

SOLUTION ENABLEMENT

PMDM for Automotive Reference Guide

Release 9.2-MP1 (updated October 25, 2019)

Table of Contents

Table of Contents	2
Automotive Reference Guide Introduction ..	9
Patch Recipe and Compatibility	10
Supported Versions and Formats	11
Importing Automotive Data	13
Import Framework	14
Import Process Overview	14
Default Workflow States and Functions	17
Validation State	19
Validation State Common Parameters ..	19
Validation State TecDoc Specific Parameter	21
Validation Service Function Details	22
Validation State Results	23
Conversion State	25
AutoCare Conversion State Parameters	25
NAPA Conversion State Parameters ..	27
TecDoc Conversion State Parameters	29
Conversion State Function Details	32
Conversion State Results	33
Delta Calculation State	35
Delta Calculation State Common Parameters	35
Delta Calculation State Function Details	36

Delta Calculation State Results	37
Error State	39
Ready for Import State	40
Rejected State	41
Import State	42
Import State Common Parameters	42
Import State Function Details	43
Import State Results	44
Import Completed State	45
Discard File State	46
Modifying Import Framework	47
Displaying Import Modifications in Web UI ..	48
ID Structures in Importers	49
Adding Reporting Extensions to Imports	51
Validation Error Handling	52
AutoCare Importers	55
AutoCare ACES Importer	56
Using AutoCare ACES Importer	57
Importing ACES Files via Web UI	58
Import Process Overview	58
Using the AutoCare ACES Importer in Web UI	59
Sample AutoCare ACES File Structure	64
Header	65
Body	65
Footer	68
ACES Import Validation Rules	69
Configuring AutoCare ACES Importer ...	71

Configuring an IIEP for AutoCare ACES Imports	72
Configuring a File Loading Widget for AutoCare ACES Imports	78
AutoCare PIES Importer	81
Using AutoCare PIES Importer	82
Importing PIES Files via Web UI	83
Import Process Overview	83
Using the AutoCare PIES Importer in Web UI	83
Sample AutoCare PIES File Structure	89
Header	95
Price Sheet Header Segment (PRCS)	101
PIES Item Segment (ITEM)	102
PIES Description Segment	104
PIES Price Segment PRCE	106
PIES Extended Product Info Segment (EXPI)	107
Packaging Segment (PACK)	110
Hazardous Material Package Segment (HAZM)	112
Kits Segment (KITS)	114
Interchange Segment (INTE)	115
Digital Asset File Information Segment (ASST)	116
Output	119
PIES Import Validation Rules	122
Configuring AutoCare PIES Importer ...	123
Configuring an IIEP for AutoCare PIES Imports	124

Configuring a File Loading Widget for AutoCare PIES Imports	130
AutoCare PAdb Importer	134
Using AutoCare PAdb Importer	135
Importing PAdb Files via Web UI	136
Import Process Overview	136
Using the AutoCare PAdb Importer in Web UI	136
PAdb Import Validation Rules	142
Configuring AutoCare PAdb Importer ...	143
Configuring an IIEP for AutoCare PAdb Imports	144
Configuring a File Loading Widget for AutoCare PAdb Imports	150
Qdb Import Validation Rules	154
Brand Table Import Validation Rules	155
PCdb Import Validation Rules	156
VCdb Import Validation Rules	157
TecDoc Reference Data Importer	158
Using TecDoc Reference Data Importer ..	159
Importing Reference Data Files via Web UI	160
Import Process Overview	160
Using the TecDoc Reference Data Importer in Web UI	160
Reference Data Import Validation Rules	166
Error Handling for Reference Data Imports	167
Link Type (Reference Type) Not Valid for Generic Article	168
Missing Country Code When	173

Importing Vehicle Country Specifications	221	Step 2. Select Objects	221
Missing Object Type When Adding Additional Descriptions to Vehicle Model Series	177	Step 3. Select Format	222
Missing Object Type When Allocating Engine Number	181	Step 5. Advanced	227
Missing Object Type When Allocating German KBA Vehicle Number	185	Example	230
Validation Error When Importing LOV Values	189	AutoCare PIES Exporter	235
Configuring TecDoc Reference Data Importer	192	Step 1. Start AutoCare PIES Exporter ..	235
Configuring an IIEP for TecDoc Reference Data Imports	193	Step 2. Select Objects	235
Configuring a File Loading Widget for Reference Data Imports	199	Step 3. Select Format	236
TecDoc Supplier Data Importer	202	Example	246
Using TecDoc Supplier Data Importer	203	NAPA Application Exporter	254
Importing Supplier Data Files via Web UI	204	Automotive Business Rule Plugins	257
Import Process Overview	204	Business Action: Change Assembly	259
Using the TecDoc Supplier Data Importer in Web UI	204	Configuring the Change Assembly Business Action in Workbench	261
Configuring TecDoc Supplier Data Importer	210	Configuring the Change Assembly Business Action in Web UI	263
Configuring an IIEP for Supplier Data Imports	211	Business Action: Change Part Type	270
Configuring a File Loading Widget for Supplier Data Imports	217	Configuring the Change Part Type Business Action in Workbench	273
Exporting Automotive Data	220	Configuring the Change Part Type Business Action in Web UI	276
AutoCare ACES Application Exporter	221	Business Action: Change Part	283
Step 1. Start ACES Application Exporter	221	Configuring the Change Part Business Action in Workbench	287
		Configuring the Change Part Business Action in Web UI	290
		Business Action: Copy Application to Other Assembly	297
		Configuring the Copy Application to Other Assembly Business Action in Workbench ..	299
		Configuring the Copy Application to Other	301

Assembly Business Action in Web UI	
Business Action: Copy Application to Other Part	309
Configuring the Copy Application to Other Part Business Action in Workbench	313
Configuring the Copy Application to Other Part Business Action in Web UI	315
Business Action: Copy Applications to Related Parts	322
Part Relation Reference Type Configuration	322
Steps for Using the Business Action in Web UI	322
Configuring the Copy Applications to Related Parts Business Action in Workbench	325
Configuring the Reference Type	325
Configuring the Business Action	325
Configuring the Copy Applications to Related Parts Business Action in Web UI	328
Business Action: Move ACES Applications for PIES Part	331
Business Action: Set Condition Links on Part Types	332
Business Action: Set Import Status Attributes	333
Configuring Change Flags for New and/or Updated Objects	334
Use Case Appendix	338
Use Cases - Tracking Newly Created Objects	339
Use Cases - Tracking Deleted Objects	343
Use Cases - Tracking Changed Objects	346

Use Cases - Clearing Delete Status Flag	350
Business Action: Sync ACES Applications Between PIES Parts	353
Business Action: Validate Application Conditions and Start in Workflow	354
Configuring Business Action for Applications with Invalid Configuration	355
Business Condition: Check Path for Missing Application	359
Improving Search Result Accuracy	359
Configuring the Check Path for Missing Application Business Condition in Workbench	361
Business Condition: Validate Application on Import	367
Application Manager	370
Variations Across Standards	372
Intelligent Search Interface	373
Using Intelligent Search Interface	376
Using Search Boxes	380
Searching for Criteria	380
Selecting Valid Criteria	380
Removing Criteria	384
Minimum Search Criteria	384
Search Box Synchronization	386
Using Make/Model Search Box	387
Using Year Search Box	392
Using Options Search Boxes	393
Using Part Type Search Box	396
Saved Search Tool	398

Saving a Search	399	Advanced Filtering	
Renaming a Saved Search	400	Error Handling	433
Deleting a Saved Search	401	Workbench Configurations Related to Validation Path Functionality	435
Missing Application Coverage Functionality	402	Vehicle Configurations	436
Improving Part Type Accuracy	404	Base Vehicle Applications	438
Application Coverage Report	405	Options on Applications	439
Application Coverage Report Results ..	405	Options / Conditions Stored as Attributes	440
Running an Application Coverage Report	406	Options / Conditions Stored as References	441
Adding the Application Coverage Report to a Background Processes Screen	409	Configuring Application Manager	443
Results Table and Toolbar	411	Own Model Considerations	443
Results Table Default Columns	418	Configuring an Application Manager Screen in Web UI	444
Application Condition Header Components	419	Configuring Make/Model Search Box	453
Controlling Display of Conditions in Application Manager	423	Configuring Attribute Search Box	455
Creating Applications	424	Configuring Options Group Search Box ..	457
Using Multi-Select	426	Configuring Part Type Search Box	463
Automotive Validation Path Functionality ...	427	Editing the Part Type Search Box Label	463
Validation Path Functionality and Search Box Criteria	428	Configuring Year Search Box	465
Validation Path Functionality and Options on Applications	430	Competitor and OE Numbers Solution	467
Automatic Population of Options on Applications	430	Adding Part Number References Manually ..	470
Options Within the Value Editor	430	Adding Part Number References from Suggestions	472
Validation Path Functionality and Missing Applications	431	Removing Existing Part Number References	474
Validation Path Functionality and	432	Removing Part Number Reference Suggestions	476
		Creating and Referencing New Competitor and OE Part Numbers	478

Normalization / Harmonization of Part Numbers	481
Configuring the Part Number References Component	483
Configuring the Part Number Suggestions Component	488
Configuring the Advanced Part Number References Component	495
Configuring the Advanced Part Number Suggestions Component	501
Configuring the Data Model for Application Competitor or OE Part Numbers Table Header	512
Configuring the Application Competitor or OE Part Numbers Table Header Component	518
Configuring the Metadata Attribute References Component	523
Country Handling Solution	528
Using Country Handling	529
Using Country Handling within Application Manager	529
Using Country Handling within Node Editor	531
Using Country Handling Value Editor to Add or Edit Country Handling Data	536
Configuring Automotive - Country Model Component Model	540
Configuring Application Country Handling Table Header	542
Configuring Country Handling Value Component	545
Application Suggestion Solution	548
Adding New Applications Manually	551

Adding Applications from Suggestions	554
Removing Existing Applications	557
Removing Application from Suggestions List	559
Configuring the Part Application Editor Component	561
Configuring the Application Suggestions Component	565
Configuration	565
Data Onboarding Solution	574
Configuring Data Onboarding Solution	575
Data Onboarding Solution Initial Setup	576
Configuring the Execute Mapper Configuration Business Action	581
Configuring Web UI for Data Onboarding Solution	583
Prerequisites	583
Onboarding Mappings Details Screen	583
Mapper Configurations - Stack Panel Component	583
Onboarding Mappings Details Screen	584
Configuration Prerequisites	584
Configuration Process	584
Implementing the Onboarding Mappings Details Screen	587
Configuring the Stack Panel	587
Configuring the Mapper Configurations to Display Onboarding Mappings Details Screen	591
Configuring Mapper Configuration Setup Entity	596
Prerequisites	596

Creating Mapper Configuration Setup Entity in Workbench	596	Object to Object Mapping Use Cases ...	648
Creating Mapper Configuration Setup Entity in Web UI	597	Use Cases - Copy Attribute Values from One Object to Another Object	649
Configuring Mapper Configuration Setup Entity in Web UI	598	Example One	649
Using the Onboarding Mappings Details Screen	602	Use Cases - Copy Data from One Object to Another Object	659
Creating Mapping Plugins	602	Example One	659
Managing Mapping Plugins	604	Example Two	670
Deleting an Existing Mapping Plugin ..	604	Use Cases - Copy Data from One Object to Child Object of Another Object	680
Executing Mapper Configuration Setup Entity	606	Example One	680
Prerequisites	606	Example Two	690
Executing by a Business Rule	606	Use Cases - Copy Data from One Object to a Referenced Object of Another Object	702
Executing by a Background Process	606	Example One	702
Create a business action to set the Mapper Configuration to execute after import	607		
Configure the business action in the Import Workflow	608		
Mapping Plugins	612		
Attribute Mapping Plugin	613		
Testing Transformations Configured in Attribute Mapping Plugin	626		
Object to Object Mapping Plugin	628		
Mapping Validation Path Functionality ..	641		
Business Action Mapping Plugin	643		
Mapping Plugins Use Case Overview	647		
Use Case Appendix	648		
Attribute Mapping Use Cases	648		

Automotive Reference Guide Introduction

This guide describes specific Product MDM (PMDM) for Automotive reference material beyond what is provided in the **Automotive Quick Start Guide**. It is expected that readers are familiar with the material in that guide as it is not generally repeated within this guide.

Patch Recipe and Compatibility

The Automotive 9.2 add-on can be installed with the following recipe:

```
to:automotive/7.0/automotive-7.0.21.spr
```

Note: The above recipe is only compatible with the STEP 9.2-MP1 baseline (`to:step/trailblazer/step-9.2-mp1.spr`).

Important: Once an Automotive add-on is installed to a base STEP system, the base system should not be upgraded without upgrading the Automotive add-on at the same time. Additionally, when upgrading any base STEP system that has any Automotive add-on installed, both install recipes must be prepared at the same time.

For assistance in applying the patch to systems with extensions or additional add-on components, contact Stibo Systems Technical Services.

Supported Versions and Formats

The following are the supported import and export versions and/or formats for the various automotive standards.

Note: If there is only one format for the data type and the format itself is not versioned, 'Format is not versioned' is listed.

Generic

- **Asset Exporter:** Format is not versioned (generic exporter not specific to any standard).

AutoCare

- **ACES Exporter:** ACES 3.0, ACES 3.2 and ACES 4.0
- **ACES Importer:** All ACES 3.X versions are supported for import, though all are validated against the 3.2 schema.

Note: Changes between the various 3.X versions are minimal so in most cases this will result in a successful import. However, the schema for the 'ApprovedFor' element in the header changed with 3.2 so attempting to load a file for an earlier version that includes the 'ApprovedFor' tag will fail schema validation. This can be corrected by removing the 'ApprovedFor' element from the header, or manually updating the version to be 3.2.

- **Brand Table Importer:** All three flat file formats are supported (original, with revision date, and with sub-brands and revision date)
- **PAdb Importer:** ASCII
- **PCdb Importer:** ASCII
- **PIES Exporter:** PIES 6.5, PIES 6.7 and PIES 7.0
- **PIES Importer:** PIES 6.5, PIES 6.7 and PIES 7.0
- **Qdb Importer:** ASCII
- **VCdb Importer:** ASCII

NAPA

- **Application Exporter:** Format is not versioned
- **Application Importer:** Format is not versioned
- **Asset Reference Exporter:** Format is not versioned
- **MPCC / Attribute Exporter:** Format is not versioned
- **MPCC / Attribute Importer:** Format is not versioned
- **Interchange Exporter:** Format is not versioned

- **Interchange Importer:** Format is not versioned
- **Translation Importer:** Format is not versioned
- **Valid Vehicles Importer:** Format is not versioned

TecDoc

- **Reference Data Importer:** TAF 2.4 using either the ZIP or 7z formats
- **Supplier Data Exporter:** TAF 2.4
- **Supplier Data Importer:** TAF 2.4 using either the ZIP or 7z formats

Importing Automotive Data

The Automotive solution provides extensive import capabilities. Information within this section applies to all of the automotive standards.

Before importing automotive data, it is recommended that you have an understanding of the following:

- Import Framework
- Default Workflow States and Functions
- Modifying Import Framework
- Displaying Import Modifications in Web UI
- ID Structures in Importers
- Adding Reporting Extensions to Imports
- Validation Error Handling

Information on each of the available automotive standard specific importers is addressed in the following sections:

- AutoCare Importers
- TecDoc Reference Data Importer

For information on supported versions, see the **Supported Versions and Formats** topic.

For general information on the way a user initiates and manages automotive imports, see the **Using Automotive Importers** section of the **Automotive Quick Start Guide**.

Import Framework

The intention of the import aspect of the Automotive core solution is to provide out-of-the-box importers for the various automotive standards, for which each customer can then apply their own validations, business processes, and data management procedures. To do this successfully, it is crucial to understand the import framework, which includes the integration endpoints and workflows created by Easy Setup (as described in the **Quick Start Setup for Admins** section of the **Automotive Quick Start Guide**).

This section details specifics of the Import Framework, which is applicable to all Automotive importers across all standards. For more information on the available automotive importers, see the **Supported Versions and Formats** topic.

Import Process Overview

Once a valid automotive data file is uploaded to a hotfolder on the application server (optionally using a File Loading Widget), the file is picked up from the hotfolder by an IIEP, and the IIEP creates an Entity in STEP that represents the file. This Entity object is called the 'controller' and contains basic data about the file and the file's status in the workflow. Web UI users are able to monitor the import status using a Status Selector Widget and an Import Controller Screen. Once an import file reaches the 'Ready for Import' state, then users can start the import by clicking on the 'Start import' button within the Control Panel screen. At that time, the BGP service (that runs as part of the Import state) allows for the configured business rules to act on the objects being imported.

For example:

Tree		AAIA VCdb2009 ASCII Complete VCDB 20170127	
		Import Flow Controller Type	References Referenced By Status State Log Tasks
Description			
Name	>	>	Value
ID	>		Controller-100347
Name	>		AAIA VCdb2009 ASCII Complete VCDB 20170127.zip
Object Type	>		Import Flow Controller Type
Revision	>		0.1 Last edited by VCDBIMPORT on Mon Apr 24 14:12:15 EDT 2017
Path	>		Entity hierarchy root/Import Flow Root/VCdb Data/AAIA VCdb2009 A
Automotive Import Flow State BGP	>	abc	<pre><?xml version="1.0" encoding="UTF-8"?> <StringMap> <Entry Key="Validation" Value="BGP_100348"/> <Entry Key="Import" Value="BGP_100351"/> <Entry Key="Conversion" Value="BGP_100349"/> <Entry Key="DeltaCalculate" Value="BGP_100350"/> </StringMap></pre>
Import Flow File Type	>	abc	VCDBData
Import Flow Overall Status	>	abc	Completed with errors: 12
Import Flow State Status	>	abc	<pre><?xml version="1.0" encoding="UTF-8"?> <StringMap> <Entry Key="Validation" Value="Validation completed"/> <Entry Key="Import" Value="Completed with errors: 12"/> <Entry Key="Conversion" Value="Conversion completed"/> <Entry Key="DeltaCalculate" Value="Delta calculation completed"/> </StringMap></pre>
Import Flow Workflow ID	>	abc	AutoCareVCdbImport

As the Entity moves through the import workflow, a series of background processes handle the various processing and import activities.

- The original file name is recorded as the STEP Name of the controller entity.
- The IDs of the background processes are stored in the Automotive Import Flow State BGP attribute.
- The Import Flow State Status attribute is also noteworthy as it stores the status of each process, as opposed to the Import Flow Overall Status attribute which displays a global status of the file (rather than a per-process status).

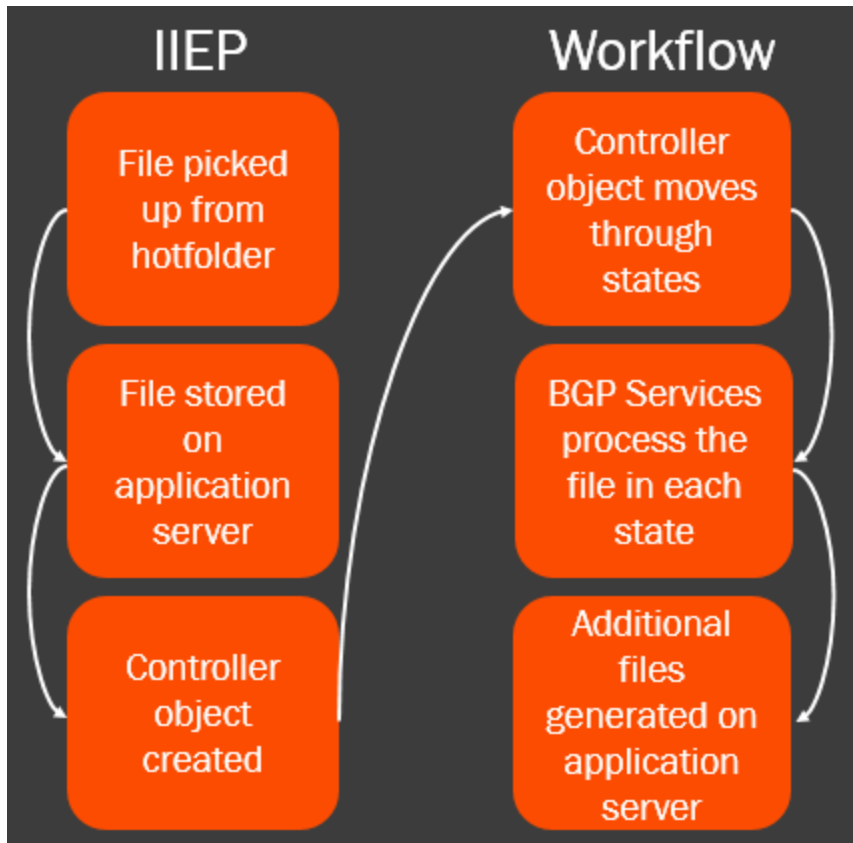
Note: All of the information displayed on the controller Entity is also displayed within the Web UI Import Controller Screen, which are discussed in the **Quick Start for Users** section of the **Automotive Quick Start Guide**.

Once created, the controller is initiated into the workflow associated with the importer, and the work of the endpoint stops. From there, the workflow takes over processing of the file via a series of states using business rules and background processes to carry out the processing of the file.

Important: It is critical to understand that it is only the controller Entity that is in the workflow - the objects being acted on (created / updated / deleted) via information supplied in the import file are not in the workflow. Therefore, running standard business actions acting on current object will impact the controller Entity only, not the objects in the input file.

Important: In order for the business action to apply to the object(s) that are getting imported from the input file, the business action needs to be added to the 'Import action' parameter in the Background Process Service Action that runs on the 'Import' state.

At a high-level, the interaction between the endpoint and the workflow is as follows:

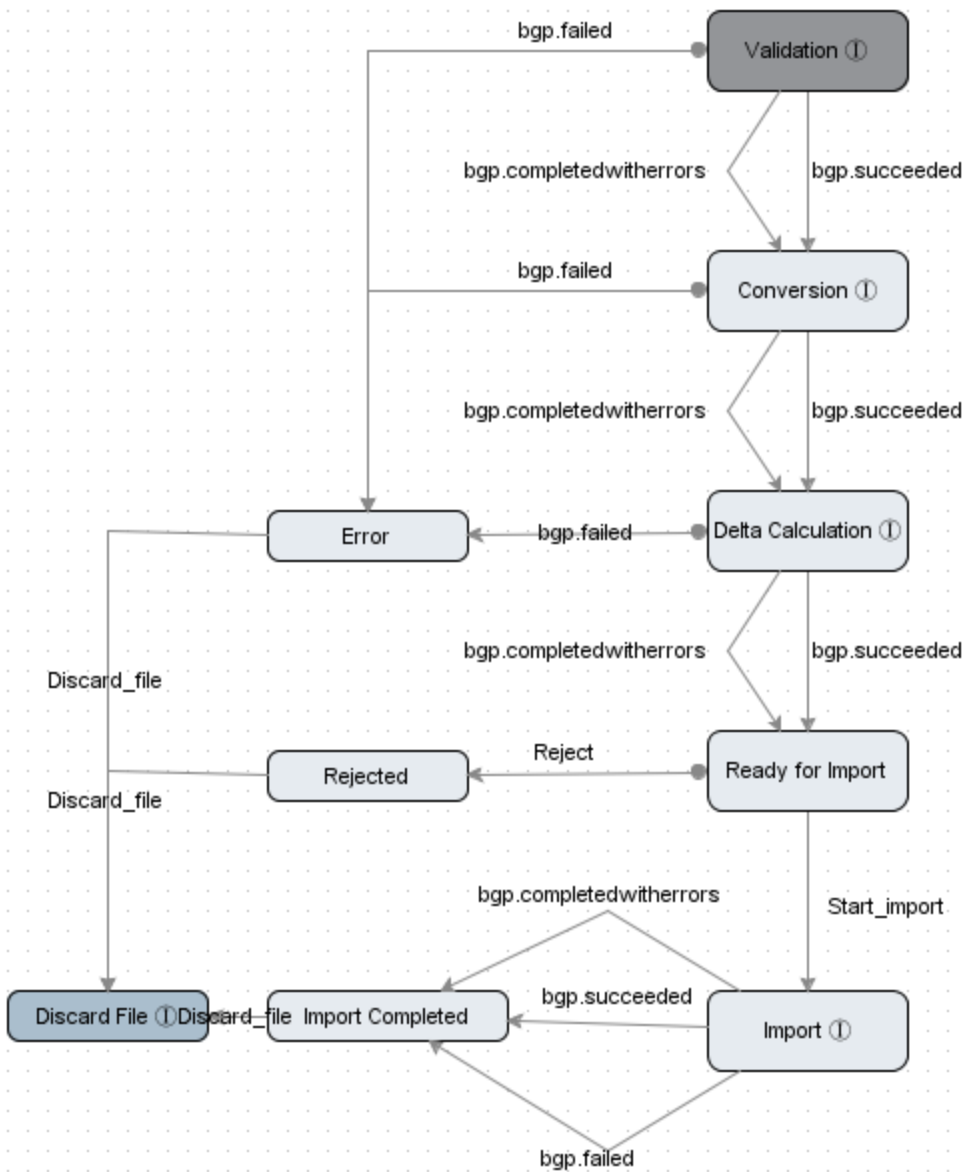


Each import has an associated workflow and all proceed through the same states by default. However, it is intended that customers will expand on the existing states and actions to add their own validations, reporting, and additional processing as needed.

Important: A new state can be inserted at any point in the workflow, and additional rules can be added to any of the existing states. Additionally, the ImportFlowExtension interface in the Extension API can be used to create new background processes to handle what the state should do.

Default Workflow States and Functions

The default workflow and processing that occurs in each state is displayed within the image below.



More details about each default workflow state can be found in the following sections:

- Validation State
- Conversion State
- Delta Calculation State
- Error State
- Ready for Import State

- Rejected State
- Import State
- Import Completed State
- Discard File State

Validation State

The validation state does some basic schema validation of the file. If the automotive standard (i.e., AutoCare, TecDoc, NAPA) supplies an XSD, the file format is validated against the XSD. If an XSD is not provided by the standard, the file is loosely validated to ensure the format conforms to what is expected for the file type.

Information on each of the available automotive standard specific importers is addressed in the following sections:

- AutoCare Importers
- TecDoc Reference Data Importer

The validation state includes one business action (**Run background process action**) by default, which runs the validation service as a background process. Each standard has its own validation service.

Important: The validation services are coded specifically for the file types provided for that standard and cannot be repurposed for other formats, even within the standard. For example, the AutoCare validation service accepts PCdb files as zipped files containing pipe delimited subfiles and ACES files in XML format. If you wanted to validate a PCdb file in Access format or a flat file supplying ACES data, a new validation service would need to be created using the Extension API.

Validation State Common Parameters

Despite having different services for each standard, each validation service contains several common parameters.

The AutoCare validation service (AutoCareValidationService) is shown as an example below.

The screenshot shows a dialog box titled "Edit Operation" with a close button (X) in the top right corner. Inside the dialog, there is a dropdown menu at the top set to "Run background process action". Below this, there are three main sections:

- 1** QueueID: A text input field containing "queue_Validation".
- Background Process Service: A dropdown menu showing "[AutoCareValidationService]".
- Service parameters: A sub-dialog containing:
 - 2** AutoCare file attachment ID: A text input field containing "original".
 - 3** Continue on Error: An unchecked checkbox.

At the bottom of the dialog are "Save" and "Cancel" buttons.

1. **QueueID:** For all three validation services, this required parameter value defaults to 'queue_Validation' for all background process actions that specify the queue in which the background process should run.

Note: This parameter is specified for the action itself, not the particular service within the action, though the outcome is the same as each action runs only a single service.

Additionally, this parameter value can be used to define the configuration property within the sharedconfig.properties for when a process needs to run in parallel.

For example, by default, files go through the following states one at a time; validation, conversion, delta calculation, and import. If it is necessary for ACES files to be imported four at a time, then the following would need to be added to the sharedconfig.properties:

```
BackgroundProcess.Queue.queue_Import.Size=4
```

Typically, reference files do not need to have parallel processes because they are only updated once a month, and it is important to have certain reference files imported before others. Therefore, running them in parallel is not ideal.

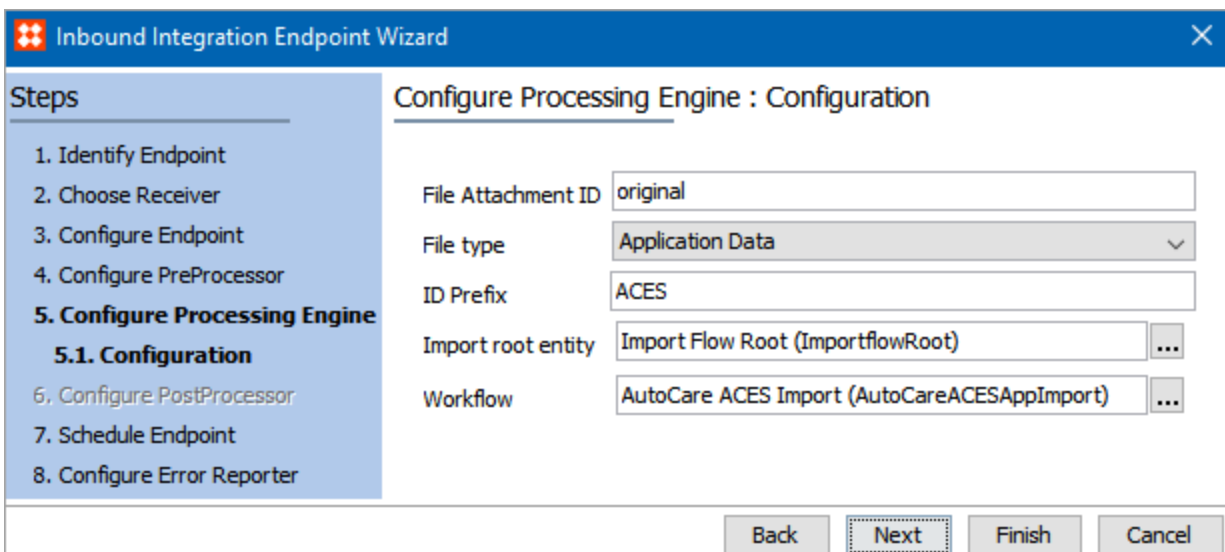
However, it is likely that multiple supplier files would need to be imported at once, because they are updated more frequently. Therefore, running them in parallel is ideal.

Note: After adding the property, the application server must be restarted for it to take effect.

2. **AutoCare file attachment ID / NAPA file attachment ID / Attachment ID for TAF file:** For all three validation services, this required parameter value:

- Defaults to 'original.'
- Forms the relationship between the file supplied to the endpoint and the file that the workflow is going to process.
- Prefixes the file name for the workflow attachment.

Important: The 'AutoCare file attachment ID' / 'NAPA file attachment ID' and/or 'Attachment ID for TAF file' parameter entry in the validation service must match the File Attachment ID value in the corresponding endpoint configuration. For example, the 'AutoCare file attachment ID:' value is 'original,' and can be viewed in both the Edit Operation dialog (shown above) and the Inbound Integration Endpoint Wizard (shown below).



3. **Continue on Error:** Specifies whether or not the file can continue to process after encountering some 'allowable' data errors. For additional information and considerations relevant for this setting, see the 7. **Determine Validation Error Handling for Each Import** topic within the **Automotive Quick Start Guide**.

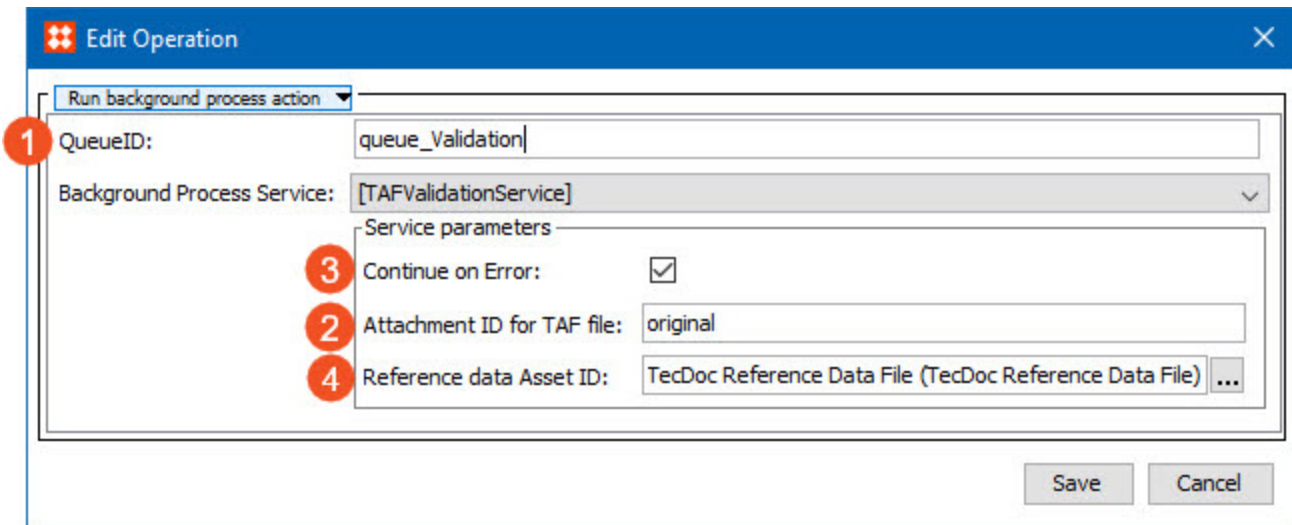
Validation State TecDoc Specific Parameter

Along with the common parameters (described above) the TecDoc validation service (TAFValidationService) offers the following additional parameter:

4. **Reference data Asset ID:** Required parameter available only for the TecDoc validation service. When importing a Reference Data file, the file itself is stored in the selected asset.

Important: When importing a Supplier Data file, the Supplier Data file is validated against the Reference Data file stored in the selected 'Reference data Asset ID.'

The TecDoc validation service (TAFValidationService) is shown as an example below.



To view the import file within Workbench, go to Tree > Configurations > TecDocConfigurations.

In the example below, the TecDoc Reference Data File is displayed below the TecDocConfigurations classification folder.

Tree

- Assets
- AutoCare Root
- Configurations
 - Import Configurations
 - Export Configurations
 - Asset Push Configurations
 - Auto Classification Rule Sets
 - Portal Configurations
 - Smart Sheet Configurations
 - TecDocConfigurations
 - TecDoc Reference Data File**
 - Asset Export Delta (ACES Delata)
- Index Words
- NAPA Reference Data
- TecDoc
- TecDoc Assets Root
- Import Flow Root
- TecDoc Manufacturer Root
- TecDoc Resource Root
- TecDoc Supplier Price Lists
- Publications

TecDoc Reference Data File rev.9.0 - Images & Documents

Images & Documents | References | Referenced By | Status | State Log | Tasks

Description

Name	Value
ID	TecDoc Reference Data File
Name	TecDoc Reference Data File
Object Type	ZIP
Revision	9.0 Last edited by TECDOCREFERENCEDATAIMPORT on Tue Jul 31 12:44:18 EDT 2018
Approved	✘ Never Been Approved
Translation	Not Translated
Path	Classification 1 root/Configurations/TecDocConfigurations/TecDoc Reference Data File

System Properties:

Name	Value
Extension	abc zip
Filename	abc REFERENCE_DATA_0618.zip
Format	abc Zip (Zip archive)
MIME Type	abc application/zip
Size	abc 50,862,472
Upload Time	abc 2018-07-31 12:44:18

Validation Service Function Details

As the validation service runs a background process (BGP), a corresponding BGP folder is created on the application server at `/workarea/background-processarea/[Standard]ValidationService`. The BGP folder contains a sub-folder whose name matches the STEP ID of the controller entity that is moving through the workflow. The original file that was loaded can be found inside the controller sub-folder, as shown below.

```
work.area/background-processarea/AutoCareValidationService/BGP_103896/Controller-103895
```

Name
↑ ..
📄 Acme_MMY.xml

Further, the file is prefixed with the value from the file attachment ID parameter in the BGP service ('original' in our example) and deposited on the application server at `/workarea/stepworkflow-area/[Standard][Format]Import/entity-[ControllerID]/attachments/ImportFlowAttachment`, as shown below.

```
work.area/stepworkflow-area/AutoCareACESApplImport/entity-Controller-103895/attachments/ImportFlowAttachment
```

Name
↑ ..
📄 original-Acme_MMY.xml

Note: If modifications need to be made to the original file as part of validation and/or following validation, it is the workflow attachment file (e.g., original-Acme_MMY.xml in the above example) that must be manipulated as this is the file that is actually carried through the workflow.

Validation State Results

The following are possible validation state results:

Failed: If the import file fails validation (e.g., does not meet the schema requirements or the system encounters an error and is unable to complete the validation process), the controller object in the workflow is sent to the Error state via the `bgp.failed` transition. This transition includes one default business rule which is used to populate the overall status of the controller entity (the overall status is displayed to end users in the Web UI Import Controller Screen). The controller will remain in the Error state until acted on by a user.

Important: Though an import can have a status of 'Validation failed,' the controller object in the workflow can still be sent to the next step of the workflow if the 'Continue on Error' parameter is enabled for the workflow. When this occurs, the validation issues are written to the execution report, but all valid data is converted and made available for the import process. For more information, see the **7. Determine Validation Error Handling for Each Import** topic within the **Automotive Quick Start Guide** and/or the **Validation Error Handling** topic within this guide.

Succeeded: If validation completes successfully without any errors, the controller automatically moves to the Conversion state via the `bgp.succeeded` transition. This transition does not include any default business rules as the controller is moved automatically so there is no need to display an overall status to the end user at this time.

Completed with errors: It is possible for a file to pass validation with errors. In this case the controller is moved to the next state via the `bgp.completedwitherrors` transition that, like the `bgp.succeeded` transition, does not contain any default business rules. This occurs when the basic schema validation is met, but some additional data-level validations are in place for which the file does not pass, and the 'Continue on Error' parameter is checked, as described in the **7. Determine Validation Error Handling for Each Import** topic of the **Automotive Quick Start Guide**. If the 'Continue on Error' parameter is unchecked, the data-level validations would result in a validation failure and the file would be moved to the Error state via the `bgp.failed` transition.

In the example below a TecDoc Reference Import Controller Screen is shown, and because the 'Continue on Error' parameter is enabled when the import processes, the import file associated with Controller-102344 is able to be Completed with errors.

TecDoc Reference Imports					
Process	File Name	Overall Status			
Controller-100349	REF_DATA_0318_minus_Table_026_044.zip	Completed with errors: 27243			
Controller-102344	REF_DATA_0318_minus_Table_026_044.zip	1 Completed with errors: 255			

1-8 of 8

Import Details

Process	Started Time	Duration	Started By	Status	Background Process Link
Validation	2018-07-31 10:03:40	25 secs	STEPSYS	2 Validation completed	3 completedwitherrors
Conversion	2018-07-31 10:04:05	1 min 30 secs	STEPSYS	Conversion completed	succeeded
Delta Calculation	2018-07-31 10:05:35	2 mins 39 secs	STEPSYS	Delta calculation completed	succeeded
Import	2018-08-01 14:25:54	10 mins 50 secs	STEPSYS	Completed with errors: 255	completedwitherrors

1-4 of 4

This is displayed within the Import Controller Screen, in the following manner:

1. The Overall Status column for controller is displayed as 'Completed with errors: 255.'
2. The Status column within the Import Details table displays the Status as 'Validation completed.'
3. The Background Process Link displays 'completedwitherrors.'

Conversion State

The conversion state converts the original file into a series of STEPXML files. This provides two benefits:

1. By converting to STEPXML, the import can be carried out using standard STEP import functionality.
2. By converting to multiple standalone files rather than a single large file, there is a performance gain in that some files can be imported in parallel.

By default, the conversion state includes one business action (**Run background process action**), which runs the conversion service as a background process. Each standard has its own conversion service and some parameters are specific to the standards. Each standard's Conversion state parameters are described below, along with the possible conversion state results.

AutoCare Conversion State Parameters

Within the STEP AutoCare solution, two conversion services are offered; [AutoCareToStepXMLConvertService] and [ACESToStepXMLConvertService]. [AutoCareToStepXMLConvertService] is used in the workflow Conversion state for the Brand, PAdb, PCdb, VCdb, Qdb and PIES imports. The [ACESToStepXMLConvertService] is used only in the workflow Conversion state for the ACES import.

[AutoCareToStepXMLConvertService]

Each of the parameters are explained in detail below.

The screenshot shows a dialog box titled "Edit Operation" with a close button (X) in the top right corner. The dialog contains a dropdown menu at the top set to "Run background process action". Below this, there are several input fields and a dropdown menu:

- QueueID:** A text input field containing "queue_Conversion".
- Background Process Service:** A dropdown menu currently showing "[AutoCareToStepXMLConvertService]".
- Service parameters:** A section containing two more text input fields:
 - AutoCare file attachment ID:** containing "original".
 - STEP XML attachment ID:** containing "convert".

At the bottom right of the dialog, there are two buttons: "Save" and "Cancel".

- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Conversion'. This parameter is specified for the action itself, not the particular service within the action, though the outcome is the same as each action runs only a single service.
- **Background Process Service:** Required parameter for all background process actions that specifies the background process, which defaults to [AutoCareToStepXMLConvertService].
- **AutoCare file attachment ID:** Required parameter used in all conversion services and functions the same as described in the Validation process (forming the relationship between the file originally supplied to the endpoint and the file that the workflow processes will act on). For more information, see the **Validation State** topic.

- **STEP XML attachment ID:** Required parameter used in all conversion services and specifies the name of the file that is the product of the conversion process, which defaults to 'convert.' If multiple files are created by the conversion service, this will be a zipped file, e.g., convert.zip.

[ACESToStepXMLConvertService]

Each of the parameters are explained in detail below.

The screenshot shows the 'Edit Operation' dialog box with the following configuration:

- Run background process action: [Dropdown menu]
- QueueID: queue_Conversion
- Background Process Service: [ACESToStepXMLConvertService]
- Service parameters:
 - ACES Default FULL Import handling: BRAND
 - ACES Default SUPPLIER Import handling: NO
 - ACES Default UPDATE Import handling: STANDARD
 - AutoCare file attachment ID: original
 - Create PIES items:
 - Omit application names:
 - STEP XML attachment ID: convert
 - Regular Expression for parsing Supplier from file name: (.*)_.*

Buttons: Save, Cancel

- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Conversion'. Note that this parameter is specified for the action itself, not the particular service within the action, though the outcome is the same as each action runs only a single service.
- **Background Process Service:** Required parameter for all background process actions that specifies the background process, which defaults to '[ACESToStepXMLConvertService]'.
- **ACES Default FULL Import handling:** By default, BRAND is selected. Defines how files with a SubmissionType=FULL are handled. For more information, see the **9. Specify Complete Replacement Handling (ACES Import Workflow Only)** topic within the **Automotive Quick Start Guide**.
- **ACES Default SUPPLIER Import handling:** By default, NO is selected. Defines whether or not Supplier is taken into consideration when evaluating replacement scenarios. For more information, see the **9. Specify Complete Replacement Handling (ACES Import Workflow Only)** topic within the **Automotive Quick Start Guide**.

- **ACES Default UPDATE Import handling:** By default, STANDARD is selected. Defines how files with a SubmissionType=UPDATE are handled. For more information, see the **9. Specify Complete Replacement Handling (ACES Import Workflow Only)** topic within the **Automotive Quick Start Guide**.
- **AutoCare file attachment ID:** Required parameter used in all conversion services and functions the same as described in the Validation process (forming the relationship between the file originally supplied to the endpoint and the file that the workflow processes will act on). For more information, see the **Validation State** topic.
- **Create PIES items:** Optional parameter used to import applications when associated part number (PIES Item) does not exist in the system. Selecting the parameter allows an ACES file to be imported even if the Part Number does not already exist. The Part Terminology product hierarchy (PIES Brand/PIES PCdb Category/PIES PCdb Sub Category/PIES PCdb Part Terminology/PIES Item) and Part Number will be created by the ACES import, and applications will be created below the Part Number. Having the parameter deselected would require that the Part Number must already exist. With this parameter deselected and if the Part Number doesn't exist, then the ACES import will not be able to create any applications because of the missing Part Number.
- **Omit application names:** By default, this optional parameter specific to the AutoCare standard is disabled and when the ACES Importer creates application objects it sets the STEP Name to be the same value as the Part Number. After the applications are created, users have the option to change the STEP Name to something other than the Part Number (i.e., Acura TSX 2010). However, an issue is encountered the next time an ACES file containing those same applications is imported, because the STEP Name will be reverted to the Part Number. This occurs because the Conversion state checks the name in the file against the name in the database, and processes them as an update. Therefore, it includes those records in the delta file, even though nothing has changed in the record. If the application name and part number should always match, then the parameter should remain disabled. This way, upon the next import of an ACES file, the Part Number's STEP Name will be reverted to the Part Number's name within the ACES file. If the application name should not be altered to match what is in the ACES file then When enabled, application names will be omitted during the conversion process so that the record will not be included in the delta file.
- **STEP XML attachment ID:** Required parameter used in all conversion services and specifies the name of the file that is the product of the conversion process, which defaults to 'convert.' If multiple files are created by the conversion service, this will be a zipped file, e.g., convert.zip.
- **Regular Expression for parsing Supplier from file name:** Optionally, populate the 'Regular Expression for parsing Supplier from file name' parameter (using the JavaScript regular expression format) to specify how the Supplier value should be extracted from the ACES file name. By default, the parameter is populated with `(.*?)_*`, meaning that the system will extract the first string prior to an underscore (typically a Supplier value) in the file name as this is how the ACES standard defines the naming convention. For example, with a file name of `ACMESupply_BrakeHardware_2016-04-23_FULL.xml`, 'ACMESupply' would be extracted as the Supplier value. If this is not desired, a Regular Expression can be inserted in the parameter to extract Supplier in an alternate way.

NAPA Conversion State Parameters

Within the STEP NAPA solution, two conversion services are offered; [NAPAToStepXMLConvertService] and [NAPATransToStepXMLConvertService]. 'NAPAToStepXMLConvertService' is used in the workflow Conversion state for Vehicle, Attribute, Application, and Interchange imports. The

'NAPATransToStepXMLConvertService' is used only in the workflow Conversion state for Translation import.

[NAPAToStepXMLConvertService]

Each of the parameters are explained in detail below.

- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Conversion'. Note that this parameter is specified for the action itself, not the particular service within the action, though the outcome is the same as each action runs only a single service.
- **Background Process Service:** Required parameter for all background process actions that specifies the background process, which defaults to 'NAPAToStepXMLConvertService'.
- **NAPA file attachment ID:** Required parameter used in all conversion services and functions the same as described in the Validation process (forming the relationship between the file originally supplied to the endpoint and the file that the workflow processes will act on). For more information, see the **Validation State** topic.
- **STEP XML attachment ID:** Required parameter used in all conversion services and specifies the name of the file that is the product of the conversion process, which defaults to 'convert.' If multiple files are created by the conversion service, this will be a zipped file, e.g., convert.zip.

[NAPATransToStepXMLConvertService]

Each of the parameters are explained in detail below.

- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Conversion'. Note that this parameter is specified for the action itself, not the particular service within the action, though the outcome is the same as each action runs only a single service.
- **Background Process Service:** Required parameter for all background process actions that specifies the background process, which defaults to 'NAPATransToStepXMLConvertService'.
- **NAPA file attachment ID:** Required parameter used in all conversion services and functions the same as described in the Validation process (forming the relationship between the file originally supplied to the endpoint and the file that the workflow processes will act on). For more information, see the **Validation State** topic.
- **STEP XML attachment ID:** Required parameter used in all conversion services and specifies the name of the file that is the product of the conversion process, which defaults to 'convert.' If multiple files are created by the conversion service, this will be a zipped file, e.g., convert.zip.

TecDoc Conversion State Parameters

Within the STEP TecDoc solution, the [TAFTToStepXMLConvertService] conversion services is offered.

[TAFTToStepXMLConvertService]

Each of the parameters are explained in detail below.

- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Conversion'. Note that this parameter is specified for the action itself, not the particular service within the action, though the outcome is the same as each action runs only a single service.
- **Attachment ID for TAF file:** Required parameter used in all conversion services and functions the same as described in the Validation process (forming the relationship between the file originally supplied to the endpoint and the file that the workflow processes will act on). For more information, see the **Validation State** topic.
- **Generic article filter:** Optional parameter specific to the TecDoc standard. This filter option allows for the reduction of unnecessary Generic Articles (Classifications) being imported into STEP. Only those Generic Articles selected within the filter are imported.

In the example below, the 0003 Product Hierarchy is shown including many Articles, the editing of the 'TecDoc Reference Import Conversion' Business Rule is displaying the 'Generic article filter' parameter with the 'Air Mass Sensor' and 'Glow Plug' generic articles selected for import, and the results are the 0003 Product Hierarchy is shown with only the 'Air Mass Sensor' and 'Glow Plug' articles.

Supplier Data without Generic Article filtering

- [-] TecDoc Supplier Root
 - [+] 0003 Product Hierarchy
 - [+] 0003 Addresses
 - [+] 0003 Article Descriptions
 - [-] 0003 Articles
 - [+] Air Mass Sensor
 - [+] Glow Plug
 - [+] Ignition Cable
 - [+] Ignition Cable Kit
 - [+] Ignition Coil
 - [+] Lambda Sensor
 - [+] NOx Sensor, NOx Catalyst
 - [+] Sensor, exhaust gas temperature
 - [+] Sensor, intake manifold pressure
 - [+] Spark Plug
 - [+] 0003 Text Blocks
 - [+] Competitor Numbers Pool
 - [+] OENumbers Pool
 - [+] TD_TradeNumbersPool

Edit Operation ✕

Run background process action ▼

QueueID:

Background Process Service: [TAFToStepXMLConvertService] ▼

Service parameters

Attachment ID for TAF file:

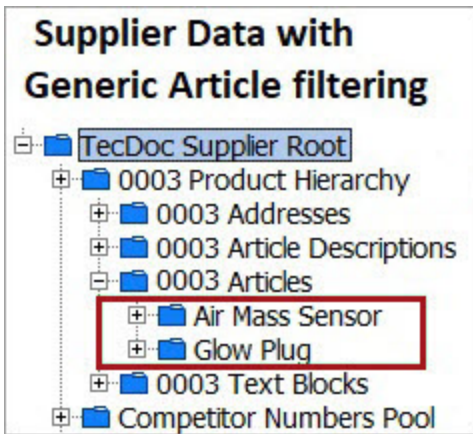
Air Mass Sensor (TD_GENART_03926)	...	✕
Air Mass Sensor (TD_USAGE_2240)	...	✕
Glow Plug (TD_GENART_00243)	...	✕
Glow Plug (TD_USAGE_0974)	...	✕



Generic article filter:

Reference data asset ID: ...

Attachment ID to store STEP xml to:

Omit search structures:



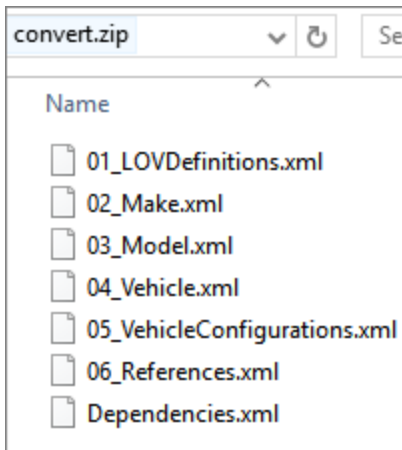
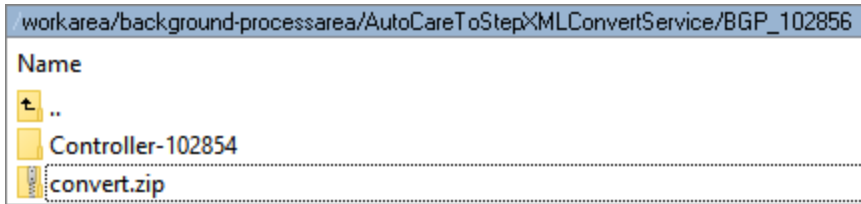
To select a Generic Article, click the green icon () within the parameter, and a blank Classification field will display. Click the ellipsis button () and a Select Classification dialog will display. Select the desired classification, and click the Select button to close the dialog. Repeat these steps to add more Generic Articles.

- **Reference data asset ID:** Required parameter used in all TecDoc conversion services that functions the same as described for the validation service (providing a link between the file being imported and the reference data asset it should be stored in or validated against). For more information, see the **Validation State** topic.
- **Attachment ID to store STEP xml to:** Required parameter used in all conversion services and specifies the name of the file that is the product of the conversion process, which defaults to 'convert.' If multiple files are created by the conversion service, this will be a zipped file, e.g., convert.zip.
- **Omit search structures:** Optional parameter specific to the TecDoc standard. This filter option allows for the reduction of unnecessary TecDoc Reference data being imported. When disabled, all Search Trees and Reference Data classification folders are imported per the TecDoc Reference Data file. When enabled, Search Trees and Reference Data classification folders (except Generic Articles and Linking Targets) are omitted. To filter Generic Articles and/or Linking Targets, the 'Generic article filter' parameter must be used. For example, if the 'Omit search structures' parameter is enabled, and the 'Generic article filter' is blank, then only the Generic Articles and Linking Targets Reference Data will be imported.

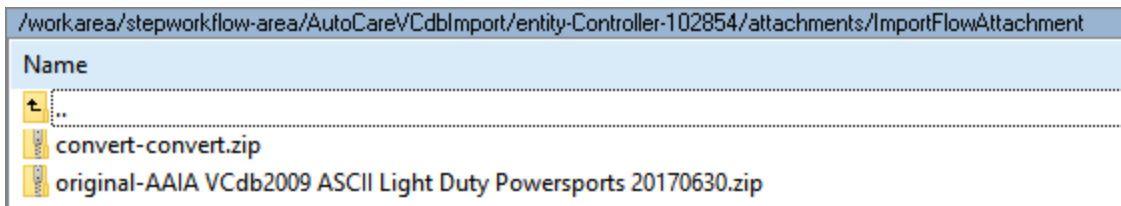
Conversion State Function Details

As the conversion service runs a background process (BGP), a corresponding BGP folder is created on the application server at /workarea/background-processarea/[Standard]toStepXMLConvertService. The BGP folder contains a sub-folder whose name matches the STEP ID of the controller entity that is moving through the workflow and contains the originally loaded file, the same as is seen in the validation service. The conversion service also writes an additional file, which is the output of the conversion process, (e.g., a zipped file with the name specified in the STEPXML attachment ID parameter). This zipped file contains sub-files in STEPXML format that are submitted to the delta calculation service for further processing. The sub-files contained in any convert file differ based on the file type.

For example, an AutoCare VCdb conversion result is shown below:



In addition, the converted file is prefixed with the value from the file attachment ID parameter in the BGP service ('convert' in our example) and deposited on the application server at /workarea/stepworkflow-area/[Standard][Format]Import/entity-[ControllerID]/attachments/ImportFlowAttachment. For example:



Note: If modifications need to be made to the converted file(s) before further processing is completed, it is the workflow attachment file (e.g., convert-convert.zip in the above example) that must be manipulated as this is the file that is actually carried through the workflow.

Conversion State Results

The following are possible conversion state results:

- **Failed:** If the import file fails conversion for some reason, the controller object in the workflow is sent to the Error state via the bgp.failed transition. This transition includes one default business rule which is used to populate the overall status of the controller entity (the overall status is displayed to end users in the Web UI). The controller will remain in the Error state until acted on by a user.
- **Succeeded:** If conversion completes successfully without any errors, the controller automatically moves to the Delta Calculation state via the bgp.succeeded transition. This transition does not include any default business

rules as the controller is moved automatically so there is no need to display an overall status to the end user at this time.

- **Completed with errors:** It is also possible for a file to complete conversion, but with errors. In this case the controller is moved to the next state via the `bgp.completedwitherrors` transition that, like the `bgp.succeeded` transition, does not contain any default business rules. As with successful completion of conversion, the controller is automatically moved to the Delta Calculation state.

Delta Calculation State

The Delta Calculation state compares the converted file(s) to either the STEP database, or to the last loaded file of that type, and generates a set of STEPXML files containing only the changed data. This is done to increase performance of the actual imports so that unnecessary data is not processed.

The determination for the comparison method is set per importer using the **Delta calculation method** parameter described in the **5. Update Delete Status Attribute and Delta Calculation Method in Import Workflows** section of the **Quick Start Setup for Admins**.

Delta Calculation State Common Parameters

The Delta Calculation state includes one **Run background process action** business action by default, which runs the delta calculation service as a background process. All standards share a common delta calculation service and the parameters are described below.

The screenshot shows a dialog box titled "Edit Operation" with a close button (X) in the top right corner. The main content area is titled "Run background process action" and contains several input fields and a section for service parameters. The "QueueID" field is set to "queue_DeltaCalculate". The "Background Process Service" is a dropdown menu showing "[StepXMLDeltaCalculationService]". Below this is a section titled "Service parameters" with the following fields: "Delta calculation method" set to "context", "Clear delete attribute" with an unchecked checkbox, "Delete status attribute" set to "DeleteStatus (DeleteStatus)", "From file Attachment ID" set to "convert", and "Output file Attachment ID" set to "delta". At the bottom right of the dialog are "Save" and "Cancel" buttons.

- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_DeltaCalculate.'

Note: This parameter is specified for the action itself, not the particular service within the action, though the outcome is the same as each action runs only a single service.

- **Delta calculation method:** Required parameter that determines how the delta file is generated (via file or context method).

For additional information and considerations for setting this parameter, see the **5. Update Delete Status Attribute and Delta Calculation Method in Import Workflows** section of the **Quick Start Setup for Admins**.

- **Clear delete attribute:** Optional parameter used to clear the existing delete flag. This parameter removes / clears the value “true” from an attribute (configured in the 'Delete status attribute' parameter) for an deleted object, when the deleted object (Object marked for deletion) is resurrected through the later imports.

For example, consider a PIES item A which already has delete attribute value set as "true" in the system. With the 'Clear delete attribute' parameter selected, importing of PIES item A via the new PIES file erases the delete flag existing in the attribute marked for deletion for the PIES item A.

- **Delete status attribute:** Optional parameter to select an attribute to store the indication for deletion as a result of the import. If not set, deletions will not be processed by the importer. Defining an attribute in the 'Delete status attribute' parameter will set the flag to “true” if the object exists in STEP, but not in the imported file.

Note: AutoCare ACES files using complete replacement concepts require a delete status attribute to be set.

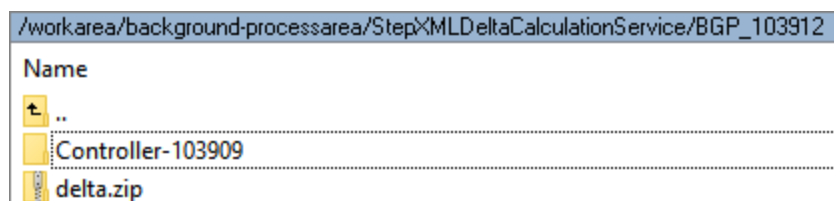
For additional information and considerations for setting this parameter, see the **5. Update Delete Status Attribute and Delta Calculation Method in Import Workflows** section of the **Quick Start Setup for Admins**.

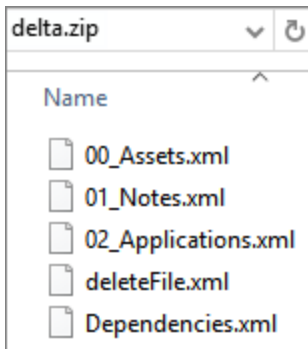
- **From file Attachment ID:** Required parameter that must be populated with the file name that should be used for the delta calculation service to act on. This defaults to 'convert' as that would be the correct file name if no additional states and/or processing were added to the default workflow. If an additional workflow state or service had been added that generated some other output file that the delta calculation should act on, the parameter should be updated accordingly.

Delta Calculation State Function Details

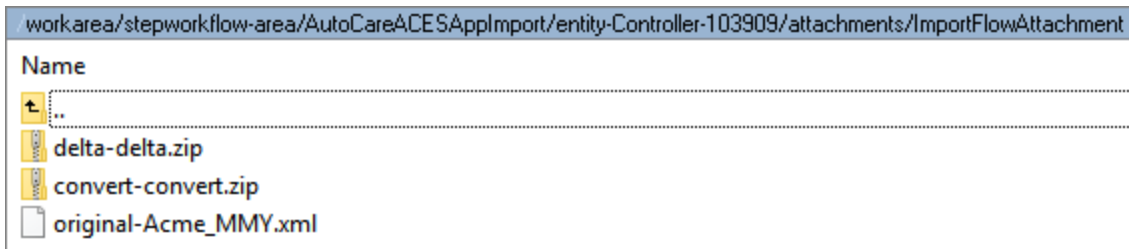
As the delta calculation service runs a background process (BGP), a corresponding BGP folder is created on the application server at /workarea/background-processarea/StepXMLDeltaCalculationService. The BGP folder contains a sub-folder whose name matches the STEP ID of the controller entity that is moving through the workflow and contains the converted file that the delta calculation was performed on. The delta calculation service also writes an additional file, which is the output of the delta calculation process, e.g., a zipped file with the name specified in the 'Output file attachment ID' parameter. This zipped file contains sub-files in STEPXML format that are submitted to the import service for further processing. The sub-files contained in any delta file differ based on the file type.

For example, an AutoCare ACES delta calculation result is shown below:





In addition, the delta file is prefixed with the value from the 'Output file attachment ID' parameter in the BGP service ('delta' in our example) and deposited on the application server at /workarea/stepworkflow-area/[Standard][Format]Import/entity-[ControllerID]/attachments/ImportFlowAttachment. For example:



Note: If modifications need to be made to the delta file(s) before further processing is completed, it is the workflow attachment file (e.g., delta-delta.zip in the above example) that must be manipulated as this is the file that is actually carried through the workflow.

Setup Objects and Delta Calculations

The delta calculation service takes setup objects into account, where setup objects (LOVs, attributes, reference types, etc.) that have not changed will be filtered away and not be included in the delta calculation file. Only changed setup objects will be included in the delta calculation file. This is done by comparing what is in the import file to what's in the database whether 'context' or 'file' delta calculation method is used.

Note: Importers will not handle deleted setup objects, they will instead be reported as a warning in the delta calculation background process Execution Report. For example, if there are LOV values in the database but are no longer in the import file, then there will be a warning in the delta calculation background process Execution Report listing the LOV and values that are missing in the import file. In this case, the user will need to check the delta calculation report and manually carry out the deletion of the LOV values.

Delta Calculation State Results

- **Failed:** If the import file fails delta calculation for some reason, the controller object in the workflow is sent to the Error state via the bgp.failed transition. This transition includes one default business rule which is used to populate the overall status of the controller entity (the overall status is displayed to end users in the Web UI). The controller will remain in the Error state until acted on by a user.

- **Succeeded:** If delta calculation completes successfully without any errors, the controller automatically moves to the Ready for Import state via the `bgp.succeeded` transition. This transition does not include any default business rules as the controller is moved automatically so there is no need to display an overall status to the end user at this time.
- **Completed with errors:** It is possible for a file to complete delta calculation, but with errors. In this case the controller is moved to the next state via the `bgp.completedwitherrors` transition that, like the `bgp.succeeded` transition, does not contain any default business rules. As with successful completion of delta calculation, the controller is automatically moved to the Ready for Import state.

Error State

Files that have failed validation, conversion, or delta calculation will have the controller end up in the Error state, and it will remain there until acted on by a user. It is not possible to do further processing of the file after it has errored, though it can be subsequently reloaded if needed. As described in the **Automotive Quick Start Guide**, users are able to view the errors generated in validation, conversion, or delta calculation by clicking the appropriate background process links to view the corresponding execution reports. They can also choose to discard the file, which will transition the file to the Discard File state, which contains a business rule that automatically removes the controller from the workflow so that it is no longer displayed in the workflow tasks in the Web UI.

Ready for Import State

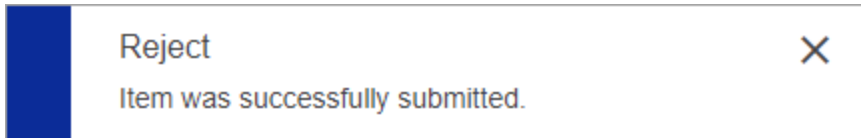
When a file has successfully completed validation, conversion, and delta calculation, the controller object is available in the Ready for Import state, which is the first state in which user interaction is required for successful files. If customer-specific reporting has been added to the implementation, it is likely that reports can be viewed in this state. Whether or not reporting has been added, the Ready for Import state is where the user must decide whether to import or reject the file. If the user rejects the file, it will be moved to the Rejected state via the Reject transition, which contains a single business rule to set the overall status of the controller entity. If the user chooses to import the file, it is moved to the Import state.

Only controller objects with files that have failed a previous process will end up in the Error state. It is also possible for files to have warnings and/or errors and still continue successfully through the process so it is important for users to view the data provided in the background processes in the Web UI to determine whether or not a file should be imported.

Rejected State

When a file has been rejected by a user, the controller is moved to the Rejected state. This indicates that the file has successfully completed all pre-import processing, but has been deemed unacceptable for import by a user, usually due to data issues. A rejected file cannot be imported unless it is reloaded, so the only option to act on a controller in this state is to discard it.

Clicking the **Reject** button will cause the Reject dialog (shown below) to display at the top of the screen. A user must click the 'X' in the top right of the dialog to close it.



Import State

The Import state carries out the actual import of the files generated from the delta calculation service. As the generated files are in STEPXML format, the standard STEP Importer is the engine used behind the scenes to carry out the import.

Import State Common Parameters

The Import state includes one business action (**Run background process action**) by default, which runs the import service as a background process. All standards share a common import service (though default actions and conditions applied to the imports differ) and the parameters are described below.

The screenshot shows a dialog box titled "Edit Operation" with a close button (X) in the top right corner. Inside the dialog, there is a dropdown menu at the top left showing "Run background process action". Below this, there are several input fields and sections:

- QueueID:** A text input field containing "queue_Import".
- Background Process Service:** A dropdown menu showing "[ImportingFlowImportService]".
- Service parameters:** A section containing:
 - File Attachment ID:** A text input field containing "delta".
 - Import action:** A large empty text area with a green plus icon (+) at the bottom left.
 - Import condition:** A text area containing "Validate Application on import (ValidateApplication)" with a green plus icon (+) at the bottom left and a close button (X) at the top right.

At the bottom right of the dialog, there are two buttons: "Save" and "Cancel".

- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run, which defaults to 'queue_Import.' Note that this parameter is specified for the action itself, not the particular service within the action, though the outcome is the same as each action runs only a single service.
- **File Attachment ID:** Required parameter that must be populated with the name of the file to be imported. This defaults to 'delta' as that would be the correct file name if no additional states and/or processing were added to the default workflow. If an additional workflow state or service had been added that generated some other output file that the import should act on, the parameter should be updated accordingly.

Note: This value must match the value provided for the 'Output file Attachment ID' parameter within the Delta Calculation state (if no other states are added to the default workflow). If it does not match, then the process will fail at the Import state.

- **Import action:** Optional parameter allowing administrators to select one or more business actions to be run as part of the import. By default, there are no actions included but customers may add any that they wish.

- **Import condition:** Optional parameter allowing administrators to select one of more business conditions to be run as part of the import. Importers that manage applications / linkages have one default condition applied (Validate application on import). This condition prevents applications from being imported if they do not contain a vehicle / assembly and part type in STEP. Additional information on the condition can be found in the **Automotive Business Rule Plugins** section.

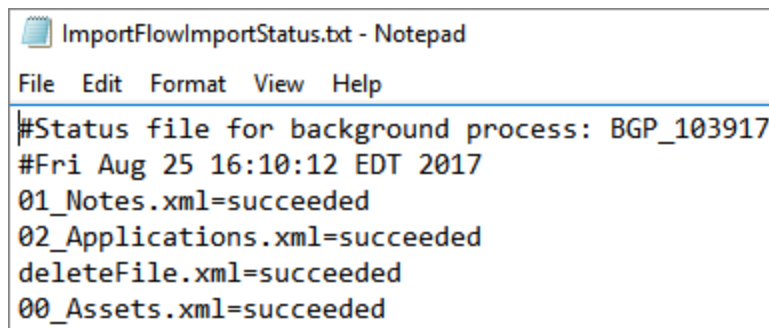
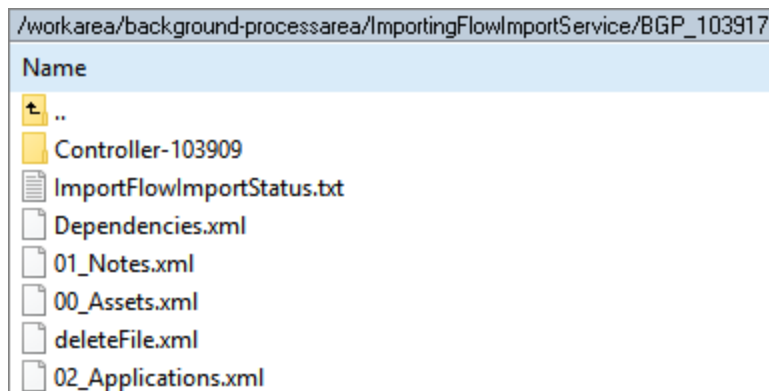
Note: Any business rules selected within the import service function the same as business rules applied in a standard STEP import, meaning that they act on the objects being imported (rather than the controller entity). If conditions are used, they must render true or the data will be excluded from the import.

Important: Automotive importers do not automatically approve objects that are created by importer. So if Objects need to be approved, then a Business Action using the following JavaScript function must be created: `'node.approve();'`. Additionally, that Business Action must be added to the 'Import action' parameter.

Import State Function Details

As the import service runs a background process (BGP), a corresponding BGP folder is created on the application server at `/workarea/background-processarea/ImportingFlowImportService`. The BGP folder contains a sub-folder whose name matches the STEP ID of the controller entity that is moving through the workflow and contains the delta calculation file that is being imported. The import service also writes a copy of each file for which import was attempted (the number of content of files will differ based on the import) and a status file which indicates success or completion of each individual import file.

For example, an AutoCare ACES import result is shown below:



Import State Results

The import can succeed, fail, or 'complete with errors' and the controller will automatically follow appropriate bgp. transitions as applicable. All transitions lead to the same Import Completed state and none contain business rules by default, though each customer can add additional handling as they see fit.

Note: Information on errors and/or failures will be visible in Web UI via the background process information provided for the import.

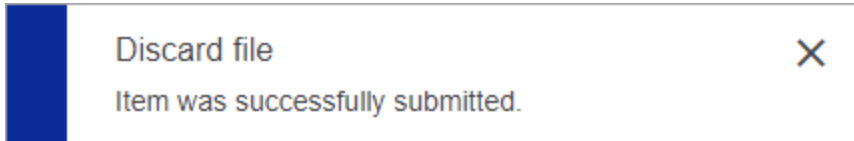
Import Completed State

When import has completed, the controller object is automatically passed to the Import Completed state where it will remain until a user takes action on it. The only option for action is to discard the file, which passes the controller object to the Discard File state, effectively removing it from the workflow.

Discard File State

When a controller enters the Discard File state, it is automatically removed from the workflow via a business rule running on entry to the state. This means that information about the file and processing of it is no longer visible in Web UI workflow screens, though the controller entity itself still exists in STEP. Customers should consider if they would like additional "clean up" actions to occur at this point, such as deleting the entity in STEP and/or deleting the generated files or background processes from the application server.

Clicking the **Discard file** button will cause the Discard file dialog (shown below) to display at the top of the screen. A user must click the 'X' in the top right of the dialog to close it.



Properties can be set in the sharedconfig.properties file on the application server to manage auto-deletion of the background processes. This is done using the format: **AutoDeleteBackgroundProcesses.AgeInHours.[Service] = [hours]**. Note that deletion of the background processes will remove the end user's ability to view warnings and errors for the processes in the Web UI. It is recommended to set up auto-deletion, but care should be taken to set the timing to an appropriate value for the end user processes.

For example, to set the AutoCare validation processes to delete after 48 hours, specify:

```
AutoDeleteBackgroundProcesses.AgeInHours.AutoCareValidationService = 48
```

Important: When using the 'file' delta calculation method, discarded files will not be used for delta calculations. The last loaded file must be retained for it to be used in delta calculation.

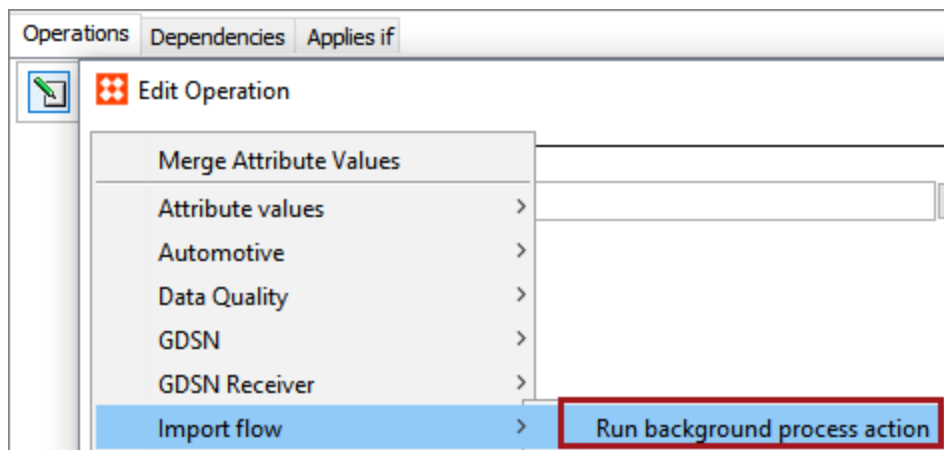
Modifying Import Framework

It is intended that the default workflows be modified to support customer-specific needs. Both the Extension API and the business rule plugins provided with the Automotive solution will assist in this.

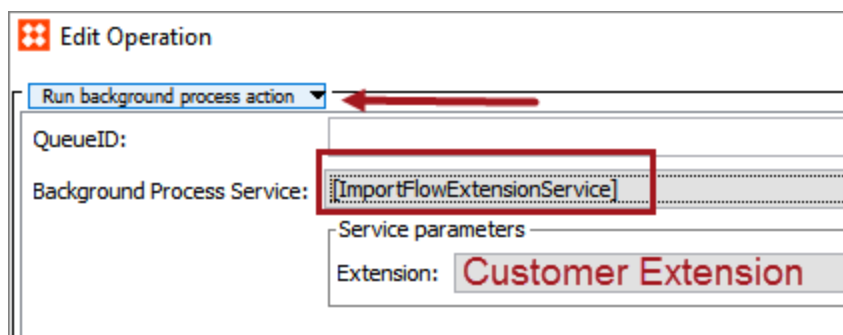
There are many out of the box business rules that can be added for advanced functionality in the importers. For more information, see the **Automotive Business Rule Plugins** section.

The Extension API provides access to all of the files generated as part of the import workflows (e.g., via conversion and delta calculation, as well as the original file submitted for import). Furthermore, it provides the ability to add new plugins to the existing framework that can then be run as part of any existing background process service, such as the Import Flow Extension Service created expressly for this purpose.

To clarify, customers may add new states to any workflow, or new rules to any existing state, where they access and manipulate the files generated by the import framework. In either case, if significant processing is needed, this should be done as a background process using the 'Run background process action.'



Within that action, the **ImportFlowExtensionService** should be selected. Any extensions that have been created by the customer via the Extension API will be available for selection in the Extension parameter dropdown.



Additional information can be found in the Extension API Javadoc. Systems with one or more Automotive licenses will have access to the **com.stibo.importflow.domain.extension** package which includes interfaces for extending the import framework.

Displaying Import Modifications in Web UI

If additional states are added to any workflow, it may be useful to also reflect them in the Web UI. If the file will spend more than a few minutes processing within the state, it may be useful to display the state to end users so the file does not appear to be missing.

Additionally, if user interaction is required for the file to move to the next state, it must be accessible to end users so that they can take action on it. On the other hand, new states that are automated only (e.g. the system will transition the task into the next state without any user intervention) do not necessarily need to be displayed to the end user.

To manage additional import workflow states in the Web UI, two actions need to be taken:

1. Add the state to the Homepage Status Selector for the workflow. An introduction to status selectors in the Automotive solution is provided in the **Automotive Quick Start Guide** within the **Solution Enablement** section of **STEP Online Help**. Detailed information for configuring status selectors is available in the **Status Selector Homepage Widget** topic within **STEP Online Help**.
2. Update the associated workflow screen to display details about the state. Each importer has a corresponding workflow screen in Web UI called '[Standard][Format]WorkflowScreen,' (e.g., AutoCareACESApplicationWorkflowScreen). Select that screen in the Web UI designer and navigate to the Data Provider to add the states, e.g., select the **go to component** link for the Node Editor child component > double click on the **Refreshable Node List** child component > click the **Edit** button for the Data Provider > select the appropriate workflow in the Workflow parameter > add the additional states using the dropdown and **Add** button for the States parameter > **Save**).

ID Structures in Importers

Prior to importing part and/or application data, it is imperative to know the ID structure for the relevant standard. If part and/or application data is imported into STEP without using the proper ID structures, then errors will occur.

The part ID and application ID structures for each of the standards (i.e., AutoCare, NAPA, and TecDoc) are described below. In the examples below, the import file information is displayed (when applicable) above a screenshot of the workbench ID, Name, and Object Type fields.

Prerequisites

The ID structures described within this section are dependent upon the completion of the setup as described in the **2. Run Easy Setup of Import Flow Process** section of the **Automotive Quick Start Guide**.

AutoCare Part ID and Application ID Structures

- **PIES ID** = AC_PIESItem_[BrandAAIAID]_[PartNumber]

```
<PartNumber>034-VC21499</PartNumber>
<BrandAAIAID>GWWQ</BrandAAIAID>
```

ID	AC_PIESItem_GWWQ_034-VC21499
Name	034-VC21499
Object Type	PIES Item

- **ACES ID** = AC_ACESApp_[hash function]

```
<Part BrandAAIAID="GWWQ">034-VC21499</Part>
```

> ID	AC_ACESApp_36fe8bb39ddf3ae9b93cf09b4237c
> Name	034-VC21499
> Object Type	ACES Application

Note: Submitting the above application will search for the above part, with the ID as shown. If not found, an error will be reported and the application will not be created.

NAPA Part ID and Application ID Structures

- **Part ID** = NAPA_Product_[ProductLine][PartNumber]

Name	>	>	Value	>
> ID			NAPA_Product_AA4AA000001	
> Name			AA000001	
> Object Type			NAPA Product	

Note: A NAPA parts importer does not exist within STEP because the NAPA standard does not offer a standard parts import format, but parts must still be created with this ID structure.

- **Application ID = NAPA_App_[hash function]**

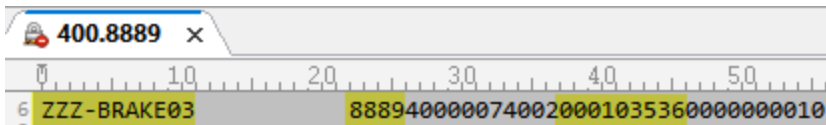
AA4 AA000001

> ID	NAPA_App_bc30f8333daa6f57dbbb658cc8b567
> Name	AA4AA000001
> Object Type	NAPA Application

Note: Submitting the above application will search for the above part, with the ID as shown. If not found, an error will be reported and the application will not be created.

TecDoc Part ID and Application ID Structures

- **Part ID = [Supplier ID]-[hash function]**



ID	8889-3684ee1ea8ea5d07c527ac1776a8fe73
Name	ZZZ-BRAKE03
Object Type	DS_Supplier Article

- **Application ID = TD_L_[hash function]**

ID	TD_L_a2f9742628a61dcf5537cbcba4efb547
Name	TD_L_000103536
Object Type	DS_Linkage

Note: The part and application data are in the same file, eliminating the 'product not found' error.

Adding Reporting Extensions to Imports

It is expected that customers may want to add some additional states to the default workflow, and some of these states may include a process to generate a report and/or other file type for end users to view.

For example, an Impact Report state could be added to generate a report listing new data that would be created as a result of the import, and existing data that would be marked for deletion. The state could be added to the workflow screen (as described in the **Displaying Import Modifications in Web UI** section), and a downloadable report could be made available to the user, as shown below:

Import Details						
Process	Started Time	Duration	Started By	Status	Background	Report
Validation	2017-08-29 09:28:15	6 secs	STEPSYS	Validation completed	succeeded	
Conversion	2017-08-29 09:28:21	7 secs	STEPSYS	Conversion completed	succeeded	
Delta Calculation	2017-08-29 09:28:28	3 secs	STEPSYS	Delta calculation completed	succeeded	
Impact Report	2017-08-29 09:28:31	11 secs	STEPSYS	Generation completed	succeeded	Download file
Import	2017-08-29 09:40:29	1 min 53 secs	STEPSYS	Import completed	succeeded	

Creation of the service to generate the report requires use of the Import Flow Extension Service (ImportFlowExtensionService), but there is some built-in functionality to be aware of to enable users to access the report file in the Web UI. For more information about the Import Flow Extension Service, see the **Modifying Import Framework** topic.

Files can be stored to the importflow using the **addFile** (String attachmentID, File file) method in **ImportFlowExtensionContext**. Later it can be accessed from another process with the **getFile** (String attachmentID) method, using the same attachmentID with which it was stored. For the file to display in the Report column in the Web UI, the attachmentID has to be equal to the ID of the workflow state in which the file should appear. See the **ImportFlowExtensionContext** interface in the Javadoc for additional information.

However, this requires the Importflow State Report header to be displayed in the associated workflow screen, as described in the **Displaying the Import Details Report Column** topic within the **Automotive Quick Start Guide** found within the **Solution Enablement** section of **STEP Online Help**.

Validation Error Handling

Each import has some basic format validations applied, and each implementation must determine how these should be handled.

Information on each of the available automotive standard specific importers is addressed in the following sections:

- AutoCare Importers
- TecDoc Reference Data Importer

The handling of validation errors can be configured by enabling / disabling the 'Continue on Error' parameter on the Validation state of each import workflow. By default, the 'Continue on Error' parameter is disabled.

For example, when reference data includes multiple files and a data point included in one is not present in a corresponding related file (e.g., a PCdb Codemaster file includes a position that is omitted from the Position file in the same PCdb), an error will be written to the execution report of the validation process. However, as reference data is typically managed by an outside source (e.g. AutoCare, TecDoc, or NAPA), it may be desired to import the data regardless of these types of errors. If so, the 'Continue on Error' parameter should be checked on the validation service. In this case the validations issues will still be written to the execution report, but all valid data will be converted and made available for the import process. If unchecked, validation errors will cause the import process to stop and will need to be corrected before the file can complete validation and continue on in the import process.

The sections below detail how this setting changes user actions when importing data.

For the examples in the sections below, consider when an importer has a strict file name validation rule (e.g., NAPA Vehicle and NAPA Translation Importers), should a bad file name stop the data from being imported? If so, then the parameter should remain disabled. If not, then the parameter should be enabled.

Continue on Error Disabled

In the example below, the NAPA Vehicle Import Controller Screen displays the import file name, if you look closely you can see that spaces were used instead of underscores. Because the 'Continue on Error' parameter was disabled at the time this import file was processed, the Import Details table displays the status as 'Validation failed,' the Background Process Link displays as 'failed,' and the Overall Status of the import displays as 'Error: Validation failed.'

NAPA Vehicle Imports

Process	File Name	Overall Status
Controller-109469	ValidVehicles Rev2000 01_01.zip	Error: Validation failed

Import Details

Process	Started T	Duration	Started By	Status	Background Process Link
Validation	2017-10-26 11:03:41	5 secs	STEPSYS	Validation failed	failed

The validation errors caused the import process to stop, and the errors must be corrected before the file can complete validation and continue on in the import process. Clicking the 'failed' Background Process Link will display the Background Process Details, where the detailed error message displays (as shown below).

ID	Type	Text
<input type="checkbox"/> 10	Error	The file name 'ValidVehicles Rev2000 01_01.zip' does not match required format 'ValidVehicles_RevYYYY_MM_DD.zip'.

Once the error is addressed the user will need to begin the file import process, as before.

Continue on Error Enabled

With the 'Continue on Error' parameter enabled, the import will continue through the workflow, but the Validation process status will display as 'Validation failed' within the Import Details section of a Web UI Import Controller Screen, and the Background Process Link will display as 'succeeded.'

In the example below, the NAPA Vehicle Import Controller Screen displays the import file name. Notice that spaces are used instead of underscores in the beginning of the file name. Because the 'Continue on Error' parameter was enabled at the time this import file was processed the Import Details table displays the status as 'Validation failed,' but the Background Process Link displays 'succeeded' and the import file was able to enter the next state of the workflow, despite the file name not meeting the validation criteria.

NAPA Vehicle Imports

Process	File Name	Overall Status
Controller-109568	ValidVehicles Rev2000 01_01.zip	Done creating delta file

Import Details

Process	Started T	Duration	Started By	Status	Background Process Link
Validation	2017-10-26 11:03:41	5 secs	STEPSYS	Validation failed	succeeded

Clicking the 'succeeded' Background Process Link will display the Background Process Details, where the detailed error message is displayed (as shown below).

ID	Type	Text
<input type="checkbox"/> 10	Error	The file name 'ValidVehicles Rev2000 01_01.zip' does not match required format 'ValidVehicles_RevYYYY_MM_DD.zip'.
<input type="checkbox"/> 20	Info	Validation finished with errors: 1 (Thu Oct 26 11:03:46 EDT 2017)

AutoCare Importers

PMDM for Automotive offers an out-of-the-box solution for importing the file types necessary to the AutoCare standard. Each of the AutoCare file types has their own importer, and more information for each can be found within their respective sections below.

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

1. Qdb (Qualifier Database)
2. Brand Table
3. PCdb (Product Classification Database)
4. PAdb (Product Attribute Database)
5. VCdb (Vehicle Configuration Database)
6. PIES (Product Information Exchange Standard)
7. ACES (Aftermarket Catalog Exchange Standard)

For information on the supported versions, see the **Supported Versions and Formats** topic.

Note: Details on each file type's data rules per the AutoCare Association can be found at www.autocare.org.

AutoCare ACES Importer

The intention of the Automotive AutoCare ACES Importer is to provide an out-of-the-box solution for importing data included within a supported ACES file format. Because only supported versions will successfully upload, before attempting to upload an AutoCare ACES file, confirm the file version being uploaded is listed within the **Supported Versions and Formats** topic.

This section includes information on:

- Using AutoCare ACES Importer
- Configuring AutoCare ACES Importer

Prerequisites

It is expected that anyone using and/or configuring the AutoCare ACES Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Using AutoCare ACES Importer

ACES file can be imported into STEP by uploading it to either a configured hotfolder on an application server, or through a File Loading Widget on a Web UI Homepage.

The following topics provide information on using the AutoCare ACES Importer.

- Importing ACES Files via Web UI
- Sample AutoCare ACES File Structure
- ACES Import Validation Rules

Prerequisites

It is expected that anyone using and/or configuring the ACES Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For more information about uploading files to an application server, see the **File Loading** section of the **Automotive Quick Start Guide** found within the **Solution Enablement** section of **STEP Online Help**.

For more information about using Workbench to monitor and process imports, see the **Workflows** section of **STEP Online Help**.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Importing ACES Files via Web UI

The intention of the AutoCare ACES Importer Web UI setup included within this topic is to provide an out-of-the-box solution for importing data included within a supported AutoCare ACES file format. Because only supported versions will successfully upload, before attempting to upload an ACES file, confirm the file version being uploaded is listed within the **Supported Versions and Formats** topic.

An ACES file includes information on application data, i.e., information about fitment of a particular part to various vehicles, engines, etc.

Prerequisites

Within this section users will be directed to view details for a Controller Entity within Web UI using a Node Details screen configured for Entities. Before this can be available to users, an Entity Details screen must be created.

All reference data files (Brand Table, Qdb, PCdb, VCdb, and PAdb) must be imported before PIES or ACES files are imported.

Important: It is critical to understand that for systems running on Automotive 9.1 version and older, the PIES file must be imported first before importing the ACES file. Failing to comply, the application records associated with the missing PIES Items in the system will be skipped from being imported. For systems running Automotive 9.1 MP1 and newer, users will have an option to import ACES files without first importing PIES files, provided that the 'Create PIES items' parameter is checked in the Conversion state of the ACES Import workflow. For more information, see the **Conversion State** topic within the **Default Workflow States and Functions** section.

To easily start and monitor an ACES Import, recommended practice is to use a Web UI Import Controller screen specific to the AutoCare ACES Importer. When the Easy Setup actions for the AutoCare standard are completed, the AutoCare Import Controller screen is automatically created and configured for use. Additional information can be found in the **Configuring AutoCare ACES Importer** topic.

It is expected that anyone using and/or configuring the AutoCare ACES Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

If the Easy Setup actions for the AutoCare Component model have been completed, then the functionality explained within this topic should be available. Otherwise, configuration is necessary. For more information, see the **AutoCare ACES Importer** topic.

Import Process Overview

Once a valid automotive data file is uploaded to a hotfolder on the application server (optionally using a File Loading Widget), the file is picked up from the hotfolder by an IIEP, and the IIEP creates an Entity in STEP that represents the file. This Entity object is called the 'controller' and contains basic data about the file and the file's status in the workflow. Web UI users are able to monitor the import status using a Status Selector Widget and an Import Controller Screen. Once an import file reaches the 'Ready for Import' state, then users can start the import by clicking on the 'Start import' button within the Control Panel screen. At that time, the BGP service (that runs as part of the Import state) allows for the configured business rules to act on the objects being imported.

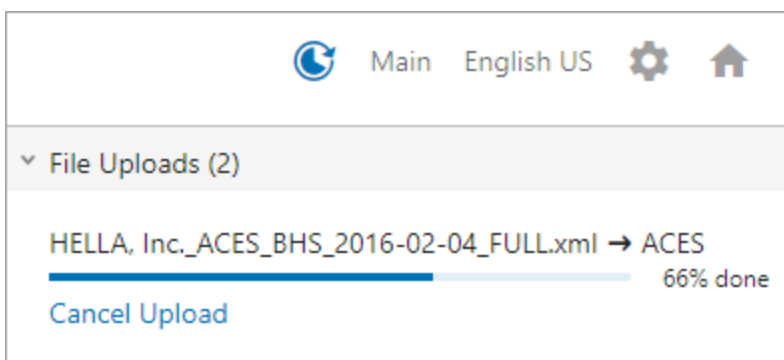
Using the AutoCare ACES Importer in Web UI

1. Access the AutoCare Web UI Homepage.
2. Upload a valid ACES file to the application server hotfolder (root/upload/hotfolders/AutoCareACESApplicationInputFolder), or use the 'ACES' drop zone of the 'SUPPLIER DATA IMPORTS' File Loading Widget.

For more information, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

3. Once the upload has started, users can view the progress of the upload using the Background Process Notification Component.

In the example below, the BGP Notification Panel is expanded and the file, 'HELLA, Inc._ACES_BHS_2016-02-04_FULL.xml' is 66% processed.



For more information on using the BGP Notification Component and side panel, see the **Background Process Notification Component** topic within the **Main Properties Overview** section of **STEP Online Help**.

Once the file has been uploaded and picked up by the IIEP, the Controller is initiated into the workflow associated with the importer (AutoCare ACES Import). From there, the AutoCare ACES Import workflow takes over processing of the file via a series of states using business rules and background processes to carry out the processing of the file. Each import has an associated workflow and all proceed through the same states by default. However, it is intended that customers will expand on the existing states and actions to add their own validations, approve objects, reporting, and additional processing as needed.

Important: It is critical to understand that it is only the Controller Entity that is in the workflow - the objects being acted on (created / updated / deleted) via information supplied in the import file are not in the workflow. Therefore, running standard business actions acting on current object will impact the Controller Entity only, not the objects in the input file. In order for the Business Action to act on the objects being imported, the Business Action must be added to the 'Import action' parameter on the 'Import' state. For more information, see the **Import State** topic within the **Default Workflow States and Functions** section.

4. If an Entity Details screen has been configured, then when the IIEP creates the Controller Entity object, users can view the Controller and its important information by navigating within the Tree side panel to the Import Flow Root > AutoCare ACES. Otherwise, if a user clicks on an Entity in the Tree side panel, an error will display indicating the configuration is invalid.

Note: If an error message displays when selecting an Entity within a Web UI, then the Entity Details screen has not been configured.

In the example below, the Tree side panel is expanded, and the 'HELLA, Inc._ACES_BHS_2016-02-04_FULL.xml' Controller with ID 'Controller- 105212' is displayed below the 'AutoCare ACES' Entity. When the Controller is selected, many details can be viewed within the Node Details screen.

Note: In the screenshot below, the 'Enable Tag Conversion' parameter within the Attribute Value Group Component is enabled.

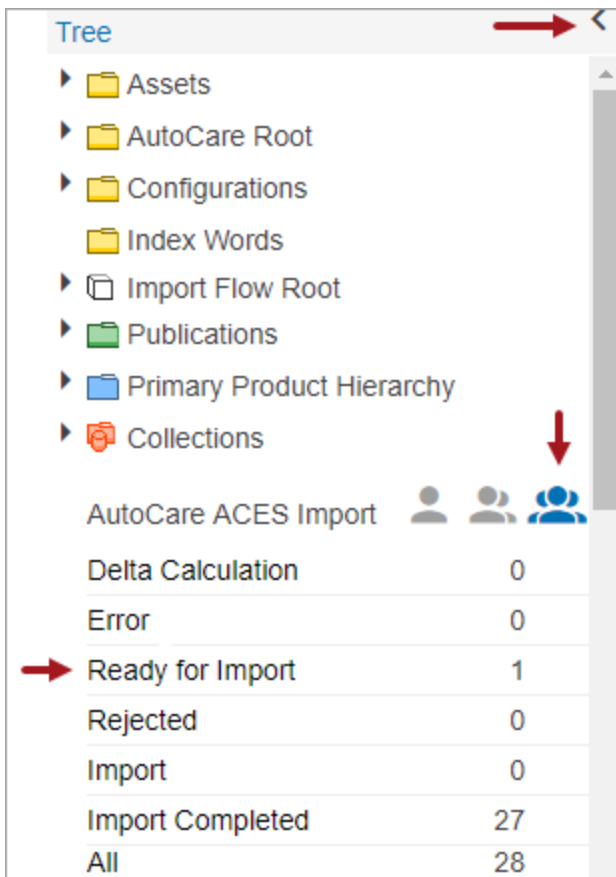
Node Details	
ID	Controller-105212
Name	HELLA, Inc._ACES_BHS_2016-02-04_FULL.xml
Object Type	Import Flow Controller Type
Revision	Date: 2019.01.30 Revision number: 0.1 Last Edited by: ACESIMPORT Time: 04:41:53
Automotive Import Flow State BGP	<<?xml version="1.0" encoding="UTF-8"?> <StringMap> <Entry Key="Validation" Value="BGP_105213"/> <Entry Key="Import" Value="BGP_105216"/> <Entry Key="Conversion" Value="BGP_105214"/> <Entry Key="DeltaCalculate" Value="BGP_105215"/> </StringMap>
Import Flow Endpoint ID	AutoCareACESApplicationInboundEndpoint
Import Flow File Type	ApplicationData
Import Flow Overall Status	Completed with errors: 29504
Import Flow State Status	<<?xml version="1.0" encoding="UTF-8"?> <StringMap> <Entry Key="Validation" Value="Validation completed"/> <Entry Key="Import" Value="Completed with errors: 29504"/> <Entry Key="Conversion" Value="Conversion completed with errors"/> <Entry Key="DeltaCalculate" Value="Delta calculation completed"/> </StringMap>
Import Flow Workflow ID	AutoCareACESAppImport

- **Name:** STEP Name of the Controller Entity. This is generated from the original name of the uploaded file.
- **Automotive Import Flow State BGP:** Attribute used to store the IDs of the background processes.

- **Import Flow State Status:** Attribute used to store and display the status of each process (rather than the overall status).
 - **Import Flow Overall Status:** Attribute used to store and display the global status of the file (rather than the process specific status).
5. As the file is uploaded, and the Controller Entity moves through the AutoCare ACES Import workflow, users can monitor the progress using the 'AUTOCARE ACES IMPORT' Status Selector Widget on the Web UI homepage and/or the left side panel.

In the example below, the Tree panel is expanded, and the **Triple user button** is selected so that all items assigned to any user are displayed. Notice that one file is in the 'Ready for Import' state, and twenty seven files in the 'Import Completed' state, for a total of twenty eight files.

For more information, see the **Moving Tasks Trough a Workflow in Web UI** topic within **STEP Online Help**.



6. Clicking on the **Ready for Import** state within the Status Selector homepage and/or the left side panel widget will navigate the user to the AutoCare ACES Import Controller Screen with only those Controllers in the 'Ready for Import' state displayed.
7. Clicking the Controller icon displays the Import Details below the Controller list.

In the example below, Controller 'HELLA, Inc._ACES_BHS_2016-02-04_FULL.xml' (ID 'Controller-105630') with an overall status of 'Done creating delta file' is selected and the Import Details are displayed below the Controller list.

AutoCare ACES Import Controller Screen

Process	File Name	Overall Status
Controller-105630	HELLA, Inc._ACES_BHS_2016-02-04_FULL.xml	Done creating delta file

1-1 of 1

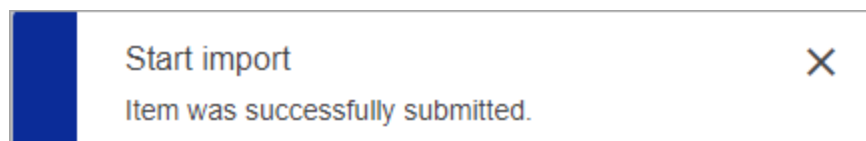
Import Details

Process	Started Time	Duration	Started By	Status	Background Process
Validation	2019-03-19 13:57:47	4 secs	STEPSYS	Validation completed	succeeded
Conversion	2019-03-19 13:57:51	9 secs	STEPSYS	Conversion completed with errors	completedwitherrors
Delta Calculation	2019-03-19 13:58:00	1 min 29 secs	STEPSYS	Delta calculation completed	succeeded

1-4 of 4

Start import Reject Discard file

- Optionally, before starting the import, clicking on the 'completedwitherrors' Background Process Link will display the Background Process Details screen where users can view or download the details of the validation process.
- Click the **Reject** button to move the Controller to the Rejected state. For more information, see the **Rejected State** topic within the **Import Framework** section.
- Click the **Start import** button, the Start import dialog (shown below) displays at the top of the screen, and the Controller is moved to the Import state of the workflow. For more information, see the **Import State** topic within the **Import Framework** section.



While importing an ACES file the import progress can be monitored using the 'AUTOCARE ACES IMPORT' Status Selector Widget on the Web UI homepage and/or the left side panel or the Import Controller Screen, Import process Status column.

In the example below, the AutoCare ACES Import Controller Screen displays the Overall Status and the Import process status as 'Importing files...'. Additionally, the Background Process Link column displays a link to the Background Process Details screen as 'running.'

AutoCare ACES Import Controller Screen

Process	File Name	Overall Status
Controller-105630	HELLA, Inc._ACES_BHS_2016-02-04_FULL.xml	Importing files... ←

Import Details

Process	Started Time	Duration	Started By	Status	Background Proc
Validation	2019-03-19 13:57:47	4 secs	STEPSYS	Validation completed	succeeded
Conversion	2019-03-19 13:57:51	9 secs	STEPSYS	Conversion completed with errors	completedwith
Delta Calculation	2019-03-19 13:58:00	1 min 29 secs	STEPSYS	Delta calculation completed	succeeded
Import	2019-03-19 15:13:23	26 secs	STEPSYS	Importing files... →	11 running

Start import Reject Discard file

- Optionally, click the 'running' Background Process Link (shown above) to view the details of the import process within the Background Process Details screen.

Once the import has completed, the Controller is automatically moved to the Import Completed state. For more information, see the **Import Completed State** topic within the **Import Framework** section.

For more information about ACES import validation rules and example error messages, see the **ACES Import Validation Rules** topic.

For more information on automotive workflow states, see the **Default Workflow States and Functions** topic.

Sample AutoCare ACES File Structure

The intention of this topic is to introduce the basic AutoCare ACES file structure and the way the objects are built in the system when the ACES file is imported into the system.

Currently, the Automotive solution supports import and export of three versions of ACES files. Namely, ACES 3.0, ACES 3.2 and ACES 4.0. All three versions are common in its structure with additional add-on features amongst the later versions. For better understanding, ACES 3.2 is considered to be an example throughout this topic.

The ACES file shall include only plain text delimited by 'valid' XML tags (as defined below). A CR/LF (ASCII 13,10 decimal) should separate lines and it is recommended that the text should use UTF-8 character encoding. However, ISO-8859-1 is also accepted.

File Naming

It is a good practice that the external file name of the ACES file shall include the company, a short catalog title, the delivery date (yyyy-mm-dd), and the word '**FULL**' or '**UPD**', or '**TEST**' in a sequence all separated by underlines and an "xml" extension. Below is an example of how an ACES file can be named.

```
ACMESupply_BrakeHardware_2001-01-18_FULL.xml
```

Zipped File Imports

When more than one file is imported simultaneously into the system or when the ACES file contains a million records so you would want to reduce the size of the file, users can use an archiving program (such as WinZip, or Windows XP 'Compressed (zipped) Folders') to compress the files into a single ZIP archive. Again, it is a good practice to name the archive as above but without the catalog title and with a '.zip' extension.

Note: When more than one ACES files are zipped, and the user drop the zip file into the hotfolder, those ACES files will not get processed simultaneously unless it is configured for the ACES import to be able to run in parallel (the default is that only one file will be processed at a time).

Delivering ACES Files

Sample File structure

Below is a sample ACES 3.2 file with two application records

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ACES version="3.2">
<Header>
  <Company>KRMA</Company>
  <SenderName>Kay Jensen</SenderName>
  <SenderPhone>6787978489</SenderPhone>
  <TransferDate>2018-04-21</TransferDate>
  <BrandAAIAID>DKGX</BrandAAIAID>
  <DocumentTitle>ACES FILE</DocumentTitle>
  <EffectiveDate>2018-04-21</EffectiveDate>
  <SubmissionType>FULL</SubmissionType >
  <MapperCompany>Maps R Us</MapperCompany>
  <MapperEmail>maps@maps.com</MapperEmail>
```

```

    <VcdbVersionDate>2018-04-20</VcdbVersionDate>
    <QdbVersionDate>2018-04-20</QdbVersionDate>
    <PcdbVersionDate>2018-04-20</PcdbVersionDate>
</Header>
  <App action="A" id="1">
    <BaseVehicle id="5289"/>
    <MfrBodyCode id="2765"/>
    <EngineBase id="1305"/>
    <Qty>1</Qty>
    <PartType id="5696"/>
    <MfrLabel>ChkEngSens</MfrLabel>
    <Part BrandAAIAID="DKGX">VC36009</Part>
  </App>
  <App action="A" id="2">
    <BaseVehicle id="5289"/>
    <BodyNumDoors id="5"/>
    <EngineBase id="1305"/>
    <Qty>2</Qty>
    <PartType id="5696"/>
    <MfrLabel>ChkEngSens</MfrLabel>
    <Part BrandAAIAID="DKGX">VC36112</Part>
  </App>
  <Footer>
    <RecordCount>2</RecordCount>
  </Footer>
</ACES>

```

The whole ACES file is divided into three required sections: Header, Body, and Footer.

Header

This constitutes the initial section of the file. This section includes generic file information such as company details, sender details, mapper details, version details of Vehicle, etc. Header information is present within the <Header> tags.

The order of the tags within the header is defined by the XML schema. In the Header section, the <Company>, <SenderName>, <SenderPhone>, <TransferDate>, <EffectiveDate>, <SubmissionType>, <VcdbVersionDate>, <QdbVersionDate>, <PcdbVersionDate> and <DocumentTitle> tags are required to be populated and can not be left empty.

Body

The Body section contains catalog data that should only include the information necessary to unambiguously select the correct part (PIES Item). Each application record will be surrounded with the <App> tags and can contain some or all of the following details.

- Application Record Number
- Base Vehicle ID
- Make
- Model

- Years
- Submodel
- Engine Base ID
- Part number
- Part Type (Numeric ID of the Part type)
- Position
- Part Quantity
- Note
- Action ('A' or 'D', 'A' indicates to add the application record and 'D' indicates to delete the application record).
- Asset ID
- Asset name
- All other attributes / vehicle options that exist for any application record create and populate a unique column.

Of the above listed details, the following information is required for each application:

- Base Vehicle ID
- Part Type
- Part Number
- Part Quantity

A sample ACES XML body structure and the details in how it is populated on the STEP system is displayed below

```
<App action="A" id="2">
  <BaseVehicle id="5289"/>
  <BodyNumDoors id="5"/>
  <EngineBase id="1305"/>
  <Qty>2</Qty>
  <PartType id="5696"/>
  <MfrLabel>ChkEngSens</MfrLabel>
  <Part BrandAAIAID="DKGX">VC36112</Part>
</App>
```

App Action value "A" adds a new ACES application (if it was the value 'D' instead of 'A' then it would notify the system to mark the application for deletion).

Note: The automotive importers do not delete any objects, it just marks objects as new, delete, or change through change flags.

ID value '2' signifies the sequential order of parsing for the applications.

The value in the <part> tag is extracted to identify the parent for the application item. The application will be created / modified under the 'VC36112' PIES Item. <BodyNumDoors id="5"/>, </MfrLabel>, and <Qty>, populates the respective attribute values as displayed in the screenshot below.

Tree

- Product Overrides
- Packaging
- AutoCare Root
 - ACES Application Notes
 - PIES Interchange Products
 - PIES Products
 - 034MOTORSPORT
 - BCVC
 - Behr Hella Service
 - Behr Thermot-Tronik
 - Exide
 - Hella
 - Pierburg
 - Wako
 - Engine
 - Sensors
 - Check Engine Light Sensor
 - JK21499
 - JK36004
 - JK36009
 - JK36112
 - VC21499
 - VC36004
 - VC36009
 - VC36009
 - VC36112
 - VC36112

PIES Item → VC36112
ACES → VC36112

VC36112 rev.0.3 - Product

Product | References | Referenced By | Images & Documents | Commercial | Tables

Description

Name	Value
ID	AC_ACESApp_470c52439c0cf4c60f496a9cc6221
Name	VC36112
Object Type	ACES Application
Revision	0.3 Last edited by ACESIMPORT on Fri Mar 08 07:51:38 EST 2019
Approved	Never Been Approved
Translation	Not Translated
Path	Primary Product Hierarchy/AutoCare Root/PIES Products/Wako/En
Changed Object	abc true
Delete Status	abc
New Object	abc
Replacement Context	abc ACES-ACESBOND-DKGX-5289-5696-VC36112

AutoCare ACES Attributes

Name	Value
ACES Aspiration	
ACES Body Num Doors	4
Front Spring Type	
Mfr Label	abc ChkEngSens
Quantity	123 2

PAdb Attributes

<BaseVehicle id="5289"/> and <EngineBase id="1305"/> creates a reference to the objects within the Vehicle Configuration (VCdb) classification and <PartType id="5696"/> creates a reference to the object within the Parts Categorization (PCdb) classification in the STEP system as displayed in the screenshot below.

Tree

- Product Overrides
- Packaging
- AutoCare Root
 - ACES Application Notes
 - PIES Interchange Products
 - PIES Products
 - 034MOTORSPORT
 - BCVC
 - Behr Hella Service
 - Behr Thermot-Tronik
 - Exide
 - Hella
 - Pierburg
 - Wako
 - Engine
 - Sensors
 - Check Engine Light Sensor
 - JK21499
 - JK36004
 - JK36009
 - JK36112
 - VC21499
 - VC36004
 - VC36009
 - VC36009
 - VC36009
 - VC36112
 - VC36112

VC36112 rev.0.3 - References

Reference Type	Target
AutoCare ACES Attributes	
Reference Type	Target
> ACES Application To Engine Base	2.4L L4, 2354CC, -CID
> ACES Application To Transmission...	
Ungrouped Classification Links	
Reference Type	Target
> Product To Part Terminology	Check Engine Light Sensor
Ungrouped Product References	
Reference Type	Target
> ACES Application To Application N...	
Ungrouped Image & Document References	
Reference Type	Target
> ACES Application To Installation I...	
> ACES Application To Owners Manual	
> ACES Application To Primary Prod...	
> ACES Application To Product Image	
Ungrouped Classification References	
Reference Type	Target
> ACES Application To Base Vehicle	1986 Ford F-150
> ACES Application To Qualifier	

Footer

The Footer section indicates the end of the data and provides a count of the applications that is being transferred in the file.

```
<Footer><RecordCount>2</RecordCount></Footer>
```

ACES Import Validation Rules

When importing an AutoCare ACES (Aftermarket Catalog Exchange Standard) file, many STEP validation rules are performed. The table below describes each of these validation rules, where the validation check occurs, what happens when a validation fails, and an example of a failed validation message (when applicable). However, it does not detail the file type's data rules per the AutoCare Association. This information can be found at www.autocare.org.

For information on the supported versions, see the **Supported Versions and Formats** topic within this guide.

Important: For best results, AutoCare imports should be executed in the order specified within the **AutoCare Import Validation Rules** topic.

Accepted File Extension: .xml and .zip

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
ACES 3.2 XSD validation	Validation state	Import fails Validation state.	cvc-pattern-valid: Value 'somethingWrongBrand' is not facet-valid with respect to pattern '[B-Z-[EIOU]][B-Z-[EIO]][B-Z-[OU]][A-Z]' for type 'brandType.'
Base vehicle existence	Import state via Business Action	Record skipped.	An application (id=13) for part: VC36004 has no assembly target.
File name matches configuration	Conversion state	BGP ends in an error triggering the import to enter the Error state.	Conversion stopped due to an error: Current import handling configuration for brand DKGX requires Supplier information that failed to be obtained from the filename.
Import mode Full and action = D	Conversion state	Record is skipped and a warning will display in the BGP report.	Application 16 will be skipped as the import mode FULL only processes the applications with A (Add) action.
Part existence	Import state via Business Condition	Record skipped.	AC_PIESItem_DKGX_BADNumber does not exist for application (id=15).
Part terminology existence	Import state via Business Condition	Record skipped.	An application (id=14) for part: VC36004 has no parttype.
Qualifier existence	Import state	Record is imported but a standard STEP reference target not found error will display.	Target 'AC_Qualifier_12946111111111111111' of reference not found.
Text in number conditions	Conversion state	BGP ends in an error, triggering the import to enter the Error state, and an exception to display within the conversion report.	

Note: The ACES importer does not validate that the VCdb, PCdb, and/or Qdb versions within STEP are the same versions within the ACES file.

Configuring AutoCare ACES Importer

The following topics provide the configuration steps necessary to allow users to be able to drag and drop ACES files onto a configured File Loading Widget, and monitor the progress of the import file using a configured Status Selector Homepage Widget and an Import Controller Screen.

- Configuring an IIEP for ACES Imports
- Configuring a File Loading Widget for ACES Imports

Prerequisites

It is expected that anyone configuring the AutoCare ACES Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, see the **Designer Access** topic within the **Web UI Getting Started** section of **STEP Online Help**.

Within this section users will be directed to view details for a Controller Entity within Web UI using a Node Details screen configured for Entities. Before this can be available to users, an Entity Details screen must be created.

Anyone configuring the AutoCare ACES Importer is expected to be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Configuring an IIEP for AutoCare ACES Imports

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

An Inbound Integration Endpoint (IIEP) can be configured in workbench to help automate the process of importing ACES data into STEP. Once an IIEP is configured for AutoCare ACES imports, ACES files can be imported after they are uploaded either to a configured hotfolder on an application server, or to a File Loading Widget on a Web UI Homepage. For more information, see the **AutoCare ACES Importer** topic.

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

Prerequisites

It is expected that anyone configuring an IIEP for use with an AutoCare ACES Import is familiar with the configuration and other processing of standard inbound integration endpoints. For more information, see the **Inbound Integration Endpoints** topic within the **Data Exchange** section of **STEP Online Help**.

Configuration Steps

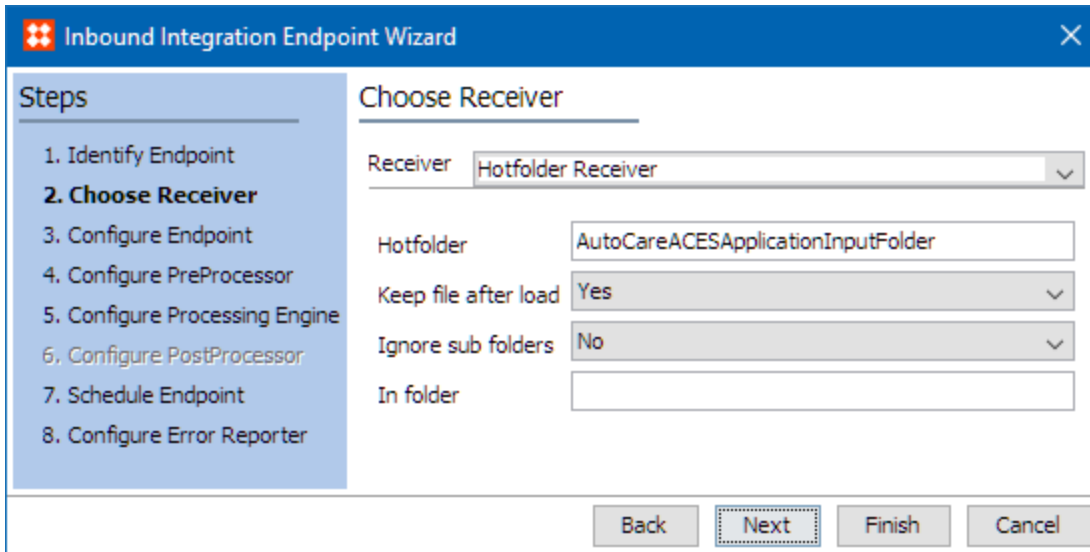
1. In workbench, go to System Setup, select and right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.
2. Once the Inbound Integration Endpoint Wizard displays, populate each parameter with values that best identify the IIEP. By default, all parameters display blank, and the following fields are mandatory: Endpoint ID, Endpoint Name, and User.

In the example below, an 'ACES Import' user was created prior to configuring this IIEP. This is recommended to more easily track when this IIEP is responsible for changes to data.

For more information about the parameters, see the **IIEP - Identify Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the Choose Receiver parameters will display. By default, the parameters are populated as recommended and shown below, except the Hotfolder parameter. This mandatory parameter must be populated with a hotfolder name before the Next button will enable.

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



For more information about the parameters, see the **IIEP - Choose Receiver** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the Configure Endpoint parameters will display. By default, the parameters are pre-populated with the recommended values as shown below, except for the following:
 - Processing Engine
 - Maximum number of old processes
 - Maximum age of old processes

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
- 3. Configure Endpoint**
4. Configure PreProcessor
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Endpoint

Processing

Processing Engine: Import Flow Processor

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: 1000

Maximum number of old processes: 1000

Maximum age of old processes: 1y

Number of messages per background process: 1

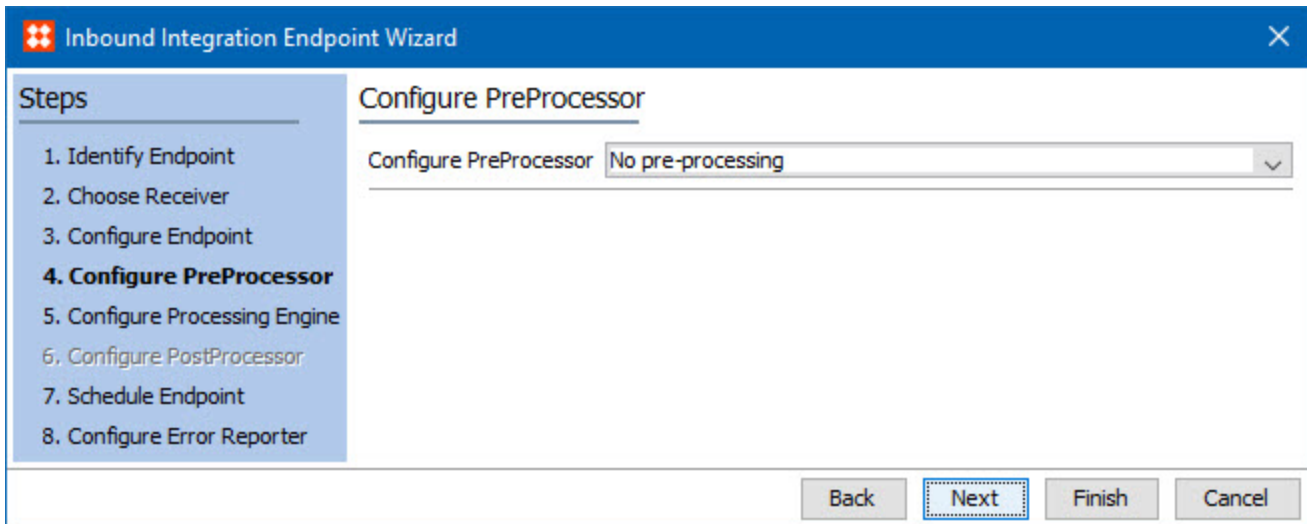
Buttons: Back, Next, Finish, Cancel

Important: The Import Flow Processor is only responsible for picking up files from the configured hotfolder, creating a Controller Entity object, and updating the Entity with the Import Flow State BGP and Status. The Import Flow Processor works with the Background Process Service for each workflow state to handle the import of a file. The IIEP / Import Flow Processor is NOT responsible for actually importing the content of the file, the BGP in each state does that processing. If a new Processing Engine is created through the Extension API, then it **cannot** be used with the automotive import framework.

- Click the dropdown option for the 'Processing Engine' parameter and select the **Import Flow Processor** option.
- Optionally, update the values for the 'Maximum number of old processes' and 'Maximum age of old processes' parameters to those shown above.

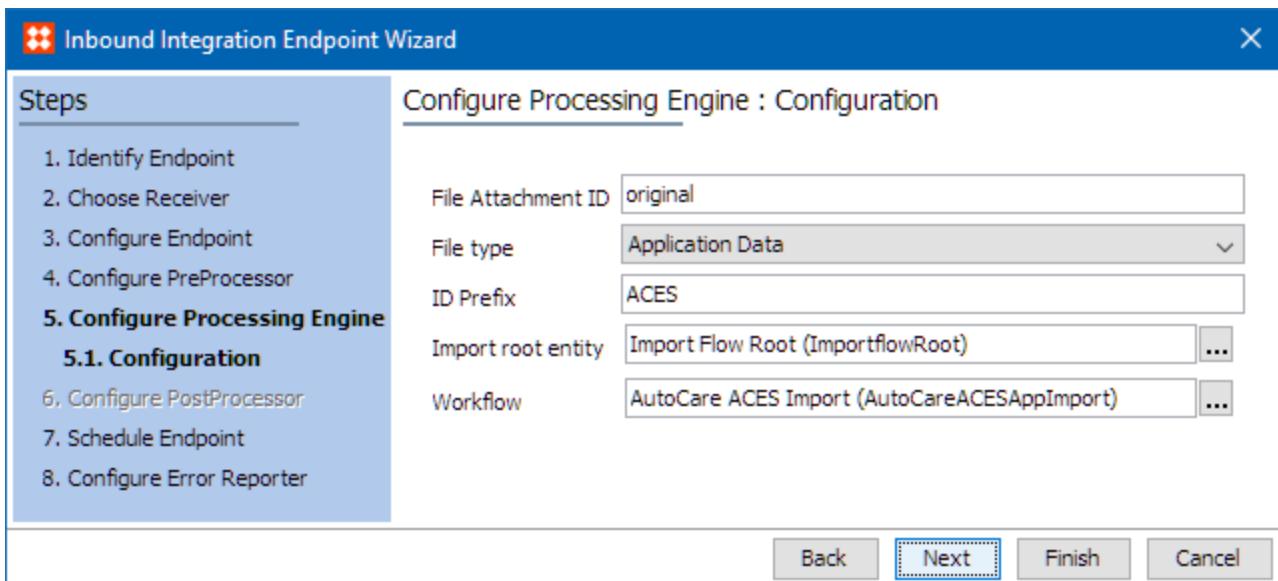
For more information about the parameters, see the **IIEP - Configure Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

5. Click the **Next** button, and the Configure PreProcessor parameter will display. By default, the parameter is populated as recommended and shown below.



For more information about the parameter, see the **IIEP - Configure PreProcessor** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the **Configure Processing Engine : Configuration** parameters for the Import Flow Processor will display. By default, only the File Attachment ID parameter is populated as shown below.



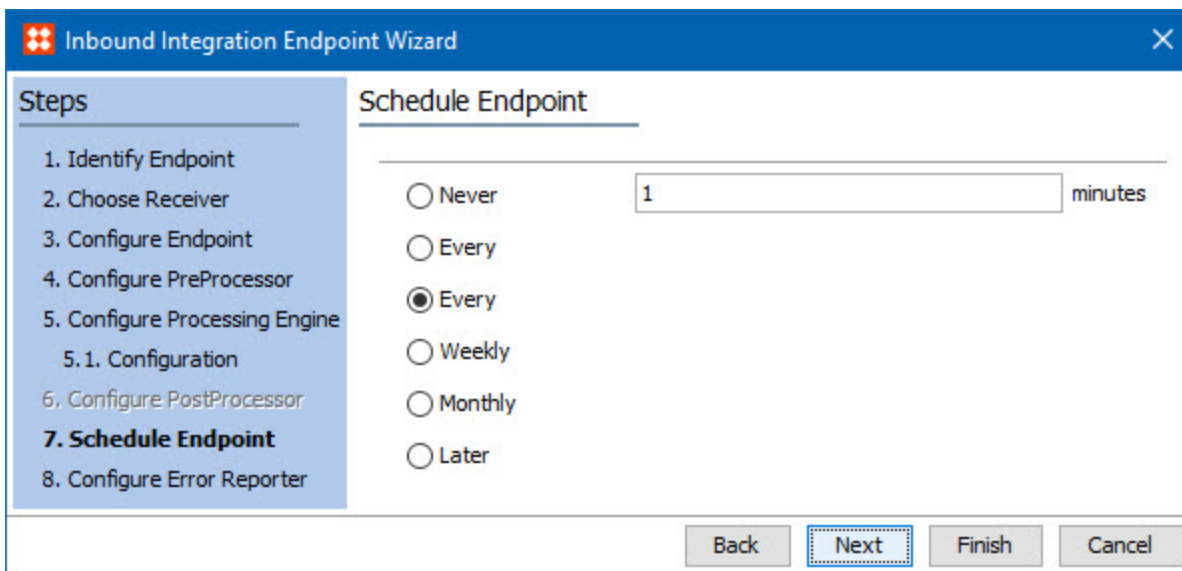
The prior Configure Endpoint step determines the options available for the Configure Processing Engine step.

- Click the 'File type' parameter, and select the **Application Data** option.
- Within the 'ID Prefix' parameter, enter a prefix value to easily identify import entities created by this IIEP.
- Click the ellipsis button (...) for the 'Import root entity' parameter, and select the 'Import Flow Root' node (or a desired root entity). This is the location where the import entities created by this IIEP will be stored.

- Click the ellipsis button (...) for the 'Workflow' parameter, and select the **AutoCare ACES Import** workflow or a desired workflow.

Note: It is possible to use a workflow that is not created by Easy Setup actions to handle an import file in a way that better fits an organization's needs. However, along with creating the workflow and selecting it within the Workflow parameter shown above, all the states for that workflow must be created along the processing steps of the file (i.e., Validation, Conversion, Import). Use of the Extension API is required to write the processing steps.

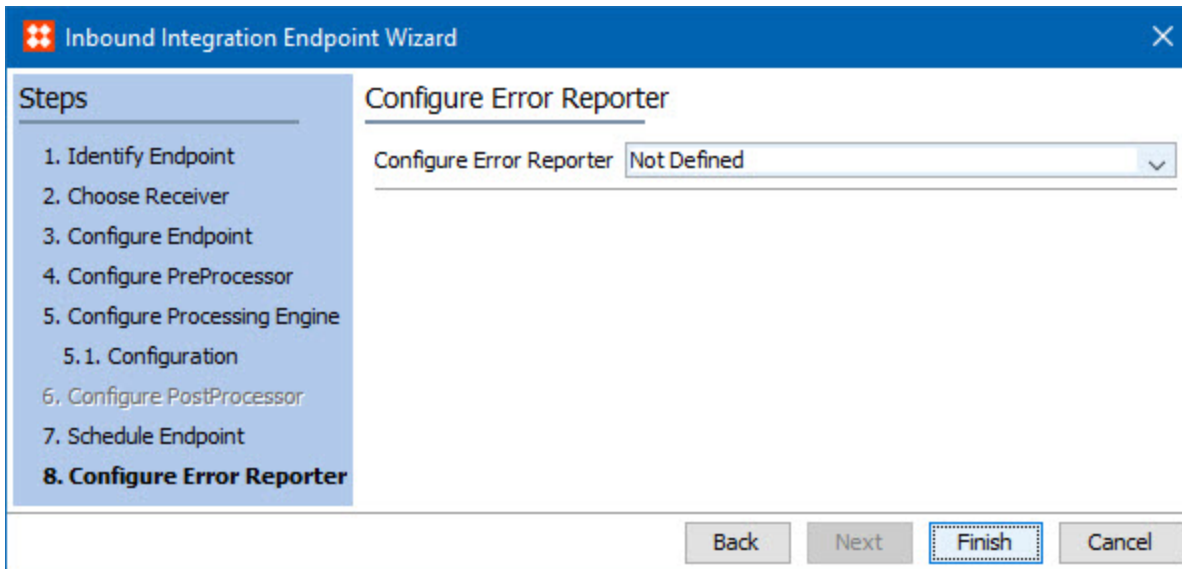
7. Click the **Next** button, and the Schedule Endpoint parameters will display. By default, 'Never' is selected. Optionally, update the values to those shown below.



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.

For more information about the parameters, see the **IIEP - Schedule Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

8. Click the **Next** button, and the Configure Error Reporter parameter will display. By default, the parameter is populated as recommended and shown below.



For more information about the parameter, see the **IIEP - Configure Error Reporter** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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

Important: An endpoint must be enabled before it can start processing data. For more information, see the **Running an Inbound Integration Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

If users need to access the IIEP via a Web UI, then the IIEP must be configured within a File Loading Widget. For more information, see the **Configuring a File Loading Widget for AutoCare ACES Imports** topic.

Configuring a File Loading Widget for AutoCare ACES Imports

When configured, Web UI users can import ACES files into STEP using a File Loading Widget. Users can also monitor the progress of imports using a Status Selector Homepage Widget and a Node Details component.

Prerequisites

Before starting to configure the Web UI portion of this solution, an IIEP for an AutoCare ACES Importer must be configured within workbench. For more information, see the **Configuring an IIEP for ACES Imports** topic.

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

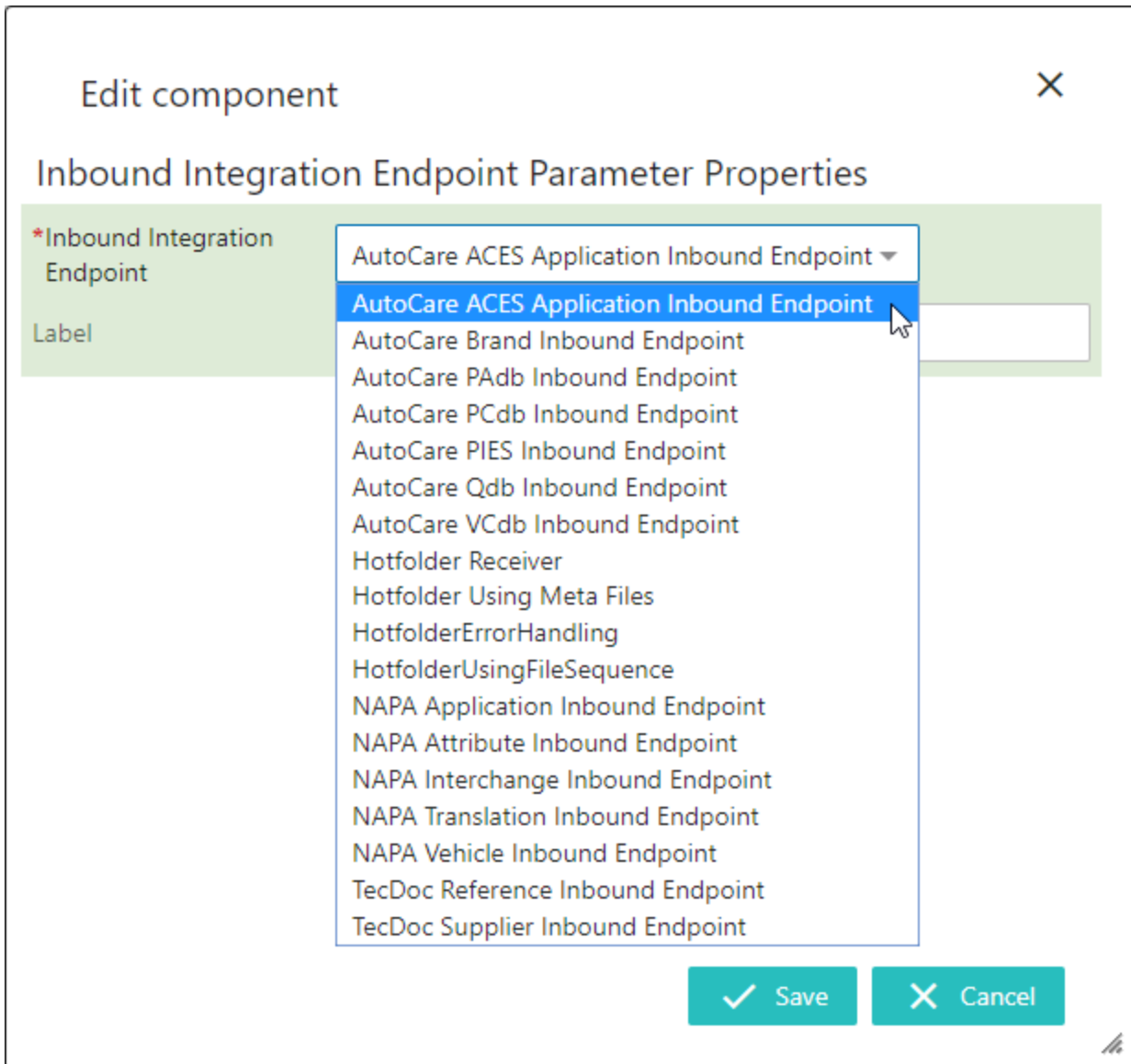
Configuration

Each screenshot example within this section provides recommended values for the parameters and AutoCare ACES Importer.

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

Note: If Easy Setup actions for the AutoCare solution have been completed as described in the **3. Run Easy Setup of Standards** topic of **Automotive Quick Start Guide**, then the 'SUPPLIER DATA IMPORTS' File Loading Widget will automatically be added to the AutoCare Web UI Homepage as shown in the examples below. Otherwise, the steps below can be used to complete configuration.

1. In the designer, select an existing File Loading Widget to be used, or add a new File Loading Widget to the Homepage Widget Grid component. For more information, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.
2. Go to the Inbound Integration Endpoint Parameters field, click the **Add** button, and the Inbound Integration Endpoint Parameter Properties dialog will display.
3. Click the dropdown for the Inbound Integration Endpoint parameter, and select **AutoCare ACES Application Inbound Endpoint** (the IIEP created for AutoCare ACES Application imports).

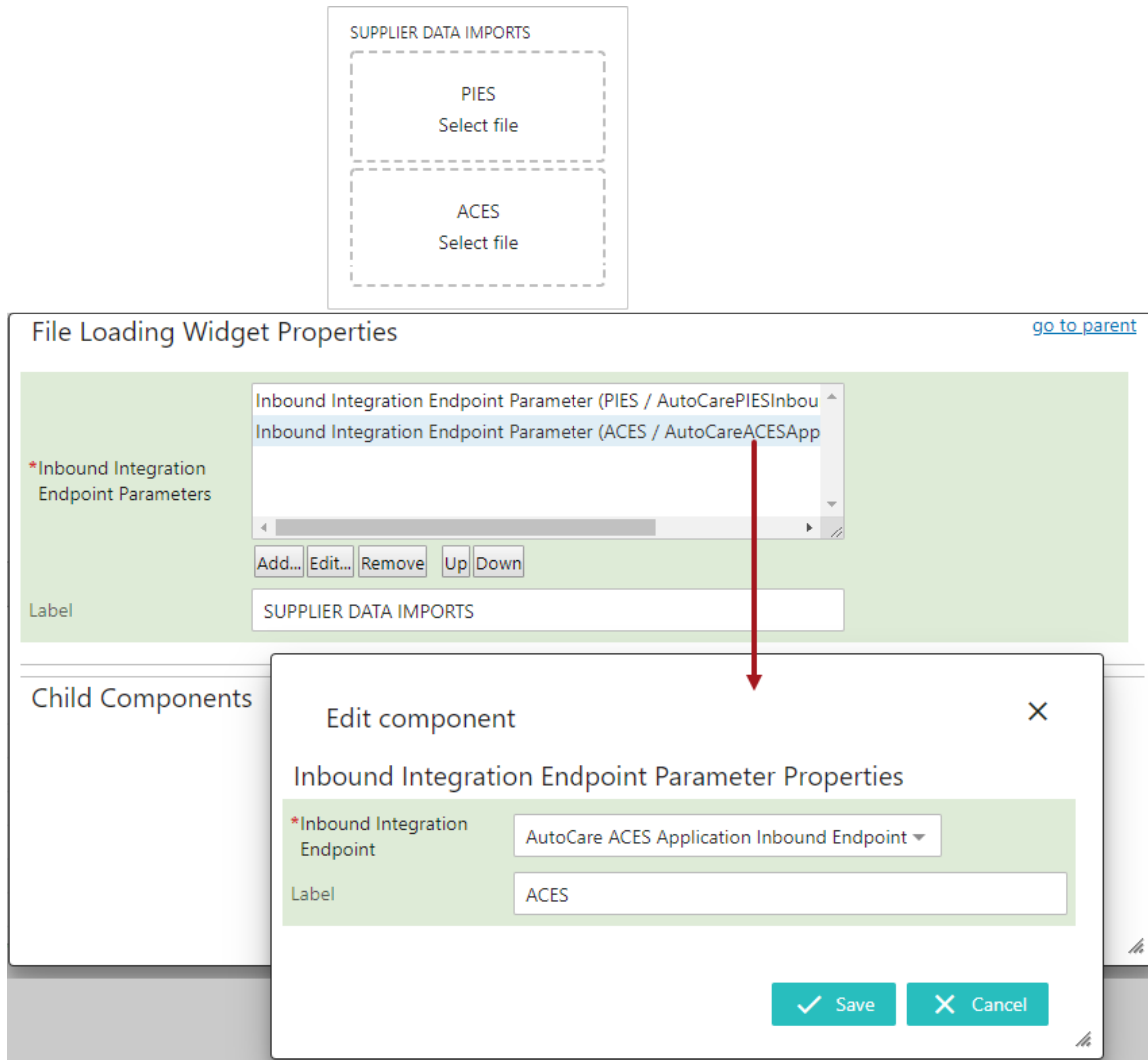


Note: If the desired IIEP does not display in the dropdown, then it can be created using the steps described in the **Configuring an IIEP for ACES Imports** topic.

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

In the example below:

- A File Loading Widget labeled as 'SUPPLIER DATA IMPORTS' is displayed above its configurations.
- The File Loading Widget and its configurations are shown with the default configurations provided automatically when Easy Setup actions for the AutoCare component are completed.
- An IIEP for AutoCare PIES imports is added within the same File Loading Widget as the IIEP for the AutoCare ACES imports.



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

AutoCare PIES Importer

The intention of the Automotive AutoCare PIES Importer is to provide an out-of-the-box solution for importing data included within a supported PIES file format. Because only supported versions will successfully upload, before attempting to upload an AutoCare PIES file, confirm the file version being uploaded is listed within the **Supported Versions and Formats** topic.

This section includes information on:

- Using AutoCare PIES Importer
- Configuring AutoCare PIES Importer

Prerequisites

It is expected that anyone using and/or configuring the AutoCare PIES Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Using AutoCare PIES Importer

PIES file can be imported into STEP by uploading it to either a configured hotfolder on an application server, or through a File Loading Widget on a Web UI Homepage.

The following topics provide information on using the AutoCare PIES Importer.

- Importing PIES Files via Web UI
- Sample AutoCare PIES File Structure
- PIES Import Validation Rules

Prerequisites

It is expected that anyone using and/or configuring the PIES Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For more information about uploading files to an application server, see the **File Loading** section of the **Automotive Quick Start Guide** found within the **Solution Enablement** section of **STEP Online Help**.

For more information about using Workbench to monitor and process imports, see the **Workflows** section of **STEP Online Help**.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Importing PIES Files via Web UI

The intention of the AutoCare PIES Importer Web UI setup included within this topic is to provide an out-of-the-box solution for importing data included within a supported AutoCare PIES file format. Because only supported versions will successfully upload, before attempting to upload a PIES file, confirm the file version being uploaded is listed within the **Supported Versions and Formats** topic.

Prerequisites

Within this section users will be directed to view details for a Controller Entity within Web UI using a Node Details screen configured for Entities. Before this can be available to users, an Entity Details screen must be created.

All reference data files (Brand Table, Qdb, PCdb, VCdb, and PAdb) must be imported before PIES files are imported.

To easily start and monitor a PIES Import, recommended practice is to use a Web UI Import Controller screen specific to the AutoCare PIES Importer. When the Easy Setup actions for the AutoCare standard are completed, the AutoCare Import Controller screen is automatically created and configured for use. Additional information can be found in the **Configuring AutoCare PIES Importer** topic.

It is expected that anyone using and/or configuring the AutoCare PIES Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

If the Easy Setup actions for the AutoCare Component model have been completed, then the functionality explained within this topic should be available. Otherwise, configuration is necessary. For more information, see the **AutoCare PIES Importer** topic.

Import Process Overview

Once a valid automotive data file is uploaded to a hotfolder on the application server (optionally using a File Loading Widget), the file is picked up from the hotfolder by an IIEP, and the IIEP creates an Entity in STEP that represents the file. This Entity object is called the 'controller' and contains basic data about the file and the file's status in the workflow. Web UI users are able to monitor the import status using a Status Selector Widget and an Import Controller Screen. Once an import file reaches the 'Ready for Import' state, then users can start the import by clicking on the 'Start import' button within the Control Panel screen. At that time, the BGP service (that runs as part of the Import state) allows for the configured business rules to act on the objects being imported.

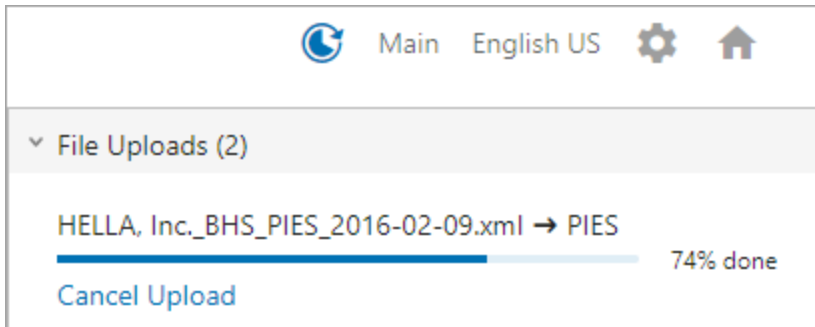
Using the AutoCare PIES Importer in Web UI

1. Access the AutoCare Web UI Homepage.
2. Upload a valid PIES file to the application server hotfolder (root/upload/hotfolders/AutoCarePIESInputFolder), or use the 'PIES' drop zone of the 'SUPPLIER DATA IMPORTS' File Loading Widget.

For more information, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

3. Once the upload has started, users can view the progress of the upload using the Background Process Notification Component.

In the example below, the BGP Notification Panel is expanded and the file, 'HELLA, Inc._BHS_PIES_2016-02-09.xml' is 74% processed.



For more information on using the BGP Notification Component and side panel, see the **Background Process Notification Component** topic within the **Main Properties Overview** section of **STEP Online Help**.

Once the file has been uploaded and picked up by the IIEP, the Controller is initiated into the workflow associated with the importer (AutoCare PIES Import). From there, the AutoCare PIES Import workflow takes over processing of the file via a series of states using business rules and background processes to carry out the processing of the file. Each import has an associated workflow and all proceed through the same states by default. However, it is intended that customers will expand on the existing states and actions to add their own validations, approve objects, reporting, and additional processing as needed.

Important: It is critical to understand that it is only the Controller Entity that is in the workflow - the objects being acted on (created / updated / deleted) via information supplied in the import file are not in the workflow. Therefore, running standard business actions acting on current object will impact the Controller Entity only, not the objects in the input file. In order for the Business Action to act on the objects being imported, the Business Action must be added to the 'Import action' parameter on the 'Import' state. For more information, see the **Import State** topic within the **Default Workflow States and Functions** section.

4. If an Entity Details screen has been configured, then when the IIEP creates the Controller Entity object, users can view the Controller and its important information by navigating within the Tree side panel to the Import Flow Root > AutoCare PIES. Otherwise, if a user clicks on an Entity in the Tree side panel, an error will display indicating the configuration is invalid.

Note: If an error message displays when selecting an Entity within a Web UI, then the Entity Details screen has not been configured.

In the example below, the Tree side panel is expanded, and the 'HELLA, Inc._BHS_PIES_2016-02-09.xml' Controller with ID 'Controller- 105202' is displayed below the 'AutoCare PIES' Entity. When the Controller is selected, many details can be viewed within the Node Details screen.

Note: In the screenshot below, the 'Enable Tag Conversion' parameter within the Attribute Value Group Component is enabled.

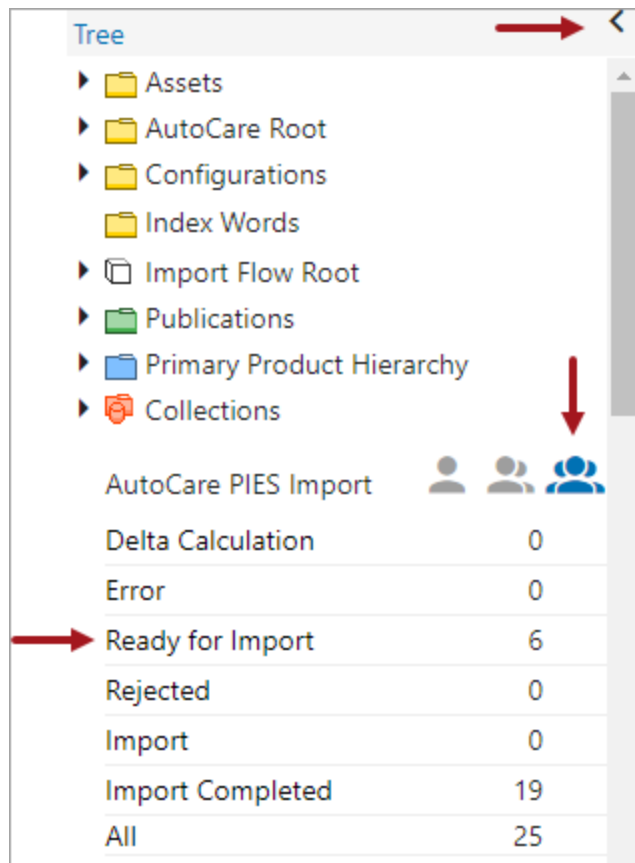
Tree		Node Details	
<ul style="list-style-type: none"> ▶ AutoCare Root ▶ Primary Product Hierarchy ▶ TecDoc Supplier Price Lists ▼ Import Flow Root <ul style="list-style-type: none"> ▶ AutoCare ACES ▶ AutoCare Brand ▶ AutoCare PAdb ▶ AutoCare PCdb ▼ AutoCare PIES <ul style="list-style-type: none"> ▶ CNXW ▶ DORMAN ▶ Exide ▼ HELLA, Inc. <ul style="list-style-type: none"> HELLA, Inc._BHS_PIES_2016-... HELLA, Inc._BHS_PIES_2016-... HELLA, Inc._BHS_PIES_2016-... ▶ PIES ▶ PIES2 ▶ THD ▶ Unknown ▶ AutoCare Qdb ▶ AutoCare VCdb ▶ NAPA Application ▶ NAPA Attribute ▶ NAPA Translation ▶ NAPA Vehicle ▶ TecDoc Reference ▶ TecDoc Supplier 		<p>Name: HELLA, Inc._BHS_PIES_2016-02-09.xml</p> <p>ID: Controller-105202</p> <p>Object Type: Import Flow Controller Type</p> <p>Revision: Date: 2019.01.29 Revision number: 0.1 Last Edited by: PIESIMPORT Time: 10:28:38</p> <p>Automotive Import Flow State BGP: <pre><?xml version="1.0" encoding="UTF-8"?> <StringMap> <Entry Key="Validation" Value="BGP_105203"/> <Entry Key="Import" Value="BGP_105206"/> <Entry Key="Conversion" Value="BGP_105204"/> <Entry Key="DeltaCalculate" Value="BGP_105205"/> </StringMap></pre> </p> <p>Import Flow Endpoint ID: AutoCarePIESInboundEndpoint</p> <p>Import Flow File Type: PIESData</p> <p>Import Flow Overall Status: Import completed</p> <p>Import Flow State Status: <pre><?xml version="1.0" encoding="UTF-8"?> <StringMap> <Entry Key="Validation" Value="Validation completed"/> <Entry Key="Import" Value="Import completed"/> <Entry Key="Conversion" Value="Conversion completed"/> <Entry Key="DeltaCalculate" Value="Delta calculation compl"/> </StringMap></pre> </p> <p>Import Flow Workflow ID: AutoCarePIESImport</p>	

- **Name:** STEP Name of the Controller Entity. This is generated from the original name of the uploaded file.
- **Automotive Import Flow State BGP:** Attribute used to store the IDs of the background processes.
- **Import Flow State Status:** Attribute used to store and display the status of each process (rather than the overall status).
- **Import Flow Overall Status:** Attribute used to store and display the global status of the file (rather than the process specific status).

5. As the file is uploaded, and the Controller Entity moves through the AutoCare PIES Import workflow, users can monitor the progress using the 'AUTOCARE PIES IMPORT' Status Selector Widget on the Web UI homepage and/or the left side panel.

In the example below, the Tree panel is expanded, and the **Triple user button** is selected so that all items assigned to any user are displayed. Notice that six file is in the 'Ready for Import' state, and nineteen files in the 'Import Completed' state, for a total of twenty five files.

For more information, see the **Moving Tasks Trough a Workflow in Web UI** topic within **STEP Online Help**.



6. Clicking on the **Ready for Import** state within the Status Selector homepage and/or the left side panel widget will navigate the user to the AutoCare PIES Import Controller Screen with only those Controllers in the 'Ready for Import' state displayed.
7. Clicking the Controller icon displays the Import Details below the Controller list.

In the example below, Controller 'HELLA, Inc._BHS_PIES_2016-02-09.xml' (ID 'Controller-106301') with an overall status of 'Done creating delta file' is selected and the Import Details are displayed below the Controller list.

AutoCare PIES Imports

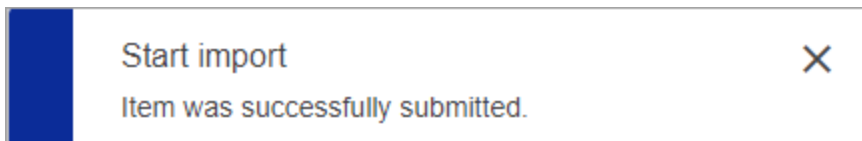
Process	File Name	Overall Status
Controller-106301	HELLA, Inc._BHS_PIES_2016-02-09.xml	Done creating delta file

1-1 of 1

Import Details

Process	Started Time	Duration	Started By	Status	Background Pri
Validation	2019-04-22 05:46:06	5 secs	PIESIMPORT	Validation completed	succeeded
Conversion	2019-04-22 05:46:11	5 secs	PIESIMPORT	Conversion completed	succeeded
Delta Calculation	2019-04-22 05:46:16	5 secs	PIESIMPORT	Delta calculation completed	succeeded
Import					

8. Optionally, before starting the import, clicking on the 'completedwitherrors' Background Process Link (only displayed when there is any error) will display the Background Process Details screen where users can view or download the details of the validation process.
9. Click the **Reject** button to move the Controller to the Rejected state. For more information, see the **Rejected State** topic within the **Import Framework** section.
10. Click the **Start import** button, the Start import dialog (shown below) displays at the top of the screen, and the Controller is moved to the Import state of the workflow. For more information, see the **Import State** topic within the **Import Framework** section.



While importing a PIES file the import progress can be monitored using the 'AUTOCARE PIES IMPORT' Status Selector Widget on the Web UI homepage and/or the left side panel or the Import Controller Screen, Import process Status column.

In the example below, the AutoCare PIES Import Controller Screen displays the Overall Status and the Import process status as 'Importing files...' Additionally, the Background Process Link column displays a link to the Background Process Details screen as 'running.'

AutoCare PIES Imports

Process	File Name	Overall Status
No existing records		
<div style="text-align: center;"> ⏪ < 0 > ⏩ </div>		

Import Details

Process	Started Time	Duration	Started By	Status	Background Pr
Validation	2019-04-22 05:46:06	5 secs	PIESIMPORT	Validation completed	succeeded
Conversion	2019-04-22 05:46:11	5 secs	PIESIMPORT	Conversion completed	succeeded
Delta Calculation	2019-04-22 05:46:16	5 secs	PIESIMPORT	Delta calculation completed	succeeded
Import	2019-04-22 06:33:38	11 secs	USERB	Importing files...	running 11

⏪ < 1-4 of 4 > ⏩

➔ Start import
➔ Reject
➔ Discard file

- Optionally, click the 'running' Background Process Link (shown above) to view the details of the import process within the Background Process Details screen.

Once the import has completed, the Controller is automatically moved to the Import Completed state. For more information, see the **Import Completed State** topic within the **Import Framework** section.

For more information about PIES import validation rules and example error messages, see the **PIES Import Validation Rules** topic.

For more information on automotive workflow states, see the **Default Workflow States and Functions** topic.

Sample AutoCare PIES File Structure

The intention of this topic is to introduce the basic AutoCare PIES file structure and the way the objects are built in the system when a PIES file is imported into the system.

Currently, the Automotive solution supports import and export of three versions of PIES files: PIES 6.5, PIES 6.7 and PIES 7.0. All three versions share a common structure with additional add-on features available in the later versions. For the purposes of this topic, PIES 6.5 will be the PIES version referenced throughout the following text.

The PIES file shall include only plain text delimited by 'valid' XML tags (as defined below). A CR/LF (ASCII 13,10 decimal) should separate lines and it is recommended that the text should use UTF-8 character encoding. However, ISO-8859-1 is also accepted.

The PIES standard is designed in a flexible format which allows each record segment to loop as many times as needed in order to meet data delivery requirements. For example, a part with multiple price levels would have multiple price detail segments, one for each price level. Also, only those segments, which are needed to provide specific information, need be sent. To accomplish this flexibility, the PIES delivery standard is designed using XML and XML Schema Definition (XSD).

Delivering PIES Files

Sample File structure

Below is a sample PIES 6.5 file with one PIES Item.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<PIES xmlns="http://www.aftermarket.org">
  <Header>
    <PIESVersion>6.5</PIESVersion>
    <SubmissionType>FULL</SubmissionType>
    <BlanketEffectiveDate>2017-03-15</BlanketEffectiveDate>
    <BrandOwnerDUNS>123456789</BrandOwnerDUNS>
    <BrandOwnerGLN>4567890120003</BrandOwnerGLN>
    <TechnicalContact>krma</TechnicalContact>
    <ContactEmail>krma</ContactEmail>
  </Header>
  <PriceSheets>
    <PriceSheet MaintenanceType="A">
      <PriceSheetNumber>2017WD</PriceSheetNumber>
      <PriceSheetName>2017 WD Prices</PriceSheetName>

      <SupersededPriceSheetNumber>2016WD</SupersededPriceSheetNumber>
      <CurrencyCode>USD</CurrencyCode>
      <PriceZone>Western</PriceZone>
      <EffectiveDate>2017-03-07</EffectiveDate>
      <ExpirationDate>2018-03-07</ExpirationDate>
    </PriceSheet>
  </PriceSheets>
  <Items>
    <Item MaintenanceType="A">
```

```

    <HazardousMaterialCode>Y</HazardousMaterialCode>
    <BaseItemID>VC21</BaseItemID>
    <ItemLevelGTIN
GTINQualifier="UP">11234567890123</ItemLevelGTIN>
    <PartNumber>VC21499</PartNumber>
    <BrandAAIAID>DKGX</BrandAAIAID>
    <BrandLabel>Wako</BrandLabel>
    <SubBrandAAIAID>DMST</SubBrandAAIAID>
    <SubBrandLabel>SubBrand Label</SubBrandLabel>
    <ACESApplications>Y</ACESApplications>
    <ItemQuantitySize UOM="EA">1</ItemQuantitySize>
    <ContainerType>BO</ContainerType>
    <QuantityPerApplication Qualifier="VAR"
UOM="EA">1</QuantityPerApplication>
    <ItemEffectiveDate>2010-08-13</ItemEffectiveDate>
    <AvailableDate>2010-08-13</AvailableDate>
    <MinimumOrderQuantity UOM="EA">123</MinimumOrderQuantity>
    <ManufacturerProductCodes>
        <Group>02347</Group>
        <SubGroup>A</SubGroup>
    </ManufacturerProductCodes>
    <AAIAProductCategoryCode>199456</AAIAProductCategoryCode>
    <UNSPSC>11223348</UNSPSC>
    <PartTerminologyID>5696</PartTerminologyID>
    <VMRSCode>123000789</VMRSCode>
    <Descriptions>
        <Description MaintenanceType="A"
DescriptionCode="MKT">MKT</Description>
    </Descriptions>
    <Prices>
        <Pricing MaintenanceType="A" PriceType="WD1">
            <PriceSheetNumber>2017WD</PriceSheetNumber>
            <CurrencyCode>USD</CurrencyCode>
            <EffectiveDate>2008-10-22</EffectiveDate>
            <ExpirationDate>2012-10-22</ExpirationDate>
            <Price UOM="PE">10</Price>
        </Pricing>
    </Prices>
    <ExtendedInformation>
        <ExtendedProductInformation MaintenanceType="A"
EXPICode="HSB">01</ExtendedProductInformation>
        <ExtendedProductInformation MaintenanceType="A"
EXPICode="RCT">Marietta, GA</ExtendedProductInformation>
        <ExtendedProductInformation MaintenanceType="A"
EXPICode="MSD">MSDS26426</ExtendedProductInformation>
        <ExtendedProductInformation MaintenanceType="A"
EXPICode="RCS">Cobb, GA</ExtendedProductInformation>
        <ExtendedProductInformation MaintenanceType="A"
EXPICode="EMS">1</ExtendedProductInformation>
    </ExtendedInformation>
    </ExtendedInformation>

```

```

    <Packages>
      <Package MaintenanceType="A">
        <PackageLevelGTIN>12345000451747</PackageLevelGTIN>

    <ElectronicProductCode>4A.356E414.B351C7.AD331A465</ElectronicProductCode>

    <PackageBarCodeCharacters>005555010</PackageBarCodeCharacters>
      <PackageUOM>PK</PackageUOM>
      <QuantityofEaches>1</QuantityofEaches>
      <Dimensions UOM="CM">
        <Height>10.2</Height>
        <Width>3.6</Width>
        <Length>2.0</Length>
      </Dimensions>
      <Weights UOM="PG">
        <Weight>12.5</Weight>
        <DimensionalWeight>45</DimensionalWeight>
      </Weights>
      <WeightVariance>90</WeightVariance>
      <HazardousMaterial MaintenanceType="A"
LanguageCode="EN">
        <ShippingScope>INT</ShippingScope>
        <Bulk>N</Bulk>
        <RegulatingCountry>US</RegulatingCountry>
        <TransportMethod>R</TransportMethod>
        <Regulated>Y</Regulated>
        <Description>Hazmat Description</Description>
        <HazardousClass>2.2</HazardousClass>

    <HazardousMaterialCodeQualifier>D</HazardousMaterialCodeQualifier>

    <HazardousMaterialClassCode>1.4</HazardousMaterialClassCode>

    <HazardousMaterialDescription>Hazardous</HazardousMaterialDescription>
      <ShippingName>Great Lakes Shipping</ShippingName>
      <UNNAIDCode>027455</UNNAIDCode>

    <HazardousPlacardNotation>Caution</HazardousPlacardNotation>
      <WHMISCode>B.4</WHMISCode>
      <WHMISFreeText>Caution</WHMISFreeText>
      <PackingGroupCode>III</PackingGroupCode>

    <RegulationsExemptionCode>4</RegulationsExemptionCode>
      <TextMessage>Be careful</TextMessage>
      <OuterPackageLabel>ORM-AIR</OuterPackageLabel>
    </HazardousMaterial>
  </Package>
</Packages>
<PartInterchangeInfo>
  <PartInterchange MaintenanceType="A" LanguageCode="EN">
    <TypeCode>0</TypeCode>

```

```

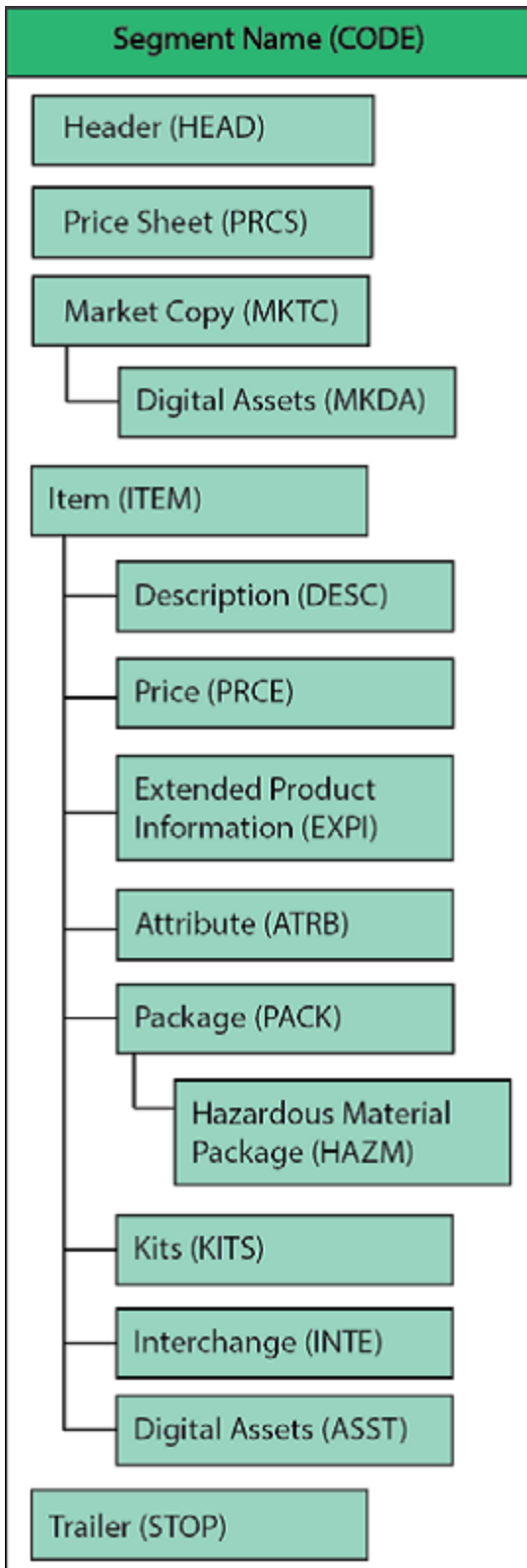
        <BrandAAIAID>BBTB</BrandAAIAID>
        <BrandLabel>Great Lakes</BrandLabel>
        <PartNumber>GL26409</PartNumber>
        <QualityGradeLevel>P</QualityGradeLevel>
        <InterchangeNotes>Yellow indicator
light</InterchangeNotes>
        <InternalNotes>May be difficult to
remove</InternalNotes>
    </PartInterchange>
</PartInterchangeInfo>
<DigitalAssets>
    <DigitalFileInformation MaintenanceType="A"
AssetID="Asset456" LanguageCode="EN">
        <FileName>VC21499_P04</FileName>
        <AssetType>P04</AssetType>
        <Representation>R</Representation>
        <Resolution>96</Resolution>
        <Background>CLI</Background>
        <OrientationView>BAC</OrientationView>
        <Details>Details/Description</Details>
        <FilePath>images\SVlight.tif</FilePath>
        <URI>www.stibo.com</URI>
        <FileDateModified>2008-04-11</FileDateModified>
        <EffectiveDate>2008-04-11</EffectiveDate>
        <ExpirationDate>2012-04-11</ExpirationDate>
        <Country>US</Country>
    </DigitalFileInformation>

</DigitalAssets>
<Kits>
    <Kit MaintenanceType="A">
        <ComponentPartNumber
IDQualifier="VN">VC36004</ComponentPartNumber>
        <ComponentBrand>DKGX</ComponentBrand>
        <Description DescriptionCode="SHO"
LanguageCode="EN">CKENGSENS</Description>
        <QuantityInKit UOM="EA">2</QuantityInKit>
        <SequenceCode>1</SequenceCode>
    </Kit>
    <Kit MaintenanceType="A">
        <ComponentPartNumber
IDQualifier="VP">VC36009</ComponentPartNumber>
        <ComponentBrand>DKGX</ComponentBrand>
        <Description DescriptionCode="SHO"
LanguageCode="EN">CKENGSENS</Description>
        <QuantityInKit UOM="EA">1</QuantityInKit>
        <SequenceCode>2</SequenceCode>
    </Kit>
</Kits>
</Item>
</Items>

```

</PIES>

A normal PIES file has five segments in it: Header, Price Sheet (PRCS), Market Copy (MKTC), Item (ITEM), and the Trailer. These five segments also includes some additional segments within them. The way it is structured is demonstrated in the screenshot below.



Note: STEP PMDM for Automotive solution currently do not handle the Market Copy segment. If Market Copy segment is included in the PIES file, it will cause the import to fail Validation.

A description of each of these five segments and some segments within them follows below.

Header

The Header constitutes the initial section of the file. This segment defines trading partner information and global value characteristics. There can only be one instance of this segment within a PIES file. All Header information is contained within <Header> tags.

The order of the tags within the Header is defined by the XML schema. In the Header section, the <PIESVersion> and <SubmissionType> tags are required to be populated. If they are left empty, the XML file will fail to load and will be moved to an Error state in the workflow.

Note: Although element <BrandOwnerAAIAID> is optional, if the <Item> segment doesn't have the <BrandAAIAID> defined (an element within the <Item> segment), then <BrandOwnerAAIAID> would need to be defined in the <Header> segment in order for the PIES Items to get created.

In the table below, tags included within the Header section of the PIES file are explained. For more information on the tags available in the PIES Header section, see the **AutoCare PIES Exporter** topic within the **Exporting Automotive Data** section of **Automotive Reference Guide**.

