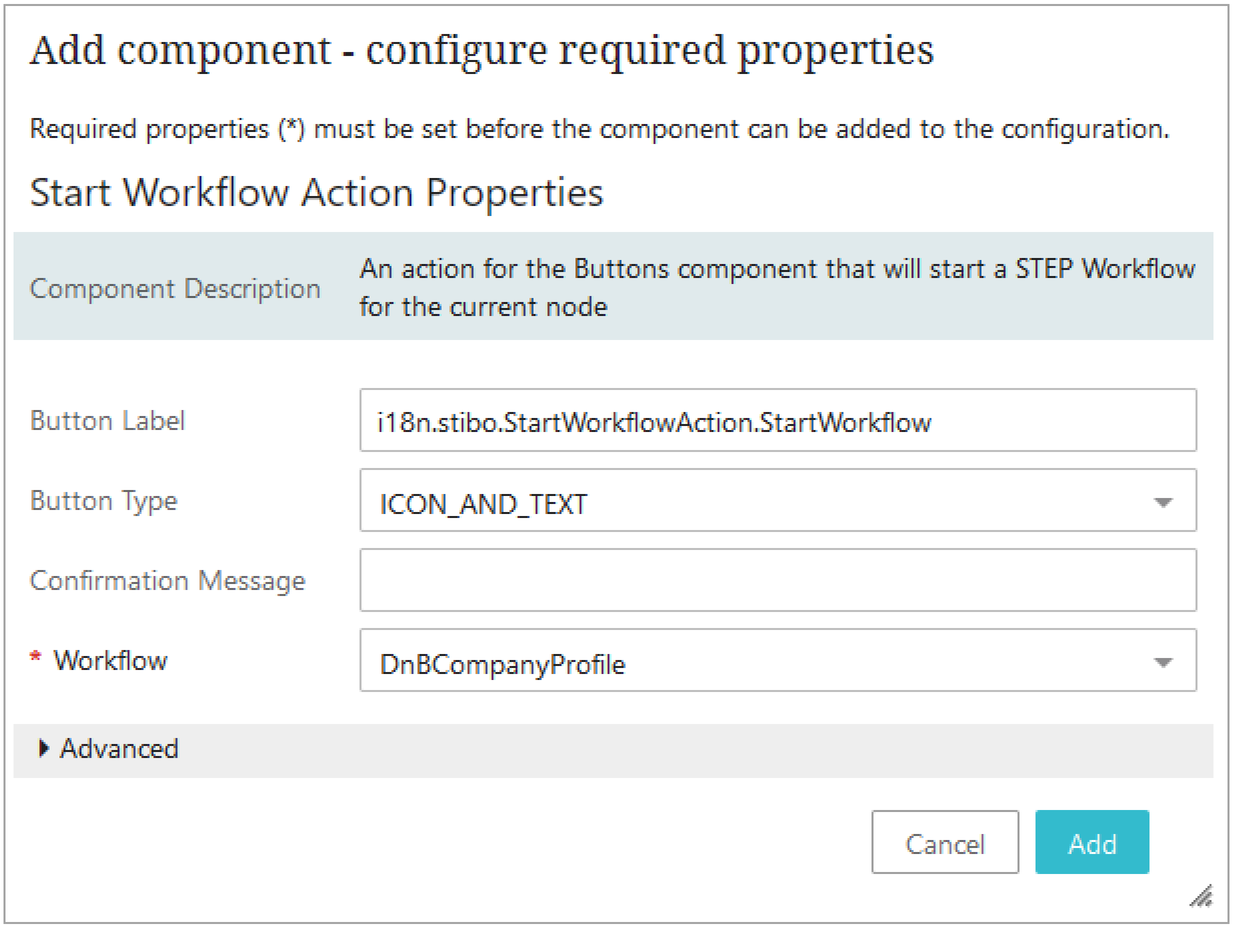
Configure Action Buttons for Asynchronous Processing

1. In Web UI design mode, open your organization's **Node Details** screen or on a screen that is part of your customer organization onboarding process.

   - For matching, add a Run Business Action button for the 'D&B Match Request Async Wrapper (DnBMatchRequestAsyncWrapper)' business action. Add a descriptive label, such as 'D&B Match'.

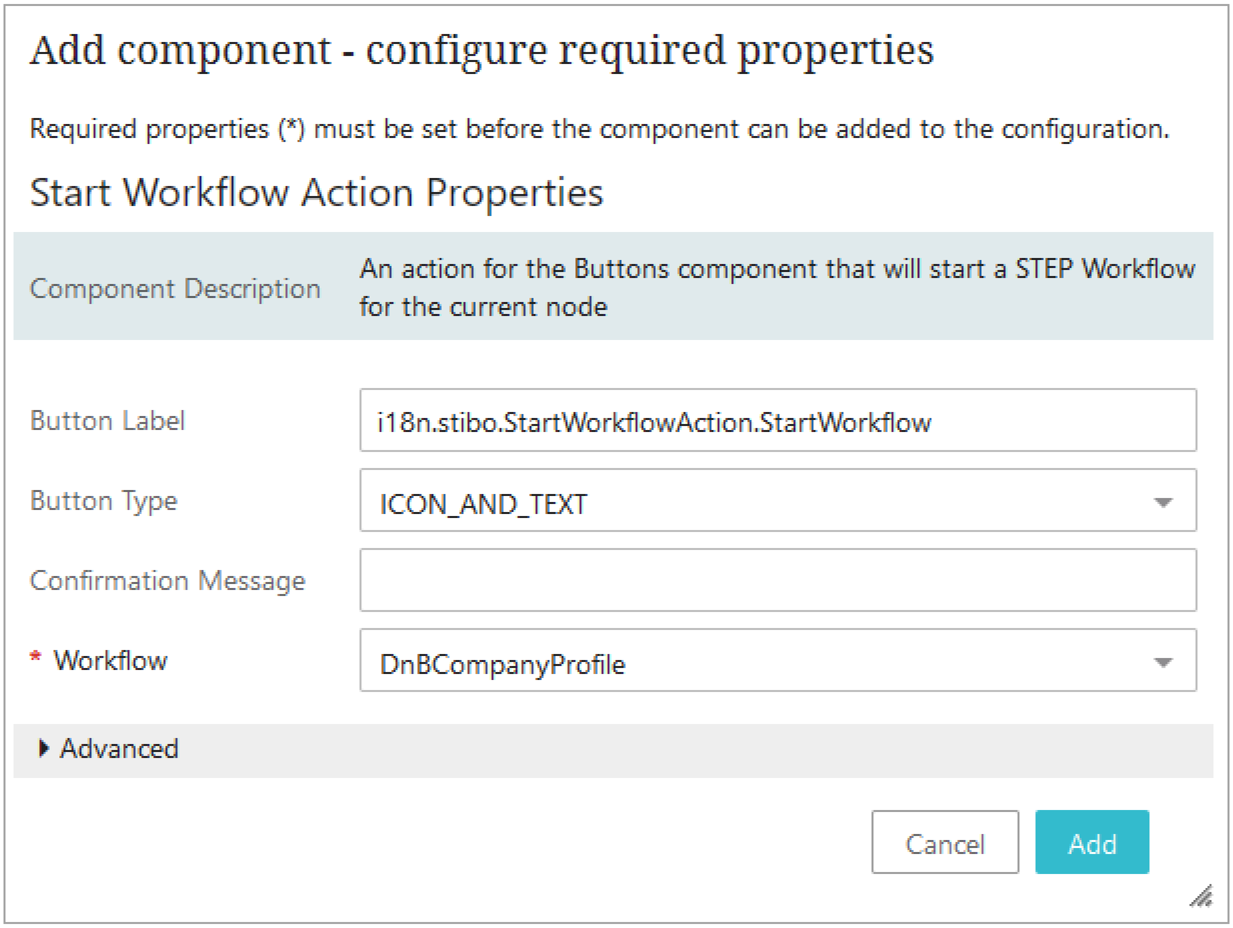


- For enriching, add a Start Workflow Action button for the 'D&B Enrich (DnBCompanyProfile)' workflow. Add a descriptive label, such as 'D&B Enrich'.



2. Open your organization's **Task List** screen or on a screen that is part of your customer organization onboarding process.

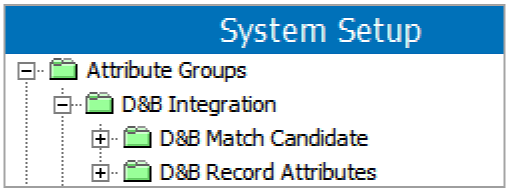
   - For enriching, add a Start Workflow From Grid action button for the 'D&B Enrich (DnBCompanyProfile)' workflow. Add a descriptive label, such as 'D&B Enrich'.



3. Click **Save** and then **Close** to exit the Web UI Designer.

D&B Attributes and Data Containers

As part of the Dun & Bradstreet (D&B) integration Easy Setup, D&B-specific attribute groups, attributes, and data containers are created. The attributes and data containers are applicable to everything related to the D&B integration: business rules, event processors, mapping and operations configurations, workflows, component model, D&B records, and STEP customer records.



Note: Not all LOV values are populated during Easy Setup. Entries are added runtime during the enrichment process.

D&B Organization Entity - Matching - Attributes and Mapping

The following are STEP attributes and the corresponding D&B API JSON path(s) on the 'D&B Organization' (ID: DnBOrganization) entity used for D&B Matching.

Attribute ID	JSON Path, Matching
DnBDUNSNumber	matchCandidates.organization.duns
DnBFamilyTreeMemberRole	matchCandidates.organization.corporateLinkage.familytreeRolesPlayed.description matchCandidates.organization.corporateLinkage.familytreeRolesPlayed.dnbCode
DnBIsMailUndeliverable	matchCandidates.organization.dunsControlStatus.isMailUndeliverable
DnBOperatingStatus	matchCandidates.organization.dunsControlStatus.operatingStatus.description matchCandidates.organization.dunsControlStatus.operatingStatus.dnbCode
DnBOrganizationName	matchCandidates.organization.primaryName
DnBStandaloneOrganizationIndicator	matchCandidates.organization.isStandalone
DnBWebPageAddress	matchCandidates.organization.websiteAddress.url

D&B Organization References - Matching - Data Containers and Mapping

The following are STEP attributes and corresponding D&B API JSON path(s) on the 'D&B candidates' (ID: DnBOrganizationCandidates) and 'D&B selected candidate' (ID: DnBOrganization) references used for D&B Matching.

Attribute ID	JSON Path, Matching
DnBMatchConfidence	matchCandidates.matchQualityInformation.confidenceCode
DnBMatchGrade	matchCandidates.matchQualityInformation.matchGrade

D&B Organization Entity - Matching - Data Containers and Mapping

The following are STEP data containers, attributes, and their D&B API JSON path on the 'D&B Organization' (ID: DnBOrganization) entity used for D&B Matching.

This table is best viewed in online help.

Data Container ID	Attribute ID	JSON Path, Matching
DnBTradeStyleName	DnBOrganizationName	matchCandidates.organization.tradeStyleNames.name
	DnBDisplaySequence	matchCandidates.organization.tradeStyleNames.priority
DnBTelephoneNumber	DnBTelecommunicationNumber	matchCandidates.organization.telephone.telephoneNumber
	DnBUnreachableIndicator	matchCandidates.organization.telephone.isUnreachable
DnBOrganizationPrimAddress	DnBCountryGroupName	matchCandidates.organization.primaryAddress.addressCountry.name
	DnBCountryISOAlpha2Code	matchCandidates.organization.primaryAddress.addressCountry.isoAlpha2Code
	DnBAddressLocalityName	matchCandidates.organization.primaryAddress.addressLocality.name
	DnBAddressRegionAbbreviatedName	matchCandidates.organization.primaryAddress.addressRegion.abbreviatedName
	DnBPostalCode	matchCandidates.organization.primaryAddress.postalCode matchCandidates.organization.primaryAddress.postalCodeExtension
	DnBStreetAddressLine	matchCandidates.organization.primaryAddress.streetAddress.li

Data Container ID	Attribute ID	JSON Path, Matching
		ne1 matchCandidates.organization.primaryAddress.streetAddress.line2
DnBOrgMailingAddress	DnBCountryGroupName	matchCandidates.organization.mailingAddress.addressCountry.name
	DnBCountryISOAlpha2Code	matchCandidates.organization.mailingAddress.addressCountry.isoAlpha2Code
	DnBAddressLocalityName	matchCandidates.organization.mailingAddress.addressLocality.name
	DnBAddressRegionAbbreviatedName	matchCandidates.organization.mailingAddress.addressRegion.abbreviatedName
	DnBPostalCode	matchCandidates.organization.primaryAddress.postalCode + '-' + matchCandidates.organization.primaryAddress.postalCodeExtension
	DnBStreetAddressLine	matchCandidates.organization.mailingAddress.streetAddress.line1 matchCandidates.organization.mailingAddress.streetAddress.line2
DnBOrganizationIdentification	DnBOrganizationIdentificationNumberType	matchCandidates.organization.registrationNumbers.typeDescription matchCandidates.organization.registrationNumbers.typeDnBCode
	DnBOrganizationIdentificationNumber	matchCandidates.organization.registrationNumbers.registrationNumber
DnBMostSeniorPrincipal	DnBFullName	matchCandidates.organization.mostSeniorPrincipals.fullName

D&B Organization Entity - Enriching - Attributes and Mapping

The following are STEP attributes and the corresponding D&B API JSON path(s) on the 'D&B Organization' (ID: DnBOrganization) entity used for D&B Enriching.

Attribute ID	JSON Path, Enriching
DnB8AFirmCertifiedDate	organization.socioEconomicInformation.8AFirmCertifiedDate
DnB8AFirmIndicator	organization.socioEconomicInformation.is8AFirm
DnBAgentIndicator	organization.isAgent
DnBBankNames	organization.banks.name
DnBBusinessEntity	organization.businessEntityType.description organization.businessEntityType.dnbCode
DnBCommCreditScoreMarketingRiskClass	organization.dnbAssessment.marketingRiskClass.description organization.dnbAssessment.marketingRiskClass.dnbCode
DnBControlOwnershipDate	organization.controlOwnershipDate
DnBControlOwnershipType	organization.controlOwnershipType.description organization.controlOwnershipType.dnbCode
DnBDefaultCurrencyISOAlpha3Code	organization.defaultCurrency
DnBDisabledOwnedIndicator	organization.socioEconomicInformation.isDisabledOwned
DnBDisadvantagedBusinessIndicator	organization.socioEconomicInformation.isDisadvantagedBusiness
DnBDomesticUltimateDUNSNumber	organization.corporateLinkage.domesticUltimate.duns
DnBDomesticUltimatePrimName	organization.corporateLinkage.domesticUltimate.primaryName
DnBDUNSNumber	organization.duns
DnBEthnicityType	organization.socioEconomicInformation.ownershipEthnicityType.description organization.socioEconomicInformation.ownershipEthnicityType.dnbCode
DnBExportIndicator	organization.isExporter
DnBFamilyTreeHierarchyLevel	organization.corporateLinkage.hierarchyLevel
DnBFamilyTreeMemberRole	organization.corporateLinkage.familytreeRolesPlayed.description organization.corporateLinkage.familytreeRolesPlayed.dnbcode
DnBFemaleOwnedCertifiedDate	organization.socioEconomicInformation.femaleOwnedCertifiedDate

Attribute ID	JSON Path, Enriching
DnBFemaleOwnedIndicator	organization.socioEconomicInformation.isFemaleOwned
DnBFranchiseOperation	organization.franchiseOperationType.description organization.franchiseOperationType.dnbCode
DnBFullReportDate	organization.dunsControlStatus.fullReportDate
DnBGlobalUltimateDUNSNumber	organization.corporateLinkage.globalUltimate.duns
DnBGlobalUltimateFamilyTreeLinkageCount	organization.corporateLinkage.globalUltimateFamilyTreeMembersCount
DnBGlobalUltimatePrimName	organization.corporateLinkage.globalUltimate.primaryName
DnBHeadquartersDUNSNumber	organization.corporateLinkage.headQuarter.duns
DnBHeadquartersPrimaryName	organization.corporateLinkage.headQuarter.primaryName
DnBHistoryRating	organization.dnbAssessment.historyRating.description organization.dnbAssessment.historyRating.dnbcode
DnBImportIndicator	organization.isImporter
DnBIsDelisted	organization.dunsControlStatus.isDelisted
DnBIsMailUndeliverable	organization.dunsControlStatus.isMailUndeliverable
DnBIsTelephoneDisconnected	organization.dunsControlStatus.isTelephoneDisconnected
DnBLaborSurplusAreaIndicator	organization.socioEconomicInformation.isLabourSurplusArea
DnBLabourSurplusAreaCertifiedDate	organization.socioEconomicInformation.labourSurplusAreaCertifiedDate
DnBLineOfBusinessDescription	organization.activities.description
DnBMarketabilityIndicator	organization.dunsControlStatus.isMarketable
DnBMarketingSegmentationClusterValue	organization.dnbAssessment.marketingSegmentationCluster
DnBMinorityOwnedCertifiedDate	organization.socioEconomicInformation.minorityOwnedCertifiedDate
DnBMinorityOwnedIndicator	organization.socioEconomicInformation.isMinorityOwned
DnBOperatingStatus	organization.dunsControlStatus.operatingStatus.description organization.dunsControlStatus.operatingStatus.dnbCode
DnBOperationsText	organization.operations.description

Attribute ID	JSON Path, Enriching
DnBOrganizationIncorporatedDate	organization.incorporatedDate
DnBOrganizationName	organization.primaryName
DnBOrganizationStartDate	organization.startDate
DnBParentDUNSNumber	organization.corporateLinkage.parent.duns
DnBParentPrimName	organization.corporateLinkage.parent.primaryName
DnBRegistrationLocationAddressRegion	organization.legalForm.registrationLocation.addressRegion
DnBSmallBusinessIndicator	organization.socioEconomicInformation.isSmallBusiness
DnBSmallBusinessOwnedCertifiedDate	organization.socioEconomicInformation.smallBusinessOwnedCertifiedDate
DnBStandaloneOrganizationIndicator	organization.isStandalone
DnBSubjectHandling	organization.dunsControlStatus.subjectHandlingDetails.description organization.dunsControlStatus.subjectHandlingDetails.dnbCode
DnBVeteranOwnedIndicator	organization.socioEconomicInformation.isVeteranOwned
DnBVietnamVeteranOwnedIndicator	organization.socioEconomicInformation.isVietnamVeteranOwned
DnBWebPageAddress	organization.websiteAddress.url

D&B Organization Entity - Enriching - Data Containers and Mapping

The following are STEP data containers, attributes, and the corresponding D&B API JSON path(s) on the 'D&B Organization' (ID: DnBOrganization) entity used for D&B Enriching.

This table is best viewed in online help.

Data Container ID	Attribute ID	JSON Path, Enriching
DnBCompetitors	DnBIndividualEmployeeQuantity	organization.competitors.individualEmployeeCount
	DnBOrganizationPrimaryName	organization.competitors.primaryName
	DnBSalesRevenueAmount	organization.competitors.salesRevenue
	DnBDUNSNumber	organization.competitors.duns

Data Container ID	Attribute ID	JSON Path, Enriching
	DnBIssuedShareCapitalAmount	organization.competitors.issuedShareCapitalAmount
DnBDomesticUltimatePrimAddress	DnBCountryISOAlpha2Code	organization.corporateLinkage.domesticUltimate.primaryAddress.addressCountry.isoAlpha2Code
	DnBPostalCode	organization.corporateLinkage.domesticUltimate.primaryAddress.postalCode
	DnBStreetAddressLine	organization.corporateLinkage.domesticUltimate.primaryAddress.streetAddress.line1 organization.corporateLinkage.domesticUltimate.primaryAddress.streetAddress.line2
	DnBCountryGroupName	organization.corporateLinkage.domesticUltimate.primaryAddress.addressCountry.name
	DnBCountyOfficialName	organization.corporateLinkage.domesticUltimate.primaryAddress.addressCounty.name
	DnBContinentalRegionName	organization.corporateLinkage.domesticUltimate.primaryAddress.continentalRegion.name
	DnBAddressLocalityName	organization.corporateLinkage.domesticUltimate.primaryAddress.addressLocality.name
	DnBAddressRegionName	organization.corporateLinkage.domesticUltimate.primaryAddress.addressRegion.name
	DnBAddressRegionAbbreviatedName	organization.corporateLinkage.domesticUltimate.primaryAddress.addressRegion.abbreviatedName
	DnBCountryFipsCode	organization.corporateLinkage.domesticUltimate.primaryAddress.addressCountry.fipsCode
DnBFacsimileNumber	DnBInternationalDialingCode	organization.fax.faxNumber
	DnBTelecommunicationNumber	organization.fax.isdCode
DnBGlobalUltimatePrimAddress	DnBCountryISOAlpha2Code	organization.corporateLinkage.globalUltimate.primaryAddress.addressCountry.isoAlpha2Code
	DnBPostalCode	organization.corporateLinkage.globalUltimate.primaryAddress.postalCode
	DnBStreetAddressLine	organization.corporateLinkage.globalUltimate.primaryAddress.streetAddress.line1

Data Container ID	Attribute ID	JSON Path, Enriching
		organization.corporateLinkage.globalUltimate.primaryAddress.streetAddress.line2
	DnBCountryGroupName	organization.corporateLinkage.globalUltimate.primaryAddress.addressCountry.name
	DnBCountyOfficialName	organization.corporateLinkage.globalUltimate.primaryAddress.addressCounty.name
	DnBContinentalRegionName	organization.corporateLinkage.globalUltimate.primaryAddress.continentalRegion.name
	DnBAddressLocalityName	organization.corporateLinkage.globalUltimate.primaryAddress.addressLocality.name
	DnBAddressRegionName	organization.corporateLinkage.globalUltimate.primaryAddress.addressRegion.name
	DnBAddressRegionAbbreviatedName	organization.corporateLinkage.globalUltimate.primaryAddress.addressRegion.abbreviatedName
	DnBCountryFipsCode	organization.corporateLinkage.globalUltimate.primaryAddress.addressCountry.fipsCode
DnBHeadquartersPrimAddress	DnBCountryISOAlpha2Code	organization.corporateLinkage.headQuarter.primaryAddress.addressCountry.isoAlpha2Code
	DnBPostalCode	organization.corporateLinkage.headQuarter.primaryAddress.postalCode
	DnBStreetAddressLine	organization.corporateLinkage.headQuarter.primaryAddress.streetAddress.line1 organization.corporateLinkage.headQuarter.primaryAddress.streetAddress.line2
	DnBCountryGroupName	organization.corporateLinkage.headQuarter.primaryAddress.addressCountry.name
	DnBCountyOfficialName	organization.corporateLinkage.headQuarter.primaryAddress.addressCounty.name
	DnBContinentalRegionName	organization.corporateLinkage.headQuarter.primaryAddress.continentalRegion.name
	DnBAddressLocalityName	organization.corporateLinkage.headQuarter.primaryAddress.addressLocality.name
	DnBAddressRegionName	organization.corporateLinkage.headQuarter.primaryAddress.addressRegion.name

Data Container ID	Attribute ID	JSON Path, Enriching
	DnBAddressRegionAbbreviatedName	organization.corporateLinkage.headQuarter.primaryAddress.addressRegion.abbreviatedName
	DnBCountryFipsCode	organization.corporateLinkage.headQuarter.primaryAddress.addressCountry.fipsCode
DnBIndustryCode	DnBIndustryCode	organization.industryCodes.code
	DnBDisplaySequence	organization.industryCodes.priority
	DnBIndustryCodeType	organization.industryCodes.typeDescription organization.industryCodes.typeDnBCode
	DnBIndustryCodeDescription	organization.industryCodes.description
DnBMostSeniorPrincipal	DnDnBFullName	organization.mostSeniorPrincipals.fullName
	DnBNameSuffix	organization.mostSeniorPrincipals.nameSuffix
	DnBLastName	organization.mostSeniorPrincipals.familyName
	DnBJobTitle	organization.mostSeniorPrincipals.jobTitles.title
	DnBMiddleName	organization.mostSeniorPrincipals.middleName
	DnBNamePrefix	organization.mostSeniorPrincipals.namePrefix
	DnBFirstName	organization.mostSeniorPrincipals.givenName
	DnBIdentificationNumber	organization.mostSeniorPrincipals.identificationNumber
	DnBGender	organization.mostSeniorPrincipals.gender.description organization.mostSeniorPrincipals.gender.dnbCode
	DnBPosition	organization.mostSeniorPrincipals.positions.description organization.mostSeniorPrincipals.positions.dnbCode
	DnBManagementResponsibility	organization.mostSeniorPrincipals.managementResponsibilities.description organization.mostSeniorPrincipals.managementResponsibilities.mrcCode

Data Container ID	Attribute ID	JSON Path, Enriching
DnBNumberOfEmployees	DnBNumberOfEmployeesValue	organization.numberOfEmployees.value
	DnBNumberOfEmployeesMinimumValue	organization.numberOfEmployees.minimumValue
	DnBNumberOfEmployeesMaximumValue	organization.numberOfEmployees.maximumValue
	DnBEmployeeFiguresDate	organization.numberOfEmployees.employeeFiguresDate
	DnBInformationScope	organization.numberOfEmployees.informationScopeDescription organization.numberOfEmployees.informationScopeDnBCode
	DnBReliability	organization.numberOfEmployees.reliabilityDescription organization.numberOfEmployees.reliabilityDnBCode
	DnBEmployeeCategory	organization.numberOfEmployees.employeeCategories.employmentBasisDescription organization.numberOfEmployees.employeeCategories.employmentBasisDnBCode
DnBOrganizationIdentification	DnBOrganizationIdentificationNumber	organization.registrationNumbers.registrationNumber
	DnBOrganizationIdentificationNumberType	organization.registrationNumbers.typeDescription organization.registrationNumbers.typeDnBCode

Data Container ID	Attribute ID	JSON Path, Enriching
DnBOrganizationPrimAddress	DnBCountryISOAlpha2Code	organization.primaryAddress.addressCountry.isoAlpha2Code

Data Container ID	Attribute ID	JSON Path, Enriching
	DnBLatitudeMeasurement	organization.primaryAddress.latitude

Data Container ID	Attribute ID	JSON Path, Enriching
	DnBAddressUsageTenureDetail	organization.primaryAddress.locationOwnership.description organization.primaryAddress.locationOwnership.dnbCode
	DnBCountryGroupName	organization.primaryAddress.addressCountry.name
	DnBMinorTownName	organization.primaryAddress.minorTownName
	DnBPostalCode	organization.primaryAddress.postalCode
	DnBLongitudeMeasurement	organization.primaryAddress.longitude
	DnBStreetAddressLine	organization.primaryAddress.streetAddress.line1 organization.primaryAddress.streetAddress.line2
	DnBCountyOfficialName	organization.primaryAddress.addressCounty.name
	DnBMetropolitanStatAreaUSCensusCode	organization.primaryAddress.statisticalArea.cbsaName
	DnBLanguageCode	organization.primaryAddress.language.description organization.primaryAddress.language.dnbCode
	DnBCountyFipsCode	organization.primaryAddress.addressCounty.fipsCode
	DnBContinentalRegionName	organization.primaryAddress.continentalRegion.name
	DnBMetropolitanStatAreaUSCensusNumCode	organization.primaryAddress.statisticalArea.cbsaCode
	DnBStreetNumber	organization.primaryAddress.streetNumber
	DnBStreetName	organization.primaryAddress.streetName
	DnBAddressRegionFipsCode	organization.primaryAddress.addressRegion.fipsCode
	DnBPostalCodePosition	organization.primaryAddress.postalCodePosition.description organization.primaryAddress.postalCodePosition.dnbCode
	DnBPostOfficeBoxNumber	organization.primaryAddress.postOfficeBox.postOfficeBoxNumber
	DnBPostOfficeBoxType	organization.primaryAddress.postOfficeBox.typeDescription

Data Container ID	Attribute ID	JSON Path, Enriching
		organization.primaryAddress.postOfficeBox.typeDnBCode

Data Container ID	Attribute ID	JSON Path, Enriching
	DnBManufacturingLocationIndicator	organization.primaryAddress.isManufacturingLocation
	DnBCountryFipsCode	organization.primaryAddress.addressCountry.fipsCode
	DnBGeographicalPrecision	organization.primaryAddress.geographicalPrecision.description organization.primaryAddress.geographicalPrecision.dnbCode
	DnBIsRegisteredAddressIndicator	organization.primaryAddress.isRegisteredAddress
	DnBIsResidentialAddressIndicator	organization.primaryAddress.isResidentialAddress
	DnBEconomicAreaOfInfluenceCode	organization.primaryAddress.statisticalArea.economicAreaOfInfluenceCode
	DnBPopulationRankNumber	organization.primaryAddress.statisticalArea.populationRank.rankNumber
	DnBPopulationRank	organization.primaryAddress.statisticalArea.populationRank.rankDnBCode organization.primaryAddress.statisticalArea.populationRank.rankDescription
	DnBPremisesAreaMeasurement	organization.primaryAddress.premisesArea.measurement
	DnBPremisesAreaMeasurementUnit	organization.primaryAddress.premisesArea.unitDescription organization.primaryAddress.premisesArea.unitDnBCode
	DnBPremisesAreaReliability	organization.primaryAddress.premisesArea.reliabilityDescription organization.primaryAddress.premisesArea.reliabilityDnBCode
	DnBCongressionalDistrict	organization.primaryAddress.congressionalDistricts.district
	DnBAddressLocalityName	organization.primaryAddress.addressLocality.name
	DnBAddressRegionName	organization.primaryAddress.addressRegion.name
	DnBAddressRegionAbbreviatedName	organization.primaryAddress.addressRegion.abbreviatedName

Data Container ID	Attribute ID	JSON Path, Enriching
DnBOrganizationRegisteredAddress	DnBCountryGroupName	organization.registeredAddress.addressCountry.name
	DnBMinorTownName	organization.registeredAddress.minorTownName
	DnBStreetNumber	organization.registeredAddress.streetNumber
	DnBStreetName	organization.registeredAddress.streetName
	DnBLanguageCode	organization.registeredAddress.language.description organization.registeredAddress.language.dnbCode
	DnBAddressLocalityName	organization.registeredAddress.addressLocality.name
	DnBAddressRegionName	organization.registeredAddress.addressRegion.name
	DnBAddressRegionAbbreviatedName	organization.registeredAddress.addressRegion.abbreviatedName
	DnBPostalCodePosition	organization.registeredAddress.postalCodePosition.description organization.registeredAddress.postalCodePosition.dnbCode
	DnBPostOfficeBoxNumber	organization.registeredAddress.postOfficeBox.postOfficeBoxNumber
	DnBPostOfficeBoxType	organization.registeredAddress.postOfficeBox.typeDescription organization.registeredAddress.postOfficeBox.typeDnBCode
	DnBCountryISOAlpha2Code	organization.registeredAddress.addressCountry.isoAlpha2Code
	DnBPostalCode	organization.registeredAddress.postalCode
	DnBStreetAddressLine	organization.registeredAddress.streetAddress.line1 organization.registeredAddress.streetAddress.line2 organization.registeredAddress.streetAddress.line3 organization.registeredAddress.streetAddress.line4
	DnBCountyOfficialName	organization.registeredAddress.addressCounty.name
DnBOrganizationSummary	DnBOrganizationSummaryText	organization.summary.text

Data Container ID	Attribute ID	JSON Path, Enriching
	DnBOrganizationSummaryTextType	organization.summary.textType.description organization.summary.textType.dnbCode
DnBOrgFormerPrimaryName	DnBOrgFormerPrimaryNameText	organization.formerPrimaryNames.name
	DnBOrgFormerPrimaryNameStartDate	organization.formerPrimaryNames.startDate
	DnBOrgFormerPrimaryNameEndDate	organization.formerPrimaryNames.endDate
DnBParentPrimAddress	DnBCountryISOAlpha2Code	organization.corporateLinkage.parent.primaryAddress.addressCountry.isoAlpha2Code
	DnBPostalCode	organization.corporateLinkage.parent.primaryAddress.postalCode
	DnBStreetAddressLine	organization.corporateLinkage.parent.primaryAddress.streetAddress.line1 organization.corporateLinkage.parent.primaryAddress.streetAddress.line2
	DnBCountryGroupName	organization.corporateLinkage.parent.primaryAddress.addressCountry.name
	DnBCountyOfficialName	organization.corporateLinkage.parent.primaryAddress.addressCounty.name
	DnBContinentalRegionName	organization.corporateLinkage.parent.primaryAddress.continentalRegion.name
	DnBAddressLocalityName	organization.corporateLinkage.parent.primaryAddress.addressLocality.name
	DnBAddressRegionName	organization.corporateLinkage.parent.primaryAddress.addressRegion.name
	DnBAddressRegionAbbreviatedName	organization.corporateLinkage.parent.primaryAddress.addressRegion.abbreviatedName
	DnBCountryFipsCode	organization.corporateLinkage.parent.primaryAddress.addressCountry.fipsCode

Data Container ID	Attribute ID	JSON Path, Enriching
DnBPrincipal	DnBFullName	organization.currentPrincipals.fullName
	DnBNameSuffix	organization.currentPrincipals.nameSuffix
	DnBLastName	organization.currentPrincipals.familyName
	DnBJobTitle	organization.currentPrincipals.jobTitles.title
	DnBMiddleName	organization.currentPrincipals.middleName
	DnBNamePrefix	organization.currentPrincipals.namePrefix
	DnBFirstName	organization.currentPrincipals.givenName
	DnBIdentificationNumber	organization.currentPrincipals.identificationNumber
	DnBGender	organization.currentPrincipals.gender.description organization.currentPrincipals.gender.dnbCode
	DnBPosition	organization.currentPrincipals.positions.description organization.currentPrincipals.positions.dnbCode
	DnBManagementResponsibility	organization.currentPrincipals.managementResponsibilities.description organization.currentPrincipals.managementResponsibilities.mrcCode
DnBShareCapitalDetails	DnBSharedCapitalStartDate	organization.capitalDetails.startDate
	DnBCurrencyISOAlpha3Code	organization.capitalDetails.capitalAmount.currency
	DnBCapitalType	organization.capitalDetails.capitalType.description organization.capitalDetails.capitalType.dnbCode
	DnBSharedCapitalValue	organization.capitalDetails.capitalAmount.value

Data Container ID	Attribute ID	JSON Path, Enriching
DnBStockExchangeDetails	DnBStockExchangeName	organization.stockExchanges.exchangeName.description
	DnBCountryISOAlpha2Code	organization.stockExchanges.exchangeCountry.isoAlpha2Code
	DnBPrimaryStockExchangeIndicator	organization.stockExchanges.isPrimary
	DnBStockTickerName	organization.stockExchanges.tickerName
DnBTelephoneNumber	DnBInternationalDialingCode	organization.telephone.isdCode
	DnBTelecommunicationNumber	organization.telephone.telephoneNumber
	DnBUnreachableIndicator	organization.telephone.isUnreachable
DnBThirdPartyAssessment	DnBAssessmentValue	organization.thirdPartyAssessment.value
	DnBAssessment	organization.thirdPartyAssessment.description organization.thirdPartyAssessment.dnbCode
	DnBAssessmentDate	organization.thirdPartyAssessment.assessmentDate

Data Container ID	Attribute ID	JSON Path, Enriching
DnBOrgMailingAddress	DnBCountryISOAlpha2Code	organization.mailingAddress.addressCountry.isoAlpha2Code
	DnBPostalCode	organization.mailingAddress.postalCode
	DnBStreetAddressLine	organization.mailingAddress.streetAddress.line1 organization.mailingAddress.streetAddress.line2
	DnBCountyOfficialName	organization.mailingAddress.addressCounty.name
	DnBCountryGroupName	organization.mailingAddress.addressCountry.name
	DnBMinorTownName	organization.mailingAddress.minorTownName
	DnBStreetNumber	organization.mailingAddress.streetNumber
	DnBStreetName	organization.mailingAddress.streetName
	DnBLanguageCode	organization.mailingAddress.language.description organization.mailingAddress.language.dnbCode
	DnBContinentalRegionName	organization.mailingAddress.continentalRegion.name
	DnBAddressLocalityName	organization.mailingAddress.addressLocality.name
	DnBAddressRegionName	organization.mailingAddress.addressRegion.name
	DnBAddressRegionAbbreviatedName	organization.mailingAddress.addressRegion.abbreviatedName
	DnBPostalCodePosition	organization.mailingAddress.postalCodePosition.description organization.mailingAddress.postalCodePosition.dnbCode
	DnBPostalRoute	organization.mailingAddress.postalRoute
	DnBPostOfficeBoxNumber	organization.mailingAddress.postOfficeBox.postOfficeBoxNumber
	DnBPostOfficeBoxType	organization.mailingAddress.postOfficeBox.typeDescription organization.mailingAddress.postOfficeBox.typeDnBCode
DnBTradeStyleName	DnBOrganizationName	organization.tradeStyleNames.name
	DnBDisplaySequence	organization.tradeStyleNames.priority

For more information regarding the easy setup for D&B integration, see the **D&B Integration Configuration and Easy Setup** topic.

Additionally, general information can be found in the **Attributes** section and the **Data Containers** section of the **System Setup / Super User Guide** documentation.

D&B Data Storage

The Dun & Bradstreet (D&B) integration automatic mapping from the D&B data received and the STEP D&B Organization is performed by the 'executeBusinessActionWithDnbIntegrationContext()' method in the business actions. The Easy Setup uses this method by default.

- Details about the default data stored and the mapping between the D&B model and the STEP model can be found in the **D&B Attributes and Data Containers** section.
- To store additional data, business actions can be extended using JavaScript. Refer to the **Scripting API** documentation, which is accessible via the **STEP API Documentation** at [system]/sdk or from the system Start Page.

As shown in the numbered list below, D&B data is stored when matching (1) and enriching (2). Data can also be stored during re-matching (3), for example, when a second customer organization matches on one or more of the same D&B Organization entities that already exist in STEP.

1. **Matching**: The data returned from D&B is limited but enough to identify the organization. The data is stored on newly created D&B Organization entities.
2. **Enriching**: Detailed data is fetched and stored on the D&B Organization entity. All existing data from the matching request is overwritten, including all instances of multivalued data containers to ensure all data is relevant and up to date.
3. **Re-Matching**: When storing the limited set of matching data on existing, already enriched, D&B Organization entities, the following rules apply:
    - For attributes and single valued data containers, the data is stored one-to-one, overwriting any existing data.
    - For multivalued data containers, an algorithm ensures existing (enrichment) data is not overwritten by comparing the new data with existing data on the data container. If there is a full match (the received subset of data is the same as the existing), nothing is updated. If there is not a full match, a new data container instance is created. No existing data container instances are ever deleted.

Note: If using more than one D&B enrichment product, the default enrichment mapping from the D&B data model to STEP is only used for the first enrichment. Any subsequent use can overwrite the previous enrichment data.

For information about migrating from D&B V2 to Direct+, see the **Dun & Bradstreet Migration: Direct 2.0 to Direct+** topic in the **System Administration** documentation.

D&B Direct+ API Products

The D&B Direct+ API has several products for matching and enriching. Listed below are the products that can be directly configured using Operation Configurations.

A complete list can be found on the web in the D&B documentation https://directplus.documentation.dnb.com/ under the API Reference section.

Matching

Only one matching product is supported:

- Identify Resolution (Cleanse and Match)

Enrichment

The following sections include enrichment products that can be directly configured using Operation Configurations.

Analytics Assessment

- Analytics Assessment Decision Headquarter with Upward Linkage (aasdhq)
- Analytics Assessment Global IT Buydex (aasbig)
- Analytics Assessment Material Change (aasmcu)
- Analytics Assessment Sales & Employee Assignment Model (aassem)

Company Profile

- Company Profile for Compliance Verification (cmpcvf)
- Company Profile for Financial Market Data (cmpfmd)
- Company Profile for Managing Trade Credit Risk - Summary (cmptcs)
- Company Profile Supplier Master Data Enrichment (cmpsup)
- Company Profile Supplier Risk Assessment (cmpsra)
- Company Profile with Diversity Details (cmpdve)
- Company Profile with Diversity Indicator (cmpdvs)
- Company Profile with Executives, Linkage, and Financials (cmpelf)
- Company Profile with Financial Comparative Data (cmpfcd)
- Company Profile with Third Party Financials (cmptpf)
- Company Profile, Linkage, and Executives (cmpelk) v2

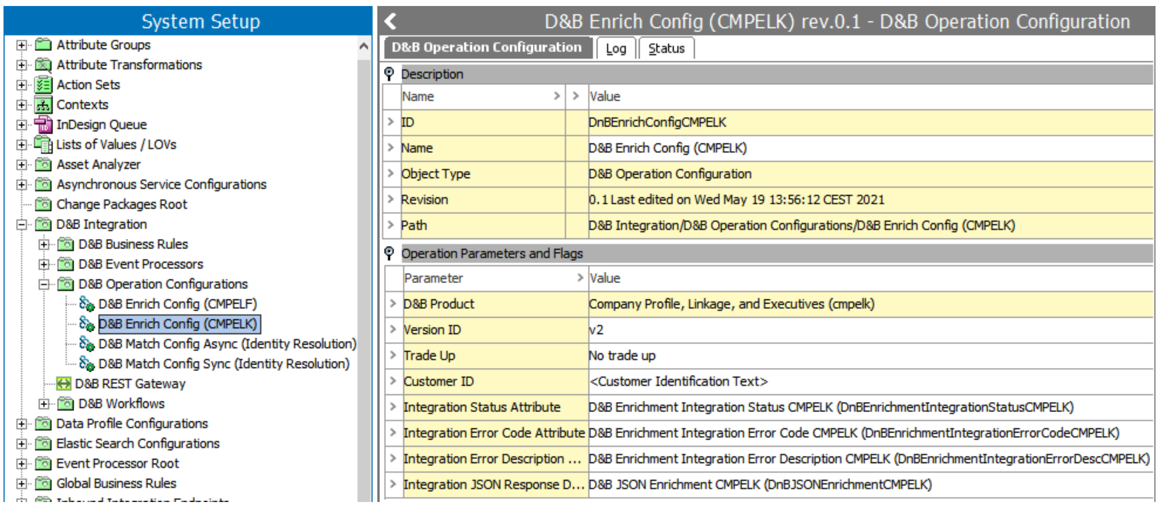
Corporate Linkage

- Corporate Linkage Alternate Linkage (lnkalt)
- Corporate Linkage Global Beneficial Ownership (lnkgbo)
- Corporate Linkage Minority Linkage (lnkmin)
- Corporate Linkage Upward Corporate Linkage (lnkupd)

By default, Easy Setup creates an enrichment configuration for the 'Company Profile with Executives, Linkage, and Financials (cmpelf)' product. When configuring enrichment for other D&B products, if the service returns the same fields / structures, the default mapping from the D&B data structure to the STEP data model is used. See a complete list of the default mappings in the **D&B Attributes and Data Containers** topic.

> **Important:** When adding operation configuration for new enrichment products, the setup entity defaults to version 1 of the product. Adjust this to use the newest version available.

Example of configuration for CMPELK with Version ID manually changed to 'v2':



D&B Error Handling with Integration Status

When processing the Dun & Bradstreet (D&B) matching or enriching response using the default business actions provided by Easy Setup, a logical categorization of the result called 'Integration Status' is available. When processing asynchronously, it is used logically for workflow post actions (see the **Asynchronous Processing: Retry and Post Actions** section below) and it is available in the Scripting API along with the D&B error code and error description. API documentation is accessible at [system]/sdk or access the **STEP API Documentation** from the Start Page.

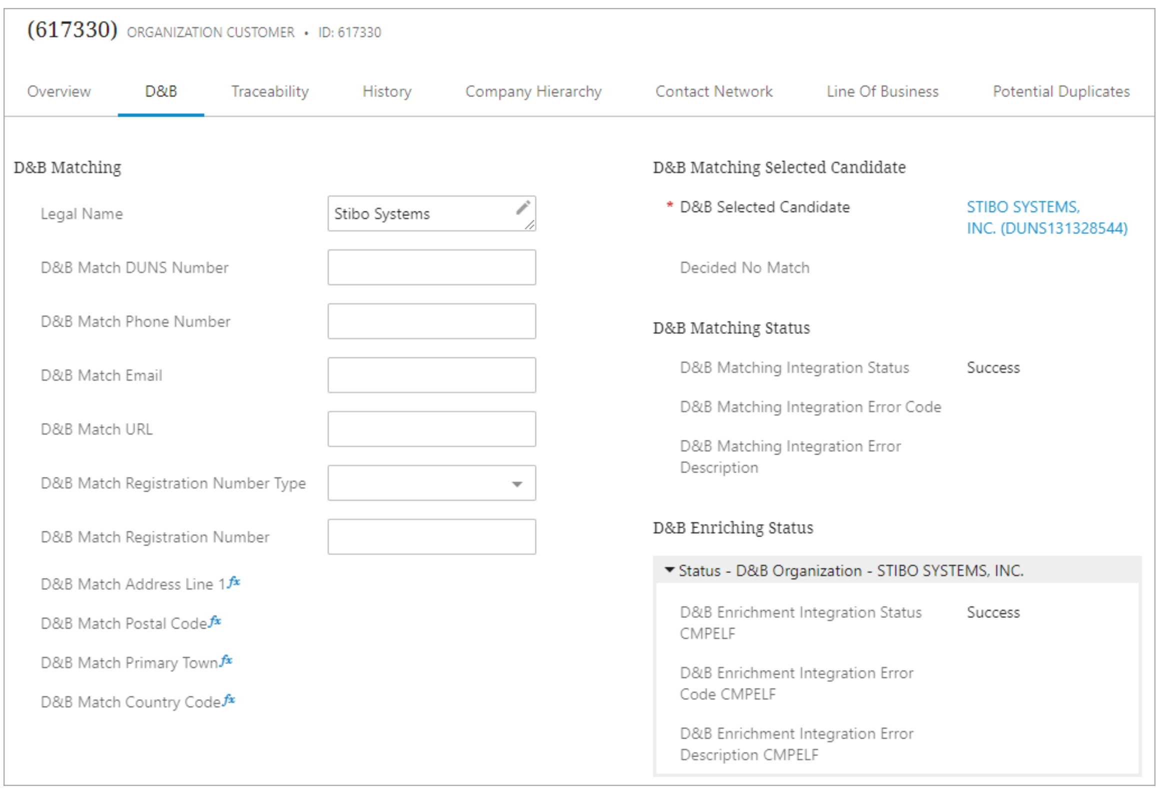
Integration Status and D&B Error Codes

Integration Status	D&B Error Codes + Description
Success	No D&B error and no STEP error
Success - limited result	10008, 10009, 10010 Truncated list of candidates returned, or enrichment result only partially available. **Note:** Only applicable for D&B Products: UBO, Dow Jones Search, and Company Report.
No match result	20505
Access token expired	40 The D&B REST gateway will automatically request a new token.
D&B internal server error - continue	05001, 05002, 05003, 05004, 05005, 40100
D&B internal server error - resubmit	05006, 00034, 00035, 00036, 05010, 00048, 00002, 05009, 00049
Enrichment result not available	40003, 40001, 10001, 10004, 10200, 40105, 40002
Invalid license	00005, 11001, 00004, 00012, 00014, 00041, 22006
Invalid request format	00006, 00008, 00009
Invalid request parameter - continue	05007, 05011, 10002, 10003, 10005, 10007, 20001, 20502, 20503, 20504, 20506, 20507, 20508
Invalid request parameter - fail	05008
Invalid request resource	40101, 40102, 00037, 00038, 10006, 40103, 40104
Throttling error - quota	00050
Throttling error - spike/tps	00045, 00046, 00047
Unknown error	All other D&B errors

Integration Status	D&B Error Codes + Description
STEP connection timeout	STEP TCP/IP connection timeout to D&B service
STEP data validation errors	STEP validation error of match parameters
STEP internal error	STEP error, e.g., when processing the result from D&B
Missing D&B STEP component license	STEP error when DnBIntegration license is not enabled on the STEP system

The Integration Status can, along with the D&B error code and D&B error description, automatically be written in configured attributes so that it is easily available for Web UI screens. See the description about Operation Configurations in the D&B matching and enriching topics. **D&B Matching** topic and the **D&B Enriching** topic. The attributes are written both for synchronous processing and for asynchronous processing, except when the event processor is stopped. Refer to the table in the next section. Errors must be handled when they arise.

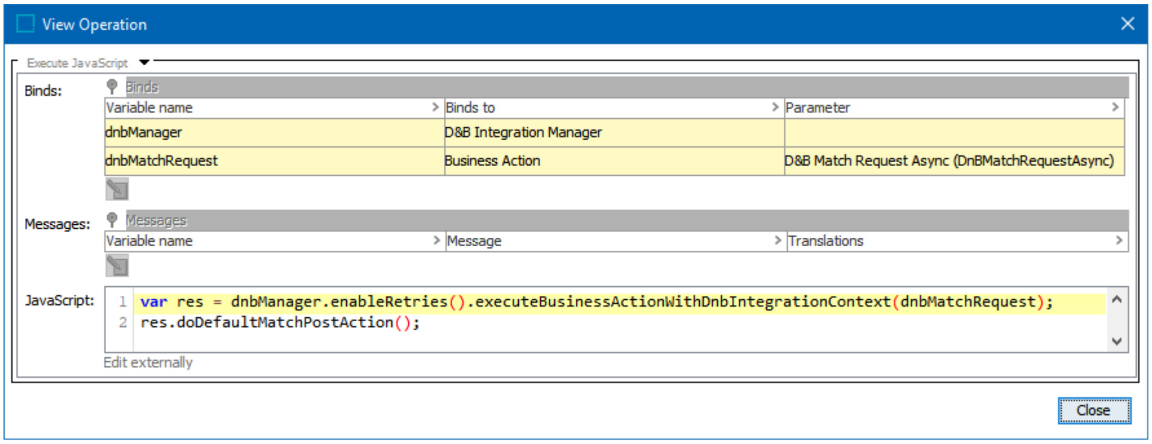
Below is an example of a Web UI showing Matching and Enriching integration status attributes including error code and error description:



Asynchronous Processing: Retry and Post Actions

The following applies when using the default configuration provided by Easy Setup.

The event processor ‘D&B Match Processor’ executes the business action ‘D&B Match Request Async Wrapper.’ After the match API call, this script executes the 'doDefaultMatchPostAction() method.' This ‘post action’ method is responsible for appropriate error handling which is performed according to the table below.



For creating customized ‘post actions’ in JavaScript instead of the doDefaultMatchPostAction() and doDefaultEnrichPostAction(), see the Scripting API reference material under ‘com.stibo.integration.dnb.directplus.bind’ - DnbIntegrationManager. The Scripting API information can be found via the **STEP API Documentation** page accessible at [system]/sdk or from the Start Page

The table below outlines the possible successes and errors during the matching and enriching D&B process. The default ‘post action’ logic is included in the Stop Event Processor column and shows if the event processor is stopped or an event is submitted to the workflow that triggers the entity into a certain workflow state. Errors that send the entity to the ‘Matching error‘ state or the ‘Profiling error‘ state requires a user to access the task and handle the error.

Integration Status	Default Retry Strategy	Stop Event Processor	Matching Workflow State	Enriching Workflow State
Success			Select Candidate [1]	End
Success - limited result			Select Candidate [1]	End
No match result			Select Candidate	
Access token expired	Yes [2]			
D&B internal server error - continue			Matching error	Profiling error
D&B internal server error - resubmit	Yes, 3 retries		Matching error	Profiling error
Enrichment result not				Profiling error

Integration Status	Default Retry Strategy	Stop Event Processor	Matching Workflow State	Enriching Workflow State
available				
Invalid license		Yes		
Invalid request format		Yes		
Invalid request parameter - continue			Matching error	Profiling error
Invalid request parameter - fail			Matching error	Profiling error
Invalid request resource			Matching error	Profiling error
Throttling error - quota		Yes		
Throttling error - spike/tps	Yes, retry forever [3]			
Unknown error			Matching error	Profiling error
STEP connection timeout	Yes, retry forever [3]			
STEP data validation errors			Matching error	Profiling error
STEP internal error			Matching error	Profiling error
Missing D&B step component license		Yes		

[1] Or 'End' state if autolinked.

[2] A new access token is automatically fetched in the D&B REST Gateway.

[3] The retry logic happens when using the enableRetries() method in the business action. When enabled, all retry logic is performed in the D&B REST Gateway. Stop the gateway to exit that retry loop.

Experian Email Validation Integration

The Experian email validation integration focuses on easily maintaining valid email contact data and providing an overview of email data quality through use of the Experian Data Quality asynchronous Clean Web Service (Experian). The Experian integration provides an email data quality solution that offers more than just syntax validation. Using the Experian Email Validation integration means email account data can be checked for domain existence, identify malicious email addresses (like spam traps), and in some cases (e.g., Gmail) user account existence.

Whenever email data is created, imported, or updated, STEP can asynchronously validate the information in the background. Once email data has been validated it can be assigned a quality rating so data stewards can monitor the overall email data quality. Also, email data (or groups of email data) can be manually selected and sent to Experian for validation. This can be useful when a data steward needs to revalidate email data in preparation of a task such as an email campaign.

Both STEP Workbench and Web UI support validating emails from within STEP using the Experian service, and allow for enriching email address information in STEP with this data.

The integration to Experian services is implemented via an asynchronous integration pattern, using the following STEP components:

- Experian Email Validation Configuration Object Type
- Email Revalidation Business Condition
- Experian Email Validation Processing Plugin
- Web UI Action Button to Validate Emails
- Email Component Model (Featuring an Easy Setup Option)

Prerequisites

This functionality has been developed to work with Experian Data Quality's bulk validation product known as the 'Clean Web Service Version 2.' The purpose of Clean Web Service Version 2 is to provide the ability to clean, enrich, and validate email address details quickly and accurately via a hosted service. Experian Data Quality recommends using the service regularly to ensure that your contact information is always correct and up-to-date.

Stibo Systems acts as reseller of Experian's email data quality web service and customers are able to use the Experian web service for other purposes. Contact your Stibo Systems account manager or partner manager to obtain login information for the Experian Data Quality self-service portal (https://portal.experianmarketingservices.com/content/) where customers can monitor their credit balance, set up an email notification, and see their secure tokens. Customers can access other features provided within the 'Clean Web Service Version 2' product by visiting the following link: https://www.edq.com/documentation/apis/clean-web-service/

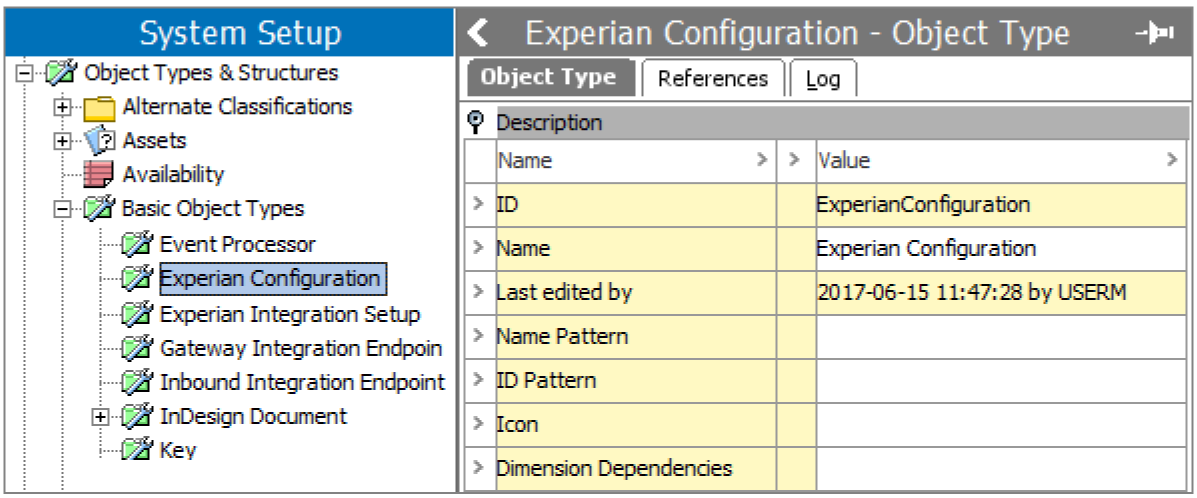
Important: To use the Experian Email Validation Integration functionality, the applicable recipe must be applied, and the Experian License must be obtained from Stibo Systems in order to have access to the Experian API key. Contact your Stibo Systems account manager for more information and licensing terms. Additionally, a Secure Sockets Layer (SSL) connection must be made to https://api.experianmarketingservices.com and http://www.qas.com.

Configuring the component model requires users to be familiar with the System Setup tab in STEP Workbench and how to configure within this area (e.g., creation and maintenance of object types, attributes, and references). Users should also be familiar with business rules, event processors, and have the privileges required to carry out these tasks. It is expected that anyone configuring Web UI components is familiar with Web UI designer. If not, the users should search online help to understand these concepts and processes.

Experian Email Validation Configuration Object

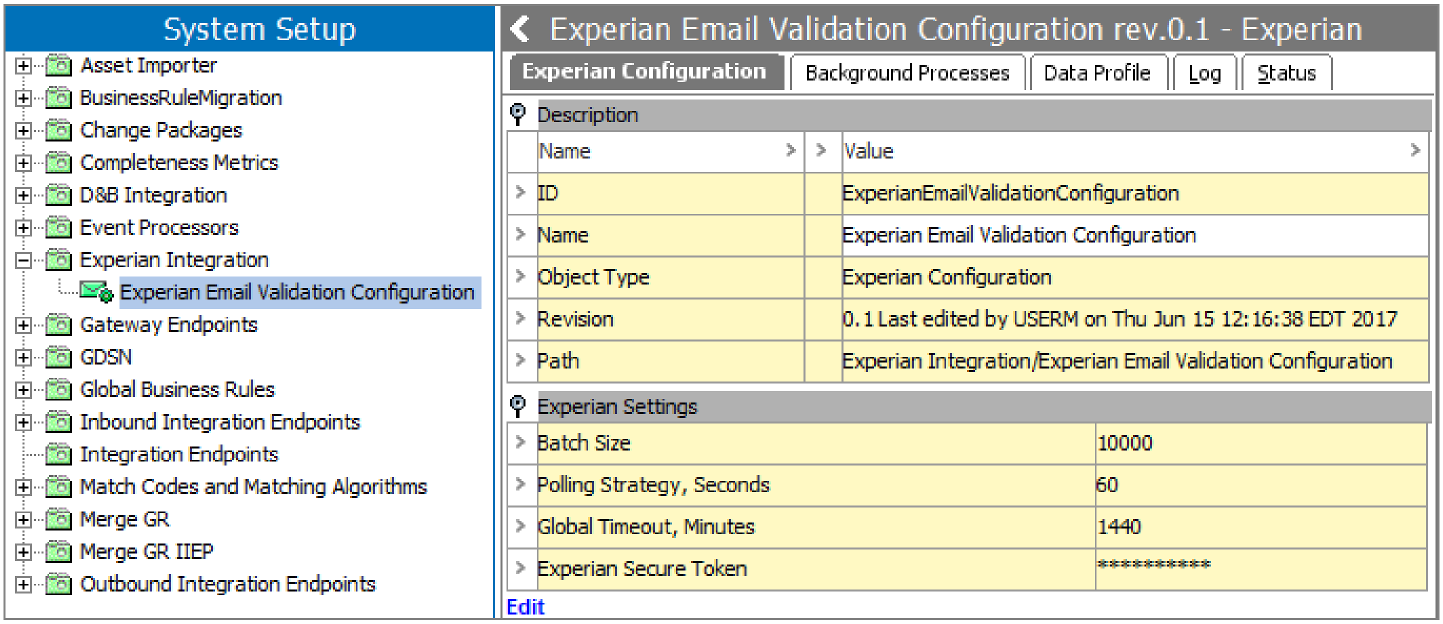
The purpose of the Experian Email Validation Configuration object is to hold the configurations for the different Experian email validation integration parameters. The Experian Email Validation Configuration object can be automatically created (using the Experian Configuration setup entity type) when the Easy Setup for Email Component Model wizard is used.

The Experian Configuration setup entity type can be found within the System Setup > Object Types & Structures > Basic Object Types (as shown in the screenshot below).



Experian Configuration objects are represented within the workbench using the icon, and include the following parameters within the Experian Settings flipper: Batch Size, Polling Strategy in Seconds, Global Timeout in Minutes, and Experian Secure Token. All validate email actions and Experian Email Validation processors must refer to an Experian Email Validation Configuration object.

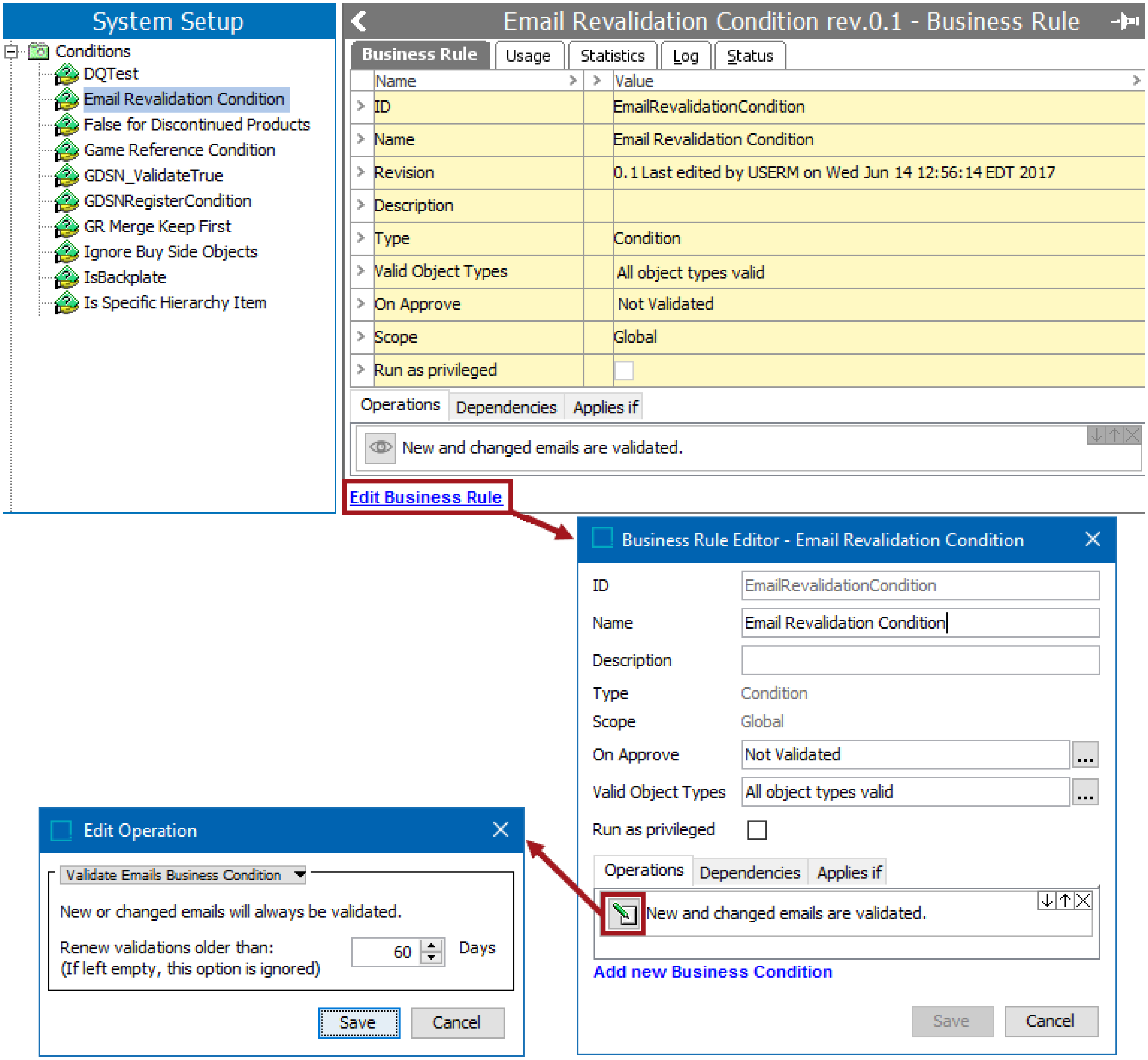
The screenshot below is an example of how the Experian Email Validation Configuration object can be set up. For this example, the optional Setup Group (Experian Integration) was created prior to starting the Easy Setup for Email Component Model wizard, and then selected as the Setup Group during the wizard step 'Create New Experian Email Validation Configuration.'



Note: The Experian Secure Token is encrypted in the database and displays masked within the workbench.

Email Revalidation Business Condition

An event processor using the Experian Email Validation processing plugin can be configured to revalidate emails by using the Revalidation Business Condition plugin along with the Experian Email Validation Configuration object, which can be created / configured automatically when the Easy Setup for Email Component Model wizard is used. The Revalidation Business Condition plugin allows for email objects (across the configured data container types defined by the Email Component Model) that have not been validated for a specified number of days to be sent for revalidation.



The screenshot above is an example of how the Email Revalidation Business Condition can be set up. For this example, the Easy Setup for Email Component Model wizard was used, and during the wizard step 'Create New Email Revalidation Condition,' the Setup Group 'Conditions' was selected.

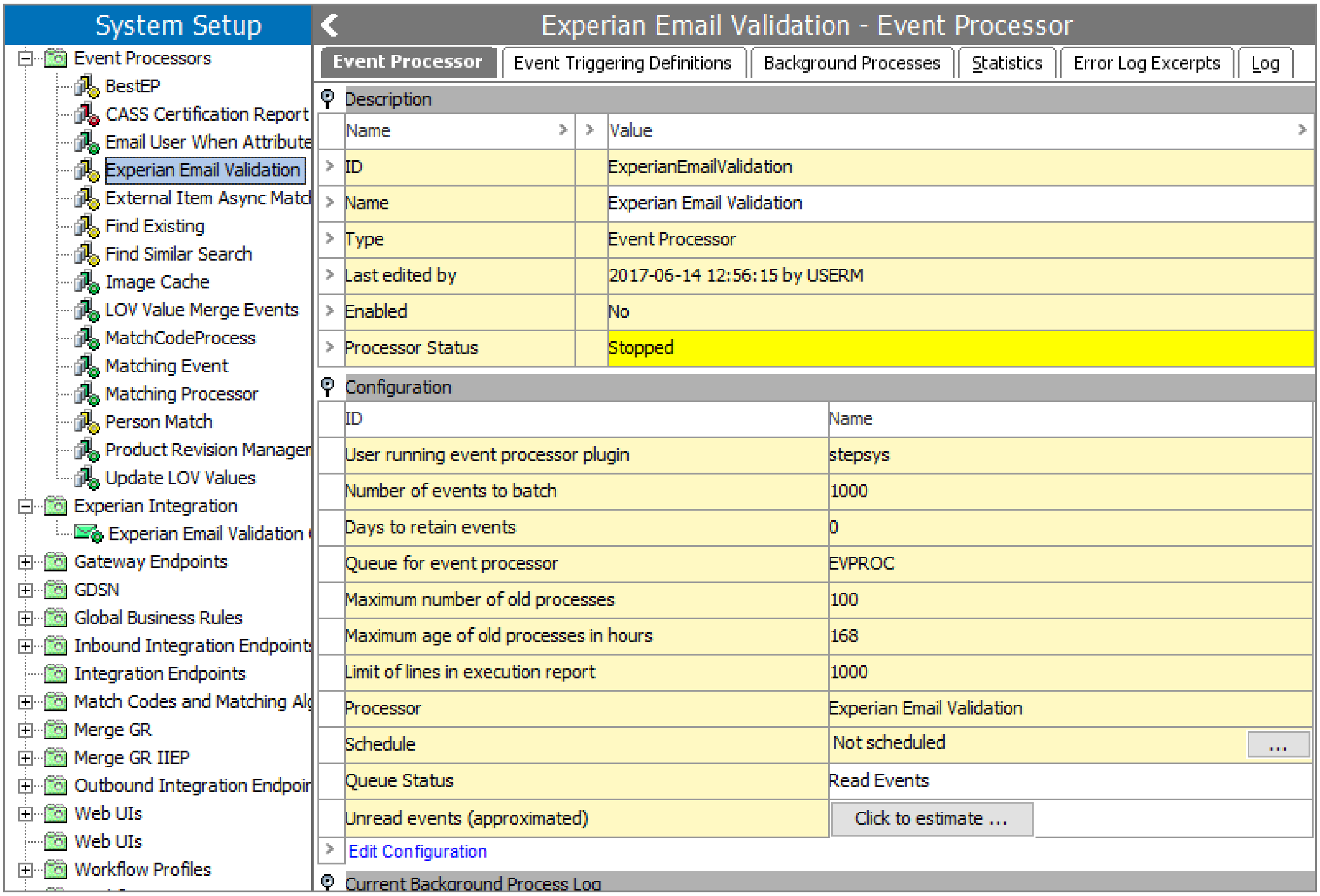
This makes it possible to define a condition like: 'Validate all billing emails that have changed since the last validation, or have not been validated in the last 2 months.' More than one email revalidation condition can be created, and using JavaScript it is possible to extend the Email Revalidation condition to take properties of the entire account into consideration. If JavaScript is used to implement the Email Revalidation condition, the bind 'Current Email' can be used to refer to the email data container that is currently being processed.

Important: All Email Revalidation Business Conditions must be valid for the object types that use the email data container. For example, if you are working with 'Customers' object types that have email data stored in a data container named 'Email,' then the Email Revalidation Business Condition must be valid for the 'Customers' object type. When the Easy Setup for Email Component Model wizard is used, this occurs automatically.

For more information on how the Email Revalidation Business Condition can be configured within the Experian Email Validation processing plugin, see **Experian Email Validation Processing Plugin Parameters and Triggers** section of the **Processing Plugins** documentation.

Experian Email Validation Processing Plugin

The Experian Email Validation processing plugin can be used to create an event processor that listens for changes to email data on account objects. This means that email updates (manual or via import) will create events for the configured event processor. The Experian Email Validation processing plugin will interface with the Experian batch API and send batches of emails for validation to the Experian service. The emails will be extracted from the accounts that originate from the events using the Experian Email Validation processing plugin. The batch being processed by the event processor will continue to poll (based upon the configurations within the Experian Configuration object), for an answer from the Experian batch API until the batch has been processed or times out. When updates are received, the email data quality fields (configured within the Email Component Model) are updated. A new batch will not be picked up until the current batch has been processed with answers received or has timed out.



The screenshot above is an example of how the Experian Email Validation processing plugin can be set up. For this example, the Easy Setup for Email Component Model wizard was used, and during the wizard step 'Create New Experian Email Validation Event Processor,' the Setup Group 'Event Processors' was selected. It is important to note, before this event processor configuration can be used, it must be enabled.

For more information, see the **Experian Email Validation Processing Plugin Parameters and Triggers** section of the **Processing Plugins** documentation.

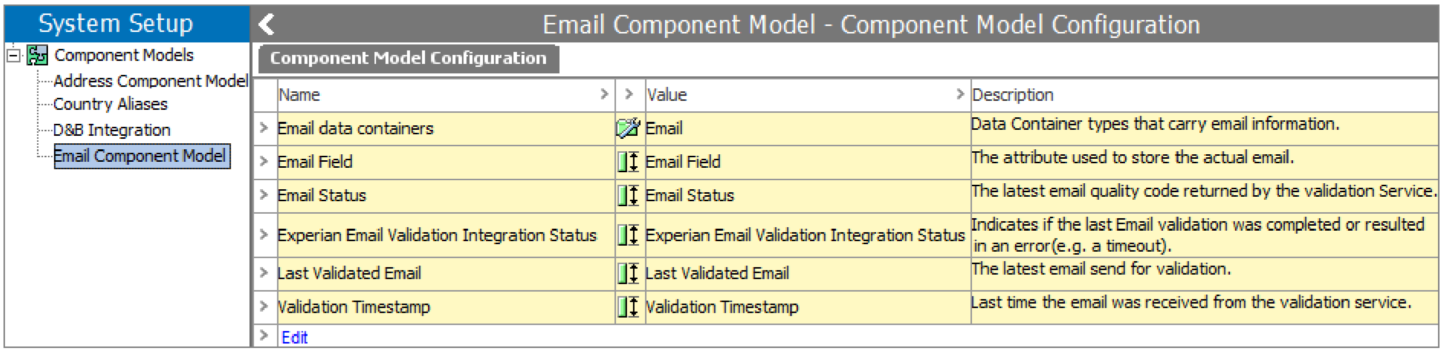
Web UI Validate Emails Action Button

The 'Validate Emails' action button is available within Web UI, and allows users to perform email validations on a node list with one click. Users can choose one or more objects (containing email account data) that need to be validated from a list of emails, then click on the 'Validate Emails' button to run the 'Experian Email Validation Job' background process. In the background, the Experian Email Validation Job will continue to poll for an answer from the Experian batch API and wait for a status message on all emails. Whenever an update is received, the available status information will be written into the execution report of the background process, and the corresponding email data quality fields (configured within the Email Component Model) will be updated.

For more information, see the **Experian Email Validation Integration in Web UI** section of this guide.

Email Component Model

The Email Component Model defines the structure and data quality of emails in STEP. It contains the configuration of the email data containers, and attributes that are necessary for proper functionality of the email validation and email data quality overview. After running the easy setup, (covered in the **Configuring Experian Email Validation Integration Using the Easy Setup Option** topic within this guide) the Email Component Model will map the component model names (Experian data fields) to STEP attributes. A completed component model using the Easy Setup option and automatic configuration options will appear like the following image.



If during the easy setup not all values could be mapped, then you will be able to see what requires manual mapping within the component model. If any of the object types, attributes / data containers being mapped are unclear, the Description column provides context for how these fields are to be used.

For more information, see the **Configuring Experian Email Validation Integration Using the Easy Setup Option** topic within this guide.

Configuring Experian Email Validation Integration Using the Easy Setup Option

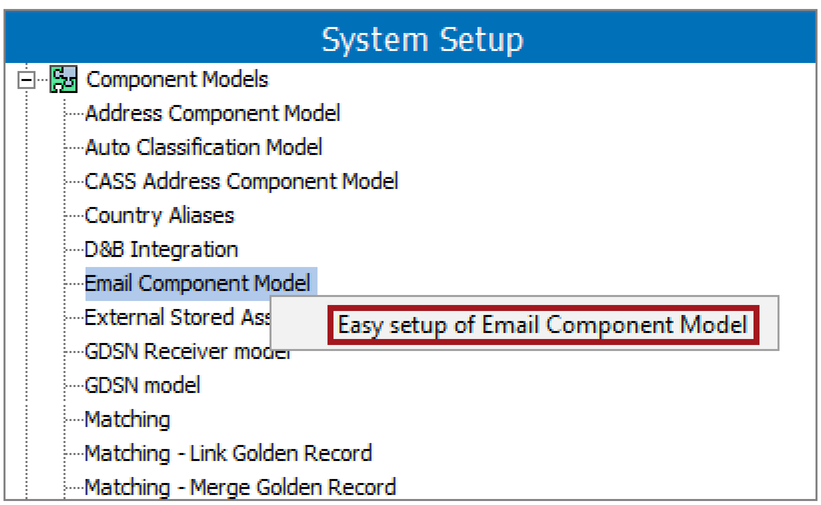
The Experian Email Validation Integration includes an Easy Setup option that will create the setup files, needed attributes, and data containers. The Easy Setup option is designed to make it so that users have very little to do configuration-wise to get up and running as soon as possible. Anyone configuring or using the Email Component Model Easy Setup option needs to have the proper privileges and understand how all the pieces work together.

Important: The Easy Setup can be finished without supplying the Experian Secure Token, however before the Experian Email Validation Integration can be used, the Token will need to be added by rerunning the Email Component Model Easy Setup wizard or editing the Experian Email Validation Configuration object.

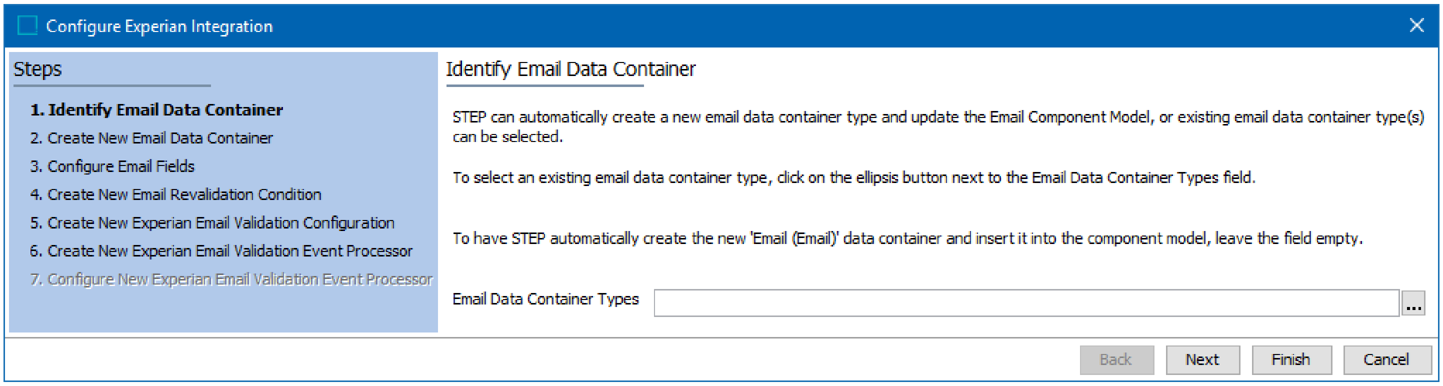
Optionally (before starting the easy setup wizard), create a setup group for the Experian Configuration to be stored in, or when prompted pick any one of the existing setup groups.

To start the automatic configuration:

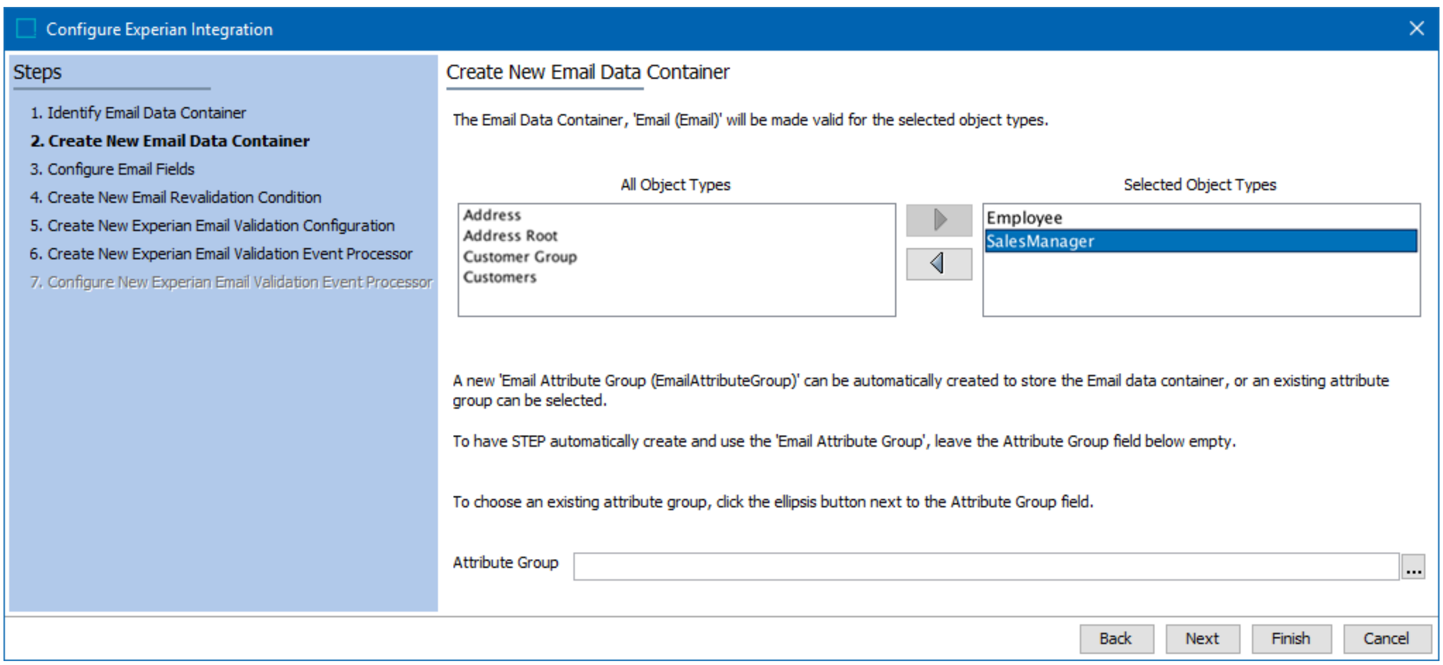
1. Go to System Setup > Component Models > Select **Email Component Model**.
2. Right-click **Email Component Model**, and select **Easy setup of Email Component Model**.



