



SETUP and USER GUIDE

Data Integration

Release 2023.3 (11.3) – September 2023

Table of Contents

Table of Contents	2
Data Integration	8
Asynchronous Translations	8
Asynchronous Translation Status in Web UI	10
Create the Asynchronous Services Object Type	12
Configuring an Asynchronous Translation Service	15
Additional Information about Lionbridge APIv2 Connector	20
Configuring an Asynchronous File Exchange Service	21
Configuring the File Exchange Service	21
Configuring Screens for Asynchronous Translation Status in Web UI	24
Screen Configuration	24
Screen Mapping	33
Setting Up a Translation Configuration ...	35
Troubleshooting Source and Target translation	37
Language Codes for Asynchronous Services	37
Attribute Filters for Asynchronous Translation Services	42
Include Contextual Attributes for Asynchronous Translations	43
Exclude Attributes for Asynchronous Translations	45

Business Rules for Asynchronous Translations	48
Translation Status Widget Configuration ..	53
Configuring the Translation Status Widget ..	53
Optional Parameters in the Properties dialog	56
Using the Translation Status Widget	59
Interacting With the Translation Status Widget	60
Dun & Bradstreet Integration	61
Prerequisites	63
Customer Data Model	64
D&B Processing Examples	66
D&B Integration Configuration and Easy Setup	70
Configure D&B Integration	71
D&B Matching	78
Prerequisites	78
Receiving Match Candidates	78
Matching Process	80
Operation Configuration	80
References and Match Candidates	82
Selecting the Candidate	83
D&B Enriching	85
Operation Configuration	86
D&B in Web UI	87
Asynchronous	87
Synchronous	87
D&B Asynchronous Match Candidates Screen	88

Configure Asynchronous D&B Matching and Enriching	88
D&B Synchronous Match Candidates Screen	91
Configure Synchronous D&B Matching and Enriching	91
D&B Status Selector	95
Configure a D&B Workflow Status Selector Widget	95
D&B Match and Enrich Action Buttons ...	98
Recommendations	98
Synchronous Processing	98
Asynchronous Processing	99
Configure Action Buttons for Synchronous Processing	99
Configure Action Buttons for Asynchronous Processing	101
D&B Attributes and Data Containers	105
D&B Data Storage	127
D&B Direct+ API Products	128
Matching	128
Enrichment	128
Analytics Assessment	128
Company Profile	128
Corporate Linkage	128
D&B Error Handling with Integration Status	130
Integration Status and D&B Error Codes ..	130
Asynchronous Processing: Retry and Post Actions	131
Experian Email Validation Integration	134

Configuring Experian Email Validation Integration	136
Experian Email Validation Configuration Object	136
Email Revalidation Business Condition	137
Experian Email Validation Processing Plugin	139
Email Component Model	139
Configuring Experian Email Validation Integration Using the Easy Setup Wizard	141
Configuring Experian Email Validation Integration for the Web UI	146
Validate Emails by Selecting from a List of Revalidation Conditions	146
Handling Failed Email Validation	146
Configuring a Validate Emails Action Button for a Node List	146
Add a Validate Email Action Button to a Node List	146
Configure a Validate Email Action Button for a Node List	147
Add Email Revalidation Business Conditions	150
Using the Experian Email Validation Integration	153
Using the Experian Email Validation Integration with the Event Processor	154
Using the Experian Email Validation Integration from the Web UI	155
Experian Email Validation Execution Process	157
Email Validation BGP Overview	158
Handling Failed Email Validation	159

Experian Throughput Limitations	159
Loqate Integration	160
Loqate Cloud	161
Loqate Local	162
Loqate Local with CASS	162
Address Validation Web Service	162
Web service endpoint	163
Examples	163
Request	163
Response	166
Loqate Integration Properties	171
Both Local and Cloud	171
Cloud Only	171
Local Only	171
Address Component Model	172
Loqate Address Fields	173
Easy Setup of Address Component Model	179
Manual Setup of Address Component Model	187
Data Model Considerations and Requirements	187
Manual Attribute Mapping	190
CASS Address Component Model	194
Prerequisites	194
CASS Address Component Model Overview	194
CASS Address Fields - Input From Address Component Model	195

CASS Address Fields - Output In CASS Address Component Model	195
Easy Setup of CASS Address Component Model	197
Default Configuration - CASS Certification Report Event Processor	201
CASS Certification Report Generation	202
Migrating Loqate from Cloud to Local ...	205
Hash Value	205
Loqate Local Command Line Tool	207
Loqate Local Troubleshooting	208
Errors	208
Loqate Integration in STEP SaaS Deployments	209
Loqate Integration in STEP On-Premises Deployments	210
Prerequisites	210
System Requirements for Loqate Local	210
Install Recipes for Loqate Local and CASS	210
Network Access Requirements	211
Loqate Integration Properties	212
Both Local and Cloud	212
Cloud Only	212
Local Only	212
Loqate Local API Testing	215
Linux	215
Windows	215
Loqate Local License Key & Data Package Update	218
Prerequisites	218

Create Data Packages	219
Deploy Data Packages	222
Product Data Exchange	223
PDX Syndication	224
Syndication Prerequisites	224
Licenses	224
STEP Components	224
PDX Account & API User	226
PDX Syndication for Instrument	226
Setting Up the PDX OIEP	227
Base Setup	228
Handling Nested Data Structures	233
Managing Composite Attribute Names Shown in PDX	236
Transferring Assets	240
Transferring the Packaging Hierarchy ...	241
Other OIEP Dependencies	244
Validating the Configuration	248
STEP Application Properties	248
OIEP Advanced STEPXML Template ...	248
The Preamble	249
Known Limitations	252
Dimension dependent Asset content	252
Transfer of Product Families	252
LOV Value IDs	252
Product-to-Product References	252
Data containers	252
Dimension dependent STEP Names	252

Classification references	253
PDX Channel Status Monitoring	254
PDX Inbound Integration Endpoint Configuration	255
PDX Status Data Container Configuration	259
Configuring the PDX Status Data Container	259
Viewing PDX Status Data Containers in the Workbench	261
PDX Status Data Container Attributes	262
Functionality of the PDX Data Container ..	262
Viewing PDX Status Data Containers in the Web UI	263
PDX Channel Status Web UI Component	264
Configuring the PDX Channel Status Web UI Component	264
PDX Channel Status Web UI Component Link	265
PDX Channel Status Monitoring in Web UI	267
Truncated Status Messages	269
Catalog Item Confirmation (CIC) Messages	270
PDX Onboarding	272
Controlling the PDX Channel for Product Onboarding	273
Updating the data standard	273
Data standard update	273
Data Standard elements that can be changed in STEP	274
Controlling the PDX Channel for	275

Product Onboarding - Product attributes

Product families and variants275

Product-Related Attributes275

 Attribute Groups276

 Family Attributes277

 Variant Attributes277

Display sequence of attributes and attribute groups278

Locked-Down Attributes279

Controlling the PDX Channel for Product Onboarding - Data Containers ..280

Controlling the PDX Channel for Product Onboarding - Mandatory Attributes284

Mandatory for Submit284

Considerations for conditionally mandatory attributes285

Controlling the PDX Channel for Product Onboarding - Validation Engine Attributes286

Configuration details286

Validation Example287

Product data validation elements288

Using expressions correctly291

Controlling the PDX Channel for Product Onboarding - Assets293

Asset References293

Primary Image in PDX293

 Maximum Number of Assets294

Category-Specific Mandatory Asset References294

Asset Validation295

Controlling the PDX Channel for Product Onboarding - LOV Cross-validation300

Additional capabilities and considerations for LOV Cross Validation 300

Controlling the PDX Channel for Product Onboarding - Maintenance302

Pushing product to PDX in Maintenance 303

Configurations related to Maintenance ..304

Controlling the PDX Channel for Product Onboarding - Supplier-facing product hierarchy306

Controlling the PDX Channel for Product Onboarding - Supplier Hierarchy307

Storing supplier products and assets309

Controlling the PDX Channel for Product Onboarding - Additional Considerations311

Packaging311

Attribute Help Text311

Import of assets312

Event Queue312

PDX Onboarding Channel Configurator 314

Configuring PDX Onboarding Channel Configurator315

PDX Onboarding Channel Configurator Solution Initial Setup316

Configuring Web UI for PDX Onboarding Channel Configurator Solution320

Prerequisites320

PDX Onboarding Channel Configurator Screen	320
PDX Onboarding Channel Configurators - Global Navigation Panel Component	320
PDX Onboarding Channel Configurator Screen	322
Configuration Prerequisites	322
Configuration Process	322
Implementing the PDX Onboarding Channel Configurator screen	325
Configuring the Global Navigation Panel	325
Configuring the PDX Onboarding Channel Configurator to Display PDX Screen	328
Configuring PDX Onboarding Channel Configurator Setup Entity	332
Prerequisites	332
Creating PDX Onboarding Channel Configurator Setup Entity in Workbench	332
Creating PDX Onboarding Channel Configurator Setup Entity in Web UI	334
Configuring PDX Onboarding Channel Configurator Setup Entity in Web UI	335
Channel Properties Tab	337
Recommended Practices to Create Channel Logo	343
Basic logo guidelines	343
Wordmark guidelines	344
Favicon guidelines	344
Supplier Classification Tab	345
Supplier User Group Set Up	347
Locations	348

Supplier Data Definition Tab	351
Assets	353
Asset Validation Object	357
Families	360
Packaging	361
Packaging Object Types	362
Workflow And Status Tab	364
Maintenance	367
Pushing product to PDX in Maintenance	368
Status Mapping	370
PDX Presentation Tab	371
PDX Rules Tab	375
LOV Restrictions	375
Business Rule Engine	377
Configuration details	377
Validation Example	378
Product data validation elements	380
Using expressions correctly	384
Publishing PDX Onboarding Channel Configurators	385

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
- Dun & Bradstreet Integration
- Experian Email Validation Integration
- Loqate Integration
- Product Data Exchange
- Syndigo

Asynchronous Translations

This topic focuses on setting up and using the automatic service capabilities for translations so that translatable data can be sent to and from your translation service via their REST API, without human intervention. Supported options include 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. For on-premise systems, instructions for installing components can be found in the **SPOT Program** topic in the **System Administration Guide** found in **Downloadable Documentation**. For SaaS systems, contact your Stibo Systems account manager. 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				
2	Job id	Nodes number	Started Date	Translation configuration
☰	BGP_113904	1	30-04-2019 13:35:27	English to French
☰	BGP_113919	3	01-05-2019 09:41:44	English to French
☰	BGP_113920	1	01-05-2019 09:44:22	English to French
☰	BGP_113932	1	01-05-2019 10:17:40	English to French
☰	BGP_113984	1	01-05-2019 11:06:37	English to French
☰	BGP_113986	1	01-05-2019 11:07:35	English to French
☰	BGP_113988	1	01-05-2019 11:08:21	English to French

1 < < 1-50 of 104 > >|

4	
5	
ID	Title
Tips	Tips

- 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.
- 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 (refer to number 5 and 6).
- 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.
- 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.
- 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.

Node Details

Started By	USER
ID	BGP_113904
Template ID	AsyncJobRunner
Status	⏸ Suspended
Started	4/30/19 1:35:31 PM
Elapsed	7 s
Finished	4/30/19 1:35:34 PM

 Export

ID	Type ●	Text
10	Info	Exchange Service' (Tue Apr 30 13:35:31 EDT 2019)
20	Info	Executing state 'Query Translation' (Tue Apr 30 13:35:31 EDT 2019)
30	Info	Starting translation query (Tue Apr 30 13:35:31 EDT 2019)
40	Info	Analyzing translation request (Tue Apr 30 13:35:31 EDT 2019)
50	Info	Analysis selected 1 out of 1 nodes for translation (Tue Apr 30 13:35:31 EDT 2019)

Prerequisites prior to configuring Web UI for asynchronous translations

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

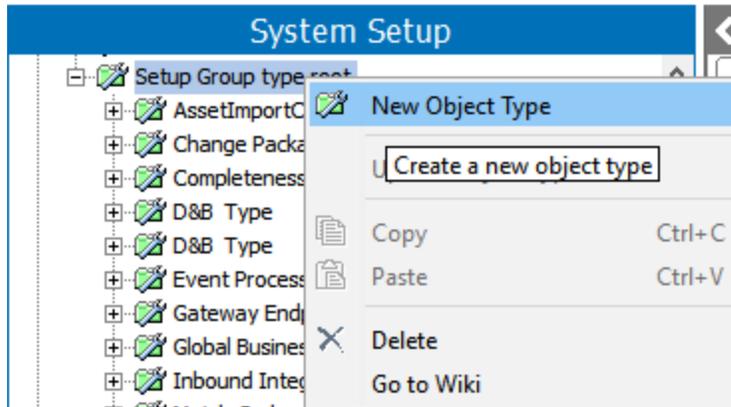
1. In the workbench, create a translation configuration and include at least one asynchronous service in that configuration. For more information, refer to the **Asynchronous Translations** topic.
2. Configure the screens and mappings necessary to view asynchronous translation status details in Web UI.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

Create the Asynchronous Services Object Type

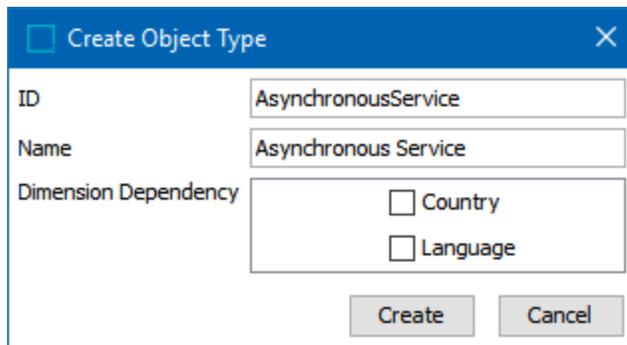
Before an asynchronous service can be set up and configured, an Asynchronous Services setup group root node and object type must be created. For more information about creating Setup Groups in general, refer to the **Setup Groups** section of the **System Setup**.

Use the following steps to create the object type.

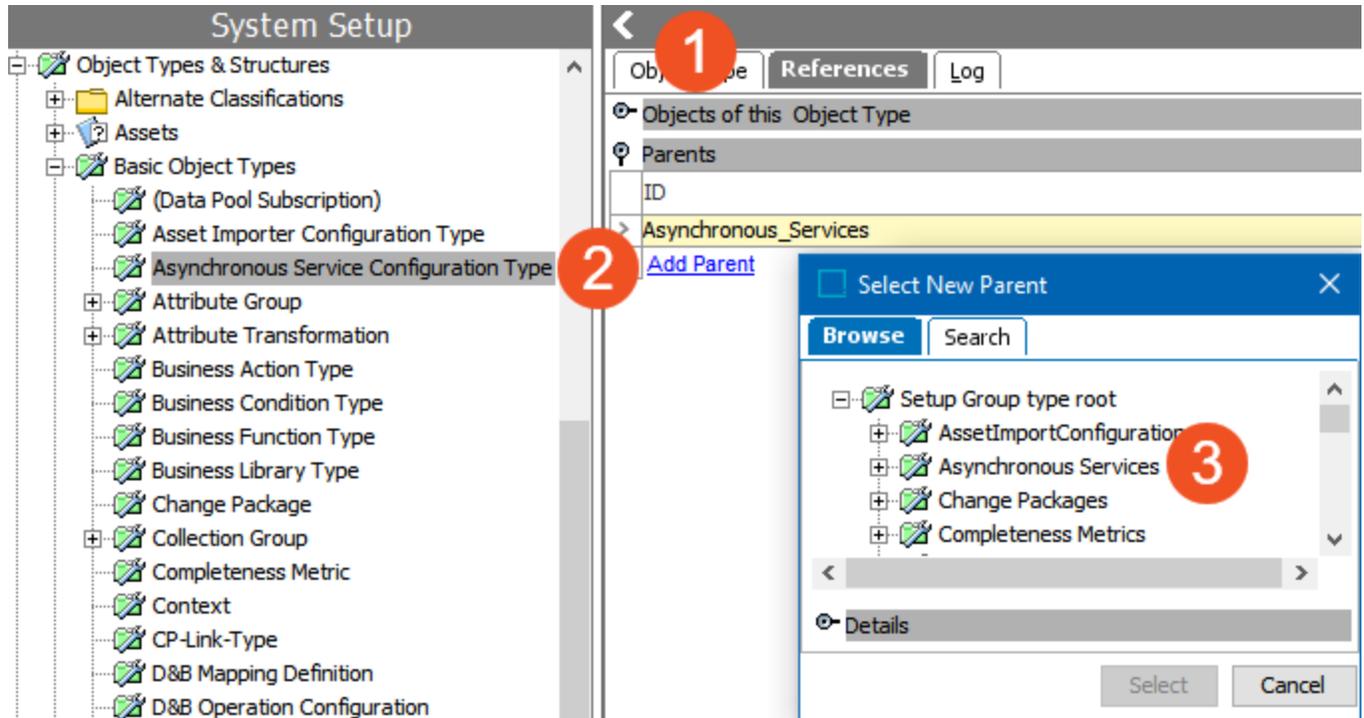
1. From 'System Setup', expand 'Object Types and Structures' and right-click **Setup Group type root**. Select **New Object Type**.



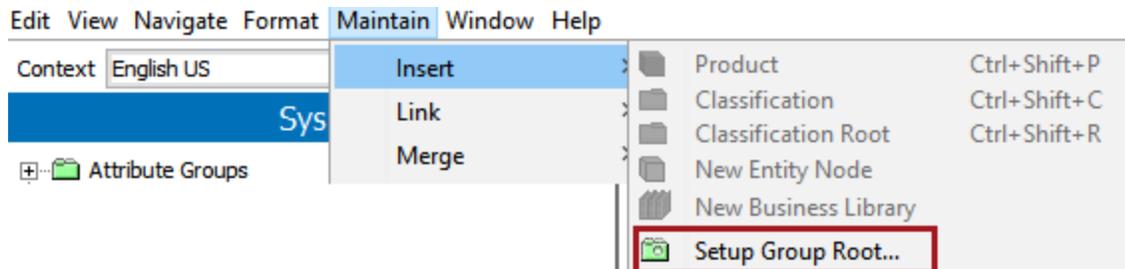
2. Enter a Name and ID in the **Create Object Type** dialog box.



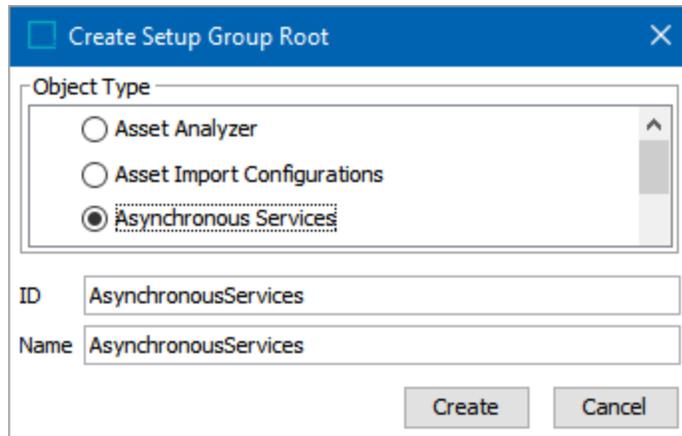
3. Remaining in 'Object Types & Structures', navigate to and expand 'Basic Object Types.' Select the 'Asynchronous Service Configuration Type.' Click on the References tab and select 'Add Parent.' In the **Select New Parent** dialog box, select the setup group root that was created in step two, and click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



- 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**.



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



Create Setup Group Root

Object Type

Asset Analyzer

Asset Import Configurations

Asynchronous Services

ID: AsynchronousServices

Name: AsynchronousServices

Create Cancel

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

All steps in this section apply to Across, SDL, and Lionbridge services except in the **Server Connection Details** parameter, which is broken out by service.

Note: For information regarding configuring an Asynchronous File Exchange Service, refer to 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:

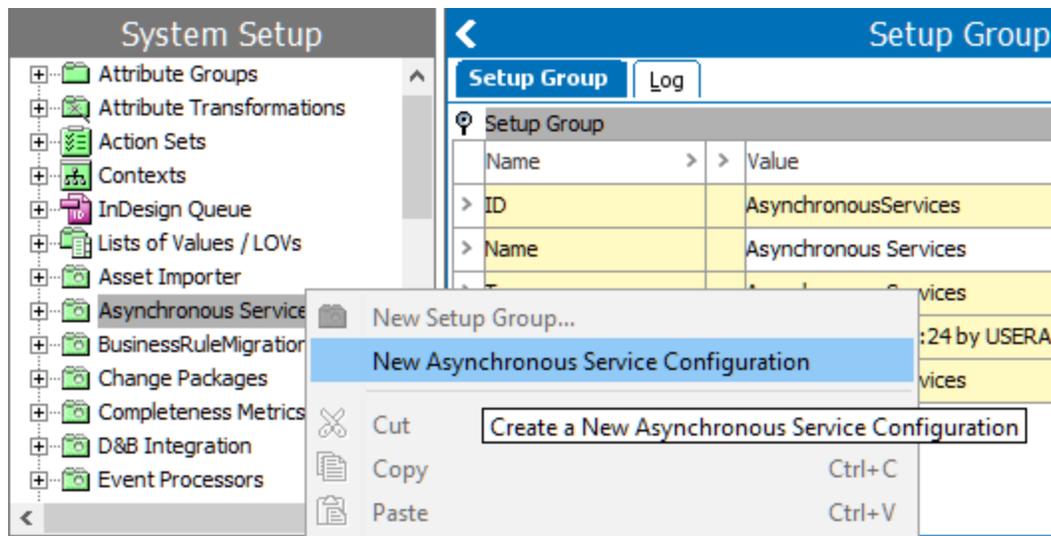
```
Async.Kernel.Config.Across.ServerURL=1=http://across.scloud.com/crossAPI/crossAPI.wsdl,2=http://across.scloud2.com/crossAPI/crossAPI.wsdl
```

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.

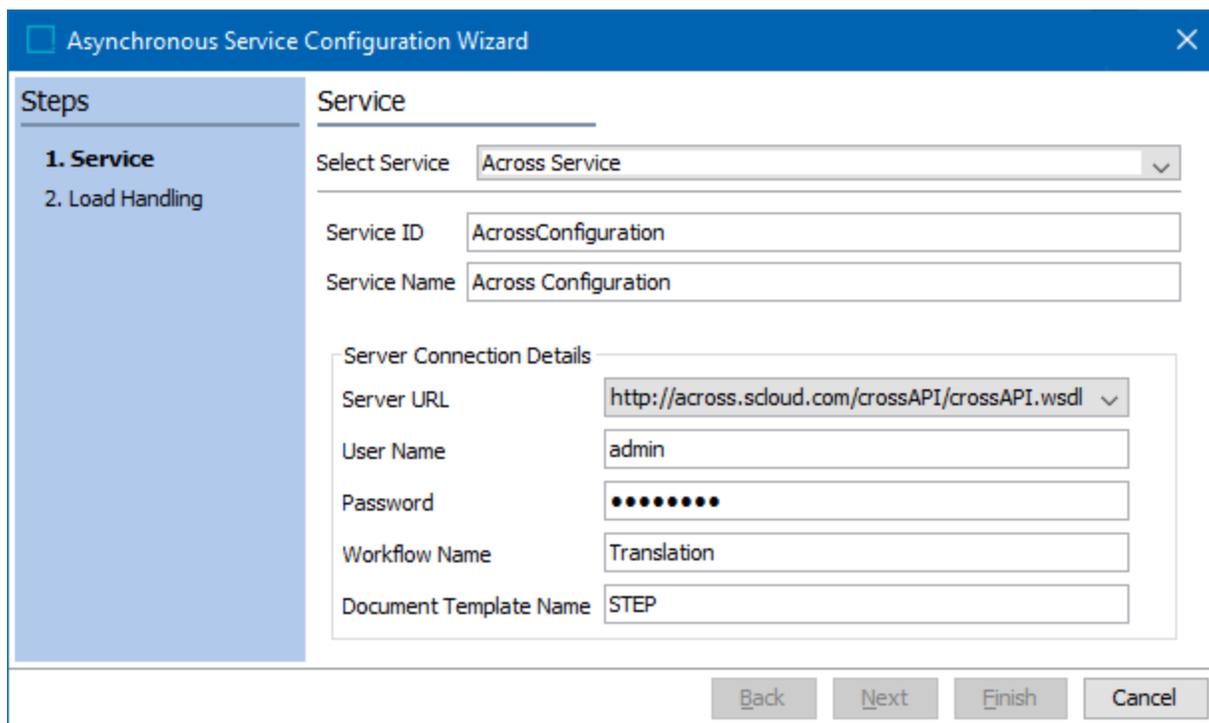
```
Async.Kernel.Config.Across.ServerURL=1=http://127.0.0.1:5000
```

Configuration

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

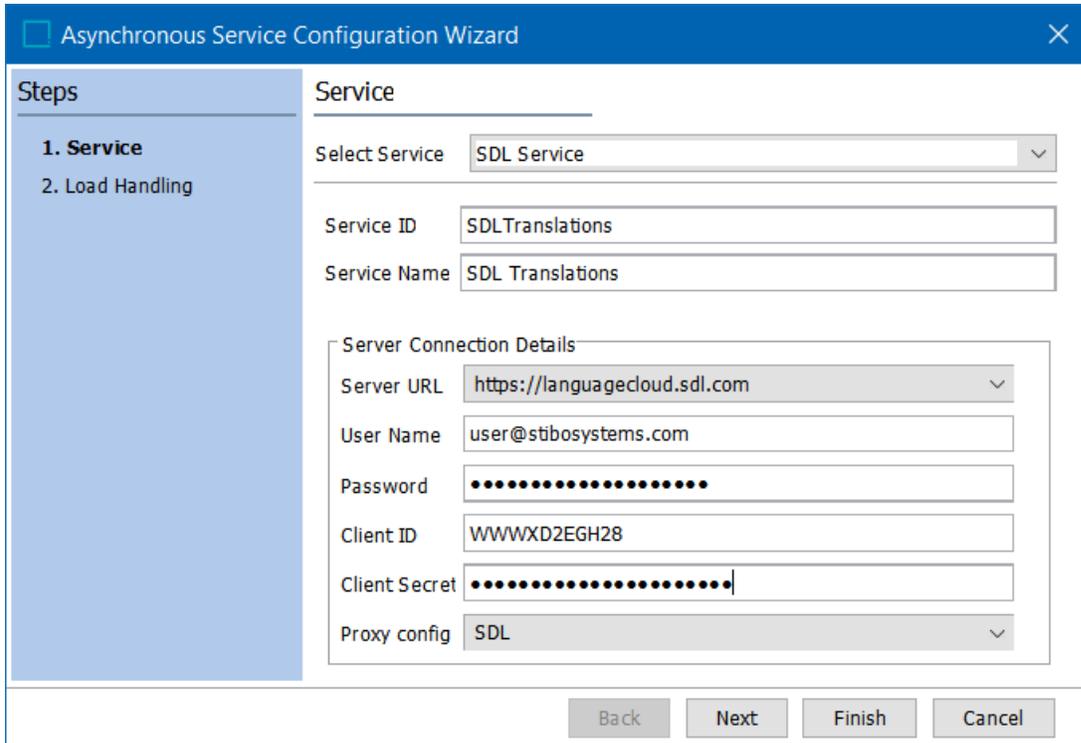


2. In the **Asynchronous Service Configuration Wizard**, complete the parameters on the **Service** step based on the required service:
 - For a local **Across** translation server, enter your own values for the parameters as shown in the example below:



- **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:



The screenshot shows the 'Asynchronous Service Configuration Wizard' window. On the left, a 'Steps' pane lists '1. Service' and '2. Load Handling'. The main area is titled 'Service' and contains the following fields:

- Select Service:** A dropdown menu with 'SDL Service' selected.
- Service ID:** A text box containing 'SDLTranslations'.
- Service Name:** A text box containing 'SDL Translations'.
- Server Connection Details:** A section containing:
 - Server URL:** A dropdown menu with 'https://languagecloud.sdl.com' selected.
 - User Name:** A text box containing 'user@stibosystems.com'.
 - Password:** A text box filled with dots.
 - Client ID:** A text box containing 'WWWXD2EGH28'.
 - Client Secret:** A text box filled with dots.
 - Proxy config:** A dropdown menu with 'SDL' selected.

At the bottom of the wizard are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

- **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
- **Proxy config** - Select 'SDL' from this dropdown if the connection must first pass through a proxy server with its own login requirement. For 'SDL' to show as an option and to successfully connect via proxy, the following properties need to be added to the sharedconfig.properties file (the values after the equal sign are examples and actual entries will differ):

```
Http.ProxyConfiguration.SDL.Host=localhost
Http.ProxyConfiguration.SDL.Port=8080
Http.ProxyConfiguration.SDL.User=user
Http.ProxyConfiguration.SDL.Password=password
```

- 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:

Note: If you are setting up a new service for Lionbridge, you must use **Lionbridge Service V2**.

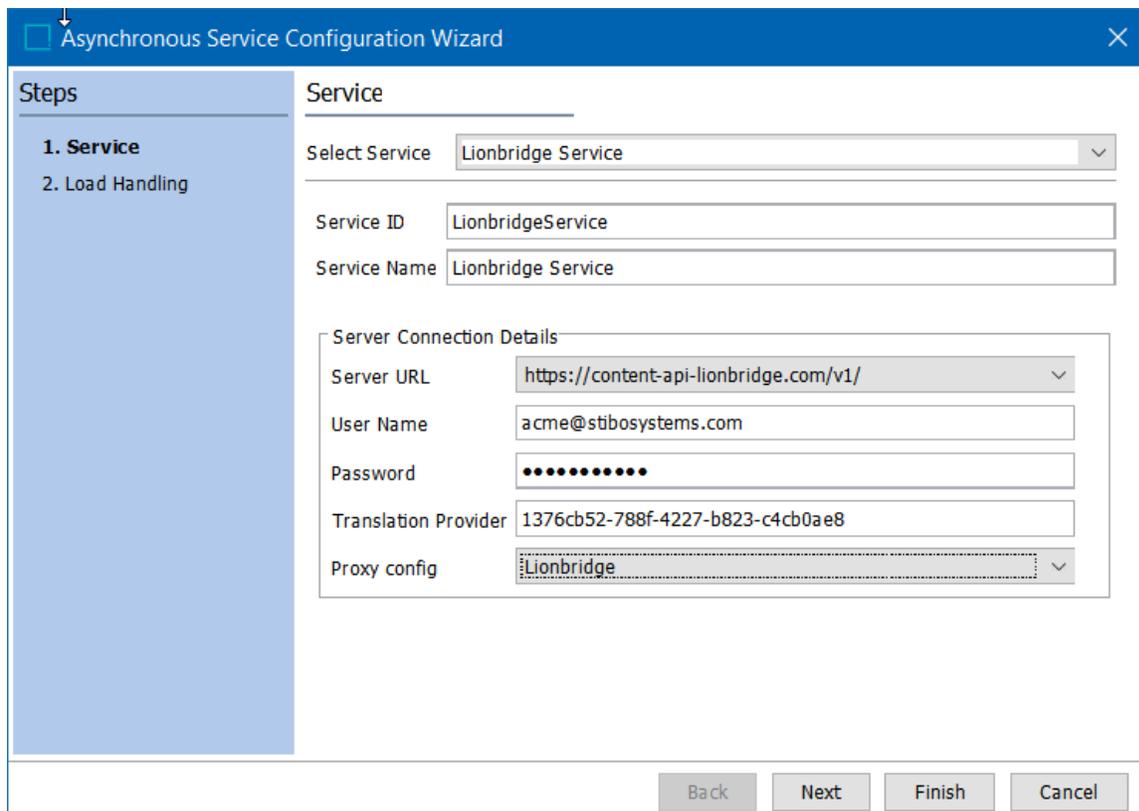
The screenshot shows the 'Asynchronous Service Configuration Wizard' window. The 'Steps' pane on the left indicates '1. Service' is the current step. The 'Service' section contains the following fields:

- Select Service:** A dropdown menu set to 'Lionbridge Service V2'.
- Service ID:** A text box containing 'LionbridgeService'.
- Service Name:** A text box containing 'Lionbridge Service'.
- Server Connection Details:** A group box containing:
 - Server URL:** A dropdown menu set to 'http://content-api-lionbridge.com/v2/'.
 - Client ID:** A text box containing 'LionbridgeClientIDSample'.
 - Client Secret:** A text box filled with 20 dots.
 - Translation Provider:** A text box containing '1376cb52-788f-4227-b823-c4cb0a56a0e8'.

At the bottom of the wizard are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

- **Select Service** - Select the **Lionbridge Service V2** 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
 - **Client ID** - Enter the client ID assigned to your application from Lionbridge
 - **Client Secret** - Enter the client secret assigned to your application from Lionbridge
 - **Translation Provider** - Enter the translation provider key provided by Lionbridge

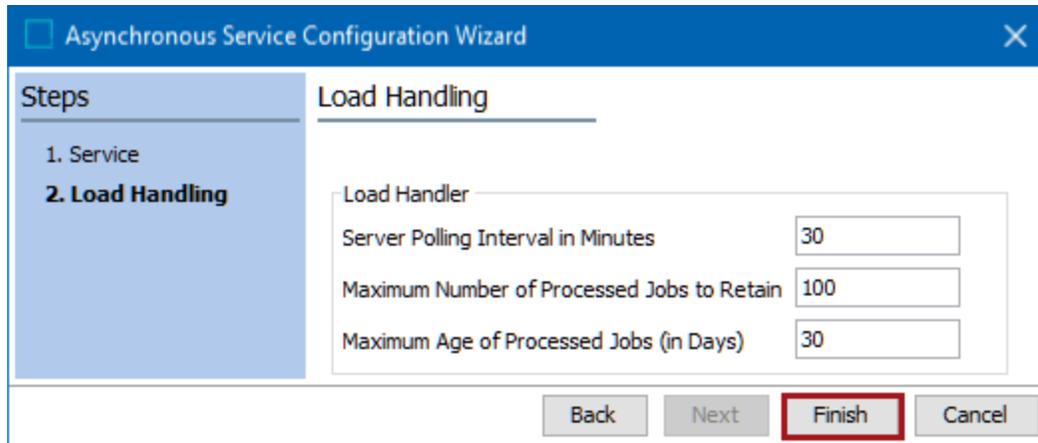
Important: Lionbridge is scheduled to discontinue support for APIv1 (shown below) in Q2 2022, and recommends existing customers contact their Lionbridge representative to access an APIv2 staging environment for testing purposes before using the new service on production systems.



- **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
- **Proxy Config** - Select 'Lionbridge' from this dropdown if the connection must first pass through a proxy server with its own login requirement. For 'Lionbridge' to show as an option and to successfully connect via proxy, the following properties need to be added to the sharedconfig.properties file (the values after the equal sign are examples and actual entries will differ):

```
Http.ProxyConfiguration.Lionbridge.Host=localhost
Http.ProxyConfiguration.Lionbridge.Port=8080
Http.ProxyConfiguration.Lionbridge.User=user
Http.ProxyConfiguration.Lionbridge.Password=password
```

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



Asynchronous Service Configuration Wizard

Steps

1. Service
- 2. Load Handling**

Load Handling

Load Handler

Server Polling Interval in Minutes:

Maximum Number of Processed Jobs to Retain:

Maximum Age of Processed Jobs (in Days):

Back Next **Finish** Cancel

- **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.

Additional Information about Lionbridge APIv2 Connector

When STEP receives translation jobs from Lionbridge API V2, the translation enters a 'Quarantine' state. The translation job resides in this state for 5 days before it is completed and archived. This interim time period allows the customer to review and, if necessary, send the translation job back for rework.

Execution reports have specific ID that Lionbridge uses to assist with errors. Also, error messages from Lionbridge will be written in the execution report. For example, if a requested language is not supported, a message in the execution report will address the unsupported language. Users can then contact Lionbridge for more support.

There is also additional support for proxy servers.

The Translation Service Connector - Lionbridge commercial license is required to use this functionality.

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. Contact your Stibo Systems account manager or partner manager 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, refer to the **Asynchronous Translations** topic.

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

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

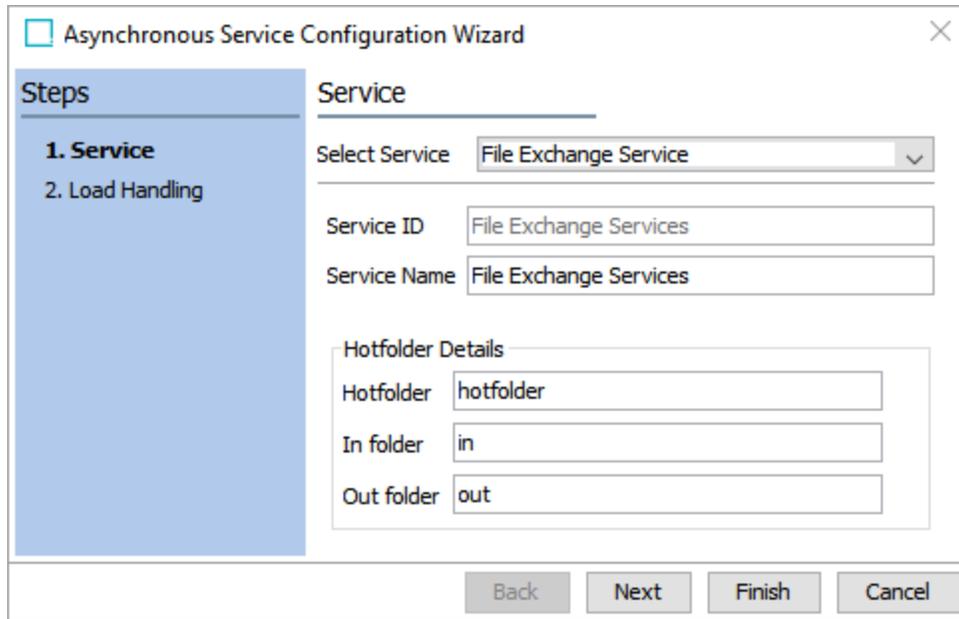
Configuring the File Exchange Service

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, refer to the **Setup Groups** section of the **System Setup**.

Use the following steps to configure the File Exchange Service:

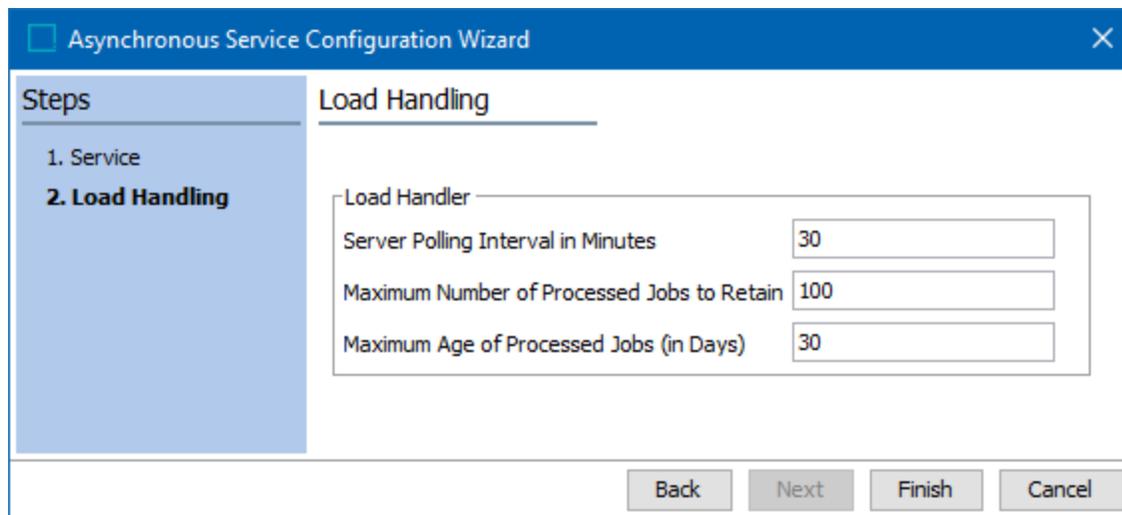
1. With the workbench open and the System Setup tab selected, right-click on the Asynchronous Services root node, and select 'New Asynchronous Service Configuration.' The Asynchronous Service Configuration Wizard opens.
2. On the Service step:



- 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, refer to the **Design Mode Basics** topic in the **Web UI Getting Started** section of the **Web User Interfaces** 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.'

Add Screen

Screen ID

This component enables users to list asynchronous jobs.

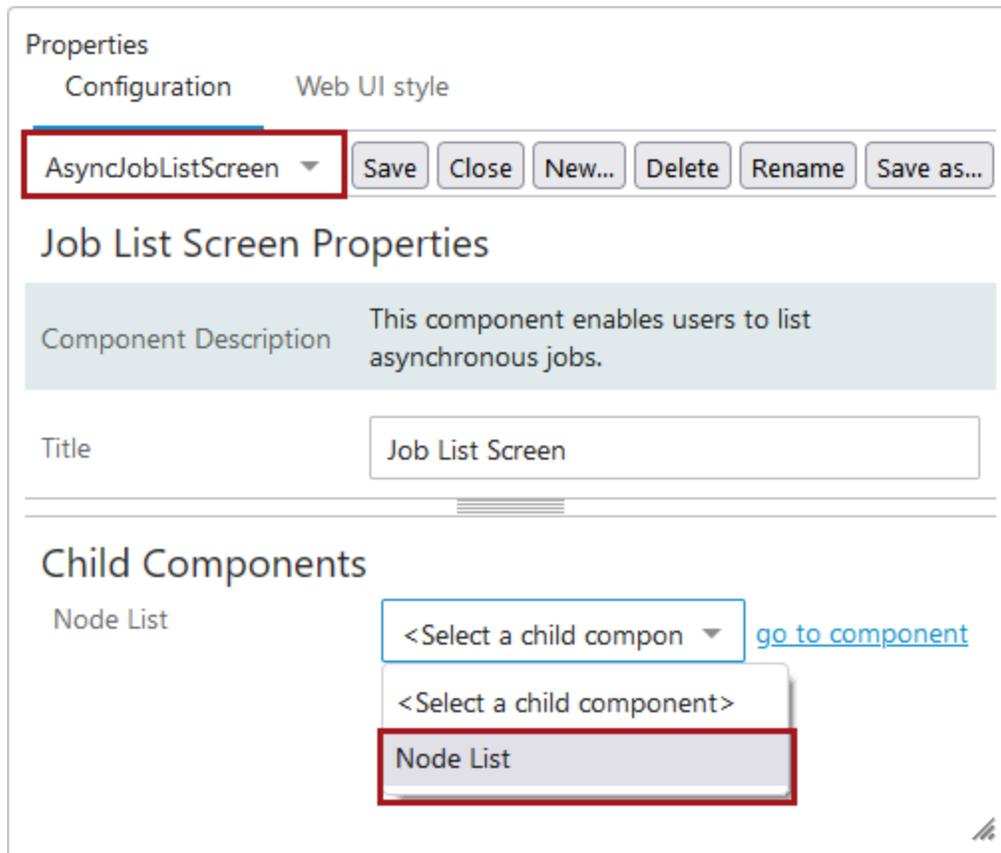
- Golden Record Clerical Review Task List
- Golden Record Source Traceability
- Home Page
- Image Deduplication Clerical Review
- Initiate Item
- Job List Screen
- Job Nodes List Screen
- List of Values Group Management Screen
- List of Values Management Screen
- Mass Creation Screen
- Merge Nodes Screen
- Multi Context Screen

Filter

Show deprecated components

Cancel
Add

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. Under the Display Modes parameter, double click 'Table Display Mode.'

Properties (edited)

Configuration Web UI style

AsyncJobListScreen Save Close New... Delete Rename

Node List Properties [go to parent](#)

Component Description The Node List displays objects presented in table or in a grid. Different Display Modes can be applied and customised with a range of headers allowing for different information about the listed objects to be displayed.

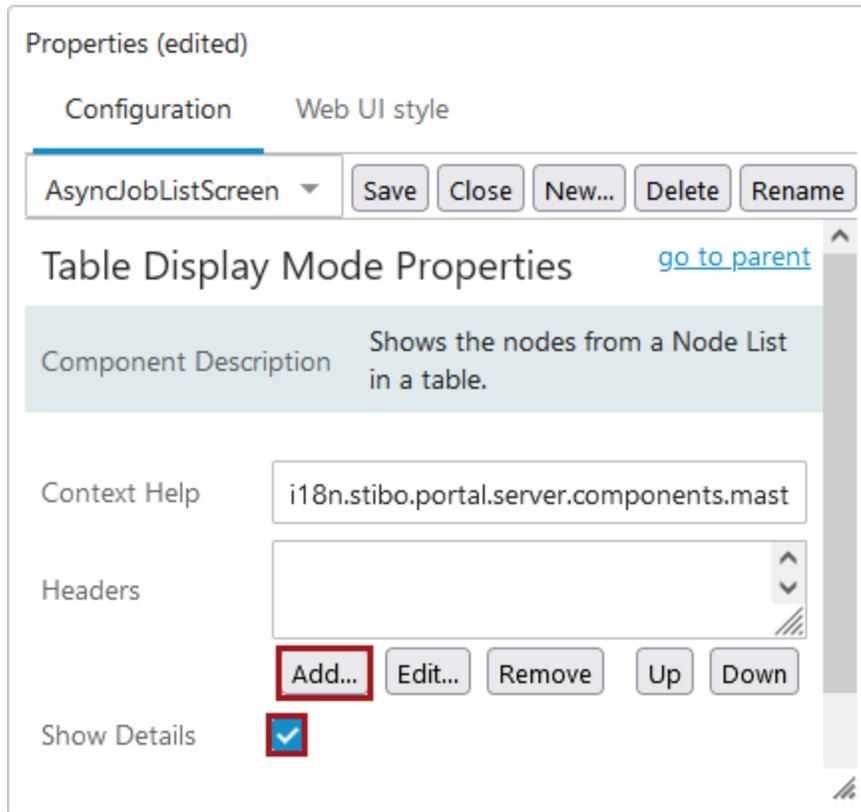
Child Components

Display Modes

Table Display Mode

Add.. Remove Up Down

- In the Table Display Mode Properties dialog, check the 'Show Details' checkbox and in the Headers parameter, 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.

Add Component

Advanced Merge Globally Configured Data Container Header	Table header that shows the ID of the Asynchronous Job in the table.
Approved Header	
Asynchronous Job Nodes ID Header	
Asynchronous Job Nodes Number Header	
Asynchronous Job Started Date Header	
Asynchronous Job Status Header	
Asynchronous Job Translation Configuration Header	
Attribute Link Wiki Header	
Attribute Value Group Header	
Attribute Value Header	
Business Function to Node	
CIC Status Header	

Filter

Show deprecated components

Cancel Add

8. Click 'Add.'

The Asynchronous Job Nodes ID Header Properties dialog opens.

Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Asynchronous Job Nodes ID Header Properties

Component Description Table header that shows the ID of the Asynchronous Job in the table.

Dimensions

<Select an option>

Edit...

Enable Link



Label

* Result Screen

homepage

Add

Table Sorting

<Select a value>

Context Help

Cancel

Add



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

Add Screen

Screen ID

JobListBGPDetails

- Actual Page Screen
- Advanced Search Screen
- Analytics Screen
- Asset Folder Screen
- Asset Preview Screen
- Attribute Group Management Screen
- Attribute Link Editor Screen
- Attribute Management Screen
- Background Process Node Details**
- Background Processes Screen
- Basket Screen
- Basket Statistics

Screen for displaying details about a selected Background Process. The screen has been preconfigured with a Node Editor and a selection of components for displaying information about a background process.

Filter

Show deprecated components

Cancel Add

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

Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Asynchronous Job Nodes ID Header Properties

Component Description	Table header that shows the ID of the Asynchronous Job in the table.	
Dimensions	<Select an option>	Edit...
Enable Link	<input checked="" type="checkbox"/>	
Label	<input type="text"/>	
* Result Screen	JobListBackGroudProcessScreen	Add
Table Sorting	<Select a value>	
Context Help	<input type="text"/>	

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.

Properties (edited)

Configuration Web UI style

AsyncJobListScreen Save Close New... Delete Rename Save as...

Table Display Mode Properties [go to parent](#)

Component Description Shows the nodes from a Node List in a table.

Context Help i18n.stibo.portal.server.components.masterdetail.TableDisplayMode

Headers

- Asynchronous Job Nodes ID Header (true / JobListBackGroudProcessScreen)
- Asynchronous Job Nodes Number Header
- Asynchronous Job Started Date Header
- Asynchronous Job Translation Configuration Header

Add... Edit... Remove Up Down

Show Details

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.'

Add Screen

Screen ID

JobNodesListScreen

Job List Screen

Job Nodes List Screen

List of Values Group Management Screen

Multi Selection Screen

Multi Workspace Screen

This component enables users to list nodes assigned to asynchronous jobs.

Filter

Show deprecated components

Cancel
Add

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, refer to 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.

Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Screen Mapping Properties

Component Description A mapping rule that will forward to the specified screen if all supplied conditions are satisfied.

* Conditions

Asynchronous Job Process Condition

Add... Edit... Remove Up Down

* Screen

JobNodesListScreen Add

Cancel Add

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, refer to 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.

The screenshot displays the 'System Setup' interface. On the left, a tree view shows various system components, with 'SDL Translations' selected under 'Asynchronous Services'. The main window title is '< SDL Translations rev.0.1 - Asynchronous Service Config...'. Below the title, there are tabs for 'Asynchronous Service Configuration Type', 'Log', and 'Status'. The 'Description' section contains a table with the following data:

Name	Value
ID	SDLTranslations
Name	SDL Translations
Object Type	Asynchronous Service Configuration Type
Revision	0.1 Last edited by USERA on Tue Nov 14 05:57:16 EST 2017
Path	Asynchronous Services/SDL Translations

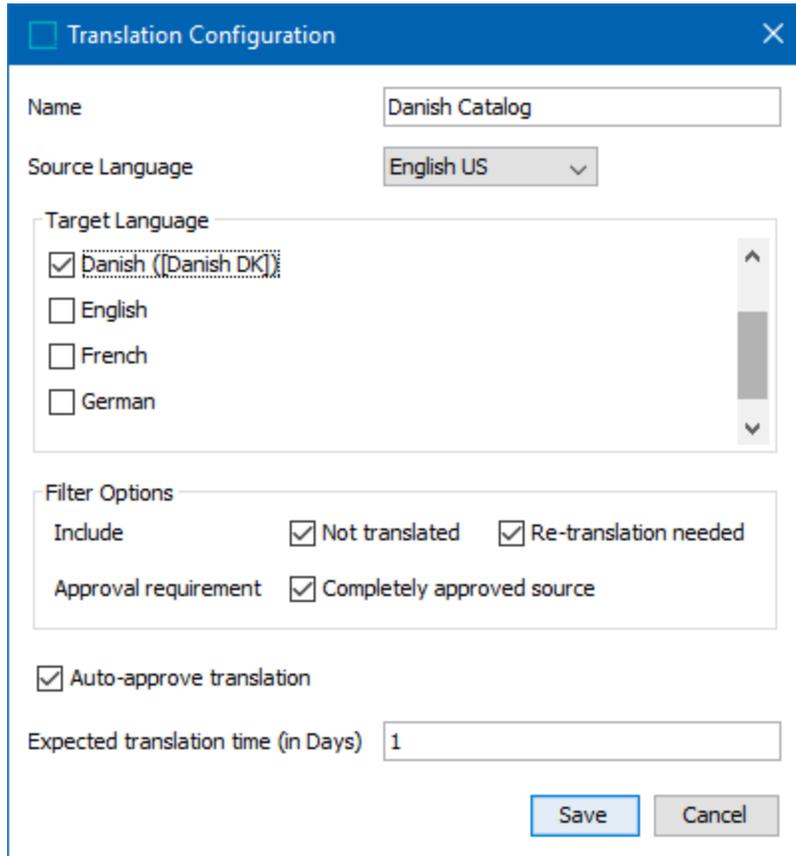
Below the description, there are sections for 'Server Connection Details' and 'Load Handling'. The 'Load Handling' section contains a table with the following data:

Server Polling Interval in Minutes	30
Maximum Number of Processed Jobs to Retain	100
Maximum Age of Processed Jobs (in Days)	30
Poller Status	Running

The 'Translation Configuration' section is expanded, showing a table with the following data:

Translation configuration	>	>
Danish Catalog	...	X
> Add		

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



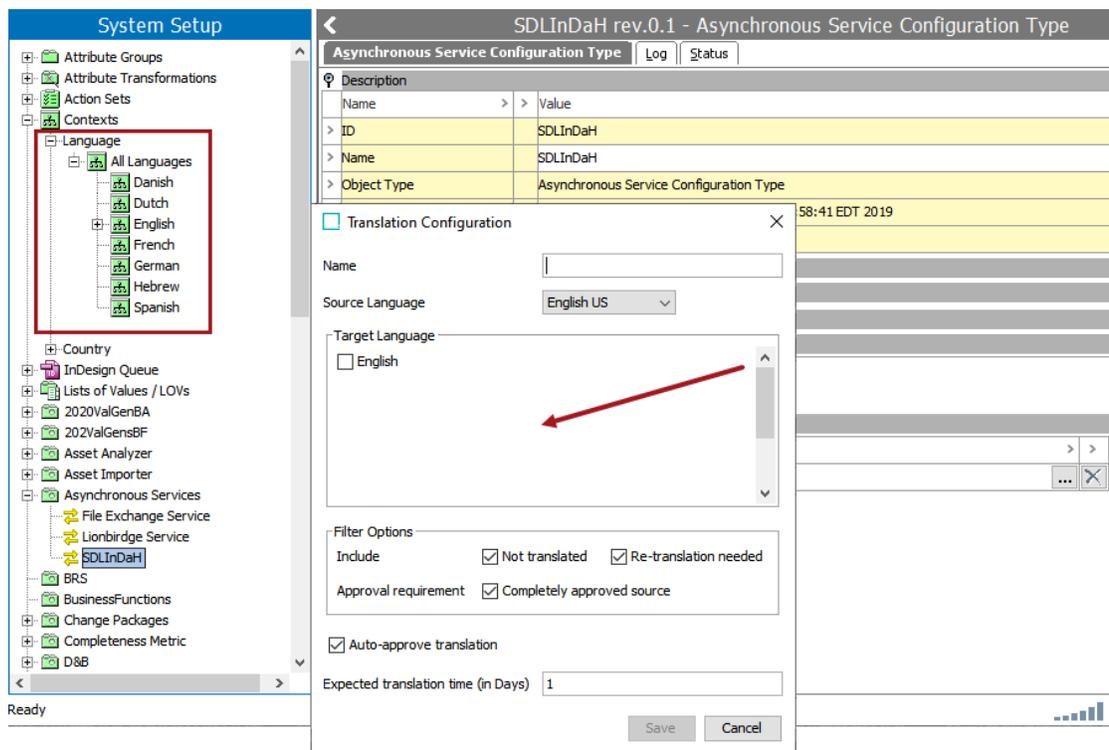
Important: If the expected source and/or target translations are not displaying in the Translation Configuration dialog, refer to a troubleshooting guide below.

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, refer to 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.

The screenshot shows the 'System Setup' interface. On the left, a tree view under 'Language' shows 'All Languages' expanded, with 'English' selected. On the right, the 'English - Dimension Point' configuration window is open, displaying a table of properties.

Description	
Name	Value
ID	en-US
Name	English
Dictionary	English (United States)
GDSN mapping	en
Parents	
Name	Value
std.lang.all	All Languages

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, refer to the **Transformation Lookup Tables** topic in the **Resource Materials** online help.

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

Description	
Name	Value
ID	fr
Name	French
Dictionary	Français Classique (France)
GDSN mapping	fr

Parents	
Name	Value
std.lang.all	All Lanquaques

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.

Tree

- Image Conversions
- Import Configurations
- Portal Configurations
- Smart Sheet Configurations
- Transformation Lookup Tables
 - Acme Smartsheet
 - AdvSTEPXML
 - Context Transformations
 - Description Attribute Merging
 - Dimension Dependency
 - Excel Template
 - Filter Template
 - ProductTables
 - Sample Configuration
 - SDL Language Mapping**
 - SmartsheetHelp
 - Standardize Address
- Index Words
- Merchandising Hierarchy
- Suppliers
- Suppliers Root
- UNSPSC-CODE
- Web Sites US
- Company Hierarchy Data Root
- Customer Root
- Entity Root
- GDSN

SDL Language Mapping rev.8.0 - Transformation Lookup Table

Transformation Lookup Table

Description

Name	Value
ID	SDL_Language_Mapping
Name	SDL Language Mapping
Object Type	Transformation Lookup Table
Revision	8.0 Last edited by USER7 on Fri Nov 17 03:40:52 EST 2017
Approved	Never Been Approved
Translation	Not Translated
Path	Classification 1 root/Configurations/SDL Language Mapping
Content In	Language=All Languages

Lookup Table

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

Ignore Case

From	To
fr	fr-FR
German	de-DE

[Add Row](#)

3 Rows Import From Clipboard Apply

- **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.

Across Translation Lookup rev.0.1 - Transformation Lookup Table

Transformation Lookup Table

Description

Lookup Table

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

Ignore Case

From	To
de-DE	1031
en-US	1033
fr-FR	1036

[Add Row](#)

4 Rows Import From Clipboard Apply

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, refer to 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.

System Setup

- Attribute Groups
- Attribute Transformations
- Action Sets
- Contexts
- InDesign Queue
- Lists of Values / LOVs
- Asset Analyzer
- Asset Import Configurations
- Asynchronous Services
 - Across
 - File Exchange Service
 - SDL Translations
- AsyncServices
- AsynService819
- Biz_Con
- Completeness Metrics
- Data Profile Configuration
- DataSufficiencyScoreGroup
- Elasticsearch Configuration
- Event Processors
- GDSNReceiverRestBusinessActions

Asynchronous Service Configuration Type | Log | Status

Description

Name	Value
ID	SDL
Name	SDL Translations
Object Type	Asynchronous Service Configuration Type
Revision	0.1 Last edited by USERK on Wed Oct 28 17:24:19 EDT 2020
Path	Asynchronous Services/SDL Translations

Server Connection Details

Load Handling

Contextual Attributes

Attribute configuration	>
Add Attribute/Group	

Excluded Attributes

Attribute configuration	>
Add Attribute/Group	

Translation Configuration

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:

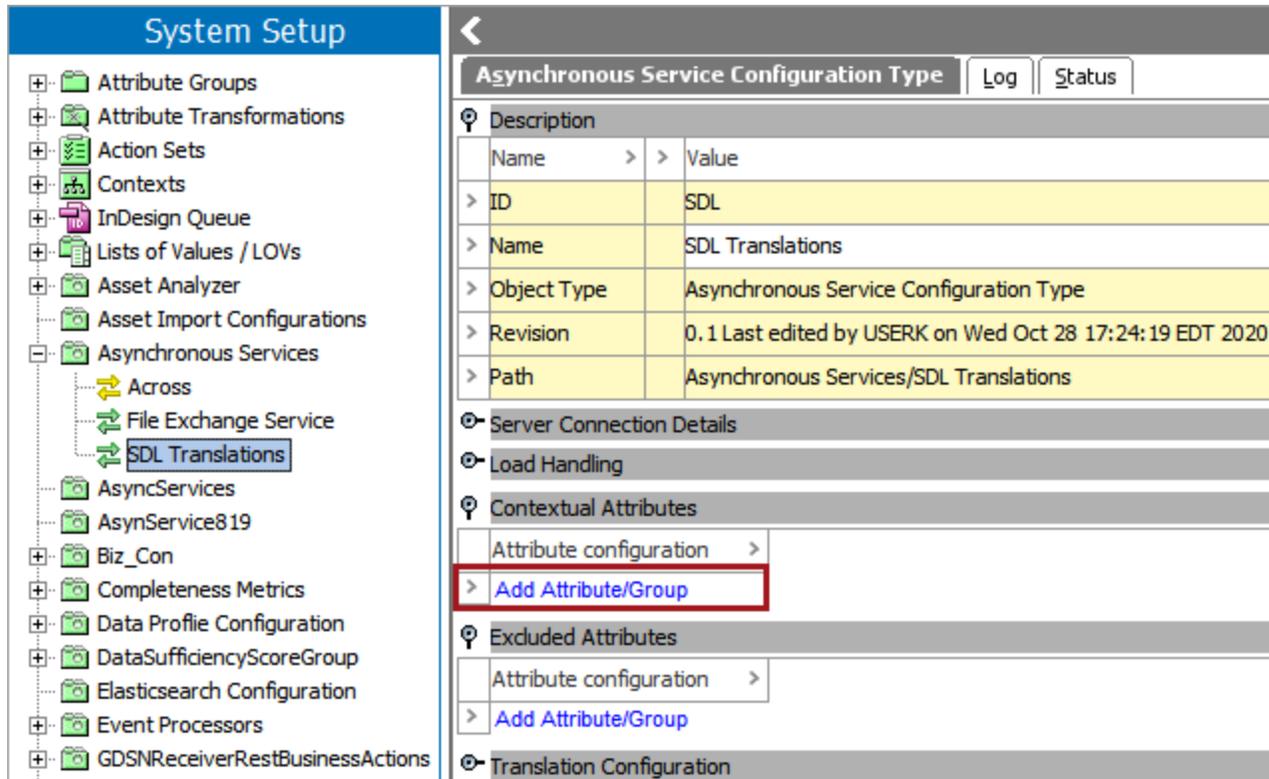
- Create an asynchronous service object type as described in the **Create the Asynchronous Services Object Type** topic.
- Complete the Asynchronous Service Configuration Wizard as described in the **Configuring an Asynchronous Translation Service** topic.

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

Include Contextual Attributes for Asynchronous Translations

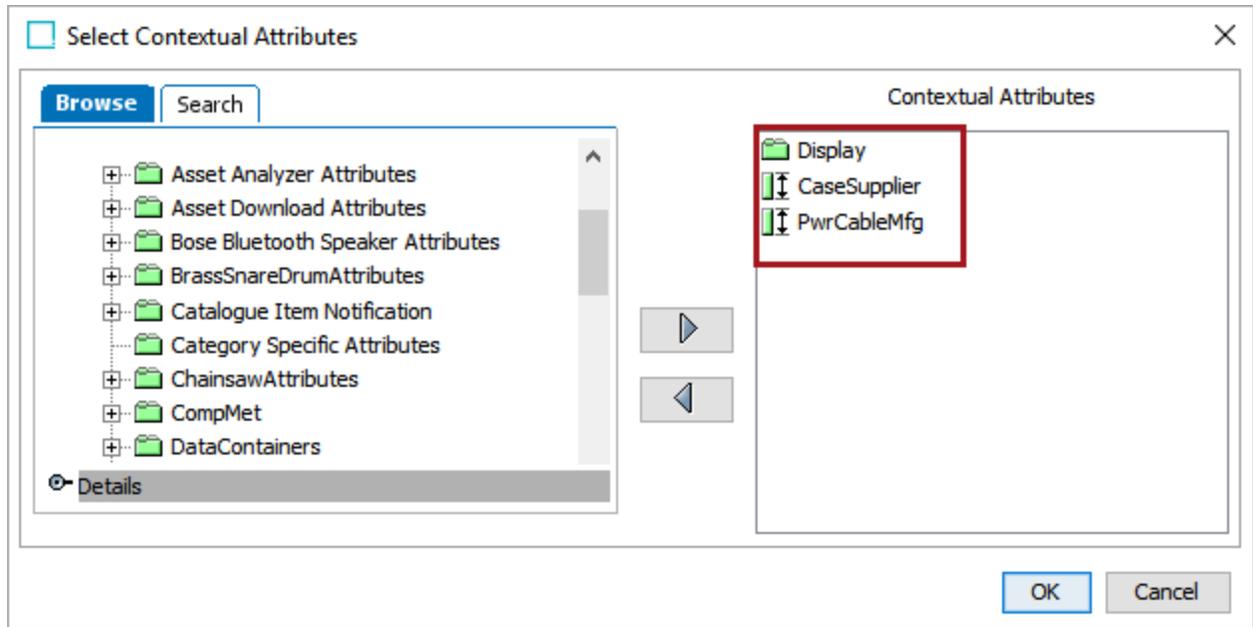
Contextual attributes are attributes that although are language-independent, provide the translator with contextual information about the object which contains the language-dependent attributes being translated. Contextual attributes selected are included within an asynchronous translation, while all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

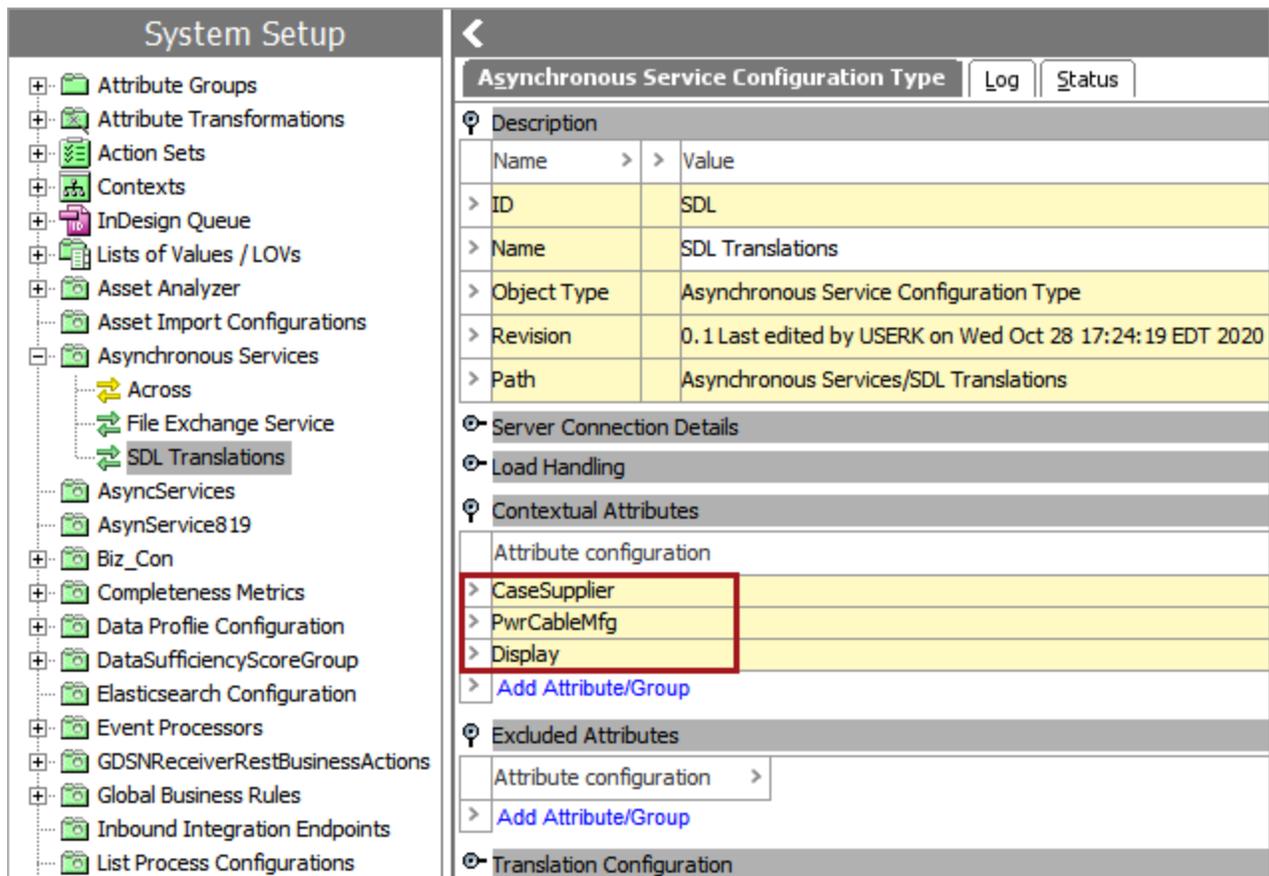


2. In the 'Select Contextual Attributes' dialog, select the desired contextual attributes / attribute groups to include for translation when a translation is run using the selected asynchronous service, and use the right arrow () button to move those selections to the 'Contextual Attributes' column.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



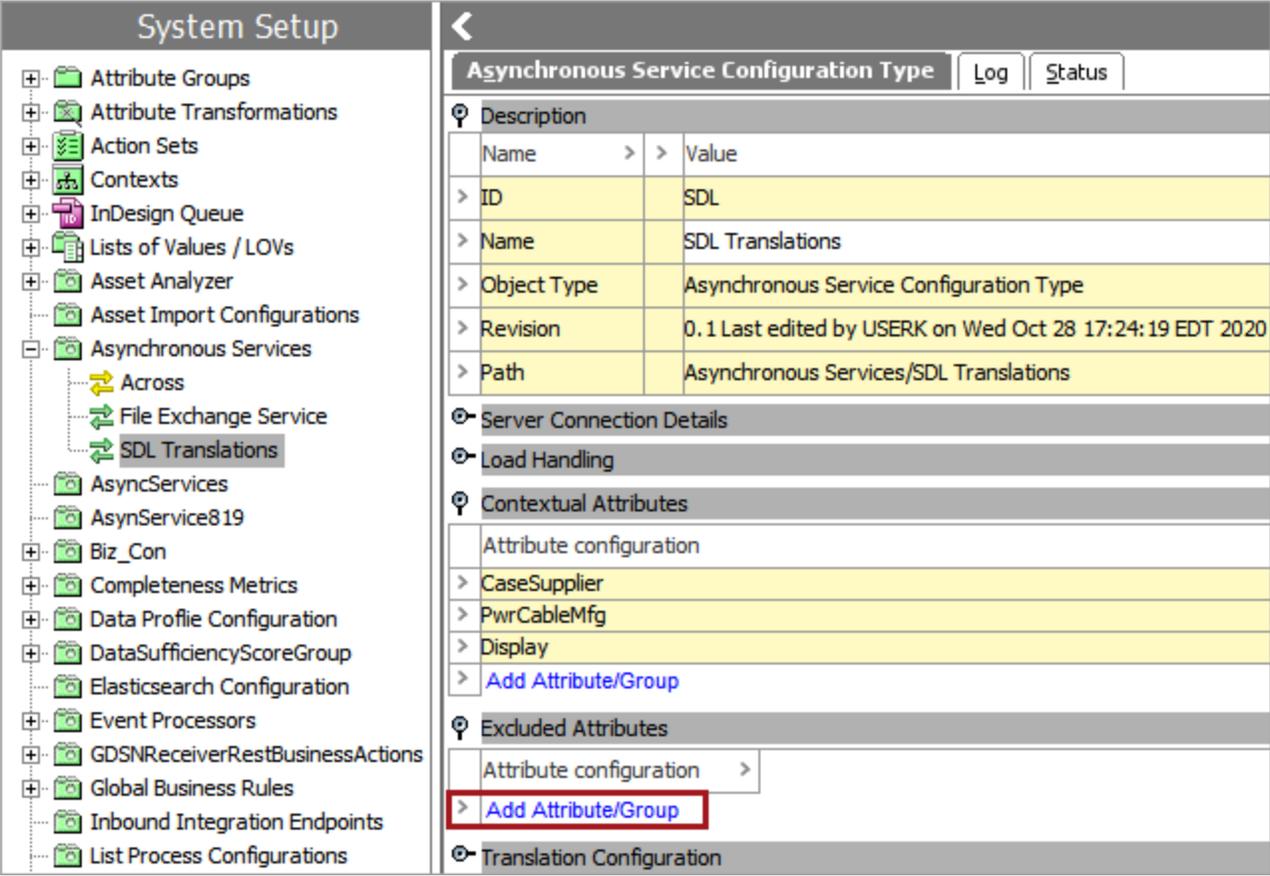
Only the selected contextual attributes / attribute groups will be included when a translation is run using the selected asynchronous service.