XML Element	Requirement	Description	Example XML Code
PIESVersion	Required		<PIESVersion>6.5</PIESVersion>
SubmissionType	Required		<SubmissionType>FULL</SubmissionType>
BlanketEffectiveDate	Optional	"Blanket Effective Date" acts as the global default value for a particular PIES file. "Blanket Effective Date" may be overridden at Price Sheet Header, Item, or Price Segment Levels.	<BlanketEffectiveDate>2013-01-31</BlanketEffectiveDate>
ChangesSinceDate	Conditional	"Changes Since Date" is a control date indicating the	<ChangesSinceDate>...</ChangesSinceDate>

XML Element	Requirement	Description	Example XML Code
		date the last PIES file was generated. This field is MANDATORY if the tag 'Submission Type', has been filled with the value 'UPDATE'	
ParentDUNSNumber	Optional	Unique 9-digit Dun & Bradstreet ID. May also use +4 format with 13 digits. The D&B D-U-N-S Number is a unique nine-digit identification sequence, which provides unique identifiers of single business entities, while linking corporate family structures together	<ParentDUNSNumber>999999990001</ParentDUNSNumber>
ParentGLN	Optional	GS1 Company ID + Location ID + Check Digit. The GLN is a standard means of identifying global trading partner	<ParentGLN>77777755551</ParentGLN>

XML Element	Requirement	Description	Example XML Code
		<p>locations. Through the process of Product Synchronization, the seller and buyer identify each other's names, addresses, and other information, so that all subsequent electronic documents can be identified, routed, and processed using only these codes. This is essential for the smooth, automated, error-free processing of electronic documents.</p>	
ParentVMRSID	Optional	A unique Company identifier for all participating Heavy Duty parts manufacturers.	<ParentVMRSID>GIANT</ParentVMRSID>
ParentAAIAID	Optional	This field is recommended to use when identifying Parent Company Ownership.	<ParentAAIAID>BBCD</ParentAAIAID>

XML Element	Requirement	Description	Example XML Code
BrandOwnerDUNS	Optional	<p>Unique 9-digit Dun & Bradstreet ID. May also use +4 format with 13 digits. The D&B D-U-N-S Number is a unique nine-digit identification sequence, which provides unique identifiers of single business entities, while linking corporate family structures together</p>	<p><BrandOwnerDUNS>88888888001</BrandOwnerDUNS></p>
BrandOwnerGLN	Optional	<p>GS1 Company ID + Location ID + Check Digit. The GLN is a standard means of identifying global trading partner locations. Through the process of Product Synchronization, the seller and buyer identify each other's names, addresses, and other information, so that all</p>	<p><BrandOwnerGLN>777777123459</BrandOwnerGLN></p>

XML Element	Requirement	Description	Example XML Code
		<p>subsequent electronic documents can be identified, routed, and processed using only these codes. This is essential for the smooth, automated, error-free processing of electronic documents.</p>	
BrandOwnerVMRSID	Optional	A unique Company identifier for all participating Heavy Duty parts manufacturers.	<BrandOwnerVMRSID>WONDR</BrandOwnerVMRSID>
BrandOwnerAAIAID	Optional	This field is recommended to be populated when identifying Brand Owner of the data supplied in the PIES file.	<BrandOwnerAAIAID>BRST</BrandOwnerAAIAID>
BuyerDuns	Optional	D&B D-U-N-S Number for PIES Trading Partner. The D&B D-U-N-S Number is a unique nine-digit identification	<BuyerDuns>888888880001</BuyerDuns>

XML Element	Requirement	Description	Example XML Code
		sequence, which provides unique identifiers of single business entities, while linking corporate family structures together	
CurrencyCode	Optional	The default value for the entire PIES file. This value may be overridden at the PRCS (Price Sheet Segment) or PRCE (Pricing Segment) levels.	<CurrencyCode>USD</CurrencyCode>
LanguageCode	Optional	The default value for the entire PIES file. This value may be overridden within many of the underlying Segments.	<LanguageCode>EN</LanguageCode>
TechnicalContact	Optional	Name of Contact for resolving technical issues with PIES file.	<TechnicalContact>John Smith</TechnicalContact>
ContactEmail	Optional	Contact Email address at data Supplier/Sender company for resolving issues with	<ContactEmail>john@smith.com</ContactEmail>

XML Element	Requirement	Description	Example XML Code
		PIES file or receiving file processing reports.	
PCdbVersionDate	Optional	Version date of the PCdb used to create the file. The date can be found in the 'Version' table of the PCdb.	<PCdbVersionDate>2014-06-27</PCdbVersionDate>
PAdbVersionDate	Optional	Version date of the PAdb used to create the file. The date can be found in the 'Version' table of the PAdb.	<PAdbVersionDate>2014-06-27</PAdbVersionDate>

Price Sheet Header Segment (PRCS)

The Price Sheet Header Segment is used to define from zero to any number of price sheets. When exchanging pricing data with a trading partner, it is recommended to define at least one Price Sheet Header. The <PriceSheetNumber> defined in the Price Sheet Header segment will be referenced within the Price Segment for an individual PIES Item record.

The Price Sheet Header Segment is defined within the <PriceSheets> tag. This segment is an optional looping segment within the specification, meaning there may be zero instances of this segment in a PIES file. If this segment is used, there must be one or more instances of a child <PriceSheet> </PriceSheet> loop. Multiple loops of a <PriceSheet> </PriceSheet> section may be used to define multiple Price Sheets.

A sample PIES Sheet Header Segment and a screenshot of how the segment is populated in STEP are displayed below. The Price Sheet information is created as Data Container valid on the PIES Products object (root node).

```
<PriceSheets>
  <PriceSheet MaintenanceType="A">
    <PriceSheetNumber>2017WD</PriceSheetNumber>
    <PriceSheetName>2017 WD Prices</PriceSheetName>

  <SupersededPriceSheetNumber>2016WD</SupersededPriceSheetNumber>
    <CurrencyCode>USD</CurrencyCode>
    <PriceZone>Western</PriceZone>
```

```

    <EffectiveDate>2017-03-07</EffectiveDate>
    <ExpirationDate>2018-03-07</ExpirationDate>
  </PriceSheet>
</PriceSheets>

```

ID	Currency Code	Effective Date	Expiration Date	Price Sheet Name	Price Sheet Number	Price Zone	Superseded Price Sheet Number
PIES_PriceSheet_1	U.S. Dollar	2017-03-07	2018-03-07	2017 WD Prices	2017WD	Western	2016WD

PIES Item Segment (ITEM)

The Item segment is the core looping (or repeating) segment within the PIES standard. Each instance of this segment is responsible for defining individual Item Part Numbers (which are also called PIES Items).

All <Item> instances are contained within the parent <Items> element. For each Item Part Number (PIES Item) to be defined, a new instance of <Item> and its child segments are initiated. Many of the child segments under <Item> are themselves segments of PIES which are defined later in this topic.

The following XML sample code shows the upper level <Items> element and the beginning of a single instance of <Item>. The child segments are part of the logical grouping known as the Item Segment. Further segments continue from the end of this code sample.

A sample PIES XML PIES Item structure and a screenshot of how it is populated in STEP is displayed below.

```

<Items>
  <Item MaintenanceType="A">
    <HazardousMaterialCode>Y</HazardousMaterialCode>
    <BaseItemID>VC21</BaseItemID>
    <ItemLevelGTIN
GTINQualifier="UP">11234567890123</ItemLevelGTIN>
    <PartNumber>VC21499</PartNumber>
    <BrandAAIAID>DKGX</BrandAAIAID>
    <BrandLabel>Wako</BrandLabel>
    <SubBrandAAIAID>DMST</SubBrandAAIAID>
    <SubBrandLabel>SubBrand Label</SubBrandLabel>
    <ACESApplications>Y</ACESApplications>
    <ItemQuantitySize UOM="EA">1</ItemQuantitySize>
    <ContainerType>B0</ContainerType>
    <QuantityPerApplication Qualifier="VAR"
UOM="EA">1</QuantityPerApplication>
    <ItemEffectiveDate>2010-08-13</ItemEffectiveDate>
    <AvailableDate>2010-08-13</AvailableDate>
  </Item>
</Items>

```

```
<MinimumOrderQuantity UOM="EA">123</MinimumOrderQuantity>
<ManufacturerProductCodes>
  <Group>02347</Group>
  <SubGroup>A</SubGroup>
</ManufacturerProductCodes>
<AAIAProductCategoryCode>199456</AAIAProductCategoryCode>
<UNSPSC>11223348</UNSPSC>
<PartTerminologyID>5696</PartTerminologyID>
<VMRSCode>123000789</VMRSCode>

<Descriptions> ... </Descriptions>
<Prices> ... </Prices>
<ExtendedInformation> ... </ExtendedInformation>
<Packages> ... </Packages>
<PartInterchangeInfo> ... </PartInterchangeInfo>
<DigitalAssets> ... </DigitalAssets>
<Kits> ... </Kits>
</Item>
</Items>
```

The screenshot displays the Stibo Systems Master Data Management interface. On the left is a hierarchical tree view of the product structure. The right pane shows the detailed data for the selected product, 'VC21499 rev.0.5 - Product', organized into two segments: 'Description' and 'PIES Item Segment ITEM'.

Description	
Name	Value
ID	AC_PIESItem_DKGX_VC21499
Name	VC21499
Object Type	PIES Item
Revision	0.5 Last edited by ACESIMPORT on Fri Mar 08 07:51:37 EST 2019
Approved	Never Been Approved
Translation	Not Translated
Path	Primary Product Hierarchy/AutoCare Root/PIES Products/Wako/Eng
Changed Object	abc true
Delete Status	abc
New Object	abc

PIES Item Segment ITEM	
Name	Value
Hazardous Material Code	Y
Base Item Number	abc VC21
Item Level GTIN	123 11234567890123
Item Level GTIN Qualifier	U.P.C./EAN Consumer Package Code (1-5-5-1)
Part Number	12a VC21499
Brand AAIA ID	12a DKGX
Brand Label	12a Wako
SubBrand AAIA ID	12a DMST
Sub Brand Label	12a SubBrand Label
ACES Applications	Y
Item Quantity Size	123 1
Container Type	Bottle
Quantity Per Application Qual	Variable - See ACES Application Field V91
Quantity Per Application	123 1
Quantity Per Application UOM	abc EA
Item Effective Date	2010-08-13
Minimum Order Quantity	123 123
Minimum Order Quantity UOM	abc EA
Group	12a 02347
Sub Group	12a A
AAIA Product Category Code	abc 199456
UNSPSC Code	123 11223348
VMRS Code (Heavy Duty)	12a 123000789
Available Date	2010-08-13
Item Quantity Size UOM	abc EA

PIES Description Segment

The Description segment allows you to send one or as many descriptions of an Item record part number (PIES Item). The flexibility to send multiple descriptions on a single PIES Item is possible because of the capabilities inherent in XML's looping structure. The Description segment (<Descriptions>...</Descriptions>) is an element group available within the <Item> tag. To send multiple descriptions about an individual Item part number (PIES Item), a user can loop (repeat) the <Description> element, changing the 'DescriptionCode' attribute as well as the string value for the Description.

A sample PIES Description Segment and a screenshot that shows how it is populated in STEP is displayed below.

Note: The Description (DESC) segment is created as Data Containers on the PIES Item starting with Automotive 9.1 MP1. Earlier versions of the PMDM for Automotive solution had the Description (DESC) segment stored as attribute values within an attribute group called 'PIES Description Segment DESC'.

Important: When a Data Container is deleted from a Product or Entity object, its ID cannot be revived, nor can its ID be reused until the object is purged from STEP. This is a known core STEP gap (not Automotive specific). User needs to be aware of this behavior while importing files to test and happen to delete the Data Container on the PIES Item. The next time when user imports the same file, that Data Container will not be created because its ID was in use and then deleted (the import process generates the Data Container ID through a hash function). But if the Data Containers are manually created in the workbench / Web UI, then user will not encounter the ID problem because a new ID is auto-generated for each new Data Container.

```
<Descriptions>
<Description MaintenanceType="A" Sequence="1" DescriptionCode="ABR">Driving
Lamp</Description>
<Description MaintenanceType="A" Sequence="2" DescriptionCode="DES">Optilux®
Model 2500 Angel Eye Driving Lamp (Single Lamp)</Description>
<Description MaintenanceType="A" Sequence="3" DescriptionCode="DES">Optilux®
Model 2500 Angel Eye Driving Lamp (Single Lamp)</Description>
<Description MaintenanceType="A" DescriptionCode="EXT">Optilux® Model 2500
Angel Eye Driving Lamp (Single Lamp)</Description>
<Description MaintenanceType="A" DescriptionCode="INV">OPTILUX 2500 ANGEL EYE
SNGL</Description>
<Description MaintenanceType="A" DescriptionCode="MKT">Optilux® Model 2500
Angel Eye Driving Lamp (Single Lamp)</Description>
<Description MaintenanceType="A" DescriptionCode="SHO">Driving
Lamp</Description>
<Description MaintenanceType="A" DescriptionCode="LAB">OPTILUX 2500 ANGEL EYE
SNGL</Description>
<Description MaintenanceType="A" DescriptionCode="KEY">driving light
optilux</Description>
</Descriptions>
```

ID	PIES Description	PIES Desc Type	PIES Description Sequence Code
DESC_03447fb75c8e759965af33d5b2038f4f	Optilux® Model 2500 Angel Eye Driving Lamp (Single Lamp)	Product Description - Extended 240	
DESC_0ee2a5687dda12f2f9544186f8a55aca	Optilux® Model 2500 Angel Eye Driving Lamp (Single Lamp)	Marketing Description - 2000	
DESC_18616dcb6f709ff7104992136c4a10f6	driving light optilux	Key Search Words - 2000	
DESC_25b567cd13f219b7213c207f1bf9a81c	Driving Lamp	Product Description - Abbreviated - 12	1
DESC_31889ee2e23ef47f03324ba3bfd5ff2f	Optilux® Model 2500 Angel Eye Driving Lamp (Single Lamp)	Product Description - Long - 80	3
DESC_7f934f210cbdeb96075a30e8d675c6d5	OPTILUX 2500 ANGEL EYE SNGL	Label Description - 80	
DESC_8d77550728df5bc6f45518526ca1bc74	Driving Lamp	Product Description - Short - 20	
DESC_8ed1da88cf7e3b81af37302cff0fa29d	OPTILUX 2500 ANGEL EYE SNGL	Product Description - Invoice - 40	
DESC_ac00a8974a61f96d60ceee86c9e56f01	Optilux® Model 2500 Angel Eye Driving Lamp (Single Lamp)	Product Description - Long - 80	2

PIES Price Segment PRCE

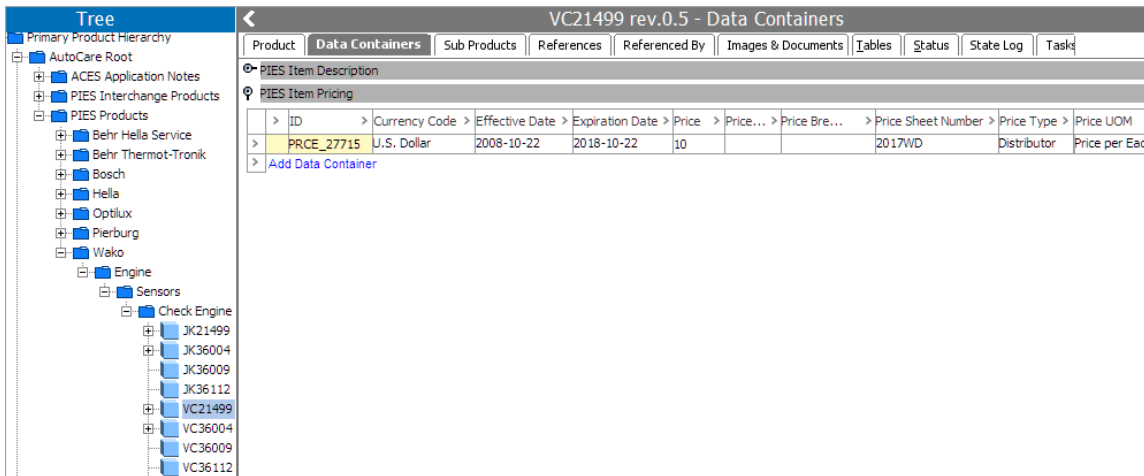
The Pricing Segment is defined by the PIES XML parent element <Prices>. Within this element group, there may be multiple instances of <Pricing> that define a single pricing instance. This element takes a 'Price Type' as an attribute indicating which type of price the following child segments needs to be defined in.

To supply multiple prices (price types) for an Item Part Number (PIES Item), the user needs to repeat the <Pricing> XML structure and redefine the new loop with the 'Price Type' attribute.

A sample PIES Price Segment and a screenshot of how it is populated in the STEP system is displayed below.

Note: The PIES Price Segment is created as Data Containers on the PIES Item starting with Automotive 9.1 MP1. Earlier versions of the PMDM for Automotive solution had the PIES Price Segment stored as attribute values within an attribute group.

```
<Prices>
  <Pricing MaintenanceType="A" PriceType="WD1">
    <PriceSheetNumber>2017WD</PriceSheetNumber>
    <CurrencyCode>USD</CurrencyCode>
    <EffectiveDate>2008-10-22</EffectiveDate>
    <ExpirationDate>2012-10-22</ExpirationDate>
    <Price UOM="PE">10</Price>
  </Pricing>
</Prices>
```



PIES Extended Product Info Segment (EXPI)

The Extended Product Information (EXPI) Segment is designed to relay information about an Item Part (PIES Item). The PIES Extended Product Information Segment contains information ranging from 'Country of Origin' to 'Warranty Terms'.

The format of the data expressed in the 'EXPI Data' field (<ExtendedProductInformation>) is dependent on the 'EXPI Code' selected.

To express information covering multiple (or one) 'EXPI Codes', the <ExtendedProductInformation> element is looped once for each code. Any and all instances of <ExtendedProductInformation> are grouped under the <ExtendedInformation> element.

A sample PIES Extended Product Info Segment and a screenshot of showing how it is populated in STEP is displayed below.

```
<ExtendedInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="HSB">01</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="HAC">HAC</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="CTS">FR</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="LIF">0</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="NAF">1</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="HTS">02</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="HZ1">12</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="TAX">Y</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="LIS">Proposed</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
```

```

EXPICode="TMC">AB</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="NPD">Top 60%</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="NPC">A</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="WS2">LF</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="WS1">SR</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="WD2">MI</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="CXP">12</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="RST">GA</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="CPN">CPN</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="WT2">MO</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="CGR">CGR</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="WT1">36</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="CCL">CCL</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="MSR">Y</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="RCT">Marietta, GA</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="MSD">MSDS26426</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="RCS">Cobb, GA</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="EMS">1</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="RPC">30060</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="CTR">GB</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="CTQ">DE</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="CTP">DK</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="CTO">US</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="OSN">OSN</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="OEP">OEP</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="OEM">OEM</ExtendedProductInformation>

```

```
        <ExtendedProductInformation MaintenanceType="A"  
EXPICode="REF">Y</ExtendedProductInformation>  
        <ExtendedProductInformation MaintenanceType="A"  
EXPICode="PTS">PTS</ExtendedProductInformation>  
        <ExtendedProductInformation MaintenanceType="A"  
EXPICode="PTN">PTN</ExtendedProductInformation>  
        <ExtendedProductInformation MaintenanceType="A"  
EXPICode="OSP">OSP</ExtendedProductInformation>  
        <ExtendedProductInformation MaintenanceType="A"  
EXPICode="WD1">500</ExtendedProductInformation>  
        <ExtendedProductInformation MaintenanceType="A"  
EXPICode="RET">RET</ExtendedProductInformation>  
        <ExtendedProductInformation MaintenanceType="A"  
EXPICode="REM">N</ExtendedProductInformation>  
        <ExtendedProductInformation MaintenanceType="A"  
EXPICode="STA">S</ExtendedProductInformation>  
        <ExtendedProductInformation MaintenanceType="A"  
EXPICode="PLM">20</ExtendedProductInformation>  
        <ExtendedProductInformation MaintenanceType="A"  
EXPICode="PLC">100</ExtendedProductInformation>  
    </ExtendedInformation>
```

Tree

- Assets
- AutoCare Root
- Configurations
- Index Words
- NAPA Reference Data
- TecDoc
- TecDoc Assets Root
- Import Flow Root
- TecDoc Manufacturer Root
- TecDoc Resource Root
- TecDoc Supplier Price Lists
- Publications
- Primary Product Hierarchy
 - Products
 - Discontinued Products
 - Product Overrides
 - Packaging
 - AutoCare Root
 - ACES Application Notes
 - PIES Interchange Products
 - PIES Products
 - 034MOTORSPORT
 - BCVC
 - Behr Hella Service
 - Behr Thermot-Tronik
 - Exide
 - Hella
 - Pierburg
 - Wako
 - Engine
 - Sensors
 - Check Engine Light Senso
 - JK21499
 - JK36004
 - JK36009
 - JK36112
 - VC21499**
 - VC36004
 - VC36009

- Collections
- eCatalogs
- Recycle Bin

VC21499 rev.0.5 - Product

Product	Sub Products	References	Referenced By	Images & Documents	Commercial
ID		AC_PIESItem_DKGX_VC21499			
Name		VC21499			
Object Type		PIES Item			
Revision		0.5 Last edited by ACESIMPORT on Fri Mar 08 07:51:37 EST 2019			
Approved		✘ Never Been Approved			
Translation		Not Translated			
Path		Primary Product Hierarchy/AutoCare Root/PIES Products/Wako/Eng			

PIES Extended Product Info Segment EXPI		
Name		Value
Canadian Harmonizing Tariff	t2a	HAC
Core Class	t2a	CCL
Core Group	t2a	CGR
Core Part Number	t2a	CPN
Core Return Days to Expiry	t23	12
Country of Origin (Primary)	abc	US
Country of Origin 2	abc	DK
Country of Origin 3	abc	DE
Country of Origin 4	abc	GB
Country of Origin 5	abc	FR
Emission Code	t23	1
Harmonized Tariff Code (HTS)	abc	02
Harmonized Tariff Code (Sche)	abc	01
Item (SKU) Level Special Hand	abc	12
Life Cycle Status Code	abc	0
Life Cycle Status Description	t2a	Proposed
Maximum Cases per Pallet La	t23	100
MSDS Required Flag	abc	Y
MSDS Sheet Number	t2a	MSDS26426
NAFTA Preference Criterion C	abc	1
National Popularity Code	abc	A
National Popularity Descriptio	t2a	Top 60%
OEM's Part number	t2a	OEP
Original Equipment Manufactu	t2a	OEM
Original Supplier	t2a	OSN
Original Supplier Part Number	t2a	OSP
Pallet Layer Maximum	t23	20
Part Number - Old	t2a	PTN
Part Number Superseded To	t2a	PTS
Refurbished Part	abc	Y
Regulating City, State	t2a	Marietta, GA

Packaging Segment (PACK)

The Packaging Segment is used to express various volumetric details for different packaging levels of PIES Items. Package information is grouped within the <Packages> child element of <Item>. Each instance of the <Package> element (looped beneath <Packages>) and its child segments corresponds to a distinct 'Package UOM'. This UOM, or Unit of Measure, identifies the type of packaging the data element contains.

The consumer-level package (UOM type 'PK') should always be defined if possible. The consumer-level pack can be defined, for example, as the Item Part in the packaging used for in-store display and stocking. This is the package level that an individual consumer may purchase.

The sample XML code below shows the Packaging Segment in relation to the other segments defined to this point, and the screenshot below shows the way it is populated in the system where the packaging object is created as an object below the PIES Item.

```
<Packages>
    <Package MaintenanceType="A">
        <PackageLevelGTIN>12345000451747</PackageLevelGTIN>

<ElectronicProductCode>4A.356E414.B351C7.AD331A465</ElectronicProductCode>

<PackageBarCodeCharacters>005555010</PackageBarCodeCharacters>
    <PackageUOM>PK</PackageUOM>
    <QuantityofEaches>1</QuantityofEaches>
    <Dimensions UOM="CM">
        <Height>10.2</Height>
        <Width>3.6</Width>
        <Length>2.0</Length>
    </Dimensions>
    <Weights UOM="PG">
        <Weight>12.5</Weight>
        <DimensionalWeight>45</DimensionalWeight>
    </Weights>
    <WeightVariance>90</WeightVariance>
    <HazardousMaterial> ... </HazardousMaterial>
    </Package>
</Packages>
```


The sample XML code below shows how a single instance of the Hazardous Material Package Segment, defined by the <HazardousMaterial> parent element, falls within the Packaging Segment (<Package>), and the screenshot below shows how it is populated in the system.

```
<HazardousMaterial MaintenanceType="A" LanguageCode="EN">
    <ShippingScope>INT</ShippingScope>
    <Bulk>N</Bulk>
    <RegulatingCountry>US</RegulatingCountry>
    <TransportMethod>R</TransportMethod>
    <Regulated>Y</Regulated>
    <Description>Hazmat Description</Description>
    <HazardousClass>2.2</HazardousClass>

<HazardousMaterialCodeQualifier>D</HazardousMaterialCodeQualifier>

<HazardousMaterialClassCode>1.4</HazardousMaterialClassCode>

<HazardousMaterialDescription>Hazardous</HazardousMaterialDescription>
    <ShippingName>Great Lakes Shipping</ShippingName>
    <UNNAIDCode>027455</UNNAIDCode>

<HazardousPlacardNotation>Caution</HazardousPlacardNotation>
    <WHMISCode>B.4</WHMISCode>
    <WHMISFreeText>Caution</WHMISFreeText>
    <PackingGroupCode>III</PackingGroupCode>
    <RegulationsExemptionCode>4</RegulationsExemptionCode>
    <TextMessage>Be careful</TextMessage>
    <OuterPackageLabel>ORM-AIR</OuterPackageLabel>
</HazardousMaterial>
```

The screenshot displays a software interface with a 'Tree' view on the left and a 'Hazmat' data table on the right. The tree view shows a hierarchy starting from 'Assets' down to 'VC21499' and 'Hazmat Description'. The 'Hazmat' table provides detailed information about the hazardous material, including its ID, name, object type, revision, and various regulatory attributes.

Description	
Name	Value
ID	AC_PIESHazMat_12345000451747_INT_N_R
Name	Hazmat Description
Object Type	PIES Haz Mat
Revision	0.3 Last edited by PIESIMPORT on Mon Jul 02 10:13:24 EDT 2018
Approved	Never Been Approved
Translation	Not Translated
Path	Primary Product Hierarchy/AutoCare Root/PIES Products/Wako/Engi

PIES Hazardous Material Package Segment HAZM	
Name	Value
Shipping Scope	INT
Bulk	N
Regulating Country	United States of America (the)
Transport Method	R
Regulated	Y
Description	Hazmat Description
Hazardous Class	2.2
Hazardous Material Code Qu	D
Hazardous Material Class Coc	1.4
Hazardous Material Descriptio	Hazardous
Shipping Name	Great Lakes Shipping
UN/NA ID Code	027455
Hazardous Placard Notation	Caution
WHMIS Code	B.4
WHMIS Free Text	Caution
Packing Group Code	III
Regulations Exemption Code	4
Text Message	Be careful
Outer Package Label	ORM-AIR

Kits Segment (KITS)

The Kits Segment is defined by the <Kits> parent element which contains the child element <KitComponent>. This element contains child segments that define the component parts and supporting data.

If the item is a kit or set, the Kit Segment of PIES conveys the part numbers contained in a kit or set Bill of Materials (BOM).

Kits are defined as an assemblage of parts or equipment that serve the specific purpose of performing a full maintenance procedure or function.

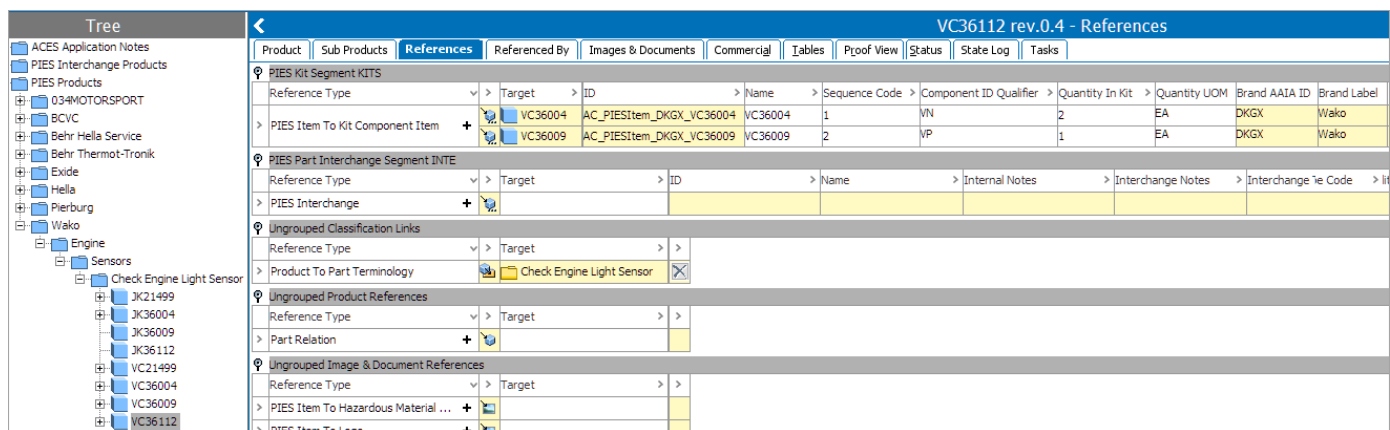
A Set is a group or collection of things that belong together, resemble one another, or are usually found together. A range of tools (such as a drill bit set, or a tool assortment), or a battery booster cable set are examples of sets.

The Kits Segment allows the data sender to express what components are in a particular kit. To send multiple components within a Kit, the segments within the Kit Segment repeat as many times as necessary to identify all components.

If the Kit object does not already exist in STEP, then the import will create the Kit as a PIES Item object type. And will establish a reference from the PIES Item to the Kit PIES Item through the 'PIES Item To Kit Component Item' reference type. The Kit Sequence, Quantity, UOM, and Qualifier is populated as metadata attributes on the Reference Type within the References tab.

A sample Kits Segment and a screenshot showing how it is populated on STEP is displayed below.

```
<Kits>
    <Kit MaintenanceType="A">
        <ComponentPartNumber
IDQualifier="VN">VC36004</ComponentPartNumber>
        <ComponentBrand>DKGX</ComponentBrand>
        <Description DescriptionCode="SHO"
LanguageCode="EN">CKENGSENS</Description>
        <QuantityInKit UOM="EA">2</QuantityInKit>
        <SequenceCode>1</SequenceCode>
    </Kit>
    <Kit MaintenanceType="A">
        <ComponentPartNumber
IDQualifier="VP">VC36009</ComponentPartNumber>
        <ComponentBrand>DKGX</ComponentBrand>
        <Description DescriptionCode="SHO"
LanguageCode="EN">CKENGSENS</Description>
        <QuantityInKit UOM="EA">1</QuantityInKit>
        <SequenceCode>2</SequenceCode>
    </Kit>
</Kits>
```



Interchange Segment (INTE)

The Interchange Segment in PIES is used to relay interchange data for the Item Part Number (PIES Item) in relation to alternative Brand Owner Part Numbers. The Interchange part number is also known as Competitor / OE Number in STEP system. The segment is defined by the <PartInterchangeInfo> grouping element. Within

<PartInterchangeInfo> is the <PartInterchange> element which contains the child segments that define an interchange. <PartNumber> element instance is repeated as many times as necessary to identify all components.

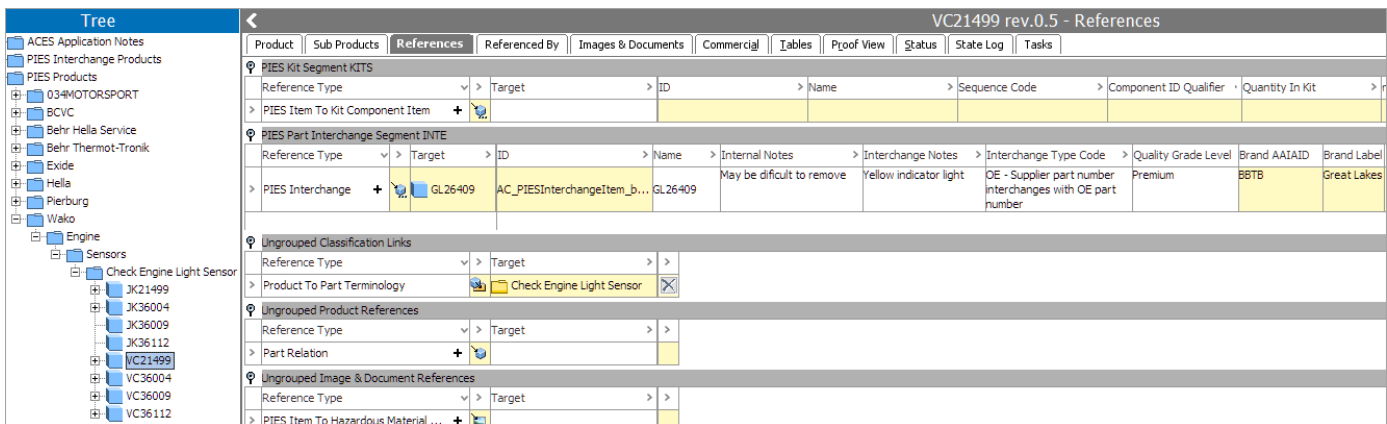
All interchange part numbers are defined by both the interchange part number and a 'Brand AAIAID'. The 'Brand AAIAID' value for the interchange supplier can be found in the industry supported Brand table. If a brand code value is not present in the Industry Brand Code table, the use of 'ZZZN' is permitted until such time as a 'Brand AAIAID' is created for the Brand in question. In such an instance, the 'Brand Label' value should be populated for clarity. Using 'ZZZN' and a 'Brand label' should be used as a last resort not as standard practice as the Brand table is updated frequently.

As with some of the other segments described above, to express multiple interchanges for an Item Part Number, the <PartInterchange> element is repeated as many times as necessary to identify all interchanges.

If the Interchange object does not already exist in STEP, then the import will create the Interchange object as a PIES Interchange Item object type. And will establish a reference from the PIES Item to the PIES Interchange Item through the 'PIES Interchange' reference type. The PIES Interchange data is populated as metadata attributes on the Reference Type within the References tab.

A sample Interchange Segment and a screenshot showing how it is populated in STEP is displayed below.

```
<PartInterchangeInfo>
  <PartInterchange MaintenanceType="A" LanguageCode="EN">
    <TypeCode>O</TypeCode>
    <BrandAAIAID>BBTB</BrandAAIAID>
    <BrandLabel>Great Lakes</BrandLabel>
    <PartNumber>GL26409</PartNumber>
    <QualityGradeLevel>P</QualityGradeLevel>
    <InterchangeNotes>Yellow indicator light</InterchangeNotes>
    <InternalNotes>May be difficult to remove</InternalNotes>
  </PartInterchange>
</PartInterchangeInfo>
```



Digital Asset File Information Segment (ASST)

The Digital Asset File Information Segment is used to relay information about many different media types which support the PIES Item. The 'Asset Type' field is used to identify what type of digital asset is being defined.

This segment is defined by the <DigitalAssets> grouping element. Below this element is the <DigitalFileInformation> element which contains corresponding child segments (fields) used to express the defined values for a digital asset. As with other segments, to supply information for multiple digital assets for a PIESItem, the <DigitalFileInformation> element and its child segments are repeated for each digital asset.

AutoCare Easy Setup creates the Asset Type attribute and LOV, and the PCdb importer populates the values for the LOV. Easy Setup also creates following six PIES asset object types:

- PIES Asset (General)
- PIES Hazardous Material Asset (AssetType = HMS)
- PIES Logo (AssetType = LGO)
- PIES Material Safety Data Sheet (AssetType = MSD)
- PIES Owners Manual (AssetType = OWN)
- PIES Product Image (AssetType = P04)

And following six Image and Document Reference Types:

- PIES Item To Hazardous Material Asset (AssetType = HMS)
- PIES Item To Logo (AssetType = LGO)
- PIES Item To Material Safety Data Sheet (AssetType = MSD)
- PIES Item To Owners Manual (AssetType = OWN)
- PIES Item To Product Image (AssetType = P04)
- PIES To Asset

Five asset types (HMS, LGO, MSD, OWN, and P04) will get its own object type and reference type, and all other asset types will be created as PIES Asset (General) object type and referenced using 'PIES To Asset' reference type.

A sample Digital Asset File Information Segment and a screenshot showing how it is populated in STEP is displayed below.

```
<DigitalAssets>
  <DigitalFileInformation MaintenanceType="A" AssetID="Asset456"
  LanguageCode="EN">
    <FileName>VC21499_P04</FileName>
    <AssetType>P04</AssetType>
    <Representation>R</Representation>
    <Resolution>96</Resolution>
    <Background>CLI</Background>
    <OrientationView>BAC</OrientationView>
    <Details>Details/Description</Details>
    <FilePath>images\SVlight.tif</FilePath>
    <URI>www.stibo.com</URI>
    <FileDateModified>2008-04-11</FileDateModified>
    <EffectiveDate>2008-04-11</EffectiveDate>
    <ExpirationDate>2012-04-11</ExpirationDate>
    <Country>US</Country>
  </DigitalFileInformation>
```

</DigitalAssets>

In the screenshot below, the Asset ID 'VC21499_P04' is linked to the PIES Item 'VC21499' via the reference type 'PIES Item To Product Image.'

Reference Type	Target	ID
> PIES Item To Kit Component Item		
> PIES Interchange	GL26409	AC_PIESInterchangeItem_b... GL26409
> Product To Part Terminology	Check Engine Light Sensor	
> PIES Item To Hazardous Material ...		
> PIES Item To Logo		
> PIES Item To Material Safety Dat...		
> PIES Item To Owners Manual		
> PIES Item To Product Image	VC21499_P04	
> PIES To Asset		

Asset details are populated as attribute values on the asset object (listed under the PIES Assets folder) based on the input data in the PIES file as shown in the screenshot below.

Tree

- THD 40RT.jpg
- THD 40RX.jpg
- THD 51RT.jpg
- THD 51RX.jpg
- THD 59T.jpg
- THD 65T.jpg
- THD 65X.jpg
- THD 75DTT.jpg
- THD 75T.jpg
- THD 75X.jpg
- THD 78DTT.jpg
- THD 78T.jpg
- THD 78X.jpg
- THD 86T.jpg
- THD 86X.jpg
- THD 90T.jpg
- THD 96RT.jpg
- THD L247X.jpg
- THD L348X.jpg
- THD L494RX.jpg
- THD L549X.jpg
- TSE 2.MAIN.jpg
- TSE 3.MAIN.jpg
- TYPE-A.tif
- TYPE-JIS.tif
- U1RSM_Spec.pdf
- U1SM_Spec.pdf
- U1XM_Spec.pdf
- VC21499_P04**
- VC21499_P04

- Qualifiers (Qdb)
- Vehicle Configuration
- Configurations
- Index Words
- NAPA Reference Data

VC21499_P04 rev.0.1 - Images & Documents

Images & Documents | References | Referenced By | Status | State Log | Tasks

Description

Name	>	>	Value
ID			AC_0ecd593302877c3b1394f08af1ea8d44
Name			VC21499_P04
Object Type			PIES Product Image
Revision			0.1 Last edited by PIESIMPORT on Tue May 22 08:03:07 EDT 2018
Approved			✘ Never Been Approved
Translation			Not Translated
Path			Classification 1 root/AutoCare Root/PIES Assets/VC21499_P04
Asset ID	abc		Asset456
Representation			Representative
Resolution			96
Background			CLI
Orientation View			Back
Details Description	abc		Details/Description
Language Code			German
File Path	abc		images\SVlight.tif
URI	abc		www.stibo.com
File Date Modified			2008-04-11
Effective Date			2008-04-11
Expiration Date			2012-04-11
Country Code			United States of America (the)
Additional Information	abc		
Asset Dimensions UOM			
Asset Height			123
Asset Width			123

Output

Upon successful import, a PIES Item will be created / modified. The PIES Importer do not delete any PIES Item. The Importer marks the PIES Item as new, delete, or change through change flags.

The <PartTerminologyID> within the <Item> segment is used to locate the Part Terminology in the PCdb classification. If the PartTerminologyID is found in the PCdb classification, then the importer will create the PIES Item under the same structure in the PIES Products Hierarchy for Brand, PCdb Category, PCdb Sub Category, and PCdb Part Terminology.

An example of PIES Item 'VC21499' (ID: AC_PIESItem_DKGX_VC21499) created under Object 'Check Engine Light Sensor' (ID: AC_PIESPCdbPartTerminology_5696_DKGX) with its various attribute values and reference type (as explained above) is shown below.

Tree

- Product Overrides
- Packaging
- AutoCare Root
 - ACES Application Notes
 - PIES Interchange Products
 - PIES Products
 - 034MOTORSPO
 - BCVC
 - Behr Hella Service
 - Behr Thermot-Tronik
 - Exide
 - Hella
 - Pierburg
 - Wako
 - Engine
 - Sensors
 - Check Engine Light Sens
 - JK21499**
 - 2010 Acura TSX
 - 2010 Acura TSX
 - 2010 Acura TSX
 - PK
 - JK36004
 - JK36009
 - JK36112
 - VC21499
 - VC36004
 - VC36009

JK21499 rev.0.2 - Product

Product

Sub Products

References

Referenced By

Images & Documents

Commercial

Tables

🔍 Description

Name		Value
ID	>	AC_PIESItem_DKGX_JK21499
Name	>	JK21499
Object Type	>	PIES Item
Revision	>	0.2 Last edited by ACESIMPORT on Fri Mar 08 07:51:38 EST 2019
Approved	>	✖ Never Been Approved
Translation	>	Not Translated
Path	>	Primary Product Hierarchy/AutoCare Root/PIES Products/Wako/Engine/Sensor
Changed Object	>	abc true
Delete Status	>	abc
New Object	>	abc

- 🔍 AutoCare ACES Attributes
- 🔍 PAdb Attributes
- 🔍 PIES Description Segment DESC
- 🔍 PIES Extended Product Info Segment EXPI
- 🔍 PIES Item Segment ITEM
- 🔍 PIES Price Segment PRCS 2017 WD Prices
- 🔍 PIES Price Segment PRCS HINC 090915
- 🔍 PIES Priceshet 2018WD

In addition, a reference will be established from the PIES Item to the Part Terminology using the 'Product To Part Terminology' reference type. This is done so that the PIES Item will display only the PAdb attributes that are valid for that Part Terminology.

© Stibo Systems - Confidential - PMDM for Automotive 9.2-MP1 (September 27, 2019)

120

Tree

- AutoCare Root
 - ACES Application Notes
 - PIES Interchange Products
 - PIES Products
 - 034MOTORSPORT
 - BCVC
 - Behr Hella Service
 - Behr Thermot-Tronik
 - Exide
 - Hella
 - Pierburg
 - Wako
 - Engine
 - Sensors
 - Check Engine Light Sensor
 - JK21499
 - JK36004
 - JK36009
 - JK36112
 - VC21499**
 - VC36004
 - VC36009
 - VC36112

VC21499 rev.0.5 - References

Images & Documents
Commercial
Tables
Proof View
Status
State Log
Tasks

Product
Data Containers
Sub Products
References

PIES Kit Segment KITS			
Reference Type		Target	ID
> PIES Item To Kit Component Item	+		

PIES Part Interchange Segment INTE			
Reference Type		Target	ID
> PIES Interchange	+	GL26409	AC_PIESInterchan

Ungrouped Classification Links			
Reference Type		Target	>
> Product To Part Terminology		Check Engine Light Sensor	

Ungrouped Product References			
Reference Type		Target	>
> Part Relation	+		
> PIES Item to NAPA Product	+		

Ungrouped Image & Document References			
---------------------------------------	--	--	--

PIES Import Validation Rules

When importing an AutoCare PIES (Product Information Exchange Standard) file many STEP validation rules are performed. The table below describes each of these validation rules, where the validation check occurs, what happens when a validation fails, and an example of a failed validation message (when applicable). However, it does not detail the file type's data rules per the AutoCare Association. This information can be found at www.autocare.org.

For information on the supported versions, see the **Supported Versions and Formats** topic within this guide.

Important: For best results, AutoCare imports should be executed in the order specified within the **AutoCare Import Validation Rules** topic.

Accepted File Extension: .xml

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
PIES 6.5 XSD validation	Validation state	Validation fails	Error while validating file against XSD: cvc-complex-type.2.4.a: Invalid content was found starting with element 'Headeasdr.' One of '{"http://www.aftermarket.org":TestFile, "http://www.aftermarket.org":Header}' is expected.
Part terminology existence	Conversion state	Item will be skipped	Part terminology with id 99999 does not exist in AutoCare reference data. All items under it will be skipped.
Duplicated item	Conversion state	Warning is displayed.	Item with id MyPart already imported.
Duplicated package	Conversion state	Warning is displayed.	Package with PackageLevelGTIN 1234567 was already imported for item MyPart.
Missing GTIN on package	Conversion state	Package skipped	Package with UOM PK does not have a PackageLevelGTIN specified for item MyPart. Package import will be skipped.
PriceSheet existence	Import state	Standard step missing attribute errors.	Attribute 'AC_PIES_PRCS_BADPricesheet_CurrencyCode' not found.

Note: Brand existence is not verified within STEP.

Configuring AutoCare PIES Importer

The following topics provide the configuration steps necessary to allow users to be able to drag and drop PIES files onto a configured File Loading Widget, and monitor the progress of the import file using a configured Status Selector Homepage Widget and an Import Controller Screen.

- Configuring an IIEP for PIES Imports
- Configuring a File Loading Widget for AutoCare PIES Imports

Prerequisites

It is expected that anyone configuring the AutoCare PIES Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, see the **Designer Access** topic within the **Web UI Getting Started** section of **STEP Online Help**.

Within this section users will be directed to view details for a Controller Entity within Web UI using a Node Details screen configured for Entities. Before this can be available to users, an Entity Details screen must be created.

Anyone configuring the AutoCare PIES Importer is expected to be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Configuring an IIEP for AutoCare PIES Imports

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

An Inbound Integration Endpoint (IIEP) can be configured in workbench to help automate the process of importing PIES data into STEP. Once an IIEP is configured for AutoCare PIES imports, PIES files can be imported after they are uploaded either to a configured hotfolder on an application server, or to a File Loading Widget on a Web UI Homepage. For more information, see the **AutoCare PIES Importer** topic.

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

Prerequisites

It is expected that anyone configuring an IIEP for use with an AutoCare PIES Import is familiar with the configuration and other processing of standard inbound integration endpoints. For more information, see the **Inbound Integration Endpoints** topic within the **Data Exchange** section of **STEP Online Help**.

Configuration Steps

1. In workbench, go to System Setup, select and right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.
2. Once the Inbound Integration Endpoint Wizard displays, populate each parameter with values that best identify the IIEP. By default, all parameters display blank, and the following fields are mandatory: Endpoint ID, Endpoint Name, and User.

In the example below, a 'PIES Import' user was created prior to configuring this IIEP. This is recommended to more easily track when this IIEP is responsible for changes to data.

The screenshot shows the 'Inbound Integration Endpoint Wizard' dialog box. On the left, a 'Steps' list includes: 1. Identify Endpoint (highlighted), 2. Choose Receiver, 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area is titled 'Identify Endpoint' and contains four input fields: 'Endpoint ID' with the value 'AutoCarePIESInboundEndpoint', 'Endpoint Name' with 'AutoCare PIES Inbound Endpoint', 'Description' (empty), and 'User' with 'PIES Import (PIESIMPORT)'. At the bottom, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

For more information about the parameters, see the **IIEP - Identify Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the Choose Receiver parameters will display. By default, the parameters are populated as recommended and shown below, except the Hotfolder parameter. This mandatory parameter must be populated with a hotfolder name before the Next button will enable.

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

The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' panel lists eight steps: 1. Identify Endpoint, 2. Choose Receiver (highlighted), 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area is titled 'Choose Receiver' and contains the following fields: 'Receiver' (a dropdown menu set to 'Hotfolder Receiver'), 'Hotfolder' (a text input field containing 'AutoCarePIESInputFolder'), 'Keep file after load' (a dropdown menu set to 'Yes'), 'Ignore sub folders' (a dropdown menu set to 'No'), and 'In folder' (an empty text input field). At the bottom, there are four buttons: 'Back', 'Next' (which is highlighted with a dashed border), 'Finish', and 'Cancel'.

For more information about the parameters, see the **IIEP - Choose Receiver** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the Configure Endpoint parameters will display. By default, the parameters are pre-populated with the recommended values as shown below, except for the following:
 - Processing Engine
 - Maximum number of old processes
 - Maximum age of old processes

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
- 3. Configure Endpoint**
4. Configure PreProcessor
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Endpoint

Processing

Processing Engine: Import Flow Processor

Transactional settings: None

Context

Workspace: Main

Context: Global

Queue Settings

Queue for endpoint: InboundQueue

Queue for endpoint processes: In

Maximum number of waiting processes: 1000

Maximum number of old processes: 1000

Maximum age of old processes: 1y

Number of messages per background process: 1

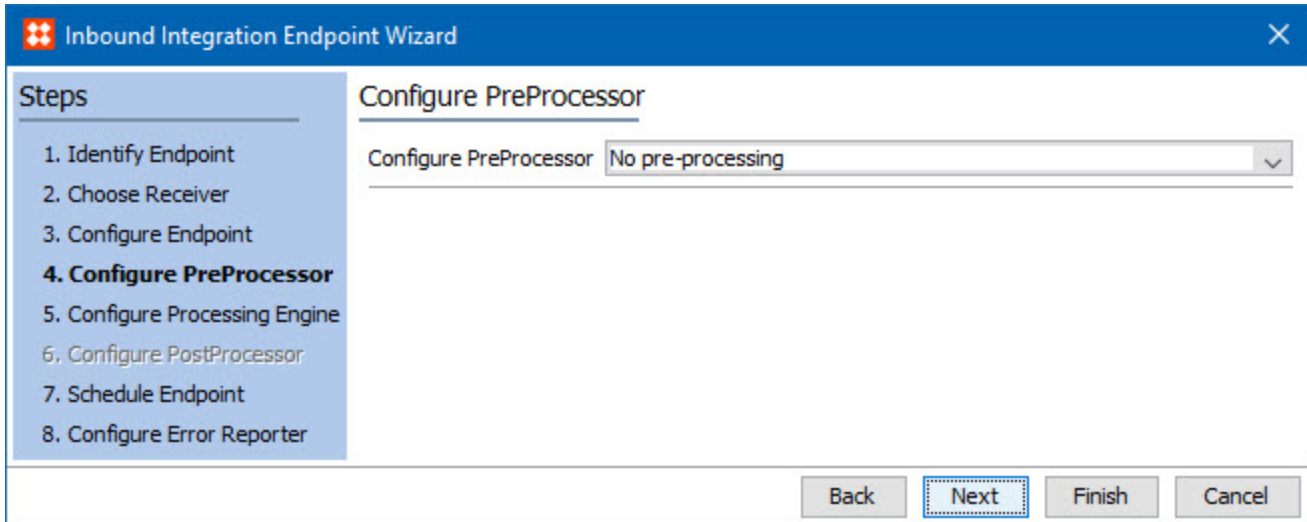
Back Next Finish Cancel

Important: The Import Flow Processor is only responsible for picking up files from the configured hotfolder, creating a Controller Entity object, and updating the Entity with the Import Flow State BGP and Status. The Import Flow Processor works with the Background Process Service for each workflow state to handle the import of a file. The IIEP / Import Flow Processor is NOT responsible for actually importing the content of the file, the BGP in each state does that processing. If a new Processing Engine is created through the Extension API, then it **cannot** be used with the automotive import framework.

- Click the dropdown option for the 'Processing Engine' parameter and select the **Import Flow Processor** option.
- Optionally, update the values for the 'Maximum number of old processes' and 'Maximum age of old processes' parameters to those shown above.

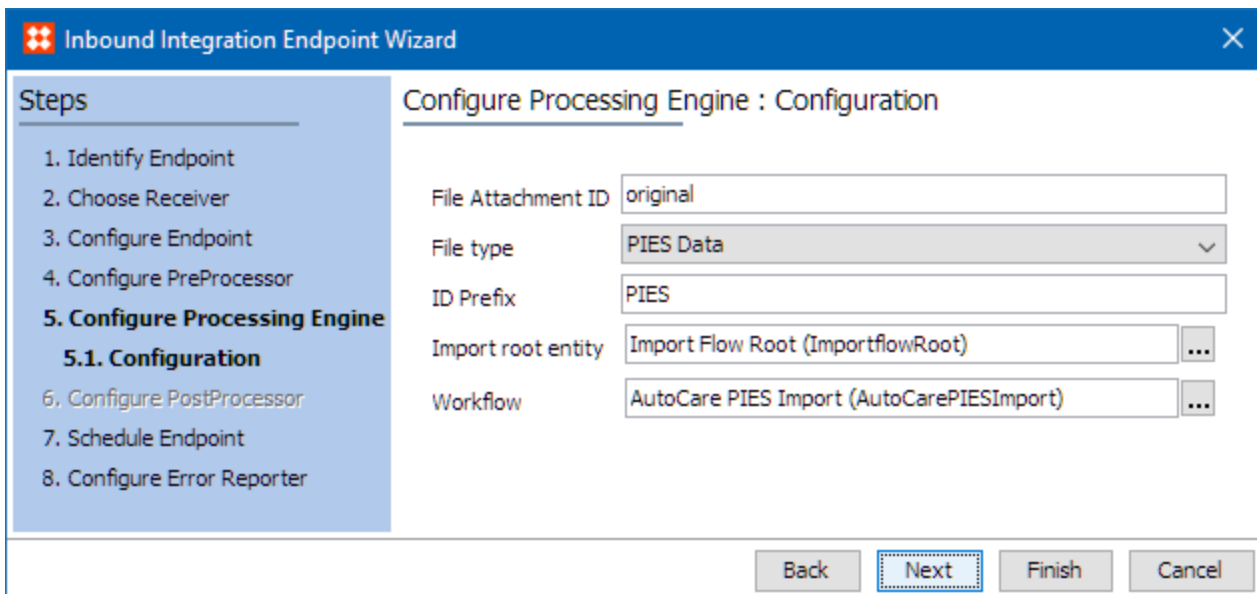
For more information about the parameters, see the **IIEP - Configure Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

5. Click the **Next** button, and the Configure PreProcessor parameter will display. By default, the parameter is populated as recommended and shown below.



For more information about the parameter, see the **IIEP - Configure PreProcessor** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the **Configure Processing Engine : Configuration** parameters for the Import Flow Processor will display. By default, only the File Attachment ID parameter is populated as shown below.



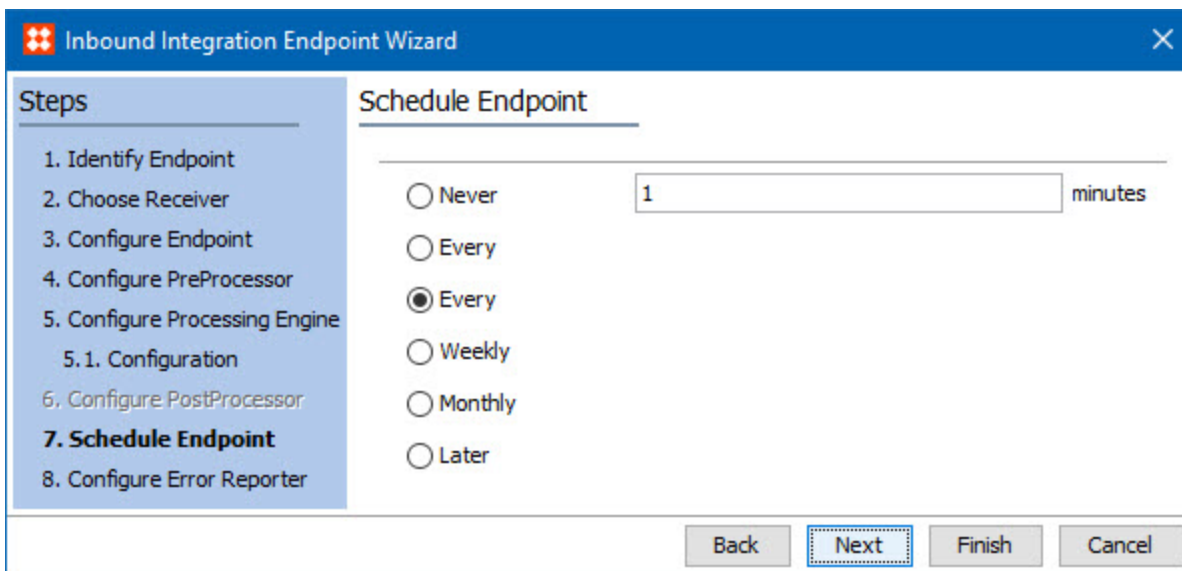
The prior Configure Endpoint step determines the options available for the Configure Processing Engine step.

- Click the 'File type' parameter, and select the **PIES Data** option.
- Within the 'ID Prefix' parameter, enter a prefix value to easily identify import entities created by this IIEP.
- Click the ellipsis button (...) for the 'Import root entity' parameter, and select the 'Import Flow Root' node (or a desired root entity). This is the location where the import entities created by this IIEP will be stored.

- Click the ellipsis button (...) for the 'Workflow' parameter, and select the **AutoCare PIES Import** workflow or a desired workflow.

Note: It is possible to use a workflow that is not created by Easy Setup actions to handle an import file in a way that better fits an organization's needs. However, along with creating the workflow and selecting it within the Workflow parameter shown above, all the states for that workflow must be created along the processing steps of the file (i.e., Validation, Conversion, Import). Use of the Extension API is required to write the processing steps.

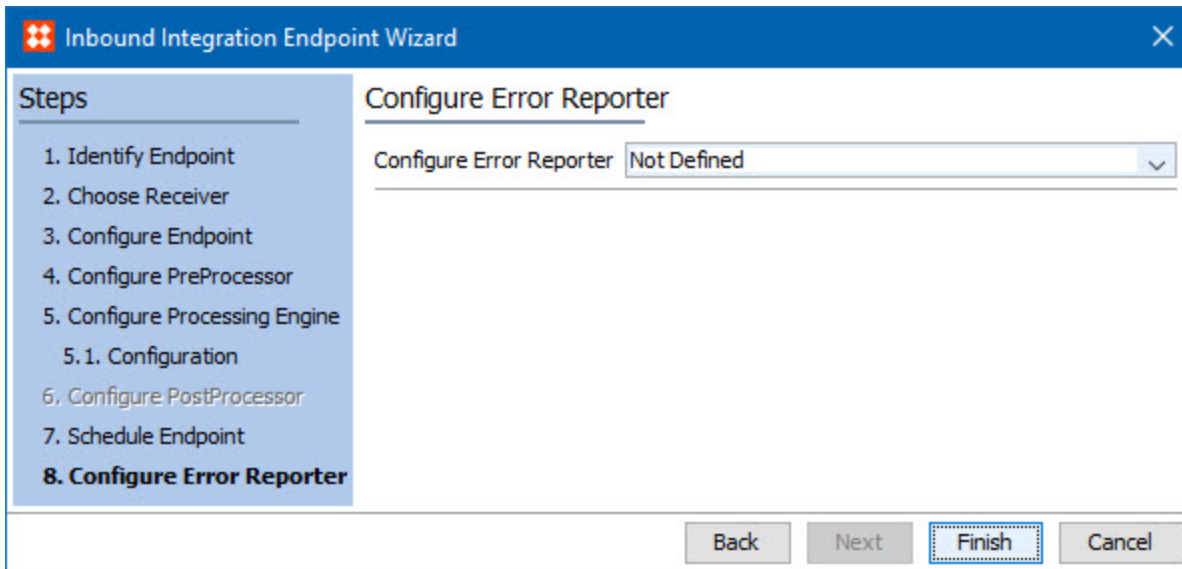
7. Click the **Next** button, and the Schedule Endpoint parameters will display. By default, 'Never' is selected. Optionally, update the values to those shown below.



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.

For more information about the parameters, see the **IIEP - Schedule Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

8. Click the **Next** button, and the Configure Error Reporter parameter will display. By default, the parameter is populated as recommended and shown below.



For more information about the parameter, see the **IIEP - Configure Error Reporter** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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

Important: An endpoint must be enabled before it can start processing data. For more information, see the **Running an Inbound Integration Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

If users need to access the IIEP via a Web UI, then the IIEP must be configured within a File Loading Widget. For more information, see the **Configuring a File Loading Widget for AutoCare PIES Imports** topic.

Configuring a File Loading Widget for AutoCare PIES Imports

When configured, Web UI users can import PIES files into STEP using a File Loading Widget. Users can also monitor the progress of imports using a Status Selector Homepage Widget and a Node Details component.

Prerequisites

Before starting to configure the Web UI portion of this solution, an IIEP for an AutoCare PIES Importer must be configured within workbench. For more information, see the **Configuring an IIEP for AutoCare PIES Imports** topic.

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

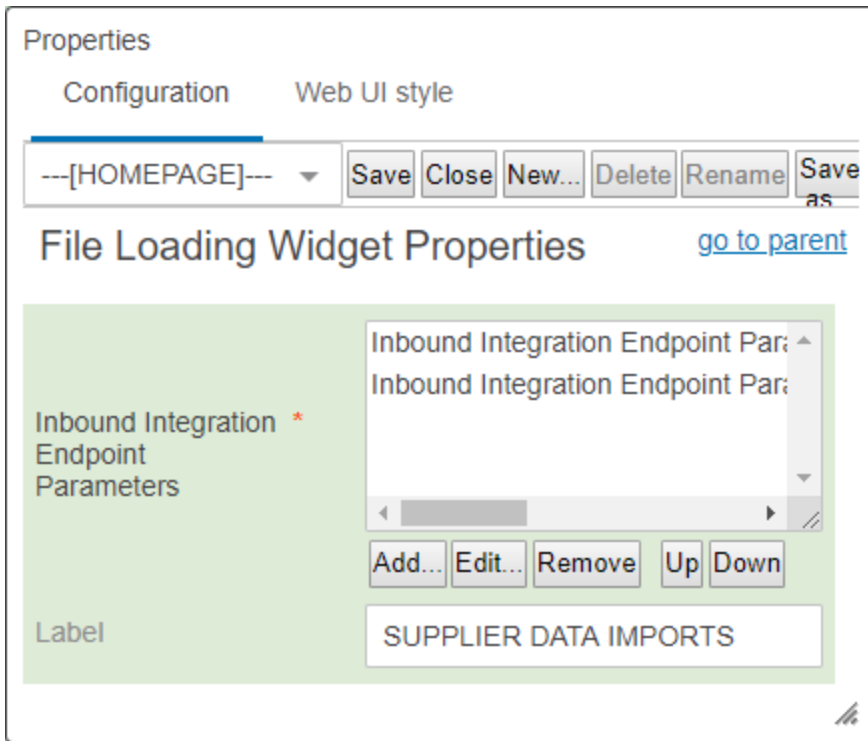
Configuration

Each screenshot example within this section provides recommended values for the parameters and AutoCare PIES Importer.

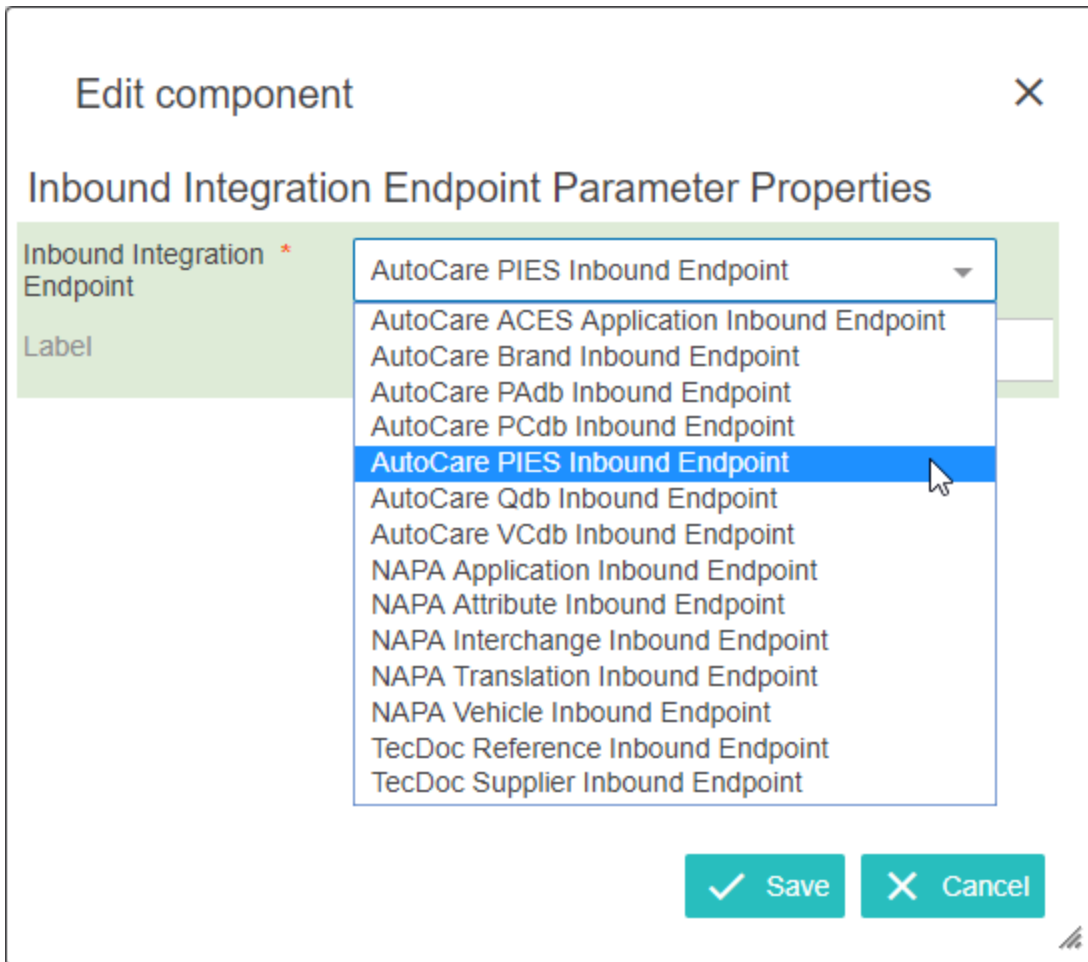
This topic describes how to configure a File Loading Widget so that users can drag and drop AutoCare PIES files onto a File Loading Widget on a Web UI Homepage.

Note: If Easy Setup actions for the AutoCare solution have been completed as described in the **3. Run Easy Setup of Standards** topic of **Automotive Quick Start Guide**, then the 'SUPPLIER DATA IMPORTS' File Loading Widget will automatically be added to the AutoCare Web UI Homepage as shown in the examples below. Otherwise, the steps below can be used to complete configuration.

1. In the designer, select an existing File Loading Widget to be used, or add a new File Loading Widget to the Homepage Widget Grid component. For more information, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.
2. Go to the Inbound Integration Endpoint Parameters field, click the **Add** button, and the Inbound Integration Endpoint Parameter Properties dialog will display.



3. Click the dropdown for the Inbound Integration Endpoint parameter, and select **AutoCare PIES Inbound Endpoint** (the IIEP created for AutoCare PIES imports).

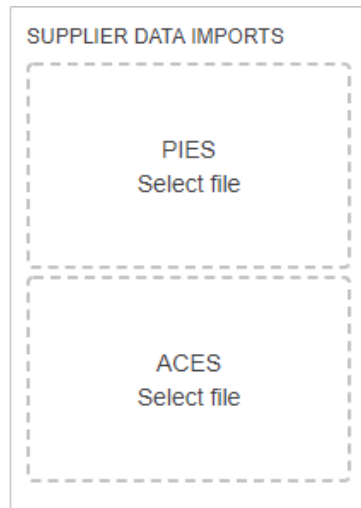


Note: If the desired IIEP does not display in the dropdown, then it can be created using the steps described in the **Configuring an IIEP for PIES Imports** topic.

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

In the example below:

- A File Loading Widget labeled as 'SUPPLIER DATA IMPORTS' is displayed above its configurations.
- The File Loading Widget and its configurations are shown with the default configurations provided automatically when Easy Setup actions for the AutoCare component are completed.
- An IIEP for AutoCare ACES imports is added within the same File Loading Widget as the IIEP for the AutoCare PIES imports.



File Loading Widget Properties [go to parent](#)

Inbound Integration *
Endpoint Parameters

- Inbound Integration Endpoint Parameter (PIES / AutoCarePIESInboundEndpoint)
- Inbound Integration Endpoint Parameter (ACES / AutoCareACESApplicationInbour

Add... Edit... Remove Up Down

Label: SUPPLIER DATA IMPORTS

Child Components

Edit component [Close]

Inbound Integration Endpoint Parameter Properties

Inbound Integration * Endpoint: AutoCare PIES Inbound Endpoint

Label: PIES

[Save] [Cancel]

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

AutoCare PAdb Importer

The intention of the Automotive AutoCare PAdb Importer is to provide an out-of-the-box solution for importing data included within a supported PAdb file format. Because only supported versions will successfully upload, before attempting to upload an AutoCare PAdb file, confirm the file version being uploaded is listed within the **Supported Versions and Formats** topic.

This section includes information on:

- Using AutoCare PAdb Importer
- Configuring AutoCare PAdb Importer

Prerequisites

It is expected that anyone using and/or configuring the AutoCare PAdb Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Using AutoCare PAdb Importer

PAdb file can be imported into STEP by uploading it to either a configured hotfolder on an application server, or through a File Loading Widget on a Web UI Homepage.

The following topics provide information on using the AutoCare PAdb Importer.

- Importing PAdb Files via Web UI
- PAdb Import Validation Rules

Prerequisites

It is expected that anyone using and/or configuring the PAdb Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For more information about uploading files to an application server, see the **File Loading** section of the **Automotive Quick Start Guide** found within the **Solution Enablement** section of **STEP Online Help**.

For more information about using Workbench to monitor and process imports, see the **Workflows** section of **STEP Online Help**.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Importing PAdb Files via Web UI

The intention of the AutoCare PAdb Importer Web UI setup included within this topic is to provide an out-of-the-box solution for importing data included within a supported AutoCare PAdb file format. Because only supported versions will successfully upload, before attempting to upload a PAdb file, confirm the file version being uploaded is listed within the **Supported Versions and Formats** topic.

A PAdb file defines the way trading partners exchange information concerning the product-specific performance and physical attributes of parts.

Important: It is always required that the latest PAdb file is imported into the system. Importing the PAdb file validates the version date against the version date stored on the 'AC_PAdb_VersionDate' attribute which is valid on the PCdb root classification object, and displays an error in the Validation state if the imported file is older than the version date that exists in STEP.

Prerequisites

Within this section users will be directed to view details for a Controller Entity within Web UI using a Node Details screen configured for Entities. Before this can be available to users, an Entity Details screen must be created.

Reference data files (Qdb, Brand Table, PCdb, and VCdb) must be imported before PAdb files are imported.

To easily start and monitor a PAdb Import, recommended practice is to use a Web UI Import Controller screen specific to the AutoCare PAdb Importer. When the Easy Setup actions for the AutoCare standard are completed, the AutoCare Import Controller screen is automatically created and configured for use. Additional information can be found in the **Configuring AutoCare PAdb Importer** topic.

It is expected that anyone using and/or configuring the AutoCare PAdb Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

If the Easy Setup actions for the AutoCare Component model have been completed, then the functionality explained within this topic should be available. Otherwise, configuration is necessary. For more information, see the **AutoCare PAdb Importer** topic.

Import Process Overview

Once a valid automotive data file is uploaded to a hotfolder on the application server (optionally using a File Loading Widget), the file is picked up from the hotfolder by an IIEP, and the IIEP creates an Entity in STEP that represents the file. This Entity object is called the 'controller' and contains basic data about the file and the file's status in the workflow. Web UI users are able to monitor the import status using a Status Selector Widget and an Import Controller Screen. Once an import file reaches the 'Ready for Import' state, then users can start the import by clicking on the 'Start import' button within the Control Panel screen. At that time, the BGP service (that runs as part of the Import state) allows for the configured business rules to act on the objects being imported.

Using the AutoCare PAdb Importer in Web UI

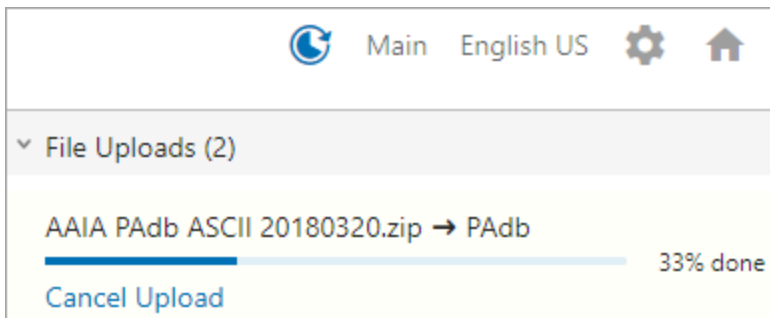
1. Access the AutoCare Web UI Homepage.

2. Upload a valid PAdb file to the application server hotfolder (root/upload/hotfolders/AutoCarePAdbInputFolder), or use the 'PAdb' drop zone of the 'REFERENCE DATA IMPORTS' File Loading Widget.

For more information, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

3. Once the upload has started, users can view the progress of the upload using the Background Process Notification Component.

In the example below, the BGP Notification Panel is expanded and the file, 'AAIA PAdb ASCII 20180320.zip' is 33% processed.



For more information on using the BGP Notification Component and side panel, see the **Background Process Notification Component** topic within the **Main Properties Overview** section of **STEP Online Help**.

Once the file has been uploaded and picked up by the IIEP, the Controller is initiated into the workflow associated with the importer (AutoCare PAdb Import). From there, the AutoCare PAdb Import workflow takes over processing of the file via a series of states using business rules and background processes to carry out the processing of the file. Each import has an associated workflow and all proceed through the same states by default. However, it is intended that customers will expand on the existing states and actions to add their own validations, approve objects, reporting, and additional processing as needed.

Important: It is critical to understand that it is only the Controller Entity that is in the workflow - the objects being acted on (created / updated / deleted) via information supplied in the import file are not in the workflow. Therefore, running standard business actions acting on current object will impact the Controller Entity only, not the objects in the input file. In order for the Business Action to act on the objects being imported, the Business Action must be added to the 'Import action' parameter on the 'Import' state. For more information, see the **Import State** topic within the **Default Workflow States and Functions** section.

4. If an Entity Details screen has been configured, then when the IIEP creates the Controller Entity object, users can view the Controller and its important information by navigating within the Tree side panel to the Import Flow Root > AutoCare PAdb. Otherwise, if a user clicks on an Entity in the Tree side panel, an error will display indicating the configuration is invalid.

Note: If an error message displays when selecting an Entity within a Web UI, then the Entity Details screen has not been configured.

In the example below, the Tree side panel is expanded, and the 'AAIA PAdb ASCII 20190329.zip' Controller with ID 'Controller- 106327' is displayed below the 'AutoCare PAdb' Entity. When the Controller is selected, many details can be viewed within the Node Details screen.

Note: In the screenshot below, the 'Enable Tag Conversion' parameter within the Attribute Value Group Component is enabled.

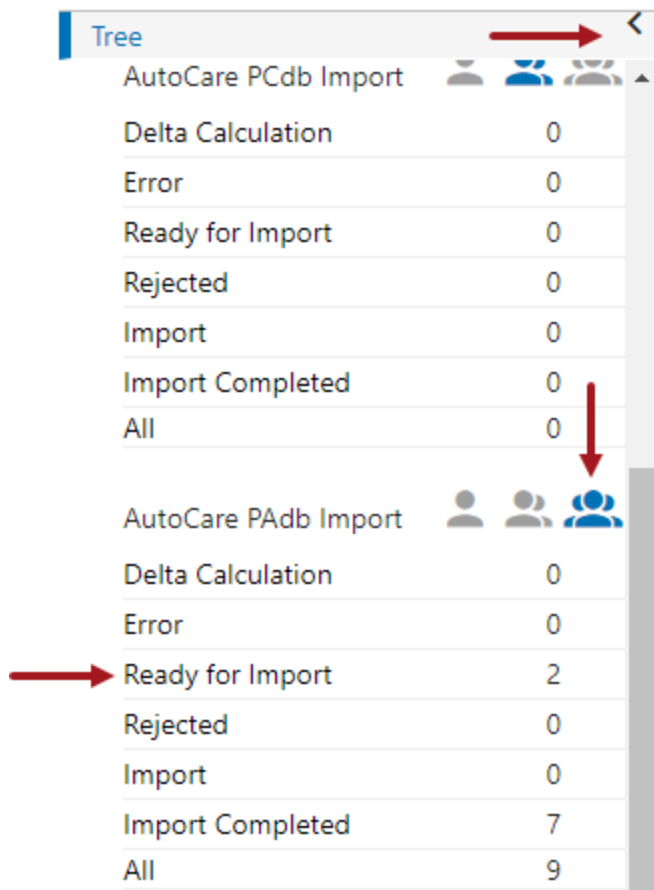
Tree	Node Details									
<ul style="list-style-type: none"> ▶ AutoCare Root ▶ Primary Product Hierarchy ▶ TecDoc Supplier Price Lists ▼ Import Flow Root <ul style="list-style-type: none"> ▶ AutoCare ACES ▶ AutoCare Brand ▼ AutoCare PAdb <ul style="list-style-type: none"> AAIA PAdb ASCII 20180320.zip AAIA PAdb ASCII 20180320.zip AAIA PAdb ASCII 20180320.zip AAIA PAdb ASCII 20190329.zip AAIA PAdb ASCII 20190329.zip AAIA PAdb ASCII 20190329.zip <li style="background-color: #0070c0; color: white;">AAIA PAdb ASCII 20190329.zip C:\Users\amna\Desktop\AAIA . C:\Users\amna\Desktop\AAIA . C:\Users\amna\Desktop\New f. PAdb_RJAG.zip ▶ AutoCare PCdb ▶ AutoCare PIES ▶ AutoCare Qdb ▶ AutoCare VCdb ▶ NAPA Application ▶ NAPA Attribute ▶ NAPA Translation ▶ NAPA Vehicle ▶ TecDoc Reference ▶ TecDoc Supplier 	<p>Name: AAIA PAdb ASCII 20190329.zip</p> <p>ID: Controller-106327</p> <p>Object Type: Import Flow Controller Type</p> <p>Revision: <table style="width: 100%; border: none;"> <tr><td>Date:</td><td>2019.04.24</td></tr> <tr><td>Revision number:</td><td>0.1</td></tr> <tr><td>Last Edited by:</td><td>PADBIMPORT</td></tr> <tr><td>Time:</td><td>03:33:46</td></tr> </table> </p> <p>Automotive Import Flow State BGP: <pre style="border: 1px solid #ccc; padding: 5px; font-family: monospace; font-size: 0.9em;"> <?xml version="1.0" encoding="UTF-8"?> <StringMap> <Entry Key="Validation" Value="BGP_106328"/> <Entry Key="Conversion" Value="BGP_106329"/> <Entry Key="DeltaCalculate" Value="BGP_106330"/> </StringMap> </pre> </p> <p>Import Flow Endpoint ID: AutoCarePAdbInboundEndpoint</p> <p>Import Flow File Type: PAdbData</p> <p>Import Flow Overall Status: Done creating delta file</p> <p>Import Flow State Status: <pre style="border: 1px solid #ccc; padding: 5px; font-family: monospace; font-size: 0.9em;"> <?xml version="1.0" encoding="UTF-8"?> <StringMap> <Entry Key="Validation" Value="Validation completed"/> <Entry Key="Conversion" Value="Conversion completed"/> <Entry Key="DeltaCalculate" Value="Delta calculation completed"/> </StringMap> </pre> </p> <p>Import Flow Workflow ID: AutoCarePAdbImport</p>	Date:	2019.04.24	Revision number:	0.1	Last Edited by:	PADBIMPORT	Time:	03:33:46	
Date:	2019.04.24									
Revision number:	0.1									
Last Edited by:	PADBIMPORT									
Time:	03:33:46									

- **Name:** STEP Name of the Controller Entity. This is generated from the original name of the uploaded file.
- **Automotive Import Flow State BGP:** Attribute used to store the IDs of the background processes.

- **Import Flow State Status:** Attribute used to store and display the status of each process (rather than the overall status).
 - **Import Flow Overall Status:** Attribute used to store and display the global status of the file (rather than the process specific status).
5. As the file is uploaded, and the Controller Entity moves through the AutoCare PAdb Import workflow, users can monitor the progress using the 'AUTOCARE PADB IMPORT' Status Selector Widget on the Web UI homepage and/or the left side panel.

In the example below, the Tree panel is expanded, and the **Triple user button** is selected so that all items assigned to any user are displayed. Notice that six file is in the 'Ready for Import' state, and nineteen files in the 'Import Completed' state, for a total of twenty five files.

For more information, see the **Moving Tasks Trough a Workflow in Web UI** topic within **STEP Online Help**.



6. Clicking on the **Ready for Import** state within the Status Selector homepage and/or the left side panel widget will navigate the user to the AutoCare PAdb Import Controller Screen with only those Controllers in the 'Ready for Import' state displayed.
7. Clicking the Controller icon displays the Import Details below the Controller list.

In the example below, Controller 'AAIA PAdb ASCII 20190329.zip' (ID 'Controller-106327') with an overall status of 'Done creating delta file' is selected and the Import Details are displayed below the Controller list.

Tree

Ready for Import	0
Rejected	0
Import	0
Import Completed	0
All	0

AutoCare PAdb Import

Delta Calculation	0
Error	0
Ready for Import	1
Rejected	0
Import	0
Import Completed	7
All	8

AutoCare Brand Import

Delta Calculation	0
Error	0
Import Completed	0
All	0

Advanced search

AutoCare PAdb Imports

Process	File Name	Overall Status
Controller-106327	AAIA PAdb ASCII 20190329.zip	Done creating delta file

1-1 of 1

Import Details

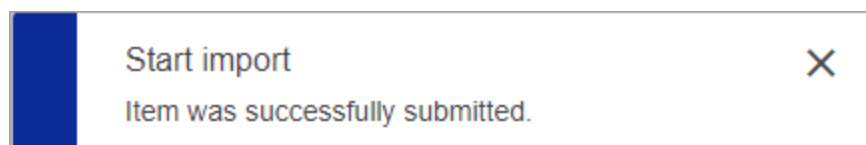
Process	Started Time	Duration	Started By	Status	Background
Validation	2019-04-24 03:33:46	6 secs	STEPSYS	Validation completed	succeeded
Conversion	2019-04-24 03:33:52	1 min 46 secs	STEPSYS	Conversion completed	succeeded
Delta Calculation	2019-04-24 03:35:38	32 secs	STEPSYS	Delta calculation completed	succeeded

Import

1-4 of 4

Start import Reject Discard file

8. Optionally, before starting the import, clicking on the 'completedwitherrors' Background Process Link (only displayed when there is any error) will display the Background Process Details screen where users can view or download the details of the validation process.
9. Click the **Reject** button to move the Controller to the Rejected state. For more information, see the **Rejected State** topic within the **Import Framework** section.
10. Click the **Start import** button, the Start import dialog (shown below) displays at the top of the screen, and the Controller is moved to the Import state of the workflow. For more information, see the **Import State** topic within the **Import Framework** section.



While importing a PAdb file the import progress can be monitored using the 'AUTOCARE PAdb IMPORT' Status Selector Widget on the Web UI homepage and/or the left side panel or the Import Controller Screen, Import process Status column.

In the example below, the AutoCare PAdb Import Controller Screen displays the Overall Status and the Import process status as 'Importing files...'. Additionally, the Background Process Link column displays a link to the Background Process Details screen as 'running.'

AutoCare PAdb Imports					
Process		File Name		Overall Status	
No existing records					
<div style="text-align: center;"> << < 0 > >> </div>					
Import Details					
Process	Started Time	Duration	Started By	Status	Background
Validation	2019-04-24 03:33:46	6 secs	STEPSYS	Validation completed	succeeded
Conversion	2019-04-24 03:33:52	1 min 46 secs	STEPSYS	Conversion completed	succeeded
Delta Calculation	2019-04-24 03:35:38	32 secs	STEPSYS	Delta calculation completed	succeeded
Import	2019-04-24 04:20:16	7 secs	STEPSYS	Importing files...	running 11
<div style="text-align: center;"> << < 1-4 of 4 > >> </div>					
➔ Start import ➔ Reject ➔ Discard file					

- Optionally, click the 'running' Background Process Link (shown above) to view the details of the import process within the Background Process Details screen.

Once the import has completed, the Controller is automatically moved to the Import Completed state. For more information, see the **Import Completed State** topic within the **Import Framework** section.

For more information about PAdb import validation rules and example error messages, see the **PAdb Import Validation Rules** topic.

For more information on automotive workflow states, see the **Default Workflow States and Functions** topic.

PAdb Import Validation Rules

When importing an AutoCare PAdb (Product Attribute Database) file many STEP validation rules are performed. The table below describes each of these validation rules, where the validation check occurs, what happens when a validation fails, and an example of a failed validation message (when applicable). However, it does not detail the file type's data rules per the AutoCare Association. This information can be found at www.autocare.org.

For information on the supported versions, see the **Supported Versions and Formats** topic within this guide.

Important: For best results, AutoCare imports should be executed in the order specified within the **AutoCare Import Validation Rules** topic.

Accepted File Extension: .zip

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
Cannot be unzipped or is not in a recognized format.	Validation state	BGP ends in an error, triggering the import to enter the Error state.	File is not in a recognized format or could not be unzipped.
Unit group existence.	Import state	Unit skipped.	The unit 'DK' could not be created as the unit group is unknown.
All tables / subfiles present. (PartAttributes.txt, MetaUoMCodes.txt, Version.txt, PartAttributeAssignment.txt, Metadata.txt)	Validation state	BGP ends in an error, triggering the import to enter the Error state.	The required table '[Required Table Name]' is missing in the [Name of zipped file] ZIP archive or is empty.
Mandatory values present	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table '[Table Name]' contains a row that misses value for the required column '[Column Name]' at line 3.
Correct number of columns / headers (If last column is optional, then the last column can be missing without affecting validation.)	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table '[Table Name]' contains a row with wrong number of columns at line 2. or [Name of subfile] does not include [missing Header].
Loading of older version	Validation state	Warning is displayed.	Existing version '2015-09-25' is newer than provided '2014-11-28.'
Empty line	Validation state	Warning is displayed.	Table 'VehicleToBedConfig' contains an empty row at line 3.
Part Type existence	Import state	All attribute links to PT skipped.	Line 10, Classification 2: No parent specified for new classification with ID 'AC_PartTerminology_99999' Line 10, Classification 2: The classification with ID 'AC_PartTerminology_99999' was skipped.

Configuring AutoCare PAdb Importer

The following topics provide the configuration steps necessary to allow users to be able to drag and drop PAdb files onto a configured File Loading Widget, and monitor the progress of the import file using a configured Status Selector Homepage Widget and an Import Controller Screen.

- Configuring an IIEP for PAdb Imports
- Configuring a File Loading Widget for AutoCare PAdb Imports

Prerequisites

It is expected that anyone configuring the AutoCare PAdb Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, see the **Designer Access** topic within the **Web UI Getting Started** section of **STEP Online Help**.

Within this section users will be directed to view details for a Controller Entity within Web UI using a Node Details screen configured for Entities. Before this can be available to users, an Entity Details screen must be created.

Anyone configuring the AutoCare PAdb Importer is expected to be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Configuring an IIEP for AutoCare PAdb Imports

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

An Inbound Integration Endpoint (IIEP) can be configured in workbench to help automate the process of importing PAdb data into STEP. Once an IIEP is configured for AutoCare PAdb imports, PAdb files can be imported after they are uploaded either to a configured hotfolder on an application server, or to a File Loading Widget on a Web UI Homepage. For more information, see the **AutoCare PAdb Importer** topic.

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

Prerequisites

It is expected that anyone configuring an IIEP for use with an AutoCare PAdb Import is familiar with the configuration and other processing of standard inbound integration endpoints. For more information, see the **Inbound Integration Endpoints** topic within the **Data Exchange** section of **STEP Online Help**.

Configuration Steps

1. In workbench, go to System Setup, select and right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.
2. Once the Inbound Integration Endpoint Wizard displays, populate each parameter with values that best identify the IIEP. By default, all parameters display blank, and the following fields are mandatory: Endpoint ID, Endpoint Name, and User.

In the example below, a 'PAdb Import' user was created prior to configuring this IIEP. This is recommended to more easily track when this IIEP is responsible for changes to data.

The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' list includes: 1. Identify Endpoint (highlighted), 2. Choose Receiver, 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area is titled 'Identify Endpoint' and contains the following fields:

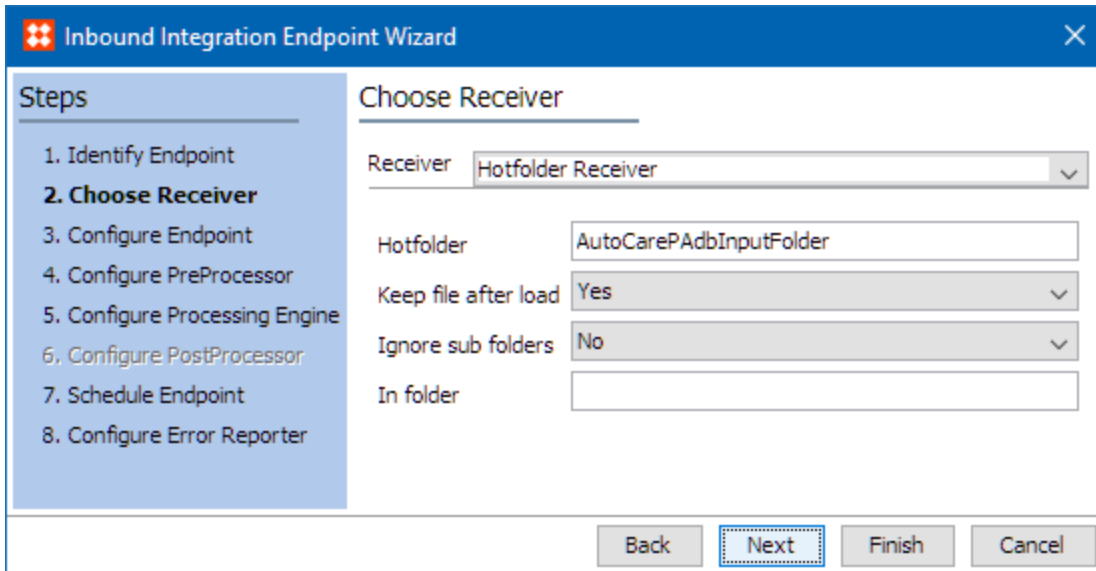
- Endpoint ID:
- Endpoint Name:
- Description:
- User:

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

For more information about the parameters, see the **IIEP - Identify Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the Choose Receiver parameters will display. By default, the parameters are populated as recommended and shown below, except the Hotfolder parameter. This mandatory parameter must be populated with a hotfolder name before the Next button will enable.

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



For more information about the parameters, see the **IIEP - Choose Receiver** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the Configure Endpoint parameters will display. By default, the parameters are pre-populated with the recommended values as shown below, except for the following:
 - Processing Engine
 - Maximum number of old processes
 - Maximum age of old processes

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
- 3. Configure Endpoint**
4. Configure PreProcessor
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Endpoint

Processing

Processing Engine: Import Flow Processor

Transactional settings: None

Context

Workspace: Main

Context: Global

Queue Settings

Queue for endpoint: InboundQueue

Queue for endpoint processes: In

Maximum number of waiting processes: 1000

Maximum number of old processes: 1000

Maximum age of old processes: 1y

Number of messages per background process: 1

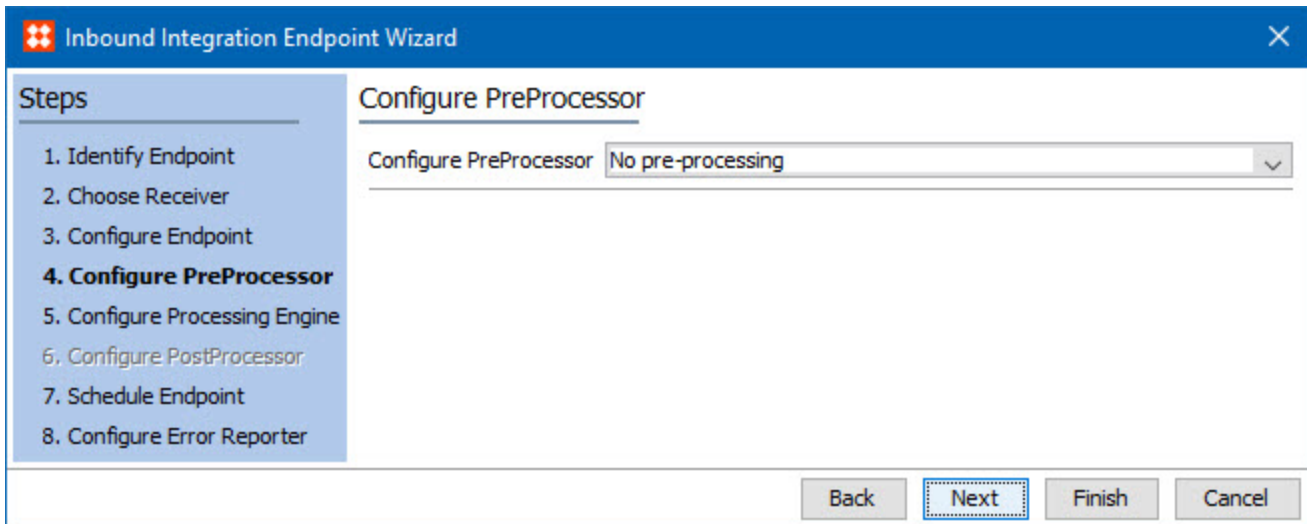
Buttons: Back, Next, Finish, Cancel

Important: The Import Flow Processor is only responsible for picking up files from the configured hotfolder, creating a Controller Entity object, and updating the Entity with the Import Flow State BGP and Status. The Import Flow Processor works with the Background Process Service for each workflow state to handle the import of a file. The IIEP / Import Flow Processor is NOT responsible for actually importing the content of the file, the BGP in each state does that processing. If a new Processing Engine is created through the Extension API, then it **cannot** be used with the automotive import framework.

- Click the dropdown option for the 'Processing Engine' parameter and select the **Import Flow Processor** option.
- Optionally, update the values for the 'Maximum number of old processes' and 'Maximum age of old processes' parameters to those shown above.

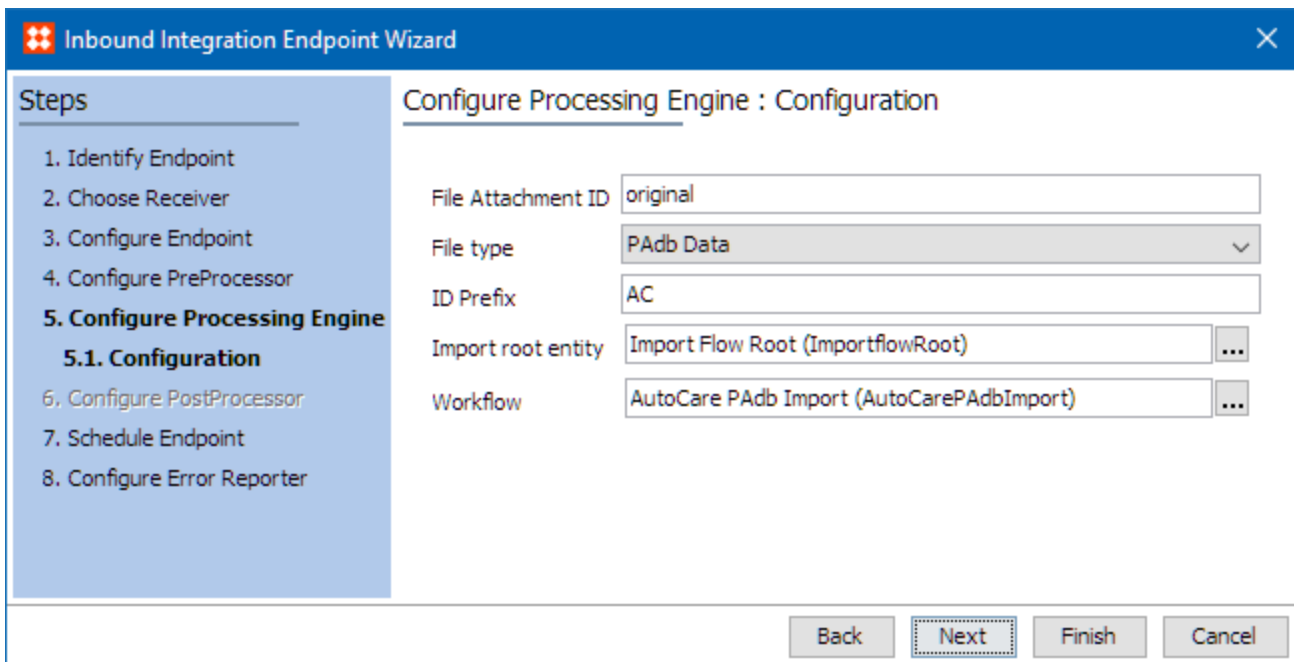
For more information about the parameters, see the **IIEP - Configure Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

5. Click the **Next** button, and the Configure PreProcessor parameter will display. By default, the parameter is populated as recommended and shown below.



For more information about the parameter, see the **IIEP - Configure PreProcessor** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the **Configure Processing Engine : Configuration** parameters for the Import Flow Processor will display. By default, only the File Attachment ID parameter is populated as shown below.



The prior Configure Endpoint step determines the options available for the Configure Processing Engine step.

- Click the 'File type' parameter, and select the **PAdb Data** option.
- Within the 'ID Prefix' parameter, enter a prefix value to easily identify import entities created by this IIEP.

- Click the ellipsis button (...) for the 'Import root entity' parameter, and select the 'Import Flow Root' node (or a desired root entity). This is the location where the import entities created by this IIEP will be stored.
- Click the ellipsis button (...) for the 'Workflow' parameter, and select the **AutoCare PAdb Import** workflow or a desired workflow.

Note: It is possible to use a workflow that is not created by Easy Setup actions to handle an import file in a way that better fits an organization's needs. However, along with creating the workflow and selecting it within the Workflow parameter shown above, all the states for that workflow must be created along the processing steps of the file (i.e., Validation, Conversion, Import). Use of the Extension API is required to write the processing steps.

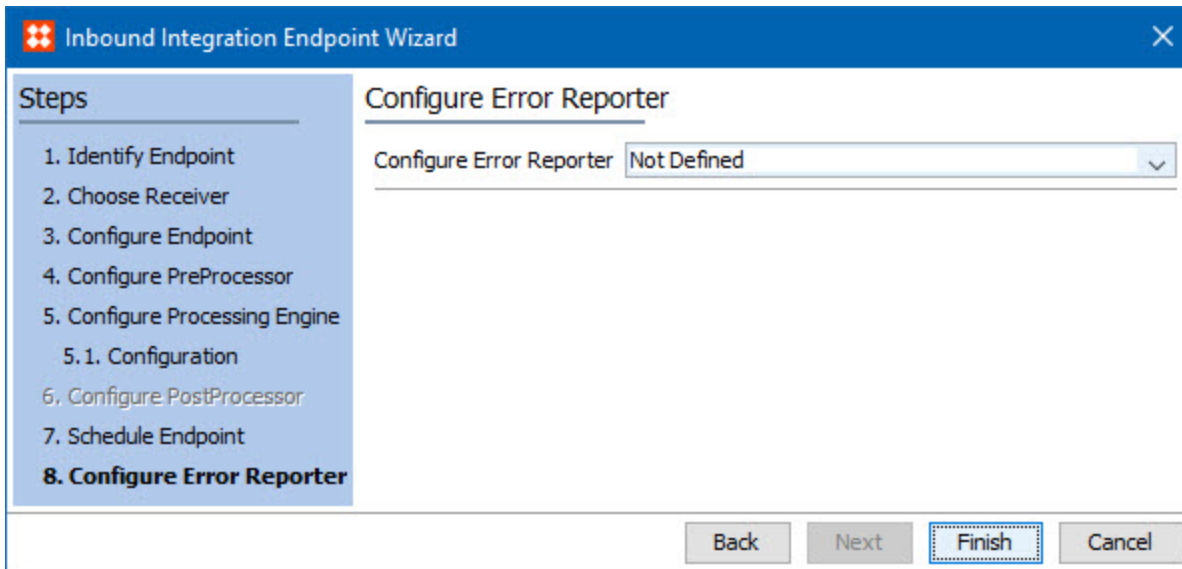
7. Click the **Next** button, and the Schedule Endpoint parameters will display. By default, 'Never' is selected. Optionally, update the values to those shown below.

The screenshot shows the 'Inbound Integration Endpoint Wizard' dialog box. On the left, a 'Steps' list includes: 1. Identify Endpoint, 2. Choose Receiver, 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine (with sub-step 5.1. Configuration), 6. Configure PostProcessor, 7. **Schedule Endpoint**, and 8. Configure Error Reporter. The 'Schedule Endpoint' section on the right contains radio buttons for 'Never', 'Every', 'Weekly', 'Monthly', and 'Later'. The 'Every' radio button is selected. To its right is a text input field containing the number '1', followed by the label 'minutes'. At the bottom of the dialog are four buttons: 'Back', 'Next' (highlighted with a dashed border), 'Finish', and 'Cancel'.

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.

For more information about the parameters, see the **IIEP - Schedule Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

8. Click the **Next** button, and the Configure Error Reporter parameter will display. By default, the parameter is populated as recommended and shown below.



For more information about the parameter, see the **IIEP - Configure Error Reporter** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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

Important: An endpoint must be enabled before it can start processing data. For more information, see the **Running an Inbound Integration Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

If users need to access the IIEP via a Web UI, then the IIEP must be configured within a File Loading Widget. For more information, see the **Configuring a File Loading Widget for AutoCare PAdb Imports** topic.

Configuring a File Loading Widget for AutoCare PAdb Imports

When configured, Web UI users can import PAdb file into STEP using a File Loading Widget. Users can also monitor the progress of imports using a Status Selector Homepage Widget and a Node Details component.

Prerequisites

Before starting to configure the Web UI portion of this solution, an IIEP for an AutoCare PAdb Importer must be configured within workbench. For more information, see the **Configuring an IIEP for PAdb Imports** topic.

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

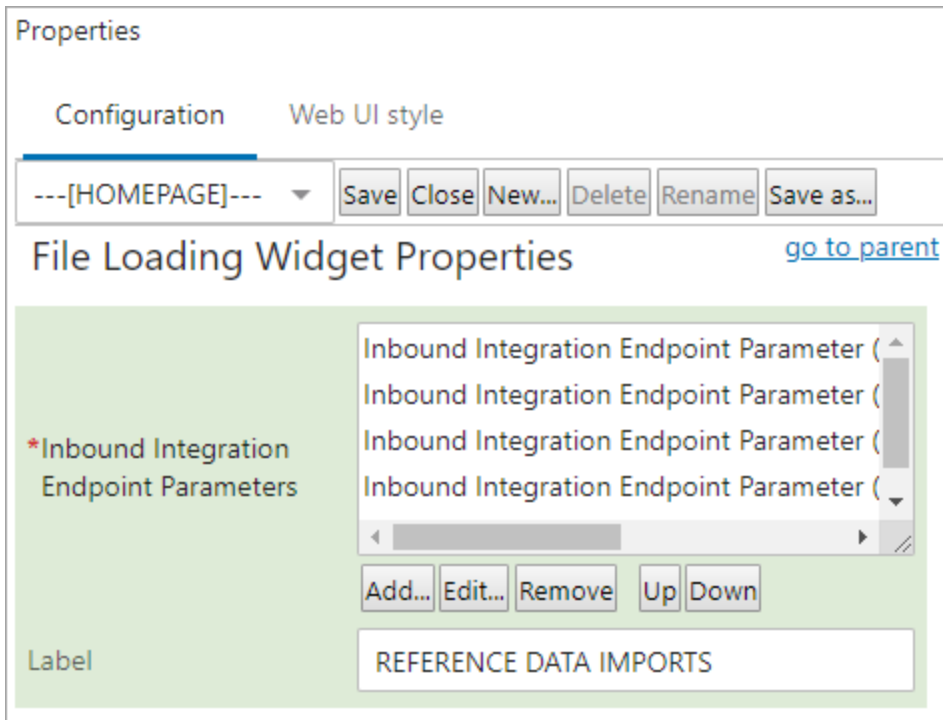
Configuration

Each screenshot example within this section provides recommended values for the parameters and AutoCare PAdb Importer.

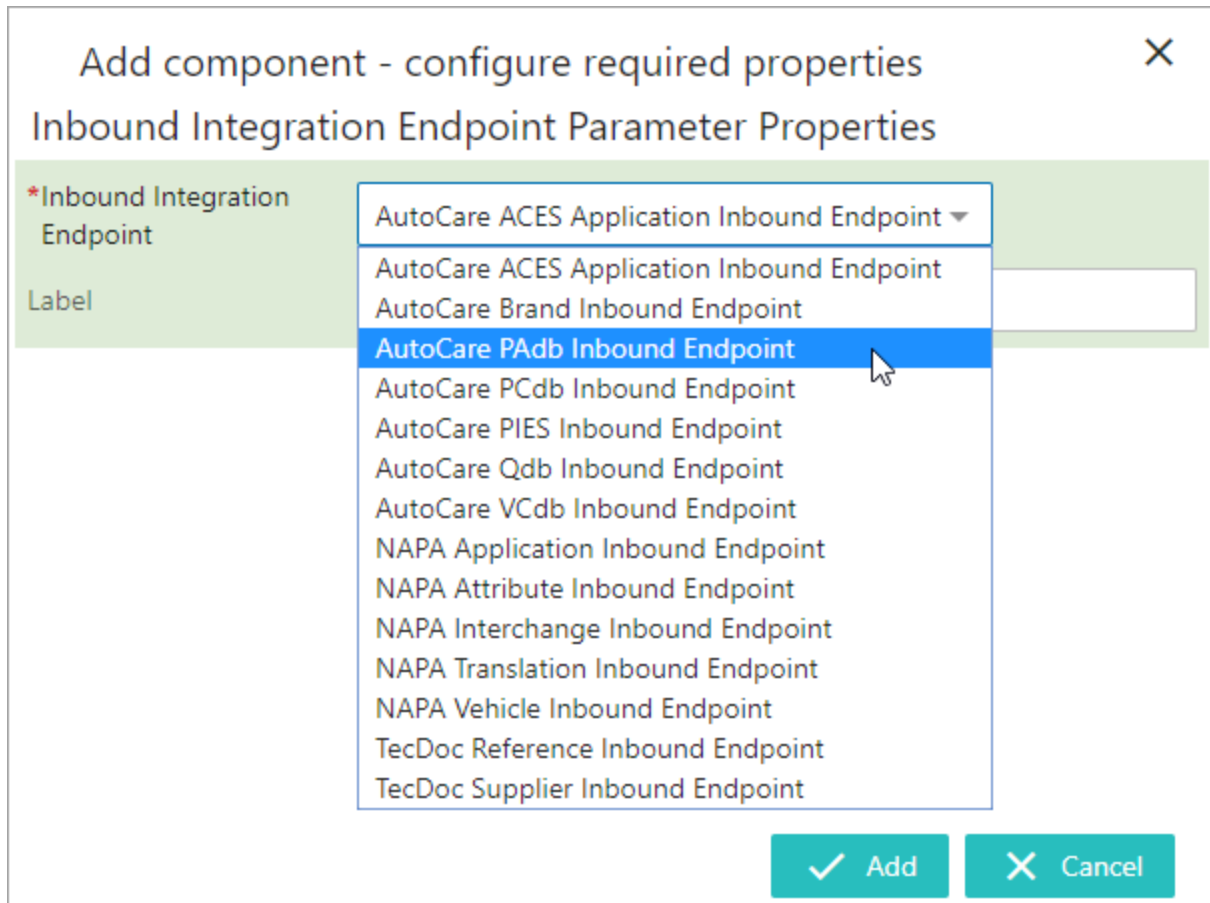
This topic describes how to configure a File Loading Widget so that users can drag and drop AutoCare PAdb files onto a File Loading Widget on a Web UI Homepage.

Note: If Easy Setup actions for the AutoCare solution have been completed as described in the **3. Run Easy Setup of Standards** topic of **Automotive Quick Start Guide**, then the 'REFERENCE DATA IMPORTS' File Loading Widget will automatically be added to the AutoCare Web UI Homepage as shown in the examples below. Otherwise, the steps below can be used to complete configuration.

1. In the designer, select an existing File Loading Widget to be used, or add a new File Loading Widget to the Homepage Widget Grid component. For more information, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.
2. Go to the Inbound Integration Endpoint Parameters field, click the **Add** button, and the Inbound Integration Endpoint Parameter Properties dialog will display.



3. Click the dropdown for the Inbound Integration Endpoint parameter, and select **AutoCare PAdb Inbound Endpoint** (the IIEP created for AutoCare PAdb imports).



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

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

In the example below:

- A File Loading Widget labeled as 'REFERENCE DATA IMPORTS' is displayed above its configurations.
- The File Loading Widget and its configurations are shown with the default configurations provided automatically when Easy Setup actions for the AutoCare component are completed.
- An IIEP for Qdb, Brand, PCdb, and VCdb imports are added within the same File Loading Widget as the IIEP for the AutoCare PAdb imports.

REFERENCE DATA IMPORTS

Qdb
Select file
Brand
Select file
PCdb
Select file
VCdb
Select file
PAdb
Select file

File Loading Widget Properties [go to parent](#)

*Inbound Integration Endpoint Parameters

- Inbound Integration Endpoint Parameter (Brand / AutoCareBrandInt
- Inbound Integration Endpoint Parameter (PCdb / AutoCarePCdbInbc
- Inbound Integration Endpoint Parameter (VCdb / AutoCareVCdbInbc
- Inbound Integration Endpoint Parameter (PAdb / AutoCarePAdbInbc

Add... Edit... Remove Up Down

Label REFERENCE DATA IMPORTS

Child Components

Edit component

Inbound Integration Endpoint Parameter Properties

*Inbound Integration Endpoint AutoCare PAdb Inbound Endpoint

Label PAdb

Save Cancel

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

Qdb Import Validation Rules

When importing an AutoCare Qdb (Qualifier Database) file many STEP validation rules are performed. The table below describes each of these validation rules, where the validation check occurs, what happens when a validation fails, and an example of a failed validation message (when applicable). However, it does not detail the file type's data rules per the AutoCare Association. This information can be found at www.autocare.org.

For information on the supported versions, see the **Supported Versions and Formats** topic within this guide.

Accepted File Extension: .zip

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
All tables present	Validation state	BGP ends in an error triggering the import to enter the Error state.	The required table '[Required Table Name]' is missing in the ZIP archive or is empty.
Mandatory values present	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table '[Table Name]' contains a row that misses value for the required column '[Column Name]' at line 3.
Correct number of columns (If last column is optional, then the last column can be missing without affecting validation.)	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table '[Table Name]' contains a row with wrong number of columns at line 2.
Loading of older version	Validation state	Warning is displayed.	Existing version '2015-09-25' is newer that provided '2014-11-28.'
Empty line	Validation state	Warning is displayed.	Table '[Table Name]' contains an empty row at line 3.

Brand Table Import Validation Rules

When importing an AutoCare Brand Table file, many STEP validation rules are performed. The table below describes each of these validation rules, where the validation check occurs, what happens when a validation fails, and an example of a failed validation message (when applicable). However, it does not detail the file type's data rules per the AutoCare Association. This information can be found at www.autocare.org.

For information on the supported versions, see the **Supported Versions and Formats** topic within this guide.

Important: For best results, AutoCare imports should be executed in the order specified within the **AutoCare Import Validation Rules** topic.

Accepted File Extension: .txt

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
All tables present	Validation state	BGP ends in an error triggering the import to enter the Error state.	The required table '[Required Table Name]' is missing in the archive or is empty.
Mandatory values present	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table '[Table Name]' contains a row that misses value for the required column '[Column Name]' at line 3.
Correct number of columns. (If last column is optional, then the last column can be missing without affecting validation.)	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table '[Table Name]' contains a row with wrong number of columns at line 2.
Loading of older version	Validation state	Warning is displayed.	Existing version '2015-09-25' is newer than provided '2014-11-28.'
Empty line	Validation state	Warning is displayed.	Table '[Table Name]' contains an empty row at line 3.

PCdb Import Validation Rules

When importing an AutoCare PCdb (Product Classification Database) file many STEP validation rules are performed. The table below describes each of these validation rules, where the validation check occurs, what happens when a validation fails, and an example of a failed validation message (when applicable). However, it does not detail the file type's data rules per the AutoCare Association. This information can be found at www.autocare.org.

For information on the supported versions, see the **Supported Versions and Formats** topic within this guide.

Important: For best results, AutoCare imports should be executed in the order specified within the **AutoCare Import Validation Rules** topic.

Accepted File Extension: .zip

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
All tables present	Validation state	BGP ends in an error, triggering the import to enter the Error state.	The required table '[Required Table Name]' is missing in the ZIP archive or is empty.
Mandatory values present	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table '[Table Name]' contains a row that misses value for the required column '[Column Name]' at line 3.
Correct number of columns (If last column is optional, then the last column can be missing without affecting validation.)	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table '[Table Name]' contains a row with wrong number of columns at line 2.
Loading of older version	Validation state	Warning is displayed.	Existing version '2015-09-25' is newer than provided '2014-11-28.'

VCdb Import Validation Rules

When importing an AutoCare VCdb (Vehicle Configuration Database) file many STEP validation rules are performed. The table below describes each of these validation rules, where the validation check occurs, what happens when a validation fails, and an example of a failed validation message (when applicable). However, it does not detail the file type's data rules per the AutoCare Association. This information can be found at www.autocare.org.

For information on the supported versions, see the **Supported Versions and Formats** topic within this guide.

Important: For best results, AutoCare imports should be executed in the order specified within the **AutoCare Import Validation Rules** topic.

Accepted File Extension: .zip

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
All tables present	Validation state	BGP ends in an error triggering the import to enter the Error state.	The required table 'BaseVehicle' is missing in the ZIP archive or is empty.
Mandatory values present	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table 'BaseVehicle' contains a row that misses value for the required column 'YearID' at line 3.
Correct number of columns (If last column is optional, then the last column can be missing without affecting validation.)	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table 'BaseVehicle' contains a row with wrong number of columns at line 2.
References between tables	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table 'Vehicle' contains a row where the 'RegionID' column value '1000' does not exist in referenced 'Region' file at line 13.
Correct type of values: date / numbers / max length	Validation state	BGP ends in an error triggering the import to enter the Error state.	Table 'Vehicle' contains a row where the 'PublicationStageDate' column value '33-10-2013 14:38:00' is not a proper instance of type 'Date with time' at line 35.
Loading of older version	Validation state	Warning is displayed.	Existing version '2015-09-25' is newer than provided '2014-11-28.'
Empty line	Validation state	Warning is displayed.	Table 'VehicleToBedConfig' contains an empty row at line 3.

TecDoc Reference Data Importer

The intention of the Automotive TecDoc Reference Data Importer is to provide an out-of-the-box solution for importing data included within a supported TecDoc Reference Data file. Because only supported versions will successfully upload, before attempting to upload a TecDoc Reference Data file, confirm the file version being uploaded is listed within the **Supported Versions and Formats** topic.

This section includes information on:

- Using TecDoc Reference Data Importer
- Configuring TecDoc Reference Data Importer

Prerequisites

It is expected that anyone using and/or configuring the TecDoc Reference Data Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Note: TecDoc is the world's largest automotive product and fitment data aggregator and is owned by the parts suppliers themselves. It was formed in 1998 to create a uniform vehicle and product description standard to allow over 500 suppliers to broadcast and share their product data with their customers.

Using TecDoc Reference Data Importer

A TecDoc Reference Data file can be imported into STEP by uploading it to either a configured hotfolder on an application server, or through a File Loading Widget on a Web UI Homepage.

The following topics provide information on using the TecDoc Reference Data Importer.

- Importing Reference Data Files via Web UI
- Reference Data Import Validation Rules
- Error Handling for Reference Data Imports

Prerequisites

It is expected that anyone using and/or configuring the TecDoc Reference Data Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For more information about uploading files to an application server, see the **File Loading** section of the **Automotive Quick Start Guide** found within the **Solution Enablement** section of **STEP Online Help**.

For more information about using Workbench to monitor and process imports, see the **Workflows** section of **STEP Online Help**.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Importing Reference Data Files via Web UI

The intention of the Automotive TecDoc Reference Data Importer Web UI setup included within this topic is to provide an out-of-the-box solution for importing data included within a supported TecDoc Reference file. Because only supported versions will successfully upload, before attempting to upload a TecDoc Reference Data file, confirm the file version being uploaded is listed within the **Supported Versions and Formats** topic.

A TecDoc Reference Data file includes information on vehicle data (i.e., cars, commercial vehicles, vans, motorcycles), vehicle engines, axles, drivers' cabs, generic articles (product classification), criteria (attributes), and key table suggestions regarding search structures.

Prerequisites

Within this section users will be directed to view details for a Controller Entity within Web UI using a Node Details screen configured for Entities. Before this can be available to users, an Entity Details screen must be created.

To easily start and monitor a TecDoc Reference Data Import, recommended practice is to use a Web UI Import Controller screen specific to the TecDoc Reference Data Importer. When the Easy Setup actions for the TecDoc standard are completed, the TecDoc Import Controller screen is automatically created and configured for use. Additional information can be found in the **Configuring TecDoc Reference Importer** topic.

It is expected that anyone using and/or configuring the TecDoc Reference Data Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

If the Easy Setup actions for the TecDoc Component model have been completed, then the functionality explained within this topic should be available. Otherwise, configuration is necessary. For more information, see the **TecDoc Reference Data Importer** topic.

Import Process Overview

Once a valid automotive data file is uploaded to a hotfolder on the application server (optionally using a File Loading Widget), the file is picked up from the hotfolder by an IIEP, and the IIEP creates an Entity in STEP that represents the file. This Entity object is called the 'controller' and contains basic data about the file and the file's status in the workflow. Web UI users are able to monitor the import status using a Status Selector Widget and an Import Controller Screen. Once an import file reaches the 'Ready for Import' state, then users can start the import by clicking on the 'Start import' button within the Control Panel screen. At that time, the BGP service (that runs as part of the Import state) allows for the configured business rules to act on the objects being imported.

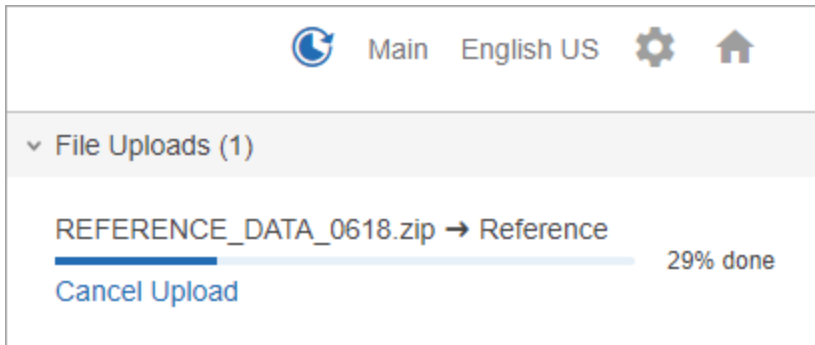
Using the TecDoc Reference Data Importer in Web UI

1. Access the TecDoc Web UI Homepage.
2. Upload a valid TecDoc Reference Data file to the application server hotfolder (root/upload/hotfolders/TecDocReferenceInputFolder), using the 'Reference' drop zone of the 'TECDOC IMPORTS' File Loading Widget.

For more information, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

- Once the upload has started, users can view the progress of the upload using the Background Process Notification Component.

In the example below, the BGP Notification Panel is expanded and the file, 'REFERENCE_DATA_0618.zip' is 29% processed.



For more information on using the BGP Notification Component and side panel, see the **Background Process Notification Component** topic within the **Main Properties Overview** section of **STEP Online Help**.

Once created, the Controller is initiated into the workflow associated with the importer (TecDoc Reference Import). From there, the TecDoc Reference Import workflow takes over processing of the file via a series of states using business rules and background processes to carry out the processing of the file. Each import has an associated workflow and all proceed through the same states by default. However, it is intended that customers will expand on the existing states and actions to add their own validations, reporting, and additional processing as needed.

Important: It is critical to understand that it is only the Controller Entity that is in the workflow - the objects being acted on (created / updated / deleted) via information supplied in the import file are not in the workflow. Therefore, running standard business actions acting on current object will impact the Controller Entity only, not the objects in the input file. In order for the Business Action to act on the objects being imported, the Business Action must be added to the 'Import action' parameter on the 'Import' state. For more information, see the **Import State** topic within the **Default Workflow States and Functions** section.

- If an Entity Details screen has been configured, then when the IIEP creates the Controller Entity object, users can view the Controller and its important information by navigating within the Tree side panel to the Import Flow Root > TecDoc Reference. Otherwise, if a user clicks on an Entity in the Tree side panel, an error will display indicating the configuration is invalid.

Note: If an error message displays when selecting an Entity within a Web UI, then the Entity Details screen has not been configured.

In the example below, the Tree side panel is expanded, and the 'REFERENCE_DATA_0618.zip' Controller with ID 'Controller- 102441' is displayed below the 'TecDoc Reference' Entity. When the Controller is selected, many details can be viewed within the Node Details screen.

Note: In the screenshot below, the 'Enable Tag Conversion' parameter within the Attribute Value Group Component is enabled.

Node Details	
ID	Controller-102441
Name	REFERENCE_DATA_0618.zip
Object Type	Import Flow Controller Type
Revision	Date: 2018.08.08 Revision number: 0.1 Last Edited by: TECDOCREFERENCEDATAIMPORT Time: 16:26:12
Automotive Import Flow State BGP	<p><?xml version="1.0" encoding="UTF-8"?></p><p><StringMap<span class="step-
Import Flow Endpoint ID	TecDocReferenceInboundEndpoint
Import Flow File Type	TecDocReferenceData
Import Flow Overall Status	Done creating delta file
Import Flow State Status	<p><?xml version="1.0" encoding="UTF-8"?></p><p><StringMap<span class="step-
Import Flow Workflow ID	TecDocReferenceImport

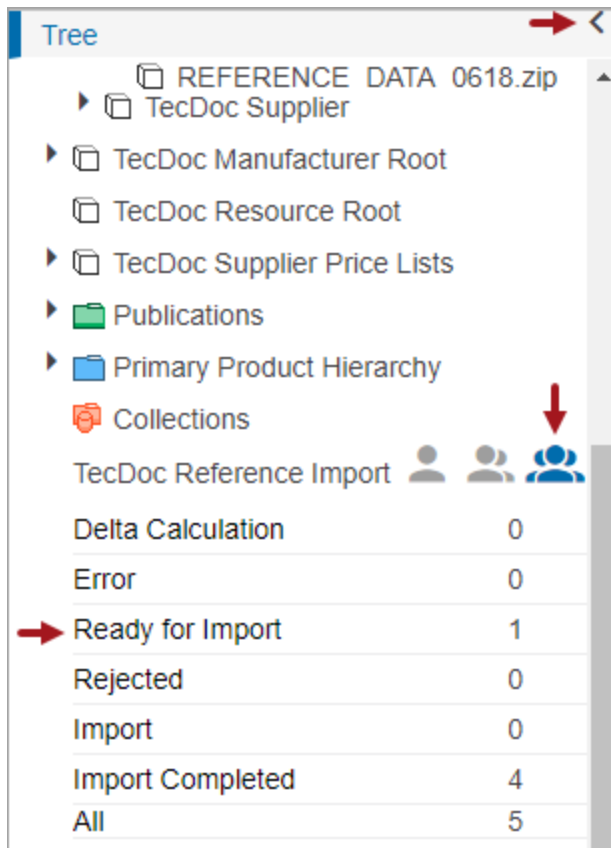
- **Name:** STEP Name of the Controller entity. This is generated from the uploaded file original name.
- **Automotive Import Flow State BGP:** Attribute used to store the IDs of the background processes.
- **Import Flow State Status:** Attribute used to store and display the status of each process (rather than the overall status).
- **Import Flow Overall Status:** Attribute used to store and display the global status of the file (rather than the process specific status).

5. As the file is uploaded, and the Entity Controller moves through the TecDoc Reference Import workflow, users can monitor the progress using the 'TECDOC REFERENCE IMPORT' Status Selector Widget on the Web UI

homepage and/or the left side panel.

In the example below, the Tree panel is expanded, and the **Triple user button** is selected so that all items assigned to any user are displayed. Notice that one file is in the 'Ready for Import' state, and four files in the 'Import Completed' state, for a total of five files.

For more information, see the **Moving Tasks Trough a Workflow in Web UI** topic within **STEP Online Help**.



- Clicking on the **Ready for Import** state within the Status Selector homepage and/or the left side panel widget will navigate the user to the TecDoc Reference Data Import Controller Screen with only those Controllers in the 'Ready for Import' state displayed.
- Clicking the Controller icon displays the Import Details below the Controller list.

In the example below, Controller 'REFERENCE_DATA_0618.zip' (ID 'Controller-102441') with an overall status of 'Done creating delta file' is selected and the Import Details are displayed below the Controller list.

TecDoc Reference Data Import Controller Screen

Process	File Name	Overall Status
Controller-102441	REFERENCE_DATA_0618.zi	Done creating delta file

1-1 of 1

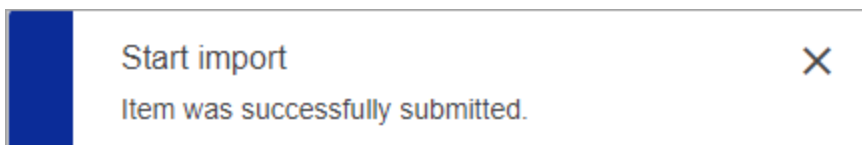
Import Details

Process	Started	Duration	Started I	Status	Background Process Link
Validation	2018-08-08 16:26:12	28 secs	STEPSYS	Validation completed	completedwitherrors
Conversion	2018-08-08 16:26:40	2 mins 19 secs	STEPSYS	Conversion completed	succeeded
Delta Calculator	2018-08-08 16:28:59	3 mins 28 secs	STEPSYS	Delta calculation completed	succeeded

1-4 of 4

[Start import](#) [Reject](#) [Discard file](#)

8. Optionally, before starting the import, clicking on the 'completedwitherrors' Background Process Link will display the Background Process Details screen where users can view or download the details of the validation process.
9. Click the **Reject** button to move the Controller to the Rejected state. For more information, see the **Rejected State** topic within the **Import Framework** section.
10. Click the **Start import** button, the Start import dialog (shown below) displays at the top of the screen, and the Controller is moved to the Import state of the workflow. For more information, see the **Import State** topic within the **Import Framework** section.



Importing a TecDoc Reference Data file can take several hours, but the import progress can be monitored using the 'TECDOC REFERENCE IMPORT' Status Selector Widget on the Web UI homepage and/or the left side panel or the Import Controller Screen, Import process Status column.

In the example below, the TecDoc Reference Data Import Controller Screen displays the Overall Status and the Import process status as 'Importing files...'. Additionally, the Background Process Link column displays a link to the Background Process Details screen as 'running.'

TecDoc Reference Data Import Controller Screen

Process	File Name	Overall Status
Controller-102441	REFERENCE_DATA_0618.zip	Importing files...

Import Details

Process	Started Time	Duration	Started By	Status	Background Process Link
Validation	2018-08-08 16:26:12	28 secs	STEPSYS	Validation completed	completedwitherrors
Conversion	2018-08-08 16:26:40	2 mins 19 secs	STEPSYS	Conversion completed	succeeded
Delta Calculation	2018-08-08 16:28:59	3 mins 28 secs	STEPSYS	Delta calculation completed	succeeded
Import	2018-08-10 12:37:44	1 hr 6 mins 44 secs	STEPSYS	Importing files... 11 running	

1-4 of 4

- Optionally, click the 'running' Background Process Link (shown above) to view the details of the import process within the Background Process Details screen.

Once the import has completed, the Controller is automatically moved to the Import Completed state. For more information, see the **Import Completed State** topic within the **Import Framework** section.

For more information about Reference Data import validation rules and example error messages, see the **Reference Data Import Validation Rules** topic.

For more information on automotive workflow states, see the **Default Workflow States and Functions** topic.

Reference Data Import Validation Rules

When importing a TecDoc Reference Data file many STEP validation rules are performed. The table below describes each of these validation rules, where the validation check occurs, what happens when a validation fails, and an example of a failed validation message (when applicable). However, it does not detail the file type's data rules per TecAlliance.

For information on the supported versions, see the **Supported Versions and Formats** topic.

Accepted File Extension: ZIP and/or 7z (A compressed archive file format by the 7-Zip achiever that supports several different data compression, encryption and pre-processing algorithms.)

Validation Rule	Occurs	When Validation Fails	Failed Validation Message Example
Attribute validity	Import state	If 'Continue on Error' is enabled, then the record is skipped and a warning displays in the BGP report. Otherwise, the import fails.	Error: The attribute 'TD_ATTR_BJvon' is not valid for 'classification 'TD_AXLE_CODE.'
Character length limits	Import state	If 'Continue on Error' is enabled, then the record is skipped and a warning displays in the BGP report. Otherwise, the import fails.	Error in this import 04_LOVDefinitions.xml setting completed with errors - Error: The value 'Value' isn't valid for LOV 'LOV ID': Length: 60 does not fit: 'Value.' The automotive importer adheres to the character length limits set by TecAlliance for each of the Delta Keys. These limits are listed within the TecAlliance TecDoc Data Format guide. When errors occur, the error message includes the necessary character length (as shown in the example above).
Missing Object Type when allocating Engine Number (MotNr)	Import state	If 'Continue on Error' is enabled, then the record is skipped and a warning displays in the BGP report. Otherwise, the import fails.	Error in this import 25_CVTypesAndEngineAllocation.xml setting completed with errors - Error: Missing object type for new object: TD_CV_TYPE_23420 of type: com.stibo.core.domain.Classification. For more information, see the Missing Object Type When Allocating Engine Number topic.
Object Type existence	Import state	If 'Continue on Error' is enabled, then the record is skipped and a warning displays in the BGP report. Otherwise, the import fails.	ObjectType 'TD_NoAssemblyGroupSynonym' not found.

Error Handling for Reference Data Imports

Errors may occur during the import of a TecDoc Data file. When errors occur, detailed error messages display within a BGP Execution Report. This section describes an error that can occur by listing an overview of the error, error message template, error message example, where the error messages can be viewed, and how best to address the error.

Important: The handling of validation errors can be configured by enabling / disabling the 'Continue on Error' parameter on the Validation state of each import workflow. By default, the 'Continue on Error' parameter is disabled. For more information, see the **Validation Error Handling** topic within the **Importing Automotive Data** section.

The following are error types related to importing a TecDoc Reference Data file:

- Link Type (Reference Type) Not Valid for Generic Article
- Missing Object Type When Allocating Engine Number
- Missing Object Type When Allocating German KBA Vehicle Number
- Missing Object Type When Adding Additional Descriptions to Vehicle Model Series
- Missing Country Code When Importing Vehicle Country Specifications
- Validation Error When Importing LOV Values

For information on importing TecDoc Reference files, see the **Importing Reference Data Files via Web UI** topic.

Link Type (Reference Type) Not Valid for Generic Article

Errors may occur during the import of a TecDoc Data file. When errors occur, detailed error messages display within a BGP Execution Report. This section describes an error that can occur by listing an overview of the error, error message template, error message example, where the error messages can be viewed, and how best to address the error.

Important: The handling of validation errors can be configured by enabling / disabling the 'Continue on Error' parameter on the Validation state of each import workflow. By default, the 'Continue on Error' parameter is disabled. For more information, see the **Validation Error Handling** topic within the **Importing Automotive Data** section.

Overview

This type of error occurs when a Generic Article exists within the Universal Assembly Group but the link type (Reference Type) in the import file is supposed to be for a Standard Generic Article. Thus, the Reference Type (Standard GA to Search Tree) is valid for a Standard Generic Article instead of a Universal Generic Article.

Conversely, this error can occur when a Generic Article exists in the Standard Assembly Group but the link type (Reference Type) in the import file is supposed to be for a Universal Generic Article. Thus, the Reference Type (Universal GA to Search Tree) is valid for a Universal Generic Article instead of a Standard Generic Article.

When a user views the error message, this is apparent because the message template states: The link type <'Reference ID'> is not valid for 'classification 'TD_GENART_<GenArtNr>.'"

When viewing the example error message below, it is clear that the Reference Type that is not valid is the Reference ID 'TD_StandardGAToSearchTree(PC)' for the Universal Generic Article 'classification 'TD_GENART_06911.'"

Error Message Template

Error in this import 06_Search_Tree.xml setting completed with errors - Error: The link type <'Reference ID'> is not valid for 'classification 'TD_GENART_<GenArtNr>.'"

Error Message Example

Error in this import [import file name] setting completed with errors - Error: The link type [Reference Type ID] is not valid for classification [Classification ID].'

Viewing the Error Message

The screenshot below displays this error message example within a Background Process Details Web UI screen.


Background Process Details

ID BGP_102235

Started By USERB

Description Running background process ImportingFlowImportService from RunBackgroundProcessServiceAction



Template ID ImportingFlowImportService

Status  Completed With Errors

Started 7/26/18 11:50:24 AM

Finished 7/26/18 12:17:08 PM

Elapsed 26 m 44 s

 Select all  Export

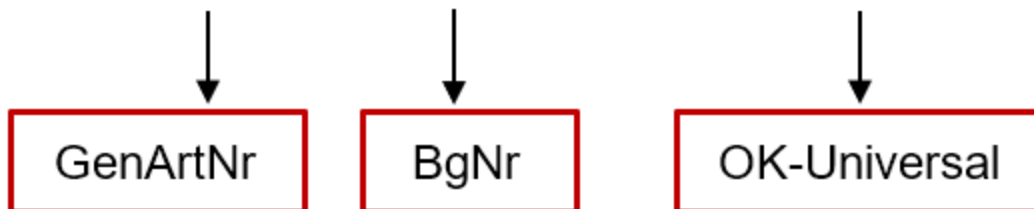
ID	Type	Text
<input type="checkbox"/> 25770	Error	Error in this import 06_Search_Tree.xml setting completed with errors - Error: The link type 'TD_StandardGAToSearchTree(Engine)' is not valid for 'classification 'TD_GENART_06911''

Explanation

Because this error pertains to a Generic Article, and all Generic Articles are listed within the TecDoc Reference Data file Data Table 320, it is important to understand data table 320.

For this explanation the following example data from Data Table 320 is used. The Delta Keys that pertain to this error type are displayed with red text.

99993200691105824016017440000066811010000



Understanding Data Table 320

The following important Delta Keys are within this table:

GenArtNr: Five digit number representing a Generic Article (GA) number. Stored in STEP as a 'Universal Assembly GA' Object Type using the following prefix: 'TD_GENART_.'

In the example below, GenArtNr 06911 is displayed in Workbench as the 'Universal Assembly GA' Object Type, Adjustment Tool Set, valve timing (TD_GENART_06911).

Classification		Sub Products	References	Referenced By	Images & Documents	Tables
Description						
Name	>	>	Value			
ID	>	TD_GENART_06911				
Name	>	Adjustment Tool Set, valve timing				
Object Type	>	Universal Assembly GA				
Revision	>	0.2 Last edited by TECDOCREFERENCEDATAIMPORT on Thu Jul 26				
Approved	>	✘ Never Been Approved				
Translation	>	Not Translated				
Path	>	Classification 1 root/TecDoc/Generic Articles/Universal Assembly Gr				
Visibility	>					
ATTR_Delete	>	<input type="checkbox"/>	0			
ATTR_Engine Linkage Propos	>	<input type="checkbox"/>				
Axle Linkage Allowed	>	<input type="checkbox"/>	false			
CV Linkage Allowed	>	<input type="checkbox"/>	false			
Engine Linkage Allowed	>	<input type="checkbox"/>	true			
Generic Article Number	>	<input type="checkbox"/>	6911			
Independent Search Allowed	>	<input type="checkbox"/>	false			
PC Linkage Allowed	>	<input type="checkbox"/>	true			
Reserved	>	<input type="checkbox"/>	123 06911			

BgNr: Four digit number representing an Assembly Group where the GenArtNr is stored.

In the example below, BgNr 0160 is displayed in Workbench as the 'Universal Assembly Group' Object Type, Special Tools, universal (TD_ASMGRP_0160).

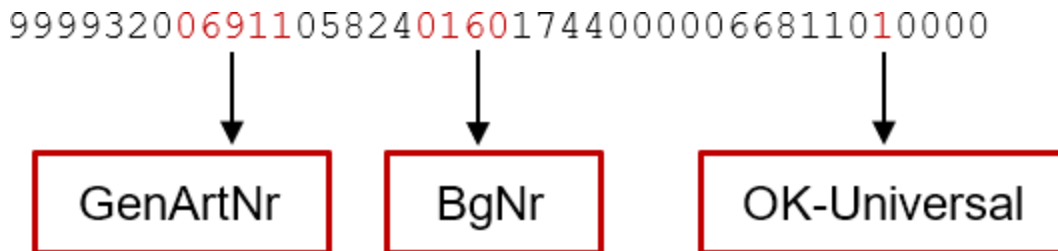
The screenshot shows the 'Special Tools, universal rev.0.1 - Classification' window. On the left is a 'Tree' view with a hierarchy: TecDoc > Generic Articles > Universal Assembly Groups > Special Tools, universal > Special Tools, universal. On the right is a table with the following data:

Name	Value
ID	TD_ASMGRP_0160
Name	Special Tools, universal
Object Type	Universal Assembly Group
Revision	0.1 Last edited by STEPSYS on Thu Apr 26 02:15:08 EDT 2018
Approved	✘ Never Been Approved
Translation	Not Translated
Path	Classification 1 root/TecDoc/Generic Articles/Universal Assembly
Visibility	

OK-Universal: One digit number defining if the Reference Type should be either Universal Assembly Group or Standard Universal Assembly Group.

- A value of '1' defines the Reference Type as Universal GA to search tree.
- A value of '0' defines the Reference Type as Standard GA to search tree.

In the example data from Data Table 320, the 'OK-Universal' value is '1.' Thus, it is expected that upon import of this data file the GenArtNr 06911 will reference the BgNr 0160.



This example data would result in:

- Creating Generic Article Number '06911' in STEP as 'TD_GENART_06911' beneath the Universal Assembly Group (TD_ASMGRP_0160). And the OK-Universal number '1' indicates that the link type / Reference Type is Standard GA to search tree. As the link type is not valid between the Generic Article (as this is under Universal assembly group and not the Standard assembly group) and the search tree, it displays an error.

If the appropriate values defining the Reference types are not present, STEP may not accurately determine the classification reference between the generic article and the search tree.

In this case, STEP will continue importing the process (Stops the process if 'Continue on Error' parameter is disabled on the Import state) and report an error to the

Resolution

To resolve this, manually edit the 'OK-Universal' number in Data Table 320 to value '0' and import again. This will change the reference type to 'Standard GA to Search Tree'.

Missing Country Code When Importing Vehicle Country Specifications

Errors may occur during the import of a TecDoc Data file. When errors occur, detailed error messages display within a BGP Execution Report. This section describes an error that can occur by listing an overview of the error, error message template, error message example, where the error messages can be viewed, and how best to address the error.

Important: The handling of validation errors can be configured by enabling / disabling the 'Continue on Error' parameter on the Validation state of each import workflow. By default, the 'Continue on Error' parameter is disabled. For more information, see the **Validation Error Handling** topic within the **Importing Automotive Data** section.

Overview

This type of error occurs when the import file tries to allocate Vehicle Country Specifications to a particular Vehicle Type (PC or CV) through the Reference Type 'Vehicle Country Deviation', but the target object 'Country Code' (known in STEP as [Country]) is missing.

When a user views the error message, this is apparent because the message template states: Target 'Object ID' of reference not found.

When viewing the example error message below, it is clear that the object that is missing is the Country Code object with the Object ID 'TD_C_IBE.'

Error Message Template

Error in this import 39_VehicleCountrySpecifications-table124.xml setting completed with errors - Error: Target 'Object ID' of reference not found.

Error Message Example

Error in this import 39_VehicleCountrySpecifications-table124.xml setting completed with errors - Error: Target 'TD_C_IBE' of reference not found.

Viewing the Error Message

The screenshot below displays this error message example within a Background Process Details Web UI screen.


Background Process Details

ID: BGP_102360

Started By: USERB

Description: Running background process ImportingFlowImportService from RunBackgroundProcessServiceAction


Template ID: ImportingFlowImportService

Status:  Completed With Errors

Started: 8/1/18 2:25:57 PM

Finished: 8/1/18 2:36:44 PM

Elapsed: 10 m 50 s

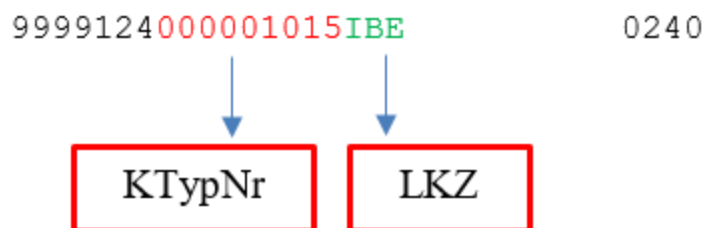
Select all  Export

ID	Type	Text
<input type="checkbox"/> 15070	Error	Error in this import 39_VehicleCountrySpecifica table124.xml setting completed with errors - Error: Target 'TD_C_IBE' of reference not found

Explanation

Because this error pertains to the Vehicle Type object, allocation of Vehicle Country Specifications and Country Code and the Country Codes are listed within the TecDoc Reference Data file Data Table 010, and the allocation of Vehicle Country Specifications is listed within the Data Table 124, it is important to understand Data Table 124.

For this explanation the following example data from Data Table 124 is used. The Delta Keys that pertain to this error type are displayed with red and green text.



Understanding Data Table 010


Data Table 010 lists Country Codes featured in Delta Key **LKZ**.

Understanding Data Table 124

The following important Delta Keys are within this table:

KTypNr: Nine digit number representing the Vehicle Types.

In the example below, KTypNr 000001015 is displayed in Workbench as the 'Vehicle Type (PC)' Object Type, 'FORD Bus - TRANSIT Bus (V__) - 2.5 D' (TD_PC_TYPE_1015).

Tree		← FORD Bus - TRANSIT Bus (V__) - 2.5 D (VAS, VBL, VIL, VUL, VZ)	
		Classification	Sub Products
		References	Referenced By
		Images & Documents	Tables
+	TRANSIT Bus (FD__, FB__, FS__, FZ__,		
+	TRANSIT Bus (T__) 10.85-09.92		
-	TRANSIT Bus (V__) 11.77-10.86		
-	FORD Bus - TRANSIT Bus (V__) - 1.6 (V		
-	FORD Bus - TRANSIT Bus (V__) - 1.6 15		
-	FORD Bus - TRANSIT Bus (V__) - 2.0 (V		
-	FORD Bus - TRANSIT Bus (V__) - 2.0 (V		
-	FORD Bus - TRANSIT Bus (V__) - 2.0 (V		
-	FORD Bus - TRANSIT Bus (V__) - 2.4 D		
-	FORD Bus - TRANSIT Bus (V__) - 2.5 D		
		Description	
	Name	>	Value
	ID	>	TD_PC_TYPE_1015
	Name	>	FORD Bus - TRANSIT Bus (V__) - 2.5 D (VAS, VBL, VIL, VUL, VZS) 1983-10-01
	Object Type	>	Vehicle Type (PC)
	Revision	>	0.1 Last edited by STEPSYS on Thu Apr 26 09:40:32 EDT 2018
	Approved	>	 Never Been Approved

LKZ: Maximum three letter alphabet representing Country Code extracted from Data Table 010.

In the example below, the first image of the screenshot displays the Vehicle type (PC) Object 'FORD Bus – TRANSIT Bus (V__) 2.5D' referenced through the reference type 'Vehicle country deviation' to the target Country 'France (F)', and the second image shows the Country object 'France(F)' displayed within Workbench.

Tree

- TRANSIT Bus (FD_ FB_ FS_ FZ_)
- TRANSIT Bus (T_) 10.85-09.92
- TRANSIT Bus (V_) 11.77-10.86
 - FORD Bus - TRANSIT Bus (V_) - 1.6 (V)
 - FORD Bus - TRANSIT Bus (V_) - 1.6 (V)
 - FORD Bus - TRANSIT Bus (V_) - 2.0 (V)
 - FORD Bus - TRANSIT Bus (V_) - 2.0 (V)
 - FORD Bus - TRANSIT Bus (V_) - 2.4 D
 - FORD Bus - TRANSIT Bus (V_) - 2.5 D
- TRANSIT Bus 08.13-
- TRANSIT CONNECT (P65_ P70_ P80_) 06
- TRANSIT CONNECT Box 02.13-
- TRANSIT CONNECT Kombi 02.13-

<FORD Bus - TRANSIT Bus (V_) - 2.5 D

Classification	Sub Products	References	Referenced B
Ungrouped Classification References			
Reference Type	>	Target	
> Body Types (PC) To Country	+		
> Type Description Country Deviation	+		
> Vehicle (PC) To Engine	+	4AB	
> Vehicle (PC) To Synonym	+		
> Vehicle Country Deviation (PC)	+	France	
		Great Britain	
		Greece	
		Italy	
		Poland	

Tree

- Ethiopia
- Faeroes
- Falkland Islands
- Fidji
- Finland
- France
- French Guyana
- French Polynesia
- French Southern
- Gabon
- Gambia
- Georgia
- Germany
- Ghana
- Gibraltar
- Great Britain
- Greece
- Greenland
- Grenada

<

Classification	Sub Products	References	Referenced By	Images & I
Description				
Name	>	Value		
> ID		TD_C_F		
> Name		France		
> Object Type		Country		
> Revision		0.1 Last edited by STEPSYS on Thu Apr 26 09:		
> Approved		Never Been Approved		
> Translation		Not Translated		
> Path		Classification 1 root/TecDoc/Countries/France		
> Visibility				
> Area Code		abc		
> ATTR_ISOCode2		abc FR		
> ATTR_ISOCode3		abc FRA		
> ATTR_ISOCode Nr		abc 250		

In this example, because the Country object with ID: 'TD_C_IBE' (Data imported from Data Table 010) is missing from the appropriate STEP classification hierarchy, and the file 124 includes instruction to allocate the Vehicle Country Specification (Country Code) to that object, STEP is displaying a missing object error.

In short, object 'TD_C_IBE' is missing from file 010, but the information associating it is present in file 124.

If information on 'Country Code' is not present in Data Table 010 (LKZ) or not loaded in STEP for some other reason, and the same is present in the Data Table 124 (LKZ), then STEP will throw a missing information error.

Resolution

To resolve this, create the missing object and either import the Reference Data import file again, or manually create the 'Vehicle Type (PC or CV)' object within the appropriate STEP classification hierarchy.

Missing Object Type When Adding Additional Descriptions to Vehicle Model Series

Errors may occur during the import of a TecDoc Data file. When errors occur, detailed error messages display within a BGP Execution Report. This section describes an error that can occur by listing an overview of the error, error message template, error message example, where the error messages can be viewed, and how best to address the error.

Important: The handling of validation errors can be configured by enabling / disabling the 'Continue on Error' parameter on the Validation state of each import workflow. By default, the 'Continue on Error' parameter is disabled. For more information, see the **Validation Error Handling** topic within the **Importing Automotive Data** section.

Overview

This type of error occurs when the import file attempts to add 'Additional Descriptions' (populated within STEP as value to the attribute(s) 'Country Dependent Model Designation' and / or 'Model Generation') to a particular Vehicle Model Series but the data within the table that imports Vehicle Model Series objects (known in STEP as [Vehicle Model Series (CV)]) is missing.

When a user views the error message, this is apparent because the message template states: Missing object type for new Object: <Object ID> of type: com.stibo.core.domain.Classification.

When viewing the example error message below, it is clear that the Object that is missing is the Vehicle Model Series (CV) Object with the Object ID 'TD_CVMDL_38367'.

Error Message Template

Error in this import [import file name] setting completed with errors - Error: Missing object type for new Object: [Object ID] of type: com.stibo.core.domain.Classification.


Error Message Example

Error in this import 31_DescriptionsToVehicleModelSeries-Table140.xml setting completed with errors - Error: Missing object type for new object: TD_CVMDL_38367 of type: com.stibo.core.domain.Classification.

Viewing the Error Message

The screenshot below displays this error message example within a Background Process Details Web UI screen.

Background Process Details

ID	BGP_102360
Started By	USERB
Description	Running background process ImportingFlowImportService from RunBackgroundProcessServiceAction
Template ID	ImportingFlowImportService
Status	 Completed With Errors
Started	8/1/18 2:25:57 PM
Finished	8/1/18 2:36:44 PM
Elapsed	10 m 50 s

Select all  Export

ID	Type	Text
<input type="checkbox"/> 14640	Error	Error in this import 31_DescriptionsToVehicleModelSeries- Table140.xml setting completed with errors - Error: Missing object type for new object: TD_CVMDL_38367 of type: com.stibo.core.domain.Classification

Explanation

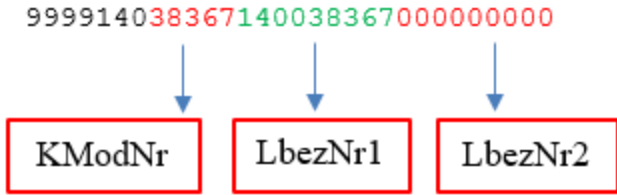
Because this error pertains to the 'Vehicle Model Series (CV)' Object (featured in Data Table 110) and adding Additional Descriptions to Vehicle Model Series (featured in Data Table 140), it is important to understand Data Tables 110 and 140.

Understanding Data Table 110

Data Table 110 lists Vehicle Model Series (CV) Objects featured in Delta Key **KModNr**.

Understanding Data Table 140

For this explanation the following example data from Data Table 140 is used. The Delta Keys that pertain to this error type are displayed with red and green text.



The following important Delta Keys are within this table:

KModNr: Five digit number representing the Vehicle Model Series [Vehicle Model Series (CV)].

LbezNr1: Nine digit number representing the Description Number of the sales description (also known in STEP by Attribute 'Country Dependent Model Designation' (STEP ID: TD_ATTR_LBezNr1)).

LbezNr2: Nine digit number representing the Description Number of the model generation (also known in STEP by Attribute 'Model Generation' (STEP ID: TD_ATTR_LBezNr2)).

In the example below, KModNr 38364 is displayed in Workbench as the 'Vehicle Model Series (CV)' Object Type 'RG Series 01.96-' (TD_CVMDL_38364). Because KModNr 38367 is not present in Data Table 110 the Vehicle Model Series (CV) with Object ID: TD_CVMDL_38367 is not created in STEP. The screenshot below is solely for representational purpose of how an object with object type Vehicle Model Series (CV) displays in the workbench.

The screenshot shows the Stibo Systems Workbench interface. On the left is a 'Tree' view showing a hierarchy of folders including HIGER, HILLMAN, HINDUSTAN, HINO, and various vehicle series. The 'RG Series 01.96-' folder is selected. On the right is a detailed view of the 'RG Series 01.96- rev.0.1 - Classification' object. The view includes a 'Description' table with the following data:

Name	Value
ID	TD_CVMDL_38364
Name	RG Series 01.96-
Object Type	Vehicle Model Series (CV)
Revision	0.1 Last edited by STEPSYS on Thu Apr 26 09:37:53 EDT 2018
Approved	Never Been Approved
Translation	Not Translated
Path	Classification 1 root/TecDoc/Linking Targets/HINO/Commercial Vehicle HINO
Visibility	
ATTR_Delete	0
ATTR_Sort Nr	008
Axle	No
Construction Year To	
Construction Year From	1996-01-01
Country Dependent Model Designation	RG Series
CV	Yes
Model Code	abc
Model Generation	abc
ModelNo	123 38364
PC	No

Three rows in the table are highlighted with red boxes: 'Country Dependent Model Designation', 'Model Code', and 'Model Generation'.

In this example, because the 'Vehicle Model Series (CV)' with ID 'TD_CVMDL_38367' is missing from the appropriate STEP classification hierarchy, and Data Table 140 includes instruction to add values to the attribute(s) 'Country Dependent Model Designation' and/or 'Model Generation,' STEP is displaying a missing object error.

If information on 'Vehicle Model Series Number' is not present in STEP (KModNr), but is present in Data Table 140 (KModNr), then STEP will display a missing information error.

Resolution

To resolve this, manually create the missing object within the appropriate STEP classification hierarchy and import the Reference Data file again.

Missing Object Type When Allocating Engine Number

Errors may occur during the import of a TecDoc Data file. When errors occur, detailed error messages display within a BGP Execution Report. This section describes an error that can occur by listing an overview of the error, error message template, error message example, where the error messages can be viewed, and how best to address the error.

Important: The handling of validation errors can be configured by enabling / disabling the 'Continue on Error' parameter on the Validation state of each import workflow. By default, the 'Continue on Error' parameter is disabled. For more information, see the **Validation Error Handling** topic within the **Importing Automotive Data** section.

Overview

This type of error occurs when the import file attempts to allocate an Engine Number to a particular TecDoc Commercial Vehicle Type (TD_CV_TYPE), but the data within the table that imports the Commercial Vehicle object, known in STEP as [Vehicle Type (CV)], is missing.

When a user views the error message, this can be understood because the message template states: Missing object type for new object: <Object ID> of type: com.stibo.core.domain.Classification.

When viewing the example error message below, it is clear that the Object that is missing is the Commercial Vehicle object with the Object ID 'TD_CV_TYPE_23420.'

Error Message Template

Error in this import 25_CVTypesAndEngineAllocation.xml setting completed with errors - Error: Missing object type for new object: <Object ID> of type: com.stibo.core.domain.Classification

Error Message Example

Error in this import 25_CVTypesAndEngineAllocation.xml setting completed with errors - Error: Missing object type for new object: TD_CV_TYPE_23420 of type: com.stibo.core.domain.Classification

Viewing the Error Message

The screenshot below displays this error message example within a Background Process Details Web UI screen.


Background Process Details

ID: BGP_102235

Started By: USERB

Description: Running background processImportingFlowImportService from RunBackgroundProcessServiceActio


Template ID: ImportingFlowImportService

Status:  Completed With Errors

Started: 7/26/18 11:50:24 AM

Finished: 7/26/18 12:17:08 PM

Elapsed: 26 m 44 s

Select all  Export

ID	Type	Text
<input type="checkbox"/> 27310	Error	Error in this import 25_CVTypesAndEngineAllocation.xml setting completed with errors - Error: Missing object type for new object: TD_CV_TYPE_23420 of type: com.stibo.core.domain.Classification

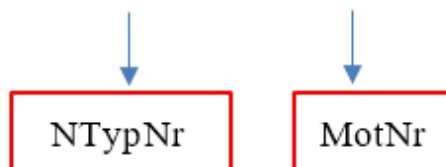
Explanation

Because this error is about a Commercial Vehicle Type object and allocation of Engine Numbers, and the Commercial Vehicle Types are handled within the TecDoc Reference Data file data table 532, and the allocation of Engine Numbers are handled within data table 537, it is important to first understand data table 537.

For this explanation the following example data from Data Table 537 is used. The Delta Keys that pertain to this error type are displayed with red text.

999953700002342000000101670

0



Understanding Data Table 537

The following important Delta Keys are within this table:

NTypNr: Nine digit number representing a Commercial Vehicle Type.

In the example below, NTypNr 000023420 is displayed in Workbench as the 'Vehicle Type (CV)' Object Type, 'HICOM Closed - MTB - MTB 145 1994-01-01-2004-12-31, kW, HP, cmTech 2771' (TD_CV_Type_23420).

The screenshot shows the Stibo Systems Workbench interface. On the left is a 'Tree' view showing a hierarchy of folders under 'TecDoc', including 'Countries', 'Country Groups', 'Generic Articles', 'Identification Keys', 'Linking Targets', and various manufacturer folders like 'HERO ELECTRIC MC', 'HESKET MOTORCYCLES', 'HESSTON', 'HESTON CORPORATION', 'HEULIEZ', 'HICOM', 'HIGER', 'HILLMAN', 'HINDUSTAN', 'HINO', 'HISPANO-SUIZA', 'HITACHI', 'HMRacing MOTORCYCLES', 'HOBBYCAR', 'HOES', 'HOLDEN', 'HOLDER', 'HOLMAN', and 'HOLMER'. The 'HICOM' folder is expanded to show 'Commercial Vehicle HICOM', which is further expanded to show 'MTB 01.94-12.04', and then 'HICOM Closed - MTB - MTB 145' is selected.

On the right, the 'HICOM Closed - MTB - MTB 145 1994-01-01-2004-12-31, kW, HP, cmTech 2771' data table is displayed. The table has a 'Description' tab selected, showing the following data:

Name	Value
ID	TD_CV_TYPE_23420
Name	HICOM Closed - MTB - MTB 145 1994-01-01-2004-12-31, kW, HP, cmTech 2771
Object Type	Vehicle Type (CV)
Revision	0.1 Last edited by TECDOCREFERENCEDATAIMPORT on Fri Aug 10 13:36:50 EC
Approved	✘ Never Been Approved
Translation	Not Translated
Path	Classification 1 root/TecDoc/Linking Targets/HICOM/Commercial Vehicle HICOM/
Visibility	
ATTR_Country Bit String	abc
ATTR_Delete	123 0
Axle Configuration	4x2
ccm Technical	123 2771
Construction Year To	2004-12-31
Construction Year From	1994-01-01
Country Dependent Model De	abc
Country Exclusion	
Country Inclusion	
CV Body Type	Platform / Chassis
CV Short Name	abc HICOM Closed - MTB - MTB 145
CV Type Number	123 23420
Engine Type	Diesel
HP From	123 80

MotNr: Five digit number representing an Engine Number.

In the example below, MotNr 01670 is displayed in Workbench as the 'Engine Code' Object Type, '4JB1' (TD_ENGINE_CODE_1670).

Tree	4JB1 rev.0.1 - Classification	
<ul style="list-style-type: none"> ISUZU <ul style="list-style-type: none"> Axles ISUZU Commercial Vehicle ISUZU Engine ISUZU <ul style="list-style-type: none"> 4JB1 4JB1-T 4JB1-TC 4JG2 4JG2 T 4JG2 TC 4JH1-TC 4JH1-X 4JJ1-TC 4JJ1-TCS 4JJ1-TCX 4JJ1-X 4JK1E5S-L 4JK-1E5-TC 	Classification	Sub Products References Referenced By Images & Documents Tables
Description		
Name >> Value		
> ID	TD_ENGINE_CODE_1670	
> Name	4JB1	
> Object Type	Engine Code	
> Revision	0.1 Last edited by STEPSYS on Thu Apr 26 02:54:43 EDT 2018	
> Approved	✘ Never Been Approved	
> Translation	Not Translated	
> Path	Classification 1 root/TecDoc/Linking Targets/ISUZU/Engine ISUZU/4JB1	
> Visibility		
> ATTR_Country Bit String	abc	
> ATTR_Delete	123 0	
> Bore	123 093000	

In this example, because the Commercial Vehicle object with ID: 'TD_CV_TYPE_23420' is missing from the appropriate STEP classification hierarchy, and file 537 includes instruction to link the Engine Number to that object using the TD_Vehicle(CV)ToEngine link type, STEP displays a missing object error.

In short, object 'TD_CV_TYPE_23420' is missing from file 532 but the information associating it is present in file 537.

To resolve this, create the missing object and either import the Reference Data import file again, or manually link the object(s) to the engine code.

Missing Object Type When Allocating German KBA Vehicle Number

Errors may occur during the import of a TecDoc Data file. When errors occur, detailed error messages display within a BGP Execution Report. This section describes an error that can occur by listing an overview of the error, error message template, error message example, where the error messages can be viewed, and how best to address the error.

Important: The handling of validation errors can be configured by enabling / disabling the 'Continue on Error' parameter on the Validation state of each import workflow. By default, the 'Continue on Error' parameter is disabled. For more information, see the **Validation Error Handling** topic within the **Importing Automotive Data** section.

Overview

This type of error occurs when the import file tries to allocate a German KBA Number to a particular Passenger Car (PC) but the data within the table that imports the Passenger Car object (known in STEP as [Vehicle (PC)]) is missing.

When a user views the error message, this is apparent because the message template states: Missing object type for new object: <Object ID> of type: com.stibo.core.domain.Classification

When viewing the example error message below, it is clear that the Object that is missing is the Passenger car object with the Object ID 'TD_PC_TYPE_128235'.

Error Message Template

Error in this import [import file name] setting completed with errors - Error: Missing object type for new object: [<Object ID>] of type: com.stibo.core.domain.Classification

Error Message Example

Error in this import 22_KbaVehicleNumbers.xml setting completed with errors - Error: Missing object type for new object: TD_PC_TYPE_128235 of type: com.stibo.core.domain.Classification

Viewing the Error Message

The screenshot below displays this error message example within a Background Process Details Web UI screen.

Background Process Details

ID: BGP_102235

Started By: USERB

Description: Running background process ImportingFlowImportService from RunBackgroundProcessServiceAction

Template ID: ImportingFlowImportService

Status: Completed With Errors

Started: 7/26/18 11:50:24 AM

Finished: 7/26/18 12:17:08 PM

Elapsed: 26 m 44 s

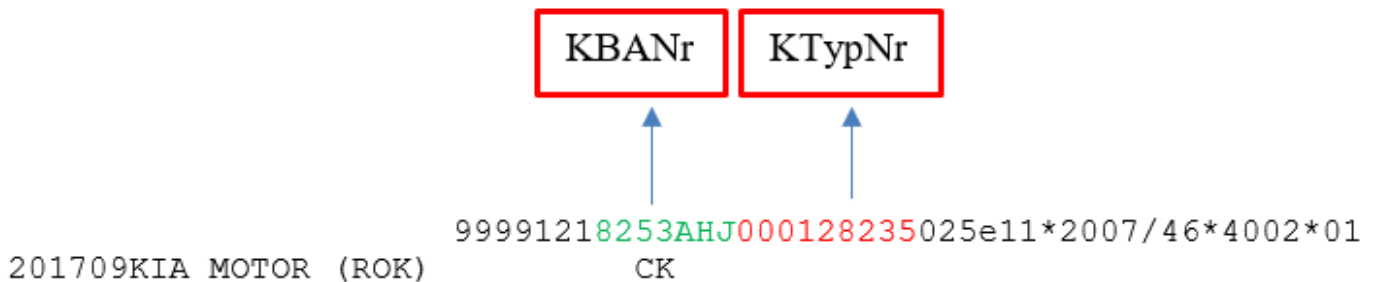
Select all Export

ID	Type	Text
<input type="checkbox"/> 27230	Error	Error in this import 22_KbaVehicleNumbers.xml setting completed with errors - Error: Missing object type for new object: TD_PC_TYPE_128235 of type: com.stibo.core.domain.Classification

Explanation

Because this error pertains to the Vehicle Type (PC) Object Type and allocation of KBA Numbers, the Passenger Car Types are listed within the TecDoc Reference Data file Data Table 120, and the allocation of KBA Numbers are listed within the data table 121, it is important to first understand Data Table 121.

For this explanation, the following example data from Data Table 121 is used. The Delta Keys that pertain to this error type are displayed with green or red text.



Understanding Data Table 120

Data Table 120 lists Passenger Car Types that are featured in Delta Key **KTypNr**.

Understanding Data Table 121

The following important Delta Keys are within Data Table 121:

KTypNr: Nine digit number representing the PC Vehicle Type number. Shown above in red text.

In the example below, KTypNr 000013972 displays in Workbench as the 'Vehicle Type (PC)' Object Type, 'LAND ROVER Open Off-Road Vehicle - DEFENDER Cabrio (L316) - 2.2' (TD_PC_Type_13972). Because KTypNr 0000128235 is not present in Data Table 121 the Vehicle Type (PC) with Object ID: TD_PC_Type_13972 is not created in STEP. The screenshot below is solely for the representational purpose of how an object with object type Vehicle Type (PC) displays in the workbench.

Description	
Name	Value
ID	TD_PC_Type_13972
Name	LAND ROVER Open Off-Road Vehicle - DEFENDER Cabrio (L316) - 2.2
Object Type	Vehicle Type (PC)
Revision	0.1 Last edited by STEPSYS on Thu Apr 26 09:39:00 EDT 2018
Approved	✘ Never Been Approved
Translation	Not Translated
Path	Classification 1 root/TecDoc/Linking Targets/LAND ROVER/Passenger
Visibility	
ATTR_ABS	abc
ATTR_ASR	abc
ATTR_Country Bit String	abc

KBANr: Seven digit number representing the German KBA Number. Shown above in green text.

In the example below, the first image within the screenshot displays an example of what a Passenger Car type Object would look like if it were present in Workbench. For this example, the 'LAND ROVER open off-road vehicle' object is referenced to the German KBA number '2143ABZ.' The second image in the screenshot displays the referenced German KBA Number '2143ABZ' within Workbench.

Reference Type	Target
> Body Types (PC) To Country	
> Type Description Country Deviation	
> Vehide (PC) To Engine	DT224
> Vehide (PC) To Synonym	4x4
> Vehide Country Deviation (PC)	
> Vehide Note Country Deviations	
> Vehide Type (PC) To KBA	2143ABZ

Name	Value
> ID	TD_KBAPCC_2143ABZ
> Name	2143ABZ
> Object Type	KBA Vehide (PC)
> Revision	0.1 Last edited by STEPSYS on Thu Apr 26 09:32:25 EDT 20
> Approved	✘ Never Been Approved
> Translation	Not Translated
> Path	Classification 1 root/TecDoc/Identification Keys/KBA Numbe
> Visibility	
> ABE Month	2012-09-01
> Body Type	Open Off-Road Vehide
> General Approval Number	abc e11*2007/46*0133*05
> KBA No	abc 2143ABZ

In this example, because the Passenger Car object with ID: 'TD_PC_TYPE_128235' (Data imported from Data Table 120) is missing from the appropriate STEP classification hierarchy, and the Data Table 121 has instruction to allocate the KBA Number to that object, STEP is displaying a missing object error.

If information on 'Passenger Car Vehicle primary type Number' is not present in Data Table 120 (KTypNr), but is present in Data Table 121(KTypNr), then STEP will throw a missing information error. In short, object 'TD_PC_TYPE_128235' is missing from Data Table 120 but the information associating it is present in file 121.

Resolution

To resolve this, create the missing object within the appropriate STEP classification hierarchy and import the Reference Data file again.

Validation Error When Importing LOV Values

Errors may occur during the import of a TecDoc Data file. When errors occur, detailed error messages display within a BGP Execution Report. This section describes an error that can occur by listing an overview of the error, error message template, error message example, where the error messages can be viewed, and how best to address the error.

Important: The handling of validation errors can be configured by enabling / disabling the 'Continue on Error' parameter on the Validation state of each import workflow. By default, the 'Continue on Error' parameter is disabled. For more information, see the **Validation Error Handling** topic within the **Importing Automotive Data** section.