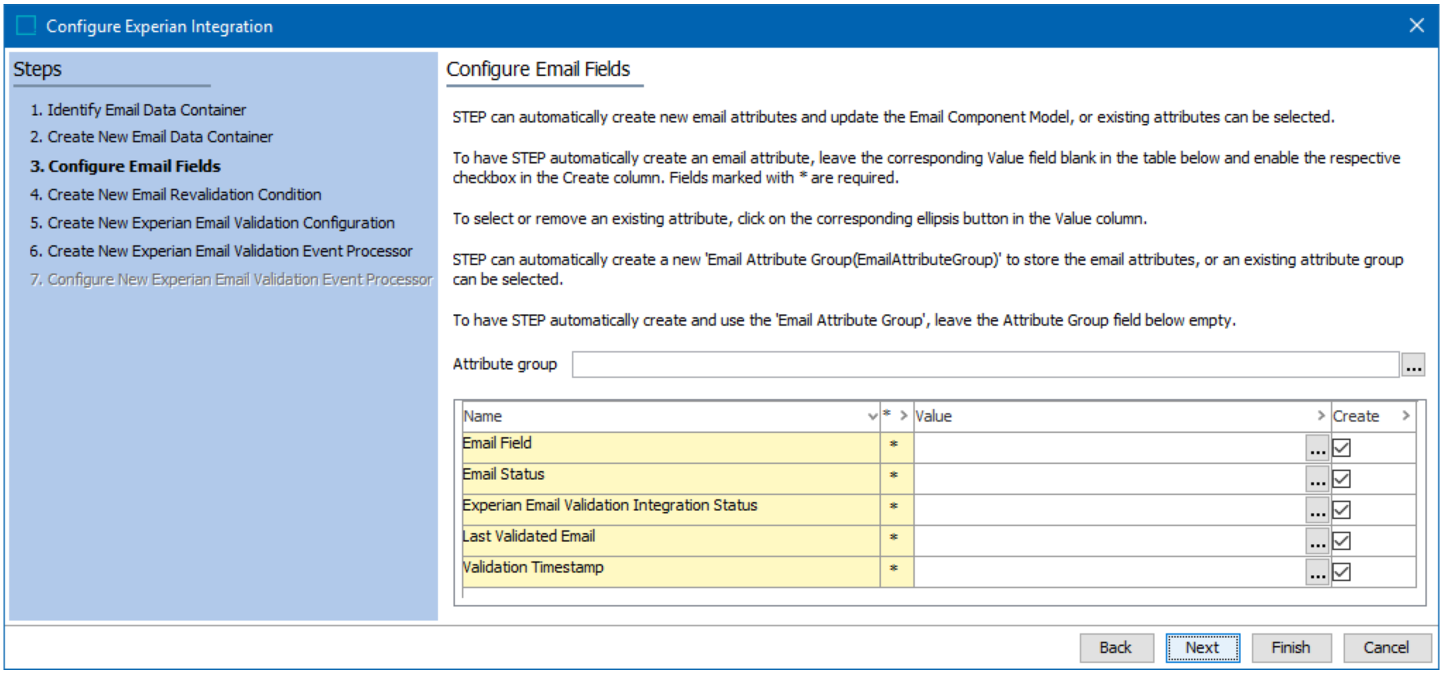
3. The Configure Experian Integration dialog will display the wizard step 'Identify Email Data Container' with detailed configuration instructions. Read the dialog text carefully to determine the necessary action.



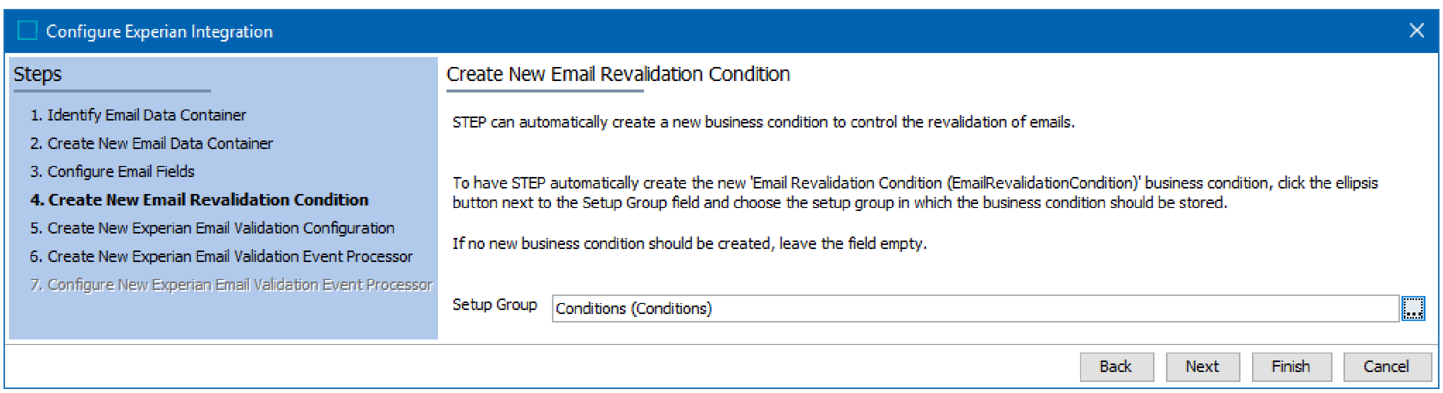
4. Click the **Next** button, and the wizard step 'Create New Email Data Container' will display with detailed configuration instructions. At a minimum, choose the object type(s) to enable the Next button.



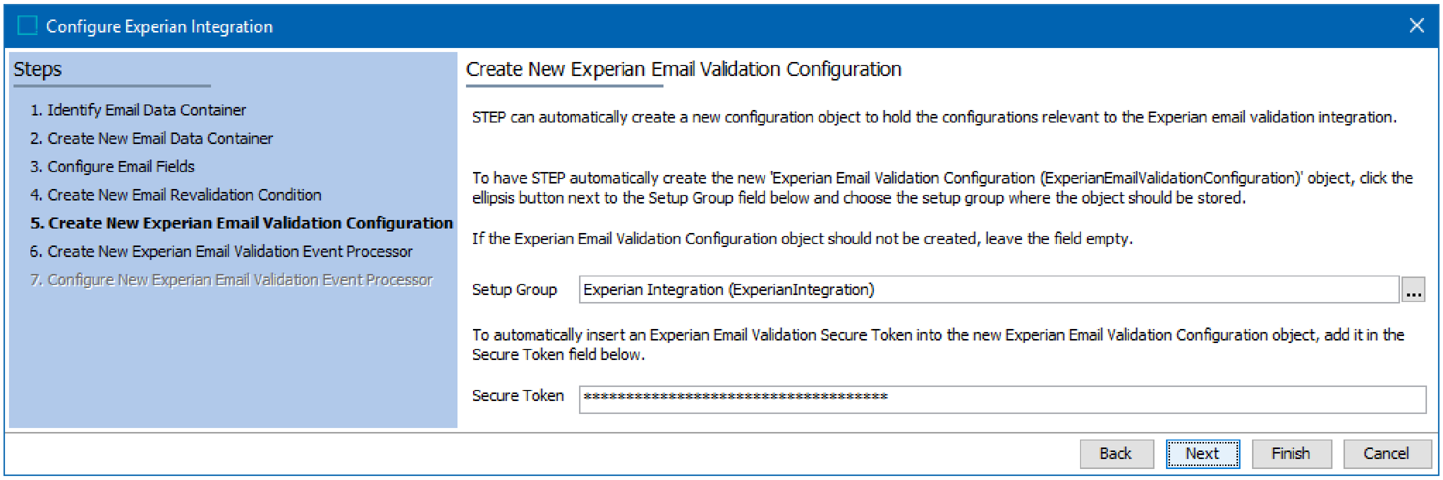
5. Click the **Next** button, and the wizard step 'Configure Email Fields' will display with detailed configuration instructions.



6. Click the **Next** button, and the wizard step 'Create New Email Revalidation Condition' will display with detailed configuration instructions.

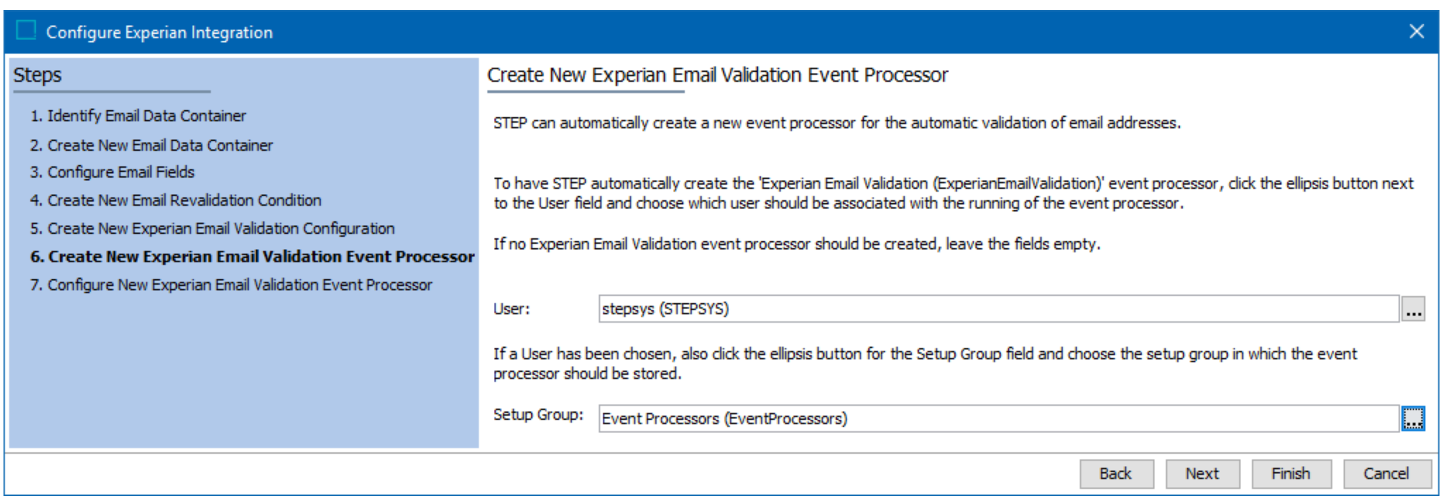


7. Click the **Next** button, and the wizard step 'Create New Experian Email Validation Configuration' will display with detailed configuration instructions.



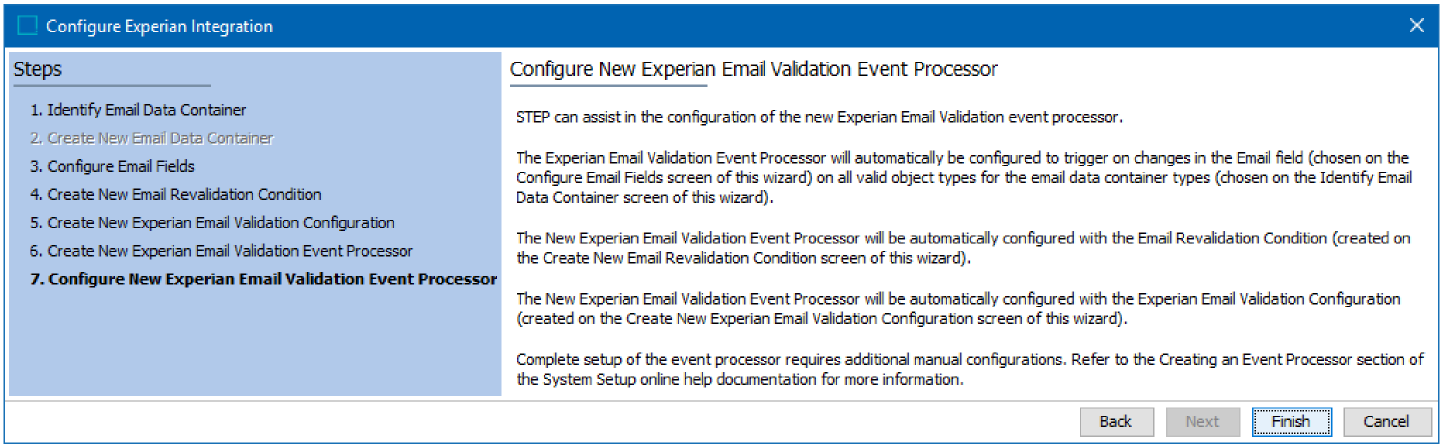
- **Setup Group:** For the example above, prior to starting the wizard a Setup Group was created (as advised in the beginning of this section). It is possible to cancel the wizard, and create your own Experian specific Setup Group, or click the ellipsis button (...) and pick any one of the existing setup groups.
- **Secure Token:** As characters are entered into the field, they are masked for security purposes. This field can be left blank, but you will need to return to this step again later to enter the token before the Experian email validation feature can be used.

8. Click the **Next** button, and the wizard step 'Create New Experian Email Validation Event Processor' will display with detailed configuration instructions.

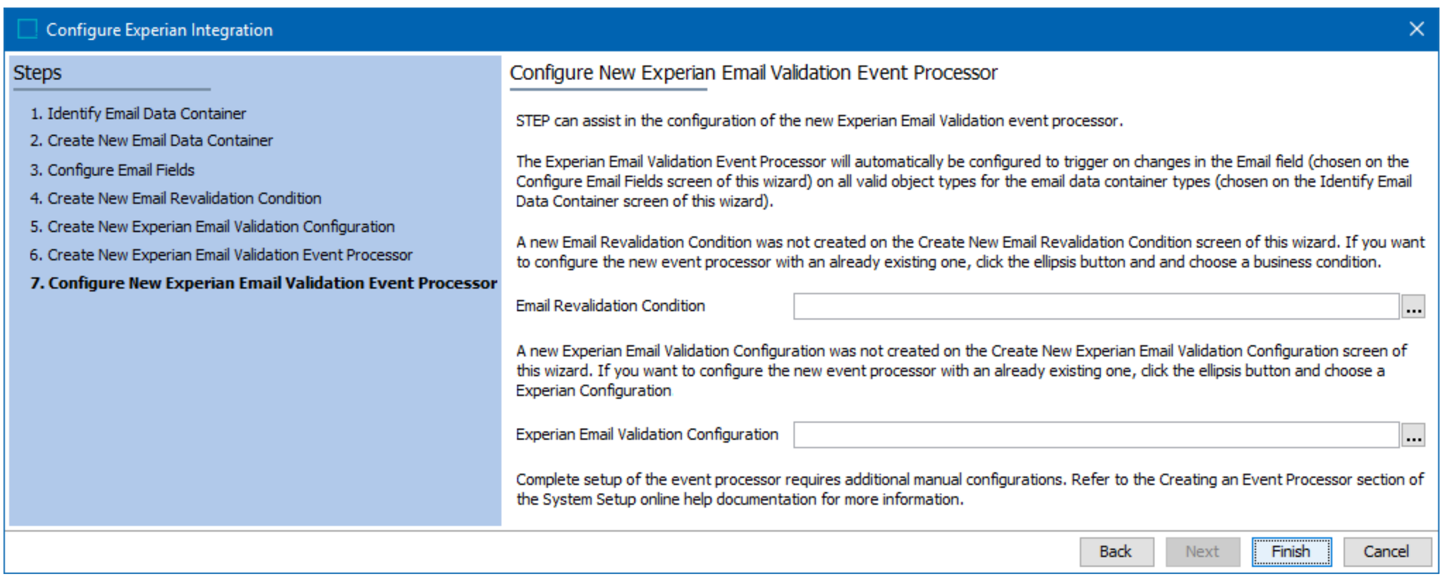


Important: The privileges of the selected user determine which actions the event processor can perform and what data can be processed. Common setup is to create a special system user for this purpose so that the effects of the event processor are easily identified. For more information, see the **Configure Action Sets and Privileges** section of the **Initial Setup for Event Processors** documentation.

9. Click the **Next** button, and the wizard step 'Configure New Experian Email Validation Event Processor' will display with detailed configuration instructions.



If a Setup Group was not selected during the wizard step 'Create New Email Revalidation Condition,' then the wizard step 'Configure New Experian Email Validation Event Processor' will display with the following.



10. Click the **Finish** button, the wizard will close, and the Email Component Model will be populated with the selections made during the wizard.

Important: If the wizard was used to create a new event processor, the additional manual steps necessary to set the Queue status and enable the event processor need to be completed. For more Information, see the **Queue Status** section of the **Event Processor Tab** documentation, and the **Enable Event Processor** section of the **Running an Event Processor** documentation.

Experian Email Validation Integration in Web UI

The Experian email validation integration provides easy maintenance of valid email contact data and provides an overview of email data quality through use of the Experian Data Quality asynchronous Clean Web Service (Experian).

Prerequisites

This section of the Experian Email Validation integration requires an understanding of the Web UI, how to create screens, and the necessary user permissions to do so. See the **Web User Interfaces** documentation for more information on these topics.

Additionally, users should be familiar with the Experian Email Validation integration setup and processes. For more information, see the **Experian Email Validation Integration** section of the **Data Integrations** documentation.

Once all setup is complete, the Experian Email Validation integration allows users to perform email validations on a node list with one click of the 'Validate Emails' button. Users can choose one or more customer objects (containing email account data) that need to be validated from a list of emails, then click on the 'Validate Emails' button to run the background process (Experian Email validation Job). In the background, the Experian Email Validation Job will continue to poll for an answer from the Experian batch API and wait for a status message on all emails. Whenever an update is received, some status information will be written into the execution report of the background process, and the corresponding email data quality fields will be updated and visible to the user in the Web UI.

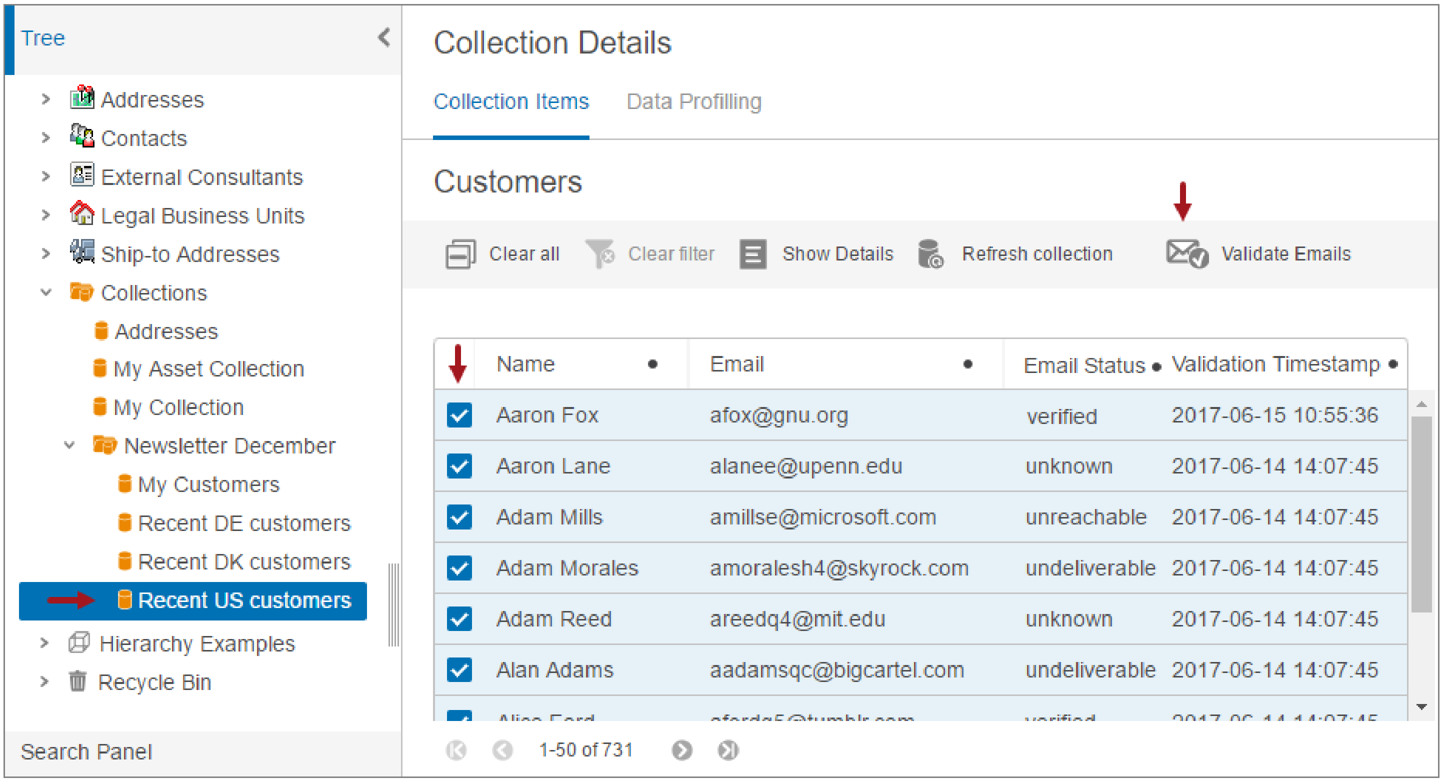
It is possible to monitor the progress of an Experian Email Validation Job using the Background Process Notification component. Also, Email data that is not validated correctly, can be easily identified.

Important: To use the Experian Email Validation Integration functionality, the applicable recipe must be applied, and the Experian License must be obtained from Stibo Systems in order to have access to the Experian API key. Contact your Stibo Systems account manager for more information and licensing terms.

Using Web UI to Validate Email Addresses

Once all setup is complete, users can navigate to a node or collection of objects with email address information stored as data containers, select all necessary objects from the displayed node list, and click the 'Validate Emails' button.

In the example below, the 'Recent US customers' collection is selected and displays the 'Collection Details' screen. On this screen the 'Collection Items' tab is selected and displays the 'Customers' list of objects with the tool bar. The 'Select all' button is used to select all 731 objects within the list which is signified by a check mark within the checkbox. Additionally, the 'Validate Emails' button displays.



When the 'Validate Emails' button is clicked, a background process notification will display (if configured). Since the Experian service does not return immediate responses, it is recommended that the corner bar background notification component be configured so that users can easily follow up with the status of the email data submitted for validation. For more information, see the **Background Process Notification Component** section of the **Corner Bar Component** documentation.

> **Note:** The Experian service is an asynchronous service, aimed at validating large batches of email data. Therefore it will not return immediate responses. The response times depend on the number of emails sent, and email data quality. The guaranteed response times are from 2 (less than 1,000 email addresses) to 20 hours (less than 100,000 email addresses).

When a response is received, one of the following result values will be written into the 'Email Status' email data quality field:

- **(blank):** Email has never been sent for validation. Typically only seen when information is first imported into STEP.
- **Disposable:** Domain is administered by a disposable email provider (e.g., Mailinator).
- **Illegitimate:** Seed, spam trap, black hole, technical role account or inactive domain.
- **Malformed:** The email is identified by STEP as having an illegal format.
- **Undeliverable:** Mailbox or domain does not exist, or mailbox is full, suspended or disabled.
- **Unknown:** We were unable to conclusively verify or invalidate this address.

- **Unreachable:** Domain has no reachable mail exchangers.
- **Verified:** Mailbox exists, is reachable, and not known to be illegitimate or disposable.

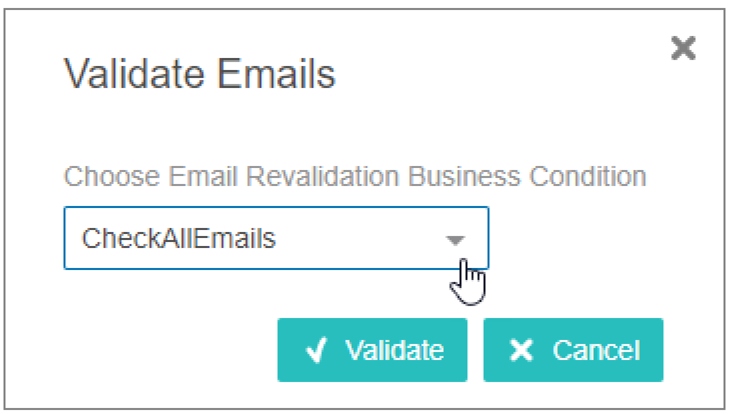
The 'Validation Timestamp' field will be updated with the date and time the email address was last received from the Experian service.

For more information, see the **Experian Email Validation Processing Plugin Parameters and Triggers** section of the **Processing Plugins** documentation.

Validate Emails by Selecting from a List of Revalidation Conditions

Optionally, more than one email revalidation condition can be configured, allowing users to quickly and easily narrow down their Experian email validation request by selecting from common business conditions (i.e., Validate new emails only, Validate new emails and emails that have not been validated in the last 30 days, Validate emails with specific domains).

When more than one email revalidation condition is configured within the designer, and a Web UI user clicks the Validate Emails action button (as described in the previous section), they will be prompted to choose from a dropdown list of available email revalidation business conditions (as shown below).



Once the desired email revalidation condition is selected from the dropdown list, and the user clicks the Validate button, then the background process notification will display (as described in the previous section).

For information on configuring multiple email revalidation business conditions, see the **Configuring a Validate Email Action Button for a Node List** topic within this guide.

Configuring a Validate Emails Action Button for a Node List

The Validate Email action button must be added to a node list and configured with a previously created Experian Email Validation Configuration. Optionally one or more email revalidation business conditions can be added, allowing users to refine the collection of emails selected to be sent for email validation.

Prerequisites

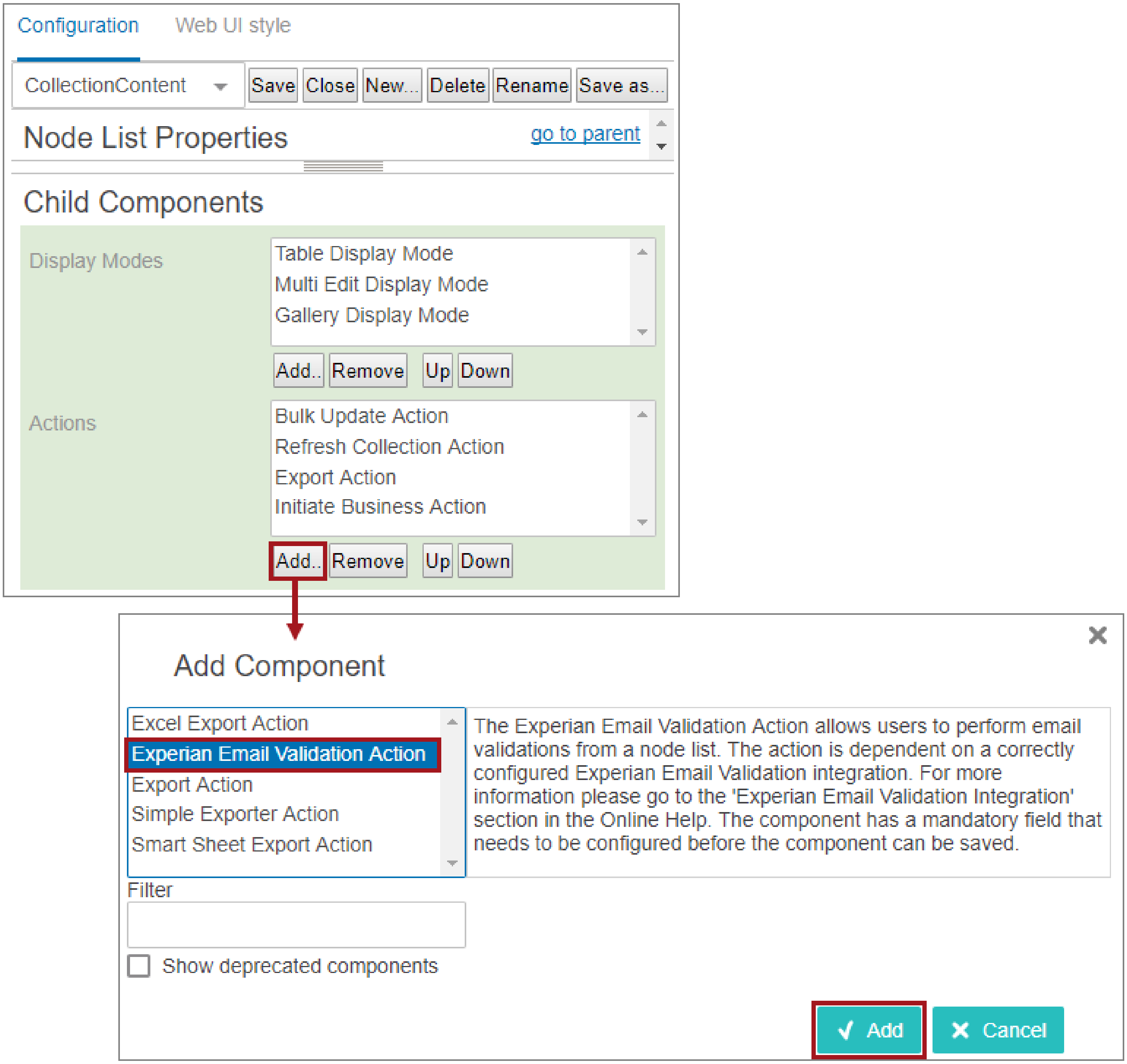
Before configuring a Validate Emails action button, it is recommended to have a firm understanding of the **Experian Email Validation Integration** section of this guide, and to have created the mandatory Experian Email Validation Configuration using the Experian Configuration object type.

For more information about action buttons on Node List, see the **Action Button Configuration on a Node List** section of the **Using a Web UI** documentation.

Add a Validate Email Action Button to a Node List

Steps for adding a Validate Email action button to a node list are below.

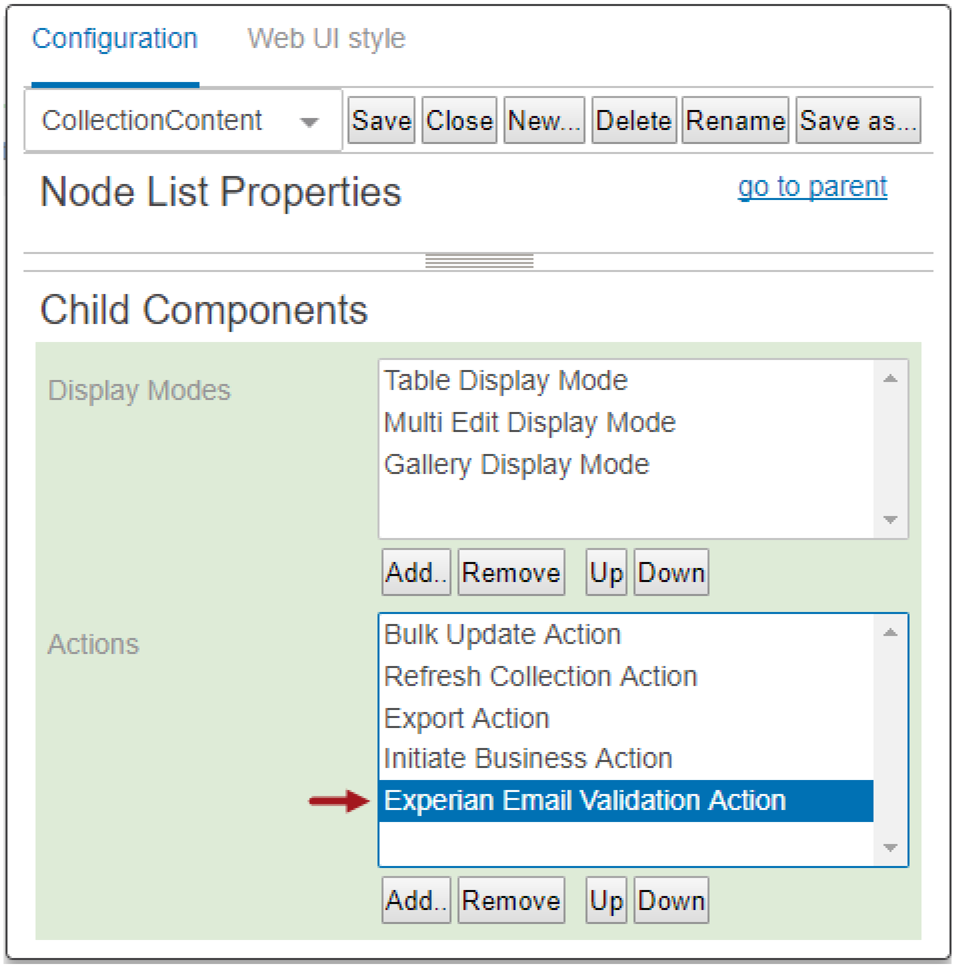
1. Go to your Node List Properties > Child Components > Actions parameter
2. Click the **Add** button, and the Add Component dialog will display.
3. Select **Experian Email Validation Action**, click the **Add** button to close the dialog, and continue to the configuration steps.



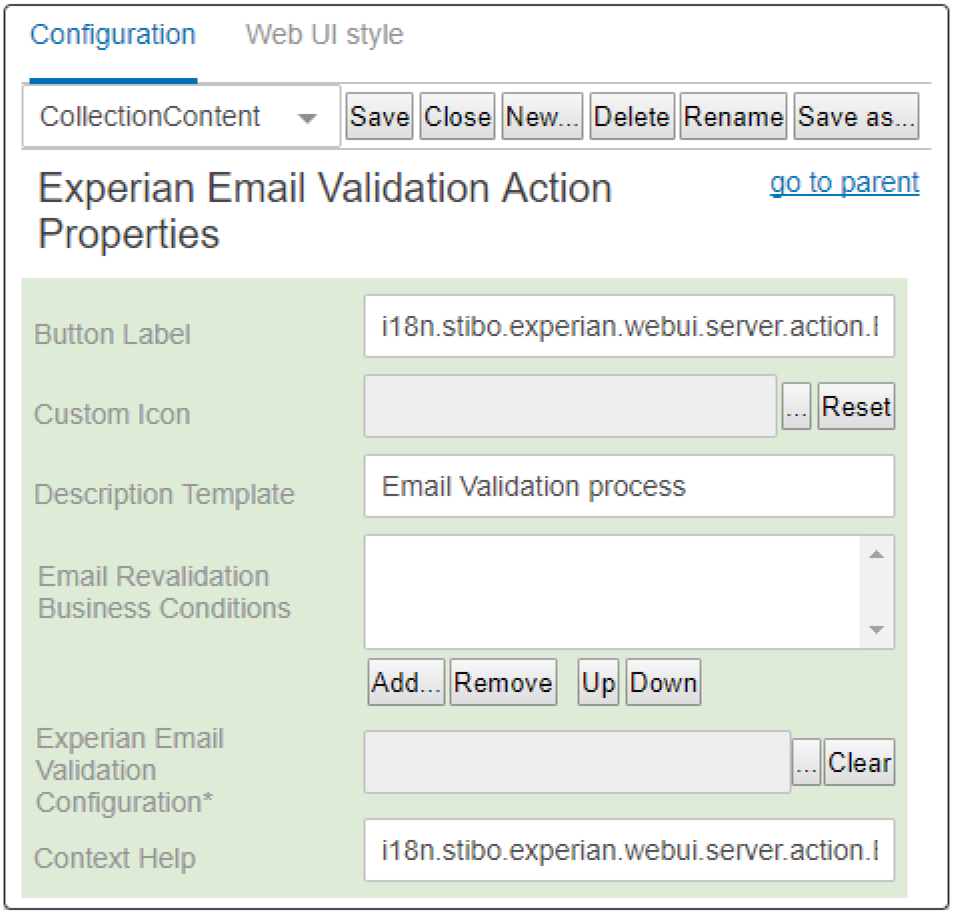
Configure a Validate Email Action Button for a Node List

Once the Experian Email Validation Action component is added to the Node List Actions parameter, follow the steps below to complete the configuration.

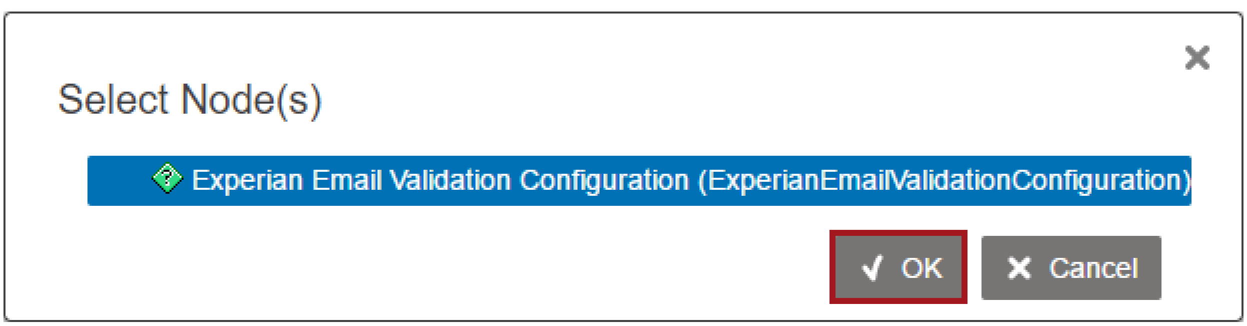
1. From the Actions parameter list, double click the newly added **Experian Email Validation Action** (as shown below).



The Experian Email Validation Action Properties dialog will display as shown below.



2. It is mandatory to populate the Experian Email Validation Configuration parameter with an Experian Configuration object. Click the ellipsis button (...) next to the Experian Email Validation Configuration parameter, and a Select Node(s) dialog will display Experian Configuration object types only.



Note: If the Email Component model easy setup wizard was used to automatically create the Experian Email Validation Configuration object, then the configuration option will appear as shown above.

3. Select **your Experian Configuration object**, and click the **OK** button. The Select Node(s) dialog will close, and the selected Experian Configuration object will display within the Experian Email Validation Configuration

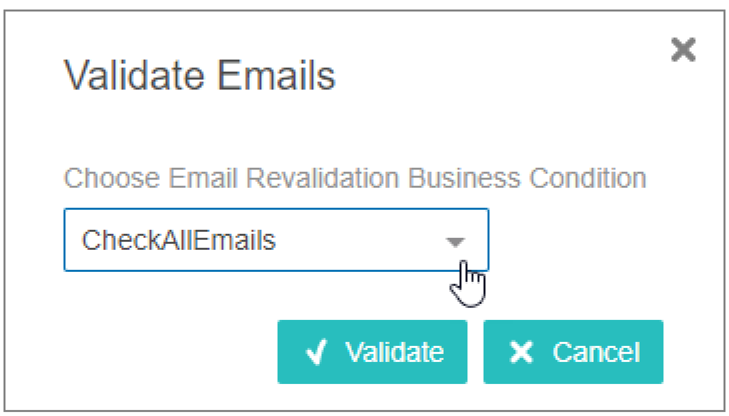
parameter.

4. Click the **Save** and **Close** buttons on the designer.

Optionally, additional parameters within the Experian Email Validation Action properties can be configured. The optional parameters are described below.

Configure Optional Parameters for the Experian Email Validation Action Properties

- **Button Label:** Enter the text to be displayed as the label for the action button icon. The label will only be displayed if the 'Include Label' parameter on the parent Node List properties is enabled.
- **Custom Icon:** Adding a custom icon will overwrite the default icon displayed in Web UI. The recommended size of the icons is 20 (height) x 35 (width) pixels. It is recommended that icons use deep jewel toned colors that fade well when disabled.
- **Description Template:** Enter the text to be used to identify the corresponding background process.
- **Context Help:** Enter text to be displayed when a user hovers over the action button in Web UI.
- **Email Revalidation Business Conditions:** Adding multiple Email Revalidation Business Conditions can help users quickly and easily narrow down their Experian email validation request by selecting from common business conditions (i.e., Validate new emails only, Validate new emails and emails that have not been validated in the last 30 days, Validate emails with specific domains).
  - When business conditions are added to this parameter, they will display for selection when a user clicks the Validate Emails action button in Web UI (as shown below).

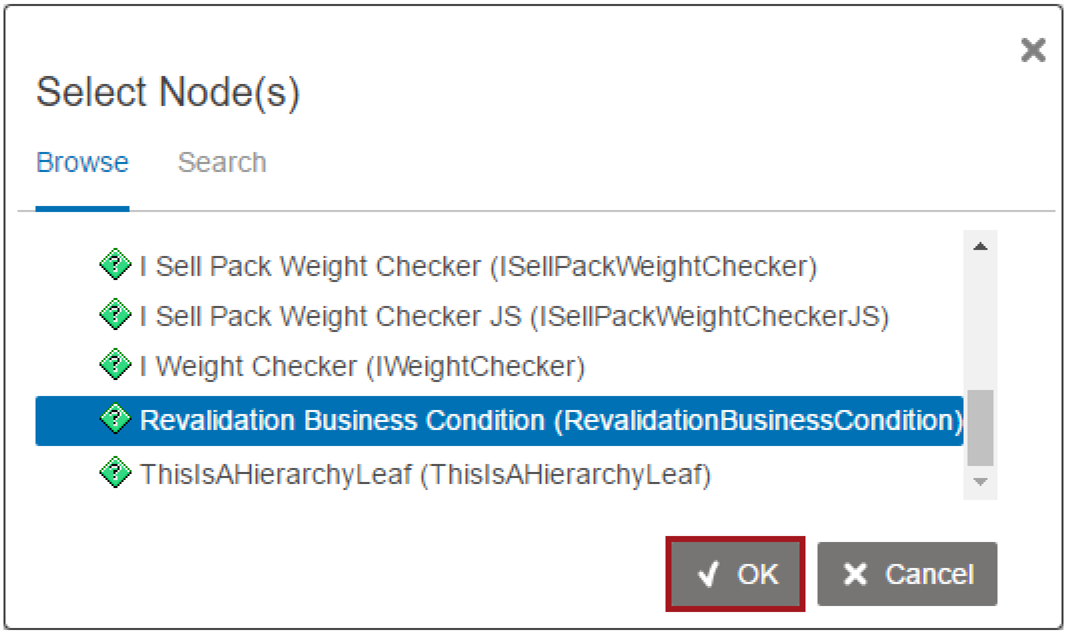


  - When the parameter is empty, and the Web UI user clicks the Validate Emails action button, the user is not prompted to select a business condition, and all selected emails will be sent for email validation.

Add Email Revalidation Business Conditions

Below are steps to add a previously configured business conditions to the Email Revalidation Business Conditions parameter.

1. Click the ellipsis button (...) next to the Email Revalidation Business Condition parameter to display a Select Node(s) dialog.

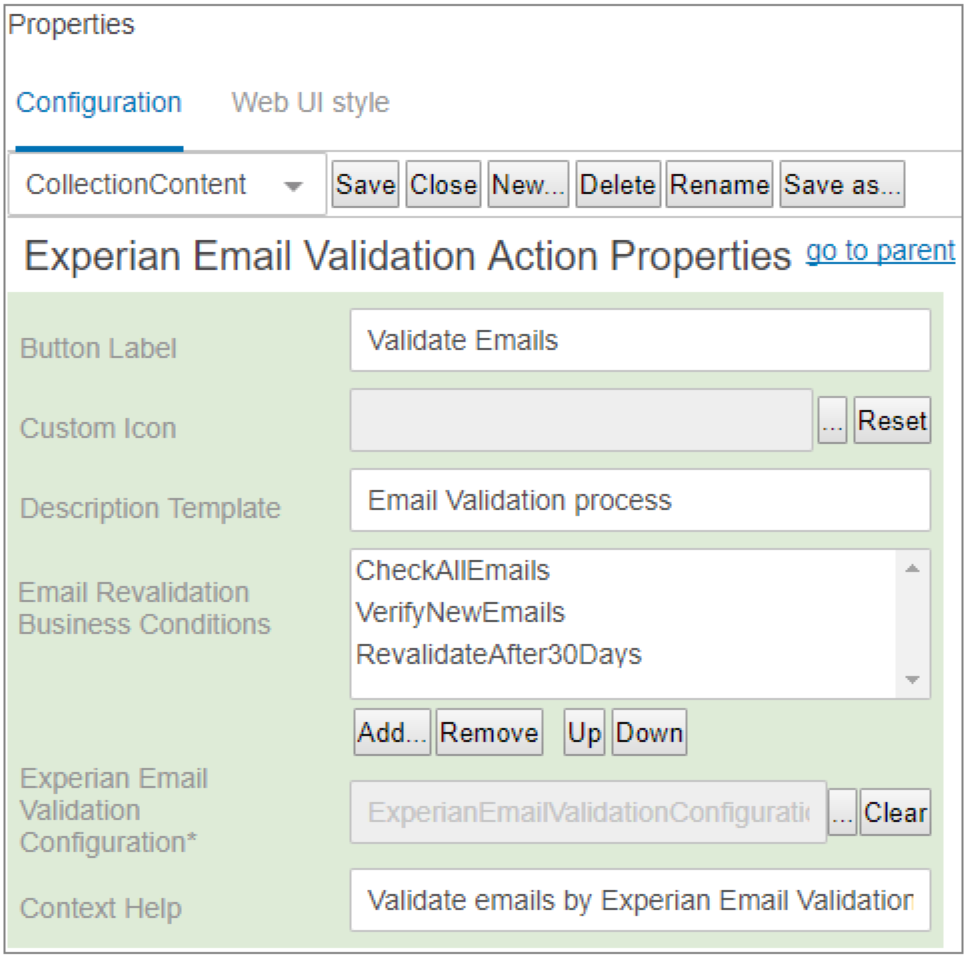


Note: If the Email Component model easy setup wizard was used to automatically create the revalidation business condition, then the condition will appear as shown above.

2. Browse or search to select **your revalidation business condition**.
3. Click the **OK** button, the Select Node(s) dialog will close, and the selected revalidation business condition will display within the Email Revalidation Condition parameter.
4. Optionally, repeat the above steps to add more than one email revalidation business condition.

Note: If one email revalidation business condition is added to the Email Revalidation Business Condition parameter, and the Web UI user clicks the Validate Emails action button, the user will not be prompted to select a business condition, and only the selected emails that comply with the configured email revalidation business condition will be sent for email validation. To allow Web UI users to choose a 'Check All Emails' option along with the added email revalidation business condition, then an additional 'Check All Emails' email revalidation business condition needs to be created and added to the Email Revalidation Business Condition parameter.

Below is an example of configuring multiple email revalidation business conditions.



For information on and images of the Email Validation business condition in the workbench, see the **Email Revalidation Business Condition** section of the **Experian Email Validation Integration** topic in this guide

Loqate Integration

Loqate is a third-party address standardization service that, when integrated with STEP, is used to return standardized addresses that adhere to the standards of local postal authorities such as the USPS. Standardized addresses are essential to securing safe postal deliveries and also help save money when doing mass mailings. A fully standardized address is also a first step in customer deduplication (finding and removing duplicate records containing these addresses) using Matching, Linking, and Merging.

The integration to Loqate services is implemented using the following STEP components:

- Business actions and bulk updates that enable address standardization and verification of addresses
- Address Component Model

Additionally, the integration to Loqate Local with CASS uses the following:

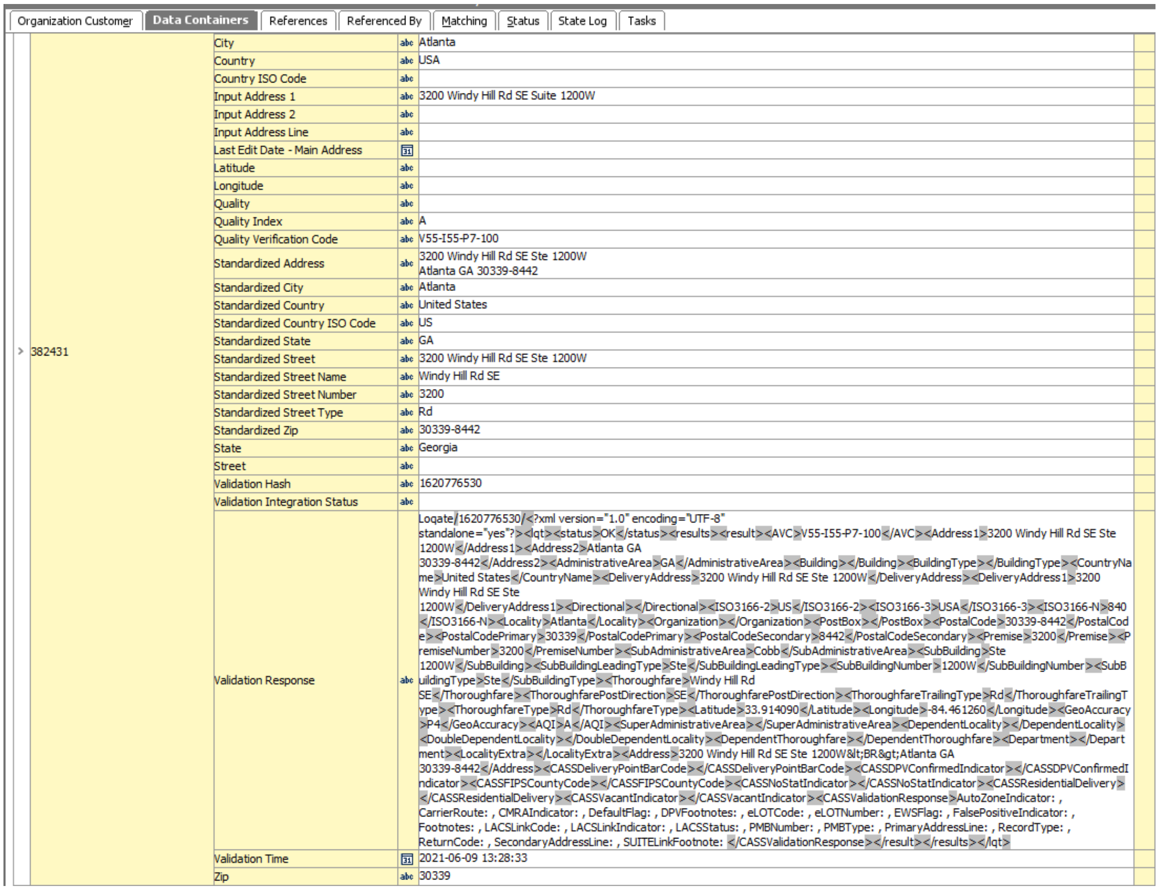
- CASS Address Component Model
- Event processor for the generation of CASS certification reports

To assist with proper configuration, easy setup actions are available to assist in the Loqate address field mappings as well as the creation of associated address object types, business rules, and event processors.

Important: A Loqate integration requires access to the following domains to function properly. For the Loqate Cloud, a Secure Sockets Layer (SSL) connection must be made to http://saas.loqate.com. For the Loqate Local solution, an SSL connection must be made to https://licensing.loqate.com and https://download.loqate.com.

By using Loqate, an address is first standardized and then verified. These are technically two separate processes, but must be executed together to ensure an address is valid. To standardize an address, Loqate will update the address input information from STEP by ensuring that it uses the approved spelling, abbreviations, and formatting of the relevant postal authorities. However, a standardized address alone does not guarantee validity or deliverability. Once the address is standardized, it is matched against Loqate's reference data to be validated. If any city, state, or ZIP code information is missing from the input, Loqate may not be able to uniquely identify and add these to the standardized output. Therefore, the output from Loqate will have quality measures that indicate how valid the address is.

The following screenshot show a simple example of street address and ZIP code values as they look before being sent to Loqate (Input Street and Input ZIP) and as they look after being standardized and returned from Loqate (Standardized Street and Standardized Zip).



STEP integrates with Loqate in three different ways: **Cloud**, **Local**, and **Local with CASS**. These are similar services, but with slight variations. This introductory topic provides a brief overview of these three methods, along with the required configuration properties.

Stibo Systems is a reseller of the Loqate solution, commercially licensing the Loqate software along with its reference data. For information on obtaining and installing any of the three variations of the Loqate address standardization service, contact your Stibo Systems account manager. Additional support information and installation instructions can be found on the Loqate website. (Users must have a Loqate account already set up before they can access the website.)

Loqate Cloud

The Loqate Cloud API service is an integration that enables STEP to communicate with an off-premises Loqate cloud server, hosted by Loqate. Input (non-standardized) address information is stored in STEP in attribute values that are mapped to Loqate fields within the Address Component Model. This information is sent from STEP to the Loqate cloud where it is validated and standardized. Once Loqate parses the data, the standardized address information is returned to STEP and stored in standardized output address fields. Loqate charges users a small fee every time a call is made to the Loqate cloud.

To integrate with the Loqate Cloud solution, STEP users must have the relevant Loqate cloud licenses and API license key.

The Loqate cloud solution may be a preferred option for users who would like to save money on infrastructure cost by using an on-demand service hosted by Loqate. However, the cloud solution has more limited functionality than the local solution. For example, fewer Loqate fields are available in the cloud solution than in the local solution. Also, users must be connected to a local Loqate engine in order to use the CASS program.

See the **Loqate Cloud API** section of the **Loqate APIs** topic.

Loqate Local

The Loqate Local API service is an integration that enables STEP users to standardize address information against reference data stored on a local server, typically on the STEP application server itself. Just like in the cloud service, input address information is stored in attribute values that are mapped to Loqate fields within the Address Component Model. However, instead of being sent to a Loqate-hosted cloud server, the address data is sent from STEP to the locally installed Loqate engine, where it is validated and standardized. Once the Loqate engine parses the data, the standardized address information is returned and stored in standardized output address fields.

To integrate with the Loqate Local solution, STEP users must have a Loqate Local license and API license key. Additionally, users must install Loqate software (provided by Stibo), then make an initial connection to an external Loqate server to download the Loqate reference data packs. Once the software and data packs are locally installed, address information can be sent directly to the Loqate installation on the local server, with no need to connect externally to the Loqate cloud. Users will need to occasionally download updated reference data packs from Loqate, but, on a day-to-day basis, address standardization will be handled by internal communications between STEP and the locally installed Loqate engine.

The Loqate Local solution may be a preferred option for users who prefer a much tighter integration with the local API. The Loqate Local server also boasts improved performance over the cloud solution, with faster response times when bulk handling large amounts of data. The local server can also be used with CASS address standardization functionality, which adds an even stricter layer of address validation.

Important: A considerable amount of server space is needed for Loqate reference data. Approximately 13 GB alone is needed for the US data packs, which include verification datasets and geocode datasets, and approximately 35 GB is needed for the full worldwide reference data set. Loqate reference data must be accessible from each STEP application server. It can be installed on a shared drive, but for best performance, it should be maintained on a hard drive local to each server. If CASS is also activated, then performance may be impacted.

See the **Loqate Local API** section of the **Loqate APIs** topic.

Loqate Local with CASS

The Loqate Local with CASS service is the same installation as the standard Loqate Local API solution but includes additional data packs of CASS reference data and some CASS-specific library files. An additional license is needed for CASS. For more information on CASS, see the **CASS Address Component Model** topic.

Note: The CASS solution does not work with the Loqate Cloud API.

See the **Loqate Local with CASS API** section of the **Loqate APIs** topic.

Address Validation Web Service

If STEP is deployed with Loqate Local, STEP also offers an Address Validation web service. This web service can be used by third-party systems to perform the same address validation and standardization that is done in STEP. This allows consistent address validation across the enterprise.

The Address Validation web service comes in REST and SOAP variants and supports CASS address certification, as applicable, and geocoding. For complete documentation for web services functionality related to Address Validation, click the **STEP API Documentation** button on the STEP Start Page and see the SOAP and REST API sections.

To access the Address Validation Web Service endpoint for a given system, navigate to the following URL: **[your system URL: port]/LoqateWS/loqate**.

Loqate Integration Properties

The following case-sensitive properties must be added to the sharedconfig.properties file to enable Loqate functionality. Customers are responsible for keeping all Loqate Local reference data updated on their servers.

Both Local and Cloud

- Address.AddressQualityPlugin=LoqateAddressQuality
- Address.Service.Loqate.GeocodeAll=[true or false]

  Optional. Default value is true. Specifies if all addresses should be geocoded. A geocode is a pair of coordinates for the latitude and longitude of a location.

Cloud Only

- Address.Service.Loqate.License=[License Key required for Production]

  Specifies the license key of the Loqate Cloud solution.

Local Only

- Loqate.Server.Data=[direct or relative path]

  The directory where data for the Loqate Local API has been installed.

  Examples:

  - Linux: Loqate.Server.Data=/opt/stibo/step/resources/loqate/2.23.0.9136/LoqateData
  - Windows: Loqate.Server.Data=C:/LoqateData

  The below screenshot shows a configuration for Loqate Local on a Linux server. The Address.Service.Loqate.License line is for the Cloud solution.

> **Note:** If the Loqate.Server.Data parameter is set, STEP assumes that a Loqate Local server is also installed regardless if the Address.Service.Loqate.License parameter is set or absent.

```
#==============================================================================#
# Loqate settings
#==============================================================================#
Address.AddressQualityPlugin=LoqateAddressQuality
Address.Service.Loqate.License=
Loqate.Server.Data=L:/loqatedata
```

Loqate APIs

This document explains how to install the Loqate Local API server engine on your local STEP application server, which will enable your STEP instance to reference both the Loqate and CASS (if applicable) address standardization data packs. This documentation will serve as a reference guide on how to install and maintain your Loqate Local server software and Loqate reference data.

STEP integrates with Loqate in three different ways: **Cloud**, **Local**, and **Local with CASS**. These are similar services, but with slight variations. Stibo Systems is a reseller of the Loqate solution, commercially licensing the Loqate software along with its reference data. To function, these solutions require different STEP licenses and Loqate License keys.

Before STEP's integration to Loqate Local and Loqate Local with CASS will become functional, a user must install and maintain the reference data needed for the Loqate Local API to function. This must happen on all STEP Application Servers. This document details the required components, licenses, configuration properties, installation recipes, installation instructions, and more that are needed to get the solution up and running on your STEP installation and to keep its associated reference data up to date.

- The **Loqate Cloud API** service is an integration that enables STEP to communicate with an off-premises Loqate cloud server, hosted by Loqate. The Cloud API is included with STEP baseline. To integrate with the Loqate Cloud solution, STEP users must have the relevant Loqate cloud licenses and API key.

  The Loqate cloud solution may be a preferred option for users who would like to save money on infrastructure cost by using an on-demand service hosted by Loqate. However, the cloud solution has more limited functionality than the local solution. Additionally, Loqate charges users a small fee every time a call is made to the Loqate cloud.

- The **Loqate Local API** service is an integration that enables STEP users to standardize address information against reference data stored on a local server, typically on the STEP application servers itself. This document focuses on the installation and maintenance of the Loqate Local solution (with or without CASS).

  Just like in the cloud service, input address information is stored in attribute values that are mapped to Loqate fields within the Address Component Model. However, instead of being sent to a Loqate-hosted cloud server, the address data is sent from STEP to the locally installed Loqate engine, where it is validated and standardized. Once the Loqate engine parses the data, the standardized address information is returned and stored in standardized output address fields.

  To integrate with the Loqate Local solution, STEP users must have a Loqate Local license (a STEP License) and API license key. Additionally, users must install Loqate software: This is provided by Stibo Systems in a separate STEP Component called “local-loqate”). Then users must make an initial connection to an external Loqate server to download the Loqate reference data packs: This is done using the software distributed in the local-loqate component.

  Once the software and data packs are locally installed, address information can be sent directly to the Loqate installation on the local server, with no need to connect externally to the Loqate cloud. Users will need to occasionally download updated reference data packs from Loqate, but, on a day-to-day basis, address standardization will be handled by internal communications between STEP and the locally installed Loqate engine.

- The **Loqate Local with CASS** service is the same installation as the standard Loqate Local API solution but includes additional data packs of CASS reference data and some CASS-specific library files.

  An additional license is needed for CASS, and CASS users must be based in the United States, as CASS is not valid outside of the US.

  To support CASS on windows servers, the local-loqate component distributes the 32-bit version of the Loqate API.

Prerequisites

The following subsections explain the prerequisites that must be met before installing the Loqate Local API and associated reference data on your server, which typically will be on the same server as your STEP application server.

System Requirements

A considerable amount of server space is needed for Loqate reference data. Approximately 13 GB alone is needed for the US data packs, which include verification datasets and geocode datasets, and approximately 35 GB is needed for the full worldwide reference data set.

Note: Ensure that you have temporary space for downloading the data and permanent space for unpacking and storing the data. Additionally, during the reference data installation Loqate will offer to make a backup of any existing reference data, which is optional. In other words, make sure you have plenty of space and/or be prepared to add disk space as needed. The Loqate installer will provide warnings if there is not enough space.

Loqate reference data must be accessible from each STEP application server. It can be installed on a shared drive, but for best performance, it should be maintained on a hard drive local to each server. If CASS is also activated, then performance may be impacted.

Required Licenses

To run the Loqate Local functionality, the Address Validation Service (Loqate Server Adapter or Loqate Cloud Adapter) commercial license must be enabled for your system. If you have a Customer Domain or Location Domain license, then you have access to the Address Component Model and the Loqate Cloud functionality. For users of CASS, both the Address Component Model and the CASS Address Component Models will display.

Loqate License Keys

To obtain Loqate license keys, contact your Stibo Systems Account Manager or Sales person.

Note: The Loqate License keys are not interchangeable between Loqate Local and Loqate Cloud.

Install Recipes

A separate install recipe is needed to integrate STEP with the Loqate Local API and CASS components required to use the address standardization and validation services of Loqate Local with STEP.

Note: Though the same recipe is used for both Loqate Local and CASS, CASS is only available to users in the US who have purchased an additional CASS license.

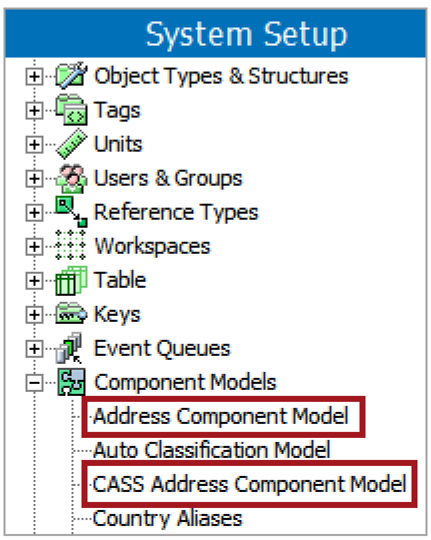
The Loqate recipes can be found in Prism or via the Internal STEP License Mapping page. Pick the correct release version and look for the link within the Release Notes and Materials section.

Note: Users should select the recipe compatible with their STEP system and always verify that it is the correct recipe before install.

Once installed, the Loqate (and CASS, if applicable) software will be available on the STEP system. However, additional configurations must be made and data installed to round out the installation.

Included in these configurations are the setups of the Address Component Model and CASS Address Component Model, shown in the following screenshot. For setup instructions, refer to the following sections of the STEP online help documentation:

- Data Integration > Loqate Integration > **Address Component Model**
- Data Integration > Loqate Integration > **CASS Address Component Model**



Network Access Requirements

To gain access to the Loqate service, the following network access is required between your STEP application server and the Internet:

- Loqate Cloud: HTTP access to saas.loqate.com
- Loqate Local: HTTPS access to licensing.loqate.com and download.loqate.com and data.loqate.com – These are required to download the Loqate data packs (reference data)

The following commands can be useful to test if the required network access is available:

- ping saas.loqate.com
- ping licensing.loqate.com
- ping download.loqate.com
- ping data.loqate.com

The output should look something like this:

```
[stibosw@brje-step ~]$ ping data.loqate.com
PING data.loqate.com (52.84.213.46) 56(84) bytes of data.
64 bytes from server-52-84-213-46.arn53.r.cloudfront.net (52.84.213.46): icmp_seq=1 ttl=245 time=13.3 ms
64 bytes from server-52-84-213-46.arn53.r.cloudfront.net (52.84.213.46): icmp_seq=2 ttl=245 time=13.3 ms
64 bytes from server-52-84-213-46.arn53.r.cloudfront.net (52.84.213.46): icmp_seq=3 ttl=245 time=13.3 ms
64 bytes from server-52-84-213-46.arn53.r.cloudfront.net (52.84.213.46): icmp_seq=4 ttl=245 time=13.3 ms
```

Address Component Model

To use the Loqate and CASS integrations, the Address Component Model must first be configured in STEP Workbench.

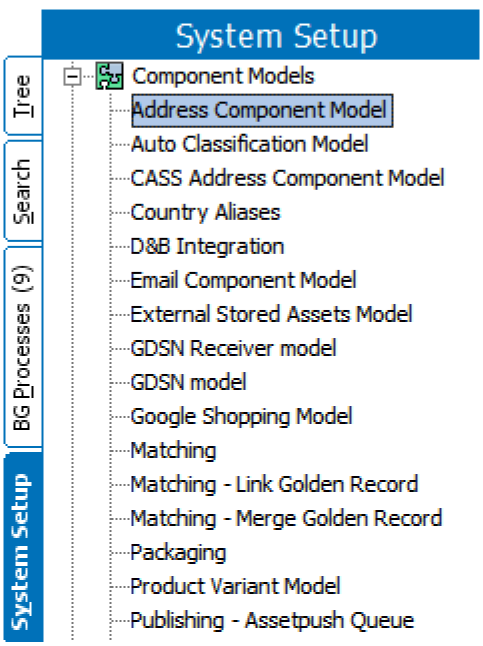
This topic provides an overview of the Address Component Model, including full descriptions of all of the Loqate address attribute fields it contains. For instructions on how to configure the component model, see the following topics in this documentation section:

- **Easy Setup of Address Component Model**
- **Manual Setup of Address Component Model** (recommended for modifications after initial setup)

Once the component model has been configured, the system will be ready to standardize addresses using the Loqate solution.

The Address Component Model determines the input and output fields for the information that is sent to Loqate from STEP and returned from Loqate to STEP.

The Address Component Model is found under Component Models on the System Setup tab. If the Address Component Model is not present, contact your Stibo Systems account manager.



Prerequisites

Configuring the Address Component Model requires users to be familiar with the System Setup tab in STEP Workbench and how to configure within this area (e.g., creation and maintenance of object types, attributes, and references). Users must also have the privileges required to carry out these tasks. Additionally, the following condition must be met:

- Users must be connected to a Loqate installation, either through the Loqate Cloud API or the Loqate Local API. For more information, see the **Loqate Integration** topic in the **Data Integration** documentation.

Loqate Address Fields

The following table lists the fields contained in the Address Component Model, along with a description of each field, the corresponding Loqate field, whether the field is available for both the Loqate Cloud service and the Loqate Local installation. Users will typically not use all address fields.

Loqate fields are mapped to STEP attributes, either manually or though easy setup of the component model. The component model contains four types of fields and are identified in the table below:

- **Object types** – The address object types used to represent addresses
- **Regular** address attributes – The **input** fields that are used as input when making a request to Loqate. These are the non-standardized, original address attributes.
- **Standardized** address attributes – The **output** fields that store the data returned from Loqate. These are the standardized and validated versions of the address fields from Loqate.
- **Data quality** attributes – The fields that store information about the validity and precision of the standardized address attributes.

Loqate responses do not overwrite the input fields, which remain as-is after standardized address data is returned. Loqate responses are kept separately in the standardized fields. For more thorough descriptions of these fields, refer to the Loqate website.

Address Component Model Field	Description	Local Loqate Field Mapping
Address Object Types and/or Address Data Container Types		
Address	The address object types used to represent addresses; can be either entities or data containers	N/A
Regular Address Attributes (INPUT fields)		
Country ISO Code	The ISO 3166 2-character country code	*Not mapped to* Loqate
Input Address 1	Input address line field, for one part of an address, e.g., street number and name	Address1 [in]
Input Address 2	Input address line field, for one part of an address, e.g., city, state abbreviation, postcode	Address2 [in]
Input Address 3	Input address line field, for one part of an address, e.g., country	Address3 [in]
Input Address 4	Input address line field, for one part of the full address Typically used if optional information is put into one of the other input Address1 - 3 fields	Address4 [in]
Input Address Line	Single line input field for an entire address	Address [in]

Address Component Model Field	Description	Local Loqate Field Mapping
Input Building	Input field for name identifying an individual location, e.g., a building	Building [in]
Input City	Input field for name of a large population center, e.g., city or municipality	Locality [in]
Input Country	Input field for country name or code	Country [in]
Input County	Input field for small geographic unit within a country, e.g., county.	SubAdministrativeArea [in]
Input Dependent Locality	Input field for a small geographic unit within a city, e.g., neighborhood	DependentLocality [in]
Input Dependent Street	Input field for street information that depends on an adjoining road	DependentThoroughfare [in]
Input Organization	Input field for business name associated with location	Organization [in]
Input PostBox	Input field for post box for a location	PostBox [in]
Input State	Input field for name of a geographic unit within a country, e.g., state or province	AdministrativeArea [in]
Input Street	Input field for street information, e.g., street name and number	Thoroughfare [in]
Input Street Name	Input field for street name	ThoroughfareName [in]
Input Street Number	Input field for street number identifying an individual location	Premise [in]
Input Subbuilding	Input field for secondary identifier of an individual location, e.g., flat or suite	SubBuilding [in]
Input Zip	Input field for complete postal code	PostalCode [in]
Latitude	The address latitude	*Not mapped to* Loqate
Longitude	The address longitude	*Not mapped to* Loqate
Standardized Address Attributes (OUTPUT Fields)		
Geocode Latitude	Output field containing the address latitude	Latitude [out]

Address Component Model Field	Description	Local Loqate Field Mapping
Geocode Longitude	Output field containing the address longitude	Longitude [out]
Standardized Building	Output field for name identifying an individual location, e.g., a building	Building [out]
Standardized City	Output field for a large population center name, e.g., city or municipality	Locality [out]
Standardized City Extra	Output field for supplemental information related to city	LocalityExtra [out]
Standardized Country	Output field containing country name or code	CountryName [out]
Standardized Country ISO Code	Output field containing the ISO 3166 2-character country code	ISO3166-2 [out]
Standardized Country ISO 3 Character Code	Output field containing the ISO 3166 3-character country code	ISO3166-3 [out]
Standardized County	Output field for small geographic unit within a country, e.g., county	SubAdministrativeArea [out]
Standardized Department	Output field for department name associated with an organization	Department [out]
Standardized Dependent Street	Output field for street information that depends on adjoining road	DependentThoroughfare [out]
Standardized Double Dependent Locality	Output field for small population center within a city, e.g., village.	DoubleDependentLocality [out]
Standardized Formatted Address	Output field for the address formatted for mailing usage, formatted using CRLF ("\n") as the line break	Address [out]
Standardized Neighbourhood	Output field for population center within a city, e.g., neighborhood	DependentLocality [out]
Standardized Organization	Output field for business name associated with location	Organization [out]

Address Component Model Field	Description	Local Loqate Field Mapping
Standardized PostBox	Output field for post box for a location	PostBox [out]
Standardized Region	Output field for the largest geographic unit within a country	SuperAdministrativeArea [out]
Standardized State	Output field for a geographic unit within a country, e.g., state or province	AdministrativeArea [out]
Standardized Street	Output field for street name	DeliveryAddress [out]
Standardized Street Name	Output field for street name	ThoroughfareName [out]
Standardized Street Number	Output field for street number	PremiseNumber [out]
Standardized Street Type	Output field for street type, e.g., Rd for road and St for street	ThoroughfareType [out]
Standardized SubBuilding	Output field for secondary identifier for an individual location, e.g., flat or suite	SubBuilding [out]
Standardized Zip	Output field for complete postal code	PostalCode [out]
Quality Measures		
Geocode Accuracy	Output field for Geocode Accuracy. This field indicates the precision level of the geocode that has been assigned to an address by Loqate.	GeoAccuracy [out]
Geocode Distance	Output field for the Geocode Distance. This field indicates the uncertainty in the physical location of the address.	GeoDistance [out]
Quality	Quality attribute for address. Validation base type is text; attribute is a calculated attribute based on other quality fields. For example, the calculation could be a combination of the values of Quality Index and Quality Verification Code. Users are responsible for writing this formula.	*Not mapped from Loqate*
Quality Index	Output field for the address quality index, which is used to indicate the quality of an address.	*Address Quality Index (AQI) response from Loqate*
Quality Verification Code	Output field for the Address Verification Code. This is used to indicate the level of verification of an address.	*Address Verification Code (AVC) response from Loqate*

Address Component Model Field	Description	Local Loqate Field Mapping
Validation Hash	Hash value for the address validation integration. Contains a hash value of all input fields. This makes it possible to determine if the address has changed since the last validation Field is updated whenever the standardized address attributes are updated by the Loqate integration. Field is calculated by STEP and is not mapped to Loqate.	*Not mapped from Loqate*
Validation Integration Status	Contains the latest status of the address validation integration. Field is updated whenever the standardized address attributes are updated by the Loqate integration. Indicates if the last address validation was completed or resulted in an error. Valid values are: <empty> and Failed. This field makes it possible to search for accounts and addresses where an error has occurred during the address validation.	*Not mapped from Loqate*
Validation Response	Output field for the cached validation response. This field contains the full response from Loqate. Validation base type is text; maximum length must be at least 1000.	*Contains the full response from Loqate*

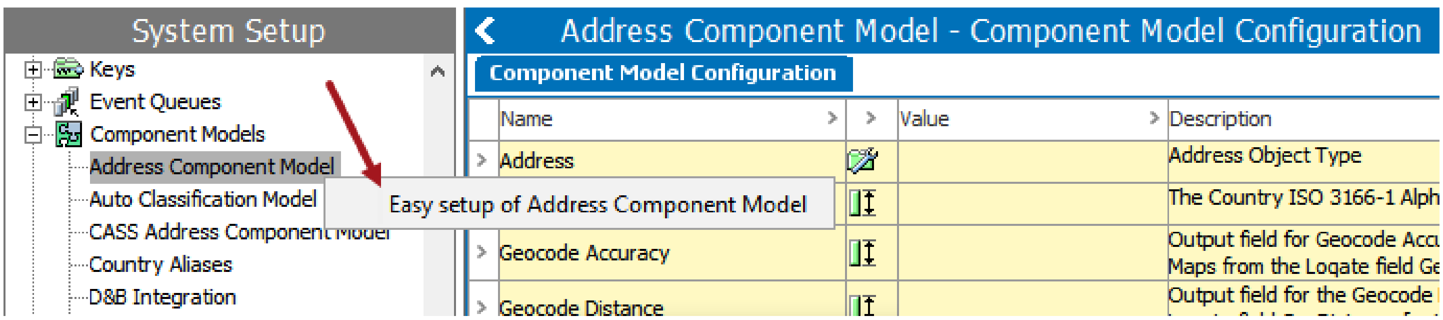
Address Component Model Field	Description	Local Loqate Field Mapping
	> Validation Response abc Loqate/1379196886/<?xml version="1.0" encoding="UTF-8"?> <lqt> <status>OK</status> <results> <result> <AQI>A</AQI> <AVC>V44-I44-P7-100</AVC> <Address1>3550 George Busbee Pkwy NW</Address1> <Address2>Kennesaw GA 30144-6608</Address2> <AdministrativeArea>GA</AdministrativeArea> <CountryName>United States</CountryName> <DeliveryAddress>3550 George Busbee Pkwy NW</DeliveryA <DeliveryAddress1>3550 George Busbee Pkwy NW</Delivery <GeoAccuracy>P4</GeoAccuracy> <GeoDistance>0.0</GeoDistance> <HyphenClass>A</HyphenClass> <ISO3166-2>US</ISO3166-2> <ISO3166-3>USA</ISO3166-3> <ISO3166-N>840</ISO3166-N> <Latitude>34.040720</Latitude> <Locality>Kennesaw</Locality> <Longitude>-84.573000</Longitude> <MatchRuleLabel>1a</MatchRuleLabel> <PostalCode>30144-6608</PostalCode> <PostalCodePrimary>30144</PostalCodePrimary> <PostalCodeSecondary>6608</PostalCodeSecondary> <Premise>3550</Premise> <PremiseNumber>3550</PremiseNumber> <SubAdministrativeArea>Cobb</SubAdministrativeArea> <Thoroughfare>George Busbee Pkwy NW</Thoroughfare> <ThoroughfareName>George Busbee</ThoroughfareName> <ThoroughfarePostDirection>Nw</ThoroughfarePostDirectio <ThoroughfareTrailingType>Pkwy</ThoroughfareTrailingTyp <ThoroughfareType>Pkwy</ThoroughfareType> </result> </results> </lqt>	
Validation Time	Date and time stamp of the most recent successful address validation. Validation base type is 'ISO Date and Time.' Field is updated whenever the standardized address attributes are updated by the Loqate integration.	*Not mapped from Loqate*

Easy Setup of Address Component Model

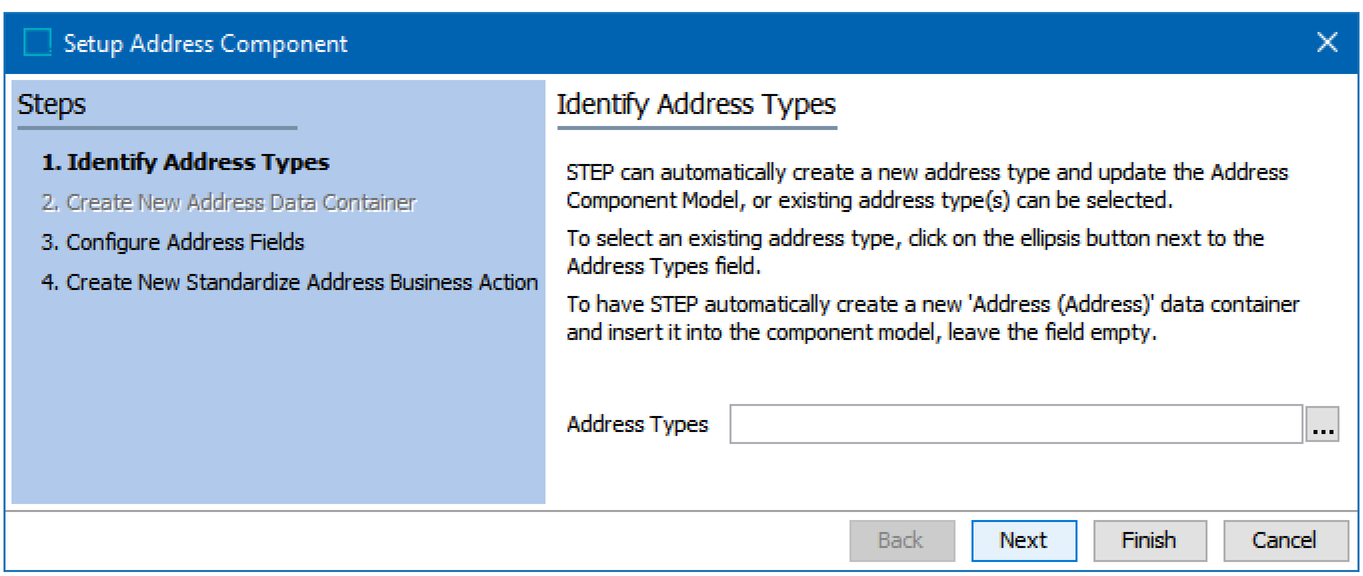
The recommended method for initially configuring the Address Component Model is to use the 'Easy setup of Address Component Model' wizard. By using this wizard, STEP can automatically create all of the address attributes required for the solution while simultaneously mapping them to the corresponding Loqate fields. Additionally, the wizard can create an address data container to house the attributes and a 'standardize address' business action. By using this wizard, little manual action needs to be taken to complete the configuration, making it a simple and straightforward way to get the solution up and running quickly.

The following steps describe how to configure the Address Component Model using the easy setup method. Also note that detailed help text is present on each step of the wizard.

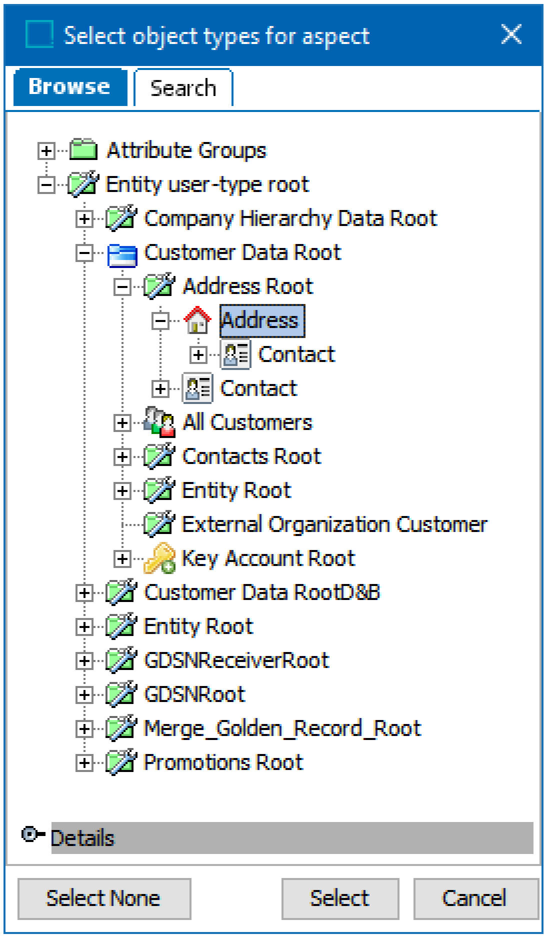
1. Locate the Address Component Model under System Setup > Component Models.
2. Right-click on the Address Component Model and click 'Easy setup of Address Component Model.'