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.

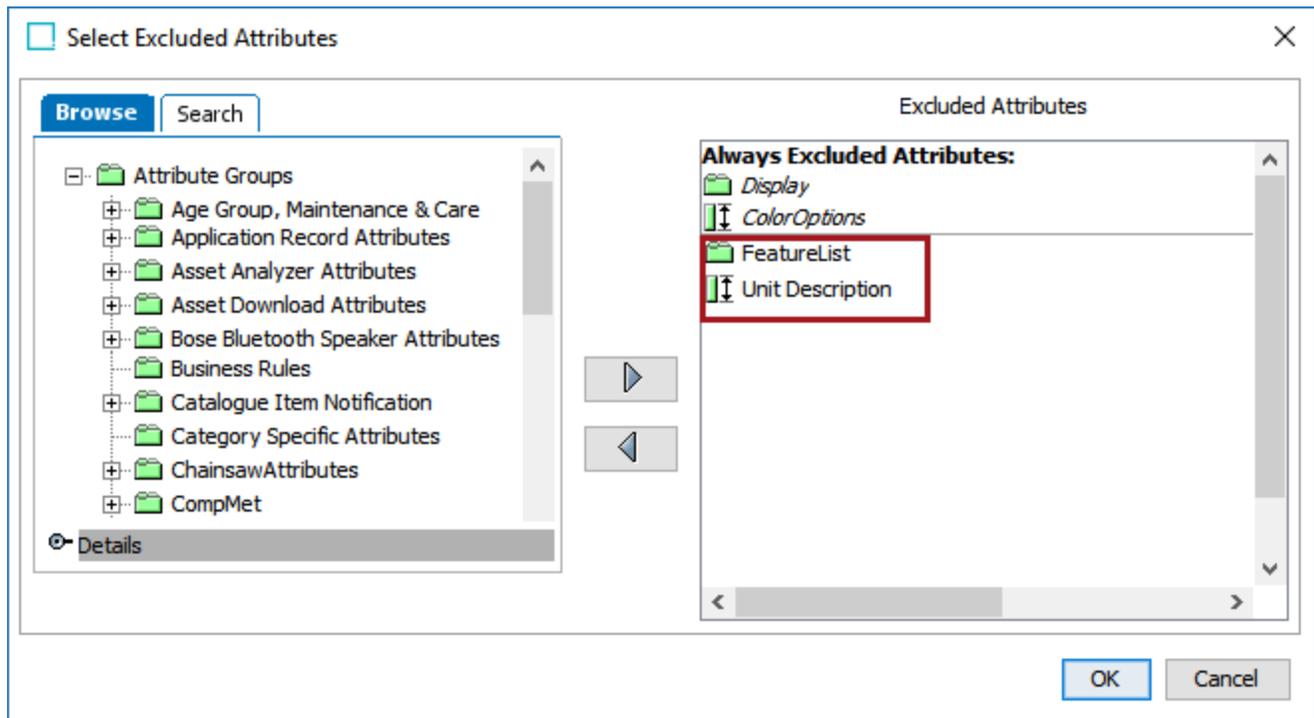
To exclude attributes and/or attribute groups from an asynchronous translation:

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, refer to the **Translation Settings** topic in the **System Settings** section of the **System Setup** 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.

System Setup

- Attribute Groups
- Attribute Transformations
- Action Sets
- Contexts
- InDesign Queue
- Lists of Values / LOVs
- Asset Analyzer
- Asset Import Configurations
- Asynchronous Services
 - Across
 - File Exchange Service
 - SDL Translations
- AsyncServices
- AsynService819
- Biz_Con
- Completeness Metrics
- Data Profile Configuration
- DataSufficiencyScoreGroup
- Elasticsearch Configuration
- Event Processors
- GDSNReceiverRestBusinessActions
- Global Business Rules
- Inbound Integration Endpoints
- List Process Configurations
- List Processing Configurations
- Match Codes and Matching Algorithm

Asynchronous Service Configuration Type | Log | Status

Description

Name	Value
ID	SDL
Name	SDL Translations
Object Type	Asynchronous Service Configuration Type
Revision	0.1 Last edited by USERK on Thu Oct 29 16:52:23 EDT 2020
Path	Asynchronous Services/SDL Translations

Server Connection Details

Load Handling

Contextual Attributes

Attribute configuration

CaseSupplier
PwrCableMfg
Display
Add Attribute/Group

Excluded Attributes

Attribute configuration

UnitDescription
FeatureList
Add Attribute/Group

Translation Configuration

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.

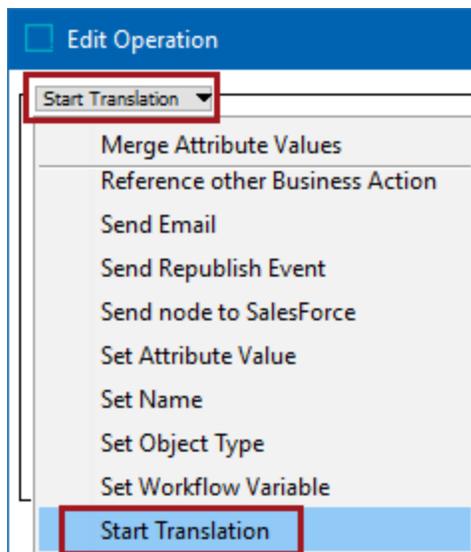
Use the following steps to configure a business action that will start a translation job for asynchronous translation.

Prerequisites

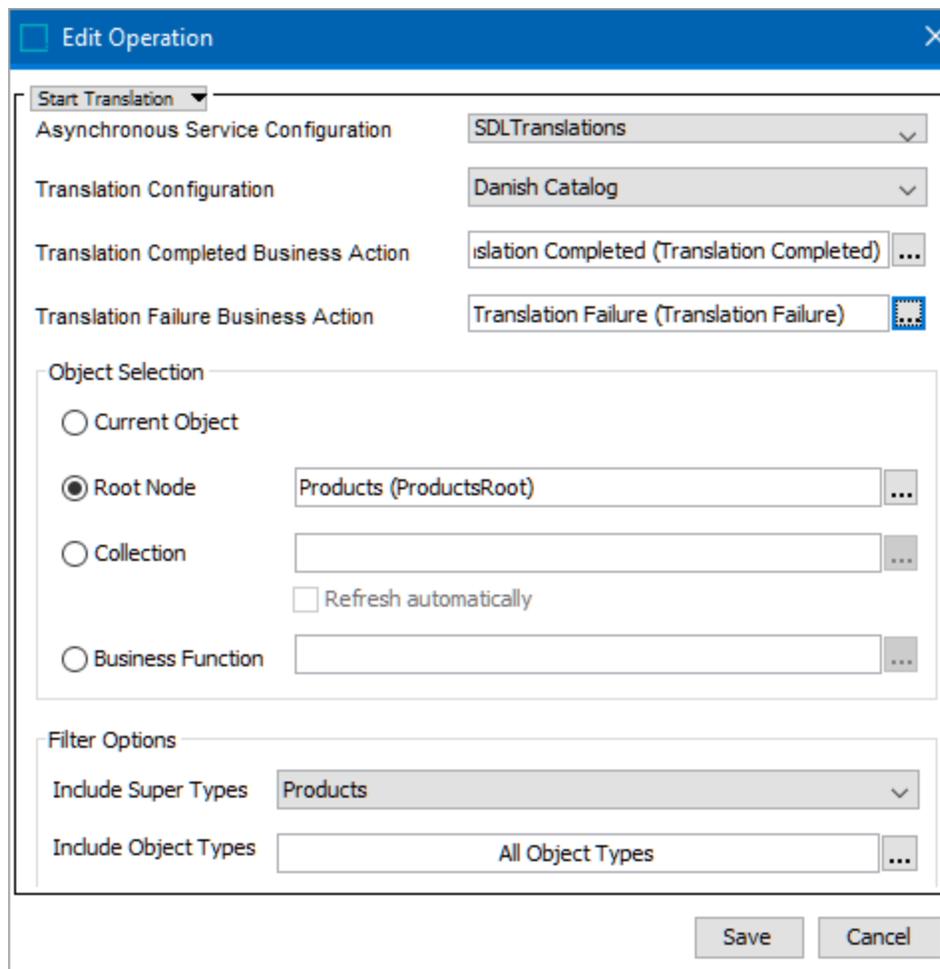
- The 'Translation Completed Business Action' and 'Translation Failure Business Action' can be created prior to being selected for the **Start Translation** Business Action. These actions can move the translation job through the workflow.
- The **Start Translation** Business Action only displays if the functionality for asynchronous translations has been activated on your system. Contact 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:



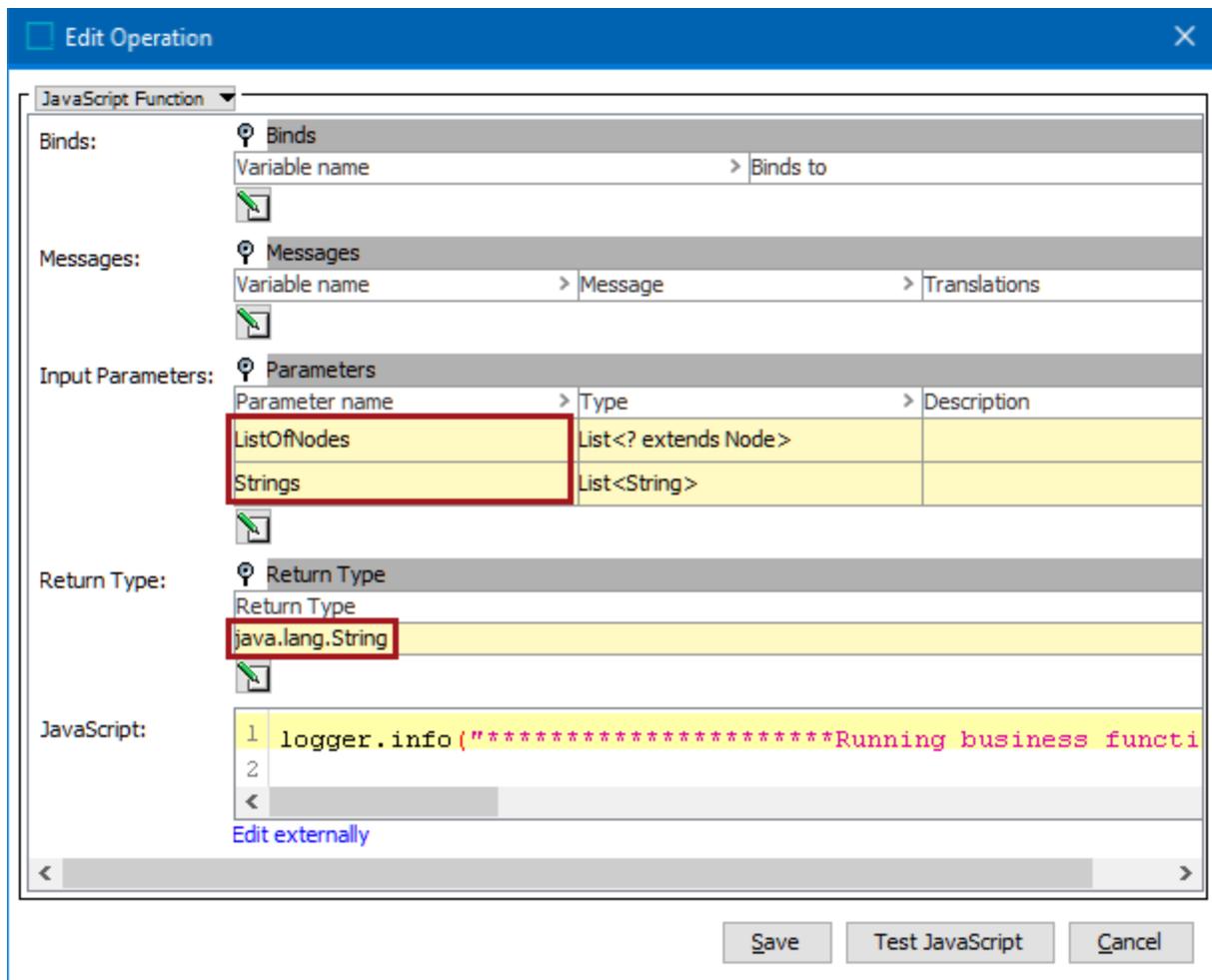
- For **Asynchronous Service Configuration** - Use this dropdown to select the Service Configuration you want to initiate.
- For **Translation Configuration** - Select the Translation Configuration to which the business rule applies. There can be many Translation Configurations for the same Service Configuration 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. Refer to 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, refer to 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.

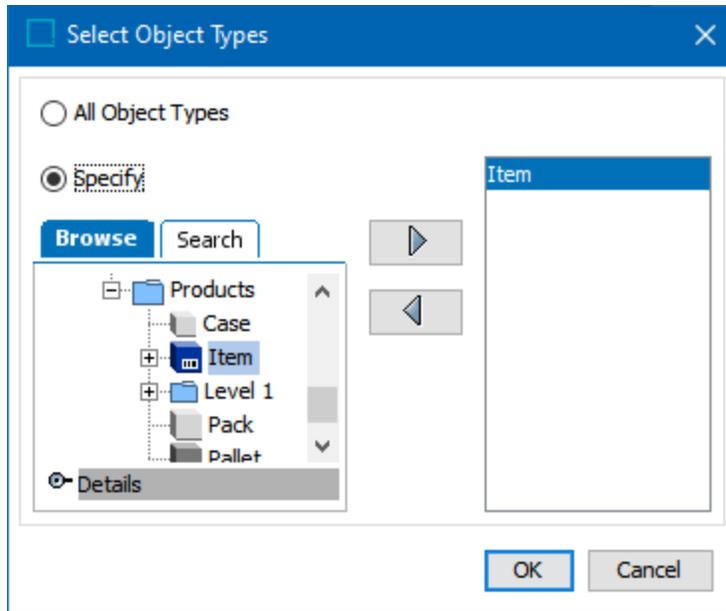


• For **Filter Options** - set the parameters as required:

- **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

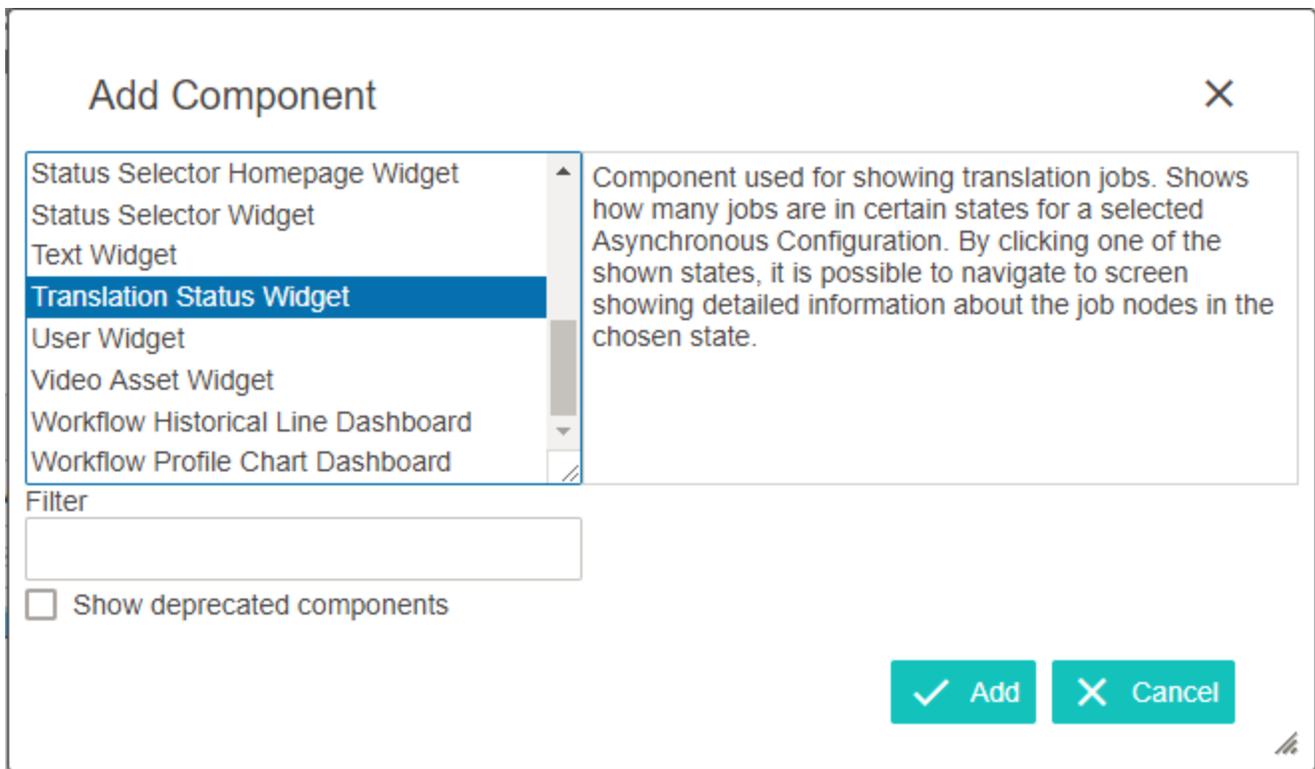
The Translation Status Widget that, once configured along with the necessary screens, enables users to view the number of translation jobs that are in certain states for a selected asynchronous configuration, and select those states for more detailed information.

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, refer to 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.

Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Translation Status Widget Properties

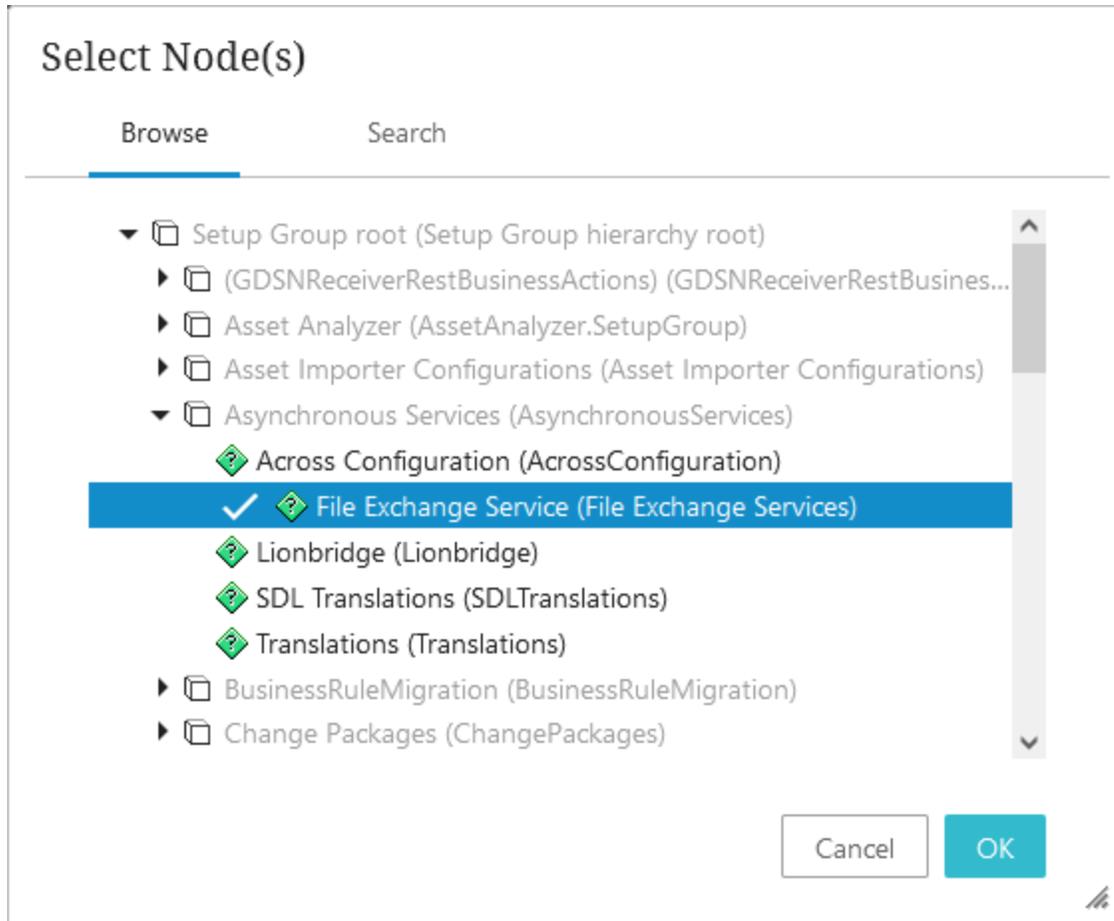
Component Description

Component used for showing translation jobs. Shows how many jobs are in certain states for a selected Asynchronous Configuration. By clicking one of the shown states, it is possible to navigate to screen showing detailed information about the job nodes in the chosen state.

<p>* Asynchronous Service Configuration ID</p>	<input type="text" value=""/> 
Double Width	<input type="checkbox"/>
Auto Refresh Interval	<input type="text" value=""/>
* Result Screen	<input type="text" value="homepage"/> ▼ Add
Title	<input type="text" value=""/>
Title On Hover	<input type="text" value=""/>
Total States Label	<input type="text" value=""/>
Use Title On Hover	<input type="checkbox"/>

Cancel Add

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



- 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.

Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Translation Status Widget Properties

Component Description

Component used for showing translation jobs. Shows how many jobs are in certain states for a selected Asynchronous Configuration. By clicking one of the shown states, it is possible to navigate to screen showing detailed information about the job nodes in the chosen state.

* Asynchronous Service Configuration ID

Double Width

Auto Refresh Interval

* Result Screen

Title

Title On Hover

Total States Label

Use Title On Hover

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.

Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Translation Status Widget Properties

Component Description

Component used for showing translation jobs. Shows how many jobs are in certain states for a selected Asynchronous Configuration. By clicking one of the shown states, it is possible to navigate to screen showing detailed information about the job nodes in the chosen state.

* Asynchronous Service Configuration ID	<input type="text" value="File Exchange Services"/>
Double Width	<input type="checkbox"/>
Auto Refresh Interval	<input type="text"/>
* Result Screen	<input type="text" value="AsyncJobList Screen"/> Add
Title	<input type="text"/>
Title On Hover	<input type="text"/>
Total States Label	<input type="text"/>
Use Title On Hover	<input type="checkbox"/>



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.

For information on how to use the Translation Status widget, refer to the **Using the Translation Status Widget** in this documentation.

Using the Translation Status Widget

The Translation Status Widget comes pre-configured with states, depending on the translation service selected. The states are part of a 'state machine' that is hardcoded for the asynchronous translation services. They cannot be renamed or altered, and they display in the order a translation job will follow. Additionally, there is nothing that an end user needs to do to make a translation job transition from one state to the next, instead it moves automatically. While each asynchronous translation service has its own set of states, many of them overlap. Below is an example and explanation for the SDL service:

SDL Translations TRANSLATION	
All configurations	▼
Waiting	0
Query Translation	0
Translation Export	0
Extract File To Translate	0
Send File To Translate	0
Waiting for Translation	0
Store Translation Result	0
Import Translation	0
Completed	0
Failed	0
Cancelled	0

1. **Waiting:** A translation job is waiting to be processed. Jobs are in this state if the system is busy processing other jobs.
2. **Query Translation:** The system searches for data to extract, based on the translation configurations and the settings made for the 'Start Translation' business action.
3. **Translation Export:** Data is exported from the database.
4. **Extract File To Translate:** The file(s) that must be sent to SDL are created.
5. **Send File to Translate:** The file(s) are sent via API calls to SDL.
6. **Waiting for Translation:** STEP waits for translation jobs to be completed by SDL. STEP polls SDL at regular intervals to view if any jobs are ready for download.
7. **Store Translation Result:** STEP receives the translation file(s) back from SDL and stores it in a Background Process folder on the application server.

8. **Import Translation:** STEP reads the content of the received file(s), imports the data, and updates the translation status.
9. There are three options for the final state:
 - **Completed:** The end state for a successful translation job.
 - **Failed:** If a job errors it will end up in this state.
 - **Cancelled:** During certain scenarios, a translation job can be cancelled by the end user.

Interacting With the Translation Status Widget

Below is an example of how to interact with a Translation Status widget. In this example, the File Exchange service was selected for the Translation Status widget type.

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

File Exchange Service TRANSLATION

All configurations
▼

Waiting	4
Query Translation	3
Translation Export	0
Extract File To Translate	13
Store in Out folder	0
Waiting for Translation	20
Store Translation Result	0
Import Translation	2
Failed	0
Cancelled	0

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, refer to 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:

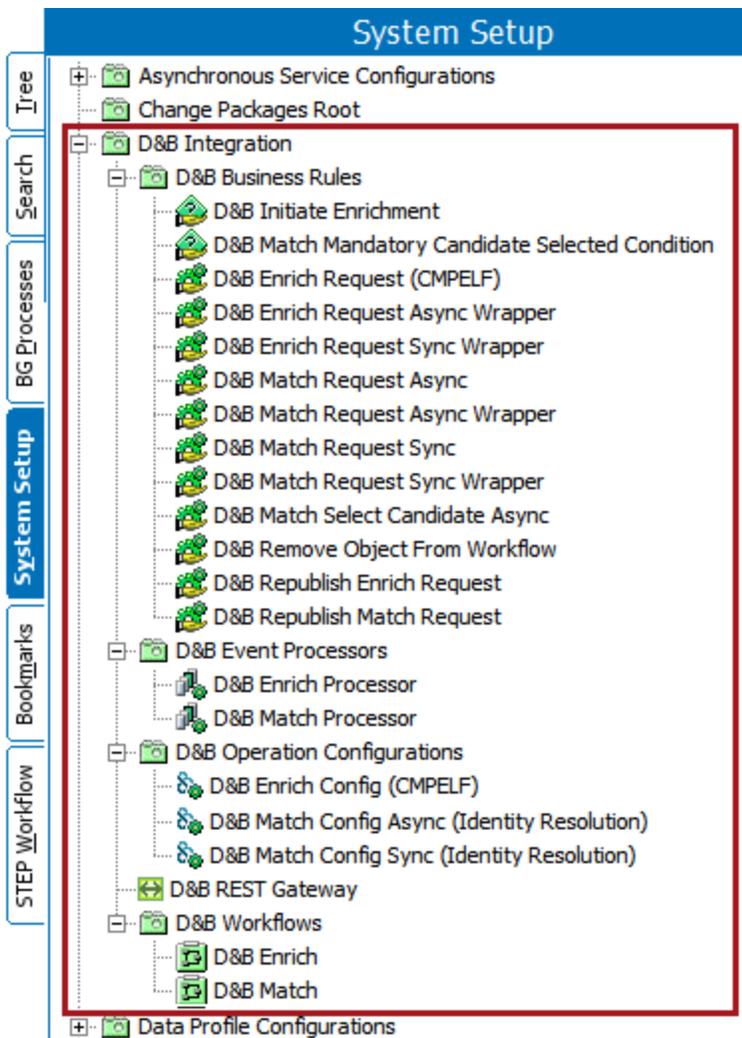
- Business actions for matching, candidate selection, and enriching
- D&B Integration component model

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, refer to 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.

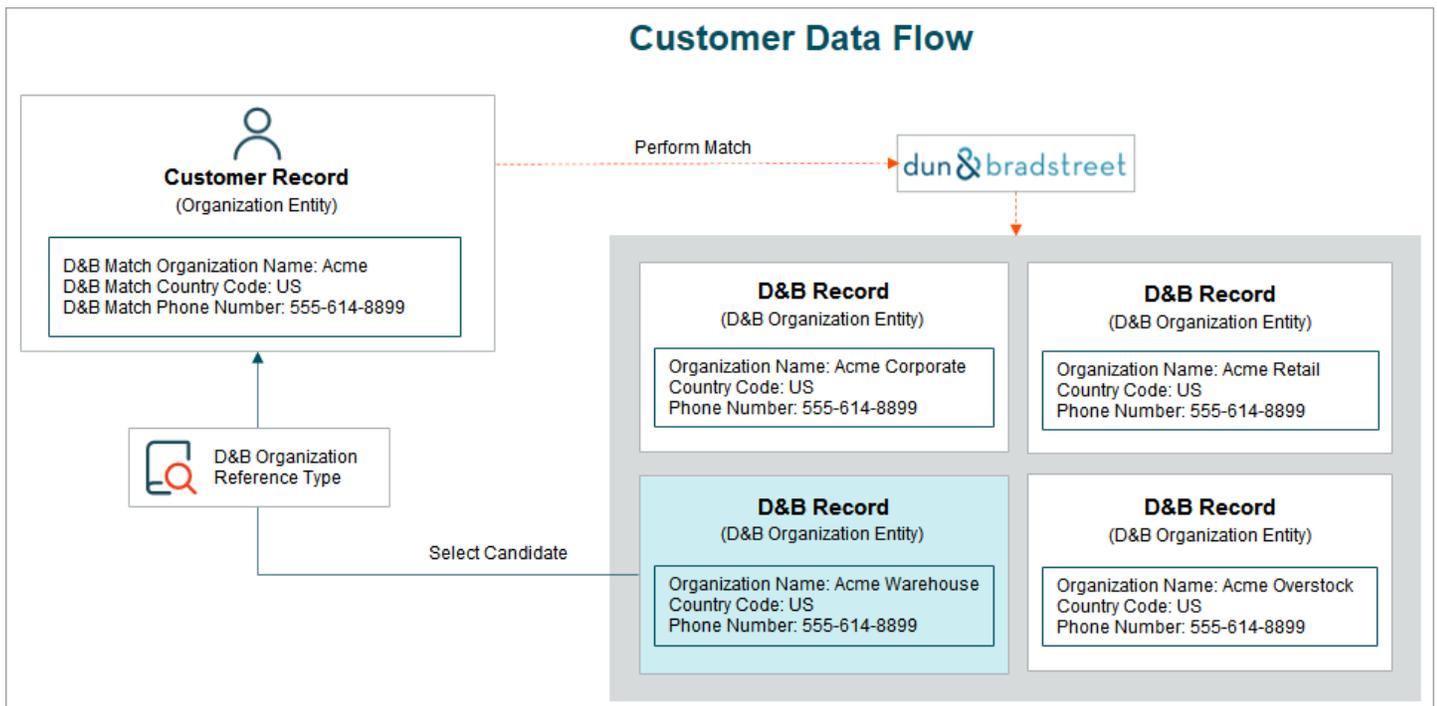
- Users should be familiar with business rules, workflows, event processors, and it is expected that anyone configuring Web UI components is familiar with Web UI designer. Most of these concepts are covered outside of this D&B documentation. For more information, search online help to understand these 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

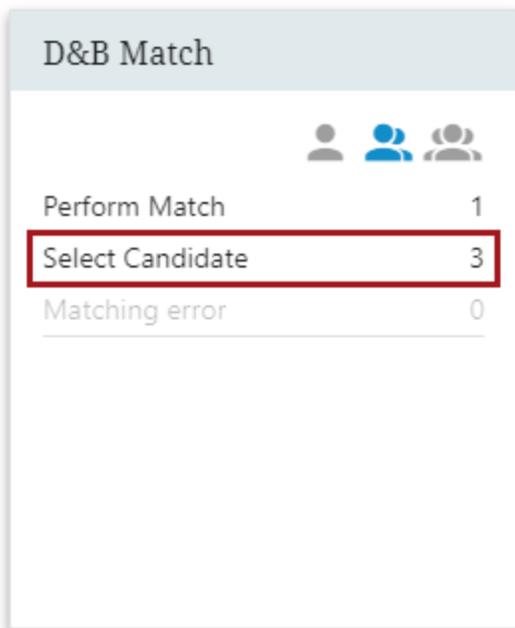
For information about migrating from D&B V2 to Direct+ , refer to the **Dun & Bradstreet Migration: Direct 2.0 to Direct+** topic in the **System Administration** documentation.

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.

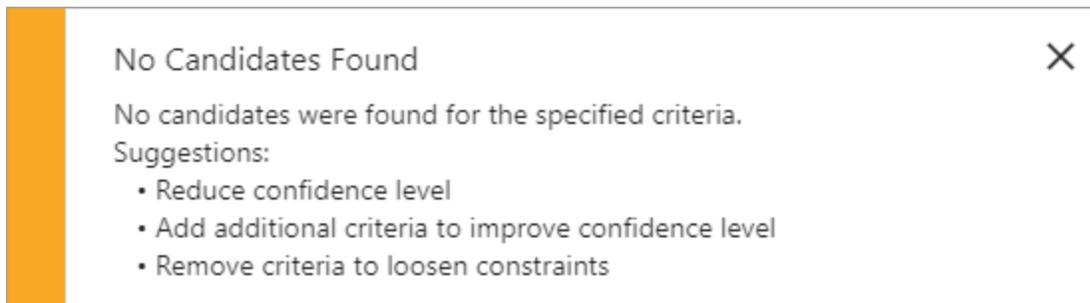
Basic functionality includes:

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.

Modify Search

Organization Name	<input type="text" value="Stibo Systems"/>
* Country Code	<input type="text" value="US - United States of America"/>
Address Line 1	<input type="text"/>
Address Line 2	<input type="text"/>
City	<input type="text"/>
Postal Code	<input type="text"/>
Territory/State	<input type="text"/>
Address County	<input type="text"/>
Phone Number	<input type="text"/>
Email	<input type="text"/>
URL	<input type="text"/>
Registration Number	<input type="text" value=""/> <input type="text"/>
DUNS Number	<input type="text"/>
Match Confidence	<input type="text" value="3"/>
Candidate Maximum Quantity	<input type="text" value="10"/>
	Value between 1-100
Exclusions	<input type="checkbox"/> Non-Headquarters <input type="checkbox"/> Non-Marketable <input type="checkbox"/> Out of Business <input type="checkbox"/> Undeliverable <input type="checkbox"/> Unreachable

- 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.

Current Organization: **Stibo**, Stibo, Success

	Confidence	DUNS number	Name
<input type="checkbox"/>	4	782885800	STIBO DATAGRAPHICS INC
<input type="checkbox"/>	4	5215461	STIBO CATALOG
<input type="checkbox"/>	4	39287762	STIBO CATALOG INC
<input type="checkbox"/>	4	24817956	STIBO CATALOG INC.
<input checked="" type="checkbox"/>	4	131328544	STIBO SYSTEMS, INC.
<input type="checkbox"/>	4	798764411	STIBO DATAGRAPHICS
<input type="checkbox"/>	4	80295529	STIBO DX, INC.
<input type="checkbox"/>	4	59962342	STIBO DATAGRAPHICS INC.
<input type="checkbox"/>	4	621413751	STIBO DATAGRAPHICS INC

A message confirms a candidate selection:

Select candidate ✕

The candidate `step://entity?id=DUNS131328544` has been selected as a match for the current customer `step://entity?id=1186874`

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, refer to the **Object Types and Structures** topic in the **System Setup** 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, refer to the **Reference and Link Types** topic in the **System Setup** 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.

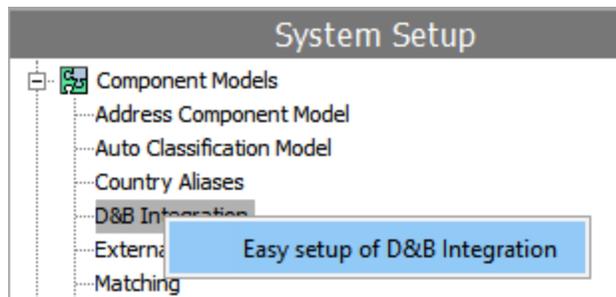
1. Configure the system with your D&B **Consumer Key** and **Consumer Secret** for Direct+. Contact your Stibo Systems account manager or partner manager with questions about this process.

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, refer to 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:

Easy setup of D&B Integration
✕

Running Easy Setup will not make modifications to any existing objects. After applying go to 'D&B Integration' setup group to check the configuration. The Web UI must also be manually configured. See STEP documentation for guidance.

Organization Entity Type	Organization Customer (OrganizationCustomer) ...
D&B Organization Entity Type	D&B Organization (DnBOrganization) ...
D&B Organization Entity Root	D&B Enrichment Records (DnBEnrichmentRecords) ...
D&B Match Candidate Reference Type	D&B Organization Candidates (DnBOrganizationCandidates) ...
D&B Organization Reference Type	D&B Organization (DnBOrganization) ...
D&B Consumer Key	DnBCustomerDirectPlus: 1798678b379446d09f186463a64fbc... ▾

Setup D&B for asynchronous processing

Event Processor Executing User	user7 (USER7) ...
Event Processor Context Language	English US ▾

Setup D&B for synchronous processing

Apply
Cancel

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, refer to 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.

The screenshot shows the 'Business Rule Editor - D&B Match Request Sync Wrapper' with an 'Edit Operation' dialog box open for the 'Execute JavaScript' operation. The 'Binds' section contains the following table:

Variable name	Binds to	Parameter
dnbManager	D&B Integration Manager	
dnbMatchRequest	Business Action	D&B Match Request Sync (DnbMatchRequestSync)
currentObject	Current Object	
webUI	Web UI Context	
dataIssuesReport	Data Issues Report	
matchIntegrationStatusAttribute	Attribute	D&B Matching Integration Status (DnbMatchingInteg)

The JavaScript code is as follows:

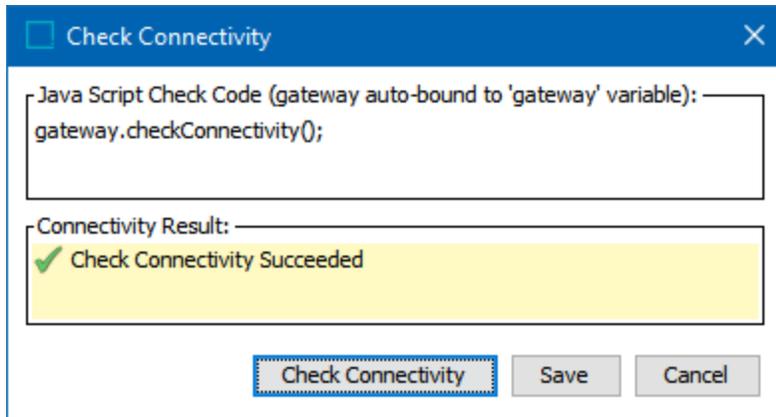
```

1 var dnbResult = dnbManager.executeBusinessActionWithDnbIntegrationContext(dnbM
2 if (dnbResult.getIntegrationException()) {
3     dataIssuesReport.addError(dnbResult.getIntegrationException().getMessage(
4 }
5 else {
6     webUI.navigate('D&B Match Candidates ScrV2', currentObject);
7 }
8 return dataIssuesReport;

```

A red arrow points to the string 'D&B Match Candidates ScrV2' in the JavaScript code. At the bottom of the dialog, there are 'Save', 'Test JavaScript', and 'Cancel' buttons.

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, refer to 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.

- 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

- 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, refer to the **D&B Matching** topic.

- 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, refer to the **Editing a Business Rule or Function** topic in the **Business Rules** documentation.

Name	Value
ID	DnBMatchRequestSync
Name	D&B Match Request Sync
Revision	0.3 Last edited by SOAM on Wed Jun 02 14:29:28 CEST 2021
Description	
Type	Action
Valid Object Types	Supplier, Organization Customer
On Approve	Not Executed
Scope	Global
Run as privileged	<input checked="" type="checkbox"/>

Operations Dependencies Applies if

D&B | D&B Match: D&B Match Config Sync (Identity Resolution), D&B REST Gateway

- 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, refer to 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.

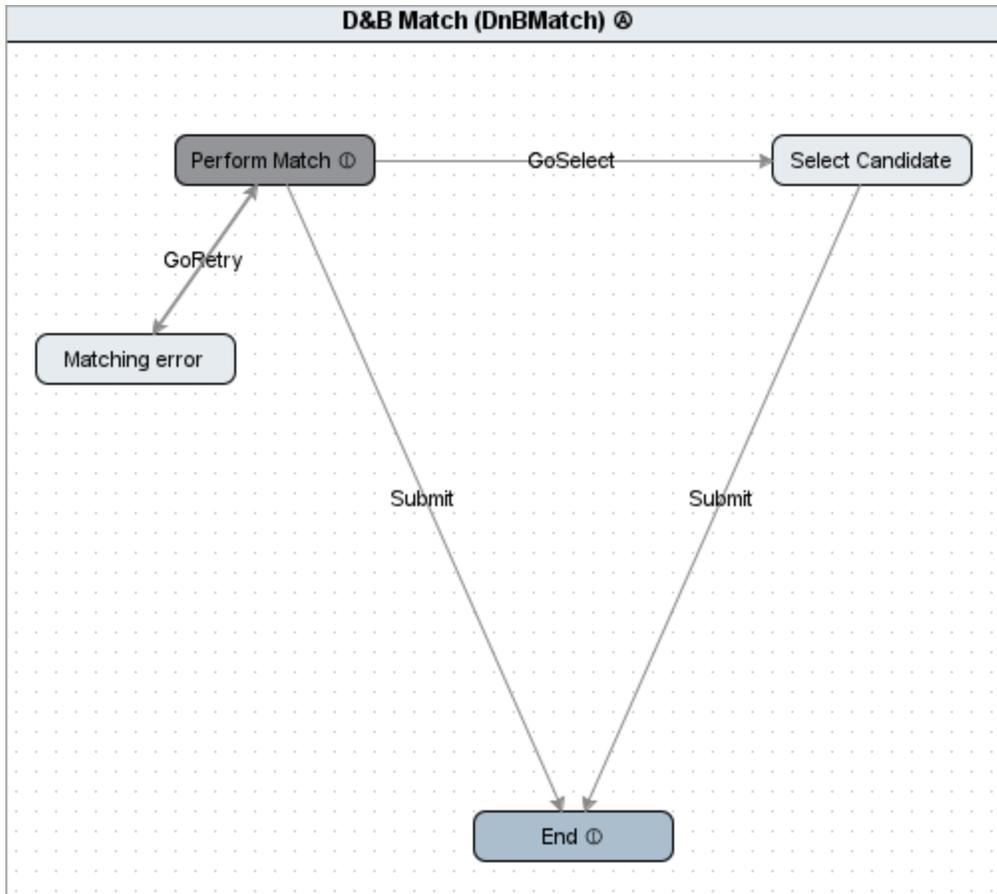
For **asynchronous** processing, matching is started by the initiation of the organization entity (as designated in the D&B Integration component model) into the D&B Match workflow. Standard methods are available for initiation including upon entity creation, through manual initiation, or by a variety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

System Setup

- Attribute Groups
- Attribute Transformations
- Action Sets
- Contexts
- InDesign Queue
- Lists of Values / LOVs
- Asset Analyzer
 - Asset Import Configurations
 - Asynchronous Services
 - AsynServices
 - AsynService819
- Biz_Con
- Completeness Metrics
- D&B Integration
 - D&B Business Rules
 - D&B Event Processors
 - D&B Operation Configurations
 - D&B Enrich Config (CMPELF)
 - D&B Match Config Async (Identity Resolution)
 - D&B Match Config Sync (Identity Resolution)**
 - D&B REST Gateway
 - D&B Workflows
- Data Profile Configuration
- DataSufficiencyScoreGroup
- Elasticsearch Configuration
- Event Processors
- Gateway Endpoints
- GDSNReceiverRestBusinessActions
- Global Business Rules
- Inbound Integration Endpoints
- List Process Configurations
- List Processing Configurations
- Match Codes and Matching Algorithms
- Metrics
- Outbound Integration Endpoints
- PDX Inbound Setup Group
- PDX Outbound Setup Group

< D&B Match Config Sync (Identity Resolution) rev.0.1 - D&B Operation Configuration ->

D&B Operation Configuration Log | Status

Description

Name	Value
ID	DnBMatchOpConfigSync
Name	D&B Match Config Sync (Identity Resolution)
Object Type	D&B Operation Configuration
Revision	0.1 Last edited by USER8 on Mon May 24 09:49:55 EDT 2021
Path	D&B Integration/D&B Operation Configurations/D&B Match Config Sync (Identity Resolution)

Operation Parameters and Flags

Parameter	Value
D&B Product	Identity Resolution (Cleanse and Match)
Default Country Code	US
Customer ID	<Customer Identification Text>
Matching In Language	English (en-US)
Candidate Maximum Quantity	30
Match Confidence	4
Perform cleanse and standard...	<input type="checkbox"/>
Autolink Threshold	9
Integration Status Attribute	D&B Matching Integration Status (DnBMatchingIntegrationStatus) ...
Integration Error Code Attribute	D&B Matching Integration Error Code (DnBMatchingIntegrationErrorCode) ...
Integration Error Description ...	D&B Matching Integration Error Description (DnBMatchingIntegrationErrorDescription) ...
Candidate Selected Business ...	D&B Enrich Request Sync Wrapper (DnBEnrichRequestSyncWrapper) ...
No Candidate Selected Busine...	...

Value	Status
Exclude Unreachable	<input type="checkbox"/>
Exclude Non HeadQuarters	<input type="checkbox"/>
Exclude Out of Business	<input checked="" type="checkbox"/>
Exclude Undeliverable	<input type="checkbox"/>
Exclude Non Marketable	<input type="checkbox"/>

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).

© Stibo Systems - Internal - Release 2023.3 (11.3) - September 2023

81

System Setup

- Attribute Groups
- Attribute Transformations
- Action Sets
- Contexts
- InDesign Queue
- Lists of Values / LOVs
- Asset Analyzer
- Asset Import Configurations
- Asynchronous Services
- AsyncServices
- AsynService819
- Biz_Con
- Completeness Metrics
- D&B Integration
 - D&B Business Rules
 - D&B Event Processors
 - D&B Operation Configurations
 - D&B Enrich Config (CMPELF)
 - D&B Match Config Async (Identity Resolution)**
 - D&B REST Gateway
 - D&B Workflows
- Data Profile Configuration
- DataSufficiencyScoreGroup
- Elasticsearch Configuration
- Event Processors
- Gateway Endpoints
- GDSNReceiverRestBusinessActions
- Global Business Rules
- Inbound Integration Endpoints
- List Process Configurations
- List Processing Configurations
- Match Codes and Matching Algorithms
- Metrics
- Outbound Integration Endpoints
- PDX Inbound Setup Group
- PDX Outbound Setup Group
- PolicyQualScoreSetupGrpRt
- Status Flags

D&B Match Config Async (Identity Resolution) rev.0.1 - D&B Operation Config

D&B Operation Configuration | Log | Status

Description

Name	Value
ID	DnBMatchOpConfig
Name	D&B Match Config Async (Identity Resolution)
Object Type	D&B Operation Configuration
Revision	0.1 Last edited by USER8 on Mon May 24 08:47:37 EDT 2021
Path	D&B Integration/D&B Operation Configurations/D&B Match Config Async (Identity Resolution)

Operation Parameters and Flags

Parameter	Value
D&B Product	Identity Resolution (Cleanse and Match)
Default Country Code	US
PRODUCT_ID	Identity Resolution (Cleanse and Match)
Customer ID	
Matching In Language	English (en-US)
Candidate Maximum Quantity	30
Match Confidence	4
Perform cleanse and standard...	<input type="checkbox"/>
Autolink Threshold	9
Integration Status Attribute	D&B Matching Integration Status (DnBMatchingIntegrationStatus) ...
Integration Error Code Attribute	D&B Matching Integration Error Code (DnBMatchingIntegrationErrorCode) ...
Integration Error Description ...	D&B Matching Integration Error Description (DnBMatchingIntegrationErrorDescription) ...
Candidate Selected Business ...	D&B Match Select Candidate Async (DnBSelectCandidateActionAsync) ...
No Candidate Selected Busine...	D&B Match Select Candidate Async (DnBSelectCandidateActionAsync) ...

Value | **Status**

Exclude Unreachable	<input type="checkbox"/>
Exclude Non HeadQuarters	<input type="checkbox"/>
Exclude Out of Business	<input type="checkbox"/>
Exclude Undeliverable	<input type="checkbox"/>
Exclude Non Marketable	<input type="checkbox"/>

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, refer to 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.

Stibo rev.0.1 - References

Organization Customer | Data Containers | **References** | Referer

D&B References

Reference Type	Target
D&B Organization	STIBO SYSTEMS, INC.
D&B Organization Candidates	

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.

Stibo rev.0.1 - References	
Organization Customer	Data Containers
References	
Referenced B	
D&B References	
Reference Type	Target
> D&B Organization +	
	STIBO CATALOG
	STIBO CATALOG INC
	STIBO CATALOG INC.
	STIBO DATAGRAPHICS
> D&B Organization Candidates +	STIBO DATAGRAPHICS INC
	STIBO DATAGRAPHICS INC
	STIBO DATAGRAPHICS INC.
	STIBO DX, INC.
	STIBO SYSTEMS, INC.

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.

D&B Match	
Perform Match	1
Select Candidate	3
Matching error	0

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.

Current Organization: **Stibo**, Success

	Confidence	DUNS number	Name
<input checked="" type="checkbox"/>	4	39287762	STIBO CATALOG INC
<input type="checkbox"/>	4	59962342	STIBO DATAGRAPHICS INC.
<input type="checkbox"/>	4	80295529	STIBO DX, INC.
<input type="checkbox"/>	4	131328544	STIBO SYSTEMS, INC.

- 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. Refer to the **D&B Error Handling with Integration Status** topic for more information.

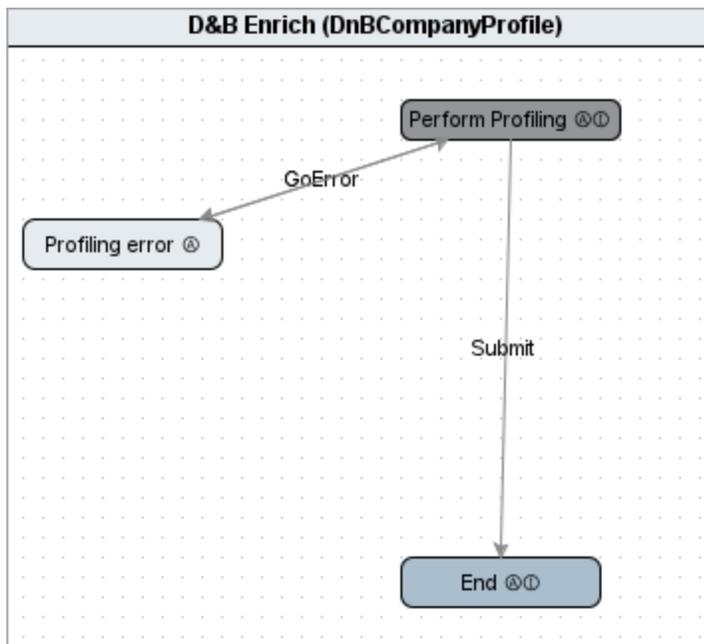
D&B Enriching

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, refer to the **Event Processors** documentation.



Important: Profiling errors that occur during the process must be handled via the workflow 'Profiling error' state. Refer to 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.

Refer to [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)

The screenshot shows the 'System Setup' sidebar on the left and the 'D&B Enrich Config (CMPELF) rev.0.1 - D&B Operation Configuration' main panel on the right. The main panel contains two tables: 'D&B Operation Configuration' and 'Operation Parameters and Flags'.

D&B Operation Configuration	
Name	Value
ID	DnBCompanyProfileOpConfig
Name	D&B Enrich Config (CMPELF)
Object Type	D&B Operation Configuration
Revision	0.1 Last edited by USER8 on Mon May 24 08:47:37 EDT 2021
Path	D&B Integration/D&B Operation Configurations/D&B Enrich Config (CMPELF)

Operation Parameters and Flags	
Parameter	Value
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 (DnBEnrichmentIntegrationStatus) ...
Integration Error Code Attribute	D&B Enrichment Integration Error Code (DnBEnrichmentIntegrationErrorCode) ...
Integration Error Description ...	D&B Enrichment Integration Error Description (DnBEnrichmentIntegrationErrorDesc) ...
Integration JSON Response Data ...	D&B JSON Company Profile Data (DnBJSONCompanyProfileData) ...

For information on the matching process, refer to 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. Refer to the **Web UI Getting Started** section of the **Web User Interfaces** documentation for more information on these topics.

Additionally, users should be familiar with the D&B integration setup and processes. For more information, refer to 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. Refer to 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. Refer to the **D&B Asynchronous Match Candidates Screen** topic.
- **Action buttons** to manually start the enrichment process via the D&B Enrich workflow. Refer to 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. Refer to the **D&B Synchronous Match Candidates Screen** topic.
- **Action button** to manually starting the match process (and optionally one to start enrichment). Refer to 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, refer to 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.

Current Organization: **stibos**, stibos

	Confidence	DUNS number	Name	D&B Match Orga...	DnBOperatingSt...	DnBStandaloneO...
<input type="checkbox"/>	4	63777671	STB SOLUTIONS LLC		Out of business	true
<input type="checkbox"/>	4	61460721	STABO SCANDINAVIA...		Active	false
<input type="checkbox"/>	4	33691574	STB SERVICE		Active	true
<input type="checkbox"/>	4	101034544	ST BASIL ACADEMY		Active	false
<input type="checkbox"/>	4	930420679	STBS IMPORT AND E...		Active	true

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.

Add Screen

Screen ID

- Compare Records Screen
- Create Object In Workflow
- Create Product Override Screen
- D&B Match Candidates Screen
- Dashboard

Screen that displays D&B match candidates.

Filter

Show deprecated components

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.

D&B Match Candidates Screen Properties

Component Description Screen that displays D&B match candidates.

* D&B Matching Business Action	<input style="width: 100%; border: 1px solid #ccc;" type="text" value="DnBMatchRequestAsync"/>
Displayed Attributes	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> D&B Match Organization Name DnBOperatingStatus DnBStandaloneOrganizationIndicator D&B Primary Address Formatted D&B Assessment </div> <div style="margin-top: 5px; display: flex; gap: 5px;"> <input type="button" value="Add..."/> <input type="button" value="Remove"/> <input type="button" value="Up"/> <input type="button" value="Down"/> </div>
Organization Attributes	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> Legal Name D&B Match Address Line 1 </div> <div style="margin-top: 5px; display: flex; gap: 5px;"> <input type="button" value="Add..."/> <input type="button" value="Remove"/> <input type="button" value="Up"/> <input type="button" value="Down"/> </div>
Organization Title Attribute	<input style="width: 100%; border: 1px solid #ccc;" type="text" value="LegalName"/> <input style="margin-left: 5px;" type="button" value="..."/> <input style="margin-left: 5px;" type="button" value="Clear"/>

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.

Main Properties

Component Description MAIN is used for configuring the overall behaviour of the Web UI. For example by setting up conditional mappings it is possible to decided the behaviour when navigating the Web UI. In addition the different side panels (left, right, top, bottom) and Corner bar can be configured on MAIN.

Mappings

DataProfile (Data Profile Condition)

DnB Match Candidates Screen (Flow = DnBMatch & State = SelectCandidate)

userdetails (Is User)

Details-Individual (ObjectType = IndividualCustomer)

Add...
Edit...
Remove
Up
Down

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, refer to 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.

Current Organization: **stibos**, stibos

	Confidence	DUNS number	Name	D&B Match Orga...	DnBOperatingSt...	DnBStandaloneO...
<input type="checkbox"/>	4	63777671	STB SOLUTIONS LLC		Out of business	true
<input type="checkbox"/>	4	61460721	STABO SCANDINAVIA...		Active	false
<input type="checkbox"/>	4	33691574	STB SERVICE		Active	true
<input type="checkbox"/>	4	101034544	ST BASIL ACADEMY		Active	false
<input type="checkbox"/>	4	930420679	STBS IMPORT AND E...		Active	true

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.

Add Screen

Screen ID

DnBCandidatesScreen

Company Hierarchy Screen

Compare Records Screen

Create Object In Workflow

Create Product Override Screen

D&B Match Candidates Screen

Display Relations Screen

Screen that displays D&B match candidates.

Filter

Show deprecated components

Cancel
Add

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.

Configuration Web UI style

DnB Sync Match Car ▾ Save Close New... Delete Rename Save as...

D&B Match Candidates Screen Properties

Component Description Screen that displays D&B match candidates.

* D&B Matching Business Action DnBMatchRequestSync

Displayed Attributes

- DnBMatchOrganizationName
- DnBOperatingStatus
- DnBStandaloneOrganizationIndicator
- DnBAssessment

Add... Remove Up Down

Organization Attributes

- DnBOrganizationName
- DnBMatchAddressLine1

Add... Remove Up Down

Organization Title Attribute DnBOrganizationPrimaryName ... Clear

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.

Main Properties

Component Description MAIN is used for configuring the overall behaviour of the Web UI. For example by setting up conditional mappings it is possible to decided the behaviour when navigating the Web UI. In addition the different side panels (left, right, top, bottom) and Corner bar can be configured on MAIN.

Mappings

- List-Policy (Policy List Condition)
- Details-HoldingCompany (ObjectType = HoldingCompany)
- DnB Match Candidates Screen Sync (Flow = DnBMatch & State = SelectCandidate)
- Details-SalesManager (ObjectType = SalesManager)
- Details ProductSubFamily (ObjectType = ProductSubFamily)
- Details SKU (ObjectType = Product)

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, refer to 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** 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.)

Status Selector Homepage Widget Properties [go to parent](#)

Component Description
Homepage widget for showing states and number of items in each state in a particular workflow. By clicking one of the shown states, it is possible to navigate to a screen with a list of tasks in that state. Appropriate screen types to use as the ResultScreen are the "TaskList" and the "AdvancedSearchScreen" showing detailed information about the items in the chosen state.

Auto Refresh Interval:

Component Title:

Initiate Label:

Initiate Screens:

* Result Screen:

Show Collection Filter:

Collection Top Nodes:

Show Initiate:

Status Flags Enabled:

Show Status Flag Headers:

Show Total:

* States:
SelectCandidate
MatchingError
ApprovalWorkflow | End

Total Label:

* Workflow:

▶ Task Mode

▶ Advanced

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

D&B Match

Perform Match	1
Select Candidate	4302
Matching error	2

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.

Workflow Status Selector	Count
D&B Match	
Perform Match	1
Select Candidate	4302
Matching error	2

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, refer to 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, refer to 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.

Monster Mini Viby ORGANIZATION CUSTOMER • ID: 615902

Overview **D&B** Traceability History Company Hierarchy Contact Network

Organization Details

Main Address

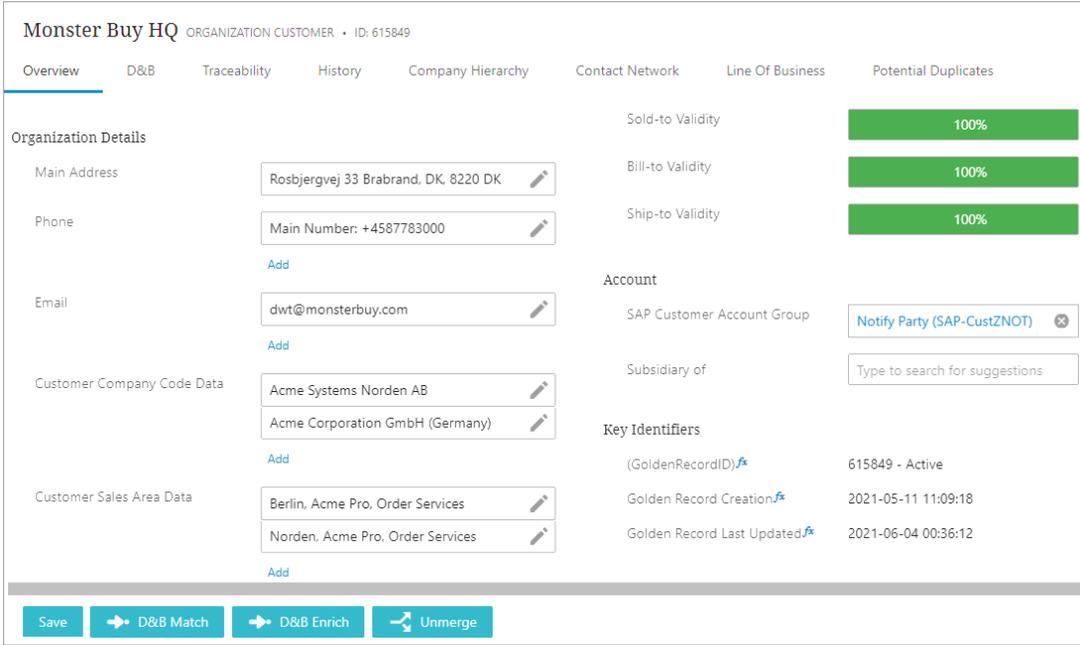
Phone

[Add](#)

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'.

Properties (edited)

Configuration Web UI style

Details-Organization ▾ Save Close New... Delete Rename Save as...

Run Business Action Properties [go to parent](#)

Component Description
By incorporating custom business rules using a special Web UI bind, this component provides additional flexibility in how users navigate through and complete tasks in the Web UI. This component also enables custom messaging in popup notifications that display when the configured button is clicked.

* Business Action DnBMatchRequestSyncWrapper

Label D&B Match Now

Add Value(s) Popup Label

Context Help

Child Component

Select Node(s)

Browse Search

- D&B Match Request Async Wrapper (DnBMatchRequestAs...
- D&B Match Request Sync (DnBMatchRequestSync)
- D&B Match Request Sync Wrapper (DnBMatchRequestSyn...
- D&B Remove Object From Workflow (DnBRemoveNodeAc...
- D&B Republish Enrich Request (DnBRepublishProfilingActi...

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'.

Properties (edited)

Configuration Web UI style

Details-Organization ▾ Save Close New... Delete Rename Save as...

Run Business Action Properties [go to parent](#)

Component Description
By incorporating custom business rules using a special Web UI bind, this component provides additional flexibility in how users navigate through and complete tasks in the Web UI. This component also enables custom messaging in popup notifications that display when the configured button is clicked.

* Business Action: DnBEnrichRequestSyncWrapper

Label: D&B Enrich Now

Add Value(s) Popup Label

Context Help

Save Before Executing

Enable On Workflow Only

Node Picker Dialog

Node Picker Position

Icon

Button Type

Select Node(s)

Browse Search

- D&B Enrich Request (CMPELF) (DnBEnrichRequest)
- D&B Enrich Request Async Wrapper (DnBEnrichRequestAs...)
- D&B Enrich Request Sync Wrapper (DnBEnrichRequestSyn...)
- D&B Match Request Async (DnBMatchRequestAsync)
- D&B Match Request Async Wrapper (DnBMatchRequestAs...)

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'.

Edit component

Run Business Action Properties

Component Description By incorporating custom business rules using a special Web UI bind, this component provides additional flexibility in how users navigate through and complete tasks in the Web UI. This component also enables custom messaging in popup notifications that display when the configured button is clicked.

* Business Action	DnBMatchRequestAsyncWrapper
Label	D&B Match
Add Value(s) Popup Label	i18n.stibo.BusinessActionWithWebUIBindPopUp.label

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

Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Start Workflow From Grid Properties

Component Description	An action that can start a STEP Workflow for the selected nodes in a NodeList	
Button Label	<input type="text" value="D&B Enrich"/>	
Css Class	<input type="text" value="SubmitButton"/>	
Custom Icon	<input type="text"/>	<input type="button" value="Reset"/>
Context Help	<input type="text" value="i18n.stibo.portal.workflow.Start.ToolTip"/>	
* Workflow	<input type="text" value="DnBCompanyProfile"/>	

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'.

Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Start Workflow From Grid Properties

Component Description An action that can start a STEP Workflow for the selected nodes in a NodeList

Button Label

D&B Enrich

Css Class

SubmitButton

Custom Icon

...

Context Help

i18n.stibo.portal.workflow.Start.ToolTip

* Workflow

DnBCompanyProfile ▼

Cancel

Add

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
DnBOrganizationPrimaryAddress	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

Data Container ID	Attribute ID	JSON Path, Matching
	DnBStreetAddressLine	matchCandidates.organization.primaryAddress.streetAddress.li ne1 matchCandidates.organization.primaryAddress.streetAddress.li ne2
DnBOrgMailingAdresses	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.postalCodeExt ension
	DnBStreetAddressLine	matchCandidates.organization.mailingAddress.streetAddress.li ne1 matchCandidates.organization.mailingAddress.streetAddress.li ne2
DnBOrganizationIdentification	DnBOrganizationIdentificationNumberType	matchCandidates.organization.registrationNumbers.typeDescri ption matchCandidates.organization.registrationNumbers.typeDnBC ode
	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

Attribute ID	JSON Path, Enriching
	organization.corporateLinkage.familytreeRolesPlayed.dnbcode
DnBFemaleOwnedCertifiedDate	organization.socioEconomicInformation.femaleOwnedCertifiedDate
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

Attribute ID	JSON Path, Enriching
	organization.dunsControlStatus.operatingStatus.dnbCode
DnBOperationsText	organization.operations.description
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

Data Container ID	Attribute ID	JSON Path, Enriching
	DnBSalesRevenueAmount	organization.competitors.salesRevenue
	DnBDUNSNumber	organization.competitors.duns
	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
	DnBCountryOfficialName	organization.corporateLinkage.domesticUltimate.primaryAddress.addressCountry.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

Data Container ID	Attribute ID	JSON Path, Enriching
	DnBStreetAddressLine	organization.corporateLinkage.globalUltimate.primaryAddress.streetAddress.line1 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
DnBHeadquartersPrim Address	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

Data Container ID	Attribute ID	JSON Path, Enriching
	DnBAddressRegionName	organization.corporateLinkage.headQuarter.primaryAddress.addressRegion.name
	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

Data Container ID	Attribute ID	JSON Path, Enriching
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	
DnBNumberOfEmployees	DnBNumberOfEmployeesValue	organization.numberOfEmployees.value
	DnBNumberOfEmployeesMinimumValue	organization.numberOfEmployees.minimumValue
	DnBNumberOfEmployeesMaximumValue	organization.numberOfEmployees.maximumValue
	DnBEmployeeFiguresDate	organization.numberOfEmployees.employeeFiguresDate
	DnBInformationScope	organization.numberOfEmployees.informationScopeDescription organization.numberOfEmployees.informationScopeDnbCode

Data Container ID	Attribute ID	JSON Path, Enriching
	DnBReliability	organization.numberOfWorkers.reliabilityDescription organization.numberOfWorkers.reliabilityDnBCode
	DnBEmployeeCategory	organization.numberOfWorkers.employeeCategories.employmentBasisDescription organization.numberOfWorkers.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
	DnBMetropolitanStatAreaUS CensusCode	organization.primaryAddress.statisticalArea.cbsaName
	DnBLanguageCode	organization.primaryAddress.language.description organization.primaryAddress.language.dnbCode
	DnBCountyFipsCode	organization.primaryAddress.addressCounty.fipsCode
	DnBContinentalRegionName	organization.primaryAddress.continentalRegion.name
	DnBMetropolitanStatAreaUS CensusNumCode	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
DnBParentPrimAddresses	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
DnBOrgMailingAddresses	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, refer to 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** 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+, refer to 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 (cmpekl) v2**

Corporate Linkage

- **Corporate Linkage Alternate Linkage (lnkalt)**
- **Corporate Linkage Global Beneficial Ownership (lnkgbo)**

- Corporate Linkage Minority Linkage (Inkmin)
- Corporate Linkage Upward Corporate Linkage (Inkupd)

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. Refer to 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’:

System Setup

- Attribute Groups
- Attribute Transformations
- Action Sets
- Contexts
- InDesign Queue
- Lists of Values / LOVs
- Asset Analyzer
- Asynchronous Service Configurations
- Change Packages Root
- D&B Integration
 - D&B Business Rules
 - D&B Event Processors
 - D&B Operation Configurations
 - D&B Enrich Config (CMPELF)
 - D&B Enrich Config (CMPELK)**
 - D&B Match Config Async (Identity Resolution)
 - D&B Match Config Sync (Identity Resolution)
 - D&B REST Gateway
- D&B Workflows
- Data Profile Configurations
- Elastic Search Configurations
- Event Processor Root
- Global Business Rules

D&B Enrich Config (CMPELK) rev.0.1 - D&B Operation Configuration

D&B Operation Configuration	
Log Status	
Description	
Name	Value
ID	DnBEnrichConfigCMPELK
Name	D&B Enrich Config (CMPELK)
Object Type	D&B Operation Configuration
Revision	0.1 Last edited on Wed May 19 13:56:12 CEST 2021
Path	D&B Integration/D&B Operation Configurations/D&B Enrich Config (CMPELK)
Operation Parameters and Flags	
Parameter	Value
D&B Product	Company Profile, Linkage, and Executives (cmpelk)
Version ID	v2
Trade Up	No trade up
Customer ID	<Customer Identification Text>
Integration Status Attribute	D&B Enrichment Integration Status CMPELK (DnBEnrichmentIntegrationStatusCMPELK)
Integration Error Code Attribute	D&B Enrichment Integration Error Code CMPELK (DnBEnrichmentIntegrationErrorCodeCMPELK)
Integration Error Description ...	D&B Enrichment Integration Error Description CMPELK (DnBEnrichmentIntegrationErrorDescCMPELK)
Integration JSON Response D...	D&B JSON Enrichment CMPELK (DnBJSONEnrichmentCMPELK)

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 (refer to 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** button 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

Integration Status	D&B Error Codes + Description
Unknown error	All other D&B errors
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. Refer to 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:

(617330) ORGANIZATION CUSTOMER • ID: 617330

Overview **D&B** Traceability History Company Hierarchy Contact Network Line Of Business Potential Duplicates

D&B Matching

Legal Name

D&B Match DUNS Number

D&B Match Phone Number

D&B Match Email

D&B Match URL

D&B Match Registration Number Type

D&B Match Registration Number

D&B Match Address Line 1

D&B Match Postal Code

D&B Match Primary Town

D&B Match Country Code

D&B Matching Selected Candidate

* D&B Selected Candidate [STIBO SYSTEMS, INC. \(DUNS131328544\)](#)

Decided No Match

D&B Matching Status

D&B Matching Integration Status **Success**

D&B Matching Integration Error Code

D&B Matching Integration Error Description

D&B Enriching Status

▼ Status - D&B Organization - STIBO SYSTEMS, INC.

D&B Enrichment Integration Status **Success**

CMPELF

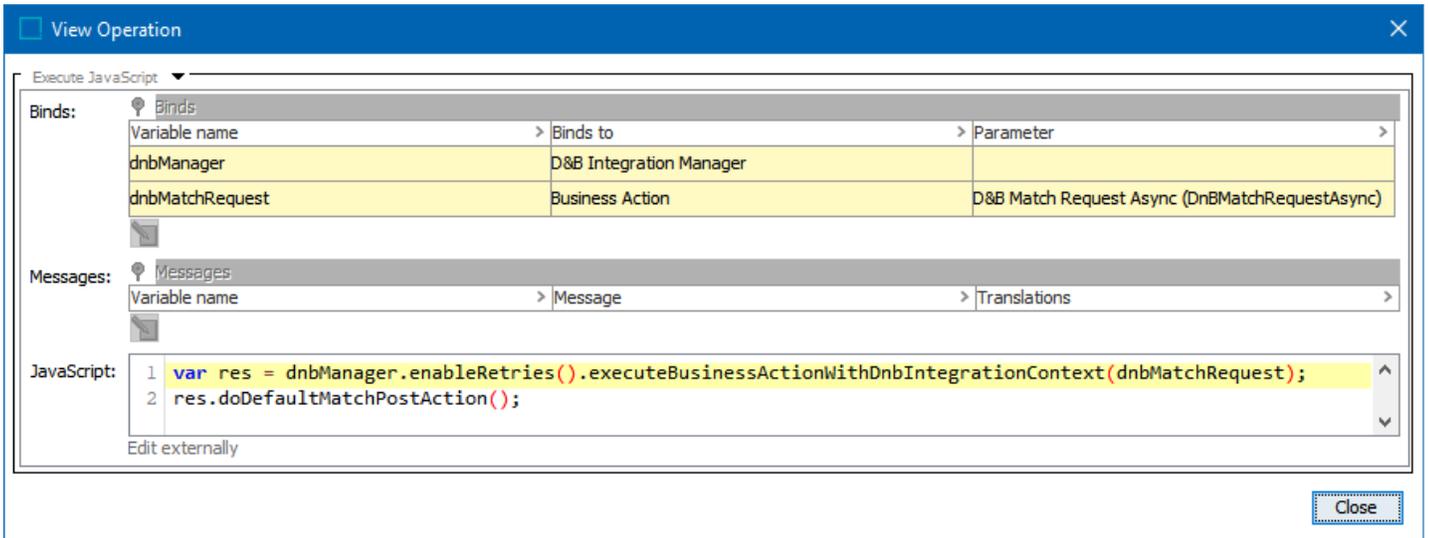
D&B Enrichment Integration Error Code CMPELF

D&B Enrichment Integration Error Description CMPELF

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(), refer to 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	Yes, 3 retries		Matching error	Profiling error

Integration Status	Default Retry Strategy	Stop Event Processor	Matching Workflow State	Enriching Workflow State
- resubmit				
Enrichment result not available				Profiling error
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, 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 wizard)

Note: Experian throughput limitations are described in the **Experian Email Validation Execution Process** topic in the **Using the Experian Email Validation Integration** documentation.