Overview

This type of error occurs when an import file attempts to import values for a particular LOV but the values in the import file do not match the validation rules set by STEP. For more information about STEP validation rules, see the **Validation Rules** section of OLH.

When a user views the error message, this can be understood because the message template states: The value 'Value' isn't valid for LOV 'LOV ID': Length: 60 does not fit: 'Value'.

Error Message Template

Error in this import [import file name] setting completed with errors - Error: The value [imported LOV value] isn't valid for LOV [LOV ID]: Length: [Maximum Length value] does not fit: [imported LOV value].

Error Message Example

Error in this import 04_LOVDefinitions.xml setting completed with errors - Error: The value 'With Support Wheel Holder, pointing rearwards & also crossways' isn't valid for LOV 'TD_LOV-713': Length: 60 does not fit: With Support Wheel Holder, pointing rearwards & also crossways

Viewing the Error Message

The screenshot below displays this error message example within a Background Process Details Web UI screen.


Background Process Details

ID: BGP_102360

Started By: USERB

Description: Running background process ImportingFlowImportService from RunBackgroundProcessServiceAction


Template ID: ImportingFlowImportService

Status:  Completed With Errors

Started: 8/1/18 2:25:57 PM

Finished: 8/1/18 2:36:44 PM

Elapsed: 10 m 50 s

Select all
 Export

	ID	Type	Text
<input type="checkbox"/>	690	Error	Error in this import 04_LOVDefinitions.xml setting completed with errors - Error: The value 'With Support Wheel Holder, pointing rearwards & also crossways' isn't valid for LOV 'TD_LOV-713': Length: 60 does not fit: With Support Wheel Holder, pointing rearwards & also crossways

Explanation

Because this error pertains to LOV validation in STEP, and LOV Values are listed within the TecDoc Reference Data file Data Table 030, it is important to understand Data Table 030.

For this explanation, the following example data from a Reference Data file Data Table 030 is used. The Delta Key within the Reference Data file that pertains to this error type (Bez) is displayed with red text.

```
9999030000034616004With Support Wheel Holder,  
pointing rearwards & also crossways 0
```

Understanding Data Table 030

This section describes the important Delta Key within this Data Table.

Bez: Maximum 60 character text representing an LOV Value (also known as 'Descriptions' in the 'TecAlliance: TecDoc Data Format' file).

In the example below, LOV 'Overrun Brake Design' (TD_LOV-713) is configured to have a 'Maximum Length' of 60. Because the import file contains the value 'With Support Wheel Holder, pointing rearwards & also crossways' which has 62 characters, STEP displays a validation error, and the LOV Value is not imported into STEP. Therefore, it does not display within the 'Values' flipper.

System Setup

- Nitrogen generator operate
- NLGI Class
- Nonslip
- Number of Applications
- Number of cone steps
- Number of Gears
- Number of mounting speed
- Operating Mode
- Operating System
- Operation_Workshop Equip
- Overrun Brake Design**
- Paint
- Panel Type
- Park Assistant
- Particulate filter type
- Payload
- Permit
- Phases
- Pipe Connector
- Pipe Cutter Type
- Pipeline socket type_Comp
- Pit jack type
- Pit safety system type
- Pliers/Shears, characterist
- Plug housing design
- Plug Version
- Pockets/Cases, characteri
- Polish Design
- Power supply
- PREVIEW_SUPPL
- Price Unit
- Processing Equipment Typ
- Product Section
- Profile
- Prohibition Sign
- Properties (chem.)
- Protection Rating
- Pull Elements

List of Values | References | Log | State Log | Tasks

Description

Name	Value
ID	TD_LOV-713
Name	Overrun Brake Design
Edited by	2018-07-26 11:51:53 by TECDOCREFERENCEDATAIMPORT
Path	Lists of Values / LOVs/TecDoc LOV Group/TecDoc LOV/Overrun Brake Design
Dimension Dependencies	Language;
Use Ids on values	Yes
Use Ids for sorting	No
Value-ID Pattern	
ATTR_Delete	0
ID Type	A

In Attribute Groups

List of Values Validation

Name	Value
Validation Base Type	Text
Allow Users to Add Values	No
Mask	
Minimum Value	N/A
Maximum Value	N/A
Maximum Length	60

Values

Values	Value ID
For DIN Pulling Eye	22
For Drawbars	35
For Handbrake Lever	29
For Overrun Brake	34
For V-Drawbar	14
Handbrake Lever with Gas Spring	4
Reversing Lever, (can be relocated)	2
Reversing Lever, (cannot be relocated)	3
Tubular Design for Plugging in	19

Resolution

To resolve the error, edit the text within the description to be less than the defined character limit, or adjust the character limit in STEP.

Configuring TecDoc Reference Data Importer

The following topics provide the configuration steps necessary to allow users to be able to drag and drop TecDoc Reference Data files onto a configured File Loading Widget and monitor the progress of the import file using a configured Status Selector Homepage Widget and an Import Controller Screen.

- Configuring an IIEP for Reference Data Imports
- Configuring a File Loading Widget for Reference Data Imports

Prerequisites

It is expected that anyone configuring the TecDoc Reference Data Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, See the **Designer Access** topic within the **Web UI Getting Started** section of **STEP Online Help**.

Within this section users will be directed to view details for a Controller Entity within Web UI using a Node Details screen configured for Entities. Before this can be available to users, an Entity Details screen must be created.

Anyone configuring the TecDoc Reference Data Importer is expected to be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Configuring an IIEP for TecDoc Reference Data Imports

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

An Inbound Integration Endpoint (IIEP) can be configured in workbench to help automate the process of importing TecDoc Reference data into STEP. Once an IIEP is configured for TecDoc Reference data imports, reference data files can be imported after they are uploaded either to a configured hotfolder on an application server, or to a File Loading Widget on a Web UI Homepage. For more information, see the **TecDoc Reference Data Importer** topic.

This section describes how to configure an IIEP that can allow for the automated processing of TecDoc Reference data files. Each screenshot example within this section provides recommended values for the TecDoc Reference Data Importer and the parameters displayed.

Prerequisites

It is expected that anyone configuring an IIEP for use with a TecDoc Reference Data Import is familiar with the configuration and other processing of standard inbound integration endpoints. For more information, see the **Inbound Integration Endpoints** topic within the **Data Exchange** section of **STEP Online Help**.

Configuration Steps

1. Go to System Setup, select and then right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.
2. Once the Inbound Integration Endpoint Wizard displays, populate each parameter with values that best identify the IIEP. By default, all parameters display blank, and the following fields are mandatory: Endpoint ID, Endpoint Name, and User.

In the example below, a 'TecDoc Reference Data Import' user was created prior to configuring this IIEP. This is recommended to more easily track when this IIEP is responsible for changes to data.

The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' sidebar lists eight steps, with '1. Identify Endpoint' selected and highlighted in blue. The main area is titled 'Identify Endpoint' and contains four input fields: 'Endpoint ID' with the value 'TecDocReferenceInboundEndpoint', 'Endpoint Name' with the value 'TecDoc Reference Inbound Endpoint', 'Description' which is empty, and 'User' with the value 'TecDoc Reference Data Import (TECDOCREFERENCEDATAIMPORT)' and a dropdown arrow. At the bottom right, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

For more information about the parameters, see the **IIEP - Identify Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the Choose Receiver parameters will display. By default, the parameters are populated as recommended and shown below, except the Hotfolder parameter. This mandatory parameter must be populated with a hotfolder name before the Next button will enable. The value within this parameter will be used to create the new hotfolder on the application server, once the Finish button is clicked.

The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' panel lists eight steps: 1. Identify Endpoint, 2. Choose Receiver (highlighted), 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area is titled 'Choose Receiver' and contains the following fields: 'Receiver' (dropdown menu with 'Hotfolder Receiver' selected), 'Hotfolder' (text input field with 'TecDocReferenceInputFolder'), 'Keep file after load' (dropdown menu with 'Yes' selected), 'Ignore sub folders' (dropdown menu with 'No' selected), and 'In folder' (empty text input field). At the bottom right, there are four buttons: 'Back', 'Next' (highlighted with a dashed border), 'Finish', and 'Cancel'.

For more information about the parameters, see the **IIEP - Choose Receiver** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the Configure Endpoint parameters will display. By default, the parameters are populated as recommended and shown below, except the following:
 - Processing Engine
 - Maximum number of old processes
 - Maximum age of old processes

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
- 3. Configure Endpoint**
4. Configure PreProcessor
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Endpoint

Processing

Processing Engine: Import Flow Processor

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: 1000

Maximum number of old processes: 1000

Maximum age of old processes: 1y

Number of messages per background process: 1

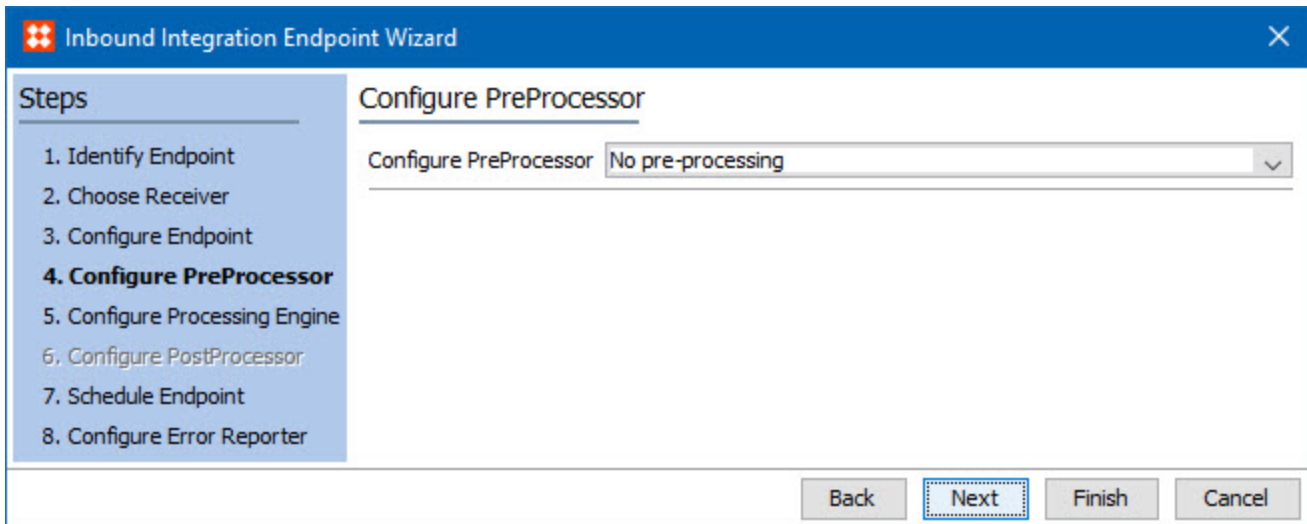
Buttons: Back, Next, Finish, Cancel

Important: The Import Flow Processor is only responsible for picking up files from the configured hotfolder, creating an Entity controller object, and updating the Entity with the Import Flow State BGP's and Status. The Import Flow Processor works with the Background Process Service for each workflow state to handle the import of a file. The IIEP / Import Flow Processor is NOT responsible for actually importing the content of the file, the BGP in each state does that processing. If a new Processing Engine is created through the Extension API, then it **cannot** be used with the automotive import framework.

- Click the dropdown option for the 'Processing Engine' parameter and select the **Import Flow Processor** option.
- Optionally, update the values for the 'Maximum number of old processes' and 'Maximum age of old processes' parameters to those shown above.

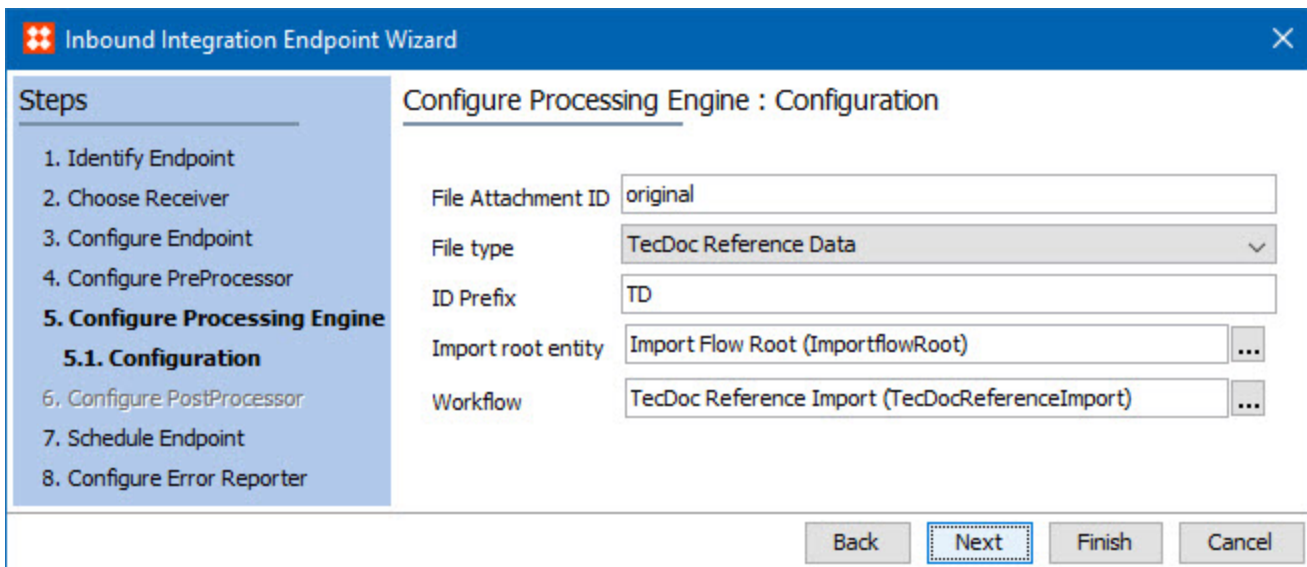
For more information about the parameters, see the **IIEP - Configure Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

5. Click the **Next** button, and the Configure PreProcessor parameter will display. By default, the parameter is populated as recommended and shown below.



For more information about the parameter, see the **IIEP - Configure PreProcessor** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the **Configure Processing Engine : Configuration** parameters for the Import Flow Processor will display. By default, only the File Attachment ID parameter is populated as shown below.



The prior **Configure Endpoint** step determines the options available for the **Configure Processing Engine** step.

- Click the 'File type' parameter, dropdown and select the **TecDoc Reference Data** option.
- Within the 'ID Prefix' parameter, enter a prefix value to easily identify import entities created by this IIEP.
- Click the ellipsis button (...) for the 'Import root entity' parameter, and select the 'Import Flow Root' node (or a desired root entity) This is the location where the import entities created by this IIEP will be stored.

- Click the ellipsis button (...) for the 'Workflow' parameter, and select the **TecDoc Reference Import** workflow or a desired workflow.

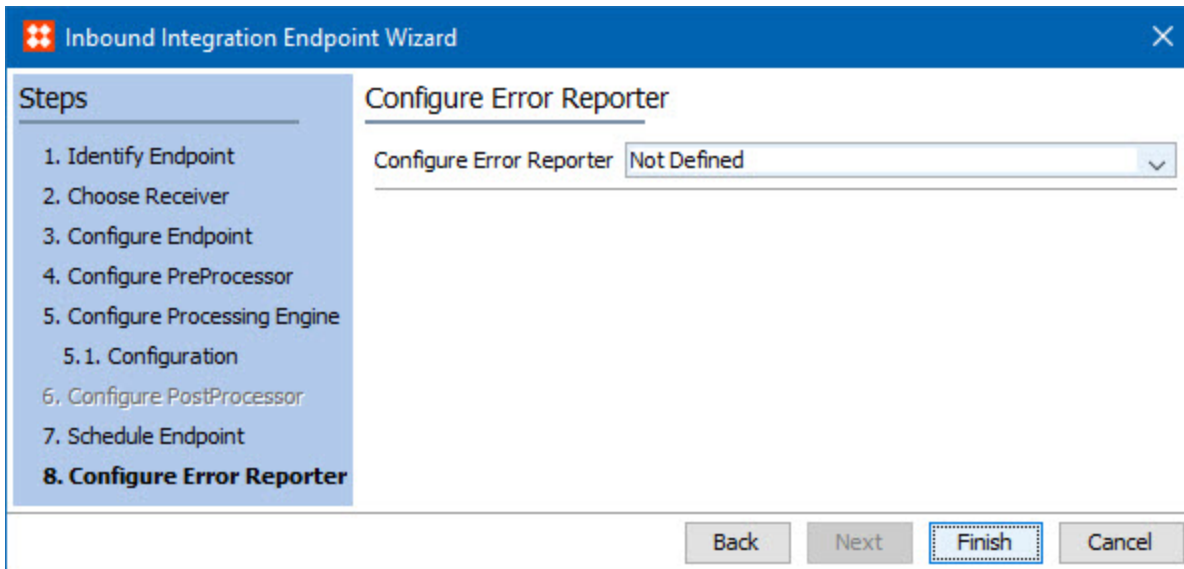
Note: It is possible to use a workflow that is not created by Easy Setup actions to handle an import file in a way that better fits an organizations needs. However, along with creating the workflow and selecting it within the Workflow parameter shown above, all the states for that workflow must be created along the processing steps of the file (i.e., Validation, Conversion, Import). Use of the Extension API is required to write the processing steps.

7. Click the **Next** button, and the Schedule Endpoint parameters will display. By default, 'Never' is selected. Optionally, update the values to those shown below.

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.

For more information about the parameters, see the **IIEP - Schedule Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

8. Click the **Next** button, and the Configure Error Reporter parameter will display. By default, the parameter is populated as recommended and shown below.



For more information about the parameter, see the **IIEP - Configure Error Reporter** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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

Important: An endpoint must be enabled before it can start processing data. For more information, see the **Running an Inbound Integration Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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

Configuring a File Loading Widget for Reference Data Imports

When configured, Web UI users can import TecDoc Reference Data files into STEP using a File Loading Widget. Users can also monitor the progress of Reference Data imports using a Status Selector Homepage Widget and a Node Details component.

Prerequisites

Before starting to configure the Web UI portion of this solution, an IIEP for a TecDoc Reference Importer must be configured within workbench. For more information, see the **Configuring an IIEP for Reference Data Imports** topic.

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

Configuration

Each screenshot example within this section provides recommended values for the TecDoc Reference Data Importer and the parameters displayed.

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

Note: If Easy Setup actions for the TecDoc solution have been completed as described in the **3. Run Easy Setup of Standards** topic of **Automotive Quick Start Guide**, then the 'TECDOC IMPORTS' File Loading Widget will automatically be added to the TecDoc Web UI Homepage as shown in the examples below. Otherwise, the steps below can be used to complete configuration.

1. Go to the Web UI Homepage where a File Loading Widget configured with the 'TECDOC IMPORTS' title is configured with a 'Reference' drop area available for users to drag and drop supplier import files.
2. Using Web UI design mode, select an existing File Loading Widget to be used or add a new File Loading Widget to the Homepage Widget Grid component. For more information, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.
3. Go to the Inbound Integration Endpoint Parameters field, click the **Add** button, and the Inbound Integration Endpoint Parameter Properties dialog will display.
4. Click the dropdown for the Inbound Integration Endpoint parameter, and select **TecDoc Reference Inbound Endpoint** (the IIEP created for TecDoc Reference imports).

Edit component
✕

Inbound Integration Endpoint Parameter Properties

Inbound Integration *
Endpoint

Label

TecDoc Reference Inbound Endpoint ▾

- AutoCare ACES Application Inbound Endpoint
- AutoCare Brand Inbound Endpoint
- AutoCare PAdb Inbound Endpoint
- AutoCare PCdb Inbound Endpoint
- AutoCare PIES Inbound Endpoint
- AutoCare Qdb Inbound Endpoint
- AutoCare VCdb Inbound Endpoint
- NAPA Application Inbound Endpoint
- NAPA Attribute Inbound Endpoint
- NAPA Interchange Inbound Endpoint
- NAPA Translation Inbound Endpoint
- NAPA Vehicle Inbound Endpoint
- TecDoc Reference Inbound Endpoint
- TecDoc Supplier Inbound Endpoint

✓ Save

✕ Cancel

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

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

In the example below:

- A File Loading Widget labeled as 'TECDOC IMPORTS' is displayed above its configurations.
- The File Loading Widget and its configurations are shown with the default configurations provided automatically when Easy Setup actions for the TecDoc component are completed.
- An IIEP for TecDoc Reference data imports is added within the same File Loading Widget as the IIEP for the TecDoc Supplier data imports.



File Loading Widget Properties

Inbound Integration Endpoint Parameters

- Inbound Integration Endpoint Parameter (Reference / TecDocReferenceInboundEndpoint)
- Inbound Integration Endpoint Parameter (Supplier / TecDocSupplierInboundEndpoint)

Add... Edit... Remove Up Down

Label: TECDOC IMPORTS

Child Components

Edit component

Inbound Integration Endpoint Parameter Properties

Inbound Integration Endpoint * TecDoc Reference Inbound Endpoint

Label Reference

Save Cancel

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

TecDoc Supplier Data Importer

The intention of the Automotive TecDoc Supplier Data Importer is to provide an out-of-the-box solution for importing data included within a supported TecDoc Supplier Data file. Because only supported versions will successfully upload, before attempting to upload a TecDoc Supplier Data file, confirm the file version being uploaded is listed within the **Supported Versions and Formats** topic.

Only after a TecDoc Reference Data file has been successfully imported into STEP, can TecDoc Supplier Data files *of the same version* be imported into STEP. For more information, see the **TecDoc Reference Data Importer** topic.

A Supplier Data file can be imported into STEP by uploading it to either a configured hotfolder on an application server, or through a File Loading Widget on a Web UI Homepage.

This section includes information on:

- Using TecDoc Supplier Data Importer
- Configuring TecDoc Supplier Data Importer

Prerequisites

It is expected that anyone using and/or configuring the TecDoc Supplier Data Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Important: To successfully import a TecDoc Supplier data import file, the file version must match the version of the last successfully imported Reference Data file.

Using TecDoc Supplier Data Importer

A TecDoc Supplier Data file can be imported into STEP by uploading it to either a configured hotfolder on an application server, or through a File Loading Widget on a Web UI Homepage.

The following information is available for the TecDoc Supplier Data Importer.

- Importing Supplier Data Files via Web UI

Prerequisites

It is expected that anyone using and/or configuring the TecDoc Supplier Data Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For more information about uploading files to an application server, see the **File Loading** section of the **Automotive Quick Start Guide** found within the **Solution Enablement** section of **STEP Online Help**.

For more information about using Workbench to monitor and process imports, see the **Workflows** section of **STEP Online Help**.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Importing Supplier Data Files via Web UI

The intention of the Automotive TecDoc Supplier Data Importer Web UI setup included within this topic is to provide an out-of-the-box solution for importing data included within a supported TecDoc Supplier file. Because only supported versions will successfully upload, before attempting to upload a TecDoc Supplier Data file, confirm the file version being uploaded is listed within the **Supported Versions and Formats** topic.

A TecDoc Supplier Data file includes information on supplier part data.

Prerequisites

Within this section users will be directed to view details for a Controller Entity within Web UI using a Node Details screen configured for Entities. Before this can be available to users, an Entity Details screen must be created.

To easily start and monitor a TecDoc Supplier Data Import, recommended practice is to configure a Web UI Import Controller screen specific to the TecDoc Supplier Data Importer. When the Easy Setup actions for the TecDoc standard are completed, the TecDoc Import Controller screen is automatically created and configured for use.

It is expected that anyone using and/or configuring the TecDoc Supplier Data Importer be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

If the Easy Setup actions for the TecDoc Component model have been completed, then the functionality explained within this topic should be available. Otherwise, configuration is necessary. For more information, see the **TecDoc Supplier Data Importer** topic.

Import Process Overview

Once a valid automotive data file is uploaded to a hotfolder on the application server (optionally using a File Loading Widget), the file is picked up from the hotfolder by an IIEP, and the IIEP creates an Entity in STEP that represents the file. This Entity object is called the 'controller' and contains basic data about the file and the file's status in the workflow. Web UI users are able to monitor the import status using a Status Selector Widget and an Import Controller Screen. Once an import file reaches the 'Ready for Import' state, then users can start the import by clicking on the 'Start import' button within the Control Panel screen. At that time, the BGP service (that runs as part of the Import state) allows for the configured business rules to act on the objects being imported.

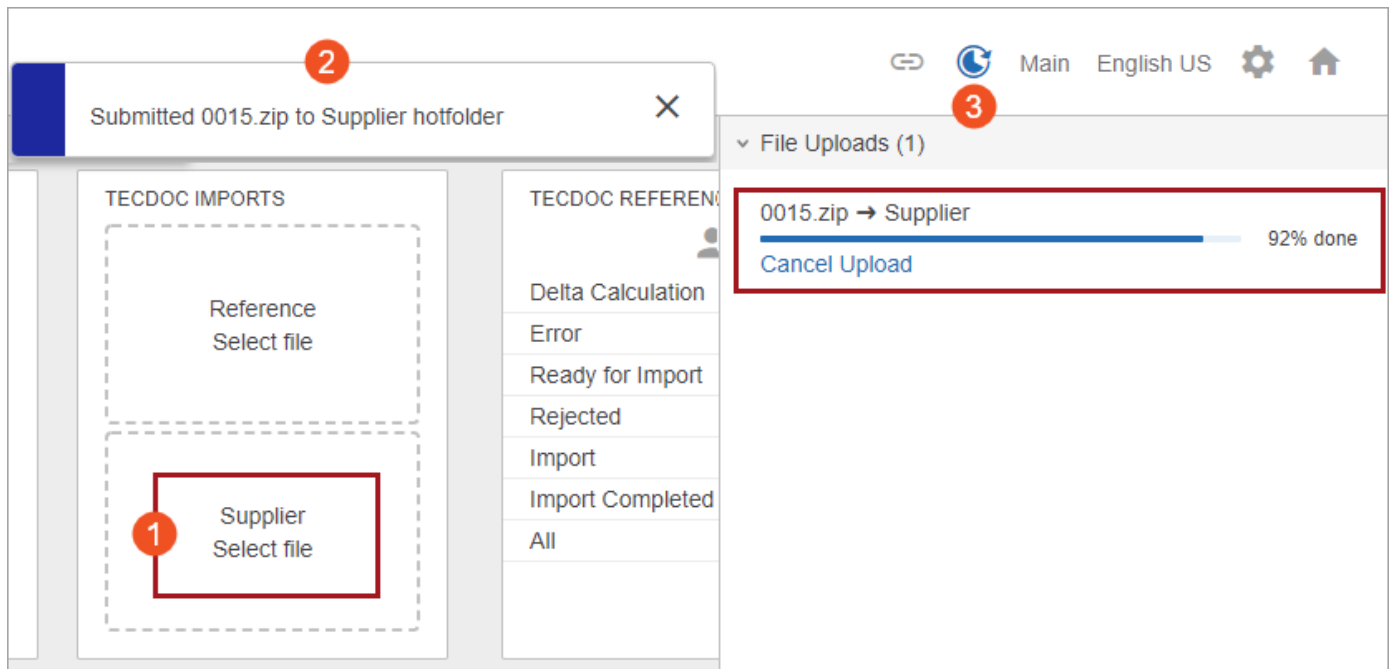
Using the TecDoc Supplier Data Importer in Web UI

1. Access the TecDoc Web UI Homepage and upload a valid TecDoc Supplier Data file to the application server hotfolder (root/upload/hotfolders/TecDocSupplierInputFolder), using the 'Supplier' drop zone of the 'TECDOC IMPORTS' File Loading Widget.

For more information, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

2. Once the upload has successfully started, a notification dialog will display at the top of the screen referencing the name of the uploaded file. In the example below, the dialog displays 'Submitted 0015.zip to Supplier hotfolder' (the name of the file being uploaded is '0015').
3. Optionally, users can view the progress of the upload using the Background Process Notification Component.

In the example below, the BGP Notification Panel is expanded and the file, '0015.zip' is 92% processed.



For more information on using the BGP Notification Component and side panel, see the **Background Process Notification Component** topic within the **Main Properties Overview** section of **STEP Online Help**.

Once created, the Controller is initiated into the workflow associated with the importer (TecDoc Supplier Import). From there, the TecDoc Supplier Import workflow takes over processing of the file via a series of states using business rules and background processes to carry out the processing of the file. Each import has an associated workflow and all proceed through the same states by default. However, it is intended that customers will expand on the existing states and actions to add their own validations, reporting, and additional processing as needed.

Important: It is critical to understand that it is only the Controller Entity that is in the workflow - the objects being acted on (created / updated / deleted) via information supplied in the import file are not in the workflow. Therefore, running standard business actions acting on current object will impact the Controller Entity only, not the objects in the input file. In order for the Business Action to act on the objects being imported, the Business Action must be added to the 'Import action' parameter on the 'Import' state. For more information, see the **Import State** topic within the **Default Workflow States and Functions** section.

4. If an Entity Details screen has been configured, then when the IIEP creates the Controller Entity object, users can view the Controller and its important information by navigating within the Tree side panel to the Import Flow Root > TecDoc Supplier. Otherwise, if a user clicks on an Entity in the Tree side panel, an error will display indicating the configuration is invalid.

Note: If an error message displays when selecting an Entity within a Web UI, then the Entity Details screen has not been configured.

In the example below, the Tree side panel is expanded, and the '0015.zip' Controller is displayed below the 'TecDoc Supplier' Entity. When the Controller is selected, many details (like the Controller ID 'Controller-101762') can be viewed within the Node Details screen.

Note: In the screenshot below, the 'Enable Tag Conversion' parameter within the Attribute Value Group Component is disabled.

Node Details

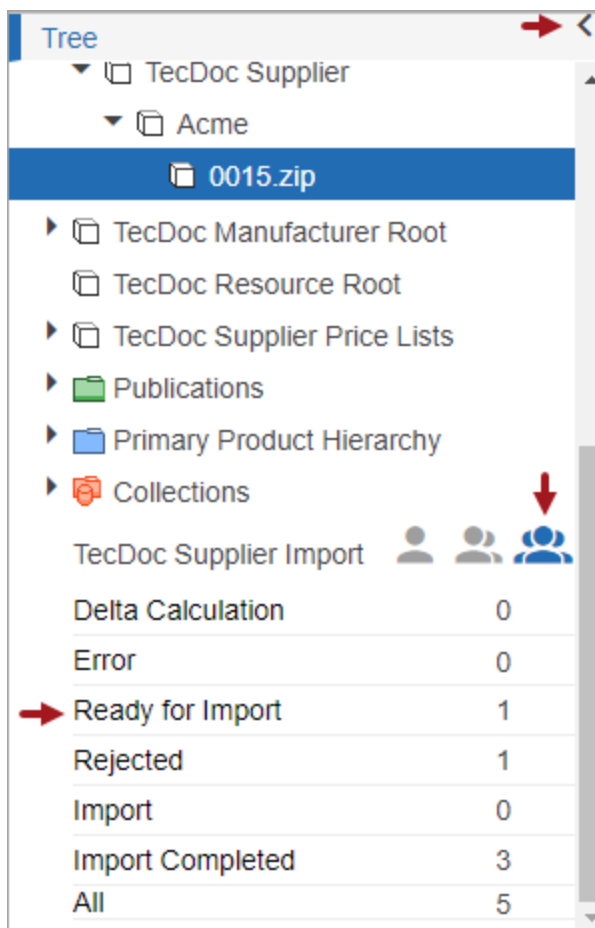
ID	Controller-101762
Name	0015.zip
Object Type	Import Flow Controller Type
Revision	Date: 2018.07.03 Revision number: 0.1 Last Edited by: TECDOCSUPPLIERDATAIMPORT Time: 16:49:51
Automotive Import Flow State BGP	<?xml version="1.0" encoding="UTF-8"?> <StringMap> <Entry Key="Validation" Value="BGP_101763"/> <Entry Key="Import" Value="BGP_101766"/> <Entry Key="Conversion" Value="BGP_101764"/> <Entry Key="DeltaCalculate" Value="BGP_101765"/> </StringMap>
Import Flow Endpoint ID	TecDocSupplierInboundEndpoint
Import Flow File Type	TecDocSupplierData
Import Flow Overall Status	Completed with errors: 34
Import Flow State Status	<?xml version="1.0" encoding="UTF-8"?> <StringMap> <Entry Key="Validation" Value="Validation completed"/> <Entry Key="Import" Value="Completed with errors: 34"/> <Entry Key="Conversion" Value="Conversion completed"/> <Entry Key="DeltaCalculate" Value="Delta calculation comp"/> </StringMap>
Import Flow Workflow ID	TecDocSupplierImport

- **Name:** STEP Name of the Controller entity. This is generated from the uploaded file original name.
- **Automotive Import Flow State BGP:** Attribute used to store the IDs of the background processes.
- **Import Flow State Status:** Attribute used to store and display the status of each process (rather than the overall status).

- **Import Flow Overall Status:** Attribute used to store and display the global status of the file (rather than the process specific status).
5. As the file is uploaded, and the Entity Controller moves through the TecDoc Supplier Import workflow, users can monitor the progress using the 'TECDOC SUPPLIER IMPORT' Status Selector Widget on the Web UI homepage and/or the left side panel.

In the example below, the Tree panel is expanded, and the **Triple user button** is selected so that all items assigned to any user are displayed. Notice that one file is in the 'Ready for Import' state, one file is in the Rejected state, and three files are in the 'Import Completed' state, for a total of five files.

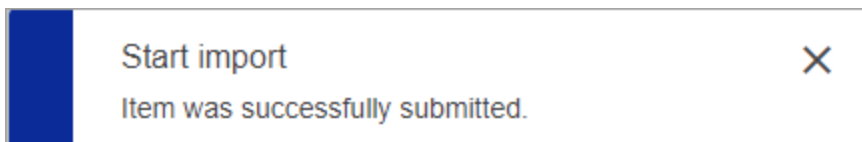
For more information, see the **Moving Tasks Trough a Workflow in Web UI** topic within **STEP Online Help**.



6. Clicking on the **Ready for Import** state within the Status Selector homepage and/or the left side panel widget will navigate the user to the TecDoc Supplier Data Import Controller Screen with only those Controllers in the 'Ready for Import' state displayed.
7. Clicking the Controller icon displays the Import Details below the Controller list.

In the example below, Controller '0015.zip' (ID 'Controller-101762') with an overall status of 'Done creating delta file' is selected and the Import Details are displayed below the Controller list.

8. Optionally, before starting the import, clicking on the 'completedwitherrors' Background Process Link will display the Background Process Details screen where users can view or download the details of the validation process.
9. Click the **Reject** button to move the Controller to the Rejected state. For more information, see the **Rejected State** topic within the **Import Framework** section.
10. Click the **Start import** button, the Start import dialog (shown below) displays at the top of the screen, and the Controller is moved to the Import state of the workflow. For more information, see the **Import State** topic within the **Import Framework** section.



Importing a TecDoc Supplier Data file can take several minutes, but the import progress can be monitored using the 'TECDOC SUPPLIER IMPORT' Status Selector Widget on the Web UI homepage and/or the left side panel, or the Import Controller Screen's 'Status' column.

In the example below, the TecDoc Supplier Data Import Controller Screen displays the Overall Status and the Import process status as 'Importing files...' Additionally, the Background Process Link column displays a link to the Background Process Details screen as 'running.'

TecDoc Supplier Data Import Controller Screen

Process	File Name	Overall Status
Controller-101762	0015.zip	Importing files...

Import Details

Process	Started Time	Duration	Started By	Status	Background Process Link
Validation	2018-08-08 16:26:12	28 secs	STEPSYS	Validation completed	completedwitherrors
Conversion	2018-08-08 16:26:40	2 mins 19 secs	STEPSYS	Conversion completed	succeeded
Delta Calculation	2018-08-08 16:28:59	3 mins 28 secs	STEPSYS	Delta calculation completed	succeeded
Import	2018-08-10 12:37:44	1 hr 6 mins 44 secs	STEPSYS	Importing files...	running

1-4 of 4

- Optionally, click the 'running' Background Process Link (shown above) to view the details of the import process within the Background Process Details screen.

Once the import has completed, the Controller is automatically moved to the Import Completed state. For more information, see the **Import Completed State** topic within the **Import Framework** section.

For more information on automotive workflow states, see the **Default Workflow States and Functions** topic.

Configuring TecDoc Supplier Data Importer

The following topics provide the configuration steps necessary to allow users to be able to drag and drop TecDoc Supplier Data files onto a configured File Loading Widget and monitor the progress of the import file using a configured Status Selector Homepage Widget and an Import Controller Screen.

- Configuring an IIEP for Supplier Data Imports
- Configuring a File Loading Widget for Supplier Data Imports

Prerequisites

It is expected that anyone configuring the TecDoc Supplier Data Import solution within a Web UI be familiar with the Web UI Designer, as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. For more information, See the **Designer Access** topic within the **Web UI Getting Started** section of **STEP Online Help**.

Within this section users will be directed to view details for a Controller Entity within Web UI using a Node Details screen configured for Entities. Before this can be available to users, an Entity Details screen must be created.

Anyone configuring the TecDoc Supplier Data Importer is expected to be familiar with the **Importing Automotive Data** section, as basic concepts for working with an automotive importer are not covered in this section.

For general information about the File Loading Widget and Status Selector Widget, including additional information about working with these widgets, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**, and the **Status Selector Homepage Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.

Configuring an IIEP for Supplier Data Imports

An Inbound Integration Endpoint (IIEP) can be configured in workbench to help automate the process of importing TecDoc Supplier data into STEP. Once an IIEP is configured for TecDoc Supplier data imports, supplier data files can be imported once they are uploaded either to a configured hotfolder on an application server, or to a File Loading Widget on a Web UI Homepage. For more information, see the **TecDoc Supplier Data Importer** topic.

This section describes how to configure an IIEP that can allow for the automated processing of TecDoc Supplier data files. Each screenshot example within this section provides recommended values for the TecDoc Supplier Data Importer and the parameters displayed.

Prerequisites

It is expected that anyone configuring an IIEP for use with a TecDoc Supplier Data Import is familiar with the configuration and other processing of standard inbound integration endpoints. For more information, see the **Inbound Integration Endpoints** topic within the **Data Exchange** section of **STEP Online Help**.

Configuration Steps

1. Go to System Setup, select and then right-click the **Inbound Integrations Endpoints** setup group, and click **Create Inbound Integration Endpoint**.
2. Once the Inbound Integration Endpoint Wizard displays, populate each parameter with values that best identify the IIEP. By default, all parameters display blank, and the following fields are mandatory: Endpoint ID, Endpoint Name, and User.

In the example below, a 'TecDoc Supplier Data Import' user was created prior to configuring this IIEP. This is recommended to more easily track when this IIEP is responsible for changes to data.

The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' sidebar lists eight steps: 1. Identify Endpoint (highlighted), 2. Choose Receiver, 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area is titled 'Identify Endpoint' and contains four input fields: 'Endpoint ID' with the value 'TecDocSupplierInboundEndpoint', 'Endpoint Name' with 'TecDoc Supplier Inbound Endpoint', 'Description' (empty), and 'User' with 'TecDoc Supplier Data Import (TECDOSUPPLIERDATAIMPORT)' and a dropdown arrow. At the bottom right, there are four buttons: 'Back', 'Next', 'Finish', and 'Cancel'.

For more information about the parameters, see the **IIEP - Identify Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the Choose Receiver parameters will display. By default, the parameters are populated as recommended and shown below, except the Hotfolder parameter. This mandatory parameter must be populated with a hotfolder name before the Next button will enable. The value within this parameter will be used to create the new hotfolder on the application server, once the Finish button is clicked.

The screenshot shows the 'Inbound Integration Endpoint Wizard' window. On the left, a 'Steps' panel lists: 1. Identify Endpoint, 2. Choose Receiver (highlighted), 3. Configure Endpoint, 4. Configure PreProcessor, 5. Configure Processing Engine, 6. Configure PostProcessor, 7. Schedule Endpoint, and 8. Configure Error Reporter. The main area is titled 'Choose Receiver' and contains the following fields: 'Receiver' (dropdown menu showing 'Hotfolder Receiver'), 'Hotfolder' (text input field containing 'TecDocSupplierInputFolder'), 'Keep file after load' (dropdown menu showing 'Yes'), 'Ignore sub folders' (dropdown menu showing 'No'), and 'In folder' (empty text input field). At the bottom right, there are four buttons: 'Back', 'Next' (which has a dashed border indicating it is the active step), 'Finish', and 'Cancel'.

For more information about the parameters, see the **IIEP - Choose Receiver** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the Configure Endpoint parameters will display. By default, the parameters are populated as recommended and shown below, except the following:
 - Processing Engine
 - Maximum number of old processes
 - Maximum age of old processes

Inbound Integration Endpoint Wizard

Steps

1. Identify Endpoint
2. Choose Receiver
- 3. Configure Endpoint**
4. Configure PreProcessor
5. Configure Processing Engine
6. Configure PostProcessor
7. Schedule Endpoint
8. Configure Error Reporter

Configure Endpoint

Processing

Processing Engine: Import Flow Processor

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: 1000

Maximum number of old processes: 1000

Maximum age of old processes: 1y

Number of messages per background process: 1

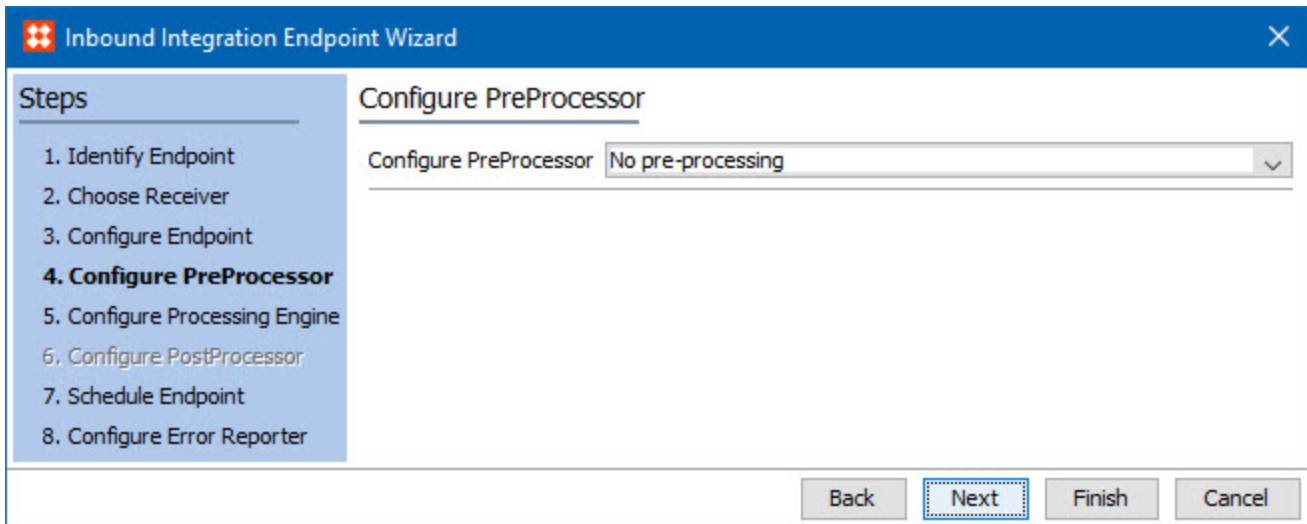
Buttons: Back, Next, Finish, Cancel

Important: The Import Flow Processor is only responsible for picking up files from the configured hotfolder, creating an Entity controller object, and updating the Entity with the Import Flow State BGP's and Status. The Import Flow Processor works with the Background Process Service for each workflow state to handle the import of a file. The IIEP / Import Flow Processor is NOT responsible for actually importing the content of the file, the BGP in each state does that processing. If a new Processing Engine is created through the Extension API, then it **cannot** be used with the automotive import framework.

- Click the dropdown option for the 'Processing Engine' parameter and select the **Import Flow Processor** option.
- Optionally, update the values for the 'Maximum number of old processes' and 'Maximum age of old processes' parameters to those shown above.

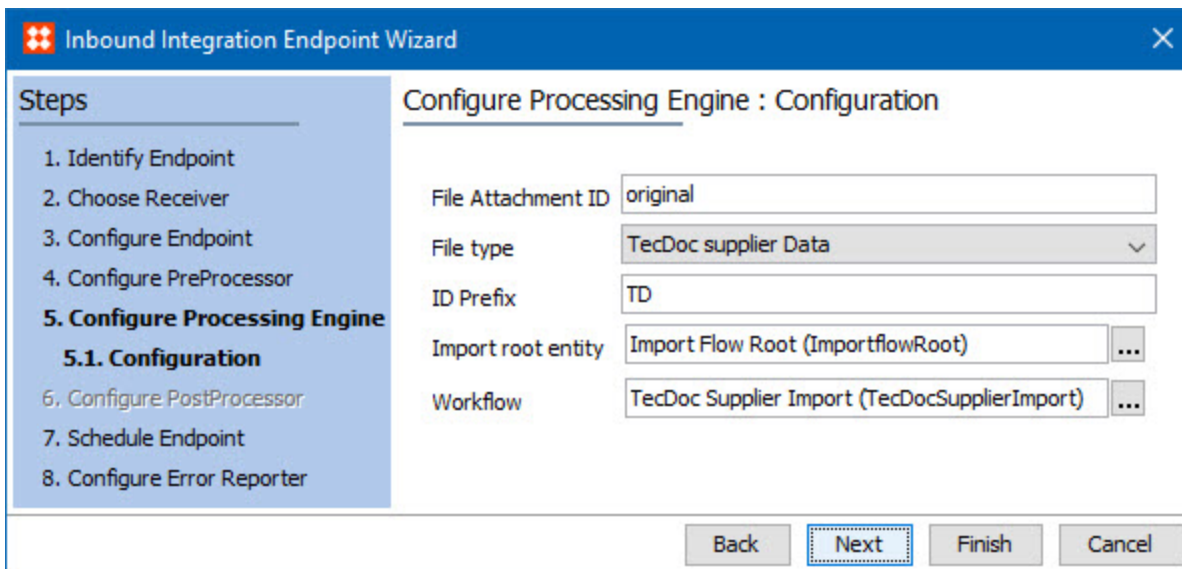
For more information about the parameters, see the **IIEP - Configure Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

5. Click the **Next** button, and the Configure PreProcessor parameter will display. By default, the parameter is populated as recommended and shown below.



For more information about the parameter, see the **IIEP - Configure PreProcessor** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

- Click the **Next** button, and the **Configure Processing Engine : Configuration** parameters for the Import Flow Processor will display. By default, only the File Attachment ID parameter is populated as recommended and shown below.



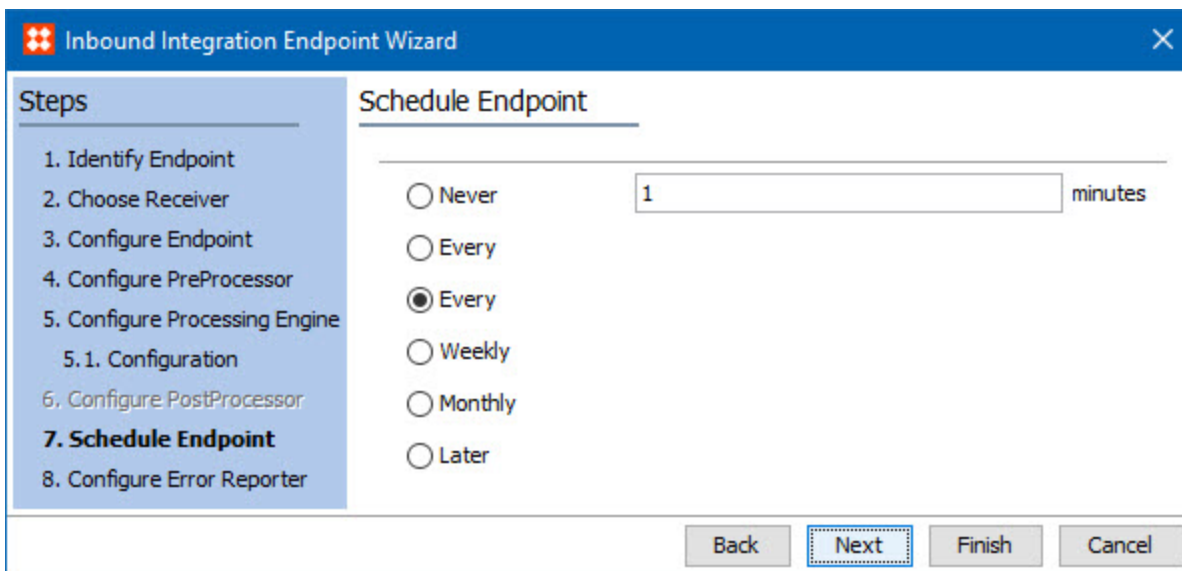
The prior **Configure Endpoint** step determines the options available for the **Configure Processing Engine** step.

- Click the 'File type' parameter dropdown and select the **TecDoc supplier Data** option.
- Within the 'ID Prefix' parameter, enter a prefix value to easily identify import entities created by this IIEP.
- Click the ellipsis button (...) for the 'Import root entity' parameter, and select a desired root entity location to store the import entities created by this IIEP.

- Click the ellipsis button (...) for the 'Workflow' parameter, and select the **TecDoc Supplier Import** workflow or a desired workflow.

Note: It is possible to use a workflow that is not created by Easy Setup actions to handle an import file in a way that better fits an organizations needs. However, along with creating the workflow and selecting it within the Workflow parameter shown above, all the states for that workflow must be created along the processing steps of the file (i.e., Validation, Conversion, Import). Use of the Extension API is required to write the processing steps.

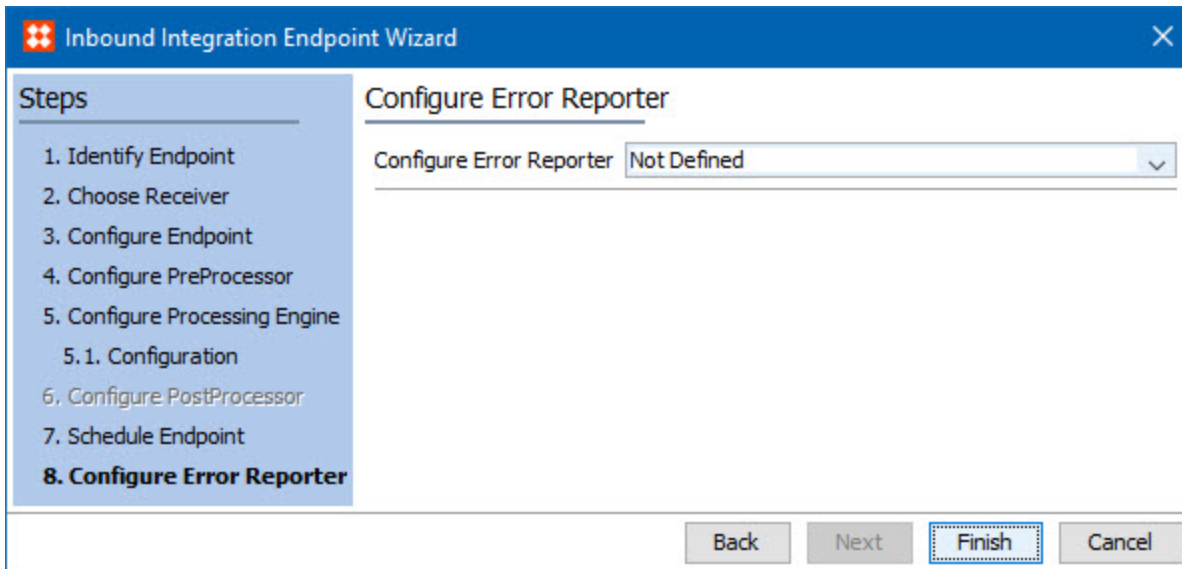
7. Click the **Next** button, and the Schedule Endpoint parameters will display. By default, 'Never' is selected. Optionally, update the values to those shown below.



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.

For more information about the parameters, see the **IIEP - Schedule Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

8. Click the **Next** button, and the Configure Error Reporter parameter will display. By default, the parameter is populated as recommended and shown below.



For more information about the parameter, see the **IIEP - Configure Error Reporter** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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

Important: An endpoint must be enabled before it can start processing data. For more information, see the **Running an Inbound Integration Endpoint** topic within the **Inbound Integration Endpoint** section of **STEP Online Help**.

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

Configuring a File Loading Widget for Supplier Data Imports

When configured, Web UI users can import TecDoc Supplier Data files into STEP using a File Loading Widget. Users can also monitor the progress of Supplier Data imports using a Status Selector Homepage Widget and a Node Details component.

Prerequisites

Before starting to configure the Web UI portion of this solution, an IIEP for TecDoc Supplier data imports must be configured within workbench. For more information, see the **Configuring an IIEP for Supplier Data Imports** topic.

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

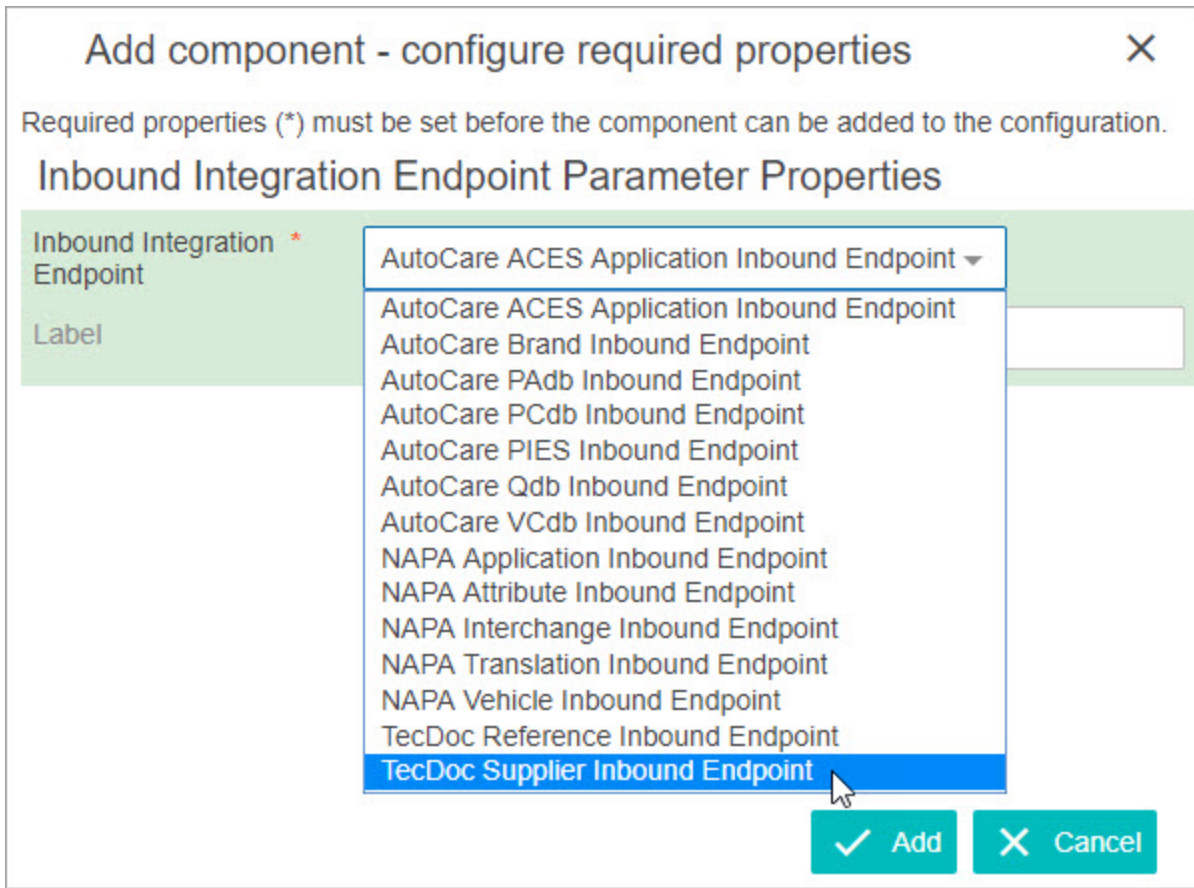
Configuration

Each screenshot example within this section provides recommended values for the TecDoc Supplier Data Importer and the parameters displayed.

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

Note: If Easy Setup actions for the TecDoc solution have been completed as described in the **3. Run Easy Setup of Standards** topic of **Automotive Quick Start Guide**, then the 'TECDOC IMPORTS' File Loading Widget will automatically be added to the TecDoc Web UI Homepage as shown in the examples below. Otherwise, the steps below can be used to complete configuration.

1. Go to the Web UI Homepage where a File Loading Widget configured with the 'TECDOC IMPORTS' title is configured with a 'Supplier' drop area available for users to drag and drop supplier import files.
2. Using Web UI design mode, select the existing File Loading Widget to be used or add a new File Loading Widget to the Homepage Widget Grid component. For more information, see the **File Loading Widget** topic within the **Web User Interfaces** section of **STEP Online Help**.
3. Go to the Inbound Integration Endpoint Parameters field, click the **Add** button, and the Inbound Integration Endpoint Parameters Properties dialog will display.
4. Click the dropdown for the Inbound Integration Endpoint parameter, and select **TecDoc Supplier Inbound Endpoint** (the IIEP created for TecDoc Supplier imports).



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

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

In the example below:

- A File Loading Widget labeled as 'TECDOC IMPORTS' is displayed above its configurations.
- The File Loading Widget and its configurations are shown with the default configurations provided automatically when Easy Setup actions for the TecDoc component model are completed.
- An IIEP for both the TecDoc Supplier and Reference importers is added within the 'TECDOC IMPORTS' File Loading Widget.



File Loading Widget Properties

Inbound Integration Endpoint Parameters

- Inbound Integration Endpoint Parameter (Reference / TecDocReferenceInboundEndp
- Inbound Integration Endpoint Parameter (Supplier / TecDocSupplierInboundEndpoint)

Add... Edit... Remove Up Down

Label: TECDOC IMPORTS

Edit component

Inbound Integration Endpoint Parameter Properties

Inbound Integration Endpoint: TecDoc Supplier Inbound Endpoint

Label: Supplier

Save Cancel

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

Exporting Automotive Data

The Automotive solution provides extensive export capabilities. For information on the supported versions, see the **Supported Versions and Formats** topic within this guide. For information on the way a user initiates and manages exports, see the **Using Automotive Exporters** section of the **Automotive Quick Start Guide**.

This section addresses the following available automotive exporters:

- AutoCare ACES Application Exporter
- AutoCare PIES Exporter
- NAPA Application Exporter

AutoCare ACES Application Exporter

The AutoCare ACES Application Exporter is used to export application data in ACES format. ACES versions 3.0, 3.2 and 4.0 are supported. The exporter requires that applications be stored in the standard AutoCare data model.

Prerequisites

The ACES exporter runs using standard Export Manager functionality. Only information specific to the ACES exporter is covered within this guide. Additional information on general Export Manager functionality is covered in the **Export Manager** topic within the **Data Exchange** section of **STEP Online Help**.

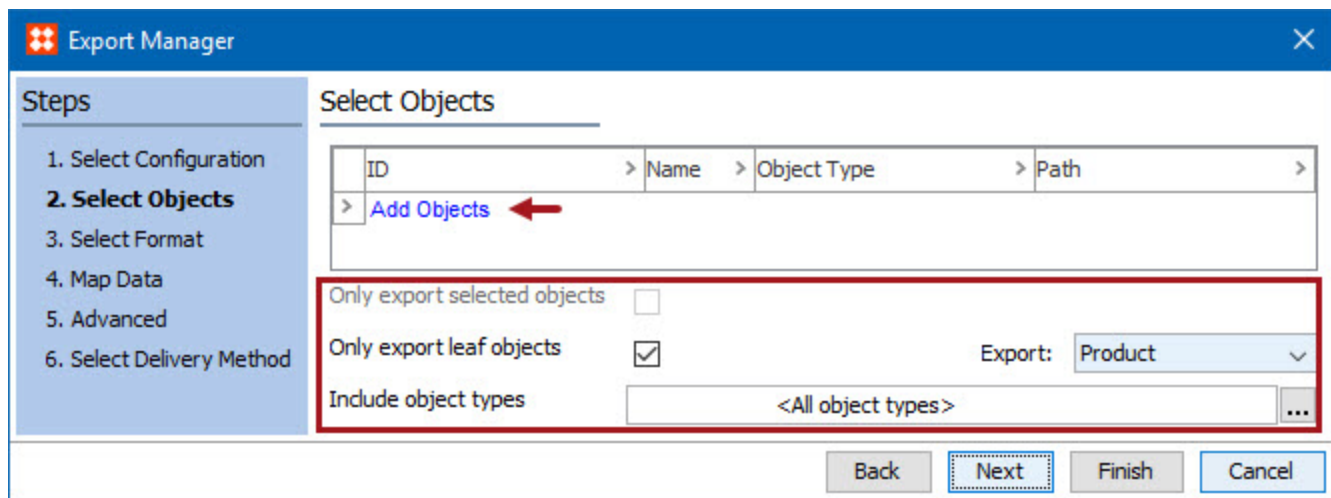
Using ACES Application Exporter

Step 1. Start ACES Application Exporter

The AutoCare ACES Application Exporter can be started via standard Export Manager methods, e.g., File > Export > Data, or by selecting an object in Tree, right-clicking, and selecting 'Export Data Below' from the menu. In the case of the right-click option, an object in the AutoCare PIES Product hierarchy or a yellow classification where the applications are linked to must be selected.

Step 2. Select Objects

In the Select Objects dialog, the only parameter that is read by the exporter is the actual object selection. The additional parameters (shown within the red box in the screenshot below) are disregarded.



Click the **Add Objects** link to add one or more blue folders in the AutoCare PIES Products hierarchy, or one or more classification folders.

All ACES Application objects beneath the selection(s) (either as child objects or via links) will result in the export file.

Step 3. Select Format

Select **AutoCare ACES Application Exporter** format to expose options specific to the ACES exporter. The version will default to 4.0, but any one of the available ACES export versions can be selected from the dropdown list.

Note: Use of the 'Approved For' parameter requires manual setup, which is covered in the **10. Add Country Codes to Approved for LOV (ACES Export Only)** topic within the **Automotive Quick Start Guide**.

Required Parameters

This section describes the required parameters that display in bold and require manual data population. The Next button will not enable until the required parameters are populated with at least one character. By default, the required parameter values are blank unless an existing configuration is selected.

- **Company:** Allows a company to be specified for the exported data within the free text field. The value (along with the 'Document Title' value) is used to create the ZIP file name and will display within the ACES export between the <Company> tags. For example, if the value 'ACME' is provided, then the exported ACES file will display as shown below.

```
<Company>ACME</Company>
```

- **Sender Name:** Allows a sender name to be specified for the exported data within the free text field. The value will display within the ACES export between the <SenderName> tags. For example, if the value 'Grace' is provided, then the exported ACES file will display as shown below.

```
<SenderName>Grace</SenderName>
```

- **Sender Phone:** Allows a sender phone to be specified for the exported data. The value will display within the ACES export between the <SenderPhone> tags. For example, if the value '770-123-1234' is provided, then the exported ACES file will display as shown below.

```
<SenderPhone>7701231234</SenderPhone>
```

- **Document Title:** Allows a document title to be specified within the free text field. The value (along with the 'Company' value) is used to create the ZIP file name and will display within the ACES export between the <DocumentTitle> tags. For example, if the parameter is populated with the value 'EXPORT_V2,' then the exported ACES file will display as shown below.

```
<DocumentTitle>EXPORT_V2</DocumentTitle>
```

Note: When an ACES export is run, the ZIP file name is created using the following template: 'Company Value (defined in step 3 of the exporter wizard)_Document Title Value (defined in step 3 of exporter wizard)_current date_FULL.xml.'

In the screenshot below, the AutoCare ACES Application Exporter is displayed with the 'Company' and 'Document Title' parameters outlined in red. The values configured for these required parameters help to determine the ZIP file name displayed within a BGP Details screen. Using the example below, the ZIP file name would display as 'ACME_BrandAAIAID in Part_2018-10-31_FULL.zip'

Export Manager

Steps

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

Select Format

AutoCare ACES Application Exporter

Exports data in AutoCare ACES format.

Version: 3.2

Company: ACME

Sender Name:

Sender Phone:

Sender Phone Ext:

Mfr Code:

Brand AAIAID: ...

Document Title: BrandAAIAID in Part

Doc Form Number:

Back Next Finish Cancel

Optional Parameters

Optional parameters are describe below. By default, the parameter values are blank unless an existing configuration is selected.

- **Version:** By default, the '4.0' version is selected, but any one of the available ACES export versions can be selected from the dropdown list. Within the ACES export, the version displays before the Header column (as shown below).

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

```
<ACES version="4.0">
```

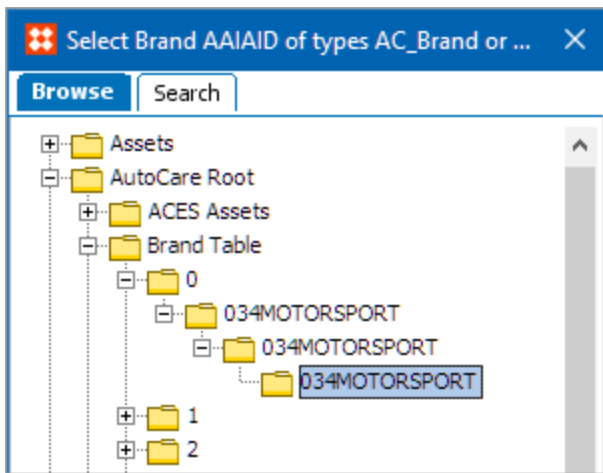
- **Sender Phone Ext:** Allows a sender phone extension to be specified within the free text field. The value will display within the ACES export between the <SenderPhoneExt> tags. For example, if the value '6789' is provided, then the exported ACES file will display as shown below.

```
<SenderPhoneExt>6789</SenderPhoneExt>
```

- **Mfr Code:** Allows a manufacturer code to be specified within the free text field. The value will display within the ACES export between the <MfrCode> tags. For example, if the value 'TNT' is provided, then the exported ACES file will display as shown below.

```
<MfrCode>TNT</MfrCode>
```

- **Brand AAIID:** Allows a Brand or Sub Brand AAIID (Automotive Aftermarket Industry Association ID) object to be selected. Click the ellipsis button (...), and the 'Select Brand AAIID' dialog will display, as shown in the image below.



Select the desired Brand, and save and close the dialog. The STEP Name of the selected Brand will display followed by the Brand AAIID in parentheses. However, only the four character Brand AAIID will display between the <BrandAAIID> tag in the export.

For Example, if the Brand object '034MOTORSPORT (ID=GWWQ)' is selected, then the exported ACES file will display as shown below.

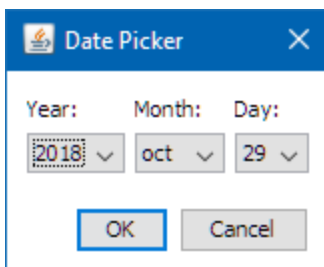
```
<BrandAAIID>GWWQ</BrandAAIID>
```

Note: The <BrandAAIID> tag embedded within the <Part> tag is not populated from this value. The embedded <BrandAAIID> tag is populated directly from the application record itself (i.e., if the application record had the parameter populated on import, it is populated with the same value on export).

- **Doc Form Number:** Allows a document or catalog number to be specified within the free text field. The value will display within the ACES export between the <DocFormNumber> tags. For example, if the value 'XX98765' is provided, then the exported ACES file will display as shown below.

```
<DocFormNumber>XX98765</DocFormNumber>
```

- **Effective Date:** Allows an effective date to be specified using the YYYY-MM-DD date format. Clicking within the field displays the Date Picker dialog (as shown below), which ensures the correct format is used when editing the value.



The Date Picker dialog defaults to the current date, regardless of what is displayed in the Effective Date field prior to selection. Any past or future date can be selected, and changes made are displayed in the Effective Date field and populated in the export as YYYY-MM-DD. Within the ACES export, the date value will display between the <EffectiveDate> tags. For example, if the value '2018-06-18' is provided, then the exported ACES file will display as shown below.

```
<EffectiveDate>2018-06-18</EffectiveDate>
```

Note: When a saved configuration is used and the value is blank, then the effective date will be generated from the export's scheduled run date.

- **Approved For:** Allows for one or more country codes to be specified. The display of available two character country code values is controlled by an LOV that is created as part of the Easy Setup actions. For more information, see the **10. Add Country Codes to Approved for LOV (ACES Export Only)** topic within the **Automotive Quick Start Guide**. Optionally, select one or more checkboxes and the selected value(s) will display within the ACES export between the <ApprovedFor> tags. However, the ApprovedFor tag in the Header segment varies based on format. Assuming values of 'US' and 'DK' are included, an exported ACES file will display for as shown below.

- Version 3.0:

```
<ApprovedFor>DK,US</ApprovedFor>
```

- Version 3.2:

```
<ApprovedFor>
<Country>DK</Country>
<Country>US</Country>
</ApprovedFor>
```

- Version 4.0:

```
<ApprovedFor>
<Country>DK</Country>
<Country>US</Country>
</ApprovedFor>
```

- **Mapper Company:** Allows for a company responsible for mapping the data to ACES to be specified within the free text field. The value will display within the ACES export between the <MapperCompany> tags. For example, if the value 'NYKA' is provided, then the exported ACES file will display as shown below.

```
<MapperCompany>NYKA</MapperCompany>
```

- **Mapper Contact:** Allows for a person to contact at the mapping company to be specified within the free text field. The value will display within the ACES export between the <MapperContact> tags. For example, if the value 'MAC' is provided, then the exported ACES file will display as shown below.

```
<MapperContact>MAC</MapperContact>
```

- **Mapper Phone:** Allows for the mapper contact's phone information to be specified within the free text field. The value will display within the ACES export between the <MapperPhone> tags. For example, if the value '7801323312' is provided, then the exported ACES file will display as shown below.

```
<MapperPhone>7801323312</MapperPhone>
```

- **Mapper Phone Ext:** Allows for the mapper contact's phone extension to be specified within the free text field. The value will display within the ACES export between the <MapperPhoneExt> tags. For example, if the value '2222' is provided, then the exported ACES file will display as shown below.

```
<MapperPhoneExt>2222</MapperPhoneExt>
```

- **Mapper Email:** Allows for the mapper contact's email information to be specified within the free text field. The value will display within the ACES export between the <MapperEmail> tags. For example, if the value 'abcd@acme.com' is provided, then the exported ACES file will display as shown below.

```
<MapperEmail>abcd@acme.com</MapperEmail>
```

- **Deliver a file if there are validation errors?:** By default, 'no' is selected, and the export file will not be delivered when validation errors occur. When 'yes' is selected, the export file will be delivered even though validation errors occur.

Step 5. Advanced

When an ACES application format is selected in the previous dialog, the Advanced step within the Export Manager includes several parameters unique to the ACES exporter (outlined in red below). Only the parameters specific to the ACES format are described within this topic. For more information on the general parameters within the Advanced Export Manager dialog, see the **Export Manager - Advanced** topic within **STEP Online Help**.

The screenshot shows the 'Export Manager' dialog box with the 'Advanced' step selected. The 'Steps' pane on the left lists: 1. Select Configuration, 2. Select Objects, 3. Select Format, 4. Map Data, 5. Advanced (selected), and 6. Select Delivery Method. The main area contains the following settings:

- Tag conversion (Coupon)
- Locale conversion from context
- Resolve Inline References
 - Export Only Local Inline References
 - Export Inherited Inline References
- Include Calculated Attribute Values
- Workspace: Main
- Context: English US
- Part Number Source:** ID Name Attribute Value (Part Number (AC_PIES_ITEMPartNumber))
- Use Attribute for Mfr Label
- Include BrandAAIAID in Part

- **Part Number Source:** The following options are available for this parameter: ID, Name, Attribute Value.
- **Attribute Value:** By default, the 'Attribute Value' radio button is selected and the 'Part Number (AC_PIES_ITEMPartNumber)' value from the parental PIES part is inserted between the <Part> tags of the export. For example, when the application '034-VC21499Disc Brake Pad Set2011 Acura TSX' (a child application of '034-VC21499') is selected for export, then the parental part's Part Number attribute value displays within the export. The application and its parent are displayed in the workbench screenshot below.

034-VC21499	> Base Item Number	abc	VC21
034-VC21499	> Item Level GTIN	123	11234567890123
034-VC21499	> Item Level GTIN Qualifier		U.P.C./EAN Consumer Package Code (1-5-5-1)
034-VC21499Disc Brake Pad Set2011 Acura TSX	> Part Number	12a	034-VC21499
PK	> Brand AAIA ID	12a	GWWQ

When the 'Include BrandAAIAID in Part' parameter is disabled, then the ACES export will display as shown below.

```
<Part>034-VC21499</Part>
```

Whereas, if the 'Include BrandAAIAID in Part' parameter is enabled, then the ACES export will display as shown below.

```
<Part BrandAAIAID="GWWQ">034-VC21499</Part>
```

- **ID:** When enabled, the STEP ID of the parental PIES part is inserted between the <Part> tags of the export. For example, when the application '034-VC21499Disc Brake Pad Set2011 Acura TSX' (a child application of '034-VC21499') is selected for export, then the parental part's STEP ID ('AC_PIESItem_GWWQ_034-VC21499') displays within the export. The application and its parent are displayed in the workbench screenshot below.

034MOTORSPORT	Name	>	Value
Belts and Cooling	> ID		AC_PIESItem_GWWQ_034-VC21499
Accessory Drive Belt System Components	> Name		034-VC21499
A/C Drive Belt Idler Pulley	> Object Type		PIES Item
034-VC21499	> Revision		0.2 Last edited by PIESIMPORT on Tue Jun 19 10:08:41
034-VC21499	> Approved		✘ Never Been Approved
034-VC21499Disc Brake Pad Set2011 Acura TSX	> Translation		Not Translated
PK			

When the 'Include BrandAAIAID in Part' parameter is disabled, then the ACES export will display as shown below.

```
<Part>AC_PIESItem_GWWQ_034-VC21499</Part>
```

Whereas, if the 'Include BrandAAIAID in Part' parameter is enabled, then the ACES export will display as shown below.

```
<Part BrandAAIAID="GWWQ">AC_PIESItem_GWWQ_034-VC21499</Part>
```

- **Name:** When enabled, the STEP Name of the parental PIES part is inserted between the <Part> tags of the export. For example, when the application '034-VC21499Disc Brake Pad Set2011 Acura TSX' (a child application of '034-VC21499') is selected for export, then the parental part's STEP Name ('034-VC21499') displays within the export. The application and its parent are displayed in the workbench screenshot below.

Name	Value
> ID	AC_PIESItem_GWWQ_034-VC21499
> Name	034-VC21499
> Object Type	PIES Item
> Revision	0.2 Last edited by PIESIMPORT on Tue Jun 19 10:08:
> Approved	✘ Never Been Approved
> Translation	Not Translated

When the 'Include BrandAAIAID in Part' parameter is disabled, then the ACES export will display as shown below.

```
<Part>034-VC21499</Part>
```

Whereas, if the 'Include BrandAAIAID in Part' parameter is enabled, then the ACES export will display as shown below.

```
<Part BrandAAIAID="GWWQ">034-VC21499</Part>
```

- **Use Attribute for Mfr Label:** Allows for any attribute value from the parental PIES Item or the ACES application object to be populated between the <MfrLabel> tag of the application record. If the option is not selected and the application has a value for the 'ACES Mfr Label' attribute, then that attribute value will get exported as the value in the <MfrLabel> tag. For example, when the application '034-VC21499Disc Brake Pad Set2011 Acura TSX' (a child application of '034-VC21499') is selected for export, and the parameter is populated with the 'ACES Mfr Body Code' attribute, then the ACES export file will display the 'ACES Mfr Body Code' attribute value (0SF08E) between the <MfrLabel> tags. The application's parent attribute 'ACES Mfr Body Code' is displayed in the workbench screenshot below with the '0SF08E' value.

Name	Value
> Path	Primary Product
AutoCare ACES Attributes	
> ACES Mfr Body Code	0SF08E
> Mfr Label	abc ABCD

Note: The automotive solution does not support the MfrLabel value coming from both the Part and the application. The MfrLabel value can only be retrieved from the Part or the application.

- If selected, and an attribute is specified from the Part to retrieve the value for the MfrLabel, then the MfrLabel value from the Part will be included for all applications belonging to that Part.

- If selected, and an attribute is specified from the Application to retrieve the value for the MfrLabel, then only the applications that have a MfrLabel attribute value will have the <MfrLabel> tag included for those specific applications in the exported file.
- If not selected, and an application has a value in the AC_ACESMfrLabel attribute, then that value will be exported only for the application that has the AC_ACESMfrLabel attribute value.
- **Include BrandAAIAID in Part:** By default this parameter is deselected so that the BrandAAIAID is excluded from the part number tag. When selected, the Brand AAIAID is inserted between the <Part> tag of the export. Examples are included within the above Part Number Source parameter details.

Note: General Export Manager functionalities like Tag conversion, Locale Conversion, Resolve Inline References, Calculated Attributes, Workspace, and Context are covered in the **Export Manager - Advanced** topic within the **Data Exchange** section of **STEP Online Help**.

Example

Below is an example of how the exporter could be configured and the expected ACES xml file results for one application.

Export Manager
✕

Steps

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

Select Format

AutoCare ACES Application Exporter

Exports data in AutoCare ACES format.

Version	3.2
Company	ACME
Sender Name	Grace
Sender Phone	770-123-1234
Sender Phone Ext	6789
Mfr Code	TNT
Brand AAIAID	034MOTORSPORT (AC_Brand_GWWQ)
Document Title	EXPORT_V2
Doc Form Number	XX98765
Effective Date	
Approved For	<input checked="" type="checkbox"/> Denmark <input checked="" type="checkbox"/> USA
Mapper Company	NYKA
Mapper Contact	MAC
Mapper Phone	7801323312
Mapper Phone Ext	2222
Mapper Email	abcd@acme.com
Deliver a file if there are validation errors?	yes

Back Next Finish Cancel

Export Manager

Steps

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

Advanced

Tag conversion Coupon

Locale conversion from context

Resolve Inline References

Export Only Local Inline References

Export Inherited Inline References

Include Calculated Attribute Values

Workspace Main

Context English US

Part Number Source ID Name Attribute Value Part Number (AC_PIES_ITEMPartNumber)

Use Attribute for Mfr Label Mfr Body Code (AC_ACESMfrBodyCode)

Include BrandAAIAID in Part

Back Next Finish Cancel

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ACES version="3.2">
<Header>
  <Company>ACME</Company>
  <SenderName>Grace</SenderName>
  <SenderPhone>770-123-1234</SenderPhone>
  <SenderPhoneExt>6789</SenderPhoneExt>
  <TransferDate>2018-11-02</TransferDate>
  <MfrCode>TNT</MfrCode>
  <BrandAAIAID>GWWQ</BrandAAIAID>
  <DocumentTitle>EXPORT_V2</DocumentTitle>
  <DocFormNumber>XX98765</DocFormNumber>
  <EffectiveDate>2018-11-02</EffectiveDate>
  <ApprovedFor>
  <Country>DK</Country>
  <Country>US</Country>
  </ApprovedFor>
  <SubmissionType>FULL</SubmissionType>
  <MapperCompany>NYKA</MapperCompany>
```

```

<MapperContact>MAC</MapperContact>
<MapperPhone>7801323312</MapperPhone>
<MapperPhoneExt>2222</MapperPhoneExt>
<MapperEmail>abcd@acme.com</MapperEmail>
<VcdbVersionDate>2018-03-30</VcdbVersionDate>
<QdbVersionDate>2018-03-30</QdbVersionDate>
<PcdbVersionDate>2018-03-30</PcdbVersionDate>
</Header>
<App action="A" id="1">
  <BaseVehicle id="31991"/>
  <EngineBase id="2127"/>
  <Qual id="1920">
    <text>Cast Iron Steel Gears are not Interchangeable. Use of Incorrect Part
    Number will Cause Severe Damage to Camshaft Gear and/or Distributor
    Gear</text>
  </Qual>
  <Qual id="6">
    <param value="1"/>
    <param value="2"/>
    <text>#1 #2 Intake Inlets are Oval</text>
  </Qual>
  <Qty>1</Qty>
  <PartType id="11279"/>
  <Part BrandAAIAID="GWWQ">AC Delco Part Number 034-VC21499</Part>
</App>
<Footer>
  <RecordCount>1</RecordCount>
</Footer>
</ACES>

```

Considerations for Version ACES 3.0, ACES 3.2 and ACES 4.0

- <ACES version="X.X"> must be populated accordingly in the Header segment
- The DigitalAsset segment is excluded for ACES 3.0 export.
- If an application has an asset linked to it, only the AssetName will get exported with the application.
- For a 3.2 and 4.0 export, AssetName will also be populated within the application record, but an additional DigitalAsset segment with details about the asset is also included.

- The value for AssetName comes from attribute ID=AC_PIES_ASSTAssetID on the asset.
- The asset included in an export must be linked to the application record using any of these Image and Document Reference Types:
 - AC_ACESApplicationToInstallation
 - AC_ACESApplicationToOwnersManual
 - AC_ACESApplicationToPrimaryProductImage
 - AC_ACESApplicationToProductImage.

Note: ACES standard only supports one asset linked to each application, but STEP supports multiple assets linked per application. If more than one asset is linked to the application in STEP, only one asset is exported randomly. An entry is written to the execution report of the export for any skipped assets.

Sending Data Downstream

In order to send data to downstream systems, the AutoCare ACES Application Exporter has a Version parameter to select either 3.0 or 3.2 or 4.0 format.

The version will default to 4.0. If 3.0 is selected, then the exported file will contain the following:

- UTF-8 encoding
- <ACES version="3.0">
- *The ApprovedFor tag in the Header is in the 3.0 format: <ApprovedFor>DK,US</ApprovedFor>*

Important: This tag must be removed from the a 3.0 file before importing through the STEP ACES Importer, otherwise it will fail validation against 3.2 and 4.0 XSD.

AutoCare PIES Exporter

The AutoCare PIES Exporter is used to export parts data in PIES (Product Information Exchange Standard) format. The exporter defaults to exporting data in the standard AutoCare data model, but configuration options are also available to export data from an Own model in PIES format.

This topic addresses using the exporter, as well as how to export Own part objects other than PIES Items.

Note: Exporting from a non-PIES data model requires that the alternate data model utilizes the PIES attributes and references from the standard model, and has Packaging and Hazmat objects as children to the PIES Items (if packaging and hazmat objects are to be included in the export).

Prerequisites

The PIES exporter runs using standard Export Manager functionality. Only information specific to the PIES exporter is covered within this guide. Additional information on general Export Manager functionality is covered in the **Export Manager** topic within the **Data Exchange** section of **STEP Online Help**.

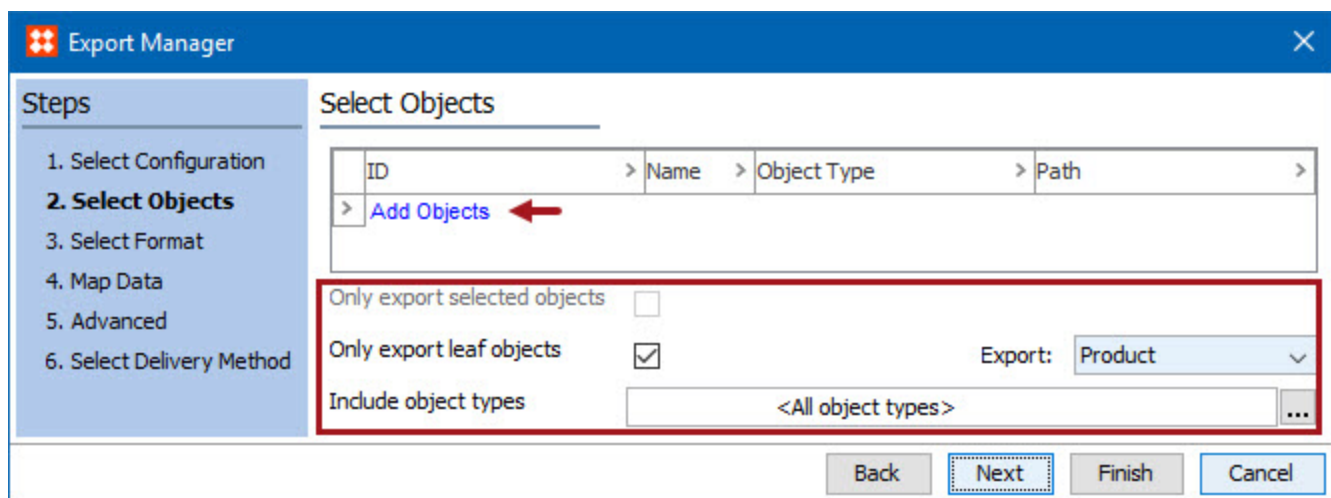
Using PIES Exporter

Step 1. Start AutoCare PIES Exporter

The AutoCare PIES Exporter can be started via standard Export Manager methods, e.g., File > Export > Data, or by selecting an object in Tree, right-clicking, and selecting 'Export Data Below' from the menu. In the case of the right-click option, an object in the AutoCare PIES Product hierarchy, an object in an Own model hierarchy that uses PIES data, or a yellow classification where part objects are linked to must be selected.

Step 2. Select Objects

In the Select Objects dialog, the only parameter that is read by the exporter is the actual object selection. The additional parameters (shown within the red box in the screenshot below) are disregarded.



Click the **Add Objects** link to add one or more blue folders in the AutoCare PIES Products hierarchy, or one or more blue folders from an Own hierarchy, or one or more classification folders.

All PIES objects beneath the selection(s) (either as child objects or via links) will result in the export file.

Important: If the standard AutoCare model is used, no additional selection is required. If an Own model is used, a PIES object type selection must be made in the next step of the Export Manager wizard.

Step 3. Select Format

Select **AutoCare PIES Exporter** format to display options specific to the PIES exporter. The version will default to 7.1, but any one of the available PIES export versions can be selected from the dropdown list.

Export Manager

Steps

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

Select Format

AutoCare PIES Exporter

Exports data in AutoCare PIES format.

Version: 7.1

Own Model Settings

Configuration to export PIES data from Own Model

Hierarchy Top Node:

Part Object Type:

Packaging Object Type:

Hazardous Materials Object Type:

Blanket Effective Date:

Parent DUNS Number:

Parent GLN:

Parent VMRSID:

Parent AAIAID:

Brand Owner VMRSID:

Brand Owner AAIAID:

Buyer Duns:

Currency Code:

Language Code:

Technical Contact:

Contact Email:

Brand Owner DUNS/GLN

Brand Owner DUNS:

Brand Owner GLN:

Deliver a file if there are validation errors?: no

Back Next Finish Cancel

Required Parameters

This section describes the required parameters that display in bold and require manual data population. The Next button will not enable until the required parameters are populated with the required number of characters. By default, the required parameter values are blank unless an existing configuration is selected.

- **Technical Contact:** Allows a contact name for resolving technical issues with PIES file to be specified for the exported data within the free text field. At least one character is required. The value will display within the PIES

export between the <TechnicalContact> tags. For example, if the value 'Grace' is provided, then the exported PIES file will display as shown below.

```
<TechnicalContact>Grace</TechnicalContact>
```

- **Contact Email:** Allows for the email of a contact for resolving technical issues with PIES file to be specified within the free text field. The value will display within the PIES export between the <ContactEmail> tags. For example, if the value 'abcd@acme.com' is provided, then the exported PIES file will display as shown below.

```
<ContactEmail>abcd@acme.com</ContactEmail>
```

Own Model Settings

If the standard AutoCare model is used, no selections should be made in the Own Model Settings section. If an Own model is used, the Own Model Settings must be populated as described below.

- **Hierarchy Top Node:** This is an optional parameter for selection of a product hierarchy top node (where Price Sheet data would be stored).
- If the PIES Pricing segment **does not** need to be included in the export, then this parameter can be left empty.
- If the PIES Pricing segment **is** to be included in the export, then the parental node where the price sheet information is stored needs to be specified in this parameter. The Data container and its attributes used to store the price sheet information must use the AutoCare IDs which is prefixed with '**AC_PIES_PRC**'.

PIES Importer creates price sheet Data Containers and price sheet attributes as they are needed, based on price sheet data in any import file. The data inherits to the product, but it is not maintainable there, and needs to be present on the parental node to be included in the export.

Important: If the export will include price data in PIES format, then a Data Container holding all Price Sheet attributes must be created and populated properly on the parental node from which they are exporting beneath.

For example, in the AutoCare PIES hierarchy, there is a PIES Products top node (ID=AC_PIESProducts) with Price Sheet attributes populated in a Data Container within the Data Containers tab:

The screenshot shows the 'PIES Products rev.0.30 - Data Containers' window. On the left is a 'Tree' view of the 'Primary Product Hierarchy' with 'PIES Products' selected. On the right is a table for the 'PIES Price Sheet' data container.

ID	Currency Code	Effective Date	Expiration Date	Price Sheet Name	Price Sheet Number	Price Zone	Superseded Price Sheet Number
PIES_PriceSheet_1	U.S. Dollar	2017-03-07	2018-03-07	2017 WD Prices	2017WD	Western	2016WD

- **Part Object Type:** This is a required parameter for selection of the Own model object type representing the PIES Item.
- The selected object must have all required PIES attributes and references (including the AC_ProductToPartTerminology reference) from the standard model made valid on the object and populated with data.
- If the required data is not provided, the file will fail XSD validation on downstream systems.

Note: The Part Object Type cannot be a direct child of the Hierarchy Top Node.

- **Packaging Object Type:** This is an optional parameter for selection of Packaging object type.
- If PIES Packaging segment is to be included in an export running outside of the AutoCare PIES hierarchy, the customer packaging object(s) must exist as child to the part object.
- If not selected, users cannot expect packaging data to export properly, and the export may fail if selections in the subsequent screen of the wizard (step 5. Advanced) try to include it.
- **Hazmat Object Type:** This is an optional parameter for selection of Hazmat object type.
- If PIES Hazmat segment is to be included in an export running outside of the AutoCare PIES hierarchy, the Hazmat object(s) must exist as child to the Packaging object.
- If not selected, users cannot expect hazmat data to export properly, and the export may fail if selections in the subsequent screen of the wizard (step 5. Advanced) try to include it.

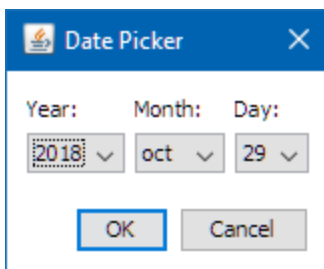
Optional Parameters

Optional parameters are describe below. By default, the parameter values are blank unless an existing configuration is selected.

- **Version:** By default, the '7.1' version is selected, but any one of the available PIES export versions can be selected from the dropdown list. Currently, PIES versions 6.5, 6.7, 7.0 and 7.1 are supported. Within the PIES export, the version displays between the <PIESVersion> tags within the Header column (as shown below).

```
<PIESVersion>7.1</PIESVersion>
```

- **Blanket Effective Date:** Allows an effective date to be specified using the YYYY-MM-DD date format. Clicking within the field displays the Date Picker dialog (as shown below), which ensures the correct format is used when editing the value.



The Date Picker dialog defaults to the current date, regardless of what is displayed in the Effective Date field prior to selection. Any past or future date can be selected, and changes made are displayed in the Effective Date field and populated in the export as YYYY-MM-DD. Within the PIES export, the date value will display between the <EffectiveDate> tags. For example, if the value '2018-10-29' is provided, then the exported PIES file will display as shown below.

```
<BlanketEffectiveDate>2018-10-29</BlanketEffectiveDate>
```

Note: When a saved configuration is used and the value is blank, then the effective date will be generated from the export's scheduled run date.

- **Parent DUNS Number:** Allows a unique 9 digit Parent DUNS Number to be specified within the free text field. May also use +4 format with 13 digits. The D&B D-U-N-S Number is a unique nine-digit identification sequence, which provides unique identifiers of single business entities, while linking corporate family structures together. The value will display within the PIES export between the <ParentDUNSNumber> tags. For example, if the value '002606342' is provided, then the exported PIES file will display as shown below.


```
<ParentDUNSNumber>002606342</ParentDUNSNumber>
```

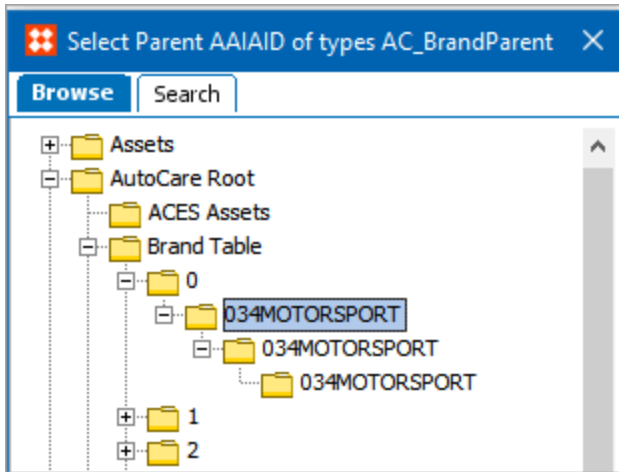
- **Parent GLN:** GS1 Company ID + Location ID + Check Digit. The GLN (Global Location Number) is a standard means of identifying global trading partner locations. Through the process of Product Synchronization, the seller and buyer identify each others names, addresses and other information, so that all subsequent electronic documents can be identified, routed and processed using only these codes. This is essential for the smooth, automated, error-free processing of electronic documents. The value will display within the PIES export between the <ParentGLN> tags. For example, if the value '1100001005984' is provided, then the exported PIES file will display as shown below.

```
<ParentGLN>1100001005984</ParentGLN>
```

- **Parent VMRSID:** Allows a unique 5 character Vehicle Maintenance Reporting Standards (VMRS) ID of the Parent Brand to be specified within the free text field. The Parent VMRSID is a unique Company identifier (required by major fleets) for all participating Heavy Duty parts manufacturers assigned by The Maintenance Council (TMC) of the American Trucking Assn. (ATA). The value will display within the PIES export between the <ParentVMRSID> tags. For example, if the value 'FL310' is provided, then the exported PIES file will display as shown below.

```
<ParentVMRSID>FL310</ParentVMRSID>
```

- **Parent AAIAID:** Allows a Brand Parent object to be selected. Click the ellipsis button () , and the 'Select Parent AAIAID of types AC_BrandParent' dialog will display, as shown in the image below.



Select the desired Parent Brand object, and save and close the dialog. The STEP Name of the selected Parent Brand will display followed by the STEP ID of the selected Object in parentheses. However, only the four character AAIA code of Parent Brand Object will display between the <ParentAAIAID> tag in the export.

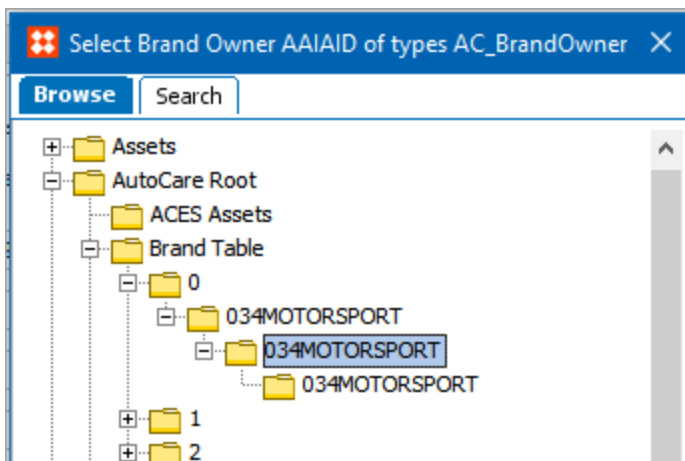
For Example, if the Parent Brand object '034MOTORSPORT' (STEP ID = AC_BrandParent_GWWN) is selected, then the exported PIES file will display as shown below.

```
<ParentAAIAID>GWWN</ParentAAIAID>
```

- **Brand Owner VMRSID:** Allows a unique 5 character Vehicle Maintenance Reporting Standards (VMRS) ID of the Brand Owner to be specified within the free text field. The Brand Owner VMRSID is a unique Company identifier (required by major fleets) for all participating Heavy Duty parts manufacturers assigned by The Maintenance Council (TMC) of the American Trucking Assn. (ATA). The value will display within the PIES export between the <BrandOwnerVMRSID> tags. For example, if the value 'FP220' is provided, then the exported PIES file will display as shown below.

```
<BrandOwnerVMRSID>FP220</BrandOwnerVMRSID>
```

- **Brand Owner AAIAID:** Allows a Brand Owner Object to be selected. Click the ellipsis button (...), and the 'Select Brand Owner AAIAID of types AC_BrandOwner' dialog will display, as shown in the image below.



Select the desired Brand Owner object, and save and close the dialog. The STEP Name of the selected Brand Owner Object will display followed by the STEP ID of the selected Object in parentheses. However, only the four character AAIA code of selected Brand Owner Object will display between the <BrandOwnerAAIAID> tag in the export.

For Example, if the Parent Brand object '034MOTORSPORT' (STEP ID = AC_BrandOwner_GWWP) is selected, then the exported PIES file will display as shown below.

```
<BrandOwnerAAIAID>GWWP/BrandOwnerAAIAID>
```

- **Buyer Duns:** Allows a unique 9 digit Buyer DUNS Number to be specified within the free text field. May also use +4 format with 13 digits. The D&B D-U-N-S Number is a unique nine-digit identification sequence, which provides unique identifiers of single business entities, while linking corporate family structures together. The value will display within the PIES export between the <BuyerDuns> tags. For example, if the value '003205242' is provided, then the exported PIES file will display as shown below.

```
<BuyerDuns>003205242</BuyerDuns>
```

- **Currency Code:** Allows a currency code to be specified within the free text field. This value may be overridden at the PRCS (Price Sheet Segment) or PRCE (Pricing Segment) levels. The value will display within the PIES export between the <CurrencyCode> tags. For example, if the value 'USD' is provided, then the exported PIES file will display as shown below.

```
<CurrencyCode>USD</CurrencyCode>
```

- **Language Code:** Allows a language code to be specified within the free text field. This value may be overridden within many of the underlying Segments. The value will display within the PIES export between the <LanguageCode> tags. For example, if the value 'EN' is provided, then the exported PIES file will display as shown below.

```
<LanguageCode>EN</LanguageCode>
```

- **Brand Owner DUNS/GLN:** The parameters must be populated with the correct number of characters. After one character is entered into either of the fields, a red fill will display until the correct number of characters is entered.
- **Brand Owner DUNS:** Allows an unique D&B DUNS ID (number to identify single business entities, while linking corporate family structures together) to be specified within the free text field. Must be between 9 and 13 digits. The value will display within the PIES export between the <BrandOwnerDUNS> tags. For example, if the value '005706338' is provided, then the exported PIES file will display as shown below.

```
<BrandOwnerDUNS>005706338</BrandOwnerDUNS>
```

- **Brand Owner GLN:** Allows a GLN (Global Location Number) (to identify global trading partner locations) to be specified within the free text field. Must have 13 digits. The value will display within the PIES export between the <BrandOwnerGLN> tags. For example, if the value '1100001009824' is provided, then the exported PIES file will display as shown below.

```
<BrandOwnerGLN>1100001009824</BrandOwnerGLN>
```

- **Deliver a file if there are validation errors?:** By default, 'no' is selected, and the export file will not be delivered when validation errors occur. When 'yes' is selected, the export file will be delivered even though

validation errors occur.

Step 4. Advanced

When a PIES format is selected in the previous dialog, the Advanced step within the Export Manager includes several parameters unique to the PIES exporter (outlined in red below). Only the parameters specific to the PIES format are described within this topic. For more information on the general parameters within the Advanced Export Manager dialog, see the **Export Manager - Advanced** topic within **STEP Online Help**.

Important: In the descriptions, the 'PIES object' is taken to mean either the standard AutoCare PIES Item object, or the Own model object representative of a PIES Item. In either case, the standard AutoCare PIES attributes for the segment must be valid on the object type and populated for the data to be included in the export.

The screenshot shows the 'Export Manager' dialog box in the 'Advanced' step. The 'Steps' sidebar on the left lists: 1. Select Configuration, 2. Select Objects, 3. Select Format, 4. Map Data, 5. **Advanced**, and 6. Select Delivery Method. The main area contains the following settings:

- Tag conversion (Coupon)
- Locale conversion from context
- Resolve Inline References
 - Export Only Local Inline References
 - Export Inherited Inline References
- Include Calculated Attribute Values
- Workspace: Main
- Context: EN All All
- Part Number Source** (highlighted in red):
 - ID
 - Name
 - Attribute Value (Part Number (AC_PIES_ITEMPartNumber))
- Price Sheets (2017WD, 090915, 070072)
- Description
- Extended Product Information (EXPI)
- Product Attributes
- Part Terminology Attributes
- Packaging
- Hazardous Materials
- Interchange

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

- **Part Number Source:** The following options are available for this parameter: ID, Name, Attribute Value.
- **Attribute Value:** By default, the 'Attribute Value' radio button is selected and the 'Part Number (AC_PIES_ITEMPartNumber)' value from the parental PIES part is inserted between the <Part> tags of the export. For example, when the application '034-VC21499Disc Brake Pad Set2011 Acura TSX' (a child application of '034-

VC21499') is selected for export, then the parental part's Part Number attribute value displays within the export. The application and its parent are displayed in the workbench screenshot below.

	> Base Item Number	abc	VC21
	> Item Level GTIN	123	11234567890123
	> Item Level GTIN Qualifier		U.P.C./EAN Consumer Package Code (1-5-5-1)
	> Part Number	12a	034-VC21499
	> Brand AAIA ID	12a	GWWQ

When the 'Include BrandAAIAID in Part' parameter is disabled, then the PIES export will display as shown below.

```
<Part>034-VC21499</Part>
```

Whereas, if the 'Include BrandAAIAID in Part' parameter is enabled, then the PIES export will display as shown below.

```
<Part BrandAAIAID="GWWQ">034-VC21499</Part>
```

- **ID:** When enabled, the STEP ID of the parental PIES part is inserted between the <Part> tags of the export. For example, when the application '034-VC21499Disc Brake Pad Set2011 Acura TSX' (a child application of '034-VC21499') is selected for export, then the parental part's STEP ID ('AC_PIESItem_GWWQ_034-VC21499') displays within the export. The application and its parent are displayed in the workbench screenshot below.

	Name	>	>	Value
	> ID			AC_PIESItem_GWWQ_034-VC21499
	> Name			034-VC21499
	> Object Type			PIES Item
	> Revision			0.2 Last edited by PIESIMPORT on Tue Jun 19 10:08:41
	> Approved			Never Been Approved
	> Translation			Not Translated

When the 'Include BrandAAIAID in Part' parameter is disabled, then the PIES export will display as shown below.

```
<Part>AC_PIESItem_GWWQ_034-VC21499</Part>
```

Whereas, if the 'Include BrandAAIAID in Part' parameter is enabled, then the PIES export will display as shown below.

```
<Part BrandAAIAID="GWWQ">AC_PIESItem_GWWQ_034-VC21499</Part>
```

- **Name:** When enabled, the STEP Name of the parental PIES part is inserted between the <Part> tags of the export. For example, when the application '034-VC21499Disc Brake Pad Set2011 Acura TSX' (a child application of '034-VC21499') is selected for export, then the parental part's STEP Name ('034-VC21499') displays within the export. The application and its parent are displayed in the workbench screenshot below.

	Name	>	>	Value
	>	ID		AC_PIESItem_GWWQ_034-VC21499
	>	Name		034-VC21499
	>	Object Type		PIES Item
	>	Revision		0.2 Last edited by PIESIMPORT on Tue Jun 19 10:08:
	>	Approved		✘ Never Been Approved
	>	Translation		Not Translated




When the 'Include BrandAAIAID in Part' parameter is disabled, then the PIES export will display as shown below.

```
<Part>034-VC21499</Part>
```

Whereas, if the 'Include BrandAAIAID in Part' parameter is enabled, then the PIES export will display as shown below.

```
<Part BrandAAIAID="GWWQ">034-VC21499</Part>
```

- **Price Sheets:** If the PIES Pricing segment (*<PriceSheets>* and *<Prices>*) is to be included in the export, then the Data Containers that holds the PIES Pricing attributes needs to be selected in this option.
- **Description:** If selected, all standard PIES Description attributes that are valid for the PIES object and have a value will be exported in the *<Descriptions>* segment.
- **Extended Product Information (EXPI):** If selected, all standard PIES EXPI attributes that are valid for the PIES object and have a value will be exported in the *<ExtendedInformation>* segment.
- **Product Attributes / Part Terminology Attributes:** Allows the selection of attribute groups that contain custom attributes to be included in the exported file (e.g., any attributes outside of those included in the PIES standard can be sent in this way). The values for these attributes will be included in the *<ProductAttributes>* segment for the PIES Item.
- **Packaging / Hazardous Materials:** If selected, exported data will be displayed in the *<Packages>* and *<HazardousMaterial>* segments.
- If PIES Packaging and Hazmat segments are to be included in an export running outside of the AutoCare PIES hierarchy, the customer's Packaging object(s) must exist as child to the PIES product object and the Hazmat object(s) must exist as a child to the Packaging object. Additionally, the standard PIES attributes for these segments must be made valid on the corresponding Own objects, and all required data for the segments must be populated or the file will fail XSD validation in downstream systems.
- **Interchange** - If selected, exported data will be displayed in the *<PartInterchangeInfo>* segment.
- The PIES Importer creates interchange records under the PIES Interchange Products node, which has a Brand and an Interchange child structure. If PIES Interchange data is to be exported, then this standard structure must be in place with the appropriate data on the interchanges (as is created with the PIES Importer).

<ul style="list-style-type: none"> Primary Product Hierarchy <ul style="list-style-type: none"> Products Discontinued Products Product Overrides Packaging AutoCare Root <ul style="list-style-type: none"> ACES Application Notes <ul style="list-style-type: none"> PIES Interchange Products <ul style="list-style-type: none"> Great Lakes <ul style="list-style-type: none"> GL26409 	<table border="1"> <thead> <tr> <th>Name ></th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>> ID</td> <td>AC_PIESInterchangeProducts</td> </tr> <tr> <td>> Name</td> <td>PIES Interchange Products</td> </tr> <tr> <td>> Object Type</td> <td>PIES Interchange Products</td> </tr> <tr> <td>> Revision</td> <td>0.3 Last edited by USER on Wed Jul 05 13:52:44 EDT 2017</td> </tr> <tr> <td>> Approved</td> <td> Never Been Approved</td> </tr> <tr> <td>> Translation</td> <td>Not Translated</td> </tr> <tr> <td>> Path</td> <td>Primary Product Hierarchy/AutoCare Root/PIES Interchange Products</td> </tr> </tbody> </table>	Name >	Value	> ID	AC_PIESInterchangeProducts	> Name	PIES Interchange Products	> Object Type	PIES Interchange Products	> Revision	0.3 Last edited by USER on Wed Jul 05 13:52:44 EDT 2017	> Approved	 Never Been Approved	> Translation	Not Translated	> Path	Primary Product Hierarchy/AutoCare Root/PIES Interchange Products
Name >	Value																
> ID	AC_PIESInterchangeProducts																
> Name	PIES Interchange Products																
> Object Type	PIES Interchange Products																
> Revision	0.3 Last edited by USER on Wed Jul 05 13:52:44 EDT 2017																
> Approved	 Never Been Approved																
> Translation	Not Translated																
> Path	Primary Product Hierarchy/AutoCare Root/PIES Interchange Products																

Important: If interchanges should be included in an export running from an Own model, the Own PIES Product object type must be added as a valid source for the Product Reference with ID= AC_PIESInterchange, and a reference must exist from the Own PIES Product to the AutoCare PIES Interchange object.

Example

Below is an example of how the exporter could be configured and the expected PIES xml file results for one part.

Export Manager
✕

Steps

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

Select Format

AutoCare PIES Exporter

Exports data in AutoCare PIES format.

Version ▲

6.5
▼

Own Model Settings **Configuration to export PIES data from Own Model**

Hierarchy Top Node ...

Part Object Type ...

Packaging Object Type ...

Hazardous Materials Object Type ...

Blanket Effective Date ...

Parent DUNS Number

Parent GLN

Parent VMRSID

Parent AAIAID ...

Brand Owner VMRSID

Brand Owner AAIAID ...

Buyer Duns

Currency Code

Language Code

Technical Contact

Contact Email

Brand Owner DUNS/GLN

Brand Owner DUNS

Brand Owner GLN

Deliver a file if there are validation errors? ▼

no
▼

Back
Next
Finish
Cancel

Export Manager

Steps

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

Advanced

Tag conversion Coupon

Locale conversion from context

Resolve Inline References

Export Only Local Inline References

Export Inherited Inline References

Include Calculated Attribute Values

Workspace Main

Context EN All All

Part Number Source ID Name Attribute Value Part Number (AC_PIES_ITEMPartNumber) ...

Price Sheets 2017WD
090915
070072

Description

Extended Product Information (EXPI)

Product Attributes ...

Part Terminology Attributes

Packaging

Hazardous Materials

Interchange

Back Next Finish Cancel

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
-<PIES xmlns="http://www.aftermarket.org">
-<Header>
  <PIESVersion>6.5</PIESVersion>
  <SubmissionType>FULL</SubmissionType>
  <ParentDUNSNumber>002606342</ParentDUNSNumber>
  <ParentGLN>1100001005984</ParentGLN>
  <ParentVMRSID>FP220</ParentVMRSID>
  <ParentAAIAID>GWWN</ParentAAIAID>
  <BrandOwnerDUNS>005706338</BrandOwnerDUNS>
  <BrandOwnerGLN>1100001009824</BrandOwnerGLN>
  <BrandOwnerVMRSID>FL220</BrandOwnerVMRSID>
  <BrandOwnerAAIAID>GWWP</BrandOwnerAAIAID>
  <BuyerDuns>003205242</BuyerDuns>
```

```

    <CurrencyCode>USD</CurrencyCode>
    <LanguageCode>EN</LanguageCode>
    <TechnicalContact>Grace</TechnicalContact>
    <ContactEmail>abcd@acme.com</ContactEmail>
</Header>
-<Items>
-<Item MaintenanceType="A">
    <HazardousMaterialCode>Y</HazardousMaterialCode>
    <BaseItemID>VC21</BaseItemID>
    <ItemLevelGTIN GTINQualifier="UP">11234567890123</ItemLevelGTIN>
    <PartNumber>034-VC21499</PartNumber>
    <BrandAAIAID>GWWQ</BrandAAIAID>
    <BrandLabel>034MOTORSPORT</BrandLabel>
    <ACESApplications>Y</ACESApplications>
    <ItemQuantitySize UOM="EA">1</ItemQuantitySize>
    <ContainerType>BO</ContainerType>
    <QuantityPerApplication UOM="EA"
    Qualifier="VAR">1</QuantityPerApplication>
    <ItemEffectiveDate>2010-08-13</ItemEffectiveDate>
    <AvailableDate>2010-08-13</AvailableDate>
    <MinimumOrderQuantity UOM="EA">123</MinimumOrderQuantity>
    -<ManufacturerProductCodes>
    <Group>02347</Group>
    <SubGroup>A</SubGroup>
    </ManufacturerProductCodes>
    <AAIAProductCategoryCode>199456</AAIAProductCategoryCode>
    <UNSPSC>11223348</UNSPSC>
    <PartTerminologyID>11279</PartTerminologyID>
    <VMRSCode>123000789</VMRSCode>
-<Descriptions>
    <Description MaintenanceType="A" DescriptionCode="VMR">VMR</Description>
    <Description MaintenanceType="A" DescriptionCode="UNS">UNS</Description>
    <Description MaintenanceType="A" DescriptionCode="SLA">SLA</Description>
    <Description MaintenanceType="A" DescriptionCode="MKT">MKT</Description>
    <Description MaintenanceType="A" DescriptionCode="EXT">EXT</Description>

```

```

<Description MaintenanceType="A" DescriptionCode="ASM">ASM</Description>
<Description MaintenanceType="A" DescriptionCode="INV">INV</Description>
<Description MaintenanceType="A"
DescriptionCode="SHO">CKENGSENS</Description>
<Description MaintenanceType="A" DescriptionCode="ABR">ABR</Description>
<Description MaintenanceType="A" DescriptionCode="KEY">Sensor Engine
Check</Description>
<Description MaintenanceType="A" DescriptionCode="ASC">ASC</Description>
<Description MaintenanceType="A" DescriptionCode="DEF">DEF</Description>
<Description MaintenanceType="A" DescriptionCode="SHP">SHP</Description>
<Description MaintenanceType="A" DescriptionCode="DES">Check Engine Light
Sensor</Description>
<Description MaintenanceType="A" DescriptionCode="LAB">LAB</Description>
</Descriptions>
-<ExtendedInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="PLC">100</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="WD1">500</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="TMC">AB</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="RET">RET</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A" EXPICode="NPD">Top
60%</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="REM">N</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="WT1">36</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="WS2">LF</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="STA">S</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="WS1">SR</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="PLM">20</ExtendedProductInformation>
    <ExtendedProductInformation MaintenanceType="A"
EXPICode="WD2">MI</ExtendedProductInformation>

```

```

<ExtendedProductInformation MaintenanceType="A"
EXPICode="RPC">30060</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="CPN">CPN</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="RST">GA</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="CGR">CGR</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="WT2">MO</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="CCL">CCL</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="MSR">Y</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="CTO">US</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="MSD">MSDS26426</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A" EXPICode="RCT">Marietta,
GA</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="EMS">1</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A" EXPICode="RCS">Cobb,
GA</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="CXP">12</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="OSN">OSN</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="CTR">GB</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="OEP">OEP</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="CTQ">DE</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="OEM">OEM</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="CTP">DK</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="HSB">01</ExtendedProductInformation>

```

```

<ExtendedProductInformation MaintenanceType="A"
EXPICode="HAC">HAC</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="CTS">FR</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="LIS">Proposed</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="OSP">OSP</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="LIF">0</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="NAF">1</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="HTS">02</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="NPC">A</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="REF">Y</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="HZ1">12</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="PTS">PTS</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="TAX">Y</ExtendedProductInformation>

<ExtendedProductInformation MaintenanceType="A"
EXPICode="PTN">PTN</ExtendedProductInformation>

</ExtendedInformation>
-<PartInterchangeInfo>
    -<PartInterchange MaintenanceType="A" LanguageCode="EN">
        <TypeCode>0</TypeCode>
        <BrandAAIAID>BBTB</BrandAAIAID>
        <BrandLabel>Great Lakes</BrandLabel>
        <PartNumber>GL26409</PartNumber>
        <QualityGradeLevel>P</QualityGradeLevel>
        <InterchangeNotes>Yellow indicator light</InterchangeNotes>
        <InternalNotes>May be difficult to remove</InternalNotes>
    </PartInterchange>
</PartInterchangeInfo>
-<DigitalAssets>

```

```
-<DigitalFileInformation MaintenanceType="A" LanguageCode="EN"  
AssetID="Asset456">  
  <FileName>VC21499_P04</FileName>  
  <AssetType>P04</AssetType>  
  <Representation>R</Representation>  
  <Resolution>96</Resolution>  
  <Background>CLI</Background>  
  <OrientationView>BAC</OrientationView>  
  <Details>Details/Description</Details>  
  <FilePath>images\SVlight.tif</FilePath>  
  <URI>www.stibo.com</URI>  
  <FileDateModified>2008-04-11</FileDateModified>  
  <EffectiveDate>2008-04-11</EffectiveDate>  
  <ExpirationDate>2012-04-11</ExpirationDate>  
  <Country>US</Country>  
</DigitalFileInformation>  
</DigitalAssets>  
</Item>  
</Items>  
-<Trailer>  
  <ItemCount>1</ItemCount>  
  <TransactionDate>2018-11-07</TransactionDate>  
</Trailer>  
</PIES>
```

NAPA Application Exporter

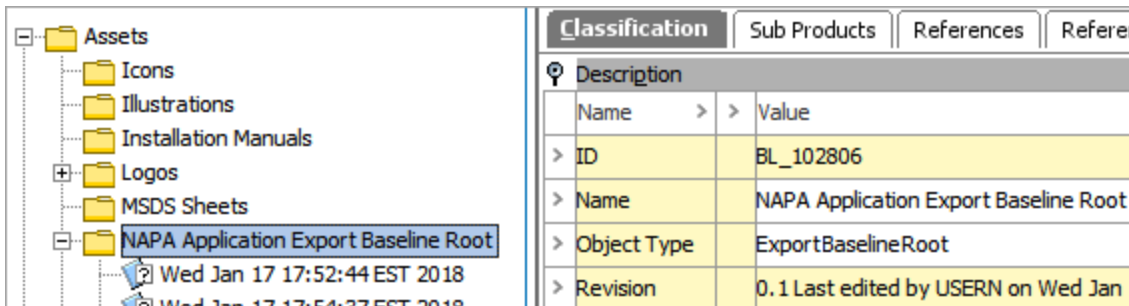
The NAPA Application Exporter is used to export application data that resides in a NAPA product hierarchy. Users can select an object from the NAPA product hierarchy, and the NAPA Application objects beneath the selected object (both child objects and linked objects) will be exported. The export file is then available in the NAPA Application format.

Prerequisites

- **NAPA Data Model Requirements:** The exporter requires applications be stored in the standard NAPA data model.
- **NAPA Product Requirements:** The NAPA Product MPCC is expected to have an ID structure prefixed with NAPA_MPCC_, and the NAPA Product is expected to have the following ID structure: NAPA_Product_[Line Abbrev][PartNumber] (i.e., NAPA_Product_FIL1515). When a NAPA Product is created, the NAPA Part Number attribute (NAPA_PartNumber) must be populated with the Part Number value, as this attribute value is used to determine the Part Number in the NAPA Application Exporter.

Important: Because some Product Lines only have two characters, the NAPA Product Number, if used without the overlying ID structure, might create an inconsistent pattern when trying to use the Product Number for the ID.

- **Object Type Requirements:** The following object types must exist within STEP: ExportBaselineRoot (Alternate Classification) and ExportBaselineAsset (Assets). As of the Automotive 8.3 release, these object types are created automatically when Easy Setup is run via System Setup > Component Models > Automotive - NAPA Model > Right-click Automotive - NAPA Model > 1. Configure NAPA Data Model.
- **Classification Requirements:** Additionally, a new classification using the alternate classification object (ExportBaselineRoot) must be manually created to hold the delta baseline assets. In the example below, NAPA Application Export Baseline Root has been created below the Assets root node.

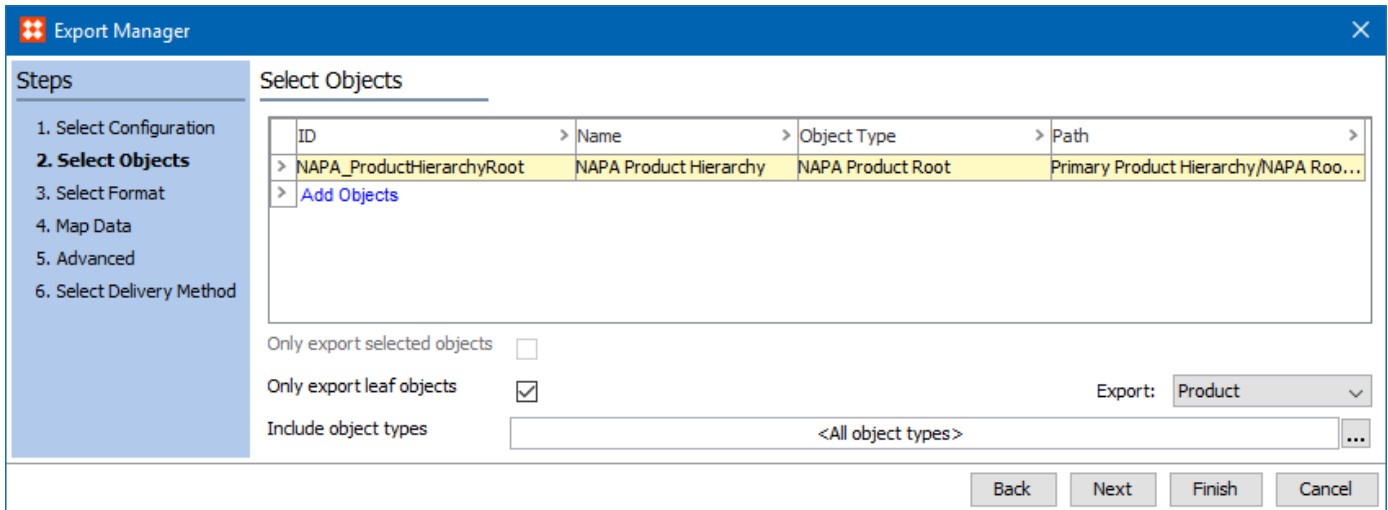


Using the NAPA Application Exporter

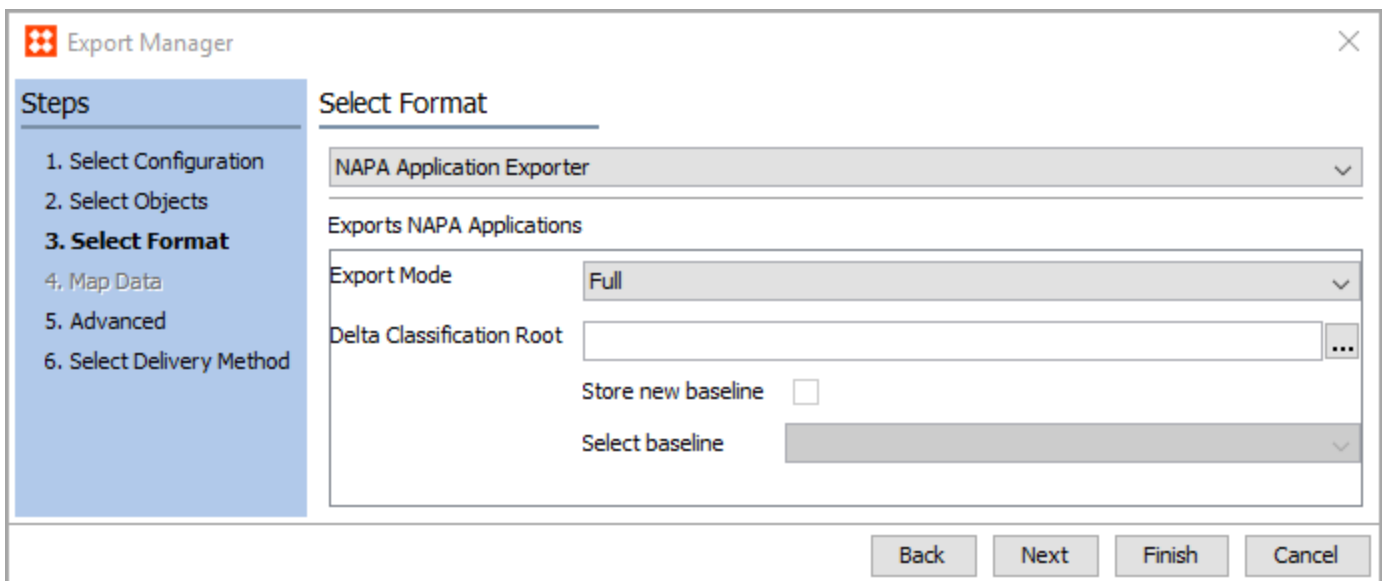
The steps below describe how to use the NAPA Application Exporter.

1. Go to the NAPA product hierarchy parent folder of the application(s) to be exported, Right-click, and select **Export Data Below**. The Export Manager will display with the parent folder object automatically added. In the

example below, the 'NAPA Product Hierarchy' was selected, and has been automatically added. Optionally, add additional objects.



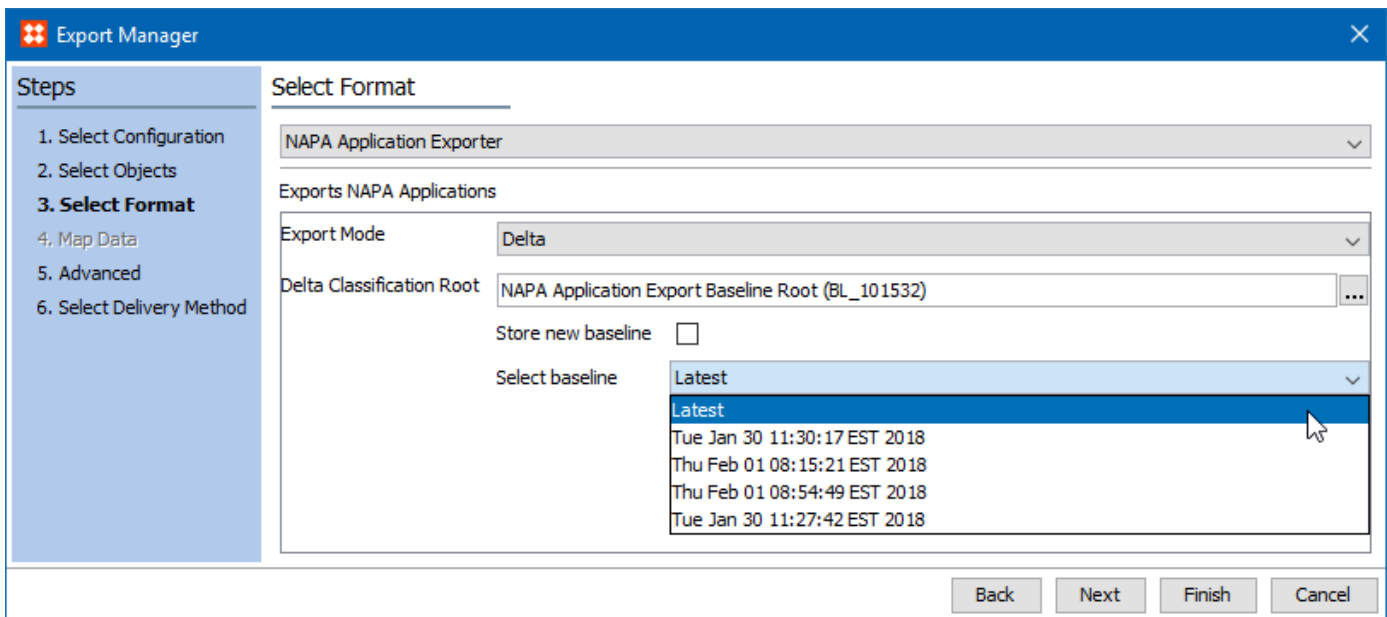
- Click the **Next** button, and the Export Manager will display the Select Format options. Use the dropdown to select the NAPA Application Exporter, and the Export Manager will display as shown below.



- Select the one of the following Export Modes from the dropdown:
 - Full:** This export mode will export applications linked to and/or beneath the objects selected in the previous step.
 - Delta:** This export mode will export only applications changed in comparison to the baseline selected below. A baseline must be selected, otherwise the delta file will be blank. Within the exported delta file, values that have changed will be marked with a 'D' Update Type for the old value on the record, and a separate line in the exported file will contain an 'A' Update Type for the new value.

Important: Before the Delta option can be used, at least one baseline must be stored by generating a FULL export and enabling the 'Store new baseline' checkbox. Otherwise the exporter has nothing to reference.

3. Within the Delta Classification Root parameter, click the ellipsis button (...) to find and select the alternate classification object (ExportBaselineRoot). Once the Delta Classification Root is populated, the Select baseline parameter dropdown will be enabled.
4. Select the desired baseline from the dropdown list. In the example below, four previous baselines and the 'Latest' baseline can be selected. (Optionally, the 'Store new baseline' checkbox can be enabled to create a new baseline).



5. Click the **Next** button and/or **Finish** button to complete the Export Manager wizard. For more information on the core parameters within the Export Manager, see the **Export Manager** topic within the Data Exchange section of Online Help.

Automotive Business Rule Plugins

A number of business rule plugins are provided to assist in extending the core Automotive solution. This section addresses many of the Automotive business rule plugins as they are available when configuring Business Actions and Business Conditions.

Prerequisites

It is assumed that the admin user has knowledge of STEP administrative functions and experience working in System Setup, including creating and editing business rules, workflows, and Web UIs. This section targets only the specific information needed for a knowledgeable STEP admin user to identify and configure the Automotive-specific business rule plugins. For more introductory material of these concepts, see the **Business Rules**, **Workflows** **Web User Interfaces** sections of **STEP Online Help**.

Business Action Operations

Menu	Operation Name	Components required for use	Description
Automotive	Change assembly	Application Manager, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Manager to change the assembly / vehicle of one or more existing applications. Will not change the assembly of a missing application. For more information, see Business Action: Change Assembly .
Automotive	Change part	Application Manager, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Manager to change the part for a selected application. For more information, see Business Action: Change Part .
Automotive	Change part type	Application Manager, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Manager to change the part type for a selected application. Will not change the part type of a missing application. For more information, see Business Action: Change Part Type .
Automotive	Copy application to other assembly	Application Manager, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Manager to copy one or more existing applications to another assembly / vehicle. For more information, see Business Action: Copy Application to Other Assembly .
Automotive	Copy application to other part	Application Manager, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Manager to copy one or more existing applications to another part. For more information, see Business Action: Copy Application to Other Part .
Automotive	Copy applications to related parts	Application Manager, Bulk Update action button	Allows users to use a Bulk Updates action button within an Application Manager to copy one or more existing applications to one or more related parts. For more information, see Business Action: Copy Applications to Related Parts .
Automotive	Copy	Anywhere a bulk	Allows users to update the product hierarchy to match the classification hierarchy. The

Menu	Operation Name	Components required for use	Description
	classification hierarchy to product hierarchy	update can be run	operation has the ability to disregard the classification prefix in order to create the product hierarchy using a defined product prefix. Can be used after a VCDB update has created new vehicles within the yellow classification folders to create blue hierarchy nodes. CAUTION: Object Types need to be similar in that they should have the same ID (without the prefix).
Automotive	Move ACES Applications for PIES Part		Moves all ACES applications from one PIES Item to another by following a reference between the parts. For more information, see Business Action: Move ACES Applications for PIES Part .
Automotive	Set Condition Links on Part Types		Links application conditions / options to part types to assist in configuring display options in the Web UI Application Record Editor. For more information, see Business Action: Set Condition Links on Part Types .
Automotive	Sync ACES Applications between PIES Parts		Synchronizes all ACES applications between two PIES Items by following a reference between the parts. For more information, see Business Action: Sync ACES Applications Between PIES Parts .
Automotive	Unique application record constraint		Calculates a unique value based on the part, assembly and the conditions for the applications.
Import flow	Set import status attributes	Workflow: Import State	Allows for the implementation of change flags; a way for users to view what data has been created or changed due to an import. Must be used within the Import state of a workflow. For more information, see Business Action: Set Import Status Attributes .

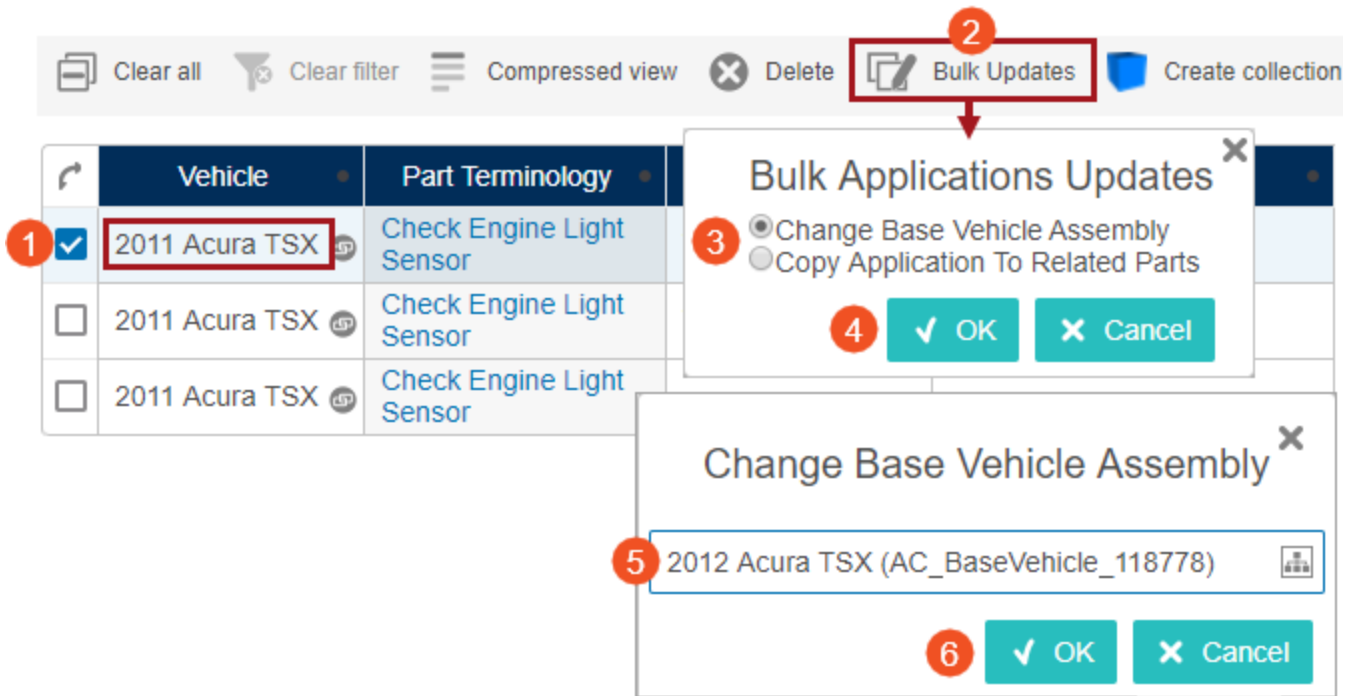
Business Condition Operations

Menu	Operation Name	Description
Automotive	Validate Application	Prevents any applications from being imported that do not have both a part type and vehicle / assembly that exists in the STEP database. When a record is found that does not meet the condition, an error is written to the execution report of the import process and the record is not imported. This condition is automatically created and added to the Import state of the relevant workflow when the Easy setup of the import process has been run.
Automotive	Check path for missing application	Improves the accuracy of Application Manager search results involving specific part types by allowing the data model to include a relationship between conditions on applications and different vehicle configurations. This can be helpful when specific part types do not apply to certain vehicle configurations. For more information, see Business Condition: Check Path for Missing Application .

Business Action: Change Assembly

This automotive business action allows users to use a Bulk Updates action button within an Application Manager to change the assembly / vehicle of one or more existing applications. However, it will not change the assembly / vehicle of a missing application, and thus any missing applications selected when the bulk update is run will be ignored. For more information about missing coverage, see the **Missing Application Coverage Functionality** topic within this guide.

Once configured, a Bulk Updates button within an Application Manager can be used to change the assembly for one or more existing application. Below are the steps for using the business action in Web UI.



Note: If only one bulk update is configured, then the Bulk Applications Updates dialog will not display.

1. Within the configured Application Manager, search for applications and select one or more applications to be changed.
2. Click the **Bulk Updates** action button. If more than one bulk update is configured then the Bulk Updates dialog will display (as shown above), otherwise this dialog is skipped and the change assembly dialog will display (skip to step 5 below).
3. Select the Change Base Vehicle Assembly radio button from the list displayed within the Bulk Applications Updates dialog. For this example, 'Change Base Vehicle Assembly' is used, but the business action name displayed within the list is dependent upon the business action's Name parameter.
4. Click the **OK** button and the change assembly dialog will display. For this example, 'Change Base Vehicle Assembly' is used, but the title of this dialog is controlled by the business action's Name parameter.

5. Select the desired assembly / vehicle for the application(s).
6. Click the **OK** button to close the dialog, and a background process notification will display.
7. Once the background process has completed, click the **Find applications** button to display the newly created application(s) within the Application Manager results table.

The diagram illustrates a change in the 'Vehicle' column of a table. On the left, a table with three rows is shown. The first row is selected (checkbox checked) and has '2011 Acura TSX' in the 'Vehicle' column, which is highlighted with a red box. A red arrow points to the right, where the same table is shown, but the first row now has '2012 Acura TSX' in the 'Vehicle' column, also highlighted with a red box. The 'Part Terminology' column for all rows is 'Check Engine Light Sensor'.

	Vehicle	Part Terminology
<input checked="" type="checkbox"/>	2011 Acura TSX	Check Engine Light Sensor
<input type="checkbox"/>	2011 Acura TSX	Check Engine Light Sensor
<input type="checkbox"/>	2011 Acura TSX	Check Engine Light Sensor

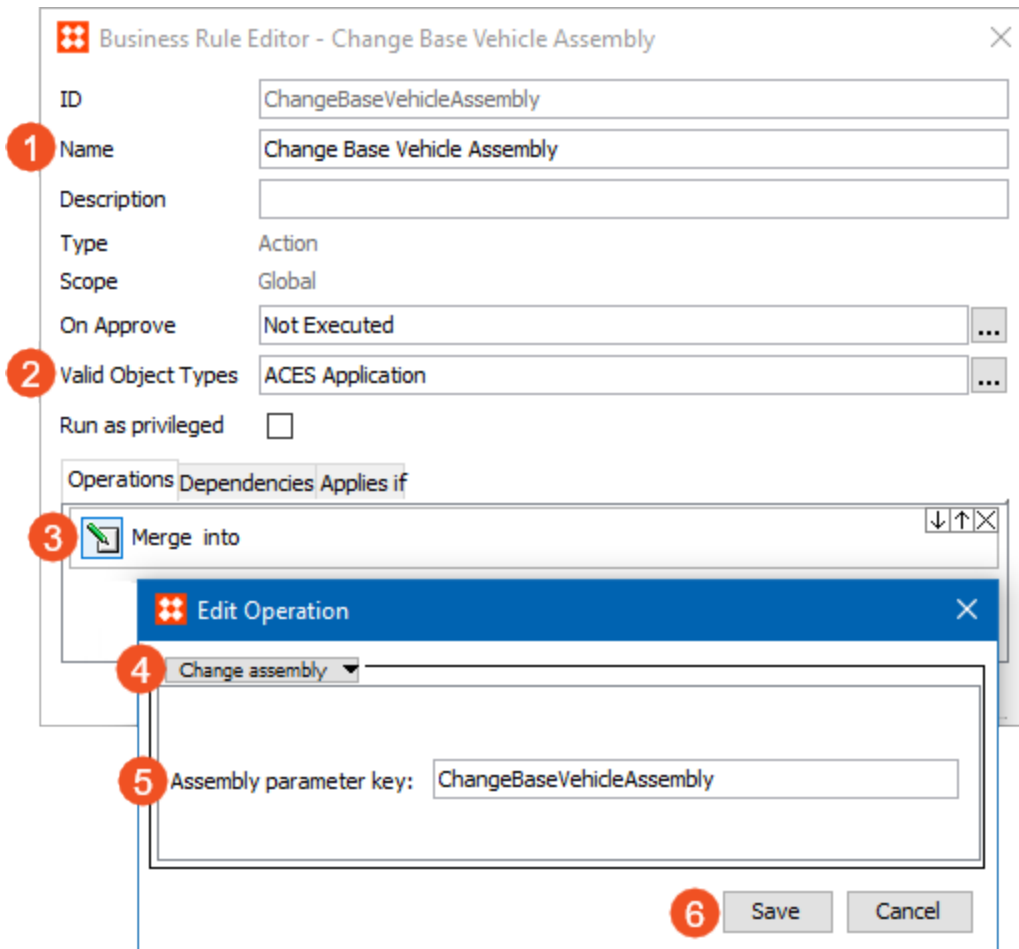
	Vehicle	Part Terminology
<input checked="" type="checkbox"/>	2012 Acura TSX	Check Engine Light Sensor
<input type="checkbox"/>	2011 Acura TSX	Check Engine Light Sensor
<input type="checkbox"/>	2011 Acura TSX	Check Engine Light Sensor

Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, see the configuration topics below:

- Configuring the Change Assembly Business Action in Workbench
- Configuring the Change Assembly Business Action in Web UI

Configuring the Change Assembly Business Action in Workbench

The 'Change assembly' business operation is found within the STEP Workbench Business Rule Editor under the Automotive menu and requires population of a single parameter (Assembly parameter key). However, setup is required within both STEP Workbench and Web UI for the action to be available to users. This section addresses the steps necessary within the workbench.



1. Create the business action with a name that accurately describes to the user what this action will do. The name of the business action displays within the Web UI and should be easy for the user to identify. For this example, the business action name is 'Change Base Vehicle Assembly.'

Note: The business action Name will display to the Application Manager user once the Bulk Updates action button is selected. If more than one bulk update is configured, then the name displays both within the Bulk Applications Update dialog, and within the change assembly dialog where the user enters the assembly for the application (as shown in the example within the **Using the Configured Change Assembly Business Action in Web UI** section of the **Business Action: Change Assembly** topic within this guide).

2. Edit the new business rule, click the ellipsis button (...) next to the Valid Object Types parameter and select the valid object types for this business action. For this example, the 'ACES Application' object type is selected, however the TecDoc and NAPA applications can also be selected within the same or separate business actions. This decision is at the discretion of the administrator.
3. On the Operations tab of the Business Rule Editor, click the **Add new Business Action** link, and click the edit button to open the Edit Operation dialog.
4. Use the dropdown menu within the Edit Operation dialog to select **Automotive > Change assembly** operation, and the parameter 'Assembly parameter key' will display (as shown above).
5. Within the parameter enter a unique way (key) to identify this rule. Uniqueness is the only restriction for this key. It will not be displayed to the user. It is case sensitive.

Important: Common setup is to copy the key so it can be pasted in the Web UI designer when configuring the business action in Web UI.

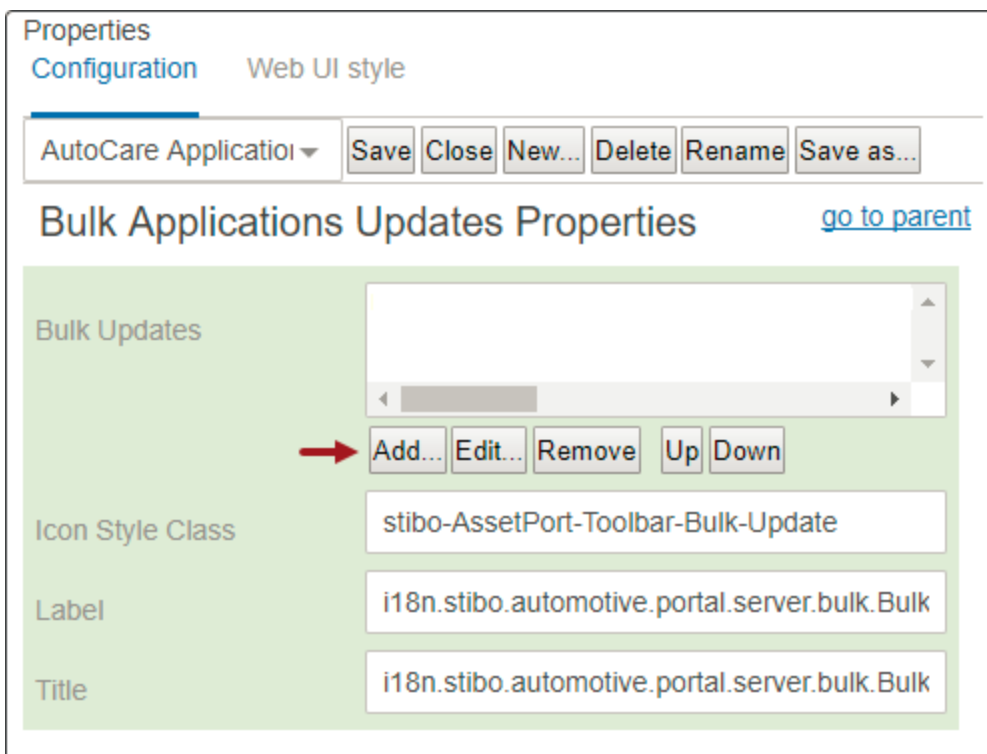
6. Click the **Save** button and continue to the next topic, **Configuring the Change Assembly Business Action in Web UI**.

Configuring the Change Assembly Business Action in Web UI

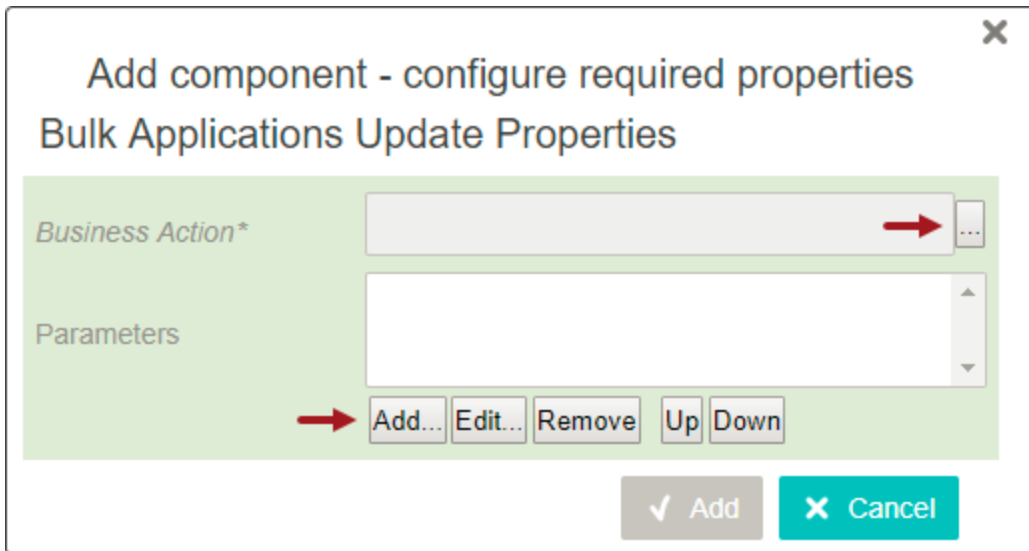
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

Note: After the initial setup, the access to the dialogs does not change, therefore these steps can be helpful when editing the configuration. However, when editing the 'Add' button will be replaced with the 'Save' button, and the 'Add component...' labels at the top of the dialogs will display as 'Edit component... '.

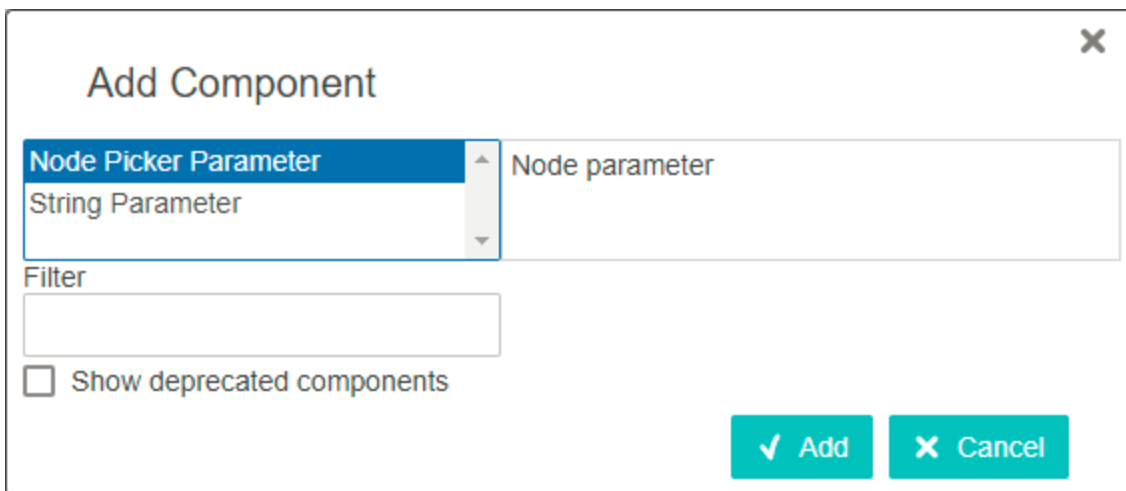
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



4. Click the ellipsis button (...) next to the Business Action parameter > select the business action previously created > click the **OK** button to close the dialog, and return to the Bulk Applications Update Properties dialog. The selected action is displayed in the Business Action parameter.
5. Click the **Add** button beneath the Parameters field, and the Add Component dialog will display (as shown below).

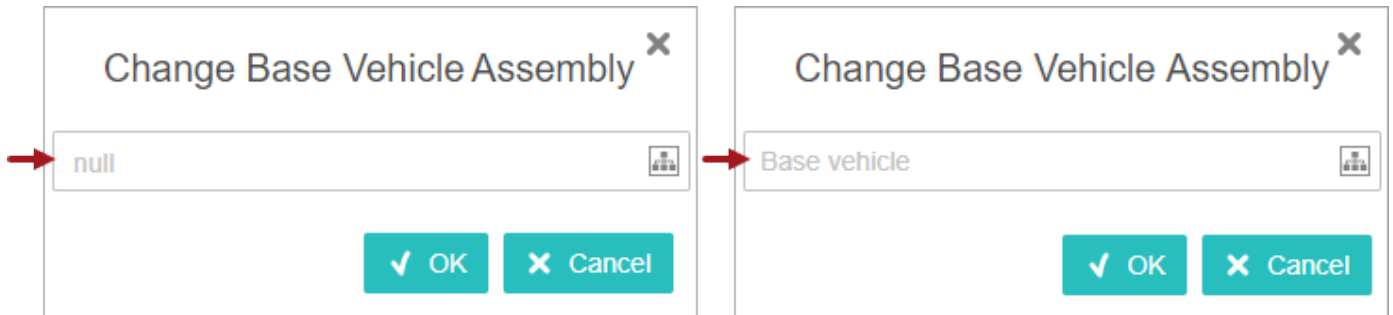


6. Select **Node Picker Parameter** > click the **Add** button to close the dialog, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display (as shown below).

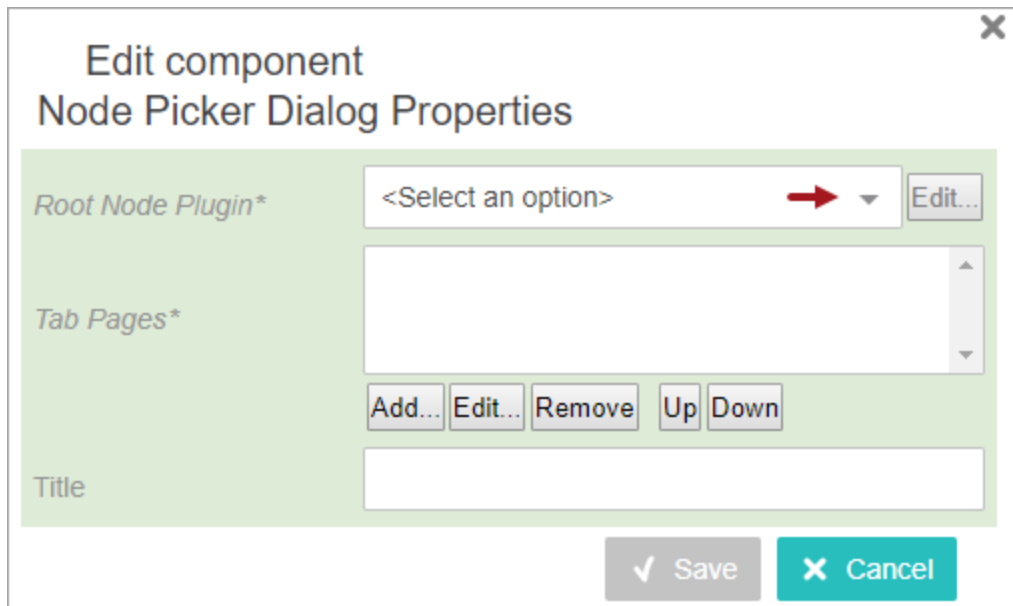
7. Within the Key parameter, enter the exact key created within the 'Assembly parameter key' parameter during step 5 of the **Configuring the Change Assembly Business Action in Workbench** topic. This is the only required parameter.

Important: If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

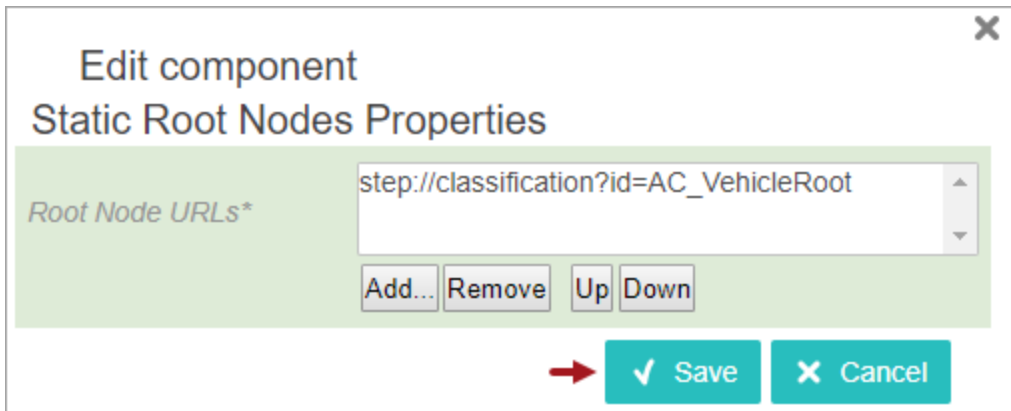
8. Within the Label parameter, enter text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.



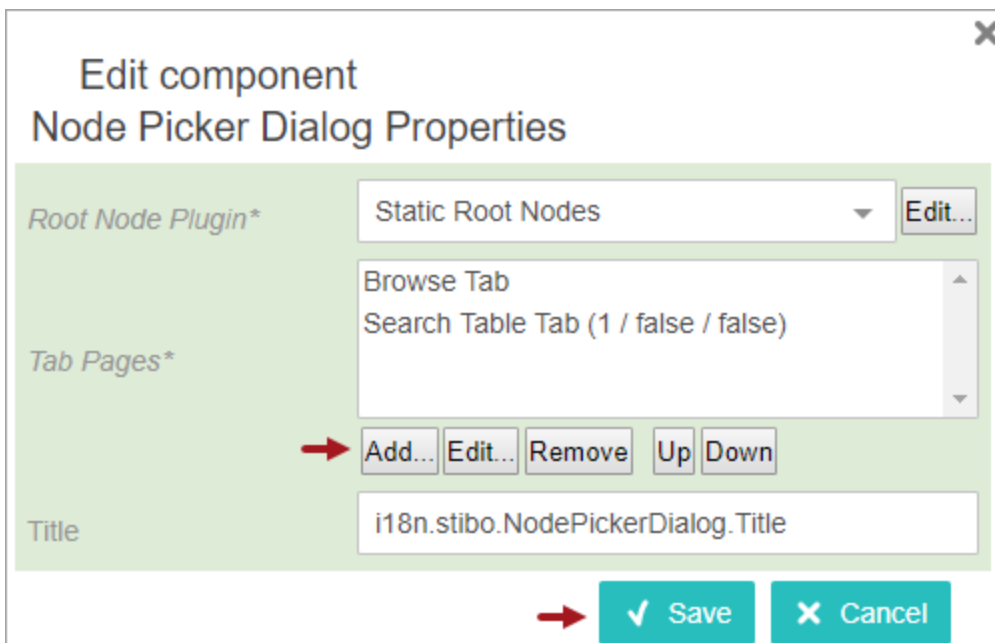
- Use the Node Picker Configuration parameter dropdown to select the **Node Picker Dialog** option, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display.



- Use the Root Node Plugin parameter dropdown to select the **Static Root Nodes** option, and the 'Edit component' for the 'Static Root Nodes Properties' dialog will display.
- Click the **Add** button beneath the Root Nodes URLs parameter > select the root nodes that the user should be able to choose from when changing the part type (for this example AC_VehicleRoot is used) > click the **OK** button to close the dialog, and return to the 'Edit component' for 'Static Root Nodes Properties' dialog. Optionally, repeat this step to add additional nodes for a user to browse from when looking for a part type.



12. Click the **Save** button, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display with the Root Node Plugin parameter populated with Static Root Nodes.
13. Click the **Add** button beneath the Tab Pages parameter > select the **Browse Tab** component > click the **Add** button to close the dialog and the 'Edit component' for 'Node Picker Dialog Properties' will display with the Tab Pages parameter populated with Browse Tab. Optionally, repeat this step and add the Search Table Tab.



Note: Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, see the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title parameter will display in place of the default 'Select Node(s)' title for the dialog used to select the vehicle assembly.

14. Click the **Save** button, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display with the Node Picker Configuration parameter populated.

15. Use the dropdown located beneath the Valid Node Types parameter to select **CLASSIFICATION_TYPE** > click the **Add** button beneath the Valid Node Types parameter so that CLASSIFICATION_TYPE is displayed within the Valid Node Types parameter (as shown below).
16. Click the **Add** button beneath the Valid Object Types parameter > select the desired valid object types > Click the **OK** button to return to the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog.

Add component - configure required properties

Node Picker Parameter Properties

Key*

Label

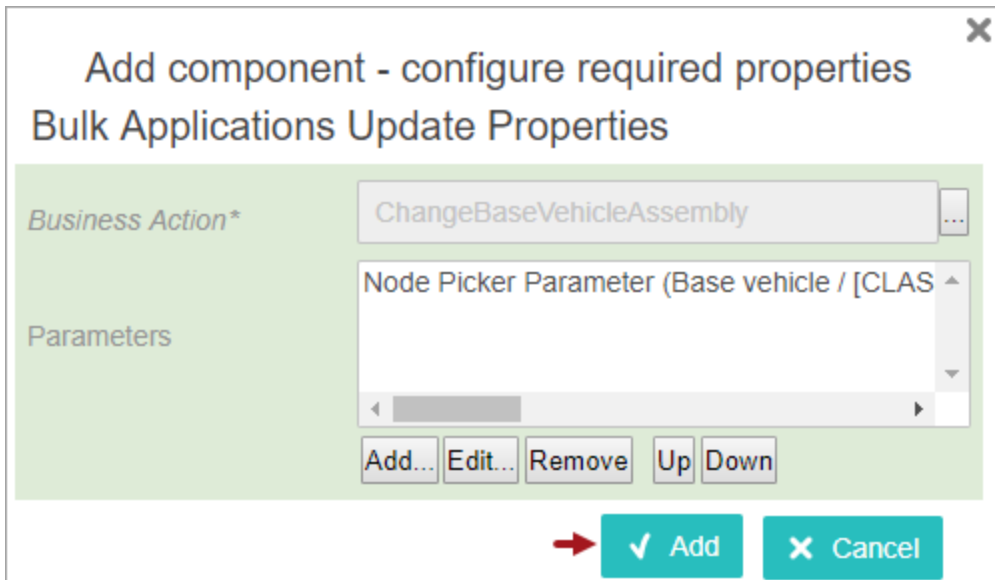
Mandatory

Node Picker Configuration

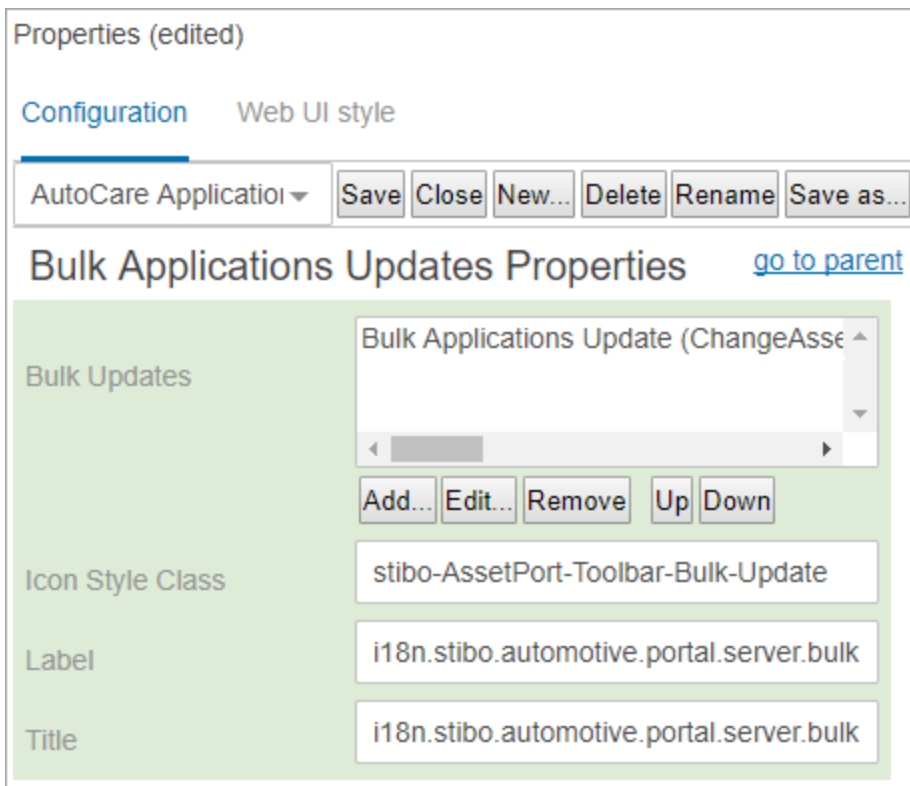
Valid Node Types

Valid Object Types

17. Click the **Add** button, and 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



- Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



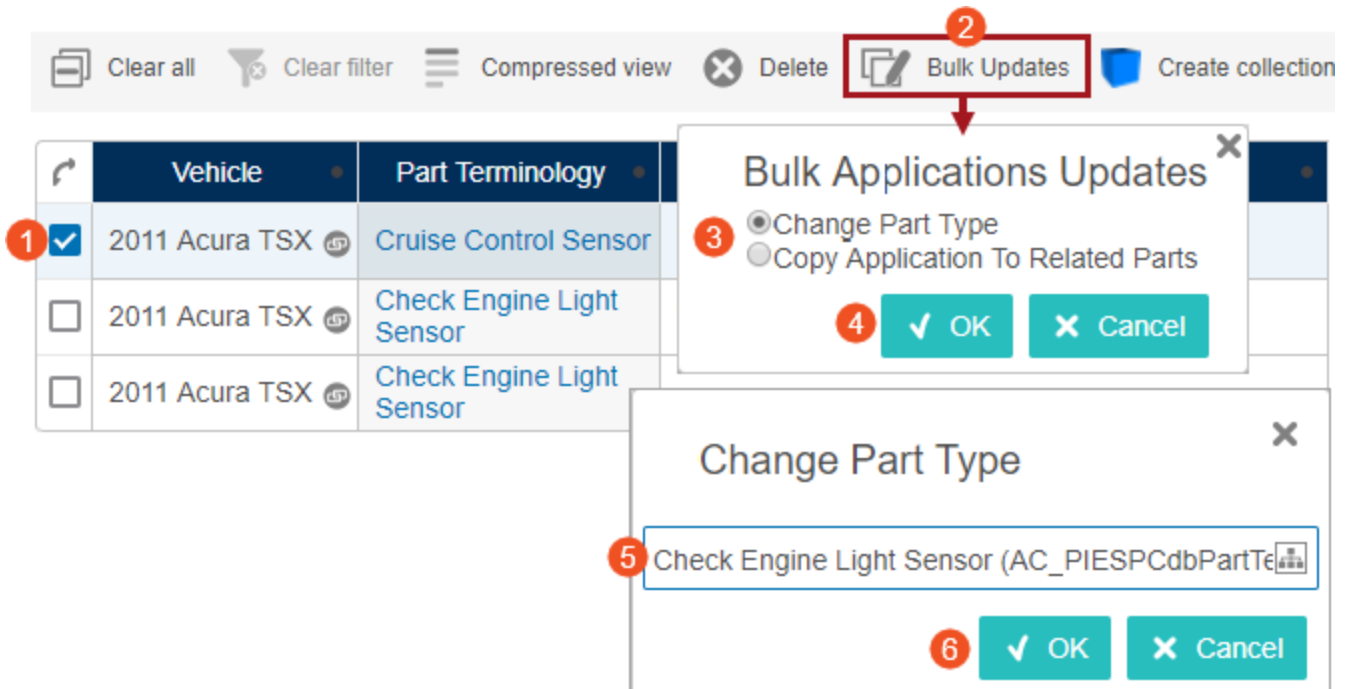
- Click the **Save** button and then click the **Close** button to close the designer.

To use the newly configured business action, see the **Using the Configured Change Assembly Business Action in Web UI** section of the **Business Action: Change Assembly** topic.

Business Action: Change Part Type





This automotive business action allows users to change the part type of one or more existing applications by clicking a Bulk Updates action button within an Application Manager. However, it cannot change the part type of a missing application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, see the **Missing Application Coverage Functionality** topic. Setup is required within both STEP Workbench and Web UI for the action to be available to users.

Once configured, a Bulk Updates button within an Application Manager can be used to change the part type for an existing application. Below are the steps for using the business action in Web UI.



Note: If only one bulk update is configured, then the Bulk Applications Updates dialog will not display.

1. Within the configured Application Manager, search for applications and select one or more applications to be changed.
2. Click the **Bulk Updates** action button. If more than one bulk update is configured, the Bulk Applications Updates dialog will display (as shown above), otherwise this dialog is skipped and the Change Part Type dialog will display (skip to step 5 below).
3. Select the Change Part Type radio button from the list displayed within the Bulk Application Updates dialog. For this example, 'Change Part Type' is used, but the business action name displayed within the list is dependent upon the business action's Name parameter.
4. Click the **OK** button and the Change Part Type dialog will display. For this example, 'Change Part Type' is used, but the title of this dialog is controlled by the business action's Name parameter.