3. On the **Identify Address Types** screen, the system provides the opportunity to automatically create a new address data container object type if one does not already exist.



If you already have an object type (or types) that you would like to use, click the ellipsis button (...) next to the Address Types field and select the relevant object type(s) from the 'Select object types for aspect' dialog. Note that only entities and or data containers can be selected. The below screenshot shows the dialog with a preexisting 'Address' entity object type selected. To multi-select objects, click and hold the Ctrl or Shift key. Click **Select** to choose the object(s) and close the dialog.



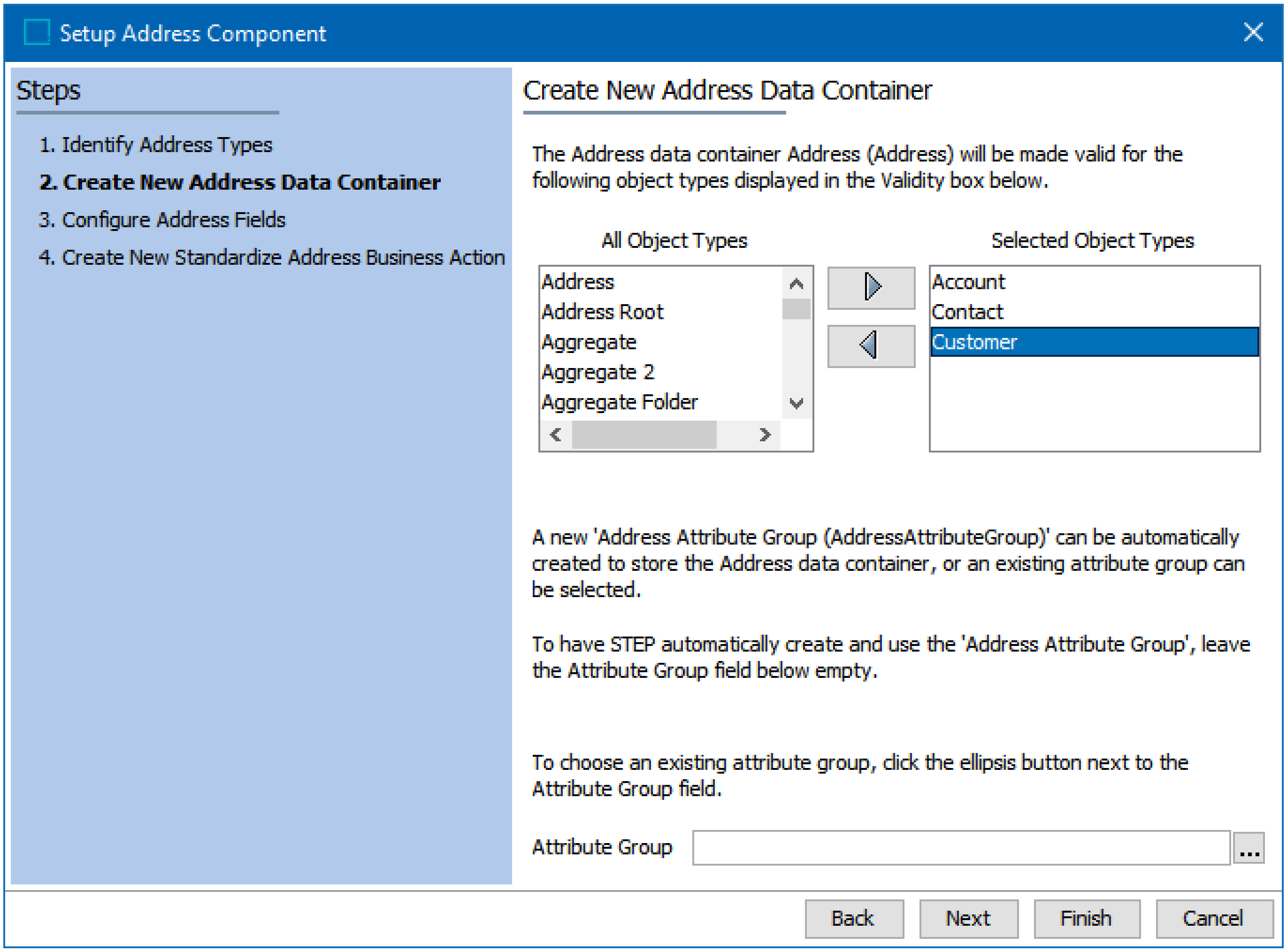
If you prefer to have STEP create the new object type, which will be named Address (ID = Address), leave the 'Address Types' field blank.

Click **Next**.

> **Note:** If 'Address Types' is left blank in this step, the **Create New Address Data Container** screen will display after clicking Next (see step 4 below). If a previously existing address object type is selected, the **Configure Address Fields** screen will display (see step 5 below).

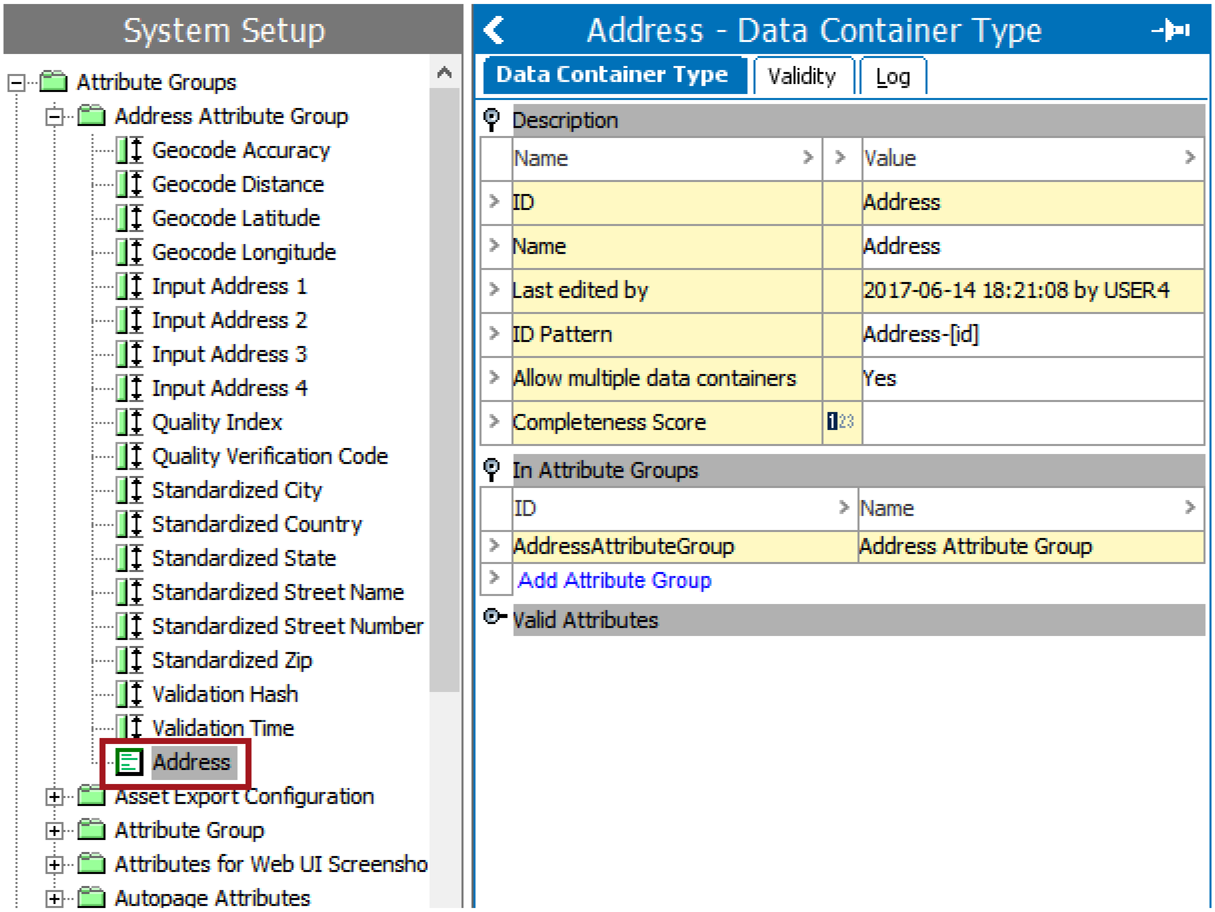
4. If the **Create New Address Data Container** screen is displayed, in the 'All Object Types' window, select the entity object types that should be valid for the data container that is created in this step. Click the top arrow to move the selections into the 'Selected Object Types' window. To remove an object from the 'Selected Object Types' window, click the bottom arrow to move the selection back to the 'All Object Types' window.

At the bottom of the screen, click the ellipsis button (...) to select the attribute group where you would like to store the data container. Leave the field blank to have STEP automatically create an attribute group, which will be named 'Address Attribute Group' (ID = AddressAttributeGroup).



After completing this step, the Address data container will be created and placed into the designated attribute group. The ID pattern will be Address-[id], and 'Yes' will be selected by default for 'Allow multiple data containers.' Any attributes chosen for auto-creation in the next step of the easy setup wizard (Configure Address Fields) will be made valid for this data container.

Note: Any address attributes that exist prior to the easy setup of the Address Component Model will not be modified by the easy setup action. I.e., they will not automatically become valid for the data container created in this step and must be manually linked to it later. For more information, see the **Setting Up Data Container Types in Workbench** topic in the **Data Containers** documentation.



5. On the **Configure Address Fields** screen, attributes are mapped to corresponding Loqate input and output fields. Mandatory attributes are indicated by an asterisk (*) in the second column. (Some fields are not mapped to Loqate, such as 'Country ISO Code', but these fields are optional.) To select or remove the displayed mapping to a STEP attribute, click on the corresponding ellipsis button (...) in the Value column. For full information on the range of fields available in the component model, see the **Address Component Model** topic.

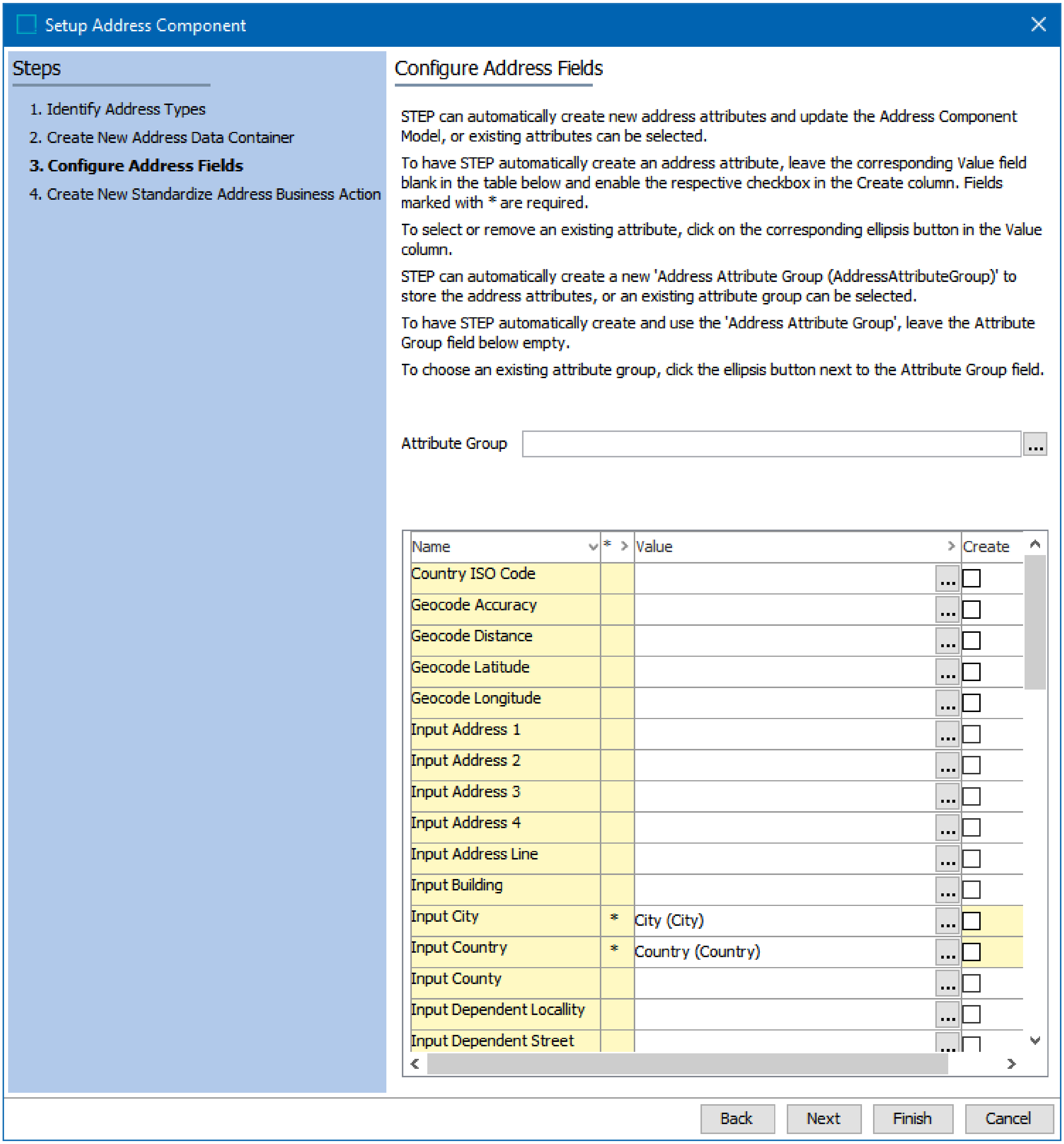
   To have STEP automatically create an address attribute, leave the corresponding Value field blank and enable the respective checkbox in the Create column. All automatically created attributes will be description attributes and be made valid for the configured address types. They will have a name equal to the field name and an ID equal to the field name with spaces removed. For example, for the 'Standardized Double Dependent Locality' field, the automatically created attribute will be named 'Standardized Double Dependent Locality' and have the ID of 'StandardizedDoubleDependentLocality.'

   It is not required to have STEP create the attributes, but is recommended. If any of the attributes already exist (based on ID) and the Create box is checked, a message will display to inform the user that the attribute already exists.

> **Note:** No 'select all' option is available for the Create column; this omission is intentional, as it forces users to give more thought as to which attributes they actually need instead of creating all attributes at once and

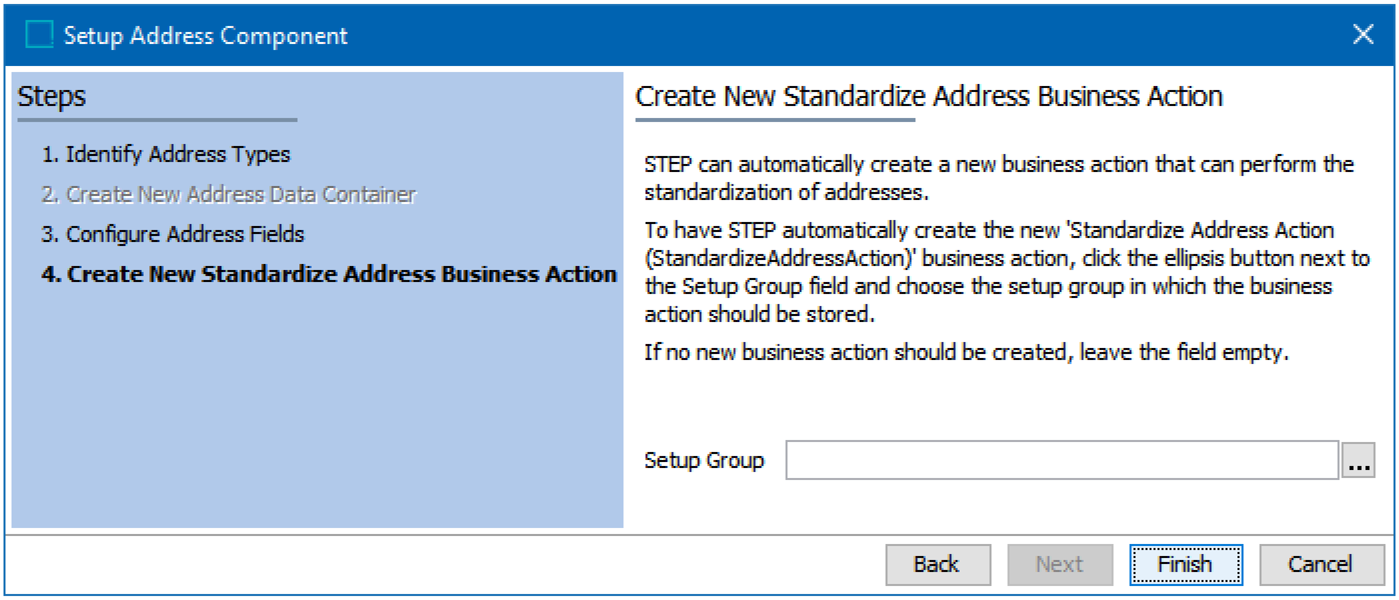
having too many. The complete set of input and output fields cover several address formats, so it is important that users consider select and/or create the attributes that match their business needs. For example, if there is a need for importing single-line addresses, then the 'Input Address Line' field should be activated.

To select an attribute group where you would like to store the newly created attributes, click the ellipsis button (...) next to the Attribute Group field to launch the 'Select Attribute Group' dialog. Leave the field blank to have STEP automatically create the attribute group, which will be named 'Address Attribute Group' (ID = AddressAttributeGroup).

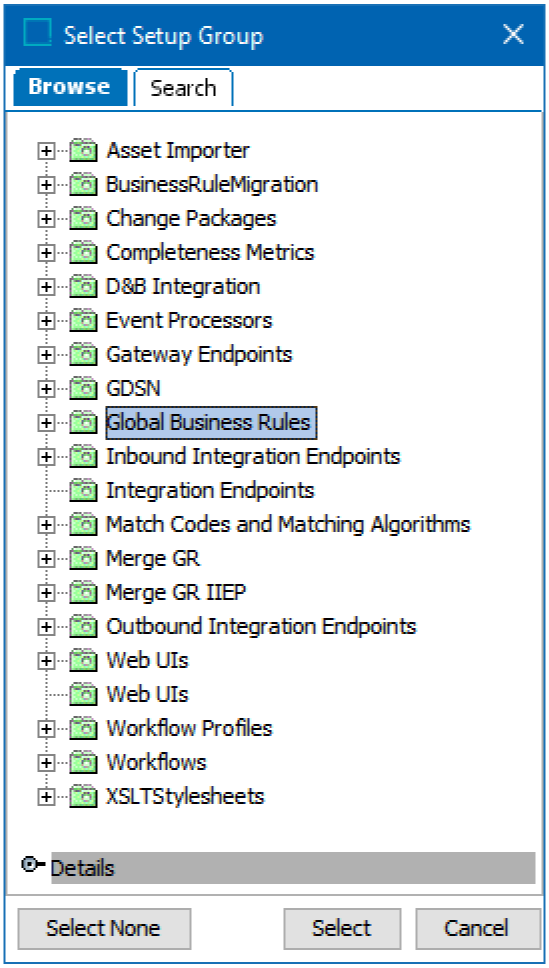


Click **Next** when you have finished mapping your attributes.

6. On the **Create New Standardize Address Business Action** screen, if you would like to have STEP automatically create a 'Standardize Address' business rule, click the ellipsis button (**...**) next to the Setup Group field and select the setup group to house the action. Leave the field blank and click **Finish** if you do not want to create the business rule or would like to create it at a later time.



Click the ellipsis button (**...**) to open the 'Select Setup Group' dialog. Choose the setup group in which you would like to store the newly created business rule, then click **Select** to close the dialog.



For more information on how to complete the configuration of this business action after it is created during the easy setup process, see the **Business Action: Standardize Address** topic in the **Business Rules** documentation.

7. Click **Finish** to complete the wizard and create the business action.

Manual Setup of Address Component Model

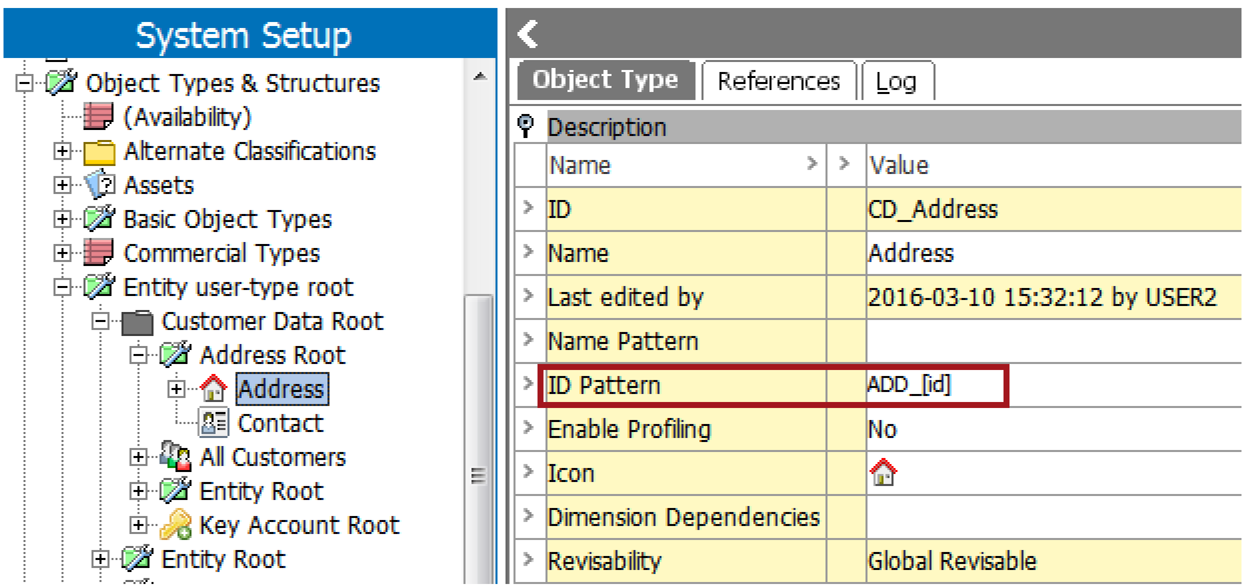
To manually set up the Address Component Model, all required object types and attributes must first be created in the workbench. Then, these attributes must be manually mapped to the corresponding Loqate fields in the component model.

Since every screen in the 'Easy setup of Address Component Model' wizard has an option where users can select preexisting object types, attributes, attribute groups, and setup groups, it can be expected that the full configuration of the Address Component Model may be a combination of both easy setup and manual setup. It is also useful to understand how to manually work with the component model since certain one-off operations, like changing an attribute mapping, may need to be performed after the easy setup is complete.

Data Model Considerations and Requirements

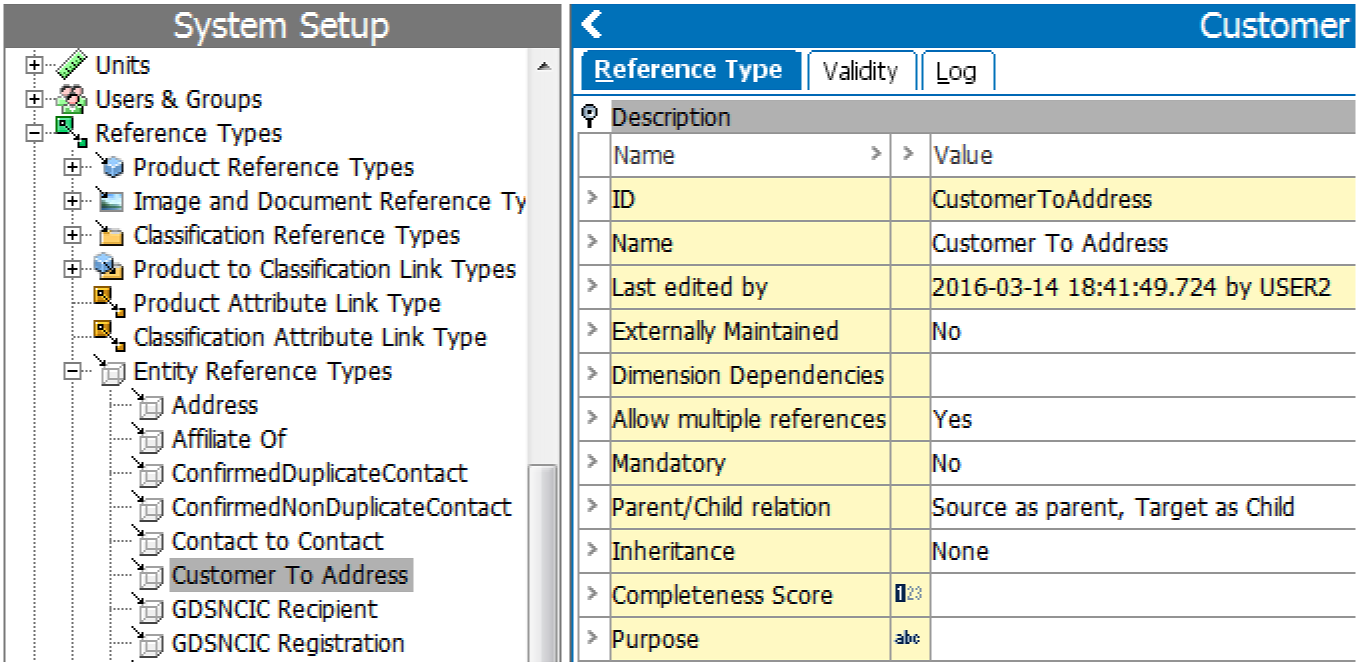
To manually set up the Address Component Model in the workbench, some basic data must be in place. Each element that is required to complete the setup is listed below.

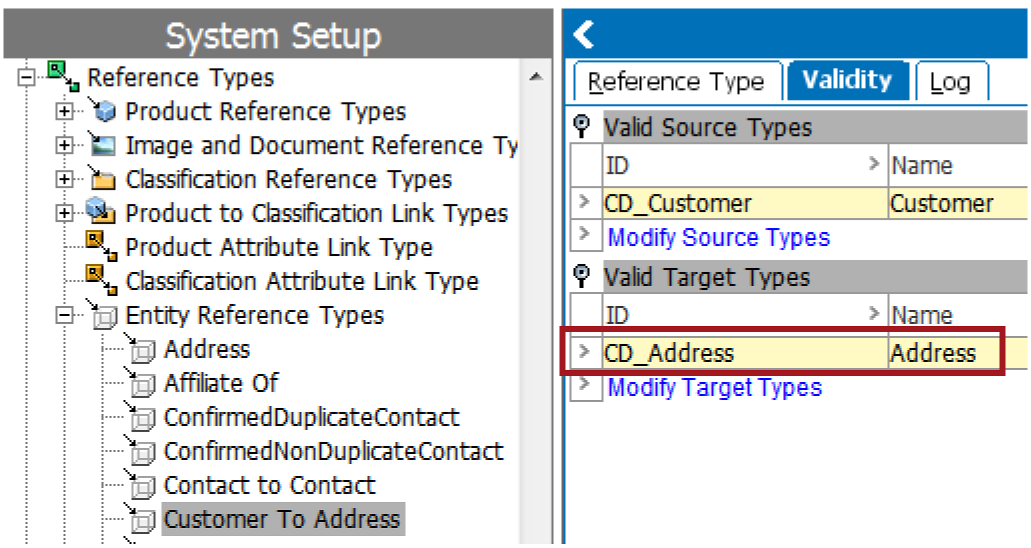
1. If you plan to use an Address object type (instead of an address data container), it must use an auto-generated ID format (e.g., [id]). An example is shown below.



> **Important:** Data container types can also be used and mapped as an Address object type. The attributes mapped to the component model fields must be valid for the specified data container type or types.

2. A reference must exist between the address object type and an associated object type (e.g., supplier, customer, location). The address object must be the *target* of the reference. An example is shown below.





3. Address attributes must exist. The full set of potential attributes to be configured is described within the component model configuration section below. Each attribute must be a Description attribute and must be valid on the address object. The attributes must also be standard text attributes, with the exception of the attribute that will be used for the 'Validation Time' field, which must have a validation base type of 'ISO Date and Time.'

   An example of a text-based Description attribute is shown below.

Attribute	References	Attribute Transformation	Validity	Profile

Description

Name	Value
ID	City
Name	City
Last edited by	2016-03-10 15:32:13 by USER2
Full Text Indexable	No
Externally Maintained	No
Hierarchical Filtering	None
Calculated	No
Type	Description
Dimension Dependencies	
Mandatory	No

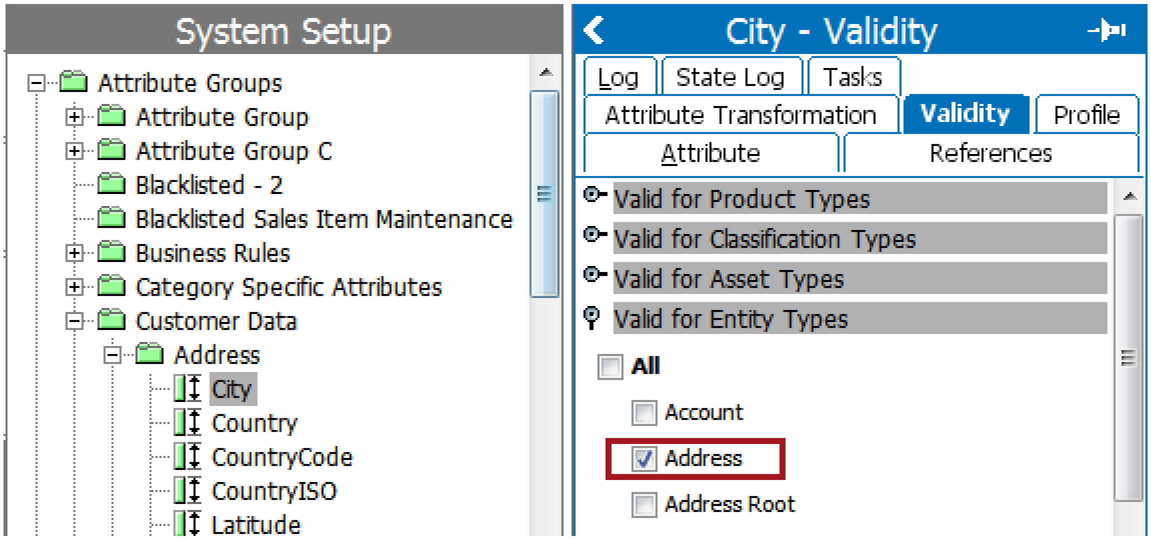
Attribute Validation

Name	Value
Validation Base Type	Text
List Of Values	N/A
Multi Valued	No
Mask	
Minimum Value	N/A
Maximum Value	N/A
Maximum Length	100

Edit Validation Rule

Aspects

Component	Name	Description
Address Component Model	City	The City



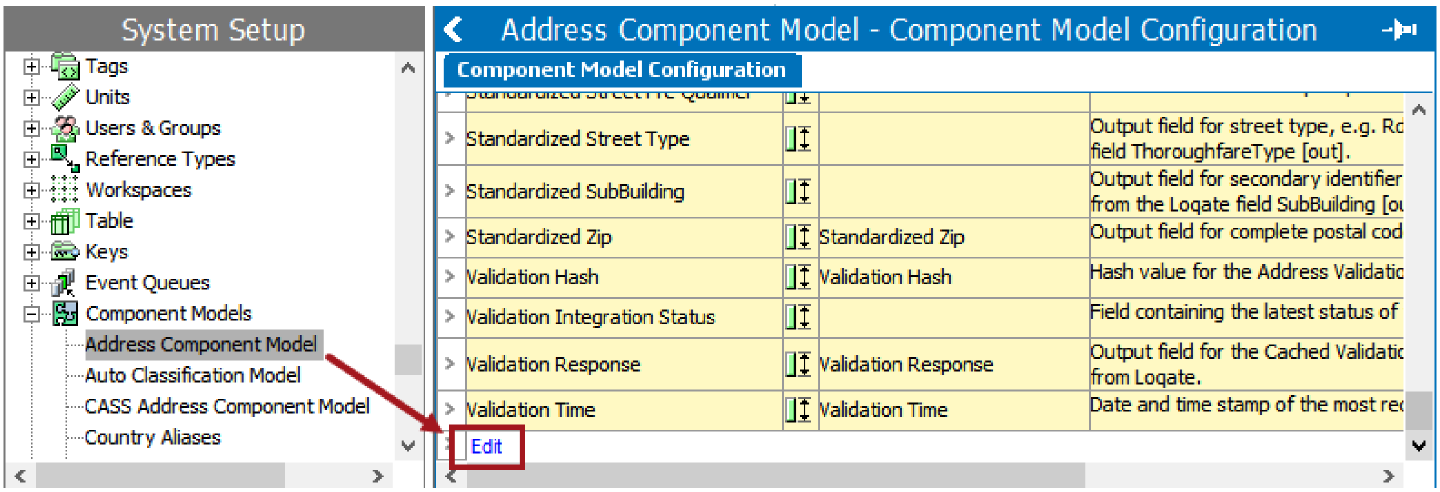
Once the required data model configuration is in place, the component model can be set up.

Manual Attribute Mapping

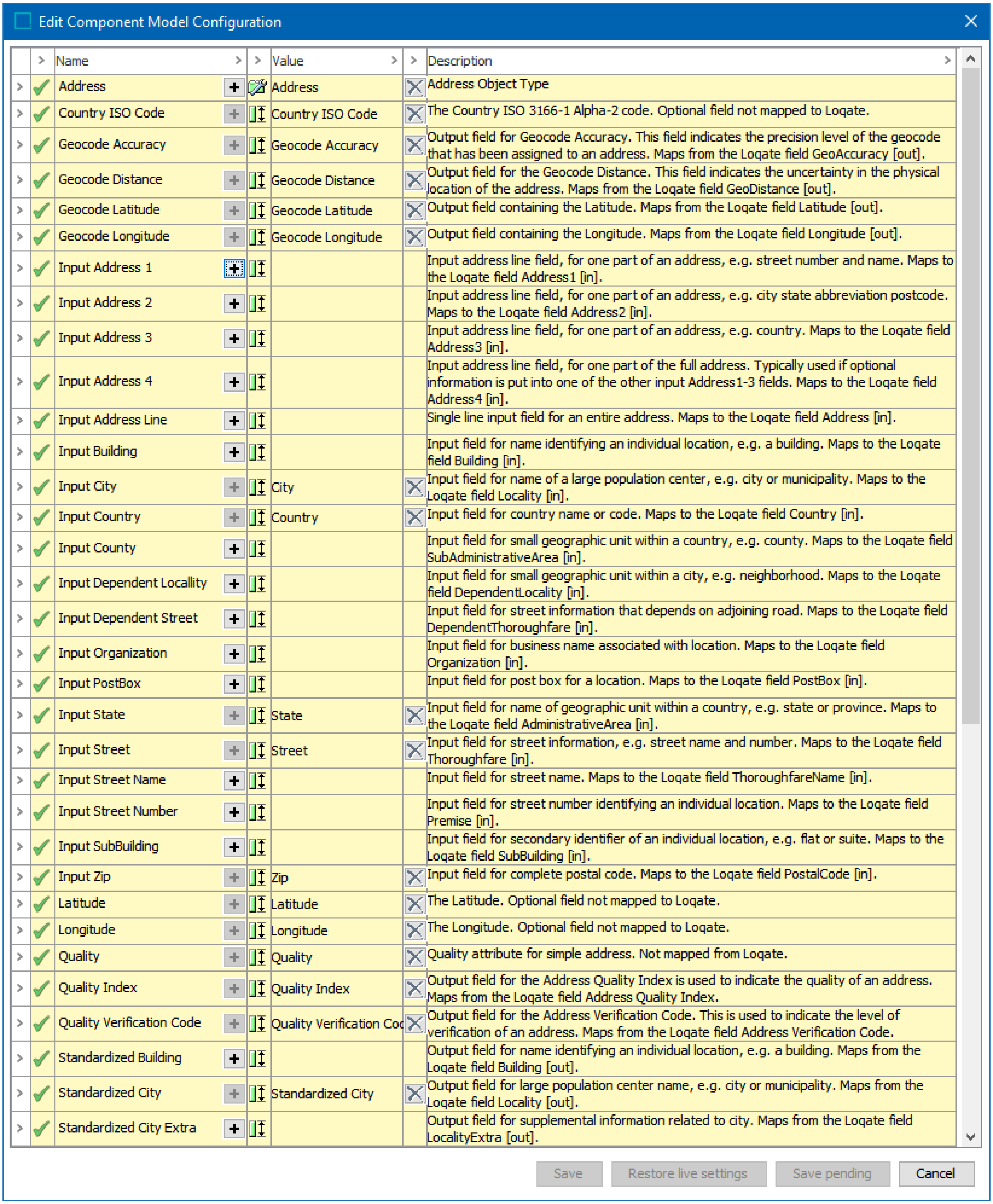
Each field in the component model has the option to be mapped. Most mappings are self-explanatory, but some tips and guidelines for completing the configuration are noted below.

- The **Address** mapping must be configured. The selection can be the address object type referred to in step 1 of the data model considerations and requirements, above, or can be any entity or address data container on which the attributes selected in the component model are valid.
- The **Input Street**, **Input Street Name**, and **Input Street Number** parameters are closely related. Input Street is used to hold a complete street address, including a house or building number, as well as a street name. Input Street Name holds the street name only, while Input Street Number holds the building / house number only. These separated fields are especially useful when dealing with standardization and/or deduplication of addresses. Note that Input Street is required, while Input Street Name and Input Street Number are optional.
- Any attributes for which address data should be stored must be populated. Some mappings are required, while others are optional. For full information on which fields are required and which fields are optional, see the attribute table in the **Address Component Model** topic.

To start the mapping process, click the blue Edit link shown at the bottom of the table.



This will open the editor, allowing you to add, edit, and remove mappings.



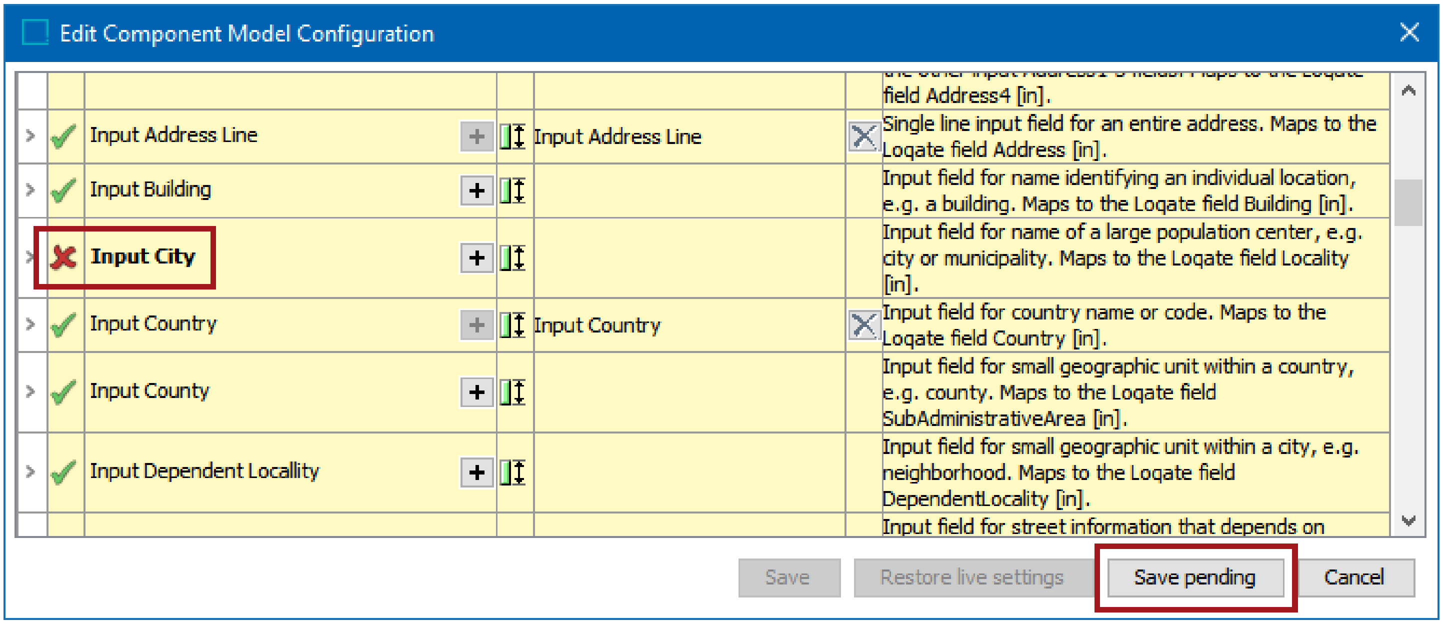
Edit Component Model Configuration

Name	Value	Description
Address	Address	Address Object Type
Country ISO Code	Country ISO Code	The Country ISO 3166-1 Alpha-2 code. Optional field not mapped to Loqate.
Geocode Accuracy	Geocode Accuracy	Output field for Geocode Accuracy. This field indicates the precision level of the geocode that has been assigned to an address. Maps from the Loqate field GeoAccuracy [out].
Geocode Distance	Geocode Distance	Output field for the Geocode Distance. This field indicates the uncertainty in the physical location of the address. Maps from the Loqate field GeoDistance [out].
Geocode Latitude	Geocode Latitude	Output field containing the Latitude. Maps from the Loqate field Latitude [out].
Geocode Longitude	Geocode Longitude	Output field containing the Longitude. Maps from the Loqate field Longitude [out].
Input Address 1		Input address line field, for one part of an address, e.g. street number and name. Maps to the Loqate field Address1 [in].
Input Address 2		Input address line field, for one part of an address, e.g. city state abbreviation postcode. Maps to the Loqate field Address2 [in].
Input Address 3		Input address line field, for one part of an address, e.g. country. Maps to the Loqate field Address3 [in].
Input Address 4		Input address line field, for one part of the full address. Typically used if optional information is put into one of the other input Address1-3 fields. Maps to the Loqate field Address4 [in].
Input Address Line		Single line input field for an entire address. Maps to the Loqate field Address [in].
Input Building		Input field for name identifying an individual location, e.g. a building. Maps to the Loqate field Building [in].
Input City	City	Input field for name of a large population center, e.g. city or municipality. Maps to the Loqate field Locality [in].
Input Country	Country	Input field for country name or code. Maps to the Loqate field Country [in].
Input County		Input field for small geographic unit within a country, e.g. county. Maps to the Loqate field SubAdministrativeArea [in].
Input Dependent Locallity		Input field for small geographic unit within a city, e.g. neighborhood. Maps to the Loqate field DependentLocality [in].
Input Dependent Street		Input field for street information that depends on adjoining road. Maps to the Loqate field DependentThoroughfare [in].
Input Organization		Input field for business name associated with location. Maps to the Loqate field Organization [in].
Input PostBox		Input field for post box for a location. Maps to the Loqate field PostBox [in].
Input State	State	Input field for name of geographic unit within a country, e.g. state or province. Maps to the Loqate field AdministrativeArea [in].
Input Street	Street	Input field for street information, e.g. street name and number. Maps to the Loqate field Thoroughfare [in].
Input Street Name		Input field for street name. Maps to the Loqate field ThoroughfareName [in].
Input Street Number		Input field for street number identifying an individual location. Maps to the Loqate field Premise [in].
Input SubBuilding		Input field for secondary identifier of an individual location, e.g. flat or suite. Maps to the Loqate field SubBuilding [in].
Input Zip	Zip	Input field for complete postal code. Maps to the Loqate field PostalCode [in].
Latitude	Latitude	The Latitude. Optional field not mapped to Loqate.
Longitude	Longitude	The Longitude. Optional field not mapped to Loqate.
Quality	Quality	Quality attribute for simple address. Not mapped from Loqate.
Quality Index	Quality Index	Output field for the Address Quality Index is used to indicate the quality of an address. Maps from the Loqate field Address Quality Index.
Quality Verification Code	Quality Verification Coc	Output field for the Address Verification Code. This is used to indicate the level of verification of an address. Maps from the Loqate field Address Verification Code.
Standardized Building		Output field for name identifying an individual location, e.g. a building. Maps from the Loqate field Building [out].
Standardized City	Standardized City	Output field for large population center name, e.g. city or municipality. Maps from the Loqate field Locality [out].
Standardized City Extra		Output field for supplemental information related to city. Maps from the Loqate field LocalityExtra [out].

Save | Restore live settings | Save pending | Cancel

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.

Rows that are required to be populated will show a red X when not populated. In this case, the Save button is disabled and only the Save pending button is available. Save pending allows you to save the current mappings until all required values can be supplied. The required values must be populated before a true save can be completed and the component model is functional.



CASS Address Component Model

Users of the Loqate Local solution who are located in the United States can obtain an additional license from Stibo Systems to take advantage of the CASS program, which provides an even stricter level of address standardization. CASS (Coding Accuracy Support System) is a certification program run by the United States Postal Service (USPS) that is offered to all mailers, service bureaus, and software vendors that would like the USPS to evaluate the quality and accuracy of their address-matching software. Mailers who use CASS-certified software to check their mailing addresses are able to qualify for discounted postage rates from the USPS. STEP makes it possible to standardize and validate address using the CASS process and to generate and extract CASS reports that can be used to certify that addresses have been CASS validated.

The CASS Address Component Model determines the output fields for the information that is returned from Loqate to STEP after addresses have been validated against CASS data on the Loqate Local server. Loqate is CASS certified and offers CASS verification of address data.

Once the CASS component model has been configured, the system will be ready to standardize addresses against CASS data using the Loqate Local solution.

For additional information on CASS input fields, see the Loqate 'CASS Overview' support website: https://support.loqate.com/support/local-apis/cass-overview-2/

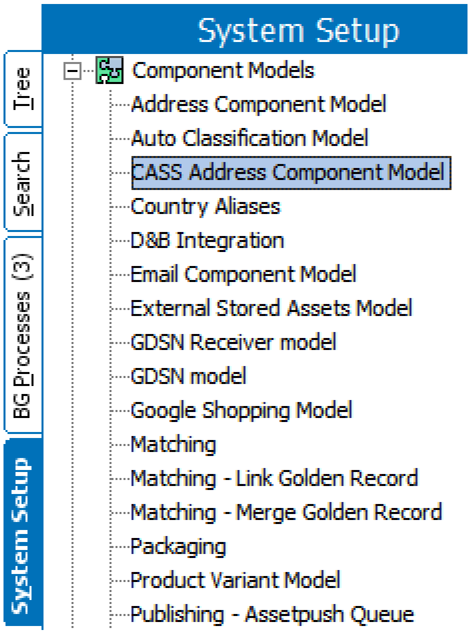
Prerequisites

Before configuring the CASS Address Component Model, the following conditions must be met:

- Users must be based in the United States, as CASS is not valid outside of the US.
- Users must be connected to a Loqate **Local** API server installation and have a CASS license. The CASS solution will not work with a Loqate Cloud API installation.
- Users must have updated the Loqate reference data with the CASS reference data and libraries
- The Address Component Model must already be configured. For more information, see the **Address Component Model** topic.

CASS Address Component Model Overview

The CASS Address Component Model is found under Component Models on the System Setup tab. If the CASS Address Component Model is not present, contact your Stibo Systems account manager.



CASS Address Fields - Input From Address Component Model

The CASS integration requires that users first complete the **Address Component Model**, which contains the basic address attributes such as city and state. The following fields within the Address Component Model are required for CASS address validation. All are mandatory in the Address Component Model except for 'Input Address 1.'

- Input Address 1 – may contain the street number and name, but not the entire address. This is not a mandatory field in the Address Component Model but is required to use CASS.
- Input City – may contain the city only, but can also contain the city, state, and ZIP combined
- Input State
- Input Zip

The following fields can also be used with CASS but are optional:

- Input Dependent Locality – input field for a small geographic unit within a city, e.g., neighborhood
- Input Organization – input field for a business name associated with location

CASS Address Fields - Output In CASS Address Component Model

The following table lists the fields contained in the CASS Address Component Model, along with a description of each field and the corresponding Loqate field. The fields are all output attributes, meaning that they are not populated by users. They are populated with the values returned from Loqate after an address has been validated against the CASS address standardization data files on the Loqate Local server.

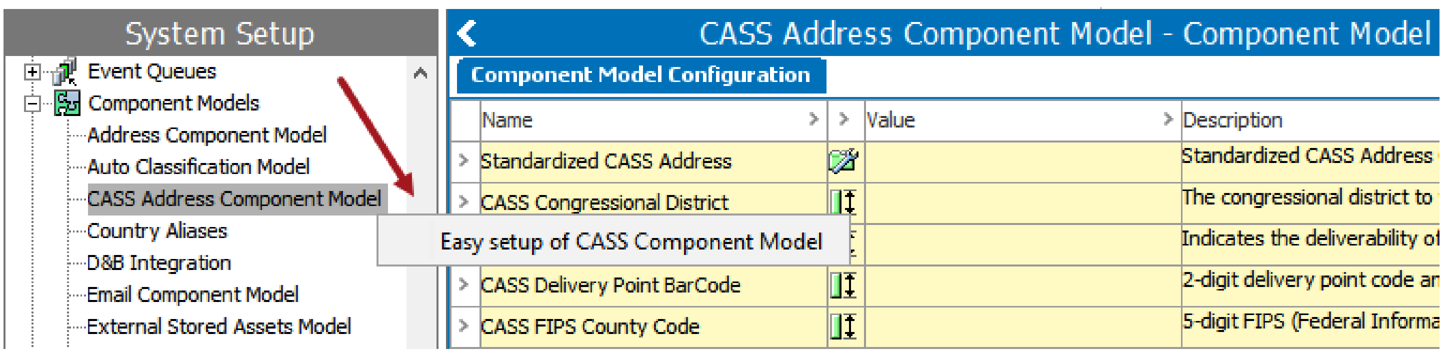
CASS Address Component Model Field	Description	Local Loqate Field Mapping
Address Object Types and/or Address Data Container Types		
Standardized CASS Address	The address types used to represent addresses, can be either entities or data containers	N/A
Standardized Address Attributes (OUTPUT Fields)		
CASS Congressional District	The congressional district to which the address belongs	CongressionalDistrict
CASS DPV Confirmed Indicator	Indicates the deliverability of the address	DPVConfirmedIndicator
CASS Delivery Point BarCode	2-digit delivery point code and 1-digit check digit	DeliveryPointBarCode
CASS FIPS County Code	5-digit FIPS (Federal Information Processing Standard) code	FIPSCountyCode
CASS No Stat Indicator	Indicates the address is not receiving delivery, and the address is not counted as a possible delivery	NoStatIndicator
CASS Residential Delivery	Indicates whether the input address is a residential address or a business address	ResidentialDelivery
CASS Vacant Indicator	Indicates that the delivery point was active in the past, but is currently vacant and is not receiving deliveries.	VacantIndicator
Remaining CASS fields		
CASS Validation Response	Collects remaining CASS output field values. Validation base type is text; maximum length must be at least 1000. CASS Validation Response abc AutoZoneIndicator: D, CarrierRoute: C014, CMRAIndicator: N, DefaultFlag: , DPVFootnotes: AABB, eLOTCode: A, eLOTNumber: 0202, EWSFlag: , FalsePositiveIndicator: , Footnotes: , LACSLinkCode: , LACSLinkIndicator: , LACSStatus: , PMBNumber: , PMBType: , PrimaryAddressLine: 3550 BUSBEE PKWY NW STE 350, RecordType: H, ReturnCode: 31, SecondaryAddressLine: KENNESAW GA 30144-2122, SUITELinkFootnote:	AutoZoneIndicator CarrierRoute CMRAIndicator DefaultFlag DPVFootnotes eLOTCode eLOTNumber EWSFlag FalsePositiveIndicator Footnotes LACSLinkCode LACSLinkIndicator LACSStatus PMBNumber PMBType PrimaryAddressLine RecordType ReturnCode SecondaryAddressLine SUITELinkFootnote

Easy Setup of CASS Address Component Model

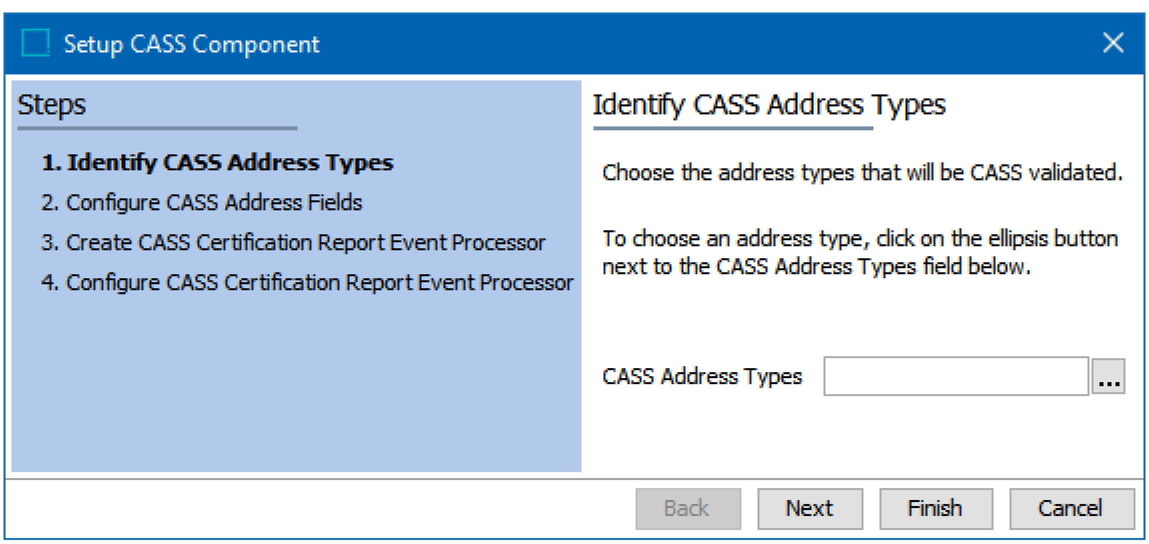
The recommended method for configuring the CASS Address Component Model is to use the 'Easy setup of CASS Component Model' wizard. By using this wizard, STEP can automatically create all of the address attributes required for the CASS solution, as well as a CASS Certification Report event processor used to generate CASS certification reports. By using this wizard, little manual action needs to be taken to complete the configuration, making it a simple and straightforward way to get the solution up and running.

The following steps describe how to configure the CASS Address Component Model using the easy setup method. Also note that detailed help text is present on each step of the wizard.

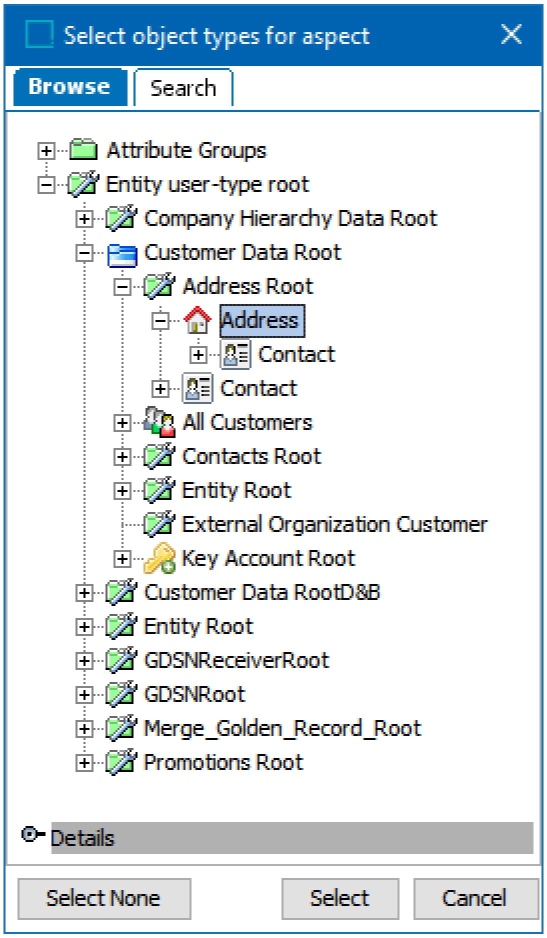
1. Locate the CASS Address Component Model under System Setup > Component Models.
2. Right-click on the CASS Address Component Model and click 'Easy setup of CASS Component Model.'



3. On the **Identify CASS Address Types** screen, click the ellipsis button (...) next to the CASS Address Types field and select the relevant object type(s) from the 'Select object types for aspect' dialog that displays. This should be one or more of the address types used in the Address Component Model.



The below screenshot shows the dialog with an 'Address' entity object type selected. Note that only entities and or data containers can be selected. To multi-select objects, click and hold the Ctrl or Shift key. Click **Select** to choose the object(s) and close the dialog.



Click **Next**.

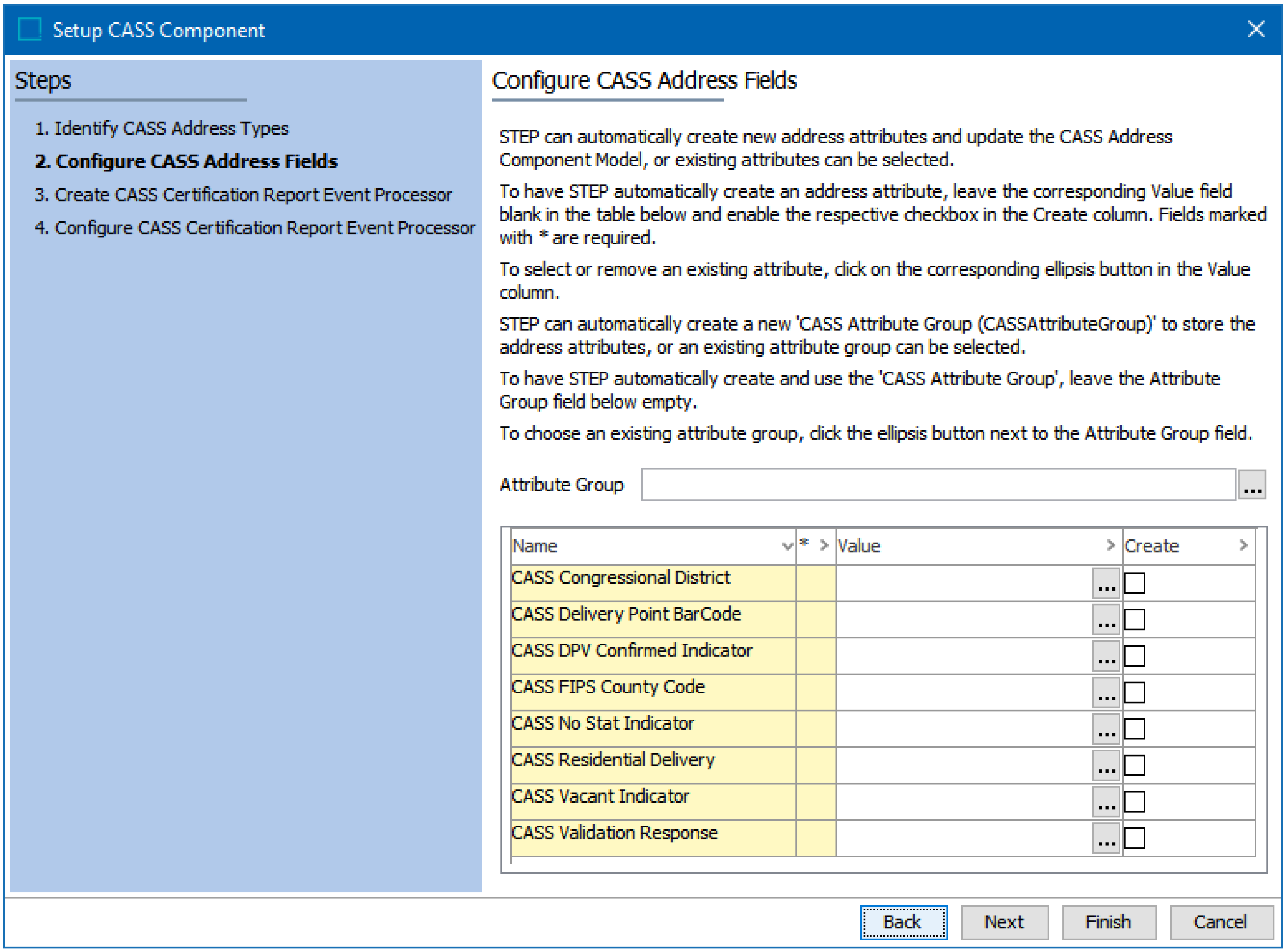
4. On the **Configure Address Fields** screen, attributes are mapped to corresponding CASS output fields. To select or remove an existing attribute, click on the corresponding ellipsis button (**...**) in the Value column.

   To have STEP automatically create an address attribute, leave the corresponding Value field blank and enable the respective checkbox in the Create column. All automatically created attributes will be Description attributes and be made valid for the configured address types. They will have a name equal to the field name and an ID equal to the field name with spaces removed. For example, for the 'CASS Congressional District' field, the automatically created attribute will be named 'CASS Congressional District' and have the ID of 'CASSCongressionalDistrict.'

   It is not required to have STEP create the attributes, but is recommended. If any of the attributes already exist (based on ID), a message will display to inform the user that the attribute already exists.

Note: No 'select all' option is available for the Create column; this omission is intentional, as it forces users to give more thought as to which attributes they actually need instead of creating all attributes at once and having too many.

To select an attribute group where you would like to store the newly created attributes, click the ellipsis button (...) next to the Attribute Group field to launch the 'Select Attribute Group' dialog. Leave the field blank to have STEP automatically create the attribute group, which will be named 'CASS Attribute Group' (ID = CASSAttributeGroup).

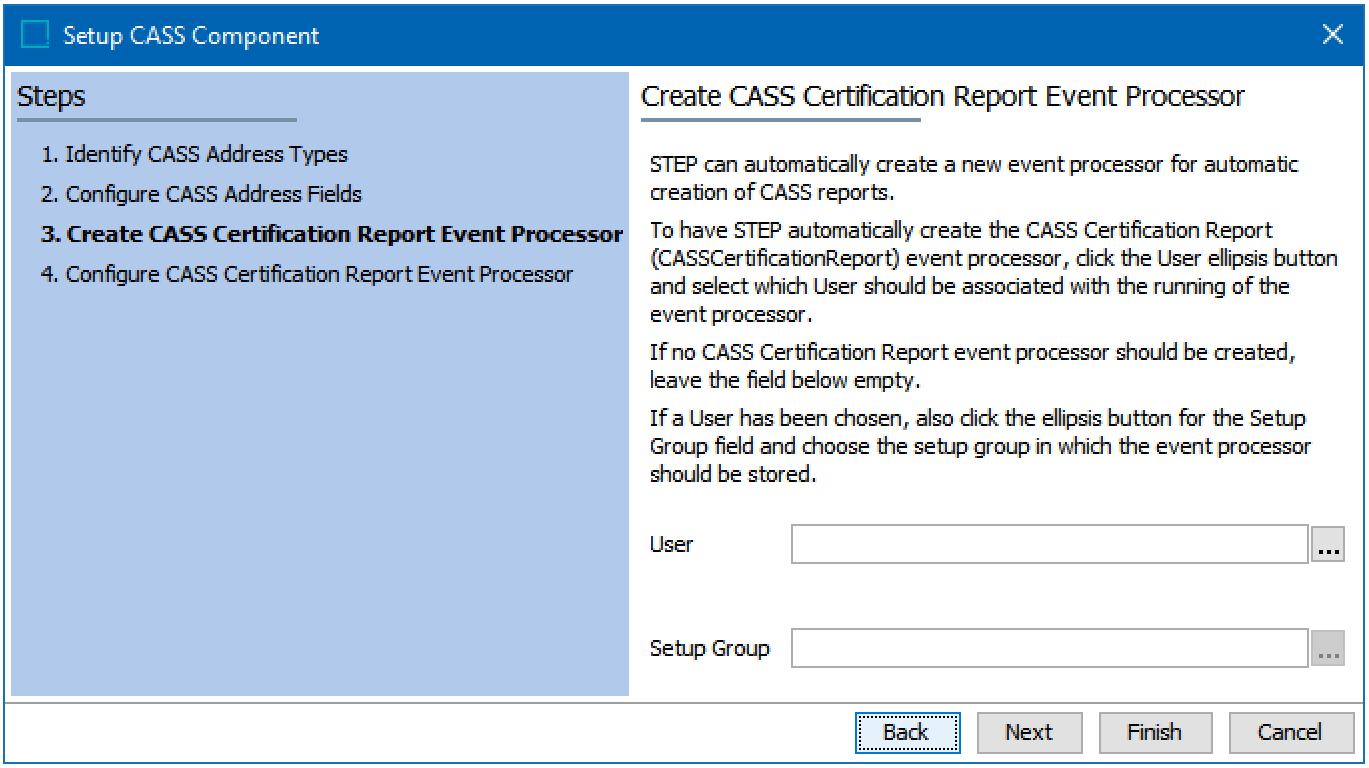


Click **Next** when you have finished mapping and/or creating your attributes.

5. On the **Create CASS Certification Report Event Processor** screen, the system prompts you to choose a User and a Setup Group to have STEP automatically create a 'CASS Certification Report' event processor. This processor will be used to generate CASS certification reports. For more information on the default configuration of this event processor, see the 'Default Configuration - CASS Certification Report Event

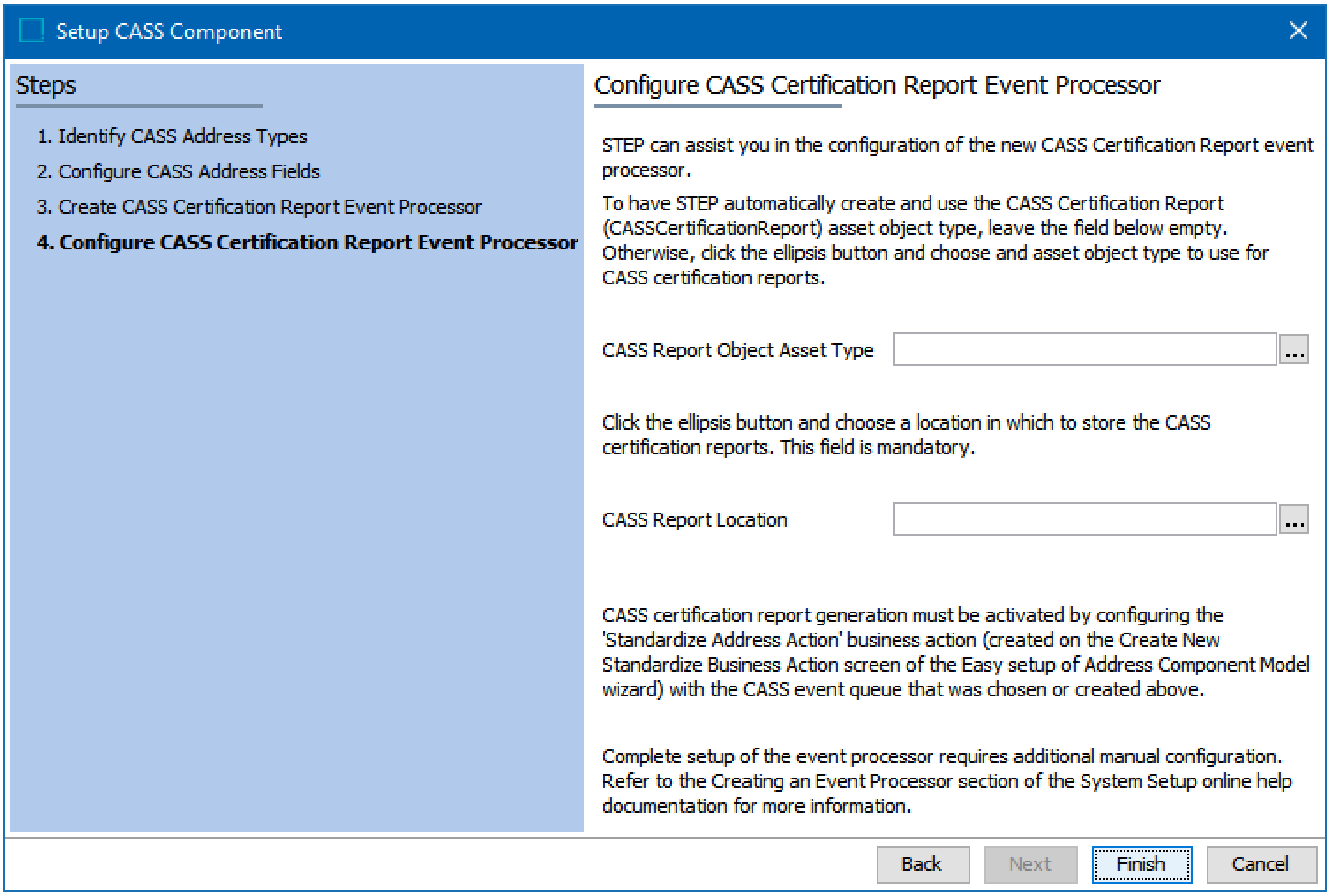
Processor' section of this topic (below). For more information on CASS certification reports, see the CASS Certification Report Generation section of this topic (below).

Click **Finish** if you do not want to create the event processor or would like to create it at a later time. Otherwise, click the ellipsis button (...) next to the User field to select the dedicated system user for the event processor. Then, click the ellipsis button (...) next to the Setup Group field to select the setup group where the event processor should be stored.



6. On the **Configure CASS Certification Report Event Processor** screen, click the ellipsis button (...) next to the 'CASS Report Object Asset Type' field to select the asset object type that will be used for the CASS certification reports. These assets will be stored in STEP as normal .txt files. If you do not already have an asset object type that you would like to use, leave the field blank to have STEP automatically create it.

   Click the ellipsis button (...) next to the 'CASS Report Location' field to choose the classification hierarchy folder where the CASS certification reports should be stored.



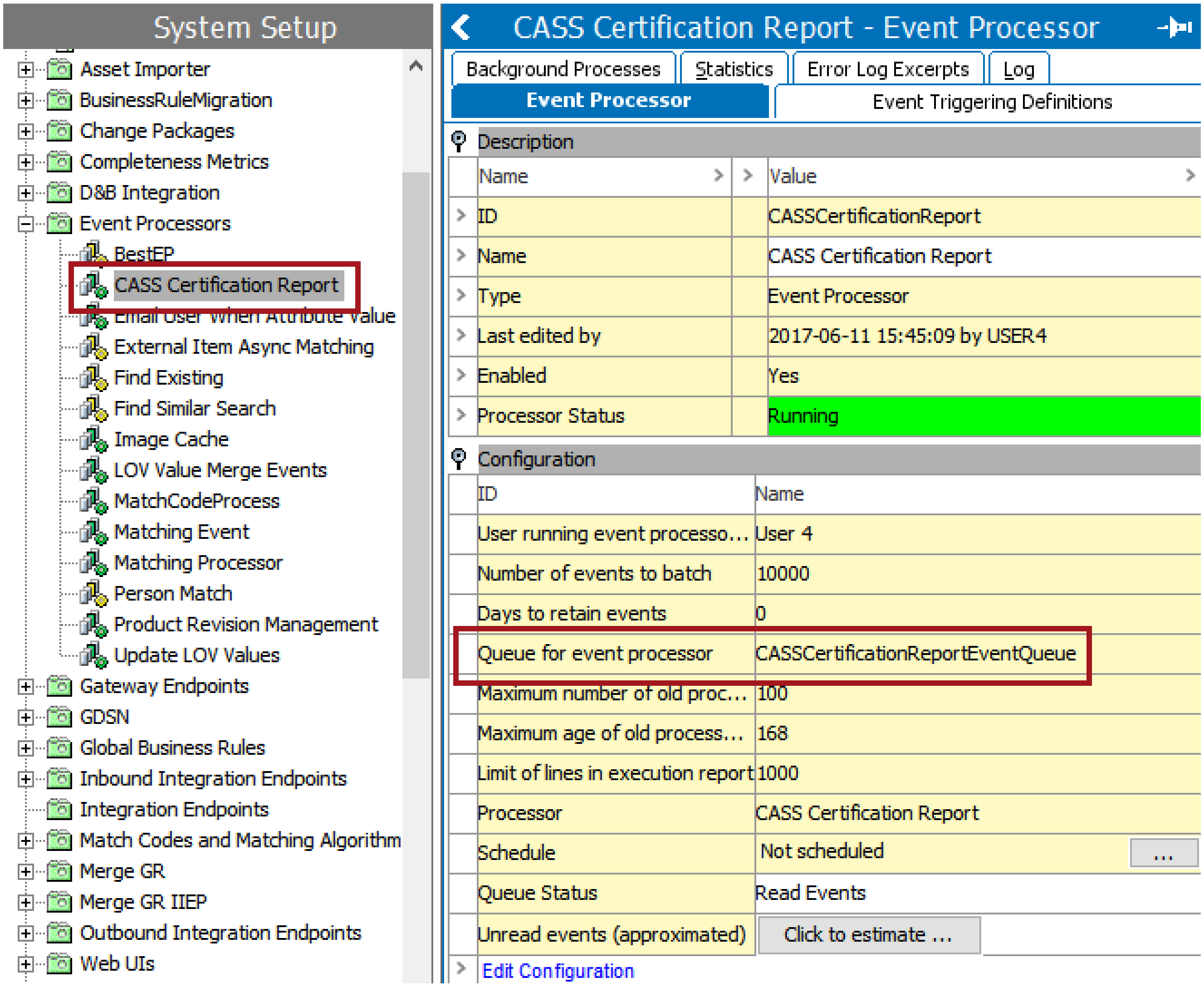
Click **Finish** to complete the easy setup of the component model.

For more information on how to complete the configuration of the event processor after you have completed the easy setup wizard, see the **CASS Certification Report Processing Plugin Parameters and Triggers** topic in the **Event Processors** documentation.

Default Configuration - CASS Certification Report Event Processor

The default configuration of a CASS certification report event processor, as created through the easy setup of the CASS Address Component Model, is pictured below. All default options can be left as-is, or users can adjust specific settings to meet their business needs. The event processor is configured, by default, with a large batch size and a low frequency.

The default queue for the event processor is called CASSCertificationReportEventQueue. If not created through the easy setup of the component model, this queue must first be created, then the name must be manually entered on the 'Configure Event Processor' screen of the Event Processor Wizard.



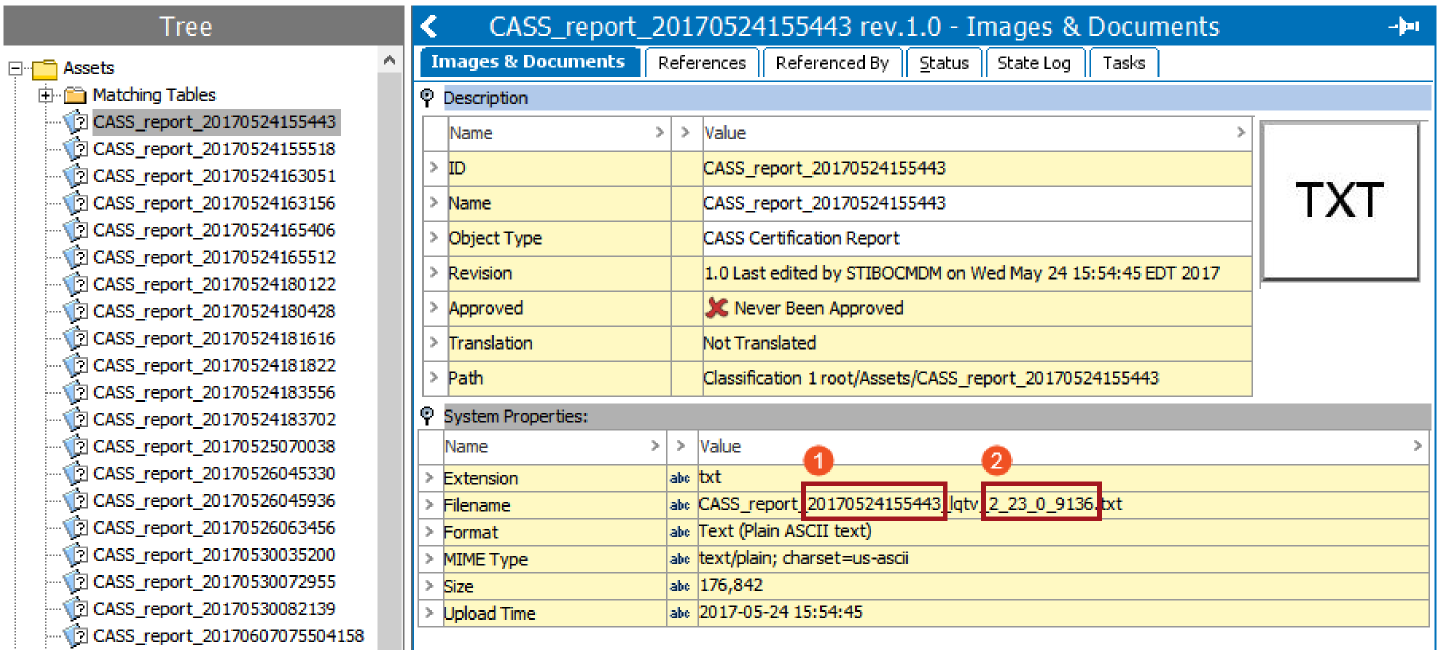
Important: By default, CASS certification event processors are not scheduled. When address standardization operations are initiated by a manual 'Standardize Address' bulk update on address / data container objects or by a 'Standardize Address' business action, these events will sit in the event processor unless the event processor is manually invoked or the processor is scheduled to pick up events at certain intervals. For more information, see the **EPW - Schedule Event Processor** topic within the **Creating an Event Processor** section of the **Event Processors** documentation.

CASS Certification Report Generation

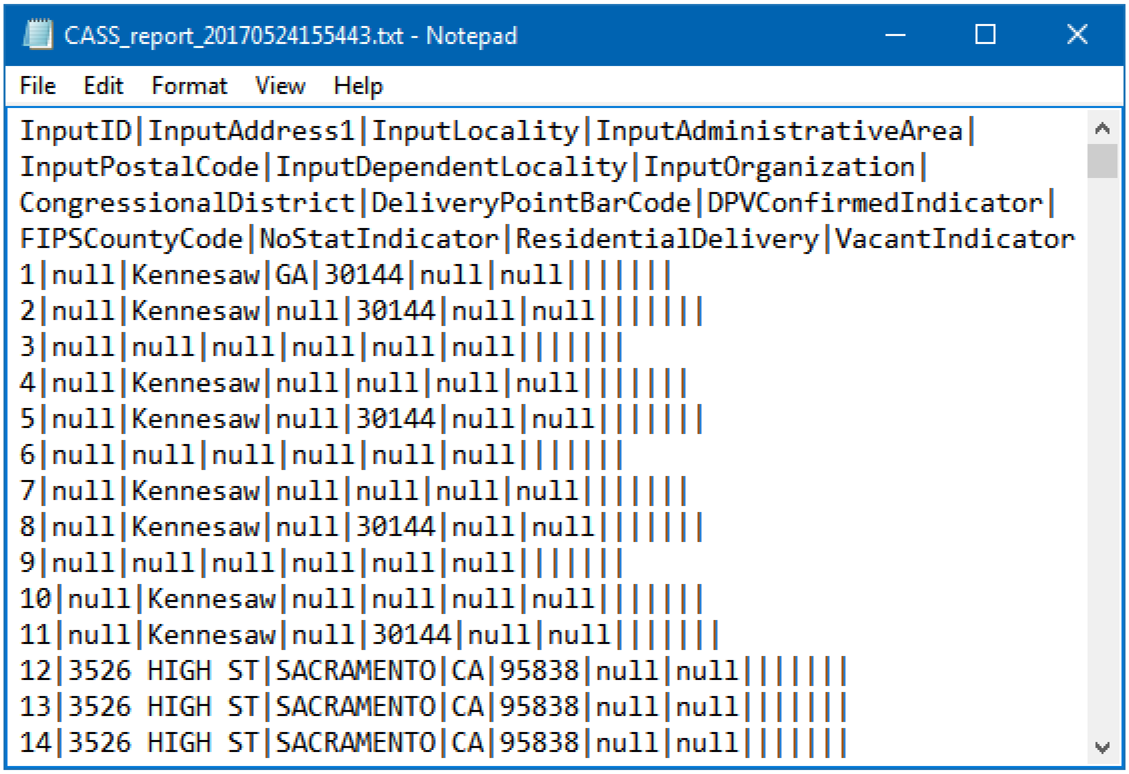
When objects are sent to the CASS certification report event processor, the Loqate Local API uses a Loqate program, lqtBatch, to generate the CASS reports. This program is delivered with the Loqate Local API. When the event processor is invoked, the CASS Certification Event processor plugin:

1. Extracts all US addresses from all originating objects of all the events in the batch
2. Generates an input text file with all these addresses
3. Sends a command to IqtBatch, using this text file as the input
4. Places the output file 'CASS_report_[time stamp]_lqtv_[loqate version].txt' in the specified CASS report location in the classification hierarchy using the specified CASS report asset object type.