Prerequisites

This functionality works 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 (manage.experianaperture.io), where customers can monitor their credit balance, set up an email notification, and refer to 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 Experian add-on component must be installed, and the Experian License must be obtained from Stibo Systems. For more information on licensing terms, contact your Stibo Systems account manager.

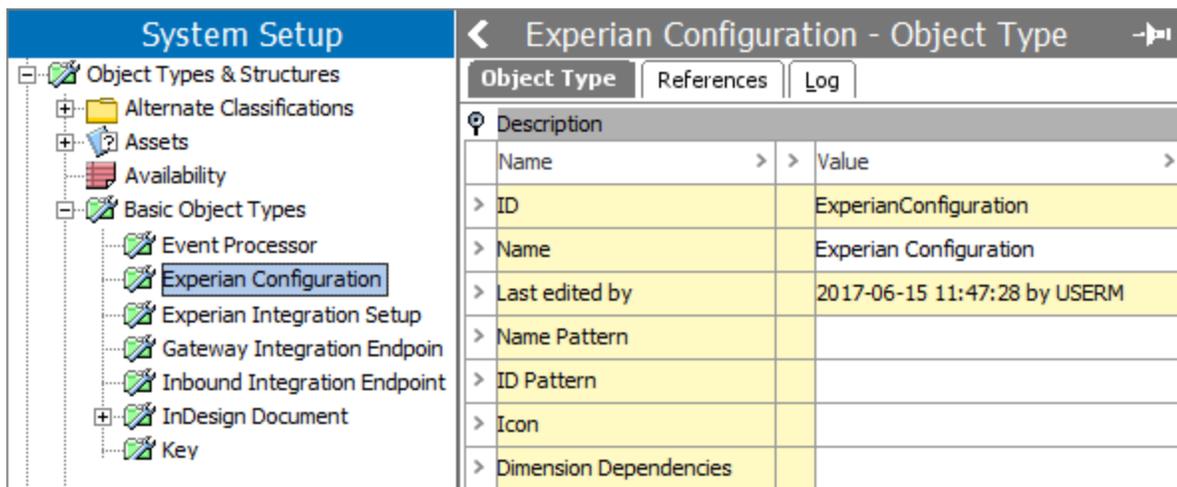
Configuring Experian Email Validation Integration

Each of the items required to implement the Experian Email Validation Integration are defined below.

Experian Email Validation Configuration Object

The Experian Email Validation Configuration object holds the configurations for the different Experian Email Validation Integration parameters. The Experian Email Validation Configuration object can be automatically created using the Easy Setup wizard.

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



The screenshot shows the 'System Setup' application with the 'Object Types & Structures' tree on the left. The 'Basic Object Types' folder is expanded, and 'Experian Configuration' is selected. The main pane displays the configuration for this object type, including a table of parameters.

Description	
Name	Value
ID	ExperianConfiguration
Name	Experian Configuration
Last edited by	2017-06-15 11:47:28 by USERM
Name Pattern	
ID Pattern	
Icon	
Dimension Dependencies	

Experian Configuration objects are represented within the workbench using the  icon, and include the following parameters within the Configuration flipper:

- **Batch Size:** The Experian Service API is limited to a maximum batch size of 100,000, so this should not be exceeded.
- **Processing Pause Period, Minutes:** The interval at which the Email Validation BGP is running and starting new Experian jobs, and polling for completeness of running jobs. Typically, the interval can be set to 30 minutes.
- **Experian Secure Token:** The token given by your Experian Service API license.

The Experian Configuration object also holds information about all running Experian jobs. For more information, refer to the **Experian Email Validation Execution Process** topic in the **Using the Experian Email Validation Integration** documentation.

Description	
Name	Value
ID	ExperianEmailValidationConfiguration
Name	Experian Email Validation Configuration
Object Type	Experian Configuration
Revision	0.2 Last edited by USER on Thu Feb 09 10:20:46 CET 2023
Path	Experian Configurations/Experian Email Validation Configuration
Configuration	
Batch Size	100000
Processing Pause Period, Minutes	30
Experian Secure Token	*****

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

The Experian Email Validation plugin running in the event processor and the Web UI Email Validation Configuration must refer to an Experian Email Validation Configuration object.

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 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.

System Setup

- Conditions
 - DQTest
 - Email Revalidation Condition**
 - False for Discontinued Products
 - Game Reference Condition
 - GDSN_ValidateTrue
 - GDSNRegisterCondition
 - GR Merge Keep First
 - Ignore Buy Side Objects
 - IsBackplate
 - Is Specific Hierarchy Item

Email Revalidation Condition rev.0.1 - Business Rule

Business Rule	Usage	Statistics	Log	Status
Name	>	>	Value	>
ID			EmailRevalidationCondition	
Name			Email Revalidation Condition	
Revision			0.1 Last edited by USERM on Wed Jun 14 12:56:14 EDT 2017	
Description				
Type			Condition	
Valid Object Types			All object types valid	
On Approve			Not Validated	
Scope			Global	
Run as privileged			<input type="checkbox"/>	

Operations Dependencies Applies if

New and changed emails are validated.

Edit Business Rule

Business Rule Editor - Email Revalidation Condition

ID: EmailRevalidationCondition
 Name: Email Revalidation Condition
 Description:
 Type: Condition
 Scope: Global
 On Approve: Not Validated
 Valid Object Types: All object types valid
 Run as privileged:

Operations Dependencies Applies if

New and changed emails are validated.

Add new Business Condition

Save Cancel

Edit Operation

Validate Emails Business Condition

New or changed emails will always be validated.

Renew validations older than: 60 Days
 (If left empty, this option is ignored)

Save Cancel

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 was used, and during the wizard step 'Create New Email Revalidation Condition,' the Setup Group 'Conditions' was selected.

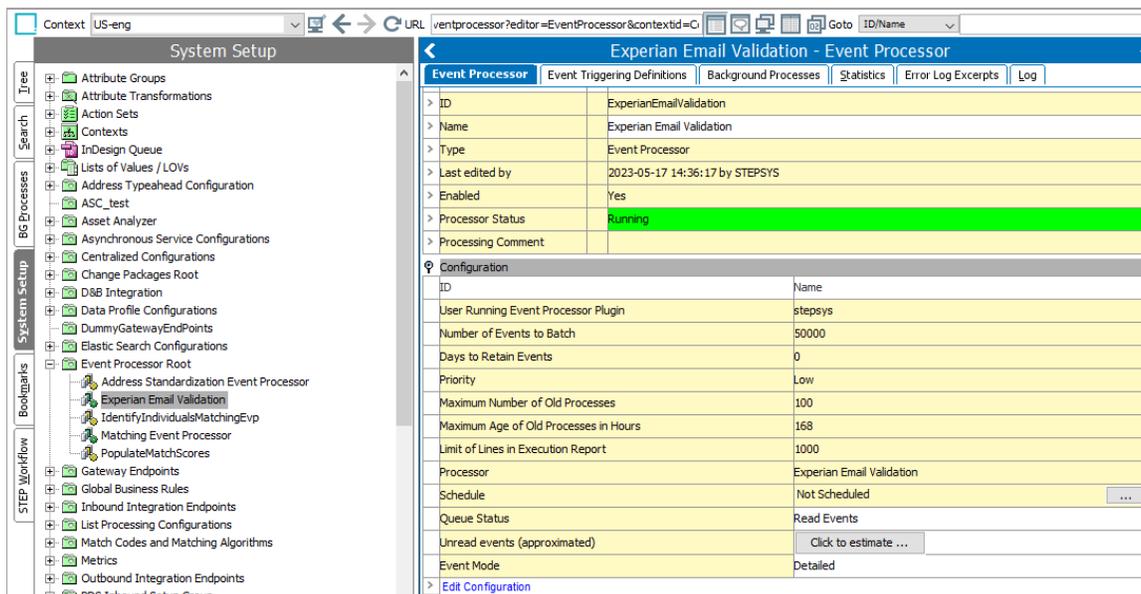
This makes it possible to define a condition like: 'Validate all billing emails that have changed since the last validation, or have not been validated in the last 2 months.' More than one email revalidation condition can be created, and using JavaScript it is possible to extend the Email Revalidation 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 is used, this occurs automatically.

For information on how the Email Revalidation Business Condition can be configured within the Experian Email Validation processing plugin, refer to the following **Experian Email Validation Processing Plugin** section.

Experian Email Validation Processing Plugin

The screenshot below 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 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 that it must be enabled before this event processor configuration can be used.



For information about the recommended settings in the event processor, refer to the following topics:

- **Using the Experian Email Validation Integration with the Event Processor.**
- **Experian Email Validation Processing Plugin Parameters and Triggers.**

Email Component Model

The Email Component Model defines the structure and data quality of the email addresses in STEP. It contains the configuration of the email data containers, the attributes that are necessary for proper functionality of the email validation, and the email data quality overview. After running the Easy Setup wizard, (covered in the **Configuring Experian Email Validation Integration Using the Easy Setup Wizard** topic), the Email

Component Model maps the component model names (Experian data fields) to STEP attributes. A completed component model using the Easy Setup wizard and automatic configuration options appears like the following image.

System Setup		Email Component Model - Component Model Configuration		
Component Model Configuration		Name	Value	Description
>	Email data containers		Email	Data Container types that carry email information.
>	Email Field		Email Field	The attribute used to store the actual email.
>	Email Status		Email Status	The latest email quality code returned by the validation Service.
>	Experian Email Validation Integration Status		Experian Email Validation Integration Status	Indicates if the last Email validation was completed or resulted in an error(e.g. a timeout).
>	Last Validated Email		Last Validated Email	The latest email send for validation.
>	Validation Timestamp		Validation Timestamp	Last time the email was received from the validation service.
>	Edit			

When during the easy setup not all values can be mapped, you can view what requires manual mapping within the component model. Refer to the Description column for context on how these fields are to be used if any of the object types, attributes, or data containers being mapped are unclear.

For more information, refer to the **Configuring Experian Email Validation Integration Using the Easy Setup Wizard** topic.

Configuring Experian Email Validation Integration Using the Easy Setup Wizard

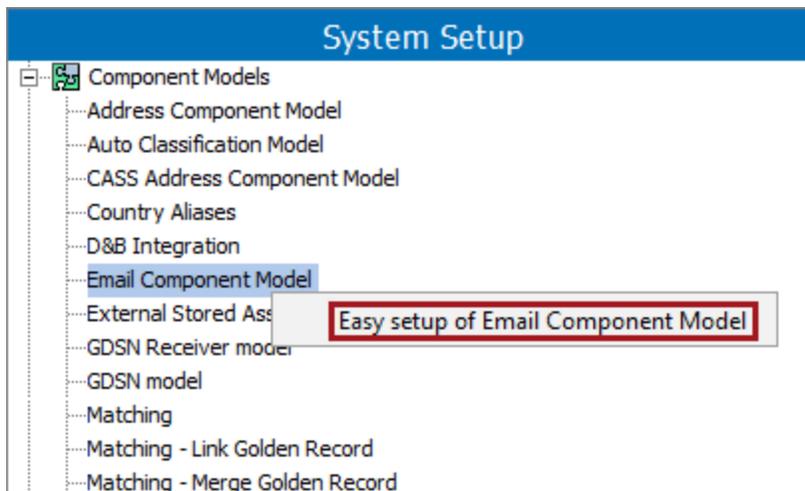
The Experian Email Validation Integration includes an Easy Setup wizard that creates the setup files, needed attributes, and data containers. The Easy Setup wizard is designed to make it so that users have very little configuration to do to get up and running as soon as possible.

Anyone configuring or using the Easy Setup wizard needs to have the proper privileges and understand how all the pieces work together.

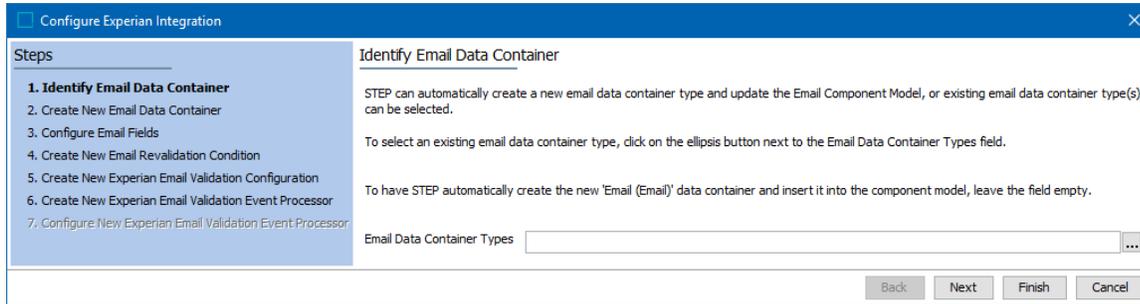
Important: The Easy Setup wizard can be finished without supplying the Experian Secure Token, however, before the Experian Email Validation Integration can be used, the Token needs to be added by rerunning the Easy Setup wizard or editing the Experian Email Validation Configuration object.

To perform the automatic configuration:

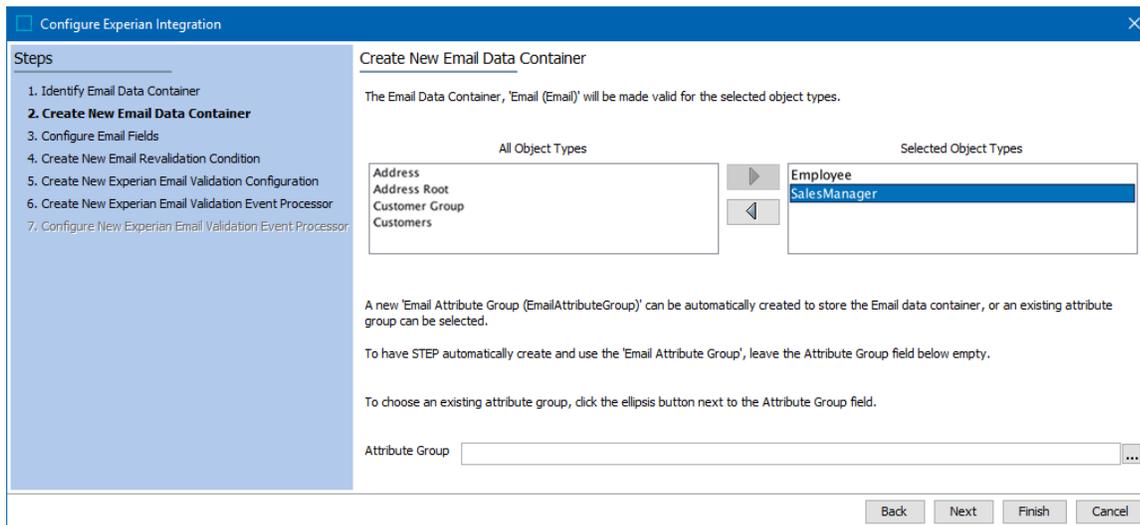
1. Optionally, create a setup group specifically to store the Experian Configuration. Otherwise, when prompted, choose an existing setup group.
2. Go to System Setup > Component Models > select **Email Component Model**.
3. Right-click **Email Component Model**, and select **Easy setup of Email Component Model**.



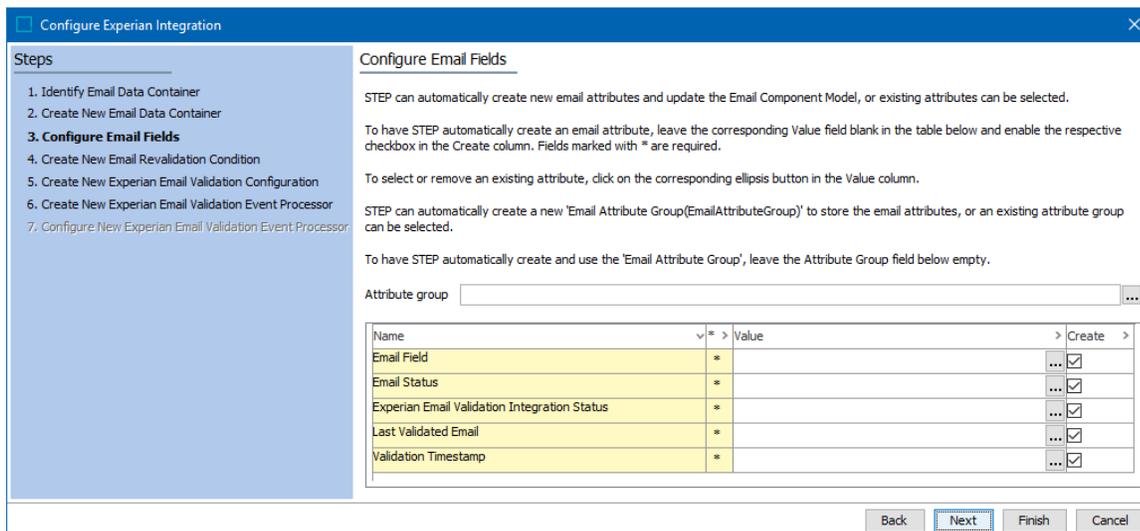
4. The Configure Experian Integration dialog displays the wizard step 'Identify Email Data Container' with detailed configuration instructions. Read the dialog text carefully to determine the necessary action.



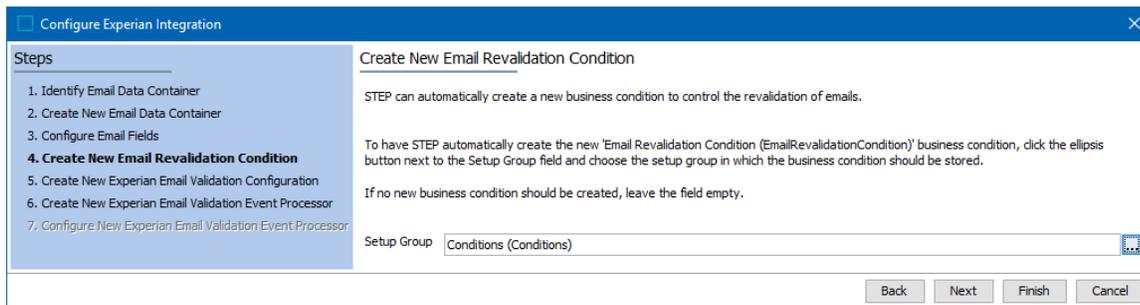
5. Click the **Next** button, and the wizard step 'Create New Email Data Container' displays with detailed configuration instructions. At a minimum, choose the object type(s) to enable the **Next** button.



6. Click the **Next** button, and the wizard step 'Configure Email Fields' displays with detailed configuration instructions.

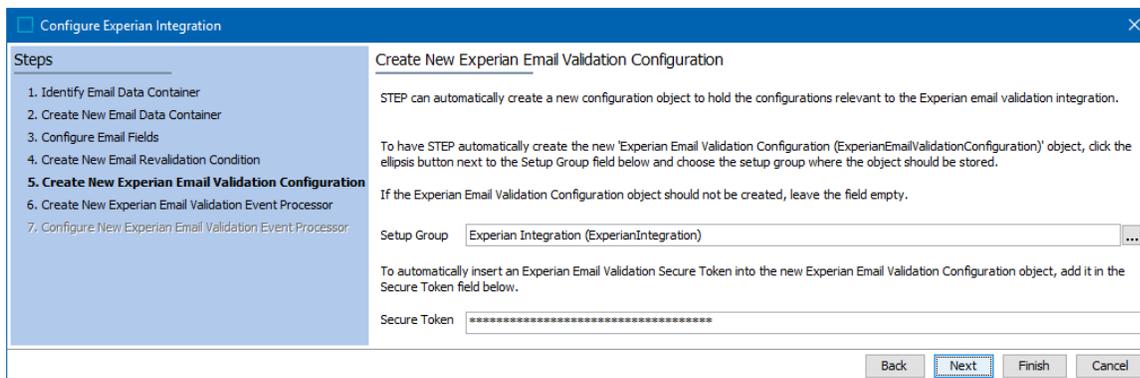


- Click the **Next** button, and the wizard step 'Create New Email Revalidation Condition' displays with detailed configuration instructions.



The screenshot shows a wizard window titled 'Configure Experian Integration'. The left sidebar lists seven steps, with step 4, 'Create New Email Revalidation Condition', highlighted in blue. The main content area is titled 'Create New Email Revalidation Condition' and contains the following text: 'STEP can automatically create a new business condition to control the revalidation of emails. To have STEP automatically create the new 'Email Revalidation Condition (EmailRevalidationCondition)' business condition, click the ellipsis button next to the Setup Group field and choose the setup group in which the business condition should be stored. If no new business condition should be created, leave the field empty.' Below this text is a 'Setup Group' field with the value 'Conditions (Conditions)' and an ellipsis button. At the bottom right are buttons for 'Back', 'Next', 'Finish', and 'Cancel'.

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

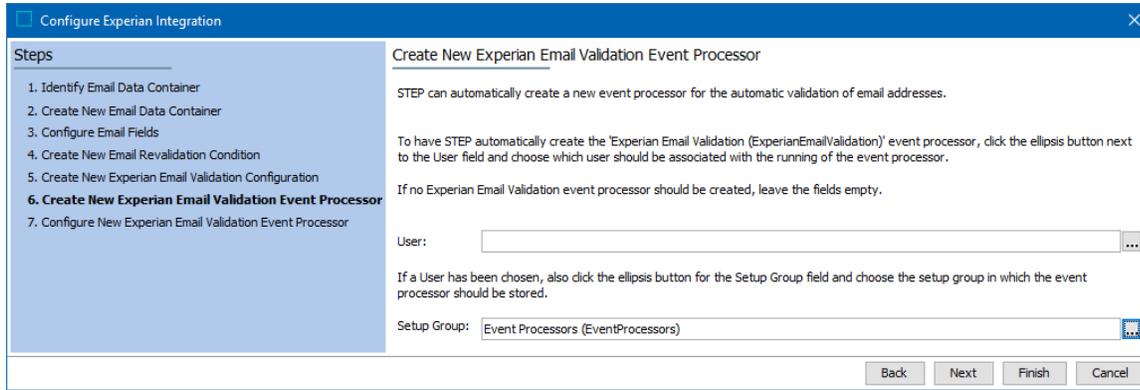


The screenshot shows the same wizard window, now at step 5, 'Create New Experian Email Validation Configuration'. The sidebar highlights step 5. The main content area contains: 'STEP can automatically create a new configuration object to hold the configurations relevant to the Experian email validation integration. To have STEP automatically create the new 'Experian Email Validation Configuration (ExperianEmailValidationConfiguration)' object, click the ellipsis button next to the Setup Group field below and choose the setup group where the object should be stored. If the Experian Email Validation Configuration object should not be created, leave the field empty.' Below this is a 'Setup Group' field with the value 'Experian Integration (ExperianIntegration)' and an ellipsis button. Further down is a 'Secure Token' field with masked characters (asterisks). At the bottom right, the 'Next' button is highlighted with a blue border.

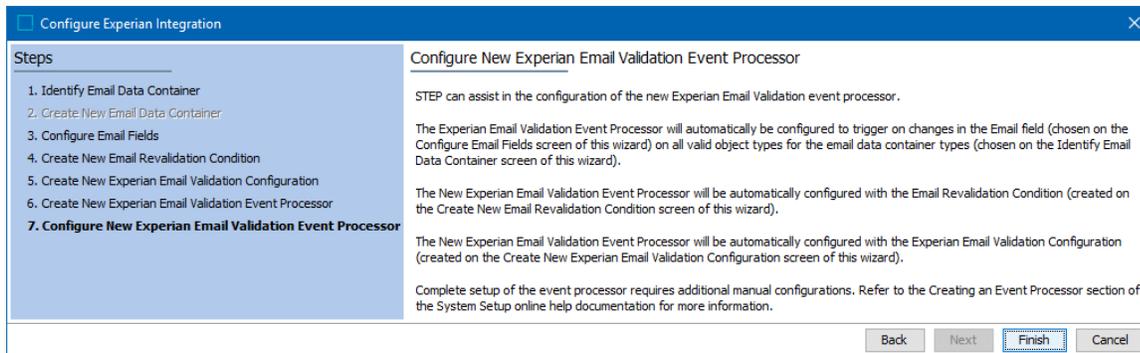
- **Setup Group:** For the example above, prior to starting the wizard a Setup Group was created (as advised in the beginning of this section). If necessary, cancel the wizard and create your own specific setup group for Experian, or click the ellipsis button (...) and choose an existing setup group.
- **Secure Token:** For security purposes, characters are masked as they are entered into the field. 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.

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

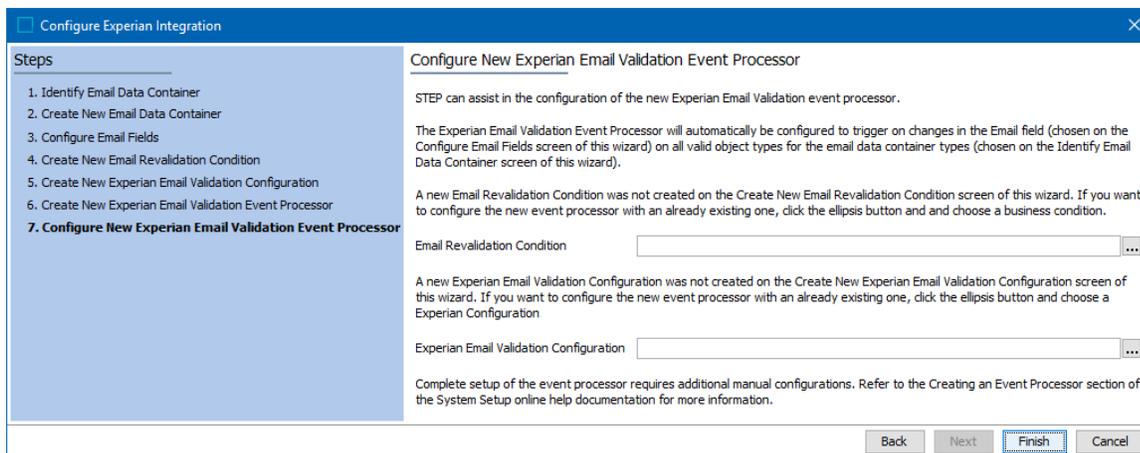
Important: The privileges of the selected user determine which actions the event processor can perform and what data can be processed. Common setup is to create a special system user for this purpose so that the effects of the event processor are easily identified. For more information, refer to the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



10. Click the **Next** button, and the wizard step 'Configure New Experian Email Validation Event Processor' displays 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' displays with the following.



11. Click the **Finish** button, the wizard closes, and the Email Component Model is populated with the selections made during the wizard.

Important: If the wizard was used to create a new event processor, additional manual steps necessary to set the Queue status and enable the event processor need to be completed. For more information, refer to the **Event-Based OIEP Status and Queue Status** topic and the **Running an Event Processor** topic, both in the **System Setup** documentation.

Configuring Experian Email Validation Integration for the Web UI

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. Refer to the **Web UI Getting Started** topic in the **Web User Interfaces** documentation for more information.

Validate Emails by Selecting from a List of Revalidation Conditions

It is possible to configure more than one email revalidation condition, allowing users to narrow down the Experian email validation request by selecting from common business conditions. For more information, refer to the **Configuring a Validate Emails Action Button for a Node List** section below.

For information on using the Experian Email Validation in the Web UI, refer to **Using the Experian Email Validation Integration from the Web UI** topic.

Handling Failed Email Validation

When a validation job fails, a collection of objects is created, and a link to the collection is displayed within the execution and summary report (within the 'Recent background processes' side panel). Go to the collection to find the list of customer accounts that have failed to be validated.

Configuring a Validate Emails Action Button for a Node List

The Validate Email action button can be added to a collection or 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, and to have created the mandatory Experian Email Validation Configuration using the Experian Configuration object type.

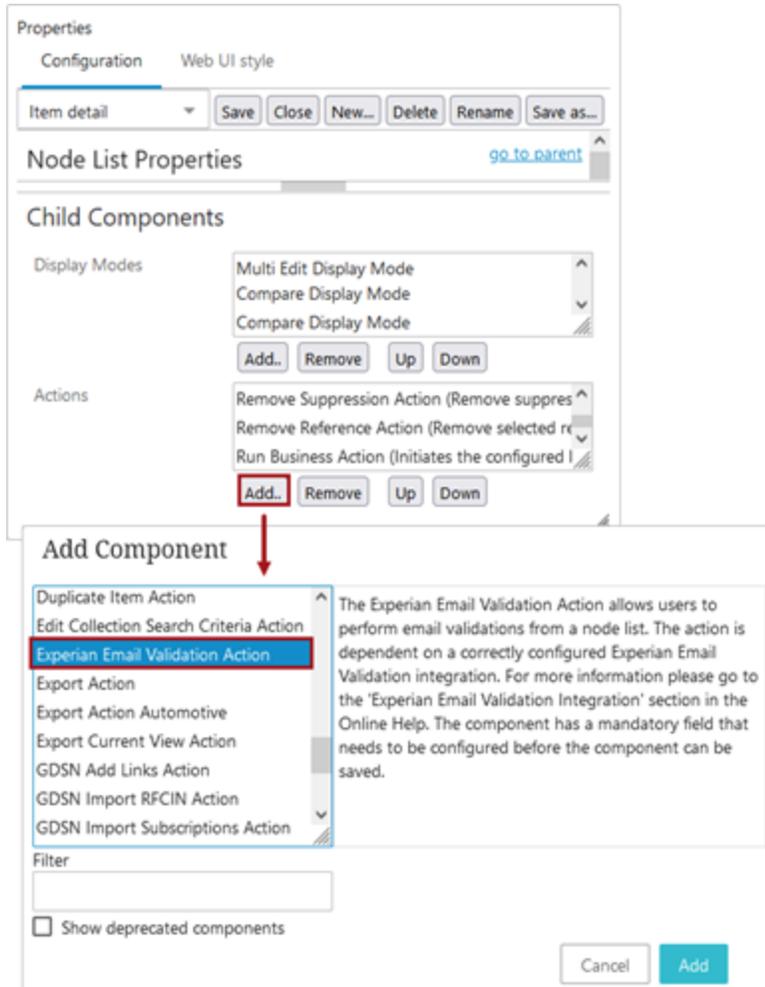
For more information about the Experian Email Validation Integration, refer to the **Experian Email Validation Integration** topic.

For more information about action buttons on Node List, refer to the **Action Button Configuration on a Node List** section of the **Web User Interface** 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.

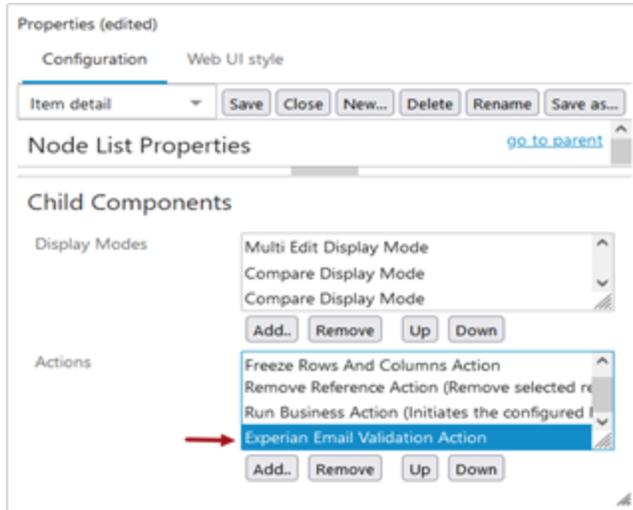
1. Go to your Node List Properties > Child Components > Actions parameter.
2. Click the **Add** button, and the Add Component dialog displays.
3. Select **Experian Email Validation Action**, click the **Add** button to close the dialog, and continue to the configuration steps.



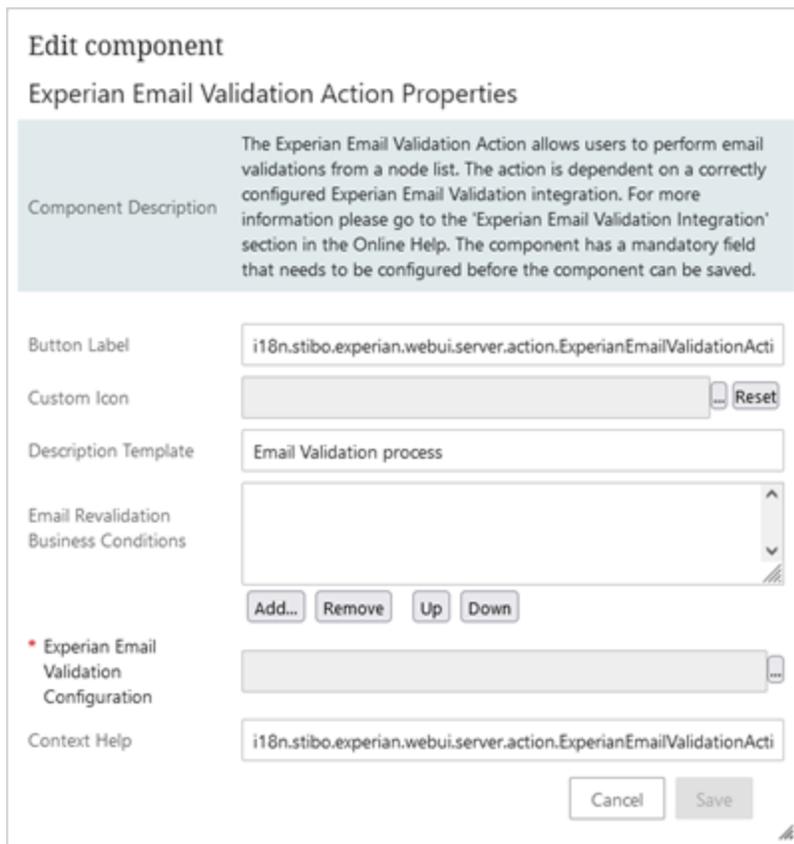
Configure a Validate Email Action Button for a Node List

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

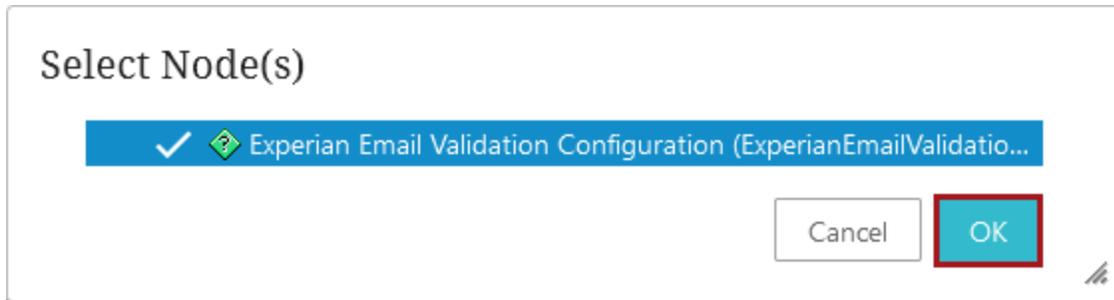
1. From the Actions parameter list, double click the newly added **Experian Email Validation Action** as shown below.



The Experian Email Validation Action Properties dialog displays as shown below.

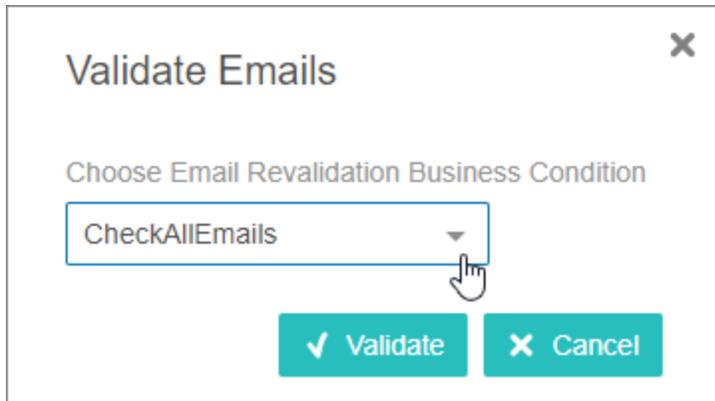


2. Populate the mandatory Experian Email Validation Configuration parameter with an Experian Configuration object. Click the ellipsis button (...) next to the Experian Email Validation Configuration parameter, and the Select Node(s) dialog will display Experian Configuration object types only.



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 closes, and the selected Experian Configuration object displays within the Experian Email Validation Configuration parameter.
4. Review and configure any of the optional parameters within 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 overwrites 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, or Validate emails with specific domains).
 - When business conditions are added to this parameter, they display for selection when a user clicks the Validate Emails action button in Web UI.



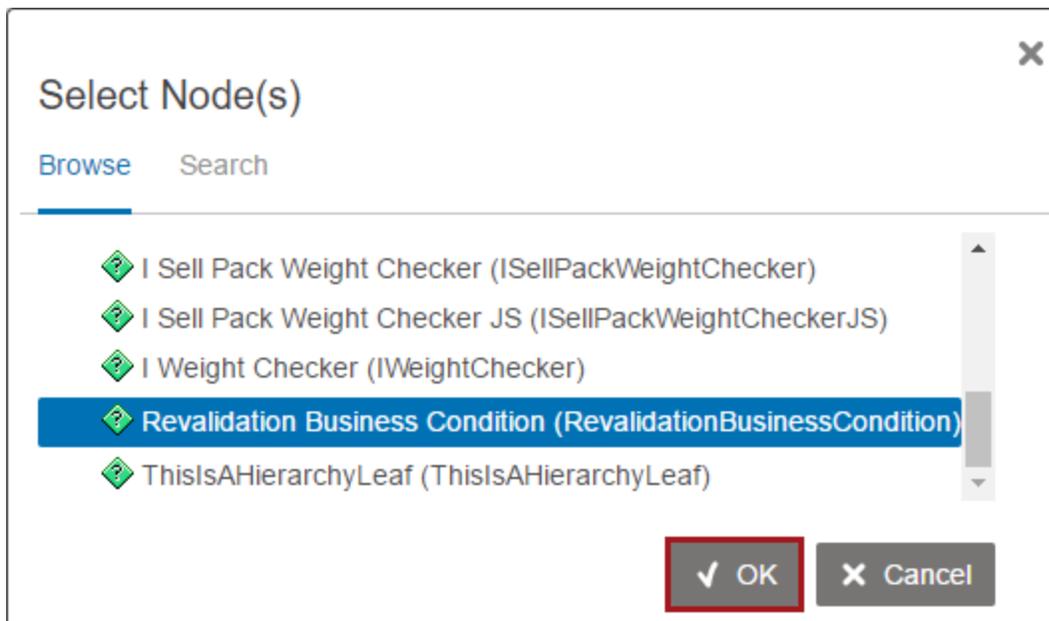
- When the parameter is empty, and the Web UI user clicks the Validate Emails action button, the user is not prompted to select a business condition, and all selected emails will be sent for email validation.

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

Add Email Revalidation Business Conditions

Below are steps to add a previously configured business condition 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 you used the Email Component model Easy Setup wizard 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.

Properties

[Configuration](#) Web UI style

CollectionContent Save Close New... Delete Rename Save as...

Experian Email Validation Action Properties [go to parent](#)

Button Label Validate Emails

Custom Icon ... Reset

Description Template Email Validation process

Email Revalidation Business Conditions

CheckAllEmails
VerifyNewEmails
RevalidateAfter30Days

Add... Remove Up Down

Experian Email Validation Configuration* ... Clear

Context Help Validate emails by Experian Email Validation

For information on and images of the Email Validation business condition in the workbench, refer to **Using the Experian Email Validation Integration from the Web UI** topic.

Using the Experian Email Validation Integration

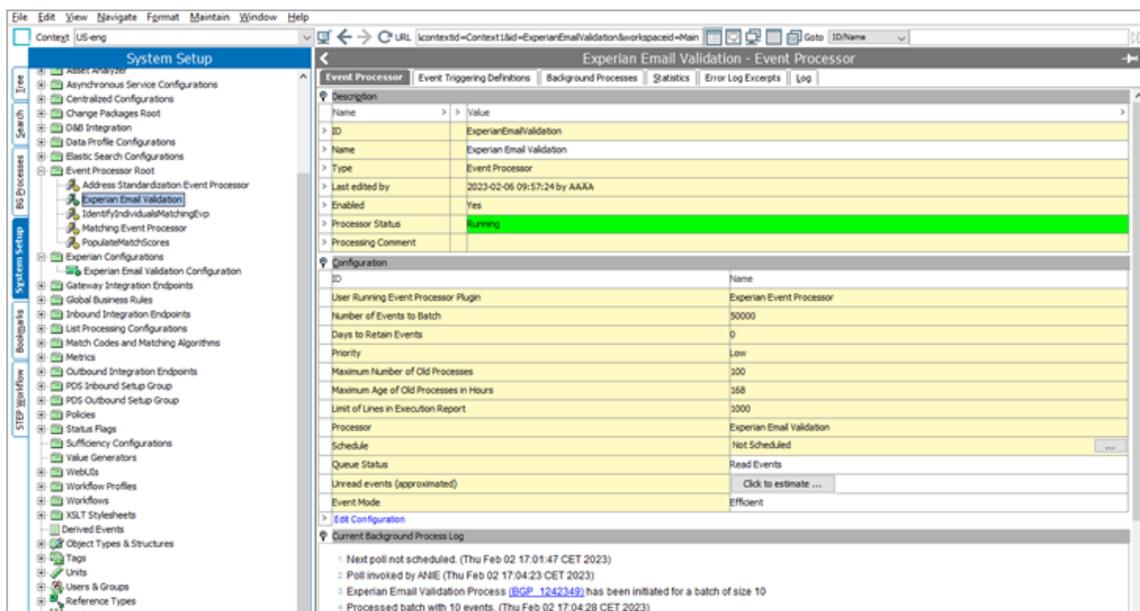
The Experian Email Validation Integration can be used in two ways:

- Configured to run using an Event Processor that can be scheduled to run at a certain interval. Based on events on new or changed email addresses, it will ensure that all email addresses are continuously validated.
- Configured in the Web UI to be able to manually start a validation of email addresses on a collection of entities.

In both scenarios, the Experian Configuration setup entity is needed, where the Background Processes tab will show the Email Validation BGPs that are executing the Experian validation jobs.

Using the Experian Email Validation Integration with the Event Processor

The Experian Email Validation processing plugin can be used in an event processor with an event triggering definition on changes in an attribute holding email data.



The Experian plugin takes batches of events and for each batch starts an 'Email Validation BGP'. For each successfully started process, the events are consumed up front and the BGP alone is responsible for executing the Experian email validation.

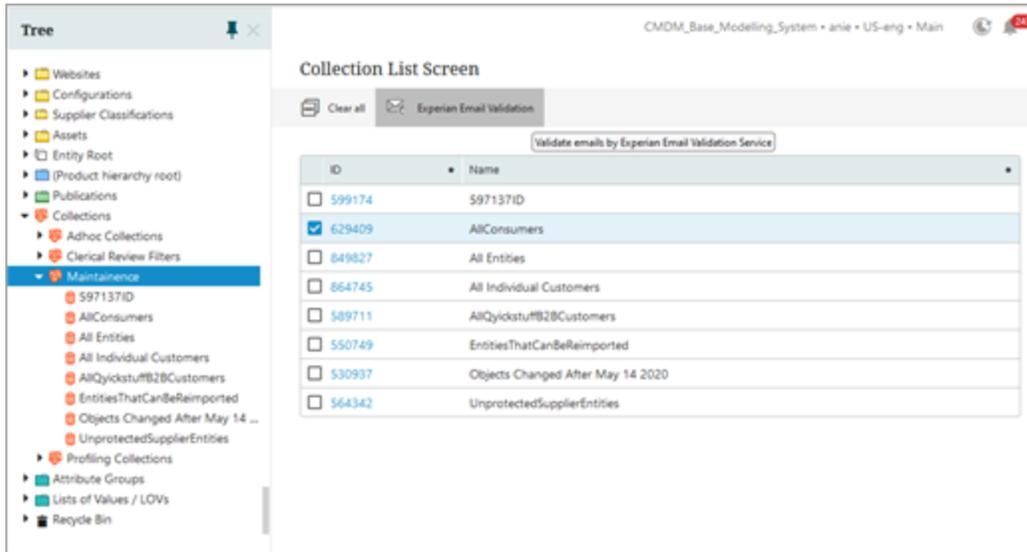
For further information about the execution process, refer to the topic **Experian Email Validation Execution Process**.

Limits and Recommendations

- There is a limit of 100,000 emails to be validated in the Experian Service API, and since entities typically hold multiple email addresses, the 'Number of Events to Batch' is recommended not to exceed 50,000.
- There is a limit in the Experian service API of max 5 running parallel 'email validation jobs', so the event processor is limited to start a maximum of 5 concurrent BGPs.
- Event Mode is recommended to be set to 'Efficient'. This removes duplicated events in other batches to make sure not to validate the same emails several times. This results in batch sizes that are lower than the configured 'Number of Events to Batch' in those other batches.

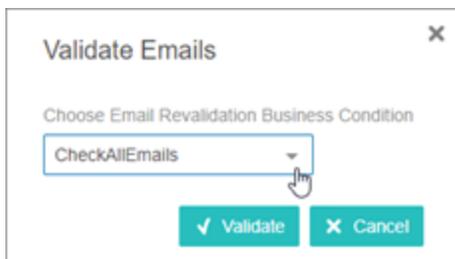
Using the Experian Email Validation Integration from the Web UI

The Experian Email Validation action button is available within Web UI and allows users to perform email validations on collections of entities or directly on one or more entities.



If more than one email revalidation condition is configured in the Web UI Designer, and a Web UI user clicks the **Validate Emails** action button, the user is prompted to choose from a dropdown list of available email revalidation business conditions. This allows the user to narrow down the Experian email validation request by selecting from common business conditions, such as:

- Validate new emails only
- Validate new emails and emails that have not been validated in the last 30 days
- Validate emails with specific domains



Once the desired email revalidation condition is selected, the background process notification displays.

For information on configuring multiple email revalidation business conditions, refer to the **Configuring a Validate Emails Action Button for a Node List** topic within the **Configuring Experian Email Validation Integration for the Web UI** topic.

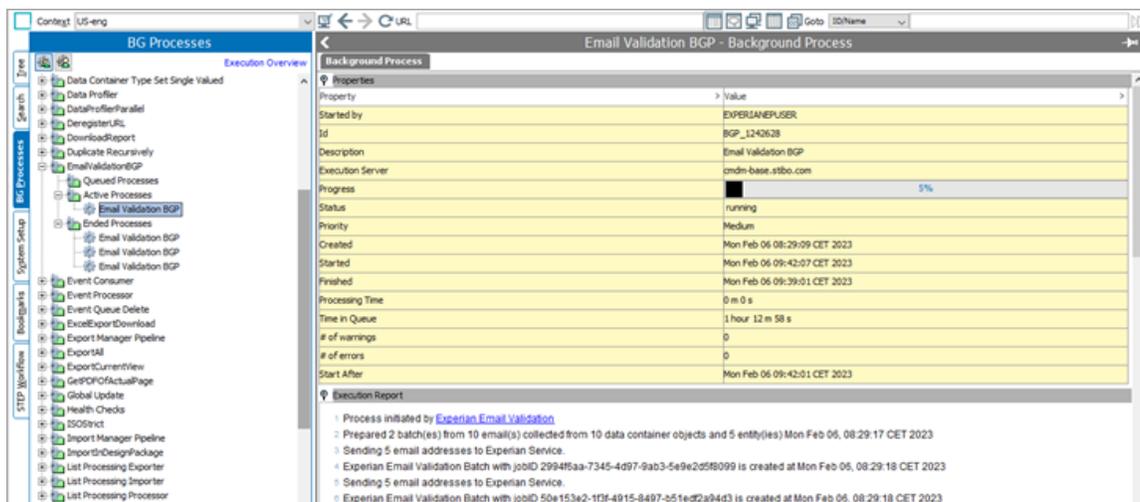
Once the validation action is clicked, the 'Email Validation BGP' starts as an asynchronous process. Refer to **Experian Email Validation Execution Process** topic for further information.

A background process notification displays in the 'Recent background processes' side panel. Since the Experian service does not return immediate responses, it is recommended that the background process be monitored in the 'Recent background processes' side panel. Refer to the **Recent Background Processes Side Panel** topic in the **Web User Interfaces** documentation for more information.

Experian Email Validation Execution Process

The Experian Email Validation BGP processes a batch of entities coming from the Event Processor using the Experian Email Validation plugin or from the Web UI where the Experian Email Validation button was clicked. The emails are extracted from the entities / data containers using the configuration in the Email Component Model and are split into batches based on the 'Batch Size' set on the Experian Configuration object. Since a single entity typically consists of multiple email addresses, the final batch of email addresses is larger than the batch of entities. For each batch of email addresses, an Experian job is started.

The Experian Service API can run up to five (5) jobs simultaneously per license key. To keep track of the started jobs, the Experian Configuration setup entity keeps a job registry of Experian job identifiers. This makes the solution resilient to system restarts (jobs are continued after a restart) and it ensures that no more than five Experian jobs are ever started at the same time across all running Email Validation BGPs. For this reason, it is important to use only one Experian Configuration setup entity per license key and to ensure not to delete and purge the setup entity while there are Experian jobs running. The job registry is not visible in the workbench.



The process polls for answers about completed jobs from the Experian Service API.

- If not all jobs are finished, the BGP goes back to 'waiting' state to free up capacity for other BGPs in the system to run. The setting 'Processing Pause Period, Minutes' on the Experian Configuration setup entity determines how long to wait before the BGP runs again and check for finished jobs.
- When a job is finished, it is removed from the job registry on the Experian Configuration setup entity, and if one or more additional batches of email addresses are pending, additional jobs are started.

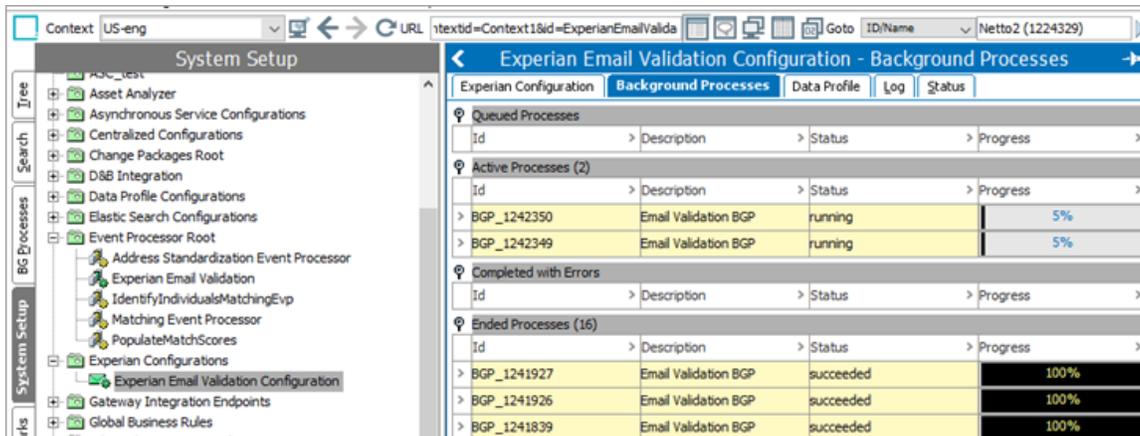
When all batches of email addresses have been processed and all jobs are completed within the BGP, the email data quality fields configured within the Email Component Model are updated:

- One of the following result values is written into the 'Email Status' email data quality field:
 - **(blank)**: Email has never been sent for validation.
 - **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**: Experian service is 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 'Last Validated Email' field is updated with the email address last sent for validation.
- The 'Validation Timestamp' field is updated with the date and time the email address was last received from the Experian service.
- The 'Experian Email Validation Integration Status' field displays one of the following values:
 - **(blank)**: Indicates the email address has never been sent for validation.
 - **Success**: Indicates the email address has been successfully validated.
 - **Processing**: Indicates the email address is in the process of being validated.
 - **Failed**: Indicates the email address validation has failed. This allows a search to be performed for accounts where an error has occurred during validation, and a collection is automatically created.

Note: If too many revisions occur (causing performance issues) for the 'Experian Email Validation Integration Status' field, then it is recommended that the attribute be set to externally maintained. For more information, refer to the **Externally Maintained Attributes** section of the **Attributes** documentation.

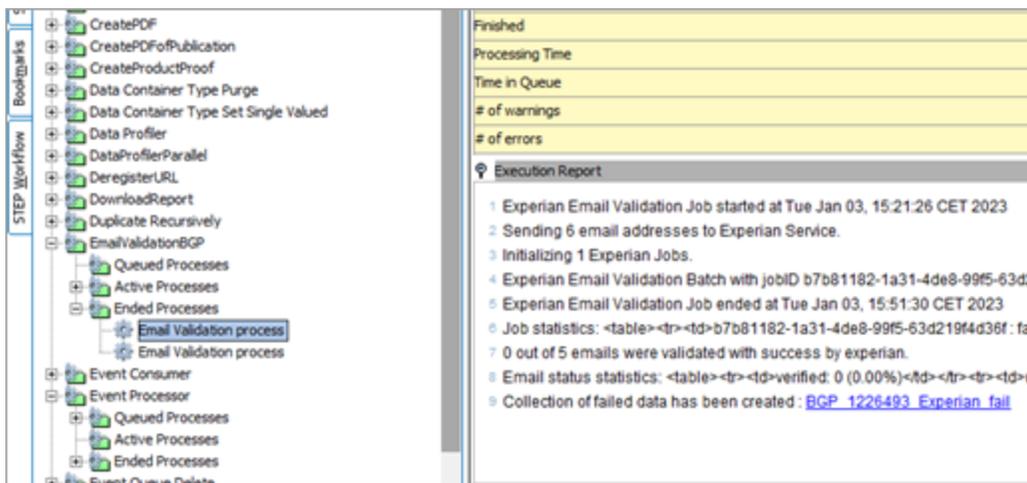
Email Validation BGP Overview

The Experian Email Validation Configuration setup entity is used in email validation processes initiated from the event processor and processes initiated from the Web UI. The Background Processes tab shows all the started processes.



Handling Failed Email Validation

When an Experian job fails, a collection of objects is created, and a link to the collection is displayed within the execution report. Go to the collection and find the list of entities that have failed to be validated.



The failed jobs can be manually restarted by relaunching the Email Validation BGP.

Experian Throughput Limitations

The Experian service is asynchronous and can validate up to 5 concurrent jobs with each a maximum of 100,000 email addresses.

Common response times for each batch job run by Experian are:

- 12 hours for up to 10,000 email addresses
- 20 hours for up to 100,000 email addresses

The time it takes for Experian to process a job varies depending on the makeup of the set of emails in the job. If the job is made up of mostly top common domains (AOL, Yahoo, Gmail, etc.), Experian can process faster than a job that is made up of mostly lesser-known domains (business domains, etc.).

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, as covered in the **Matching, Linking, and Merging** documentation.

Customers can receive an API key with a free trial account. Contact your Stibo Systems account manager or partner manager to purchase a Loqate license through Stibo Systems.

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. By using Loqate, an address is first standardized and then verified. These are two separate processes that must be executed together to ensure an address is valid. To standardize an address, Loqate updates 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 shows 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):

Organization Customer	Data Containers	References	Referenced By	Matching	Status	State Log	Tasks
382431	City	abc	Atlanta				
	Country	abc	USA				
	Country ISO Code	abc					
	Input Address 1	abc	3200 Windy Hill Rd SE Suite 1200W				
	Input Address 2	abc					
	Input Address Line	abc					
	Last Edit Date - Main Address	abc					
	Latitude	abc					
	Longitude	abc					
	Quality	abc					
	Quality Index	abc	A				
	Quality Verification Code	abc	V55-155-P7-100				
	Standardized Address	abc	3200 Windy Hill Rd SE Ste 1200W Atlanta GA 30339-8442				
	Standardized City	abc	Atlanta				
	Standardized Country	abc	United States				
	Standardized Country ISO Code	abc	US				
	Standardized State	abc	GA				
	Standardized Street	abc	3200 Windy Hill Rd SE Ste 1200W				
	Standardized Street Name	abc	Windy Hill Rd SE				
	Standardized Street Number	abc	3200				
	Standardized Street Type	abc	Rd				
	Standardized Zip	abc	30339-8442				
	State	abc	Georgia				
	Street	abc					
	Validation Hash	abc	1620776530				
Validation Integration Status	abc						
Validation Response	abc	<pre> Loqate[1620776530]<?xml version="1.0" encoding="UTF-8" standalone="yes"?><!DOCTYPE status><OK status="OK" result="AVC" V55-155-P7-100/AVC/Address1>3200 Windy Hill Rd SE Ste 1200W/Address2>Atlanta GA 30339-8442/Address3><AdministrativeArea>GA</AdministrativeArea><Building></Building><BuildingType></BuildingType><CountryName> United States</CountryName><DeliveryAddress>3200 Windy Hill Rd SE Ste 1200W</DeliveryAddress><DeliveryAddress1>3200 Windy Hill Rd SE Ste 1200W</DeliveryAddress1><Directional></Directional><ISO3166-2>US</ISO3166-2><ISO3166-3>USA</ISO3166-3><ISO3166-N>840 </ISO3166-N><Locality>Atlanta</Locality><Organization></Organization><PostBox></PostBox><PostalCode>30339-8442</PostalCode> <PostalCodePrimary>30339</PostalCodePrimary><PostalCodeSecondary>8442</PostalCodeSecondary><Premise>3200</Premise><P remiseNumber>3200</PremiseNumber><SubAdministrativeArea>Cobb</SubAdministrativeArea><SubBuilding></SubBuilding><SubB uildingNumber>1200W</SubBuildingNumber><SubBuildingType></SubBuildingType><SubBuildingLeadingType></SubBuildingLeadingType><SubB uildingType></SubBuildingType><Thoroughfare>Windy Hill Rd SE</Thoroughfare><ThoroughfarePostDirection></ThoroughfarePostDirection><ThoroughfareTrailingType></ThoroughfareTrailingT ype><ThoroughfareType></ThoroughfareType><Latitude>33.914090</Latitude><Longitude>-84.461260</Longitude><GeoAccuracy> P-4</GeoAccuracy><AQI>A</AQI><SuperAdministrativeArea></SuperAdministrativeArea><DependentLocality></DependentLocality> <DoubleDependentLocality></DoubleDependentLocality><DependentThoroughfare></DependentThoroughfare><Department></Depart ment><LocalityExtra></LocalityExtra><Address>3200 Windy Hill Rd SE Ste 1200W&lt;br>Atlanta GA 30339-8442</Address><CASSDeliveryPointBarCode></CASSDeliveryPointBarCode><CASSDPVConfirmedIndicator></CASSDPVConfirmedI ndicator><CASSFIPSCountyCode></CASSFIPSCountyCode><CASSNoStatIndicator></CASSNoStatIndicator><CASSResidentialDelivery ></CASSResidentialDelivery><CASSVacantIndicator></CASSVacantIndicator><CASSValidationResponse>AutoZoneIndicator: CarrierRoute: CMRAIndicator: DefaultFlag: DPVFootnotes: eLTCCode: eLTONumber: EWSFlag: FalsePositiveIndicator: Footnotes: LACSLinkCode: LACSLinkIndicator: LACSStatus: PMBNumber: PMBType: PrimaryAddressLine: RecordType: ReturnCode: SecondaryAddressLine: SUTELinkFootnote: </CASSValidationResponse></result></results></!DOCTYPE status> </pre>					
Validation Time	abc	2021-06-09 13:28:33					
Zip	abc	30339					

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.

Regardless of your STEP deployment type (SaaS or on-prem), all the above integration options are applicable with Loqate.

Stibo Systems is a reseller of the Loqate solution, commercially licensing the Loqate software along with its reference data. For information on obtaining and deployment of the three variations of the Loqate address standardization service, contact your Stibo Systems account manager or partner manager. Additional support information and deployment instructions can be found on the Loqate website. Users must have a Loqate account already set up to access the additional support section on the Loqate website.

Loqate Cloud

The Loqate Cloud API service is an integration that enables STEP to communicate with an off-premise 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 a Loqate Cloud API license key.

The Loqate Cloud solution may be a preferred option for users with smaller data sets (less than 1 million records) looking to only pay for their predefined data needs. Also, users must be connected to a local Loqate engine in order to use the CASS program.

To begin integrating your STEP deployment with the Loqate Cloud API, contact your Stibo Systems account manager or partner manager.

Loqate Local

The Loqate Local API service is an integration that enables STEP users to standardize address information against reference data stored 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 embedded 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 API license key.

Other reasons users might prefer the Loqate Local solution include:

- Commercially more cost-effective for work with large data sets (over 1 million records)
- Better performance and faster response times
- Zero additional downtime
- Option to use CASS address standardization

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, refer to the **CASS Address Component Model** topic.

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

Address Validation Web Service

When STEP is deployed with Loqate Local, STEP 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 for 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, access the STEP API Documentation button on the Start Page and refer to the SOAP and REST API sections.

Web service endpoint

The Address Standardization Web Service supports both SOAP and REST API. To download the WSDL and obtain access to the Address Validation Web Service, users must have access to the Stibo SDK and API documentation.

Prior to using the Address Standardization Web Service, Loqate Local must already have been deployed in your target STEP environment.

The following is the endpoint for the REST API:

```
https://[URL for STEP environment]/restapi/loqate/address/verify
```

The following is the endpoint for the SOAP API:

```
https://[URL for STEP environment]/LoqateWS/loqate
```

For information on gaining access to the SDK and API documentation, refer to the Granting Access to the SDK and API Documentation topic in the System Setup documentation, or contact your Stibo Systems account manager for further details.

Address information provided by this web service is subject to the Loqate subscription license provisioning. For more information, contact your account manager.

Examples

The Address Validation Web Service standardizes address information outside of STEP. For information on deciphering Loqate's Address Verification Code, refer to the **Evaluating Loqate Address Verification Codes** topic in the **Solutions Enablement** documentation.

Request

For the REST request, refer to the code below:

```

1  {
2      "Input": {
3          "Address": [
4              {
5                  "Address": "",
6                  "Address1": "4200 amphitheatre",
7                  "Address2": "",
8                  "Address3": "",
9                  "Address4": "",
10                 "Locality": "mountain view",

```

```

11         "Country": "US",
12         "AdministrativeArea": "",
13         "Thoroughfare": "",
14         "DependentThoroughfare": "",
15         "ThoroughfareName": "",
16         "Premise": "",
17         "PostCode": "",
18         "SubAdministrativeArea": "",
19         "DependentLocality": "",
20         "Building": "",
21         "SubBuilding": "",
22         "Organization": "",
23         "PostBox": ""
24     }
25 ]
26 },
27 "Options": {
28     "AddressLineSeparator": "",
29     "OutputAddressFormat": "",
30     "EnhancedCountryTool": ""
31 }
32 }

```

For the SOAP request, refer to the code below:

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://stibo.com/step/ws/loqate/1.0">
2 <soapenv:Header/>
3 <soapenv:Body>
4 <ns:verify>
5 <VerifyAddressRequest>
6 <Authentication>
7 <userid>user</userid>
8 <password>secret</password>
9 </Authentication>
1 <Input>

```

```

0
1   <Address>
1
1   <Address/>
2
1   <Address1>Google</Address1>
3
1   <Address2>4200 amphitheatre</Address2>
4
1   <Address3/>
5
1   <Address4/>
6
1   <Locality>mountain view</Locality>
7
1   <Country>US</Country>
8
1   <CountryISO/>
9
2   <AdministrativeArea>CA</AdministrativeArea>
0
2   <Thoroughfare/>
1
2   <DependentThoroughfare/>
2
2   <ThoroughfareName/>
3
2   <Premise/>
4
2   <PostCode/>
5
2   <SubAdministrativeArea/>
6
2   <DependentLocality/>
7
2   <Building/>
8
2   <SubBuilding/>
9
3   <Organization/>

```

```

0
3   <PostBox/>
1
3   </Address>
2
3   </Input>
3
3   <Options>
4
3   <AddressLineSeparator>&#xA;</AddressLineSeparator>
5
3   <OutputAddressFormat/>
6
3   <EnhancedCountryTool/>
7
3   </Options>
8
3   <Process>
9
4   <CertifyProcess/>
0
4   <GeocodeProcess></GeocodeProcess>
1
4   </Process>
2
4   </VerifyAddressRequest>
3
4   </ns:verify>
4
4   </soapenv:Body>
5
4   </soapenv:Envelope>
6

```

Response

For the REST response, refer to the code below:

```

1   {"Output": {"Address": [{
2     "AddressContent": {

```

```

3      "Address": "4200 Amphitheatre Pkwy<BR>Mountain View CA 94043-1381",
4      "Address1": "4200 Amphitheatre Pkwy",
5      "Address2": "Mountain View CA 94043-1381",
6      "Address3": "",
7      "Address4": "",
8      "AdministrativeArea": "CA",
9      "AddressFormat": "",
10     "Building": "",
11     "CountryName": "United States",
12     "DeliveryAddress": "4200 Amphitheatre Pkwy",
13     "DeliveryAddress1": "4200 Amphitheatre Pkwy",
14     "DeliveryAddress2": "",
15     "Department": "",
16     "DependentLocality": "",
17     "DependentThoroughfare": "",
18     "DoubleDependentLocality": "",
19     "Iso31662": "US",
20     "Iso31663": "USA",
21     "Iso3166N": "840",
22     "Latitude": "",
23     "LocalityExtra": "",
24     "Longitude": "",
25     "Locality": "Mountain View",
26     "Organization": "",
27     "PostBox": "",
28     "PostalCode": "94043-1381",
29     "Premise": "4200",
30     "PremiseNumber": "4200",
31     "SubAdministrativeArea": "Santa Clara",
32     "SubBuilding": "",
33     "SuperAdministrativeArea": "",
34     "Thoroughfare": "Amphitheatre Pkwy",
35     "ThoroughfareName": "",
36     "ThoroughfareType": ""
37   },
38   "AddressQuality": {
39     "AddressVerificationCode": "V42-I44-P3-092",

```

```

40     "AddressQualityIndex": "C",
41     "GeoAccuracy": "",
42     "GeoDistance": ""
43   },
44   "AddressVerificationDetails": {
45     "ValidationTime": 1667241105193,
46     "ValidationStatus": "psOK",
47     "ValidationStatusDetails": "",
48     "ValidationHash": "OEsCDGuSCMwvfYvv9LCaRQ=="
49   },
50   "AddressCertificationDetails": null
51 }]]}

```

For the SOAP response, refer to the code below:

```

1  <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
2  <S:Body>
3  <ns2:verifyResponse xmlns:ns2="http://stibo.com/step/ws/loqate/1.0">
4  <VerifyAddressResponse>
5  <Output>
6  <Address>
7  <AddressContent>
8  <Address>Google
9  4200 Amphitheatre Pkwy
10 Mountain View CA 94043-1381</Address>
11 </Address1>Google</Address1>
12 <Address2>4200 Amphitheatre Pkwy</Address2>
13 <Address3>Mountain View CA 94043-1381</Address3>
14 <Address4/>
15 <AdministrativeArea>CA</AdministrativeArea>
16 <AddressFormat/>
17 <Building/>
18 <CountryName>United States</CountryName>
19 <DeliveryAddress>4200 Amphitheatre Pkwy</DeliveryAddress>
20 <DeliveryAddress1>4200 Amphitheatre Pkwy</DeliveryAddress1>
21 <Department/>

```

```

22 <DependentLocality/>
23 <DependentThoroughfare/>
24 <DoubleDependentLocality/>
25 <Iso31662>US</Iso31662>
26 <Iso31663>USA</Iso31663>
27 <Iso3166N>840</Iso3166N>
28 <Latitude/>
29 <Longitude/>
30 <LocalityExtra/>
31 <Locality>Mountain View</Locality>
32 <Organization>Google</Organization>
33 <PostBox/>
34 <PostalCode>94043-1381</PostalCode>
35 <Premise>4200</Premise>
36 <PremiseNumber>4200</PremiseNumber>
37 <SubAdministrativeArea>Santa Clara</SubAdministrativeArea>
38 <SubBuilding/>
39 <SuperAdministrativeArea/>
40 <Thoroughfare>Amphitheatre Pkwy</Thoroughfare>
41 <ThoroughfareName>Amphitheatre</ThoroughfareName>
42 <ThoroughfareType/>
43 </AddressContent>
44 <AddressQuality>
45 <AddressVerificationCode>V42-I45-P3-094</AddressVerificationCode>
46 <AddressQualityIndex>C</AddressQualityIndex>
47 <GeoAccuracy/>
48 <GeoDistance/>
49 </AddressQuality>
50 <AddressVerificationDetails>
51 <ValidationHash>mP16Z8j98vYiMxYzb1XC2A==</ValidationHash>
52 <ValidationTime>1664478336487</ValidationTime>
53 <ValidationStatus>psOK</ValidationStatus>
54 </AddressVerificationDetails>
55 </Address>
56 </Output>
57 </VerifyAddressResponse>
58 </ns2:verifyResponse>

```

```
59 | </S:Body>  
60 | </S:Envelope>
```

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.

Note: This solution can only be used in an on-premise deployment.

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
```

Address Component Model

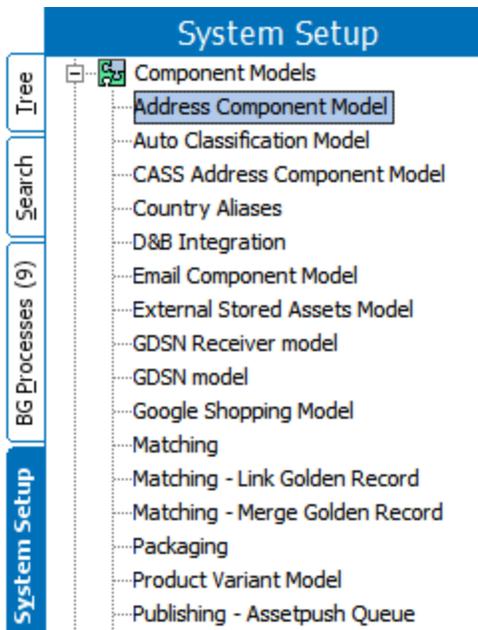
The address component model allows the integration of an address standardization solution with STEP. You must configure it to enable the use of Loqate and CASS integrations. It can also be used to integrate with Google Places for use in the Web UI Address Typeahead functionality, as defined in the **Address Typeahead** topic in the **Web UI Interfaces** documentation.