5. Select the desired part type for the application(s).
6. Click the **OK** button to close the dialog, and a background process notification will display.
7. Click the BGP link to view the Background Process Details screen, or look for the gray background process (BGP) notification icon to display with a green dot  or orange dot  Notification icon. Once the business action has completed successfully, the gray background process (BGP) notification icon will display with a green dot . If the business action has failed or completed with errors, the BGP notification icon will display with an orange dot . For more information, see the **Background Process Notification Component** topic.
If the business action BGP fails, access the Background Process Details page for more information.

Note: Before completing this business action, by default, STEP confirms that the application's part type matches one or more part types assigned to the part. If one or more part types associated with the application's part do not match the application's part type, then an error can occur. However, this functionality can be overridden by using the 'Ignore Part Type' parameter within the business action. For more information, see the **Configuring the Change Part Type Business Action in Workbench** topic.

In the example below, the Background Process Details screen displays the failed BGP with the following error message: Selected part did not match part type from application [application ID].


Background Process Details

ID BGP_104103

Started By USERB

Description Run application bulk update with action: Change Part


Template ID AppBulkUpdService

Status  Failed

Started 10/29/18 9:23:35 AM

Finished 10/29/18 9:23:35 AM

Elapsed 1 s

Select all  Export

ID	Type	Text
<input type="checkbox"/> 10	Error	Selected part did not match part type from application: AC_ACESApp_2626c57b1edf6aa287c3be9d6172e
<input type="checkbox"/> 20	Error	com.stibo.core.domain.businessrule.plugin.BusinessRulePluginException: Selected part did not match part type from application: AC_ACESApp_2626c57b1edf6aa287c3be9d6172e

8. Once the background process has completed, click the **Find applications** button to display the newly created application(s) within the Application Manager results table.

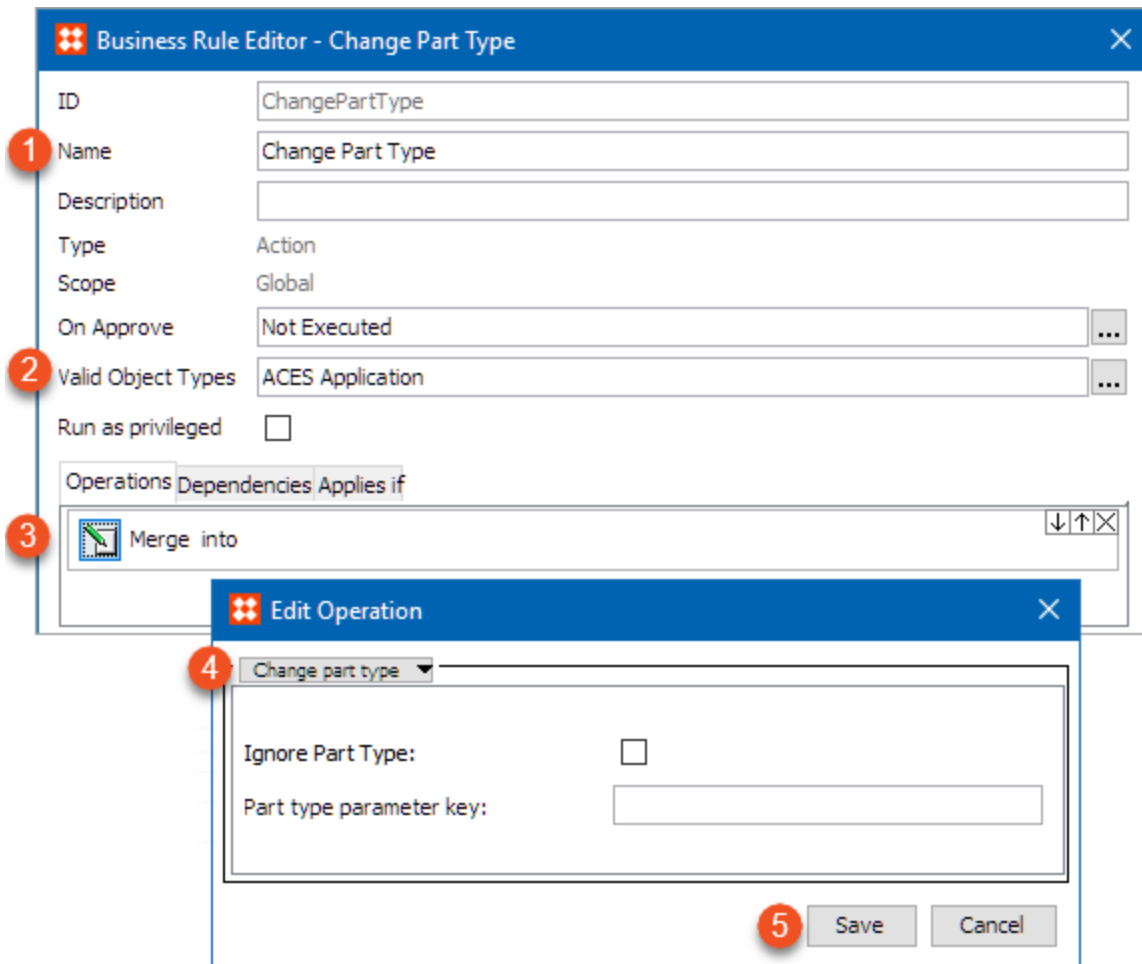
	Vehicle	Part Terminology
<input checked="" type="checkbox"/>	2011 Acura TSX 	Cruise Control Sensor
<input checked="" type="checkbox"/>	2011 Acura TSX 	Check Engine Light Sensor

Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, see the configuration topics below:

- Configuring the Change Part Type Business Action in Workbench
- Configuring the Change Part Type Business Action in Web UI

Configuring the Change Part Type Business Action in Workbench

The 'Change part type' business operation is found within the STEP Workbench Business Rule Editor under the Automotive menu and requires population of a single parameter (Part type parameter key). However, setup is required within both STEP Workbench and Web UI for the action to be available to users. This section addresses the steps necessary within the workbench.



1. Create the business action with a name that accurately describes to the user what this action will do. The name of the business action displays within the Web UI and should be easy for the user to identify. For this example, the business action name is 'Change Part Type.'

Note: The business action Name will display to the Application Manager user once the Bulk Updates action button is selected. If more than one bulk update is configured, then the name displays both within the Bulk Applications Update dialog, and within the change part type dialog where the user enters the part type for the application (as shown in the example within the **Using the Configured Change Part Type Business Action in Web UI** section of the **Business Action: Change Part Type** topic).

2. Edit the new business rule, click the ellipsis button (...) next to the Valid Object Types parameter and select the valid object types for this business action. For this example, the 'ACES Application' object type is selected, however the TecDoc and NAPA applications can also be selected within the same or separate business actions. This decision is at the discretion of the administrator.
3. On the Operations tab of the Business Rule Editor, click the **Add new Business Action** link, and click the edit button to open the Edit Operation dialog.
4. Use the dropdown menu within the Edit Operation dialog to select **Automotive > Change part type** operation, and the following parameters will display (as shown above):
 - **Ignore Part Type:** By default, this optional parameter is disabled. This parameter gives an option to check for a part type or not before changing a part for an application. When disabled, and a user tries to change the part number for an application to a part that does not have the same part type, the action will fail. When enabled, a user can change the part number for an application without concern for the part type.

In the example below, the Background Process Details page displays the business action status is Failed, and the error messages text includes 'selected part did not match part type from application.'

Background Process Details

ID	BGP_104103
Started By	USERB
Description	Run application bulk update with action: Change Part
Template ID	AppBulkUpdService
Status	▲ Failed
Started	10/29/18 9:23:35 AM
Finished	10/29/18 9:23:35 AM
Elapsed	1 s

Select all Export

	ID	Type	Text
<input type="checkbox"/>	10	Error	Selected part did not match part type from application: AC_ACESApp_2626c57b1edf6aa287c3be9d6172e
<input type="checkbox"/>	20	Error	com.stibo.core.domain.businessrule.plugin.BusinessRulePluginException: Selected part did not match part type from application: AC_ACESApp_2626c57b1edf6aa287c3be9d6172e

- **Part Type parameter key:** By default, this required parameter is blank. Within the parameter enter a unique way (key) to identify this rule. Uniqueness is the only restriction for this key. It will not be displayed to the user. It is case sensitive.

Important: Common setup is to copy the key so it can be pasted in the Web UI designer when configuring the business action in Web UI.

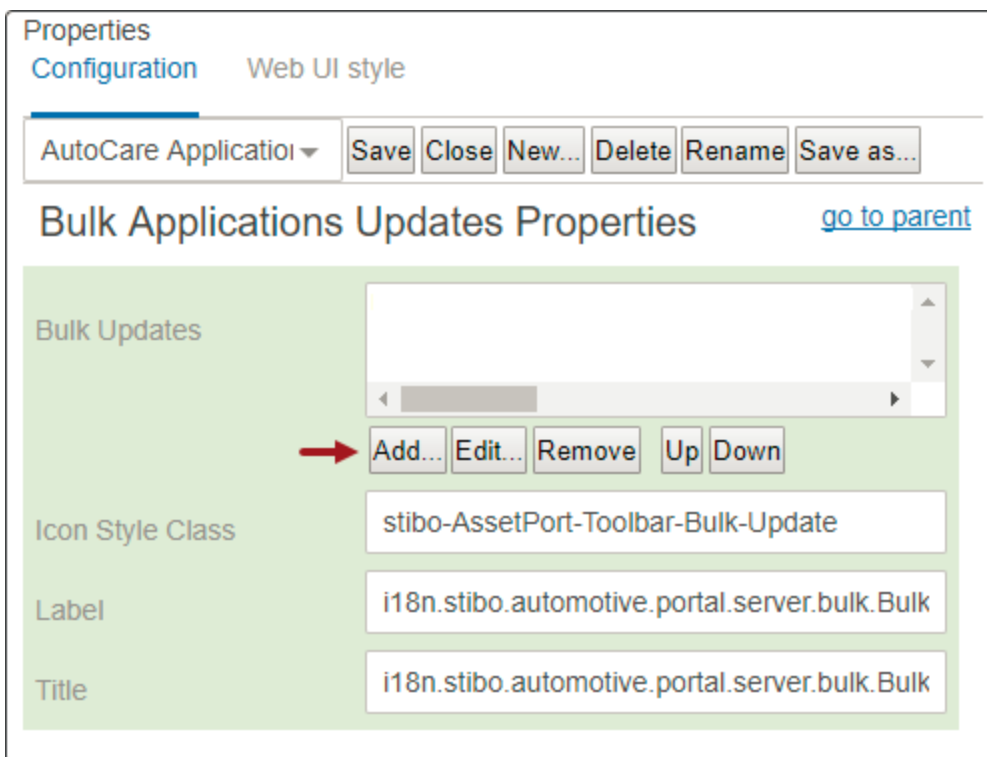
5. Click the **Save** button and continue to the next topic, **Configuring the Change Part Type Business Action in Web UI**.

Configuring the Change Part Type Business Action in Web UI

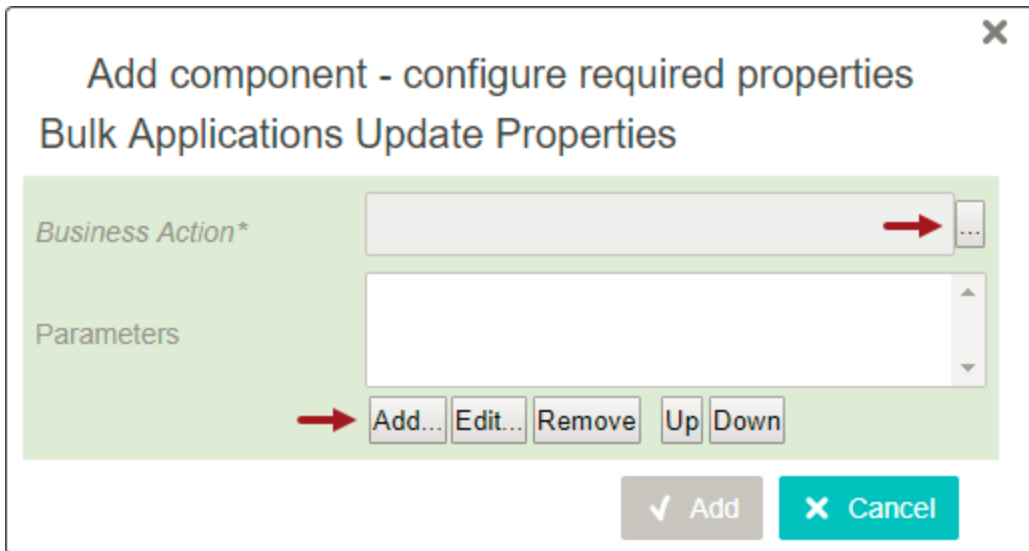
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

Note: After the initial setup, the access to the dialogs does not change, therefore these steps can be helpful when editing the configuration. However, when editing the 'Add' button will be replaced with the 'Save' button, and the 'Add component...' labels at the top of the dialogs will display as 'Edit component... '.

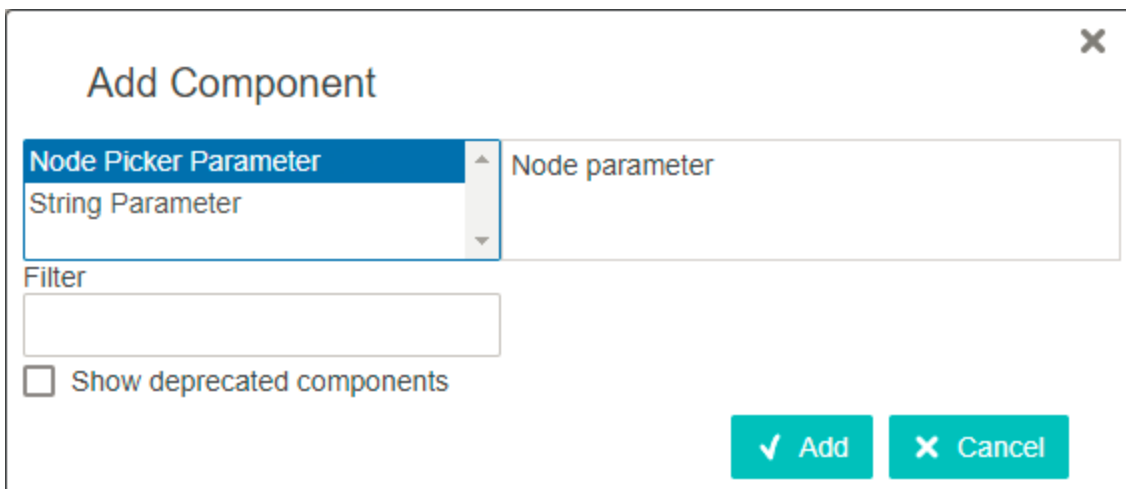
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



4. Click the ellipsis button (...) next to the Business Action parameter > select the business action previously created > click the **OK** button to close the dialog, and return to the Bulk Applications Update Properties dialog. The selected action is displayed in the Business Action parameter.
5. Click the **Add** button beneath the Parameters field, and the Add Component dialog will display (as shown below).

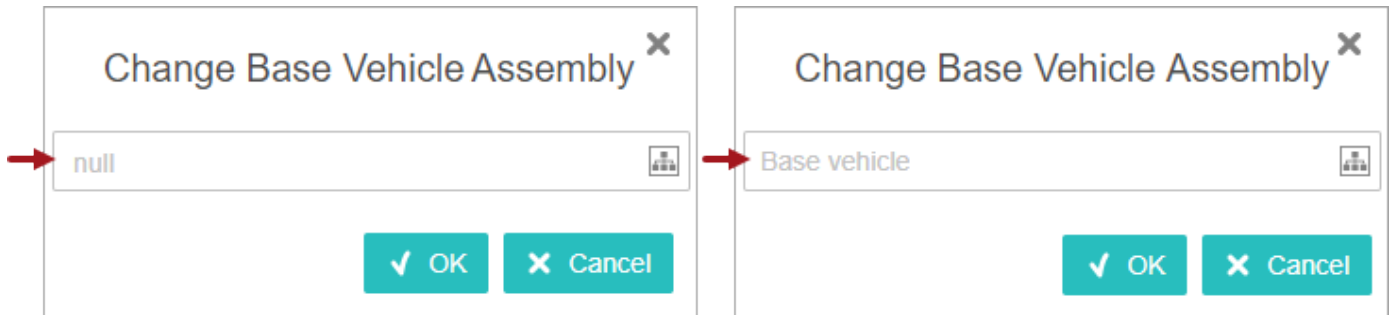


6. Select **Node Picker Parameter** > click the **Add** button to close the dialog, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display (as shown below).

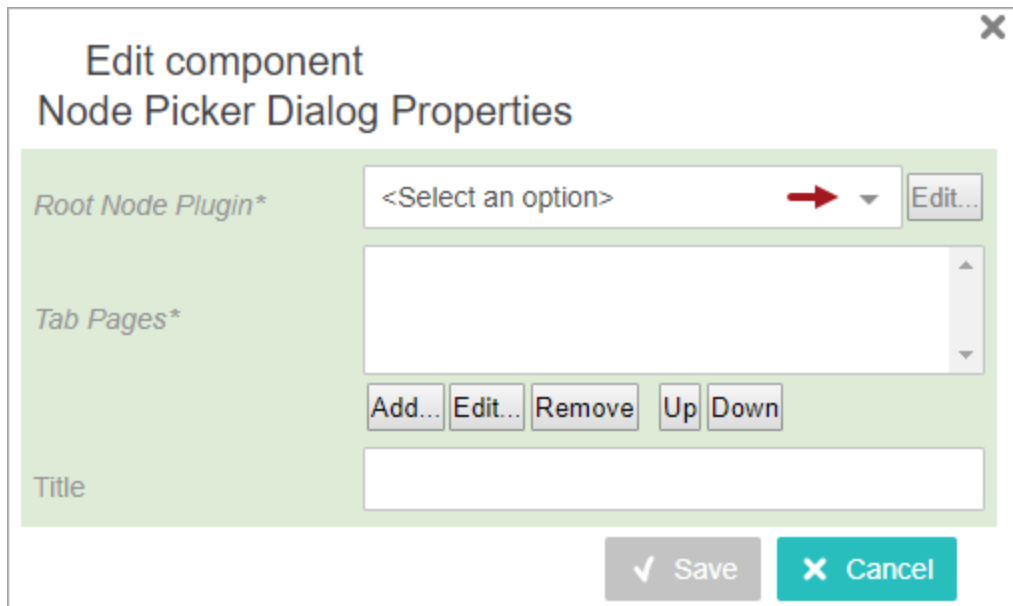
7. Within the Key parameter, enter the exact key created within the 'Part type parameter key' parameter during step 5 of the **Configuring the Change Part Type Business Action in Workbench** topic. This is the only required parameter.

Important: If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

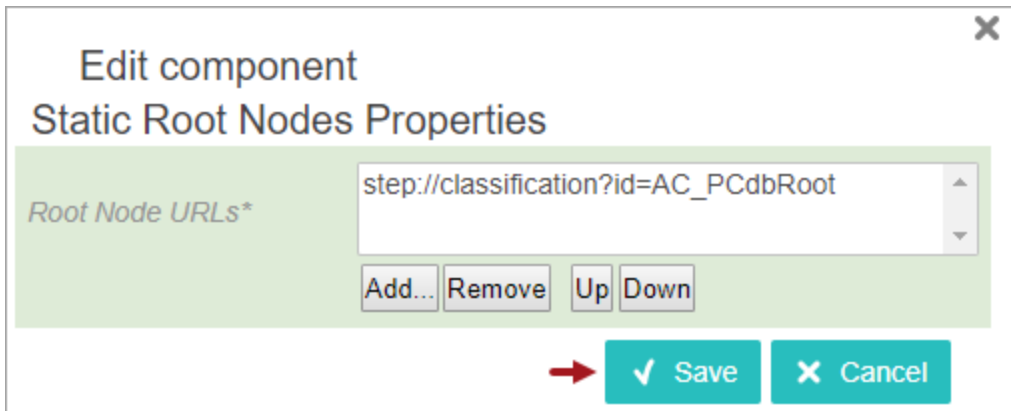
8. Within the Label parameter, enter text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.



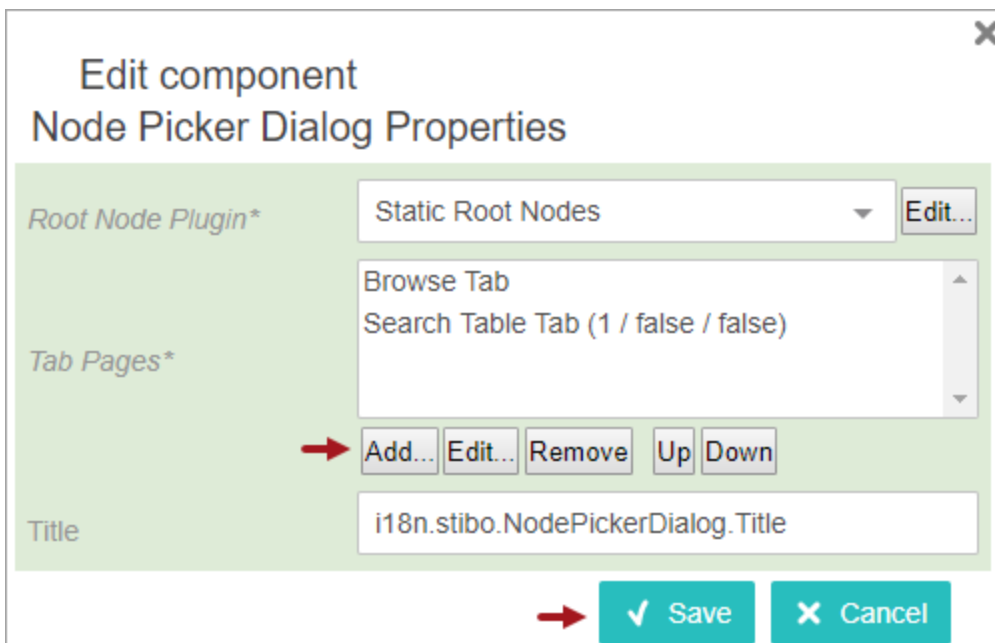
- Use the Node Picker Configuration parameter dropdown to select the **Node Picker Dialog** option, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display.



- Use the Root Node Plugin parameter dropdown to select the **Static Root Nodes** option, and the 'Edit component' for the 'Static Root Nodes Properties' dialog will display.
- Click the **Add** button beneath the Root Nodes URLs parameter > select the Root Nodes that the user should be able to choose from when changing the part type (For this example AC_PCdbRoot is used) > click the **OK** button to close the dialog, and return to the 'Edit component' for 'Static Root Nodes Properties' dialog. Optionally repeat this step to add additional nodes for a user to browse from when looking for a part type.



12. Click the **Save** button, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display with the Root Node Plugin parameter populated with Static Root Nodes.
13. Click the **Add** button beneath the Tab Pages parameter > select the **Browse Tab** component > click the **Add** button to close the dialog and the 'Edit component' for 'Node Picker Dialog Properties' will display with the Tab Pages parameter populated with Browse Tab. Optionally, repeat this step and add the Search Table Tab.

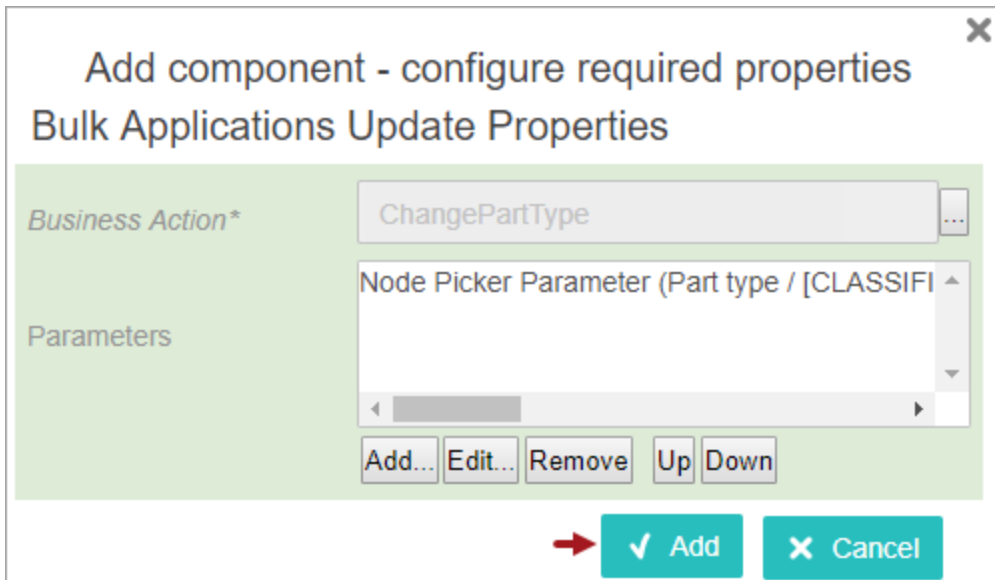


Note: Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, see the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title parameter will display in place of the default 'Select Node(s)' title for the dialog used to select the vehicle assembly.

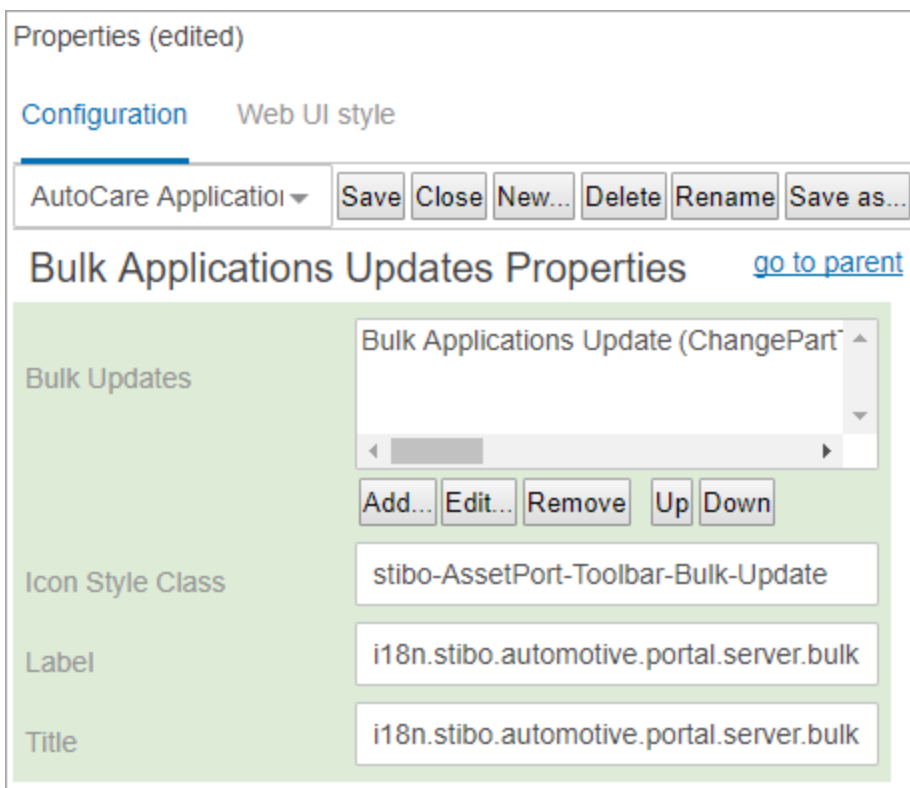
14. Click the **Save** button, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display with the Node Picker Configuration parameter populated.

15. Use the dropdown located beneath the Valid Node Types parameter to select **CLASSIFICATION_TYPE** > click the **Add** button beneath the Valid Node Types parameter so that CLASSIFICATION_TYPE is displayed within the Valid Node Types parameter (as shown below).
16. Click the **Add** button beneath the Valid Object Types parameter > select the desired valid object types (For this example AC_PartTerminology is used)> click the **OK** button to return to the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog.

17. Click the **Add** button, and 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



18. Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



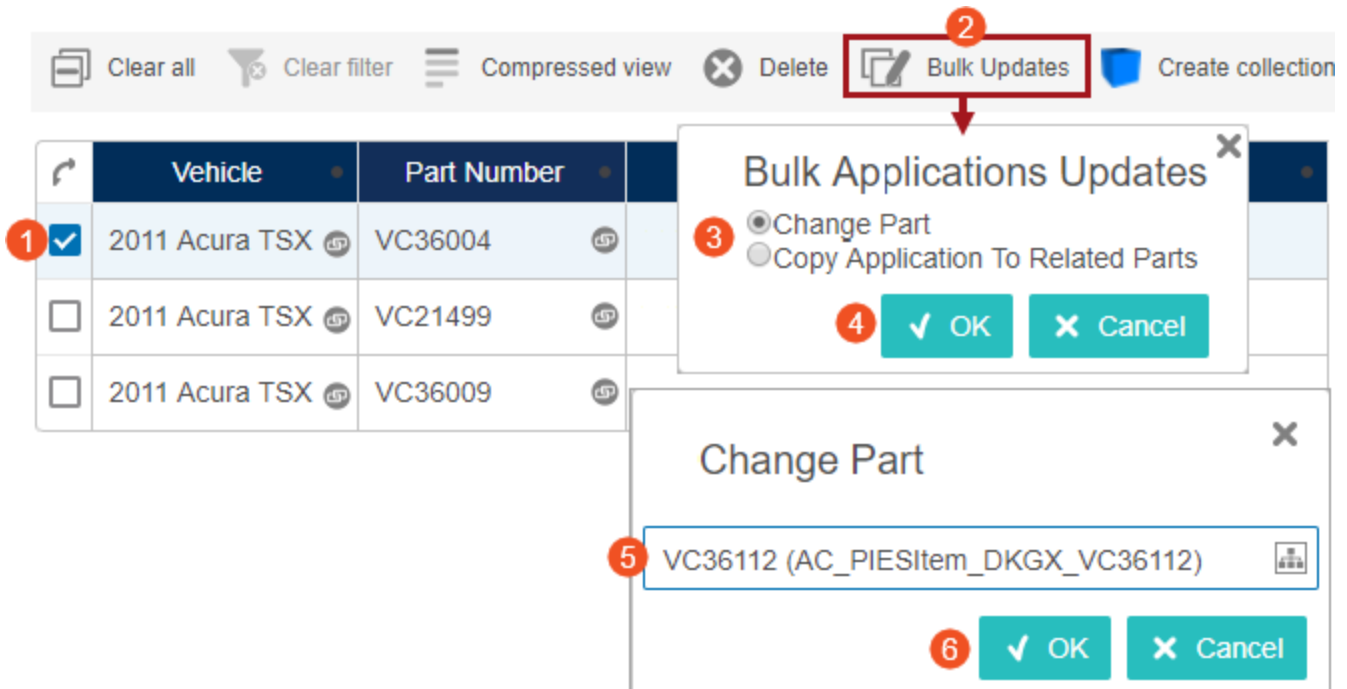
19. Click the **Save** button and then click the **Close** button to close the designer.

To use the newly configured business action, see the **Using the Configured Change Part Type Business Action in Web UI** section of the **Business Action: Change Part Type** topic.

Business Action: Change Part





This automotive business action allows users to change the part of an existing application by clicking a Bulk Updates action button within an Application Manager. However, it cannot change the part of a missing application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, see the **Missing Application Coverage Functionality** topic. Setup is required within both STEP Workbench and Web UI for the action to be available to users.

Once configured, a Bulk Updates button within an Application Manager can be used to change the part for an existing application. Below are the steps for using the business action in Web UI.



Note: If only one bulk update is configured, then the Bulk Applications Updates dialog will not display.

1. Within the configured Application Manager, search for applications and select one or more applications to be changed.
2. Click the **Bulk Updates** action button. If more than one bulk update is configured then the Bulk Updates dialog will display (as shown above), otherwise this dialog is skipped and the Change Part dialog will display (skip to step 5 below).
3. Select the Change Part radio button from the list displayed within the Bulk Application Updates dialog. For this example, 'Change Part' is used, but the business action name displayed within the list is dependent upon the business action's Name parameter.
4. Click the **OK** button and the Change Part dialog will display. For this example, 'Change Part' is used, but the title of this dialog is controlled by the business action's Name parameter.
5. Select the desired part for the application(s).

6. Click the **OK** button to close the dialog, and a background process notification will display.
7. Click the BGP link to view the Background Process Details screen, or look for the gray background process (BGP) notification icon to display with a green dot  or orange dot  Notification icon. Once the business action has completed successfully, the gray background process (BGP) notification icon will display with a green dot . If the business action has failed or completed with errors, the BGP notification icon will display with an orange dot . For more information, see the **Background Process Notification Component** topic.
If the business action BGP fails, access the Background Process Details page for more information.

Note: Before completing this business action, by default, STEP confirms that the application's part type matches one or more part types assigned to the part. If one or more part types associated with the application's part do not match the application's part type, then an error can occur. However, this functionality can be overridden by using the 'Ignore Part Type' parameter within the business action. For more information, see the **Configuring the Change Part Business Action in Workbench** topic.

In the example below, the Background Process Details screen displays the failed BGP with the following error message: Selected part did not match part type from application [application ID].


Background Process Details

ID BGP_104103

Started By USERB

Description Run application bulk update with action: Change Part


Template ID AppBulkUpdService

Status  Failed

Started 10/29/18 9:23:35 AM
















Finished 10/29/18 9:23:35 AM

Elapsed 1 s

 Select all  Export

	ID	Type	Text
<input type="checkbox"/>	10	Error	Selected part did not match part type from application: AC_ACESApp_2626c57b1edf6aa287c3be9d6172e
<input type="checkbox"/>	20	Error	com.stibo.core.domain.businessrule.plugin.BusinessRulePluginException: Selected part did not match part type from application: AC_ACESApp_2626c57b1edf6aa287c3be9d6172e

8. Once the background process has successfully completed, click the **Find applications** button to display the updated application(s) within the Application Manager results table.

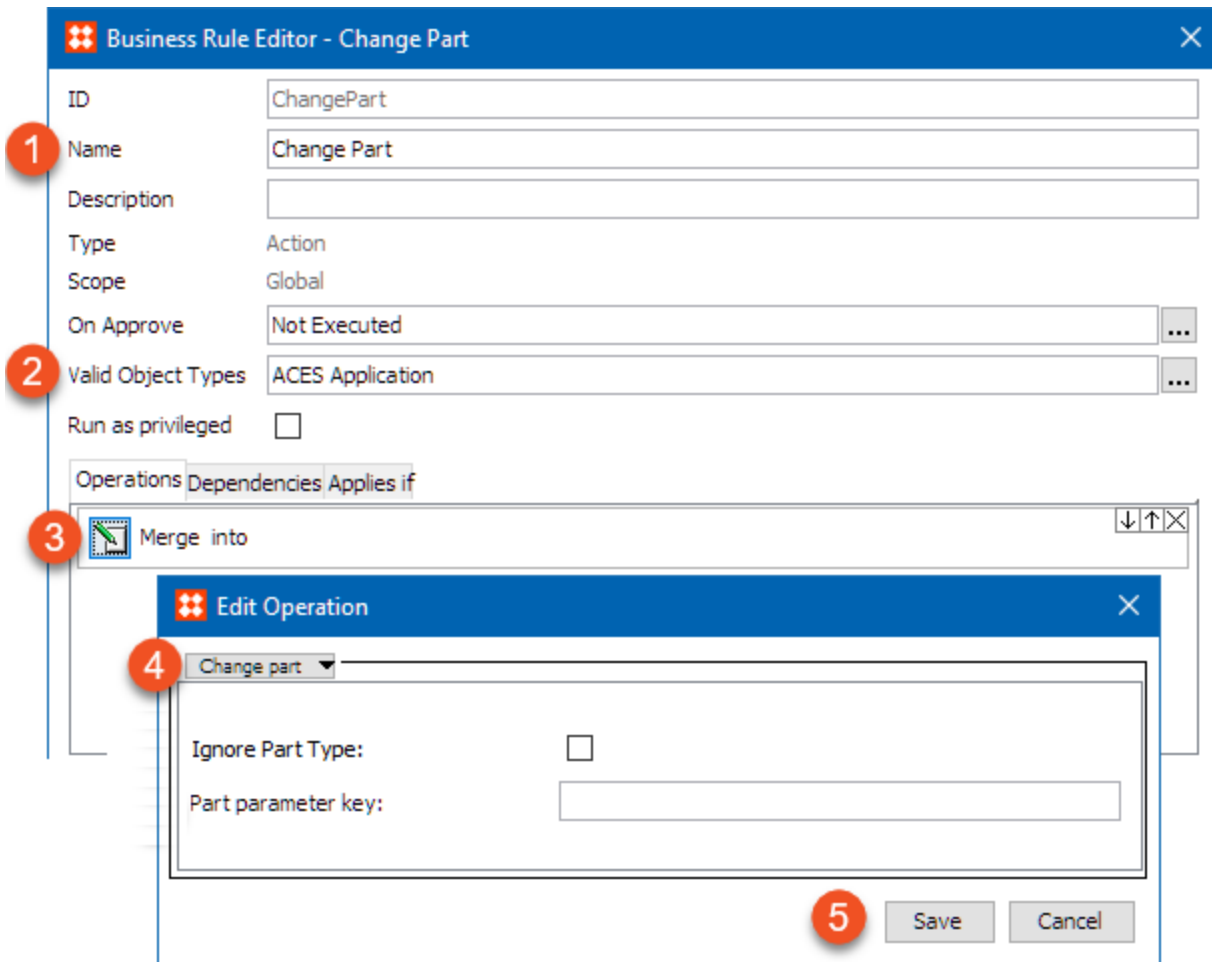
	Vehicle	Part Number		Vehicle	Part Number	
<input checked="" type="checkbox"/>	2011 Acura TSX 	VC36004 		<input checked="" type="checkbox"/>	2011 Acura TSX 	VC36112 
<input type="checkbox"/>	2011 Acura TSX 	VC21499 		<input type="checkbox"/>	2011 Acura TSX 	VC21499 
<input type="checkbox"/>	2011 Acura TSX 	VC36009 		<input type="checkbox"/>	2011 Acura TSX 	VC36009 

Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, see the configuration topics below:

- Configuring the Change Part Business Action in Workbench
- Configuring the Change Part Business Action in Web UI

Configuring the Change Part Business Action in Workbench

The 'Change part' business operation is found within the STEP Workbench Business Rule Editor under the Automotive menu and requires population of a single parameter (Part parameter key). However, setup is required within both STEP Workbench and Web UI for the action to be available to users. This section addresses the steps necessary within the workbench.



1. Create the business action with a name that accurately describes to the user what this action will do. The name of the business action displays within the Web UI and should be easy for the user to identify. For this example, the business action name is 'Change Part.'

Note: The business action Name will display to the Application Manager user once the Bulk Updates action button is selected. If more than one bulk update is configured, then the name displays both within the Bulk Applications Update dialog, and within the change part dialog where the user enters the part for the application (as shown in the example within the **Business Action: Change Part** topic).

2. Edit the new business rule, click the ellipsis button (...) next to the Valid Object Types parameter and select the valid object types for this business action. For this example, the 'ACES Application' object type is selected,

however the TecDoc and NAPA applications can also be selected within the same or separate business actions. This decision is at the discretion of the administrator.

3. On the Operations tab of the Business Rule Editor, click the **Add new Business Action** link, and click the edit button to open the Edit Operation dialog.
4. Use the dropdown menu within the Edit Operation dialog to select **Automotive > Change part** operation, and the following parameters will display (as shown above):
 - **Ignore Part Type:** By default, this optional parameter is disabled. This parameter gives an option to check for a part type or not before changing a part for an application. When disabled, and a user tries to change the part number for an application to a part that does not have the same part type, the action will fail. When enabled, a user can change the part number for an application without concern for the part type.

Background Process Details

ID	BGP_104103
Started By	USERB
Description	Run application bulk update with action: Change Part
Template ID	AppBulkUpdService
Status	▲ Failed
Started	10/29/18 9:23:35 AM
Finished	10/29/18 9:23:35 AM
Elapsed	1 s

Select all Export

	ID	Type		Text
<input type="checkbox"/>	10	Error	•	Selected part did not match part type from application: AC_ACESApp_2626c57b1edf6aa287c3be9d6172e
<input type="checkbox"/>	20	Error		com.stibo.core.domain.businessrule.plugin.BusinessRulePluginException: Selected part did not match part type from application: AC_ACESApp_2626c57b1edf6aa287c3be9d6172e

- **Part Type parameter key:** By default, this required parameter is blank. Within the parameter enter a unique way (key) to identify this rule. Uniqueness is the only restriction for this key. It will not be displayed

to the user. It is case sensitive.

Important: Common setup is to copy the key so it can be pasted in the Web UI designer when configuring the business action in Web UI.

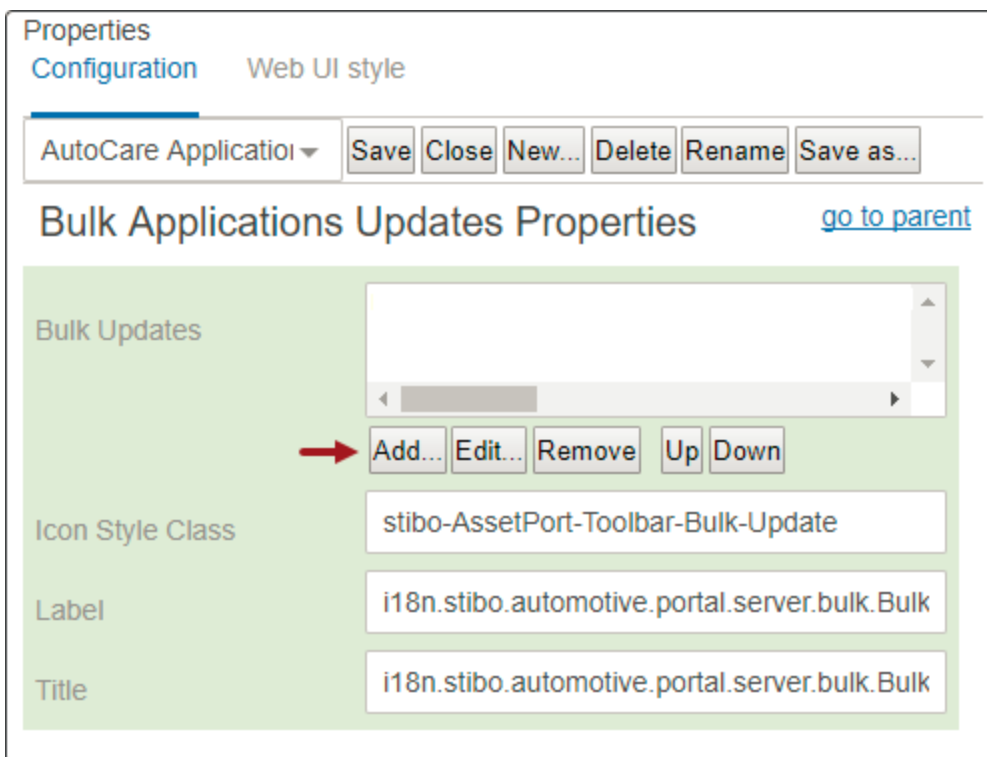
5. Click the **Save** button and continue to the next topic, **Configuring the Change Part Business Action in Web UI**.

Configuring the Change Part Business Action in Web UI

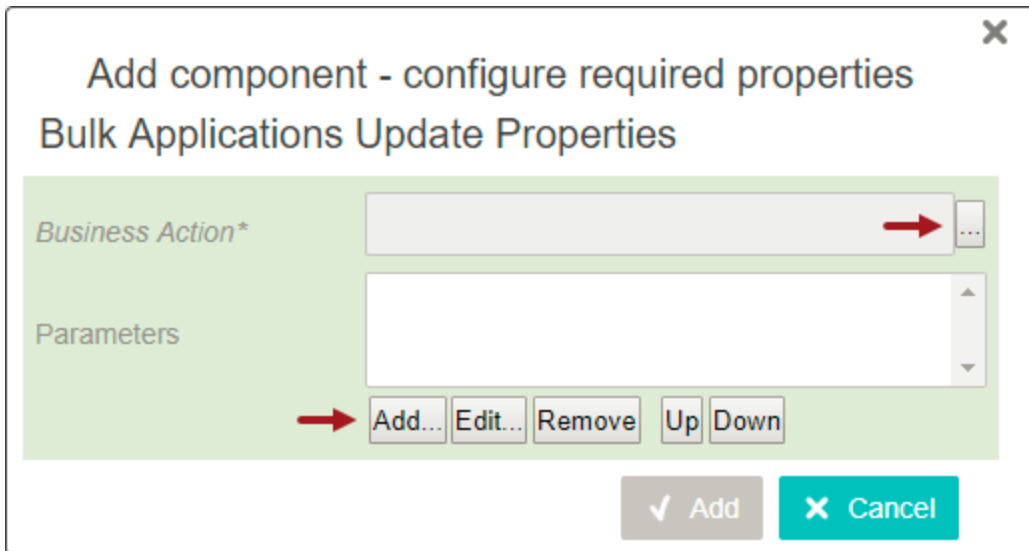
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

Note: After the initial setup, the access to the dialogs does not change, therefore these steps can be helpful when editing the configuration. However, when editing the 'Add' button will be replaced with the 'Save' button, and the 'Add component...' labels at the top of the dialogs will display as 'Edit component... '.

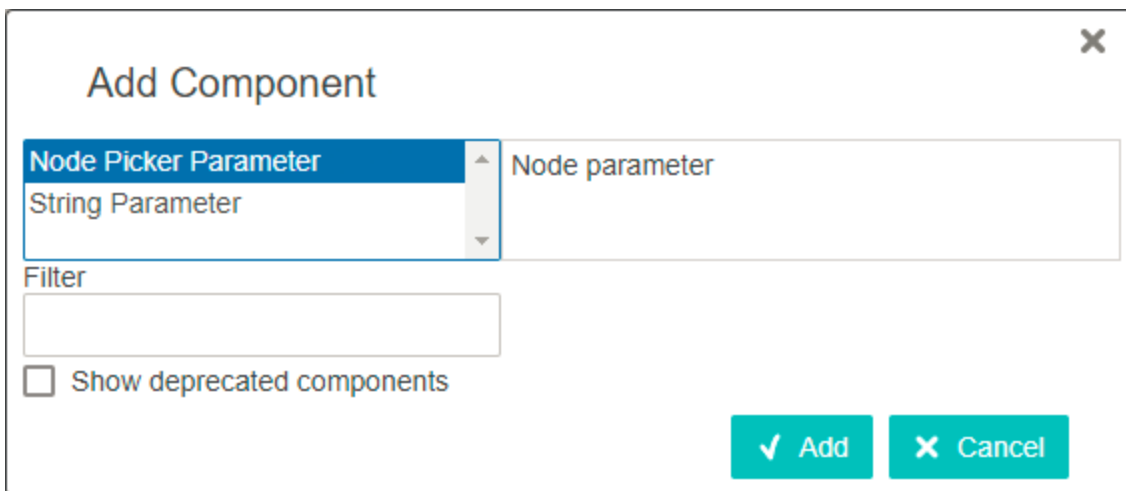
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



4. Click the ellipsis button (...) next to the Business Action parameter > select the business action previously created > click the **OK** button to close the dialog, and return to the Bulk Applications Update Properties dialog. The selected action is displayed in the Business Action parameter.
5. Click the **Add** button beneath the Parameters field, and the Add Component dialog will display (as shown below).

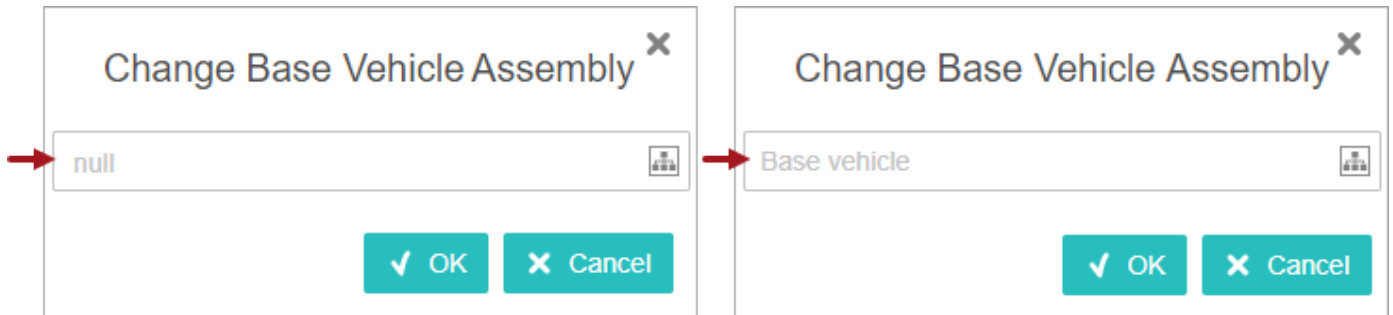


6. Select **Node Picker Parameter** > click the **Add** button to close the dialog, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display (as shown below).

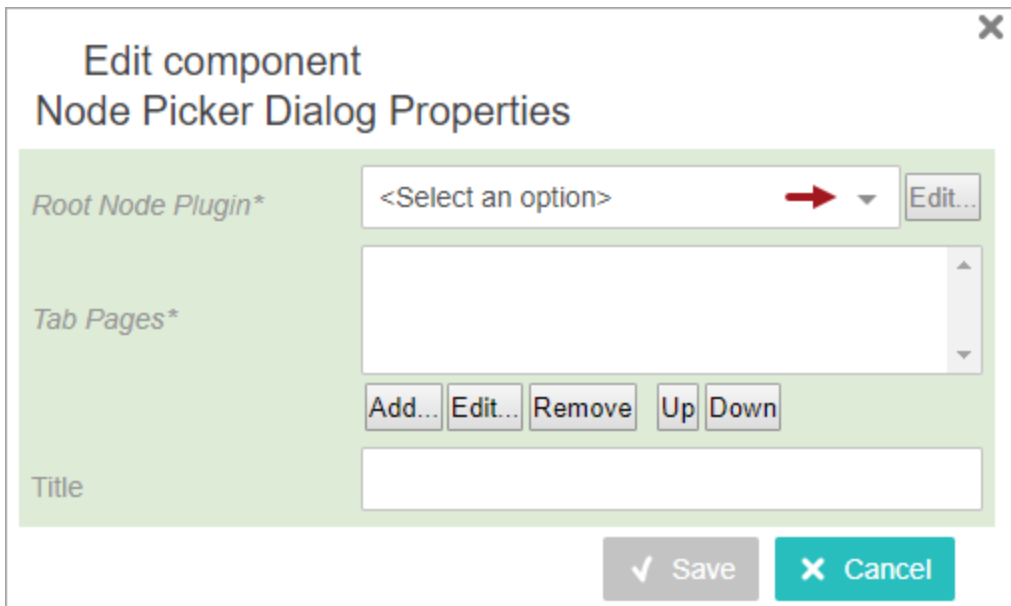
7. Within the Key parameter, enter the exact key created within the 'Part parameter key' parameter during step 5 of the **Configuring the Change Part Business Action in Workbench** topic. This is the only required parameter.

Important: If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

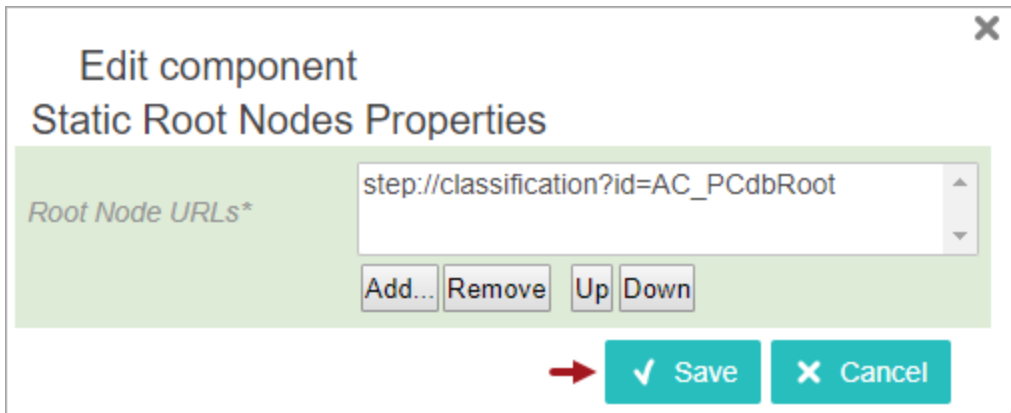
8. Within the Label parameter, enter text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.



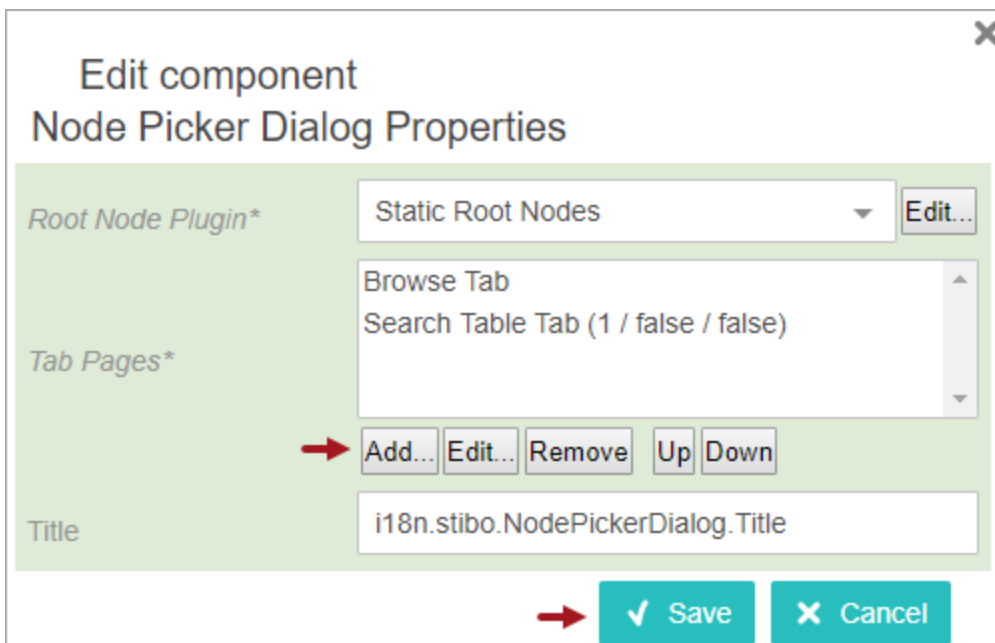
- Use the Node Picker Configuration parameter dropdown to select the **Node Picker Dialog** option, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display.



- Use the Root Node Plugin parameter dropdown to select the **Static Root Nodes** option, and the 'Edit component' for the 'Static Root Nodes Properties' dialog will display.
- Click the **Add** button beneath the Root Nodes URLs parameter > select the Root Nodes that the user should be able to choose from when changing the part (For this example AC_PCdbRoot is used) > click the **OK** button to close the dialog, and return to the 'Edit component' for 'Static Root Nodes Properties' dialog. Optionally repeat this step to add additional nodes for a user to browse from when looking for a part type.



12. Click the **Save** button, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display with the Root Node Plugin parameter populated with Static Root Nodes.
13. Click the **Add** button beneath the Tab Pages parameter > select the **Browse Tab** component > click the **Add** button to close the dialog and the 'Edit component' for 'Node Picker Dialog Properties' will display with the Tab Pages parameter populated with Browse Tab. Optionally, repeat this step and add the Search Table Tab.



Note: Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, see the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title parameter will display in place of the default 'Select Node(s)' title for the dialog used to select the vehicle assembly.

14. Click the **Save** button, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display with the Node Picker Configuration parameter populated.

15. Use the dropdown located beneath the Valid Node Types parameter to select **PRODUCT_TYPE** > click the **Add** button beneath the Valid Node Types parameter so that PRODUCT_TYPE is displayed within the Valid Node Types parameter (as shown below).
16. Click the **Add** button beneath the Valid Object Types parameter > select the desired valid object types (For this example AC_PIESItem is used) > click the **OK** button to return to the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog.

Add component - configure required properties

Node Picker Parameter Properties

Key*

Label

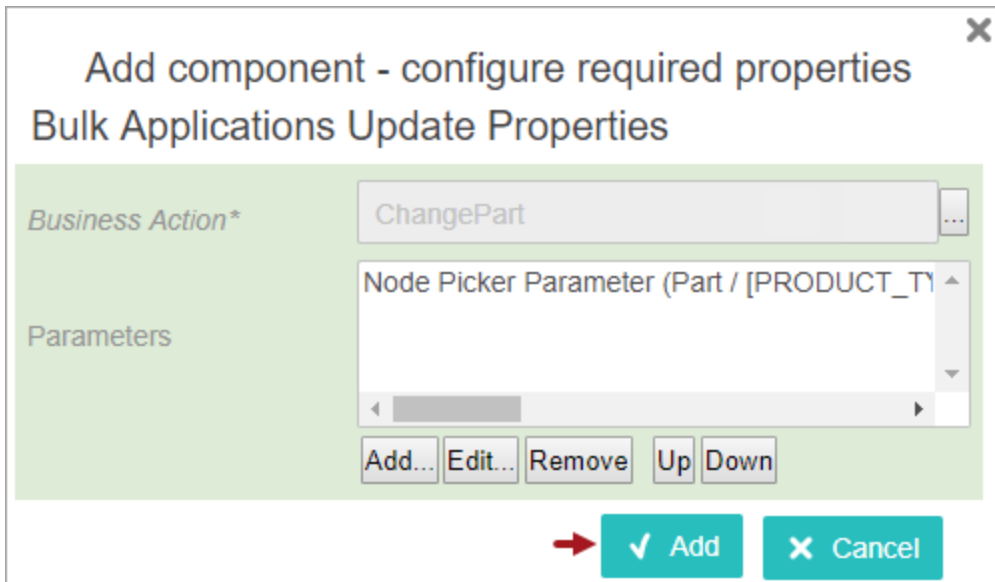
Mandatory

Node Picker Configuration

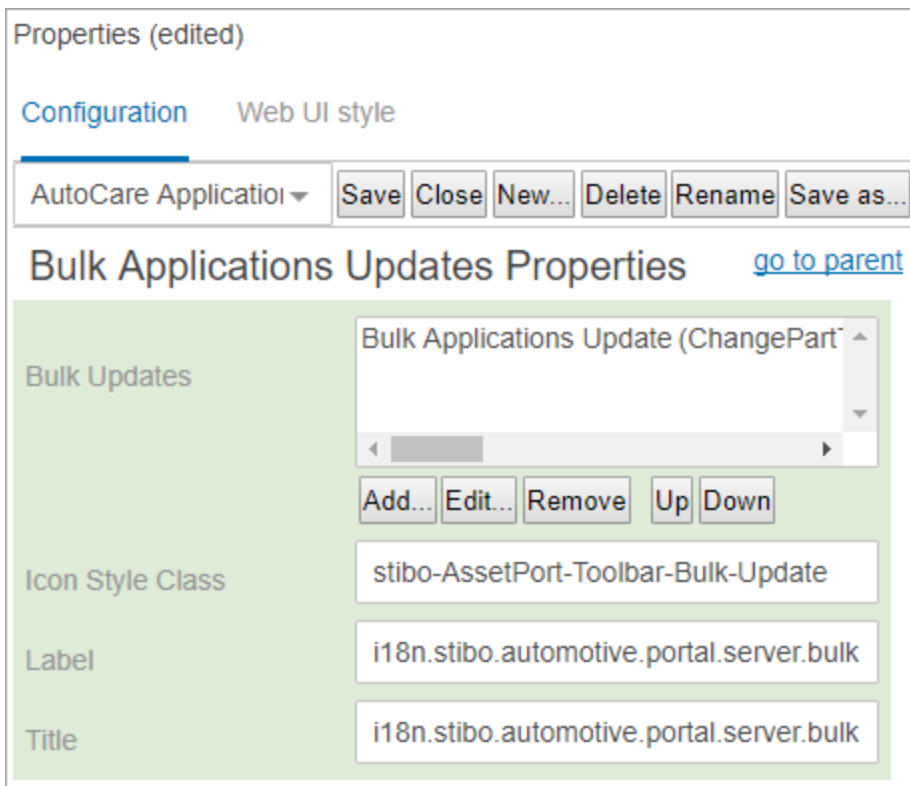
Valid Node Types

Valid Object Types

17. Click the **Add** button, and 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



- Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



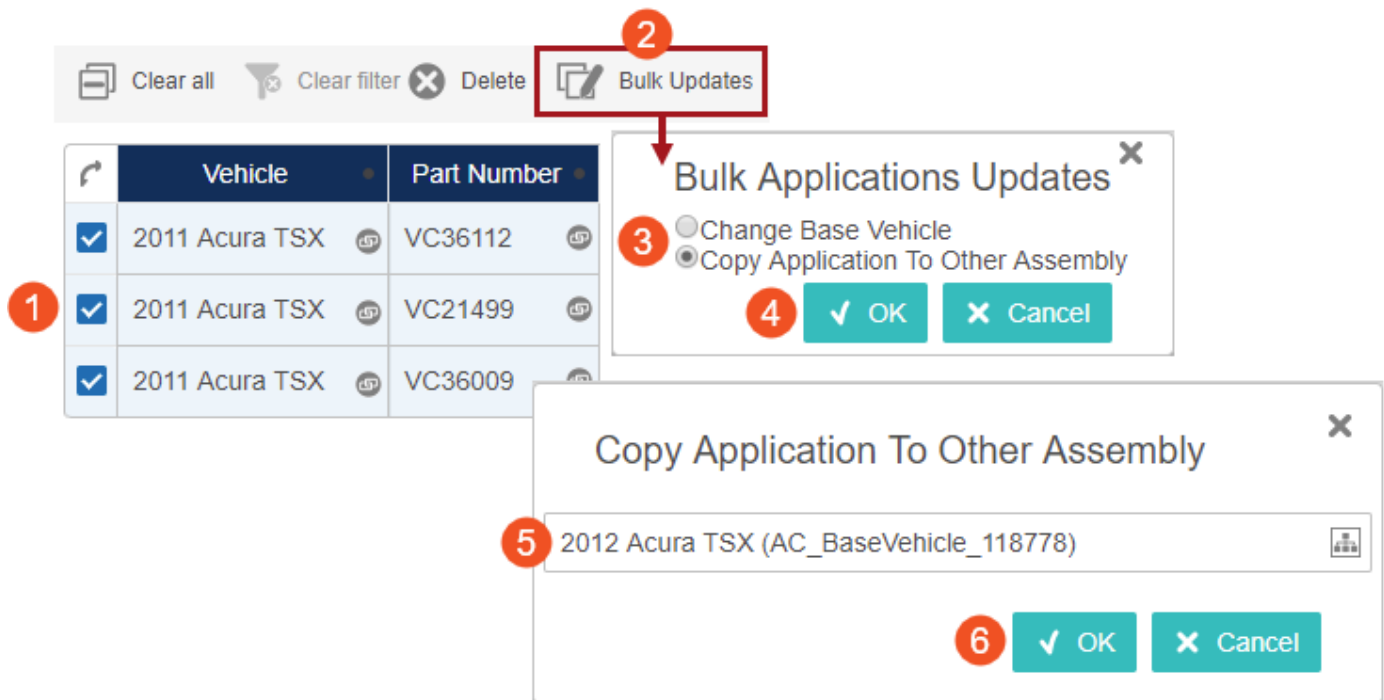
- Click the **Save** button and then click the **Close** button to close the designer.

To use the newly configured business action, see the **Business Action: Change Part** topic.

Business Action: Copy Application to Other Assembly

This automotive business action allows users to use a Bulk Updates action button within an Application Manager to copy one or more existing applications to another assembly / vehicle. However, it will not copy an assembly / vehicle that is missing an application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, see the **Missing Application Coverage Functionality** topic within this guide. Setup is required within both STEP Workbench and Web UI for the action to be available to users.

Once configured, a Bulk Updates button within an Application Manager can be used to copy one or more existing applications to another assembly / vehicle. Below are the steps for using the business action in Web UI.



Note: If only one bulk update is configured, then the Bulk Applications Updates dialog will not display.

1. Within the configured Application Manager, search for applications and select one or more applications to be copied.
2. Click the **Bulk Updates** action button. If more than one bulk update is configured, then the Bulk Updates dialog will display (as shown above), otherwise this dialog is skipped and the Copy Application To Other Assembly dialog will display (skip to step 5 below).
3. Select the Copy Application To Other Assembly radio button from the list displayed within the Bulk Application Updates dialog. For this example, 'Copy Application To Other Assembly' is used, but the business action name displayed within the list is dependent upon the business action's Name parameter.

4. Click the **OK** button and the Copy Application To Other Assembly dialog will display. For this example, 'Copy Application To Other Assembly' is used, but the title of this dialog is controlled by the business action's Name parameter.
5. Select the desired assembly for the application(s).
6. Click the **OK** button to close the dialog, and a background process notification will display.
7. Once the background process has completed, click the **Find applications** button to display the newly created application(s) within the Application Manager results table.

	Vehicle	Part Number
✓	2011 Acura TSX	VC36112
✓	2011 Acura TSX	VC21499
✓	2011 Acura TSX	VC36009

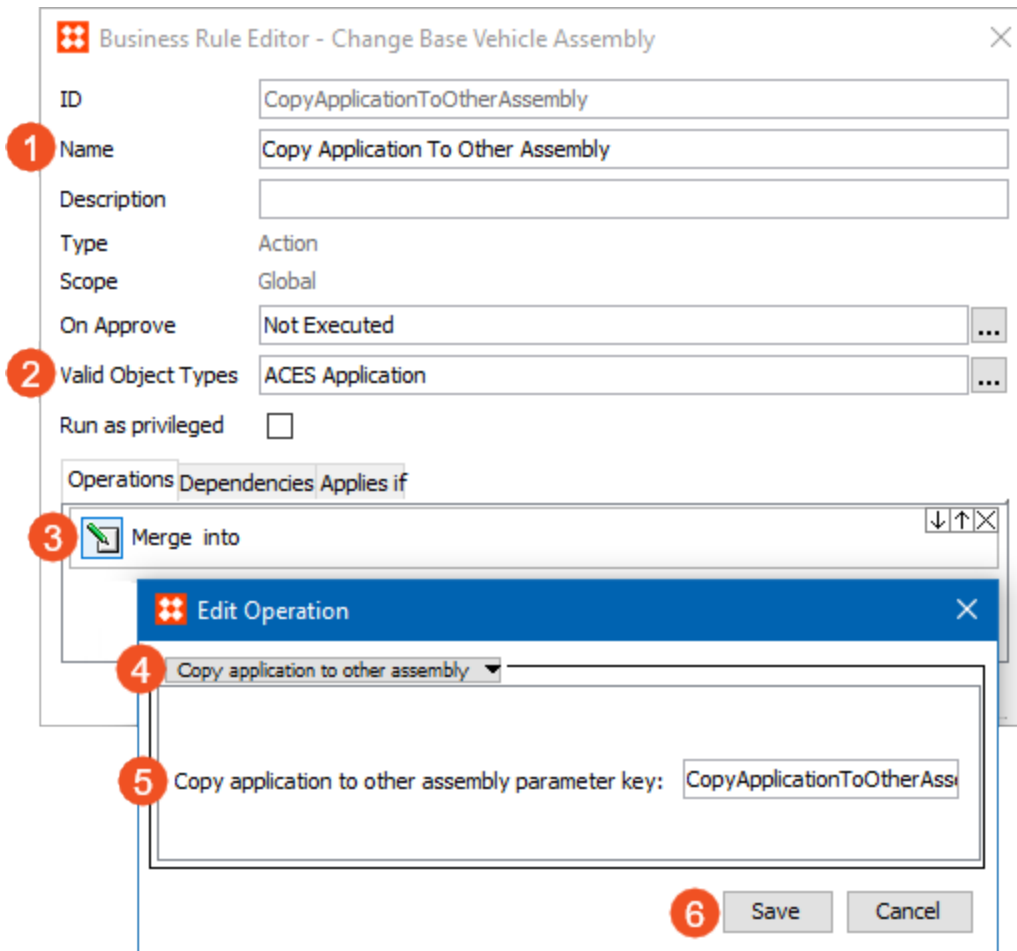
	Vehicle	Part Number
✓	2011 Acura TSX	VC36112
✓	2011 Acura TSX	VC21499
✓	2011 Acura TSX	VC36009
✓	2012 Acura TSX	VC21499
✓	2012 Acura TSX	VC36009
✓	2012 Acura TSX	VC36112

Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, see the configuration topics below:

- Configuring the Copy Application to Other Assembly Business Action in Workbench
- Configuring the Copy Application to Other Assembly Business Action in Web UI

Configuring the Copy Application to Other Assembly Business Action in Workbench

The 'Copy application to other assembly' business operation is found within the STEP Workbench Business Rule Editor under the Automotive menu and requires population of a single parameter (Copy application to other assembly parameter key). However, setup is required within both STEP Workbench and Web UI for the action to be available to users. This section addresses the steps necessary within the workbench.



1. Create the business action with a name that accurately describes to the user what this action will do. The name of the business action displays within the Web UI and should be easy for the user to identify. For this example, the business action name is 'Copy Application To Other Assembly.'

Note: The business action Name will display to the Application Manager user once the Bulk Updates action button is selected. If more than one bulk update is configured, then the name displays both within the Bulk Applications Update dialog, and within the copy application to other assembly dialog where the user enters the assembly for the application (as shown in the example within the **Using the Configured Copy Application To Other Assembly Business Action in Web UI** section of the **Business Action: Copy Application To Other Assembly** topic).

2. Edit the new business rule, click the ellipsis button (...) next to the Valid Object Types parameter and select the valid object types for this business action. For this example, the 'ACES Application' object type is selected, however the TecDoc and NAPA applications can also be selected within the same or separate business actions. This decision is at the discretion of the administrator.
3. On the Operations tab of the Business Rule Editor, click the **Add new Business Action** link, and click the edit button to open the Edit Operation dialog.
4. Use the dropdown menu within the Edit Operation dialog to select **Automotive > Copy application to other assembly** operation, and the parameter 'Copy application to other assembly parameter key' will display (as shown above).
5. Within the parameter enter a unique way (key) to identify this rule. Uniqueness is the only restriction for this key. It will not be displayed to the user. It is case sensitive.

Important: Common setup is to copy the key so it can be pasted in the Web UI designer when configuring the business action in Web UI.

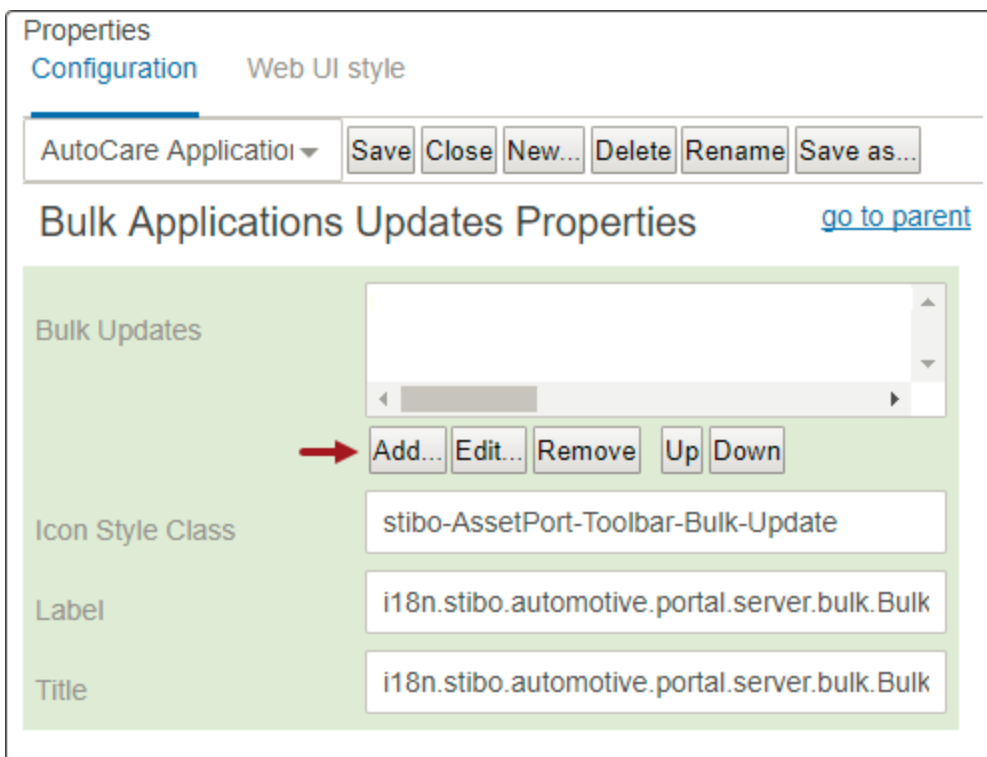
6. Click the **Save** button and continue to the next topic, **Copy Application to Other Assembly Business Action in Web UI**.

Configuring the Copy Application to Other Assembly Business Action in Web UI

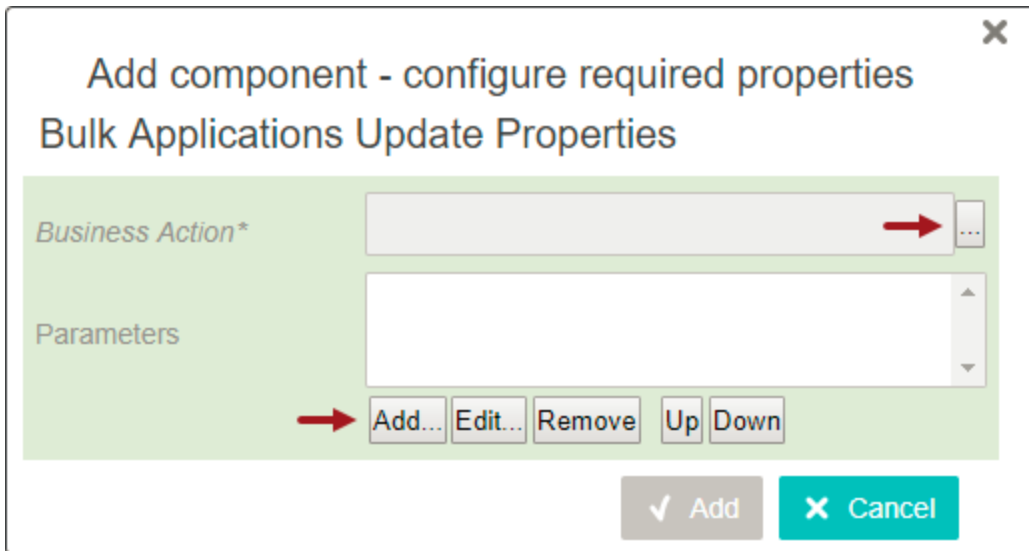
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

Note: After the initial setup, the access to the dialogs does not change, therefore these steps can be helpful when editing the configuration. However, when editing the 'Add' button will be replaced with the 'Save' button, and the 'Add component...' labels at the top of the dialogs will display as 'Edit component... '.

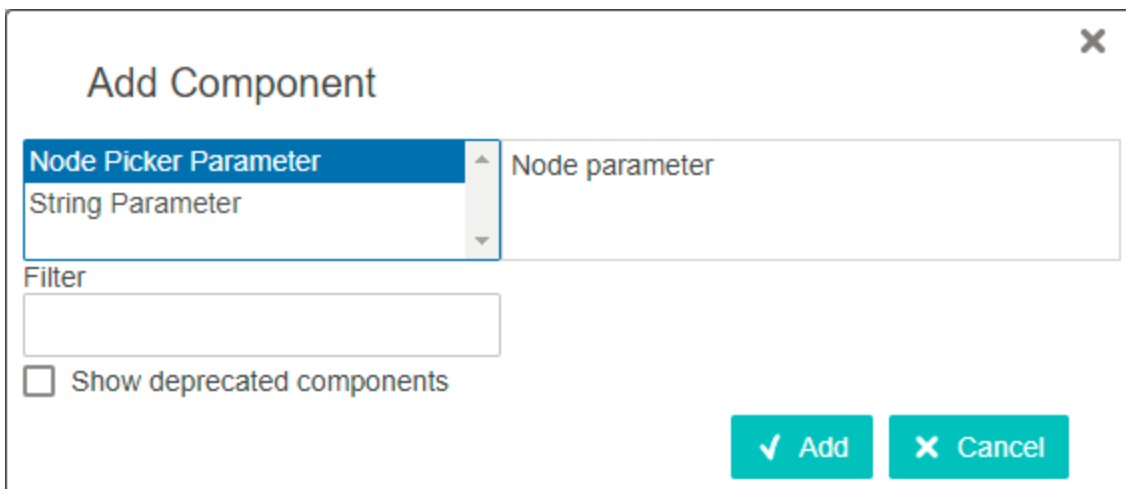
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



4. Click the ellipsis button (...) next to the Business Action parameter > select the business action previously created > click the **OK** button to close the dialog, and return to the Bulk Applications Update Properties dialog. The selected action is displayed in the Business Action parameter.
5. Click the **Add** button beneath the Parameters field, and the Add Component dialog will display (as shown below).

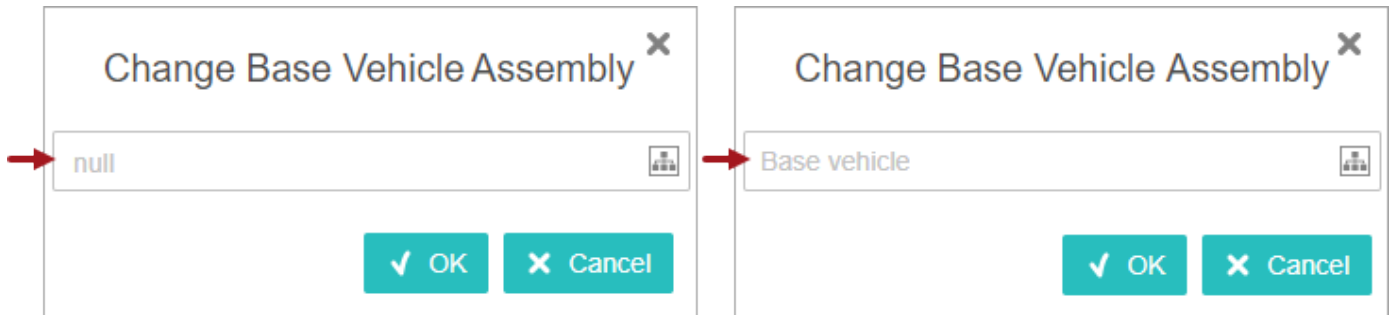


6. Select **Node Picker Parameter** > click the **Add** button to close the dialog, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display (as shown below).

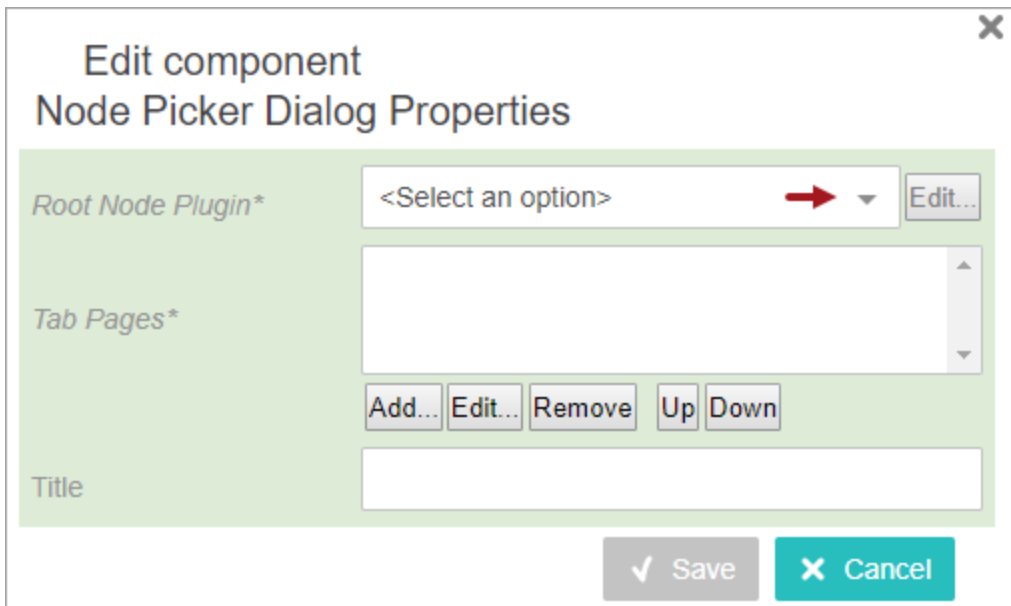
7. Within the Key parameter, enter the exact key created within the 'Assembly parameter key' parameter during step 5 of the **Configuring the Copy Application to Other Assembly Business Action in Workbench** topic. This is the only required parameter.

Important: If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

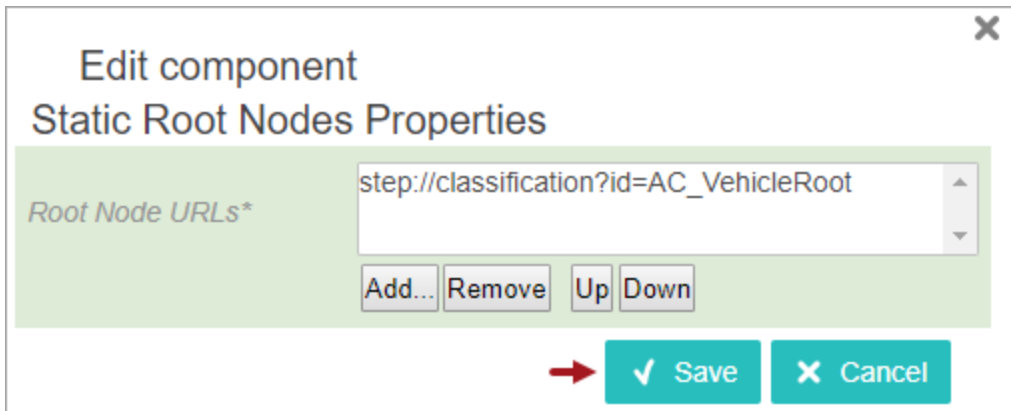
8. Within the Label parameter, enter text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.



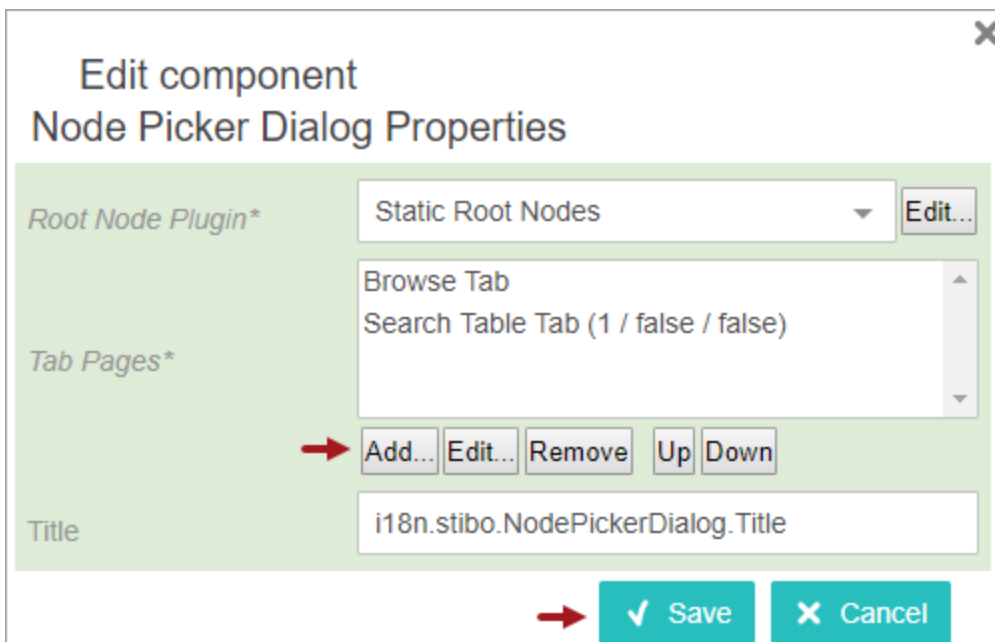
- Use the Node Picker Configuration parameter dropdown to select the **Node Picker Dialog** option, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display.



- Use the Root Node Plugin parameter dropdown to select the **Static Root Nodes** option, and the 'Edit component' for the 'Static Root Nodes Properties' dialog will display.
- Click the **Add** button beneath the Root Nodes URLs parameter > select the Root Nodes that the user should be able to choose from when changing the part type (For this example AC_VehicleRoot is used) > click the **OK** button to close the dialog, and return to the 'Edit component' for 'Static Root Nodes Properties' dialog. Optionally repeat this step to add additional nodes for a user to browse from when looking for a part type.



12. Click the **Save** button, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display with the Root Node Plugin parameter populated with Static Root Nodes.
13. Click the **Add** button beneath the Tab Pages parameter > select the **Browse Tab** component > click the **Add** button to close the dialog and the 'Edit component' for 'Node Picker Dialog Properties' will display with the Tab Pages parameter populated with Browse Tab. Optionally, repeat this step and add the Search Table Tab.



Note: Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, see the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title parameter will display in place of the default 'Select Node(s)' title for the dialog used to select the vehicle assembly.

14. Click the **Save** button, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display with the Node Picker Configuration parameter populated.

15. Use the dropdown located beneath the Valid Node Types parameter to select **CLASSIFICATION_TYPE** > click the **Add** button so that CLASSIFICATION_TYPE is displayed within the Valid Node Types parameter (as shown below).
16. Click the **Add** button beneath the Valid Object Types parameter > select the desired valid object types > click the **OK** button to return to the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog.

Add component - configure required properties

Node Picker Parameter Properties

Key*

Label

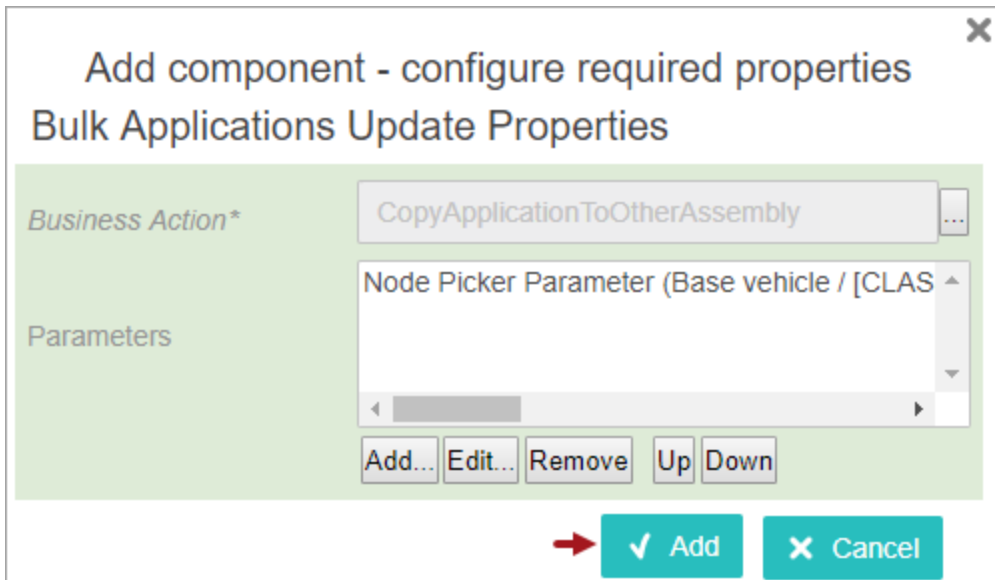
Mandatory

Node Picker Configuration

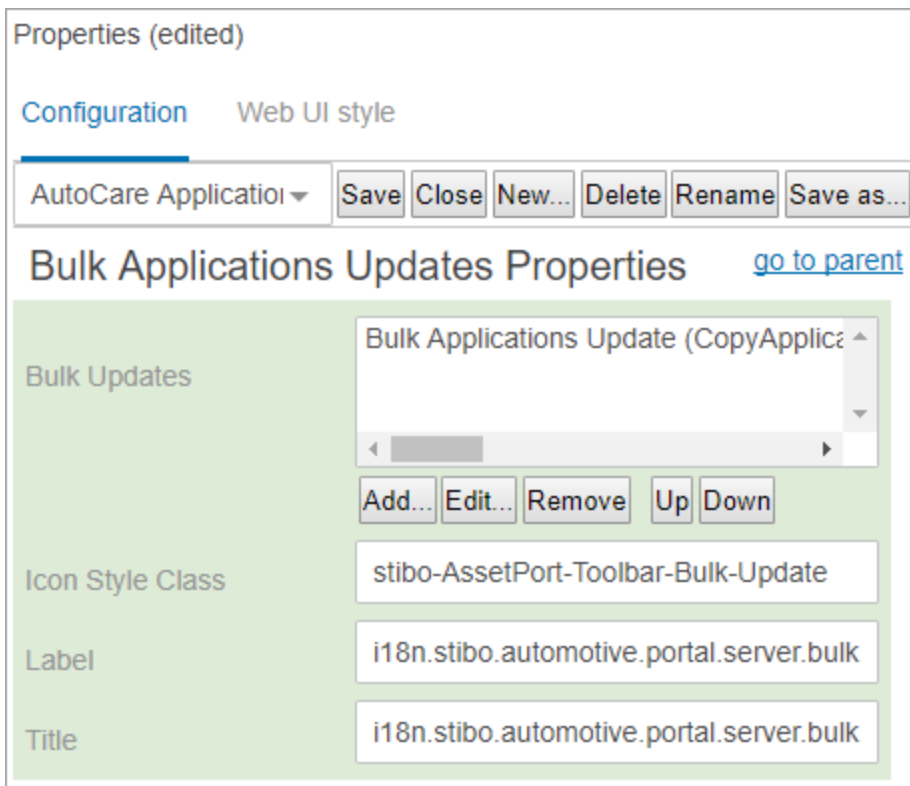
Valid Node Types

Valid Object Types

17. Click the **Add** button, and 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



- Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



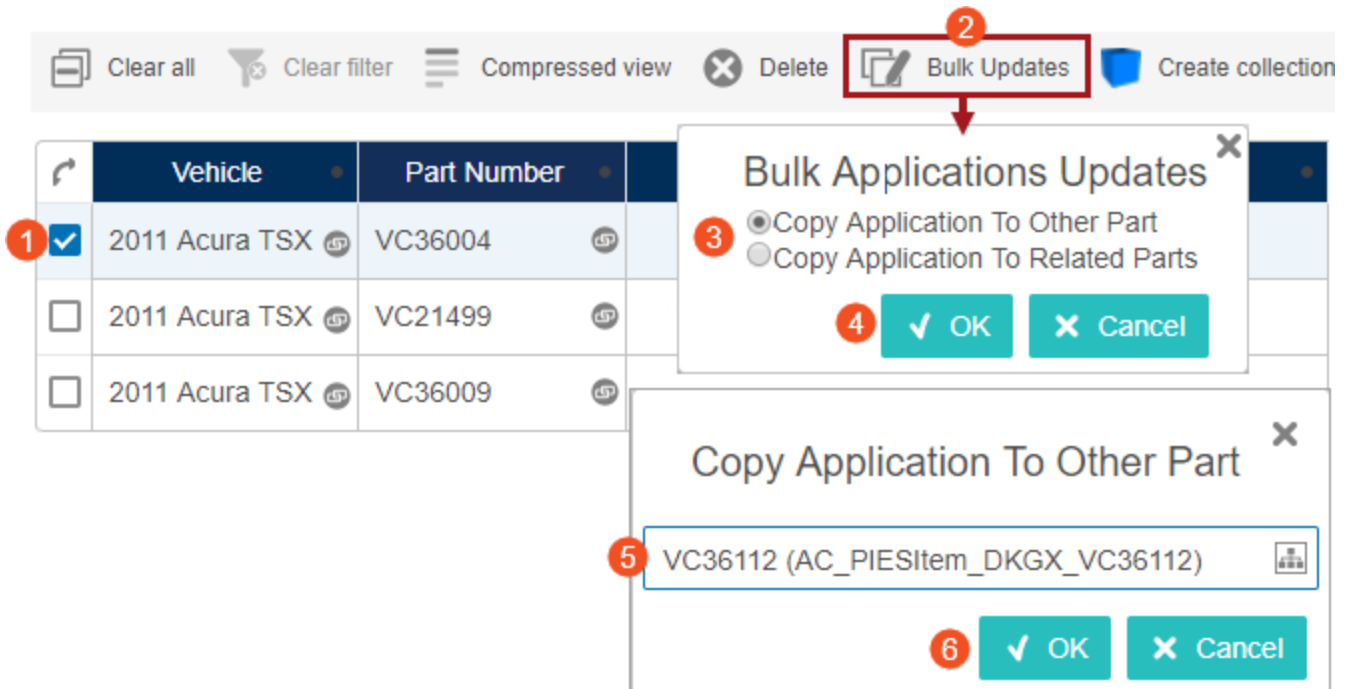
- Click the **Save** button and then click the **Close** button to close the designer.

To use the newly configured business action, see the **Using the Configured Copy Application to Other Assembly Business Action in Web UI** section of the **Business Action: Copy Application to Other Assembly** topic.

Business Action: Copy Application to Other Part





This automotive business action allows users to copy one or more existing applications to another part by clicking a Bulk Updates action button within an Application Manager. However, it cannot copy the part of a missing application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, see the **Missing Application Coverage Functionality** topic. Setup is required within both STEP Workbench and Web UI for the action to be available to users.

Once configured, a Bulk Updates button within an Application Manager can be used to copy one or more existing applications to another part. Below are the steps for using the business action in Web UI.



Note: If only one bulk update is configured, then the Bulk Applications Updates dialog will not display.

1. Within the configured Application Manager, search for applications and select one or more applications to be copied.
2. Click the **Bulk Updates** action button. If more than one bulk update is configured then the Bulk Updates dialog will display (as shown above), otherwise this dialog is skipped and the Copy Application To Other Part dialog will display (skip to step 5 below).
3. Select the Copy Application To Other Part radio button from the list displayed within the Bulk Application Updates dialog. For this example, 'Copy Application to Other Part' is used, but the business action name displayed within the list is dependent upon the business action's Name parameter.
4. Click the **OK** button and the Copy Application To Other Part dialog will display. For this example, 'Copy Application To Other Part' is used, but the title of this dialog is controlled by the business action's Name parameter.

5. Select the desired part for the application(s).
6. Click the **OK** button to close the dialog, and a background process notification will display.
7. Click the BGP link to view the Background Process Details screen, or look for the gray background process (BGP) notification icon to display with a green dot  or orange dot  Notification icon. Once the business action has completed successfully, the gray background process (BGP) notification icon will display with a green dot . If the business action has failed or completed with errors, the BGP notification icon will display with an orange dot . For more information, see the **Background Process Notification Component** topic.

If the business action BGP fails, access the Background Process Details page for more information.

Note: Before completing this business action, by default, STEP confirms that the application's part type matches one or more part types assigned to the part. If one or more part types associated with the application's part do not match the application's part type, then an error can occur. However, this functionality can be overridden by using the 'Ignore Part Type' parameter within the business action. For more information, see the **Configuring the Copy Application to Other Part Business Action in Workbench** topic.

In the example below, the Background Process Details screen displays the failed BGP with the following error message: Selected part did not match part type from application [application ID].


Background Process Details

ID BGP_104103

Started By USERB

Description Run application bulk update with action: Change Part



Template ID AppBulkUpdService

Status  Failed

Started 10/29/18 9:23:35 AM

Finished 10/29/18 9:23:35 AM

Elapsed 1 s

 Select all  Export

	ID	Type	Text
<input type="checkbox"/>	10	Error	Selected part did not match part type from application: AC_ACESApp_2626c57b1edf6aa287c3be9d6172e
<input type="checkbox"/>	20	Error	com.stibo.core.domain.businessrule.plugin.BusinessRulePluginException: Selected part did not match part type from application: AC_ACESApp_2626c57b1edf6aa287c3be9d6172e

8. Once the background process has completed, click the **Find applications** button to display the newly created application(s) within the Application Manager results table.

	Vehicle	Part Number
<input checked="" type="checkbox"/>	2011 Acura TSX	VC36004
<input type="checkbox"/>	2011 Acura TSX	VC21499
<input type="checkbox"/>	2011 Acura TSX	VC36009

→

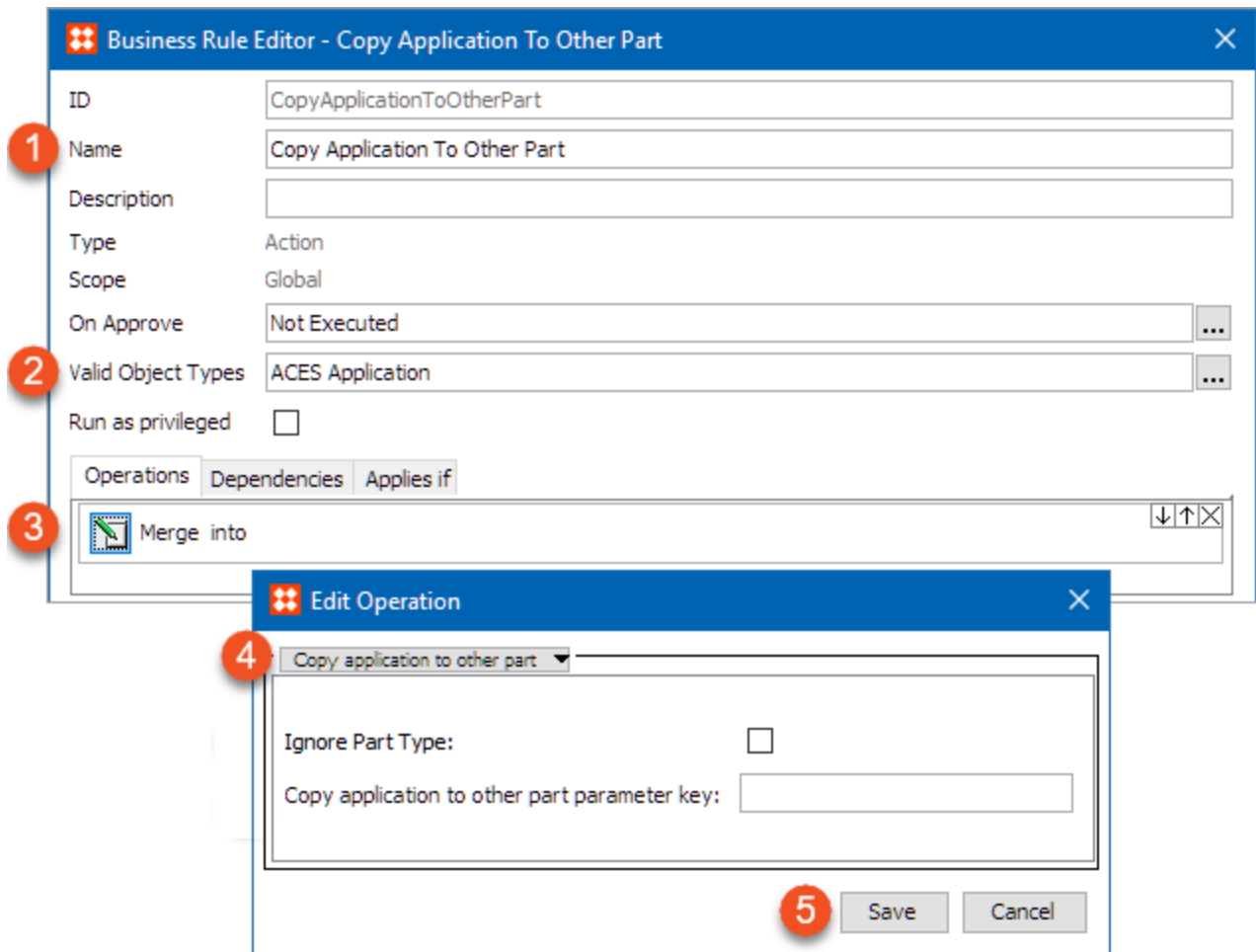
	Vehicle	Part Number
<input checked="" type="checkbox"/>	2011 Acura TSX	VC36112
<input type="checkbox"/>	2011 Acura TSX	VC21499
<input type="checkbox"/>	2011 Acura TSX	VC36009

Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, see the configuration topics below:

- Configuring the Copy Application to Other Part Business Action in Workbench
- Configuring the Copy Application to Other Part Business Action in Web UI

Configuring the Copy Application to Other Part Business Action in Workbench

The 'Copy application to other part' business operation is found within the STEP Workbench Business Rule Editor under the Automotive menu and requires population of a single parameter (Copy application to other part parameter key). However, setup is required within both STEP Workbench and Web UI for the action to be available to users. This section addresses the steps necessary within the workbench.



1. Create the business action with a name that accurately describes to the user what this action will do. The name of the business action displays within the Web UI and should be easy for the user to identify. For this example, the business action name is 'Copy Application To Other Part.'

Note: The business action Name will display to the Application Manager user once the Bulk Updates action button is selected. If more than one bulk update is configured, then the name displays both within the Bulk Applications Update dialog, and within the change part dialog where the user enters the part for the application (as shown in the example within the **Using the Configured Copy Application To Other Part Business Action in Web UI** section of the **Business Action: Copy Application To Other Part** topic).

2. Edit the new business rule, click the ellipsis button (...) next to the Valid Object Types parameter and select the valid object types for this business action. For this example, the 'ACES Application' object type is selected, however the TecDoc and NAPA applications can also be selected within the same or separate business actions. This decision is at the discretion of the administrator.
3. On the Operations tab of the Business Rule Editor, click the **Add new Business Action** link, and click the edit button to open the Edit Operation dialog.
4. Use the dropdown menu within the Edit Operation dialog to select **Automotive > Copy application to other part** operation, and the following parameters will display (as shown above):
 - **Ignore Part Type:** By default, this optional parameter is disabled. This parameter gives an option to check for a part type or not before copying an application for a part. If unchecked, then trying to copy an application to another part that does not have the same part type will fail. If checked, User can copy an application to a part of same and / or different part type.
 - **Copy application to other part parameter key:** By default, this required parameter is blank. Within the parameter enter a unique way (key) to identify this rule. Uniqueness is the only restriction for this key. It will not be displayed to the user. It is case sensitive.

Important: Common setup is to copy the key so it can be pasted in the Web UI designer when configuring the business action in Web UI.

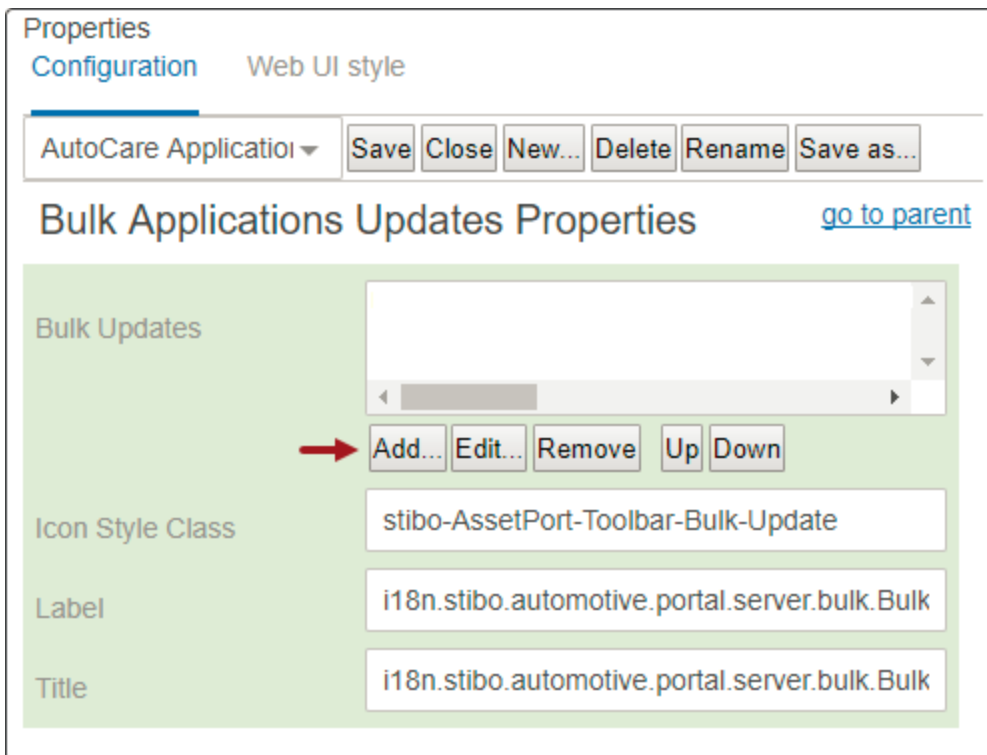
5. Click the **Save** button and continue to the next topic, **Configuring the Copy Application to Other Part Business Action in Web UI**.

Configuring the Copy Application to Other Part Business Action in Web UI

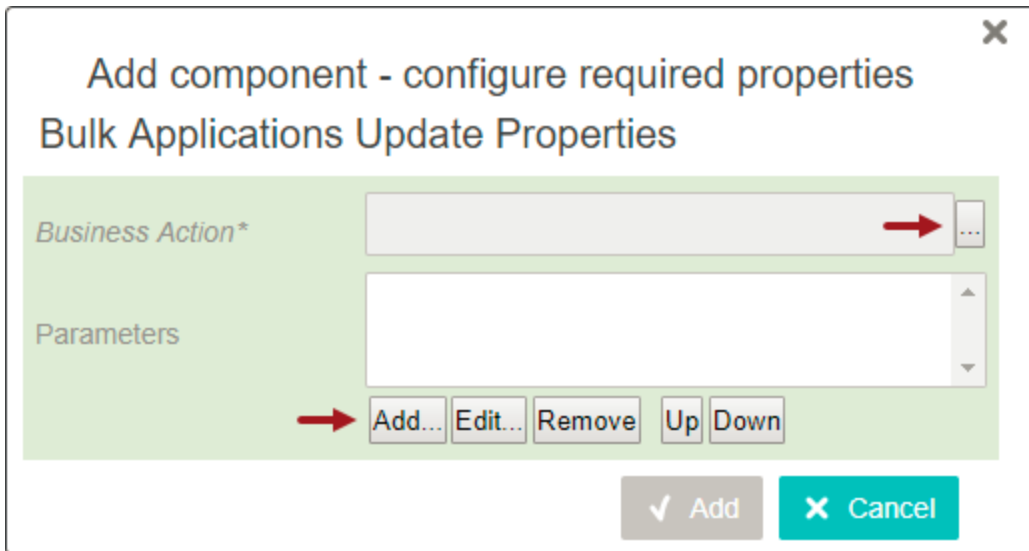
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

Note: After the initial setup, the access to the dialogs does not change, therefore these steps can be helpful when editing the configuration. However, when editing the 'Add' button will be replaced with the 'Save' button, and the 'Add component...' labels at the top of the dialogs will display as 'Edit component... '.

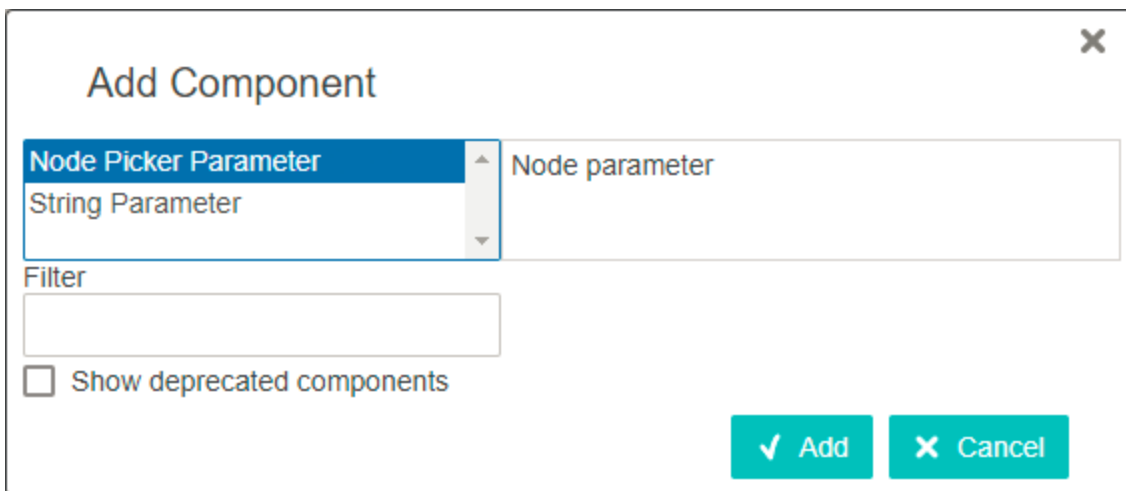
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



4. Click the ellipsis button (...) next to the Business Action parameter > select the business action previously created > click the **OK** button to close the dialog, and return to the Bulk Applications Update Properties dialog. The selected action is displayed in the Business Action parameter.
5. Click the **Add** button beneath the Parameters field, and the Add Component dialog will display (as shown below).



6. Select **Node Picker Parameter** > click the **Add** button to close the dialog, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display (as shown below).

Add component - configure required properties

Node Picker Parameter Properties

Key*

Label

Mandatory

Node Picker Configuration

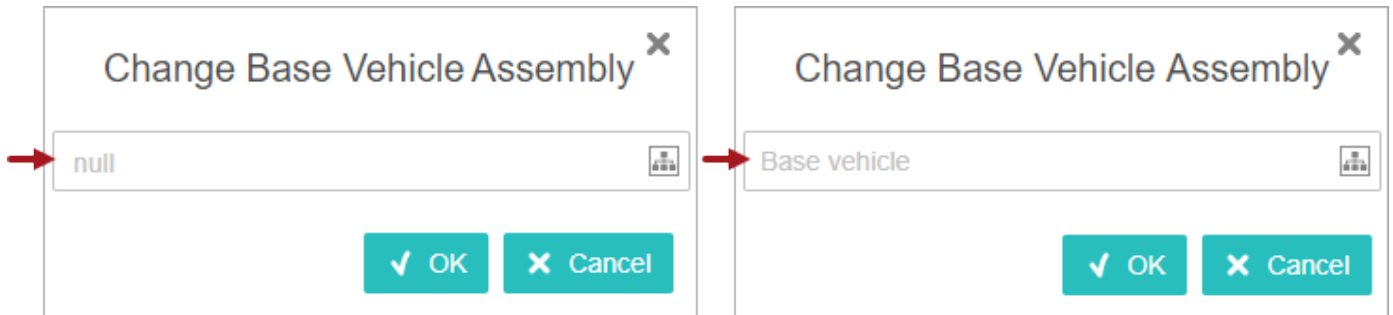
Valid Node Types

Valid Object Types

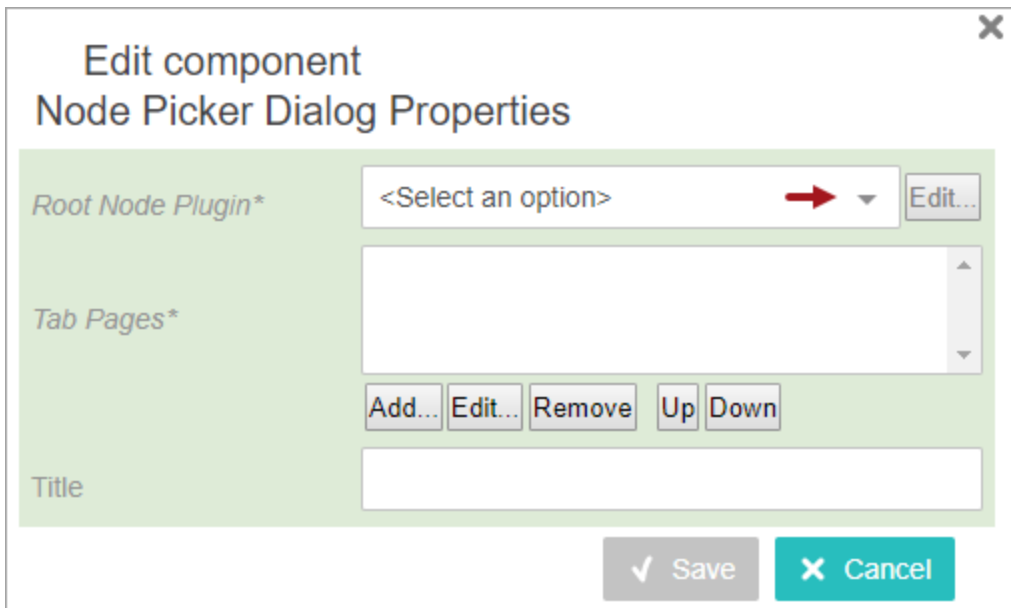
7. Within the Key parameter, enter the exact key created within the 'Copy application to other part parameter key' parameter during step 5 of the **Configuring the Copy Application to Other Part Business Action in Workbench** topic. This is the only required parameter.

Important: If this key is not entered exactly in both places, then the business action will not properly function within the Web UI.

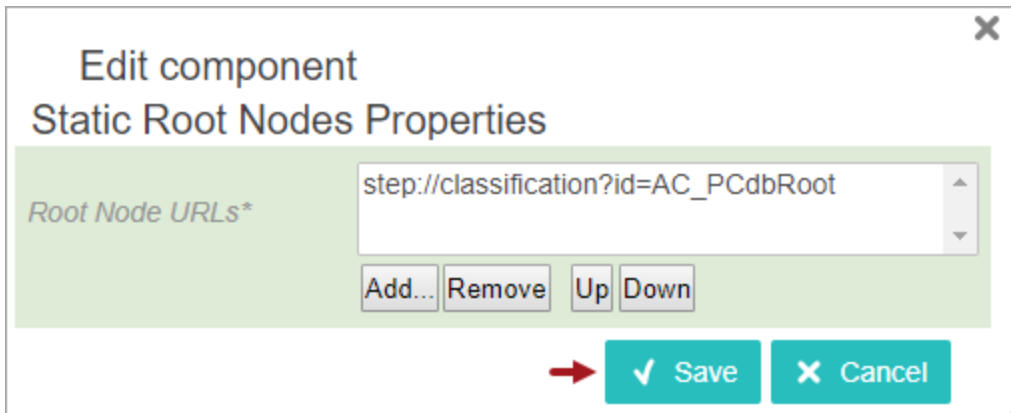
8. Within the Label parameter, enter text that will prompt the user as to what they should select when using this business action. For example, when the Label parameter is blank the parameter within the dialog will display as 'null' as shown in the image on the left. If the Label parameter is populated with 'Base vehicle' the parameter within the dialog will display with 'Base vehicle' as shown within the image on the right.



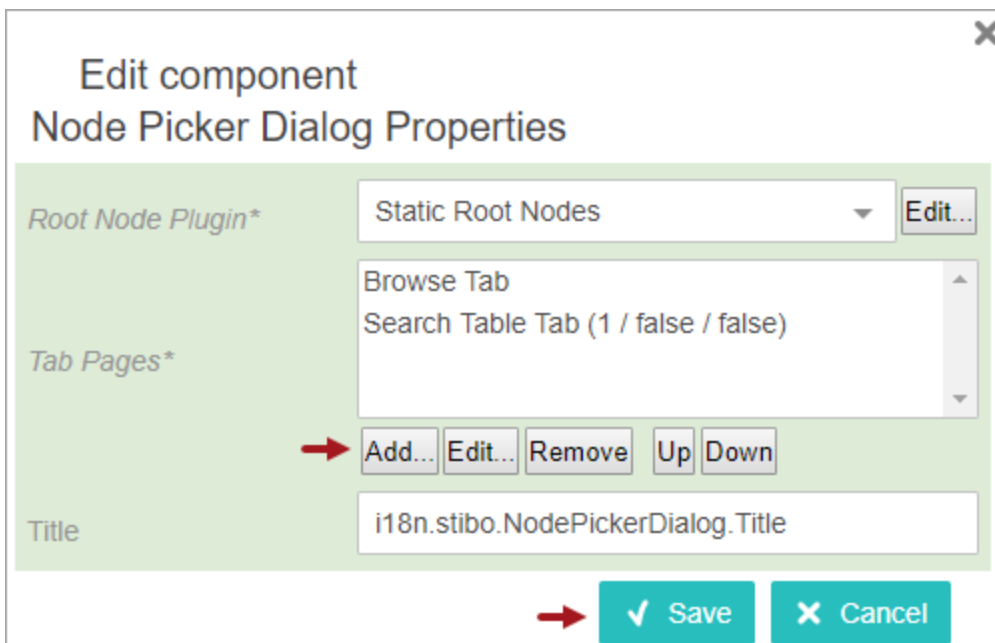
- Use the Node Picker Configuration parameter dropdown to select the **Node Picker Dialog** option, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display.



- Use the Root Node Plugin parameter dropdown to select the **Static Root Nodes** option, and the 'Edit component' for the 'Static Root Nodes Properties' dialog will display.
- Click the **Add** button beneath the Root Nodes URLs parameter > select the Root Nodes that the user should be able to choose from when changing the part (For this example AC_PCdbRoot is used) > click the **OK** button to close the dialog, and return to the 'Edit component' for 'Static Root Nodes Properties' dialog. Optionally repeat this step to add additional nodes for a user to browse from when looking for a part type.



12. Click the **Save** button, and the 'Edit component' for the 'Node Picker Dialog Properties' dialog will display with the Root Node Plugin parameter populated with Static Root Nodes.
13. Click the **Add** button beneath the Tab Pages parameter > select the **Browse Tab** component > click the **Add** button to close the dialog and the 'Edit component' for 'Node Picker Dialog Properties' will display with the Tab Pages parameter populated with Browse Tab. Optionally, repeat this step and add the Search Table Tab.



Note: Leave the Title parameter blank and *after the configuration is saved* an i18n key will be populated (as shown above). For more information, see the **Localization** topic within the **Administration Portal** section of the **STEP Online Help**. Otherwise, text entered within the Title parameter will display in place of the default 'Select Node(s)' title for the dialog used to select the vehicle assembly.

14. Click the **Save** button, and the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog will display with the Node Picker Configuration parameter populated.

15. Use the dropdown located beneath the Valid Node Types parameter to select **PRODUCT_TYPE** > click the **Add** button so that PRODUCT_TYPE is displayed within the Valid Node Types parameter (as shown below).
16. Click the **Add** button beneath the Valid Object Types parameter > select the desired valid object types (For this example AC_PIESItem is used)> click the **OK** button to return to the 'Add component - configure required properties' for the 'Node Picker Parameter Properties' dialog.

Add component - configure required properties

Node Picker Parameter Properties

Key*

Label

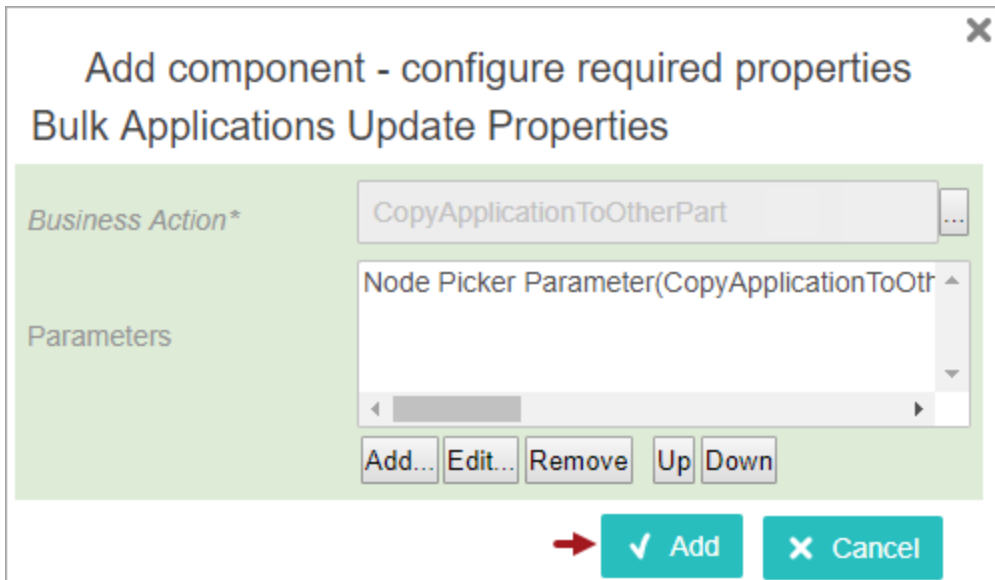
Mandatory

Node Picker Configuration

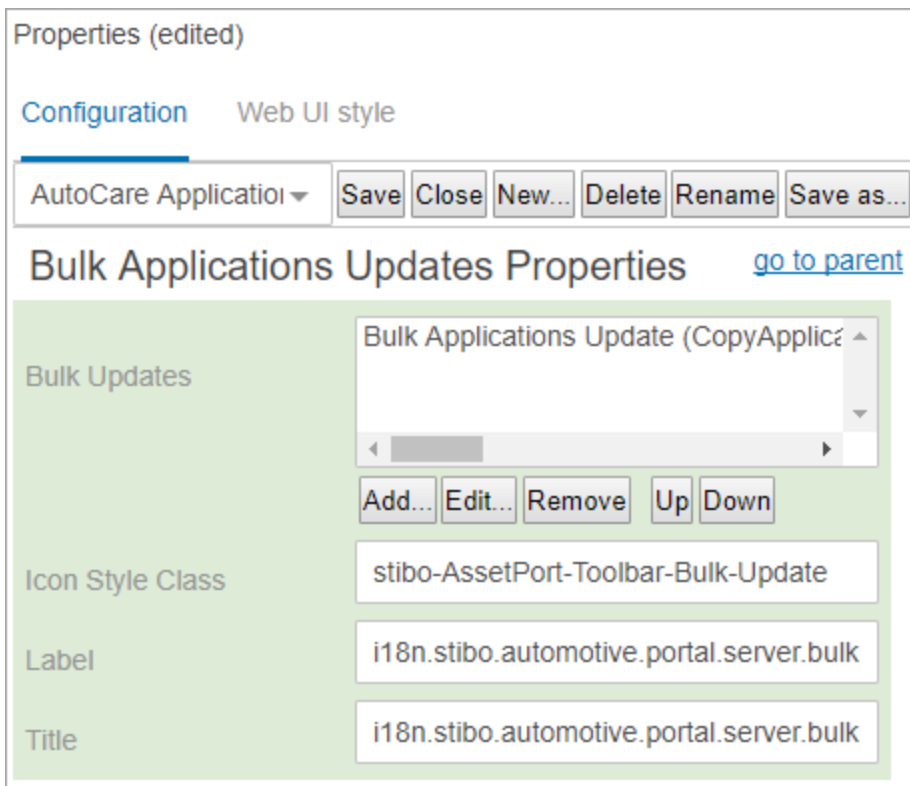
Valid Node Types

Valid Object Types

17. Click the **Add** button, and 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



- Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



- Click the **Save** button and then click the **Close** button to close the designer.

To use the newly configured business action, see the **Using the Configured Copy Application to Other Part Business Action in Web UI** section of the **Business Action: Copy Application to Other Part** topic.

Business Action: Copy Applications to Related Parts

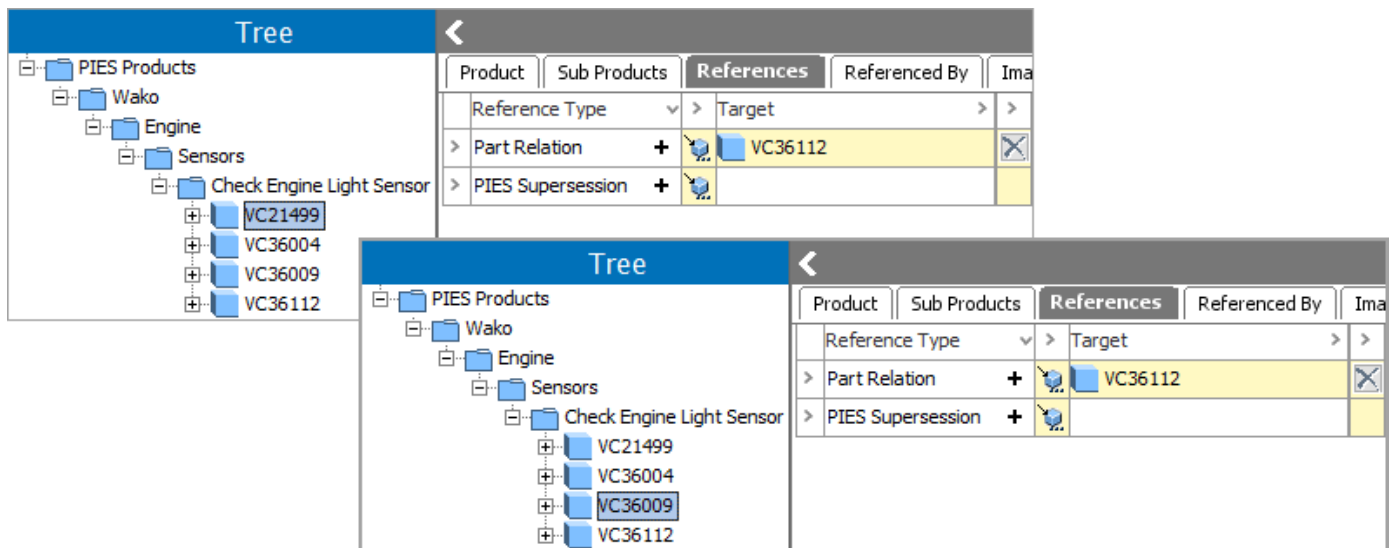
This automotive business action allows users to copy one or more existing applications to one or more related parts by clicking a Bulk Updates action button within an Application Manager. However, it cannot copy the part of a missing application, and thus any missing applications selected when the bulk update is run will be ignored. For more information, see the **Missing Application Coverage Functionality** topic within this guide. Setup is required within both STEP Workbench and Web UI for the action to be available to users.

Once configured, a Bulk Updates button within an Application Manager can be used to copy one or more existing applications to one or more related parts based upon the population of the Part Relation reference type. To understand the results of the business action, it is important to understand a Part Relation reference type configuration. This section will first review a Part Relation reference type configuration, and then provide the steps for using the business action in Web UI.

Part Relation Reference Type Configuration

When the Copy applications to related parts business action is applied within a Web UI, the selected application will be copied to the part(s) that have the selected application's part listed within the reference type 'Part Relation.' In other words, the business action will create applications for any parts that have a relationship with the part used within the selected application.

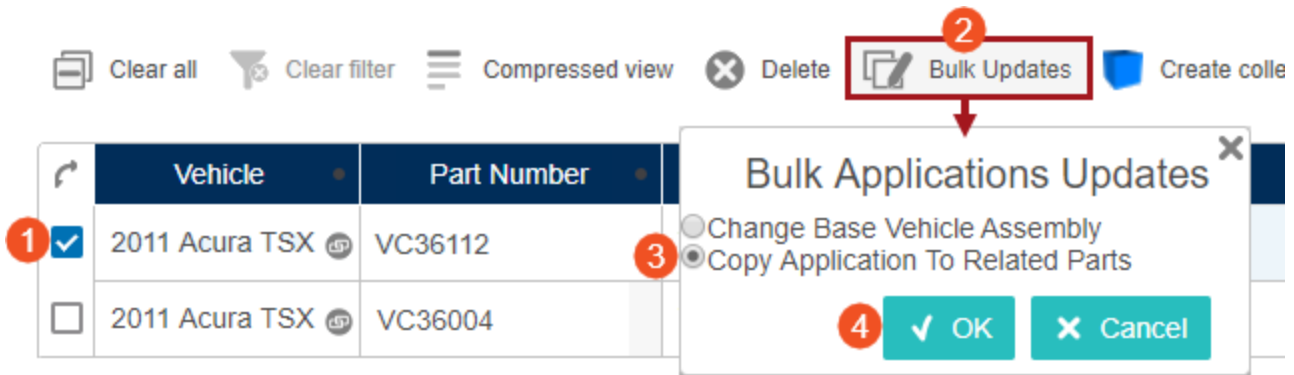
For example, when using the References tab in workbench to view part VC21499 and VC36009, the Part Relation Reference Type can be viewed as 'VC36112.'



With the above configuration in place, the Application Manager can be used to select an application using part VC36112, and when the Bulk Updates action button is clicked, applications will be created for the parts VC21499 and VC36009 (as shown below).

Steps for Using the Business Action in Web UI

When the Parts Relation Reference Type is populated as shown above, the steps below can be used to copy an application to related parts using the Web UI.



Note: If only one bulk update is configured, then the Bulk Applications Updates dialog will not display.

1. Within the configured Application Manager, search for the applications to be copied. Select one or more applications with a part that uses the Part Relation reference type. For this example, part number VC36112 is used.
2. Click the **Bulk Updates** action button. If more than one bulk update is configured then the Bulk Updates dialog will display (as shown above), otherwise this dialog is skipped and the copy applications to related parts bulk update background process will run (skip to step 5 below).
3. Select the Copy Application To Related Parts radio button from the list displayed within the Bulk Applications Updates dialog. For this example, 'Copy Application to Related Parts' is used, but the business action name displayed within the list is dependent upon the business action's Name parameter.
4. Click the **OK** button and the copy applications to related parts business action will run for the selected applications.
5. Once the background process has completed, click the **Find applications** button to display the newly created application(s) within the Application Manager results table. For this example, notice that two new applications for the related parts have been created (as shown below).

	Vehicle	Part Number
<input type="checkbox"/>	2011 Acura TSX	VC36112
<input type="checkbox"/>	2011 Acura TSX	VC36004
<input type="checkbox"/>	2011 Acura TSX	VC21499
<input type="checkbox"/>	2011 Acura TSX	VC36009

Setup is required within both STEP Workbench and Web UI for the action to be available to users. For more information, see the configuration topics below:

- Configuring the Copy Applications to Related Parts Business Action in Workbench
- Configuring the Copy Applications to Related Parts Business Action in Web UI

Configuring the Copy Applications to Related Parts Business Action in Workbench

The 'Copy applications to related parts' business operation is found within the STEP Workbench Business Rule Editor under the Automotive menu and does not offer additional parameters, but does require the use of the reference type 'Part Relation.' Setup is required within both STEP Workbench and Web UI for the action to be available to users. This section addresses configuring the reference type and business action within the workbench.

Configuring the Reference Type

Before the business action can be used within Web UI, the reference type 'Part Relation' must be populated with one or more parts that relate to the selected part. If the business rule is run against an application using a part that has a blank Part Relation parameter, then no changes will occur. In the example below, parts VC21499 and VC36009 have part VC 36112 populated within the Part Relation reference type.

The image displays two screenshots of the Stibo Systems workbench interface, illustrating the configuration of the 'Part Relation' reference type. Both screenshots show a tree view on the left and a table on the right.

Top Screenshot: The tree view shows the hierarchy: PIES Products > Wako > Engine > Sensors > Check Engine Light Sensor. The part VC21499 is selected. The table on the right has the following data:

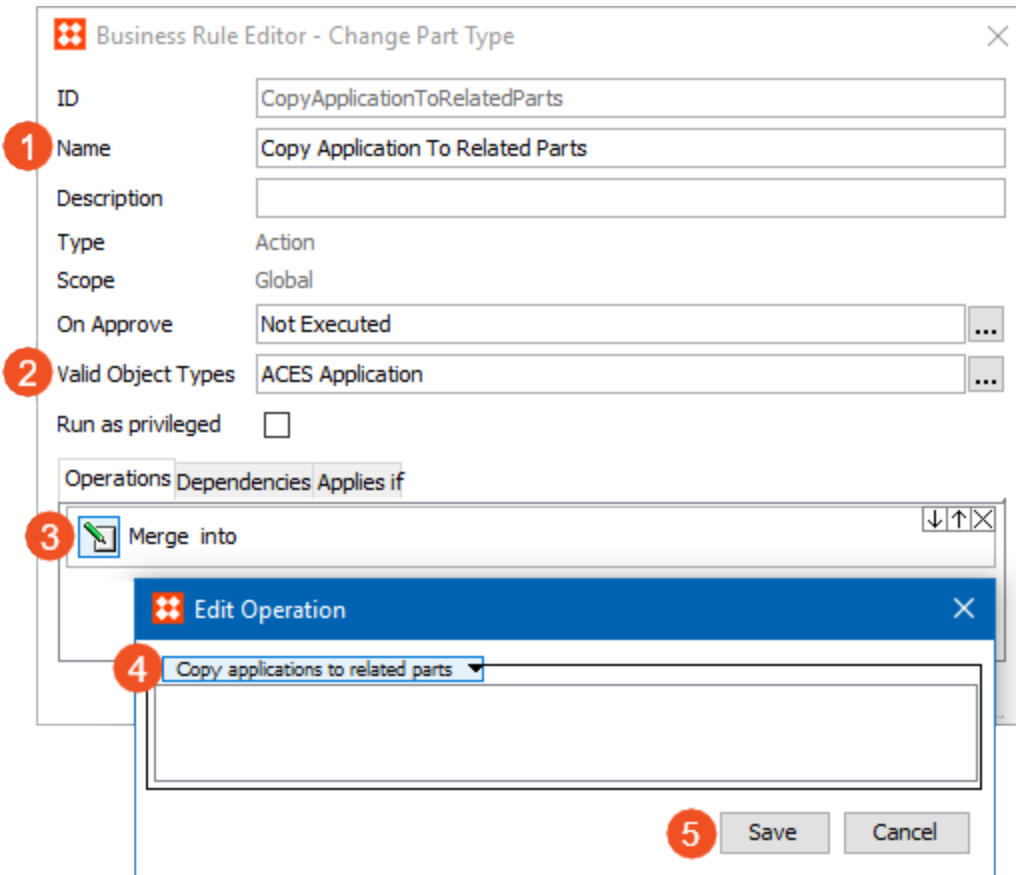
Product	Sub Products	References	Referenced By	Ima
Reference Type	>	Target	>	>
> Part Relation	+	VC36112		
> PIES Supersession	+			

Bottom Screenshot: The tree view shows the same hierarchy, but VC36009 is selected. The table on the right has the following data:

Product	Sub Products	References	Referenced By	Ima
Reference Type	>	Target	>	>
> Part Relation	+	VC36112		
> PIES Supersession	+			

Configuring the Business Action

Before the business action can be used within Web UI, the business action must be created. Below are the steps required to create the business action.



1. Create the business action with a name that accurately describes to the user what this action will do. The name of the business action displays within the Web UI and should be easy for the user to identify. For this example, the business action name is 'Copy Applications To Related Parts.'

Note: The business action Name will display to the Application Manager user once the Bulk Updates action button is selected. If more than one bulk update is configured, then the name displays both within the Bulk Applications Update dialog, and within the copy applications to related parts dialog where the user enters the part for the application (as shown in the example within the **Using the Configured Copy Applications To Related Parts Business Action in Web UI** section of the **Business Action: Copy Applications To Related Parts** topic within this guide).

2. Edit the new business rule, click the ellipsis button (...) next to the Valid Object Types parameter and select the valid object types for this business action. For this example, the 'ACES Application' object type is selected, however the TecDoc and NAPA applications can also be selected within the same or separate business actions. This decision is at the discretion of the administrator.
3. On the Operations tab of the Business Rule Editor, click the **Add new Business Action** link, and click the edit button to open the Edit Operation dialog.
4. Use the dropdown menu within the Edit Operation dialog to select **Automotive > Copy applications to related parts** operation, and the Save button will become active (as shown above).

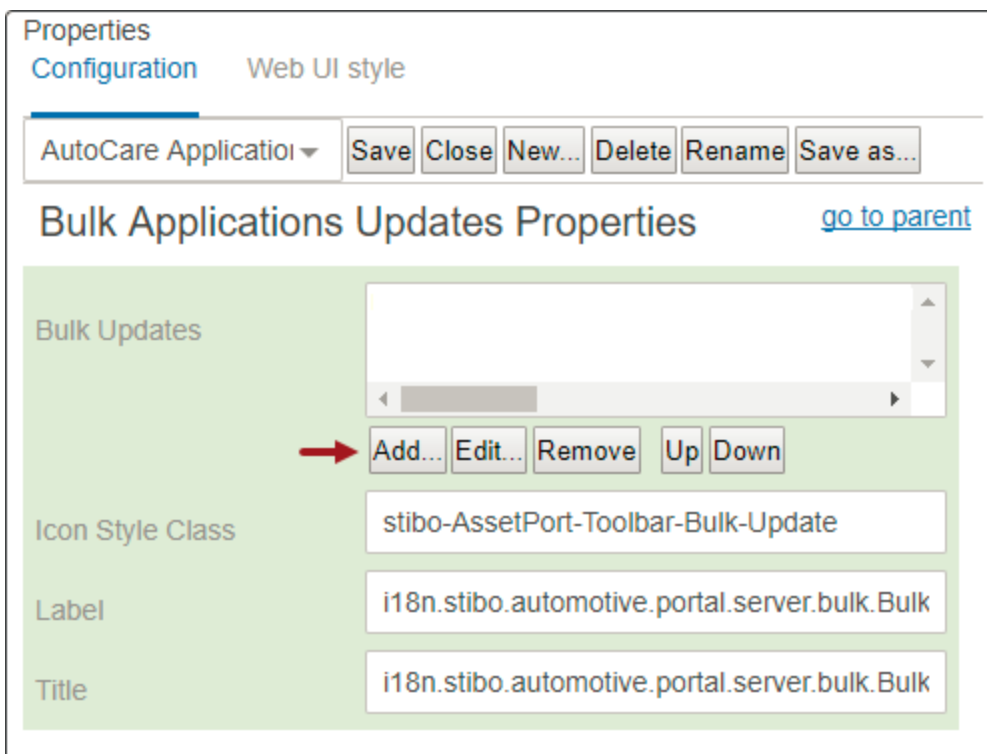
5. Click the **Save** button and continue to the next topic, **Configuring the Copy Applications To Related Parts Business Action in Web UI**.

Configuring the Copy Applications to Related Parts Business Action in Web UI

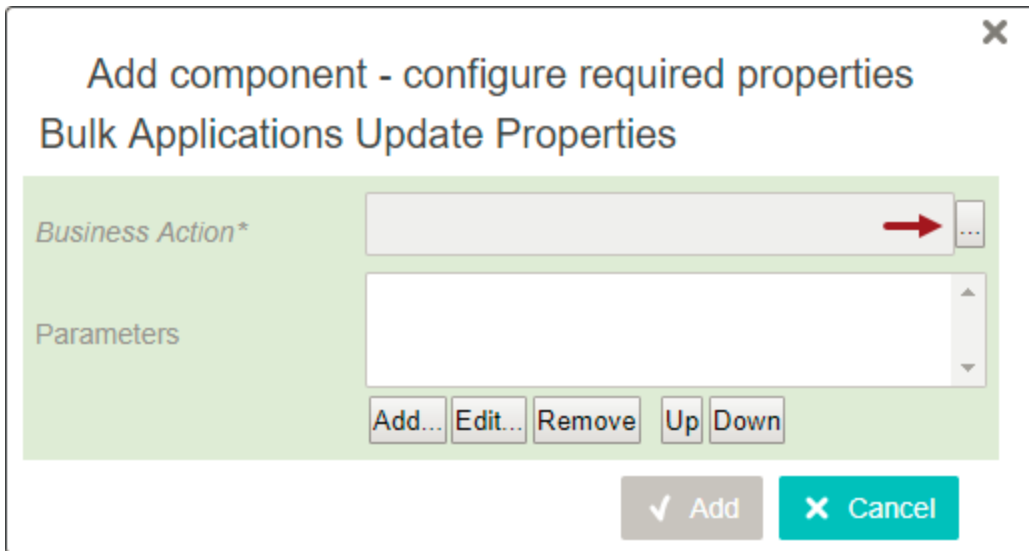
Once the business action has been configured in the workbench, configuration within Web UI is necessary. This section addresses the steps necessary within Web UI.

Note: After the initial setup, the access to the dialogs does not change, therefore these steps can be helpful when editing the configuration. However, when editing the 'Add' button will be replaced with the 'Save' button, and the 'Add component..' labels at the top of the dialogs will display as 'Edit component... '.

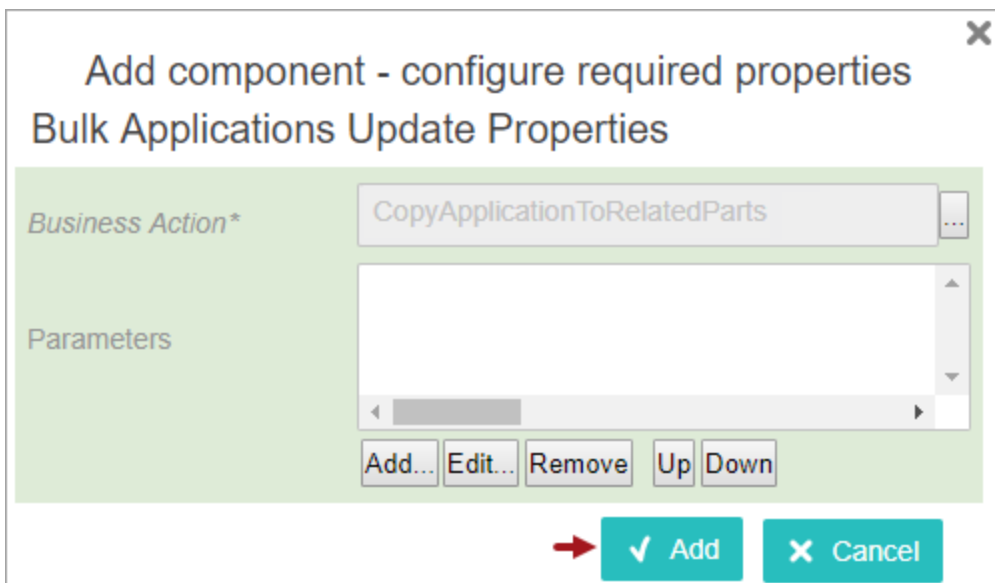
1. Access the Application Manager screen where the business action needs to be available to users.
2. Access the Designer > navigate to Node List Properties > Child Components > Actions > Double click **Bulk Applications Updates** and the 'Bulk Applications Updates Properties' dialog will display (as shown below).



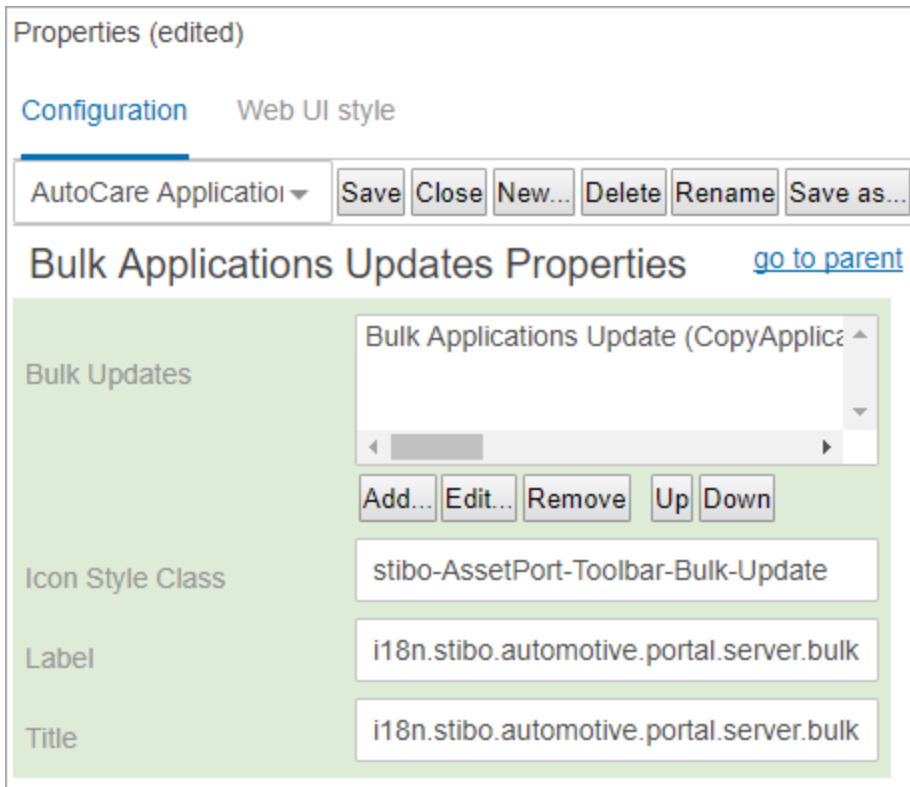
3. Click the **Add** button beneath the Bulk Updates parameter, and the 'Add component - configure required properties' for the 'Bulk Applications Update Properties' dialog will display (as shown below).



4. Click the ellipsis button (...) next to the Business Action parameter > select the business action previously created > click the **OK** button to close the dialog, and return to the Bulk Applications Update Properties dialog. The selected action is displayed in the Business Action parameter.



5. Click the **Add** button, and the 'Bulk Applications Updates Properties' dialog will display with the newly added Bulk Update listed.



6. Click the **Save** button and then click the **Close** button to close the designer.

To use the newly configured business action, see the **Using the Configured Copy Applications to Related Parts Business Action in Web UI** section of the **Business Action: Copy Applications to Related Parts** topic.

Business Action: Move ACES Applications for PIES Part

This rule is a business action found under the Automotive menu that will move all ACES applications from one PIES Item to another by following a reference between the parts. The rule has a single parameter (Reference Type), which requires selection of a reference. For the rule to function properly, the selected reference must be a product reference with the PIES Item object type (ID=AC_PIESItem) as both the source and the target, and the rule must be set with the PIES Item as the valid object type.

The action evaluates all ACES applications that are child to the selected part on which the rule is being executed and moves the child applications from the current object to the target(s) of the reference identified in the rule. If the PIES Item has no reference of this type, no action is taken and the existing applications will remain in place. If the PIES Item has one or multiple targets of the reference, the applications will be duplicated to all targets and removed from the source object.

When a move occurs, the application is moved with all data on the original record retained, except the part number. The part that the application is being moved to is applied as the part number in both the application name and the hash function of the application ID, as well as within the ACES Replacement Context metadata value. The brand element of the replacement context is also updated if the part that the application is being moved to is from another brand (indicated by the value of the attribute with ID=AC_PIES_ITEMBrandAAIAID on the PIES Item).

Important: ACES supports only a single asset being linked to any application. If multiple assets are present on an application, all will be linked to the moved record, but only one will be accounted for in the application ID. This could lead to duplicate applications in the system so it is important to ensure that the standard is adhered to and only one asset is present for any application.

Note that this rule cannot be used to move applications outside of the AutoCare model (using PIES Item and ACES Application object types) as the functionality for construction of the application ID and other related data relies on the standard model.

Business Action: Set Condition Links on Part Types

This rule is a business action found under the Automotive menu that serves to link application conditions / options to part types to assist in configuring display options in the Web UI Application Record Editor (Application Manager Screen). Additional information on the display settings can be found in the Application Manager section of this guide (specifically, the section titled Controlling Display of Conditions in the Editor) so the details of this functionality are not described.

The rule requires selection of one or more application records and if it should be applied en masse, it may be useful to use search functionality and apply the rule as a bulk update using the Run Business Rule operation in the Bulk Update wizard. When run, the rule will evaluate all selected applications and identify which conditions are populated on the applications, per part type. It will then link the attributes or references representing the conditions to the part types, using standard attribute links or the Reference Part Type Links metadata attribute as appropriate.

The business action has two parameters:

- **Attribute groups:** Select one or more attribute groups that hold the condition attributes. All attributes and references in the selected group will be evaluated and any that are populated on the selected application records will be linked to the part type of the application on which it was populated. Attributes and references that are not part of the selected group(s) will remain unaltered.
- **Display condition:** Check this box if the display condition should be set to true for the link, meaning that the attribute or reference will display in its own column within the editor using the 'Application Condition Header - Individual' column. If unchecked, conditions will be linked without the 'true' display condition, meaning that they will show up within the 'Application Condition Header - Group' consolidated column in the editor.

If attribute or reference links already exist on any part type for the populated conditions, the existing links are not altered by running the rule - only new links are added.

Business Action: Set Import Status Attributes

While not strictly required, it may be helpful to implement change flags; a way for users to view what data has been created or changed due to an import.

In order to enable the system to indicate when data is new and/or changed, metadata attributes and the 'Set import status attributes' business action can be used within the Import state of a workflow. At a minimum, at least one attribute is required for delete statuses, new statuses, and changed statuses for all imports. If additional distinction is desired, unique attributes can be created to store delete, new, and changed statuses for varying objects or imports. In either case, when an importer determines that data should be flagged as delete, new, or changed, it writes 'true' in the attributes indicated in the business action configuration. This allows each customer to determine their own strategy for managing delete, new, and/or changed data, such as processing the new and/or changed data via a workflow.

In the example below, an ACES import created a new application for part VC21499. Because the application is new, the configured attribute New Object displays the value of 'true.'

Name	Value
ID	AC_ACESApp_39792bde003b9d01d4a3171aa8241
Name	VC21499
Object Type	ACES Application
Revision	0.4 Last edited by USERN on Fri Feb 02 15:10:14 E
Approved	✘ Never Been Approved
Translation	Not Translated
Path	Primary Product Hierarchy/AutoCare Root/PIES Pro
ACES Replacement Context	abc
Changed Object	abc
Delete Status	abc
New Object	abc true ←

If the application had previously existed and had information changed during the import, then only the Changed Object attribute value would display 'true.'

Name	Value
ID	AC_ACESApp_39792bde003b9d01d4a3171aa8241
Name	VC21499
Object Type	ACES Application
Revision	0.4 Last edited by USERN on Fri Feb 02 15:12:17 ES
Approved	✘ Never Been Approved
Translation	Not Translated
Path	Primary Product Hierarchy/AutoCare Root/PIES Proc
ACES Replacement Context	abc
Changed Object	abc true ←
Delete Status	abc
New Object	abc

Important: This business action can ONLY be used within the Import state of a workflow. Additionally, an error will occur when delete flags are used for TecDoc reference data on an 8.2 system. It is recommended that change flags for TecDoc reference data only be implemented on 8.3 (or newer) releases.

Configuring Change Flags for New and/or Updated Objects

To properly configure this business action to track new and/or updated objects, the following is needed:

- Create / Identify Valid Attributes
- Configure a Business Action to Use the Set Import Status Attributes Operation
- Update Workflow Settings

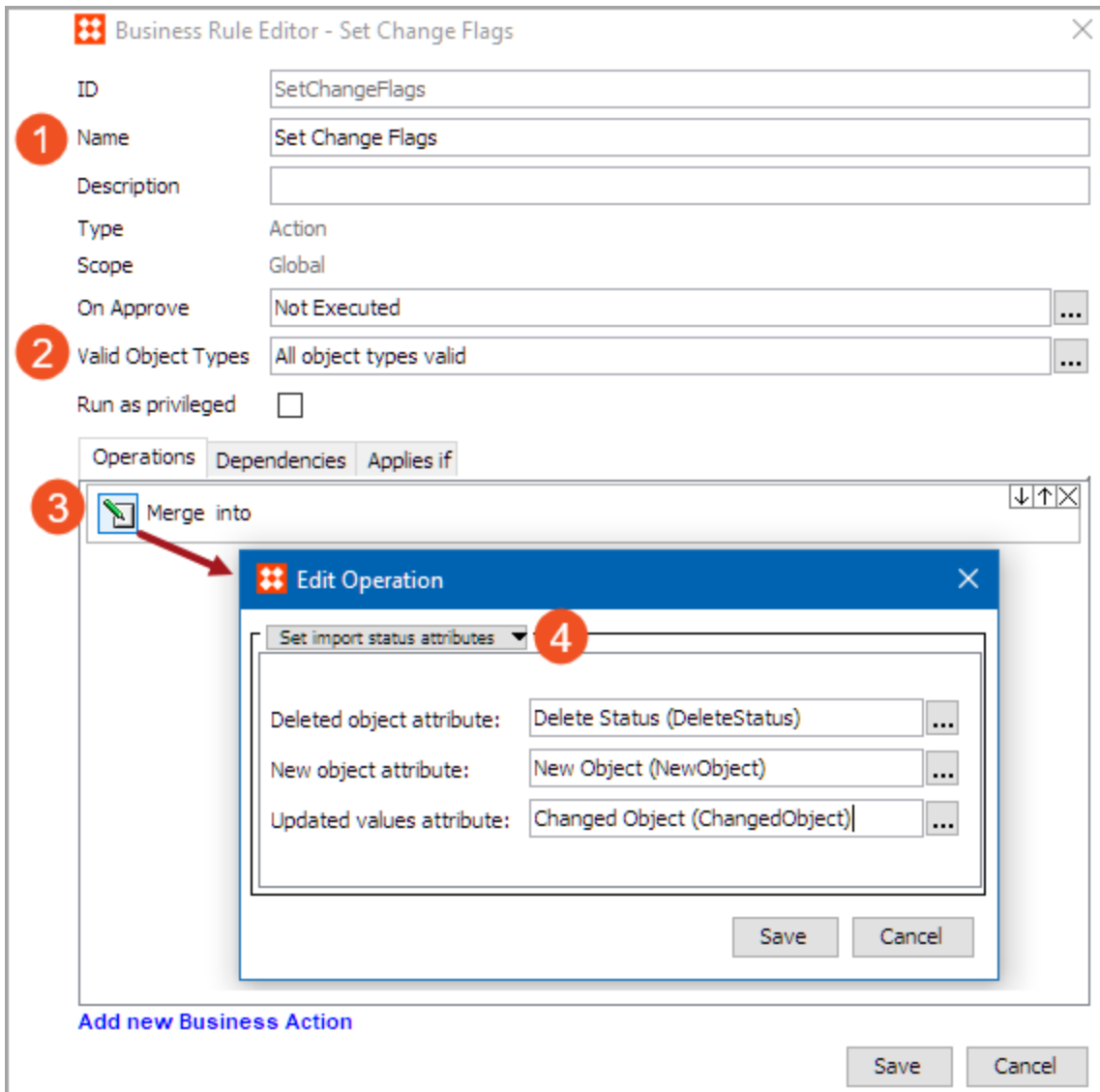
Create / Identify Valid Attributes

1. Create the necessary attributes (or verify that they exist) to be used to store the new and/or changed data (i.e. Delete Status, New Object, Changed Object).
2. Make sure the attributes are valid for the object(s) being managed by the import.

For example, an attribute used in the AutoCare ACES Application Importer must be valid on ACES Application objects. Alternatively, an attribute used in the AutoCare PCdb Importer must be valid on all nodes in the PCdb classification hierarchy.

Configure a Business Action to Use the Set Import Status Attributes Operation

1. Create a business action and give it a name that users can easily identify. In the example below, Set Change Flags is used.
2. Click the ellipsis button (...) to the right of the Valid Object Types parameter to find and select the necessary object type(s). In the Set Change Flags example below, 'All object types valid' is selected.
3. On the Operations tab, click the 'Merge into' button, and the Edit Operation dialog will display.
4. Click on the Edit Operation dropdown, click **Import flow**, and then click **Set import status attributes**.



5. Three parameters will display. Configure the parameters to use the necessary attributes.

- **Deleted object attribute:** Click the ellipsis button (...) to find and select the attribute which is configured in the Delta Calculation state within the 'Delete status attribute' parameter. This parameter clears the delete flag status (if any) upon successful import of the same object again. The attribute that is used in this parameter should be the same attribute that is configured in the Delta Calculation state. For the example above, the Delete Status attribute (created / identified in the previous section) is selected.
- **New object attribute:** Click the ellipsis button (...) to find and select an attribute to be used to identify when an object is added. For the example above, the New Object attribute (created / identified in the previous section) is selected.

- **Updated values attribute:** Click the ellipsis button (...) to find and select an attribute to be used to identify when an object is changed. For the example above, the Changed Object (created / identified in the previous section) is selected.

Note: PMDM for Automotive solution, upon configuration, only provides an option to automatically erase the existing flags in the attribute marked for deletion. Wherein the flags (attribute value) marked for notifying newly added objects and changed objects cannot be erased automatically. User needs to determine their own strategy for erasing the flags (attribute value). For information on erasing the existing delete status flags, see topic **Use Cases - Clearing Delete Status Flag**.

6. Click the **Save** buttons to save and close the business rule.

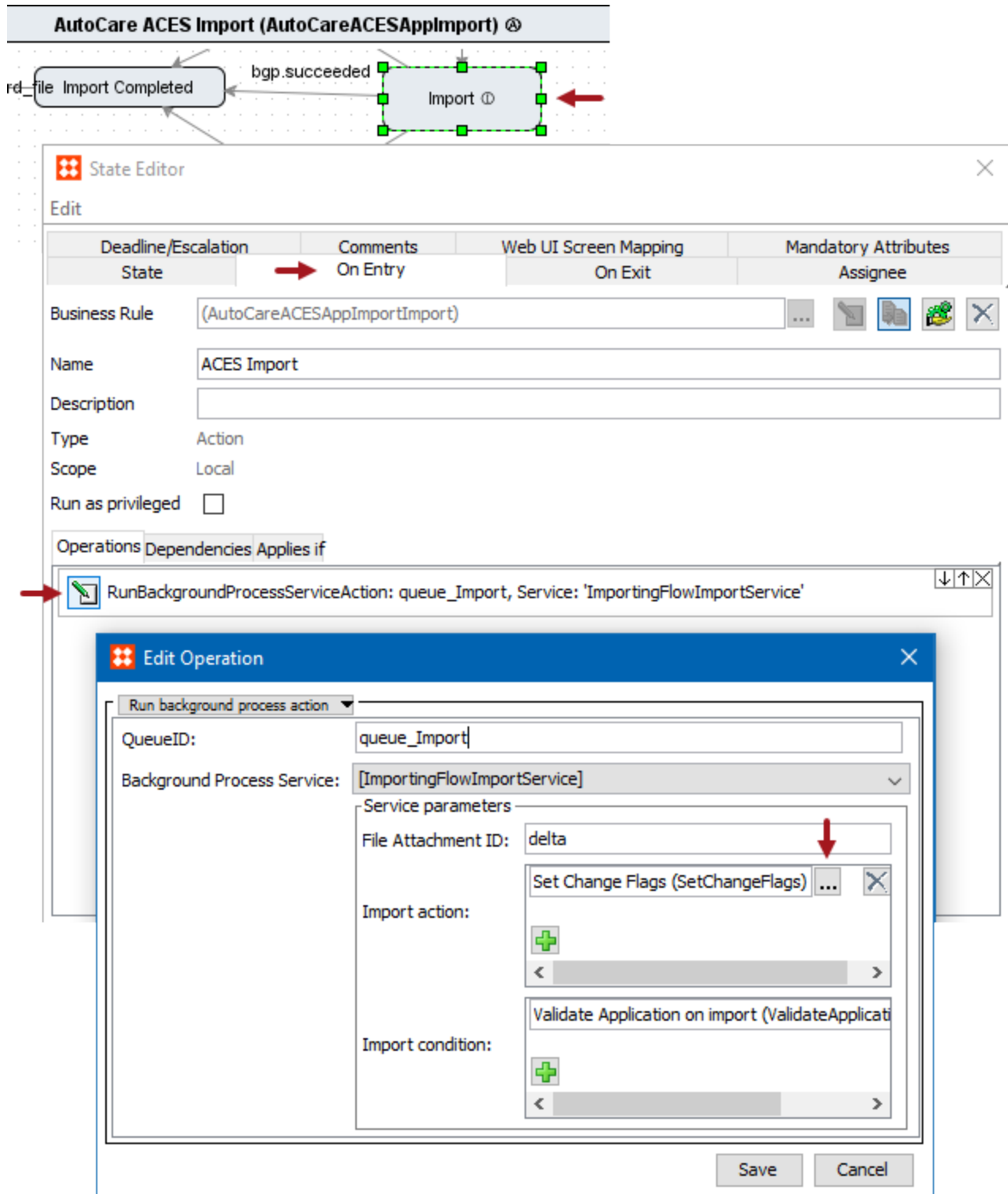
For more information on creating and/or editing business rules, see the **Creating a Business Rule or Library** topic, and the **Editing a Business Rule** topic within the **Business Rules** guide.

Update Workflow Settings

Once the business rule is configured, the necessary workflows need to be updated to use the business rule. Below are the steps to update a workflow to use the business rule created in the previous section.

Important: Because the business rule created in the previous section is configured to work with ACES Application valid object types, the ACES Import Workflow is used for the example below. If the business rule is configured to also work with TecDoc and/or NAPA Applications, then those import workflows would need to also be edited.

1. Go to workbench > System Setup > Workflows > select a workflow.
2. Right-click, and select **Edit STEP Workflow**.
3. On the STEP Workflow Designer window, double click on the **Import** state to open the State Editor.
4. Select the **On Entry** tab, and within the Operations tab, click the Edit icon to display the Edit Operations Dialog (as shown below).



5. In the Edit Operation dialog, Click the ellipsis button (...) to the right of the Import action parameter to find and select the business action previously created.
6. Click the **Save** button, and exit the workflow.

Use Case Appendix

The following use cases address various circumstances where the Business Action operation 'Set import status attributes' can be applied when using the PMDM for Automotive solution.

- Use Cases - Tracking Newly Created Objects
- Use Cases - Tracking Deleted Objects
- Use Cases - Tracking Changed Objects
- Use Cases - Clearing Delete Status Flag

It is important to note that the delete status flag (an attribute defined to indicate the delete status) is configured in the Delta Calculation state, so the user should have detailed knowledge about the Delta Calculation state. For more information, see topics **5. Update Delete Status Attribute and Delta Calculation Method in Import Workflows** and **Delta Calculation State** within the **Automotive Reference Guide**.

Use Cases - Tracking Newly Created Objects

The user is responsible for configuring the PMDM for Automotive system to keep track of all newly created objects on subsequent imports.

Consider an example that a new ACES file is imported into the system and all applications are stored in the system under their respective hierarchy. The next time the same file with an addition of few more applications is imported into the system, the newly created applications are tracked as defined below.

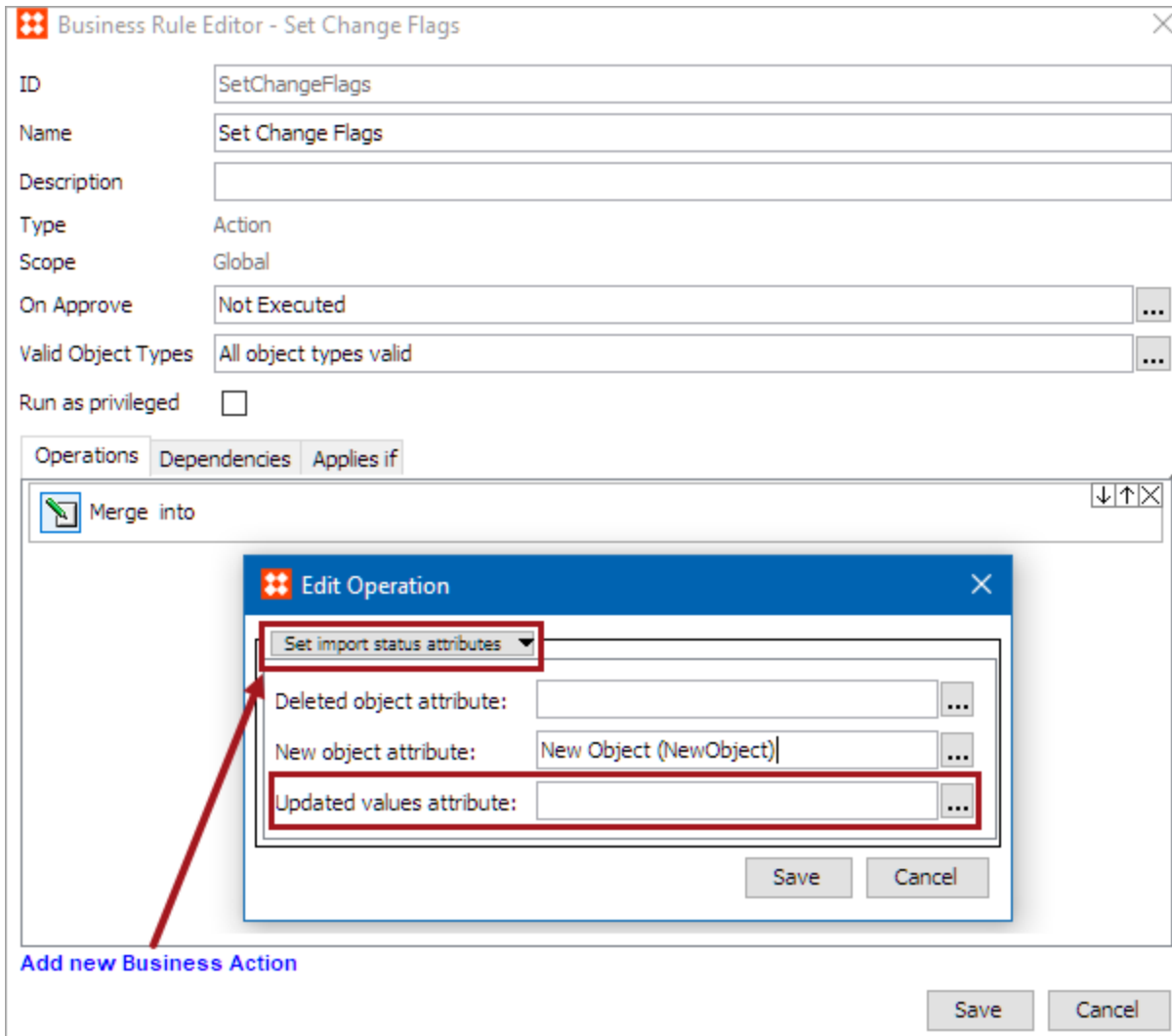
Track new objects created during import

1. Create an attribute (or verify that one exists) with the validity set to the object type(s) that need to be tracked. In this example, an attribute 'New Object' is created with the validity set to object type ACES Application.