The below screenshot shows an example of CASS reports stored within the Assets folder in the STEP classification hierarchy. In the below pictured Filename, the first number string (1) is a report generation time stamp the second number string (2) is the Loqate version number.



The below is a sample CASS certification report.




```
InputID|InputAddress1|InputLocality|InputAdministrativeArea|
InputPostalCode|InputDependentLocality|InputOrganization|
CongressionalDistrict|DeliveryPointBarCode|DPVConfirmedIndicator|
FIPSCountyCode|NoStatIndicator|ResidentialDelivery|VacantIndicator
1|null|Kennesaw|GA|30144|null|null|||||||
2|null|Kennesaw|null|30144|null|null|||||||
3|null|null|null|null|null|null|||||||
4|null|Kennesaw|null|null|null|null|||||||
5|null|Kennesaw|null|30144|null|null|||||||
6|null|null|null|null|null|null|||||||
7|null|Kennesaw|null|null|null|null|||||||
8|null|Kennesaw|null|30144|null|null|||||||
9|null|null|null|null|null|null|||||||
10|null|Kennesaw|null|null|null|null|||||||
11|null|Kennesaw|null|30144|null|null|||||||
12|3526 HIGH ST|SACRAMENTO|CA|95838|null|null|||||||
13|3526 HIGH ST|SACRAMENTO|CA|95838|null|null|||||||
14|3526 HIGH ST|SACRAMENTO|CA|95838|null|null|||||||
```

Loqate Local Integration

Loqate is a third-party address standardization service that, when integrated with STEP, is used to return standardized addresses that adhere to the standards of local postal authorities such as the USPS. Standardized addresses are essential to securing safe postal deliveries and also help save money when doing mass mailings. A fully standardized address is also a first step in customer deduplication (finding and removing duplicate records containing these addresses) using Matching, Linking, and Merging.

Installation information can be found in online help in the **Loqate Integration** section of the **Data Integration** documentation.

Information can be found in the following topics:

- Loqate Local Software Install
- Loqate Local Performance Considerations
- Loqate Local Upgrade and Migration
- Loqate Local Troubleshooting
- Loqate Local Command Line Tool

Loqate Local Software Install

This topic includes the following information:

- Installation Considerations
- Accessing the Loqate Software
- Installing Basic Data Packs

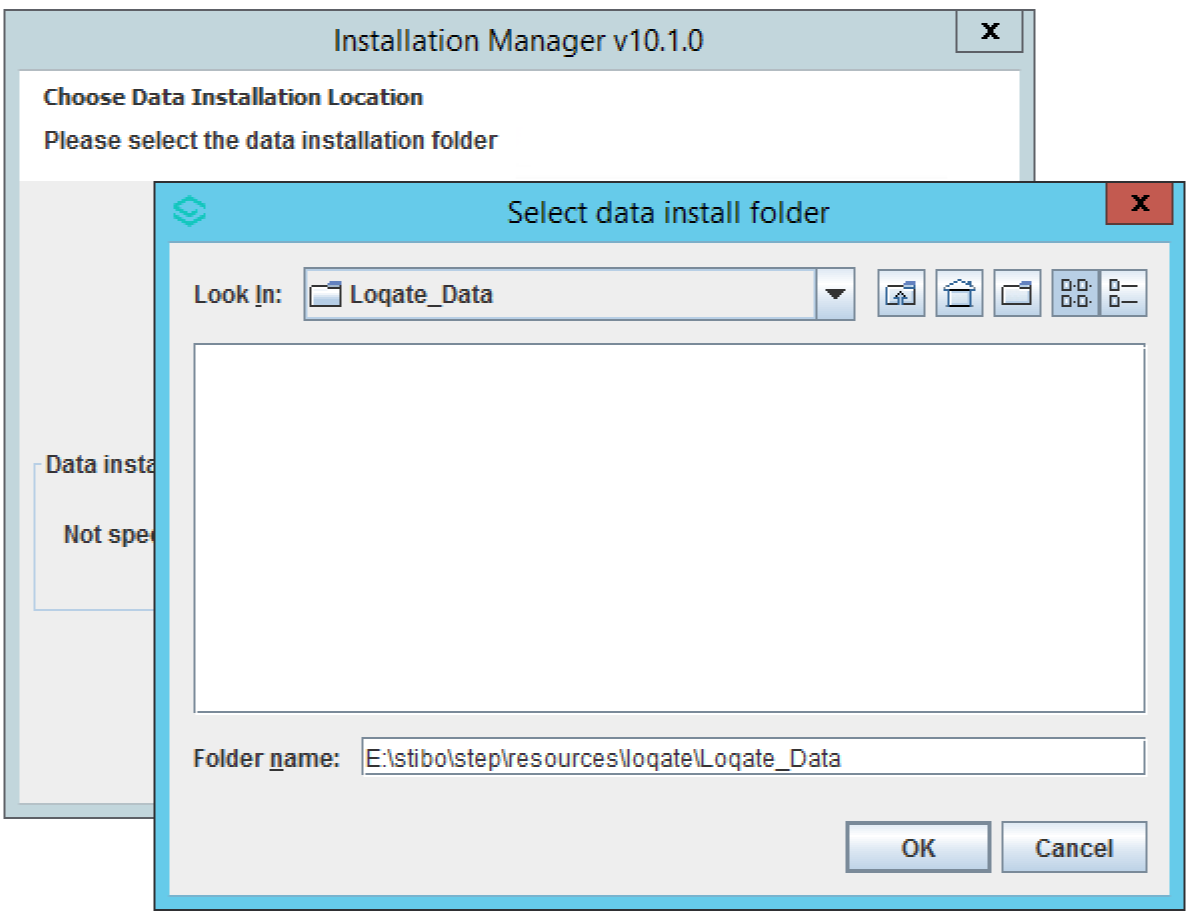
Prerequisites

Before installing the Loqate Local Software, see the **Loqate APIs** topic.

Installation Considerations

For details on the wizard steps, see the installation instructions PDF: **2019Q2.0_InstallationInstructions.pdf**. Additional considerations to keep in mind during the installation process include:

- The folder chosen for data installation on the 'Choose Data Installation Location' wizard screen must be the same as that listed in the sharedconfig.properties file as shown below for Linux and Windows.



Linux

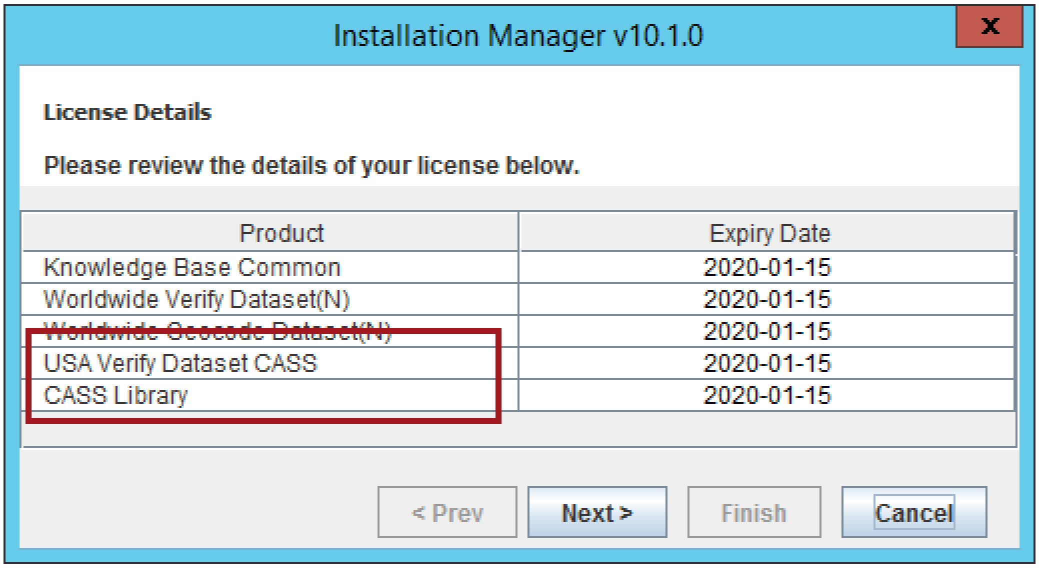
```
#============================================================#
#LOQATE ADDRESS STANDARDIZATION
#=============================================#
Address.AddressQualityPlugin=LoqateAddressQuality
Address.Service.Loqate.GeocodeAll=true
# Address.Service.Loqate.License=

Loqate.Server.Data=/opt/stibo/step/resources/loqate/Loqate_Data
#Loqate.Server.PoolEnabled=True
#Loqate.Server.MaxPoolSize=100
#Loqate.Server.PoolTimeoutMs=5000
```

Windows

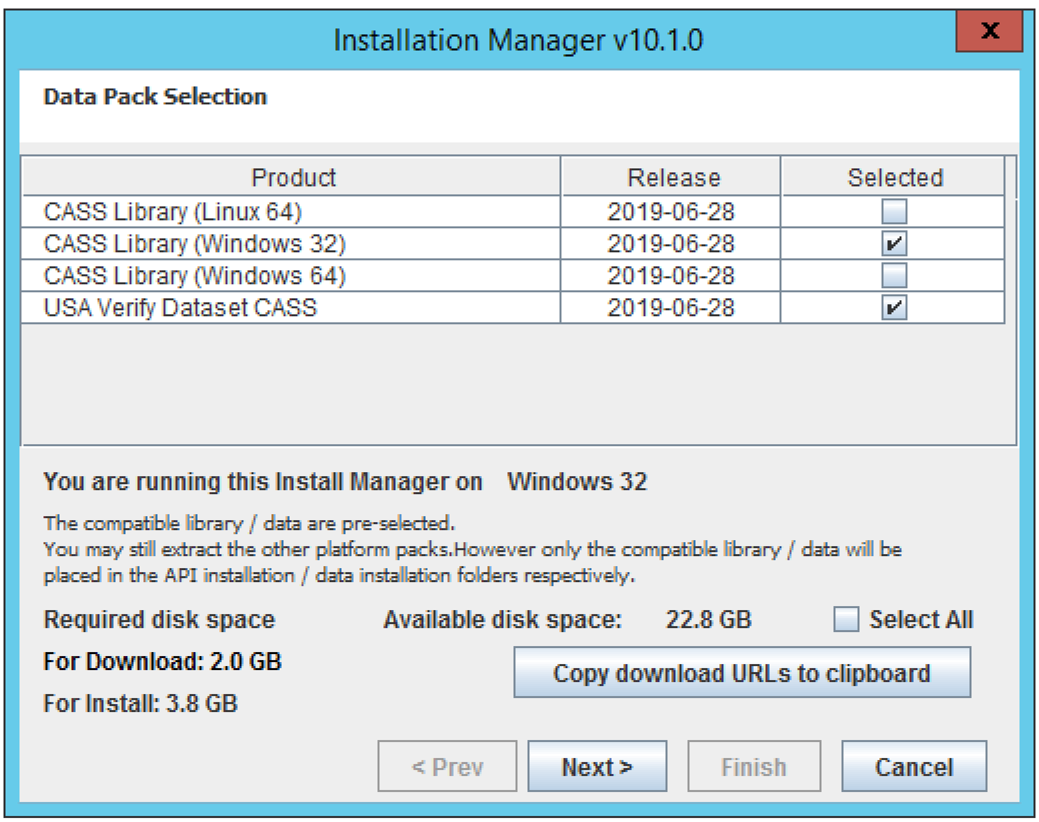
```
#==============================================================================#
# Loqate settings
#==============================================================================#
Address.AddressQualityPlugin=LoqateAddressQuality
Address.Service.Loqate.License=
Loqate.Server.Data=E:/stibo/step/resources/loqate/Loqate_Data

#Log.Level.com.stibo.loqate=FINEST
#Log.Level.com.stibo.loqate.localapi=FINEST
#Log.Level.com.stibo.addressaspect=FINEST
#Log.Level.com.stibo.addressaspect.workbench=FINEST
#Log.Level.com.stibo.addressaspect.workbench.services=FINEST
#Log.Level.com.stibo.addressaspect.workbench.easysetup=FINEST
```

Important: The ‘Data’ directory serves a different function than the ‘Download’ directory that is chosen in a later step in the wizard. The ‘Download’ directory is only used for temporary storage of the downloaded and compressed reference packs. After the installation has been completed on all STEP servers, the ‘Download’ directory may be cleared.

- During the install, if you have a CASS license you will also see the CASS Library appear on the License Details screen. CASS is only for users within the United States.



- For CASS users, the Install Manager will show an additional step towards the end of the installation. Make sure that you select the library that is suitable for your platform (Windows or Linux) and the CASS Dataset.



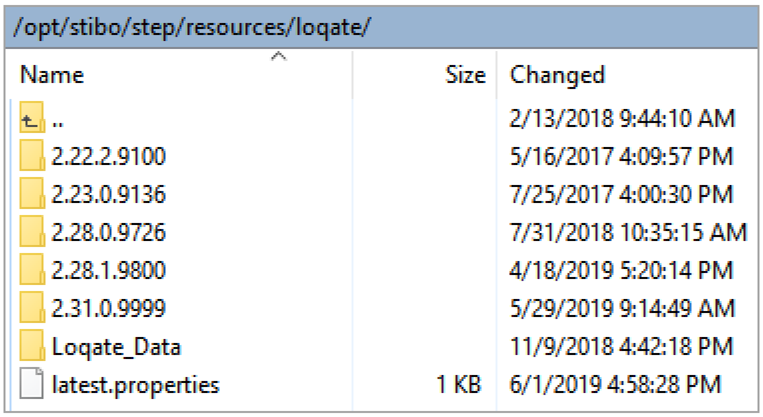
Accessing the Loqate Software

The software needed to install the **Loqate Local API** server is distributed by Stibo Systems along with your STEP installation. The server software should already be installed; if not, contact Stibo Systems. The software used to install the Loqate reference data packs is called **Install Manager** and its location is provided in the following section.

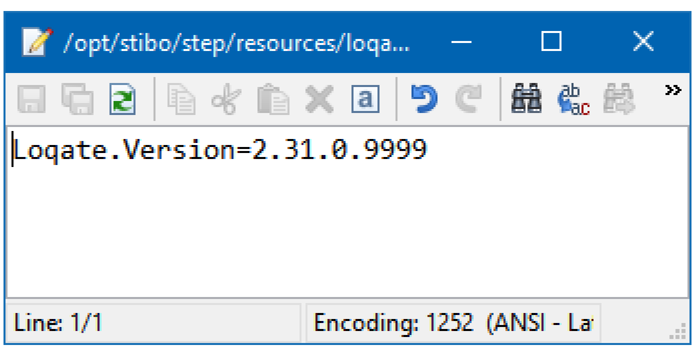
Install Manager

On Linux systems, the **Install Manager** software that is needed to install the Loqate data packs is located inside the folder containing the latest version of the software.

```
/opt/stibo/step/resources/loqate/2.31.0.9999
```



The **latest.properties** file contains the latest version of the software, and this will be the version that Stibo Systems Enterprise Platform (STEP) will be using. Contact Stibo Systems if there is not a folder with the corresponding version number loaded on your system.



On Windows machines, the software will be located in this directory: E:\stibo\step\resources\loqate

This PC ▸ home (E:) ▸ stibo ▸ step ▸ resources ▸ loqate

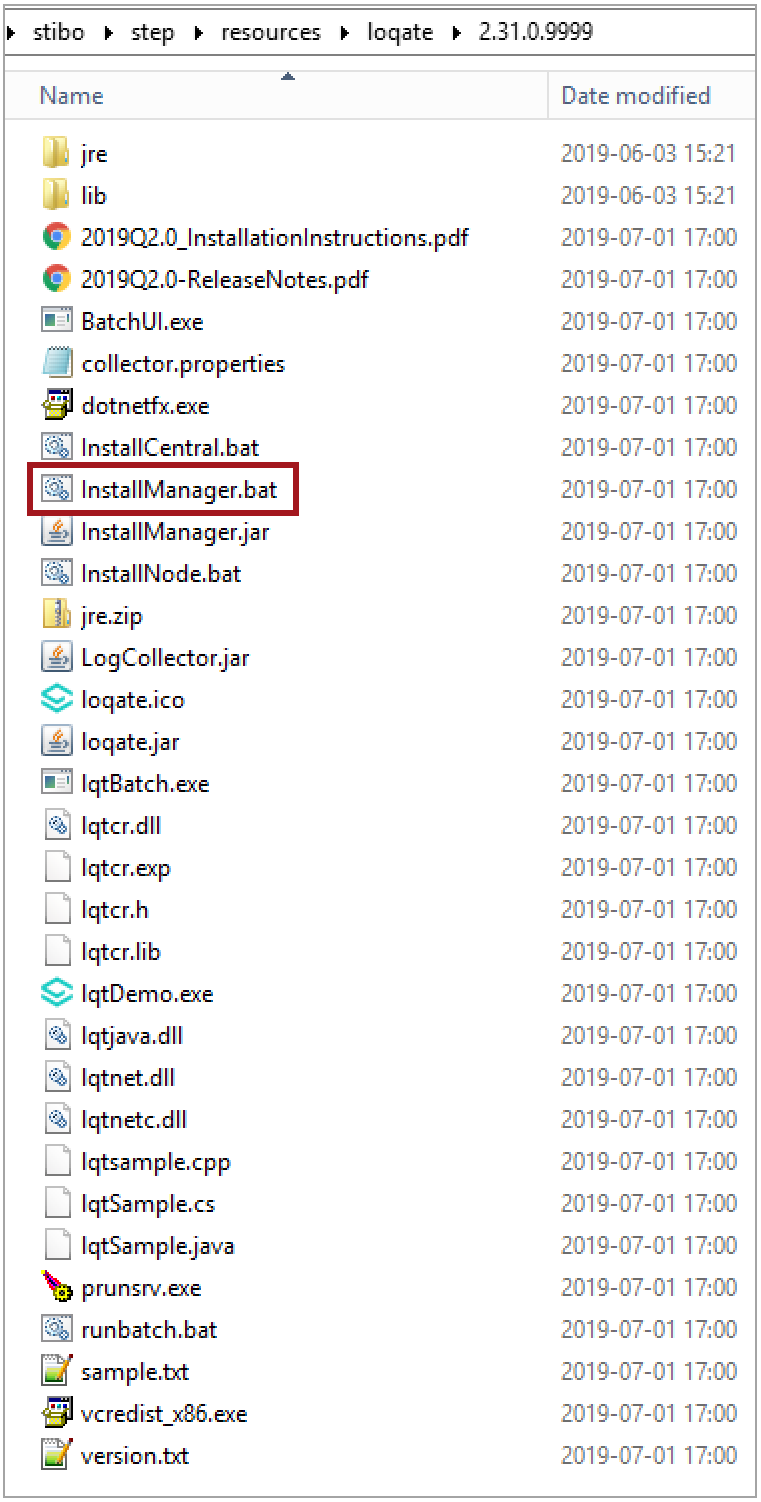
Name	Date modified	Type
2.23.0.9136	2018-10-02 16:12	File folder
2.28.1.9800	2018-11-20 14:50	File folder
2.31.0.9999	2019-06-03 15:21	File folder
DATA	2018-10-01 11:58	File folder
latest.properties	2019-07-01 17:00	PROPERTIES File

Contained within the latest version folder (in this example, 2.31.0.9999) are several files. The key file for use is the InstallManager.jar, which will be covered in another section of this guide.

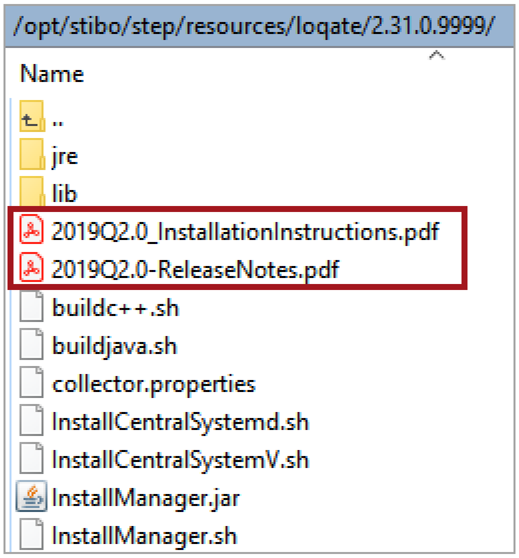
Linux

```
[stibosw@doc-dev loqate]$ ls
2.22.2.9100  2.28.0.9726  2.31.0.9999        Loqate_Data
2.23.0.9136  2.28.1.9800  latest.properties
[stibosw@doc-dev loqate]$ cd 2.31.0.9999/
[stibosw@doc-dev 2.31.0.9999]$ ls
2019Q2.0_InstallationInstructions.pdf  LogCollector.jar
2019Q2.0-ReleaseNotes.pdf              loqate.jar
buildc++.sh                            lqtbatch
buildjava.sh                           LQTCentralCollector.service
collector.properties                   lqtcr.h
InstallCentralSystemd.sh               LQTNodeCollector.service
InstallCentralSystemV.sh               lqtquerysample.cpp
InstallManager.jar                     lqtsample.cpp
InstallManager.sh                      lqtSample.java
InstallNodeSystemd.sh                  runbatch.sh
InstallNodeSystemV.sh                  runc++.sh
jre                                    runjava.sh
lib                                    sample.txt
liblqtcr.so                            version.txt
liblqtjava.so
```

Windows



In addition, the Release Notes file provides information on the latest updates to the Loqate software. The Installation Instructions file provides instructions on how to install the Loqate software on both Linux / UNIX and Windows platforms.



Installing Basic Data Packs

Typically, data packs for Loqate Local are installed on the same server as the STEP application server, though they can be stored in another server via a shared drive.

Loqate software installation involves installation of the API and the GKR (Global Knowledge Repository) data packs. The Installation Instructions PDF file walks through the steps of the installation process for the Local API, followed by instructions for the installation of the GKR using the Install Manager. In Windows, the Install Manager runs in a Graphical User Interface mode. For Linux / UNIX, a command line interface is used. The Local API is installed first, followed by the data packs, which are used using the install manager.

Note: STEP is delivered with the Local API pre-installed, so users may ignore the Local API instructions in the installation instructions PDF and go directly to the GKR section.

Installing Loqate Reference Data Packs

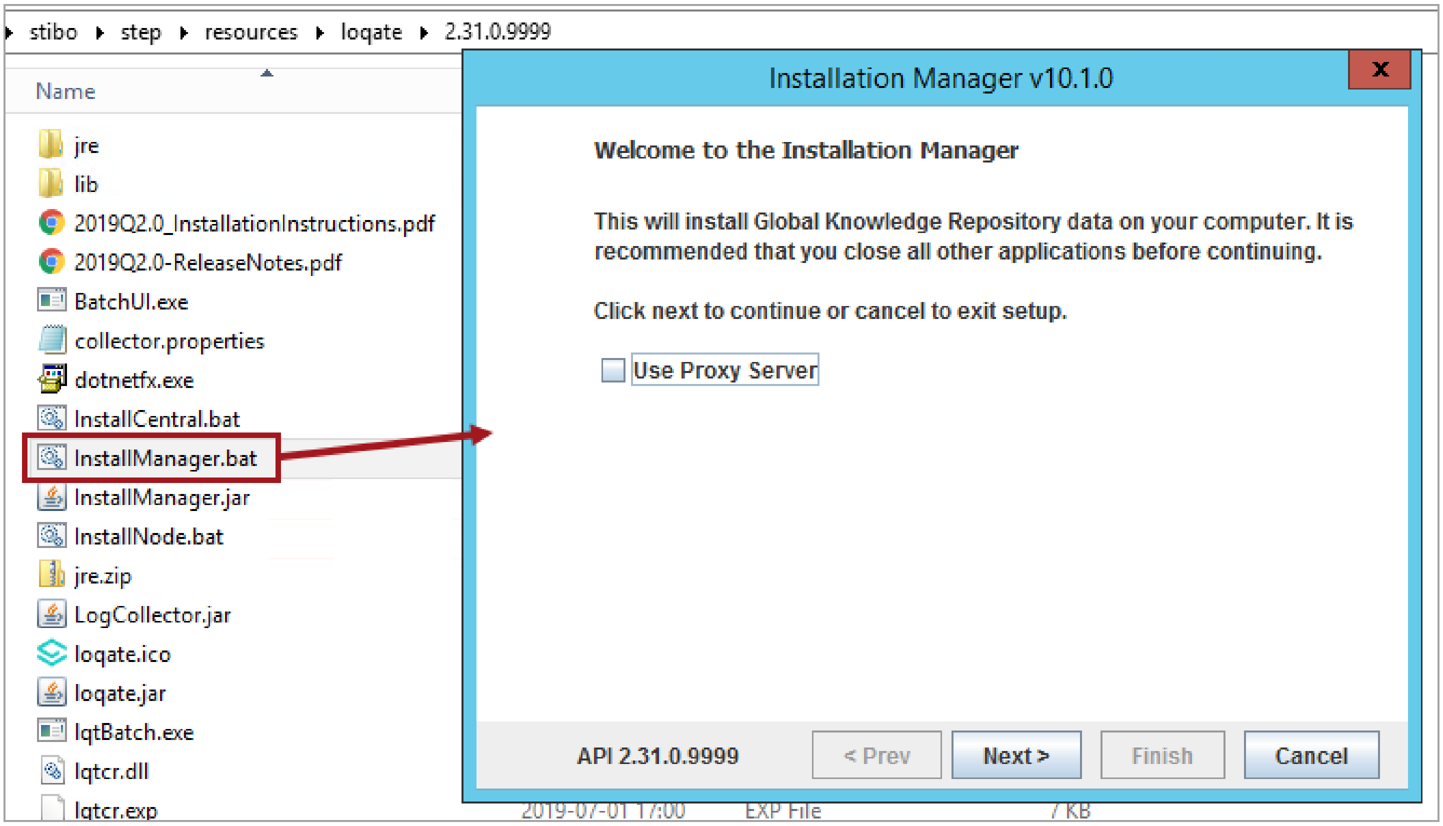
Loqate reference data packs are installed using the Install Manager, which is software distributed as a part of the local-loqate component that was installed using the recipe from the Install Recipes section of this document (section 4.4).

The Install Manager uses a GUI on Windows and is run as a cmd line tool on Linux. The InstallManager can handle offline installation of data packs and the cmd line version can take advantage of a so-called Silent file, which can be used to streamline and automate the installation process.

Full details on the different install options are available in the Installation Instructions PDF file as well as on the Loqate website https://support.loqate.com/getting-started/installers/data-installation-and-update-process/.

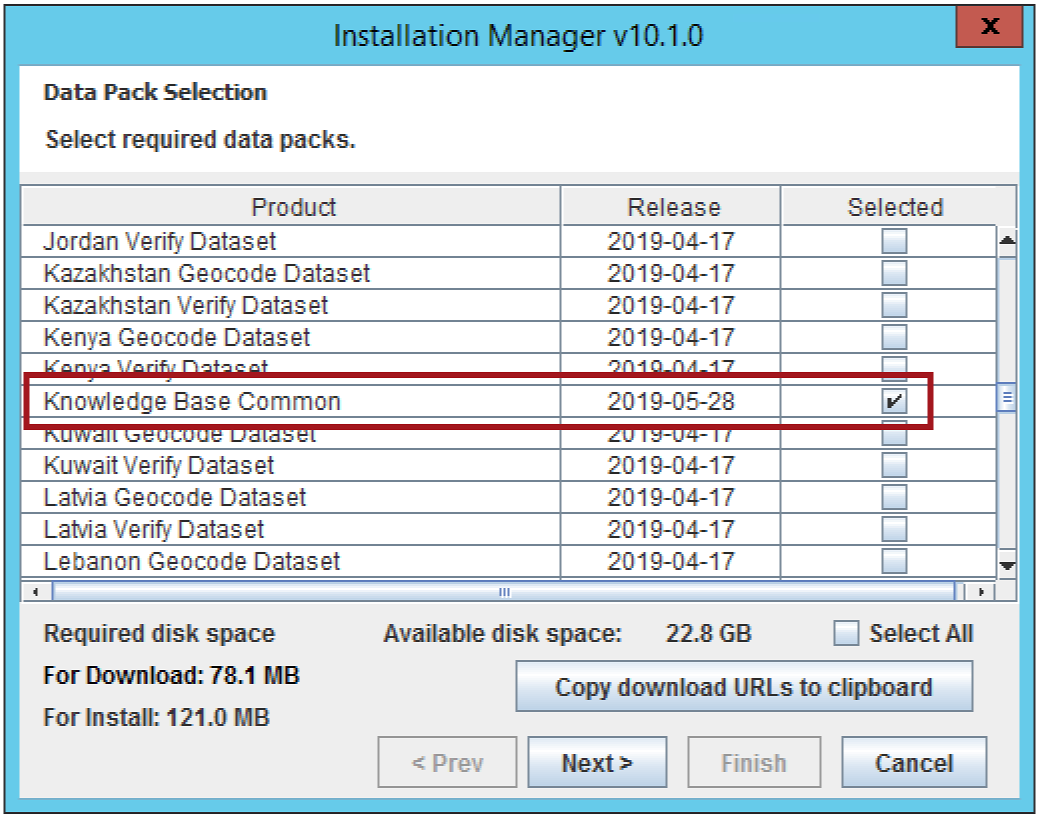
Windows

On Windows machines, double-click the InstallManager.bat file to launch the Installation Manager application.



Full instructions for the installation are provided in the Installation Instructions PDF or on the Loqate website https://support.loqate.com/support/local-apis/installers/data-installation-and-update-process/.

Important: On the Data Pack Selection screen of the wizard, Knowledge Base Common is mandatory and must be selected for installation. Its location is not mentioned in Loqate's installation instructions and can easily be missed, since it appears between the entries for Kenya and Kuwait.



Once finished, restart your STEP server.

Linux

On Linux machines, the InstallManager.sh file will be launched through a command line interface.

```
[stibosw@doc-dev 2.31.0.9999]$ ls
2019Q2.0_InstallationInstructions.pdf  LogCollector.jar
2019Q2.0-ReleaseNotes.pdf              loqate.jar
buildc++.sh                            lqtbatch
buildjava.sh                           LQTCentralCollector.service
collector.properties                   lqtcr.h
InstallCentralSystemd.sh               LQTNodeCollector.service
InstallCentralSystemV.sh               lqtquerysample.cpp
InstallManager.jar                     lqtsample.cpp
InstallManager.sh                      lqtSample.java
InstallNodeSystemd.sh                  runbatch.sh
InstallNodeSystemV.sh                  runc++.sh
jre                                    runjava.sh
lib                                    sample.txt
liblqtcr.so                            version.txt
liblqtjava.so
```

Full instructions for the installation are provided in the Installation Instructions PDF or on the Loqate website https://support.loqate.com/support/local-apis/installers/data-installation-and-update-process/.

Important: Just as in the Windows installation, installing the Knowledge Base Common data package is required. Its location is not mentioned in Loqate's installation instructions and can easily be missed, since it appears between the entries for Kenya and Kuwait.

```
stibosw@doc-dev:~/step/resources/loqate/2.31.0.9999
124   Iraq Verify Dataset             2019-04-17
125   Ireland Geocode Dataset         2019-04-17
126   Ireland Verify Dataset          2019-04-17
127   Israel Geocode Dataset          2019-04-17
128   Israel Verify Dataset           2019-04-17
129   Italy Geocode Dataset           2019-04-17
130   Italy Verify Dataset            2019-04-17
131   Jamaica Geocode Dataset         2019-04-17
132   Jamaica Verify Dataset          2019-04-17
133   Japan Geocode Dataset Premium   2019-04-17
134   Japan Verify Dataset Premium    2019-04-17
135   Jordan Geocode Dataset          2019-04-17
136   Jordan Verify Dataset           2019-04-17
137   Kazakhstan Geocode Dataset      2019-04-17
138   Kazakhstan Verify Dataset       2019-04-17
139   Kenya Geocode Dataset           2019-04-17
140   Kenya Verify Dataset            2019-04-17
141   Knowledge Base Common           2019-05-28
142   Kuwait Geocode Dataset          2019-04-17
143   Kuwait Verify Dataset           2019-04-17
144   Latvia Geocode Dataset          2019-04-17
145   Latvia Verify Dataset           2019-04-17
146   Lebanon Geocode Dataset         2019-04-17
147   Lebanon Verify Dataset          2019-04-17
148   Lesotho Geocode Dataset         2019-04-17
149   Lesotho Verify Dataset          2019-04-17
150   Libya Geocode Dataset           2019-04-17
151   Libya Verify Dataset            2019-04-17
152   Lithuania Geocode Dataset       2019-04-17
153   Lithuania Verify Dataset        2019-04-17
```

Once finished, restart your STEP server.

Offline Mode

This method involves running the Install Manager to download the data packs but, instead of installing them onto the sever, they are downloaded to a local hard drive and coped to the server later.

Silent Installer

Silent-mode installation allows you to define an installation configuration only once and then use the configuration to duplicate the installation on many machines. During installation in silent mode, the installation program reads the settings for your configuration from a file that you create before beginning the installation. The Install Manager program does not display any configuration options during the installation process. Silent-mode installation works on both Windows and Linux systems.

Loqate Local Performance Considerations

This topic covers methods to ensure that the Loqate Local installation is performing optimally. For information about Loqate Local's APIs, see the **Loqate APIs** topic. For information on installing Loqate Local, see the **Loqate Local Software Install** topic.

To improve performance, STEP keeps a pool of connections to Loqate. The connection pool is automatically activated using the default values below.

Connections are configured in the sharedconfig.properties file using the following case -sensitive **Loqate. Server** parameters. These properties should only be introduced and changed if a need arises, e.g., in cases of bad Loqate performance. In other words, while the configuration properties are pictured in this document, it is not required to add them to the sharedconfig.properties file on your system.

```
#==================================================
#LOQATE ADDRESS STANDARDIZATION
#============================================#
Address.AddressQualityPlugin=LoqateAddressQuality
Address.Service.Loqate.GeocodeAll=true
# Address.Service.Loqate.License=x6I5b7ubMzEmMBjega

Loqate.Server.Data=/opt/stibo/step/resources/loqate
Loqate.Server.PoolEnabled=True
Loqate.Server.MaxPoolSize=100
Loqate.Server.PoolTimeoutMs=5000
```

- **Loqate.Server.PoolEnabled** – Default value is True, enables a pool of connections between STEP and Loqate.
- **Loqate.Server.MaxPoolSize** – Default value is 100, sets the maximum number of Loqate objects in a pool (the number of Java objects that are able to process Loqate requests).

  Set this value the same as (or a bit more than) the expected number of concurrent threads that will be executed. To run a maximum of 10 threads in parallel, the size should be set to 10.

  A number set lower than the number of executors means requests may have to wait for the Loqate pool and the process will be slower than it would be if the number were higher.

  Objects are created on demand, so while a pool size of 100 may be too high for some systems, setting the number too high does not mean that too many objects will be created. Even if the number is set to 100 but there are only 10 threads, no more than 10 objects should ever be created. The only risk of setting the number too high is if Loqate turns out to be unstable and, at some point, objects cannot be reused. In this scenario, Loqate will keep creating new objects and could possibly create the maximum 100 objects, causing an excessive amount of system memory to be used.
- **Loqate.Server.PoolTimeoutMS** – Default value is 5000 ms, the time a Loqate connection will remain in the connection pool.

Loqate Local Upgrade and Migration

Running Loqate locally include the following considerations for upgrades to STEP as well as migrating from Loqate Cloud to Loqate Local.

When a new Loqate version is released, users may update the reference data by running the Install Manager again, as defined in the **Loqate Local Software Install** topic. If a new version is released, the changes will show up in the data directory. For more information on the Loqate releases, see https://support.loqate.com/releases/, and for the Loqate release test plans, see https://support.loqate.com/release-test-plan/.

Upgrading STEP

When upgrading to a new version of the Stibo Systems Enterprise Platform (STEP), any updates to the Loqate component may contain a new Local Loqate Server version. This will be mentioned in the STEP release notes. In such cases the Loqate reference data packs must be updated.

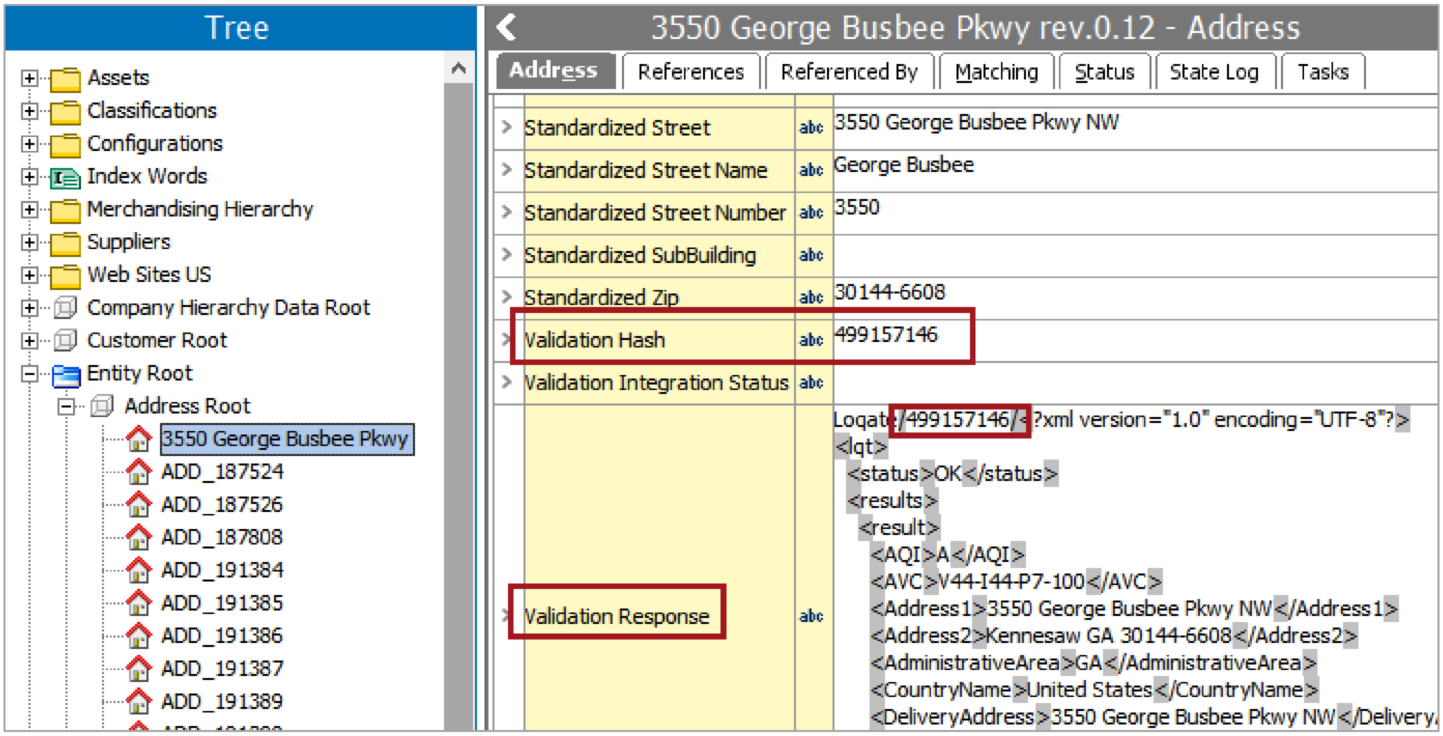
Loqate reference data packs must also be updated on a regular basis even if you are not upgrading to a new version of STEP. Loqate provides a quarterly release. This release will typically include updates to the Knowledge base (GKR) and often also reference data. Loqate’s release schedule can be found on their site: https://support.loqate.com/releases/.

Migrating Loqate from Cloud to Local

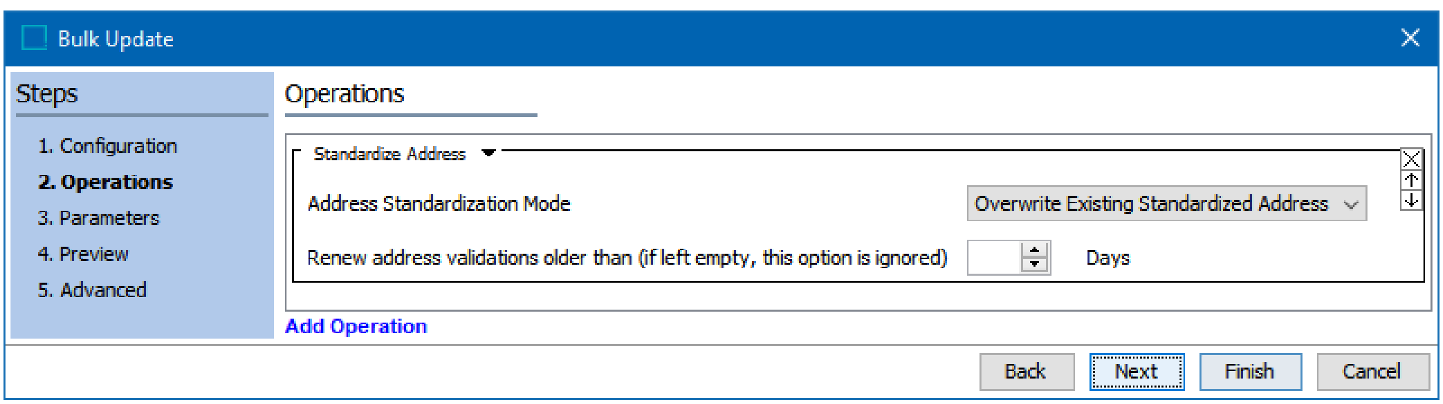
Current users of the Loqate Cloud solution can easily upgrade to the Loqate Local solution. A few differences exist between the Cloud and Local solutions, but the underlying functionality is the same.

Hash Value

The hash value is stored in a separate Validation Hash field in addition to the Validation Response field.



The ‘Generate Hash Value Only’ option can be used when running a Standardize Address bulk update or business action to populate the Validation Hash field on addresses that were previously validated using an older Loqate Cloud installation.



For more information on the ‘Generate Hash Value Only’ option, see the following topics:

- **Business Action Standardize Address** in the **Business Rules** documentation
- **Data Quality Standardize Address Operation** in the **Bulk Updates** documentation

Loqate Local Troubleshooting

To enable logging in the step.0 log for troubleshooting the Loqate and address services, add the following case-sensitive properties to the sharedconfig.properties file.

Important: These settings will create a large number of logging messages and should only be used during initial setup or for short-term troubleshooting.

```
#============================================================#
#LOQATE ADDRESS STANDARDIZATION
#=============================================#
Address.AddressQualityPlugin=LoqateAddressQuality
Address.Service.Loqate.GeocodeAll=true
# Address.Service.Loqate.License=x6I5b7ubMzEmMBjega0JaHEinBLKcrQY7FLCITMt

Loqate.Server.Data=/opt/stibo/step/resources/loqate/2.23.0.9136/LoqateData
Loqate.Server.PoolEnabled=True
Loqate.Server.MaxPoolSize=100
Loqate.Server.PoolTimeoutMs=5000

Log.Level.com.stibo.loqate=FINEST
Log.Level.com.stibo.loqate.localapi=FINEST
Log.Level.com.stibo.addressaspect=FINEST
Log.Level.com.stibo.addressaspect.workbench=FINEST
Log.Level.com.stibo.addressaspect.workbench.services=FINEST
Log.Level.com.stibo.addressaspect.workbench.easysetup=FINEST
```

Errors

An address validation call from STEP to Loqate that results in an error will be indicated on the address itself in the ‘Validation Integration Status’ field with the value: ‘Failed.’

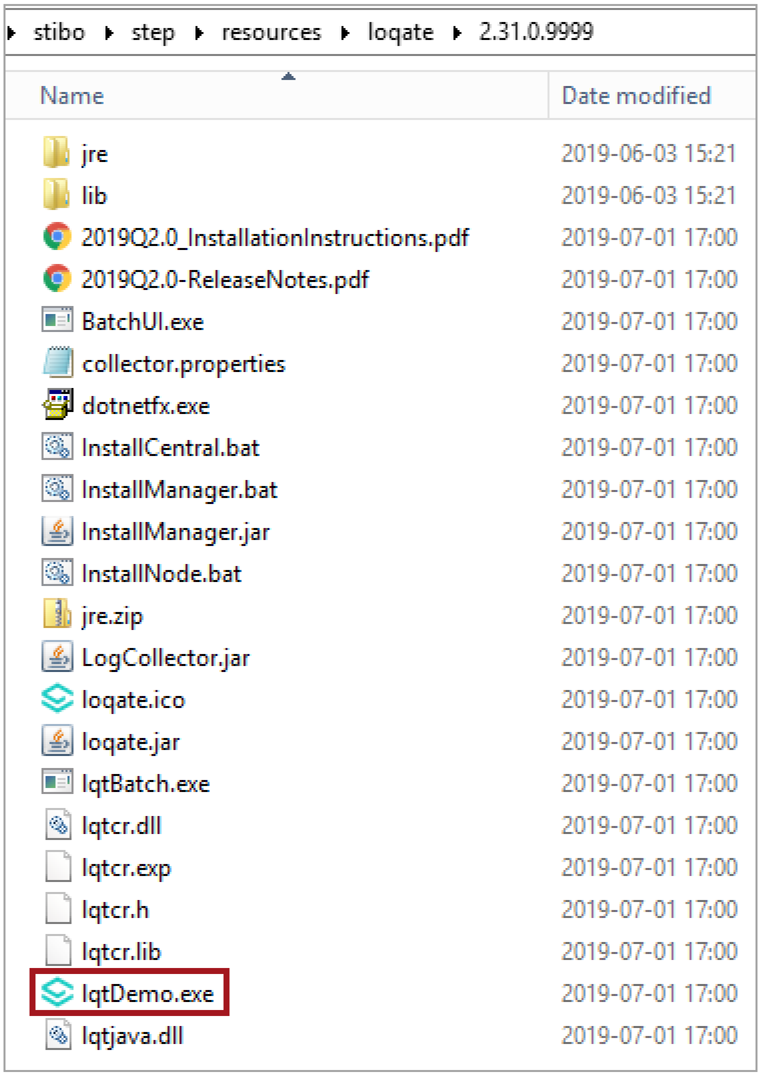
In case of errors, the ‘Validation Response’ field will include an XML element named ‘statusDetails.’ This field will contain the error message that was provided by the Local Loqate server. Of the possible error messages, this one will typically indicate that Loqate was not (properly) installed or not properly configured:

> 'Since Loqate returned an empty response, some settings might be misconfigured'

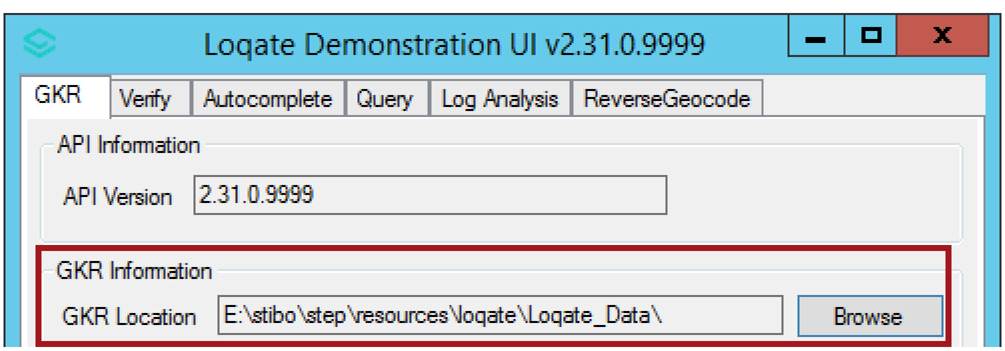
For example, this improper install or misconfiguration will occur if the ‘Knowledge Base Common’ data package was not installed. For more information, see the **Loqate Local Software Install** topic.

Loqate API Testing

The **lqtDemo.exe** file provides feedback to determine if an issue is with Loqate or with STEP. This file is in the Loqate server folder.

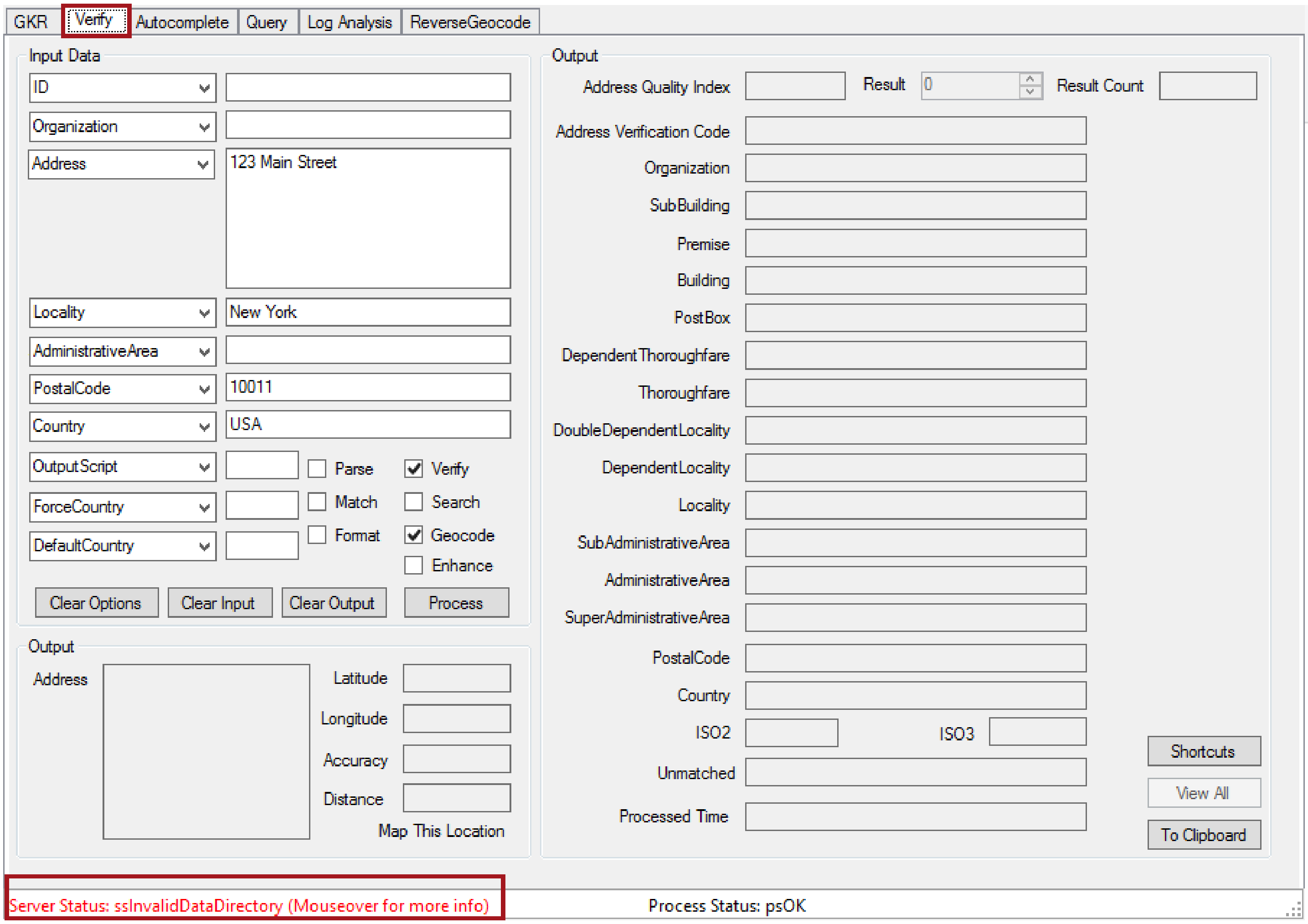


Open the lqtDemo.exe file and provide the location where the data packs have been created.

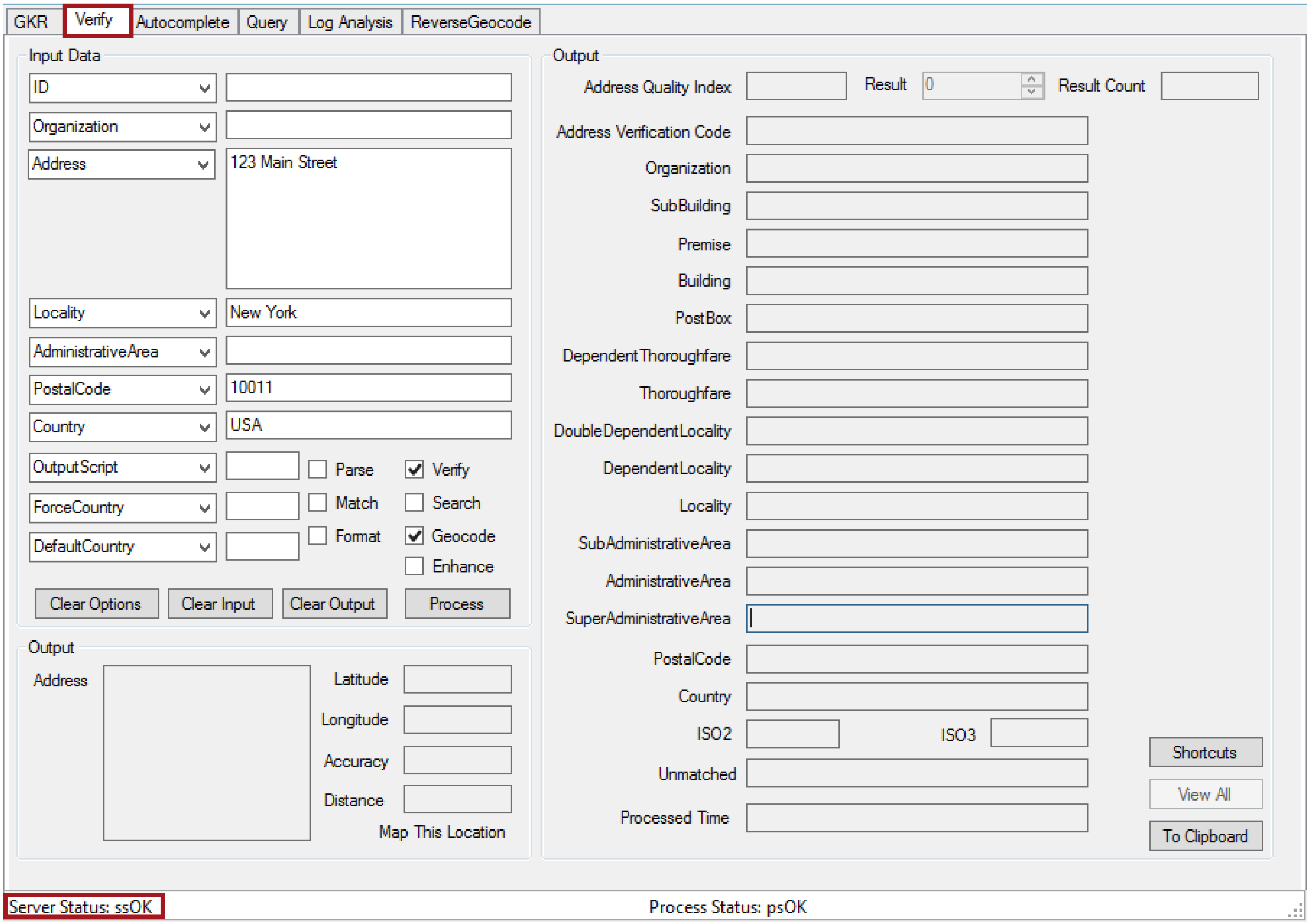


After setting the data packs location, select the 'Verify' tab at the top. This tab allows users to enter an address and test that standardized information is returned. The lower left corner error information displays.

In this example, the GKR location is set to a directory that does not exist.



Below, another address is entered, and returns no errors. This means any issues are not with the Loqate API and reference data installation, but with STEP or the STEP-Loqate connection (for example, a misconfiguration in sharedconfig.properties file).


GKR
Verify
Autocomplete
Query
Log Analysis
ReverseGeocode
Input Data
ID
Organization
Address
123 Main Street
Locality
New York
AdministrativeArea
PostalCode
10011
Country
USA
OutputScript
Parse
Verify
ForceCountry
Match
Search
DefaultCountry
Format
Geocode
Enhance
Clear Options
Clear Input
Clear Output
Process
Output
Address
Latitude
Longitude
Accuracy
Distance
Map This Location
Output
Address Quality Index
Result
0
Result Count
Address Verification Code
Organization
SubBuilding
Premise
Building
PostBox
DependentThoroughfare
Thoroughfare
DoubleDependentLocality
DependentLocality
Locality
SubAdministrativeArea
AdministrativeArea
SuperAdministrativeArea
PostalCode
Country
ISO2
ISO3
Shortcuts
Unmatched
View All
Processed Time
To Clipboard
Server Status: ssOK
Process Status: psOK

Loqate Local Command Line Tool

Poor address data can result in unusable match tuning because during the import to STEP addresses are standardized but algorithm tuning is based on the input address values. The Loqate command line tool is a service available within the Local Loqate solution that allows users to use the Loqate address standardization without integrating directly with the API or interfacing with STEP.

The command line tool can process input addresses that are stored in valid UTF-8 delimited files, which makes improves matching. Many input and output parameters are available to customize the results. More information on the available parameters can be found on the web at: https://support.loqate.com/support/batch-processor/

For example, use the Loqate command line tool to get better address data as part of the match tuning process. By tuning with standardized data, the output will more closely resemble the actual data migration.

Machine Learning-Based Auto Classification Integration

Machine learning is available for auto classification as a cloud service referred to as Machine Learning Auto Classification (MLAC). This service provides assisted or automated classification of products into the primary product hierarchy and/or into classification hierarchies where the product can exist only once.

The MLAC service uses supplied hierarchies and products already classified to make suggestions, or predictions, for classifying products that are not yet classified. The topic **Classification Predictions** shows the format of the prediction service requests and responses while the following topics describe how the prediction functionality can be integrated into / tested from STEP:

- Gateway IEP Configuration for MLAC
- Business Action Configuration for MLAC
- Testing MLAC Service Predictions
- Integrating MLAC Predictions in a Workflow

Before prediction requests can be made, the service algorithm must have been trained using customer specific data. For initial evaluation of the service, data for hierarchies and already classified products can be submitted manually to Stibo Systems. Manual data export is covered in the following topics:

- Manual Export of Hierarchies for MLAC
- Manual Export of Classified Products for MLAC

It is, however, also possible to configure STEP to upload the data to the service directly. This functionality is covered in the topics:

- Uploading Hierarchies
- Uploading Classified Products

Note: For information on standard automatic classification functionality without the cloud service, see the **Automatic Classification** documentation.

Classification Predictions

Predictions are suggestions for how products might be classified. The following examples show the format of the service requests and responses. The service API is documented in detail in Swagger available at **https://app.stibosystems.com/ds/mlac/v1**.

The prediction resource **https://app.stibosystems.com/ds/mlac/v1/predict** accepts POST requests with a JSON message body.

In STEP, the predict request can be sent from a business action via a gateway integration endpoint. For more information, see the topics in this section:

- Gateway IEP Configuration for MLAC
- Business Action Configuration for MLAC

Example Request

The request can contain information about a single or multiple unclassified products. Responses are synchronous with a number of classification predictions for each product.

The request body format is:

```
{
  "products": [
    {
      "id": "string",
      "sourceId": "string",
      "description": "string"
    }
  ]
}
```

The request body fields are:

- 'id' - Mandatory. Required for each product.
- 'description' - Optional. A description can improve the accuracy of predictions.
- 'sourceId' - Optional. A source ID should only be populated if it is significant for the classification and if it is supplied when uploading already classified products.

The request query parameters are:

- 'hierarchyIds' - Specifies the hierarchies that should be used to generate predictions. The hierarchy IDs provided must match the IDs used when uploading hierarchies and training products.
- 'maxSuggestionsPerHierarchy' - Optional. Limits how many predictions are returned for each hierarchy.

Example Response

A response for a single product, for two hierarchies with IDs ‘primary’ and ‘gs’ with ‘maxSuggestionsPerHierarchy’ set to 2, could look as follows:


```
{
  "predictions": [
    {
      "id": "ABC-123",
      "hierarchies": [
        {
          "hierarchyId": "primary",
          "predictions": [
            {
              "classificationId": "Lvl4-16772",
              "confidence": 92
            },
            {
              "classificationId": "Lvl4-16723",
              "confidence": 72
            }
          ]
        },
        {
          "hierarchyId": "gs",
          "predictions": [
            {
              "classificationId": "Bike Wheels",
              "confidence": 95
            },
            {
              "classificationId": "Bike Wheel Parts",
              "confidence": 69
            }
          ]
        }
      ]
    }
  ]
}
```


Notice that each prediction is accompanied with a confidence number indicating how confident the service is about the prediction.

Gateway IEP Configuration for MLAC

A Gateway Integration Endpoint (IEP) allows STEP to communicate with the MLAC service. Setup includes editing the properties file and configuring the endpoint.

For general information, see the **Gateway Integration Endpoints** topic in the **Data Exchange** documentation.

Prerequisites

Perform these prerequisites before configuring the IEP.

1. In the 'workarea' directory shared between the application servers, edit the sharedconfig.properties file. Add or modify the case-sensitive **RESTGateway.ServerURL** property to identify https://app.stibosystems.com/ds/mlac/v1 as the URL for the auto classification service as shown below.

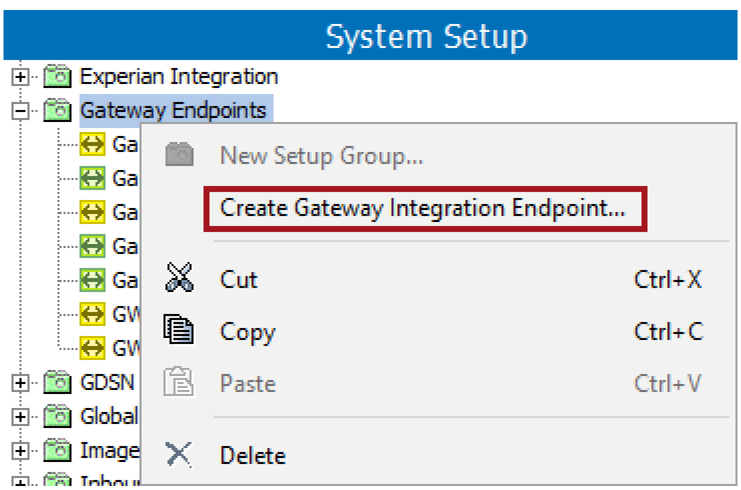
```
RESTGateway.ServerURL=1= https://app.stibosystems.com/ds/mlac/v1
```

2. Restart the server to implement the properties file changes.

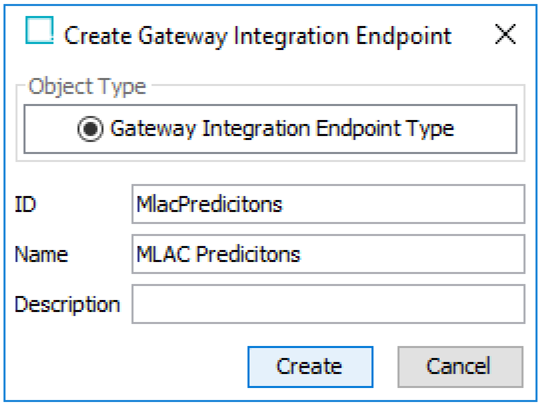
Configure the Gateway IEP

Use the following steps to create and configure the gateway endpoint for MLAC.

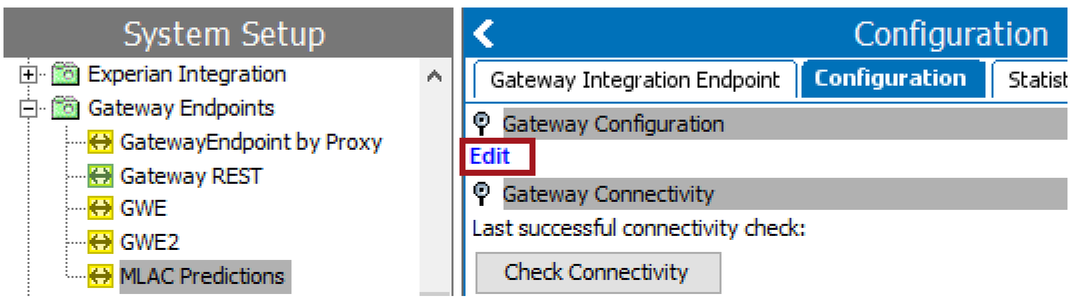
1. In workbench on System Setup, select the Gateway Integration Endpoint node, right-click the node, and select the **Create Gateway Integration Endpoint** option.



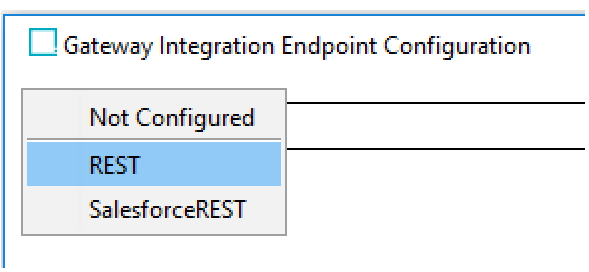
2. On the Create Gateway Integration Endpoint dialog, select an object type if required, and type an **ID** and **Name**. Description is optional. Click the **Create** button.



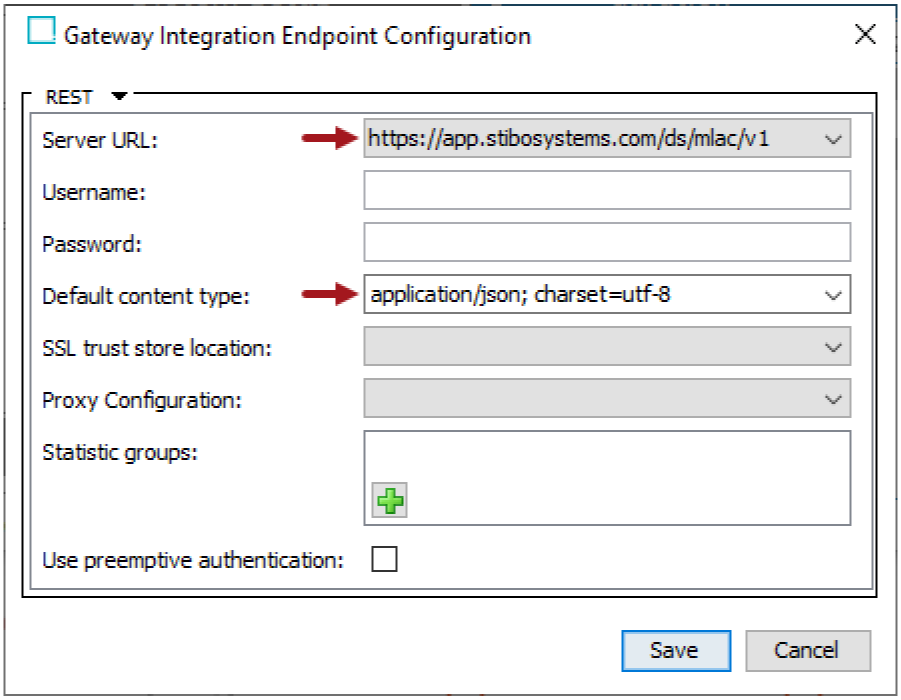
3. Select the new IEP to display the Gateway Integration Endpoint editor. On the Configuration tab, click the **Edit** link to open the ‘Gateway Integration Endpoint Configuration’ dialog.



4. Click the 'Not Configured' dropdown and select the **REST** option to display the configuration parameters.



5. Add settings for the following parameters:
   - For Server URL, select the prediction resource URL from the dropdown.
   - For Default content type, type **application/json; charset=utf-8**
   - Leave other parameters blank.
   - Click the **Save** button.



6. Select the IEP, right-click, and select **Enable Integration Endpoint** option.
7. Create a business action for use with MLAC as defined in the **Basic Business Action Configuration for MLAC** topic.

Business Action Configuration for MLAC

A business action is used to communicate with the MLAC prediction service and to get predictions for a single product.

For general information about business actions and functions, see the **Business Rules** documentation.

Important: The example scripts should not be used as-is without thorough testing, including updating the script to match object and link types that exist on your system. JavaScript variable names are case-sensitive.

Prerequisites

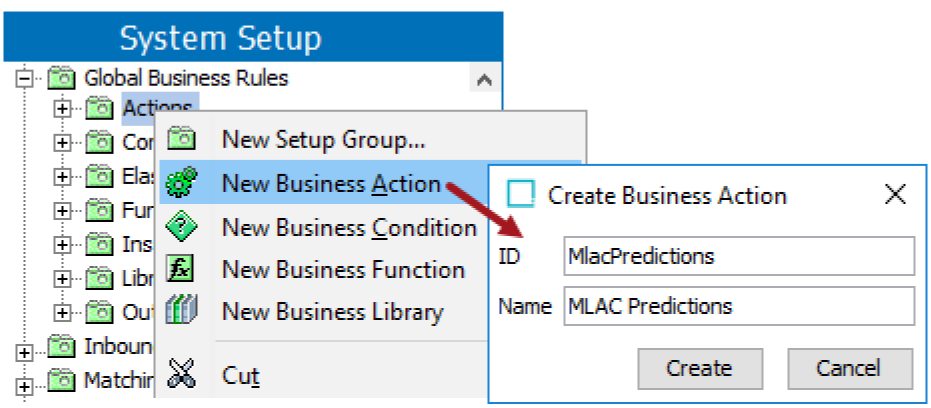
This business rule requires the gateway integration endpoint created in the **Gateway IEP Configuration for MLAC** topic.

Configure the Business Action

Use the following steps to create a basic MLAC business action:

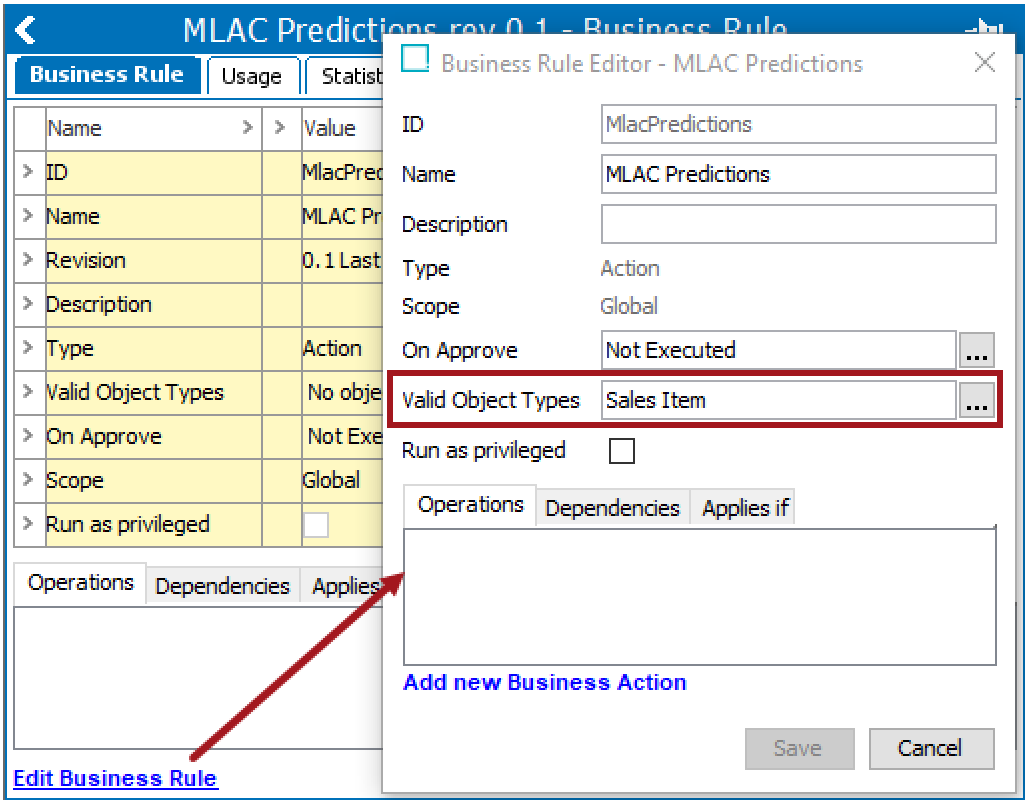
1. On System Setup, select the Global Business Rules node where the MLAC business rule will be saved. Right-click the node and select the **New Business Action** option.

   On the Create Business Action dialog, add an **ID** and a **Name** and click the **Create** button.

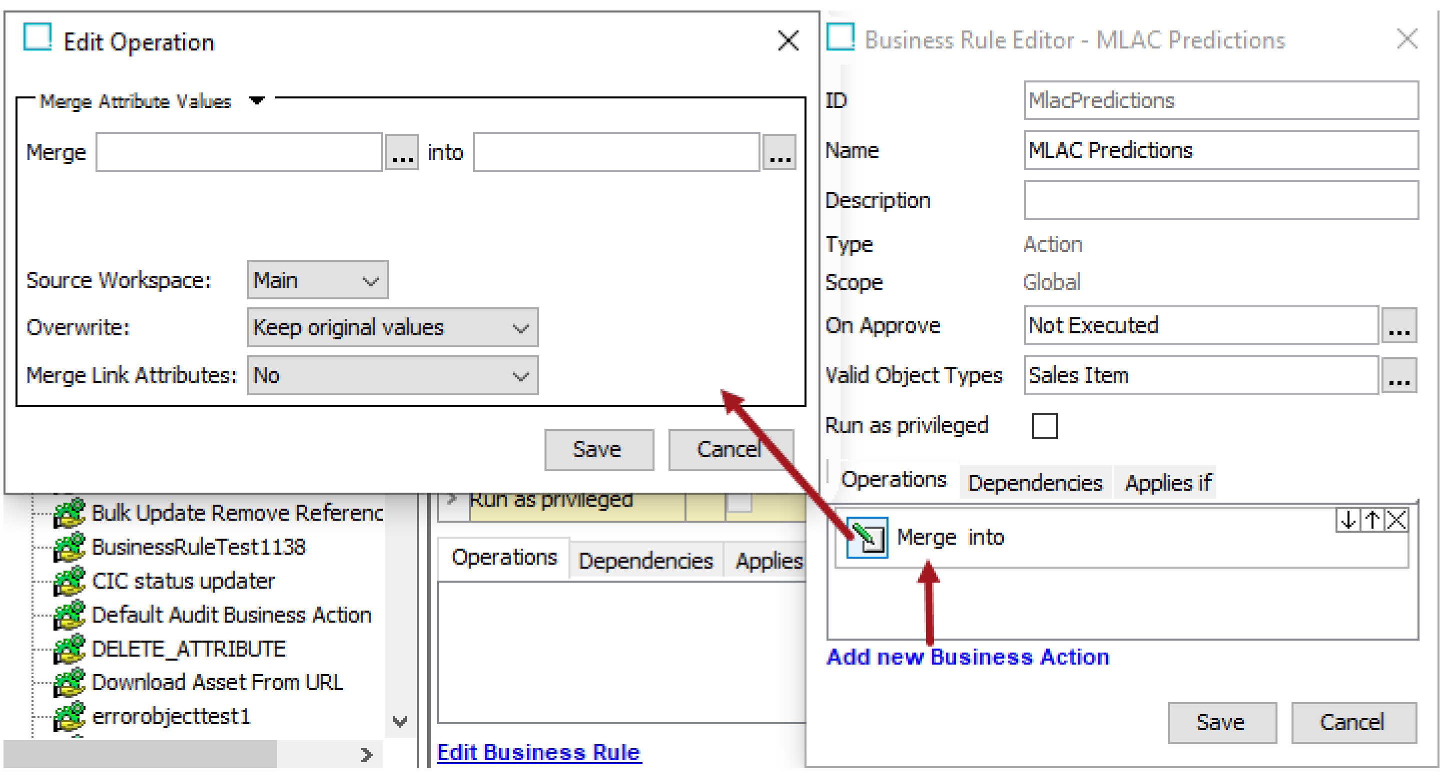


2. On the business rule editor, click the **Edit Business Rule** link dialog to display the Business Rule Editor dialog.

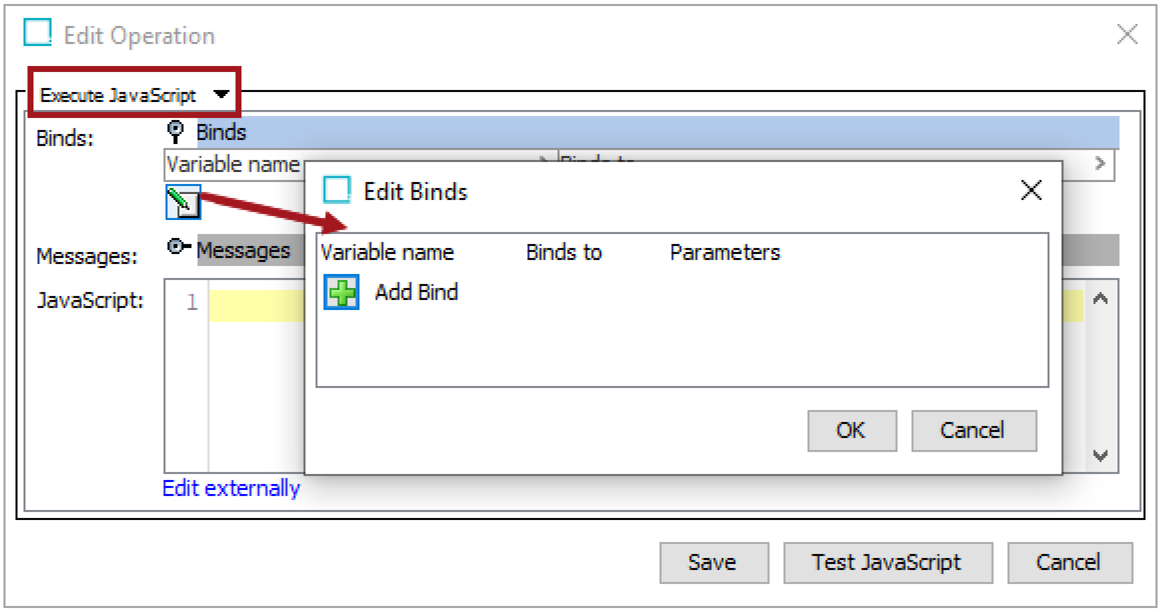
   - For the Valid Object Types parameter, add the product object type for which predictions are to be generated.