Note: Several components require that the address component model be configured before they work. This includes the Party Data Matching Component, the Standardize Address Business Action, and the Data Quality: Standardize Address Bulk Operation. This topic provides an overview of the Address Component Model, including full descriptions of the Loqate address attribute fields it contains. For instructions on how to configure the component model, refer to 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)

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

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 the 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.

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. Typically, users do not use all address fields.

Loqate fields are mapped to STEP attributes, either manually or through 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 that are 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	<i>Not mapped to Loqate</i>
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]

Address Component Model Field	Description	Local Loqate Field Mapping
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]
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]

Address Component Model Field	Description	Local Loqate Field Mapping
Latitude	The address latitude	<i>Not mapped to Loqate</i>
Longitude	The address longitude	<i>Not mapped to Loqate</i>
Standardized Address Attributes (OUTPUT Fields)		
Geocode Latitude	Output field containing the address latitude	Latitude [out]
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	Output field for small population center within a city, e.g., village.	DoubleDependentLocality [out]

Address Component Model Field	Description	Local Loqate Field Mapping
Locality		
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]
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]

Address Component Model Field	Description	Local Loqate Field Mapping
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.	<i>Not mapped from Loqate</i>
Quality Index	Output field for the address quality index, which is used to indicate the quality of an address.	<i>Address Quality Index (AQI) response from Loqate</i>
Quality Verification Code	Output field for the Address Verification Code. This is used to indicate the level of verification of an address.	<i>Address Verification Code (AVC) response from Loqate</i>
Validation Hash	<p>Hash value (numerical representation) for the address validation integration.</p> <p>Contains a hash value of all input fields. Since comparing two numbers is simpler than comparing all involved values of the address, the hash value is used to determine if the address has changed since the last validation.</p> <p>If the hash values of two addresses are equal, then the actual addresses are most likely also equal. For two identical addresses, the hash values are always be the same. If two addresses are different, the hash values for the two objects are likely to be distinct, but there is no guarantee.</p> <p>For example, on a system with more than 100,000 addresses, it is expected that several addresses that are very different have the same hash value. However, for any specific address, the likeliness that the next version of that address has the same hash value is 1 per 4 billion. This means that in a system with 100 million addresses, if a user updates every address, there is less than a 5 percent chance that even a single one of those address changes is not sent to Loqate because the hash value is unchanged.</p> <p>This field is calculated by STEP and updated whenever the standardized address attributes are updated by the Loqate integration.</p>	<p><i>Not mapped to Loqate</i></p> <p><i>Not mapped from Loqate</i></p>
Validation Integration Status	<p>Contains the latest status of the address validation integration.</p> <p>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.</p> <p>This field makes it possible to search for accounts and addresses where an error has occurred during the address validation.</p>	<i>Not mapped from Loqate</i>
Validation Response	<p>Output field for the cached validation response. This field contains the full response from Loqate.</p> <p>Validation base type is text; maximum length must be at least 1000.</p>	<i>Contains the full response from Loqate</i>

Address Component Model Field	Description	Local Loqate Field Mapping
	<div style="background-color: #ffffcc; padding: 5px;"> <p>> Validation Response abc</p> </div> <pre data-bbox="669 390 1284 1417"> Loqate/1379196886/<?xml version="1.0" encoding="UTF-8"?> <!qt> <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</DeliveryAddress> <DeliveryAddress1>3550 George Busbee Pkwy NW</DeliveryAddress1> <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</ThoroughfarePostDirection> <ThoroughfareTrailingType>Pkwy</ThoroughfareTrailingType> <ThoroughfareType>Pkwy</ThoroughfareType> </result> </results> </!qt> </pre>	
Validation Time	<p>Date and time stamp of the most recent successful address validation.</p> <p>Validation base type is 'ISO Date and Time.' Field is updated whenever the standardized address attributes are updated by the Loqate integration.</p>	Not mapped from Loqate

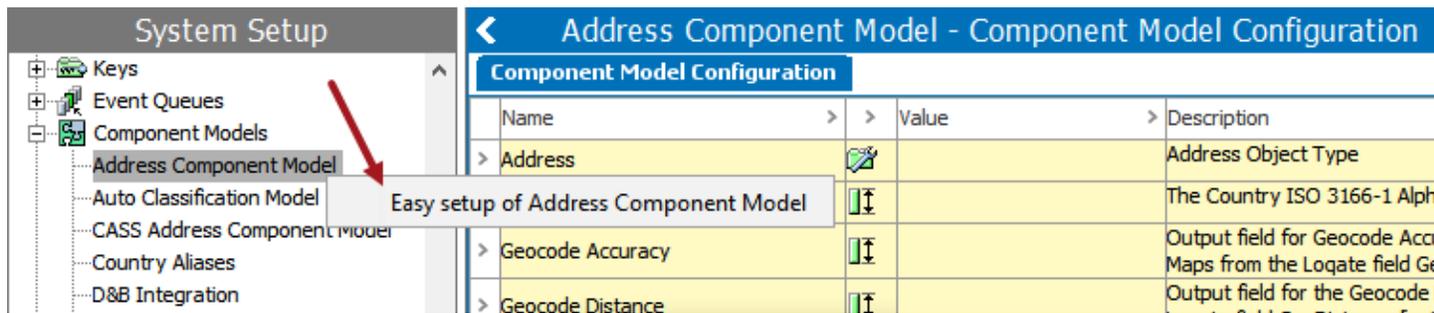
Easy Setup of Address Component Model

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.

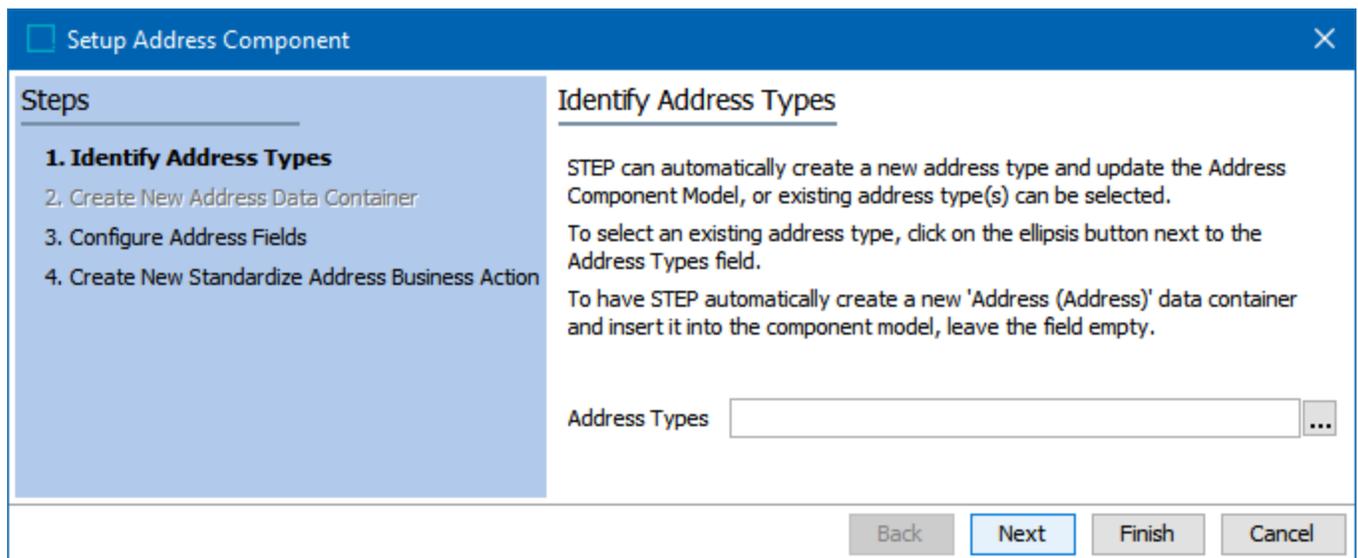
Note: This solution can be used both in on-premise deployments and in SaaS deployments.

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.

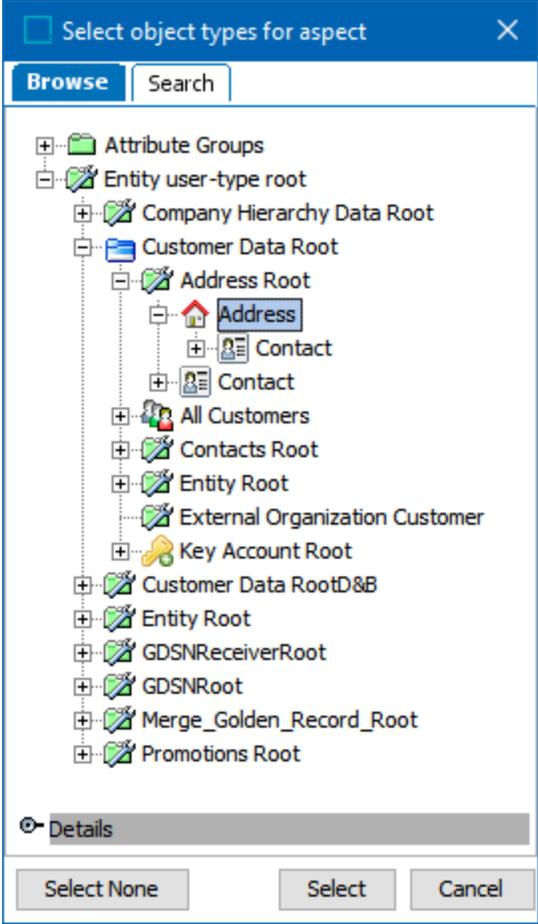
1. Locate the Address Component Model under System Setup > Component Models.
2. Right-click on the Address Component Model and click 'Easy setup of Address Component Model.'



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.



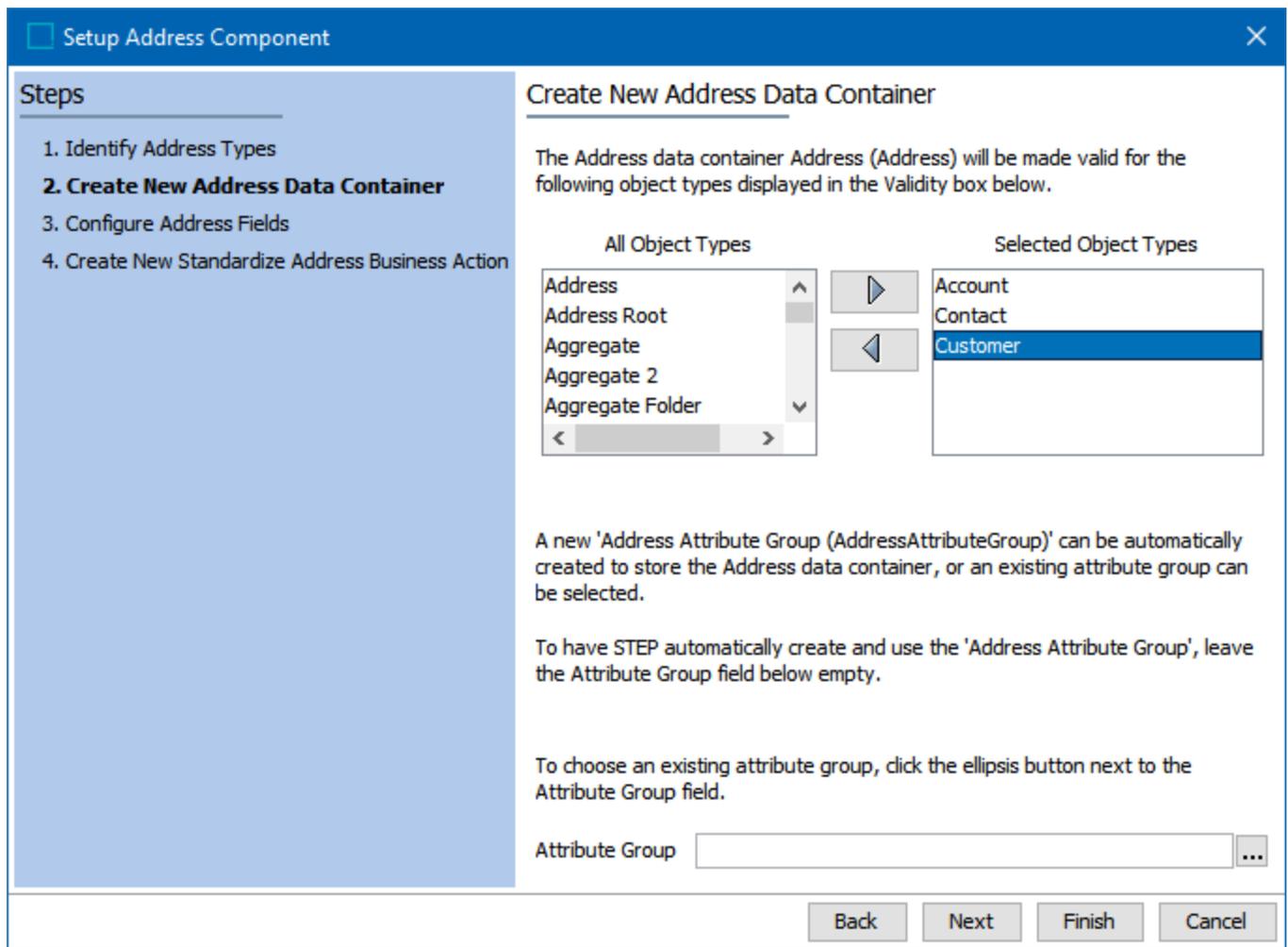
If you prefer to have STEP create the new object type, which will be named Address (ID = Address), leave the 'Address Types' field blank.

Click **Next**.

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



The screenshot shows a software window titled "Setup Address Component" with a close button (X) in the top right corner. On the left, a "Steps" sidebar lists four steps: 1. Identify Address Types, 2. Create New Address Data Container (highlighted), 3. Configure Address Fields, and 4. Create New Standardize Address Business Action. The main area is titled "Create New Address Data Container" and contains the following text: "The Address data container Address (Address) will be made valid for the following object types displayed in the Validity box below." Below this text are two list boxes: "All Object Types" containing "Address", "Address Root", "Aggregate", "Aggregate 2", and "Aggregate Folder"; and "Selected Object Types" containing "Account", "Contact", and "Customer" (which is highlighted). Between the lists are right and left arrow buttons. Further down, there is explanatory text: "A new 'Address Attribute Group (AddressAttributeGroup)' can be automatically created to store the Address data container, or an existing attribute group can be selected." and "To have STEP automatically create and use the 'Address Attribute Group', leave the Attribute Group field below empty." Below that, it says: "To choose an existing attribute group, click the ellipsis button next to the Attribute Group field." At the bottom of the main area is an "Attribute Group" text box with an ellipsis button (...). At the very bottom of the window are four buttons: "Back", "Next", "Finish", and "Cancel".

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, refer to the **Setting Up Data Container Types in Workbench** topic in the **Data Containers** documentation.

- 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, refer to 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).

Setup Address Component
✕

Steps

1. Identify Address Types
2. Create New Address Data Container
- 3. Configure Address Fields**
4. Create New Standardize Address Business Action

Configure Address Fields

STEP can automatically create new address attributes and update the Address Component Model, or existing attributes can be selected.

To have STEP automatically create an address attribute, leave the corresponding Value field blank in the table below and enable the respective checkbox in the Create column. Fields marked with * are required.

To select or remove an existing attribute, click on the corresponding ellipsis button in the Value column.

STEP can automatically create a new 'Address Attribute Group (AddressAttributeGroup)' to store the address attributes, or an existing attribute group can be selected.

To have STEP automatically create and use the 'Address Attribute Group', leave the Attribute Group field below empty.

To choose an existing attribute group, click the ellipsis button next to the Attribute Group field.

Attribute Group

Name	*	Value	>	Create
Country ISO Code			...	<input type="checkbox"/>
Geocode Accuracy			...	<input type="checkbox"/>
Geocode Distance			...	<input type="checkbox"/>
Geocode Latitude			...	<input type="checkbox"/>
Geocode Longitude			...	<input type="checkbox"/>
Input Address 1			...	<input type="checkbox"/>
Input Address 2			...	<input type="checkbox"/>
Input Address 3			...	<input type="checkbox"/>
Input Address 4			...	<input type="checkbox"/>
Input Address Line			...	<input type="checkbox"/>
Input Building			...	<input type="checkbox"/>
Input City	*	City (City)	...	<input type="checkbox"/>
Input Country	*	Country (Country)	...	<input type="checkbox"/>
Input County			...	<input type="checkbox"/>
Input Dependent Locality			...	<input type="checkbox"/>
Input Dependent Street			...	<input type="checkbox"/>

Back
Next
Finish
Cancel

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

- 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.

Setup Address Component

Steps

1. Identify Address Types
2. Create New Address Data Container
3. Configure Address Fields
- 4. Create New Standardize Address Business Action**

Create New Standardize Address Business Action

STEP can automatically create a new business action that can perform the standardization of addresses.

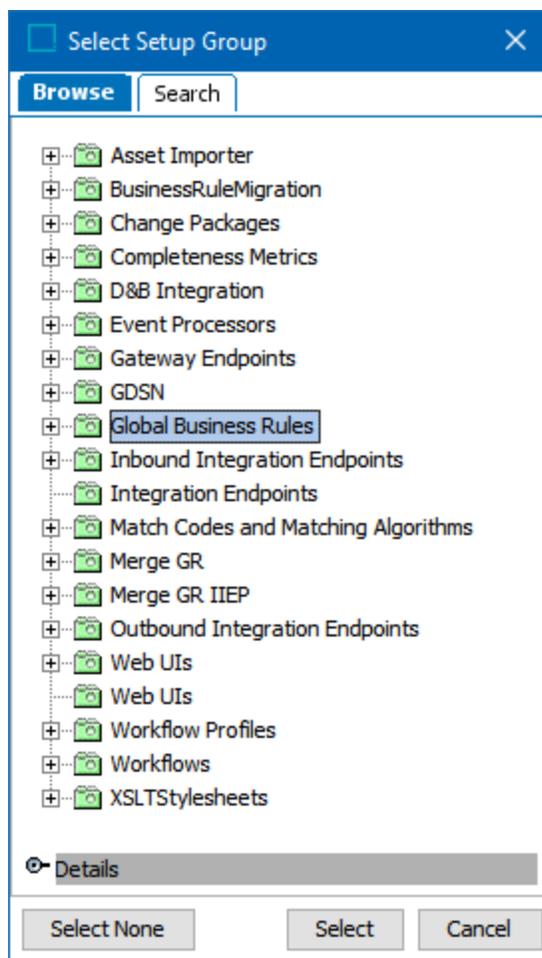
To have STEP automatically create the new 'Standardize Address Action (StandardizeAddressAction)' business action, click the ellipsis button next to the Setup Group field and choose the setup group in which the business action should be stored.

If no new business action should be created, leave the field empty.

Setup Group ...

Back Next **Finish** Cancel

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, refer to 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.

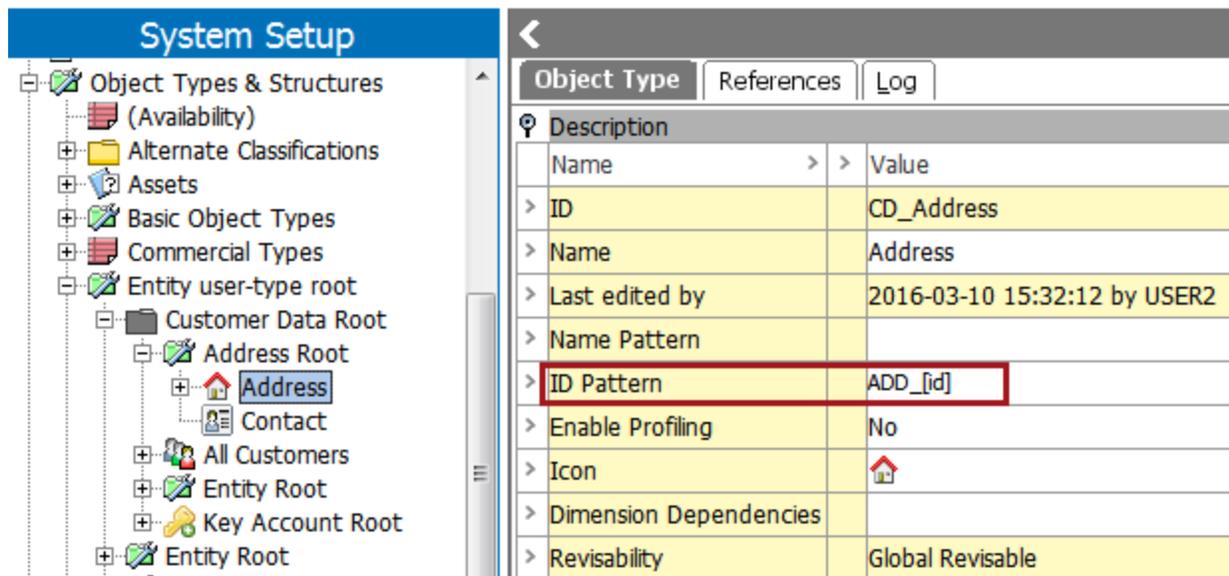
Note: This solution can only be used in an on-premise deployment.

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.

The screenshot shows the 'System Setup' interface on the left and the configuration details for the 'Customer To Address' reference type on the right. The left pane shows a tree view with 'Entity Reference Types' expanded to 'Customer To Address'. The right pane has tabs for 'Reference Type', 'Validity', and 'Log'. The 'Description' section contains the following table:

Name	Value
ID	CustomerToAddress
Name	Customer To Address
Last edited by	2016-03-14 18:41:49.724 by USER2
Externally Maintained	No
Dimension Dependencies	
Allow multiple references	Yes
Mandatory	No
Parent/Child relation	Source as parent, Target as Child
Inheritance	None
Completeness Score	123
Purpose	abc

The screenshot shows the 'System Setup' interface on the left and the configuration details for the 'Customer To Address' reference type on the right. The left pane shows a tree view with 'Entity Reference Types' expanded to 'Customer To Address'. The right pane has tabs for 'Reference Type', 'Validity', and 'Log'. The 'Valid Source Types' and 'Valid Target Types' sections contain the following tables:

ID	Name
CD_Customer	Customer

ID	Name
CD_Address	Address

- 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.

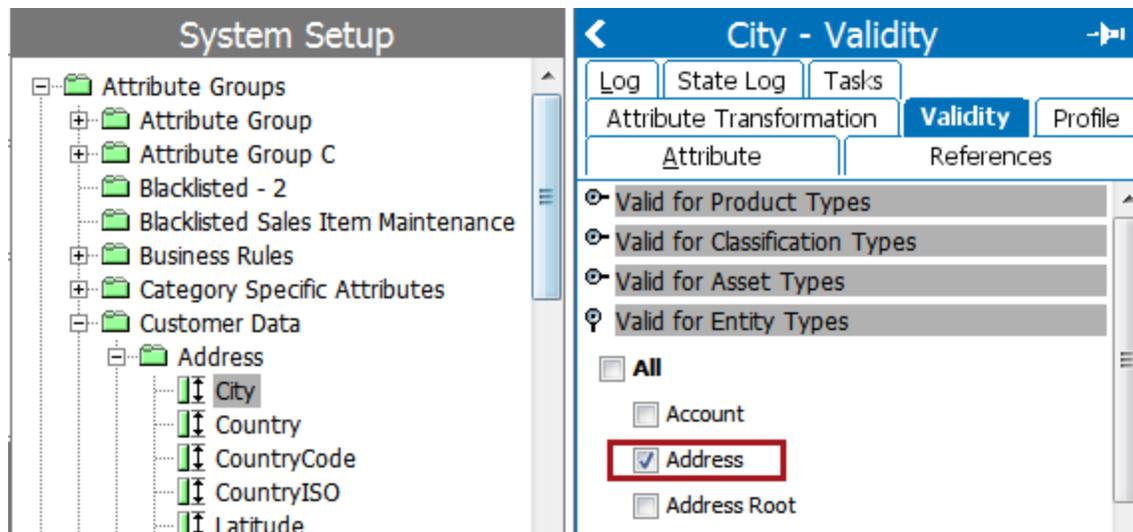
Attribute	References	Attribute Transformation	Validity	Profile
------------------	------------	--------------------------	----------	---------

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, refer to 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.

The screenshot displays the 'System Setup' navigation pane on the left, with 'Address Component Model' selected. The main window shows the 'Address Component Model - Component Model Configuration' table. The table lists various configuration items with their respective input and output fields. An 'Edit' button is located at the bottom of the table, highlighted with a red box and a red arrow pointing from the 'Address Component Model' entry in the left-hand tree.

Component Model Configuration			
> Standardized Street Type			Output field for street type, e.g. Road field ThoroughfareType [out].
> Standardized SubBuilding			Output field for secondary identifier from the Loqate field SubBuilding [ou
> Standardized Zip	Standardized Zip		Output field for complete postal cod
> Validation Hash	Validation Hash		Hash value for the Address Validatio
> Validation Integration Status			Field containing the latest status of
> Validation Response	Validation Response		Output field for the Cached Validatio
> Validation Time	Validation Time		Date and time stamp of the most rec

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 Locality	+					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 Code	⊗		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

CASS Address Component Model

Users of the Loqate Local solution 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 can qualify for discounted postage rates from the USPS. STEP makes it possible to standardize and validate addresses 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, refer to the Loqate 'CASS Overview' support website: <https://support.loqate.com/support/local-apis/cass-overview-2/>

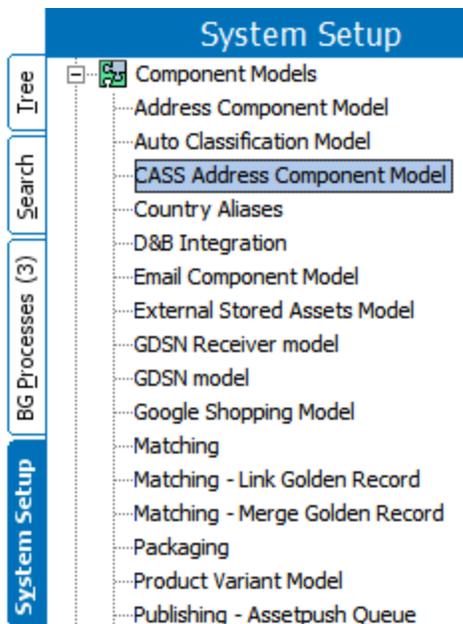
Prerequisites

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

- The STEP system deployment must be based in the United States due to regulations set by the United States Postal Service. However, users can use the output of the CASS validation process both within and outside of the United States.
- Users must be connected to a **Loqate Local** deployment and have a CASS license. The CASS solution will not work with a Loqate Cloud deployment.
- The Address Component Model must already be configured. For more information, refer to 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.

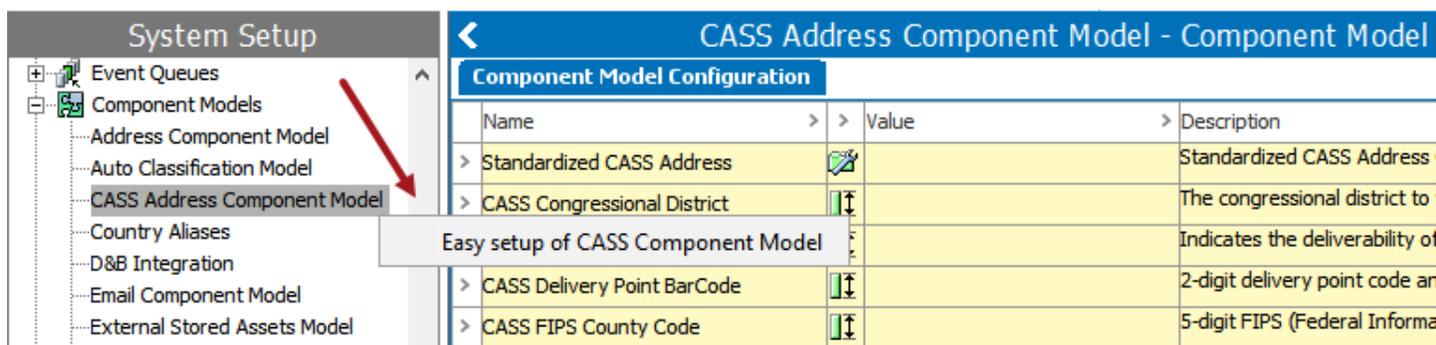
CASS Address Component Model Field	Description	Loqate Local Field Mapping
Address Object Types and/or Address Data Container Types		
Standardized CASS Address	The address types used to represent addresses, can be either entities or data containers	N/A
Standardized Address Attributes (OUTPUT Fields)		
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	<p>Collects remaining CASS output field values.</p> <p>Validation base type is text; maximum length must be at least 1000.</p> <div data-bbox="496 1125 1200 1465" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>CASS Validation Response abc</p> <p>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:</p> </div>	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

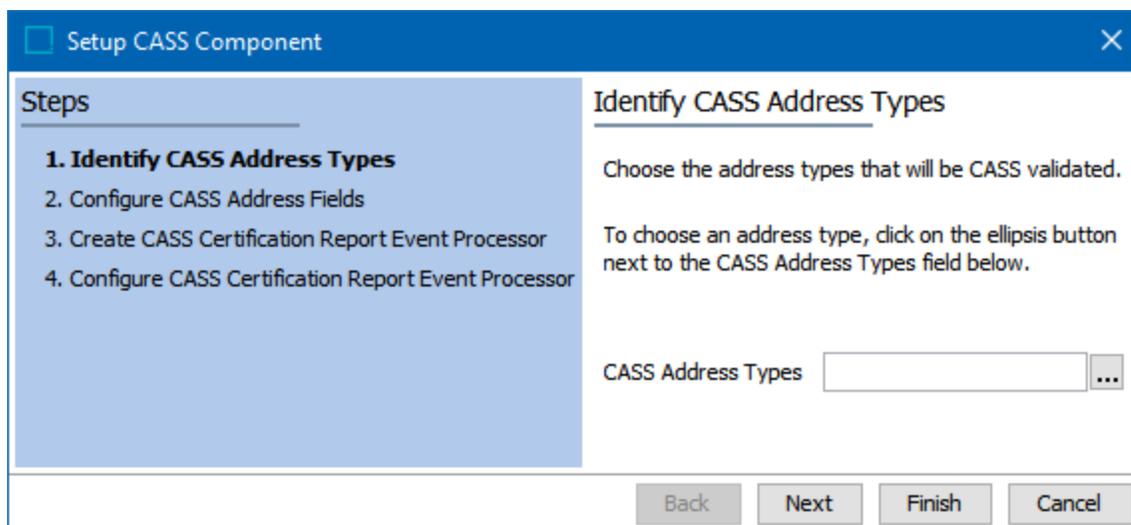
The recommended method for configuring the CASS Address Component Model is to use the 'Easy setup of CASS Component Model' wizard. By using this wizard, STEP can automatically create all of the address attributes required for the CASS solution, as well as a CASS Certification Report event processor used 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.

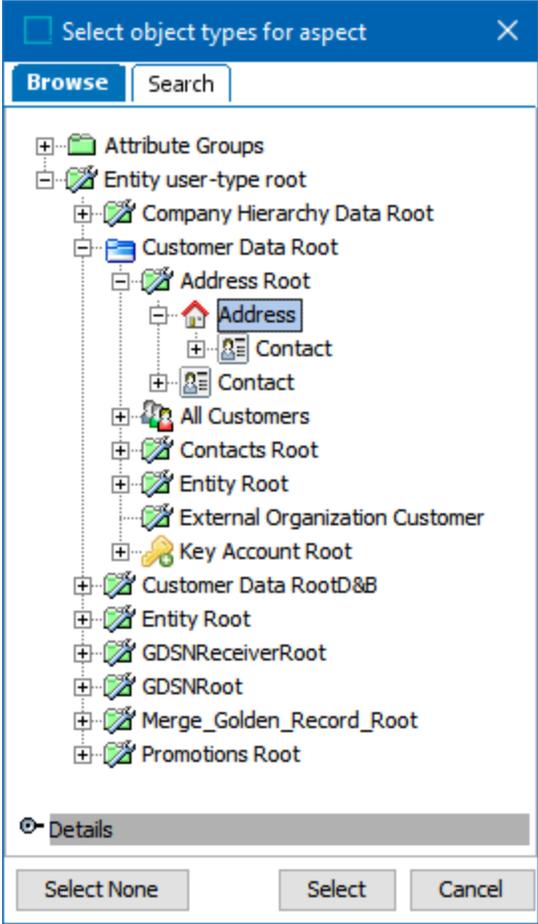
1. Locate the CASS Address Component Model under System Setup > Component Models.
2. Right-click on the CASS Address Component Model and click 'Easy setup of CASS Component Model.'



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).

Setup CASS Component
✕

Steps

1. Identify CASS Address Types
- 2. Configure CASS Address Fields**
3. Create CASS Certification Report Event Processor
4. Configure CASS Certification Report Event Processor

Configure CASS Address Fields

STEP can automatically create new address attributes and update the CASS Address Component Model, or existing attributes can be selected.

To have STEP automatically create an address attribute, leave the corresponding Value field blank in the table below and enable the respective checkbox in the Create column. Fields marked with * are required.

To select or remove an existing attribute, click on the corresponding ellipsis button in the Value column.

STEP can automatically create a new 'CASS Attribute Group (CASSAttributeGroup)' to store the address attributes, or an existing attribute group can be selected.

To have STEP automatically create and use the 'CASS Attribute Group', leave the Attribute Group field below empty.

To choose an existing attribute group, click the ellipsis button next to the Attribute Group field.

Attribute Group

Name	*	Value	>	Create	>
CASS Congressional District			...	<input type="checkbox"/>	
CASS Delivery Point BarCode			...	<input type="checkbox"/>	
CASS DPV Confirmed Indicator			...	<input type="checkbox"/>	
CASS FIPS County Code			...	<input type="checkbox"/>	
CASS No Stat Indicator			...	<input type="checkbox"/>	
CASS Residential Delivery			...	<input type="checkbox"/>	
CASS Vacant Indicator			...	<input type="checkbox"/>	
CASS Validation Response			...	<input type="checkbox"/>	

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

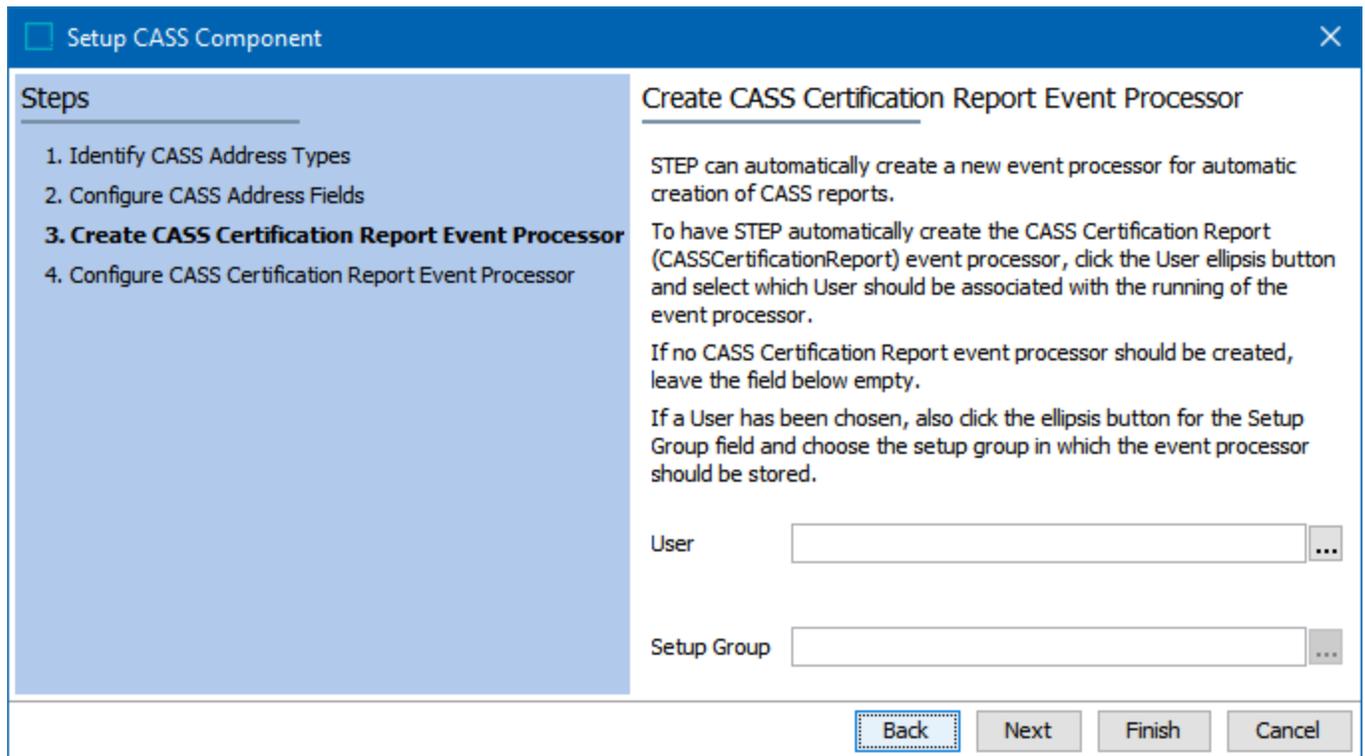
5. On the **Create CASS Certification Report Event Processor** screen, the system prompts you to choose a User and a Setup Group to have STEP automatically create a 'CASS Certification Report' event processor. This processor will be used to generate CASS certification reports. For more information on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

© Stibo Systems - Internal - Release 2023.3 (11.3) - September 2023

199

Processor' section of this topic (below). For more information on CASS certification reports, refer to 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.



The screenshot shows a software dialog box titled "Setup CASS Component" with a close button (X) in the top right corner. On the left, a "Steps" panel lists four steps: 1. Identify CASS Address Types, 2. Configure CASS Address Fields, 3. **Create CASS Certification Report Event Processor** (highlighted), and 4. Configure CASS Certification Report Event Processor. The main area is titled "Create CASS Certification Report Event Processor" and contains the following text: "STEP can automatically create a new event processor for automatic creation of CASS reports. To have STEP automatically create the CASS Certification Report (CASSCertificationReport) event processor, click the User ellipsis button and select which User should be associated with the running of the event processor. If no CASS Certification Report event processor should be created, leave the field below empty. If a User has been chosen, also click the ellipsis button for the Setup Group field and choose the setup group in which the event processor should be stored." Below this text are two input fields: "User" and "Setup Group", each with an ellipsis button (...) to its right. At the bottom right, there are four buttons: "Back" (highlighted with a dashed border), "Next", "Finish", and "Cancel".

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.

Setup CASS Component
✕

Steps

1. Identify CASS Address Types
2. Configure CASS Address Fields
3. Create CASS Certification Report Event Processor
- 4. Configure CASS Certification Report Event Processor**

Configure CASS Certification Report Event Processor

STEP can assist you in the configuration of the new CASS Certification Report event processor.

To have STEP automatically create and use the CASS Certification Report (CASSCertificationReport) asset object type, leave the field below empty. Otherwise, click the ellipsis button and choose an asset object type to use for CASS certification reports.

CASS Report Object Asset Type ...

Click the ellipsis button and choose a location in which to store the CASS certification reports. This field is mandatory.

CASS Report Location ...

CASS certification report generation must be activated by configuring the 'Standardize Address Action' business action (created on the Create New Standardize Business Action screen of the Easy setup of Address Component Model wizard) with the CASS event queue that was chosen or created above.

Complete setup of the event processor requires additional manual configuration. Refer to the Creating an Event Processor section of the System Setup online help documentation for more information.

Back
Next
Finish
Cancel

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, refer to 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.

System Setup

- Asset Importer
- BusinessRuleMigration
- Change Packages
- Completeness Metrics
- D&B Integration
- Event Processors
 - BestEP
 - CASS Certification Report
 - Email User when Attribute value
 - External Item Async Matching
 - Find Existing
 - Find Similar Search
 - Image Cache
 - LOV Value Merge Events
 - MatchCodeProcess
 - Matching Event
 - Matching Processor
 - Person Match
 - Product Revision Management
 - Update LOV Values
- Gateway Endpoints
- GDSN
- Global Business Rules
- Inbound Integration Endpoints
- Integration Endpoints
- Match Codes and Matching Algorithm
- Merge GR
- Merge GR IIEP
- Outbound Integration Endpoints
- Web UIs

CASS Certification Report - Event Processor

Background Processes
Statistics
Error Log Excerpts
Log

Event Processor

Event Triggering Definitions

Description

Name	Value
> ID	CASSCertificationReport
> Name	CASS Certification Report
> Type	Event Processor
> Last edited by	2017-06-11 15:45:09 by USER4
> Enabled	Yes
> Processor Status	Running

Configuration

ID	Name
User running event process...	User 4
Number of events to batch	10000
Days to retain events	0
Queue for event processor	CASSCertificationReportEventQueue
Maximum number of old proc...	100
Maximum age of old process...	168
Limit of lines in execution report	1000
Processor	CASS Certification Report
Schedule	Not scheduled ...
Queue Status	Read Events
Unread events (approximated)	Click to estimate ...

[Edit Configuration](#)

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, refer to the **EP - 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, IqtBatch, 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 lqtBatch, 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 screenshot displays a file management interface. On the left, a tree view shows a folder named 'Assets' containing a sub-folder 'Matching Tables' and a list of CASS report files. The selected file is 'CASS_report_20170524155443'. The main pane shows the details for this file, including its ID, name, object type, revision, and path. The 'System Properties' section is expanded, showing a table of file attributes. The 'Filename' attribute is highlighted with a red box and a red circle containing the number '1', and the version number '2' is also highlighted with a red box and a red circle containing the number '2'.

Name	Value
ID	CASS_report_20170524155443
Name	CASS_report_20170524155443
Object Type	CASS Certification Report
Revision	1.0 Last edited by STIBOCMDM on Wed May 24 15:54:45 EDT 2017
Approved	✘ Never Been Approved
Translation	Not Translated
Path	Classification 1 root/Assets/CASS_report_20170524155443

Name	Value
Extension	abc txt
Filename	abc CASS_report_20170524155443_lqtv_2_23_0_9136.txt
Format	abc Text (Plain ASCII text)
MIME Type	abc text/plain; charset=us-ascii
Size	abc 176,842
Upload Time	abc 2017-05-24 15:54:45

The below is a sample CASS certification report.

```

CASS_report_20170524155443.txt - Notepad
File Edit Format View Help
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|||
4|null|Kennesaw|null|null|null|||
5|null|Kennesaw|null|30144|null|null|||
6|null|null|null|null|null|||
7|null|Kennesaw|null|null|null|||
8|null|Kennesaw|null|30144|null|null|||
9|null|null|null|null|null|||
10|null|Kennesaw|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|||

```

Migrating Loqate from Cloud to Local

Current users of the Loqate Cloud solution can easily migrate 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 screenshot shows the Loqate interface. On the left is a 'Tree' view with a hierarchy: Assets, Classifications, Configurations, Index Words, Merchandising Hierarchy, Suppliers, Web Sites US, Company Hierarchy Data Root, Customer Root, Entity Root, and Address Root. Under 'Address Root', the address '3550 George Busbee Pkwy' is selected. On the right, the '3550 George Busbee Pkwy rev.0.12 - Address' details are shown. The 'Validation Hash' field contains the value '499157146'. The 'Validation Response' field contains the following XML:

```

Loqate/499157146/?xml version="1.0" encoding="UTF-8"?>
<qt>
  <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</Delivery
  
```

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 deployment.

The screenshot shows the 'Bulk Update' dialog box. The 'Steps' list on the left includes: 1. Configuration, 2. Operations (selected), 3. Parameters, 4. Preview, and 5. Advanced. The 'Operations' section shows 'Standardize Address' selected. The 'Address Standardization Mode' is set to 'Overwrite Existing Standardized Address'. There is a field for 'Renew address validations older than (if left empty, this option is ignored) Days'. At the bottom, there are buttons for 'Back', 'Next', 'Finish', and 'Cancel'.

For more information on the 'Generate Hash Value Only' option, refer to the following topics:

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

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 Loqate Local solution that allows users to use the Loqate address standardization without integrating directly with the API or interfacing with STEP.

For example, you can 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.

The command line tool can process input addresses that are stored in valid UTF-8 delimited files, which 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/>.

Stibo Systems advises you download and install locally the software from Loqate website to use it.

You can download the Local command line tool from a Loqate FTP site. To acquire the access credentials, contact Stibo Systems support.

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
#=====#

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 Loqate Local 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 more information on SaaS deployments, refer to the process for raising support issues with Stibo Systems in the **Loqate Integration in STEP SaaS Deployments** topic.

For more information for on-premises deployments, refer to the **Loqate Integration in STEP On-Premises Deployments** topic.

Loqate Integration in STEP SaaS Deployments

Integrating Loqate with STEP on a SaaS deployment is completely hands free. All that is needed is to create an issue in the customer support portal requesting Stibo Systems begin the integration process. To do so, while in the support portal, create an issue and write your request in the issue description field. From there, Stibo Systems completes the rest of the process for you. If you need more information on how to create an issue, so the videos available to you in the support portal via the 'How To' Videos Clips dashboard available to all users.

Important: The above process is valid only when the Loqate Local has been purchased via your Stibo Systems account manager

You can create two types of issues in the support portal when integrating Loqate in a SaaS deployment: one to request an initial installation of Loqate, and one to update the data packs. Be aware that while updating the data packs, there is no downtime in STEP or Loqate services, whereas with the initial installation there is downtime.

Below is an example of a support issue. The mandatory fields to include in the issue are:

- Issue Type: [Task]
- Summary: [Install Loqate Local on env <environment>]
- Description: [Provide a description of the issue (e.g., 'Please install Loqate Local on env <environment>')]
- Workday Task: [Customer name - Client Support]
- STEP Release: [Your STEP release version]
- Reporter: [Your name]
- Issue Category: [STEP System Administration]
- Business Domain: [CMDM]

If you need the deployment in a particular environment or on a specific timeline, include this information in the issue.

Loqate Integration in STEP On-Premises Deployments

For users that keep their Loqate integration service implemented solely on-premise, the following topics outline Loqate's capabilities and the deployment and upgrade processes:

- Loqate Integration Properties
- Loqate Local API Testing
- Loqate Local License Key & Data Package Update

Prerequisites

The following subsections explain the prerequisites that must be met before installing the Loqate Cloud and Local and associated reference data on your STEP application server.

System Requirements for Loqate Local

A considerable amount of disk space is needed for Loqate reference data.

Additionally, a number of storage areas are required. Review the description of each storage area in **Local Only** section of the **Loqate Integration Properties** topic.

- **LoqateLocal.DataPack.SharedFolder** (Shared storage): It is recommended to allocate 150 GB of storage to this location.
- **LoqateLocal.DataPack.LocalRoot** (Local storage): Stibo Systems recommends you allocate 150 GB of storage to this location.
- **LoqateLocal.DataPack.LocalFolder** (Local storage): Stibo Systems recommends you allocate 100 GB of storage to this location.

Review the general shared storage requirements in the System Administration documentation. As this storage is used for random access at runtime, it is required to use RAIDed/SSD storage for optimal performance.

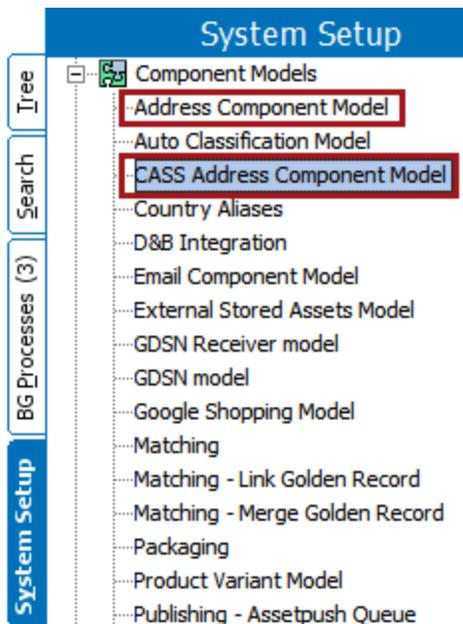
Install Recipes for Loqate Local and CASS

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. To obtain the install recipe, speak with your Stibo Systems account manager.

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. Furthermore, users should select the recipe compatible with their STEP system and always verify that it is the correct recipe before installation.

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

Included in the Loqate (and CASS) installation are the Address Component Model and the CASS Address Component Model. These must be configured for an on-premise deployment to work.



Refer to the **Address Component Model** and the **CASS Address Component Model** topics in this documentation for more information.

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, 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

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** - This address quality plugin is used for validation and address quality measurement. Insert the value, 'LoqateAddressQuality'
- **Address.Service.Loqate.GeocodeAll** - This property is to enable / retrieve geocode coordinates when making an address standardization call. Insert 'true' or 'false' according to whether or not you have a license for it.

Important: If this property is not set, its default value is 'true'

Cloud Only

The following properties are required for the Loqate Cloud solution:

```
#=====
# Loqate Install Test
#=====
Address.Service.Loqate.License = <Insert License Key>
Address.AddressQualityPlugin = LoqateAddressQuality
Address.Service.Loqate.GeocodeAll = true
```

Configuration Property	Description
<code>Address.Service.Loqate.License</code>	This sharedconfig.properties specifies the license key of the Loqate Cloud solution.

Local Only

The following properties are required for the Loqate Local solution:

```

#=====
# Loqate Install Test
#=====
LoqateLocal.DataPack.LicenseKey = <Insert License Key>
LoqateLocal.DataPack.LocalRoot = /mnt/loqate-disk
LoqateLocal.DataPack.SharedFolder = /shared/workarea/localloqate
LoqateLocal.DataPack.LocalFolder = /mnt/loqate-disk/data-packs
LoqateLocal.DataPack.DeployID = <Insert Deploy ID>

```

Configuration Property	Description
<code>LoqateLocal.DataPack.LicenseKey</code>	The license key provided by Loqate.
<code>LoqateLocal.DataPack.LocalRoot</code>	<p>The local root where the Data Pack Creator processes and downloads the files from the Loqate server. Once all files are downloaded and installed, the process will package them</p> <p>If this property is not set in its default value is:</p> <p><code>/mnt/loqate-disk</code></p>
<code>LoqateLocal.DataPack.SharedFolder</code>	<p>A shared folder path which is accessible from all other cluster nodes and the Data Pack Creator. After downloading the Loqate files, the scripts create a .zip file directly into the shared folder (the folder with the package ID). Inside the folder is the Loqate version file. After deploying the data packs to the local folder, this folder can be cleared.</p> <p>If this property is not set, the default value is:</p> <p><code>/shared/workarea/localloqate</code></p>
<code>LoqateLocal.DataPack.LocalFolder</code>	<p>This determines where the data pack should be deployed after it is unzipped from the shared folder.</p> <p>If this property is not set, the default value is:</p>

Configuration Property	Description
	/mnt/loqate-disk
LoqateLocal.DataPack.DeployID	The ID used is from the Data Package Creator process.

Loqate Local API Testing

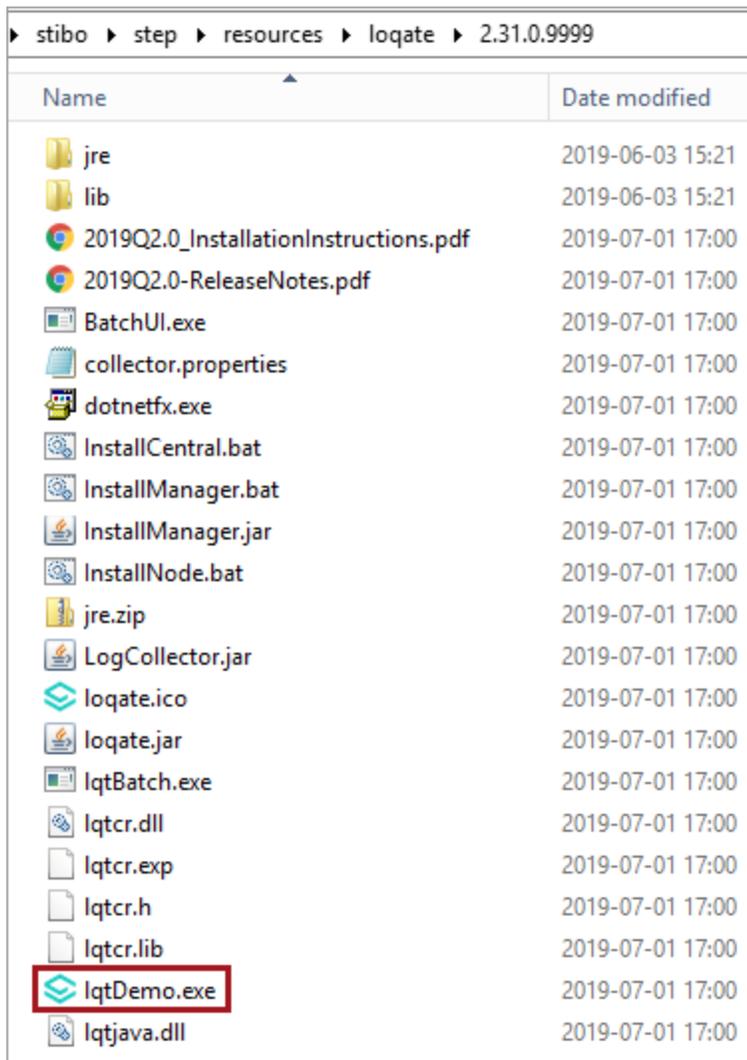
As part of troubleshooting problem with integration between STEP and Loqate Local, it is possible to test if the installed Loqate Engine and Loqate Reference Data works as intended and is ready to be used by STEP. This can be done by logging on to the STEP application server and executing the Local Loqate engine directly.

Linux

For the Linux users, use the lqtBatch.exe app and follow the steps (outlined at <https://support.loqate.com/batch-processor/>) to go through the process on a command line interface.

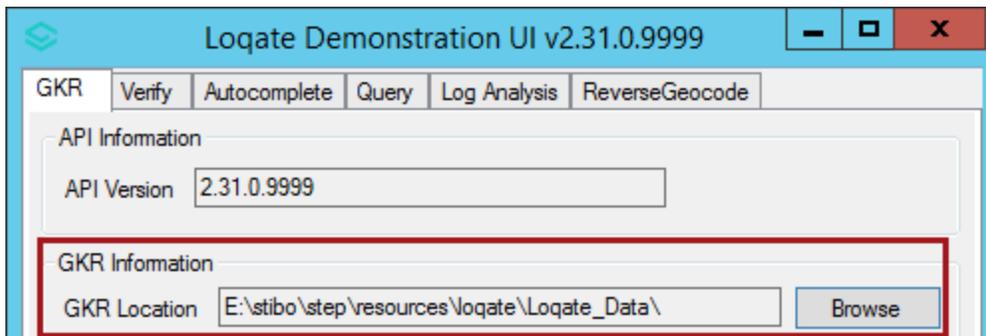
Windows

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.



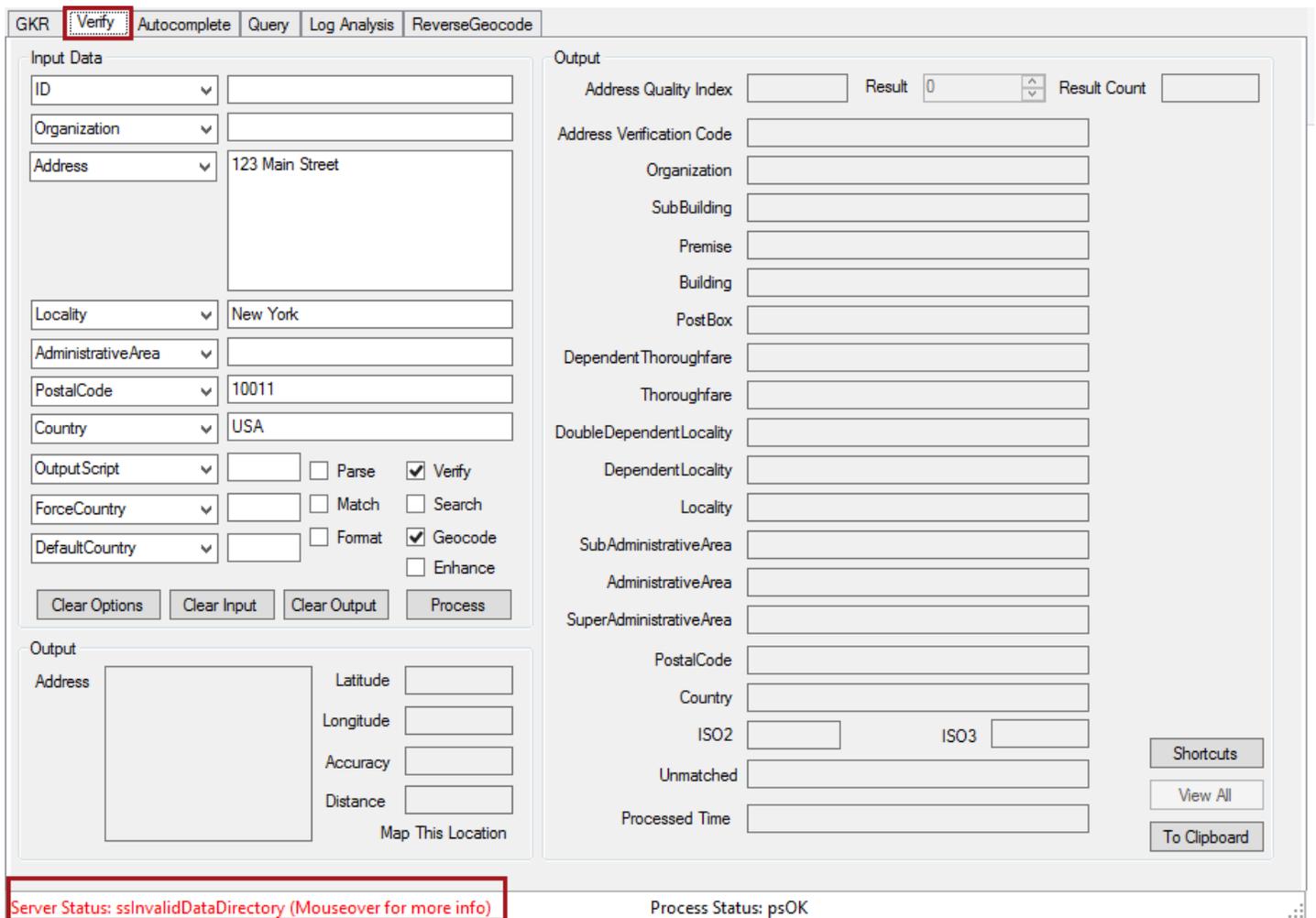
Name	Date modified
Folder jre	2019-06-03 15:21
Folder lib	2019-06-03 15:21
2019Q2.0_InstallationInstructions.pdf	2019-07-01 17:00
2019Q2.0-ReleaseNotes.pdf	2019-07-01 17:00
BatchUI.exe	2019-07-01 17:00
collector.properties	2019-07-01 17:00
dotnetfx.exe	2019-07-01 17:00
InstallCentral.bat	2019-07-01 17:00
InstallManager.bat	2019-07-01 17:00
InstallManager.jar	2019-07-01 17:00
InstallNode.bat	2019-07-01 17:00
jre.zip	2019-07-01 17:00
LogCollector.jar	2019-07-01 17:00
loqate.ico	2019-07-01 17:00
loqate.jar	2019-07-01 17:00
lqtBatch.exe	2019-07-01 17:00
lqtcr.dll	2019-07-01 17:00
lqtcr.exp	2019-07-01 17:00
lqtcr.h	2019-07-01 17:00
lqtcr.lib	2019-07-01 17:00
lqtDemo.exe	2019-07-01 17:00
lqtjava.dll	2019-07-01 17:00

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 there are no issues with the Loqate API and reference data deployment, 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

Locality

AdministrativeArea

PostalCode

Country

OutputScript Parse Verify

ForceCountry Match Search

DefaultCountry Format Geocode Enhance

Output

Address Quality Index Result Result Count

Address Verification Code

Organization

SubBuilding

Premise

Building

PostBox

DependentThoroughfare

Thoroughfare

DoubleDependentLocality

DependentLocality

Locality

SubAdministrativeArea

AdministrativeArea

SuperAdministrativeArea

PostalCode

Country

ISO2 ISO3

Unmatched

Processed Time

Output

Address

Latitude

Longitude

Accuracy

Distance

Server Status: ssOK
Process Status: psOK

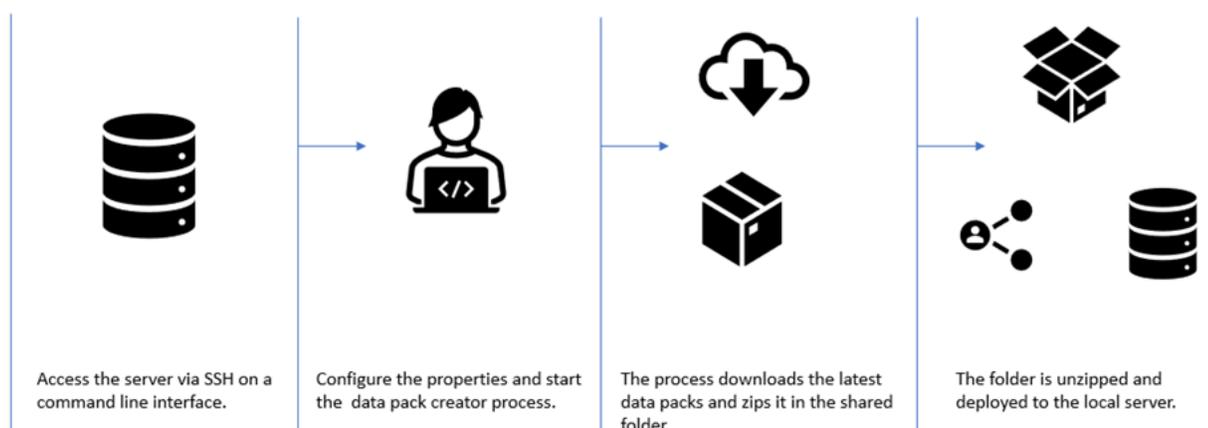
Loqate Local License Key & Data Package Update

Loqate reference data packs must be updated on a regular basis as 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/release-information/>. Additionally, any upgrades to a newer STEP version warrant an update to the Loqate reference data packs.

Deploying the Loqate reference data packs for on-premise servers consists of several steps that are outlined below:

1. Access the server where you have deployed STEP via SSH in a command line interface.
2. Change the properties on the sharedconfig.properties file.
3. Start the data pack creator process by using the scripts (Windows / Linux) outlined in the Create Data Packages section.
4. The most up-to-date data pack the system is licensed for will install, and will be placed into a temporary shared folder.
5. Lastly, start the deploy process outlined in the Deploy Data Package section of this topic.

During both the creation and deployment stages, there is no downtime in STEP or Loqate services.



Prerequisites

- Loqate Local component version 7.0.20 or newer must be installed.
- STEP version 10.3 MP4 or newer must be installed.
- System must have the required storage space, as defined in the **System Requirements for Loqate Local** section of the **Loqate Integration in STEP On-Premises Deployments** topic.
- Review the **Loqate Integration Properties** topic to ensure that the properties described in the Local Only section are live.

Create Data Packages

The following steps outline how to create a data package, which is used to start the update process:

1. Open your SSH connection and open the sharedconfig.properties file inside the STEP server.

```

login as: root
root@stibotest-tsus.stibo.com's password:
Last login: Thu Oct 13 08:24:29 2022 from 10.0.246.53
[root@stibotest-tsus ~]# su - stibosw
Last login: Thu Oct 13 08:24:50 EDT 2022 on pts/0
Source /opt/stibo/step/admin/profile.d/10-generic.sh
Source /opt/stibo/step/admin/profile.d/20-oracle-instant-client.sh
[stibosw@stibotest-tsus ~]$ cd step/
[stibosw@stibotest-tsus step]$ ./spot -c
Stibo Patch Operations Tool - stibotest-tsus - test - stibotest-tsus-stibotest-t
sus.stibo.com-20220428183853 - stibotest-tsus.stibo.com
This STEP system has the following software installed:

Baseline:
    step-11.0-mp1-2022-07-01-13-10-16

Components installed in addition to baseline:
    spot-7.0.142
    local-loqate-7.0.20
    loadbalancer-frontend-7.0.33
    loadbalancer-backend-7.0.23

Hotfix:
    HOTFIX-5930

This is a summarized view of the system state, in order to produce a complete
image of the system state, please use: spot --snapshot=snapshot.spr
[stibosw@stibotest-tsus step]$ █

```

2. Navigate to the Data Pack Creator scripts under the STEP-ROOT/resources/loqate/datapack_creator. You can then view the start_datapack_creator.sh script file.

```

This is a summarized view of the system state, in order to produce a complete
image of the system state, please use: spot --snapshot=snapshot.spr
[stibosw@stibotest-tsus step]$ cd resources/loqate/datapack_creator/
[stibosw@stibotest-tsus datapack_creator]$ ll
total 416
-rw-rw-r--+ 1 stibosw stibosw 23185 Jun  3 12:41 com.stibo.loqate.localapi.installer-trunk-1.0.0.jar
-rw-rw-r--+ 1 stibosw stibosw   926 Oct 12 11:09 datapack_creator.log
-rw-rw-r--+ 1 stibosw stibosw 46012 Oct 11 10:21 hs_err_pid18202.log
-rw-rw-r--+ 1 stibosw stibosw 46010 Aug 19 13:51 hs_err_pid22052.log
-rw-rw-r--+ 1 stibosw stibosw 51626 Jul 13 18:24 hs_err_pid32567.log
-rw-rw-r--+ 1 stibosw stibosw 46024 Jul 21 22:49 hs_err_pid3551.log
-rw-rw-r--+ 1 stibosw stibosw 45678 Jul 21 10:10 hs_err_pid5104.log
-rw-rw-r--+ 1 stibosw stibosw 46063 Jul 19 11:09 hs_err_pid6120.log
-rw-rw-r--+ 1 stibosw stibosw 51842 Jul 13 18:30 hs_err_pid732.log
-rwxrwxr-x+ 1 stibosw stibosw   360 Oct 12 03:22 start_datapack_creator.sh
[stibosw@stibotest-tsus datapack_creator]$ vi start_datapack_creator.sh

```

3. Edit the data pack creator script by inserting the full path of your sharedconfig.properties file (e.g., /workarea/sharedconfig.properties) after both instances of the 'com.stibo.loqate.localapi.installer.LocalLoqateInstaller' path below.

```

if which java ; then java -cp ".:*" com.stibo.loqate.localapi.installer.LocalLoqateInstaller /workarea/sharedconfig.properties; else /opt/stibo/step/resources/jdk/`cat /opt/stibo/step/resources/jdk/jdk.installed | grep JavaHome | cut -d'=' -f2`/bin/java -cp ".:*" com.stibo.loqate.localapi.installer.LocalLoqateInstaller /workarea/sharedconfig.properties; fi

```

4. Run the data pack creator script.

```

[stibosw@stibotest-tsus step]$ cd resources/loqate/datapack_creator
[stibosw@stibotest-tsus datapack_creator]$ ll
total 416
-rw-rw-r--+ 1 stibosw stibosw 23185 Jun  3 12:41 com.stibo.loqate.localapi.installer-trunk-1.0.0.jar
-rw-rw-r--+ 1 stibosw stibosw   924 Oct 12 06:52 datapack_creator.log
-rw-rw-r--+ 1 stibosw stibosw 46012 Oct 11 10:21 hs_err_pid18202.log
-rw-rw-r--+ 1 stibosw stibosw 46010 Aug 19 13:51 hs_err_pid22052.log
-rw-rw-r--+ 1 stibosw stibosw 51626 Jul 13 18:24 hs_err_pid32567.log
-rw-rw-r--+ 1 stibosw stibosw 46024 Jul 21 22:49 hs_err_pid3551.log
-rw-rw-r--+ 1 stibosw stibosw 45678 Jul 21 10:10 hs_err_pid5104.log
-rw-rw-r--+ 1 stibosw stibosw 46063 Jul 19 11:09 hs_err_pid6120.log
-rw-rw-r--+ 1 stibosw stibosw 51842 Jul 13 18:30 hs_err_pid732.log
-rwxrwxr-x+ 1 stibosw stibosw   360 Oct 12 03:22 start_datapack_creator.sh
[stibosw@stibotest-tsus datapack_creator]$ vi start_datapack_creator.sh
[stibosw@stibotest-tsus datapack_creator]$ ./start_datapack_creator.sh
/bin/java

```

STEP begins downloading the data pack into the shared folder and returns to the command line upon finishing. This process should take approximately 40 minutes.

```
This is a summarized view of the system state, in order to produce a complete
image of the system state, please use: spot --snapshot=snapshot.spr
[stibosw@stibotest-tsus step]$ cd resources/loqate/datapack_creator/
[stibosw@stibotest-tsus datapack_creator]$ ll
total 416
-rw-rw-r--+ 1 stibosw stibosw 23185 Jun  3 12:41 com.stibo.loqate.localapi.installer-trunk-1.0.0.jar
-rw-rw-r--+ 1 stibosw stibosw   926 Oct 12 11:09 datapack_creator.log
-rw-rw-r--+ 1 stibosw stibosw 46012 Oct 11 10:21 hs_err_pid18202.log
-rw-rw-r--+ 1 stibosw stibosw 46010 Aug 19 13:51 hs_err_pid22052.log
-rw-rw-r--+ 1 stibosw stibosw 51626 Jul 13 18:24 hs_err_pid32567.log
-rw-rw-r--+ 1 stibosw stibosw 46024 Jul 21 22:49 hs_err_pid3551.log
-rw-rw-r--+ 1 stibosw stibosw 45678 Jul 21 10:10 hs_err_pid5104.log
-rw-rw-r--+ 1 stibosw stibosw 46063 Jul 19 11:09 hs_err_pid6120.log
-rw-rw-r--+ 1 stibosw stibosw 51842 Jul 13 18:30 hs_err_pid732.log
-rwxrwxr-x+ 1 stibosw stibosw   360 Oct 12 03:22 start_datapack_creator.sh
[stibosw@stibotest-tsus datapack_creator]$ vi start_datapack_creator.sh
2 files to edit
[stibosw@stibotest-tsus datapack_creator]$ vi start_datapack_creator.sh
[stibosw@stibotest-tsus datapack_creator]$ ./start_datapack_creator.sh
/bin/java
[stibosw@stibotest-tsus datapack_creator]$ █
```

5. Confirm that the process was successfully completed by going to the data pack creator log file.

```
[stibosw@stibotest-tsus step]$ cd resources/loqate/datapack_creator/
[stibosw@stibotest-tsus datapack_creator]$ ll
total 416
-rw-rw-r--+ 1 stibosw stibosw 23185 Jun  3 12:41 com.stibo.loqate.localapi.installer-trunk-1.0.0.jar
-rw-rw-r--+ 1 stibosw stibosw   914 Oct 13 09:37 datapack_creator.log
-rw-rw-r--+ 1 stibosw stibosw 46012 Oct 11 10:21 hs_err_pid18202.log
-rw-rw-r--+ 1 stibosw stibosw 46010 Aug 19 13:51 hs_err_pid22052.log
-rw-rw-r--+ 1 stibosw stibosw 51626 Jul 13 18:24 hs_err_pid32567.log
-rw-rw-r--+ 1 stibosw stibosw 46024 Jul 21 22:49 hs_err_pid3551.log
-rw-rw-r--+ 1 stibosw stibosw 45678 Jul 21 10:10 hs_err_pid5104.log
-rw-rw-r--+ 1 stibosw stibosw 46063 Jul 19 11:09 hs_err_pid6120.log
-rw-rw-r--+ 1 stibosw stibosw 51842 Jul 13 18:30 hs_err_pid732.log
-rwxrwxr-x+ 1 stibosw stibosw   360 Oct 13 08:47 start_datapack_creator.sh
[stibosw@stibotest-tsus datapack_creator]$ vi start_datapack_creator.sh
[stibosw@stibotest-tsus datapack_creator]$ ./start_datapack_creator.sh
/bin/java
[stibosw@stibotest-tsus datapack_creator]$ vi datapack_creator.log █
```

A success message should display.

```

This will create a Data Pack for Loqate.
Reading sharedconfig.properties file.
█
Properties used by Data Pack Creator:
- JAVA_HOME: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.171-8.b10.el7_5.x86_64/jre
- LoqateLocal.DataPack.SharedFolder: /opt/stibo/step/LOQATE/sharedfolder
- LoqateLocal.DataPack.LocalRoot: /opt/stibo/step/LOQATE/localroot
- Clean Local: true

Starting up Loqate Data Pack Manager.

Downloading Loqate Data Pack .....
.....
Loqate Data Pack Manager finished with code: 0
Loqate Data Pack Manager finished successfully.

Compressing Loqate Data Pack .....Cleaning local files.

Loqate Data Pack with id: <Data Pack id> was successfully created.