The screenshot shows the 'System Setup' interface with a tree view on the left and a configuration panel on the right. The tree view shows 'Attribute Groups' expanded to 'Metadata', where 'New Object' is selected. The configuration panel is titled 'New Object - Attribute' and has tabs for 'Attribute', 'References', 'Attribute Transformation', 'Validity', and 'Profile'. The 'Attribute' tab is active, showing a table with the following data:

Description	
Name	Value
ID	NewObject
Name	New Object
Last edited by	2019-03-08 02:19:16 by USERB
Full Text Indexable	No
Externally Maintained	No
Completeness Score	
Hierarchical Filtering	None
Calculated	No
Type	Description
Dimension Dependencies	
Mandatory	No
ATTR_Article OK	
ATTR_Delete	

2. Configure the business rule to include the newly created attribute within the 'New object attribute' parameter in the 'Set import status attributes' operation. For more information on configuring the attribute in the business action, see the topic **Business Action: Set Import Status Attributes**. In the example below, the 'New Object' attribute is defined in the 'Set import status attributes' operation within the business action 'Set Change Flags.'



3. Configure the **Import** state of the workflow with the updated business action discussed in the previous step. For detailed procedures on updating the workflow settings, see the topic **Business Action: Set Import Status Attributes**. In the example below, the Import state in the AutoCare ACES Import workflow is configured with Set Change Flags business action.

AutoCare ACES Import (AutoCareACESAppImport) @

file Import Completed ← bgp.succeeded → Import @

State Editor

Edit

Deadline/Escalation	Comments	Web UI Screen Mapping	Mandatory Attributes
State	On Entry	On Exit	Assignee

Business Rule: (AutoCareACESAppImportImport)

Name: ACES Import

Description:

Type: Action

Scope: Local

Run as privileged:

Operations Dependencies Applies if

RunBackgroundProcessServiceAction: queue_Import, Service: 'ImportingFlowImportService'

Edit Operation

Run background process action

QueueID: queue_Import

Background Process Service: [ImportingFlowImportService]

Service parameters

File Attachment ID: delta

Import action: Set Change Flags (SetChangeFlags)

Import condition: Validate Application on import (ValidateApplicati)

Save Cancel

When a file is imported, the system evaluates the contents of the import file as compared to the contents of the database, or against the previously loaded file of that type. For data that is observed to be a new object, the importer writes to the defined attribute with a value of 'true.' Hereafter, users shall determine their own strategy for managing the newly added objects.

In the example below, the ACES import created a new application for part VC21499. Because the application is new, the configured attribute New Object displays the value of 'true.'

The screenshot shows a product hierarchy on the left and a data table on the right. The hierarchy is as follows:

- PIES Products
 - BCVC
 - Wako
 - Engine
 - Sensors
 - Check Engine Light Se
 - VC21499
 - PK
 - VC21499 (highlighted)
 - VC21499
 - VC21499
 - VC21499
 - VC21499
 - VC21499
 - VC21499

The data table on the right contains the following information:

Name	Value
ID	AC_ACESApp_39792bde003b9d01d4a3171aa8241
Name	VC21499
Object Type	ACES Application
Revision	0.4 Last edited by USERN on Fri Feb 02 15:10:14 E
Approved	✘ Never Been Approved
Translation	Not Translated
Path	Primary Product Hierarchy/AutoCare Root/PIES Pro
ACES Replacement Context	abc
Changed Object	abc
Delete Status	abc
New Object	abc true ←

Use Cases - Tracking Deleted Objects

The user is responsible for configuring STEP to keep track of all the deleted objects on subsequent imports and handle the deleted objects manually in the system.

Note: While importing, if an object is missing in the import file compared to what is in STEP, the import does not automatically delete the object. However, it will populate the delete flag attribute with a value of 'true' on the object to indicate that the object is missing and/or deleted in the imported file. This allows the user to determine their own strategy for managing deletions, such as processing the deletions via a workflow.

Consider an example that a file containing the Brand details is imported into the system and all Brands including their additional information are stored in the system under their respective hierarchy. The next time the same type of file is imported with some brand name deleted, the deleted Brand names are tracked as defined below.

Track objects deleted during import

1. Create an attribute (or verify that one exists) with the validity set to the object type(s) that need to be tracked. In this example, an attribute 'Delete Status' is created with the validity set to object type Brand.

System Setup		Delete Status - Attribute	
		Attribute	References
Attribute Groups		Attribute Transformation	
Application Record Attributes		Validity	
Asset Export Configuration		Profile	
AutoCare Attributes		Description	
Business Rules		Name	>> Value
Category Specific Attributes		ID	DeleteStatus
Display		Name	Delete Status
Event Filtering		Last edited by	2019-03-08 02:06:00 by USERB
General Attributes		Full Text Indexable	No
Import Flow Attribute Group		Externally Maintained	No
Metadata		Completeness Score	
Attribute Help Text		Hierarchical Filtering	None
Automotive Validation Path		Calculated	No
Changed Object		Type	Description
Delete Status		Dimension Dependencies	
Display Condition		Mandatory	No
Display Sequence			
Missing Application Conditions			

2. Within the Delta Calculation state of the workflow, configure the newly created attribute within the 'Delete status attribute' parameter. For more information on configuring the attribute in the workflow, see the topic **5. Update Delete Status Attribute and Delta Calculation Method in Import Workflows**. In the example below, the attribute Delete Status is configured within the Delta Calculation state in the AutoCare Brand Import workflow. Setting the Delete Status attribute in this state is what writes the value to the Delete flag.

Edit Operation

Run background process action ▼

QueueID: queue_DeltaCalculate

Background Process Service: [StepXMLDeltaCalculationService] ▼

Service parameters

Delta calculation method: file

Clear delete attribute:

Delete status attribute: Delete Status (DeleteStatus) [icon]

From file Attachment ID: convert

Output file Attachment ID: delta

Save Cancel

Each workflow has a Delta calculation method where the importer evaluates the contents of the imported file to the contents in the database ('context' method), or against the previously loaded file of that type ('file' method) to remove unchanged objects (it will take failed imports and errors into account). While doing this it will collect all IDs in the current import file. However, the Delta Calculation to calculate the delete flag (regardless if 'file' or 'context' method is used) will go through the entire hierarchy in the database and see what is present in the database but not in the current file. The objects that are in the database but not in the import file will be added to the delete file and have the configured delete status attribute set to 'true.' For data that is determined to require deletion, the importer does not actually delete the data, but instead writes to the defined attribute that the data *should* be deleted. This allows each customer to determine their own strategy for managing deletions, such as processing the deletions via a workflow.

In the example below, the Brand import has a brand name '3M' deleted. Because the brand name '3M' is deleted in import file, the configured attribute Delete Status displays the value of 'true.'

Tree

- Assets
 - AutoCare Root
 - ACES Assets
 - Brand Table
 - 0
 - 1
 - 2
 - 3
 - 312 Motoring
 - 31 Incorporated
 - 356 Devotion
 - 3D Auto Parts
 - 3D Carbon
 - 3D Mention Media LLC
 - 3Gen LLC
 - 3L Industries Inc
 - 3M
 - 3M
 - 3M
 - ScotchBlue
 - Accuspray
 - Bondo
 - Command

3M rev.0.2 - Classification

Classification
Sub Products
References
Referenced By
Images & Documents
Tables

Description	
Name	Value
> ID	AC_Brand_BBBB
> Name	3M
> Object Type	Brand
> Revision	0.2 Last edited by USERB on Thu Apr 11 09:54:05 EDT 2019
> Approved	Never Been Approved
> Translation	Not Translated
> Path	Classification 1 root/AutoCare Root/Brand Table/3/3M/3M/3M
> Visibility	
> ACES Import Handling FULL	
> ACES Import Handling SUPPL	
> ACES Import Handling UPDAT	
> Brand Revision Date	2007-07-12
> Changed Object	abc
> Delete Status	abc True
> New Object	abc

Use Cases - Tracking Changed Objects

The user is responsible for configuring the STEP to keep track of all changed objects on subsequent imports.

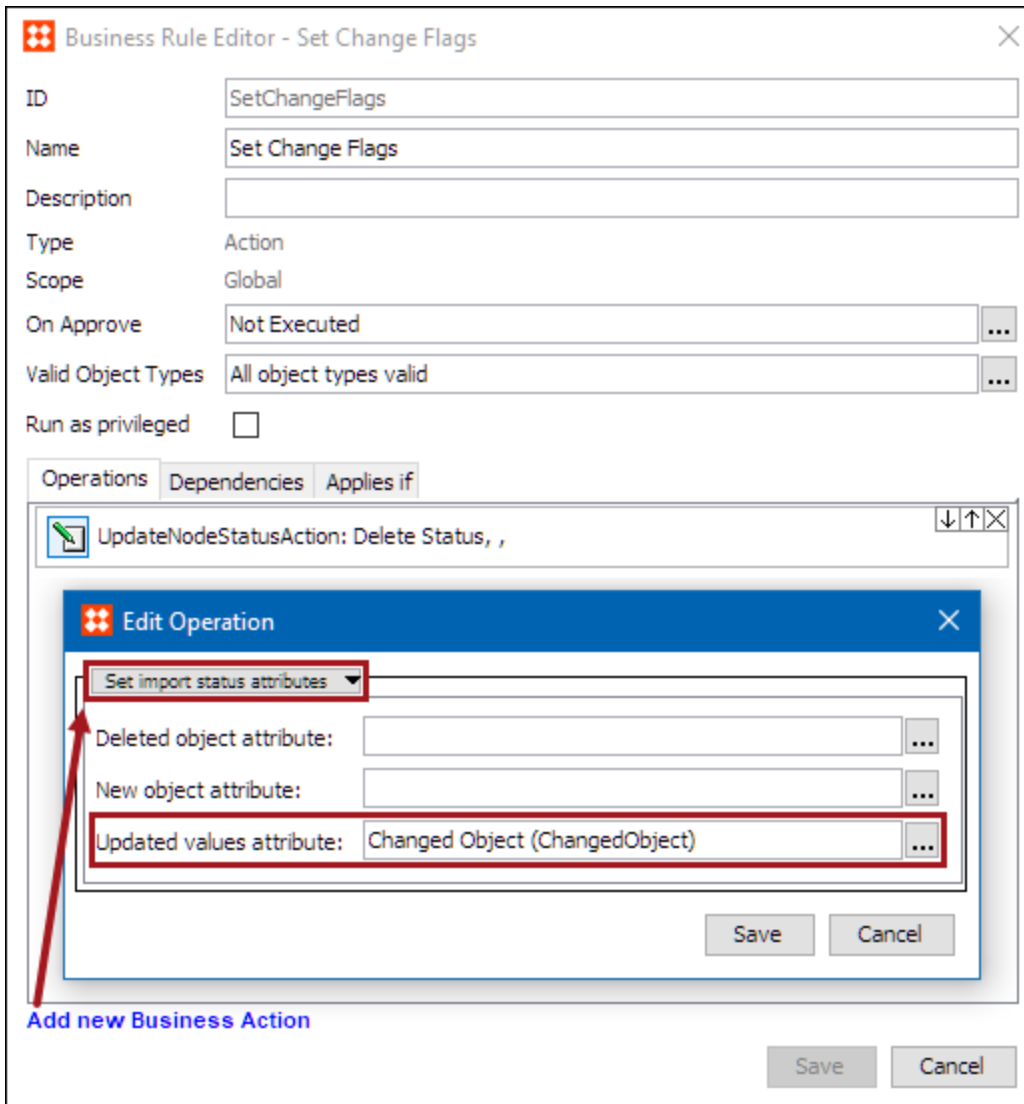
Consider an example that a file containing the Brand details is imported into the system and all Brands including their additional information are stored in the system under their respective hierarchy. The next time the same file with any changes in the existing Brands are imported into the system, the Brands with changes are tracked as defined below.

Track changed objects created during import

1. Create an attribute (or verify that one exists) with the validity set to the object type(s) that need to be tracked. In this example, an attribute 'Changed Object' is created with the validity set to object type Brand.

System Setup		Changed Object - Attribute	
		Attribute	References
Attribute Groups		Attribute Transformation	Validity
Application Record Attributes		Profile	
Asset Export Configuration		Description	
AutoCare Attributes		Name	Value
Business Rules		ID	ChangedObject
Calculated Attributes		Name	Changed Object
Category Specific Attributes		Last edited by	2018-04-12 14:03:07 by USERN
Display		Full Text Indexable	No
Event Filtering		Externally Maintained	No
General Attributes		Completeness Score	
Import Flow Attribute Group		Hierarchical Filtering	None
Metadata		Calculated	No
ApplicationSource		Type	Description
Attribute Help Text		Dimension Dependencies	
Automotive Validation Path		Mandatory	No
Changed Object			
Delete Status			
Display Condition			

2. Configure the business rule to include the newly created attribute within the 'Updated values attribute' parameter in the 'Set import status attributes' operation. For more information on configuring the attribute in the business action, see the topic **Business Action: Set Import Status Attributes**. In the example below, 'Changed Object' attribute is defined in the 'Set import status attributes' operation within the business action 'Set Change Flags.'



3. Configure the **Import** state of the workflow with the updated business action discussed in the previous step. For detailed procedures on updating the workflow settings, see the topic **Business Action: Set Import Status Attributes**. In the example below, the Import state in the AutoCare Brand Import workflow is configured with Set Change Flags business action.

AutoCare ACES Import (AutoCareACESAppImport) Ⓜ

rd-file Import Completed ← bgp.succeeded → Import Ⓜ →

State Editor

Edit

Deadline/Escalation	Comments	Web UI Screen Mapping	Mandatory Attributes
State	On Entry →	On Exit	Assignee

Business Rule: (AutoCareACESAppImportImport)

Name: ACES Import

Description:

Type: Action

Scope: Local

Run as privileged:

Operations Dependencies Applies if

RunBackgroundProcessServiceAction: queue_Import, Service: 'ImportingFlowImportService'

Edit Operation

Run background process action

QueueID: queue_Import

Background Process Service: [ImportingFlowImportService]

Service parameters

File Attachment ID: delta

Import action: Set Change Flags (SetChangeFlags)

Import condition: Validate Application on import (ValidateApplicati)

Save Cancel

When a file is uploaded, the system evaluates the contents of the import file as compared to the contents of the database, or against the previously loaded file of that type. For data that is observed to have information changed during the import, the importer writes to the defined attribute with a value of 'true.' Hereafter, users shall determine their own strategy for managing the changed objects.

In the example below, the brand 3M had previously existed and had information changed during the import. Because the brand 3M is changed, the configured attribute Changed Object displays the value of 'true.'

3M rev.0.3 - Classification	
Classification	Sub Products
Description	
Name	Value
> ID	AC_Brand_BBBB
> Name	3M
> Object Type	Brand
> Revision	0.3 Last edited by BRANDIMPORT on Mon Feb 25 10:56:39 EST 2019
> Approved	✘ Never Been Approved
> Translation	Not Translated
> Path	Classification 1 root/AutoCare Root/Brand Table/3/3M/3M/3M
> Visibility	
> Brand Revision Date	2007-07-12
> Changed Object	abc true ←
> Delete Status	abc
> Import Handling FULL	
> Import Handling SUPPLIER	
> Import Handling UPDATE	
> New Object	abc

Use Cases - Clearing Delete Status Flag

When an object marked for deletion is resurrected in a later import, it is necessary that the previously set delete status flags are erased and the latest status information of an object is displayed in the system.

The PMDM for Automotive solution provides an option to automatically erase the existing flags in the attribute marked for deletion. However, the flags (attribute values) used for identifying newly added objects and changed objects cannot be erased automatically. In this use case, the user is responsible for configuring the PMDM for Automotive such that all the existing delete status flags are erased with the reimport of the deleted objects.

To clear a delete status flag, choose one of the two configuration methods as defined below.

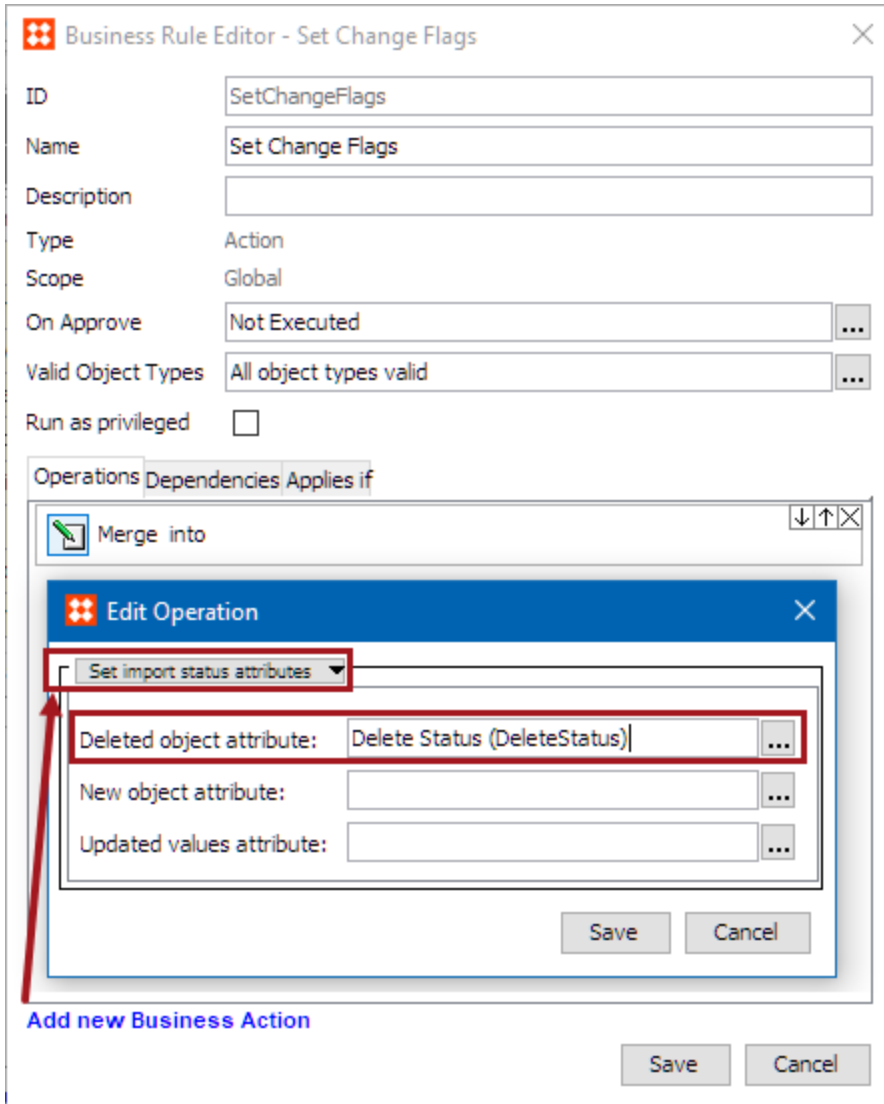
Method 1: Using 'Clear delete attribute' parameter in Delta Calculation state

1. In the workflow, double click on the **Delta Calculation** state to open the State Editor. Select the **On Entry** tab and click the Edit icon on the existing business rule.
2. Enable the 'Clear delete attribute' parameter. For more information on accessing this parameter, see the topic **Delta Calculation State**.

The screenshot shows the 'Edit Operation' dialog box for a background process action. The 'Run background process action' dropdown is set to 'Run background process action'. The 'QueueID' is 'queue_DeltaCalculateClear'. The 'Background Process Service' is '[StepXMLDeltaCalculationService]'. Under 'Service parameters', the 'Delta calculation method' is 'context'. The 'Clear delete attribute' checkbox is checked and highlighted with a red box. The 'Delete status attribute' is 'Delete Status (DeleteStatus)'. The 'From file Attachment ID' is 'convert'. The 'Output file Attachment ID' is 'delta'. There are 'Save' and 'Cancel' buttons at the bottom right.

Method 2: Using 'Set import status attributes' Operation

1. Within the Delta Calculation State of the importer workflow, identify the attribute configured in the 'Delete status attribute' parameter.
2. Configure the business rule to include the same attribute within the 'Deleted object attribute' parameter in the 'Set import status attributes' operation. For more information about accessing this parameter and business action, see the topic **Business Action: Set Import Status Attributes**. In the example below, 'Delete Status' attribute is configured in the 'Set import status attributes' operation within the business action 'Set Change Flags.'



3. Configure the **Import** state of the workflow with the updated business action discussed in the previous step. For detailed procedures on updating the workflow settings, see the topic **Business Action: Set Import Status Attributes**. In the example below, the Import state in the AutoCare ACES Import workflow is configured with Set Change Flags business action.

AutoCare ACES Import (AutoCareACESAppImport) @

State Editor

Edit

Deadline/Escalation	Comments	Web UI Screen Mapping	Mandatory Attributes
State	On Entry	On Exit	Assignee
Business Rule	(AutoCareACESAppImportImport)		
Name	ACES Import		
Description			
Type	Action		
Scope	Local		
Run as privileged	<input type="checkbox"/>		

Operations Dependencies Applies if

RunBackgroundProcessServiceAction: queue_Import, Service: 'ImportingFlowImportService'

Edit Operation

Run background process action

QueueID: queue_Import

Background Process Service: [ImportingFlowImportService]

Service parameters

File Attachment ID: delta

Import action: Set Change Flags (SetChangeFlags)

Import condition: Validate Application on import (ValidateApplicati)

Save Cancel

Business Action: Sync ACES Applications Between PIES Parts

This rule is a business action found under the Automotive menu that will synchronize all ACES applications between two PIES Items by following a reference between the parts.

Note: This rule cannot be used to move applications outside of the AutoCare model (using PIES Item and ACES Application object types) as the functionality for construction of the application ID and other related data relies on the standard model.

The rule has a single parameter (Reference Type), which requires selection of a reference. For the rule to function properly, the selected reference must be a product reference with the PIES Item object type (ID=AC_PIESItem) as both the source and the target, and the rule must be set with the PIES Item as the valid object type.

The action evaluates all ACES applications that are child to the selected part on which the rule is being executed, and the applications on the target(s) of the reference indicated in the business rule configuration. The records are then synchronized between the parts, excluding duplicates. For example, assume Part A has 3 records and Part B has 2 records. One of the records on each part is a match to the other, meaning that all data for the application is identical, except the part number. Following the sync, both parts will have 4 records.

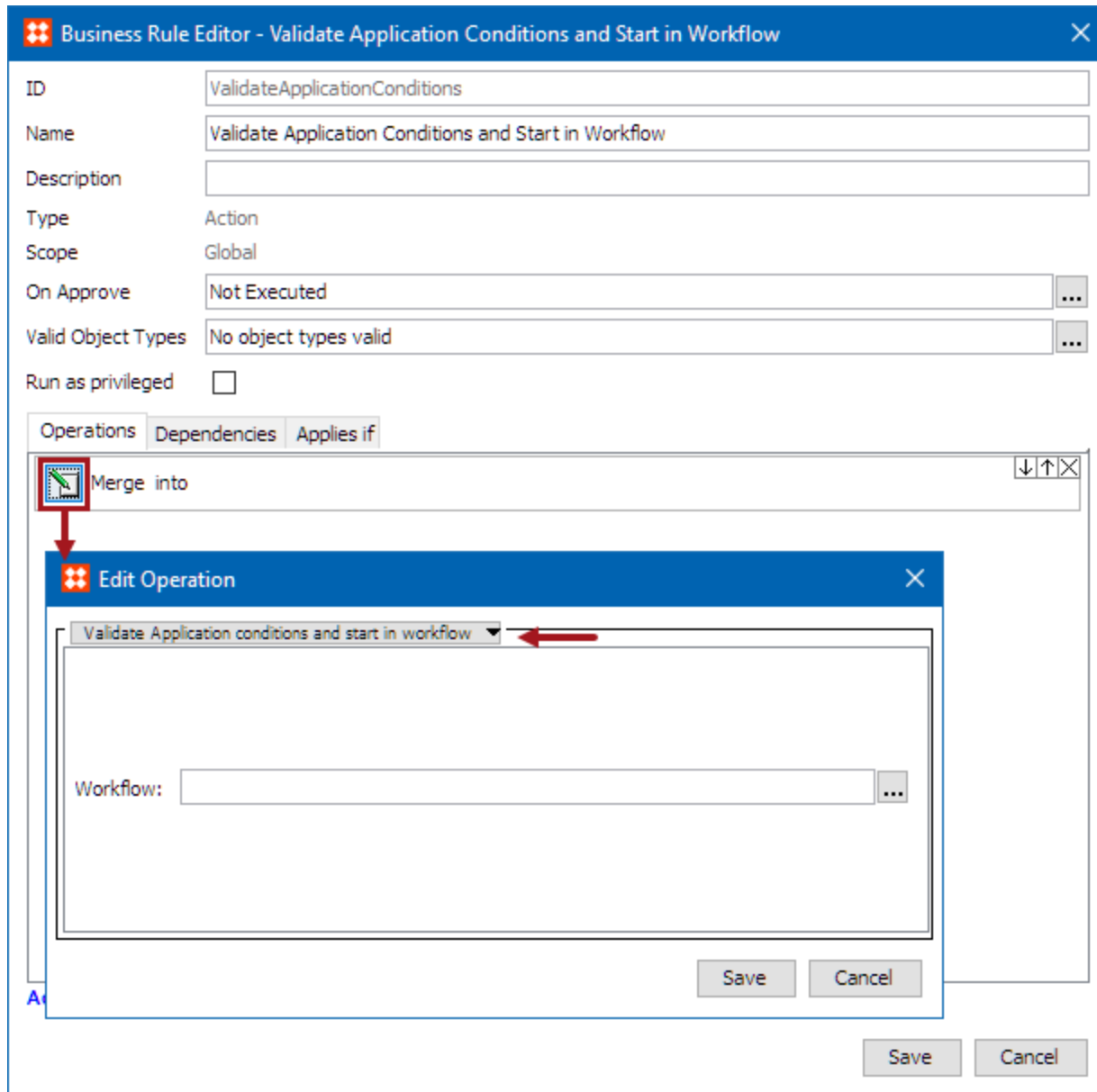
When a sync occurs, the applications are copied between the referenced parts with all data on the original records retained, except the part number. The part that the application is being applied to is written as the part number in both the application name and the hash function of the application ID, as well as within the ACES Replacement Context metadata value. The brand element of the replacement context is also updated if the part that the application is being moved to is from another brand (indicated by the value of the attribute with ID=AC_PIES_ITEMBrandAAIAID on the PIES Item).

Important: ACES supports only a single asset being linked to any application. If multiple assets are present on an application, all will be linked to the record being created by the sync, but only one will be accounted for in the application ID. This could lead to duplicate applications in the system so it is important to ensure that the standard is adhered to and only one asset is present for any application.

Business Action: Validate Application Conditions and Start in Workflow

The 'Validate Application conditions and start in workflow' operation is found within the STEP Workbench Business Rule Editor under the Automotive menu and requires population of a single parameter (Workflow). Setup is required in the STEP Workbench for the action to be available to users.

Important: This business action can ONLY be used within the Import state of a workflow.



Configuring this business action in the Import state of the relevant workflow checks for the application with invalid configurations and, when a record is found that does not meet the condition, it initiates the invalid application records into a workflow configured within Workflow parameter of the business action.

Note: The 'Validate Application conditions and start in workflow' business action does the same check as 'Validate Application on Import' business condition, but instead of rejecting the application, the 'Validate Application conditions and start in workflow' business action imports the invalid application records and starts them in the configured workflow.

In order to divert invalid application records into a different workflow when applications with invalid configurations are imported, a business action with the 'Validate Application conditions and start in workflow' operation can be used within the Import state of a workflow. As a prerequisite, one unique workflow that can handle the invalid applications is necessary for the invalid application record import. In all cases, when an importer determines that applications with invalid configurations should be handled separately, it routes those application records to the workflow indicated in the business action configuration. This allows each customer to determine their own strategy for managing applications with invalid configurations.

Configuring Business Action for Applications with Invalid Configuration

To properly configure this business action to handle applications with invalid configurations, the following is needed:

- Create / Identify a Valid Workflow that Handles Applications with Invalid Configurations
- Configure a Business Action to Use the 'Validate Application conditions and start in workflow' Operation
- Update Workflow Settings

Create / Identify a Valid Workflow

1. Create the necessary workflow (or verify that it exists) to be used to handle the applications with invalid configurations.
2. Make the workflow valid for the object type being managed by the import.

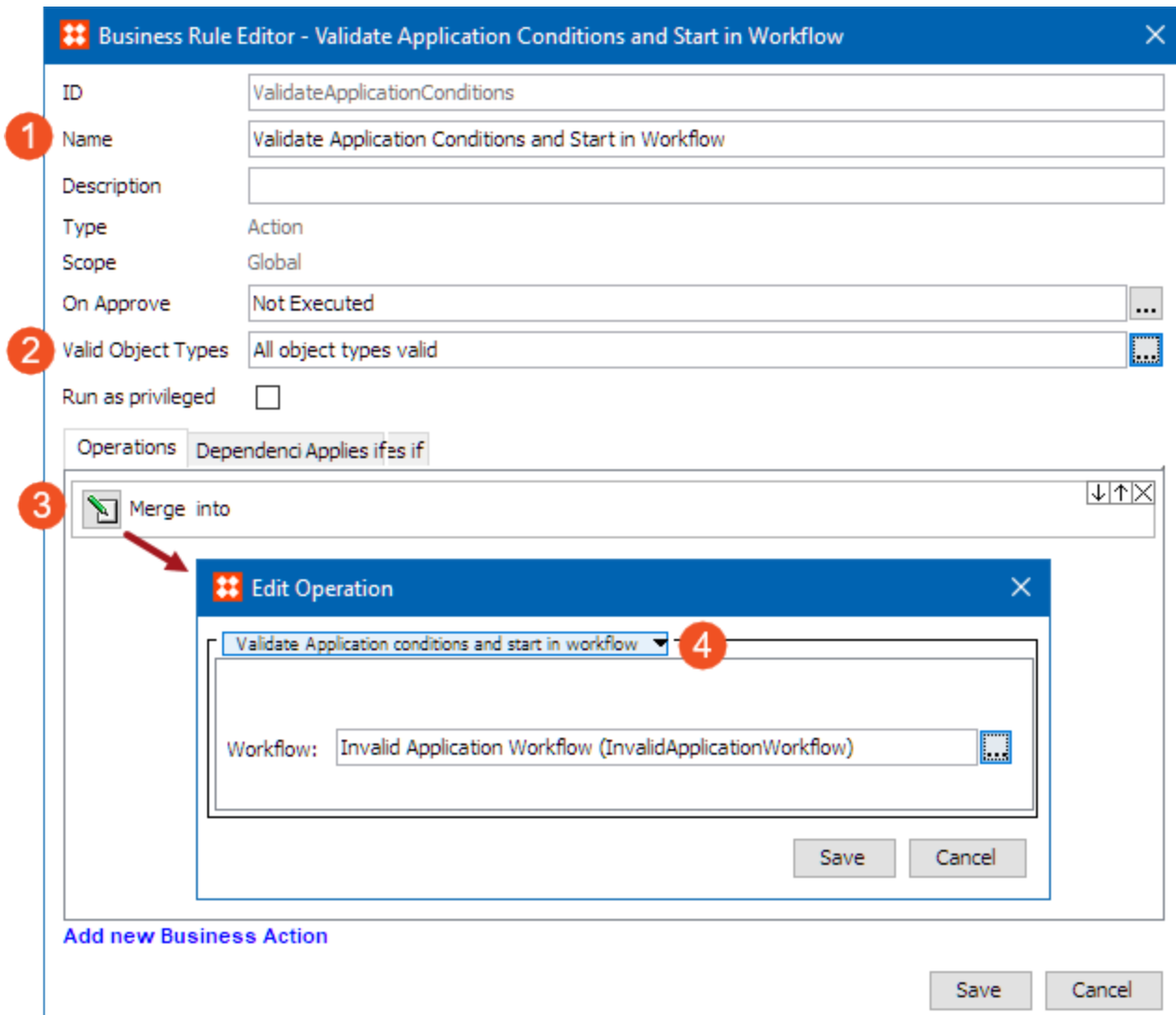
For example:

- the workflow used to handle ACES applications with invalid configurations must be valid on the AC_ACESApplication object type.
- the workflow used to handle TecDoc linkages with invalid configurations must be valid on the DS_Linkage object type.
- the workflow used to handle NAPA applications with invalid configurations must be valid on the NAPA_Application object type.

Configure a Business Action to Use the 'Validate Application conditions and start in workflow' Operation

1. Create a business action and give it a name that users can easily identify. In the example below, the 'Validate Application Conditions and Start in Workflow' name is used.

2. Click the ellipsis button (...) to the right of the Valid Object Types parameter to find and select the necessary object type(s). In the Validate Application Conditions and Start in Workflow example below, 'All object types valid' is selected.
3. On the Operations tab, click the 'Merge into' button, and the Edit Operation dialog will display.
4. Click on the Edit Operation dropdown, click **Automotive**, and then click **Validate Application conditions and start in workflow**.



5. The Workflow parameter will display. Configure the parameter to use the necessary workflow.
 - **Workflow:** Click the ellipsis button (...) to find and select the workflow. The configured workflow should be able to handle the applications with invalid configurations (if any) upon import of the relevant application file. The workflow defined in this parameter should be made valid to the object type being managed by the import. For the example above, the Invalid Application Workflow (created / identified in the previous section) valid for AC_ACESApplication object type is selected.

Note: Upon configuration of the 'Validate Application conditions and start in workflow' business action, the PMDM for Automotive solution only provides an option to import the applications with invalid configuration and start them in another workflow. Users need to determine their own strategy for handling such invalid applications in another workflow.

6. Click the **Save** buttons to save and close the business rule.

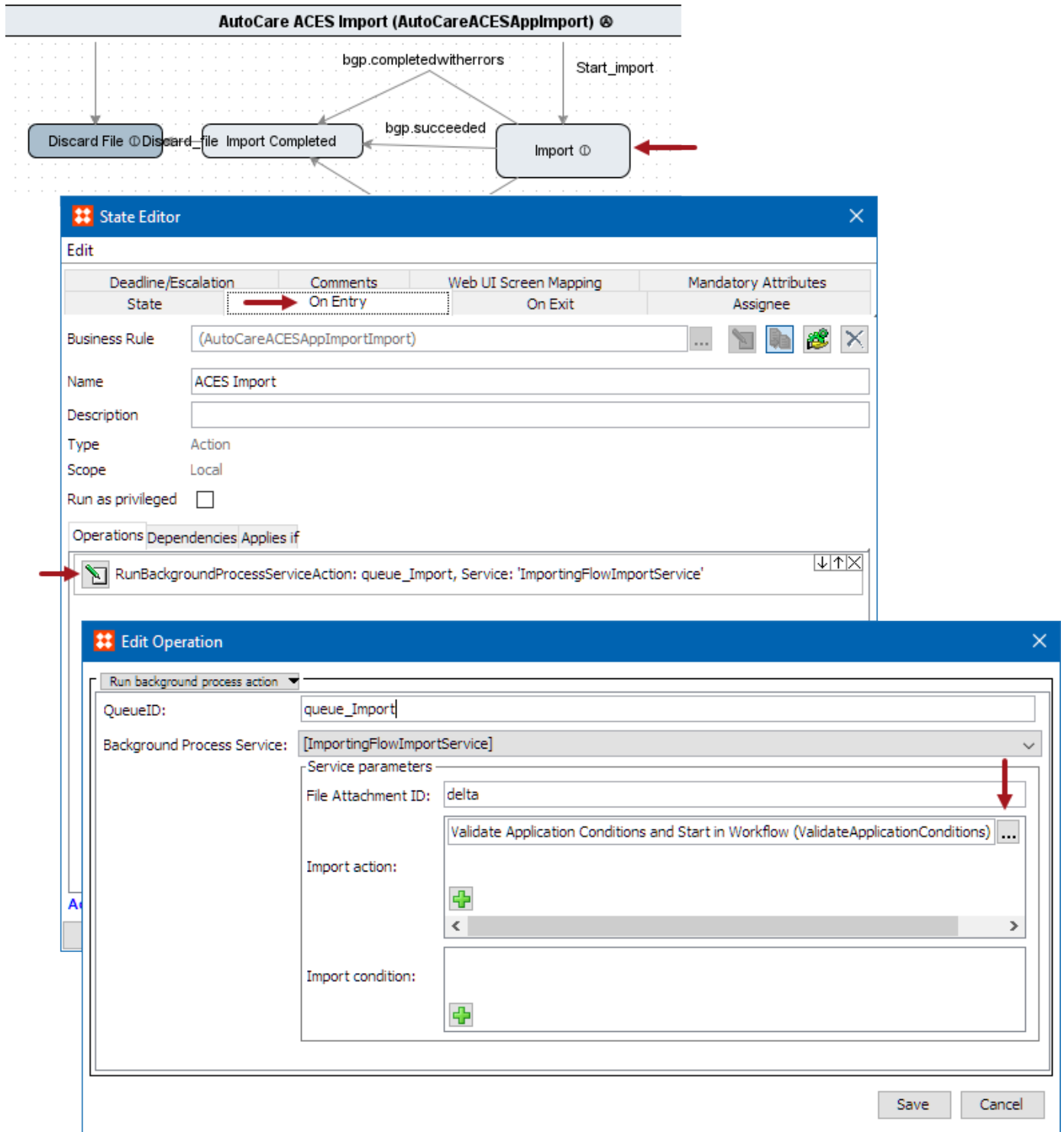
For more information on creating and/or editing business rules, see the **Creating a Business Rule or Library** topic, and the **Editing a Business Rule** topic within the **Business Rules** guide.

Update Workflow Settings

Once the business rule is configured, the necessary workflows need to be updated to use the business rule. Below are the steps to update a workflow to use the business rule created in the previous section.

Important: Because the business rule created in the previous section is configured to work with ACES Application valid object types, the ACES Import Workflow is used for the example below. If the business rule is configured to also work with TecDoc and/or NAPA Applications, then those import workflows would need to also be edited.

1. Go to workbench > System Setup > Workflows > select the workflow to be used when validating applications.
2. Right-click and select **Edit STEP Workflow**.
3. On the STEP Workflow Designer window, double click on the **Import** state to open the State Editor.
4. Select the **On Entry** tab, and within the Operations tab, click the Edit button to display the Edit Operations Dialog (as shown below).



5. In the Edit Operation dialog, click the ellipsis button (...) to the right of the 'Import action' parameter to find and select the business action previously created.
6. Click the **Save** button and exit the workflow.

Business Condition: Check Path for Missing Application

This business condition improves the accuracy of search results involving specific part types by allowing the data model to include a relationship between conditions on applications and different vehicle configurations. This can be helpful when specific part types do not apply to certain vehicle configurations. Additionally, it allows the Web UI Application Manager to search for applications with conditions for all sub models.

Prerequisites

Easy Setup actions automatically create the Missing Application Conditions attribute (MissingApplicationConditions) and the 'Check path for missing application' plugin; however, configuration is required prior to use.

When Easy Setup actions are used to create the Missing Application Conditions attribute (MissingApplicationConditions), the attribute is also made valid for NAPA MPCC, Part Terminology, and Standard Assembly GA object types. This validity must be applied manually if the attribute was created manually.

When Easy Setup actions are completed, the 'Check path for missing applications' plugin is created. However, the business condition must be made valid for the necessary object types.

Improving Search Result Accuracy

When an Application Manager is used to search for existing and missing applications for a specific part type, it is possible to get results that are not accurate based upon the relationship between the vehicle configuration and options for a part.

For example, when searching for missing applications for spark plugs on a 2013 Audi A3, the result table will list both Gas and Diesel engines, since Diesel engines do not use spark plugs, this is inaccurate. However, when the validation path functionality is implemented in conjunction with the 'Missing Application Conditions' attribute and the 'Check path for missing application' business condition, the search results will display more accurately because the application coverage will consider both the Condition and Part Type values.

Existing and Missing Applications | All Brands | Report | Clear Criteria | Search

Select all | Clear filter | Compressed view

Vehicle	Part Terminology	Part Number	ACES Sub Model	Options	Notes
<input type="checkbox"/> 2013 Audi A3	Spark Plug	00729571		ACES Fuel Type: GAS	
<input type="checkbox"/> 2013 Audi A3	Spark Plug			ACES Fuel Type: DIESEL	

Number of items : 2

Once the application coverage is configured to consider the Condition and Part Type values, then searches like the 2013 Audi A3 example above will no longer include the DIESEL vehicle configuration results.

For more information, see the **Missing Application Coverage Functionality** topic within this guide.

For configuration instructions, see the **Configuring the Check Path for Missing Application in Workbench** topic of this guide.

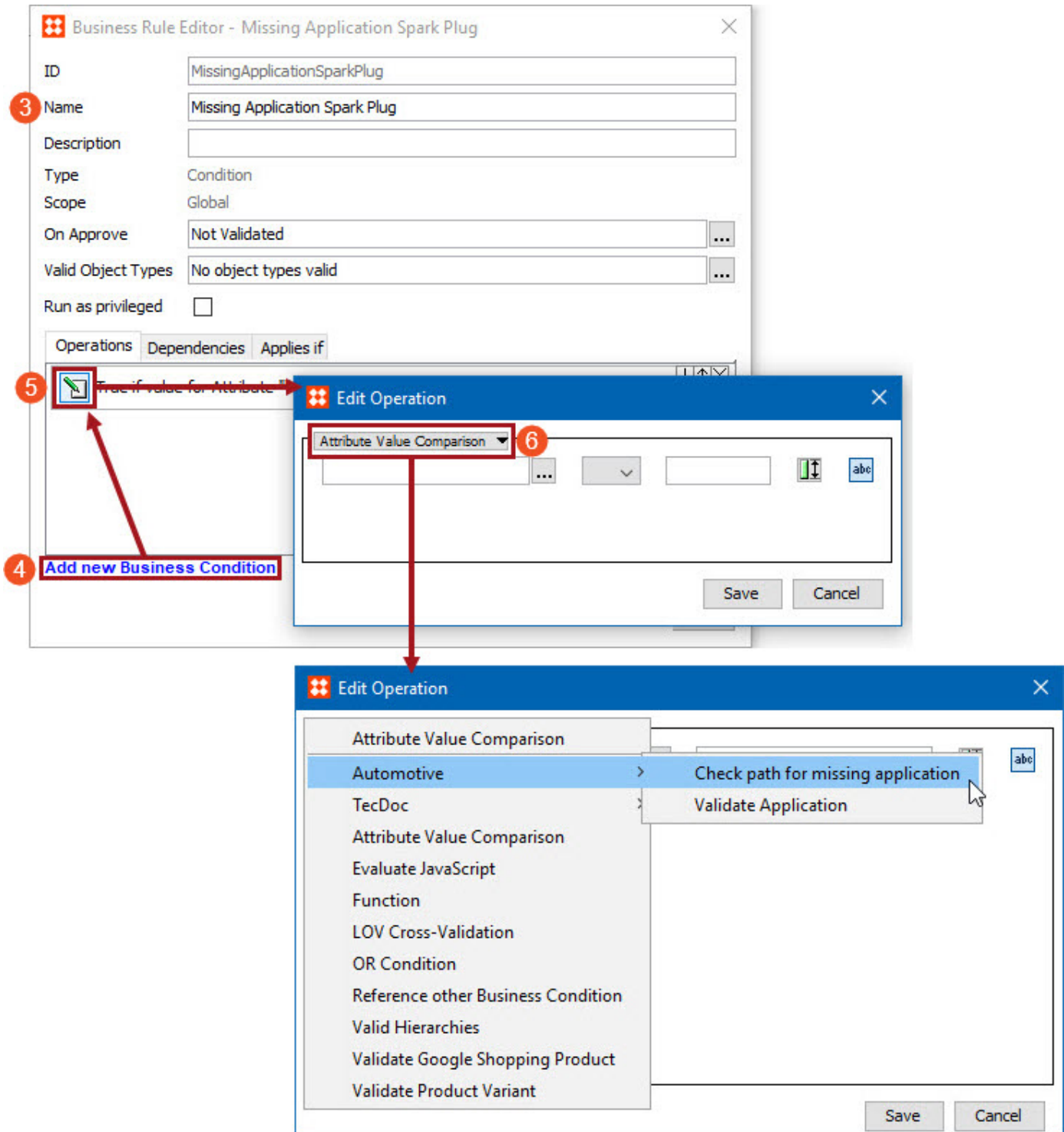
Configuring the Check Path for Missing Application Business Condition in Workbench

Implementing this solution includes configuring a Business Condition using the 'Check path for missing application' plugin, and then adding the Business Condition to the 'Missing Application Conditions' attribute.

Configuring the Business Condition

This section provides steps for configuring a Business Condition using the 'Check path for missing application' plugin within Workbench.

1. Confirm the 'Missing Application Conditions' attribute (MissingApplicationConditions) exists, otherwise create it.
2. Confirm the Missing Application Conditions attribute (MissingApplicationConditions) is valid for the necessary object types for each standard being used. For example, when Easy Setup Actions are completed, the following are made valid when the AutoCare, NAPA, and TecDoc standards are used: Part Terminology, NAPA MPCC, and Standard Assembly GA.
3. Create a new Business Condition with a name that represents the Condition or Part Type value to be filtered. In the example below, the Spark Plug Part Type value will determine if a vehicle with the Diesel option will display within the results table, thus the Name is **Missing Application Spark Plug**.
4. Within the Business Rule Editor, click the **Add new Business Condition** link, and a Business Condition will display within the Operations tab.
5. Click the **Edit icon**, and an Edit Operation dialog will display.
6. Click the dropdown within the Edit Operation dialog to select the **Automotive** option, and then the **Check path for missing application** plugin (as shown below).



7. Once the plugin is selected, the Edit Operation dialog will display the following parameters:

- **Path:** Defines the reference that should be followed on the application to retrieve the attribute value defined in the path. The Path information for an attribute lives on a referenced object that is referenced from another referenced object from the child of the actual vehicle. For this example, the Path information for Fuel Type (GAS/DIESEL) is stored on the Automotive Validation Path attribute (as shown below).

Note: For more information on the Automotive Validation Path, see the Automotive Validation Path Functionality section within the **Automotive Reference Guide** found within the **Solution Enablement** section of **STEP Online Help**.

The screenshot shows the 'System Setup' window with a tree view on the left and a detailed view of an attribute on the right. In the tree view, 'Fuel Type' is highlighted with a red box. In the detailed view, the 'Automotive Validation Path' attribute is also highlighted with a red box, showing its value as 'child.reference[type:'AC_VehideToEngineConfig'].attribute[id:'AC_VCdbFuelType']'.

Name	Value
ID	AC_ACESFuelType
Name	Fuel Type
Last edited by	2018-06-27 09:08:47 by USERN
Full Text Indexable	No
Externally Maintained	No
Completeness Score	
Hierarchical Filtering	None
Calculated	No
Type	Specification
ATTR_Sort Nr	123
ATTR_Sucesser To	abc
ATTR_Valid For	
Attribute Help Text	abc
Automotive Validation Path	child.reference[type:'AC_VehideToEngineConfig'].attribute[id:'AC_VCdbFuelType']


The Path information (attribute value) must be copied from the Fuel Type specification Automotive Validation Path attribute, and pasted into the Path parameter (as shown below).

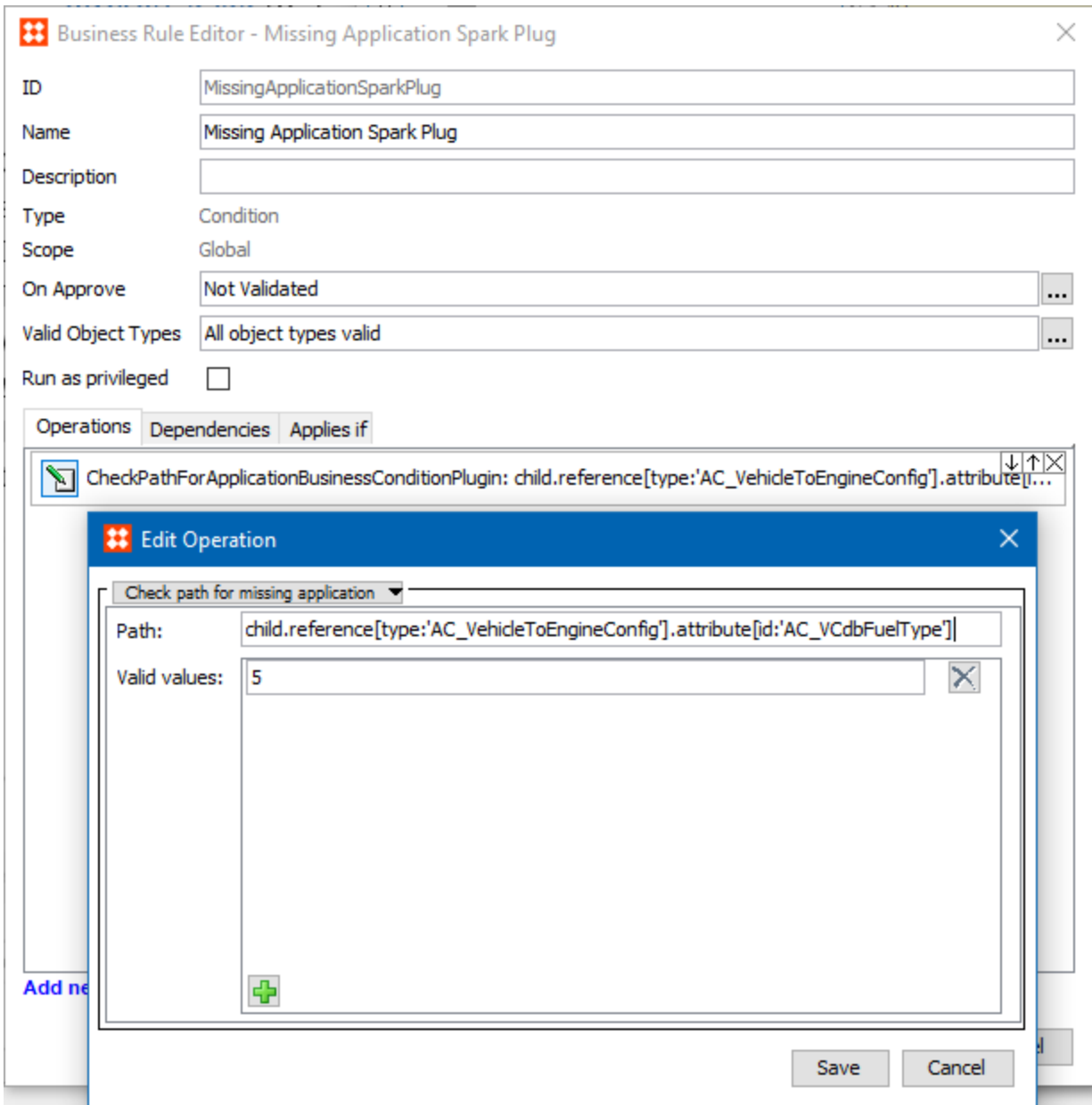
The 'Edit Operation' dialog box shows the 'Path' field populated with the value 'child.reference[type:'AC_VehideToEngineConfig'].attribute[id:'AC_VCdbFuelType']'. There is a 'Valid values' field below it, which is currently empty. At the bottom of the dialog are 'Save' and 'Cancel' buttons.


- **Valid Values:** One or more IDs of an LOV value must be entered. IDs entered will be displayed within the Application Manager Results Table when applicable. In the example below, the value ID for the Fuel Type - List of Vales GAS is 5.

Values	Value ID
> BI-FUEL	13
> BIODIESEL	23
> CNG	8
> DIESEL	6
> ELECTRIC	10
> ELECTRIC/DIESEL	16
> ELECTRIC/FLEX	17
> ELECTRIC/GAS	14
> ELECTRIC/HYDROGEN	22
> FLEX	7
> GAS	5
> LPG	9
> N/A	20
> U/K	18

[Add Value](#)

8. Within the Edit Operation dialog, click the **green plus sign** icon () to add a value ID to the **Valid values** parameter.
9. In the example below the number 5 is entered so that the GAS Fuel Type will display when the Spark Plug part type is searched.



10. Optionally, click the **green plus sign** icon () to add more value IDs, if more Fuel Types should display when the Spark Plug part type is searched.
11. Click the **Save** and **Close** buttons to save the Business Condition.

Note: Copy the ID of the Business Condition for the next steps.

Adding the Business Condition to the Missing Application Conditions Attribute

Once the Business Condition has been configured, it must be added to the Missing Application Conditions attribute. Below are the steps necessary to add the Business Condition to the Part Type attribute.

1. Go to the Part Type that needs its results improved. For the example below, the 'Spark Plug' Part Type is used.
2. Go to the 'Missing Application Conditions' attribute, double click to add a value, and the Value Editor will display.
3. Click the **Add Value** link, and a blank row will display.
4. Paste the ID of the Business Condition into the value row.
5. Click the **OK** button to save, and the Business Condition ID will display as the attribute value.

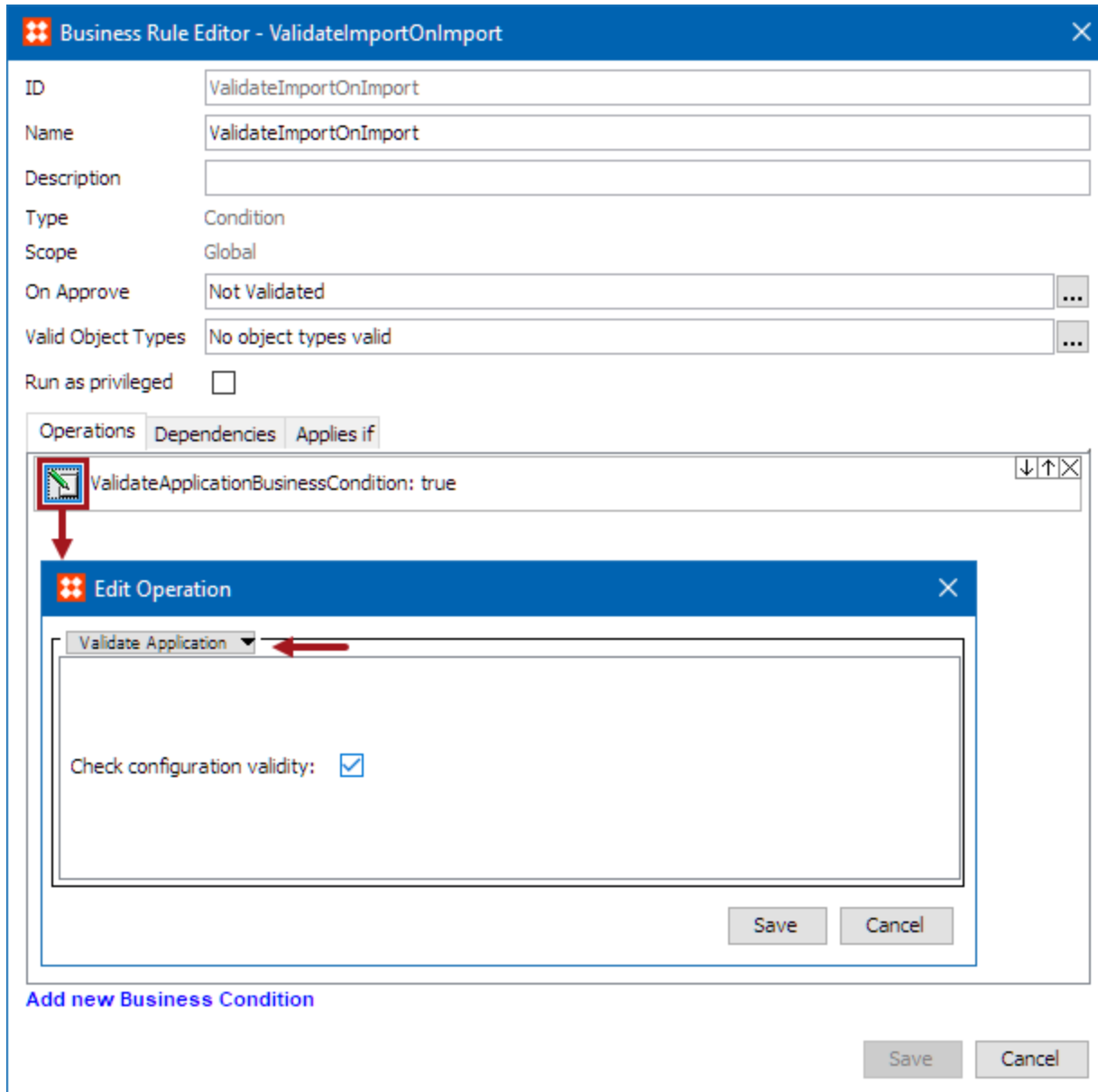
The screenshot displays the 'Spark Plug rev.0.3 - Classification' record in the Stibo Systems interface. The record details include:

- ID:** AC_PartTerminology_7212
- Name:** Spark Plug
- Object Type:** Part Terminology
- Revision:** 0.3 Last edited by STEPSYS on Mon May 28 08:45:55 EDT 2018
- Approved:** Never Been Approved
- Translation:** Not Translated
- Path:** Classification 1 root/AutoCare Root/Parts Categorization (PCdb)/Ig...
- Missing Application Conditions:** (This field is highlighted with a red circle 2)

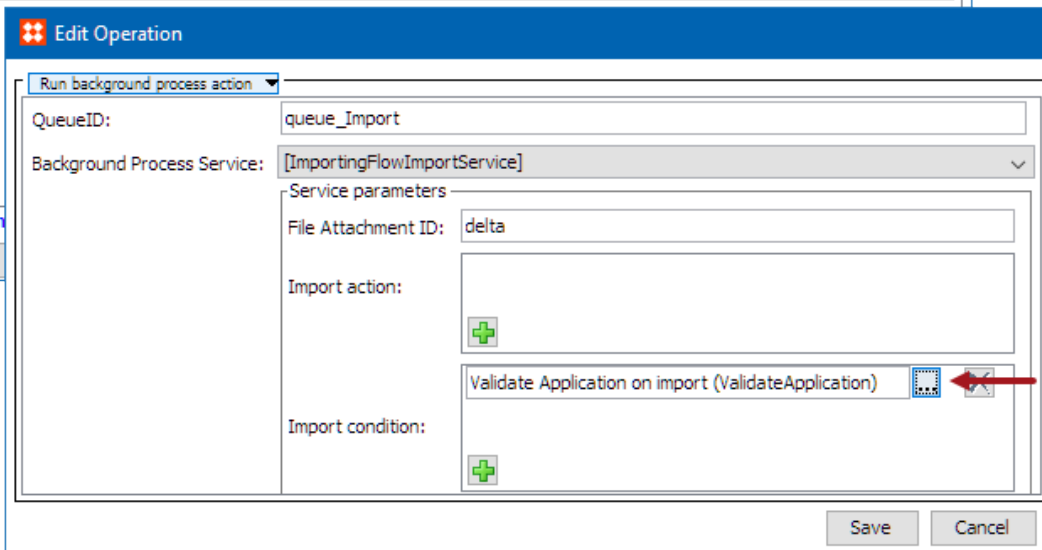
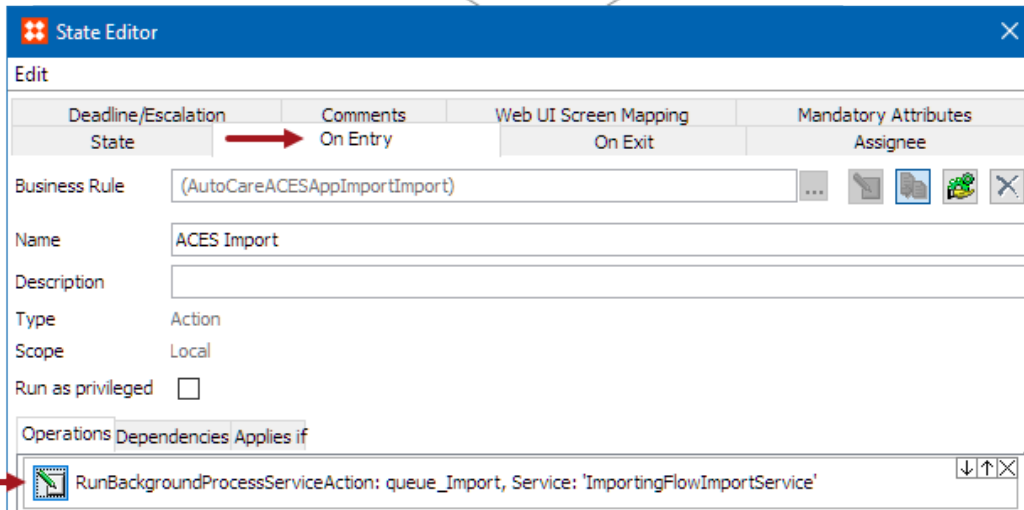
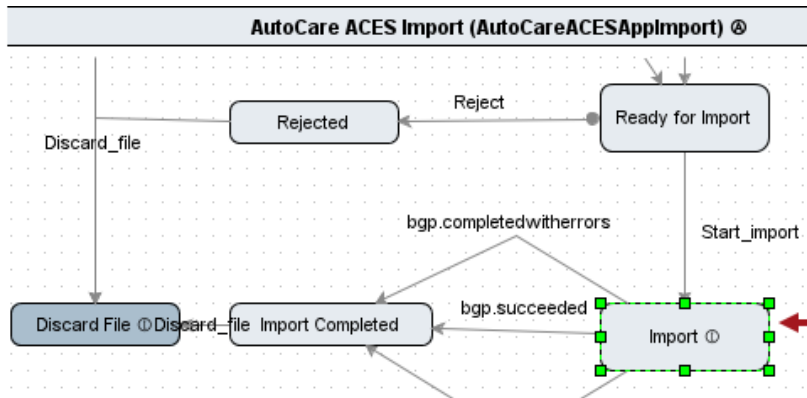
The 'Value Editor' dialog box is open, showing a list of values. The value 'MissingApplicationSparkPlug' is selected (highlighted with a red circle 4). The 'Add Value' button is visible (highlighted with a red circle 3), and the 'OK' button is highlighted with a red circle 5.

Business Condition: Validate Application on Import

The 'Validate Application' operation is found within the STEP Workbench Business Rule Editor under the Automotive menu and requires selecting / deselecting of the 'Check configuration validity' parameter.



The 'Validate Application' operation is executed via the business condition 'Validate Application on import.' Running the Easy Setup of the import process automatically creates the 'Validate Application on import' business condition and adds it to the Import state of the relevant workflow.



The 'Check configuration validity' parameter determines whether the applications with invalid configurations (applications that do not have both a part type and vehicle / assembly that exists in the STEP database) are imported or not.

Checking the 'Check configuration validity' parameter means applications with invalid configurations are not imported. When an application record is found that does not meet the condition, an error is written to the execution

report of the import process and the record is not imported. The error contains the application ID from the ACES file, as well as IDs of the objects that it failed on.

Below is an example of an error that is written to the execution report when an assembly (Base Vehicle ID) mentioned in the ACES file does not exist in the STEP database but the ACES file is trying to reference an application record to the missing assembly.

☰ 🔄 Main English US ⚙️ 🏠

Background Process Details

ID: BGP_108465
 Started By: USERB
 Description: Running background process ImportingFlowImportService from RunBackgroundProcessServiceAction
 Template ID: ImportingFlowImportService
 Status: ▲ Completed With Errors
 Started: 9/12/19 11:54:19 AM
 Finished: 9/12/19 12:25:16 PM
 Elapsed: 31 m 1 s

Export

ID	Type		Text
26230	Error		Error: An application (id=28769) for part: 1K 0121 251 CJ had conditions not matching vehicle
26240	Error		Error in this import 02_Applications.xml setting completed with errors - Error: An application (id=28790) for part: 8K0121251R had conditions not matching vehicle
26250	Error		Error in this import 02_Applications.xml setting completed with errors - Error: An application (id=28856) for part: 8K0121251R had conditions not matching vehicle
26270	Error	➔	Error in this import 02_Applications.xml setting completed with errors - Error: An application (id=29649) for part: 1K0959455P has no assembly target
26280	Info		Finished handling 2 deferred Cross References, time: 6 ms
26290	Info		End handling file '02_Applications.xml'. Duration: 30 mins 49 secs. 0.00 lines per second
26300	Warning		File(s) was imported with error, flagging status as completed with errors.
26310	Info		Import completed
26320	Info		Completed with errors: 1535

⏪ < 2576-2591 of 2591

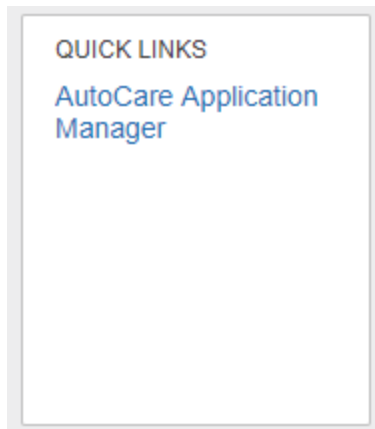
If the 'Check configuration validity' parameter is not checked, importing ACES applications that have invalid configurations will pass the import and create the applications.

Application Manager

Application Manager is an interactive and robust tool that combines the power of an easy-to-use intelligent search interface with an effective results table. This allows Web UI users to quickly and easily build custom searches involving valid combinations of different vehicle types, makes, models, years, options, regions, and part types. Once a broad (or narrow) search criteria is selected, users can view, create, edit, and delete part applications, along with many other aspects of the search results. Additionally, users can easily create an Excel report for the customized search criteria results by clicking the Report button.

When Easy Setup actions have been used to create a standard data model, by default, much of the Application Manager is ready to be used, and for ease of access a link to the Application Manager screen is added to the Quick Links widget on the Web UI Homepage. In the example below, Easy Setup has configured the AutoCare Application Manager and provided a link within the Quick Links widget.

Note: Prior to completing the Easy Setup actions, if links exist in the Quick Links widget, then the Application Manager link will be added at the very bottom of the list.



Important: An error will occur when trying to access an Application Manager screen if the Vehicle table classifications have not been properly configured. When this error displays, the user must log out and then back in to the Web UI. Once the Vehicle reference data is imported, the link to the Application Manager will function properly.

Until the Vehicle table classification is properly configured, the Quick Links widget cannot be used to access the Application Manager screen because when Easy Setup actions are used to configure an Application Manager, the mandatory parameters within the Designer are populated with Vehicle root nodes. Thus, if Vehicle tables have not been imported, and the link to the Application Manager within the Quick Links widget is clicked, the error shown below will display because the Vehicle root does not exist.

Unexpected error. Please contact your system administrator. Index: 0, Size: 0

Note: The Application Manager is highly configurable via the Web UI Designer, and more than one Application Manager can be created within each Web UI (i.e., for each automotive standard). Therefore, it is expected that a Web UI administrator will configure an Application Manager in a manner that best meets their user's needs. Thus, your Application Manager may look different from what is described within this guide. For more information on configuring an Application Manager, see the **Configuring Application Manager** topic.

The top half of the Application Manager screen offers features related to the intelligent search interface. The bottom half of the screen is dedicated to the search results table and its toolbar. Both halves of the screen can benefit from the automotive validation path functionality.

The screenshot displays the Application Manager search interface. At the top, there are five filter panels: Make/Model (Toyota), Year (2018), Sub Model (TRD Pro), Region (United States (USA)), and Part Type (Spark Plug). Below these filters is a toolbar with options like 'New Toyota TRD Pro', 'Existing and Missing Applications', 'All Brands', 'Report', 'Clear All', and 'Search'. The search results table has columns for Vehicle, Part Terminology, Part Number, Options, Notes, Qualifiers, and Assets. Two results are shown: 2015 Toyota 4Runner and 2015 Toyota Tacoma, both with Spark Plug parts. The number of items is 14.

Vehicle	Part Terminology	Part Number	Options	Notes	Qualifiers	Assets
2015 Toyota 4Runner	Spark Plug		United States (USA), TRD Pro			
2015 Toyota Tacoma	Spark Plug		United States (USA), TRD Pro			

In the example above, an Application Manager is used to find existing and missing spark plug applications for Toyotas manufactured in the United States with the TRD Pro sub model option.

This section provides details on the use and configuration of the many powerful features that can be configured within an Application Manager screen with the following:

- Variations Across Standards
- Intelligent Search Interface
- Results Table and Toolbar
- Automotive Validation Path
- Configuring Application Manager

Variations Across Standards

There are terminology variations within the Application Manager for each standard, as well as minor variations in the available headers for the results table.

For example:

- AutoCare offers search results column headers labeled as Vehicle, Part Terminology, and Part.
- AutoCare contains a Qualifiers header, whereas a TecDoc editor does not.
- NAPA offers search results column headers labeled as Vehicle, MPCC, NAPA Product, Conditions, and Comments.
- TecDoc offers search results column headers labeled as Assembly, Generic Article, and Article.

Despite these differences, the overall functionality is the same, allowing users to search for applications, and view and edit application data. The upper half of the screen provides the search functionality, and the lower half of the screen displays the results and allows for editing of application records.

Note: Each implementation has the option to add and remove data to the results display. Additionally, the vehicle options / criteria that are displayed vary dynamically by the part types included in the results, based on configuration options which are described below.

Intelligent Search Interface

The top half of the Application Manager screen offers features related to the intelligent search interface. The bottom half of the screen is dedicated to the search results table and its toolbar. For more information about the Results Table, see the **Results Table and Toolbar** topic. This section provides an overview of the interface, and then addresses the following:

- **Using Intelligent Search Interface**
- **Saved Search Tool**
- **Missing Application Coverage Functionality**
- **Application Coverage Report**

Intelligent Search Interface Overview

The intelligent search interface of an Application Manager screen is composed of the following:

The screenshot displays the Intelligent Search Interface with the following components and annotations:

- 1:** Vehicle type icon (car) on the left side.
- 2:** Search criteria fields: Make/Model (Toyota), Year, Sub Model (TRD Pro), Region (United States (USA)), and Part Type (Spark Plug). An AND operator is visible between Sub Model and Region.
- 3:** Saved search tool showing 'New Toyota TRD Pro' with save, edit, and delete icons.
- 4:** Filter dropdown menu showing 'Existing and Missing Applications'.
- 5:** Filter dropdown menu showing 'All Brands'.
- 6:** 'Report' button.
- 7:** 'Clear All' button.
- 8:** 'Search' button.

- 1. Vehicle type icon:** A visual representation of the 'Vehicle Type Search Panel' contents. Clicking a vehicle type icon displays the configured Vehicle Type Search Panel for the vehicle type (i.e., Personal Cars, Buses, Marine, Street Bikes). Hovering over the icon will display the name configured for the Vehicle Type Search Panel. Each Application Manager must have at least one vehicle type icon. Up to six icons can be displayed within the same Application Manager. For a list of available icons, and more information about the Vehicle type icon, see the **Configuring Application Manager** topic within this guide.
- 2. Vehicle Type Search Panel:** Has many configuration parameters, and can consist of up to four 'search box types' (Make/Model, Year, Options Group, and Part Type). Because many of the search boxes are synchronized, search criteria can be entered into any one of the search boxes in any order, and the available search criteria will display in the typeahead dropdown based upon the criteria entered within the other search boxes. For more information on using the Search Boxes within the Vehicle Type Search Panel, see the **Using Intelligent Search Interface** topic within this guide.

Note: Though only four search box types exist, the above screenshot displays five. This is possible because the search boxes labeled Sub Model and Region are both part of the Options Group search box type. For more information on the Options search boxes, see the **Using the Options Search Boxes** topic within this guide.

3. **Saved search tool:** Allows users to save previously selected search criteria with custom names, and then quickly populate previously saved search criteria by selecting the name of a saved search from the saved search dropdown. The tool consists of a saved search dropdown field, as well as Save, Rename, and Delete icons. For more information about the saved search tool, see the **Saved Search Tool** section of this guide.
4. **Existing and missing applications dropdown:** Allows users to determine if their search should or should not contain missing applications. To search or report on missing applications, users can select either the 'Missing Applications Only' or 'Existing and Missing Applications' options from the Existing and Missing Applications dropdown. Additionally, users can select the 'Existing Applications Only' option, if the desired results should only display existing applications. For more information about missing applications, see the **Missing Application Coverage Functionality** topic within this guide.
5. **Hierarchy Restriction dropdown:** Allows users to restrict the Application Manager search results by object type. Users can select from a pre-configured dropdown list of object types. This can be helpful when search results needs to be filtered by Brand, Manufacturer, OEM, Product Line, etc. When the parameter is blank, the Hierarchy Restriction dropdown will not display within the Application Manager. In other words, this option can be removed from an Application Manager screen by removing all object types from the Hierarchy Restriction Object Types parameter. For more information, see the **Configuring Application Manager** topic within this guide.

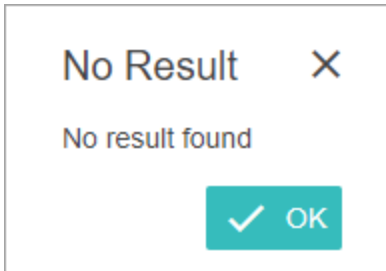
Note: Both the 'Existing and missing applications' and 'Hierarchy Restriction' dropdowns are tied to the Vehicle type icon. Each time a Vehicle Type icon is selected, the two dropdowns will reset to their defaults ('Existing Applications Only' and 'All Brands').

6. **Application Coverage Report button:** Allows users to export an Application Coverage Report in an Excel spreadsheet (XLSX file type). This report provides results based upon the combination of the search criteria, Existing and Missing Applications dropdown, and Brand dropdown selections. Because this report is built to work with the intelligent search interface of the Application Manager, it is extremely configurable. This allows users the flexibility to choose the report criteria that best suits their needs on demand, and then easily export the search results. The Report button remains disabled until at least one search box is populated. For more information, see the **Application Coverage Report** topic within this guide.
7. **Clear All link:** Allows users to easily clear all search criteria, results, and filters applied to Results Table. The Clear All link remains disabled until at least one criterion for one of the search boxes is populated. When text is typed into a search box, but is not yet selected, clicking the link will also remove the text typed into the search box. Additionally, criterion can be cleared one at a time by clicking the 'X' to the right of a value within a search box. Clearing criteria in either way does not update the search results. Once any unnecessary criteria are removed, the Report or Search buttons must be used to update the results table.

Note: When the results table is populated, and one or more criteria are changed, clicking the Search button will initiate a new search and the results table will only display results related to that search.

8. **Search button:** When clicked, the Search button will use the criteria provided within the search boxes, selections made within the Existing and Missing Applications dropdown, and/or Brand dropdown, and then provide any results in the results table. The Search button remains disabled until at least search boxes is populated. Requiring only one criterion within the Make/Model, Part Type, or Options search boxes, allows

users to conduct very broad searches. For example, to view existing applications for Spark Plugs, type 'Spark Plugs' into the Part Type search box. With 'Existing Applications Only' selected, click the Search button to display all the existing applications for Spark Plugs in the results table. However, to view missing applications, this type of broad search would be very time consuming, therefore it is recommended to use additional search criteria, or use the Application Coverage Report button, instead of the Search button. When the Search button is clicked, and no results are available, a 'No Result' dialog will display, as shown below. Clicking the **OK** button closes the dialog, and allows the remaining search criteria to be adjusted.



Using Intelligent Search Interface

The Application Manager intelligent search interface has been carefully designed to allow users to quickly and easily select criteria for valid combinations of different vehicle types, makes, models, years, options (i.e., sub models, regions), and part types. This allows users to create accurate search criteria on the fly and benefit from the customized search results. This is accomplished by allowing users to select only valid criteria across multiple search boxes per the configuration of the Vehicle Type Search Panel.

Within an Application Manager screen, the Vehicle Type Search Panel can be configured to display up to four search box types. Because the Options Group search box type allows for up to two Options search boxes, up to five search boxes can be displayed. Within each search box, a typeahead field is used to view and select valid search criteria. Because only valid criteria can be selected from each dropdown list, the Application Manager provides automated intelligence for each search. Thanks to the synchronization of the search boxes, the valid criteria is considered upon each selection and within any of the displayed search boxes, no matter the order of entry.

For example, if the Make/Model search box contains a vehicle that was only produced from 1994-2010, then the dropdown within the Year search box will only display those years for selection. Because Chrysler established Ram trucks as a new brand (Make) beginning with the 2011 model year, the available years are displayed as 1994-2010 when Make/Model 'Dodge Ram 3500' is selected (as shown below).

<p>Make/Model</p> <p>Dodge Ram 3500 ×</p>	<p>Year</p>
<p>Enter Make/Model</p>	<p>Enter Year</p>
	<p>1994</p> <p>1995</p> <p>1996</p> <p>1997</p> <p>1998</p> <p>1999</p> <p>2000</p> <p>2001</p> <p>2002</p> <p>2003</p> <p>2004</p> <p>2005</p> <p>2006</p> <p>2007</p> <p>2008</p> <p>2009</p> <p>2010</p>

Note: The Vehicle Type Search Panel, and its search boxes, have many configuration options. If your screen, or search criteria dropdown options display differently than the examples within this section, contact your Web UI administrator. For more information on configuring an Application Manager, see the **Configuring Application Manager** topic within this guide. For information about the general layout and features of the Application Manager, see the **Application Manager** topic within this guide.

However, if the Ram 3500 is added to the Make/Model search box, then the available years display as 1994-2018 (as shown below).

Make/Model	Year
<input type="text" value="Dodge Ram 3500 x"/> <input type="text" value="Ram 3500 x"/>	
<input type="text" value="Enter Make/Model"/>	<input type="text" value="Enter Year"/>
	<ul style="list-style-type: none"> 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018

Additionally, consider being able to easily view the engines used for selected vehicles. In the screenshot below, a Dodge Ram 3500 is selected for the Make/Model search box, and some available years have also been selected. Therefore, when the Engine option is selected for the Options search box, and the cursor is placed within the search box field, only those engines used for those vehicles display for selection.

Make/Model 1 <input type="text" value="Dodge Ram 3500"/>	Year 2 <input type="text" value="2007"/> <input type="text" value="2008"/> <input type="text" value="2009"/> <input type="text" value="2010"/>	Sub Model ▾	Engine ▾ <div style="text-align: center;">AND</div>
Enter Make/Model	Enter Year	Enter Sub Model	Enter Engine 3 <div style="border: 1px solid gray; padding: 2px;"> 5.7L L8, -CC, 345CID 5.9L L6, -CC, 359CID 6.7L L6, -CC, 408CID </div>

Taking this example a step further, after selecting Dodge Ram 3500 for the Make/Model search box, and 6.7L L6, -CC, 408CID for the Engine options search box, users can easily view the valid years the selected engine has been manufactured within the selected vehicle (as shown below).

Make/Model 1 <input type="text" value="Dodge Ram 3500"/>	Year <input type="text"/>	Sub Model ▾	Engine 2 ▾ <input type="text" value="6.7L L6, -CC, 408CID"/>
Enter Make/Model	Enter Year 3 <div style="border: 1px solid gray; padding: 2px;"> 2007 2008 2009 2010 </div>	Enter Sub Model	Enter Engine

This section addresses the following:

- **Using Search Boxes**
- **Search Box Synchronization**
- **Using Make/Model Search Box**
- **Using Year Search Box**

- **Using Options Search Boxes**
- **Using Part Type Search Box**

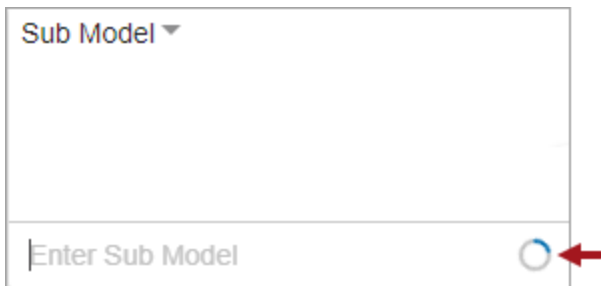
Using Search Boxes

Though each search box contains its own typeahead field, and each typeahead field displays data specific to that search box, the behavior of the field is similar, and common keyboard shortcuts are compatible. Functionality specific to each search box is addressed in subsequent sections of this guide. This section addresses the following functionality that is consistent across the Application Manager search boxes:

- Searching for Criteria
- Selecting Valid Criteria
- Removing Criteria

Searching for Criteria

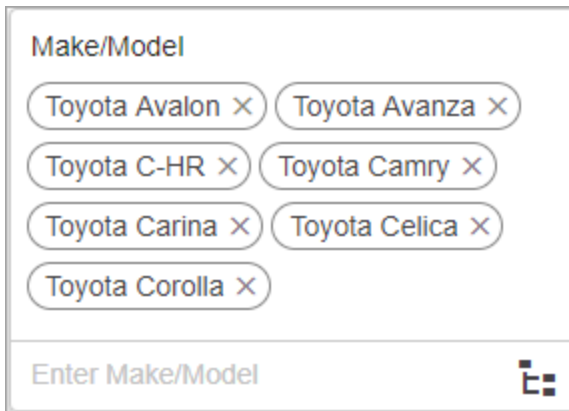
The cursor is activated for each search box either by clicking within the box itself or within the typeahead field at the bottom of the box. Text typed into the field is used to filter and then display only valid criteria within a dropdown list. Typing an asterisk (*) displays any available criteria for selection in the dropdown. Typing additional text allows for a wildcard search to occur against the available criteria. If the display of a dropdown list is delayed, a blue and gray circle will display within the typeahead field indicating the request is in progress (as shown below).



Once the dropdown list for a search box displays, if the list of options is too long to display within the screen, then a scroll bar will display to the right of the list. The scroll feature on a mouse and/or the Up and Down arrows on a keyboard can be used to navigate up and down the dropdown list.

Selecting Valid Criteria

Once one or more desired criteria are highlighted, pressing the Enter key on the keyboard (or using the mouse to select) will add the highlighted criteria to the search box. Each selected criterion will display as a tile within the search box. In the example below, seven Toyota model criterion tiles have been added to the Make/Model search box.



Once a criterion is selected to display within a search box, it will no longer display within the dropdown for selection. This allows users to focus on valid options they have not yet selected.


The Tab key on the keyboard can be used to move the cursor from left to right across different search boxes. Holding the keyboard Shift key down while pressing the Tab key will move the cursor from right to left.

Note: When multiple options are available for selection within an Options search box, the Tab key will move the cursor to the Options dropdown, and the down and up arrow keys can be used to change the selected option.

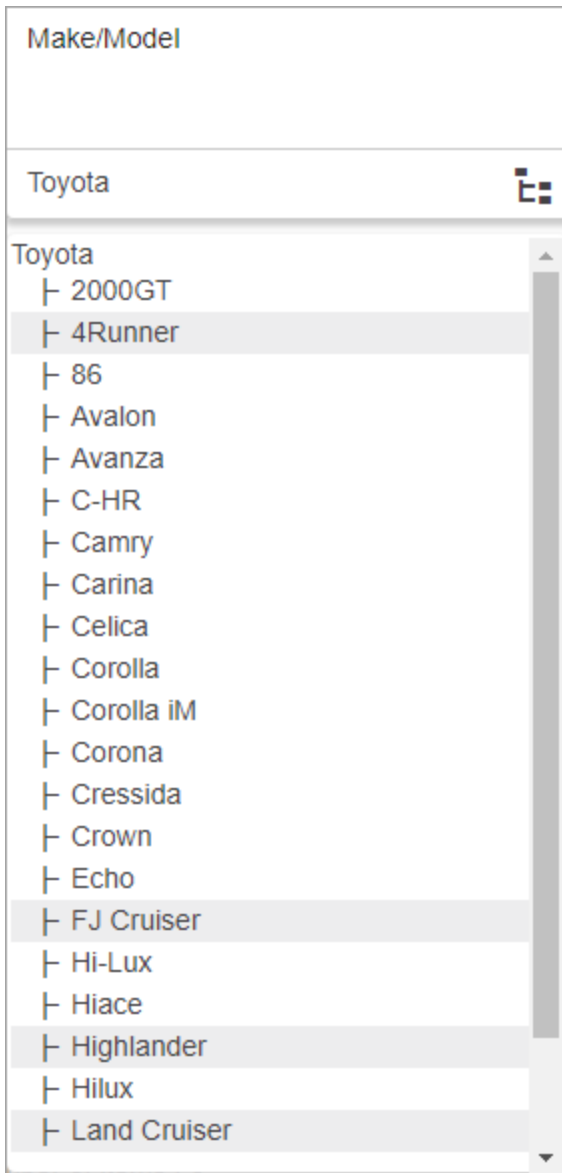
Selecting Multiple Criteria

Within a dropdown list, multiple criteria (either as a group or scattered throughout the list) can be selected and added to a search box. Holding down the Shift key on a keyboard will enable the selection of multiple criteria. While holding down the Shift key, the mouse pointer can be used to click on a criterion listed at the top and/or bottom of a group of desired criteria, next the mouse pointer can be used to click on a criterion listed at the bottom and/or top of that group, resulting in the group of selected criteria being highlighted. Optionally, a keyboard can be used to highlight a collective group by holding down the Shift key, and using the arrow up and down keys to highlight a group of criteria. Once the group is highlighted, pressing the Enter key on the keyboard will add the selected criteria as tiles within the search box.

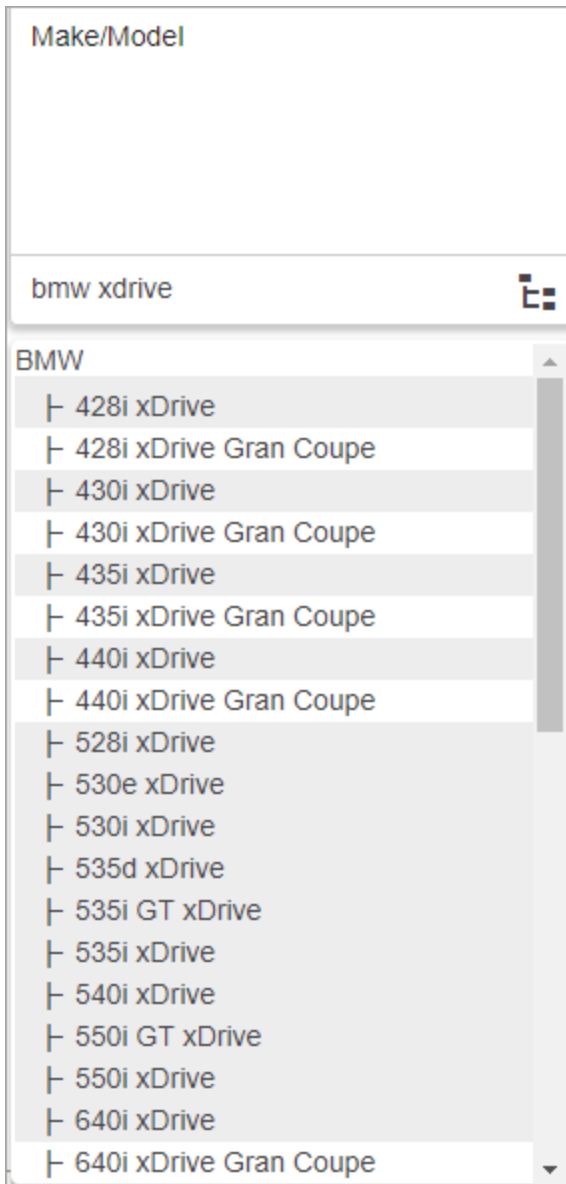
In the example below, after a list of valid criteria options displayed are within the dropdown list, the down arrow key was used to highlight the Audi A4 option, the Shift key was held down, and the down arrow key was used to highlight the two additional Audi model options.

Make/Model
A4 
Audi
└ A4
└ A4 Quattro
└ A4 allroad
Austin
└ A40
Checker
└ A4
Mercedes-Benz
└ A45
└ A45 AMG

To select multiple criteria that are not grouped next to each other in the dropdown, hold down the Ctrl key on the keyboard, and use the mouse to click on the desired criterion. Once the desired criteria are highlighted, pressing the Enter key on the keyboard will add the selected criteria to the search box.



Optionally, when a group of criteria are highlighted, but one or more criteria within the list should not be selected, holding down the Ctrl key on the key board, and then using the mouse to click the undesired criterion will remove the highlight. In the example below, a large group of BMW xDrive Models have been highlighted for selection, and then those with 'Gran Coupe' have been deselected.



Removing Criteria

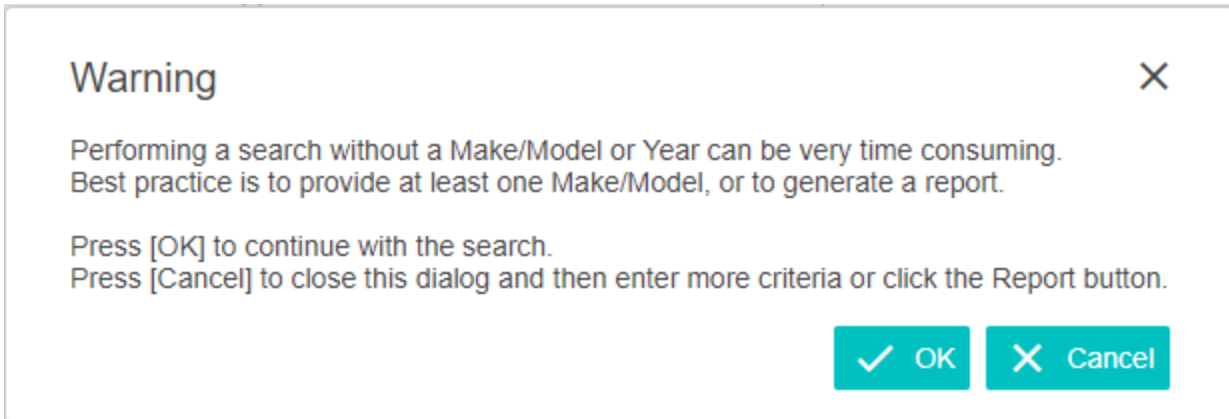
Criterion can be individually removed from a search box, or the Clear All link can be used to remove all selected criteria. To remove a criterion from a search box using the mouse, click the 'X' at the right of a criterion tile. Optionally when the Tab key on the keyboard is used to select a criterion, pressing the Enter key on the keyboard will remove the criterion.

Minimum Search Criteria

Though only one value within one search box is required for the Search button to be enabled, recommended practice is to provide as much detail as possible prior to running a search. When broad search criteria is provided (especially by leaving the Make/Model search box type blank) results can take much longer than expected to display.

When the use of broad search criteria is necessary, recommended practice is to use the Application Coverage Report by clicking the Report button within the Application Manager. For more information about using the Report button, see the **Application Coverage Report** topic.

When users attempt to run a search without populating the Make/Model search box, a dialog will display warning that the search may be time consuming, and reminding them of the Report button option. Below is an example of the warning dialog.



Search Box Synchronization

Because the Make/Model, Year, and Options Group Search Boxes are synchronized, search criteria can be entered into any one of the search boxes in any order, and the available search criteria will display in the dropdown based upon the criteria entered within the other search boxes. For example, when 'Tahoe' is selected for the Make/Model search box, 'Z71' is selected for the Sub Model search box, and the cursor is placed within the Year search box field, then the years displayed within the typeahead are only those that the Chevy Tahoe with the Z71 option were manufactured.

Make/Model 1 <input type="text" value="Chevrolet Tahoe X"/>	Year	Sub Model 2 <input type="text" value="Z71 X"/>
Enter Make/Model ☰	Enter Year 3	Enter Sub Model
	2000 2003 2004 2005 2006 2008 2009 2016 2017	

Note: To remove criteria that is displayed within a search box, click the 'X' to the right of the value, or use the Clear All link to clear all criteria within all search boxes.

Using Make/Model Search Box

The Make/Model search box allows users to add one or more make and/or model criteria to their search. This section addresses functionality that is specific to the Make/Model search box. For more information on the general functionality found within the search boxes, see the **Using Search Boxes** topic within this guide.

Within the Make/Model Search Box, the typeahead field can be used to search for make and/or model criteria. The dropdown list will display the criteria options alphabetically by Make, and then by Model. Models are always displayed below their respective Make. For example, if 'a4' is typed into the field, then the dropdown list displays the Audi, Austin, Checker, and Mercedes-Benz makes alphabetically, and the Models that match the information typed into the field (a4) will be listed alphabetically below their respective makes (as shown below).


The image shows a screenshot of a search box interface. At the top, there is a text input field labeled "Make/Model" containing the text "a4". To the right of the input field is a small icon consisting of three horizontal lines and a square. Below the input field, a dropdown list is displayed, showing the following items:

- Audi
 - ├ A4
 - ├ A4 Quattro
 - └ A4 allroad
- Austin
 - └ A40
- Checker
 - └ A4
- Mercedes-Benz
 - ├ A45
 - └ A45 AMG


If the text typed in matches a make only, then all Models below that make will be displayed. For example, if 'maz' is typed into the field, then the dropdown list will display the Mazda make with all its available Models listed below.

The image shows a screenshot of a software interface. At the top, there is a text input field labeled 'Make/Model'. Below this field, the text 'maz' is entered. To the right of the input field is a small icon consisting of three squares. Below the input field, a dropdown list is open, displaying the word 'Mazda' at the top. Underneath 'Mazda', there is a list of models, each preceded by a small horizontal line and a vertical bar (┆). The models listed are: 1200, 1500, 1800, 2, 3, 3 Sport, 323, 5, 6, 616, 618, 626, 808, 929, B1600, B1800, B2000, B2200, B2300, B2500, B2600, and B3000. A vertical scrollbar is visible on the right side of the dropdown list, indicating that the list can be scrolled.


If the text typed in matches a model only, then the make for the matching model is displayed with only the model that matches the search string. For example, if 'fox' is typed into the field, then the dropdown list will display the Audi and Volkswagen makes with the 'Fox' Models listed below.

Make/Model	
fox	
Audi └ Fox Volkswagen └ Fox	

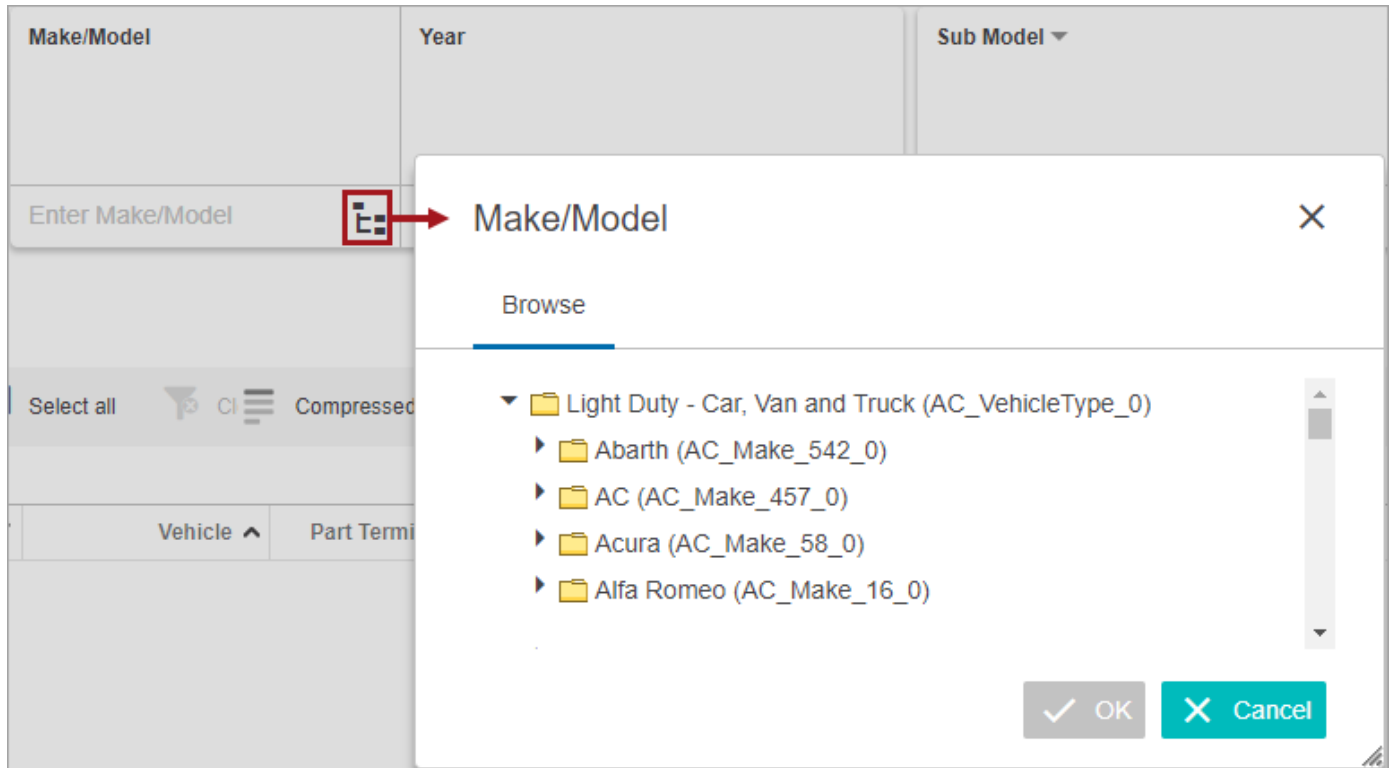
If the text typed in matches the make AND Model, then the matching makes and/or Models will display within the dropdown for selection. For example, if 'opel' is typed into the field, then the dropdown list will display the Buick make with the Opel model beneath, and then the Opel make will display with all of its available Models below.

Make/Model	
opel	
Buick └ Opel Opel └ 1900 └ Caravan └ Deluxe └ Kadett └ Kapitan └ Manta └ Olympia └ Olympia Rekord └ Opel └ Rallye	

To search for a make within a specific Model, two or more characters can be typed into the field to narrow down the Make, then using the keyboard space bar allows for the following text to conduct a 'contains' search for Models. For example, to display all the Audi models containing Quattro, then the following can be entered into the Make/Model field: 'aud [space bar] qua.'

Make/Model
Aud qua 
Audi
┆ 100 Quattro
┆ 200 Quattro
┆ 4000 Quattro
┆ 5000 Quattro
┆ 80 Quattro
┆ 90 Quattro
┆ A3 Quattro
┆ A4 Quattro
┆ A5 Quattro
┆ A6 Quattro
┆ A7 Quattro
┆ A8 Quattro
┆ Allroad Quattro
┆ Coupe Quattro
┆ Q2 Quattro
┆ Q3 Quattro
┆ Quattro
┆ TT Quattro
┆ V8 Quattro

Additionally, users can browse for the Make/Model criteria by clicking the browse icon located to the right of the Make/Model typeahead field. In the screenshot below, the Make/Model Browse dialog displays after the node picker browse icon is selected.



Important: Recommended practice is to provide at least one value for the Make/Model search box, otherwise longer than expected wait times can occur. When the use of broad search criteria is required, recommended practice is to use the Application Coverage Report by clicking the Report button within the Application Manager. For more information about running a search without the Make/Model search box populated, see the **Minimum Search Criteria** section of the **Using Search Boxes** topic.

Using Year Search Box

The Year search box allows users to add one or more year criteria to their search. This section addresses functionality that is specific to the Year search box. For more information on the general functionality found within the search boxes, see the **Using Search Boxes** topic within this guide.

When the cursor is placed within the typeahead field for the Year search box, and all other search boxes are blank, the dropdown will not display until at least one number is typed into the typeahead field. However, when one or more criteria are added to the Make/Model search box, and then the cursor is placed within the Year field, a dropdown list will display with only those years that pertain to the criteria within the other synchronized search boxes. For example, when the Make/Model search box is populated with the '4Runner' criterion, and the cursor is placed within the typeahead field of the Year search box, then a list of years from 1984-2018 displays. However, when the Sub Model search box is populated with the 'TRD Pro' criterion, and then the cursor is placed within the typeahead field of the Year search box, then a list of years from 2015-2018 displays (as shown below).

Make/Model Toyota 4Runner X	Year	Sub Model ▼ TRD Pro X
Enter Make/Model	Enter Year	Enter Sub Model
	2015 2016 2017 2018	

Note: The Make/Model search box must be populated with at least one criterion for the dropdown options for the Year and Options search boxes to automatically display when the cursor is placed within the search box field.

For more information about synchronized search boxes, see the **Search Box Synchronization** topic within the **Using Intelligent Search Interface** section of this guide.

Using Options Search Boxes

Attributes or references within STEP can be displayed as options within an Application Manager. The attribute and/or reference values can then be selected as search criteria.

The Options search boxes allow users to choose pre-configured attribute and/or reference type options, and the valid criteria, to their search. This section addresses functionality that is specific to the Options search boxes. For more information on the general functionality found within the search boxes, see the **Using Search Boxes** topic within this guide.

The Options search boxes can be configured to allow users to select up to two attribute and/or reference type options. The attribute and/or reference type option dropdown is located in the top left corner of the Options search box. Clicking the down arrow to the right of the Options search box label, or clicking the label itself will display the available options within the dropdown. Using the Options search box dropdown, users can choose criteria related to different search options (i.e., Sub Model, Region, Engine, Fuel Type). In the example below, two Options search boxes have been configured. The Options search box on the left displays the option 'Sub Model' by default, and the one on the right displays the option 'Region' by default. Clicking the option label displays a dropdown of available options for each Options search box (as shown below).

Make/Model	Year	Sub Model ▾ Sub Model Region Engine Fuel Type	Region ▾ Transmission Speeds Region Drive Type	Part Type
Enter Make/Model	Enter Year	Enter Sub Model	Enter Region	Enter Part Type

A Search Panel can have up to two Options search boxes. When two Options search box types are configured, an AND / OR toggle button displays between the two boxes. This toggle button allows users to determine if the search criteria within the two Options search box types should be inclusive or exclusive of each other. The AND / OR toggle button only applies to the criteria within the two Options search box types. It does not affect the inclusion of the criteria within the other search box types (i.e., Make/Model, Year, Part Type).

When the toggle button displays 'AND,' then the search results only display a match for the criteria provided within Options search box 1 **and** Options search box 2. However, when the toggle button displays 'OR,' then the search results will display a match for the criteria provided within Options search box 1 **or** Options search box 2.

The Options search box typeahead field results are dependent upon selections made in the Make/Model and Year search boxes. In the example below, the Toyota 4Runner is added to the Make/Model search box, and the 2015 year is added to the Year search box, allowing the Sub Model search box dropdown to only display options related to the 2015 4Runner. When the Sub Model option criteria is selected, then the dropdown displays only those sub models available for the 2015 4Runner.

Make/Model Toyota 4Runner ×	Year 2015 ×	Sub Model ▾
Enter Make/Model	Enter Year	Enter Sub Model
		Limited SR5 TRD Pro Trail

Adding additional makes or models to the Make/Model search box will expand the Options search box dropdown results. In the example below, the Toyota and Jeep makes have been added, and result in all Toyota, Jeep, and 4Runner sub models displaying within the Sub Model Options search box dropdown. (Though the long list of available sub models cannot be seen in the screenshot, the display of a mixture of available sub models is apparent with the inclusion of the Jeep Rubicon, Toyota Hybrid SE, and Toyota 4Runner TRD Pro.)

Make/Model Toyota 4Runner × Toyota × Jeep ×	Year 2015 ×	Sub Model ▾
Enter Make/Model	Enter Year	Enter Sub Model
		L One Hybrid SE Technology TRD Pro Unlimited Rubicon VX XLE Touring

By changing the Option selection from Sub Model to Engine, the results dropdown only displays the one engine available for the 2015 4Runner.

Make/Model Toyota 4Runner ×	Year 2015 ×	Engine ▼
Enter Make/Model	Enter Year	Enter Engine
		4.0L L6, 3956CC, 241CID

Additionally, when two Options search boxes are enabled, allowing for the 'AND / OR' toggle button to be set to 'AND,' the dropdown for the options cards will also be dependent upon selections made between the two Options search boxes. In the example below, the Region criterion has been set to Canada, so the Sub Model search box displays only those sub model options available for the 2015 4Runner in Canada. Notice from the previous example, the TRD Pro was available for selection, until the Canada region criterion was added, because the 2015 TRD Pro sub model is not available in Canada.

Make/Model Toyota 4Runner ×	Year 2015 ×	Sub Model ▼	Region Canada (CAN) ×
Enter Make/Model	Enter Year	Enter Sub Model	Enter Region
		Limited SR5 Trail	

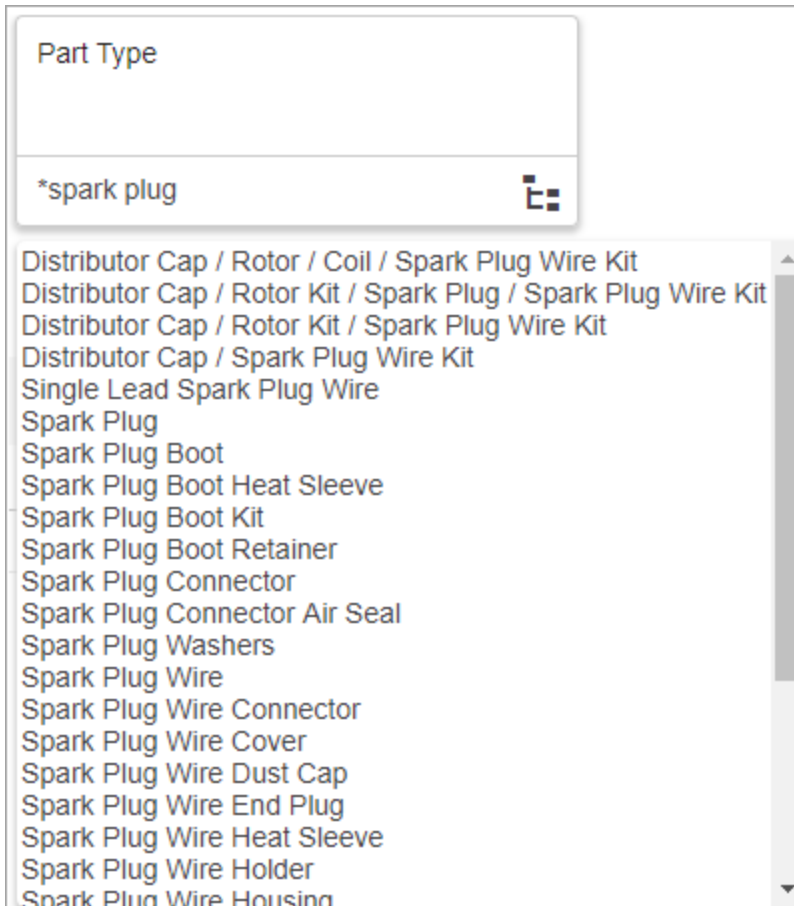
Note: The Options search boxes will not automatically display the suggested dropdown options until at least one value is added to the Make/Model search box.

Optionally, the Report button can be used to export these results to an Excel file. For more information, see the **Using Application Coverage Report** section of this guide.

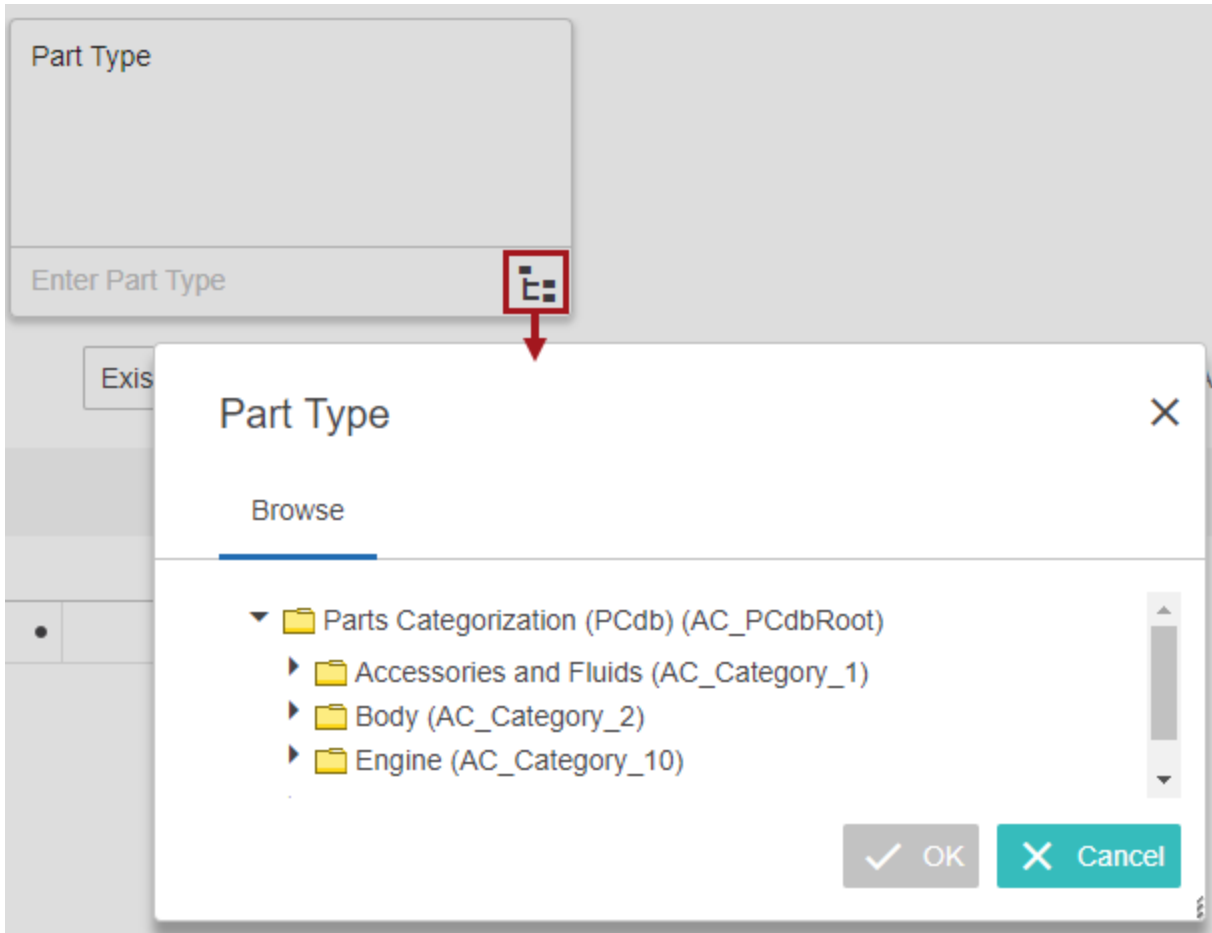
Using Part Type Search Box

The Part Type Search Box allows users to add one or more part type criteria to their search. This section addresses functionality that is specific to the Part type search box. For more information on the general functionality found within the search boxes, see the **Using Search Boxes** topic.

Within the Part Type Search Box, the typeahead field can be used to search for part type criteria, and the use of an asterisk will allow for a wildcard search. Only part type Names are searched, not IDs. The dropdown list displays criteria options alphabetically. For example, if '*spark plug' is typed into the field, then the dropdown list displays the any part type name containing 'spark plug' alphabetically (as shown below).



Additionally, users can browse for the Part Type criteria by clicking the node picker browse icon located to the right of the Make/Model typeahead field. In the screenshot below, the Part Type Browse dialog displays after the node picker browse icon is selected.



Important: When searching on Part Type, the part type link must be established on an application or part in order for the applications to be displayed in the search results table. If the part type link is on the part and the application does not have a link, then it will inherit from the part. If the application has a link, then it will override the link that's on the part. For example, if a user is searching for the Part Type Spark Plugs, and an application is not linked to Spark Plugs (in AutoCare, the Product to Classification Link Type ID = AC_ProductToPartTerminology), then that application will not display in the search results table.

Saved Search Tool

Users can quickly populate previously saved search criteria by selecting the name of a saved search from the saved search dropdown. Saved searches can consist of many different combinations of intelligent search interface selections, including criteria within one or more search boxes, existing and/or missing application coverage selection, and/or brand selection.

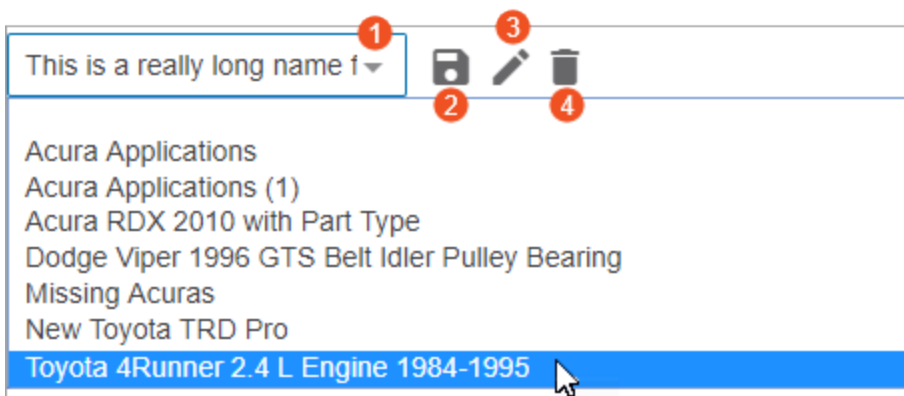
Once one or more search criteria selections are made, users can save the selections with a custom name. Using previously saved searches can be helpful when the same search criteria is needed repeatedly, or when using a basic set of saved search criteria aids users in more quickly building complex searches.

Saved searches are restricted by User login, Web UI, Vehicle Type Search Panel, and Screen ID. In other words, users will only see the saved searches created with their login for the specific Web UI being used. Additionally, this allows users to create a succinct list of saved searches that is relative to each Vehicle Type Search Panel.

Note: The saved search tool does not save search results, but rather the search box criteria, existing and/or missing application dropdown selection, and/or brand selection. Once a saved search is selected from the dropdown, the Search and/or Report buttons can be used to view the search results.

The saved search tool is located at the bottom left of the intelligent search interface (below the first search box), and consists of the following:

1. **Saved search dropdown:** Allows users to select the name of a previously saved search from an alphabetized list of saved search names.
2. **Save icon:** Allows users to save the intelligent search interface selections (i.e., criteria within search boxes, existing or missing application coverage selection, brand selection) by either creating a new saved search or overwriting an existing saved search.
3. **Rename icon:** Allows users to edit the name of the selected saved search.
4. **Delete icon:** Allows users to delete the selected saved search.



Saving a Search

The Save icon is disabled until at least one criterion is added, removed, or edited. The steps involved with saving search criteria are different when a new saved search is being created, versus an existing search being overwritten. This section will first address the steps involved with creating a new saved search, and then overwriting an existing saved search.

Creating a Saved Search

Once one or more criteria have been added to a search box, and the Save icon is enabled, the following steps can be used to create a new saved search.

1. Click the Save icon, and the 'Save search' dialog will display as shown below.

Note: As long as a saved search has not been selected (thus the saved search dropdown is blank), then the Save search dialog will display as shown above.

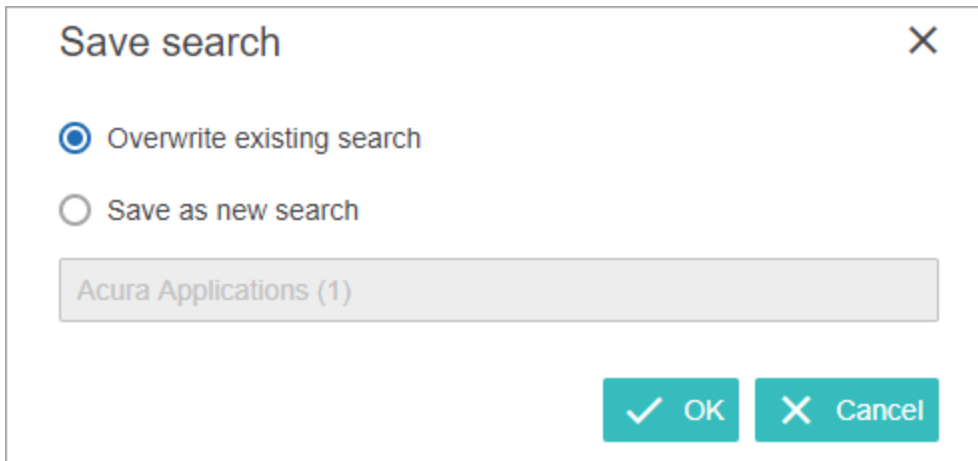
2. Within the Name parameter, type a name that best describes the saved search. The name can be created using any alphanumeric combination, but should be easy to identify for future use in the saved search dropdown.
3. Once at least one character is entered into the Name parameter, the OK button is enabled. Click the OK button to finish creating the saved search. The Save search dialog will close and the newly created saved search will display in the saved search dropdown.

Note: The Cancel button can be clicked at any time. Doing so will close the dialog without saving any changes.

Overwriting an Existing Search

When a saved search is selected in the dropdown, one or more criteria have been added or removed from a search box, and the Save icon is enabled, the following steps can be used to overwrite a saved search.

1. Click the Save icon, and the 'Save search' dialog will display as shown below.



The 'Save search' dialog box features a title bar with a close button (X) in the top right corner. Below the title, there are two radio button options: 'Overwrite existing search' (which is selected) and 'Save as new search'. A text input field below these options contains the text 'Acura Applications (1)'. At the bottom right, there are two buttons: a teal 'OK' button with a checkmark icon and a teal 'Cancel' button with an X icon.

Note: As long as a saved search has been selected (thus the saved search dropdown is populated), then the Save search dialog will display as shown above.

Overwrite existing search: By default, the 'Overwrite existing search' option is selected. Choosing this option will overwrite the currently selected saved search with the newly edited criteria.

Save as new search: Selecting the 'Save as new search' option will enable the Name parameter. By default, the Name parameter is populated with the currently selected saved search name appended with '(1)'. Populate the Name parameter in a way that is easy to identify for future use in the saved search dropdown.

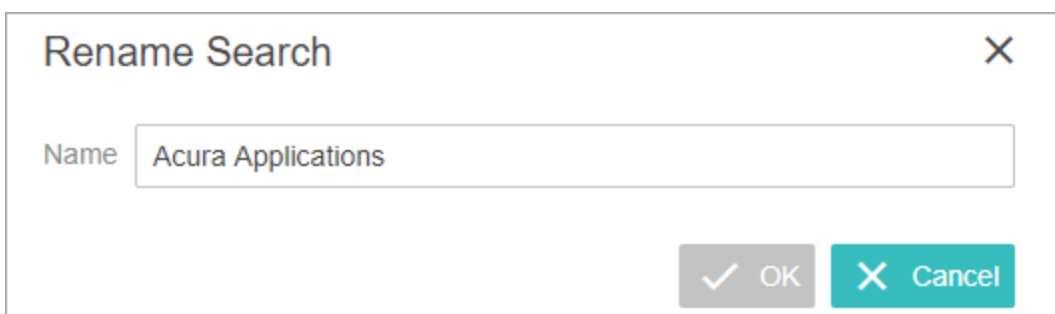
2. Once the desired option is selected and/or populated, click the OK button. The Save search dialog will close and the newly created saved search will display in the saved search dropdown.

Note: The Cancel button can be clicked at any time. Doing so will close the dialog without saving any changes.

Renaming a Saved Search

Once a previously saved search name is selected from the dropdown and the Rename icon is enabled, the following steps can be used to edit the name of a saved search.

1. Click the Rename icon, and the Rename Search dialog will display the Name parameter populated with the current name of the saved search. In the example below, the Acura Applications Saved search was used.



The 'Rename Search' dialog box has a title bar with a close button (X) in the top right corner. Below the title, there is a label 'Name' followed by a text input field containing the text 'Acura Applications'. At the bottom right, there are two buttons: a greyed-out 'OK' button with a checkmark icon and a teal 'Cancel' button with an X icon.

2. Populate the Name parameter in a way that is easy to identify for future use in the saved search dropdown.
3. Once the desired Name is populated, click the OK button to save the change.

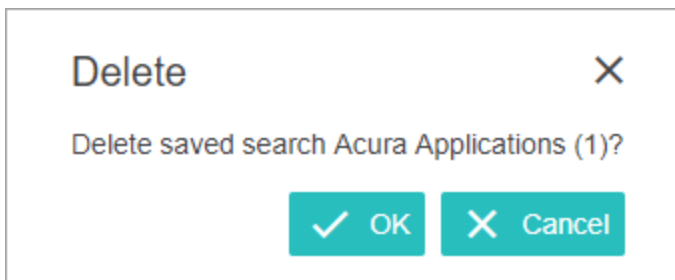
Note: The Cancel button can be clicked at any time. Doing so will close the dialog without saving any changes.

Deleting a Saved Search

Once a previously saved search name is selected from the dropdown, and the Delete icon is enabled, the following steps can be used to delete the saved search.

Important: Once the saved search is deleted, it will no longer display for selection in the dropdown, and cannot be recovered.

1. Click the Delete icon, and the Delete dialog will display confirming the Name of the saved search to be deleted (as shown below).



2. Click the OK button, the dialog will close, and the saved search will no longer display in the dropdown list.

Note: The Cancel button can be clicked at any time. Doing so will close the dialog without saving any changes.

Missing Application Coverage Functionality

Application coverage is determined by both the Condition and Part Type values. The existing and missing applications dropdown allows users to determine if their search should or should not contain existing and/or missing applications. The following three options are available for the existing and missing applications dropdown:

- Existing Applications Only
- Existing and Missing Applications
- Missing Applications Only

The screenshot shows a search interface with two input fields: 'Make/Model' and 'Year'. Below these fields are two text boxes labeled 'Enter Make/Model' and 'Enter Year'. To the right of the 'Enter Make/Model' box is a small icon of a grid. Below the input fields is a dropdown menu with three options: 'Existing Applications Only', 'Existing and Missing Applications', and 'Missing Applications Only'. The 'Missing Applications Only' option is highlighted in blue, and a mouse cursor is pointing at it. To the left of the dropdown menu are three icons: a folder, a pencil, and a trash can.

To search or report on missing applications, users can select either the 'Missing Applications Only' or 'Existing and Missing Applications' options from the dropdown located below the search boxes, and to the left of the Report button.

Important: Though it is not required, it is strongly recommended to include criteria for at least one Vehicle and one Part Type before conducting a missing application search. Failure to do so may result in very long search times, timeout, or inconsistent results.

When users select the 'Missing Applications Only' option, and the Search or Report button is clicked, the following occurs:

1. STEP searches for and displays existing combinations of the vehicles and part types defined in the search that do not have applications.
2. If the Part Type search box is left blank, then the results table will display a new row for each combination of the vehicle(s) valid for the search criteria and the part types valid for that vehicle (and/or any additional search criteria).
3. If applications exist for the vehicles and part types defined in the search, then STEP checks if the applications cover all the different configurations of the vehicle. This is found by using attributes that are populated and have a validation path. The additional configurations of the vehicle(s) are displayed as a new row in the results table.

For more information about validation paths, see the **Automotive Validation Path Functionality** topic within this guide.

Note: By default, search results are limited to 5,000 rows maximum. If a search needs to return more than the maximum results allowed, then an Application Coverage Report can be run to view the full results. For more information, see the **Application Coverage Report** section.

When the 'Existing and Missing Applications' option is selected, in addition to the actions described above, STEP also searches for and displays existing combinations of the vehicles and part types defined in the search.

In other words, when a search for missing applications is run, by selecting either the 'Missing Applications Only' or 'Existing and Missing Applications' option, the populated values on the existing applications are evaluated to ensure that every option / criteria that is populated with a value (and included in the attribute group identified in the 'Application Condition Header - Group') is considered for all valid configurations of the selected vehicle(s). Each combination that is considered valid, and does not have an application, is displayed as a single row within both the Application Manager results table, and the Application Coverage Report.

Note: A blank / null value for any option / criteria equates to that option being valid for ALL values of that option.

For example, consider a vehicle that has two Sub Models, each of which is available with Gas or Diesel fuel types, and each fuel type has a valid configuration with each Sub Model; totaling four valid configurations of this vehicle. Further, assume that only Sub Model and Fuel Type are populated on the records we are working with (e.g., no other options have data populated in any of the existing records for the selected vehicle and part type), and that Sub Model and Fuel Type are both included in the attribute group identified in the application condition header. Some of the scenarios based on data population and the expected missing coverage outcome are as follows:

Existing Record	Missing Application Record
One record where Sub Model and Fuel Type are null	None, as the existing record is understood to work for all Sub Model and Fuel Type options
One record where Sub Model is populated with Sub Model 1 and Fuel Type is null	One record: Sub Model 2 + null Fuel Type
One record where Fuel Type is populated with Gas and Sub Model is null	One record: null Sub Model + Diesel
One record where Sub Model is populated with Sub Model 1 and Fuel Type is Gas	Three records: Sub Model 2 + Diesel, Sub Model 1 + Diesel, Sub Model 2 + Gas
Two records: Sub Model 1 + null Fuel Type, Sub Model 2 + Gas	One record: Sub Model 2 + Diesel
No records	One record with both Sub Model and Fuel Type as null

Improving Part Type Accuracy

As previously mentioned, the application data model (Automotive - Application Model) includes a relationship between conditions on applications and different configurations of vehicles. This is possible due to the automotive validation path functionality. However, sometimes the conditions on specific part types need to be considered to ensure that missing applications display more accurately.

For example, when searching for missing applications for spark plugs on a 2013 Audi A3, the result table will list both Gas and Diesel engines, since Diesel engines do not use spark plugs, this is inaccurate. However, when the validation path functionality is implemented in conjunction with the 'Missing Application Conditions' attribute and the 'Check path for missing application' business condition, the search results will display more accurately because the application coverage will consider both the Condition and Part Type values.

To implement this improvement, the automotive validation path functionality and Check Path for Missing Application Business Condition must be implemented. For more information, see the **Automotive Validation Path Functionality** and the **Business Condition: Check Path for Missing Application** topics within this guide.

Application Coverage Report

The Application Coverage Report can display missing and/or existing application data in an Excel spreadsheet (XLSX file type). The results displayed within an Application Coverage Report are based on the search criteria selected within an Application Manager screen. Since the Application Coverage Report is built to work with the Application Manager's intelligent search interface, it is extremely configurable, and provides users on-demand flexibility to choose the report criteria that best meets their needs.

Before an Application Coverage Report can be run, the desired search criteria must be entered into the Application Manager Search Panel, either manually or by selecting from the list of saved searches. Once criteria have been selected, clicking the 'Report' button allows users to export the Application Coverage Report.

Within an Application Coverage Report, each application and/or missing application will display as its own row. Missing application rows are prepopulated with the condition(s) that are currently missing coverage. Because the missing coverage is visible within the Application Manager results table, it becomes easy for the user to apply the part by entering the part number necessary for the vehicle and its conditions.

Prerequisites

Before attempting to understand how the Application Coverage Report is used, it is recommended to first become familiar with the **Application Manager Overview** and **Using Application Manager Intelligent Search Interface** topics within this guide.

Application Coverage Report Results

The Web UI designer can be used to easily add and/or remove the display of columns within the Application Manager results table. Within an Application Coverage Report, when the following column headings are configured within the Application Manager results table, they will display within the Application Coverage Report:

- Application Assembly Value
- Application Competitor or OE Part Numbers
- Application Condition Header - Group
- Application Condition Header - Individual
- Application Part Type Value
- Application Part Type Title Header
- Application Part Value
- Application Set Assembly
- Application Set Part

Along with the columns configured to display within the results table, several ID columns (that may not display within the results table) will display to the right of their respective Name columns within the report. Below is an example of the attribute and/or classification references that will display an ID column:

- NAPA_ApplicationToEngine
- AC_ACESApplicationToEngineBase
- AC_ACESApplicationToTransmissionBase

- LOVs using an ID (i.e., NAPA_BodyStyleLOV, NAPA_EngineDesignatorLOV)
- Application Part Type Title Header
- Application Set Assembly
- Application Set Part

Additionally, attributes that are both valid for the search criteria provided, and configured within the 'Application Condition Header - Group' component, will display within the Application Coverage Report. For ease of use, each valid attribute name is listed within a column header of the report. For more information on the Application Condition Header - Group' component, see the **Application Condition Headers** topic within this guide.

Important: Only options that contain a value will be exported. When an option is exported, its name will display as a column heading. Even when the following options are added to the Node List Properties Headers, they will not be exported because they are not handled: Application comments (AC Comment), ACES Application Qualifiers, Comments, NAPA Notes, and Application Asset References.

Running an Application Coverage Report

Users can choose to export existing applications and/or missing applications by selecting one of the following options in the dropdown to the left of the Report button:

- Existing Applications Only
- Existing and Missing Applications
- Missing Applications Only

When the 'Existing and Missing Applications' or 'Missing Applications Only' options are selected, STEP performs specific actions known as missing application coverage functionality. For more information about missing applications, see the **Missing Application Coverage Functionality** topic within this guide.

After at least one criterion has been added to a search box, the Report button will enable. Once enabled, the Report button can be clicked before or after clicking the Search button. In other words, users are not required to use the Search button prior to the Report button. In fact, for searches that are time consuming or vague, it is recommended to use the Report button instead of the Search.

Once the Report button is clicked, a background process is initiated, and a notification displays informing users of the linked background process ID, and that the process has started. In the example below, the background process notification displays with the ID link displayed as 'BGP_101869.'



Within the notification, clicking the 'X' will close the dialog, whereas clicking the ID link will close the dialog and display the Background Process Details (as shown below).

Background Process Details

ID BGP_101869

Started By USERM

Description Run application coverage report.

Template ID AppCoverageReportSrvc

Status ✓ Succeeded
[ApplicationCoverageReport-2018-05-08.xlsx](#)

Started 5/8/18 1:13:15 PM

Finished 5/8/18 1:13:16 PM

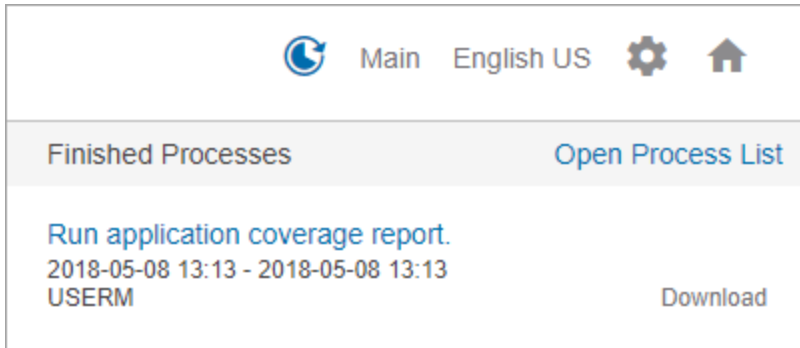
Elapsed 2 s

Select all Export

ID	Type	Text
<input type="checkbox"/> 10	Info	Application Coverage Report Service - start. (Tue May 08 13:13:15 EDT 2018)
<input type="checkbox"/> 20	Info	Created Application Search Specification. (Tue May 08 13:13:15 EDT 2018)
<input type="checkbox"/> 40	Info	Finding applications. (Tue May 08 13:13:15 EDT 2018)
<input type="checkbox"/> 50	Info	Found 2 applications. (Tue May 08 13:13:16 EDT 2018)
<input type="checkbox"/> 70	Info	Generating Application Coverage Report. (Tue May 08 13:13:16 EDT 2018)
<input type="checkbox"/> 80	Info	Created workbook. (Tue May 08 13:13:16 EDT 2018)
<input type="checkbox"/> 120	Info	Populated data to workbook. (Tue May 08 13:13:16 EDT 2018)
<input type="checkbox"/> 130	Info	Start writing data to excel file (Tue May 08 13:13:16 EDT 2018)
		Excel file written (Tue May 08

Once the Background Process Details screen Status parameter displays as 'Succeeded,' the download link for the Application Coverage Report will display. In the example above, the link displays as 'ApplicationCoverageReport-2018-05-08.xlsx.'

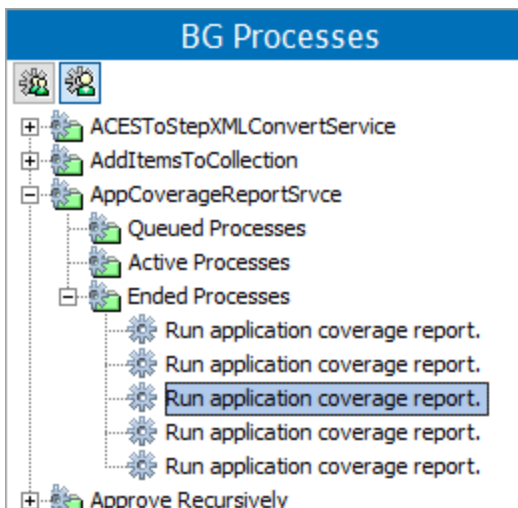
If the Background Process Notification Component is enabled, then users can easily track the progress of the report in the right side panel. In the example below, the 'Run application coverage report' link displays within the 'Finished Processes' of the side panel.



From within the notification of the Background Process side panel, users can click the background process link (Run application coverage report), and the Background Process Details screen will display. Users can also click the 'Download' link to download the .XLSX Application Coverage Report. For more information on the Background Process Notification Component, see the **Background Process Notification Component** topic within **STEP Online Help**.

Additionally, the report can be downloaded from the workbench. To download the report from the workbench, go to the BG Processes tab > AppCoverageReprtSrvce > Ended Processes. Locate the desired process and click the save icon in the lower right corner of the workbench.

In the example below, the third of six background processes for the Application Coverage reports is selected.



Adding the Application Coverage Report to a Background Processes Screen

Optionally, the progress of a download for the Application Coverage Report can be configured to display within a Background Processes Screen. When configured, users can navigate to a Background Processes screen to view the following: TemplateID, ID, Started By, Progress, Start Date (and time), Finish Date (and time), Results download (as shown below).

Background Processes List							
Clear filter							
TemplateID	ID	Started By	Progress	Start Date	Finish Date	Result	
AppCoverageReportSrvc	BGP_101833	USERM	✓ Succeeded	5/7/18 5:03 PM	5/7/18 5:03 PM		→ ⬇

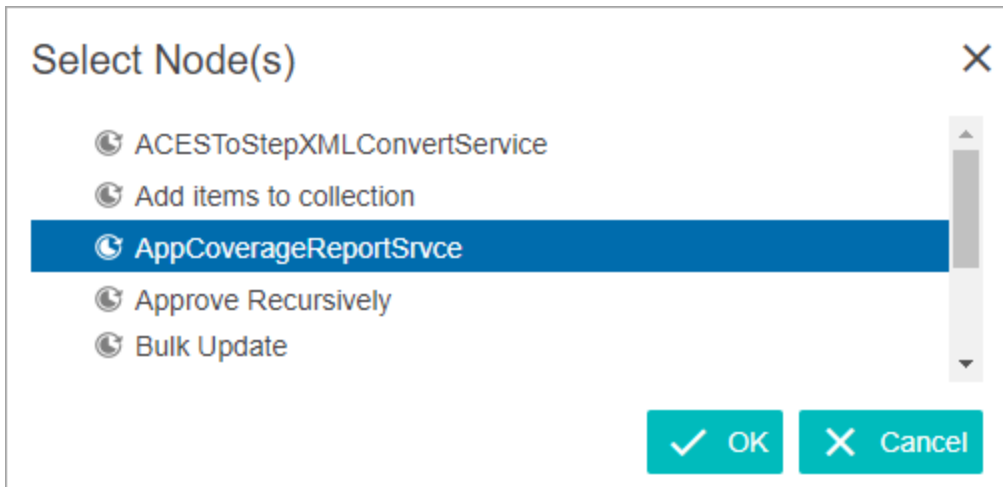
From this table, users can optionally click on the BGP ID and the Background Process Details screen will display, or click the Results download to download the report.

Below are the steps to add the Application Coverage Report to a Background Processes Screen.

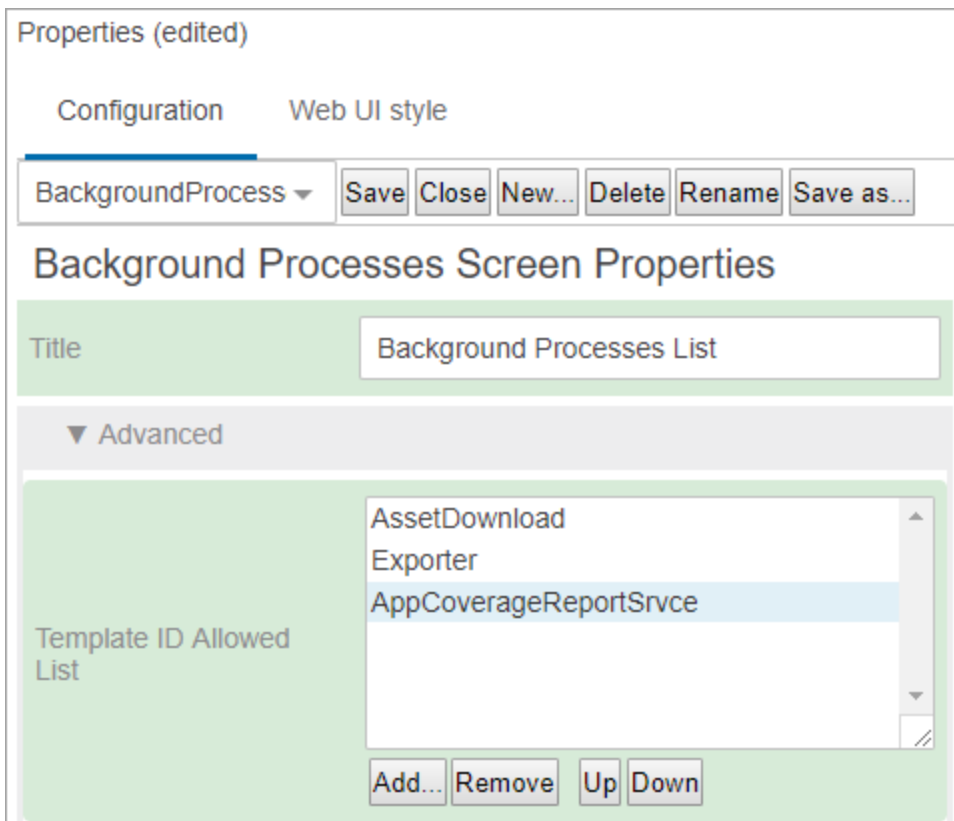
1. Within a Web UI, go to a Background Processes Screen, access the Designer, and the Background Processes Screen Properties will display as shown below.

The screenshot shows the 'Background Processes Screen Properties' dialog. At the top, there are tabs for 'Configuration' and 'Web UI style'. Below the tabs, there is a dropdown menu for 'BackgroundProcess' and several action buttons: 'Save', 'Close', 'New...', 'Delete', 'Rename', and 'Save as...'. The main area is titled 'Background Processes Screen Properties'. Under the 'Title' field, the value is 'Background Processes List'. Below this is an 'Advanced' section with a dropdown arrow. Underneath, there is a 'Template ID Allowed List' which is currently empty. A list box shows 'AssetDownload' and 'Exporter'. At the bottom of the list box are buttons for 'Add...', 'Remove', 'Up', and 'Down'.

2. Click the **Add** button below the 'Template ID Allowed List' parameter, and the Select Nodes list will display.
3. Select the **AppCoverageReportSrvc** node, as shown below.



- Click the **OK** button, the dialog will close, and the 'Template ID Allowed List' parameter will display with the newly added AppCoverageReportSrvc (as shown below).



- Click the **Save** and **Close** buttons on the Designer.

For more information about the Background Process List Screen, see the **Background Process List Screen** topic within **STEP Online Help**.

Results Table and Toolbar

The bottom half of the Application Manager screen offers features related to the results table and its toolbar. The top half of the screen is dedicated to the intelligent search interface. For more information about the intelligent search interface, see the **Intelligent Search Interface** topic.

This section provides an overview of the results table and toolbar and addresses the following:

- Results Table Default Columns
- Controlling Display of Conditions in Application Manager
- Creating Applications

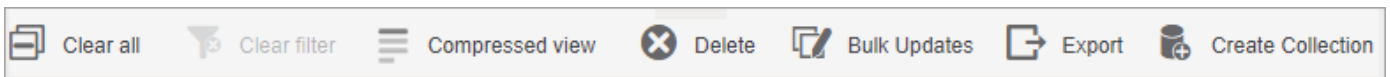
Results Table and Toolbar Overview

The results table and toolbar of an Application Manager screen are composed of the following:

The screenshot shows the results table and toolbar of an Application Manager screen. The toolbar (1) includes icons for Clear all, Clear filter, Compressed view, Delete, Bulk Updates, Export, and Create Collection. The table (2) has columns for Vehicle, Part Terminology, Part Number, ACES Fuel Type, ACES Position, Options, Notes, Qualifiers, and Assets. The table contains four rows of data. The first row is for a 2006 Acura CSX with a Spark Plug (Part Number 12345678). The second row is for a 2008 Acura CSX with an Air Bag Sensor (Part Number 034-VC30002, ACES Position Left, ACES Quantity: 1). The third row is for a 2008 Acura CSX with an A/C Drive Belt Idler Pulley Bearing (Part Number 034-VC36004, ACES Sub Model: Technology). The fourth row is for a 2009 Acura CSX with an A/C Drive Belt Idler Pulley (Part Number 034-VC30002, ACES Quantity: 1). The table also shows a summary row with a total of 15 items (4). The table is scrollable, and the right side has a vertical toolbar (6) with icons for back, save, refresh, and forward. The table is also filtered by 'Notes' (5).

Vehicle	Part Terminology	Part Number	ACES Fuel Type	ACES Position	Options	Notes	Qualifiers	Assets
2006 Acura CSX	Spark Plug	12345678						
2008 Acura CSX	Air Bag Sensor	034-VC30002		Left	ACES Quantity: 1			
2008 Acura CSX	A/C Drive Belt Idler Pulley Bearing	034-VC36004	GAS		ACES Sub Model: Technology			
2009 Acura CSX	A/C Drive Belt Idler Pulley	034-VC30002			ACES Quantity: 1		12946 - with Air Bags 393 - 15 / 20 Press Fit Diameter 6 - #1 & #2 In take Inlets are Oval	
Number of items : 15								

1. Results table toolbar



Contains action buttons relative to the Application Manager results table. Action buttons can be configured using the Web UI Designer. For more information about action buttons, see the **Action Buttons** section of the **Web User Interfaces** guide within **STEP Online Help**.

- **Button labels:** When the 'Include Labels' parameter within the Node List Properties is disabled, the labels for each icon within the toolbar will not display. However, when a user hovers over an icon, help text will display.
- **Clear all / Select all:** The 'Clear all' button displays when one or more rows are selected using the 'row selection boxes.' When no rows are selected, the 'Select all' button will display. Clicking the 'Clear all' button will remove the selections within the 'row selection boxes.' Clicking this button will not clear the actual table results, but rather disables the selection of 'row selection boxes.' Clicking the 'Select all' button enables the 'row selection boxes' for each row within the results table.
- **Clear filter:** The 'Clear filter' button displays when one or more filters are applied to columns in the Results table. Clicking the 'Clear filter' button will remove any filter selections within the Results table columns but will not clear the actual table results. Within the filter dialog for each column is a 'Reset filter' button that can be used to reset the filter setting for each column. Additionally, the 'Clear All' link above the Results table (not to be confused with the 'Clear all' action button within the Results table toolbar) can be used to clear all filters and search criteria. For more information, see the **Clear All link** section of the **Intelligent Search Interface** topic.
- **Compressed / Normal view:** By default, the table results display using the 'Normal view.' Clicking the 'Compressed view' button will compress the values displayed in the Options column by removing the attribute labels and displaying only the attribute values. When the table displays as compressed, the 'Normal view' button will display within the toolbar instead of the 'Compressed view' button.
- **Delete:** For the Delete button to display, one or more rows must be selected by enabling 'row selection boxes.' No warning nor confirmation dialog will display, and the application (and its data) will be permanently deleted.
- **Bulk Updates:** For the Bulk Updates button to display, one or more rows must be selected by enabling the 'row selection boxes.' The Bulk Updates action button is not configured automatically by Easy Setup Actions, and must be configured for use with Automotive Business Rule Plugins. For more information, see the **Automotive Business Rule Plugins** section within the **Automotive Reference Guide**.
- **Export:** For the Export button to display, one or more rows must be selected by enabling 'row selection boxes.' The Export action button is not configured automatically by Easy Setup Actions. To properly function, a Web UI Designer User must add an export configuration to the 'Narrowed Export Configurations' parameter of the Export action button. If the Export button has not yet been configured, and the button is selected, a 'No export configurations found' warning message will display (as shown below).



- **Create Collection:** For the Create Collection button to display, one or more rows must be selected by enabling 'row selection boxes.' Clicking the Create Collection button displays a 'Create New Collection' dialog where users can provide a Name of the new collection, and select a Collection Group to store the collection in. For more information, see the **Collections** topic within the **Getting Started Section** of the **STEP Online Help**.

2. Flip table icon



Allows users to change or flip the orientation of the table.

3. Row selection boxes



Allows for selection of one or more rows within the table. Selecting at least one row can change the action buttons that display within the toolbar.

4. Number of items

Number of items : 15

List the total number of rows available for display within the results table.

Note: By default, search results are limited to 5,000 rows maximum. If a search needs to return more than the maximum results allowed, then an Application Coverage Report can be run to view the full results. For more information, see the **Application Coverage Report** section.

5. Column Headers

Vehicle •	Part Terminology •	Part Number •	ACES Fuel Type •	ACES Position •	Options •	Notes •	Qualifiers •	Assets •
-----------	--------------------	---------------	------------------	-----------------	-----------	---------	--------------	----------

Displays the column headings as configured within the Headers parameter for the Node List Properties Child Component of the Application Manager.

- **Sorting and filtering columns:** Each column displays a black dot to the right of the Heading. Clicking the black dot will display the sort and filter options for the column. By default, search results are sorted by the Vehicle column. When a sort selection is made an up or down arrow displays in place of the black dot, indicating the search is respectively ascending or descending. When a filter selection is made, a black filter displays in place of the black dot. This makes it easy for users to identify any sorting and/or filtering applied to the results table. When one or more filters are applied, the 'Clear filter' button within the toolbar can be used to clear all the filters. Otherwise, any sort and filter selections made will remain (even after conducting a new search) until the Application Manager page is exited.

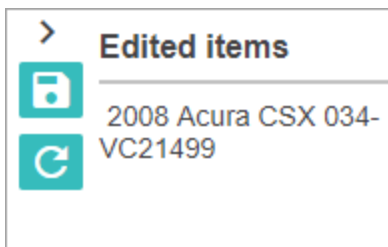
By default, the following components are configured within the Headers parameter for the Node List Properties Child Component of the Application Manager:

- Application Set Assembly (labeled as 'Vehicle')
- Application Part Type Title Header (labeled as 'Part Terminology')
- Application Set Part (labeled as 'Part Number')
- Application Condition Header - Individual (labeled as 'Additional Options,' but displays the relative attribute name.) In the example above, ACES Fuel Type and ACES Position are displayed because the attributes are linked to a part type with the display condition set to 'true.' For more information on how to display linked attributes, see the **Business Action : Set condition Links on Part Types** topic within the **Automotive Business Rule Plugins** section.
- Application Condition Header - Group (labeled as 'Options')
- Application Comment (labeled as 'Notes')
- Application Asset Reference (labeled as 'Assets')

For information on adding columns to the results table, see the **Adding Additional Headers to Application Screens** topic within the **Automotive Quick Start Guide**.

Note: When the Easy Setup actions for the AutoCare solution are completed, the 'ACES Application Qualifiers' component is automatically configured (labeled as 'Qualifiers') in addition to the components listed above.

6. Edited items panel



Contains an expand / collapse arrow, Save , and Reset buttons. When the panel is expanded and changes are made to the results table, the changes are displayed as a list in the panel. In the example above, it is easy to see that the 2008 Acura application has been edited. By default, the 'Use Immediate Save' parameter within

the Node List Properties is disabled. When enabled, the Save button will not display, and the list of edited items will briefly display.

7. Edit icons



Clicking on any one of the Edit icons within the results table will display a Value editor dialog where the value ID and/or Name are displayed. From a Value editor dialog, users can add and/or remove References.

8. Hyperlinks

A/C Drive
Belt Idler
Pulley
Bearing

When text within the results table displays as a hyperlink, users are able to click the text and a screen specific to that text will display. The most common use of this is with the Part Terminology (classifications) and Vehicle columns.

- **Classification Hyperlinks:** For classification hyperlinks to display within a results table, a Node Details component should be used to create a 'Classification Details' screen that is configured to display details about the classification object types. To make sure this screen displays when a classification is selected within the results table, be sure to add the necessary Mappings to the Web UI MAIN screen. For more information, see the **Mappings** topic within the **Main Properties Overview** of the **STEP Online Help**.
- **Part Number Hyperlinks:** For Part Number hyperlinks to display within a results table, the 'Enable Link' parameter must be enabled within the 'Application Set Part' component. By default, this is enabled. When a user clicks a part number hyperlink, an Application Editor screen will display. Values within cells can be edited by double-clicking in a cell. Once a cell is active, a typeahead or Node Picker can be used to select a new value.
- **Vehicle Hyperlinks:** For Vehicle hyperlinks to display within a results table, the 'Enable Link' parameter must be enabled within the 'Application Set Assembly' component. By default, this is enabled. When a user clicks a vehicle hyperlink, an Application Editor screen will display. Values within cells can be edited by double-clicking in a cell. Once a cell is active, a typeahead or Node Picker can be used to select a new value.

9. Results table

Vehicle •	Part Terminology •	Part Number •	Options •	Notes •	Qualifiers •	Assets •
2008 Acura CSX	Air Bag Sensor	034-VC30001	ACES Quantity: 1	🔗		🔗
2008 Acura CSX	A/C Drive Belt Idler Pulley Bearing	034-VC21499	ACES Sub Model: Technology	🔗		🔗
2009 Acura CSX	A/C Drive Belt Idler Pulley	034-VC36004	ACES Quantity: 1	🔗	12946 - with Air Bags 393 - 15 / 20 Press Fit Diameter 6 - #1 & #2 In take Inlets are Oval	🔗

Note: The example above is from an AutoCare Application Manager. However, NAPA and TecDoc Application Managers use the same results table settings, but the AutoCare specific headers are removed. The vehicle names and part numbers shown above display as plain text because the 'Enable Link' parameter for the 'Application Set Assembly' and 'Application Set Part' components are disabled.

A Node List component is used to display the results of selections made within the intelligent search interface. This includes criteria selected within the Vehicle Type Search Panel, and selections made within the Existing and Missing Applications and Hierarchy Restriction dropdowns. For more information on the intelligent search interface, see the **Intelligent Search Interface** topic.

Once the Search button is clicked, valid results display within the results table. However, it is possible to click the Search button and have zero results display. For example, if the 'Existing Applications Only' dropdown option is selected, and the search criteria is limited to vehicles types that do not have any applications, then a 'No Result' dialog will display, and the results table will only display column headings. When results are displayed within the results table, users have the option to create, edit, or delete part application data.

For more information on column headers, and editing data using the different Header components, see the **Results Table Default Columns** topic.

Note: Each column label can be edited as needed by accessing the Application Manager Node List Properties Child Component Headers parameter.

Results Table Default Columns

As shown in the image below, by default, an Application Manager results table uses a Node List component to display:

- 'Vehicle' column as configured with the **Application Set Assembly Component**.
- 'Part Terminology' column as configured with the **Application Part Type Title Header Component**.
- 'Part Number' column as configured with the **Application Set Part Component**.
- 'Options' column as configured with the Application Condition Header - Group Component found within the **Application Condition Header Components** topic.
- 'Notes' column as configured with the **Application Comment Component**.
- 'Qualifiers' column (AutoCare solution only) as configured with the **ACES Application Qualifiers Component**.
- 'Assets' column as configured with the **Application Asset Reference Component**.

Note: Each column label can be edited as needed by accessing the Application Manager Node List Properties Child Component Headers parameter.

Vehicle •	Part Terminology •	Part Number •	Options •	Notes •	Qualifiers •	Assets •
2008 Acura CSX	Air Bag Sensor	034-VC30001	ACES Quantity: 1	🔍		🔍
2008 Acura CSX	A/C Drive Belt Idler Pulley Bearing	034-VC21499	ACES Sub Model: Technology	🔍		🔍
2009 Acura CSX	A/C Drive Belt Idler Pulley	034-VC36004	ACES Quantity: 1	🔍	12946 - with Air Bags 393 - 15 / 20 Press Fit Diameter 6 - #1 & #2 In take Inlets are Oval	🔍

Note: The example above is from an AutoCare Application Manager. However, NAPA and TecDoc Application Managers use the same results table settings, but the AutoCare specific headers are removed.

Application Condition Header Components

The Application Manager results table can be configured to display vehicle option data for applications within columns using the following two header components:

- **Application Condition Header - Individual:** Allows table headers to be expanded to display condition attributes and/or references in individual columns. Condition attributes can be displayed in their own column in the results table and/or Application Coverage Report when one of the following are configured:
 - Condition attribute is linked to the part type of the application, and has metadata DisplayCondition=true.
 - Condition attribute is linked to the part type of the application, and has an attribute ID = ReferencePartTypeLinks valid on a part type that contains 'true' after the reference ID.
- **Application Condition Header - Group:** This is used to group multiple conditions into a single column and requires that one or more attribute groups be selected for which all attributes and references in that group will be evaluated. Any attributes or references that have values but do *not* have the attribute linked to the part type with a DisplayCondition=true are displayed in a single column using this header. Clicking in the cell to edit options will open the Value editor where all populated options are displayed, along with all unpopulated options that are linked to the part type but do not have a true display condition. A search feature is also available to add unlinked options to the Value editor for population.

As described above, whether a vehicle option is displayed in its own column or in a consolidated column, and whether or not it displays by default in the Value editor or must be searched for, is determined by if and how the condition is linked to the part type.

If the condition is modeled using an attribute, the condition link is handled via a simple attribute link to the part type classification.

In the example below:

- **ACES Brake ABS** and **ACES Brake System** are both linked to the part type and have a 'true' Display Condition, allowing them to display within the Value editor, in their own columns and in the order indicated by the Display Sequence within the Application Manager results table.
- **ACES Mfr Label** and **ACES Quantity** both have values but are not linked to the part type, so they are displayed in the consolidated column and in the Value editor.
- **ACES Position** is also linked, but without a true Display Condition so it is displayed in the Value editor by default. Any other conditions can be accessed by typing in the search field.

The screenshot shows the 'Disc Brake Pad' reference table. The table has columns for ID, Name, Display Condition, and Display Sequence. A red box highlights the following rows:

ID	Name	Display Condition	Display Sequence
AC_PAdb_9059	Abutment Clips Included		
AC_ACESBrakeABS	ACES Brake ABS	true	2
AC_ACESBrakeSystem	ACES Brake System	true	1
AC_ACESPosition	ACES Position		

In the screenshot below, the **ACES Brake System** and **ACES Brake ABS** attributes are displayed as individual columns within the results table of an Application Manager. Whereas, the **ACES Mfr Label**, **ACES Position**, and **ACES Quantity** display within the Value editor.

The screenshot shows a table with columns: Vehicle, Part Terminology, Part Number, ACES Brake System, ACES Brake ABS, and Options. The 'Options' column contains 'ACES Mfr Label: Acura' and 'ACES Quantity: 1'. A red arrow points from this row to a 'Value editor - 1 item selected' dialog box. The dialog box has the following fields:

- ACES Mfr Label: Acura
- ACES Position: (dropdown menu)
- ACES Quantity: 1

At the bottom of the dialog box, there is a text input field with the placeholder 'Type name of new condition and hit Enter to add' and two buttons: 'OK' and 'Cancel'.

The behavior of the Value editor is the same, whether the condition is modeled using an attribute or a reference. However, on references, the condition link is modeled using the Reference Part Type Links metadata attribute (ID=ReferencePartTypeLinks). The Reference Part Type Links attribute is created by Easy Setup and made valid on part type objects in the Classification Hierarchy for any standards for which setup has been run (e.g., Part Terminology for AutoCare, MPCC for NAPA, and Generic Article for TecDoc). It can then be populated on part types using the same concepts as with an attribute, where both a display condition and display sequence are indicated. For example, in an AutoCare model where Engine Base and Transmission Base are references, if you wanted to make these options available on a particular part type, you would populate the Reference Part Type Links attribute on that part type using the following syntax:

[ReferenceType ID(string)]:[should it be displayed in separate column(boolean - true/false)]:
[IntegerForSortSequence(integer)]

For example:

Disc Brake Pad Set rev.0.4 - Classification								
Classification	Sub Products	References	Referenced By	Images & Documents	Tables	Status	State Log	Tasks
Description								
Name	>	>	Value					
ID			AC_PartTerminology_1684					
Name			Disc Brake Pad Set					
Object Type			Part Terminology					
Revision			0.4 Last edited by USER on Wed Sep 20 14:29:15 EDT 2017					
Approved			✘ Never Been Approved					
Translation			Not Translated					
Path			Classification 1 root/AutoCare Root/Parts Categorization (PCdb)/Brake/Disc Pads and Brake Shoes/Disc Brake Pad Set					
Visibility								
Delete Status		abc						
PCdb Part Alias		abc						
PCdb Part Description		abc						
PCdb Part Use			ACES PIES					
PCdb Revision Date			2003-02-07 00:00:00					
Reference Part Type Links		abc	AC_ACESApplicationToTransmissionBase:false:2;AC_ACESApplicationToEngineBase:true:1					

Since the transmission reference is set to false and the engine reference is set to true, the engine option is displayed within its own column in the Application Manager results table, and the transmission option is only displayed within the Value editor (as shown below).

	Vehicle	Part Terminology	Part Number	ACES Application To Engine Base	ACES Brake System	ACES Brake ABS	Options
<input type="checkbox"/>	2011 Acura TSX	Disc Brake Pad Set	034-VC21499				ACES Mfr Label: Acura ACES Quantity: 1

Value editor - 1 item selected

ACES Application To Transmission Base

ACES Mfr Label: Acura

ACES Position

ACES Quantity: 1

Type name of new condition and hit Enter to add

OK Cancel

Controlling Display of Conditions in Application Manager

The Application Manager results table can be configured to display vehicle option data for applications using an Application Condition Header component. For more information about the components, see the **Application Condition Header Component** topic.

Creating Applications

Creating an application for a vehicle can be done within the results table.

Once a vehicle is listed within the results table, a user can create an application by clicking into the Part Number cell, and providing a part number for the vehicle. In the example below the Part Number cell was blank, upon double clicking within the cell, the gray borders of the cell change to blue, indicating it is active and ready to be edited, and a node selector icon displays.

Vehicle •	Part Terminology •	Part Number •
1997 Acura CL	Auxiliary Light	<input type="text" value=""/>

Once the Part Number cell is active, users can type in the necessary part number. As the Part Number is typed in, the typeahead dropdown will display a list of potential part number matches. Users can use the down arrow key to navigate from the field down into the list, and then use the Enter key to populate the field with the part number. Optionally, the mouse pointer can be used to click on the valid part number.

Vehicle •	Part Terminology •	Part Number •
1997 Acura CL	Auxiliary Light	<input type="text" value="034"/> <ul style="list-style-type: none"> 034 034-VC30002 034-VC21499 034-VC36004 034-VC36004

Optionally, the node browser icon can be clicked, and the Part Number Search dialog will display (as shown below).

Part Number Search
✕

Search
Browse

🔍
Search

ID	Name

⏪
<
0
>
⏩

✓
OK

✕
Cancel

When the Search tab within the Part Number Search dialog is used, a portion of a part number can be typed in, and the Search button can be clicked to display a list of part IDs and Names. In the example below, '034' is typed into the search field, and four potential matches are displayed.

Part Number Search

✕

Search
Browse

🔍
Search

ID	Name
AC_PIEItem_GWWQ_034-VC21499	034-VC36004
AC_PIEItem_GWWQ_034-VC30002	034-VC30002
AC_PIEItem_GWWQ_034-VC36004	034-VC36004
AC_PIEItem_GWWQ_NikkiPart	034-VC36004

⏪
⏴
1-4 of 4
⏵
⏩

✓
OK

✕
Cancel

Using Multi-Select

In the Application Manager results table, it is possible to multi-select a set of records and edit them. If you use the consolidated options header for editing, a value editor dialog will display all attributes and references in the attribute group(s) identified in the header configuration AND have data populated on any of the selected records or have the attributes linked to one or more of the selected part types (in the case of references, the 'linking' is via the Reference Part Type Links metadata attribute on the part type).

If the records you have selected already have values populated in the displayed attributes / references, then the Value editor will display the data values for the cell that is double clicked by the user. In other words, if three records are selected and then the cell in the middle is double clicked, those records are the data values that will display.

Automotive Validation Path Functionality

Having the optional automotive validation path functionality in place ensures Application Manager users are able to select valid configurations when selecting criteria for their search, applying parts to vehicles, adding / editing options for applications, and maintaining missing applications. The premise behind the concept is that when some level of 'valid configuration' data for vehicles is provided, the Application Manager is able to intelligently decipher those.

Data for an automotive validation path is represented in STEP via a series of objects, references, and attributes. The values for those attributes and references are stored on the Vehicle Configuration objects, and the Application to Base Vehicle reference is applied to the individual application. Regardless of the model / standard you are working in, the system always evaluates the automotive validation path by beginning with the Base Vehicle / assembly object used in application records.

For example, for the Fuel Type attribute (AC_ACESFuelType), the validation path is configured as:

```
child.reference[type:'AC_VehicleToEngineConfig'].attribute[id:'AC_VCdbFuelType']
```

This automotive validation path is determined by:

1. Identifying the application and following the ACES Application To Base Vehicle reference (AC_ACESApplicationToBaseVehicle) to the Base Vehicle object type (AC_BaseVehicle).
2. Examining all vehicle children (AC Vehicle) of the Base Vehicle.
3. Following the Vehicle To Engine Config reference (AC_VehicleToEngineConfig) on the vehicle to the target object.
4. Evaluating the value of the VCdb Fuel Type attribute (AC_VCdbFuelType) that is stored on the Engine Config object (AC_EngineConfig).

After running Easy Setup, the application data model (Automotive - Application Model) includes configuration for object types and link types that define the relationship between different configurations of vehicles and conditions on applications. This relationship data is captured within the Automotive Validation Path attribute.

Within the AutoCare model, the validation path attribute is populated on some standard ACES vehicle options during Easy Setup when the attribute and/or reference modeling option is first created. This allows many features (including validation error handling) to be available within the Application Manager intelligent search interface, and results table value editor. Details on these features can be found within the **Intelligent Search Interface** and **Results Table and Toolbar** topics within this guide.

This section addresses:

- **Validation Path Functionality and Search Box Criteria**
- **Validation Path Functionality and Options on Applications**
- **Validation Path Functionality and Missing Applications**
- **Validation Path Functionality and Advanced Filtering**
- **Workbench Configurations Related to Validation Path**

Validation Path Functionality and Search Box Criteria

The validation path concept drives the intelligence that allows Application Manager search boxes to display only valid options filtered by previous selections and/or specific vehicle configurations.

For example, the 2013 Audi A3 was manufactured with six engine options, and some of those engines require diesel fuel instead of gas. When the Make/Model search box is populated with the Audi A3 criterion, and the Year search box is populated with the 2013 criterion, then the Fuel Type options search box will display the two available fuel types; Diesel and Gas (as shown below).

Note: Options search boxes can be configured with many different attributes and/or references. For the following examples, the Options search boxes are displayed using the Fuel Type and Engines attributes.

Make/Model Audi A3 ×	Year 2013 ×	Fuel Type ▾	Engine ▾
Enter Make/Model	Enter Year	Enter Fuel Type	Enter Engine
		DIESEL GAS	

When the Fuel Type options search box is left blank, then the Engine options search box will display the six engines available for the 2013 Audi A3 (as shown below).

Make/Model Audi A3 ×	Year 2013 ×	Fuel Type ▾	Engine ▾
Enter Make/Model	Enter Year	Enter Fuel Type	Enter Engine
			1.4L L4, 1390CC, 85CID 1.4L L4, 1395CC, 85CID 1.8L L4, 1798CC, 110CID 2.0L L4, 1968CC, 120CID 2.0L L4, 1984CC, 121CID 2.0L L4, 1984CC, 121CID

Whereas, when the Diesel criterion is selected for the Fuel Type options search box, then the Engine options search box will display the engine that uses diesel fuel and is available for the 2013 Audi A3 (as shown below).

Make/Model <input type="text" value="Audi A3"/>	Year <input type="text" value="2013"/>	Fuel Type ▾ <input type="text" value="DIESEL"/>	Engine ▾ <input type="text" value="2.0L L4, 1968CC, 120CID"/>
<input type="text" value="Enter Make/Model"/>	<input type="text" value="Enter Year"/>	<input type="text" value="Enter Fuel Type"/>	<input type="text" value="Enter Engine"/>

Notice in the examples provided above, the AND/OR toggle button is set to AND. When the toggle button is set to OR, then the Options search boxes (displaying the 'Fuel Type' and 'Engine' options in the screenshot below) do not limit the display of valid criteria based upon each other. However, the Options search boxes always consider the Make/Model and Year search box criteria before displaying the valid criteria options within the dropdown. In the example below, the same criteria is selected from the previous example, however the OR toggle button is active, and all the valid engines for the 2013 Audi A3 display.