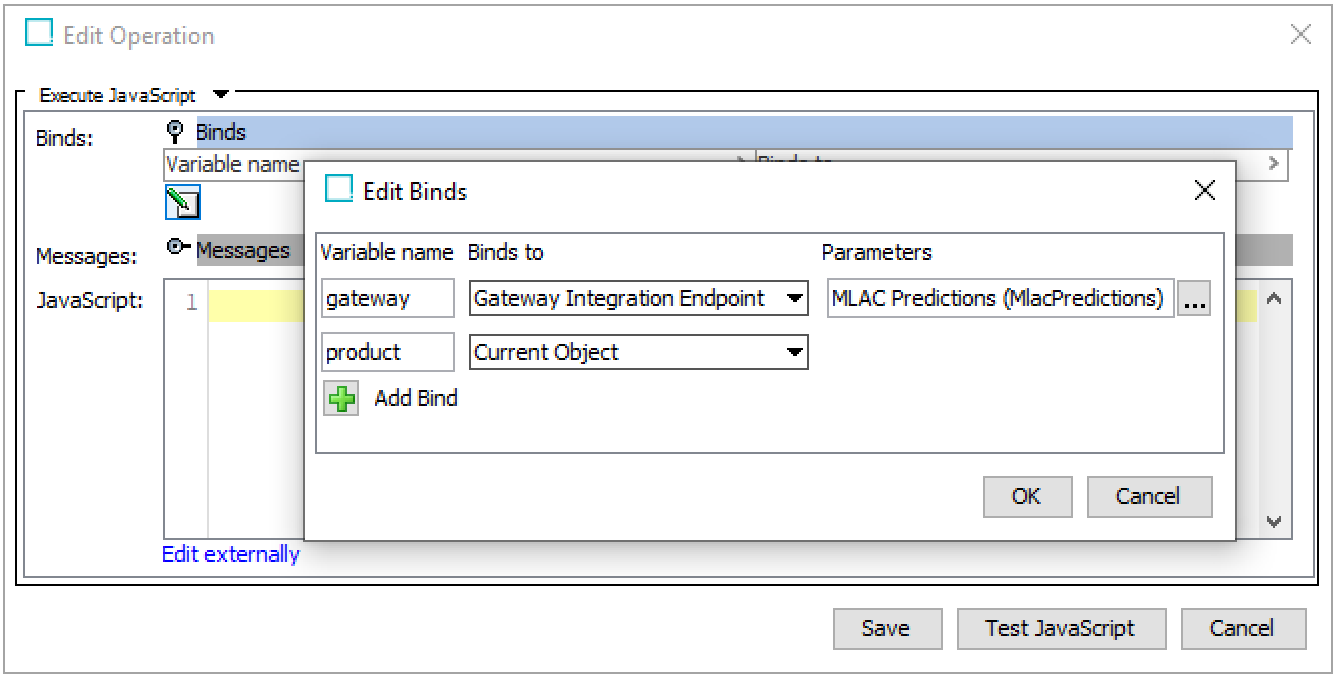
3. Click the **Add new Business Action** link to add an operation, and click the operation's edit button ( ).



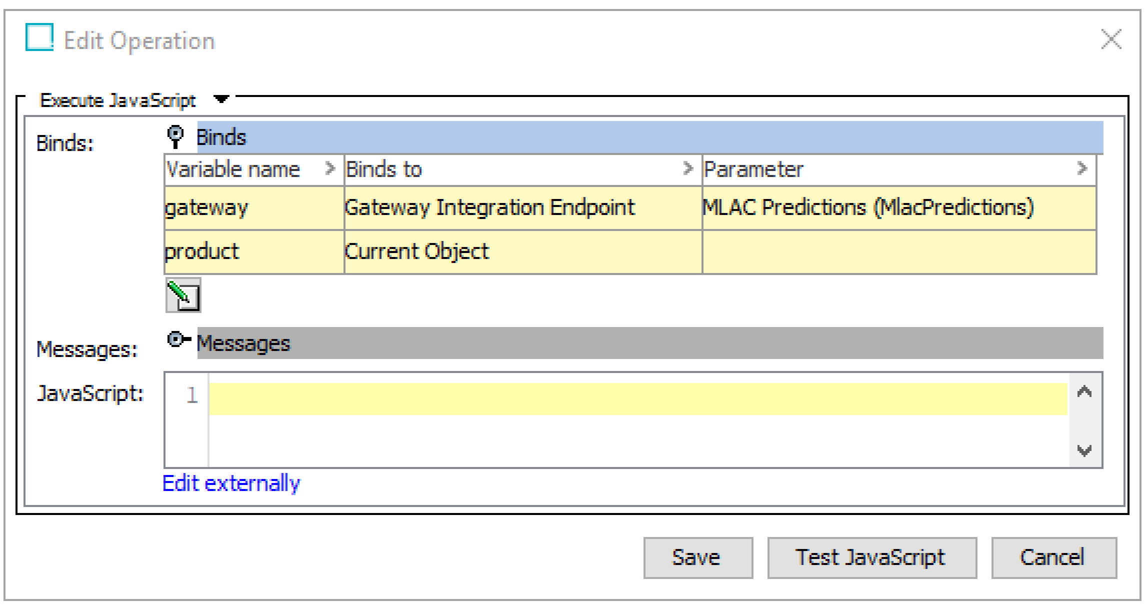
4. On the Edit Operation dialog:
   - Select the **Execute JavaScript** operation from the dropdown.
   - Click the edit button ( ) under the Binds flipper.



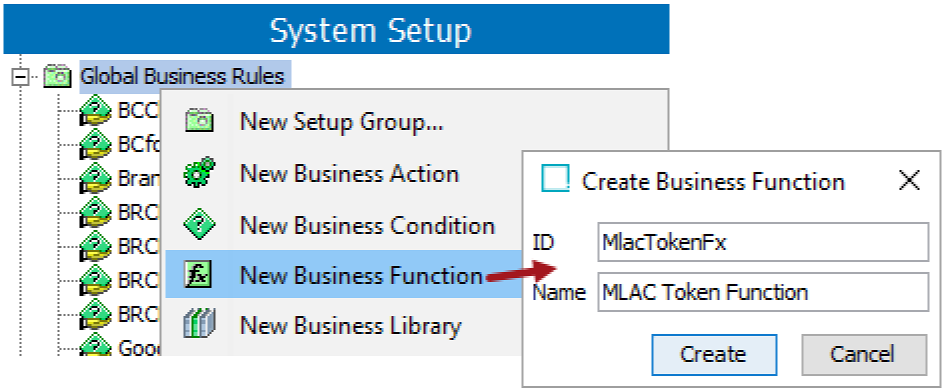
5. On the Edit Binds dialog:
   - Click the **Add Bind** button.
     - For 'Variable name' type **gateway**.
     - For 'Binds to' select the Configuration option, and select the **Gateway Integration Endpoint** option.
     - For 'Parameters' select the **MLAC gateway IEP** previously created.
   - Click the **Add Bind** button again.
     - For 'Variable name' type **product**.
     - For 'Binds to' select **Current Object**.
   - Click the **OK** button.



6. On the Edit Operation dialog, click the **Save** button.



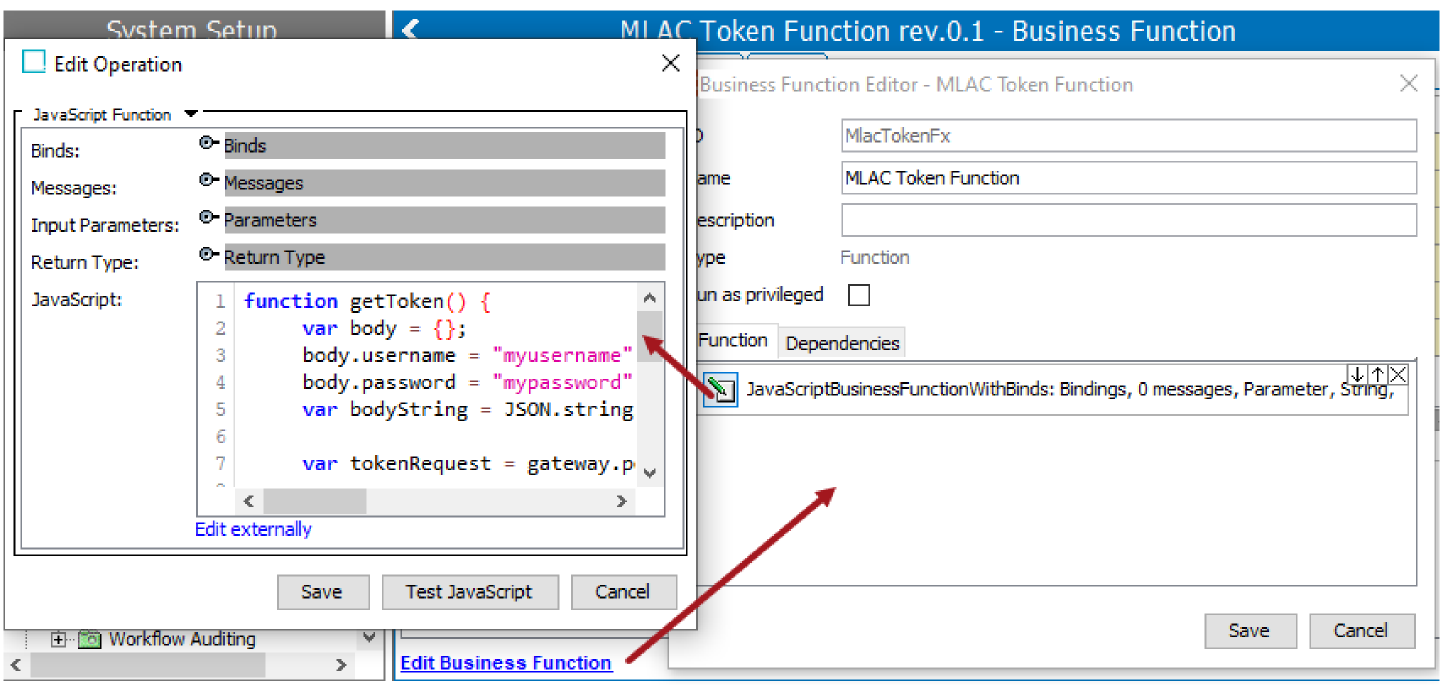
7. On the Business Rule Editor dialog, click the **Save** button to complete the business rule creation.
8. On System Setup, select the Global Business Rules node where the MLAC token function will be saved. Right-click the node and select the **New Business Action** option. On the Create Business Action dialog, add an **ID** and a **Name** and click the **Create** button.



9. In the business function editor, click the **Edit Business Function** link. Click the edit button () for the JavaScript function.
10. In the JavaScript parameter:
    - Add the **getToken** function shown below for obtaining a token from the token endpoint.

> **Important:** Replace the username ("myusername") and password ("mypassword") with your credentials.

- Click the **Save** buttons to close both dialogs and complete this function.



For example:

```
function getToken() {
	var body = {};
	body.username = "myusername";
	body.password = "mypassword";
	var bodyString = JSON.stringify(body);

	var tokenRequest = gateway.post().pathElements
("token").acceptContentType("text/plain").body(bodyString);

	var tokenResponse;
	try {
		tokenResponse = tokenRequest.invoke();
	} catch(e) {
		if (e.javaException instanceof
com.stibo.gateway.rest.RESTGatewayException) {
			throw "Error getting token: " + e.javaException.getMessage();
		} else {
			throw(e);
		}
	}
	return tokenResponse;
}
```

11. Create a **getProductJsObject** function that produces the JSON body for the request to be sent to the prediction service. This function uses the same relevant attributes that are also used when uploading data for already classified products.

> **Important:** When creating a JavaScript object, the string values retrieved from STEP should be concatenated with an empty string to produce JavaScript strings.

For example:

```
function getProductJsObject(prod) {
	var id = prod.getID();
	var vendor = prod.getValue("Vendor").getSimpleValue();
	var description = prod.getValue("ConsumerShortDescription").getSimpleValue();

	var jsObject = {};
	jsObject.id = "" + id;
	if (description) {
		jsObject.description = "" + description;
	}
	if (vendor) {
		jsObject.sourceId = "" + vendor;
	}
	return jsObject;
}

function createBody() {
	var prodJsObject = getProductJsObject(product);
	var prodsArray = [];
	prodsArray.push(prodJsObject);
	var bodyJsObject = {};
	bodyJsObject.products = prodsArray;
	return JSON.stringify(bodyJsObject);
}
```

12. Create a **getPredictionsString** function that composes the prediction request, sends it, and returns the result.

Important: Hierarchy IDs must match IDs used when uploading hierarchies.

For example:

```
function getPredictionsString(tokenResponse, body) {
	var queryParams = new java.util.HashMap();
	queryParams.put("hierarchyIds", "primaryHierarchy");
	queryParams.put("maxSuggestionsPerHierarchy", "2");

	var request = gateway
		.post()
		.pathElements("predict")
		.header("Authorization", tokenResponse)
		.pathQuery(queryParams)
		.body(body);
```

```
        var response;
        try {
                response = request.invoke();
        } catch(e) {
                if (e.javaException instanceof
com.stibo.gateway.rest.RESTGatewayException) {
                        throw "Error getting prediction: " + e.javaException.getMessage();
                } else {
                        throw(e);
                }
        }
        return response;
}
```

13. Create an optional **getResponseJsObject** function for converting the Java String prediction response to a JavaScript object. This function is not required if, for example, the prediction response is written as an attribute value.

    For example:

```
function getResponseJsObject(responseString) {
        var jsResponseString = "" + responseString;
        return JSON.parse(jsResponseString);
}
```

14. Add code to call the functions.

    For example:

```
var tokenResponse = getToken();

var body = createBody();

var responseString = getPredictionsString(tokenResponse, body);
```

15. Create a script to handle the response. For example, classify the product, create a workflow task, or write the prediction results to a value.

    Below is a full example script.

```
function getToken() {
        var body = {};
        body.username = "myusername";
        body.password = "mypassword";
        var bodyString = JSON.stringify(body);

        var tokenRequest = gateway.post().pathElements
("token").acceptContentType("text/plain").body(bodyString);

        var tokenResponse;
        try {
                tokenResponse = tokenRequest.invoke();
```

```
        } catch(e) {
                if (e.javaException instanceof
com.stibo.gateway.rest.RESTGatewayException) {
                        throw "Error getting token: " + e.javaException.getMessage();
                } else {
                        throw(e);
                }
        }
        return tokenResponse;
}

function getProductJsObject(prod) {
        var id = prod.getID();
        var vendor = prod.getValue("Vendor").getSimpleValue();
        var description = prod.getValue
("ConsumerShortDescription").getSimpleValue();

        var jsObject = {};
        jsObject.id = "" + id;
        if (description) {
                jsObject.description = "" + description;
        }
        if (vendor) {
                jsObject.sourceId = "" + vendor;
        }
        return jsObject;
}

function createBody() {
        var prodJsObject = getProductJsObject(product);
        var prodsArray = [];
        prodsArray.push(prodJsObject);
        var bodyJsObject = {};
        bodyJsObject.products = prodsArray;
        return JSON.stringify(bodyJsObject);
}

function getPredictionsString(tokenResponse, body) {
        var queryParams = new java.util.HashMap();
        queryParams.put("hierarchyIds", "primaryHierarchy");
        queryParams.put("maxSuggestionsPerHierarchy", "2");

        var request = gateway
                .post()
                .pathElements("predict")
                .header("Authorization", tokenResponse)
                .pathQuery(queryParams)
                .body(body);


        var response;
```

```
        try {
                response = request.invoke();
        } catch(e) {
                if (e.javaException instanceof
com.stibo.gateway.rest.RESTGatewayException) {
                        throw "Error getting prediction: " + e.javaException.getMessage();
                } else {
                        throw(e);
                }
        }
        return response;
}

function getResponseJsObject(responseString) {
        var jsResponseString = "" + responseString;
        return JSON.parse(jsResponseString);
}

var tokenResponse = getToken();

var body = createBody();

var responseString = getPredictionsString(tokenResponse, body);
```

Testing MLAC Service Predictions

After creating the basic business action as defined in the **Business Action Configuration for MLAC** topic, testing can be performed using the following methods:

- via a business action to test unclassified products individually.
- via a bulk update process to test a large number of products as a group.

For general information about editing and testing business actions, see the **Editing a Business Rule** topic and the **Testing a Business Rule** topic, both in the **Business Rules** documentation.

Important: The example scripts should not be used as-is without thorough testing, including updating the script to match object and link types that exist on your system. JavaScript variable names are case-sensitive.

Testing Products Individually

Use the following steps to modify your business action to test an individual product.

1. Edit the MLAC predictions business action to include a Logger bind.
2. In the JavaScript parameter, log the 'responseString'. For example:

```
logger.info(responseString);
```

3. Test the business action against an unclassified product.

Testing Products in Groups

To test a set of unclassified products, the MLAC business action can first write the returned predictions to an attribute on the unclassified products. Then the MLAC business action can be executed from a bulk update process for a group of products.

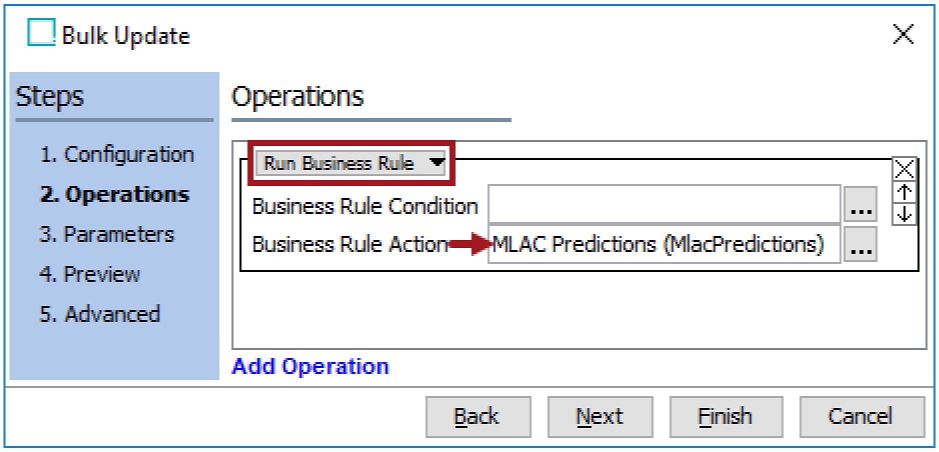
For general information, see the **Bulk Updates** documentation.

Use the following steps to modify your business action to test a group of products.

1. Edit the MLAC predictions business action to write the predict response to a value. For example:

```
product.getValue('Prediction').setSimpleValue(responseString)
```

2. Use Tree or Search to select the products to test via bulk update.
3. Run the Bulk Update option for the selected products.
4. Choose the ‘Run Business Rule’ bulk update operation from the dropdown.
5. For the Business Rule Action parameter, select the MLAC predictions business action.



6. Click the **Finish** button to run the bulk update process.

Integrating MLAC Predictions in a Workflow

The prediction functionality can be used in a production setup by calling the business action that retrieves the predictions from a product onboarding workflow. The workflow can then branch based on the prediction confidence, resulting in the product being automatically classified for a prediction with a high confidence, while a creating a user task with the prediction information if no predictions meet the desired confidence threshold.

Given that the prediction service response time can be several seconds, to avoid blocking users, the configuration examples in this topic make use of an event processor (EP). The EP can be configured to use no batch and batch options and sends the prediction request from the event processor instead of sending it from an action executed in the workflow. Options are:

- Using the ‘Execute Business Action’ event processor plugin — a business action will be executed once per queued event and a prediction request is sent for each event.
- Using the 'Execute Business Action for Event Batch' event processor plugin — a business action will be executed via a batch, meaning that data for multiple events will be handled with a single action invocation and a single request.

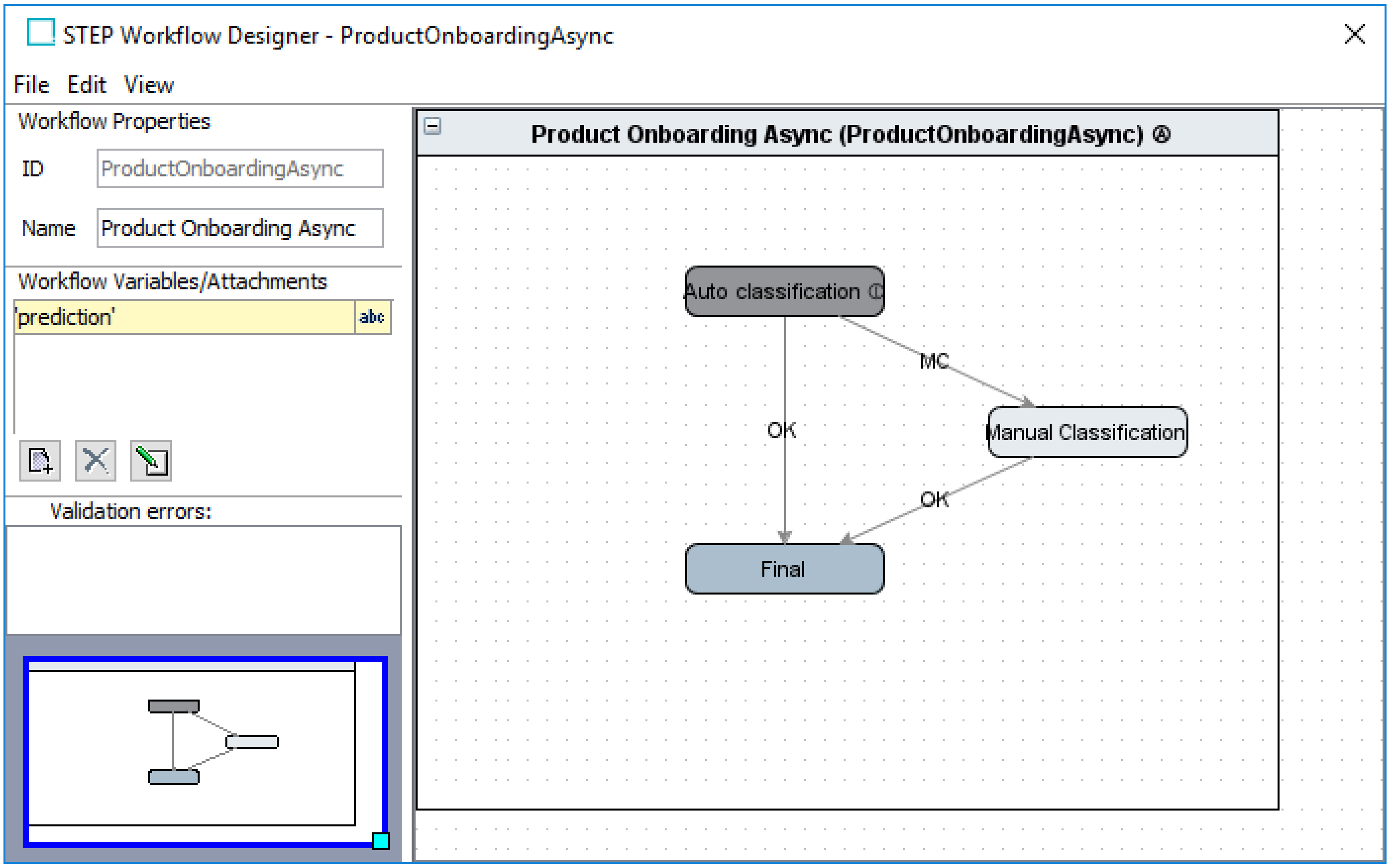
The full process of implementing the MLAC auto classification functionality in a workflow includes these steps:

1. Create an Onboarding Workflow
2. Define a Derived Event Type
3. Create an Event Processor Filter Condition
4. Create a Prediction Business Action
5. Create and Configure an MLAC Event Processor
6. Create a Workflow Business Action

Create an Onboarding Workflow

While the MLAC auto classification logic could be part of a larger product onboarding flow, states not relevant for auto classification have been left out of this example.

For initial (and simplified) testing, create a workflow based on the image below, including the defined workflow states. After verifying the functionality, these states can be added to your existing onboarding workflow.



For general information, see the **Workflows** documentation.

Workflow States

The states for this onboarding / classification workflow are:

- **Auto classification** is a system state that does not represent a human task. When a product enters this state, an event is placed on the queue for the event processor to make the prediction request and, dependent on the confidence of the best prediction, either classify the product and trigger the 'OK' event or trigger the 'MC' event.
- **Manual Classification** is a human task to manually classify a product with a low prediction confidence. The predictions returned from the service are made visible to the user via the ‘prediction’ workflow variable to which the prediction response will be written.
- **Final** is a basic mandatory final state.

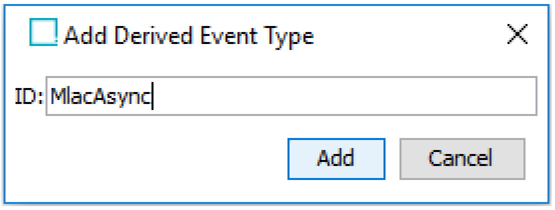
The following sections describe the configuration necessary for this setup.

Define a Derived Event Type

The event to be produced from within the workflow and placed on the queue for the event processor should be of a user-defined type, also called a derived event.

To create a derived event:

1. In System Setup, click the 'Derived Events node to open the editor.
2. Click the **Add Derived Event Type** link.
3. Add an **ID** for the event type.



4. Click the **Add** button to save the event type.

For more information, see the **Derived Events** topic in the **System Setup / Super User Guide** documentation.

Create an Event Processor Filter Condition

An event filter is required on the Event Triggering Definitions tab of the event processor to ensure that only events generated from the workflow are allowed on the event processor queue. For more information, event triggering definitions for event processors are the same as for OIEPs as defined in the **OIEP - Event-Based - Event Triggering Definitions Tab** topic in the **Data Exchange** documentation.

This event filter is a JavaScript-based business condition that is valid for all object types.

In the JavaScript editor create a bind with:

- 'Variable name' set to **currentEventType**
- 'Binds to' set to **Current Event Type** (in the Event Handling group)
- Add the following to the JavaScript parameter:

```
var isDerivedEvent = currentEventType instanceof
com.stibo.core.domain.eventqueue.DerivedEventType;
var idIsCorrect = currentEventType.getID() == "MlacAsync";

return isDerivedEvent && idIsCorrect;
```

Create a Prediction Business Action

No Batch Processing

Using the 'Execute Business Action' event processor plugin executes the configured prediction (confidence) business action once for each event in a non-batched manner. This business action has access to the node (product) for which the event was generated via the 'Current Object' bind. This allows the logic described in the **Business Action Configuration for MLAC** topic to be reused.

Below is a full-script example with bodies of functions described earlier omitted and a TODO for actually classifying the product given a prediction with a sufficient confidence is returned. If not duplicating and extending a business action created earlier based on the information in the Business Action Configuration for MLAC topic, binds should be configured as explained in that section and the action should either be made applicable for the product object type or for all object types.

```
function getToken() {
	//Omitted. See "Business Action Configuration for MLAC"
}

function getProductJsObject(prod) {
	//Omitted. See "Business Action Configuration for MLAC"
}

function createBody() {
	//Omitted. See "Business Action Configuration for MLAC"
}

function getPredictionsString(tokenResponse, body) {
	//Omitted. See "Business Action Configuration for MLAC"
}

function getResponseJsObject(responseString) {
	//Omitted. See "Business Action Configuration for MLAC"
}

function createPredictionsVariableValue(responseJsObject) {
	var predictions = responseJsObject.predictions[0].hierarchies
[0].predictions;
	var predictionsString = "";
	for (var i = 0; i < predictions.length; i++) {
		if (i > 0) {
			predictionsString = predictionsString + ", "
		}
		var prediction = predictions[i];
		predictionsString = predictionsString + prediction.classificationId +
": " + prediction.confidence;
	}
	return predictionsString;
}

var confidenceThreshold = 90;
var wfId = "ProductOnboardingAsync";
var stateId = "Auto-classification";

var instance = product.getWorkflowInstanceByID(wfId);
if (!instance) {
	throw "Product with ID '" + product.getID() + "' is not active in
workflow '" + wfId + "'";
}
var task = instance.getTaskByID(stateId);
```

```
if (!task) {
	throw "Product with ID '" + product.getID() + "' is not in expected
state '" + stateId + "'";
}

var tokenResponse = getToken();
var body = createBody();
var responseString = getPredictionsString(tokenResponse, body);
var responseJsObject = getResponseJsObject(responseString);

var confidence = parseInt(responseJsObject.predictions[0].hierarchies
[0].predictions[0].confidence);

if (confidence >= confidenceThreshold) {
	// TODO: Classify the product in classification jsonResp.predictions
[0].hierarchies[0].predictions[0].classificationId
	task.triggerByID("OK", "Triggered by script");
} else {
	var predictionsString = createPredictionsVariableValue
(responseJsObject);
	instance.setSimpleVariable("prediction", predictionsString);
	task.triggerByID("MC", "Triggered by script");
}
```

Batch Processing

For improved business action processing, a batch business action allows multiple events to be handled with a single action invocation and a single request.

The batch business action to be used for the batch event processor plugin is similar to the one described above. The main difference is that the batch version does not use 'Current Object' as its input, but instead uses the 'Current Event Processor Event Batch' bind to access a batch of queued events. Additionally, the batch business action composes a single request with data for multiple products and similarly handles the response containing predictions for multiple products.

Use the following steps to create the batch business action:

1. The business action should have the following binds:
    - gateway = Gateway Integration Endpoint (under the Configuration group), with the MLAC IEP selected
    - logger = Logger
    - manager = STEP Manager
    - eventBatch = Current Event Processor Event Batch (under the Event Handling group)

Binds

Variable name	Binds to	Parameter
gateway	Gateway Integration Endpoint	MLAC Predicitons (MlacPredicitons)
logger	Logger	
manager	STEP Manager	
eventBatch	Current Event Processor Event Batch	

2. For the JavaScript parameter, add a script to compose a request and handle the response.

   The full JavaScript example is below.

```
function getToken() {
        var body = {};
        body.username = "myusername";
        body.password = "mypassword";
        var bodyString = JSON.stringify(body);

        var tokenRequest = gateway.post().pathElements
("token").acceptContentType("text/plain").body(bodyString);

        var tokenResponse;
        try {
                tokenResponse = tokenRequest.invoke();
        } catch(e) {
                if (e.javaException instanceof
com.stibo.gateway.rest.RESTGatewayException) {
                        throw "Error getting token: " + e.javaException.getMessage();
                } else {
                        throw(e);
                }
        }
        return tokenResponse;
}

function getProductJsObject(prod) {
        var id = prod.getID();
        var vendor = prod.getValue("Vendor").getSimpleValue();
        var description = prod.getValue
("ConsumerShortDescription").getSimpleValue();

        var jsObject = {};
        jsObject.id = "" + id;
        if (description) {
                jsObject.description = "" + description;
        }
        if (vendor) {
                jsObject.sourceId = "" + vendor;
        }
```

```
        return jsObject;
}

function createBody(prodsArray) {
        var bodyJsObject = {};
        bodyJsObject.products = prodsArray;
        return JSON.stringify(bodyJsObject);
}

function getPredictionsString(tokenResponse, body) {
        var queryParams = new java.util.HashMap();
        queryParams.put("hierarchyIds", "primaryHierarchy");
        queryParams.put("maxSuggestionsPerHierarchy", "5");

        var request = gateway
                .post()
                .pathElements("predict")
                .header("Authorization", tokenResponse)
                .pathQuery(queryParams)
                .body(body);

        var response;
        try {
                response = request.invoke();
        } catch(e) {
                if (e.javaException instanceof
com.stibo.gateway.rest.RESTGatewayException) {
                        throw "Error getting prediction: " + e.javaException.getMessage();
                } else {
                        throw(e);
                }
        }
        return response;
}

function getResponseJsObject(responseString) {
        var jsResponseString = "" + responseString;
        return JSON.parse(jsResponseString);
}

function createPredictionsVariableValue(predictions) {
        var predictionsString = "";
        for (var i = 0; i < predictions.length; i++) {
                if (i > 0) {
                        predictionsString = predictionsString + ", "
                }
                var prediction = predictions[i];
                predictionsString = predictionsString + prediction.classificationId +
": " + prediction.confidence;
        }
        return predictionsString;
```

```
}

function getProductsArrayFromBatch() {
	var products = [];

	var batchIterator = eventBatch.getEvents().iterator();

	while (batchIterator.hasNext()) {
		var node = batchIterator.next().getNode();
		if (node instanceof com.stibo.core.domain.Product) {
			var instance = getInstance(node);
			var task = getTask(instance);
			if (task) {
				products.push(getProductJsObject(node));
			} else {
				logger.warning("Product with ID '" + node.getID() +
				 "' is not in the expected state '" + stateId +
				 "' in workflow '" + wfId + "'. No predictions will be
				 obtained for the product.");
			}
		}
	}
	return products;
}

function getInstance(node) {
	return node.getWorkflowInstanceByID(wfId);
}

function getTask(instance) {
	if (!instance) {
		return null;
	}
	return instance.getTaskByID(stateId);
}

function handlePredictions(predictions) {
	for (var index in predictions) {
		var prediction = predictions[index];
		var id = prediction.id;
		var product = manager.getProductHome().getProductByID(prediction.id);
		if (product) {
			var instance = getInstance(product);
			var task = getTask(instance);
			if (task) {
				var confidence =
parseInt(prediction.hierarchies[0].predictions[0].confidence);
				if (confidence >= confidenceThreshold) {
					// TODO: Classify the product in classification
```

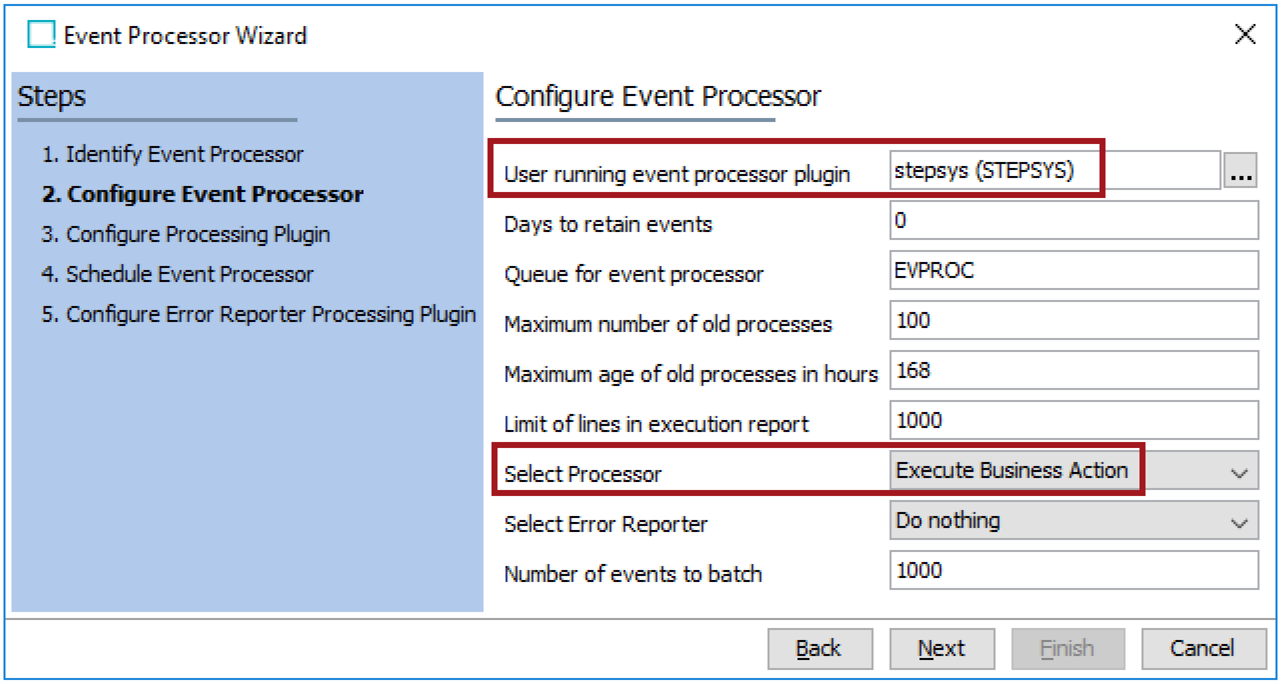
```
prediction.hierarchies[0].predictions[0].classificationId
                                        task.triggerByID("OK", "Triggered by script");
                                } else {
                                        var predictionsString =
createPredictionsVariableValue(prediction.hierarchies[0].predictions);
                                        instance.setSimpleVariable("prediction",
                                         predictionsString); task.triggerByID("MC",
                                         "Triggered by script");
                                }
                        }
                }
        }
}

var confidenceThreshold = 90;
var wfId = "ProductOnboardingAsync";
var stateId = "Auto-classification";

var products = getProductsArrayFromBatch();
var body = createBody(products);

var tokenResponse = getToken();
var responseString = getPredictionsString(tokenResponse, body);
var responseJsObject = getResponseJsObject(responseString);

handlePredictions(responseJsObject.predictions);
```

Create and Configure an MLAC Event Processor

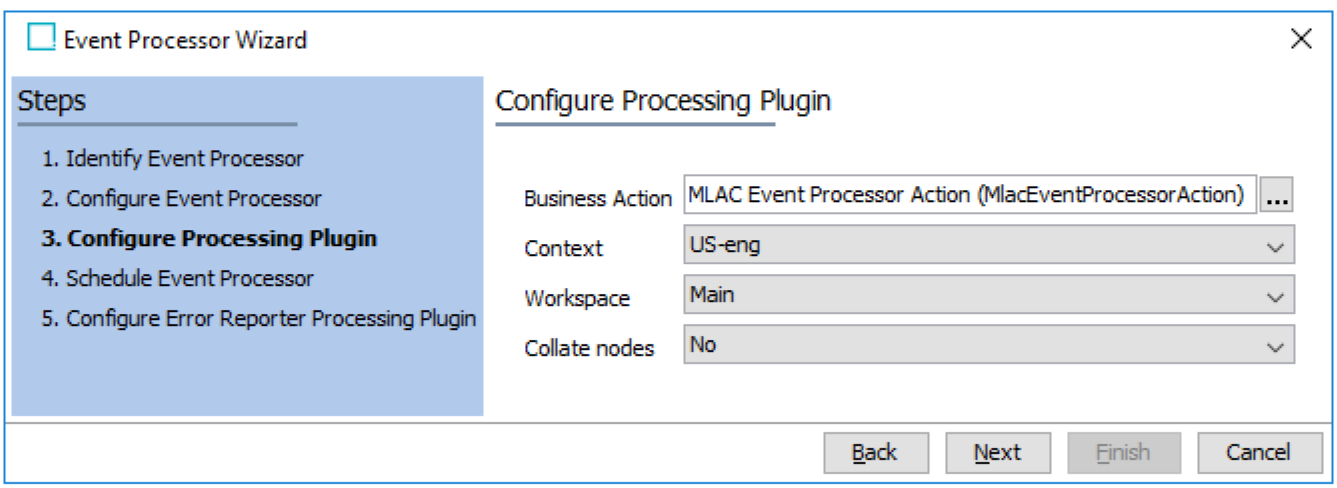
Use the steps below to create and configure the event processor required for the auto classification setup. For detailed information on event processors, see the **Creating an Event Processor** topic in the **System Setup / Super User Guide** documentation.

No Batch Processing

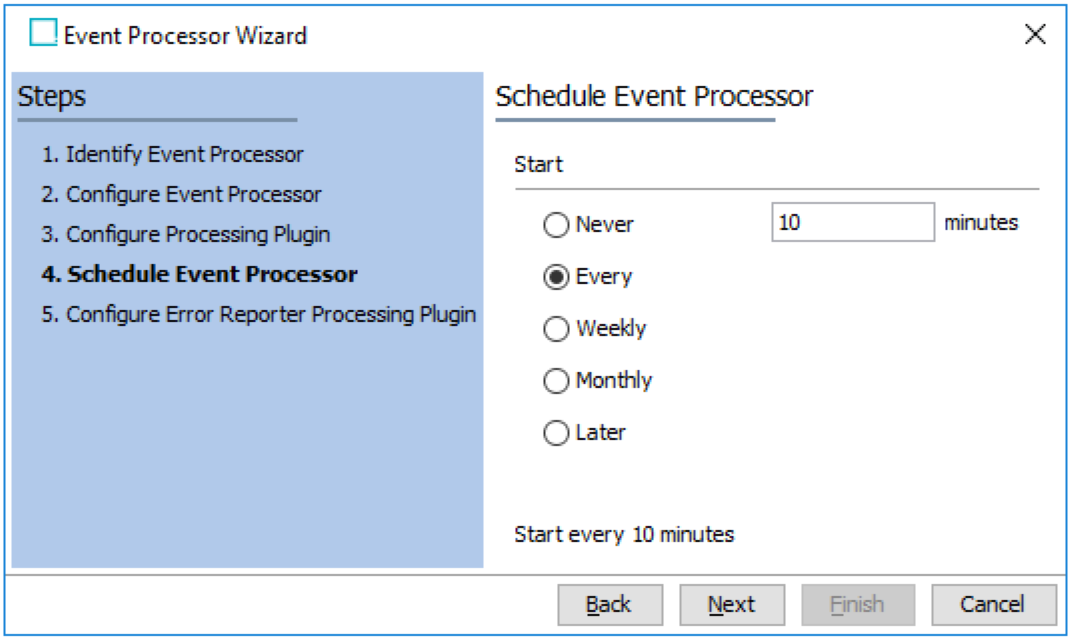
1. Create a new event processor and on the 'Configure Event Processor' step:
    - For the 'User running event processor plugin' parameter, select **a privileged system user**.
    - For the 'Select Processor' parameter, choose the **Execute Business Action** processor plugin.



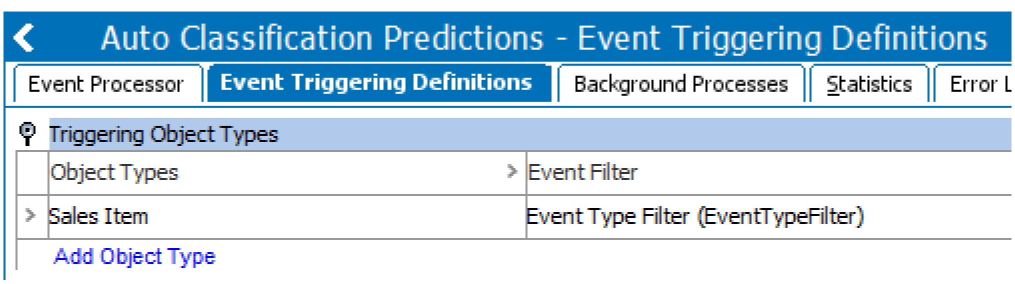
2. On the 'Configure Processing Plugin' step:
   - For the Business Action parameter, select **the previously created business action**.
   - For the Context parameter, select the **context in which the action should be executed**. Product data (like the description) will be read from this context.
   - For the Workspace parameter, select the **workspace in which the action should be executed**. Product data (like the description) will be read from this context.



3. On the ‘Schedule Event Processor’ step, specify how frequently the event processor should handle events on the queue.



4. Click the **Next** button and the **Finish** button to close the wizard.
5. On the event processor, click the Event Triggering Definitions tab and open the Triggering Object Types flipper.
   - Click the **Add Object Type** link and add your product object type. It displays in the Object Types column.
   - Click in the Event Filter column to display an ellipsis (…), click the ellipsis button (…) and add the event filter business condition created previously.



6. On the Event Processor tab, set the Queue Status parameter to **Read Events**.
7. Right-click the event processor and select the **Enable Event Processor** option.

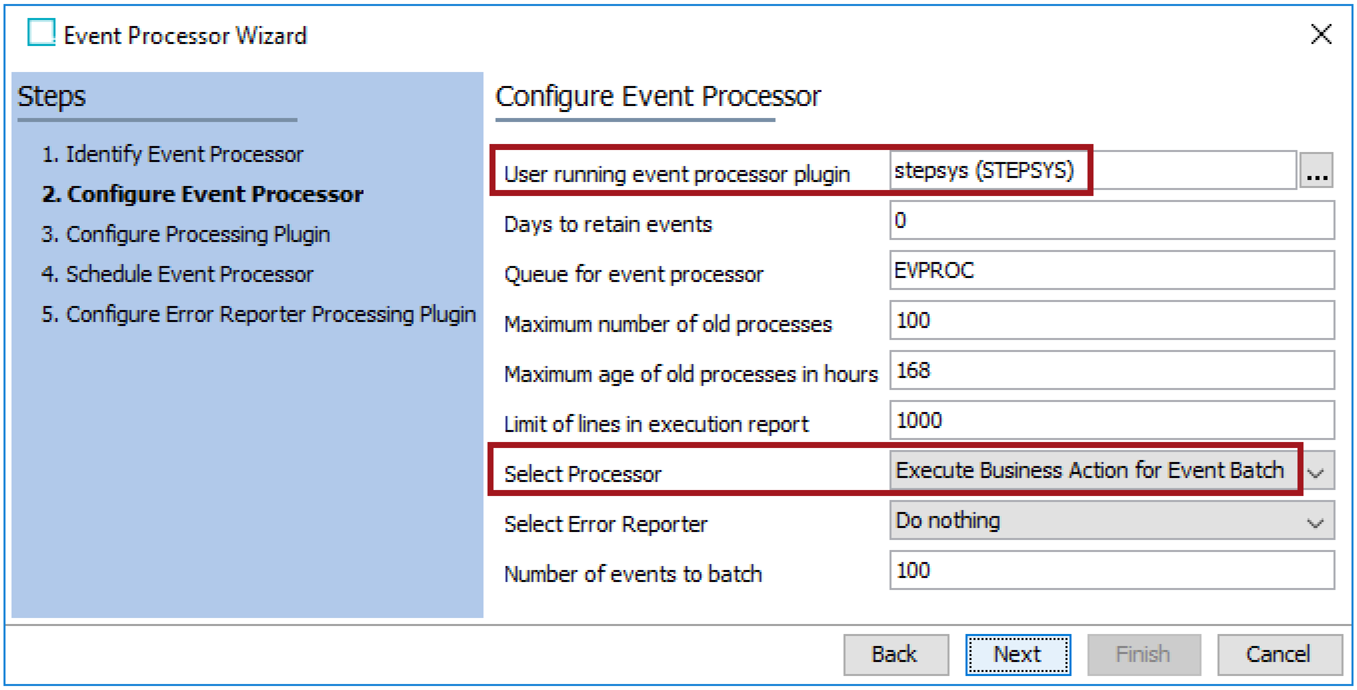
Batch Processing

The event processor used for the batch functionality is configured almost exactly like the non-batched version described in the previous section. The difference is that the batch processor uses the 'Execute Business Action for Event Batch' plugin and references the business action that can handle a batch of events.

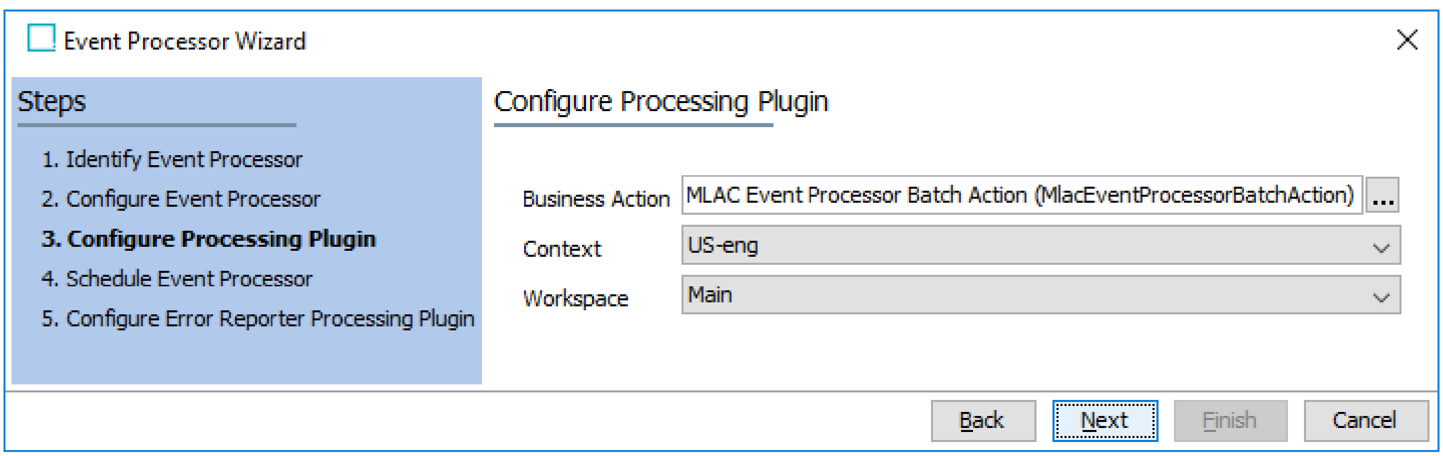
Use the following steps to create a batch event processor:

1. On the Configure Event Processor step:

- For the 'User running event processor plugin' parameter, select **a privileged system user**.
- For the 'Select Processor' parameter, select the **Execute Business Action for Event Batch** option.



2. On the Configure Processing Plugin step, select the **batch business action** created previously.



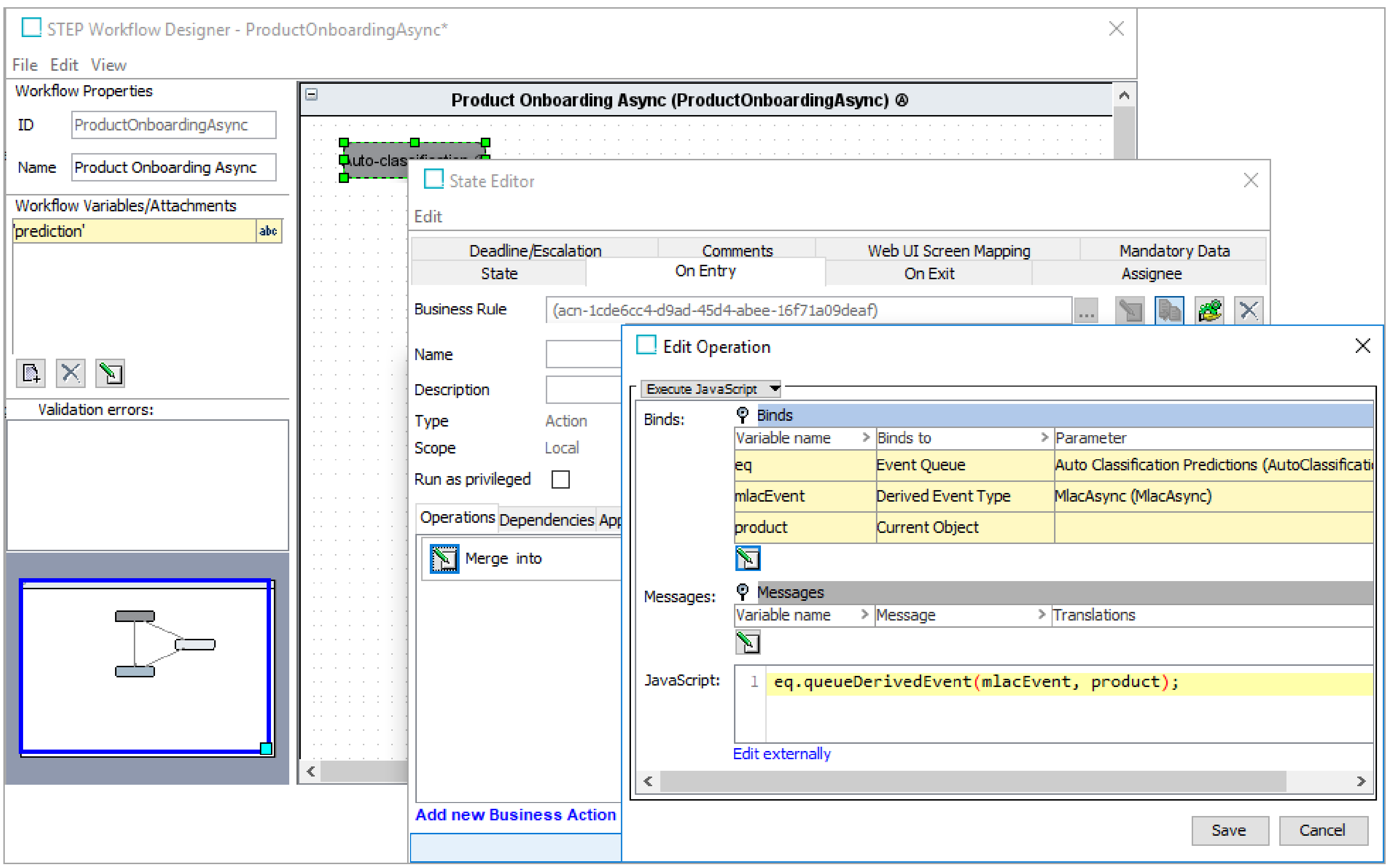
3. If a workflow has already been configured using the non-batched event processor, it can be used for the batch process with a single modification. The workflow action that queues events must be modified to queue events for the new event processor.

Create a Workflow Business Action

Create an action to generate the required event type and set it to be executed when a product enters the ‘Auto classification’ system state. The example shown in this section is for the non-batched EP and the batch EP would work similarly using he info in the prior sections.

1. Edit the onboarding workflow, select the 'Auto classification' state, right-click and choose the Edit State option.
2. Click the 'On Entry' tab click the **Add new Business Action** link.
3. On the Operations tab, click the edit button ( ) and select **Execute JavaScript** from the operations dropdown.
4. On the Edit Operation dialog:
   - In the Binds flipper, create three (3) binds:
     - **eq** = the event queue
     - **mlacEvent** = the derived event type
     - **product** = the current object
   - For the JavaScript parameter, add the following script:

```
eq.queueDerivedEvent(mlacEvent, product);
```



5. Click the **Save** button, and close the State Editor dialog.
6. On the STEP Workflow Designer, open the File menu and click the **Save and Exit** button.

Uploading Hierarchies

The MLAC service offers a REST endpoint for uploading entire hierarchies using a CSV file. Uploading data requires an OIEP for a single hierarchy with the required output format to publish the data.

To upload hierarchies, perform all steps in the following sections:

1. Prerequisites
2. Configure a Hierarchy OIEP
3. Publish Hierarchy Data to MLAC

Important: Repeat all of the steps in this topic for each hierarchy being uploaded.

Prerequisites

Complete the following setup to successfully configure the OIEP and publish data.

1. In the 'workarea' directory shared between the application servers, edit the sharedconfig.properties file. Add or modify the case-sensitive **RestDirectDeliveryURL** property to identify the hierarchy-specific URL for the MLAC service hierarchy upload resource. The URL is used to supply values for the REST Direct Delivery Method in the OIEP. The URL uses the format: https://app.stibosystems.com/ds/mlac/v1/hierarchies/[hierarchyId]

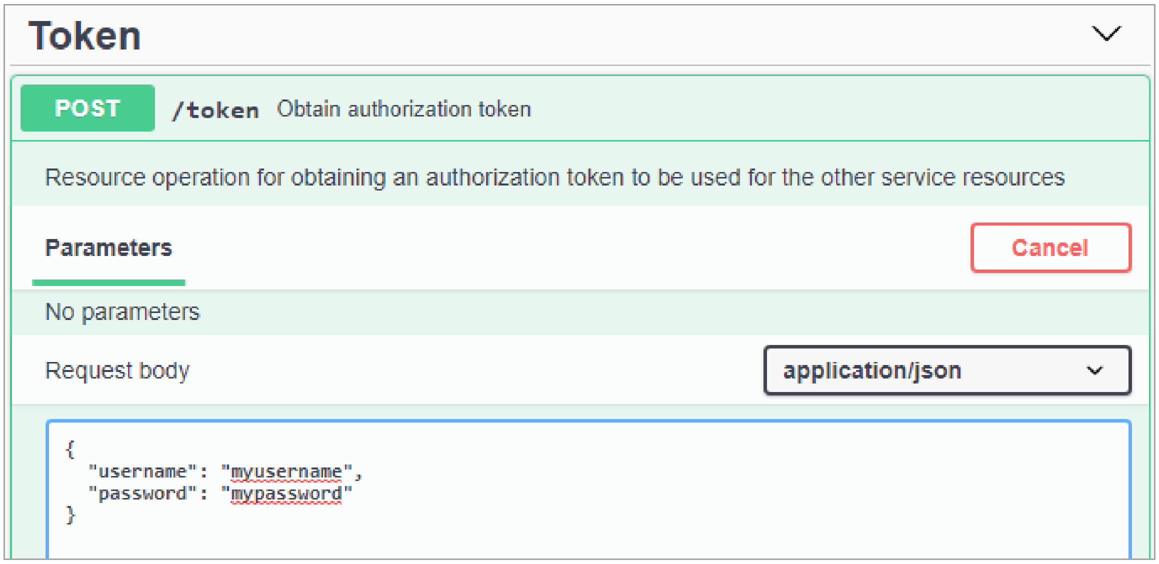
   Note: The hierarchy identifier ([hierarchyId]) is also used when uploading classified products.

   ```
   RestDirectDeliveryURL=1=https://app.stibosystems.com/ds/mlac/v1/hierarchies/primary
   ```

   In this example, the hierarchy ID of 'primary' has replaced the placeholder text '[hierarchyId]'.

2. Restart the server to implement the properties file changes.
3. Confirm your username and a password to get a token.
4. Use the service token endpoint (https://app.stibosystems.com/ds/mlac/v1/token) to obtain a token. See the Swagger UI available at https://app.stibosystems.com/ds/mlac/v1/. By default, tokens are valid for one (1) month.

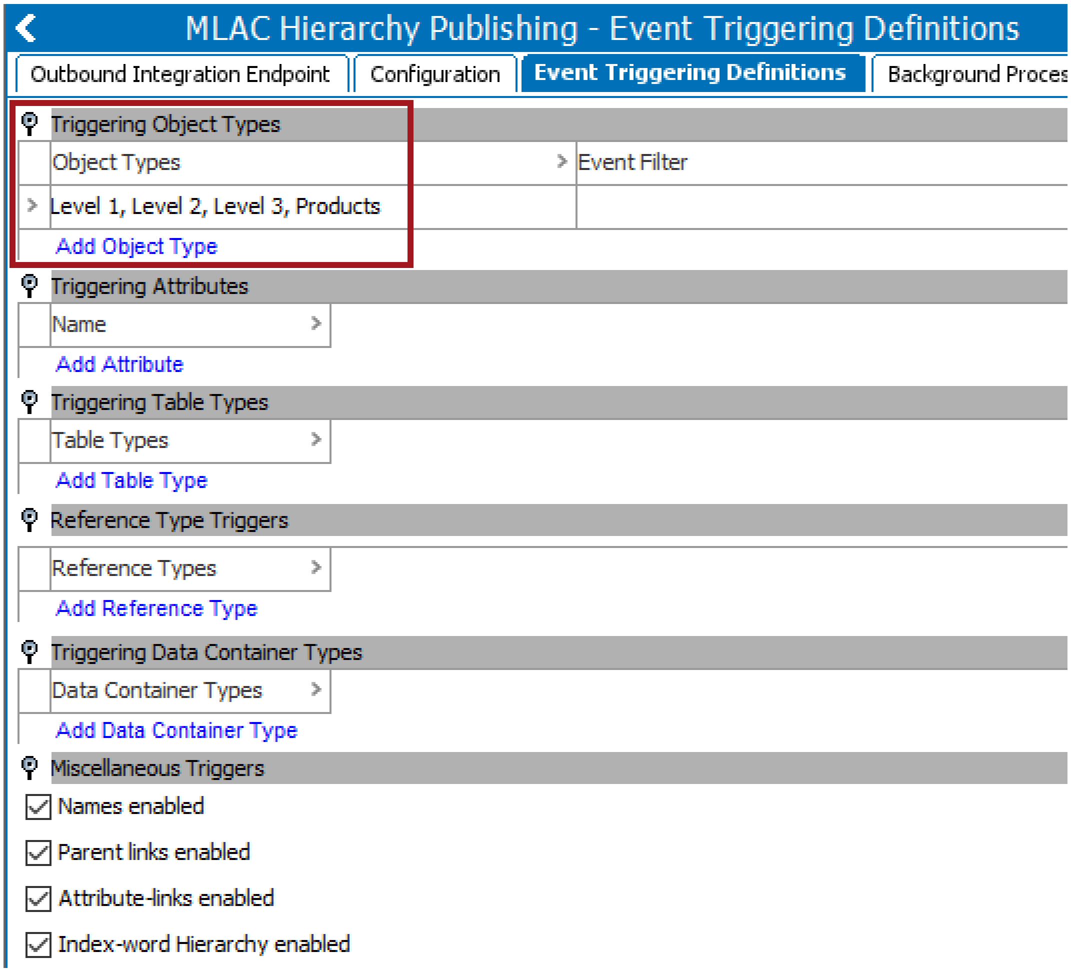
   Make note of the token for use in configuring the REST Direct Delivery Method Headers.



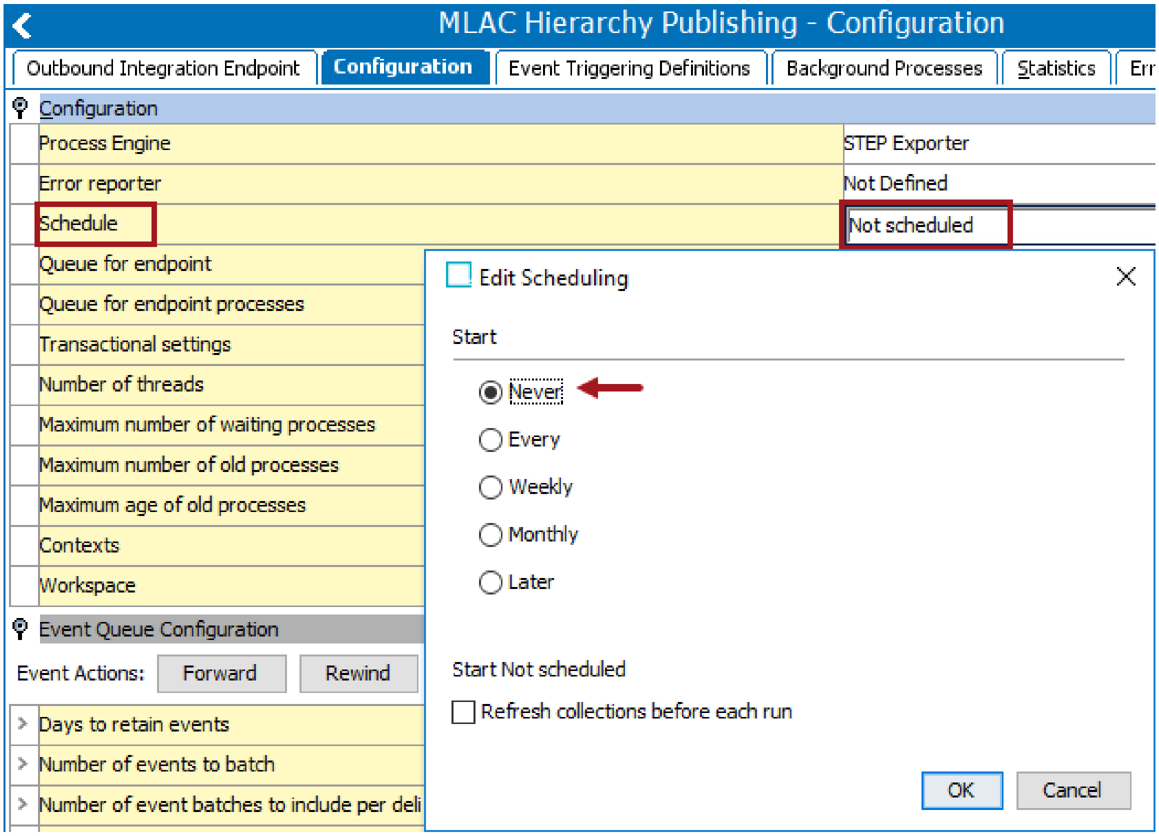
Configure a Hierarchy OIEP

Use the following steps to configure the necessary OIEP. For more general information on OIEPs, see the **Outbound Integration Endpoints** topic in the **Data Exchange** documentation.

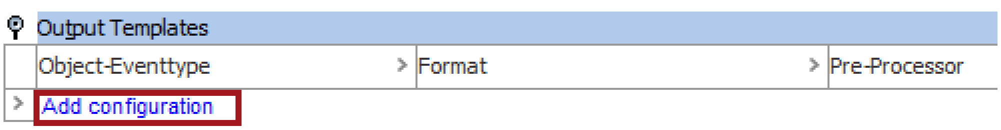
1. Create an event-based OIEP (as defined in the **Creating an Event-Based Outbound Integration Endpoint** topic of the **Data Exchange** documentation) with the following settings in the wizard:
   - On the 'Identify Endpoint' step, set the **User** parameter to a user who is privileged to see all hierarchy nodes.
   - On the 'Configure Endpoint' step, the **Contexts** parameter identifies the location of the classified products to be published.
   - On the 'Configure Endpoint' step, the **Workspace** parameter identifies the location of the classified products to be published. This is typically the 'Approved' workspace.
   - Click the **Finish** button to complete the wizard.
2. Open the OIEP's Event Triggering Definitions tab, open the Triggering Object Types flipper, and click the **Add Object Type** link to add the hierarchy object types. Leave the other settings as their default values.



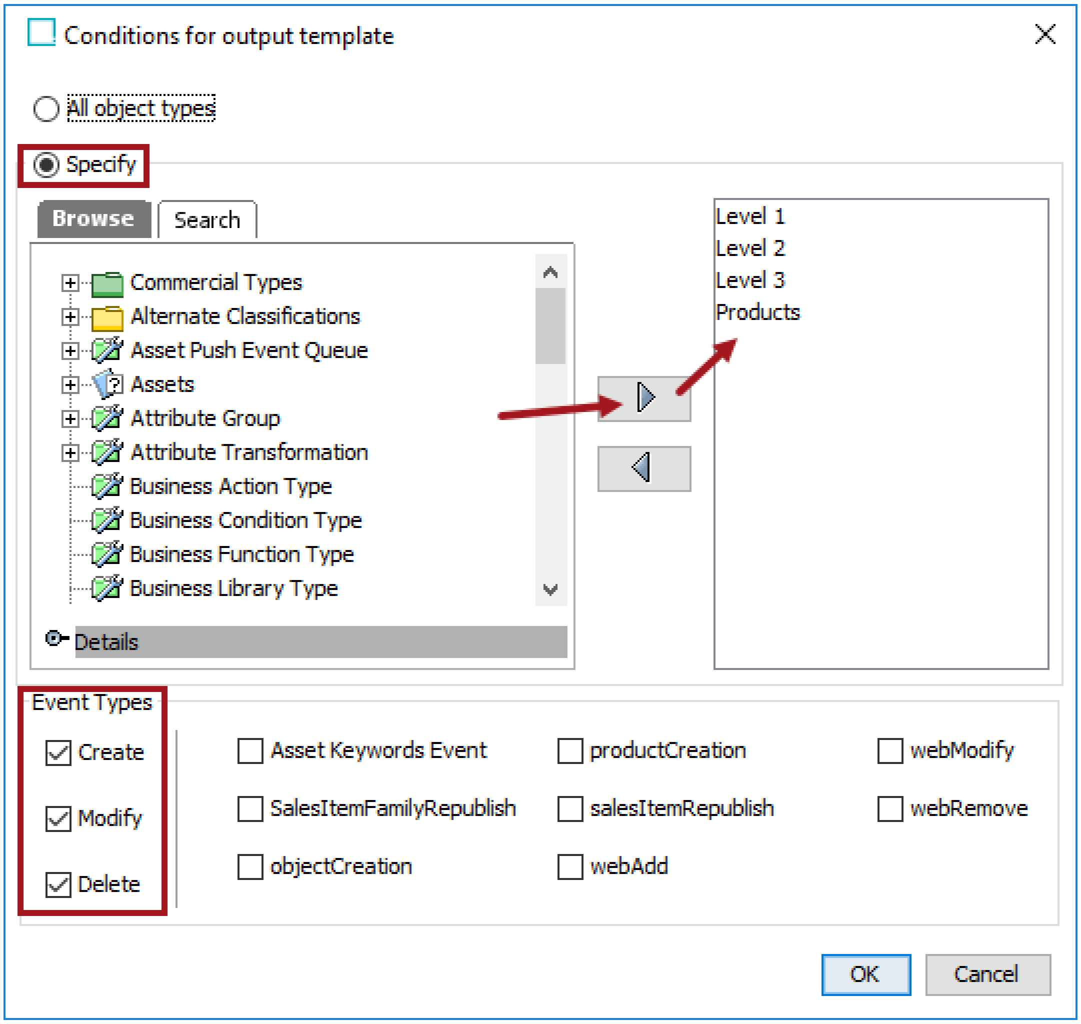
3. On the Configuration tab, open the Configuration flipper and set the 'Schedule' parameter to **Never** and click the **OK** button.



4. On the Configuration tab, open the Output Templates flipper, and set the following for the Object-Eventtype column:

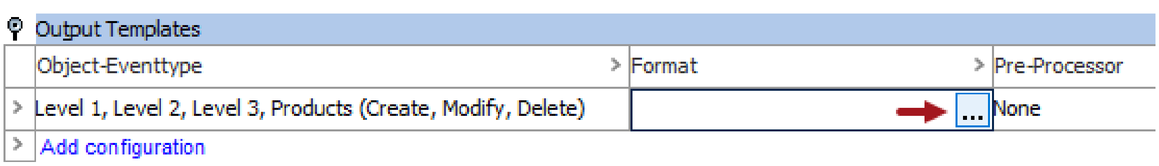


- Click the **Add configuration** link to display the 'Conditions for output template' dialog.
- Select the **Specify** radio button and add the hierarchy object types to be output.
- Select the **Create**, **Modify**, and **Delete** event types, and click the **OK** button.



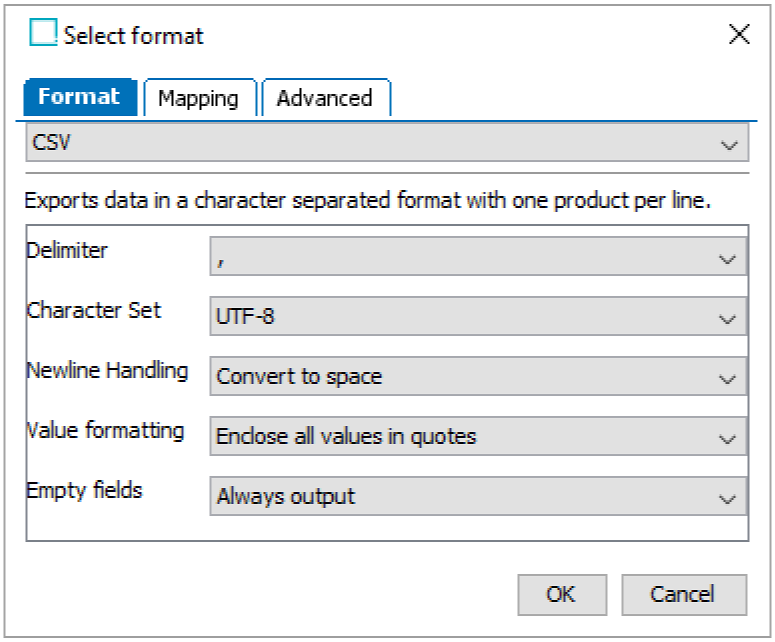
4. On the Output Templates flipper set the following for the Format column:

- Click the Format column cell to display an ellipsis button.
- Click the ellipsis button (...) to display the 'Select format' dialog.

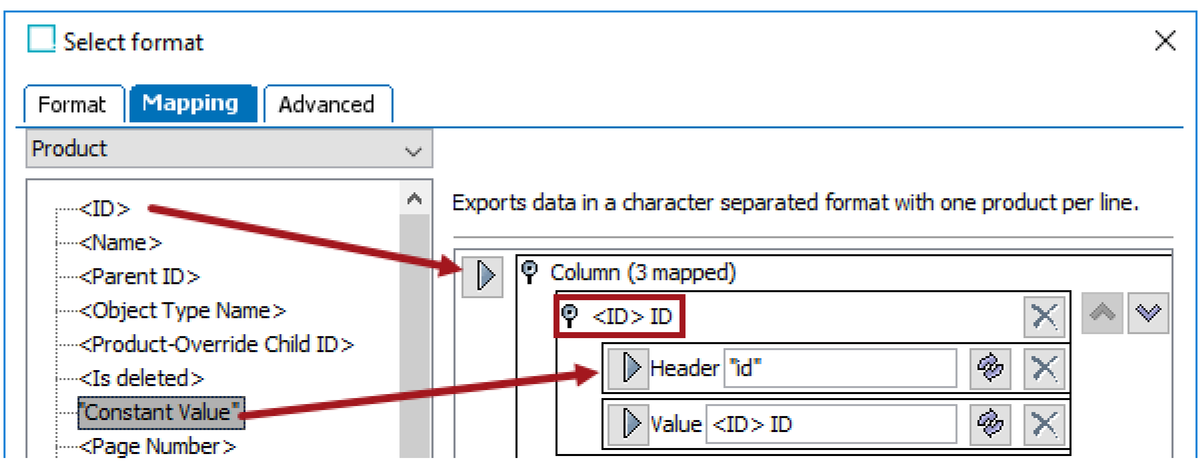


- On the 'Select format' dialog 'Format' tab (shown below), select **CSV** from the dropdown and set the following parameters:
  - For the 'Delimiter' parameter, select the comma (**,**).
  - For the 'Character Set' parameter, select **UTF-8**.
  - For the 'Newline Handling' parameter, select **Convert to space**.

- For the 'Value formatting' parameter, select **Enclose all values in quotes**.
- For the 'Empty fields' parameter, select **Always output**.

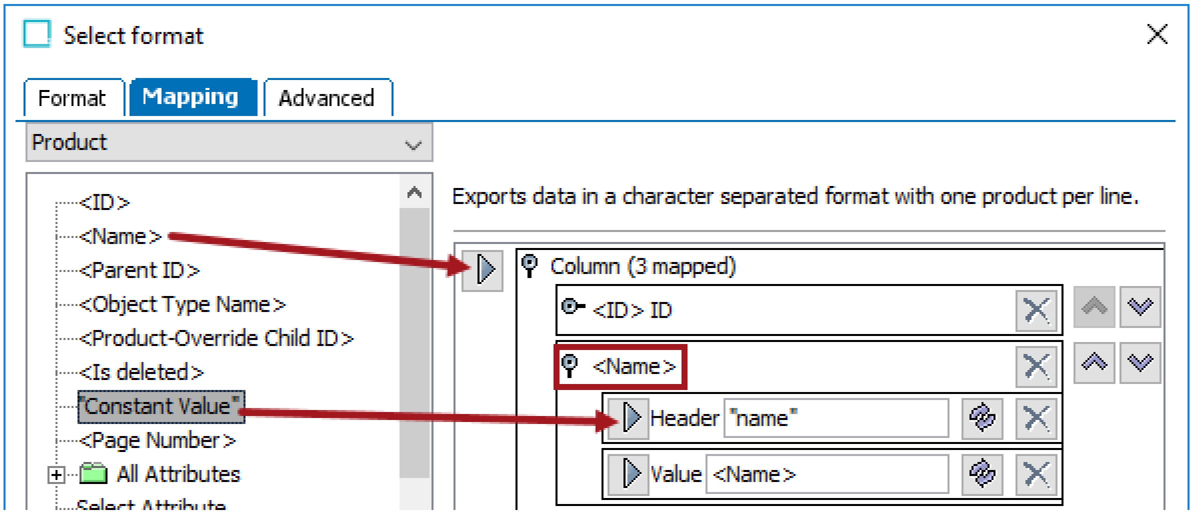


- On the 'Mapping' tab:
  - Map the **<ID>** data source.
  - Open the '<ID> ID' flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **id** (without quotes) and click the **Save** button.
  - Click the delete button () for the original header so the Header mapping matches the image below.

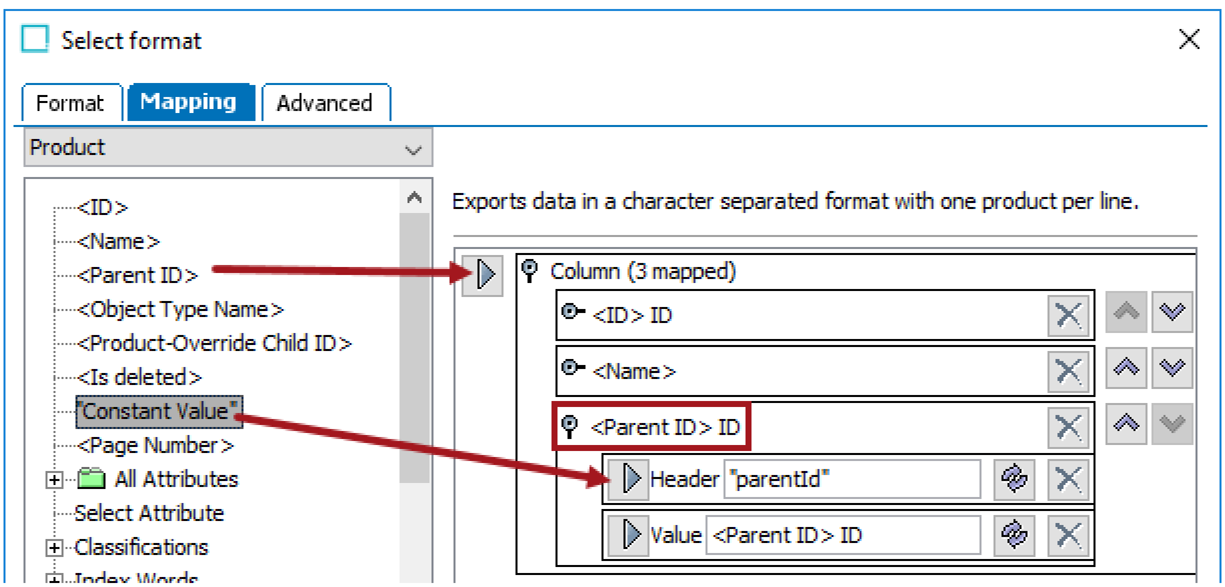


  - Map the **<Name>** data source.
  - Open the '<Name>' flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **name** (without quotes) and click the **Save** button.

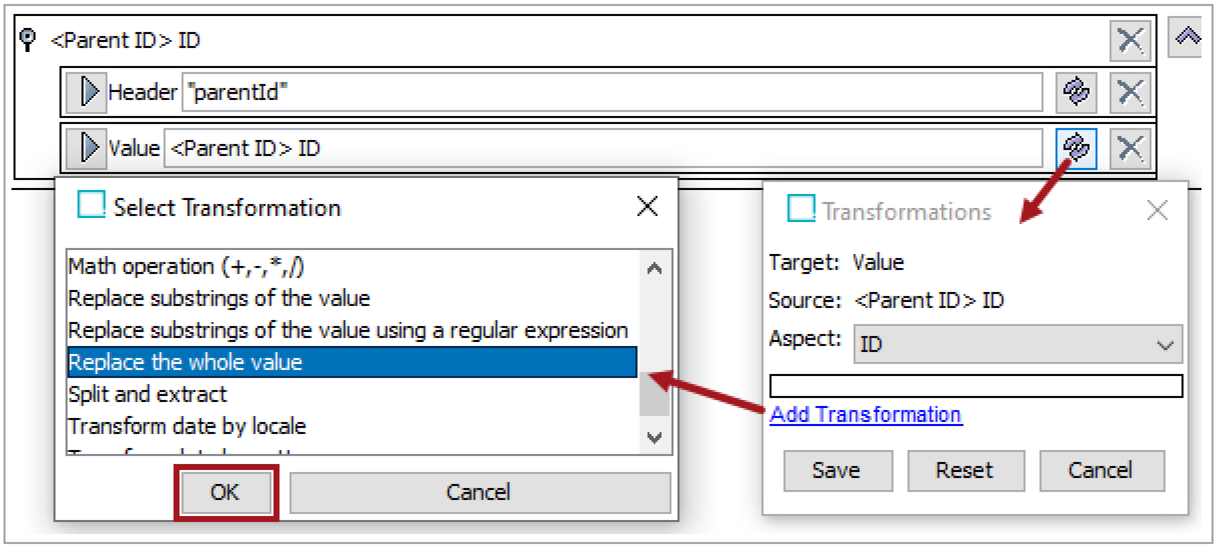
- Click the delete button () for the original header so the Header mapping matches the image below.



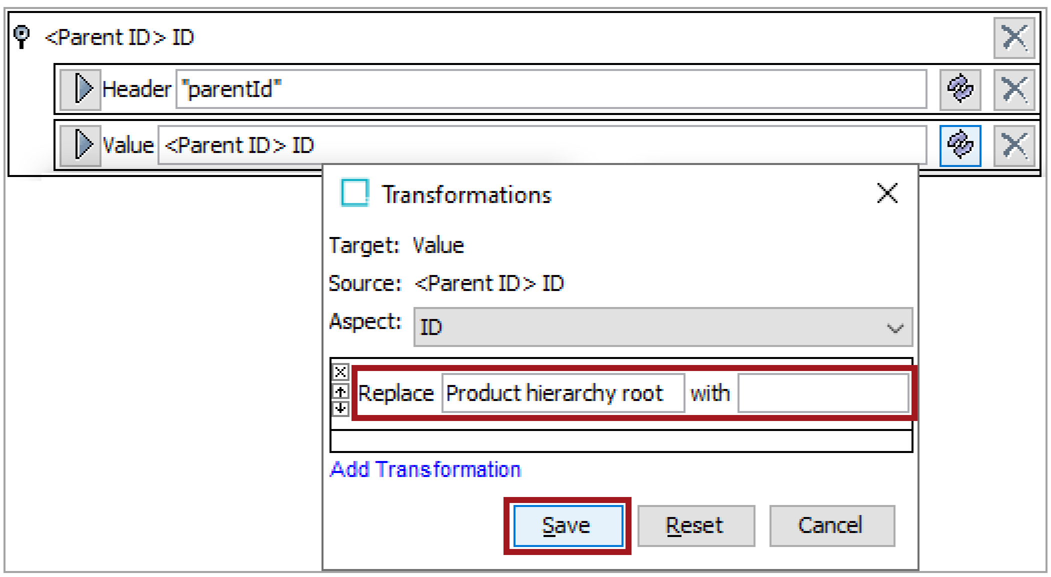
- Map the **<Parent ID>** data source.
- Open the '<Parent ID> ID' flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **parentId** (without quotes) and click the **Save** button.
- Click the delete button () for the original header so the Header mapping matches the image below.