```

Important: Make note of the 'Loqate Data Pack with id' line and the ID created. You will need this ID for the Deploy Data Pack section.

Deploy Data Packages

Once you have created a data pack, STEP can package and move it to the local root. From there, STEP deploys it to a local folder for use, and the Loqate reference data packs are updated. The following steps outline how to deploy a data pack.

1. Open your SSH connection and open the sharedconfig.properties file inside the STEP server.
2. Edit the LoqateLocal.DataPack.DeployID property by entering the value of the data pack ID you created in the Create Data Pack section. Save, and the deployment process begins automatically. This should take approximately 20 minutes.
3. Confirm the successful deployment by checking the STEP.0 logs.

```

SPOT is already running
For the last 219 seconds, this command has been running: spot --apply=/opt/stibo/step/admin/spot/recipes
/upgrade/upgrade.LISTED.2022-10-13-11-16-28.spr
Try again later when the other SPOT instance has finished.
[stibosw@stibotest-tsus step]$ /opt/stibo/step/diag/logs/step.0.log
-bash: /opt/stibo/step/diag/logs/step.0.log: Permission denied
[stibosw@stibotest-tsus step]$ vi step.0.log
[stibosw@stibotest-tsus step]$ vi /opt/stibo/step/diag/logs/step.0.log
[stibosw@stibotest-tsus step]$ vi /opt/stibo/step/diag/logs/step.0.log

```

A successful message should appear.

```

2022/10/13-11:19:13 1 com.stibo.loqate.localapi.impl.autostart.DeployDataPackAutostart start INFO: New Node is starting. Will deploy and activate t
he loqate Data Pack with ID: a132ceb2-7f75-4721-94e9-2bbf4d294ae7, on node: stibotest-tsus.stibo.com
2022/10/13-11:19:13 1 com.stibo.loqate.localapi.impl.autostart.DeployDataPackAutostart deploy INFO: Deploying Data Pack with ID: a132ceb2-7f75-4721-
94e9-2bbf4d294ae7, on node: stibotest-tsus.stibo.com
2022/10/13-11:28:41 1 com.stibo.loqate.localapi.impl.autostart.DeployDataPackAutostart activate INFO: Activating Data Pack with ID: a132ceb2-7f75-4721-94e9-2bbf4d294ae7, on node: stibotest-tsus.stibo.com
2022/10/13-11:28:41 1 com.stibo.loqate.localapi.impl.autostart.DeployDataPackAutostart activate INFO: Activated on: stibotest-tsus.stibo.com

```

No restart of STEP is required to update.

Product Data Exchange

Product Data Exchange is a platform or foundation which holds product data and provides powerful capabilities to reuse and transform master data to meet various requirements of downstream recipients such as: retailers, e-commerce platforms, internal systems, and other digital channels.

Product Data Exchange has two different services: PDX Syndication and PDX Onboarding.

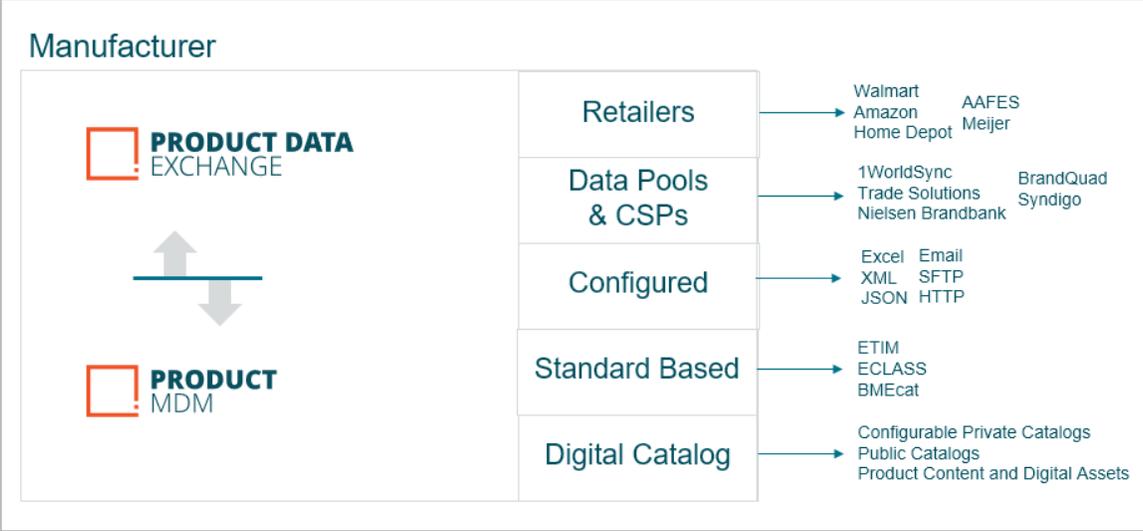
PDX Syndication allows brand manufacturers and distributors to utilize the complete toolkit of the platform to prepare product data for submission to a variety of pre-built and configured channels. For more information on how to configure STEP to provide PDX the product information for later syndication to the retailers of interest, refer to the **PDX Syndication** topic in this documentation.

PDX Onboarding allows retailers to invite their suppliers to PDX, providing the capabilities necessary to syndicate their product data to the retailer who invited them. This enables retailers to consistently receive product data of high quality and completeness. For more information on how to configure STEP to receive product information from all the suppliers using PDX, refer to the **PDX Onboarding** topic in this documentation.

For detailed information on how to get an administrative account for PDX and specific PDX documentation, contact your Stibo Systems representative.

PDX Syndication

PDX Syndication provides a regularized method 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.



On-the-fly error-checking in PDX is performed against channel data standards to prevent 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.

Syndication Prerequisites

Licenses

To access the PDX platform and functionality, the 'X.Adapter.ProductDataSyndication' license must be activated on your server, contact your Stibo Systems Account Manager for more information on enabling this license.

STEP Components

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 of which will require additional

configuration to enable. Once activated, STEP communication with PDX is handled via the PDX External API.

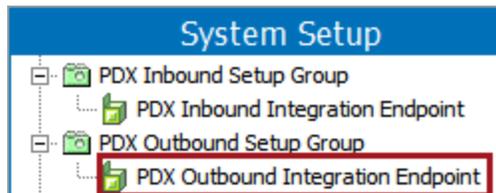
To install the required component and learn about other possible STEP requirements, contact your Stibo Systems representative.

Configuration

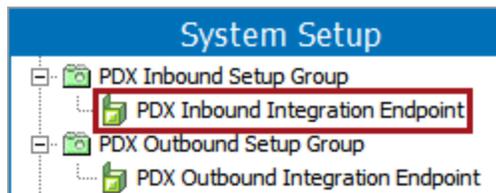
Important: The configuration steps described below pertain to PDX Syndication and not to PDX Onboarding. The PDX outbound and inbound integration endpoints are included in the PDX Syndication solution and require a license. For additional information on examples of PDX Onboarding, refer to a series of onboarding topics in the Accelerator for Retail solution enablement documentation, specifically **Suppliers via PDX**, **PIM from Suppliers**, and **Supplier Product Proposal**.

Some manual configuration setup is required for the following objects:

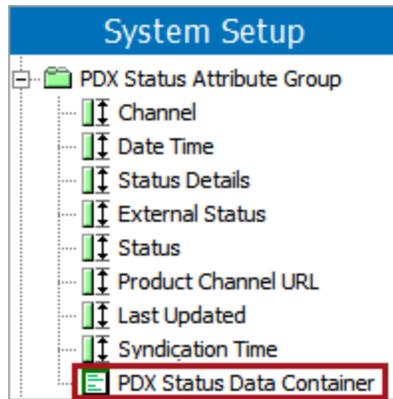
1. **PDX Outbound Integration Endpoint** - the OIEP sends master data to PDX and must be modified to work with your data. The Setting Up the PDX OIEP topic includes details for:
 - Manual required setup
 - Packaging hierarchy export setup
 - Optional product references metadata export setup



2. **PDX Inbound Integration Endpoint** - the IIEP receives the PDX channel status information. For details, refer to 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, refer to the **PDX Channel Status Monitoring** section of this documentation.



PDX Account & API User

To set up the integration, you'll need a PDX account and an API user, contact your Stibo Systems account manager for more information on setup.

You will also need a standard user. The standard user allows you to log into the PDX application and verify the data transfer. Your PDX Account Administrator from your organization 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 create the password. You will use these credentials to log into the PDX system. If your account does not yet have an appointed Account Administrator, contact your Stibo Systems Account manager to get one created.

PDX Syndication for Instrument

In Instrument users can view, add, and remove PDX channels for one or more products in a perspective table view. This will enable Category Managers to easily check if products are assigned to the desired PDX channels. If the products have not been assigned to the desired PDX channels, the Category Manager can add or remove PDX channels to ensure that products will be syndicated to the correct receivers via PDX. Furthermore, the PDX Channel solution in Instrument supports business processes to be executed before the actual syndication takes place. This is triggered via a Business Action that will run when channels are either added or removed.

To utilize PDX Channels in Instrument, users will need to implement the PDX Channels JavaScript Bind. For more information on JavaScript binds refer to [JavaScript Binds](#). For information on the PDX Channels bind, refer to [PDX Channels Bind](#) documentation.

For more information, refer to [PDX Syndication](#) documentation. For more information on Instrument, refer to the Instrument Documentation that is included when Instrument is enabled on a system. It is located on the Start Page under Resources.

Setting Up the PDX OIEP

The STEPXML-based OIEP produces cross-contextual STEPXML files that get enriched to make it self-contained. The final payload is then sent to PDX through the external API available, where it gets parsed into PDX-native JSON and imported.



The following topics describe how this integration is set up and configured to transfer the different types of product information that comprises a product record.

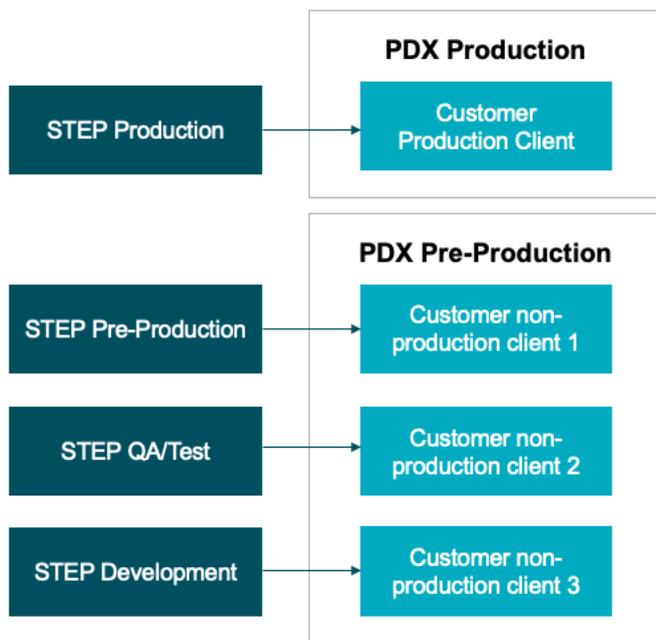
- Base Setup
- Handling Nested Data Structure
- Transferring Assets
- Transferring the Packaging Hierarchy
- Other OIEP Dependencies
- Validating the Configuration
- Known Limitations

Base Setup

The server-side setting of the `sharedconfig.properties` `PDX.Url` determines the PDX environments eligible in the OIEP configuration of the delivery method. Two PDX environments are relevant to this setting:

- PDX Production (<https://api.pdx.stibosystems.com>)
- PDX Pre-production (<https://api.pdx-preprod.stibosystems.com>)

General setup will include one account on the Production environment, which processes and syndicates live data, while multiple non-production accounts / clients may exist on the Pre-production environment. In the image below, STEP environments are in the column on the left, and those on the right represent PDX environments with PDX accounts.



Note: Changes to the properties file, outlined below, are implemented when the server is restarted.

The `sharedconfig.properties` on the STEP Production environment would specify the PDX Production environment (containing the customer’s production account / client) as the valid target of the PDX OIEP configured, as seen in the example below:

```
PDX.Url=1=https://api.pdx.stibosystems.com
```

To ensure product information maintained in all the contexts selected for syndication in the PDX OIEP setup can be transferred, the following property should be set to ‘false’:

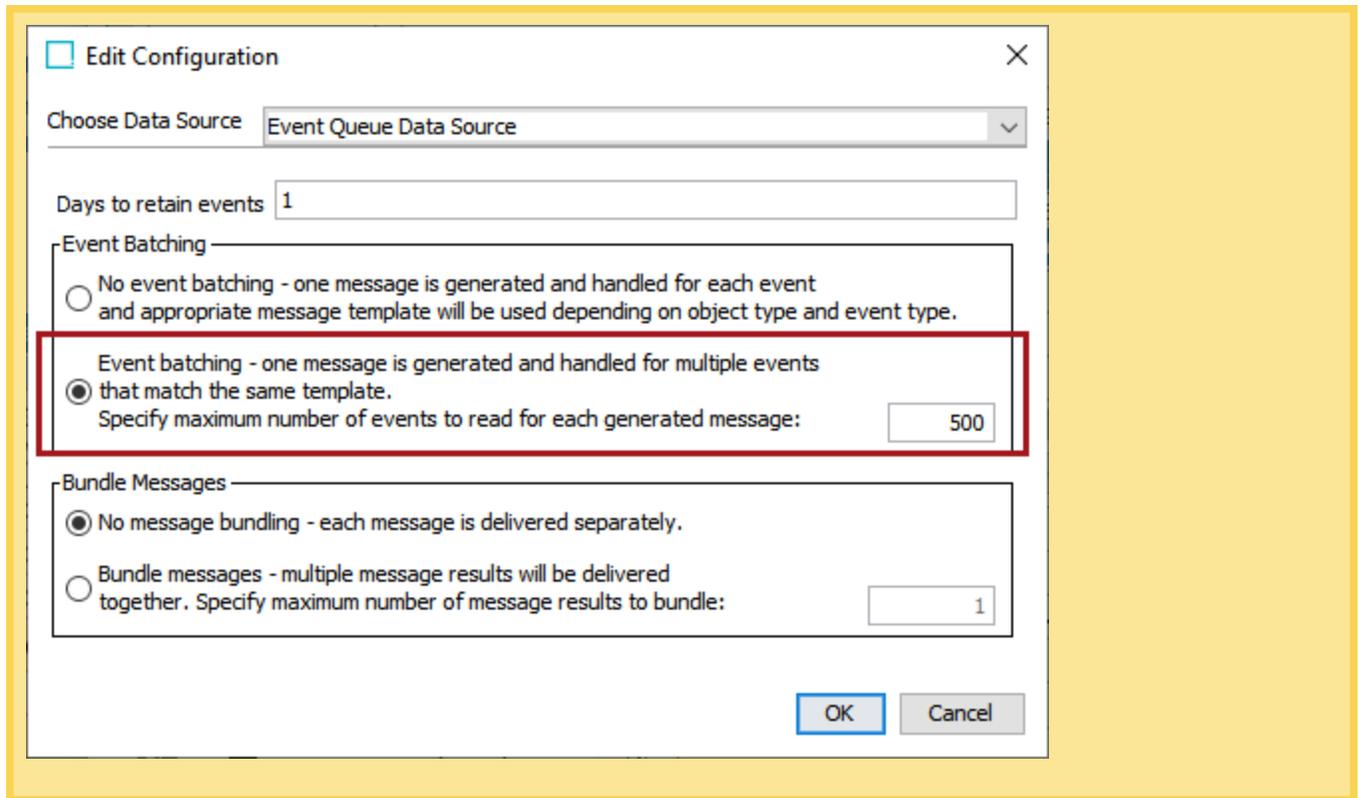
```
PDXDelivery2.LocaleChecking=false
```

Once these are set, continue setup by following the steps below:

1. **The STEPXML Type:** Ensure that the OIEP always sends cross context formatted STEPXML to PDX by setting the 'Context Mode' to 'Cross Context Format' as highlighted below:

PDX OIEP for English - Configuration				
Background Processes	Statistics	Error Log Excerpts	Log	Status
Outbound Integration Endpoint	Configuration	Event Triggering Definitions		
Configuration				
Process Engine	STEP Exporter			
Error reporter	Not Defined			
Schedule	Not scheduled			
Queue for endpoint	OutboundQueue			
Queue for endpoint processes	Out			
Transactional settings	Strict			
Maximum number of threads	1			
Maximum number of waiting processes	1			
Maximum number of old processes	100			
Maximum age of old processes	1w			
Context Mode	Cross Context Format			
Contexts	English			
Workspace	Approved			

Important: Ensure the event batching is set to a maximum of 500 to ensure optimal performance of the integration.



2. **The STEPXML Template:** The Format used in the PDX integration is Advanced STEPXML, and the template should always contain the following tags:

- ContextList
- CrossReferenceTypes
- UnitList
- AttributeList

The following is an example template:

```

1 | <STEP-ProductInformation
   | ResolveInlineRefs="true" FollowOverrideSubProducts="true">
2 | <ContextList ExportSize="Minimum"/>
3 | <CrossReferenceTypes ExportSize="Minimum"/>
4 | <UnitList ExportSize="Minimum"/>
5 | <AttributeList ExportSize="Minimum"/>
6 | <Products ExportSize="Minimum" FlattenHierarchy="false">

```

```

7 | <Product/>
8 | </Products>
9 | <Assets ExportSize="Referenced"/>
10| </STEP-ProductInformation>

```

3. **The Pre-processor:** The Pre-processor used should be the 'PDX Pre-Processor 2.'

It has a single setting called 'Only Changes.' If this is set to 'No,' then all products in the event queue will be sent. If it is set to 'Yes,' then only products with a changed revision will be sent.

4. **The Delivery Method:** When setting up the Delivery Method of the PDX OIEP, ensure that the 'Product Data Exchange 2' is chosen.

- Select the appropriate target PDX environment
- Enter the username and the API password of a PDX user associated with a relevant account on the targeted environment.

Note: If a HTTP Proxy has been configured, you're able to choose it in the "Proxy Config" dropdown list. For more information on how to configure a proxy, refer to the topic **HTTP Proxy Configurations** in the **Data Exchange** documentation.

- Specify which of the contexts included in the OIEP configuration is the 'default context.' The data in this context will be used to populate the default language layer in PDX.

Important: Channel assignment rules in PDX will be evaluated based on the values available in the default language layer.

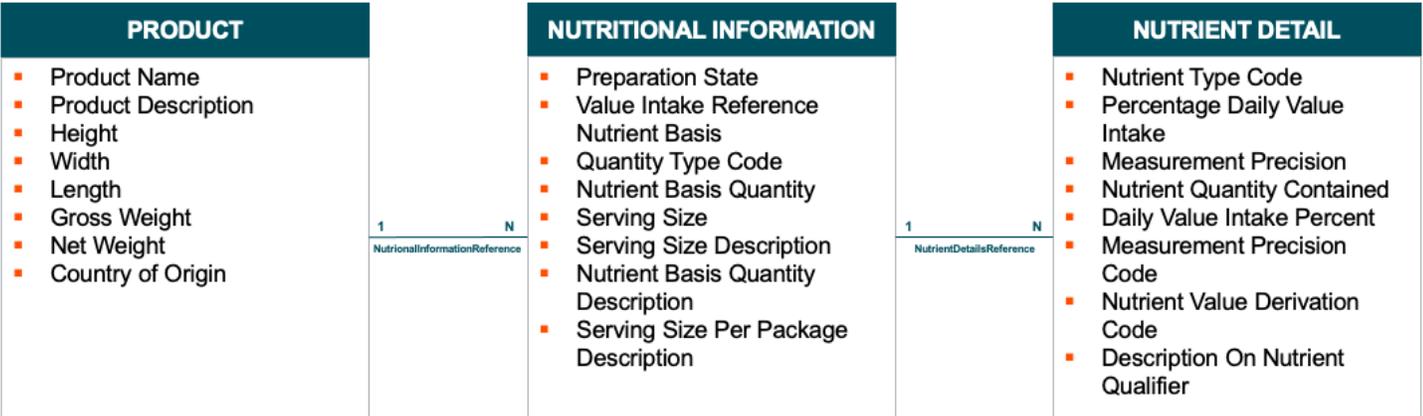
- Set the 'Upload only changed assets' setting to 'Yes,' if you only want to include referenced asset objects where either attribute values or the asset content has changed.

Handling Nested Data Structures

Nested data structures are sometimes required to capture pieces of product information. A good example is nutritional information describing the contents of a product at various scales. Consider the GDSN modeling of this data:

```
Nutritional Information: group{
  Preparation State: lov,
  Daily Value Intake Reference: string+,
  Nutrient Basis Quantity Type Code: lov,
  Nutrient Basis Quantity: decimal,
  Serving Size: decimal[],
  Serving Size Description: string+,
  Nutrient Detail: group{
    Nutrient Type Code*: string,
    Percentage of Daily Value Intake: decimal,
    Measurement Precision: lov,
    Nutrient Quantity Contained: decimal[],
    Daily Value Intake Percent Measurement Precision Code: lov,
    Nutrient Value Derivation Code: lov,
    Description On Nutrient Qualifier: string[],
    Nutrient Source: string+
  }[],
  Nutrient Basis Quantity Description: string+,
  Servings Per Package Description: string+
}[]
```

To model this data as accurately as possible, the STEP data model is currently required to have a referential nature (i.e. using references to other objects) as illustrated below:



Cinnamon Sprinkles Cereal rev.0.10 - References

Reference Type	Target
Primary Prod...	
Product Image	
Product References	
Reference Type	Target
Quantity	
Accessories	
Sell Side - Bu...	
Product To Classification Link	
Reference Type	Target
Front Page	
Web Link	
Ungrouped Product References	
Reference Type	Target
Nutrient Percentage Reference	Nutrient Quantity Reference
Nutritional Information	Nutritional Panel 1

Nutritional Panel 1 rev.0.4 - Product

Reference Type	Target
Nutrient 1	
Nutrient 2	
Nutrient 3	
Nutrient 4	
Nutrient 5	
Nutrient 6	

Index Words

Inherited From	Index

Publications

ID	Name

Linked Attributes from Product Hierarchy

Display Sequence	ID	Name	Attribute Groups

Linked Attributes from Classification Hierarchy

Ready

The referential structure needs to be included in the Advanced STEPXML Template of the PDX OIEP, by embedding the referenced data. Consider the example below:

```

1 <STEP-ProductInformation ResolveInlineRefs="true" FollowOverrideSubProducts="true">
2   <ContextList ExportSize="Minimum"/>
3   <CrossReferenceTypes ExportSize="Minimum"/>
4   <UnitList ExportSize="Minimum"/>
5   <AttributeList ExportSize="Minimum"/>

```

```

6      <Products ExportSize="Minimum" FlattenHierarchy="false">
7          <Product>
8              <Name/>
9              <Values/>
1             <ProductCrossReference
0             Type="NutritionalInformationReference" ExportSize="Minimum">
1                 <MetaData/>
1                 <Product Referenced="True" Embedded="True">
2                     <Values/>
1                     <ProductCrossReference
4                     Type="NutrientDetailsReference" ExportSize="Minimum">
1                         <MetaData/>
5                         <Product Referenced="True" Embedded="True">
6                             <Values/>
1                             </Product>
8                             </ProductCrossReference>
1                             </ProductCrossReference>
9                             </Product>
2                             </ProductCrossReference>
1                             </Product>
2                             </ProductCrossReference>
2                             </Product>
3                             </Products>
2                             <Assets ExportSize="Referenced"/>
4                             </STEP-ProductInformation>
5

```

Important: In the examples used, the referential structure is modeled using STEP Objects of the Product type tied together by Product cross-references, but Entity-based structures would be equally valid for transferring a nested data structure to PDX.

On the PDX side, the attribute values of each reference and its target object are consolidated to form a row of values in a Composite Attribute. PDX Composite attributes closely resembles multi-valued Data Containers in STEP, but they are able to contain other Composite Attributes, thus forming a 'tree' of data.

The screenshots illustrate the navigation path in the Master data interface:

- Product List:** A table with columns ID, NAME, and NUTRITIONAL INFORMATION. A row for 'SalesItem-105807 Cinnamon Sprinkles Cereal' is highlighted.
- Product Detail:** The 'Cinnamon Sprinkles Cereal' page showing 'ID: SalesItem-105807' and a breadcrumb path: 'Syndication / Nutritional Information'. A table below shows 'ANALYTICAL_AND_C...' with values 2, 22 Gram, 4, Per Serving, and Unprepared.
- Nutrient Details:** The 'Cinnamon Sprinkles Cereal' page showing 'ID: SalesItem-105807' and a breadcrumb path: 'Syndication / Nutritional Information / Nutrient Details'. A table below shows nutrient source details.

ANALYTICAL_AND_C...	DAILY INTAKE REFERENCE	SERVING SIZE	SERVINGS PER PACKAGE DESCRIPTION	SERVING SIZE DESCRIPTION	PREPARATION STATE
2	22 Gram	4	Per Serving	Unprepared	

NUTRIENT SOURCE	LABEL	NUTRIENT TYPE CODE	NUTRIENT QUANTITY CONTAINED	DAILY VALUE INTAKE PERCENT	MEASUREMENT PRECISION CODE
ANALYTICAL_AND_CAL...	1	ENER-	13	2	Approximately
ANALYTICAL_AND_CAL...	2	PRO-	8	16	Approximately
ANALYTICAL_AND_CAL...	3	CHO-	0	0	Approximately
ANALYTICAL_AND_CAL...	4	FATNLEA	0	0	Approximately
ANALYTICAL_AND_CAL...	5	VITA	2		Approximately
ANALYTICAL_AND_CAL...	6	VITK	4		Approximately

Managing Composite Attribute Names Shown in PDX

By default, composite attributes are titled according to the name of the given STEP reference type, or the ID of the reference type if no name is available. However, the recommendation is to create a meta data attribute for the reference types tying together the nested data structures that explicitly sets the ID of the Composite Attribute in PDX. This can be done through the following steps:

- 1. Creating the metadata attribute:** A new metadata attribute is needed to store the desired name of the composite attributes to be shown in PDX. Find an appropriate attribute group and create a new attribute within it. It should be a single-valued description attribute with a suitable name and no dimension dependencies, that can be used to store text.
- 2. Adding Reference Type Validity:** Go to the Object Types & Structures section in the System Setup tab, and expand the Basic Object Types subsection. Find the 'Reference-Type' object type, and add the newly created metadata attribute to the list of valid attributes.
- 3. Adding Metadata Values to the Reference Types:** The reference types constructing the nested data structure are then updated with the appropriate names for the composite attributes they will be converted into on the PDX side, by entering a value for the newly created metadata attribute.

For example, two references used to model nutritional information, i.e. 'Product to Nutritional Information' (NutritionalInformationReference) and 'Nutritional Information to Nutrient Details' (NutrientDetailsReference) will be given the values 'Nutritional Information' and 'Nutrient Details' respectively.

4. **Adding to the sharedconfig.properties file:** In order to inform PDX of which attribute should be used to determine the composite attribute name, the property 'PDSDelivery.CompositeAttributeID' must be added to the sharedconfig.properties file. This will have to be set to the STEP attribute ID of the newly created metadata attribute.

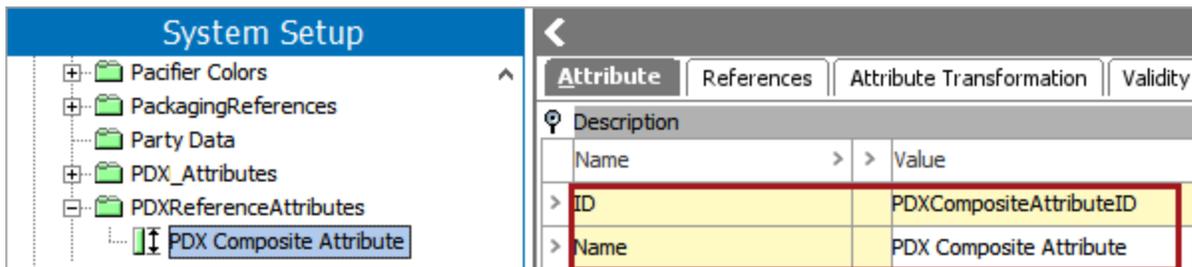
In this example case, the addition to the sharedconfig.properties file would be the following:

```
PDSDelivery.CompositeAttributeID=PDXCompositeAttribute
```

5. **Validate the Composite Attribute Names:** Lastly, a product with the nested data structures can be pushed to PDX through the OIEP, allowing validation of the composite attribute names (the column headers) which reflect the values added to the reference types in Step 3.

Example 1: The name of the attribute is 'PDX Composite Attribute,' the ID of the attribute is 'PDXCompositeAttribute,' and it has been created within an attribute group named 'PDX Reference Attributes.'

A value can be set for the created meta data attribute, allowing the user to change the composite attribute ID in PDX.



Example 2: The 'PDX Composite Attribute' attribute value is set, which will now be included as a description attribute within the 'NutritionalInformationReference' and 'NutrientDetailsReference' reference types, changing the IDs of the two composite attributes in PDX to 'Nutritional Information' and 'Nutrient Details' respectively.

Note: The 'PDX composite attribute ID' attribute is made valid on the reference type *object*, not on the reference type *link* between the source and target product.

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.'

System Setup

- User
- User Group
- Value Generator Configuration Type
- Web Service Endpoint
- WebUI Steppers Configuration Type
- Workspace
- XSLT Stylesheet
- Commercial Types
- Entity user-type root
- Event Queue Object Types
- Primary Product Classification
- Product-Overrides
- Publication group types
- Publication section types
- Publication types
- Setup Group type root
- Tags
- Units
- Users & Groups
- Reference Types
 - Product Reference Types
 - BicycleWheel
 - Case To Child
 - DC_Reference_Type
 - Package To Material
 - Pallet To Child
 - Product To Package
 - Product to Product Case**
 - Product to Product Power

Reference Type | Validity | Log

Description

Name	Value
ID	Product to Product Case
Name	Product to Product Case
Last edited by	2020-11-04 13:48:42.047 by USERK
Externally Maintained	No
Dimension Dependencies	
Allow multiple references	Yes
Mandatory	No
Inheritance	None
Attribute Completeness Score	123
Completeness Score	123
PDX Composite Attribute	abc ProductCase

In Attribute Groups

ID	Name
Add Attribute Group	

Valid Attributes

ID
CaseSupplier
Add 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.

✕ Bose Soundlink Micro (Black)

Summary Product attributes Packaging Product variants Digital assets

Ungrouped attributes

ProductCase *

AenIlosi|SKB|Gator|Penguin



ProductImage *

BTSImage2, BTCImage1



Transferring Assets

Assets can easily be included in the outbound data to PDX by adding the appropriate tags to the Advanced STEPXML template of the PDX OIEP. For more information, refer to **Advanced STEPXML Format** topic in the Data Format documentation.

The integration to PDX also allows transfer of context-specific assets, when the dimension dependency is managed on the reference type, refer to the **Dimension Dependency on a Reference Type** section of the **Dimension Dependent Reference and Link Types** topic for additional details.

If the assets are stored in an external DAM integrating with STEP, the following property can be added to the sharedconfig.properties file on the STEP application server, in order to disable the default behavior of STEP pushing asset binaries to PDX:

```
PDSDelivery.IncludeAssetContentsFromDAM=false
```

This is the preferred option in cases where an asset URL is stored on the relevant asset objects allowing PDX to automatically download the assets from the DAM. If this URL can be forwarded to the recipients downstream of PDX, it would significantly reduce the total amount of data transferred between the parties involved.

Transferring the Packaging Hierarchy

In order to transfer the packaging hierarchy associated with the product information, the Packaging Component Model has to be configured with the appropriate object types, reference types, and quantity metadata attributes valid for those object types. For more information on configuring the Packaging Component Model, refer to the **Configuring the Packaging Component in STEP Workbench** topic in the **Web User Interface** documentation.

Packaging - Component Model Configuration			
Component Model Configuration			
Name		Value	Description
> Packaging object types		Case	List of product object types that are used in the Packaging Component Model
		Display Shipper	
		Each	
		Inner Pack	
		Pallet	
> Quantity of the next lower level package		Package Reference Quantity	Description attribute containing the quantity of the next lower level packages
> Packaging reference types		Pallet to Child	Reference types that relates a packaging object to its child packaging object
		InnerPack to Child	
		Case to Child	
> Edit			

Note: The lowest level of the syndicated packaging hierarchy should be the 'Each' in order to avoid products being ineligible for submission to recipients downstream of PDX.

The relevant object types can then be added to the list of Object Event Types of the PDX OIEP, so updates to packages trigger the export of the change to PDX.

Example:

Conditions for output template

All object types

Specify

Browse Search

Each (Each) Search

Advanced

Name

Name
> Each ID = Each

Case

- Each
- Inner Pack
- Mixed Module
- Pallet
- Shrink Wrap

Event Types

Create Test

Modify

Delete

OK Cancel

The appropriate packaging references has to be added to the Advanced STEPXML Template of the PDX OIEP. Consider the below example:

```

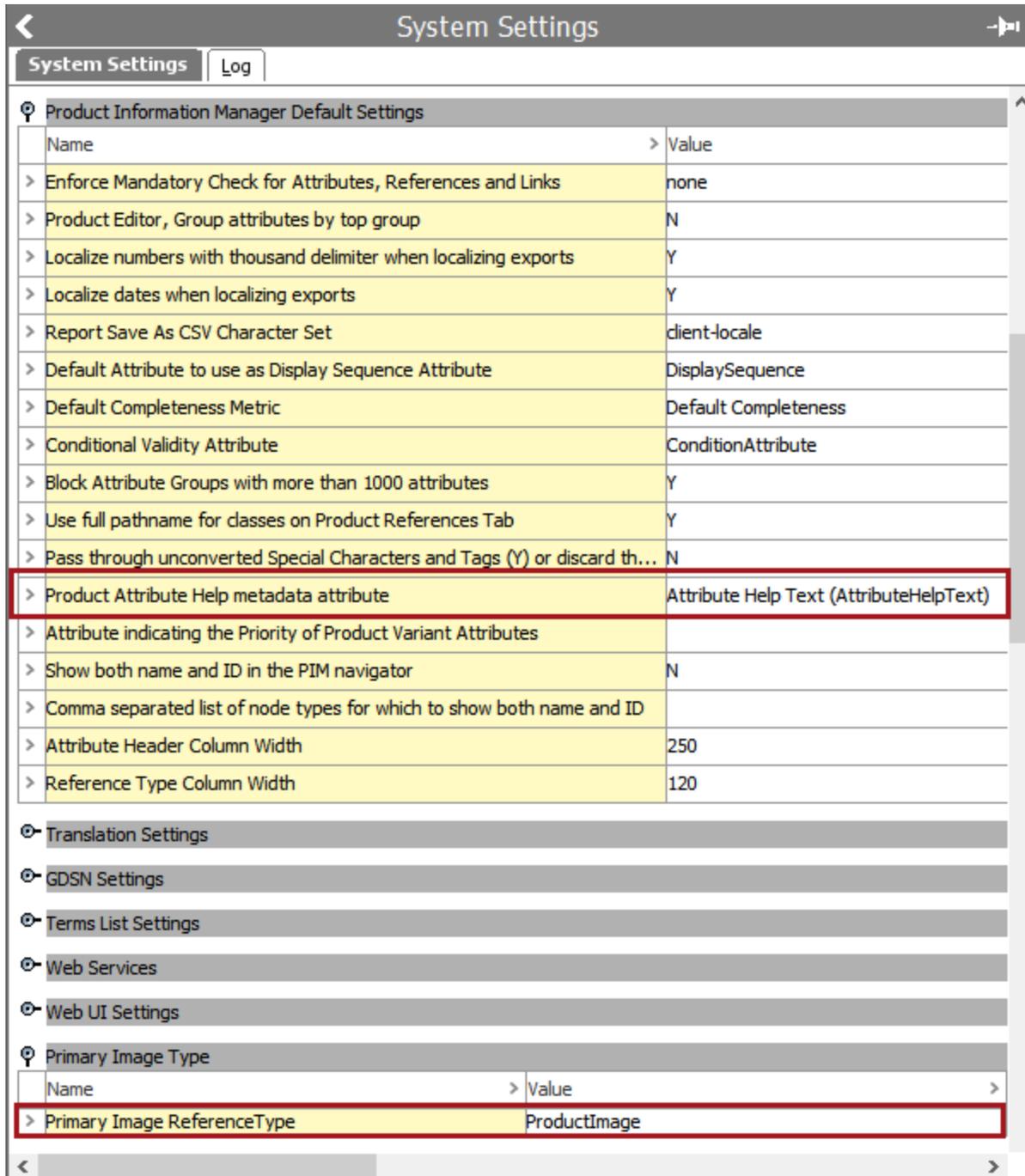
1 <STEP-ProductInformation ResolveInlineRefs="true" FollowOverrideSubProducts="true">
2 <ContextList ExportSize="Minimum"/>
3 <CrossReferenceTypes ExportSize="Minimum"/>
4 <UnitList ExportSize="Minimum"/>
5 <AttributeList ExportSize="Minimum"/>
6 <Products ExportSize="Minimum" FlattenHierarchy="false">
7 <Product>
8 <Values/>
9 <ProductCrossReference Type="Pallet_to_Child"/>
10 <ProductCrossReference Type="InnerPack_to_Child"/>
11 <ProductCrossReference Type="Case_to_Child"/>
12 </Product>
13 </Products>

```

```
14 | <Assets ExportSize="Referenced"/>  
15 | </STEP-ProductInformation>
```

Other OIEP Dependencies

The PDX OIEP automatically includes information of the Primary Product Image and the Attribute Help text retrieved from the baseline STEP configuration:



The screenshot shows the 'System Settings' interface. It features a 'System Settings' tab and a 'Log' button. The main content is a scrollable list of settings. Two settings are highlighted with red boxes:

- Product Attribute Help metadata attribute**: Value is 'Attribute Help Text (AttributeHelpText)'. This setting is highlighted in yellow.
- Primary Image ReferenceType**: Value is 'ProductImage'. This setting is highlighted in yellow.

Other settings visible include: Enforce Mandatory Check for Attributes, References and Links (none); Product Editor, Group attributes by top group (N); Localize numbers with thousand delimiter when localizing exports (Y); Localize dates when localizing exports (Y); Report Save As CSV Character Set (client-locale); Default Attribute to use as Display Sequence Attribute (DisplaySequence); Default Completeness Metric (Default Completeness); Conditional Validity Attribute (ConditionAttribute); Block Attribute Groups with more than 1000 attributes (Y); Use full pathname for classes on Product References Tab (Y); Pass through unconverted Special Characters and Tags (Y) or discard th... (N); Attribute indicating the Priority of Product Variant Attributes; Show both name and ID in the PIM navigator (N); Comma separated list of node types for which to show both name and ID; Attribute Header Column Width (250); Reference Type Column Width (120).

Below the main settings are sections for Translation Settings, GDSN Settings, Terms List Settings, Web Services, Web UI Settings, and Primary Image Type. The Primary Image Type section contains a table with the following data:

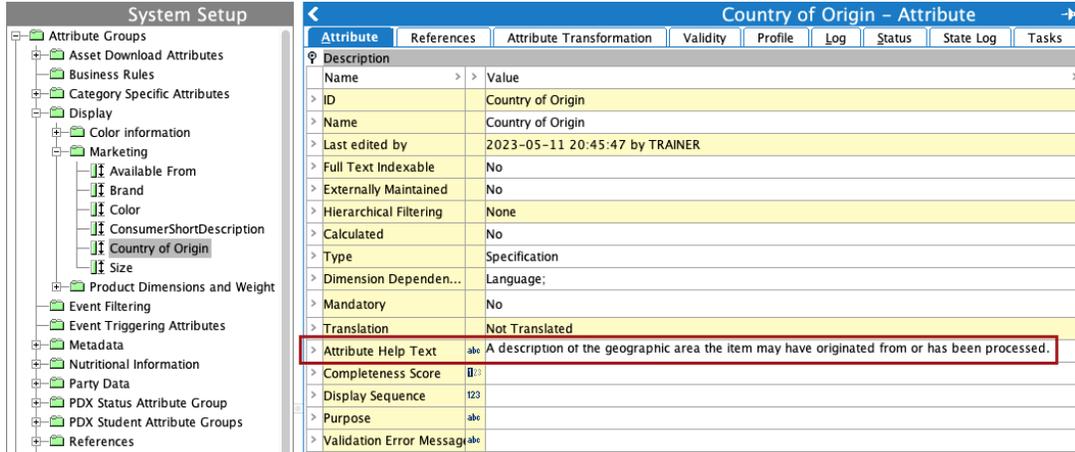
Name	Value
Primary Image ReferenceType	ProductImage

The integration to PDX will transfer available help text stored in the meta data attribute selected, in the example shown above, it is 'Attribute Help Text,' for the product attributes included in the export. However, only the

attribute help text in the Default Context will be imported to PDX. This will make the help text available when viewing the PDX Master Data. For more information, refer to the **Base Setup** topic in this documentation.

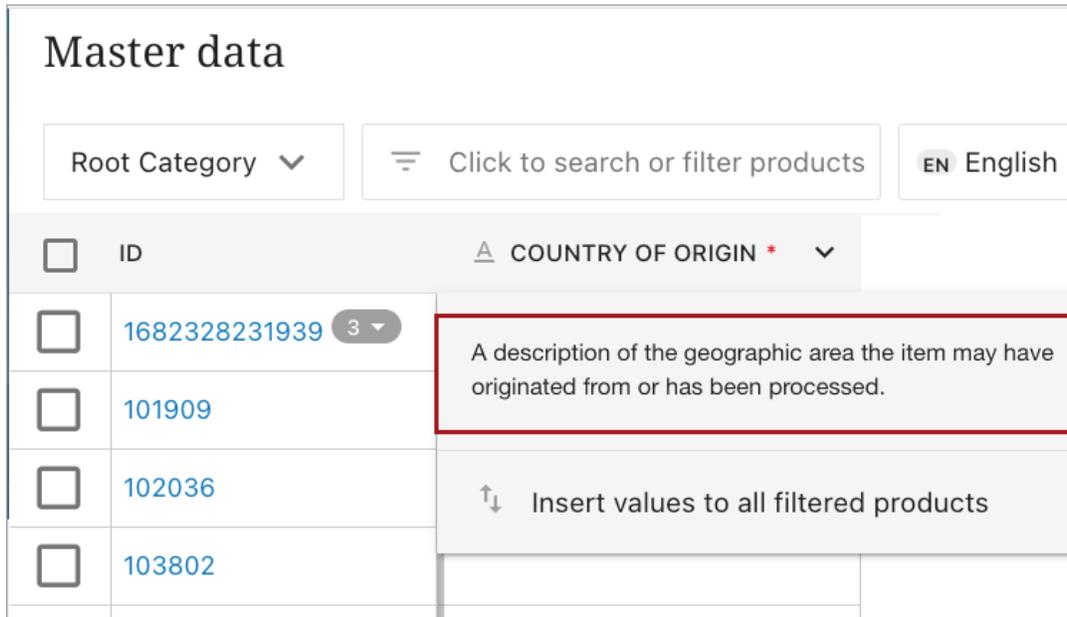
In the example below:

1. An attribute is shown with a meta data attribute storing help text.



The screenshot shows the 'System Setup' interface for the 'Country of Origin' attribute. The left sidebar lists various attribute groups, with 'Country of Origin' selected under 'Marketing'. The main window displays the attribute configuration table, where the 'Attribute Help Text' row is highlighted with a red box. The help text is: 'A description of the geographic area the item may have originated from or has been processed.'

The help text is then visible on the column headers in PDX Master data as seen in the two locations below:



The screenshot shows the 'Master data' interface. At the top, there are filters for 'Root Category', a search bar, and 'EN English'. Below, a table is displayed with columns 'ID' and 'COUNTRY OF ORIGIN *'. The 'COUNTRY OF ORIGIN *' column header is highlighted with a red box, containing the help text: 'A description of the geographic area the item may have originated from or has been processed.'

ID	COUNTRY OF ORIGIN *
<input type="checkbox"/> 1682328231939	A description of the geographic area the item may have originated from or has been processed.
<input type="checkbox"/> 101909	
<input type="checkbox"/> 102036	
<input type="checkbox"/> 103802	

Mapping to Nile.com

Create mappings from master data columns to Nile.com

Nile.com columns
Master data
count
Details

Nile.com columns	Master data
Product Title	Suggested
Production year	Other
Price	Country of Origin
Sellable in store	
Sellable online	
Front	

Channel data

[nile.com](#)

No description available

Master data

Country of Origin
(ID: Country of Origin)

A description of the geographic area the item may have originated from or has been processed.

Transformation results

Germany In

Your mappings
16

Size	Size	X
Sellable from	Available From	X
Battery Full Name	Battery	X
Rounded Height (cm)	Height	X
Depth (in)	Depth	X

Cancel
Done

- Similarly, the designated primary image reference type, 'ProductImage,' will determine what images shows up in the thumbnails of the PDX UI.

In the screenshot below, there is a product with multiple asset references.

Tree

- Primary Product Hierarchy
 - Products
 - Apparel
 - Audio Visual Equipment
 - Audio Visual Accessories
 - Portable Audio/Video
 - Televisions
 - SU10-LCD
 - SU1-LCD
 - SU2-LCD
 - SU3-LCD
 - SU4-LCD
 - SU4-P7000
 - SU4-P7000-55**
 - SU4-P7000-65
 - SU4-P7000-75
 - SU4-P7000-80
 - SU4-P7000 Supplier A

SU4-P7000-55 rev.0.9 – References

Product	Data Containers	References	Referenced By	Images & Documents
Commercial	Tables	Proof View	Status	State Log
Tasks				
Images References				
Reference Type	Target	Photographer		
> Primary Product Image	P_P7000			
	P_AC-XNPR746			
> Product Image	P_P7000 es			
	P_P7000 fr			
Product References				
Reference Type	Target	Quantity		
> Accessories	AE Bracket - max60			
	AE Bracket - max80			
> Sell Side - Buy Side	Series P7000 model 55			
Product To Classification Link				
Reference Type	Target	Front Page		
> Web Link				
Ungrouped Product References				

When viewed in the master data grid view in PDX, the asset references display under the corresponding columns.

Master data

Root Category Starts with EN English

ID	PRODUCTIMAGE *	PRIMARYPRODUCTIMAGE *
SalesItem-101326	100512, 100515, 100514	100516

If the Primary Image Reference Type is not defined in STEP, assets will still be transferred and available to recipients downstream of PDX.

Validating the Configuration

Before enabling the configured PDX OIEP, it might be worth validating the configuration and settings required for the integration to be successful. Consider the topics below.

STEP Application Properties

The sharedconfig.properties should contain

- Mandatory properties
- The declaration of the STEP attribute controlling the PDX composite attribute naming
- The setting to exclude the transfer of DAM-hosted binaries

Example:

```
#####
# PDX Settings
#####
PDX.Url=https://api.pdx-preprod.stibosystems.com
PDXDelivery2.LocaleChecking=false
PDSDelivery.CompositeAttributeID=PDXCompositeAttribute
PDSDelivery.IncludeAssetContentsFromDAM=false
```

OIEP Advanced STEPXML Template

The Advanced STEPXML Template of the PDX OIEP should contain the

- Mandatory tags
- Embedded reference targets for nested data structures
- References to relevant assets
- References forming the packaging hierarchy

Example:

```
1 <STEP-ProductInformation ResolveInlineRefs="true" FollowOverrideSubProducts="true">
2   <ContextList ExportSize="Minimum"/>
3   <CrossReferenceTypes ExportSize="Minimum"/>
4   <UnitList ExportSize="Minimum"/>
5   <AttributeList ExportSize="Minimum"/>
6   <Products ExportSize="Minimum" FlattenHierarchy="false">
```

```

7      <Product>
8          <Name/>
9          <Values/>
10         <ProductCrossReference Type="Product_NI" ExportSize="Minimum">
11             <MetaData/>
12             <Product Referenced="True" Embedded="True">
13                 <Values/>
14                 <ProductCrossReference Type="NI_ND" ExportSize="Minimum">
15                     <MetaData/>
16                     <Product Referenced="True" Embedded="True">
17                         <Values/>
18                     </Product>
19                 </ProductCrossReference>
20             </Product>
21         </ProductCrossReference>
22         <ProductCrossReference Type="Pallet_to_Child"/>
23         <ProductCrossReference Type="InnerPack_to_Child"/>
24         <ProductCrossReference Type="Case_to_Child"/>
25         <AssetCrossReference Type="ProductImage"/>
26     </Product>
27 </Products>
28 <Assets ExportSize="Referenced"/>
29 </STEP-ProductInformation>

```

The Preamble

In order for PDX to correctly interpret and parse the inbound STEPXML, STEP needs to enrich the OIEP-generated payload with critical information about the data model implemented. This enrichment is referred to as the 'preamble' and is stored in a separate file in the BGP folder.

Remote site: /shared/workarea/background-processarea/Outbound/BGP_70192556

- Outbound
 - BGP_70184749
 - BGP_70184953
 - BGP_70184955
 - BGP_70184957
 - BGP_70192448
 - BGP_70192548
 - BGP_70192549
 - BGP_70192550
 - BGP_70192556
 - OutboundPoller

Filename	Filesize	Filetype	Last modified	Permissions
..				
17972455-17972455.xml	28570	XML document	05/03/23 15:19:09	-rw-r--r--
preamble.xml	970	XML document	05/03/23 15:19:11	-rw-r--r--
recorder-BGP-Out-BGP_70192556...	1313	XML document	05/03/23 15:19:08	-rw-r--r--

The content of the preamble is a mix of data model and component configurations, as well as server-side settings.

Example:

```

1  <AuxiliaryTypes>
2    <PackagingTypes QuantityAttribute="Quantity">
3      <ReferenceType ID="Pallet_to_Child"/>
4      <ReferenceType ID="Case_to_Child"/>
5      <ReferenceType ID="InnerPack_to_Child"/>
6      <UserType ID="PA"/>
7      <UserType ID="CA"/>
8      <UserType ID="EA"/>
9    </PackagingTypes>
10 </AuxiliaryTypes>
11
12 <ConfigurationProperties>
13
14   <AttributeHelpAttributeID>AttributeHelpText</AttributeHelpAttributeID>
15
16   <DefaultContext>EN_GB</DefaultContext>
17 </ConfigurationProperties>

```

```

1      <PDSDelivery.AssetUploadedAttributeID>PDS Syndication
4      Time</PDSDelivery.AssetUploadedAttributeID>
1
5      <PDSDelivery.CategoryAttributeID>PDSCategory</PDSDelivery.CategoryAttributeID>
1
6      <PDSDelivery.CompositeAttributeID>
      PDXCompositeAttribute</PDSDelivery.CompositeAttributeID>
1
7      <PDS_Status_Attribute_Group>PDS Status Attribute Group</PDS_Status_Attribute_Group>
1
8      <PrimaryImageType>ProductImage</PrimaryImageType>
1
9      </ConfigurationProperties>

```

Important: It is recommended to review the preamble of initial exports to PDX to ensure that all elements are present / populated, as missing declarations often results in incomplete data transfers.

Known Limitations

Below is a list of known limitations for the PDX STEPXML based outbound integration endpoint.

Dimension dependent Asset content

It is not currently possible to transfer context-specific images by making the asset content dimension dependent as described in the **Asset Dimension Dependencies** subsection of the **Maintaining Assets** topic in the **Getting Started** documentation. However, a similar result can be achieved by making the asset reference type dimension dependent instead.

Transfer of Product Families

It is not currently possible to transfer objects modeling a product family to PDX. However, if the ID of the family object is included on the object representing the core product record, PDX will allow a virtual grouping of products based on the value of such an attribute.

LOV Value IDs

It is not currently possible to transfer the value IDs of attributes using the LOV Base Type validation. Only the raw value of the attribute is included in the exports to PDX.

Product-to-Product References

It is not currently possible to transfer information about a products relationship to other products (e.g. accessories, bundles it's included in, cross-sell opportunities, etc.) modeled through references in STEP with identical source and target object types.

Data containers

It is not currently possible to transfer data stored in STEP Data Containers. This means that the data either needs to be modeled as flat attributes on the Object itself, or as a referential structure. Refer to the **Handling Nested Data Structures** topic in this documentation for more information.

Dimension dependent STEP Names

It is currently not possible to transfer language-specific attribute, reference, and unit names to PDX. The STEP Names available in the 'default context' are the only source of information used in the data transfer. For more information, refer to the Base Setup topic in this documentation.

Classification references

It is currently not possible to transfer references from products to classifications. Product-relevant information stored in the classification structure, should be made available on the Object representing the core product record through Business Actions or Calculated Attributes.

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

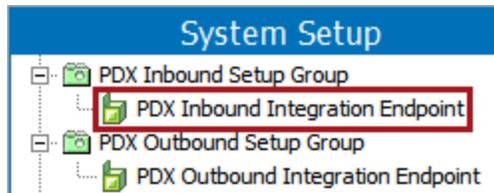
Note: This documentation section assumes that users have a basic understanding of data containers. For more information on data containers, refer to the **Data Containers** section of the **System Setup** documentation.

PDX Inbound Integration Endpoint Configuration

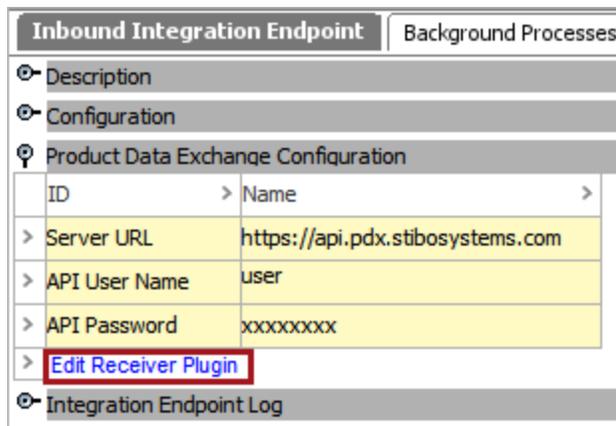
The PDX Inbound Integration Endpoint (IIEP) is used to receive the PDX status information for integrated products.

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 the data model.

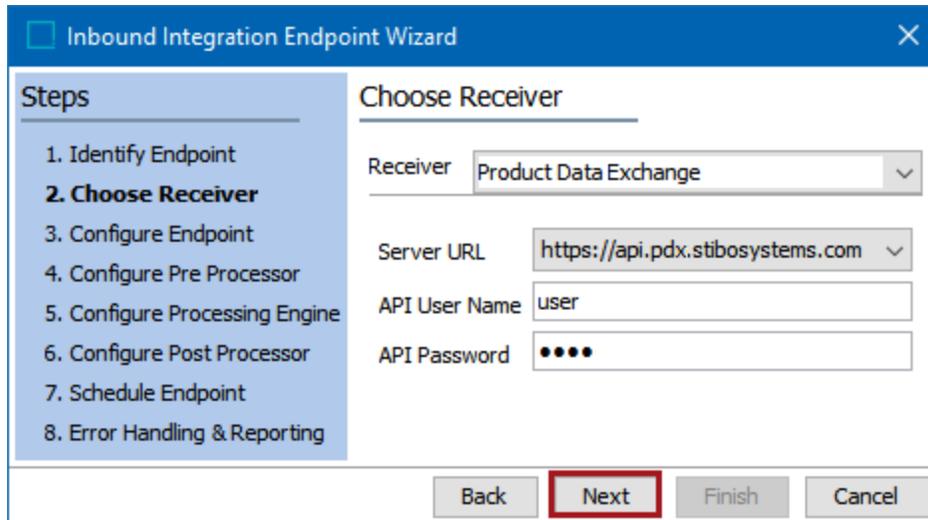
1. In the System Setup > PDX Inbound Setup Group node, open the **PDX Inbound Integration Endpoint**.



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.



Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
- 2. Choose Receiver**
3. Configure Endpoint
4. Configure Pre Processor
5. Configure Processing Engine
6. Configure Post Processor
7. Schedule Endpoint
8. Error Handling & Reporting

Choose Receiver

Receiver: Product Data Exchange

Server URL: https://api.pdx.stibosystems.com

API User Name: user

API Password: ●●●●

Buttons: Back, **Next**, Finish, Cancel

For details, refer to 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, refer to 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.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
- 3. Configure Endpoint**
4. Configure Pre Processor
5. Configure Processing Engine
6. Configure Post Processor
7. Schedule Endpoint
8. Error Handling & Reporting

Configure Endpoint

Processing

Processing Engine: Product Data Exchange

Transactional settings: None

Context

Workspace: Main

Context: English US

Queue Settings

Queue for endpoint: InboundQueue

Queue for endpoint processes: In

Maximum number of waiting processes: 1

Maximum number of old processes: 100

Maximum age of old processes: 1w

Number of messages per background process: 1

Back Next Finish Cancel

6. On the Schedule Endpoint step, set the desired schedule and click the **Finish** button.

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
3. Configure Endpoint
4. Configure PreProcessor
5. Configure Processing Engine
 - 5.1. Configuration
- 6. Configure PostProcessor**
- 7. Schedule Endpoint**
8. Error Handling & Reporting

Schedule Endpoint

Start

Never
 Every
 Weekly
 Monthly
 Later

1 minutes

Start every minute

Back Next Finish Cancel

Updates made in the previous steps are displayed in the 'Configuration' and 'Product Data Exchange Configuration' flippers.

< PDX Inbound Integration Endpoint rev.0.2 - Inbound Integr

Inbound Integration Endpoint | Background Processes | Statistics | Error Log Exc

☰ Description

🔑 Configuration

Pre-Processor	No pre-processing
Process Engine	Product Data Exchange
Post-Processor	No post-processing
Error Handling & Reporting	Not Defined
Schedule	Start every minute ...
Queue for endpoint	InboundQueue
Queue for endpoint processes	In
Transactional settings	None
Maximum number of old processes	100
Maximum age of old processes	1 week
Number of messages per background process	1
Contexts	defaultcontext
Workspace	Main

> [Edit Configuration](#)

🔑 Product Data Exchange Configuration

ID	Name
> Server URL	https://api.pdx.stibosystems.com
> API User Name	user
> API Password	xxxxxxxx

> [Edit Receiver Plugin](#)

☰ Integration Endpoint Log

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 found on the Data Containers tab of the product editor.

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

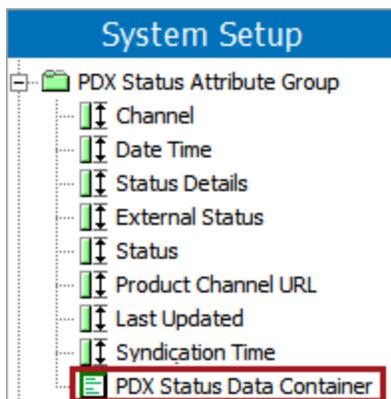
This topic provides information on:

- 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.

< PDS Status Data Container - Validity

Data Container Type | References | **Validity** | Log

Valid for Entity Types

All

Valid for Product Types

All

Archived Products

Business Unit Root

Buy Side

Case

Customer Specific Data

Display Shipper

Each

FormID

- 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.'

Data Container Type | References | Validity | Log

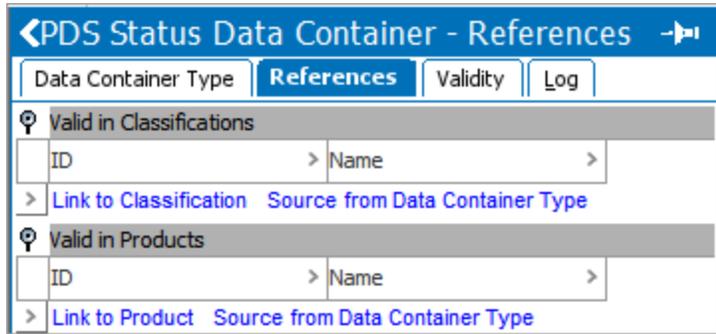
Description

Name	Value
ID	PDX Status Data Container
Name	PDX Status Data Container
Last edited by	2021-11-15 15:14:48 by STEPSYS
ID Pattern	
Allow multiple data containers	Yes
Inheritance	None
Restriction	<div style="border: 1px solid red; padding: 2px;"> None None Validity restricted to hierarchies </div>
Mandatory	None

In Attribute Groups

ID	Name
PDX Status Attribute Group	PDX Status Attribute Group
Add Attribute Group	

- On the 'References' tab, select the required classification or product hierarchies.



For more information regarding configuring data containers, refer to the **Setting Up Data Container Types in Workbench** topic in the **Data Containers** section of the **System Setup** 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, refer to 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, refer to 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 found 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, refer to the Setting Up the PDX OIEP topic in the Data Integration documentation.

For more information on the PDX Inbound Integration Endpoint, refer to 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, refer to 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. Contact 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, refer to the **Design Mode Basics** topic in the **Web UI Getting Started** section of the **Web User Interfaces** 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, refer to the **Node Editor Component** topic in the **Using a Web UI** documentation.

For more information on adding a component to a Node Details screen, refer to 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, refer to the **Localization** topic within the **Administration Portal** documentation.

Properties (edited)

Configuration Web UI style

Channel Status Save Close New... Delete Rename Save as...

PDX Channel Status Properties [go to parent](#)

Component Description
Provides a tabular read-only view of PDX channel statuses for a single product. The component features an option to navigate to the product in PDX in the context of a given PDX channel.

* Data Container Type PDX Status Data Container

Title i18n.stibo.PDXChannelStatus.title

Attributes

- PDS Channel
- PDS Status Details
- PDS Date Time
- PDS External Status
- PDS Status

Add... Remove Up Down

Child Components

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.

Product Details		Channel Status	Reference Editor		
Channel	Date Time	External Status ▼	Status	Status Details	
Added to PDX		2018-11-15 16:00:55			
1WorldSync US		2018-11-15 16:28:15	GDSN Status: Registration sent	Submitted	
1WorldSync PreProd		2018-11-30 16:52:11	GDSN Status: CIC Rejected	Returned	CIC state: Rejected

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, refer to 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. Contact 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.), refer to 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 view the entire text in a pop-up:

	Status Details
CIC state: Rejected CIC001: Wrong publication type;	
CIC state: Rejected CIC001: Wrong publication type; was new should have been initial item load. Addition Description: Item already exists. Cannot resend as New Corrective action: INITIAL_ITEM_LOAD:Received as new item, please resend as an initial item load CIC013: Retailer issue Corrective action: CONTACT_TRADING_PARTNER:Call or email data recipient"	

- 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.

	Status Details
CIC state: Rejected CIC001: Wrong publication type; was new should have been initial item load. Addition Description: Item already exists. Cannot resend as New Corrective action: INITIAL_ITEM_LOAD:Received as new item, please resend as an initial item load CIC013: Retailer issue Corrective action: CONTACT_TRADING_PARTNER:Call or email data recipient"	

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.

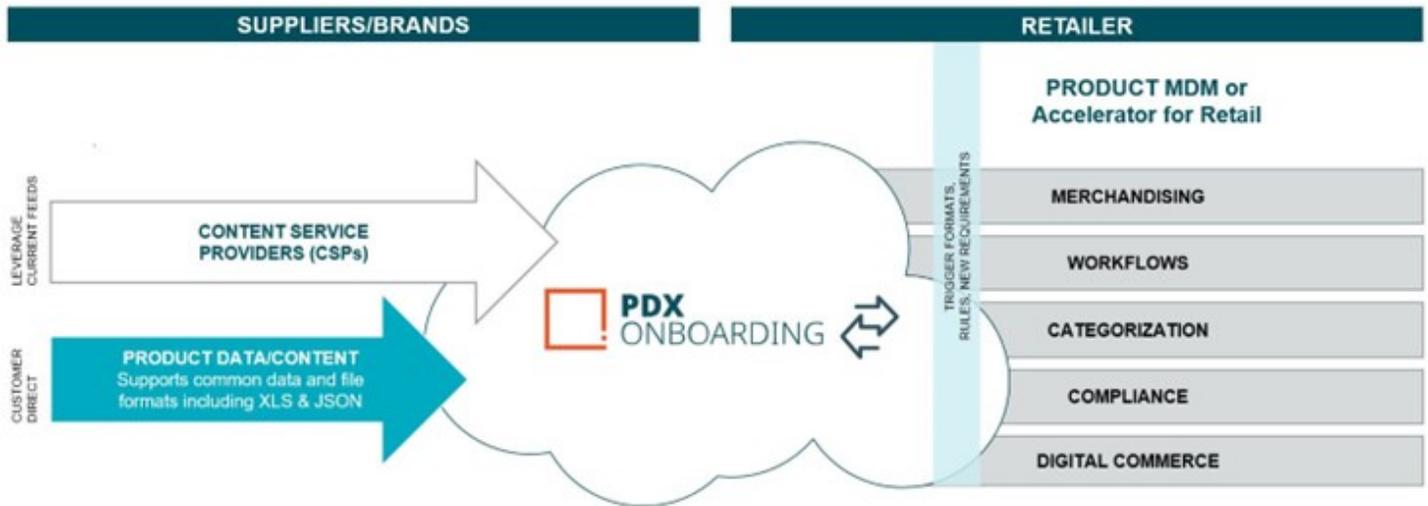
Rejected: Item will not be stored in the recipient's system. These items will not be sent anymore.

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.

PDX Onboarding

PDX Onboarding enables retailers to quickly acquire high-quality product data from suppliers, manufacturers, and brands. From product onboarding to product acceptance, the PDX Onboarding solution automates communication between retailers and their suppliers regarding the retailer’s data standards and requirements.



The PDX Onboarding solution connects with the STEP product MDM solution to enable manufacturers, suppliers, and brands to receive requirements from the retailer, such as new SKUs, new logistics requirements, and updates to regulatory data.

Two-way communication is central to how Product MDM and PDX Onboarding work together to support customers. This 'always on' communication provides manufacturers, suppliers, and brands with full visibility into the retailer’s product data standard requirements. All changes made by the retailer are maintained within—and made accessible via—the product MDM solution, and the ‘Accepted’ / ‘Rejected’ status information relevant to the incoming products automatically flows between the two systems.

Initial setup for the PDX Onboarding channel can be done using one of two approaches:

1. The pre-configured Accelerator for Retail solution
2. The PDX Onboarding Channel Configurator which allows a channel to be created from an existing STEP solution using a wizard-type setup tool within STEP

Both approaches are described in this documentation. For additional information about which approach is optimal for your business requirements, contact your Stibo Systems account manager.

Controlling the PDX Channel for Product Onboarding

Customers who use both PDX and the STEP Accelerator for Retail solution to send product information back and forth via a channel as part of their product onboarding workflow must be able to configure:

- proper display of relevant data in the PDX interface
- optimal channel behavior
- effective product data validations

Many elements of how data in the PDX UI displays, is validated, and how the channel performs are determined by configuration decisions made in STEP.

This topic and the topics nested beneath contain information about how the PDX Schema, a digital snapshot of PDX-relevant STEP objects that is imported into PDX, can be controlled in Accelerator for Retail, how an Accelerator for Retail implementation can be updated to account for changes to the PDX Data Model, and how STEP can be configured to ensure retailers and suppliers have the tools they need when sending products via PDX.

Updating the data standard

To enable suppliers to onboard products from PDX into a retailer's Accelerator for Retail STEP implementation, all elements that make up the Accelerator for Retail data model must be replicated in PDX. These elements include, for example, the supplier-facing product hierarchy, attributes, attribute groups, lists of values (LOVs), and asset references. The process of synchronizing the supplier-facing data model in STEP with the corresponding data model in PDX is called a Data Standard update. This update allows suppliers to validate their product information against the retailer's. Beyond the Data Standard update, additional processes and methods deployed by both STEP and PDX are in place to ensure synchronicity between the two systems.

Data standard update

When a change is made to the Accelerator for Retail data model that affects the PDX channel, the channel-specific Data Standard in PDX must be updated to account for the change.

The following actions are examples of changes made in STEP that require a Data Standard update in PDX:

- Adding, modifying, or deleting supplier-facing attributes or LOVs.
- Adding, modifying, or deleting nodes in the supplier-facing product hierarchy.
- Linking or unlinking attributes within the supplier-facing product hierarchy.
- Adding or deleting supplier-facing asset reference types.

Data Standard updates can either be made by request to your PDX team, or as part of a scheduled process if schema changes are likely to be more frequent.

The Data Standard update is implemented by the PDX team. In the update process, a PDX Channel Schema is exported from STEP using the 'PDX: Schema Export' export configuration. This export, generated from the 'Approved' workspace in STEP, then converts the PDX Channel Schema into the channel-specific Data Standard in PDX, so the changes made to the product hierarchy, relevant attributes, etc., are properly reflected in PDX.

Data Standard elements that can be changed in STEP

When suppliers bring products into Accelerator for Retail via PDX Onboarding, the retailer's PDX channel should reflect the relevant part of the data model from Accelerator for Retail. This symmetry of data between PDX and Accelerator for Retail is maintained to allow the suppliers to categorize products, map attributes, and transform values to meet the retailer's requirements.

Each section that follows describes either an element of the PDX Data Standard that users can configure or change in STEP via the Accelerator for Retail implementation, or an aspect of the PDX-to-Accelerator connection that is helpful for users to know.

Controlling the PDX Channel for Product Onboarding - Product attributes

Accelerator for Retail enables admins to apply precise controls over how PDX users may interact with the attributes that make up product data. Described below are some of the methods that may be used to:

- Enable product families and product variants
- Ensure specific attributes display in PDX
- Set the display sequence of attributes in PDX
- 'Lock down' certain attributes for PDX users

Product families and variants

Products in PDX can be grouped into product families that may include product variants. When those product variants are sent from PDX to Accelerator for Retail, they arrive ungrouped but tagged with a unique identifying attribute called a 'PDX Master Product ID'. The value for this attribute, which is assigned to each variant product, enables users to recategorize the variants into those same product families in STEP. This recategorization work is done later in the onboarding process.

At least one attribute must be defined as a variant attribute for a category to enable product family grouping (and the associated validations) in this category in PDX.

In the screenshot below, the red-boxed color swatches icon indicates that the selected category supports product families. Further, on the selected object in the table beneath the icon, a gray bubble with the number '6' displays with a small white arrow. This indicates the selection is a collapsed product family. The attributes on this family can only be edited at this level: if the family is expanded, the displaying attributes cannot be edited.

ACC4R DEV For System Demo

Beauty/Personal Care/Hygiene ...  Select Account

<input type="checkbox"/>	ID	STATUS	NAME
<input type="checkbox"/>	00885612543356	 Submitted	test 1
<input type="checkbox"/>	00885612543363	Ready	test 2
<input type="checkbox"/>	5XT-25688240	Submitted	Honey scented lotion

For additional information on product variants as they relate to Accelerator for Retail, refer to the **Accelerator for Retail PIM Product Variants** topic.