Make/Model <input type="text" value="Audi A3"/>	Year <input type="text" value="2013"/>	Fuel Type ▾ <input type="text" value="DIESEL"/>	Engine ▾ <input type="text" value="1.4L L4, 1390CC, 85CID"/>
<input type="text" value="Enter Make/Model"/>	<input type="text" value="Enter Year"/>	<input type="text" value="Enter Fuel Type"/>	<input type="text" value="Enter Engine"/>

For more information on the Options search boxes, see the **Using the Options Search Boxes** topic within the Intelligent Search Interface section of this guide.




Validation Path Functionality and Options on Applications

The application data model (Automotive - Application Model) includes configurations for object types and link types that define the relationship between conditions / options on applications and different configurations of vehicles. This allows an Application Manager to be used to search for applications with specified conditions / options (i.e., engines, fuel type, regions, sub models). The validation path concept affects the options on missing applications by automatically populating selected options on applications, and allowing the selection of valid options within the Value editor.

Automatic Population of Options on Applications

When criteria are selected within the Options search boxes (displaying the 'Fuel Type' and 'Engine' options in the screenshot below), and the Search button is clicked, the Results table will display each vehicle that is missing an application with the Options cell automatically populated based upon the Options search box criteria. Once the part is applied to the vehicle the automatically populated options will be stored as references on the application.

For example, when an Options search box has the Fuel Type option type displayed with the Diesel criterion selected, the second Options search box has the Engine option type displayed with the '2.0L L4, 1968CC, 120CID' criterion selected, and the Search button is clicked, the results table will display the valid vehicle with the Options cell automatically populated with the specified Fuel Type and Engine (as shown below).

Make/Model Audi A3 X	Year 2013 X	Fuel Type DIESEL X	Engine 2.0L L4, 1968CC, 120CID X	Part Type Car Cover X
Enter Make/Model	Enter Year	Enter Fuel Type	Enter Engine	Enter Part Ty
Existing and Missing Applications		All Brands	Report	Clear All
Search				
Select all Clear filter Compressed view				
Vehicle	Part Terminology	Part Number	Options	
2013 Audi A3	Car Cover		<ul style="list-style-type: none"> → ACES Application To Engine Base: "2.0L L4, 1968CC, 120CID" → Fuel Type: DIESEL 	  

Options Within the Value Editor

Once an application has been created within the Results Table at the bottom of an Application Manager, double clicking the Options cell of the application row displays the 'Value editor.' Within the Value editor, some options automatically display, however additional options (conditions) can be selected to display for population. The selection of valid options, and their valid values, is driven by the validation path functionality.

Validation Path Functionality and Missing Applications

The automotive validation path functionality aids in displaying missing application coverage by evaluating all options and/or criteria for existing applications and displaying only those options that are valid for the vehicle configuration. In other words, consider when a vehicle is available with two different engines, and a part is applied to engine A, vehicles with engine B will display as a missing application.

When automotive validation path functionality is in place, and a part is applied for a vehicle, a missing application will display for each condition related to the part in the search as a single row within both the Application Manager results table, and the Application Coverage Report.

More information on missing application coverage can be found within the **Missing Application Coverage Functionality** topic of this guide.

Validation Path Functionality and Advanced Filtering

This can be helpful when searching for a part type that only pertains to specific configurations of a vehicle. For example, this type of relationship can be helpful when searching for spark plugs for a 2013 Audi A3, and the result table lists both Gas and Diesel engines.

Since Diesel engines do not use spark plugs, listing the Diesel engines as missing application coverage for spark plugs is inaccurate. However, when the 'Missing Application Conditions' attribute is used in conjunction with the 'Check path for missing application' business condition, and a validation path, the results table will no longer display the inaccurate option.

Error Handling

If an attribute contains an invalid Automotive Validation Path, the condition attribute with the invalid path will display all available options, other condition attributes will retain the valid filtered options in the dropdown list.

If the condition attribute with the invalid path is displayed in its own column, then a warning icon will be displayed and hovering over the icon will display a message indicating which reference or attribute contains the error.

If the condition attribute with the invalid path is not displayed in its own column, then accessing the value editor will display a yellow square with the error message below the condition with the invalid path.

In the example below, the ACES Aspiration attribute has an invalid path, and the error message is displayed.

Note: When using a Mac computer, invalid values (options) do not change style (gray out, italic font), but are instead represented with three dashes (---) before the invalid value. For example, in the screenshot below the ACES Sub Model dropdown on the left is from a PC using the Chrome browser, and the dropdown on the right is from a Mac using the Chrome browser. This problem exist only on Mac computers for both Safari and Chrome browsers.

Value editor - 1 item selected

ACES Application To Engine Base 2.4L L4, 2354CC, -CID ▾

ACES Engine Mfr Honda ▾

ACES Engine Version i-VTEC ▾

ACES Sub Model ▾

Type name of new c... ▾

Base
V6

✓ Base
---V6

✓ OK ✕ Cancel

Workbench Configurations Related to Validation Path Functionality

The automotive validation path functionality is automatically configured for the AutoCare standard through Easy Setup in the workbench. The validation path functionality must be manually configured within the workbench for the NAPA and TecDoc standards. Though configuration occurs in the workbench, it is not intended that the workbench be used to create, nor edit, part application data. Rather the Application Manager has been designed specifically for this. Nonetheless, it can be helpful to review the following workbench configurations related to automotive validation path prior to using or configuring an Application Manager.

This section addresses the following workbench configurations related to the automotive validation path functionality:

- **Vehicle Configurations**
- **Base Vehicle Applications**
- **Options on Applications**

Note: Within the AutoCare model, validation paths are populated on some standard ACES vehicle options during Easy Setup when the reference modeling option is first created. If an automotive validation path is subsequently deleted, it can be reapplied manually using the syntax described within this guide. The attribute or reference can be deleted and Easy Setup can be re-run (System Setup > Component Models > Automotive - AutoCare Model > 1. Configure AutoCare Data Model). However, this is not the recommended method for production environments since deleting the reference also deletes all existing data for the reference. It is recommended to re-populate a deleted validation path manually on production environments.

Vehicle Configurations

Options for different vehicle configurations are stored within STEP as attributes on a classification object type. Classification object types are referenced by Vehicles that are children of the Base Vehicle object type.

In the example below, the **VCdb Body Num Doors**, and **VCdb Body Type** AutoCare options are modeled as attributes within the **Body Style Config** classification object type. The valid configuration for the '2 door, Coupe' consists of '2' doors and a 'Coupe' body type.

Tree		2 door, Coup	
		Classification	Sub Products
AutoCare Root		Description	
ACES Assets		Name	Value
Brand Table		ID	AC_BodyStyleConfig_7
Parts Categorization (PCdb)		Name	2 door, Coupe
PIES Assets		Object Type	Body Style Config
Qualifiers (Qdb)		Revision	0.2 Last edited by USER on
Vehicle Configuration (VCdb)		Approved	✘ Never Been Approved
Vehicle Configurations		Translation	Not Translated
Bed Configurations		Path	Classification 1 root/AutoCa
Body Style Configurations		Visibility	
0 door, Cab & Chassis		Delete Status	abc
0 door, Motor Home - !		VCdb Body Num Doors	2
0 door, Sport Utility		VCdb Body Type	Coupe
0 door, Stripped Chass			
0 door, Stripped Chass			
2 door, Cab & Chassis			
2 door, Cab & Chassis			
2 door, Convertible			
2 door, Coupe			

All 2 door coupe vehicles reference the **2 door, Coupe** body style using the **Vehicle To Body Style Config** Reference Type. A vehicle can have more than one 'Vehicle To Body Style Config' reference and if so, it will display as multiple references on the vehicle (as shown below).

Tree

- Assets
 - AutoCare Root
 - ACES Assets
 - Brand Table
 - Parts Categorization (PCdb)
 - PIES Assets
 - Qualifiers (Qdb)
 - Vehicle Configuration (VCdb)
 - Vehicle Configurations
 - Bed Configurations
 - Body Style Configurations
 - 0 door, Cab & Chassis
 - 0 door, Motor Home - S
 - 0 door, Sport Utility
 - 0 door, Stripped Chassis
 - 0 door, Stripped Chassis
 - 2 door, Cab & Chassis
 - 2 door, Cab & Chassis
 - 2 door, Convertible
 - 2 door, Coupe
 - 2 door, Mini Passenger

2 door, Coupe rev.0.2 - Referenced By

Classification	Sub Products	References	Referenced By	Images & Documents	Tables
Ungrouped Referenced by Products, Classifications or Entities					
Reference Type	Source				
	1975 Lamborghini Countach, LP400, United States				
	1976 Lamborghini Countach, LP400, United States				
	1978 Plymouth Sapporo, Base, United States				
	1979 Plymouth Sapporo, Base, United States				
	1980 Plymouth Sapporo, Base, United States				
	1981 Lamborghini Countach, LP400S, United States				
	1982 Lamborghini Countach, LP400S, United States				
	1983 Lamborghini Countach, LP500S, United States				
	1984 Lamborghini Countach, LP500S, United States				
	1989 Lamborghini Countach, 25th Anniversary, United States				
	1990 Lamborghini Diablo, Base, United States				
	1991 Lamborghini Diablo, Base, United States				
	1992 Lamborghini Diablo, Base, United States				
	1993 Eagle Summit, DL, United States				
	1996 Eagle Summit, DL, United States				
	1996 Eagle Summit, FSI, United States				

Vehicle To Body Style Config +

In the example below, the **1996 Dodge Viper GTS, United States** has only one valid **Vehicle To Body Style Config** reference target, the **2 door, Coupe**.

Tree

- Viper
 - 1992 Dodge Viper
 - 1993 Dodge Viper
 - 1994 Dodge Viper
 - 1995 Dodge Viper
 - 1996 Dodge Viper
 - 1996 Dodge Viper, Base, Canada
 - 1996 Dodge Viper, Base, United States
 - 1996 Dodge Viper, GTS, Canada
 - 1996 Dodge Viper, GTS, United States

1996 Dodge Viper, GTS, United States

Images & Documents	Tables	Status	State Log	Tasks
Classification	Sub Products	References		
Ungrouped Classification References				
Reference Type	Target			
	N/R, N/R, N/R			
Vehicle To Body Style Config +	2 door, Coupe			
Vehicle To Brake Config +	Power, Non-ABS, Front=Disc			
Vehicle To Drive Type +	RWD			
Vehicle To Engine Config +	8.0L L10, 7990CC, 488CID, F			

Since only one valid body style exists for this vehicle, it would be invalid for an application to be supplied for the vehicle where any other body style was implied. Additionally, within the Application Manager, it is not possible for a user to select a value other than 2 for the Body Num Doors value, or Coupe for the Body Type value.

Base Vehicle Applications

Part application data is not applied to Vehicle object types (i.e., **1996 Dodge Viper, GTS, United States**) but rather to Base Vehicle object types (i.e., **1996 Dodge Viper**). Thus, an application is applied to the parent of the Vehicle object type.

In the example below, the **1996 Dodge Viper** Base Vehicle object type is displayed along with its four children vehicle object types

Tree		1996 Dodge Viper rev.0.2 - Classification																																																																							
<ul style="list-style-type: none"> Viper <ul style="list-style-type: none"> 1992 Dodge Viper 1993 Dodge Viper 1994 Dodge Viper 1995 Dodge Viper 1996 Dodge Viper <ul style="list-style-type: none"> 1996 Dodge Viper, Base, Canada 1996 Dodge Viper, Base, United States 1996 Dodge Viper, GTS, Canada 1996 Dodge Viper, GTS, United States 1997 Dodge Viper 1998 Dodge Viper 1999 Dodge Viper 		<table border="1"> <thead> <tr> <th colspan="2">Classification</th> <th>Sub Products</th> <th>References</th> <th>Referenced By</th> <th>Images & Documents</th> <th>Tables</th> </tr> </thead> <tbody> <tr> <td colspan="7">Description</td> </tr> <tr> <td>Name</td> <td>></td> <td>></td> <td colspan="4">Value</td> </tr> <tr> <td>ID</td> <td>></td> <td colspan="5">AC_BaseVehicle_2090</td> </tr> <tr> <td>Name</td> <td>></td> <td colspan="5">1996 Dodge Viper</td> </tr> <tr> <td>Object Type</td> <td>></td> <td colspan="5">Base Vehicle</td> </tr> <tr> <td>Revision</td> <td>></td> <td colspan="5">0.2 Last edited by NIFE on Fri Aug 18 15:09:31 EDT 2017</td> </tr> <tr> <td>Approved</td> <td>></td> <td colspan="5">✘ Never Been Approved</td> </tr> <tr> <td>Translation</td> <td>></td> <td colspan="5">Not Translated</td> </tr> <tr> <td>Path</td> <td>></td> <td colspan="5">Classification 1 root/AutoCare Root/Vehicle Configuration (VCdb)/Vehi</td> </tr> </tbody> </table>		Classification		Sub Products	References	Referenced By	Images & Documents	Tables	Description							Name	>	>	Value				ID	>	AC_BaseVehicle_2090					Name	>	1996 Dodge Viper					Object Type	>	Base Vehicle					Revision	>	0.2 Last edited by NIFE on Fri Aug 18 15:09:31 EDT 2017					Approved	>	✘ Never Been Approved					Translation	>	Not Translated					Path	>	Classification 1 root/AutoCare Root/Vehicle Configuration (VCdb)/Vehi				
Classification		Sub Products	References	Referenced By	Images & Documents	Tables																																																																			
Description																																																																									
Name	>	>	Value																																																																						
ID	>	AC_BaseVehicle_2090																																																																							
Name	>	1996 Dodge Viper																																																																							
Object Type	>	Base Vehicle																																																																							
Revision	>	0.2 Last edited by NIFE on Fri Aug 18 15:09:31 EDT 2017																																																																							
Approved	>	✘ Never Been Approved																																																																							
Translation	>	Not Translated																																																																							
Path	>	Classification 1 root/AutoCare Root/Vehicle Configuration (VCdb)/Vehi																																																																							

In order to know all valid body style configurations for a Base Vehicle object type, all child vehicles (the 'Base' and 'GTS' vehicles for the United States and Canada) must first be evaluated. In doing this, it can be seen that the 1996 Dodge Viper Base Canada and United States models have a **2 door, Convertible** body style (as shown below) whereas the 1996 Dodge Viper GTS Canada and United States models have the **2 door, Coupe** body style (as previously mentioned).

Tree		1996 Dodge Viper, Base, United States rev.0.2 - Reference																																																																																												
<ul style="list-style-type: none"> Viper <ul style="list-style-type: none"> 1992 Dodge Viper 1993 Dodge Viper 1994 Dodge Viper 1995 Dodge Viper 1996 Dodge Viper <ul style="list-style-type: none"> 1996 Dodge Viper, Base, Canada 1996 Dodge Viper, Base, United States 1996 Dodge Viper, GTS, Canada 1996 Dodge Viper, GTS, United States 1997 Dodge Viper 1998 Dodge Viper 1999 Dodge Viper 2000 Dodge Viper 2001 Dodge Viper 		<table border="1"> <thead> <tr> <th colspan="2">Referenced By</th> <th>Images & Documents</th> <th>Tables</th> <th>Status</th> <th>State Log</th> <th>Tasks</th> </tr> </thead> <tbody> <tr> <td colspan="2">Classification</td> <td colspan="2">Sub Products</td> <td colspan="3">Referen</td> </tr> <tr> <td colspan="7">Ungrouped Classification References</td> </tr> <tr> <td>Reference Type</td> <td>></td> <td>></td> <td colspan="4">Target</td> </tr> <tr> <td>Vehicle To Bed Config</td> <td>+</td> <td>📁</td> <td colspan="4">N/R, N/R, N/R</td> </tr> <tr> <td>Vehicle To Body Style Config</td> <td>+</td> <td>📁</td> <td colspan="4">2 door, Convertible</td> </tr> <tr> <td>Vehicle To Brake Config</td> <td>+</td> <td>📁</td> <td colspan="4">Power, Non-ABS, Front=Disc, Rear=Disc</td> </tr> <tr> <td>Vehicle To Drive Type</td> <td>+</td> <td>📁</td> <td colspan="4">RWD</td> </tr> <tr> <td>Vehicle To Engine Config</td> <td>+</td> <td>📁</td> <td colspan="4">8.0L L10, 7990CC, 488CID, FI, MFI, Electronic, SFI</td> </tr> <tr> <td>Vehicle To Mfr Body Code</td> <td>+</td> <td>📁</td> <td colspan="4">SR</td> </tr> <tr> <td>Vehicle To Spring Type Config</td> <td>+</td> <td>📁</td> <td colspan="4">Front=Coil, Rear=Coil</td> </tr> <tr> <td>Vehicle To Steering Config</td> <td>+</td> <td>📁</td> <td colspan="4">Power, Rack</td> </tr> <tr> <td>Vehicle To Transmission</td> <td>+</td> <td>📁</td> <td colspan="4">Standard, 6 Speed, T56, Borg Warner</td> </tr> </tbody> </table>		Referenced By		Images & Documents	Tables	Status	State Log	Tasks	Classification		Sub Products		Referen			Ungrouped Classification References							Reference Type	>	>	Target				Vehicle To Bed Config	+	📁	N/R, N/R, N/R				Vehicle To Body Style Config	+	📁	2 door, Convertible				Vehicle To Brake Config	+	📁	Power, Non-ABS, Front=Disc, Rear=Disc				Vehicle To Drive Type	+	📁	RWD				Vehicle To Engine Config	+	📁	8.0L L10, 7990CC, 488CID, FI, MFI, Electronic, SFI				Vehicle To Mfr Body Code	+	📁	SR				Vehicle To Spring Type Config	+	📁	Front=Coil, Rear=Coil				Vehicle To Steering Config	+	📁	Power, Rack				Vehicle To Transmission	+	📁	Standard, 6 Speed, T56, Borg Warner			
Referenced By		Images & Documents	Tables	Status	State Log	Tasks																																																																																								
Classification		Sub Products		Referen																																																																																										
Ungrouped Classification References																																																																																														
Reference Type	>	>	Target																																																																																											
Vehicle To Bed Config	+	📁	N/R, N/R, N/R																																																																																											
Vehicle To Body Style Config	+	📁	2 door, Convertible																																																																																											
Vehicle To Brake Config	+	📁	Power, Non-ABS, Front=Disc, Rear=Disc																																																																																											
Vehicle To Drive Type	+	📁	RWD																																																																																											
Vehicle To Engine Config	+	📁	8.0L L10, 7990CC, 488CID, FI, MFI, Electronic, SFI																																																																																											
Vehicle To Mfr Body Code	+	📁	SR																																																																																											
Vehicle To Spring Type Config	+	📁	Front=Coil, Rear=Coil																																																																																											
Vehicle To Steering Config	+	📁	Power, Rack																																																																																											
Vehicle To Transmission	+	📁	Standard, 6 Speed, T56, Borg Warner																																																																																											

Therefore the **1996 Dodge Viper Base Vehicle** object type has two valid body style configurations; both have only 2 doors, but the body type can be either Coupe or Convertible. If a user tries to provide an application for the vehicle that specifies '4' doors or a 'Sport Utility' Vehicle body style, it would be allowed, but also invalid.

Options on Applications

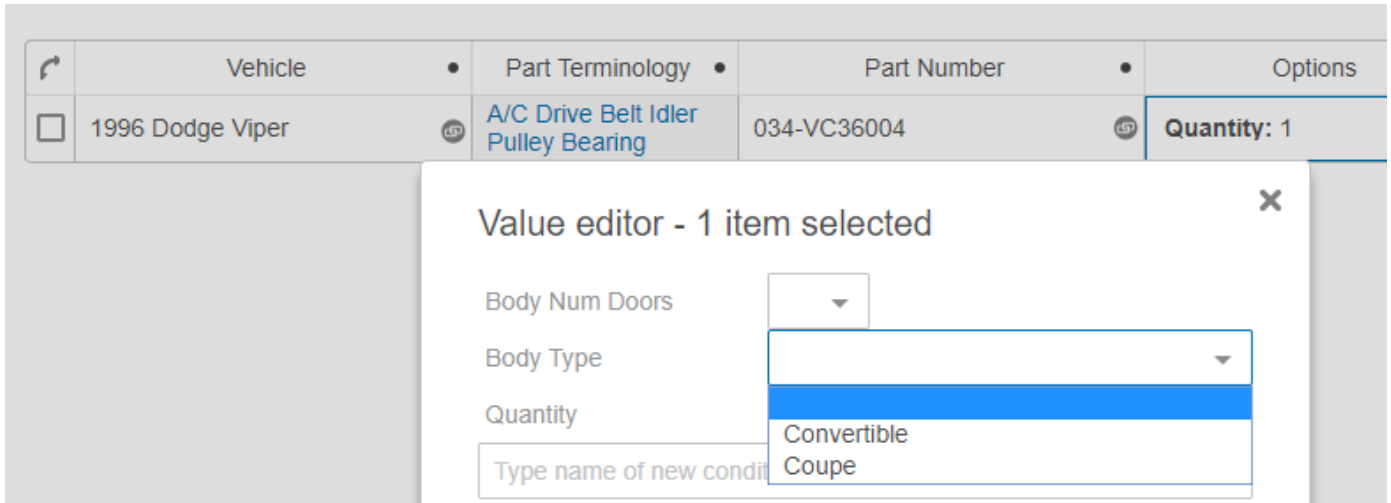
On an application, an attribute or reference exists to model each data point in a valid configuration. For the **Vehicle To Body Style Config** reference type an attribute for **Body Num Doors** and **Body Type** exists (similar to the configuration classification object shown above).

In the workbench example below, the dropdown for the **Body Type** attribute displays all the available body types that can be chosen for the selected application.

AutoCare ACES Attributes	
Name	Value
> Aspiration	
> Bed Length	
> Bed Type	
> Body Num Doors	
> Body Type	<ul style="list-style-type: none"> Cab & Chassis - Medium Conventional (98) Cab & Chassis - Stripped Chassis (58) Convertible (10) Coupe (9) Crew Cab Pickup (22) Cutaway (102) Cutaway - Cab Forward (108) Cutaway Van (26)
> Brake ABS	
> Brake System	
> Cylinder Head Type	
> Drive Type	
> Engine Designation	
> Engine Mfr	
> Engine Version	
> Engine VIN	

Though the workbench could be used to select any one of the available options, this would be done without the safeguarding of the intelligent automotive validation path functionality, and therefore is not recommended. Applications are not intended to be created nor edited in the workbench. Instead, a robust results table and value editor are provided within an Application Manager Web UI screen.

Within a Web UI Value editor, users can view and select only the valid options for the vehicle they are working with. In the example below, the Application Manager results table 'Value editor' is displayed for the **1996 Dodge Viper** Base Vehicle, and only the **Convertible** and **Coupe** options display for selection in the dropdown.



When an application is created, validation paths are available for conditions / options stored as either attributes or references. However, each has its own configuration, as described below.

Options / Conditions Stored as Attributes

When the data model uses an attribute to store the vehicle option / criteria, the automotive validation path data is managed as described below.

Easy Setup creates an Automotive Validation Path attribute (AutomotiveValidationPath). This attribute is made valid as metadata on the attribute basic object type, making it available for population on any attribute.

Attribute	References	Attribute Transformation	Validity	Profile	Log	State Log	Tasks
Description							
Name	>	>	Value				
ID			AC_ACESBodyNumDoors				
Name			Body Num Doors				
Last edited by			2017-07-07 09:47:16 by USER				
Full Text Indexable			No				
Externally Maintained			No				
Completeness Score							
Hierarchical Filtering			None				
Calculated			No				
Type			Specification				
Dimension Dependencies							
Mandatory			No				
Attribute Help Text		abc					
Automotive Validation Path		abc	child.reference[type:'AC_VehicleToBodyStyleConfig'].attribute[id:'AC_VCdbBodyNumDoors']				

Regardless of the model / standard you are working in, the system always evaluates the automotive validation path by beginning with the Base Vehicle / assembly object used in application records.

In the example above, the automotive validation path data within the data model is determined by:

1. Identifying the application and following the ACES Application To Base Vehicle reference (AC_ACESApplicationToBaseVehicle) to the Base Vehicle.
2. Examining all vehicle children of the Base Vehicle.
3. Following the Vehicle To Body Style Config (AC_VehicleToBodyStyleConfig) reference on the vehicle to the Body Style Config target object.
4. Evaluating the value of the VCdb Body Num Doors attribute (AC_VCdbBodyNumDoors) on that target object.

Syntax

The syntax of the automotive validation path is a series of commands, each separated by a period (.). Using the child, reference, ID (shown below), and attribute elements (including IDs of the applicable references and attributes to be followed), an automotive validation path can be applied to any attribute for which the data is modeled in STEP. When the vehicle option is modeled using an attribute, the final element of the syntax should always be an attribute where the value should be retrieved.

Note: Within the AutoCare model, validation paths are populated on some standard ACES vehicle options during Easy Setup when the reference modeling option is first created. If an automotive validation path is subsequently deleted, it can be reapplied manually using the syntax described within this guide. The attribute or reference can be deleted and Easy Setup can be re-run (System Setup > Component Models > Automotive - AutoCare Model > 1. Configure AutoCare Data Model). However, this is not the recommended method for production environments since deleting the reference also deletes all existing data for the reference. It is recommended to re-populate a deleted validation path manually on production environments.

Options / Conditions Stored as References

When the data model uses a reference to store the vehicle option / criteria, the automotive validation path data is managed as described below.

The Automotive Validation Path attribute described in the previous section is still relevant. When created by Easy Setup, the attribute is made valid on the Reference Type basic object type (in addition to the attribute basic object type). Thus, the Automotive Validation Path attribute can then be populated on any reference.

Reference Type	Validity	Log
Description		
Name	>	Value
> ID		AC_ACESApplicationToEngineBase
> Name		ACES Application To Engine Base
> Last edited by		2017-07-07 10:24:13.0 by USER
> Externally Maintained		No
> Dimension Dependencies		
> Completeness Score		
> Allow multiple references		No
> Mandatory		No
> Inheritance		None
> Automotive Validation Path	abc	child.reference[type:'AC_VehideToEngineConfig'].reference[type:'AC_EngineConfigToEngineBase'].ID

In the example above, the automotive validation path data within the data model is determined by:

1. Identifying the application and following the ACES Application To Base Vehicle reference (AC_ACESApplicationToBaseVehicle) to the Base Vehicle.
2. Examining all vehicle children of the Base Vehicle.
3. Following the Vehicle To Engine Config reference (AC_VehideToEngineConfig) on the vehicle to the Engine Config target object.
4. Following the Engine Config To Engine Base reference (AC_EngineConfigToEngineBase) from the Engine Config Object (which are now considered sources of the Engine Config To Engine Base reference) to the Engine Base target object.
5. Retrieving the STEP Name of the Engine Base object to be displayed in Application Manager.

Syntax

The syntax of the automotive validation path is a series of commands, each separated by a period (.). Using the child, reference, ID, and attribute elements (including IDs of the applicable references and attributes to be followed), an automotive validation path can be applied to any reference for which the data is modeled in STEP. When the vehicle option is modeled using a reference, the final element of the syntax should always be an ID of the reference where the STEP Name should be displayed.

Configuring Application Manager

The Easy Setup actions must be run for each standard that will be accessed using an Application Manager, and by default each standard is configured to use their own instance of an Application Manager. Optionally, after Easy Setup actions have configured an Application Manager, the Vehicle Type Search Panel and results table columns can be manually modified. Before doing this, it is helpful to review how the Vehicle Type Search Panel and results table are used per the **Using Intelligent Search Interface** topic, the **Results Table and Toolbar** topic, and then review the search box configuration options within this topic. Additionally, for information on adding columns to the results table, see the **Adding Additional Headers to Application Screens** topic within the **Automotive Quick Start Guide**.

When Easy Setup actions have been used to create a standard data model, by default, much of the Application Manager is ready to be used, including the addition of an easy to use link to the Application Manager screen from the Quick Links widget on the Web UI Homepage. For more information, see the **Application Manager** topic.

Important: An error will occur when trying to access an Application Manager screen if the Vehicle table classifications have not been properly configured. When this error displays, the user must log out and then back in to the Web UI. Once the Vehicle reference data is imported, the link to the Application Manager will function properly.

Own Model Considerations

An Application Manager screen can be configured for use with an Own model. It is easiest to accomplish this after first using the Easy Setup actions to configure the Automotive - Application Model for an automotive standard.

Important: For an Application Manager to function properly, some component model types (i.e., Part Type) may need to be configured, even if otherwise not needed.

To implement an Own model within an Application Manager screen, the following must be in place:

1. All components within the Automotive - Application Model must be populated with a value.
2. The Application, Assembly, Part, Part type, Part type list, Application to assembly, Leading part relation, Leading part type relation, Part type link type, and Part type list to part type aspects must be configured with Own object types and references in the Automotive - Application Model component.
3. Create a new Product to Classification Link Type and define the own part and application object types as valid Product Types, and the own part type object type as a valid Classification Type. Then add this new Product to Classification Link Type to the 'Part type link type' aspect within the Automotive - Application Model.
4. The 'Part type list to part type' aspect requires a Classification Reference Type to connect a part type list / group to a part type that is used in behind the scene code (not visible for a user to interact with). The object type that is used for the Valid Source Type in the reference must also be defined in the 'Part type list' aspect. The object type that is used for the Valid Target Type must also be defined in the 'Part type' aspect.
5. The Vehicle Type Search Panel within an Application Manager must be configured to use the Own application and vehicle object types, and Own vehicle root node.

6. The own application must have the references established to the own vehicle and part type.

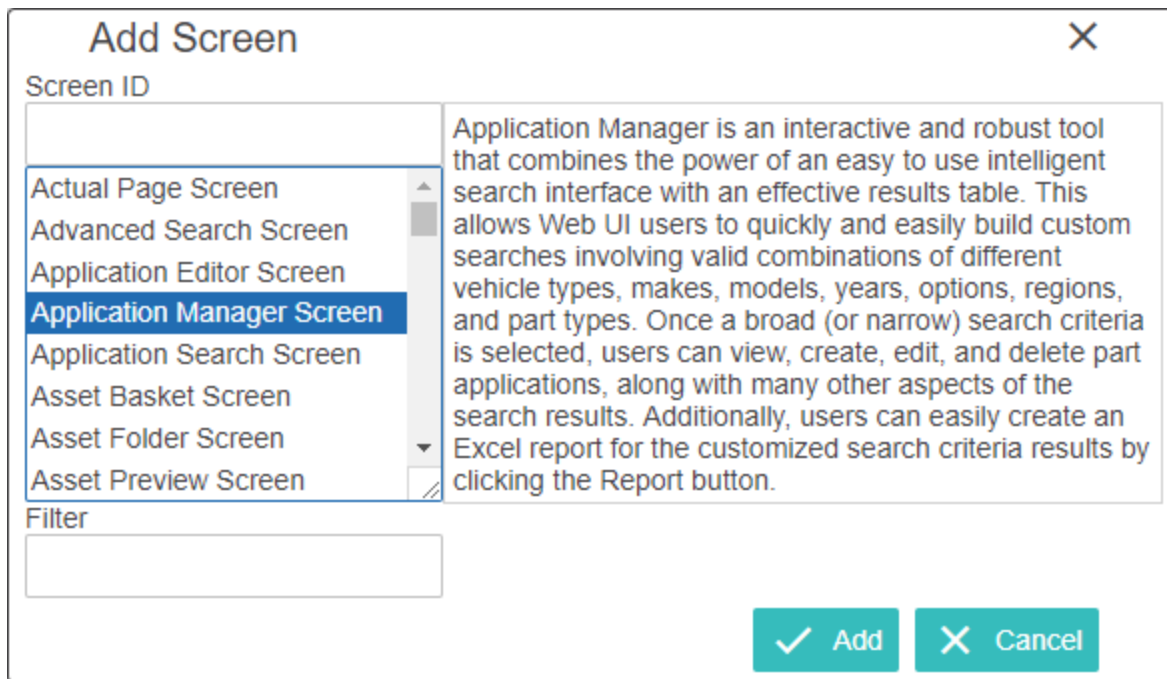
Configuring an Application Manager Screen in Web UI

Steps to configure an Application Manager screen in Web UI are below.

Note: After the initial setup, access to the configuration dialogs does not change, therefore these steps can be helpful when editing the configuration. However, when editing, the 'Add' button is replaced with the 'Save' button, and the 'Add component...' labels at the top of the dialogs display as 'Edit component....'

Important: Easy Setup actions for at least one standard must be run before an Application Manager Screen can be manually added to a Web UI and configured for that standard.

1. Log in to the Web UI where the Application Manager will be used, and access the Web UI Designer.
2. Click the **New** button at the top of the Designer, and the Add Screen dialog will display.



3. Search for and select the 'Application Manager Screen,' enter a Screen ID, click the **Add** button, and the Application Manager Screen Properties dialog will display as shown below.

Application Manager Screen Properties

Application Scope Value:

Hierarchy Restriction Label:

Hierarchy Restriction Object Types:

Title:

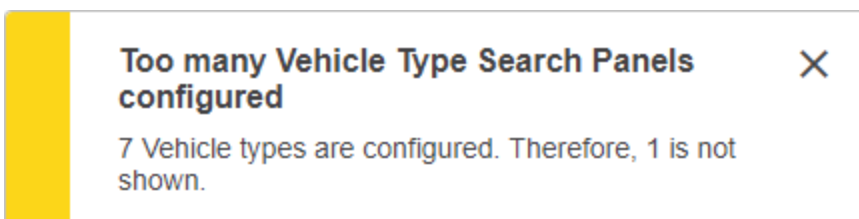
Child Components

Default Node List: [go to component](#)

*Vehicle Type Search Panels:

Within the Application Manager Screen Properties dialog, the following parameter is required:

Vehicle Type Search Panel: Allows for up to six different Vehicle Type Search Panels to be configured within each Application Manager. Populating this required parameter allows for a Vehicle type icon and Vehicle Type Search Panel to display within the Application Manager. If more than six vehicle type search panels are added, only the first six vehicle types listed will display for the Application Manager user. Additionally, a warning dialog will display each time the Designer is accessed informing the user that more vehicle type search panels are configured than can be displayed. In the example below, seven vehicle type search panels have been configured.



Note: If a vehicle search type is created, and the vehicle root node that has been configured for the Vehicle Type is later removed from STEP, then the vehicle search type created within the Application Manager will no longer display. If the vehicle root node is re-created, then the Application Manager will display as it did before the vehicle root node was deleted.

Within the Application Manager Screen Properties dialog, the following parameters are optional:

Application Scope Value: This parameter sets the default drop down value in the Missing Application Coverage functionality within Application Manager Screen. Within this parameter, user can set default values to either 'Existing Applications Only' or 'Existing and Missing Applications' or 'Missing Applications.' Typically, this parameter is used to set the default value in the dropdown located below the search boxes of Application Manager Screen, which can help users to perform required search without changing the options every time. Despite the default setting in the Application Scope Value parameter, user can still swap amongst the options in Application Manager Screen to perform required search. For information on missing application coverage functionality, see the **Missing Application Coverage Functionality** topic.

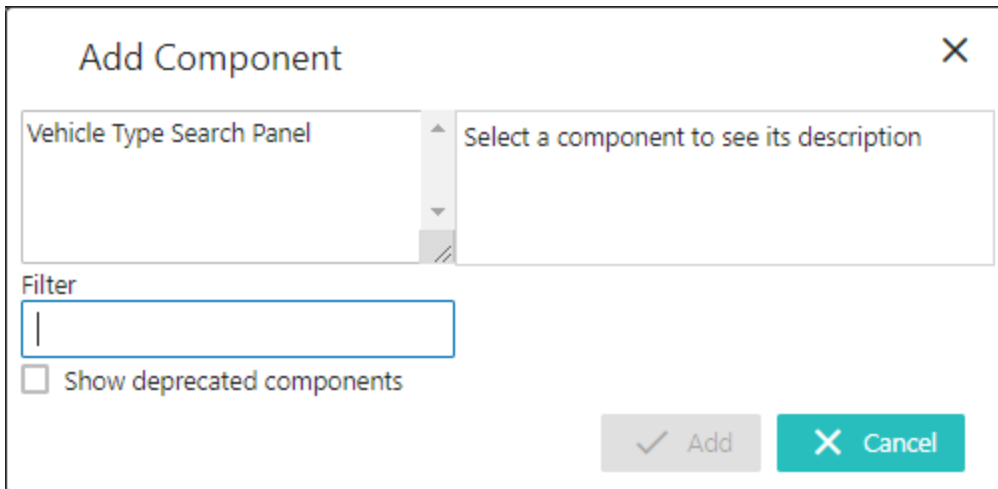
Hierarchy Restriction Label: Type a label that best describes the Hierarchy Restriction option to the Application Manager user. By default, the label is populated with 'All data.' If users are filtering by brand, then 'Brand' could be an appropriate label.

Hierarchy Restriction Object Types: Allows users to restrict the Application Manager search results by an object type. Users can select from a pre-configured dropdown list of product or classification object types. This can be helpful when search results needs to be filtered by Brand, Manufacturer, OEM, Product Line, etc. Once an object types is added, it will display in the dropdown list. Use the Up and/or Down buttons to edit the order of the dropdown list, as the objects will display to the user in the order listed within the parameter. When the parameter is blank, the Hierarchy Restriction dropdown will not display within the Application Manager. Adding at least one object type to the parameter will result in the Hierarchy Restriction dropdown displaying below the search boxes of the Application Manager.

Title: If desired, type in a title which will display on the screen in the top left corner. If no text is entered, no label is displayed.

Default Node List: This parameter functions as the generic Node List component for all the Vehicle Type Search Panel listed within the Vehicle Type Search Panels parameter unless a specific configuration is done on each individual Vehicle Type Search Panel. Select 'Node List' option in the dropdown and click on the 'go to component' to further configure the Node List component. For more information, see the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

4. Click the **Add** button below the Vehicle Type Search Panels parameter, and the 'Add component' dialog will display as shown below.



5. Select the Vehicle Type Search Panel component, click **Add**, and the newly added component will display within the Vehicle Type Search Panel parameter.
6. Double click the newly added component available in the Vehicle Type Search Panel parameter, and the Vehicle Type Search Panel Properties dialog will display (as shown below). Parameter options are explained below:

Vehicle Type Search Panel Properties [go to parent](#)

Application Object Types

Search Box Types

Vehicle Type Icon ▼

Name Cars

Root Nodes

UUID ee5fcd1c-839d-4282-ac38-548d29a474fe

Use Default Application Editor

Vehicle Object Types

Child Components

Node List Node List ▼ [go to component](#)

Within the Vehicle Type Search Panel Properties dialog, the following parameters are required:

Whenever a new Vehicle Type Search Panel is added, all required parameters that need to be populated are displayed in the main screen (as shown below). To populate the required parameters, manually navigate to the Vehicle Type Search Panel Properties dialog for the newly added Vehicle Type Search Panel.

The screenshot shows the StiboSystems Application Manager interface. On the left is a 'Tree' view with a list of applications and their counts. A red box highlights an error message: 'Error in: Vehicle Type Search Panel (#6)' with sub-items: 'Application Object Types', 'Vehicle Object Types', 'Search Box Types', and 'Root Nodes'. The main area shows the 'Configuration' tab for 'AutoCare Application' with buttons for 'Save', 'Close', 'New...', 'Delete', 'Rename', and 'Save as...'. Below this is the 'Application Manager Screen Properties' section, specifically the 'Child Components' area. It shows a 'Default Application Editor' with a 'Node List' dropdown and a 'go to component' link. Underneath, a list of 'Vehicle Type Search Panels' is shown, each with an 'Add..' button and 'Remove', 'Up', and 'Down' buttons.

Application Object Types: Allows for applications created within the Application Manager to be restricted to the listed object type. Click the Add button below the parameter field to browse or search for the necessary object types.

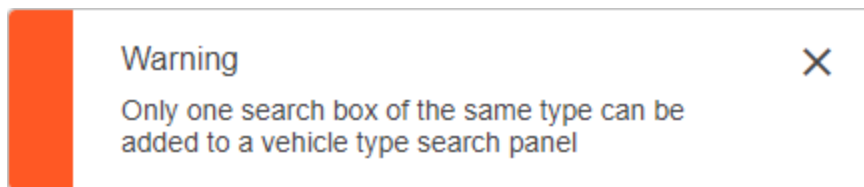
- Object type(s) selected for this parameter must be configured as an Application within the 'Automotive - Application Model.' Easy Setup configures the Automotive - Application Model in this way.
- Any object that is selected, and is configured as an Application in the Automotive - Application Model, will be displayed in the search results table. For example, a user can have their own application, as well as an AutoCare ACES application.
- When applications are created, their object type is determined based on the part number that is applied to the Vehicle. For example, if part number 123 is a PIES Item, and part number 456 is a Own part number, then when a user applies part number 123 to a Vehicle, then the application record is created using an ACES application object type.

Search Box Types: Allows one to four search boxes to be added to the Vehicle Type Search Panel. Only one search box is required, and each search box type can be added only once. The order listed determines the order of display within the Application Manager search panel. The Part Type search box type does not require

additional configuration, whereas the other search box types do. Clicking the Add button below the Search Box Types parameter displays an Add Component dialog with the option to add one of the following search box types: Make/Model, Options Group, Part Type, and Year. Steps for configuring each search box type can be found in their respective sections below:

- Configuring Make/Model Search Box
- Configuring Attribute Search Box
- Configuring Options Group Search Box
- Configuring Part Type Search Box
- Configuring Year Search Box

Because each search box type can be only be configured once for each Application Manager, if a second search box of the same type is added, a Warning dialog will display as shown below, and the Search Box Types parameter will remain unchanged.



Name: Allows for a custom name for the Vehicle Type Search Panel to display when a user hovers over the Vehicle type icon. Type in the desired name.

Root Nodes: Allows for the restriction of root nodes to be used within the Application Manager when searching for a vehicle. The root node(s) listed should correspond to the Icon and Name parameters because the search input field displays only suggestions for Make/Model objects that belong in the root node defined within this parameter. Additionally, when the Node Picker is used, its dialog will display the only those classification folders that pertain to the root nodes defined within this parameter. Click the Add button below the parameter field to browse or search for the necessary vehicle root node object(s).











Use Default Application Editor: Allows the user to toggle between the use of the default Node List configured on the Application Manager Screen Properties dialog and the Node List configured on the Vehicle Type Search Panel Properties dialog.

Vehicle Object Types: Allows vehicles within the Application Manager to be of the listed object type. Object type(s) selected for this parameter must be configured as an Assembly aspect within the 'Automotive - Application Model.' Easy Setup configures the Automotive - Application Model in this way. Click the Add button below the parameter field to browse or search for the necessary object types.

Within the Vehicle Type Search Panel Properties dialog, the following parameters are optional:

Vehicle Type Icon: The Vehicle type icon is a visual representation of the Vehicle Type Search Panel. Clicking a Vehicle type icon displays the Vehicle Type Search Panel configured specifically for that vehicle type (i.e., Personal Cars, Buses, Marine, Street Bikes). Hovering over the icon will display the name configured for the Vehicle Type Search Panel. Each Application Manager must have at least one Vehicle type icon. However, only up to six icons can be displayed within the same Application Manager. Therefore up to six Vehicle Type

Search Panels can be specifically configured to best meet the needs of different search types. The following Vehicle Type icons are available for selection:

-  Directions Car
-  Motorcycle
-  Directions Bus
-  Directions Boat
-  Agricultural And Farm
-  Commercial Vehicles And Vans
-  Industrial
-  Large Commercial Vehicles And Trucks
-  Marine
-  Recreational And Sports

Note: The values populated for the required fields should all correspond to one or more vehicle types being configured for the Application Manager.

Double click on the newly added Vehicle Type Search Panel in the Application Manager screen Properties dialog to configure the following optional parameter:

Node List: Allows configuration of the Node List component for the selected Vehicle Type Search Panel. Node List configured under this parameter will only be applicable for the selected Vehicle Type Search Panel and shall overwrite all the configuration made within Default Node List parameter in the Application Manager screen properties dialog. Select 'Node List' option in the dropdown and click on the 'go to component' to configure the Node List component for the selected Vehicle Type Search Panel. For more information, see the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

Below is an example of a configured Vehicle Type Search Panel Properties dialog for an AutoCare Application Manager.

Vehicle Type Search Panel Properties [go to parent](#)

Application Object Types * AC_ACESApplication

Add... Remove Up Down

Search Box Types *

Make/Model Search Box (AC_Model / AC_Make)
Year Search Box (AC_VCdbYear / Year)
Options Group Search Box (And) (Options Search |
Part Type Search Box (Part Type)

Add... Edit... Remove Up Down

Vehicle Type Icon 🚗 Recreational and Sports

Name * AutoCare Recreational and Sports Vehicles

Root Nodes *

step://classification?id=AC_VehicleType_44
step://classification?id=AC_VehicleType_48

Add... Remove Up Down

Vehicle Object Types * AC_BaseVehicle

Add... Remove Up Down

Child Components

Node List <Select a child component> [go to component](#)

7. Once the required parameters are populated, click the **Save** and **Close** buttons to exit the Designer.

Many optional parameters are available within the Application Manager Screen Properties > Child Components > Node List.

Configuring Make/Model Search Box

Steps for adding and configuring the Make/Model search box type for a Vehicle Type Search Panel are below.

1. From the Vehicle Type Search Panel Properties dialog, click the **Add** button below the Search Box Types parameter field, and the Add Component dialog will display.

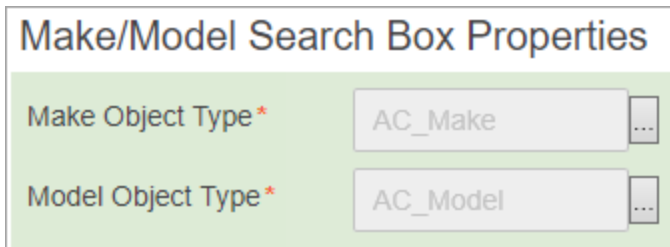
2. Click **Make/Model Search Box** from the list of components.
3. Click the **Add** button, and the Make/Model Search Box Properties will display two required fields, as shown below.

Make Object Type: Determines the Make objects that a user can select within the Make/Model search box. Click the ellipsis button (...) to find and select the Make Object Type for the Makes that exist in the vehicle classification (defined in the 'Root Nodes' parameter) and should be displayed for a user to select within the search box.

Model Object Type: Determines the Model objects that a user can select within the Make/Model search box. Click the ellipsis button (...) to find and select the Model Object Type for the Models that exist in the vehicle classification (defined in the 'Root Nodes' parameter) and should be displayed for a user to select within the search box.

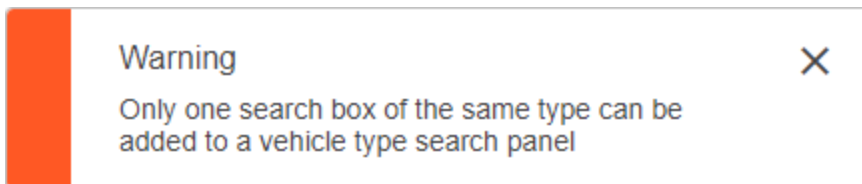
4. Click the ellipsis button (...) to select the Object Type for the makes that should display within the search box for a user to select.
5. Click the ellipsis button (...) to select the Object Type for the Models that should display within the search box for a user to select.

Below is an example of a Make/Model Search Box Properties dialog configured for the AutoCare standard.



6. Once the required parameters are populated, click the **Add** button to return to the Vehicle Type Search Panel Properties, and optionally add another search box type, or move on to the next required parameter.

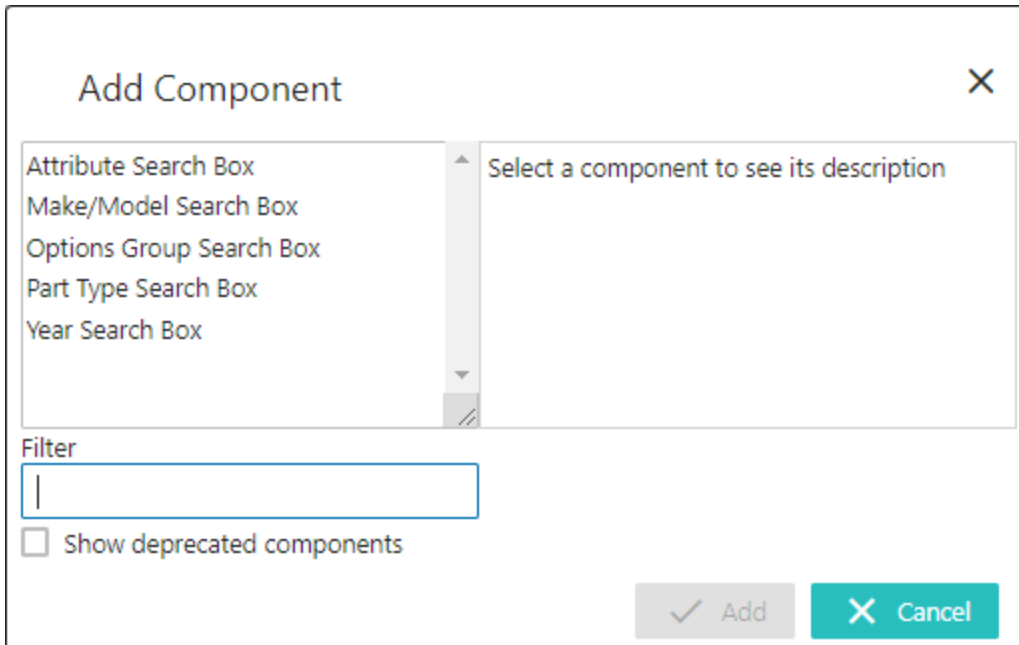
Because each search box type can be only be configured once for each Application Manager, if a second search box of the same type is added, a Warning dialog will display as shown below, and the Search Box Types parameter will remain unchanged.



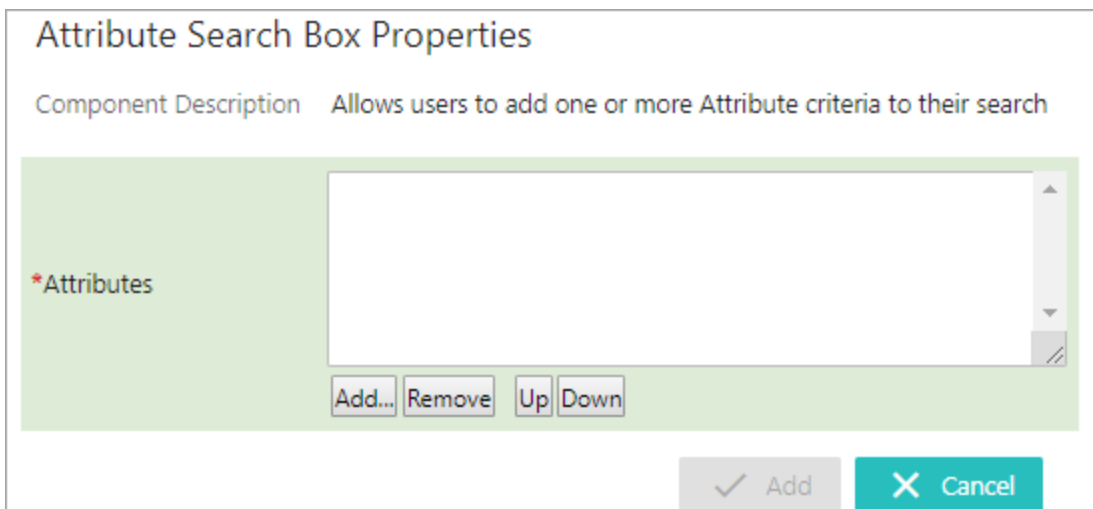
Configuring Attribute Search Box

Steps for adding the Attribute search box type to a Vehicle Type Search Panel are below.

1. From the Vehicle Type Search Panel Properties dialog, click the **Add** button below the Search Box Types parameter field, and the Add Component dialog will display.



2. Click **Attribute Search Box** from the list of components.
3. Click the **Add** button, and the Year Search Box Properties will display, as shown below.

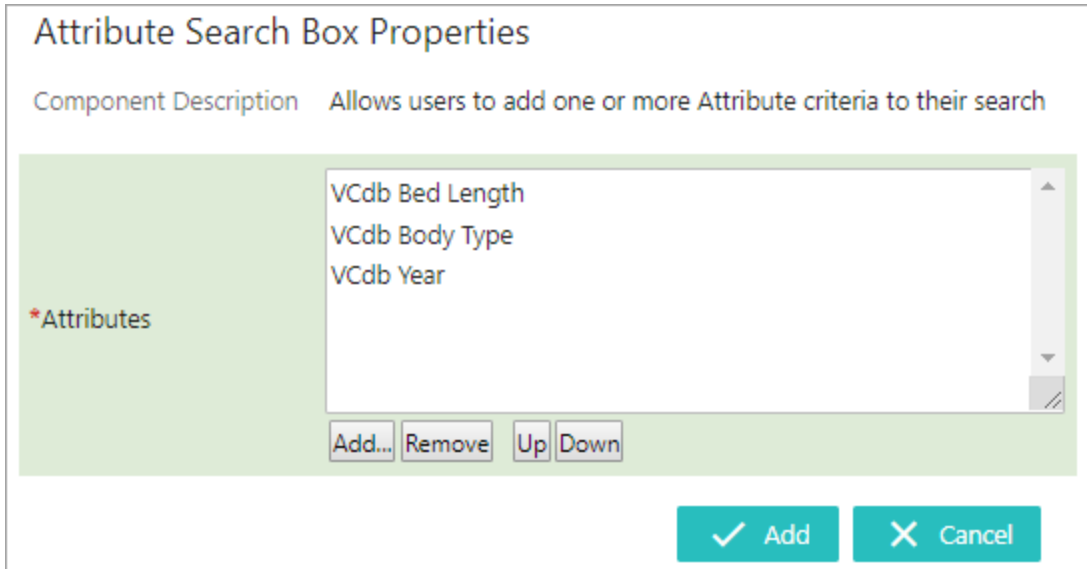


Attributes: This required field determines the attributes that allows users to add one or more attribute value criteria to their search. All attribute validation base types (e.g., Text, Number, List Of Values) are allowed. Click

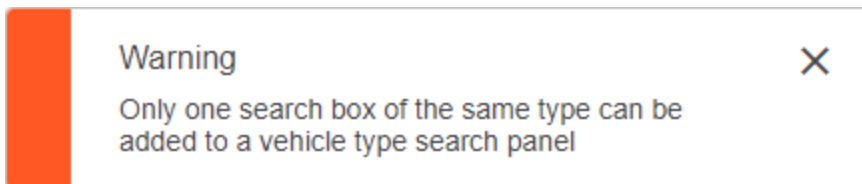
the Add button to find and select the required attributes. The selected attributes must be valid for the object type that is defined in the Vehicle Object Types parameter, and the attribute value must be populated on that object in order for it to display correctly.

- Once the required parameters are populated, click the **Save** button to return to the Vehicle Type Search Panel Properties, and optionally add another search box type, or move on to the next required parameter.

Below is an example of a configured Attribute search box type for the AutoCare standard.



Because each search box type can be only be configured once for each Application Manager, if a second search box of the same type is added, a Warning dialog will display as shown below, and the Search Box Types parameter will remain unchanged.

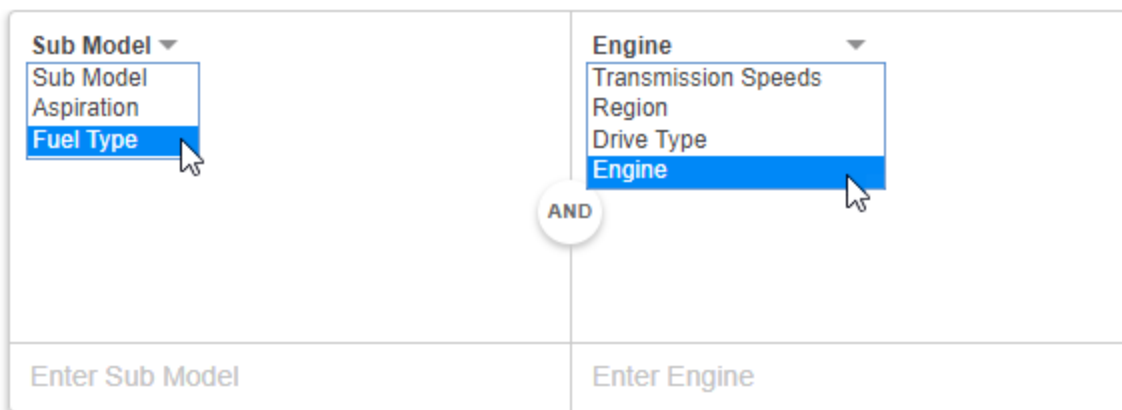


Configuring Options Group Search Box

Attributes or References within STEP can be displayed as Options within an Application Manager Options Search Box. The Option values (attribute and/or reference values) can then be selected as criteria for a search.

Though many options (attributes and/or references) can be added to an Options Search Box, only one Options Group search box type can be configured for each 'Vehicle Type Search Panel.' However, either one or two Options Search Boxes can be configured within the Options Group search box type. In other words, one Options Search Box can be configured to list multiple options or two Options Search Boxes can be configured. When two Options Search Boxes are configured, then an AND/OR toggle button displays between the two Options Search Boxes.

In the example below, an Options Group search box type is configured with two Options Search Boxes. Each Options Search Box is configured to display multiple attributes and/or References available for selection. In the screenshot below, the dropdown selections for both Options Search Boxes are displayed for the purposes of this example.



Note: It is possible to instead list all of the options within one Options Search Box, however users would lose the ability to apply the 'AND' function to their search. Additionally, each Options Search Box can be configured to list the same options as the other Options Search Box.

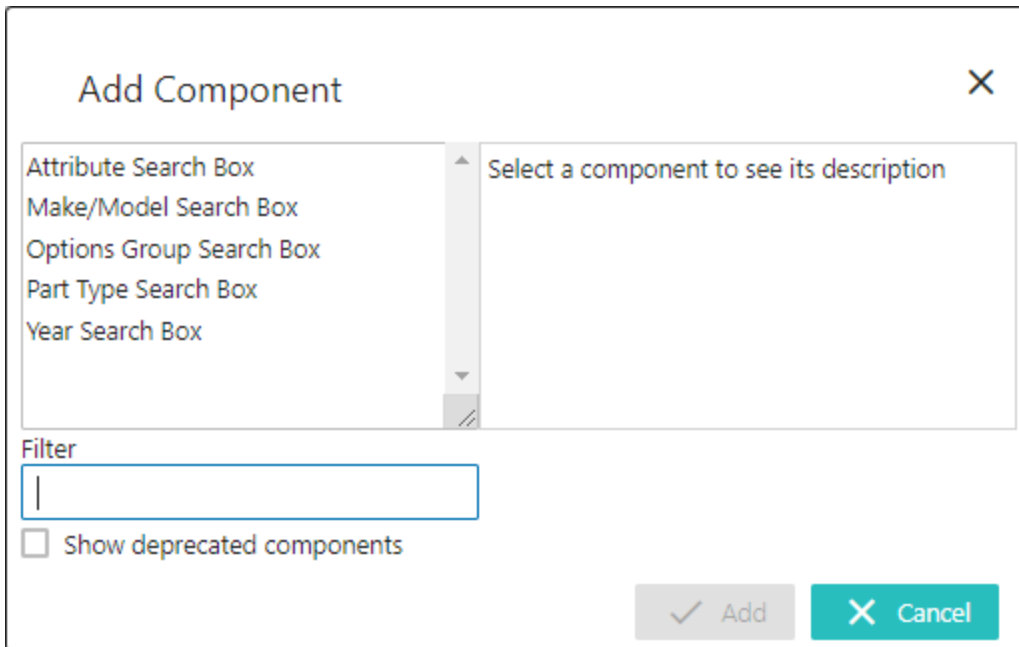
Prerequisites

Prior to configuring an Options Group Search Box, it is helpful to know one or more attributes and/or References that will be displayed as an option, along with the exact validation path for each. Setup requires an exact validation path for each of the attributes and/or References used. Though this topic provides example for each validation path type, more information about validation paths, can be found within the **Automotive Validation Path** topic of this guide.

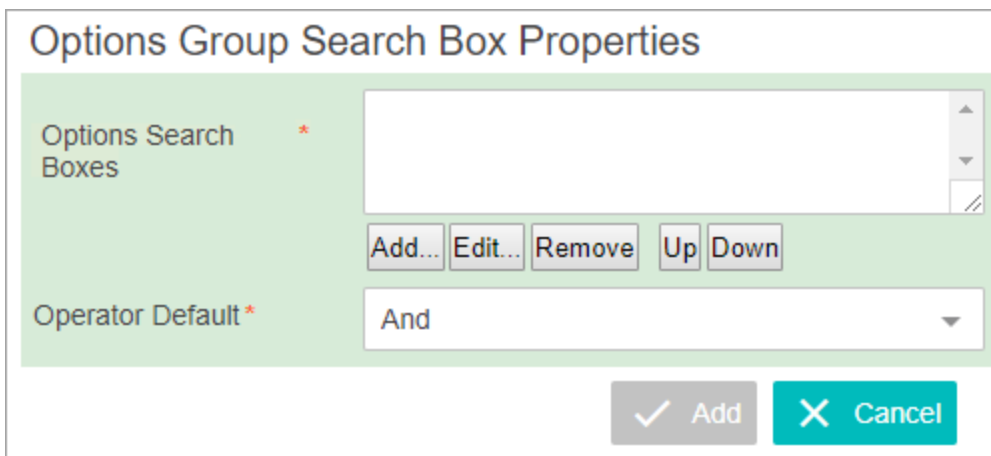
Configuring Options Group Search Box

Steps for adding and configuring the Options Group search box type for a Vehicle Type Search Panel are below.

1. From the Vehicle Type Search Panel Properties dialog, click the **Add** button below the Search Box Types parameter field, and the Add Component dialog will display.



2. Click **Options Group Search Box** from the list of components.
3. Click the **Add** button, and the Options Group Search Box Properties will display as shown below.



Within the Options Group Search Box Properties dialog, the following parameters are required:

Options Search Boxes: Allows either one or two Options search boxes to display within the Vehicle Type Search Panel. Each Options search box can be configured to allow users to choose from a dropdown list of options. At a maximum, only two Options Search Boxes can be added to an Options Group Search Box.

Operator Default: Allows the default AND / OR operator to be configured for two Options search boxes. By default, AND is selected.

Below is an example of an Options Group Search Box Properties dialog configured for the AutoCare standard.

Options Group Search Box Properties

Options Search Boxes *

Options Search Box (Sub Model) (Option (child.attribute[id:'AC_

Options Search Box (Engine) (Option (child.reference[type:'AC_

Add...
Edit...
Remove
Up
Down

Operator Default *

And

✓ Save
✗ Cancel

Because only two Options Search Boxes can be configured for each Option Group search box type, if a user tries to add a third Options Search Box, a Warning dialog will display as shown below.

Warning

A maximum of 2 option cards can be configured on an options group

✗

4. Click the **Add** button beneath the Options Search Boxes parameter, and the Options Search Box Properties will display as shown below.

Options Search Box Properties

Default Option *

Options *

Add...
Edit...
Remove
Up
Down

✓ Add
✗ Cancel

Within the Options Search Box Properties dialog, the following parameters are required:

Default Option: Allows for a default option to display when users access the Application Manager. The value within this parameter must match the 'Label' parameter configured within the Options parameter listed below this parameter.

Options: Allows for one or more attributes or References to be displayed as Options within the Options Search Box. Once an Option is selected, the typeahead dropdown field will display only values that pertain to the selected option. When more than one Options search box is used, the attributes and/or reference Options configured within one Options search box can optionally be configured to display in the second Options search box.

Below is an example of an Options Search Box Properties dialog configured for the AutoCare standard.

Options Search Box Properties

*Default **
Option

Sub Model

*Options **

- Criterion Option (child.reference[type:'AC_VehicleToEngineConfig
- Criterion Option (child.attribute[id:'AC_VCdbSubmodel'] / Sub Mod**
- Criterion Option (child.attribute[id:'AC_VCdbRegion'] / Region)
- Criterion Option (child.reference[type:'AC_VehicleToEngineConfig
- Criterion Option (child.attribute[id:'AC_VCdbPublicationStage'] / P

Add... Edit... Remove Up Down

5. Click the **Add** button to add an option to an Options Search Box, and the Option Properties dialog will display two required parameters:

Option Properties

*Label **

*Validation Path **

✓ Add X Cancel

Within the Option Properties dialog, the following parameters are required:

Label: Provide a label that best describes the option being added to the Options Search Box. The value can be used within the 'Default Option' parameter on the previous dialog.

Validation Path: Provide a validation path for the option being configured. The option should correlate to an attribute or reference. It is recommended to copy the 'Automotive Validation Path' value for the attribute or

reference from the workbench, and paste it into the Validation Path parameter field. Examples for each type of validation path can be found below.

- Attribute Validation Path Example:

Below is an example of a Option Properties dialog configured for the ACES attribute, Sub Model.

Option Properties

<i>Label*</i>	<input type="text" value="Sub Model"/>
<i>Validation Path*</i>	<input type="text" value="child.attribute[id:'AC_VCdbSubmodel']"/>

The Validation Path is configured by the Easy Setup and displayed in the workbench. For this example, the Validation Path for the ACES Sub Model can be found by navigating to System Setup > Attribute Groups > AutoCare Attributes > AutoCare ACES Attributes > ACES Sub Model.

ACES Sub Model - Attribute	
Attribute	References
Description	
Name >>	Value >
> ID	AC_ACESSubModel
> Name	Sub Model
> Last edited by	2017-11-17 14:14:33 by EASYSETUP
> Full Text Indexable	No
> Externally Maintained	No
> Completeness Score	
> Hierarchical Filtering	None
> Calculated	No
> Type	Specification
> Automotive Validation Path abc	child.attribute[id:'AC_VCdbSubmodel'] ←

- Reference Validation Path Example:

Below is an example of how to configure the Option Properties for the ACES reference, ACES Application To Engine Base.

Option Properties

Label *	<input type="text" value="Engine"/>
Validation * Path	<input type="text" value="child.reference[type:'AC_VehicleToEngineConfig'].reference[type:'AC_EngineConfigToEngineBase'].ID"/>

The Automotive Validation Path for the **ACES Application To Engine Base** reference type can be found in the workbench by navigating to System Setup > Reference Types > Classification Reference Types > ACES Application To Engine Base.

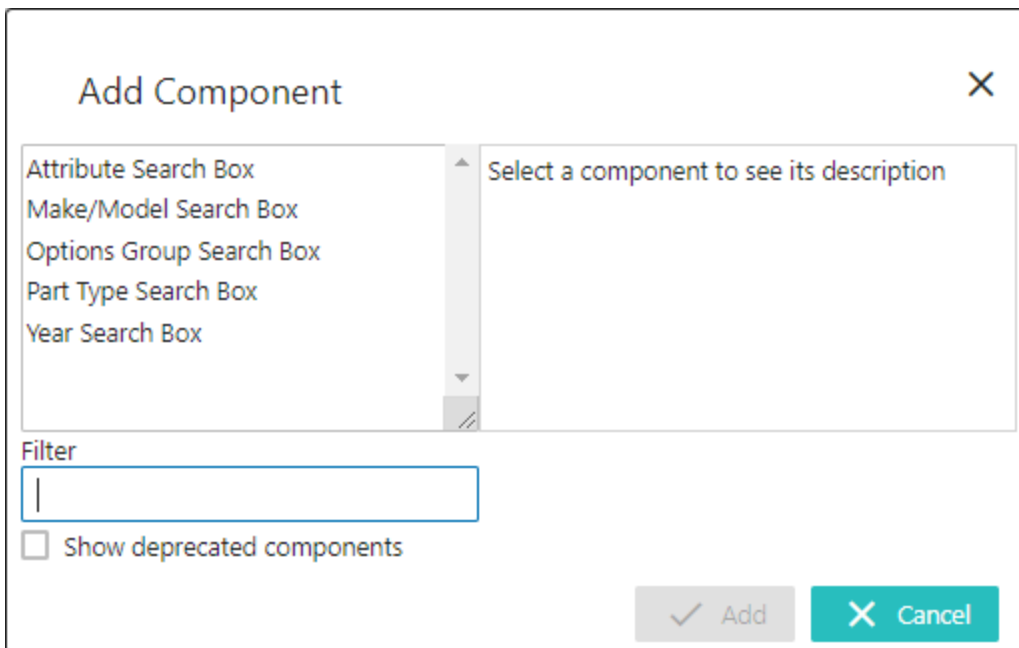
← ACES Application To Engine Base - Reference Type →		
Reference Type	Validity	Log
🔍 Description		
Name	>	Value >
> ID		AC_ACESApplicationToEngineBase
> Name		ACES Application To Engine Base
> Last edited by		2018-04-05 13:53:26.0 by EASYSETUP
> Externally Maintained		No
> Allow multiple references		No
> Mandatory		No
> Inheritance		None
> Automotive Validation Path	abc	child.reference[type:'AC_VehicleToEngineConfig'].reference[type:'AC_EngineConfigToEngineBase'].ID

6. Once the required parameters are populated, click the **Add** button to return to the Options Search Box Properties, and optionally add additional options for the Options Search Box, or click the **Add** button and return to the Options Group Search Box Properties.
7. Optionally repeat the steps above to add an additional Options Search Box, or click the **Add** button to return to the Vehicle Type Search Panel Properties.
8. Once the required parameters are populated, click the **Add** button to return to the Vehicle Type Search Panel Properties, and optionally add another search box type, or move on to the next required parameter.

Configuring Part Type Search Box

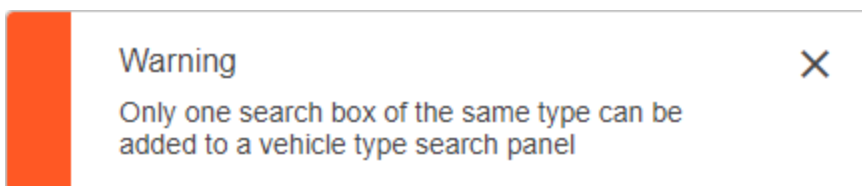
Because the Part Type search box type does not require additional configuration, it can quickly be added to the Search Box Types parameter within the Vehicle Type Search Panel Properties dialog by following the steps listed below:

1. From the Vehicle Type Search Panel Properties dialog, click the **Add** button below the Search Box Types parameter field, and the Add Component dialog will display.



2. Click **Part Type Search Box** from the list of components.
3. Click the **Add** button, and the Vehicle Type Search Panel Properties will display with the Part Type Search Box added to the Search Box Types parameter.

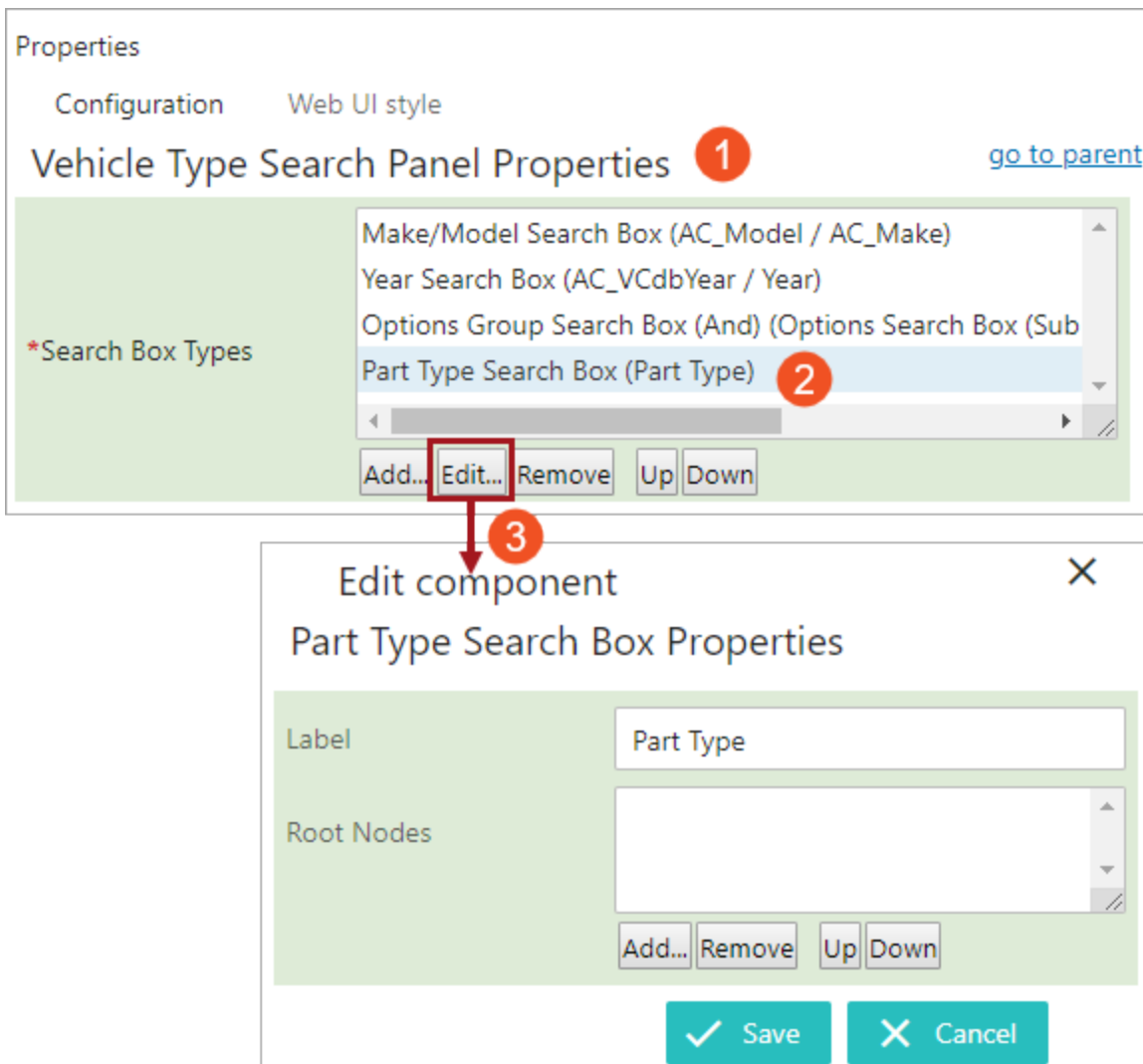
Because each search box type can be only be configured once for each Application Manager, if a second search box of the same type is added, a Warning dialog will display as shown below, and the Search Box Types parameter will remain unchanged.



Editing the Part Type Search Box Label

The Part Type search box label displayed within the Application Manager can be edited by following the steps listed below:

1. Go to the Vehicle Type Search Panel Properties for the Application Manager.
2. Within the Search Box Types parameter, select the **Part Type Search Box**.
3. Click the **Edit** button, and the Part Type Search Box Properties dialog will display as shown below.
4. Edit the following parameters as necessary.
 - **Label:** By default, the Label parameter is populated as 'Part Type.' Optionally edit the parameter to a more suitable label.
 - **Root Nodes:** Allows for the restriction of root nodes to be used within the Part Type search box when searching / filtering for a Part Type. Once the root node(s) is defined, the search input field displays only suggestions of objects that belong in the root node defined within this parameter. Additionally, when the Node Picker is used, its dialog will display the only those classification folders that pertain to the root nodes defined within this parameter. Click the Add button below the parameter field to browse or search for the necessary vehicle root node object(s).

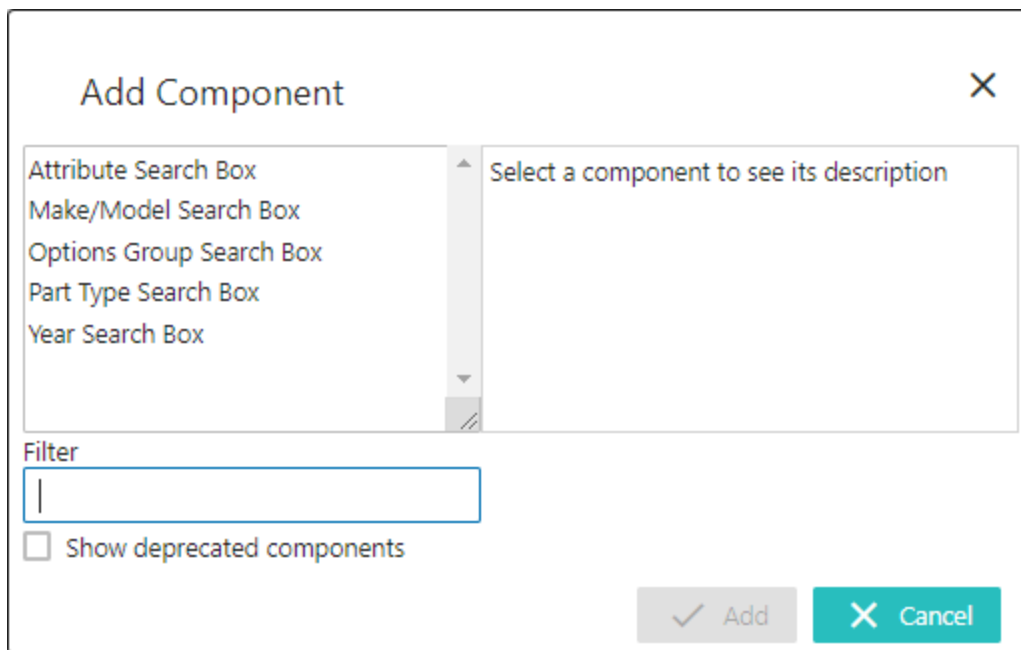


Configuring Year Search Box

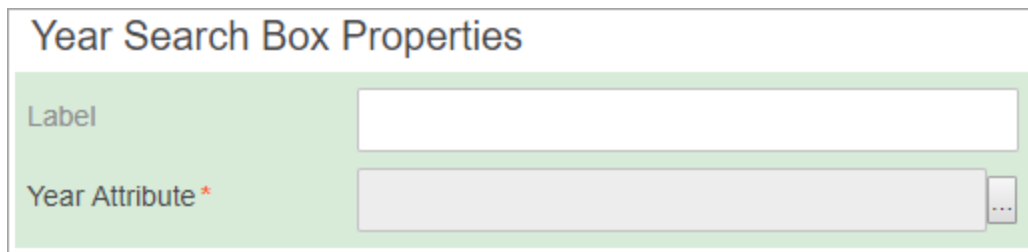
Important: With effective from Automotive 9.1 MP2 patch, the Year Search Box component has been superseded by the Attribute Search Box component and therefore may be removed in a future release. It is recommended that users transition to using Attribute Search Box component, which are described in the **Configuring Attribute Search Box** documentation.

Steps for adding the Year search box type to a Vehicle Type Search Panel are below.

1. From the Vehicle Type Search Panel Properties dialog, click the **Add** button below the Search Box Types parameter field, and the Add Component dialog will display.



2. Click **Year Search Box** from the list of components.
3. Click the **Add** button, and the Year Search Box Properties will display, as shown below.



Label: Optionally, provide a label for the Year search box that best communicates to the user what criteria are available for selection within the Application Manager. By default, the label is set to 'Year.'

Year Attribute: This required field determines the Year Attribute that allows users to add one or more year criteria to their search. All attribute validation base types (e.g., Text, Number, List Of Values) are allowed. Click the ellipsis button (...) to find and select the necessary Year Attribute. The selected attribute must be valid for the object type that is defined in the Vehicle Object Types parameter, and the Year Attribute value must be populated on that object in order for it to display correctly.

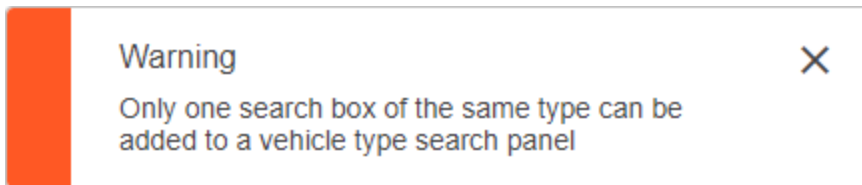
- Once the required parameters are populated, click the **Save** button to return to the Vehicle Type Search Panel Properties, and optionally add another search box type, or move on to the next required parameter.

Below is an example of a configured Year search box type for the AutoCare standard.

Year Search Box Properties

Label	<input style="width: 95%;" type="text" value="Year"/>
Year Attribute *	<input style="width: 95%;" type="text" value="AC_VCdbYear"/> ...

Because each search box type can be only be configured once for each Application Manager, if a second search box of the same type is added, a Warning dialog will display as shown below, and the Search Box Types parameter will remain unchanged.



Competitor and OE Numbers Solution

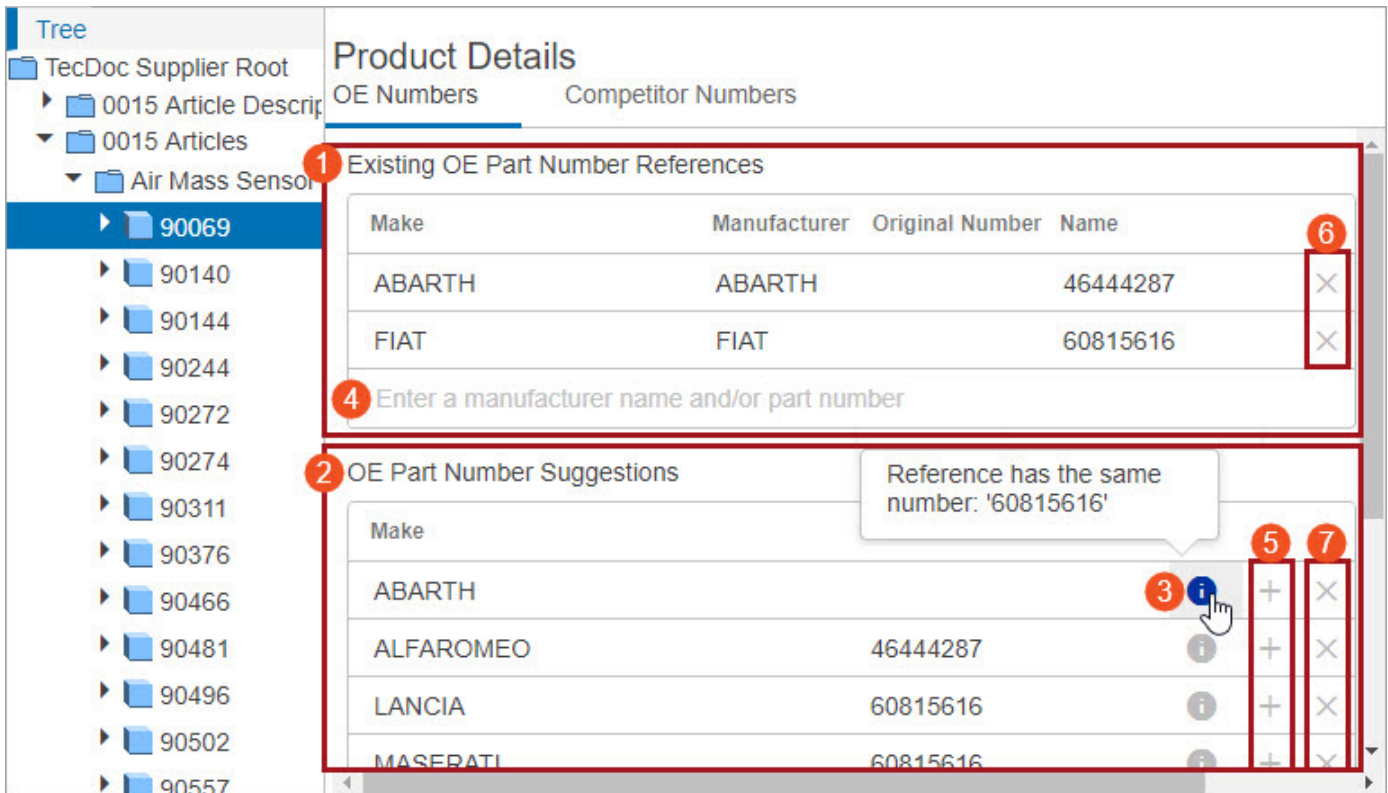
A competitor number refers to the part number used by a competitor. Whereas an OE number refers to the original parts used by the original equipment manufacturer to assemble a vehicle at the factory. This number is only unique within parts from that manufacturer. For example, an OE number should always be displayed with the OEM (Original Equipment Manufacturer) name behind the number. This means that ABC123 AUDI will usually not be the same part as ABC123 BMW. But on the other hand, they could be the same part if both manufacturers acquired their part from the same supplier.

With the automotive solution, it is easy to maintain data related to both the competitor part number and OE number. The automotive solution not only helps users more easily view, edit, and create competitor and/or OE part number references for specific parts, but also suggest potential matches. In other words, when configured, STEP can suggest potential matches for competitor and/or OE part numbers based on existing competitor and/or OE part number references and/or matching part numbers within STEP. This occurs when a part is selected, STEP looks at the configured Reference Types and displays their targets, attribute values, and/or metadata via configurations within the 'Part Number References' Node Editor component.

When the Part Number References and/or Part Number Suggestions components are configured as shown below, Web UI users can:

1. View existing competitor and/or OE part number references and related data.
2. View competitor and/or OE part number reference suggestions and related data.
3. View information about the suggested part number.
4. Add competitor and/or OE part number references manually.
5. Add competitor and/or OE part number references from suggestions.
6. Remove existing competitor and/or OE part number references.
7. Remove competitor and/or OE part number references from suggestions.

Note: The components used for this solution are extremely flexible and can be configured to display both the competitor and OE numbers on the same screen. Also, additional column headers can be added to the components to display more information than is shown in this example.



In the example above, part 90069 is selected within a Web UI Tree, and a Web UI screen (Product Details) is configured to display the competitor and OE numbers on separate tabs (OE Numbers, Competitor Numbers). Though only the content for the 'OE Numbers' tab is displayed below, the layout and features on both tabs are the same.

This section addresses:

- Prerequisites
- Adding Part Number References Manually
- Adding Part Number References from Suggestions
- Removing Existing Part Number References
- Removing Part Number Reference Suggestions
- Creating and Referencing New Competitor and/or OE Part Numbers
- Configuring the Part Number References Component
- Configuring the Part Number Suggestions Component
- Configuring the Advanced Part Number References Component
- Configuring the Advanced Part Number Suggestions Component
- Configuring the Data Model for Application Competitor or OE Part Numbers Table Header

- Configuring the Application Competitor or OE Part Numbers Table Header Component
- Configuring the Metadata Attribute References Component

Prerequisites

When implementing this solution, it is suggested to create a 'Product Details' screen within a Web UI and configure it to display when a product is selected within the Tree navigator. The Product Details screen should be created via a Node Details screen with a Tab Control that is configured to display a 'Competitor Numbers' and an 'OE Numbers' Tab Page. Within each of these Tab Pages a Node Editor can be configured with 'Part Number References' and/or 'Part Number Suggestions' components. The 'Part Number References' component can be configured to display competitor and/or OE number data. Whereas, the 'Part Number Suggestions' component can be configured to display suggested competitor and/or OE part numbers.

For more details on configuring screen mappings so that a 'Product Details' screen will display when a product is selected within the Tree navigator, see the **Mappings** topic within the **Main Properties Overview** section of Web User Interfaces / Web UI Setup and User Guide in **STEP Online Help**.

Adding Part Number References Manually

Users can easily add an existing competitor and/or OE part number reference to a selected part within Web UI. When the Part Number References component is configured, and a part is selected, STEP looks at the configured Reference Types and displays their targets, attribute values, and/or metadata via configurations within the 'Part Number References' Node Editor component. Whether or not references are displayed, at a minimum a blank typeahead field will display prompting users to 'Enter a manufacturer name and/or part number.' This typeahead field can be used to manually add a reference from an existing part number to the selected part.

In the example below, part 90069 is selected within the Web UI Tree, and an 'OE Numbers' tab is configured to display results for 'Existing OE Part Number References.' Within the results, two Make / Manufacturer part numbers are displayed, along with the blank typeahead field that prompts users to 'Enter a manufacturer name and/or part number.'

The screenshot shows the 'Product Details' page for part 90069. The 'OE Numbers' tab is selected, showing a table of existing OE part number references. The table has the following data:

Make	Manufacturer	Original Number	Name
ABARTH	ABARTH	46444287	✕
FIAT	FIAT	60815616	✕

Below the table is a typeahead field with the placeholder text: "Enter a manufacturer name and/or part number".

Note: Though a new part number can be created using the Create New Part dialog, a new manufacturer name cannot be created.

When the manufacturer name and part number exist in STEP, users can type in the manufacturer name [space bar] part number to easily find the desired part number. Once a manufacturer name is typed into the typeahead field, available part numbers for the manufacturer will display in the dropdown list. Clicking a listed part number will add the part number to the top of the references list.

In the example below, 'Lancia 46444' is typed into the typeahead field and the Lancia part numbers that begin with 46444 display within the dropdown list.

Product Details

OE Numbers Competitor Numbers

Existing OE Part Number References

Make
ABARTH
FIAT
<u>lancia</u> 46444

LANCIA

- ├ 46444284
- └ 46444287

Clicking on 46444287 adds the part number to the top of the references list, as shown below.

Product Details

OE Numbers Competitor Numbers

Existing OE Part Number References

Make	Manufacturer	Original Number	Name
LANCIA	LANCIA	46444287	×
ABARTH	ABARTH	60815616	×
FIAT	FIAT	60815616	×

Enter a manufacturer name and/or part number

If the Part Number Suggestions component is configured below the references list, then the suggestions list will update as new part numbers are added to the references list.

Adding Part Number References from Suggestions

Users can add an existing competitor and/or OE part number reference to a selected part from the Part Number Suggestions list within Web UI. When the Part Number References and the Part Number Suggestions components are configured, and a part is selected, STEP looks at the configured Reference Types and along with displaying their targets, attribute values, and/or metadata via configurations, it also suggests potential part numbers as suggestions within the configured Node Editor component.

When any OE Number / Competitor Number is displayed in the Part Number Suggestions list, an add button (+) displays alongside each OE Number / Competitor Number. Clicking the add button will add an OE Number / Competitor Number to the Part Number References list.

Clicking the remove button (X) will remove the OE Number / Competitor Number reference from the Part Number References list / Part Number Suggestions list. For more information, see the **Removing Existing Part Number References** and the **Removing Part Number Reference Suggestions** topics within this section.

In the example below, part '90069' is selected within the Web UI Tree, and the 'Competitor Numbers' tab is configured to display results for 'Existing Competitor Part Number References' and 'Competitor Part Number Suggestions.' Within the Competitor Part Number Suggestions list, two Manufacturer part numbers display along with the (+) buttons.

Tree	Node Details																									
<ul style="list-style-type: none"> ▼ TecDoc Supplier Root <ul style="list-style-type: none"> ▶ 0011 BERU Product Hierarchy ▼ 0015 NGK Product Hierarchy <ul style="list-style-type: none"> ▶ 0015 Addresses ▶ 0015 Article Descriptions ▼ 0015 Articles <ul style="list-style-type: none"> ▼ Air Mass Sensor <ul style="list-style-type: none"> ▶ 90069 ▶ 90140 ▶ 90144 ▶ 90244 ▶ 90272 ▶ 90274 ▶ 90311 ▶ 90376 ▶ 90466 	<div style="display: flex; justify-content: space-around; border-bottom: 1px solid #ccc;"> OE Numbers <u>Competitor Numbers</u> </div> <div style="border: 1px solid #ccc; padding: 5px;"> <p>Existing Competitor Part Number References</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 40%;">Manufacturer</th> <th style="width: 40%;">Name</th> <th style="width: 20%;"></th> </tr> </thead> <tbody> <tr> <td>HELLA</td> <td>8ET009142421</td> <td style="text-align: right;">X</td> </tr> <tr> <td>INTERMOTOR</td> <td>19370</td> <td style="text-align: right;">X</td> </tr> <tr> <td>VDO</td> <td>5WK97001Z</td> <td style="text-align: right;">X</td> </tr> <tr> <td colspan="3" style="text-align: center; padding: 5px;">Enter a manufacturer name and/or part number</td> </tr> </tbody> </table> <p>Competitor Part Number Suggestions</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 40%;">Manufacturer</th> <th style="width: 40%;">Name</th> <th style="width: 20%;"></th> </tr> </thead> <tbody> <tr> <td>FEBIBILSTEIN</td> <td>19370</td> <td style="text-align: right;">i + X</td> </tr> <tr> <td>MAPCO</td> <td>19370</td> <td style="text-align: right;">i + X</td> </tr> </tbody> </table> </div>		Manufacturer	Name		HELLA	8ET009142421	X	INTERMOTOR	19370	X	VDO	5WK97001Z	X	Enter a manufacturer name and/or part number			Manufacturer	Name		FEBIBILSTEIN	19370	i + X	MAPCO	19370	i + X
Manufacturer	Name																									
HELLA	8ET009142421	X																								
INTERMOTOR	19370	X																								
VDO	5WK97001Z	X																								
Enter a manufacturer name and/or part number																										
Manufacturer	Name																									
FEBIBILSTEIN	19370	i + X																								
MAPCO	19370	i + X																								

Clicking the (+) button to the right of the 'FEBIBILSTEIN 19370' suggestions list will add the 'FEBIBILSTEIN 19370' part to the Existing Competitor Part Number References list (as shown below).

Tree

- ▼ TecDoc Supplier Root
 - ▶ 0011 BERU Product Hierarchy
 - ▼ 0015 NGK Product Hierarchy
 - ▶ 0015 Addresses
 - ▶ 0015 Article Descriptions
 - ▼ 0015 Articles
 - ▼ Air Mass Sensor
 - ▶ 90069
 - ▶ 90140
 - ▶ 90144
 - ▶ 90244
 - ▶ 90272
 - ▶ 90274
 - ▶ 90311
 - ▶ 90376
 - ▶ 90466

Node Details

OE Numbers
Competitor Numbers

Existing Competitor Part Number References

Manufacturer	Name	
HELLA	8ET009142421	✕
INTERMOTOR	19370	✕
VDO	5WK97001Z	✕
FEBIBILSTEIN	19370	✕

Enter a manufacturer name and/or part number

Competitor Part Number Suggestions

Manufacturer	Name	
MAPCO	19370	i + ✕

Once the part number reference to the selected part is created and now displays within the Part Number References table, the OE Number / Competitor part number will no longer display within the Part Number Suggestions list.

Removing Existing Part Number References

Users can remove an existing competitor and/or OE part number reference from a selected part within Web UI. When the Part Number References component is configured, and a part is selected, a remove button (X) displays to the right of each OE Number / Competitor Number. Clicking the remove (X) button will remove the OE Number / Competitor Number reference from the Part Number References list, and therefore the number will no longer display within the list.

In the example below, part '90069' is selected within the Web UI Tree, and an 'OE Numbers' tab is configured to display results for 'Existing OE Part Number References.' Within the results, two Make / Manufacturer part numbers are displayed along with the remove (X) buttons that allow users to remove the part number reference from the selected part.

The screenshot shows a tree view on the left with '90069' selected under 'Air Mass Sensor'. The main panel is titled 'Product Details' and has two tabs: 'OE Numbers' (active) and 'Competitor Numbers'. Below the tabs is a section titled 'Existing OE Part Number References' containing a table with the following data:

Make	Manufacturer	Original Number	Name
ABARTH	ABARTH	46444287	X
FIAT	FIAT	60815616	X

Below the table is a text input field with the placeholder text: 'Enter a manufacturer name and/or part number'.

Clicking the remove (X) button to the right of the 'Fiat 60815616' part number removes the part number reference and the part number no longer displays within the Existing OE Part Number Reference grid (as shown below).

The screenshot shows the same tree view with '90069' selected. The 'Product Details' page is still on the 'OE Numbers' tab. The 'Existing OE Part Number References' table now only contains one entry:

Make	Manufacturer	Original Number	Name
ABARTH	ABARTH	46444287	X

The text input field 'Enter a manufacturer name and/or part number' is still present below the table.

If an OE Number / Competitor part number is removed from the Part Number References list, and it previously displayed within the Part Number Suggestions list, then the unlinked OE Number / Competitor part number will display within the suggestions list. However, if the removed part number was not previously displayed within the suggestions list (i.e., manually added), then it will not display within the suggestions list.

Removing Part Number Reference Suggestions

Users can remove an existing competitor and/or OE part number reference suggestions from a selected part within Web UI. When the Part Number Suggestions component is configured, and a part is selected, a remove button (X) button displays to the right of each OE Number / Competitor Number. Clicking the remove button will remove the OE Number / Competitor Number reference from the Part Number Suggestions list, and therefore the number will no longer display within the list.

In the example below, part '90069' is selected within the Web UI Tree, and the 'Competitor Numbers' tab is configured to display results for 'Existing Competitor Part Number References' and 'Competitor Part Number Suggestions.' Within the results, in the 'Competitor Part Number Suggestions' list, two Make / Manufacturer part numbers are displayed along with the remove (X) button that allows users to remove a part number reference of a suggested part number from the selected part.

Tree

- ▼ TecDoc Supplier Root
 - ▶ 0011 BERU Product Hierarchy
 - ▼ 0015 NGK Product Hierarchy
 - ▶ 0015 Addresses
 - ▶ 0015 Article Descriptions
 - ▼ 0015 Articles
 - ▼ Air Mass Sensor
 - ▶ **90069**
 - ▶ 90140
 - ▶ 90144
 - ▶ 90244
 - ▶ 90272
 - ▶ 90274
 - ▶ 90311
 - ▶ 90376
 - ▶ 90466

Node Details

OE Numbers Competitor Numbers

Existing Competitor Part Number References

Manufacturer	Name	
HELLA	8ET009142421	X
INTERMOTOR	19370	X
VDO	5WK97001Z	X
Enter a manufacturer name and/or part number		

Competitor Part Number Suggestions

Manufacturer	Name			
FEBIBILSTEIN	19370	i	+	X
MAPCO	19370	i	+	X

In the example, after removing the reference of part number 'MAPCO 19370' from part '90069' the Competitor Part Number Suggestions list displays as shown below.

Tree

- ▼ ■ TecDoc Supplier Root
 - ▶ ■ 0011 BERU Product Hierarchy
 - ▼ ■ 0015 NGK Product Hierarchy
 - ▶ ■ 0015 Addresses
 - ▶ ■ 0015 Article Descriptions
 - ▼ ■ 0015 Articles
 - ▼ ■ Air Mass Sensor
 - ▶ ■ 90069
 - ▶ ■ 90140
 - ▶ ■ 90144
 - ▶ ■ 90244
 - ▶ ■ 90272
 - ▶ ■ 90274
 - ▶ ■ 90311
 - ▶ ■ 90376
 - ▶ ■ 90466
 - ▶ ■ 90481

Node Details

OE Numbers
Competitor Numbers

Existing Competitor Part Number References

Manufacturer	Name	
HELLA	8ET009142421	✕
INTERMOTOR	19370	✕
VDO	5WK97001Z	✕
Enter a manufacturer name and/or part number		

Competitor Part Number Suggestions

Manufacturer	Name	
FEBIBILSTEIN	19370	i + ✕

If the Part Number is removed from the Part Number Suggestions list, then the unlinked OE Number / Competitor part number will no longer display within the Part Number Suggestions list.

Creating and Referencing New Competitor and OE Part Numbers

When the Part Number References component is configured, Web UI users can create new competitor and/or OE part numbers for existing manufacturers and simultaneously reference them to a selected part.

When a part is selected within the Web UI Tree, STEP looks at the configured Reference Types and displays their targets, attribute values, and/or metadata via configurations within the 'Part Number References' Node Editor component. Whether or not references are displayed, at a minimum the blank typeahead field will display, prompting users to 'Enter a manufacturer name and/or part number.' This typeahead field can be used to access the 'Create New Part' dialog where users can create a new part number for an existing manufacturer and reference it to the selected part. This typeahead field can also be used to quickly select an existing part number to be referenced. For more information, see the **Adding Part Number References Manually** topic.

In the example below, part '90069' is selected within the Web UI Tree, and an 'OE Numbers' tab is configured to display results for 'Existing OE Part Number References.' Within the results, two part numbers display with different Makes and Manufacturers, along with the blank typeahead field that prompts users to 'Enter a manufacturer name and/or part number.'

Make	Manufacturer	Original Number	Name
ABARTH	ABARTH	46444287	X
FIAT	FIAT	60815616	X

Enter a manufacturer name and/or part number

Note: Though a new part number can be created using the Create New Part dialog, a new manufacturer name cannot be created.

When the manufacturer name exists in STEP, but the part number does not, users can type at least two letters of the beginning of an existing manufacturer name, and the dropdown list will display a list of manufacturers and their part numbers.

In the example below, 'lan' is typed in and the typeahead suggestion list displays the matching manufacturer name, part numbers, and a 'Create new' button.

Product Details

OE Numbers Competitor Numbers OE Applications

Existing OE Part Number References

Make	Manufacturer	Original Nu...	Name
ABARTH	ABARTH		46444287 ✕
FIAT	FIAT		60815616 ✕

lan

LANCIA

- └ 0046480309
- └ 0046480362
- └ 22451AA060

Clicking on a manufacturer name in the dropdown list, or using the arrow keys to navigate to the desired manufacturer name and pressing the Enter key, or clicking the 'Create new' button will display the Create New Part dialog with the Manufacturer Name field pre-populated (as shown below).

Create New Part ✕

Manufacturer Name*

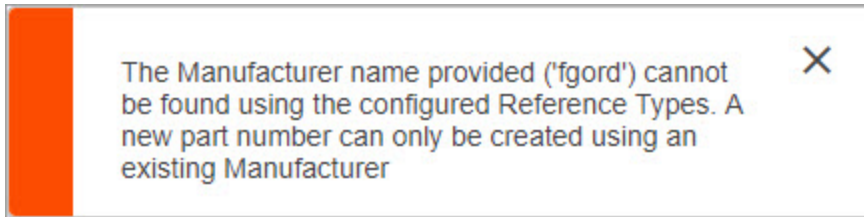
Part Number*

Once the Create New Part dialog displays, the Part Number field must be populated before the OK button will enable.

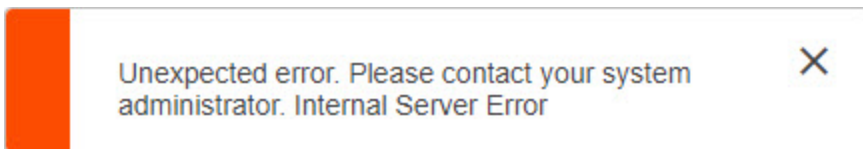
Optionally, the text within the pre-populated Manufacturer Name field can be edited. Typing text into this typeahead field will display a list of existing manufacturer names that begin with the provided text.

Once a part number is typed into the Part Number field and the OK button is clicked, the new part number is added to STEP and displays at the top of the references list.

If the Manufacturer name cannot be found, a warning dialog will display, as shown below.



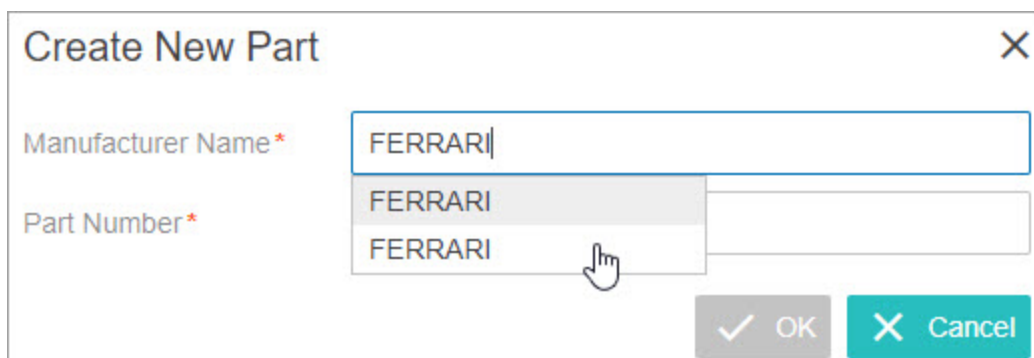
Additionally, if the attribute configured within the 'Original Number Attribute' parameter is not valid for the object type used for the new part number created, a warning dialog will display (as shown below). For more information, see the 'Original Number Attribute' parameter within the **Configuring the Part Number References Component** topic.



If the Part Number References list / Part Number Suggestions list is configured to display attributes of Competitor / OE Part Number, those attributes values (whether single valued or multi-valued) can only be displayed and cannot be edited in the Part Number References list / Part Number Suggestions list. To edit any Competitor / OE Part Number attribute value, users should navigate to the Competitor / OE Part Number using the Web UI tree.

When more than one manufacturer with same manufacturer name exists in STEP but different IDs, the typeahead field cannot determine the actual manufacturer and so does not auto populate. In this case, the system assumes the value should be created as the new part number. When users access the 'Create New Part' dialog, they can choose the desired manufacturer and type in the correct part number.

For example, let us consider that there are two existing manufacturers with the name 'Ferrari' present in STEP. STEP cannot auto populate the Manufacturer Name field with 'Ferrari.' However, within the 'Create New Part' dialog, users can retype the manufacturer name and select the correct instance of 'Ferrari.'



Note: To ensure users are able to easily distinguish the correct manufacturer name, be sure the manufacturer names within STEP are unique.

Normalization / Harmonization of Part Numbers

When the Create New Part dialog is used, part numbers created using special characters, spaces, and/or non-ASCII characters are normalized / harmonized to use alphanumeric characters only. The original part number can be stored (and searched for) within a separate attribute value configured within the Part Number References 'Original Number Attribute' parameter. For more information, see the **Configuring the Part Number References Component** topic.

In the example below, a new part number '12ET00@#12/*2' is created and then normalized / harmonized to '12ET00122.' The actual value '12ET00@#12/*2' is retained in the attribute 'ATTR_Original.'

The screenshot shows the 'Create New Part' dialog box with the following fields:

- Manufacturer Name*: FORD
- Part Number*: 12ET00@#12/*2
- Buttons: OK, Cancel

Below the dialog is the 'OE number references' table. A red arrow points from the 'Part Number' field in the dialog to the 'Name' column in the table.

Manufacturer	Name	ATTR_Original	
FORD	12ET00122	12ET00@#12/*2	×
FORD	1419677		×
CITRON	1920GN		×
PEUGEOT	1920GN		×
LANDROVER	PHF500090		×
JAGUAR	C2C12919		×
FORD	4R8Q12B579AC		×
Enter a manufacturer name and/or part number			

The original number can sometimes have illegal characters or spaces, the original value can be stored but a normalized value can also be stored and displayed along with the original value.

Once a new competitor and/or OE part number is created, a business action can be executed if configured. The business action must be valid for the following standards: AutoCare, TecDoc, and NAPA.

Configuring the Part Number References Component

The Part Number References component is a Node Editor component used to display referenced part numbers and their attribute and/or metadata values when a part number is selected within the Tree navigator.

This section describes how to configure the Part Number References component. For information about how to use this component, and prerequisites, see the **Competitor and OE Numbers Solution** topic.

Prerequisites

Anyone configuring the Part Number References Component within a Node Editor is expected to be familiar with the basic configuration and functionalities of Node Editor Component as basic concepts for working with Node Editor component are not covered in this section. For more information, see the **Node Editor Component** topic within the **Using a Web UI** section of **STEP Online Help**.

Configuration Steps

Below are the steps to configure the Part Number References component within a Node Editor. Each of the parameters for the Part Number References Properties are described below.

1. Using the Designer, go to Node Editor Properties Child Components Rows parameter for the screen that needs to display competitor and/or OE number information.
2. Click the **Add** button for the Child Components Rows parameter, and the Add Component dialog will display.
3. Find and select the **Part Number References** component, click the **Add** button, and the Part Number References Properties dialog will display (as shown below).

Part Number References Properties

Business Action to Invoke ... Clear

Table Headers * ...

Add... Edit... Remove Up Down

New Number Prefix *

Original Number Attribute ... Clear

Reference Types *

Add... Remove Up Down

✓ Save X Cancel

Business Action to Invoke: By default, this field is blank. Optionally, select a Business Action to be invoked after a new part is created using the Create New Part dialog. This can be helpful when a Business Action is configured to normalize / harmonize a part number. For more information about using this feature, see the **Creating and Referencing New Competitor and/or OE Part Numbers** topic within this section.

Table Headers: By default, the following Table Header components are added to the parameter:

Manufacturer References: Table header component used to display the manufacturer and/or attribute value references of a selected part. Includes the following parameters:

Manufacturer References Properties

Label:

Display Value:

Attribute:

- **Label:** By default, the Label parameter is populated as 'Manufacturer.' Optionally edit the parameter to a more suitable label.
- **Display Value:** By default, parameter uses the 'PARENT_NAME' value. This allows the parent of the applicated competitor and/or OE part number to display within the Web UI. Optionally, the 'ATTRIBUTE_VALUE' value can be selected. The ATTRIBUTE_VALUE allows the values of the attribute selected within the 'Attribute' parameter to display within the Web UI.
- **Attribute:** Parameter must be populated with an attribute that is valid for the competitor or OE number Object Types.

Note: The Attribute parameter should only be populated when the 'ATTRIBUTE_VALUE' value is selected. Otherwise the component should be configured as shown in the example above.

Name Header: Table header component that displays the name of the part numbers in the table. Includes the following parameters:

Name Header Properties

Dimensions:

Label:

Table Sorting:

- **Dimensions:** By default, the parameter is blank. Optionally, select to use the 'Table Header Dimensions.'

- **Label:** By default, when the Label parameter is blank, the column label will display as 'Name.'
Optionally edit the parameter to a more suitable label.
- **Table Sorting:** Optionally, specifies the default sorting to be applied to the header.

Note: Additional components can be added to the Table Headers parameter. It can be especially helpful to add the 'Headline' child component to provide a visual separation and heading between components, and/or components used within the Headers parameter of an Application Manager Screen (i.e., Application Set Assembly, Application Part Type Title Header, Application Set Part, Application Condition Header - Individual, Application Condition Header - Group, Application Comment, Application Asset References).

New Number Prefix: By default, this mandatory parameter is populated with the following prefix: 'NewReferenceNumber_'. This prefix will be used for the ID of any part number that is created using this Create New Part dialog (as shown below). This must be a unique prefix. Do not use an existing prefix if this default prefix is currently in use for your implementation, then edit the prefix so it is unique, otherwise errors will occur. Optionally, if a different prefix is desired, edit the prefix to something more suitable, but unique.

The image shows a 'Create New Part' dialog box. The title bar contains the text 'Enter a manufacturer name and/or part number'. The dialog itself has a title 'Create New Part' and a close button (X). Below the title, there are two input fields: 'Manufacturer Name*' and 'Part Number*'. At the bottom of the dialog, there are two buttons: 'OK' and 'Cancel'.

Original Number Attribute: By default, this field is blank. Because part numbers created using the Create New Part dialog are normalized / harmonized to use alphanumeric characters only, this parameter allows for the original part numbers created using special characters, spaces, and/or non ASCII characters to be stored and searchable. Optionally, select an attribute that will be used to store the original part number exactly as it is entered when creating a new part using the Create New Part dialog. The attribute selected for this parameter must be valid for the competitor / OE number object type. The normalized / harmonized version will be stored as the STEP Name of the new part number.

Important: An error message will display, and users will not be able to create a new part if the attribute configured within the 'Original Number Attribute' parameter is not valid for the competitor and/or OE part number object type.

Reference Types: By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected STEP can follow the reference types listed within this parameter and display the target and/or related attribute values on the targets.

For example, within the AutoCare solution, when a part is selected STEP would need to follow the **PIES Interchange** reference and display the targets and/or a related attribute values on the targets.

For example, within the NAPA solution, when a part is selected STEP would need to follow the **Product to Interchange Product** reference and display the targets and/or a related attribute values on the targets.

For example, within the TecDoc solution, when a part is selected STEP would need to follow the **Supplier Article to Competitor Number** and/or the **Supplier Article to OE Number** references and display their targets and/or a related attribute values on the targets.

4. Click the **Save** button and the Part Number References Properties dialog will close, and the newly added component will display within the Rows parameter.
5. Click the **Save** button and then click the **Close** button to close the designer.

Configuring the Part Number Suggestions Component

The Part Number Suggestions component is a Node Editor component usually used along with the Part Number References component to display potential part numbers listed as suggestions. Part number suggestions displays the suggestion list based on the configured reference types and Suggestion Plugins when a part is selected in the Tree navigator. For information about how to use this component, and prerequisites, see the **Competitor and OE Numbers Solution** topic.

Configuration Steps

Below are steps to configure the Part Number Suggestions component within a Node Editor. Each of the parameters for the Part Number Suggestions Properties are described below.

1. Using the Designer, go to Node Editor Properties Child Components for the screen that needs to display part number suggestions.
2. Click the **Add** button for the Child Components Rows parameter, and the Add Component dialog will display.
3. Search for and select the Part Number Suggestions component.
4. Click the **Add** button, the Add Component dialog will close, and the Part Number Suggestions Properties dialog will display as shown below.

Properties

Configuration Web UI style

ProductDetails Save Close New... Delete Rename Save as...

Node Editor Properties [go to parent](#)

Child Components 1

Rows

Headline
Part Number References
Headline

2 Add... Remove Up Down

Add Component

Parent Selector

Parent Type Ahead

Parent Value

Part Number References

Part Number Suggestions 3

Planned Spread Revision Compare In Workflow

Filter

Show deprecated components

4 ✓ Add ✕ Cancel

A component used for suggesting parts that may need to be referenced to a selected part based on existing data (i.e., competitor and/or OE numbers).

Edit component

Part Number Suggestions Properties

Table Headers *

Manufacturer References
Name Header

Add... Edit... Remove Up Down

Reference Types *

--

Add... Remove Up Down

✓ Save ✕ Cancel

Table Headers: By default, this required parameter is pre-populated with the Manufacturer References and Name Header components. Optionally edit these components, and/or add additional Headers components to display information about the suggested competitor and/or OE number.

Reference Types: By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected, STEP can follow the reference types listed within this parameter and directly display the targets and/or target related information based on the plugins configured in the Suggestion Plugins parameter.

- For example, within the AutoCare solution, when a part is selected STEP would need to follow the **PIES Interchange** reference and display the targets and/or target related information.
 - For example, within the NAPA solution, when a part is selected STEP would need to follow the **Product to Interchange Product** reference and display the targets and/or target related information.
 - For example, within the TecDoc solution, when a part is selected STEP would need to follow the **Supplier Article to Competitor Number** and/or the **Supplier Article to OE Number** references and display their targets and/or target related information.
5. Click the **Save** button, the Part Number Suggestions Properties dialog will close, and the newly added component will display within the Rows parameter.
 6. Double click the newly added Part Number Suggestions component, and the Part Number Suggestions Properties will display as shown below.

Properties

Configuration Web UI style

ProductDetails Save Close New... Delete Rename Save as...

Part Number Suggestions Properties [go to parent](#)

Table Headers *

Manufacturer References (Make / PARENT Name Header)

Add... Edit... Remove Up Down

Reference Types *

TD_SupplierArticleToOENumber

Add... Remove Up Down

Child Components

Suggestion Plugins *

Same Number

Add... Remove Up Down

Suggestion Plugins: By default, this required parameter is pre-populated with the Same Number plugin. Optionally, a user can add one or more plugins by clicking the **Add** button. It is mandatory to add at least one plugin so that when a part is selected, information about the suggested competitor and/or OE number can be displayed.

Common applications: This plugin is used to suggest other part numbers that have the same applications as the current part number. Additional configuration parameters are not available for this plugin.

Common number: This plugin suggests the other part number(s) that are referencing the same part number (or Competitor Number or OE Number) as current part number does. In other words, If the current part refers to a part number (OE Number / Competitor Number) that is referenced by an other part number (OE Number / Competitor Number), then the other part number is suggested via this plugin. For example, if two different parts 'JK21499' and 'JK36004' are referenced to a part 'GL26409', then part 'JK36004' is suggested in the Part Number Suggestions component when part 'JK21499' is selected.

Double click on the plugin name in the Suggestions Plugins parameter to further configure the plugin.

- **Reference Types:** By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected, the reference types listed within this parameter are followed and display the target and/or related competitor and/or OE number information on the targets.

Referenced By: This plugin is used for suggesting part numbers that are referencing the current part number. Additional configuration is required for this plugin. Double click the plugin name in the Suggestions Plugins parameter and click on the **Add** button to add the reference type. When a part is selected, the reference types listed within this parameter are followed, and the source part number and its related information is displayed.

Referenced By Properties [go to parent](#)

Component Description: A plugin used for suggesting other part numbers that are referencing current number

Referenced By Types:

- For AutoCare solution this may be adding reference type with id 'AC_PartRelation'
- For NAPA solution this may be adding reference type with id 'NAPA_PartRelation'
- For TecDoc solution this may be adding reference type with id 'TD_PartRelation'

Referenced Chain: This plugin is used for suggesting other part numbers that are referenced by the part number which is already referenced by the current part number. Double click on the plugin name in the Suggestions Plugins parameter to further configure the plugin.

Referenced Chain Properties [go to parent](#)

Component Description: A plugin used for suggesting other part numbers that are referenced by numbers current part is referencing

Max Chain Size:

Reference Types:

- **Max Chain Size:** By default, this required parameter is blank and allows a numeric value. This parameter defines the maximum allowed chain size to follow the reference types listed within the next parameter Reference Types. For example, If the value '4' is entered against the Max Chain Size parameter, then the maximum chain links of references that the STEP looks to suggest potential Part Number / OE Numbers / Competitor Number is four.
- **Reference Types:** By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type. When a part is selected, the reference types listed within this parameter are followed and the target and/or related competitor and/or OE number information on the targets are displayed.

For example, consider there are six different parts 'JK21499', 'JK36004', 'JK36009', 'JK36112', 'VC21499' and 'VC36004' interlinked in a manner, 'JK21499' is referenced by 'JK36004', 'JK36004' is referenced by 'JK36009', 'JK36009' is referenced by 'JK36112', 'JK36112' is referenced by 'VC21499' and 'VC21499' is referenced by 'VC36004'. Assuming that the Max Chain Size value as '4' and all the references connecting these parts are listed within the Reference Types parameter, the Part Number Suggestions component would suggest the parts 'JK36004', 'JK36009', 'JK36112' and 'VC21499' when the part 'JK21499' is selected.

Same Number: This plugin is used to suggest other part numbers based upon matching part number names used in the Part Number Suggestions component. Additional configuration parameters are not available for this plugin. However, customers with the Extension API license can create their own plugins. For more information, click the STEP API Documentation button on the Start Page, or contact your Stibo Systems representative.

7. Click the **Save** button and then click the **Close** button to close the designer.

Configuring the Advanced Part Number References Component

The Advanced Part Number References component is a Node Editor component used to display referenced part numbers (Competitor / OE Number) and their attribute and/or metadata values in a filtered and sorted way when a part number is selected within the Tree navigator. This is an extended feature of the Part Number References component with an additional feature of Node List properties.

This section describes how to configure the Advanced Part Number References component. For information about how to use this component, and prerequisites, see the **Competitor and OE Numbers Solution** topic.

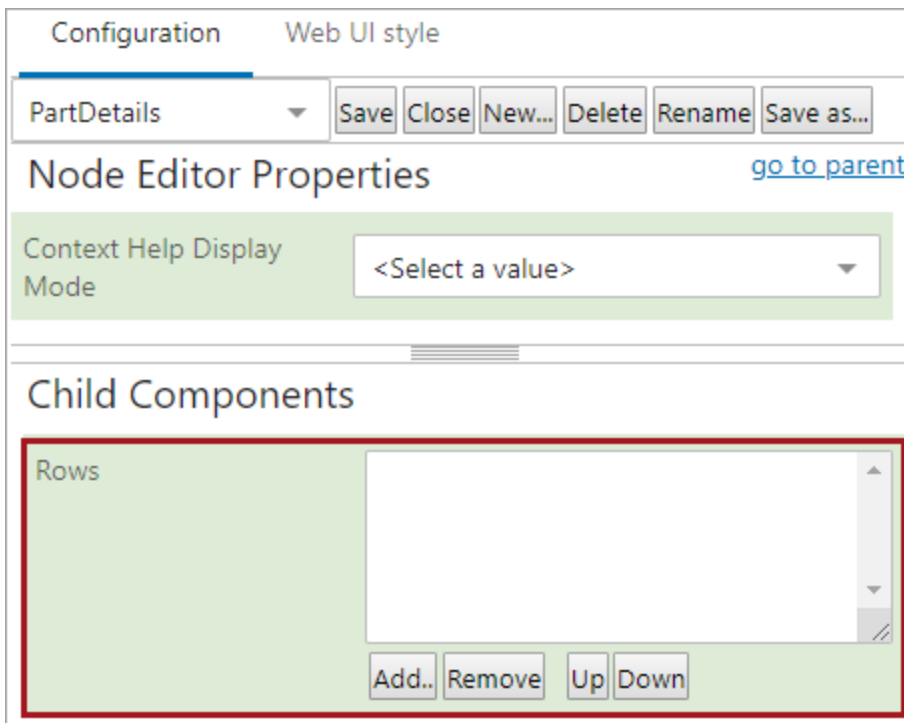
Prerequisites

Anyone configuring the Advanced Part Number References component within a Node Editor is expected to be familiar with the basic configuration and functionalities of a Node Editor component and a Node List component. For more information, see the **Node Editor Component** and **Node List Component** topics within the **Using a Web UI** section of **STEP Online Help**.

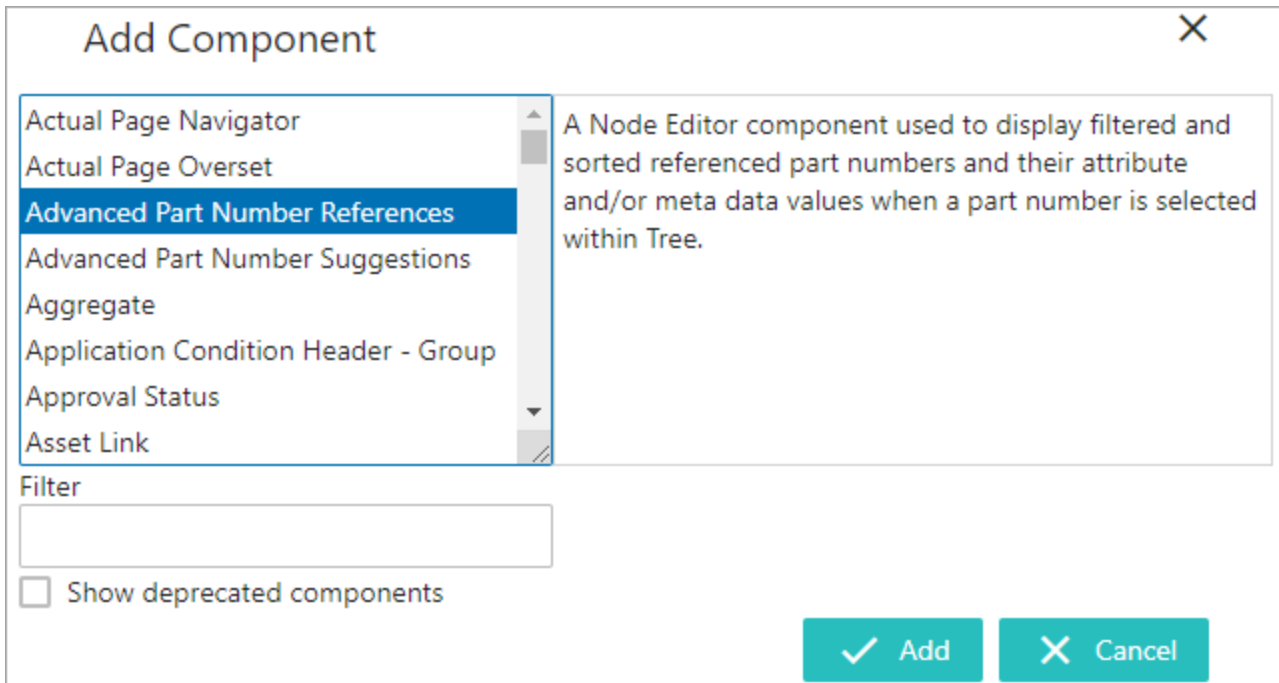
Configuration Steps

Below are the steps to configure the Advanced Part Number References component within a Node Editor. Each of the parameters for the Advanced Part Number References Properties are described below.

1. In the Web UI Designer, go to Node Editor Properties > Child Components > Rows parameter for the screen that needs to display competitor and/or OE number information.



2. Click the **Add** button , and the Add Component dialog will display.



3. Select the **Advanced Part Number References** component, and click **Add** button. The Advanced Part Number References Properties dialog will display. Parameter options are explained below:

Advanced Part Number References Properties [go to parent](#)

Business Action to Invoke ... Clear

Height

*New Number Prefix

Original Number Attribute ... Clear

*Reference Types

Add... Remove Up Down

▼ Advanced

Suggestion Manufacturer Name

Suggestion Manufacturer Attribute ... Clear

Reference Types to create

Add... Remove Up Down

Business Action to Invoke: By default, this field is blank. Optionally, select a Business Action to be invoked after a new part is created using the Create New Part dialog. This can be helpful when a Business Action is configured to normalize / harmonize a part number. For more information about using this feature, see the **Creating and Referencing New Competitor and/or OE Part Numbers** topic within this section.

Height: This optional parameter specifies the visual height of the Advanced Part Number References component in pixels. By default, this parameter is pre-populated with a value '400.' User can enter a numeric value to alter height of the component.

New Number Prefix: By default, this mandatory parameter is populated with the following prefix: 'NewReferenceNumber_.' This prefix will be used for the ID of any part number that is created using this Create New Part dialog (as shown below). Optionally, if a different prefix is desired, edit the prefix to something more suitable.

Important: The prefix must be unique, otherwise errors will occur. For example, if the existing default prefix is already in use for the implementation, then edit the prefix so it is unique.

Original Number Attribute: By default, this field is blank. Because part numbers created using the Create New Part dialog are normalized / harmonized to use alphanumeric characters only, this parameter allows for the original part numbers created using special characters, spaces, and/or non ASCII characters to be stored and made searchable. Optionally, select an attribute to be used to store the original part number exactly as it is entered when creating a new part using the Create New Part dialog. The attribute selected for this parameter must be valid for the competitor / OE number object type. The normalized / harmonized version will be stored as the STEP Name of the new part number.

Important: An error message will display, and users will not be able to create a new part if the attribute configured within the 'Original Number Attribute' parameter is not valid for the competitor and/or OE part number object type.

Reference Types: By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected STEP can follow the reference types listed within this parameter and display the target and/or related attribute values on the targets. Some examples include:

- Within the AutoCare solution, when a part is selected STEP would need to follow the **PIES Interchange** reference and display the targets and/or a related attribute values on the targets.
- Within the NAPA solution, when a part is selected STEP would need to follow the **Product to Interchange Product** reference and display the targets and/or a related attribute values on the targets.
- Within the TecDoc solution, when a part is selected STEP would need to follow the **Supplier Article to Competitor Number** and/or the **Supplier Article to OE Number** references and display their targets and/or a related attribute values on the targets.

Suggestion Manufacturer Name: By default, the parameter is populated with a value 'PARENT_NAME'. This parameter defines the search criteria in the typeahead field whether if the competitor / OE number search

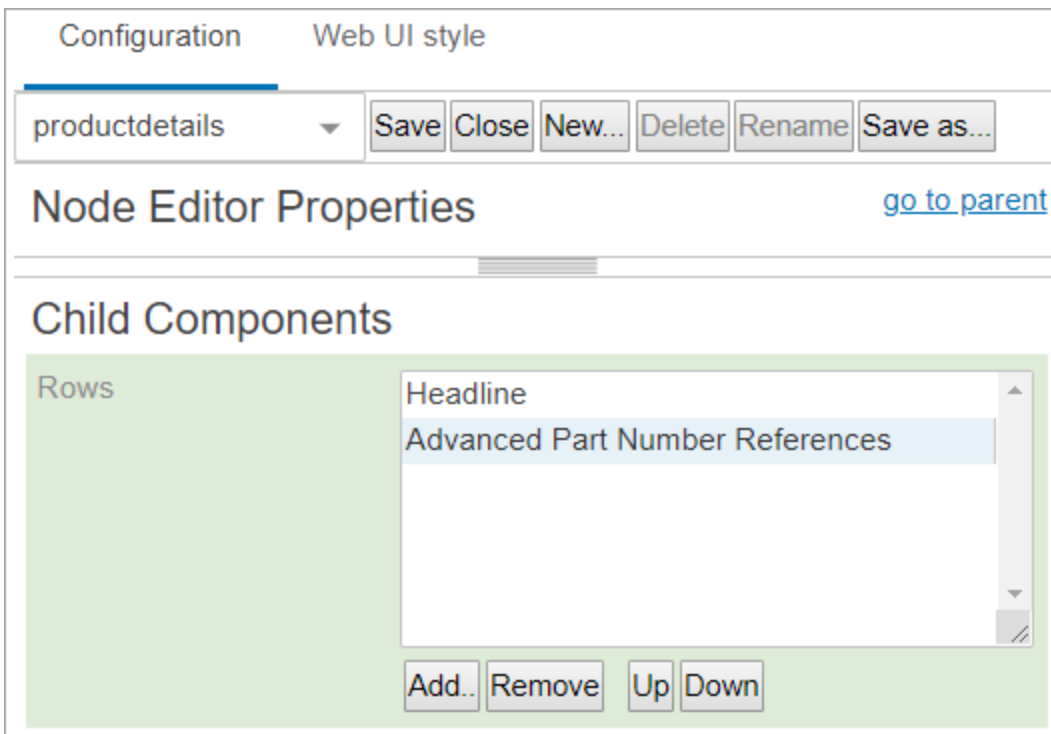
is based on the manufacturer name or the configured attribute value (configured in the Suggestion Manufacturer Attribute parameter explained below). Following options are available within this parameter:

- **PARENT_NAME:** Selecting this option defines the search criteria in the typeahead field to search the competitor / OE number based on the manufacturer name.
- **ATTRIBUTE_VALUE:** Selecting this option defines the search criteria in the typeahead field to search the competitor / OE number through the configured attribute value. The valid attribute is configured in the next parameter Suggestion Manufacturer Attribute.

Suggestion Manufacturer Attribute: By default, this parameter is blank. The value of an attribute configured in this parameter is used to search the competitor / OE number in the typeahead field. To search through an attribute value, user must have the 'ATTRIBUTE_VALUE' option selected in the Suggestion Manufacturer Name parameter above. To select an attribute, click the ellipsis to the right of the field. A Select Node(s) dialog will appear and an attribute can be selected by browsing and navigating down to the desired attribute or by using the search feature. The attribute selected for this parameter must be valid for the competitor / OE number object type.

Reference Types to create: By default, this parameter is blank. This parameter defines the reference type target that the part gets created while creating a part through the typeahead field. Adding a reference type in this parameter becomes necessary when there is more than two reference types present in the Reference Types parameter. If no reference type is listed in this parameter than the creation of new part through the typeahead field will randomly be the target of any one of the reference type listed in the Reference Types parameter.

4. Click the **Save** button and the Advanced Part Number References Properties dialog will close, and the newly added component will display within the Rows parameter of the Node Editor Properties as shown in the screenshot below. Double click the added component to make further edits if needed.



Note: The Node List component defined within this parameter displays the competitor and/or OE number information in a table / grid format. The view can be further customized by configuring the data to display via different display modes and adding different action buttons that users can click while working with the competitor and/or OE number information. The display modes can then be customized with a range of headers, allowing for different information about the listed objects to be displayed. The behavior of many of the various actions can be further configured, also. For more information, see the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

5. Click the **Save** button and then click the **Close** button to close the designer.

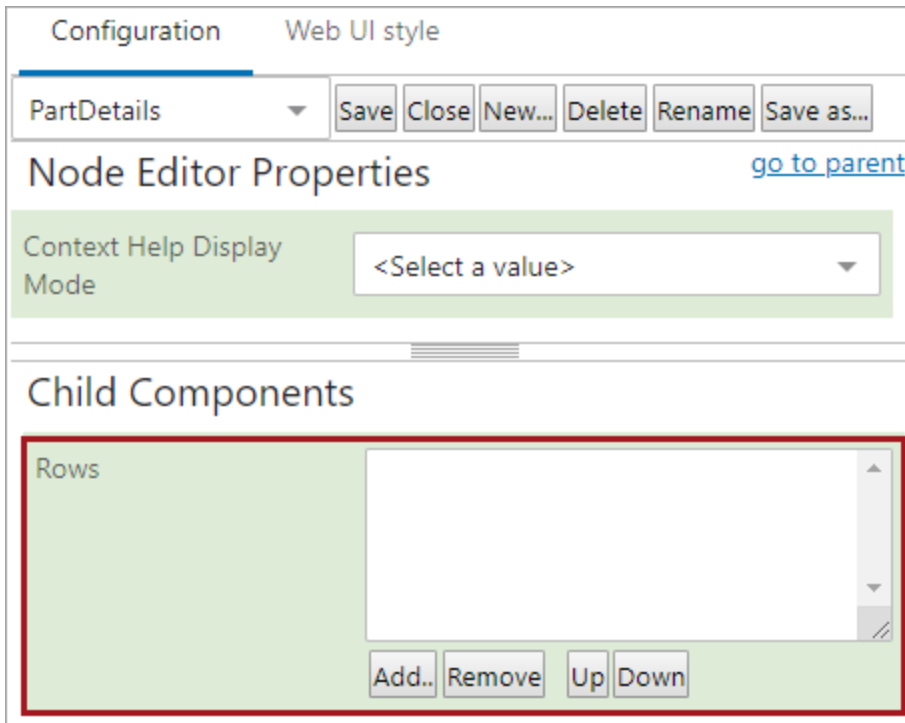
Configuring the Advanced Part Number Suggestions Component

The Advanced Part Number Suggestions component is a Node Editor component, often used along with the Advanced Part Number References component, to display potential part numbers listed as suggestions. The Advanced Part Number Suggestions component displays the suggestion list based on the configured reference types and Suggestion Plugins when a part is selected in the Tree navigator. This component is an advanced feature of Part Number Suggestions Component with additional features as explained in this document. For information about how to use this component, and prerequisites, see the **Competitor and OE Numbers Solution** topic.

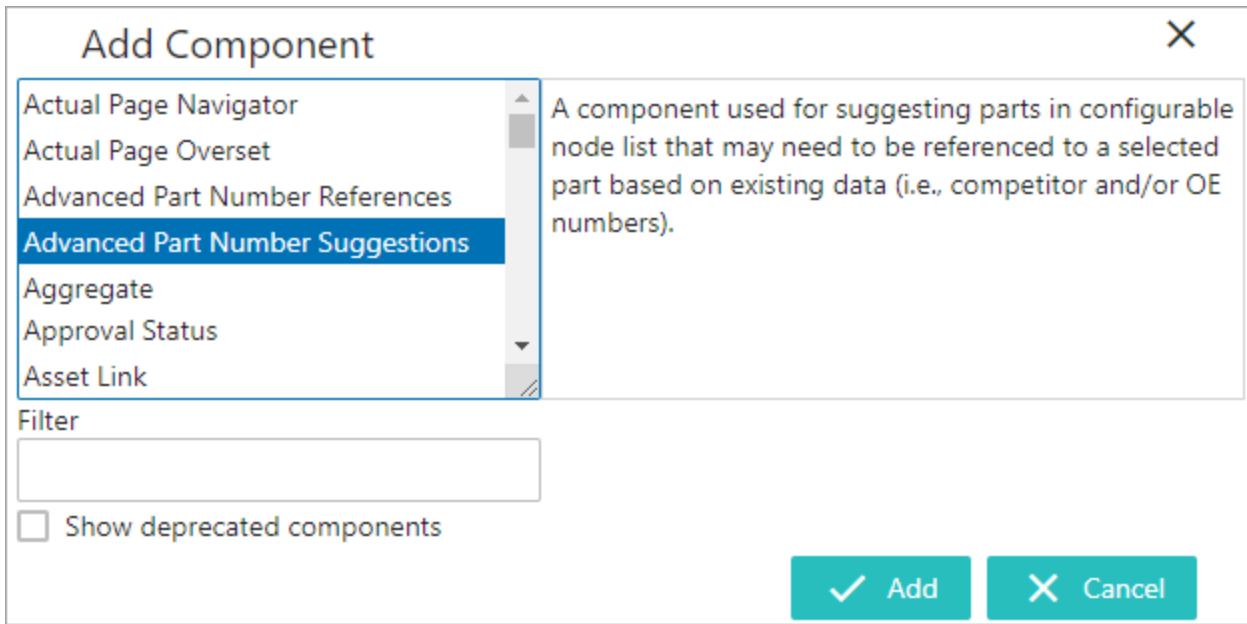
Configuration Steps

Below are steps to configure the Advanced Part Number Suggestions component within a Node Editor. Each of the parameters for the Advanced Part Number Suggestions Properties are described below.

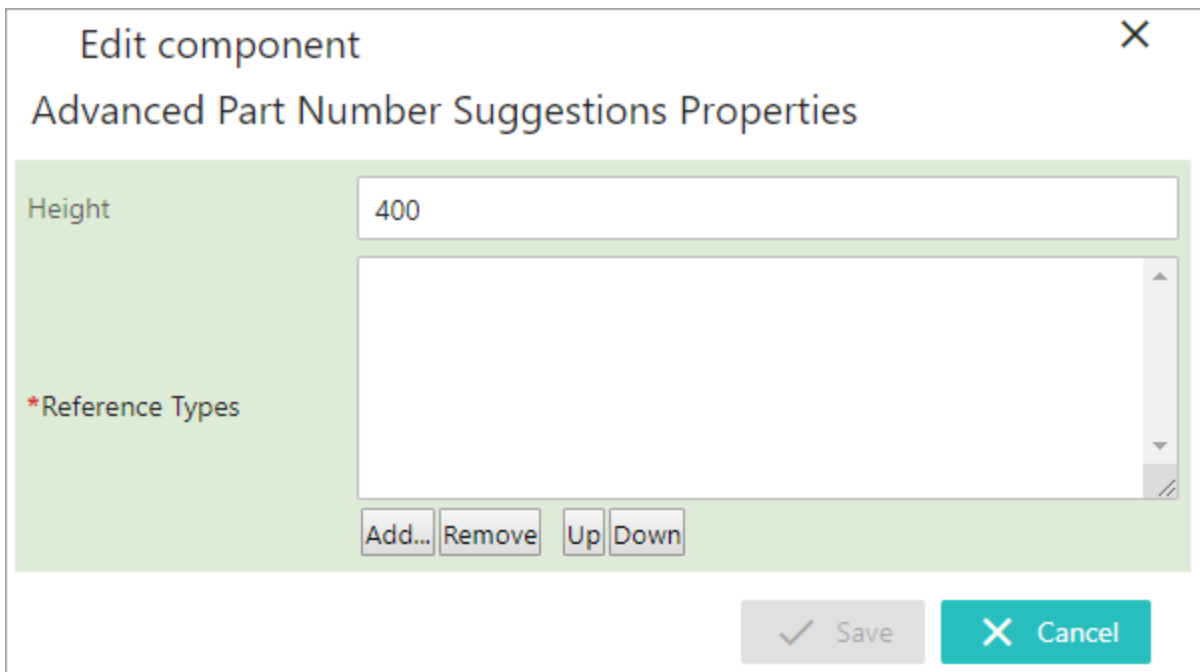
1. In the Web UI the Designer, go to Node Editor Properties > Child Components > Rows parameter for the screen that needs to display part number suggestions.



2. Click the **Add** button, and the Add Component dialog will display.



3. Select the Advanced Part Number Suggestions component, click **Add**, and the Advanced Part Number Suggestions Properties dialog will display. Parameter options are explained below:

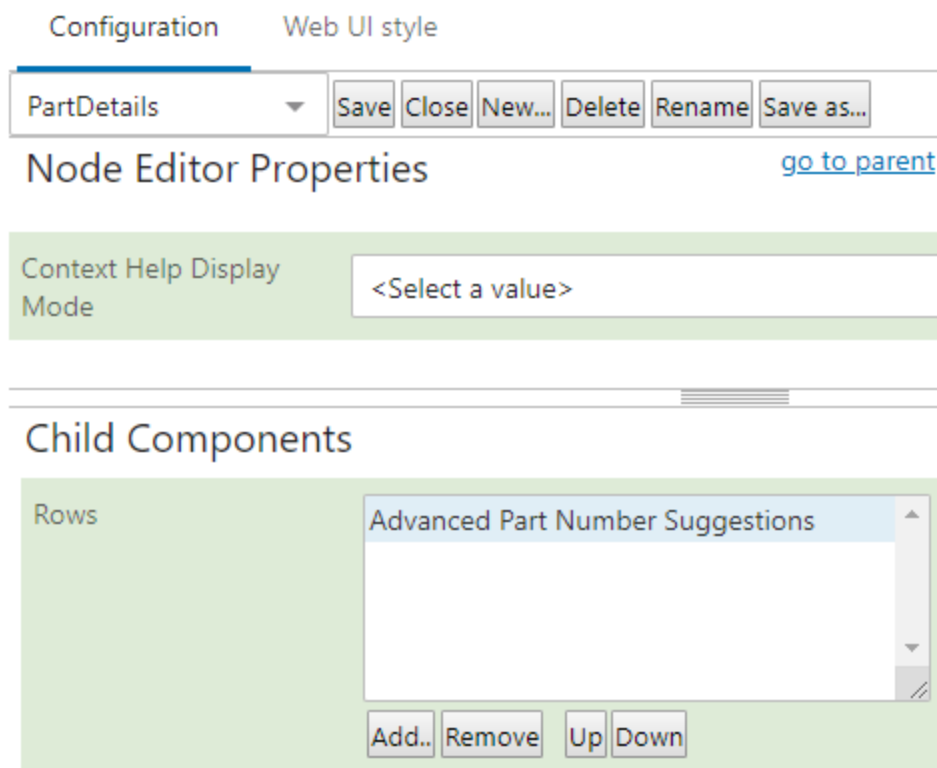


Height: This optional parameter specifies the visual height of the Advanced Part Number Suggestions component in pixels. By default, this parameter is pre-populated with a value '400.' Users can enter a numeric value to alter height of the component.

Reference Types: By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected, STEP can follow the reference types listed within this

parameter and directly display the target and/or targets related information based on the plugins configured in the Suggestion Plugins parameter. Some examples are:

- Within the AutoCare solution, when a part is selected STEP would need to follow the **PIES Interchange** reference and display the target and/or targets related information.
 - Within the NAPA solution, when a part is selected STEP would need to follow the **Product to Interchange Product** reference and display the target and/or targets related information.
 - Within the TecDoc solution, when a part is selected STEP would need to either follow the **Supplier Article to Competitor Number** and/or the **Supplier Article to OE Number** reference and display the target and/or targets related to the selected reference.
4. Click the **Save** button, the Advanced Part Number Suggestions Properties dialog will close, and the newly added component will display within the Rows parameter.



5. Double click the newly added Advanced Part Number Suggestions component, and the Advanced Part Number Suggestions Properties will display as shown below.

Properties (edited)

Configuration Web UI style

PartDetails ▾ Save Close New... Delete Rename Save as...

Advanced Part Number Suggestions Properties [go to parent](#)

Height

*Reference Types

AC_PIESInterchange

AC_PartRelation

Add... Remove Up Down

Child Components

Node List [go to component](#)

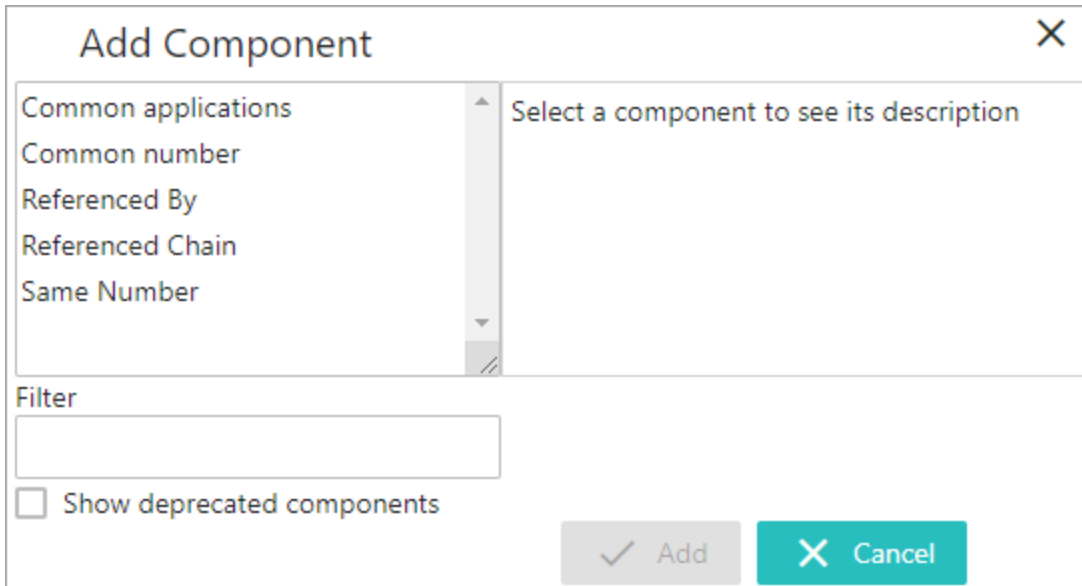
*Suggestion Plugins

Same Number

Add.. Remove Up Down

Note: The Node List component defined within this parameter displays the competitor and/or OE number information in a table / grid format. The view can be further customized by configuring the data to display via different display modes and adding different action buttons that users can click while working with the competitor and/or OE number information. The display modes can then be customized with a range of headers, allowing for different information about the listed objects to be displayed. The behavior of many of the various actions can be further configured, also. For more information, see the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

Suggestion Plugins: By default, this required parameter is pre-populated with the Same Number plugin. Optionally, a user can add one or more plugins by clicking the **Add** button. It is mandatory to have at least one plugin so that when a part is selected, information about the suggested competitor and/or OE number can be displayed.



The Suggestion Plugins options are the following:

Common applications: This plugin is used to suggest other part numbers that have the same applications as the current part number. Additional configuration parameters are not available for this plugin.

Common number: This plugin suggests other part number(s) that are referencing the same part number (or Competitor Number or OE Number) as the currently selected part number does. In other words, if the current part refers to a part number (OE Number / Competitor Number) that is referenced by another part number (OE Number / Competitor Number), then the other part number is suggested via this plugin. For example, if two different parts 'JK21499' and 'JK36004' are referenced to a part 'GL26409', then part 'JK36004' is suggested in the Part Number Suggestions component when part 'JK21499' is selected.

Double click on the plugin name in the Suggestions Plugins parameter to further configure the plugin.

Common number Properties [go to parent](#)

Component Description: A plugin used for suggesting other part numbers are referencing the same numbers as current part

Reference Types:

- **Reference Types:** By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected, the reference types listed within this parameter are followed and display the target and/or related competitor and/or OE number information on the targets.

Referenced By: This plugin is used for suggesting part numbers that are referencing the current part number. Additional configuration is required for this plugin. Double click the plugin name in the Suggestions Plugins parameter and click on the **Add** button to add the reference type. When a part is selected, the reference types listed within this parameter are followed, and the source part number and its related information is displayed.

Referenced By Properties [go to parent](#)

Component Description: A plugin used for suggesting other part numbers that are referencing current number

Referenced By Types:

- For AutoCare solution this may be adding reference type with id 'AC_PartRelation'
- For NAPA solution this may be adding reference type with id 'NAPA_PartRelation'
- For TecDoc solution this may be adding reference type with id 'TD_PartRelation'

Referenced Chain: This plugin is used for suggesting a series of part numbers that reference one another. In other words, part number 'A' suggests part number 'B.' Part number 'B' suggests Part number 'C' and so on. Double click on the plugin name in the Suggestions Plugins parameter to further configure the plugin.

Referenced Chain Properties [go to parent](#)

Component Description A plugin used for suggesting other part numbers that are referenced by numbers current part is referencing

Max Chain Size

Reference Types

TD_PartRelation
 TD_SupplierArticleToOENumber
 TD_SupplierArticleToCompetitorNumber

- **Max Chain Size:** By default, this required parameter is blank and allows a numeric value. This parameter defines the maximum allowed chain size to follow the reference types listed within the next parameter Reference Types. For example, If the value '4' is entered against the Max Chain Size parameter, then the maximum chain links of references that the STEP looks to suggest as potential Part Number / OE Numbers / Competitor Number is four.
- **Reference Types:** By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type. When a part is selected, the reference types listed within this parameter are followed, and the target and/or related competitor and/or OE number information on the targets are displayed.

For example, consider there are six different parts 'JK21499', 'JK36004', 'JK36009', 'JK36112', 'VC21499' and 'VC36004' interlinked in a manner, 'JK21499' is referenced by 'JK36004', 'JK36004' is referenced by 'JK36009', 'JK36009' is referenced by 'JK36112', 'JK36112' is referenced by 'VC21499' and 'VC21499' is referenced by 'VC36004'. Assuming that the Max Chain Size value as '4' and all the references connecting these parts are listed within the Reference Types parameter, the Part Number Suggestions component would suggest only the parts 'JK36004', 'JK36009', 'JK36112' and 'VC21499' when the part 'JK21499' is selected.

Same Number: This plugin is used to suggest other part numbers based upon matching part number names used in the Part Number Suggestions component. Additional configuration parameters are not available for this plugin. However, customers with the Extension API license can create their own plugins. For more information, click the STEP API Documentation button on the Start Page, or contact your Stibo Systems representative.

6. Go to Child Components > Node List > in the dropdown select 'Node List' > click on 'go to component.' The Node List Properties dialog with two pre-configured action buttons will display as shown below.

Configuration
Web UI style

PartDetails
Save
Close
New...
Delete
Rename
Save as...

Node List Properties [go to parent](#)

Hide Standard Buttons

*ID

Include Labels

Lookup Screen Type For Navigation

Page Size

Use Details Overlay

Child Components

Display Modes

Add.. Remove Up Down

Actions

Add suggestion(s) to Reference Numbers Action

Remove from suggestions Action

Add.. Remove Up Down

Add suggestion(s) to Reference Numbers Action: This required action button adds the selected competitor and/or OE part number reference to a selected part from the Part Number Suggestions list. Further configuration is required as explained below.

Remove from suggestions Action: This action button removes the selected Competitor and/or OE part number reference of a selected part from the Part Number Suggestions list. This is a ready to use button and user can double click on the 'Remove from suggestions Action' to further configure the Icon style, title and label of the action button if necessary.

7. Double click on the 'Add suggestion(s) to Reference Numbers Action' to configure, and the Add suggestion(s) to Reference Numbers Action Properties will display.

Configuration Web UI style

PartDetails Save Close New... Delete Rename Save as...

Add suggestion(s) to Reference Numbers Action Properties [go to parent](#)

Icon Style Class: stibo-AssetPort-Toolbar-ActiveIcon-add

Label: i18n.stibo.automotive.portal.server.referenceNumbersSuggestions.acti

*Reference Types to create

Buttons: Add... Remove Up Down

Title

Select Node(s)

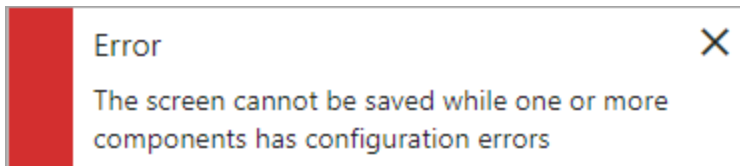
Browse Search

- Part of Kit (Part of Kit)
- Part Relation (NAPA_PartRelation)
- Part Relation (TD_PartRelation)
- Part Relation (AC_PartRelation)**
- Part Terminology To Application Note (AC_PartTerminologyTo...)
- PIES Interchange (AC_PIESInterchange)
- PIES Item To Kit Component Item (AC_PIESItemToKitCompon...)
- PlanBasketProdRef (PlanBasketProdRef)
- Product To Interchange Product (NAPA_ProductToInterchang...)
- ProductInheritance (ProductInheritance)
- RefImpTestAllowMultiple (RefImpTestAllowMultiple)
- RefImpTestDisAllowMultiple (RefImpTestDisAllowMultiple)
- RefInheritanceTestProdRef (RefInheritanceTestProdRef)
- RefViewDefinitionTest (RefViewDefinitionTest)

OK Cancel

- In the Reference Types to Create parameter, Click the **Add** button and select the Reference Type to determine the reference type which the action button follows to create a reference. The action button follows through the reference type(s) listed within this parameter and creates the references of the selected Competitor and/or OE part number to the selected part.

Note: It is required to add a minimum of one reference type in the Reference Types to Create parameter. Attempting to save the configuration prior to configuring the Reference Types to Create parameter will result in the following warning: The screen cannot be saved while one or more components has configuration errors.



- Click the **Save** button, and then click the **Close** button to close the designer.

Configuring the Data Model for Application Competitor or OE Part Numbers Table Header

The Application Competitor or OE Part Numbers Table Header involves two levels of configuration. The first level of configuration is performed in the workbench and the part is configured in the Web UI. After the Easy Setup action, the Application Competitor or OE Part Numbers Table Header Component requires additional setup in the data model and needs to be manually modified. The user needs to be aware of the data model structure of the modifying automotive standard. This document only describes the procedure to configure / modify the data model in the workbench. For information on configuring the Web UI portion, see the **Configuring the Application Competitor or OE Part Numbers Table Header Component** topic.

By the end of this configuration in the workbench, there must be a link established from Interchange application (competitor or OE number application) object to Part Type (Part Terminology / MPCC / Generic Article) and a reference established from Interchange application (competitor or OE number application) object to Vehicle (Base Vehicle / NAPA Year / VehicleType(CV or PC)).

Below are the steps to configure in the workbench.

Important: Throughout this topic, all the manually created Object Types / Objects are provided with example IDs for easier understanding. It is up to the user to either follow the same ID or their own while creating the Object Types / Objects.

1. Log in to the workbench, and access the System Setup tab.
2. Create a new Interchange application (competitor or OE number application) object type under an Interchange product object type.
 - For AutoCare solution this may be creating 'AC_PIESInterchangeApplication' under 'AC_PIESInterchangeItem'
 - For NAPA solution this may be creating 'NAPA_InterchangeApplication' under 'NAPA_InterchangeProduct'
 - For TecDoc solution this may be creating 'TD_DS_OENumberApplication' under 'TD_DS_OENumber' and/or 'TD_DS_CompetitorNumberApplication' under 'TD_DS_SupplierCompetitorNumber'

Below is an example of a created OE Number application for the TecDoc solution.

The screenshot shows the 'System Setup' interface. On the left is a tree view of the system structure, with 'OENumber Application' selected under 'OENumbers'. On the right is the configuration page for 'OENumber Application - Object Type'. It includes tabs for 'Object Type', 'References', and 'Log'. The 'Description' section contains a table with the following data:

Name	Value
ID	TD_DS_OENumberApplication
Name	OENumber Application
Last edited by	2019-01-29 02:51:44 by USERB
Name Pattern	
ID Pattern	
Manually Sorted	No
Enable Profiling	No
InDesign Template Allowed	No
Icon	
Dimension Dependencies	
Reference Target Lock Policy	Strict

The 'Aspects' section contains a table with columns: Component, Name, and Description. The 'Valid Attributes' section contains a table with columns: ID and Name, and an 'Add Attribute' link.

- Define the newly created Interchange application object type as valid Product Type in the following Classification Link Type.
 - For AutoCare solution this may be defining 'AC_PIESInterchangeApplication' as valid Product Type to 'AC_ProductToPartTerminology' Classification Link Type
 - For NAPA solution this may be defining 'NAPA_InterchangeApplication' as valid Product Type to 'NAPA_ProductToMPCC' Classification Link Type
 - For TecDoc solution this may be defining 'TD_DS_OENumberApplication' and/or 'TD_DS_CompeterNumberApplication' as valid Product Type to 'TD_SupplierArticleToGenericArticle' Classification Link Type

Below is an example of OE Number application being defined as the valid Product Type to 'TD_SupplierArticleToGenericArticle' Classification Link Type.

System Setup

- Users & Groups
- Reference Types
 - Product Reference Types
 - Image and Document Reference Types
 - Classification Reference Types
 - Product to Classification Link Types
 - Accessory List To Manufacturer
 - Accessory List To Model
 - Product To MPCC
 - Product To Part Terminology
 - Supplier Article To Driver Cab
 - Supplier Article To Generic Article**
 - Supplier Article To Universal GA
 - Supplier Brand To Supplier Asset Folder
 - Product Attribute Link Type
 - Classification Attribute Link Type
 - Entity Reference Types
 - Context Reference Types

Supplier Article To Generic Article - Validity

Product to Classification Link Type | Validity | Log

Valid Product Types

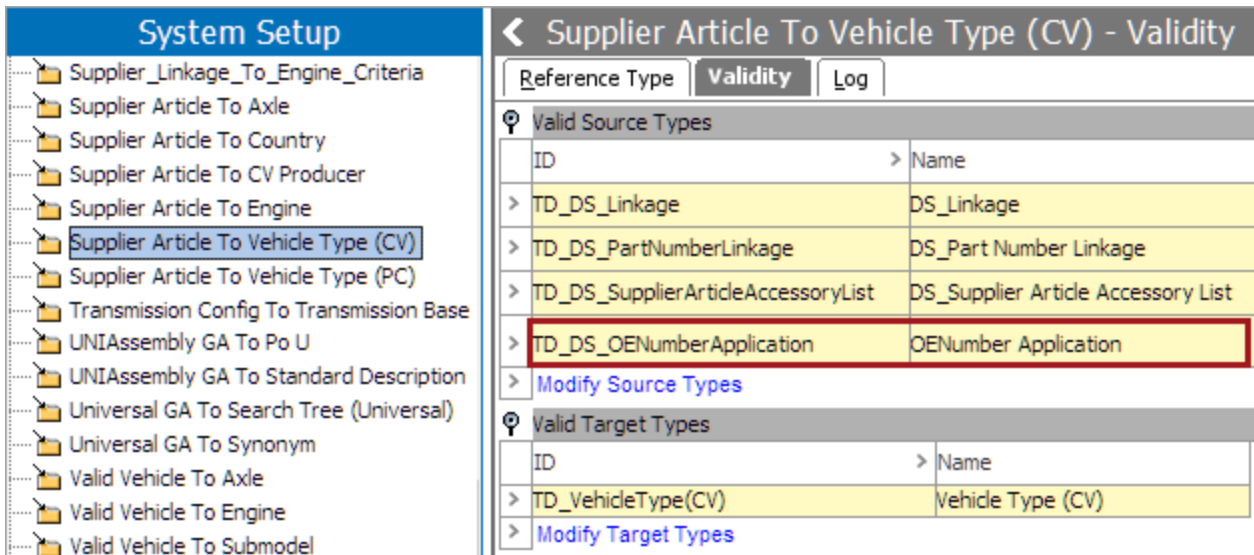
ID	Name
> TD_DS_AccessoryListMember	DS_Accessory List Member
> TD_DS_Linkage	DS_Linkage
> TD_DS_SupplierArticle	DS_Supplier Article
> TD_DS_SupplierGAfolder	DS_Supplier GA folder
> TD_DS_SupplierPartsList	DS_Supplier Parts List
> TD_LOVFilterGA	LOV Filter GA
> TD_DS_OENumberApplication	OENumber Application
Modify Product Types	

Valid Classification Types

ID	Name
> TD_StandardAssemblyGA	Standard Assembly GA
Modify or Move Classification Types	

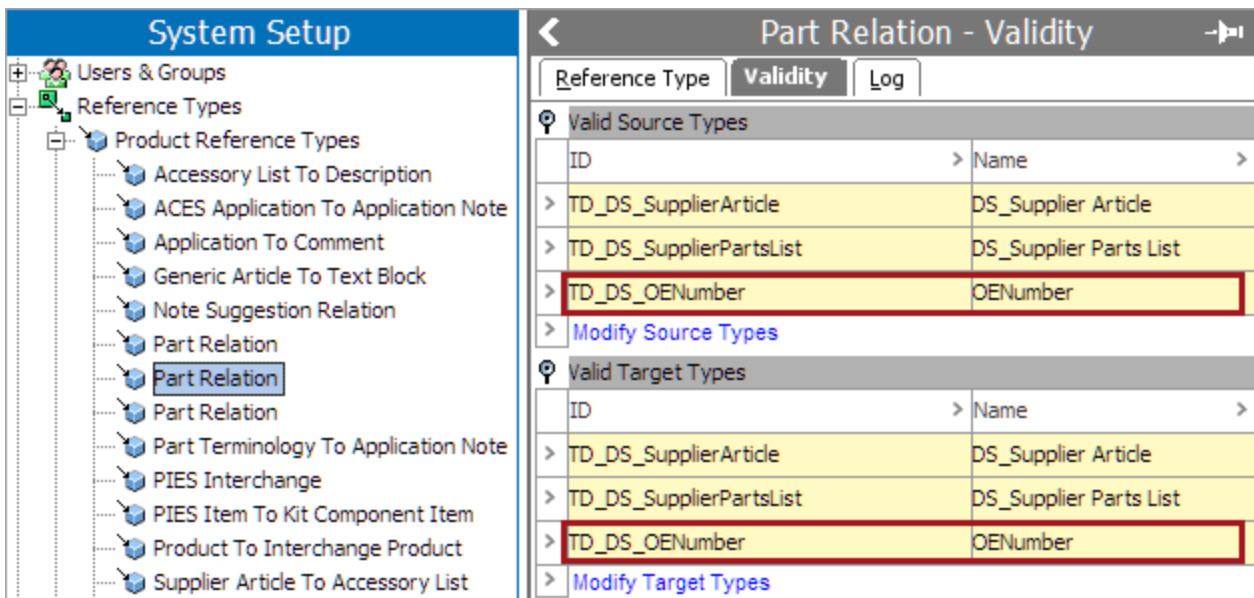
4. Define the created Interchange application object type as valid Source Type in the following Classification Reference Type.
- For Autocare solution this may be defining 'AC_PIESInterchangeApplication' as valid Source Type to 'AC_ACESApplicationToBaseVehicle' Classification Reference Type
 - For NAPA solution this may be defining 'NAPA_InterchangeApplication' as valid Source Type to 'NAPA_ApplicationToYear' Classification Reference Type
 - For TecDoc solution this may be defining 'TD_DS_OENumberApplication' and/or 'TD_DS_CopetitorNumberApplication' as valid Source Type to 'TD_SupplierArticleToVehicleType(CV)' and/or 'TD_SupplierArticleToVehicleType(PC)' Classification Reference Type

Below is an example of OE Number application object type being defined as the valid Source Type to 'TD_SupplierArticleToVehicleType(CV)' Classification Reference Type.



5. Define the Interchange (Competitor / OENumber) product object type as valid Source Type and valid Target Type in the following Product Reference Type.
 - Define 'AC_PIESInterchangeItem' object type as valid Source Type and valid Target Type to 'AC_PartRelation' Product Reference Type.
 - Define 'NAPA_InterchangeProduct' object type as valid Source Type and valid Target Type to 'NAPA_PartRelation' Product Reference Type.
 - Define 'TD_DS_OENumber' and 'TD_DS_SupplierCompetitorNumber' object type as valid Source Type and valid Target Type to 'TD_PartRelation' Product Reference Type.

Below is an example of OE Number object type being defined as the valid Source Type and valid Target Type to 'TD_PartRelation' Product Reference Type.



6. Add the Interchange (Competitor / OENumber) product object type to 'Part' aspect in Automotive - Application Model.
 - For the AutoCare solution, this may be 'AC_PIESInterchangeItem'
 - For the NAPA solution, this may be 'NAPA_InterchangeProduct'
 - For the TecDoc solution, this may be 'TD_DS_SupplierCompetitorNumber' or 'TD_DS_OENumber'

Below is an example of OE Number application object type being added to 'Part' aspect in Automotive - Application Model.

Automotive - Application Model - Component Model Configuration			
Component Model Configuration			
> Dummy types that has part type link	DS_Supplier Parts List		
	LOV Filter GA		
	PIES PCdb Part Terminology		
	Primary Product Classification		
> Part	DS_Supplier Article		The part object type
	DS_Supplier Parts List		
	NAPA Product		
	OENumber		
	PIES Item		
> Part type	NAPA MPCC		The part type object type
	Part Terminology		
	Standard Assembly GA		
	Universal Assembly GA		
> Part type list	NAPA Part Terminology List		The object type for a logical list of part types
	Part Terminology List		
	Part Type Group		

7. Add the created Interchange application object type to 'Application' aspect in Automotive - Application Model.
 - For the AutoCare solution, this may be 'AC_PIESInterchangeApplication'
 - For the NAPA solution, this may be 'NAPA_InterchangeApplication'
 - For the TecDoc solution, this may be 'TD_DS_OENumberApplication' and/or 'TD_DS_CompeteritorNumberApplication'

Below is an example of OE Number object type being added to 'Application' aspect in Automotive - Application Model.