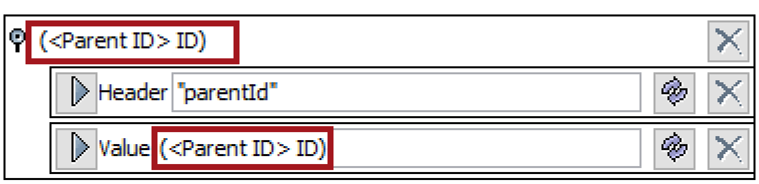
- On the value part of the mapped Parent ID, click the transform button ().
- On the Transformations dialog, click the **Add Transformation** link.
- On the Select Transformation dialog, select the **Replace whole value** option and click the **OK** button to update the Transformations dialog.



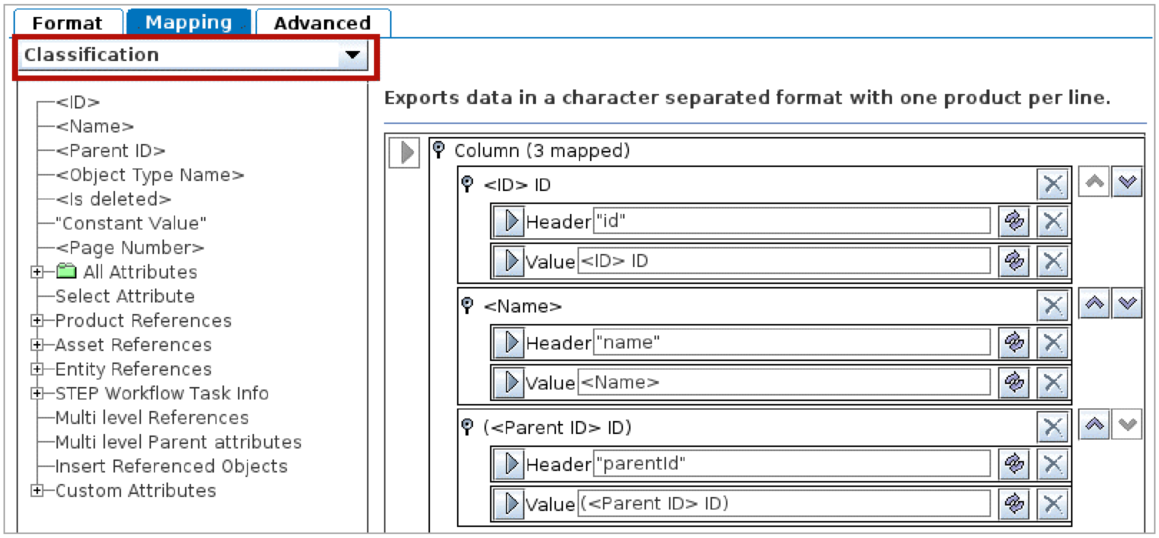
- On the updated Transformations dialog, in the 'Replace' parameter type **ID of the hierarchy root** and in the 'with' parameter, remove the value. This causes the 'parentId' for the hierarchy root node to be blank and the MLAC service sees this node as the root.
- Click the **Save** button.



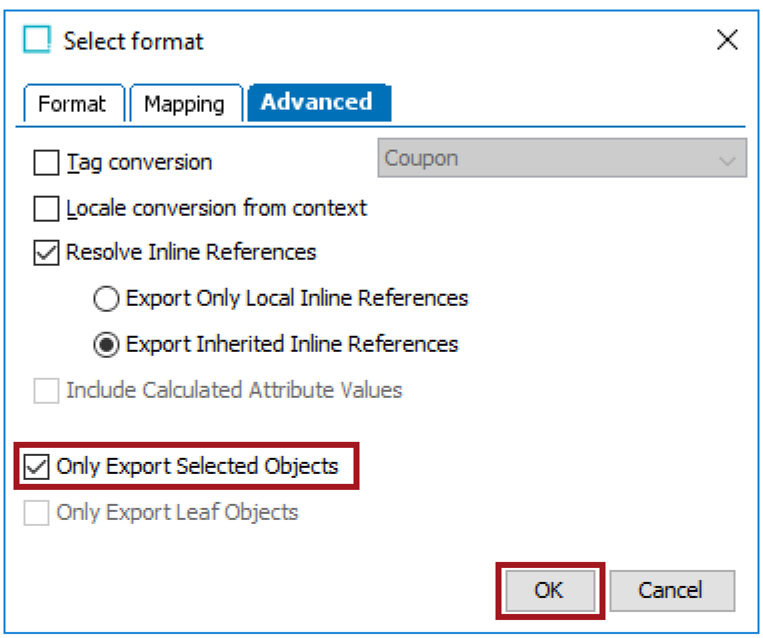
- Now the final '<Parent ID> ID' mapping is in parentheses, like (<Parent ID> ID), to indicate a transformation is applied for the Value element.



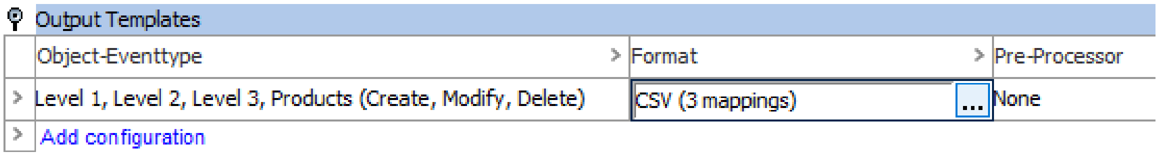
- In the upper left of the dialog, beneath the mapping tab, set the object type of the hierarchy to be exported (typically 'Classification').



- On the 'Advanced' tab:
  - Check the **Only Export Selected Objects** checkbox.
  - Click the **OK** button.



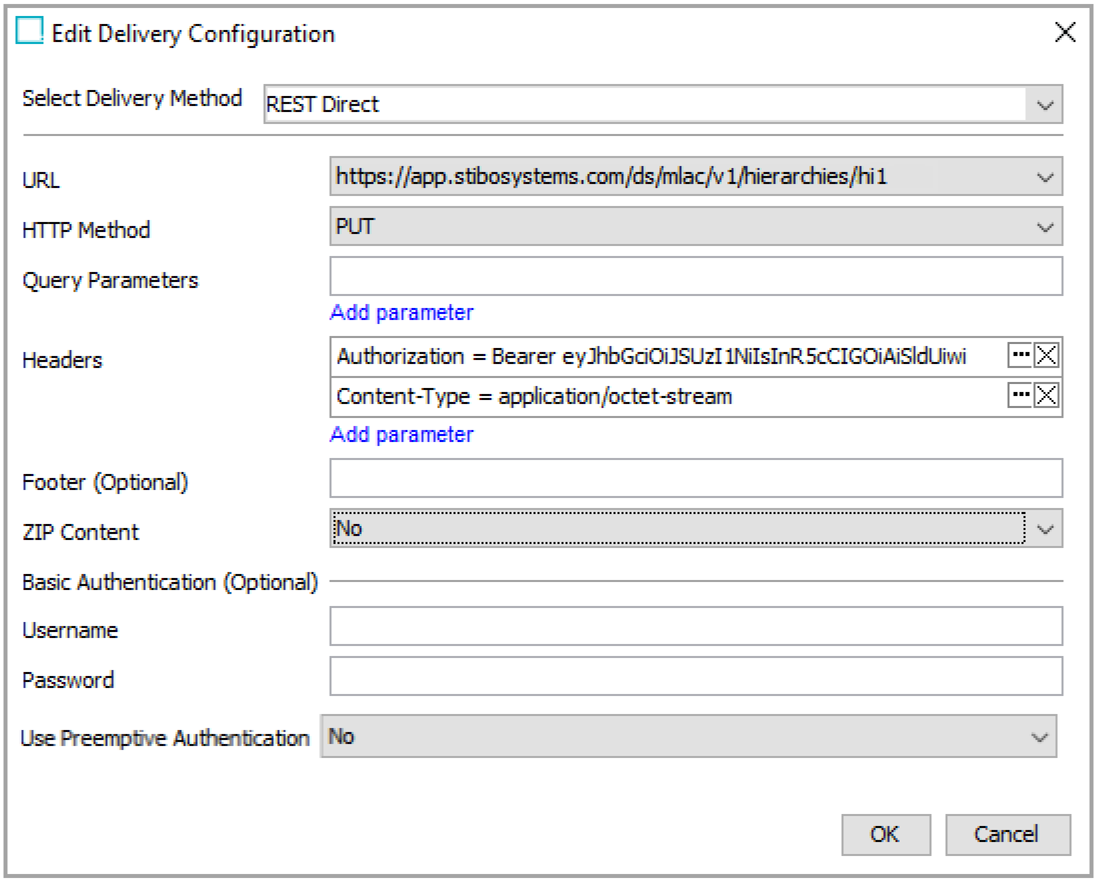
- The final state of the Output Templates flipper fields are displayed.



6. On the Delivery Method flipper, click the **Edit Delivery** link to open the 'Edit Delivery Configuration' dialog.

> **Note:** To validate the configuration, instead of setting the REST Direct method, first use the 'Copy to directory' or 'Email' delivery method and check the generated CSV file. See the **OIEP Delivery Methods** topic in the **Data Exchange** documentation for details.

- For the Select Delivery Method parameter, select **REST Direct** from the dropdown.
- For the URL parameter, select the **hierarchy-specific URL** in the dropdown.
- For the HTTP Method parameter, select **PUT**.
- For the Headers parameter, click the **Add parameter** link to display the 'Add Parameter' dialog. Set the Key parameter to **Content-Type**, the Value parameter to **application/octet-stream**, and click **OK**.
- For the Headers parameter, click the **Add parameter** link again to display the 'Add Parameter' dialog. Set the Key parameter to **Authorization**, the Value parameter to **the response from the token endpoint**, and click **OK**.
- For the ZIP Content parameter, select **No**.
- Leave other parameters blank.
- Click **OK** on the Edit Delivery Configuration dialog.



7. Repeat this setup for each hierarchy that will hold the newly classified products.

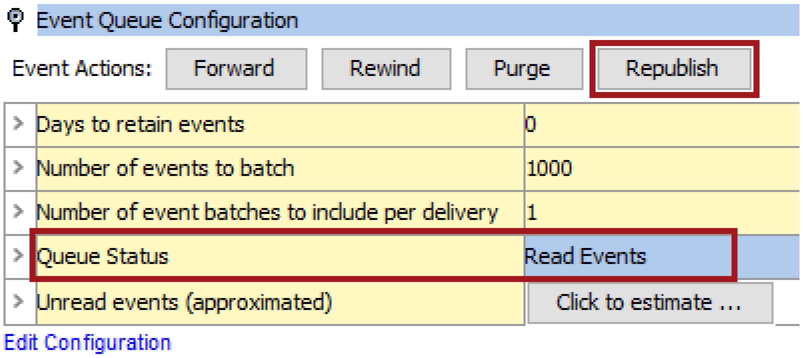
Publish Hierarchy Data to MLAC

Publishing hierarchy data to the MLAC service requires invoking an OIEP with unread events. For details about these steps, see the **Running an Outbound Integration Endpoint** topic and the **Event-Based OIEP Forward, Rewind, Purge, and Republish** topic, both in the Data Exchange documentation.

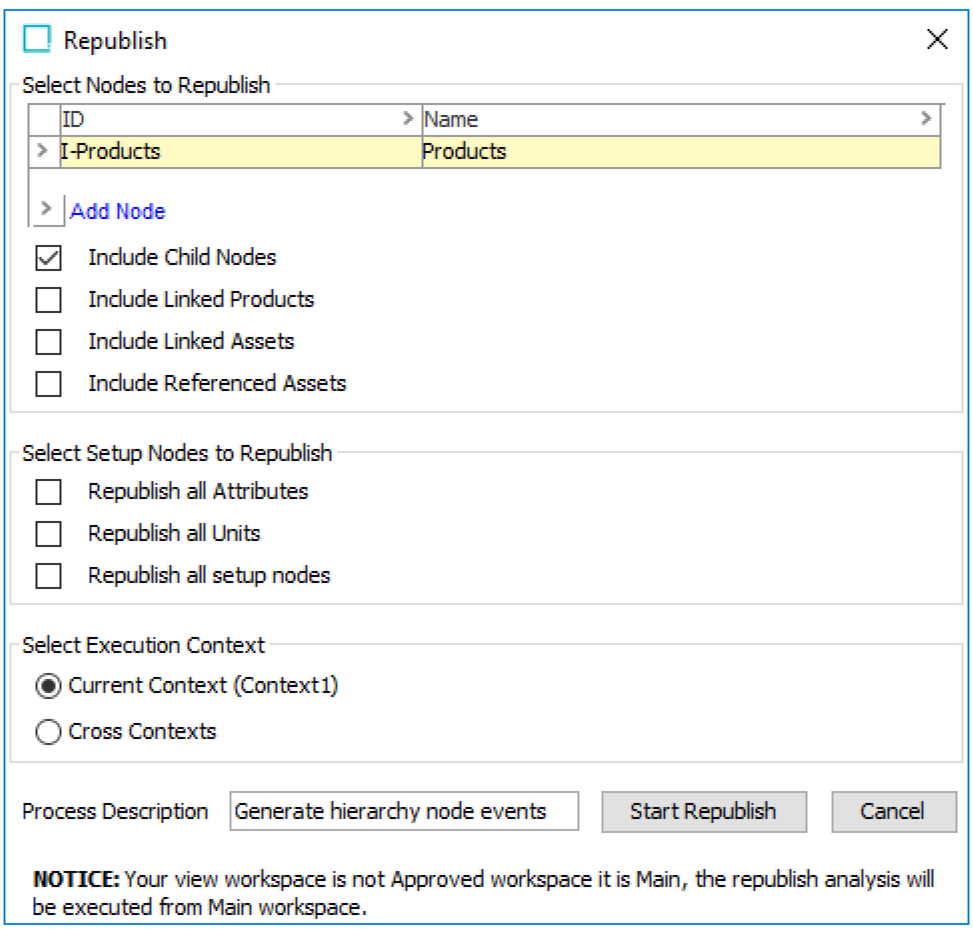
Important: Before publishing data to the MLAC service, it can be useful to first validate the configuration by using either the 'Copy to directory' or 'Email' delivery method and checking the generated CSV file. See the **OIEP Delivery Methods** topic in the **Data Exchange** documentation for details.

Use the following steps to publish hierarchy data to the MLAC service.

1. On System Setup, right-click the configured OIEP and click the **Enable Integration Endpoint** option.
2. On the OIEP's Configuration tab, open the Event Queue Configuration flipper and set the Queue Status parameter to **Read Events**.
3. Click the **Republish** button to display the Republish dialog.



4. On the Republish dialog:
   - Click the **Add Node** link and add the hierarchy root node.
   - Check the **Include Child Nodes** checkbox.
   - For the Process Description parameter, type a description. 'Generate hierarchy node events' is the description in the following image.
   - Click the **Start Republish** button.



5. Verify that the republish background process finishes successfully.
6. Right-click the OIEP and click the **Invoke** option to manually publish the hierarchy data.
7. Right-click the OIEP and click the **Disable Integration Endpoint** option.
8. On the OIEP's Configuration tab, open the Event Queue Configuration flipper and set the Queue Status parameter to **Discard Events**.

Uploading Classified Products

The MLAC service offers a REST Direct OIEP for uploading sets of products that are already classified, or 'labeled,' using a CSV file. Uploading data requires an OIEP for a single hierarchy with the required output format to publish the data.

To upload hierarchies, perform all steps in the following sections:

1. Prerequisites
2. Configure a Classified Products OIEP
3. Publish Classified Products Data to MLAC

File Requirements

As outlined in the Mapping steps below, the REST Direct OIEP requires a CSV file with the following headers / columns in the specified order.

- 'id' - ID of the classified product. Mandatory.
- 'description' - The description of the product. Mandatory. This must be a 'raw' source description not modified after the product was classified. Values for multiple description attributes can be supplied either by creating a calculated attribute that concatenates the values or by mapping multiple attributes to this column. More about mappings is defined below.
- 'sourceId' - Source identification, for instance a vendor ID. Optional. A column with this header must be present in the generated file even when data is not exported. Include data if the source information is available and if it has significance for the classification of the product.
- '[Hierarchy ID 1..n]' - ID of the classification (hierarchy node ID) in the hierarchy identified via the header value. Header values must correspond with the agreed upon IDs for the hierarchies.

For the primary product hierarchy case, if there are intermediate hierarchy levels between the product and the classification / hierarchy node (for example, a product family level), a calculated attribute can be used to output the appropriate hierarchy node ID for this column.

> **Important:** The example scripts should not be used as-is without thorough testing, including updating the script to match object and link types that exist on your system. JavaScript variable names are case-sensitive.

The example below shows a function that will output either the ID of the parent or ID of the grandparent, based on whether the parent is of a specific object type.

```
{
  i:=path(),
  parent:=listitem(i,listlen(i)-1),
  grandparent:=listitem(i,listlen(i)-2),
  parentobjecttype:=iterate(parent, 'stepobjecttype()')
} if (exact(parentobjecttype, "Sales Item Family"), iterate(grandparent, 'stepid()'),
iterate(parent, 'stepid()'))
```

In this example, 'stepobjecttype()' returns the object type name and not the ID. Thus, on systems with dimension dependent object type names, consider the object type for the Context of the export when making the comparison on the name.

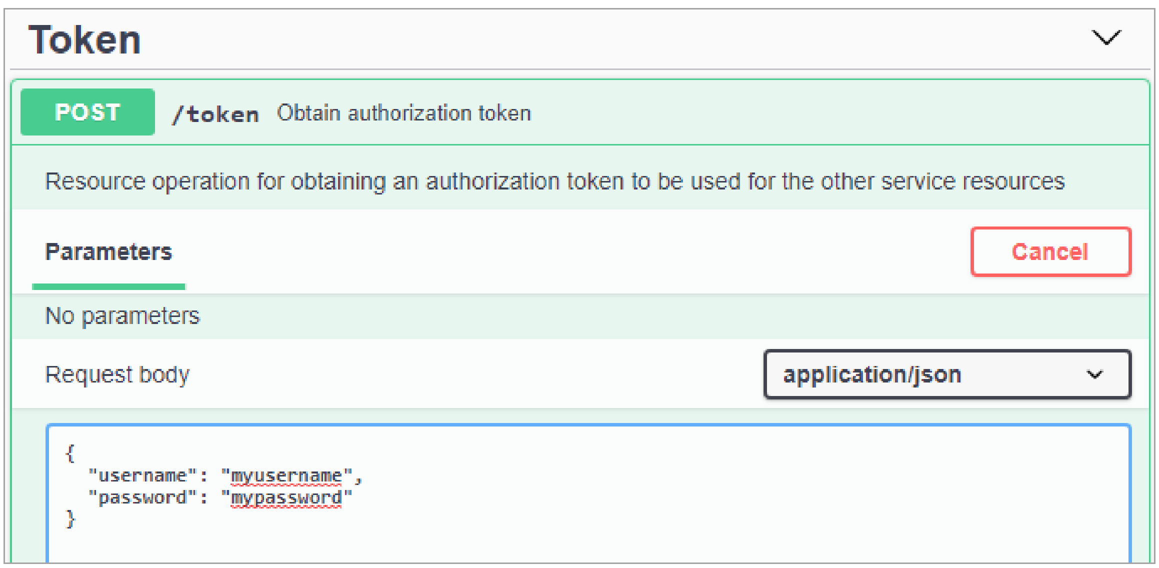
Prerequisites

Complete the following setup to successfully configure the OIEP and publish data.

1. In the 'workarea' directory shared between the application servers, edit the sharedconfig.properties file. Add the case-sensitive **RestDirectDeliveryURL** property to identify the hierarchy-specific URL for the MLAC service training products upload resource. The URL is used to supply values for the REST Direct Delivery Method in the OIEP.

   ```
   RestDirectDeliveryURL=1= https://app.stibosystems.com/ds/mlac/v1/products
   ```
2. Restart the server to implement the properties file changes.
3. If you do not already have a token, confirm your username and a password to get a token.
4. If you do not already have a token, use the service token endpoint (https://app.stibosystems.com/ds/mlac/v1/token) to obtain a token. See the Swagger UI available at https://app.stibosystems.com/ds/mlac/v1/. By default, tokens are valid for one (1) month.

   Make note of the token for use in configuring the REST Direct Delivery Method Headers.



Configure a Classified Products OIEP

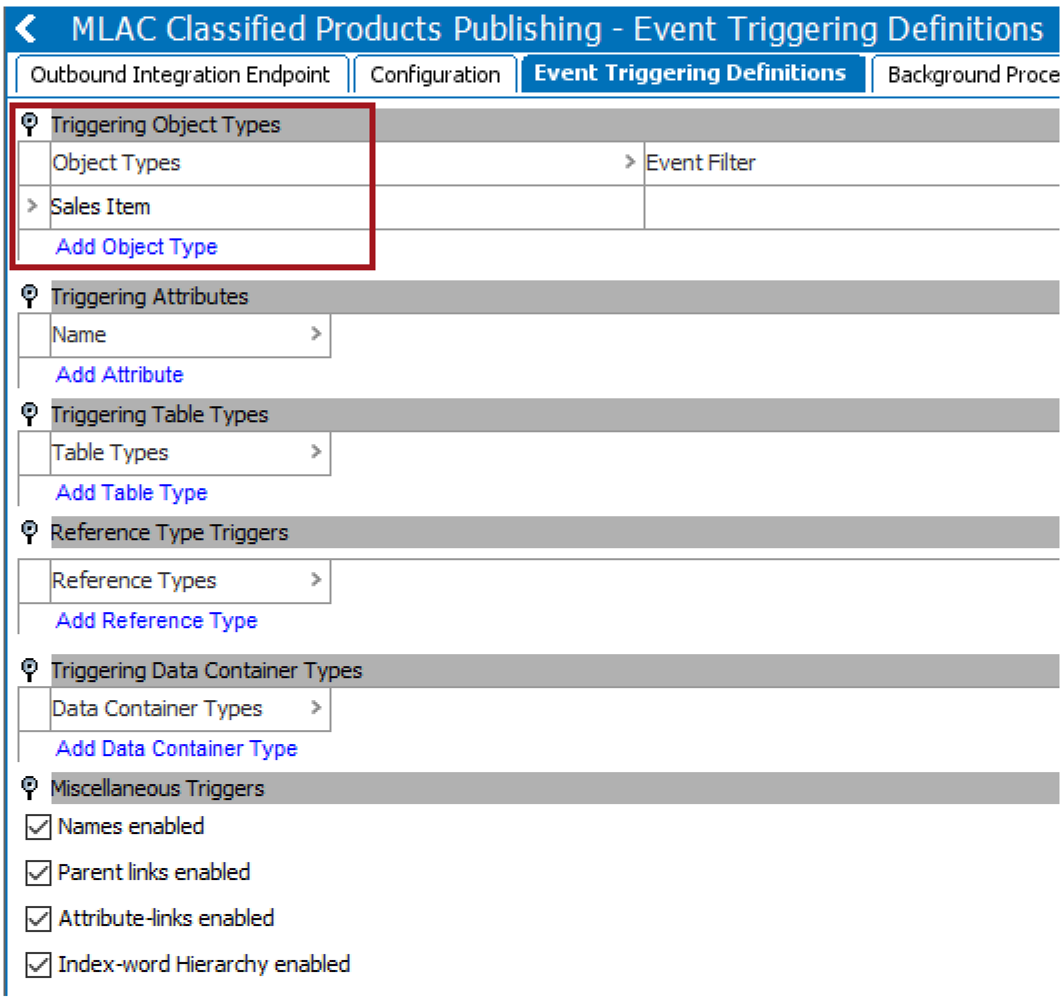
Use the following steps to configure the necessary OIEP. For more general information on OIEPs, see the **Outbound Integration Endpoints** topic in the **Data Exchange** documentation.

1. Create an event-based OIEP (as defined in the **Creating an Event-Based Outbound Integration Endpoint** topic of the **Data Exchange** documentation) with the following settings in the wizard:

- On the 'Identify Endpoint' step, set the **User** parameter to a user who is privileged to see all classified products and hierarchy relations.
- On the 'Configure Endpoint' step, the **Contexts** parameter identifies the location of the classified products to be published.
- On the 'Configure Endpoint' step, the **Workspace** parameter identifies the location of the classified products to be published. This is typically the 'Approved' workspace.
- Click the **Finish** button to complete the wizard.

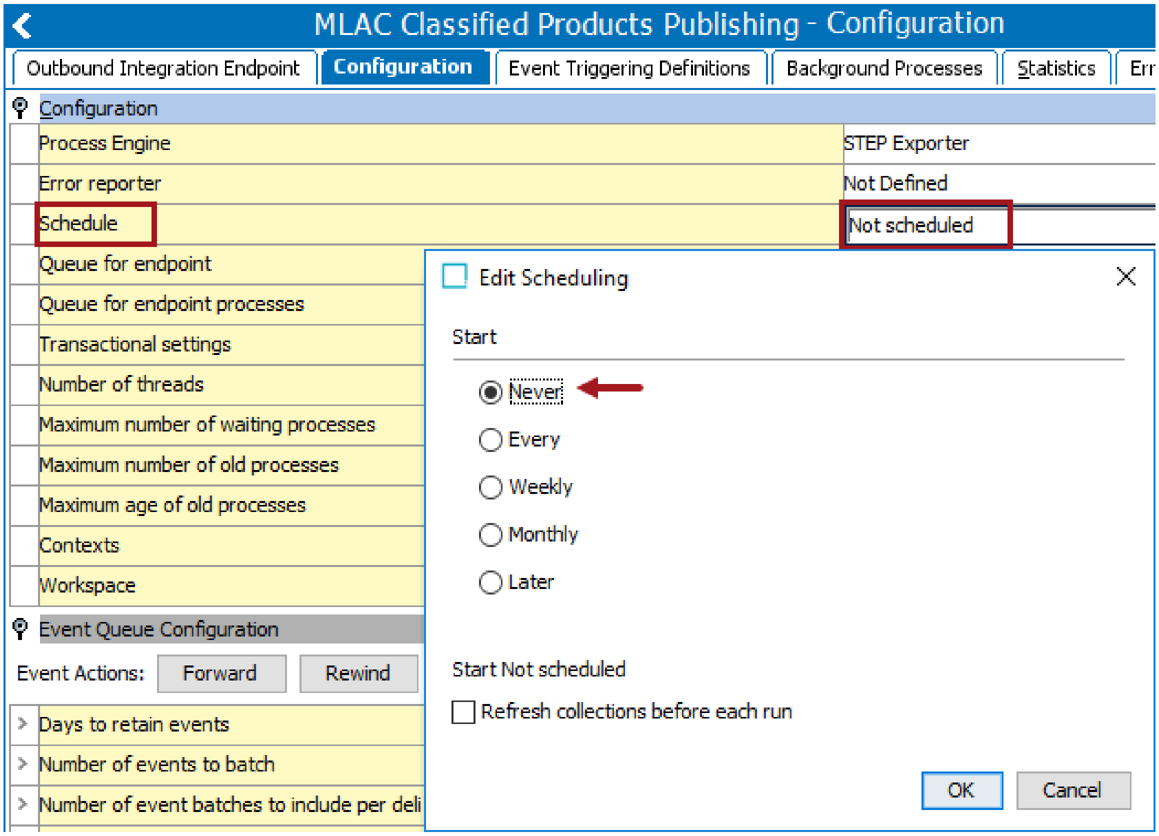
2. Open the OIEP's Event Triggering Definitions tab, open the Triggering Object Types flipper, and click the **Add Object Type** link to add the product object type(s).

- For manual publishing leave the other settings as their default values.
- For automatic publishing as events are generated for the products, select the relevant triggering types (attributes, classification product link types, etc.) to monitor for changes that should generate an event.

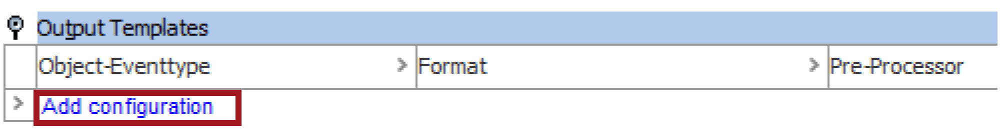


3. On the Configuration tab, open the Configuration flipper and open the 'Schedule' parameter:

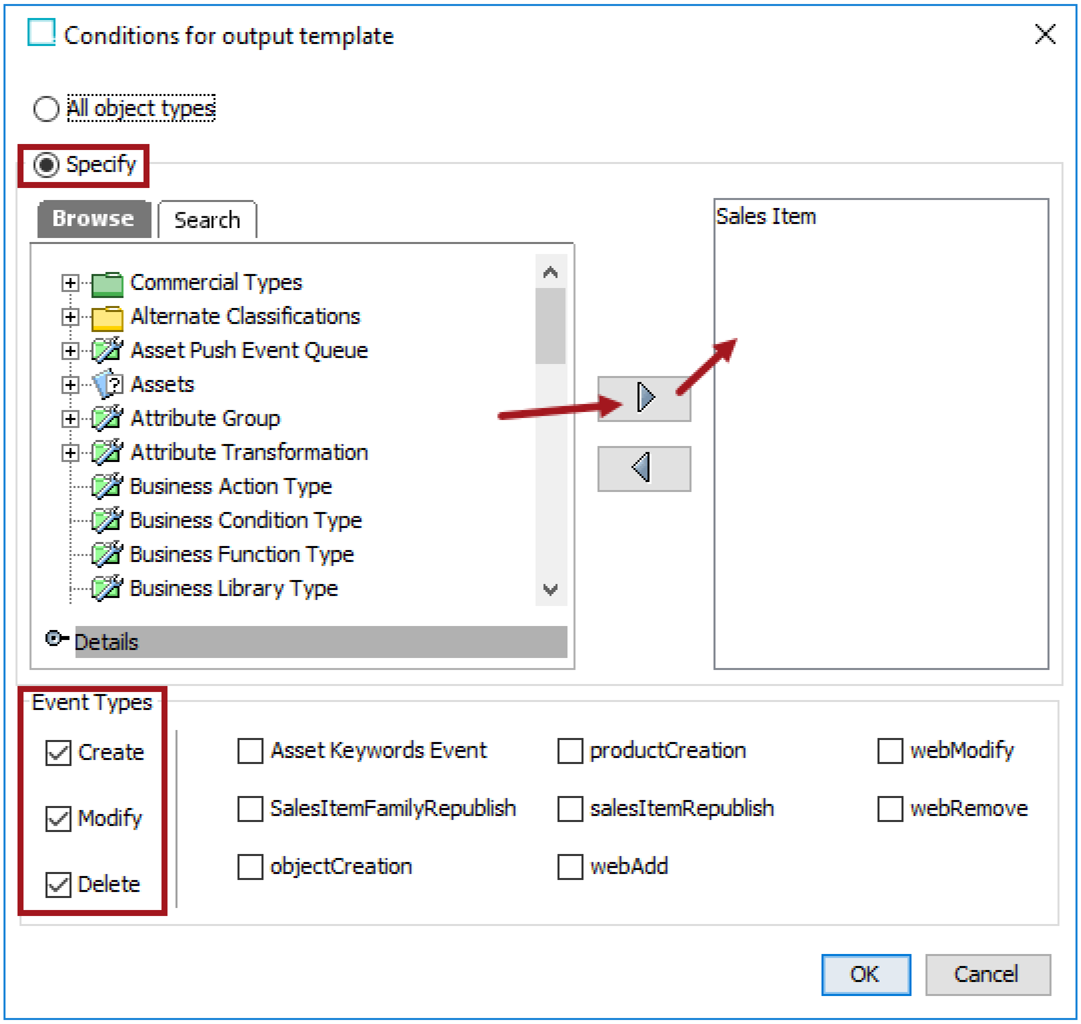
- For manual publishing, set to **Never** and click the **OK** button.
- For automatic publishing, set a schedule for publishing labeled products' data upon approval.



4. On the Configuration tab, open the Output Templates flipper, and set the following for the Object-Eventtype column:

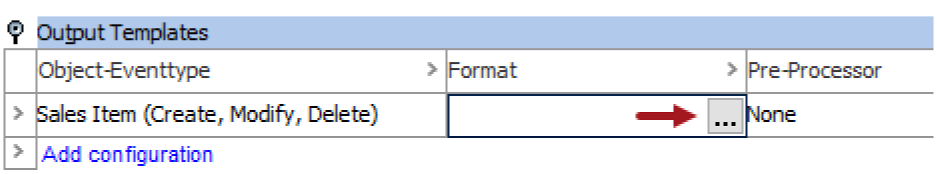


- Click the **Add configuration** link to display the 'Conditions for output template' dialog.
- Select the **Specify** radio button and add the classified product object types to be output.
- Select the **Create**, **Modify**, and **Delete** event types, and click the **OK** button.



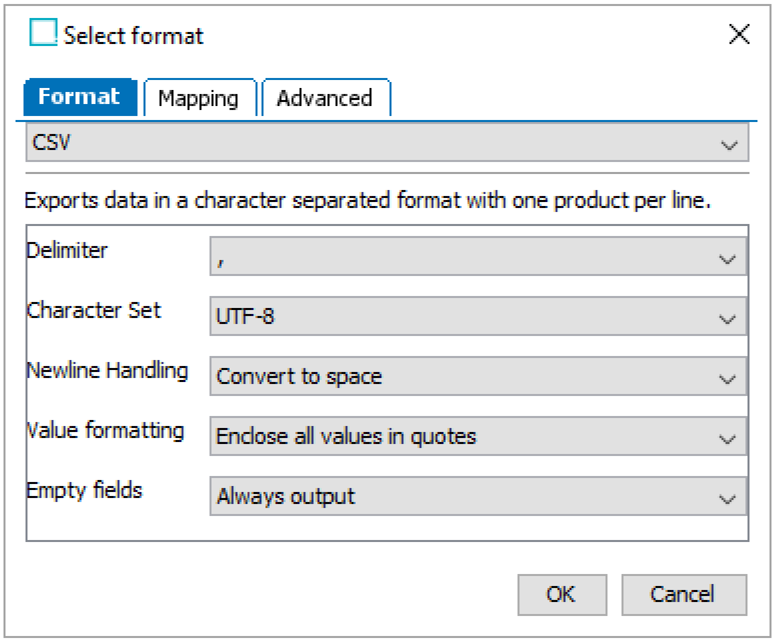
4. On the Output Templates flipper set the following for the Format column:

- Click the Format column cell to display an ellipsis button.
- Click the ellipsis button (...) to display the 'Select format' dialog.



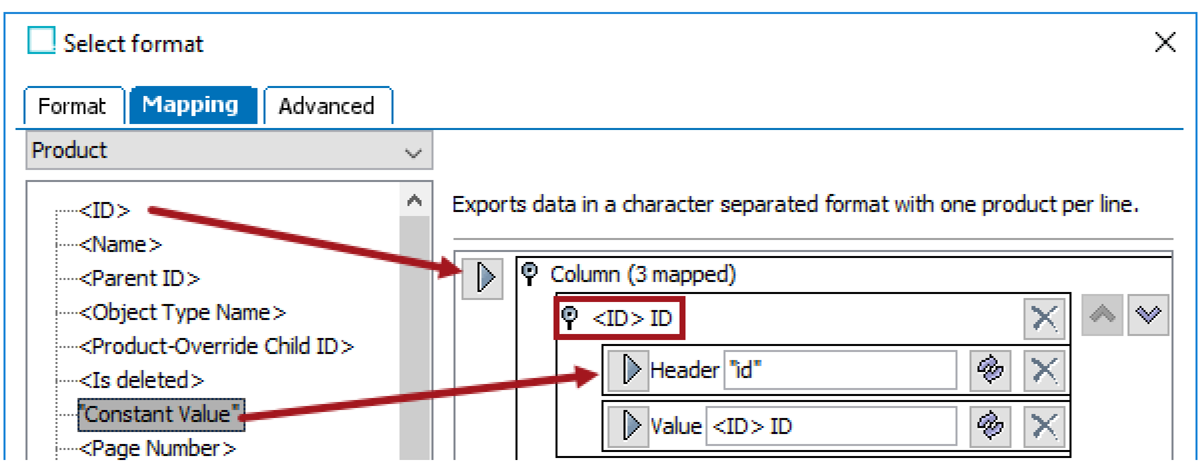
- On the 'Select format' dialog 'Format' tab (shown below), select **CSV** from the dropdown and set the following parameters:
  - For the 'Delimiter' parameter, select the comma (**,**).
  - For the 'Character Set' parameter, select **UTF-8**.
  - For the 'Newline Handling' parameter, select **Convert to space**.

- For the 'Value formatting' parameter, select **Enclose all values in quotes**.
- For the 'Empty fields' parameter, select **Always output**.



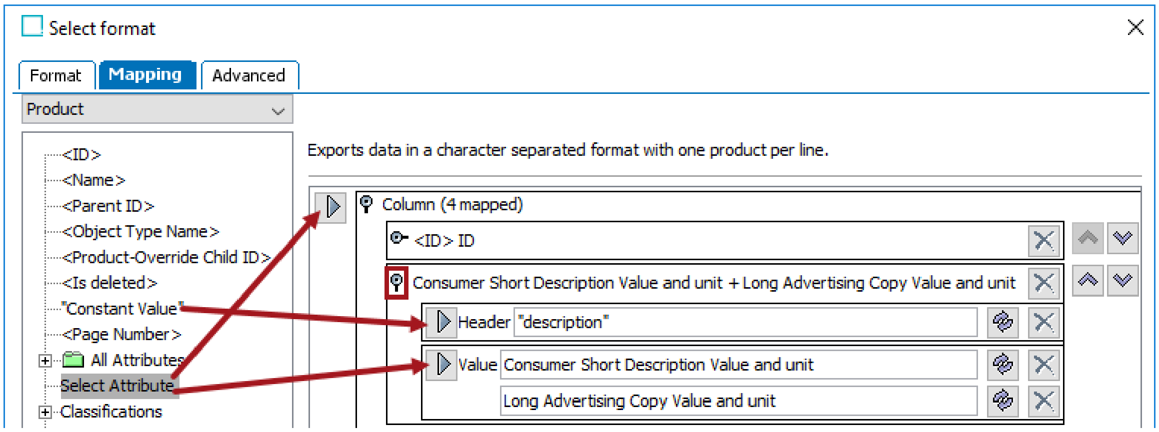
Note: See the **File Requirements** section above for definitions of the mapping elements below.

- On the 'Mapping' tab:
  - Map the **<ID>** data source.
  - Open the '<ID> ID' flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **id** (without quotes) and click the **Save** button.
  - Click the delete button ( ) for the original header so the Header mapping matches the image below.



  - Map the **Select Attribute** data source and choose the attribute that holds the description.

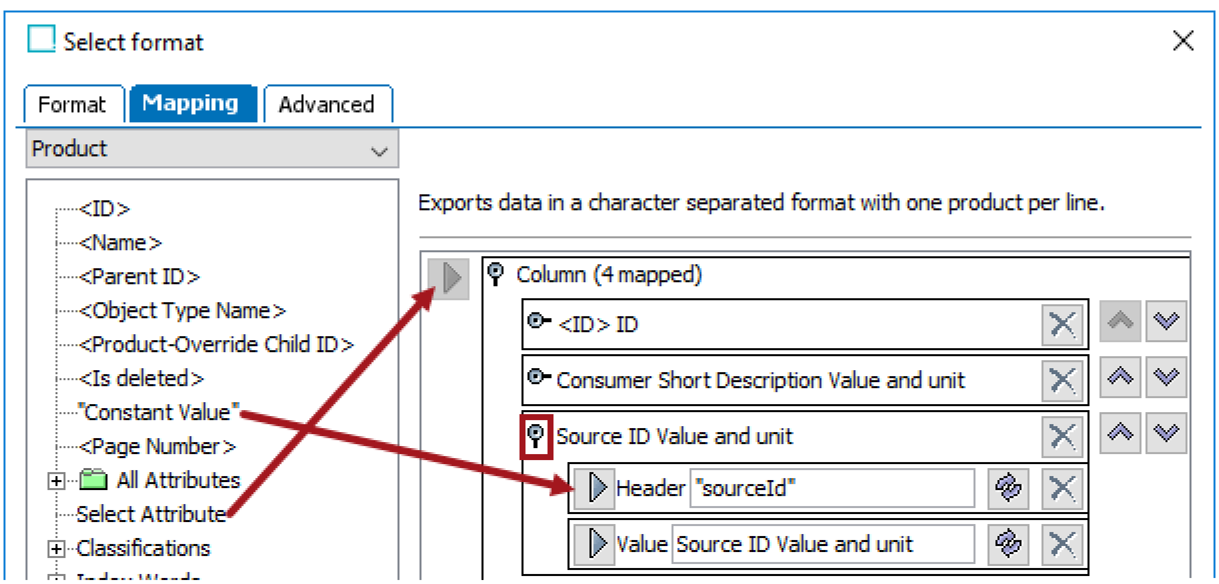
- Open the description attribute flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **description** (without quotes) and click the **Save** button.
- Click the delete button ( ) for the original header so the Header mapping matches the image below.
- If multiple attributes hold description information, map them as additional value elements as have been mapped below for the 'Consumer Short Description' and the 'Long Advertising Copy Value' attributes.



- Choose a method to handle the source ID with significance for the classification:

  If a source ID that has significance for the classification is available in an attribute, map the **Select Attribute** data source and choose that attribute.

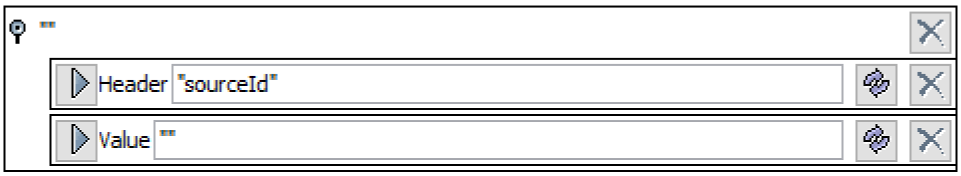
  - Open the source ID attribute flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **sourceId** (without quotes) and click the **Save** button.
  - Click the delete button ( ) for the original header so the Header mapping matches the image below.



If a source ID that has significance for the classification is not available in an attribute, map the **Constant Value** data source.

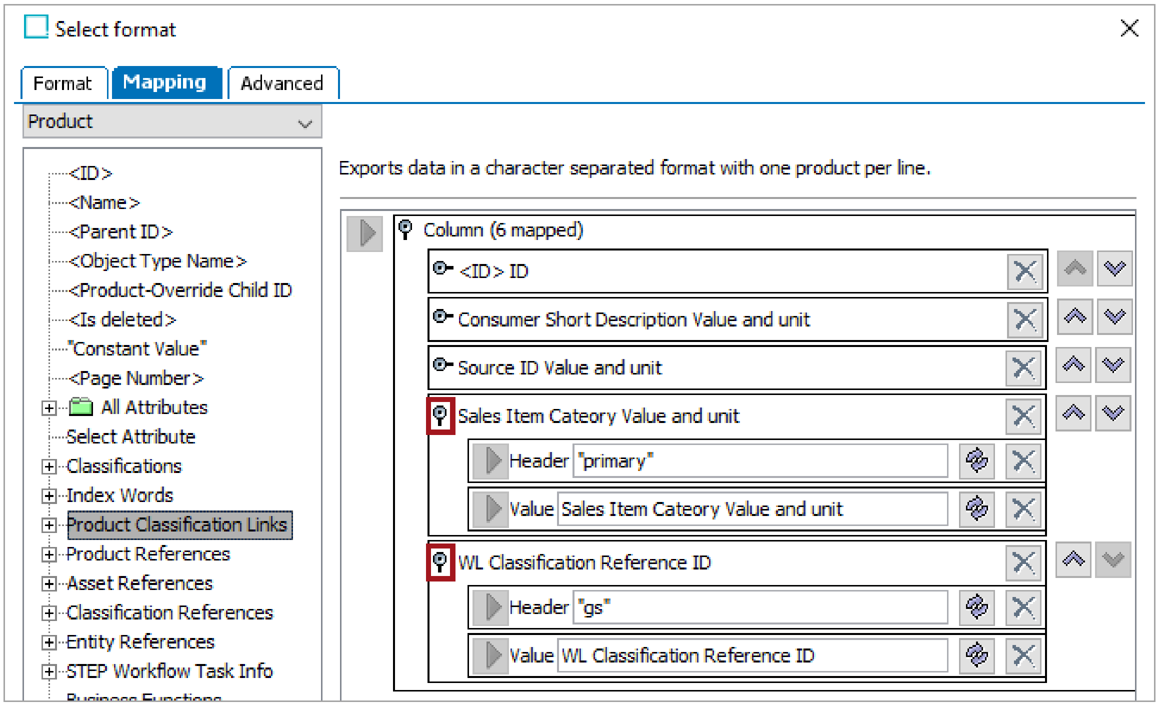
- On the 'Enter Value' dialog, leave the 'Enter constant value' parameter blank and click the **Save** button.
- Open the 'Constant' flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **sourceId** (without quotes) and click the **Save** button.
- Click the delete button () for the original header so the Header mapping matches the image below.
- For the Value row, click the transform button (), delete the 'Constant' text from the Source parameter, and click the **Save** button.

  The final Header and Value mapping should match the image below.

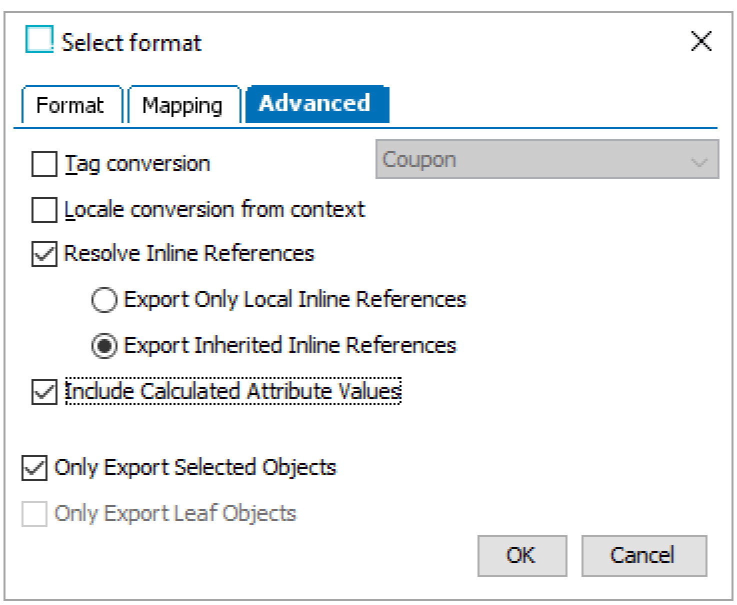


- Choose a method to map a column for each hierarchy; in each instance replace the header with the hierarchy ID:
  - For a primary product hierarchy where there are no intermediate levels between the product and the hierarchy node, use <Parent ID> for the value.
  - For intermediate level hierarchies, map a calculated attribute that holds the appropriate hierarchy node ID.
  - For classification hierarchies, use 'Product Classification Links' data source to map the appropriate link type.

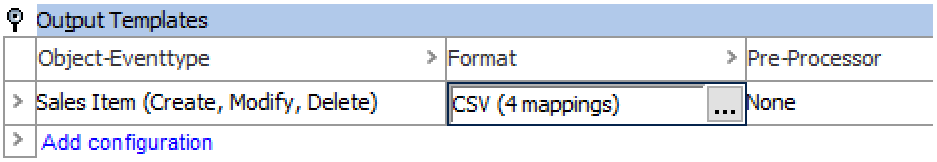
  For example, in the screenshot below, two hierarchy columns have been mapped: one by mapping a calculated attribute, and one by mapping a link type.



- On the 'Advanced' tab:
  - Check the **Only Export Selected Objects** checkbox.
  - Check the **Include Calculated Attribute Values** checkbox. A mapped calculated attribute is required for this box to become active (able to be checked).
  - Click the **OK** button.



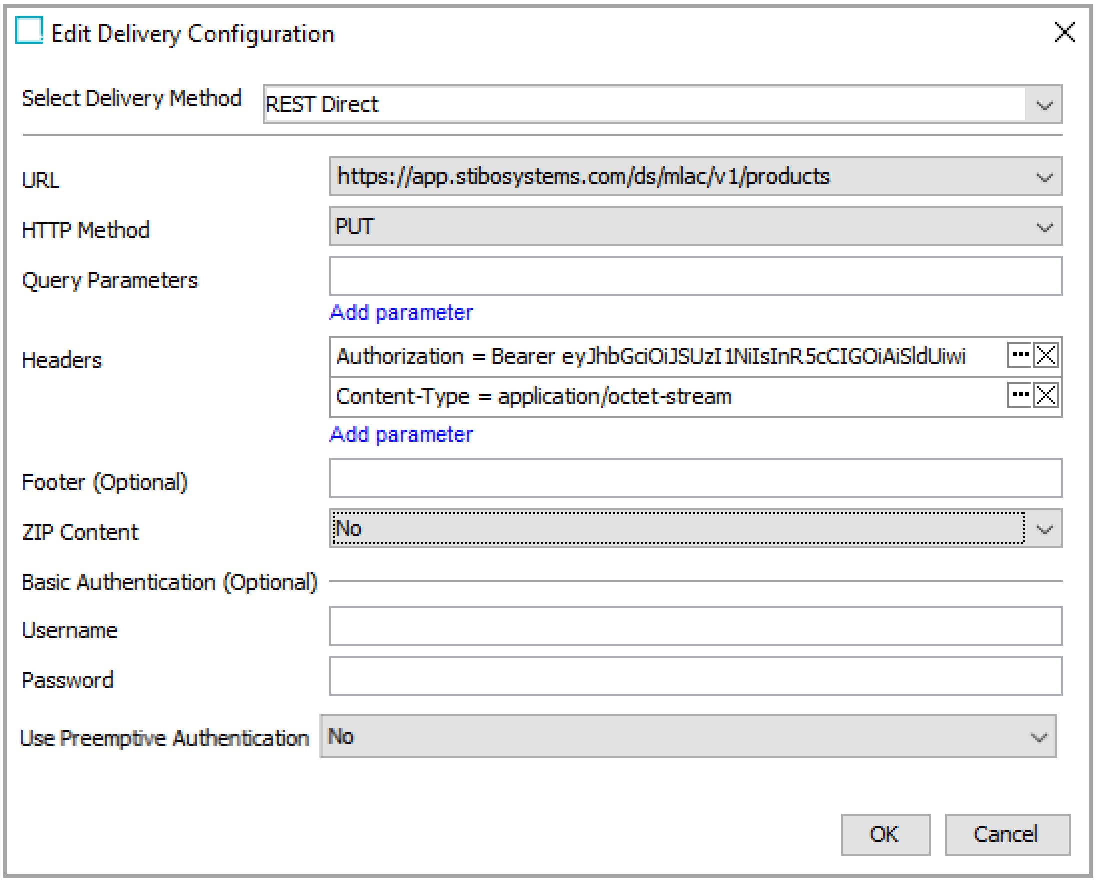
- The final state of the Output Templates flipper fields are displayed.



6. On the Delivery Method flipper, click the **Edit Delivery** link to open the 'Edit Delivery Configuration' dialog.

> **Note:** To validate the configuration, instead of setting the REST Direct method, first use the 'Copy to directory' or 'Email' delivery method and check the generated CSV file. See the **OIEP Delivery Methods** topic in the **Data Exchange** documentation for details.

- For the Select Delivery Method parameter, select **REST Direct** from the dropdown.
- For the URL parameter, select the **URL** in the dropdown.
- For the HTTP Method parameter, select **PUT**.
- For the Headers parameter, click the **Add parameter** link to display the 'Add Parameter' dialog. Set the Key parameter to **Content-Type**, the Value parameter to **application/octet-stream**, and click **OK**.
- For the Headers parameter, click the **Add parameter** link again to display the 'Add Parameter' dialog. Set the Key parameter to **Authorization**, the Value parameter to **the response from the token endpoint**, and click **OK**.
- For the ZIP Content parameter, select **No**.
- Leave other parameters blank.
- Click **OK** on the Edit Delivery Configuration dialog.



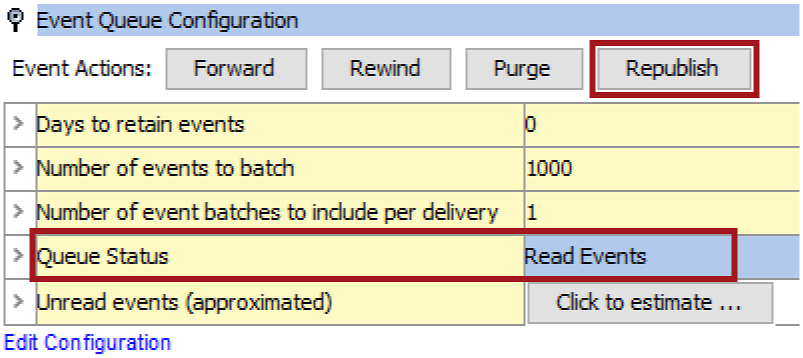
Publish Classified Products Data to MLAC

Publishing hierarchy data to the MLAC service requires invoking an OIEP with unread events. For details about these steps, see the **Running an Outbound Integration Endpoint** topic and the **Event-Based OIEP Forward, Rewind, Purge, and Republish** topic, both in the Data Exchange documentation.

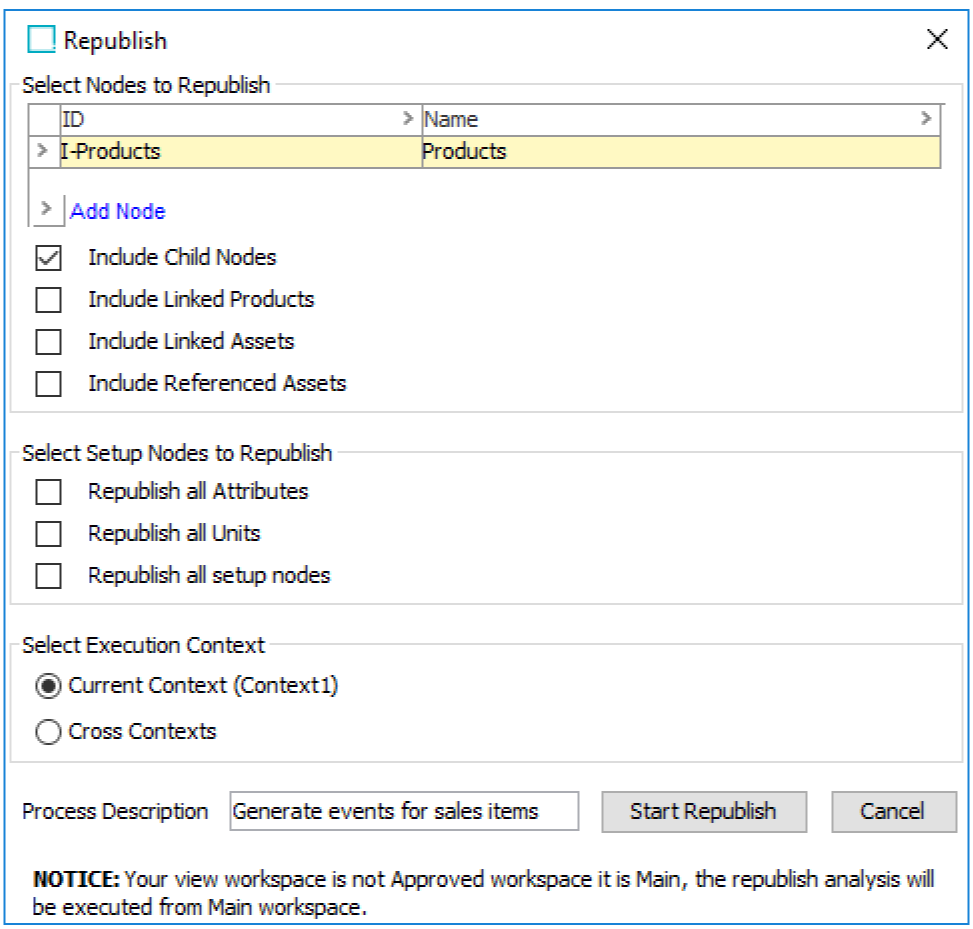
> **Important:** Before publishing data to the MLAC service, it can be useful to first validate the configuration by using either the 'Copy to directory' or 'Email' delivery method and checking the generated CSV file. See the **OIEP Delivery Methods** topic in the **Data Exchange** documentation for details.

Use the following steps to publish hierarchy data to the MLAC service.

1. On System Setup, right-click the configured OIEP and click the **Enable Integration Endpoint** option.
2. On the OIEP's Configuration tab, open the Event Queue Configuration flipper and set the Queue Status parameter to **Read Events**.
3. Click the **Republish** button to display the Republish dialog.



4. On the Republish dialog:
    - Click the **Add Node** link and add a root node that holds the products to publish.
    - Check the **Include Child Nodes** checkbox.
    - For the Process Description parameter, type a description. 'Generate events for sales items' is the description in the following image.
    - Click the **Start Republish** button.



5. Verify that the republish background process finishes successfully.
6. Right-click the OIEP and click the **Invoke** option to manually publish the hierarchy data.
7. Right-click the OIEP and click the **Disable Integration Endpoint** option.
8. On the OIEP's Configuration tab, open the Event Queue Configuration flipper and set the Queue Status parameter to **Discard Events**.

> **Important:** A model must be trained by Stibo Systems and the model made available before predictions can be generated for uploaded data.

Manual Export of Hierarchies for MLAC

Using the MLAC service requires manually exporting hierarchies from STEP as Excel or CSV files that can be handled by the service.

For general information about manual exports, see the **Export Manager** topic in the **Data Exchange** documentation.

Prerequisites

Ensure that the hierarchies being exported comply with the following standards:

- The hierarchy must not classify the same product more than once. This includes the primary product hierarchy (PPH, or the 'blue hierarchy') and classification hierarchies (the 'yellow hierarchies').

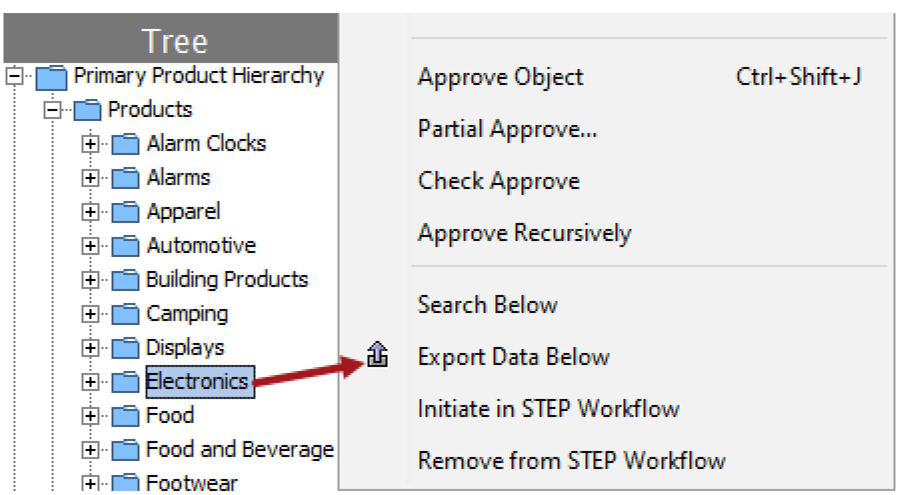
> **Important:** Classification hierarchies where the same product can be present in multiple classifications (such as website) are not allowed.

- Dimension dependent classifications are not supported.
- The hierarchy must have only one root node.
- The hierarchy must be constructed using object types different from those object types used for modeling the products already classified in the hierarchy.

Export a Hierarchy

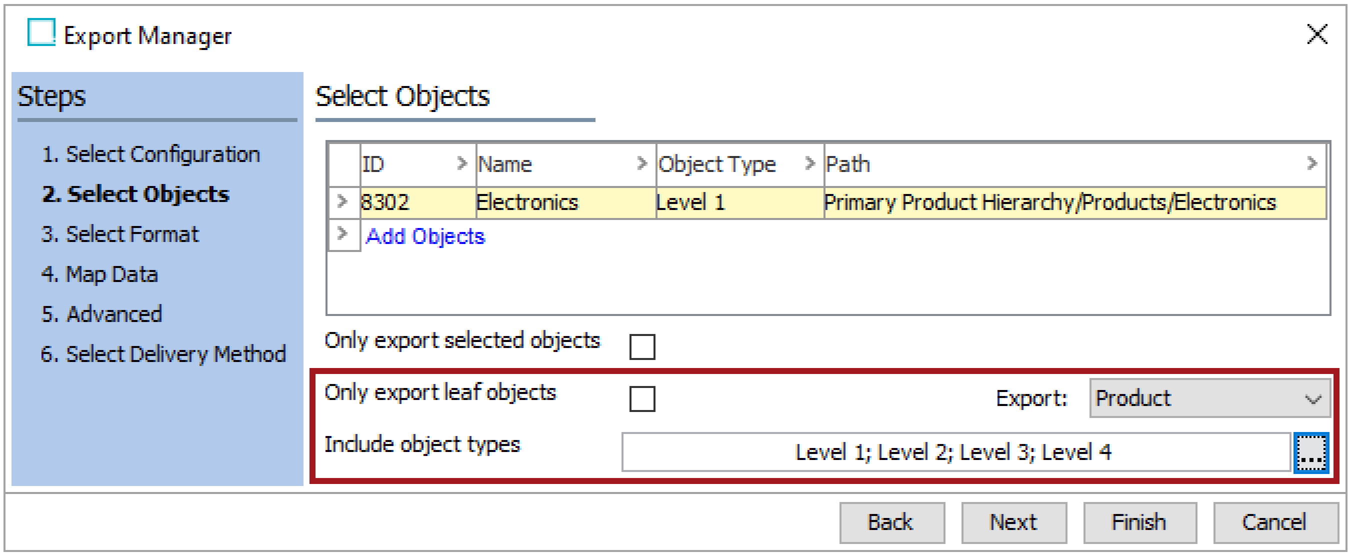
Confirm that the Prerequisites above are met and use the following steps to export a hierarchy:

1. In workbench on the Tree tab, select a hierarchy root node, right-click and select the 'Export Data Below' option to display the Export Manager with the selected objects.



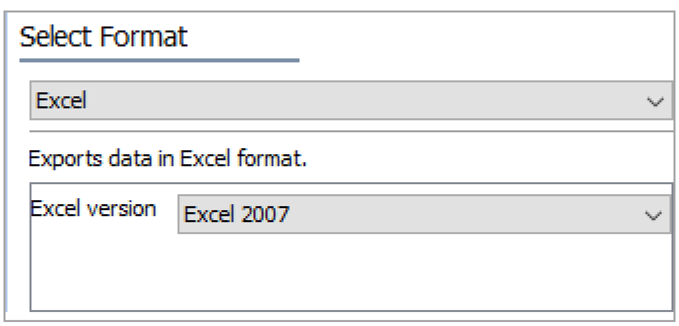
2. On the 'Select Objects' step:

- Uncheck the 'Only export leaf object' option.
- In the 'Export' dropdown, select 'Product' or 'Classification' based on the type of hierarchy being exported.
- In the 'Include object types' parameter, click the ellipsis button (...) to display the 'Select Object Types for Export' dialog. Select the object types that constitute the hierarchy and press the **OK** button.

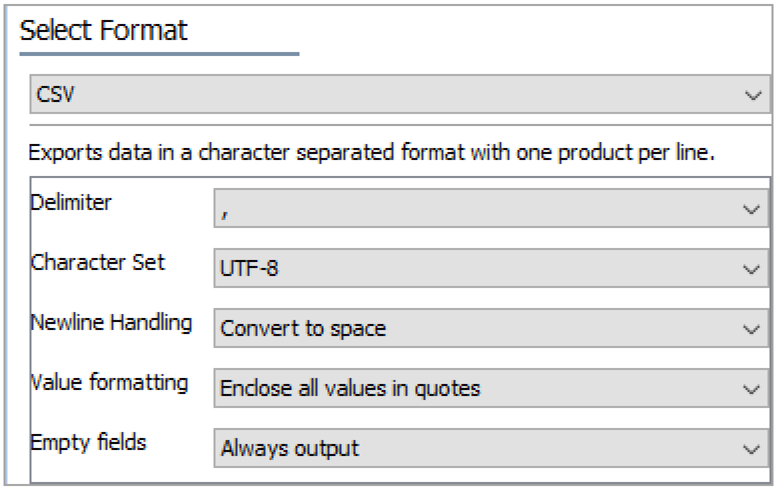


3. Click the **Next** button, and on the 'Select Format' step, choose one of the following file formats from the dropdown:

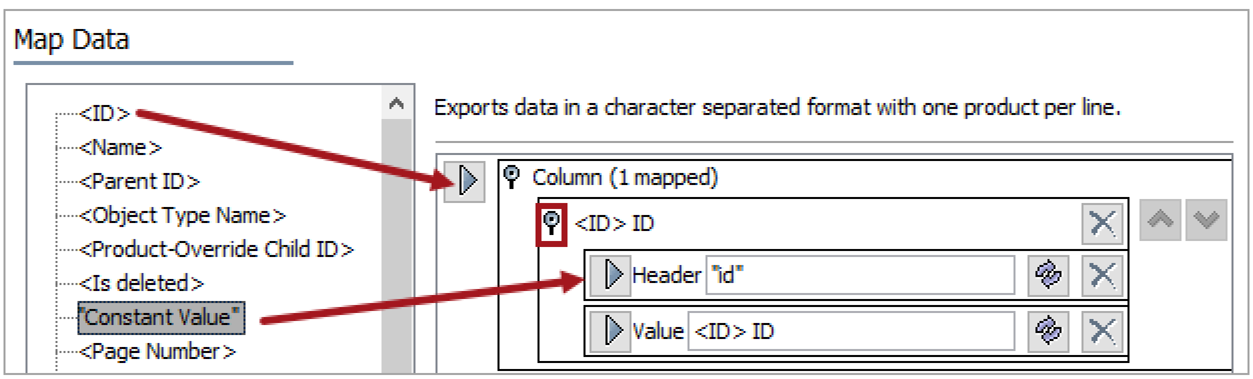
- **Excel**
  - For the 'Excel version' parameter, select **Excel 2007**.



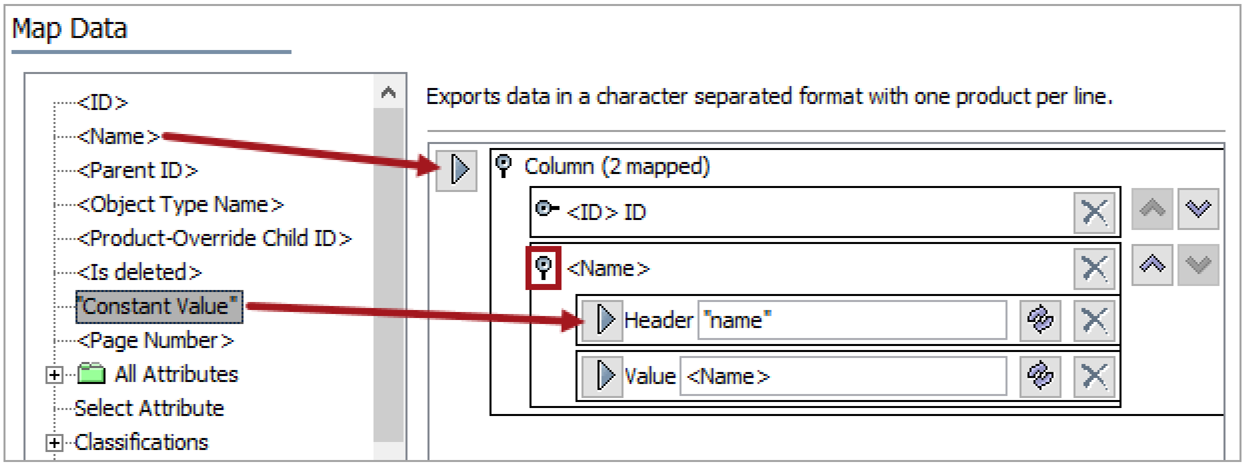
- **CSV**
  - For the 'Delimiter' parameter, select the comma (**,**).
  - For the 'Character Set' parameter, select **UTF-8**.
  - For the 'Newline Handling' parameter, select **Convert to space**.
  - For the 'Value formatting' parameter, select **Enclose all values in quotes**.
  - For the 'Empty fields' parameter, select **Always output**.



4. Click the **Next** button to display the 'Map Data' step.
5. Map the **<ID>** data source.
   - Open the '<ID> ID' flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **id** (without quotes) and click the **Save** button.
   - Click the delete button () for the original header so the Header mapping matches the image below.

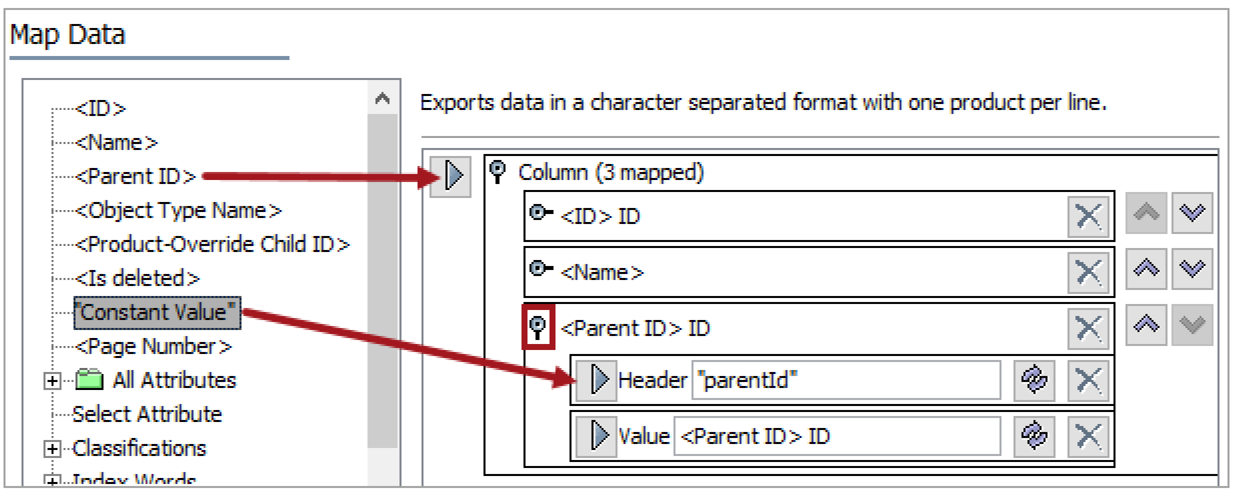


6. Map the **<Name>** data source.
   - Open the '<Name>' flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **name** (without quotes) and click the **Save** button.
   - Click the delete button () for the original header so the Header mapping matches the image below.



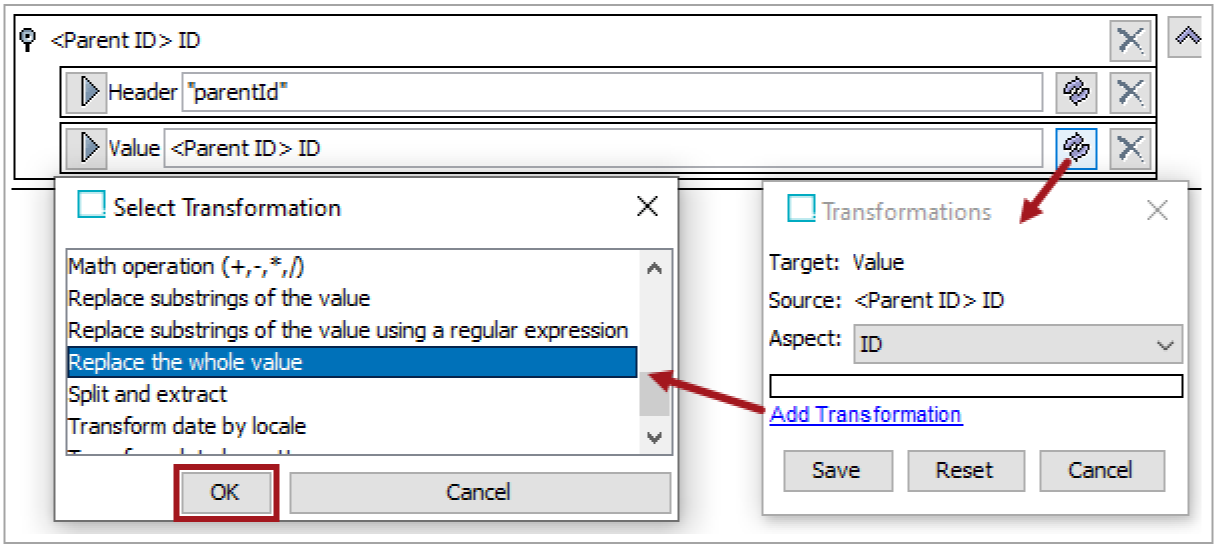
7. Map the **<Parent ID>** data source.

- Open the '<Parent ID> ID' flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **parentId** (without quotes) and click the **Save** button.
- Click the delete button ( ) for the original header so the Header mapping matches the image below.

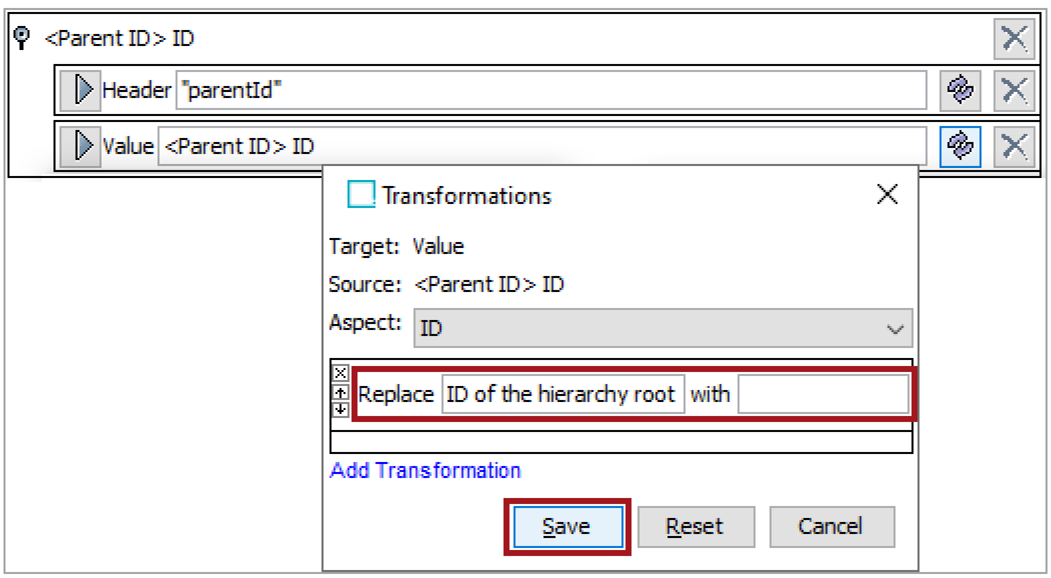


8. Transform the Parent ID Value as follows:

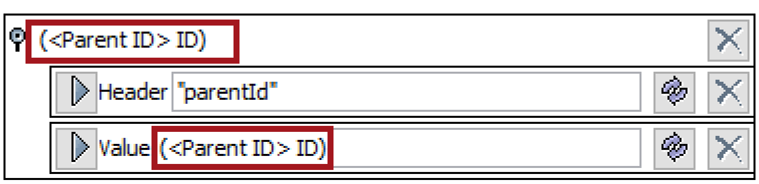
- Click the transform button ( ) for the value part of the mapped Parent ID.
- On the Transformations dialog, click the 'Add Transformation' link.
- On the Select Transformation dialog, select the **Replace whole value** option and click the **OK** button to update the Transformations dialog.



- On the updated Transformations dialog, in the 'Replace' parameter type **ID of the hierarchy root** and in the 'with' parameter, remove the value. Click the **Save** button.

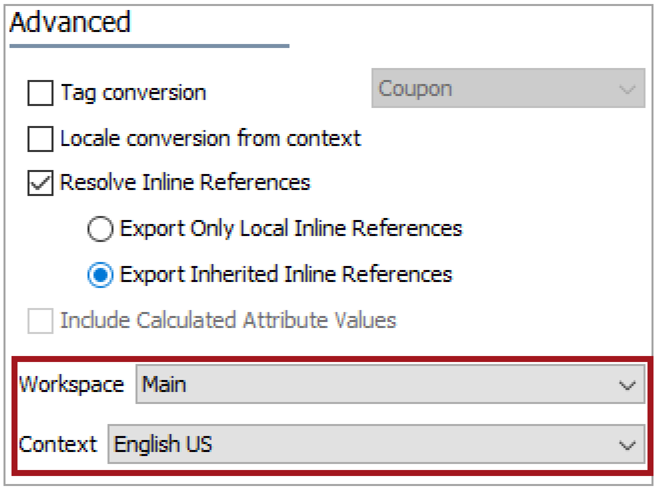


As shown in the image below, the final '<Parent ID> ID' mapping is in parentheses, like (<Parent ID> ID), to indicate a transformation is applied for the Value element.



9. Click the **Next** button to display the 'Advanced' step.

- For the Workspace parameter, choose the workspace to be used for the export. This is typically the 'Approved' workspace.
- For the Context parameter, choose the context for export.



10. Click the **Next** button to display the 'Select Delivery Method' step and click the **Finish** button.
11. On the 'Save Export Configuration' dialog, optionally save the configuration. For more information, see the **Running a Data Export** topic in the **Data Exchange** documentation.
12. Click the **OK** button to start the Export Manager Pipeline background process. For more information, see **Monitoring a Data Export** topic in the **Data Exchange** documentation.

Output Examples

The generated file should look similar to the examples below.

Excel

1	id	name	parentId
2	I-Products	Products	
3	I-Level1-1	Audio Visual Equipment	I-Products
4	I-Level2-11	Audio Visual Accessorie:	I-Level1-1
5	I-Level3-111	3D Glasses	I-Level2-11
6	I-Level3-112	Audio Visual Stands/Bra	I-Level2-11
7	I-Level2-12	Portable Audio/Video	I-Level1-1
8	I-Level3-121	Portable Music Players	I-Level2-12
9	I-Level3-122	Portable DVD Players	I-Level2-12
10	I-Level2-13	Televisions	I-Level1-1
11	I-Level3-131	LCD	I-Level2-13

CSV

```
"id","name","parentId"
"I-Products","Products",""
"I-Level1-1","Audio Visual Equipment","I-Products"
"I-Level2-11","Audio Visual Accessories","I-Level1-1"
"I-Level3-111","3D Glasses","I-Level2-11"
"I-Level3-112","Audio Visual Stands/Brackets","I-Level2-11"
"I-Level2-12","Portable Audio/Video","I-Level1-1"
"I-Level3-121","Portable Music Players","I-Level2-12"
"I-Level3-122","Portable DVD Players","I-Level2-12"
"I-Level2-13","Televisions","I-Level1-1"
"I-Level3-131","LCD","I-Level2-13"
```

Manual Export of Classified Products for MLAC

Products that are already classified in STEP can be handled by the MLAC service via an exported Excel or CSV file. While data for a hierarchy must be submitted in a single file (as defined in the **Manual Export of Hierarchies for MLAC** topic), data for products that are already classified can be submitted in one or multiple Excel or CSV files. Each file should contain hierarchy information for all hierarchies for which predictions should be generated.

File Requirements

As outlined in the Map Data steps below, the export file must have the following headers / columns in the specified order. For Excel, no quotes should be included in the header values.

- 'id' - ID of the classified product. Mandatory.
- 'description' - The description of the product. Mandatory. This must be a 'raw' source description not modified after the product was classified. Values for multiple description attributes can be supplied either by creating a calculated attribute that concatenates the values or by mapping multiple attributes to this column. More about mappings is defined below.
- 'sourceId' - Source identification, for instance a vendor ID. Optional. A column with this header must be present in the generated file even when data is not exported. Include data if the source information is available and if it has significance for the classification of the product.
- '[Hierarchy ID 1..n]' - ID of the classification (hierarchy node ID) in the hierarchy identified via the header value. Header values must correspond with the agreed upon IDs for the hierarchies.