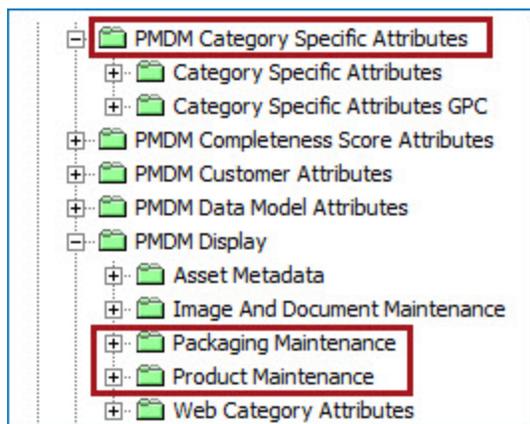
Product-Related Attributes

The display of attributes in PDX is configured in both PDX and Accelerator for Retail. What follows is a description of the various kinds of product attributes found in PDX, and where these attributes can be managed, reviewed, and maintained.

Attribute Groups

All attributes and attribute groups intended for display on products and packaging objects in PDX are stored in the Accelerator for Retail solution's STEP system. These attributes and attribute groups are contained in one of three attribute nodes found on the System Setup tab in the STEP Workbench. As shown in the screenshot below, those nodes are

- Product Maintenance
- Packaging Maintenance
- Category Specific Attributes



All attributes and attribute groups contained in these three folders are visible to and editable by suppliers in PDX, provided the attributes are valid for any of the following object types:

- External Source Record
- Case
- Pack
- Pallet

Further, these attributes and attribute groups must also be linked to two 'user privilege-controlling' attribute groups: 'Supplier, View' and 'Supplier, Modify'. These attributes are located below the 'PMDM User Privileges' attribute group. (Attributes linked to either 'PDX: Hide From Supplier' or 'PDX: Locked after Proposal' are exceptions and are described later in this topic.) If the customer determines that specific attributes contained in the 'Product Maintenance', 'Packaging Maintenance', or 'Category Specific Attributes' attribute groups should not be shown to suppliers in PDX, the attributes must also be linked to the 'PDX: Hide From supplier' attribute group. Linking an attribute to the 'PDX: Hide From supplier' attribute group will hide that attribute from users in PDX, even if the attribute is linked to one of these three display-enabling attribute groups.

Family Attributes

Family attributes in PDX enable a structure through which an attribute on a product on the family level, defined as one level above the sellable product level, can have a specific value that filters down to the products nested beneath. This helps ensure data consistency throughout the products in that family.

When an attribute is defined as being a 'family' attribute, validation is automatically applied that prevents users from submitting products for which the family attribute values differ from other products in that family. By restricting maintenance of family attributes in this way, suppliers are assured that all products contained within the family will have the same value for a given family attribute. For instance, if the family of products is a model of shoe, the value for the family attribute 'Style Number' should be the same for all model variants. The value for this family attribute is set at the family level and is applied automatically to all variants.

When a user submits a single product contained inside of a family, the system automatically submits all products in that family. In this way, all products in the family are validated and submitted to STEP simultaneously.

To configure an attribute as being a 'family attribute', the attribute 'PDX: Family Attribute' must be set to 'Yes' on the attribute link between the product category and the desired attribute.

For more information on setting up family attributes in Accelerator for Retail, refer to the **Family Attributes** section of the **Variant – Setup** topic in the Accelerator for Retail documentation.

Variant Attributes

When product families are used, it is possible to define certain attributes as being variant attributes.

An attribute defined as a variant attribute in STEP will also be treated as a variant attribute in PDX. Once defined, the variant attribute has validation applied that ensures each variant product that uses the attribute is unique. In other words, any combination of values assigned to a given variant product must be unique within the family. As an example, if a variant product has two variant attributes called 'Size' and 'Color', only a single variant product within the product family can have the value 'Medium' for the attribute 'Size', and the value 'Blue' for the attribute 'Color'. This prevents two shoe variants from being 'Medium' and 'Blue', which would make them, in terms of the defined product family, the same product.

To define an attribute as a variant attribute for a specific section of the product hierarchy in PDX, the metadata attribute 'Product Variant Priority', which is found on the product family node under the 'References' tab, can have integer or non-integer values added, but PDX will only register integers and ignore non-integer values. Which integer is added does not have an effect in PDX; the specific number assigned determines the order in which the attributes display in STEP.

For more information on assigning values to the 'Product Variant Priority' metadata attribute, refer to the **Setting Up Product Variants** topic in the **System Setup** documentation.

Once an attribute has been defined as a 'variant' attribute:

- The category is enabled for family product grouping



- The 'Group into family' and 'Ungroup family' actions become available to the user



- A special icon displays beside the category name, indicating families can be created within the category



- The view of product families in the 'Grid view' can be expanded or collapsed, as needed. In the screenshot below, the red-boxed product family '1' shows a collapsed product family, while red-boxed product family '2' shows an expanded product family.

<input type="checkbox"/>	ID	STATUS	NAME	INTERNAL
<input type="checkbox"/>	1677593779425	Multiple	Honey scented lotion	1
<input type="checkbox"/>	8Z5-16827670	Ready	Flower scented moisturizer	2
<input type="checkbox"/>	BCP-22651868	Ready	After hours deluxe perfume	
<input type="checkbox"/>	6RH-78377177	Rejected		

For more information on setting up variant attributes in Accelerator for Retail, refer to the **Variant Attributes** section of the **Variant – Setup** topic in the **Accelerator for Retail** documentation.

Display sequence of attributes and attribute groups

The order in which attributes and attribute groups display on products in PDX can be changed in the Accelerator for Retail solution in STEP.

By making the 'Display Sequence' (ID: PMDM.AT.DisplaySequence) metadata attribute valid on attribute and attribute group definitions, the display order of attributes and attribute groups in PDX can be managed via STEP. Described below is how the 'Display Sequence' attribute can be used to define display sequence.

Attribute groups -- When the 'Display Sequence' metadata attribute is applied to an attribute group, the display order of that attribute group and others can be controlled in the corresponding channel in PDX.

Attributes -- When applied to a regular attribute, the 'Display Sequence' metadata attribute can be used to control the ordering of attributes within an attribute group in PDX.

Data container attributes -- For attributes in a data container (called composites in PDX), the order of the columns is also determined by the 'Display Sequence' metadata attribute.

Note: Attributes and attribute groups with an assigned value for the 'Display Sequence' metadata attribute are ordered from lowest to highest, with those assigned a low 'Display Sequence' value displaying higher in a list, and those with a higher value displaying lower. Based on their 'Display Sequence' metadata attribute, data container attribute columns with a lower number will display to the left, and those with a higher number to the right.

Locked-Down Attributes

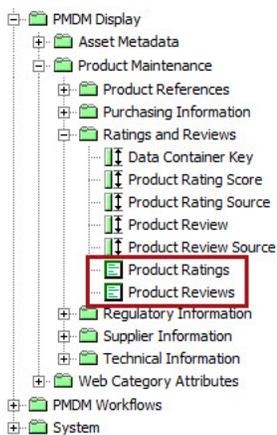
In certain instances, users may require that specific attributes be made un-editable, or locked, following the onboarding of a product into the Accelerator for Retail solution. Locking these attributes prevents suppliers from editing the relevant attribute values after that supplier has submitted the product to the retailer. For example, if a supplier has submitted a product with a specific GTIN, the retailer would want to lock down that attribute so as to disallow the supplier to re-submit the same product with a new GTIN. To lock attributes in this way, admins must link them to the 'PDX: Locked After Proposal' attribute group.

Controlling the PDX Channel for Product Onboarding - Data Containers

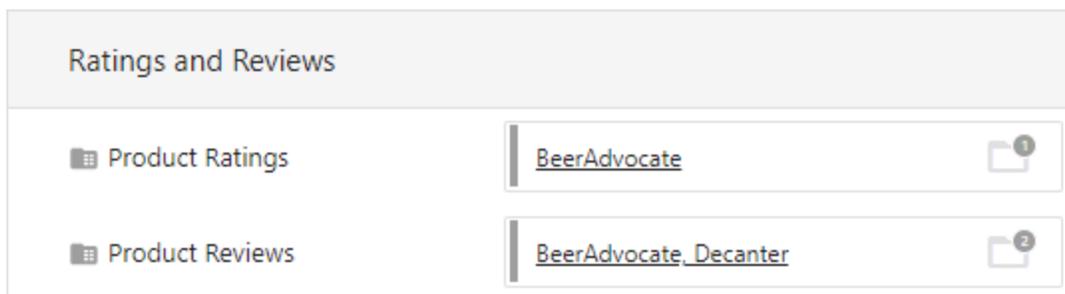
Data container types in STEP are available in PDX as single- or multi-valued “composites,” which is the term used in PDX to describe data container-analogous objects. To display a composite in PDX, the data container type and the supplier-relevant attributes within the data container must be linked, in STEP, to one of the three PDX-relevant attribute groups:

- Product Maintenance
- Packaging Maintenance
- Category Specific Attributes

To illustrate the different ways data containers and composites display in their respective systems (STEP and PDX), this section will outline, via screenshots, how two sample data containers, 'Product Ratings' and 'Product Reviews', will display. In the screenshot below are the two sample data containers, both of which are delivered with Accelerator for Retail packages.



If a data container is linked to any of the three PDX-relevant attribute groups Product Maintenance, Packaging Maintenance, and Category Specific Attributes, and is valid for either the 'External Source Record', 'Case', 'Pack', or 'Pallet' object types, the data container will display on a product in PDX as a composite, as shown in the screenshot below:



When a PDX user clicks on the 'Product Reviews' composite's details, the data container attributes and their values display in PDX as shown in the screenshot below:

<input type="checkbox"/>	☰ PRODUCT REVIEW SOURCE	▲ PRODUCT REVIEW	▲ DATA CONTAINER KEY
<input type="checkbox"/>	BeerAdvocate	The beer is a very clear c...	Key_001
<input type="checkbox"/>	Decanter	With the caramel and sw...	Key_002

For reference, the same data containers, when linked to a product submitted to Accelerator for Retail in STEP, display as shown in the screenshot below:

Product Reviews

	Product Review
BeerAdvocate	The beer is a very clear copper color with just a slight tinge of orange. The aroma has light caramel notes and a moderate golden raisin fruity character. Hints of cherry show up in the aroma as the beer warms.
Decanter	With the caramel and sweetness, this beer would pair nicely with a simple garlic and black pepper spiced braised butterflied chicken breast and a side of asparagus.

Number of items: 2

Product Ratings

	Product Rating Score
BeerAdvocate	85

Number of items: 1

To ensure proper display of data containers as composites in PDX, it is helpful to know some additional aspects of the functionality beyond what is described above. What follows is additional information relevant to admins working to configure data containers in STEP to achieve the desired results in PDX:

- Only data containers and data container attributes linked to one of the three PDX-facing attribute nodes—Product Maintenance, Packaging Maintenance, and Category Specific Attributes—will be available in PDX.
- Data container types that need to be available in PDX must have an ID Pattern with the auto-generated data container ID defined.
- Both single-valued and multi-valued data containers are supported.
- PDX composites support the ability to embed a composite within a composite, creating a two-level composite. STEP’s comparable data container functionality does not support (nor necessitate) a multi-level structure. Because of this difference, only one-level composites are supported by this solution.

- Text added to the metadata attribute on the data container type called 'Description' can display in PDX on the associated composite for suppliers.

Data containers and data container attributes linked to the 'PDX: Mandatory for Submit' attribute group will be mandatory in PDX. The mandatory designation in PDX means:

- the attribute will display with a black asterisk in a detail view, and a red asterisk in a grid view
- products cannot be submitted until a value is provided for the attribute
- If users attempt to submit products without a value for one or more mandatory attributes, an error message will display, listing which attributes are missing values

For more information on configuring attributes in STEP to be mandatory in PDX, refer to the 'Mandatory Attributes' section later in this topic.

Data containers linked to the "PDX: Locked after Proposal" attribute group will not be editable in PDX after the initial submission, even if the product is in an editable state.

- There is no support for category-specific data containers. Because PDX does not support STEP's 'Validity restricted to hierarchies' setting on the 'Restriction' attribute on the data container type, this setting must be set to 'None' to ensure data container information is properly displayed in PDX.
- Data containers and data container attributes can be sorted in PDX using the 'Display Sequence' (ID: PMDM.AT.DisplaySequence) metadata attribute on data container types and attribute definitions.

For more information on the usage of the 'Display Sequence' attribute, refer to the **Display Sequence Attribute** topic in the **System Setup** documentation.

- If a data container type does not have a data container key defined, all data container instances of that data container type (that are also included in one of the three PDX-facing attribute nodes) will be overwritten by the data container instances included in the submission. For example, if data container 'Product Reviews' did not have a data container key configured, and 'Product Reviews' was held in one of the three PDX-facing attribute nodes, all instances of the 'Product Reviews' data container on the product with the ID '1234' will be overwritten each time the '1234' product is submitted from PDX.

For more information on data container keys, refer to the **Data Container Keys** topic in the **Attributes** section of the **System Setup** documentation.

- If a data container key is defined for a data container type, existing data container instances will be updated if composites with the same key are submitted from PDX. A validation rule in PDX ensures data container keys are unique within a multi-valued composite.

For example, shown in the screenshot below are two data containers of the type 'Product Review'. They are linked to a product. One has a 'Data Container Key' of '001' and the other '002'.

Product Reviews								
>	ID	>	Data Container Key	>	Product Review	>	Product Review Source	>
>	DC-8705456		001		Best Champagne 2022		Decanter	
>	DC-8705457		002		You have to try this...!		James Suckling	
>	Add Data Container							

If the product is re-submitted to STEP from PDX with a data container instance of '001' and another of '003', the '001' data container will be updated in STEP with any changed values. The '002' data container will be deleted and a new data container with a new data container key will be created and populated with the values from '003'.

Data containers are supported in Maintenance. For additional information about the Maintenance process, review the Maintenance section of this topic.

Controlling the PDX Channel for Product Onboarding - Mandatory Attributes

Information related to two methods of designating attributes as mandatory is described below: Mandatory for Submit and Conditionally Mandatory.

Mandatory for Submit

To designate attributes in PDX as mandatory, meaning attributes on a product that must have a value added to enable submission of the product to PDX, users must link the specified attributes to the 'PDX: Mandatory for Submit' attribute group.

Category-specific attributes will display as mandatory in PDX by setting the value of the 'PDX: Mandatory for Submit' metadata attribute to 'Yes' on the attribute link in STEP.

Setting the 'Mandatory' metadata attribute on the relevant attribute itself will make the attribute mandatory in STEP but will not make the attribute mandatory in PDX.

Mandatory attributes configured with a condition, referred to as conditionally mandatory attributes, are mandatory unless the configured condition is not met. When the condition is not met, the attributes in question do not require values to proceed. To apply a conditionally mandatory attribute, add the 'PDX: Conditionally Mandatory' attribute (STEP ID: PMDM.AT.ConditionAttribute) as metadata on the Product Attribute link. (For more information on Product Attribute links, refer to the **Product Attribute Link Type** topic in the **System Setup** documentation.)

For example, the 'PDX: Conditionally Mandatory' attribute (shown in the screenshot below) specifies that 'Hazmat', a mandatory attribute, will only be considered mandatory if the 'HasLead' attribute has a value of 'Yes'. This value, 'HasLead = Yes', uses the syntax [AttributeID] = [Value1] to enable this use case.

Linked Attributes from Product Hierarchy				
Display Sequence	ID	Name	PDX: Conditionally Mandatory	
> - All -	∨ - All -	∨ - Equals Hazmat -	∨ - All -	∨
>	Hazmat	Hazmat	HasLead = Yes	
>	Link to Attribute			

If more than one value can meet the condition, all acceptable values can be listed with a semicolon delimiter, as in '[AttributeID] = [Value1];[Value2]'.

To ensure the conditionally mandatory attribute is correctly applied, follow the rules described below when writing the condition text:

- The attribute upon which the condition is based (in the above example, it is 'HasLead') must not have spaces in its ID.
- The attribute upon which the condition is based must not have a colon (':') in its ID.

- If the 'Condition Attribute' value contains a semicolon (';'), it must be escaped with a backslash ('\').
- If the 'Condition Attribute' value contains a backslash ('\'), it must be escaped with another backslash ('\\').

Considerations for conditionally mandatory attributes

The ability to make attributes conditionally mandatory can, potentially, create problems if certain considerations are not taken into account. Listed below are the primary considerations admins must keep in mind when configuring conditionally mandatory attributes.

- If your implementation of Accelerator for Retail provides the ability to make attributes conditionally mandatory using either the conditionally mandatory functionality described above, or the Validation Engine (described below), be sure to use one method or the other consistently. It is recommended that, of the two options, the Validation Engine is used as this is a more powerful tool that supports a wider range of related use cases. Further, conditionally mandatory rules will be deprecated for new Accelerator for Retail installations.
- Apply caution when configuring conditionally mandatory attributes. It is possible to create conflicting rules resulting in a product that cannot be submitted regardless of which values are populated.

Note: If conditional expressions use attributes that use Lists of Values (LOVs) that use value IDs, the conditional expression will evaluate the value ID. For example, an attribute, 'AttributeA', uses a 'Yes' / 'No' LOV, and that LOV uses value IDs where 1=Yes and 2=No. In this instance, the conditional expression must be written to include the value ID rather than the value itself. In this example, if the attribute should only be available when the value of AttributeA is 'Yes', the conditional expression should be written as 'AttributeA = 1' rather than 'AttributeA = Yes'.

Controlling the PDX Channel for Product Onboarding - Validation Engine Attributes

Accelerator for Retail offers the ability to configure advanced category-specific data validations in the PDX channel via a 'configuration language' referred to in PDX as the Validation Engine.

The Validation Engine is a tool that can be used to build advanced 'product data validations' without requiring admins to write code. Instead, an expression language is used to create the validations.

As with conditionally mandatory attributes (described above), the Validation Engine is controlled using a metadata attribute on the 'PDX: Validation Engine' (STEP ID: PDX: Validation Engine) Product Attribute link. In this way, advanced category-specific data validations can be run on attributes. The 'PDX: Validation Engine' attribute is multi-valued and uses the 'Regular Expression' validation base type.

Configuration details

Every product data validation created in the 'Validation Engine' consists of two or three parts:

1. Conditions
2. Error Message
3. Rule ID (optional)

Product data validations to create validations in PDX are written in the following format, with each element separated by a 'colon' delimiter. Below is a screenshot of the validation format with each element contained within a red box, each specified by a lower-case letter. Below the screenshot is a description of each element:

<Condition>:error:<Error Message>:ruleid:<Display Name of Product data validation >

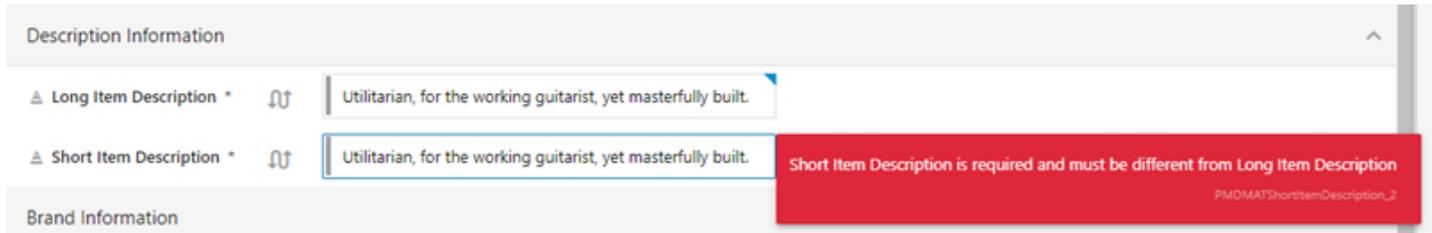


- a. The logic condition
- b. The keyword that indicates that what follows is the error message
- c. The error message
- d. Keyword that indicates that what follows is the name of the rule
- e. The name of the rule

Note: The error 'occurs' on the attribute if the condition is met.

To ensure each product data validation is unique in the system, all generated product data validations will have an ID with an incrementing number appended to it (e.g., 'PMDMATShortitemDescription_2' with

'PMDMATShortitemDescription' being the product data validation's ID and the '_2' the incrementing number appended to the ID).



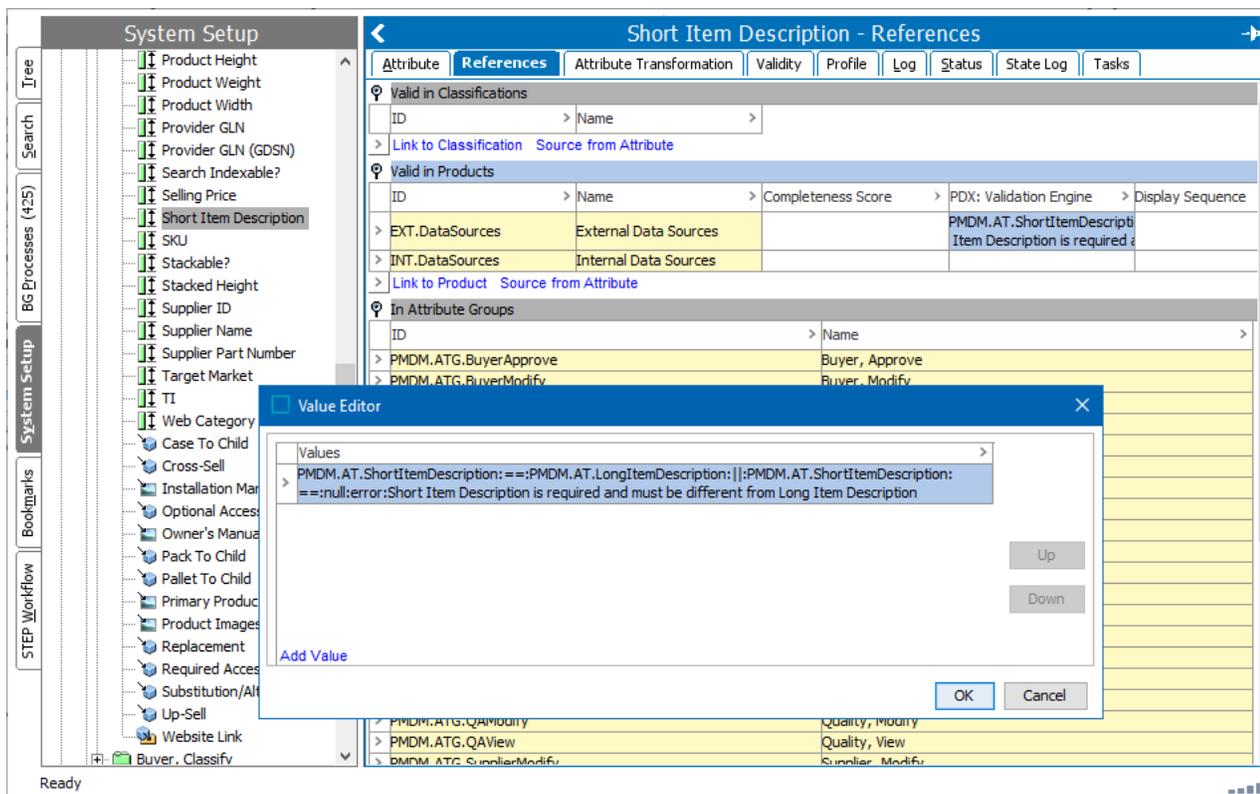
If no 'ruleid:' is provided in the product data validation, the ID will use a simplified version of the attribute (e.g., 'PMDM.AT.IsBaseUnit' becomes 'PMDMATIsBaseUnit' with an incrementing number appended to it).



Validation Example

To illustrate how a product data validation is written, a product data validation example, written in the STEP Workbench, is shown in the screenshot below and described in detail in this section.

In this example, the product data validation is written and applied to the 'Short Item Description' attribute.



When properly written, this product data validation presents an error to the PDX user when the 'Short Item Description' attribute either has no value, or the value for 'Short Item Description' is equal to the value for 'Long Item Description', meaning the values for both are the same.

Below is the example product data validation as it must be written to accomplish the validation task:

```
PMDM.AT.ShortItemDescription==:PMDM.AT.LongItemDescription:||:PMDM.AT.ShortItem
Description==:null:error:ShortItem Description is required and must be different
from Long Item Description
```

In this example, the product data validation is made up of the following elements:

- The condition -- This part of the product data validation specifies the data and which aspect of the data is being validated.

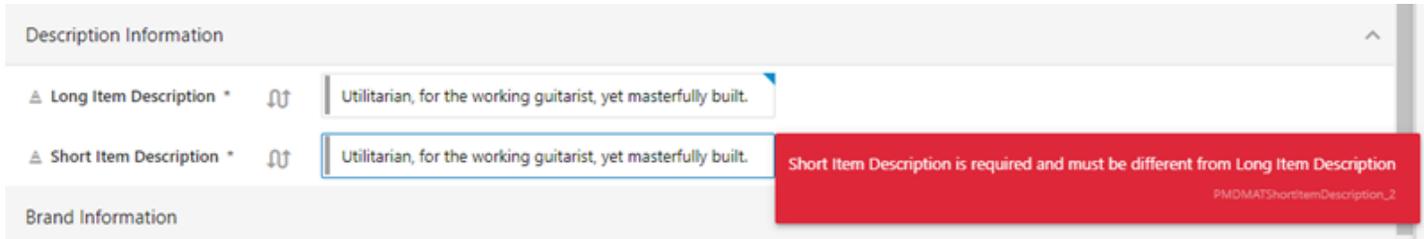
```
(<Condition>=
PMDM.AT.ShortItemDescription==:PMDM.AT.LongItemDescription:||:PMDM.AT.ShortItemD
escription==:null:
```

- The error message -- If the conditions defined in the product data validation are not met, the text defined in this element are presented to the user.

```
<Error Message>= Short Item Description is required and must be different from
Long Item Description
```

- Product data validation ID -- In this example, no product data validation ID is set (the ':ruleid:' element). As described previously, the PDX system will generate an ID for the rule that is a version of the validated attribute with an incrementing number appended to the product data validation to ensure the ID is unique in the system.

The error shown in PDX will display as shown in the screenshot below:



Product data validation elements

A product data validation, as described in the example above, is built using simple logical operators, attribute IDs, numbers, and text strings. These elements can be combined to produce advanced validations. In the table below, each element is listed and includes a description of the element and an example of how that element can be written into a product data validation.

Elements	Example	Description
STEP Attribute ID	PMDM.AT.LongItemDescription	Refers to a specific attribute value.
Fixed numbers or text strings	20 'Y' 'unsaturated fat'	Refers to a fixed value. Note: When comparing with LOV values, the 'ValueID' should be used. Literals should be enclosed in single quotes (').
Comparison	< ==	Simple comparisons can be done to compare numeric values and text strings. Only values from single-valued attributes are supported. The full list includes: < -- Less than > -- Greater than =< -- Less than or equal to >= -- Greater than or equal to == -- Equal to = -- Not equal to
Null		The constant null is used to indicate 'no value' and can be used to check if an attribute has no value. Only values from single-valued attributes are supported.
Logical operators	(Logical operators can be used to group and combine conditions into more complex conditions. The full list includes: (-- Start grouping) -- End grouping

		<p> -- Logical 'OR'</p> <p>&& -- Logical 'AND'</p>
Mathematical operators	<p>+</p> <p>-</p>	<p>The mathematical operators can be used to combine numeric values and produce a calculated value.</p> <p>Only values from single-valued attributes are supported.</p> <p>The full list includes:</p> <p>+ -- Additional</p> <p>- -- Subtraction</p> <p>* -- Multiplication</p>
contains		<p>The 'contains' method can be used to check if a value (numeric or textual) includes another value. Multivalued and single-valued attributes are supported.</p>
notContains		<p>The 'notContains' method can be used to check if a value (numeric or textual) does not include another value. Multi-valued and single-valued attributes are supported.</p>
isEmpty		<p>The 'isEmpty' method can be used to check if an attribute has no values. Multi-valued and single-valued attributes are supported.</p>
isNotEmpty		<p>The 'isNotEmpty' method can be used to check if an attribute has values. Multi-valued and single-valued attributes are supported.</p>

To further illustrate how product data validations can be applied to ensure data supplied in PDX meets the configured requirements, listed below are a series of product data validation examples covering a variety of use cases:

- PMDM.AT.CountryOfOrigin:==:'BS':&&:PMDM.AT.BrandName:==:null:error:If Country of Origin is equal to BAHAMAS, then Brand Name is required

- In this example, 'BS' is the ValueID of the value 'BAHAMAS' in the list of values (LOV) for the 'Country of Origin' attribute.
- (:PERCENTAGE_1+:PERCENTAGE_2+:PERCENTAGE_3):>:100:error:The total value of Percentage 1, Percentage 2 and Percentage 3 cannot be more than 100:ruleid:100 Percentage rule
- SHOELACE_MATERIAL:==:'NOT_APPLICABLE':&&:(ACCESSORIES_MULTY:contains:LACES:|:ACCESSORIES_MULTY:contains:SHOELACE:):error:'Not Applicable' is NOT an acceptable value for Shoelace Material when Lace is selected for attribute Accessories List.
- ALL_FEATURES_MULTY:contains:'ADDITIONAL_BURNERS':&&:isEmpty:ADDITIONAL_BURNER_TYPE:error:The Additional Burner Type is required when All Features contains Additional Burners
- POWERED_MULTY:notContains:"BATTERY_POWERED":&&:isNotEmpty:BATTERY_TYPE_MULTY:error:The list of Battery types must be empty, when product is not powered by batteries

When writing product data validations, it is important to note the following considerations and limitations:

- Comparisons, such as '=' and 'contains' are case-sensitive.
- The method 'isEmpty' can be either prepended or appended to the relevant [text] for the same result. This also applies to the function 'isNotEmpty'. For example, the notation 'isEmpty:ADDITIONAL_BURNER_TYPE:' and ':ADDITIONAL_BURNER_TYPE:isEmpty' can be used interchangeably.
- The 'Validation Engine' does not support product data validations that validate data containers or data container attributes.
- Most operations and comparisons do not support attributes with units.
- Product data validations can only be added and / or defined on attribute links to specification attributes.
- Product data validations cannot be written to include assets.
- When writing product data validations, it is important to note that the colon character (':') is a reserved character and should be used only to separate elements in the product data validation. If the character is used in an attribute ID or in an error message, for instance, this may cause the product data validation to error.

Using expressions correctly

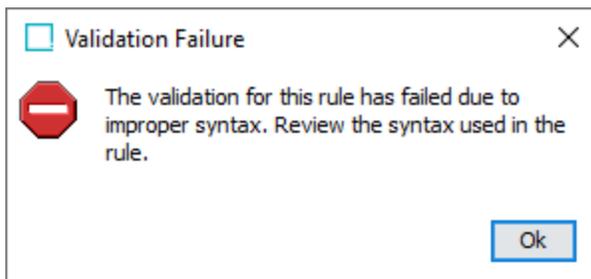
As mentioned previously, the 'PDX: Validation Engine' attribute uses the 'Regular Expression' validation base type. Regular expressions prevent product data validation writers from committing the most frequent notational errors when writing product data validations, but cannot address certain kinds of human error, like misspelled attribute names, for instance.

Listed in the table below are rules that must be followed:

Description	Invalid examples	Valid examples
“:error:” must be present and have non-empty message	ATTRIB_A:<=:ATTRIB_B:error: ATTRIB_A:<=:ATTRIB_B	ATTRIB_A:<=:ATTRIB_B:error:A should be greater than B
At least one truth valued operator must be present (i.e. + or - is not enough) and attribute names must be separated by operators or parentheses	ATTRIB_A:error:Some message (:ATTRIB_A:ATTRIB_B):error:Some message (:ATTRIB_A+:ATTRIB_B):error:Some message	(:ATTRIB_A+:ATTRIB_B:=:1):error:Some message
“(“ is matched by “)”	(ATTRIB_A:<=:ATTRIB_B:error:Some message	(ATTRIB_A:<=:ATTRIB_B):error:Some message
if “:ruleid:” is present, then it has non-empty content	ATTRIB_A:<=:ATTRIB_B:error:Some message:ruleid:	ATTRIB_A:<=:ATTRIB_B:error:Some message:ruleid:Rule 1
:ruleid can only contain alphanumeric characters and -_#:	width*:height>:1000:error:too large:ruleid:Unsanitized\$ID	width*:height>:1000:error:too large:ruleid:Unsanitized_ID

Some of the rules above will be caught by the regular expression check.

When product data validations fail, an error message displays that reads, ‘The validation for this rule has failed due to improper syntax. Review the syntax used in the rule.’



Controlling the PDX Channel for Product Onboarding - Assets

Accelerator for Retail supports multiple methods of configuring access to assets in PDX, and how the systems can validate assets via automation.

Asset References

Accelerator for Retail users can configure which assets display in PDX by updating the desired asset reference type definitions for those assets. On the asset reference type definition for the PDX-relevant asset reference types, users can configure the 'PDX: Relevance' attribute. This attribute helps users define which assets will display on products in PDX. Four settings are available for 'PDX: Relevance'. They are:

- 'Yes', which directs the system to display assets of that type on the product
- 'Primary', which directs the system to display an asset of that reference type as the primary image
- 'No', which tells the system the asset reference is not relevant for suppliers and will not display in PDX
- [blank], which gives the same direction to the system as 'no'. Each asset reference type being used in PDX can only have a single valid target type.

When submitting products with referenced assets from PDX to STEP, PDX uses the logic described in the table below to determine which asset type to use when creating the assets in STEP:

Asset reference	Asset object type
PMDM.IDRT.InstallationManual	'InstallationManual'
PMDM.IDRT.OwnersManual	'OwnersManual'
All other asset references	'ProductImage'

For example, if a retailer wants an 'installation manual' reference available for the supplier when the supplier creates the product record, the retailer can go to the reference type definition in STEP, in this case 'PMDM.IDRT.InstallationManual', and the retailer can set the attribute 'PDX: Relevance' to 'Yes'. This allows the supplier to link an installation manual to the product to be submitted. The installation manual submitted via PDX to STEP will be of the asset object type 'InstallationManual', as per this table.

Primary Image in PDX

When configuring which image should display on the product in the 'List' and 'Grid' views on the Product Summary page for PDX, it is important to note that only one asset reference type should have the 'PDX:

Relevance' attribute set to 'Primary'. This is because this asset reference will then be used to supply the image asset to display as the 'Primary Image' in PDX.

If multiple asset reference types have the 'PDX: Relevance' attribute set to 'Primary', the system will select an asset at random to display as the Primary Image in PDX. If no asset reference type has the 'PDX: Relevance' attribute set to 'Primary', the asset linked via the reference type PMDM.IDRT.PrimaryProductImage will be used as the default primary image in PDX.

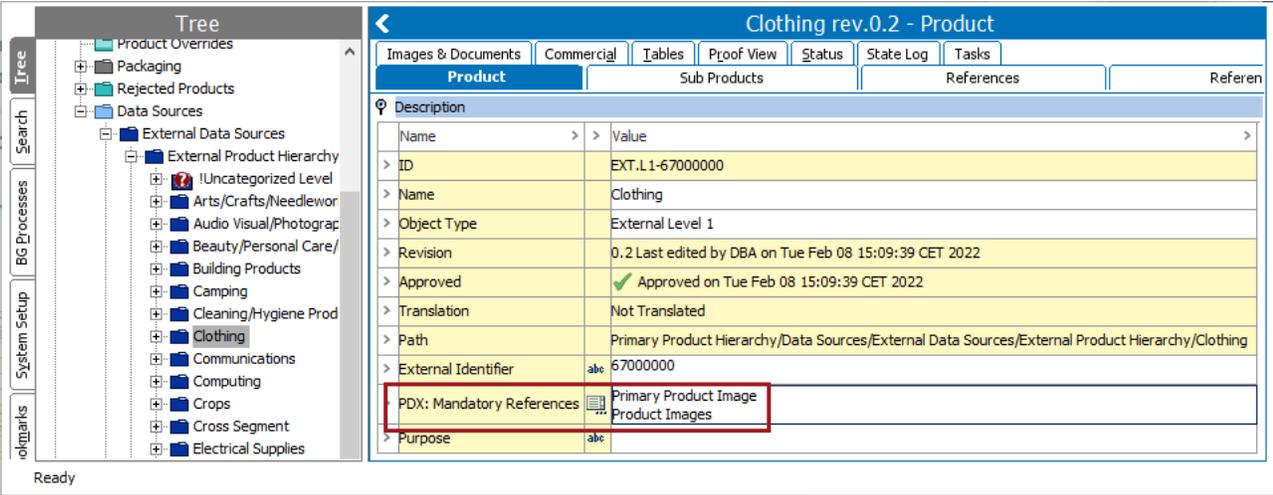
Maximum Number of Assets

Accelerator for Retail supports the ability to set a maximum number of references allowed on a product for a given asset reference type. As shown in the screenshot below, on the 'PDX: Maximum Number of References' attribute, the admin user has set this maximum to '8'. This pre-set maximum will then be enforced in PDX. In practice, this means that a retailer can decide, for instance, that they want the supplier to provide no more than eight product images per submitted product, and the system enforcement means suppliers must abide by this setting. Products that do not satisfy this rule cannot be submitted and the corresponding asset attribute will be marked with a red error.

Product Images - Reference Type	
Reference Type	Validity Log
Description	
Name	Value
ID	PMDM.IDRT.ProductImages
Name	Product Images
Last edited by	2022-11-15 16:36:43.0 by PVMA
Externally Maintained	No
Dimension Dependencies	
Allow multiple references	Yes
Mandatory	No
Inheritance	Inherited
PDX: Description	abc
PDX: Maximum Number Of References	8
PDX: Relevance	Yes
Purpose	abc
In Attribute Groups	
ID	Name
PMDM.ATG.BuyerApprove	Buyer, Approve
PMDM.ATG.BuyerCreate	Buyer, Create
PMDM.ATG.BuyerDelete	Buyer, Delete
PMDM.ATG.BuyerModify	Buyer, Modify
PMDM.ATG.BuyerView	Buyer, View

Category-Specific Mandatory Asset References

Category-specific asset references that must be mandatory for users in PDX are controlled on the product category node itself. The 'PDX: Mandatory References' attribute, which allows for multiple values, contains the list of mandatory references on that category. For example, the retailer has determined that the 'Primary Product Image' and 'Product Images' asset references should be mandatory for suppliers in PDX. On the 'Clothing' product category node, as shown in the screenshot below, the retailer has added both references to the 'PDX: Mandatory References' attribute.



Asset Validation

For retailers with specific requirements for supplier-delivered assets linked to submitted products, Accelerator for Retail supports the ability to validate assets in the PDX Channel. For instance, if a retailer requires product-linked image assets to be .jpg files, a validation can be applied in Accelerator for Retail in STEP that will ensure that image asset files submitted via PDX will be .jpgs.

Note: The asset validations are tied to the Asset Type in STEP, not the Asset Reference Type.

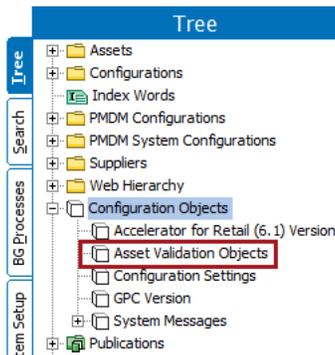
Described below is a list of asset validations that may be applied in STEP and enforced in PDX.

- MIME type – By applying a MIME type limitation, retailers can restrict submitted assets to only those with the desired MIME types. For instance, if a retailer wants all instruction manuals associated with the submitted product to be of the .pdf MIME type, the validation applied in the configured product data validation would be 'application/pdf'. This validation is applied directly on the object type, found under 'Object Types & Structures' on the 'System Setup' tab in workbench. In the example shown in the screenshot below, the object type is 'ProductImage'. In this case, the admin has determined that if images submitted for products are not of the MIME types listed in the 'MIME Types' field, the product cannot be submitted.

Product Image - Object Type	
Object Type	References
Description	
Name	Value
> ID	ProductImage
> Name	Product Image
> Last edited by	2023-02-02 11:04:58 by DBA
> Name Pattern	
> ID Pattern	
> Icon	
> MIME Types	image/tiff image/tif application/postscript image/*
> Dimension Dependencies	
> Reference Target Lock Policy	Strict
> Asset Keywords	abc
> Asset URL Attribute	URL >>
> PDX: Asset Content Hash	abc
> PDX: Asset File Name	abc
> PDX: Asset URL	URL >>

For more information on MIME types, refer to the **MIME Types** topic in the **Object Types and Structures** section of the **System Setup** documentation.

The validations listed below can be applied on 'Asset Validation Objects', which can be created below the 'Asset Validation Objects' entity, found under 'Configuration Objects' on the 'Tree' tab in workbench.



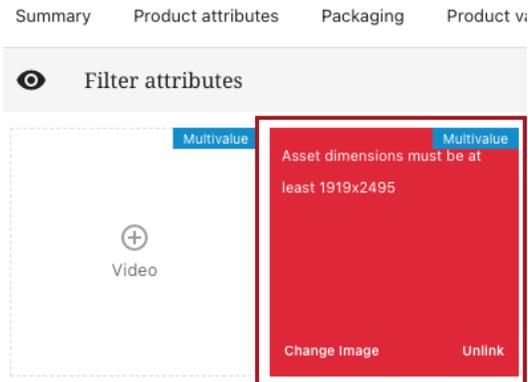
Asset validation objects are created to define additional asset validation rules on assets in PDX. The validations display like this on the asset validation object in STEP:

Product Images rev.0.1 - PDX: Asset Validation Object	
PDX: Asset Validation Object	
References Referenced By Proof View Status State Log Tasks	
Description	
Name	Value
ID	18648652
Name	Product Images
Object Type	PDX: Asset Validation Object
Revision	0.1 Last edited by BRRCR on Fri Mar 17 23:47:01 CET 2023
Path	Entity hierarchy root/Configuration Objects/Asset Validation Objects/Product Images
Aspect Ratio	1:1
Asset Object Types	Product Image
Maximum File Size (Bytes)	
Maximum Pixel Height	4000
Maximum Pixel Width	4000
Minimum Pixel Height	1200
Minimum Pixels (Longest Side)	4000
Minimum Pixel Width	1200
Valid Color Spaces	RGB color

Listed below are the various validation settings that can be applied on the asset validation object, accompanied with a description of each:

- **Aspect Ratio** – Allows retailers to specify the aspect ratio (the ratio of an image’s width to its height) of the product-linked images submitted by suppliers via PDX. Select from the options available in the dropdown: 1:1, 2:33, 1:1, 2:3, 3:4, 16:9 and 16:10. If the submitted image does not conform to the specified aspect ratio, the product submission will be rejected, and an explanatory error message will display.
- **Maximum File Size (Bytes)** -- A product-linked image submitted via PDX cannot have a larger file size than the number of bytes set for this parameter.
- **Asset dimension (in pixels)** -- The size, in pixels, of product-linked image assets submitted by suppliers via PDX can be strictly controlled by setting a series of attributes on the asset reference type definition. Below is a list of these settings along with a description of the image aspect they control:
 - **Maximum Pixel Height** – The image height of submitted images can be no greater than the number (in pixels) set in this parameter.
 - **Maximum Pixel Width** – The image width of submitted images can be no greater than the number (in pixels) set in this parameter.
 - **Minimum Pixel Height** – The image height of submitted images can be no less than the number (in pixels) set in this parameter.
 - **Minimum Pixels (Longest Side)** -- The image width or height of submitted images, whichever is greater, can be no less than the number (in pixels) set in this parameter.
 - **Minimum Pixel Width** – The image width of submitted images can be no less than the number (in pixels) set in this parameter.
- **Valid Color Spaces** – Allows retailers to specify the image mode of the product-linked image asset being submitted. Select from the options available in the dropdown: Bitmap, CMYK, Grayscale, Indexed, and RGB. If the submitted image does not conform to the specified image mode, the product submission will be rejected, and an explanatory error message will display.

If an asset does not satisfy any of these configured rules, the corresponding product cannot be submitted and the asset will be marked with a red error, as shown in the screenshot below:



Additional information about the various validations that can be applied on the asset validation object are described below:

Name	ID	Type	Description
Asset Object Types	PMDM.AT.PDS.AssetObjectTypes	Multi-valued Lists of Values (LOV) of asset object types.	This attribute contains the asset object types this asset validation object is relevant for.
Maximum File Size (Bytes)	PMDM.AT.PDS.MaxFileSizeBytes	Integer	This attribute contains the maximum allowed file size a digital asset of this type may have
Maximum Pixel Height	PMDM.AT.PDS.MaxPixelHeight	Integer	This attribute contains the maximum allowed pixel height that a digital asset of this type may have
Maximum Pixel Width	PMDM.AT.PDS.MaxPixelWidth	Integer	This attribute contains the maximum allowed pixel width that a digital asset of this type may have
Minimum Pixel Width	PMDM.AT.PDS.MinPixelWidth	Integer	This attribute contains the minimally required pixel width that a digital asset of this type must have
Minimum Pixel Height	PMDM.AT.PDS.MinPixelHeight	Integer	This attribute contains the minimally required pixel height that a digital asset of this type must have
Valid Color Spaces	PMDM.AT.PDS.ValidColorSpaces	Multivalued LOV of color spaces	This attribute contains the color spaces that a digital asset of this type must have
Aspect Ratio	PMDM.AT.PDS.AspectRatio	Multivalued LOV of color spaces	This attribute contains the aspect ratios that a digital asset of this type must have

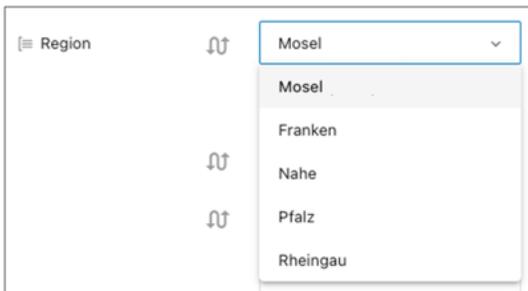
Controlling the PDX Channel for Product Onboarding - LOV Cross-validation

Retailers can configure their Accelerator for Retail system so that when PDX users who are onboarding products select a specific value for one List of Values (LOV) attribute, the options available to them in a different LOV attribute will be pre-filtered, presenting the user with a retailer-configured subset of LOV values from which to select. This can be done by configuring a product data validation with an LOV cross-validation condition.

As an example, an LOV attribute named 'Country' is a product attribute a supplier can enrich in PDX. From the LOV dropdown, the user has selected the value 'Germany', as shown in the screenshot below.

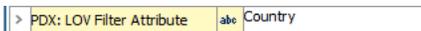


Because the LOV cross-validation has been implemented, when the PDX user selects 'Germany', the LOV options available in the LOV attribute 'Region' will display only German regions.



This pre-filtering is controlled via the 'PDX: LOV Filter Attribute' metadata attribute, which is found on the attribute definition.

The 'PDX: LOV Filter Attribute' metadata attribute is used to enable the connection between the two relevant LOV-based attributes. For the dependent attribute, which is the LOV-based attribute whose values will be filtered, the value assigned to 'PDX: LOV Filter Attribute' should be the ID of the defining attribute, which is the LOV-based attribute that determines the valid values. To continue the example described above, the dependent attribute 'Region' would have 'PDX: LOV Filter Attribute' populated with the value 'Country'.



Additional capabilities and considerations for LOV Cross Validation

The ValueIDs for the two LOVs configured for LOV Cross-Validation must also satisfy additional requirements for the cross-validation to work.

- The ValueIDs of the dependent LOV values, the values being filtered, must be prepended with the value ID of the defining LOV for the cross-validation to work properly.

To illustrate using the example described previously, the LOV attribute 'Country' includes an LOV value 'Germany' with the ValueID 'germany'. To restrict the region options available in the LOV attribute 'Regions' to only German regions when 'Germany' is selected, the ValueID for the German region 'Mosel' must begin with the ValueID for 'Germany', which is 'germany'. Using this logic, a possible ValueID for 'Mosel' could be 'germany_mosel'.

- The LOV cross-validation allows for multiple layers of dependent attributes, not just two.

To expand upon the previous example, let us say the retailer wants to include a second dependent LOV attribute, in this case 'City', to the cross-validation. In this scenario, the 'PDX: LOV Filter Attribute' for 'City' must be set to 'Region'. A supplier would first select 'Germany' for 'Country', then 'Mosel' for 'Region', and then, for the 'City' LOV attribute, only retailer-relevant German cities in the selected region should be available for the supplier to select. So, if an LOV value from the LOV 'City' is to be a selectable value when the user selects 'Mosel', the ValueID for the valid city, in this case 'Koblenz', will need to have both of its defining LOV values prepended to its ValueID. Using this logic, a possible ValueID for 'Koblenz' could be 'germany_mosel_koblenz'.

Controlling the PDX Channel for Product Onboarding - Maintenance

When PDX is adopted by retailers as the primary product-onboarding tool for bringing products into an Accelerator for Retail STEP instance, the retailer has likely migrated supplier-provided product records into STEP from a different tool previously. Because the retailer has transitioned to a second tool in this scenario, the supplier-provided products in the retailer's STEP system are now out of sync with the same products in the supplier's system. This can present challenges when retailers and suppliers move forward together with Accelerator for Retail and PDX.

The 'Maintenance' functionality enables retailers to continue receiving product data from suppliers for new and existing products and allows suppliers to maintain those products after the retailer transitions to the PDX platform. 'Maintenance' is not intended to establish a continuous two-way data synchronization between PDX and STEP, but rather to push existing products, on-demand, from STEP to suppliers in PDX.

The Maintenance functionality allows retailers to push existing products from STEP into the relevant Accelerator for Retail channel in PDX, in effect sending them back to the supplier. In that channel, the products will be processed as so-called 'retrieved products' for the supplier in question. At this point the supplier can reclaim the products and maintain them as they would any other.

When retrieved products enter PDX, two things happen:

- The products will be created in the PDX channel for the relevant supplier
- The products will appear in the master data area in PDX

The product's ID in PDX, as well as the attribute values the retailer wants displayed in the channel and in master data, can be configured in STEP. The product data shared in this way can be used by the supplier to recognize the incoming product and match it to the supplier's existing product record.

A product that enters PDX in this way will be marked as a 'retrieved product' and enter the PDX onboarding process in the workflow state named 'Made available for maintenance in <name of channel>,' as shown in the screenshot below.

✕ 05901851220645

Summary Product attributes Packaging Digital assets

Product details

Retrieved product

Name	05901851220645
Category	Camping Beds/Sleeping Mats Product Root > External Product Hierarchy > Camping > Camping Beds/Sleeping Mats
Introduced to channel	Feb 28, 2023, 10:37:11 AM
Last updated	Mar 6, 2023, 12:07:31 PM
Submitted	-
Online	-
Status	Progress
External product ID	EXT-1893010
Errors	-

Workflow state

Made available for maintenance in ACC4R DEV

Data pushed from STEP to the PDX channel is stored in the ‘retrieved data layer’ in PDX. Changes to a product made by the supplier, either manually via direct edit into the channel or automatically on import via a standard product record update, will overwrite the retrieved data, making it possible for the supplier to maintain the product fully.

Recommended practice for retailers using the Maintenance functionality is to exchange with the supplier only those product identifiers and relevant data that will ensure suppliers can recognize and match the retailer’s products to the products in the supplier’s product database. It is not recommended that retailers push all product details from STEP to PDX. Although the product is pushed from the retailer (STEP) to the supplier (PDX) using the Maintenance process, it remains the supplier’s product, the supplier’s data, and the supplier’s responsibility to maintain all product data, only now the data is being maintained from PDX.

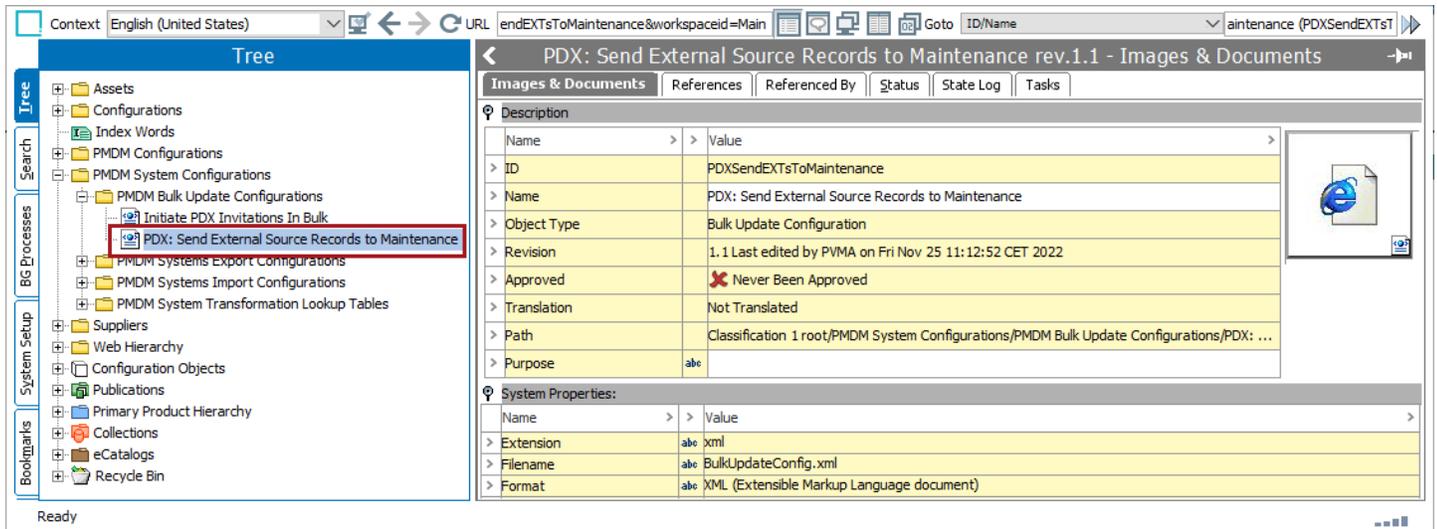
Pushing product to PDX in Maintenance

In the Maintenance process, products (referred to as ‘External Source Records’ in STEP) are pushed from retailers to suppliers by generating events on the ‘PDX: Event Queue – Maintenance’ (ID: PMDM.EQ.PDS.EventQueueMaintenance) event queue.

Note: If multiple events are generated simultaneously, PDX may require additional time (up to 24 hours or more) to consume and distribute all events.

The event queue will automatically include any packaging objects in the messages made available to PDX, meaning that when a product is pushed to PDX, the parent packaging hierarchy will also be pushed to PDX.

The Accelerator for Retail solution includes a bulk action called ‘PDX: Send External Source Records to Maintenance’ (ID: PDXSendEXTsToMaintenance) that can be used to generate an event on a collection of products. This might be done, for example, on all products for a specific supplier.



When the bulk action is run, it checks that all products included in the bulk action satisfy the minimal requirements for being pushed to PDX, which are:

- The product must reference a supplier
- The product must have a Maintenance ID

If the products included in the bulk action fail to comply with these two requirements, the validation for that product has failed, no event is generated, and the process moves to the next product.

Note: For a channel with Maintenance enabled, a given supplier can only be authenticated under one PDX Client login. This means that if the same supplier attempts to add the channel in another PDX client, they will get the following error message: "This vendor has already been added to this channel. Please configure another Vendor Identifier and try again."

Configurations related to Maintenance

To fully enable Maintenance in Accelerator for Retail, additional configuration steps are required. These steps are listed and described below.

- **Attributes to include in Maintenance** -- Attributes and asset references that should be added to the product in the PDX channel and stored in the retrieved layer should be linked into the attribute group 'PDX: Include in Maintenance' (ID = PMDM.ATG.PDS.IncludeInMaintenance).
- **Attribute to store Maintenance ID** -- As mentioned above, the product ID used in PDX can be configured from STEP. The STEP attribute for storing this ID is called 'PDX: Maintenance ID' (ID = PMDM.AT.PDS.MaintenanceID). By default, this product ID will be used in PDX to match a retailer's retrieved product with the corresponding product in the supplier's master data. This matching occurs when suppliers upload updates to their products via Excel sheet import or an integration using the external API.

- **Attributes to “promote” to master data** -- Attributes that will be visible in master data in PDX should be linked into the attribute group ‘PDX: Matching Keys for Maintenance’ (ID = PMDM.ATG.PDS.MaintenanceMatchingKeys).

These attributes should be IDs like GTINs or supplier product identifiers. Their values should make it possible for a supplier to first recognize the product and then deploy PDX’s ‘Match-based import’ (as shown in the screenshot below) and mapping capabilities to maintain the product.

Controlling the PDX Channel for Product Onboarding - Supplier-facing product hierarchy

The Accelerator for Retail solution is delivered with pre-configured object types that provide structure for a product hierarchy up to six levels deep. These object types also populate the selectable product categories in the PDX channel. The default object types that make up this structure are named generically: External Level 1, External Level 2, etc, and are defined in STEP below the 'External Data Sources' node, as shown in the screenshot below:

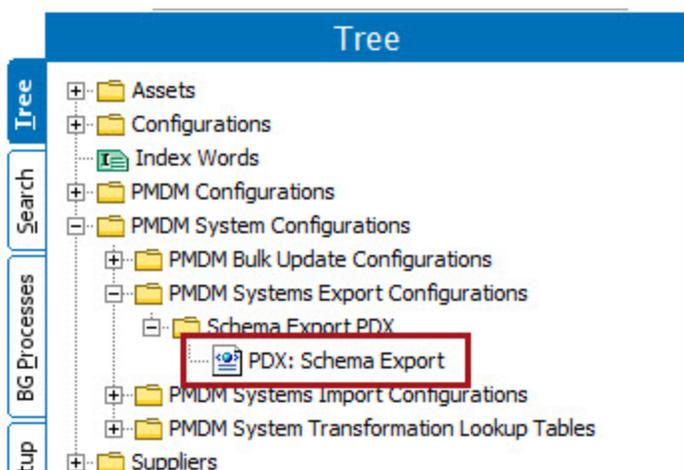


These default object types, as provided in the base setup, are often sufficient for customers and are used for the Global Product Classification (GPC) hierarchy. If needed by the customer, this hierarchy can be provided with Accelerator for Retail solutions.

Note: Sellable products, referred to as 'External Source Records' in STEP, can be created as children objects under 'External Level 3' through 'External Level 6' object types, but not under 'External Level 1' or '2'.

For customers who do not require changes to the default object types, adding additional category nodes using these object types will require no updates to the export configuration; the changes will automatically be included in a Data Standard update. However, if customers want to, for example, add a new level to the hierarchy (i.e., an 'External Level 7' object type), the export configuration must be updated so the changes are properly mirrored in PDX.

To make these changes, users must edit the export configuration named 'PDX: Schema Export', shown in the screenshot below.



For more information on editing export configurations, refer to the **Export Manager** documentation.

Controlling the PDX Channel for Product Onboarding - Supplier Hierarchy

A supplier hierarchy (shown in the screenshot below) is used by retailers to manage suppliers, allowed locations (shipping points), as well as products and assets submitted from the suppliers.



Before a supplier can submit products from PDX to Accelerator for Retail, the following attributes on the Supplier Account classification must be defined:

- Supplier ID
- Supplier Name
- Type of Supplier
- General Rank Type of Supplier
- General Rank for Supplier
- General Handling of New Source Records
- General Handling of Updated Source Records

As an example of how these attributes might display, review the screenshot below.

For additional information on the supplier specific attributes, refer to the Setup of New Suppliers topic.

Supplier Account

Suppliers

Acme Stationery

41d6c4a9-344b-4943-a4ed-d779918c70fc-CL • Approved • 0.2 • Last edited on September 8, 2022 at 10:23:53 AM UTC-4

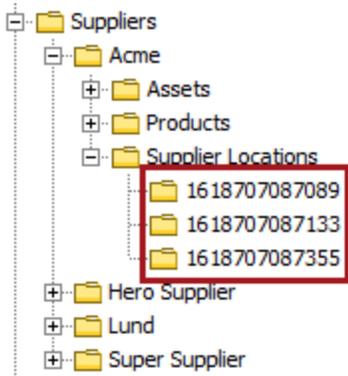
0 Profile Findings

Sub objects **Basic** PDX

Supplier ID	Acme_Stationery	<h4>Product Onboarding Status</h4> <table border="1"> <tr> <td>Total Products</td> <td>0</td> </tr> <tr> <td>Accepted Products</td> <td>0</td> </tr> <tr> <td>Products Being Reworked</td> <td>0</td> </tr> <tr> <td>Rejected Products</td> <td>0</td> </tr> </table>	Total Products	0	Accepted Products	0	Products Being Reworked	0	Rejected Products	0
Total Products	0									
Accepted Products	0									
Products Being Reworked	0									
Rejected Products	0									
Supplier Name	Acme Stationery 									
Type of Supplier	Manufacturer 									
General Rank Type of Supplier	Approved Content Source 									
General Rank for Supplier	2 									
General Handling of New Source Records	Manual approval of new Source 									
General Handling of Updated Source Records	Automatic approval of updated S 									

 Save
  Reset
  Save & Approve
  Reset PDX Channel Account Association
 

Each supplier location or Global Location Number (GLN) submitted from a supplier is created in Accelerator for Retail as a separate ‘Supplier Location’ classification. For example, the screenshot below shows three GLNs stored in the ‘Supplier Locations’ node.



A ‘Supplier Location’ object (shown in the Web UI in the screenshot below) holds the following attributes:

- ‘Provider GLN’ -- an optional field used by the retailer or supplier (depending on the solution implemented) to identify the specific location from which products are shipped to the retailer.
- ‘Is Active’ -- if set to ‘Yes’, the retailer can accept products delivered from the location (shipping point); if set to ‘No’, the location is invalidated, and a supplier cannot set the location on a product in PDX to this location.

Supplier Location

Suppliers ▶ Acme ▶ Supplier Locations

1618707087089

LOC_100521 • Approved • 0.2 • Last edited by stepsys on March 1, 2023 at 4:24:19 PM UTC+1

Sub objects
Basic

Name	<input type="text" value="1618707087089"/>
Provider GLN	1618707087089
Is Active	<input type="text" value="Yes"/>

Note: If the retailer has both Accelerator for Retail and the Supplier Onboarding / Supplier Self-service package from Customer Master Data Management (CMDM), the supplier can provide the GLNs. If the retailer has only Accelerator for Retail, the retailer adds and maintains the GLNs for this field.

For each product, the supplier must select a primary location (shipping point) from a list of active / allowed locations for the supplier. This location should be added to the 'Primary Location (GLN)' attribute on the relevant 'Supplier Location' object. Additional locations can be added to the 'Locations (GLNs)' attribute. When a product is onboarded from PDX into Accelerator for Retail, the GLN set for the 'Primary Location' attribute will be used to create a reference from the product to the 'Supplier Location' object for which the GLN has been set. The GLNs set for the 'Secondary Location' attribute are used to create references from the product to those 'Supplier Location' objects for which the GLNs have been set.

Note: The list of active locations for a supplier is built in PDX. By collecting all 'Supplier Location' objects having 'Is Active != No' for the supplier (with '!=' meaning 'does not equal'). The list collects all 'Supplier Location' objects with either the value 'Yes' for the 'Is Active' attribute, or no value at all. This list is generated when the channel is added in PDX and refreshed each time the supplier logs into PDX. No Data Standard update is required to ensure location changes are up to date in PDX.

Note: If the use of Locations is not needed, it can be disabled by unlinking the 'Primary Location (GLN)' attribute from the 'PDX: Mandatory for Submit' attribute group.

Storing supplier products and assets

To categorize products in the product hierarchy in Accelerator for Retail, the 'Supplier Link Product to Classification' reference is used to link the products and their associated packaging objects to the 'Products'

folder.

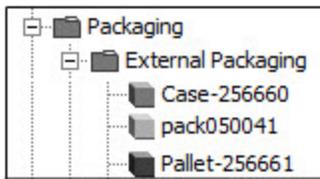
Assets are loaded into the appropriate 'Supplier Assets' folder.

Controlling the PDX Channel for Product Onboarding - Additional Considerations

When configuring PDX through a STEP instance of Accelerator for Retail, there are some additional considerations to keep in mind. Described below are aspects of setting up both the display and behavior of PDX via Accelerator that are important to consider.

Packaging

Packaging objects ('Pallet', 'Case', 'Pack') are created as separate objects directly below the 'External Packaging' node in STEP.



References connect each packaging object to the next lower-level packaging object and from the lowest level to the sellable product (also referred to as External Source Records).

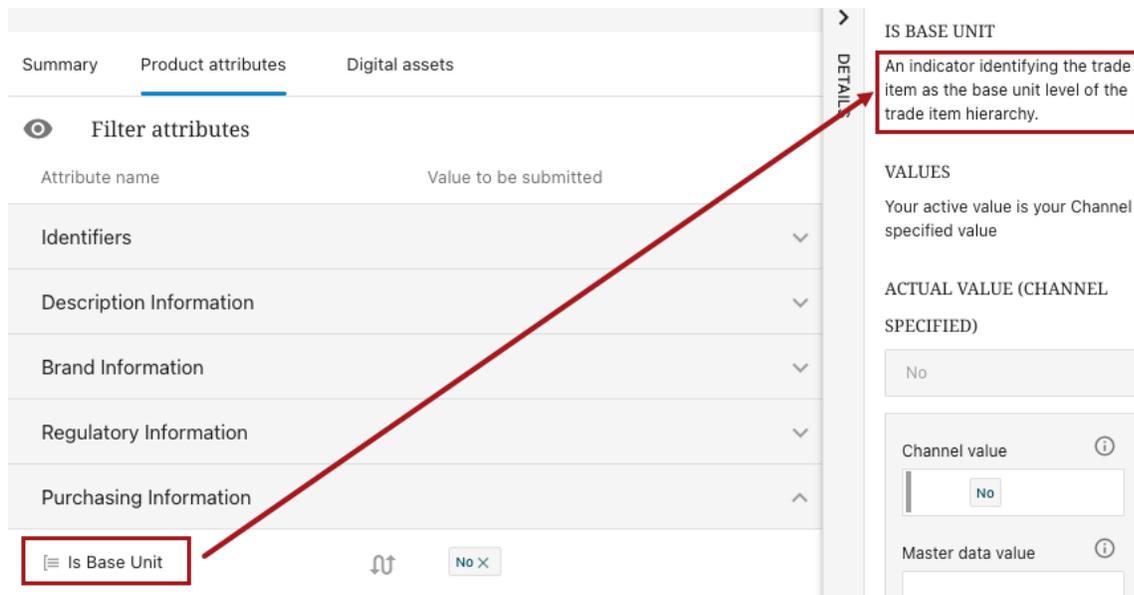
The quantity in each packaging object is held in the 'Qty of Next Lower Package' metadata attribute found on the packaging reference.

Attribute Help Text

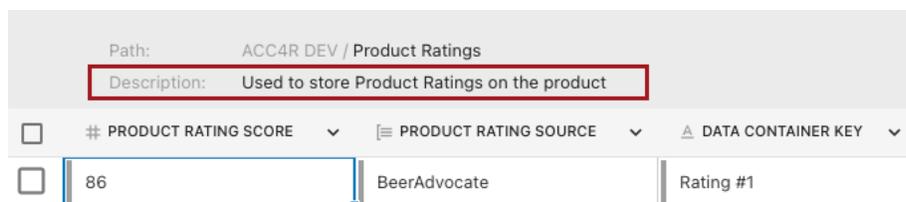
To provide additional guidance to suppliers and other users enriching product data, admins may write help text in Accelerator for Retail in STEP that will be available to suppliers in PDX. To display help text on attributes, data containers, and asset references accessible by PDX users, the objects for which help text has been written must be linked to the PDS: Description (STEP ID: PMDM.AT.PDS.Description) attribute. Content added to this attribute will sit as metadata on the attribute, attribute group, data container, or reference.

In PDX, the help text will display in one of two ways:

- When the attribute name is clicked, the "Details" panel will display on the right-hand side of the screen. The configured help text will display in this panel.



- On a data container, the configured help text will display above the columns to the right of the field name, 'Description', as shown in the screenshot below.



DX uses the 'PDX: Product Import' import configuration to import product information from STEP.

The product import configuration executes the 'PDX: Proceed Product on Import' business action on each product being imported.

Import of assets

PDX uses the 'PDX: Asset Import' import configuration to import assets from STEP.

The asset import configuration executes the 'PDX: Upload Asset on Import' business action on each asset being imported.

Event Queue

Two event queues are used in the communication between STEP and PDX.

PDX: Event Queue - Status update -- When a product is submitted from PDX to STEP, or proceeds through the 'External Record Handling' workflow, or the product is being moved by the retailer, a status update is sent to PDX. This update is sent by creating an event on the 'PDX: Event Queue - Status update' event queue, which is read by PDX. For this process to work correctly, 'Queue Status' must be set to 'Read Events', and 'Consumer Read' must be enabled.

The status update sent to PDX contains the following information about the product:

- **STEP ID** -- This is displayed as the External Product ID on the product in PDX.
- **'PDX: ID' attribute value** -- This is used as the key between Accelerator for Retail and PDX.
- **'PDX: Workflow Event' attribute value** -- This attribute contains one of the following values in STEP:
 - PDS_ACCEPTED
 - PDS_REJECTED
 - PDS_RETURNED
 - PDS_SUBMITTED
 - These values map to the following statuses in PDX, respectively:
 - Accepted
 - Rejected
 - Returned
 - Submitted
- **'PDX: Workflow States' attribute value** -- This is a calculated attribute containing information about the progress of the product onboarding. The value will be shown as 'Workflow state' on the product in PDX.
- **'PDX: Message to Supplier' attribute value** -- This attribute contains product onboarding-relevant information from the retailer. For example, the reason a retailer has returned the product for rework. The value given to this attribute will be added as a new entry in the product's 'Product history' in PDX.

PDX: Event Queue – Maintenance -- When products are sent to PDX using the Maintenance feature, events are created on the 'PDX: Event Queue - Maintenance' event queue, which is read by PDX. For this process to work correctly, 'Queue Status' must be set to 'Read Events', and 'Consumer Read' must be enabled.

PDX Onboarding Channel Configurator

The PDX Onboarding Channel Configurator solution in the STEP system allows users to create PDX Onboarding channels with ease. This feature is designed to simplify the PDX channel creation and management in the Web UI. This enables more STEP implementations to easily adopt PDX as the vendor-facing product data onboarding solution. This feature significantly improves the ability to create onboarding channels easily, providing a more efficient solution for implementing and maintaining an onboarding channel from within the STEP system.

With the PDX Onboarding Channel Configurator component, retailers and their customers can benefit from an exclusive channel management solution that is standardized and driven by user input.

However, before using the PDX Onboarding Channel Configurator for PDX channel creation, it is recommended to complete a discovery /data model analysis as some adjustments may be required for the setup process.

The actual process of deploying an onboarding channel into the PDX environment requires the involvement of the PDX team and is subject to license requirements.

Configuring PDX Onboarding Channel Configurator

The topics presented in this section outline the configuration processes that are essential for users to enable the setup of the PDX Onboarding Channel Configurator solution. The preliminary phase of the configuration is conducted via the STEP Workbench, whereas the latter segment is configured through the Web User Interface (UI).