System Setup

- Users & Groups
- Reference Types
- Workspaces
- Table
- Keys
- Event Queues
- Component Models
 - Address Component Model
 - Auto Classification Model
 - Automotive - Application Model**
 - Automotive - AutoCare Model
 - Automotive - Country Model
 - Automotive - Import Flow Process
 - Automotive - Mapper Extension
 - Automotive - NAPA Model
 - Automotive - TecDoc Model
 - Country Aliases
 - External Stored Assets Model
 - Google Shopping Model
 - Matching

Automotive - Application Model - Component Model Configuration

Component Model Configuration		
Name >	Value >	Description >
> Application	ACES Application	The object that represent an application record, should be child of the Part object type
	DS_Linkage	
	NAPA Application	
	OENumber Application	
> Assembly	Air Suspension Unit Axle	The object type for the object parts can be linked to. A car, engine or an abstract configuration of cars.
	Base Vehicle	
	Engine Code	
	Leave Spring Unit Axle	
	NAPA Year	
	Producer ID (CV)	
	Tractor Axle	
	Trailer Axle	
	Vehicle Type (CV)	
	Vehicle Type (PC)	

Configuring the Application Competitor or OE Part Numbers Table Header Component

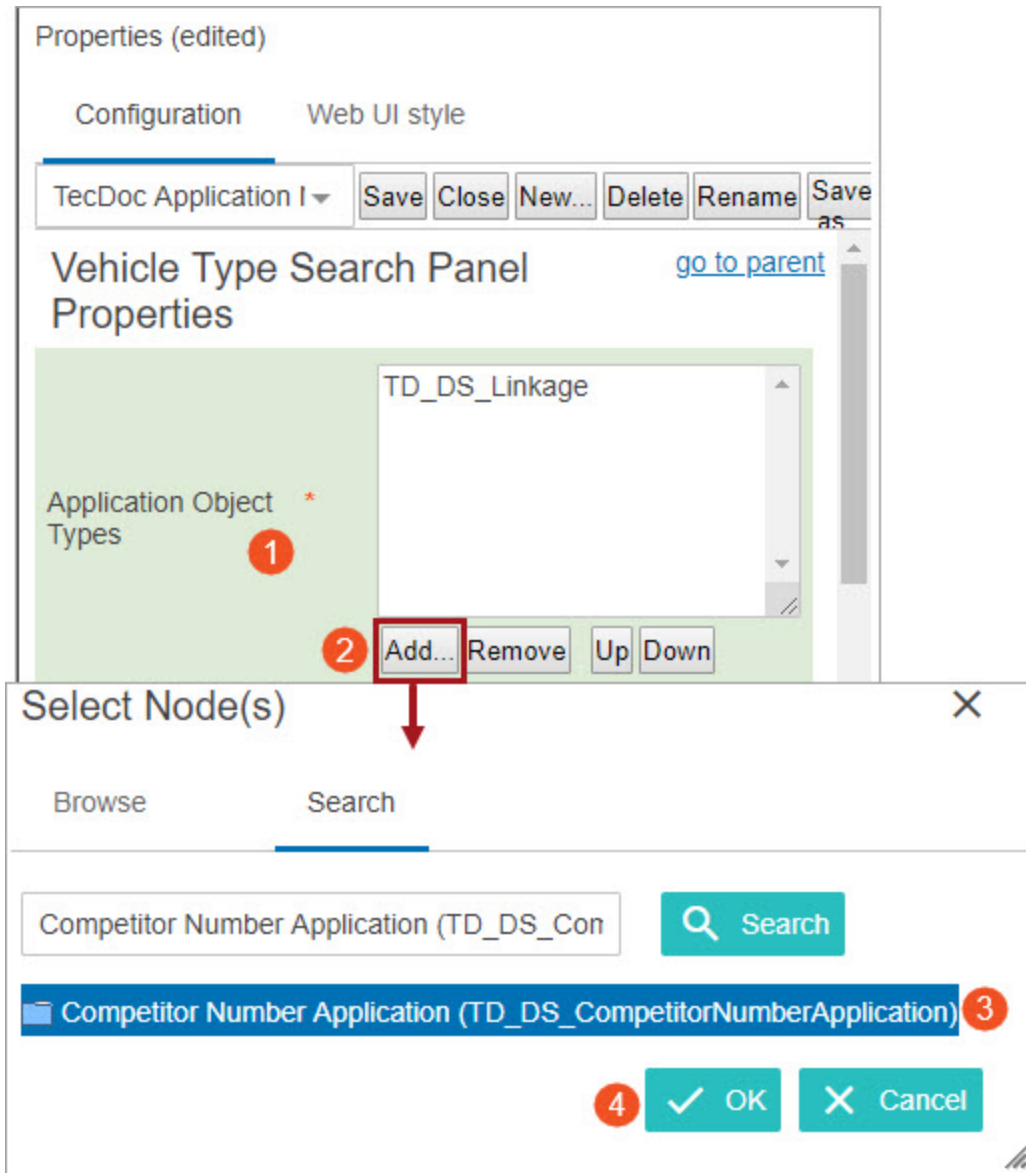
The Application Competitor or OE Part Numbers Table Header component is a Table Header component used to display the competitor or OE manufacturer part number and name in the Application Manager results table. Additionally, when a part number value within the column is clicked, a read-only dialog will display the competitor or OEM applications. For information about how to use this component and its prerequisites, see the **Competitor and OE Numbers Solution** topic.

Within the search result displayed in the Application Manager screen, the Application Competitor or OE Part Numbers Table Header suggests potential matches for the searched applications based on existing competitor and/or OE part number application references and/or matching applications within STEP.

As a prerequisites, after Easy Setup actions, user needs to configure the standard data model in workbench along with the Web UI configuration steps mentioned below. For more information about how to setup the data model to use this component, see the **Configuring the Data Model for Application Competitor or OE Part Numbers Table Header** topic.

Below are steps to configure the Application Competitor or OE Part Numbers Table Header Component within an Application Manager Node List. Each of the parameters for the Application Competitor or OE Part Numbers Properties are described below.

1. Using the Designer, go to Vehicle Type Search Panel Properties for the Application Manager screen that needs to display competitor and/or OE number (Interchange product) information.
2. Click the **Add** button for the Application Object Types parameter, and the Select Node(s) dialog will display.
3. Find and select the Interchange application (competitor or OE number application) object type whose part numbers and manufacturer name (optional) for each application will display within the column.
 - For the AutoCare solution, this may be 'AC_PIESInterchangeApplication'
 - For the NAPA solution, this may be 'NAPA_InterchangeApplication'
 - For the TecDoc solution, this may be 'TD_DS_CompetitorNumberApplication' and/or 'TD_DS_OENumberApplication'



4. In the Designer, navigate to Node List Properties Headers parameter for the Application Manager screen that needs to display competitor and/or OE number information.
5. Click the **Add** button for the Headers parameter, and the Add Component dialog will display.
6. Find and select the **Application Competitor or OE Part Numbers** component.
7. Click the **Add** button, and the Application Competitor or OE Part Numbers Properties dialog will display (as shown below).

Properties

Configuration Web UI style

TecDoc Application I ▾ Save Close New... Delete Rename Save as

Node List Properties [go to parent](#)

Headers **4**

- Application Set Assembly (Assem
- Application Part Type Title Heade
- Application Set Part (false / false
- Application Competitor or OE Par

5 Add... Edit... Remove Up Down

Add Component ✕

ACES Application Qualifiers	Table Header used to display competitor or OEM part numbers and manufacturer names (optional) in the Application Manager results table. Additionally, when users click a part number value within the column, a read-only dialog will display the competitor or OEM applications.
Application Assembly Value	
Application Asset Reference	
Application Comment	
Application Competitor or OE Part Numbers 6	
Application Condition Header - Group	
Application Condition Header - Individual	

Application Country Handling Table Header

Filter

Show deprecated components

7 ✓ Add ✕ Cancel

Add component - configure required properties ✕

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

Application Competitor or OE Part Numbers Properties

Component Description: Table Header used to display competitor or OEM part numbers and manufacturer names (optional) in the Application Manager results table. Additionally, when users click a part number value within the column, a read-only dialog will display the competitor or OEM applications.

Part Number Object *
Type

Display Manufacturer Name

Alternate Manufacturer Attribute ... Clear

Column Label

Dialog Label

Blacklisted Attributes //
Add... Remove Up Down

Blacklisted Attribute Groups //
Add... Remove Up Down

Blacklisted References //
Add... Remove Up Down

Dimensions Edit...

Table Sorting

✓ Add
✕ Cancel

8. Populate the required parameter listed below.

- **Part Number Object Type:** This required parameter is used to select a competitor or OEM part number object type whose part numbers and manufacturer name (optional) for each application will display within the

column. Click the ellipsis button (...) to browse and/or search for the necessary Object Type.

- For the AutoCare solution, this may be 'AC_PIESInterchangeItem'
 - For the NAPA solution, this may be 'NAPA_InterchangeProduct'
 - For the TecDoc solution, this may be 'TD_DS_SupplierCompetitorNumber' or 'TD_DS_OENumber'
9. Optionally, populate the other parameters listed below.
- **Display Manufacturer Name:** By default, this parameter is disabled, and only the manufacturer part number of the object type configured within the 'Part Number Object Type' parameter will display. Enable this parameter to display part numbers with their default manufacturer name.
 - **Alternate Manufacturer Attribute:** This parameter is used to select an attribute where the result cell displays the selected attribute value along with the part number. For this value to display within the results table column, the Display Manufacturer Name parameter must be enabled. This parameter is usually used when user needs to display an alternate manufacturer name other than the name provided in the parent object. Click the ellipsis button (...) to browse and/or search for the necessary attribute valid to the object type selected in the 'Part Number Object Type' parameter.
 - **Column Label:** By default, when the parameter is blank, the column label will display as 'Competitor / OEM.' Optionally, edit the parameter to a more suitable label.
 - **Dialog Label:** By default, when the parameter is blank, the dialog that displays when a user clicks the part number value within the column will use the 'Competitor or OEM Applications' title. Optionally, edit the parameter to a more suitable title.
 - **Blacklisted Attributes:** Click the Add button to add one or more attributes that do not need to display on the details dialog when a user clicks the part number value within the column.
 - **Blacklisted Attribute Groups:** Click the Add button to add one or more Attribute Groups that will hide attributes that should not display on the details dialog when a user clicks the part number value within the column. More information about the **Blacklisted Attribute Group** parameter can be found in the **Setting up Blacklisted Attribute Groups in Web UI** section of the **Handling Duplicated Attributes in Web UI** documentation.
 - **Blacklisted References:** Click the Add button to add one or more reference types that will not display on the details dialog when a user clicks the part number value within the column.
 - **Dimensions:** Optionally, determine the table header height and/or width dimensions. Only valid when the Multi Edit display mode is used.
 - **Table Sorting:** Optionally, specifies the default sorting to be applied to the header.
10. Click the **Add** button

Configuring the Metadata Attribute References Component

The Metadata Attribute References component is configured as a table header in many Web UI components. It is used to display the metadata attributes value of Reference and Link types within the Part Number References, Part Number Suggestions, Advanced Part Number References, and Advanced Part Number Suggestions components.

It is expected that anyone configuring the Metadata Attribute References component as a table header is familiar with the process required to add table headers into components, as basic concepts for accessing and adding the table headers are not covered in this section.

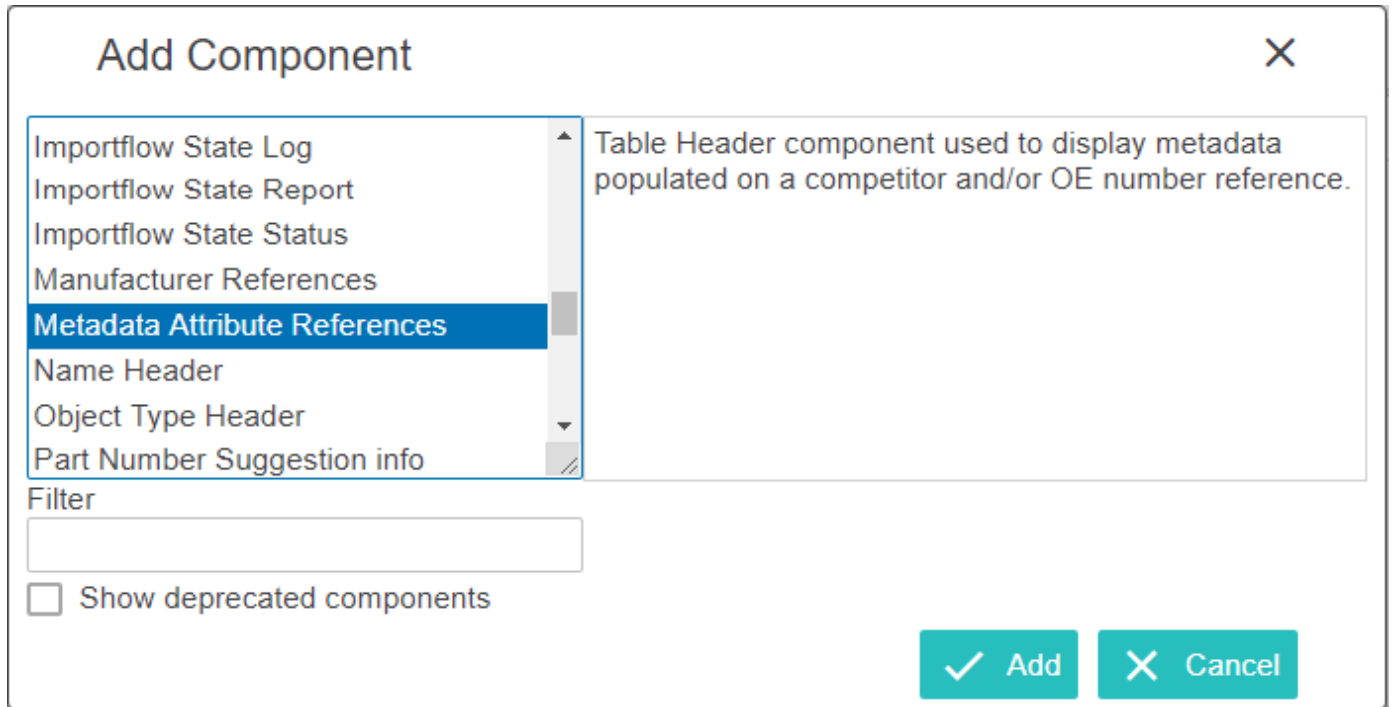
Configuration Steps

Below are the steps required to configure the Metadata Attribute References component within a Part Number References or Part Number Suggestions component. Each of the parameters for the Metadata Attribute References Properties are described below.

1. In the Web UI Designer, navigate to the Table Header / Header parameter for the component that will display metadata attributes. The example below displays the Table Header parameter within the Part Number References Properties dialog.

The screenshot shows the 'Part Number References Properties' dialog in the Web UI Designer. The 'Configuration' tab is selected, and the dropdown menu is set to 'productdetails'. The main area is titled 'Part Number References Properties' with a 'go to parent' link. The dialog contains several sections: 'Business Action to Invoke' with a text field and a 'Clear' button; 'Table Headers *' with a list box containing 'Manufacturer References (Manufa' and 'Name Header', and 'Reference Country Handling Table', with 'Add...', 'Edit...', 'Remove', 'Up', and 'Down' buttons below; 'New Number Prefix *' with a text field containing 'NewReferenceNumber_'; and 'Original Number Attribute' with a text field and a 'Clear' button.

2. Click the **Add** button found beneath the Header field (screenshot shown above), and the Add Component dialog will display (as shown below).



3. Search for and select the 'Metadata Attribute References', click **Add**, and the Metadata Attribute References dialog will display (as shown below). Parameter options are explained below:

Add component - configure required properties

✕

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

Metadata Attribute References Properties

Attribute *	<input style="width: 90%;" type="text" value=""/>	...
Dimensions	<input style="width: 90%;" type="text" value=" <Select an option>"/>	▼ Edit...
Label	<input style="width: 100%;" type="text" value=""/>	
Mandatory	<input style="width: 90%;" type="text" value=" <Select a value>"/>	▼
Readonly	<input type="checkbox"/>	
Table Sorting	<input style="width: 90%;" type="text" value=" <Select a value>"/>	▼

▼ Advanced

Enable Locale Formatting	<input type="checkbox"/>
Show Invalid Inherited Values	<input type="checkbox"/>
Show LOV IDs	<input type="checkbox"/>
No Wrap	<input type="checkbox"/>
Context Help	<input style="width: 90%;" type="text" value=""/>
Display Context Help	<input checked="" type="checkbox"/>

✓ Add

✕ Cancel


Attribute: The only mandatory parameter for this component. To select an attribute, click the ellipsis to the right of the Attribute field. A Select Node(s) dialog will appear and an attribute can be selected by browsing and navigating down to the desired attribute or by using the search feature.


Dimensions: By default, the parameter is blank. Optionally, to configure column width and height, select 'Table Header Dimensions.'

Label: Override the attribute name by adding new text that will display in its place. By default, the component title comes from the name of the configured attribute.

Mandatory: This parameter is used in Multi Edit Display mode to visually prompt the user to add a value for the defined metadata attribute displayed in the attribute table header. If 'Visually Mandatory' is selected from the dropdown, the header label will display with an asterisk (*), which informs the user a value must be added to the configured metadata attributes. In the example below, an asterisk (*) displays to the right of the metadata attribute 'ATTR Original'.

Part Number References

Select all  Multi edit view

	Name •	ID •	ATTR Original* •
<input type="checkbox"/> 1920GN	1920GN	TD_OE-63b821d198acfc1f26752288ece39043	1920.GN
<input type="checkbox"/> 4R8Q12B579AC	4R8Q12B579AC	TD_OE-f0cc379af0b2d8453b8c7069dab778aa	4R8Q-12B57-9AC
<input type="checkbox"/> C2C12919	C2C12919	TD_OE-7ff7baa72b9900f4919b98fbd7cc750f	C2C 12919
<input type="checkbox"/> PHF500090	PHF500090	TD_OE-7cce17ce646667cee6198a040e1e053b	PHF 500090

ReadOnly: Determines if the value field should be editable or read only.

- If disabled (default), then users can edit the field
- If enabled, the metadata attribute value on the screen will be locked so end users cannot make edits

Table Sorting: By default, the parameter is blank. Optionally, select from the dropdown values to specify the default sorting order of the column data.

Enable Locale Formatting: Determines if the 'ISO Date' or 'ISO Date and Time' attribute values are displayed per the locale selection.

- If enabled (not default), the 'ISO Date' or 'ISO Date and Time' attribute values will be displayed in the manner defined by the locale, selection of which is made when logging in.

Important: Additional setup is required to configure the 'Enable Locale Formatting' parameter. For more information, see the **Localizable Dates in Web UI** section of the **Web User Interfaces** documentation.

Show Invalid Inherited Values: Determines if the inherited attribute values need to be displayed even if the attribute is invalid to the part number displayed in the list.

- If enabled (not default), and the attribute has values inherited from the parent, then the inherited attribute value will display against the referenced / suggested part number.

Show LOV IDs: Determines whether the attribute's LOV ID displays for the end user.

- If enabled (not default), and the attribute's LOV has IDs enabled, then the LOV ID will display next to the value.

No Wrap: When unchecked, cell content (value of the attribute) will break, or 'wrap,' into multiple lines enabling display of as much of the cell content. When the content in the cell has excess of characters to display, then the metadata attribute value breaks to a second line, allowing the display of the whole value.

Context Help: Enter help text to display when a user hovers over the attribute name. Help texts can be used to provide information to end users about an attribute and how it should be populated. Display of this text requires that the Display Context Help parameter, described below, is enabled. More information about the Context Help parameter can be found in the **Set Component Properties in Web UI** section of the **Attribute Help Text in Web UI** documentation

Display Context Help: Determines if the help text supplied in the Context Help parameter above should be available to users.

4. Click the **Add** button, the newly added component will display within the Table Headers / Headers parameter.
5. Click the **Save** button, and then click the **Close** button to close the designer.

Country Handling Solution

Within the automotive industry, it is vital to properly maintain parts data pertaining to country inclusion and/or exclusion to ensure market-specific coverage and legal compliance. The Country Handling solution helps users view, edit, and create accurate country handling inclusion and exclusion references for specific parts and applications. This Web UI solution also prevents users from creating conflicting inclusion and/or exclusion relationships.

The Country Handling solution can be used within an Application Manager Results Table via an 'Application Country Handling Table Header' component and within a Node Editor screen via a 'Country Handling Value' component.

This section addresses

- Prerequisites
- Using Country Handling
- Using Country Handling Value Editor to Add or Edit Country Handling Data
- Configuring Automotive - Country Model Component Model
- Configuring Application Country Handling Table Header
- Configuring Country Handling Value Component

Prerequisites

1. Apply the automotive recipe to the STEP instance to display the 'Automotive - Country Model' component model within Workbench > System Setup > Component Models.
2. Configure the 'Automotive - Country Model' component model.
For more information, see the **Configuring Automotive - Country Model Component Model** topic.
3. If needed, configure the 'Application Country Handling Table Header' component.
For more information, see the **Configuring Application Country Handling Table Header** topic.
4. If needed, configure the 'Country Handling Value' component.
For more information, see the **Configuring Country Handling Value Component** topic.

Additionally, when implementing this solution, it is recommended practice to create a 'Product Details' screen within a Web UI and configure it to display the Country Handling Value component when a product of a specified object type is selected within the Tree navigator. For more details on configuring screen mappings for this configuration, see the **Mappings** topic within the **Main Properties Overview** of **STEP Online Help**.

Using Country Handling

The Country Handling solution helps users view, edit, and create accurate country handling inclusion and exclusion references for specific parts and applications. This Web UI solution also prevents users from creating conflicting inclusion and/or exclusion relationships.

The Country Handling solution can be used within an Application Manager Results Table via an 'Application Country Handling Table Header' component and/or within a Node Editor screen via a 'Country Handling Value' component.

Prerequisites

Configuration is required before the Country Handling solution can be used. For more information, see the **Prerequisites** section of the **Country Handling Solution** topic.

Using Country Handling within Application Manager

When using the Country Handling solution within an Application Manager, the 'Application Country Handling Table Header' component can be configured to display the country inclusion data within the 'Country' column of the Results Table where users can view country handling data on applications. This component only displays inclusion data for the selected application. Therefore, it does not display inclusion nor exclusion data from the application's parent (i.e., Part). Double clicking a cell within the 'Application Country Handling Table Header' column will display a Value editor where users can add and/or remove country inclusion data.

In the example below, the Application Country Handling Table Header component is displayed with the default column label 'Countries.' In the first row of the Results Table, an application for an ACURA Coupe - NSX displays with 'All (0/249)' within the Countries column. This indicates that the between the configured exclusion and inclusion attributes, zero values are populated. In other words, for the possible 249 countries, no exclusion nor inclusions have been added to STEP. Therefore, the inclusion of all countries is assumed because no data exists in the exclusion and/or inclusion attributes.

The screenshot shows the application manager interface with the following components:

- Search Filters:**
 - Make/Model: ACURA
 - Engine: (empty)
 - Model: (empty)
 - Logic: AND
 - Buttons: Enter Make/Model, Enter Engine, Enter Model
- Global Filters:**
 - Acura Existing Applications
 - Existing Applications Only
 - All Brands
- Table Headers:**
 - Assembly
 - Generic Article
 - Supplier Article
 - Countries
- Table Data:**

Assembly	Generic Article	Supplier Article	Countries
<input type="checkbox"/> ACURA Coupe - NSX - 3.2 1997-08-01-2005-12-31, 206kW, 280HP, cmTech 3179	Ignition Coil	49065	All (0/249)
<input type="checkbox"/> ACURA Coupe - NSX - 3.2 1997-08-01-2005-12-31,	Ignition Coil	49064	All (247/249), ASEAN (10/10), Africa (53/53), Asia / Pacific (16/16), Baltic Countries (3/3), Benelux (3/3), Central America (8/8), Community of Independent States (7/7), Countries of the

To add or edit country inclusion / exclusion data to an application using the 'Application Country Handling Table Header' component, double click the desired cell within the 'Countries' column for the application row, and a Value editor dialog will display. For more information, on using the Value editor, see the **Using Country Handling Value Editor** topic.

Once the Value editor is used to edit the country handling data, the Application Manager Results Table will update to display the country data referenced for inclusion.

In the screenshot below, the cell that once displayed 'All (0/249)' now displays 'All (56/249),' and then lists the country group names with their count of selected countries / number of countries within the group. For example, five of the ten countries included within the ASEAN country group are displayed as selected because they are also included in the 'Left-hand traffic' country group that was selected.

	Assembly	Generic Article	Supplier Article	Countries
<input type="checkbox"/>	ACURA Coupe - NSX - 3.2 1997-08-01-2005-12-31, 206kW, 280HP, cmTech 3179	Ignition Coil	49065	All (56/249), ASEAN (5/10), Africa (13/53), Asia / Pacific (9/16), Countries of the European Union (4/25), Europe (4/40), Except Europe (38/120), Far East (7/13), Left-hand traffic (55/55), South America (2/12), Western Europe (Total) (2/20), Western Europe (without Germany) (2/18)

To remove any of the selected inclusion countries, double click on the cell to access the Value editor. Uncheck any country checkboxes, and click the OK button.

Important: Once a country value is added for inclusion, if it is then added to the exclusion attribute, it is automatically removed from inclusion attribute. The same country cannot exist in both the inclusion attribute and the exclusion attribute.

Using Country Handling within Node Editor

When using the Country Handling solution within a Node Editor, the 'Country Handling Value' component can be configured to display the country inclusion data within a Node Editor screen used to view and/or edit country handling data on parts and/or applications (i.e., Product Details screen).

When configured, this component can be used to view and edit the country inclusion data for applications and/or parts. Once an application and/or part object type is selected, then the Country Handling Value component displays only those countries currently referenced to the attribute or part for inclusion. It only displays inclusion data for the selected object. Therefore, it does not display inclusion nor exclusion data from the part's parent and/or child objects (i.e., Part Types, Applications).

In the example below, when part '49065' is selected within a Web UI Tree, a 'Product Details' screen displays a 'Country Inclusion / Exclusion' tab page where the Country Handling Value component displays 'All Countries (6/273),' indicating that six countries of the 273 available are referenced for inclusion. To the right of that heading, the list of individual country names referenced for inclusion display (Denmark, France, Germany, Great Britain, Norway, and Sweden).

The screenshot displays the 'Product Details' page for product 49065. The interface includes a navigation tree on the left, a breadcrumb trail, and a 'Selected Countries' section with a filter input and a list of country groups.

Tree

- ▶ 49060
- ▶ 49061
- ▶ 49062
- ▶ 49063
- ▶ 49064
- ▼ **49065**
 - ▶ HON
 - ▶ 49066
 - ▶ 49067
 - ▶ 49068
 - ▶ 49069
 - ▶ 49070
 - ▶ 49071
 - ▶ 49072
 - ▶ 49073
 - ▶ 49075
 - ▶ 49076
 - ▶ 49077
 - ▶ 49086
 - ▶ 49097
 - ▶ 49098

Product Details

Country Inclusion / Exclusion OE Applications Application Editor

Node Details

Primary Product Hierarchy > TecDoc Supplier Root > 0015 NGK Product Hierarchy > 0015 Articles > 4

Selected Countries

Filter countries

All Countries (6/273)	Denmark
Countries of the European Union (5/25)	France
Europe (6/40)	Germany
Left-hand traffic (1/55)	Great Britain
Scandinavia (3/4)	Norway
Western Europe (Total) (6/20)	Sweden
Western Europe (without Germany) (5/18)	

When the component first displays, the 'All Countries (number of countries referenced for inclusion / total number of countries)' group heading displays using bold text and highlighted in blue. Each country group containing a country referenced for inclusion will display in bold below the 'All Countries' heading. Because each country group heading displays the number of countries referenced for inclusion out of the total number of countries within the group, it is easy to get an understanding of the country groups that contain referenced countries for inclusion.

Clicking any one of the country group names will move the blue highlight to that header and display the countries referenced for inclusion within that group to the right of the header. In the example below, the 'Left-hand traffic' country group is selected and its one country referenced for inclusion (Great Britain) displays to the right. From this display, it is easy to see not only the specific country groups that the six countries referenced for inclusion belong to, but also, of the six countries referenced for inclusion, only one is included in the 'Left-hand traffic' group that includes 55 countries.

Selected Countries


✎

All Countries (6/273)	Great Britain
Countries of the European Union (5/25)	
Europe (6/40)	
Left-hand traffic (1/55)	
Scandinavia (3/4)	
Western Europe (Total) (6/20)	
Western Europe (without Germany) (5/18)	

When country inclusion and/or exclusion data is not provided, then the component will display 'No countries selected' (as shown below). The edit button (✎) can be used to add country handling data.

✎

No countries selected




To get started, use the pen icon to add or remove countries.


To search for a specific country within any of the listed country groups with inclusion data, type text into the 'Filter countries' text box and the results will display country names that match the typed text to the right of the country group headers.

In the example below, 'den' is typed into the 'Filter countries' text box and where the component once displayed '6/273' it now displays '1/273.' To the right of the country group headers, the filter result (Denmark) displays. Each country group that contains an inclusion filter results match displays in bold. Whereas, any country groups that do not contain a filter result match ('Left-hand traffic') display in gray text.

Selected Countries	
<input type="text" value="den"/>	
All Countries (1/273)	Denmark
Countries of the European Union (1/25)	
Europe (1/40)	
Left-hand traffic (0/55)	
Scandinavia (1/4)	
Western Europe (Total) (1/20)	
Western Europe (without Germany) (1/18)	

When the text typed into the 'Filter countries' text box does not match any of the country names referenced for inclusion, then the 'Country not found' message will display in the results (as shown below).


<input type="text" value="ind"/> 	
<p>All Countries (0/273)</p> <p>Countries of the European Union (0/25)</p> <p>Europe (0/40)</p> <p>Left-hand traffic (0/55)</p> <p>Scandinavia (0/4)</p> <p>Western Europe (Total) (0/20)</p> <p>Western Europe (without Germany) (0/18)</p>	<p>Country not found</p> <p>The country you are looking for cannot be found. You can add and remove countries by clicking the pen icon.</p>

To add or edit country inclusion / exclusion data to an application or part using the 'Country Handling Value' component, click the edit button () to the right of the Filter countries field, and a Value editor dialog will display. For more information on using the Value editor, see the **Using Country Handling Value Editor to Add or Edit Country Handling Data** topic.

Using Country Handling Value Editor to Add or Edit Country Handling Data

Whether the 'Application Country Handling Table Header' or 'Country Handling Value' component is used, the Value editor functions the same.

To add or edit country inclusion / exclusion data to an application using the 'Application Country Handling Table Header' component, double click the desired cell within the 'Countries' column for the application row, and a Value editor dialog will display as shown below. For more information, see the **Configuring Application Country Handling Table Header** topic.

To add or edit country inclusion / exclusion data to an application or part using the 'Country Handling Value' component, click the edit button () to the right of the Filter countries field, and a Value editor dialog will display as shown below. For more information, see the **Configuring Country Handling Value Component** topic.

Value editor - 1 item selected ✕

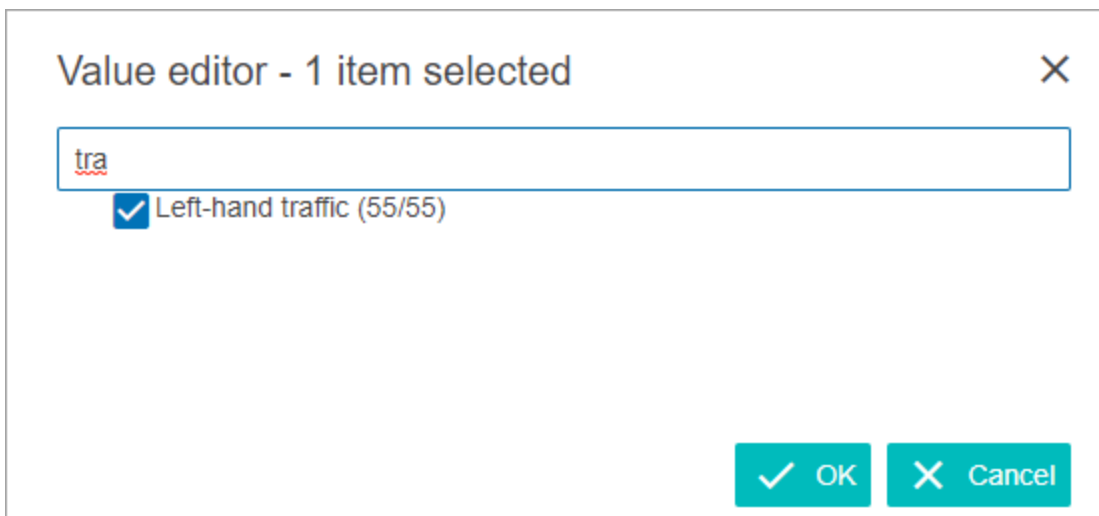
- ▼ All Countries (0/273)
 - ▶ ASEAN (0/10)
 - ▶ Africa (0/53)
 - ▶ Asia / Pacific (0/16)
 - ▶ Baltic Countries (0/3)
 - ▶ Benelux (0/3)
 - ▶ Central America (0/8)
 - ▶ Community of Independent States (0/7)
 - ▶ Countries of the European Union (0/25)
 - ▶ Eastern Europe (0/7)
 - ▶ Europe (0/40)
 - ▶ Except Europe (0/120)
 - ▶ Far East (0/13)
 - ▶ Gulf Countries (0/6)
 - ▶ Iberian Peninsula (0/2)
 - ▶ Left-hand traffic (0/55)
 - ▶ Middle East (0/17)
 - ▶ Middle East (0/4)
 - ▶ NAFTA (0/3)
 - ▶ North Africa (0/4)
 - ▶ North America (0/2)
 - ▶ Scandinavia (0/4)
 - ▶ South America (0/12)

Note: The display of the countries and their groupings vary based upon the configuration of the Automotive - Country Model component model. For more information, see the **Configuring Automotive - Country Model Component Model** topic.

Using the 'Filter Countries' typeahead field, users can type in the first letters of any word within the country name to quickly find the country for selection. Otherwise, scroll through the list of country groups and click on the black arrow heads to expand the groups to view the individual countries within each one. Notice to the right of each country group name, the parentheses show the count of selected countries / number of countries within the group.

When selecting a country group name, the child hierarchy will also automatically be selected. This selection is represented with a blue and white checkmark (☑). If only one country is listed within a country group, and the country is selected, then the country group will also be selected. Otherwise, the country group name displays with a blue and white dash icon (▬) for the group. Additionally, if the selected country belongs to more than one country group, then it will display as checked below each of its country groups. Clicking the blue and white dash icon deselects the country or all countries within the group.

In the example below, 'tra' is entered into the 'Filter Countries' typeahead field, and the country group 'Left-hand traffic (55/55)' is selected and displays with a blue and white checkmark (☑).

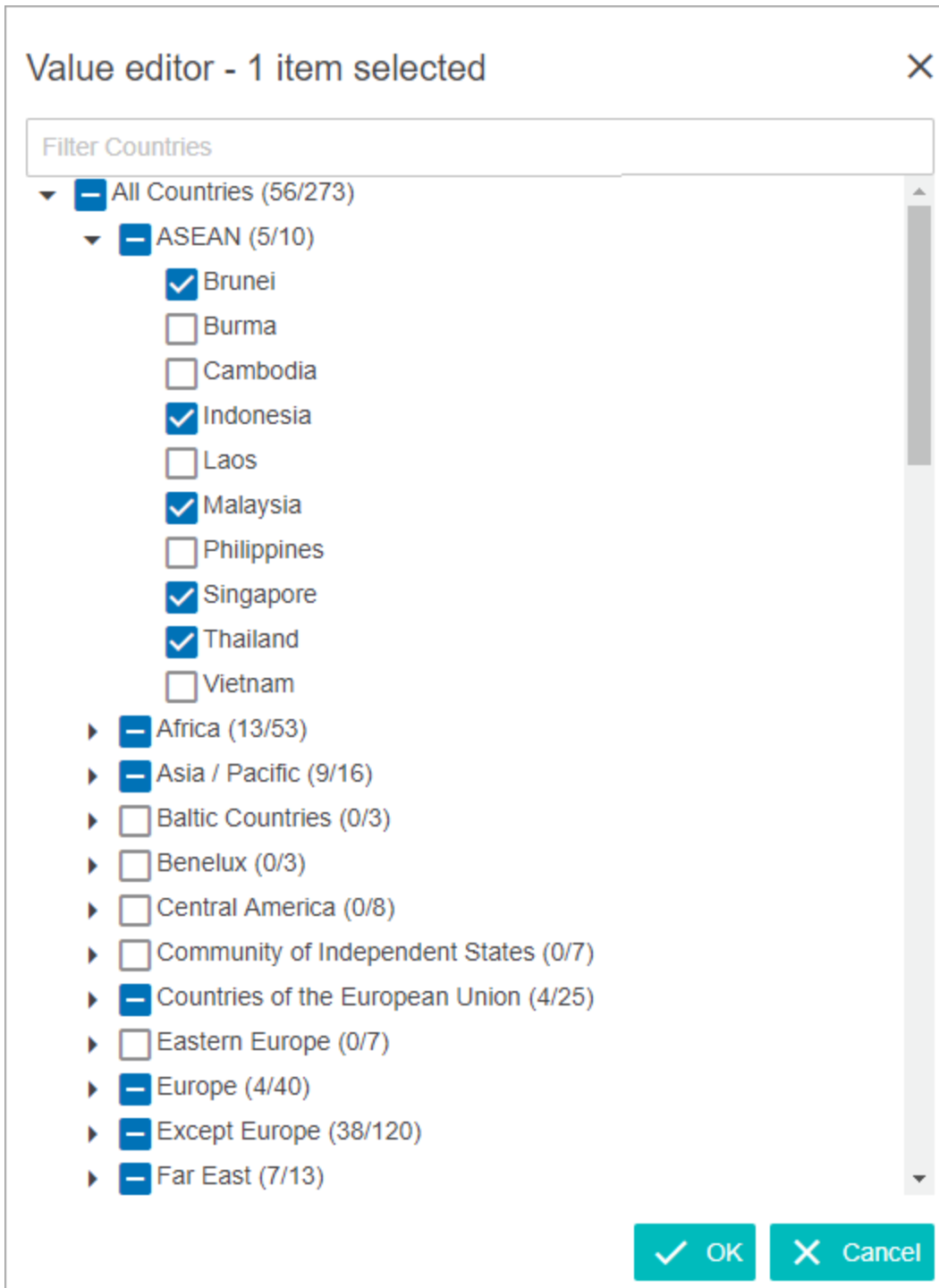


Clicking the OK button closes the dialog and displays the 55 countries selected from the 'Left-hand traffic' country group.

However, as shown in the following example, it is also possible to clear the previously entered text and either enter new search text or view the individual countries selected.

In the example below, the 'tra' text was removed from the 'Filter Countries' field using the backspace key, and the selected countries from the 'Left-hand traffic' country group display selected within their additional country groups.

The blue and white dash icon (▬) is used when only some of the countries in a country group are selected. Whereas, the blue and white checkmark icon (☑) is used when all the countries within a group are selected. Clicking the checkmark or empty checkbox toggles between selecting all countries and selecting none.



Note: The OK button is only enabled when there are changes to be saved.

Once the desired countries for inclusion are selected, click the OK button to close the dialog and return to the previous screen.

Clearing the checkbox for any of the countries or groups on the list will remove them from inclusion.

Configuring Automotive - Country Model Component Model

Once a STEP system is setup to display the 'Automotive - Application Model' component model, then the 'Automotive - Country Model' component model will also display within Workbench > System Setup > Component Models.

The component model will need to be manually configured. In the example below, the standard TecDoc object types, LOV, and Reference Type have been used to configure the component model values.

Name	Value	Description
> Country	Country	Countries and country groups
	Country Group	Countries and country groups
> Country LOV link	Country Code (single value)	Country LOV link
> Country group link	Country To Country Group	Country group link

Note: Click the **Edit** link located below the component parameters to add or remove values for the parameters. All of the parameters must be properly populated for the Country Handling solution to function as expected.

The 'Automotive - Country Model' component model provides the following required parameters:

- **Country:** Required parameter used to define the Country and/or Country groups to be used within the Country Handling solution. Add one or more Country and/or Country groups. In the example above, the following values are added: 'Country' (TD_Country) and 'Country Group' (TD_CountryGroup). Though the label is 'Country' (EU can be considered a country, and DK can be considered a country) This solution only considers leaf countries. Countries are modeled within Country Groups via the Classification files and the 'Country to Country Group' (TD_CountryToCountryGroup) Reference Type. The classification folders and references are created upon import of the TD Reference Data importer.

Important: The Country and Country groups classifications must be configured with the Reference Type configured within the 'Country group link' parameter or the solution will not work.

- **Country LOV link:** Required parameter used to define the one LOV list of countries to be used for both the inclusion and exclusion attributes. It is important to only use one LOV list as this helps to guarantee the same country will not be referenced for both inclusion and exclusion attributes. The LOV list should be single valued and use LOV Value ID's. In the example above, the following LOV value is added: 'Country Code (single value)' (TD_ATTR_CountryCode).

Important: The 'Country Code (single value)' (TD_ATTR_CountryCode) LOV must contain the country codes of all the countries and/or country groups to be displayed, and the LOV must be made valid for both the inclusion and exclusion attributes.

- **Country group link:** Required parameter used to define the countries that should be listed within each Country group. In the example above, the following value is added: 'Country to Country Group' (TD_CountryToCountryGroup).

To apply this concept to more than one Country Handling solution (i.e., OWN model), Country LOV Link attribute, Country Object Type, and Country group link values need to be configured separate from those used for the TecDoc or AutoCare solutions.

Once the Automotive - Country Model component model is configured, the 'Application Country Handling Table Header' and/or 'Country Handling Value' components need to be configured. For more information, see the **Configuring Application Country Handling Table Header**, and the **Configuring Country Handling Value Component** topics.

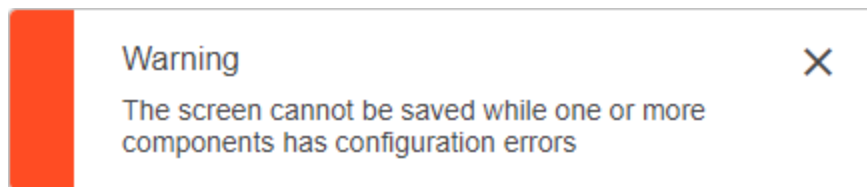
Configuring Application Country Handling Table Header

When configured, the 'Application Country Handling Table Header' component can be used to display the country handling data within an Application Manager Results Table. Additionally, when a cell within the Table Header is clicked, a Value editor will display allowing users to edit country handling data.

Prerequisites

Configuration of the Automotive - Country Model component model is required before the Application Country Handling Table Header component can be configured. For more information, see the **Prerequisites** section of the **Country Handling Solution** topic.

Attempting to add the Application Country Handling Table Header component to a Web UI prior to configuring the Automotive - Country Model component model will result in the following warning: The screen cannot be saved while one or more components has configuration errors.



Configuration Steps

Below are steps to configure the Application Country Handling Table Header component within a Node Editor. Each of the parameters for the Application Country Handling Table Header Properties are described below.

1. Using the Web UI designer as shown in the image below, go to Node List Properties Headers parameter for the Application Manager screen that needs to display country handling data.
2. Click the **Add** button for the Headers parameter, and the Add Component dialog will display.
3. Find and select the **Application Country Handling Table Header** component.
4. Click the **Add** button, and the Application Country Handling Table Header Properties dialog will display (as shown below).

The screenshot illustrates the process of adding a component to a configuration. It is divided into three main sections:

- Properties Panel (Top):** Shows the 'Node List Properties' for 'TecDoc Application I'. A dropdown menu is open, listing various components. The 'Add...' button is highlighted with a red circle and a number '2'. A red arrow points from this button to the 'Add Component' dialog.
- Add Component Dialog (Middle):** A dialog box with a list of components. 'Application Country Handling Table Header' is selected and highlighted with a red circle and a number '3'. A red arrow points from this selection to the 'Add' button at the bottom right, which is also highlighted with a red circle and a number '4'.
- Add component - configure required properties (Bottom):** A configuration dialog for the selected component. It contains several fields: 'Dimensions' (dropdown), 'Exclude Attribute' (text input), 'Include Attribute*' (text input), 'Label' (text input), and 'Table Sorting' (dropdown). At the bottom right, there are 'Add' and 'Cancel' buttons.

5. Populate the required parameters listed below.

- **Include Attribute:** This required parameter is used to select a multivalued attribute where the country inclusion data will be stored. **Click the ellipsis button (...)** to browse and/or search for the necessary attribute value.
- For the TecDoc solution, this may be 'TD_ATTR_LKZ_Include'

Important: When an attribute value selection is made within the Country Handling Value editor, the referenced attribute values are stored within the configured inclusion attribute. However, when an attribute value is deselected from inclusion, it is **NOT** added to the exclusion attribute. When this occurs, the assumption is made that desired attribute values for inclusion are specified within the inclusion attribute, and therefore all others are excluded.

6. Optionally, populate the other parameters listed below.

- **Dimensions:** By default, the parameter is blank. Optionally, to configure column width and height, select 'Table Header Dimensions.'
- **Exclude Attribute:** This required parameter is used to select a multivalued attribute where the country exclusion data is stored. Attribute values (country names) within this attribute will not be displayed as included within the country handling components. Click the ellipsis button (...) to browse and/or search for the necessary attribute value.
- For the TecDoc solution, this may be 'TD_ATTR_LKZ_Exclude'

Important: A value cannot exist in both the configured exclude attribute and in the include attribute. If a value is stored within the exclusion attribute, and is later selected for inclusion within the Country Handling Value editor, then the referenced value is removed from the exclusion attribute. However, if that attribute value is deselected from inclusion, it is **NOT** added back to the exclusion attribute. When this occurs, the assumption is that desired attribute values for inclusion are specified within the inclusion attribute, and therefore all others are excluded.

- **Label:** By default, the parameter is blank. But after clicking the **Add** button, the default column header label (Countries) will be stored. Optionally enter a desired label to be displayed as the column header within the Application Manager Results Table.
 - **Table Sorting:** By default, the parameter is blank. Optionally, select from the dropdown values to specify the default sorting order of the column data.
7. Once the required parameters are populated, the Add button will activate. Click the **Add** button to save the newly added component to the Headers parameter.
8. Optionally, on the Node List Properties Headers parameter, use the **Up** and/or **Down** buttons to configure the order in which the newly added column should display within the Results Table.
9. Click the **Save** and **Close** buttons for the designer.

Configuring Country Handling Value Component

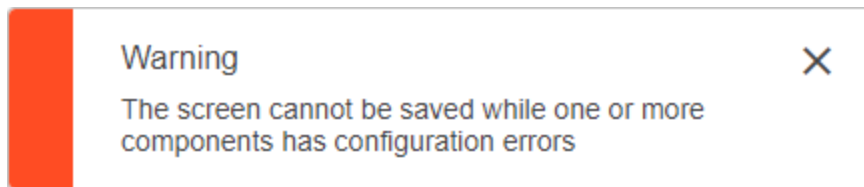
When using the Country Handling solution within a Node Editor, the 'Country Handling Value' component can be configured to display the country inclusion / exclusion data within a Node Editor screen used to view and/or edit country handling data on parts and/or applications (i.e., Product Details screen).

When configured, this component can be used to view and edit the country inclusion data for applications and/or parts. Once an application and/or part object type is selected, then the Country Handling Value component displays only for those countries currently referenced to the attribute or part for inclusion. It only displays inclusion and exclusion data for the selected object. Therefore it does not display any inclusion nor exclusion data from the part's parent and/or child objects (i.e., Part Types, Applications).

Prerequisites

Configuration of the Automotive - Country Model component model is required before the Country Handling Value component can be configured. For more information, see the **Prerequisites** section of the **Country Handling Solution** topic.

Attempting to add the Country Handling Value component to a Web UI prior to configuring the Automotive - Country Model component model will result in the following warning: The screen cannot be saved while one or more components has configuration errors.



Configuration Steps

Below are steps to configure the Country Handling Value component within a Node Editor. Each of the parameters for the Country Handling Value Properties are described below.

1. Using Web UI designer, go to Node Editor Properties Child Components Rows parameter for the screen that needs to display country handling data.
2. Click the **Add** button for the Child Components Rows parameter, and the Add Component dialog will display.
3. Find and select the **Country Handling Value** component, click the **Add** button, and the Country Handling Value Properties dialog will display (as shown below).

Add component - configure required properties ✕

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

Country Handling Value Properties

Context Help

Exclude Attribute ... Clear

Include Attribute * ...

Included By Default

Label

✓ Add
✕ Cancel

4. Populate the required parameters listed below.

- **Exclude Attribute:** This required parameter is used to select a multivalued attribute where the country exclusion data is stored. Attribute values (country names) within this attribute will not be displayed as included within the country handling components. Click the ellipsis button (...) to browse and/or search for the necessary attribute value.
- For the TecDoc solution, this may be 'TD_ATTR_LKZ_Exclude'

Important: A value cannot exist in both the configured exclude attribute and in the include attribute. If a value is stored within the exclusion attribute, and is later selected for inclusion within the Country Handling Value editor, then the referenced value is removed from the exclusion attribute. However, if that attribute value is deselected from inclusion, it is **NOT** added back to the exclusion attribute. When this occurs, the assumption is that desired attribute values for inclusion are specified within the inclusion attribute, and therefore all others are excluded.

- **Include Attribute:** This required parameter is used to select a multivalued attribute where the country inclusion data will be stored. **Click the ellipsis button (...)** to browse and/or search for the necessary attribute value.
- For the TecDoc solution, this may be 'TD_ATTR_LKZ_Include'

Important: When an attribute value selection is made within the Country Handling Value editor, the referenced attribute values are stored within the configured inclusion attribute. However, when an attribute value is deselected from inclusion, it is **NOT** added to the exclusion attribute. When this occurs, the

assumption is made that desired attribute values for inclusion are specified within the inclusion attribute, and therefore all others are excluded.

5. Optionally, populate the other parameters listed below.
 - **Context Help:** By default, the parameter is blank. But after clicking the **Add** button, the default label (Selected Countries) will be stored. Optionally, enter desired text to be displayed when a user hovers over the component label.
 - **Label:** By default, the parameter is blank. But after clicking the **Add** button, the default label (Selected Countries) will be stored. Optionally, enter a desired label to be displayed within the Node Editor and to the left of the component.
6. Once the required parameters are populated, the Add button will activate. Click the **Add** button to save the newly added component to the Headers parameter.
7. Optionally, on the Node Editor Properties Child Components Rows parameter, use the **Up** and/or **Down** buttons to configure the order in which the newly added component should display within the node editor.
8. Click the **Save** and **Close** buttons for the designer.

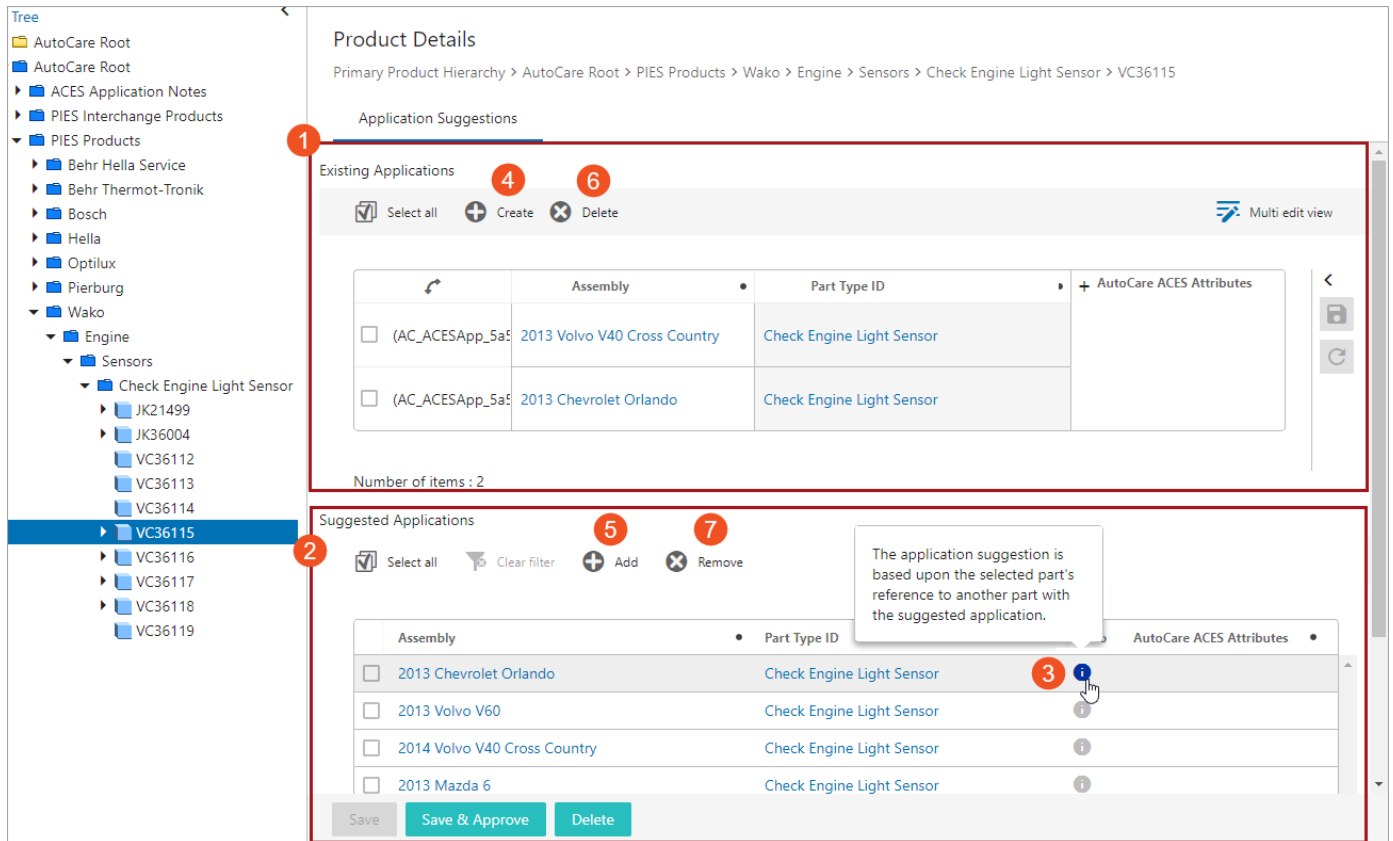
Application Suggestion Solution

With the automotive solution, it is easy to maintain application data. The automotive solution not only helps users more easily view, edit, and create applications for specific parts, but also suggest potential matches. In other words, when configured, STEP can suggest potential matches for applications based on existing applications within STEP. This occurs when a part is selected, STEP looks at the existing applications and displays their attribute values via configurations within the 'Part Application Editor' Node Editor component.

When the new Part Application Editor and/or Application Suggestions components are configured as shown below, Web UI users can:

1. View existing applications and related data.
2. View Target (competitor and/or OE part number) part's application suggestions and related data.
3. View information about the suggested application.
4. Add applications manually.
5. Add applications from suggestions.
6. Remove existing applications.
7. Remove applications from suggestions.

Note: The components used for this solution are extremely flexible and can be configured to display the existing applications on the same screen. Also, additional column headers can be added to the components to display more information than is shown in this example.



In the example above, part VC36115 is selected within a Web UI Tree, and a Web UI screen (Product Details) is configured to display the existing and the potential application on a tab page (Application Suggestions).

This section addresses:

- Prerequisites
- Adding New Applications Manually
- Adding Applications from Suggestions
- Removing Existing Applications
- Removing Application from Suggestions List
- Configuring the Part Application Editor Component
- Configuring the Application Suggestions Component

Prerequisites

When implementing this solution, it is suggested to create a 'Product Details' screen within a Web UI and configure it to display when a product is selected within the Tree navigator. The Product Details screen should be created via a Node Details screen with a Tab Control that is configured to display a 'Application Suggestion' Tab Page. Within each of these Tab Pages a Node Editor can be configured with 'Part Application Editor' and/or

'Application Suggestions' components. The 'Part Application Editor' component can be configured to display existing applications. Whereas, the 'Application Suggestions' component can be configured to display suggested applications.

For more details on configuring screen mappings so that a 'Product Details' screen will display when a product is selected within the Tree navigator, see the **Mappings** topic within the **Main Properties Overview** section of Web User Interfaces / Web UI Setup and User Guide in **STEP Online Help**.

Adding New Applications Manually

Users can easily add part applications to a selected part within Web UI. When the Part Application Editor component is configured and a part is selected, STEP displays the existing application records and its attribute values based on the configurations made within the 'Part Application Editor' Node Editor component. Whether or not the application records are displayed, at a minimum a 'Create' button will display. This 'Create' button can be used to manually add a new application record to the selected part.

In the example below, part VC36115 is selected within the Web UI Tree, and an 'Application Suggestions' tab is configured to display existing applications. Within the results, two applications are displayed along with the 'Create' button that allows users to create new application records.

The screenshot displays the 'Product Details' page for part VC36115. The left sidebar shows a tree view with 'VC36115' selected. The main area shows 'Application Suggestions' with a table of existing applications and a 'Create' button highlighted in a red box.

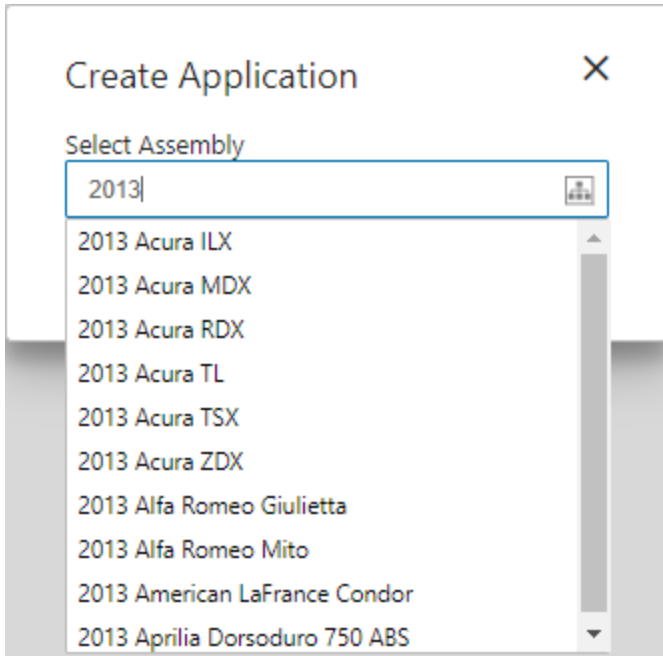
	Assembly	Part Type ID
<input type="checkbox"/> VC36115 2013 Volvo V40 Cross Country	2013 Volvo V40 Cross Country	Check Engine Light Sensor
<input type="checkbox"/> VC36115 2013 Chevrolet Orlando	2013 Chevrolet Orlando	Check Engine Light Sensor

When a user clicks on the 'Create' button, a Create Application dialog displays prompting users to select the vehicles / assemblies. Once the vehicle / assembly name (or ID) is typed in the typeahead field, available list of vehicles / assemblies will display in the dropdown list. Alternatively, the user can select the Node Picker icon available in the right of the typeahead field to select the vehicles / assemblies. Selecting a listed base vehicle will add the application to the existing applications list. The user can select multiple vehicles / assemblies in the dialog to create more than one application records simultaneously.

Following are the IDs of vehicles / assemblies that are valid to be selected across different standards.

- AutoCare: AC_BaseVehicle
- NAPA: NAPA_Year
- TecDoc: TD_VehicleType(PC) / TD_VehicleType(CV) / TD_EngineCode

In the example below, '2013' is typed into the typeahead field and the application records that begin with 2013 are displayed within the dropdown list.



Selecting 2013 Acura ILX and clicking on **OK** adds a new application record to the existing list of applications, as shown below.

Product Details

Application Suggestions

Existing Applications

Select all Create Multi edit view

	Assembly	Part Type ID	
<input type="checkbox"/> VC36115 2013 Volvo V40 Cross Country	2013 Volvo V40 Cross Country	Check Engine Light Sensor	
<input type="checkbox"/> VC36115 Application Suggestion - Check Engine Light Sensor - 2013 Acura ILX	2013 Acura ILX	Check Engine Light Sensor	
<input type="checkbox"/> VC36115 2013 Chevrolet Orlando	2013 Chevrolet Orlando	Check Engine Light Sensor	

Number of items : 3

If there is an Application Suggestions component configured below the existing applications list, then the suggestions list too is updated as new application records are added to the existing applications list.

Adding Applications from Suggestions

Users can add an existing application record of another part to a selected part from the Application Suggestions list within Web UI. When the Part Application Editor and the Application Suggestions components are configured and a part is selected, STEP displays the existing application records and also suggests the potential application records via configurations within the configured Node Editor component.

When any potential application records are displayed in the Application Suggestions list, selecting an application record from the suggestion list will activate and display the **Add** button (+) at the component tool bar. Clicking the **Add** button will add the selected application record(s) to the Part Application Editor list.

Clicking the **Remove** button (X) will remove the suggested application record from the Application Suggestions list. For more information, see the **Removing Application Suggestions** topic within this section.

In the example below, part 'VC36115' is selected within the Web UI Tree, and the 'Application Suggestions' tab is configured to display results for 'Existing Applications' and 'Suggested Applications.' Within the Suggested Applications list, four application records display along.

Tree

- AutoCare Root
 - AutoCare Root
 - ACES Application Notes
 - PIES Interchange Products
 - PIES Products
 - Behr Hella Service
 - Behr Thermot-Tronik
 - Bosch
 - Hella
 - Optilux
 - Pierburg
 - Wako
 - Engine
 - Sensors
 - Check Engine Light Sensor
 - VC36009
 - VC36112
 - VC36113
 - VC36114
 - VC36115**
 - VC36116
 - VC36117
 - VC36118

Product Details

Application Suggestions

Existing Applications

Clear all Create Delete

	Assembly	Part Type ID
<input type="checkbox"/> VC36115 2013 Volvo V40 Cross Country	2013 Volvo V40 Cross Country	Check Engine Light Sensor
<input type="checkbox"/> VC36115 2013 Chevrolet Orlando	2013 Chevrolet Orlando	Check Engine Light Sensor

Number of items : 2

Suggested Applications

Clear all Clear filter Add Remove

Assembly	Part Type ID	Info
<input checked="" type="checkbox"/> 2013 Chevrolet Orlando	Check Engine Light Sensor	<i>i</i>
<input type="checkbox"/> 2013 Volvo V60	Check Engine Light Sensor	<i>i</i>
<input type="checkbox"/> 2014 Volvo V40 Cross Country	Check Engine Light Sensor	<i>i</i>
<input type="checkbox"/> 2013 Mazda 6	Check Engine Light Sensor	<i>i</i>

Advanced search Save Save & Approve Delete

With an application record (2013 Chevrolet Orlando) selected, clicking the **Add (+)** button available at the toolbar of the suggestions list will add the '2013 Chevrolet Orlando' application record to the Existing Applications list (as shown below).

Tree

- ▶ AutoCare Root
- ▼ AutoCare Root
 - ▶ ACES Application Notes
 - ▶ PIES Interchange Products
 - ▼ PIES Products
 - ▶ Behr Hella Service
 - ▶ Behr Thermot-Tronik
 - ▶ Bosch
 - ▶ Hella
 - ▶ Optilux
 - ▶ Pierburg
 - ▼ Wako
 - ▼ Engine
 - ▼ Sensors
 - ▼ Check Engine Light Sensor
 - ▶ JK21499
 - ▶ JK36004
 - ▶ JK36009
 - ▶ VC36009
 - ▶ VC36112
 - ▶ VC36113
 - ▶ VC36114
 - ▶ VC36115
 - ▶ VC36116
 - ▶ VC36117
 - ▶ VC36118

Advanced search

Product Details

Application Suggestions

Existing Applications

Clear all Create Delete

	Assembly	Part Type ID
<input type="checkbox"/>	2013 Volvo V40 Cross Country	Check Engine Light Sensor
<input type="checkbox"/>	2013 Chevrolet Orlando	Check Engine Light Sensor
<input type="checkbox"/>	2013 Chevrolet Orlando	Check Engine Light Sensor

Number of items : 3

Suggested Applications

Select all Clear filter

	Assembly	Part Type ID	Info
<input type="checkbox"/>	2013 Volvo V60	Check Engine Light Sensor	
<input type="checkbox"/>	2014 Volvo V40 Cross Country	Check Engine Light Sensor	
<input type="checkbox"/>	2013 Mazda 6	Check Engine Light Sensor	

Save Save & Approve Delete

Once the application record is added to the selected part and now displays within the Part Application Editor table, the same application record will no longer display within the Application Suggestions list.

Removing Existing Applications

Users can remove an existing application record for a selected part within Web UI. When the Part Application Editor component is configured, and an application record is selected within the Existing Applications list, a **Delete** button (✕) displays at the component toolbar. Clicking the **Delete** button (✕) will delete the application record for the selected part, and therefore, the application record will no longer display within the list.

In the example below, part 'VC36115' is selected within the Web UI Tree, and an 'Applications Suggestions' tab is configured to display results for 'Existing Applications.' Within the results, two application records are displayed. An application record 'VC36115 2013 Volvo V40 Cross Country' is selected which activates and displays **Delete** (✕) button thereby allowing users to remove that application record from the selected part.

	Assembly	Part Type ID
<input checked="" type="checkbox"/> VC36115 2013 Volvo V40 Cross Country	2013 Volvo V40 Cross Country	Check Engine Light Sensor
<input type="checkbox"/> VC36115 2013 Chevrolet Orlando	2013 Chevrolet Orlando	Check Engine Light Sensor

Clicking the Delete (✕) button at the component toolbar deletes the application record and the application record no longer displays within the Existing Applications grid (as shown below).

The screenshot displays the 'Product Details' page in StiboSystems. On the left is a 'Tree' view showing a hierarchical structure of folders: AutoCare Root, ACES Application Notes, PIES Interchange Products, PIES Products, Behr Hella Service, Wako, Engine, and Sensors. Under Sensors, 'Check Engine Light Sensor' is expanded, showing sub-items like JK21499, VC36004, VC36009, VC36112, VC36113, VC36114, VC36115 (highlighted), VC36116, and VC36117.

The main area is titled 'Product Details' and contains 'Application Suggestions' and 'Existing Applications' sections. The 'Existing Applications' section has a toolbar with 'Clear all', 'Create', and 'Multi edit view' buttons. Below the toolbar is a table with the following data:

	Assembly	Part Type ID
<input type="checkbox"/> VC36115 2013 <input type="checkbox"/> Chevrolet Orlando	2013 Chevrolet Orlando	Check Engine Light Sensor

At the bottom of the table area, it says 'Number of items : 1'.

If an application record is removed from the Existing Applications list, and it previously displayed within the Suggested Applications list, then the deleted application record will display within the suggestions list. However, if the removed application record was not previously displayed within the suggestions list (i.e., manually added), then it will not display within the suggestions list.

Removing Application from Suggestions List

Users can remove the suggested application record from the suggested list for a selected part within Web UI. When the Application Suggestions component is configured and an application record is selected within the suggested application list, a **Remove** button (X) displays at the component tool bar. Clicking the **Remove** button will remove the suggested application record from the Suggested Applications list, and therefore, the application record will no longer display within the list.

In the example below, part 'VC36115' is selected within the Web UI Tree, and the 'Application Suggestions' tab is configured to display results for 'Existing Applications' and 'Suggested Applications.' Within the results, in the 'Suggested Applications' list, four application records are displayed. An application record '2013 Chevrolet Orlando' is selected which activates and displays **Remove** (X) button thereby allowing users to remove an application record for the selected part.

The screenshot displays the 'Application Suggestions' interface. On the left, a tree view shows the hierarchy: AutoCare Root > PIES Products > Engine > Sensors > Check Engine Light Sensor > VC36115. The main area is titled 'Product Details' and 'Application Suggestions'. It is divided into two sections: 'Existing Applications' and 'Suggested Applications'.

Existing Applications: This section contains a toolbar with 'Clear all', 'Create', and 'Delete' buttons. Below it is a table with columns for 'Assembly' and 'Part Type ID'. Two records are listed:

	Assembly	Part Type ID
<input type="checkbox"/> VC36115 2013 Volvo V40 Cross Country	2013 Volvo V40 Cross Country	Check Engine Light Sensor
<input type="checkbox"/> VC36115 2013 Chevrolet Orlando	2013 Chevrolet Orlando	Check Engine Light Sensor

Below the table, it indicates 'Number of items : 2'.

Suggested Applications: This section contains a toolbar with 'Clear all', 'Clear filter', 'Add', and 'Remove' buttons. Below it is a table with columns for 'Assembly', 'Part Type ID', and 'Info'. Four records are listed:

	Assembly	Part Type ID	Info
<input checked="" type="checkbox"/>	2013 Chevrolet Orlando	Check Engine Light Sensor	i
<input type="checkbox"/>	2013 Volvo V60	Check Engine Light Sensor	i
<input type="checkbox"/>	2014 Volvo V40 Cross Country	Check Engine Light Sensor	i
<input type="checkbox"/>	2013 Mazda 6	Check Engine Light Sensor	i

At the bottom of the interface, there are three buttons: 'Save', 'Save & Approve', and 'Delete'. A red arrow points to the 'Remove' button in the 'Suggested Applications' toolbar.

In the example, after removing the application record '2013 Chevrolet Orlando' from part 'VC36115' the Suggested Applications list displays as shown below.

Tree

- AutoCare Root
 - AutoCare Root
 - ACES Application Notes
 - PIES Interchange Products
 - PIES Products
 - Behr Hella Service
 - Behr Thermot-Tronik
 - Bosch
 - Hella
 - Optilux
 - Pierburg
 - Wako
 - Engine
 - Sensors
 - Check Engine Light Sensor
 - VC36009
 - VC36112
 - VC36113
 - VC36114
 - VC36115**
 - VC36116
 - VC36117
 - VC36118

Product Details

Application Suggestions

Existing Applications

Clear all Create Delete

	Assembly	Part Type ID
<input type="checkbox"/>	VC36115 2013 Volvo V40 Cross Country	Check Engine Light Sensor
<input type="checkbox"/>	VC36115 2013 Chevrolet Orlando	Check Engine Light Sensor

Number of items : 2

Suggested Applications

Clear all Clear filter Add Remove

	Assembly	Part Type ID	Info
<input type="checkbox"/>	2013 Volvo V60	Check Engine Light Sensor	<i>i</i>
<input type="checkbox"/>	2014 Volvo V40 Cross Country	Check Engine Light Sensor	<i>i</i>
<input type="checkbox"/>	2013 Mazda 6	Check Engine Light Sensor	<i>i</i>

Advanced search

Save Save & Approve Delete

If the application record is removed from the Suggested Applications list, then the unlinked application record will no longer display within the Suggested Applications list.

Configuring the Part Application Editor Component

The Part Application Editor component is a Node Editor component used to display existing application records and their attribute values when a part number is selected within the Tree navigator.

This section describes how to configure the Part Application Editor component. For information about how to use this component, and prerequisites, see the **Application Suggestion Solution** topic.

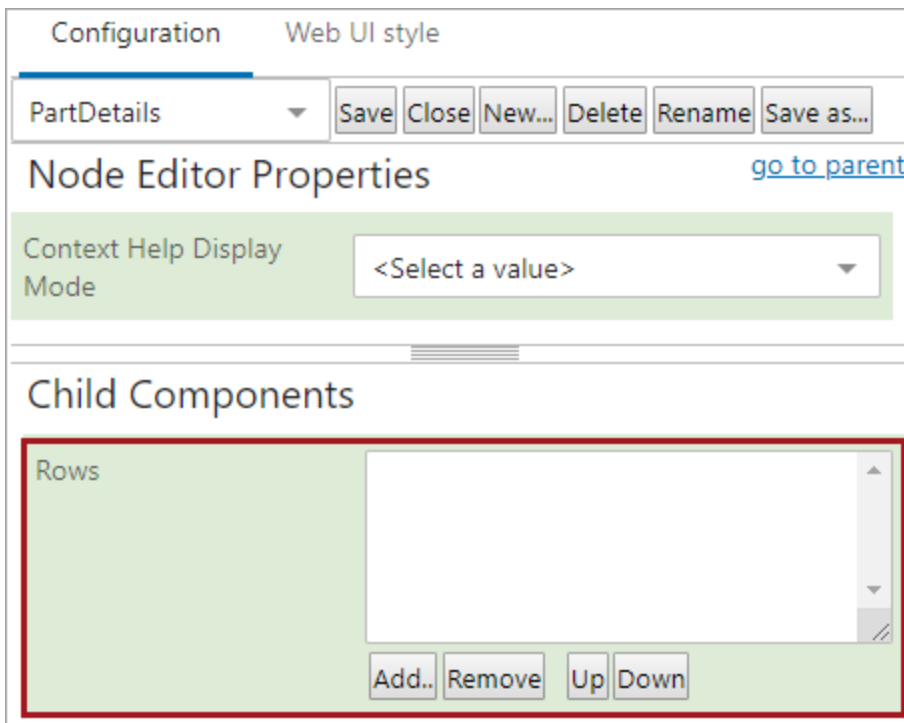
Prerequisites

Anyone configuring the Part Application Editor component within a Node Editor is expected to be familiar with the basic configuration and functionalities of Node Editor Component as basic concepts for working with Node Editor component are not covered in this section. For more information, see the **Node Editor Component** topic within the **Using a Web UI** section of **STEP Online Help**.

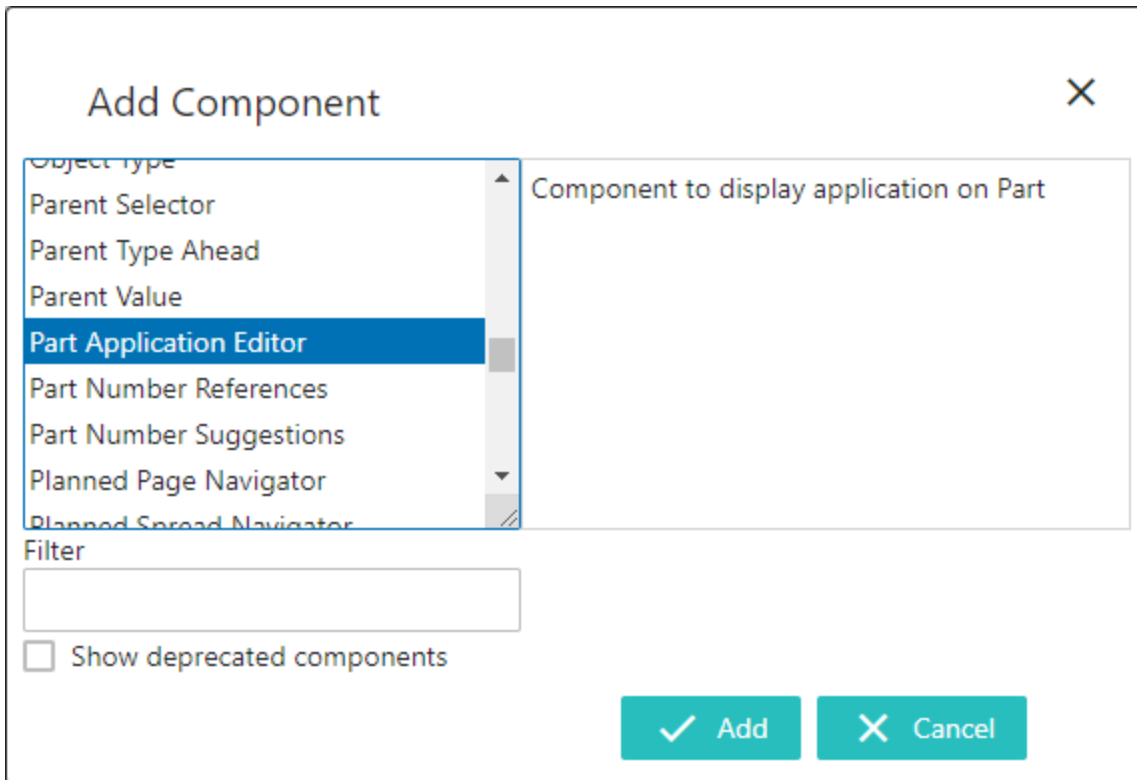
Configuration Steps

Below are the steps to configure the Part Application Editor component within a Node Editor. Each of the parameters for the Part Application Editor Properties are described below.

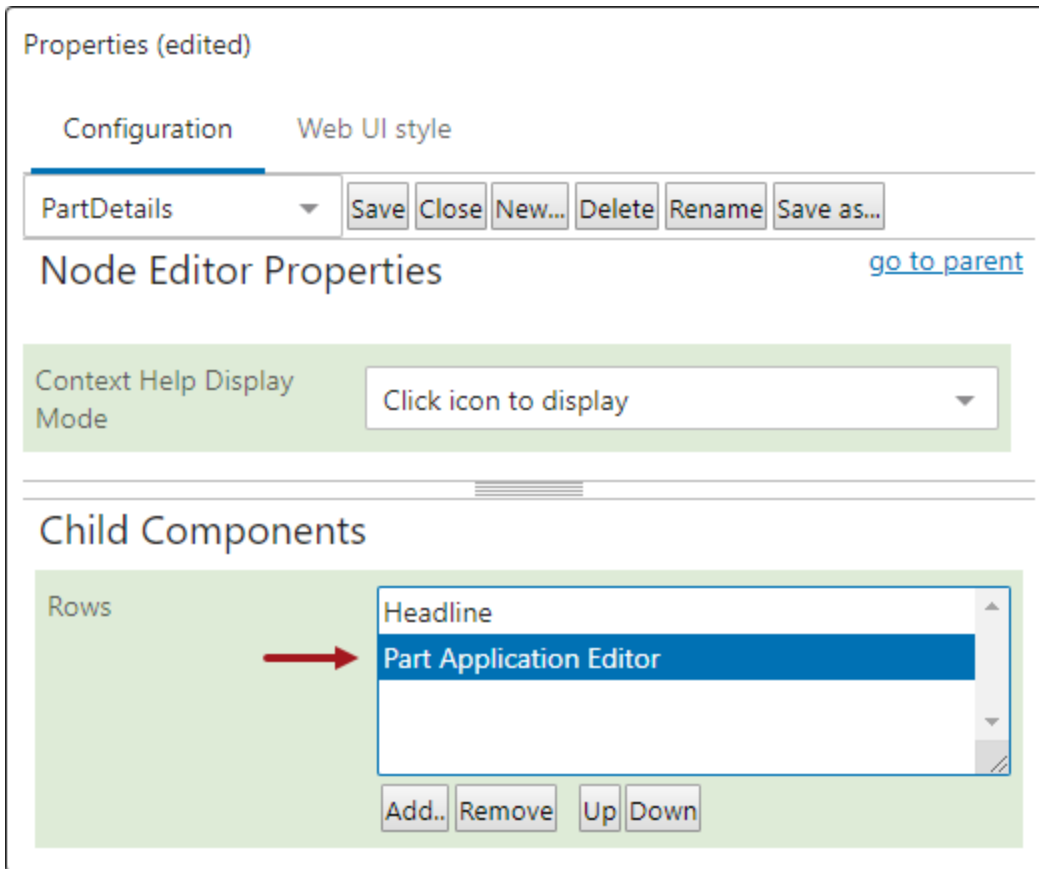
1. In the Web UI Designer, go to Node Editor Properties > Child Components > Rows parameter for the screen that needs to display application record information.



2. Click the **Add** button for the Child Components Rows parameter, and the Add Component dialog will display.



3. Find and select the Part Application Editor component, click the **Add** button, and the newly added component will display within the Rows parameter. In the screenshot below, an extra Headline component is also added to display a headline on a screen.



4. Double click the newly added component available in the Rows parameter, and the Part Application Editor Properties dialog will display (as shown below). Parameter options are explained below:

Properties (edited)

Configuration Web UI style

PartDetails ▾ Save Close New... Delete Rename Save as...

Part Application Editor Properties [go to parent](#)

Component Description Component to display application on Part

Height

Child Components

Node List ▾ [go to component](#)

Note: The Node List component defined within this parameter displays the application records in a table / grid format. The view can be further customized by configuring the data to display via different display modes and adding different action buttons that users can click while working with the application records information. The display modes can then be customized with a range of headers, allowing for different information about the listed objects to be displayed. The behavior of many of the various actions can also be further configured. For more information, see the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

Height: This optional parameter specifies the visual height of the Part Application Editor component in pixels. By default, this parameter is pre-populated with a value '400.' Users can enter a numeric value to alter height of the component.

5. Click the **Save** button, and then, click the **Close** button to close the designer.

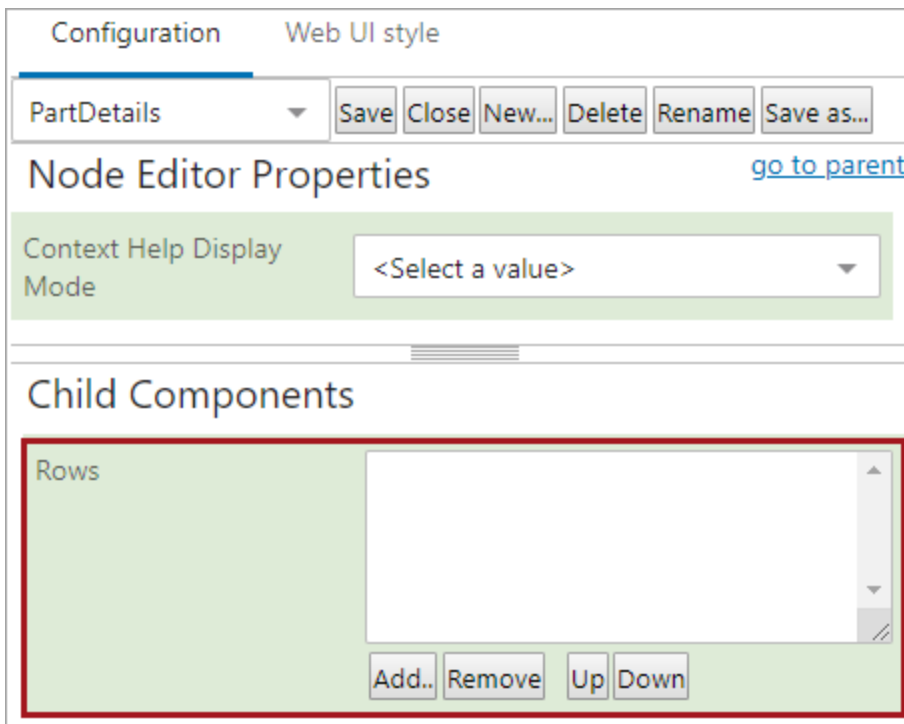
Configuring the Application Suggestions Component

The Application Suggestions component is a Node Editor component, often used along with the Part Application Editor component, to display potential applications listed as suggestions. The Application Suggestions component displays the suggestion list based on the configured reference types and plugins configured within the Application Suggestion Plugins when a part is selected in the Tree navigator. For information about how to use this component and its prerequisites, see the **Application Suggestion Solution** topic.

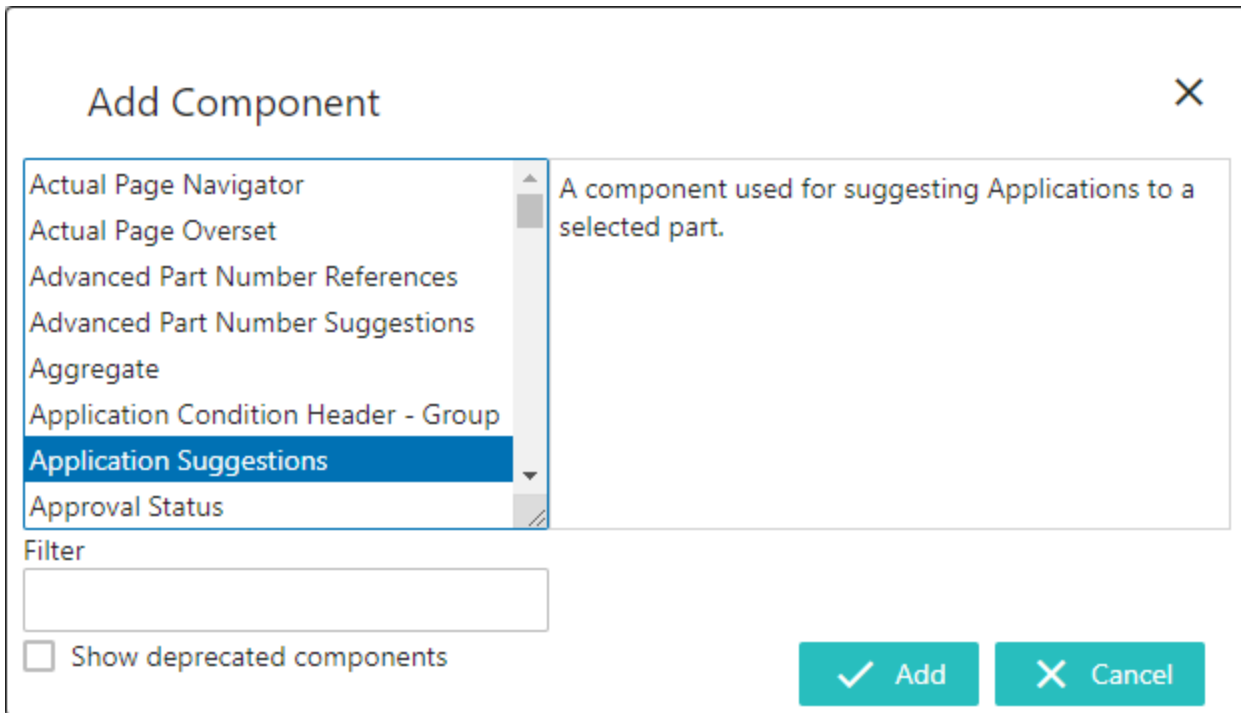
Configuration

Below are steps to configure the Application Suggestions component within a Node Editor. Each of the parameters for the Application Suggestions Properties are described below.

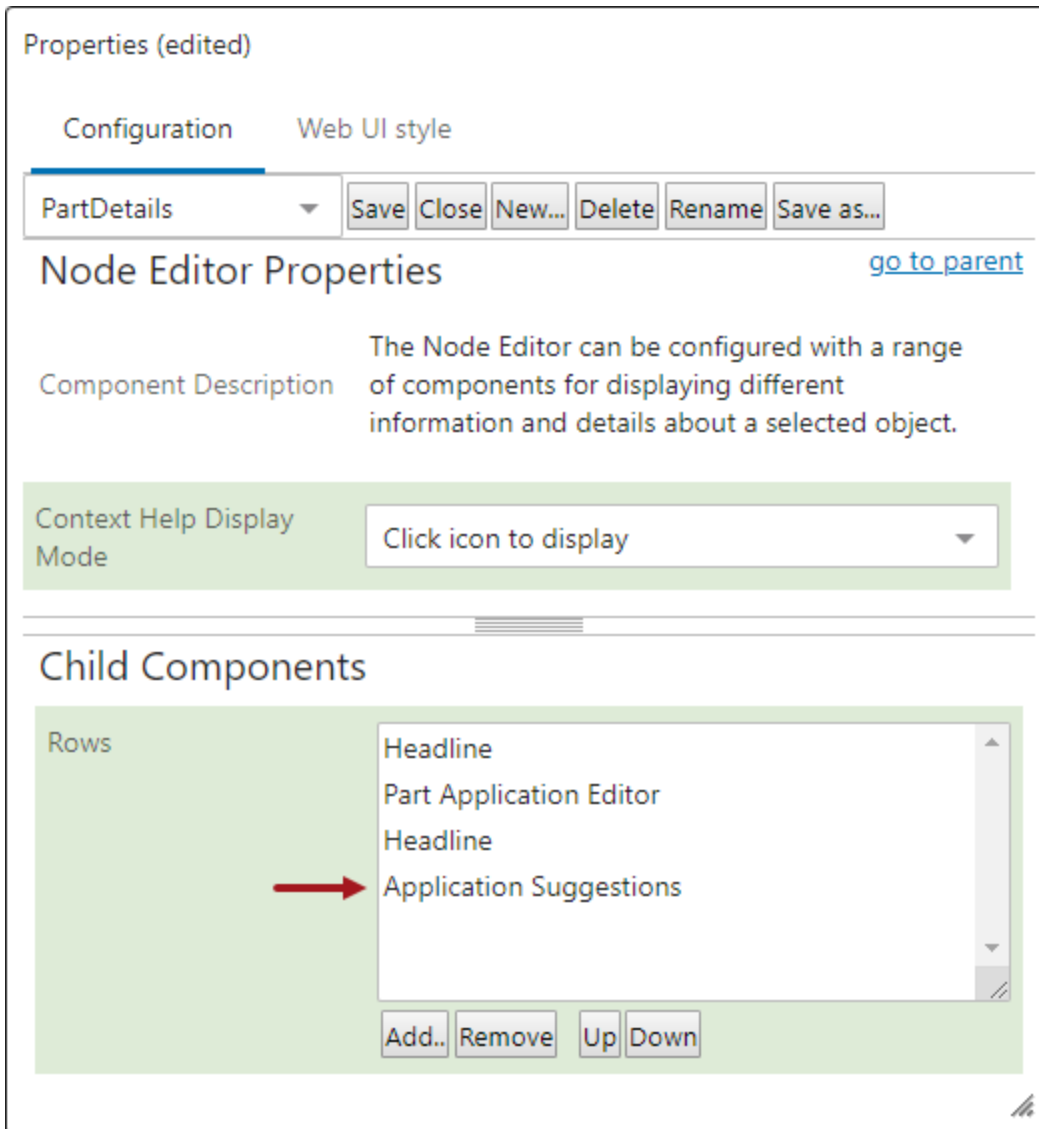
1. In the Web UI Designer, go to Node Editor Properties > Child Components > Rows parameter for the screen that needs to display application record suggestions.



2. Click the **Add** button, and the Add Component dialog will display.



3. Select the Application Suggestions component, click **Add**, and the newly added component will display within the Rows parameter. In the screenshot below, the Application Suggestions component is configured along with the Part Application Editor and Headline components.



4. Double click the newly added component available in the Rows parameter, and the Application Suggestions Properties dialog will display (as shown below). Parameter options are explained below:

Height: This optional parameter specifies the visual height of the Advanced Part Number Suggestions component in pixels. By default, this parameter is pre-populated with a value '400.' Users can enter a numeric value to alter height of the component.

Attribute Group: This optional parameter, when configured with an attribute group, sets up an additional unique condition for the suggested application records. An attribute group which is valid for the application records can be configured. For example, consider there are two application records with the same reference target to the Part Type and the Vehicle. Then the attribute values present within this attribute group determines the uniqueness of the application records. If both the application records has the same attribute values, then the system will suggest as one single potential application record. If any one of the attribute values differs for the application records, then the system would suggest as two different potential application records.

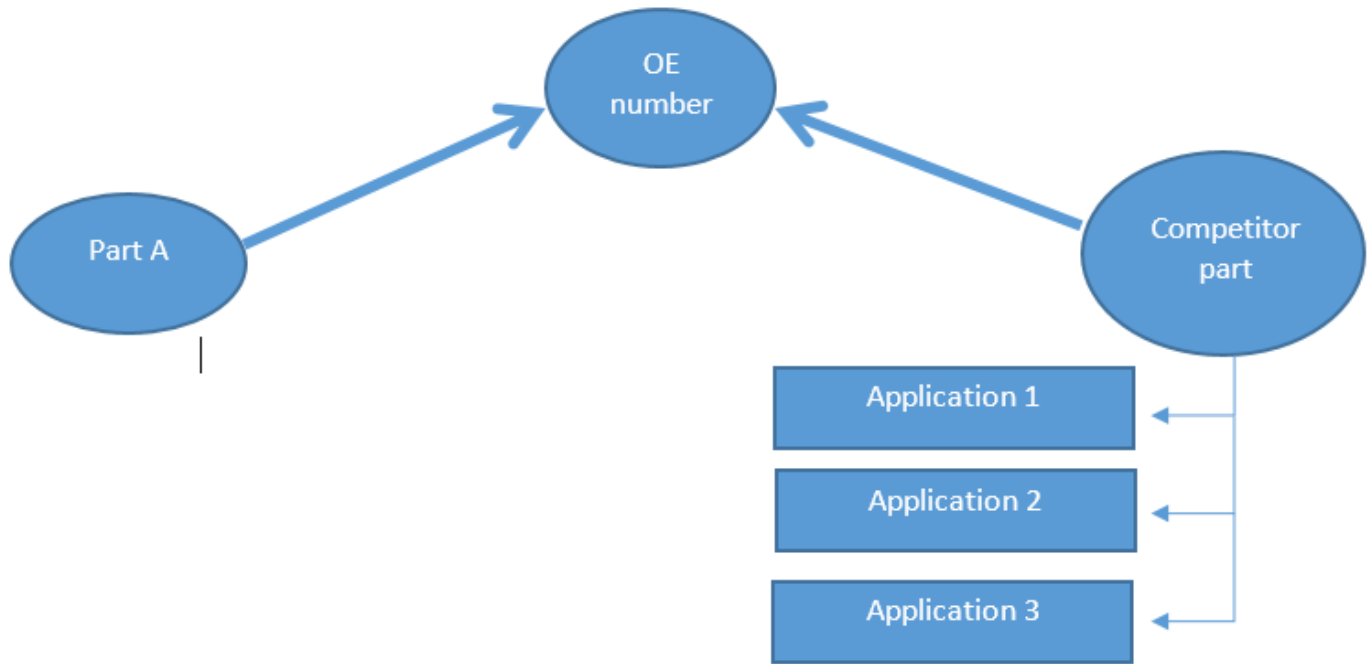
Note: The Node List component defined within this parameter displays the suggested application records information in a table / grid format. The view can be further customized by configuring the data to display via different display modes and adding different action buttons that users can click while working with the application records information. The display modes can then be customized with a range of headers, allowing for different information about the listed objects to be displayed. The behavior of many of the various actions can also be further configured. For more information, see the **Node List Component** topic within the **Web User Interfaces** section of **STEP Online Help**.

Application Suggestion Plugins: By default, this required parameter is blank. The user can add one or more plugins by clicking the **Add** button. It is mandatory to have at least one plugin so that when a part is selected, system would follow the configurations provided in the plugin and display the potential application records in accordance.

The screenshot shows a dialog box titled "Add Component" with a close button (X) in the top right corner. Inside the dialog, there is a list of components on the left and a description area on the right. The list contains two items: "Referenced Competitor / OE Number Part's Applications" and "Referenced Part's Application Suggestion Plugin". The description area contains the text "Select a component to see its description". Below the list is a "Filter" input field and a checkbox labeled "Show deprecated components". At the bottom right of the dialog are two buttons: "Add" (with a checkmark icon) and "Cancel" (with an X icon).

The Application Suggestion Plugins options are the following:

Referenced Competitor / OE Number Part's Applications: This plugin is used to suggest the Competitor's application records where the Competitor Number is referencing the OE Number of the current part. In the example chart below, Application 1, Application 2, and Application 3 are suggested as the potential applications to Part A. Additional configurations are required for this plugin.



When a user adds this plugin, an 'Add component' dialog is displayed prompting users to configure the associating reference types linked between the part and the OE Number, and the reference types linked between the Competitor and OE Number.

Referenced Competitor / OE Number Part's Applications Properties

Component Description Application Suggestion Plugin based on the selected part's reference to a Competitor / OE Number part that references the suggested applications.

*Competitor Reference Type

*OE Number Reference Type

- Competitor Reference Type:** By default, this required parameter is blank. However, it is mandatory to add a Product Reference Type that establish a reference link between the Competitor to OE Number so that when a part is selected, the reference types listed within this parameter are followed and display the application records of the related competitor number on the

targets. In order to have this parameter configured, user must establish a reference link between the Competitor to OE Number.

- For AutoCare solution this may be adding reference type which bears a relation from id 'AC_PIESInterchangeItem' to 'AC_PIESInterchangeItem'
- For NAPA solution this may be adding reference type which bears a relation from id 'NAPA_InterchangeProduct' to 'NAPA_InterchangeProduct'
- For TecDoc solution this may be adding reference type which bears a relation from id 'TD_DS_SupplierCompetitorNumber' to 'TD_DS_OENumber'
 - **OE Number Reference Type:** By default, this required parameter is blank. However, it is mandatory to add one Product Reference Type so that when a part is selected, the reference types listed within this parameter are followed and fetch the Competitor number of the target OE number.
- For AutoCare solution this may be adding reference type with id 'AC_PIESInterchange'
- For NAPA solution this may be adding reference type with id 'NAPA_ProductToInterchangeProduct'
- For TecDoc solution this may be adding reference type with id 'TD_SupplierArticleToOENumber'

Referenced Part's Application Suggestion Plugin: This plugin is used to suggest an other part's application records. Additional configuration is required for this plugin. When a user adds this plugin, an 'Add component' dialog is displayed prompting users to configure the associating reference type linked between the two parts.

Reference Type: By default, this required parameter is blank. However, it is mandatory to add at least one Product Reference Type so that when a part is selected, the reference types listed within this parameter are followed and display the application records of the target part.

- For AutoCare solution this may be adding reference type with id 'AC_PartRelation'
 - For NAPA solution this may be adding reference type with id 'NAPA_PartRelation'
 - For TecDoc solution this may be adding reference type with id 'TD_PartRelation'
5. Under the Child Components section of the Node List properties, in the dropdown select 'Node List' > click on 'go to component.' The Node List Properties dialog with two pre-configured action buttons will display as shown below.

Add Applications to Part Action: This required action button adds the selected application record to the selected part from the Application Suggestions list. This is a ready-to-use button, and the user can double click on the 'Add Applications to Part Action' to further configure the Icon style, Title, and Label of the action button if necessary.

Remove Application Suggestions Action: This action button removes the selected application record from the Part Number Suggestions list. This is a ready-to-use button, and the user can double click on the 'Remove Application Suggestions Action' to further configure the Icon style, Title and Label of the action button if necessary.

6. Click the **Save** button, and then click the **Close** button to close the designer.

Data Onboarding Solution

Within the automotive industry, it is often difficult to organize supplier and/or competitor data across different automotive industry standards and data models. The Data Onboarding solution helps to provide tools for suppliers and retailers to be able to map data from one industry standard to another, or to and from your own data model. This is done by setting up mapping configurations to onboard or outboard objects from one standard to another standard as defined by the user.

The mapping functionality can copy and/or transform data from one STEP object to another STEP object. The data can be as simple as copying attribute values, or as complex as retrieving data from the source object to create an entirely new product or classification hierarchy.

The actual data mapping is executed via Business Rules and/or Background processes inside a workflow or upon approvals. After initial configuration, Web UI users can view, edit, and create Mapper Configuration setup entities and Mapper plugins to better handle the automation of data onboarding.

The Data Onboarding solution is used within an Onboarding Mappings Detail Screen.

This section addresses

- Prerequisites
- Configuring Data Onboarding solution
- Configuring Web UI for Data Onboarding solution
- Executing Mapper Configuration setup entity
- Mapping plugins
- Mapping plugins use case overview

Prerequisites

Prior to implementing the Data Onboarding Automation solution, the following items included in the list below should be carefully considered when planning the data onboarding solution.

- Source Object Type
- Target Object Type
- Reference Type

Configuring Data Onboarding Solution

The following topics provide the configuration processes necessary to allow users to be able to set up the Data Onboarding solution. The initial part of the configuration is done in the workbench and the later part is configured in the Web UI.

- Data Onboarding Solution Initial Setup
- Configuring the Execute mapper configuration Business Action in Workbench
- Configuring Web UI for Data Onboarding Solution

This solution is available only in Product MDM for Automotive 9.0-MP3 and later versions.

Prerequisites

Apply the automotive recipe to the STEP instance to display the 'Automotive - Mapper Extension' component model within Workbench > System Setup > Component Models (as shown below).

The screenshot displays the SAP Workbench interface. On the left, the 'System Setup' tree is expanded to 'Component Models', where 'Automotive - Mapper Extension' is selected. The main area shows the configuration for this component model.

Component Model Configuration		
Name	Value	Description
Mapper Extension Point		Objects that the mapper should use the extension to find the paths from Assembly to Parts
Edit		

Data Onboarding Solution Initial Setup

The Mapper Configuration is the STEP setup entity object, which holds the definition of what the mapper should do. After the Easy Setup action is run, a one-time initial setup is required before any Mapper Configurations can be created. A setup group must be created to hold the Mapper Configurations. 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 'Mappings' is created where users can add new Mapper Configurations. Further configuration of the Mapper Configurations can be only done in the Web UI. For information on configuring Mapper configurations in the Web UI, see the **Configuring Web UI for Data Onboarding Solution** topic.

Important: Running the Easy Setup action creates the **Mapper Configuration** object type under Basic Object Types hierarchy within System Setup tab as displayed in the screenshot below.

The screenshot displays the 'System Setup' interface. On the left, a tree view shows the hierarchy: System Setup > Object Types & Structures > Basic Object Types > Mapper Configuration. On the right, the 'Mapper Configuration - Object Type' configuration window is open, showing a table of attributes and their values.

Object Type		References	Log
Description			
Name	>	>	Value >
ID	>		MapperConfiguration
Name	>		Mapper Configuration
Last edited by	>		2018-12-14 08:01:34 by DBA
Name Pattern	>		
ID Pattern	>		
Icon	>		
Dimension Dependencies	>		
Valid Attributes			
ID	>	Name	>
Add Attribute			

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., MapperConfigurations), enter a Name in the Name parameter (e.g., Mapper Configurations), then click the **Create** button. This creates a setup group object type that will be used for the root node of the Mapper Configurations object to store all mapper configurations below.

The sample below illustrates a hierarchy of setup group object types. A Setup Group named 'Mapper Configurations' has been created.

The screenshot displays the 'System Setup' interface. On the left, a tree view shows the hierarchy: System Setup > Object Types & Structures > Basic Object Types > Mapper Configurations. The right pane shows the details for the 'Mapper Configurations' object type, including a 'Description' table and 'Valid Attributes'.

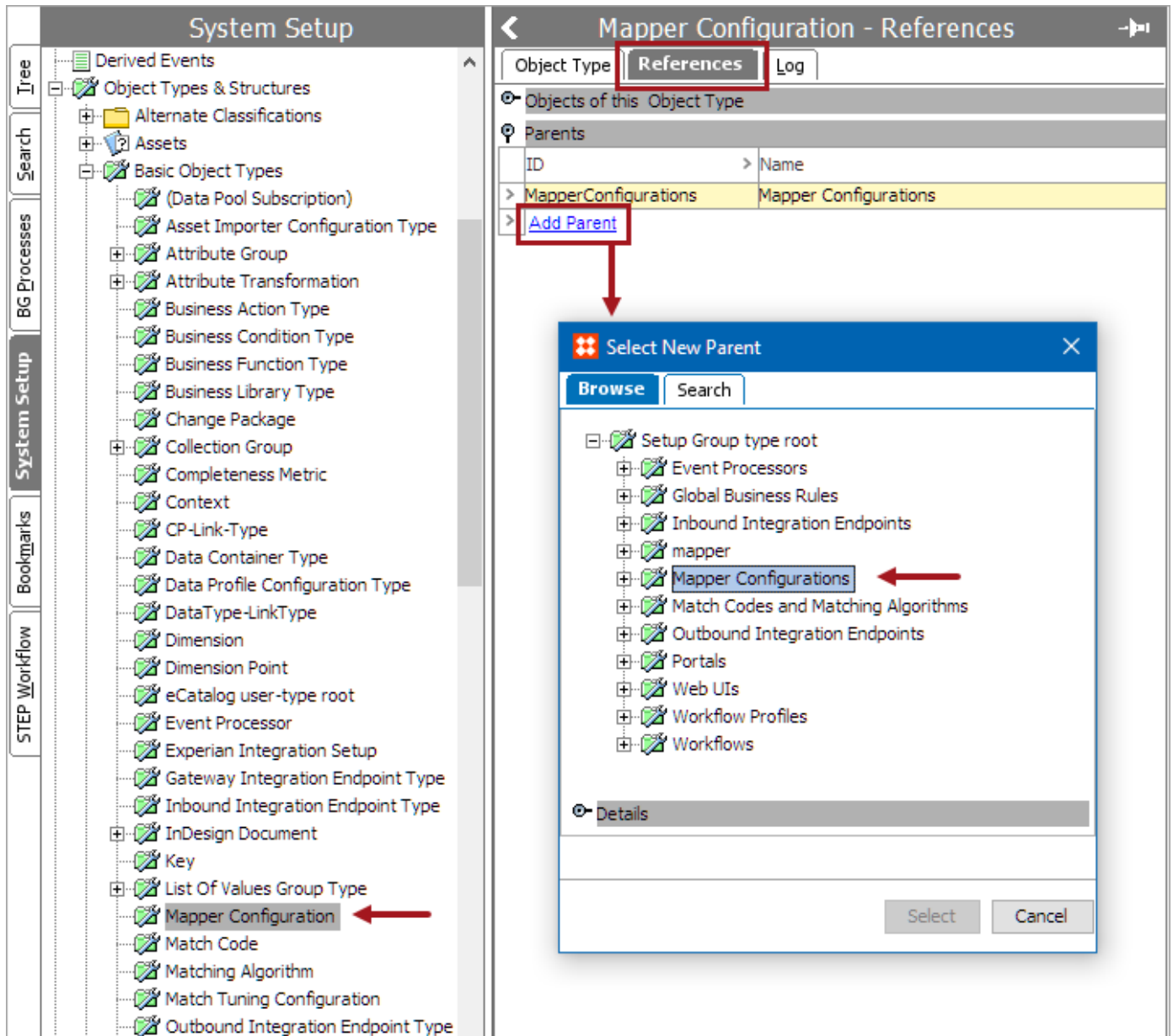
Description	
Name	Value
ID	MapperConfigurations
Name	Mapper Configurations
Last edited by	2018-12-14 06:39:58 by USERB
Name Pattern	
ID Pattern	
Icon	
Dimension Dependencies	

Valid Attributes	
ID	Name
Add Attribute	

Once the setup group is made, the 'Mapper Configurations' basic object type must be linked to this newly created node as a child. Later, when new Mapper Configurations are created and added to the system, they will use this object type.

3. Navigate to **Object Types & Structures > Basic Object Types > Mapper Configurations**, and click the **References** tab.

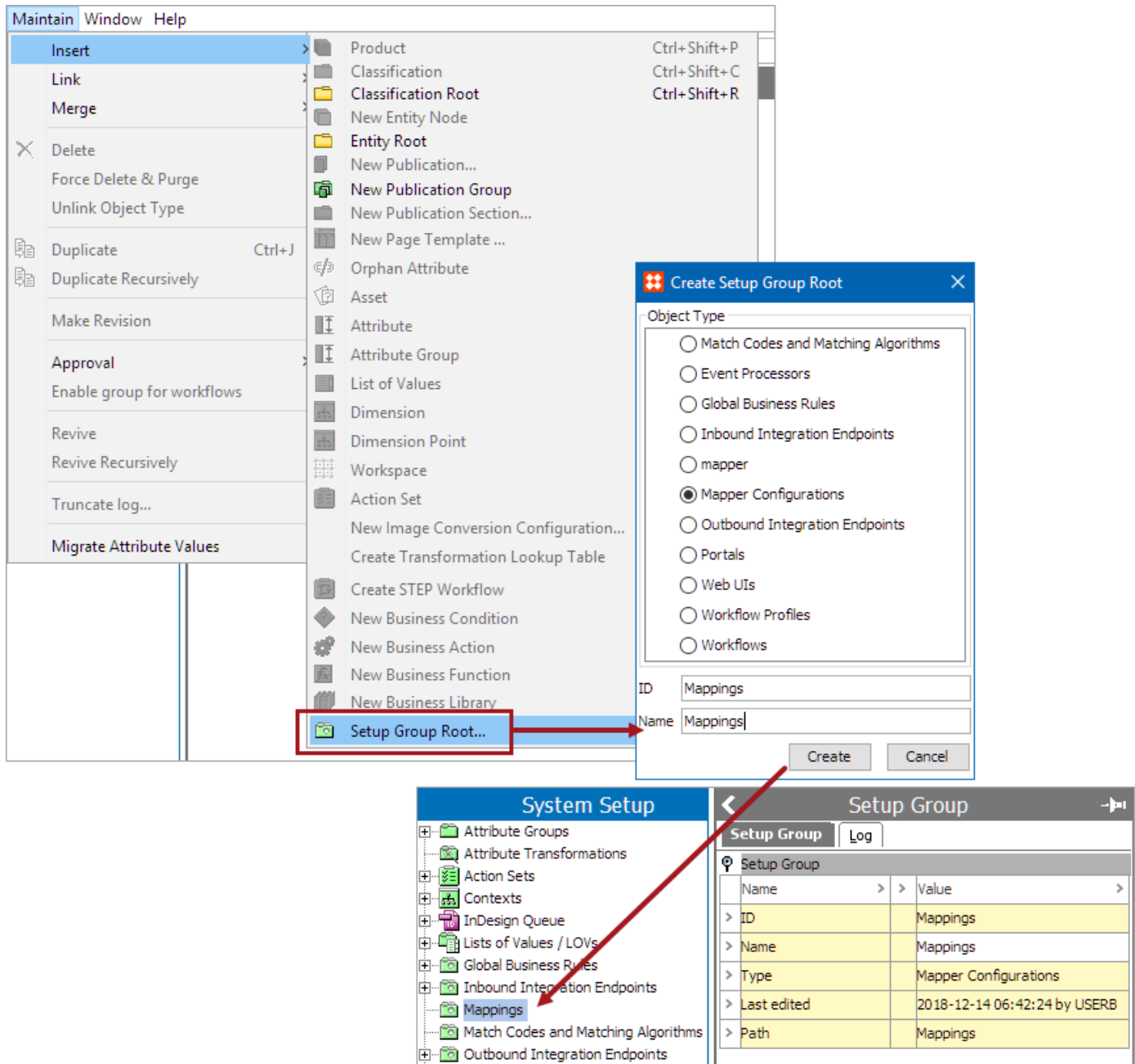
Under the **Parents** flipper, click **Add Parent**, and in the node selector dialog, specify the **Mapper Configurations** node that was created in the above step.



The next step is to create a new setup group root, where all Mapper Configurations 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 'Mapper Configurations' object type.
3. Specify the **ID** and **Name** for the setup group root object. In this setup, ID and Name value of 'Mappings' is entered.
4. Click **Create**.

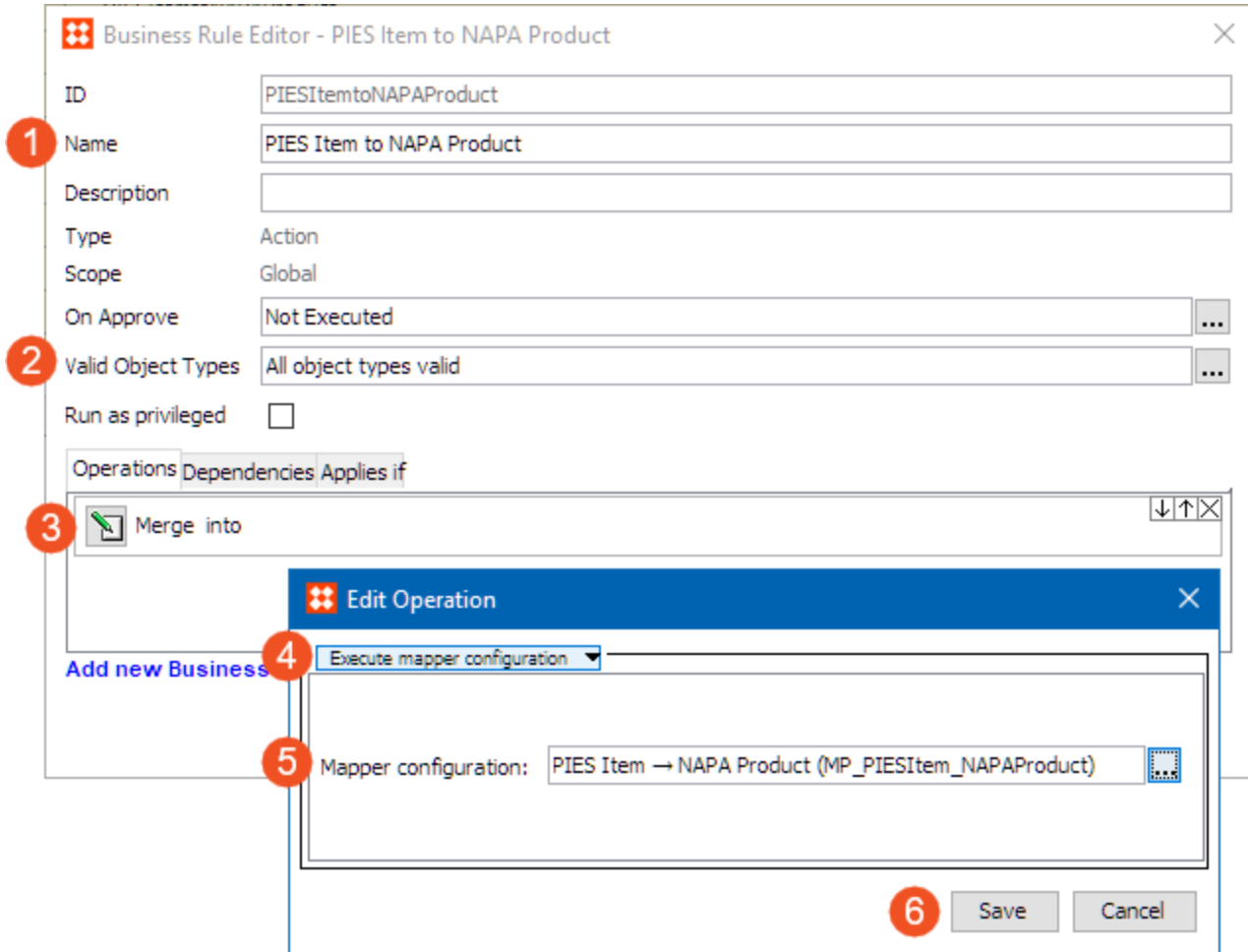


The setup group called 'Mappings' of the object type 'Mapper Configurations' is created. From this newly created node, users can right-click and add new Mapper Configurations.

The Mapper Configurations under the setup group 'Mappings' 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 the Execute Mapper Configuration Business Action

The 'Execute mapper configuration' business operation is found within the STEP Workbench Business Rule Editor under the Mapping menu and requires population of a mapper configuration. This section addresses the steps necessary for configuring the 'Execute mapper configuration' business action within the workbench.



1. Create the business action with a name that accurately describes to the user what this action will do. For this example, the business action name is 'PIES Item to NAPA Product.'
2. Edit the new business rule, click the ellipsis button (...) next to the Valid Object Types parameter, and select the 'All Object Types' for this business action.
3. On the Operations tab of the Business Rule Editor, click the **Add new Business Action** link, and click the edit button to open the Edit Operation dialog.
4. Use the dropdown menu within the Edit Operation dialog to select **Mapping > Execute mapper configuration** operation, and the parameter 'Mapper configuration' will display (as shown above).
5. Within the parameter, click the ellipsis button (...) and select the desired Mapper Configuration from the list of available Mapper Configurations.

Important: A different Business Action will need to be created for each different mapper configuration so that the Business Action can be used wherever the mapping needs to be executed.

6. Click the **Save** button.

Configuring Web UI for Data Onboarding Solution

This topic covers how to configure the Web UI for Onboarding / Outboarding data in Web UI.

Prerequisites

Designers need to create and configure an Onboarding Mappings Details Screen, to configure left stack panel as an access to Mapper Configurations, and to configure different Mapper Configurations to meet user requirements. Detailed configuration and usage instructions are provided in the following topics of this documentation:

- Onboarding Mappings Details Screen
- Configuring Mapper Configuration Setup Entity
- Using an Onboarding Mappings Details Screen

Onboarding Mappings Details Screen

The Onboarding Mappings Details screen is used to set up Mapper Configurations in Web UI. User can further configure the Mapper Configurations within this screen to configure Mapping plugins. For more information, see **Using an Onboarding Mappings Details Screen** topic.

Mapper Configurations - Stack Panel Component

The left Stack Panel should be configured as an entry point to display the list of available Mapper Configurations wherein the display and further configuration of these available Mapper Configurations is done using Onboarding Mappings Details screen.

Multiple Mapper Configurations can be created under this stack panel component to meet different data onboarding needs. As an example, there may be multiple requirements to onboard or offboard different objects, attributes, etc. In such cases, the creation of multiple Mapper Configurations is required to meet different needs. All the available Mapper Configurations are listed in the left Stack Panel component. For more information, see **Onboarding Mappings Details Screen** topic.

Onboarding Mappings Details Screen

The Onboarding Mappings Details screen allows users to quickly display mapping configurations. The **Onboarding mappings navigator** component functions as a Stack Panel component. For more information, see the **Tree Navigator Component** in the **Web User Interfaces / Web UI Getting Started** documentation.

Configuration Prerequisites

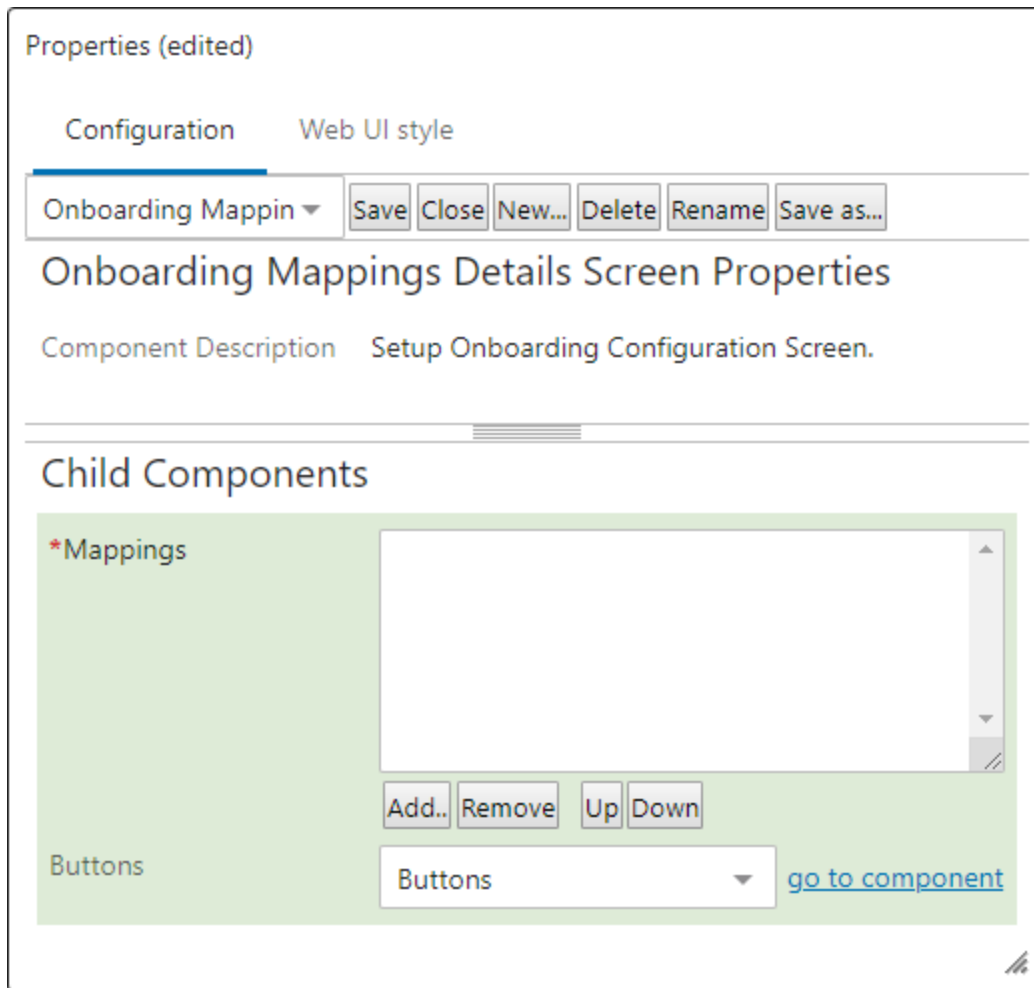
It is expected that anyone configuring the Onboarding Mappings Details Screen component is familiar with the Web UI Designer as basic concepts for working with the designer are not covered in this section. In addition, the user must have appropriate privileges to access the designer. Additional information can be found in the **Designer Access** section of the **Web User Interfaces / Web UI Getting Started** documentation.

Configuration Process

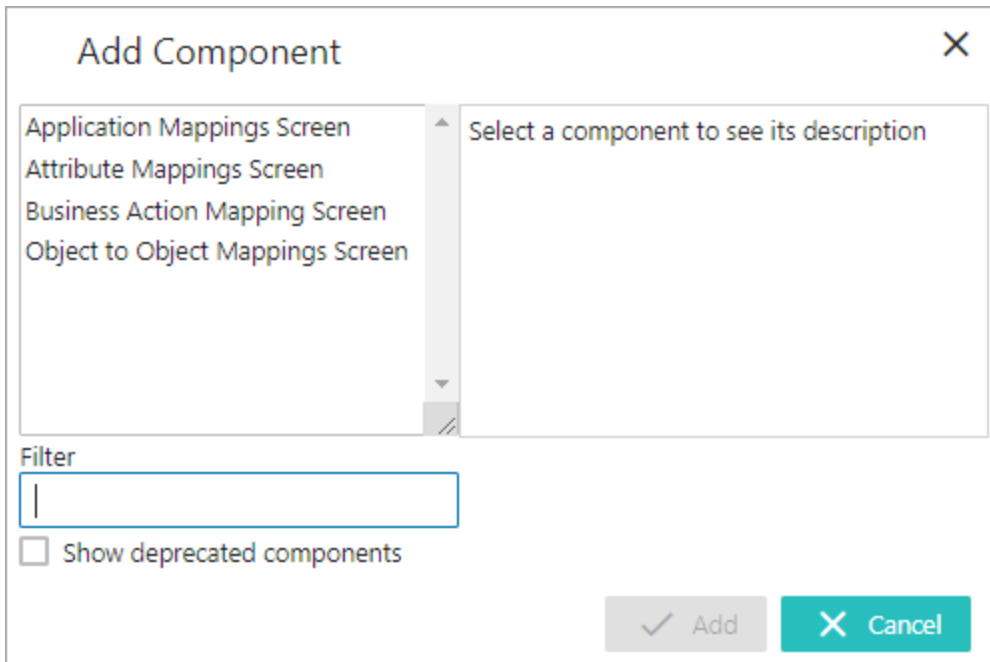
In the designer, create a new screen by selecting 'New.' Select the 'Onboarding Mappings Details Screen,' and create a name for the screen in the 'Screen ID' field. In the example below, the Screen ID is 'Onboarding Mappings Details Screen'. Click 'Add.'

The screenshot shows the 'Properties' panel for the 'Home Page Properties' component. The 'Configuration' tab is active, and the 'New...' button is highlighted with a red box. Below the properties panel, the 'Add Screen' dialog is open. The 'Screen ID' field contains 'Onboarding Mappings Details Screen'. A list of screen templates is displayed, with 'Onboarding Mappings Details Screen' highlighted by a red box. The 'Add' button is highlighted in teal.

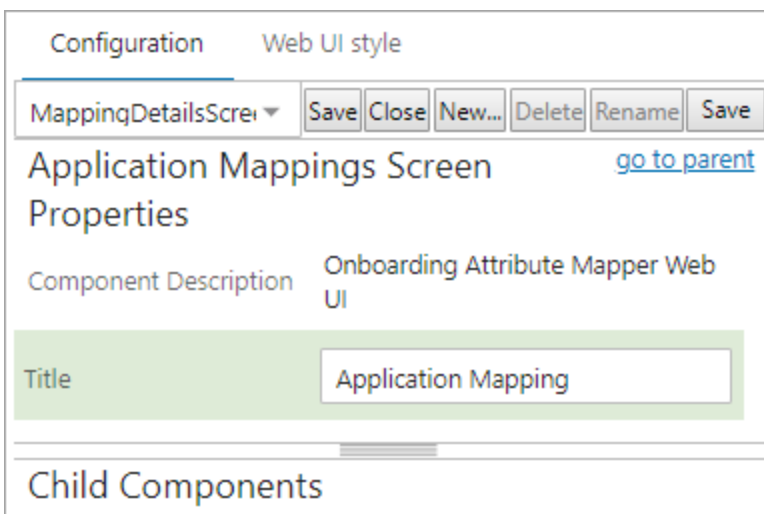
Once this screen is created, the Onboarding Mappings Details Screen Properties designer screen displays, which contains the following parameters for configuration:



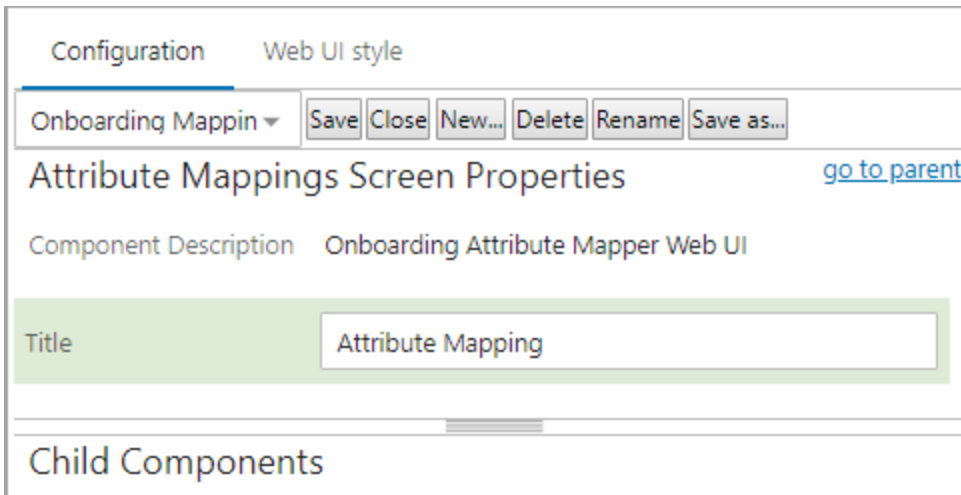
Mappings: By default, this required parameter is blank. Click on the 'Add' button to add the mapping plugins that admin users should be able to view and select when configuring the mappings. The following are the available mapping plugins:



Application Mappings Screen: This mapping plugin, when configured in the Mapper Configuration, will take data from the applications on the Source object to create new applications in the Target object, as well as establish references for the Target applications to Target vehicles and Target part types. Double-click the added 'Application Mappings Screen' in the Mappings field to rename the plugin name if necessary by changing the value in the 'Title' parameter.



Attribute Mappings Screen: This plugin, when configured in the Mapper Configuration, will copy attribute values from the Source object to populate attributes in the Target object. In other words, If the Source object has an attribute value of 'X,' then this value can be copied into a valid attribute of the Target object (with transformations, if required). Double-click the added 'Attribute Mappings Screen' in the Mappings field to rename the plugin name if necessary by changing the value in the Title parameter.



Object to Object Mappings Screen: This plugin, when configured in the Mapper Configuration, copies any data from the Source object by using the STEP path to either create child objects on the Target object, or store the data as attribute values, or as references on the Target objects. Double-click the added 'Object to Object Mappings Screen' in the Mappings field to rename the plugin name if necessary by changing the value in the Title parameter.

Buttons: A number of action buttons can be added to the 'Onboarding Mappings Details Screen' to be used with the Mapper Configuration list. These action buttons are added and configured within the 'Actions' field on the Buttons parameter. The 'Save Action' is required to be configured. Other actions should be added only when necessary.

Important: The **Delete**, **Duplicate**, and **Reset** buttons configured within the Buttons field will delete / duplicate / reset the Mapper Configuration, NOT the individual mapping plugin rows within the configuration. If the user does not want to be able to delete the Mapper Configuration in Web UI, then do not configure the **Delete** Action.

If the Mapper Configuration is deleted in Web UI, then it will get moved to the Recycle Bin in System Setup. Users can revive the Mapper Configuration via the 'Maintain' dropdown > Revive.

Implementing the Onboarding Mappings Details Screen

Once the 'Onboarding Mappings Details Screen' is created, implementation of the screen requires the following setup:

- A Stack Panel needs to be configured to access the Mapper Configurations
- A Mapper Configuration object type must be configured to display the 'Onboarding Mappings Details Screen' when a Mapper Configuration is selected

Configuring the Stack Panel

1. From the **Main Properties** screen, add a 'Stack Panel' to the 'Left' section of the child components. Once added, double-click the Stack Panel component.

Properties

Configuration Web UI style

---[MAIN]---

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

Main Properties

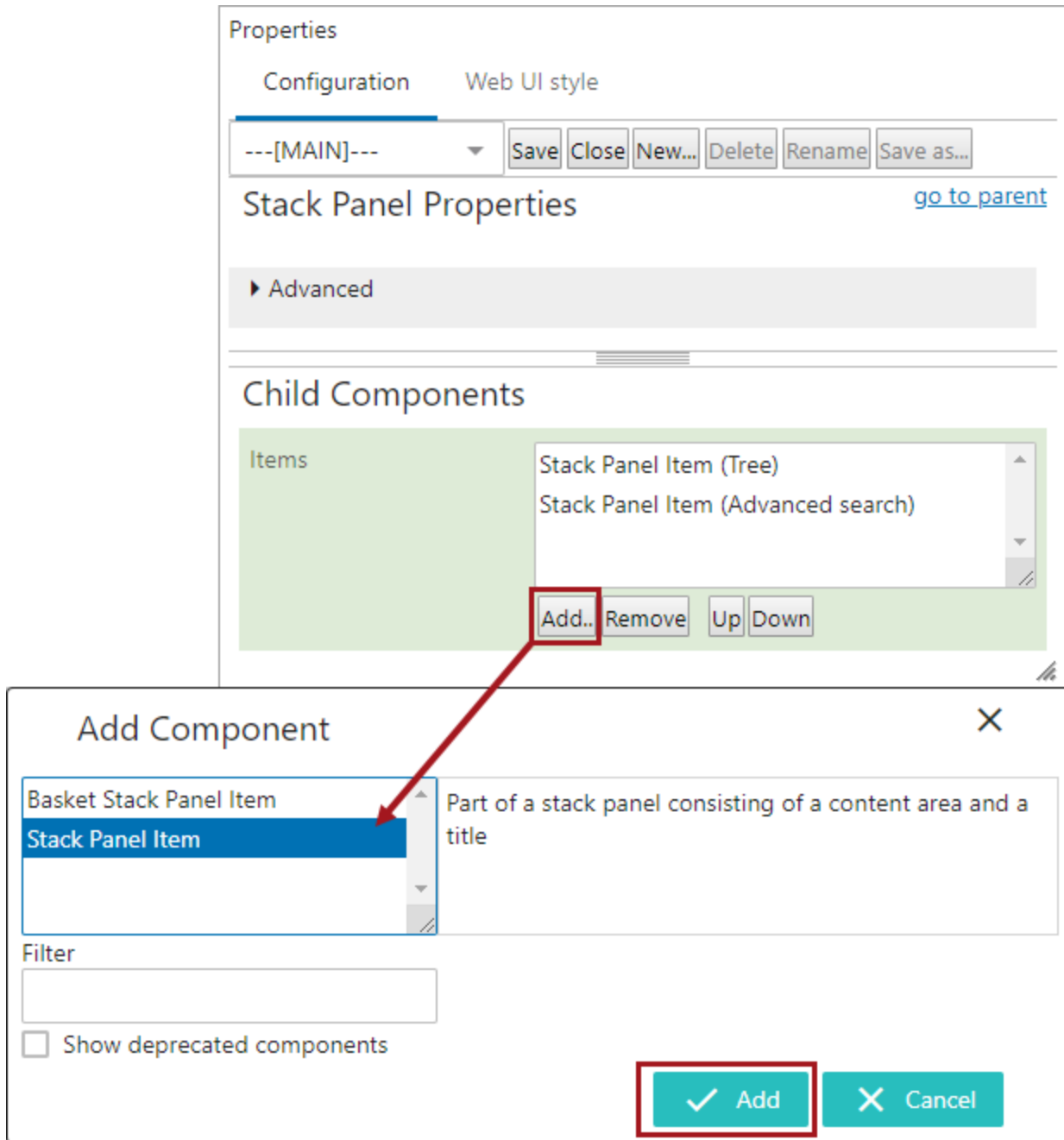
Component Description

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.

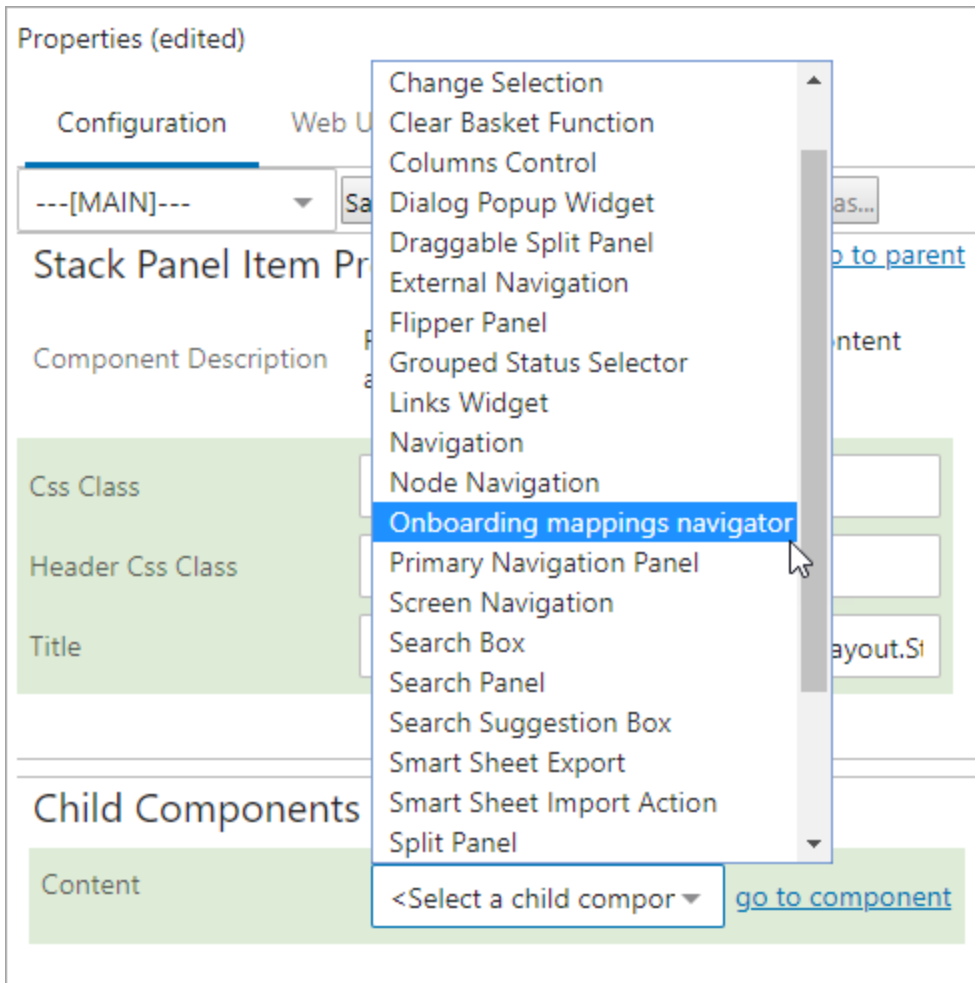
Child Components

Top	Add.. Remove Up Down
Left	Stack Panel
Corner Bar	Corner Bar go to component

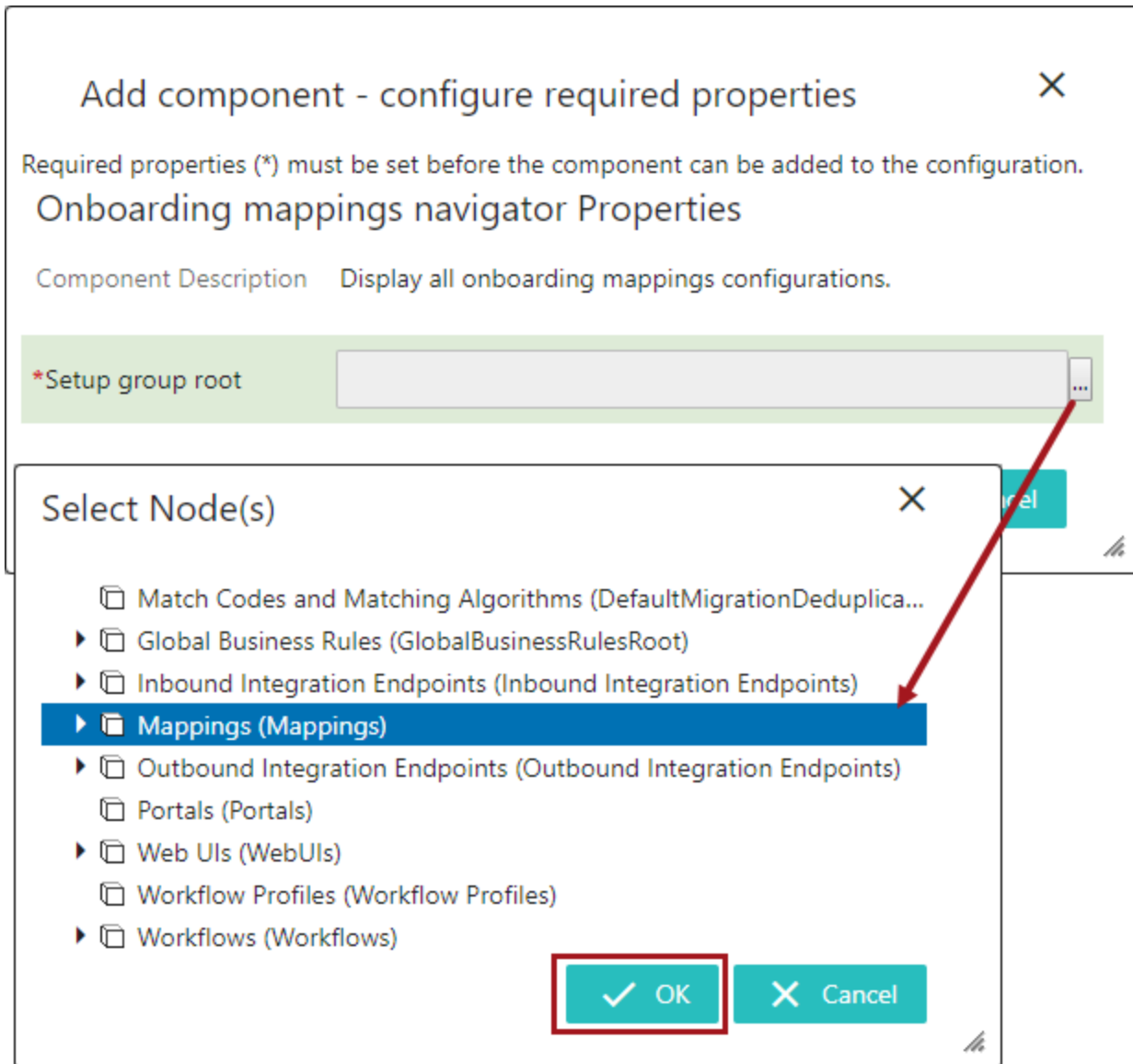
- From the Stack Panel Properties, select 'Add' in the Items field of the Child Components section. Select 'Stack Panel Item' from this dialog.



3. Double-click the Stack Panel Item component to access the configuration. In the Stack Panel Item Properties, select the 'Onboarding mappings navigator' in the Content field dropdown of the Child Components section.



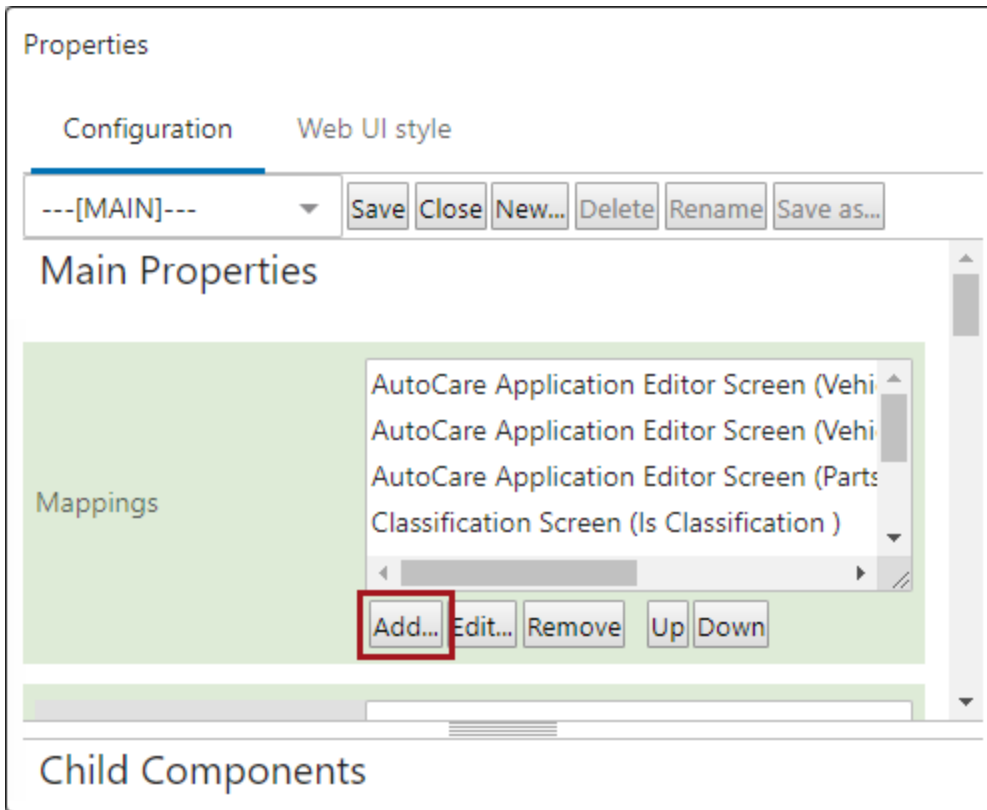
4. Click on the **go to component** link and, in the 'Onboarding mappings navigator Properties,' click the ellipsis button (...) next to the 'Setup group root' parameter. Then select the **Mappings** node to define the setup group root for mapping configurations. Click 'OK' to add.



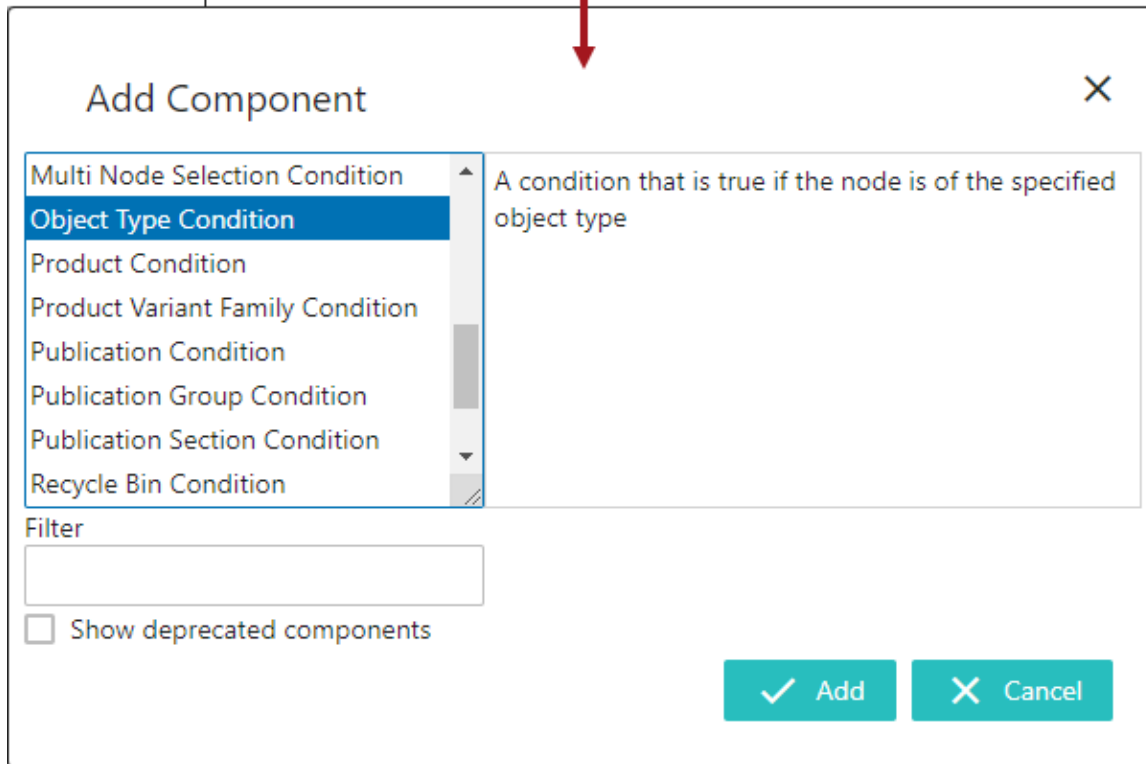
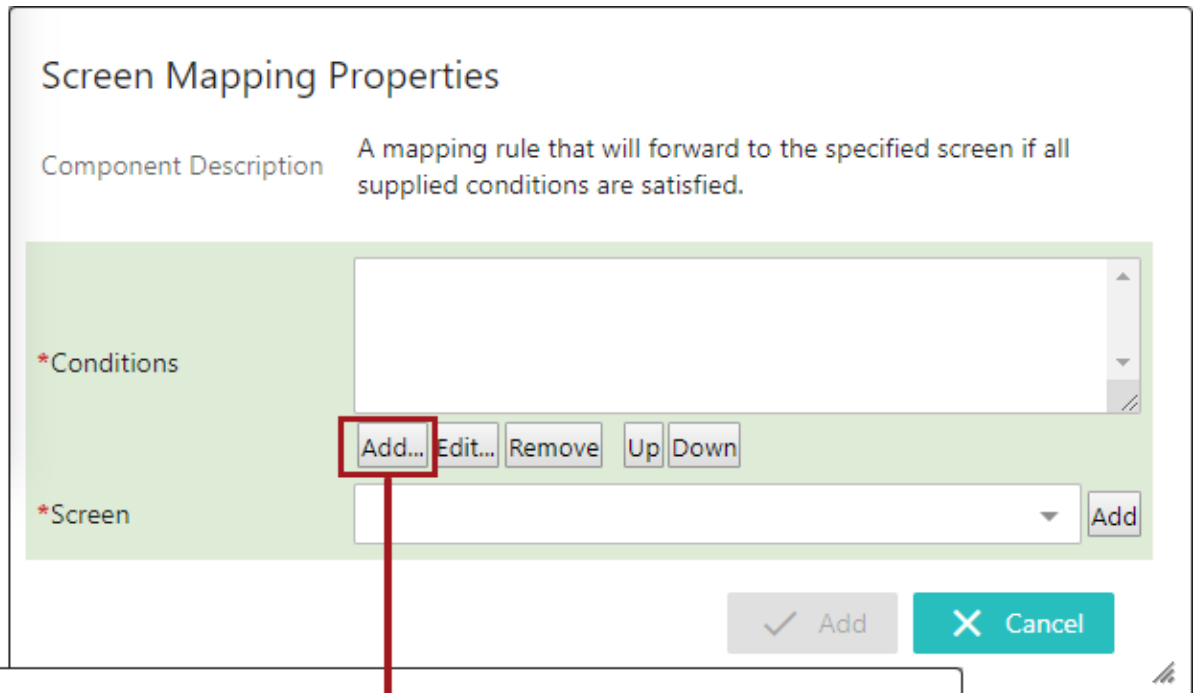
When finished, click **Save** in the designer to save the current settings. Do not close the designer unless you have finished configuring all the other parameters (outlined in the rest of this topic).

Configuring the Mapper Configurations to Display Onboarding Mappings Details 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 Mapper Configuration (MapperConfiguration) object type to define that this object type should use the Onboarding Mappings Details Screen. Click **OK** to add the object type. Then click **Add** to add the Object Type Condition.

Add component - configure required properties [Close]

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:

[Add] [Cancel]

- In the Screen Mapping Properties, select the previously created Onboarding Mappings Detail Screen using the dropdown selector next to the 'Screen' parameter.

Add component - configure required properties [Close]

Screen Mapping Properties

*Conditions:

[Add...] [Edit...] [Remove] [Up] [Down]

*Screen: [Add]

[Add] [Cancel]

- Click **Add**. The new mapping is added in the 'Mappings' field (as shown below). Click **Save** in the designer to save the current settings.

Properties (edited)

Configuration Web UI style

---[MAIN]--- Save Close New... Delete Rename Save as...

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

- Onboarding Mappings Details Screen (ObjectType = MapperConfiguration)
- AutoCare Application Editor Screen (Vehicles) (ObjectType = AC_BaseVehicle)
- AutoCare Application Editor Screen (Vehicles) (ObjectType = NAPA_Year)
- AutoCare Application Editor Screen (Parts) (ObjectType = AC_PIESItem)
- Classification Screen (Is Classification)
- BackgroundProcessDetails (Is BackgroundProcess)
- assetdetails (Is Asset)

Add... Edit... Remove Up Down

Child Components

Configuring Mapper Configuration Setup Entity

The Mapper Configuration is a STEP setup object. It is stored as a STEP setup entity with object type 'MapperConfiguration.' Each Mapper Configuration holds the definition of what the mapper should do.

Important: Creation of the Mapper Configuration setup entity can be done in both workbench and Web UI, but further configurations can only be done in Web UI and not in Workbench.

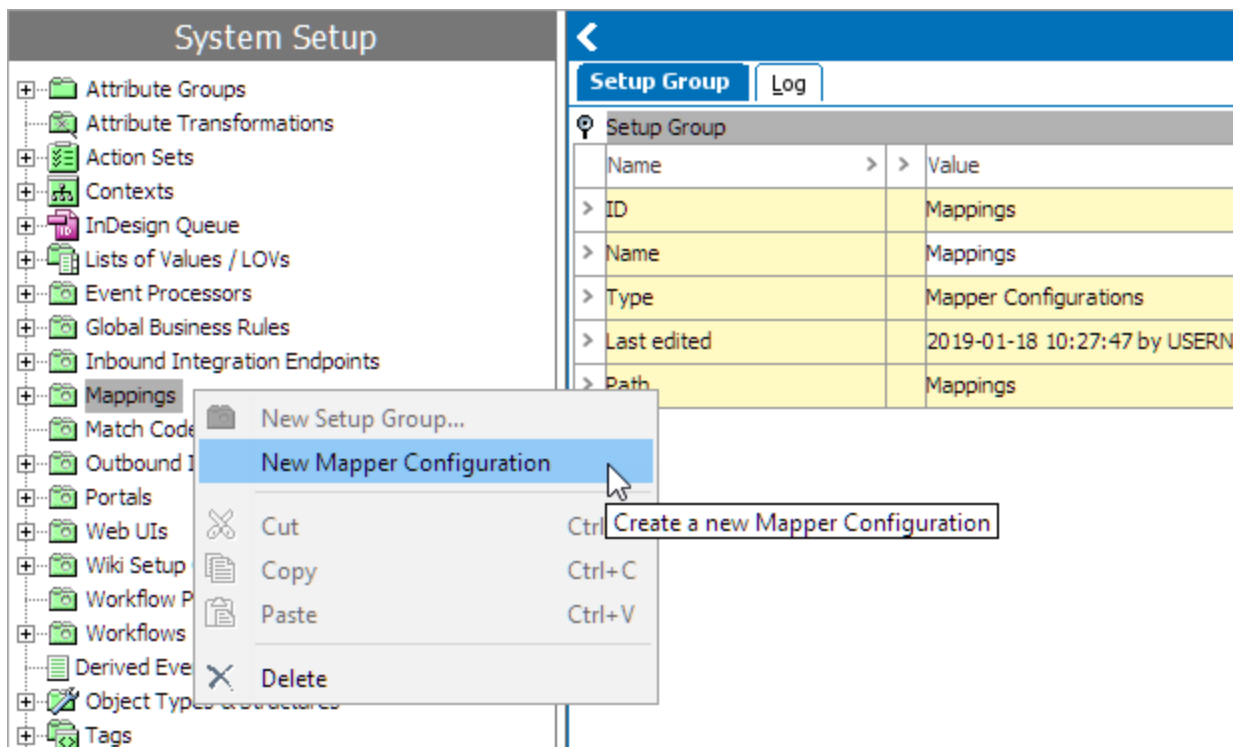
Prerequisites

A setup group must be created to hold the Mapper Configurations. This is a one-time setup in the system. For more information about creating a setup group, see the **Data Onboarding Automation Solution Initial Setup** topic within this section.

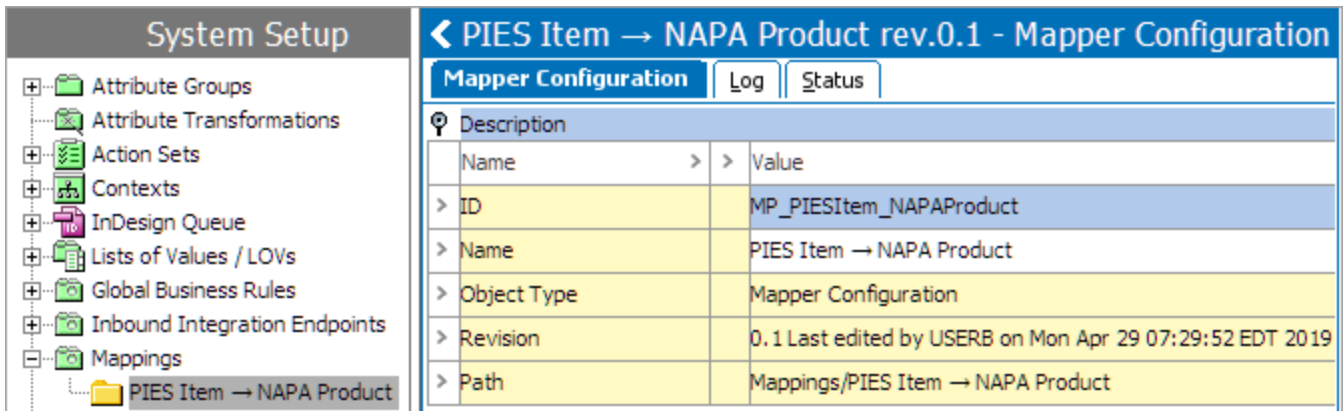
Since basic concepts for working with the designer are not covered in this section, a user configuring the Mapper Configuration setup entity within a Web UI is expected to be familiar with the Web UI Designer. In addition, the user must have appropriate privileges to access the designer. For more information, see the **Designer Access** topic within the **Web UI Getting Started** section of **STEP Online Help**.

Creating Mapper Configuration Setup Entity in Workbench

1. To create the Mapper Configuration setup entity, select the Mappings setup group root node, right-click and select New Mapper Configuration.

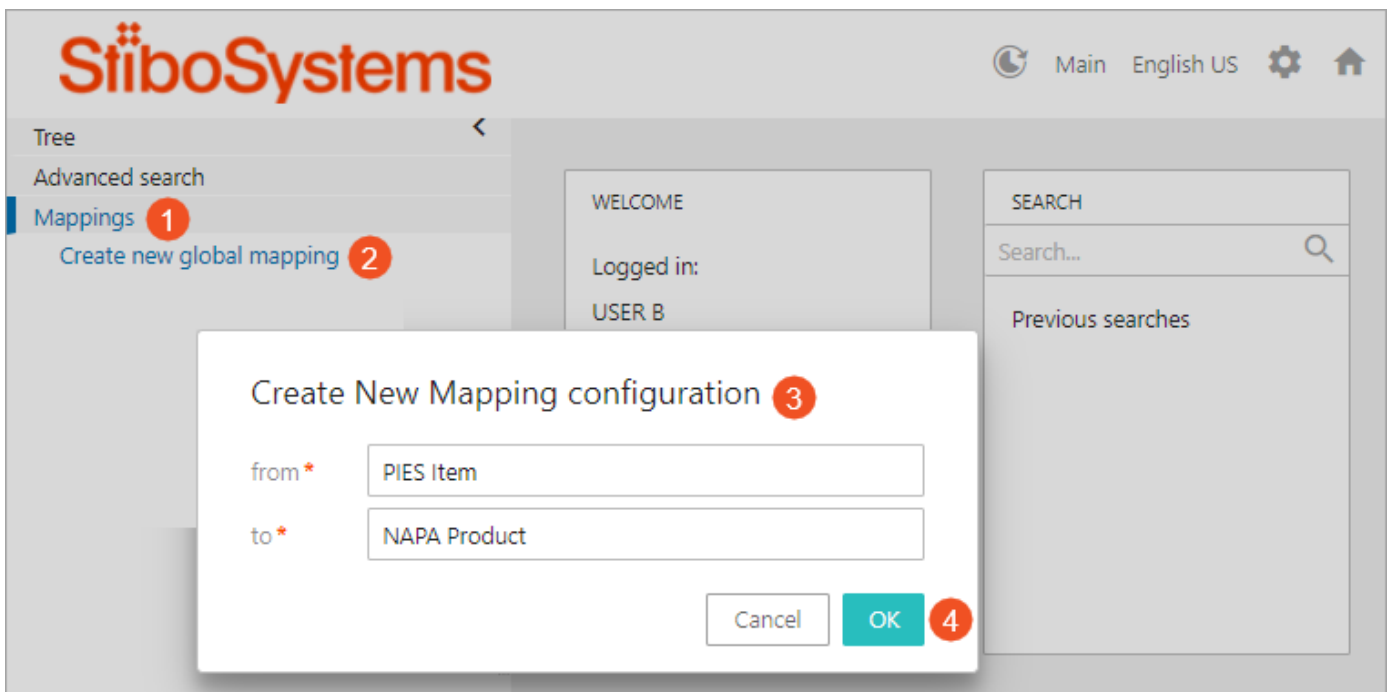


2. In the dialog that appears, type in a suitable ID and Name, and then click 'Create.' In the screenshot below, a Mapper Configuration named 'PIES Item → NAPA Product' is created under setup group called 'Mappings.'



Creating Mapper Configuration Setup Entity in Web UI

1. From the Web UI homepage, select the 'Mappings' component from the left navigation.



2. Click on the 'Create new global mapping' link, and the 'Create New Mapping configuration' dialog displays.
3. Type in a suitable name in the 'from' and 'to' fields. Usually, these fields are defined by an identifiable name of the source and target object types. In the example above, 'from' and 'to' fields are populated with values 'PIES Item' and 'NAPA Product' respectively.
4. Click **OK**, and the new mapper configuration named 'PIES Item → NAPA Product' is created under setup group 'Mappings' in the left stack panel (as shown below). Also in the image below, a Unicode arrow → is created between the Source and the Target names demonstrating the direction of the intended mapping.

SiiboSystems Main English US

Tree <

- Advanced search
- Mappings
- Create new global mapping
- PIES Item → NAPA Product

PIES Item → NAPA Product

Setup

Naming

- * Map from: PIES Item
- * Map to: NAPA Product

Global Configuration

- * Source: []
- * Reference: []
- * Target: []
- Reverse Reference:
- Approved Workspace:

Mapping description

Description: []

Save Duplicate Delete

Important: If the Mapper Configuration is deleted in Web UI, then it will get moved to the Recycle Bin folder in System Setup. If the user tries to create a new mapping with the same text string in the 'from' and 'to' field, then the user will get the Unexpected error. If users want to reuse the same text string in the 'from' and 'to' field when creating a new mapping, then either purge the deleted Mapper Configuration setup entity from the Recycle Bin, or create a Mapper Configuration with some different text, click OK, then go back and change it to be what the user requires.

Configuring Mapper Configuration Setup Entity in Web UI

A newly created Mapper Configuration setup entity must first have the Naming and Global Configuration parameters defined and saved in the Setup tab before the Mappings tab will appear for the user to define mapping rules. This is achieved in the Onboarding Mappings Details Screen for the Mapper Configuration.

When a new Mapper Configuration is first created and selected in Web UI, the Setup tab will be empty (as shown below).

The screenshot shows the StiboSystems web interface for configuring a mapping. The left sidebar contains a 'Tree' view with 'Mappings' selected, showing a list of mappings including 'AC Part Type → Own Part Type' and 'PIES Item → NAPA Product'. The main content area is titled 'PIES Item → NAPA Product' and has a 'Setup' tab. The 'Naming' section includes 'Map from' (PIES Item) and 'Map to' (NAPA Product). The 'Global Configuration' section includes 'Source', 'Reference', and 'Target' fields, each with a node picker icon. There are also checkboxes for 'Reverse Reference' and 'Approved Workspace'. At the bottom, there are buttons for 'Save', 'Duplicate', 'Delete', and 'Reset'.

1. Populate the available parameters as explained below.

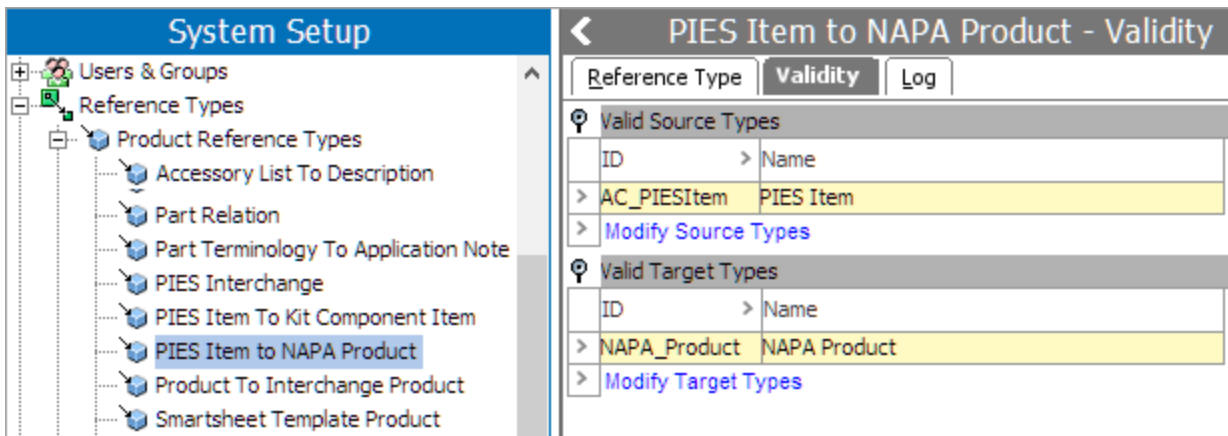
Map from: By default, this mandatory parameter is prepopulated with the value defined while creating the Mapper Configuration in Web UI. However, this field is blank if the Mapper Configuration is created in workbench and needs to be populated with an identifiable name of the source object type (this can be any string that the user wants to enter, e.g., PIES Item). Optionally, if a different name is desired, edit the name to be something more suitable.

Map to: By default, this mandatory parameter is prepopulated with the value defined while creating the Mapper Configuration in Web UI. However, this field is blank if the Mapper Configuration is created in workbench and needs to be populated with an identifiable name of the target object type (this can be any string that the user wants to enter, e.g., NAPA Product). Optionally, if a different name is desired, edit the name to be something more suitable.

Source: By default, this mandatory parameter is blank. Start typing in the empty field to enter the source object type, or click the node picker to browse or search for the object type which defines the source object in the system. This parameter defines the source object type of where it is being mapped from (this must be an existing object in STEP, e.g., AC_PIESItem).

Reference: By default, this required parameter is blank. Start typing in the empty field to enter the reference type, or click the node picker to browse or search for the Reference Type in system, so that STEP can follow this reference type and will establish the link between the Source object and the Target object. This Reference

Type should contain the object types defined in the Source and Target parameters as Valid Source Types and Valid Target Types respectively in the system.



Target: By default, this mandatory parameter is blank. Start typing in the empty field to enter the target object type, or click the node picker to browse or search for the target object type in STEP. This parameter defines the target object type of where it is being mapped to (this must be an existing object in STEP, e.g., NAPA_Product).

Reverse Reference: By default, this optional parameter is unchecked. Selecting this field will work the other way of the reference type defined in the Reference parameter above by using the 'Referenced By' tab to find the target. For example, if a mapper configuration with mapping defined from PIES Item to NAPA Product, checking this parameter will reverse the mapping definition, i.e., the mapping shall be defined to happen from NAPA Product to PIES Item.

Approved Workspace: By default, this optional parameter is unchecked. Selecting this field will take all values from the Approved workspace from the Source object and create them in the Main workspace on the Target. For example, if a mapper configuration with mapping defined from PIES Item to NAPA Product, checking this parameter will select all values present only in the Approved workspace of the PIES Item object and create them in the Main workspace on the NAPA Product.