For the primary product hierarchy case, if there are intermediate hierarchy levels between the product and the classification / hierarchy node (for example, a product family level), a calculated attribute can be used to output the appropriate hierarchy node ID for this column.

> **Important:** The example scripts should not be used as-is without thorough testing, including updating the script to match object and link types that exist on your system. JavaScript variable names are case-sensitive.

The example below shows a function that will output either the ID of the parent or ID of the grandparent, based on whether the parent is of a specific object type.

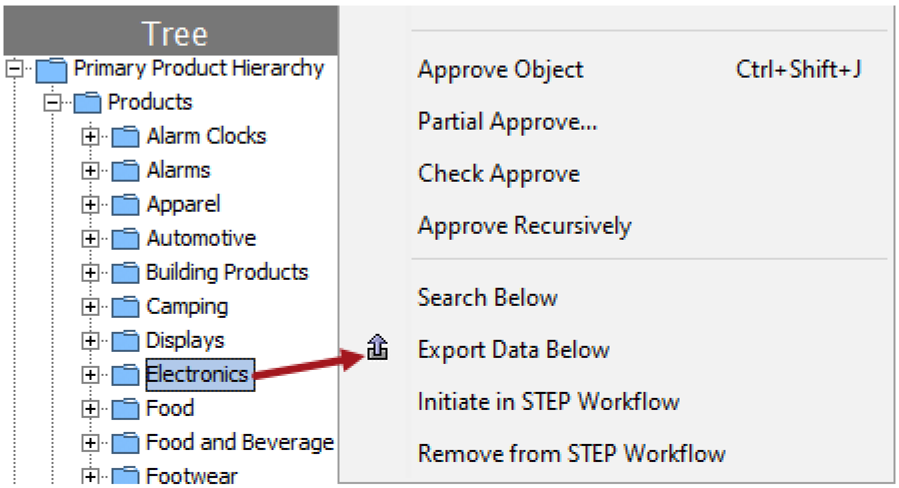
```
{
  i:=path(),
  parent:=listitem(i,listlen(i)-1),
  grandparent:=listitem(i,listlen(i)-2),
  parentobjecttype:=iterate(parent, 'stepobjecttype()')
} if (exact(parentobjecttype, "Sales Item Family"), iterate(grandparent, 'stepid()'),
iterate(parent, 'stepid()'))
```

In this example, 'stepobjecttype()' returns the object type name and not the ID. Thus, on systems with dimension dependent object type names, consider the object type for the Context of the export when making the comparison on the name.

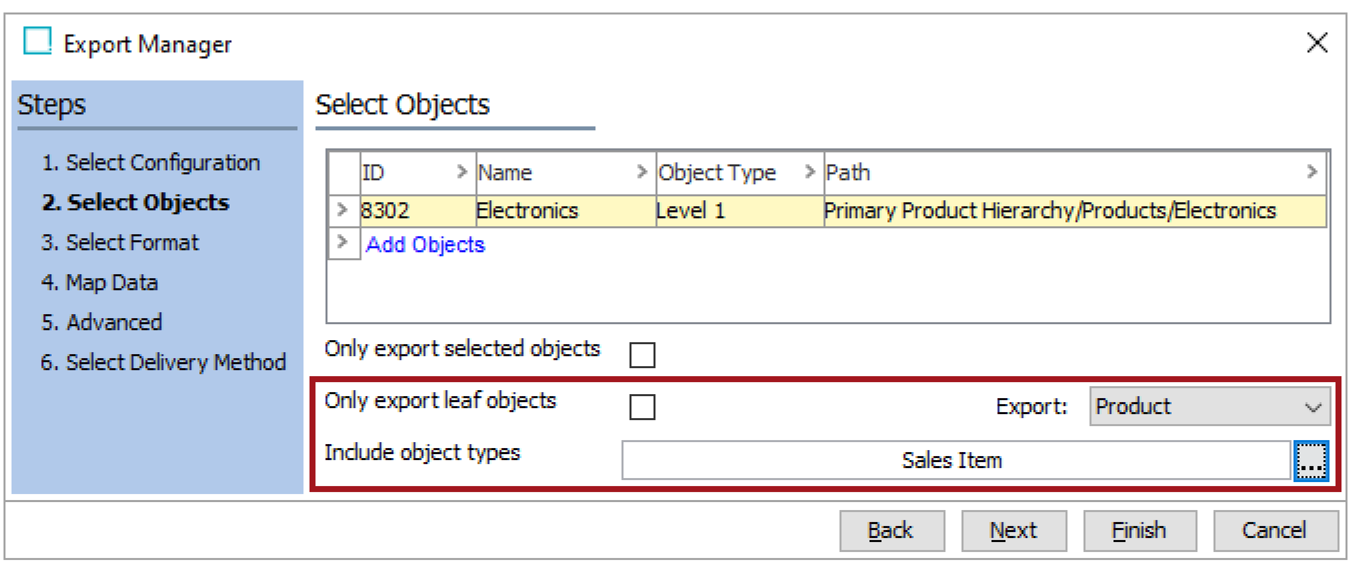
Export Classified Products

Exporting classified products involves the following steps:

1. In workbench on the Tree tab, select an appropriate root node for the export, right-click and select the 'Export Data Below' option to display the Export Manager with the selected objects.
   - If the MLAC service is only used for classifying products into a STEP classification hierarchy ('yellow hierarchy'), the root node can be the classification hierarchy root.
   - If the MLAC service is used for classifying into the product hierarchy or multiple different hierarchies, the root node should be a product node below which all relevant products can be found.

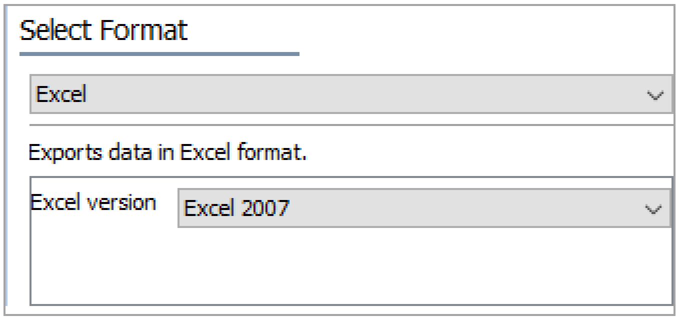


2. On the 'Select Objects' step:
   - Uncheck the 'Only export leaf object' option.
   - In the 'Export' dropdown select 'Product' based on the type of hierarchy being exported.
   - In the 'Include object types' parameter, click the ellipsis button (...) to display the 'Select Object Types for Export' dialog. Select the object type(s) for the hierarchy and press the **OK** button.

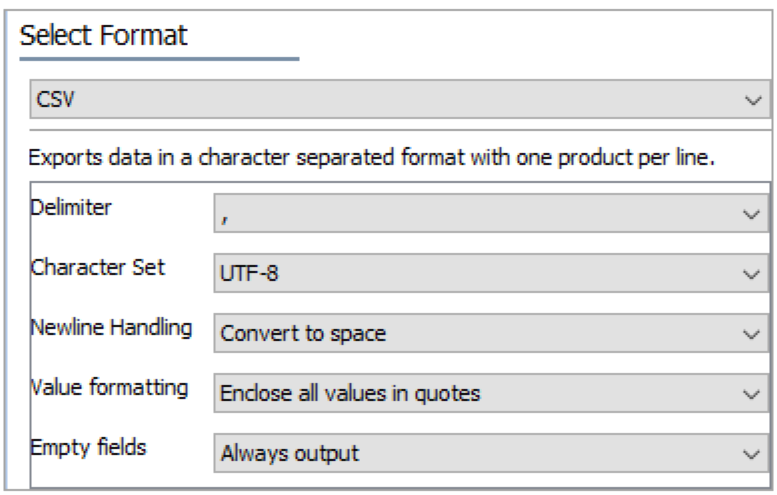


3. Click the **Next** button, and on the 'Select Format' step, choose one of the following file formats from the dropdown:

- **Excel**
  - For the 'Excel version' parameter, select **Excel 2007**.



- **CSV**
  - For the 'Delimiter' parameter, select the comma (**,**).
  - For the 'Character Set' parameter, select **UTF-8**.
  - For the 'Newline Handling' parameter, select **Convert to space**.
  - For the 'Value formatting' parameter, select **Enclose all values in quotes**.
  - For the 'Empty fields' parameter, select **Always output**.

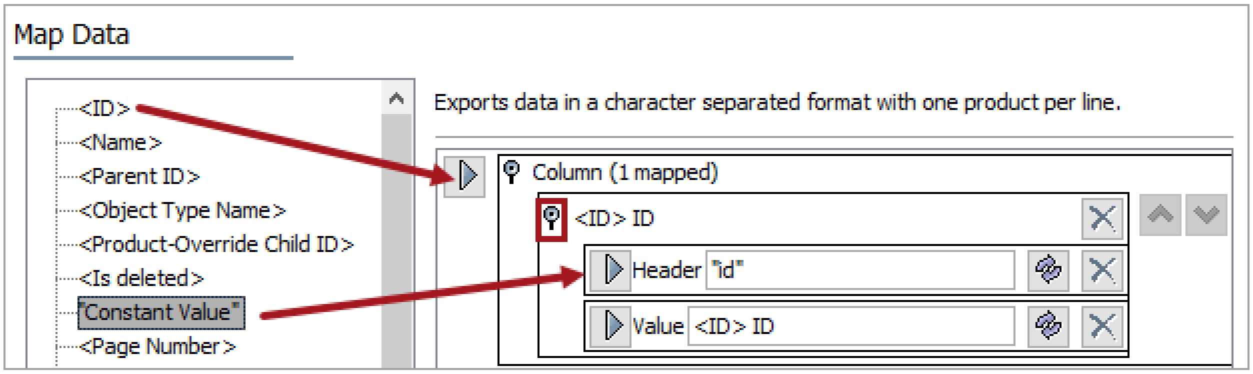


Note: See the **File Requirements** section above for definitions of the mapping elements below.

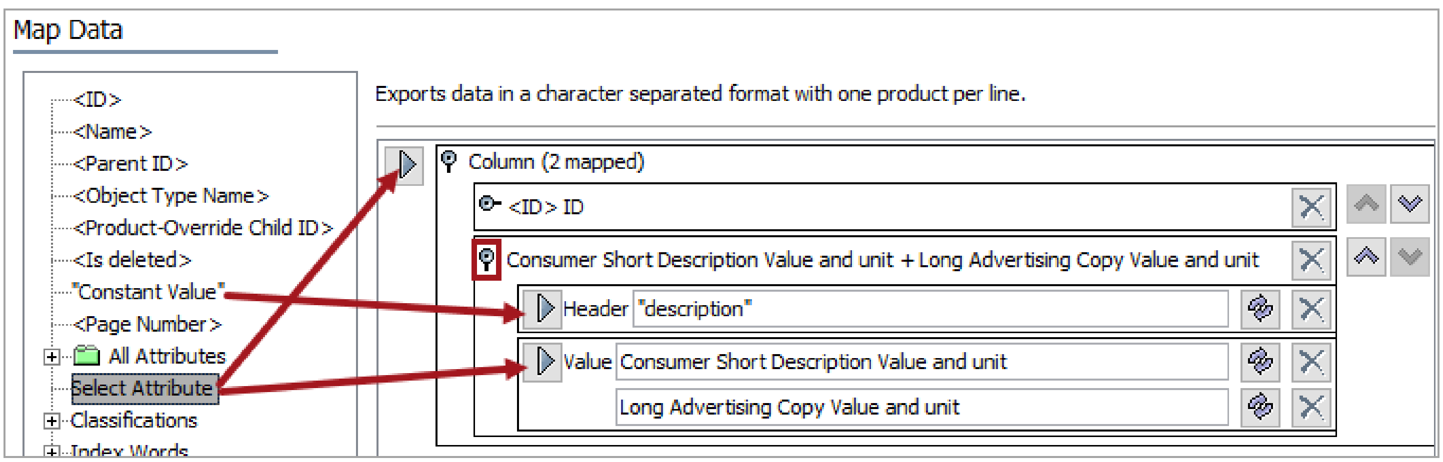
4. Click the **Next** button to display the 'Map Data' step.
5. Map the **<ID>** data source.

- Open the '<ID> ID' flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **id** (without quotes) and click the **Save** button.

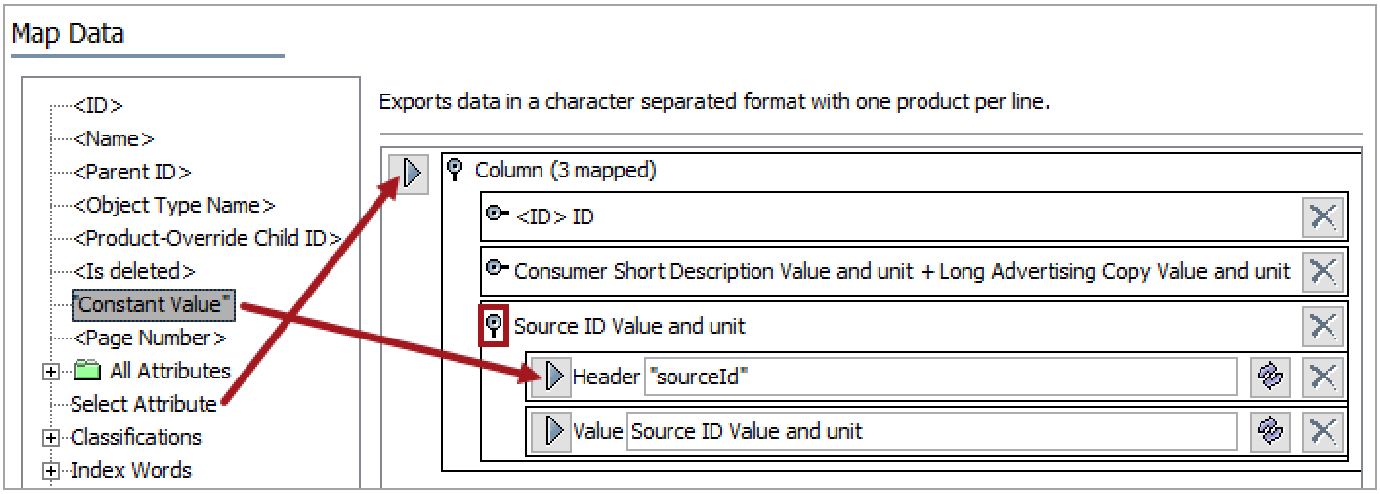
- Click the delete button ( ) for the original header so the Header mapping matches the image below.



6. Map the **Select Attribute** data source and choose the attribute that holds the description.
   - Open the description attribute flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **description** (without quotes) and click the **Save** button.
   - Click the delete button ( ) for the original header so the Header mapping matches the image below.
   - If multiple attributes hold description information, map them as additional value elements as have been mapped below for the 'Consumer Short Description' and the 'Long Advertising Copy Value' attributes.

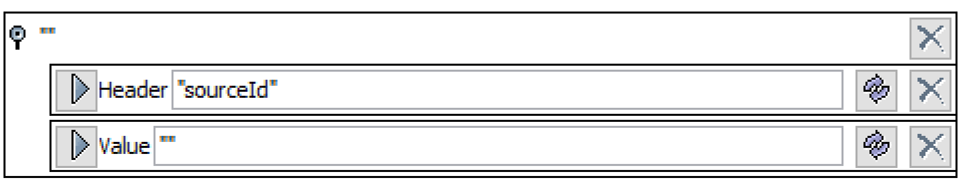


7. Choose a method to handle the source ID with significance for the classification:
   - If a source ID that has significance for the classification is available in an attribute, map the **Select Attribute** data source and choose that attribute.
     - Open the source ID attribute flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **sourceId** (without quotes) and click the **Save** button.
     - Click the delete button ( ) for the original header so the Header mapping matches the image below.



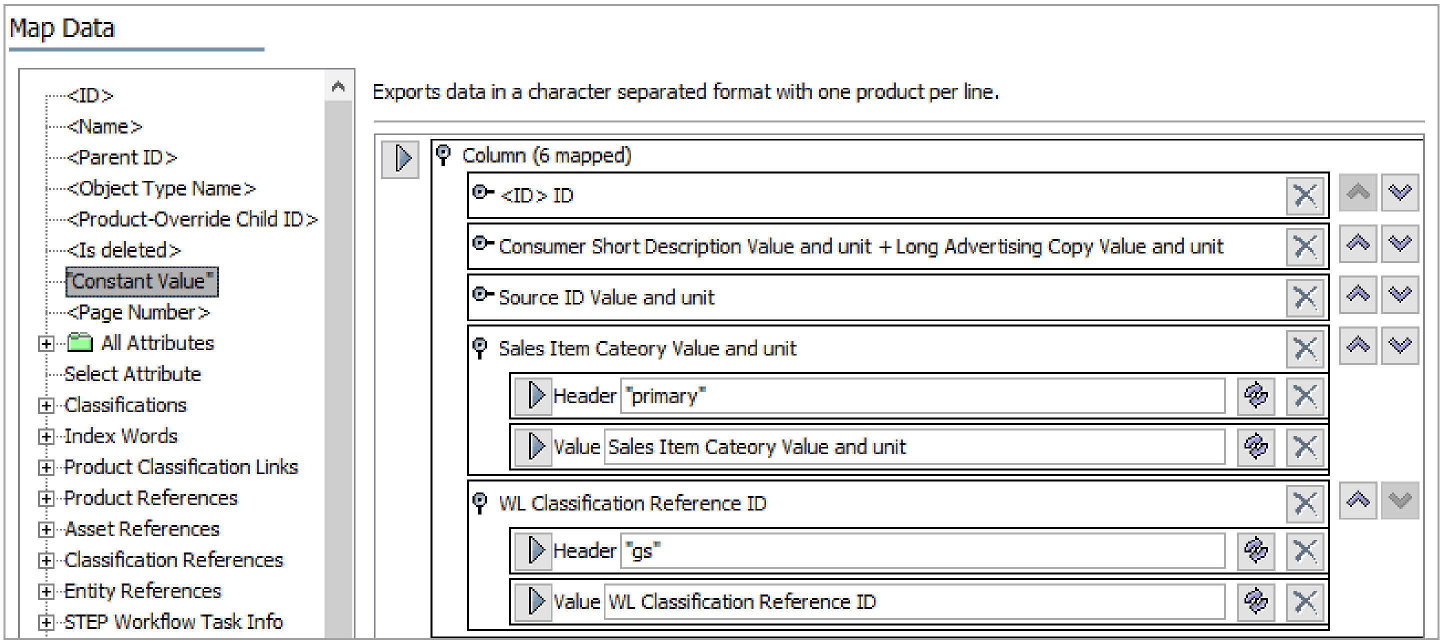
- If a source ID that has significance for the classification is not available in an attribute, map the **Constant Value** data source.
  - On the 'Enter Value' dialog, leave the 'Enter constant value' parameter blank and click the **Save** button.
  - Open the 'Constant' flipper and map 'Constant Value' as a Header row. On the 'Enter Value' dialog enter **sourceId** (without quotes) and click the **Save** button.
  - Click the delete button ( ) for the original header so the Header mapping matches the image below.
  - For the Value row, click the transform button ( ), delete the 'Constant' text from the Source parameter, and click the **Save** button.

  The final Header and Value mapping should match the image below.

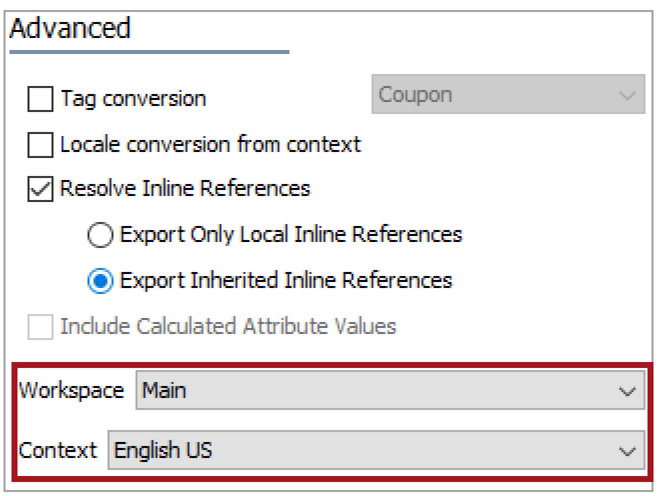


8. Choose a method to map a column for each hierarchy, in each instance replace the header with the hierarchy ID:
   - For a primary product hierarchy where there are no intermediate levels between the product and the hierarchy node, use <Parent ID> for the value.
   - For intermediate level hierarchies, map a calculated attribute that holds the appropriate hierarchy node ID.
   - For classification hierarchies, use 'Product Classification Links' data source to map the appropriate link type.

   For example, in the screenshot below, two hierarchy columns have been mapped. One by mapping a calculated attribute, and one by mapping a link type.



9. Click the **Next** button to display the 'Advanced' step.
   - For the Workspace parameter, choose the workspace to be used for the export. This is typically the 'Approved' workspace and should be the same as used for exporting hierarchies.
   - For the Context parameter, choose the same context for export as was used for exporting hierarchies.



10. Click the **Next** button to display the 'Select Delivery Method' step and click the **Finish** button.
11. On the 'Save Export Configuration' dialog, optionally save the configuration. For more information, see the **Running a Data Export** topic in the **Data Exchange** documentation.
12. Click the **OK** button to start the Export Manager Pipeline background process. For more information, see **Monitoring a Data Export** topic in the **Data Exchange** documentation.

Output Examples

The generated file should look similar to the examples below. The number of hierarchy columns depends on the use case.

Excel

	id	description	sourceId	primary	gs
2	I-SalesItem-1111	Active 3D GlassesActive 3D Glasses		I-Level3-111	W-343232
3	I-SalesItem-1121	Wall Bracket for P Series Acme TVsThe perf		I-Level3-112	W-31234395
4	I-SalesItem-1122	Wall Bracket for Z Series Acme TVsThe perf		I-Level3-112	W-55434396
5	I-SalesItem-12111	Premium Headband Music PlayerHang it ar		I-Level3-121	W-3444397
6	I-SalesItem-12113	Premium Headband Music PlayerHang it ar		I-Level3-121	W-2343342598
7	I-SalesItem-12112	Premium Headband Music PlayerHang it ar		I-Level3-121	W-34399
8	I-SalesItem-12114	Premium Headband Music PlayerHang it ar		I-Level3-121	W-34400

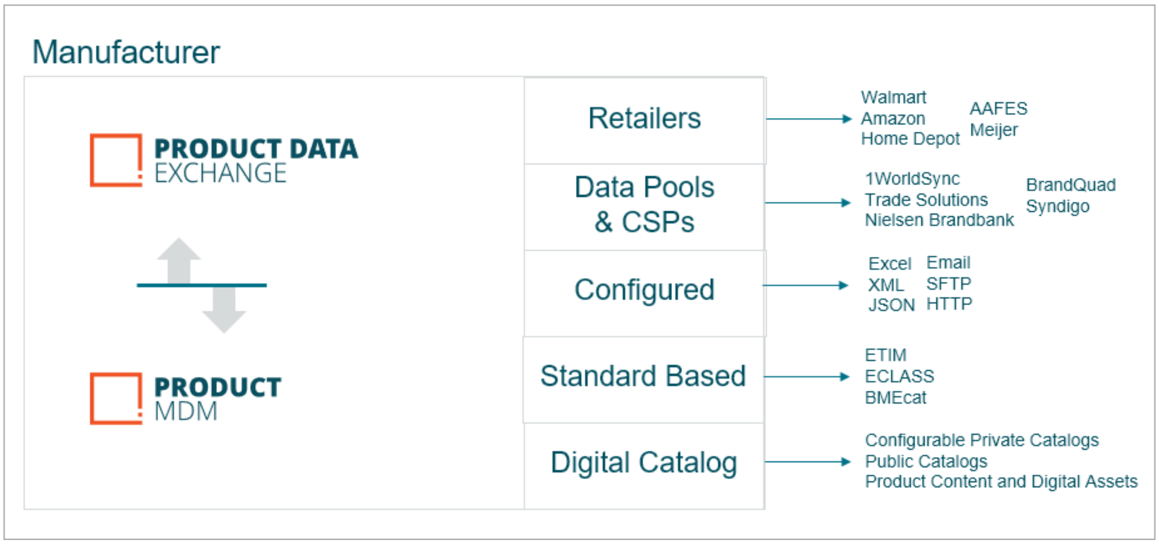
CSV

```
"id","description","sourceId","primary","gs"
"I-SalesItem-1111","Active 3D GlassesActive 3D Glasses","","I-Level3-111","W-343232"
"I-SalesItem-1121","Wall Bracket for P Series Acme TVsThe perfect fit for Acme LCD P Series TVs sized 32"" to 65""","","I-Level3-112","W-31234395"
"I-SalesItem-1122,"Wall Bracket for Z Series Acme TVsThe perfect fit for Acme LCD Z Series TVs sized 32"" to 65""",,I-Level3-112,W-55434396
"I-SalesItem-12111","Premium Headband Music PlayerHang it around your neck for personal surround sound or share your music aloud with your friends. Wi
Enjoy high quality reproduction of rich bass and clear sound with this integrated wireless MP3 player. With 16GB for your expansive playlist from cont
music.","","I-Level3-121","W-3444397"
```

Product Data Exchange

Product Data Exchange is a platform or foundation which holds product data and supports the process of mapping and transforming between different formats and requirements. It has two different services: PDX Syndication and PDX Onboarding. **PDX Syndication** provides a regularized mechanism for manufacturers to efficiently syndicate authoritative data from their internal PIM systems and use mappings and transformations to send that data to many different recipients or partners. **PDX Onboarding** allows Retailers to meet their vendors where they are and provides the functionality for the Retailer to onboard new and updated products that meet their unique requirements from vendors directly, from content service providers or from data aggregators.

The below graphic is a representation of PDX Syndication.



On-the-fly error checking in PDX is performed against channel data standards to prevent you from publishing non-standard data. Corrections for errors can be saved as rules for future use, reducing the overall time spent generating quality data.

Integration between STEP and PDX involves:

- Master data sent from STEP to PDX via an outbound integration endpoint. This can include a product's primary image, product references as well as metadata on those references, and packaging hierarchy information.
- PDX channel status information is received by STEP via an inbound integration endpoint. This allows STEP users to monitor the status of products being syndicated by PDX.

Prerequisites

To get an administrative account for PDX, talk to your Stibo Systems account manager. Your account administrator has the ability to invite you to the PDX account from the PDX UI. In doing so, you will receive a sign-up email where you define the password. You will use these credentials to log into the PDX system at https://pdx.stibosystems.com.

If you encounter any issues within the PDX site, use the live chat help in PDX – available on the question mark icon in the top right-hand corner of the application. Issues while in the STEP Web UI or STEP Workbench should be reported using your standard reporting methods.

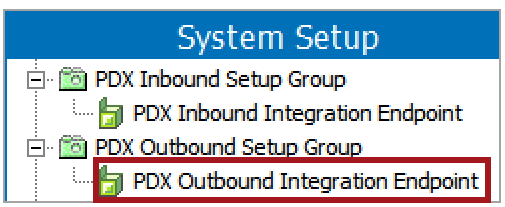
Configuration

To access the PDX platform and functionality, the 'X.Adapter.ProductDataSyndication' license must be activated on your server and the 'productdatasyndication-integration' add-on component must be installed on your system. The add-on component will create an Outbound and Inbound Integration Endpoint, both pre-configured to the greatest extent possible. Once activated, STEP communication with PDX is handled via the API using auto-generated integration endpoint objects. To install the component, you will need to know the appropriate patch level based on the version of STEP you are using. For more information regarding activating / installing the required license / component, contact your PDX representative.

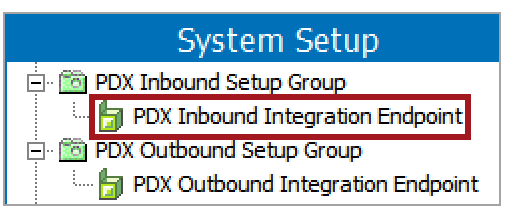
Some manual configuration setup is required for the following objects:

1. **PDX Outbound Integration Endpoint** - the OIEP sends master data to PDX and must be modified to work with your data. The **PDX Outbound Integration Endpoint Configuration** topic includes details for:

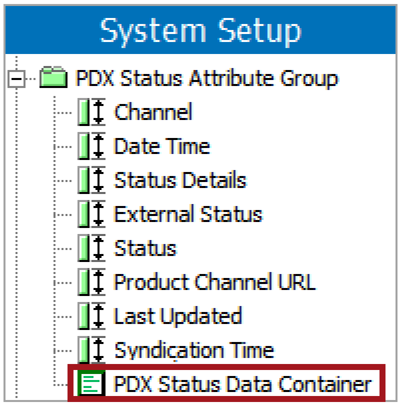
   - Manual required setup
   - Packaging hierarchy export setup
   - Optional product references metadata export setup

2. **PDX Inbound Integration Endpoint** - the IIEP receives the PDX channel status information. For details, see the **PDX Inbound Integration Endpoint Configuration** topic.

3. **PDX Status Data Container** - the data container used to display and monitor the PDX product status in the Web UI and the workbench, using the attributes contained within the PDX Status Attribute Group. For details, see the **PDX Channel Status Monitoring** section of this documentation.



4. **pdsPostActionID** - the business rule used to generate derived events from edits made to the Main workspace by clicking the Save button on the Product Editor screen in Web UI. For details, see the **Events Generated on Main Workspace** topic of the **System Setup / Super User** documentation.

PDX Category Attribute

Within the master data of PDX, products can be assigned to specific master data categories. Users that want to add additional master data categories to the PDX master data can do so by including a specific property within the sharedconfig.properties on their STEP application server. The property is used for designating which attribute contains category information for products. The value of the designated attribute is used as the category in the master data within PDX. When a product is exported from STEP to PDX and includes information regarding the category in which the product belongs, the category is added to the master data categories.

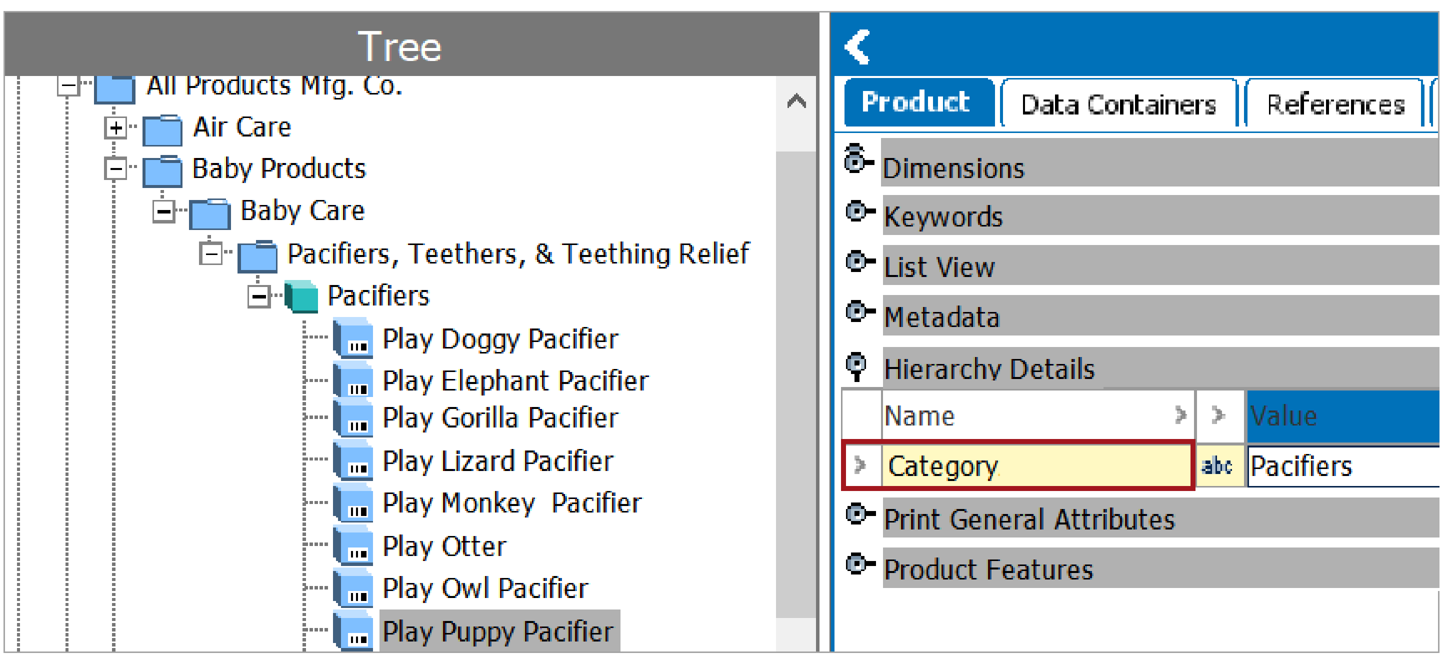
Important: While this topic will cover the specifics regarding the PDX Category Attribute, it is assumed that users have read the **Product Data Exchange** topic as well as the relevant topics that are referenced within this topic. Additionally, it is assumed that users are familiar with the procedure of creating, editing, and assigning attributes; detailed information regarding these procedures can be accessed within the topic **Attributes** in the **System Setup** documentation.

Configuring the PDX Category Attribute

Before designating the attribute that will be used to convey product category details from STEP to PDX, the property 'PDSDelivery.CategoryAttributeID' must be added to the sharedconfig.properties file on your STEP application server. The ID of the attribute that is created to house the category name (in the form of a value) must be assigned to this property. In the example below, the attribute 'Category' has been created and will be used to house the category details; therefore, the property in the sharedconfig.properties file would look like this:

```
PDSDelivery.CategoryAttributeID=Category
```

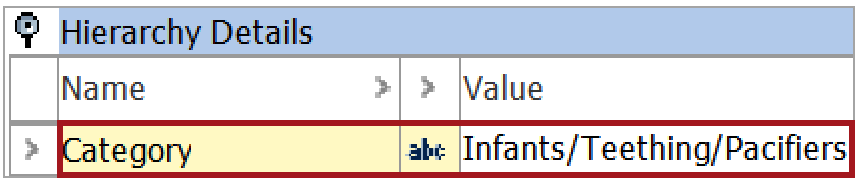
The image below shows the product 'Play Puppy Pacifier' with the assigned attribute 'Category.' This attribute has a value of 'Pacifiers,' which will be the category in which 'Play Puppy Pacifier' will be included in PDX.



Running a PDX Outbound Integration Endpoint(OIEP) that has Advanced STEPXML as the configured format of data delivery will produce the following data, as part of a JSON file, which can be read by PDX.

```
   "products": [
{
   "__ID": "141435",
   "__NAME": "Play Puppy Pacifier",
   "__CATEGORY": "Pacifiers",
```

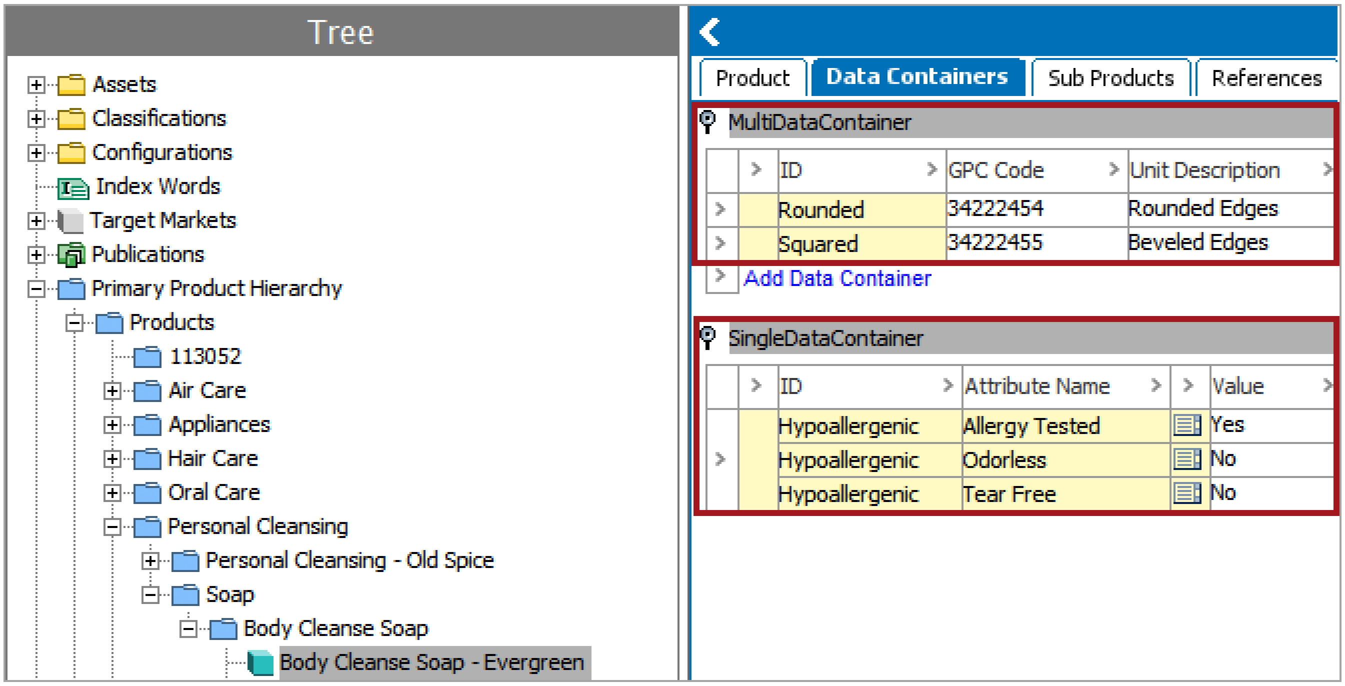
Note the Category value 'Pacifiers.' Because the attribute 'Category' was assigned as the value for 'PDSDelivery.CategoryAttributeID' in the sharedconfig.properties file, PDX will assign the product 'Play Puppy Pacifier' to the category 'Pacifiers.' To create a nested category, create a file path for the value of the 'Category' attribute, as shown below.



For more information on the PDX OIEP, see the **PDX Outbound Integration Endpoint Configuration** topic.

PDX Data Container Exports

By providing specific information via the Advanced STEP XML template, users are able to export selected data containers from STEP to PDX using the PDX outbound integration endpoint.



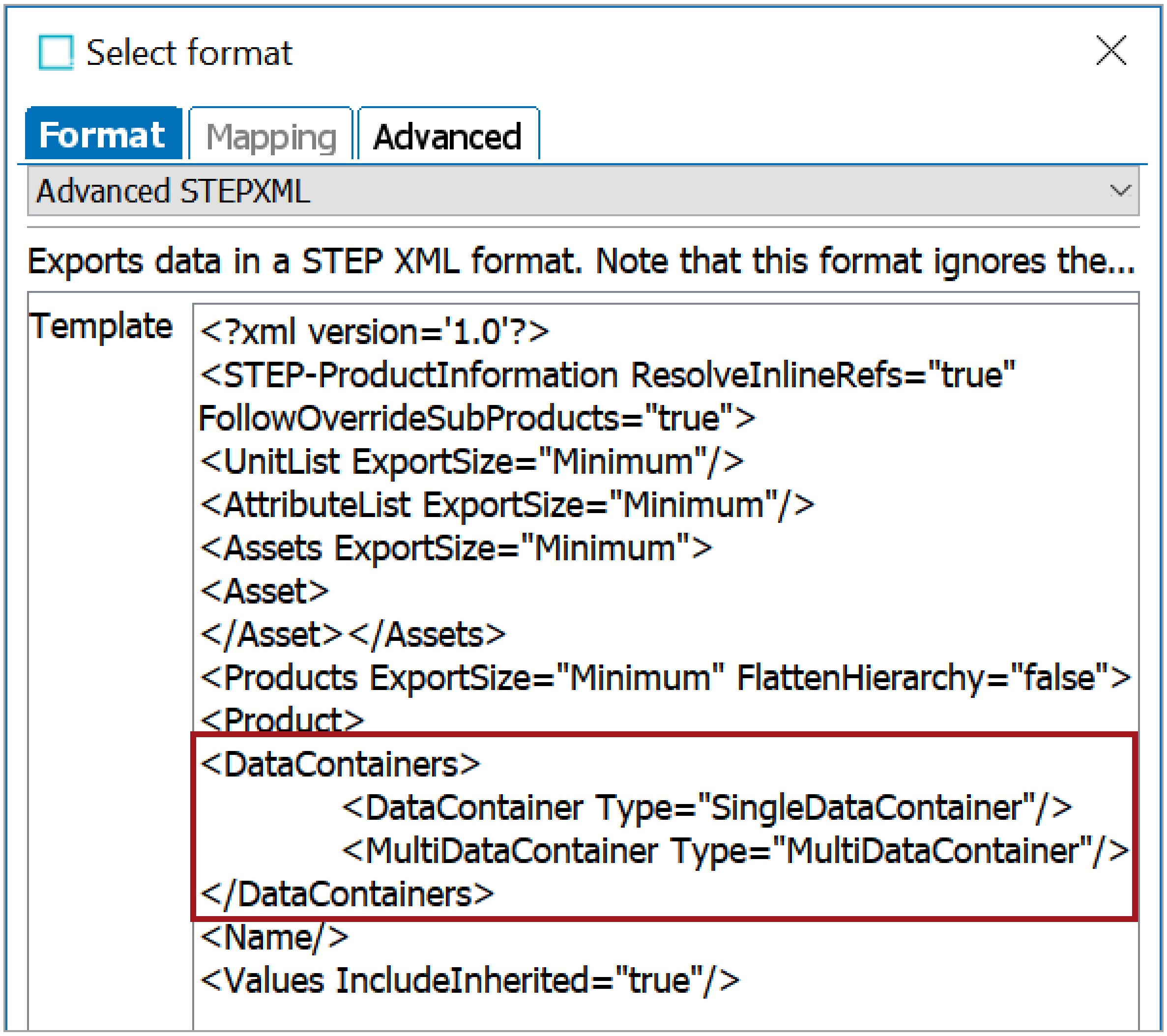
In the example above, the selected product includes both a single and multi data container, as well as a PDX Status Data Container. In this topic we will be focusing on the single and multi data containers.

For more information regarding data containers, including single and multi data containers, see the **Data Containers** topic with the **System Setup** documentation.

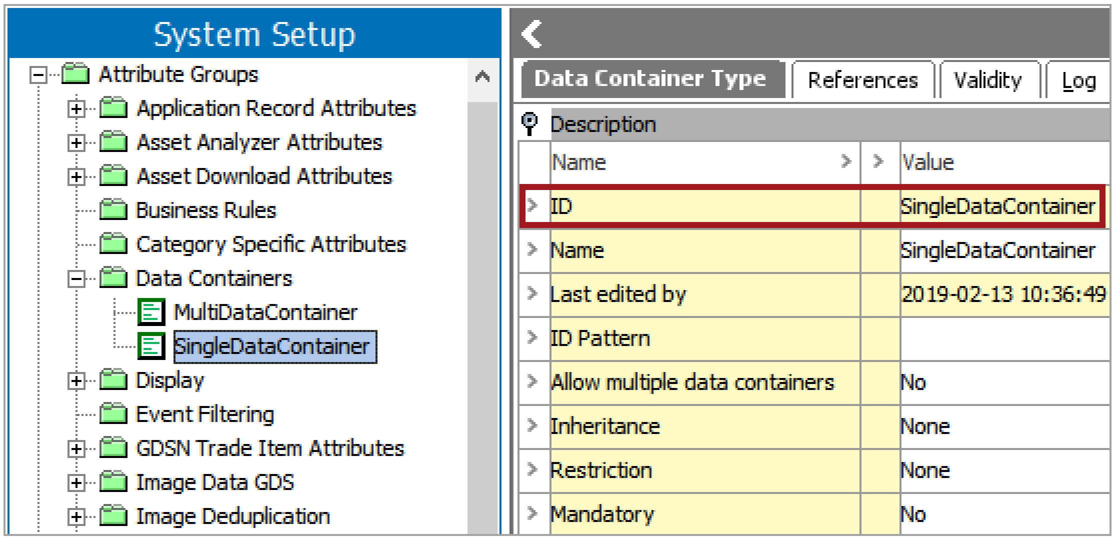
Note: The PDX Status Data Container will never be exported to PDX, as any information provided by this container is not relevant within PDX. For more information regarding the PDX Status Data Container, see the **PDX Status Data Container Configuration** topic in the **Product Data Exchange** documentation.

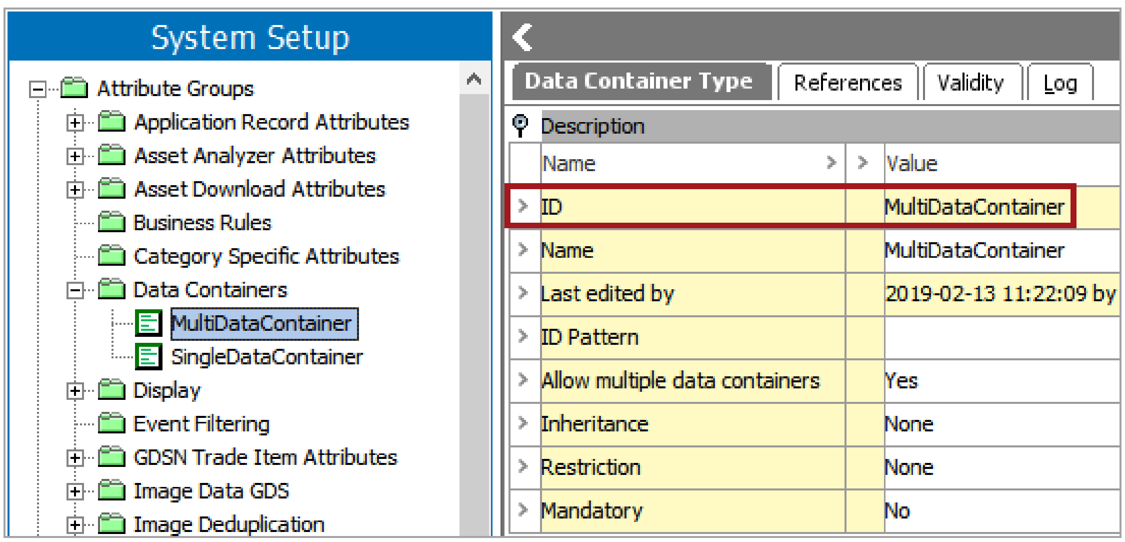
Configuring Advanced STEPXML for Data Container Exports

In order to export selected data containers to PDX using the PDX outbound integration endpoint, the Advanced StepXML template must be configured in a way that includes the data container type ID within a '<DataContainers>' tag. As the image below shows, the '<DataContainers>' tag includes the IDs for both the SingleDataContainer and MultiDataContainer data container types as values for '<DataContainer Type>' tags.



In the example above, the DataContainer and MultiDataContainer tag with the Type attribute is used. Only data containers within the open and closed DataContainer tags and identified by the type ID (in this case, SingleDataContainer and MultiDataContainer) are exported. If you only specify <DataContainers/> without any specific data container types in the template, then all data containers for your product (except the PDX Status Data Container) will be extracted. For more information, see the **Filter Data Containers in STEPXML** topic in the **STEPXML Format** documentation.





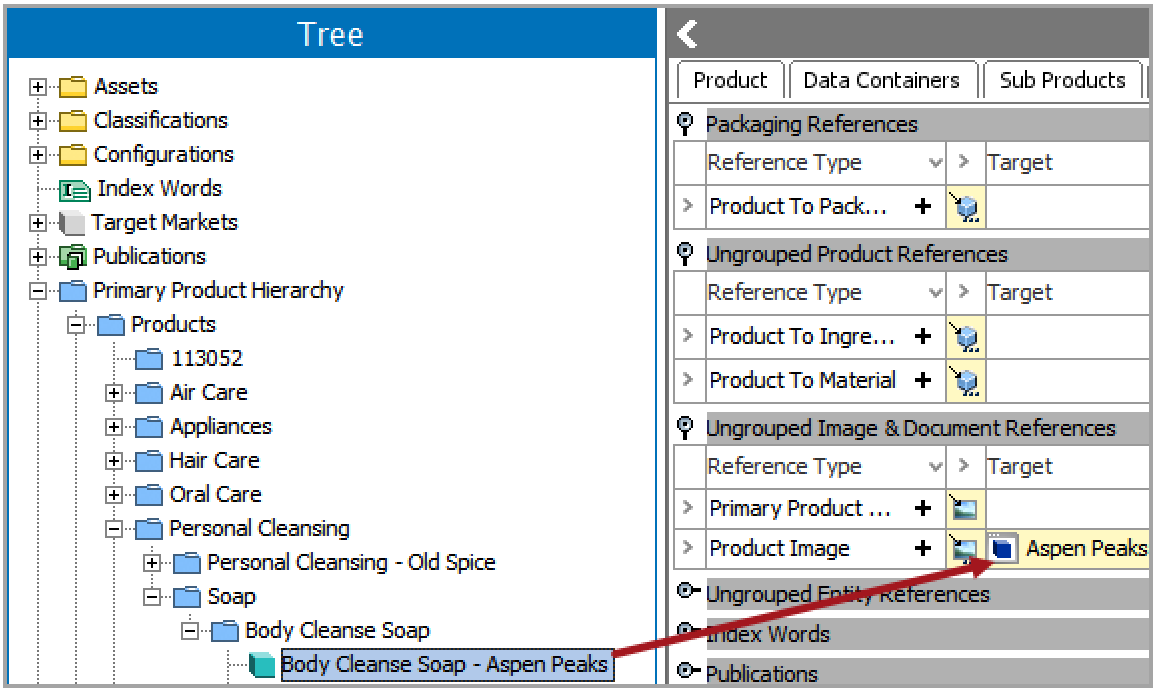
As shown above, the user-created ID for the SingleDataContainer and the MultiDataContainer matches the DataContainer Type and MultiDataContainer Type values in the Advanced STEPXML template. When the PDX Outbound Integration Endpoint is invoked, the data containers will be created as composite attributes in PDX, and the names of those composite attributes will correspond with the ID of the data container type in STEP. For instance, if a tag in the Advanced STEPXML template is <DataContainerType="SingleDataContainer"/> then the name of the composite attribute will be SingleDataContainer.

For more information on data containers in STEPXML, see the **Data Containers in STEPXML** topic in the **STEPXML Format** documentation.

Exporting Metadata Attributes of an Asset to PDX

By configuring the PDX Outbound Integration Endpoint (OIEP), users are able to export metadata attributes on assets that are referenced by a product that is configured to be (or has already been) exported to PDX.

As shown in the graphic below, the asset 'Aspen Peaks,' of type 'Product Image,' is being referenced by the product 'Body Cleanse Soap - Aspen Peaks.' This topic will cover how the metadata for an asset referenced by a product is delivered to PDX.



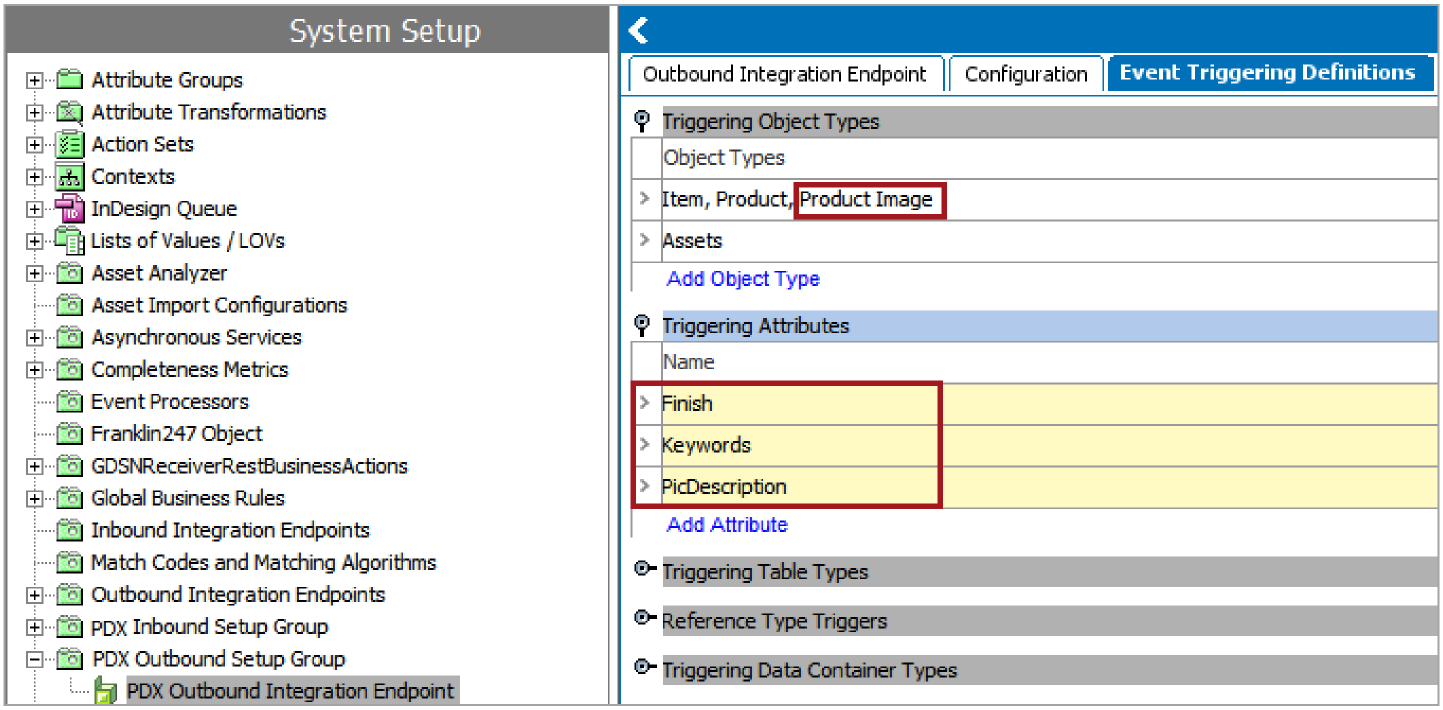
For more information regarding linking assets to products, see the **Linking Assets to Products** topic in the **Asset Maintenance** documentation.

Configuring the PDX Outbound Integration Endpoint for Exporting Asset Metadata

To export asset metadata referenced by a product in PDX, the PDX OIEP must be configured. In addition to this configuration, there is also an option to include specific metadata attributes of the asset that can be used to trigger the endpoint to be invoked. For information regarding enabling and invoking an endpoint, see the **Running an Outbound Integration Endpoint** topic in the **Outbound Integration Endpoints** documentation. For information regarding event-based triggering definitions, see the **OIEP - Event-Based - Event Triggering Definitions Tab** topic in the **Outbound Integration Endpoints** documentation.

As shown below, the 'Product Image' must be included as an Object-Eventtype for metadata of the asset to be exported to PDX. The 'PDX Outbound Preprocessor' Pre-Processor handles change events on reference assets, and forwards this information to the PDX product.

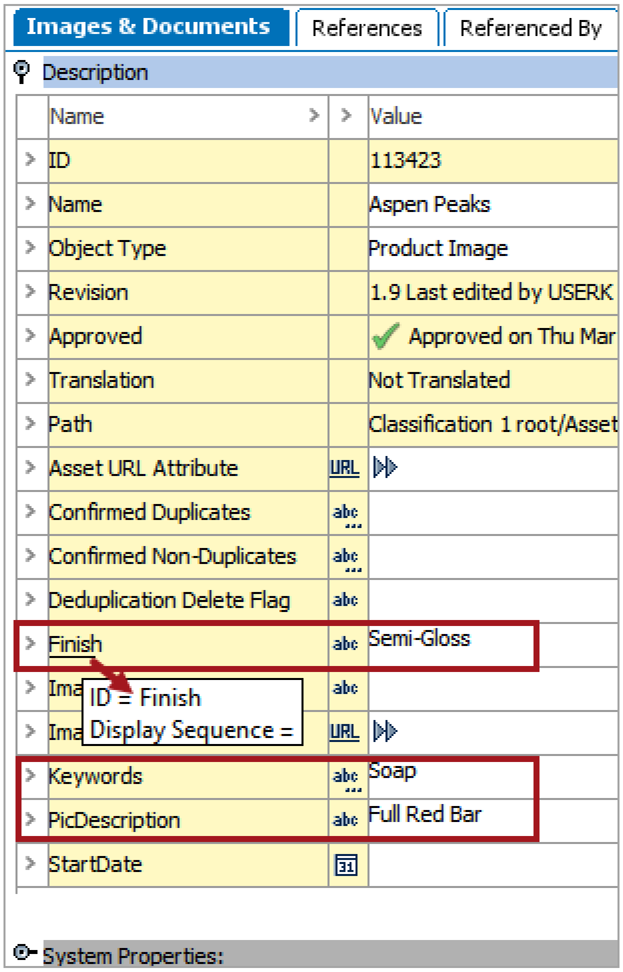
To use specific attributes to invoke the PDX OIEP and export any updated asset metadata to PDX, 'Product Image' must also be included as one of the Triggering Object Types. In addition, one or more attributes of the asset must be added to the Triggering Attributes list. In the example below, the attributes 'Finish', 'Keywords', and 'PicDescription' have been added as triggering attributes. Any changes to these attributes, once approved, will resend any updated changes to the asset metadata to PDX.



For more information regarding configuring the PDX Outbound Integration Endpoint, see the **PDX Outbound Integration Endpoint Configuration** topic in the **Product Data Exchange** documentation.

Configuring the Advanced STEPXML Template for Asset Metadata Export

By configuring the Advanced STEPXML template, users can specify whether they want specific metadata attributes for assets to be exported, or all metadata attributes for assets to be exported. In the example below, three attributes on the asset and their values are highlighted.



Note: To quickly identify the attribute ID for the purpose of insertion into the XML template, hover over the attribute name. A popup will display with the attribute ID.

Using the 'Value AttributeID' tag, the corresponding Advanced STEPXML template has been configured to include those same attributes, identified by their IDs, within the <Values></Values> element.

```
<?xml version='1.0'?>
<STEP-ProductInformation ResolveInlineRefs="true" FollowOverrideSubProducts="true">
<UnitList ExportSize="Minimum"/>
<AttributeList ExportSize="Minimum"/>
<Assets ExportSize="Minimum">
<Asset>
<Values>
<Value AttributeID="AssetAnalyzer.Keywords"/>
<Value AttributeID="Finish"/>
<Value AttributeID="PicDescription"/>
</Values>
</Asset>
</Assets>
```

When the PDX OIEP is invoked, the resulting JSON structure exported to PDX will display the specified metadata attribute values, as well as the assetID of the specific asset, as a composite attribute with the name 'ProductImage.' This composite attribute will be displayed on the product in PDX, not on the asset.

```
"ProductImage": [
    {
        "assetID": "113423",
        "AssetAnalyzer.Keywords":"Soap",
        "Finish": "Semi-Gloss"
        "PicDescription": "Full Red Bar",

    }
]
```

Note: In the example above, the ID of the asset reference type is 'ProductImage.'

For more information regarding Advanced STEPXML, see the **Advanced STEPXML Format** topic in the **Data Formats** documentation.

PDX Outbound Integration Endpoint Configuration

The PDX Outbound Integration Endpoint (OIEP) is used to send STEP master data (for example, a product's primary image, product references as well as metadata on those references, and packaging hierarchy information) to PDX.

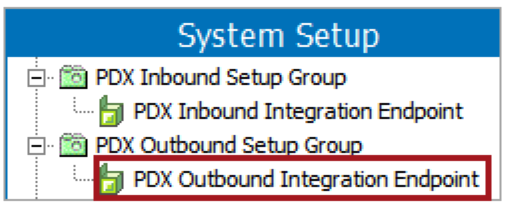
The PDX OIEP includes a predefined template and a number of other settings to reduce the setup required. However, the following updates are necessary for it to work with your data model.

This topic includes information and/or links to more information for the following configurations:

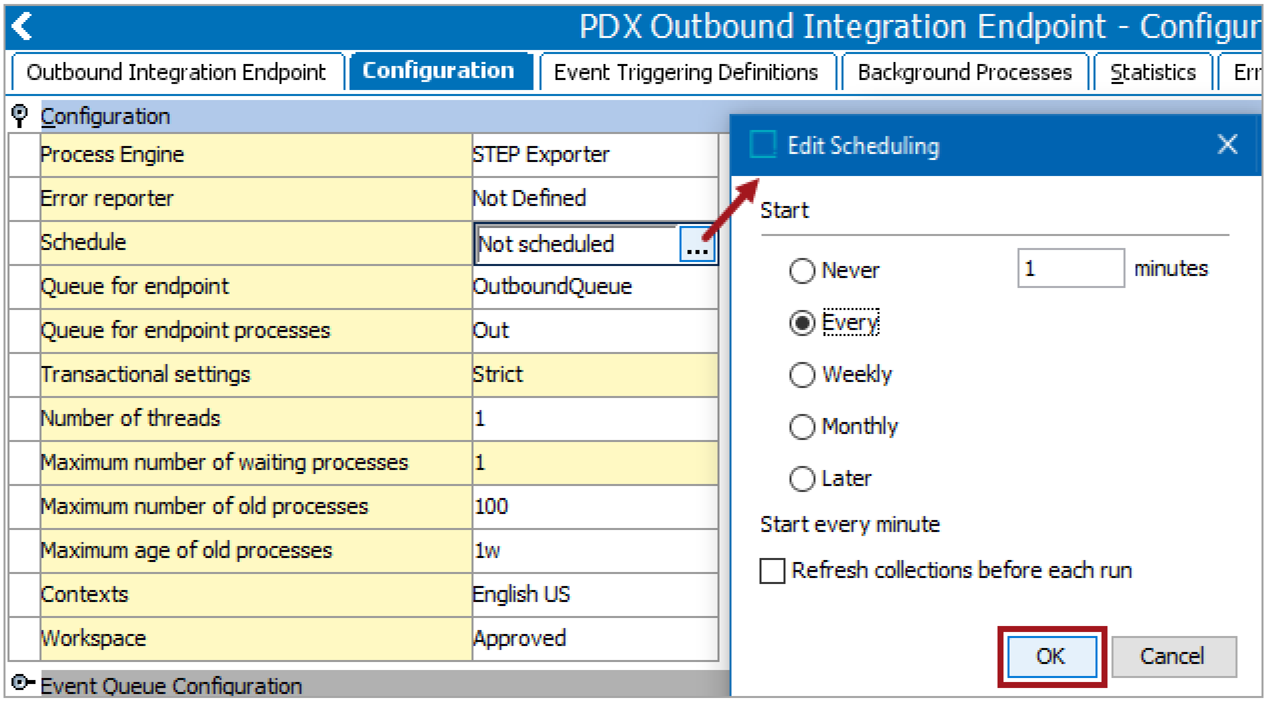
- Required manual setup
- Packaging hierarchy information export setup

Required Manual Setup

1. In the System Setup > PDX Outbound Setup Group node, open the PDX Outbound Integration Endpoint.



2. On the OIEP Configuration tab, open the Configuration flipper, and click into the **Schedule** parameter value to display an ellipsis button (...).
3. Click the ellipsis button (...) to display the Edit Scheduling dialog and set the required Start interval. Click the **OK** button to update the Schedule parameter.

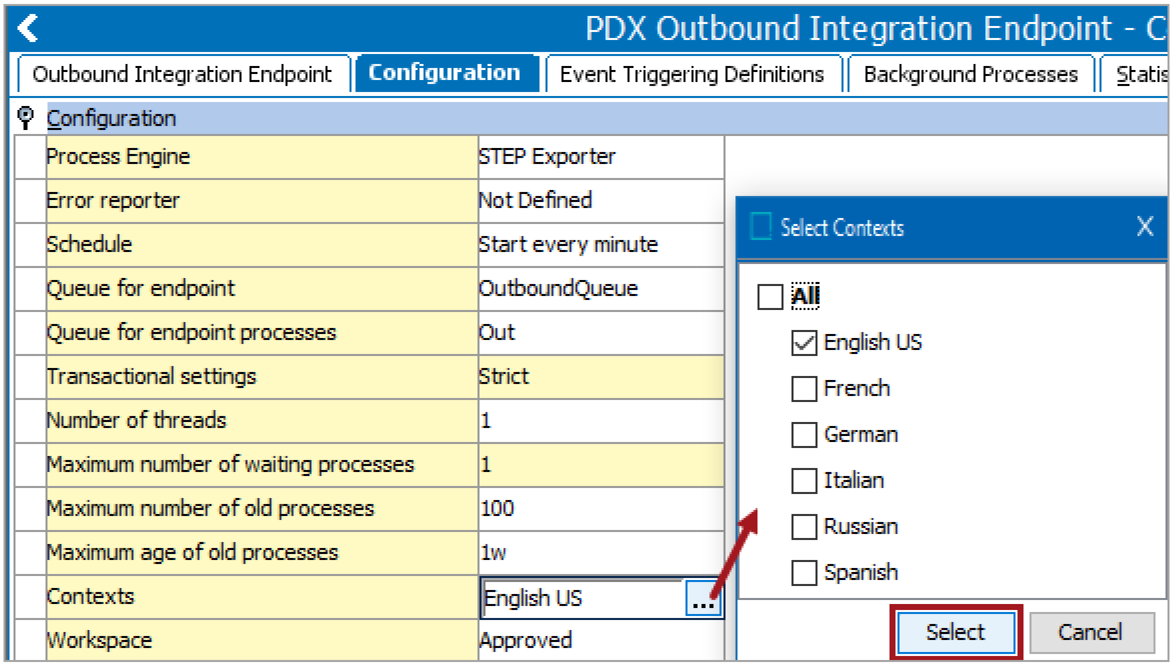


Important: Consider the time zone of the application server compared to that of the workbench (the client) where the schedule is created or viewed. When scheduling a job, the local time zone is displayed in the workbench, but the time zone of the server is used to run the background process. Although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

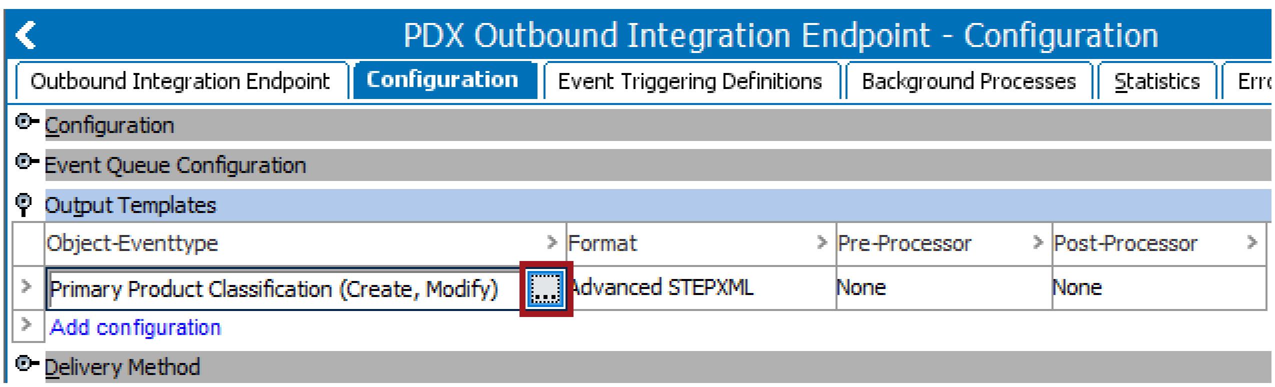
4. Click into the **Contexts** parameter value to display an ellipsis button (**...**).

5. Click the ellipsis button (**...**) to display the Select Contexts dialog and set the required single context to export. Click the **Select** button to update the Contexts parameter.

Note: A locale has a language ISO code associated with it; the ISO code is a language qualifier in the JSON that is sent to PDX. PDX is able to read that qualifier and 'translate' it into a language. For more information regarding locales, see the **Context Locales** topic in the **Contexts** documentation.

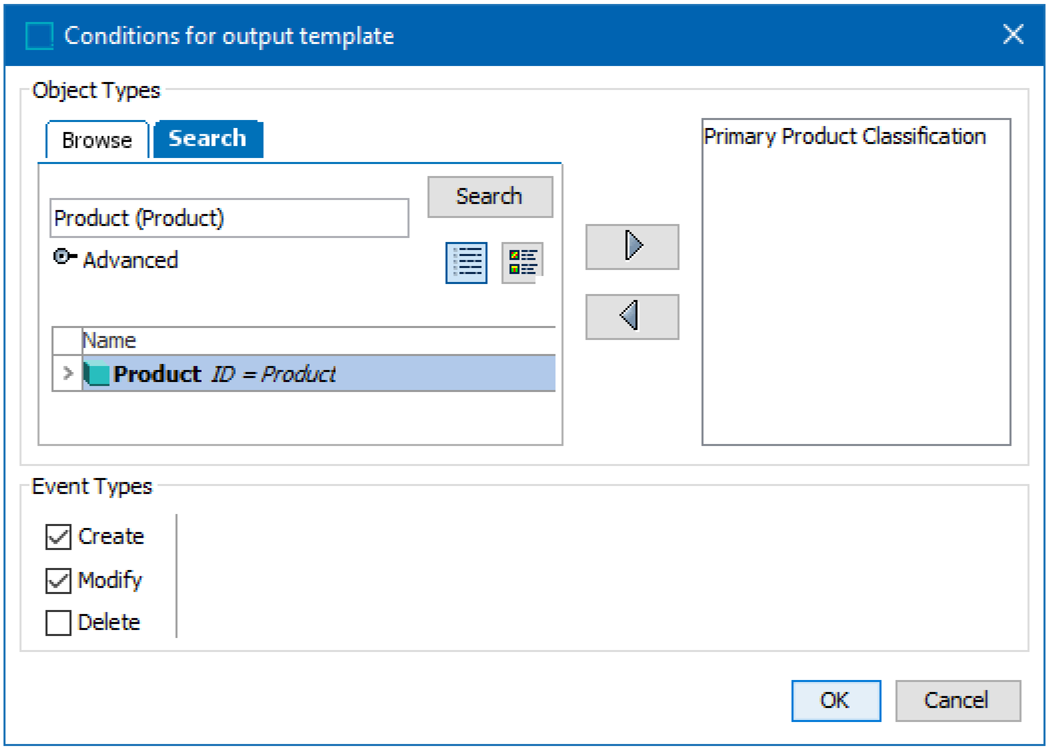


6. Open the Output Templates flipper and click into the **Object-Eventtype** parameter value to display an ellipsis button (...).



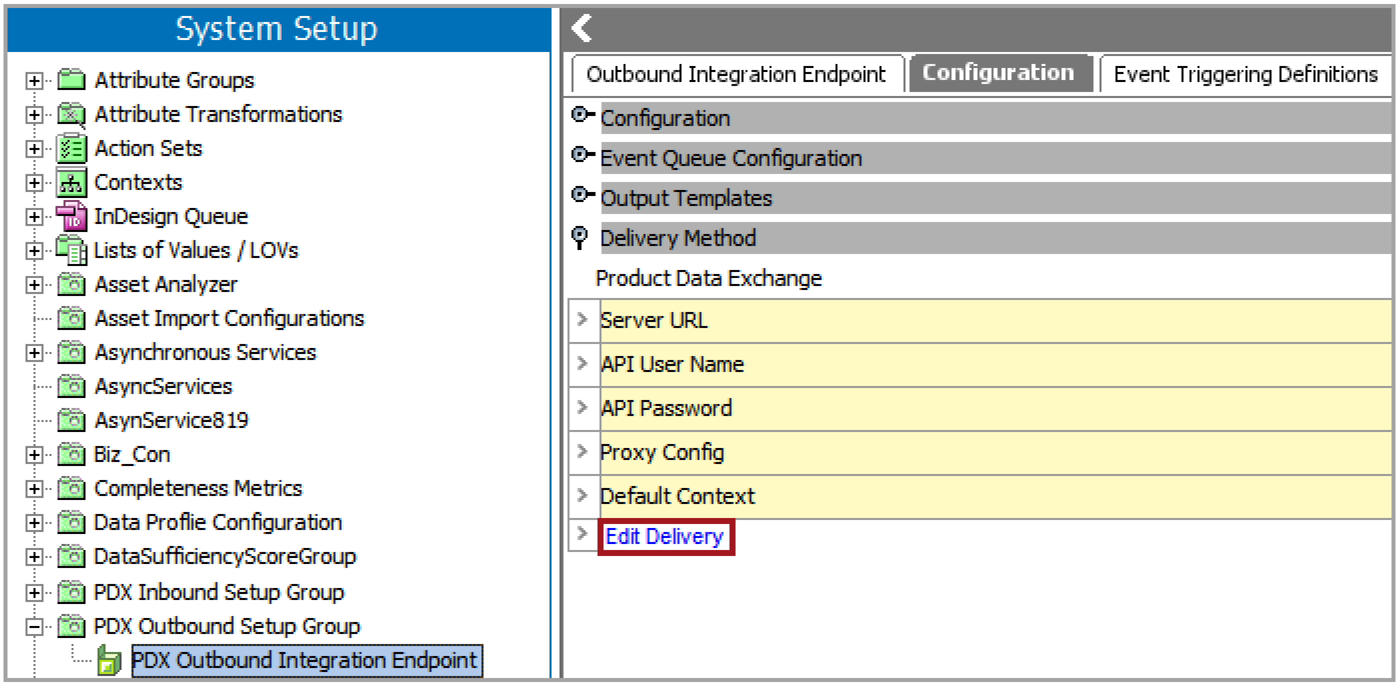
7. Click the ellipsis button (...) to display the 'Conditions for output template' dialog and modify the object types and event types to export to PDX. Click the **OK** button to update the output template.

   For more information, see the **Configure the Object Types and Event Types** section of the **OIEP - Event-Based - Output Templates Flipper** topic in the **Data Exchange** documentation.

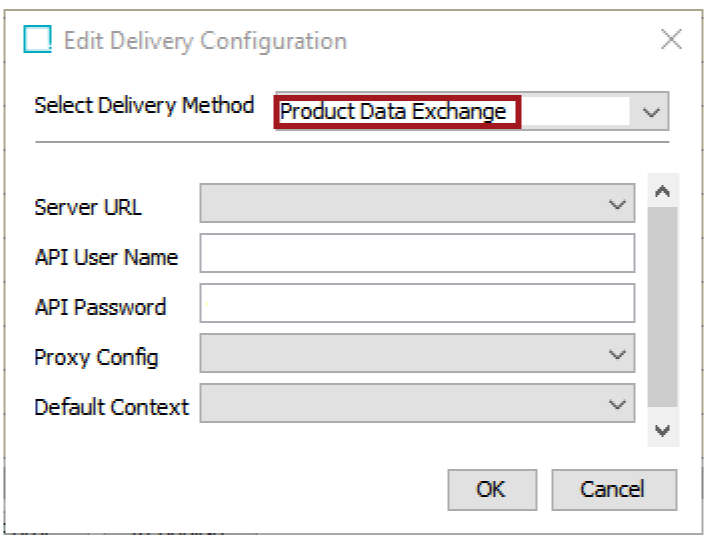


Note: The default Advanced STEPXML format template can be modified if necessary, as required to include product reference metadata, defined in the **Optional Product References Metadata Export Setup** section below.

8. Open the Delivery Method flipper and click the **Edit Delivery** link.

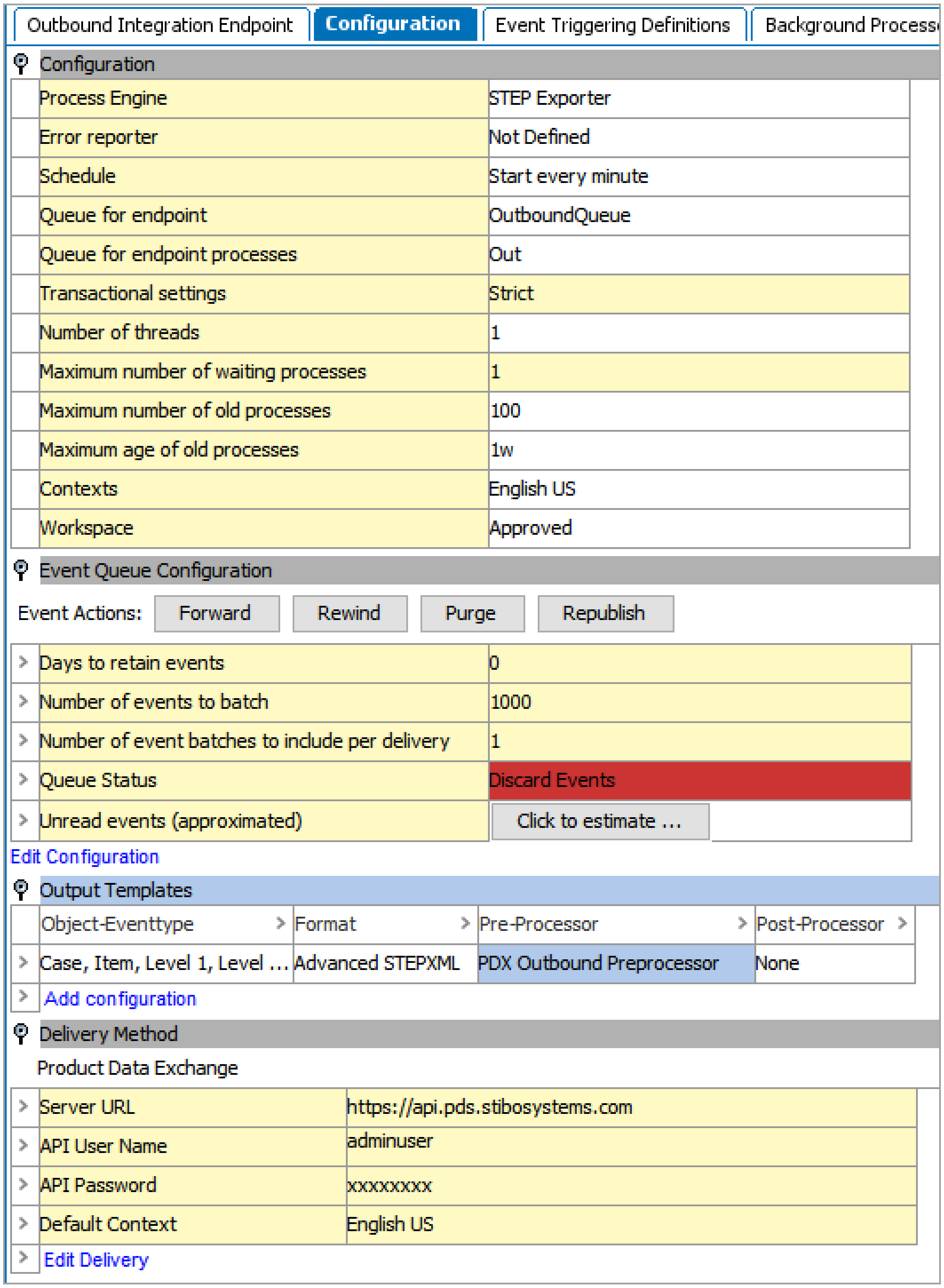


9. On the Edit Delivery Configuration dialog, configure the 'Product Data Exchange' delivery method to establish the connection from STEP to PDX. For details, see the **Product Data Exchange Delivery Method** topic in the **Data Exchange** documentation.

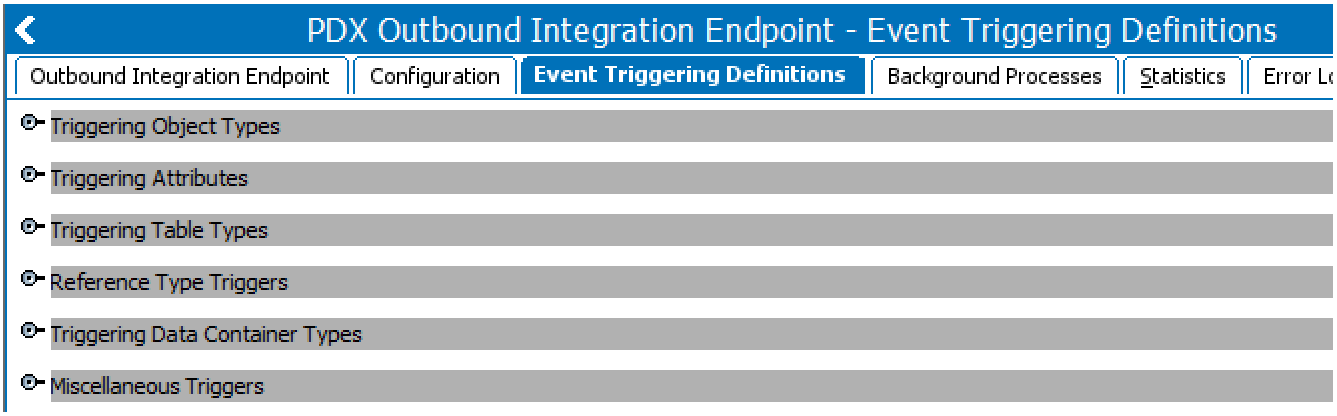


Updates made in the previous steps are displayed on the 'Configuration' tab, under the 'Configuration' flipper, 'Output Templates' flipper, and 'Delivery Method' flipper.