- PDX Onboarding Channel Configurator Solution Initial Setup
- Configuring Web UI for PDX Onboarding Channel Configurator Solution

Prerequisites

In order to access the PDX Onboarding Channel Configurator within the STEP system, it is imperative that the 'pdx-channel' add-on component is first installed on your system. Before embarking on the configuration of a PDX channel within the STEP system, the following prerequisites are fulfilled:

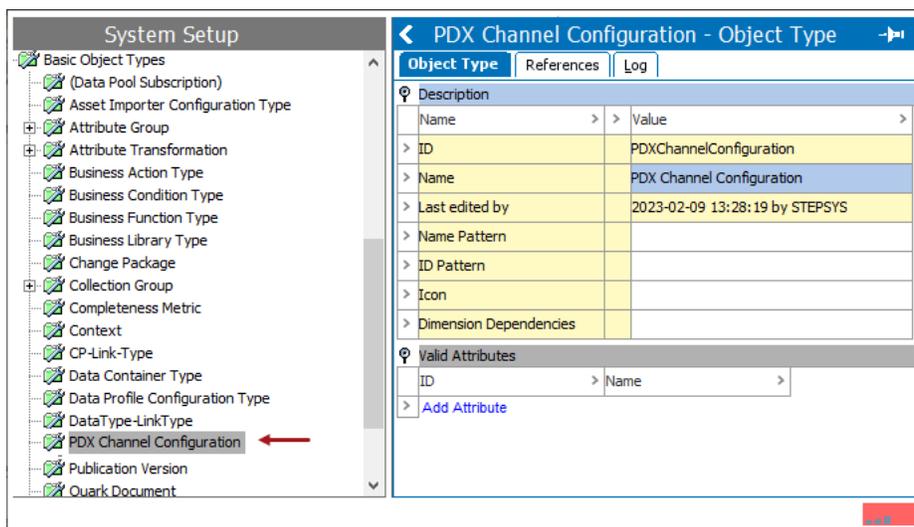
1. Installation of the PDX Onboarding Channel Configurator component
2. Installation of the Tagglo component. This includes the creation of some PDX upload folders on the STEP application server

For on-premise systems, instructions for installing components can be found in the SPOT Program topic in the System Administration Guide found in Downloadable Documentation. For SaaS systems, contact your Stibo Systems account manager.

PDX Onboarding Channel Configurator Solution Initial Setup

The PDX Onboarding Channel Configurator is the STEP setup entity object, which holds the definition of what the configurator should do. A one-time initial setup is required before any channel configurators can be created. A setup group must be created to hold the PDX Onboarding Channel Configurators. This document describes the process of initial setup, which constitutes setting up the data model in workbench.

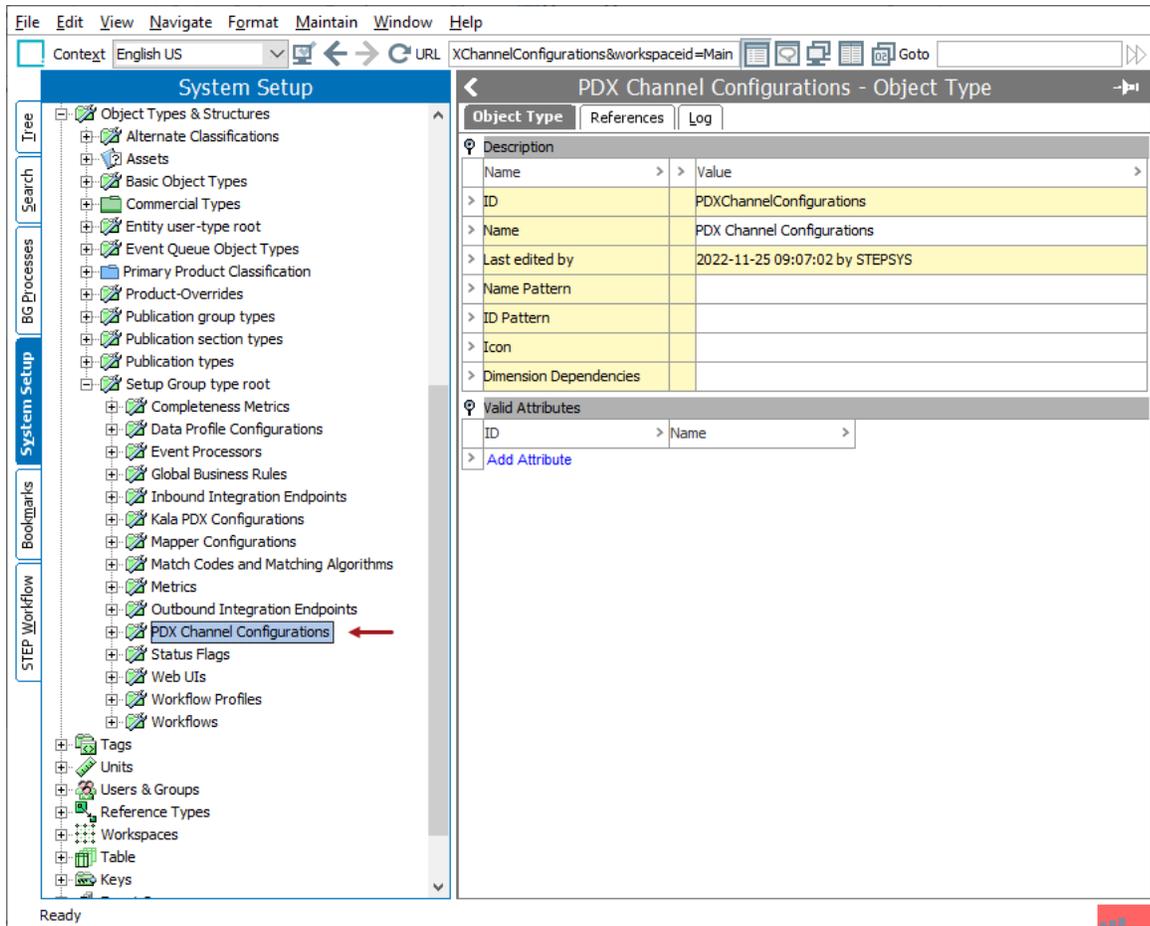
By the end of this configuration in the workbench, a setup group called 'Channel Configs' is created where users can add new PDX Onboarding Channel Configurators. Further configuration of the channel configurators can be only done in the Web UI. For information on configuring PDX Onboarding Channel Configurators in the Web UI, refer to the **Configuring Web UI for PDX Onboarding Channel Configurator Solution** topic.



Below are the steps for an initial setup in the workbench.

1. Navigate to **System Setup > Object Types & Structures**, select then right-click the **Setup Group type** root node, and select **New Object Type**.
2. In the dialog that appears, enter an ID in the ID parameter (e.g., PDXChannelConfigurations), enter a Name in the Name parameter (e.g., PDX Channel Configurations), then click the **Create** button. This creates a setup group object type that will be used for the root node of the configurators object to store all PDX Onboarding Channel Configurators below.

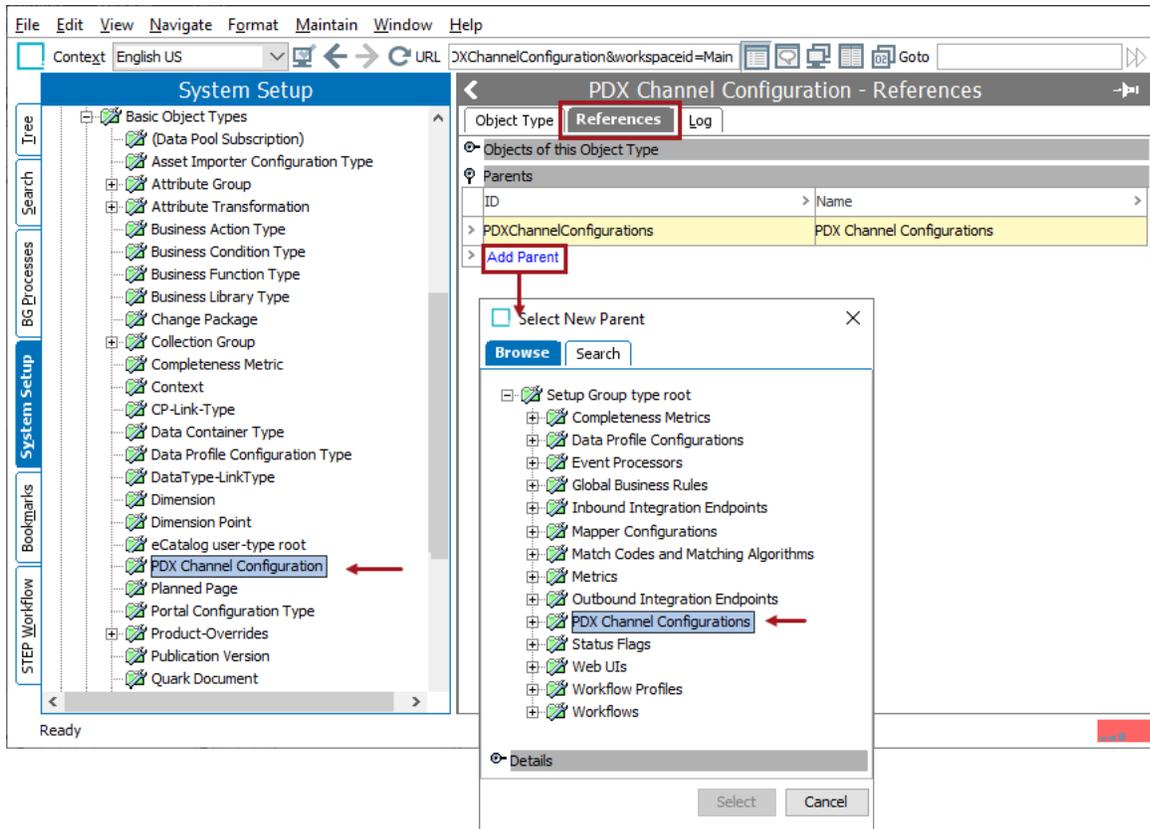
The sample below illustrates a hierarchy of setup group object types. A Setup Group named 'PDX Configurations' has been created.



After creating the setup group, it is essential to establish a parent-child relationship between the newly formed node and the 'PDX Channel Configuration' basic object type. By doing so, any future channel configurators added to the system will inherit this object type.

3. Navigate to **Object Types & Structures > Basic Object Types > PDX Channel Configuration**, and click the **References** tab.

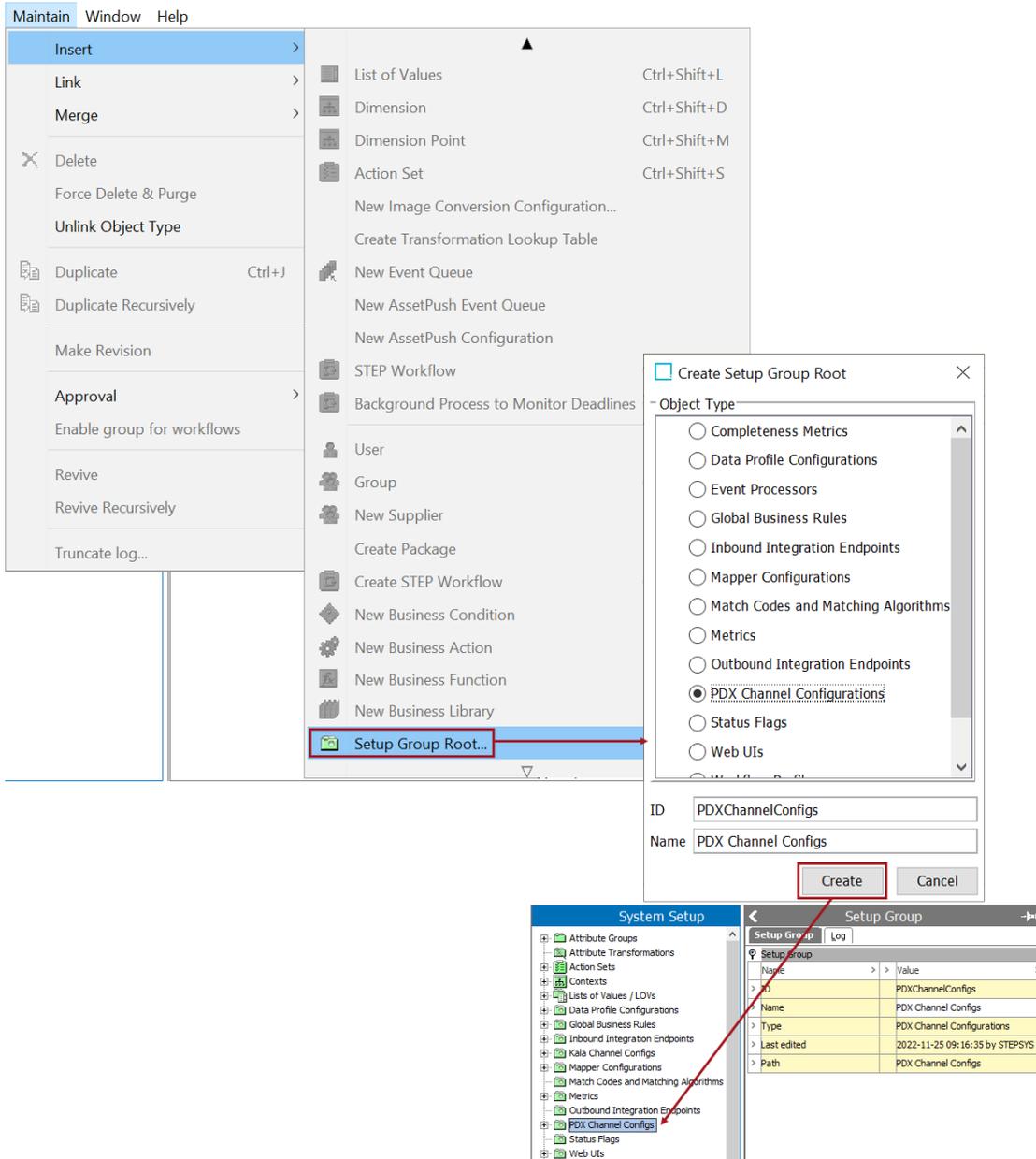
Under the **Parents** flipper, click **Add Parent**, and in the node selector dialog, specify the **PDX Channel Configurations** node that was created in the above step.



The next step is to create a new setup group root, where all channel configurators will be stored below.

Creating a New Setup Group Root

1. To create the setup group root, navigate to **Maintain** (drop-down menu) > **Insert** > **Setup Group Root....**
2. Select the 'PDX Channel Configurations' object type.
3. Specify the **ID** and **Name** for the setup group root object. In this setup, ID and Name value of 'PDXChannelConfigs' is entered.
4. Click **Create**.



The setup group called 'PDX Channel Configs' of the object type 'PDX Channel Configurations' is created. From this newly created node, users can right-click and add new PDX Onboarding Channel Configurators.

The PDX Onboarding Channel Configurators under the setup group 'PDX Channel Configs' are stored in STEP as a setup entity that can be exported and imported as STEP <Setup Entity> objects. It cannot be imported through Excel.

Configuring Web UI for PDX Onboarding Channel Configurator Solution

This topic covers configuring the PDX Onboarding Channel Configurator solution in the Web UI.

Prerequisites

Designers need to configure a Node Details Screen, a Global Navigation Panel that enables access to PDX Onboarding Channel Configurators, and different PDX Onboarding Channel Configurators to meet user requirements. Detailed configuration and usage instructions are provided in the following topics:

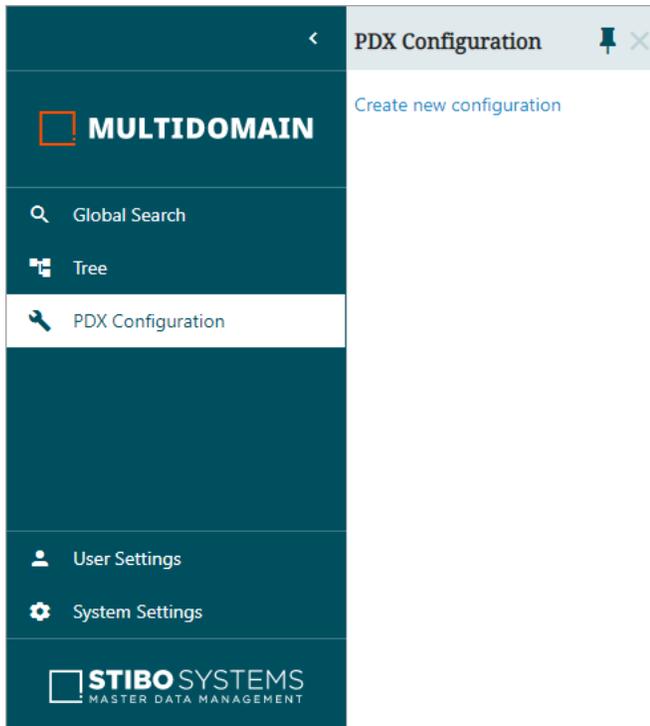
- Node Details Screen for PDX Onboarding Channel Configurator
- Configuring PDX Onboarding Channel Configurator Setup Entity

PDX Onboarding Channel Configurator Screen

A Node Details screen with the 'PDX channel Configurations Editor' component is used to view and edit PDX Onboarding Channel Configurators in the Web UI. Users can further define the parameters of the configurators available within this screen to configure the channels. For more information, refer to **Configuring PDX Onboarding Channel Configurator Setup Entity** topic.

PDX Onboarding Channel Configurators - Global Navigation Panel Component

The Global Navigation Panel should be configured as an entry point to display the list of available PDX Onboarding Channel Configurators. When accessed, a configured PDX Onboarding Channel Configurator screen is displayed that allows the user to both view and make additional configuration changes to the configurator object.



Multiple PDX Onboarding Channel Configurators can be created under the Global Navigation Panel to meet different PDX channel requirements. For example, there may be a need for multiple onboarding channels, I.e. to support multiple brands. All the available PDX Onboarding Channel Configurators are listed within the 'PDX channel configurations navigator' component of the Global Navigation Panel. For more information, refer to **PDX Onboarding Channel Configurator Screen** topic.

PDX Onboarding Channel Configurator Screen

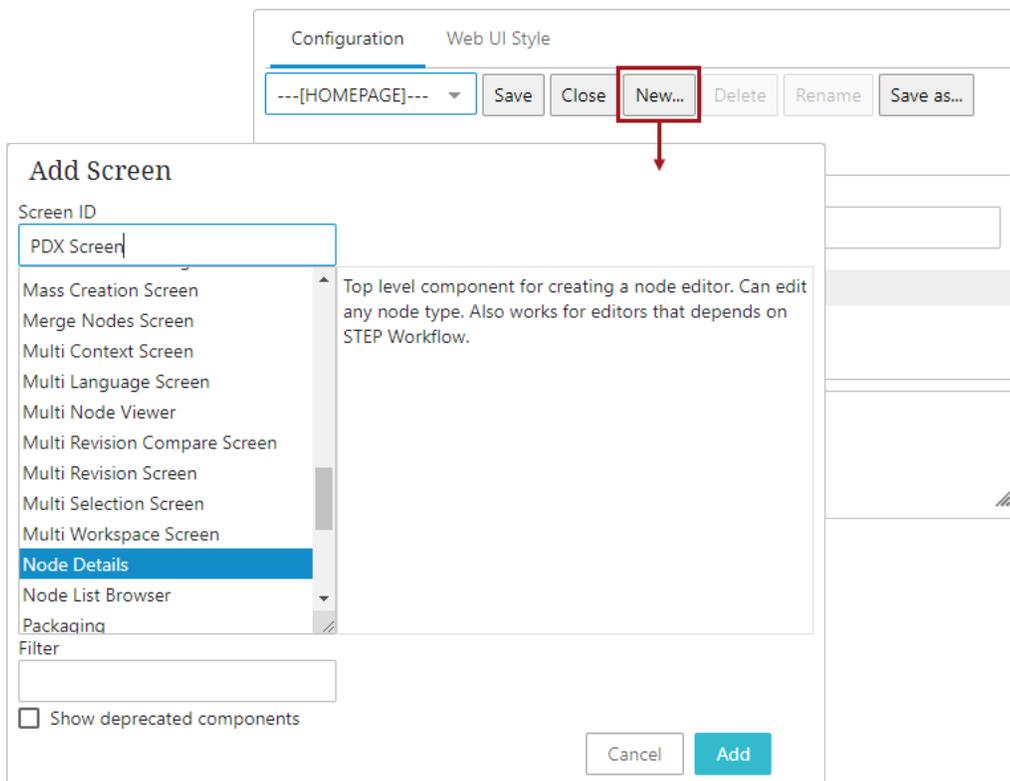
The PDX Onboarding Channel Configurator screen is a Node Details screen with a PDX channel Configurations Editor component configured in it. This screen allows users to quickly display PDX Onboarding Channel Configurators. The **PDX channel configurations navigator** component functions as a Global Navigation Panel component. For more information about Global Navigation Panel, refer to the **Global Navigation Panel** topic in the **Web User Interfaces** documentation.

Configuration Prerequisites

It is expected that anyone configuring a Node Details screen as a PDX Onboarding Channel Configurator screen is familiar with the Web UI Designer as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** section of the **Web User Interfaces** documentation.

Configuration Process

In the designer, create a new screen by selecting 'New.' Select the 'Node Details Screen,' and create a name for the screen in the 'Screen ID' field. In the example below, the Screen ID is 'PDX Screen.' Click 'Add.'



Once this screen is created, the Node Details designer screen displays, which contains the following parameters for configuration:

Configuration
Web UI Style

PDX Screen ▾

Save

Close

New...

Delete

Rename

Save as...

Node Details

Component Description Top level component for creating a node editor. Can edit any node type. Also works for editors that depends on STEP Workflow.

Title

Css Class

Show Title

▶ Validation

▶ Multiple Target References

Child Components

Below Title	<input style="width: 95%;" type="text" value="<Select a child component>"/>	go to component
Main	<input style="width: 95%;" type="text" value="<Select a child component>"/>	go to component
Buttons	<input style="width: 95%;" type="text" value="<Select a child component>"/>	go to component

Main: To access and modify the PDX Onboarding Channel Configurators, the 'PDX channel Configurations Editor' component must be configured within the Main parameter. This component will add a distinct functionality to the Node Details screen, allowing the PDX Onboarding Channel Configurators to be viewed and edited. To select this component, choose the 'PDX channel Configurations Editor' option from the dropdown menu in the Main parameter, as illustrated below.

Configuration Web UI Style

PDX Screen Save Close New... Delete Rename Save as...

Node Details

Component Description Top level component for creating a node editor. Can edit any node type. Also works for editors that depends on STEP Workflow.

Title

Css Class

Show Title

Validation

Multiple Target References

Child Components

Below Title [go to component](#)

Main [go to component](#)

Buttons [go to component](#)

Buttons: The PDX Onboarding Channel Configurator screen can include several action buttons that can be customized to perform specific tasks with the PDX Onboarding Channel Configurators. To add these buttons, click on the 'go to component' link adjacent to the Buttons parameter. The 'Actions' field allows the addition and configuration of the desired action buttons. It is mandatory to configure the 'Save Action' and 'PDX channel configurations publish' buttons, while the addition of other actions should only occur if necessary.

Configuration Web UI Style

PDXConfigurationSc Save Close New... Delete Rename Save as... [go to parent](#)

Buttons

Component Description Contains a number of buttons to display

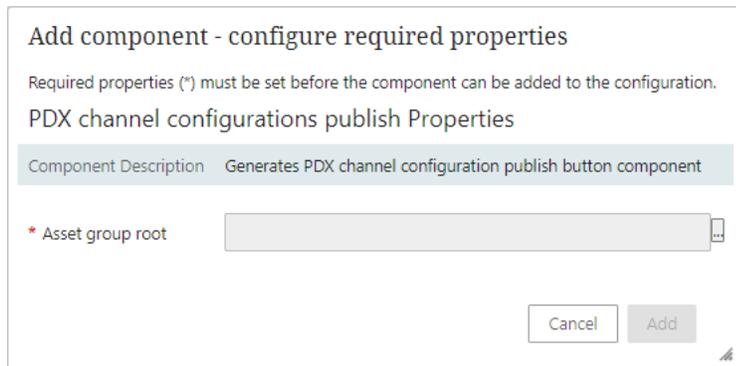
Child Components

Actions

- Save Action
- PDX channel configurations publish
- Delete Action
- Reset Action
- Duplicate Action

Add.. Remove Up Down

The inclusion of the 'PDX channel configurations publish' button is crucial as it enables users to publish the channel. To configure this action button, an asset folder must be designated to store all the published XML files. To select an asset folder, click the ellipsis button (...) next to the 'Asset group root' parameter.



The Save and Publish action generates an XML representation of the Channel Configuration and saves it as an XML file asset in the designated asset folder. This file can be extracted by the PDX platform to deploy or update the channel as needed.

When a PDX Onboarding Channel Configurator is deleted through the Web UI, it will be transferred to the Recycle Bin in System Setup. Users can revive the PDX Onboarding Channel Configurators via the 'Maintain' dropdown > Revive.

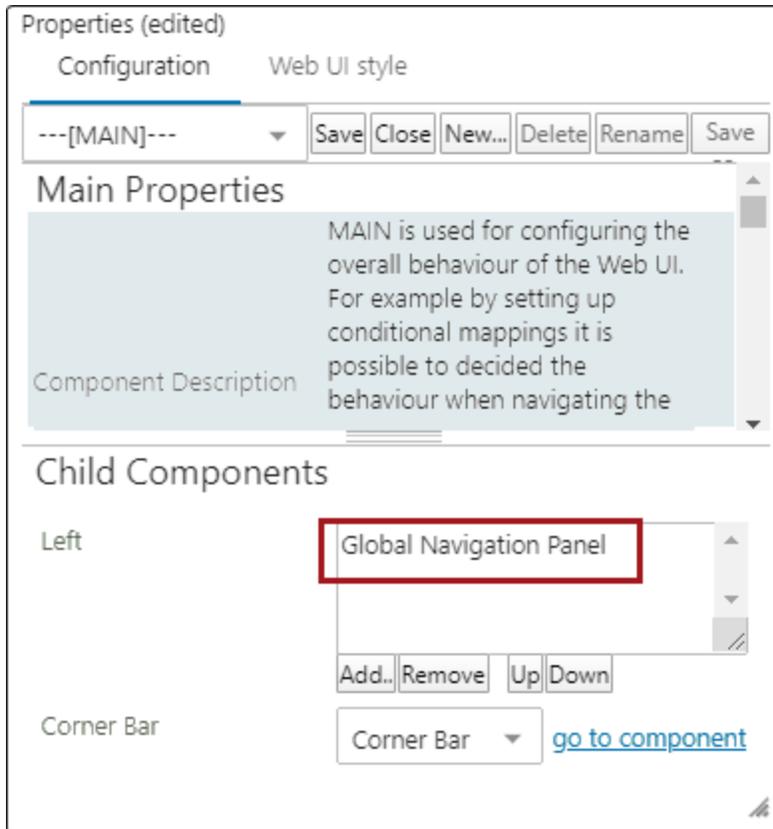
Implementing the PDX Onboarding Channel Configurator screen

After creating the 'PDX Screen,' the following setup is necessary to implement the screen:

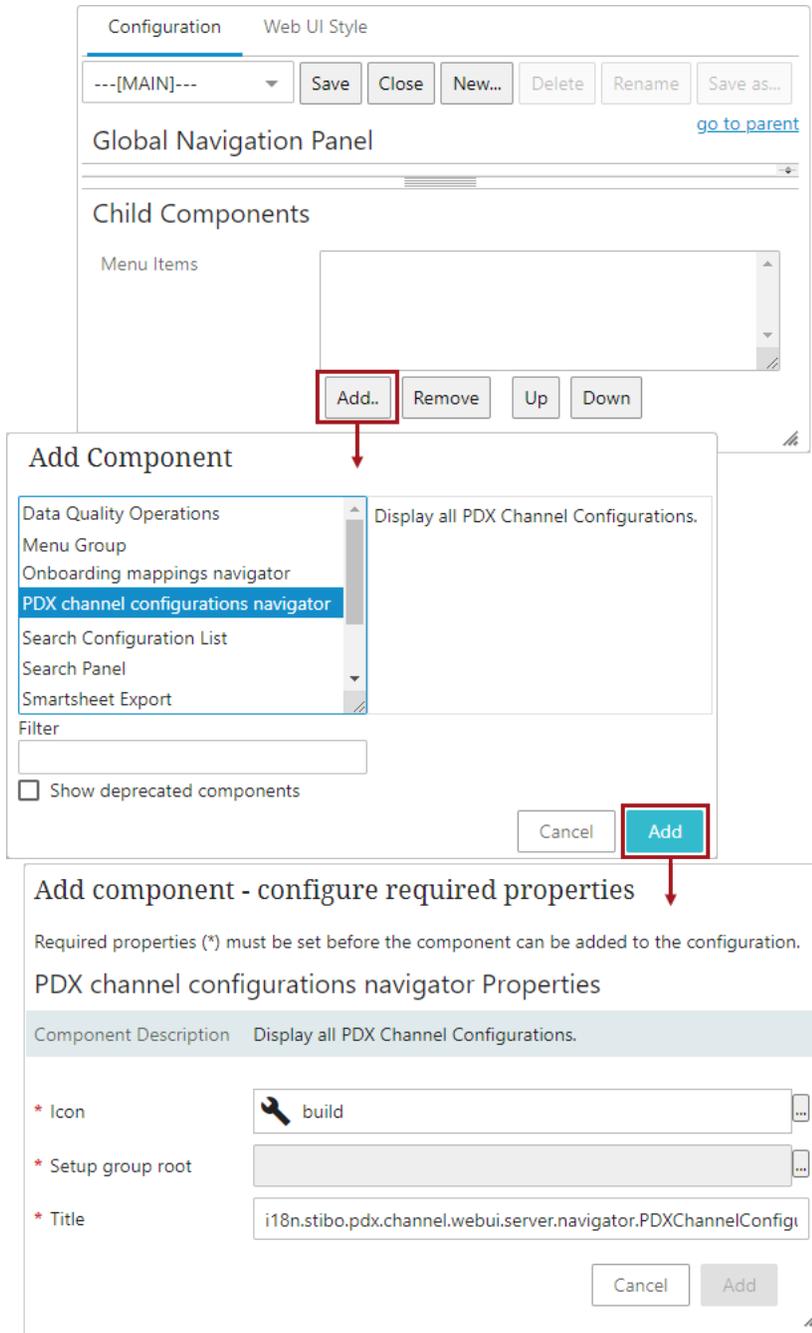
- Configure a Global Navigation Panel that allows access to the PDX Onboarding Channel Configurators.
- The configurator's object type (this object type serves as a container for the PDX Onboarding Channel Configurator setup entity within the system) must be configured to display the PDX Onboarding Channel Configurator screen when a PDX Onboarding Channel Configurators is selected.

Configuring the Global Navigation Panel

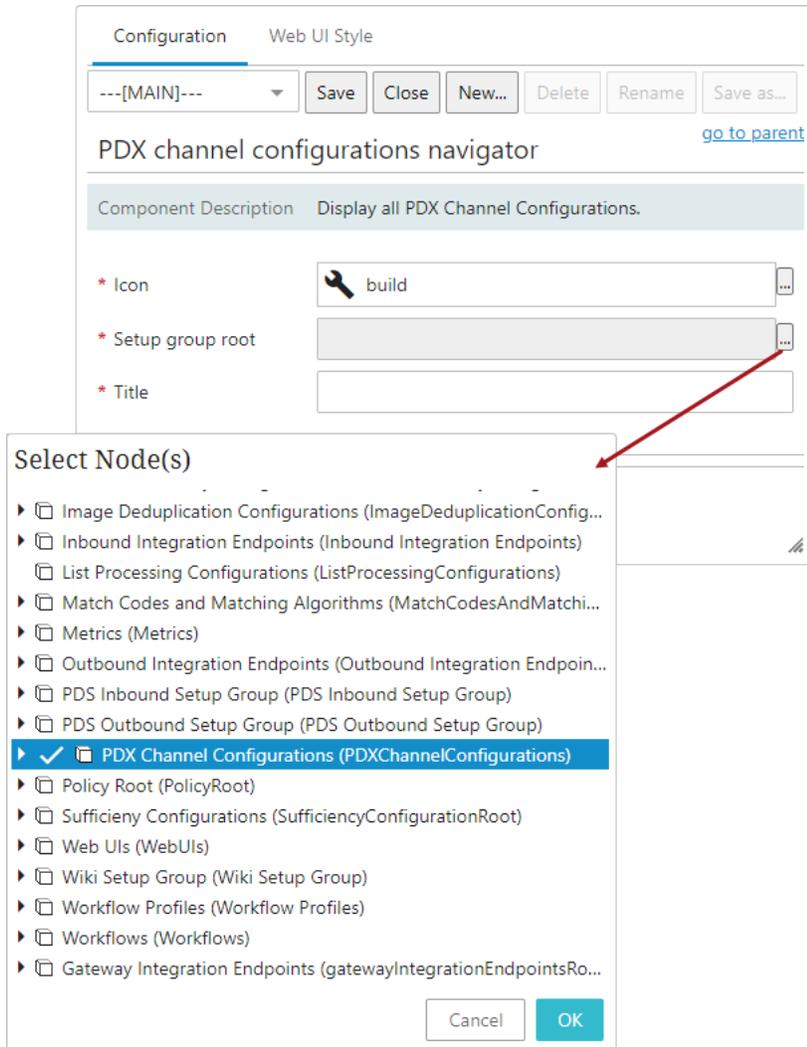
1. From the **Main Properties** screen, add a 'Global Navigation Panel' to the 'Left' section of the child components. Once added, double-click the Global Navigation Panel component. For Information on how to add the Global Navigation Panel, refer to the **Global Navigation Panel** topic in the **Web User Interfaces** documentation of the **STEP Online Help**.



- From the Global Navigation Panel Properties, select 'Add' in the Menu Items field of the Child Components section. Select 'PDX channel configurations navigator' from this dialog and then click 'Add.' The 'PDX channel configurations navigator properties' designer screen will be displayed.



3. In the 'PDX channel configurations navigator Properties,' click the ellipsis button (...) next to the 'Setup group root' parameter. Then select the **PDXChannelConfigurations** node to define the setup group root for channel configurators. Click 'OK' to add.

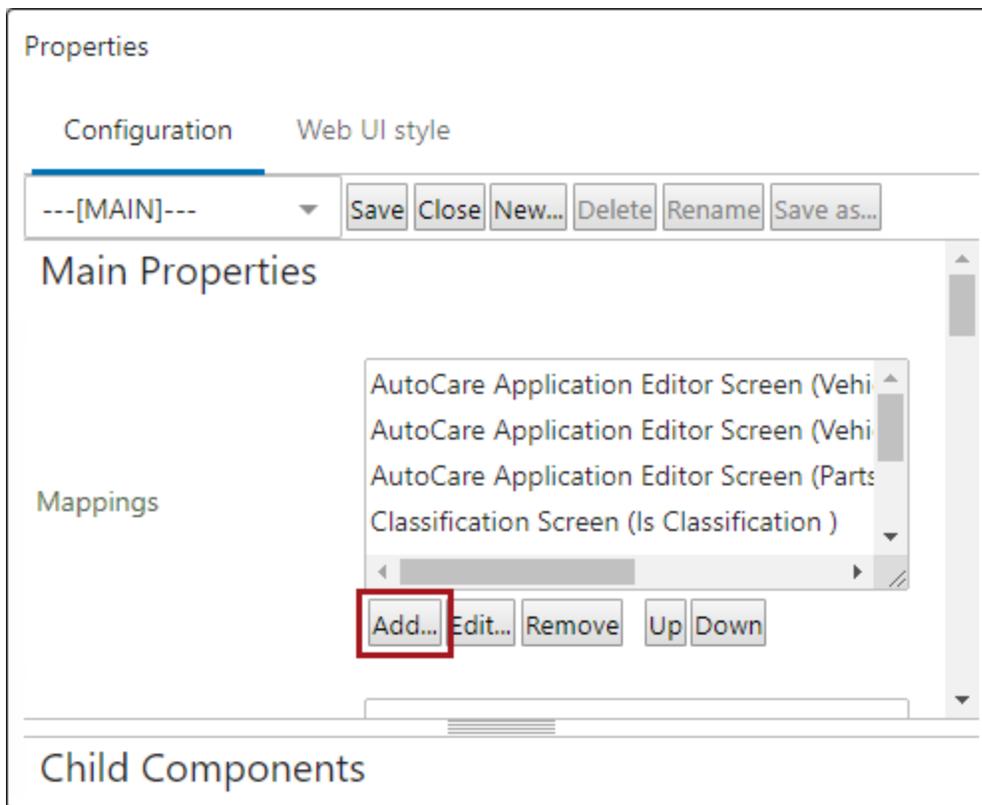


4. If required, type in a suitable title within the Title parameter that displays the 'PDX channel configurations navigator' component in the Global Navigation Panel. If left blank, the default value is set as 'PDX Configuration.'

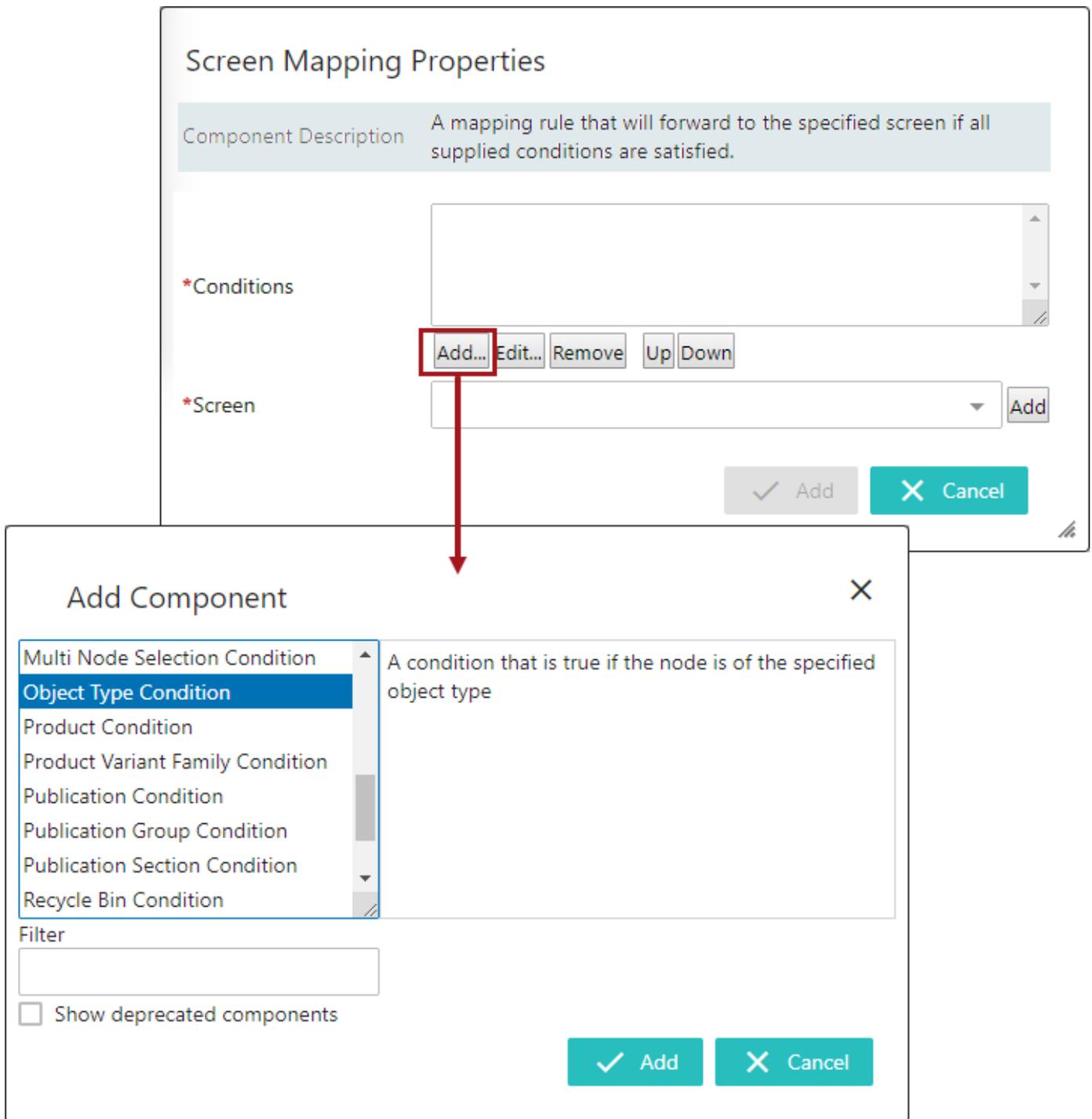
When finished, click **Save** in the designer to save the current settings. Do not close the designer unless you have finished configuring all the other parameters (outlined in the rest of this topic).

Configuring the PDX Onboarding Channel Configurator to Display PDX Screen

1. From the Main Properties, click the 'Add...' button for the Mappings parameter, and the Screen Mapping Properties will display.



2. Add a condition by clicking the **Add...** button next to the 'Conditions' parameter, and the Add Component dialog will display.



3. Within the Add Component dialog, select the Object Type Condition component and then click **Add**. In the Object Type Condition Properties, click the ellipsis button (...) next to the Object Type parameter, expand Basic Object Types, and select the PDX Channel Configuration (PDXChannelConfiguration) object type to define that this object type should use the channel configurator screen. Click **OK** to add the object type. Then click **Add** to add the Object Type Condition.

Add component - configure required properties

Required properties (*) must be set before the component can be added to the configuration.

Object Type Condition Properties

Component Description A condition that is true if the node is of the specified object type

* Object Type

- In the Screen Mapping Properties, select the previously created Node details screen for configurators using the dropdown selector next to the 'Screen' parameter.

Configuration Web UI Style

---[MAIN]---

Screen Mapping

[go to parent](#)

Component Description A mapping rule that will forward to the specified screen if all supplied conditions are satisfied.

* Conditions

* Screen

- Click **Add**. The new mapping is added in the 'Mappings' field (as shown below). Click **Save** in the designer to save the current settings.

Configuration Web UI Style

---[MAIN]---

Main

Mappings

→ PDXConfigurationScreen (ObjectType = PDXChannelConfiguration) (Is User) Name and ID (Always True)

Configuring PDX Onboarding Channel Configurator Setup Entity

The PDX Onboarding Channel Configurator is a STEP setup object. It is stored as a STEP setup entity with object type 'PDX Channel Configuration.' Each PDX Onboarding Channel Configurator holds the definition of what the configurator should do.

Important: Creation of the PDX Onboarding Channel Configurator setup entity can be done in both the workbench and the Web UI, but further configurations can only be done in the Web UI and not in the Workbench.

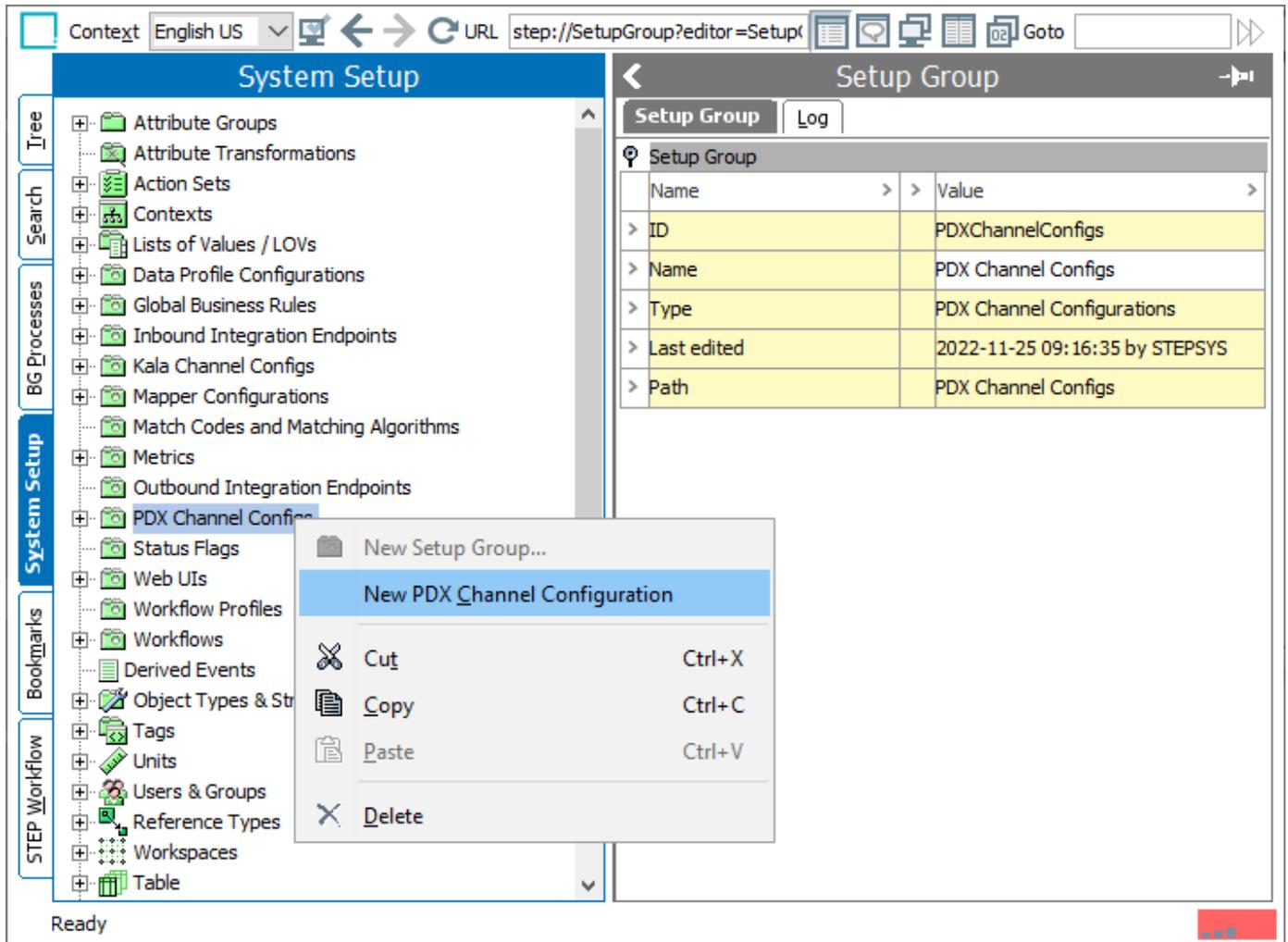
Prerequisites

A setup group must be created to hold the PDX Onboarding Channel Configurators. This is a one-time setup in the system. For more information about creating a setup group, refer to the **PDX Onboarding Channel Configurator Solution Initial Setup** topic within this section.

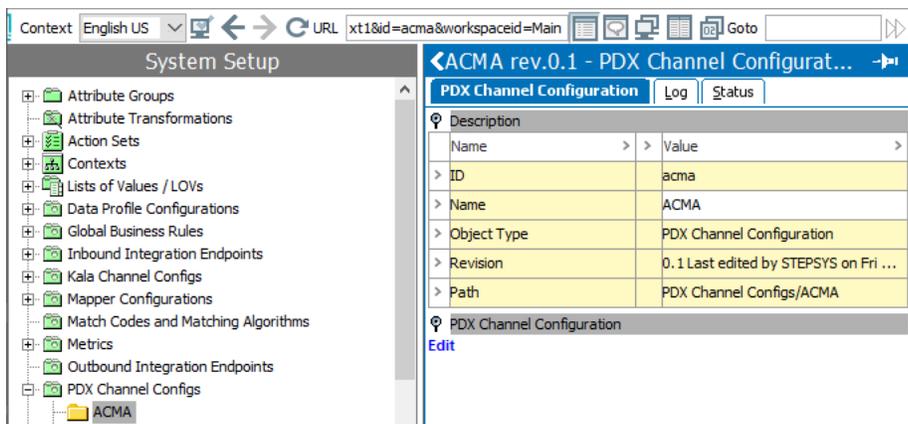
Since basic concepts for working with the designer are not covered in this section, a user configuring the PDX Onboarding Channel Configurator setup entity within a Web UI is expected to be familiar with the Web UI Designer. In addition, the user must have appropriate privileges to access the designer. For more information, refer to the **Designer Access** topic within the **Web User Interfaces** section of the **STEP Online Help**.

Creating PDX Onboarding Channel Configurator Setup Entity in Workbench

1. To create the PDX Onboarding Channel Configurator setup entity, right-click the PDX Channel Configs setup group root node found in the System Setup tab in the workbench. From the dropdown that displays, select New PDX Channel Configuration.



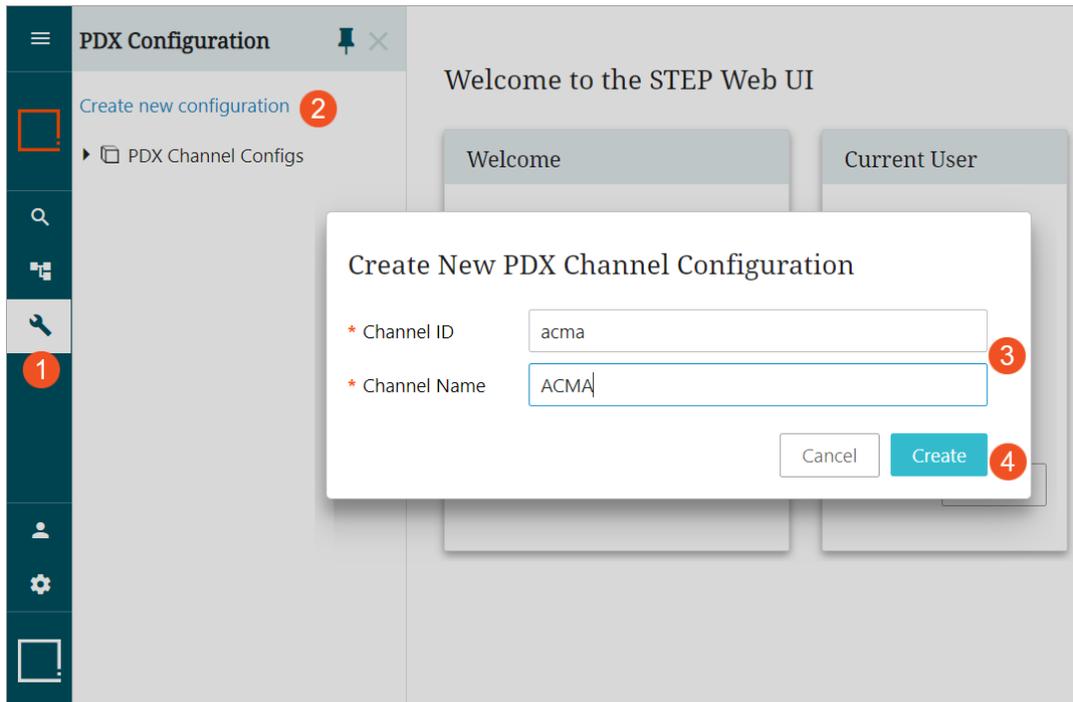
- In the dialog that appears, type in a suitable ID and Name, and then click 'Create.' In the screenshot below, a PDX Onboarding Channel Configurator named 'ACMA' is created under the setup group called 'PDX Channel Configs.'



Creating PDX Onboarding Channel Configurator Setup Entity in Web UI

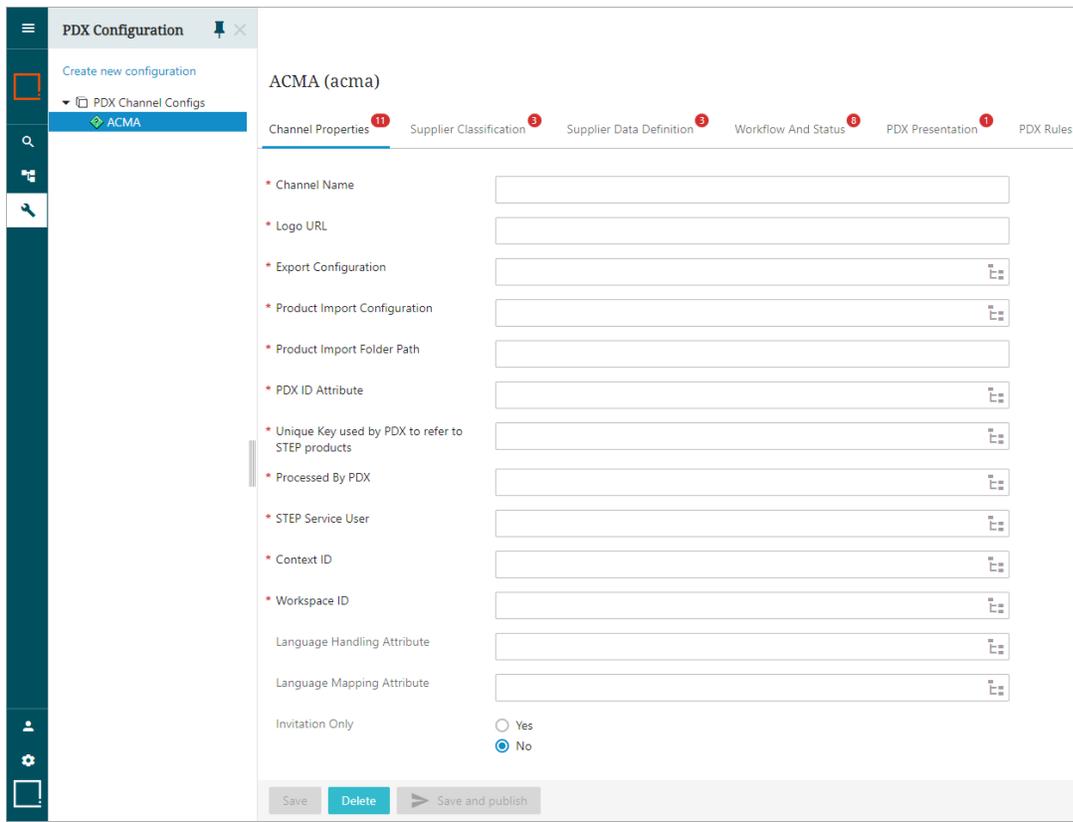
This section describes a detailed procedure on how to create a Onboarding Channel Configurator Setup Entity in Web UI. In the screenshot below, numbers have been added beside the various elements contained in the screen. Each numbered item in the steps below corresponds to the numbered element in the screenshot.

1. From the Global Navigation Panel, select the 'PDX channel Configurations Editor' component .



2. Click the 'Create new configuration' link, and the 'Create New PDX Channel configuration' dialog displays.
3. Type in a suitable ID and name for the channel in the Channel ID and Channel Name fields, respectively. In the example above, Channel ID and Channel Name fields are populated with the values 'acma' and 'ACMA,' respectively.
4. Click **Create** and the new PDX Onboarding Channel Configurator named 'ACMA' is created under setup group 'PDX Channel Configs'. The new channel configurator object displays in the Global Navigation Panel

(as shown below).



Important: If the PDX Onboarding Channel Configurator is deleted in the Web UI, it will be moved to the Recycle Bin folder in the System Setup tab in the workbench. If the user tries to create a new configurator with the same Channel Name value as was added to the recently deleted configurator, then the user will get the Unexpected error. If users want to reuse the same name for a new configurator that was used in the deleted configurator, then they must either purge the deleted PDX Onboarding Channel Configurator setup entity from the Recycle Bin, or create a PDX Onboarding Channel Configurator with a different Channel Name, click Create, then go back and change the Channel Name to what the user requires.

Configuring PDX Onboarding Channel Configurator Setup Entity in Web UI

When a new PDX Configurator is first created and selected in the Web UI, it displays six tabs containing multiple empty parameters to be defined. For ease of understanding, each tab and the parameters available within them are explained as separate topics within this section. The following are the six tabs available for configuration:

- **Channel Properties:** This section outlines the essential characteristics of the channel, encompassing its name, logo, preset context, workspace, schema export configuration, product import configuration, etc.

- **Supplier Classification:** In this section, you will find the information necessary for identifying supplier-related structures within the PDX channel. These structures are used when submitting products to STEP to ensure the correct supplier classification and reference type are assigned to each product.
- **Supplier Data Definition:** This section provides a comprehensive definition of the products' data and data structures that suppliers will use when onboarding their products through the PDX Channel. The parameters encompass product-related attributes, packaging details, assets, and also regulate the data model accessible to PDX users.
- **Workflow And Status:** In this section, you will find the essential components necessary for constructing the interaction flow between STEP and PDX. These include event queues, product status definitions, pertinent attributes, and attribute groups, among others.
- **PDX Presentation:** This section outlines several key parameters that affect the PDX user interface (UI). For instance, one of these parameters determines the order in which attributes are displayed in the PDX UI.
- **PDX Rules:** In this section, you will find the configurations required to facilitate diverse forms of data validation in PDX. This encompasses attribute definitions that are mandatory or become read-only once submitted, as well as configurations for the product data validation engine.

Channel Properties Tab

The Channel Properties tab of the PDX Onboarding Channel Configurator solution encompasses several parameters that define various attributes of the PDX onboarding channel. This tab defines basic channel properties, including channel name, channel logo image, default context, default workspace, schema export config, product import config, etc.

ACMA (acma)

Channel Properties ¹¹

Supplier Classification ³

Supplier Data Definition ³

* Channel Name	<input type="text"/>
* Logo URL	<input type="text"/>
* Export Configuration	<input type="text"/> 
* Product Import Configuration	<input type="text"/> 
* Product Import Folder Path	<input type="text"/>
* PDX ID Attribute	<input type="text"/> 
* Unique Key used by PDX to refer to STEP products	<input type="text"/> 
* Processed By PDX	<input type="text"/> 
* STEP Service User	<input type="text"/> 
* Context ID	<input type="text"/> 
* Workspace ID	<input type="text"/> 
Language Handling Attribute	<input type="text"/> 
Language Mapping Attribute	<input type="text"/> 
Invitation Only	<input type="radio"/> Yes <input checked="" type="radio"/> No
Is Public Channel	<input type="radio"/> Yes <input checked="" type="radio"/> No

The available parameters within the Channel Properties tab are as follows:

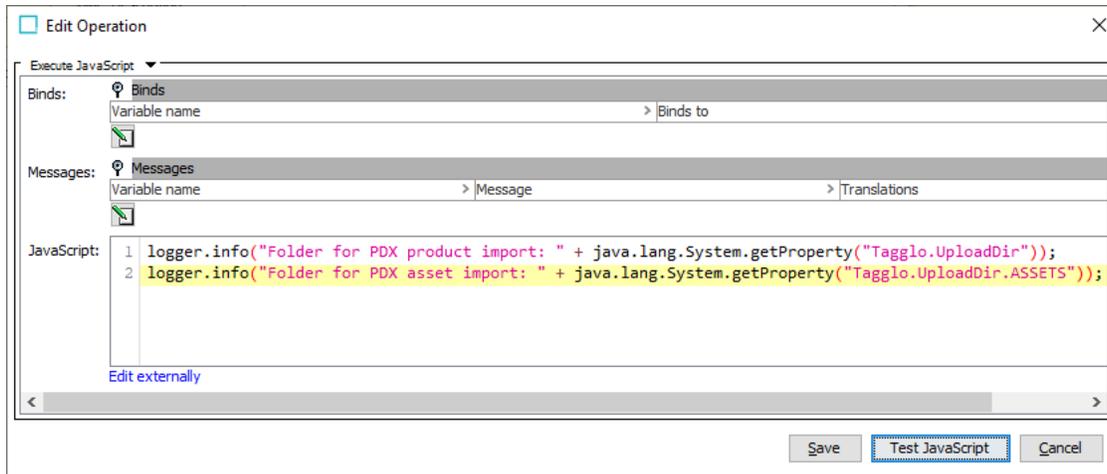
- **Channel Name:** The name of the PDX onboarding channel that is displayed on the PDX User Interface (UI). This can be changed after the channel creation.
- **Logo URL:** The parameter requires a URL that grants PDX access to download the channel logo image. It is worth noting that logos rendered in vector format, such as SVG, are optimal, as they retain quality even when resized. For information about the recommended practices for creating a logo, refer to **Recommended Practices to Create Channel Logo** topic within this guide.
- **Export Configuration:** This feature is used to export PDX related data model from PMDM, including Product hierarchies, attribute groups, attributes, LOVs, Reference types, UOMs, Entities, etc. An automated and scheduled (Jenkins) job on the PDX side can trigger an export BGP to retrieve PDX related data model from PMDM. This occurs when a channel is created for the first time and for ongoing PDX channel data standard refresh.
- **Product Import Configuration:** This parameter will be configured with an Import Configuration that is used to import product data into STEP that is submitted from PDX by vendors. When a product data submission occurs in the PDX channel, PDX uploads STEPXML file(s) to the STEP server and triggers an import BGP via API calls. Usually, a business rule is needed to send the imported products into an onboarding workflow for internal user review. The products might be rejected, returned for vendor reworking, or become approved. All product statuses, along with the necessary message to vendors, can be sent back to PDX through workflow integration. For more information, refer to the 'Workflow And Status Tab' topic.

If assets are also collected from vendors via PDX, there will be a separate import configuration for the asset.

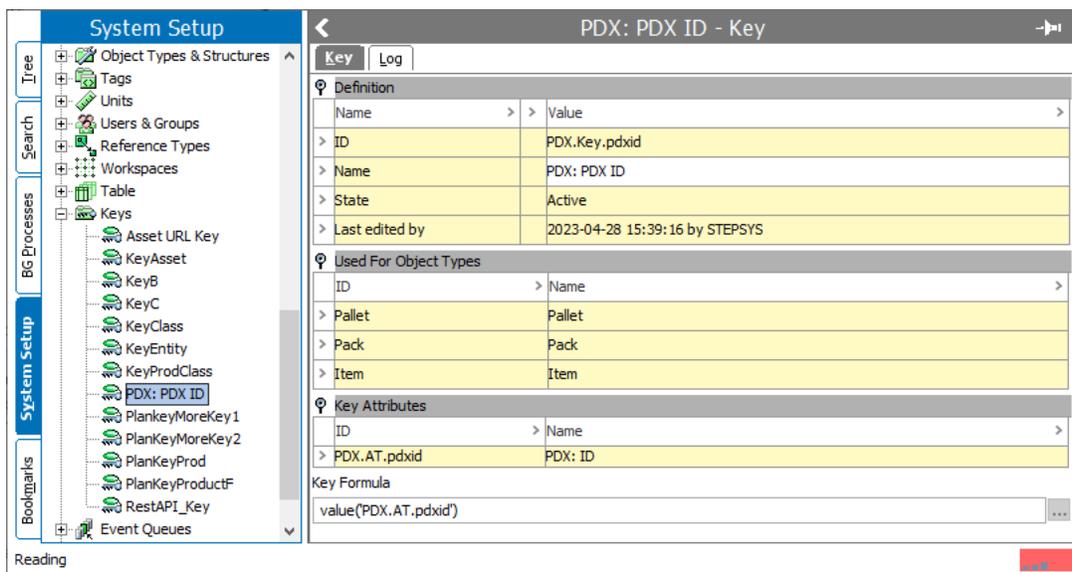
- **Product Import Folder Path:** This parameter is to set up a file folder on the STEP application server to which PDX uploads product STEPXML files when a submission occurs. The folder for uploading asset files (STEPXML file and/or asset binary files) shall be separate.

This folder needs to be the same as the configuration set in the 'The Standalone.JVMArgs' property of the Tagglo component, which is located in the 'sharedproperties.config' file under the 'Tagglo.UploadDir' section.

The below business rule can be used to check the status of both the upload folders.



- **PDX ID Attribute:** This parameter must be configured with an attribute valid on the product object type (and packaging and family object types if relevant). It is used to store PDX product ID. The product's unique key will be based on the value in this attribute to ensure its uniqueness.
- **Unique Key used by PDX to refer to STEP products:** This parameter is to be configured with the STEP key that is used by PDX when submitting or retrieving products to/from PMDM. It is important to note that the selected key should be based on the PDX ID Attribute chosen above. For instance, refer to the example provided below.



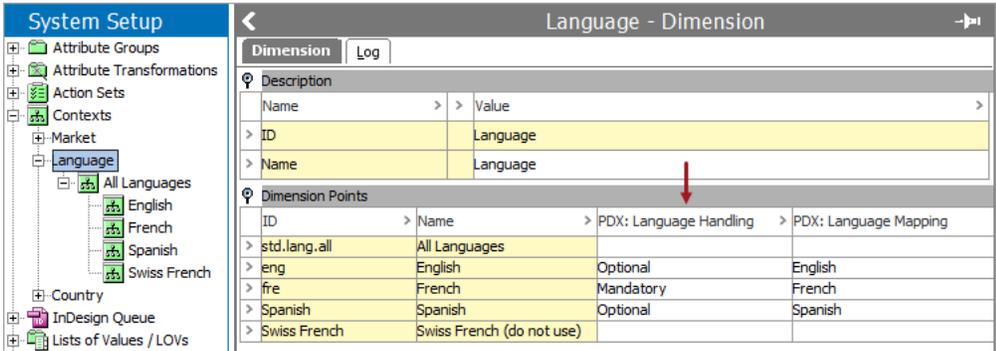
- **Processed By PDX:** This parameter is to be configured with an attribute that is valid on the product object type (also packaging and family object types if relevant) and is used to indicate that a product has been processed by PDX. This attribute is solely for reporting purposes in PMDM and does not drive any other logic.
- **STEP Service User:** The parameter presented herein pertains to the identification of a STEP user, who will be utilized by PDX for issuing STEP API calls. These calls facilitate data standard exports and the monitoring

of the Status Event queue, as displayed in the Workflow and Status Tab. It is important to note that the user assigned to this parameter should possess super user privileges.

- **Context ID:** The users can choose a context using this parameter. PDX uses this as the default context into which vendors will submit products. The language dimension point of this context will be designated as the default language for the corresponding PDX channel.
- **Workspace ID:** This parameter designates the workspace where users will submit their products in PDX. It is recommend setting this parameter to 'Main' as the default option.
- **Language Handling Attribute:** This parameter is for multiple language support. This parameter is to be configured with an attribute that controls the availability and requirement of input language in the PDX channel. The attribute must be valid on Dimension Points in STEP (as shown below) and be LOV-based with the following value IDs: 'Mandatory' and 'Optional.'

If a Language holds the attribute value as 'Mandatory' in STEP, the corresponding language layer in the PDX channel must be completed for a product before it can be submitted to STEP.

On the other hand, if a Language holds the attribute value as 'Optional' in STEP, then it is optional for the vendors to add data to the corresponding language layer in the PDX channel. However, if any data have been added to a language layer on a product, the language layer must be completed before the product can be submitted. Further, if a Language is configured with an empty value or with the ValueID 'No' in STEP, it will not be added to the channel in PDX.



Language - Dimension			
Description			
Name	Value		
ID	Language		
Name	Language		
Dimension Points			
ID	Name	PDX: Language Handling	PDX: Language Mapping
> std.lang.all	All Languages		
> eng	English	Optional	English
> fre	French	Mandatory	French
> Spanish	Spanish	Optional	Spanish
> Swiss French	Swiss French (do not use)		

- **Language Mapping Attribute:** This parameter is for multiple language support. It is to be configured with an attribute that controls the name and icon of the corresponding language layer in the PDX UI. For example, the language dropdown in the channel grid view in PDX (as shown below). The attribute must be valid on Dimension Points in STEP and be LOV based.

ACME

root Select Account

<input type="checkbox"/>	NAME	STATUS	DESCRIPTION	MULTILANGUAGE DESCRIPTION	PRICE
<input type="checkbox"/>	Aloe vera hand cre...	Progress	The benefits of seaweed are in this...	Other Gold	15.00
<input type="checkbox"/>	Black metal polish	Accepted	This is black, shiny, and opaque nail...		15.00
<input type="checkbox"/>	Jasmine Face cream	Accepted	This face cream is jasmine-scented.		15.00
<input type="checkbox"/>	Nectarine Face cre...	Progress	The benefits of seaweed are in this...	Gold	15.00
<input type="checkbox"/>	Peppermint hand c...	Progress		Citrus	15.00
<input type="checkbox"/>	Limoncello hand cr...	Progress	The benefits of seaweed are in this...	Citrus Gold	15.00
<input type="checkbox"/>	Papaya Face cream	Progress	Extracts of papaya are in this face...		15.00
<input type="checkbox"/>	Rose hand cream	Progress	Rose scented hand cream	Floral	15.00

EN English en

EN English en

English (United States) en-US

English (United Kingdom) en-GB

French fr

French (Canada) fr-CA

Spanish es

Spanish (Spain) es-ES

USD

- **Invitation Only:** The options available within this parameter determines whether access to the channel is restricted to an invite-only flow.
- **Is Public Channel:** The options available within this parameter determines whether the channel will be visible as a public channel in the PDX Channel management UI.

Recommended Practices to Create Channel Logo

This topic explores some of the recommended practices to create an effective channel logo that can help the channel to be identifiable easily.

In the PDX platform, the channel logo is displayed in a rectangular-shaped logo placeholder with specific dimensions. The recommended size for the logo placeholder is 320.99 pixels in width and 55.99 pixels in height, resulting in an aspect ratio of 5.741:1.

To ensure that the logo works well in the PDX user interface, it is important to follow some guidelines.

For example, a logo can consist of two parts: a favicon and a wordmark. This two-part design works well in a rectangular-shaped placeholder.



Basic logo guidelines



- Ensure that there is no margin around the logo placeholder.
- Avoid using a background color for the logo.
- A vector SVG format is recommended as it can be scaled to any size without losing quality.
- If using a PNG format, compress and resize the logo using an optimizer tool.

Wordmark guidelines

STIBO SYSTEMS
MASTER DATA MANAGEMENT

- Select a font that is easy to read, even at smaller sizes
- Avoid decorative or complicated fonts that are difficult to read.
- For longer logos, it is recommended to use a rectangular shape rather than a square one
- Ensure that there is no additional padding around the wordmark

Favicon guidelines



- Favicon logos should be designed with simplicity in mind, as they are small and have limited space for detail.
- There should be no padding around the Favicon
- If the Favicon logo is square in shape, it may be centered within a square or circular shape

Supplier Classification Tab

The Supplier Classification tab within the PDX Onboarding Channel Configurator encompasses various parameters that define supplier-related structures in the PDX channel. This tab specifies the necessary information for establishing supplier classifications in PDX, which is crucial when submitting products to STEP. This ensures the correct linkage of products with the appropriate STEP supplier classification object type and reference types.