2. Click **Save**, and a new Mappings tab next to the Setup tab gets created to add and configure newer mapping plugins.

StiboSystems Main English US

Tree <

- Advanced search
- Mappings
 - Create new global mapping
 - AC Part Type -> Own Part Type
 - PIES Item -> NAPA Product**

PIES Item -> NAPA Product

Mappings Setup

Naming

- * Map from: PIES Item
- * Map to: NAPA Product

Global Configuration

- * Source: PIES Item (AC_PIESItem)
- * Reference: PIES Item to NAPA Product (PIESItemtoNAPAProduct)
- * Target: NAPA Product (NAPA_Product)

Reverse Reference

Approved Workspace

Save Duplicate Delete Reset

StiboSystems Main English US

Tree <

- Advanced search
- Mappings
 - Create new global mapping
 - AC Part Type -> Own Part Type
 - PIES ITEM -> NAPA Product**

PIES ITEM -> NAPA Product

Mappings Mappings Setup

+ New mapping

Title	Mapping Type	Health

Save Duplicate Delete Reset

Using the Onboarding Mappings Details Screen

The Onboarding Mappings Details screen allows users to view and edit Mapper Configurations in Web UI. Key features include:

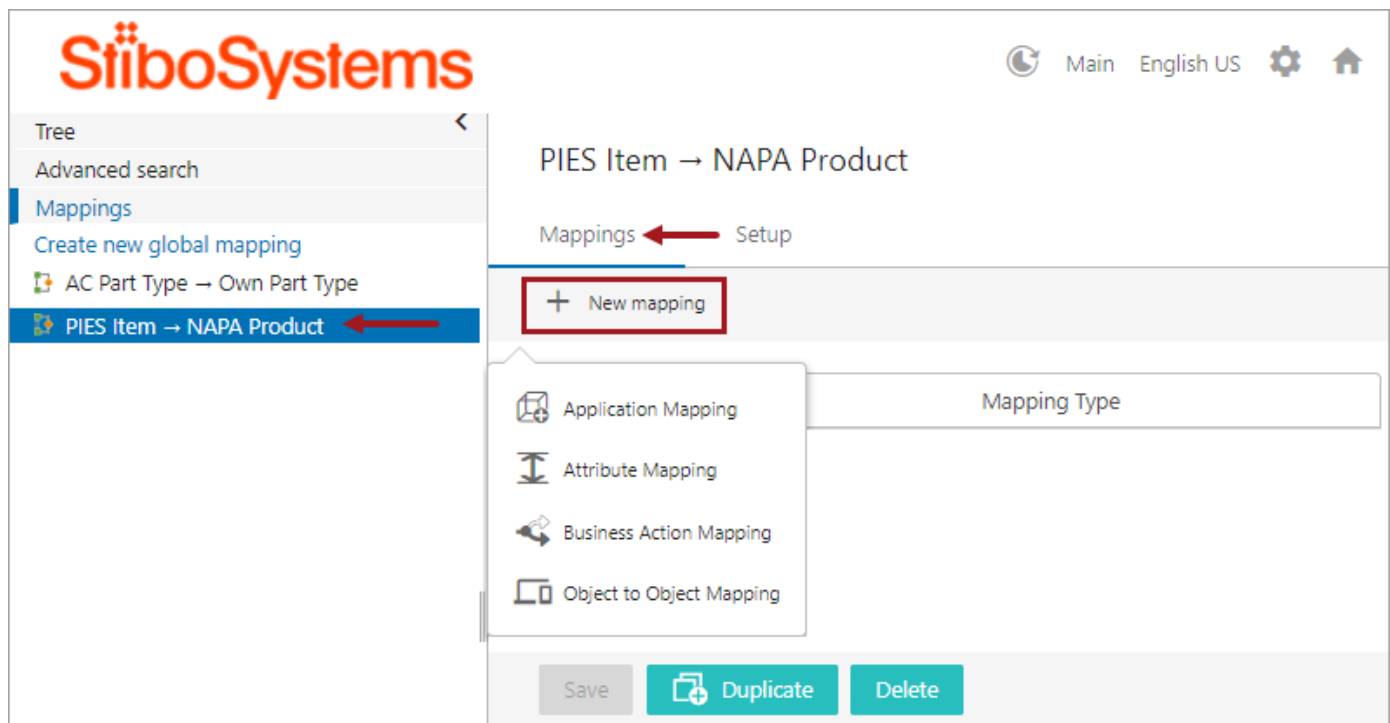
- Can view existing mapping plugin and its configuration
- Can create new mapping plugins
- Can define global configurations for the selected Mapper Configuration setup entity

Mapper Configuration setup entity can be configured to meet different Onboarding and Offboarding requirements in Onboarding Mappings Details screen.

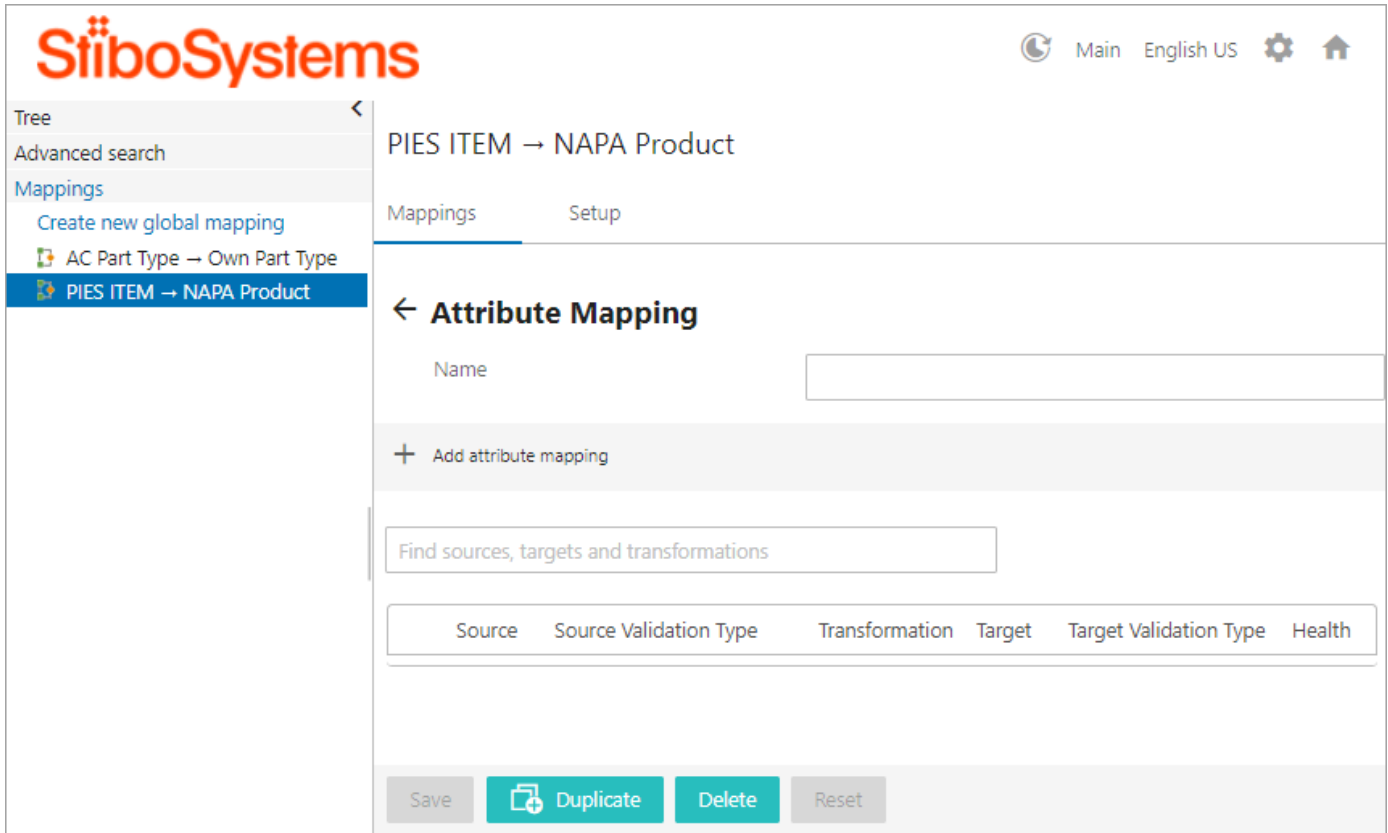
Creating Mapping Plugins

Before adding any plugin for the Mapper Configuration in Onboarding Mappings Details screen, it is essential that the plugins that need to be listed in the Onboarding Mappings Details screen are configured within the Mappings parameter of Onboarding Mappings Details Screen Properties. For more information, see topic **Onboarding Mappings Details Screen**.

1. With the required Mapper Configuration selected, click on **New mapping** within the Mappings tab of the Onboarding Mappings Details screen.

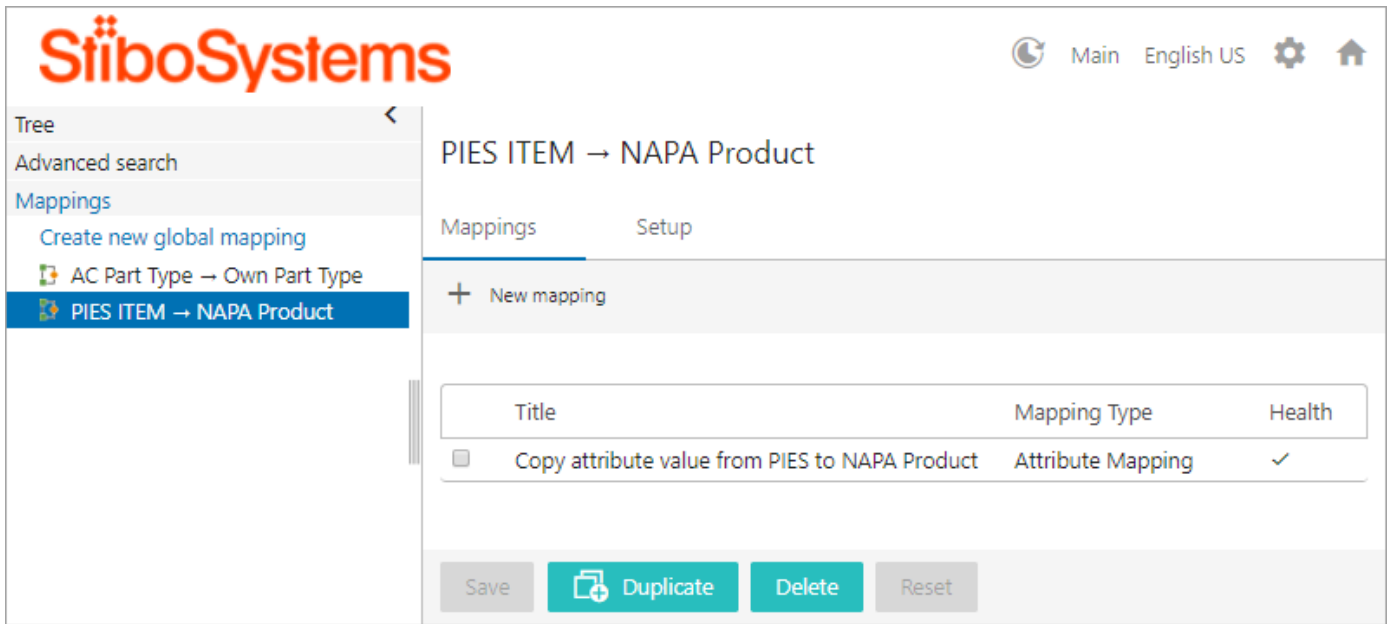


2. Select the desired type of mapping plugin, and a screen is displayed prompting the user to further configure the plugin. For example, in the screenshot below, a screen is displayed when the Attribute Mapping plugin is selected.



3. When done configuring the plugin, click **Save** button available at the bottom of the screen. The newly added mapping plugin is listed within the Mappings tab of the Onboarding Mappings Details screen (as shown below).

Important: The **Delete**, **Duplicate** and **Reset** buttons present at the below toolbar will delete / duplicate / reset the Mapper Configuration, NOT the individual mapping plugin rows within the configuration. If the user does not want to be able to delete the Mapper Configuration in Web UI, then do not configure the **Delete** Action.



Managing Mapping Plugins

All of the created Mapping plugins are listed in a table format within the Mappings tab of the Onboarding Mappings Details screen for the selected Mapper Configuration. The user can create, test, and delete the mapping plugins within the Mappings tab of the Onboarding Mappings Details screen.

Deleting an Existing Mapping Plugin

1. Select the Mapping plugins by clicking on the checkbox available at the left side of the listed Mapping plugins. A **Delete mapping** button will be displayed at the top bar (as shown below).

The screenshot shows the SiiboSystems interface for managing mappings. The left sidebar contains a tree view with 'PIES ITEM → NAPA Product' selected. The main area is titled 'PIES ITEM → NAPA Product' and has two tabs: 'Mappings' (active) and 'Setup'. In the 'Mappings' tab, there are three buttons: '+ New mapping', 'Delete mapping' (highlighted with a red box), and 'Test mapping'. Below the buttons is a table with the following data:

Title	Mapping Type	Health
<input type="checkbox"/> Copy attribute value from PIES to NAPA Product	Attribute Mapping	✓
<input checked="" type="checkbox"/> Create AC PIES Items for NAPA Parts	Object to Object Mapping	✗

At the bottom of the interface, there are four buttons: 'Save', 'Duplicate', 'Delete', and 'Reset'.

2. Click on the **Delete mapping** button, and the mapping plugin will be deleted from the table.

Executing Mapper Configuration Setup Entity

The Mapper Configuration Setup Entity can be executed in the following ways:

- By configuring it in a business rule and executing the business rule on the source object
- By a background process

Prerequisites

It is assumed that the admin user has knowledge of STEP administrative functions and experience working in System Setup, including creating and editing business rules, workflows, and Web UIs. This section targets only the specific information needed for a knowledgeable STEP admin user to execute the Mapper Configuration in different possible ways. For more introductory material of these concepts, see the **Business Rules**, **Workflows** and **Web User Interfaces** sections of **STEP Online Help**.

Before any Mapper Configuration is executed, the relationship must be established through references between the source object to the target object.

Before any Mapper Configuration is executed, it is necessary that the configuration aspects of the Data Onboarding solution in both the workbench and the Web UI are completed. For more information about creating and configuring the mapper configuration setup entity, see the **Data Onboarding Automation Solution Initial Setup**, and the **Configuring Web UI for Data Onboarding Automation solution** topics within this section.

Executing by a Business Rule

This process involves configuring a business action to hold the Mapper Configuration and run this business action on the source object in different possible ways. The execution may be by running this business action inside a workflow, or run on approving the object, or any other places that can run a business action (e.g., run as a bulk update on a Collection, or initiating the business action while a node is selected in Web UI).

1. Create and configure a business action using the 'Execute mapper configuration' operation. For the full set of instructions on configuring the 'Execute mapper configuration' business action, see the **Configuring the Execute Mapper Configuration Business Action** documentation.
2. Review and implement the most common ways to use business rules as described in **Using Business Rules in STEP** topic within the **Business Rules** section of **STEP Online Help**.

Executing by a Background Process

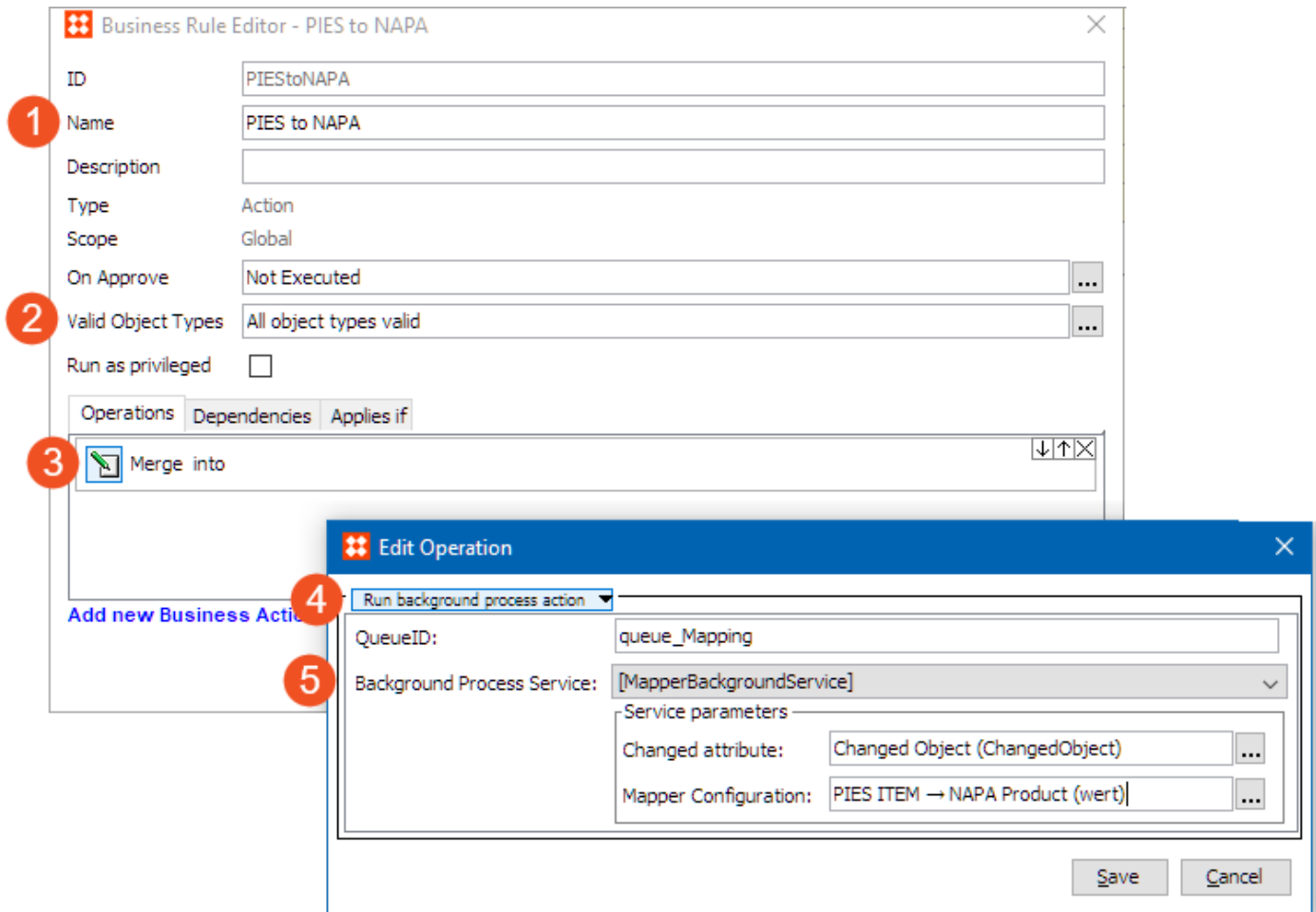
To execute the Mapper Configuration by a background process, the following two levels of configuration are needed:

- Create a Business Action with 'Run background process action' operation
- Create a new state after the 'Import' state in the Import workflow and add the business action in the newly created state

These two levels of configuration are explained in detail below. For more information on creating / editing the business action and workflows, see the **Business Action** section, and the **Workflows** section of the **STEP Online Help**.

Create a business action to set the Mapper Configuration to execute after import

Below are the steps for creating a business action with the 'Run background process action' operation. For more information on creating a business rule, see the **Creating a Business Rule or Library** topic of the **Business Rules** section of the **STEP Online Help**.



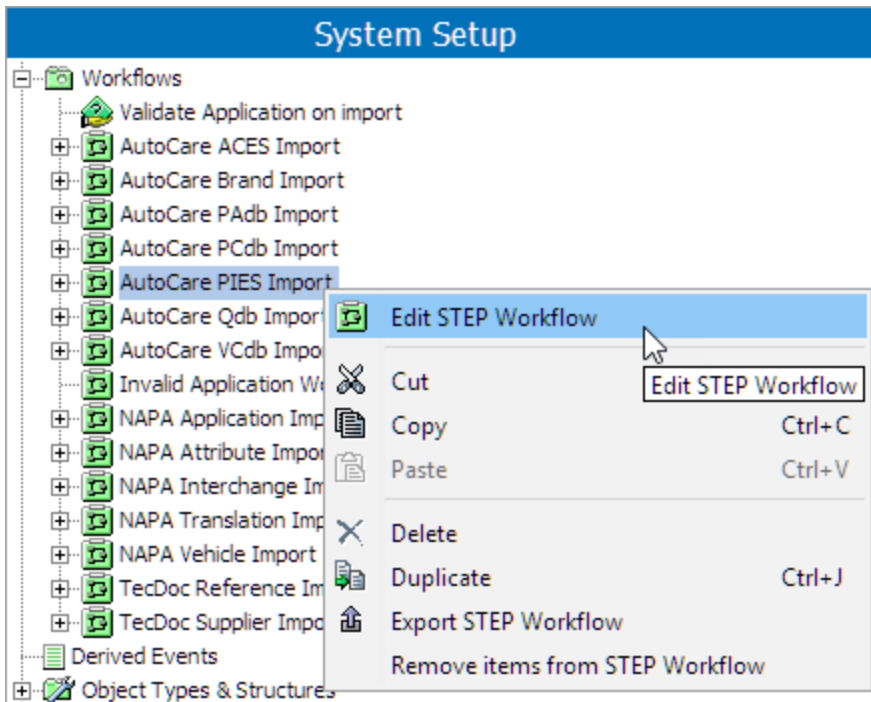
1. Create the business action with a name that accurately describes to the user what this action will do. For this example, the business action name is 'PIES to NAPA.'
2. Edit the new business rule, click the ellipsis button (...) next to the Valid Object Types parameter and select the 'All Object Types' for this business action. If the business rule should be more restrictive to specific object types, then the user should specify those object types instead of make it valid for all object types.
3. On the Operations tab of the Business Rule Editor, click the **Add new Business Action** link, and click the edit button to open the Edit Operation dialog.
4. Use the dropdown menu within the Edit Operation dialog to select **Import flow > Run background process action** operation.
5. Select **[MapperBackgroundService]** in the dropdown next to the Background Process Service field and the following parameters will display (as shown above). Configure the following parameters:

- **QueueID:** Required parameter for all background process actions that specifies the queue in which the background process should run. By default, this field is empty. In the example above, the value 'queue_Mapping' is populated. This parameter is specified for the action itself, not the particular service within the action, though the outcome is the same as each action runs only a single service.
 - **Background Process Service:** Required parameter for all background process actions that specifies the background process. This should always be populated with [MapperBackgroundService].
 - **Changed attribute:** Click the ellipsis button (...) to find and select an attribute that is used to identify when an object is changed while importing. This must be the same attribute that is configured within the 'Updated values attribute' parameter in the 'Set import status attributes' operation. When the attribute is configured, the system will search for all the objects where the configured attribute holds the value 'true' and executes the Mapper Configuration setup entity on all the objects with Changed attribute value = true. For more information on how to configure this attribute in 'Set import status attributes' operation, see the **Business Action: Set Import Status Attributes** topic of the **Automotive Reference Guide** found within the **Solution Enablement** section of **STEP Online Help**.
 - **Mapper Configuration:** Required parameter to specify the Mapper Configuration that needs to be executed.
6. Click the **Save** button to save and close the configuration.

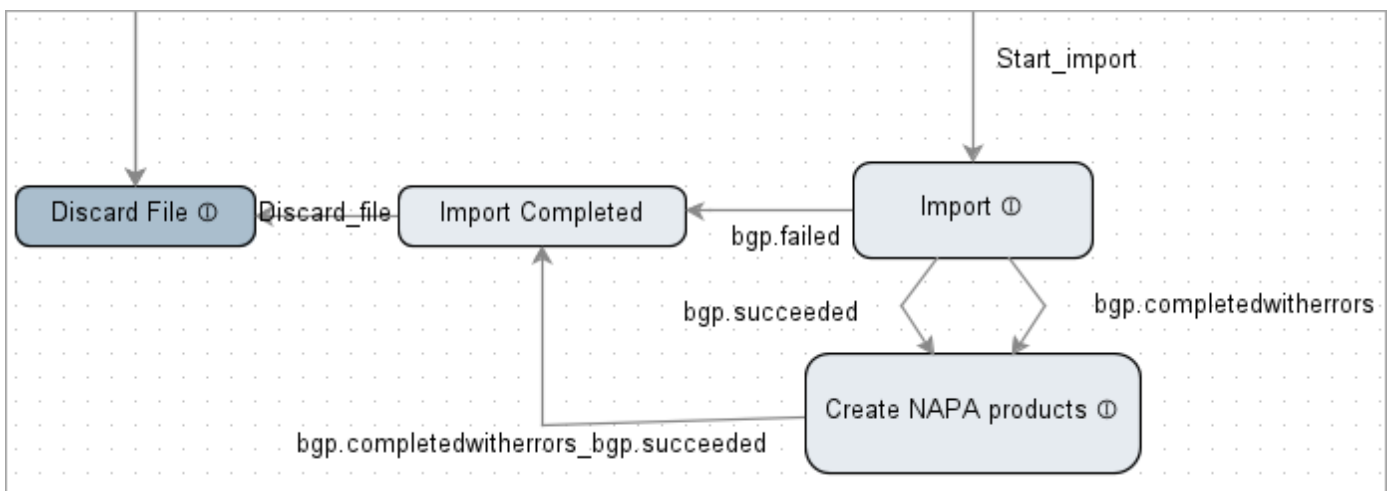
Configure the business action in the Import Workflow

Once the business action is created, the import workflow needs to be updated to use the business rule. To achieve this, a new state after the Import state is created, and the business action is added to the newly created state in the import workflow. Below are the steps to update a workflow to use the business rule created in the previous section.

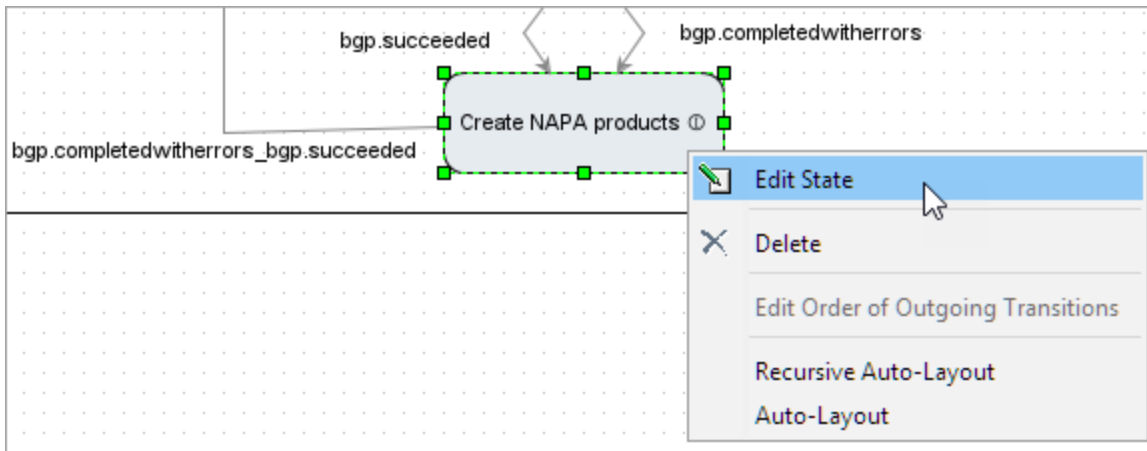
1. Go to System Setup > Workflows > Select the workflow used by the importer.
For the business action example provided above, the configured valid object types allow for the business action to be used in AutoCare PIES Import.
2. Right-click the respective workflow, select **Edit STEP Workflow**, and the STEP Workflow Designer will display.



3. Within the STEP Workflow Designer, create a new state between the Import state and Import Completed state. In this example, a new state called 'Create NAPA Products' is created (as shown below). For a detailed procedure on creating States and Transitions, see **Creating a Workflow** topic of the **Workflows** section of the **STEP Online Help**.



4. Right-click the newly created state (in this case it is 'Create NAPA products' state), select **Edit State**, and the State Editor will display.



5. Within the State Editor, click the **On Entry** tab (as shown below).

6. Within the 'Business Rule' parameter, click the ellipsis button (...) to find and select the business action created to set the Mapper Configuration after import.

Below is an example of how the State Editor should look prior to saving:

State Editor
✕

Edit

Deadline/Escalation	Comments	Web UI Screen Mapping	Mandatory Attributes
State	On Entry	On Exit	Assignee

Business Rule:

Name:

Description:

Type:

Scope:

Valid Object Types: ...

Run as privileged:

Operations | Dependencies | Applies if

RunBackgroundProcessServiceAction: queue_Mapping, Service: 'MapperBackgroundService'
↓↑✕

Add new Business Action

Show Edited State

7. Close the State Editor, and save and close the STEP Workflow Designer.

Mapping Plugins

Within each Mapper Configuration, mapping plugins are added to define what data needs to be retrieved and moved between the Source and the Target object. The settings available in these plugins allow more control over the behavior of the Mapper Configuration. The plugin suite for automotive data mapping consists of the following four plugins:

- Application Mapping Plugin
- Attribute Mapping Plugin
- Business Action Mapping Plugin
- Object to Object Mapping Plugin

Details on how to configure and use these plugins are provided in their respective topics below. For more information about how to add these plugins, see the **Using the Onboarding Mappings Details Screen** topic.

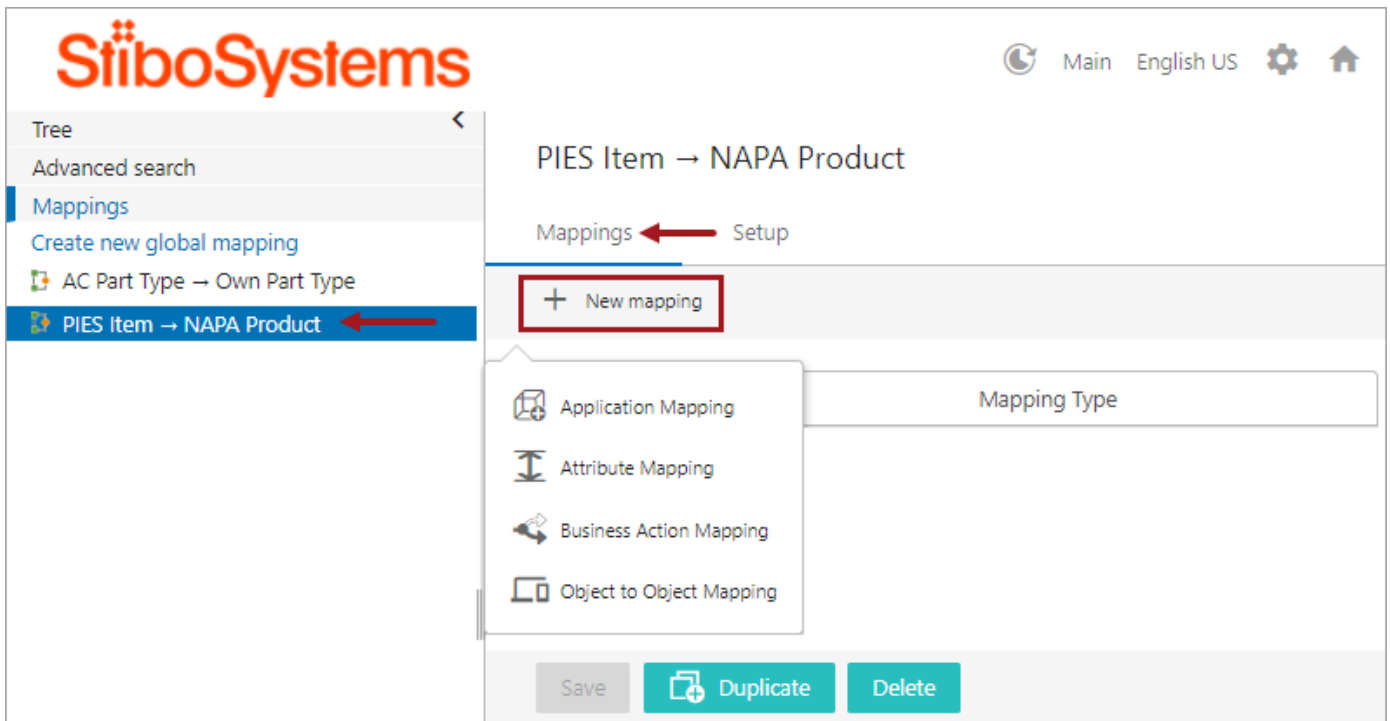
Attribute Mapping Plugin

The Attribute Mapping plugin will copy attribute values from the source object to the target object. It is configured on the Onboarding Mappings Details Screen for the selected Mapper Configuration. The settings available in this plugin allow the user to define what the Mapper Configuration should do with the attribute values.

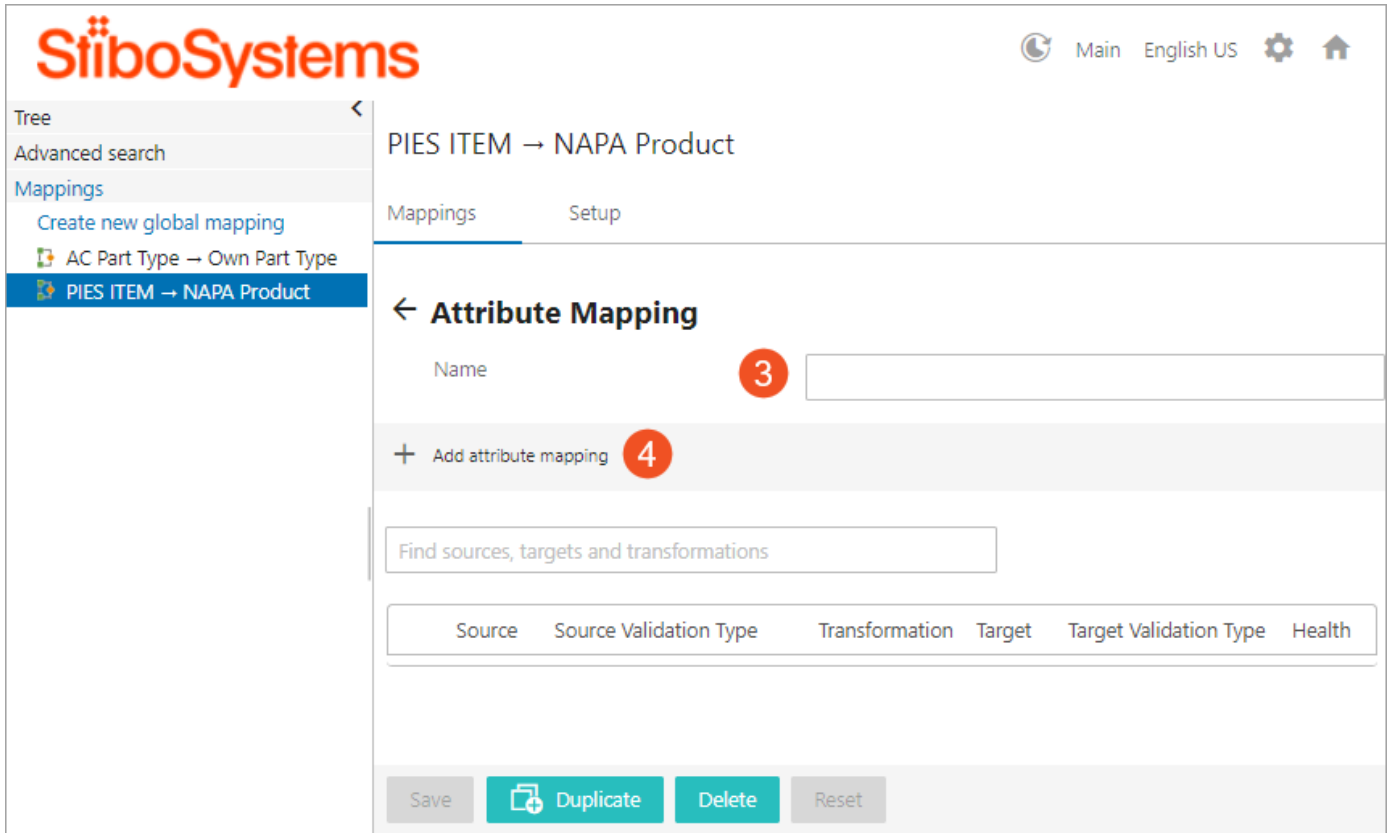
Note: All settings are not necessary for all attributes. The specific combinations chosen are dependent on the unique requirements for the attribute being configured. The settings contained within this plugin can include transformations. The plugin can be configured to either Onboard / Outboard the data or in some cases can be configured to work along with transformations, too.

To configure the Attribute Mapping plugin, follow these steps.

1. With the required Mapper Configuration selected, click on **New mapping** available within the Mappings tab of the Onboarding Mappings Details screen.



2. Select Attribute Mapping, and a screen is displayed prompting the user to further configure the plugin (as shown below).



3. Type in a suitable name next to Name field. This could be any unique name that clearly describes the mapping functionality.
4. Click the 'Add attribute mapping' icon, and the Mapping Guide window will display (as shown below).

Mapping Guide ✕

Mapping

	ID	Name	Validation Type
Source			
Target			
Transformation			

☰+
 Suppress

Sources Targets Transformations

Search

Find sources

Select Attribute

ID	Title	Validation Type	Attribute Group(S)
AC_PAdb_7966	12 Volt DC Outlet Quantity	Number	PAdb Attributes
AC_PAdb_9304	2nd Ring Face Design	Alpha Numeric	PAdb Attributes
AC_PAdb_9309	2nd Ring Material	Alpha Numeric	PAdb Attributes
AC_PAdb_9321	2nd Ring Radial Wall Thickness	Number	PAdb Attributes
AC_PAdb_9316	2nd Ring Width	Number	PAdb Attributes
AC_PAdb_7914	360 Degree Rotation	LOV(Text)	PAdb Attributes
AC_PAdb_9303	3rd Ring Face Design	Alpha Numeric	PAdb Attributes
AC_PAdb_9308	3rd Ring Material	Alpha Numeric	PAdb Attributes
AC_PAdb_9320	3rd Ring Radial Wall Thickness	Number	PAdb Attributes
AC_PAdb_9315	3rd Ring Width	Number	PAdb Attributes

⏪ ◀ 1-10 of 8,089 ▶ ⏩

✓ OK
✕ Cancel

- With the Sources tab selected, click in the Search field and start typing the initial letters of the attribute name or ID. This brings up a dropdown of typeahead search results listing the attributes of the Source object available in the system from where the attribute value could be retrieved. Select the attribute from the result list displayed below the search bar.

Mapping Guide ✕

Mapping

ID	Name	Validation Type
Source		
Target		
Transformation		☰+

Suppress

→ Sources Targets Transformations

Search

5

Select Attribute

ID	Title	Validation Type	Attribute Group(S)
AC_PAdb_2170	Connector Color ←	Text	PAdb Attributes
AC_PAdb_5120	Electrical Connector Color	Text	PAdb Attributes
AC_PAdb_4500	End 1 Connector Color	Text	PAdb Attributes
AC_PAdb_4501	End 2 Connector Color	Text	PAdb Attributes
AC_PAdb_6712	End 3 Connector Color	Text	PAdb Attributes
AC_PAdb_6713	End 4 Connector Color	Text	PAdb Attributes
AC_PAdb_4738	Hose Connector Color	Text	PAdb Attributes

⏪ ⏩ 1-7 of 7

Once selected, the attribute will be populated in the Source field (as shown below)

Mapping Guide ✕

Mapping

	ID	Name	Validation Type
Source	AC_PAdb_2170	Connector Color	Text
Target			
Transformation			☰+

Suppress

Sources Targets Transformations

Search

Connector color

Select Attribute

ID	Title	Validation Type	Attribute Group(S)
AC_PAdb_2170	Connector Color	Text	PAdb Attributes
AC_PAdb_5120	Electrical Connector Color	Text	PAdb Attributes
AC_PAdb_4500	End 1 Connector Color	Text	PAdb Attributes
AC_PAdb_4501	End 2 Connector Color	Text	PAdb Attributes
AC_PAdb_6712	End 3 Connector Color	Text	PAdb Attributes
AC_PAdb_6713	End 4 Connector Color	Text	PAdb Attributes
AC_PAdb_4738	Hose Connector Color	Text	PAdb Attributes

⏪ ⏩ 1-7 of 7 ⏪ ⏩

✓ OK
✕ Cancel

6. With the Targets tab selected, click in the Search field and start typing the initial letters of the attribute name or ID. This searches for the attribute from the Target object where the retrieved value should be copied to. Select the attribute from the result list displayed below the search bar.

Mapping Guide ✕

Mapping

	ID	Name	Validation Type
Source	AC_PAdb_2170	Connector Color	Text
Target			
Transformation			☰+

Suppress

Sources Targets Transformations

Search

Color

Select Attribute

ID	Title	Validation Type	Attribute Group(S)
NAPA_PHdb_23731	Color	Text	NAPA Product Attributes

⏪ ⏩ 1-1 of 1 ⏪ ⏩

✓ OK
✕ Cancel

Once selected, the attribute will be populated in the Target field (as shown below).

Mapping Guide ✕

Mapping

	ID	Name	Validation Type
Source	AC_PAdb_2170	Connector Color	Text
Target	NAPA_PHdb_23731	Color	Text

Transformation ≡+

✓ Valid configuration Suppress

Sources Targets Transformations

Search

Color

Select Attribute

ID	Title	Validation Type	Attribute Group(S)
NAPA_PHdb_23731	Color	Text	NAPA Product Attributes

⏪ ⏩ 1-1 of 1

✓ OK
✕ Cancel

Note: Only attributes that are valid for the object type that is defined in the mapping Setup tab will be displayed as options to select from in the Sources and Targets tabs.

After the source and target attributes are defined, the system evaluates the validity match between the source and the target attributes. A hyperlink text explaining the reason for the validity match / mismatch gets displayed. Clicking on the hyperlink will open the 'Detailed Information' dialog and displays the validity match / mismatch information of the attribute mapping. Below is an example of a validation mismatch between the source and the target attribute displayed in the 'Detailed Information' dialog:

Mapping Guide

Mapping

	ID	Name	Validation Type
Source	AC_PIES_ITEMHazardousMaterialCode	Hazardous Material Code	LOV(Text)
Target	NumAttr	Number Attribute	Number

Transformation

▲ Validation Types does not match

 Suppress

Sources **Targets** Transformations

Search

Detailed Information

	Source	Target	Evaluation
ID	AC_PIES_ITEMHazardousMaterialCode	NumAttr	
Name	Hazardous Material Code	Number Attribute	
Title	Hazardous Material Code	Number Attribute	
Node Type	attribute	attribute	
Validation Type	LOV(Text)	Number	Validation Types does not match
Unit			
LOV	Yes	No	
Mandatory	No	No	
Derived	No	No	
Dependent	No	No	
Description	No	No	
Inherited	Yes	Yes	
Multi Valued	No	No	
Language Dependent	No	No	
Health	Yes	Yes	No

✓ Close

The user can suppress the validation mismatch warning message by clicking on the 'Suppress' checkbox in the Mapping Guide window. Selecting the 'Suppress' checkbox will *only* remove the data type mismatch warnings displayed on the Mapping Guide window and *does not resolve* the mismatch irregularities.

Mapping Guide

Mapping

	ID	Name	Validation Type
Source	AC_PIES_ITEMHazardousMaterialCode	Hazardous Material Code	LOV(Text)
Target	NumAttr	Number Attribute	Number

Transformation ≡+

✓ Valid configuration ← ☑ Suppress

Sources Targets Transformations

Detailed Information

	Source	Target	Evaluation
ID	AC_PIES_ITEMHazardousMaterialCode	NumAttr	
Name	Hazardous Material Code	Number Attribute	
Title	Hazardous Material Code	Number Attribute	
Node Type	attribute	attribute	
Validation Type	LOV(Text)	Number →	Validation Types does not match
Unit			
LOV	Yes	No	
Mandatory	No	No	
Derived	No	No	
Dependent	No	No	
Description	No	No	
Inherited	Yes	Yes	
Multi Valued	No	No	
Language Dependent	No	No	
Health	Yes	Yes	No

Close

- With the 'Transformations' tab selected, click in the search field and start typing the initial letters of the transformation name or ID. This brings up a dropdown of typeahead search results listing the attribute transformations available in the system. Select the relevant transformation from the list displayed below the search bar.

Note: The transformations must be created and defined in workbench System Setup prior to being available to be selected in the Mapping Guide.

Mapping Guide

Mapping			
	ID	Name	Validation Type
Source	AC_PAdb_2170	Connector Color	Text
Target	NAPA_PHdb_23731	Color	Text

Transformation ☰+

✓ Valid configuration Suppress

Sources Targets Transformations

Search

7 Append

Select transformation

ID	Title	Parent Node	Type
AppendColor	Append Color	Transformations	Change case

1-1 of 1

The selected transformation will be populated in the Transformation field (as shown below).

Mapping Guide

Mapping			
	ID	Name	Validation Type
Source	AC_PAdb_2170	Connector Color	Text
Target	NAPA_PHdb_23731	Color	Text
Transformation	AppendColor	Append Color	Change case

✓ Valid configuration Suppress

Sources Targets Transformations

Search


Find transformations

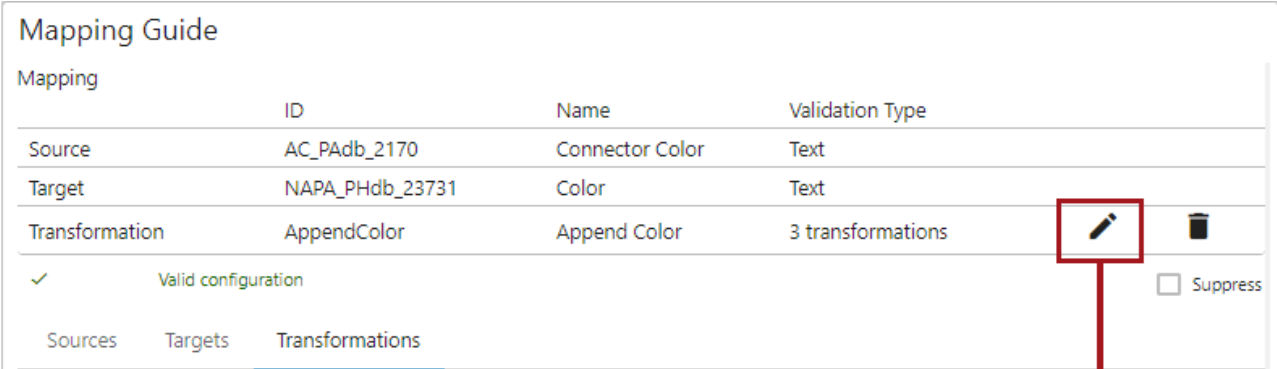
Select transformation

ID	Title	Parent Node	Type
AppendColor	Append Color	Transformations	Change case



1-1 of 1

The populated transformation can be deleted by clicking the delete icon () in the Transformation field.

Clicking the edit icon () will open the Transformation Overview dialog and lets the user test the functionality of the transformation via the **Test** button (as shown below). For more information on this **Test** button, see the **Testing Transformations Configured in Attribute Mapping Plugin** topic within this guide.




Mapping Guide

Mapping	ID	Name	Validation Type	
Source	AC_PAdb_2170	Connector Color	Text	
Target	NAPA_PHdb_23731	Color	Text	
Transformation	AppendColor	Append Color	3 transformations	 
✓	Valid configuration			<input type="checkbox"/> Suppress


Sources Targets Transformations


Transformation Overview

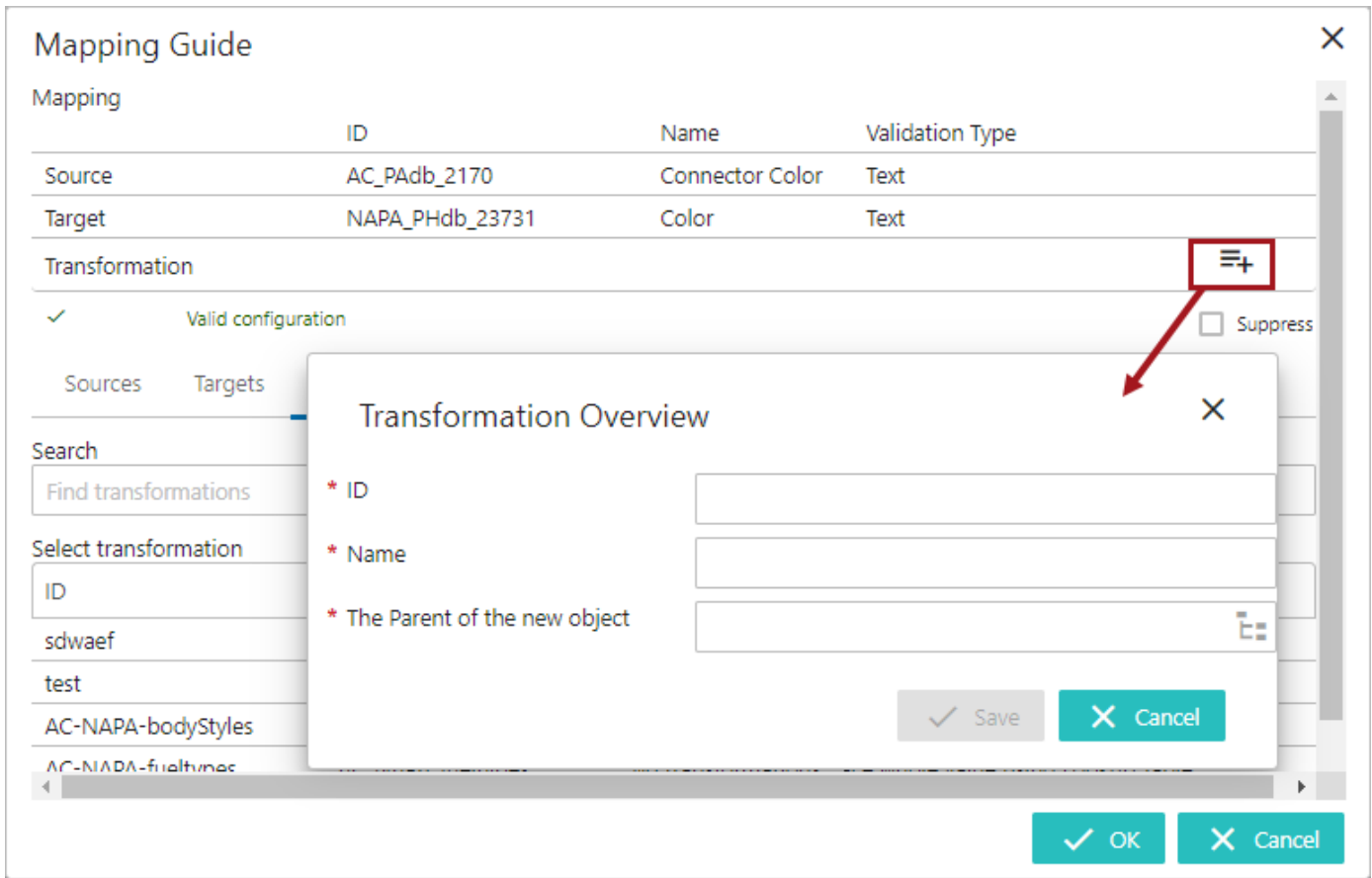
 Test

ID	Title	Prompt
ChangeCase	Change case	Change to "Upper" case
Append	Append	Append "NAPA"
NumberWithUnitFormat	Format number with unit	Format number according to format "#.00". Localize "LocalizeBooleanFalse"

Save Cancel

The attribute transformations can also be created in the Mapping guide window by clicking on the Create New Transformation icon () available within the Transformation field. Users can define the ID, Name, and the hierarchy in which the attribute transformation should be stored in.

Important: The Create New Transformation icon () would only create the attribute transformation, and the user needs to configure and define the functionality of the transformation separately in the workbench.



For information on configuring the attribute transformations, see the **Attribute Transformations** topic within the **System Setup** section of **STEP Online Help**. And for details on each of the available transformation options, see the **Transformations** topic in the **Resource Materials** online help.

- Click **OK** to close the Mapping Guide window and then click **Save** to save the changes. The newly added attribute mapping row will be listed as shown below. To edit the attribute mapping selections, click on the row and it will open up the Mapping Guide window in order to make edits.

Note: Once the user adds a mapping and selects the Source and Target attribute and clicks **OK** to close out of the Mapping Guide window, the **Save** button is enabled until the user clicks on **Save** button. If the user fails to click **Save**, then the mapping will be lost once user selects a new mapping configuration.

The screenshot shows the 'Attribute Mapping' configuration interface. At the top, the page title is 'PIES ITEM → NAPA Product'. Below the title, there are tabs for 'Mappings' and 'Setup'. The main heading is 'Attribute Mapping'. A text input field for 'Name' contains the value 'Copy attribute value from PIES to NAPA Product'. Below this, there are three action buttons: '+ Add attribute mapping', 'Delete', and 'Disable'. A search bar with the placeholder text 'Find sources, targets and transformations' is present. Below the search bar is a table with the following columns: Source, Source Validation Type, Transformation, Target, and Target Validation Type. The table contains one row: 'Connector Color' (Source), 'Text' (Source Validation Type), 'Append Color' (Transformation), 'Color' (Target), and 'Text' (Target Validation Type). A red arrow points to a checkmark in the left margin of the table row. At the bottom of the table, there are four buttons: 'Save', 'Duplicate', 'Delete', and 'Reset'.

9. Repeat the above steps 4 to 8 to add more attribute mapping rows for the same plugin.

User can add any number of mapping rows in the mapping plugin. When there are multiple mapper rows available within the mapping plugin, the order of execution of each mapper row is based on the order in which it is listed within the mapping plugin.

The health of the mapping row is displayed next to each mapping row. Users can also add some additional information describing each of the mapper rows. The user has the flexibility to disable, delete, or rearrange the listing order of the mapper rows.

Testing Transformations Configured in Attribute Mapping Plugin

Users can test the attribute transformation configured in the Attribute Mapping plugin in Web UI. When the Attribute Mapping plugin is defined with the source and the target attributes, users can add an attribute transformation if necessary. Users can test the performance and result of the added transformation by entering a sample test value within Web UI as explained below. This helps to test the transformation by evaluating whether the attribute value is getting transformed correctly or not without actually moving the data from one attribute to the other.

When a transformation is added for an attribute mapping, an edit icon (✎) is displayed within the Transformation field in the Mapping Guide window. Clicking the edit icon (✎) will open the Transformation Overview dialog and lets users test the functionality of the transformation via Test button (as shown below).

The screenshot shows the 'Mapping Guide' window with a table of mappings. The 'Transformation' row for 'AppendColor' has an edit icon (✎) highlighted with a red box. A red arrow points from this icon to the 'Test' button in the 'Transformation Overview' dialog. The 'Transformation Overview' dialog has a 'Test' button with a checkmark icon, also highlighted with a red box. Below the 'Test' button is a table of transformations:

ID	Title	Prompt
ChangeCase	Change case	Change to "Upper" case
Append	Append	Append "NAPA"
NumberWithUnitFormat	Format number with unit	Format number according to format "#.00". Localize "LocalizeBooleanFalse"

At the bottom of the dialog are 'Save' and 'Cancel' buttons.

Clicking the Test button in the Transformation Overview dialog will open the Test Transformation dialog with the parameters explained as follows:

- **Test Value:** This editable blank field lets users enter text which shall be considered as the sample value for testing.
- **Timing:** This uneditable field displays the time elapsed in transforming the value mentioned in the 'Test value' field when the **Test** action button is clicked in the Test Transformation dialog.
- **Status:** This uneditable field displays the status of the transformation (success / failure) when the **Test** action button is clicked in the Test Transformation dialog.
- **Result:** This uneditable field displays the final transformed value (result) of the value mentioned in the 'Test value' field.

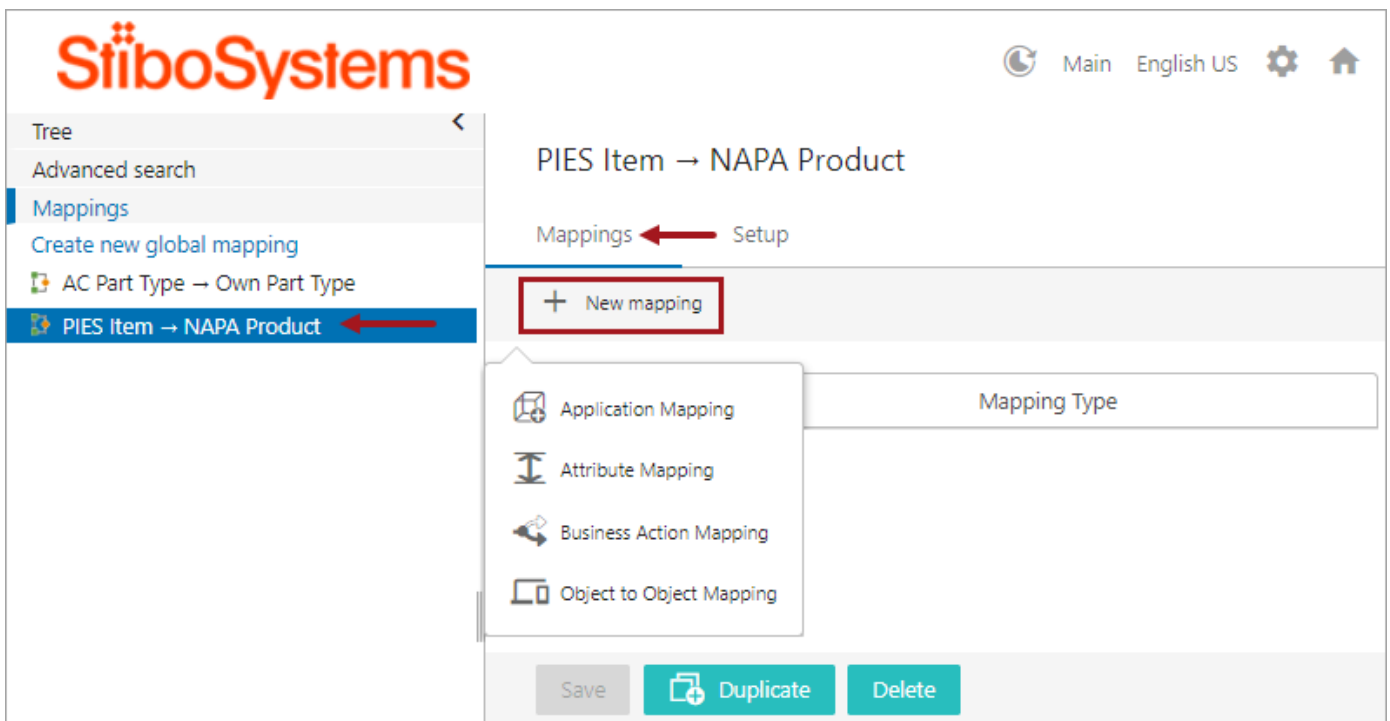
In the example below, a value 'Blue' is entered in the 'Test value' field and, the **Test** action button is clicked to display the result 'NaN.'

Object to Object Mapping Plugin

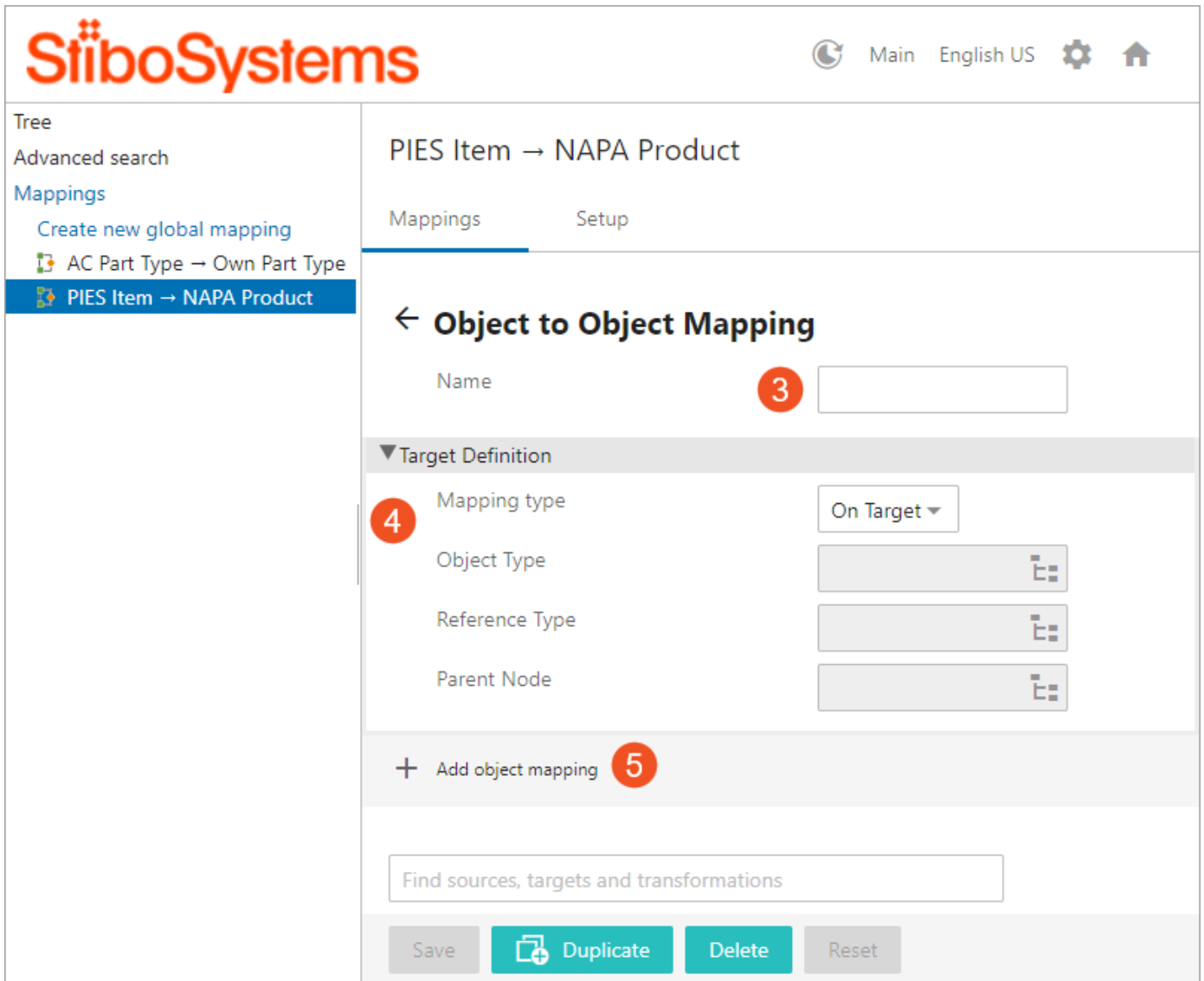
Object to Object Mapping plugin will copy the information present on the Source object or from its related object (Source STEP Path) to the Target object or to its related object. The information retrieved from the Source object could be the ID, name, attribute values, or references. The information from the Source object could be stored directly on Target object, as the child of the Target object, or as the referenced object of the Target object. The Object to Object Mapping plugin is added and configured in the Onboarding Mappings Details Screen for the selected Mapper Configuration.

To configure the Object to Object Mapping plugin, follow these steps:

1. With the required Mapper Configuration selected, click on **New mapping** which is available on the Mappings tab of the Onboarding Mappings Details screen.



2. Select Object to Object Mapping, and a screen displays prompting the user to further configure the plugin. Below is a screenshot of the Mappings tab on the Onboarding Mappings Details screen. The steps required to configure the Object to Object Mapping are denoted with a number in the screenshot. Below the screenshot is a numbered list that describes each step, corresponding to the number shown in the image.



3. Type in a suitable name next to the Name field. This could be any unique name that describes the mapping functionality.
4. Populate the parameters available under the Target Definition flipper as explained below:
 - **Mapping type:** This parameter defines where the retrieved information from the source object need to be stored in the target location. The available options are:
 - **On Target:** Selecting this option will evaluate the Source STEP Path and store the results in the Target object either as an attribute value or as a referenced object. This will *not* create any new objects. When this option is selected, the Object Type, Reference Type, and Parent Node parameters available under the Target Definition flipper are grayed out and are not used with the On Target Mapping type.
 - **Child:** Selecting this option will evaluate the Source STEP Path in order to get the data to be used to create child objects below the Target object. The child objects that are created can either be

Classification or Product objects. The data that is retrieved from the Source object can be used in the new objects as an ID, Name, attribute value, or reference. The object type to be used for the new children objects that will be created below the Target object must already exist. This object type is what will be selected in the Object Type parameter.

When this option is selected, the Reference Type and Parent Node parameters are grayed out and are **not** used with the Child Mapping type.

- **Reference:** Selecting this option will evaluate the STEP Path on the Source object to create new objects below another object, and then establish a reference between the Target object and the newly created object. The data that is retrieved from the Source object can be used in the new objects as an ID, Name, attribute value, or reference. The new object can be created under any node (called as Parent Node) and then a reference is established between the Target object and the new object.
 - **Object Type:** This parameter is used when either the Child or Reference Mapping option is selected within the 'Mapping type' parameter. The user can select the object type to be used for the new objects that will be created below the Target object, or below the defined parent node if the Reference Mapping option is selected.
 - **Reference Type:** This parameter is used when Reference Mapping option is selected within the 'Mapping type' parameter. This parameter allows the user to select the Reference Type that should be used to establish the reference between the Target object and the newly created objects.
 - **Parent Node:** This parameter is only used when Reference Mapping option is selected within the 'Mapping type' parameter. The user can select the parent node under which the newly created objects will reside.
5. Click the 'Add object mapping' icon, and the Mapping Guide window will display.

Mapping Guide

Mapping

ID	Name	Validation Type
Source		
Target		
Transformation		

Suppress

Targets Transformations

Search Filter Options


Find targets Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Attribute	NAPA_PHdb_1161	# Steering gear Box Mounting Holes	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_1205	# Steering Gear Box Turns Lock to Lock	Text	NAPA Product Attributes
Attribute	ChangedObject	Changed Object	Text	Metadata
Attribute	NAPA_PHdb_23731	Color	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_61	Contents	Text	NAPA Product Attributes
Attribute	DeleteStatus	Delete Status	Text	Metadata
Attribute	NAPA_PHdb_3153485	E-Waste	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_1831	Hose Port Type	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_29378	Input Shaft Spline Count	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_3229512	Input Shaft Type	Text	NAPA Product Attributes

1-10 of 28

OK Cancel

- To define which information (ID, Name, attribute values, or references) from the Source object needs to be retrieved, click on the Edit icon () in the Source field, and the Step Path Editor window will display (as shown below). The user also has the option to select specific data points from the Source object, or any of its related objects by selecting a combination of the elements available in the Keywords field and thereby creating a STEP Validation path. For more information on defining the Source STEP Path, see the **Mapping Validation Path Functionality** topic within this guide.

Step Path Editor [X]

Starting Points: PIES Item

Validation Path: [Empty Field] [Trash Icon]

Keywords: [attribute[id: '[id]']] [child[objecttype: '[id]']] [id] [name] [parent] [reference[type: '[id]']] [text['[text]']]

Information: [Empty Field]

[OK] [Cancel]

- Double-click on the required element(s) from the 'Keywords' field. The selected element will be populated in the 'Validation Path' field (as shown below). The specific combination of elements are chosen depending on the unique requirements for the data being retrieved. Selecting the elements will define the Source STEP Path which describes the path from which the source data must be retrieved. Depending on the element selected, further configurations may be required. In the example below, element **attribute[id: '[id]']** is selected and displays in the Validation Path field. The red colored field indicates that additional configuration is required.

Step Path Editor [X]

Starting Points: PIES Item

Validation Path: [attribute[id: '[id]']] [Trash Icon]

Keywords: [Empty Field]

Information: Defines a valid Attribute from where a value may be retrieved or set.

[OK] [Cancel]

Users can add one element or a combination of elements to create a Source STEP Path. For more information on creating the Source STEP Path, see the **Mapping Validation Path Functionality** topic within this guide.

In this example, the selected element 'attribute[id: '[id]']' is configured to retrieve the value from attribute with ID 'AC_PIES_DESCDES'.

Step Path Editor

Starting Points PIES Item

Validation Path

Keywords

Information

Once selected, the Source STEP Path with the defined data will be populated in the Source field.

Mapping Guide

Mapping

	ID	Name	Validation Type
Source	→	attribute[id:'AC_PIES_DESCDES']	
Target			
Transformation			

Suppress

Targets Transformations

Search

Find targets

Filter Options

Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Attribute	NAPA_PHdb_1161	# Steering gear Box Mounting Holes	Text	NAPA Product Attributes

- With the 'Targets' tab selected, click in the Search field and start typing the initial letters of the attribute / reference name or ID of the target attribute / reference to which the value should be copied. Select the attribute / reference from the typeahead results displayed below the search bar.

Mapping Guide ✕

Mapping

	ID	Name	Validation Type
Source	attribute[id:'AC_PIES_DESCDES']		
Target			
Transformation			

↓

Targets Transformations Suppress

Search Filter Options

8

Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Attribute	NAPATextBlock	NAPA Text Block	Text	NAPA Product Attributes

1-1 of 1

✓ OK
✕ Cancel

Once selected, the attribute / reference will be populated in the Target field.

Mapping Guide

Mapping

	ID	Name	Validation Type	
Source	attribute[id:'AC_PIES_DESCDES']			
Target	NAPATextBlock	NAPA Text Block	Text	

Transformation

✓ Valid configuration Suppress

Targets Transformations

Search

Find targets Filter Options
 Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Attribute	NAPA_PHdb_1161	# Steering gear Box Mounting Holes	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_1205	# Steering Gear Box Turns Lock to Lock	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_3153485	E-Waste	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_1831	Hose Port Type	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_29378	Input Shaft Spline Count	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_3229512	Input Shaft Type	Text	NAPA Product Attributes

1-10 of 28

OK Cancel

Note: Only attributes / references that are valid for the object type defined in the mapping Setup tab will be displayed as options to select on the Targets tab.

After the Source STEP path and target attribute / reference are defined, the system evaluates the validity match between the Source STEP path and the target attribute / reference. A hyperlink text explaining the reason for the validity match / mismatch displays. Clicking on the hyperlink opens the 'Detailed Information' dialog which displays the validity match / mismatch information of the Object to Object mapping. Below is an example of a valid match between the Source STEP path and the target attribute displayed in the 'Detailed Information' dialog.

The screenshot shows the 'Mapping Guide' window with a mapping configuration table. A red box highlights the 'Valid configuration' status, with an arrow pointing to the 'Detailed Information' dialog. The dialog contains a table with the following data:

	Source	Target	Evaluation
ID	AC_PIES_DESCDES	NAPATextBlock	
Name	Description Long	NAPA Text Block	
Title	Description Long	NAPA Text Block	
Node Type	attribute	attribute	
Validation Type	Text	Text	
Unit			
LOV	No	No	
Mandatory	No	No	
Derived	No	No	
Dependent	No	No	
Description	No	No	
Inherited	Yes	Yes	
Language Dependent	No	No	
Health	Yes	Yes	Yes

Users can suppress the validation mismatch warning message by clicking on the 'Suppress' checkbox in the Mapping Guide window. Selecting the 'Suppress' checkbox *only* removes the data type mismatch warnings displayed on the Mapping Guide window and does not resolve the mismatch irregularities.

- With the 'Transformations' tab selected, click in the search field and start typing the initial letters of the transformation name or ID. This brings up a dropdown of typeahead search results listing the attribute

transformations available in the system. Select the relevant transformation from the list displayed below the search bar.

Note: The transformations must be created and defined in workbench System Setup prior to being available to be selected in the Mapping Guide.

The screenshot shows the 'Mapping Guide' interface. At the top, there is a 'Mapping' section with a table containing source and target information. Below this is a 'Transformation' section with a status indicator 'Valid configuration' and a 'Suppress' checkbox. A red arrow points to the 'Transformations' tab. Below the tabs is a search bar labeled 'Find transformations' with a red circle containing the number '9'. Underneath is a 'Select transformation' table with columns for ID, Title, Parent Node, and Type. A red arrow points to the row with ID 'AC-NAPA-fueltypes'. At the bottom right, there are 'OK' and 'Cancel' buttons.

ID	Name	Validation Type
Source	attribute[id:'AC_PIES_DESCDES']	
Target	NAPATextBlock	Text

Transformation

Valid configuration Suppress

Targets Transformations

Search

Find transformations

ID	Title	Parent Node	Type
AC-NAPA-bodyStyles	AC-NAPA-bodyStyles	Mapping transformations	Replace whole value using Lookup Table
AC-NAPA-fueltypes	AC-NAPA-fueltypes	Mapping transformations	Replace whole value using Lookup Table
AppendColor	Append Color	Transformations	3 transformations

1-5 of 5

OK Cancel

The selected transformation will be populated in the 'Transformation' field (as shown below).

Mapping Guide

Mapping

	ID	Name	Validation Type	
Source	attribute[id:'AC_PIES_DESCDES']			
Target	NAPATextBlock	NAPA Text Block	Text	
Transformation	AC-NAPA-fueltypes	AC-NAPA-fueltypes	Replace whole value using Lookup Table	

Valid configuration Suppress

Targets Transformations

Search

Find transformations

Select transformation

ID	Title	Parent Node	Type
AC-NAPA-bodyStyles	AC-NAPA-bodyStyles	Mapping transformations	Replace whole value using Lookup Table
AC-NAPA-fueltypes	AC-NAPA-fueltypes	Mapping transformations	Replace whole value using Lookup Table
AppendColor	Append Color	Transformations	3 transformations

1-5 of 5

OK Cancel

For information on configuring the attribute transformations, see the **Attribute Transformations** topic within the **System Setup** section of **STEP Online Help**. And for details on each of the available transformation options, see the **Transformations** topic in the **Resource Materials** online help.

The populated transformation can be deleted by clicking the delete icon () in the 'Transformation' field. Similarly the populated Target attribute / reference can also be deleted by clicking the delete icon () in the 'Target' field.

Mapping Guide

Mapping

	ID	Name	Validation Type	
Source	attribute[id:'AC_PIES_DESCDES']			
Target	NAPATextBlock	NAPA Text Block	Text	
Transformation	AC-NAPA-fueltypes	AC-NAPA-fueltypes	Replace whole value using Lookup Table	

✓ Valid configuration Suppress

Targets Transformations

Search

Find transformations

Select transformation

ID	Title	Parent Node	Type
AC-NAPA-bodyStyles	AC-NAPA-bodyStyles	Mapping transformations	Replace whole value using Lookup Table
AC-NAPA-fueltypes	AC-NAPA-fueltypes	Mapping transformations	Replace whole value using Lookup Table
AppendColor	Append Color	Transformations	3 transformations

1-5 of 5

OK

- Click **OK** to save and close the 'Mapping Guide' window and then click **Save** to save the changes. As shown in the screenshot below, the newly added object mapping row will display. To edit the object mapping, click on the row to open the Mapping Guide window in which edits can be made.

Note: Once the user adds a mapping and selects the Source Step Path and Target and clicks **OK** to close out of the Mapping Guide window, the **Save** button is enabled until the user clicks on **Save** button. If the user fails to click **Save**, than the mapping will be lost once user selects a new mapping configuration.

The screenshot shows the 'Object to Object Mapping' configuration page. The left sidebar contains a 'Tree' menu with 'Mappings' selected, and a sub-menu with 'PIES Item → NAPA Product' highlighted. The main content area has a title 'Object to Object Mapping' and a 'Name' field containing 'Create NAPA Products from PIES Items'. Below this is a 'Target Definition' section with 'Add object mapping', 'Delete', and 'Disable' buttons. A search box 'Find sources, targets and transformations' is present. A table lists the mapping configuration:

Source	Transformation	Target	Validation Type
<input checked="" type="checkbox"/> attribute[id:'AC_PIES_DESCDES']	AC-NAPA-fueltypes	NAPA Part Number	Text

At the bottom of the table are buttons for 'Save', 'Duplicate', 'Delete', and 'Reset'. A red arrow points to the checkbox in the first row of the table.

11. Repeat the above steps 5 to 10 to add more object mapping rows for the same plugin.

User can add any number of mapping rows in the mapping plugin. When there are multiple mapper rows available within the mapping plugin, the order of execution of each mapper row is based on the order in which it is listed within the mapping plugin.

The health of the mapping row is displayed next to each mapping row. Users can also add some additional information describing each of the mapper rows. The user has the flexibility to disable, delete, or rearrange the listing order of the mapper rows.

Mapping Validation Path Functionality

The Mapping Validation Path functionality in the STEP Path Editor dialog ensures that users are able to select valid Source and Target objects to define their mapping setup in Object to Object Mapping or Application Mapping configurations. The premise behind this concept is that when some level of valid Source STEP Path data (or valid Target STEP Path data) for objects is provided, the mapping is able to retrieve the ID / Name / attribute values / references of the object present at the defined path.

Based on the Source and the Target object type defined for the Mapper Configuration, the system always evaluates the Source STEP path (or Target STEP path) and suggests the valid attributes / reference types / object types from the system.

The Starting Points parameter available in the STEP Path Editor dialog displays the object type that the Validation Path will begin evaluation on in order to follow the path to retrieve information from.

This relationship data is captured within the Validation Path field based on the data model setup that defines the relationship between different object types.

Each of the elements available within the Keywords field in the STEP Path Editor dialog is explained in detail below. Note that all elements are not necessarily used in all requirements. The specific combinations of elements are chosen depending on how the data needs to be retrieved from the Source object.

- **attribute[id: '[id]]**: This element defines the valid attribute where a value may be retrieved from the Source object. Further configuration is required to define the attribute. Once the attribute[id: '[id]] keyword has been selected, double-click on the Validation Path field to select an attribute (only attributes valid to the Source object type are displayed) from the Select Attribute dialog.
- **child[objecttype: '[id]]**: This element allows the user to retrieve the data from the child of the child of the object that is displayed in the Starting Points parameter. Further configuration is required to define the object type and also what value of the child object (ID / Name / attribute value / references) is to be retrieved. This element should always be accompanied by some of these **id** / **name** / **attribute[id: '[id]]** / **reference[type: '[id]]** elements.
- **id**: This element retrieves the ID of the defined object.
- **name** : This element retrieves the name of the defined object.

- **parent**: This element retrieves the data from the parent of the defined object. Further configuration is required to define what value of the parent object (ID / Name / attribute value /references) is to be retrieved. This element can always be accompanied by **id** / **name** / **attribute[id:'[id]']** / **reference [type:'[id]']** elements.
- **reference[type:'[id]']**: This element retrieves the data from the referenced object of the defined object. Further configuration is required to define the reference type and is also required to define what value of the referenced object (ID / Name / attribute value /references) is to be retrieved. This element can always be accompanied by **id** / **name** / **attribute[id:'[id]']** / **reference[type:'[id]']** elements.
- **referenceBy[type:'[id]']**: This element retrieves the data from the source object when the defined object is the target object of the configured reference. Further configuration is required to define the reference type and is also required to define what value of the source object (ID / Name / attribute value /references) is to be retrieved. This element can always be accompanied by **id** / **name** / **attribute[id:'[id]']** / **reference[type:'[id]']** elements.
- **text['[text]']**: This element lets the user enter a static text of the user's choice.

For example, for the ACES Body Num Doors attribute (AC_ACESBodyNumDoors) stored at the child of the Source object (PIES Item), the Source STEP path is configured as:

```
child[objecttype:'AC_ACESApplication'].attribute[id:'AC_ACESBodyNumDoors']
```

This Source STEP Path is determined by:

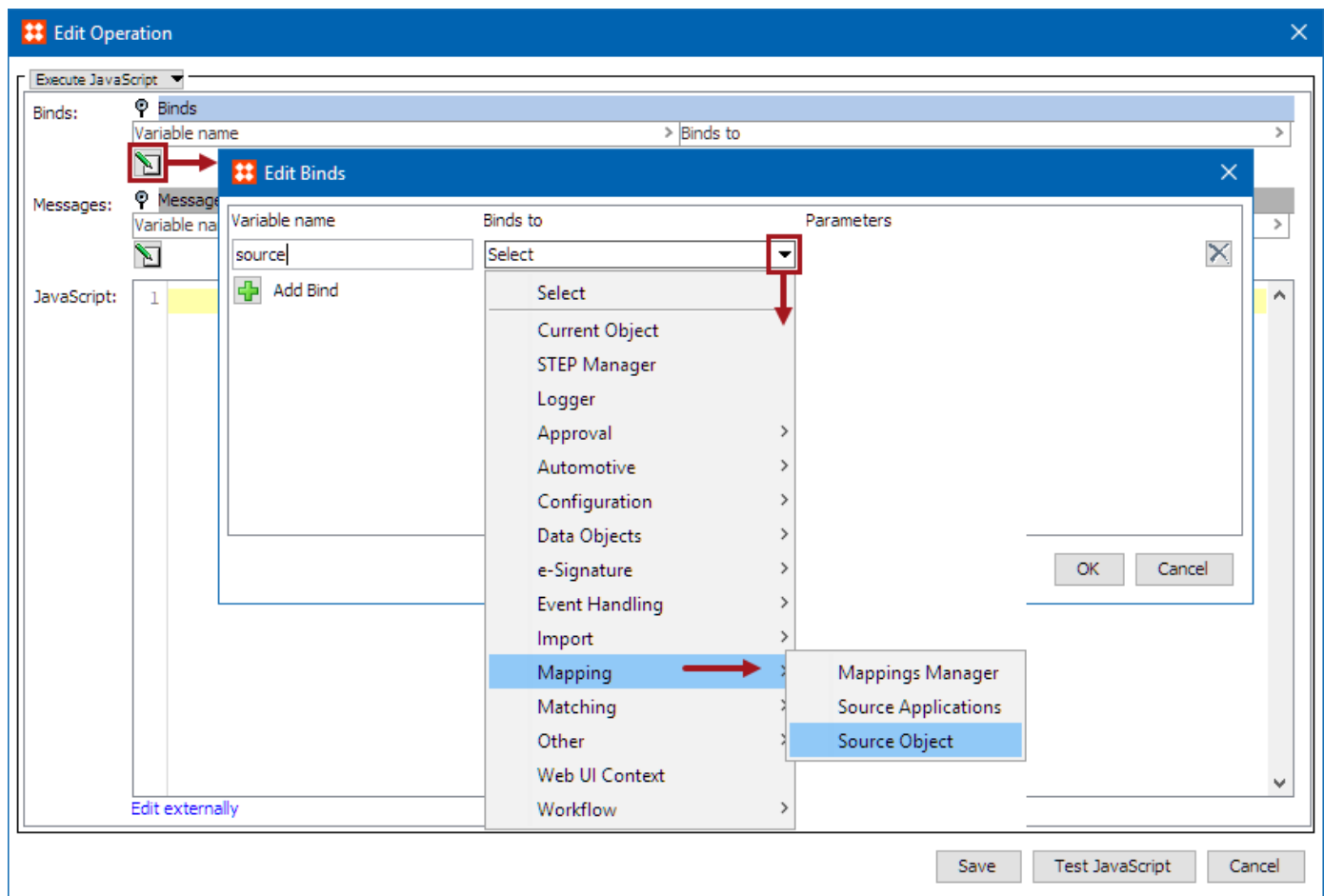
1. Identifying the source object.
2. Examining all children with object type AC_ACESApplication of the source object.
3. Evaluating the value of the ACES Body Num Doors attribute (AC_ACESBodyNumDoors) that is stored on the child object.

Business Action Mapping Plugin

This Mapping plugin gives extended flexibility for users to configure and solve unique mapping cases which cannot be achieved by the other Mapping plugins (Attribute Mapping, Application Mapping, and Object to Object Mapping). The Business Action Mapping plugin takes a business action as a configuration and when executed, executes the business action on the target object. This Mapping plugin is configured on the Onboarding Mappings Details Screen for the selected Mapper Configuration.

The data copied from a Source object is determined by the business action defined on the Select Business Action parameter within the Business Action Mapping plugin. This business action is manually created and configured independently in the workbench.

The business action can use the Source Object bind to gain access to the Source object. This bind is available within the 'Binds to' dropdown, as shown below.



Prerequisites

First create a business action to define the Source node using Execute JavaScript > Add Bind > Mapping > Source Object. The Source Object bind provides access to the Source node. For information on accessing source

object via the Source Object bind, see the **Source Object Bind** topic within the **Resource Materials** section of STEP Online Help.

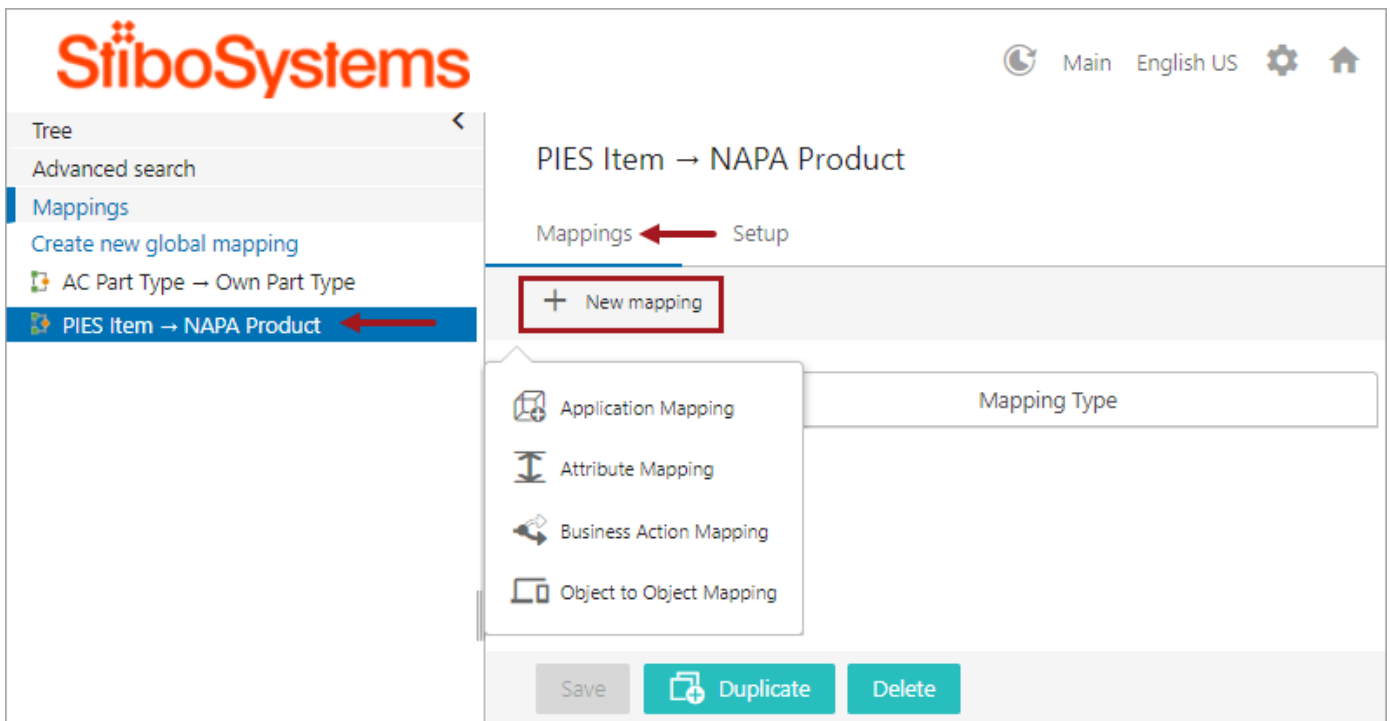
For more information on adding other binds, see **JavaScript Binds** in the **Resource Materials** online help.

Configuration

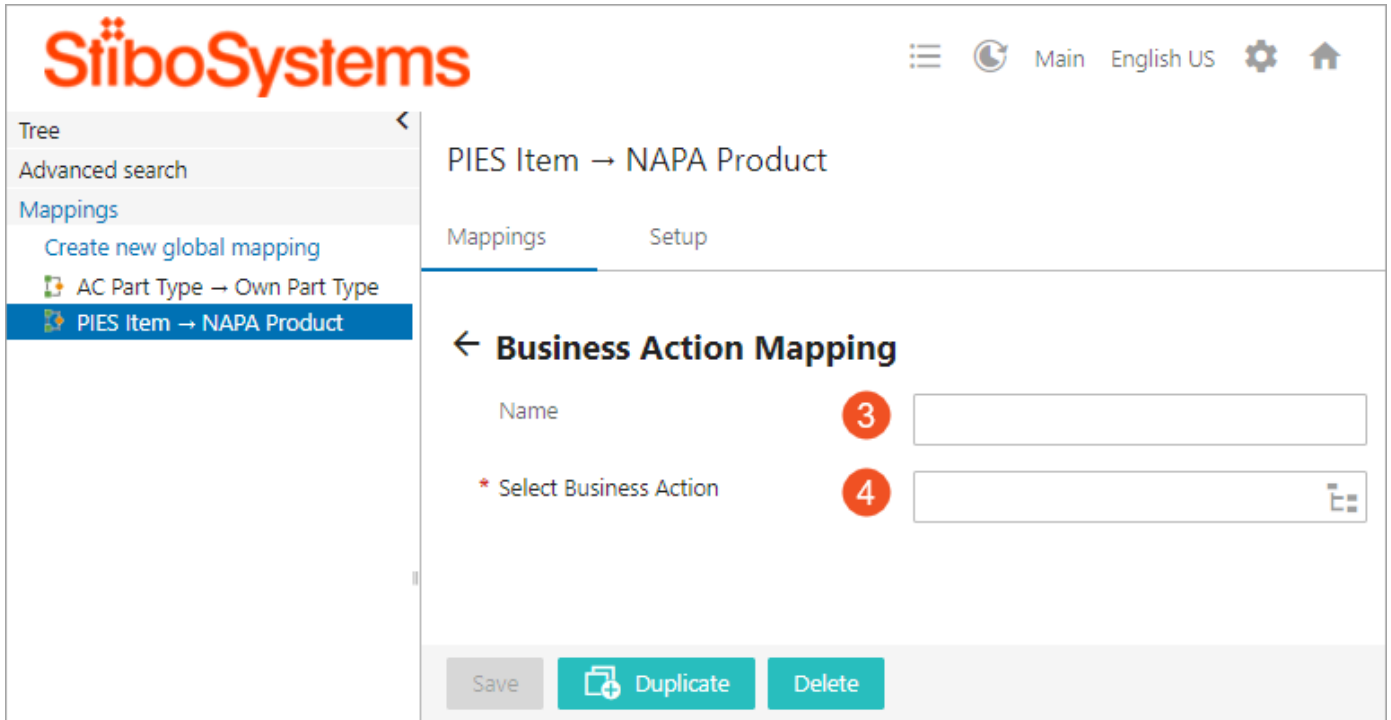
Only one business action can be configured in the mapping plugin. To execute multiple business actions, add multiple Business Action Mapping plugins to the Mapper Configuration setup entity.

To configure the Business Action Mapping plugin, follow these steps:

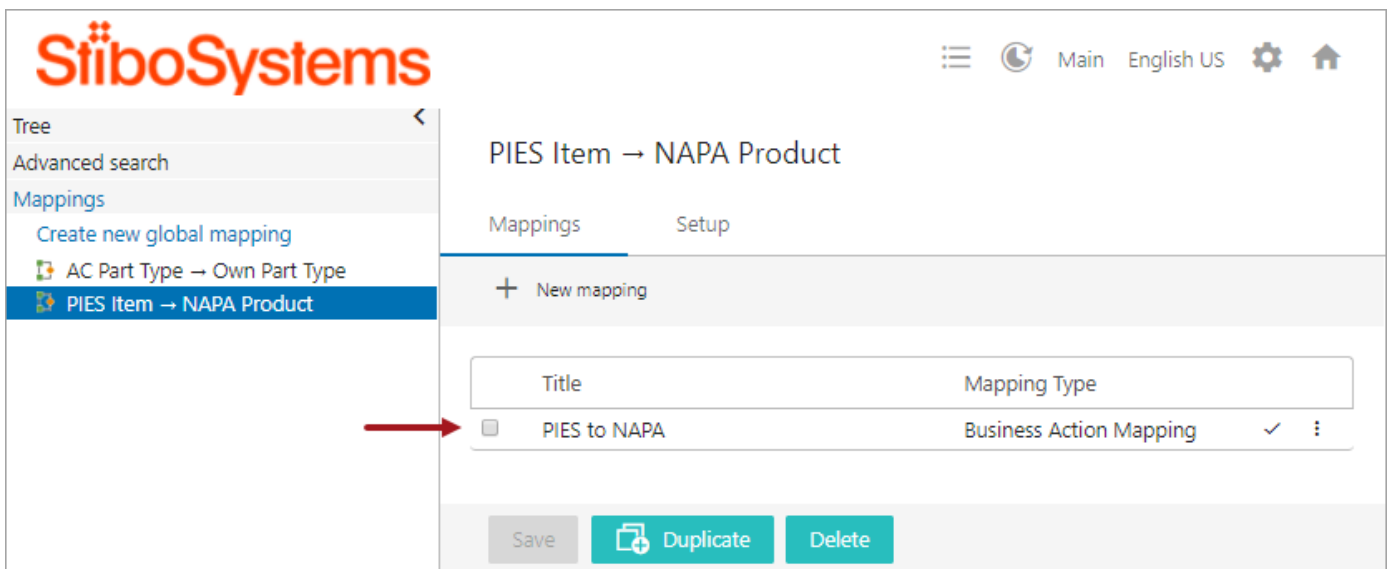
1. With the required Mapper Configuration selected, click on **New mapping** available within the Mappings tab of the Onboarding Mappings Details screen.



2. Select Business Action Mapping and a screen is displayed prompting the user to further configure the plugin (as shown below).



3. In the Name parameter, type in a suitable name. This should be a unique name that clearly describes the mapping functionality.
4. In the 'Select Business Action' parameter, type in the initial letters of the business action name or use the node picker to select and populate the desired business action. The 'Select Business Action' parameter allows the specified business action to run on the Source objects.
5. Click **Save** to save the changes. The newly added Business Action Mapping will be listed as shown below.



The health of the Business Action Mapping plugin is displayed next to each Mapping plugin. The user has the flexibility to disable, delete, or rearrange the listing order of the Mapping plugins. The check mark displayed in the image above indicates that the mapping was successful.

6. If multiple business actions are required, repeat the steps above to add multiple Business Action Mapping plugins to the Mapper Configuration setup entity.

Mapping Plugins Use Case Overview

To illustrate how the Mapping solution can address common challenges, the following use cases present a requirement with typical setup considerations. Understanding these use case examples will provide additional context to the configuration steps included in this guide, as well as the more granular use cases that concern different mapping solutions.

For more information on component-specific use cases, see the **Use Case Appendix** section of this guide.

Use Case Appendix

The following use cases address a number of common challenges users face when Onboarding / Outboarding data, and how they are solved by a PMDM for Automotive solution.

Attribute Mapping Use Cases

- Copy attribute values from one object to another object

Object to Object Mapping Use Cases

- Transfer data (ID / Name / Attribute values / References) from one object to another object
- Transfer data (ID / Name / Attribute values / References) from Source object to a new child object of the Target object
- Transfer data (ID / Name / Attribute values / References) of Source object to a referenced object of the Target object

Use Cases - Copy Attribute Values from One Object to Another Object

Because the Attribute Mapping plugin within Onboarding Mappings Details screen has many uses for how the mappings can be configured, describing the components within the mapper does not fully cover all of the functional use cases.. With that in mind, a use case is described below that outlines instances where this Attribute Mapping plugin has been successfully implemented to copy attribute values from one object to another object.

The use case for the **Attribute mapping** plugin are described below.

Example One

The requirement for this use case entails that the user has an attribute value in the Source object that needs to be copied over to the Target object. The user can do this either by using the Attribute Mapping plugin or by using the Object to Object Mapping plugin where the Source STEP path and the Target attribute is defined to hold the Source and the Target attribute value respectively. As the main intention of this topic is to address the various possible use cases of Attribute Mapping plugin, the Attribute Mapping method will be described below to achieve this requirement.

For this example, the Source object is PIES Item '034-VC21499' and the Target object is NAPA Product 'ATA64A' interlinked through the reference type 'PIESItemtoNAPAProduct' in the system.

Prerequisites

The validity of the Source and the Target attribute shall not have any mismatch.

In the Web UI, create and configure the Mapper Configuration and define the Source and Target objects and the Reference type to follow to determine the relationship between the Source and Target object. For more information on configuring and setup procedure about the Mapper Configuration, see topic **Configuring Mapper Configuration Setup Entity** within this guide.

PIES Item → NAPA Product

Mappings
Setup

Naming

* Map from

* Map to

Global Configuration

* Source

* Reference

* Target

Reverse Reference

Approved Workspace

Mapping description

Description

Save

 Duplicate

Delete
Reset

Configure an Initiate Business Action button within the Product Details screen of the Source object which shall execute the Mapper Configuration. For more information on executing the Mapper Configuration, see topic **Executing Mapper Configuration Setup Entity** within this guide.

1. Add a new Attribute Mapping plugin (or edit the one that exists) in the required Mapper Configuration.
2. Type in a suitable name next to Name field. In this example, the Name field is entered with the text 'Copy PIES Attribute to NAPA Product Attribute'

PIES Item → NAPA Product

Mappings Setup

← Attribute Mapping

Name

+ Add attribute mapping

Source	Source Validation Type	Transformation	Target	Target Validation Type
--------	------------------------	----------------	--------	------------------------

Save Duplicate Delete Reset

3. Click the 'Add attribute mapping' icon, and the Mapping Guide window will display (as shown below).

Mapping Guide

Mapping

ID	Name	Validation Type
Source		
Target		
Transformation		

Sources Targets Transformations

Search

Select Attribute

ID	Title	Validation Type	Attribute Group(S)
AC_PAdb_7966	12 Volt DC Outlet Quantity	Number	PAdb Attributes
AC_PAdb_9304	2nd Ring Face Design	Alpha Numeric	PAdb Attributes
AC_PAdb_9309	2nd Ring Material	Alpha Numeric	PAdb Attributes

Cancel OK

- With the Sources tab selected, click in the search field and start typing the initial letters of the attribute name or ID. This searches for the attribute of the Source object from where the attribute value could be retrieved. Select the attribute from the result list displayed below the search bar. In this example, the attribute 'Description Long' is used as source attribute.

Mapping Guide

Mapping

ID	Name	Validation Type
Source		
Target		
Transformation		

Sources Targets Transformations

Search

Description L

Select Attribute

ID	Title	Validation Type	Attribute Group(S)
AC_PIES_DESCDES	Description Long	Text	PIES Description Segment DESC

1-1 of 1

Cancel OK

5. Click **OK** and the selected attribute will be populated in the Source field (as shown below)

Mapping Guide

Mapping

ID	Name	Validation Type
Source	AC_PIES_DESCDES	Description Long
Target		
Transformation		

Sources Targets Transformations

Search

Description Lon

Select Attribute

ID	Title	Validation Type	Attribute Group(S)
AC_PIES_DESCDES	Description Long	Text	PIES Description Segment DESC

1-1 of 1

Cancel OK

- With the Targets tab selected, search and select the attribute of the Target object where the retrieved value from the Source object shall reside.

Mapping Guide

Mapping

	ID	Name	Validation Type
Source	AC_PIES_DESCDES	Description Long	Text
Target			
Transformation			≡+

Sources **Targets** Transformations

Search

Text

Select Attribute

ID	Title	Validation Type	Attribute Group(S)
NAPATextBlock	NAPA Text Block	Text	NAPA Product Attributes ←

⏪ ⏩ 1-1 of 1 ⏪ ⏩



Cancel
OK

Note: The Source object attribute and the Target object attribute should have a validity match.

Once selected, the selected attribute will be populated in the Target field.

Mapping Guide

Mapping

	ID	Name	Validation Type
Source	AC_PIES_DESCDES	Description Long	Text
Target	NAPATextBlock	NAPA Text Block	Text 
Transformation			

✓ Valid configuration

Sources **Targets** Transformations

Search

Select Attribute

ID	Title	Validation Type	Attribute Group(S)
NAPATextBlock	NAPA Text Block	Text	NAPA Product Attributes

1 1 of 1

- Click **OK** to save and close the Mapping Guide window. The newly added object mapping row will be listed as shown below.

Tree
Advanced search
Mappings
Create new global mapping
AC Part Type → Own Part Type
PIES Item → NAPA Product

PIES Item → NAPA Product

Mappings Setup

← **Attribute Mapping**

Name

+ Add attribute mapping

Source	Source Validation Type	Transformation	Target	Target Validation Type
<input type="checkbox"/> Description Long	Text		NAPA Text Block	Text ✓ +

Save Duplicate Delete Reset

- In the Web UI, standing on the Source object, execute the Mapper Configuration that holds the mapping plugin. For more information on executing the Mapper Configuration, see topic **Executing Mapper Configuration Setup Entity** within this guide.

Tree

- ▶ AutoCare Root
- ▼ Primary Product Hierarchy
 - ▶ Products
 - ▶ Discontinued Products
 - ▶ Product Overrides
 - ▶ Packaging
 - ▼ AutoCare Root
 - ▶ ACES Application Notes
 - ▶ PIES Interchange Products
 - ▼ PIES Products
 - ▼ 034MOTORSPORT
 - ▼ Belts and Cooling
 - ▼ Accessory Drive Belt S...
 - ▼ A/C Drive Belt Idler ...
 - ▶ **034-VC21499**
 - ▶ 034-VC36004
 - ▶ BCVC
 - ▶ Behr Hella Service
 - ▶ Behr Thermot-Tronik
 - ▶ Exide
 - ▶ Hella
 - ▶ Pierburg
 - ▶ Wako
 - ▶ TecDoc Supplier Root
 - ▼ NAPA Hierarchy

Advanced search


Mappings

Node Details

Product Details

ID: AC_PIESItem_GWWQ_034-VC21499

Name:

PIES Item To Product Image:  VC21499_P04

Object Type: PIES Item

PIES Item to NAPA Product:

ID	Title
NAPA_Product_ATA64A	ATA64A ✕

Approved: ✕ Never been approved.

ACES Mfr Body Code:




Mfr Label: ✎

Inside Diameter: ✎

Description Long: ✎

▶ Execute Mapper

After the user runs the mapping, the value (Check Engine Light Sensor) of the 'Description Long' attribute from the Source object is copied over to the Target object attribute 'NAPA Text Block.'

Tree	Node Details																					
<ul style="list-style-type: none"> ▶ AutoCare Root ▼ Primary Product Hierarchy <ul style="list-style-type: none"> ▶ Products ▶ Discontinued Products ▶ Product Overrides ▶ Packaging ▼ AutoCare Root <ul style="list-style-type: none"> ▶ ACES Application Notes ▶ PIES Interchange Products ▶ PIES Products ▶ TecDoc Supplier Root ▼ NAPA Hierarchy <ul style="list-style-type: none"> ▶ NAPA Comment Hierarchy ▶ NAPA Interchange Product Hi... ▼ NAPA Product Hierarchy <ul style="list-style-type: none"> ▼ ATA Fuel Injector - New <ul style="list-style-type: none"> <li style="background-color: #0056b3; color: white;">▶ ATA64A ▶ RPC 63858 ▶ Wiper Blade - Bosch Icon - ... ▶ TecDoc Supplier Price Lists ▶ Import Flow Root 	<h3>Product Details</h3> <table border="0"> <tr> <td>ID</td> <td>NAPA_Product_ATA64A</td> </tr> <tr> <td>Name</td> <td><input type="text" value="ATA64A"/></td> </tr> <tr> <td>Object Type</td> <td>NAPA Product</td> </tr> <tr> <td>Approved</td> <td>✗ Never been approved.</td> </tr> <tr> <td>Part Number To Image</td> <td> <div style="display: flex; align-items: center;"> +  </div> <p style="text-align: center;">VC21499_P04</p> </td> </tr> <tr> <td># Steering gear Box Mounting Holes</td> <td><input type="text"/></td> </tr> <tr> <td>Color</td> <td><input type="text"/></td> </tr> <tr> <td>NAPA Package INfo</td> <td><input type="text"/></td> </tr> <tr> <td>* NAPA Part Number</td> <td><input type="text"/></td> </tr> <tr> <td>NAPA Text Block</td> <td><input type="text" value="Check Engine Light Sensor"/></td> </tr> </table>		ID	NAPA_Product_ATA64A	Name	<input type="text" value="ATA64A"/>	Object Type	NAPA Product	Approved	✗ Never been approved.	Part Number To Image	<div style="display: flex; align-items: center;"> +  </div> <p style="text-align: center;">VC21499_P04</p>	# Steering gear Box Mounting Holes	<input type="text"/>	Color	<input type="text"/>	NAPA Package INfo	<input type="text"/>	* NAPA Part Number	<input type="text"/>	NAPA Text Block	<input type="text" value="Check Engine Light Sensor"/>
ID	NAPA_Product_ATA64A																					
Name	<input type="text" value="ATA64A"/>																					
Object Type	NAPA Product																					
Approved	✗ Never been approved.																					
Part Number To Image	<div style="display: flex; align-items: center;"> +  </div> <p style="text-align: center;">VC21499_P04</p>																					
# Steering gear Box Mounting Holes	<input type="text"/>																					
Color	<input type="text"/>																					
NAPA Package INfo	<input type="text"/>																					
* NAPA Part Number	<input type="text"/>																					
NAPA Text Block	<input type="text" value="Check Engine Light Sensor"/>																					
Advanced search																						
Mappings																						

Use Cases - Copy Data from One Object to Another Object

Because the Object to Object mapping plugin within the Onboarding Mappings Details screen has many uses for how the mappings can be configured, describing the components within the mapper does not fully cover all of the functional use cases. With that in mind, two full use cases are described below. They outline instances where this Object to Object Mapping plugin has been successfully implemented to copy data from one object to another object.

Example One

Consider an asset that is referenced as the Primary Product Image on the Source object (for example, PIES Item) and the user needs to also use the same asset on the Target object (for example, NAPA Product) as the Primary Product Image.

For this example, the Source object is a PIES Item '034-VC21499' and the Target object is a NAPA Product 'ATA64A' interlinked through the reference type 'PIESItemtoNAPAPProduct' in the system.

Following are the detailed steps describing how to configure the mapping to create the asset that is used as the Primary Product Image reference on a PIES Item to be used as the Primary Product Image reference on a NAPA Product, too.

Prerequisites

Define the asset object type as the Valid Target Type for the 'Image and Document Reference Type' that is used from the Target object (NAPA Product). In this example, the existing NAPA Product Image and Document Reference Type 'Part Number To Image' is defined to hold asset object type 'AC_PIESProductImage' as a Valid Target Type. For information about configuring the reference types, see the **Reference Types** topic within the **System Setup / Super User Guide** section of **STEP Online Help**.

The screenshot displays the 'System Setup' interface. On the left, a tree view shows the navigation structure under 'Reference Types', with 'Part Number To Image' selected. On the right, the 'Part Number To Image - Validity' configuration page is shown. It features a 'Reference Type' dropdown set to 'validity' and a 'Log' button. Below this, there are two sections: 'Valid Source Types' and 'Valid Target Types', each containing a table with 'ID' and 'Name' columns. In the 'Valid Source Types' table, 'NAPA_Product' is selected. In the 'Valid Target Types' table, 'AC_PIESProductImage' is selected and highlighted with a red border.

Valid Source Types	
ID	Name
NAPA_Product	NAPA Product
Modify Source Types	

Valid Target Types	
ID	Name
NAPA_DigitalAsset	NAPA Digital Asset
AC_PIESProductImage	PIES Product Image
Modify Target Types	

In the Web UI, create and configure the Mapper Configuration and define the Source and Target object types as well as the Reference type to follow to determine the relationship between the Source and Target object. For more information on configuring and setup procedure about the Mapper Configuration, see topic **Configuring Mapper Configuration Setup Entity** within this guide.

PIES Item → NAPA Product

Mappings
Setup

Naming

* Map from

* Map to

Global Configuration

* Source

* Reference

* Target

Reverse Reference

Approved Workspace

Mapping description

Description

Save
 Duplicate
Delete
Reset

Configure an Initiate Business Action button within the Product Details screen of the Source object which shall execute the Mapper Configuration. For more information on executing the Mapper Configuration, see topic **Executing Mapper Configuration Setup Entity** within this guide.

1. Add a new Object to Object Mapping plugin (or edit the one that exists) in the required Mapper Configuration.
2. Type in a suitable name next to Name field. In this example, the Name field is entered with the text 'Copy Primary Product Image from PIES to NAPA Product.'

PIES Item → NAPA Product

Mappings Setup

← Object to Object Mapping

Name

▼ Target Definition

Mapping type	<input type="text" value="On Target"/>
Object Type	<input type="text"/>
Reference Type	<input type="text"/>
Parent Node	<input type="text"/>

+ Add object mapping

3. Expand the Target Definition flipper, select 'On Target' option in the dropdown next to 'Mapping type' field.
4. Click the 'Add object mapping' icon, and the Mapping Guide window will display.

Mapping Guide

Mapping

ID	Name	Validation Type
Source		
Target		
Transformation		

Targets Transformations

Search Filter Options

Find targets Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Attribute	NAPA_PHdb_1161	# Steering gear Box Mounting Holes	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_1205	# Steering Gear Box Turns Lock to Lock	Text	NAPA Product Attributes
Attribute	ChangedObject	Changed Object	Text	Metadata

Cancel OK

- Click on the Edit icon (✎) in the Source field, and the Step Path Editor window will display (as shown below). For more information on defining the Source STEP path, see topic **Mapping Validation Path Functionality** within this guide.

Step Path Editor

Starting Points PIES Item

Validation Path

Keywords

Information

- First, double click to add the **reference[type:'[id]']** element from the Keywords field. Also, the ID of the asset must be configured from the Source to use in the Target reference, so double click to add the **id** element from

the Keywords field (do not use the name of the asset). The selected elements will be populated in the Validation Path field (as shown below). The red colored field depicts that further configuration is required.

Step Path Editor

Starting Points: PIES Item

Validation Path: reference[type:'[id]'] · id 🗑️

Keywords:

Information: Retrieves the ID of the previous element.

Cancel
OK

7. Click on the empty red colored **reference[type:'[id]']** element available within the Validation Path field and select the reference type that is referencing the Primary Product Image on the Source object. In this example, it is the reference type with id 'AC_PIESItemToProductImage.'

Once populated, the elements in the Validation path turn green in color as displayed below.

Step Path Editor

Starting Points: PIES Item

Validation Path: reference[type:'AC_PIESItemToProductImage'] · id 🗑️

Keywords:


Information: Retrieves the ID of the previous element.

Cancel
OK

8. Click **OK** and the Source STEP Path will be populated in the Source field.

Mapping Guide

Mapping

	ID	Name	Validation Type
Source		reference[type:'AC_PIESItemToProductImage'].id	
Target			
Transformation			

Targets Transformations

Search

Find targets

Filter Options

Attributes References

Select Attribute


Node Type	ID	Title	Validation Type	Attribute Group(S)
Attribute	NAPA_PHdb_1161	# Steering gear Box Mounting Holes	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_1205	# Steering Gear Box Turns Lock to Lock	Text	NAPA Product Attributes
Attribute	ChangedObject	Changed Object	Text	Metadata


Cancel **OK**

- With the Targets tab selected, search and select the 'Image and Document Reference Types' of the Target object. In this reference type, the Primary Product Image of the Source object will reside.

Mapping Guide

Mapping


	ID	Name	Validation Type
Source		reference[type:'AC_PIESItemToProductImage'].id	
Target			
Transformation			

 **Targets** Transformations

Search

Filter Options Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Reference	NAPA_PartNumberToImage	Part Number To Image	Reference	
Reference	NAPA_PartRelation	Part Relation	Reference	
Reference	NAPA_ProductToInterchangeProduct	Product To Interchange Product	Reference	

Note: The asset object type must be defined as the Valid Target Type for the Image Reference Type for both the Source and the Target object.

Once selected, the selected reference type will be populated in the Target field.

Mapping Guide

Mapping

	ID	Name	Validation Type	
Source	reference[type:'AC_PIESItemToProductImage'].id			
Target	NAPA_PartNumberToImage	Part Number To Image	Reference	

Transformation

✓ Valid configuration

Targets Transformations

Search

Filter Options
 Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Reference	NAPA_PartNumberToImage	Part Number To Image	Reference	

Cancel OK




- Click **OK** to close the Mapping Guide window and then click **Save** to save the changes. The newly added object mapping row will be listed.

The screenshot shows the SiiboSystems web interface. On the left is a navigation tree with 'Mappings' expanded, listing several mappings, with 'PIES Item → NAPA Product' selected. The main area is titled 'PIES Item → NAPA Product' and has tabs for 'Mappings' and 'Setup'. The 'Object to Object Mapping' section is active, showing a name field with the text 'Copy Primary Product Image from PIES to NAPA Product'. Below this is a 'Target Definition' section with an 'Add object mapping' button and a search box labeled 'Find sources, targets and transformations'. A table below the search box lists the mapping configuration:

Source	Transformation	Target	Validation Type
<input type="checkbox"/> reference[type:'AC_PIESItemToProductImage'].id	Part Number To Image	Reference	✓ + ⋮

At the bottom of the configuration area are buttons for 'Save', 'Duplicate', 'Delete', and 'Reset'. A red arrow points to the checkbox in the first row of the table.

11. In the Web UI, standing on the Source object, execute the Mapper Configuration that holds the mapping plugin. For more information on executing the Mapper Configuration, see topic **Executing Mapper Configuration Setup Entity** within this guide.

<p>Tree</p> <ul style="list-style-type: none"> ▶ AutoCare Root ▼ Primary Product Hierarchy <ul style="list-style-type: none"> Products Discontinued Products Product Overrides Packaging ▶ AutoCare Root ▶ TecDoc Supplier Root ▼ NAPA Hierarchy <ul style="list-style-type: none"> ▶ NAPA Comment Hierarchy ▶ NAPA Interchange Product Hi... ▼ NAPA Product Hierarchy <ul style="list-style-type: none"> ▼ ATA Fuel Injector - New <ul style="list-style-type: none"> ▶ ATA64A ▶ RPC 63858 ▶ Wiper Blade - Bosch Icon - ... ▶ TecDoc Supplier Price Lists ▶ Import Flow Root <p>Advanced search</p> <p>Mappings</p>	<p>Node Details</p> <p>Product Details</p> <hr/> <table> <tr> <td>ID</td> <td>NAPA_Product_ATA64A</td> </tr> <tr> <td>Name</td> <td><input type="text" value="ATA64A"/></td> </tr> <tr> <td>Object Type</td> <td>NAPA Product</td> </tr> <tr> <td>Approved</td> <td>✗ Never been approved.</td> </tr> <tr> <td>Part Number To Image</td> <td> <div style="display: flex; align-items: center;"> +  </div> <p>VC21499_P04</p> </td> </tr> </table>	ID	NAPA_Product_ATA64A	Name	<input type="text" value="ATA64A"/>	Object Type	NAPA Product	Approved	✗ Never been approved.	Part Number To Image	<div style="display: flex; align-items: center;"> +  </div> <p>VC21499_P04</p>
ID	NAPA_Product_ATA64A										
Name	<input type="text" value="ATA64A"/>										
Object Type	NAPA Product										
Approved	✗ Never been approved.										
Part Number To Image	<div style="display: flex; align-items: center;"> +  </div> <p>VC21499_P04</p>										

Example Two

An additional requirement for this use case entails that the user has an attribute value in the Source object that needs to be copied over to the Target object. The user can do this either by using the Attribute Mapping plugin or the Object to Object Mapping plugin. As the main intention of this topic is to address the various possible use cases of Object to Object Mapping plugin, the latter method will be used to achieve this requirement. Also, for the ease of understanding and to skip repeating some basic steps, the same Object to Object Mapping plugin detailed above (Copy Primary Product Image from PIES to NAPA Product) shall be used to achieve this requirement.

Prerequisites

The validity of the Source and the Target attribute shall not have any mismatch.

Configure an Initiate Business Action button within the Product Details screen of the Source object which shall execute the Mapper Configuration. For more information on executing the Mapper Configuration, see topic **Executing Mapper Configuration Setup Entity** within this guide.

Within the Object to Object Mapping plugin, select the 'On Target' option in the 'Mapping type' field under the Target Definition flipper.

PIES Item → NAPA Product

Mappings Setup

← Object to Object Mapping

Name

▼ Target Definition

Mapping type	<input type="text" value="On Target"/>
Object Type	<input type="text"/>
Reference Type	<input type="text"/>
Parent Node	<input type="text"/>

+ Add object mapping

1. Click the 'Add object mapping' icon, and the Mapping Guide window will display.

Mapping Guide

Mapping

ID	Name	Validation Type
Source		
Target		
Transformation		

Targets Transformations

Search Filter Options

Find targets Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Attribute	NAPA_PHdb_1161	# Steering gear Box Mounting Holes	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_1205	# Steering Gear Box Turns Lock to Lock	Text	NAPA Product Attributes
Attribute	ChangedObject	Changed Object	Text	Metadata

Cancel OK

- Click on the Edit icon (✎) in the Source field, and the Step Path Editor window will display. For more information on defining the Source STEP path, see topic **Mapping Validation Path Functionality** within this guide.

Step Path Editor

Starting Points PIES Item

Validation Path

Keywords attribute[id:'[id]'] child[objecttype:'[id]'] id name parent reference[type:'[id]'] text['[text]']

Information

- Double click on add the **attribute[id:'[id]']** element from the Keywords field. The selected element will be populated in the Validation Path field (as shown below). The red colored field depicts that further configuration

is required.

Step Path Editor

Starting Points PIES Item

Validation Path attribute[id:'[id]']

Keywords

Information Defines a valid Attribute from where a value may be retrieved or set.

Cancel
OK

- Click on the empty red colored **attribute[id:'[id]']** element available within the Validation Path field and select the Source attribute from which the value will be copied over to the Target object. In this example, the attribute with id 'AC_PIES_DESCDES' is selected.

Once populated, the elements in the Validation path turn green in color as displayed below.

Step Path Editor

Starting Points PIES Item

Validation Path attribute[id:'AC_PIES_DESCDES']

Keywords


Information Defines a valid Attribute from where a value may be retrieved or set.

Cancel
OK

- Click **OK** and the selected element with the defined data will be populated in the Source field.

Mapping Guide

Mapping

	ID	Name	Validation Type
Source		attribute[id:'AC_PIES_DESCDES']	
Target			
Transformation			

Targets Transformations

Search

Filter Options

Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Attribute	NAPA_PHdb_1161	# Steering gear Box Mounting Holes	Text	NAPA Product Attributes
Attribute	NAPA_PHdb_1205	# Steering Gear Box Turns Lock to Lock	Text	NAPA Product Attributes
Attribute	ChangedObject	Changed Object	Text	Metadata

Cancel OK

- With the Targets tab selected, search and select the attribute of the Target object where the retrieved value from the Source object shall reside.

Mapping Guide

Mapping

	ID	Name	Validation Type
Source		attribute[id:'AC_PIES_DESCDES']	
Target			
Transformation			

Targets Transformations

Search

Filter Options
 Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Attribute	NAPATextBlock	NAPA Text Block	Text	NAPA Product Attributes

Cancel OK

Note: The Source object attribute and the Target object attribute should have the validity match.

Once selected, the selected attribute will be populated in the Target field (as shown below).

Mapping Guide

Mapping

	ID	Name	Validation Type
Source	attribute[id:'AC_PIES_DESCDES']		
Target	NAPATextBlock	NAPA Text Block	Text
Transformation			

✓ Valid configuration

Targets Transformations

Search

Filter Options Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Attribute	NAPATextBlock	NAPA Text Block	Text	NAPA Product Attributes

Navigation: ⏪ ⏩

Buttons: Cancel OK

7. Click **OK** to close the Mapping Guide window and then click **Save** to save the changes. The newly added object mapping row will be listed.

The screenshot shows the StiboSystems web interface for configuring an object-to-object mapping. The left sidebar contains a 'Tree' view with 'Mappings' expanded, showing a list of mappings including 'PIES Item → NAPA Product' which is selected. The main content area is titled 'PIES Item → NAPA Product' and has tabs for 'Mappings' and 'Setup'. The 'Mappings' tab is active, displaying the 'Object to Object Mapping' configuration. The 'Name' field contains 'Copy Primary Product Image from PIES to NAPA Product'. Below this is a 'Target Definition' section with an 'Add object mapping' button and a search box labeled 'Find sources, targets and transformations'. A table lists the mapping configuration:

Source	Transformation	Target	Validation Type
<input type="checkbox"/> reference[type:'AC_PIESItemToProductImage'].id		Part Number To Image	Reference ✓ + ⋮
<input type="checkbox"/> attribute[id:'AC_PIES_DESCDES']		NAPA Text Block	Text ✓ + ⋮

A red arrow points to the second row of the table. At the bottom of the configuration area are buttons for 'Save', 'Duplicate', 'Delete', and 'Reset'.

- In the Web UI, standing on the Source object, execute the Mapper Configuration that holds the mapping plugin. For more information on executing the Mapper Configuration, see topic **Executing Mapper Configuration Setup Entity** within this guide.

Tree

- ▶ AutoCare Root
- ▼ Primary Product Hierarchy
 - ▶ Products
 - ▶ Discontinued Products
 - ▶ Product Overrides
 - ▶ Packaging
 - ▼ AutoCare Root
 - ▶ ACES Application Notes
 - ▶ PIES Interchange Products
 - ▼ PIES Products
 - ▼ 034MOTORSPORT
 - ▼ Belts and Cooling
 - ▼ Accessory Drive Belt S...
 - ▼ A/C Drive Belt Idler ...
 - ▶ **034-VC21499**
 - ▶ 034-VC36004
 - ▶ BCVC
 - ▶ Behr Hella Service
 - ▶ Behr Thermot-Tronik
 - ▶ Exide
 - ▶ Hella
 - ▶ Pierburg
 - ▶ Wako
 - ▶ TecDoc Supplier Root
 - ▼ NAPA Hierarchy

Advanced search


Mappings

Node Details

Product Details

ID: AC_PIESItem_GWWQ_034-VC21499

Name:

PIES Item To Product Image + 
VC21499_P04

Object Type: PIES Item

PIES Item to NAPA Product

ID	Title
NAPA_Product_ATA64A	ATA64A ✕

Approved: ✕ Never been approved.

ACES Mfr Body Code:




Mfr Label:

Inside Diameter:

Description Long:

▶ Execute Mapper

After the user runs the mapping, the value (Check Engine Light Sensor) of the Description Long attribute from Source object is copied over to the Target object attribute NAPA Text Block.

Tree	Node Details																					
<ul style="list-style-type: none"> ▶ AutoCare Root ▼ Primary Product Hierarchy <ul style="list-style-type: none"> ▶ Products ▶ Discontinued Products ▶ Product Overrides ▶ Packaging ▼ AutoCare Root <ul style="list-style-type: none"> ▶ ACES Application Notes ▶ PIES Interchange Products ▶ PIES Products ▶ TecDoc Supplier Root ▼ NAPA Hierarchy <ul style="list-style-type: none"> ▶ NAPA Comment Hierarchy ▶ NAPA Interchange Product Hi... ▼ NAPA Product Hierarchy <ul style="list-style-type: none"> ▼ ATA Fuel Injector - New <ul style="list-style-type: none"> <li style="background-color: #0056b3; color: white;">▶ ATA64A ▶ RPC 63858 ▶ Wiper Blade - Bosch Icon - ... ▶ TecDoc Supplier Price Lists ▶ Import Flow Root 	<h3>Product Details</h3> <table border="0"> <tr> <td>ID</td> <td>NAPA_Product_ATA64A</td> </tr> <tr> <td>Name</td> <td><input type="text" value="ATA64A"/></td> </tr> <tr> <td>Object Type</td> <td>NAPA Product</td> </tr> <tr> <td>Approved</td> <td>✗ Never been approved.</td> </tr> <tr> <td>Part Number To Image</td> <td> <div style="display: flex; align-items: center;"> +  </div> <p style="text-align: center;">VC21499_P04</p> </td> </tr> <tr> <td># Steering gear Box Mounting Holes</td> <td><input type="text"/></td> </tr> <tr> <td>Color</td> <td><input type="text"/></td> </tr> <tr> <td>NAPA Package INfo</td> <td><input type="text"/></td> </tr> <tr> <td>* NAPA Part Number</td> <td><input type="text"/></td> </tr> <tr> <td>NAPA Text Block</td> <td><input type="text" value="Check Engine Light Sensor"/></td> </tr> </table>		ID	NAPA_Product_ATA64A	Name	<input type="text" value="ATA64A"/>	Object Type	NAPA Product	Approved	✗ Never been approved.	Part Number To Image	<div style="display: flex; align-items: center;"> +  </div> <p style="text-align: center;">VC21499_P04</p>	# Steering gear Box Mounting Holes	<input type="text"/>	Color	<input type="text"/>	NAPA Package INfo	<input type="text"/>	* NAPA Part Number	<input type="text"/>	NAPA Text Block	<input type="text" value="Check Engine Light Sensor"/>
ID	NAPA_Product_ATA64A																					
Name	<input type="text" value="ATA64A"/>																					
Object Type	NAPA Product																					
Approved	✗ Never been approved.																					
Part Number To Image	<div style="display: flex; align-items: center;"> +  </div> <p style="text-align: center;">VC21499_P04</p>																					
# Steering gear Box Mounting Holes	<input type="text"/>																					
Color	<input type="text"/>																					
NAPA Package INfo	<input type="text"/>																					
* NAPA Part Number	<input type="text"/>																					
NAPA Text Block	<input type="text" value="Check Engine Light Sensor"/>																					
Advanced search																						
Mappings																						

Use Cases - Copy Data from One Object to Child Object of Another Object

Because the Object to Object mapping plugin within Onboarding Mappings Details screen has many uses for how the mappings can be configured, describing the components within the mapper does not fully cover all of the functional use cases. With that in mind, two full use cases are described below that outline instances where this Object to Object Mapping plugin has been successfully implemented to copy data from one object to child object of another object.

Example One

For the ease of understanding in this topic, data is to be copied from AutoCare standard to our Own standard. The AutoCare Vehicle hierarchy creates multiple objects that are split up for Sub Models and Years below the Model object. In this example, the Model object in the AutoCare Vehicle hierarchy will be used to illustrate how you are able to create your Own Vehicle hierarchy that combines the Sub Model and Year into one object and store the Sub Model and Year as attribute values on that object. And, consideration has been given to use the Engine Base value from the AutoCare hierarchy in the ID and Name of the newly created objects in the Own Vehicle hierarchy.

In this example, the AutoCare Audi 100 model is used as the Source object and take all Sub Models and Years below 100 model to create as new objects below the Own object (example Target Vehicle 100 hierarchy). Below is the image displaying AutoCare hierarchy of Source object (Model).

The screenshot displays the 'Tree' view on the left and a detailed object view on the right. The tree view shows a hierarchy starting from 'Assets' down to 'Vehicles' and then 'Light Duty - Car, Van and Truck'. Under 'Vehicles', there is an 'Audi' folder containing a '100' folder. The '100' folder contains several sub-objects: '1989 Audi 100', '1990 Audi 100', '1991 Audi 100', '1992 Audi 100', and '1993 Audi 100'. The '1989 Audi 100' folder is expanded to show its sub-objects: '1989 Audi 100, Base, Canada', '1989 Audi 100, Base, United States', '1989 Audi 100, E, Canada', and '1989 Audi 100, E, United States'. Red arrows point from the '100' folder to the label 'Model', from the '1989 Audi 100' folder to 'Base Vehicle (Year)', and from the '1989 Audi 100, Base, Canada' folder to 'Vehicle (Submodel)'. The detailed view on the right shows the object '1989 Audi 100, Base, Canada rev.0.1 - Clas' with various attributes and values.

Classification		Sub Products	References	Referenced By
Description				
Name	>	>	Value	
ID	>	>	AC_Vehicle_52749	
Name	>	>	1989 Audi 100, Base, Canada	
Object Type	>	>	Vehicle	
Revision	>	>	0.1 Last edited by VCDBIMPORT	
Approved	>	>	✘ Never Been Approved	
Translation	>	>	Not Translated	
Path	>	>	Classification 1 root/AutoCare Rc	
Visibility	>	>		
Changed Object	>	>	abc	
Delete Status	>	>	abc	
New Object	>	>	abc	
VCdb Publication Stage	>	>	4	
VCdb Publication Stage Date	>	>	2012-01-27 00:00:00	
VCdb Publication Stage Sourc	>	>	abc	
VCdb Region	>	>	Canada (CAN)	
VCdb Source	>	>	abc	
VCdb Submodel	>	>	Base	

Below is the image displaying the Own hierarchy of Target object (Target Vehicle hierarchy).

Following are the detailed steps describing to achieve the same via the mapping solution.

Prerequisites

Create a new Classification Reference Type and establish the reference between the AutoCare Audi model 100 and Own Target Vehicle 100 object. For information about creating reference types and establishing references, see the **Reference Types** topic within the **System Setup / Super User Guide** section of **STEP Online Help**.

In the Web UI, create and configure the Mapper Configuration and define the Source and Target objects and the Reference type to follow to determine the relationship between the Source and Target object. For more information on configuring and setup procedure for the Mapper Configuration, see topic **Configuring Mapper Configuration Setup Entity** within this guide.

AutoCare Vehicle Model → Own Vehicle Model


Mappings Setup


Naming


* Map from: AutoCare Vehicle Model

* Map to: Own Vehicle Model

Global Configuration

* Source: Model (AC_Model) 

* Reference: AC Model To Target Vehicles (ACModelToTargetVehicles) 

* Target: Target Vehicles (TargetVehicles) 

Reverse Reference:

Approved Workspace:

Mapping description

Description:

Save Duplicate Delete Reset

Configure an Initiate Business Action button within the Product Details screen of the Source object which shall execute the Mapper Configuration. For more information on executing the Mapper Configuration, see topic **Executing Mapper Configuration Setup Entity** within this guide.

1. Add a new Object to Object Mapping plugin (or edit the one that exists) in the required Mapper Configuration.
2. Type in a suitable name next to Name field. In this example, the Name field is entered with the text 'Create Target Vehicles from AutoCare Model.'

AutoCare Vehicle Model → Own Vehicle Model

Mappings
Setup

← Object to Object Mapping

Name

Create Target Vehicles from AutoCare Model

▼ Target Definition

Mapping type

Child ▼

Object Type

Target_Vehicle (Target_Vehicle) ⋮

Reference Type

⋮

Parent Node

⋮

+ Add object mapping

Find sources, targets and transformations

Save

⋮ Duplicate


Delete

Reset

3. Expand the Target Definition flipper, select 'Child' option in the dropdown next to 'Mapping type' field.
4. Choose the Object Type that should be used for the new objects that are getting created. In this example, object type with id 'Target_Vehicle' is selected.
5. Click the 'Add object mapping' icon, and the Mapping Guide window will display (as shown below).

Mapping Guide

Mapping

ID	Name	Validation Type
Source		
Target		
Transformation		

Targets Transformations


Search

Filter Options
 Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
1-1 of 0				

Cancel OK

- Click on the Edit icon () in the Source field, and the Step Path Editor window will display. For more information on defining the Source STEP path, see topic **Mapping Validation Path Functionality** within this guide.

Step Path Editor

Starting Points PIES Item

Validation Path

Keywords attribute[id:'[id]'] child[objecttype:'[id]'] id name parent reference[type:'[id]'] text['[text]']

Information

OK Cancel

- For this use case, since there are multiple functions required to be executed, the following logic is used to accomplish the requirement through the STEP path.
 - To define what to use for the ID of the new objects that will get created.

In this example, the new objects that are going to get created are to have a prefix of 'Target_' in the ID. So, using the 'text' element in the STEP path > enter the static text of 'Target_' > select Use in ID checkbox.

AutoCare Vehicle Model → Own Vehicle Model



Mappings Setup

← Object to Object Mapping

Name

▶ Target Definition

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Transformation	Target	Validation Type
<input type="checkbox"/> text['Target_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			 + ⋮

- To define what to use for the Name of the new objects that are getting created, as well as what values to use in the ID to make it unique.

In this example, the AutoCare Sub Model is used in the ID and the Name of the new objects as well as storing the value as an attribute value in the new objects. So, using the **child[objecttype:'id']** and **attribute[id:'id']** elements in the STEP path will follow child objects of the Source object and retrieve the Sub Model that is stored as an attribute value on the Vehicle object. The 'Use in ID,' 'Use in Name,' and 'Hash' checkboxes are selected. (The 'Hash' checkbox is used if the combined values for the ID will be >40 characters since there is a maximum of 40 characters limit for IDs.) Also, the Target attribute of Sub Model is selected to store the value in.

AutoCare Vehicle Model → Own Vehicle Model

Mappings Setup

← **Object to Object Mapping**

Name

▶ Target Definition

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Transformation	Target	Validation Type
<input type="checkbox"/> text['Target_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> child.child.attribute[id:'AC_VCdbSubmodel']	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> ←		Sub Model	Text ✓ + ⋮

- The use case requires to use the Engine Base ID in the ID of the new objects.

Using the **child[objecttype:'[id]'], reference[type:'[id]'],** and **id** elements in the STEP path to follow child objects and references of the Source object, the Engine Base ID is retrieved to be used in the ID of the new objects. The 'Use in ID' and 'Hash' checkboxes are selected.

AutoCare Vehicle Model → Own Vehicle Model

Mappings Setup

← **Object to Object Mapping**

Name

▶ Target Definition

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Target	Validation Type
<input type="checkbox"/> text['Target_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		✓ + ⋮
<input type="checkbox"/> child.child.attribute[id:'AC_VCdbSubmodel']	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sub Model	Text ✓ + ⋮
<input type="checkbox"/> child.child.reference[type:'AC_VehicleToEngineConfig'].reference[type:'AC_EngineConfigToEngineBase'].id	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> ←		🔍 + ⋮

- The use case requires the use of the Engine Base Name in the Name of the new objects.

Using the **child[objecttype:'[id]'], reference[type:'[id]'],** and **name** elements in the STEP path to follow child objects and references of the Source object, the Engine Base Name is retrieved to be used as the Name of the new objects (as shown below).

AutoCare Vehicle Model → Own Vehicle Model

Mappings Setup

← Object to Object Mapping

Name

▼ Target Definition

Mapping type

Object Type

Reference Type

Parent Node

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Target	Validation Type
<input type="checkbox"/> text['Target_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		✓ + ⋮
<input type="checkbox"/> child.child.attribute[id:'AC_VCdbSubmodel']	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sub Model	Text ✓ + ⋮
<input type="checkbox"/> child.child.reference[type:'AC_VehicleToEngineConfig'].reference[type:'AC_EngineConfigToEngineBase'].id	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		🔍 + ⋮
<input type="checkbox"/> child.child.reference[type:'AC_VehicleToEngineConfig'].reference[type:'AC_EngineConfigToEngineBase'].name	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	←	🔍 + ⋮

- The use case requires the use of the Year attribute value and store it as an attribute value on the new objects.

Using the **child[objecttype:'[id]']** and **attribute[id:'[id]']** elements in the STEP path to follow the child object of the Source object, the AC_VCdbYear attribute value is retrieved for Year. Also, the Target attribute of Year is selected to store the value in (as shown below).

AutoCare Vehicle Model → Own Vehicle Model

Mappings Setup

← **Object to Object Mapping**

Name

▶ Target Definition

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Target	Validation Type
<input type="checkbox"/> text[Target_]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		✓ + ⋮
<input type="checkbox"/> child.child.attribute[id:'AC_VCdbSubmodel']	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sub Model	Text ✓ + ⋮
<input type="checkbox"/> child.child.reference[type:'AC_VehicleToEngineConfig'].reference[type:'AC_EngineConfigToEngineBase'].id	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		✓ + ⋮
<input type="checkbox"/> child.child.reference[type:'AC_VehicleToEngineConfig'].reference[type:'AC_EngineConfigToEngineBase'].name	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		✓ + ⋮
<input type="checkbox"/> child.attribute[id:'AC_VCdbYear'] ←	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Own_Year	Text ✓ + ⋮

8. Click **OK** to save and close the Mapping Guide window. The newly added object mapping row will be listed.

AutoCare Vehicle Model → Own Vehicle Model

Mappings Setup

← **Object to Object Mapping**

Name

▶ Target Definition

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Target	Validation Type
<input type="checkbox"/> text[Target_]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		✓ + ⋮
<input type="checkbox"/> child.child.attribute[id:'AC_VCdbSubmodel']	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sub Model	Text ✓ + ⋮
<input type="checkbox"/> child.child.reference[type:'AC_VehicleToEngineConfig'].reference[type:'AC_EngineConfigToEngineBase'].id	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		✓ + ⋮
<input type="checkbox"/> child.child.reference[type:'AC_VehicleToEngineConfig'].reference[type:'AC_EngineConfigToEngineBase'].name	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		✓ + ⋮
<input type="checkbox"/> child.attribute[id:'AC_VCdbYear']	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Own_Year	Text ✓ + ⋮

9. In the Web UI, standing on the Source object, execute the Mapper Configuration that holds the mapping plugin. For more information on executing the Mapper Configuration, see topic **Executing Mapper Configuration Setup Entity** within this guide.

- Asuna
- Auburn
- Audi
- 100
- 1989 Audi 100
 - 1989 Audi 100, Base, Canada
 - 1989 Audi 100, Base, United States
 - 1989 Audi 100, E, Canada
 - 1989 Audi 100, E, United States
- 1990 Audi 100
 - 1990 Audi 100, Base, Canada
 - 1990 Audi 100, Base, United States
- 1991 Audi 100
 - 1991 Audi 100, Base, Canada
 - 1991 Audi 100, Base, United States
- 1992 Audi 100
 - 1992 Audi 100, Base, Canada
 - 1992 Audi 100, Base, United States
 - 1992 Audi 100, CS, Canada
 - 1992 Audi 100, CS, United States
 - 1992 Audi 100, S, Canada
 - 1992 Audi 100, S, United States
- 1993 Audi 100
 - 1993 Audi 100, Base, Canada
 - 1993 Audi 100, Base, United States
 - 1993 Audi 100, CS, Canada
 - 1993 Audi 100, CS, United States
 - 1993 Audi 100, S, Canada
 - 1993 Audi 100, S, United States
- 1994 Audi 100
 - 1994 Audi 100, CS, Canada
 - 1994 Audi 100, CS, United States
 - 1994 Audi 100, S, Canada
 - 1994 Audi 100, S, United States
- 100 Quattro
- 100 Series
- 200

Classification Details

<< are Root > Vehicle Configuration (VCdb) > Vehicles > Light Duty - Car, Van and Truck > Audi > 100

Basic

Sub objects

References

ID: AC_Model_73_973

Name:

Object Type: Model

VCdb Vehicle Type:

AC Model To Target Vehicles

ID	Title
TargetVehicle100	Target Vehicle 100

arch

Execute Mapper

Save

Reset

Save&Approve

Delete

After the user runs the mapping, the following changes are implemented in the system.

- New objects are created as single child objects below the Target object that is defined in the Mapper Configuration Setup.
- The new objects are created with ID of Target_[Hash].
- The Name of the new object is the value of the Engine Base Name.

- The Object Type that is used for the new object is the one that is defined in the Object Type parameter in the Mapping Target Definition flipper.
- The Year values are stored in the Own_Year attribute on the new object.
- The Sub Model value is stored in the Sub Model attribute on the new object.

Tree

- ▶ AutoCare Root
- ▶ Primary Product Hierarchy
- ▶ TecDoc Supplier Price Lists
- ▶ Import Flow Root
- ▶ AutoCare Root
- ▼ Target Vehicle Model
 - ▼ Target Vehicle 100
 - 12.8L L6, 2771CC, -CID
 - 202.3L L5, 2309CC, -CID
 - 202.8L L6, 2771CC, -CID
 - 4292.8L L6, 2771CC, -CID
 - 702.3L L5, 2309CC, -CID**

Classification Details

Target Vehicle Model > Target Vehicle 100 > 702.3L L5, 2309CC, -CID

Basic Sub objects References

ID	Target_b14585811b739a42a2e7cfeff3804691
Name	702.3L L5, 2309CC, -CID
Object Type	Target_Vehicle
Own_Year	1989
Sub Model	70
	70

Example Two

The second requirement for this use case entails that the user has maintained part numbers (PIES Items) and applications (ACES) in the AutoCare Product hierarchy, but also want to create the PIES Items as part numbers in their Own product hierarchy and establish the reference from their Own part number to the PIES Item.

In this example, the AutoCare part type 'Vehicle Battery' is used as the Source object and all PIES Items below 'Vehicle Battery' are used to create new objects below the Own object (example 'Own Vehicle Battery' hierarchy).

Prerequisites

Create a new Product Reference Type to link the PIES Item parent object to the Target parent object and establish the reference between the PIES Item parent part type object to the Own parent part type object that user needs to create the new part numbers below. In this example, the Product Reference Type 'PIES Part Type to Target Part Type' is defined to create the link between PIES Item parent and Own part number parent object. Below is the image displaying hierarchy of Source object (Vehicle Battery), Target object (Own Vehicle Battery), and the reference linking them.

The screenshot displays the Stibo Systems interface. On the left is a 'Tree' view showing a product hierarchy. The 'Own Root' folder contains 'Own Vehicle Battery', which is highlighted with a red arrow and labeled 'Target object'. Above it, 'Battery and Related Components' contains 'Vehicle Battery', highlighted with a red arrow and labeled 'Source object'. The right pane is titled 'Vehicle Battery rev.0.3 - References' and shows a configuration for a reference type. A red box highlights the 'Reference Type' dropdown set to 'Target' and the 'PIES Part Type to Target Part Type' dropdown set to 'Own Vehicle Battery'. Below this, there are sections for 'Index Words', 'Publications', and 'Linked Attributes from Product Hierarchy'.

Also, create a new Product Reference Type to link the new Target part number and existing PIES Item. In this example, the Product Reference Type 'OWN Part To PIES Part' is defined to create link between Own part number and PIES Item object (as shown below). For information about creating reference types and establishing references, see the **Reference Types** topic within the **System Setup / Super User Guide** section of **STEP Online Help**.

The screenshot displays the 'System Setup' interface. On the left, a tree view shows the navigation path: **System Setup** > **Reference Types** > **Product Reference Types** > **OWN Part To PIES Part**. The main panel shows the configuration for this reference type.

OWN Part To PIES Part - Reference Type

Reference Type	Validity	Log
Description		
Name	>	Value >
ID	>	OWNPartToPIESPart
Name	>	OWN Part To PIES Part
Last edited by	>	2019-07-30 13:30:56.025 by USERB
Externally Maintained	>	No
Dimension Dependencies	>	
Completeness Score	>	
Allow multiple references	>	Yes
Mandatory	>	No
Inheritance	>	None
Attribute Help Text	>	abc
Automotive Validation Path	>	abc
Purpose	>	abc
In Attribute Groups		
ID	>	Name >
> Add Attribute Group		
Valid Attributes		
ID	>	Name >
> Add Attribute		

In the Web UI, create and configure the Mapper Configuration and define the Source and Target objects and the Reference type to follow to determine the relationship between the Source and Target object. For more information on configuring and setup procedure for the Mapper Configuration, see topic **Configuring Mapper Configuration Setup Entity** within this guide.

<p>Tree</p> <p>Advanced search</p> <p>Mappings</p> <p>Create new global mapping</p> <p>AC Part Type → Own Part Type</p> <p>Auto Care → Napa Item</p> <p>AutoCare Vehicle Model → Own Vehicle</p>	<h3>AC Part Type → Own Part Type</h3> <p>Setup</p> <p>Naming</p> <p>* Map from: AC Part Type</p> <p>* Map to: Own Part Type</p> <p>Global Configuration</p> <p>* Source: PIES PCdb Part Terminology (AC_PIESPCdbPartTerminology)</p> <p>* Reference: PIES Part Type to Target Part Type (PIESPartTypetoTargetPartType)</p> <p>* Target: Own Product Container (Own_ProductContainer)</p> <p>Reverse Reference: <input type="checkbox"/></p> <p>Approved Workspace: <input type="checkbox"/></p> <p>Mapping description</p> <p>Description: <input type="text"/></p>
---	--

Configure an Initiate Business Action button within the Product Details screen of the Source object which shall execute the Mapper Configuration. For more information on executing the Mapper Configuration, see topic **Executing Mapper Configuration Setup Entity** within this guide.

1. Add a new Object to Object Mapping plugin (or edit the one that exists) in the required Mapper Configuration.
2. Type in a suitable name next to Name field. In this example, the Name field is entered with the text 'Create AutoCare PIES Items for Own Parts.'

AC Part Type → Own Part Type

Mappings Setup

← Object to Object Mapping

Name

▼ Target Definition

Mapping type

Object Type

Reference Type

Parent Node

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Transformation	Target	Validation Type

Save Duplicate Delete Reset

- Expand the Target Definition flipper, select 'Child' option in the dropdown next to 'Mapping type' field.
- Choose the Object Type that should be used for the new objects that are getting created. In this example, object type with id 'Own_Partnumber' is selected.
- Click the 'Add object mapping' icon, and the Mapping Guide window will display.

Mapping Guide

Mapping

	ID	Name	Validation Type
Source			
Target			
Transformation			

Targets Transformations

Search Filter Options


Find targets Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Reference	OWNPartToPIESPart	OWN Part To PIES Part	Reference	

1-1 of 1

Cancel OK

- Click on the Edit icon () in the Source field, and the Step Path Editor window will display. For more information on defining the Source STEP path, see topic **Mapping Validation Path Functionality** within this guide.

Step Path Editor

Starting Points PIES Item

Validation Path

Keywords

Information

- For this use case, since there are multiple functions required to be executed, the following series of logic is used to accomplish the requirement through the STEP path.
 - To define what to use for the ID of the new objects that will get created.

In this example, the new objects that are going to get created need to have an ID pattern of Own_
[BrandAAIAID]_[PIESItemName]. To achieve prefix of 'Own_' in the ID, we are using the 'text' element in the STEP path > enter the static text of 'Own_' > select Use in ID checkbox (as shown below).

AC Part Type → Own Part Type



Mappings Setup

← **Object to Object Mapping**

Name

▶ Target Definition

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Transformation	Target	Validation Type
<input type="checkbox"/> text['Own_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			 + ⋮

- To get the BrandAAIAID value in the ID of the new objects that will get created.

In this example, the BrandAAIAID value is used in the ID of the new objects. So, using the **child [objecttype:'id']** and **attribute[id:'id']** elements in the STEP path will follow child objects of the Source object and retrieve the BrandAAIAID value that is stored as an attribute value on the PIES Item. The 'Use in ID' checkbox is selected (as shown below).

AC Part Type → Own Part Type

Mappings Setup


← **Object to Object Mapping**

Name

▶ Target Definition

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Transformation	Target	Validation Type
<input type="checkbox"/> text['Own_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> child[objecttype:'AC_PIESItem'].attribute[id:'AC_PIES_ITEMBrandAAIAID']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮



- To get the '_' value in the ID of the new objects that will get created.

In this example, the new objects that are going to get created have to have an ID pattern of Own_
[BrandAAIAID]_[PIESItemName]. To achieve prefix of '_' after the [BrandAAIAID] in the ID, we are using the 'text' element in the STEP path > enter the static text of '_' > select Use in ID checkbox (as shown below).

AC Part Type → Own Part Type

Mappings Setup


← **Object to Object Mapping**

Name

▶ Target Definition

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Transformation	Target	Validation Type
<input type="checkbox"/> text['Own_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> child[objecttype:'AC_PIESItem'].attribute[id:'AC_PIES_ITEMBrandAAIAID']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> text['_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮



- To define what to use for the Name of the new objects that are getting created, as well to get the PIES Item Name value in the ID of the new objects that will get created.

In this example, the PIES Item Name is used in the ID and the Name of the new objects. So, using the **child[objecttype:'[id]']** and **name** elements in the STEP path will follow child objects of the Source object and retrieve the PIES Item Name. The 'Use in ID' and 'Use in Name,' checkboxes are selected (as shown below).

AC Part Type → Own Part Type

Mappings Setup

← Object to Object Mapping

Name

▶ Target Definition

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Transformation	Target Validation	Type
<input type="checkbox"/> text['Own_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		✓	+ ⋮
<input type="checkbox"/> child[objecttype:'AC_PIESItem'].attribute[id:'AC_PIES_ITEMBrandAAIAID']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		✓	+ ⋮
<input type="checkbox"/> text['_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		✓	+ ⋮
<input type="checkbox"/> child[objecttype:'AC_PIESItem'].name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		🔍	+ ⋮

- The use case requires the establishment of the reference from the newly created Own part number to the PIES Item.

Using the **child[objecttype:'[id]']** and **id** elements in the STEP path to follow the child object of the Source object, the PIES Item is retrieved as the Source Object. Also the reference type 'OwnPartToPIESPart' is defined in the Target field define the Target reference type (as shown below).

AC Part Type → Own Part Type

Mappings Setup

← **Object to Object Mapping**

Name

▶ Target Definition

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Transformation	Target	Validation Type
<input type="checkbox"/> text['Own_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> child[objecttype:'AC_PIESItem'].attribute[id:'AC_PIES_ITEMBrandAAIAID']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> text['_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> child[objecttype:'AC_PIESItem'].name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> child[objecttype:'AC_PIESItem'].id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		OWN Part To PIES Part	Reference ✖ + ⋮

8. Click **OK** to save and close the Mapping Guide window. The newly added object mapping row will be listed.

AC Part Type → Own Part Type

Mappings Setup

← **Object to Object Mapping**

Name

▶ Target Definition

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Transformation	Target	Validation Type
<input type="checkbox"/> text['Own_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> child[objecttype:'AC_PIESItem'].attribute[id:'AC_PIES_ITEMBrandAAIAID']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> text['_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> child[objecttype:'AC_PIESItem'].name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>			✓ + ⋮
<input type="checkbox"/> child[objecttype:'AC_PIESItem'].id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		OWN Part To PIES Part	Reference ✖ + ⋮

9. In the Web UI, standing on the Source object, execute the Mapper Configuration that holds the mapping plugin. For more information on executing the Mapper Configuration, see topic **Executing Mapper**

Configuration Setup Entity within this guide.

The screenshot displays the Stibo Systems configuration interface. On the left is a 'Tree' view showing a hierarchical structure of product categories. The 'Vehicle Battery' category is selected and highlighted in blue. Below it, several sub-categories are listed: 124RX, 12-BS, 14A-A2, 14-BS, and 20L-BS. At the bottom of the tree view, there are sections for 'Advanced search' and 'Mappings'. On the right is the 'Node Details' panel, which shows the following information:

- Product Details** (Section Header)
- ID**: AC_PIESPCdbPartTerminology_2476_BBQH
- Name**: Vehicle Battery (in a text input field)
- Object Type**: PIES PCdb Part Terminology
- Approved**: ✗ Never been approved.
- Mfr Label**: (in a text input field with a pencil icon)
- Brand AAIA ID**: (in a text input field with a pencil icon)

At the bottom of the Node Details panel, there is a teal button with a play icon and the text 'Execute Mapper'.

After the user runs the mapping, the following changes are implemented in the system:

- New objects (Own part numbers) are created as child objects below the Target object (Target Part type) that is defined in the Mapper Configuration Setup.
- The new objects are created with ID pattern of Own_[BrandAAIAID]_[PIESItemName].
- The Name of the new objects are same as the PIES Item.
- The Object Type that is used for the new object is the one that is defined in the Object Type parameter available within the Mapping Target Definition flipper.
- The new Own part number is referenced to the PIES Item.

Tree

- ▶ AutoCare Root
- ▼ Primary Product Hierarchy
 - Products
 - Discontinued Products
 - Product Overrides
 - Packaging
- ▶ AutoCare Root
- ▶ TecDoc Supplier Root
- ▶ NAPA Hierarchy
- ▼ Own Root
 - ▼ Own Vehicle Battery
 - 124RX**
 - 12-BS
 - 14A-A2
 - 14-BS
 - 20L-BS

Node Details

Product Details

ID: Own_BBQH_124RX

Name:

Object Type: Own Part Number

Approved: ✗ Never been approved.

OWN Part To PIES Part

ID	Title	
AC_PIESItem_BBQH_124RX	124RX	✗

Advanced search

Mappings

▶ Execute Mapper

Use Cases - Copy Data from One Object to a Referenced Object of Another Object

Because the Object to Object mapping plugin within the Onboarding Mappings Details screen has many uses for how the mappings can be configured, describing the components within the mapper does not fully cover all of the functional use cases. With that in mind, two full use cases are described below that outline instances where this Object to Object Mapping plugin has been successfully implemented to copy data from one object to a referenced object of another object.

Example One

The requirement for this use case entails that the user has maintained part numbers (PIES Items) and Interchange objects (PIES Interchange objects) in the AutoCare Product hierarchy, but also wants to create the PIES Interchange objects as Interchange objects in the NAPA product hierarchy and establish the reference from the NAPA part number (NAPA Product) to the newly created interchange objects.

In this example, the AutoCare PIES Item 'JK21499' is used as the Source object and NAPA Product 'RPC12345' as the Target object. All referenced Interchange objects of Source object 'JK21499' will be used to create new objects in the selected Target Parent Node 'Great Lakes' (existing (or can be created) within NAPA Interchange Product Hierarchy) and establish references from the Target object 'RPC12345' to the new objects. The Source object and the Target object is interlinked through the reference type 'PIES Item to NAPA Product' in the system.

Following are the detailed steps describing how to achieve the same via the mapping solution.

Prerequisites

Create a new Product Reference Type to link the Source object PIES Item to the Target object NAPA Product and establish the reference between the PIES Item to the NAPA Product. In this example, the Product Reference Type 'PIES Item to NAPA Product' is defined to create a link between PIES Item and NAPA Product. Below is the image displaying hierarchy of Source object (JK21499), Target object (RPC12345), and the reference linking them.

The screenshot displays the Stibo Systems interface for managing references. On the left, a tree view shows the product hierarchy. On the right, the 'References' table shows the relationship between source and target objects.

Tree View:

- Packaging
 - AutoCare Root
 - ACES Application Notes
 - PIES Interchange Products
 - PIES Products
 - 034MOTORSPORT
 - BCVC
 - Behr Hella Service
 - Behr Thermot-Tronik
 - Exide
 - Hella
 - Pierburg
 - Wako
 - Engine
 - Sensors
 - Check Engine Light Sensor
 - JK21499** (Source object)
 - JK36004
 - JK36009
 - JK36112
 - VC21499
 - VC36004
 - VC36009
 - VC36112
 - TecDoc Supplier Root
 - NAPA Hierarchy
 - NAPA Comment Hierarchy
 - NAPA Interchange Product Hierarchy
 - NAPA Product Hierarchy
 - ATA Fuel Injector - New
 - RPC 63858
 - RPC12345** (Target object)
 - Wiper Blade - Bosch Icon - Driver's Side
- Own Root

References Table:

Reference Type	Target	ID	Name
PIES Interchange	GL26409	AC_PIESInterchangeItem_6...	GL26409
Part Relation			
PIES Item to NAPA Product	RPC12345		

Also, create a new Product Reference Type (or use the one that exists) to link the Target object NAPA Product and newly created objects. In this example, the existing Product Reference Type 'Product To Interchange Product' shall be used to create link between NAPA Product and newly created Interchange object (as shown below). For information about creating reference types and establishing references, see the **Reference Types** topic within the **System Setup / Super User Guide** documentation in **STEP Online Help**.

The screenshot displays the Stibo Systems Web UI interface. On the left, a 'Tree' view shows a hierarchical structure of product categories. The 'NAPA Product Hierarchy' is expanded, and the 'RPC12345' node is highlighted. A red arrow points from the text 'Target object' to this node. On the right, the 'RPC12345 rev.0.2 - References' panel is shown. It contains a table with the following structure:

Reference Type	Target
> Part Relation	
> Product To Interchange Product	

The 'Product To Interchange Product' row is highlighted with a red box. Below the table, there are several sections for 'Ungrouped Classification Links', 'Ungrouped Product References', 'Ungrouped Image & Document References', 'Index Words', 'Publications', 'Linked Attributes from Product Hierarchy', and 'Linked Attributes from Classification Hierarchy'.

In the Web UI, create and configure the Mapper Configuration and define the Source and Target objects and the Reference type to follow to determine the relationship between the Source and Target object. For more information on configuring and setup procedure for the Mapper Configuration, see topic **Configuring Mapper Configuration Setup Entity** within this guide.

PIES Item → NAPA Product

Mappings
Setup

Naming

* Map from

* Map to

Global Configuration

* Source

* Reference

* Target

Reverse Reference

Approved Workspace

Mapping description

Description

Save

 Duplicate

Delete
Reset

Configure an Initiate Business Action button within the Product Details screen of the Source object which shall execute the Mapper Configuration. For more information on executing the Mapper Configuration, see topic **Executing Mapper Configuration Setup Entity** within this guide.

1. Add a new Object to Object Mapping plugin (or edit the one that exists) in the required Mapper Configuration.
2. Type in a suitable name next to Name field. In this example, the Name field is entered with the text 'PIES Item to NAPA Product Reference Mapping Type.'

PIES Item → NAPA Product

Mappings Setup

← Object to Object Mapping

Name

▼ Target Definition

Mapping type

Object Type

Reference Type

Parent Node

+ Add object mapping

3. Expand the Target Definition flipper, and select 'Reference' option in the dropdown next to 'Mapping type' field.
4. Choose the Object Type that should be used for the new objects that are getting created. In this example, object type with ID 'NAPA_InterchangeProduct' is selected.
5. Choose the Reference Type that should be used to reference from the Target object to the newly created object. In this example, an existing reference type with ID 'NAPA_ProductToInterchangeProduct' is selected.
6. Choose the Parent Node where the new objects will be created below. In this example, an existing node from NAPA hierarchy 'Great Lakes' with ID 'NAPA_InterchangeManufacturer_5692' is selected.
7. Click the 'Add object mapping' icon, and the Mapping Guide window will display.

Mapping Guide

Mapping

ID	Name	Validation Type
Source		
Target		
Transformation		


Targets Transformations

Search: Filter Options: Attributes References

Select Attribute

Node Type	ID	Title	Validation Type	Attribute Group(S)
Attribute	ChangedObject	Changed Object	Text	Metadata
Attribute	DeleteStatus	Delete Status	Text	Metadata
Attribute	NewObject	New Object	Text	Metadata

Cancel OK

8. Click on the Edit icon () in the Source field, and the Step Path Editor window will display (as shown below). For more information on defining the Source STEP path, see topic **Mapping Validation Path Functionality** within this guide.

Step Path Editor

Starting Points: PIES Item

Validation Path:

Keywords: attribute[id:'[id]'] child[objecttype:'[id]'] id name parent reference[type:'[id]'] text['[text]']

Information

OK Cancel

9. For this use case, since there are multiple functions required to be executed, the following series of logic is used to accomplish the requirement through the Step Path Editor.

- To define what to use for the ID of the new objects that will get created.

In this example, the new objects that are going to get created need to have an ID pattern of NAPA_Interchange_[PIESInterchangeName]. To achieve the prefix of 'NAPA_Interchange_' in the ID, we are using the 'text' element in the STEP path > enter the static text of 'NAPA_Interchange_' > select 'Use in ID' checkbox (shown below).

PIES Item → NAPA Product

Mappings Setup

← **Object to Object Mapping**

Name

▼ Target Definition

Mapping type

Object Type

Reference Type

Parent Node

+ Add object mapping

Source	Use in ID	Use in Name	Hash	Use in Parent	Hash Parent	Transformation	Target	Validation Type
<input type="checkbox"/> text['NAPA_Interchange_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	←		<input checked="" type="checkbox"/> +

- To define what to use for the Name of the new objects that are getting created, as well to get the PIES Interchange object name value in the ID of the new objects that will get created.

In this example, the PIES Interchange object name will be used in the ID and the Name of the new objects. So, using the **reference[type:'id']** and **name** elements in the STEP path will follow defined referenced objects of the Source object and retrieve the PIES Interchange object name. The 'Use in ID' and 'Use in Name' checkboxes are selected (as shown below).

PIES Item → NAPA Product

Mappings Setup

← Object to Object Mapping

Name: PIES Item to NAPA Product Reference Mapping Type

▼ Target Definition

Mapping type: Reference

Object Type: NAPA Interchange Product (NAPA_InterchangeProduct)

Reference Type: Product To Interchange Product (NAPA_ProductToInterchangeProd)

Parent Node: Great Lakes (NAPA_InterchangeManufacturer_5692)

+ Add object mapping

Find sources, targets and transformations

Source	Use in ID	Use in Name	Hash	Use in Parent	Hash Parent	Transformation	Target	Validation Type
<input type="checkbox"/> text['NAPA_Interchange_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
<input type="checkbox"/> reference[type:'AC_PIESInterchange'].name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

Save Duplicate Delete Reset

10. Click **OK** to save and close the Mapping Guide window. The newly added object mapping row will be listed.

PIES Item → NAPA Product

Mappings Setup

← Object to Object Mapping

Name: PIES Item to NAPA Product Reference Mapping Type

▼ Target Definition

Mapping type: Reference

Object Type: NAPA Interchange Product (NAPA_InterchangeProduct)

Reference Type: Product To Interchange Product (NAPA_ProductToInterchangeProd)

Parent Node: Great Lakes (NAPA_InterchangeManufacturer_5692)

+ Add object mapping

Find sources, targets and transformations

Source	Use in ID	Use in Name	Hash	Use in Parent	Hash Parent	Transformation	Target	Validation Type
<input type="checkbox"/> text['NAPA_Interchange_']	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
<input type="checkbox"/> reference[type:'AC_PIESInterchange'].name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

- In the Web UI, standing on the Source object, execute the Mapper Configuration that holds the mapping plugin. For more information on executing the Mapper Configuration, see topic **Executing Mapper Configuration Setup Entity** within this guide.

Tree

- ▶ AutoCare Root
- ▼ Primary Product Hierarchy
 - Products
 - Discontinued Products
 - Product Overrides
 - Packaging
- ▼ AutoCare Root
 - ▶ ACES Application Notes
 - ▶ PIES Interchange Products
 - ▼ PIES Products
 - ▶ 034MOTORSPORT
 - ▶ BCVC
 - ▶ Behr Hella Service
 - ▶ Behr Thermot-Tronik
 - ▶ Exide
 - ▶ Hella
 - ▶ Pierburg
 - ▼ Wako
 - ▼ Engine
 - ▼ Sensors
 - ▼ Check Engine Light ...
 - ▶ **JK21499**
 - ▶ JK36004
 - ▶ JK36009
 - ▶ JK36112
 - ▶ VC36004
 - ▶ VC36009
 - ▶ VC36112
 - ▶ TecDoc Supplier Root

Advanced search


Mappings

Node Details

Product Details

ID: AC_PIESItem_DKGX_JK21499

Name:

PIES Item To Product Image: 
 VC21499_P04

Object Type: PIES Item

PIES Item to NAPA Product:

ID	Title
NAPA_Product_RPC12345	RPC12345 ✕

Approved: ✕ Never been approved.

PIES Interchange:

ID	Title	+
AC_PIESInterchangeItem_6e509aabf7573f74b	GL26409	✕

ACES Mfr Body Code:

Mfr Label:

Connector Gender:

Connector Shape:

▶ Execute Mapper

After the user runs the mapping, the following changes are implemented in the system.

- New objects (Interchange products) are created as child objects below the Parent Node (Great Lakes) as defined in the Mapper Configuration Setup.
- The new objects are created with an ID pattern as NAPA_Interchange_[PIESInterchangeName].
- The Name of the new objects are the same as the PIES Interchange.
- The Object Type that is used for the new object is the one that is defined in the Object Type parameter available within the Mapping Target Definition flipper.

Tree

- ▶ AutoCare Root
- ▼ Primary Product Hierarchy
 - ▶ Products
 - ▶ Discontinued Products
 - ▶ Product Overrides
 - ▶ Packaging
- ▶ AutoCare Root
- ▶ TecDoc Supplier Root
- ▼ NAPA Hierarchy
 - ▶ NAPA Comment Hierarchy
 - ▼ NAPA Interchange Product Hierarchy
 - ▼ Great Lakes
 - ▶ GL26409
- ▶ NAPA Product Hierarchy
 - ▶ ATA Fuel Injector - New
 - ▼ RPC 63858
 - ▶ RPC12345
 - ▶ Wiper Blade - Bosch Icon - Driver's Si...
- ▶ Own Root

- ▶ TecDoc Supplier Price Lists
- ▶ Import Flow Root
- ▶ AutoCare Root
- ▶ Target Vehicle Model

Advanced search

Mappings

Node Details

Product Details

ID	NAPA_Interchange_GL26409
Name	<input style="width: 90%;" type="text" value="GL26409"/>
Object Type	NAPA Interchange Product
Approved	✗ Never been approved.

▶ Execute Mapper

- The NAPA Product that was referenced by the PIES Item will be referenced to the newly created NAPA Interchange part using the reference type that is defined in the Reference Type parameter.

Tree

- ▶ AutoCare Root
- ▼ Primary Product Hierarchy
 - Products
 - Discontinued Products
 - Product Overrides
 - Packaging
- ▶ AutoCare Root
- ▶ TecDoc Supplier Root
- ▼ NAPA Hierarchy
 - ▶ NAPA Comment Hierarchy
 - ▼ NAPA Interchange Product Hierarchy
 - ▼ Great Lakes
 - GL26409
 - ▼ NAPA Product Hierarchy
 - ▶ ATA Fuel Injector - New
 - ▼ RPC 63858
 - RPC12345**
 - ▶ Wiper Blade - Bosch Icon - Driver's Si..
 - ▶ Own Root
- ▶ TecDoc Supplier Price Lists
- ▶ Import Flow Root
- ▶ AutoCare Root
- ▶ Target Vehicle Model

Advanced search

Mappings

Node Details

Product Details

ID	NAPA_Product_RPC12345						
Name	<input type="text" value="RPC12345"/>						
Object Type	NAPA Product						
Product To Interchange Product	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">ID</th> <th style="width: 50%;">Title</th> <th style="width: 20px; text-align: center;">+</th> </tr> </thead> <tbody> <tr> <td>NAPA_Interchange_GL26409</td> <td>GL26409</td> <td style="text-align: center;">×</td> </tr> </tbody> </table>	ID	Title	+	NAPA_Interchange_GL26409	GL26409	×
ID	Title	+					
NAPA_Interchange_GL26409	GL26409	×					
Approved	✗ Never been approved.						
Part Number To Image	+						
# Steering gear Box Mounting	<input type="text"/>						
# Steering Gear Box Turns Lock Lock	<input type="text"/>						
Color	<input type="text"/>						
Contents	<input type="text"/>						
E-Waste	<input type="text"/>						

→

▶ Execute Mapper