Note: The API User Name / API Password credentials for the PDX OIEP and PDX IIEP are not necessarily the same credentials that are needed to log into the PDX system. The API User Name and API Password credentials can be obtained by contacting your PDX representative.



10. On the 'Event Triggering Definitions' tab, select the necessary actions that should trigger events for this OIEP. For more information, see the **OIEP - Event-Based - Event Triggering Definitions Tab** topic in the **Data Exchange** documentation.



Note: When using either of the setups defined in the following sections, reference types must be set as triggers. For more information, see the **Packaging Hierarchy Information Export Setup** section or the **1WorldSync Composite Attribute Values Export Setup** section below.

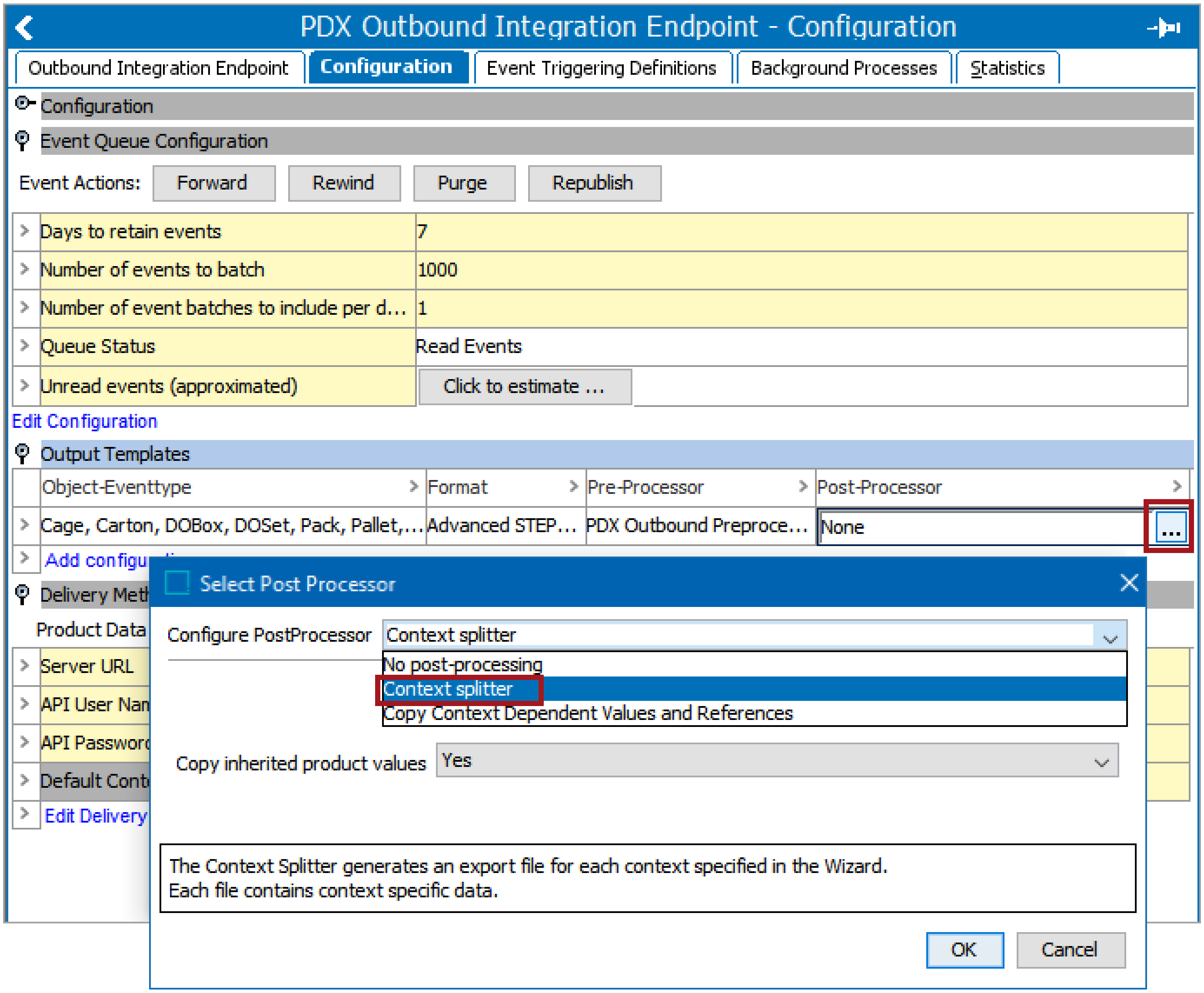
11. Enable the endpoint as described in the **Running an Outbound Integration Endpoint** topic of the **Data Exchange** documentation.

Multi-Language Exchange Configuration

Users that want to syndicate product master data from STEP to PDX in multiple languages can do so by selecting 'Context splitter' from the Post-Processor dropdown menu, located in the Output Templates parameter. The language is retrieved from the selected context; this language must have a locale configuration.

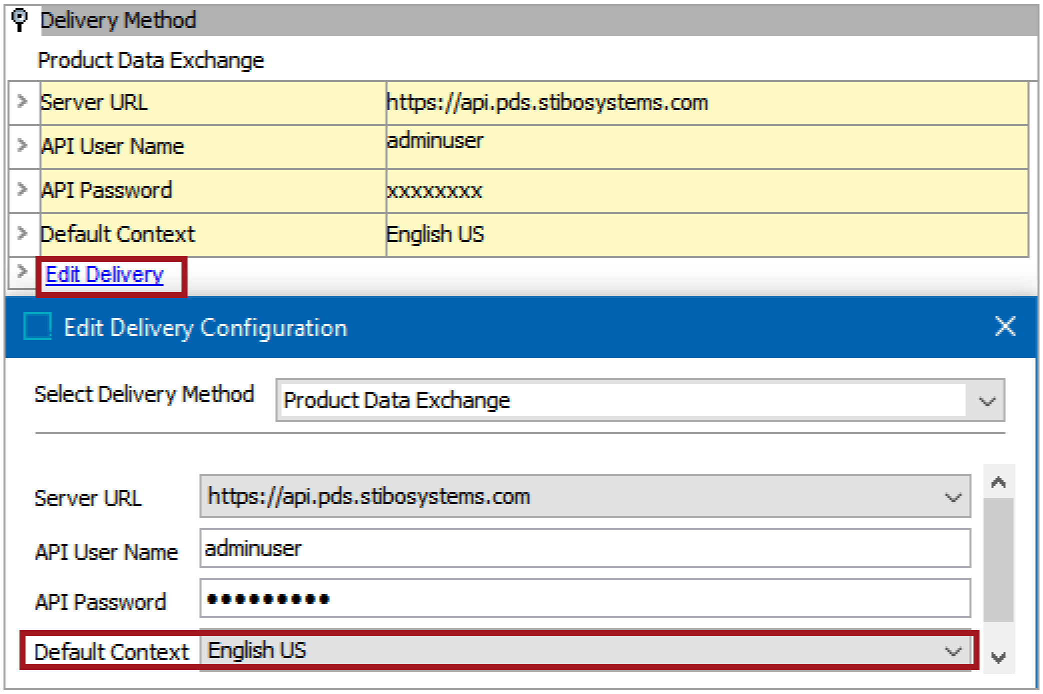
To configure the PDX Outbound Integration Endpoint for multi-language syndication:

1. Confirm that step 5 from the 'Required Manual Setup' section located earlier in this topic has been completed and more than one context has been selected.
2. Click the ellipsis button (...) to open the Select Post-Processor window, and from the dropdown menu, select "Context splitter." Users also have the ability to copy inherited product value.



3. In the Delivery Method parameter, click 'Edit Delivery' and choose a Default Context. In this example, the user has selected 'English'; this is the language send to PDX as the default language. All other contexts / languages not marked as the default context are exported following the default language.

> **Note:** Because PDX operates with English as the default language, it is recommended that users select an English - based language for the Default Context. If a language is selected for the Default Context that is not English - based (e.g., Spanish), then the output would display Spanish data for the English language in PDX.



Packaging Hierarchy Information Export Setup

The Packaging component model allows you to send products in a packaging hierarchy (e.g., pallet-to-case-to-pack) to PDX and also include this hierarchy in PDX. The activated and configured packaging reference types determine the packaging data exported to PDX.

If your products do not need the packaging hierarchy in PDX, this setup is not necessary.

Configuration

1. If necessary, complete the Packaging component setup as defined in **Configuring the Packaging Component in STEP Workbench** under the **Web User Interfaces** documentation.

2. Edit the PDX OIEP triggering definitions as follows to determine the STEP data activity that causes events to be sent to PDX. At a minimum, one Object Type trigger must be set before the OIEP can be enabled.

   On the Event Triggering Definitions tab, set triggers for the identified packaging reference types. For more information, see the **OIEP - Event-Based - Event Triggering Definitions Tab** topic in the **Data Exchange** documentation.

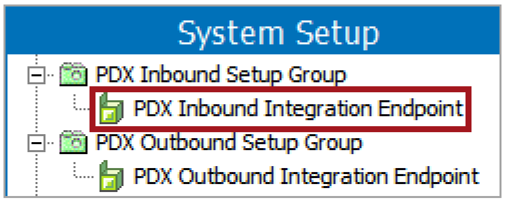
3. Enable and invoke the PDX OIEP (manually or on schedule) as described in the **Running an Outbound Integration Endpoint** topic of the **Data Exchange** documentation.

PDX Inbound Integration Endpoint Configuration

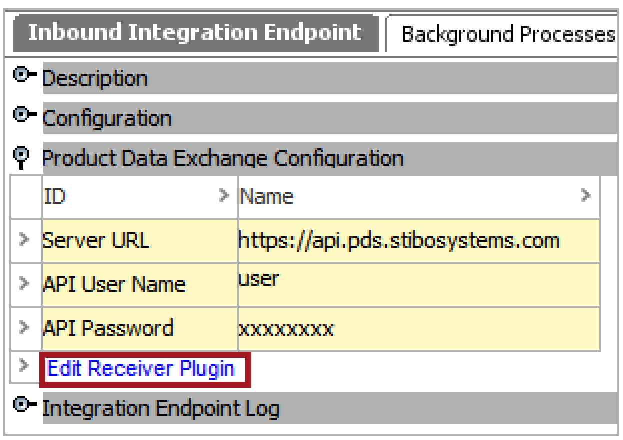
The PDX Inbound Integration Endpoint (IIEP) is used to receive the PDX status information for integrated products.

The PDX IIEP includes a number of preset parameters to reduce the setup required. However, the following updates are necessary for the IIEP to work with your data model.

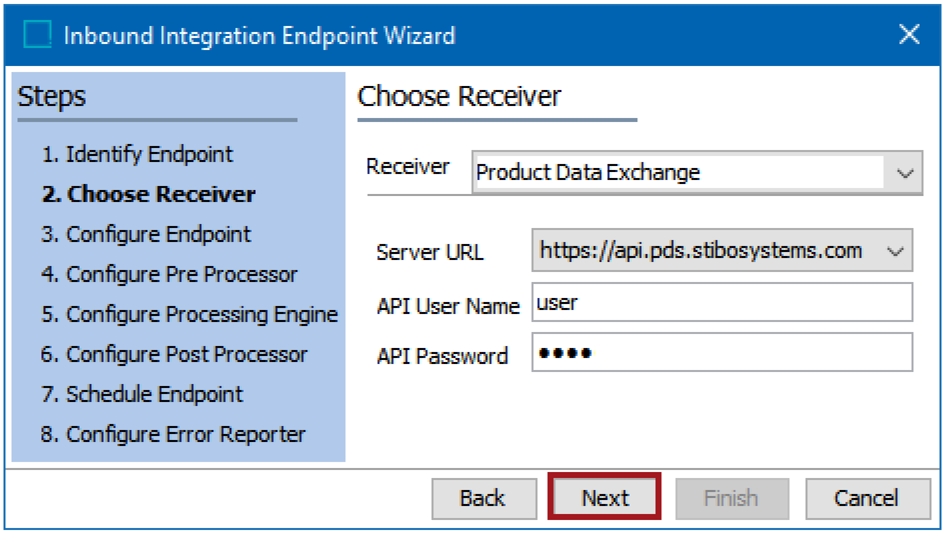
1. In the System Setup > PDX Inbound Setup Group node, open the **PDX Inbound Integration Endpoint**.



2. Display the 'Inbound Integration Endpoint' tab.
3. Open the 'Product Data Exchange Configuration' flipper and click the **Edit Receiver Plugin** link.



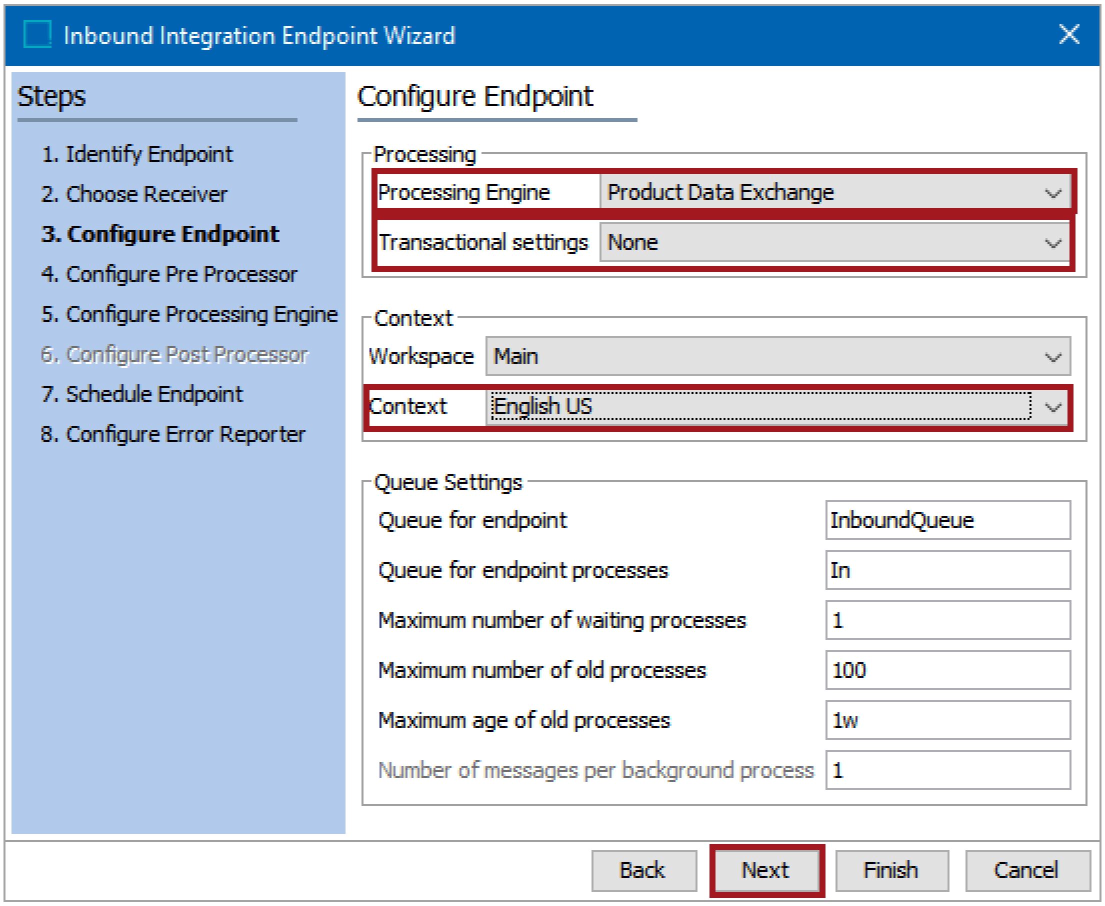
4. On the Inbound Integration Endpoint Wizard dialog, configure the 'Product Data Exchange' receiver option and click the **Next** button.



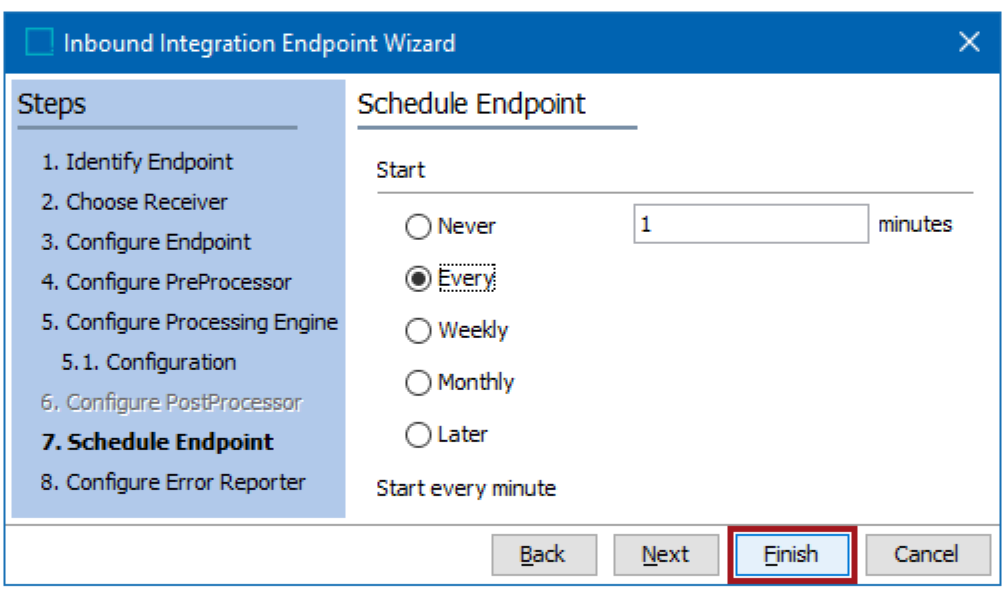
For details, see the **Product Data Exchange Receiver** topic in the **Data Exchange** documentation.

5. On the Inbound Integration Endpoint Wizard, update the following parameters:
    - Ensure the 'Processing Engine' parameter is set to **Product Data Exchange** and 'Transactional settings' is set to **None**.
    - Set 'Context' parameter as required for your inbound data. The Workspace must remain set to 'Main' since new data can only be added to this workspace. For more information, see the **IIEP - Configure Endpoint** topic in the **Data Exchange** documentation.
    - Click the **Next** button repeatedly until the Schedule Endpoint step displays.

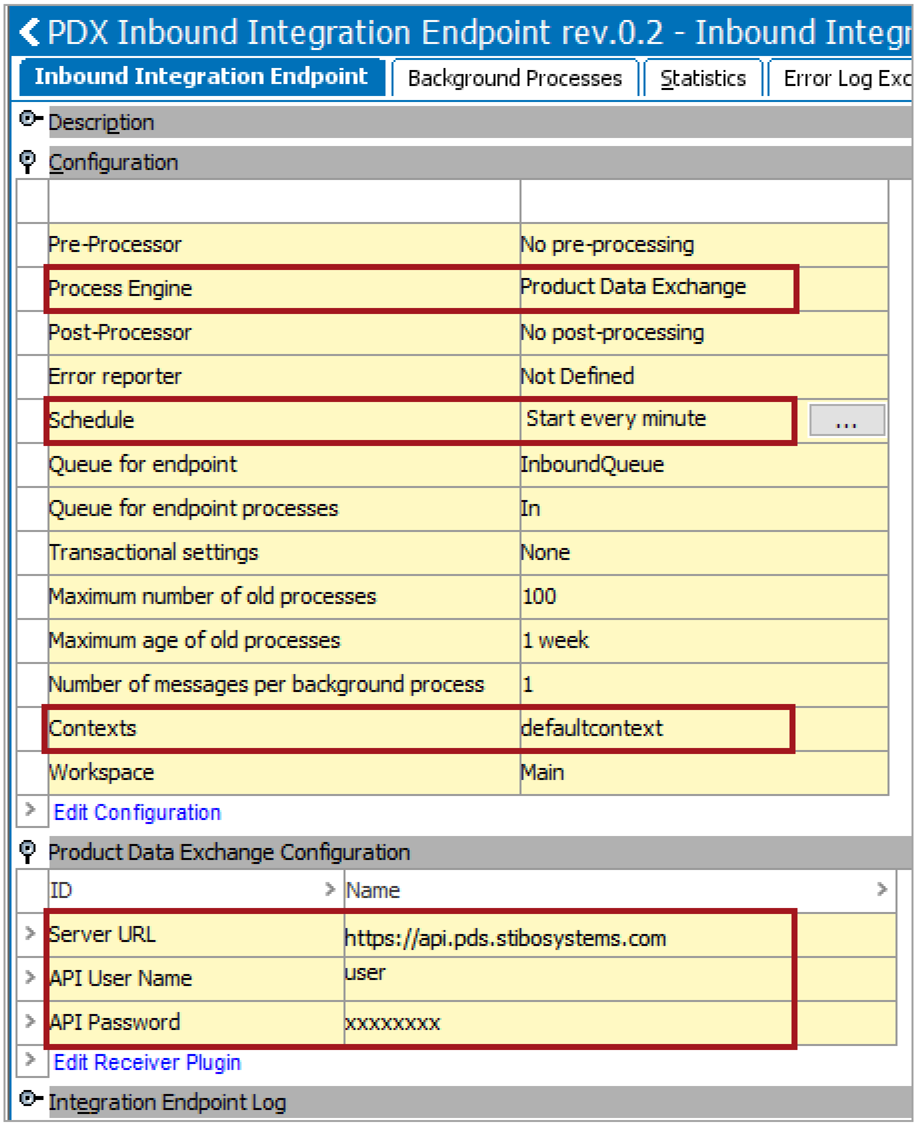
Important: Consider the time zone of the application server compared to that of the workbench (the client) where the schedule is created or viewed. When scheduling a job, the local time zone is displayed in the workbench, but the time zone of the server is used to run the background process. Although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



6. On the Schedule Endpoint step, set the desired schedule and click the **Finish** button.



Updates made in the previous steps are displayed in the 'Configuration' and 'Product Data Exchange Configuration' flippers.



7. Enable the endpoint as described in the **Running an Inbound Integration Endpoint** topic of the **Data Exchange** documentation.

After successful setup, and invocation, PDX product status information can be seen on the Data Containers tab of the product editor.

Referenced Object Metadata as PDX Composite Attributes

Using specific tags within Advanced STEPXML as defined on the PDX Outbound Integration Endpoint (OIEP), users can export metadata on product, image and document, and entity reference types from STEP to PDX. Once exported, reference metadata values will display as part of a composite attribute in PDX. With this additional information, users within PDX are able to access and utilize more detailed information regarding references of products exported to PDX.

Note: Although a product reference type is used in the example below, the setup in workbench will be the same regardless of whether the user is exporting product, image and document, and/or entity reference types metadata to PDX. However, the STEPXML tags will differ, depending on the reference type. For more information regarding tags in STEPXML, see the **STEPXML Tags and Examples** topic in the **STEPXML** documentation.

Prerequisites

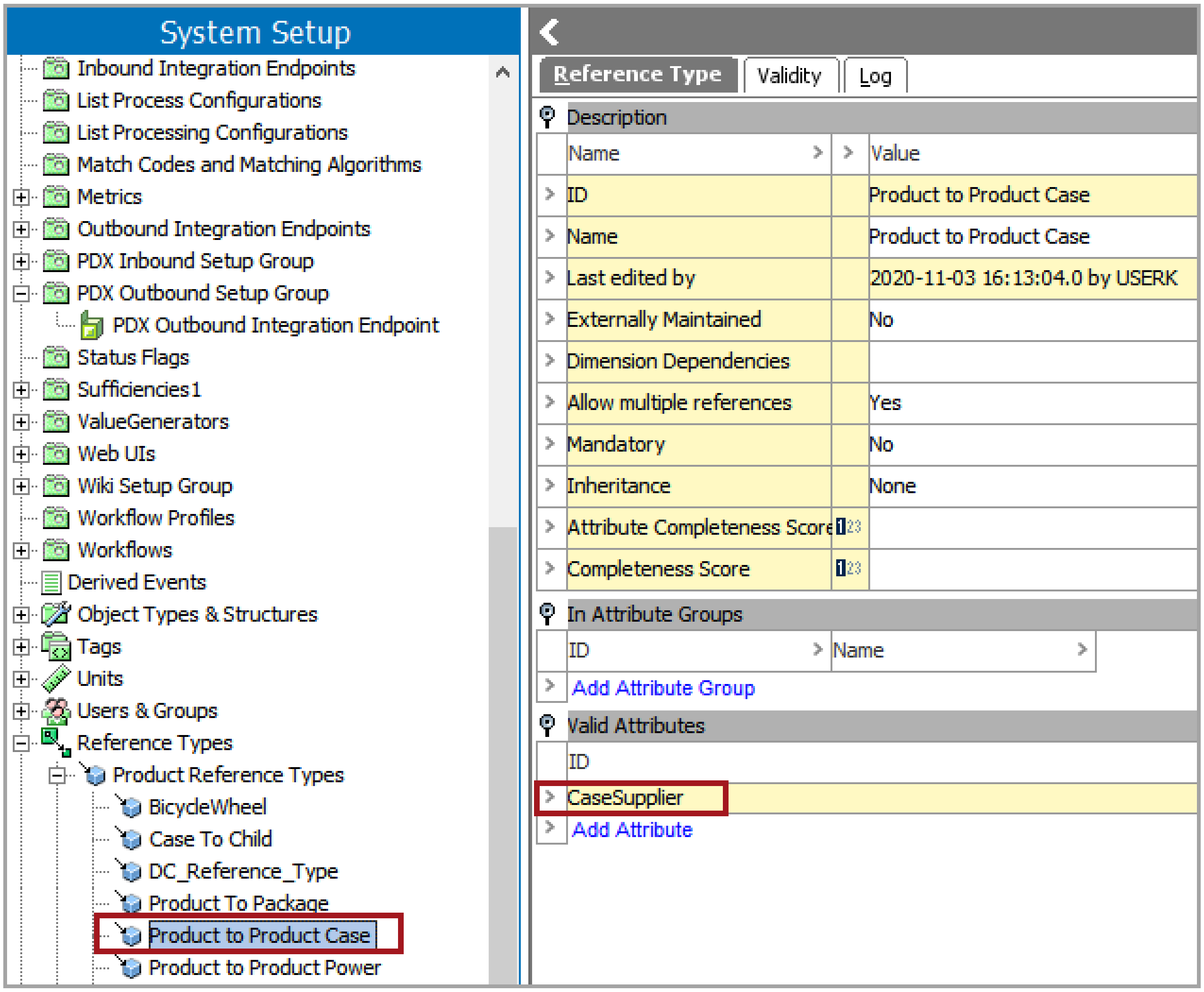
While this topic will provide an overview of the steps /configurations required to export metadata on various references types, it is expected that anyone exporting metadata from references to PDX have an overall understanding of the following topics:

- References and Link Types
- Product Data Exchange
- Advanced STEPXML Format

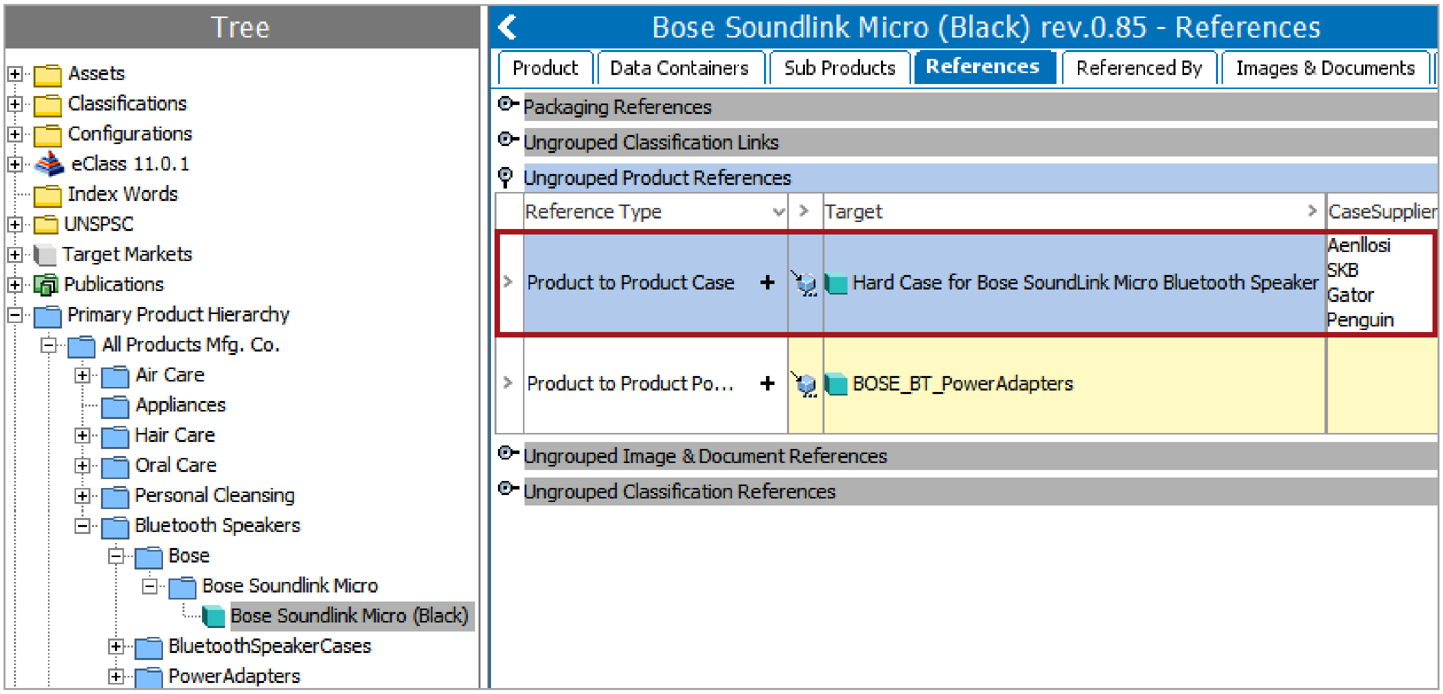
Exporting Metadata on Reference Types

Users can export metadata from reference types to PDX as part of a composite attribute by including specific STEPXML tags within the Advanced STEPXML format. Before exporting metadata on a reference type to PDX, a reference type (in this example, a product reference type) must first be configured.

1. Create a product reference type. In this example, the user has created a product reference type with the ID and Name 'Product to Product Case.' Note the highlighted valid attribute 'CaseSupplier.' The 'CaseSupplier' attribute will contain the metadata on the product reference type.



2. Make the newly created product reference type 'Product to Product Case' valid on the product you would like to export to PDX. In the example below, the user has made the 'Product to Product Case' reference type, with a target of 'Hard Case for Bose SoundLink Micro Bluetooth Speaker,' valid for the product 'Bose Soundlink Micro (Black).' The attribute 'Case Supplier' has four metadata values (Aenllosi, SKB, Gator, Penguin) for this reference. The goal is to make these metadata values part of a composite attribute within PDX once the 'Bose Soundlink Micro (Black)' is exported to PDX.



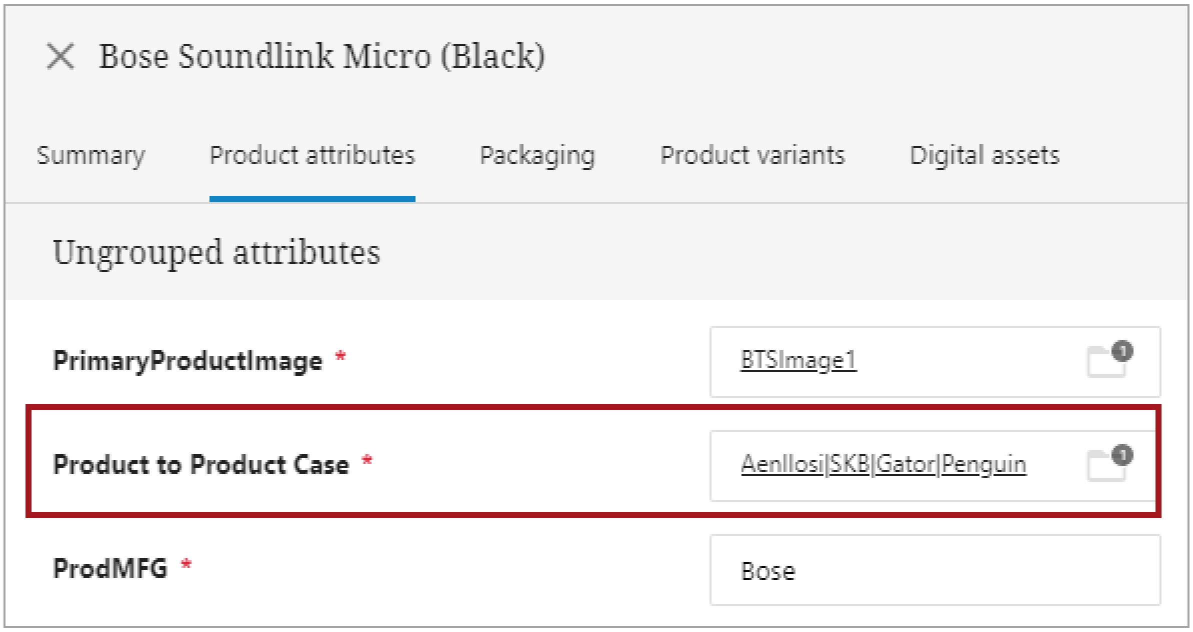
3. In the Advanced STEPXML format template, include the following tags, as shown in the example below.

> **Note:** The example presented in Step three is only a portion of the Advanced STEPXML template, and not the entire template.

4. Using the PDX OIEP, export the selected product to PDX.

   The product 'Bose Soundlink Micro (Black),' along with the metadata on the 'Product to Product Case' reference type, has been exported to PDX.

   The image directly below shows how the composite attribute would be presented in PDX. Note that the title of the composite attribute is titled 'Product to Product Case,' based on the ID of the product reference type.



For more information on the PDX Outbound Integration Endpoint, see the **PDX Outbound Integration Endpoint Configuration** topic.

Assigning an Alternative Title for Reference Types and Metadata Attributes within PDX

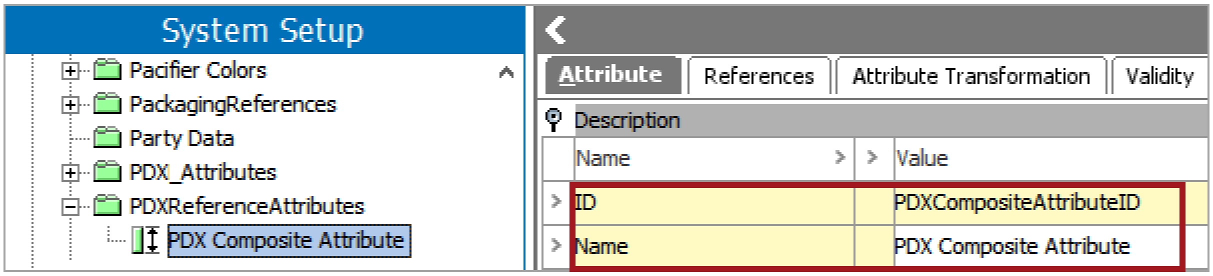
By default, composite attributes made up of reference type metadata are titled based on the ID of the reference type that contains the metadata, i.e., in the example above, the title of the composite attribute in PDX would be the ID of the reference type, 'Product to Product Case.' However, users may want the composite attribute that contains the reference type metadata to have a different title. This can be achieved by assigning the 'PDX Composite Attribute' attribute to the reference type.

The following configuration steps must be performed to set up and use the 'PDX Composite Attribute ID' attribute:

- First, a description attribute must be created. The ID of this attribute must be added to the **PDSDelivery.CompositeAttributeID=[attribute_id]** property in the sharedconfig.properties file on your STEP application server. For example:

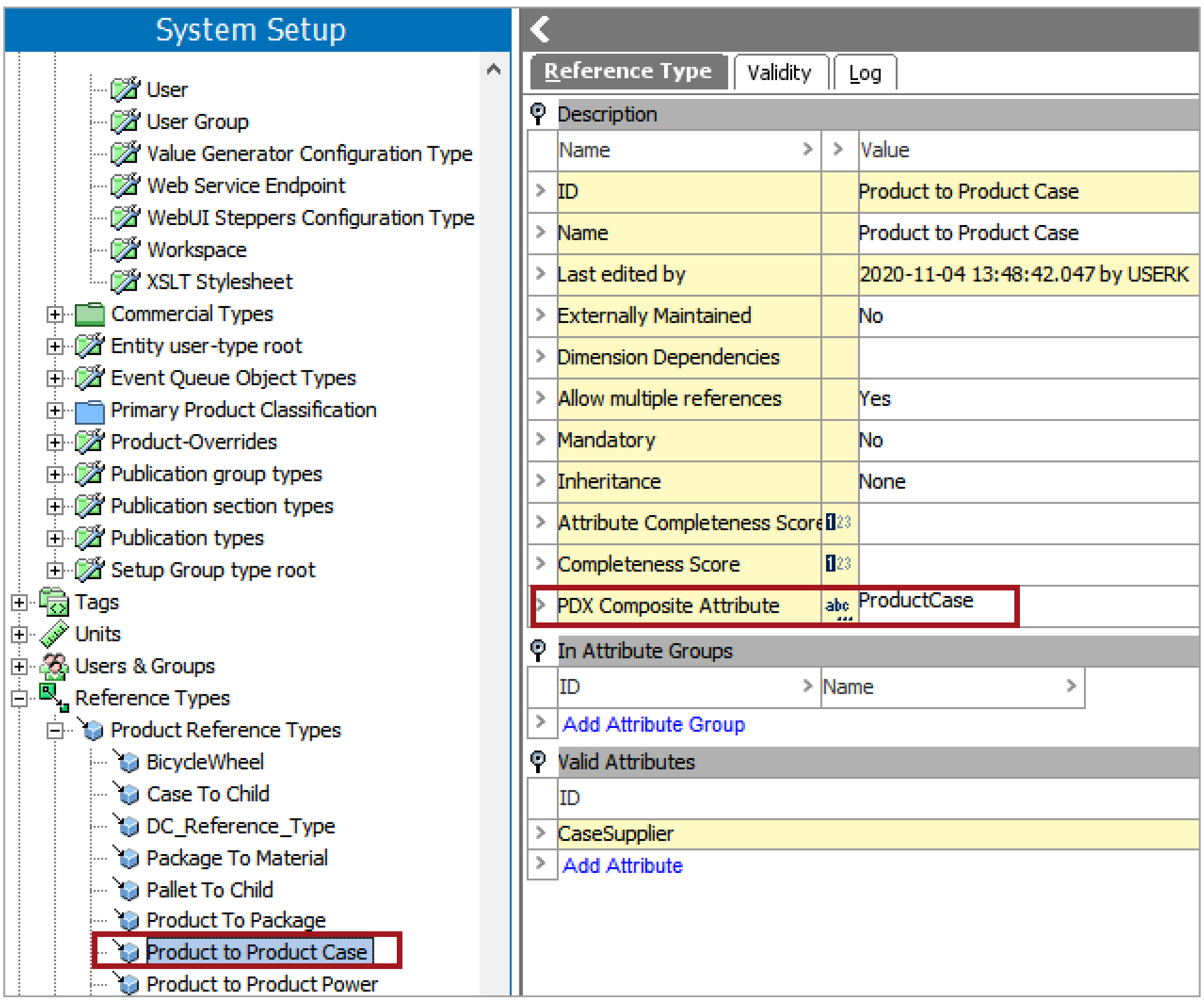
  ```
  PDSDelivery.CompositeAttributeID=PDXCompositeAttributeID
  ```

  The 'PDX Composite Attribute' attribute can be given any ID or name and can be created anywhere in your system. In this example, the name of the attribute is 'PDX Composite Attribute' and the ID of the attribute is 'PDXCompositeAttributeID' and has been created within an attribute group named 'PDXReferenceAttributes.'

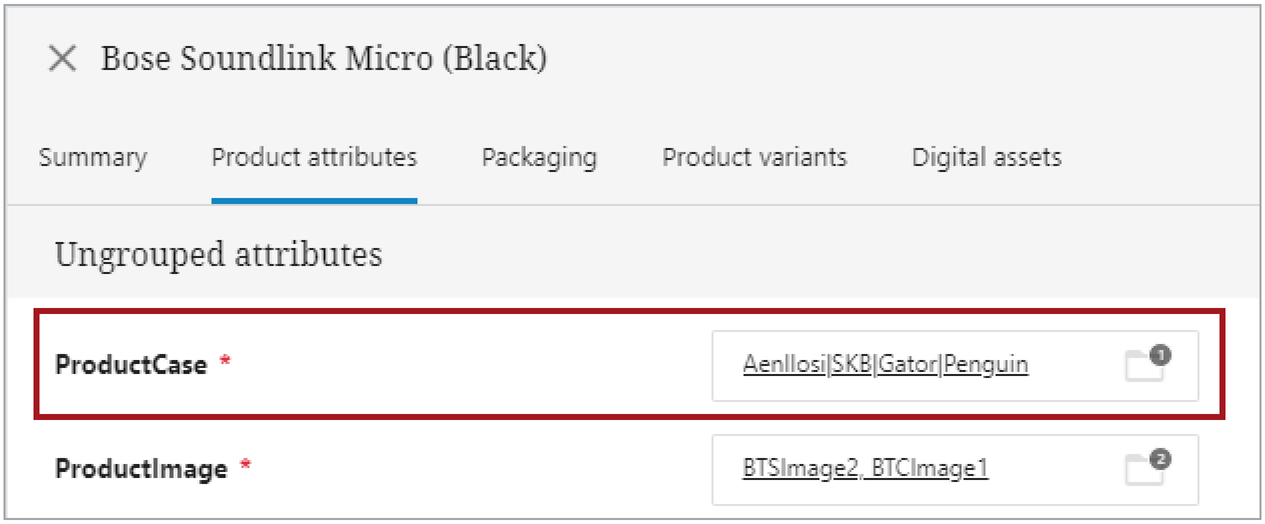


- The attribute must be made valid on the Reference-Type (Reference-Type user-type root) object type, which is located in System Setup under Object Types & Structures > Basic Object Types.

Once the bulleted items listed above are implemented, create a value for the 'PDX Composite Attribute' attribute, which will now be included as a description attribute within the 'Product to Product Case' reference type. In the example below, the user has assigned the value 'ProductCase' to the 'PDX Composite Attribute.'



With the 'PDX Composite Attribute' now assigned a value, when the 'Bose Soundlink Micro' (Black) product is exported to PDX, the composite attribute on the 'Product to Product Case' reference will be titled 'ProductCase,' based on the value of the 'PDX Composite Attribute' attribute.



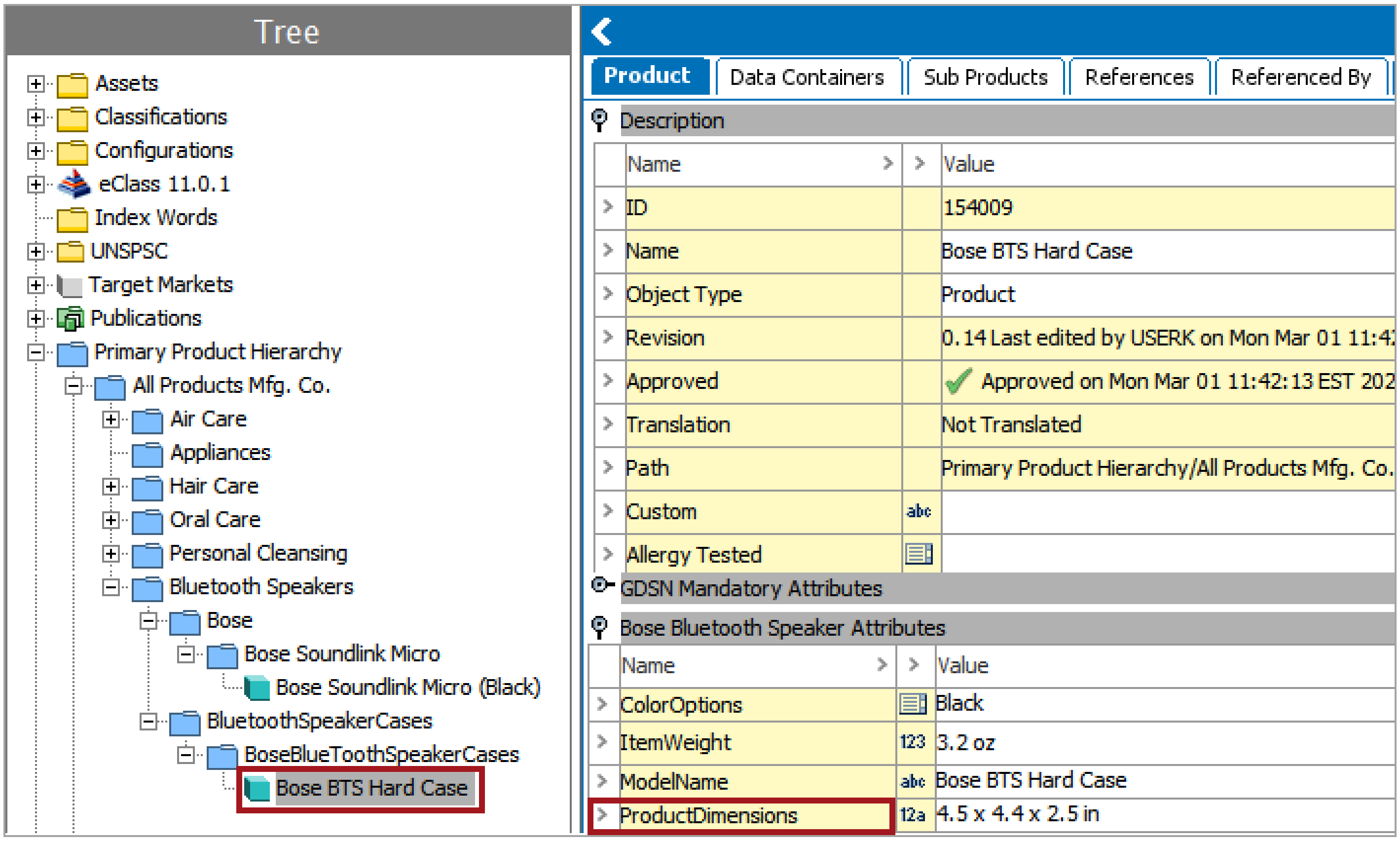
For more information on the PDX Status Attribute Group attributes, see the **PDX Channel Status Monitoring** section of this documentation. For more information on creating attributes, see the **Creating Attributes** topic in the **System Setup** documentation.

Assigning an Alternative Title for an Attribute that is Part of a PDX Composite Attribute

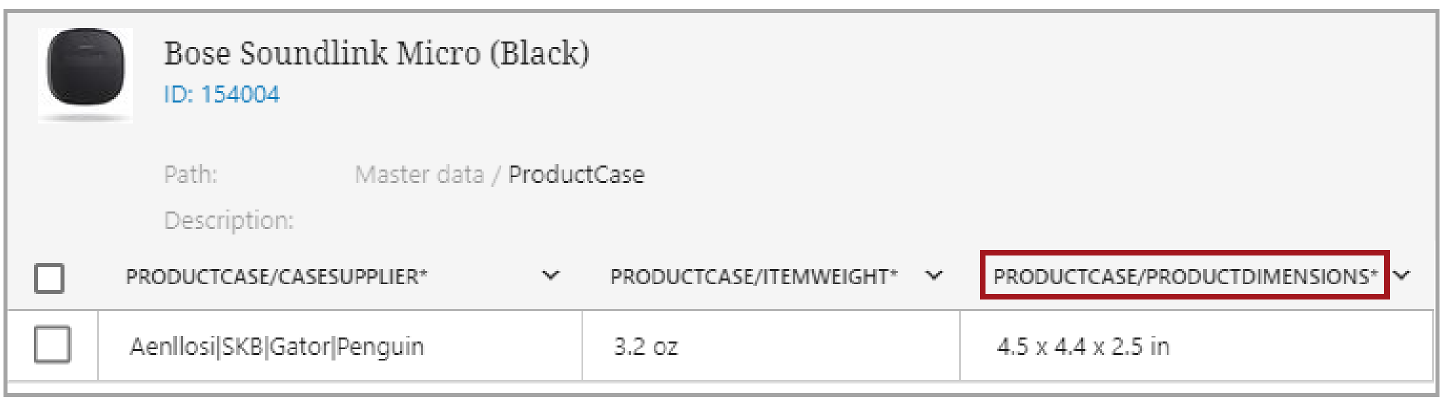
Just as the 'PDX Composite Attribute' attribute can be used to create an alternate title for a reference type that is exported to PDX, it can also be used to create an alternative ID for an attribute that is part of a composite attribute.

Note: To assign an alternative title for an attribute that is part of a PDX Composite Attribute, the attribute must be made valid on the Attribute (stibo.normalattribute) object type.

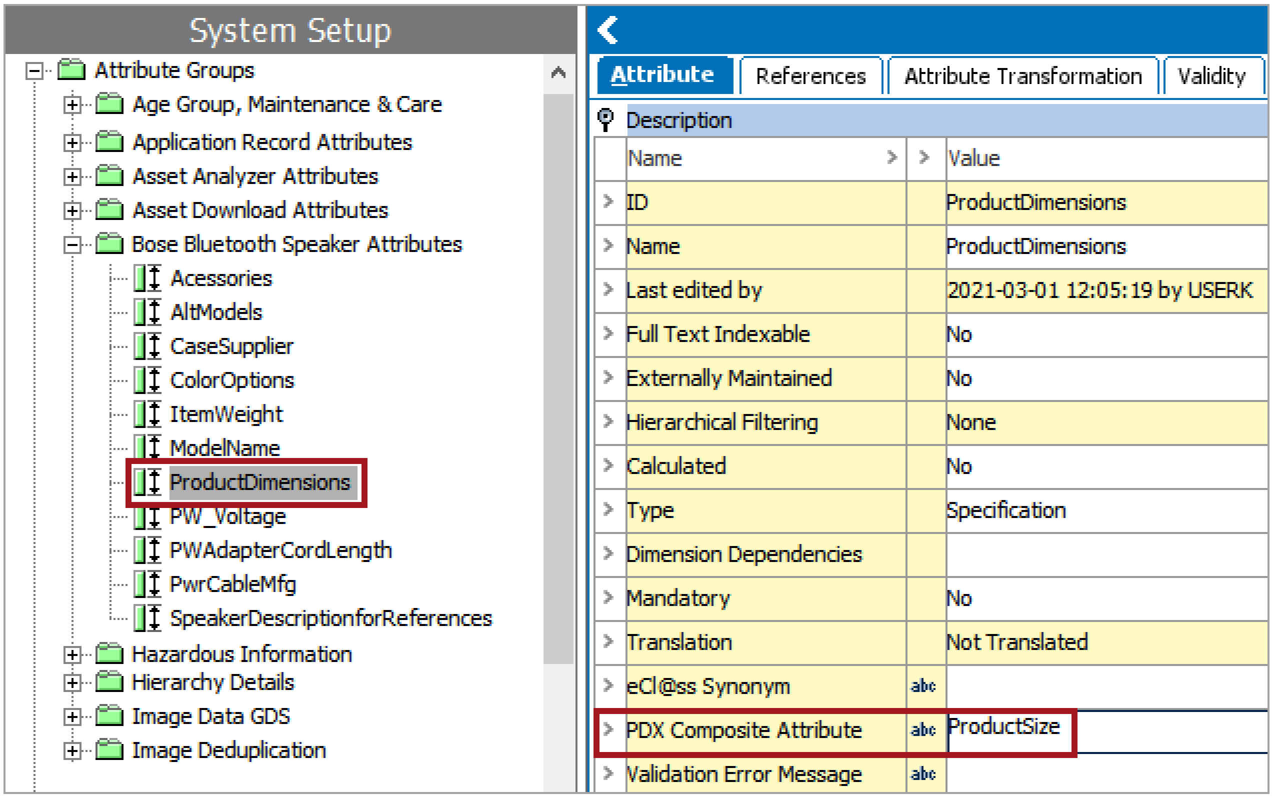
As an example, the image below shows the product 'Bose BTS Hard Case' with 'ProductDimensions' as one of its attributes:



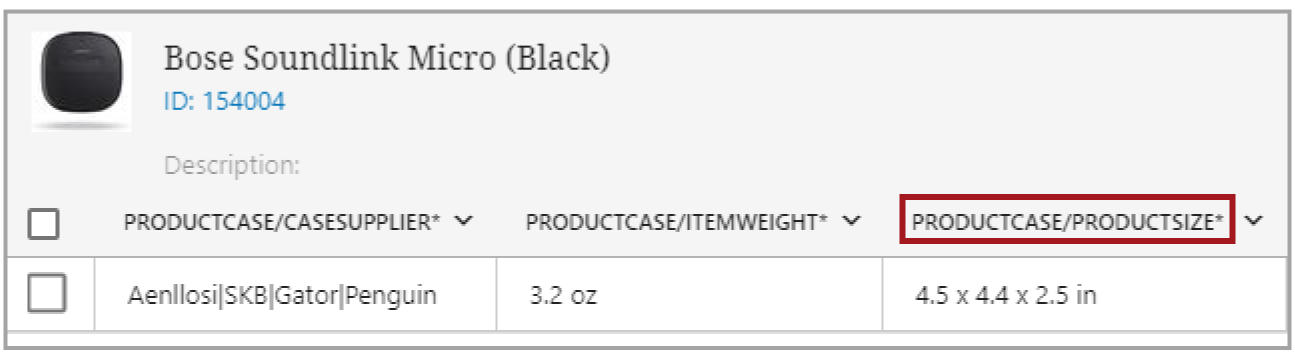
Because the 'Bose BTS Hard Case' product is referenced by the 'Bose Soundlink Micro (Black)' product using the 'Product to Product Case' reference type (as detailed earlier in this topic), when the Bose Soundlink Micro (Black) is exported to PDX, the 'ProductDimensions' attribute will be displayed as such:



However, if the user needed the title of the attribute to be something different than the ID of the attribute itself, a value could be assigned to the 'PDX Composite Attribute' attribute, as shown in the example below:



With the ' PDX Composite Attribute' attribute assigned a value (in this case, 'ProductSize'), the attribute, once exported along with the product, would instead be displayed as shown in the image below:



The 'ProductDimensions' attribute is now identified within PDX by the value assigned to the 'PDX Composite Attribute' attribute, 'ProductSize.'

For information on how to make an attribute valid on the Attribute object type and the Reference object type, see the **Attribute Metadata on Attributes** topic in the **Attributes** section of the **System Setup** / **Super User Guide** documentation. The steps for making an attribute valid on the Reference object type are the same as those used for making an attribute valid on the Attribute object type.

For information regarding Reference and Link Types, see the **Reference and Link Types** topic in the **System Setup / Super User Guide** documentation.

For more information regarding Advanced STEPXML, see the **Advanced STEPXML Format** topic in the **Data Formats** documentation.

PDX Channel Status Monitoring

Status information related to products sent to PDX via STEP, as well as products added / submitted to channels in PDX, can be viewed and monitored in the **PDX Status Data Container** in the workbench and within the **PDX Channel Status** component in the Web UI.

The PDX Status Data Container is located within the **PDX Status Attribute Group**, which also contains the PDX status attributes that are valid for the PDX Status Data Container. These attributes are used to monitor the send and receive status of products between STEP and PDX, including the relevant PDX channel statuses. The data container and attributes are created automatically when the PDX integration is activated, and rows within the data container are added as products are submitted to PDX from STEP and from PDX to channels.

Note: While the PDX Data Container within the workbench and the PDX Channel Status component in the Web UI may be referred to separately in this documentation section, details will be relevant for both unless otherwise noted, as function and output are similar.

This documentation section includes the following topics related to the monitoring of PDX channel statuses:

- PDX Status Data Container Configuration
- PDX Channel Status Web UI Component
- PDX Channel Status Monitoring in Web UI

Note: This documentation section assumes that users have a basic understanding of data containers. For more information on data containers, see the **Data Containers** section of the **System Setup** / **Super User Guide** documentation.

PDX Status Data Container Configuration

The PDX Status Data Container displays information relevant to:

- Products that are submitted from STEP to the master data list in PDX
- Products that are submitted from PDX to specific stand-alone channels (e.g., Amazon, Walmart), and group channels (e.g., 1WorldSync)
- Acceptance / rejection statuses received back from the channels within PDX

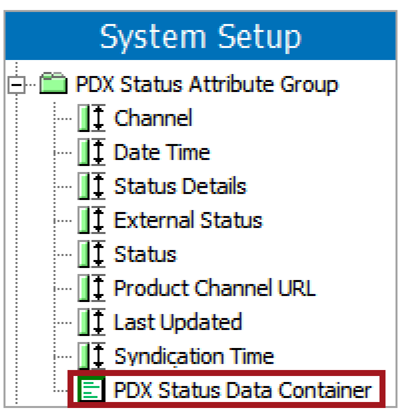
This topic provides information on:

- How to configure the PDX Status Data Container in STEP Workbench
- The attributes that make up the PDX Status Data Container
- How the data container functions when products are submitted to PDX and when status messages are returned from PDX

Configuring the PDX Status Data Container

The PDX Status Data Container is created automatically when the PDX integration is activated on your STEP system. To configure it for initial use, follow these steps:

1. In System Setup, open the 'Attribute Groups' node, select 'PDX Status Attribute Group' and click 'PDX Status Data Container.'



2. On the Validity tab, make the data container valid for the product object types that will be integrated with PDX.
3. Optionally, restrict the data container to specified hierarchies. Select the 'Data Container Types' tab, click the 'Restriction' parameter and set it to 'Validity restricted to hierarchies.' On the 'References' tab, select the required classification or product hierarchies.

For more information regarding configuring data containers, see the **Setting Up Data Container Types in Workbench** topic in the **Data Containers** section of the **System Setup / Super User Guide** documentation.

Viewing PDX Status Data Containers in the Workbench

In the product editor, valid object types display a 'Data Containers' tab and contain a flipper for the PDX Status Container. No data containers will display beneath the flipper until the product is initially submitted to PDX.

The example below displays a PDX Status Data Container in the workbench that includes pertinent channel names, date and times of product submission, status of products, the URL where the specific product is located, and any relevant status messages. The ID column contains the ID of the data container, which does not display in the Web UI PDX Channel Status component.

Product | **Data Containers** | References | Referenced By | Images & Documents | Commercial | Tables | Proof View | Status | State Log | Tasks

HazardWarningSufficiency

MultiDataContainer

PDX Status Data Container

ID	Channel	Date Time	External Status	Product Channel URL	Status	Status Details
0a64385b-45...	Canada	2020-01-03 15:05:02		https://pds-qa.stibosystems.com/#/products/143801/channels/1worldsync-channel#CA	Progress	
0bb681b9-71...	Pierre's Bistro	2020-01-10 10:40:10	GDSN Status: Registration sent	https://pds-qa.stibosystems.com/#/products/143801/channels/1worldsync-pool-channel#FR#1597930042136	Submitted	
186f3b9b-8c...	Franklin's	2020-01-10 10:40:10	GDSN Status: Registration sent	https://pds-qa.stibosystems.com/#/products/143801/channels/1worldsync-pool-channel#US#4531620868238	Submitted	
1bb38e15-f8...	1WorldSync Pre-Prod	2020-01-10 10:40:10		https://pds-qa.stibosystems.com/#/products/143801/channels/1worldsync-pool-channel	Progress	
2b84ad93-aa...	Bozzutos	2020-01-08 10:39:43	GDSN Status: Registration sent	https://pds-qa.stibosystems.com/#/products/143801/channels/1worldsync-channel#DK#1100001001118	Submitted	
5e78d896-49f...	Added to PDS	2020-01-03 14:24:07				
8895b8f8-806...	Amazon	2020-01-03 14:55:19		https://pds-qa.stibosystems.com/#/products/143801/channels/amazon-afs-channel	Rejected	
8bd4419a-b9...	Jungle Jim's	2020-01-10 10:40:10	GDSN Status: Registration sent	https://pds-qa.stibosystems.com/#/products/143801/channels/1worldsync-pool-channel#US#9643296662920	Submitted	
9bb89c10-37...	ACME	2020-01-09 13:52:37		https://pds-qa.stibosystems.com/#/products/143801/channels/acme-channel	Accepted	
bb3aa1af-1b...	United Kingdom	2020-01-10 10:40:10		https://pds-qa.stibosystems.com/#/products/143801/channels/1worldsync-pool-channel#GB	Multiple	

For more information about the functionality of PDX Status Data Containers, see the 'Functionality of the PDX Data Container' subsection below.

PDX Status Data Container Attributes

The attributes found in the PDX Status Data Container are the same attributes contained within the PDX Status Attribute Group. The PDX Status Container comes pre-configured with these attributes. Their descriptions and functions are as follows:

- **Channel**: Indicates the name of the channel in which an action has taken place. When a product is initially submitted to PDX, the Channel field does not display the name of a channel but instead contains the value 'Added to PDX.' In the Web UI PDX Channel Status component, the Channel attribute also contains the link to the product URL. For further information regarding the product URL link, see the **PDX Channel Status Web UI Component** topic.
- **Date Time**: Specifies when an action (such as a product submission to channel) has occurred. By default, the time and date is displayed in the ISO format.
- **Status Details**: When applicable, displays pertinent and additional information related to an External Status message e.g., Catalog Item Confirmation (CIC) message details.

- **External Status**: Indicates status of products submitted to a channel. These messages are generally channel-specific. Depending on the channel, a higher level of detailed information may be displayed than the information presented in the Status attribute text field.
- **Status**: Indicates both status of a product added and submitted to a channel. This is the same channel status that can be seen within PDX.
- **Product Channel URL**: Displays the URL of the product in PDX. In the workbench, the URL can be copied and then pasted into a browser to access the specific product's location. In the Web UI the Product Channel URL attribute can be added to the PDX Channel Status component, but is not necessary as a link is automatically provided in the Channel attribute.

Functionality of the PDX Data Container

The PDX Status Data Container is empty until a product is submitted to PDX. Since the PDX Status Data Container is configured to allow multiple data containers, multiple data containers will be added as products flow back and forth between STEP and PDX and their statuses are updated.

New rows are added to the data container when:

- A product is sent from STEP Workbench to PDX via the PDX Outbound Integration Endpoint.
- A product is added to a PDX channel.

Rows are updated within the data container when a change of status is received from PDX via the PDX Inbound Integration Endpoint. Generally, these updates are due to a product being submitted to a channel within PDX or messages from channels, e.g., a submitted or rejected product message.

For more information on the PDX Outbound Integration Endpoint, see the **PDX Outbound Integration Endpoint Configuration** topic.

For more information on the PDX Inbound Integration Endpoint, see the **PDX Inbound Integration Endpoint Configuration** topic.

Important: Adding a product to a channel and submitting a product to a channel are two different actions. Adding a product to a channel is when the product within the master data list in PDX is added to a channel. Submitting a product to a channel is when a product that has already been added to a channel within PDX is submitted to a channel outside of PDX, meaning product data is being sent out of PDX to the receiving systems.

Viewing PDX Status Data Containers in the Web UI

Though initially configured in STEP Workbench, end users will typically view the contents of the PDX Status Data Container in the Web UI, using the 'PDX Channel Status' component. For more information on the configuration of this component and how it is used to monitor PDX channel status, see the **PDX Channel Status Web UI Component** topic.

PDX Channel Status Web UI Component

The PDX Channel Status Web UI component is used to track the status of products as they are submitted between STEP, PDX, and channels into which PDX sends data. Users that have a PDX integration in their STEP system have access to the PDX Channel Status in the Web UI. Provided users are also logged into their PDX system, users can directly access a given product via a link located in the PDX Channel Status Web UI component.

Note: To access and use the PDX Channel Status Web UI component, the ‘pds-channel-status’ add-on component must be activated on your system. See your Stibo Systems representative for more information.

Note: This topic assumes that you are familiar with the Web UI Designer and basic Web UI component configurations. For more information, see the **Design Mode Basics** topic in the **Web UI Getting Started** documentation.

Configuring the PDX Channel Status Web UI Component

The PDX Channel Status Web UI component can be added to the Web UI as a component on a Node Details screen or a row within a Node Editor.

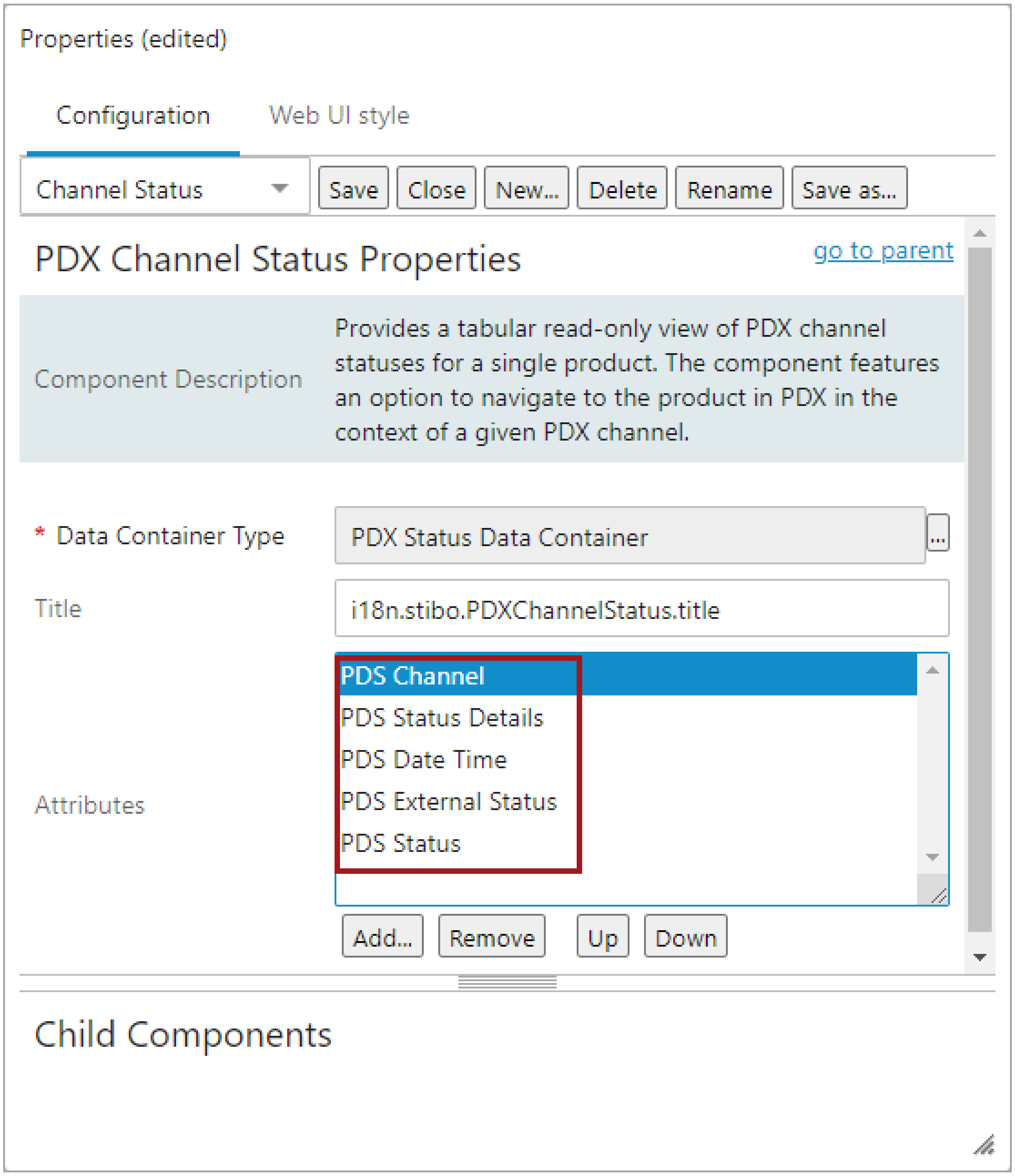
For more information on adding a row to a Node Editor, see the **Node Editor Component** topic in the **Using a Web UI** documentation.

For more information on adding a component to a Node Details screen, see the **Node Details Screen** topic in the **Using a Web UI** documentation.

The PDX Channel Status Properties window comes pre-configured with the Data Container Type, Title, and Attributes necessary to use the PDX Channel Status component. No other configuration is necessary.

It is important to note that a title value entered for the PDX Channel Status component, either default or custom, will only display when added as a row in a Node Editor.

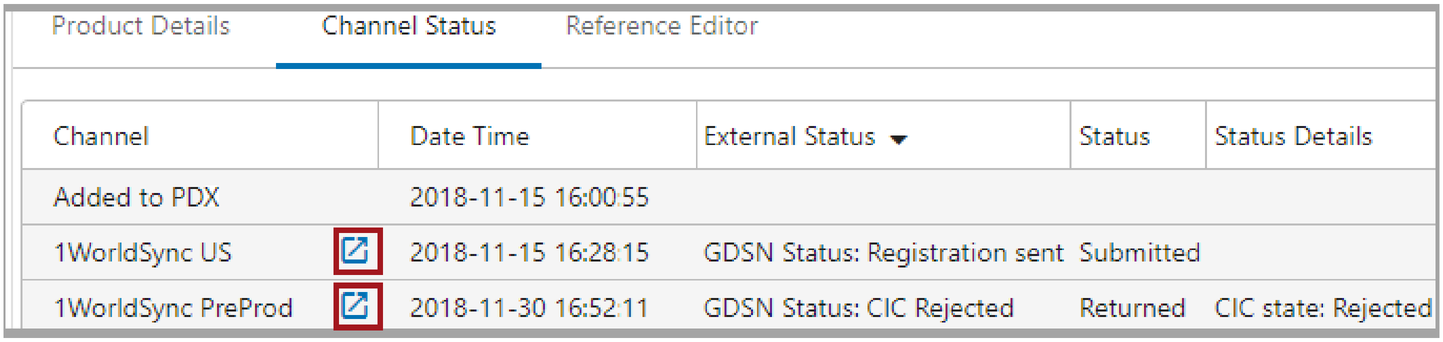
Note: Before entering a custom title, it is important to know the title parameter offers i18n key functionality. Presence of an i18n key indicates that the field can be included in extraction for external translation, and that a value has not been manually populated. Once a value has been manually populated within the designer, it is no longer available for extraction unless the manual value is removed and the configuration saved, closed, and then reopened. Customers planning to pursue or utilize Web UI translations should not overwrite i18n values manually in the designer, and should instead populate texts within the Web UI locale files on the application server. For more information, see the **Localization** topic within the **Administration Portal** documentation.



Note: Although the attribute 'Product Channel URL' can be added to the component, it is not necessary as the link for the URL is already included in the 'PDX Channel' column. It is recommended that the pre-configured attributes are left as-is for optimal performance.

PDX Channel Status Web UI Component Link

The PDX Channel Status Web UI component includes a link to products that have been submitted to a channel within the PDX system. Users that are also logged into their PDX system will be redirected to the relevant product in the context of the channel when clicking the provided link.



In the example above, the relevant product has been added to PDX, and within the PDX system, submitted to the 1WorldSync US and the 1WorldSyncPreProd channels. Clicking the URL link button located within the Channel row (the blue box with upward arrow, highlighted in the graphic) opens the product in the channel connected to that link.

For more information regarding the functionality of the PDX Channel Status Web UI component, see the next topic in this documentation section, **PDX Channel Status Monitoring in Web UI**.

PDX Channel Status Monitoring in Web UI

The PDX Channel Status Web UI component can be used to monitor the status of products within the Web UI as they are submitted between STEP, PDX, channels, and when applicable, specific markets and receivers within channels.

Note: To access and use the PDX Channel Status Web UI component, the ‘pds-channel-status’ add-on component must be activated on your system. See your Stibo Systems representative for more information.

For more detailed information regarding PDX channel status monitoring, including configuring the PDX Channel Status Web UI component and the PDX Status Data Container attributes (i.e., Channel, Date Time, Status. etc.), see the **PDX Channel Status Monitoring** topic in the **Product Data Exchange** documentation.

Note: Channels are considered as being either a 'stand-alone' or 'group' channel. A stand-alone channel (e.g., Amazon, The Home Depot), does not have target markets / receivers, whereas a 'group' channel (e.g., 1WorldSync) does have target markets / receivers. A market is geographical (i.e., a specific country) and a receiver is a specific retailer. A stand-alone channel will have one product status, while a group channel can contain multiple product statuses regarding product submission, depending on the number of markets / receivers that have had products submitted within the channel.

The image below shows an example of the status of a product that has been added to PDX and then submitted to both stand-alone channels (i.e., ACME and Amazon) and group channels (i.e., 1WorldSync and 1WorldSync Pre-Prod).

Channel	Date Time	Status	Status Details	External Status
Added to PDX	2020-01-03 14:24:07			
1WorldSync	2020-01-03 15:05:02	Progress		
1WorldSync Pre-Prod	2020-01-10 10:40:10	Progress		
ACME	2020-01-09 13:52:37	Accepted		
Amazon	2020-01-03 14:55:19	Rejected		

Both group channels have dropdown arrows. When these arrows are clicked, the target markets that the product was submitted to within the group channels are revealed.

Channel	Date Time	Status	Status Details	External Status
Added to PDX	2020-01-03 14:24:07			
1WorldSync	2020-01-03 15:05:02	Progress		
Canada	2020-01-03 15:05:02	Progress		
Denmark	2020-01-03 15:05:02	Progress		
1WorldSync Pre-Prod	2020-01-10 10:40:10	Progress		
France	2020-01-10 10:40:10	Progress		
United Kingdom	2020-01-10 10:40:10	Progress		
United States of ...	2020-01-10 10:40:10	Multiple		
ACME	2020-01-09 13:52:37	Accepted		
Amazon	2020-01-03 14:55:19	Rejected		

Clicking on the dropdown arrows located next to the target markets (Canada, Denmark, etc.) reveals the receivers within the target markets that the product has been submitted to. In the image below, notice the highlighted blue icon in the 'Canada' row; this icon is a link to the product's URL within PDX and will appear when a channel name, market, or receiver is hovered over.

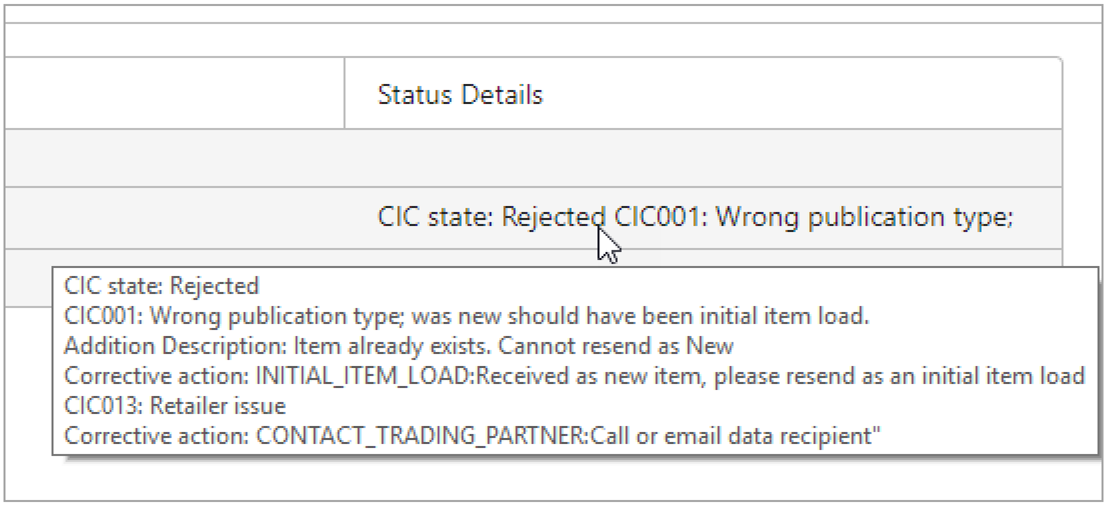
Channel	Date Time	Status	Status Details	External Status
Added to PDX	2020-01-03 14:24:07			
1WorldSync	2020-01-03 15:05:02	Progress		
Canada	2020-01-03 15:05:02	Progress		
Bunzi Distributi...	2020-01-02 14:17:20	Submitted		GDSN Status: Regi...
Denmark	2020-01-03 15:05:02	Progress		
Bozzutos	2020-01-08 10:39:43	Submitted		GDSN Status: Regi...
1WorldSync Pre-Prod	2020-01-10 10:40:10	Progress		
France	2020-01-10 10:40:10	Progress		
Pierre's Bistro	2020-01-10 10:40:10	Submitted		GDSN Status: Regi...
United Kingdom	2020-01-10 10:40:10	Progress		
Bobo's Place	2020-01-10 10:40:10	Submitted		GDSN Status: Regi...
United States of ...	2020-01-10 10:40:10	Multiple		
Franklin's	2020-01-10 10:40:10	Submitted		GDSN Status: Regi...
Jungle Jim's	2020-01-10 10:40:10	Submitted		GDSN Status: Regi...
ACME	2020-01-09 13:52:37	Accepted		
Amazon	2020-01-03 14:55:19	Rejected		

Note: If a product is submitted to the same channel more than once but without any changes to attribute values after the initial submission, there will be no change indicated in regards to submission status. Additionally, certain attributes (e.g., packaging size) cannot be changed for a product once the product is submitted to a channel.

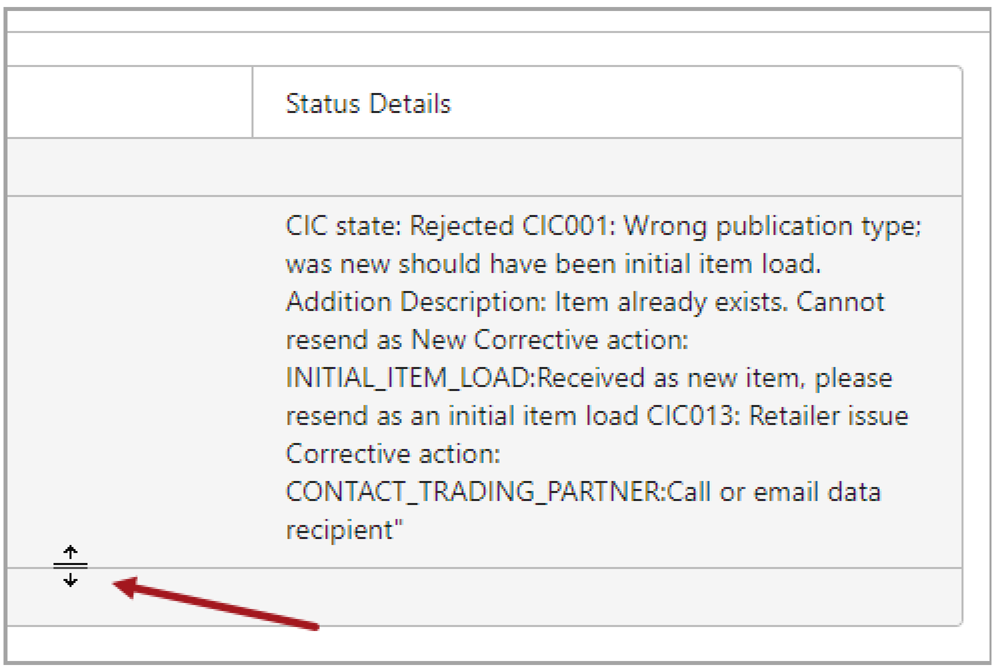
Truncated Status Messages

In the PDX Channel Status component, there will be instances (like the one shown directly above) where the full text of the message will not be displayed. To display this message in full, users can either:

- Hover over the truncated text to see the entire text in a pop-up:



- Drag the bottom rule line on the row until the row is large enough to view all the text. Alternatively, the column can be dragged wider.



Catalog Item Confirmation (CIC) Messages

A Catalog Item Confirmation (CIC) message is a GDSN-specific message type that refers to communication between a data recipient and a data source, indicating the status of an item in regards to its standing with the recipient. CIC statuses are displayed within the External Status field and additional details are displayed within the Status Details field.

There are four possible CIC status responses:

Received: Item has been received by the recipient and has passed any GDSN validation rules.

Review: Item has not been stored in the recipient's system because of validation errors.

Rejected: Item will not be stored in the recipient's system. These items will not be sent anymore.

Synchronized: Item was successfully stored in the recipient's system.

Note: While the above information discusses CIC messages specifically, be aware that different PDX channels may have messages specific to their system and will not necessarily reflect the information within a CIC message.