ACMA (acma)

Channel Properties 11
Supplier Classification 3
Supplier Data Definition 3

* Supplier Classification Object Type

* Supplier Products Classification Object Type

* Supplier Products Reference Type

Locations

Location Object Type

Location Root Object Type

Location Label

Is Active Location

Location Reference Type

Primary Location Reference Type

Location Key

Location Attribute

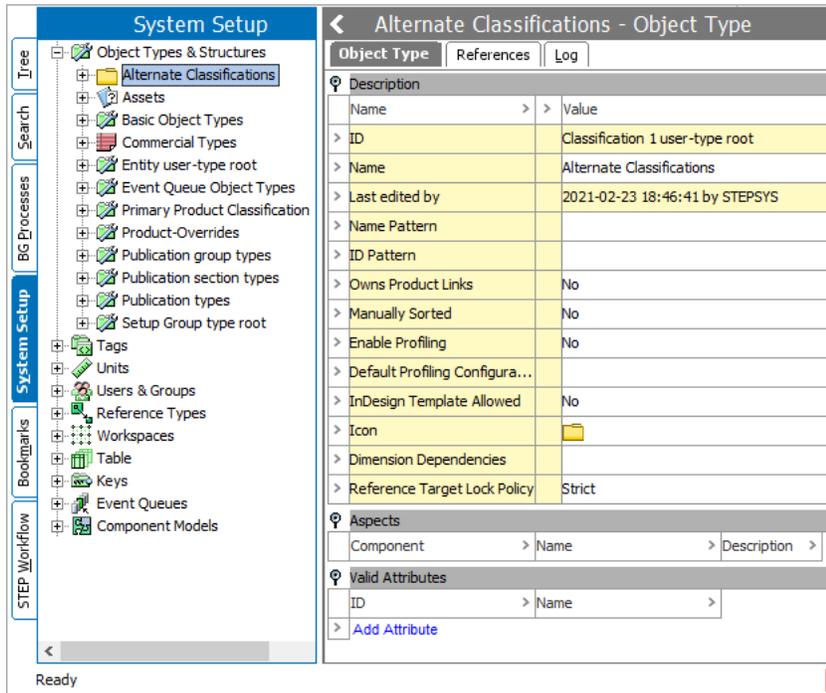
Primary Location Attribute

Save
Delete

 Save and publish

The Supplier Classification tab offers the following parameters:

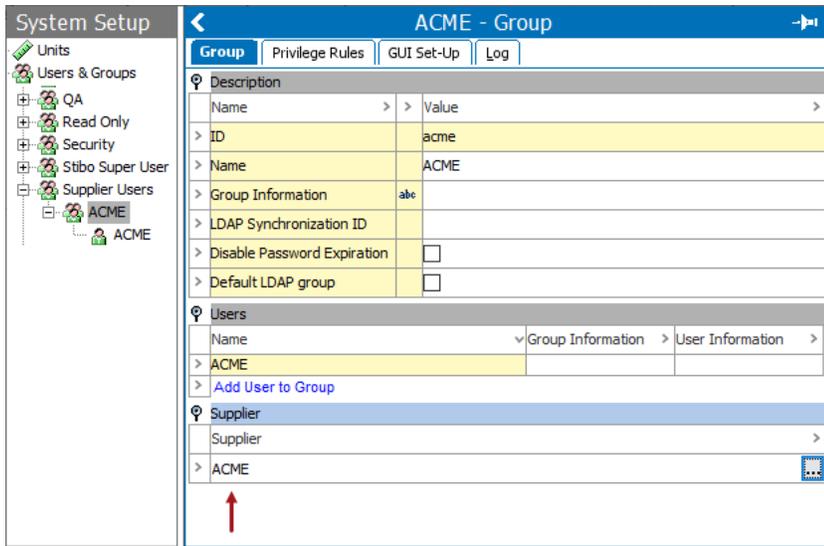
- **Supplier Classification Object Type:** This field is to be configured with the classification object type that is used to model suppliers in STEP. To select the object type, click on the node picker icon adjacent to the parameter.



- **Supplier Products Classification Object Type:** This parameter allows users to configure the classification object type used to store products that fall under the supplier object defined in the Supplier Classification Object Type parameter above. Users can find and select the appropriate object type by clicking on the node picker icon located next to the parameter.
- **Supplier Products Reference Type:** This parameter enables users to configure the product-to-classification link type that references the product to the Supplier Products Classification. Users can find and select the appropriate reference type by clicking on the node picker icon located next to the parameter.

Supplier User Group Set Up

In addition to the supplier classifications mentioned above, it is necessary to set up supplier user groups for every vendor in order to onboard products from a PDX using the onboarding channel.



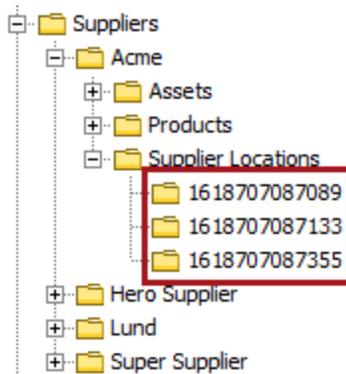
To authenticate a PDX client to a specific onboarding channel in PDX, a supplier user should be set up under each supplier's user group. The supplier needs access to the username and password for this purpose. It is important to note that if the Supplier User Group ID is different from the linked Supplier Classification ID, the supplier will also need the Supplier Classification ID, as that will have to be added to the 'ID of Supplier in STEP system' field in PDX.

Locations

The parameters in this canvas are used to enable and configure the use of allowed locations (shipping points) in the Onboarding channel. Taking this functionality into use is optional.

To use the allowed locations (shipping points) in the Onboarding channel, it is necessary to create a separate 'Supplier Location' classification in STEP for each supplier location provided by the supplier.

In the screenshot below, for instance, there are three locations (GLNs) stored in the 'Supplier Locations' node.



Once this functionality is activated and fully configured, the supplier can select one primary location from a list of active / allowed locations for the supplier and add additional locations to the 'Locations' attribute as secondary locations.

When a product is onboarded from PDX into STEP, the 'Primary Location' attribute will be used to create a reference from the product to the 'Supplier Location' object for which the location has been set, while the 'Secondary Location' attribute locations will be used to create references from the product to those 'Supplier Location' objects.

- **Location Object Type:** This parameter allows users to configure an object type. The selected object type should be the 'Supplier Location' classification object type. A 'Supplier Location' object has to hold the following attributes:
 - Location Label: an optional field used by the supplier to identify the specific location from which products are shipped to the retailer.
 - Is Active: if set to 'Yes', the retailer can accept products delivered from the location (shipping point). If set to 'No', the location is invalidated, and a supplier cannot set the location on a product in PDX to this location.
- **Location Root Object Type:** This parameter allows users to configure an object type. The selected object type should be the parent classification object type used to store 'Supplier Location.'
- **Location Label:** This parameter allows users to configure an attribute. The selected attribute identifies the attribute used as 'Location Label'. An attribute handling GLN values would be suitable for this field.
- **Is Active Location:** This parameter allows users to configure an attribute. The selected attribute identifies the attribute used as 'Is Active', (referenced above).

The attribute should be LOV based with allowed values "No" (Value ID=N) and "Yes" (Value ID=Y).

The list of active locations for a supplier is built in PDX. By collecting all 'Supplier Location' objects having Is Active != No for the supplier (with '!=' meaning 'does not equal'). The list collects all 'Supplier Location' objects with either the value 'Yes' for the 'Is Active' attribute, or no value at all. This list is generated when the channel is added in PDX and refreshed each time the supplier logs into PDX. No Data Standard update is required to ensure location changes are up to date in PDX.

- **Location Reference Type:** This field contains a product-to-classification link type that is used to establish a reference from the product object type to the above defined supplier location. The reference type for this should be multivalued and have the correct source (product object type) and target (location object type).
- **Primary Location Reference Type:** This field contains a product-to-classification link type that is used to establish a reference from the product object type to the above defined supplier location. The reference type for this should be multivalued and have the correct source (product object type) and target (locations object type).
- **Location Key:** This parameter can be configured with a key to uniquely identify locations. This key should be valid on the Location Object Type. For instance, if the Location Label attribute is used to carry a GLN value, the Location Key could be based on the same attribute selected as the Location Label.
- **Location Attribute:** This parameter allows users to configure an attribute. The selected attribute must be valid on the Products Object Type where it will be used to collect all secondary locations. This attribute will only be visible in PDX, if it has been made visible via the Attribute Groups parameter.
- **Primary Location Attribute:** This parameter allows users to configure an attribute. The selected attribute

must be valid on the Products Object Type where it will be used to collect the primary location. This attribute will only be visible in PDX, if it has been made visible via the Attribute Groups parameter.

Supplier Data Definition Tab

The Supplier Data Definition tab within the PDX Onboarding Channel Configurator offers multiple parameters that define data properties provided by suppliers through PDX. These properties relate to products, packaging products, and assets.

Node Details

Channel Properties ¹¹ Supplier Classification ³ **Supplier Data Definition ³** Workflow And Status ⁸ PDX Presentation ¹ PD

General

* Product Object Types

* Product Hierarchy Root

* Attribute Groups ×
Add

Hidden Groups ×
Add

Assets

Asset Import Configuration

Asset Import Folder Path

Supplier Assets Classification Object Type

Asset Relevance

Maximum Number of References

Category Mandatory References Attribute

Product To Asset Reference	Asset Object Type	+
<input type="text"/>	<input type="text"/>	<input type="text"/>

Asset File Name

Asset URL attribute

Asset Content Hash

Asset Validation Object

Asset Validation Object Type

Asset Object Types

Valid Color Spaces

Minimum Pixel Width

Minimum Pixel Height

Maximum Pixel Width

Maximum Pixel Height

Minimum Pixels (Longest Side)

Maximum File Size Bytes

Aspect Ratio

Families

Family Indicator Attribute

Family ID Attribute

Within the General canvas of the tab, users can find the following parameters:

- **Product Object Type:** This parameter enables users to configure the product object type utilized for exchanging product data between PDX and PMDM. Users can select the desired object type by clicking on the adjacent node picker icon.
- **Product Hierarchy Root:** This parameter permits users to select the top root folder of the product hierarchy, which will subsequently become the PDX category root. Users can find and select the product root folder by clicking on the node picker icon located next to the parameter.
- **Attribute Groups:** This parameter allows users to configure an attribute group that stores all vendor-facing product attributes. This attribute group can have multiple child attribute groups. Users can find and select the appropriate attribute group by clicking on the node picker icon located next to the parameter. Vendor users can view all the attributes in the group, grouped by sub-groups in PDX UI channel view, and can edit them unless attributes are defined otherwise. Users can add multiple attribute groups by clicking on the Add link provided below the parameter field.
- **Hidden Groups:** This parameter enables users to configure an attribute group that contains the attributes to be excluded from vendors. This makes it possible to create vendor facing attribute groups (used in the Attribute Groups parameter), that have exception attributes that need to be hidden from vendors.

Assets

The parameters available within this canvas are relevant if the PDX channel should support the submission of assets.

Once the asset functionality is activated, users can perform the following configurations in STEP:

- Specify the asset attributes that will be available to vendors. This is achieved via a configuration on the asset references in STEP.
- Determine the mandatory assets
- Define the structure of Asset Object Types and Asset Reference Types that vendors will be able to create via the channel.
- Control how assets are sent from PDX to STEP

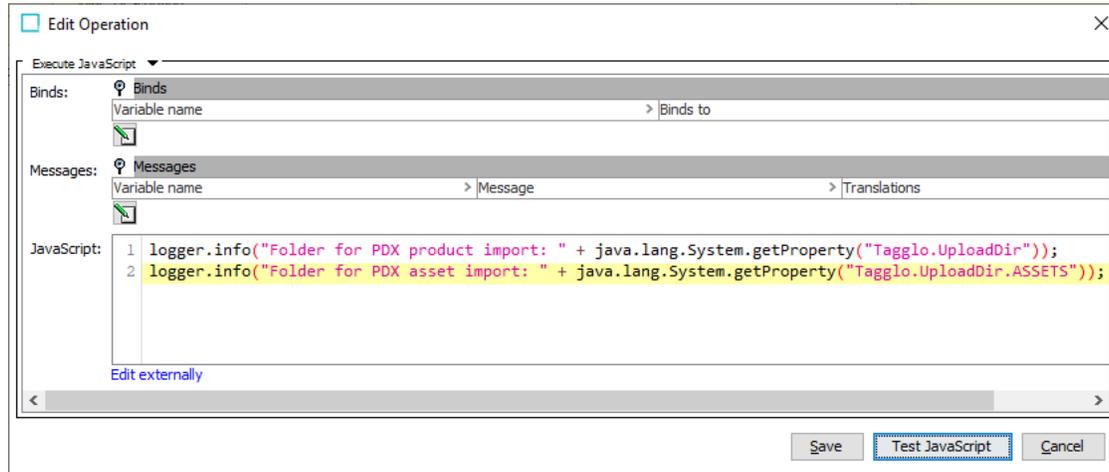
The relevant parameters are:

- **Asset Import Configuration:** This parameter allows users to configure the import configurations that will be used when importing assets. Click the node picker icon next to the parameter to find and select the import configuration. If the parameters 'Asset URL attribute' or 'Asset Content Hash' are used, then it is expected that some custom business rules are added to this import configuration.
- **Asset Import Folder Path:** This field is to be configured with a file folder available on the STEP application server that is designated for the upload of Asset STEPXML files and optional binary assets submitted by

PDX. The upload folder for product files is distinct and separate from this folder.

Note: This folder needs to be the same as the configuration set in the 'The Standalone.JVMArgs' property of the Tagglo component, which is located in the 'sharedproperties.config' file under the 'Tagglo.UploadDir.ASSET' section.

The below business rule can be used to check the status of both the upload folders.



- **Supplier Assets Classification Object Type:** This parameter is to be configured with an object type of the classification folder that is used to store assets for a supplier. To link assets to the Alternate Classification, the Reference Type must be configured in the 'Asset Reference Type Map' parameter as described below.
- **Asset Relevance:** To determine which asset attributes are to be displayed on products in PDX, this parameter must be set with a metadata attribute that is valid for the asset reference. The attribute should be LOV-based and have the following values:
 - 'Yes' (ValueID 'Y') which directs the system to display assets of that type on the product.
 - 'Primary', (ValueID 'Primary') which directs the system to display an asset of that reference type as the primary image.
 - 'No' (ValueID 'N') to indicate that the asset reference is not relevant for suppliers and will not be displayed in PDX.
 - Within the STEP system, the attribute value is maintained on the asset reference types.

If the attribute configured in this parameter is left blank, then the system assumes the value as 'No'.

To set up which image is displayed for products in the 'List' and 'Grid' views on the Product Summary page in PDX, it is essential to ensure that only one asset reference type has the 'Asset Relevance' attribute set to 'Primary'. This is because the asset reference with this setting will be used to provide the image asset that displays as the 'Primary Image' in PDX. If more than one asset reference type has the 'Asset Relevance' attribute set to 'Primary,' the system will randomly choose one to display as the Primary Image in PDX.

- **Maximum Number of References:** This parameter is to be configured with a metadata attribute that can restrict the number of references allowed on a product for a given asset reference type by setting a maximum on the asset reference type.
- In the example below a 'PDX: Maximum Number of References' attribute has been configured in 'Maximum Number of References'. The admin user has then set this maximum to '8' on a vendor facing asset reference in STEP. This pre-set maximum will then be enforced in PDX. In practice, this means that a retailer can decide, for instance, that they want the supplier to provide no more than eight Product Images per submitted product, and the system enforcement means suppliers must abide by this setting. Products that do not satisfy this rule cannot be submitted and the corresponding asset attribute will be marked with a red error.

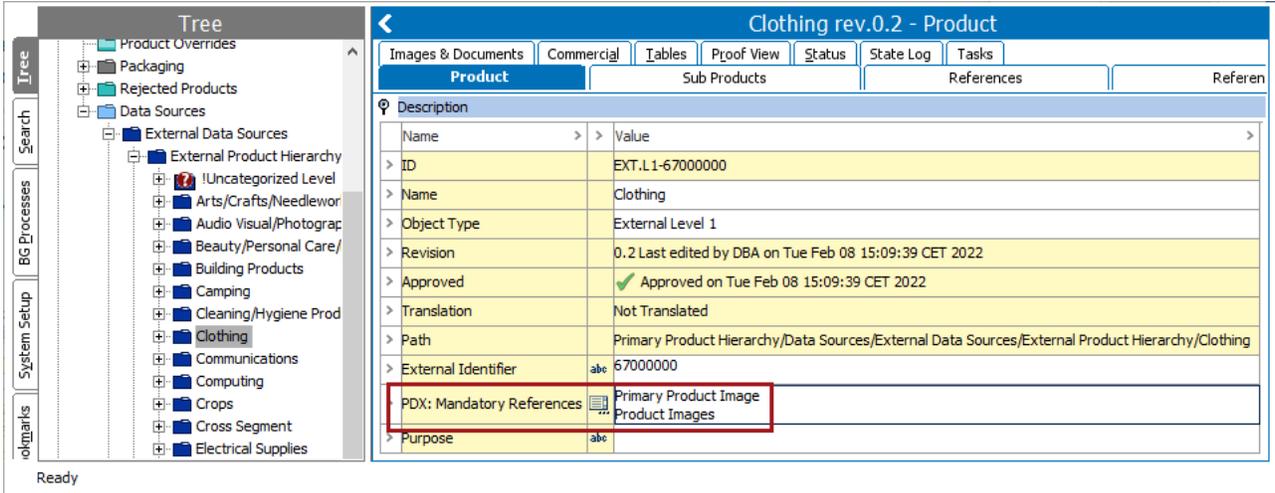
Product Images - Reference Type	
Reference Type	Validity Log
Description	
Name	Value
ID	PMDM.IDRT.ProductImages
Name	Product Images
Last edited by	2022-11-15 16:36:43.0 by PVMA
Externally Maintained	No
Dimension Dependencies	
Allow multiple references	Yes
Mandatory	No
Inheritance	Inherited
PDX: Description	abc
PDX: Maximum Number Of References	8
PDX: Relevance	Yes
Purpose	abc
In Attribute Groups	
ID	Name
PMDM.ATG.BuyerApprove	Buyer, Approve
PMDM.ATG.BuyerCreate	Buyer, Create
PMDM.ATG.BuyerDelete	Buyer, Delete
PMDM.ATG.BuyerModify	Buyer, Modify
PMDM.ATG.BuyerView	Buyer, View

- **Category Mandatory References Attribute:** This parameter is to be configured with a attribute where the category-specific asset references that must be mandatory for users in PDX are controlled. This attribute is valid on the product category node itself. The attribute configured in this parameter should allow for multiple values and must contain the list of mandatory references on that category.

The attribute should be LOV based with these valid values:

- LOV Value = asset reference name
- LOV Value ID = asset reference ID

As an example, a retailer has chosen the 'PDX: Mandatory References' attribute as the Category Mandatory References Attribute. In addition, the retailer has decided that both 'Primary Product Image' and 'Product Images' asset references should be mandatory for suppliers in PDX in the 'Clothing' category. To implement this, the retailer has added both references to the 'PDX: Mandatory References' attribute on the 'Clothing' product category node, as depicted in the screenshot below.



- Asset Reference Type Map:** The ability to create a table within this parameter allows users to map Product To Asset Reference with the Asset Object Type. The parameter determines the Asset Object Type that PDX will use when a product with assets is submitted.

The first row in the example below defines that if an asset is added to the 'Primary Product Image' asset attribute on a product in PDX, the asset will be submitted to STEP as an asset of type 'Product Image.' Similarly with the next three rows.

The last row defines that if an asset is added to an asset attribute on a product in PDX, and if that asset attribute is not one of 'Primary Product Image', 'Documents', 'Product Video', or 'Certificate', then the asset will be submitted to STEP as an asset of type 'Product Image'

Product To Asset Reference	Asset Object Type	
Primary Product Image	Product Image	+
Documents	PDF	+
Product Video	Product Video	+
Certificate	Legal documents	+
	Product Image	+

- Asset File Name:** This mandatory field is to be defined with an attribute belonging to an asset. It is used to configure the attribute carrying the original file name of the asset.
- Asset URL attribute:** This optional field is to be defined with an attribute belonging to an asset. It is used to configure the attribute carrying the PDX URL of the asset. The attribute chosen for this field must be valid for all vendor-facing assets.

If left empty, the assets will be uploaded to STEP as binary asset files. However, if populated with a valid attribute, PDX will upload an asset XML file containing the asset URL. Then STEP can download the asset file.

To ensure the smooth functioning of this solution, it is recommended to download the asset as part of the asset import using a simple business action. The validity of asset URLs cannot be assumed indefinitely. PDX

guarantees the URL validity for a minimum of 7 days, after which the URL may expire. Additionally, the URL may vary between submissions and over time.

- **Asset Content Hash:** This optional field is to be defined with an attribute that is used to carry Asset Hash information and is intended to be used in combination with the Asset URL attribute. The attribute must be valid for all vendor-facing assets.

When this field is populated, the PDX will add an image hash as the value in the configured attribute sent to STEP. On the STEP side, this can be used in a business rule run by the asset importer to decide if this asset is already present in STEP and therefore does not need to be downloaded and imported again

Asset Validation Object

The parameters available within this canvas can be used to configure a range of asset validations. These are basically the rules that assets have to satisfy before they can be submitted to STEP.

For retailers with specific requirements for supplier-delivered assets linked to submitted products, PMDM supports the ability to validate assets in the PDX Channel. For instance, if a retailer requires product-linked image assets to be .jpg files, a validation can be applied in the parameters available within this canvas which ensures that image asset files submitted via PDX will be .jpgs.

Note: The asset validations are tied to the Asset Type in STEP, not the Asset Reference Type.

Described below is a list of asset validations that may be applied in STEP and enforced in PDX.

- **MIME type** – By applying a MIME type limitation, retailers can restrict submitted assets to only those with the desired MIME types. For instance, if a retailer wants all instruction manuals associated with the submitted product to be of the .pdf MIME type, the validation applied in the configured product data validation would be 'application/pdf'. This validation is applied directly on the object type, found under 'Object Types & Structures' on the 'System Setup' tab in workbench. In the example shown in the screenshot below, the object type is 'ProductImage'. In this case, the admin has determined that if images submitted for products are not of the MIME types listed in the 'MIME Types' field, the product cannot be submitted.

Product Image - Object Type	
Object Type	References
Description	
Name	Value
ID	ProductImage
Name	Product Image
Last edited by	2023-02-02 11:04:58 by DBA
Name Pattern	
ID Pattern	
Icon	
MIME Types	image/tiff image/tif application/postscript image/*
Dimension Dependencies	
Reference Target Lock Policy	Strict
Asset Keywords	abc
Asset URL Attribute	URL
PDX: Asset Content Hash	abc
PDX: Asset File Name	abc
PDX: Asset URL	URL

For more information on MIME types, refer to the **MIME Types** topic in the **Object Types and Structures** section of the **System Setup** documentation.

In order to use the validations below a dedicated entity type 'Asset Validation Object' with some special attributes, needs to be setup.

Asset validation objects are created to define specific asset validation rules on specific assets in PDX. The validations display like this on the asset validation object in STEP:

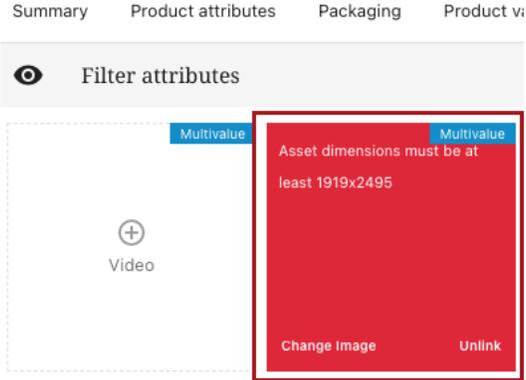
Product Images rev.0.1 - PDX: Asset Validation Object	
PDX: Asset Validation Object	References
Description	
Name	Value
ID	18648652
Name	Product Images
Object Type	PDX: Asset Validation Object
Revision	0.1 Last edited by BRCC on Fri Mar 17 23:47:01 CET 2023
Path	Entity hierarchy root/Configuration Objects/Asset Validation Objects/Product Images
Aspect Ratio	1:1
Asset Object Types	Product Image
Maximum File Size (Bytes)	
Maximum Pixel Height	4000
Maximum Pixel Width	4000
Minimum Pixel Height	1200
Minimum Pixels (Longest Side)	4000
Minimum Pixel Width	1200
Valid Color Spaces	RGB color

Each Asset validation object is tied to the asset object types that the validations should apply to via a dedicated attribute.

Listed below are the various validation settings that can be applied on the asset validation object, accompanied with a description of each:

- **Aspect Ratio** – Allows retailers to specify the aspect ratio (the ratio of an image’s width to its height) of the product-linked images submitted by suppliers via PDX. Select from the options available in the dropdown: 1:1, 2:33, 1:1, 2:3, 3:4, 16:9 and 16:10. If the submitted image does not conform to the specified aspect ratio, the product submission will be rejected, and an explanatory error message will display.
- **Maximum File Size (Bytes)** -- A product-linked image submitted via PDX cannot have a larger file size than the number of bytes set for this parameter.
- **Asset dimension (in pixels)** -- The size, in pixels, of product-linked image assets submitted by suppliers via PDX can be strictly controlled by setting a series of attributes on the asset reference type definition. Below is a list of these settings along with a description of the image aspect they control:
 - **Maximum Pixel Height** – The image height of submitted images can be no greater than the number (in pixels) set in this parameter.
 - **Maximum Pixel Width** – The image width of submitted images can be no greater than the number (in pixels) set in this parameter.
 - **Minimum Pixel Height** – The image height of submitted images can be no less than the number (in pixels) set in this parameter.
 - **Minimum Pixels (Longest Side)** -- The image width or height of submitted images, whichever is greater, can be no less than the number (in pixels) set in this parameter.
 - **Minimum Pixel Width** – The image width of submitted images can be no less than the number (in pixels) set in this parameter.

If an asset does not satisfy one of these configured rules, the corresponding product cannot be submitted and the asset will be marked with a red error, as shown in the screenshot below:



Additional information about the various settings that are needed on the asset validation object are described below:

- **Asset Validation Object Type:** This parameter is to be configured with an entity object type. This object type must be the object type used to store asset validation object.

- **Asset Object Types:** This parameter is to be configured with an attribute. The values of the selected attribute are the asset object types this asset validation object is relevant for. The attribute must be valid for the Asset Validation Object Type and should be a Multi-valued Lists of Values (LOV) of asset object types.
- **Valid Color Spaces:** This parameter is to be configured with an attribute that contains the color spaces that a digital asset of this type must have. The attribute must be valid for the Asset Validation Object Type and should be a Multivalued LOV of color spaces. The valid options are:
 - Bitmap
 - CMYK color
 - Grayscale
 - Indexed color
 - RGB color
- **Minimum Pixel Width:** This parameter is to be configured with an attribute where the values of the selected attribute contains the minimally required pixel width that a digital asset of this type must have. The attribute must be valid for the Asset Validation Object Type and should be of type integer.
- **Minimum Pixel Height:** This parameter is to be configured with an attribute where the values of the selected attribute contains the minimally required pixel height that a digital asset of this type must have. The attribute must be valid for the Asset Validation Object Type and should be of type integer.
- **Minimum Pixels (Longest Side):** This optional parameter is to be configured with an Entity attribute that defines the minimum pixel size of the longest side of the asset, i.e., the minimum pixel size of at least one of the asset edges.
- **Maximum Pixel Width:** This parameter is to be configured with an attribute where the values of the selected attribute contains the maximum allowed pixel width that a digital asset of this type may have. The attribute must be valid for the Asset Validation Object Type and should be of type integer.
- **Maximum Pixel Height:** This parameter is to be configured with an attribute where the values of the selected attribute contains the maximum allowed pixel height that a digital asset of this type may have. The attribute must be valid for the Asset Validation Object Type and should be of type integer.
- **Maximum File Size Bytes:** This parameter is to be configured with an attribute where the values of the selected attribute contains the maximum allowed file size a digital asset of this type may have. The attribute must the valid for the Asset Validation Object Type and should be of type integer.
- **Aspect Ratio:** This parameter is to be configured with an attribute where the values of the selected attribute contains the aspect ratios that a digital asset of this type must have. The attribute must the valid for the Asset Validation Object Type and should be a Multivalued LOV of aspect ratios.

Families

The parameters available within this canvas are mainly in relevance to the setup of family and variant related functionality.

Products in PDX can be grouped into product families that may include product variants. When those product variants are sent to STEP, they arrive tagged with a unique identifying attribute called a 'Family ID'. The value for this attribute, which is assigned to each variant product, enables users to recognize that these variants belong to the same product family in STEP.

Family attributes: Family attributes in PDX enable a structure through which an attribute on a product on the family level, defined as one level above the sellable products, can have a specific value that filters down to the products nested beneath that product. This helps ensure data consistency throughout the products in that family. When an attribute is defined as being a "family" attribute, validations in PDX are automatically applied that prevents users from submitting products for which the family attribute values differ between other products in that family. When a user submits a single product contained inside of a family, the PDX system automatically submits all products in that family. In this way, all products in the family are validated and submitted to STEP at the same time. To configure an attribute as being a 'family attribute', a Family Indicator Attribute must be set to 'Yes' on the attribute link between the product category and the desired attribute.

Variant Attributes: When product families are used, it is possible to define certain attributes as being variant attributes. When attributes are defined as variant attributes, validations in PDX are automatically applied that ensure that any combination of values assigned to a given variant product must be unique within the family. To define an attribute as a variant attribute for a specific section of the product hierarchy in PDX, the metadata attribute 'Variant Indicator' must be given a value: An integer must be assigned for that metadata attribute.

The following parameters are available within this canvas:

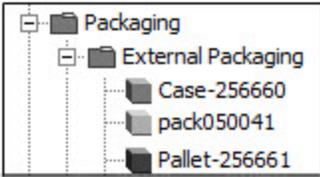
- **Variant Indicator Attribute:** This parameter is to be configured with an attribute. The content should be a meta data attribute on the attribute to products category link determining if an attribute is a Variant attribute. An attribute becomes a variant attribute if an integer has been assigned for that metadata attribute.
- **Family Indicator Attribute:** This parameter is to be configured with an attribute. The content should be a meta data attribute on the attribute to products category link determining if an attribute is a Family attribute. An attribute becomes a family attribute if the value 'Yes' is assigned for that metadata attribute. The attribute should be LOV based with allowed values 'No' (Value ID=N) and 'Yes' (Value ID=Y).
- **Family ID Attribute:** This parameter is to be configured with an attribute which will be used to store the PDX family identifier and can be used to group products into families in STEP.
- **Family Object Type:** This optional field is to be configured with a product object type that defines the STEP Family object type. If configured, this product object type will be used to create a classical 2 level parent / family-child / variant structure whenever a family is submitted from PDX to STEP rather than a flat structure. However, if not used, it is expected that all vendor facing attributes (regular, family, and variant) are valid on the Product Object Type.

If this option is used it is expected that all family attributes are valid for the STEP Family object type and regular and variant are valid on the Product Object Type

Packaging

The parameters available within this canvas are mainly relevant to the setup of packaging, which covers the ability to structure and manage packaging information in PDX.

Packaging objects ('Pallet', 'Case', 'Pack') are created as separate objects directly below a dedicated 'PackagingRoot' node in STEP.



References connect each packaging object to the next lower-level packaging object and from the lowest level to the sellable product (the Product Object Type).

The quantity in each packaging object is held in the 'Qty of Next Lower Package' metadata attribute found on the packaging reference.

The following parameters are available within this canvas:

- **Packaging Attribute Group:** This parameter is to be configured with an attribute group that contains all Vendor Facing Packaging Attributes across all levels.
- **PackagingRoot:** The field defines the root node in the Primary Product Hierarchy where all packaging objects are created when submitted from PDX.
- **Packaging References Attribute Group:** This parameter contains an attribute group. This configuration controls which attribute group the packaging references in PDX are visualized below.
- **Packaging Type Attribute:** This parameter contains an attribute that is used in PDX to identify the packaging type of the PDX product. The attribute should be a LOV with a value for each relevant packaging level. For example, 'Pallet', 'Case', 'Pack', and 'Each.' PDX supports up to 4 levels in a packaging hierarchy
- **Next Lower Package Quantity:** This parameter contains an attribute and identifies the 'Qty of Next Lower Package' metadata attribute. The metadata attribute must be valid for all packaging references.

Packaging Object Types

The parameters available within this canvas are relevant to the setup of the packaging hierarchy. This parameter contains up to 4 entries, one for each relevant packaging level. For each packaging level the following must be configured:

- Packaging product object type
- ValueID of the relevant entry in the LOV for the Packaging Type Attribute selected above
- Packaging reference that point to the next lower level (not relevant for level 1)

Only the levels taken into use need to be filled out, i.e., one of the following:

- Level 1 and level 2
- Level 1 and level 2 and Level 3

- Level 1 and level 2 and Level 3 and Level 4

Below is an example of the configuration of a 4 level packaging hierarchy:

Packaging Object Types			
Level 1	ID	Packaging Type	
	BuyItem	EA	
Level 2	ID	Child Reference Type	Packaging Type
	Pack	PackToChild	PAK
Level 3	ID	Child Reference Type	Packaging Type
	Case	CaseToPack	CS
Level 4	ID	Child Reference Type	Packaging Type
	Pallet	PalletToCase	PAL

In this example above it is assumed that the Packaging Type Attribute parameter is configured with an attribute with this packaging type LOV:

- Value: 'Pallet', ValueID: PAL
- Value: 'Case', ValueID: CS
- Value: 'Pack', ValueID: PAK
- Value: 'Each', ValueID: EA

Workflow And Status Tab

The Workflow And Status tab of the PDX Channel Configurator contains multiple parameters that define the elements required to build the interaction flow between STEP and PDX, this includes event queues, product status definitions, involved attributes and attribute groups, etc.

An event queue is used in the status communication between STEP and PDX. The intended use of this is that when a product passes through a vendor relevant workflow (e.g. 'New Product introduction' or 'Product Enrichment') appropriate events are triggers at workflow transitions. This makes it possible for a retailer to communicate the current status of the product to a vendor and also send messages to a vendor.

- Event Queue: When a product proceeds through workflows in STEP, or the product is being moved by the retailer, a status update can be sent to PDX. This update is sent by creating an event on the status event queue, which is read by PDX.

For this process to work correctly, 'Queue Status' must be set to 'Read Events', and 'Consumer Read' must be enabled.

- The content of the status update sent to PDX must contain the following information about the product:
 - STEP ID: This is displayed as the External Product ID on the product in PDX.
 - The 'PDX ID Attribute' attribute value: This is used as the key between STEP and PDX.
 - The 'PDX Status Attribute' attribute value: This attribute contains the PDX status, that should be displayed to the vendor:
 - 'STEP States Attribute' attribute value: This is an attribute containing information about the progress of the product onboarding. The value will be shown as Workflow state on the product in PDX.
 - The 'Supplier Message Attribute' attribute value: This attribute contains product onboarding-relevant information from the retailer. For example, the reason a retailer has returned the product for rework. The value given to this attribute will be added as a new entry in the product's 'Product history' in PDX.

Reference the examples of PDX Status, Workflow State and Message to Supplier shown below:

Summary Product attributes Packaging Product variants Digital assets

Product details

Name	20230221105148724A-EA01
Category	Jeans Product Root > Apparel > Lower body wear > Trousers > Jeans
Introduced to channel	Feb 21, 2023, 11:52:39 AM
Last updated	Mar 7, 2023, 7:51:33 PM
Submitted	Feb 28, 2023, 5:28:00 PM
Online	-
Status	✔ Accepted
External product ID	1881905
Errors	1 error

Workflow state

STEP workflow state

Product history

<p>2023-02-28 17:28:00</p> <p style="text-align: center;">✔ Accepted</p>	<p>Product accepted by Pdxrd-dev this is an important message to supplier</p>
<p>2023-02-28 17:27:25</p> <p style="text-align: center;">🔄 Submitted</p>	<p>Product submitted to Pdxrd-dev Submit action to channel finished with success Go to submission log See submission log details</p>
<p>2023-02-28 17:27:20</p> <p style="text-align: center;">🔄 Submitted</p>	<p>Product submitted to Pdxrd-dev Processing submission Go to submission log See submission log details</p>
<p>2023-02-21 11:52:39</p> <p style="text-align: center;">✔ Added</p>	<p>Product added to channel Action type: MANUAL</p>

[Show less](#)

The following images displays the parameters available within the Workflow And Status tab:

ACMA (acma)

<
Supplier Data Definition
Workflow And Status 8
PDX Presentation

* PDX Status Attribute	<input type="text"/>	
* STEP States Attribute	<input type="text"/>	
* Supplier Message Attribute	<input type="text"/>	
* Event queue	<input type="text"/>	
Maintenance		
Event Queue Maintenance	<input type="text"/>	
Include in Maintenance	<input type="text"/>	
Promote to Master Data	<input type="text"/>	
Maintenance ID Attribute	<input type="text"/>	
Maintenance ID Attribute on Reference	<input type="text"/>	
Status Mapping		
* PDX Success Attribute Mapping	<input type="text"/>	
* PDX Rejected Attribute Mapping	<input type="text"/>	
* PDX Returned Attribute Mapping	<input type="text"/>	
* PDX Sent Attribute Mapping	<input type="text"/>	

Save
Delete

➤ Save and publish

- **PDX Status Attribute:** This parameter is to be configured with a LOV based attribute valid on product object type, that carries the product status being sent back to PDX to notify vendors. This attribute should be updated on state changes in the product onboarding or maintenance workflows, wherever PMDM needs to send product status back to PDX, notifying vendors if the product is rejected, or needs reworks, or is approved. Updates are sent to PDX via events on the Product Status Event Queue, described in detail below. For information about how to map these attribute values to valid PDX Statuses, refer to the 'Status Mapping' section defined later in this topic.
- **STEP States Attribute:** This parameter is to be configured with a text attribute that is valid on product object type. This attribute is being used to carry the STEP Workflow state information sent back to PDX. This attribute should be updated on state changes in the product onboarding or maintenance workflows, wherever PMDM needs to send product status back to PDX, notifying vendors if the product is rejected, or needs

reworks, or is approved.

- **Supplier Message Attribute:** This parameter is to be configured with a text attribute valid on product object type. This attribute is being used to carry a notification message sent back PDX. This attribute should be updated on state changes in the product onboarding or maintenance workflow, wherever PMDM needs to send product status back to PDX, notifying vendors if the product is rejected, or needs reworks, or is approved.
- **Event queue:** This parameter is to be configured with an event queue which is the integration method from STEP to PDX for sending product status updates. Whenever STEP needs to notify vendors about a product status change (together with messages), the product should have Product Status, STEP states and Supplier Message attributes set, and be published to this event queue.

The PDX side has an API call issued automatically every 1 minute, which fetches messages from the event queue, then consumes them accordingly. Setting product status related attributes and publishing to event queue is usually handled within a workflow in STEP.

Maintenance

When PDX is adopted by retailers as the primary product onboarding tool for bringing products into a STEP instance, the retailer has likely migrated supplier-provided product records into STEP from a different tool. In this scenario, the supplier-provided products in the retailer's STEP system are now out of sync with the same products in the supplier's system. This can present challenges when retailers and suppliers move forward together with PDX. The purpose of the 'Maintenance' functionality is to enable retailers to continue to receive product data from suppliers for new and existing products and enable suppliers to maintain those products after the retailer transitions to the PDX platform.

Note: 'Maintenance' is not intended to establish a continuous two-way data synchronization between PDX and STEP, but rather to push existing products from STEP to suppliers in PDX during the initial onboarding of the supplier to PDX.

The parameters available within this canvas are mainly relevant to the Maintenance set up.

The Maintenance functionality gives retailers the ability to push existing products from STEP into the relevant channel in PDX, in effect sending them back to the supplier. In that PDX channel, the products will be processed as so-called 'Retrieved products' for the supplier in question. Next, the supplier can reclaim the products and maintain them as they would any other.

When retrieved products enter PDX, two things happen:

- The products will be created in the PDX channel for the relevant supplier
- The products will appear in the master data area in PDX

The product's ID in PDX, as well as the attribute values the retailer wants to be viewable in the channel and in master data, can be configured in STEP. The product data shared in this way can be used by the supplier to recognize the incoming product and match it to the supplier's existing product record.

A product that enters PDX in this way will be marked as a 'retrieved product' and will enter the PDX onboarding process in the workflow state named, 'Made available for maintenance in <name of channel>', as shown in the screenshot below.

Summary Product attributes Packaging Product variants Digital assets

Product details

Retrieved product

Name	Name 2WS-30098(2)
Category	Jeans Product Root > Apparel > Lower body wear > Trousers > Jeans
Introduced to channel	Feb 28, 2023, 12:30:01 PM
Last updated	Feb 28, 2023, 6:09:27 PM
Submitted	-
Online	-
Status	 Progress
External product ID	1881883
Errors	-

Workflow state

Made available for maintenance in Pdxrd-dev

Product history

2023-02-28 11:31:00	 Added	Product added to channel Made available for maintenance in Pdxrd-dev
------------------------	---	--

Pushing product to PDX in Maintenance

In the Maintenance process, products are pushed from retailers to suppliers by generating events on a dedicated event queue.

Note: If many events are generated at once, PDX may require additional time (up to 24 hours or more) to consume and distribute all events.

If packaging information is managed by vendors, the event queue should be extended to automatically include any packaging objects in the messages made available to PDX. This means that when a product is pushed to PDX, the parent packaging hierarchy will also be pushed to PDX. This can be done using business actions.

Another useful business rule to make is a business action that can generate events on e.g. a collection of products. This might be done, for example, on all products for a specific supplier.

Note: Before a product can be sent to PDX the product must reference a supplier and have a Maintenance ID (attribute is configured below).

The event generation business rule above should be extended, so that it checks these requirements. Then, if the validation for a product failed, no event is generated, and the process can move to the next product.

For a channel with Maintenance enabled, a given supplier can only be authenticated under one PDX Client login. This means that if the same supplier attempts to add the channel in another PDX client, they will get the following error message: 'This vendor has already been added to this channel. Please configure another Vendor Identifier and try again.'

To enable the PDX maintenance flow, the following parameters are required:

- **Event Queue Maintenance:** This parameter is to be configured with an event queue which is the integration method from STEP to PDX for sending existing products to PDX as described above.
- **Include in Maintenance:** This parameter is to be configured with an attribute group. Attributes and asset references that should be added to the product in the PDX channel and stored in the retrieved layer should be linked into this attribute group.
- **Promote to Master Data:** This parameter is to be configured with an attribute group. Attributes to 'promote' to master data, i.e., Attributes that will be visible in master data in PDX should be linked into the attribute group.
- **Maintenance ID Attribute:** This parameter is to be configured with an attribute. As mentioned above, the product ID used in PDX can be configured from STEP. The STEP attribute (the Maintenance ID) for storing this ID must be configured for this parameter. It should be valid for the Product Object Type and all packaging object types (if relevant).

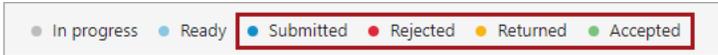
By default, this product ID will be used in PDX to match retailer's retrieved products with their corresponding product in the supplier's master data. This matching occurs when suppliers upload updates to their products via Excel sheet import or an integration using the external API.

This attribute should be IDs like GTIN or some supplier product identifier. The values should make it possible for a supplier to first recognize the product and then deploy PDX's 'Match-based import' and mapping capabilities to maintain the product.

- **Maintenance ID Attribute on Reference:** This parameter contains a metadata attribute. This is relevant if packaging objects are added to the maintenance queue. It can be used to help build an automated process that adds packaging objects to the maintenance queue (as recommended above). The metadata attribute must be valid on the packaging reference types. It should be a calculated attribute that on a given packaging reference include the Maintenance ID of the parent. This makes it possible for a business rule to deduce the next member in a packaging hierarchy to add to the maintenance queue.

Status Mapping

The parameters available within this canvas are mainly relevant to the mapping of values of the PDX Status Attribute and the status in PDX.



- **PDX Success Attribute Mapping:** This parameter is to be populated with a value ID of a valid value of the PDX Status Attribute. Products with PDX Status Attribute equal to this value will appear with PDX Status equal to Accepted.
- **PDX Rejected Attribute Mapping:** This parameter is to be populated with a value ID of a valid value of the PDX Status Attribute. Products with PDX Status Attribute equal to this value will appear with PDX Status equal to Rejected.
- **PDX Returned Attribute Mapping:** This parameter is to be populated with a value ID of a valid value of the PDX Status Attribute. Products with PDX Status Attribute equal to this value will appear with PDX Status equal to Returned.
- **PDX Sent Attribute Mapping:** This parameter is to be populated with a value ID of a valid value of the PDX Status Attribute. Products with PDX Status Attribute equal to this value will appear with PDX Status equal to Submitted.

PDX Presentation Tab

The PDX Presentation tab of the PDX Onboarding Channel Configurator contains multiple elements that define parameters related to the UI in PDX.

The order in which attributes and attribute groups display on products in PDX can be controlled from STEP. This requires a 'Display Sequence' metadata attribute valid on attribute data container and attribute group definitions:

- For attribute groups, when the 'Display Sequence' metadata attribute is applied to an attribute group, the display order of that attribute group and others can be controlled in the corresponding channel in PDX.
- For attributes, when applied to a regular attribute, the 'Display Sequence' metadata attribute can be used to control the ordering of attributes within an attribute group in PDX.
- For attributes in a data container (called composites in PDX), the order of the columns is also determined by the 'Display Sequence' metadata attribute.

Attributes and attribute groups with an assigned value for the 'Display Sequence' metadata attribute are ordered from lowest to highest, with those assigned a low 'Display Sequence' value displaying higher in a list, and those with a higher value displaying lower. Based on their 'Display Sequence' metadata attribute, data container attribute columns with a lower number will display to the left, and those with a higher number to the right.

To provide additional guidance to suppliers and other users enriching product data, users may write help text in STEP that will be available to suppliers in PDX. To display help text on attributes, data containers, and asset references accessible by PDX users, the objects for which help text has been written must be linked to the 'Help Text' attribute. Content added to this attribute will sit as metadata on the attribute, data container.

In PDX, the help text will display in one of two ways:

- When the attribute name is clicked, the "Details" panel will display on the right-hand side of the screen. The configured help text will display in this panel.

The screenshot shows a 'Filter attributes' table with columns for 'Attribute name' and 'Value to be submitted'. The 'Is Base Unit' attribute is highlighted with a red box. A red arrow points from this attribute to the 'DETAILS' sidebar, which contains the following text:

IS BASE UNIT
An indicator identifying the trade item as the base unit level of the trade item hierarchy.

VALUES
Your active value is your Channel specified value

ACTUAL VALUE (CHANNEL SPECIFIED)
No

Channel value ⓘ
No

Master data value ⓘ

- On a data container, the configured help text will display above the columns to the right of the field name, 'Description', as shown in the screenshot below.

The screenshot shows a data container table with the following structure:

Path: ACC4R DEV / Product Ratings

Description: Used to store Product Ratings on the product

#	PRODUCT RATING SCORE	PRODUCT RATING SOURCE	DATA CONTAINER KEY
86	BeerAdvocate	Rating #1	

Following are the parameters available within the PDX Presentation Tab

Node Details

Channel Properties ¹¹ Supplier Classification ³ **PDX Presentation ¹** PDX Rules

Display Sequence

* Help Text

Attribute Group for Account Identifier

- **Display Sequence:** Users can configure a metadata attribute for Attributes, data containers and Attribute groups. This metadata attribute serves to determine the order in which attribute groups, attributes within an attribute group, and data container columns are presented within PDX UI.
- **Help Text:** This parameter shall be configured with a metadata attribute that is valid for a regular attribute. This metadata attribute provides vendors with additional context and assistance in understanding the

purpose and usage of a given attribute.

- **Attribute Group for Account Identifier:** PDX supports that multiple collaborating suppliers can share one PDX client. In this case multiple channel accounts will be added to the channel configuration of the Onboarding channel. Each channel account corresponds to a STEP supplier classification.

An example of a channel setup with multiple channel account can be see below:

General Accounts Lookup tables			
Identifier	Vendor Portal Username	ID of the Supplier in STEP System	Actions
ACME	ACMEPDS	PMDM.UG.Acme	⋮
MALM	24222292-8E3D-461D-82B8-E077B097D2CC	d54759e4-589b-44b0-9f4d-a366f1616668-CL	Add account +

In this case the Account attribute in PDX is used on products to indicate the correct Supplier of the product. The Account attribute in the PDX channel will accept values from an LOV of the configured channel accounts. The Account attribute is not a STEP attribute but rather an internal PDX attribute that represents STEP supplier classifications, see below for an example:

Account

MALM

MALM

ACME

This parameter is to be configured with a subgroup that is within the Vendor Facing attribute group for PDX. This is used to contain the PDX virtual attribute called 'Account.'

In the absence of a configured attribute group, the Account attribute will be automatically assigned to the 'Ungrouped Attribute' group.

PDX Rules Tab

The PDX Rules tab of the PDX Channel Configurator contains multiple parameters that define the setup of data validation rules in PDX.

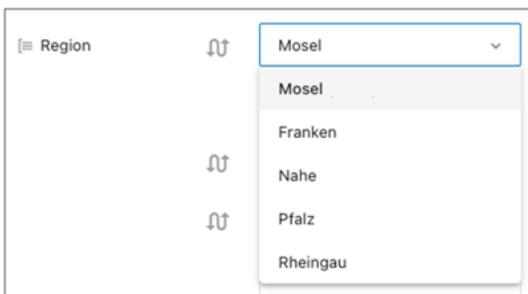
LOV Restrictions

Retailers can configure their channel so that when PDX users onboard products, they can make a selection in one LOV attribute and the options available to them in an entirely different LOV attribute will be pre-filtered. This presents the PDX user with a retailer-configured subset of LOV values from which to select. This can be done by configuring a product data validation with a LOV restriction condition.

As an example, an LOV attribute named 'Country' is a product attribute a supplier can enrich in PDX. From the LOV dropdown, the user has selected the value 'Germany', as shown in the screenshot below.

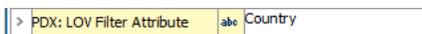


Because the LOV restriction has been implemented, when the PDX user selects 'Germany' the LOV options available in the LOV attribute 'Region' will display only German regions, even though the LOV from which the values are drawn contains regions for many countries.



This pre-filtering is controlled via an Attribute metadata attribute, which must be made available on the attribute definition.

The LOV restriction metadata attribute is used to enable the connection between the two relevant LOV-based attributes. For the dependent attribute, which is the LOV-based attribute whose values will be filtered, the value assigned to the LOV restriction attribute should be the ID of the defining attribute, which is the LOV-based attribute that determines the valid values. To continue the example described above, the dependent attribute 'Region' would have the LOV restriction attribute populated with the value 'Country'.



Additionally, the ValueIDs for the two LOVs configured for LOV restriction must also satisfy additional requirements for the cross-validation to work.

- The ValueIDs of the dependent LOV values, the values being filtered, must be prepended with the value ID of the defining LOV for the cross-validation to work properly.

To continue the example described previously, the LOV attribute 'Country' includes an LOV value 'Germany' with the ValueID 'germany'. To restrict the region options available in the LOV attribute 'Regions' to only German regions when 'Germany' is selected, the ValueID for the German region 'Mosel' must begin with the ValueID for 'Germany', which is 'germany'. Using this logic, a possible ValueID for 'Mosel' could be 'germany_mosel'.

- The LOV restriction allows for multiple layers of dependent attributes, not just two.

To expand upon the previous example, let us say the retailer wants to include a second dependent LOV attribute, in this case 'City', to the cross-validation. In this scenario, the LOV restriction attribute for 'City' must be set to 'Region'. A supplier would first select 'Germany' for 'Country', then 'Mosel' for 'Region', and then, for the 'City' LOV attribute, only retailer-relevant German cities in the selected region should be available for the supplier to select. So, if an LOV value from the LOV 'City' is to be a selectable value when the user selects 'Mosel', the ValueID for the valid city, in this case 'Koblenz', will need to have both of its defining LOV values prepended to its ValueID. Using this logic, a possible ValueID for 'Koblenz' could be 'germany_mosel_koblenz'.

• English US • Main

ACMA (acma)

Workflow And Status
PDX Presentation
PDX Rules

Mandatory For Submit	<input type="text"/>	
Category Mandatory For Submit	<input type="text"/>	
Locked After Initiate	<input type="text"/>	
LOV Restriction	<input type="text"/>	
Business Rule Rule Engine		
Conditionally Mandatory Attribute	<input type="text"/>	
Validation Engine Attribute	<input type="text"/>	

Save
Delete

➤ Save and publish

Following are the parameters available within the PDX Rules tab:

- **Mandatory For Submit:** This parameter shall be configured with an attribute group. The attribute group holds those attributes that vendors must provide before submitting a product from PDX. If an attribute is mandatory for specific categories, it should not be included in this attribute group.

- **Category Mandatory For Submit:** This parameter shall be configured with a metadata attribute. An LOV-based metadata attribute for Product Attribute Link, which defines if an attribute is mandatory for specific categories. The attribute should be LOV-based with allowed values of 'No' (Value ID=N) and 'Yes' (Value ID=Y).
- **Locked After Initiate:** This parameter shall be configured with an attribute group. This attribute group holds attributes that the retailer would like to be locked after initial submission or after sending into PDX via the maintenance flow. For example, if a supplier has submitted a product with a specific GTIN, the retailer would want to lock down that attribute so as not to allow the supplier to re-submit the same product with a new GTIN.
- **LOV Restriction:** This parameter contains a metadata attribute and defines the LOV restriction metadata attribute as explained above.

Business Rule Engine

The parameters available within this section are relevant to the configuration of advanced product data validations.

Below are the parameters that are available within the Business Rule Engine section:

- **Conditionally Mandatory Attribute:** Making attributes conditionally mandatory using the conditionally mandatory functionality should be avoided. It is recommended that the Validation Engine (reference the bullet displayed below) is used instead as this is a more powerful tool that supports a wider range of related use cases.
- **Validation Engine Attribute:** This parameter is to be configured with the Validation Engine metadata attribute. The Validation Engine attribute must be multi-valued and use the 'Regular Expression' validation base type.

The correct Regular Expression' to use can be obtained via request to the PDX Team

A PDX onboarding channel offers the ability to configure advanced category-specific data validations in the PDX channel via a 'configuration language' referred to in PDX as the Validation Engine.

The Validation Engine is a tool that can be used to build advanced 'product data validations' without requiring admins to write code. Instead, an expression language is used to create the validations.

The 'Validation Engine' is controlled using a metadata attribute on the Product Attribute link. In this way, advanced category-specific data validations can be run on attributes. The Validation Engine attribute must be multi-valued and use the 'Regular Expression' validation base type.

Configuration details

Every product data validation created in the 'Validation Engine' consists of two or three parts:

- Conditions
- Error Message
- Rule ID (optional)

Product data validations to create validations in PDX are written in the following format, with each element separated by a 'colon' delimiter. Below is a screenshot of the validation format with each element contained within a red box, each specified by a lower-case letter. Below the screenshot is a description of each element:

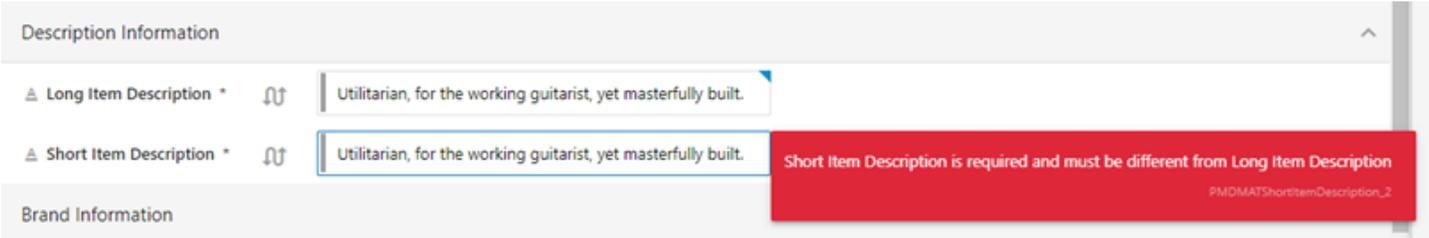
```
<Condition>:error:<Error Message>:ruleid:<Display Name of Product data validation >
```

- a. The logic condition
- b. The keyword that indicates that what follows is the error message
- c. The error message
- d. Keyword that indicates that what follows is the name of the rule
- e. The name of the rule

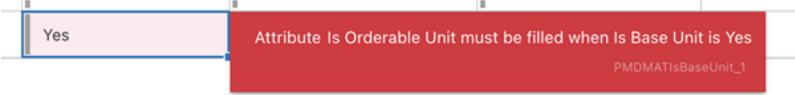
Note: The error 'occurs' on the attribute if the condition is met.



To ensure each product data validation is unique in the system, all generated product data validations will have an ID with an incrementing number appended to it (e.g., 'PMDMATShortitemDescription_2' with 'PMDMATShortitemDescription' being the product data validation's ID and the '_2' the incrementing number appended to the ID).



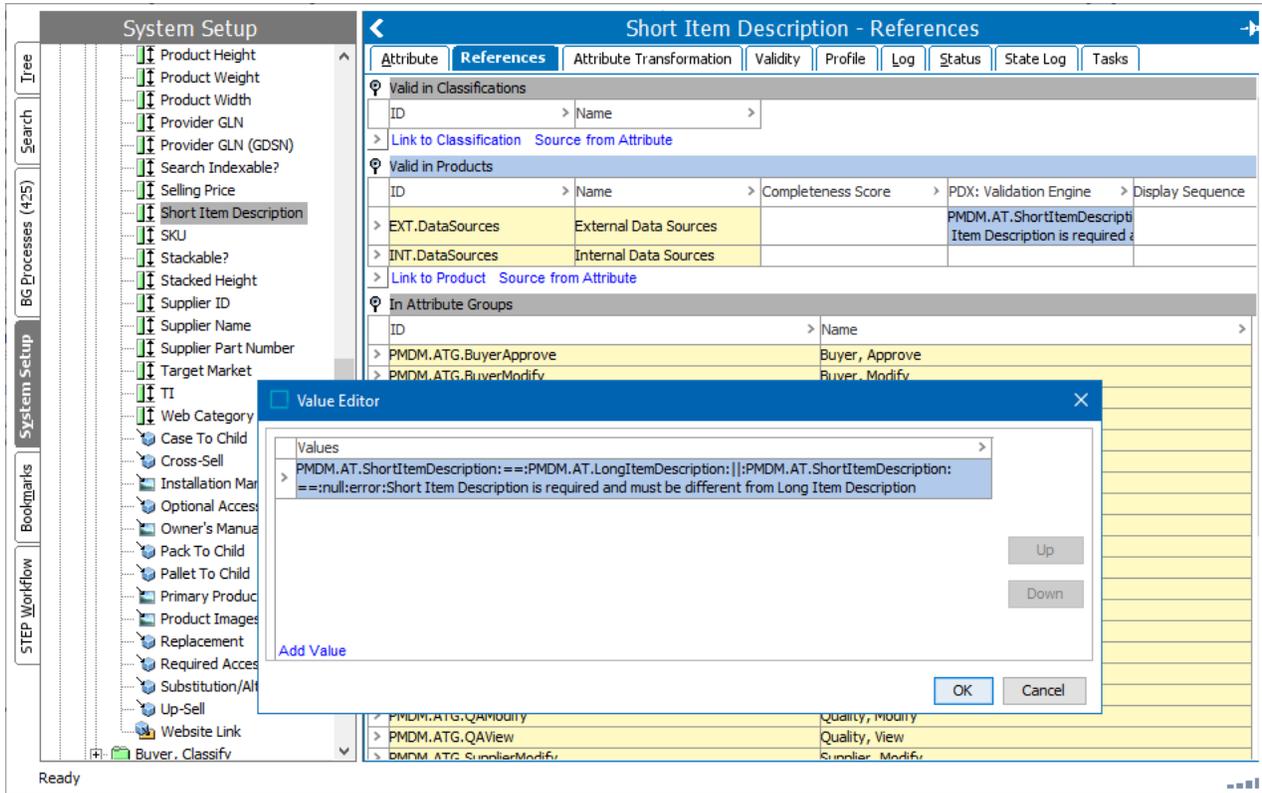
If no ':ruleid:' is provided in the product data validation, the ID will use a simplified version of the attribute (e.g., 'PMDM.AT.IsBaseUnit' becomes 'PMDMATIsBaseUnit' with an incrementing number appended to it).



Validation Example

To illustrate how a product data validation is written, a product data validation example, written in the STEP Workbench, is shown in the screenshot below and described in detail in this section.

In this example, the product data validation is written and applied to the 'Short Item Description' attribute.



When properly written, this product data validation presents an error to the PDX user when the 'Short Item Description' attribute either has no value, or the value for 'Short Item Description' is equal to the value for 'Long Item Description', meaning the values for both are the same.

Below is the example product data validation as it must be written to accomplish the validation task:

```
PMDM.AT.ShortItemDescription:==:PMDM.AT.LongItemDescription:||:PMDM.AT.ShortItem
Description:==:null:error:ShortItem Description is required and must be different
from Long Item Description
```

In this example, the product data validation is made up of the following elements:

- The condition -- This part of the product data validation specifies the data and which aspect of the data is being validated.

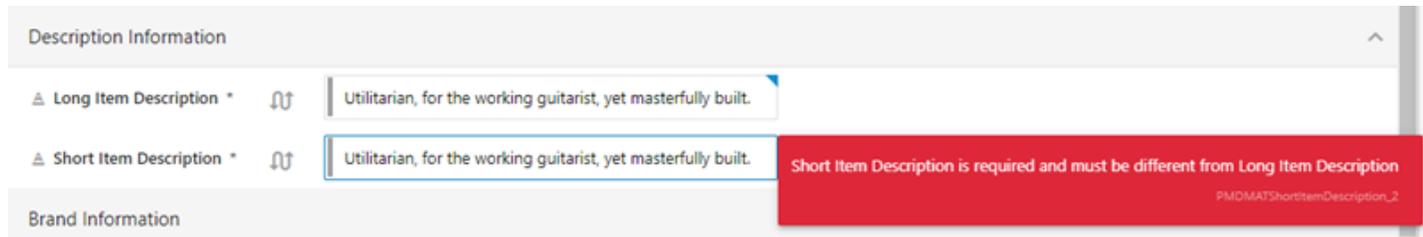
```
<Condition>=
PMDM.AT.ShortItemDescription:==:PMDM.AT.LongItemDescription:||:PMDM.AT.ShortItem
Description:==:null:
```

- The error message -- If the conditions defined in the product data validation are not met, the text defined in this element are presented to the user.

```
<Error Message>= Short Item Description is required and must be different from
Long Item Description
```

- Product data validation ID -- In this example, no product data validation ID is set (the ':ruleid:' element). As described previously, the PDX system will generate an ID for the rule that is a version of the validated attribute with an incrementing number appended to the product data validation to ensure the ID is unique in the system.

The error shown in PDX will display as shown in the screenshot below:



Product data validation elements

A product data validation, as described in the example above, is built using simple logical operators, attribute IDs, numbers, and text strings. These elements can be combined to produce advanced validations. In the table below, each element is listed and includes a description of the element and an example of how that element can be written into a product data validation.

Elements	Example	Description
STEP Attribute ID	PMDM.AT.LongItemDescription	Refers to a specific attribute value.
Fixed numbers or text strings	20 'Y' 'unsaturated fat'	Refers to a fixed value. Note: When comparing with LOV values, the 'ValueID' should be used. Literals should be enclosed in single quotes (').
Comparison	< ==	Simple comparisons can be done to compare numeric values and text strings. Only values from single-valued attributes are supported.

		<p>The full list includes:</p> <p>< -- Less than</p> <p>> -- Greater than</p> <p>=< -- Less than or equal to</p> <p>>= -- Greater than or equal to</p> <p>== -- Equal to</p> <p> = -- Not equal to</p>
Null		<p>The constant null is used to indicate 'no value' and can be used to check if an attribute has no value.</p> <p>Only values from single-valued attributes are supported.</p>
Logical operators	(<p>Logical operators can be used to group and combine conditions into more complex conditions.</p> <p>The full list includes:</p> <p>(-- Start grouping</p> <p>) -- End grouping</p> <p> -- Logical 'OR'</p> <p>&& -- Logical 'AND'</p>
Mathematical operators	+ -	<p>The mathematical operators can be used to combine numeric values and produce a</p>

		<p>calculated value.</p> <p>Only values from single-valued attributes are supported.</p> <p>The full list includes:</p> <ul style="list-style-type: none"> + -- Additional - -- Subtraction * -- Multiplication
contains		<p>The 'contains' method can be used to check if a value (numeric or textual) includes another value. Multivalued and single-valued attributes are supported.</p>
notContains		<p>The 'notContains' method can be used to check if a value (numeric or textual) does not include another value. Multi-valued and single-valued attributes are supported.</p>
isEmpty		<p>The 'isEmpty' method can be used to check if an attribute has no values Multi-valued and single-valued attributes are supported.</p>
isNotEmpty		<p>The 'isNotEmpty' method can be used to check if an attribute has values. Multi-</p>

		valued and single-valued attributes are supported.
--	--	--

To further illustrate how product data validations can be applied to ensure data supplied in PDX meets the configured requirements, listed below are a series of product data validation examples covering a variety of use cases:

- `PMDM.AT.CountryOfOrigin:=='BS':&&:PMDM.AT.BrandName:==:null:error`:If Country of Origin is equal to BAHAMAS, then Brand Name is required
 - In this example, 'BS' is the ValueID of the value 'BAHAMAS' in the list of values (LOV) for the 'Country of Origin' attribute.
- `(:PERCENTAGE_1+: PERCENTAGE_2+: PERCENTAGE_3):>:100:error`:The total value of Percentage 1, Percentage 2 and Percentage 3 cannot be more than 100:ruleid:100 Percentage rule
- `SHOELACE_MATERIAL:=='NOT_APPLICABLE':&&:(:ACCESSORIES_MULTY:contains:LACES:|:ACCESSORIES_MULTY:contains:SHOELACE:):error`: 'Not Applicable' is NOT an acceptable value for Shoelace Material when Lace is selected for attribute Accessories List.
- `ALL_FEATURES_MULTY:contains:'ADDITIONAL_BURNERS':&&:isEmpty:ADDITIONAL_BURNER_TYPE:error`:The Additional Burner Type is required when All Features contains Additional Burners
- `POWERED_MULTY:notContains:"BATTERY_POWERED":&&:isNotEmpty:BATTERY_TYPE_MULTY:error`:The list of Battery types must be empty, when product is not powered by batteries

When writing product data validations, it is important to note the following considerations and limitations:

- Comparisons, such as '==' and 'contains' are case-sensitive.
- The method 'isEmpty' can be either prepended or appended to the relevant [text] for the same result. This also applies to the function 'isNotEmpty'. For example, the notation 'isEmpty:ADDITIONAL_BURNER_TYPE:' and ':ADDITIONAL_BURNER_TYPE:isEmpty' can be used interchangeably.
- The 'Validation Engine' does not support product data validations that validate data containers or data container attributes.
- Most operations and comparisons do not support attributes with units.
- Product data validations can only be added and / or defined on attribute links to specification attribute.
- Product data validations cannot be written to include assets.
- When writing product data validations, it is important to note that the colon character (':') is a reserved character and should be used only to separate elements in the product data validation. If the character is used in an attribute ID or in an error message, for instance, this may cause the product data validation to error.

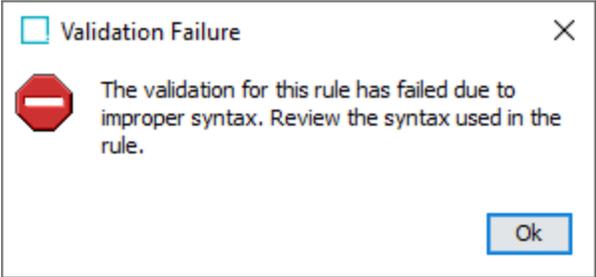
Using expressions correctly

As mentioned previously, the 'PDX: Validation Engine' attribute uses the 'Regular Expression' validation base type. Regular expressions prevent product data validation writers from committing the most frequent notational errors when writing product data validations, but cannot address certain kinds of human error, like misspelled attribute names, for instance.

Listed in the table below are rules that must be followed:

Description	Invalid examples	Valid examples
":error:" must be present and have non-empty message	ATTRIB_A:<=:ATTRIB_B:error: ATTRIB_A:<=:ATTRIB_B	ATTRIB_A:<=:ATTRIB_B:error:A should be greater than B
At least one truth valued operator must be present (i.e. + or - is not enough) and attribute names must be separated by operators or parentheses	ATTRIB_A:error:Some message (:ATTRIB_A:ATTRIB_B):error:Some message (:ATTRIB_A+:ATTRIB_B):error:Some message	(:ATTRIB_A+:ATTRIB_B:=:1):error:Some message
"(" is matched by ")"	(ATTRIB_A:<=:ATTRIB_B:error:Some message	(ATTRIB_A:<=:ATTRIB_B):error:Some message
if ":ruleid:" is present, then it has non-empty content	ATTRIB_A:<=:ATTRIB_B:error:Some message:ruleid:	ATTRIB_A:<=:ATTRIB_B:error:Some message:ruleid:Rule 1
:ruleid can only contain alphanumeric characters and -_#:	width*:height>:1000:error:too large:ruleid:Unsanitized\$ID	width*:height>:1000:error:too large:ruleid:Unsanitized_ID

If the regular expression is implemented, some of the rules above will be caught by the regular expression check when changes are saved.



Publishing PDX Onboarding Channel Configurators

Once all the mandatory parameters are filled and have successfully passed the validation process, the PDX Onboarding Channel Configurator is ready to be published.

Clicking the 'Save and publish' button available at the bottom of the screen triggers the creation of an XML-type asset object.

channel (channel)

Channel Properties Supplier Classification Supplier Data Definition Workflow And Status PDX Presentation

* Channel Name:

* Logo URL:

* Export Configuration:

* Product Import Configuration:

* Product Import Folder Path:

* PDX ID Attribute:

* Unique Key used by PDX to refer to STEP products:

* Processed By PDX:

* STEP Service User:

* Context ID:

* Workspace ID:

Language Handling Attribute:

Language Mapping Attribute:

Invitation Only: Yes No

Is Public Channel: Yes No

Below is a screenshot of an XML file stored as an asset inside a folder named 'PDX ConfigFile'

The screenshot displays the STIBO SYSTEMS interface. On the left is a 'Tree' view showing a hierarchical structure of assets and configurations. The 'PDX ConfigFile' folder is expanded, showing various channel configurations, with 'channel' selected. On the right, the 'channel rev.1.0 - Images & Documents' window is open, showing a table of metadata for the selected channel.

Name	Value
ID	channel
Name	channel
Object Type	XML File
Revision	1.0 Last edited by PDX on Mon Apr 24 14:04:45 CEST 2023
Approved	✗ Never Been Approved
Translation	Not Translated
Path	Classification 1 root/Assets/PDX ConfigFile/Morc channel
Access Time	
Asset Locking	
Asset URL Attribute	URL
Caption 1	abc
Caption 2	abc
Caption 3	abc
Certificate Valid from	abc
Certificate Valid to	
Completeness Asset Metadat	0 %
Confirmed Duplicates	abc
Confirmed Non-Duplicates	abc
Copyright (Exp.)	
Deduplication Delete Flag	abc
eCl@ss Primary Key	abc
eMail - external recipient(s)	abc
Keywords	abc
Output Channel Use	
Recipient(s)	
Regional Use	

Name	Value
Extension	abc xml
Filename	abc Morc channel.xml
Format	abc XML (Extensible Markup Language document)
MIME Type	abc application/xml
Size	abc 8,728
Upload Time	abc 2023-04-24 14:04:45

Based on these configurations it is possible from the PDX environment to build a new onboarding channel and connect to the designated STEP System

The 'Channel Data Standard generation' can be made via a request to your PDX team.

At the initial channel creation, the following information must be shared with the PDX Team

- URL and path to the restapi endpoint on the STEP Instance
- The STEP ID of the published Channel Configuration mentioned above
- Username and password of a user that can used to access the restapi endpoint on the STEP Instance
- Password of the STEP service user, defined in the Channel Properties Tab
- The PDX environment the channel is to be used on.

- Typically, the initial testing will take place on <https://pdx-preprod.stibosystems.com/> and the production version of the channel will be installed on <https://pdx.stibosystems.com/>

The Data Standard update is implemented by the PDX team. Besides exporting the channel configuration, the update process will also export a PDX Channel Schema from STEP using the export configuration defined within the Channel Properties tab. This export, generated from the 'Approved' workspace in STEP, then converts the PDX Channel Schema and the Channel Configuration into the channel-specific Data Standard in PDX, so that changes made to the product hierarchy, relevant attributes, etc., are properly reflected in PDX.

When a change is made to the channel configuration or to the underlying data model that affects the PDX channel, the channel-specific Data Standard in PDX must be updated to account for the change.

The following actions are examples of changes made in STEP that require a Data Standard update in PDX:

- Changes made to the Channel Configuration
- Adding, modifying, or deleting supplier-facing attributes or LOVs.
- Adding, modifying, or deleting nodes in the supplier-facing product hierarchy.
- Linking or unlinking attributes within the supplier-facing product hierarchy.
- Adding or deleting supplier-facing asset reference types.

Data Standard updates can either be made by request to your PDX team, or as part of a scheduled process if schema changes are likely to be more frequent.